



Red Hat build of MicroShift 4.16

API 参考

MicroShift API 参考

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档提供有关 MicroShift 中的 REST API 功能的详细信息。

目录

第 1 章 了解 API 层	5
1.1. API 层	5
1.2. 将 API 层映射到 API 组	5
1.3. API 弃用策略	6
第 2 章 了解 API 兼容性指南	9
2.1. API 兼容性指南	9
2.2. API 兼容性例外	9
2.3. API 兼容性常见术语	10
第 3 章 API 索引	12
第 4 章 API 对象引用	15
4.1. 通用对象引用	15
第 5 章	212
5.1.	212
5.2. CUSTOMRESOURCEDEFINITION [APIEXTENSIONS.K8S.IO/V1]	212
第 6 章 REGISTRATION APIS	238
6.1. API 注册 API	238
6.2. APISERVICE [APIREGISTRATION.K8S.IO/V1]	238
第 7 章 APPS API	261
7.1. APPS API	261
7.2. CONTROLLERREVISION [APPS/V1]	261
7.3. DAEMONSET [APPS/V1]	285
7.4. DEPLOYMENT [APPS/V1]	317
7.5. REPLICASET [APPS/V1]	348
7.6. STATEFULSET [APPS/V1]	377
第 8 章 身份验证 API	411
8.1. 身份验证 API	411
8.2. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]	411
8.3. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]	415
第 9 章 授权 API	420
9.1. 授权 API	420
9.2. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	420
9.3. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	426
9.4. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]	430
9.5. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]	436
第 10 章 自动缩放 API	442
10.1. 自动缩放 API	442
10.2. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]	442
10.3. SCALE [AUTOSCALING/V1]	493
第 11 章 批处理 API	508
11.1. 批处理 API	508
11.2. CRONJOB [BATCH/V1]	508
11.3. 作业 [BATCH/V1]	544
第 12 章 证书 API	581

12.1. 证书 API	581
12.2. CERTIFICATESIGNINGREQUEST [CERTIFICATES.K8S.IO/V1]	581
第 13 章 协调 API	611
13.1. 协调 API	611
13.2. LEASE [COORDINATION.K8S.IO/V1]	611
第 14 章 核心 API	635
14.1. 核心 API	635
14.2. 绑定 [V1]	638
14.3. COMPONENTSTATUS [V1]	642
14.4. CONFIGMAP [V1]	647
14.5. ENDPOINTS [V1]	671
14.6. EVENT [V1]	701
14.7. LIMITRANGE [V1]	728
14.8. 命名空间 [V1]	753
14.9. 节点 [V1]	774
14.10. PERSISTENTVOLUME [V1]	808
14.11. PERSISTENTVOLUMECLAIM [V1]	860
14.12. POD [V1]	894
14.13. PODTEMPLATE [V1]	1145
14.14. REPLICATIONCONTROLLER [V1]	1365
14.15. RESOURCEQUOTA [V1]	1591
14.16. SECRET [V1]	1620
14.17. SERVICE [V1]	1644
14.18. SERVICEACCOUNT [V1]	1684
第 15 章 DISCOVERY API	1711
15.1. DISCOVERY API	1711
15.2. ENDPOINTSlice [DISCOVERY.K8S.IO/V1]	1711
第 16 章 EVENTS API	1740
16.1. EVENTS API	1740
16.2. EVENT [EVENTS.K8S.IO/V1]	1740
第 17 章 流控制 API	1766
17.1. 流控制 API	1766
17.2. FLOWSchema [FLOWCONTROL.APISERVER.K8S.IO/V1BETA3]	1766
17.3. PRIORITYLEVELCONFIGURATION [FLOWCONTROL.APISERVER.K8S.IO/V1BETA3]	1794
第 18 章 网络 API	1820
18.1. 网络 API	1820
18.2. INGRESS [NETWORKING.K8S.IO/V1]	1820
18.3. INGRESSCLASS [NETWORKING.K8S.IO/V1]	1857
18.4. NETWORKPOLICY [NETWORKING.K8S.IO/V1]	1876
第 19 章 节点 API	1912
19.1. 节点 API	1912
19.2. RUNTIMECLASS [NODE.K8S.IO/V1]	1912
第 20 章 策略 API	1931
20.1. 策略 API	1931
20.2. 驱除 [POLICY/V1]	1931
20.3. PODDISRUPTIONBUDGET [POLICY/V1]	1933
第 21 章 RBAC API	1964

21.1. RBAC API	1964
21.2. CLUSTERROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	1964
21.3. CLUSTERROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	1983
21.4. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]	2001
21.5. ROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]	2025
第 22 章 网络 API	2051
22.1. 网络 API	2051
22.2. ROUTE [ROUTE.OPENSIFT.IO/V1]	2051
第 23 章 调度 API	2076
23.1. 调度 API	2076
23.2. PRIORITYCLASS [SCHEDULING.K8S.IO/V1]	2076
第 24 章 安全 API	2095
24.1. 安全 API	2095
24.2. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSIFT.IO/V1]	2095
第 25 章 SECURITY-INTERNAL API	2111
25.1. 安全内部 API	2111
25.2. RANGEALLOCATION [SECURITY.INTERNAL.OPENSIFT.IO/V1]	2111
第 26 章 快照 API	2123
26.1. 快照 API	2123
26.2. VOLUMESNAPSHOT [SNAPSHOT.STORAGE.K8S.IO/V1]	2123
26.3. VOLUMESNAPSHOTCLASS [SNAPSHOT.STORAGE.K8S.IO/V1]	2145
26.4. VOLUMESNAPSHOTCONTENT [SNAPSHOT.STORAGE.K8S.IO/V1]	2157
第 27 章 存储 API	2178
27.1. 存储 API	2178
27.2. CSIDRIVER [STORAGE.K8S.IO/V1]	2179
27.3. CSINODE [STORAGE.K8S.IO/V1]	2202
27.4. CSISTORAGECAPACITY [STORAGE.K8S.IO/V1]	2222
27.5. STORAGECLASS [STORAGE.K8S.IO/V1]	2246
27.6. VOLUMEATTACHMENT [STORAGE.K8S.IO/V1]	2265
第 28 章 存储版本迁移 API	2288
28.1. 存储版本迁移 API	2288
28.2. STORAGEVERSIONMIGRATION [MIGRATION.K8S.IO/V1ALPHA1]	2288
第 29 章 TOPOLVM API	2305
29.1. TOPOLVM API	2305
29.2. LOGICALVOLUME [TOPOLVM.IO/V1]	2305
第 30 章 WEBHOOK API	2322
30.1. WEBHOOK API	2322
30.2. MUTATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]	2322
30.3. VALIDATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]	2350

第1章 了解 API 层



重要

这个指南不包括层次的红帽构建的 MicroShift 产品。

红帽要求应用程序开发人员验证他们所依赖的所有行为是否在正式的 API 文档中明确定义，以防止引入特定实施特定行为或对 API 特定实施错误的依赖项。例如，如果应用程序使用未文档化的 API 或依赖于未定义行为，则入口路由器的新版本可能无法与旧的版本兼容。

1.1. API 层

所有商业支持的 API、组件和功能都与以下支持级别之一相关联：

API 层 1

API 和应用程序操作环境(AOE)在主发行版本中保持稳定。它们在主版本中可能被弃用，但在后续的主发行版本前不会删除它们。

API 层 2

API 和 AOE 在主发行版本中至少为 9 个月或 3 个次版本的发布周期（以更长的时间为准）。

API 层 3

此级别适用于通过 Operator Hub 的红帽构建 MicroShift 附带的语言、工具、应用程序和可选 Operator。每个组件都将指定一个生命周期，在其中支持 API 和 AOE。较新的语言运行时组件版本将尽量作为 API 和 AOE 兼容次版本。但是，无法保证次版本的兼容性。

通过 Operator Hub 接收持续更新的组件和开发人员工具（称为 Operator 和操作对象）应该被视为 API 层 3。开发人员应谨慎使用，并了解这些组件如何在每个次发行版本中改变。我们鼓励用户参考组件所记录的兼容性指南。

API 层 4

不提供兼容性。API 和 AOE 可在任何时候更改。这些功能不应由需要长期支持的应用程序使用。

Operator 在内部使用自定义资源定义 (CRD) 来完成任务是常见的。这些对象并应由 Operator 外部执行者使用，它旨在隐藏。如果任何 CRD 没有被 Operator 外部执行者使用，则 Operator **ClusterServiceVersion (CSV)** 中的 **operators.operatorframework.io/internal-objects** 注解应指定给对应资源使用范围，并且 CRD 可能会明确标记为层 4。

1.2. 将 API 层映射到 API 组

对于由红帽定义的每个 API 层，我们为上游社区致力于保持向前兼容性的特定 API 组提供映射表。默认情况下，任何没有指定显式兼容性级别且未特别讨论的 API 组都会被分配 API 层 3，但 **v1alpha1** API 默认被分配层 4。

1.2.1. 支持 Kubernetes API 组

以后缀 ***.k8s.io** 结尾的 API 组，或带有任何后缀的 **version.<name>** 均由 Kubernetes 弃用策略管理，并遵循公开和对应的支持层之间的通用映射，除非另有指定。

API 版本示例	API 层
v1	1 级

API 版本示例	API 层
v1beta1	2 级
v1alpha1	4 级

1.2.2. 支持 OpenShift API 组

以后缀以 **openshift.io** 结尾的 API 组由红帽构建的 MicroShift 弃用策略管理，并遵循公开和对应的兼容性级别之间的通用映射，除非另有指定。

API 版本示例	API 层
route.openshift.io/v1	1 级
security.openshift.io/v1	1 级，除 RangeAllocation (4 级) 和 *Reviews (2 级)

1.3. API 弃用策略

红帽构建的 MicroShift 由许多来自上游社区的组件组成。因此，我们预计许多组件、关联的 API 接口和关联的功能集合将随着时间推移而有所变化，并且在功能被完成删除前会提供一个弃用的阶段。

1.3.1. 弃用的 API 部分

红帽构建的 MicroShift 是一个分布式系统，多个组件通过一组结构化 API 与由集群 control plane 管理的共享状态进行交互。根据 Kubernetes 惯例，红帽构建的 MicroShift 提供的每个 API 都与组标识符关联，每个 API 组都是独立的版本。每个 API 组都在不同的上游社区中管理，包括 Kubernetes、Metal3、Multus、Operator Framework、Open Cluster Management、OpenShift 本身等。

虽然每个上游社区可能会为给定的 API 组和版本定义自己的唯一弃用策略，但红帽会将社区特定策略规范化为之前定义的兼容性级别之一，具体取决于我们的集成和认知，以简化最终用户消费和支持。

API 的弃用策略和调度因兼容性级别而异。

弃用策略涵盖 API 的所有元素，包括：

- REST 资源，也称为 API 对象
- REST 资源的字段
- REST 资源的注解，不包括特定于版本的限定符
- Enumerated 或 constant 值

除每个组中最新的 API 版本外，在宣布弃用后，必须支持旧的 API 版本，且时间不少于：

API 层	Duration
1 级	在一个主版本内是稳定的。它们在主版本中可能被弃用，但在后续的主发行版本前不会删除它们。
2 级	从公布弃用后的 9 个月或 3 个发行版本（以更长的时间为准）。
3 级	请参阅特定于组件的调度。
4 级	无。无法保证兼容性。

以下规则适用于所有 1 级 API：

- API 元素只能通过递增组版本来删除。
- API 对象必须能够在没有信息丢失的情况下在 API 版本之间往返，除非整个 REST 资源在某些版本中不存在。如果在版本之间没有对等的字段，数据将在转换过程中以注解的形式保留。
- 一个特定组中的 API 版本在新的 API 版本至少发行了一个稳定的版本前不会被弃用，除非整个 API 对象已被删除。

1.3.2. 弃用的 CLI 元素

对于面向客户端的 CLI 命令，它们的版本模式与 API 不同，它们是面向用户的组件系统。用户与 CLI 交互的两种主要方法是通过命令或标志（在此上下文中称为 CLI 元素）。

除非另有说明，否则所有 CLI 元素都默认为 API 层 1，除非另有说明或 CLI 依赖于较低层 API。

	元素	API 层
正式发布(GA)	标志和命令	1 级
技术预览	标志和命令	3 级
开发者预览	标志和命令	4 级

1.3.3. 弃用整个组件

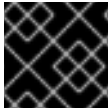
弃用整个组件的持续时间和调度直接映射到与该组件公开的 API 的最高 API 层关联的持续时间。例如，在满足第 1 层弃用计划前，无法删除带有级别 1 和 2 的 API 的组件。

API 层	Duration
1 级	在一个主版本内是稳定的。它们在主版本中可能被弃用，但在后续的主发行版本前不会删除它们。

API 层	Duration
2 级	从公布弃用后的 9 个月或 3 个发行版本（以更长的时间为准）。
3 级	请参阅特定于组件的调度。
4 级	无。无法保证兼容性。

第 2 章 了解 API 兼容性指南

按照兼容性指南以了解为 MicroShift 的红帽构建启用的 API。



重要

这个指南不包括层次的红帽构建的 MicroShift 产品。

2.1. API 兼容性指南

红帽建议应用程序开发人员使用以下原则来改进与红帽构建的 MicroShift 的兼容性：

- 使用与应用需求匹配的 API 和组件，支持层。
- 尽可能使用公布的客户端库构建应用程序。
- 只有应用程序在作为新环境构建的环境中执行时，才能保证正确运行。为红帽构建的 MicroShift 4.14 构建的应用程序无法保证在红帽构建的 MicroShift 4.13 上正常工作。
- 不要设计依赖于系统软件包或其他组件提供的配置文件的应用程序。除非上游社区明确提交来保留它们，否则这些文件可以在不同版本之间更改。在适当的情况下，依靠任何红帽在这些配置文件中提供接口抽象，以便保持向前兼容性。不鼓励直接修改配置文件，强烈建议用户与提供的 API 集成，以避免双写器冲突。
- 不要依赖于以 **unsupported<FieldName>** 或在产品文档中未明确提及的注解作为前缀的 API 字段。
- 不要依赖于比您的应用程序更短的兼容性保证。
- 不要在 etcd 服务器上执行直接存储操作。所有 etcd 访问都必须通过 api-server 或记录的备份和恢复步骤执行。

红帽建议应用程序开发人员遵循 Red Hat Enterprise Linux (RHEL) 定义的[兼容性指南](#)。红帽构建的 MicroShift 强烈推荐以下准则，在构建应用程序或在此平台上托管应用程序：

- 不要依赖于特定的 Linux 内核或 MicroShift 版本的红帽构建。
- 避免从 **proc**、**sys** 和 **debug** 文件系统，或任何其他伪文件系统读取。
- 避免使用 **ioctl**s 直接与硬件交互。
- 避免直接与 **cgroups** 交互，以便不与红帽构建的 MicroShift 主机代理冲突，后者提供容器执行环境。



注意

在发行版本的生命周期中，红帽采取商业合理的努力来维护所有次发行版本和 z-stream 版本中的 API 和应用程序操作环境 (AOE) 兼容性。但是如有必要，红帽会就安全影响级别为“关键 (Critical)”的安全问题或者其他严重问题在兼容性方面做例外处理。

2.2. API 兼容性例外

以下是红帽构建的 MicroShift 中兼容性的例外：

升级的集群和新安装之间的功能默认

目前，产品次版本的新安装功能与之前次版本一起安装的产品版本相同，不会进行保证。例如，未来版本的产品可能会置备云基础架构，其默认值与之前的次版本不同。此外，未来版本的产品中可能会进行不同的默认安全选择。产品的过去版本将会向前升级，但仍然保留了旧的选择，以便保持向后兼容性。

使用前缀 "unsupported" 或 undocumented 注解的 API 字段

在产品公开字段中，选择带有前缀 **unsupported<FieldName>** 的 API。不保证，目前在不同版本间或一个版本内支持使用这个字段。产品支持可在调试特定问题时请求在此字段中指定值，但在该交互之外不支持其使用。在未明确记录的对象中使用注解在次版本间不会保证支持。

每个产品安装拓扑的 API 可用性

OpenShift 发行版将继续演进其支持的安装拓扑，且不是一个安装拓扑中的所有 API 都必须包含在另一个安装中。例如，如果某些拓扑与产品安装拓扑冲突，或者根本不包括特定的 API，则某些拓扑可能会限制对特定 API 的读/写访问。根据上面定义的兼容性层，支持给定拓扑中存在的 API。

2.3. API 兼容性常见术语

2.3.1. 应用程序编程接口 (API)

API 是一个由软件程序实现的公共接口，它允许它与其他软件交互。在 MicroShift 的红帽构建中，API 从集中式 API 服务器提供，用作所有系统交互的中心。

2.3.2. 应用程序操作环境 (AOE)

AOE 是执行最终用户应用程序程序的集成环境。AOE 是一个容器化环境，提供与主机操作系统 (OS) 隔离的。至少，AOE 允许应用程序以独立于主机操作系统库和二进制文件运行，但仍然与主机上的所有其他容器共享相同的 OS 内核。AOE 在运行时强制执行，它描述了应用程序及其操作环境之间的接口。它包括平台操作系统和环境的交集点，以及用户应用程序，包括 Downward API、DNS、资源核算、设备访问、平台工作负载身份、容器之间隔离、容器和主机操作系统之间的隔离。

AOE 不包括安装可能因安装而异的组件，如 Container Network Interface (CNI) 插件选择或对产品的扩展，如准入 hook。在容器环境下面的级别与集群集成的组件可能会受到不同版本之间的其他变化。

2.3.3. 虚拟化环境中的兼容性

虚拟环境模拟裸机环境，这样在裸机环境中运行的非特权应用程序将在对应的虚拟环境中运行。虚拟环境改进了物理资源的抽象视图，因此可能存在一些差异。

2.3.4. 云环境中的兼容性

红帽构建的 MicroShift 可能会选择通过云供应商特定集成提供与托管云环境的集成点。这些集成点的兼容性特定于原生云供应商提供的保证，及其与红帽构建的 MicroShift 兼容性窗口交集。如果红帽构建的 MicroShift 作为默认安装的一部分原生地与云环境集成，红帽针对稳定的云 API 端点开发，以提供合理的商业支持，包括稳定的弃用策略。云供应商与红帽构建的 MicroShift 之间的集成示例包括：但不仅限于动态卷置备、服务负载均衡器集成、pod 工作负载身份、计算的动态管理，以及作为初始安装的一部分置备的基础架构。

2.3.5. 主版本、次版本和 z-stream 版本

红帽主发行版本代表了开发产品的显著步骤。次发行版本会出现在主发行版本范围内，并代表弃用界限可能会影响将来的应用程序兼容性。z-stream 版本是次版本的更新，它为关联的次版本提供持续修复流。API 和 AOE 兼容性永远不会在 z-stream 版本中中断，除非明确覆盖此策略以响应不可预见的安全影响。

例如，在版本 4.13.2 中：

- 4 是主版本
- 13 是次版本
- 2 是 z-stream 版本

2.3.6. 扩展的用户支持 (EUS)

一个 Red Hat build of MicroShift 主发行版本中的一个次发行版本，它有一个对关键程序错误修复的延长支持窗口。通过在 EUS 版本之间使用次版本，用户可以在 EUS 版本之间迁移 EUS 版本。请注意，弃用策略是在次发行版本中定义，而不是 EUS 版本。因此，在通过每个次版本按顺序升级时，EUS 用户可能需要在迁移到将来的 EUS 时响应弃用。

2.3.7. 开发者预览

可选的产品功能不受红帽官方支持，但旨在提供探索早期阶段技术的机制。默认情况下，开发人员预览功能是 opt-in，可以随时删除。启用开发人员预览功能可能会导致集群无法支持，这取决于该功能的范围。

2.3.8. 技术预览

提供即将推出的产品创新的可选产品功能，以测试功能并在开发过程中提供反馈。这个功能没有被完全支持，可能无法正常工作，且不适用于生产环境。使用技术预览功能需要明确选择。了解有关 [技术预览功能支持范围](#) 的更多信息。

第 3 章 API 索引

API	API 组
APIService	apiregistration.k8s.io/v1
绑定	v1
CertificateSigningRequest	certificates.k8s.io/v1
ClusterRole	rbac.authorization.k8s.io/v1
ClusterRoleBinding	rbac.authorization.k8s.io/v1
ComponentStatus	v1
ConfigMap	v1
ControllerRevision	apps/v1
CronJob	batch/v1
CSIDriver	storage.k8s.io/v1
CSINode	storage.k8s.io/v1
CSIStorageCapacity	storage.k8s.io/v1
CustomResourceDefinition	apiextensions.k8s.io/v1
DaemonSet	apps/v1
Deployment	apps/v1
Endpoints	v1
EndpointSlice	discovery.k8s.io/v1
事件	v1
事件	events.k8s.io/v1
驱除	policy/v1
FlowSchema	flowcontrol.apiserver.k8s.io/v1beta3
HorizontalPodAutoscaler	autoscaling/v2

API	API 组
入口	networking.k8s.io/v1
IngressClass	networking.k8s.io/v1
作业	batch/v1
Lease	coordination.k8s.io/v1
LimitRange	v1
LocalSubjectAccessReview	authorization.k8s.io/v1
LogicalVolume	topolvm.io/v1
MutatingWebhookConfiguration	admissionregistration.k8s.io/v1
命名空间	v1
NetworkPolicy	networking.k8s.io/v1
节点	v1
PersistentVolume	v1
PersistentVolumeClaim	v1
Pod	v1
PodDisruptionBudget	policy/v1
PodTemplate	v1
PriorityClass	scheduling.k8s.io/v1
PriorityLevelConfiguration	flowcontrol.apiserver.k8s.io/v1beta3
RangeAllocation	security.internal.openshift.io/v1
ReplicaSet	apps/v1
ReplicationController	v1
ResourceQuota	v1
角色	rbac.authorization.k8s.io/v1

API	API 组
RoleBinding	rbac.authorization.k8s.io/v1
Route	route.openshift.io/v1
RuntimeClass	node.k8s.io/v1
扩展	autoscaling/v1
Secret	v1
SecurityContextConstraints	security.openshift.io/v1
SelfSubjectAccessReview	authorization.k8s.io/v1
SelfSubjectRulesReview	authorization.k8s.io/v1
Service	v1
ServiceAccount	v1
StatefulSet	apps/v1
StorageClass	storage.k8s.io/v1
StorageVersionMigration	migration.k8s.io/v1alpha1
SubjectAccessReview	authorization.k8s.io/v1
TokenRequest	authentication.k8s.io/v1
TokenReview	authentication.k8s.io/v1
ValidatingWebhookConfiguration	admissionregistration.k8s.io/v1
VolumeAttachment	storage.k8s.io/v1
VolumeSnapshot	snapshot.storage.k8s.io/v1
VolumeSnapshotClass	snapshot.storage.k8s.io/v1
VolumeSnapshotContent	snapshot.storage.k8s.io/v1

第 4 章 API 对象引用

4.1. 通用对象引用

4.1.1. io.k8s.api.admissionregistration.v1.MutatingWebhookConfigurationList schema

描述

MutatingWebhookConfigurationList 是 MutatingWebhookConfiguration 列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (MutatingWebhookConfiguration)	MutatingWebhookConfiguration 列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.2. io.k8s.api.admissionregistration.v1.ValidatingWebhookConfigurationList schema

描述

ValidatingWebhookConfigurationList 是 ValidatingWebhookConfiguration 列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (ValidatingWebhookConfiguration)	ValidatingWebhookConfiguration 列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.3. io.k8s.api.apps.v1.ControllerRevisionList schema

描述

ControllerRevisionList 是一个包含 ControllerRevision 对象列表的资源。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ControllerRevision)	items 是 ControllerRevisions 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.4. io.k8s.api.apps.v1.DaemonSetList schema

描述

DaemonSetList 是守护进程集的集合。

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(DaemonSet)	守护进程集列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法(CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.5. io.k8s.api.apps.v1.DeploymentList schema

描述

DeploymentList 是一个 Deployments 列表。

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Deployment)	items 是 Deployments 列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。

4.1.6. io.k8s.api.apps.v1.ReplicaSetList schema

描述

ReplicaSetList 是 ReplicaSets 的集合。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
items	数组(ReplicaSet)	ReplicaSets 列表。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.7. io.k8s.api.apps.v1.StatefulSetList schema

描述

StatefulSetList 是 StatefulSets 的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(StatefulSet)	items 是有状态的集合的列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表的元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.8. io.k8s.api.autoscaling.v2.HorizontalPodAutoscalerList schema

描述

HorizontalPodAutoscalerList 是 pod 横向自动扩展对象的列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (HorizontalPodAutoscaler)	items 是 pod 横向自动扩展对象的列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	metadata 是标准列表元数据。

4.1.9. io.k8s.api.batch.v1.CronJobList schema

描述

CronJobList 是 cron 任务的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(CronJob)	items 是 CronJobs 的列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.10. io.k8s.api.batch.v1.JobList schema

描述

JobList 是作业的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Job)	items 是作业列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.11. io.k8s.api.certificates.v1.CertificateSigningRequestList schema

描述

CertificateSigningRequestList 是 CertificateSigningRequest 对象的集合

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (CertificateSigningRequest)	items 是 CertificateSigningRequest 对象的集合
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	

4.1.12. io.k8s.api.coordination.v1.LeaseList schema

描述

LeaseList 是 Lease 对象的列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Lease)	items 是 schema 对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.13. io.k8s.api.core.v1.Affinity schema

描述

affinity 是一组关联性调度规则。

类型

对象

模式

属性	类型	描述
nodeAffinity	nodeAffinity	描述 pod 的节点关联性调度规则。
podAffinity	podAffinity	描述 pod 关联性调度规则（例如，将这个 pod 放置到同一节点、区域等）中。
podAntiAffinity	podAntiAffinity	描述 pod 反关联性调度规则（例如，避免将此 pod 放置到同一节点、区域等）中。

属性	类型	描述
----	----	----

4.1.14. io.k8s.api.core.v1.AWSElasticBlockStoreVolumeSource schema

描述

代表 AWS 中的 Persistent Disk 资源。

挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumeID**

模式

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。
readOnly	布尔值	readonly 值 true 将强制 VolumeMounts 中的 readOnly 设置。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

属性	类型	描述
volumelD	string	VolumelD 是 AWS (Amazon EBS 卷) 中的持久磁盘资源的唯一 ID。 更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

4.1.15. io.k8s.api.core.v1.AzureDiskVolumeSource schema

描述

azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

类型

object

必填

- **diskName**
- **diskURI**

模式

属性	类型	描述
cachingMode	string	cachingMode 是主机缓存模式： None、Read Only、Read Write。 可能枚举值：- "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskname 是 blob 存储中的数据磁盘的名称
diskURI	string	diskURI 是 blob 存储中数据磁盘的 URI
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。

属性	类型	描述
kind	string	kind expected 值是 Shared: multiple blob disk per storage account Managed: azure disk per storage account Managed: azure managed data disk (only in managed availability set))。默认为 shared possible enum 值 : - "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。

4.1.16. io.k8s.api.core.v1.AzureFilePersistentVolumeSource schema

描述

azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

模式

属性	类型	描述
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称
secretNamespace	string	secretNamespace 是包含 Azure Storage Account Name 和 Key default 的 secret 的命名空间与 Pod 相同

属性	类型	描述
shareName	string	sharename 是 azure Share Name

4.1.17. io.k8s.api.core.v1.AzureFileVolumeSource schema

描述

azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

模式

属性	类型	描述
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称
shareName	string	sharename 是 azure 共享名称

4.1.18. io.k8s.api.core.v1.Capabilities schema

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

对象

模式

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

4.1.19. io.k8s.api.core.v1.CephFSPersistentVolumeSource schema

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

模式

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选的：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	secretReference	secretRef 是可选的：SecretRef 是对 User 的身份验证 secret 的引用，默认为空。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

4.1.20. io.k8s.api.core.v1.CephFSVolumeSource schema

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

模式

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	LocalObjectReference	secretRef 是可选的：SecretRef 是对 User 的身份验证 secret 的引用，默认为空。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

属性	类型	描述
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

4.1.21. io.k8s.api.core.v1.CinderPersistentVolumeSource schema

描述

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

模式

属性	类型	描述
fsType	string	要挂载的 fstype Filesystem 类型。必须是主机操作系统支持的文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	secretReference	secretRef 是可选的：指向包含用于连接 OpenStack 的参数的 secret 对象。
volumelD	string	用于识别 cinder 中的卷的卷 ID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

4.1.22. io.k8s.api.core.v1.CinderVolumeSource schema

描述

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	LocalObjectReference	secretRef 是可选的：指向包含用于连接 OpenStack 的参数的 secret 对象。
volumelD	string	用于识别 cinder 中的卷的卷 ID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

4.1.23. io.k8s.api.core.v1.ClaimSource schema**描述**

ClaimSource 描述了对 ResourceClaim 的引用。

应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

类型

对象

模式

属性	类型	描述
resourceClaimName	string	ResourceClaimName 是与这个 pod 相同的命名空间中的 ResourceClaim 对象的名称。
resourceClaimTemplateName	string	<p>ResourceClaimTemplateName 是与这个 pod 相同的命名空间中的 ResourceClaimTemplate 对象的名称。</p> <p>模板将用于创建新的 ResourceClaim，它将绑定到此 pod。删除此 pod 后，ResourceClaim 也会被删除。ResourceClaim 的名称将是 <pod name>-<resource name>，其中 <resource name> 是 PodResourceClaim.Name。如果串联名称对 ResourceClaim 无效（如太长，则 Pod 验证将拒绝 pod）。</p> <p>没有为 pod 拥有该名称的现有 ResourceClaim 不会被用于 pod，以避免错误地使用不相关的资源。然后，调度和 pod 启动会被阻断，直到删除不相关的 ResourceClaim。</p> <p>此字段不可变，在创建 ResourceClaim 后，control plane 不会对对应的 ResourceClaim 进行更改。</p>

4.1.24. io.k8s.api.core.v1.ComponentStatusList schema

描述

组件的所有条件的状态作为 ComponentStatus 对象列表。弃用：此 API 在 v1.19+ 中弃用

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ComponentStatus)	组件状态对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.25. io.k8s.api.core.v1.ConfigMapEnvSource schema

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

对象

模式

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

4.1.26. io.k8s.api.core.v1.ConfigMapKeySelector schema

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

模式

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

4.1.27. io.k8s.api.core.v1.ConfigMapList schema

描述

ConfigMapList 是一个包含 ConfigMap 对象列表的资源。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ConfigMap)	items 是 ConfigMap 的列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.28. io.k8s.api.core.v1.ConfigMapProjection schema

描述

将 ConfigMap 调整为投射卷。

目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。

类型

对象

模式

属性	类型	描述
items	数组(KeyToPath)	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

4.1.29. io.k8s.api.core.v1.ConfigMapVolumeSource schema

描述

将 ConfigMap 调整为卷。

目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
defaultMode	整数	defaultMode 是可选的：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组(KeyToPath)	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

4.1.30. io.k8s.api.core.v1.Container schema

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- name

模式

属性	类型	描述
args	数组 (字符串)	入口点的参数。如果没有提供, 则使用容器镜像的 CMD。变量引用 \$(VAR_NAME) 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 \$, 它允许转义 \$(VAR_NAME) 语法: 例如: "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组 (字符串)	ENTRYPOINT 数组, 不会在 shell 中执行。如果没有提供, 则使用容器镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME) 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 \$, 它允许转义 \$(VAR_NAME) 语法: 例如: "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	数组(EnvVar)	要在容器中设置的环境变量列表。无法更新。
envFrom	数组(EnvFromSource)	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时, 所有无效的密钥都会被报告为事件。当多个源中存在键时, 与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

属性	类型	描述
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
生命周期	生命周期	管理系统应采取的操作来响应容器生命周期事件。无法更新。
livenessProbe	probe	容器存活度的定期探测。如果探测失败，将重启容器。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。

属性	类型	描述
ports	数组(ContainerPort)	要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
readinessProbe	probe	容器服务的定期探测就绪。如果探测失败，容器将从服务端点中删除。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
resizePolicy	数组(ContainerResizePolicy)	容器的资源重新定义策略大小。
resources	ResourceRequirements	此容器所需的计算资源。无法更新。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
securityContext	SecurityContext	securityContext 定义容器应使用运行的安全选项。如果设置，SecurityContext 的字段会覆盖 PodSecurityContext 的等效字段。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
startupProbe	probe	startupProbe 表示 Pod 已成功初始化。如果指定，则不会执行其他探测，直到成功完成为止。如果这个探测失败，则 Pod 将重启，就像 livenessProbe 失败一样。这可用于在 Pod 生命周期开始时提供不同的探测参数，当可能需要很长时间才能加载数据或温缓存，而不是在 steady-state 操作期间进行。这不能被更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

属性	类型	描述
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。

属性	类型	描述
volumeDevices	数组(VolumeDevice)	volumeDevices 是容器使用的块设备列表。
volumeMounts	数组(VolumeMount)	要挂载到容器文件系统 Pod 卷。无法更新。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

4.1.31. io.k8s.api.core.v1.ContainerPort schema

描述

containerPort 代表单一容器中的网络端口。

类型

object

必填

- containerPort

模式

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。

属性	类型	描述
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

4.1.32. io.k8s.api.core.v1.ContainerResizePolicy schema

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

模式

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

4.1.33. io.k8s.api.core.v1.CSIPersistentVolumeSource schema

描述

代表由外部 CSI 卷驱动程序(Beta 功能)管理的存储

类型

object

必填

- **driver**
- **volumeHandle**

模式

属性	类型	描述
controllerExpandSecretRef	secretReference	controllerExpandSecretRef 是对包含要传递给 CSI 驱动程序的敏感信息的 secret 对象的引用，以完成 CSI ControllerExpandVolume 调用。此字段是可选的，如果不需要 secret，则可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会被传递。
controllerPublishSecretRef	secretReference	controllerPublishSecretRef 是对包含要传递给 CSI 驱动程序的 secret 对象的引用，以完成 CSI ControllerPublishVolume 和 ControllerUnpublishVolume 调用。此字段是可选的，如果不需要 secret，则可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会被传递。
driver	string	driver 是要用于此卷的驱动程序名称。必需。
fsType	string	要挂载的 fstype。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".
nodeExpandSecretRef	secretReference	nodeExpandSecretRef 是对包含要传递给 CSI 驱动程序的 secret 对象的引用，以完成 CSI NodeExpandVolume 调用。这是一个 beta 字段，由 CSINodeExpandSecret 功能门启用。如果不需要 secret，则可以省略此字段。如果 secret 对象包含多个 secret，则所有 secret 都会被传递。
nodePublishSecretRef	secretReference	nodePublishSecretRef 是对包含要传递给 CSI 驱动程序的 secret 对象的引用，以完成 CSI NodePublishVolume 和 NodeUnpublishVolume 调用。此字段是可选的，如果不需要 secret，则可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会被传递。

属性	类型	描述
nodeStageSecretRef	secretReference	nodeStageSecretRef 是包含传递给 CSI 驱动程序的 secret 对象的引用，以完成 CSI NodeStageVolume 和 NodeUnstageVolume 调用。此字段是可选的，如果不需要 secret，则可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会被传递。
readOnly	布尔值	传递给 ControllerPublishVolumeRequest 的 readOnly 值。默认为 false（读/写）。
volumeAttributes	对象（字符串）	要发布的卷的 volumeAttributes。
volumeHandle	string	volumeHandle 是 CSI 卷插件的 CreateVolume 返回的唯一卷名称，以在所有后续调用中引用卷。必需。

4.1.34. io.k8s.api.core.v1.CSIVolumeSource schema

描述

代表由外部 CSI 驱动程序管理的卷的源位置

类型

object

必填

- **driver**

模式

属性	类型	描述
driver	string	driver 是处理这个卷的 CSI 驱动程序的名称。请参考您的管理员获取集群中注册的正确名称。
fsType	string	要挂载的 fstype。Ex. "ext4", "xfs", "ntfs".如果没有提供，空值会被传递给关联的 CSI 驱动程序，该驱动程序将决定要应用的默认文件系统。

属性	类型	描述
nodePublishSecretRef	LocalObjectReference	nodePublishSecretRef 是对包含要传递给 CSI 驱动程序的 secret 对象的引用，以完成 CSI NodePublishVolume 和 NodeUnpublishVolume 调用。此字段是可选的，如果不需要 secret，则可能为空。如果 secret 对象包含多个 secret，则所有 secret 引用都会传递。
readOnly	布尔值	readonly 指定卷的只读配置。默认为 false（读/写）。
volumeAttributes	对象（字符串）	volumeAttributes 存储传递给 CSI 驱动程序的特定于驱动程序的属性。如需支持的值，请参考您的驱动程序文档。

4.1.35. io.k8s.api.core.v1.DownwardAPIProjection schema

描述

代表下下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。

类型

对象

模式

属性	类型	描述
items	数组 (DownwardAPIVolumeFile)	items 是 DownwardAPIVolume 文件列表

4.1.36. io.k8s.api.core.v1.DownwardAPIVolumeFile schema

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- path

模式

属性	类型	描述
fieldRef	ObjectFieldSelector	必需：选择 pod 的字段：只支持注解、标签、名称和命名空间。
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	ResourceFieldSelector	选择容器的资源：当前仅支持资源限值和请求(limits.cpu、limit.memory、request.cpu 和 requests.memory)。

4.1.37. io.k8s.api.core.v1.DownwardAPIVolumeSource schema

描述

DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
defaultMode	整数	可选：默认在创建的文件中使用的模式位。必须是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。

属性	类型	描述
items	数组 (DownwardAPIVolumeFile)	items 是 Downward API 卷文件列表

4.1.38. io.k8s.api.core.v1.EmptyDirVolumeSource schema

描述

表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
中	string	Medium 代表这个目录应该支持哪些存储类型。默认值为 "", 它需要使用节点的默认介质。必须是空字符串（默认）或 Memory。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit	数量	sizeLimit 是此 EmptyDir 卷所需的本地存储总量。大小限制也适用于内存介质。内存用量 EmptyDir 的最大用法是此处指定的 SizeLimit 和 pod 中所有容器的内存限值总和。默认值为 nil, 表示限制未定义。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir

4.1.39. io.k8s.api.core.v1.EndpointsList schema

描述

EndpointsList 是端点列表。

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Endpoints)	端点列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.40. io.k8s.api.core.v1.EnvFromSource schema

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

对象

模式

属性	类型	描述
configMapRef	ConfigMapEnvSource	要从中选择的 ConfigMap
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	SecretEnvSource	要从中选择的 Secret

4.1.41. io.k8s.api.core.v1.EnvVar schema

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

模式

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	EnvVarSource	环境变量的值的源。如果 value 不是空，则无法使用。

4.1.42. io.k8s.api.core.v1.EnvVarSource schema

描述

EnvVarSource 代表 EnvVar 的值的源。

类型

对象

模式

属性	类型	描述
configMapKeyRef	ConfigMapKeySelector	选择 ConfigMap 的密钥。

属性	类型	描述
fieldRef	ObjectFieldSelector	选择 pod 的字段：支持 metadata.name, metadata.namespace, metadata.labels['<KEY>'] , metadata.annotations['<KEY>'] , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
resourceFieldRef	ResourceFieldSelector	选择容器的资源：当前仅支持资源限值和请求(limits.cpu、limits.memory、limits.ephemeral-storage、request.cpu、request.memory 和 requests.ephemeral-storage)。
secretKeyRef	SecretKeySelector	在 pod 命名空间中选择 secret 的密钥

4.1.43. io.k8s.api.core.v1.EphemeralContainer schema

描述

EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。

要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。

类型

object

必填

- **name**

模式

属性	类型	描述
----	----	----

属性	类型	描述
args	数组 (字符串)	入口点的参数。如果没有提供, 则使用镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组 (字符串)	ENTRYPOINT 数组, 不会在 shell 中执行。如果没有提供, 则使用镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	数组(EnvVar)	要在容器中设置的环境变量列表。无法更新。
envFrom	数组(EnvFromSource)	用于在容器中填充环境变量的源列表。源中定义的键必须是 <code>C_IDENTIFIER</code> 。当容器启动时, 所有无效的密钥都会被报告为事件。当多个源中存在键时, 与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
image	string	容器镜像名称。更多信息: https://kubernetes.io/docs/concepts/containers/images

属性	类型	描述
imagePullPolicy	字符串	<p>镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签, 则默认为 Always, 否则为 IfNotPresent。无法更新。更多信息 :</p> <p>https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>可能枚举值 : - "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败, 则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在, 且拉取失败, 则容器将失败。- "Never" 表示 kubelet 不会拉取镜像, 但只使用本地镜像。如果镜像不存在, 容器将失败</p>
生命周期	生命周期	临时容器不允许使用生命周期。
livenessProbe	probe	临时容器不允许探测。
name	字符串	指定为 DNS_LABEL 的临时容器的名称。此名称在所有容器、init 容器和临时容器中必须是唯一的。
ports	数组(ContainerPort)	临时容器不允许使用端口。
readinessProbe	probe	临时容器不允许探测。
resizePolicy	数组(ContainerResizePolicy)	容器的资源重新定义策略大小。
resources	ResourceRequirements	临时容器不允许使用资源。临时容器使用已分配给 pod 的备用资源。
securityContext	SecurityContext	可选 : SecurityContext 定义临时容器应使用的安全选项。如果设置, SecurityContext 的字段会覆盖 PodSecurityContext 的等效字段。
startupProbe	probe	临时容器不允许探测。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置, 则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。

属性	类型	描述
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
targetContainerName	字符串	<p>如果设置，则此临时容器目标的 PodSpec 中的容器名称。临时容器将在此容器的命名空间中运行 (IPC、PID 等)。如果没有设置，则临时容器将使用 Pod spec 中配置的命名空间。</p> <p>容器运行时必须实施对此功能的支持。如果运行时不支持以命名空间为目标，则设置此字段的结果未定义。</p>
terminationMessagePath	字符串	<p>可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。</p>

属性	类型	描述
terminationMessagePolicy	字符串	<p>指明如何填充终止消息。文件将使用 <code>terminationMessagePath</code> 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 <code>FallbackToLogsOnError</code> 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 <code>file</code>。无法更新。</p> <p>可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 <code>terminationMessagePath</code> 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 <code>MessagePath</code> 的内容。</p>
tty	布尔值	此容器是否应为自己分配 TTY，还需要 <code>'stdin'</code> 为 <code>true</code> 。默认值为 <code>false</code> 。
volumeDevices	数组(VolumeDevice)	<code>volumeDevices</code> 是容器使用的块设备列表。
volumeMounts	数组(VolumeMount)	要挂载到容器文件系统 Pod 卷。临时容器不允许进行 <code>subPath</code> 挂载。无法更新。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

4.1.44. io.k8s.api.core.v1.EphemeralVolumeSource schema

描述

代表由普通存储驱动程序处理的临时卷。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
volumeClaimTemplate	PersistentVolumeClaimTemplate	<p>将用于创建独立 PVC 来置备卷。嵌入了此 EphemeralVolumeSource 的 pod 将是 PVC 的所有者，即 PVC 将与 pod 一起删除。PVC 的名称将是 <pod name>-<volume name>，其中 <volume name> 是 PodSpec.Volumes 数组条目中的名称。如果串联的名称对 PVC 无效，则 Pod 验证将拒绝 pod（例如，太长）。</p> <p>没有为 pod 拥有该名称的现有 PVC 不会被 用于 pod，以避免错误地使用不相关的卷。然后，启动 pod 会被阻断，直到删除不相关的 PVC。如果 pod 使用预先创建的 PVC，则在 pod 存在后，PVC 必须使用所有者引用 pod 更新。通常，这不应该是必需的，但在手动重新创建损坏的集群时可能很有用。</p> <p>此字段是只读的，在创建后 Kubernetes 不会对 PVC 进行任何更改。</p> <p>必需，不能是 nil。</p>

4.1.45. io.k8s.api.core.v1.EventList schema

描述

EventList 是事件列表。

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组（事件）	事件列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.46. io.k8s.api.core.v1.EventSource schema

描述

EventSource 包含事件的信息。

类型

对象

模式

属性	类型	描述
component	字符串	生成事件的组件。
主机	字符串	生成事件的节点名称。

4.1.47. io.k8s.api.core.v1.ExecAction schema

描述

ExecAction 描述了 "run in container" 操作。

类型

对象

模式

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

4.1.48. io.k8s.api.core.v1.FCVolumeSource schema

描述

代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。
LUN	整数	LUN 是可选的：FC 目标 lun 号
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组（字符串）	targetWWN 可选：FC 目标全球名称(WWN)
wwids	数组（字符串）	wwids 可选：FC 卷全局广泛的标识符(wwids) Either wwids 或 targetWWNs 和 lun 的组合必须同时设置，但不能同时设置。

4.1.49. io.k8s.api.core.v1.FlexPersistentVolumeSource schema

描述

FlexPersistentVolumeSource 代表使用基于 exec 的插件置备/附加的通用持久性卷资源。

类型

object

必填

- **driver**

模式

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认文件系统取决于 FlexVolume 脚本。
options	对象 (字符串)	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	secretReference	secretRef 是可选的：SecretRef 是对包含要传递给插件脚本的敏感信息的 secret 对象的引用。如果没有指定 secret 对象，则这可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会传递给插件脚本。

4.1.50. io.k8s.api.core.v1.FlexVolumeSource schema**描述**

FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。

类型

object

必填

- **driver**

模式

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认文件系统取决于 FlexVolume 脚本。
options	对象 (字符串)	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	LocalObjectReference	secretRef 是可选的：secretRef 是对包含要传递给插件脚本的敏感信息的 secret 对象的引用。如果没有指定 secret 对象，则这可能为空。如果 secret 对象包含多个 secret，则所有 secret 都会传递给插件脚本。

4.1.51. io.k8s.api.core.v1.FlockerVolumeSource schema

描述

代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
datasetName	string	datasetName 是存储在 Flocker 的 dataset 的 dataset 的 dataset 的 Name。
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

4.1.52. io.k8s.api.core.v1.GCEPersistentDiskVolumeSource schema

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **pdName**

模式

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4"，"xfs"，"ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为"1"。同样，/dev/sda 的卷分区是"0"（或者您可以将属性留空）。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	string	pdName 是 GCE 中的 PD 资源的唯一名称。用于识别 GCE 中的磁盘。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

4.1.53. io.k8s.api.core.v1.GitRepoVolumeSource schema

描述

代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。

DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。

类型

object

必填

- 软件仓库

模式

属性	类型	描述
目录	string	directory 是目标目录名称。不得包含或以 '!' 开头。如果提供了 '!'，则卷目录将是 git 存储库。否则，如果指定，卷将包含具有指定名称的子目录中的 git 存储库。
软件仓库	string	repository 是 URL
revision	string	revision 是指定修订版本的提交哈希。

4.1.54. io.k8s.api.core.v1.GlusterfsPersistentVolumeSource schema

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

模式

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

属性	类型	描述
endpointsNamespace	string	endpointsNamespace 是包含 Glusterfs 端点的命名空间。如果此字段为空，则 EndpointNamespace 默认为与绑定 PVC 相同的命名空间。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

4.1.55. io.k8s.api.core.v1.GlusterfsVolumeSource schema

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

模式

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

属性	类型	描述
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

4.1.56. io.k8s.api.core.v1.GRPCAction schema

描述

类型

object

必填

- **port**

模式

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

4.1.57. io.k8s.api.core.v1.HostAlias schema

描述

HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。

类型

对象

模式

属性	类型	描述
主机名	数组（字符串）	以上 IP 地址的主机名。
ip	字符串	主机文件条目的 IP 地址。

4.1.58. io.k8s.api.core.v1.HostPathVolumeSource schema

描述

代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- path

模式

属性	类型	描述
path	string	主机上目录的路径。如果路径是符号链接，它将遵循到实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath
type	string	HostPath Volume Defaults 为 "" more info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath 可能枚举值：- "" 对于向后兼容，如果未设置 - "BlockDevice" A 块设备必须存在于给定路径 - "CharDevice" A 字符设备必须存在于给定路径上 - "Directory" A 目录必须存在于给定路径 - "DirectoryOrCreate" 中（如果给定路径不存在），将根据需要创建一个空目录，且文件模式为 0755，其具有与 Kubelet 相同的组和所有权。- "File" A 文件必须存在于给定路径上 - "FileOrCreate" 如果给定路径上不存在，将根据需要创建一个空文件，该文件与文件模式 0644 具有相同的组和所有权。- "Socket" A UNIX 套接字必须存在于给定路径上。

4.1.59. io.k8s.api.core.v1.HTTPGetAction schema

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

模式

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	数组(HTTPHeader)	在请求中设置的自定义标头。HTTP 允许重复标头。
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

4.1.60. io.k8s.api.core.v1.HTTPHeader schema

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

模式

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

4.1.61. io.k8s.api.core.v1.ISCSIPersistentVolumeSource schema

描述

ISCSIPersistentVolumeSource 代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

模式

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是 iSCSI 限定名称的目标。

属性	类型	描述
iscsilInterface	string	iscsilInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 是 iSCSI 目标 Lun 编号。
门户	数组 (字符串)	门户是 iSCSI 目标门户列表。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	secretReference	secretRef 是 iSCSI 目标和启动器验证的 CHAP Secret
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。

4.1.62. io.k8s.api.core.v1.ISCSIVolumeSource schema

描述

代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

模式

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是目标 iSCSI 限定名称。
iscsiInterface	string	iscsiInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 代表 iSCSI 目标 Lun 编号。
门户	数组（字符串）	门户是 iSCSI 目标门户列表。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	LocalObjectReference	secretRef 是 iSCSI 目标和启动器验证的 CHAP Secret
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。

4.1.63. io.k8s.api.core.v1.KeyToPath schema

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**

- path

模式

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

4.1.64. io.k8s.api.core.v1.Lifecycle schema

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

对象

模式

属性	类型	描述
postStart	LifecycleHandler	在创建容器后，postStart 会立即调用。如果处理程序失败，容器会根据重启策略终止并重启。对容器块的其他管理，直到 hook 完成为止。更多信息： https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

属性	类型	描述
preStop	LifecycleHandler	由于 API 请求或管理事件（如存活度/启动探测失败、抢占、资源争用等）在容器终止前立即调用 preStop。如果容器崩溃或退出，则不会调用处理程序。Pod 的终止宽限期倒计时在执行 PreStop hook 前开始。无论处理程序的结果如何，容器最终都会在 Pod 终止宽限期内终止（除非终结器没有延迟）。对容器进行其他管理，直到 hook 完成或到达终止宽限期为止。更多信息： https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

4.1.65. io.k8s.api.core.v1.LifecycleHandler schema

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCP Socket。

类型

对象

模式

属性	类型	描述
exec	ExecAction	exec 指定要执行的操作。
httpGet	HTTPGetAction	httpGet 指定要执行的 http 请求。
tcpSocket	TCP Socket Action	已弃用。tcpSocket 不支持作为 LifecycleHandler，并保持向后兼容。当指定了 tcp 处理程序时，没有此字段验证，生命周期 hook 会在运行时失败。

4.1.66. io.k8s.api.core.v1.LimitRangeList schema

描述

LimitRangeList 是 LimitRange 项目的列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(LimitRange)	items 是 LimitRange 对象列表。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法(CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.67. io.k8s.api.core.v1.LocalObjectReference schema

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

4.1.68. io.k8s.api.core.v1.LocalVolumeSource schema

描述

local 代表与节点关联性直接附加的存储(Beta 功能)

类型

object

必填

- path

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。只有在路径是块设备时，它才会应用。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".如果未指定，则默认值为自动选择文件系统。
path	string	到节点上卷的完整路径的路径。它可以是目录或块设备(disk, partition, ...)。

4.1.69. io.k8s.api.core.v1.NamespaceList schema

描述

NamespaceList 是命名空间列表。

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Namespace)	items 是列表中的 Namespace 对象列表。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.70. io.k8s.api.core.v1.NFSVolumeSource schema

描述

代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **server**
- **path**

模式

属性	类型	描述
path	string	NFS 服务器导出的路径。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	布尔值	此处的只读将强制使用只读权限挂载 NFS 导出。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server 是 NFS 服务器的主机名或 IP 地址。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs

4.1.71. io.k8s.api.core.v1.NodeAffinity schema

描述

节点关联性是一组节点关联性调度规则。

类型

对象

模式

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	数组 (PreferredSchedulingTerm)	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。
requiredDuringSchedulingIgnoredDuringExecution	NodeSelector	如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 Pod 执行期间由此字段指定的关联性要求满足（例如，由于更新），系统可能无法尝试最终从其节点中驱除 pod。

4.1.72. io.k8s.api.core.v1.NodeList schema

描述

NodeList 是所有使用 master 注册的节点的完整列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Node)	节点列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.73. io.k8s.api.core.v1.NodeSelector schema

描述

节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

类型

object

必填

- `nodeSelectorTerms`

模式

属性	类型	描述
<code>nodeSelectorTerms</code>	数组(NodeSelectorTerm)	必需。节点选择器术语列表。条款是 ORed。

4.1.74. `io.k8s.api.core.v1.NodeSelectorRequirement` schema

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- `key`
- `operator`

模式

属性	类型	描述
<code>key</code>	字符串	选择器应用到的标签键。
<code>operator</code>	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
<code>values</code>	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

4.1.75. `io.k8s.api.core.v1.NodeSelectorTerm` schema

描述

`null` 或 `empty` 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

对象

模式

属性	类型	描述
matchExpressions	数组 (NodeSelectorRequirement)	节点标签(label)的节点选择器要求列表。
matchFields	数组 (NodeSelectorRequirement)	节点字段的节点选择器要求列表。

4.1.76. io.k8s.api.core.v1.ObjectFieldSelector schema

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

模式

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

4.1.77. io.k8s.api.core.v1.ObjectReference schema

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

对象

模式

属性	类型	描述
apiVersion	字符串	引用的 API 版本。

属性	类型	描述
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 <code>resourceVersion</code> （如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

4.1.78. io.k8s.api.core.v1.PersistentVolumeClaim schema

描述

`PersistentVolumeClaim` 是用户对持久性卷的请求并声明

类型

对象

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源
status	对象	PersistentVolumeClaimStatus 是持久性卷声明的当前状态。

..spec Description::

+

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含卷应具有的所有访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	对象	TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。
dataSourceRef	object	dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。
resources	对象	ResourceRequirements 描述计算资源要求。

属性	类型	描述
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统进行或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

..spec.dataSource Description::

+

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

..spec.dataSourceRef Description::

+

dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称
namespace	string	namespace 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 gateway.networking.k8s.io/ReferenceGrant 对象，以允许该命名空间的所有者接受引用。详情请查看 ReferenceGrant 文档。(alpha)此字段需要启用 CrossNamespaceVolumeDataSource 功能门。

..spec.resources Description::

+

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

..spec.resources.claims Description::

+

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。

这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

..spec.resources.claims[] Description::

+

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

..status Description::

+

PersistentVolumeClaimStatus 是持久性卷声明的当前状态。

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含支持 PVC 的卷的实际访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
allocatedResources	对象（数量）	allocatedResources 是 AllocatedResources 中的存储资源，跟踪分配给 PVC 的容量。当请求卷扩展操作时，它可能大于实际容量。对于存储配额，使用 allocatedResources 和 PVC.spec.resources 的较大的值。如果没有设置 allocatedResources，则只用于配额计算的 PVC.spec.resources。如果卷扩展容量请求较低，则仅当没有扩展操作且实际卷容量等于或小于请求的容量时，才会降低 allocatedResources。这是一个 alpha 字段，需要启用 RecoverVolumeExpansionFailure 功能。

属性	类型	描述
capacity	对象 (数量)	capacity 代表底层卷的实际资源。
conditions	数组	conditions 是持久性卷声明的当前条件。如果底层持久性卷大小调整，则 Condition 将设置为 'ResizeStarted'。
conditions[]	object	PersistentVolumeClaimCondition 包含有关 pvc 状态的详细信息
phase	string	<p>phase 代表 PersistentVolumeClaim 的当前阶段。</p> <p>可能枚举值： - "Bound" 用于绑定的 PersistentVolumeClaims - "Lost" 用于丢失其底层 PersistentVolume 的 PersistentVolumeClaim。这个声明被绑定到一个 PersistentVolume，这个卷不再存在，且其上的所有数据都丢失。 - "Pending" 用于尚未绑定的 PersistentVolumeClaims</p>

属性	类型	描述
resizeStatus	string	<p>resizeStatus 存储调整大小操作的状态。默认情况下，不会设置 ResizeStatus，但当扩展为 complete resizeStatus 时，通过调整大小 controller 或 kubelet 被设为空字符串。这是一个 alpha 字段，需要启用 RecoverVolumeExpansionFailure 功能。</p> <p>可能枚举值： - "" when expansion is complete, 通过调整大小 controller 或 kubelet 设置空字符串。 - 当扩展失败时，会在扩展失败时使用终端错误设置</p> <p>"ControllerExpansionFailed" State set. 临时错误（如 timeout）不应设置这个状态，且应该修改 ResizeStatus，因为调整大小的控制器可以恢复卷扩展。 - 在 control-plane 中扩展卷时，</p> <p>"ControllerExpansionInProgress" State set failed with an terminal error. 临时错误不会设置 NodeExpansionFailed. - 当 kubelet 启动扩展卷时，</p> <p>"NodeExpansionInProgress" State set when</p> <p>"NodeExpansionPending" State set when resize controller has completed the volume but on the node is required on the node.</p>

..status.conditions Description::

+

conditions 是持久性卷声明的当前条件。如果底层持久性卷大小调整，则 Condition 将设置为 'ResizeStarted'。

类型

数组

..status.conditions[] Description::

+

PersistentVolumeClaimCondition 包含有关 pvc 状态的详细信息

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastProbeTime	时间	lastProbeTime 是我们探测到条件的 时间。
lastTransitionTime	时间	lastTransitionTime 是条件从一个 状态转换到另一个状态的时间。
message	string	Message 是人类可读的消息，指示 最后一次转换的详细信息。
reason	string	原因是唯一的，这应该是简短的、 机器易理解的字符串，它给出了状 况最后一次转换的原因。如果报告 了"ResizeStarted"，这表示基础持 久卷正在调整大小。
status	字符串	
type	字符串	

4.1.79. io.k8s.api.core.v1.PersistentVolumeClaimList schema

描述

PersistentVolumeClaimList 是 PersistentVolumeClaim 项列表。

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(PersistentVolumeClaim)	items 是持久性卷声明列表。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.80. io.k8s.api.core.v1.PersistentVolumeClaimSpec schema

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

对象

模式

属性	类型	描述
accessModes	数组 (字符串)	accessModes 包含卷应具有的所需访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

属性	类型	描述
dataSource	TypedLocalObjectReference	DataSource 字段可以用来指定： * 一个现有的 VolumeSnapshot 对象 (snapshot.storage.k8s.io/VolumeSnapshot)* 一个现有的 PVC (PersistentVolumeClaim)，如果置备程序或外部控制器可以支持指定的数据源，它将根据指定数据源的内容创建一个新卷。当启用 AnyVolumeDataSource 功能门时，DataSource 内容将复制到 dataSourceRef，当没有指定 dataSourceRef.namespace 时，dataSourceRef 内容将复制到 dataSource。如果指定了命名空间，则 dataSourceRef 将不会复制到 dataSource 中。

属性	类型	描述
dataSourceRef	TypedObjectReference	<p>dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。</p>
resources	ResourceRequirements	<p>resources 代表卷应具有的最少资源。如果启用了 RecoverVolumeExpansionFailure 功能，则允许用户指定低于之前值的资源要求，但必须大于声明的 status 字段中记录的容量。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources</p>

属性	类型	描述
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

4.1.81. io.k8s.api.core.v1.PersistentVolumeClaimTemplate schema

描述

PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

类型

object

必填

- **spec**

模式

属性	类型	描述
metadata	ObjectMeta	可以包含在创建时复制到 PVC 中的标签和注解。不允许其他字段，并在验证过程中被拒绝。
spec	PersistentVolumeClaimSpec	PersistentVolumeClaim 的规格。整个内容都会不动地复制到从该模板创建的 PVC 中。PersistentVolumeClaim 中的相同字段也有效。

4.1.82. io.k8s.api.core.v1.PersistentVolumeClaimVolumeSource schema

描述

PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。

类型

object

必填

- **claimName**

模式

属性	类型	描述
claimName	string	claimName 是使用此卷与 pod 相同的命名空间中的 PersistentVolumeClaim 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	布尔值	readonly 将强制 VolumeMount 中的 ReadOnly 设置。默认 false。

4.1.83. io.k8s.api.core.v1.PersistentVolumeList schema

描述

PersistentVolumeList 是 PersistentVolume 项列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
items	数组(PersistentVolume)	items 是持久性卷列表。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.84. io.k8s.api.core.v1.PersistentVolumeSpec schema

描述

PersistentVolumeSpec 是持久性卷的规格。

类型

对象

模式

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含可挂载卷的所有方法。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes
awsElasticBlockStore	AWSElasticBlockStoreVolumeSource	awsElasticBlockStore 代表附加到 kubelet 主机机器的 AWS Disk 资源，然后公开给 pod。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
azureDisk	AzureDiskVolumeSource	azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

属性	类型	描述
azureFile	AzureFilePersistentVolumeSource	azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。
capacity	对象（数量）	capacity 是持久性卷资源和容量的描述。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity
cephfs	CephFSPersistentVolumeSource	CephFS 代表共享 pod 生命周期的主机上的 Ceph FS 挂载
cinder	CinderPersistentVolumeSource	Cinder 代表附加并挂载到 kubelet 主机上的 cinder 卷。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
claimRef	ObjectReference	claimRef 是 PersistentVolume 和 PersistentVolumeClaim 之间的双向绑定的一部分。当绑定时，预期为非 nil。claim.VolumeName 是 PV 和 PVC 之间的权威绑定。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#binding
csi	CSIPersistentVolumeSource	CSI 代表由外部 CSI 驱动程序 (Beta 功能)处理的存储。
fc	FCVolumeSource	FC 代表附加到 kubelet 主机的 Fibre Channel 资源，然后公开给 pod。
flexVolume	FlexPersistentVolumeSource	FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。
flocker	FlockerVolumeSource	flocker 代表附加到 kubelet 主机机器的 Flocker 卷，并公开给 pod 以供其使用。这取决于运行 Flocker 控制服务

属性	类型	描述
gcePersistentDisk	GCEPersistentDiskVolumeSource	gcePersistentDisk 代表附加到 kubelet 主机机器的 GCE Disk 资源，然后公开给 pod。由管理员调配。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
glusterfs	GlusterfsPersistentVolumeSource	GlusterFS 代表附加到主机并公开给 pod 的 Glusterfs 卷。由管理员调配。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md
hostPath	HostPathVolumeSource	hostpath 代表主机上的目录。由开发人员或测试程序调配。这仅适用于单节点开发和测试！在多节点集群中，不支持主机存储，在多节点集群中不支持 WILL NOT WORK。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath
iscsi	ISCSIPersistentVolumeSource	iSCSI 代表附加到 kubelet 主机机器的 iSCSI 磁盘资源，然后公开给 pod。由管理员调配。
local	LocalVolumeSource	local 代表与节点关联性直接附加的存储
mountOptions	数组（字符串）	mountOptions 是挂载选项列表，如 ["ro", "soft"]。未验证 - 如果挂载无效，则挂载将失败。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes/#mount-options
nfs	NFSVolumeSource	NFS 代表主机上的 NFS 挂载。由管理员调配。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
nodeAffinity	VolumeNodeAffinity	nodeAffinity 定义限制此卷可从中访问哪些节点。此字段会影响使用这个卷的 pod 的调度。

属性	类型	描述
persistentVolumeReclaimPolicy	string	<p><code>persistentVolumeReclaimPolicy</code> 定义从声明中释放持久性卷时会发生什么。有效选项为 <code>Retain</code>（为手动创建的 <code>PersistentVolume</code> 的默认设置）、<code>Delete</code>（动态置备的 <code>PersistentVolume</code> 的默认）和 <code>Recycle</code>（已弃用）。<code>Recycle</code> 必须被此 <code>PersistentVolume</code> 底层的卷插件支持。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#reclaiming</p> <p>可能枚举值：- "Delete" 表示该卷将从其声明中的 Kubernetes 中删除。卷插件必须支持 <code>Deletion</code>。 - "Recycle" 意味着卷将被重新回收到其声明中未绑定持久性卷的池。卷插件必须支持 <code>Recycling</code>。 - "Retain" 表示该卷将保留在其当前阶段(<code>Released</code>)中，供管理员手动回收。默认策略为 <code>Retain</code>。</p>
photonPersistentDisk	PhotonPersistentDiskVolumeSource	<code>photonPersistentDisk</code> 代表在 kubelets 主机上附加并挂载的 <code>PhotonController</code> 持久磁盘
portworxVolume	PortworxVolumeSource	<code>portworxVolume</code> 代表附加并挂载到 kubelet 主机上的 <code>portworx</code> 卷
quobyte	QuobyteVolumeSource	<code>quobyte</code> 代表共享 pod 生命周期的主机上的 <code>Quobyte</code> 挂载
rbd	RBDPersistentVolumeSource	<code>RBD</code> 代表共享 pod 生命周期的主机上的 <code>Rados</code> 块设备挂载。更多信息： https://examples.k8s.io/volumes/rbd/README.md
scaleIO	ScaleIOPersistentVolumeSource	<code>scaleIO</code> 代表一个在 Kubernetes 节点上附加并挂载的 <code>ScaleIO</code> 持久性卷。
storageClassName	string	<code>storageClassName</code> 是这个持久性卷所属的 <code>StorageClass</code> 的名称。空值意味着这个卷不属于任何 <code>StorageClass</code> 。

属性	类型	描述
storageos	StorageOSPersistentVolumeSource	storageos 代表附加到 kubelet 主机机器的 StorageOS 卷，并挂载到 pod 更多信息： https://examples.k8s.io/volumes/storageos/README.md
volumeMode	字符串	volumeMode 定义卷是否要与格式化的文件系统一起使用，或者保持在原始块状态。当没有包括在 spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统或格式化。
vsphereVolume	VsphereVirtualDiskVolumeSource	vsphereVolume 代表在 kubelets 主机上附加并挂载的 vSphere 卷

4.1.85. io.k8s.api.core.v1.PhotonPersistentDiskVolumeSource schema

描述

代表 Photon Controller 持久磁盘资源。

类型

object

必填

- **pdID**

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。
pdID	string	pdID 是标识 Photon Controller 持久磁盘的 ID

4.1.86. io.k8s.api.core.v1.PodAffinity schema

描述

Pod 关联性是一组 pod 关联性调度规则。

类型

对象

模式

属性	类型	描述
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	数组 (WeightedPodAffinityTerm)	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、 <code>requiredDuringSchedulingIgnoredDuringExecution</code> 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中，如果节点具有与相应 <code>podAffinityTerm</code> 匹配的 pod；具有最高总和最高的节点是首选的节点。
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	数组(PodAffinityTerm)	如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 <code>podAffinityTerm</code> 对应的节点列表会被干扰，例如，所有术语都必须满足。

4.1.87. io.k8s.api.core.v1.PodAffinityTerm schema

描述

定义一组 pod（与给定命名空间匹配的 `labelSelector` 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 `co-located` 定义在带有键 `<topologyKey>` 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- `topologyKey`

模式

属性	类型	描述
<code>labelSelector</code>	labelSelector	一个对一组资源的标签查询，本例中为 pod。

属性	类型	描述
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

4.1.88. io.k8s.api.core.v1.PodAntiAffinity schema

描述

Pod 反关联性是一组 pod 反关联性调度规则。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	数组 (WeightedPodAffinityTerm)	调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringSchedulingIgnoredDuringExecution 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 podAffinityTerm 匹配的 pod；如果节点具有与相应 podAffinityTerm 匹配的 pod，则该节点是首选的。
requiredDuringSchedulingIgnoredDuringExecution	数组(PodAffinityTerm)	如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

4.1.89. io.k8s.api.core.v1.PodDNSConfig schema

描述

PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

类型

对象

模式

属性	类型	描述
nameservers	数组（字符串）	DNS 名称服务器 IP 地址列表。这将附加到从 DNSPolicy 生成的基本名称服务器。将删除重复的名称服务器。
options	数组(PodDNSConfigOption)	DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。

属性	类型	描述
搜索	数组（字符串）	host-name 查找的 DNS 搜索域列表。这将附加到 DNSPolicy 生成的基本搜索路径中。将删除重复的搜索路径。

4.1.90. io.k8s.api.core.v1.PodDNSConfigOption schema

描述

PodDNSConfigOption 定义 pod 的 DNS 解析器选项。

类型

对象

模式

属性	类型	描述
name	字符串	必需。
value	string	

4.1.91. io.k8s.api.core.v1.PodList schema

描述

PodList 是一个 Pod 列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
items	数组(Pod)	pod 列表。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.92. io.k8s.api.core.v1.PodOS schema

描述

PodOS 定义 pod 的 OS 参数。

类型

object

必填

- **name**

模式

属性	类型	描述
name	string	name 是操作系统的名称。目前支持的值有 linux 和 windows。额外值可能会在以后定义，可以是： https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration 客户端应该应该处理附加值，并将此字段中的未识别的值视为 os: null

4.1.93. io.k8s.api.core.v1.PodReadinessGate schema

描述

PodReadinessGate 包含对 pod 条件的引用

类型

object

必填

- **conditionType**

模式

属性	类型	描述
conditionType	字符串	conditionType 指的是 pod 条件列表中具有匹配类型的条件。

4.1.94. io.k8s.api.core.v1.PodResourceClaim schema

描述

PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

类型

object

必填

- **name**

模式

属性	类型	描述
name	string	在 pod 中唯一标识此资源声明的名称。这必须是 DNS_LABEL。
source	ClaimSource	source 描述在什么位置查找 ResourceClaim。

4.1.95. io.k8s.api.core.v1.PodSchedulingGate schema

描述

PodSchedulingGate 与 Pod 关联，以保护其调度。

类型

object

必填

- **name**

模式

属性	类型	描述
name	string	调度授权的名称。每个调度最低要求都必须具有唯一的 name 字段。

4.1.96. io.k8s.api.core.v1.PodSecurityContext schema

描述

PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。

类型

对象

模式

属性	类型	描述
fsGroup	整数	<p>适用于 pod 中所有容器的特殊补充组。有些卷类型允许 Kubelet 将该卷的所有权更改为 pod 拥有：</p> <ol style="list-style-type: none"> 1.所属 GID 将是 FSGroup 2. set setgid 位（在卷中创建的新文件将归 FSGroup 所有） 3. 权限位为 OR'd with rw-rw---- <p>如果未设置，Kubelet 将不修改任何卷的所有权和权限。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
fsGroupChangePolicy	字符串	<p>fsGroupChangePolicy 定义在 Pod 中公开卷之前更改卷的所有权和权限的行为。此字段仅适用于支持 fsGroup 基于所有权（和权限）的卷类型。它对临时卷类型没有影响，如：secret、configmaps 和 emptydir。有效值为 "OnRootMismatch" 和 "Always"。如果没有指定，则使用 "Always"。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值： - "Always" 表示当卷挂载到 Pod 中时，应始终更改卷的所有权和权限。这个默认行为。 - "OnRootMismatch" 表示只有在根目录的权限与卷上预期权限不匹配时才会更改卷的所有权和权限。这有助于缩短更改卷的所有权和权限所需的时间。</p>

属性	类型	描述
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seLinuxOptions	seLinuxOptions	要应用到所有容器的 SELinux 上下文。如果未指定，容器运行时将为每个容器分配一个随机 SELinux 上下文。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seccompProfile	SeccompProfile	此 pod 中容器使用的 seccomp 选项。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
supplementalGroups	数组（整数）	除了容器的主 GID、fsGroup（如果指定）以及容器镜像中定义的组成员资格（如果指定）之外，应用于每个容器中的第一个进程运行的组列表。如果未指定，则不会将额外的组添加到任何容器中。请注意，容器进程的 uid 中定义的组成员资格仍然有效，即使它们没有包括在此列表中。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls	数组(Sysctl)	sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。
windowsOptions	WindowsSecurityContextOptions	应用到所有容器的 Windows 特定设置。如果未指定，则使用容器的 SecurityContext 选项。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 linux 时，无法设置此字段。

4.1.97. io.k8s.api.core.v1.PodSpec schema

描述

PodSpec 是 pod 的描述。

类型

object

必填

- **containers**

模式

属性	类型	描述
activeDeadlineSeconds	整数	在系统主动尝试标记失败并终止相关的容器前，pod 可以在相对于 StartTime 的节点上处于活跃状态。值必须是正整数。
关联性	关联性	如果指定，pod 的调度限制

属性	类型	描述
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 指明是否应该自动挂载服务帐户令牌。
containers	数组(Container)	属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。
dnsConfig	PodDNSConfig	指定 pod 的 DNS 参数。此处指定的参数将根据 DNSPolicy 合并到生成的 DNS 配置中。
dnsPolicy	字符串	<p>为 pod 设置 DNS 策略。默认为 "ClusterFirst"。有效值为 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' 或 'None'。DNSConfig 中给定的 DNS 参数将与选择与 DNSPolicy 的策略合并。要将 DNS 选项与 hostNetwork 一起设置，您必须将 DNS 策略明确指定为 'ClusterFirstWithHostNet'。</p> <p>可能枚举值：- "ClusterFirst" 表示 pod 应该先使用集群 DNS，除非 hostNetwork 为 true，如果可用，则回退到默认的（由 kubelet 决定）DNS 设置。- "ClusterFirstWithHostNet" 表示 pod 应该首先使用集群 DNS，如果可用，则回退到它。在默认的（由 kubelet 决定）DNS 设置中。- "Default" 表示 pod 应该使用默认（由 kubelet 决定）DNS 设置。- "None" 表示 pod 应该使用空的 DNS 设置。名称服务器和搜索路径等 DNS 参数应该通过 DNSConfig 定义。</p>
enableServiceLinks	布尔值	EnableServiceLinks 指示是否应将有关服务的信息注入到 pod 的环境变量中，与 Docker 链接的语法匹配。可选：默认为 true。

属性	类型	描述
ephemeralContainers	数组(EphemeralContainer)	此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。
hostAliases	数组(HostAlias)	hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。
hostIPC	布尔值	使用主机的 ipc 命名空间。可选：默认为 false。
hostNetwork	布尔值	为此 pod 请求的主机网络。使用主机的网络命名空间。如果设置了这个选项，则必须指定要使用的端口。默认为 false。
hostPID	布尔值	使用主机的 pid 命名空间。可选：默认为 false。
hostUsers	布尔值	使用主机的用户命名空间。可选：默认为 true。如果设置为 true 或不存在，则 pod 将在主机用户命名空间中运行，当 pod 需要可供主机用户命名空间使用的功能时，如加载带有 CAP_SYS_MODULE 的内核模块。当设置为 false 时，会为 pod 创建一个新的 usersns。设置 false 有助于缓解容器中漏洞，即使允许用户以 root 身份运行其容器，而无需实际在主机上具有 root 特权。此字段是 alpha-level，仅被启用 UserNamespacesSupport 功能的服务器所接受。
hostname	字符串	如果没有指定 Pod 的主机名，则 pod 的主机名将设置为系统定义的值。

属性	类型	描述
imagePullSecrets	数组(LocalObjectReference)	imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息： https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
initContainers	数组(Container)	属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
nodeName	字符串	nodeName 是一个将此 pod 调度到特定节点的请求。如果是非空，调度程序只会将此 pod 调度到该节点上，假设它符合资源要求。
nodeSelector	对象 (字符串)	nodeSelector 是一个选择器，必须为 true 才能使 pod 适合节点。必须与 pod 标签匹配的选择器，才能调度到该节点上。更多信息： https://kubernetes.io/docs/concepts/configuration/assign-pod-node/

属性	类型	描述
OS	PodOS	<p>指定 pod 中容器的 OS。如果设置了，则一些 pod 和容器字段会被限制。</p> <p>如果 OS 字段设为 linux，则必须取消设置以下字段： - securityContext.windowsOptions</p> <p>如果 OS 字段设置为 windows，则必须取消设置以下字段： - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.hostUsers - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser - spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.containers[].securityContext.seLinuxOptions - spec.containers[].securityContext.seccompProfile - spec.securityContext.capabilities - spec.containers[].securityContext.readOnlyRootFilesystem - spec.containers[].securityContext.privileged - spec.containers[].securityContext.allowPrivilegeEscalation - spec.containers[].securityContext.procMount - spec.containers[].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup</p>

属性	类型	描述
开销	对象 (数量)	<p>开销代表了与为给定 RuntimeClass 运行 pod 相关的资源开销。此字段将由 RuntimeClass 准入控制器在准入时自动填充。如果启用了 RuntimeClass 准入控制器，则无法在 Pod 创建请求中设置开销。RuntimeClass 准入控制器将拒绝 Pod 创建已设置开销的请求。如果在 PodSpec 中配置并选择了 RuntimeClass，则 Overhead 将被设置为对应的 RuntimeClass 中定义的值，否则它将保留为未设置并被视为零。更多信息： https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md</p>
preemptionPolicy	字符串	<p>preemptionPolicy 是抢占优先级较低的 pod 的策略。Never, PreemptLowerPriority 之一。如果未设置，则默认为 PreemptLowerPriority。</p> <p>可能枚举值： - "Never" 表示 pod 不会抢占优先级较低的其他 pod。 - "PreemptLowerPriority" 表示 pod 可以抢占优先级较低的其他 pod。</p>
priority	整数	<p>优先级值。各种系统组件使用此字段查找 pod 的优先级。启用 Priority Admission Controller 时，它会阻止用户设置此字段。准入控制器从 PriorityClassName 填充此字段。数值越大，优先级越高。</p>
priorityClassName	字符串	<p>如果指定，则表示 pod 的优先级。"system-node-critical"和"system-cluster-critical"是两个特殊关键字，表示前者最高的优先级。必须使用该名称创建 PriorityClass 对象来定义任何其他名称。如果没有指定，如果没有默认值，pod 优先级将为 default 或零。</p>

属性	类型	描述
readinessGates	数组(PodReadinessGate)	如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪： https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
resourceClaims	数组(PodResourceClaim)	ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。
restartPolicy	字符串	为 pod 中的所有容器重启策略。Always、OnFailure、Never 之一。在某些上下文中，仅允许这些值的子集。默认值为 Always。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy 可能枚举值：- "Always" - "Never" - "OnFailure"
runtimeClassName	字符串	runtimeClassName 指的是 node.k8s.io 组中的 RuntimeClass 对象，它应该用于运行此 pod。如果没有 RuntimeClass 资源与命名类匹配，则不会运行 pod。如果未设置或为空，则使用 "legacy" RuntimeClass，它是一个带有使用默认运行时处理器的空定义的隐式类。更多信息： https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class
schedulerName	字符串	如果指定，pod 将由指定的调度程序分配。如果没有指定，pod 将默认调度程序分配。

属性	类型	描述
schedulingGates	数组(PodSchedulingGate)	<p>SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。</p> <p>SchedulingGates 只能在创建 pod 时设置，之后只能删除。</p> <p>这是 PodSchedulingReadiness 功能门启用的 beta 功能。</p>
securityContext	PodSecurityContext	<p>securityContext 包含 pod 级别的安全属性和通用容器设置。可选：默认为空。有关每个字段的默认值，请参阅类型描述。</p>
serviceAccount	字符串	<p>deprecatedServiceAccount 是 ServiceAccountName 的前文别名。deprecated：改为使用 serviceAccountName。</p>
serviceAccountName	字符串	<p>serviceAccountName 是用于运行此 pod 的 ServiceAccount 的名称。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/</p>
setHostnameAsFQDN	布尔值	<p>如果为 true，则 pod 的主机名将配置为 pod 的 FQDN，而不是叶名称（默认值）。在 Linux 容器中，这意味着在内核的 hostname 字段中设置 FQDN (struct utsname 的 nodename 字段)。在 Windows 容器中，这意味着将 registry 键 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 的 registry 值设置为 FQDN。如果 pod 没有 FQDN，则无效。默认为 false。</p>

属性	类型	描述
shareProcessNamespace	布尔值	在 pod 中所有容器间共享单个进程命名空间。设置此容器后，将能够查看和信号来自同一 pod 中其他容器的进程，并且每个容器中的第一个进程不会被分配 PID 1。无法同时设置 hostPID 和 ShareProcessNamespace。可选：默认为 false。
subdomain	字符串	如果指定，完全限定的 Pod 主机名将是 "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>"。如果没有指定，pod 根本不会具有 domainname。
terminationGracePeriodSeconds	整数	pod 需要安全终止的可选持续时间（以秒为单位）。在删除请求中可能会减少。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。如果这个值为 nil，则使用默认的宽限期。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。默认值为 30 秒。
容限 (tolerations)	数组(Toleration)	如果指定，pod 的容限。
topologySpreadConstraints	数组 (TopologySpreadConstraint)	topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。
卷	数组(Volume)	可由属于 pod 的容器挂载的卷列表。更多信息： https://kubernetes.io/docs/concepts/storage/volumes

4.1.98. io.k8s.api.core.v1.PodTemplateList schema

描述

PodTemplateList 是 PodTemplates 列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(PodTemplate)	pod 模板列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.99. io.k8s.api.core.v1.PodTemplateSpec schema

描述

PodTemplateSpec 描述了从模板创建 pod 时应该具有的数据

类型

对象

模式

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
spec	PodSpec	pod 所需的行为的规格。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

4.1.100. io.k8s.api.core.v1.PortworxVolumeSource schema

描述

PortworxVolumeSource 代表一个 Portworx volume 资源。

类型

object

必填

- **volumelD**

模式

属性	类型	描述
fsType	string	fstype 代表要挂载的文件系统类型，必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs".如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
volumelD	string	VolumelD 唯一标识一个 Portworx 卷

4.1.101. io.k8s.api.core.v1.PreferredSchedulingTerm schema

描述

空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。

类型

object

必填

- **weight**
- **preference**

模式

属性	类型	描述
preference	NodeSelectorTerm	节点选择器术语，与对应的权重关联。
weight	整数	与对应的 nodeSelectorTerm 匹配的权重，范围为 1-100。

4.1.102. io.k8s.api.core.v1.Probe schema

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

对象

模式

属性	类型	描述
exec	ExecAction	exec 指定要执行的操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	GRPCAction	GRPC 指定涉及 GRPC 端口的操作。
httpGet	HTTPGetAction	httpGet 指定要执行的 http 请求。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	TCPsocketAction	tcpSocket 指定涉及 TCP 端口的操作。

属性	类型	描述
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

4.1.103. io.k8s.api.core.v1.ProjectedVolumeSource schema

描述

代表投射卷源

类型

对象

模式

属性	类型	描述
defaultMode	整数	<code>defaultMode</code> 是默认用来对创建文件设置权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 <code>fsGroup</code> ）冲突，结果可以是设置其他模式位。
sources	数组(VolumeProjection)	<code>sources</code> 是卷投射列表

4.1.104. io.k8s.api.core.v1.QuobyteVolumeSource schema

描述

代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- registry
- 卷

模式

属性	类型	描述
group	string	将卷访问映射到 Default 的组没有组
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Quobyte 卷。默认为false。
registry	string	registry 代表一个或多个 Quobyte Registry 服务，作为字符串指定为 host:port 对（多个条目用逗号分隔），后者充当卷的中央 registry
tenant	string	在带有动态置备 Quobyte 卷的后端中拥有给定 Quobyte 卷的租户，值由插件设置
user	string	用户将卷访问权限映射到默认值到 serviceaccount 用户
卷	string	卷是一个字符串，它根据名称引用已创建的 Quobyte 卷。

4.1.105. io.k8s.api.core.v1.RBDPersistentVolumeSource schema

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- monitor
- image

模式

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	secretReference	secretRef 是 RBDUser 的身份验证 secret 的名称。如果提供，则会覆盖密钥环。默认为 nil。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

4.1.106. io.k8s.api.core.v1.RBDVolumeSource schema

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **monitor**
- **image**

模式

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

属性	类型	描述
secretRef	LocalObjectReference	secretRef 是 RBDUser 的身份验证 secret 的名称。如果提供，则会覆盖密钥环。默认为 nil。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

4.1.107. io.k8s.api.core.v1.ReplicationControllerList schema

描述

ReplicationControllerList 是复制控制器的集合。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ReplicationController)	复制控制器列表。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.108. io.k8s.api.core.v1.ResourceClaim schema

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

模式

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

4.1.109. io.k8s.api.core.v1.ResourceFieldSelector schema

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

模式

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

4.1.110. io.k8s.api.core.v1.ResourceQuotaList schema

描述

ResourceQuotaList 是 ResourceQuota 项目的列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ResourceQuota)	items 是 ResourceQuota 对象的列表。更多信息： https://kubernetes.io/docs/concepts/policy/resource-quotas/
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.111. io.k8s.api.core.v1.ResourceRequirements schema

描述

ResourceRequirements 描述计算资源要求。

类型

对象

模式

属性	类型	描述
声明	数组(ResourceClaim)	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
limits	对象 (数量)	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象 (数量)	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

4.1.112. io.k8s.api.core.v1.ScaleIOPersistentVolumeSource schema

描述

ScaleIOPersistentVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- gateway
- system
- secretRef

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认为 "xfs"
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	secretReference	secretRef 对 secret for ScaleIO 用户和其他敏感信息的引用。如果没有提供，则 Login 操作将失败。
sslEnabled	布尔值	sslEnabled 是启用/禁用与网关的 SSL 通信的标记，默认为 false
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

4.1.113. io.k8s.api.core.v1.ScaleIOVolumeSource schema

描述

ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- gateway
- system
- secretRef

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".默认为 "xfs".
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认值为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	LocalObjectReference	secretRef 对 secret for ScaleIO 用户和其他敏感信息的引用。如果没有提供，则 Login 操作将失败。
sslEnabled	布尔值	sslEnabled Flag 启用/禁用与网关的 SSL 通信，默认为 false
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。

属性	类型	描述
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

4.1.114. io.k8s.api.core.v1.SeccompProfile schema

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

模式

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

4.1.115. io.k8s.api.core.v1.SecretEnvSource schema

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

对象

模式

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

4.1.116. io.k8s.api.core.v1.SecretKeySelector schema

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- key

模式

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

4.1.117. io.k8s.api.core.v1.SecretList schema

描述

SecretList 是一个 Secret 列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Secret)	items 是 secret 对象的列表。更多信息： https://kubernetes.io/docs/concepts/configuration/secret
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.118. io.k8s.api.core.v1.SecretProjection schema

描述

将 secret 适应到投射卷中。

目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。

类型

对象

模式

属性	类型	描述
items	数组(KeyToPath)	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥

4.1.119. io.k8s.api.core.v1.SecretReference schema

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

对象

模式

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

4.1.120. io.k8s.api.core.v1.SecretVolumeSource schema

描述

将 Secret 调整为卷。

目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。

类型

对象

模式

属性	类型	描述
defaultMode	整数	defaultMode 是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组(KeyToPath)	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥
secretName	string	secretName 是要使用的 pod 命名空间中的 secret 名称。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#secret

4.1.121. io.k8s.api.core.v1.SecurityContext schema

描述

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	功能	运行容器时添加/过滤的功能。默认为容器运行时授予的默认功能集。请注意，当 spec.os.name 是窗口时，无法设置此字段。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
procMount	字符串	procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。 可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。
readOnlyRootFilesystem	布尔值	此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seLinuxOptions	seLinuxOptions	要应用到容器的 SELinux 上下文。如果未指定，容器运行时将为每个容器分配一个随机 SELinux 上下文。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seccompProfile	SeccompProfile	此容器使用的 seccomp 选项。如果 pod 和容器级别都提供了 seccomp 选项，容器选项将覆盖 pod 选项。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
windowsOptions	WindowsSecurityContextOptions	应用到所有容器的 Windows 特定设置。如果未指定，则使用 PodSecurityContext 中的选项。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 linux 时，无法设置此字段。

4.1.122. io.k8s.api.core.v1.SELinuxOptions schema

描述

seLinuxOptions 是要应用到容器的标签

类型

对象

模式

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

4.1.123. io.k8s.api.core.v1.ServiceAccountList schema

描述

ServiceAccountList 是 ServiceAccount 对象列表

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ServiceAccount)	ServiceAccounts 列表。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.124. io.k8s.api.core.v1.ServiceAccountTokenProjection schema

描述

ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

类型

object

必填

- path

模式

属性	类型	描述
受众	string	受众是令牌的预期受众。令牌的接收者必须使用令牌使用者中指定的标识符来识别自己，否则应拒绝令牌。使用者默认为 apiserver 的标识符。
expirationSeconds	整数	expirationSeconds 是服务帐户令牌的请求持续时间。随着令牌方法到期，kubelet 卷插件将主动轮转服务帐户令牌。如果令牌存在的时间超过这个值的 80%，或者令牌超过 24 小时，则 kubelet 将开始尝试轮转令牌，且必须至少为 10 分钟。
path	string	path 是相对于文件挂载点的路径，以将令牌项目到其中。

4.1.125. io.k8s.api.core.v1.ServiceList schema

描述

ServiceList 包含服务列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Service)	服务列表

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.126. io.k8s.api.core.v1.StorageOSPersistentVolumeSource schema

描述

代表 StorageOS 持久性卷资源。

类型

对象

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	ObjectReference	secretRef 指定用于获取 StorageOS API 凭证的 secret。如果没有指定，则会尝试默认值。
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。

属性	类型	描述
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

4.1.127. io.k8s.api.core.v1.StorageOSVolumeSource schema

描述

代表 StorageOS 持久性卷资源。

类型

对象

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs"。如果未指定，则隐式推断为 "ext4"。
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	LocalObjectReference	secretRef 指定用于获取 StorageOS API 凭证的 secret。如果没有指定，则会尝试默认值。
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。

属性	类型	描述
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

4.1.128. io.k8s.api.core.v1.Sysctl schema

描述

sysctl 定义要设置的内核参数

类型

object

必填

- **name**
- **value**

模式

属性	类型	描述
name	字符串	要设置的属性的名称
value	字符串	要设置的属性值

4.1.129. io.k8s.api.core.v1.TCPSocketAction schema

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

模式

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

4.1.130. io.k8s.api.core.v1.Toleration schema

描述

附加到此 Toleration 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。

类型

对象

模式

属性	类型	描述
effect	字符串	<p>effect 表示污点效果要匹配。空意味着匹配所有污点效果。指定后，允许的值为 NoSchedule、PreferNoSchedule 和 NoExecute。</p> <p>可能枚举值： - "NoExecute" Evict 任何已在运行的 pod，这些 pod 不容许该污点。目前由 NodeController 实施。 - "NoSchedule" 不允许新 pod 调度到该节点上，除非它们容许污点，但允许提交给 Kubelet 的所有 pod 而无需启动调度程序，并允许所有已在运行的 pod 继续运行。调度程序强制使用 - "PreferNoSchedule" Like TaintEffectNoSchedule，但调度程序会尝试不将新 pod 调度到节点上，而不是禁止新 pod 完全调度到该节点上。由调度程序实施。</p>
key	字符串	<p>key 是容限应用到的污点键。empty 表示与所有污点键匹配。如果键为空，则运算符必须是 Exists；此组合意味着匹配所有值和所有键。</p>

属性	类型	描述
operator	字符串	Operator 代表键与值的关系。有效的运算符是 Exists 和 Equal。默认为 Equal。exists 等同于 value 的通配符，以便 pod 可以容忍特定类别的所有污点。 可能枚举值： - "Equal" - "Exists"
tolerationSeconds	整数	TolerationSeconds 代表容限的期间（该容限必须是 NoExecute，否则将忽略此字段）容许污点。默认情况下，它没有被设置，这意味着容许永久污点（不驱除）。系统会将零值和负值视为 0（立即撤离）。
value	字符串	值是容限匹配的污点值。如果运算符是 Exists，则该值应为空，否则只是一个常规字符串。

4.1.131. io.k8s.api.core.v1.TopologySelectorLabelRequirement schema

描述

拓扑选择器要求是与给定标签匹配的选择器。这是一个 alpha 功能，将来可能会改变。

类型

object

必填

- key
- values

模式

属性	类型	描述
key	字符串	选择器应用到的标签键。
values	数组（字符串）	字符串值数组。一个值必须与要选择的标签匹配。Values 中的每个条目都是 ORed。

4.1.132. io.k8s.api.core.v1.TopologySelectorTerm schema

描述

拓扑选择器术语代表标签查询的结果。null 或空拓扑选择器术语不匹配任何对象。它们的要求是 AND。它提供功能子集作为 NodeSelectorTerm。这是一个 alpha 功能，将来可能会改变。

类型

对象

模式

属性	类型	描述
matchLabelExpressions	数组 (TopologySelectorLabelRequirement)	按标签划分的拓扑选择器要求列表。

4.1.133. io.k8s.api.core.v1.TopologySpreadConstraint schema

描述

TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。

类型

object

必填

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

模式

属性	类型	描述
labelSelector	labelSelector	labelSelector 用于查找匹配的 pod。与此标签选择器匹配的 Pod 被计算，以确定其对应拓扑域中的 pod 数量。
matchLabelKeys	数组（字符串）	matchLabelKeys 是一组 pod 标签键，用于选择要计算分布的 pod。键用于从传入的 pod 标签中查找值，这些键值标签带有 labelSelector 的 ANDed 来选择要针对传入 pod 计算的现有 pod 组。禁止在 MatchLabelKeys 和 LabelSelector 中存在相同的键。当 LabelSelector 没有设置时，无法设置 matchLabelKeys。传入的 pod 标签中不存在的键将被忽略。null 或 empty 列表表示仅匹配 labelSelector。 这是一个 beta 字段，需要启用 MatchLabelKeysInPodTopologySpread 功能门（默认启用）。

属性	类型	描述
maxSkew	整数	<p>maxSkew 描述了 pod 可能不均匀分布的程度。当 whenUnsatisfiable=DoNotSchedule 时，目标拓扑中匹配 pod 数量和全局最小值之间允许的最大区别。如果有资格的域数量小于 MinDomains，则全局最小值是符合域或零的匹配 pod 的最小数量。例如，在 3 区集群中，MaxSkew 设置为 1，具有相同 labelSelector 的 pod 分布为 2/2/1：在这种情况下，全局最小值为 1。 zone1 zone2 zone3 P P P P P P P P P - 如果 MaxSkew 为 1，则传入的 pod 只能调度到 zone3，以变为 2/2/2；将其调度到 zone1 (zone2)上的 ActualSkew (3-1)违反 MaxSkew (1)- if MaxSkew.当 whenUnsatisfiable=ScheduleAnyway 时，用来为满足它的拓扑提供更高优先级。它是必填字段。默认值为 1，不允许使用 0。</p>

属性	类型	描述
minDomains	整数	<p>MinDomains 表示最少的合格域数。当匹配拓扑键的合格域数量小于 minDomains 时，Pod Topology Spread 会将"global minimum"视为 0，然后执行 Skew 计算。当匹配拓扑键的合格域数量等于或大于 minDomains 时，这个值对调度没有影响。因此，当有资格的域数量小于 minDomains 时，调度程序不会将这些域的 maxSkew Pod 调度到这些域。如果值为 nil，则约束的行为就像 MinDomains 等于 1。有效值为大于 0 的整数。当值不是 nil 时，WhenUnsatisfiable 必须是 DoNotSchedule。</p> <p>例如，在 3 区集群中，MaxSkew 设置为 2，MinDomains 被设置为 5，而带有与 2/2/2: zone1 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone3 zone3 zone3 区域数量小于 5 (MinDomains)，因此"global minimum"被视为 0。在这种情况下，无法调度具有相同 labelSelector 的新 pod，因为如果新 Pod 调度到三个区，则 computed skew 将为 3 (3 - 0)，它将违反 MaxSkew。</p> <p>这是一个 beta 字段，需要启用 MinDomainsInPodTopologySpread 功能门（默认启用）。</p>

属性	类型	描述
nodeAffinityPolicy	string	<p>NodeAffinityPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理 Pod 的 nodeAffinity/nodeSelector。选项为：- Honor：计算中仅包含与 nodeAffinity/nodeSelector 匹配的节点。忽略 Ignore: nodeAffinity/nodeSelector。所有节点包含在计算中。</p> <p>如果这个值为 nil，则行为等同于 Honor 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>
nodeTaintsPolicy	string	<p>NodeTaintsPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理节点污点。选项为：- Honor: 没有污点的节点，以及传入 pod 具有容量的污点节点被包含。- Ignore: node taint is ignored。包括所有节点。</p> <p>如果这个值为 nil，则行为等同于 Ignore 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>

属性	类型	描述
topologyKey	字符串	<p>topologyKey 是节点标签的密钥。带有具有此键和相同值标签的节点被视为在同一拓扑中。我们将每个 <key, value> 视为 "bucket", 并尝试在每个存储桶中均衡的 pod 数量。我们将域定义为拓扑的特定实例。另外, 我们定义了一个有资格的域, 其节点满足 nodeAffinityPolicy 和 nodeTaintsPolicy 的要求。例如, 如果 TopologyKey 为 "kubernetes.io/hostname", 则每个节点都是该拓扑的域。如果 TopologyKey 是 "topology.kubernetes.io/zone", 则每个区都是该拓扑的域。它是必填字段。</p>
whenUnsatisfiable	字符串	<p>whenUnsatisfiable 表示在不满足分散约束的情况下如何处理 pod。</p> <ul style="list-style-type: none"> - DoNotSchedule (默认) 告知调度程序不调度它。 - ScheduleAnyway 告知调度程序将 pod 调度到任何位置, 但为拓扑赋予更高优先级, 有助于减少偏移量。只有在每个可能的节点分配在某些拓扑中违反 "MaxSkew" 时, 对传入 pod 的一个约束被视为 "Unsatisfiable"。例如, 在 3 区集群中, MaxSkew 设置为 1, 具有相同 labelSelector 的 pod 分布为 3/1/1: zone1 zone3 zone3 P P P P If WhenUnsatisfiable 被设置为 DoNotSchedule, 传入的 pod 只能调度到 zone2 (zone3) 要成为 3/2/1 (3/1/2), 在 zone2 (zone3) 上作为 ActualSkew (2-1) 满足 MaxSkew (1)。换句话说, 集群仍然可以进行平衡, 但调度程序不会使其更不平衡。它是必填字段。 <p>可能枚举值 : -</p> <ul style="list-style-type: none"> "DoNotSchedule" 指示调度程序在不满足约束时不调度 pod。 "ScheduleAnyway" 指示调度程序调度 pod, 即使没有满足限制。

4.1.134. io.k8s.api.core.v1.TypedLocalObjectReference schema

描述

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- **kind**
- **name**

模式

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

4.1.135. io.k8s.api.core.v1.TypedObjectReference schema**描述****类型**

object

必填

- **kind**
- **name**

模式

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

属性	类型	描述
namespace	string	namespace 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 gateway.networking.k8s.io/ReferenceGrant 对象，以允许该命名空间的所有者接受引用。详情请查看 ReferenceGrant 文档。(alpha)此字段需要启用 CrossNamespaceVolumeDataSource 功能门。

4.1.136. io.k8s.api.core.v1.Volume schema

描述

卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

类型

object

必填

- **name**

模式

属性	类型	描述
awsElasticBlockStore	AWSElasticBlockStoreVolumeSource	awsElasticBlockStore 代表附加到 kubelet 主机机器的 AWS Disk 资源，然后公开给 pod。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
azureDisk	AzureDiskVolumeSource	azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。
azureFile	AzureFileVolumeSource	azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。
cephfs	CephFSVolumeSource	CephFS 代表共享 pod 生命周期的主机上的 Ceph FS 挂载
cinder	CinderVolumeSource	Cinder 代表附加并挂载到 kubelet 主机上的 cinder 卷。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

属性	类型	描述
configMap	ConfigMapVolumeSource	ConfigMap 代表一个应该填充此卷的 configMap
csi	CSIVolumeSource	CSI（容器存储接口）代表由某些外部 CSI 驱动程序(Beta 功能)处理的临时存储。
downwardAPI	DownwardAPIVolumeSource	downwardAPI 代表有关应该填充此卷的 pod 的 Downward API
emptyDir	EmptyDirVolumeSource	emptyDir 代表共享 pod 生命周期的临时目录。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir
ephemeral	EphemeralVolumeSource	<p>ephemeral 代表由集群存储驱动程序处理的卷。卷的生命周期与定义它的 pod 关联 - 它将在 pod 启动前创建，并在 pod 被删除时删除。</p> <p>使用这个选项时：只有在 pod 运行、b) 常规卷（如从快照或容量跟踪中恢复）时，才需要使用这个卷。c) 存储驱动程序通过存储类来指定，而 d) 存储驱动程序支持通过 PersistentVolumeClaim 进行动态卷置备（请参阅 EphemeralVolumeSource）。</p> <p>对于比单个 pod 生命周期更长的卷，使用 PersistentVolumeClaim 或特定于供应商的 API。</p> <p>如果 CSI 驱动程序旨在以这种方式使用该方法，请使用 CSI 作为轻量级本地临时卷 - 如需更多信息，请参阅驱动程序文档。</p> <p>pod 可以同时使用两种类型的临时卷和持久性卷。</p>
fc	FCVolumeSource	FC 代表附加到 kubelet 主机的 Fibre Channel 资源，然后公开给 pod。
flexVolume	FlexVolumeSource	FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。

属性	类型	描述
flocker	FlockerVolumeSource	flocker 代表附加到 kubelet 主机机器的 Flocker 卷。这取决于运行 Flocker 控制服务
gcePersistentDisk	GCEPersistentDiskVolumeSource	gcePersistentDisk 代表附加到 kubelet 主机机器的 GCE Disk 资源，然后公开给 pod。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
gitRepo	GitRepoVolumeSource	gitRepo 代表特定修订版本的 git 存储库。DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。
glusterfs	GlusterfsVolumeSource	GlusterFS 代表共享 pod 生命周期的主机上的 Glusterfs 挂载。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md
hostPath	HostPathVolumeSource	hostpath 代表主机机器上直接公开给容器已存在的文件或目录。这通常用于系统代理或其他允许查看主机机器的特权。大多数容器都不需要此操作。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath
iscsi	ISCSIVolumeSource	iSCSI 代表附加到 kubelet 主机机器的 iSCSI 磁盘资源，然后公开给 pod。更多信息： https://examples.k8s.io/volumes/iscsi/README.md
name	string	卷的名称。必须是 DNS_LABEL，且在 pod 中唯一。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	NFSVolumeSource	NFS 代表共享 pod 生命周期的主机上的 NFS 挂载： https://kubernetes.io/docs/concepts/storage/volumes#nfs

属性	类型	描述
persistentVolumeClaim	PersistentVolumeClaimVolumeSource	persistentVolumeClaimVolumeSource 代表对同一命名空间中的 PersistentVolumeClaim 的引用。 更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
photonPersistentDisk	PhotonPersistentDiskVolumeSource	photonPersistentDisk 代表在 kubelets 主机上附加并挂载的 PhotonController 持久磁盘
portworxVolume	PortworxVolumeSource	portworxVolume 代表附加并挂载到 kubelet 主机上的 portworx 卷
projected	ProjectedVolumeSource	在一个资源 secret、configmap 和 Downward API 中所有的项目
quobyte	QuobyteVolumeSource	quobyte 代表共享 pod 生命周期的主机上的 Quobyte 挂载
rbd	RBDVolumeSource	RBD 代表共享 pod 生命周期的主机上的 Rados 块设备挂载。更多信息： https://examples.k8s.io/volumes/rbd/README.md
scaleIO	ScaleIOVolumeSource	scaleIO 代表一个在 Kubernetes 节点上附加并挂载的 ScaleIO 持久性卷。
secret	SecretVolumeSource	secret 代表应填充此卷的 secret。 更多信息： https://kubernetes.io/docs/concepts/storage/volumes#secret
storageos	StorageOSVolumeSource	storageos 代表在 Kubernetes 节点上附加并挂载的 StorageOS 卷。
vsphereVolume	VsphereVirtualDiskVolumeSource	vsphereVolume 代表在 kubelets 主机上附加并挂载的 vSphere 卷

4.1.137. io.k8s.api.core.v1.VolumeDevice schema

描述

volumeDevice 描述容器中原始块设备的映射。

类型

object**必填**

- **name**
- **devicePath**

模式

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

4.1.138. io.k8s.api.core.v1.VolumeMount schema**描述**

VolumeMount 描述了在容器中挂载卷。

类型**object****必填**

- **name**
- **mountPath**

模式

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 '..'。

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值： - "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。 - "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。 - "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

4.1.139. io.k8s.api.core.v1.VolumeNodeAffinity schema

描述

VolumeNodeAffinity 定义限制此卷可从中访问哪些节点。

类型

对象

模式

属性	类型	描述
required	NodeSelector	必需指定必须满足的硬节点限制。

4.1.140. io.k8s.api.core.v1.VolumeProjection schema

描述

可与其他支持的卷类型一起投射的预测

类型

对象

模式

属性	类型	描述
configMap	ConfigMapProjection	有关项目的 configMap 数据的 ConfigMap 信息
downwardAPI	DownwardAPIProjection	downwardAPI有关项目的 downwardAPI 数据的信息
secret	SecretProjection	有关项目的 secret 数据的机密信息
serviceAccountToken	ServiceAccountTokenProjection	serviceAccountToken 是项目的 serviceAccountToken 数据的信息

4.1.141. io.k8s.api.core.v1.VsphereVirtualDiskVolumeSource schema

描述

代表 vSphere 卷资源。

类型

object

必填

- **volumePath**

模式

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。

属性	类型	描述
storagePolicyID	string	StoragePolicyID 是与 StoragePolicyName 关联的存储 Policy Based Management (SPBM)配置集 ID。
storagePolicyName	string	StoragePolicyName 是存储基于策略的管理(SPBM)配置集名称。
volumePath	string	volumePath 是标识 vSphere 卷 vmdk 的路径

4.1.142. io.k8s.api.core.v1.WeightedPodAffinityTerm schema

描述

所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- **weight**
- **podAffinityTerm**

模式

属性	类型	描述
podAffinityTerm	PodAffinityTerm	必需。pod 关联性术语，与对应的权重关联。
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

4.1.143. io.k8s.api.core.v1.WindowsSecurityContextOptions schema

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 Username，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

4.1.144. io.k8s.api.discovery.v1.EndpointSliceList schema

描述

EndpointSliceList 代表端点片段列表

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(EndpointSlice)	items 是端点片段列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。

4.1.145. io.k8s.api.events.v1.EventList schema

描述

EventList 是 Event 对象列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
items	数组（事件）	items 是 schema 对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.146. io.k8s.api.flowcontrol.v1beta3.FlowSchemaList schema

描述

FlowSchemaList 是 FlowSchema 对象列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(FlowSchema)	items 是 FlowSchemas 列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	metadata 是标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.147. io.k8s.api.flowcontrol.v1beta3.PriorityLevelConfigurationList schema

描述

PriorityLevelConfigurationList 是 PriorityLevelConfiguration 对象的列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (PriorityLevelConfiguration)	items 是 request-priorities 的列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	metadata 是标准对象的元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.148. io.k8s.api.networking.v1.IngressClassList schema

描述

IngressClassList 是 IngressClasses 的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(IngressClass)	items 是 IngressClasses 列表。

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。

4.1.149. io.k8s.api.networking.v1.IngressList schema

描述

IngressList 是 Ingress 的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Ingress)	items 是 Ingress 列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ListMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.150. io.k8s.api.networking.v1.NetworkPolicyList schema

描述

NetworkPolicyList 是 NetworkPolicy 对象列表。

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(NetworkPolicy)	items 是 schema 对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.151. io.k8s.api.node.v1.RuntimeClassList schema

描述

RuntimeClassList 是 RuntimeClass 对象列表。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(RuntimeClass)	items 是 schema 对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.152. io.k8s.api.policy.v1.PodDisruptionBudgetList schema

描述

PodDisruptionBudgetList 是 PodDisruptionBudgets 的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(PodDisruptionBudget)	items 是 PodDisruptionBudgets 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.153. io.k8s.api.rbac.v1.ClusterRoleBindingList schema

描述

ClusterRoleBindingList 是 ClusterRoleBindings 的集合

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(ClusterRoleBinding)	items 是 ClusterRoleBindings 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象元数据。

4.1.154. io.k8s.api.rbac.v1.ClusterRoleList schema

描述

ClusterRoleList 是 ClusterRole 的集合

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
items	数组(ClusterRole)	items 是 ClusterRole 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象元数据。

4.1.155. io.k8s.api.rbac.v1.RoleBindingList schema

描述

RoleBindingList 是 RoleBindings 的集合

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(RoleBinding)	items 是 RoleBindings 列表

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象元数据。

4.1.156. io.k8s.api.rbac.v1.RoleList schema

描述

RoleList 是角色的集合

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(Role)	items 是角色列表

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象元数据。

4.1.157. io.k8s.api.scheduling.v1.PriorityClassList schema

描述

PriorityClassList 是优先级类的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(PriorityClass)	items 是 PriorityClasses 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.158. io.k8s.api.storage.v1.CSIDriverList schema

描述

CSIDriverList 是 CSIDriver 对象的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(CSIDriver)	items 是 CSIDriver 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.159. io.k8s.api.storage.v1.CSINodeList schema

描述

CSINodeList 是 CSINode 对象的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(CSINode)	items 是 CSINode 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.160. io.k8s.api.storage.v1.CSISStorageCapacityList schema

描述

CSISStorageCapacityList 是 CSISStorageCapacity 对象的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(CSIStorageCapacity)	items 是 CSIStorageCapacity 对象列表。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.161. io.k8s.api.storage.v1.StorageClassList schema

描述

StorageClassList 是存储类的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(StorageClass)	items 是 StorageClasses 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.162. io.k8s.api.storage.v1.VolumeAttachmentList schema

描述

VolumeAttachmentList 是 VolumeAttachment 对象的集合。

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组(VolumeAttachment)	items 是 VolumeAttachments 列表
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.163. io.k8s.apiextensions-apiserver.pkg.apis.apiextensions.v1.CustomResourceDefinitionList schema

描述

CustomResourceDefinitionList 是 CustomResourceDefinition 对象列表。

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items	数组 (CustomResourceDefinition)	items 列出单个 CustomResourceDefinition 对象
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准对象的元数据更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

4.1.164. io.k8s.apixtensions-apiserver.pkg.apis.apixtensions.v1.ExternalDocumentation schema

描述

外部文档允许引用外部资源以获取扩展文档。

类型

对象

模式

属性	类型	描述
description	string	
url	string	

4.1.165. io.k8s.apixtensions-apiserver.pkg.apis.apixtensions.v1.JSON schema

描述

JSON 代表任何有效的 JSON 值。支持这些类型：bool, int64, float64, string, []interface{}, map[string]interface{} 和 nil。

类型

..

4.1.166. io.k8s.apiextensions-apiserver.pkg.apis.apiextensions.v1.JSONSchemaProps schema

描述

JSONSchemaProps 是符合规范 4 的 JSON 架构(<http://json-schema.org/>)。

类型

对象

模式

属性	类型	描述
\$ref	字符串	
\$schema	字符串	
additionalItems	JSONSchemaPropsOrBool	
additionalProperties	JSONSchemaPropsOrBool	
allOf	数组 (未定义)	
anyOf	数组 (未定义)	
default	JSON	Default 是未定义对象字段的默认值。默认是 CustomResourceDefaulting 功能门下的 beta 功能。默认要求 spec.preserveUnknownFields 为 false。
定义	对象 (未定义)	
dependencies	对象 (未定义)	
description	字符串	
Enum	数组(JSON)	
示例	JSON	
exclusiveMaximum	布尔值	
exclusiveMinimum	布尔值	

属性	类型	描述
externalDocs	外部文档	
格式	字符串	<p>格式是 OpenAPI v3 格式字符串。未知格式将被忽略。验证以下格式：</p> <ul style="list-style-type: none"> - bsonobjectid: 一个 bson 对象 ID, 即 24 个字符十六进制字符串 - uri: 一个 URI 作为 Golang net/url.ParseRequestURI 解析的 - email: 一个电子邮件地址, 由 Golang net/mail.ParseAddress - hostname: 一个 Internet 主机名的有效表示, 如 RFC 1034, section 3.1 [RFC1034] 所定义。 - ipv4: 一个 IPv4 IP 由 Golang net.ParseIP - ipv6: ipv6: 一个 IPv6 IP 解析, 由 Golang net.ParseIP - cidr: 一个 CIDR 作为 Golang net.ParseCIDR - mac: MAC 地址解析, 如 Golang net.ParseMAC - uuid: 一个 UUID, 允许由 regex <code>(?i)^[0-9a-f]{8}-?[8\]-?[a]{8}\-</code> - uuid3: 一个 UUID3, 它允许由 regex <code>(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?3[0-9a-f]{3}-?[0-9a-f]{4}-?[0-9a-f]{12}\$</code> - uuid4: 一个 UUID4, 它允许由 regex <code>(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?4[0-9a-f]{3}-?[89ab][0-9a-f]{3}-?[0-9a-f]{12}\$</code> - uuid5: 一个 UUID5, 它允许由 regex <code>(?i)^[0-9a-f]{8}-?[0-9a-f]{4}-?5[0-9a-f]{3}-?[89ab][0-9a-f]{3}-?[0-9a-f]{12}\$</code> - isbn: 一个 ISBN10 或 ISBN13 number 字符串, 如 "0321751043" 或 "978-0321751041" - isbn10: a ISBN10 number string like "0321751043" - isbn13: an ISBN13 number string, 如 "978-0321751041" - creditcard: 一个信用卡号, 由 regex <code>^(?:4[0-9]{12}(?:[0-9]{3})? 5[1-5][0-9]{14} 6(?:011 5[0-9][0-9])[0-9]{12} 3[47][0-9]{13} (?:[0-5][8-5][1][1][0-9])</code> 如果有任何非数字字符混合在 - ssn 中: 一个 U.S. social security number after the regex <code>^\d{3}[-]?\d{2}[-]?\d{4}\$</code> - hexcolor: 一个十六进制颜色代码, 如 "FFFFFF: following regex <code>^? ([0-9a-fA-F]{3}[-F]{3}[-F]\{3\}\$</code> - rgbcolor: 一个 RGB 颜色代码, 如 "rgb (255,

属性	类型	描述
		255,2559" - byte: base64 编码的 二进制数据 - password: any kind of string - date - date string like "2006-01-02" like "2006-01-02" defined by RFC3339 - duration: 一个持续时间字符串, 如 Golang time.ParseDuration 或与 Scala 持 续时间格式兼容, 如 "2014-12- 15T19:30:20.000:30:20.000"
id	字符串	
items	JSONSchemaPropsOrArray	
maxItems	整数	
maxLength	整数	
maxProperties	整数	
最大	number	
minItems	整数	
minLength	整数	
minProperties	整数	
最小值	number	
multipleOf	number	
not	JSONSchemaProps	
nullable	布尔值	
oneOf	数组 (未定义)	
pattern	字符串	
patternProperties	对象 (未定义)	
属性	对象 (未定义)	
required	数组 (字符串)	
title	字符串	
type	字符串	

属性	类型	描述
uniqueItems	布尔值	
x-kubernetes-embedded-resource	布尔值	x-kubernetes-embedded-resource 定义值是嵌入式 Kubernetes runtime.Object，带有 TypeMeta 和 ObjectMeta。类型必须是 object。允许进一步限制嵌入式对象。apiVersion 和 metadata 会自动验证。x-kubernetes-preserve-unknown-fields 为 true，但如果对象被完全指定（默认为 apiVersion, metadata）不需要被验证。
x-kubernetes-int-or-string	布尔值	x-kubernetes-int-or-string 指定这个值是一个整数或字符串。如果为 true，则允许一个空类型，如果遵循以下模式之一，则允许作为 anyOf 的子类型： 1) anyOf: - type: integer - type: string 2) allOf: - anyOf: - type: integer - type: string - ... zero or more
x-kubernetes-list-map-keys	数组（字符串）	x-kubernetes-list-map-keys 通过指定用作 映射 索引的键来注解带有 x-kubernetes-list-type 映射的数组。 此标签仅应用于将 "x-kubernetes-list-type" 扩展设置为 "map" 的列表。另外，为此属性指定的值必须是子结构的 scalar 类型字段（不支持嵌套）。 指定的属性必须是 required 或具有默认值，以确保所有列表项都存在这些属性。

属性	类型	描述
x-kubernetes-list-type	字符串	<p>x-kubernetes-list-type 注解一个数组来进一步描述其拓扑。这个扩展必须只用于列表，且可能有 3 个可能的值：</p> <p>1) Atomic：列表被视为单一实体，如 scalar。更新时，原子列表将完全被替换。此扩展可用于任何类型的列表(struct, scalar, ...)。2) set: Sets 是不能具有相同值的多个项目的列表。每个值都必须是一个 scalar，一个具有 x-kubernetes-map-type atomic 的对象或带有 x-kubernetes-list-type atomic 的数组。3) 映射：这些列表类似于映射，其元素具有用于识别它们的非索引键。订购会在合并时保留。map 标签必须仅用于带有 type 对象的元素的列表。对于数组，默认为 atomic。</p>
x-kubernetes-map-type	字符串	<p>x-kubernetes-map-type 注解一个对象来进一步描述其拓扑。只有在类型为 object 且可能具有 2 个可能的值时，才使用此扩展：</p> <p>1) 精细的：这些映射是实际映射（键值对），每个字段都相互独立（它们可以相互独立操作）。这是所有映射的默认行为。2) Atomic：列表被视为单个实体，如 scalar。更新时，原子映射将完全被替换。</p>
x-kubernetes-preserve-unknown-fields	布尔值	<p>x-kubernetes-preserve-unknown-fields 会停止 API 服务器解码步骤，从修剪没有在验证 schema 中指定的字段。这会以递归方式影响字段，但如果在 schema 中指定嵌套属性或附加属性，则切换回正常的修剪行为。这可以是 true 或 undefined。false 被禁止。</p>

属性	类型	描述
x-kubernetes-validations	数组(ValidationRule)	x-kubernetes-validations 描述使用 CEL 表达式语言编写的验证规则列表。此字段是一个 alpha-level。使用此字段需要启用功能门 CustomResourceValidationExpressions 。

4.1.167. io.k8s.apimachinery.pkg.apis.extensions.v1.JSONSchemaPropsOrArray schema

描述

JSONSchemaPropsOrArray 代表一个值，可以是 JSONSchemaProps 或 JSONSchemaProps 数组。这里主要用于序列化目的。

类型

..

4.1.168. io.k8s.apimachinery.pkg.apis.extensions.v1.JSONSchemaPropsOrBool schema

描述

JSONSchemaPropsOrBool 代表 JSONSchemaProps 或布尔值。对于布尔值属性，默认为 true。

类型

..

4.1.169. io.k8s.apimachinery.pkg.apis.extensions.v1.JSONSchemaPropsOrStringArray schema

描述

JSONSchemaPropsOrStringArray 代表 JSONSchemaProps 或字符串数组。

类型

..

4.1.170. io.k8s.apimachinery.pkg.apis.extensions.v1.ValidationRule schema

描述

ValidationRule 描述了使用 CEL 表达式语言编写的验证规则。

类型

object

必填

- 规则

模式

属性	类型	描述
message	string	Message 代表验证失败时显示的消息。如果 Rule 包含换行符，则需要消息。消息不得包含换行符。如果未设置，则消息为 "failed rule: {Rule}"。例如"必须是一个带有主机匹配 spec.host" 的 URL。
messageExpression	string	MessageExpression 声明了一个 CEL 表达式，用于评估此规则失败时返回的验证失败消息。由于 messageExpression 被用作失败消息，因此它必须评估为字符串。如果规则中存在 message 和 messageExpression，则在验证失败时将使用 messageExpression。如果 messageExpression 产生运行时错误，则会记录运行时错误，并且生成验证失败消息，就像 messageExpression 字段未设置一样。如果 messageExpression 评估为一个空字符串、仅包含空格的字符串或包含换行符的字符串，则验证失败消息也会生成，就像 messageExpression 字段未设置一样，而 messageExpression 的事实会生成空字符串/字符串，且带有换行符的字符串才会被记录。messageExpression 可以访问所有与规则相同的变量；唯一的区别是返回类型。示例："x 必须小于 max ("string (self.max))")"
规则	string	<p>规则代表 CEL 将评估的表达式。 ref: https://github.com/google/cel-spec 规则的范围仅限于架构中 x-kubernetes-validations 扩展的位置。CEL 表达式中的 self 变量绑定到范围值。示例：- 规则范围到具有状态子资源的资源根：{"rule": "self.status.actual categories self.spec.maxDesired"}</p> <p>如果 Rule 限定有属性的对象，则可通过 self.field 和 field presence (self.field) 检查对象的可访问属性。null valued 字段被视为 CEL 表达式中的 absent 字段。如果 Rule 的范围仅限于带有额外属性（如映射）的对象，则映射的值可以通过 self[mapKey] 访问，则可以通过 map containment 在 self 中通过 mapKey 进行检</p>

属性	类型	描述
		<p>查，并且映射的所有条目都可通过宏和函数（如 self.all (...)）访问。如果 Rule 限定在一个数组中，则数组的元素可以通过 self[i] 访问，也可由宏和函数访问。如果 Rule 限定于一个 scalar，则 self 绑定到 scalar 值。示例：- Rule 范围到对象映射：{"rule": "self.components['Widget'].priority < 10"} - Rule scoped to a integers: {"rule": "self.values.all (value, value >= 0 && value < 100)"} - Rule scoped to a string value: {"rule": "self.startsWith ('kube')"} }</p> <p>apiVersion,kind,metadata.name 和 metadata.generateName 始终可从对象的根目录和任何 x-kubernetes-embedded-resource 注解的对象访问。没有可以访问其他元数据属性。</p> <p>CEL 表达式中无法通过 x-kubernetes-preserve-unknown-fields 在自定义资源中保留未知数据。这包括 - Unknown 字段值，由带有 x-kubernetes-preserve-unknown-fields. - 对象属性 schema 为 "unknown type" 的对象模式保留。"未知类型"被递归定义为：- 无类型和 x-kubernetes-preserve-unknown-fields 的模式设置为 true - items 模式是"未知类型"的数组 - additionalProperties 模式是"unknown type"的对象</p> <p>仅可以访问 [a-zA-Z_./][a-zA-Z0-9_./]* 的属性名称。在表达式中访问时可访问的属性名称会被转义：- "escapes to <code>'_underscores_'</code> - <code>'.'</code> escapes to <code>'__dot__'</code> - <code>'-'</code> escapes to <code>'__dash__'</code> - <code>'/'</code> escapes to <code>'__slash__'</code> - Property name is exactly match a CEL RESERVEDVED_{key} - <code>'/'</code> escapes to <code>'_underscores_'</code> - <code>'.'</code> escapes to <code>'__dot__'</code> 关键字 为：<code>"true"</code>, <code>"false"</code>, <code>"null"</code>, <code>"in"</code>, <code>"in"</code>, <code>"as"</code>, <code>"break"</code>, <code>"const"</code>, <code>"continue"</code>, <code>"else"</code>, <code>"for"</code>, <code>"if"</code>, <code>"import"</code>, <code>"let"</code>, <code>"loop"</code>, <code>"package"</code>, <code>"namespace"</code>, <code>"namespace"</code>, <code>"return"</code>。示例：- 访问名为 "namespace": {"rule": self.__namespace__ → 0} 的属性 - 规则访问名为 "x-prop": {"rule":</p>

属性	类型	描述
		<p><code>self.x_dash_prop → 0}</code> 的属性 - 访问名为 "redactd": {"rule": "self.redact_underscores_d → 0}</p> <p>在带有 <code>x-kubernetes-list-type</code> 为 'set' 或 'map' 的数组中相等，例如 <code>[1, 2] == [2, 1]</code>。带有 <code>x-kubernetes-list-type</code> 的数组上的串联使用列表类型的语义： - 'set': X + Y 会执行一个组合，其中 X 中所有元素的数组位置会被保留，而 Y 中的非中间元素被附加，保留其部分顺序。 - 'map': X + Y 执行一个合并，其中 X 中所有键的数组位置都会被保留，当 X 和 Y intersect 键集合时，值会被 Y 中的值覆盖。带有非中断密钥的 Y 中的元素会被附加，保留其部分顺序。</p>

4.1.171. io.k8s.apimachinery.pkg.api.resource.Quantity schema

描述

quantity 是数字的固定点表示。除了 String () 和 AsInt64 () 访问器外，它还在 JSON 和 YAML 中提供方便的 marshaling/unmarshaling。序列化格式为：

```
<quantity> ::= <signedNumber><suffix>
```

(Note that <suffix> may be empty, from the "" case in <decimalSI>.)

```
<digit> ::= 0 | 1 | ... | 9
<digits> ::= <digit> | <digit><digits>
<number> ::= <digits> | <digits>.<digits> |
<digits>.<digits> <sign> ::= "+" | "-"
<signedNumber> ::= <number> | <sign><number>
<suffix> ::= <binarySI> | <decimalExponent> | <decimalSI> <binarySI> ::= Ki | Mi | Gi | Ti | Pi | Ei
```

(International System of units; See: <http://physics.nist.gov/cuu/Units/binary.html>)

```
<decimalSI> ::= m | "" | k | M | G | T | P | E
```

(Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.)

```
<decimalExponent> ::= "e" <signedNumber> | "E" <signedNumber>
```

无论使用了三种经验，任何数量都不能代表大于 $2^{63}-1$ 的数字，或者可能有超过 3 个十进制位置。更大或更精确的数字将被上限或舍入。（例如：0.1m 将向上舍入为 1m。）如果我们需要更大或较小的数量，则这可能会在以后进行扩展。

当从字符串解析 Quantity 时，它会记住它拥有的后缀类型，并在其序列化时再次使用相同的类型。

在序列化之前，数量将置于"规范形式"。这意味着，Exponent/suffix 将调整或缩减（在 Mantissa 中相应的增加或减少），以便：

- Norecision is lost - 不会发出分数数 - exponent（或后缀）会尽可能大。

除非数字为负数，否则将省略该符号。

示例：

- 1.5 将序列化为 "1500m" - 1.5Gi 将序列化为 "1536Mi"

请注意，数量将在内部表示，由浮点数表示。这是整个练习。

非规范值仍会解析，只要它们被编写，但会以规范形式重新提供。（因此始终使用规范形式，或者不 diff。）

这个格式的目的是在不编写一些特殊处理代码的情况下，很难使用这些数字，而希望这样会使实施器也使用固定点实现。

类型

字符串

4.1.172. io.k8s.apimachinery.pkg.apis.meta.v1.Condition schema

描述

condition 包含此 API 资源当前状态的一个方面的详情。

类型

object

必填

- type
- status
- lastTransitionTime
- reason
- message

模式

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件从一个状态转换到另一个状态最后一次的时间。这应该是底层条件变化的时间。如果为未知，则使用 API 字段更改的时间是可以接受的。
message	字符串	Message 是人类可读的消息，指示有关转换的详细信息。这可以是空字符串。

属性	类型	描述
observedGeneration	整数	observedGeneration 代表 .metadata.generation，这是条件设置所基于的条件。例如，如果 .metadata.generation 目前为 12，但 .status.conditions[x].observedGeneration 是 9，则代表条件与实例的当前状态已不匹配。
reason	字符串	reason 包含程序标识符，指示条件最后一次转换的原因。特定条件类型的制作者可能会定义预期的值和此字段的含义，以及这些值是否被视为有保证的 API。该值应该是 CamelCase 字符串。此字段不能为空。
status	字符串	条件的状态，True, False, Unknown 之一。
type	字符串	CamelCase 或 foo.example.com/CamelCase 中的条件类型。

4.1.173. io.k8s.apimachinery.pkg.apis.meta.v1.DeleteOptions schema

描述

删除 API 对象时，可以提供 DeleteOptions。

类型

对象

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
dryRun	数组（字符串）	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
preconditions	preconditions	在执行删除前必须完成。如果不可能，则返回 409 Conflict 状态。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

4.1.174. io.k8s.apimachinery.pkg.apis.meta.v1.FieldsV1 schema

描述

FieldsV1 以 JSON 格式在诸如 Trie 的数据结构中存储一组字段。

每个键可以是代表字段本身的 '.', 始终映射到一个空集, 也可以是代表子字段或项的字符串。字符串将遵循以下四种格式之一: 'f:<name>', 其中 <name> 是 struct 中的字段名称, 或映射 'v:<value>' 中的键, 其中 <value> 是列表项 'i:<index>' 的确切 json 格式值, 其中 <index> 是列表 'k:<keys>' 中的项位置, 其中 <value> 是列表项 'i:<index>' 的确切 json 格式值。其中 <keys> 是列表项的键字段到其唯一值的映射, 如果键映射到空字段值, 则键代表的字段是集合的一部分。

确切的格式在 iwl.k8s.io/structured-merge-diff 中定义

类型

object

4.1.175. io.k8s.apimachinery.pkg.apis.meta.v1.LabelSelector schema

描述

标签选择器, 即一组资源的标签查询。matchLabels 和 matchExpressions 的逻辑关系是 AND。空标签选择器匹配所有对象。null 标签选择器不匹配任何对象。

类型

对象

模式

属性	类型	描述
matchExpressions	数组 (LabelSelectorRequirement)	matchExpressions 是标签选择器要求列表。要求的逻辑关系是 AND。
matchLabels	对象 (字符串)	matchLabels 是 {key,value} 对的映射。MatchLabels 映射中的单个 {key,value} 等同于 matchExpressions 的一个元素, 其 key 字段是 "key", 运算符是 "In", 值数组仅包含 "value"。要求的逻辑关系是 AND。

4.1.176. io.k8s.apimachinery.pkg.apis.meta.v1.LabelSelectorRequirement schema

描述

标签选择器要求是一个选择器, 其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

模式

属性	类型	描述
key	字符串	key 是选择器应用到的标签键。
operator	字符串	运算符代表一个键与一组值的关系。有效的运算符是 In、NotIn、Exists 和 DoesNotExist。
值	数组（字符串）	值是字符串值的数组。如果运算符是 In 或 NotIn，则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist，则值数组必须为空。这个数组会在策略性合并补丁中被替换。

4.1.177. io.k8s.apimachinery.pkg.apis.meta.v1.ListMeta schema

描述

ListMeta 描述了复合资源必须具有的元数据，包括列表和各种状态对象。资源可能只有一个 {ObjectMeta, ListMeta} 之一。

类型

对象

模式

属性	类型	描述
继续	字符串	如果用户对返回的项目数量设置了限制，则可以设置继续，并且表示服务器有更多可用的数据。该值不透明，可用于向提供此列表的端点发出另一个请求，以检索下一个可用对象集合。如果服务器配置已更改或几分钟以上，则无法继续一致的列表。使用此 continue 值时返回的 resourceVersion 字段将与第一个响应中的值相同，除非您从错误消息收到此令牌。

属性	类型	描述
remainingItemCount	整数	remainingItemCount 是列表中未包含在此列表响应中的后续项目数量。如果列表请求包含标签或字段选择器，则其余项目的数量为未知，且该字段将保留为未设置，并在序列化过程中省略。如果列表完成（因为不是块或因为这是最后一个块），则没有更多剩余项，且此字段将被取消设置，并在序列化过程中省略并省略。早于 v1.15 的服务器不会设置此字段。 remainingItemCount 的预期 使用估计 集合的大小。客户端不应依赖 remainingItemCount 设置或准确。
resourceVersion	字符串	标识此对象的内部版本的字符串，供客户端用来确定对象何时更改。值必须被客户端视为不透明，并将未修改传递回服务器。由系统填充。只读。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
selfLink	string	deprecated: selfLink 是一个旧的只读字段，它不再由系统填充。

4.1.178. io.k8s.apimachinery.pkg.apis.meta.v1.ManagedFieldsEntry schema

描述

ManagedFieldsEntry 是一个 workflow-id、FieldSet 和 fieldset 应用到的资源的组版本。

类型

对象

模式

属性	类型	描述
apiVersion	string	apiVersion 定义此字段应用到的此资源版本。类似于顶级 APIVersion 字段的格式为 "group/version"。需要跟踪字段集的版本，因为它无法自动转换。

属性	类型	描述
fieldsType	string	FieldsType 是不同字段格式和版本的差异。当前只有一个可能的值："FieldsV1"
fieldsV1	FieldsV1	FieldsV1 包含第一个 JSON 版本格式，如 "FieldsV1" 类型中所述。
Manager	string	Manager 是管理这些字段的工作流的标识符。
operation	string	操作是导致创建此 ManagedFieldsEntry 的操作类型。此字段的唯一有效值为 'Apply' 和 'Update'。
subresource	string	subresource 是用来更新该对象的子资源的名称，如果对象通过主资源更新，则为空字符串。此字段的值用于区分管理器，即使它们共享相同的名称。例如，状态更新将与使用相同的管理器名称的常规更新不同。请注意，APIVersion 字段与 Subresource 字段无关，它始终与主资源的版本对应。
time	时间	Time 是添加 ManagedFields 条目的时间戳。添加了字段时也会更新时间戳，管理器会更改任何拥有的字段值或删除字段。当从条目中删除字段时，时间戳不会更新，因为另一个管理器接管了它。

4.1.179. io.k8s.apimachinery.pkg.apis.meta.v1.MicroTime schema

描述

MicroTime 是微秒级精度的时间版本。

类型

字符串

4.1.180. io.k8s.apimachinery.pkg.apis.meta.v1.ObjectMeta schema

描述

ObjectMeta 是所有持久的资源必须具有的元数据，其中包括用户必须创建的所有对象。

类型

对象

模式

属性	类型	描述
annotations	对象 (字符串)	<p>annotations 是一个无结构的键值映射，它存储了一个资源，可通过外部工具设置它来存储和检索任意元数据。它们不可查询，在修改对象时应保留。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations</p>
creationTimestamp	时间	<p>creationTimestamp 是一个时间戳，代表创建此对象时的服务器时间。无法保证在单独的操作之间按顺序设置。客户端可能不会设置这个值。它以 RFC3339 格式表示，并以 UTC 为单位。</p> <p>由系统填充。只读。null 表示列表。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</p>
deletionGracePeriodSeconds	整数	<p>在将这个对象从系统中删除前，允许此对象安全终止的秒数。仅在同设置 deletionTimestamp 时设置。只能缩短。只读。</p>

属性	类型	描述
deletionTimestamp	时间	<p>deletionTimestamp 是 RFC 3339 日期和时间，将删除此资源。当用户请求安全删除时，服务器会设置此字段，且客户端不会直接设置。当 finalizers 列表为空后，资源应该会被删除（不再在资源列表中可见且名称无法从名称访问）。只要终结器列表包含项目，删除就被阻止。设置 deletionTimestamp 后，这个值可能不会取消设置，或者在以后在以后设置，但它可能会被缩短，或者资源可在此时间删除。例如，用户可以请求在 30 秒内删除 pod。Kubelet 将通过向 pod 中的容器发送安全终止信号来响应。30 秒后，Kubelet 将向容器发送一个硬终止信号(SIGKILL)，并在清理后从 API 中删除 pod。在存在网络分区时，此对象可能仍然存在于此时间戳后，直到管理员或自动进程可以确定资源完全被终止为止。如果没有设置，则不请求删除对象。</p> <p>当请求安全删除时，系统填充。只读。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</p>
finalizers	数组（字符串）	<p>必须在从 registry 中删除对象前为空。每个条目都是负责的组件的标识符，该组件将从列表中删除该条目。如果对象的 deletionTimestamp 不是非nil，则此列表中的条目只能被删除。结束程序可能会以任何顺序处理和删除。顺序不会被强制，因为它引入了卡住终结器的风险。finalizers 是一个共享字段，任何具有权限的操作都可能会重新排序它。如果按顺序处理了终结器列表，则这可能会导致在列表后面负责第一个终结器（字段值、外部系统或其他）生成的信号（字段值、外部系统或其他）导致问题。如果没有强制排序终结器，可以自由地订购自己，且不会受到列表中排序更改的影响。</p>

属性	类型	描述
generateName	字符串	<p>generateName 是一个可选前缀，供服务器用来生成唯一的名称。如果没有提供 Name 字段，如果使用此字段，返回到客户端的名称将与传递的名称不同。这个值也会与唯一的后缀合并。提供的值具有与 Name 字段相同的验证规则，可通过在服务器中使值唯一的后缀长度截断。</p> <p>如果指定了此字段且生成的名称存在，服务器将返回 409。</p> <p>仅在未指定 Name 时才应用。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#idempotency</p>
generation	整数	代表所需状态的特定生成状态的序列号。由系统填充。只读。
labels	对象 (字符串)	<p>字符串键和值映射，可用于组织和分类（范围并选择）对象。可以与复制控制器和服务的选择器匹配。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels</p>
managedFields	数组(ManagedFieldsEntry)	managedFields 将 workflow-id 和 version 映射到由该工作流管理的字段集合。这主要用于内部内务，用户通常不需要设置或了解此字段。工作流可以是用户名、控制器的名称，也可以是特定应用路径的名称，如 "ci-cd"。字段集始终位于修改对象时使用的工作流的版本。
name	字符串	<p>名称在命名空间内必须是唯一的。在创建资源时，需要一些资源，但有些资源可能允许客户端自动请求生成适当名称。名称主要用于创建 idempotence 和配置定义。无法更新。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names</p>

属性	类型	描述
namespace	字符串	<p>namespace 定义每个名称必须唯一的空间。空命名空间等同于 "default" 命名空间，但 "default" 是规范表示。并非所有对象都需要限定到命名空间 - 这些对象的此字段的值将为空。</p> <p>必须是 DNS_LABEL。无法更新。 更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces</p>
ownerReferences	数组(OwnerReference)	<p>依赖于此对象的对象列表。如果删除了列表中的所有对象，则此对象将被垃圾回收。如果此对象由控制器管理，则此列表中的条目将指向此控制器，controller 被设置为 true。不能有多个管理控制器。</p>
resourceVersion	字符串	<p>代表此对象的内部版本的 opaque 值，供客户端用来确定对象何时更改。可用于对资源或一组资源进行最佳并发、更改检测和监视操作。客户端必须将这些值视为不透明，并将未修改的回服务器传递。它们可能仅对特定资源或一组资源有效。</p> <p>由系统填充。只读。值必须被客户端和被视为不透明。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency</p>
selfLink	string	<p>deprecated: selfLink 是一个旧的只读字段，它不再由系统填充。</p>
uid	字符串	<p>UID 是此对象的时间和空间值的唯一值。它通常由服务器成功创建资源时生成，不允许更改 PUT 操作。</p> <p>由系统填充。只读。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids</p>

4.1.181. io.k8s.apimachinery.pkg.apis.meta.v1.OwnerReference schema

描述

OwnerReference 包含足够信息，供您识别自己的对象。您自己的对象必须与依赖或集群范围位于同一个命名空间中，因此没有 namespace 字段。

类型

object

必填

- **apiVersion**
- **kind**
- **name**
- **uid**

模式

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
blockOwnerDeletion	布尔值	如果为 true，如果所有者具有 "foregroundDeletion" finalizer，则无法从 key-value 存储中删除所有者，直到此引用被删除为止。有关垃圾收集器如何与此字段交互并强制实施前台删除，请参阅 https://kubernetes.io/docs/concepts/architecture/garbage-collection/#foreground-deletion 。默认为 false。要设置此字段，用户需要"删除"所有者权限，否则将返回 422 (Unprocessable Entity)。
controller	布尔值	如果为 true，则此引用指向管理控制器。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names

属性	类型	描述
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids

4.1.182. io.k8s.apimachinery.pkg.apis.meta.v1.Patch schema

描述

提供了补丁来为 Kubernetes PATCH 请求正文提供具体的名称和类型。

类型

object

4.1.183. io.k8s.apimachinery.pkg.apis.meta.v1.Preconditions schema

描述

在执行操作（更新、删除等）之前，必须满足 preconditions。

类型

对象

模式

属性	类型	描述
resourceVersion	string	指定目标 ResourceVersion
uid	string	指定目标 UID。

4.1.184. io.k8s.apimachinery.pkg.apis.meta.v1.Status schema

描述

status 是不返回其他对象的调用的返回值。

类型

对象

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
code	整数	此状态的建议 HTTP 返回码 0（如果未设置）。
详情	StatusDetails	扩展与原因相关的数据。每个原因都可以定义自己的扩展详情。此字段是可选的，且返回的数据无法保证符合任何模式，除非 reason 类型定义。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
message	字符串	此操作状态的人类可读描述。
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
reason	字符串	此操作处于"Failure"状态的机器可读描述。如果这个值为空，则没有可用的信息。Reason 阐明 HTTP 状态代码，但不会覆盖它。
status	字符串	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

4.1.185.

描述

类型

对象

模式

属性	类型	描述
field	string	可选。
message	string	
reason	string	

4.1.186. io.k8s.apimachinery.pkg.apis.meta.v1.StatusDetails schema

描述

类型

对象

模式

属性	类型	描述
group	string	
kind	string	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	
retryAfterSeconds	整数	
uid	string	

4.1.187. io.k8s.apimachinery.pkg.apis.meta.v1.Time schema

描述

类型

字符串

4.1.188. io.k8s.apimachinery.pkg.apis.meta.v1.WatchEvent schema

描述**类型****object****必填**

- **type**
- 对象

模式

属性	类型	描述
对象		
type	字符串	

4.1.189. io.k8s.apimachinery.pkg.runtime.RawExtension schema**描述**

```
type MyAPIObject struct {
  runtime.TypeMeta `json:",inline"`
  MyPlugin runtime.Object `json:"myPlugin"`
}
```

```
type PluginA struct {
  AOption string `json:"aOption"`
}
```

```
type MyAPIObject struct {
  runtime.TypeMeta `json:",inline"`
  MyPlugin runtime.RawExtension `json:"myPlugin"`
}
```

```
type PluginA struct {
  AOption string `json:"aOption"`
}
```

```
{
  "kind":"MyAPIObject",
  "apiVersion":"v1",
  "myPlugin": {
    "kind":"PluginA",
    "aOption":"foo",
  },
}
```

类型

对象

4.1.190. io.k8s.apimachinery.pkg.util.intstr.IntOrString schema

描述

类型

字符串

4.1.191. io.k8s.kube-aggregator.pkg.apis.apiregistration.v1.APIServiceList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	

4.1.192. io.k8s.migration.v1alpha1.StorageVersionMigrationList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.193. io.k8s.storage.snapshot.v1.VolumeSnapshotClassList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.194. io.k8s.storage.snapshot.v1.VolumeSnapshotContentList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.195. io.k8s.storage.snapshot.v1.VolumeSnapshotList schema

描述

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.196. io.openshift.internal.security.v1.RangeAllocationList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.197. io.openshift.route.v1.RouteList schema

描述

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.198. io.openshift.security.v1.SecurityContextConstraintsList schema

描述

类型

object

必填

- items

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.1.199. io.topolvm.v1.LogicalVolumeList schema

描述

类型

object

必填

- **items**

模式

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
items		更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ListMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

第 5 章

5.1.

5.1.1. CustomResourceDefinition [apiextensions.k8s.io/v1]

描述

类型

对象

5.2. CUSTOMRESOURCEDEFINITION [APIEXTENSIONS.K8S.IO/V1]

描述

类型

object

必填

- spec

5.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	
spec	对象	

属性	类型	描述
status	对象	

5.2.1.1. .spec

描述

类型

object

必填

- **group**
-
- **scope**
-

属性	类型	描述
	对象	
group	字符串	
	对象	
preserveUnknownFields	布尔值	
scope	字符串	
	array	
versions[]	对象	

5.2.1.2. .spec.conversion

描述

类型

object

必填

-

属性	类型	描述
	字符串	
webhook	对象	

5.2.1.3. .spec.conversion.webhook

描述

类型

object

必填

- **conversionReviewVersions**

属性	类型	描述
clientConfig	对象	
conversionReviewVersions	数组（字符串）	

5.2.1.4. .spec.conversion.webhook.clientConfig

描述

类型

object

属性	类型	描述
caBundle	字符串	
service	对象	
url	字符串	

5.2.1.5. .spec.conversion.webhook.clientConfig.service

描述

类型

object

必填

- **namespace**

- name

属性	类型	描述
name	字符串	必填
namespace	字符串	必填
path	字符串	
port	整数	

5.2.1.6. .spec.names

描述

类型

object

必填

-
- kind

属性	类型	描述
	数组（字符串）	
kind	字符串	
listKind	字符串	
	字符串	
shortNames	数组（字符串）	
	字符串	

5.2.1.7. .spec.versions

描述

类型

array

5.2.1.8. .spec.versions[]

描述**类型****object****必填**

- **name**
-
- **storage**

属性	类型	描述
additionalPrinterColumns	array	
additionalPrinterColumns[]	对象	
已弃用	布尔值	默认为false。
deprecationWarning	字符串	
name	字符串	
schema	对象	
	布尔值	
storage	布尔值	
	对象	

5.2.1.9. .spec.versions[].additionalPrinterColumns**描述****类型****array****5.2.1.10. .spec.versions[].additionalPrinterColumns[]****描述****类型****object****必填**

- **name**
- **type**

- jsonPath

属性	类型	描述
description	字符串	
格式	字符串	
jsonPath	字符串	
name	字符串	
priority	整数	
type	字符串	

5.2.1.11. .spec.versions[].schema

描述

类型

object

属性	类型	描述
openAPIV3Schema	JSONSchemaProps	

5.2.1.12. .spec.versions[].subresources

描述

类型

object

属性	类型	描述
scale	对象	
status	对象	

5.2.1.13. .spec.versions[].subresources.scale

描述

类型

object

必填

- `specReplicasPath`
- `statusReplicasPath`

属性	类型	描述
<code>labelSelectorPath</code>	字符串	
<code>specReplicasPath</code>	字符串	
<code>statusReplicasPath</code>	字符串	

5.2.1.14. `..spec.versions[].subresources.status`

描述

类型

对象

5.2.1.15. `..status`

描述

类型

object

属性	类型	描述
<code>acceptedNames</code>	对象	
<code>conditions</code>	array	
<code>conditions[]</code>	对象	
<code>storedVersions</code>	数组（字符串）	

5.2.1.16. `..status.acceptedNames`

描述

类型

object

必填

-
- `kind`

属性	类型	描述
	数组 (字符串)	
kind	字符串	
listKind	字符串	
	字符串	
shortNames	数组 (字符串)	
	字符串	

5.2.1.17. .status.conditions

描述

类型

array

5.2.1.18. .status.conditions[]

描述

类型

object

必填

- type
- status

属性	类型	描述
lastTransitionTime		
message	字符串	
reason	字符串	
status	字符串	可以是 True, False, Unknown。
type	字符串	

5.2.2. API 端点

可用的 API 端点如下：

- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions**
 -
 -
 -
- **/apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions**
 -
- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}**
 -
 -
 -
 -
- **/apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions/{name}**
 -
- **/apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status**
 -
 -
 -

5.2.2.1. /apis/apiextensions.k8s.io/v1/customresourcedefinitions

表 5.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

表 5.2. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	默认为任何内容。
gracePeriodSeconds	整数	
labelSelector	字符串	默认为任何内容。
limit	整数	
orphanDependents	布尔值	
propagationPolicy	字符串	
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	默认为未设置
sendInitialEvents	布尔值	
timeoutSeconds	整数	

表 5.3. 主体参数

参数	类型	描述
正文 (body)		

表 5.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	
401 - Unauthorized	空

HTTP 方法

GET

描述

表 5.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 5.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinitionList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 CustomResourceDefinition

表 5.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 5.8. 主体参数

参数	类型	描述
正文 (body)	CustomResourceDefinition 模式	

表 5.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
201 - Created	CustomResourceDefinition 模式
202 - Accepted	CustomResourceDefinition 模式
401 - Unauthorized	空

5.2.2.2. /apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions

表 5.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CustomResourceDefinition 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 5.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

5.2.2.3. /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}

表 5.12. 全局路径参数

参数	类型	描述
name	字符串	CustomResourceDefinition 的名称

表 5.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CustomResourceDefinition

表 5.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 5.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 5.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 CustomResourceDefinition

表 5.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 CustomResourceDefinition

表 5.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 5.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 5.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
201 - Created	CustomResourceDefinition 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 CustomResourceDefinition

表 5.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 5.22. 主体参数

参数	类型	描述
正文 (body)	CustomResourceDefinition 模式	

表 5.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
201 - Created	CustomResourceDefinition 模式
401 - Unauthorized	空

5.2.2.4. /apis/apiextensions.k8s.io/v1/watch/customresourcedefinitions/{name}

表 5.24. 全局路径参数

参数	类型	描述
name	字符串	CustomResourceDefinition 的名称

表 5.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind CustomResourceDefinition. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 5.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

5.2.2.5. /apis/apiextensions.k8s.io/v1/customresourcedefinitions/{name}/status

表 5.27. 全局路径参数

参数	类型	描述
name	字符串	CustomResourceDefinition 的名称

表 5.28. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定的 CustomResourceDefinition 的读取状态

表 5.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 CustomResourceDefinition 的部分更新状态

表 5.30. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 5.31. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 5.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
201 - Created	CustomResourceDefinition 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 CustomResourceDefinition 的状态

表 5.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 5.34. 主体参数

参数	类型	描述
正文 (body)	CustomResourceDefinition 模式	

表 5.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	CustomResourceDefinition 模式
201 - Created	CustomResourceDefinition 模式
401 - Unauthorized	空

第 6 章 REGISTRATION APIS

6.1. API 注册 API

6.1.1. APIService [apiregistration.k8s.io/v1]

描述

APIService 代表特定 GroupVersion 的服务器。名称必须是 "version.group"。

类型

对象

6.2. APISERVICE [APIREGISTRATION.K8S.IO/V1]

描述

APIService 代表特定 GroupVersion 的服务器。名称必须是 "version.group"。

类型

对象

6.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
spec	对象	APIServiceSpec 包含用于查找和与服务器通信的信息。仅支持 https，但您可以禁用证书验证。
status	对象	APIServiceStatus 包含有关 API 服务器的派生信息

6.2.1.1. .spec

描述

APIServiceSpec 包含用于查找和与服务器通信的信息。仅支持 https，但您可以禁用证书验证。

类型

object

必填

- **groupPriorityMinimum**
- **versionPriority**

属性	类型	描述
caBundle	字符串	cabundle 是一个 PEM 编码的 CA 捆绑包，用于验证 API 服务器的服务证书。如果未指定，则使用 apiserver 中的系统信任 root。
group	字符串	group 是此服务器主机的 API 组名称
groupPriorityMinimum	整数	GroupPriorityMinimum 是这个组应至少具有的优先级。优先级更高的意味着客户端优先于较低优先级的组。请注意，此组的其它版本可能会指定更高的 GroupPriorityMinimum 值，以便整个组获得更高的优先级。主要排序基于 GroupPriorityMinimum，按最高数字排序为最低值(10 之前的 20 个)。辅助排序基于对象名称的字母顺序比较。(v1.foo 之前的 v1.bar)我们推荐使用如下内容： *.k8s.io（扩展除外）18000 和 PaaS (OpenShift、Deis)在 2000s 中。

属性	类型	描述
insecureSkipTLSVerify	布尔值	insecureSkipTLSVerify 在与此服务器通信时禁用 TLS 证书验证。强烈建议您这样做。您应该使用 CABundle。
service	对象	ServiceReference 包含对 Service.legacy.k8s.io 的引用
version	字符串	version 是这个服务器主机的 API 版本。例如："v1"
versionPriority	整数	VersionPriority 控制其组内此 API 版本的顺序。必须大于零。主要排序基于 VersionPriority，最高排序为最低(10 之前的 20 个)。由于它是组内部，因此数字可能比较小，可能在 10s 中。如果优先级相等，则版本字符串将用于计算组内的顺序。如果版本字符串是"kube-like"，它将按字典顺序排序在非"kube 类似"版本字符串上。"kube-like"版本以"v"开头，后跟一个数字（主版本），然后可以选择字符串"alpha"或"beta"，另一个数字（次版本）。它们首先按 GA > beta > alpha 排序（其中 GA 是没有后缀（如 beta 或 alpha）的版本，然后比较主版本，然后是次版本。版本排序列表示例：v10, v2, v1, v11beta2, v10beta3, v3beta1, v12alpha1, v11alpha2, foo1, foo10。

6.2.1.2. .spec.service

描述

ServiceReference 包含对 Service.legacy.k8s.io 的引用

类型

object

属性	类型	描述
name	字符串	name 是服务的名称
namespace	字符串	namespace 是服务的命名空间

属性	类型	描述
port	整数	如果指定，托管 Webhook 的服务上的端口。默认为 443，以便向后兼容。 端口 应该是有效的端口号 (1-65535，包含)。

6.2.1.3. .status

描述

APIServiceStatus 包含有关 API 服务器的派生信息

类型

object

属性	类型	描述
conditions	array	apiService 的当前服务状态。
conditions[]	对象	APIServiceCondition 描述了 APIService 在特定点的状态

6.2.1.4. .status.conditions

描述

apiService 的当前服务状态。

类型

array

6.2.1.5. .status.conditions[]

描述

APIServiceCondition 描述了 APIService 在特定点的状态

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。

属性	类型	描述
message	字符串	人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	条件最后一次转换的唯一、一词、CamelCase 原因。
status	字符串	status 是条件的状态。可以是 True, False, Unknown。
type	字符串	type 是条件的类型。

6.2.2. API 端点

可用的 API 端点如下：

- **/apis/apiregistration.k8s.io/v1/apiservices**
 - **DELETE** : 删除 APIService 的集合
 - **GET**: 列出或监视类型为 APIService 的对象
 - **POST** : 创建 APIService
- **/apis/apiregistration.k8s.io/v1/watch/apiservices**
 - **GET**: 观察单个对 APIService 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apiregistration.k8s.io/v1/apiservices/{name}**
 - **DELETE** : 删除 APIService
 - **GET** : 读取指定的 APIService
 - **PATCH** : 部分更新指定的 APIService
 - **PUT** : 替换指定的 APIService
- **/apis/apiregistration.k8s.io/v1/watch/apiservices/{name}**
 - **GET**: 观察对 APIService 类型的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/apiregistration.k8s.io/v1/apiservices/{name}/status**
 - **GET** : 指定 APIService 的读取状态
 - **PATCH** : 部分更新指定 APIService 的状态
 - **PUT** : 替换指定 APIService 的状态

6.2.2.1. /apis/apiregistration.k8s.io/v1/apiservices

表 6.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 APIService 集合

表 6.2. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 6.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 6.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 APIService 的对象

表 6.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 6.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIServiceList 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 APIService

表 6.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 6.8. 主体参数

参数	类型	描述
正文 (body)	APIService 模式	

表 6.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式

HTTP 代码	响应正文
201 - Created	APIService 模式
202 - Accepted	APIService 模式
401 - Unauthorized	空

6.2.2.2. /apis/apiregistration.k8s.io/v1/watch/apiservices

表 6.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 APIService 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 6.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

6.2.2.3. /apis/apiregistration.k8s.io/v1/apiservices/{name}

表 6.12. 全局路径参数

参数	类型	描述
name	字符串	APIService 的名称

表 6.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 APIService

表 6.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 6.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 6.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 APIService

表 6.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 APIService

表 6.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 6.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 6.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
201 - Created	APIService 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 APIService

表 6.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 6.22. 主体参数

参数	类型	描述
正文 (body)	APIService 模式	

表 6.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
201 - Created	APIService 模式
401 - Unauthorized	空

6.2.2.4. /apis/apiregistration.k8s.io/v1/watch/apiservices/{name}

表 6.24. 全局路径参数

参数	类型	描述
name	字符串	APIService 的名称

表 6.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind APIService 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 6.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

6.2.2.5. /apis/apiregistration.k8s.io/v1/apiservices/{name}/status

表 6.27. 全局路径参数

参数	类型	描述
name	字符串	APIService 的名称

表 6.28. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 APIService 的读取状态

表 6.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 APIService 的部分更新状态

表 6.30. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 6.31. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 6.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
201 - Created	APIService 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 APIService 的状态

表 6.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 6.34. 主体参数

参数	类型	描述
正文 (body)	APIService 模式	

表 6.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	APIService 模式
201 - Created	APIService 模式
401 - Unauthorized	空

第 7 章 APPS API

7.1. APPS API

7.1.1. ControllerRevision [apps/v1]

描述

ControllerRevision 实现状态数据的不可变快照。客户端负责对包含其内部状态的对象进行序列化和停用。成功创建 ControllerRevision 后，无法更新它。API 服务器将失败，验证尝试修改 Data 字段的所有请求。但是，控制器修订可能会被删除。请注意，由于 DaemonSet 和 StatefulSet 控制器同时使用它进行更新和回滚，所以这个对象是 beta。但是，未来版本中可能会存在名称和表示的变化，并且客户端不应依赖于其稳定性。它主要用于控制器的内部使用。

类型

object

7.1.2. DaemonSet [apps/v1]

描述

DaemonSet 代表守护进程集的配置。

类型

object

7.1.3. Deployment [apps/v1]

描述

Deployment 为 Pod 和 ReplicaSet 启用声明更新。

类型

object

7.1.4. ReplicaSet [apps/v1]

描述

ReplicaSet 确保指定数量的 pod 副本在任意给定时间运行。

类型

object

7.1.5. StatefulSet [apps/v1]

描述

StatefulSet 代表一组具有一致身份的 pod。身份被定义为：
- 网络：单个稳定的 DNS 和主机名。
- Storage: 作为请求的多个 VolumeClaims。
StatefulSet 确保给定网络身份始终映射到同一存储身份。

类型

对象

7.2. CONTROLLERREVISION [APPS/V1]

描述

ControllerRevision 实现状态数据的不可变快照。客户端负责对包含其内部状态的对象进行序列化和停用。成功创建 ControllerRevision 后，无法更新它。API 服务器将失败，验证尝试修改 Data 字段的所有请求。但是，控制器修订可能会被删除。请注意，由于 DaemonSet 和 StatefulSet 控制器同时使用它进行更新和回滚，所以这个对象是 beta。但是，未来版本中可能会存在名称和表示的变化，并且客户端不应依赖于其稳定性。它主要用于控制器的内部使用。

类型

object

必填

- **revision**

7.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
data	RawExtension	数据是状态的序列化表示。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
revision	整数	revision 表示数据代表的状态的修订。

7.2.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/controllerrevisions**
 - **GET**: 列出或监视类型为 ControllerRevision 的对象
- **/apis/apps/v1/watch/controllerrevisions**
 - **GET**: 观察单个对 ControllerRevision 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/controllerrevisions**
 - **DELETE** : 删除 ControllerRevision 的集合
 - **GET**: 列出或监视类型为 ControllerRevision 的对象
 - **POST** : 创建 ControllerRevision
- **/apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions**
 - **GET**: 观察单个对 ControllerRevision 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/controllerrevisions/{name}**
 - **DELETE** : 删除 ControllerRevision
 - **GET** : 读取指定的 ControllerRevision
 - **PATCH** : 部分更新指定的 ControllerRevision
 - **PUT** : 替换指定的 ControllerRevision
- **/apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions/{name}**
 - **GET**: 观察对类型 ControllerRevision. deprecated 的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

7.2.2.1. /apis/apps/v1/controllerrevisions

表 7.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 ControllerRevision 的对象

表 7.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	ControllerRevisionList schema
401 - Unauthorized	空

7.2.2.2. /apis/apps/v1/watch/controllerrevisions

表 7.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ControllerRevision 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.2.2.3. /apis/apps/v1/namespaces/{namespace}/controllerrevisions

表 7.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ControllerRevision 的集合

表 7.7. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 7.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ControllerRevision 的对象

表 7.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 7.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	ControllerRevisionList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 ControllerRevision

表 7.12. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.13. 主体参数

参数	类型	描述
正文 (body)	ControllerRevision 模式	

表 7.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	ControllerRevision 模式
201 - Created	ControllerRevision 模式
202 - Accepted	ControllerRevision 模式
401 - Unauthorized	空

7.2.2.4. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions

表 7.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ControllerRevision 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.2.2.5. /apis/apps/v1/namespaces/{namespace}/controllerrevisions/{name}

表 7.18. 全局路径参数

参数	类型	描述
name	字符串	ControllerRevision 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ControllerRevision

表 7.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 7.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 7.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 `ControllerRevision`

表 7.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ControllerRevision</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `ControllerRevision`

表 7.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	ControllerRevision 模式
201 - Created	ControllerRevision 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ControllerRevision

表 7.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.28. 主体参数

参数	类型	描述
正文 (body)	ControllerRevision 模式	

表 7.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	ControllerRevision 模式
201 - Created	ControllerRevision 模式
401 - Unauthorized	空

7.2.2.6. /apis/apps/v1/watch/namespaces/{namespace}/controllerrevisions/{name}

表 7.30. 全局路径参数

参数	类型	描述
name	字符串	ControllerRevision 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind ControllerRevision. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 7.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.3. DAEMONSET [APPS/V1]

描述

DaemonSet 代表守护进程集的配置。

类型

对象

7.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	DaemonSetSpec 是守护进程集的规格。
status	object	DaemonSetStatus 代表守护进程集的当前状态。

7.3.1.1. .spec

描述

DaemonSetSpec 是守护进程集的规格。

类型

object

必填

- selector
- 模板

属性	类型	描述
minReadySeconds	整数	新创建的 DaemonSet pod 应该就绪的最小秒数，且没有其容器崩溃，以便它被视为可用。默认为 0 (pod 在就绪后将被视为可用)。
revisionHistoryLimit	整数	要保留的旧历史记录数量，以允许回滚。这是区分显式零而非指定的指针。默认值为 10。
selector	labelSelector	对由守护进程集管理的 pod 的标签查询。必须匹配才能控制。它必须与 pod 模板的标签匹配。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
模板	PodTemplateSpec	描述要创建的 pod 的对象。DaemonSet 将在每个节点上创建与模板的节点选择器匹配的一个副本（如果未指定节点选择器，则在每个节点上）。唯一允许的 <code>template.spec.restartPolicy</code> 值是 "Always"。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template
updateStrategy	object	DaemonSetUpdateStrategy 是一个结构，用于控制 DaemonSet 的更新策略。

7.3.1.2. .spec.updateStrategy

描述

DaemonSetUpdateStrategy 是一个结构，用于控制 DaemonSet 的更新策略。

类型

object

属性	类型	描述
rollingUpdate	object	控制守护进程设置滚动更新所需的行为的 spec。

属性	类型	描述
type	string	<p>守护进程集更新的类型。可以是 "RollingUpdate" 或 "OnDelete"。默认为 RollingUpdate。</p> <p>可能枚举值： - "OnDelete" 仅在终止时替换旧守护进程 - "RollingUpdate" 使用滚动更新来将旧守护进程替换为新的守护进程，即替换每个节点中的它们。</p>

7.3.1.3. .spec.updateStrategy.rollingUpdate

描述

控制守护进程设置滚动更新所需的的行为的 spec。

类型

object

属性	类型	描述
maxSurge	IntOrString	<p>具有现有可用 DaemonSet pod 的最大节点数，这些 pod 在更新过程中可能会具有更新的 DaemonSet pod。值可以是绝对数量（例如：5）或所需 pod 的百分比（例如：10%）。如果 MaxUnavailable 为 0，则不能是 0。绝对数从百分比计算，方法是舍入到最小 1。默认值为 0。示例：当它被设置为 30% 时，在运行守护进程 pod 的节点总数最多 30%（如 status.desiredNumberScheduled）可以在旧 pod 标记为已删除前创建它的新 pod。更新首先在 30% 的节点上启动新 pod。当一个更新的 pod 可用(Ready 代表至少 minReadySeconds)后，该节点上的旧 DaemonSet pod 将被标记为已删除。如果旧的 pod 因任何原因(Ready 转换为 false)不可用，或者被驱除或排空，则更新的 pod 会在那个节点上立即创建，而无需考虑激增限制。允许激增意味着，如果就绪度检查失败，则任何给定节点上 daemonset 所消耗的资源可能会加倍，因此资源密集型 daemonset 应该考虑在中断期间可能会导致驱除。</p>

属性	类型	描述
maxUnavailable	IntOrString	更新过程中可以无法使用的 DaemonSet pod 的最大数量。值可以是绝对数量（例如：5）或处于更新开始时的 DaemonSet pod 总数的百分比（例如 10%）。绝对数通过向上舍入来从百分比计算。如果 MaxSurge 为 0 默认值，则不能是 0。示例：当它被设置为 30% 时，在运行守护进程 pod 的节点总数最多 30%（如 status.desiredNumberScheduled）可以在任何给定时间停止其 pod。更新首先停止这些 DaemonSet pod 的 30%，然后启动新的 DaemonSet pod。新 pod 可用后，它会继续其他 DaemonSet pod，从而确保更新过程中始终都提供至少 70% 的原始 DaemonSet pod 数量。

7.3.1.4. .status

描述

DaemonSetStatus 代表守护进程集的当前状态。

类型

object

必填

- **currentNumberScheduled**
- **numberMisscheduled**
- **desiredNumberScheduled**
- **numberReady**

属性	类型	描述
collisionCount	整数	DaemonSet 的哈希冲突计数。当需要为最新的 ControllerRevision 创建名称时，DaemonSet 控制器使用此字段作为冲突避免机制。
conditions	数组	代表 DaemonSet 的当前状态的最新可用影响。

属性	类型	描述
conditions[]	object	DaemonSetCondition 描述了 DaemonSet 在特定点的状态。
currentNumberScheduled	整数	运行至少 1 个守护进程 pod 的节点数量，应该运行守护进程 pod。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
desiredNumberScheduled	整数	应该运行守护进程 pod 的节点总数（包括正确运行守护进程 pod 的节点）。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
numberAvailable	整数	应运行守护进程 pod 的节点数量，并有一个或多个守护进程 pod 正在运行并可用（至少为 spec.minReadySeconds）
numberMisscheduled	整数	运行守护进程 pod 的节点数量，但不运行守护进程 pod。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
numberReady	整数	numberReady 是应该运行守护进程 pod 的节点数量，且一个或多个守护进程 pod 使用 Ready Condition 运行。
numberUnavailable	整数	应运行守护进程 pod 的节点数量，且没有守护进程 pod 正在运行并可用（至少为 spec.minReadySeconds）
observedGeneration	整数	守护进程设置控制器观察到的最近生成。
updatedNumberScheduled	整数	运行更新的守护进程 pod 的节点总数

7.3.1.5. .status.conditions

描述

代表 DaemonSet 的当前状态的最新可用影响。

类型

array

7.3.1.6. .status.conditions[]**描述**

DaemonSetCondition 描述了 DaemonSet 在特定点的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	string	DaemonSet 条件的类型。

7.3.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/daemonsets**
 - **GET**: 列出或监视类型为 DaemonSet 的对象
- **/apis/apps/v1/watch/daemonsets**
 - **GET**: 观察单个对 DaemonSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/daemonsets**
 - **DELETE** : 删除 DaemonSet 的集合
 - **GET**: 列出或监视类型为 DaemonSet 的对象

- **POST** : 创建 DaemonSet
- **/apis/apps/v1/watch/namespaces/{namespace}/daemonsets**
 - **GET**: 观察单个对 DaemonSet 列表的更改。已弃用 : 改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}**
 - **DELETE** : 删除 DaemonSet
 - **GET**: 读取指定的 DaemonSet
 - **PATCH** : 部分更新指定的 DaemonSet
 - **PUT** : 替换指定的 DaemonSet
- **/apis/apps/v1/watch/namespaces/{namespace}/daemonsets/{name}**
 - **GET**: 观察对类型为 DaemonSet 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status**
 - **GET**: 指定 DaemonSet 的读取状态
 - **PATCH** : 部分更新指定 DaemonSet 的状态
 - **PUT** : 替换指定 DaemonSet 的状态

7.3.2.1. /apis/apps/v1/daemonsets

表 7.33. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签, 由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回, 也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视, 则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 DaemonSet 的对象

表 7.34. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSetList schema
401 - 未授权	空

7.3.2.2. /apis/apps/v1/watch/daemonsets

表 7.35. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 DaemonSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.36. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.3.2.3. /apis/apps/v1/namespaces/{namespace}/daemonsets

表 7.37. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.38. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 DaemonSet 的集合

表 7.39. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下：</p> <ul style="list-style-type: none"> - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 7.40. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 DaemonSet 的对象

表 7.42. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 7.43. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSetList schema
401 - 未授权	空

HTTP 方法

POST

描述

创建 DaemonSet

表 7.44. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.45. 主体参数

参数	类型	描述
正文 (body)	DaemonSet 模式	

表 7.46. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
201 - Created	DaemonSet 模式
202 - Accepted	DaemonSet 模式
401 - 未授权	空

7.3.2.4. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets

表 7.47. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.48. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 DaemonSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.3.2.5. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}

表 7.50. 全局路径参数

参数	类型	描述
name	string	DaemonSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.51. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 DaemonSet

表 7.52. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 7.53. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.54. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 DaemonSet

表 7.55. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 DaemonSet

表 7.56. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.57. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.58. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
201 - Created	DaemonSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 DaemonSet

表 7.59. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.60. 主体参数

参数	类型	描述
正文 (body)	DaemonSet 模式	

表 7.61. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
201 - Created	DaemonSet 模式
401 - 未授权	空

7.3.2.6. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets/{name}

表 7.62. 全局路径参数

参数	类型	描述
name	string	DaemonSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.63. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 DaemonSet 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 7.64. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.3.2.7. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status

表 7.65. 全局路径参数

参数	类型	描述
name	string	DaemonSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.66. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 DaemonSet 的读取状态

表 7.67. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 DaemonSet 的部分更新状态

表 7.68. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为： <ul style="list-style-type: none"> - Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.69. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.70. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
201 - Created	DaemonSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 DaemonSet 的状态

表 7.71. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为： <ul style="list-style-type: none"> - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.72. 主体参数

参数	类型	描述
正文 (body)	DaemonSet 模式	

表 7.73. HTTP 响应

HTTP 代码	响应正文
200 - OK	DaemonSet 模式
201 - Created	DaemonSet 模式
401 - 未授权	空

7.4. DEPLOYMENT [APPS/V1]

描述

Deployment 为 Pod 和 ReplicaSet 启用声明更新。

类型

对象

7.4.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	DeploymentSpec 是部署所需的行为的规格。
status	object	DeploymentStatus 是部署的最新观察状态。

7.4.1.1. .spec

描述

DeploymentSpec 是部署所需的行为的规格。

类型

object

必填

- selector
- 模板

属性	类型	描述
----	----	----

属性	类型	描述
minReadySeconds	整数	新创建的 pod 应该就绪的最少秒数，且没有其容器崩溃，以便它被视为可用。默认为 0 (pod 在就绪后将被视为可用)
paused	布尔值	表示部署已暂停。
progressDeadlineSeconds	整数	部署在被视为失败前进行进度的最长时间（以秒为单位）。部署控制器将继续处理失败的部署，且具有 ProgressDeadlineExceeded 原因的条件将出现在部署状态中。请注意，部署暂停期间不会估算进度。默认值为 600s。
replicas	整数	所需 pod 的数量。这是区分显式零而非指定的指针。默认为 1。
revisionHistoryLimit	整数	要保留的旧 ReplicaSet 的数量，以允许回滚。这是区分显式零而非指定的指针。默认值为 10。
selector	labelSelector	pod 的标签选择器。Pod 被选择为受此部署影响的现有 ReplicaSet。它必须与 pod 模板的标签匹配。
策略	object	DeploymentStrategy 描述如何将现有 pod 替换为新 pod。
模板	PodTemplateSpec	template 描述了要创建的 pod。唯一允许的 template.spec.restartPolicy 值是 "Always"。

7.4.1.2. .spec.strategy

描述

DeploymentStrategy 描述如何将现有 pod 替换为新 pod。

类型

object

属性	类型	描述
rollingUpdate	object	spec 来控制滚动更新所需的行为。

属性	类型	描述
type	字符串	<p>部署类型。可以是 "Recreate" 或 "RollingUpdate"。默认为 RollingUpdate。</p> <p>可能枚举值： - "Recreate" Kill all existing pods before new pods. - "RollingUpdate" 使用滚动更新来替换旧的 ReplicaSet。例如，逐步缩减旧的 ReplicaSet 并扩展新的 pod。</p>

7.4.1.3. .spec.strategy.rollingUpdate

描述

spec 来控制滚动更新所需的行为。

类型

object

属性	类型	描述
maxSurge	IntOrString	<p>可以在所需 pod 数量之上调度的最大 pod 数量。值可以是绝对数量（例如：5）或所需 pod 的百分比（例如：10%）。如果 MaxUnavailable 为 0，则不能是 0。绝对数通过向上舍入来从百分比计算。默认值为 25%。示例：当它被设置为 30% 时，可在滚动更新启动时立即扩展新的 ReplicaSet，以便旧 pod 和新 pod 的总数不超过所需 pod 的 130%。旧 pod 被终止后，可以进一步扩展新的 ReplicaSet，确保更新期间随时运行的 pod 总数是所需 pod 的 130%。</p>

属性	类型	描述
maxUnavailable	IntOrString	更新过程中可以无法使用的最大 pod 数量。值可以是绝对数量（例如：5）或所需 pod 的百分比（例如：10%）。绝对数通过向下舍入来从百分比计算。如果 MaxSurge 为 0，则不能是 0。默认值为 25%。示例：当它被设置为 30% 时，在滚动更新启动时，旧的 ReplicaSet 可以缩减到所需 pod 的 70%。新 pod 就绪后，可以进一步缩减旧的 ReplicaSet，然后扩展新的 ReplicaSet，确保更新期间始终可用的 pod 总数至少为 70% 的所需 pod。

7.4.1.4. .status

描述

DeploymentStatus 是部署的最新观察状态。

类型

object

属性	类型	描述
availableReplicas	整数	此部署的目标可用 pod 的总数（至少 minReadySeconds）目标。
collisionCount	整数	部署哈希冲突计数。当需要为最新 ReplicaSet 创建名称时，Deployment 控制器使用此字段作为冲突避免机制。
conditions	数组	代表部署当前状态的最新可用影响。
conditions[]	object	DeploymentCondition 描述了在某个时间点上部署的状态。
observedGeneration	整数	部署控制器观察到的生成。
readyReplicas	整数	readyReplicas 是此 Deployment 的 pod 数量，并带有 Ready Condition。
replicas	整数	此部署目标的非终止 pod 的总数（标签与选择器匹配）。

属性	类型	描述
unavailableReplicas	整数	此部署目标不可用 pod 的总数。这是部署仍然需要具有 100% 可用容量的 pod 的总数。它们可能是正在运行但还没有创建的 pod。
updatedReplicas	整数	此部署的目标非终止的 pod 总数，具有所需模板规格。

7.4.1.5. .status.conditions

描述

代表部署当前状态的最新可用影响。

类型

array

7.4.1.6. .status.conditions[]

描述

DeploymentCondition 描述了在某个时间点上部署的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。
lastUpdateTime	时间	最后一次更新此条件的时间。
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态， True, False, Unknown 之一。
type	string	部署条件的类型。

7.4.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/deployments**
 - **GET**: 列出或监视类型为 Deployment 的对象
- **/apis/apps/v1/watch/deployments**
 - **GET**: 观察单个对 Deployment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/deployments**
 - **DELETE** : 删除部署的集合
 - **GET**: 列出或监视类型为 Deployment 的对象
 - **POST** : 创建部署
- **/apis/apps/v1/watch/namespaces/{namespace}/deployments**
 - **GET**: 观察单个对 Deployment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}**
 - **DELETE** : 删除部署
 - **GET** : 读取指定的 Deployment
 - **PATCH** : 部分更新指定的部署
 - **PUT** : 替换指定的部署
- **/apis/apps/v1/watch/namespaces/{namespace}/deployments/{name}**
 - **GET**: 观察对类型为 Deployment 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status**
 - **GET**: 指定 Deployment 的读取状态
 - **PATCH** : 部分更新指定部署的状态
 - **PUT** : 替换指定部署的状态

7.4.2.1. /apis/apps/v1/deployments

表 7.74. 全局查询参数

参数	类型	描述
----	----	----

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Deployment 的对象

表 7.75. HTTP 响应

HTTP 代码	响应正文
200 - OK	DeploymentList 模式
401 - 未授权	空

7.4.2.2. /apis/apps/v1/watch/deployments

表 7.76. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Deployment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.77. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.4.2.3. /apis/apps/v1/namespaces/{namespace}/deployments

表 7.78. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.79. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE**描述**

删除部署集合

表 7.80. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 7.81. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.82. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Deployment 的对象

表 7.83. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 7.84. HTTP 响应

HTTP 代码	响应正文
200 - OK	DeploymentList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建部署

表 7.85. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.86. 主体参数

参数	类型	描述
正文 (body)	Deployment 模式	

表 7.87. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
201 - Created	Deployment 模式
202 - Accepted	Deployment 模式
401 - 未授权	空

7.4.2.4. /apis/apps/v1/watch/namespaces/{namespace}/deployments

表 7.88. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.89. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Deployment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.90. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.4.2.5. /apis/apps/v1/namespaces/{namespace}/deployments/{name}

表 7.91. 全局路径参数

参数	类型	描述
name	string	部署的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.92. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除部署

表 7.93. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 7.94. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 7.95. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的部署

表 7.96. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Deployment</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的部署

表 7.97. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.98. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.99. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
201 - Created	Deployment 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 Deployment

表 7.100. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.101. 主体参数

参数	类型	描述
正文 (body)	Deployment 模式	

表 7.102. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
201 - Created	Deployment 模式
401 - 未授权	空

7.4.2.6. /apis/apps/v1/watch/namespaces/{namespace}/deployments/{name}

表 7.103. 全局路径参数

参数	类型	描述
name	string	部署的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.104. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Deployment. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 7.105. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.4.2.7. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status

表 7.106. 全局路径参数

参数	类型	描述
name	string	部署的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.107. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 Deployment 的读取状态

表 7.108. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定部署的部分更新状态

表 7.109. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.110. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.111. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
201 - Created	Deployment 模式
401 - 未授权	空

HTTP 方法**PUT****描述**

替换指定部署的状态

表 7.112. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.113. 主体参数

参数	类型	描述
正文 (body)	Deployment 模式	

表 7.114. HTTP 响应

HTTP 代码	响应正文
200 - OK	Deployment 模式
201 - Created	Deployment 模式
401 - 未授权	空

7.5. REPLICASET [APPS/V1]

描述

ReplicaSet 确保指定数量的 pod 副本在任意给定时间运行。

类型

对象

7.5.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	如果 ReplicaSet 的 Labels 为空，则默认为与 ReplicaSet 管理的 Pod 相同。标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	ReplicaSetSpec 是 ReplicaSet 的规格。
status	object	ReplicaSetStatus 代表 ReplicaSet 的当前状态。

7.5.1.1. .spec

描述

ReplicaSetSpec 是 ReplicaSet 的规格。

类型

object

必填

- **selector**

属性	类型	描述
minReadySeconds	整数	新创建的 pod 应该就绪的最少秒数，且没有其容器崩溃，以便它被视为可用。默认为 0 (pod 在就绪后将被视为可用)
replicas	整数	replicas 是所需副本数量。这是区分显式零和未指定的指针。默认为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/#what-is-a-replicationcontroller
selector	labelSelector	selector 是 pod 的标签查询，应与副本数匹配。标记必须匹配的键和值，以便由此副本集控制。它必须与 pod 模板的标签匹配。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
模板	PodTemplateSpec	template 是描述在检测到副本不足时创建的 pod 的对象。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template

7.5.1.2. .status

描述

ReplicaSetStatus 代表 ReplicaSet 的当前状态。

类型

object

必填

- replicas

属性	类型	描述
availableReplicas	整数	此副本集的可用副本数（至少为 minReadySeconds）。
conditions	数组	代表副本集的当前状态的最新可用影响。

属性	类型	描述
conditions[]	object	ReplicaSetCondition 描述了特定点上副本集的状态。
fullyLabeledReplicas	整数	具有与 replicaset 的 pod 模板标签匹配的标签的 pod 数量。
observedGeneration	整数	observedGeneration 反映了最近观察到的 ReplicaSet 的生成。
readyReplicas	整数	readyReplicas 是此 ReplicaSet 的目标带有 Ready Condition 的 pod 数量。
replicas	整数	replicas 是最近观察到的副本数。 更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/#what-is-a-replicationcontroller

7.5.1.3. .status.conditions

描述

代表副本集的当前状态的最新可用影响。

类型

array

7.5.1.4. .status.conditions[]

描述

ReplicaSetCondition 描述了特定点上副本集的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。

属性	类型	描述
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	string	副本集条件的类型。

7.5.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/replicasets**
 - **GET**: 列出或监视类型为 ReplicaSet 的对象
- **/apis/apps/v1/watch/replicasets**
 - **GET**: 观察单个对 ReplicaSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/replicasets**
 - **DELETE** : 删除 ReplicaSet 的集合
 - **GET**: 列出或监视类型为 ReplicaSet 的对象
 - **POST** : 创建 ReplicaSet
- **/apis/apps/v1/watch/namespaces/{namespace}/replicasets**
 - **GET**: 观察单个对 ReplicaSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}**
 - **DELETE** : 删除 ReplicaSet
 - **GET**: 读取指定的 ReplicaSet
 - **PATCH** : 部分更新指定的 ReplicaSet
 - **PUT** : 替换指定的 ReplicaSet
- **/apis/apps/v1/watch/namespaces/{namespace}/replicasets/{name}**
 - **GET**: 观察对类型为 ReplicaSet 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}/status**
 - **GET**: 指定 ReplicaSet 的读取状态

- **PATCH** : 部分更新指定 ReplicaSet 的状态
- **PUT** : 替换指定 ReplicaSet 的状态

7.5.2.1. /apis/apps/v1/replicasets

表 7.115. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 ReplicaSet 的对象

表 7.116. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSetList 模式
401 - 未授权	空

7.5.2.2. /apis/apps/v1/watch/replicasets

表 7.117. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ReplicaSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.118. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.5.2.3. /apis/apps/v1/namespaces/{namespace}/replicasets

表 7.119. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.120. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ReplicaSet 的集合

表 7.121. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 7.122. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.123. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ReplicaSet 的对象

表 7.124. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 7.125. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSetList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建 ReplicaSet

表 7.126. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.127. 主体参数

参数	类型	描述
正文 (body)	ReplicaSet 模式	

表 7.128. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
201 - Created	ReplicaSet 模式
202 - Accepted	ReplicaSet 模式
401 - 未授权	空

7.5.2.4. /apis/apps/v1/watch/namespaces/{namespace}/replicasets

表 7.129. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.130. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ReplicaSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.131. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.5.2.5. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}

表 7.132. 全局路径参数

参数	类型	描述
name	string	ReplicaSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.133. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ReplicaSet

表 7.134. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 7.135. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 7.136. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 ReplicaSet

表 7.137. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ReplicaSet</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 ReplicaSet

表 7.138. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.139. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.140. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
201 - Created	ReplicaSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 ReplicaSet

表 7.141. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.142. 主体参数

参数	类型	描述
正文 (body)	ReplicaSet 模式	

表 7.143. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
201 - Created	ReplicaSet 模式
401 - 未授权	空

7.5.2.6. /apis/apps/v1/watch/namespaces/{namespace}/replicasets/{name}

表 7.144. 全局路径参数

参数	类型	描述
name	string	ReplicaSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.145. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind ReplicaSet 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 7.146. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.5.2.7. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}/status

表 7.147. 全局路径参数

参数	类型	描述
name	string	ReplicaSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.148. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 ReplicaSet 的读取状态

表 7.149. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 ReplicaSet 的部分更新状态

表 7.150. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.151. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.152. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
201 - Created	ReplicaSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 ReplicaSet 的状态

表 7.153. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.154. 主体参数

参数	类型	描述
正文 (body)	ReplicaSet 模式	

表 7.155. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicaSet 模式
201 - Created	ReplicaSet 模式
401 - 未授权	空

7.6. STATEFULSET [APPS/V1]

描述

StatefulSet 代表一组具有一致身份的 pod。身份被定义为：- 网络：单个稳定的 DNS 和主机名。- Storage: 作为请求的多个 VolumeClaims。
StatefulSet 确保给定网络身份始终映射到同一存储身份。

类型

对象

7.6.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	StatefulSetSpec 是 StatefulSet 的规格。
status	object	StatefulSetStatus 代表 StatefulSet 的当前状态。

7.6.1.1. .spec

描述

StatefulSetSpec 是 StatefulSet 的规格。

类型

object

必填

- **selector**
- **模板**
- **serviceName**

属性	类型	描述
minReadySeconds	整数	新创建的 pod 应该就绪的最少秒数，且没有其任何容器崩溃才被视为可用。默认为 0 (pod 在就绪后将被视为可用)
ordinals	object	StatefulSetOrdinals 描述了此 StatefulSet 中副本或内部分配的策略。
persistentVolumeClaimRetentionPolicy	object	StatefulSetPersistentVolumeClaimRetentionPolicy 描述了从 StatefulSet VolumeClaimTemplates 创建的 PVC 的策略。
podManagementPolicy	string	<p>podManagementPolicy 控制在初始扩展、替换节点上的 pod 或缩减时创建 pod。默认策略是 OrderedReady，其中 pod 会以递增顺序(pod-0、pod-1 等)创建，控制器将等到每个 pod 就绪后再继续。缩减时，pod 会根据相反的顺序删除。替代的策略是 Parallel，它将并行创建 pod，以匹配所需的规模，而缩减则一次性删除所有 pod。</p> <p>可能枚举值：-</p> <ul style="list-style-type: none"> "OrderedReady" 将在扩展时严格增加 pod，并严格减少缩减顺序，仅当上一个 pod 就绪或终止时进行。大多数 pod 可以随时更改。 "Parallel" 会在有状态设置副本数改变后立即创建和删除 pod，且不会等待 pod 就绪或完成终止。
replicas	整数	replicas 是给定模板的所需副本数。这些是相同模板的实例化的副本，但单个副本也具有一致的身份。如果未指定，则默认为 1。
revisionHistoryLimit	整数	revisionHistoryLimit 是 StatefulSet 的修订历史记录中维护的最大修订版本数。修订历史记录由当前应用的 StatefulSetSpec 版本表示的所有修订版本组成。默认值为 10。

属性	类型	描述
selector	labelSelector	selector 是 pod 的标签查询，应与副本数匹配。它必须与 pod 模板的标签匹配。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
serviceName	string	serviceName 是管理此 StatefulSet 的服务名称。此服务必须在 StatefulSet 之前存在，并负责集合的网络身份。Pod 获取遵循特征的 DNS/hostnames: pod-specific-string.serviceName.default.svc.cluster.local，其中 "pod-specific-string" 由 StatefulSet 控制器管理。
模板	PodTemplateSpec	template 是描述在检测到副本不足时创建的 pod 的对象。StatefulSet 标记的每个 pod 都履行此模板，但具有来自 StatefulSet 的其余部分的唯一身份。每个 pod 将以 <statefulsetname>-<podindex> 格式命名。例如，名为 "web" 的 StatefulSet 中的 pod，索引号为 "3" 将命名为 "web-3"。唯一允许的 template.spec.restartPolicy 值是 "Always"。
updateStrategy	object	StatefulSetUpdateStrategy 表示 StatefulSet 控制器用来执行更新的策略。它包括为指定策略执行更新所需的任何其他参数。
volumeClaimTemplates	数组(PersistentVolumeClaim)	VolumeClaimTemplates 是一个声明列表，允许 pod 可被引用。StatefulSet 控制器负责映射网络身份以维护 pod 身份的方式进行声明。此列表中的每个声明必须在模板的一个容器中至少有一个匹配（按名称）volumeMount。此列表中的声明优先于模板中的任何卷，其名称相同。

7.6.1.2. .spec.ordinals

描述

StatefulSetOrdinals 描述了此 StatefulSet 中副本或内部分配的策略。

类型

object

属性	类型	描述
start	整数	start 是代表第一个副本索引的编号。它可用于通过默认的 0 索引名称来对备用索引（例如：1索引）中的副本数，或编配从一个 StatefulSet 到另一个 StatefulSet 的正向移动。如果设置，副本索引将位于范围： <code>[.spec.ordinals.start, .spec.ordinals.start + .spec.replicas)</code> 。如果未设置，则默认为 0。副本索引将位于范围： <code>[0, .spec.replicas)</code> 。

7.6.1.3. `.spec.persistentVolumeClaimRetentionPolicy`

描述

`StatefulSetPersistentVolumeClaimRetentionPolicy` 描述了从 `StatefulSet VolumeClaimTemplates` 创建的 PVC 的策略。

类型

object

属性	类型	描述
whenDeleted	string	WhenDeleted 指定在 <code>StatefulSet</code> 被删除时从 <code>StatefulSet VolumeClaimTemplates</code> 创建的 PVC 会发生什么。 Retain 的默认策略会导致 PVC 不受 <code>StatefulSet</code> 删除的影响。 Delete 策略会导致这些 PVC 被删除。
whenScaled	string	WhenScaled 指定在 <code>StatefulSet</code> 缩减时从 <code>StatefulSet VolumeClaimTemplates</code> 创建的 PVC 会发生什么。 Retain 的默认策略会导致 PVC 不受缩减的影响。 Delete 策略会导致删除副本数以上的任何超额 PVC。

7.6.1.4. `.spec.updateStrategy`

描述

`StatefulSetUpdateStrategy` 表示 `StatefulSet` 控制器用来执行更新的策略。它包括为指定策略执行更新所需的任何其他参数。

类型

object

属性	类型	描述
rollingUpdate	object	RollingUpdateStatefulSetStrategy 用于为 RollingUpdateStatefulSetStrategyType 通信参数。
type	string	<p>type 表示 StatefulSetUpdateStrategy 的类型。默认为 RollingUpdate。</p> <p>可能枚举值： - "OnDelete" 会触发旧行为。禁用版本跟踪和排序滚动重启。当 Pod 被删除时，会从 StatefulSetSpec 重新创建 Pod。当使用此策略执行扩展操作时，StatefulSet 的 currentRevision 表示的特定版本。</p> <p>- "RollingUpdate" 表示更新将应用到 StatefulSet 排序约束的所有 Pod。当使用此策略执行扩展操作时，会从 StatefulSet 的 updateRevision 指示的规格版本创建新的 Pod。</p>

7.6.1.5. .spec.updateStrategy.rollingUpdate**描述**

RollingUpdateStatefulSetStrategy 用于为 RollingUpdateStatefulSetStrategyType 通信参数。

类型**object**

属性	类型	描述
maxUnavailable	IntOrString	<p>更新过程中可以无法使用的最大 pod 数量。值可以是绝对数量（例如：5）或所需 pod 的百分比（例如：10%）。绝对数通过向上舍入来从百分比计算。不能是 0。默认为 1。此字段是 alpha-level，仅适用于启用 MaxUnavailableStatefulSet 功能的服务器。该字段适用于范围 0 到 Replicas-1 范围内的所有 pod。这意味着，如果 0 到 Replicas-1 范围内的不可用 pod，它将计为 MaxUnavailable。</p>

属性	类型	描述
分区	整数	partition 表示 StatefulSet 应该在其中对更新进行分区。在滚动更新过程中，所有来自 ordinal Replicas-1 到分区的 pod 都会被更新。来自 ordinal Partition-1 到 0 的所有 pod 保持不变。这有助于执行基于 canary 的部署。默认值为 0。

7.6.1.6. .status

描述

StatefulSetStatus 代表 StatefulSet 的当前状态。

类型

object

必填

- replicas

属性	类型	描述
availableReplicas	整数	此 statefulset 的目标可用 pod 的总数（至少 minReadySeconds）目标为 statefulset。
collisionCount	整数	conflictCount 是 StatefulSet 的哈希冲突计数。StatefulSet 控制器使用此字段作为冲突避免机制，当它需要为最新的 ControllerRevision 创建名称时。
conditions	数组	代表 statefulset 的当前状态的最新可用影响。
conditions[]	object	StatefulSetCondition 描述了 statefulset 在特定点的状态。
currentReplicas	整数	currentReplicas 是 StatefulSet 控制器从 currentRevision 指示的 StatefulSet 版本创建的 Pod 数量。

属性	类型	描述
currentRevision	string	currentRevision（如果不是空的），指示用于在 [0,currentReplicas) 序列中用于生成 Pod 的 StatefulSet 版本。
observedGeneration	整数	observedGeneration 是此 StatefulSet 观察到的最新生成。它与 StatefulSet 的生成对应，该生成由 API 服务器在变异时进行更新。
readyReplicas	整数	readyReplicas 是为此 StatefulSet 创建的 pod 数量，并带有 Ready Condition。
replicas	整数	replicas 是 StatefulSet 控制器创建的 Pod 数量。
updateRevision	string	updateRevision（如果不是空的），指示用于在 [replicas-updatedReplicas,replicas) 序列中生成 Pod 的 StatefulSet 版本。
updatedReplicas	整数	updatedReplicas 是 StatefulSet 控制器从 updateRevision 指示的 StatefulSet 版本创建的 Pod 数量。

7.6.1.7. .status.conditions

描述

代表 statefulset 的当前状态的最新可用影响。

类型

array

7.6.1.8. .status.conditions[]

描述

StatefulSetCondition 描述了 statefulset 在特定点的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	string	statefulset 条件的类型。

7.6.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/statefulsets**
 - **GET**: 列出或监视类型为 StatefulSet 的对象
- **/apis/apps/v1/watch/statefulsets**
 - **GET**: 观察单个对 StatefulSet 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/statefulsets**
 - **DELETE**：删除 StatefulSet 的集合
 - **GET**: 列出或监视类型为 StatefulSet 的对象
 - **POST**：创建 StatefulSet
- **/apis/apps/v1/watch/namespaces/{namespace}/statefulsets**
 - **GET**: 观察单个对 StatefulSet 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}**
 - **DELETE**：删除 StatefulSet
 - **GET**：读取指定的 StatefulSet
 - **PATCH**：部分更新指定的 StatefulSet
 - **PUT**：替换指定的 StatefulSet
- **/apis/apps/v1/watch/namespaces/{namespace}/statefulsets/{name}**
 - **GET**: 观察对类型为 StatefulSet 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status**

- **GET**: 指定 StatefulSet 的读取状态
- **PATCH** : 部分更新指定 StatefulSet 的状态
- **PUT** : 替换指定 StatefulSet 的状态

7.6.2.1. /apis/apps/v1/statefulsets

表 7.156. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 StatefulSet 的对象

表 7.157. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSetList 模式
401 - 未授权	空

7.6.2.2. /apis/apps/v1/watch/statefulsets

表 7.158. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 StatefulSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.159. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.6.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets

表 7.160. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.161. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除 StatefulSet 集合

表 7.162. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 7.163. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 7.164. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 StatefulSet 的对象

表 7.165. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 7.166. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSetList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建 StatefulSet

表 7.167. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.168. 主体参数

参数	类型	描述
正文 (body)	StatefulSet 模式	

表 7.169. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
201 - Created	StatefulSet 模式
202 - Accepted	StatefulSet 模式
401 - 未授权	空

7.6.2.4. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets

表 7.170. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 7.171. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 StatefulSet 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 7.172. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.6.2.5. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}

表 7.173. 全局路径参数

参数	类型	描述
name	string	StatefulSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.174. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 StatefulSet

表 7.175. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 7.176. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 7.177. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 StatefulSet

表 7.178. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>StatefulSet</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 StatefulSet

表 7.179. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.180. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.181. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
201 - Created	StatefulSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 StatefulSet

表 7.182. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.183. 主体参数

参数	类型	描述
正文 (body)	StatefulSet 模式	

表 7.184. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
201 - Created	StatefulSet 模式
401 - 未授权	空

7.6.2.6. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets/{name}

表 7.185. 全局路径参数

参数	类型	描述
name	string	StatefulSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.186. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind StatefulSet.deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 7.187. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

7.6.2.7. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status

表 7.188. 全局路径参数

参数	类型	描述
name	string	StatefulSet 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 7.189. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 StatefulSet 的读取状态

表 7.190. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 StatefulSet 的部分更新状态

表 7.191. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 7.192. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 7.193. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
201 - Created	StatefulSet 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 StatefulSet 的状态

表 7.194. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 7.195. 主体参数

参数	类型	描述
正文 (body)	StatefulSet 模式	

表 7.196. HTTP 响应

HTTP 代码	响应正文
200 - OK	StatefulSet 模式
201 - Created	StatefulSet 模式
401 - 未授权	空

第 8 章 身份验证 API

8.1. 身份验证 API

8.1.1. TokenRequest [authentication.k8s.io/v1]

描述

TokenRequest 为给定服务帐户请求一个令牌。

类型

对象

8.1.2. TokenReview [authentication.k8s.io/v1]

描述

TokenReview 会尝试向一个已知用户验证令牌。备注：TokenReview 请求可能会被 kube-apiserver 中的 Webhook 令牌验证器插件缓存。

类型

对象

8.2. TOKENREQUEST [AUTHENTICATION.K8S.IO/V1]

描述

TokenRequest 为给定服务帐户请求一个令牌。

类型

object

必填

- spec

8.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	TokenRequestSpec 包括由客户端提供的令牌请求的参数。
status	object	TokenRequestStatus 是令牌请求的结果。

8.2.1.1. .spec

描述

TokenRequestSpec 包括由客户端提供的令牌请求的参数。

类型

object

必填

- **audiences**

属性	类型	描述
audiences	数组（字符串）	audiences 是令牌的受众对象。令牌的接收者必须使用令牌的 audiences 列表中的一个标识符来识别自己，否则应拒绝令牌。为多个 audiences 发布的令牌可以用于针对列出的 audiences 进行验证，但意味着目标 audiences 之间有高的信任度。
boundObjectRef	object	BoundObjectReference 是对令牌绑定到的一个对象的引用。

属性	类型	描述
expirationSeconds	整数	ExpirationSeconds 是请求的有效期。令牌签发者可能会返回具有不同有效期持续时间的令牌，因此客户端需要检查响应中的 'expiration' 字段。

8.2.1.2. .spec.boundObjectRef

描述

BoundObjectReference 是对令牌绑定到的一个对象的引用。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
kind	字符串	引用的类型。有效类型为 'Pod' 和 'Secret'。
name	字符串	引用的名称。
uid	字符串	引用的 UID。

8.2.1.3. .status

描述

TokenRequestStatus 是令牌请求的结果。

类型

object

必填

- **token**
- **expirationTimestamp**

属性	类型	描述
expirationTimestamp	时间	ExpirationTimestamp 是返回令牌的过期时间。
token	字符串	令牌是 opaque bearer 令牌。

8.2.2. API 端点

可用的 API 端点如下：

- **/api/v1/namespaces/{namespace}/serviceaccounts/{name}/token**
 - **POST** : 创建 ServiceAccount 令牌

8.2.2.1. /api/v1/namespaces/{namespace}/serviceaccounts/{name}/token

表 8.1. 全局路径参数

参数	类型	描述
name	字符串	TokenRequest 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 8.2. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建 ServiceAccount 的令牌

表 8.3. 主体参数

参数	类型	描述
正文 (body)	TokenRequest schema	

表 8.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	TokenRequest schema
201 - Created	TokenRequest schema
202 - Accepted	TokenRequest schema
401 - Unauthorized	空

8.3. TOKENREVIEW [AUTHENTICATION.K8S.IO/V1]

描述

TokenReview 会尝试向一个已知用户验证令牌。备注：TokenReview 请求可能会被 kube-apiserver 中的 Webhook 令牌验证器插件缓存。

类型

object

必填

- **spec**

8.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	TokenReviewSpec 是令牌身份验证请求的描述。
status	object	TokenReviewStatus 是令牌身份验证请求的结果。

8.3.1.1. .spec

描述

TokenReviewSpec 是令牌身份验证请求的描述。

类型

object

属性	类型	描述
audiences	数组（字符串）	Audiences 是资源服务器通过令牌标识的标识符的列表。支持 Audiences 的令牌验证器将验证至少针对此列表中的一个 audience 的令牌。如果没有提供 audiences，则 audiences 将默认为 Kubernetes apiserver 的 audiences。
token	字符串	令牌是 opaque bearer 令牌。

8.3.1.2. .status

描述

TokenReviewStatus 是令牌身份验证请求的结果。

类型

object

属性	类型	描述
audiences	数组（字符串）	audiences 是由与 TokenReview 和 token 兼容的验证器选择的 audiences 标识符。标识符是 TokenReviewSpec audiences 的交集和令牌的受众中的任何标识符。设置 spec.audiences 字段的 TokenReview API 客户端应该验证 status.audiences 字段中是否返回兼容的受众标识符，以确保 TokenReview 服务器可被受众了解。如果 TokenReview 返回了一个空 status.audience 字段，其中 status.authenticated 为 "true"，则令牌对 Kubernetes API 服务器的受众有效。
authenticated	布尔值	已验证表示令牌已与一个已知的用户关联。
错误	字符串	Error 表示无法检查令牌
user	object	userinfo 包含有关实现 user.Info 接口所需的用户的信息。

8.3.1.3. .status.user**描述**

userinfo 包含有关实现 user.Info 接口所需的用户的信息。

类型**object**

属性	类型	描述
extra	object	验证器提供的任何额外信息。
extra{}	数组（字符串）	
groups	数组（字符串）	此用户所属的组的名称。
uid	字符串	一次标识此用户的唯一值。如果删除了此用户，并且添加了相同名称的另一个用户，则它们将具有不同的 UID。

属性	类型	描述
username	字符串	在所有活跃用户间唯一标识此用户的名称。

8.3.1.4. .status.user.extra

描述

验证器提供的任何额外信息。

类型

object

8.3.2. API 端点

可用的 API 端点如下：

- **/apis/authentication.k8s.io/v1/tokenreviews**
 - **POST** : 创建一个 TokenReview

8.3.2.1. /apis/authentication.k8s.io/v1/tokenreviews

表 8.5. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**POST****描述**

创建 TokenReview

表 8.6. 主体参数

参数	类型	描述
正文 (body)	TokenReview 模式	

表 8.7. HTTP 响应

HTTP 代码	响应正文
200 - OK	TokenReview 模式
201 - Created	TokenReview 模式
202 - Accepted	TokenReview 模式
401 - Unauthorized	空

第 9 章 授权 API

9.1. 授权 API

9.1.1. LocalSubjectAccessReview [authorization.k8s.io/v1]

描述

LocalSubjectAccessReview 检查一个用户或组是否可以在给定命名空间中执行操作。使用命名空间范围资源可以更容易授予包含权限检查的命名空间范围策略。

类型

对象

9.1.2. SelfSubjectAccessReview [authorization.k8s.io/v1]

描述

SelfSubjectAccessReview 检查当前用户是否可以执行一个操作。不填写 spec.namespace 表示"在所有命名空间中"。Self 是一个特殊的情况，因为用户应始终可以检查自己是否可以执行操作

类型

object

9.1.3. SelfSubjectRulesReview [authorization.k8s.io/v1]

描述

SelfSubjectRulesReview 枚举了当前用户可以在一个命名空间内执行的操作集合。返回的操作列表可能并不完整，具体取决于服务器的授权模式，以及在评估过程中出现的任何错误。UI 应使用 SelfSubjectRulesReview 来显示/隐藏操作，或者快速让最终用户有关其权限的原因。它不应该被外部系统用来驱动授权决策，因为这会导致缓存生命周期/撤销、正确性的混淆。SubjectAccessReview 和 LocalAccessReview 是将授权决策延迟到 API 服务器的正确方法。

类型

object

9.1.4. SubjectAccessReview [authorization.k8s.io/v1]

描述

SubjectAccessReview 检查一个用户或组是否可以执行一个操作。

类型

对象

9.2. LOCALSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

描述

LocalSubjectAccessReview 检查一个用户或组是否可以在给定命名空间中执行操作。使用命名空间范围资源可以更容易授予包含权限检查的命名空间范围策略。

类型

object

必填

- spec

9.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	SubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一
status	object	SubjectAccessReviewStatus

9.2.1.1. .spec

描述

SubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一

类型

object

属性	类型	描述
extra	object	额外对应于验证器中的 <code>user.Info.GetExtra ()</code> 方法。由于这是授权者的输入，因此这里需要反映出来。
extra{}	数组（字符串）	
groups	数组（字符串）	组是您正在测试的组。
nonResourceAttributes	object	<code>nonResourceAttributes</code> 包括可用于对 Authorizer 接口的非资源请求的授权属性
resourceAttributes	object	<code>resourceAttributes</code> 包括可用于向 Authorizer 接口的资源请求的授权属性
uid	字符串	有关请求用户的 UID 信息。
user	字符串	用户是您正在测试的用户。如果您指定了 "User" 而不是 "Groups"，那么它将解释为"如果用户不是任何组的成员"

9.2.1.2. .spec.extra

描述

额外对应于验证器中的 `user.Info.GetExtra ()` 方法。由于这是授权者的输入，因此这里需要反映出来。

类型

object

9.2.1.3. .spec.nonResourceAttributes

描述

`nonResourceAttributes` 包括可用于对 Authorizer 接口的非资源请求的授权属性

类型

object

属性	类型	描述
path	字符串	<code>path</code> 是请求的 URL 路径

属性	类型	描述
verb	字符串	verb 是标准 HTTP 动词

9.2.1.4. .spec.resourceAttributes

描述

resourceAttributes 包括可用于向 Authorizer 接口的资源请求的授权属性

类型

object

属性	类型	描述
group	字符串	group 是资源的 API 组。"*"表示全部。
name	字符串	name 是为"get"请求的资源的名 称，或删除"delete"。"" (empty)表 示 all。
namespace	字符串	namespace 是所请求操作的命名 空间。目前，对于集群范围的资源 "" (empty)，没有命名空间和所有 命名空间"" (空)为 LocalSubjectAccessReviews "" (empty)为空，代表来自 SubjectAccessReview 或 SelfSubjectAccessReview 的命名 空间范围资源"all"
resource	字符串	Resource 是现有资源类型之一。 "*"表示全部。
subresource	字符串	子资源是现有资源类型之一。""表 示 none。
verb	字符串	verb 是一个 kubernetes 资源 API 动词，如 get, list, watch, create, update, delete, proxy. "*"。
version	字符串	Version 是资源的 API 版本。"*"表 示全部。

9.2.1.5. .status

描述

SubjectAccessReviewStatus

类型

object

必填

- **allowed**

属性	类型	描述
allowed	布尔值	Allowed 是必需的。True 代表操作被允许，false 代表不允许。
已拒绝	布尔值	Denied 是可选的。如果被拒绝，则为 True，否则为 false。如果允许都为 false，并且拒绝为 false，则授权者在是否授权该操作时没有建议。如果允许为 true，则拒绝的可能不是 true。
evaluationError	字符串	EvaluationError 表示在授权检查过程中发生一些错误。完全有可能出错，并且能够继续确定其授权状态。例如，RBAC 可以缺少角色，但仍存在足够的角色并绑定到请求的原因。
reason	字符串	reason 是可选的。它表示为什么允许或拒绝请求。

9.2.2. API 端点

可用的 API 端点如下：

- **/apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessreviews**
 - **POST:** 创建一个 LocalSubjectAccessReview

9.2.2.1. /apis/authorization.k8s.io/v1/namespaces/{namespace}/localsubjectaccessreviews

表 9.1. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 9.2. 全局查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建 LocalSubjectAccessReview

表 9.3. 主体参数

参数	类型	描述
正文 (body)	LocalSubjectAccessReview schema	

表 9.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	LocalSubjectAccessReview schema
201 - Created	LocalSubjectAccessReview schema
202 - Accepted	LocalSubjectAccessReview schema

HTTP 代码	响应正文
401 - Unauthorized	空

9.3. SELFSUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

描述

SelfSubjectAccessReview 检查当前用户是否可以执行一个操作。不填写 `spec.namespace` 表示"在所有命名空间中"。Self 是一个特殊的情况，因为用户应始终可以检查自己是否可以执行操作

类型

object

必填

- `spec`

9.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
spec	object	SelfSubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一
status	object	SubjectAccessReviewStatus

9.3.1.1. .spec

描述

SelfSubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一

类型

object

属性	类型	描述
nonResourceAttributes	object	nonResourceAttributes 包括可用于对 Authorizer 接口的非资源请求的授权属性
resourceAttributes	object	resourceAttributes 包括可用于向 Authorizer 接口的资源请求的授权属性

9.3.1.2. .spec.nonResourceAttributes

描述

nonResourceAttributes 包括可用于对 Authorizer 接口的非资源请求的授权属性

类型

object

属性	类型	描述
path	字符串	path 是请求的 URL 路径
verb	字符串	verb 是标准 HTTP 动词

9.3.1.3. .spec.resourceAttributes

描述

resourceAttributes 包括可用于向 Authorizer 接口的资源请求的授权属性

类型

object

属性	类型	描述
group	字符串	group 是资源的 API 组。"*"表示全部。
name	字符串	name 是为"get"请求的资源的名称，或删除"delete"。"" (empty)表示 all。
namespace	字符串	namespace 是所请求操作的命名空间。目前，对于集群范围的资源 "" (empty)，没有命名空间和所有命名空间"" (空) 为 LocalSubjectAccessReviews "" (empty)为空，代表来自 SubjectAccessReview 或 SelfSubjectAccessReview 的命名空间范围资源"all"
resource	字符串	Resource 是现有资源类型之一。"*"表示全部。
subresource	字符串	子资源是现有资源类型之一。""表示 none。
verb	字符串	verb 是一个 kubernetes 资源 API 动词，如 get, list, watch, create, update, delete, proxy. "*"。
version	字符串	Version 是资源的 API 版本。"*"表示全部。

9.3.1.4. .status

描述

SubjectAccessReviewStatus

类型

object

必填

- **allowed**

属性	类型	描述
allowed	布尔值	Allowed 是必需的。True 代表操作被允许，false 代表不允许。
已拒绝	布尔值	Denied 是可选的。如果被拒绝，则为 True，否则为 false。如果允许都为 false，并且拒绝为 false，则授权者在是否授权该操作时没有建议。如果允许为 true，则拒绝的可能不是 true。
evaluationError	字符串	EvaluationError 表示在授权检查过程中发生一些错误。完全有可能出错，并且能够继续确定其授权状态。例如，RBAC 可以缺少角色，但仍存在足够的角色并绑定到请求的原因。
reason	字符串	reason 是可选的。它表示为什么允许或拒绝请求。

9.3.2. API 端点

可用的 API 端点如下：

- **/apis/authorization.k8s.io/v1/selfsubjectaccessreviews**
 - **POST** : 创建一个 SelfSubjectAccessReview

9.3.2.1. /apis/authorization.k8s.io/v1/selfsubjectaccessreviews

表 9.5. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建一个 SelfSubjectAccessReview

表 9.6. 主体参数

参数	类型	描述
正文 (body)	SelfSubjectAccessReview 模式	

表 9.7. HTTP 响应

HTTP 代码	响应正文
200 - OK	SelfSubjectAccessReview 模式
201 - Created	SelfSubjectAccessReview 模式
202 - Accepted	SelfSubjectAccessReview 模式
401 - Unauthorized	空

9.4. SELFSUBJECTRULESREVIEW [AUTHORIZATION.K8S.IO/V1]

描述

SelfSubjectRulesReview 枚举了当前用户可以在一个命名空间内执行的操作集合。返回的操作列表可能并不完整，具体取决于服务器的授权模式，以及在评估过程中出现的任何错误。UI 应使用 SelfSubjectRulesReview 来显示/隐藏操作，或者快速让最终用户有关其权限的原因。它不应该被外部

系统用来驱动授权决策，因为这会导致缓存生命周期/撤销、正确性的混淆 SubjectAccessReview 和 LocalAccessReview 是将授权决策延迟到 API 服务器的正确方法。

类型

object

必填

- **spec**

9.4.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	SelfSubjectRulesReviewSpec 定义 SelfSubjectRulesReview 的规格。
status	object	SubjectRulesReviewStatus 包含规则检查的结果。根据服务器配置的授权程序集以及评估过程中遇到的任何错误，此检查可能不完整。由于授权规则是 additive，因此如果规则出现在列表中，它可以安全地假定主题具有该权限，即使该列表不完整。

9.4.1.1. .spec

描述

SelfSubjectRulesReviewSpec 定义 SelfSubjectRulesReview 的规格。

类型

object

属性	类型	描述
namespace	字符串	用于评估规则的命名空间。必需。

9.4.1.2. .status

描述

SubjectRulesReviewStatus 包含规则检查的结果。根据服务器配置的授权程序集以及评估过程中遇到的任何错误，此检查可能不完整。由于授权规则是 additive，因此如果规则出现在列表中，它可以安全地假定主题具有该权限，即使该列表不完整。

类型

object

必填

- **resourceRules**
- **nonResourceRules**
- **incomplete**

属性	类型	描述
evaluationError	字符串	EvaluationError 可以与 Rules 结合使用。它表示在规则评估过程中出现错误，如不支持规则评估的授权器，并且 ResourceRules 和/或 NonResourceRules 可能不完整。
incomplete	布尔值	当此调用返回的规则不完整时，不完整。当授权者（如外部授权器）不支持规则评估时，这最常遇到。
nonResourceRules	array	nonResourceRules 是主题允许在非资源上执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。
nonResourceRules[]	object	NonResourceRule 包含描述非资源规则的信息

属性	类型	描述
resourceRules	array	resourceRules 是主题允许对资源执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。
resourceRules[]	object	ResourceRule 是主题允许对资源执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。

9.4.1.3. .status.nonResourceRules

描述

nonResourceRules 是主题允许在非资源上执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。

类型

array

9.4.1.4. .status.nonResourceRules[]

描述

NonResourceRule 包含描述非资源规则的信息

类型

object

必填

- verbs

属性	类型	描述
nonResourceURLs	数组（字符串）	NonResourceURLs 是用户应有权访问的一组部分 url。允许 s，但仅作为路径中的最后一步。"*" 表示全部。
verbs	数组（字符串）	verb 是 kubernetes 非资源 API 动词列表，如 get, post, put, delete, patch, head, options。"*" 表示 all。

9.4.1.5. .status.resourceRules

描述

`resourceRules` 是主题允许对资源执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。

类型

array

9.4.1.6. `.status.resourceRules[]`

描述

`ResourceRule` 是主题允许对资源执行的操作列表。列表排序不重要，可能包含重复项，可能不完整。

类型

object

必填

- **verbs**

属性	类型	描述
apiGroups	数组（字符串）	<code>APIGroups</code> 是包含资源的 <code>APIGroup</code> 的名称。如果指定了多个 <code>API</code> 组，则允许针对任何 <code>API</code> 组中的一个枚举的资源请求任何操作。"*"表示全部。
resourceNames	数组（字符串）	<code>resourceNames</code> 是一个可选的规则应用到的名称白名单。空集表示允许一切。"*"表示全部。
资源	数组（字符串）	<code>resources</code> 是此规则应用到的资源列表。"表示指定的 <code>apiGroups</code> 。"/foo" 中的所有资源都代表指定 <code>apiGroups</code> 中所有资源的子资源 'foo'。
verbs	数组（字符串）	<code>verb</code> 是 <code>kubernetes</code> 资源 <code>API</code> 动词列表，例如： <code>get</code> 、 <code>list</code> 、 <code>watch</code> 、 <code>create</code> 、 <code>update</code> 、 <code>delete</code> 、 <code>proxy</code> 。"*"。

9.4.2. API 端点

可用的 API 端点如下：

- `/apis/authorization.k8s.io/v1/selfsubjectrulesreviews`
 - **POST** : 创建一个 `SelfSubjectRulesReview`

9.4.2.1. `/apis/authorization.k8s.io/v1/selfsubjectrulesreviews`

表 9.8. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建一个 SelfSubjectRulesReview

表 9.9. 主体参数

参数	类型	描述
正文 (body)	SelfSubjectRulesReview 模式	

表 9.10. HTTP 响应

HTTP 代码	响应正文
200 - OK	SelfSubjectRulesReview 模式
201 - Created	SelfSubjectRulesReview 模式
202 - Accepted	SelfSubjectRulesReview 模式

HTTP 代码	响应正文
401 - Unauthorized	空

9.5. SUBJECTACCESSREVIEW [AUTHORIZATION.K8S.IO/V1]

描述

SubjectAccessReview 检查一个用户或组是否可以执行一个操作。

类型

object

必填

- **spec**

9.5.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准列表元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
spec	object	SubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一
status	object	SubjectAccessReviewStatus

9.5.1.1. .spec

描述

SubjectAccessReviewSpec 是访问请求的描述。必须设置 ResourceAuthorizationAttributes 和 NonResourceAuthorizationAttributes 之一

类型

object

属性	类型	描述
extra	object	额外对应于验证器中的 user.Info.GetExtra () 方法。由于这是授权者的输入，因此这里需要反映出来。
extra{}	数组 (字符串)	
groups	数组 (字符串)	组是您正在测试的组。
nonResourceAttributes	object	nonResourceAttributes 包括可用于对 Authorizer 接口的非资源请求的授权属性
resourceAttributes	object	resourceAttributes 包括可用于向 Authorizer 接口的资源请求的授权属性
uid	字符串	有关请求用户的 UID 信息。
user	字符串	用户是您正在测试的用户。如果您指定了 "User" 而不是 "Groups", 那么它将解释为"如果用户不是任何组的成员"

9.5.1.2. .spec.extra

描述

额外对应于验证器中的 `user.Info.GetExtra ()` 方法。由于这是授权者的输入，因此这里需要反映出来。

类型

object

9.5.1.3. .spec.nonResourceAttributes

描述

`nonResourceAttributes` 包括可用于对 Authorizer 接口的非资源请求的授权属性

类型

object

属性	类型	描述
path	字符串	path 是请求的 URL 路径
verb	字符串	verb 是标准 HTTP 动词

9.5.1.4. .spec.resourceAttributes

描述

`resourceAttributes` 包括可用于向 Authorizer 接口的资源请求的授权属性

类型

object

属性	类型	描述
group	字符串	group 是资源的 API 组。"*"表示全部。
name	字符串	name 是为"get"请求的资源的名称，或删除"delete"。"" (empty)表示 all。
namespace	字符串	namespace 是所请求操作的命名空间。目前，对于集群范围的资源 "" (empty)，没有命名空间和所有命名空间 "" (空) 为 LocalSubjectAccessReviews "" (empty)为空，代表来自 SubjectAccessReview 或 SelfSubjectAccessReview 的命名空间范围资源"all"

属性	类型	描述
resource	字符串	Resource 是现有资源类型之一。"*"表示全部。
subresource	字符串	子资源是现有资源类型之一。""表示 none。
verb	字符串	verb 是一个 kubernetes 资源 API 动词，如 get, list, watch, create, update, delete, proxy. "*"。
version	字符串	Version 是资源的 API 版本。"*"表示全部。

9.5.1.5. .status

描述

SubjectAccessReviewStatus

类型

object

必填

- **allowed**

属性	类型	描述
allowed	布尔值	Allowed 是必需的。True 代表操作被允许，false 代表不允许。
已拒绝	布尔值	Denied 是可选的。如果被拒绝，则为 True，否则为 false。如果允许都为 false，并且拒绝为 false，则授权者是在是否授权该操作时没有建议。如果允许为 true，则拒绝的可能不是 true。
evaluationError	字符串	EvaluationError 表示在授权检查过程中发生一些错误。完全有可能出错，并且能够继续确定其授权状态。例如，RBAC 可以缺少角色，但仍存在足够的角色并绑定到请求的原因。
reason	字符串	reason 是可选的。它表示为什么允许或拒绝请求。

9.5.2. API 端点

可用的 API 端点如下：

- `/apis/authorization.k8s.io/v1/subjectaccessreviews`
 - **POST**: 创建一个 SubjectAccessReview

9.5.2.1. `/apis/authorization.k8s.io/v1/subjectaccessreviews`

表 9.11. 全局查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
<code>fieldValidation</code>	string	<code>fieldValidation</code> 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 <code>BadRequest</code> 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
<code>pretty</code>	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建一个 SubjectAccessReview

表 9.12. 主体参数

参数	类型	描述
正文 (body)	<code>SubjectAccessReview</code> 模式	

表 9.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	SubjectAccessReview 模式
201 - Created	SubjectAccessReview 模式
202 - Accepted	SubjectAccessReview 模式
401 - Unauthorized	空

第 10 章 自动缩放 API

10.1. 自动缩放 API

10.1.1. scale [autoscaling/v1]

描述

scale 代表资源的扩展请求。

类型

object

10.1.2. HorizontalPodAutoscaler [autoscaling/v2]

描述

HorizontalPodAutoscaler 是 pod 横向自动扩展的配置，它根据指定的指标自动管理实施 scale 子资源的任何资源的副本数。

类型

object

10.2. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]

描述

HorizontalPodAutoscaler 是 pod 横向自动扩展的配置，它根据指定的指标自动管理实施 scale 子资源的任何资源的副本数。

类型

对象

10.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata 是标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	HorizontalPodAutoscalerSpec 描述了 HorizontalPodAutoscaler 的所需功能。
status	object	HorizontalPodAutoscalerStatus 描述了 pod 横向自动扩展的当前状态。

10.2.1.1. .spec

描述

HorizontalPodAutoscalerSpec 描述了 HorizontalPodAutoscaler 的所需功能。

类型

object

必填

- **scaleTargetRef**
- **maxReplicas**

属性	类型	描述
行为	object	HorizontalPodAutoscalerBehavior 在 Up 和 Down directions (scaleUp 和 scaleDown 字段)中配置目标的扩展行为。
maxReplicas	整数	maxReplicas 是自动扩展可以扩展到的副本数的上限。它不能小于 minReplicas。

属性	类型	描述
metrics	数组	指标包含用于计算所需副本数（将使用所有指标的最大副本数）的规格。所需的副本数根据当前 pod 数量来计算目标值和当前值之间的比率。Ergo，使用的指标必须减少，因为 pod 数量会增加，反之亦然。如需有关每种指标类型响应的更多信息，请参阅单独的指标源类型。如果没有设置，则默认指标将设置为 80% 平均 CPU 使用率。
metrics[]	object	MetricSpec 指定如何基于单一指标（仅 类型 和 另一个匹配字段应一次性设置）进行扩展。
minReplicas	整数	minReplicas 是自动扩展可以缩减副本数的较低限制。如果启用了 alpha 功能门 HPAScaleToZero，并且至少配置一个 Object 或 External 指标，则默认为 1 个 pod。minReplicas 被允许为 0。只要至少有一个指标值可用，扩展就会处于活跃状态。
scaleTargetRef	对象	CrossVersionObjectReference 包含足够信息，供您识别引用的资源。

10.2.1.2. .spec.behavior

描述

HorizontalPodAutoscalerBehavior 在 Up 和 Down directions (scaleUp 和 scaleDown 字段)中配置目标的扩展行为。

类型

object

属性	类型	描述
scaleDown	object	HPAScalingRules 配置一个方向的扩展行为。在为 HPA 的指标计算 DesiredReplicas 后，会应用这些规则。它们可以通过指定扩展策略来限制扩展速度。它们可以通过指定 stabilization 窗口来防止 flapping，以便不立即设置副本数，而是选择 stabilization 窗口中的安全值。

属性	类型	描述
scaleUp	object	HPAScalingRules 配置一个方向的扩展行为。在为 HPA 的指标计算 DesiredReplicas 后，会应用这些规则。它们可以通过指定扩展策略来限制扩展速度。它们可以通过指定 stabilization 窗口来防止 flapping，以便不立即设置副本数，而是选择 stabilization 窗口中的安全值。

10.2.1.3. .spec.behavior.scaleDown

描述

HPAScalingRules 配置一个方向的扩展行为。在为 HPA 的指标计算 DesiredReplicas 后，会应用这些规则。它们可以通过指定扩展策略来限制扩展速度。它们可以通过指定 stabilization 窗口来防止 flapping，以便不立即设置副本数，而是选择 stabilization 窗口中的安全值。

类型

object

属性	类型	描述
policies	数组	策略(policy)是一个潜在的扩展策略列表，可在扩展过程中使用。必须至少指定一个策略，否则 HPAScalingRules 将被丢弃为无效
policies[]	object	HPAScalingPolicy 是一个单一策略，必须为指定过去的时间间隔保存 true。
selectPolicy	string	selectPolicy 用于指定应使用哪个策略。如果没有设置，则使用默认值 Max。
stabilizationWindowSeconds	整数	stabilizationWindowSeconds 是扩展或缩减时应考虑过去建议的秒数。stabilizationWindowSeconds 必须大于或等于零，且小于或等于 3600（一小时）。如果没有设置，使用默认值：- 对于扩展：0（不进行稳定）。- 用于缩减：300（例如，stabilization 窗口为 300 秒）。

10.2.1.4. .spec.behavior.scaleDown.policies

描述

策略(policy)是一个潜在的扩展策略列表，可在扩展过程中使用。必须至少指定一个策略，否则 HPAScalingRules 将被丢弃为无效

类型

数组

10.2.1.5. .spec.behavior.scaleDown.policies[]

描述

HPAScalingPolicy 是一个单一策略，必须为指定过去的时间间隔保存 true。

类型

object

必填

- **type**
- **value**
- **periodSeconds**

属性	类型	描述
periodSeconds	整数	periodSeconds 指定策略应满足以下条件的时间窗口。 periodSeconds 必须大于零，且小于或等于 1800 (30 分钟)。
type	string	type 用于指定扩展策略。
value	整数	value 包含策略允许的更改量。它必须大于零

10.2.1.6. .spec.behavior.scaleUp

描述

HPAScalingRules 配置一个方向的扩展行为。在为 HPA 的指标计算 DesiredReplicas 后，会应用这些规则。它们可以通过指定扩展策略来限制扩展速度。它们可以通过指定 stabilization 窗口来防止 flapping，以便不立即设置副本数，而是选择 stabilization 窗口中的安全值。

类型

object

属性	类型	描述
policies	数组	策略(policy)是一个潜在的扩展策略列表，可在扩展过程中使用。必须至少指定一个策略，否则 HPAScalingRules 将被丢弃为无效

属性	类型	描述
policies[]	object	HPAScalingPolicy 是一个单一策略，必须为指定过去的时间间隔保存 true。
selectPolicy	string	selectPolicy 用于指定应使用哪个策略。如果没有设置，则使用默认值 Max。
stabilizationWindowSeconds	整数	stabilizationWindowSeconds 是扩展或缩减时应考虑过去建议的秒数。stabilizationWindowSeconds 必须大于或等于零，且小于或等于 3600（一小时）。如果没有设置，使用默认值：- 对于扩展：0（不进行稳定）。- 用于缩减：300（例如，stabilization 窗口为 300 秒）。

10.2.1.7. .spec.behavior.scaleUp.policies

描述

策略(policy)是一个潜在的扩展策略列表，可在扩展过程中使用。必须至少指定一个策略，否则 HPAScalingRules 将被丢弃为无效

类型

数组

10.2.1.8. .spec.behavior.scaleUp.policies[]

描述

HPAScalingPolicy 是一个单一策略，必须为指定过去的时间间隔保存 true。

类型

object

必填

- **type**
- **value**
- **periodSeconds**

属性	类型	描述
----	----	----

属性	类型	描述
periodSeconds	整数	periodSeconds 指定策略应满足以下条件的时间窗口。 periodSeconds 必须大于零，且小于或等于 1800 (30 分钟)。
type	string	type 用于指定扩展策略。
value	整数	value 包含策略允许的更改量。它必须大于零

10.2.1.9. .spec.metrics

描述

指标包含用于计算所需副本数（将使用所有指标的最大副本数）的规格。所需的副本数根据当前 pod 数量来计算目标值和当前值之间的比率。Ergo，使用的指标必须减少，因为 pod 数量会增加，反之亦然。如需有关每种指标类型响应的更多信息，请参阅单独的指标源类型。如果没有设置，则默认指标将设置为 80% 平均 CPU 使用率。

类型

数组

10.2.1.10. .spec.metrics[]

描述

MetricSpec 指定如何基于单一指标（仅 **类型**和 另一个匹配字段应一次性设置）进行扩展。

类型

object

必填

- **type**

属性	类型	描述
containerResource	object	ContainerResourceMetricSource 指明如何扩展 Kubernetes 的资源指标，如请求和限值中指定的资源指标，描述当前扩展目标中的每个 pod（如 CPU 或内存）。在与目标进行比较前，这些值将平均化在一起。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。应该只设置一个 "target" 类型。

属性	类型	描述
external	object	ExternalMetricSource 指明如何在不与任何 Kubernetes 对象关联的指标上扩展（例如，云消息传递服务中的队列长度，或者从集群外运行的 loadbalancer 中的 QPS）。
object	object	ObjectMetricSource 指明如何在描述 kubernetes 对象的指标上扩展（例如，Ingress 对象上的 hits-per-second）。
Pods	object	PodsMetricSource 指明了如何在一个指标上扩展，描述当前扩展目标中的每个 pod（如 transaction-processed-second）。在与目标值进行比较前，这些值将平均化在一起。
resource	object	ResourceMetricSource 指示如何扩展 Kubernetes 的资源指标，如请求和限值中指定的资源指标，描述当前扩展目标中的每个 pod（如 CPU 或内存）。在与目标进行比较前，这些值将平均化在一起。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。应该只设置一个 "target" 类型。
type	string	type 是指标源的类型。它应该是 "ContainerResource", "External", "Object", "Pods" 或 "Resource" 之一，每个映射到对象中的匹配字段。注："ContainerResource" 类型在启用了 feature-gate HPAContainerMetrics 时可用

10.2.1.11. .spec.metrics[].containerResource

描述

ContainerResourceMetricSource 指明如何扩展 Kubernetes 的资源指标，如请求和限值中指定的资源指标，描述当前扩展目标中的每个 pod（如 CPU 或内存）。在与目标进行比较前，这些值将平均化在一起。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。应该只设置一个 "target" 类型。

类型

object

必填

ContainerResource

- **name**
- **target**
- **container**

属性	类型	描述
container	string	Container 是扩展目标 pod 中的容器的名称
name	string	name 是问题的资源的名称。
target	object	MetricTarget 定义特定指标的目标值、平均值或平均利用率

10.2.1.12. .spec.metrics[].containerResource.target

描述

MetricTarget 定义特定指标的目标值、平均值或平均利用率

类型

object

必填

- **type**

属性	类型	描述
averageUtilization	整数	averageUtilization 是所有相关 pod 中资源指标的平均目标值，以 pod 的资源值的百分比表示。目前仅对 Resource metric 源类型有效
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的目标值（作为数量）
type	string	type 代表指标类型是 Utilization、Value 或 AverageValue
value	数量	value 是指标的目标值（作为数量）。

10.2.1.13. .spec.metrics[].external

描述

ExternalMetricSource 指明如何在不与任何 Kubernetes 对象关联的指标上扩展（例如，云消息传递服务中的队列长度，或者从集群外运行的 loadbalancer 中的 QPS）。

类型

object

必填

- **metric**
- **target**

属性	类型	描述
metric	object	MetricIdentifier 定义指标的名称和可选选择器
target	object	MetricTarget 定义特定指标的目标值、平均值或平均利用率

10.2.1.14. .spec.metrics[].external.metric

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是给定指标的名称
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 metricName 来收集指标。

10.2.1.15. .spec.metrics[].external.target

描述

MetricTarget 定义特定指标的目标值、平均值或平均利用率

类型

object

必填

- **type**

属性	类型	描述
averageUtilization	整数	averageUtilization 是所有相关 pod 中资源指标的平均目标值，以 pod 的资源值的百分比表示。目前仅对 Resource metric 源类型有效
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的目标值（作为数量）
type	string	type 代表指标类型是 Utilization、Value 或 AverageValue
value	数量	value 是指标的目标值（作为数量）。

10.2.1.16. .spec.metrics[].object

描述

ObjectMetricSource 指明如何在描述 kubernetes 对象的指标上扩展（例如，Ingress 对象上的 hits-per-second）。

类型

object

必填

- **describedObject**
- **target**
- **metric**

属性	类型	描述
describedObject	object	CrossVersionObjectReference 包含足够信息，供您识别引用的资源。
metric	object	MetricIdentifier 定义指标的名称和可选选择器
target	object	MetricTarget 定义特定指标的目标值、平均值或平均利用率

10.2.1.17. .spec.metrics[].object.describedObject

描述

CrossVersionObjectReference 包含足够信息，供您识别引用的资源。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiVersion	string	apiVersion 是引用的 API 版本
kind	string	kind 是引用的类型；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	name 是引用的名称；更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.2.1.18. `.spec.metrics[].object.metric`

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是给定指标的名称
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 metricName 来收集指标。

10.2.1.19. `.spec.metrics[].object.target`

描述

MetricTarget 定义特定指标的目标值、平均值或平均利用率

类型

object

必填

- type

属性	类型	描述
averageUtilization	整数	averageUtilization 是所有相关 pod 中资源指标的平均目标值，以 pod 的资源值的百分比表示。目前仅对 Resource metric 源类型有效
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的目标值（作为数量）
type	string	type 代表指标类型是 Utilization、Value 或 AverageValue
value	数量	value 是指标的目标值（作为数量）。

10.2.1.20. .spec.metrics[].pods

描述

PodsMetricSource 指明了如何在一个指标上扩展，描述当前扩展目标中的每个 pod（如 transaction-processed-second）。在与目标值进行比较前，这些值将平均化在一起。

类型

object

必填

- metric
- target

属性	类型	描述
metric	object	MetricIdentifier 定义指标的名称和可选选择器
target	object	MetricTarget 定义特定指标的目标值、平均值或平均利用率

10.2.1.21. .spec.metrics[].pods.metric

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是给定指标的名称
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 metricName 来收集指标。

10.2.1.22. .spec.metrics[].pods.target

描述

MetricTarget 定义特定指标的目标值、平均值或平均利用率

类型

object

必填

- **type**

属性	类型	描述
averageUtilization	整数	averageUtilization 是所有相关 pod 中资源指标的平均目标值，以 pod 的资源值的百分比表示。目前仅对 Resource metric 源类型有效
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的目标值（作为数量）
type	string	type 代表指标类型是 Utilization、Value 或 AverageValue
value	数量	value 是指标的目标值（作为数量）。

10.2.1.23. .spec.metrics[].resource

描述

ResourceMetricSource 指示如何扩展 Kubernetes 的资源指标，如请求和限值中指定的资源指标，描

述当前扩展目标中的每个 pod（如 CPU 或内存）。在与目标进行比较前，这些值将平均化在一起。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。应该只设置一个 "target" 类型。

类型

object

必填

- **name**
- **target**

属性	类型	描述
name	string	name 是问题的资源的名称。
target	object	MetricTarget 定义特定指标的目标值、平均值或平均利用率

10.2.1.24. .spec.metrics[].resource.target

描述

MetricTarget 定义特定指标的目标值、平均值或平均利用率

类型

object

必填

- **type**

属性	类型	描述
averageUtilization	整数	averageUtilization 是所有相关 pod 中资源指标的平均目标值，以 pod 的资源值的百分比表示。目前仅对 Resource metric 源类型有效
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的目标值（作为数量）
type	string	type 代表指标类型是 Utilization、Value 或 AverageValue
value	数量	value 是指标的目标值（作为数量）。

10.2.1.25. .spec.scaleTargetRef

描述

CrossVersionObjectReference 包含足够信息，供您识别引用的资源。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiVersion	string	apiVersion 是引用的 API 版本
kind	string	kind 是引用的类型；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	name 是引用的名称；更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.2.1.26. .status

描述

HorizontalPodAutoscalerStatus 描述了 pod 横向自动扩展的当前状态。

类型

object

必填

- **desiredReplicas**

属性	类型	描述
conditions	数组	条件是该自动扩展器扩展其目标所需的条件集合，指示是否满足这些条件。
conditions[]	object	HorizontalPodAutoscalerCondition 描述了 HorizontalPodAutoscaler 在特定点的状态。
currentMetrics	数组	currentMetrics 是该自动扩展器使用的指标的最后读取状态。

属性	类型	描述
currentMetrics[]	object	MetricStatus 描述了单个指标的最后读取状态。
currentReplicas	整数	currentReplicas 是此自动扩展管理的 pod 副本数，如自动扩展器的最后一个所示。
desiredReplicas	整数	desiredReplicas 是此自动扩展管理的 pod 所需的副本数，如自动扩展器计算的最后计算。
lastScaleTime	时间	lastScaleTime 是 HorizontalPodAutoscaler 最后一次扩展 pod 数量的时间，供自动扩展器用于控制 pod 数量更改的频率。
observedGeneration	整数	observedGeneration 是此自动扩展观察到的最新生成。

10.2.1.27. .status.conditions

描述

条件是此自动扩展扩展其目标所需的条件集合，指示是否满足这些条件。

类型

array

10.2.1.28. .status.conditions[]

描述

HorizontalPodAutoscalerCondition 描述了 HorizontalPodAutoscaler 在特定点的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件从一个状态转换到另一个状态最后一次的时间

属性	类型	描述
message	string	Message 是人类可读的解释，其中包含有关转换的详细信息
reason	字符串	reason 是条件最后一次转换的原因。
status	string	status 是条件的状态(True, False, Unknown)
type	string	type 描述了当前条件

10.2.1.29. .status.currentMetrics

描述

currentMetrics 是此自动扩展器使用的指标的最后读取状态。

类型

数组

10.2.1.30. .status.currentMetrics[]

描述

MetricStatus 描述了单个指标的最后读取状态。

类型

object

必填

- type

属性	类型	描述
containerResource	object	ContainerResourceMetricStatus 表示 Kubernetes 已知资源指标的当前值，如请求和限值中指定的资源指标，描述当前扩展目标中的每个 pod 中的单个容器（如 CPU 或内存）。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。
external	object	ExternalMetricStatus 表示没有与任何 Kubernetes 对象关联的全局指标的当前值。

属性	类型	描述
object	object	ObjectMetricStatus 表示指标的当前值，描述 kubernetes 对象（如 Ingress 对象上的 hits-per-second）。
Pods	object	PodsMetricStatus 表示指标的当前值，用于描述当前 scale 目标中的每个 pod（如 transaction-processed-second）。
resource	object	ResourceMetricStatus 表示 Kubernetes 已知的资源指标的当前值，如请求和限值中指定的，描述当前扩展目标中的每个 pod（如 CPU 或内存）。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。
type	string	type 是指标源的类型。它将是 "ContainerResource", "External", "Object", "Pods" 或 "Resource" 之一，每个都对应于对象中的匹配字段。注："ContainerResource" 类型在启用了 feature-gate HPAContainerMetrics 时可用

10.2.1.31. .status.currentMetrics[].containerResource

描述

ContainerResourceMetricStatus 表示 Kubernetes 已知资源指标的当前值，如请求和限值中指定的资源指标，描述当前扩展目标中的每个 pod 中的单个容器（如 CPU 或内存）。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。

类型

object

必填

- **name**
- **current**
- **container**

属性	类型	描述
container	string	Container 是扩展目标 pod 中的容器的名称
current	object	MetricValueStatus 包含指标的当前值
name	string	name 是问题的资源的名称。

10.2.1.32. .status.currentMetrics[].containerResource.current

描述

MetricValueStatus 包含指标的当前值

类型

object

属性	类型	描述
averageUtilization	整数	currentAverageUtilization 是所有相关 pod 中资源指标的平均值的当前值，以 pod 的资源请求值的百分比表示。
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的当前值（作为数量）
value	数量	value 是指标的当前值（作为数量）。

10.2.1.33. .status.currentMetrics[].external

描述

ExternalMetricStatus 表示没有与任何 Kubernetes 对象关联的全局指标的当前值。

类型

object

必填

- **metric**
- **current**

属性	类型	描述
current	object	MetricValueStatus 包含指标的当前值
metric	object	MetricIdentifier 定义指标的名称和可选选择器

10.2.1.34. .status.currentMetrics[].external.current

描述

MetricValueStatus 包含指标的当前值

类型

object

属性	类型	描述
averageUtilization	整数	currentAverageUtilization 是所有相关 pod 中资源指标的平均值的当前值，以 pod 的资源请求值的百分比表示。
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的当前值（作为数量）
value	数量	value 是指标的当前值（作为数量）。

10.2.1.35. .status.currentMetrics[].external.metric

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是给定指标的名称

属性	类型	描述
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 <code>metricName</code> 来收集指标。

10.2.1.36. `.status.currentMetrics[].object`

描述

`ObjectMetricStatus` 表示指标的当前值，描述 kubernetes 对象（如 Ingress 对象上的 `hits-per-second`）。

类型

object

必填

- **metric**
- **current**
- **describedObject**

属性	类型	描述
current	object	<code>MetricValueStatus</code> 包含指标的当前值
describedObject	object	<code>CrossVersionObjectReference</code> 包含足够信息，供您识别引用的资源。
metric	object	<code>MetricIdentifier</code> 定义指标的名称和可选选择器

10.2.1.37. `.status.currentMetrics[].object.current`

描述

`MetricValueStatus` 包含指标的当前值

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
averageUtilization	整数	currentAverageUtilization 是所有相关 pod 中资源指标的平均值的当前值，以 pod 的资源请求值的百分比表示。
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的当前值（作为数量）
value	数量	value 是指标的当前值（作为数量）。

10.2.1.38. .status.currentMetrics[].object.describedObject

描述

CrossVersionObjectReference 包含足够信息，供您识别引用的资源。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiVersion	string	apiVersion 是引用的 API 版本
kind	string	kind 是引用的类型；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	name 是引用的名称；更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

10.2.1.39. .status.currentMetrics[].object.metric

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是给定指标的名称
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 metricName 来收集指标。

10.2.1.40. .status.currentMetrics[].pods

描述

PodsMetricStatus 表示指标的当前值，用于描述当前 scale 目标中的每个 pod（如 transaction-processed-second）。

类型

object

必填

- **metric**
- **current**

属性	类型	描述
current	object	MetricValueStatus 包含指标的当前值
metric	object	MetricIdentifier 定义指标的名称和可选选择器

10.2.1.41. .status.currentMetrics[].pods.current

描述

MetricValueStatus 包含指标的当前值

类型

object

属性	类型	描述
averageUtilization	整数	currentAverageUtilization 是所有相关 pod 中资源指标的平均值的当前值，以 pod 的资源请求值的百分比表示。
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的当前值（作为数量）
value	数量	value 是指标的当前值（作为数量）。

10.2.1.42. .status.currentMetrics[].pods.metric

描述

MetricIdentifier 定义指标的名称和可选选择器

类型

object

必填

- name

属性	类型	描述
name	string	name 是给定指标的名称
selector	labelSelector	选择器是给定指标时的标准 kubernetes 标签选择器的字符串编码形式，它将作为额外的参数传递给指标服务器，以获取更具体的指标范围。如果未设置，只需使用 metricName 来收集指标。

10.2.1.43. .status.currentMetrics[].resource

描述

ResourceMetricStatus 表示 Kubernetes 已知的资源指标的当前值，如请求和限值中指定的，描述当前扩展目标中的每个 pod（如 CPU 或内存）。此类指标内置在 Kubernetes 中，并在那些可用于使用 "pods" 源的普通每个 pod 指标上具有特殊的扩展选项。

类型

object

必填

- name
- current

属性	类型	描述
current	object	MetricValueStatus 包含指标的当前值
name	string	name 是问题的资源的名称。

10.2.1.44. .status.currentMetrics[].resource.current

描述

MetricValueStatus 包含指标的当前值

类型

object

属性	类型	描述
averageUtilization	整数	currentAverageUtilization 是所有相关 pod 中资源指标的平均值的当前值，以 pod 的资源请求值的百分比表示。
averageValue	数量	AverageValue 是所有相关 pod 的指标平均值的当前值（作为数量）
value	数量	value 是指标的当前值（作为数量）。

10.2.2. API 端点

可用的 API 端点如下：

- **/apis/autoscaling/v2/horizontalpodautoscalers**
 - **GET**: 列出或监视类型为 HorizontalPodAutoscaler 的对象
- **/apis/autoscaling/v2/watch/horizontalpodautoscalers**
 - **GET**: 观察单个对 HorizontalPodAutoscaler 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers**
 - **DELETE**：删除 HorizontalPodAutoscaler 的集合
 - **GET**: 列出或监视类型为 HorizontalPodAutoscaler 的对象
 - **POST**：创建一个 HorizontalPodAutoscaler
- **/apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers**
 - **GET**: 观察单个对 HorizontalPodAutoscaler 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作

1152 页码 1150

- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}**
 - **DELETE** : 删除 HorizontalPodAutoscaler
 - **GET**: 读取指定的 HorizontalPodAutoscaler
 - **PATCH** : 部分更新指定的 HorizontalPodAutoscaler
 - **PUT** : 替换指定的 HorizontalPodAutoscaler
- **/apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers/{name}**
 - **GET**: 观察对类型为 HorizontalPodAutoscaler 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/status**
 - **GET**: 指定的 HorizontalPodAutoscaler 的读取状态
 - **PATCH** : 部分更新指定 HorizontalPodAutoscaler 的状态
 - **PUT** : 替换指定 HorizontalPodAutoscaler 的状态

10.2.2.1. /apis/autoscaling/v2/horizontalpodautoscalers

表 10.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签, 由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回, 也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视, 则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 HorizontalPodAutoscaler 类型的对象

表 10.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscalerList schema
401 - Unauthorized	空

10.2.2.2. /apis/autoscaling/v2/watch/horizontalpodautoscalers

表 10.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 HorizontalPodAutoscaler 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 10.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

10.2.2.3. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers

表 10.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 10.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 HorizontalPodAutoscaler 集合

表 10.7. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 10.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 10.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 HorizontalPodAutoscaler 类型的对象

表 10.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 10.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscalerList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 HorizontalPodAutoscaler

表 10.12. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.13. 主体参数

参数	类型	描述
正文 (body)	HorizontalPodAutoscaler 模式	

表 10.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
201 - Created	HorizontalPodAutoscaler 模式
202 - Accepted	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

10.2.2.4. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers

表 10.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 10.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 HorizontalPodAutoscaler 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 10.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

10.2.2.5. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}

表 10.18. 全局路径参数

参数	类型	描述
name	字符串	HorizontalPodAutoscaler 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 10.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

delete a HorizontalPodAutoscaler

表 10.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 10.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 10.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 `HorizontalPodAutoscaler`

表 10.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>HorizontalPodAutoscaler</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `HorizontalPodAutoscaler`

表 10.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
201 - Created	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 HorizontalPodAutoscaler

表 10.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.28. 主体参数

参数	类型	描述
正文 (body)	HorizontalPodAutoscaler 模式	

表 10.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
201 - Created	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

10.2.2.6. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers/{

表 10.30. 全局路径参数

参数	类型	描述
name	字符串	HorizontalPodAutoscaler 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 10.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind HorizontalPodAutoscaler 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 10.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

10.2.2.7. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/

表 10.33. 全局路径参数

参数	类型	描述
name	字符串	HorizontalPodAutoscaler 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 10.34. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 HorizontalPodAutoscaler 的读取状态

表 10.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 HorizontalPodAutoscaler 的部分更新状态

表 10.36. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.37. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.38. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
201 - Created	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 HorizontalPodAutoscaler 的状态

表 10.39. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.40. 主体参数

参数	类型	描述
正文 (body)	HorizontalPodAutoscaler 模式	

表 10.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	HorizontalPodAutoscaler 模式
201 - Created	HorizontalPodAutoscaler 模式
401 - Unauthorized	空

10.3. SCALE [AUTOSCALING/V1]

描述

scale 代表资源的扩展请求。

类型

对象

10.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata 。
spec	对象	ScaleSpec 描述 scale 子资源的属性。
status	对象	ScaleStatus 代表 scale 子资源的当前状态。

10.3.1.1. .spec

描述

ScaleSpec 描述 scale 子资源的属性。

类型

object

属性	类型	描述
replicas	整数	replicas 是扩展对象的所需实例数量。

10.3.1.2. .status

描述

ScaleStatus 代表 scale 子资源的当前状态。

类型

object

必填

- replicas

属性	类型	描述
replicas	整数	replicas 是扩展对象的观察实例的实际数量。
selector	string	selector 是 pod 的标签查询，应与副本数匹配。这与标签选择器相同，但使用字符串格式以避免客户端内省。字符串的格式与 query-param 语法相同。有关标签选择器的更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/

10.3.2. API 端点

可用的 API 端点如下：

- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale**
 - **GET**：指定 Deployment 的读取扩展
 - **PATCH**：部分更新指定的部署扩展
 - **PUT**：替换指定部署的规模
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}/scale**
 - **GET**：指定 ReplicaSet 的读取扩展
 - **PATCH**：部分更新指定 ReplicaSet 的扩展
 - **PUT**：替换指定 ReplicaSet 的扩展
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale**
 - **GET**：指定 StatefulSet 的读取扩展
 - **PATCH**：部分更新指定 StatefulSet 的扩展
 - **PUT**：替换指定 StatefulSet 的扩展
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale**
 - **GET**：指定 ReplicationController 的读取扩展

- **PATCH** : 部分更新指定的 ReplicationController 的扩展
- **PUT** : 替换指定 ReplicationController 的扩展

10.3.2.1. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale

表 10.42. 全局路径参数

参数	类型	描述
name	字符串	扩展的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 10.43. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定部署的读取扩展

表 10.44. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定部署的部分更新扩展

表 10.45. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为： <ul style="list-style-type: none"> - Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn：这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict：如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.46. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.47. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 Deployment 的扩展

表 10.48. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.49. 主体参数

参数	类型	描述
正文 (body)	scale 模式	

表 10.50. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

10.3.2.2. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}/scale

表 10.51. 全局路径参数

参数	类型	描述
name	字符串	扩展的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 10.52. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 ReplicaSet 的读取扩展

表 10.53. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 ReplicaSet 的部分更新扩展

表 10.54. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为： <ul style="list-style-type: none"> - Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.55. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.56. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 ReplicaSet 的扩展

表 10.57. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为： <ul style="list-style-type: none"> - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.58. 主体参数

参数	类型	描述
正文 (body)	scale 模式	

表 10.59. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

10.3.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale

表 10.60. 全局路径参数

参数	类型	描述
name	字符串	扩展的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 10.61. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

指定 StatefulSet 的读取扩展

表 10.62. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

指定 StatefulSet 的部分更新扩展

表 10.63. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.64. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.65. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 StatefulSet 的扩展

表 10.66. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.67. 主体参数

参数	类型	描述
正文 (body)	scale 模式	

表 10.68. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

10.3.2.4. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale

表 10.69. 全局路径参数

参数	类型	描述
name	字符串	扩展的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 10.70. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

读取指定的 ReplicationController 的扩展

表 10.71. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定 ReplicationController 的扩展

表 10.72. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 10.73. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 10.74. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 ReplicationController 的规模

表 10.75. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 10.76. 主体参数

参数	类型	描述
正文 (body)	scale 模式	

表 10.77. HTTP 响应

HTTP 代码	响应正文
200 - OK	scale 模式
201 - Created	scale 模式
401 - Unauthorized	空

第 11 章 批处理 API

11.1. 批处理 API

11.1.1. CronJob [batch/v1]

描述

CronJob 代表单个 cron 作业的配置。

类型

object

11.1.2. 作业 [batch/v1]

描述

Job 代表单个作业的配置。

类型

object

11.2. CRONJOB [BATCH/V1]

描述

CronJob 代表单个 cron 作业的配置。

类型

对象

11.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	CronJobSpec 描述了作业执行如何样子以及实际运行的时间。
status	object	CronJobStatus 代表 cron 作业的当前状态。

11.2.1.1. .spec

描述

CronJobSpec 描述了作业执行如何样子以及实际运行的时间。

类型

object

必填

- 调度
- `jobTemplate`

属性	类型	描述
concurrencyPolicy	string	指定如何对待作业的并发执行。有效值为： - "allow"（默认）：允许 CronJob 同时运行； - "Forbid": forbids 并发运行，如果上一个运行尚未完成，则跳过下一个运行； - "Replace": 取消当前运行的作业并将其替换为新作业 可能枚举值： - "Allow" 允许 CronJob 同时运行。 - "Forbid" forbids 并发运行，如果之前还没有完成，则跳过下一个运行。 - "Replace" 取消当前运行的作业，并将其替换为新的作业。
failedJobsHistoryLimit	整数	要保留的失败完成作业数量。值必须是非负整数。默认为 1。

属性	类型	描述
jobTemplate	object	JobTemplateSpec 描述了从模板创建时作业应具有的数据
调度	string	Cron 格式的调度，请参阅 https://en.wikipedia.org/wiki/Cron 。
startingDeadlineSeconds	整数	如果因任何原因而错过调度时间，则启动该作业的可选期限（以秒为单位）。错过的作业执行计为失败的作业。
successfulJobsHistoryLimit	整数	要保留的成功完成作业数量。值必须是非负整数。默认值为 3。
suspend	布尔值	此标志告知控制器挂起后续执行，它不适用于已经开始的执行。默认为 false。
timeZone	string	给定调度的时区名称，请参阅 https://en.wikipedia.org/wiki/List_of_tz_database_time_zones 。如果没有指定，则默认为 kube-controller-manager 进程的时区。在 CronJob 验证和控制器管理器期间，有效时区名称和时区偏移从 API 服务器的系统范围的时区数据库加载。如果没有系统范围的时区数据库，则可以使用捆绑的数据库版本。如果时区名称在 CronJob 的生命周期内无效，或者因为主机配置中发生了变化，控制器将停止创建新作业，并创建一个原因为 UnknownTimeZone 的系统事件。如需更多信息，请参阅 https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/#time-zones

11.2.1.2. .spec.jobTemplate

描述

JobTemplateSpec 描述了从模板创建时作业应具有的数据

类型

object

属性	类型	描述
metadata	ObjectMeta	标准对象从该模板创建的作业元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	JobSpec 描述了作业执行如何。

11.2.1.3. .spec.jobTemplate.spec

描述

JobSpec 描述了作业执行如何。

类型

object

必填

- 模板

属性	类型	描述
activeDeadlineSeconds	整数	指定与 <code>startTime</code> 相对的持续时间（以秒为单位），在系统尝试终止前可以持续激活；值必须是正整数。如果作业被暂停（创建或通过更新），在作业再次恢复时，此计时器将有效停止并重置。
backoffLimit	整数	指定在标记此作业失败前的重试次数。默认值为 6

属性	类型	描述
completionMode	string	<p>completionMode 指定 Pod 的完成情况。它可以是 NonIndexed (默认) 或 Indexed。</p> <p>NonIndexed 表示当有 .spec.completions 成功完成 Pod 时，作业被视为已完成。每个 Pod 完成都相互不同。</p> <p>indexed 表示作业的 Pod 从 0 变为 (.spec.completions - 1)，在注解 batch.kubernetes.io/job-completion-index 中提供。当每个索引都有一个成功完成 Pod 时，作业被视为已完成。当值被 索引 时，必须指定 .spec.completions, .spec.parallelism 必须小于或等于 10^5。另外，Pod 名称的格式是 \$(job-name)-\$(index)-\$(random-string)，Pod 主机名采用 \$(job-name)-\$(index) 的形式。</p> <p>以后可以添加更多完成模式。如果作业控制器观察一个没有识别的模式，则在升级过程中会因为版本偏移而可能，控制器会跳过该作业的更新。</p> <p>可能的值有：- "Indexed" 是一个作业完成模式。在这个模式中，作业的 Pod 从 0 变为 (.spec.completions - 1)。当 Pod 完成每个完成索引时，作业被视为已完成。- "NonIndexed" 是一个作业完成模式。在这个模式中，当有 .spec.completions 成功完成 Pod 时，作业被视为已完成。Pod 完成将相互同构。</p>
completions	整数	<p>指定作业应使用的成功完成 pod 数量。设置为 null 表示任何 pod 的成功表示所有 pod 的成功，并允许并行具有任何正值。设置为 1 表示并行性仅限于 1，并且该 pod 的成功表示作业成功。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/</p>

属性	类型	描述
manualSelector	布尔值	<p>manualSelector 控制 pod 标签和 pod 选择器的生成。请保留 manualSelector 取消设置，除非您确定您要做什么。为 false 或未设置时，系统会选择此作业的唯一标签，并将这些标签附加到 pod 模板。为 true 时，用户负责选择唯一标签并指定选择器。无法选择唯一标签可能会导致这个和其他作业无法正常工作。但是，在使用旧 extensions/v1beta1 API 创建的作业中可能会看到 manualSelector=true。更多信息：</p> <p>https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector</p>
parallelism	整数	<p>指定作业应在任意给定时间运行的最大 pod 数量。当 $((\text{spec.completions} - \text{.status.successful}) < \text{.spec.parallelism}) < \text{.spec.parallelism}$ 时，以 steady 状态运行的 pod 数量将小于这个数字。更多信息：</p> <p>https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/</p>
podFailurePolicy	object	PodFailurePolicy 描述了 pod 失败的 pod 如何影响 backoffLimit。
selector	labelSelector	<p>对 pod 数量应该匹配的 pod 的标签查询。通常，系统会为您设置此字段。更多信息：</p> <p>https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors</p>

属性	类型	描述
suspend	布尔值	suspend 指定作业控制器是否应该创建 Pod。如果创建了作业，并将 suspend 设置为 true，则作业控制器不会创建任何 Pod。如果在创建后作业被暂停（例如，标志从 false 变为 true），则作业控制器将删除与此作业关联的所有活跃 Pod。用户必须设计其工作负载才能正常处理此问题。暂停作业将重置作业的 StartTime 字段，从而有效地重置 ActiveDeadlineSeconds 计时器。默认为 false。
模板	PodTemplateSpec	描述在执行作业时创建的 pod。唯一允许的 template.spec.restartPolicy 值为 "Never" 或 "OnFailure"。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
ttlSecondsAfterFinished	整数	ttlSecondsAfterFinished 限制已完成执行的作业的生命周期(Complete 或 Failed)。如果设置了此字段，在作业完成后，ttlSecondsAfterFinished 会被自动删除。当作业被删除时，其生命周期保证（如终结器）将会被遵守。如果此字段未设置，则作业不会被自动删除。如果此字段设为零，则作业在完成后立即被删除。

11.2.1.4. .spec.jobTemplate.spec.podFailurePolicy

描述

PodFailurePolicy 描述了 pod 失败的 pod 如何影响 backoffLimit。

类型

object

必填

- rules

属性	类型	描述
----	----	----

属性	类型	描述
rules	数组	Pod 失败策略规则列表。规则按顺序评估。当规则与 Pod 失败时，会忽略其余的规则。如果没有规则与 Pod 失败匹配，则应用默认处理 - pod 故障计数器会递增，并根据 backoffLimit 检查。最多允许 20 个元素。
rules[]	object	PodFailurePolicyRule 描述了在满足要求时如何处理 pod 失败。每个规则都可以使用 onExitCodes 和 onPodConditions 中的一个。

11.2.1.5. .spec.jobTemplate.spec.podFailurePolicy.rules

描述

Pod 失败策略规则列表。规则按顺序评估。当规则与 Pod 失败时，会忽略其余的规则。如果没有规则与 Pod 失败匹配，则应用默认处理 - pod 故障计数器会递增，并根据 backoffLimit 检查。最多允许 20 个元素。

类型

数组

11.2.1.6. .spec.jobTemplate.spec.podFailurePolicy.rules[]

描述

PodFailurePolicyRule 描述了在满足要求时如何处理 pod 失败。每个规则都可以使用 onExitCodes 和 onPodConditions 中的一个。

类型

object

必填

- **action**
- **onPodConditions**

属性	类型	描述
----	----	----

属性	类型	描述
action	string	<p>指定在满足要求时对 pod 失败执行的操作。可能的值有：</p> <ul style="list-style-type: none"> - FailJob: 表示 pod 的作业标记为 Failed，所有正在运行的 pod 都被终止。 - Ignore: 表示与 .backoffLimit 相关的计数器没有被递增，并创建了替换 pod。 - Count: 表示 pod 以默认处理 - .backoffLimit 的计数器递增。以后会考虑添加其他值。客户端应通过跳过规则来响应未知操作。 <p>possible enum value: - "Count" 是一个在 pod 失败时执行的操作 - pod 失败以默认的方式处理 - pod 失败 - 对应的 .backoffLimit 计数器，由作业的 .status.failed 字段表示。 - "FailJob" is an action, may be 在 pod 失败时 - 将 pod 的作业标记为 Failed 并终止所有正在运行的 pod。 - "Ignore" 是一个在 pod 失败时执行的操作 - .backoffLimit 的计数器。作业的 .status.failed 字段代表不会被递增，并创建了替代 pod。</p>
onExitCodes	object	<p>PodFailurePolicyOnExitCodesRequirement 描述了根据容器退出代码处理失败的 pod 的要求。特别是，它会为每个应用程序容器和 init 容器状态查找 .state.terminated.exitCode，分别由 Pod 状态中的 .status.containerStatuses 和 .status.initContainerStatuses 字段表示。使用成功(exit code 0)完成的容器不包括在要求检查中。</p>
onPodConditions	数组	<p>代表 pod 条件的要求。要求以 pod 条件模式列表表示。如果至少有一个模式匹配实际 pod 条件，则需要满足要求。最多允许 20 个元素。</p>
onPodConditions[]	object	<p>PodFailurePolicyOnPodConditionPattern 描述与实际 pod 条件类型匹配的模式。</p>

11.2.1.7. .spec.jobTemplate.spec.podFailurePolicy.rules[].onExitCodes

描述

PodFailurePolicyOnExitCodesRequirement 描述了根据容器退出代码处理失败的 pod 的要求。特别是，它会为每个应用程序容器和 init 容器状态查找 `.state.terminated.exitCode`，分别由 Pod 状态中的 `.status.containerStatuses` 和 `.status.initContainerStatuses` 字段表示。使用成功(exit code 0)完成的容器不包括在要求检查中。

类型

object

必填

- **operator**
- **values**

属性	类型	描述
containerName	string	使用指定名称将退出代码限制为容器的检查。当 null 时，该规则适用于所有容器。指定后，它应当与 pod 模板中的一个容器或 initContainer 名称匹配。
operator	string	表示容器退出代码和指定的值之间的关系。使用成功(exit code 0)完成的容器不包括在要求检查中。可能的值有： - in: 如果至少有一个容器退出代码（如果多个容器没有由"containerName"字段限制）限制了多个容器，则满足要求。- 如果至少一个容器退出代码（如果多个容器不受"containerName"字段限制），则满足要求。以后会考虑添加其他值。客户端应不满足要求来响应未知操作器。 possible enum 值： - "In" - "NotIn"
values	数组（整数）	指定一组值。每个返回的容器退出代码（如果有多个容器）都会针对与 Operator 相关的这组值进行检查。值列表必须排序，且不得包含重复值。值 '0' 不能用于 In operator。至少需要一个元素。最多允许 255 个元素。

11.2.1.8. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions

描述

此规则描述了根据容器退出代码处理失败的 pod 的要求。特别是，它会为每个应用程序容器和 init 容器状态查找 `.state.terminated.exitCode`，分别由 Pod 状态中的 `.status.containerStatuses` 和 `.status.initContainerStatuses` 字段表示。使用成功(exit code 0)完成的容器不包括在要求检查中。

代表 pod 条件的要求。要求以 pod 条件模式列表表示。如果至少有一个模式匹配实际 pod 条件，则需要满足要求。最多允许 20 个元素。

类型

数组

11.2.1.9. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions[]

描述

PodFailurePolicyOnPodConditionsPattern 描述与实际 pod 条件类型匹配的模式。

类型

object

必填

- **type**
- **status**

属性	类型	描述
status	string	指定所需的 Pod 条件状态。要匹配 pod 条件，需要指定状态等于 pod 条件状态。默认值为 True。
type	string	指定所需的 Pod 条件类型。要匹配 pod 条件，需要指定类型等于 pod 条件类型。

11.2.1.10. .status

描述

CronJobStatus 代表 cron 作业的当前状态。

类型

object

属性	类型	描述
active	数组(ObjectReference)	到当前运行的作业的指针列表。
lastScheduleTime	时间	当作业最后一次调度作业的时间时的信息。
lastSuccessfulTime	时间	作业最后一次完成后的信息。

11.2.2. API 端点

可用的 API 端点如下：

- **/apis/batch/v1/cronjobs**
 - **GET**: 列表或监视类型为 CronJob 的对象
- **/apis/batch/v1/watch/cronjobs**
 - **GET**: 观察单个对 CronJob 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/batch/v1/namespaces/{namespace}/cronjobs**
 - **DELETE** : 删除 CronJob 的集合
 - **GET**: 列表或监视类型为 CronJob 的对象
 - **POST** : 创建 CronJob
- **/apis/batch/v1/watch/namespaces/{namespace}/cronjobs**
 - **GET**: 观察单个对 CronJob 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/batch/v1/namespaces/{namespace}/cronjobs/{name}**
 - **DELETE** : 删除 CronJob
 - **GET**: 读取指定的 CronJob
 - **PATCH** : 部分更新指定的 CronJob
 - **PUT** : 替换指定的 CronJob
- **/apis/batch/v1/watch/namespaces/{namespace}/cronjobs/{name}**
 - **GET**: 观察对类型为 CronJob 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status**
 - **GET**: 指定 CronJob 的读取状态
 - **PATCH** : 部分更新指定 CronJob 的状态
 - **PUT** : 替换指定 CronJob 的状态

11.2.2.1. /apis/batch/v1/cronjobs

表 11.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 <code>continue</code> 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 <code>continue</code> 值，并带有相同的查询参数（除 <code>continue</code> 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 <code>continue</code> 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 <code>ResourceExpired</code> 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 <code>true</code> 时，不支持此字段。客户端可以从服务器返回的最后一个 <code>resourceVersion</code> 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p><code>limit</code> 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 '<code>continue</code>' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 <code>continue</code> 字段的存在来确定是否有可用的结果。服务器可能选择不支持 <code>limit</code> 参数，并将返回所有可用结果。如果指定了 <code>limit</code>，并且 <code>continue</code> 字段为空，客户端可能会假设没有更多结果可用。如果 <code>watch</code> 为 <code>true</code>，则不支持此字段。</p> <p>服务器保证，在使用 <code>continue</code> 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 ' <code>true</code> '，则输出会经过 <code>pretty print</code> 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 CronJob 的对象

表 11.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJobList 模式
401 - 未授权	空

11.2.2.2. /apis/batch/v1/watch/cronjobs

表 11.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CronJob 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 11.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.2.2.3. /apis/batch/v1/namespaces/{namespace}/cronjobs

表 11.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CronJob 的集合

表 11.7. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 11.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 11.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 CronJob 的对象

表 11.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 11.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJobList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建 CronJob

表 11.12. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.13. 主体参数

参数	类型	描述
正文 (body)	CronJob 模式	

表 11.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
201 - Created	CronJob 模式
202 - Accepted	CronJob 模式
401 - 未授权	空

11.2.2.4. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs

表 11.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CronJob 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 11.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.2.2.5. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}

表 11.18. 全局路径参数

参数	类型	描述
name	string	CronJob 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 11.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CronJob

表 11.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 11.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 11.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 CronJob

表 11.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>CronJob</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 CronJob

表 11.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 11.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 11.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
201 - Created	CronJob 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 CronJob

表 11.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.28. 主体参数

参数	类型	描述
正文 (body)	CronJob 模式	

表 11.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
201 - Created	CronJob 模式
401 - 未授权	空

11.2.2.6. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs/{name}

表 11.30. 全局路径参数

参数	类型	描述
name	string	CronJob 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 11.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

监视对类型为 CronJob 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 11.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.2.2.7. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status

表 11.33. 全局路径参数

参数	类型	描述
name	string	CronJob 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 11.34. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 CronJob 的读取状态

表 11.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 CronJob 的部分更新状态

表 11.36. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 11.37. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 11.38. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
201 - Created	CronJob 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 CronJob 的状态

表 11.39. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.40. 主体参数

参数	类型	描述
正文 (body)	CronJob 模式	

表 11.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	CronJob 模式
201 - Created	CronJob 模式
401 - 未授权	空

11.3. 作业 [BATCH/V1]

描述

Job 代表单个作业的配置。

类型

对象

11.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	JobSpec 描述了作业执行如何。
status	object	JobStatus 代表作业的当前状态。

11.3.1.1. .spec

描述

JobSpec 描述了作业执行如何。

类型

object

必填

- 模板

属性	类型	描述
----	----	----

属性	类型	描述
activeDeadlineSeconds	整数	指定与 <code>startTime</code> 相对的持续时间（以秒为单位），在系统尝试终止前可以持续激活；值必须是正整数。如果作业被暂停（创建或通过更新），在作业再次恢复时，此计时器将有效停止并重置。
backoffLimit	整数	指定在标记此作业失败前的重试次数。默认值为 6
completionMode	string	<p><code>completionMode</code> 指定 Pod 的完成情况。它可以是 NonIndexed（默认）或 Indexed。</p> <p>NonIndexed 表示当有 <code>.spec.completions</code> 成功完成 Pod 时，作业被视为已完成。每个 Pod 完成都相互不同。</p> <p>indexed 表示作业的 Pod 从 0 变为 <code>(.spec.completions - 1)</code>，在注解 <code>batch.kubernetes.io/job-completion-index</code> 中提供。当每个索引都有一个成功完成 Pod 时，作业被视为已完成。当值被索引时，必须指定 <code>.spec.completions</code>，<code>.spec.parallelism</code> 必须小于或等于 10^5。另外，Pod 名称的格式是 <code>\$(job-name)-\$(index)-\$(random-string)</code>，Pod 主机名采用 <code>\$(job-name)-\$(index)</code> 的形式。</p> <p>以后可以添加更多完成模式。如果作业控制器观察一个没有识别的模式，则在升级过程中会因为版本偏移而可能，控制器会跳过该作业的更新。</p> <p>可能的值有：- "Indexed" 是一个作业完成模式。在这个模式中，作业的 Pod 从 0 变为 <code>(.spec.completions - 1)</code>。当 Pod 完成每个完成索引时，作业被视为已完成。- "NonIndexed" 是一个作业完成模式。在这个模式中，当有 <code>.spec.completions</code> 成功完成 Pod 时，作业被视为已完成。Pod 完成将相互同构。</p>

属性	类型	描述
completions	整数	指定作业应使用的成功完成 pod 数量。设置为 null 表示任何 pod 的成功表示所有 pod 的成功，并允许并行具有任何正值。设置为 1 表示并行性仅限于 1，并且该 pod 的成功表示作业成功。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
manualSelector	布尔值	manualSelector 控制 pod 标签和 pod 选择器的生成。请保留 manualSelector 取消设置，除非您确定您要做什么。为 false 或未设置时，系统会选择此作业的唯一标签，并将这些标签附加到 pod 模板。为 true 时，用户负责选择唯一标签并指定选择器。无法选择唯一标签可能会导致这个和其他作业无法正常工作。但是，在使用旧 extensions/v1beta1 API 创建的作业中可能会看到 manualSelector=true 。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector
parallelism	整数	指定作业应在任意给定时间运行的最大 pod 数量。当 $((\text{.spec.completions} - \text{.status.successful}) < \text{.spec.parallelism})$ 时，以 steady 状态运行的 pod 数量将小于这个数字。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
podFailurePolicy	object	PodFailurePolicy 描述了 pod 失败的 pod 如何影响 backoffLimit。

属性	类型	描述
selector	labelSelector	对 pod 数量应该匹配的 pod 的标签查询。通常，系统会为您设置此字段。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
suspend	布尔值	suspend 指定作业控制器是否应该创建 Pod。如果创建了作业，并将 suspend 设置为 true，则作业控制器不会创建任何 Pod。如果在创建后作业被暂停（例如，标志从 false 变为 true），则作业控制器将删除与此作业关联的所有活跃 Pod。用户必须设计其工作负载才能正常处理此问题。暂停作业将重置作业的 StartTime 字段，从而有效地重置 ActiveDeadlineSeconds 计时器。默认为 false。
模板	PodTemplateSpec	描述在执行作业时创建的 pod。唯一允许的 template.spec.restartPolicy 值为 "Never" 或 "OnFailure"。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/job-run-to-completion/
ttlSecondsAfterFinished	整数	ttlSecondsAfterFinished 限制已完成执行的作业的生命周期(Complete 或 Failed)。如果设置了此字段，在作业完成后，ttlSecondsAfterFinished 会被自动删除。当作业被删除时，其生命周期保证（如终结器）将会被遵守。如果此字段未设置，则作业不会被自动删除。如果此字段设为零，则作业在完成后立即被删除。

11.3.1.2. .spec.podFailurePolicy

描述

PodFailurePolicy 描述了 pod 失败的 pod 如何影响 backoffLimit。

类型

object

必填

- rules

属性	类型	描述
rules	数组	Pod 失败策略规则列表。规则按顺序评估。当规则与 Pod 失败时，会忽略其余的规则。如果没有规则与 Pod 失败匹配，则应用默认处理 - pod 故障计数器会递增，并根据 backoffLimit 检查。最多允许 20 个元素。
rules[]	object	PodFailurePolicyRule 描述了在满足要求时如何处理 pod 失败。每个规则都可以使用 onExitCodes 和 onPodConditions 中的一个。

11.3.1.3. .spec.podFailurePolicy.rules

描述

Pod 失败策略规则列表。规则按顺序评估。当规则与 Pod 失败时，会忽略其余的规则。如果没有规则与 Pod 失败匹配，则应用默认处理 - pod 故障计数器会递增，并根据 backoffLimit 检查。最多允许 20 个元素。

类型

数组

11.3.1.4. .spec.podFailurePolicy.rules[]

描述

PodFailurePolicyRule 描述了在满足要求时如何处理 pod 失败。每个规则都可以使用 onExitCodes 和 onPodConditions 中的一个。

类型

object

必填

- action
- onPodConditions

属性	类型	描述
----	----	----

属性	类型	描述
action	string	<p>指定在满足要求时对 pod 失败执行的操作。可能的值有：</p> <ul style="list-style-type: none"> - FailJob: 表示 pod 的作业标记为 Failed，所有正在运行的 pod 都被终止。 - Ignore: 表示与 .backoffLimit 相关的计数器没有被递增，并创建了替换 pod。 - Count: 表示 pod 以默认处理 - .backoffLimit 的计数器递增。以后会考虑添加其他值。客户端应通过跳过规则来响应未知操作。 <p>possible enum value: - "Count" 是一个在 pod 失败时执行的操作 - pod 失败以默认的方式处理 - pod 失败 - 对应的 .backoffLimit 计数器，由作业的 .status.failed 字段表示。 - "FailJob" is an action, may be 在 pod 失败时 - 将 pod 的作业标记为 Failed 并终止所有正在运行的 pod。 - "Ignore" 是一个在 pod 失败时执行的操作 - .backoffLimit 的计数器。作业的 .status.failed 字段代表不会被递增，并创建了替代 pod。</p>
onExitCodes	object	<p>PodFailurePolicyOnExitCodesRequirement 描述了根据容器退出代码处理失败的 pod 的要求。特别是，它会为每个应用程序容器和 init 容器状态查找 .state.terminated.exitCode，分别由 Pod 状态中的 .status.containerStatuses 和 .status.initContainerStatuses 字段表示。使用成功(exit code 0)完成的容器不包括在要求检查中。</p>
onPodConditions	数组	<p>代表 pod 条件的要求。要求以 pod 条件模式列表表示。如果至少有一个模式匹配实际 pod 条件，则需要满足要求。最多允许 20 个元素。</p>
onPodConditions[]	object	<p>PodFailurePolicyOnPodConditionsPattern 描述与实际 pod 条件类型匹配的模式。</p>

11.3.1.5. .spec.podFailurePolicy.rules[].onExitCodes

描述

PodFailurePolicyOnExitCodesRequirement 描述了根据容器退出代码处理失败的 pod 的要求。特别是，它会为每个应用程序容器和 init 容器状态查找 `.state.terminated.exitCode`，分别由 Pod 状态中的 `.status.containerStatuses` 和 `.status.initContainerStatuses` 字段表示。使用成功(exit code 0)完成的容器不包括在要求检查中。

类型

object

必填

- operator
- values

属性	类型	描述
<code>containerName</code>	string	使用指定名称将退出代码限制为容器的检查。当 null 时，该规则适用于所有容器。指定后，它应当与 pod 模板中的一个容器或 initContainer 名称匹配。
<code>operator</code>	string	表示容器退出代码和指定的值之间的关系。使用成功(exit code 0)完成的容器不包括在要求检查中。可能的值有： - in: 如果至少有一个容器退出代码（如果多个容器没有由"containerName"字段限制）限制了多个容器，则满足要求。- 如果至少一个容器退出代码（如果多个容器不受"containerName"字段限制），则满足要求。以后会考虑添加其他值。客户端应不满足要求来响应未知操作器。 possible enum 值： - "In" - "NotIn"
<code>values</code>	数组（整数）	指定一组值。每个返回的容器退出代码（如果有多个容器）都会针对与 Operator 相关的这组值进行检查。值列表必须排序，且不得包含重复值。值 '0' 不能用于 In operator。至少需要一个元素。最多允许 255 个元素。

11.3.1.6. .spec.podFailurePolicy.rules[].onPodConditions

描述

代表 pod 条件的要求。要求以 pod 条件模式列表表示。如果至少有一个模式匹配实际 pod 条件，则需要满足要求。最多允许 20 个元素。

类型**数组****11.3.1.7. .spec.podFailurePolicy.rules[].onPodConditions[]****描述**

PodFailurePolicyOnPodConditionsPattern 描述与实际 pod 条件类型匹配的模式。

类型**object****必填**

- **type**
- **status**

属性	类型	描述
status	string	指定所需的 Pod 条件状态。要匹配 pod 条件，需要指定状态等于 pod 条件状态。默认值为 True。
type	string	指定所需的 Pod 条件类型。要匹配 pod 条件，需要指定类型等于 pod 条件类型。

11.3.1.8. .status**描述**

JobStatus 代表作业的当前状态。

类型**object**

属性	类型	描述
active	整数	待处理的并运行 pod 的数量。
completedIndexes	string	当 .spec.completionMode = "Indexed" 以文本格式时，completedIndexes 包含已完成的索引。索引以十进制整数表示，用逗号分开。数字以递增的顺序列出。三个或更多连续的数字由系列的第一个和最后一个元素压缩，以连字符分开。例如，如果已完成的索引为 1、3、4、5 和 7，它们表示为 "1,3-5,7"。

属性	类型	描述
completionTime	时间	表示作业完成时的时间。无法保证在单独的操作之间按顺序设置。它以 RFC3339 格式表示，并以 UTC 为单位。只有在作业成功完成时才会设置完成时间。
conditions	数组	对象当前状态的最新可用影响。当作业失败时，其中一个条件将类型为 "Failed" 和 status true。当作业暂停时，其中一个条件为 "Suspended" 和状态 true；当作业恢复时，此条件的状态将变为 false。当作业完成后，其中一个条件将键入 "Complete" 和 status true。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
conditions[]	object	JobCondition 描述了作业的当前状态。
失败	整数	到达阶段失败的 pod 数量。
ready	整数	具有 Ready 条件的 pod 数量。 此字段是 beta 级。当启用功能门 JobReadyPods 时（默认启用）时，作业控制器会填充字段。
startTime	时间	代表作业控制器开始处理作业时的时间。当作业以暂停状态创建时，在第一次恢复前不会设置此字段。每次作业从挂起中恢复时，都会重置此字段。它以 RFC3339 格式表示，并以 UTC 为单位。
succeeded	整数	达到阶段 Succeeded 的 pod 数量。
uncountedTerminatedPods	object	UncountedTerminatedPods 包含已终止但没有在作业状态计数器中考虑的 Pod 的 UID。

11.3.1.9. .status.conditions

描述

对象当前状态的最新可用影响。当作业失败时，其中一个条件将类型为 "Failed" 和 status true。当作业暂停时，其中一个条件为 "Suspended" 和状态 true；当作业恢复时，此条件的状态将变为 false。当

作业完成后，其中一个条件将键入 "Complete" 和 status true。更多信息：

<https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/>

类型

array

11.3.1.10. .status.conditions[]

描述

JobCondition 描述了作业的当前状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastProbeTime	时间	最后一次检查条件的时间。
lastTransitionTime	时间	条件从一个状态传输到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	(brief)条件最后一次转换的原因。
status	字符串	条件的状态， True, False, Unknown 之一。
type	string	作业条件类型、完成或失败。

11.3.1.11. .status.uncountedTerminatedPods

描述

UncountedTerminatedPods 包含已终止但没有在作业状态计数器中考虑的 Pod 的 UID。

类型

object

属性	类型	描述
failed	数组（字符串）	失败包含故障 Pod 的 UID。
succeeded	数组（字符串）	成功包含成功 Pod 的 UID。

11.3.2. API 端点

可用的 API 端点如下：

- **/apis/batch/v1/jobs**
 - **GET**: 列出或监视类型为 Job 的对象
- **/apis/batch/v1/watch/jobs**
 - **GET**: 观察单个对作业列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/batch/v1/namespaces/{namespace}/jobs**
 - **DELETE**：删除作业集合
 - **GET**: 列出或监视类型为 Job 的对象
 - **POST**：创建作业
- **/apis/batch/v1/watch/namespaces/{namespace}/jobs**
 - **GET**: 观察单个对作业列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/batch/v1/namespaces/{namespace}/jobs/{name}**
 - **DELETE**：删除作业
 - **GET**: 读取指定的作业
 - **PATCH**：部分更新指定的作业
 - **PUT**：替换指定的作业
- **/apis/batch/v1/watch/namespaces/{namespace}/jobs/{name}**
 - **GET**: 观察对 kind Job 对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/batch/v1/namespaces/{namespace}/jobs/{name}/status**
 - **GET**: 指定作业的读取状态
 - **PATCH**：部分更新指定作业的状态
 - **PUT**：替换指定作业的状态

11.3.2.1. /apis/batch/v1/jobs

表 11.42. 全局查询参数

参数	类型	描述
----	----	----

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Job 的对象

表 11.43. HTTP 响应

HTTP 代码	响应正文
200 - OK	JobList 模式
401 - 未授权	空

11.3.2.2. /apis/batch/v1/watch/jobs

表 11.44. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个作业列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 11.45. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.3.2.3. /apis/batch/v1/namespaces/{namespace}/jobs

表 11.46. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.47. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除作业集合

表 11.48. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 11.49. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 11.50. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Job 的对象

表 11.51. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 11.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	JobList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建作业

表 11.53. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.54. 主体参数

参数	类型	描述
正文 (body)	Job 模式	

表 11.55. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
201 - Created	Job 模式
202 - Accepted	Job 模式
401 - 未授权	空

11.3.2.4. /apis/batch/v1/watch/namespaces/{namespace}/jobs

表 11.56. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.57. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个作业列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 11.58. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.3.2.5. /apis/batch/v1/namespaces/{namespace}/jobs/{name}

表 11.59. 全局路径参数

参数	类型	描述
name	string	作业名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.60. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除作业

表 11.61. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 11.62. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 11.63. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的作业

表 11.64. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Job</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的作业

表 11.65. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 11.66. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 11.67. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
201 - Created	Job 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的作业

表 11.68. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.69. 主体参数

参数	类型	描述
正文 (body)	Job 模式	

表 11.70. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
201 - Created	Job 模式
401 - 未授权	空

11.3.2.6. /apis/batch/v1/watch/namespaces/{namespace}/jobs/{name}

表 11.71. 全局路径参数

参数	类型	描述
name	string	作业名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 11.72. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Job 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 11.73. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

11.3.2.7. /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status

表 11.74. 全局路径参数

参数	类型	描述
name	string	作业名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 11.75. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定作业的读取状态

表 11.76. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定作业的部分更新状态

表 11.77. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 11.78. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 11.79. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
201 - Created	Job 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定作业的状态

表 11.80. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 11.81. 主体参数

参数	类型	描述
正文 (body)	Job 模式	

表 11.82. HTTP 响应

HTTP 代码	响应正文
200 - OK	Job 模式
201 - Created	Job 模式
401 - 未授权	空

第 12 章 证书 API

12.1. 证书 API

12.1.1. CertificateSigningRequest [certificates.k8s.io/v1]

描述

CertificateSigningRequest 对象提供了一种机制，可通过提交证书签名请求来获取 x509 证书，并使其异步批准并发出。

kubelet 使用此 API 获取：1. 客户端证书向 kube-apiserver 进行身份验证（使用 "kubernetes.io/kube-apiserver-client-kubelet" signerName）。2. 为 TLS 端点提供证书 kube-apiserver 可以安全地连接到（使用 "kubernetes.io/kubelet-serving" signerName）。

此 API 可用于请求客户端证书向 kube-apiserver 进行身份验证（使用 "kubernetes.io/kube-apiserver-client" signerName），或者从自定义非 Kubernetes 签名器获取证书。

类型

对象

12.2. CERTIFICATESIGNINGREQUEST [CERTIFICATES.K8S.IO/V1]

描述

CertificateSigningRequest 对象提供了一种机制，可通过提交证书签名请求来获取 x509 证书，并使其异步批准并发出。

kubelet 使用此 API 获取：1. 客户端证书向 kube-apiserver 进行身份验证（使用 "kubernetes.io/kube-apiserver-client-kubelet" signerName）。2. 为 TLS 端点提供证书 kube-apiserver 可以安全地连接到（使用 "kubernetes.io/kubelet-serving" signerName）。

此 API 可用于请求客户端证书向 kube-apiserver 进行身份验证（使用 "kubernetes.io/kube-apiserver-client" signerName），或者从自定义非 Kubernetes 签名器获取证书。

类型

object

必填

- spec

12.2.1. 规格

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	
spec	对象	CertificateSigningRequestSpec 包含证书请求。
status	对象	CertificateSigningRequestStatus 包含用于指示请求批准/拒绝/失败状态的条件，以及发布的证书。

12.2.1.1. .spec

描述

CertificateSigningRequestSpec 包含证书请求。

类型

object

必填

- **Request** (请求)
- **signerName**

属性	类型	描述
----	----	----

属性	类型	描述
expirationSeconds	整数	<p>expirationSeconds 是签发证书的请求的有效期。证书签名者可能会发布具有不同有效期持续时间的证书，因此客户端必须检查发布证书中的 notBefore 和 notAfter 字段之间的 delta，以确定实际持续时间。</p> <p>只要请求的持续时间不超过 Kubernetes 控制器管理器的 --cluster-signing-duration CLI 标记，则 v1.22+ 的树内实现将遵循这个字段。</p> <p>由于各种原因，证书签名者可能无法遵守此字段：</p> <ol style="list-style-type: none"> 在 v1.22) 2 之前，旧的 signer 不知道字段（如 in-tree 实现）。配置的最大值小于请求持续时间 3 的签名者。配置最小值超过请求持续时间的签名者 <p>expirationSeconds 的有效值为 600，即 10 分钟。</p>
extra	对象	extra 包含创建 CertificateSigningRequest 的用户的额外属性。在创建和不可变时由 API 服务器填充。
extra{}	数组（字符串）	
groups	数组（字符串）	组包含创建 CertificateSigningRequest 的用户的组成员资格。在创建和不可变时由 API 服务器填充。
Request（请求）	字符串	请求包含以 "CERTIFICATE REQUEST" PEM 块中编码的 x509 证书签名请求。当序列化为 JSON 或 YAML 时，数据会额外采用 base64 编码。

属性	类型	描述
signerName	字符串	<p>signerName 表示请求的签名者，是合格的名称。</p> <p>list/watch 对 CertificateSigningRequests 的请求可以使用 "spec.signerName=NAME" fieldSelector 在此字段上过滤。</p> <p>知名的 Kubernetes 符号是：1. "kubernetes.io/kube-apiserver-client": 签发可用于向 kube-apiserver 进行身份验证的客户端证书。对这个 signer 的请求永远不会由 kube-controller-manager 自动批准，由 kube-controller-manager 中的 "csrsigning" 控制器发出。2. "Kubernetes.io/kube-apiserver-client-kubelet": 签发 kubelet 用来向 kube-apiserver 进行身份验证的客户端证书。对这个签名者的请求可由 kube-controller-manager 中的 "csrapproving" 控制器自动批准，并可以被 kube-controller-manager 中的 "csrsigning" 控制器发出。3. "Kubernetes.io/kubelet-serving" 问题为 kubelet 用来为 TLS 端点提供服务的证书，kube-apiserver 可以安全地连接。对这个 signer 的请求永远不会由 kube-controller-manager 自动批准，并可由 kube-controller-manager 中的 "csrsigning" 控制器发出。</p> <p>详情请参考 https://k8s.io/docs/reference/access-authn-authz/certificate-signing-requests/#kubernetes-signers</p> <p>也可以指定自定义 signerNames。签名人定义：1.信任分发：信任 (CA 捆绑包)的分布方式。2.允许的主题：请求禁止的主题时的行为。3.请求中的必需、允许或禁止的 x509 扩展（包括是否允许 subjectAltNames，对允许的值的限制）以及请求禁止的扩展时的行为。4.必需、允许或禁止的密钥使用/扩展密钥使用情况。5.expiration/certificate 生命周</p>

属性	类型	描述
uid	字符串	期：它是否被签名者修复，由管理 证书。 UID 包含创建 CertificateSigningRequest 的用户的 uid。在创建和不可变时由 API 服务器填充。
usages	数组（字符串）	usages 指定在签发的证书中请求的一组密钥用法。 对 TLS 客户端证书的请求通常会请求："数字签名", "key encipherment", "client auth"。 对 TLS 服务证书的请求通常会请求："密钥加密", "数字签名", "server auth"。 有效值为："signing", "digital signature", "content commitment", "content commitment", "key encipherment", "key encipherment", "cert sign", "crl sign", "encipher only", "decipher only", "any", "server auth", "client auth", "code signing", "code signing", "电子邮件保护", "s/mime", "ipsec end system", "ipsec tunnel", "ipsec user", "timestamping", "ocsp signing", "microsoft sgc", "netscape sgc"
username	字符串	username 包含创建 CertificateSigningRequest 的用户的名称。在创建和不可变时由 API 服务器填充。

12.2.1.2. .spec.extra

描述

extra 包含创建 CertificateSigningRequest 的用户的额外属性。在创建和不可变时由 API 服务器填充。

类型

对象

12.2.1.3. .status

描述

CertificateSigningRequestStatus 包含用于指示请求批准/拒绝/失败状态的条件，以及发布的证书。

类型

object

属性	类型	描述
certificate	字符串	<p>在存在 Approved 条件后，证书由 signer 填充。此字段通过 /status 子资源设置。填充后，此字段不可变。</p> <p>如果证书签名请求被拒绝，则会添加类型为 "Denied" 的条件，且此字段为空。如果签名者无法发布证书，则会添加类型为 "Failed" 的条件，且此字段为空。</p> <p>验证要求：1. 证书必须包含一个或多个 PEM 块。2. 所有 PEM 块都必须具有 "CERTIFICATE" 标签，不包含标头，编码的数据必须是 BER 编码的 ASN.1 证书结构，如 RFC5280 节 4 所述。3. 非 PEM 内容可能会在 "CERTIFICATE" PEM 块之前或之后出现，且未验证，以允许解释的文本，如 RFC7468 的第 5.2 节中所述。</p> <p>如果存在多个 PEM 块，并且请求的 spec.signerName 的定义没有指定，则第一个块是签发的证书，后续块应被视为中间证书，并在 TLS 握手中呈现。</p> <p>证书以 PEM 格式编码。</p> <p>当序列化为 JSON 或 YAML 时，数据会额外进行 base64 编码，因此它由以下组成：</p> <pre>base64(-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----)</pre>
conditions	array	应用到请求的条件。已知条件为 "Approved", "Denied" 和 "Failed"。
conditions[]	对象	CertificateSigningRequestCondition 描述了 CertificateSigningRequest 对象的条件

12.2.1.4. .status.conditions

描述

应用到请求的条件。已知条件为 "Approved", "Denied" 和 "Failed"。

类型

array

12.2.1.5. .status.conditions[]

描述

CertificateSigningRequestCondition 描述了 CertificateSigningRequest 对象的条件

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件最后一次从一个状态转换到另一个状态的时间。如果未设置，当添加新条件类型或更改现有条件类型时，服务器会默认使用当前时间。
lastUpdateTime	时间	lastUpdateTime 是最后一次更新到此条件的时间
message	字符串	消息包含人类可读的消息，其中包含有关请求状态的详细信息
reason	字符串	reason 表示请求状态的简短原因
status	字符串	条件的状态，True, False, Unknown 之一。批准、拒绝和失败条件可能不是 "False" 或 "Unknown"。

属性	类型	描述
type	字符串	<p>条件的类型。已知条件为 "Approved", "Denied" 和 "Failed"。</p> <p>通过 /approval 子资源添加 "Approved" 条件，表示请求已批准，并应由签名者发出。</p> <p>通过 /approval 子资源添加 "Denied" 条件，表示请求被拒绝，不应由签名者发出。</p> <p>通过 /status 子资源添加 "Failed" 条件，表示签名者无法发布证书。</p> <p>批准和拒绝条件是互斥的。添加后，无法删除批准、拒绝和失败条件。</p> <p>只允许给定类型的一个条件。</p>

12.2.2. API 端点

可用的 API 端点如下：

- **/apis/certificates.k8s.io/v1/certificatesigningrequests**
 - **DELETE** : 删除 CertificateSigningRequest 的集合
 - **GET**: 列出或监视类型为 CertificateSigningRequest 的对象
 - **POST** : 创建一个 CertificateSigningRequest
- **/apis/certificates.k8s.io/v1/watch/certificatesigningrequests**
 - **GET**: 观察单个对 CertificateSigningRequest. deprecated 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}**
 - **DELETE** : 删除 CertificateSigningRequest
 - **GET** : 读取指定的 CertificateSigningRequest
 - **PATCH**: 部分更新指定的 CertificateSigningRequest
 - **PUT** : 替换指定的 CertificateSigningRequest
- **/apis/certificates.k8s.io/v1/watch/certificatesigningrequests/{name}**
 - **GET**: 观察对类型为 CertificateSigningRequest. deprecated 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status**

- **GET**: 指定的 CertificateSigningRequest 的读取状态
- **PATCH**: 部分更新指定 CertificateSigningRequest 的状态
- **PUT** : 替换指定 CertificateSigningRequest 的状态
- **/apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval**
 - **GET** : 读取指定 CertificateSigningRequest 的批准
 - **PATCH** : 部分更新指定的 CertificateSigningRequest 的批准
 - **PUT** : 替换指定 CertificateSigningRequest 的批准

12.2.2.1. /apis/certificates.k8s.io/v1/certificatesigningrequests

表 12.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CertificateSigningRequest 的集合

表 12.2. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时, 应设置 continue 选项。由于这个值是定义的服务器, 因此客户端只能使用之前查询结果中的 continue 值, 并带有相同的查询参数 (除 continue 值除外), 因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效, 无论是因为过期时间 (通常为 5 到十五分钟) 还是服务器上的配置更改, 服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表, 则必须重启其列表, 而无需 continue 字段。否则, 客户端可能会发送另一个带有 410 错误的令牌的列表请求, 服务器将使用从下一个密钥开始的列表进行响应, 但从最新的快照开始, 从上一个列表结果 (创建、修改或删除) 后, 创建、修改或删除第一个列表请求将包含在响应中, 只要它们的键位于"下一密钥"后。</p> <p>当监视为 true 时, 不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视, 而不错过任何修改。</p>

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 12.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 12.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 CertificateSigningRequest 的对象

表 12.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 12.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequestList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 CertificateSigningRequest

表 12.7. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 12.8. 主体参数

参数	类型	描述
正文 (body)	CertificateSigningRequest 模式	

表 12.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
202 - Accepted	CertificateSigningRequest 模式
401 - Unauthorized	空

12.2.2.2. /apis/certificates.k8s.io/v1/watch/certificatesigningrequests

表 12.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个更改对 CertificateSigningRequest. deprecated 的列表：改为使用 'watch' 参数和 list 操作。

表 12.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

12.2.2.3. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}

表 12.12. 全局路径参数

参数	类型	描述
name	字符串	CertificateSigningRequest 的名称

表 12.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CertificateSigningRequest

表 12.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 12.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 12.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 CertificateSigningRequest

表 12.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 CertificateSigningRequest

表 12.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 12.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 12.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 CertificateSigningRequest

表 12.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 12.22. 主体参数

参数	类型	描述
正文 (body)	CertificateSigningRequest 模式	

表 12.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

12.2.2.4. /apis/certificates.k8s.io/v1/watch/certificatesigningrequests/{name}

表 12.24. 全局路径参数

参数	类型	描述
name	字符串	CertificateSigningRequest 的名称

表 12.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind CertificateSigningRequest. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 12.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

12.2.2.5. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/status

表 12.27. 全局路径参数

参数	类型	描述
name	字符串	CertificateSigningRequest 的名称

表 12.28. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 CertificateSigningRequest 的读取状态

表 12.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 CertificateSigningRequest 的部分更新状态

表 12.30. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 12.31. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 12.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 CertificateSigningRequest 的状态

表 12.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 12.34. 主体参数

参数	类型	描述
正文 (body)	CertificateSigningRequest 模式	

表 12.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

12.2.2.6. /apis/certificates.k8s.io/v1/certificatesigningrequests/{name}/approval

表 12.36. 全局路径参数

参数	类型	描述
name	字符串	CertificateSigningRequest 的名称

表 12.37. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

读取指定 CertificateSigningRequest 的批准

表 12.38. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定 CertificateSigningRequest 的批准

表 12.39. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 12.40. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 12.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 CertificateSigningRequest 的批准

表 12.42. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 12.43. 主体参数

参数	类型	描述
正文 (body)	CertificateSigningRequest 模式	

表 12.44. HTTP 响应

HTTP 代码	响应正文
200 - OK	CertificateSigningRequest 模式
201 - Created	CertificateSigningRequest 模式
401 - Unauthorized	空

第 13 章 协调 API

13.1. 协调 API

13.1.1. lease [coordination.k8s.io/v1]

描述

租期定义了租期概念。

类型

对象

13.2. LEASE [COORDINATION.K8S.IO/V1]

描述

租期定义了租期概念。

类型

对象

13.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	LeaseSpec 是租用的规范。

13.2.1.1. .spec

描述

LeaseSpec 是租用的规范。

类型

object

属性	类型	描述
acquireTime	MicroTime	acquireTime 是获取当前租期的时间。
holderIdentity	字符串	holderIdentity 包含当前租期的拥有者的身份。
leaseDurationSeconds	整数	leaseDurationSeconds 是租期的候选者需要等待强制获取它的时间。这是根据最后观察到的 renewTime 的时间进行测量。
leaseTransitions	整数	leaseTransitions 是拥有者之间租期的转换数量。
renewTime	MicroTime	renewTime 是租期的当前拥有者最近更新租期的时间。

13.2.2. API 端点

可用的 API 端点如下：

- **/apis/coordination.k8s.io/v1/leases**
 - **GET**: 列出或监视类型为 Lease 的对象
- **/apis/coordination.k8s.io/v1/watch/leases**
 - **GET**: 观察单个对 Lease 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/coordination.k8s.io/v1/namespaces/{namespace}/leases**
 - **DELETE** : 删除 Lease 集合
 - **GET**: 列出或监视类型为 Lease 的对象
 - **POST** : 创建一个 Lease
- **/apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases**
 - **GET**: 观察单个对 Lease 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}**
 - **DELETE** : 删除 Lease

- **GET** : 读取指定的 Lease
- **PATCH** : 部分更新指定的 Lease
- **PUT** : 替换指定的 Lease
- **/apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases/{name}**
 - **GET**: 观察对类型为 Lease 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

13.2.2.1. /apis/coordination.k8s.io/v1/leases

表 13.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Lease 的对象

表 13.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	LeaseList 模式
401 - Unauthorized	空

13.2.2.2. /apis/coordination.k8s.io/v1/watch/leases

表 13.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Lease 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 13.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

13.2.2.3. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases

表 13.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 13.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Lease 集合

表 13.7. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 13.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 13.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Lease 的对象

表 13.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 13.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	LeaseList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 Lease

表 13.12. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 13.13. 主体参数

参数	类型	描述
正文 (body)	lease 模式	

表 13.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	lease 模式
201 - Created	lease 模式
202 - Accepted	lease 模式
401 - Unauthorized	空

13.2.2.4. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases

表 13.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 13.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Lease 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 13.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

13.2.2.5. /apis/coordination.k8s.io/v1/namespaces/{namespace}/leases/{name}

表 13.18. 全局路径参数

参数	类型	描述
name	字符串	Lease 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 13.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除一个 Lease

表 13.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 13.21. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 13.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 Lease

表 13.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	lease 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 Lease

表 13.24. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 13.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 13.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	lease 模式
201 - Created	lease 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 Lease

表 13.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 13.28. 主体参数

参数	类型	描述
正文 (body)	lease 模式	

表 13.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	lease 模式
201 - Created	lease 模式
401 - Unauthorized	空

13.2.2.6. /apis/coordination.k8s.io/v1/watch/namespaces/{namespace}/leases/{name}

表 13.30. 全局路径参数

参数	类型	描述
name	字符串	Lease 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 13.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Lease. deprecated 的对象更改：使用 'watch' 参数和 list 操作，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 13.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 14 章 核心 API

14.1. 核心 API

14.1.1. 绑定 [v1]

描述

绑定将一个对象绑定到另一个对象；例如，pod 由调度程序绑定到节点。在 1.7 中弃用了，请使用 pods 的 bindings 子资源。

类型

对象

14.1.2. ComponentStatus [v1]

描述

ComponentStatus（和 ComponentStatusList）包含集群验证信息。弃用：此 API 在 v1.19+ 中弃用

类型

对象

14.1.3. ConfigMap [v1]

描述

ConfigMap 包含要使用的 pod 的配置数据。

类型

对象

14.1.4. Endpoints [v1]

描述

端点是实现实际服务的端点集合。Example:

```
Name: "mysvc",
Subsets: [
  {
    Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
    Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
  },
  {
    Addresses: [{"ip": "10.10.3.3"}],
    Ports: [{"name": "a", "port": 93}, {"name": "b", "port": 76}]
  },
]
```

类型

对象

14.1.5. Event [v1]

描述

事件是集群中某一位置的事件报告。事件具有有限的保留时间和触发器，消息可能会随时间而演进。事件消费者不应依赖给定 Reason 反映一致的底层触发器的事件的时间，或者该 Reason 持续存在事件的时间。事件应被视为 informative, best-effort, supplemental 数据。

类型

对象

14.1.6. LimitRange [v1]

描述

LimitRange 为命名空间中的每种资源设置资源使用量限制。

类型

对象

14.1.7. 命名空间 [v1]

描述

namespace 为 Names 提供范围。使用多个命名空间是可选的。

类型

对象

14.1.8. 节点 [v1]

描述

节点是 Kubernetes 中的 worker 节点。每个节点在缓存中都有一个唯一标识符（例如在 etcd 中）。

类型

object

14.1.9. PersistentVolume [v1]

描述

PersistentVolume (PV)是管理员置备的存储资源。它类似于节点。更多信息：
<https://kubernetes.io/docs/concepts/storage/persistent-volumes>

类型

object

14.1.10. PersistentVolumeClaim [v1]

描述

PersistentVolumeClaim 是用户对持久性卷的请求并声明

类型

object

14.1.11. Pod [v1]

描述

Pod 是可以在主机上运行的容器的集合。此资源由客户端创建，并调度到主机上。

类型

object

14.1.12. PodTemplate [v1]

描述

PodTemplate 描述用于创建预定义 pod 副本的模板。

类型

object

14.1.13. ReplicationController [v1]

描述

ReplicationController 代表复制控制器的配置。

类型

对象

14.1.14. ResourceQuota [v1]

描述

ResourceQuota 设置每个命名空间强制的聚合配额限制

类型

对象

14.1.15. Secret [v1]

描述

secret 包含特定类型的 secret 数据。Data 字段中值的总字节数必须小于 MaxSecretSize 字节。

类型

对象

14.1.16. Service [v1]

描述

服务是软件服务的命名抽象（例如，mysql）由本地端口（如 3306）组成，以及代理侦听的选择器，决定哪些 pod 将回答通过代理发送的请求。

类型

对象

14.1.17. ServiceAccount [v1]

描述

ServiceAccount 将一个名称绑定在一起：一个名称，用户理解，可能由外部系统理解，对于一个身份，可以验证并授权了一组 secret 的主体。

类型

对象

14.2. 绑定 [V1]

描述

绑定将一个对象绑定到另一个对象；例如，pod 由调度程序绑定到节点。在 1.7 中弃用了，请使用 pods 的 bindings 子资源。

类型

object

必填

- **target**

14.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
target	对象	ObjectReference 包含足够信息供您检查或修改引用的对象。

14.2.1.1. .target

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 <code>resourceVersion</code> （如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.2.2. API 端点

可用的 API 端点如下：

- **/api/v1/namespaces/{namespace}/bindings**
 - **POST** : 创建绑定
- **/api/v1/namespaces/{namespace}/pods/{name}/binding**
 - **POST** : 创建 Pod 的绑定

14.2.2.1. /api/v1/namespaces/{namespace}/bindings

表 14.1. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.2. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建绑定

表 14.3. 主体参数

参数	类型	描述
正文 (body)	binding 模式	

表 14.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	binding 模式
201 - Created	binding 模式
202 - Accepted	binding 模式
401 - Unauthorized	空

14.2.2.2. /api/v1/namespaces/{namespace}/pods/{name}/binding

表 14.5. 全局路径参数

参数	类型	描述
name	字符串	绑定的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.6. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建 Pod 的绑定

表 14.7. 主体参数

参数	类型	描述
正文 (body)	binding 模式	

表 14.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	binding 模式
201 - Created	binding 模式
202 - Accepted	binding 模式
401 - Unauthorized	空

14.3. COMPONENTSTATUS [V1]

描述

ComponentStatus (和 ComponentStatusList) 包含集群验证信息。弃用：此 API 在 v1.19+ 中弃用

类型

对象

14.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
conditions	array	观察到的组件条件列表
conditions[]	对象	有关组件条件的信息。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

14.3.1.1. .conditions

描述

观察到的组件条件列表

类型

array

14.3.1.2. .conditions[]

描述

有关组件条件的信息。

类型

object

必填

- **type**

- **status**

属性	类型	描述
错误	字符串	组件的错误代码。例如，健康检查错误代码。
message	字符串	有关组件条件的信息。例如，关于健康检查的信息。
status	字符串	组件条件的状态。"Healthy": "True", "False", 或 "Unknown" 的有效值。
type	字符串	组件的条件类型。有效值: "Healthy"

14.3.2. API 端点

可用的 API 端点如下：

- **/api/v1/componentstatuses**
 - **GET**: 列出 kind ComponentStatus 的对象
- **/api/v1/componentstatuses/{name}**
 - **GET**: 读取指定的 ComponentStatus

14.3.2.1. /api/v1/componentstatuses

表 14.9. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出 kind componentStatus 的对象

表 14.10. HTTP 响应

HTTP 代码	响应正文
200 - OK	ComponentStatusList 模式
401 - Unauthorized	空

14.3.2.2. /api/v1/componentstatuses/{name}

表 14.11. 全局路径参数

参数	类型	描述
name	字符串	组件状态的名称

表 14.12. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

读取指定的 componentStatus

表 14.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	ComponentStatus 模式
401 - Unauthorized	空

14.4. CONFIGMAP [V1]

描述

ConfigMap 包含要使用的 pod 的配置数据。

类型

对象

14.4.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
binaryData	对象 (字符串)	binaryData 包含二进制数据。每个键都必须包含字母数字字符、'-'、'_' 或 '.'。binaryData 可以包含不在 UTF-8 范围内的字节序列。存储在 BinaryData 中的密钥不得与 Data 字段中的重叠，这在验证过程中强制执行。使用此字段将需要 1.10+ apiserver 和 kubelet。
data	对象 (字符串)	数据包含配置数据。每个键都必须包含字母数字字符、'-'、'_' 或 '.'。具有非 UTF-8 字节序列的值必须使用 BinaryData 字段。存储在 Data 中的密钥不得与 BinaryData 字段中的密钥重叠，这在验证过程中强制执行。
immutable	布尔值	immutable, 如果设为 true, 则确保 ConfigMap 中存储的数据无法更新 (只能修改对象元数据)。如果没有设置 true, 则可以随时修改该字段。默认为 nil。
kind	字符串	kind 是一个字符串值, 代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

14.4.2. API 端点

可用的 API 端点如下：

- **/api/v1/configmaps**
 - **GET**: 列出或监视类型为 ConfigMap 的对象
- **/api/v1/watch/configmaps**
 - **GET**: 观察单个对 ConfigMap 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/configmaps**
 - **DELETE** : 删除 ConfigMap 的集合
 - **GET**: 列出或监视类型为 ConfigMap 的对象
 - **POST** : 创建 ConfigMap
- **/api/v1/watch/namespaces/{namespace}/configmaps**
 - **GET**: 观察单个对 ConfigMap 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/configmaps/{name}**
 - **DELETE** : 删除 ConfigMap
 - **GET**: 读取指定的 ConfigMap
 - **PATCH** : 部分更新指定的 ConfigMap
 - **PUT** : 替换指定的 ConfigMap
- **/api/v1/watch/namespaces/{namespace}/configmaps/{name}**
 - **GET**: 观察对类型为 ConfigMap 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

14.4.2.1. /api/v1/configmaps

表 14.14. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 ConfigMap 的对象

表 14.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	ConfigMapList 模式
401 - Unauthorized	空

14.4.2.2. /api/v1/watch/configmaps

表 14.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ConfigMap 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.4.2.3. /api/v1/namespaces/{namespace}/configmaps

表 14.18. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ConfigMap 集合

表 14.20. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.21. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ConfigMap 的对象

表 14.23. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.24. HTTP 响应

HTTP 代码	响应正文
200 - OK	ConfigMapList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 ConfigMap

表 14.25. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.26. 主体参数

参数	类型	描述
正文 (body)	ConfigMap 模式	

表 14.27. HTTP 响应

HTTP 代码	响应正文
200 - OK	ConfigMap 模式
201 - Created	ConfigMap 模式
202 - Accepted	ConfigMap 模式
401 - Unauthorized	空

14.4.2.4. /api/v1/watch/namespaces/{namespace}/configmaps

表 14.28. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.29. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ConfigMap 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.30. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.4.2.5. /api/v1/namespaces/{namespace}/configmaps/{name}

表 14.31. 全局路径参数

参数	类型	描述
name	字符串	ConfigMap 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.32. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ConfigMap

表 14.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.34. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 ConfigMap

表 14.36. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ConfigMap</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 ConfigMap

表 14.37. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.38. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.39. HTTP 响应

HTTP 代码	响应正文
200 - OK	ConfigMap 模式
201 - Created	ConfigMap 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ConfigMap

表 14.40. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.41. 主体参数

参数	类型	描述
正文 (body)	ConfigMap 模式	

表 14.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	ConfigMap 模式
201 - Created	ConfigMap 模式
401 - Unauthorized	空

14.4.2.6. /api/v1/watch/namespaces/{namespace}/configmaps/{name}

表 14.43. 全局路径参数

参数	类型	描述
name	字符串	ConfigMap 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.44. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 ConfigMap 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.45. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.5. ENDPOINTS [V1]

描述

端点是实现实际服务的端点集合。Example:

```

Name: "mysvc",
Subsets: [
  {
    Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
    Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
  },
  {
    Addresses: [{"ip": "10.10.3.3"}],
    Ports: [{"name": "a", "port": 93}, {"name": "b", "port": 76}]
  },
]

```

类型

对象

14.5.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
subsets	array	所有端点的集合都是所有子集的合并。地址按照其共享的 IP 放入子集中。具有多个端口的单个地址，其中一些已就绪，并且其中一些未就绪（因为它们来自不同的容器）将导致地址显示在不同端口的不同子集中。没有地址出现在同一子集中的 Addresses 和 NotReadyAddresses 中。组成服务的地址和端口集合。
subsets[]	对象	EndpointSubset 是一组带有一组常用端口的地址。扩展的端点集合是 Cartesian 产品，即 Addresses x Ports。例如，给定： <pre>{ address: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}], Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]}</pre> 生成的端点集可视为： <pre>答： [10.10.1.1:8675, 10.10.2.2:8675], b: [10.10.1.1:309, 10.10.2.2:309]</pre>

14.5.1.1. .subsets

描述

所有端点的集合都是所有子集的合并。地址按照其共享的 IP 放入子集中。具有多个端口的单个地址，其中一些已就绪，并且其中一些未就绪（因为它们来自不同的容器）将导致地址显示在不同端口的不同子集中。没有地址出现在同一子集中的 Addresses 和 NotReadyAddresses 中。组成服务的地址和端口集合。

类型

array

14.5.1.2. .subsets[]

描述

EndpointSubset 是一组带有一组常用端口的地址。扩展的端点集合是 Cartesian 产品，即 Addresses x Ports。例如，给定：

```
{
  Addresses: [{"ip": "10.10.1.1"}, {"ip": "10.10.2.2"}],
  Ports: [{"name": "a", "port": 8675}, {"name": "b", "port": 309}]
}
```

生成的端点集可视为：

```
a: [ 10.10.1.1:8675, 10.10.2.2:8675 ],
b: [ 10.10.1.1:309, 10.10.2.2:309 ]
```

类型**object**

属性	类型	描述
addresses	array	提供标记为就绪的相关端口的 IP 地址。这些端点应该被视为安全供负载均衡器和客户端使用。
addresses[]	对象	EndpointAddress 是一个描述单个 IP 地址的元组。
notReadyAddresses	array	提供相关端口但当前未标记为就绪的 IP 地址，因为它们尚未开始，最近失败了就绪度检查，或者最近失败了存活度检查。
notReadyAddresses[]	对象	EndpointAddress 是一个描述单个 IP 地址的元组。
ports	array	相关 IP 地址上可用的端口号。
ports[]	对象	EndpointPort 是一个描述单个端口的元组。

14.5.1.3. .subsets[].addresses**描述**

提供标记为就绪的相关端口的 IP 地址。这些端点应该被视为安全供负载均衡器和客户端使用。

类型**array****14.5.1.4. .subsets[].addresses[]****描述**

EndpointAddress 是一个描述单个 IP 地址的元组。

类型**object****必填**

- ip

属性	类型	描述
hostname	字符串	此端点的主机名
ip	字符串	此端点的 IP。可能不是 loopback (127.0.0.0/8 或 ::1), link-local (169.254.0.0/16 or fe80::/10) 或 link-local multicast (224.0.0.0/24 或 ff02::/16)。
nodeName	字符串	可选：托管此端点的节点。这可用于确定节点本地的端点。
targetRef	对象	ObjectReference 包含足够信息供您检查或修改引用的对象。

14.5.1.5. .subsets[].addresses[].targetRef

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.5.1.6. .subsets[].notReadyAddresses

描述

提供相关端口但当前未标记为就绪的 IP 地址，因为它们尚未开始，最近失败了就绪度检查，或者最近失败了存活度检查。

类型

array

14.5.1.7. .subsets[].notReadyAddresses[]

描述

EndpointAddress 是一个描述单个 IP 地址的元组。

类型

object

必填

- ip

属性	类型	描述
hostname	字符串	此端点的主机名
ip	字符串	此端点的 IP。可能不是 loopback (127.0.0.0/8 或 ::1), link-local (169.254.0.0/16 or fe80::/10) 或 link-local multicast (224.0.0.0/24 或 ff02::/16)。
nodeName	字符串	可选：托管此端点的节点。这可用于确定节点本地的端点。
targetRef	对象	ObjectReference 包含足够信息供您检查或修改引用的对象。

14.5.1.8. .subsets[].notReadyAddresses[].targetRef

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.5.1.9. .subsets[].ports

描述

相关 IP 地址上可用的端口号。

类型

array

14.5.1.10. .subsets[].ports[]

描述

EndpointPort 是一个描述单个端口的元组。

类型

object

必填

- **port**

属性	类型	描述
----	----	----

属性	类型	描述
appProtocol	字符串	<p>此端口的应用程序协议。这用作实现的提示，以便为他们理解的协议提供更丰富的行为。此字段遵循标准 Kubernetes 标签语法。有效值为：</p> <ul style="list-style-type: none"> * 无前缀协议名称 - 为 IANA 标准服务名称保留（根据 RFC-6335 和 https://www.iana.org/assignments/service-names保留）。 * Kubernetes-defined prefixed name: * 'kubernetes.io/h2c' - HTTP/2 over cleartext, 如 https://www.rfc-editor.org/rfc/rfc7540所述 * 其他协议应使用实现定义的前缀名称，如 mycompany.com/my-custom-protocol。
name	字符串	此端口的名称。这必须与对应 ServicePort 中的 'name' 字段匹配。必须是 DNS_LABEL。仅在定义一个端口时可选。
port	整数	端点的端口号。
protocol	字符串	<p>此端口的 IP 协议。必须是 UDP、TCP 或 SCTP。默认为 TCP。</p> <p>可能的值有： - "SCTP" 是 SCTP 协议。 - "TCP" 是 TCP 协议。 - "UDP" 是 UDP 协议。</p>

14.5.2. API 端点

可用的 API 端点如下：

- **/api/v1/endpoints**
 - **GET**: 列出或监视类型为 Endpoints 的对象
- **/api/v1/watch/endpoints**
 - **GET**: 观察单个对 Endpoints 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/endpoints**
 - **DELETE** : 删除端点的集合

- **GET**: 列出或监视类型为 Endpoints 的对象
- **POST** : 创建端点
- **/api/v1/watch/namespaces/{namespace}/endpoints**
 - **GET**: 观察单个对 Endpoints 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/endpoints/{name}**
 - **DELETE** : 删除端点
 - **GET** : 读取指定的端点
 - **PATCH** : 部分更新指定的端点
 - **PUT** : 替换指定的端点
- **/api/v1/watch/namespaces/{namespace}/endpoints/{name}**
 - **GET**: 观察对类型为 Endpoints 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

14.5.2.1. /api/v1/endpoints

表 14.46. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Endpoints 的对象

表 14.47. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointsList 模式
401 - Unauthorized	空

14.5.2.2. /api/v1/watch/endpoints

表 14.48. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Endpoints 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.5.2.3. /api/v1/namespaces/{namespace}/endpoints

表 14.50. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.51. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE**描述**

删除端点集合

表 14.52. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.53. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.54. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Endpoints 的对象

表 14.55. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.56. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointsList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建端点

表 14.57. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.58. 主体参数

参数	类型	描述
正文 (body)	Endpoints 模式	

表 14.59. HTTP 响应

HTTP 代码	响应正文
200 - OK	Endpoints 模式
201 - Created	Endpoints 模式
202 - Accepted	Endpoints 模式
401 - Unauthorized	空

14.5.2.4. /api/v1/watch/namespaces/{namespace}/endpoints

表 14.60. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.61. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Endpoints 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.62. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.5.2.5. /api/v1/namespaces/{namespace}/endpoints/{name}

表 14.63. 全局路径参数

参数	类型	描述
name	字符串	端点的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.64. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除端点

表 14.65. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.66. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.67. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的端点

表 14.68. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Endpoints</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的端点

表 14.69. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.70. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.71. HTTP 响应

HTTP 代码	响应正文
200 - OK	Endpoints 模式
201 - Created	Endpoints 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的端点

表 14.72. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.73. 主体参数

参数	类型	描述
正文 (body)	Endpoints 模式	

表 14.74. HTTP 响应

HTTP 代码	响应正文
200 - OK	Endpoints 模式
201 - Created	Endpoints 模式
401 - Unauthorized	空

14.5.2.6. /api/v1/watch/namespaces/{namespace}/endpoints/{name}

表 14.75. 全局路径参数

参数	类型	描述
name	字符串	端点的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.76. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Endpoints 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.77. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.6. EVENT [V1]

描述

事件是集群中某一位置的事件报告。事件具有有限的保留时间和触发器，消息可能会随时间而演进。事件消费者不应依赖给定 Reason 反映一致的底层触发器的事件的时间，或者该 Reason 持续存在事件的时间。事件应被视为 informative, best-effort, supplemental 数据。

类型

object

必填

- **metadata**
- **involvedObject**

14.6.1. 规格

属性	类型	描述
action	字符串	关于 Regarding 对象采取哪些操作/失败。
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
数量	整数	发生此事件的次数。
eventTime	MicroTime	首次观察此事件的时间。
firstTimestamp	时间	事件首次记录的时间。（服务器接收的时间为 TypeMeta.）
involvedObject	对象	ObjectReference 包含足够信息供您检查或修改引用的对象。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
lastTimestamp	时间	记录此事件最近发生的时间。

属性	类型	描述
message	字符串	此操作状态的人类可读描述。
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
reason	字符串	这应该是一个简短的、可理解的字符串，使转换至对象的当前状态的原因。
相关	对象	ObjectReference 包含足够信息供您检查或修改引用的对象。
reportingComponent	字符串	发送此事件的控制器的名称，如 kubernetes.io/kubelet 。
reportingInstance	字符串	控制器实例的 ID，如 kubelet-xyzf 。
系列	对象	EventSeries 包含有关一系列事件的信息，即在一段时间内持续发生/发生的内容。
source	对象	EventSource 包含事件的信息。
type	字符串	此事件类型(Normal, Warning)，将来可能会添加新类型

14.6.1.1. .involvedObject

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。

属性	类型	描述
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 <code>resourceVersion</code> （如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.6.1.2. .related

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 <code>resourceVersion</code> （如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.6.1.3. .series

描述

EventSeries 包含有关一系列事件的信息，即在一段时间内持续发生/发生的内容。

类型

object

属性	类型	描述
数量	整数	此系列中发生次数最高到最后的心跳时间
lastObservedTime	MicroTime	最后一次观察到的时间

14.6.1.4. .source

描述

EventSource 包含事件的信息。

类型

object

属性	类型	描述
component	字符串	生成事件的组件。
主机	字符串	生成事件的节点名称。

14.6.2. API 端点

可用的 API 端点如下：

- **/api/v1/events**
 - **GET**: 列出或监视类型为 Event 的对象
- **/api/v1/watch/events**
 - **GET**: 观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/events**
 - **DELETE**：删除事件集合
 - **GET**: 列出或监视类型为 Event 的对象
 - **POST**：创建事件
- **/api/v1/watch/namespaces/{namespace}/events**
 - **GET**: 观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/events/{name}**

- **DELETE** : 删除事件
- **GET** : 读取指定的事件
- **PATCH** : 部分更新指定的事件
- **PUT** : 替换指定的事件
- **/api/v1/watch/namespaces/{namespace}/events/{name}**
 - **GET**: 观察对类型为 Event 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

14.6.2.1. /api/v1/events

表 14.78. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签, 由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回, 也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视, 则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时, 应设置 continue 选项。由于这个值是定义的服务器, 因此客户端只能使用之前查询结果中的 continue 值, 并带有相同的查询参数 (除 continue 值除外), 因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效, 无论是因为过期时间 (通常为 5 到十五分钟) 还是服务器上的配置更改, 服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表, 则必须重启其列表, 而无需 continue 字段。否则, 客户端可能会发送另一个带有 410 错误的令牌的列表请求, 服务器将使用从下一个密钥开始的列表进行响应, 但从最新的快照开始, 从上一个列表结果 (创建、修改或删除) 后, 创建、修改或删除第一个列表请求将包含在响应中, 只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时, 不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视, 而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器, 用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Event 的对象

表 14.79. HTTP 响应

HTTP 代码	响应正文
200 - OK	EventList 模式
401 - Unauthorized	空

14.6.2.2. /api/v1/watch/events

表 14.80. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 'resourceVersion' 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.81. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.6.2.3. /api/v1/namespaces/{namespace}/events

表 14.82. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.83. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除事件集合

表 14.84. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.85. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.86. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Event 的对象

表 14.87. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.88. HTTP 响应

HTTP 代码	响应正文
200 - OK	EventList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建事件

表 14.89. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.90. 主体参数

参数	类型	描述
正文 (body)	event 模式	

表 14.91. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
202 - Accepted	event 模式
401 - Unauthorized	空

14.6.2.4. /api/v1/watch/namespaces/{namespace}/events

表 14.92. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.93. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.94. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.6.2.5. /api/v1/namespaces/{namespace}/events/{name}

表 14.95. 全局路径参数

参数	类型	描述
name	字符串	事件名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.96. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除事件

表 14.97. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.98. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.99. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的事件

表 14.100. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>event</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的事件

表 14.101. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.102. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.103. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的事件

表 14.104. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.105. 主体参数

参数	类型	描述
正文 (body)	event 模式	

表 14.106. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
401 - Unauthorized	空

14.6.2.6. /api/v1/watch/namespaces/{namespace}/events/{name}

表 14.107. 全局路径参数

参数	类型	描述
name	字符串	事件名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.108. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Event.deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.109. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.7. LIMITRANGE [V1]

描述

LimitRange 为命名空间中的每种资源设置资源使用量限制。

类型

对象

14.7.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	LimitRangeSpec 为与 kind 上匹配的资源定义 min/max 用量限制。

14.7.1.1. .spec

描述

LimitRangeSpec 为与 kind 上匹配的资源定义 min/max 用量限制。

类型

object

必填

- **limits**

属性	类型	描述
----	----	----

属性	类型	描述
limits	array	限制是强制的 LimitRangeItem 对象列表。
limits[]	对象	LimitRangeItem 为 kind 上匹配的任何资源定义一个 min/max 用量限制。

14.7.1.2. .spec.limits

描述

限制是强制的 LimitRangeItem 对象列表。

类型

array

14.7.1.3. .spec.limits[]

描述

LimitRangeItem 为 kind 上匹配的任何资源定义一个 min/max 用量限制。

类型

object

必填

- **type**

属性	类型	描述
default	对象 (数量)	如果省略了资源限值，则资源名称的默认资源要求限制值。
defaultRequest	对象 (数量)	如果省略了资源请求，defaultRequest 是由资源名称的默认资源要求请求值。
max	对象 (数量)	按资源名称对此类型的最大使用限制。
maxLimitRequestRatio	对象 (数量)	如果指定了 maxLimitRequestRatio，则指定资源必须具有非零的请求和限制，其中限制除请求小于或等于枚举的值；这代表指定资源的最大突发。
分钟	对象 (数量)	根据资源名称，对这个类型的最小使用限制。

属性	类型	描述
type	字符串	此限制应用到的资源类型。

14.7.2. API 端点

可用的 API 端点如下：

- **/api/v1/limitranges**
 - **GET**: 列出或监视类型为 LimitRange 的对象
- **/api/v1/watch/limitranges**
 - **GET**: 观察单个对 LimitRange 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/limitranges**
 - **DELETE**：删除 LimitRange 集合
 - **GET**: 列出或监视类型为 LimitRange 的对象
 - **POST**：创建一个 LimitRange
- **/api/v1/watch/namespaces/{namespace}/limitranges**
 - **GET**: 观察单个对 LimitRange 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/limitranges/{name}**
 - **DELETE**：删除 LimitRange
 - **GET**：读取指定的 LimitRange
 - **PATCH**：部分更新指定的 LimitRange
 - **PUT**：替换指定的 LimitRange
- **/api/v1/watch/namespaces/{namespace}/limitranges/{name}**
 - **GET**: 观察对类型为 LimitRange 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

14.7.2.1. /api/v1/limitranges

表 14.110. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion"，当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 LimitRange 类型的对象

表 14.111. HTTP 响应

HTTP 代码	响应正文
200 - OK	LimitRangeList 模式
401 - Unauthorized	空

14.7.2.2. /api/v1/watch/limitranges

表 14.112. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 LimitRange 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.113. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.7.2.3. /api/v1/namespaces/{namespace}/limitranges

表 14.114. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.115. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 LimitRange 集合

表 14.116. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.117. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.118. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 LimitRange 类型的对象

表 14.119. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.120. HTTP 响应

HTTP 代码	响应正文
200 - OK	LimitRangeList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 LimitRange

表 14.121. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.122. 主体参数

参数	类型	描述
正文 (body)	LimitRange 模式	

表 14.123. HTTP 响应

HTTP 代码	响应正文
200 - OK	LimitRange 模式
201 - Created	LimitRange 模式
202 - Accepted	LimitRange 模式
401 - Unauthorized	空

14.7.2.4. /api/v1/watch/namespaces/{namespace}/limitranges

表 14.124. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.125. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 LimitRange 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.126. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.7.2.5. /api/v1/namespaces/{namespace}/limitranges/{name}

表 14.127. 全局路径参数

参数	类型	描述
name	字符串	LimitRange 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.128. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 LimitRange

表 14.129. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.130. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.131. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 `LimitRange`

表 14.132. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>LimitRange</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `LimitRange`

表 14.133. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.134. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.135. HTTP 响应

HTTP 代码	响应正文
200 - OK	LimitRange 模式
201 - Created	LimitRange 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 LimitRange

表 14.136. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.137. 主体参数

参数	类型	描述
正文 (body)	LimitRange 模式	

表 14.138. HTTP 响应

HTTP 代码	响应正文
200 - OK	LimitRange 模式
201 - Created	LimitRange 模式
401 - Unauthorized	空

14.7.2.6. /api/v1/watch/namespaces/{namespace}/limitranges/{name}

表 14.139. 全局路径参数

参数	类型	描述
name	字符串	LimitRange 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.140. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 LimitRange 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.141. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.8. 命名空间 [V1]

描述

namespace 为 Names 提供范围。使用多个命名空间是可选的。

类型

对象

14.8.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	namespaceSpec 描述 Namespace 上的属性。
status	对象	NamespaceStatus 是命名空间当前状态的信息。

14.8.1.1. .spec

描述

namespaceSpec 描述 Namespace 上的属性。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
finalizers	数组（字符串）	finalizers 是不透明的值列表，必须为空才能从存储中永久删除对象。 更多信息： https://kubernetes.io/docs/tasks/administer-cluster/namespaces/

14.8.1.2. .status

描述

NamespaceStatus 是命名空间当前状态的信息。

类型

object

属性	类型	描述
conditions	array	代表命名空间当前状态的最新可用影响。
conditions[]	对象	NamespaceCondition 包含有关命名空间状态的详细信息。
phase	字符串	phase 是命名空间的当前生命周期阶段。更多信息： https://kubernetes.io/docs/tasks/administer-cluster/namespaces/ possible enum value: - "Active" 表示命名空间可用于 system - "Terminating" 表示命名空间正在安全终止

14.8.1.3. .status.conditions

描述

代表命名空间当前状态的最新可用影响。

类型

array

14.8.1.4. .status.conditions[]

描述

NamespaceCondition 包含有关命名空间状态的详细信息。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	
message	字符串	
reason	字符串	
status	字符串	条件的状态， True, False, Unknown 之一。
type	字符串	命名空间控制器条件的类型。

14.8.2. API 端点

可用的 API 端点如下：

- **/api/v1/namespaces**
 - **GET**: 列出或监视类型为 Namespace 的对象
 - **POST** : 创建一个命名空间
- **/api/v1/watch/namespaces**
 - **GET**: 观察单个对命名空间列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{name}**
 - **DELETE** : 删除命名空间
 - **GET**: 读取指定的命名空间
 - **PATCH** : 部分更新指定的命名空间
 - **PUT** : 替换指定的命名空间
- **/api/v1/watch/namespaces/{name}**
 - **GET**: 观察对类型为 Namespace 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{name}/status**
 - **GET**: 指定命名空间的读取状态
 - **PATCH** : 部分更新指定命名空间的状态
 - **PUT** : 替换指定命名空间的状态

- `/api/v1/namespaces/{name}/finalize`
 - **PUT** : 替换指定命名空间的完成

14.8.2.1. /api/v1/namespaces

表 14.142. 全局查询参数

参数	类型	描述
<code>pretty</code>	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

列出或监视类型为 Namespace 的对象

表 14.143. 查询参数

参数	类型	描述
<code>allowWatchBookmarks</code>	布尔值	<code>allowWatchBookmarks</code> 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 <code>continue</code> 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 <code>continue</code> 值，并带有相同的查询参数（除 <code>continue</code> 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 <code>continue</code> 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 <code>ResourceExpired</code> 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 <code>true</code> 时，不支持此字段。客户端可以从服务器返回的最后一个 <code>resourceVersion</code> 值开始监视，而不错过任何修改。</p>
<code>fieldSelector</code>	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.144. HTTP 响应

HTTP 代码	响应正文
200 - OK	NamespaceList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个命名空间

表 14.145. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.146. 主体参数

参数	类型	描述
正文 (body)	Namespace 模式	

表 14.147. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
202 - Accepted	Namespace 模式
401 - Unauthorized	空

14.8.2.2. /api/v1/watch/namespaces

表 14.148. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对命名空间列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.149. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.8.2.3. /api/v1/namespaces/{name}

表 14.150. 全局路径参数

参数	类型	描述
name	字符串	命名空间的名称

表 14.151. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除命名空间

表 14.152. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.153. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.154. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的命名空间

表 14.155. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的命名空间

表 14.156. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.157. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.158. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的命名空间

表 14.159. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.160. 主体参数

参数	类型	描述
正文 (body)	Namespace 模式	

表 14.161. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
401 - Unauthorized	空

14.8.2.4. /api/v1/watch/namespaces/{name}

表 14.162. 全局路径参数

参数	类型	描述
name	字符串	命名空间的名称

表 14.163. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

监视对类型为 Namespace 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.164. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.8.2.5. /api/v1/namespaces/{name}/status

表 14.165. 全局路径参数

参数	类型	描述
name	字符串	命名空间的名称

表 14.166. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

指定命名空间的读取状态

表 14.167. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

指定命名空间的部分更新状态

表 14.168. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.169. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.170. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定命名空间的状态

表 14.171. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.172. 主体参数

参数	类型	描述
正文 (body)	Namespace 模式	

表 14.173. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
401 - Unauthorized	空

14.8.2.6. /api/v1/namespaces/{name}/finalize

表 14.174. 全局路径参数

参数	类型	描述
name	字符串	命名空间的名称

表 14.175. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

PUT

描述

替换指定命名空间的完成

表 14.176. 主体参数

参数	类型	描述
正文 (body)	Namespace 模式	

表 14.177. HTTP 响应

HTTP 代码	响应正文
200 - OK	Namespace 模式
201 - Created	Namespace 模式
401 - Unauthorized	空

14.9. 节点 [V1]

描述

节点是 Kubernetes 中的 worker 节点。每个节点在缓存中都有一个唯一标识符（例如在 etcd 中）。

类型

对象

14.9.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	nodeSpec 描述了节点创建的属性。
status	对象	NodeStatus 是节点当前状态的信息。

14.9.1.1. .spec

描述

nodeSpec 描述了节点创建的属性。

类型

object

属性	类型	描述
configSource	对象	NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用
externalID	字符串	已弃用。并非所有 kubelet 将设置此字段。1.13 后删除字段： https://issues.k8s.io/61966
podCIDR	字符串	PodCIDR 代表分配给节点的 pod IP 范围。
podCIDRs	数组（字符串）	podCIDRs 代表分配给节点的 IP 范围，供该节点上的 Pod 使用。如果指定了此字段，则 0th 条目必须与 podCIDR 字段匹配。它最多可包含每个 IPv4 和 IPv6 的值。
providerID	字符串	由云供应商分配的节点的 ID，格式为： <ProviderName>://<ProviderSpecificNodeID>
污点	array	如果指定，节点的污点。
taints[]	对象	附加到此 Taint 的节点，在任何不容许 Taint 的 pod 上具有"effect"。
不可调度	布尔值	不可调度控制新 pod 的节点调度性。默认情况下，节点可以调度。更多信息： https://kubernetes.io/docs/concepts/nodes/node/#manual-node-administration

14.9.1.2. .spec.configSource

描述

NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
configMap	对象	ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用： https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration

14.9.1.3. .spec.configSource.configMap

描述

ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用：
<https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

类型

object

必填

- namespace
- name
- kubeletConfigKey

属性	类型	描述
kubeletConfigKey	字符串	KubeletConfigKey 声明引用的 ConfigMap 的哪个键对应于 KubeletConfiguration 结构此字段。
name	字符串	name 是引用的 ConfigMap 的 metadata.name。所有情况下都需要此字段。
namespace	字符串	namespace 是引用的 ConfigMap 的 metadata.namespace。所有情况下都需要此字段。
resourceVersion	字符串	resourceVersion 是引用的 ConfigMap 的 metadata.ResourceVersion。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

属性	类型	描述
uid	字符串	UID 是所引用 ConfigMap 的 metadata.UID。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

14.9.1.4. .spec.taints

描述

如果指定，节点的污点。

类型

array

14.9.1.5. .spec.taints[]

描述

附加到此 Taint 的节点，在任何不容许 Taint 的 pod 上具有"effect"。

类型

object

必填

- **key**
- **effect**

属性	类型	描述
----	----	----

属性	类型	描述
effect	字符串	<p>必需。污点对不容许污点的 pod 的效果。有效效果为 NoSchedule、PreferNoSchedule 和 NoExecute。</p> <p>可能枚举值： - "NoExecute" Evict 任何已在运行的 pod，这些 pod 不容许该污点。目前由 NodeController 实施。 - "NoSchedule" 不允许新 pod 调度到该节点上，除非它们容许污点，但允许提交给 Kubelet 的所有 pod 而无需启动调度程序，并允许所有已在运行的 pod 继续运行。调度程序强制使用 - "PreferNoSchedule" Like TaintEffectNoSchedule，但调度程序会尝试不将新 pod 调度到节点上，而不是禁止新 pod 完全调度到该节点上。由调度程序实施。</p>
key	字符串	必需。要应用到节点的污点键。
timeAdded	时间	TimeAdded 代表添加污点的时间。它仅针对 NoExecute 污点编写。
value	字符串	与污点键对应的污点值。

14.9.1.6. .status

描述

NodeStatus 是节点当前状态的信息。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
addresses	array	节点可访问的地址列表。从云供应商查询（如果可用）。更多信息： https://kubernetes.io/docs/concepts/nodes/node/#addresses 注意：此字段被声明为合并，但合并密钥并不够唯一，这可能会在合并时导致数据崩溃。调用者应该使用完整的替换补丁。例如，请参阅 https://pr.k8s.io/79391 。消费者应假定地址可能会在节点生命周期内有所变化。但是，有些例外情况可能无法实现，例如，在 Downward API (status.hostIP) 中继承节点地址的 Pod。
addresses[]	对象	NodeAddress 包含节点地址的信息。
allocatable	对象（数量）	可分配量代表可用于调度的节点资源。默认为 Capacity。
capacity	对象（数量）	capacity 代表节点的总资源。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity
conditions	array	conditions 是当前观察到的节点状况的数组。更多信息： https://kubernetes.io/docs/concepts/nodes/node/#condition
conditions[]	对象	NodeCondition 包含节点的条件信息。
config	对象	NodeConfigStatus 描述了 Node.Spec.ConfigSource 分配的配置的状态。
daemonEndpoints	对象	NodeDaemonEndpoints 列出了由节点上运行的守护进程打开的端口。
images	array	此节点上的容器镜像列表
images[]	对象	描述容器镜像

属性	类型	描述
nodeInfo	对象	NodeSystemInfo 是一组 ids/uuids, 用于唯一标识该节点。
phase	字符串	NodePhase 是节点最近观察到的生命周期阶段。更多信息： https://kubernetes.io/docs/concepts/nodes/node/#phase 字段永远不会填充, 现在已弃用。 可能枚举值：- "Pending" 表示节点已被系统创建/添加, 但没有配置。- "Running" 表示节点已经配置并运行了 Kubernetes 组件。- "Terminated" 表示节点已从集群中移除。
volumesAttached	array	附加到节点的卷列表。
volumesAttached[]	对象	AttachedVolume 描述了附加到节点的卷
volumesInUse	数组 (字符串)	节点使用(mounted)中的可附加卷列表。

14.9.1.7. .status.addresses

描述

节点可访问的地址列表。从云供应商查询 (如果可用)。更多信息：

<https://kubernetes.io/docs/concepts/nodes/node/#addresses> 注意：此字段被声明为合并, 但合并密钥并不够唯一, 这可能会在合并时导致数据崩溃。调用者应该使用完整的替换补丁。例如, 请参阅 <https://pr.k8s.io/79391>。消费者应假定地址可能会在节点生命周期内有所变化。但是, 有些例外情况可能无法实现, 例如, 在 Downward API (status.hostIP) 中继承节点地址的 Pod。

类型

array

14.9.1.8. .status.addresses[]

描述

NodeAddress 包含节点地址的信息。

类型

object

必填

- **type**
- **address**

属性	类型	描述
address	字符串	节点地址。
type	字符串	节点地址类型，主机名、ExternallIP 或 InternallIP 之一。

14.9.1.9. .status.conditions

描述

conditions 是当前观察到的节点状况的数组。更多信息：
<https://kubernetes.io/docs/concepts/nodes/node/#condition>

类型

array

14.9.1.10. .status.conditions[]

描述

NodeCondition 包含节点的条件信息。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastHeartbeatTime	时间	最后一次对给定条件进行更新。
lastTransitionTime	时间	条件从一个状态传输到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	(brief)条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	字符串	节点状况的类型。

14.9.1.11. .status.config

描述

NodeConfigStatus 描述了 Node.Spec.ConfigSource 分配的配置的状态。

类型

object

属性	类型	描述
active	对象	NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用
已分配	对象	NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用
错误	字符串	错误描述了将 Spec.ConfigSource 协调到 Active 配置时的任何问题。可能会出现错误，例如尝试检查点 Spec.ConfigSource 到本地 Assigned 记录，尝试检查点与 Spec.ConfigSource 关联的有效负载，尝试加载或验证 Assigned 配置等。同步配置时在不同点上可能会出现错误。较早的错误（如下载或检查点错误）将导致回滚到 LastKnownGood，并可能会在 Kubelet 重试过程中解决。之后的错误（如加载或验证检查点的配置）将导致回滚到 LastKnownGood。在后者的情况下，通常可以通过修复 Spec.ConfigSource 中分配的配置来解决错误。您可以通过在 Kubelet 日志中搜索错误消息来查找用于调试的附加信息。error 是错误状态的人类可读描述；机器可以检查 Error 是否为空，但不应依赖 Kubelet 版本间 Error 文本的稳定性。
lastKnownGood	对象	NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用

14.9.1.12. .status.config.active

描述

NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用

类型

object

属性	类型	描述
configMap	对象	ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用： https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration

14.9.1.13. .status.config.active.configMap**描述**

ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用：<https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

类型**object****必填**

- **namespace**
- **name**
- **kubeletConfigKey**

属性	类型	描述
kubeletConfigKey	字符串	KubeletConfigKey 声明引用的 ConfigMap 的哪个键对应于 KubeletConfiguration 结构此字段。
name	字符串	name 是引用的 ConfigMap 的 metadata.name。所有情况下都需要此字段。
namespace	字符串	namespace 是引用的 ConfigMap 的 metadata.namespace。所有情况下都需要此字段。
resourceVersion	字符串	resourceVersion 是引用的 ConfigMap 的 metadata.ResourceVersion。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

属性	类型	描述
uid	字符串	UID 是所引用 ConfigMap 的 metadata.UID。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

14.9.1.14. .status.config.assigned

描述

NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用

类型

object

属性	类型	描述
configMap	对象	ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用： https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration

14.9.1.15. .status.config.assigned.configMap

描述

ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用：
<https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

类型

object

必填

- **namespace**
- **name**
- **kubeletConfigKey**

属性	类型	描述
----	----	----

属性	类型	描述
kubeletConfigKey	字符串	KubeletConfigKey 声明引用的 ConfigMap 的哪个键对应于 KubeletConfiguration 结构此字段。
name	字符串	name 是引用的 ConfigMap 的 metadata.name。所有情况下都需要此字段。
namespace	字符串	namespace 是引用的 ConfigMap 的 metadata.namespace。所有情况下都需要此字段。
resourceVersion	字符串	resourceVersion 是引用的 ConfigMap 的 metadata.ResourceVersion。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。
uid	字符串	UID 是所引用 ConfigMap 的 metadata.UID。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

14.9.1.16. .status.config.lastKnownGood

描述

NodeConfigSource 指定节点配置源。正好有一个子字段（不包括元数据）必须是非 nil。此 API 从 1.22 开始已弃用

类型

object

属性	类型	描述
configMap	对象	ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用： https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration

14.9.1.17. .status.config.lastKnownGood.configMap

描述

ConfigMapNodeConfigSource 包含将 ConfigMap 引用为节点的配置源的信息。此 API 从 1.22 开始已弃用：
<https://git.k8s.io/enhancements/keps/sig-node/281-dynamic-kubelet-configuration>

类型

object

必填

- namespace
- name
- kubeletConfigKey

属性	类型	描述
kubeletConfigKey	字符串	KubeletConfigKey 声明引用的 ConfigMap 的哪个键对应于 KubeletConfiguration 结构此字段。
name	字符串	name 是引用的 ConfigMap 的 metadata.name。所有情况下都需要此字段。
namespace	字符串	namespace 是引用的 ConfigMap 的 metadata.namespace。所有情况下都需要此字段。
resourceVersion	字符串	resourceVersion 是引用的 ConfigMap 的 metadata.ResourceVersion。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。
uid	字符串	UID 是所引用 ConfigMap 的 metadata.UID。在 Node.Spec 中禁止此字段，在 Node.Status 中需要。

14.9.1.18. .status.daemonEndpoints

描述

NodeDaemonEndpoints 列出了由节点上运行的守护进程打开的端口。

类型

object

属性	类型	描述
kubeletEndpoint	对象	DaemonEndpoint 包含有关单个守护进程端点的信息。

14.9.1.19. .status.daemonEndpoints.kubeletEndpoint

描述

DaemonEndpoint 包含有关单个守护进程端点的信息。

类型

object

必填

- 端口

属性	类型	描述
端口	整数	给定端点的端口号。

14.9.1.20. .status.images

描述

此节点上的容器镜像列表

类型

array

14.9.1.21. .status.images[]

描述

描述容器镜像

类型

object

属性	类型	描述
名称	数组（字符串）	此镜像的名称已知。例如 ["kubernetes.example/hyperkube:v1.0.7", "cloud-vendor.registry.example/cloud-vendor/hyperkube:v1.0.7"]
sizeBytes	整数	镜像的大小（以字节为单位）。

14.9.1.22. .status.nodeInfo

描述

NodeSystemInfo 是一组 ids/uuids，用于唯一标识该节点。

类型

object

必填

- machineID

- machineID
- systemUUID
- bootID
- kernelVersion
- osImage
- containerRuntimeVersion
- kubeletVersion
- kubeProxyVersion
- operatingSystem
- 架构

属性	类型	描述
架构	字符串	节点报告的架构
bootID	字符串	节点报告的引导 ID。
containerRuntimeVersion	string	节点通过运行时远程 API（如 containerd://1.4.2）报告的容器运行时版本。
kernelVersion	字符串	节点从 'uname -r'（如 3.16.0-0.bpo.4-amd64）报告的内核版本。
kubeProxyVersion	字符串	节点报告的 kubeProxy 版本。
kubeletVersion	字符串	节点报告的 kubelet 版本。
machineID	字符串	节点报告的 MachineID。对于集群中唯一机器标识，首选此字段。从 man (5) machine-id 了解更多： http://man7.org/linux/man-pages/man5/machine-id.5.html
operatingSystem	字符串	节点报告的操作系统
osImage	字符串	节点从 /etc/os-release 报告的操作系统镜像（如 Debian GNU/Linux 7 (wheezy)）。

属性	类型	描述
systemUUID	字符串	节点报告的 SystemUUID。对于唯一机器标识 MachineID。此字段特定于红帽主机 https://access.redhat.com/documentation/zh-cn/red_hat_subscription_management/1/html/rhsm/uuid

14.9.1.23. .status.volumesAttached

描述

附加到节点的卷列表。

类型

array

14.9.1.24. .status.volumesAttached[]

描述

AttachedVolume 描述了附加到节点的卷

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 代表卷应可用的设备路径
name	字符串	附加卷的名称

14.9.2. API 端点

可用的 API 端点如下：

- **/api/v1/nodes**
 - **DELETE** : 删除节点集合
 - **GET**: 列出或监视类型为 Node 的对象
 - **POST** : 创建节点
- **/api/v1/watch/nodes**

- **GET**: 观察单个对节点列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/nodes/{name}**
 - **DELETE** : 删除节点
 - **GET**: 读取指定的节点
 - **PATCH** : 部分更新指定节点
 - **PUT** : 替换指定节点
- **/api/v1/watch/nodes/{name}**
 - **GET**: 观察对类型为 Node 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/nodes/{name}/status**
 - **GET**: 指定节点的读取状态
 - **PATCH** : 部分更新指定节点的状态
 - **PUT** : 替换指定节点的状态

14.9.2.1. /api/v1/nodes

表 14.178. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除节点集合

表 14.179. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.180. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.181. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Node 的对象

表 14.182. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.183. HTTP 响应

HTTP 代码	响应正文
200 - OK	NodeList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建节点

表 14.184. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.185. 主体参数

参数	类型	描述
正文 (body)	Node 模式	

表 14.186. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
201 - Created	Node 模式
202 - Accepted	Node 模式
401 - Unauthorized	空

14.9.2.2. /api/v1/watch/nodes

表 14.187. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对节点列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.188. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.9.2.3. /api/v1/nodes/{name}

表 14.189. 全局路径参数

参数	类型	描述
name	字符串	节点的名称

表 14.190. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除节点

表 14.191. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 14.192. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.193. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的节点

表 14.194. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的节点

表 14.195. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.196. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.197. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
201 - Created	Node 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的节点

表 14.198. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.199. 主体参数

参数	类型	描述
正文 (body)	Node 模式	

表 14.200. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
201 - Created	Node 模式
401 - Unauthorized	空

14.9.2.4. /api/v1/watch/nodes/{name}

表 14.201. 全局路径参数

参数	类型	描述
name	字符串	节点的名称

表 14.202. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 `kind Node.deprecated` 的对象更改。已弃用：使用带有列表操作的 `'watch'` 参数，而是过滤到带有 `'fieldSelector'` 参数的单个项目。

表 14.203. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.9.2.5. /api/v1/nodes/{name}/status

表 14.204. 全局路径参数

参数	类型	描述
name	字符串	节点的名称

表 14.205. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 <code>'true'</code> ，则输出会经过 <code>pretty print</code> 处理。

HTTP 方法

GET

描述

指定节点的读取状态

表 14.206. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定节点的部分更新状态

表 14.207. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.208. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.209. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
201 - Created	Node 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定节点的状态

表 14.210. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.211. 主体参数

参数	类型	描述
正文 (body)	Node 模式	

表 14.212. HTTP 响应

HTTP 代码	响应正文
200 - OK	Node 模式
201 - Created	Node 模式
401 - Unauthorized	空

14.10. PERSISTENTVOLUME [V1]

描述

PersistentVolume (PV)是管理员置备的存储资源。它类似于节点。更多信息：

<https://kubernetes.io/docs/concepts/storage/persistent-volumes>

类型

对象

14.10.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PersistentVolumeSpec 是持久性卷的规格。
status	object	PersistentVolumeStatus 是持久性卷的当前状态。

14.10.1.1. .spec

描述

PersistentVolumeSpec 是持久性卷的规格。

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含可挂载卷的所有方法。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes
awsElasticBlockStore	object	代表 AWS 中的 Persistent Disk 资源。 挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。
azureDisk	对象	azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。
azureFile	对象	azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。
capacity	对象（数量）	capacity 是持久性卷资源和容量的描述。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity
cephfs	object	代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。
cinder	object	代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。
claimRef	object	ObjectReference 包含足够信息供您检查或修改引用的对象。
csi	object	代表由外部 CSI 卷驱动程序(Beta 功能)管理的存储
fc	object	代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

属性	类型	描述
flexVolume	object	FlexPersistentVolumeSource 代表使用基于 exec 的插件置备/附加的通用持久性卷资源。
flocker	object	代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。
gcePersistentDisk	object	代表 Google Compute Engine 中的 Persistent Disk 资源。 在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。
glusterfs	object	代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。
hostPath	object	代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。
iscsi	object	ISCSIPersistentVolumeSource 代表 ISCSI 磁盘。ISCSI 卷只能以读写模式挂载。ISCSI 卷支持所有权管理和 SELinux 重新标记。
local	object	local 代表与节点关联性直接附加的存储(Beta 功能)
mountOptions	数组 (字符串)	mountOptions 是挂载选项列表，如 ["ro", "soft"]。未验证 - 如果挂载无效，则挂载将失败。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes/#mount-options
nfs	object	代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

属性	类型	描述
nodeAffinity	object	VolumeNodeAffinity 定义限制此卷可从中访问哪些节点。
persistentVolumeReclaimPolicy	string	<p>persistentVolumeReclaimPolicy 定义从声明中释放持久性卷时会发生什么。有效选项为 Retain（为手动创建的 PersistentVolume 的默认设置）、Delete（动态置备的 PersistentVolume 的默认）和 Recycle（已弃用）。Recycle 必须被此 PersistentVolume 底层的卷插件支持。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#reclaiming</p> <p>可能枚举值： - "Delete" 表示该卷将从其声明中的 Kubernetes 中删除。卷插件必须支持 Deletion。 - "Recycle" 意味着卷将被重新回收到其声明中未绑定持久性卷的池。卷插件必须支持 Recycling。 - "Retain" 表示该卷将保留在其当前阶段(Released)中，供管理员手动回收。默认策略为 Retain。</p>
photonPersistentDisk	object	代表 Photon Controller 持久磁盘资源。
portworxVolume	object	PortworxVolumeSource 代表一个 Portworx volume 资源。
quobyte	object	代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。
rbd	object	代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。
scaleIO	object	ScaleIOPersistentVolumeSource 代表一个持久的 ScaleIO 卷
storageClassName	string	storageClassName 是这个持久性卷所属的 StorageClass 的名称。空值意味着这个卷不属于任何 StorageClass。
storageos	object	代表 StorageOS 持久性卷资源。

属性	类型	描述
volumeMode	字符串	<p>volumeMode 定义卷是否要与格式化的文件系统一起使用，或者保持在原始块状态。当没有包括在 spec 中时，文件系统的值会被简化。</p> <p>可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统或格式化。</p>
vsphereVolume	object	代表 vSphere 卷资源。

14.10.1.2. .spec.awsElasticBlockStore

描述

代表 AWS 中的 Persistent Disk 资源。

挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumeID**

属性	类型	描述
fsType	string	<p>fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore</p>
分区	整数	<p>partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。</p>

属性	类型	描述
readOnly	布尔值	readOnly 值 true 将强制 VolumeMounts 中的 readOnly 设置。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	VolumeID 是 AWS (Amazon EBS 卷) 中的持久磁盘资源的唯一 ID。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

14.10.1.3. .spec.azureDisk

描述

azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

类型

object

必填

- **diskName**
- **diskURI**

属性	类型	描述
cachingMode	string	cachingMode 是主机缓存模式：None、Read Only、Read Write。 可能枚举值：- "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskname 是 blob 存储中的数据磁盘的名称
diskURI	string	diskURI 是 blob 存储中数据磁盘的 URI
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。

属性	类型	描述
kind	string	kind expected 值是 Shared: multiple blob disk per storage account Managed: azure disk per storage account Managed: azure managed data disk (only in managed availability set))。默认为 shared possible enum 值： - "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。

14.10.1.4. .spec.azureFile

描述

azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

属性	类型	描述
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称
secretNamespace	string	secretNamespace 是包含 Azure Storage Account Name 和 Key default 的 secret 的命名空间与 Pod 相同
shareName	string	sharename 是 azure Share Name

14.10.1.5. .spec.cephfs

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

14.10.1.6. .spec.cephfs.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.7. .spec.cinder**描述**

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumeID**

属性	类型	描述
fsType	string	要挂载的 fstype Filesystem 类型。必须是主机操作系统支持的文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

属性	类型	描述
volumeID	string	用于识别 cinder 中的卷的卷 ID。 更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

14.10.1.8. .spec.cinder.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.9. .spec.claimRef

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。

属性	类型	描述
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.10.1.10. .spec.csi

描述

代表由外部 CSI 卷驱动程序(Beta 功能)管理的存储

类型

object

必填

- **driver**
- **volumeHandle**

属性	类型	描述
----	----	----

属性	类型	描述
controllerExpandSecretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
controllerPublishSecretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
driver	string	driver 是要用于此卷的驱动程序名称。必需。
fsType	string	要挂载的 fstype。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".
nodeExpandSecretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
nodePublishSecretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
nodeStageSecretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
readOnly	布尔值	传递给 ControllerPublishVolumeRequest 的 readOnly 值。默认为 false (读/写)。
volumeAttributes	对象 (字符串)	要发布的卷的 volumeAttributes。
volumeHandle	string	volumeHandle 是 CSI 卷插件的 CreateVolume 返回的唯一卷名称，以在所有后续调用中引用卷。必需。

14.10.1.11. .spec.csi.controllerExpandSecretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.12. .spec.csi.controllerPublishSecretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.13. .spec.csi.nodeExpandSecretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.14. .spec.csi.nodePublishSecretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.15. .spec.csi.nodeStageSecretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.16. .spec.fc

描述

代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。
LUN	整数	LUN 是可选的：FC 目标 lun 号
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组（字符串）	targetWWN 可选：FC 目标全球名称(WWN)

属性	类型	描述
wwids	数组（字符串）	wwids 可选：FC 卷全局广泛的标识符(wwids) Either wwids 或 targetWWNs 和 lun 的组合必须同时设置，但不能同时设置。

14.10.1.17. .spec.flexVolume

描述

FlexPersistentVolumeSource 代表使用基于 exec 的插件置备/附加的通用持久性卷资源。

类型

object

必填

- **driver**

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".默认文件系统取决于 FlexVolume 脚本。
options	对象（字符串）	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

14.10.1.18. .spec.flexVolume.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.19. .spec.flocker

描述

代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。

类型

object

属性	类型	描述
datasetName	string	datasetName 是存储在 Flocker 的 dataset 的 dataset 的 dataset 的 Name。
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

14.10.1.20. .spec.gcePersistentDisk

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **pdName**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	string	pdName 是 GCE 中的 PD 资源的唯一名称。用于识别 GCE 中的磁盘。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

14.10.1.21. .spec.glusterfs

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
endpointsNamespace	string	endpointsNamespace 是包含 Glusterfs 端点的命名空间。如果此字段为空，则 EndpointNamespace 默认为与绑定 PVC 相同的命名空间。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

14.10.1.22. .spec.hostPath

描述

代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- path

属性	类型	描述
path	string	主机上目录的路径。如果路径是符号链接，它将遵循到实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath

属性	类型	描述
type	string	<p>HostPath Volume Defaults 为 "" more info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>可能枚举值：- "" 对于向后兼容，如果未设置 - "BlockDevice" A 块设备必须存在于给定路径 - "CharDevice" A 字符设备必须存在于给定路径上 - "Directory" A 目录必须存在于给定路径 - "DirectoryOrCreate" 中（如果给定路径不存在），将根据需要创建一个空目录，且文件模式为 0755，其具有与 Kubelet 相同的组和所有权。- "File" A 文件必须存在于给定路径上 - "FileOrCreate" 如果给定路径上不存在，将根据需要创建一个空文件，该文件与文件模式 0644 具有相同的组和所有权。- "Socket" A UNIX 套接字必须存在于给定路径上。</p>

14.10.1.23. .spec.iscsi

描述

ISCSIPersistentVolumeSource 代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是 iSCSI 限定名称的目标。
iscsiInterface	string	iscsiInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 是 iSCSI 目标 Lun 编号。
门户	数组 (字符串)	门户是 iSCSI 目标门户列表。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。

14.10.1.24. .spec.iscsi.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.25. .spec.local

描述

local 代表与节点关联性直接附加的存储(Beta 功能)

类型

object

必填

- **path**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。只有在路径是块设备时，它才会应用。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".如果未指定，则默认值为自动选择文件系统。
path	string	到节点上卷的完整路径的路径。它可以是目录或块设备(disk, partition, ...)。

14.10.1.26. .spec.nfs

描述

代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **server**
- **path**

属性	类型	描述
----	----	----

属性	类型	描述
path	string	NFS 服务器导出的路径。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	布尔值	此处的只读将强制使用只读权限挂载 NFS 导出。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server 是 NFS 服务器的主机名或 IP 地址。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs

14.10.1.27. .spec.nodeAffinity

描述

VolumeNodeAffinity 定义限制此卷可从中访问哪些节点。

类型

object

属性	类型	描述
required	object	节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

14.10.1.28. .spec.nodeAffinity.required

描述

节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

类型

object

必填

- **nodeSelectorTerms**

属性	类型	描述
nodeSelectorTerms	array	必需。节点选择器术语列表。条款是 ORed。
nodeSelectorTerms[]	对象	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

14.10.1.29. .spec.nodeAffinity.required.nodeSelectorTerms

描述

必需。节点选择器术语列表。条款是 ORed。

类型

数组

14.10.1.30. .spec.nodeAffinity.required.nodeSelectorTerms[]

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.10.1.31. .spec.nodeAffinity.required.nodeSelectorTerms[].matchExpressions

描述

节点标签(label)的节点选择器要求列表。

类型

数组

14.10.1.32. `.spec.nodeAffinity.required.nodeSelectorTerms[].matchExpressions[]`

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组（字符串）	字符串值数组。如果运算符是 In 或 NotIn，则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist，则值数组必须为空。如果 Operator 是 Gt 或 Lt，则值数组必须具有单个元素，该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.10.1.33. `.spec.nodeAffinity.required.nodeSelectorTerms[].matchFields`

描述

节点字段的节点选择器要求列表。

类型

数组

14.10.1.34. `.spec.nodeAffinity.required.nodeSelectorTerms[].matchFields[]`

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**

- operator

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.10.1.35. .spec.photonPersistentDisk

描述

代表 Photon Controller 持久磁盘资源。

类型

object

必填

- pdID

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。
pdID	string	pdID 是标识 Photon Controller 持久磁盘的 ID

14.10.1.36. .spec.portworxVolume

描述

PortworxVolumeSource 代表一个 Portworx volume 资源。

类型

object

必填

- `volumelD`

属性	类型	描述
<code>fsType</code>	string	<code>fstype</code> 代表要挂载的文件系统类型，必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs". 如果未指定，则隐式推断为"ext4"。
<code>readOnly</code>	布尔值	只读默认为 false（读/写）。此处的 <code>readOnly</code> 将强制使用 <code>VolumeMount</code> 中的 <code>ReadOnly</code> 设置。
<code>volumelD</code>	string	<code>VolumelD</code> 唯一标识一个 Portworx 卷

14.10.1.37. ..spec.quobyte

描述

代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- `registry`
- 卷

属性	类型	描述
<code>group</code>	string	将卷访问映射到 Default 的组没有组
<code>readOnly</code>	布尔值	此处的只读将强制使用只读权限挂载 Quobyte 卷。默认为 false。
<code>registry</code>	string	<code>registry</code> 代表一个或多个 Quobyte Registry 服务，作为字符串指定为 <code>host:port</code> 对（多个条目用逗号分隔），后者充当卷的中央 <code>registry</code>

属性	类型	描述
tenant	string	在带有动态置备 Quobyte 卷的后端中拥有给定 Quobyte 卷的租户，值由插件设置
user	string	用户将卷访问权限映射到默认值到 serviceaccount 用户
卷	string	卷是一个字符串，它根据名称引用已创建的 Quobyte 卷。

14.10.1.38. .spec.rbd

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **monitor**
- **image**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

属性	类型	描述
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。 更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

14.10.1.39. .spec.rbd.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.40. .spec.scaleIO

描述

ScaleIOPersistentVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- gateway
- system
- secretRef

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认为 "xfs"
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret
sslEnabled	布尔值	sslEnabled 是启用/禁用与网关的 SSL 通信的标记，默认为 false
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

14.10.1.41. .spec.scaleIO.secretRef

描述

secretReference 代表一个 Secret Reference。它有足够的信息来检索任何命名空间中的 secret

类型

object

属性	类型	描述
name	string	名称在命名空间内是唯一的，以引用 secret 资源。
namespace	string	namespace 定义 secret 名称必须唯一的空间。

14.10.1.42. .spec.storageos**描述**

代表 StorageOS 持久性卷资源。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	ObjectReference 包含足够信息供您检查或修改引用的对象。
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。

属性	类型	描述
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

14.10.1.43. .spec.storageos.secretRef

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 desiredState.manifest.containers[2]。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似："spec.containers{name}"（其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 "spec.containers[2]"（使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.10.1.44. .spec.vsphereVolume

描述

代表 vSphere 卷资源。

类型

object

必填

- **volumePath**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为"ext4"。
storagePolicyID	string	StoragePolicyID 是与 StoragePolicyName 关联的存储 Policy Based Management (SPBM)配置集 ID。

属性	类型	描述
storagePolicyName	string	StoragePolicyName 是存储基于策略的管理(SPBM)配置集名称。
volumePath	string	volumePath 是标识 vSphere 卷 vmdk 的路径

14.10.1.45. .status

描述

PersistentVolumeStatus 是持久性卷的当前状态。

类型

object

属性	类型	描述
message	string	Message 是人类可读的消息，指示卷处于此状态的详细信息。
phase	string	<p>phase 表示卷是否可用、绑定到声明或被声明释放。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#phase</p> <p>可能枚举的值： - "Available" 用于尚未绑定的 PersistentVolume，并被 binder 保存，并与 PersistentVolumeClaims - "Bound" 用于绑定的 - "Failed" 用于从其中释放后正确回收或删除的 PersistentVolume。claim - "Pending" 用于不可用的 PersistentVolume - "Released" 用于 PersistentVolume，其中绑定 PersistentVolumeClaim 被删除了发布的卷，然后必须重新回收该阶段，供持久性卷声明绑定程序用于向另一个进程发出信号，以回收资源</p>
reason	string	reason 是一个简单的 CamelCase 字符串，用于描述任何失败，用于 CLI 中的机器解析和截断显示。

14.10.2. API 端点

可用的 API 端点如下：

- **/api/v1/persistentvolumes**
 - **DELETE** : 删除 PersistentVolume 的集合
 - **GET**: 列出或监视类型为 PersistentVolume 的对象
 - **POST** : 创建 PersistentVolume
- **/api/v1/watch/persistentvolumes**
 - **GET**: 观察单个对 PersistentVolume 列表的更改。已弃用 : 改为使用 'watch' 参数和 list 操作。
- **/api/v1/persistentvolumes/{name}**
 - **DELETE** : 删除 PersistentVolume
 - **GET** : 读取指定的 PersistentVolume
 - **PATCH** : 部分更新指定的 PersistentVolume
 - **PUT** : 替换指定的 PersistentVolume
- **/api/v1/watch/persistentvolumes/{name}**
 - **GET**: 观察对类型为 PersistentVolume 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/persistentvolumes/{name}/status**
 - **GET**: 指定 PersistentVolume 的读取状态
 - **PATCH** : 部分更新指定 PersistentVolume 的状态
 - **PUT** : 替换指定 PersistentVolume 的状态

14.10.2.1. /api/v1/persistentvolumes

表 14.213. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PersistentVolume 的集合

表 14.214. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.215. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.216. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PersistentVolume 的对象

表 14.217. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.218. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建 PersistentVolume

表 14.219. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.220. 主体参数

参数	类型	描述
正文 (body)	PersistentVolume 模式	

表 14.221. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
201 - Created	PersistentVolume 模式
202 - Accepted	PersistentVolume 模式
401 - 未授权	空

14.10.2.2. /api/v1/watch/persistentvolumes

表 14.222. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PersistentVolume 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.223. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.10.2.3. /api/v1/persistentvolumes/{name}

表 14.224. 全局路径参数

参数	类型	描述
name	string	PersistentVolume 的名称

表 14.225. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PersistentVolume

表 14.226. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 14.227. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.228. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
202 - Accepted	PersistentVolume 模式
401 - 未授权	空

HTTP 方法

GET

描述

读取指定的 PersistentVolume

表 14.229. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 PersistentVolume

表 14.230. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.231. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.232. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
201 - Created	PersistentVolume 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 PersistentVolume

表 14.233. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.234. 主体参数

参数	类型	描述
正文 (body)	PersistentVolume 模式	

表 14.235. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
201 - Created	PersistentVolume 模式
401 - 未授权	空

14.10.2.4. /api/v1/watch/persistentvolumes/{name}

表 14.236. 全局路径参数

参数	类型	描述
name	string	PersistentVolume 的名称

表 14.237. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 PersistentVolume 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.238. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.10.2.5. /api/v1/persistentvolumes/{name}/status

表 14.239. 全局路径参数

参数	类型	描述
name	string	PersistentVolume 的名称

表 14.240. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 PersistentVolume 的读取状态

表 14.241. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 PersistentVolume 的部分更新状态

表 14.242. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.243. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.244. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
201 - Created	PersistentVolume 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 PersistentVolume 的状态

表 14.245. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.246. 主体参数

参数	类型	描述
正文 (body)	PersistentVolume 模式	

表 14.247. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolume 模式
201 - Created	PersistentVolume 模式
401 - 未授权	空

14.11. PERSISTENTVOLUMECLAIM [V1]

描述

PersistentVolumeClaim 是用户对持久性卷的请求并声明

类型

对象

14.11.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源
status	对象	PersistentVolumeClaimStatus 是持久性卷声明的当前状态。

14.11.1.1. .spec

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含卷应具有的所有访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	对象	TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。
dataSourceRef	object	dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。
resources	对象	ResourceRequirements 描述计算资源要求。

属性	类型	描述
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

14.11.1.2. .spec.dataSource

描述

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

14.11.1.3. .spec.dataSourceRef

描述

`dataSourceRef` 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 `PersistentVolumeClaim` 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 `dataSource` 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 `namespace` 没有在 `dataSourceRef` 中指定时，如果其中一个字段(`dataSource` 和 `dataSourceRef`)都自动设置为相同的值，另一个是非空的值。当在 `dataSourceRef` 中指定命名空间时，`dataSource` 不会设置为相同的值，且必须为空。`dataSource` 和 `dataSourceRef`: * While `dataSource` 有三个重要区别：但 `dataSourceRef` 只允许两种特定类型的对象，`dataSourceRef` 允许任何非核心对象和 `PersistentVolumeClaim` 对象。* 虽然 `dataSource` 忽略禁止的值（过滤它们），`dataSourceRef` 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 `dataSource` 只允许本地对象，但 `dataSourceRef` 允许任何命名空间中的对象。(beta)使用此字段需要启用 `AnyVolumeDataSource` 功能门。(alpha)使用 `dataSourceRef` 的 `namespace` 字段需要启用 `CrossNamespaceVolumeDataSource` 功能门。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	<code>apiGroup</code> 是被引用资源的组。如果没有指定 <code>APIGroup</code> ，则指定的 <code>Kind</code> 必须位于 <code>core</code> API 组中。对于任何其他第三方类型，需要 <code>APIGroup</code> 。
kind	字符串	<code>kind</code> 是被引用的资源类型
name	字符串	<code>name</code> 是被引用的资源的名称
namespace	string	<code>namespace</code> 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 <code>gateway.networking.k8s.io/ReferenceGrant</code> 对象，以允许该命名空间的所有者接受引用。详情请查看 <code>ReferenceGrant</code> 文档。(alpha)此字段需要启用 <code>CrossNamespaceVolumeDataSource</code> 功能门。

14.11.1.4. .spec.resources

描述

`ResourceRequirements` 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.11.1.5. .spec.resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.11.1.6. .spec.resources.claims[]

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- name

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.11.1.7. .status

描述

PersistentVolumeClaimStatus 是持久性卷声明的当前状态。

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含支持 PVC 的卷的实际访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
allocatedResources	对象（数量）	allocatedResources 是 AllocatedResources 中的存储资源，跟踪分配给 PVC 的容量。当请求卷扩展操作时，它可能大于实际容量。对于存储配额，使用 allocatedResources 和 PVC.spec.resources 的较大的值。如果没有设置 allocatedResources，则只用于配额计算的 PVC.spec.resources。如果卷扩展容量请求较低，则仅当没有扩展操作且实际卷容量等于或小于请求的容量时，才会降低 allocatedResources。这是一个 alpha 字段，需要启用 RecoverVolumeExpansionFailure 功能。
capacity	对象（数量）	capacity 代表底层卷的实际资源。

属性	类型	描述
conditions	数组	conditions 是持久性卷声明的当前条件。如果底层持久性卷大小调整，则 Condition 将设置为 'ResizeStarted'。
conditions[]	object	PersistentVolumeClaimCondition 包含有关 pvc 状态的详细信息
phase	string	<p>phase 代表 PersistentVolumeClaim 的当前阶段。</p> <p>可能枚举值： - "Bound" 用于绑定的 PersistentVolumeClaims - "Lost" 用于丢失其底层 PersistentVolume 的 PersistentVolumeClaim。这个声明被绑定到一个 PersistentVolume，这个卷不再存在，且其上的所有数据都丢失。 - "Pending" 用于尚未绑定的 PersistentVolumeClaims</p>

属性	类型	描述
resizeStatus	string	<p>resizeStatus 存储调整大小操作的状态。默认情况下，不会设置 ResizeStatus，但当扩展为 complete resizeStatus 时，通过调整大小 controller 或 kubelet 被设为空字符串。这是一个 alpha 字段，需要启用 RecoverVolumeExpansionFailure 功能。</p> <p>可能枚举值： - "" when expansion is complete, 通过调整大小 controller 或 kubelet 设置空字符串。 - 当扩展失败时，会在扩展失败时使用终端错误设置</p> <p>"ControllerExpansionFailed" State set. 临时错误（如 timeout）不应设置这个状态，且应该修改 ResizeStatus，因为调整大小的控制器可以恢复卷扩展。 - 在 control-plane 中扩展卷时，</p> <p>"ControllerExpansionInProgress" State set failed with an terminal error. 临时错误不会设置 NodeExpansionFailed. - 当 kubelet 启动扩展卷时，</p> <p>"NodeExpansionInProgress" State set when</p> <p>"NodeExpansionPending" State set when resize controller has completed the volume but on the node is required on the node.</p>

14.11.1.8. .status.conditions

描述

conditions 是持久性卷声明的当前条件。如果底层持久性卷大小调整，则 Condition 将设置为 'ResizeStarted'。

类型

array

14.11.1.9. .status.conditions[]

描述

PersistentVolumeClaimCondition 包含有关 pvc 状态的详细信息

类型

object

必填

- **type**

- **status**

属性	类型	描述
lastProbeTime	时间	lastProbeTime 是我们探测到条件的 时间。
lastTransitionTime	时间	lastTransitionTime 是条件从一个 状态转换到另一个状态的时间。
message	string	Message 是人类可读的消息，指示 最后一次转换的详细信息。
reason	string	原因是唯一的，这应该是简短的、 机器易理解的字符串，它给出了状 况最后一次转换的原因。如果报告 了"ResizeStarted"，这表示基础持 久卷正在调整大小。
status	字符串	
type	字符串	

14.11.2. API 端点

可用的 API 端点如下：

- **/api/v1/persistentvolumeclaims**
 - **GET**: 列出或监视类型为 PersistentVolumeClaim 的对象
- **/api/v1/watch/persistentvolumeclaims**
 - **GET**: 观察单个对 PersistentVolumeClaim 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/persistentvolumeclaims**
 - **DELETE** : 删除 PersistentVolumeClaim 的集合
 - **GET**: 列出或监视类型为 PersistentVolumeClaim 的对象
 - **POST** : 创建 PersistentVolumeClaim
- **/api/v1/watch/namespaces/{namespace}/persistentvolumeclaims**
 - **GET**: 观察单个对 PersistentVolumeClaim 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}**
 - **DELETE** : 删除 PersistentVolumeClaim

- **GET**: 阅读指定的 PersistentVolumeClaim
- **PATCH** : 部分更新指定的 PersistentVolumeClaim
- **PUT** : 替换指定的 PersistentVolumeClaim
- **/api/v1/watch/namespaces/{namespace}/persistentvolumeclaims/{name}**
 - **GET**: 观察对类型为 PersistentVolumeClaim. deprecated 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status**
 - **GET**: 指定 PersistentVolumeClaim 的读取状态
 - **PATCH** : 部分更新指定 PersistentVolumeClaim 的状态
 - **PUT** : 替换指定 PersistentVolumeClaim 的状态

14.11.2.1. /api/v1/persistentvolumeclaims

表 14.248. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 PersistentVolumeClaim 的对象

表 14.249. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaimList 模式
401 - Unauthorized	空

14.11.2.2. /api/v1/watch/persistentvolumeclaims

表 14.250. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法**GET****描述**

观察单个对 PersistentVolumeClaim 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.251. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.11.2.3. /api/v1/namespaces/{namespace}/persistentvolumeclaims

表 14.252. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.253. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PersistentVolumeClaim 的集合

表 14.254. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.255. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.256. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PersistentVolumeClaim 的对象

表 14.257. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.258. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaimList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 PersistentVolumeClaim

表 14.259. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.260. 主体参数

参数	类型	描述
正文 (body)	PersistentVolumeClaim 模式	

表 14.261. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
201 - Created	PersistentVolumeClaim 模式
202 - Accepted	PersistentVolumeClaim 模式
401 - Unauthorized	空

14.11.2.4. /api/v1/watch/namespaces/{namespace}/persistentvolumeclaims

表 14.262. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.263. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	<code>allowWatchBookmarks</code> 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 <code>continue</code> 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 <code>continue</code> 值，并带有相同的查询参数（除 <code>continue</code> 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 <code>continue</code> 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 <code>ResourceExpired</code> 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 <code>true</code> 时，不支持此字段。客户端可以从服务器返回的最后一个 <code>resourceVersion</code> 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PersistentVolumeClaim 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.264. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.11.2.5. /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

表 14.265. 全局路径参数

参数	类型	描述
name	string	PersistentVolumeClaim 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.266. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PersistentVolumeClaim

表 14.267. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.268. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.269. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>PersistentVolumeClaim</code> 模式
202 - Accepted	<code>PersistentVolumeClaim</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 `PersistentVolumeClaim`

表 14.270. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>PersistentVolumeClaim</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `PersistentVolumeClaim`

表 14.271. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.272. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.273. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
201 - Created	PersistentVolumeClaim 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 PersistentVolumeClaim

表 14.274. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.275. 主体参数

参数	类型	描述
正文 (body)	PersistentVolumeClaim 模式	

表 14.276. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
201 - Created	PersistentVolumeClaim 模式
401 - Unauthorized	空

14.11.2.6. /api/v1/watch/namespaces/{namespace}/persistentvolumeclaims/{name}

表 14.277. 全局路径参数

参数	类型	描述
name	string	PersistentVolumeClaim 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.278. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind PersistentVolumeClaim. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.279. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.11.2.7. /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status

表 14.280. 全局路径参数

参数	类型	描述
name	string	PersistentVolumeClaim 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.281. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 PersistentVolumeClaim 的读取状态

表 14.282. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 PersistentVolumeClaim 的部分更新状态

表 14.283. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.284. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.285. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
201 - Created	PersistentVolumeClaim 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 PersistentVolumeClaim 的状态

表 14.286. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.287. 主体参数

参数	类型	描述
正文 (body)	PersistentVolumeClaim 模式	

表 14.288. HTTP 响应

HTTP 代码	响应正文
200 - OK	PersistentVolumeClaim 模式
201 - Created	PersistentVolumeClaim 模式
401 - Unauthorized	空

14.12. POD [V1]

描述

Pod 是可以在主机上运行的容器的集合。此资源由客户端创建，并调度到主机上。

类型

对象

14.12.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec 是 pod 的描述。
status	object	PodStatus 代表有关 pod 状态的信息。状态可能会跟踪系统的实际状态，特别是在托管 pod 的节点无法联系 control plane 时。

14.12.1.1. .spec

描述

PodSpec 是 pod 的描述。

类型

object

必填

- **containers**

属性	类型	描述
----	----	----

属性	类型	描述
activeDeadlineSeconds	整数	在系统主动尝试标记失败并终止相关的容器前， pod 可以在相对于 StartTime 的节点上处于活跃状态。值必须是正整数。
关联性	object	affinity 是一组关联性调度规则。
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 指明是否应该自动挂载服务帐户令牌。
containers	array	属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。
containers[]	对象	要在 pod 中运行的单一应用程序容器。
dnsConfig	object	PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。
dnsPolicy	字符串	<p>为 pod 设置 DNS 策略。默认为 "ClusterFirst"。有效值为 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' 或 'None'。DNSConfig 中给定的 DNS 参数将与选择与 DNSPolicy 的策略合并。要将 DNS 选项与 hostNetwork 一起设置，您必须将 DNS 策略明确指定为 'ClusterFirstWithHostNet'。</p> <p>可能枚举值： - "ClusterFirst" 表示 pod 应该先使用集群 DNS，除非 hostNetwork 为 true，如果可用，则回退到默认的（由 kubelet 决定）DNS 设置。 - "ClusterFirstWithHostNet" 表示 pod 应该首先使用集群 DNS，如果可用，则回退到它。在默认的（由 kubelet 决定）DNS 设置中。 - "Default" 表示 pod 应该使用默认（由 kubelet 决定）DNS 设置。 - "None" 表示 pod 应该使用空的 DNS 设置。名称服务器和搜索路径等 DNS 参数应该通过 DNSConfig 定义。</p>

属性	类型	描述
enableServiceLinks	布尔值	EnableServiceLinks 指示是否应将有关服务的信息注入到 pod 的环境变量中，与 Docker 链接的语法匹配。可选：默认为 true。
ephemeralContainers	array	此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。
ephemeralContainers[]	object	EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。 要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。
hostAliases	array	hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。
hostAliases[]	对象	HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。
hostIPC	布尔值	使用主机的 ipc 命名空间。可选：默认为 false。
hostNetwork	布尔值	为此 pod 请求的主机网络。使用主机的网络命名空间。如果设置了这个选项，则必须指定要使用的端口。默认为 false。
hostPID	布尔值	使用主机的 pid 命名空间。可选：默认为 false。

属性	类型	描述
hostUsers	布尔值	使用主机的用户命名空间。可选：默认为 true。如果设置为 true 或不存在，则 pod 将在主机用户命名空间中运行，当 pod 需要可供主机用户命名空间使用的功能时，如加载带有 CAP_SYS_MODULE 的内核模块。当设置为 false 时，会为 pod 创建一个新的 usersns。设置 false 有助于缓解容器中漏洞，即使允许用户以 root 身份运行其容器，而无需实际在主机上具有 root 特权。此字段是 alpha-level，仅被启用 UserNamespacesSupport 功能的服务器所接受。
hostname	字符串	如果没有指定 Pod 的主机名，则 pod 的主机名将设置为系统定义的值。
imagePullSecrets	array	imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息： https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	对象	LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

属性	类型	描述
initContainers	array	属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
initContainers[]	对象	要在 pod 中运行的单一应用程序容器。
nodeName	字符串	nodeName 是一个将此 pod 调度到特定节点的请求。如果是非空，调度程序只会将此 pod 调度到该节点上，假设它符合资源要求。
nodeSelector	对象（字符串）	nodeSelector 是一个选择器，必须为 true 才能使 pod 适合节点。必须与 pod 标签匹配的选择器，才能调度到该节点上。更多信息： https://kubernetes.io/docs/concepts/configuration/assign-pod-node/
OS	object	PodOS 定义 pod 的 OS 参数。

属性	类型	描述
开销	对象 (数量)	<p>开销代表了与为给定 RuntimeClass 运行 pod 相关的资源开销。此字段将由 RuntimeClass 准入控制器在准入时自动填充。如果启用了 RuntimeClass 准入控制器，则无法在 Pod 创建请求中设置开销。RuntimeClass 准入控制器将拒绝 Pod 创建已设置开销的请求。如果在 PodSpec 中配置并选择了 RuntimeClass，则 Overhead 将被设置为对应的 RuntimeClass 中定义的值，否则它将保留为未设置并被视为零。更多信息： https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md</p>
preemptionPolicy	字符串	<p>preemptionPolicy 是抢占优先级较低的 pod 的策略。Never, PreemptLowerPriority 之一。如果未设置，则默认为 PreemptLowerPriority。</p> <p>可能枚举值： - "Never" 表示 pod 不会抢占优先级较低的其他 pod。 - "PreemptLowerPriority" 表示 pod 可以抢占优先级较低的其他 pod。</p>
priority	整数	<p>优先级值。各种系统组件使用此字段查找 pod 的优先级。启用 Priority Admission Controller 时，它会阻止用户设置此字段。准入控制器从 PriorityClassName 填充此字段。数值越大，优先级越高。</p>
priorityClassName	字符串	<p>如果指定，则表示 pod 的优先级。"system-node-critical"和"system-cluster-critical"是两个特殊关键字，表示前者最高的优先级。必须使用该名称创建 PriorityClass 对象来定义任何其他名称。如果没有指定，如果没有默认值，pod 优先级将为 default 或零。</p>

属性	类型	描述
readinessGates	array	如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪： https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	对象	PodReadinessGate 包含对 pod 条件的引用
resourceClaims	数组	ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。
resourceClaims[]	object	PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。
restartPolicy	字符串	为 pod 中的所有容器重启策略。Always、OnFailure、Never 之一。在某些上下文中，仅允许这些值的子集。默认值为 Always。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy 可能枚举值：- "Always" - "Never" - "OnFailure"

属性	类型	描述
runtimeClassName	字符串	runtimeClassName 指的是 node.k8s.io 组中的 RuntimeClass 对象，它应该用于运行此 pod。如果没有 RuntimeClass 资源与命名类匹配，则不会运行 pod。如果未设置或为空，则使用 "legacy" RuntimeClass，它是一个带有使用默认运行时处理器的空定义的隐式类。更多信息： https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class
schedulerName	字符串	如果指定，pod 将由指定的调度程序分配。如果没有指定，pod 将默认调度程序分配。
schedulingGates	数组	SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。 SchedulingGates 只能在创建 pod 时设置，之后只能删除。 这是 PodSchedulingReadiness 功能门启用的 beta 功能。
schedulingGates[]	object	PodSchedulingGate 与 Pod 关联，以保护其调度。
securityContext	object	PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。 container.securityContext 的字段值优先于 PodSecurityContext 的字段值。
serviceAccount	字符串	deprecatedServiceAccount 是 ServiceAccountName 的前文别名。deprecated：改为使用 serviceAccountName。

属性	类型	描述
serviceAccountName	字符串	serviceAccountName 是用于运行此 pod 的 ServiceAccount 的名称。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
setHostnameAsFQDN	布尔值	如果为 true，则 pod 的主机名将配置为 pod 的 FQDN，而不是叶名称（默认值）。在 Linux 容器中，这意味着在内核的 hostname 字段中设置 FQDN (struct utsname 的 nodename 字段)。在 Windows 容器中，这意味着将 registry 键 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 的 registry 值设置为 FQDN。如果 pod 没有 FQDN，则无效。默认为 false。
shareProcessNamespace	布尔值	在 pod 中所有容器间共享单个进程命名空间。设置此容器后，将能够查看和信号来自同一 pod 中其他容器的进程，并且每个容器中的第一个进程不会被分配 PID 1。无法同时设置 hostPID 和 ShareProcessNamespace。可选：默认为 false。
subdomain	字符串	如果指定，完全限定的 Pod 主机名将是 "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>"。如果没有指定，pod 根本不会具有 domainname。
terminationGracePeriodSeconds	整数	pod 需要安全终止的可选持续时间（以秒为单位）。在删除请求中可能会减少。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。如果这个值为 nil，则使用默认的宽限期。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。默认值为 30 秒。

属性	类型	描述
容限 (tolerations)	array	如果指定, pod 的容限。
tolerations[]	对象	附加到此 Tolerant pod, 以容许任何与 triple <key,value,effect> 匹配的污点, 使用匹配的 operator <operator>。
topologySpreadConstraints	array	topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。
topologySpreadConstraints[]	对象	TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。
卷	array	可由属于 pod 的容器挂载的卷列表。更多信息： https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	对象	卷代表一个 pod 中的命名卷, 可以被 pod 中的任何容器访问。

14.12.1.2. .spec.affinity

描述

affinity 是一组关联性调度规则。

类型

object

属性	类型	描述
nodeAffinity	object	节点关联性是一组节点关联性调度规则。
podAffinity	object	Pod 关联性是一组 pod 关联性调度规则。
podAntiAffinity	object	Pod 反关联性是一组 pod 反关联性调度规则。

14.12.1.3. .spec.affinity.nodeAffinity

描述

节点关联性是一组节点关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。
requiredDuringSchedulingIgnoredDuringExecution	object	节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

14.12.1.4. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。

类型

array

14.12.1.5. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

描述

空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。

类型

object

必填

- **weight**

- preference

属性	类型	描述
preference	object	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。
weight	整数	与对应的 nodeSelectorTerm 匹配的权重，范围为 1-100。

14.12.1.6. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pre

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.12.1.7. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pre

描述

节点标签(label)的节点选择器要求列表。

类型

array

14.12.1.8. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pre

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.12.1.9. `.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pre`

描述

节点字段的节点选择器要求列表。

类型

array

14.12.1.10. `.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pr`

描述

节点选择器要求是一个选择器, 其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.12.1.11. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

描述

节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

类型

object

必填

- **nodeSelectorTerms**

属性	类型	描述
nodeSelectorTerms	array	必需。节点选择器术语列表。条款是 ORed。
nodeSelectorTerms[]	对象	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

14.12.1.12. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node!

描述

必需。节点选择器术语列表。条款是 ORed。

类型

array**14.12.1.13. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node:****描述**

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型**object**

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.12.1.14. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node:**描述**

节点标签(label)的节点选择器要求列表。

类型**array****14.12.1.15. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node:****描述**

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型**object****必填**

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。

属性	类型	描述
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.12.1.16. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node

描述

节点字段的节点选择器要求列表。

类型

array

14.12.1.17. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.node!

描述

节点选择器要求是一个选择器, 其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"

属性	类型	描述
values	数组（字符串）	字符串值数组。如果运算符是 In 或 NotIn，则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist，则值数组必须为空。如果 Operator 是 Gt 或 Lt，则值数组必须具有单个元素，该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.12.1.18. .spec.affinity.podAffinity

描述

Pod 关联性是一组 pod 关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中，如果节点具有与相应 podAffinityTerm 匹配的 pod；具有最高总和最高的节点是首选的节点。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。
requiredDuringSchedulingIgnoredDuringExecution	array	如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

属性	类型	描述
requiredDuringSchedulingIgnoredDuringExecution[]	对象	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.12.1.19. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中，如果节点具有与相应 podAffinityTerm 匹配的 pod；具有最高总和最高的节点是首选的节点。

类型

array

14.12.1.20. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

描述

所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- **weight**
- **podAffinityTerm**

属性	类型	描述
podAffinityTerm	object	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

14.12.1.21. `.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`.pod

描述

定义一组 pod（与给定命名空间匹配的 `labelSelector` 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 `co-located` 定义在带有键 `<topologyKey>` 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- `topologyKey`

属性	类型	描述
<code>labelSelector</code>	<code>labelSelector</code>	一个对一组资源的标签查询，本例中为 pod。
<code>namespaceSelector</code>	<code>labelSelector</code>	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 <code>namespaces</code> 字段中列出的命名空间。 <code>null selector</code> 和 <code>null</code> 或 <code>empty namespace</code> 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	<code>namespaces</code> 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 <code>namespaceSelector.null</code> 或空命名空间列表和 <code>null namespaceSelector</code> 选择的命名空间合并意味着"此 pod 的命名空间"。
<code>topologyKey</code>	字符串	此 pod 应该与指定命名空间中 <code>labelSelector</code> 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 <code>co-located</code> 定义为在具有键 <code>topologyKey</code> 标签的节点中运行。不允许空的 <code>topologyKey</code> 。

14.12.1.22. `.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

描述

如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 `podAffinityTerm` 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

array**14.12.1.23. .spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]****描述**

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型**object****必填**

- **topologyKey**

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.12.1.24. .spec.affinity.podAntiAffinity**描述**

Pod 反关联性是一组 pod 反关联性调度规则。

类型**object**

属性	类型	描述
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、 <code>requiredDuringSchedulingIgnoredDuringExecution</code> 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 <code>podAffinityTerm</code> 匹配的 pod；如果节点具有与相应 <code>podAffinityTerm</code> 匹配的 pod，则该节点是首选的。
<code>preferredDuringSchedulingIgnoredDuringExecution[]</code>	对象	所有匹配的 <code>WeightedPodAffinityTerm</code> 字段的权重都会被添加，以查找首选节点。
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 <code>podAffinityTerm</code> 对应的节点列表会被干扰，例如，所有术语都必须满足。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（与给定命名空间匹配的 <code>labelSelector</code> 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 <code>co-located</code> 定义在带有键 <code><topologyKey></code> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.12.1.25. `.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、`requiredDuringSchedulingIgnoredDuringExecution` 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 `podAffinityTerm` 匹配的 pod；如果节点具有与相应 `podAffinityTerm` 匹配的 pod，则该节点是首选的。

类型

array

14.12.1.26. `.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

所有匹配的 `WeightedPodAffinityTerm` 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- `weight`
- `podAffinityTerm`

属性	类型	描述
<code>podAffinityTerm</code>	object	定义一组 pod（与给定命名空间匹配的 <code>labelSelector</code> 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 <code>co-located</code> 定义在带有键 <code><topologyKey></code> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
<code>weight</code>	整数	与对应的 <code>podAffinityTerm</code> 匹配的权重，范围为 1-100。

14.12.1.27. `.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

定义一组 pod（与给定命名空间匹配的 `labelSelector` 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 `co-located` 定义在带有键 `<topologyKey>` 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- `topologyKey`

属性	类型	描述
<code>labelSelector</code>	labelSelector	一个对一组资源的标签查询，本例中为 pod。

属性	类型	描述
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.12.1.28. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

描述

如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

array

14.12.1.29. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- **topologyKey**

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.12.1.30. .spec.containers

描述

属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。

类型

array

14.12.1.31. .spec.containers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- **name**

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组（字符串）	ENTRYPOINT 数组。不会在 shell 中执行。如果没有提供，则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 <code>C_IDENTIFIER</code> 。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

属性	类型	描述
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。

属性	类型	描述
ports	array	要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。

属性	类型	描述
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。

属性	类型	描述
volumeMounts	array	要挂载到容器文件系统上的 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.12.1.32. .spec.containers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

array

14.12.1.33. .spec.containers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- **name**

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的一个源。

14.12.1.34. `.spec.containers[].env[].valueFrom`

描述

EnvVarSource 代表 EnvVar 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.12.1.35. `.spec.containers[].env[].valueFrom.configMapKeyRef`

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.12.1.36. `.spec.containers[].env[].valueFrom.fieldRef`

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.12.1.37. .spec.containers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.12.1.38. .spec.containers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.12.1.39. .spec.containers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

array

14.12.1.40. .spec.containers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.12.1.41. .spec.containers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.12.1.42. .spec.containers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.12.1.43. .spec.containers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.12.1.44. .spec.containers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.12.1.45. .spec.containers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root（'/'）。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（' ' 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.46. .spec.containers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.47. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.48. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.49. .spec.containers[].lifecycle.postStart.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.50. .spec.containers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.12.1.51. .spec.containers[].lifecycle.preStop.exec**描述**

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/')。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.52. .spec.containers[].lifecycle.preStop.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.53. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.54. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.55. .spec.containers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.56. .spec.containers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。

属性	类型	描述
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.57. .spec.containers[].livenessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.58. .spec.containers[].livenessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.59. .spec.containers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.60. .spec.containers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.61. .spec.containers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.62. .spec.containers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.63. .spec.containers[].ports

描述

要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

array

14.12.1.64. .spec.containers[].ports[]

描述

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.12.1.65. .spec.containers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.66. .spec.containers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令 (' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.12.1.67. .spec.containers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.68. .spec.containers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.69. .spec.containers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.70. .spec.containers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.71. .spec.containers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.72. .spec.containers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.12.1.73. .spec.containers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.12.1.74. .spec.containers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.75. .spec.containers[].resources.claims

描述

声明列出了此容器使用的 `spec.resourceClaims` 中定义的资源名称。
这是一个 alpha 字段，需要启用 `DynamicResourceAllocation` 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.76. `.spec.containers[].resources.claims[]`

描述

`ResourceClaim` 引用 `PodSpec.ResourceClaims` 中的一个条目。

类型

object

必填

- `name`

属性	类型	描述
<code>name</code>	<code>string</code>	<code>name</code> 必须与使用此字段的 Pod 的 <code>pod.spec.resourceClaims</code> 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.77. `.spec.containers[].securityContext`

描述

`securityContext` 包含将应用到容器的安全配置。`SecurityContext` 和 `PodSecurityContext` 中都存在一些字段。当同时设置这两个时，`SecurityContext` 中的值就会优先使用。

类型

object

属性	类型	描述
<code>allowPrivilegeEscalation</code>	布尔值	<code>allowPrivilegeEscalation</code> 控制进程是否可以获得比父进程更多的特权。此 <code>bool</code> 直接控制容器进程中是否设置了 <code>no_new_privs</code> 标志。当容器以 <code>Privileged 2</code> 运行行为 <code>Privileged 2</code> 时， <code>allowPrivilegeEscalation</code> 为 <code>true</code> 时，当 <code>spec.os.name</code> 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。

属性	类型	描述
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.12.1.78. .spec.containers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.12.1.79. .spec.containers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.12.1.80. .spec.containers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。

属性	类型	描述
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.12.1.81. .spec.containers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	<p>GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。</p>
gmsaCredentialSpecName	字符串	<p>GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。</p>

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level, 仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外, 如果 HostProcess 为 true, 则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName, 以运行容器进程的入口点。如果未指定, 则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置, 则优先使用 SecurityContext 指定的值。

14.12.1.82. .spec.containers[].startupProbe

描述

probe 描述了对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后, 探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.83. `.spec.containers[].startupProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录（ <code>/</code> ）。命令只是 <code>exec'd</code> ，它不在 shell 中运行，因此传统的 shell 指令（ <code> </code> 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 <code>live/healthy</code> ，非零为不健康。

14.12.1.84. `.spec.containers[].startupProbe.grpc`**描述**

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.85. `.spec.containers[].startupProbe.httpGet`**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.86. .spec.containers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.87. .spec.containers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.88. .spec.containers[].startupProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.89. .spec.containers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

array

14.12.1.90. .spec.containers[].volumeDevices[]

描述

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.12.1.91. .spec.containers[].volumeMounts

描述

要挂载到容器文件系统 Pod 卷。无法更新。

类型

array

14.12.1.92. .spec.containers[].volumeMounts[]

描述

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- **name**
- **mountPath**

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 '..'。

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.12.1.93. .spec.dnsConfig

描述

PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

类型

object

属性	类型	描述
nameservers	数组（字符串）	DNS 名称服务器 IP 地址列表。这将附加到从 DNSPolicy 生成的基本名称服务器。将删除重复的名称服务器。
options	array	DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。
options[]	对象	PodDNSConfigOption 定义 pod 的 DNS 解析器选项。
搜索	数组（字符串）	host-name 查找的 DNS 搜索域列表。这将附加到 DNSPolicy 生成的基本搜索路径中。将删除重复的搜索路径。

14.12.1.94. .spec.dnsConfig.options

描述

DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。

类型

数组

14.12.1.95. .spec.dnsConfig.options[]

描述

PodDNSConfigOption 定义 pod 的 DNS 解析器选项。

类型

object

属性	类型	描述
name	字符串	必需。
value	string	

14.12.1.96. .spec.ephemeralContainers

描述

此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。

类型**数组****14.12.1.97. .spec.ephemeralContainers[]****描述**

EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。

要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。

类型**object****必填**

- **name**

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
命令	数组（字符串）	ENTRYPOINT 数组.不会在 shell 中执行。如果没有提供，则使用镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME)使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME)语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images

属性	类型	描述
imagePullPolicy	字符串	<p>镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签, 则默认为 Always, 否则为 IfNotPresent。无法更新。更多信息 :</p> <p>https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>可能枚举值 : - "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败, 则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在, 且拉取失败, 则容器将失败。- "Never" 表示 kubelet 不会拉取镜像, 但只使用本地镜像。如果镜像不存在, 容器将失败</p>
生命周期	object	<p>生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序, 在操作完成前管理容器块, 除非容器进程失败, 在这种情况下, 处理程序会中止。</p>
livenessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
name	字符串	<p>指定为 DNS_LABEL 的临时容器的名称。此名称在所有容器、init 容器和临时容器中必须是唯一的。</p>
ports	array	<p>临时容器不允许使用端口。</p>
ports[]	对象	<p>containerPort 代表单一容器中的网络端口。</p>
readinessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
resizePolicy	数组	<p>容器的资源重新定义策略大小。</p>
resizePolicy[]	object	<p>ContainerResizePolicy 代表容器的资源大小策略。</p>

属性	类型	描述
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
targetContainerName	字符串	<p>如果设置，则此临时容器目标的 PodSpec 中的容器名称。临时容器将在此容器的命名空间中运行 (IPC、PID 等)。如果没有设置，则临时容器将使用 Pod spec 中配置的命名空间。</p> <p>容器运行时必须实施对此功能的支持。如果运行时不支持以命名空间为目标，则设置此字段的结果未定义。</p>

属性	类型	描述
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。

属性	类型	描述
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.12.1.98. .spec.ephemeralContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.12.1.99. .spec.ephemeralContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的一个源。

14.12.1.100. .spec.ephemeralContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.12.1.101. .spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- key

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.12.1.102. .spec.ephemeralContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- fieldPath

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.12.1.103. .spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.12.1.104. .spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.12.1.105. .spec.ephemeralContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.12.1.106. .spec.ephemeralContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.12.1.107. .spec.ephemeralContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.12.1.108. .spec.ephemeralContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.12.1.109. .spec.ephemeralContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

属性	类型	描述
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.12.1.110. .spec.ephemeralContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.12.1.111. .spec.ephemeralContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录（'/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（' ' 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.112. .spec.ephemeralContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.113. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.12.1.114. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**

- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.115. .spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.116. .spec.ephemeralContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
HttpGet	object	HttpGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.12.1.117. .spec.ephemeralContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.118. .spec.ephemeralContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头

属性	类型	描述
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.119. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.12.1.120. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.121. .spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.122. .spec.ephemeralContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.123. .spec.ephemeralContainers[].livenessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root（'/'）。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（' ' 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.124. .spec.ephemeralContainers[].livenessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.125. .spec.ephemeralContainers[].livenessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.126. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.12.1.127. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.128. .spec.ephemeralContainers[].livenessProbe.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.129. .spec.ephemeralContainers[].ports**描述**

临时容器不允许使用端口。

类型

数组

14.12.1.130. .spec.ephemeralContainers[].ports[]**描述**

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.12.1.131. .spec.ephemeralContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.132. .spec.ephemeralContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root（/）。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（ 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.133. .spec.ephemeralContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.134. .spec.ephemeralContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.135. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.12.1.136. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.137. .spec.ephemeralContainers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.138. .spec.ephemeralContainers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.12.1.139. .spec.ephemeralContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- resourceName
- restartPolicy

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.12.1.140. .spec.ephemeralContainers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.141. .spec.ephemeralContainers[].resources.claims**描述**

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.142. .spec.ephemeralContainers[].resources.claims[]**描述**

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.143. .spec.ephemeralContainers[].securityContext

描述

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型

object

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行行为 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.12.1.144. .spec.ephemeralContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.12.1.145. .spec.ephemeralContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.12.1.146. .spec.ephemeralContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.12.1.147. .spec.ephemeralContainers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.12.1.148. .spec.ephemeralContainers[].startupProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.149. .spec.ephemeralContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.12.1.150. .spec.ephemeralContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.12.1.151. .spec.ephemeralContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.12.1.152. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.12.1.153. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.154. .spec.ephemeralContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.155. .spec.ephemeralContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.12.1.156. .spec.ephemeralContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- name
- devicePath

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.12.1.157. .spec.ephemeralContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。

类型

数组

14.12.1.158. .spec.ephemeralContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- name
- mountPath

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 `.`。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.12.1.159. .spec.hostAliases

描述

hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。

类型

数组

14.12.1.160. .spec.hostAliases[]

描述

HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。

类型

object

属性	类型	描述
主机名	数组（字符串）	以上 IP 地址的主机名。
ip	字符串	主机文件条目的 IP 地址。

14.12.1.161. .spec.imagePullSecrets

描述

imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息：

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

类型

array

14.12.1.162. .spec.imagePullSecrets[]

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.163. .spec.initContainers

描述

属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息：

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

类型

array

14.12.1.164. .spec.initContainers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- **name**

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组（字符串）	ENTRYPOINT 数组。不会在 shell 中执行。如果没有提供，则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败

属性	类型	描述
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。
ports	array	要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

属性	类型	描述
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。

属性	类型	描述
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.12.1.165. .spec.initContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

array

14.12.1.166. .spec.initContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。

属性	类型	描述
value	字符串	变量引用 $\$(VAR_NAME)$ 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 $\$$ ，它允许转义 $\$(VAR_NAME)$ 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVarar 的值的一个源。

14.12.1.167. .spec.initContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVarar 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.12.1.168. .spec.initContainers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.12.1.169. .spec.initContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.12.1.170. .spec.initContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的

属性	类型	描述
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.12.1.171. .spec.initContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.12.1.172. .spec.initContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

array

14.12.1.173. .spec.initContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.12.1.174. .spec.initContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.12.1.175. .spec.initContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.12.1.176. .spec.initContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.12.1.177. .spec.initContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
HttpGet	object	HttpGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.12.1.178. .spec.initContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组 (字符串)	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.179. .spec.initContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.180. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.181. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.182. .spec.initContainers[].lifecycle.postStart.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.183. .spec.initContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCP Socket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCP SocketAction 描述了基于打开套接字的操作

14.12.1.184. .spec.initContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.185. .spec.initContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.186. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.187. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.188. .spec.initContainers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.189. .spec.initContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

属性	类型	描述
terminationGracePeriodSeconds	整数	<p>可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code>。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。</p> <p><code>spec.terminationGracePeriodSeconds</code>。</p>
timeoutSeconds	整数	<p>探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</p>

14.12.1.190. `.spec.initContainers[].livenessProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	<p>命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。</p>

14.12.1.191. `.spec.initContainers[].livenessProbe.grpc`

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.192. .spec.initContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.193. .spec.initContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.194. .spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.195. .spec.initContainers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object**必填**

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.196. .spec.initContainers[].ports**描述**

要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

array

14.12.1.197. .spec.initContainers[].ports[]**描述**

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.12.1.198. .spec.initContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.199. .spec.initContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.12.1.200. .spec.initContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.12.1.201. .spec.initContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.12.1.202. .spec.initContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.203. .spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.204. .spec.initContainers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.205. .spec.initContainers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.12.1.206. .spec.initContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- resourceName
- restartPolicy

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.12.1.207. .spec.initContainers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.208. .spec.initContainers[].resources.claims**描述**

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.209. .spec.initContainers[].resources.claims[]**描述**

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.210. .spec.initContainers[].securityContext**描述**

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型**object**

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行行为 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.12.1.211. .spec.initContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.12.1.212. .spec.initContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.12.1.213. .spec.initContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.12.1.214. .spec.initContainers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.12.1.215. .spec.initContainers[].startupProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.12.1.216. .spec.initContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.12.1.217. .spec.initContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.12.1.218. .spec.initContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.12.1.219. .spec.initContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

array

14.12.1.220. .spec.initContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.12.1.221. .spec.initContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.12.1.222. .spec.initContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

array

14.12.1.223. .spec.initContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.12.1.224. .spec.initContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。无法更新。

类型

array

14.12.1.225. .spec.initContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- **name**
- **mountPath**

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 `.`。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.12.1.226. .spec.os

描述

PodOS 定义 pod 的 OS 参数。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是操作系统的名称。目前支持的值有 linux 和 windows。额外值可能会在以后定义，可以是： https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration 客户端应该应该处理附加值，并将此字段中的未识别的值视为 os: null

14.12.1.227. .spec.readinessGates**描述**

如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪：<https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

类型

数组

14.12.1.228. .spec.readinessGates[]**描述**

PodReadinessGate 包含对 pod 条件的引用

类型

object

必填

- **conditionType**

属性	类型	描述
conditionType	字符串	conditionType 指的是 pod 条件列表中具有匹配类型的条件。

14.12.1.229. .spec.resourceClaims**描述**

ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。

这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。

类型

数组

14.12.1.230. .spec.resourceClaims[]

描述

PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

类型

object

必填

- name

属性	类型	描述
name	string	在 pod 中唯一标识此资源声明的名称。这必须是 DNS_LABEL。
source	object	ClaimSource 描述了对 ResourceClaim 的引用。 应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

14.12.1.231. .spec.resourceClaims[].source

描述

ClaimSource 描述了对 ResourceClaim 的引用。
应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

类型

object

属性	类型	描述
resourceClaimName	string	ResourceClaimName 是与这个 pod 相同的命名空间中的 ResourceClaim 对象的名称。

属性	类型	描述
resourceClaimTemplateName	string	<p>ResourceClaimTemplateName 是与这个 pod 相同的命名空间中的 ResourceClaimTemplate 对象的名称。</p> <p>模板将用于创建新的 ResourceClaim，它将绑定到此 pod。删除此 pod 后，ResourceClaim 也会被删除。ResourceClaim 的名称将是 <pod name>-<resource name>，其中 <resource name> 是 PodResourceClaim.Name。如果串联名称对 ResourceClaim 无效（如太长，则 Pod 验证将拒绝 pod）。</p> <p>没有为 pod 拥有该名称的现有 ResourceClaim 不会被用于 pod，以避免错误地使用不相关的资源。然后，调度和 pod 启动会被阻断，直到删除不相关的 ResourceClaim。</p> <p>此字段不可变，在创建 ResourceClaim 后，control plane 不会对对应的 ResourceClaim 进行更改。</p>

14.12.1.232. .spec.schedulingGates

描述

SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。SchedulingGates 只能在创建 pod 时设置，之后只能删除。

这是 PodSchedulingReadiness 功能门启用的 beta 功能。

类型

数组

14.12.1.233. .spec.schedulingGates[]

描述

PodSchedulingGate 与 Pod 关联，以保护其调度。

类型

object

必填

- name

属性	类型	描述
name	string	调度授权的名称。每个调度最低要求都必须具有唯一的 name 字段。

14.12.1.234. .spec.securityContext

描述

PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。

类型

object

属性	类型	描述
fsGroup	整数	<p>适用于 pod 中所有容器的特殊补充组。有些卷类型允许 Kubelet 将该卷的所有权更改为 pod 拥有：</p> <ol style="list-style-type: none"> 1.所属 GID 将是 FSGroup 2. set setgid 位（在卷中创建的新文件将归 FSGroup 所有） 3. 权限位为 OR'd with rw-rw---- <p>如果未设置，Kubelet 将不修改任何卷的所有权和权限。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
fsGroupChangePolicy	字符串	<p>fsGroupChangePolicy 定义在 Pod 中公开卷之前更改卷的所有权和权限的行为。此字段仅适用于支持 fsGroup 基于所有权（和权限）的卷类型。它对临时卷类型没有影响，如：secret、configmaps 和 emptydir。有效值为 "OnRootMismatch" 和 "Always"。如果没有指定，则使用 "Always"。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值： - "Always" 表示当卷挂载到 Pod 中时，应始终更改卷的所有权和权限。这个默认行为。 - "OnRootMismatch" 表示只有在根目录的权限与卷上预期权限不匹配时才会更改卷的所有权和权限。这有助于缩短更改卷的所有权和权限所需的时间。</p>

属性	类型	描述
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

属性	类型	描述
supplementalGroups	数组（整数）	除了容器的主 GID、fsGroup（如果指定）以及容器镜像中定义的组成员资格（如果指定）之外，应用于每个容器中的第一个进程运行的组列表。如果未指定，则不会将额外的组添加到任何容器中。请注意，容器进程的 uid 中定义的组成员资格仍然有效，即使它们没有包括在此列表中。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls	array	sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls[]	对象	sysctl 定义要设置的内核参数
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.12.1.235. .spec.securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.12.1.236. .spec.securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.12.1.237. .spec.securityContext.sysctls

描述

sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。

类型

array

14.12.1.238. .spec.securityContext.sysctls[]

描述

sysctl 定义要设置的内核参数

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	字符串	要设置的属性的名称
value	字符串	要设置的属性值

14.12.1.239. .spec.securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。

属性	类型	描述
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.12.1.240. .spec.tolerations

描述

如果指定，pod 的容限。

类型

array

14.12.1.241. .spec.tolerations[]

描述

附加到此 Toleration 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
effect	字符串	<p>effect 表示污点效果要匹配。空意味着匹配所有污点效果。指定后，允许的值为 NoSchedule、PreferNoSchedule 和 NoExecute。</p> <p>可能枚举值： - "NoExecute" Evict 任何已在运行的 pod，这些 pod 不容许该污点。目前由 NodeController 实施。 - "NoSchedule" 不允许新 pod 调度到该节点上，除非它们容许污点，但允许提交给 Kubelet 的所有 pod 而无需启动调度程序，并允许所有已在运行的 pod 继续运行。调度程序强制使用 - "PreferNoSchedule" Like TaintEffectNoSchedule，但调度程序会尝试不将新 pod 调度到节点上，而不是禁止新 pod 完全调度到该节点上。由调度程序实施。</p>
key	字符串	<p>key 是容限应用到的污点键。empty 表示与所有污点键匹配。如果键为空，则运算符必须是 Exists；此组合意味着匹配所有值和所有键。</p>
operator	字符串	<p>Operator 代表键与值的关系。有效的运算符是 Exists 和 Equal。默认为 Equal。exists 等同于 value 的通配符，以便 pod 可以容忍特定类别的所有污点。</p> <p>可能枚举值： - "Equal" - "Exists"</p>
tolerationSeconds	整数	<p>TolerationSeconds 代表容限的期间（该容限必须是 NoExecute，否则将忽略此字段）容许污点。默认情况下，它没有被设置，这意味着容许永久污点（不驱除）。系统会将零值和负值视为 0（立即撤离）。</p>
value	字符串	<p>值是容限匹配的污点值。如果运算符是 Exists，则该值应为空，否则只是一个常规字符串。</p>

14.12.1.242. .spec.topologySpreadConstraints

描述

topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。

类型

array

14.12.1.243. .spec.topologySpreadConstraints[]**描述**

TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。

类型

object

必填

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

属性	类型	描述
labelSelector	labelSelector	labelSelector 用于查找匹配的 pod。与此标签选择器匹配的 Pod 被计算，以确定其对应拓扑域中的 pod 数量。
matchLabelKeys	数组（字符串）	matchLabelKeys 是一组 pod 标签键，用于选择要计算分布的 pod。键用于从传入的 pod 标签中查找值，这些键值标签带有 labelSelector 的 ANDed 来选择要针对传入 pod 计算的现有 pod 组。禁止在 MatchLabelKeys 和 LabelSelector 中存在相同的键。当 LabelSelector 没有设置时，无法设置 matchLabelKeys。传入的 pod 标签中不存在的键将被忽略。null 或 empty 列表表示仅匹配 labelSelector。 这是一个 beta 字段，需要启用 MatchLabelKeysInPodTopologySpread 功能门（默认启用）。

属性	类型	描述
maxSkew	整数	<p>maxSkew 描述了 pod 可能不均匀分布的程度。当 whenUnsatisfiable=DoNotSchedule 时，目标拓扑中匹配 pod 数量和全局最小值之间允许的最大区别。如果有资格的域数量小于 MinDomains，则全局最小值是符合域或零的匹配 pod 的最小数量。例如，在 3 区集群中，MaxSkew 设置为 1，具有相同 labelSelector 的 pod 分布为 2/2/1：在这种情况下，全局最小值为 1。 zone1 zone2 zone3 P P P P P P P P P - 如果 MaxSkew 为 1，则传入的 pod 只能调度到 zone3，以变为 2/2/2；将其调度到 zone1 (zone2)上的 ActualSkew (3-1)违反 MaxSkew (1)- if MaxSkew.当 whenUnsatisfiable=ScheduleAnyway 时，用来为满足它的拓扑提供更高优先级。它是必填字段。默认值为 1，不允许使用 0。</p>

属性	类型	描述
minDomains	整数	<p>MinDomains 表示最少的合格域数。当匹配拓扑键的合格域数量小于 minDomains 时，Pod Topology Spread 会将"global minimum"视为 0，然后执行 Skew 计算。当匹配拓扑键的合格域数量等于或大于 minDomains 时，这个值对调度没有影响。因此，当有资格的域数量小于 minDomains 时，调度程序不会将这些域的 maxSkew Pod 调度到这些域。如果值为 nil，则约束的行为就像 MinDomains 等于 1。有效值为大于 0 的整数。当值不是 nil 时，WhenUnsatisfiable 必须是 DoNotSchedule。</p> <p>例如，在 3 区集群中，MaxSkew 设置为 2，MinDomains 被设置为 5，而带有与 2/2/2: zone1 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone3 zone3 zone3 区域数量小于 5 (MinDomains)，因此"global minimum"被视为 0。在这种情况下，无法调度具有相同 labelSelector 的新 pod，因为如果新 Pod 调度到三个区，则 computed skew 将为 3 (3 - 0)，它将违反 MaxSkew。</p> <p>这是一个 beta 字段，需要启用 MinDomainsInPodTopologySpread 功能门（默认启用）。</p>

属性	类型	描述
nodeAffinityPolicy	string	<p>NodeAffinityPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理 Pod 的 nodeAffinity/nodeSelector。选项为：- Honor：计算中仅包含与 nodeAffinity/nodeSelector 匹配的节点。忽略 Ignore: nodeAffinity/nodeSelector。所有节点包含在计算中。</p> <p>如果这个值为 nil，则行为等同于 Honor 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>
nodeTaintsPolicy	string	<p>NodeTaintsPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理节点污点。选项为：- Honor: 没有污点的节点，以及传入 pod 具有容量的污点节点被包含。- Ignore: node taint is ignored。包括所有节点。</p> <p>如果这个值为 nil，则行为等同于 Ignore 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>

属性	类型	描述
topologyKey	字符串	<p>topologyKey 是节点标签的密钥。带有具有此键和相同值标签的节点被视为在同一拓扑中。我们将每个 <key, value> 视为 "bucket", 并尝试在每个存储桶中均衡的 pod 数量。我们将域定义为拓扑的特定实例。另外, 我们定义了一个有资格的域, 其节点满足 nodeAffinityPolicy 和 nodeTaintsPolicy 的要求。例如, 如果 TopologyKey 为 "kubernetes.io/hostname", 则每个节点都是该拓扑的域。如果 TopologyKey 是 "topology.kubernetes.io/zone", 则每个区都是该拓扑的域。它是必填字段。</p>
whenUnsatisfiable	字符串	<p>whenUnsatisfiable 表示在不满足分散约束的情况下如何处理 pod。</p> <ul style="list-style-type: none"> - DoNotSchedule (默认) 告知调度程序不调度它。 - ScheduleAnyway 告知调度程序将 pod 调度到任何位置, 但为拓扑赋予更高优先级, 有助于减少偏移量。只有在每个可能的节点分配在某些拓扑中违反 "MaxSkew" 时, 对传入 pod 的一个约束被视为 "Unsatisfiable"。例如, 在 3 区集群中, MaxSkew 设置为 1, 具有相同 labelSelector 的 pod 分布为 3/1/1: zone1 zone3 zone3 P P P P If WhenUnsatisfiable 被设置为 DoNotSchedule, 传入的 pod 只能调度到 zone2 (zone3) 要成为 3/2/1 (3/1/2), 在 zone2 (zone3) 上作为 ActualSkew (2-1) 满足 MaxSkew (1)。换句话说, 集群仍然可以进行平衡, 但调度程序不会使其更不平衡。它是必填字段。 <p>可能枚举值: -</p> <ul style="list-style-type: none"> "DoNotSchedule" 指示调度程序在不满足约束时不调度 pod。 "ScheduleAnyway" 指示调度程序调度 pod, 即使没有满足限制。

14.12.1.244. .spec.volumes

描述

可由属于 pod 的容器挂载的卷列表。更多信息：
<https://kubernetes.io/docs/concepts/storage/volumes>

类型

array

14.12.1.245. .spec.volumes[]

描述

卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

类型

object

必填

- **name**

属性	类型	描述
awsElasticBlockStore	object	代表 AWS 中的 Persistent Disk 资源。 挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。
azureDisk	对象	azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。
azureFile	对象	azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。
cephfs	object	代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。
cinder	object	代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

属性	类型	描述
configMap	object	<p>将 ConfigMap 调整为卷。</p> <p>目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。</p>
csi	object	代表由外部 CSI 驱动程序管理的卷的源位置
downwardAPI	object	DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。
emptyDir	object	表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。
ephemeral	object	代表由普通存储驱动程序处理的临时卷。
fc	object	代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。
flexVolume	对象	FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。
flocker	object	代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。
gcePersistentDisk	object	<p>代表 Google Compute Engine 中的 Persistent Disk 资源。</p> <p>在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。</p>

属性	类型	描述
gitRepo	object	代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。 DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。
glusterfs	object	代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。
hostPath	object	代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。
iscsi	object	代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。
name	string	卷的名称。必须是 DNS_LABEL，且在 pod 中唯一。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。 PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。
photonPersistentDisk	object	代表 Photon Controller 持久磁盘资源。
portworxVolume	object	PortworxVolumeSource 代表一个 Portworx volume 资源。
projected	object	代表投射卷源

属性	类型	描述
quobyte	object	代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。
rbd	object	代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。
scaleIO	object	ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷
secret	object	将 Secret 调整为卷。 目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。
storageos	object	代表 StorageOS 持久性卷资源。
vsphereVolume	object	代表 vSphere 卷资源。

14.12.1.246. .spec.volumes[].awsElasticBlockStore

描述

代表 AWS 中的 Persistent Disk 资源。

挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4"，"xfs"，"ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

属性	类型	描述
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。
readOnly	布尔值	readonly 值 true 将强制 VolumeMounts 中的 readOnly 设置。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	VolumeID 是 AWS (Amazon EBS 卷) 中的持久磁盘资源的唯一 ID。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

14.12.1.247. .spec.volumes[].azureDisk

描述

azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

类型

object

必填

- diskName
- diskURI

属性	类型	描述
cacheMode	string	cacheMode 是主机缓存模式：None、Read Only、Read Write。 可能枚举值：- "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskname 是 blob 存储中的数据磁盘的名称
diskURI	string	diskURI 是 blob 存储中数据磁盘的 URI

属性	类型	描述
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
kind	string	kind expected 值是 Shared: multiple blob disk per storage account Managed: azure disk per storage account Managed: azure managed data disk (only in managed availability set))。默认为 shared possible enum 值: - "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。

14.12.1.248. .spec.volumes[].azureFile

描述

azureFile 代表主机上的 Azure File Service 挂载, 并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

属性	类型	描述
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称

属性	类型	描述
shareName	string	sharename 是 azure 共享名称

14.12.1.249. .spec.volumes[].cephfs

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中 找到引用的对象。

属性	类型	描述
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

14.12.1.250. .spec.volumes[].cephfs.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.251. .spec.volumes[].cinder

描述

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。示例："ext4"，"xfs"，"ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

属性	类型	描述
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
volumeID	string	用于识别 cinder 中的卷的卷 ID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

14.12.1.252. .spec.volumes[].cinder.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.253. .spec.volumes[].configMap

描述

将 ConfigMap 调整为卷。

目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选的：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.12.1.254. .spec.volumes[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

array

14.12.1.255. .spec.volumes[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.12.1.256. .spec.volumes[].csi

描述

代表由外部 CSI 驱动程序管理的卷的源位置

类型

object

必填

- **driver**

属性	类型	描述
driver	string	driver 是处理这个卷的 CSI 驱动程序的名称。请参考您的管理员获取集群中注册的正确名称。

属性	类型	描述
fsType	string	要挂载的 fstype。Ex. "ext4", "xfs", "ntfs".如果没有提供, 空值会被传递给关联的 CSI 驱动程序, 该驱动程序将决定要应用的默认文件系统。
nodePublishSecretRef	object	LocalObjectReference 包含足够的信息, 供您在同一个命名空间中找到引用的对象。
readOnly	布尔值	readonly 指定卷的只读配置。默认为 false (读/写)。
volumeAttributes	对象 (字符串)	volumeAttributes 存储传递给 CSI 驱动程序的特定于驱动程序的属性。如需支持的值, 请参考您的驱动程序文档。

14.12.1.257. .spec.volumes[].csi.nodePublishSecretRef

描述

LocalObjectReference 包含足够的信息, 供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.258. .spec.volumes[].downwardAPI

描述

DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	可选：默认在创建的文件中使用的模式位。必须是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	array	items 是 Downward API 卷文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.12.1.259. .spec.volumes[].downwardAPI.items

描述

items 是 Downward API 卷文件列表

类型

array

14.12.1.260. .spec.volumes[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- path

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。

属性	类型	描述
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.12.1.261. .spec.volumes[].downwardAPI.items[].fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.12.1.262. .spec.volumes[].downwardAPI.items[].resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- resource

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.12.1.263. .spec.volumes[].emptyDir

描述

表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
中	string	Medium 代表这个目录应该支持哪些存储类型。默认值为 "", 它需要使用节点的默认介质。必须是空字符串（默认）或 Memory。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit	数量	sizeLimit 是此 EmptyDir 卷所需的本地存储总量。大小限制也适用于内存介质。内存用量 EmptyDir 的最大用法是此处指定的 SizeLimit 和 pod 中所有容器的内存限值总和。默认值为 nil，表示限制未定义。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir

14.12.1.264. .spec.volumes[].ephemeral

描述

代表由普通存储驱动程序处理的临时卷。

类型

object

属性	类型	描述
volumeClaimTemplate	object	PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

14.12.1.265. .spec.volumes[].ephemeral.volumeClaimTemplate

描述

PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

类型

object

必填

- **spec**

属性	类型	描述
metadata	ObjectMeta	可以包含在创建时复制到 PVC 中的标签和注解。不允许其他字段，并在验证过程中被拒绝。
spec	对象	PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

14.12.1.266. .spec.volumes[].ephemeral.volumeClaimTemplate.spec

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含卷应具有的所需访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	对象	TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

属性	类型	描述
dataSourceRef	object	<p>dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。</p>
resources	对象	ResourceRequirements 描述计算资源要求。
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

属性	类型	描述
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统进行或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

14.12.1.267. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

描述

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

14.12.1.268. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

描述

dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。

dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称
namespace	string	namespace 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 gateway.networking.k8s.io/ReferenceGrant 对象，以允许该命名空间的所有者接受引用。详情请查看 ReferenceGrant 文档。(alpha)此字段需要启用 CrossNamespaceVolumeDataSource 功能门。

14.12.1.269. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.270. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.271. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claims[]

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- name

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.272. .spec.volumes[].fc

描述

代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
LUN	整数	LUN 是可选的: FC 目标 lun 号
readOnly	布尔值	readonly 是可选的: 默认为 false (读/写)。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组 (字符串)	targetWWN 可选: FC 目标全球名称(WWN)
wwids	数组 (字符串)	wwids 可选: FC 卷全局广泛的标识符(wwids) Either wwids 或 targetWWNs 和 lun 的组合必须同时设置, 但不能同时设置。

14.12.1.273. .spec.volumes[].flexVolume

描述

FlexVolume 代表一个通用卷资源, 它使用基于 exec 的插件置备/附加。

类型

object

必填

- driver

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认文件系统取决于 FlexVolume 脚本。
options	对象 (字符串)	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，让您在同一个命名空间中找到引用的对象。

14.12.1.274. `.spec.volumes[].flexVolume.secretRef`

描述

LocalObjectReference 包含足够的信息，让您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.275. `.spec.volumes[].flocker`

描述

代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。

类型

object

属性	类型	描述
datasetName	string	datasetName 是存储在 Flocker 的 dataset 的 dataset 的 dataset 的 Name。
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

14.12.1.276. .spec.volumes[].gcePersistentDisk

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **pdName**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

属性	类型	描述
pdName	string	pdName 是 GCE 中的 PD 资源的唯一名称。用于识别 GCE 中的磁盘。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

14.12.1.277. .spec.volumes[].gitRepo

描述

代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。
DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。

类型

object

必填

- 软件仓库

属性	类型	描述
目录	string	directory 是目标目录名称。不得包含或以 './' 开头。如果提供了 '.', 则卷目录将是 git 存储库。否则，如果指定，卷将包含具有指定名称的子目录中的 git 存储库。
软件仓库	string	repository 是 URL
revision	string	revision 是指定修订版本的提交哈希。

14.12.1.278. .spec.volumes[].glusterfs

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

14.12.1.279. .spec.volumes[].hostPath

描述

代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- path

属性	类型	描述
path	string	主机上目录的路径。如果路径是符号链接，它将遵循到实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath

属性	类型	描述
type	string	<p>HostPath Volume Defaults 为 "" more info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>可能枚举值：- "" 对于向后兼容，如果未设置 - "BlockDevice" A 块设备必须存在于给定路径 - "CharDevice" A 字符设备必须存在于给定路径上 - "Directory" A 目录必须存在于给定路径 - "DirectoryOrCreate" 中（如果给定路径不存在），将根据需要创建一个空目录，且文件模式为 0755，其具有与 Kubelet 相同的组和所有权。- "File" A 文件必须存在于给定路径上 - "FileOrCreate" 如果给定路径上不存在，将根据需要创建一个空文件，该文件与文件模式 0644 具有相同的组和所有权。- "Socket" A UNIX 套接字必须存在于给定路径上。</p>

14.12.1.280. .spec.volumes[].iscsi

描述

代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是目标 iSCSI 限定名称。
iscsiInterface	string	iscsiInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 代表 iSCSI 目标 Lun 编号。
门户	数组（字符串）	门户是 iSCSI 目标门户列表。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。

14.12.1.281. .spec.volumes[].iscsi.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.282. .spec.volumes[].nfs

描述

代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **server**
- **path**

属性	类型	描述
path	string	NFS 服务器导出的路径。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	布尔值	此处的只读将强制使用只读权限挂载 NFS 导出。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server 是 NFS 服务器的主机名或 IP 地址。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs

14.12.1.283. .spec.volumes[].persistentVolumeClaim

描述

PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。

类型

object

必填

- **claimName**

属性	类型	描述
claimName	string	claimName 是使用此卷与 pod 相同的命名空间中的 PersistentVolumeClaim 的名称。 更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	布尔值	readonly 将强制 VolumeMount 中的 ReadOnly 设置。默认 false。

14.12.1.284. .spec.volumes[].photonPersistentDisk

描述

代表 Photon Controller 持久磁盘资源。

类型

object

必填

- **pdID**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
pdID	string	pdID 是标识 Photon Controller 持久磁盘的 ID

14.12.1.285. .spec.volumes[].portworxVolume

描述

PortworxVolumeSource 代表一个 Portworx volume 资源。

类型

object

必填

- **volumeID**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 代表要挂载的文件系统类型，必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs". 如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
volumeID	string	VolumeID 唯一标识一个 Portworx 卷

14.12.1.286. .spec.volumes[].projected

描述

代表投射卷源

类型

object

属性	类型	描述
defaultMode	整数	defaultMode 是默认用来对创建文件设置权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
sources	数组	sources 是卷投射列表
sources[]	对象	可与其他支持的卷类型一起投射的预测

14.12.1.287. .spec.volumes[].projected.sources

描述

sources 是卷投射列表

类型

array

14.12.1.288. `.spec.volumes[].projected.sources[]`

描述

可与其他支持的卷类型一起投射的预测

类型

object

属性	类型	描述
configMap	object	<p>将 ConfigMap 调整为投射卷。</p> <p>目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。</p>
downwardAPI	object	<p>代表下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。</p>
secret	object	<p>将 secret 适应到投射卷中。</p> <p>目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。</p>
serviceAccountToken	object	<p>ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。</p>

14.12.1.289. `.spec.volumes[].projected.sources[].configMap`

描述

将 ConfigMap 调整为投射卷。

目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.12.1.290. .spec.volumes[].projected.sources[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

array

14.12.1.291. .spec.volumes[].projected.sources[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.12.1.292. .spec.volumes[].projected.sources[].downwardAPI

描述

代表下下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。

类型

object

属性	类型	描述
items	array	items 是 DownwardAPIVolume 文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.12.1.293. .spec.volumes[].projected.sources[].downwardAPI.items

描述

items 是 DownwardAPIVolume 文件列表

类型

array

14.12.1.294. .spec.volumes[].projected.sources[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- **path**

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.12.1.295. `.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.12.1.296. .spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- resource

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.12.1.297. .spec.volumes[].projected.sources[].secret

描述

将 secret 适应到投射卷中。

目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。

类型

object

属性	类型	描述
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥

14.12.1.298. .spec.volumes[].projected.sources[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

array

14.12.1.299. .spec.volumes[].projected.sources[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。

属性	类型	描述
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.12.1.300. .spec.volumes[].projected.sources[].serviceAccountToken

描述

ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

类型

object

必填

- **path**

属性	类型	描述
受众	string	受众是令牌的预期受众。令牌的接收者必须使用令牌使用者中指定的标识符来识别自己，否则应拒绝令牌。使用者默认为 apiserver 的标识符。
expirationSeconds	整数	expirationSeconds 是服务帐户令牌请求持续时间。随着令牌方法到期，kubelet 卷插件将主动轮转服务帐户令牌。如果令牌存在的时间超过这个值的 80%，或者令牌超过 24 小时，则 kubelet 将开始尝试轮转令牌，且必须至少为 10 分钟。
path	string	path 是相对于文件挂载点的路径，以将令牌项目到其中。

14.12.1.301. .spec.volumes[].quobyte

描述

代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **registry**

- 卷

属性	类型	描述
group	string	将卷访问映射到 Default 的组没有组
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Quobyte 卷。默认为false。
registry	string	registry 代表一个或多个 Quobyte Registry 服务，作为字符串指定为 host:port 对（多个条目用逗号分隔），后者充当卷的中央 registry
tenant	string	在带有动态置备 Quobyte 卷的后端中拥有给定 Quobyte 卷的租户，值由插件设置
user	string	用户将卷访问权限映射到默认值到 serviceaccount 用户
卷	string	卷是一个字符串，它根据名称引用已创建的 Quobyte 卷。

14.12.1.302. .spec.volumes[].rbd

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **monitor**
- **image**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd

属性	类型	描述
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

14.12.1.303. .spec.volumes[].rbd.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.304. .spec.volumes[].scaleIO

描述

ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- **gateway**
- **system**
- **secretRef**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认为 "xfs".
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
sslEnabled	布尔值	sslEnabled Flag 启用/禁用与网关的 SSL 通信，默认为 false

属性	类型	描述
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

14.12.1.305. .spec.volumes[].scaleIO.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.306. .spec.volumes[].secret

描述

将 Secret 调整为卷。

目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥
secretName	string	secretName 是要使用的 pod 命名空间中的 secret 名称。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#secret

14.12.1.307. .spec.volumes[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

array

14.12.1.308. .spec.volumes[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.12.1.309. .spec.volumes[].storageos

描述

代表 StorageOS 持久性卷资源。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

属性	类型	描述
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

14.12.1.310. .spec.volumes[].storageos.secretRef

描述

LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.12.1.311. .spec.volumes[].vsphereVolume

描述

代表 vSphere 卷资源。

类型

object

必填

- **volumePath**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。
storagePolicyID	string	StoragePolicyID 是与 StoragePolicyName 关联的存储 Policy Based Management (SPBM)配置集 ID。
storagePolicyName	string	StoragePolicyName 是存储基于策略的管理(SPBM)配置集名称。
volumePath	string	volumePath 是标识 vSphere 卷 vmdk 的路径

14.12.1.312. .status

描述

PodStatus 代表有关 pod 状态的信息。状态可能会跟踪系统的实际状态, 特别是在托管 pod 的节点无法联系 control plane 时。

类型

object

属性	类型	描述
conditions	数组	pod 的当前服务状态。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions
conditions[]	object	PodCondition 包含此 pod 当前条件的详情。
containerStatuses	数组	列表中每个容器有一个条目。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status
containerStatuses[]	object	ContainerStatus 包含此容器当前状态的详细信息。
ephemeralContainerStatuses	数组	此 pod 中运行的任何临时容器的状态。

属性	类型	描述
ephemeralContainerStatuses []	object	ContainerStatus 包含此容器当前状态的详细信息。
hostIP	string	分配给 pod 的主机的 IP 地址。如果尚未调度，则为空。
initContainerStatuses	数组	列表中每个 init 容器有一个条目。最近成功的 init 容器将准备好 = true，最近启动的容器将设置 startTime。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status
initContainerStatuses[]	object	ContainerStatus 包含此容器当前状态的详细信息。
message	string	人类可读的消息，指示 pod 处于此条件的详情。
nominatedNodeName	string	只有在此 pod 抢占节点上的其他 pod 时，才会设置 nominatedNodeName，但它无法立即调度，因为抢占会收到其安全终止期限。此字段不能保证将 pod 调度到此节点上。如果其他节点更早可用，调度程序可能会决定将 pod 放置到其他位置。调度程序也可以决定将此节点上的资源赋予一个在抢占后创建的优先级更高的 pod。因此，当 pod 被调度时，此字段可能与 PodSpec.nodeName 不同。

属性	类型	描述
phase	string	<p>Pod 的阶段是一个简单的、高级别的总结，用于 Pod 在其生命周期中。conditions 数组、原因和消息字段和单个容器状态数组包含有关 pod 的状态的更多详情。可能存在五个阶段值：</p> <p>Pending：Kubernetes 系统接受 pod，但尚未创建一个或多个容器镜像。这包括调度前的时间，以及通过网络下载镜像的时间，这可能需要一些时间。Running: pod 已被绑定到某个节点，并且已创建所有容器。至少一个容器仍在运行，或者正在启动或重启过程中。</p> <p>Succeeded：pod 中的所有容器都成功终止，且不会重启。Failed：pod 中的所有容器都终止，至少一个容器出现故障终止。容器以非零状态退出，或者被系统终止。</p> <p>unknown：由于无法获取 pod 状态的原因，通常源自与 pod 主机通信的错误。</p> <p>更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-phase</p> <p>可能枚举值：- "Failed" 表示 pod 中的所有容器都已终止，并且至少有一个容器出现故障（以非零退出代码退出或被系统停止）。- "Pending" 表示 pod 已被系统接受，但一个或多个容器尚未启动。这包括在绑定到节点前的时间，以及将镜像提取到主机上的时间。- "Running" 表示 pod 已绑定到某个节点，并且所有容器都已启动。至少一个容器仍在运行或正在重启过程中。- "Succeeded" 表示 pod 中的所有容器都错误地以容器退出代码 0 终止，系统不会重启任何这些容器。"Unknown" 表示 pod 的状态无法获得，这通常是因为与 pod 主机通信的错误。弃用：它不是从 2015 年起设置 (74da3b14b0c0f658b3bb8d2def5094686d0e9095)</p>

属性	类型	描述
podIP	string	分配给 pod 的 IP 地址。在集群中至少可路由。如果尚未分配，则为空。
podIPs	数组	podIP 包含分配给 pod 的 IP 地址。如果指定了此字段，则 Oth 条目必须与 podIP 字段匹配。可以为每个 IPv4 和 IPv6 分配最多 1 个值。如果尚未分配 IP，则此列表为空。
podIPs[]	object	(复数) PodIPs 字段中条目的 IP 地址信息。每个条目包括： IP：分配给 pod 的 IP 地址。在集群中至少可路由。
qosClass	string	根据资源要求分配给 pod 的服务质量(QoS)分类，请参阅可用 QOS 类更多信息的 PodQOSClass 类型： https://kubernetes.io/docs/concepts/workloads/pods/pod-qos/#quality-of-service-classes 可能枚举值：- "BestEffort" 是 BestEffort qos 类。- "Burstable" 是 Burstable qos 类。- "Guaranteed" 是 Guaranteed qos 类。
reason	string	简短 CamelCase 消息指示 pod 处于此状态的详情。例如：'evicted'
调整大小	string	pod 容器所需的资源状态。如果没有资源调整大小，则为空。对容器资源的任何更改将自动将其设置为 "Proposed"
startTime	时间	Kubelet 确认对象的 RFC 3339 日期和时间。这是在 Kubelet 拉取 pod 容器镜像之前。

14.12.1.313. .status.conditions

描述

pod 的当前服务状态。更多信息：<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions>

类型

array

14.12.1.314. .status.conditions[]

描述

PodCondition 包含此 pod 当前条件的详情。

类型

object

必填

- type
- status

属性	类型	描述
lastProbeTime	时间	我们最后一次探测到条件的时间。
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	条件最后一次转换的唯一、一词、CamelCase 原因。
status	字符串	status 是条件的状态。可以是 True, False, Unknown。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions
type	字符串	type 是条件的类型。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions

14.12.1.315. .status.containerStatuses

描述

列表中每个容器有一个条目。更多信息：

<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status>

类型

数组

14.12.1.316. `.status.containerStatuses[]`

描述

ContainerStatus 包含此容器当前状态的详细信息。

类型

object

必填

- **name**
- **ready**
- **restartCount**
- **image**
- **imageID**

属性	类型	描述
allocatedResources	对象 (数量)	AllocatedResources 代表节点为此容器分配的計算资源。kubelet 在成功 pod 准入时将此值设置为 Container.Resources.Requests, 并在成功接受所需的 pod 大小后将其设置为 Container.Resources.Requests。
containerID	string	containerID 是容器的 ID, 格式为 ' <code><type>://<container_id></code> '。其中 type 是容器运行时标识符, 从 CRI API 的 Version 调用返回 (例如 "containerd")。
image	string	image 是容器运行的容器镜像的名称。容器镜像可能与 PodSpec 中使用的镜像不匹配, 因为它可能已被运行时解析。更多信息: https://kubernetes.io/docs/concepts/containers/images
imageID	string	imageID 是容器镜像的镜像 ID。镜像 ID 可能与 PodSpec 中使用的镜像 ID 不匹配, 因为它可能已被运行时解决。
lastState	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定, 则默认是 ContainerStateWaiting。

属性	类型	描述
name	string	name 是一个 DNS_LABEL，代表容器的唯一名称。pod 中的每个容器必须在所有容器类型之间具有唯一的名称。无法更新。
ready	布尔值	ready 指定容器当前是否通过就绪度检查。当就绪度探测保持执行时，该值将改变。如果没有指定就绪度探测，则此字段在容器完全启动后默认为 true（请参阅 Started 字段）。 该值通常用于确定容器是否准备好接受流量。
resources	对象	ResourceRequirements 描述计算资源要求。
restartCount	整数	restartCount 保存容器重启的次数。kubelet 努力始终递增值，但在某些情况下，状态可能会因为节点重启而丢失，然后该值可能会重置为 0。该值永远不会为负数。
started	布尔值	started 指示容器是否已完成其 postStart 生命周期 hook 并通过其启动探测。初始化为 false，在 startupProbe 被视为成功后变为 true。当容器重启时，或者 kubelet 暂时丢失状态时，重置为 false。在这两种情况下，启动探测将再次运行。当未定义 startupProbe 且容器正在运行并且传递了 postStart 生命周期 hook 时，始终为 true。null 值必须与 false 相同。
state	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

14.12.1.317. .status.containerStatuses[].lastState

描述

ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

类型

object

属性	类型	描述
running	object	ContainerStateRunning 是容器的运行状态。
已终止	object	ContainerStateTerminated 是容器的终止状态。
等待	object	ContainerStateWaiting 是容器的等待状态。

14.12.1.318. .status.containerStatuses[].lastState.running

描述

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.319. .status.containerStatuses[].lastState.terminated

描述

ContainerStateTerminated 是容器的终止状态。

类型

object

必填

- **exitCode**

属性	类型	描述
containerID	string	容器 ID, 格式为 ' <code><type>://<container_id></code> '
exitCode	整数	容器最后一次终止的退出状态
finishedAt	时间	容器最后一次终止的时间
message	string	有关容器最后一次终止的消息
reason	string	(brief)容器最后一次终止的原因

属性	类型	描述
signal	整数	容器最后一次终止的信号
startedAt	时间	容器以前执行的时间启动

14.12.1.320. .status.containerStatuses[].lastState.waiting

描述

ContainerStateWaiting 是容器的等待状态。

类型

object

属性	类型	描述
message	string	有关容器尚未运行的原因的消息。
reason	string	(brief)容器没有运行的原因。

14.12.1.321. .status.containerStatuses[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

属性	类型	描述
limits	对象 (数量)	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象 (数量)	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.322. .status.containerStatuses[].resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.323. .status.containerStatuses[].resources.claims[]

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- name

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.324. .status.containerStatuses[].state

描述

ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

类型

object

属性	类型	描述
running	object	ContainerStateRunning 是容器的运行状态。
已终止	object	ContainerStateTerminated 是容器的终止状态。
等待	object	ContainerStateWaiting 是容器的等待状态。

14.12.1.325. .status.containerStatuses[].state.running**描述**

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.326. .status.containerStatuses[].state.terminated**描述**

ContainerStateTerminated 是容器的终止状态。

类型

object

必填

- **exitCode**

属性	类型	描述
containerID	string	容器 ID，格式为 ' <code><type>://<container_id></code> '
exitCode	整数	容器最后一次终止的退出状态

属性	类型	描述
finishedAt	时间	容器最后一次终止的时间
message	string	有关容器最后一次终止的消息
reason	string	(brief)容器最后一次终止的原因
signal	整数	容器最后一次终止的信号
startedAt	时间	容器以前执行的时间启动

14.12.1.327. .status.containerStatuses[].state.waiting

描述

ContainerStateWaiting 是容器的等待状态。

类型

object

属性	类型	描述
message	string	有关容器尚未运行的原因的消息。
reason	string	(brief)容器没有运行的原因。

14.12.1.328. .status.ephemeralContainerStatuses

描述

此 pod 中运行的任何临时容器的状态。

类型

数组

14.12.1.329. .status.ephemeralContainerStatuses[]

描述

ContainerStatus 包含此容器当前状态的详细信息。

类型

object

必填

- name
- ready
- restartCount
- image

- **imageID**

属性	类型	描述
allocatedResources	对象 (数量)	AllocatedResources 代表节点为此容器分配的計算资源。kubelet 在成功 pod 准入时将此值设置为 Container.Resources.Requests, 并在成功接受所需的 pod 大小后将其设置为 Container.Resources.Requests。
containerID	string	containerID 是容器的 ID, 格式为 '<type>://<container_id>'。其中 type 是容器运行时标识符, 从 CRI API 的 Version 调用返回 (例如 "containerd")。
image	string	image 是容器运行的容器镜像的名称。容器镜像可能与 PodSpec 中使用的镜像不匹配, 因为它可能已被运行时解析。更多信息: https://kubernetes.io/docs/concepts/containers/images
imageID	string	imageID 是容器镜像的镜像 ID。镜像 ID 可能与 PodSpec 中使用的镜像 ID 不匹配, 因为它可能已被运行时解决。
lastState	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定, 则默认是 ContainerStateWaiting。
name	string	name 是一个 DNS_LABEL, 代表容器的唯一名称。pod 中的每个容器必须在所有容器类型之间具有唯一的名称。无法更新。
ready	布尔值	ready 指定容器当前是否通过就绪度检查。当就绪度探测保持执行时, 该值将改变。如果没有指定就绪度探测, 则此字段在容器完全启动后默认为 true (请参阅 Started 字段)。 该值通常用于确定容器是否准备好接受流量。

属性	类型	描述
resources	对象	ResourceRequirements 描述计算资源要求。
restartCount	整数	restartCount 保存容器重启的次数。kubelet 努力始终递增值，但在某些情况下，状态可能会因为节点重启而丢失，然后该值可能会重置为 0。该值永远不会为负数。
started	布尔值	started 指示容器是否已完成其 postStart 生命周期 hook 并通过其启动探测。初始化为 false，在 startupProbe 被视为成功后变为 true。当容器重启时，或者 kubelet 暂时丢失状态时，重置为 false。在这两种情况下，启动探测将再次运行。当未定义 startupProbe 且容器正在运行并且传递了 postStart 生命周期 hook 时，始终为 true。null 值必须与 false 相同。
state	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

14.12.1.330. .status.ephemeralContainerStatuses[].lastState

描述

ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

类型

object

属性	类型	描述
running	object	ContainerStateRunning 是容器的运行状态。
已终止	object	ContainerStateTerminated 是容器的终止状态。
等待	object	ContainerStateWaiting 是容器的等待状态。

14.12.1.331. .status.ephemeralContainerStatuses[].lastState.running

描述

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.332. .status.ephemeralContainerStatuses[].lastState.terminated**描述**

ContainerStateTerminated 是容器的终止状态。

类型

object

必填

- **exitCode**

属性	类型	描述
containerID	string	容器 ID, 格式为 ' <code><type>://<container_id></code> '
exitCode	整数	容器最后一次终止的退出状态
finishedAt	时间	容器最后一次终止的时间
message	string	有关容器最后一次终止的消息
reason	string	(brief)容器最后一次终止的原因
signal	整数	容器最后一次终止的信号
startedAt	时间	容器以前执行的时间启动

14.12.1.333. .status.ephemeralContainerStatuses[].lastState.waiting**描述**

ContainerStateWaiting 是容器的等待状态。

类型

object

属性	类型	描述
message	string	有关容器尚未运行的原因的消息。
reason	string	(brief)容器没有运行的原因。

14.12.1.334. .status.ephemeralContainerStatuses[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象 (数量)	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象 (数量)	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.335. .status.ephemeralContainerStatuses[].resources.claims

描述

声明列出了此容器使用的 `spec.resourceClaims` 中定义的资源名称。这是一个 `alpha` 字段，需要启用 `DynamicResourceAllocation` 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.336. `.status.ephemeralContainerStatuses[].resources.claims[]`

描述

`ResourceClaim` 引用 `PodSpec.ResourceClaims` 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	<code>name</code> 必须与使用此字段的 Pod 的 <code>pod.spec.resourceClaims</code> 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.337. `.status.ephemeralContainerStatuses[].state`

描述

`ContainerState` 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 `ContainerStateWaiting`。

类型

object

属性	类型	描述
running	object	<code>ContainerStateRunning</code> 是容器的运行状态。
已终止	object	<code>ContainerStateTerminated</code> 是容器的终止状态。
等待	object	<code>ContainerStateWaiting</code> 是容器的等待状态。

14.12.1.338. `.status.ephemeralContainerStatuses[].state.running`

描述

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.339. .status.ephemeralContainerStatuses[].state.terminated

描述

ContainerStateTerminated 是容器的终止状态。

类型

object

必填

- **exitCode**

属性	类型	描述
containerID	string	容器 ID, 格式为 ' <code><type>://<container_id></code> '
exitCode	整数	容器最后一次终止的退出状态
finishedAt	时间	容器最后一次终止的时间
message	string	有关容器最后一次终止的消息
reason	string	(brief)容器最后一次终止的原因
signal	整数	容器最后一次终止的信号
startedAt	时间	容器以前执行的时间启动

14.12.1.340. .status.ephemeralContainerStatuses[].state.waiting

描述

ContainerStateWaiting 是容器的等待状态。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
message	string	有关容器尚未运行的原因的消息。
reason	string	(brief)容器没有运行的原因。

14.12.1.341. .status.initContainerStatuses

描述

列表中每个 init 容器有一个条目。最近成功的 init 容器将准备好 = true，最近启动的容器将设置 startTime。更多信息：<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status>

类型

数组

14.12.1.342. .status.initContainerStatuses[]

描述

ContainerStatus 包含此容器当前状态的详细信息。

类型

object

必填

- name
- ready
- restartCount
- image
- imageID

属性	类型	描述
allocatedResources	对象 (数量)	AllocatedResources 代表节点为此容器分配的計算资源。kubelet 在成功 pod 准入时将此值设置为 Container.Resources.Requests，并在成功接受所需的 pod 大小后将其设置为 Container.Resources.Requests。
containerID	string	containerID 是容器的 ID，格式为 '<type>://<container_id>'。其中 type 是容器运行时标识符，从 CRI API 的 Version 调用返回（例如 "containerd"）。

属性	类型	描述
image	string	image 是容器运行的容器镜像的名称。容器镜像可能与 PodSpec 中使用的镜像不匹配，因为它可能已被运行时解析。更多信息： https://kubernetes.io/docs/concepts/containers/images
imageID	string	imageID 是容器镜像的镜像 ID。镜像 ID 可能与 PodSpec 中使用的镜像 ID 不匹配，因为它可能已被运行时解决。
lastState	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。
name	string	name 是一个 DNS_LABEL，代表容器的唯一名称。pod 中的每个容器必须在所有容器类型之间具有唯一的名称。无法更新。
ready	布尔值	ready 指定容器当前是否通过就绪度检查。当就绪度探测保持执行时，该值将改变。如果没有指定就绪度探测，则此字段在容器完全启动后默认为 true（请参阅 Started 字段）。 该值通常用于确定容器是否准备好接受流量。
resources	对象	ResourceRequirements 描述计算资源要求。
restartCount	整数	restartCount 保存容器重启的次数。kubelet 努力始终递增值，但在某些情况下，状态可能会因为节点重启而丢失，然后该值可能会重置为 0。该值永远不会为负数。

属性	类型	描述
started	布尔值	started 指示容器是否已完成其 postStart 生命周期 hook 并通过其启动探测。初始化为 false，在 startupProbe 被视为成功后变为 true。当容器重启时，或者 kubelet 暂时丢失状态时，重置为 false。在这两种情况下，启动探测将再次运行。当未定义 startupProbe 且容器正在运行并且传递了 postStart 生命周期 hook 时，始终为 true。null 值必须与 false 相同。
state	object	ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

14.12.1.343. .status.initContainerStatuses[].lastState

描述

ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

类型

object

属性	类型	描述
running	object	ContainerStateRunning 是容器的运行状态。
已终止	object	ContainerStateTerminated 是容器的终止状态。
等待	object	ContainerStateWaiting 是容器的等待状态。

14.12.1.344. .status.initContainerStatuses[].lastState.running

描述

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.345. .status.initContainerStatuses[].lastState.terminated

描述

ContainerStateTerminated 是容器的终止状态。

类型

object

必填

- **exitCode**

属性	类型	描述
containerID	string	容器 ID, 格式为 ' <code><type>://<container_id></code> '
exitCode	整数	容器最后一次终止的退出状态
finishedAt	时间	容器最后一次终止的时间
message	string	有关容器最后一次终止的消息
reason	string	(brief)容器最后一次终止的原因
signal	整数	容器最后一次终止的信号
startedAt	时间	容器以前执行的时间启动

14.12.1.346. .status.initContainerStatuses[].lastState.waiting

描述

ContainerStateWaiting 是容器的等待状态。

类型

object

属性	类型	描述
message	string	有关容器尚未运行的原因的消息。
reason	string	(brief)容器没有运行的原因。

14.12.1.347. `.status.initContainerStatuses[].resources`**描述**

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 <code>spec.resourceClaims</code> 中定义的资源名称。 这是一个 alpha 字段，需要启用 <code>DynamicResourceAllocation</code> 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 <code>PodSpec.ResourceClaims</code> 中的一个条目。
limits	对象 (数量)	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象 (数量)	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.12.1.348. `.status.initContainerStatuses[].resources.claims`**描述**

声明列出了此容器使用的 `spec.resourceClaims` 中定义的资源名称。
这是一个 alpha 字段，需要启用 `DynamicResourceAllocation` 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.12.1.349. `.status.initContainerStatuses[].resources.claims[]`

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.12.1.350. `.status.initContainerStatuses[].state`

描述

ContainerState 具有容器可能的状态。只能指定其其中一个成员。如果没有指定，则默认是 ContainerStateWaiting。

类型

object

属性	类型	描述
running	object	ContainerStateRunning 是容器的运行状态。
已终止	object	ContainerStateTerminated 是容器的终止状态。
等待	object	ContainerStateWaiting 是容器的等待状态。

14.12.1.351. `.status.initContainerStatuses[].state.running`

描述

ContainerStateRunning 是容器的运行状态。

类型

object

属性	类型	描述
startedAt	时间	容器最后一次（重新）启动的时间

14.12.1.352. `.status.initContainerStatuses[].state.terminated`

描述

`ContainerStateTerminated` 是容器的终止状态。

类型

object

必填

- `exitCode`

属性	类型	描述
<code>containerID</code>	string	容器 ID, 格式为 ' <code><type>://<container_id></code> '
<code>exitCode</code>	整数	容器最后一次终止的退出状态
<code>finishedAt</code>	时间	容器最后一次终止的时间
<code>message</code>	string	有关容器最后一次终止的消息
<code>reason</code>	string	(brief)容器最后一次终止的原因
<code>signal</code>	整数	容器最后一次终止的信号
<code>startedAt</code>	时间	容器以前执行的时间启动

14.12.1.353. `.status.initContainerStatuses[].state.waiting`

描述

`ContainerStateWaiting` 是容器的等待状态。

类型

object

属性	类型	描述
<code>message</code>	string	有关容器尚未运行的原因的消息。
<code>reason</code>	string	(brief)容器没有运行的原因。

14.12.1.354. `.status.podIPs`

描述

`podIP` 包含分配给 pod 的 IP 地址。如果指定了此字段, 则 `0th` 条目必须与 `podIP` 字段匹配。可以为每个 IPv4 和 IPv6 分配最多 1 个值。如果尚未分配 IP, 则此列表为空。

类型

数组

14.12.1.355. .status.podIPs[]

描述

(复数) PodIPs 字段中条目的 IP 地址信息。每个条目包括：

IP: An IP address allocated to the pod. Routable at least within the cluster.

类型

object

属性	类型	描述
ip	string	IP 是分配给 pod 的 IP 地址(IPv4 或 IPv6)

14.12.2. API 端点

可用的 API 端点如下：

- **/api/v1/pods**
 - **GET**: 列出或监视 Pod 类型的对象
- **/api/v1/watch/pods**
 - **GET**: 观察单个对 Pod 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/pods**
 - **DELETE** : 删除 Pod 集合
 - **GET**: 列出或监视 Pod 类型的对象
 - **POST** : 创建 Pod
- **/api/v1/watch/namespaces/{namespace}/pods**
 - **GET**: 观察单个对 Pod 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/pods/{name}**
 - **DELETE** : 删除 Pod
 - **GET**: 读取指定的 Pod
 - **PATCH** : 部分更新指定的 Pod
 - **PUT** : 替换指定的 Pod
- **/api/v1/namespaces/{namespace}/pods/{name}/log**
 - **GET** : 读取指定 Pod 的日志

- **/api/v1/watch/namespaces/{namespace}/pods/{name}**
 - **GET**: 观察对 kind Pod 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{namespace}/pods/{name}/status**
 - **GET**: 指定 Pod 的读取状态
 - **PATCH** : 部分更新指定 Pod 的状态
 - **PUT** : 替换指定 Pod 的状态
- **/api/v1/namespaces/{namespace}/pods/{name}/ephemeralcontainers**
 - **GET**: 指定 Pod 的 read ephemeralcontainers
 - **PATCH** : 部分更新指定 Pod 的临时容器
 - **PUT** : 替换指定 Pod 的 ephemeralcontainers

14.12.2.1. /api/v1/pods

表 14.289. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 Pod 类型的对象

表 14.290. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodList 模式
401 - 未授权	空

14.12.2.2. /api/v1/watch/pods

表 14.291. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Pod 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.292. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.12.2.3. /api/v1/namespaces/{namespace}/pods

表 14.293. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.294. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Pod 集合

表 14.295. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.296. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.297. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 Pod 类型的对象

表 14.298. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.299. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodList 模式
401 - 未授权	空

HTTP 方法

POST

描述

创建 Pod

表 14.300. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.301. 主体参数

参数	类型	描述
正文 (body)	Pod 模式	

表 14.302. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
202 - Accepted	Pod 模式
401 - 未授权	空

14.12.2.4. /api/v1/watch/namespaces/{namespace}/pods

表 14.303. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.304. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Pod 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.305. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.12.2.5. /api/v1/namespaces/{namespace}/pods/{name}

表 14.306. 全局路径参数

参数	类型	描述
name	string	Pod 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.307. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Pod

表 14.308. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.309. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.310. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
202 - Accepted	Pod 模式
401 - 未授权	空

HTTP 方法

GET

描述

读取指定的 Pod

表 14.311. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 Pod

表 14.312. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.313. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.314. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 Pod

表 14.315. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.316. 主体参数

参数	类型	描述
正文 (body)	Pod 模式	

表 14.317. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

14.12.2.6. /api/v1/namespaces/{namespace}/pods/{name}/log

表 14.318. 全局路径参数

参数	类型	描述
name	string	Pod 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.319. 全局查询参数

参数	类型	描述
container	string	用于流传输日志的容器。如果 pod 中有一个容器，则默认为 container。
follow	布尔值	遵循 pod 的日志流。默认为false。
insecureSkipTLSVerifyBackend	布尔值	insecureSkipTLSVerifyBackend 表示 apiserver 不应确认它所连接的后端的 serving 证书的有效性。这将使 apiserver 和后端间的 HTTPS 连接不安全。这意味着 apiserver 无法从实际 kubelet 验证接收的日志数据。如果 kubelet 配置为验证 apiserver 的 TLS 凭证，并不意味着与实际 kubelet 的连接容易受到中间攻击（例如，攻击者无法截获来自真实 kubelet 的实际日志数据）的攻击。
limitBytes	整数	如果设置，在终止日志输出前从服务器读取的字节数。这不会显示完整的日志最终行，并且可能会稍微多或稍微低于指定限制。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
previous	布尔值	返回之前终止的容器日志。默认为false。
sinceSeconds	整数	从当前显示日志前的相对时间（以秒为单位）。如果这个值前面是 pod 启动的时间，则仅返回从 pod 启动以来的日志。如果这个值在以后，则不会返回任何日志。只能指定 sinceSeconds 或 sinceTime 中的一个。
tailLines	整数	如果设置，则会显示日志末尾的行数。如果没有指定，则从创建容器或 sinceSeconds 或 sinceTime 显示日志
timestamps	布尔值	如果为 true，请在日志输出的每一行的开头添加 RFC3339 或 RFC3339Nano 时间戳。默认为false。

HTTP 方法

GET

描述

指定 Pod 的读取日志

表 14.320. HTTP 响应

HTTP 代码	响应正文
200 - OK	string

HTTP 代码	响应正文
401 - 未授权	空

14.12.2.7. /api/v1/watch/namespaces/{namespace}/pods/{name}

表 14.321. 全局路径参数

参数	类型	描述
name	string	Pod 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.322. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Pod 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.323. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.12.2.8. /api/v1/namespaces/{namespace}/pods/{name}/status

表 14.324. 全局路径参数

参数	类型	描述
name	string	Pod 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.325. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 Pod 的读取状态

表 14.326. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 Pod 的部分更新状态

表 14.327. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.328. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.329. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 Pod 的状态

表 14.330. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.331. 主体参数

参数	类型	描述
正文 (body)	Pod 模式	

表 14.332. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

14.12.2.9. /api/v1/namespaces/{namespace}/pods/{name}/ephemeralcontainers

表 14.333. 全局路径参数

参数	类型	描述
name	string	Pod 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.334. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 Pod 的 read ephemeralcontainers

表 14.335. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定 Pod 的 ephemeralcontainers

表 14.336. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长, 且仅包含可打印的字符, 如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch), 但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.337. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.338. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定 Pod 的 ephemeralcontainers

表 14.339. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.340. 主体参数

参数	类型	描述
正文 (body)	Pod 模式	

表 14.341. HTTP 响应

HTTP 代码	响应正文
200 - OK	Pod 模式
201 - Created	Pod 模式
401 - 未授权	空

14.13. PODTEMPLATE [V1]

描述

PodTemplate 描述用于创建预定义 pod 副本的模板。

类型

对象

14.13.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
模板	object	PodTemplateSpec 描述了从模板创建 pod 时应该具有的数据

14.13.1.1. .template

描述

PodTemplateSpec 描述了从模板创建 pod 时应该具有的数据

类型

object

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec 是 pod 的描述。

14.13.1.2. .template.spec

描述

PodSpec 是 pod 的描述。

类型

object

必填

- **containers**

属性	类型	描述
activeDeadlineSeconds	整数	在系统主动尝试标记失败并终止相关的容器前，pod 可以在相对于 StartTime 的节点上处于活跃状态。值必须是正整数。
关联性	object	affinity 是一组关联性调度规则。
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 指明是否应该自动挂载服务帐户令牌。
containers	array	属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。
containers[]	对象	要在 pod 中运行的单一应用程序容器。
dnsConfig	object	PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

属性	类型	描述
dnsPolicy	字符串	<p>为 pod 设置 DNS 策略。默认为 "ClusterFirst"。有效值为 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' 或 'None'。DNSConfig 中给定的 DNS 参数将与选择与 DNSPolicy 的策略合并。要将 DNS 选项与 hostNetwork 一起设置，您必须将 DNS 策略明确指定为 'ClusterFirstWithHostNet'。</p> <p>可能枚举值： - "ClusterFirst" 表示 pod 应该先使用集群 DNS，除非 hostNetwork 为 true，如果可用，则回退到默认的（由 kubelet 决定）DNS 设置。 - "ClusterFirstWithHostNet" 表示 pod 应该首先使用集群 DNS，如果可用，则回退到它。在默认的（由 kubelet 决定）DNS 设置中。 - "Default" 表示 pod 应该使用默认（由 kubelet 决定）DNS 设置。 - "None" 表示 pod 应该使用空的 DNS 设置。名称服务器和搜索路径等 DNS 参数应该通过 DNSConfig 定义。</p>
enableServiceLinks	布尔值	<p>EnableServiceLinks 指示是否应将有关服务的信息注入到 pod 的环境变量中，与 Docker 链接的语法匹配。可选：默认为 true。</p>
ephemeralContainers	array	<p>此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。</p>

属性	类型	描述
ephemeralContainers[]	object	<p>EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。</p> <p>要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。</p>
hostAliases	array	hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。
hostAliases[]	对象	HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。
hostIPC	布尔值	使用主机的 ipc 命名空间。可选：默认为 false。
hostNetwork	布尔值	为此 pod 请求的主机网络。使用主机的网络命名空间。如果设置了这个选项，则必须指定要使用的端口。默认为 false。
hostPID	布尔值	使用主机的 pid 命名空间。可选：默认为 false。
hostUsers	布尔值	<p>使用主机的用户命名空间。可选：默认为 true。如果设置为 true 或不存在，则 pod 将在主机用户命名空间中运行，当 pod 需要可供主机用户命名空间使用的功能时，如加载带有 CAP_SYS_MODULE 的内核模块。当设置为 false 时，会为 pod 创建一个新的 usersns。设置 false 有助于缓解容器中断漏洞，即使允许用户以 root 身份运行其容器，而无需实际在主机上具有 root 特权。此字段是 alpha-level，仅被启用 UserNamespacesSupport 功能的服务器所接受。</p>

属性	类型	描述
hostname	字符串	如果没有指定 Pod 的主机名，则 pod 的主机名将设置为系统定义的值。
imagePullSecrets	array	imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息： https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	对象	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
initContainers	array	属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
initContainers[]	对象	要在 pod 中运行的单一应用程序容器。
nodeName	字符串	nodeName 是一个将此 pod 调度到特定节点请求。如果是非空，调度程序只会将此 pod 调度到该节点上，假设它符合资源要求。

属性	类型	描述
<code>nodeSelector</code>	对象 (字符串)	<code>nodeSelector</code> 是一个选择器，必须为 <code>true</code> 才能使 pod 适合节点。必须与 pod 标签匹配的选择器，才能调度到该节点上。更多信息： https://kubernetes.io/docs/concepts/configuration/assign-pod-node/
<code>OS</code>	object	PodOS 定义 pod 的 OS 参数。
开销	对象 (数量)	开销代表了与为给定 <code>RuntimeClass</code> 运行 pod 相关的资源开销。此字段将由 <code>RuntimeClass</code> 准入控制器在准入时自动填充。如果启用了 <code>RuntimeClass</code> 准入控制器，则无法在 Pod 创建请求中设置开销。 <code>RuntimeClass</code> 准入控制器将拒绝 Pod 创建已设置开销的请求。如果在 <code>PodSpec</code> 中配置并选择了 <code>RuntimeClass</code> ，则 <code>Overhead</code> 将被设置为对应的 <code>RuntimeClass</code> 中定义的值，否则它将保留为未设置并被视为零。更多信息： https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md
<code>preemptionPolicy</code>	字符串	<code>preemptionPolicy</code> 是抢占优先级较低的 pod 的策略。Never, PreemptLowerPriority 之一。如果未设置，则默认为 PreemptLowerPriority。 可能枚举值： - "Never" 表示 pod 不会抢占优先级较低的其他 pod。 - "PreemptLowerPriority" 表示 pod 可以抢占优先级较低的其他 pod。
<code>priority</code>	整数	优先级值。各种系统组件使用此字段查找 pod 的优先级。启用 Priority Admission Controller 时，它会阻止用户设置此字段。准入控制器从 <code>PriorityClassName</code> 填充此字段。数值越大，优先级越高。

属性	类型	描述
priorityClassName	字符串	如果指定，则表示 pod 的优先级。 "system-node-critical"和"system-cluster-critical"是两个特殊关键字，表示前者最高的优先级。必须使用该名称创建 PriorityClass 对象来定义任何其他名称。如果没有指定，如果没有默认值，pod 优先级将为 default 或零。
readinessGates	array	如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息 时，pod 已就绪： https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	对象	PodReadinessGate 包含对 pod 条件的引用
resourceClaims	数组	ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。
resourceClaims[]	object	PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

属性	类型	描述
restartPolicy	字符串	<p>为 pod 中的所有容器重启策略。Always、OnFailure、Never 之一。在某些上下文中，仅允许这些值的子集。默认值为 Always。更多信息：</p> <p>https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy</p> <p>可能枚举值：- "Always" - "Never" - "OnFailure"</p>
runtimeClassName	字符串	<p>runtimeClassName 指的是 node.k8s.io 组中的 RuntimeClass 对象，它应该用于运行此 pod。如果没有 RuntimeClass 资源与命名类匹配，则不会运行 pod。如果未设置或为空，则使用 "legacy" RuntimeClass，它是一个带有使用默认运行时处理器的空定义的隐式类。更多信息：</p> <p>https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class</p>
schedulerName	字符串	<p>如果指定，pod 将由指定的调度程序分配。如果没有指定，pod 将默认调度程序分配。</p>
schedulingGates	数组	<p>SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。</p> <p>SchedulingGates 只能在创建 pod 时设置，之后只能删除。</p> <p>这是 PodSchedulingReadiness 功能门启用的 beta 功能。</p>
schedulingGates[]	object	<p>PodSchedulingGate 与 Pod 关联，以保护其调度。</p>

属性	类型	描述
securityContext	object	PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。
serviceAccount	字符串	deprecatedServiceAccount 是 ServiceAccountName 的前文别名。deprecated : 改为使用 serviceAccountName。
serviceAccountName	字符串	serviceAccountName 是用于运行此 pod 的 ServiceAccount 的名称。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
setHostnameAsFQDN	布尔值	如果为 true, 则 pod 的主机名将配置为 pod 的 FQDN, 而不是叶名称 (默认值)。在 Linux 容器中, 这意味着在内核的 hostname 字段中设置 FQDN (struct utsname 的 nodename 字段)。在 Windows 容器中, 这意味着将 registry 键 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 的 registry 值设置为 FQDN。如果 pod 没有 FQDN, 则无效。默认为 false。
shareProcessNamespace	布尔值	在 pod 中所有容器间共享单个进程命名空间。设置此容器后, 将能够查看和信号来自同一 pod 中其他容器的进程, 并且每个容器中的第一个进程不会被分配 PID 1。无法同时设置 hostPID 和 ShareProcessNamespace。可选: 默认为 false。
subdomain	字符串	如果指定, 完全限定的 Pod 主机名将是 "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>"。如果没有指定, pod 根本不会具有 domainname。

属性	类型	描述
terminationGracePeriodSeconds	整数	pod 需要安全终止的可选持续时间（以秒为单位）。在删除请求中可能会减少。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。如果这个值为 nil，则使用默认的宽限期。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。默认值为 30 秒。
容限 (tolerations)	array	如果指定，pod 的容限。
tolerations[]	对象	附加到此 Tolerant 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。
topologySpreadConstraints	array	topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。
topologySpreadConstraints[]	对象	TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。
卷	array	可由属于 pod 的容器挂载的卷列表。更多信息： https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	对象	卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

14.13.1.3. .template.spec.affinity

描述

affinity 是一组关联性调度规则。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
nodeAffinity	object	节点关联性是一组节点关联性调度规则。
podAffinity	object	Pod 关联性是一组 pod 关联性调度规则。
podAntiAffinity	object	Pod 反关联性是一组 pod 反关联性调度规则。

14.13.1.4. .template.spec.affinity.nodeAffinity

描述

节点关联性是一组节点关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。
requiredDuringSchedulingIgnoredDuringExecution	object	节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

14.13.1.5. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。

类型

数组

14.13.1.6. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecu

描述

空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。

类型

object

必填

- weight
- preference

属性	类型	描述
preference	object	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。
weight	整数	与对应的 nodeSelectorTerm 匹配的权重，范围为 1-100。

14.13.1.7. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecu

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

属性	类型	描述
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.13.1.8. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecu

描述

节点标签(label)的节点选择器要求列表。

类型

数组

14.13.1.9. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecu

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.13.1.10. `.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExec`**描述**

节点字段的节点选择器要求列表。

类型

数组

14.13.1.11. `.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecu`**描述**

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- `key`
- `operator`

属性	类型	描述
<code>key</code>	字符串	选择器应用到的标签键。
<code>operator</code>	字符串	代表键与一组值的关系。有效的运算符为 <code>In</code> 、 <code>NotIn</code> 、 <code>Exists</code> 、 <code>DoesNotExist</code> 、 <code>gt</code> 和 <code>Lt</code> 。 possible enum 值：- "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
<code>values</code>	数组（字符串）	字符串值数组。如果运算符是 <code>In</code> 或 <code>NotIn</code> ，则值数组必须是非空的。如果运算符是 <code>Exists</code> 或 <code>DoesNotExist</code> ，则值数组必须为空。如果 Operator 是 <code>Gt</code> 或 <code>Lt</code> ，则值数组必须具有单个元素，该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.13.1.12. `.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu`**描述**

节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

类型

object

必填

- `nodeSelectorTerms`

nodeSelectorTerms

属性	类型	描述
nodeSelectorTerms	array	必需。节点选择器术语列表。条款是 ORed。
nodeSelectorTerms[]	对象	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

14.13.1.13. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

必需。节点选择器术语列表。条款是 ORed。

类型

数组

14.13.1.14. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.13.1.15. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

节点标签(label)的节点选择器要求列表。

类型

数组

14.13.1.16. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- key
- operator

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.13.1.17. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

节点字段的节点选择器要求列表。

类型

数组

14.13.1.18. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecu

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object**必填**

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.13.1.19. .template.spec.affinity.podAffinity**描述**

Pod 关联性是一组 pod 关联性调度规则。

类型**object**

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点, 但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点, 即对于满足所有调度要求 (资源请求、requiredDuringScheduling 关联性表达式等) 的每个节点, 它通过迭代此字段元素并将 "weight" 添加到摘要中, 如果节点具有与相应 podAffinityTerm 匹配的 pod; 具有最高总和最高的节点是首选的节点。

属性	类型	描述
<code>preferredDuringSchedulingIgnoredDuringExecution[]</code>	对象	所有匹配的 <code>WeightedPodAffinityTerm</code> 字段的权重都会被添加，以查找首选节点。
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 <code>podAffinityTerm</code> 对应的节点列表会被干扰，例如，所有术语都必须满足。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（与给定命名空间匹配的 <code>labelSelector</code> 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 <code>co-located</code> 定义在带有键 <code><topologyKey></code> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.13.1.20. `.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、`requiredDuringSchedulingIgnoredDuringExecution` 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中，如果节点具有与相应 `podAffinityTerm` 匹配的 pod；具有最高总和最高的节点是首选的节点。

类型

数组

14.13.1.21. `.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

所有匹配的 `WeightedPodAffinityTerm` 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- `weight`
- `podAffinityTerm`

属性	类型	描述
podAffinityTerm	object	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

14.13.1.22. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecu

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- **topologyKey**

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。

属性	类型	描述
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.13.1.23. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecut

描述

如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

数组

14.13.1.24. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecut

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。

属性	类型	描述
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.13.1.25. .template.spec.affinity.podAntiAffinity

描述

Pod 反关联性是一组 pod 反关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 podAffinityTerm 匹配的 pod；如果节点具有与相应 podAffinityTerm 匹配的 pod，则该节点是首选的。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

属性	类型	描述
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.13.1.26. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringE

描述

调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将“权重”添加到节点具有与相应 podAffinityTerm 匹配的 pod；如果节点具有与相应 podAffinityTerm 匹配的 pod，则该节点是首选的。

类型

数组

14.13.1.27. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringE

描述

所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- weight
- podAffinityTerm

属性	类型	描述
----	----	----

属性	类型	描述
podAffinityTerm	object	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

14.13.1.28. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringE

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。

属性	类型	描述
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.13.1.29. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

数组

14.13.1.30. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。

属性	类型	描述
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.13.1.31. .template.spec.containers

描述

属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。

类型

数组

14.13.1.32. .template.spec.containers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- name

属性	类型	描述
----	----	----

属性	类型	描述
args	数组 (字符串)	入口点的参数。如果没有提供, 则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组 (字符串)	ENTRYPOINT 数组, 不会在 shell 中执行。如果没有提供, 则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 <code>C_IDENTIFIER</code> 。当容器启动时, 所有无效的密钥都会被报告为事件。当多个源中存在键时, 与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

属性	类型	描述
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。

属性	类型	描述
ports	array	要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。

属性	类型	描述
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。

属性	类型	描述
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统上的 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.13.1.33. .template.spec.containers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.13.1.34. .template.spec.containers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。

属性	类型	描述
valueFrom	object	EnvVarSource 代表 EnvVarar 的值的一个源。

14.13.1.35. .template.spec.containers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVarar 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.13.1.36. .template.spec.containers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

属性	类型	描述
----	----	----

14.13.1.37. .template.spec.containers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.13.1.38. .template.spec.containers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.13.1.39. .template.spec.containers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.13.1.40. .template.spec.containers[].envFrom**描述**

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.13.1.41. .template.spec.containers[].envFrom[]**描述**

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

属性	类型	描述
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.13.1.42. .template.spec.containers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.13.1.43. .template.spec.containers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

属性	类型	描述
optional	布尔值	指定是否需要定义 Secret

14.13.1.44. .template.spec.containers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

14.13.1.45. .template.spec.containers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
HttpGet	object	HttpGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPsocketAction 描述了基于打开套接字的操作

14.13.1.46. .template.spec.containers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组 (字符串)	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.47. .template.spec.containers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

属性	类型	描述
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.48. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.49. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.50. .template.spec.containers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.51. .template.spec.containers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.13.1.52. .template.spec.containers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.53. .template.spec.containers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.54. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders**描述**

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.55. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.56. .template.spec.containers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.57. .template.spec.containers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。

属性	类型	描述
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.58. `.template.spec.containers[].livenessProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组 (字符串)	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.59. .template.spec.containers[].livenessProbe.grpc**描述**

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称 (请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md) 。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.60. .template.spec.containers[].livenessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.61. .template.spec.containers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.62. .template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.63. .template.spec.containers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.64. .template.spec.containers[].ports

描述

要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

数组

14.13.1.65. .template.spec.containers[].ports[]

描述

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.13.1.66. .template.spec.containers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.67. .template.spec.containers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令 ('' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.68. .template.spec.containers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.69. .template.spec.containers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.70. .template.spec.containers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.71. .template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.72. .template.spec.containers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.73. .template.spec.containers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.13.1.74. .template.spec.containers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.13.1.75. .template.spec.containers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.13.1.76. .template.spec.containers[].resources.claims

描述

声明列出了此容器使用的 `spec.resourceClaims` 中定义的资源名称。
这是一个 alpha 字段，需要启用 `DynamicResourceAllocation` 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.13.1.77. `.template.spec.containers[].resources.claims[]`

描述

`ResourceClaim` 引用 `PodSpec.ResourceClaims` 中的一个条目。

类型

object

必填

- `name`

属性	类型	描述
<code>name</code>	string	<code>name</code> 必须与使用此字段的 Pod 的 <code>pod.spec.resourceClaims</code> 中一个条目的名称匹配。它使得资源在容器内可用。

14.13.1.78. `.template.spec.containers[].securityContext`

描述

`securityContext` 包含将应用到容器的安全配置。`SecurityContext` 和 `PodSecurityContext` 中都存在一些字段。当同时设置这两个时，`SecurityContext` 中的值就会优先使用。

类型

object

属性	类型	描述
<code>allowPrivilegeEscalation</code>	布尔值	<code>allowPrivilegeEscalation</code> 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 <code>no_new_privs</code> 标志。当容器以 <code>Privileged 2</code> 运行行为 <code>Privileged 2</code> 时， <code>allowPrivilegeEscalation</code> 为 <code>true</code> 时，当 <code>spec.os.name</code> 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。

属性	类型	描述
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.13.1.79. .template.spec.containers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.13.1.80. .template.spec.containers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.13.1.81. .template.spec.containers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。

属性	类型	描述
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值：- "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp。- "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。- "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.13.1.82. .template.spec.containers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	<p>GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。</p>
gmsaCredentialSpecName	字符串	<p>GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。</p>

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level, 仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外, 如果 HostProcess 为 true, 则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName, 以运行容器进程的入口点。如果未指定, 则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置, 则优先使用 SecurityContext 指定的值。

14.13.1.83. .template.spec.containers[].startupProbe

描述

probe 描述了对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后, 探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.84. .template.spec.containers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.85. `..template.spec.containers[].startupProbe.grpc`**描述**

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.86. `..template.spec.containers[].startupProbe.httpGet`**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.87. .template.spec.containers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.88. .template.spec.containers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.89. .template.spec.containers[].startupProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.90. .template.spec.containers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.13.1.91. .template.spec.containers[].volumeDevices[]

描述

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.13.1.92. .template.spec.containers[].volumeMounts

描述

要挂载到容器文件系统 Pod 卷。无法更新。

类型

数组

14.13.1.93. .template.spec.containers[].volumeMounts[]

描述

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- **name**
- **mountPath**

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 '..'。

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.13.1.94. .template.spec.dnsConfig

描述

PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

类型

object

属性	类型	描述
nameservers	数组（字符串）	DNS 名称服务器 IP 地址列表。这将附加到从 DNSPolicy 生成的基本名称服务器。将删除重复的名称服务器。
options	array	DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。
options[]	对象	PodDNSConfigOption 定义 pod 的 DNS 解析器选项。
搜索	数组（字符串）	host-name 查找的 DNS 搜索域列表。这将附加到 DNSPolicy 生成的基本搜索路径中。将删除重复的搜索路径。

14.13.1.95. .template.spec.dnsConfig.options

描述

DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。

类型

数组

14.13.1.96. .template.spec.dnsConfig.options[]

描述

PodDNSConfigOption 定义 pod 的 DNS 解析器选项。

类型

object

属性	类型	描述
name	字符串	必需。
value	string	

14.13.1.97. .template.spec.ephemeralContainers

描述

此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。

类型**数组****14.13.1.98. .template.spec.ephemeralContainers[]****描述**

EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。

要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。

类型**object****必填**

- **name**

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
命令	数组（字符串）	ENTRYPOINT 数组.不会在 shell 中执行。如果没有提供，则使用镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME)使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME)语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images

属性	类型	描述
imagePullPolicy	字符串	<p>镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签, 则默认为 Always, 否则为 IfNotPresent。无法更新。更多信息 :</p> <p>https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>可能枚举值 : - "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败, 则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在, 且拉取失败, 则容器将失败。- "Never" 表示 kubelet 不会拉取镜像, 但只使用本地镜像。如果镜像不存在, 容器将失败</p>
生命周期	object	<p>生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序, 在操作完成前管理容器块, 除非容器进程失败, 在这种情况下, 处理程序会中止。</p>
livenessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
name	字符串	<p>指定为 DNS_LABEL 的临时容器的名称。此名称在所有容器、init 容器和临时容器中必须是唯一的。</p>
ports	array	<p>临时容器不允许使用端口。</p>
ports[]	对象	<p>containerPort 代表单一容器中的网络端口。</p>
readinessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
resizePolicy	数组	<p>容器的资源重新定义策略大小。</p>
resizePolicy[]	object	<p>ContainerResizePolicy 代表容器的资源大小策略。</p>

属性	类型	描述
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
targetContainerName	字符串	<p>如果设置，则此临时容器目标的 PodSpec 中的容器名称。临时容器将在此容器的命名空间中运行 (IPC、PID 等)。如果没有设置，则临时容器将使用 Pod spec 中配置的命名空间。</p> <p>容器运行时必须实施对此功能的支持。如果运行时不支持以命名空间为目标，则设置此字段的结果未定义。</p>

属性	类型	描述
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。

属性	类型	描述
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.13.1.99. .template.spec.ephemeralContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.13.1.100. .template.spec.ephemeralContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的一个源。

14.13.1.101. .template.spec.ephemeralContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.13.1.102. .template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- key

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.13.1.103. .template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- fieldPath

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.13.1.104. .template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.13.1.105. .template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.13.1.106. .template.spec.ephemeralContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.13.1.107. .template.spec.ephemeralContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.13.1.108. .template.spec.ephemeralContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.13.1.109. .template.spec.ephemeralContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.13.1.110. .template.spec.ephemeralContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

属性	类型	描述
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

14.13.1.111. .template.spec.ephemeralContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPsocketAction 描述了基于打开套接字的操作

14.13.1.112. .template.spec.ephemeralContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录（'/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（' ' 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.113. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.114. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.115. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name

- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.116. .template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.117. .template.spec.ephemeralContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.13.1.118. .template.spec.ephemeralContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.119. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头

属性	类型	描述
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.120. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.121. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.122. .template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.123. .template.spec.ephemeralContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.124. .template.spec.ephemeralContainers[].livenessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.125. .template.spec.ephemeralContainers[].livenessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.126. .template.spec.ephemeralContainers[].livenessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.127. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.128. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.129. .template.spec.ephemeralContainers[].livenessProbe.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.130. .template.spec.ephemeralContainers[].ports**描述**

临时容器不允许使用端口。

类型

数组

14.13.1.131. .template.spec.ephemeralContainers[].ports[]**描述**

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.13.1.132. .template.spec.ephemeralContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.133. .template.spec.ephemeralContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.134. .template.spec.ephemeralContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.135. .template.spec.ephemeralContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.136. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.137. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.138. .template.spec.ephemeralContainers[].readinessProbe.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.139. .template.spec.ephemeralContainers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.13.1.140. .template.spec.ephemeralContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- resourceName
- restartPolicy

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.13.1.141. .template.spec.ephemeralContainers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.13.1.142. .template.spec.ephemeralContainers[].resources.claims**描述**

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.13.1.143. .template.spec.ephemeralContainers[].resources.claims[]**描述**

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.13.1.144. .template.spec.ephemeralContainers[].securityContext**描述**

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型**object**

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行行为 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.13.1.145. .template.spec.ephemeralContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.13.1.146. .template.spec.ephemeralContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.13.1.147. .template.spec.ephemeralContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.13.1.148. .template.spec.ephemeralContainers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.13.1.149. .template.spec.ephemeralContainers[].startupProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.150. .template.spec.ephemeralContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.151. .template.spec.ephemeralContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.13.1.152. .template.spec.ephemeralContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.13.1.153. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.154. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.155. .template.spec.ephemeralContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.156. .template.spec.ephemeralContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.13.1.157. .template.spec.ephemeralContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- name
- devicePath

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.13.1.158. .template.spec.ephemeralContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。

类型

数组

14.13.1.159. .template.spec.ephemeralContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- name
- mountPath

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 `.`。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.13.1.160. .template.spec.hostAliases

描述

hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。

类型

数组

14.13.1.161. .template.spec.hostAliases[]

描述

HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。

类型

object

属性	类型	描述
主机名	数组（字符串）	以上 IP 地址的主机名。
ip	字符串	主机文件条目的 IP 地址。

14.13.1.162. .template.spec.imagePullSecrets

描述

imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息：

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

类型

数组

14.13.1.163. .template.spec.imagePullSecrets[]

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.164. .template.spec.initContainers

描述

属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息：

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

类型

数组

14.13.1.165. .template.spec.initContainers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- name

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组（字符串）	ENTRYPOINT 数组。不会在 shell 中执行。如果没有提供，则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败

属性	类型	描述
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。
ports	array	要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

属性	类型	描述
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。

属性	类型	描述
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.13.1.166. .template.spec.initContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.13.1.167. .template.spec.initContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。

属性	类型	描述
value	字符串	变量引用 <code>\$(VAR_NAME)</code> 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVarar 的值的一个源。

14.13.1.168. .template.spec.initContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVarar 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.13.1.169. .template.spec.initContainers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.13.1.170. .template.spec.initContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.13.1.171. .template.spec.initContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的

属性	类型	描述
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.13.1.172. .template.spec.initContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- key

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.13.1.173. .template.spec.initContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.13.1.174. .template.spec.initContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.13.1.175. .template.spec.initContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.13.1.176. .template.spec.initContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.13.1.177. .template.spec.initContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.13.1.178. .template.spec.initContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
HttpGet	object	HttpGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.13.1.179. .template.spec.initContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.180. .template.spec.initContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.181. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.182. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.183. .template.spec.initContainers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.184. .template.spec.initContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCP Socket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCP SocketAction 描述了基于打开套接字的操作

14.13.1.185. .template.spec.initContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.186. .template.spec.initContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.187. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.188. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.189. .template.spec.initContainers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.190. .template.spec.initContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

属性	类型	描述
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.191. `.template.spec.initContainers[].livenessProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令 ('' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.13.1.192. `.template.spec.initContainers[].livenessProbe.grpc`

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.193. .template.spec.initContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.194. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.195. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.196. .template.spec.initContainers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object**必填**

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.197. .template.spec.initContainers[].ports**描述**

要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

数组

14.13.1.198. .template.spec.initContainers[].ports[]**描述**

containerPort 代表单一容器中的网络端口。

类型**object****必填**

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.13.1.199. .template.spec.initContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.200. .template.spec.initContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.201. .template.spec.initContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.13.1.202. .template.spec.initContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.13.1.203. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.204. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.205. .template.spec.initContainers[].readinessProbe.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.206. .template.spec.initContainers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.13.1.207. .template.spec.initContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- resourceName
- restartPolicy

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.13.1.208. .template.spec.initContainers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.13.1.209. .template.spec.initContainers[].resources.claims**描述**

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.13.1.210. .template.spec.initContainers[].resources.claims[]**描述**

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.13.1.211. .template.spec.initContainers[].securityContext

描述

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型

object

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行行为 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.13.1.212. .template.spec.initContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.13.1.213. .template.spec.initContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.13.1.214. .template.spec.initContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.13.1.215. .template.spec.initContainers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机 Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.13.1.216. .template.spec.initContainers[].startupProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.13.1.217. .template.spec.initContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.13.1.218. .template.spec.initContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.13.1.219. .template.spec.initContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.13.1.220. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.13.1.221. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.13.1.222. .template.spec.initContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.13.1.223. .template.spec.initContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.13.1.224. .template.spec.initContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.13.1.225. .template.spec.initContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。无法更新。

类型

数组

14.13.1.226. .template.spec.initContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- **name**
- **mountPath**

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 `.`。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值：- "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。- "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.13.1.227. .template.spec.os

描述

PodOS 定义 pod 的 OS 参数。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是操作系统的名称。目前支持的值有 linux 和 windows。额外值可能会在以后定义，可以是： https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration 客户端应该应该处理附加值，并将此字段中的未识别的值视为 os: null

14.13.1.228. .template.spec.readinessGates**描述**

如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪：<https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

类型

数组

14.13.1.229. .template.spec.readinessGates[]**描述**

PodReadinessGate 包含对 pod 条件的引用

类型

object

必填

- **conditionType**

属性	类型	描述
conditionType	字符串	conditionType 指的是 pod 条件列表中具有匹配类型的条件。

14.13.1.230. .template.spec.resourceClaims**描述**

ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。

这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。

类型

数组

14.13.1.231. .template.spec.resourceClaims[]

描述

PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

类型

object

必填

- name

属性	类型	描述
name	string	在 pod 中唯一标识此资源声明的名称。这必须是 DNS_LABEL。
source	object	ClaimSource 描述了对 ResourceClaim 的引用。 应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

14.13.1.232. .template.spec.resourceClaims[].source

描述

ClaimSource 描述了对 ResourceClaim 的引用。

应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

类型

object

属性	类型	描述
resourceClaimName	string	ResourceClaimName 是与这个 pod 相同的命名空间中的 ResourceClaim 对象的名称。

属性	类型	描述
resourceClaimTemplateName	string	<p>ResourceClaimTemplateName 是与这个 pod 相同的命名空间中的 ResourceClaimTemplate 对象的名称。</p> <p>模板将用于创建新的 ResourceClaim，它将绑定到此 pod。删除此 pod 后，ResourceClaim 也会被删除。ResourceClaim 的名称将是 <pod name>-<resource name>，其中 <resource name> 是 PodResourceClaim.Name。如果串联名称对 ResourceClaim 无效（如太长，则 Pod 验证将拒绝 pod）。</p> <p>没有为 pod 拥有该名称的现有 ResourceClaim 不会被用于 pod，以避免错误地使用不相关的资源。然后，调度和 pod 启动会被阻断，直到删除不相关的 ResourceClaim。</p> <p>此字段不可变，在创建 ResourceClaim 后，control plane 不会对对应的 ResourceClaim 进行更改。</p>

14.13.1.233. .template.spec.schedulingGates

描述

SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。SchedulingGates 只能在创建 pod 时设置，之后只能删除。

这是 PodSchedulingReadiness 功能门启用的 beta 功能。

类型

数组

14.13.1.234. .template.spec.schedulingGates[]

描述

PodSchedulingGate 与 Pod 关联，以保护其调度。

类型

object

必填

- name

属性	类型	描述
name	string	调度授权的名称。每个调度最低要求都必须具有唯一的 name 字段。

14.13.1.235. .template.spec.securityContext

描述

PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。

类型

object

属性	类型	描述
fsGroup	整数	<p>适用于 pod 中所有容器的特殊补充组。有些卷类型允许 Kubelet 将该卷的所有权更改为 pod 拥有：</p> <ol style="list-style-type: none"> 1.所属 GID 将是 FSGroup 2. set setgid 位（在卷中创建的新文件将归 FSGroup 所有） 3. 权限位为 OR'd with rw-rw---- <p>如果未设置，Kubelet 将不修改任何卷的所有权和权限。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
fsGroupChangePolicy	字符串	<p>fsGroupChangePolicy 定义在 Pod 中公开卷之前更改卷的所有权和权限的行为。此字段仅适用于支持 fsGroup 基于所有权（和权限）的卷类型。它对临时卷类型没有影响，如：secret、configmaps 和 emptydir。有效值为 "OnRootMismatch" 和 "Always"。如果没有指定，则使用 "Always"。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值： - "Always" 表示当卷挂载到 Pod 中时，应始终更改卷的所有权和权限。这个默认行为。 - "OnRootMismatch" 表示只有在根目录的权限与卷上预期权限不匹配时才会更改卷的所有权和权限。这有助于缩短更改卷的所有权和权限所需的时间。</p>

属性	类型	描述
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

属性	类型	描述
supplementalGroups	数组（整数）	除了容器的主 GID、fsGroup（如果指定）以及容器镜像中定义的组成员资格（如果指定）之外，应用于每个容器中的第一个进程运行的组列表。如果未指定，则不会将额外的组添加到任何容器中。请注意，容器进程的 uid 中定义的组成员资格仍然有效，即使它们没有包括在此列表中。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls	array	sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls[]	对象	sysctl 定义要设置的内核参数
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.13.1.236. .template.spec.securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.13.1.237. .template.spec.securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.13.1.238. .template.spec.securityContext.sysctls

描述

sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。

类型

数组

14.13.1.239. .template.spec.securityContext.sysctls[]

描述

sysctl 定义要设置的内核参数

类型

object

必填

- name
- value

属性	类型	描述
name	字符串	要设置的属性的名称
value	字符串	要设置的属性值

14.13.1.240. .template.spec.securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。

属性	类型	描述
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.13.1.241. .template.spec.tolerations

描述

如果指定，pod 的容限。

类型

数组

14.13.1.242. .template.spec.tolerations[]

描述

附加到此 Tolerations 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
effect	字符串	<p>effect 表示污点效果要匹配。空意味着匹配所有污点效果。指定后，允许的值为 NoSchedule、PreferNoSchedule 和 NoExecute。</p> <p>可能枚举值： - "NoExecute" Evict 任何已在运行的 pod，这些 pod 不容许该污点。目前由 NodeController 实施。 - "NoSchedule" 不允许新 pod 调度到该节点上，除非它们容许污点，但允许提交给 Kubelet 的所有 pod 而无需启动调度程序，并允许所有已在运行的 pod 继续运行。调度程序强制使用 - "PreferNoSchedule" Like TaintEffectNoSchedule，但调度程序会尝试不将新 pod 调度到节点上，而不是禁止新 pod 完全调度到该节点上。由调度程序实施。</p>
key	字符串	<p>key 是容限应用到的污点键。empty 表示与所有污点键匹配。如果键为空，则运算符必须是 Exists；此组合意味着匹配所有值和所有键。</p>
operator	字符串	<p>Operator 代表键与值的关系。有效的运算符是 Exists 和 Equal。默认为 Equal。exists 等同于 value 的通配符，以便 pod 可以容忍特定类别的所有污点。</p> <p>可能枚举值： - "Equal" - "Exists"</p>
tolerationSeconds	整数	<p>TolerationSeconds 代表容限的期间（该容限必须是 NoExecute，否则将忽略此字段）容许污点。默认情况下，它没有被设置，这意味着容许永久污点（不驱除）。系统会将零值和负值视为 0（立即撤离）。</p>
value	字符串	<p>值是容限匹配的污点值。如果运算符是 Exists，则该值应为空，否则只是一个常规字符串。</p>

14.13.1.243. .template.spec.topologySpreadConstraints

描述

topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。

类型

数组

14.13.1.244. .template.spec.topologySpreadConstraints[]

描述

TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。

类型

object

必填

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

属性	类型	描述
labelSelector	labelSelector	labelSelector 用于查找匹配的 pod。与此标签选择器匹配的 Pod 被计算，以确定其对应拓扑域中的 pod 数量。
matchLabelKeys	数组（字符串）	matchLabelKeys 是一组 pod 标签键，用于选择要计算分布的 pod。键用于从传入的 pod 标签中查找值，这些键值标签带有 labelSelector 的 ANDed 来选择要针对传入 pod 计算的现有 pod 组。禁止在 MatchLabelKeys 和 LabelSelector 中存在相同的键。当 LabelSelector 没有设置时，无法设置 matchLabelKeys。传入的 pod 标签中不存在的键将被忽略。null 或 empty 列表表示仅匹配 labelSelector。 这是一个 beta 字段，需要启用 MatchLabelKeysInPodTopologySpread 功能门（默认启用）。

属性	类型	描述
maxSkew	整数	<p>maxSkew 描述了 pod 可能不均匀分布的程度。当 whenUnsatisfiable=DoNotSchedule 时，目标拓扑中匹配 pod 数量和全局最小值之间允许的最大区别。如果有资格的域数量小于 MinDomains，则全局最小值是符合域或零的匹配 pod 的最小数量。例如，在 3 区集群中，MaxSkew 设置为 1，具有相同 labelSelector 的 pod 分布为 2/2/1：在这种情况下，全局最小值为 1。 zone1 zone2 zone3 P P P P P P P P P - 如果 MaxSkew 为 1，则传入的 pod 只能调度到 zone3，以变为 2/2/2；将其调度到 zone1 (zone2) 上的 ActualSkew (3-1) 违反 MaxSkew (1) - if MaxSkew. 当 whenUnsatisfiable=ScheduleAnyway 时，用来为满足它的拓扑提供更高优先级。它是必填字段。默认值为 1，不允许使用 0。</p>

属性	类型	描述
minDomains	整数	<p>MinDomains 表示最少的合格域数。当匹配拓扑键的合格域数量小于 minDomains 时，Pod Topology Spread 会将"global minimum"视为 0，然后执行 Skew 计算。当匹配拓扑键的合格域数量等于或大于 minDomains 时，这个值对调度没有影响。因此，当有资格的域数量小于 minDomains 时，调度程序不会将这些域的 maxSkew Pod 调度到这些域。如果值为 nil，则约束的行为就像 MinDomains 等于 1。有效值为大于 0 的整数。当值不是 nil 时，WhenUnsatisfiable 必须是 DoNotSchedule。</p> <p>例如，在 3 区集群中，MaxSkew 设置为 2，MinDomains 被设置为 5，而带有与 2/2/2: zone1 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone3 zone3 zone3 区域数量小于 5 (MinDomains)，因此"global minimum"被视为 0。在这种情况下，无法调度具有相同 labelSelector 的新 pod，因为如果新 Pod 调度到三个区，则 computed skew 将为 3 (3 - 0)，它将违反 MaxSkew。</p> <p>这是一个 beta 字段，需要启用 MinDomainsInPodTopologySpread 功能门（默认启用）。</p>

属性	类型	描述
nodeAffinityPolicy	string	<p>NodeAffinityPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理 Pod 的 nodeAffinity/nodeSelector。选项为：- Honor：计算中仅包含与 nodeAffinity/nodeSelector 匹配的节点。忽略 Ignore: nodeAffinity/nodeSelector。所有节点包含在计算中。</p> <p>如果这个值为 nil，则行为等同于 Honor 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>
nodeTaintsPolicy	string	<p>NodeTaintsPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理节点污点。选项为：- Honor: 没有污点的节点，以及传入 pod 具有容量的污点节点被包含。- Ignore: node taint is ignored。包括所有节点。</p> <p>如果这个值为 nil，则行为等同于 Ignore 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>

属性	类型	描述
topologyKey	字符串	<p>topologyKey 是节点标签的密钥。带有具有此键和相同值标签的节点被视为在同一拓扑中。我们将每个 <key, value> 视为 "bucket", 并尝试在每个存储桶中均衡的 pod 数量。我们将域定义为拓扑的特定实例。另外, 我们定义了一个有资格的域, 其节点满足 nodeAffinityPolicy 和 nodeTaintsPolicy 的要求。例如, 如果 TopologyKey 为 "kubernetes.io/hostname", 则每个节点都是该拓扑的域。如果 TopologyKey 是 "topology.kubernetes.io/zone", 则每个区都是该拓扑的域。它是必填字段。</p>
whenUnsatisfiable	字符串	<p>whenUnsatisfiable 表示在不满足分散约束的情况下如何处理 pod。</p> <ul style="list-style-type: none"> - DoNotSchedule (默认) 告知调度程序不调度它。 - ScheduleAnyway 告知调度程序将 pod 调度到任何位置, 但为拓扑赋予更高优先级, 有助于减少偏移量。只有在每个可能的节点分配在某些拓扑中违反 "MaxSkew" 时, 对传入 pod 的一个约束被视为 "Unsatisfiable"。例如, 在 3 区集群中, MaxSkew 设置为 1, 具有相同 labelSelector 的 pod 分布为 3/1/1: zone1 zone3 zone3 P P P P If WhenUnsatisfiable 被设置为 DoNotSchedule, 传入的 pod 只能调度到 zone2 (zone3) 要成为 3/2/1 (3/1/2), 在 zone2 (zone3) 上作为 ActualSkew (2-1) 满足 MaxSkew (1)。换句话说, 集群仍然可以进行平衡, 但调度程序不会使其更不平衡。它是必填字段。 <p>可能枚举值: -</p> <ul style="list-style-type: none"> "DoNotSchedule" 指示调度程序在不满足约束时不调度 pod。 "ScheduleAnyway" 指示调度程序调度 pod, 即使没有满足限制。

14.13.1.245. .template.spec.volumes

描述

可由属于 pod 的容器挂载的卷列表。更多信息：
<https://kubernetes.io/docs/concepts/storage/volumes>

类型

数组

14.13.1.246. .template.spec.volumes[]

描述

卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

类型

object

必填

- name

属性	类型	描述
awsElasticBlockStore	object	代表 AWS 中的 Persistent Disk 资源。 挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。
azureDisk	对象	azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。
azureFile	对象	azureFile 代表主机上的 Azure File Service 挂载，并绑定到 pod。
cephfs	object	代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。
cinder	object	代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

属性	类型	描述
configMap	object	<p>将 ConfigMap 调整为卷。</p> <p>目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。</p>
csi	object	代表由外部 CSI 驱动程序管理的卷的源位置
downwardAPI	object	DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。
emptyDir	object	表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。
ephemeral	object	代表由普通存储驱动程序处理的临时卷。
fc	object	代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。
flexVolume	对象	FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。
flocker	object	代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。
gcePersistentDisk	object	<p>代表 Google Compute Engine 中的 Persistent Disk 资源。</p> <p>在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。</p>

属性	类型	描述
gitRepo	object	代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。 DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。
glusterfs	object	代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。
hostPath	object	代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。
iscsi	object	代表 ISCSI 磁盘。ISCSI 卷只能以读写模式挂载。ISCSI 卷支持所有权管理和 SELinux 重新标记。
name	string	卷的名称。必须是 DNS_LABEL，且在 pod 中唯一。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。 PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。
photonPersistentDisk	object	代表 Photon Controller 持久磁盘资源。
portworxVolume	object	PortworxVolumeSource 代表一个 Portworx volume 资源。
projected	object	代表投射卷源

属性	类型	描述
quobyte	object	代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。
rbd	object	代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。
scaleIO	object	ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷
secret	object	将 Secret 调整为卷。 目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。
storageos	object	代表 StorageOS 持久性卷资源。
vsphereVolume	object	代表 vSphere 卷资源。

14.13.1.247. .template.spec.volumes[].awsElasticBlockStore

描述

代表 AWS 中的 Persistent Disk 资源。

挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumeID**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4"，"xfs"，"ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

属性	类型	描述
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。
readOnly	布尔值	readonly 值 true 将强制 VolumeMounts 中的 readOnly 设置。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	VolumeID 是 AWS (Amazon EBS 卷) 中的持久磁盘资源的唯一 ID。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

14.13.1.248. .template.spec.volumes[].azureDisk

描述

azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

类型

object

必填

- **diskName**
- **diskURI**

属性	类型	描述
cacheMode	string	cacheMode 是主机缓存模式：None、Read Only、Read Write。 可能枚举值：- "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskname 是 blob 存储中的数据磁盘的名称
diskURI	string	diskURI 是 blob 存储中数据磁盘的 URI

属性	类型	描述
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。
kind	string	kind expected 值是 Shared: multiple blob disk per storage account Managed: azure disk per storage account Managed: azure managed data disk (only in managed availability set))。默认为 shared possible enum 值: - "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。

14.13.1.249. .template.spec.volumes[].azureFile

描述

azureFile 代表主机上的 Azure File Service 挂载, 并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

属性	类型	描述
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称

属性	类型	描述
shareName	string	sharename 是 azure 共享名称

14.13.1.250. .template.spec.volumes[].cephfs

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您 在同一个命名空间中 找到引用的对象。

属性	类型	描述
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

14.13.1.251. .template.spec.volumes[].cephfs.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.252. .template.spec.volumes[].cinder

描述

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

属性	类型	描述
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
volumeID	string	用于识别 cinder 中的卷的卷 ID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

14.13.1.253. .template.spec.volumes[].cinder.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.254. .template.spec.volumes[].configMap

描述

将 ConfigMap 调整为卷。

目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选的：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.13.1.255. .template.spec.volumes[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.13.1.256. .template.spec.volumes[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.13.1.257. .template.spec.volumes[].csi

描述

代表由外部 CSI 驱动程序管理的卷的源位置

类型

object

必填

- **driver**

属性	类型	描述
driver	string	driver 是处理这个卷的 CSI 驱动程序的名称。请参考您的管理员获取集群中注册的正确名称。

属性	类型	描述
fsType	string	要挂载的 fstype。Ex. "ext4", "xfs", "ntfs".如果没有提供, 空值会被传递给关联的 CSI 驱动程序, 该驱动程序将决定要应用的默认文件系统。
nodePublishSecretRef	object	LocalObjectReference 包含足够的信息, 供您在同一个命名空间中找到引用的对象。
readOnly	布尔值	readonly 指定卷的只读配置。默认为 false (读/写)。
volumeAttributes	对象 (字符串)	volumeAttributes 存储传递给 CSI 驱动程序的特定于驱动程序的属性。如需支持的值, 请参考您的驱动程序文档。

14.13.1.258. .template.spec.volumes[].csi.nodePublishSecretRef

描述

LocalObjectReference 包含足够的信息, 供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.259. .template.spec.volumes[].downwardAPI

描述

DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	可选：默认在创建的文件中使用的模式位。必须是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	array	items 是 Downward API 卷文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.13.1.260. .template.spec.volumes[].downwardAPI.items

描述

items 是 Downward API 卷文件列表

类型

数组

14.13.1.261. .template.spec.volumes[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- path

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。

属性	类型	描述
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.13.1.262. .template.spec.volumes[].downwardAPI.items[].fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.13.1.263. .template.spec.volumes[].downwardAPI.items[].resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- resource

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.13.1.264. .template.spec.volumes[].emptyDir

描述

表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
中	string	Medium 代表这个目录应该支持哪些存储类型。默认值为 "", 它需要使用节点的默认介质。必须是空字符串（默认）或 Memory。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit	数量	sizeLimit 是此 EmptyDir 卷所需的本地存储总量。大小限制也适用于内存介质。内存用量 EmptyDir 的最大用法是此处指定的 SizeLimit 和 pod 中所有容器的内存限值总和。默认值为 nil，表示限制未定义。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir

14.13.1.265. .template.spec.volumes[].ephemeral

描述

代表由普通存储驱动程序处理的临时卷。

类型

object

属性	类型	描述
volumeClaimTemplate	object	PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

14.13.1.266. .template.spec.volumes[].ephemeral.volumeClaimTemplate

描述

PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

类型

object

必填

- **spec**

属性	类型	描述
metadata	ObjectMeta	可以包含在创建时复制到 PVC 中的标签和注解。不允许其他字段，并在验证过程中被拒绝。
spec	对象	PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

14.13.1.267. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含卷应具有的所有所需访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	对象	TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

属性	类型	描述
dataSourceRef	object	<p>dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。</p>
resources	对象	ResourceRequirements 描述计算资源要求。
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

属性	类型	描述
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统进行或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

14.13.1.268. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

描述

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

14.13.1.269. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

描述

dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。

dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称
namespace	string	namespace 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 gateway.networking.k8s.io/ReferenceGrant 对象，以允许该命名空间的所有者接受引用。详情请查看 ReferenceGrant 文档。(alpha)此字段需要启用 CrossNamespaceVolumeDataSource 功能门。

14.13.1.270. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.13.1.271. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.13.1.272. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claims

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- name

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.13.1.273. .template.spec.volumes[].fc

描述

代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
LUN	整数	LUN 是可选的: FC 目标 lun 号
readOnly	布尔值	readonly 是可选的: 默认为 false (读/写)。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组 (字符串)	targetWWN 可选: FC 目标全球名称(WWN)
wwids	数组 (字符串)	wwids 可选: FC 卷全局广泛的标识符(wwids) Either wwids 或 targetWWNs 和 lun 的组合必须同时设置, 但不能同时设置。

14.13.1.274. .template.spec.volumes[].flexVolume

描述

FlexVolume 代表一个通用卷资源, 它使用基于 exec 的插件置备/附加。

类型

object

必填

- driver

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs".默认文件系统取决于 FlexVolume 脚本。
options	对象 (字符串)	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

14.13.1.275. .template.spec.volumes[].flexVolume.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.276. .template.spec.volumes[].flocker

描述

代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。

类型

object

属性	类型	描述
datasetName	string	datasetName 是存储在 Flocker 的 dataset 的 dataset 的 dataset 的 Name。
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

14.13.1.277. .template.spec.volumes[].gcePersistentDisk

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **pdName**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

属性	类型	描述
pdName	string	pdName 是 GCE 中的 PD 资源的唯一名称。用于识别 GCE 中的磁盘。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

14.13.1.278. .template.spec.volumes[].gitRepo

描述

代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。
DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。

类型

object

必填

- 软件仓库

属性	类型	描述
目录	string	directory 是目标目录名称。不得包含或以 './' 开头。如果提供了 '.', 则卷目录将是 git 存储库。否则，如果指定，卷将包含具有指定名称的子目录中的 git 存储库。
软件仓库	string	repository 是 URL
revision	string	revision 是指定修订版本的提交哈希。

14.13.1.279. .template.spec.volumes[].glusterfs

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

14.13.1.280. .template.spec.volumes[].hostPath

描述

代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- path

属性	类型	描述
path	string	主机上目录的路径。如果路径是符号链接，它将遵循到实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath

属性	类型	描述
type	string	<p>HostPath Volume Defaults 为 "" more info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>可能枚举值： - "" 对于向后兼容，如果未设置 - "BlockDevice" A 块设备必须存在于给定路径 - "CharDevice" A 字符设备必须存在于给定路径上 - "Directory" A 目录必须存在于给定路径 - "DirectoryOrCreate" 中（如果给定路径不存在），将根据需要创建一个空目录，且文件模式为 0755，其具有与 Kubelet 相同的组和所有权。 - "File" A 文件必须存在于给定路径上 - "FileOrCreate" 如果给定路径上不存在，将根据需要创建一个空文件，该文件与文件模式 0644 具有相同的组和所有权。 - "Socket" A UNIX 套接字必须存在于给定路径上。</p>

14.13.1.281. .template.spec.volumes[].iscsi

描述

代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是目标 iSCSI 限定名称。
iscsiInterface	string	iscsiInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 代表 iSCSI 目标 Lun 编号。
门户	数组（字符串）	门户是 iSCSI 目标门户列表。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值（通常是 TCP 端口 860 和 3260），则门户可以是 IP 或 ip_addr:port。

14.13.1.282. .template.spec.volumes[].iscsi.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.283. .template.spec.volumes[].nfs

描述

代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **server**
- **path**

属性	类型	描述
path	string	NFS 服务器导出的路径。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	布尔值	此处的只读将强制使用只读权限挂载 NFS 导出。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server 是 NFS 服务器的主机名或 IP 地址。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs

14.13.1.284. .template.spec.volumes[].persistentVolumeClaim

描述

PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。

类型

object

必填

- **claimName**

属性	类型	描述
claimName	string	claimName 是使用此卷与 pod 相同的命名空间中的 PersistentVolumeClaim 的名称。 更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	布尔值	readonly 将强制 VolumeMount 中的 ReadOnly 设置。默认 false。

14.13.1.285. .template.spec.volumes[].photonPersistentDisk

描述

代表 Photon Controller 持久磁盘资源。

类型

object

必填

- **pdID**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。
pdID	string	pdID 是标识 Photon Controller 持久磁盘的 ID

14.13.1.286. .template.spec.volumes[].portworxVolume

描述

PortworxVolumeSource 代表一个 Portworx volume 资源。

类型

object

必填

- **volumeID**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 代表要挂载的文件系统类型，必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs".如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
volumeID	string	VolumeID 唯一标识一个 Portworx 卷

14.13.1.287. .template.spec.volumes[].projected

描述

代表投射卷源

类型

object

属性	类型	描述
defaultMode	整数	defaultMode 是默认用来对创建文件设置权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
sources	数组	sources 是卷投射列表
sources[]	对象	可与其他支持的卷类型一起投射的预测

14.13.1.288. .template.spec.volumes[].projected.sources

描述

sources 是卷投射列表

类型

数组

14.13.1.289. `.template.spec.volumes[].projected.sources[]`

描述

可与其他支持的卷类型一起投射的预测

类型

object

属性	类型	描述
configMap	object	将 ConfigMap 调整为投射卷。 目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。
downwardAPI	object	代表下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。
secret	object	将 secret 适应到投射卷中。 目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。
serviceAccountToken	object	ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

14.13.1.290. `.template.spec.volumes[].projected.sources[].configMap`

描述

将 ConfigMap 调整为投射卷。

目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.13.1.291. .template.spec.volumes[].projected.sources[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.13.1.292. .template.spec.volumes[].projected.sources[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- key
- path

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.13.1.293. .template.spec.volumes[].projected.sources[].downwardAPI

描述

代表下下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。

类型

object

属性	类型	描述
items	array	items 是 DownwardAPIVolume 文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.13.1.294. .template.spec.volumes[].projected.sources[].downwardAPI.items

描述

items 是 DownwardAPIVolume 文件列表

类型

数组

14.13.1.295. .template.spec.volumes[].projected.sources[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- **path**

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的相对路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.13.1.296. .template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.13.1.297. .template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceField

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- resource

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.13.1.298. .template.spec.volumes[].projected.sources[].secret

描述

将 secret 适应到投射卷中。

目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。

类型

object

属性	类型	描述
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥

14.13.1.299. .template.spec.volumes[].projected.sources[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.13.1.300. .template.spec.volumes[].projected.sources[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。

属性	类型	描述
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.13.1.301. .template.spec.volumes[].projected.sources[].serviceAccountToken

描述

ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

类型

object

必填

- **path**

属性	类型	描述
受众	string	受众是令牌的预期受众。令牌的接收者必须使用令牌使用者中指定的标识符来识别自己，否则应拒绝令牌。使用者默认为 apiserver 的标识符。
expirationSeconds	整数	expirationSeconds 是服务帐户令牌请求持续时间。随着令牌方法到期，kubelet 卷插件将主动轮转服务帐户令牌。如果令牌存在的时间超过这个值的 80%，或者令牌超过 24 小时，则 kubelet 将开始尝试轮转令牌，且必须至少为 10 分钟。
path	string	path 是相对于文件挂载点的路径，以将令牌项目到其中。

14.13.1.302. .template.spec.volumes[].quobyte

描述

代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **registry**

- 卷

属性	类型	描述
group	string	将卷访问映射到 Default 的组没有组
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Quobyte 卷。默认为false。
registry	string	registry 代表一个或多个 Quobyte Registry 服务，作为字符串指定为 host:port 对（多个条目用逗号分隔），后者充当卷的中央 registry
tenant	string	在带有动态置备 Quobyte 卷的后端中拥有给定 Quobyte 卷的租户，值由插件设置
user	string	用户将卷访问权限映射到默认值到 serviceaccount 用户
卷	string	卷是一个字符串，它根据名称引用已创建的 Quobyte 卷。

14.13.1.303. .template.spec.volumes[].rbd

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **monitor**
- **image**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd

属性	类型	描述
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

14.13.1.304. .template.spec.volumes[].rbd.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.305. .template.spec.volumes[].scaleIO

描述

ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- **gateway**
- **system**
- **secretRef**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认为 "xfs".
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
sslEnabled	布尔值	sslEnabled Flag 启用/禁用与网关的 SSL 通信，默认为 false

属性	类型	描述
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

14.13.1.306. .template.spec.volumes[].scaleIO.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.307. .template.spec.volumes[].secret

描述

将 Secret 调整为卷。

目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥
secretName	string	secretName 是要使用的 pod 命名空间中的 secret 名称。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#secret

14.13.1.308. .template.spec.volumes[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.13.1.309. .template.spec.volumes[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.13.1.310. .template.spec.volumes[].storageos

描述

代表 StorageOS 持久性卷资源。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

属性	类型	描述
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

14.13.1.311. .template.spec.volumes[].storageos.secretRef

描述

LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.13.1.312. .template.spec.volumes[].vsphereVolume

描述

代表 vSphere 卷资源。

类型

object

必填

- **volumePath**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。
storagePolicyID	string	StoragePolicyID 是与 StoragePolicyName 关联的存储 Policy Based Management (SPBM)配置集 ID。
storagePolicyName	string	StoragePolicyName 是存储基于策略的管理(SPBM)配置集名称。
volumePath	string	volumePath 是标识 vSphere 卷 vmdk 的路径

14.13.2. API 端点

可用的 API 端点如下：

- **/api/v1/podtemplates**
 - **GET**: 列出或监视类型为 PodTemplate 的对象
- **/api/v1/watch/podtemplates**
 - **GET**: 观察单个对 PodTemplate 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/podtemplates**
 - **DELETE**：删除 PodTemplate 的集合
 - **GET**: 列出或监视类型为 PodTemplate 的对象
 - **POST**：创建 PodTemplate
- **/api/v1/watch/namespaces/{namespace}/podtemplates**
 - **GET**: 观察单个对 PodTemplate 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/podtemplates/{name}**
 - **DELETE**：删除 PodTemplate
 - **GET**：读取指定的 PodTemplate
 - **PATCH**：部分更新指定的 PodTemplate
 - **PUT**：替换指定的 PodTemplate
- **/api/v1/watch/namespaces/{namespace}/podtemplates/{name}**
 - **GET**: 观察对 PodTemplate 类型的对象的更改。已弃用：使用带有列表操作的 'watch' 参数, 而且过滤到带有 fieldSelector 参数的单个项目

而是过滤到带有 fieldSelector 参数的单个项目。

14.13.2.1. /api/v1/podtemplates

表 14.342. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 PodTemplate 的对象

表 14.343. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodTemplateList 模式
401 - Unauthorized	空

14.13.2.2. /api/v1/watch/podtemplates

表 14.344. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PodTemplate 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.345. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.13.2.3. /api/v1/namespaces/{namespace}/podtemplates

表 14.346. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.347. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除 PodTemplate 的集合

表 14.348. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.349. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.350. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PodTemplate 的对象

表 14.351. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.352. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodTemplateList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 PodTemplate

表 14.353. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.354. 主体参数

参数	类型	描述
正文 (body)	PodTemplate 模式	

表 14.355. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodTemplate 模式
201 - Created	PodTemplate 模式
202 - Accepted	PodTemplate 模式
401 - Unauthorized	空

14.13.2.4. /api/v1/watch/namespaces/{namespace}/podtemplates

表 14.356. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.357. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PodTemplate 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.358. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.13.2.5. /api/v1/namespaces/{namespace}/podtemplates/{name}

表 14.359. 全局路径参数

参数	类型	描述
name	string	PodTemplate 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.360. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PodTemplate

表 14.361. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.362. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.363. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>PodTemplate</code> 模式
202 - Accepted	<code>PodTemplate</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 PodTemplate

表 14.364. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>PodTemplate</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 PodTemplate

表 14.365. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.366. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.367. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodTemplate 模式
201 - Created	PodTemplate 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 PodTemplate

表 14.368. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.369. 主体参数

参数	类型	描述
正文 (body)	PodTemplate 模式	

表 14.370. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodTemplate 模式
201 - Created	PodTemplate 模式
401 - Unauthorized	空

14.13.2.6. /api/v1/watch/namespaces/{namespace}/podtemplates/{name}

表 14.371. 全局路径参数

参数	类型	描述
name	string	PodTemplate 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.372. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind PodTemplate. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.373. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.14. REPLICATIONCONTROLLER [V1]

描述

ReplicationController 代表复制控制器的配置。

类型

对象

14.14.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	如果 ReplicationController 的 Labels 为空，则默认为与复制控制器管理的 Pod 相同。标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	ReplicationControllerSpec 是复制控制器的规格。
status	object	ReplicationControllerStatus 代表复制控制器的当前状态。

14.14.1.1. .spec

描述

ReplicationControllerSpec 是复制控制器的规格。

类型

object

属性	类型	描述
minReadySeconds	整数	新创建的 pod 应该就绪的最少秒数，且没有其容器崩溃，以便它被视为可用。默认为 0 (pod 在就绪后将被视为可用)
replicas	整数	replicas 是所需副本数量。这是区分显式零和未指定的指针。默认为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller
selector	对象 (字符串)	selector 是 pod 的标签查询，它应该与 Replicas 数匹配。如果 Selector 为空，则默认为 Pod 模板上的标签。标记必须匹配的键和值，以便由此复制控制器控制（如果为空默认为 Pod 模板上的标签）。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
模板	object	PodTemplateSpec 描述了从模板创建 pod 时应该具有的数据

14.14.1.2. .spec.template

描述

PodTemplateSpec 描述了从模板创建 pod 时应该具有的数据

类型

object

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec 是 pod 的描述。

14.14.1.3. .spec.template.spec

描述

PodSpec 是 pod 的描述。

类型

object

必填

- **containers**

属性	类型	描述
activeDeadlineSeconds	整数	在系统主动尝试标记失败并终止相关的容器前，pod 可以在相对于 StartTime 的节点上处于活跃状态。值必须是正整数。
关联性	object	affinity 是一组关联性调度规则。
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 指明是否应该自动挂载服务帐户令牌。
containers	array	属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。
containers[]	对象	要在 pod 中运行的单一应用程序容器。
dnsConfig	object	PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

属性	类型	描述
dnsPolicy	字符串	<p>为 pod 设置 DNS 策略。默认为 "ClusterFirst"。有效值为 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' 或 'None'。DNSConfig 中给定的 DNS 参数将与选择与 DNSPolicy 的策略合并。要将 DNS 选项与 hostNetwork 一起设置，您必须将 DNS 策略明确指定为 'ClusterFirstWithHostNet'。</p> <p>可能枚举值： - "ClusterFirst" 表示 pod 应该先使用集群 DNS，除非 hostNetwork 为 true，如果可用，则回退到默认的（由 kubelet 决定）DNS 设置。 - "ClusterFirstWithHostNet" 表示 pod 应该首先使用集群 DNS，如果可用，则回退到它。在默认的（由 kubelet 决定）DNS 设置中。 - "Default" 表示 pod 应该使用默认（由 kubelet 决定）DNS 设置。 - "None" 表示 pod 应该使用空的 DNS 设置。名称服务器和搜索路径等 DNS 参数应该通过 DNSConfig 定义。</p>
enableServiceLinks	布尔值	<p>EnableServiceLinks 指示是否应将有关服务的信息注入到 pod 的环境变量中，与 Docker 链接的语法匹配。可选：默认为 true。</p>
ephemeralContainers	array	<p>此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。</p>

属性	类型	描述
ephemeralContainers[]	object	<p>EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。</p> <p>要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。</p>
hostAliases	array	hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。
hostAliases[]	对象	HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。
hostIPC	布尔值	使用主机的 ipc 命名空间。可选：默认为 false。
hostNetwork	布尔值	为此 pod 请求的主机网络。使用主机的网络命名空间。如果设置了这个选项，则必须指定要使用的端口。默认为 false。
hostPID	布尔值	使用主机的 pid 命名空间。可选：默认为 false。
hostUsers	布尔值	<p>使用主机的用户命名空间。可选：默认为 true。如果设置为 true 或不存在，则 pod 将在主机用户命名空间中运行，当 pod 需要可供主机用户命名空间使用的功能时，如加载带有 CAP_SYS_MODULE 的内核模块。当设置为 false 时，会为 pod 创建一个新的 usersns。设置 false 有助于缓解容器中断漏洞，即使允许用户以 root 身份运行其容器，而无需实际在主机上具有 root 特权。此字段是 alpha-level，仅被启用 UserNamespacesSupport 功能的服务器所接受。</p>

属性	类型	描述
hostname	字符串	如果没有指定 Pod 的主机名，则 pod 的主机名将设置为系统定义的值。
imagePullSecrets	array	imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息： https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	对象	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
initContainers	array	属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
initContainers[]	对象	要在 pod 中运行的单一应用程序容器。
nodeName	字符串	nodeName 是一个将此 pod 调度到特定节点请求。如果是非空，调度程序只会将此 pod 调度到该节点上，假设它符合资源要求。

属性	类型	描述
nodeSelector	对象 (字符串)	nodeSelector 是一个选择器，必须为 true 才能使 pod 适合节点。必须与 pod 标签匹配的选择器，才能调度到该节点上。更多信息： https://kubernetes.io/docs/concepts/configuration/assign-pod-node/
OS	object	PodOS 定义 pod 的 OS 参数。
开销	对象 (数量)	开销代表了与为给定 RuntimeClass 运行 pod 相关的资源开销。此字段将由 RuntimeClass 准入控制器在准入时自动填充。如果启用了 RuntimeClass 准入控制器，则无法在 Pod 创建请求中设置开销。RuntimeClass 准入控制器将拒绝 Pod 创建已设置开销的请求。如果在 PodSpec 中配置并选择了 RuntimeClass，则 Overhead 将被设置为对应的 RuntimeClass 中定义的值，否则它将保留为未设置并被视为零。更多信息： https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md
preemptionPolicy	字符串	preemptionPolicy 是抢占优先级较低的 pod 的策略。Never, PreemptLowerPriority 之一。如果未设置，则默认为 PreemptLowerPriority。 可能枚举值：- "Never" 表示 pod 不会抢占优先级较低的其他 pod。 - "PreemptLowerPriority" 表示 pod 可以抢占优先级较低的其他 pod。
priority	整数	优先级值。各种系统组件使用此字段查找 pod 的优先级。启用 Priority Admission Controller 时，它会阻止用户设置此字段。准入控制器从 PriorityClassName 填充此字段。数值越大，优先级越高。

属性	类型	描述
priorityClassName	字符串	如果指定，则表示 pod 的优先级。 "system-node-critical"和"system-cluster-critical"是两个特殊关键字，表示前者最高的优先级。必须使用该名称创建 PriorityClass 对象来定义任何其他名称。如果没有指定，如果没有默认值，pod 优先级将为 default 或零。
readinessGates	array	如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪： https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	对象	PodReadinessGate 包含对 pod 条件的引用
resourceClaims	数组	ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。
resourceClaims[]	object	PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

属性	类型	描述
restartPolicy	字符串	<p>为 pod 中的所有容器重启策略。Always、OnFailure、Never 之一。在某些上下文中，仅允许这些值的子集。默认值为 Always。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy</p> <p>可能枚举值：- "Always" - "Never" - "OnFailure"</p>
runtimeClassName	字符串	<p>runtimeClassName 指的是 node.k8s.io 组中的 RuntimeClass 对象，它应该用于运行此 pod。如果没有 RuntimeClass 资源与命名类匹配，则不会运行 pod。如果未设置或为空，则使用 "legacy" RuntimeClass，它是一个带有使用默认运行时处理器的空定义的隐式类。更多信息： https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class</p>
schedulerName	字符串	<p>如果指定，pod 将由指定的调度程序分配。如果没有指定，pod 将默认调度程序分配。</p>
schedulingGates	数组	<p>SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。</p> <p>SchedulingGates 只能在创建 pod 时设置，之后只能删除。</p> <p>这是 PodSchedulingReadiness 功能门启用的 beta 功能。</p>
schedulingGates[]	object	<p>PodSchedulingGate 与 Pod 关联，以保护其调度。</p>

属性	类型	描述
securityContext	object	PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。
serviceAccount	字符串	deprecatedServiceAccount 是 ServiceAccountName 的前文别名。deprecated : 改为使用 serviceAccountName。
serviceAccountName	字符串	serviceAccountName 是用于运行此 pod 的 ServiceAccount 的名称。更多信息： https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
setHostnameAsFQDN	布尔值	如果为 true, 则 pod 的主机名将配置为 pod 的 FQDN, 而不是叶名称 (默认值)。在 Linux 容器中, 这意味着在内核的 hostname 字段中设置 FQDN (struct utsname 的 nodename 字段)。在 Windows 容器中, 这意味着将 registry 键 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 的 registry 值设置为 FQDN。如果 pod 没有 FQDN, 则无效。默认为 false。
shareProcessNamespace	布尔值	在 pod 中所有容器间共享单个进程命名空间。设置此容器后, 将能够查看和信号来自同一 pod 中其他容器的进程, 并且每个容器中的第一个进程不会被分配 PID 1。无法同时设置 hostPID 和 ShareProcessNamespace。可选: 默认为 false。
subdomain	字符串	如果指定, 完全限定的 Pod 主机名将是 "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>"。如果没有指定, pod 根本不会具有 domainname。

属性	类型	描述
terminationGracePeriodSeconds	整数	pod 需要安全终止的可选持续时间（以秒为单位）。在删除请求中可能会减少。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。如果这个值为 nil，则使用默认的宽限期。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。默认值为 30 秒。
容限 (tolerations)	array	如果指定，pod 的容限。
tolerations[]	对象	附加到此 Tolerations 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。
topologySpreadConstraints	array	topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。
topologySpreadConstraints[]	对象	TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。
卷	array	可由属于 pod 的容器挂载的卷列表。更多信息： https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	对象	卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

14.14.1.4. .spec.template.spec.affinity

描述

affinity 是一组关联性调度规则。

类型

object

属性	类型	描述
nodeAffinity	object	节点关联性是一组节点关联性调度规则。
podAffinity	object	Pod 关联性是一组 pod 关联性调度规则。
podAntiAffinity	object	Pod 反关联性是一组 pod 反关联性调度规则。

14.14.1.5. .spec.template.spec.affinity.nodeAffinity

描述

节点关联性是一组节点关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中（如果节点与相应的 matchExpressions 匹配）；节点具有最高总和和最高的节点。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	空首选调度术语与所有带有隐式权重 0 的对象匹配（例如，它是一个 no-op）。null 首选调度术语不匹配任何对象（例如，也是 no-op）。
requiredDuringSchedulingIgnoredDuringExecution	object	节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

14.14.1.6. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuring

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、

requiredDuringScheduling 关联性表达式等) 的每个节点, 它通过迭代此字段元素并将 "weight" 添加到摘要中 (如果节点与相应的 matchExpressions 匹配) ; 节点具有最高总和和最高的节点。

类型

数组

14.14.1.7. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringI

描述

空首选调度术语与所有带有隐式权重 0 的对象匹配 (例如, 它是一个 no-op)。null 首选调度术语不匹配任何对象 (例如, 也是 no-op)。

类型

object

必填

- weight
- preference

属性	类型	描述
preference	object	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。
weight	整数	与对应的 nodeSelectorTerm 匹配的权重, 范围为 1-100。

14.14.1.8. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringI

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器, 其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。

属性	类型	描述
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.14.1.9. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuring

描述

节点标签(label)的节点选择器要求列表。

类型

数组

14.14.1.10. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuring

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.14.1.11. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuring

描述

节点字段的节点选择器要求列表。

类型

数组

14.14.1.12. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuring**描述**

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- key
- operator

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.14.1.13. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE**描述**

节点选择器表示一个或多个对一组节点的标签查询结果的联合；即，它代表节点选择器术语代表的选择器的 OR。

类型

object

必填

- nodeSelectorTerms

属性	类型	描述
nodeSelectorTerms	array	必需。节点选择器术语列表。条款是 ORed。
nodeSelectorTerms[]	对象	null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

14.14.1.14. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE

描述

必需。节点选择器术语列表。条款是 ORed。

类型

数组

14.14.1.15. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE

描述

null 或 empty 节点选择器术语不匹配任何对象。它们的要求是 AND。TopologySelectorTerm 类型实现 NodeSelectorTerm 的子集。

类型

object

属性	类型	描述
matchExpressions	array	节点标签(label)的节点选择器要求列表。
matchExpressions[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。
matchFields	array	节点字段的节点选择器要求列表。
matchFields[]	对象	节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

14.14.1.16. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE

描述

节点标签(label)的节点选择器要求列表。

类型

数组

14.14.1.17. `.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE`

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**
- **operator**

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.14.1.18. `.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE`

描述

节点字段的节点选择器要求列表。

类型

数组

14.14.1.19. `.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE`

描述

节点选择器要求是一个选择器，其中包含值、键和与键和值关联的操作器。

类型

object

必填

- **key**

- operator

属性	类型	描述
key	字符串	选择器应用到的标签键。
operator	字符串	代表键与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。gt, 和 Lt. possible enum 值 : - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	数组 (字符串)	字符串值数组。如果运算符是 In 或 NotIn, 则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist, 则值数组必须为空。如果 Operator 是 Gt 或 Lt, 则值数组必须具有单个元素, 该元素将解释为整数。这个数组会在策略性合并补丁中被替换。

14.14.1.20. .spec.template.spec.affinity.podAffinity

描述

Pod 关联性是一组 pod 关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点, 但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点, 即对于满足所有调度要求 (资源请求、requiredDuringScheduling 关联性表达式等) 的每个节点, 它通过迭代此字段元素并将 "weight" 添加到摘要中, 如果节点具有与相应 podAffinityTerm 匹配的 pod; 具有最高总和最高的节点是首选的节点。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加, 以查找首选节点。

属性	类型	描述
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.14.1.21. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringE

描述

调度程序希望将 pod 调度到满足此字段指定的关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 关联性表达式等）的每个节点，它通过迭代此字段元素并将 "weight" 添加到摘要中，如果节点具有与相应 podAffinityTerm 匹配的 pod；具有最高总和最高的节点是首选的节点。

类型

数组

14.14.1.22. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringI

描述

所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- weight
- podAffinityTerm

属性	类型	描述
----	----	----

属性	类型	描述
podAffinityTerm	object	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

14.14.1.23. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuring

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- **topologyKey**

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。

属性	类型	描述
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.14.1.24. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringE

描述

如果在调度时不满足此字段指定的关联性要求，则 pod 不会调度到该节点上。如果在 pod 执行期间由此字段指定的关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

数组

14.14.1.25. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringE

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。

属性	类型	描述
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.14.1.26. .spec.template.spec.affinity.podAntiAffinity

描述

Pod 反关联性是一组 pod 反关联性调度规则。

类型

object

属性	类型	描述
preferredDuringSchedulingIgnoredDuringExecution	array	调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、requiredDuringScheduling 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 podAffinityTerm 匹配的 pod；如果节点具有与相应 podAffinityTerm 匹配的 pod，则该节点是首选的。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	所有匹配的 WeightedPodAffinityTerm 字段的权重都会被添加，以查找首选节点。

属性	类型	描述
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 <code>podAffinityTerm</code> 对应的节点列表会被干扰，例如，所有术语都必须满足。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（与给定命名空间匹配的 <code>labelSelector</code> 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 <code>co-located</code> 定义在带有键 <code><topologyKey></code> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

14.14.1.27. `.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDu`

描述

调度程序希望将 pod 调度到满足此字段指定的反关联性表达式的节点，但可以选择违反了一个或多个表达式的节点。最首选节点是具有最大权重总和的节点，即对于满足所有调度要求（资源请求、`requiredDuringScheduling` 反关联性表达式等）的每个节点，它通过迭代此字段元素来计算和将"权重"添加到节点具有与相应 `podAffinityTerm` 匹配的 pod；如果节点具有与相应 `podAffinityTerm` 匹配的 pod，则该节点是首选的。

类型

数组

14.14.1.28. `.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDu`

描述

所有匹配的 `WeightedPodAffinityTerm` 字段的权重都会被添加，以查找首选节点。

类型

object

必填

- `weight`
- `podAffinityTerm`

属性	类型	描述
----	----	----

属性	类型	描述
podAffinityTerm	object	定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配
weight	整数	与对应的 podAffinityTerm 匹配的权重，范围为 1-100。

14.14.1.29. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDu

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- **topologyKey**

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。

属性	类型	描述
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector.null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.14.1.30. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDurii

描述

如果在调度时不满足由此字段指定的反关联性要求，则 pod 不会调度到节点上。如果在 pod 执行期间由此字段指定的反关联性要求满足（例如，因为 pod 标签更新），则系统可能无法最终尝试从其节点中驱除 pod。当有多个元素时，每个 podAffinityTerm 对应的节点列表会被干扰，例如，所有术语都必须满足。

类型

数组

14.14.1.31. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDurii

描述

定义一组 pod（与给定命名空间匹配的 labelSelector 匹配），此 pod 应该与 pod 共存（关联性）或不共存（反关联性）匹配，其中 co-located 定义在带有键 <topologyKey> 标签的节点上运行，其标签的值与运行 pod 的任何节点匹配

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	labelSelector	一个对一组资源的标签查询，本例中为 pod。

属性	类型	描述
namespaceSelector	labelSelector	对术语应用到的命名空间集合的标签查询。术语应用于此字段选择的命名空间的联合，以及 namespaces 字段中列出的命名空间。null selector 和 null 或 empty namespace 列表表示"此 pod 的命名空间"。空选择器({})与所有命名空间匹配。
命名空间	数组（字符串）	namespaces 指定术语应用到的命名空间名称的静态列表。术语应用于此字段中列出的命名空间，由 namespaceSelector. null 或空命名空间列表和 null namespaceSelector 选择的命名空间合并意味着"此 pod 的命名空间"。
topologyKey	字符串	此 pod 应该与指定命名空间中 labelSelector 匹配的 pod 在一起（关联性）或不并置（反关联性），其中 co-located 定义为在具有键 topologyKey 标签的节点中运行。不允许空的 topologyKey。

14.14.1.32. .spec.template.spec.containers

描述

属于 pod 的容器列表。容器目前无法添加或删除。Pod 中必须至少有一个容器。无法更新。

类型

数组

14.14.1.33. .spec.template.spec.containers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- name

属性	类型	描述
----	----	----

属性	类型	描述
args	数组 (字符串)	入口点的参数。如果没有提供, 则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组 (字符串)	ENTRYPOINT 数组不会在 shell 中执行。如果没有提供, 则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> , 它允许转义 <code>\$(VAR_NAME)</code> 语法: 例如: <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 <code>C_IDENTIFIER</code> 。当容器启动时, 所有无效的密钥都会被报告为事件。当多个源中存在键时, 与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

属性	类型	描述
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。

属性	类型	描述
ports	array	要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。

属性	类型	描述
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。

属性	类型	描述
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统上的 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.14.1.34. .spec.template.spec.containers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.14.1.35. .spec.template.spec.containers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。

属性	类型	描述
valueFrom	object	EnvVarSource 代表 EnvVarar 的值的一个源。

14.14.1.36. .spec.template.spec.containers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVarar 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.14.1.37. .spec.template.spec.containers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

属性	类型	描述
----	----	----

14.14.1.38. .spec.template.spec.containers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.14.1.39. .spec.template.spec.containers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.14.1.40. .spec.template.spec.containers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.14.1.41. .spec.template.spec.containers[].envFrom**描述**

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.14.1.42. .spec.template.spec.containers[].envFrom[]**描述**

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

属性	类型	描述
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.14.1.43. .spec.template.spec.containers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.14.1.44. .spec.template.spec.containers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

属性	类型	描述
optional	布尔值	指定是否需要定义 Secret

14.14.1.45. .spec.template.spec.containers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.14.1.46. .spec.template.spec.containers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.14.1.47. .spec.template.spec.containers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组 (字符串)	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.48. .spec.template.spec.containers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

属性	类型	描述
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.49. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.50. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.51. .spec.template.spec.containers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.52. .spec.template.spec.containers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCP socket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCP socket action 描述了基于打开套接字的操作

14.14.1.53. .spec.template.spec.containers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.54. .spec.template.spec.containers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.55. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders**描述**

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.56. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.57. `..spec.template.spec.containers[].lifecycle.preStop.tcpSocket`

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.58. `..spec.template.spec.containers[].livenessProbe`

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。

属性	类型	描述
<code>grpc</code>	<code>object</code>	GRPC 指定涉及 GRPC 端口的操作。
<code>httpGet</code>	<code>object</code>	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
<code>initialDelaySeconds</code>	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
<code>periodSeconds</code>	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
<code>successThreshold</code>	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
<code>tcpSocket</code>	<code>object</code>	TCPSocketAction 描述了基于打开套接字的操作
<code>terminationGracePeriodSeconds</code>	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
<code>timeoutSeconds</code>	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.59. `..spec.template.spec.containers[].livenessProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.60. .spec.template.spec.containers[].livenessProbe.grpc**描述**

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.61. .spec.template.spec.containers[].livenessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.62. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.63. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.64. .spec.template.spec.containers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.65. .spec.template.spec.containers[].ports

描述

要从容器公开的端口列表。不在此处指定端口 不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

数组

14.14.1.66. .spec.template.spec.containers[].ports[]

描述

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.14.1.67. .spec.template.spec.containers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.68. .spec.template.spec.containers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.69. .spec.template.spec.containers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.70. .spec.template.spec.containers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.71. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.72. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.73. .spec.template.spec.containers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.74. .spec.template.spec.containers[].resizePolicy

描述

容器的资源重新定义策略大小。

类型

数组

14.14.1.75. .spec.template.spec.containers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.14.1.76. .spec.template.spec.containers[].resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象 (数量)	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象 (数量)	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.14.1.77. .spec.template.spec.containers[].resources.claims

描述

声明列出了此容器使用的 `spec.resourceClaims` 中定义的资源名称。
这是一个 alpha 字段，需要启用 `DynamicResourceAllocation` 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.14.1.78. `.spec.template.spec.containers[].resources.claims[]`

描述

`ResourceClaim` 引用 `PodSpec.ResourceClaims` 中的一个条目。

类型

object

必填

- `name`

属性	类型	描述
<code>name</code>	string	<code>name</code> 必须与使用此字段的 Pod 的 <code>pod.spec.resourceClaims</code> 中一个条目的名称匹配。它使得资源在容器内可用。

14.14.1.79. `.spec.template.spec.containers[].securityContext`

描述

`securityContext` 包含将应用到容器的安全配置。`SecurityContext` 和 `PodSecurityContext` 中都存在一些字段。当同时设置这两个时，`SecurityContext` 中的值就会优先使用。

类型

object

属性	类型	描述
<code>allowPrivilegeEscalation</code>	布尔值	<code>allowPrivilegeEscalation</code> 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 <code>no_new_privs</code> 标志。当容器以 <code>Privileged 2</code> 运行行为 <code>Privileged 2</code> 时， <code>allowPrivilegeEscalation</code> 为 true 时，当 <code>spec.os.name</code> 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。

属性	类型	描述
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.14.1.80. .spec.template.spec.containers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.14.1.81. .spec.template.spec.containers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.14.1.82. .spec.template.spec.containers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。

属性	类型	描述
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值：- "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp。- "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。- "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.14.1.83. .spec.template.spec.containers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	<p>GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 <code>GMSACredentialSpecName</code> 字段命名。</p>
gmsaCredentialSpecName	字符串	<p>GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。</p>

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level, 仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外, 如果 HostProcess 为 true, 则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName, 以运行容器进程的入口点。如果未指定, 则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置, 则优先使用 SecurityContext 指定的值。

14.14.1.84. .spec.template.spec.containers[].startupProbe

描述

probe 描述了对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后, 探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.85. `..spec.template.spec.containers[].startupProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录（ <code>/</code> ）。命令只是 <code>exec'd</code> ，它不在 shell 中运行，因此传统的 shell 指令（ <code> </code> 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 <code>live/healthy</code> ，非零为不健康。

14.14.1.86. .spec.template.spec.containers[].startupProbe.grpc**描述**

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.87. .spec.template.spec.containers[].startupProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.88. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.89. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
----	----	----

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.90. .spec.template.spec.containers[].startupProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.91. .spec.template.spec.containers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.14.1.92. .spec.template.spec.containers[].volumeDevices[]

描述

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- **name**
- **devicePath**

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.14.1.93. .spec.template.spec.containers[].volumeMounts

描述

要挂载到容器文件系统 Pod 卷。无法更新。

类型

数组

14.14.1.94. .spec.template.spec.containers[].volumeMounts[]

描述

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- name
- mountPath

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 '..'。

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值： - "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。 - "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。 - "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.14.1.95. .spec.template.spec.dnsConfig

描述

PodDNSConfig 还定义了 pod 的 DNS 参数，除了从 DNSPolicy 生成的。

类型

object

属性	类型	描述
nameservers	数组（字符串）	DNS 名称服务器 IP 地址列表。这将附加到从 DNSPolicy 生成的基本名称服务器。将删除重复的名称服务器。
options	array	DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。
options[]	对象	PodDNSConfigOption 定义 pod 的 DNS 解析器选项。
搜索	数组（字符串）	host-name 查找的 DNS 搜索域列表。这将附加到 DNSPolicy 生成的基本搜索路径中。将删除重复的搜索路径。

14.14.1.96. .spec.template.spec.dnsConfig.options

描述

DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。将删除重复的条目。Options 中给出的解析选项将覆盖基本 DNSPolicy 中显示的解析选项。

类型

数组

14.14.1.97. .spec.template.spec.dnsConfig.options[]

描述

PodDNSConfigOption 定义 pod 的 DNS 解析器选项。

类型

object

属性	类型	描述
name	字符串	必需。
value	string	

14.14.1.98. .spec.template.spec.ephemeralContainers

描述

此 pod 中运行的临时容器列表。临时容器可以在现有 pod 中运行，以执行用户发起的操作，如调试。创建 pod 时无法指定此列表，无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod 中，请使用 pod 的 ephemeralcontainers 子资源。

类型**数组****14.14.1.99. .spec.template.spec.ephemeralContainers[]****描述**

EphemeralContainer 是一个临时容器，您可以为用户发起的活动（如调试）添加到现有 Pod 中。临时容器没有资源或调度保证，在 Pod 退出或重启时不会重启它们。如果临时容器导致 Pod 超过其资源分配，kubelet 可能会驱除 Pod。

要添加临时容器，请使用现有 Pod 的 ephemeralcontainers 子资源。可能无法删除或重启临时容器。

类型**object****必填**

- **name**

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用镜像的 CMD。变量引用 \$(VAR_NAME) 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
命令	数组 (字符串)	ENTRYPOINT 数组. 不会在 shell 中执行。如果没有提供, 则使用镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME) 使用容器环境扩展。如果无法解析变量, 输入字符串中的引用将不会改变。双引号被减少为一个 \$, 它允许转义 \$(VAR_NAME) 语法: 例如: "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展, 无论变量是否存在。无法更新。更多信息: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时, 所有无效的密钥都会被报告为事件。当多个源中存在键时, 与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息: https://kubernetes.io/docs/concepts/containers/images

属性	类型	描述
imagePullPolicy	字符串	<p>镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签, 则默认为 Always, 否则为 IfNotPresent。无法更新。更多信息 :</p> <p>https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>可能枚举值 : - "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败, 则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在, 且拉取失败, 则容器将失败。- "Never" 表示 kubelet 不会拉取镜像, 但只使用本地镜像。如果镜像不存在, 容器将失败</p>
生命周期	object	<p>生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序, 在操作完成前管理容器块, 除非容器进程失败, 在这种情况下, 处理程序会中止。</p>
livenessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
name	字符串	<p>指定为 DNS_LABEL 的临时容器的名称。此名称在所有容器、init 容器和临时容器中必须是唯一的。</p>
ports	array	<p>临时容器不允许使用端口。</p>
ports[]	对象	<p>containerPort 代表单一容器中的网络端口。</p>
readinessProbe	object	<p>probe 描述了要对容器执行的健康检查, 以确定它是否处于活动状态或准备好接收流量。</p>
resizePolicy	数组	<p>容器的资源重新定义策略大小。</p>
resizePolicy[]	object	<p>ContainerResizePolicy 代表容器的资源大小策略。</p>

属性	类型	描述
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
targetContainerName	字符串	<p>如果设置，则此临时容器目标的 PodSpec 中的容器名称。临时容器将在此容器的命名空间中运行 (IPC、PID 等)。如果没有设置，则临时容器将使用 Pod spec 中配置的命名空间。</p> <p>容器运行时必须实施对此功能的支持。如果运行时不支持以命名空间为目标，则设置此字段的结果未定义。</p>

属性	类型	描述
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。

属性	类型	描述
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.14.1.100. .spec.template.spec.ephemeralContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.14.1.101. .spec.template.spec.ephemeralContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。
value	字符串	变量引用 \$(VAR_NAME) 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 \$，它允许转义 \$(VAR_NAME) 语法：例如："(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会被扩展，无论变量是否存在。默认值为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的一个源。

14.14.1.102. .spec.template.spec.ephemeralContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 ConfigMap 中选择一个键。
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	SecretKeySelector 选择 Secret 的密钥。

14.14.1.103. .spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

描述

从 ConfigMap 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.14.1.104. .spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.14.1.105. .spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.14.1.106. .spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.14.1.107. .spec.template.spec.ephemeralContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.14.1.108. .spec.template.spec.ephemeralContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.14.1.109. .spec.template.spec.ephemeralContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.14.1.110. .spec.template.spec.ephemeralContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.14.1.111. .spec.template.spec.ephemeralContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

属性	类型	描述
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

14.14.1.112. .spec.template.spec.ephemeralContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPsocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPsocketAction 描述了基于打开套接字的操作

14.14.1.113. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.114. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.115. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeade

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.116. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeade

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**

- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.117. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.118. .spec.template.spec.ephemeralContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.14.1.119. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.120. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头

属性	类型	描述
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器中要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.121. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeader:

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.122. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeader

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.123. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.124. .spec.template.spec.ephemeralContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.125. .spec.template.spec.ephemeralContainers[].livenessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.126. .spec.template.spec.ephemeralContainers[].livenessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.127. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.128. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.129. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.130. .spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket

描述

TCPsocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.131. .spec.template.spec.ephemeralContainers[].ports

描述

临时容器不允许使用端口。

类型

数组

14.14.1.132. .spec.template.spec.ephemeralContainers[].ports[]

描述

containerPort 代表单一容器中的网络端口。

类型

object

必填

- containerPort

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.14.1.133. .spec.template.spec.ephemeralContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPsocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.134. .spec.template.spec.ephemeralContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.135. .spec.template.spec.ephemeralContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.136. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.137. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.138. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.139. .spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.140. .spec.template.spec.ephemeralContainers[].resizePolicy**描述**

容器的资源重新定义策略大小。

类型

数组

14.14.1.141. .spec.template.spec.ephemeralContainers[].resizePolicy[]**描述**

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.14.1.142. .spec.template.spec.ephemeralContainers[].resources**描述**

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.14.1.143. .spec.template.spec.ephemeralContainers[].resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.14.1.144. .spec.template.spec.ephemeralContainers[].resources.claims[]

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object**必填**

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.14.1.145. .spec.template.spec.ephemeralContainers[].securityContext**描述**

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型**object**

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.14.1.146. .spec.template.spec.ephemeralContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.14.1.147. .spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.14.1.148. .spec.template.spec.ephemeralContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- type

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.14.1.149. `.spec.template.spec.ephemeralContainers[].securityContext.windowsOptions`

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 <code>GMSACredentialSpecName</code> 字段命名。
gmsaCredentialSpecName	字符串	<code>GMSACredentialSpecName</code> 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	<code>HostProcess</code> 确定容器是否应该作为“主机进程”容器运行。此字段是 alpha-level，仅由启用 <code>WindowsHostProcessContainers</code> 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 <code>HostProcess</code> 值（不允许混合 <code>HostProcess</code> 容器和非主机 <code>Process</code> 容器）。另外，如果 <code>HostProcess</code> 为 <code>true</code> ，则 <code>HostNetwork</code> 还必须设置为 <code>true</code> 。
runAsUserName	字符串	Windows 中的 <code>UserName</code> ，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 <code>PodSecurityContext</code> 中设置。如果在 <code>SecurityContext</code> 和 <code>PodSecurityContext</code> 中设置，则优先使用 <code>SecurityContext</code> 指定的值。

14.14.1.150. `.spec.template.spec.ephemeralContainers[].startupProbe`

描述

`probe` 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为1秒。最小值为1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.151. .spec.template.spec.ephemeralContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.152. .spec.template.spec.ephemeralContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- port

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.14.1.153. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.14.1.154. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.155. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.156. .spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.157. .spec.template.spec.ephemeralContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.14.1.158. .spec.template.spec.ephemeralContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- name
- devicePath

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.14.1.159. .spec.template.spec.ephemeralContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。临时容器不允许进行 subPath 挂载。无法更新。

类型

数组

14.14.1.160. .spec.template.spec.ephemeralContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- name
- mountPath

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 ':'。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值： - "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。 - "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。 - "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.14.1.161. .spec.template.spec.hostAliases

描述

hostAliases 是主机和 IP 的可选列表，如果指定，将注入到 pod 的主机文件中。这只适用于非主机网络 pod。

类型

数组

14.14.1.162. `.spec.template.spec.hostAliases[]`

描述

HostAlias 包含将作为 pod 主机文件中的条目注入的 IP 和主机名之间的映射。

类型

object

属性	类型	描述
主机名	数组（字符串）	以上 IP 地址的主机名。
ip	字符串	主机文件条目的 IP 地址。

14.14.1.163. `.spec.template.spec.imagePullSecrets`

描述

imagePullSecrets 是同一命名空间中 secret 的引用列表，用于拉取此 PodSpec 使用的任何镜像。如果指定，这些 secret 将传递给单独的 puller 实现，供它们使用。更多信息：

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

类型

数组

14.14.1.164. `.spec.template.spec.imagePullSecrets[]`

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.165. `.spec.template.spec.initContainers`

描述

属于 pod 的初始化容器列表。init 容器在容器启动前先执行。如果任何 init 容器失败，pod 被视为失败，并根据其 restartPolicy 处理。init 容器或普通容器的名称必须在所有容器间唯一。Init 容器可能没有生命周期操作、就绪度探测、存活度探测或启动探测。init 容器的 resourceRequirements 在调度过程中考虑，方法是每个资源类型找到最高 request/limit，然后使用该值或普通容器的总和。限制以类似的方式应用到 init 容器。目前无法添加或删除 init 容器。无法更新。更多信息：

<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

类型

数组

14.14.1.166. .spec.template.spec.initContainers[]

描述

要在 pod 中运行的单一应用程序容器。

类型

object

必填

- name

属性	类型	描述
args	数组（字符串）	入口点的参数。如果没有提供，则使用容器镜像的 CMD。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
命令	数组（字符串）	ENTRYPOINT 数组。不会在 shell 中执行。如果没有提供，则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器环境扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。无法更新。更多信息： https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

属性	类型	描述
env	array	要在容器中设置的环境变量列表。无法更新。
env[]	对象	EnvVar 代表容器中存在的环境变量。
envFrom	array	用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。
envFrom[]	对象	EnvFromSource 代表一组 ConfigMap 的源
image	string	容器镜像名称。更多信息： https://kubernetes.io/docs/concepts/containers/images 此字段是可选的，允许更高级别的配置管理默认或覆盖工作负载控制器中的容器镜像，如 Deployment 和 StatefulSets。
imagePullPolicy	字符串	镜像拉取策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则为 IfNotPresent。无法更新。更多信息： https://kubernetes.io/docs/concepts/containers/images#updating-images 可能枚举值：- "Always" 表示 kubelet 始终会尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示 kubelet 在磁盘上不存在时拉取。如果镜像不存在，且拉取失败，则容器将失败。- "Never" 表示 kubelet 不会拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败

属性	类型	描述
生命周期	object	生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。
livenessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
name	字符串	指定为 DNS_LABEL 的容器的名称。pod 中的每个容器都必须具有唯一名称(DNS_LABEL)。无法更新。
ports	array	要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 https://github.com/kubernetes/kubernetes/issues/108255 。无法更新。
ports[]	对象	containerPort 代表单一容器中的网络端口。
readinessProbe	object	probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
resizePolicy	数组	容器的资源重新定义策略大小。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源大小策略。
resources	对象	ResourceRequirements 描述计算资源要求。
securityContext	object	securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

属性	类型	描述
startupProbe	object	probe 描述了要对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。
stdin	布尔值	此容器是否应该为容器运行时中的 stdin 分配缓冲区。如果没有设置，则从容器中的 stdin 读取总是会导致 EOF。默认值为 false。
stdinOnce	布尔值	容器运行时是否应该在单个附加打开后关闭 stdin 频道。当 stdin 为 true 时，stdin 流将在多个附加会话之间保持打开。如果将 stdinOnce 设为 true，则 stdin 在容器启动时打开，会为空，直到第一个客户端附加到 stdin，然后保持打开并接受数据，直到客户端断开连接，而时间 stdin 被关闭并保持关闭，直到容器重启为止。如果此标志为 false，则从 stdin 读取的容器进程永远不会收到 EOF。默认值为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件路径被挂载到容器文件系统中。编写的消息应该是简短的最终状态，如断言失败消息。如果大于 4096 字节，则节点将被截断。所有容器的消息长度限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功和失败时填充容器状态消息。如果终止消息文件为空，并且容器退出并显示错误，则 FallbackToLogsOnError 将使用容器日志输出的最后块。日志输出限制为 2048 字节或 80 行，以较大者为准。默认为 file。无法更新。 可能的 enum 值： - "FallbackToLogsOnError" 会在容器退出时读取容器状态消息的最新内容，且 terminationMessagePath 没有内容。 - "File" 是默认行为，并将容器状态消息设置为容器终止 MessagePath 的内容。

属性	类型	描述
tty	布尔值	此容器是否应为自己分配 TTY，还需要 'stdin' 为 true。默认值为 false。
volumeDevices	array	volumeDevices 是容器使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了在容器中挂载卷。
workingDir	字符串	容器的工作目录。如果没有指定，则使用容器运行时的默认值，这可能会在容器镜像中配置。无法更新。

14.14.1.167. .spec.template.spec.initContainers[].env

描述

要在容器中设置的环境变量列表。无法更新。

类型

数组

14.14.1.168. .spec.template.spec.initContainers[].env[]

描述

EnvVar 代表容器中存在的环境变量。

类型

object

必填

- name

属性	类型	描述
name	字符串	环境变量的名称。必须是 C_IDENTIFIER。

属性	类型	描述
value	字符串	变量引用 <code>\$(VAR_NAME)</code> 使用容器中之前定义的环境变量和任何服务环境变量进行扩展。如果无法解析变量，输入字符串中的引用将不会改变。双引号被减少为一个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：例如： <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>"\$(VAR_NAME)"</code> 。转义的引用永远不会被扩展，无论变量是否存在。默认值为 <code>""</code> 。
valueFrom	object	<code>EnvVarSource</code> 代表 <code>EnvVarar</code> 的值的一个源。

14.14.1.169. `.spec.template.spec.initContainers[].env[].valueFrom`

描述

`EnvVarSource` 代表 `EnvVarar` 的值的一个源。

类型

object

属性	类型	描述
configMapKeyRef	object	从 <code>ConfigMap</code> 中选择一个键。
fieldRef	object	<code>ObjectFieldSelector</code> 选择对象的 <code>APIVersioned</code> 字段。
resourceFieldRef	object	<code>ResourceFieldSelector</code> 代表容器资源(cpu、memory)及其输出格式
secretKeyRef	object	<code>SecretKeySelector</code> 选择 <code>Secret</code> 的密钥。

14.14.1.170. `.spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef`

描述

从 `ConfigMap` 中选择一个键。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要选择的密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap 还是其键

14.14.1.171. .spec.template.spec.initContainers[].env[].valueFrom.fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- **fieldPath**

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.14.1.172. .spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的

属性	类型	描述
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.14.1.173. .spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef

描述

SecretKeySelector 选择 Secret 的密钥。

类型

object

必填

- **key**

属性	类型	描述
key	字符串	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret 或其密钥

14.14.1.174. .spec.template.spec.initContainers[].envFrom

描述

用于在容器中填充环境变量的源列表。源中定义的键必须是 C_IDENTIFIER。当容器启动时，所有无效的密钥都会被报告为事件。当多个源中存在键时，与最后一个源关联的值将具有优先权。由带有重复键的 Env 定义的值将具有优先权。无法更新。

类型

数组

14.14.1.175. .spec.template.spec.initContainers[].envFrom[]

描述

EnvFromSource 代表一组 ConfigMap 的源

类型

object

属性	类型	描述
configMapRef	object	ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。 目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。
prefix	字符串	用于添加到 ConfigMap 中每个键的可选标识符。必须是 C_IDENTIFIER。
secretRef	object	SecretEnvSource 选择一个 Secret 来填充环境变量。 目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

14.14.1.176. .spec.template.spec.initContainers[].envFrom[].configMapRef

描述

ConfigMapEnvSource 选择一个 ConfigMap 来填充环境变量。
目标 ConfigMap 的 Data 字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否必须定义 ConfigMap

14.14.1.177. .spec.template.spec.initContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的数据字段的内容将以环境变量的形式表示键值对。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	指定是否需要定义 Secret

14.14.1.178. .spec.template.spec.initContainers[].lifecycle

描述

生命周期描述了管理系统在响应容器生命周期事件时应采取的操作。对于 PostStart 和 PreStop 生命周期处理程序，在操作完成前管理容器块，除非容器进程失败，在这种情况下，处理程序会中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。
preStop	object	LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

14.14.1.179. .spec.template.spec.initContainers[].lifecycle.postStart

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
HttpGet	object	HttpGetAction 描述了基于 HTTP 获取请求的操作。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.14.1.180. .spec.template.spec.initContainers[].lifecycle.postStart.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.181. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.182. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.183. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.184. .spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.185. .spec.template.spec.initContainers[].lifecycle.preStop

描述

LifecycleHandler 定义在生命周期 hook 中应执行的特定操作。必须指定其中一个字段，但必须指定 TCPSocket。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

14.14.1.186. .spec.template.spec.initContainers[].lifecycle.preStop.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统根目录的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令('!' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.187. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.188. `.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders`

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.189. `.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]`

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.190. `.spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket`

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.191. .spec.template.spec.initContainers[].livenessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

属性	类型	描述
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 <code>terminationGracePeriodSeconds</code> 。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 <code>ProbeTerminationGracePeriod</code> 功能门。如果未设置，则使用最小值为 1。 <code>spec.terminationGracePeriodSeconds</code> 。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.192. `.spec.template.spec.initContainers[].livenessProbe.exec`

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令 ('' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.193. `.spec.template.spec.initContainers[].livenessProbe.grpc`

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.194. .spec.template.spec.initContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.195. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.196. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.197. .spec.template.spec.initContainers[].livenessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object**必填**

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.198. .spec.template.spec.initContainers[].ports**描述**

要从容器公开的端口列表。不在此处指定端口不会阻止公开该端口。任何正在侦听容器内的默认"0.0.0.0"地址的端口都可以从网络访问。使用战略合并补丁修改此阵列可能会破坏数据。如需更多信息，请参阅 <https://github.com/kubernetes/kubernetes/issues/108255>。无法更新。

类型

数组

14.14.1.199. .spec.template.spec.initContainers[].ports[]**描述**

containerPort 代表单一容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	在 pod 的 IP 地址上公开的端口号。这必须是有效的端口号 $0 < x < 65536$ 。
hostIP	字符串	将外部端口绑定到的主机 IP。
hostPort	整数	要在主机上公开的端口号。如果指定，这必须是有效的端口号， $0 < x < 65536$ 。如果指定了 HostNetwork，则必须匹配 ContainerPort。大多数容器都不需要此操作。

属性	类型	描述
name	字符串	如果指定，这必须是 IANA_SVC_NAME，并在 pod 中唯一。pod 中的每个指定端口都必须具有唯一的名称。服务可以引用的端口的名称。
protocol	字符串	端口的协议。必须是 UDP、TCP 或 SCTP。默认为 "TCP"。 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.14.1.200. .spec.template.spec.initContainers[].readinessProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。

属性	类型	描述
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.201. .spec.template.spec.initContainers[].readinessProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root（/）。命令只是 exec'd，它不在 shell 中运行，因此传统的 shell 指令（ 等）不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零为不健康。

14.14.1.202. .spec.template.spec.initContainers[].readinessProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。
service	string	service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md ）。 如果没有指定，则默认的行为由 gRPC 定义。

14.14.1.203. .spec.template.spec.initContainers[].readinessProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。

属性	类型	描述
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	用于连接到主机的方案。默认为 HTTP。 可能枚举的值：- "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://

14.14.1.204. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.205. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.206. .spec.template.spec.initContainers[].readinessProbe.tcpSocket

描述

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.207. .spec.template.spec.initContainers[].resizePolicy**描述**

容器的资源重新定义策略大小。

类型

数组

14.14.1.208. .spec.template.spec.initContainers[].resizePolicy[]**描述**

ContainerResizePolicy 代表容器的资源大小策略。

类型

object

必填

- **resourceName**
- **restartPolicy**

属性	类型	描述
resourceName	string	应用此资源调整大小策略的资源名称。支持的值：cpu、memory。
restartPolicy	string	在调整指定资源大小时，重启策略会应用。如果没有指定，则默认为 NotRequired。

14.14.1.209. .spec.template.spec.initContainers[].resources**描述**

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.14.1.210. .spec.template.spec.initContainers[].resources.claims

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.14.1.211. .spec.template.spec.initContainers[].resources.claims[]

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object**必填**

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.14.1.212. .spec.template.spec.initContainers[].securityContext**描述**

securityContext 包含将应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当同时设置这两个时，SecurityContext 中的值就会优先使用。

类型**object**

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以获得比父进程更多的特权。此 bool 直接控制容器进程中是否设置了 no_new_privs 标志。当容器以 Privileged 2 运行 Privileged 2 时，allowPrivilegeEscalation 为 true 时，当 spec.os.name 为窗口时，无法设置此字段。
功能	object	从正在运行的容器中添加和删除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程实质上相当于主机上的 root 用户。默认为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。

属性	类型	描述
procMount	字符串	<p>procMount 表示用于容器的 proc 挂载类型。默认为 DefaultProcMount，它对只读路径和屏蔽的路径使用容器运行时默认值。这需要启用 ProcMountType 功能标记。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值：- "Default" 将容器运行时默认值用于 /proc 的 readonly 和 masked 路径。大多数容器运行时屏蔽 /proc 中的某些路径，以避免意外暴露特殊设备或信息。- "Unmasked" 绕过容器运行时的默认屏蔽行为，并确保新创建的 /proc 容器在没有修改的情况下保持不变。</p>
readOnlyRootFilesystem	布尔值	<p>此容器是否具有只读根文件系统。默认值为 false。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsGroup	整数	<p>运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
runAsNonRoot	布尔值	<p>表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。</p>

属性	类型	描述
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。请注意，当 spec.os.name 是 windows 时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.14.1.213. .spec.template.spec.initContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和删除 POSIX 功能。

类型

object

属性	类型	描述
add	数组（字符串）	添加了功能
drop	数组（字符串）	删除的功能

14.14.1.214. .spec.template.spec.initContainers[].securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。

属性	类型	描述
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.14.1.215. .spec.template.spec.initContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- **type**

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	<p>type 表示将应用了哪些 seccomp 配置集。有效选项有：</p> <p>localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。</p> <p>可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。</p>

14.14.1.216. .spec.template.spec.initContainers[].securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa) 内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.14.1.217. .spec.template.spec.initContainers[].startupProbe

描述

probe 描述了对容器执行的健康检查，以确定它是否处于活动状态或准备好接收流量。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。
failureThreshold	整数	在成功后，探测被视为失败的连续最小失败。默认值为 3。最小值为 1。
grpc	object	GRPC 指定涉及 GRPC 端口的操作。
httpGet	object	HTTPGetAction 描述了基于 HTTP 获取请求的操作。
initialDelaySeconds	整数	容器启动后的秒数，然后再启动存活度探测。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	整数	执行探测的频率（以秒为单位）。默认值为 10 秒。最小值为 1。
successThreshold	整数	在失败后，探测被视为成功的最低连续成功。默认为 1。对于存活度和启动，必须为 1。最小值为 1。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作
terminationGracePeriodSeconds	整数	可选持续时间（以秒为单位），pod 需要在探测失败时安全终止。宽限期是 pod 中运行的进程发送终止信号的持续时间（以秒为单位），以及进程通过 kill 信号强制停止的时间。设置这个值比您的进程预期的清理时间长。如果这个值为 nil，则使用 pod 的 terminationGracePeriodSeconds。否则，这个值会覆盖 pod 规格提供的值。值必须是非负整数。值零表示通过 kill 信号立即停止（没有关闭的机会）。这是一个 beta 字段，需要启用 ProbeTerminationGracePeriod 功能门。如果未设置，则使用最小值为 1。 spec.terminationGracePeriodSeconds。

属性	类型	描述
timeoutSeconds	整数	探测超时的秒数。默认值为 1 秒。最小值为 1。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.14.1.218. .spec.template.spec.initContainers[].startupProbe.exec

描述

ExecAction 描述了 "run in container" 操作。

类型

object

属性	类型	描述
命令	数组（字符串）	命令是在容器内执行的命令行，命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不在 shell 中运行，因此传统的 shell 指令(' ' 等)不起作用。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy, 非零为不健康。

14.14.1.219. .spec.template.spec.initContainers[].startupProbe.grpc

描述

GRPC 指定涉及 GRPC 端口的操作。

类型

object

必填

- **port**

属性	类型	描述
port	整数	gRPC 服务的端口号。数字必须在 1 到 65535 之间。

属性	类型	描述
service	string	<p>service 是要放在 gRPC HealthCheckRequest 中的服务名称（请参阅 https://github.com/grpc/grpc/blob/master/doc/health-checking.md）。</p> <p>如果没有指定，则默认的行为由 gRPC 定义。</p>

14.14.1.220. .spec.template.spec.initContainers[].startupProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP 获取请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接的主机名，默认为 pod IP。您可能想在 httpHeaders 中设置"主机"。
httpHeaders	array	在请求中设置的自定义标头。HTTP 允许重复标头。
httpHeaders[]	对象	HTTPHeader 描述了 HTTP 探测中使用的自定义标头
path	字符串	在 HTTP 服务器上访问的路径。
port	IntOrString	在容器上要访问的端口的名称或数量。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。
scheme	字符串	<p>用于连接到主机的方案。默认为 HTTP。</p> <p>可能枚举的值： - "HTTP" 表示使用的方案将是 http:// - "HTTPS" 意味着所使用的方案将是 https://</p>

14.14.1.221. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复标头。

类型

数组

14.14.1.222. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了 HTTP 探测中使用的自定义标头

类型

object

必填

- name
- value

属性	类型	描述
name	string	标头字段名称。这将在输出时规范，因此案例名称将理解为相同的标头。
value	字符串	标头字段值

14.14.1.223. .spec.template.spec.initContainers[].startupProbe.tcpSocket**描述**

TCPSocketAction 描述了基于打开套接字的操作

类型

object

必填

- port

属性	类型	描述
主机	字符串	可选：要连接到的主机名，默认为 pod IP。
port	IntOrString	在容器中访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

14.14.1.224. .spec.template.spec.initContainers[].volumeDevices

描述

volumeDevices 是容器使用的块设备列表。

类型

数组

14.14.1.225. .spec.template.spec.initContainers[].volumeDevices[]**描述**

volumeDevice 描述容器中原始块设备的映射。

类型

object

必填

- name
- devicePath

属性	类型	描述
devicePath	字符串	DevicePath 是设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中的 persistentVolumeClaim 的名称匹配

14.14.1.226. .spec.template.spec.initContainers[].volumeMounts**描述**

要挂载到容器文件系统 Pod 卷。无法更新。

类型

数组

14.14.1.227. .spec.template.spec.initContainers[].volumeMounts[]**描述**

VolumeMount 描述了在容器中挂载卷。

类型

object

必填

- name
- mountPath

属性	类型	描述
mountPath	字符串	在容器中挂载卷的路径。不得包含 `.`。
mountPropagation	字符串	<p>mountPropagation 决定如何将挂载从主机传播到容器，以及相关的其他方法。如果没有设置，则使用 MountPropagationNone。此字段是 1.10 中的 beta。</p> <p>可能枚举值： - "Bidirectional" 表示容器中的卷将从主机或其他容器接收新挂载，并且其自身的挂载将从容器传播到主机或其他容器。请注意，这个模式会递归应用到 Linux 术语中的卷中的所有挂载。 - "HostToContainer" 表示容器中的卷将从主机或其他容器中接收新挂载，但挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式会递归应用到卷中的所有挂载("rslave")。 - "None" 表示容器中的卷不会从主机或其他容器接收新挂载，以及挂载到容器中的文件系统不会传播到主机或其他容器。请注意，这个模式与 Linux 术语中的"private"对应。</p>
name	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读形式挂载（否则为 读写(false 或未指定)。默认为 false。
subPath	字符串	应从中挂载容器卷的卷中的路径。默认为 ""（卷的根目录）。
subPathExpr	字符串	应从中挂载容器卷的卷中扩展路径。行为与 SubPath 相似，但环境变量引用 \$(VAR_NAME)使用容器的环境进行扩展。默认为 ""（卷的根目录）。SubPathExpr 和 SubPath 是互斥的。

14.14.1.228. .spec.template.spec.os

描述

PodOS 定义 pod 的 OS 参数。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是操作系统的名称。目前支持的值有 linux 和 windows。额外值可能会在以后定义，可以是： https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration 客户端应该处理附加值，并将此字段中的未识别的值视为 os: null

14.14.1.229. .spec.template.spec.readinessGates**描述**

如果指定，则会针对 pod 就绪度评估所有就绪度。当它的所有容器都就绪且在就绪度授权中指定的所有条件都等于"True"更多信息时，pod 已就绪：<https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

类型

数组

14.14.1.230. .spec.template.spec.readinessGates[]**描述**

PodReadinessGate 包含对 pod 条件的引用

类型

object

必填

- **conditionType**

属性	类型	描述
conditionType	字符串	conditionType 指的是 pod 条件列表中具有匹配类型的条件。

14.14.1.231. .spec.template.spec.resourceClaims**描述**

ResourceClaims 定义在允许 Pod 启动前必须分配和保留哪些 ResourceClaims。资源将提供给那些按名称消耗它们的容器。

这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。

类型

数组

14.14.1.232. .spec.template.spec.resourceClaims[]

描述

PodResourceClaim 通过 ClaimSource 准确引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称引用它。

类型

object

必填

- name

属性	类型	描述
name	string	在 pod 中唯一标识此资源声明的名称。这必须是 DNS_LABEL。
source	object	ClaimSource 描述了对 ResourceClaim 的引用。 应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

14.14.1.233. .spec.template.spec.resourceClaims[].source

描述

ClaimSource 描述了对 ResourceClaim 的引用。
应设置其中一个字段。这个类型的消费者必须将空对象视为一个未知值。

类型

object

属性	类型	描述
resourceClaimName	string	ResourceClaimName 是与这个 pod 相同的命名空间中的 ResourceClaim 对象的名称。

属性	类型	描述
resourceClaimTemplateName	string	<p>ResourceClaimTemplateName 是与这个 pod 相同的命名空间中的 ResourceClaimTemplate 对象的名称。</p> <p>模板将用于创建新的 ResourceClaim，它将绑定到此 pod。删除此 pod 后，ResourceClaim 也会被删除。ResourceClaim 的名称将是 <pod name>-<resource name>，其中 <resource name> 是 PodResourceClaim.Name。如果串联名称对 ResourceClaim 无效（如太长，则 Pod 验证将拒绝 pod）。</p> <p>没有为 pod 拥有该名称的现有 ResourceClaim 不会被用于 pod，以避免错误地使用不相关的资源。然后，调度和 pod 启动会被阻断，直到删除不相关的 ResourceClaim。</p> <p>此字段不可变，在创建 ResourceClaim 后，control plane 不会对对应的 ResourceClaim 进行更改。</p>

14.14.1.234. .spec.template.spec.schedulingGates

描述

SchedulingGates 是值（如果指定）将阻止调度 pod 的不透明列表。如果 schedulingGates 不是空的，pod 将保持在 SchedulingGated 状态，调度程序也不会尝试调度 pod。SchedulingGates 只能在创建 pod 时设置，之后只能删除。

这是 PodSchedulingReadiness 功能门启用的 beta 功能。

类型

数组

14.14.1.235. .spec.template.spec.schedulingGates[]

描述

PodSchedulingGate 与 Pod 关联，以保护其调度。

类型

object

必填

- **name**

属性	类型	描述
name	string	调度授权的名称。每个调度最低要求都必须具有唯一的 name 字段。

14.14.1.236. .spec.template.spec.securityContext

描述

PodSecurityContext 包含 pod 级别的安全属性和通用容器设置。container.securityContext 中也存在一些字段。container.securityContext 的字段值优先于 PodSecurityContext 的字段值。

类型

object

属性	类型	描述
fsGroup	整数	<p>适用于 pod 中所有容器的特殊补充组。有些卷类型允许 Kubelet 将该卷的所有权更改为 pod 拥有：</p> <ol style="list-style-type: none"> 1.所属 GID 将是 FSGroup 2. set setgid 位（在卷中创建的新文件将归 FSGroup 所有） 3. 权限位为 OR'd with rw-rw---- <p>如果未设置，Kubelet 将不修改任何卷的所有权和权限。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p>
fsGroupChangePolicy	字符串	<p>fsGroupChangePolicy 定义在 Pod 中公开卷之前更改卷的所有权和权限的行为。此字段仅适用于支持 fsGroup 基于所有权（和权限）的卷类型。它对临时卷类型没有影响，如：secret、configmaps 和 emptydir。有效值为 "OnRootMismatch" 和 "Always"。如果没有指定，则使用 "Always"。请注意，当 spec.os.name 是窗口时，无法设置此字段。</p> <p>可能枚举值： - "Always" 表示当卷挂载到 Pod 中时，应始终更改卷的所有权和权限。这个默认行为。 - "OnRootMismatch" 表示只有在根目录的权限与卷上预期权限不匹配时才会更改卷的所有权和权限。这有助于缩短更改卷的所有权和权限所需的时间。</p>

属性	类型	描述
runAsGroup	整数	运行容器进程的入口点的 GID。如果未设置，则使用运行时默认值。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
runAsNonRoot	布尔值	表示容器必须以非 root 用户身份运行。如果为 true，Kubelet 将在运行时验证镜像，以确保它不会以 UID 0 (root) 运行，如果容器确实无法启动它，则无法启动它。如果未设置或 false，则不会执行这样的验证。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。
runAsUser	整数	运行容器进程的入口点的 UID。如果未指定，则默认为镜像元数据中指定的用户。也可以在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则在 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是窗口时，无法设置此字段。
seLinuxOptions	object	seLinuxOptions 是要应用到容器的标签
seccompProfile	object	SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

属性	类型	描述
supplementalGroups	数组（整数）	除了容器的主 GID、fsGroup（如果指定）以及容器镜像中定义的组成员资格（如果指定）之外，应用于每个容器中的第一个进程运行的组列表。如果未指定，则不会将额外的组添加到任何容器中。请注意，容器进程的 uid 中定义的组成员资格仍然有效，即使它们没有包括在此列表中。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls	array	sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。
sysctls[]	对象	sysctl 定义要设置的内核参数
windowsOptions	object	WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

14.14.1.237. .spec.template.spec.securityContext.seLinuxOptions

描述

seLinuxOptions 是要应用到容器的标签

类型

object

属性	类型	描述
level	字符串	level 是适用于容器的 SELinux 级别标签。
role	字符串	role 是适用于容器的 SELinux 角色标签。
type	字符串	type 是适用于容器的 SELinux 类型标签。
user	字符串	user 是适用于容器的 SELinux 用户标签。

14.14.1.238. .spec.template.spec.securityContext.seccompProfile

描述

SeccompProfile 定义 pod/container 的 seccomp 配置集设置。只能设置一个配置文件源。

类型

object

必填

- type

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示使用节点上文件中定义的配置集。该配置集必须在节点上预先配置才能正常工作。必须是相对于 kubelet 配置的 seccomp 配置集位置的降序路径。只有当 type 为 "Localhost" 时，才必须设置。
type	字符串	type 表示将应用了哪些 seccomp 配置集。有效选项有： localhost - 应该使用节点上文件中定义的配置集。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用任何配置集。 可能枚举值： - "Localhost" 表示应使用节点上文件中定义的配置集。文件相对于 <kubelet-root-dir>/seccomp. - "RuntimeDefault" 代表默认的容器运行时 seccomp 配置集。 - "Unconfined" 表示没有应用 seccomp 配置集(A.K.A. unconfined)。

14.14.1.239. .spec.template.spec.securityContext.sysctls

描述

sysctl 包含用于 pod 的命名空间 sysctl 的列表。带有不支持的 sysctl（由容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是窗口时，无法设置此字段。

类型

数组

14.14.1.240. .spec.template.spec.securityContext.sysctls[]

描述

sysctl 定义要设置的内核参数

类型

object

必填

- **name**
- **value**

属性	类型	描述
name	字符串	要设置的属性的名称
value	字符串	要设置的属性值

14.14.1.241. .spec.template.spec.securityContext.windowsOptions

描述

WindowsSecurityContextOptions 包含特定于 Windows 的选项和凭证。

类型

object

属性	类型	描述
gmsaCredentialSpec	字符串	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联 GMSA 凭证规格的内容，由 GMSACredentialSpecName 字段命名。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证 spec 的名称。
hostProcess	布尔值	HostProcess 确定容器是否应该作为"主机进程"容器运行。此字段是 alpha-level，仅由启用 WindowsHostProcessContainers 功能标记的组件所接受。在没有功能标记的情况下设置此字段将导致验证 Pod 时出现错误。Pod 的容器必须具有相同的有效 HostProcess 值（不允许混合 HostProcess 容器和非主机Process 容器）。另外，如果 HostProcess 为 true，则 HostNetwork 还必须设置为 true。

属性	类型	描述
runAsUserName	字符串	Windows 中的 UserName，以运行容器进程的入口点。如果未指定，则默认为镜像元数据中指定的用户。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则优先使用 SecurityContext 指定的值。

14.14.1.242. .spec.template.spec.tolerations

描述

如果指定，pod 的容限。

类型

数组

14.14.1.243. .spec.template.spec.tolerations[]

描述

附加到此 Tolerations 的 pod，以容许任何与 triple <key,value,effect> 匹配的污点，使用匹配的 operator <operator>。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
effect	字符串	<p>effect 表示污点效果要匹配。空意味着匹配所有污点效果。指定后，允许的值为 NoSchedule、PreferNoSchedule 和 NoExecute。</p> <p>可能枚举值： - "NoExecute" Evict 任何已在运行的 pod，这些 pod 不容许该污点。目前由 NodeController 实施。 - "NoSchedule" 不允许新 pod 调度到该节点上，除非它们容许污点，但允许提交给 Kubelet 的所有 pod 而无需启动调度程序，并允许所有已在运行的 pod 继续运行。调度程序强制使用 - "PreferNoSchedule" Like TaintEffectNoSchedule，但调度程序会尝试不将新 pod 调度到节点上，而不是禁止新 pod 完全调度到该节点上。由调度程序实施。</p>
key	字符串	<p>key 是容限应用到的污点键。empty 表示与所有污点键匹配。如果键为空，则运算符必须是 Exists；此组合意味着匹配所有值和所有键。</p>
operator	字符串	<p>Operator 代表键与值的关系。有效的运算符是 Exists 和 Equal。默认为 Equal。exists 等同于 value 的通配符，以便 pod 可以容忍特定类别的所有污点。</p> <p>可能枚举值： - "Equal" - "Exists"</p>
tolerationSeconds	整数	<p>TolerationSeconds 代表容限的期间（该容限必须是 NoExecute，否则将忽略此字段）容许污点。默认情况下，它没有被设置，这意味着容许永久污点（不驱除）。系统会将零值和负值视为 0（立即撤离）。</p>
value	字符串	<p>值是容限匹配的污点值。如果运算符是 Exists，则该值应为空，否则只是一个常规字符串。</p>

14.14.1.244. .spec.template.spec.topologySpreadConstraints

描述

topologySpreadConstraints 描述如何在拓扑域中分散的 pod 组。调度程序将通过约束来调度 pod。所有 topologySpreadConstraints 都是 AND。

类型

数组

14.14.1.245. .spec.template.spec.topologySpreadConstraints[]**描述**

TopologySpreadConstraint 指定如何在给定的拓扑中分散匹配的 pod。

类型

object

必填

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

属性	类型	描述
labelSelector	labelSelector	labelSelector 用于查找匹配的 pod。与此标签选择器匹配的 Pod 被计算，以确定其对应拓扑域中的 pod 数量。
matchLabelKeys	数组（字符串）	matchLabelKeys 是一组 pod 标签键，用于选择要计算分布的 pod。键用于从传入的 pod 标签中查找值，这些键值标签带有 labelSelector 的 ANDed 来选择要针对传入 pod 计算的现有 pod 组。禁止在 MatchLabelKeys 和 LabelSelector 中存在相同的键。当 LabelSelector 没有设置时，无法设置 matchLabelKeys。传入的 pod 标签中不存在的键将被忽略。null 或 empty 列表表示仅匹配 labelSelector。 这是一个 beta 字段，需要启用 MatchLabelKeysInPodTopologySpread 功能门（默认启用）。

属性	类型	描述
maxSkew	整数	<p>maxSkew 描述了 pod 可能不均匀分布的程度。当 whenUnsatisfiable=DoNotSchedule 时，目标拓扑中匹配 pod 数量和全局最小值之间允许的最大区别。如果有资格的域数量小于 MinDomains，则全局最小值是符合域或零的匹配 pod 的最小数量。例如，在 3 区集群中，MaxSkew 设置为 1，具有相同 labelSelector 的 pod 分布为 2/2/1：在这种情况下，全局最小值为 1。 zone1 zone2 zone3 P P P P P P P P P - 如果 MaxSkew 为 1，则传入的 pod 只能调度到 zone3，以变为 2/2/2；将其调度到 zone1 (zone2)上的 ActualSkew (3-1)违反 MaxSkew (1)- if MaxSkew.当 whenUnsatisfiable=ScheduleAnyway 时，用来为满足它的拓扑提供更高优先级。它是必填字段。默认值为 1，不允许使用 0。</p>

属性	类型	描述
minDomains	整数	<p>MinDomains 表示最少的合格域数。当匹配拓扑键的合格域数量小于 minDomains 时，Pod Topology Spread 会将"global minimum"视为 0，然后执行 Skew 计算。当匹配拓扑键的合格域数量等于或大于 minDomains 时，这个值对调度没有影响。因此，当有资格的域数量小于 minDomains 时，调度程序不会将这些域的 maxSkew Pod 调度到这些域。如果值为 nil，则约束的行为就像 MinDomains 等于 1。有效值为大于 0 的整数。当值不是 nil 时，WhenUnsatisfiable 必须是 DoNotSchedule。</p> <p>例如，在 3 区集群中，MaxSkew 设置为 2，MinDomains 被设置为 5，而带有与 2/2/2: zone1 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone2 zone3 zone3 zone3 区域数量小于 5 (MinDomains)，因此"global minimum"被视为 0。在这种情况下，无法调度具有相同 labelSelector 的新 pod，因为如果新 Pod 调度到三个区，则 computed skew 将为 3 (3 - 0)，它将违反 MaxSkew。</p> <p>这是一个 beta 字段，需要启用 MinDomainsInPodTopologySpread 功能门（默认启用）。</p>

属性	类型	描述
nodeAffinityPolicy	string	<p>NodeAffinityPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理 Pod 的 nodeAffinity/nodeSelector。选项为：- Honor：计算中仅包含与 nodeAffinity/nodeSelector 匹配的节点。忽略 Ignore: nodeAffinity/nodeSelector。所有节点包含在计算中。</p> <p>如果这个值为 nil，则行为等同于 Honor 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>
nodeTaintsPolicy	string	<p>NodeTaintsPolicy 表示在计算 pod 拓扑分布偏移时，我们将如何处理节点污点。选项为：- Honor: 没有污点的节点，以及传入 pod 具有容量的污点节点被包含。- Ignore: node taint is ignored。包括所有节点。</p> <p>如果这个值为 nil，则行为等同于 Ignore 策略。这是一个 beta 级别的功能，由 NodeInclusionPolicyInPodTopologySpread 功能标记启用。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布 skew 时使用此调度指令。- "Ignore" 表示在计算 pod 拓扑分布 skew 时忽略此调度指令。</p>

属性	类型	描述
topologyKey	字符串	<p>topologyKey 是节点标签的密钥。带有具有此键和相同值标签的节点被视为在同一拓扑中。我们将每个 <key, value> 视为 "bucket", 并尝试在每个存储桶中均衡的 pod 数量。我们将域定义为拓扑的特定实例。另外, 我们定义了一个有资格的域, 其节点满足 nodeAffinityPolicy 和 nodeTaintsPolicy 的要求。例如, 如果 TopologyKey 为 "kubernetes.io/hostname", 则每个节点都是该拓扑的域。如果 TopologyKey 是 "topology.kubernetes.io/zone", 则每个区都是该拓扑的域。它是必填字段。</p>
whenUnsatisfiable	字符串	<p>whenUnsatisfiable 表示在不满足分散约束的情况下如何处理 pod。</p> <ul style="list-style-type: none"> - DoNotSchedule (默认) 告知调度程序不调度它。 - ScheduleAnyway 告知调度程序将 pod 调度到任何位置, 但为拓扑赋予更高优先级, 有助于减少偏移量。只有在每个可能的节点分配在某些拓扑中违反 "MaxSkew" 时, 对传入 pod 的一个约束被视为 "Unsatisfiable"。例如, 在 3 区集群中, MaxSkew 设置为 1, 具有相同 labelSelector 的 pod 分布为 3/1/1: zone1 zone3 zone3 P P P P If WhenUnsatisfiable 被设置为 DoNotSchedule, 传入的 pod 只能调度到 zone2 (zone3) 要成为 3/2/1 (3/1/2), 在 zone2 (zone3) 上作为 ActualSkew (2-1) 满足 MaxSkew (1)。换句话说, 集群仍然可以进行平衡, 但调度程序不会使其更不平衡。它是必填字段。 <p>可能枚举值: -</p> <ul style="list-style-type: none"> "DoNotSchedule" 指示调度程序在不满足约束时不调度 pod。 "ScheduleAnyway" 指示调度程序调度 pod, 即使没有满足限制。

14.14.1.246. `.spec.template.spec.volumes`

描述

可由属于 pod 的容器挂载的卷列表。更多信息：
<https://kubernetes.io/docs/concepts/storage/volumes>

类型

数组

14.14.1.247. `.spec.template.spec.volumes[]`

描述

卷代表一个 pod 中的命名卷，可以被 pod 中的任何容器访问。

类型

object

必填

- name

属性	类型	描述
<code>awsElasticBlockStore</code>	object	代表 AWS 中的 Persistent Disk 资源。 挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。
<code>azureDisk</code>	对象	<code>azureDisk</code> 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。
<code>azureFile</code>	对象	<code>azureFile</code> 代表主机上的 Azure File Service 挂载，并绑定到 pod。
<code>cephfs</code>	object	代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。
<code>cinder</code>	object	代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

属性	类型	描述
configMap	object	<p>将 ConfigMap 调整为卷。</p> <p>目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。</p>
csi	object	代表由外部 CSI 驱动程序管理的卷的源位置
downwardAPI	object	DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。
emptyDir	object	表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。
ephemeral	object	代表由普通存储驱动程序处理的临时卷。
fc	object	代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。
flexVolume	对象	FlexVolume 代表一个通用卷资源，它使用基于 exec 的插件置备/附加。
flocker	object	代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。
gcePersistentDisk	object	<p>代表 Google Compute Engine 中的 Persistent Disk 资源。</p> <p>在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。</p>

属性	类型	描述
gitRepo	object	代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。 DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。
glusterfs	object	代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。
hostPath	object	代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。
iscsi	object	代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。
name	string	卷的名称。必须是 DNS_LABEL，且在 pod 中唯一。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。 PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。
photonPersistentDisk	object	代表 Photon Controller 持久磁盘资源。
portworxVolume	object	PortworxVolumeSource 代表一个 Portworx volume 资源。
projected	object	代表投射卷源

属性	类型	描述
quobyte	object	代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。
rbd	object	代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。
scaleIO	object	ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷
secret	object	将 Secret 调整为卷。 目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。
storageos	object	代表 StorageOS 持久性卷资源。
vsphereVolume	object	代表 vSphere 卷资源。

14.14.1.248. .spec.template.spec.volumes[].awsElasticBlockStore

描述

代表 AWS 中的 Persistent Disk 资源。

挂载到容器前必须存在 AWS EBS 磁盘。磁盘还必须与 kubelet 位于同一个 AWS 区域。AWS EBS 磁盘只能以读写模式挂载。AWS EBS 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4"，"xfs"，"ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

属性	类型	描述
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。
readOnly	布尔值	readonly 值 true 将强制 VolumeMounts 中的 readOnly 设置。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	VolumeID 是 AWS (Amazon EBS 卷) 中的持久磁盘资源的唯一 ID。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

14.14.1.249. .spec.template.spec.volumes[].azureDisk

描述

azureDisk 代表主机上的 Azure Data Disk 挂载，并绑定到 pod。

类型

object

必填

- **diskName**
- **diskURI**

属性	类型	描述
cacheMode	string	cacheMode 是主机缓存模式：None、Read Only、Read Write。 可能枚举值：- "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskname 是 blob 存储中的数据磁盘的名称
diskURI	string	diskURI 是 blob 存储中数据磁盘的 URI

属性	类型	描述
fsType	string	fstype 是要挂载的 Filesystem 类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
kind	string	kind expected 值是 Shared: multiple blob disk per storage account Managed: azure disk per storage account Managed: azure managed data disk (only in managed availability set))。默认为 shared possible enum 值: - "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。

14.14.1.250. .spec.template.spec.volumes[].azureFile

描述

azureFile 代表主机上的 Azure File Service 挂载, 并绑定到 pod。

类型

object

必填

- **secretName**
- **shareName**

属性	类型	描述
readOnly	布尔值	只读默认为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretName	string	secretName 是包含 Azure Storage Account Name 和 Key 的 secret 名称

属性	类型	描述
shareName	string	sharename 是 azure 共享名称

14.14.1.251. .spec.template.spec.volumes[].cephfs

描述

代表 Ceph 文件系统挂载，最后一个 pod Cephfs 卷的生命周期不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **monitor**

属性	类型	描述
monitor	数组（字符串）	Monitors is Required: Monitor is a collection of Ceph monitors more info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	路径为可选：作为挂载的 root 使用，而不是完整的 Ceph 树，默认为 /
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretFile	string	secretfile 是可选：SecretFile 是 User 的密钥环的路径，默认为 /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您您在同一个命名空间中找到引用的对象。

属性	类型	描述
user	string	用户是可选的：User 是 rados 用户名，默认为 admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

14.14.1.252. .spec.template.spec.volumes[].cephfs.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.253. .spec.template.spec.volumes[].cinder

描述

代表 Openstack 中的 cinder 卷资源。挂载到容器之前必须存在 Cinder 卷。卷还必须与 kubelet 位于同一区域。Cinder 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **volumelD**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

属性	类型	描述
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
volumeID	string	用于识别 cinder 中的卷的卷 ID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

14.14.1.254. .spec.template.spec.volumes[].cinder.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.255. .spec.template.spec.volumes[].configMap

描述

将 ConfigMap 调整为卷。

目标 ConfigMap 的 Data 字段的内容将作为文件显示在卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。ConfigMap 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选的：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.14.1.256. .spec.template.spec.volumes[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.14.1.257. .spec.template.spec.volumes[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.14.1.258. .spec.template.spec.volumes[].csi

描述

代表由外部 CSI 驱动程序管理的卷的源位置

类型

object

必填

- **driver**

属性	类型	描述
driver	string	driver 是处理这个卷的 CSI 驱动程序的名称。请参考您的管理员获取集群中注册的正确名称。

属性	类型	描述
fsType	string	要挂载的 fstype。Ex. "ext4", "xfs", "ntfs". 如果没有提供, 空值会被传递给关联的 CSI 驱动程序, 该驱动程序将决定要应用的默认文件系统。
nodePublishSecretRef	object	LocalObjectReference 包含足够的信息, 供您在同一个命名空间中 找到引用的对象。
readOnly	布尔值	readonly 指定卷的只读配置。默认为 false (读/写)。
volumeAttributes	对象 (字符串)	volumeAttributes 存储传递给 CSI 驱动程序的特定于驱动程序的属性。如需支持的值, 请参考您的驱动程序文档。

14.14.1.259. .spec.template.spec.volumes[].csi.nodePublishSecretRef

描述

LocalObjectReference 包含足够的信息, 供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.260. .spec.template.spec.volumes[].downwardAPI

描述

DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。Downward API 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	可选：默认在创建的文件中使用的模式位。必须是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	array	items 是 Downward API 卷文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.14.1.261. .spec.template.spec.volumes[].downwardAPI.items

描述

items 是 Downward API 卷文件列表

类型

数组

14.14.1.262. .spec.template.spec.volumes[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- path

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。

属性	类型	描述
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.14.1.263. .spec.template.spec.volumes[].downwardAPI.items[].fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- fieldPath

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.14.1.264. .spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- resource

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.14.1.265. .spec.template.spec.volumes[].emptyDir

描述

表示 pod 的空目录。空目录卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
中	string	Medium 代表这个目录应该支持哪些存储类型。默认值为 "", 它需要使用节点的默认介质。必须是空字符串（默认）或 Memory。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir
sizeLimit	数量	sizeLimit 是此 EmptyDir 卷所需的本地存储总量。大小限制也适用于内存介质。内存用量 EmptyDir 的最大用法是此处指定的 SizeLimit 和 pod 中所有容器的内存限值总和。默认值为 nil, 表示限制未定义。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir

14.14.1.266. .spec.template.spec.volumes[].ephemeral

描述

代表由普通存储驱动程序处理的临时卷。

类型

object

属性	类型	描述
volumeClaimTemplate	object	PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

14.14.1.267. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

描述

PersistentVolumeClaimTemplate 用于生成 PersistentVolumeClaim 对象，作为 EphemeralVolumeSource 的一部分。

类型

object

必填

- **spec**

属性	类型	描述
metadata	ObjectMeta	可以包含在创建时复制到 PVC 中的标签和注解。不允许其他字段，并在验证过程中被拒绝。
spec	对象	PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

14.14.1.268. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许特定于供应商的属性的源

类型

object

属性	类型	描述
accessModes	数组（字符串）	accessModes 包含卷应具有的所需访问模式。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	对象	TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

属性	类型	描述
dataSourceRef	object	<p>dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。</p>
resources	对象	ResourceRequirements 描述计算资源要求。
selector	labelSelector	selector 是卷上的标签查询，用于绑定。
storageClassName	string	storageClassName 是声明所需的 StorageClass 的名称。更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

属性	类型	描述
volumeMode	字符串	volumeMode 定义声明所需的卷类型。当没有包括在 claim spec 中时，文件系统的值会被简化。 可能枚举值：- "Block" 表示卷不会被格式化为文件系统，并将保留原始块设备。- "Filesystem" 表示该卷将使用文件系统进行或格式化。
volumeName	string	volumeName 是支持此声明的 PersistentVolume 的绑定引用。

14.14.1.269. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

描述

TypedLocalObjectReference 包含足够信息，可让您找到同一命名空间中的 typed referenced 对象。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

14.14.1.270. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

描述

dataSourceRef 指定在需要非空卷时，使用数据填充卷的对象。这可以是来自非空 API 组（非核心对象）或 PersistentVolumeClaim 对象的任何对象。当指定此字段时，只有在指定对象的类型与一些已安装的卷填充器或动态置备程序匹配时才会成功。此字段将替换 dataSource 字段的功能，因此如果这两个字段都不是空的，则它们必须具有相同的值。为向后兼容，当 namespace 没有在 dataSourceRef 中指定时，如果其中一个字段(dataSource 和 dataSourceRef)都自动设置为相同的值，另一个是非空的值。当在 dataSourceRef 中指定命名空间时，dataSource 不会设置为相同的值，且必须为空。

dataSource 和 dataSourceRef: * While dataSource 有三个重要区别：但 dataSourceRef 只允许两种特定类型的对象，dataSourceRef 允许任何非核心对象和 PersistentVolumeClaim 对象。* 虽然 dataSource 忽略禁止的值（过滤它们），dataSourceRef 会保留所有值，并在指定了禁止的值时生成错误。* 虽然 dataSource 只允许本地对象，但 dataSourceRef 允许任何命名空间中的对象。(beta)使用此字段需要启用 AnyVolumeDataSource 功能门。(alpha)使用 dataSourceRef 的 namespace 字段需要启用 CrossNamespaceVolumeDataSource 功能门。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称
namespace	string	namespace 是被引用的资源的命名空间。在指定命名空间时，引用命名空间中需要 gateway.networking.k8s.io/ReferenceGrant 对象，以允许该命名空间的所有者接受引用。详情请查看 ReferenceGrant 文档。(alpha)此字段需要启用 CrossNamespaceVolumeDataSource 功能门。

14.14.1.271. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

描述

ResourceRequirements 描述计算资源要求。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
声明	数组	声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。 这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。 此字段是不可变的。它只能为容器设置。
claim[]	object	ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。
limits	对象（数量）	限制描述了允许的最大计算资源量。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	对象（数量）	Requests 描述了所需的最少计算资源。如果容器省略了 Requests，则默认为 Limits（如果明确指定），否则默认为实现定义的值。请求不能超过限值。更多信息： https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.14.1.272. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.c

描述

声明列出了此容器使用的 spec.resourceClaims 中定义的资源名称。
这是一个 alpha 字段，需要启用 DynamicResourceAllocation 功能门。

此字段是不可变的。它只能为容器设置。

类型

数组

14.14.1.273. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.c

描述

ResourceClaim 引用 PodSpec.ResourceClaims 中的一个条目。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中一个条目的名称匹配。它使得资源在容器内可用。

14.14.1.274. .spec.template.spec.volumes[].fc**描述**

代表光纤通道卷。Fibre Channel 卷只能以读写模式挂载。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
LUN	整数	LUN 是可选的: FC 目标 lun 号
readOnly	布尔值	readonly 是可选的: 默认为 false (读/写)。此处的 readonly 将强制使用 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组 (字符串)	targetWWN 可选: FC 目标全球名称(WWN)
wwids	数组 (字符串)	wwids 可选: FC 卷全局广泛的标识符(wwids) Either wwids 或 targetWWNs 和 lun 的组合必须同时设置, 但不能同时设置。

14.14.1.275. .spec.template.spec.volumes[].flexVolume**描述**

FlexVolume 代表一个通用卷资源, 它使用基于 exec 的插件置备/附加。

类型

object

必填

- driver

属性	类型	描述
driver	string	driver 是要用于此卷的驱动程序名称。
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认文件系统取决于 FlexVolume 脚本。
options	对象 (字符串)	选项是可选的：此字段包含额外的命令选项（若有）。
readOnly	布尔值	readonly 是可选的：默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

14.14.1.276. .spec.template.spec.volumes[].flexVolume.secretRef

描述

LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.277. .spec.template.spec.volumes[].flocker

描述

代表由 Flocker 代理挂载的 Flocker 卷。应该只设置其中一个 datasetName 和 datasetUUID。flocker 卷不支持所有权管理或 SELinux 重新标记。

类型

object

属性	类型	描述
datasetName	string	datasetName 是存储在 Flocker 的 dataset 的 dataset 的 dataset 的 Name。
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

14.14.1.278. .spec.template.spec.volumes[].gcePersistentDisk

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

在挂载到容器前，GCE PD 必须已经存在。磁盘还必须与 kubelet 位于同一个 GCE 项目和区。GCE PD 只能以读写一次挂载，或者多次只读。GCE PDs 支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **pdName**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
分区	整数	partition 是您要挂载的卷中的分区。如果省略，则默认为按卷名称挂载。示例：对于卷 /dev/sda1，您可以将分区指定为 "1"。同样，/dev/sda 的卷分区是 "0"（或者您可以将属性留空）。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

属性	类型	描述
pdName	string	pdName 是 GCE 中的 PD 资源的唯一名称。用于识别 GCE 中的磁盘。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

14.14.1.279. .spec.template.spec.volumes[].gitRepo

描述

代表填充 git 存储库内容的卷。Git 仓库卷不支持所有权管理。Git repo 卷支持 SELinux 重新标记。
DEPRECATED: GitRepo 已被弃用。要使用 git 仓库置备容器，请将 EmptyDir 挂载到使用 git 克隆存储库的 InitContainer 中，然后将 EmptyDir 挂载到 Pod 的容器中。

类型

object

必填

- 软件仓库

属性	类型	描述
目录	string	directory 是目标目录名称。不得包含或以 '..' 开头。如果提供了 '.', 则卷目录将是 git 存储库。否则，如果指定，卷将包含具有指定名称的子目录中的 git 存储库。
软件仓库	string	repository 是 URL
revision	string	revision 是指定修订版本的提交哈希。

14.14.1.280. .spec.template.spec.volumes[].glusterfs

描述

代表 pod 生命周期的 Glusterfs 挂载。GlusterFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- 端点
- path

属性	类型	描述
端点	string	Endpoints 是详细信息 Glusterfs 拓扑的端点名称。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path 是 Glusterfs 卷路径。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Glusterfs 卷。默认为false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

14.14.1.281. .spec.template.spec.volumes[].hostPath

描述

代表映射到 pod 的主机路径。主机路径卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- path

属性	类型	描述
path	string	主机上目录的路径。如果路径是符号链接，它将遵循到实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath

属性	类型	描述
type	string	<p>HostPath Volume Defaults 为 "" more info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>可能枚举值：- "" 对于向后兼容，如果未设置 - "BlockDevice" A 块设备必须存在于给定路径 - "CharDevice" A 字符设备必须存在于给定路径上 - "Directory" A 目录必须存在于给定路径 - "DirectoryOrCreate" 中（如果给定路径不存在），将根据需要创建一个空目录，且文件模式为 0755，其具有与 Kubelet 相同的组和所有权。- "File" A 文件必须存在于给定路径上 - "FileOrCreate" 如果给定路径上不存在，将根据需要创建一个空文件，该文件与文件模式 0644 具有相同的组和所有权。- "Socket" A UNIX 套接字必须存在于给定路径上。</p>

14.14.1.282. .spec.template.spec.volumes[].iscsi

描述

代表 iSCSI 磁盘。iSCSI 卷只能以读写模式挂载。iSCSI 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **targetPortal**
- **IQN**
- **LUN**

属性	类型	描述
chapAuthDiscovery	布尔值	chapAuthDiscovery 定义是否支持 iSCSI Discovery CHAP 身份验证
chapAuthSession	布尔值	chapAuthSession 定义是否支持 iSCSI Session CHAP 身份验证

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorname 是自定义 iSCSI 启动器名称。如果同时使用 iscsiInterface 指定 initiatorName，则会为连接创建新的 iSCSI 接口 <target portal>: <volume name>。
IQN	string	IQN 是目标 iSCSI 限定名称。
iscsiInterface	string	iscsiInterface 是使用 iSCSI 传输的接口名称。默认为 'default'(tcp)。
LUN	整数	LUN 代表 iSCSI 目标 Lun 编号。
门户	数组 (字符串)	门户是 iSCSI 目标门户列表。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
targetPortal	string	targetPortal 是 iSCSI 目标门户。如果端口不是默认值 (通常是 TCP 端口 860 和 3260)，则门户可以是 IP 或 ip_addr:port。

14.14.1.283. .spec.template.spec.volumes[].iscsi.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.284. .spec.template.spec.volumes[].nfs

描述

代表 pod 的生命周期最后一个 NFS 挂载。NFS 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **server**
- **path**

属性	类型	描述
path	string	NFS 服务器导出的路径。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	布尔值	此处的只读将强制使用只读权限挂载 NFS 导出。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server 是 NFS 服务器的主机名或 IP 地址。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#nfs

14.14.1.285. .spec.template.spec.volumes[].persistentVolumeClaim

描述

PersistentVolumeClaimVolumeSource 引用同一命名空间中的用户 PVC。此卷找到绑定的 PV，并为 pod 挂载这个卷。PersistentVolumeClaimVolumeSource 本质上是另一种类型的卷，它们归其他人（系统）所有。

类型

object

必填

- **claimName**

属性	类型	描述
claimName	string	claimName 是使用此卷与 pod 相同的命名空间中的 PersistentVolumeClaim 的名称。 更多信息： https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	布尔值	readonly 将强制 VolumeMount 中的 ReadOnly 设置。默认 false。

14.14.1.286. .spec.template.spec.volumes[].photonPersistentDisk

描述

代表 Photon Controller 持久磁盘资源。

类型

object

必填

- **pdID**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为 "ext4"。
pdID	string	pdID 是标识 Photon Controller 持久磁盘的 ID

14.14.1.287. .spec.template.spec.volumes[].portworxVolume

描述

PortworxVolumeSource 代表一个 Portworx volume 资源。

类型

object

必填

- **volumeID**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 代表要挂载的文件系统类型，必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs". 如果未指定，则隐式推断为"ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
volumeID	string	VolumeID 唯一标识一个 Portworx 卷

14.14.1.288. .spec.template.spec.volumes[].projected

描述

代表投射卷源

类型

object

属性	类型	描述
defaultMode	整数	defaultMode 是默认用来对创建文件设置权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
sources	数组	sources 是卷投射列表
sources[]	对象	可与其他支持的卷类型一起投射的预测

14.14.1.289. .spec.template.spec.volumes[].projected.sources

描述

sources 是卷投射列表

类型

数组

14.14.1.290. `.spec.template.spec.volumes[].projected.sources[]`

描述

可与其他支持的卷类型一起投射的预测

类型

object

属性	类型	描述
configMap	object	将 ConfigMap 调整为投射卷。 目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。
downwardAPI	object	代表下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。
secret	object	将 secret 适应到投射卷中。 目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。
serviceAccountToken	object	ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

14.14.1.291. `.spec.template.spec.volumes[].projected.sources[].configMap`

描述

将 ConfigMap 调整为投射卷。

目标 ConfigMap 的 Data 字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名，除非 items 元素填充了键到路径的特定映射。请注意，这与没有默认模式的 configmap 卷源相同。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
items	数组	如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选指定是否必须定义 ConfigMap 还是其键

14.14.1.292. .spec.template.spec.volumes[].projected.sources[].configMap.items

描述

如果未指定项目，则引用 ConfigMap 的 Data 字段中的每一键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 ConfigMap 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.14.1.293. .spec.template.spec.volumes[].projected.sources[].configMap.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- key
- path

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.14.1.294. .spec.template.spec.volumes[].projected.sources[].downwardAPI

描述

代表下下 API 信息，用于项目到投射卷中。请注意，这与没有默认模式的卷源相同。

类型

object

属性	类型	描述
items	array	items 是 DownwardAPIVolume 文件列表
items[]	对象	DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

14.14.1.295. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items

描述

items 是 DownwardAPIVolume 文件列表

类型

数组

14.14.1.296. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表信息，以创建包含 pod 字段的文件

类型

object

必填

- path

属性	类型	描述
fieldRef	object	ObjectFieldSelector 选择对象的 APIVersioned 字段。
模式	整数	可选：用于在此文件上设置权限的模式位，必须是 0000 到 0777 之间或 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能绝对或包含 '..' 路径。必须采用 utf-8 编码。相对路径的第一个项目不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

14.14.1.297. .spec.template.spec.volumes[*i*].projected.sources[*j*].downwardAPI.items[*k*].fieldRef

描述

ObjectFieldSelector 选择对象的 APIVersioned 字段。

类型

object

必填

- fieldPath

属性	类型	描述
apiVersion	字符串	FieldPath 用术语编写的 schema 版本，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

14.14.1.298. `.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resource`

描述

ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

类型

object

必填

- **resource**

属性	类型	描述
containerName	字符串	容器名称：卷所需，对于 env vars 是可选的
divisor	数量	指定公开资源的输出格式，默认为 "1"
resource	字符串	必需：要选择的资源

14.14.1.299. `.spec.template.spec.volumes[].projected.sources[].secret`

描述

将 secret 适应到投射卷中。

目标 Secret 的数据字段的内容将作为文件出现在投射卷中，使用 Data 字段中的密钥作为文件名。请注意，这与没有默认模式的 secret 卷源相同。

类型

object

属性	类型	描述
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥

14.14.1.300. .spec.template.spec.volumes[].projected.sources[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.14.1.301. .spec.template.spec.volumes[].projected.sources[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object

必填

- key
- path

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。

属性	类型	描述
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.14.1.302. .spec.template.spec.volumes[].projected.sources[].serviceAccountToken

描述

ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此投射可用于将服务帐户令牌插入到 pod 运行时文件系统中，以用于 API (Kubernetes API 服务器或其他)。

类型

object

必填

- **path**

属性	类型	描述
受众	string	受众是令牌的预期受众。令牌的接收者必须使用令牌使用者中指定的标识符来识别自己，否则应拒绝令牌。使用者默认为 apiserver 的标识符。
expirationSeconds	整数	expirationSeconds 是服务帐户令牌请求持续时间。随着令牌方法到期，kubelet 卷插件将主动轮转服务帐户令牌。如果令牌存在的时间超过这个值的 80%，或者令牌超过 24 小时，则 kubelet 将开始尝试轮转令牌，且必须至少为 10 分钟。
path	string	path 是相对于文件挂载点的路径，以将令牌项目到其中。

14.14.1.303. .spec.template.spec.volumes[].quobyte

描述

代表 pod 的生命周期持续的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。

类型

object

必填

- **registry**

- 卷

属性	类型	描述
group	string	将卷访问映射到 Default 的组没有组
readOnly	布尔值	此处的只读将强制使用只读权限挂载 Quobyte 卷。默认为false。
registry	string	registry 代表一个或多个 Quobyte Registry 服务，作为字符串指定为 host:port 对（多个条目用逗号分隔），后者充当卷的中央 registry
tenant	string	在带有动态置备 Quobyte 卷的后端中拥有给定 Quobyte 卷的租户，值由插件设置
user	string	用户将卷访问权限映射到默认值到 serviceaccount 用户
卷	string	卷是一个字符串，它根据名称引用已创建的 Quobyte 卷。

14.14.1.304. .spec.template.spec.volumes[].rbid

描述

代表 Rados 块设备挂载，该挂载最后是 pod 的生命周期。RBD 卷支持所有权管理和 SELinux 重新标记。

类型

object

必填

- **monitor**
- **image**

属性	类型	描述
fsType	string	fstype 是您要挂载的卷的文件系统类型。提示：确保主机操作系统支持文件系统类型。示例："ext4", "xfs", "ntfs"。如果未指定，则隐式推断为"ext4"。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#rbd

属性	类型	描述
image	string	image 是 rados 镜像名称。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring 是 RBDUser 的密钥环的路径。默认为 /etc/ceph/keyring。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitor	数组（字符串）	monitor 是 Ceph 监视器的集合。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool 是 rados 池名称。默认为 rbd。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	布尔值	此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。默认为 false。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
user	string	user 是 rados 用户名。默认为 admin。更多信息： https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

14.14.1.305. .spec.template.spec.volumes[].rbd.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.306. .spec.template.spec.volumes[].scaleIO

描述

ScaleIOVolumeSource 代表一个持久的 ScaleIO 卷

类型

object

必填

- **gateway**
- **system**
- **secretRef**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 默认为 "xfs".
gateway	string	gateway 是 ScaleIO API 网关的主机地址。
protectionDomain	string	protectionDomain 是配置的存储的 ScaleIO 保护域的名称。
readOnly	布尔值	只读默认值为 false (读/写)。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
sslEnabled	布尔值	sslEnabled Flag 启用/禁用与网关的 SSL 通信，默认为 false

属性	类型	描述
storageMode	string	storageMode 指示卷的存储应该是 ThickProvisioned 或 ThinProvisioned。默认为 ThinProvisioned。
storagePool	string	storagePool 是与保护域关联的 ScaleIO 存储池。
system	string	system 是存储系统的名称，如 ScaleIO 中配置。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

14.14.1.307. .spec.template.spec.volumes[].scaleIO.secretRef

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.308. .spec.template.spec.volumes[].secret

描述

将 Secret 调整为卷。

目标 Secret 的数据字段的内容将作为文件显示在卷中，并将 Data 字段中的密钥用作文件名。Secret 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
defaultMode	整数	defaultMode 是可选：用于默认设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。默认值为 0644。该路径中的目录不受此设置的影响。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。
items[]	对象	将字符串键映射到卷中的路径。
optional	布尔值	可选字段指定是否需要定义 Secret 或其密钥
secretName	string	secretName 是要使用的 pod 命名空间中的 secret 名称。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#secret

14.14.1.309. .spec.template.spec.volumes[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中每个键值对将投射到卷中，名称为键，内容是值。如果指定，列出的键将投射到指定的路径中，且未列出的密钥将不存在。如果指定了在 Secret 中不存在的密钥，则卷设置将错误，除非标记为可选。路径必须是 relative，且不能包含 '..' 路径或以 '..' 开头。

类型

数组

14.14.1.310. .spec.template.spec.volumes[].secret.items[]

描述

将字符串键映射到卷中的路径。

类型

object**必填**

- **key**
- **path**

属性	类型	描述
key	string	key 是项目的关键。
模式	整数	模式是可选的：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，也可以是 0 到 511 之间的十进制值。YAML 接受八进制值和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能会与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是设置其他模式位。
path	string	path 是要将密钥映射到的文件的相对路径。可能不是绝对路径。不得包含 path 元素 '..'。不能以字符串 '..' 开始。

14.14.1.311. .spec.template.spec.volumes[].storageos**描述**

代表 StorageOS 持久性卷资源。

类型**object**

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定，则隐式推断为 "ext4"。
readOnly	布尔值	只读默认为 false（读/写）。此处的 readOnly 将强制使用 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

属性	类型	描述
volumeName	string	volumeName 是 StorageOS 卷的人类可读名称。卷名称只在命名空间中是唯一的。
volumeNamespace	string	volumeNamespace 指定 StorageOS 中卷的范围。如果没有指定命名空间，则使用 Pod 的命名空间。这允许在 StorageOS 中镜像 Kubernetes 名称范围，以便在更紧密的集成。将 VolumeName 设置为任何名称来覆盖默认的行为。如果没有在 StorageOS 中使用命名空间，则设置为 "default"。将创建在 StorageOS 中没有预先存在的命名空间。

14.14.1.312. .spec.template.spec.volumes[].storageos.secretRef

描述

LocalObjectReference 包含足够的信息，供您同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.14.1.313. .spec.template.spec.volumes[].vsphereVolume

描述

代表 vSphere 卷资源。

类型

object

必填

- **volumePath**

属性	类型	描述
----	----	----

属性	类型	描述
fsType	string	fstype 是要挂载的文件系统类型。必须是主机操作系统支持的文件系统类型。Ex. "ext4", "xfs", "ntfs". 如果未指定, 则隐式推断为"ext4"。
storagePolicyID	string	StoragePolicyID 是与 StoragePolicyName 关联的存储 Policy Based Management (SPBM)配置集 ID。
storagePolicyName	string	StoragePolicyName 是存储基于策略的管理(SPBM)配置集名称。
volumePath	string	volumePath 是标识 vSphere 卷 vmdk 的路径

14.14.1.314. .status

描述

ReplicationControllerStatus 代表复制控制器的当前状态。

类型

object

必填

- **replicas**

属性	类型	描述
availableReplicas	整数	此复制控制器的可用副本数 (至少为 minReadySeconds) 。
conditions	数组	代表复制控制器的当前状态的最新可用影响。
conditions[]	object	ReplicationControllerCondition 描述了复制控制器在特定点的状态。
fullyLabeledReplicas	整数	具有与复制控制器 pod 模板标签匹配的标签的 pod 数量。
observedGeneration	整数	observedGeneration 反映了最近观察到的复制控制器的生成。
readyReplicas	整数	此复制控制器的就绪副本数。

属性	类型	描述
replicas	整数	replicas 是最近观察到的副本数。 更多信息： https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller

14.14.1.315. .status.conditions

描述

代表复制控制器的当前状态的最新可用影响。

类型

array

14.14.1.316. .status.conditions[]

描述

ReplicationControllerCondition 描述了复制控制器在特定点的状态。

类型

object

必填

- **type**
- **status**

属性	类型	描述
lastTransitionTime	时间	条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	string	复制控制器条件的类型。

14.14.2. API 端点

可用的 API 端点如下：

- **/api/v1/replicationcontrollers**
 - **GET**: 列出或监视类型为 ReplicationController 的对象
- **/api/v1/watch/replicationcontrollers**
 - **GET**: 观察单个对 ReplicationController 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/replicationcontrollers**
 - **DELETE** : 删除 ReplicationController 的集合
 - **GET**: 列出或监视类型为 ReplicationController 的对象
 - **POST** : 创建 ReplicationController
- **/api/v1/watch/namespaces/{namespace}/replicationcontrollers**
 - **GET**: 观察单个对 ReplicationController 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}**
 - **DELETE** : 删除 ReplicationController
 - **GET** : 读取指定的 ReplicationController
 - **PATCH** : 部分更新指定的 ReplicationController
 - **PUT** : 替换指定的 ReplicationController
- **/api/v1/watch/namespaces/{namespace}/replicationcontrollers/{name}**
 - **GET**: 观察对类型为 ReplicationController 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status**
 - **GET** : 指定 ReplicationController 的读取状态
 - **PATCH** : 部分更新指定 ReplicationController 的状态
 - **PUT** : 替换指定 ReplicationController 的状态

14.14.2.1. /api/v1/replicationcontrollers

表 14.374. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 'resourceVersion' 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 ReplicationController 类型的对象

表 14.375. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationControllerList schema
401 - 未授权	空

14.14.2.2. /api/v1/watch/replicationcontrollers

表 14.376. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ReplicationController 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.377. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.14.2.3. /api/v1/namespaces/{namespace}/replicationcontrollers

表 14.378. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.379. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ReplicationController 的集合

表 14.380. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.381. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.382. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 ReplicationController 类型的对象

表 14.383. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.384. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationControllerList schema
401 - 未授权	空

HTTP 方法

POST

描述

创建 ReplicationController

表 14.385. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.386. 主体参数

参数	类型	描述
正文 (body)	ReplicationController 模式	

表 14.387. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
201 - Created	ReplicationController 模式
202 - Accepted	ReplicationController 模式
401 - 未授权	空

14.14.2.4. /api/v1/watch/namespaces/{namespace}/replicationcontrollers

表 14.388. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.389. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ReplicationController 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.390. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.14.2.5. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}

表 14.391. 全局路径参数

参数	类型	描述
name	string	ReplicationController 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.392. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ReplicationController

表 14.393. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.394. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.395. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 ReplicationController

表 14.396. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ReplicationController</code> 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

部分更新指定的 ReplicationController

表 14.397. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.398. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.399. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
201 - Created	ReplicationController 模式
401 - 未授权	空

HTTP 方法

PUT

描述

替换指定的 ReplicationController

表 14.400. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.401. 主体参数

参数	类型	描述
正文 (body)	ReplicationController 模式	

表 14.402. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
201 - Created	ReplicationController 模式
401 - 未授权	空

14.14.2.6. /api/v1/watch/namespaces/{namespace}/replicationcontrollers/{name}

表 14.403. 全局路径参数

参数	类型	描述
name	string	ReplicationController 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.404. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 ReplicationController 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.405. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.14.2.7. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status

表 14.406. 全局路径参数

参数	类型	描述
name	string	ReplicationController 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.407. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 ReplicationController 的读取状态

表 14.408. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
401 - 未授权	空

HTTP 方法

PATCH

描述

指定 ReplicationController 的部分更新状态

表 14.409. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为： <ul style="list-style-type: none"> - Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.410. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.411. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
201 - Created	ReplicationController 模式
401 - 未授权	空

HTTP 方法**PUT****描述**

替换指定 ReplicationController 的状态

表 14.412. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为： <ul style="list-style-type: none"> - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.413. 主体参数

参数	类型	描述
正文 (body)	ReplicationController 模式	

表 14.414. HTTP 响应

HTTP 代码	响应正文
200 - OK	ReplicationController 模式
201 - Created	ReplicationController 模式
401 - 未授权	空

14.15. RESOURCEQUOTA [V1]

描述

ResourceQuota 设置每个命名空间强制的聚合配额限制

类型

对象

14.15.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	ResourceQuotaSpec 定义执行配额所需的硬限制。
status	对象	ResourceQuotaStatus 定义强制硬限制和观察到的使用。

14.15.1.1. .spec

描述

ResourceQuotaSpec 定义执行配额所需的硬限制。

类型

object

属性	类型	描述
hard	对象（数量）	hard 是每个指定资源所需的硬限制集合。更多信息： https://kubernetes.io/docs/concepts/policy/resource-quotas/

属性	类型	描述
scopeSelector	对象	范围选择器代表由 scoped-resource 选择器要求代表的选择器的 AND。
scopes	数组（字符串）	一个过滤器集合，必须与配额跟踪的每个对象匹配。如果没有指定，配额会匹配所有对象。

14.15.1.2. .spec.scopeSelector

描述

范围选择器代表由 scoped-resource 选择器要求代表的选择器的 AND。

类型

object

属性	类型	描述
matchExpressions	array	根据资源范围划分的范围选择器要求列表。
matchExpressions[]	对象	有范围的资源选择器要求是一个选择器，其中包含值、范围名称以及与范围名称和值相关的 Operator。

14.15.1.3. .spec.scopeSelector.matchExpressions

描述

根据资源范围划分的范围选择器要求列表。

类型

array

14.15.1.4. .spec.scopeSelector.matchExpressions[]

描述

有范围的资源选择器要求是一个选择器，其中包含值、范围名称以及与范围名称和值相关的 Operator。

类型

object

必填

- **scopeName**
- **operator**

属性	类型	描述
operator	字符串	表示范围与一组值的关系。有效的运算符为 In、NotIn、Exists、DoesNotExist。 possible enum 值： - "DoesNotExist" - "Exists" - "In" - "NotIn"
scopeName	字符串	选择器应用到的范围的名称。 可能枚举值： - "BestEffort" 匹配所有具有最佳服务质量的 pod 对象 - "CrossNamespacePodAffinity" 匹配所有具有跨命名空间 pod (anti) affinity 的 pod 对象。 - "NotBestEffort" 匹配所有还没有匹配的 pod 对象 最佳服务质量 - "NotTerminating" Match all pod objects where spec.activeDeadlineSeconds 为 nil - "PriorityClass" Match all pod objects with priority class mentioned - "Terminating" Match all pod objects where spec.activeDeadlineSeconds >=0
values	数组（字符串）	字符串值数组。如果运算符是 In 或 NotIn，则值数组必须是非空的。如果运算符是 Exists 或 DoesNotExist，则值数组必须为空。这个数组会在策略性合并补丁中被替换。

14.15.1.5. .status

描述

ResourceQuotaStatus 定义强制硬限制和观察到的使用。

类型

object

属性	类型	描述
hard	对象（数量）	hard 是每个命名资源的强制硬限制集合。更多信息： https://kubernetes.io/docs/concepts/policy/resource-quotas/

属性	类型	描述
使用的	对象（数量）	使用的是命名空间中资源的总使用量。

14.15.2. API 端点

可用的 API 端点如下：

- **/api/v1/resourcequotas**
 - **GET**: 列出或监视类型为 ResourceQuota 的对象
- **/api/v1/watch/resourcequotas**
 - **GET**: 观察单个对 ResourceQuota 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/resourcequotas**
 - **DELETE**：删除 ResourceQuota 集合
 - **GET**: 列出或监视类型为 ResourceQuota 的对象
 - **POST**：创建一个 ResourceQuota
- **/api/v1/watch/namespaces/{namespace}/resourcequotas**
 - **GET**: 观察单个对 ResourceQuota 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/resourcequotas/{name}**
 - **DELETE**：删除 ResourceQuota
 - **GET**：读取指定的 ResourceQuota
 - **PATCH**：部分更新指定的 ResourceQuota
 - **PUT**：替换指定的 ResourceQuota
- **/api/v1/watch/namespaces/{namespace}/resourcequotas/{name}**
 - **GET**: 观察对类型为 ResourceQuota 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{namespace}/resourcequotas/{name}/status**
 - **GET**: 指定 ResourceQuota 的读取状态
 - **PATCH**：部分更新指定 ResourceQuota 的状态
 - **PUT**：替换指定 ResourceQuota 的状态

14.15.2.1. /api/v1/resourcequotas

表 14.415. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 ResourceQuota 的对象

表 14.416. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuotaList 模式
401 - Unauthorized	空

14.15.2.2. /api/v1/watch/resourcequotas

表 14.417. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ResourceQuota 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.418. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.15.2.3. /api/v1/namespaces/{namespace}/resourcequotas

表 14.419. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.420. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ResourceQuota 集合

表 14.421. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.422. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.423. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ResourceQuota 的对象

表 14.424. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.425. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuotaList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 ResourceQuota

表 14.426. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.427. 主体参数

参数	类型	描述
正文 (body)	ResourceQuota 架构	

表 14.428. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
201 - Created	ResourceQuota 架构
202 - Accepted	ResourceQuota 架构
401 - Unauthorized	空

14.15.2.4. /api/v1/watch/namespaces/{namespace}/resourcequotas

表 14.429. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.430. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ResourceQuota 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.431. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.15.2.5. /api/v1/namespaces/{namespace}/resourcequotas/{name}

表 14.432. 全局路径参数

参数	类型	描述
name	字符串	ResourceQuota 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.433. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ResourceQuota

表 14.434. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.435. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.436. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ResourceQuota</code> 架构
202 - Accepted	<code>ResourceQuota</code> 架构
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 `ResourceQuota`

表 14.437. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ResourceQuota</code> 架构
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `ResourceQuota`

表 14.438. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.439. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.440. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
201 - Created	ResourceQuota 架构
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ResourceQuota

表 14.441. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.442. 主体参数

参数	类型	描述
正文 (body)	ResourceQuota 架构	

表 14.443. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
201 - Created	ResourceQuota 架构
401 - Unauthorized	空

14.15.2.6. /api/v1/watch/namespaces/{namespace}/resourcequotas/{name}

表 14.444. 全局路径参数

参数	类型	描述
name	字符串	ResourceQuota 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.445. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 ResourceQuota 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.446. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.15.2.7. /api/v1/namespaces/{namespace}/resourcequotas/{name}/status

表 14.447. 全局路径参数

参数	类型	描述
name	字符串	ResourceQuota 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.448. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 ResourceQuota 的读取状态

表 14.449. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 ResourceQuota 的部分更新状态

表 14.450. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.451. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.452. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
201 - Created	ResourceQuota 架构
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 ResourceQuota 的状态

表 14.453. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.454. 主体参数

参数	类型	描述
正文 (body)	ResourceQuota 架构	

表 14.455. HTTP 响应

HTTP 代码	响应正文
200 - OK	ResourceQuota 架构
201 - Created	ResourceQuota 架构
401 - Unauthorized	空

14.16. SECRET [V1]

描述

secret 包含特定类型的 secret 数据。Data 字段中值的总字节数必须小于 MaxSecretSize 字节。

类型

对象

14.16.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
data	对象 (字符串)	数据包含 secret 数据。每个键都必须包含字母数字字符、'-'、'_' 或 '.'。secret 数据的序列化形式是一个 base64 编码字符串，代表此处的任意（可能非字符串）数据值。描述在 https://tools.ietf.org/html/rfc4648#section-4 中
immutable	布尔值	immutable，如果设为 true，则确保无法更新存储在 Secret 中的数据（只能修改对象元数据）。如果没有设置 true，则可以随时修改该字段。默认为 nil。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
stringData	对象 (字符串)	stringData 允许以字符串形式指定非二进制 secret 数据。它作为一个只写的输入字段提供。所有键和值都会合并到写入时的数据字段中，并覆盖任何现有的值。从 API 读取时，stringData 字段永远不会输出。

属性	类型	描述
type	字符串	用于促进对机密数据的编程处理。 更多信息： https://kubernetes.io/docs/concepts/configuration/secret/#secret-types

14.16.2. API 端点

可用的 API 端点如下：

- **/api/v1/secrets**
 - **GET**: 列出或监视类型为 Secret 的对象
- **/api/v1/watch/secrets**
 - **GET**: 观察单个对 Secret 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/secrets**
 - **DELETE**：删除 Secret 集合
 - **GET**: 列出或监视类型为 Secret 的对象
 - **POST**：创建 Secret
- **/api/v1/watch/namespaces/{namespace}/secrets**
 - **GET**: 观察单个对 Secret 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/secrets/{name}**
 - **DELETE**：删除 Secret
 - **GET**：读取指定的 Secret
 - **PATCH**：部分更新指定的 Secret
 - **PUT**：替换指定的 Secret
- **/api/v1/watch/namespaces/{namespace}/secrets/{name}**
 - **GET**: 观察对类型为 Secret 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

14.16.2.1. /api/v1/secrets

表 14.456. 全局查询参数

参数	类型	描述
----	----	----

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided 'resourceVersion", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 Secret 类型的对象

表 14.457. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecretList 模式
401 - Unauthorized	空

14.16.2.2. /api/v1/watch/secrets

表 14.458. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Secret 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.459. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.16.2.3. /api/v1/namespaces/{namespace}/secrets

表 14.460. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.461. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Secret 集合

表 14.462. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下：</p> <ul style="list-style-type: none"> - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.463. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.464. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 Secret 类型的对象

表 14.465. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.466. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecretList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 Secret

表 14.467. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.468. 主体参数

参数	类型	描述
正文 (body)	Secret 模式	

表 14.469. HTTP 响应

HTTP 代码	响应正文
200 - OK	Secret 模式
201 - Created	Secret 模式
202 - Accepted	Secret 模式
401 - Unauthorized	空

14.16.2.4. /api/v1/watch/namespaces/{namespace}/secrets

表 14.470. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.471. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Secret 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.472. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.16.2.5. /api/v1/namespaces/{namespace}/secrets/{name}

表 14.473. 全局路径参数

参数	类型	描述
name	string	Secret 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.474. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Secret

表 14.475. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.476. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.477. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 Secret

表 14.478. HTTP 响应

HTTP 代码	响应正文
200 - OK	Secret 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 Secret

表 14.479. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.480. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.481. HTTP 响应

HTTP 代码	响应正文
200 - OK	Secret 模式
201 - Created	Secret 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 Secret

表 14.482. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.483. 主体参数

参数	类型	描述
正文 (body)	Secret 模式	

表 14.484. HTTP 响应

HTTP 代码	响应正文
200 - OK	Secret 模式
201 - Created	Secret 模式
401 - Unauthorized	空

14.16.2.6. /api/v1/watch/namespaces/{namespace}/secrets/{name}

表 14.485. 全局路径参数

参数	类型	描述
name	string	Secret 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.486. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Secret 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.487. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.17. SERVICE [V1]

描述

服务是软件服务的命名抽象（例如，mysql）由本地端口（如 3306）组成，以及代理侦听的选择器，决定哪些 pod 将回答通过代理发送的请求。

类型

对象

14.17.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	ServiceSpec 描述了用户在服务中创建的属性。
status	对象	ServiceStatus 代表服务的当前状态。

14.17.1.1. .spec

描述

ServiceSpec 描述了用户在服务中创建的属性。

类型

object

属性	类型	描述
allocateLoadBalancerNodePorts	布尔值	allocateLoadBalancerNodePorts 定义是否为类型为 LoadBalancer 的服务自动分配 NodePort。默认为 "true"。如果集群负载均衡器不依赖于 NodePort，它可以设置为 "false"。如果调用者请求特定的 NodePort（通过指定值），无论此字段是什么，这些请求都会被遵守。此字段只能为类型为 LoadBalancer 的服务设置，并在类型改为任何其他类型时清除。
clusterIP	字符串	ClusterIP 是服务的 IP 地址，通常随机分配。如果手动指定地址，是范围为 in-range（根据每个系统配置），且不使用，则它将分配给该服务；否则，服务创建将失败。无法通过更新更改此字段，除非 type 字段也改为 ExternalName（这需要此字段为空），或者将 type 字段从 ExternalName 更改（在这种情况下，可以选择性地指定此字段，如上面描述）。有效值为 "None"、空字符串("")或有效的 IP 地址。把它设置为 "None" 会导致"无头服务"（没有虚拟 IP），这在首选端点连接时很有用，且不需要代理。只适用于 ClusterIP、NodePort 和 LoadBalancer 类型。如果在创建类型为 ExternalName 的 Service 时指定了此字段，则创建将失败。当更新一个 Service 以键入 ExternalName 时，此字段将被擦除。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies

属性	类型	描述
clusterIPs	数组（字符串）	<p>clusterIP 是分配给此服务的 IP 地址列表，通常随机分配。如果手动指定地址，是范围为 in-range（根据每个系统配置），且不使用，则它将分配给该服务；否则，服务创建将失败。无法通过更新更改此字段，除非 type 字段也改为 ExternalName（这需要此字段为空）或 type 字段从 ExternalName 中更改（在这种情况下，可以选择性地指定此字段，如上面描述）。有效值为 "None"、空字符串("")或有效的 IP 地址。把它设置为 "None" 会导致"无头服务"（没有虚拟 IP），这在首选端点连接时很有用，且不需要代理。只适用于 ClusterIP、NodePort 和 LoadBalancer 类型。如果在创建类型为 ExternalName 的 Service 时指定了此字段，则创建将失败。当更新一个 Service 以键入 ExternalName 时，此字段将被擦除。如果没有指定此字段，它将从 clusterIP 字段初始化。如果指定了此字段，客户端必须确保 clusterIPs[0] 和 clusterIP 的值相同。</p> <p>此字段最多可保留两个条目(dual-stack IP, 以任一顺序排列)。这些 IP 必须与 ipFamilies 字段的值对应。clusterIP 和 ipFamilies 都由 ipFamilyPolicy 字段管理。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies</p>
externalIPs	数组（字符串）	<p>externalIPs 是 IP 地址列表，集群中的节点也会接受此服务的流量。这些 IP 不由 Kubernetes 管理。用户负责确保流量通过这个 IP 到达节点。一个常见示例是不是 Kubernetes 系统一部分的外部负载均衡器。</p>

属性	类型	描述
externalName	字符串	ExternalName 是发现机制将返回到此服务的别名（例如 DNS CNAME 记录）的外部引用。不会涉及代理。必须是小写的 RFC-1123 主机名 (https://tools.ietf.org/html/rfc1123)，且需要类型为 "ExternalName"。
externalTrafficPolicy	string	ExternalTrafficPolicy 描述了节点如何在服务"面向外部的"地址 (NodePort、ExternalIP 和 LoadBalancer IP)上分发服务流量。如果设置为 "Local"，代理将以一种方式配置该服务，假设外部负载均衡器在节点间平衡服务流量，因此每个节点将仅向服务的节点本地端点发送流量，而不伪装客户端源 IP。（流量错误地发送到没有端点的节点。）默认值 "Cluster" 使用标准路由到所有端点的标准行为（可能由拓扑和其他功能修改）。请注意，从集群内部发送到外部 IP 或 LoadBalancer IP 的流量始终会获得 "Cluster" 语义，但从集群中发送到 NodePort 的客户端可能需要在选择节点时考虑流量策略。 可能枚举值： - "Cluster" - "Cluster" 将流量路由到所有端点。 - "Local" - "Local" 通过只路由到与接收流量的同一节点上的端点来保留流量的源 IP（如果没有本地端点，则过滤流量）。

属性	类型	描述
healthCheckNodePort	整数	HealthCheckNodePort 指定服务的健康检查 nodePort。这只有在 type 设置为 LoadBalancer 时适用，并将 externalTrafficPolicy 设置为 Local。如果指定了值，则代表范围为 in-range，且不使用，则会使用它。如果没有指定，则会自动分配一个值。外部系统（如负载均衡器）可以使用此端口来确定给定节点是否包含此服务的端点。如果在创建不需要的服务时指定了此字段，则创建将失败。当更新一个服务不再需要时（例如，更改类型），此字段将会被擦除。设置后，无法更新此字段。
internalTrafficPolicy	string	internalTrafficPolicy 描述节点如何在 ClusterIP 上分发它们接收的服务流量。如果设置为 "Local"，代理将假设 pod 只想与 pod 相同的节点上的服务端点进行通信，如果没有本地端点，则丢弃流量。默认值 "Cluster" 使用标准路由到所有端点的标准行为（可能由拓扑和其他功能修改）。 可能枚举值：- "Cluster" 将流量路由到所有端点。- "Local" 将流量仅路由到与客户端 pod 相同的节点上的端点（如果没有本地端点，则过滤流量）。

属性	类型	描述
ipFamilies	数组（字符串）	<p>ipFamilies 是分配给此服务的 IP 系列（如 IPv4、IPv6）的列表。此字段通常根据集群配置和 ipFamilyPolicy 字段自动分配。如果手动指定此字段，则请求中的系列位于集群中，而 ipFamilyPolicy 允许它，它将被使用；否则创建服务将失败。此字段有条件的可变：它允许添加或删除次要 IP 系列，但它不允许更改服务的主 IP 系列。有效值为 "IPv4" 和 "IPv6"。此字段只适用于 ClusterIP、NodePort 和 LoadBalancer 类型的 Services，并应用到"无头"服务。当更新一个 Service 以键入 ExternalName 时，此字段将被擦除。</p> <p>此字段最多可保留两个条目(dual-stack 系列，按任一顺序排列)。如果指定，这些系列必须与 clusterIPs 字段的值对应。clusterIP 和 ipFamilies 都由 ipFamilyPolicy 字段管理。</p>

属性	类型	描述
ipFamilyPolicy	string	<p>ipFamilyPolicy 代表这个服务请求的双栈性。如果没有提供值，则此字段将设置为 SingleStack。服务可以是"SingleStack"（一个 IP 系列）、"PreferDualStack"（双栈上配置集群的两个 IP 系列或单堆栈集群上的一个 IP 系列）或"RequireDualStack"（双栈配置的集群中有两个 IP 系列），否则失败。ipFamilies 和 clusterIPs 字段取决于此字段的值。当更新服务以键入 ExternalName 时，此字段将被擦除。</p> <p>可能枚举值：-</p> <p>"PreferDualStack" 表示当为双栈配置集群时，此服务首选双栈。如果没有为双栈配置集群，则会为该服务分配一个 IPFamily。如果在 service.spec.ipFamilies 中未设置 IPFamily，则会为该服务分配集群中配置的默认 IPFamily -</p> <p>"RequireDualStack" 表示该服务需要双栈。在单个堆栈集群中使用 IPFamilyPolicyRequireDualStack 将导致验证错误。分配给此服务的 IPFamilies（及其顺序）基于 service.spec.ipFamilies。如果没有提供 service.spec.ipFamilies，则它将根据集群上配置的方式进行分配。如果 service.spec.ipFamilies 只有一个条目，则替代的 IPFamily 将由 apiserver - "SingleStack" 添加，表示这个服务需要具有单个 IPFamily。分配的 IPFamily 基于集群使用的默认 IPFamily，或者由 service.spec.ipFamilies 字段标识</p>

属性	类型	描述
loadBalancerClass	字符串	loadBalancerClass 是此服务所属的负载均衡器实现的类。如果指定，此字段的值必须是标签样式的标识符，其是可选的前缀，如 "internal-vip" 或 "example.com/internal-vip"。Unprefixed name 为最终用户保留。只有 Service type 为 'LoadBalancer' 时，才能设置此字段。如果没有设置，则使用默认的负载均衡器实现，现在这通常通过云提供商集成完成，但应适用于任何默认实现。如果设置，则假定负载均衡器的实现正在监视具有匹配类的服务。任何默认的负载均衡器实现（如云供应商）都应忽略设置此字段的 Service。此字段只能在创建或更新 Service 以类型 'LoadBalancer' 时设置。设置之后，它就无法更改。当服务已更新至非 "LoadBalancer" 类型时，此字段将被擦除。
loadBalancerIP	string	只适用于 Service Type: LoadBalancer。此功能取决于底层的 cloud-provider 是否支持在创建负载均衡器时指定 loadBalancerIP。如果 cloud-provider 不支持该功能，则此字段将被忽略。deprecated：此字段已被指定，其含义因实现而异，它不能支持双栈。从 Kubernetes v1.24 开始，建议用户在可用时使用特定于实现的注解。此字段可能在以后的 API 版本中删除。
loadBalancerSourceRanges	数组（字符串）	如果平台指定和支持，这将限制通过 cloud-provider 负载均衡器的流量将限制为指定的客户端 IP。如果 cloud-provider 不支持该功能，则此字段将被忽略。更多信息： https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/

属性	类型	描述
ports	array	此服务公开的端口列表。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies
ports[]	对象	ServicePort 包含有关服务端口的信息。
publishNotReadyAddresses	布尔值	publishNotReadyAddresses 表示任何处理此服务端点的代理都应忽略任何 ready/not-ready 的代表。设置此字段的主要用例是 StatefulSet 的无头服务为其 Pod 传播 SRV DNS 记录，以满足对等发现的目的。为服务生成 Endpoints 和 EndpointSlice 资源的 Kubernetes 控制器会解释这一点，这意味着即使 Pod 本身没有，所有端点也被视为"就绪"。只有通过 Endpoints 或 EndpointSlice 资源使用 Kubernetes 生成的端点的代理可以安全地假设此行为。
selector	对象 (字符串)	将服务流量路由到具有与此选择器匹配的标签键和值的 pod。如果为空或不存在，则假定服务具有管理其端点的外部进程，Kubernetes 将不修改。只适用于 ClusterIP、NodePort 和 LoadBalancer 类型。如果 type 是 ExternalName，则忽略。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/
sessionAffinity	字符串	支持 "ClientIP" 和 "None"。用于维护会话关联。启用基于客户端 IP 的会话关联性。必须是 ClientIP 或 None。默认值为 None。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies 可能枚举值：- "ClientIP" 是基于客户端 IP。- "None" - 无会话关联。

属性	类型	描述
sessionAffinityConfig	对象	sessionAffinityConfig 代表会话关联性的配置。
type	字符串	<p>Type 决定服务是如何公开的。默认为 ClusterIP。有效选项包括 ExternalName、ClusterIP、NodePort 和 LoadBalancer。</p> <p>"ClusterIP" 为负载均衡端点分配集群内部 IP 地址。端点由选择器决定，或者未指定端点，方法是手动构建 Endpoints 对象或 EndpointSlice 对象。如果 clusterIP 是 "None"，则不会分配虚拟 IP，端点作为一组端点而不是虚拟 IP 发布。"NodePort" 基于 ClusterIP 构建，并在每个节点上分配一个端口，该端口路由到与 clusterIP 相同的端点。</p> <p>"LoadBalancer" 基于 NodePort 构建，并创建一个外部负载均衡器（如果在当前云中受支持），后者路由到与 clusterIP 相同的端点。</p> <p>"ExternalName" 将该服务别名化到指定的 externalName。一些其他字段不适用于 ExternalName 服务。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types</p> <p>可能枚举值： - "ClusterIP" 意味着一个服务只能在集群中访问，通过集群 IP 访问。 - "ExternalName" 表示服务只包括对 kubernetes 或 equivalent 作为外部名称的引用，且无公开或代理任何涉及的 pod。 - "LoadBalancer" 表示服务将通过外部负载均衡器（如果云提供商支持）公开。除了 'NodePort' 类型外， - "NodePort" 意味着除了 'ClusterIP' 类型外，还会在每个节点的一个端口上公开服务。</p>

14.17.1.2. .spec.ports

描述

此服务公开的端口列表。更多信息：<https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

类型

array**14.17.1.3. .spec.ports[]****描述**

ServicePort 包含有关服务端口的信息。

类型

object

必填

- **port**

属性	类型	描述
appProtocol	字符串	此端口的应用程序协议。此字段遵循标准 Kubernetes 标签语法。未命名的名称用于 IANA 标准服务名称（根据 RFC-6335 和 https://www.iana.org/assignments/service-names ）。非标准协议应使用前缀名称，如 mycompany.com/my-custom-protocol。
name	字符串	该服务中的此端口的名称。这必须是 DNS_LABEL。ServiceSpec 中的所有端口必须具有唯一的名称。在考虑服务的端点时，这必须与 EndpointPort 中的 'name' 字段匹配。如果此服务上只定义一个 ServicePort，则可选。
nodePort	整数	在类型 NodePort 或 LoadBalancer 时公开此服务的每个节点上的端口。通常由系统分配。如果指定了值(in-range)，并且不使用它，则将使用它，否则操作将失败。如果没有指定，如果这个服务需要端口，则会分配一个端口。如果在创建不需要的服务时指定了此字段，则创建将失败。当更新服务不再需要时，此字段将擦除（例如，将类型从 NodePort 更改为 ClusterIP）。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport
port	整数	此服务公开的端口。

属性	类型	描述
protocol	字符串	此端口的 IP 协议。支持"TCP"、"UDP"和"SCTP"。默认为 TCP。 可能的值有： - "SCTP" 是 SCTP 协议。 - "TCP" 是 TCP 协议。 - "UDP" 是 UDP 协议。
targetPort	IntOrString	服务目标 pod 上要访问的端口的数量或名称。数字必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。如果这是字符串，它将在目标 Pod 的容器端口中作为命名端口查找。如果没有指定，则使用 'port' 字段的值（身份映射）。对于带有 clusterIP=None 的服务，会忽略此字段，应该省略或设置为 "port" 字段。更多信息： https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service

14.17.1.4. .spec.sessionAffinityConfig

描述

sessionAffinityConfig 代表会话关联性的配置。

类型

object

属性	类型	描述
clientIP	对象	ClientIPConfig 代表基于客户端 IP 的会话关联性的配置。

14.17.1.5. .spec.sessionAffinityConfig.clientIP

描述

ClientIPConfig 代表基于客户端 IP 的会话关联性的配置。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
timeoutSeconds	整数	timeoutSeconds 指定 ClientIP 类型会话粘性时间的秒数。如果 ServiceAffinity == "ClientIP", 则该值必须是 >0 && mvapich86400 (for 1 day)。默认值为 10800 (3 小时)。

14.17.1.6. .status

描述

ServiceStatus 代表服务的当前状态。

类型

object

属性	类型	描述
conditions	数组(Condition)	当前服务状态
loadBalancer	对象	LoadBalancerStatus 代表负载均衡器的状态。

14.17.1.7. .status.loadBalancer

描述

LoadBalancerStatus 代表负载均衡器的状态。

类型

object

属性	类型	描述
ingress	array	Ingress 是一个包含负载均衡器入口点的列表。适合服务的流量应发送到这些入口点。
ingress[]	对象	LoadBalancerIngress 代表负载均衡器入口点的状态：用于服务的流量应发送到入口点。

14.17.1.8. .status.loadBalancer.ingress

描述

Ingress 是一个包含负载均衡器入口点的列表。适合服务的流量应发送到这些入口点。

类型

array

14.17.1.9. `.status.loadBalancer.ingress[]`

描述

`LoadBalancerIngress` 代表负载均衡器入口点的状态：用于服务的流量应发送到入口点。

类型

object

属性	类型	描述
<code>hostname</code>	字符串	为基于 DNS 的负载均衡器入口点设置主机名（通常是 AWS 负载均衡器）
<code>ip</code>	字符串	为基于 IP 的负载均衡器入口点设置 IP（通常是 GCE 或 OpenStack 负载均衡器）
<code>ports</code>	array	<code>port</code> 是服务端口的记录列表（如果使用），服务中定义的每个端口都应该有一个条目
<code>ports[]</code>	对象	

14.17.1.10. `.status.loadBalancer.ingress[].ports`

描述

`port` 是服务端口的记录列表（如果使用），服务中定义的每个端口都应该有一个条目

类型

array

14.17.1.11. `.status.loadBalancer.ingress[].ports[]`

描述

类型

object

必填

- `port`
- `protocol`

属性	类型	描述
----	----	----

属性	类型	描述
错误	字符串	错误是记录服务端口的问题。错误的格式应遵循以下规则：- 在此文件中指定内置错误值，它们应使用 CamelCase 名称 - 云供应商特定错误值必须符合 foo.example.com/CamelCase 格式的名称。
port	整数	port 是记录状态的服务端口的端口号
protocol	字符串	protocol 是记录状态的服务端口的协议，这里支持的值有："TCP", "UDP", "SCTP" 可能的值有：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

14.17.2. API 端点

可用的 API 端点如下：

- **/api/v1/services**
 - **GET**: 列出或监视类型为 Service 的对象
- **/api/v1/watch/services**
 - **GET**: 观察单个对 Service 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/services**
 - **DELETE** : 删除服务集合
 - **GET**: 列出或监视类型为 Service 的对象
 - **POST** : 创建服务
- **/api/v1/watch/namespaces/{namespace}/services**
 - **GET**: 观察单个对 Service 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/services/{name}**
 - **DELETE** : 删除服务
 - **GET** : 读取指定的服务
 - **PATCH** : 部分更新指定的服务
 - **PUT** : 替换指定的服务

- **/api/v1/watch/namespaces/{namespace}/services/{name}**
 - **GET**: 观察对类型为 Service 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/api/v1/namespaces/{namespace}/services/{name}/status**
 - **GET**: 指定服务的读取状态
 - **PATCH** : 部分更新指定服务的状态
 - **PUT** : 替换指定服务的状态

14.17.2.1. /api/v1/services

表 14.488. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Service 的对象

表 14.489. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceList 模式
401 - Unauthorized	空

14.17.2.2. /api/v1/watch/services

表 14.490. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Service 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.491. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.17.2.3. /api/v1/namespaces/{namespace}/services

表 14.492. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.493. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE**描述**

删除服务集合

表 14.494. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.495. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.496. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Service 的对象

表 14.497. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.498. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建服务

表 14.499. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.500. 主体参数

参数	类型	描述
正文 (body)	Service 模式	

表 14.501. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
201 - Created	Service 模式
202 - Accepted	Service 模式
401 - Unauthorized	空

14.17.2.4. /api/v1/watch/namespaces/{namespace}/services

表 14.502. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.503. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Service 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.504. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.17.2.5. /api/v1/namespaces/{namespace}/services/{name}

表 14.505. 全局路径参数

参数	类型	描述
name	字符串	服务的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.506. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除服务

表 14.507. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.508. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.509. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Service</code> 模式
202 - Accepted	<code>Service</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的服务

表 14.510. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Service</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的服务

表 14.511. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.512. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.513. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
201 - Created	Service 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 Service

表 14.514. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.515. 主体参数

参数	类型	描述
正文 (body)	Service 模式	

表 14.516. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
201 - Created	Service 模式
401 - Unauthorized	空

14.17.2.6. /api/v1/watch/namespaces/{namespace}/services/{name}

表 14.517. 全局路径参数

参数	类型	描述
name	字符串	服务的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.518. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Service 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.519. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.17.2.7. /api/v1/namespaces/{namespace}/services/{name}/status

表 14.520. 全局路径参数

参数	类型	描述
name	字符串	服务的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.521. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定服务的读取状态

表 14.522. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定服务的部分更新状态

表 14.523. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.524. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.525. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
201 - Created	Service 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定服务的状态

表 14.526. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.527. 主体参数

参数	类型	描述
正文 (body)	Service 模式	

表 14.528. HTTP 响应

HTTP 代码	响应正文
200 - OK	Service 模式
201 - Created	Service 模式
401 - Unauthorized	空

14.18. SERVICEACCOUNT [V1]

描述

ServiceAccount 将一个名称绑定在一起：一个名称，用户理解，可能由外部系统理解，对于一个身份，可以验证并授权了一组 secret 的主体。

类型

对象

14.18.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 指示作为此服务帐户运行的 pod 是否应该自动挂载 API 令牌。可以在 pod 级别上覆盖。
imagePullSecrets	数组	imagePullSecrets 是同一命名空间中用于拉取引用此 ServiceAccount 的 pod 中的 secret 的引用列表。 imagePullSecrets 与 Secret 不同，因为 Secret 可以挂载到 pod 中，但 ImagePullSecrets 只能被 kubelet 访问。更多信息： https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	对象	LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
secrets	数组	secret 是允许使用此 ServiceAccount 运行的 pod 的同一命名空间中的 secret 列表。只有此服务帐户将 "kubernetes.io/enforce-mountable-secrets" 注解设置为 "true" 时，Pod 才会仅限于此列表。此字段不应用于查找在 pod 外部使用自动生成的服务帐户令牌 secret。反之，可以使用 TokenRequest API 直接请求令牌，或者可以手动创建服务帐户令牌 secret。更多信息： https://kubernetes.io/docs/concepts/configuration/secret
secrets[]	object	ObjectReference 包含足够信息供您检查或修改引用的对象。

14.18.1.1. .imagePullSecrets

描述

imagePullSecrets 是同一命名空间中用于拉取引用此 ServiceAccount 的 pod 中的 secret 的引用列表。imagePullSecrets 与 Secret 不同，因为 Secret 可以挂载到 pod 中，但 ImagePullSecrets 只能被 kubelet 访问。更多信息：<https://kubernetes.io/docs/concepts/containers/images/#specifying-imagepullsecrets-on-a-pod>

类型

数组

14.18.1.2. .imagePullSecrets[]

描述

LocalObjectReference 包含足够的信息，供您在同一个命名空间中找到引用的对象。

类型

object

属性	类型	描述
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.18.1.3. .secrets

描述

secret 是允许使用此 ServiceAccount 运行的 pod 的同一命名空间中的 secret 列表。只有此服务帐户

将 "kubernetes.io/enforce-mountable-secrets" 注解设置为 "true" 时, Pod 才会仅限于此列表。此字段不应用于查找在 pod 外部使用自动生成的服务帐户令牌 secret。反之, 可以使用 TokenRequest API 直接请求令牌, 或者可以手动创建服务帐户令牌 secret。更多信息:

<https://kubernetes.io/docs/concepts/configuration/secret>

类型

数组

14.18.1.4. .secrets[]

描述

ObjectReference 包含足够信息供您检查或修改引用的对象。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象, 则该字符串应包含有效的 JSON/Go 字段访问声明, 如 <code>desiredState.manifest.containers[2]</code> 。例如, 如果对象引用是一个 pod 中的容器, 它应该还以下类似: <code>"spec.containers{name}"</code> (其中 "name" 是指触发事件的容器的名称), 如果不使用容器名称, 则应类似于 <code>"spec.containers[2]"</code> (使用此 pod 中索引值为 2 的容器)。选择这个语法的原因是, 为了有一些指向对象部分的正确定义方法。
kind	字符串	引用的类型。更多信息: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

属性	类型	描述
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

14.18.2. API 端点

可用的 API 端点如下：

- **/api/v1/serviceaccounts**
 - **GET**: 列出或监视类型为 ServiceAccount 的对象
- **/api/v1/watch/serviceaccounts**
 - **GET**: 观察单个对 ServiceAccount 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/serviceaccounts**
 - **DELETE**：删除 ServiceAccount 集合
 - **GET**: 列出或监视类型为 ServiceAccount 的对象
 - **POST**：创建一个 ServiceAccount
- **/api/v1/watch/namespaces/{namespace}/serviceaccounts**
 - **GET**: 观察单个对 ServiceAccount 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/api/v1/namespaces/{namespace}/serviceaccounts/{name}**
 - **DELETE**：删除 ServiceAccount
 - **GET**: 读取指定的 ServiceAccount
 - **PATCH**：部分更新指定的 ServiceAccount
 - **PUT**：替换指定的 ServiceAccount
- **/api/v1/watch/namespaces/{namespace}/serviceaccounts/{name}**
 - **GET**: 观察对 ServiceAccount 类型的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

14.18.2.1. /api/v1/serviceaccounts

表 14.529. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 ServiceAccount 的对象

表 14.530. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceAccountList 模式
401 - Unauthorized	空

14.18.2.2. /api/v1/watch/serviceaccounts

表 14.531. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ServiceAccount 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.532. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.18.2.3. /api/v1/namespaces/{namespace}/serviceaccounts

表 14.533. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.534. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE**描述**

删除 ServiceAccount 集合

表 14.535. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 14.536. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 14.537. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ServiceAccount 的对象

表 14.538. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 14.539. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceAccountList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 ServiceAccount

表 14.540. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.541. 主体参数

参数	类型	描述
正文 (body)	ServiceAccount 模式	

表 14.542. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceAccount 模式
201 - Created	ServiceAccount 模式
202 - Accepted	ServiceAccount 模式
401 - Unauthorized	空

14.18.2.4. /api/v1/watch/namespaces/{namespace}/serviceaccounts

表 14.543. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.544. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ServiceAccount 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 14.545. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

14.18.2.5. /api/v1/namespaces/{namespace}/serviceaccounts/{name}

表 14.546. 全局路径参数

参数	类型	描述
name	string	ServiceAccount 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 14.547. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ServiceAccount

表 14.548. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 14.549. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 14.550. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ServiceAccount</code> 模式
202 - Accepted	<code>ServiceAccount</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 `ServiceAccount`

表 14.551. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>ServiceAccount</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `ServiceAccount`

表 14.552. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 14.553. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 14.554. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceAccount 模式
201 - Created	ServiceAccount 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ServiceAccount

表 14.555. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 14.556. 主体参数

参数	类型	描述
正文 (body)	ServiceAccount 模式	

表 14.557. HTTP 响应

HTTP 代码	响应正文
200 - OK	ServiceAccount 模式
201 - Created	ServiceAccount 模式
401 - Unauthorized	空

14.18.2.6. /api/v1/watch/namespaces/{namespace}/serviceaccounts/{name}

表 14.558. 全局路径参数

参数	类型	描述
name	string	ServiceAccount 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 14.559. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind ServiceAccount. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 14.560. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 15 章 DISCOVERY API

15.1. DISCOVERY API

15.1.1. EndpointSlice [discovery.k8s.io/v1]

描述

EndpointSlice 代表实现服务的端点的子集。对于给定服务，可能有多个 EndpointSlice 对象，根据标签选择，该对象必须加入来生成完整的端点集。

类型

对象

15.2. ENDPPOINTSICE [DISCOVERY.K8S.IO/V1]

描述

EndpointSlice 代表实现服务的端点的子集。对于给定服务，可能有多个 EndpointSlice 对象，根据标签选择，该对象必须加入来生成完整的端点集。

类型

object

必填

- **addressType**
- **端点**

15.2.1. 规格

属性	类型	描述
addressType	字符串	<p>addressType 指定此 EndpointSlice 所传输的地址类型。此片段中的所有地址都必须相同类型。此字段在创建后不可变。目前支持以下地址类型：* IPv4: 代表 IPv4 地址。* IPv6: 代表 IPv6 地址。* FQDN: 代表完全限定域名。</p> <p>可能枚举值：- "FQDN" 代表 FQDN。- "IPv4" 代表 IPv4 地址。- "IPv6" 代表 IPv6 地址。</p>

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
端点	array	Endpoints 是此片段中唯一端点的列表。每个片段最多可能包括 1000 个端点。
endpoints[]	对象	端点代表单一逻辑 "backend" 实现服务。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。
ports	array	ports 指定此片段中每个端点公开的网络端口列表。每个端口必须具有唯一的名称。当端口为空时，这表示没有定义的端口。当使用 nil port 值定义端口时，它表示 "all ports"。每个片段最多可能包括 100 个端口。
ports[]	对象	EndpointPort 代表 EndpointSlice 使用的端口

15.2.1.1. .endpoints

描述

Endpoints 是此片段中唯一端点的列表。每个片段最多可能包括 1000 个端点。

类型

array

15.2.1.2. .endpoints[]

描述

端点代表单一逻辑 "backend" 实现服务。

类型

object

必填

- **addresses**

属性	类型	描述
addresses	数组 (字符串)	此端点的地址。此字段的内容根据对应的 EndpointSlice addressType 字段进行解释。消费者必须在自己的功能上下文中处理不同类型的地址。这必须至少包含一个地址，但没有超过 100 个地址。这些都假定是允许的，客户端可以选择只使用第一个元素。请参阅： https://issue.k8s.io/106267
conditions	对象	EndpointConditions 代表端点的当前条件。
deprecatedTopology	对象 (字符串)	deprecatedTopology 包含 v1beta1 API 的拓扑信息部分。此字段已弃用，当 v1beta1 API 被删除时（不早于 kubernetes v1.24），将被删除。虽然此字段可以保存值，但无法通过 v1 API 可写入，但任何尝试写入它都会被静默忽略。拓扑信息可在 zone 和 nodeName 字段中找到。
hints	对象	EndpointHints 提供了描述如何消耗端点的提示。
hostname	字符串	此端点的主机名。此字段可供端点使用者使用，以区分端点相互区分（例如在 DNS 名称中）。使用相同的主机名的多个端点应被视为是不可取的（例如，DNS 中的多个 A 值）。必须是小写并传递 DNS 标签(RFC 1123)验证。
nodeName	字符串	nodeName 代表托管此端点的节点名称。这可用于确定节点本地的端点。

属性	类型	描述
targetRef	ObjectReference	TargetRef 是代表此端点的 Kubernetes 对象的引用。
zone	字符串	zone 是存在此端点的 Zone 名称。

15.2.1.3. .endpoints[].conditions

描述

EndpointConditions 代表端点的当前条件。

类型

object

属性	类型	描述
ready	布尔值	Ready 表示此端点已准备好根据管理端点的任何系统接收流量。nil 值表示未知状态。在大多数情况下，消费者应将此未知状态解析为 ready。出于兼容性的原因，对于终止端点，ready 不应该为"true"，除非明确覆盖正常就绪度行为，例如当关联的服务设置了 publishNotReadyAddresses 标志时。
serving	布尔值	服务与 ready 相同，除非它是设置的，无论端点的终止状态如何。对于正在终止的就绪端点，此条件应设为 true。如果为 nil，消费者应延迟就绪条件。
Terminating	布尔值	Terminating 表示此端点正在终止。nil 值表示未知状态。消费者应解释这个未知状态，这意味着端点没有终止。

15.2.1.4. .endpoints[].hints

描述

EndpointHints 提供了描述如何消耗端点的提示。

类型

object

属性	类型	描述
forZones	array	forZones 表示此端点应该被使用的区域来启用拓扑感知路由。
forZones[]	对象	ForZone 提供有关哪些区域应使用此端点的信息。

15.2.1.5. .endpoints[].hints.forZones

描述

forZones 表示此端点应该被使用的区域来启用拓扑感知路由。

类型

array

15.2.1.6. .endpoints[].hints.forZones[]

描述

ForZone 提供有关哪些区域应使用此端点的信息。

类型

object

必填

- **name**

属性	类型	描述
name	字符串	name 表示区域的名称。

15.2.1.7. .ports

描述

ports 指定此片段中每个端点公开的网络端口列表。每个端口必须具有唯一的名称。当端口为空时，这表示没有定义的端口。当使用 nil port 值定义端口时，它表示 "all ports"。每个片段最多可能包括 100 个端口。

类型

array

15.2.1.8. .ports[]

描述

EndpointPort 代表 EndpointSlice 使用的端口

类型

object

属性	类型	描述
appProtocol	字符串	<p>此端口的应用程序协议。这用作实现的提示，以便为他们理解的协议提供更丰富的行为。此字段遵循标准 Kubernetes 标签语法。有效值为：</p> <ul style="list-style-type: none"> * 无前缀协议名称 - 为 IANA 标准服务名称保留（根据 RFC-6335 和 https://www.iana.org/assignments/service-names保留）。 * Kubernetes-defined prefixed name: * 'kubernetes.io/h2c' - HTTP/2 over cleartext, 如 https://www.rfc-editor.org/rfc/rfc7540所述 * 其他协议应使用实现定义的前缀名称，如 mycompany.com/my-custom-protocol。
name	string	<p>name 表示此端口的名称。EndpointSlice 中的所有端口必须具有唯一的名称。如果 EndpointSlice 从 Kubernetes 服务派生，这对应于 Service.ports[].name。name 必须是空字符串或传递 DNS_LABEL 验证：* 不得超过 63 个字符。* 必须包含小写字母数字字符或 '-'。* 必须以字母数字字符开头和结尾。默认为空字符串。</p>
port	整数	<p>port 代表端口的端口号。如果没有指定，端口不会被限制，且必须在特定消费者的上下文中解释。</p>
protocol	string	<p>protocol 代表此端口的 IP 协议。必须是 UDP、TCP 或 SCTP。默认为 TCP。</p> <p>可能的值有： - "SCTP" 是 SCTP 协议。 - "TCP" 是 TCP 协议。 - "UDP" 是 UDP 协议。</p>

15.2.2. API 端点

可用的 API 端点如下：

- `/apis/discovery.k8s.io/v1/endpointlices`

• **SCTP** 列出来监视类型为 `SCTP` 的对象

- **GET**: 列出或监视类型为 EndpointSlice 的对象
- **/apis/discovery.k8s.io/v1/watch/endpointslices**
 - **GET**: 观察单个对 EndpointSlice 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices**
 - **DELETE** : 删除 EndpointSlice 的集合
 - **GET**: 列出或监视类型为 EndpointSlice 的对象
 - **POST** : 创建 EndpointSlice
- **/apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices**
 - **GET**: 观察单个对 EndpointSlice 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}**
 - **DELETE** : 删除 EndpointSlice
 - **GET** : 阅读指定的 EndpointSlice
 - **PATCH** : 部分更新指定的 EndpointSlice
 - **PUT** : 替换指定的 EndpointSlice
- **/apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices/{name}**
 - **GET** : 观察对类型为 EndpointSlice. deprecated 的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

15.2.2.1. /apis/discovery.k8s.io/v1/endpointslices

表 15.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 EndpointSlice 的对象

表 15.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointSliceList schema
401 - Unauthorized	空

15.2.2.2. /apis/discovery.k8s.io/v1/watch/endpointlices

表 15.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 EndpointSlice 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 15.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

15.2.2.3. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices

表 15.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 15.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 EndpointSlice 集合

表 15.7. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 15.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 15.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 EndpointSlice 的对象

表 15.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 15.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointSliceList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 EndpointSlice

表 15.12. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 15.13. 主体参数

参数	类型	描述
正文 (body)	EndpointSlice 模式	

表 15.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointSlice 模式
201 - Created	EndpointSlice 模式
202 - Accepted	EndpointSlice 模式
401 - Unauthorized	空

15.2.2.4. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices

表 15.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 15.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 EndpointSlice 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 15.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

15.2.2.5. /apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices/{name}

表 15.18. 全局路径参数

参数	类型	描述
name	字符串	EndpointSlice 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 15.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 EndpointSlice

表 15.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 15.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 15.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 EndpointSlice

表 15.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>EndpointSlice</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 EndpointSlice

表 15.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 15.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 15.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointSlice 模式
201 - Created	EndpointSlice 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 EndpointSlice

表 15.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 15.28. 主体参数

参数	类型	描述
正文 (body)	EndpointSlice 模式	

表 15.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	EndpointSlice 模式
201 - Created	EndpointSlice 模式
401 - Unauthorized	空

15.2.2.6. /apis/discovery.k8s.io/v1/watch/namespaces/{namespace}/endpointslices/{name}

表 15.30. 全局路径参数

参数	类型	描述
name	字符串	EndpointSlice 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 15.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 EndpointSlice. deprecated 对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 15.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 16 章 EVENTS API

16.1. EVENTS API

16.1.1. Event [events.k8s.io/v1]

描述

事件是集群中某一位置的事件报告。它通常表示系统中的一些状态更改。事件具有有限的保留时间和触发器，消息可能会随时间而演进。事件消费者不应依赖给定 Reason 反映一致的底层触发器的事件的时间，或者该 Reason 持续存在事件的时间。事件应被视为 informative, best-effort, supplemental 数据。

类型

对象

16.2. EVENT [EVENTS.K8S.IO/V1]

描述

事件是集群中某一位置的事件报告。它通常表示系统中的一些状态更改。事件具有有限的保留时间和触发器，消息可能会随时间而演进。事件消费者不应依赖给定 Reason 反映一致的底层触发器的事件的时间，或者该 Reason 持续存在事件的时间。事件应被视为 informative, best-effort, supplemental 数据。

类型

object

必填

- `eventTime`

16.2.1. 规格

属性	类型	描述
<code>action</code>	字符串	关于相关对象采取的操作/失败的操作。它是机器可读。对于新事件，此字段不能为空，且最多可有 128 个字符。
<code>apiVersion</code>	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
deprecatedCount	整数	deprecatedCount 是已弃用的字段，与 core.v1 事件类型向后兼容性。
deprecatedFirstTimestamp	时间	deprecatedFirstTimestamp 是已弃用的字段，与 core.v1 事件类型向后兼容性。
deprecatedLastTimestamp	时间	deprecatedLastTimestamp 是已弃用的字段，与 core.v1 事件类型向后兼容性。
deprecatedSource	EventSource	deprecatedSource 是已弃用的字段，与 core.v1 事件类型向后兼容性。
eventTime	MicroTime	eventTime 是第一次观察此事件的时间。它是必需的。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
备注：	字符串	注意 是此操作状态的人类可读描述。注意的最大长度为 1kB，但应该准备处理最多 64kB 的值。
reason	字符串	原因就是采取该操作的原因。它是人类可读的。对于新事件，此字段不能为空，且最多可有 128 个字符。

属性	类型	描述
关于	ObjectReference	关于包含此事件所针对的对象。在大多数情况下，对象报告控制器实现，如 <code>ReplicaSetController</code> 实现 <code>ReplicaSet</code> ，并发出此事件，因为它操作 <code>ReplicaSet</code> 对象中的一些更改。
相关	ObjectReference	相关的是可选二级对象，用于更复杂的操作。例如，当有关对象触发创建或删除相关对象时。
<code>reportingController</code>	字符串	<code>reportingController</code> 是发出此事件的控制器的名称，如 kubernetes.io/kubelet 。对于新事件，此字段不能为空。
<code>reportingInstance</code>	字符串	<code>reportingInstance</code> 是控制器实例的 ID，如 kubelet-xyzf 。对于新事件，此字段不能为空，且最多可有 128 个字符。
系列	对象	<code>EventSeries</code> 包含有关一系列事件的信息，即在一段时间内持续发生/发生的内容。更新事件Series 的频率取决于事件报告者。 "k8s.io/client-go/tools/events/event_broadcaster.go" 中的默认事件报告器显示如何在 heartbeat 中更新这个 struct，并可以指导自定义的报告程序实现。
<code>type</code>	字符串	<code>type</code> 是此事件的类型(<code>Normal</code> , <code>Warning</code>)，将来可能会添加新类型。它是机器可读。对于新事件，此字段不能为空。

16.2.1.1. .series

描述

`EventSeries` 包含有关一系列事件的信息，即在一段时间内持续发生/发生的内容。更新事件Series 的频率取决于事件报告者。"k8s.io/client-go/tools/events/event_broadcaster.go" 中的默认事件报告器显示如何在 heartbeat 中更新这个 struct，并可以指导自定义的报告程序实现。

类型

object

必填

- 数量
- lastObservedTime

属性	类型	描述
数量	整数	count 是此系列中发生次数，最高为最后的心跳时间。
lastObservedTime	MicroTime	lastObservedTime 是最后 heartbeat 之前看到系列的最后一个事件的时间。

16.2.2. API 端点

可用的 API 端点如下：

- **/apis/events.k8s.io/v1/events**
 - **GET**: 列出或监视类型为 Event 的对象
- **/apis/events.k8s.io/v1/watch/events**
 - **GET**: 观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/events.k8s.io/v1/namespaces/{namespace}/events**
 - **DELETE** : 删除事件集合
 - **GET**: 列出或监视类型为 Event 的对象
 - **POST** : 创建事件
- **/apis/events.k8s.io/v1/watch/namespaces/{namespace}/events**
 - **GET**: 观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}**
 - **DELETE** : 删除事件
 - **GET** : 读取指定的事件
 - **PATCH** : 部分更新指定的事件
 - **PUT** : 替换指定的事件
- **/apis/events.k8s.io/v1/watch/namespaces/{namespace}/events/{name}**
 - **GET**: 观察对类型为 Event 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

16.2.2.1. /apis/events.k8s.io/v1/events

表 16.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Event 的对象

表 16.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	EventList 模式
401 - Unauthorized	空

16.2.2.2. /apis/events.k8s.io/v1/watch/events

表 16.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 16.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

16.2.2.3. /apis/events.k8s.io/v1/namespaces/{namespace}/events

表 16.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 16.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除事件集合

表 16.7. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 16.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 16.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Event 的对象

表 16.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 16.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	EventList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建事件

表 16.12. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	<code>fieldValidation</code> 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 <code>BadRequest</code> 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 16.13. 主体参数

参数	类型	描述
正文 (body)	event 模式	

表 16.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
202 - Accepted	event 模式
401 - Unauthorized	空

16.2.2.4. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events

表 16.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 16.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对事件列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 16.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

16.2.2.5. /apis/events.k8s.io/v1/namespaces/{namespace}/events/{name}

表 16.18. 全局路径参数

参数	类型	描述
name	字符串	事件名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 16.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除事件

表 16.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 16.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 16.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的事件

表 16.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>event</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的事件

表 16.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 16.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 16.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的事件

表 16.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 16.28. 主体参数

参数	类型	描述
正文 (body)	event 模式	

表 16.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	event 模式
201 - Created	event 模式
401 - Unauthorized	空

16.2.2.6. /apis/events.k8s.io/v1/watch/namespaces/{namespace}/events/{name}

表 16.30. 全局路径参数

参数	类型	描述
name	字符串	事件名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 16.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Event.deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 16.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 17 章 流控制 API

17.1. 流控制 API

17.1.1. FlowSchema [flowcontrol.apiserver.k8s.io/v1beta3]

描述

FlowSchema 定义一组流的 schema。请注意，流由一组具有类似属性的入站 API 请求组成，并由字符串对字符串识别：FlowSchema 的名称和 "flow distinguisher"。

类型

object

17.1.2. PriorityLevelConfiguration [flowcontrol.apiserver.k8s.io/v1beta3]

描述

priorityLevelConfiguration 代表优先级级别的配置。

类型

object

17.2. FLOWSCHEMA [FLOWCONTROL.APISERVER.K8S.IO/V1BETA3]

描述

FlowSchema 定义一组流的 schema。请注意，流由一组具有类似属性的入站 API 请求组成，并由字符串对字符串识别：FlowSchema 的名称和 "flow distinguisher"。

类型

对象

17.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata 是标准对象的元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	FlowSchemaSpec 描述了 FlowSchema 的规格如何。
status	对象	FlowSchemaStatus 代表 FlowSchema 的当前状态。

17.2.1.1. .spec

描述

FlowSchemaSpec 描述了 FlowSchema 的规格如何。

类型

object

必填

- **priorityLevelConfiguration**

属性	类型	描述
distinguisherMethod	对象	FlowDistinguisherMethod 指定流区分器的方法。

属性	类型	描述
matchingPrecedence	整数	matchingPrecedence 用于选择与给定请求匹配的 FlowSchema。所选的 FlowSchema 是那些具有数字最低（我们需要逻辑最高的） MatchingPrecedence。每个 MatchingPrecedence 值都必须以 [1,10000] 范围。请注意，如果没有指定优先级，它将被设置为 1000 作为默认值。
priorityLevelConfiguration	对象	PriorityLevelConfigurationReference 包含指向正在使用的 "request-priority" 的信息。
rules	array	规则 描述了哪些请求与此流模式匹配。只有至少有一个规则成员与请求匹配时，此 FlowSchema 才会匹配请求。如果它是一个空片段，则没有与 FlowSchema 匹配的请求。
rules[]	对象	PolicyRulesWithSubjects 假定一个测试适用于一个 apiserver 的请求。测试会考虑主题进行请求、请求的动词以及要操作的资源。此 PolicyRulesWithSubjects 只在同时 (a)至少一个主体成员与 request 和 (b)至少匹配请求时匹配一个请求，或者(b)至少匹配 resourceRules 或 nonResourceRules 的请求。

17.2.1.2. .spec.distinguisherMethod

描述

FlowDistinguisherMethod 指定流区分器的方法。

类型

object

必填

- **type**

属性	类型	描述
----	----	----

属性	类型	描述
type	字符串	type 是流区分器方法的类型。支持的类型是 "ByUser" 和 "ByNamespace"。必需。

17.2.1.3. .spec.priorityLevelConfiguration

描述

PriorityLevelConfigurationReference 包含指向正在使用的 "request-priority" 的信息。

类型

object

必填

- **name**

属性	类型	描述
name	字符串	name 是引用 Required 的优先级级别配置的名称。

17.2.1.4. .spec.rules

描述

规则 描述了哪些请求与此流模式匹配。只有至少有一个规则成员与请求匹配时，此 FlowSchema 才会匹配请求。如果它是一个空片段，则没有与 FlowSchema 匹配的请求。

类型

array

17.2.1.5. .spec.rules[]

描述

PolicyRulesWithSubjects 假定一个测试适用于一个 apiserver 的请求。测试会考虑主题进行请求、请求的动词以及要操作的资源。此 PolicyRulesWithSubjects 只在同时(a)至少一个主体成员与 request 和 (b)至少匹配请求时匹配一个请求，或者(b)至少匹配 resourceRules 或 nonResourceRules 的请求。

类型

object

必填

- **主题**

属性	类型	描述
----	----	----

属性	类型	描述
nonResourceRules	array	nonResourceRules 是 NonResourcePolicyRules 列表，它根据其操作动词和目标非资源 URL 识别匹配的请求。
nonResourceRules[]	对象	NonResourcePolicyRule 是一个 predicate，它根据其操作动词和目标非资源 URL 匹配非资源请求。NonResourcePolicyRule 在请求匹配时，只有最少一个操作动词成员与请求匹配，则 NonResourcePolicyRule 才会匹配请求，(b)至少匹配一个非 ResourceURL 的成员。
resourceRules	array	resourceRules 是 ResourcePolicyRules 的片段，它根据其操作动词和目标资源识别匹配的请求。至少一个 resourceRules 和 nonResourceRules 必须是非空的。
resourceRules[]	对象	ResourcePolicyRule 是一个与一些资源请求匹配的 predicate，测试请求的动词和目标资源。ResourcePolicyRule 匹配资源请求 (仅当：(d)至少有一个成员与请求匹配，(b)至少一个 apiGroups 的成员与请求匹配，(c)至少一个资源成员与请求匹配，(d1)请求不指定命名空间 (例如， Namespace="")，而 clusterScope 则指定命名空间的一个成员，以及 (d2 个命名空间)与一个命名空间匹配。
主题	array	subjects 是此规则所关注的普通用户、serviceaccount 或组的列表。这个片段中必须至少有一个成员。包含 system:authenticated 和 system:unauthenticated 用户组的片段与每个请求匹配。必需。
subjects[]	对象	主题与请求身份验证系统的发起者匹配，具体由请求身份验证系统识别。有三种匹配发起者的方法：用户、组或服务帐户。

17.2.1.6. .spec.rules[].nonResourceRules

描述

nonResourceRules 是 NonResourcePolicyRules 列表，它根据其操作动词和目标非资源 URL 识别匹配的请求。

类型

array

17.2.1.7. .spec.rules[].nonResourceRules[]

描述

NonResourcePolicyRule 是一个 predicate，它根据其操作动词和目标非资源 URL 匹配非资源请求。NonResourcePolicyRule 在请求匹配时，只有最少一个操作动词成员与请求匹配，则 NonResourcePolicyRule 才会匹配请求，(b)至少匹配一个非ResourceURL 的成员。

类型

object

必填

- **verbs**
- **nonResourceURLs**

属性	类型	描述
nonResourceURLs	数组（字符串）	nonResourceURLs 是一组 url 前缀，用户应有权访问且可能不为空。例如：- "/healthz" 是法律 - "/hea*" is illegal - "/hea" is legal but match nothing - "/hea/ " also match nothing - "/healthz/" match all per-resource urls. if it is present, it must only entry.必需。
verbs	数组（字符串）	verbs 是匹配的操作动词列表，可能不为空。"*" 匹配所有动词。如果存在，它必须是唯一的条目。必需。

17.2.1.8. .spec.rules[].resourceRules

描述

resourceRules 是 ResourcePolicyRules 的片段，它根据其操作动词和目标资源识别匹配的请求。至少一个 **resourceRules** 和 **nonResourceRules** 必须是非空的。

类型

array

17.2.1.9. .spec.rules[].resourceRules[]

描述

ResourcePolicyRule 是一个与一些资源请求匹配的 predicate，测试请求的动词和目标资源。ResourcePolicyRule 匹配资源请求（仅当：(d)至少有一个成员与请求匹配，(b)至少一个 apiGroups

的成员与请求匹配, (c)至少一个资源成员与请求匹配, (d1)请求不指定命名空间 (例如, **Namespace=""**), 而 **clusterScope** 则指定命名空间的一个成员, 以及 (d2 个命名空间)与一个命名空间匹配。

类型

object

必填

- **verbs**
- **apiGroups**
- **resources**

属性	类型	描述
apiGroups	数组 (字符串)	apiGroups 是匹配的 API 组列表, 且可能不为空。"*" 匹配所有 API 组, 如果存在, 则必须是唯一的条目。必需。
clusterScope	布尔值	ClusterScope 指明是否匹配没有指定命名空间的请求 (因为资源没有命名空间, 或请求以所有命名空间为目标)。如果省略此字段或 false , 则 namespaces 字段必须包含非空列表。
命名空间	数组 (字符串)	命名空间 是限制匹配的目标命名空间列表。只有在此列表包含该目标命名空间(b)此列表包含 "" 时, 指定目标命名空间的请求才会匹配。请注意, ""与任何指定命名空间匹配, 但不与未指定命名空间的请求匹配 (请参阅该命名空间的 clusterScope 字段)。此列表可能为空, 但只有 clusterScope 为 true 时。
resources	数组 (字符串)	资源是匹配资源 (如小写和复数) 的列表 (如果需要, 子资源)。例如, ["services", "nodes/status"]。此列表不能为空。""与所有资源匹配, 如果存在, 则必须是唯一的条目。必需。

属性	类型	描述
verbs	数组（字符串）	verbs 是匹配的操作动词列表，可能不为空。"*" 匹配所有动词，如果存在，则必须是唯一的条目。必需。

17.2.1.10. .spec.rules[].subjects

描述

subjects 是此规则所关注的普通用户、serviceaccount 或组的列表。这个片段中必须至少有一个成员。包含 system:authenticated 和 system:unauthenticated 用户组的片段与每个请求匹配。必需。

类型

array

17.2.1.11. .spec.rules[].subjects[]

描述

主题与请求身份验证系统的发起者匹配，具体由请求身份验证系统识别。有三种匹配发起者的方法：用户、组或服务帐户。

类型

object

必填

- **kind**

属性	类型	描述
group	对象	GroupSubject 包含 group-kind 主题的详细信息。
kind	字符串	kind 表示其中一个其他字段是非空的。必填
serviceAccount	对象	ServiceAccountSubject 包含 service-account-kind 主题的详细信息。
user	对象	UserSubject 包含 user-kind 主题的详细信息。

17.2.1.12. .spec.rules[].subjects[].group

描述

GroupSubject 包含 group-kind 主题的详细信息。

类型

object

必填

- **name**

属性	类型	描述
name	字符串	name 是匹配的用户组，或 "*" 以匹配所有用户组。有关某些已知的组名称，请参阅 https://github.com/kubernetes/apiserver/blob/master/pkg/authentication/user/user.go 。必需。

17.2.1.13. `.spec.rules[].subjects[].serviceAccount`

描述

ServiceAccountSubject 包含 service-account-kind 主题的详细信息。

类型

object

必填

- **namespace**
- **name**

属性	类型	描述
name	字符串	name 是匹配的 ServiceAccount 对象的名称，或 "*" 以匹配，无论名称是什么。必需。
namespace	字符串	namespace 是匹配的 ServiceAccount 对象的命名空间。必需。

17.2.1.14. `.spec.rules[].subjects[].user`

描述

UserSubject 包含 user-kind 主题的详细信息。

类型

object

必填

- **name**

属性	类型	描述
name	字符串	name 是匹配所有用户名的用户名，或 "*"。必需。

17.2.1.15. .status

描述

FlowSchemaStatus 代表 FlowSchema 的当前状态。

类型

object

属性	类型	描述
conditions	array	conditions 是 FlowSchema 的当前状态列表。
conditions[]	对象	FlowSchemaCondition 描述了 FlowSchema 的条件。

17.2.1.16. .status.conditions

描述

conditions 是 FlowSchema 的当前状态列表。

类型

array

17.2.1.17. .status.conditions[]

描述

FlowSchemaCondition 描述了 FlowSchema 的条件。

类型

object

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	Message 是人类可读的消息，指示最后一次转换的详细信息。

属性	类型	描述
reason	字符串	reason 是条件最后一次转换的唯一单词 CamelCase 原因。
status	字符串	status 是条件的状态。可以是 True, False, Unknown。必需。
type	字符串	type 是条件的类型。必需。

17.2.2. API 端点

可用的 API 端点如下：

- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas**
 - **DELETE** : 删除 FlowSchema 的集合
 - **GET**: 列出或监视类型为 FlowSchema 的对象
 - **POST** : 创建 FlowSchema
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/flowschemas**
 - **GET**: 观察单个对 FlowSchema 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}**
 - **DELETE** : 删除 FlowSchema
 - **GET**: 读取指定的 FlowSchema
 - **PATCH** : 部分更新指定的 FlowSchema
 - **PUT** : 替换指定的 FlowSchema
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/flowschemas/{name}**
 - **GET**: 观察对类型为 FlowSchema 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}/status**
 - **GET**: 指定 FlowSchema 的读取状态
 - **PATCH**: 部分更新指定 FlowSchema 的状态
 - **PUT** : 替换指定 FlowSchema 的状态

17.2.2.1. /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas

表 17.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 FlowSchema 的集合

表 17.2. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 17.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 17.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 FlowSchema 的对象

表 17.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 17.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchemaList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 FlowSchema

表 17.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.8. 主体参数

参数	类型	描述
正文 (body)	FlowSchema 模式	

表 17.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
201 - Created	FlowSchema 模式
202 - Accepted	FlowSchema 模式
401 - Unauthorized	空

17.2.2.2. /apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/flowschemas

表 17.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 FlowSchema 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 17.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

17.2.2.3. /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}

表 17.12. 全局路径参数

参数	类型	描述
name	字符串	FlowSchema 的名称

表 17.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 FlowSchema

表 17.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 17.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 17.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 FlowSchema

表 17.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 FlowSchema

表 17.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 17.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 17.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
201 - Created	FlowSchema 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 FlowSchema

表 17.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.22. 主体参数

参数	类型	描述
正文 (body)	FlowSchema 模式	

表 17.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
201 - Created	FlowSchema 模式
401 - Unauthorized	空

17.2.2.4. /apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/flowschemas/{name}

表 17.24. 全局路径参数

参数	类型	描述
name	字符串	FlowSchema 的名称

表 17.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind FlowSchema. deprecated 的对象的更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 17.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

17.2.2.5. /apis/flowcontrol.apiserver.k8s.io/v1beta3/flowschemas/{name}/status

表 17.27. 全局路径参数

参数	类型	描述
name	字符串	FlowSchema 的名称

表 17.28. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 FlowSchema 的读取状态

表 17.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 FlowSchema 的部分更新状态

表 17.30. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 17.31. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 17.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
201 - Created	FlowSchema 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 FlowSchema 的状态

表 17.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.34. 主体参数

参数	类型	描述
正文 (body)	FlowSchema 模式	

表 17.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	FlowSchema 模式
201 - Created	FlowSchema 模式
401 - Unauthorized	空

17.3. PRIORITYLEVELCONFIGURATION [FLOWCONTROL.APISERVER.K8S.IO/V1BETA3]

描述

priorityLevelConfiguration 代表优先级级别的配置。

类型

对象

17.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	metadata 是标准对象的元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	PriorityLevelConfigurationSpec 指定优先级级别的配置。
status	对象	PriorityLevelConfigurationStatus 代表"request-priority"的当前状态。

17.3.1.1. .spec**描述**

PriorityLevelConfigurationSpec 指定优先级级别的配置。

类型

object

必填

- **type**

属性	类型	描述
有限	对象	LimitedPriorityLevelConfiguration 指定如何处理受限制约束的请求。它解决了两个问题：- 如何对此优先级级别的请求有限？- 应该通过超过限制的请求完成哪些操作？
type	字符串	type 表示此优先级级别是否受到请求执行的限制。 "Exempt" 值表示此优先级级别的请求不受到限制（因此不会排队），且不会从提供到其他优先级级别的容量中推断出来。 "Limited" 值表示(a)此优先级级别的请求会受到限制，(b)服务器的一些有限容量只可用于这个优先级级别。必需。

17.3.1.2. .spec.limited

描述

LimitedPriorityLevelConfiguration 指定如何处理受限制约束的请求。它解决了两个问题：- 如何对此优先级级别的请求有限？- 应该通过超过限制的请求完成哪些操作？

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
borrowingLimitPercent	整数	<p>在存在时，borrowingLimitPercent 会为这个优先级级别分离多少位配置限制。这个限制被称为这个级别的 BorrowingConcurrencyLimit (BorrowingCL)，它是这个级别可能随时浏览的信号总数的限制。此字段具有该限制与级别 nominal concurrency 限制的比例。当此字段不是 nil 时，它必须保存非负整数，并且限制计算如下：</p> $\text{BorrowingCL}(i) = \text{round}(\text{NominalCL}(i) * \text{borrowingLimitPercent}(i)/100.0)$ <p>此字段的值可以大于 100，这意味着这个优先级级别可以借借多个席位，其大于其自身的并发限制 (NominalCL)。当此字段保留为 nil 时，限制会有效地无限。</p>
lendablePercent	整数	<p>lendablePercent 规定了级别 NominalCL 的比例，该部分可以被其他优先级级别借鉴。此字段的值必须介于 0 到 100 之间，其中包含，默认值为 0。其他级别可以从这个级别分离的席位数量，称为这个级别的 LendableConcurrencyLimit (LendableCL)，其定义如下。</p> $\text{LendableCL}(i) = \text{round}(\text{NominalCL}(i) * \text{lendablePercent}(i)/100.0)$
limitResponse	对象	<p>LimitResponse 定义如何处理目前无法执行的请求。</p>

属性	类型	描述
nominalConcurrencyShares	整数	<p>nominalConcurrencyShares (NCS)贡献这个级别的 NominalConcurrencyLimit (NominalCL)的计算。这是此优先级级别的执行席数。这可用于从此优先级级别分配的请求，以及从这个级别分配其他优先级级别的请求。服务器的并发限制(ServerCL)被划分成其 NCS 值中的有限优先级级别：</p> $\text{NominalCL}(i) = \text{ceil}(\text{ServerCL} * \text{NCS}(i) / \text{sum_ncs})$ $\text{sum_ncs} = \text{sum}[\text{limited priority level } k] \text{ NCS}(k)$ <p>较大的数字意味着一个更大的非并发限制，但会牺牲所有其他有限优先级级别。此字段的默认值为 30。</p>

17.3.1.3. .spec.limited.limitResponse

描述

LimitResponse 定义如何处理目前无法执行的请求。

类型

object

必填

- **type**

属性	类型	描述
排队	对象	QueuingConfiguration 包含用于排队的配置参数
type	字符串	type 为 "Queue" 或 "Reject"。"queue"表示无法在队列中保存时无法执行的请求，直到它们的执行或达到队列限制为止。"reject"表示在 arrival 时无法执行的请求将被拒绝。必需。

17.3.1.4. .spec.limited.limitResponse.queuing

描述

QueuingConfiguration 包含用于排队的配置参数

类型

object

属性	类型	描述
handSize	整数	handSize 是一个小数，它将请求的分片配置为队列。在此优先级级别排队请求时，请求的流标识符（字符串对）会被哈希处理，哈希值用于对队列列表进行清理，并处理此处指定大小的手动。该请求将置于该手中最短的队列中。 handSize 不能大于队列，因此应该显著小（因此有些重度流不会使大多数队列饱和）。有关设置此字段的更多指导，请参阅面向用户的文档。此字段的默认值为 8。
queueLengthLimit	整数	queueLengthLimit 是允许一次在这个优先级级别的给定队列中等待的最大请求数；过量请求被拒绝。这个值必须是正数。如果未指定，它将默认为 50。
队列	整数	Queue 是此优先级级别的队列数量。队列在每个 apiserver 之间独立存在。该值必须是正数。将它设置为 1 有效地导致对 shufflesharding 的影响，从而使相关的流方案的分辨方法无关。此字段的默认值为 64。

17.3.1.5. .status

描述

PriorityLevelConfigurationStatus 代表"request-priority"的当前状态。

类型

object

属性	类型	描述
conditions	array	条件 是"请求优先级"的当前状态。
conditions[]	对象	PriorityLevelConfigurationCondition 定义优先级级别的条件。

17.3.1.6. .status.conditions

描述

条件是"请求优先级"的当前状态。

类型

array

17.3.1.7. .status.conditions[]

描述

PriorityLevelConfigurationCondition 定义优先级级别的条件。

类型

object

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件从一个状态转换到另一个状态最后一次的时间。
message	字符串	Message 是人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	reason 是条件最后一次转换的唯一单词 CamelCase 原因。
status	字符串	status 是条件的状态。可以是 True, False, Unknown。必需。
type	字符串	type 是条件的类型。必需。

17.3.2. API 端点

可用的 API 端点如下：

- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations**
 - **DELETE** : 删除 PriorityLevelConfiguration 集合
 - **GET**: 列表或监视类型为 PriorityLevelConfiguration 的对象
 - **POST** : 创建一个 PriorityLevelConfiguration
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/prioritylevelconfigurations**
 - **GET**: 观察单个对 PriorityLevelConfiguration 列表的更改。已弃用：使用 'watch' 参数和一个 list 操作。

- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}**
 - **DELETE** : 删除 PriorityLevelConfiguration
 - **GET**: 读取指定的 PriorityLevelConfiguration
 - **PATCH**: 部分更新指定的 PriorityLevelConfiguration
 - **PUT** : 替换指定的 PriorityLevelConfiguration
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/prioritylevelconfigurations/{name}**
 - **GET**: 观察对类型为 PriorityLevelConfiguration. deprecated 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}/status**
 - **GET**: 指定的 PriorityLevelConfiguration 的读取状态
 - **PATCH**: 部分更新指定 PriorityLevelConfiguration 的状态
 - **PUT** : 替换指定 PriorityLevelConfiguration 的状态

17.3.2.1. /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations

表 17.36. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PriorityLevelConfiguration 集合

表 17.37. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 <code>continue</code> 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 <code>continue</code> 值，并带有相同的查询参数（除 <code>continue</code> 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 <code>continue</code> 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 <code>ResourceExpired</code> 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 <code>true</code> 时，不支持此字段。客户端可以从服务器返回的最后一个 <code>resourceVersion</code> 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 <code>nil</code>，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 17.38. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 17.39. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PriorityLevelConfiguration 的对象

表 17.40. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 17.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityLevelConfigurationList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 PriorityLevelConfiguration

表 17.42. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.43. 主体参数

参数	类型	描述
正文 (body)	priorityLevelConfiguration 模式	

表 17.44. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
201 - Created	priorityLevelConfiguration 模式
202 - Accepted	priorityLevelConfiguration 模式
401 - Unauthorized	空

17.3.2.2. /apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/prioritylevelconfigurations

表 17.45. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察到 PriorityLevelConfiguration 列表的单独更改。已弃用：使用 'watch' 参数和 list 操作。

表 17.46. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

17.3.2.3. /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}

表 17.47. 全局路径参数

参数	类型	描述
name	字符串	PriorityLevelConfiguration 的名称

表 17.48. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PriorityLevelConfiguration

表 17.49. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 17.50. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 17.51. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

读取指定的 PriorityLevelConfiguration

表 17.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 PriorityLevelConfiguration

表 17.53. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 17.54. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 17.55. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
201 - Created	priorityLevelConfiguration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 PriorityLevelConfiguration

表 17.56. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.57. 主体参数

参数	类型	描述
正文 (body)	priorityLevelConfiguration 模式	

表 17.58. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
201 - Created	priorityLevelConfiguration 模式
401 - Unauthorized	空

17.3.2.4. /apis/flowcontrol.apiserver.k8s.io/v1beta3/watch/prioritylevelconfigurations/{name}

表 17.59. 全局路径参数

参数	类型	描述
name	字符串	PriorityLevelConfiguration 的名称

表 17.60. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 `PriorityLevelConfiguration.deprecated` 的对象的更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 17.61. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>WatchEvent</code> 模式
401 - Unauthorized	空

17.3.2.5. /apis/flowcontrol.apiserver.k8s.io/v1beta3/prioritylevelconfigurations/{name}/status

表 17.62. 全局路径参数

参数	类型	描述
<code>name</code>	字符串	PriorityLevelConfiguration 的名称

表 17.63. 全局查询参数

参数	类型	描述
<code>pretty</code>	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 PriorityLevelConfiguration 的读取状态

表 17.64. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>priorityLevelConfiguration</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 PriorityLevelConfiguration 的部分更新状态

表 17.65. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 17.66. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 17.67. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
201 - Created	priorityLevelConfiguration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 PriorityLevelConfiguration 的状态

表 17.68. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 17.69. 主体参数

参数	类型	描述
正文 (body)	priorityLevelConfiguration 模式	

表 17.70. HTTP 响应

HTTP 代码	响应正文
200 - OK	priorityLevelConfiguration 模式
201 - Created	priorityLevelConfiguration 模式
401 - Unauthorized	空

第 18 章 网络 API

18.1. 网络 API

18.1.1. Ingress [networking.k8s.io/v1]

描述

Ingress 是规则的集合，允许入站连接访问后端定义的端点。可将 Ingress 配置为为外部可访问的 urls 提供服务，负载均衡流量、终止 SSL，提供基于名称的虚拟主机等。

类型

对象

18.1.2. IngressClass [networking.k8s.io/v1]

描述

IngressClass 代表 Ingress Spec 引用的 Ingress 类。**ingressclass.kubernetes.io/is-default-class** 注解可以用来指示 IngressClass 应该被视为默认值。当单个 IngressClass 资源将此注解设置为 true 时，没有指定类的新 Ingress 资源将被分配此默认存储类。

类型

对象

18.1.3. NetworkPolicy [networking.k8s.io/v1]

描述

NetworkPolicy 描述了一组 Pod 允许哪些网络流量

类型

对象

18.2. INGRESS [NETWORKING.K8S.IO/V1]

描述

Ingress 是规则的集合，允许入站连接访问后端定义的端点。可将 Ingress 配置为为外部可访问的 urls 提供服务，负载均衡流量、终止 SSL，提供基于名称的虚拟主机等。

类型

对象

18.2.1. 规格

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	IngressSpec 描述了用户希望存在的 Ingress。
status	对象	IngressStatus 描述 Ingress 的当前状态。

18.2.1.1. .spec

描述

IngressSpec 描述了用户希望存在的 Ingress。

类型

object

属性	类型	描述
defaultBackend	对象	IngressBackend 描述给定服务和端口的所有端点。

属性	类型	描述
ingressClassName	string	ingressClassName 是 IngressClass 集群资源的名称。Ingress 控制器实现使用此字段知道它们是否应该通过传输连接(controller → IngressClass → Ingress resource) 提供此 Ingress 资源。虽然 kubernetes.io/ingress.class 注解(simple constant name)没有被正式定义, 但 Ingress 控制器广泛支持 Ingress 控制器和 Ingress 资源之间的直接绑定。新创建的 Ingress 资源应该首选使用字段。但是, 即使注解被正式弃用, 为了向后兼容的原因, 入口控制器仍应遵循该注解 (如果存在)。
rules	数组	规则是用于配置 Ingress 的主机规则列表。如果未指定, 或者没有规则匹配, 所有流量都会发送到默认后端。
rules[]	对象	IngressRule 代表将指定主机下的路径映射到相关后端服务的规则。传入的请求会首先为主机匹配评估, 然后路由到与匹配的 IngressRuleValue 关联的后端。
tls	数组	TLS 代表 TLS 配置。目前, Ingress 只支持单个 TLS 端口 443。如果此列表的多个成员指定了不同的主机, 则根据通过 SNI TLS 扩展指定的主机名的多个主机名, 如果入口控制器满足 ingress 支持 SNI, 则它们将在同一端口上进行多路复用。
tls[]	object	IngressTLS 描述了与入口关联的传输层安全性。

18.2.1.2. .spec.defaultBackend

描述

IngressBackend 描述给定服务和端口的所有端点。

类型

object

属性	类型	描述
resource	TypedLocalObjectReference	resource 是 Ingress 对象命名空间中的另一个 Kubernetes 资源的 ObjectRef。如果指定了 resource，则不能指定 service.Name 和 service.Port。这是与 "Service" 相互排斥的设置。
service	对象	IngressServiceBackend 引用 Kubernetes 服务作为后端。

18.2.1.3. .spec.defaultBackend.service

描述

IngressServiceBackend 引用 Kubernetes 服务作为后端。

类型

object

必填

- name

属性	类型	描述
name	string	name 是引用的服务。该服务必须与 Ingress 对象位于同一个命名空间中。
port	对象	ServiceBackendPort 是被引用的服务端口。

18.2.1.4. .spec.defaultBackend.service.port

描述

ServiceBackendPort 是被引用的服务端口。

类型

object

属性	类型	描述
name	string	name 是 Service 上端口的名称。这是与 "Number" 相互排斥的设置。

属性	类型	描述
number	整数	number 是服务中的数字端口号（如 80）。这是与 "Name" 相互排斥的设置。

18.2.1.5. .spec.rules

描述

规则是用于配置 Ingress 的主机规则列表。如果未指定，或者没有规则匹配，所有流量都会发送到默认后端。

类型

array

18.2.1.6. .spec.rules[]

描述

IngressRule 代表将指定主机下的路径映射到相关后端服务的规则。传入的请求会首先为主机匹配评估，然后路由到与匹配的 IngressRuleValue 关联的后端。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
主机	string	<p>host 是网络主机的完全限定域名，由 RFC 3986 定义。请注意，以下来自 URI 的"主机"部分，如 RFC 3986:1 中所定义。不允许 IP。目前，IngressRuleValue 只能应用到父入口的 Spec 中的 IP。2.: 分隔符不会被遵守，因为不允许端口。目前，Ingress 的端口被隐式用于 http, https 为 :443。它们可能会在以后改变。在 IngressRuleValue 之前，传入的请求会与主机匹配。如果未指定主机，Ingress 会根据指定的 IngressRuleValue 路由所有流量。</p> <p>主机可以是 "precise"，它是一个没有网络主机的终止点的域名（如 "foo.bar.com"）或 "wildcard"，它是一个带有单个通配符标签的域名（如 ".foo.com"）。通配符字符 " 必须显示为第一个 DNS 标签，且只匹配单个标签。您不能自行具有通配符标签（如 Host == "*"）。请求将通过以下方式与 Host 字段匹配：1.如果 host 准确，如果 http 主机标头等于 Host，则请求会匹配此规则。2.如果 host 是通配符，如果 http 主机标头等于通配符规则的后缀（删除第一个标签），则请求会匹配此规则。</p>
http	对象	<p>HTTPIngressRuleValue 是指向后端的 http 选择器的列表。在示例中：http://<host>/<path>?<searchpart> → backend where where where the url 部分与 RFC 3986 对应，这个资源将用于与最后一个 '/' 以及第一个 '?' 或 '#' 后的所有内容匹配。</p>

18.2.1.7. .spec.rules[].http

描述

HTTPIngressRuleValue 是指向后端的 http 选择器的列表。在示例中：[http://<host>/<path>?<searchpart>](#) → backend where where where the url 部分与 RFC 3986 对应，这个资源将用于与最后一个 '/' 以及第一个 '?' 或 '#' 后的所有内容匹配。

类型

object

必填

- 路径

属性	类型	描述
路径	数组	paths 是将请求映射到后端的路径集合。
paths[]	对象	HTTPIngressPath 将路径与后端相关联。与路径匹配的传入 url 转发到后端。

18.2.1.8. .spec.rules[].http.paths

描述

paths 是将请求映射到后端的路径集合。

类型

array

18.2.1.9. .spec.rules[].http.paths[]

描述

HTTPIngressPath 将路径与后端相关联。与路径匹配的传入 url 转发到后端。

类型

object

必填

- **pathType**
- **后端**

属性	类型	描述
后端	对象	IngressBackend 描述给定服务和端口的所有端点。
path	string	路径与传入请求的路径匹配。目前，它可以包含不允许来自 RFC 3986 定义的 URL 的传统"path"部分的字符。路径必须以 '/' 开头，且在使用值为 "Exact" 或 "Prefix" 的 PathType 时必须存在。

属性	类型	描述
pathType	string	<p>pathType 决定路径匹配的解释。pathType 可以是以下值之一：*</p> <ul style="list-style-type: none"> Exact: 精确匹配 URL 路径。* prefix: 根据 URL 路径前缀按 '/' 进行匹配。匹配通过 path 元素根据路径元素完成。path 元素引用路径中的标签列表，用 '/' 分隔符分割。如果每个 p 都是一个请求路径路径的元素范围前缀，则请求与路径 p 匹配。请注意，如果路径的最后元素是请求路径中最后一个元素的子字符串，则它不是匹配项（例如 /foo/bar 匹配 /foo/bar/baz），但没有匹配 /foo/barbaz。* <p>* 特定于实现：与路径匹配的修改是 IngressClass。实现可将其视为单独的 PathType，或者将其与 Prefix 或 Exact 路径类型相同。需要实现来支持所有路径类型。</p> <p>可能枚举值： - "Exact" 准确匹配 URL 路径，以及区分大小写。 - "ImplementationSpecific" 匹配最多为 IngressClass。实现可将其视为单独的 PathType，或者将其与 Prefix 或 Exact 路径类型相同。 - "Prefix" 匹配基于 URL 路径前缀被 '/' 分割。匹配区分大小写，并基于 path 元素完成。path 元素引用路径中的标签列表，用 '/' 分隔符进行分割。如果每个 p 都是一个请求路径路径的元素范围前缀，则请求与路径 p 匹配。请注意，如果路径的最后元素是请求路径中最后一个元素的子字符串，则它不是匹配项（例如 /foo/bar 匹配 /foo/bar/baz），但没有匹配 /foo/barbaz。如果 Ingress spec 中存在多个匹配路径，则指定优先级最长匹配的路径。示例： - /foo/bar 不匹配对 /foo/barbaz - /foo/bar 和 /foo/bar/baz 的请求， /foo/bar/baz - /foo 和 /foo/ 都匹配请求到 /foo 和 /foo/。如果 Ingress spec 中存在这两个路径，则会赋予最长匹配的路径 (/foo/)</p>

18.2.1.10. .spec.rules[].http.paths[].backend

描述

IngressBackend 描述给定服务和端口的所有端点。

类型

object

属性	类型	描述
resource	TypedLocalObjectReference	resource 是 Ingress 对象命名空间中的另一个 Kubernetes 资源的 ObjectRef。如果指定了 resource，则不能指定 service.Name 和 service.Port。这是与 "Service" 相互排斥的设置。
service	对象	IngressServiceBackend 引用 Kubernetes 服务作为后端。

18.2.1.11. .spec.rules[].http.paths[].backend.service

描述

IngressServiceBackend 引用 Kubernetes 服务作为后端。

类型

object

必填

- **name**

属性	类型	描述
name	string	name 是引用的服务。该服务必须与 Ingress 对象位于同一个命名空间中。
port	对象	ServiceBackendPort 是被引用的服务端口。

18.2.1.12. .spec.rules[].http.paths[].backend.service.port

描述

ServiceBackendPort 是被引用的服务端口。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
name	string	name 是 Service 上端口的名称。这是与 "Number" 相互排斥的设置。
number	整数	number 是服务中的数字端口号（如 80）。这是与 "Name" 相互排斥的设置。

18.2.1.13. .spec.tls

描述

TLS 代表 TLS 配置。目前，Ingress 只支持单个 TLS 端口 443。如果此列表的多个成员指定了不同的主机，则根据通过 SNI TLS 扩展指定的主机名的多个主机名，如果入口控制器满足 ingress 支持 SNI，则它们将在同一端口上进行多路复用。

类型

array

18.2.1.14. .spec.tls[]

描述

IngressTLS 描述了与入口关联的传输层安全性。

类型

object

属性	类型	描述
主机	数组（字符串）	主机是 TLS 证书中包含的主机列表。此列表中的值必须与 tlsSecret 中使用的名称/匹配。如果未指定，则默认为 loadbalancer 控制器的通配符 host 设置来满足这个 Ingress。
secretName	string	secretName 是用于在端口 443 上终止 TLS 流量的 secret 名称。字段是可选的，以允许仅基于 SNI 主机名的 TLS 路由。如果监听器中的 SNI 主机与 IngressRule 使用的 "Host" 标头字段冲突，则 SNI 主机用于终止，并使用 "Host" 标头的值进行路由。

18.2.1.15. .status

描述

IngressStatus 描述 Ingress 的当前状态。

类型

object

属性	类型	描述
loadBalancer	object	IngressLoadBalancerStatus 代表负载均衡器的状态。

18.2.1.16. `.status.loadBalancer`

描述

IngressLoadBalancerStatus 代表负载均衡器的状态。

类型

object

属性	类型	描述
ingress	数组	Ingress 是一个包含负载均衡器入口点的列表。
ingress[]	object	IngressLoadBalancerIngress 代表负载均衡器入口点的状态。

18.2.1.17. `.status.loadBalancer.ingress`

描述

Ingress 是一个包含负载均衡器入口点的列表。

类型

array18.2.1.18. `.status.loadBalancer.ingress[]`

描述

IngressLoadBalancerIngress 代表负载均衡器入口点的状态。

类型

object

属性	类型	描述
hostname	string	为基于 DNS 的负载均衡器入口点设置 hostname。
ip	string	为基于 IP 的负载均衡器入口点设置 IP。

属性	类型	描述
ports	数组	端口提供有关此 LoadBalancer 公开的端口的信息。
ports[]	object	IngressPortStatus 代表服务端口的错误条件

18.2.1.19. .status.loadBalancer.ingress[].ports

描述

端口提供有关此 LoadBalancer 公开的端口的信息。

类型

array

18.2.1.20. .status.loadBalancer.ingress[].ports[]

描述

IngressPortStatus 代表服务端口的错误条件

类型

object

必填

- **port**
- **protocol**

属性	类型	描述
错误	string	错误是记录服务端口的问题。错误的格式应遵循以下规则：- 在此文件中指定内置错误值，它们应使用 CamelCase 名称 - 云供应商特定错误值必须符合 foo.example.com/CamelCase 格式的名称。
port	整数	port 是入口端口的端口号。
protocol	string	protocol 是入口端口的协议。支持的值有："TCP", "UDP", "SCTP" 可能的值有：- " SCTP " 是 SCTP 协议。- " TCP " 是 TCP 协议。- " UDP " 是 UDP 协议。

18.2.2. API 端点

可用的 API 端点如下：

- **/apis/networking.k8s.io/v1/ingresses**
 - **GET**: 列出或监视类型为 Ingress 的对象
- **/apis/networking.k8s.io/v1/watch/ingresses**
 - **GET**: 观察单个对 Ingress 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses**
 - **DELETE** : 删除 Ingress 集合
 - **GET**: 列出或监视类型为 Ingress 的对象
 - **POST** : 创建 Ingress
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses**
 - **GET**: 观察单个对 Ingress 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}**
 - **DELETE** : 删除 Ingress
 - **GET**: 读取指定的入口
 - **PATCH** : 部分更新指定的入口
 - **PUT** : 替换指定的 Ingress
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses/{name}**
 - **GET**: 观察对类型为 Ingress 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status**
 - **GET**: 指定 Ingress 的读取状态
 - **PATCH** : 部分更新指定入口的状态
 - **PUT** : 替换指定 Ingress 的状态

18.2.2.1. /apis/networking.k8s.io/v1/ingresses

表 18.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Ingress 的对象

表 18.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressList 模式
401 - Unauthorized	空

18.2.2.2. /apis/networking.k8s.io/v1/watch/ingresses

表 18.3. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Ingress 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 18.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.2.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses

表 18.5. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.6. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Ingress 集合

表 18.7. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下：</p> <ul style="list-style-type: none"> - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 18.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 18.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式

HTTP 代码	响应正文
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Ingress 的对象

表 18.10. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 18.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressList 模式
401 - Unauthorized	空

HTTP 方法**POST****描述**

创建 Ingress

表 18.12. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.13. 主体参数

参数	类型	描述
正文 (body)	Ingress 模式	

表 18.14. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
201 - Created	Ingress 模式
202 - Accepted	Ingress 模式
401 - Unauthorized	空

18.2.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses

表 18.15. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.16. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Ingress 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 18.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.2.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}

表 18.18. 全局路径参数

参数	类型	描述
name	字符串	Ingress 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.19. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 Ingress

表 18.20. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 18.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 18.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的入口

表 18.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Ingress</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的入口

表 18.24. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 18.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 18.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
201 - Created	Ingress 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 Ingress

表 18.27. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.28. 主体参数

参数	类型	描述
正文 (body)	Ingress 模式	

表 18.29. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
201 - Created	Ingress 模式
401 - Unauthorized	空

18.2.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/ingresses/{name}

表 18.30. 全局路径参数

参数	类型	描述
name	字符串	Ingress 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 18.31. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 Ingress 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 18.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.2.2.7. /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status

表 18.33. 全局路径参数

参数	类型	描述
name	字符串	Ingress 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 18.34. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 Ingress 的读取状态

表 18.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 Ingress 的部分更新状态

表 18.36. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 18.37. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 18.38. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
201 - Created	Ingress 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 Ingress 的状态

表 18.39. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.40. 主体参数

参数	类型	描述
正文 (body)	Ingress 模式	

表 18.41. HTTP 响应

HTTP 代码	响应正文
200 - OK	Ingress 模式
201 - Created	Ingress 模式
401 - Unauthorized	空

18.3. INGRESSCLASS [NETWORKING.K8S.IO/V1]

描述

IngressClass 代表 Ingress Spec 引用的 Ingress 类。**ingressclass.kubernetes.io/is-default-class** 注解可以用来指示 IngressClass 应该被视为默认值。当单个 IngressClass 资源将此注解设置为 true 时，没有指定类的新 Ingress 资源将被分配此默认存储类。

类型

对象

18.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	IngressClassSpec 提供有关 Ingress 类的信息。

18.3.1.1. .spec

描述

IngressClassSpec 提供有关 Ingress 类的信息。

类型

object

属性	类型	描述
controller	string	Controller 指的是应处理此类的控制器名称。这允许由同一控制器控制的不同"flavors"。例如，对于相同的实现控制器，您可能有不同的参数。这应该被指定为 domain-prefixed 路径长度不超过 250 个字符，例如 "acme.io/ingress-controller"。此字段是不可变的。

属性	类型	描述
parameters	对象	IngressClassParametersReference 标识 API 对象。这可用于指定集群或命名空间范围的资源。

18.3.1.2. .spec.parameters

描述

IngressClassParametersReference 标识 API 对象。这可用于指定集群或命名空间范围的资源。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	string	apiGroup 是被引用资源的组。如果没有指定 APIGroup，则指定的 Kind 必须位于 core API 组中。对于任何其他第三方类型，需要 APIGroup。
kind	string	kind 是被引用的资源的类型。
name	string	name 是被引用的资源的名称。
namespace	string	namespace 是被引用的资源的命名空间。当 scope 设为 "Namespace" 时，需要此字段，且当 scope 设置为 "Cluster" 时，必须取消设置。
scope	string	scope 代表是否指的是集群或命名空间范围资源。这可设置为 "Cluster"（默认）或 "Namespace"。

18.3.2. API 端点

可用的 API 端点如下：

- **/apis/networking.k8s.io/v1/ingressclasses**

- **DELETE** : 删除 IngressClass 的集合
- **GET**: 列表或监视类型为 IngressClass 的对象
- **POST** : 创建 IngressClass
- **/apis/networking.k8s.io/v1/watch/ingressclasses**
 - **GET**: 观察单个对 IngressClass 列表的更改。已弃用 : 改为使用 'watch' 参数和 list 操作。
- **/apis/networking.k8s.io/v1/ingressclasses/{name}**
 - **DELETE** : 删除 IngressClass
 - **GET**: 读取指定的 IngressClass
 - **PATCH** : 部分更新指定的 IngressClass
 - **PUT** : 替换指定的 IngressClass
- **/apis/networking.k8s.io/v1/watch/ingressclasses/{name}**
 - **GET**: 观察对类型为 IngressClass 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

18.3.2.1. /apis/networking.k8s.io/v1/ingressclasses

表 18.42. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 IngressClass 集合

表 18.43. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 18.44. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 18.45. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 IngressClass 的对象

表 18.46. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 18.47. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressClassList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 IngressClass

表 18.48. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.49. 主体参数

参数	类型	描述
正文 (body)	IngressClass 模式	

表 18.50. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressClass 模式
201 - Created	IngressClass 模式
202 - Accepted	IngressClass 模式
401 - Unauthorized	空

18.3.2.2. /apis/networking.k8s.io/v1/watch/ingressclasses

表 18.51. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 IngressClass 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 18.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.3.2.3. /apis/networking.k8s.io/v1/ingressclasses/{name}

表 18.53. 全局路径参数

参数	类型	描述
name	字符串	IngressClass 的名称

表 18.54. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 IngressClass

表 18.55. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 18.56. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 18.57. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 IngressClass

表 18.58. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressClass 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 IngressClass

表 18.59. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 18.60. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 18.61. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressClass 模式
201 - Created	IngressClass 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 IngressClass

表 18.62. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.63. 主体参数

参数	类型	描述
正文 (body)	IngressClass 模式	

表 18.64. HTTP 响应

HTTP 代码	响应正文
200 - OK	IngressClass 模式
201 - Created	IngressClass 模式
401 - Unauthorized	空

18.3.2.4. /apis/networking.k8s.io/v1/watch/ingressclasses/{name}

表 18.65. 全局路径参数

参数	类型	描述
name	字符串	IngressClass 的名称

表 18.66. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 `IngressClass.deprecated` 的对象的更改：使用 `'watch'` 参数和 `list` 操作，而是过滤到带有 `'fieldSelector'` 参数的单个项目。

表 18.67. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.4. NETWORKPOLICY [NETWORKING.K8S.IO/V1]

描述

`NetworkPolicy` 描述了一组 Pod 允许哪些网络流量

类型

对象

18.4.1. 规格

属性	类型	描述
apiVersion	字符串	<code>APIVersion</code> 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	<code>kind</code> 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	<code>NetworkPolicySpec</code> 提供 <code>NetworkPolicy</code> 规格

属性	类型	描述
status	object	NetworkPolicyStatus 描述了 NetworkPolicy 的当前状态。

18.4.1.1. .spec

描述

NetworkPolicySpec 提供 NetworkPolicy 规格

类型

object

必填

- **podSelector**

属性	类型	描述
egress	数组	egress 是要应用到所选 pod 的出口规则列表。如果没有 NetworkPolicies 选择 pod（集群策略其他允许流量），或者流量至少匹配 podSelector 与 podSelector 匹配的所有 NetworkPolicy 对象，则允许传出流量。如果此字段为空，则此 NetworkPolicy 会限制所有传出流量（仅提供确保默认隔离它的 pod）。这个字段在 1.8 中是 beta 级
egress[]	对象	NetworkPolicyEgressRule 描述了一组特定的流量，这些流量允许来自 NetworkPolicySpec 的 podSelector 匹配的 pod。流量必须匹配端口 和。这个类型是 1.8 中的 beta 级别

属性	类型	描述
ingress	数组	Ingress 是要应用到所选 pod 的入站规则列表。如果没有 NetworkPolicies 选择 pod（否则集群策略允许流量），或者流量源是 pod 的本地节点，或者流量源是 pod 的本地节点，或者流量在 podSelector 匹配的所有 NetworkPolicy 对象中至少匹配一个入口规则，或者流量在 podSelector 匹配的所有 NetworkPolicy 对象中至少匹配一个入口规则。如果此字段为空，则此 NetworkPolicy 不允许任何流量（仅提供确保默认隔离它的 pod）
ingress[]	对象	NetworkPolicyIngressRule 描述了允许与 NetworkPolicySpec 的 pod 匹配的 pod 的特定流量集合。流量必须与端口 和 from 匹配。
podSelector	labelSelector	podSelector 选择此 NetworkPolicy 对象应用到的 pod。入口规则的数组应用于此字段选择的任何 pod。多个网络策略可以选择同一组 pod。在这种情况下，每个系统的入站规则组合使用。此字段不是可选，并遵循标准标签选择器语义。空 podSelector 匹配此命名空间中的所有 pod。
policyTypes	数组（字符串）	policyTypes 是 NetworkPolicy 相关的规则类型列表。有效选项为 ["Ingress"], ["Egress"], 或 ["Ingress", "Egress"]。如果没有指定此字段，它将默认基于入口或出口规则存在；包含 egress 部分的策略会被假定为影响出口，所有策略（无论它们是否包含 ingress 部分）都假定为影响 ingress。如果要编写仅出口策略，您必须明确指定 policyTypes ["Egress"]。同样，如果您想要编写指定不允许出口的策略，您必须指定包含 "Egress" 的 policyTypes 值（因为这样的策略不包括 egress 部分，否则默认为 ["Ingress"]）。这个字段在 1.8 中是 beta 级

18.4.1.2. .spec.egress

描述

egress 是要应用到所选 pod 的出口规则列表。如果没有 NetworkPolicies 选择 pod（集群策略其他允许流量），或者流量至少匹配 podSelector 与 podSelector 匹配的所有 NetworkPolicy 对象，则允许传出流量。如果此字段为空，则此 NetworkPolicy 会限制所有传出流量（仅提供确保默认隔离它的 pod）。这个字段在 1.8 中是 beta 级

类型

array

18.4.1.3. .spec.egress[]

描述

NetworkPolicyEgressRule 描述了一组特定的流量，这些流量允许来自 NetworkPolicySpec 的 podSelector 匹配的 pod。流量必须匹配端口 和。这个类型是 1.8 中的 beta 级别

类型

object

属性	类型	描述
ports	数组	端口是传出流量的目标端口列表。此列表中的每个项目都使用逻辑 OR 合并。如果此字段为空或缺失，则该规则匹配所有端口（不受端口限制的流量）。如果存在此字段并至少包含一个项目，则此规则仅在流量与列表中至少匹配一个端口时才允许流量。
ports[]	对象	NetworkPolicyPort 描述了允许流量的端口
至	数组	to 是为这个规则选择的 pod 传出流量的目的地列表。此列表中的项目使用逻辑 OR 操作组合。如果此字段为空或缺失，则该规则匹配所有目的地（不受目的地限制的流量）。如果此字段存在并至少包含一个项目，则该规则仅在流量与要列表中的至少一个项目匹配时才允许流量。
to[]	对象	NetworkPolicyPeer 描述允许流量到/来自的对等点。只允许某些字段组合

18.4.1.4. .spec.egress[].ports

描述

端口是传出流量的目标端口列表。此列表中的每个项目都使用逻辑 OR 合并。如果此字段为空或缺失

端口是传出流量的目标端口列表。此列表中的每个项目都使用逻辑 OR 台开。如果此字段为空或缺失，则该规则匹配所有端口（不受端口限制的流量）。如果存在此字段并至少包含一个项目，则此规则仅在流量与列表中至少匹配一个端口时才允许流量。

类型

array

18.4.1.5. .spec.egress[].ports[]

描述

NetworkPolicyPort 描述了允许流量的端口

类型

object

属性	类型	描述
endPort	整数	endPort 表示策略应允许端口的范围从端口到 endPort。如果没有定义 port 字段，或者 port 字段定义为 named（字符串）端口，则无法定义此字段。endPort 必须等于或大于 port。
port	IntOrString	port 代表给定协议上的端口。这可以是 pod 上的数字或指定端口。如果没有提供此字段，这将匹配所有端口名称和数字。如果存在，则仅匹配指定协议和端口上的流量。
protocol	string	protocol 代表流量必须匹配的协议 (TCP、UDP 或 SCTP)。如果没有指定，此字段默认为 TCP。 可能的值有： - "SCTP" 是 SCTP 协议。 - "TCP" 是 TCP 协议。 - "UDP" 是 UDP 协议。

18.4.1.6. .spec.egress[].to

描述

to 是为这个规则选择的 pod 传出流量的目的地列表。此列表中的项目使用逻辑 OR 操作组合。如果此字段为空或缺失，则该规则匹配所有目的地（不受目的地限制的流量）。如果此字段存在并至少包含一个项目，则该规则仅在流量与要列表中的至少一个项目匹配时才允许流量。

类型

array

18.4.1.7. .spec.egress[].to[]

描述

NetworkPolicyPeer 描述允许流量到/来自的对等点。只允许某些字段组合

类型

object

属性	类型	描述
ipBlock	对象	IPBlock 描述了特定的 CIDR (Ex)。"192.168.1.0/24","2001:db8::/64"，允许与 NetworkPolicySpec 的 podSelector 匹配的 pod。之外的条目描述了此规则中不应包含的 CIDR。
namespaceSelector	labelSelector	namespaceSelector 使用集群范围的标签选择命名空间。此字段遵循标准标签选择器语义；如果存在但为空，它会选择所有命名空间。 如果还设置了 podSelector，则 NetworkPolicyPeer 作为整个设置会选择与 namespaceSelector 选择的命名空间中与 podSelector 匹配的 pod。否则，它会选择 namespaceSelector 选择的命名空间中的所有 pod。
podSelector	labelSelector	podSelector 是一个选择 pod 的标签选择器。此字段遵循标准标签选择器语义；如果存在但为空，它会选择所有 pod。 如果还设置了 namespaceSelector，则 NetworkPolicyPeer 作为整个策略选择的命名空间中与 podSelector 匹配的 pod。否则，它会在策略自己的命名空间中选择匹配的 podSelector。

18.4.1.8. .spec.egress[].to[].ipBlock**描述**

IPBlock 描述了特定的 CIDR (Ex)。"192.168.1.0/24","2001:db8::/64"，允许与 NetworkPolicySpec 的 podSelector 匹配的 pod。之外的条目描述了此规则中不应包含的 CIDR。

类型**object****必填**

- cidr

属性	类型	描述
cidr	string	CIDR 是一个代表 IPBlock Valid 示例的字符串，即 "192.168.1.0/24" 或 "2001:db8::/64"
except	数组（字符串）	除了 是不应包含在 IPBlock Valid 示例中的 CIDR 的片段为 "192.168.1.0/24" 或 "2001:db8::/64" Except 值（如果它们不在 cidr 范围外），则会被拒绝

18.4.1.9. .spec.ingress

描述

Ingress 是要应用到所选 pod 的进站规则列表。如果没有 NetworkPolicies 选择 pod（否则集群策略允许流量），或者流量源是 pod 的本地节点，或者流量源是 pod 的本地节点，或者流量在 podSelector 匹配的所有 NetworkPolicy 对象中至少匹配一个入口规则，或者流量在 podSelector 匹配的所有 NetworkPolicy 对象中至少匹配一个入口规则。如果此字段为空，则此 NetworkPolicy 不允许任何流量（仅提供确保默认隔离它的 pod）

类型

array

18.4.1.10. .spec.ingress[]

描述

NetworkPolicyIngressRule 描述了允许与 NetworkPolicySpec 的 pod 匹配的 pod 的特定流量集合。流量必须与端口 和 from 匹配。

类型

object

属性	类型	描述
from	数组	from 是源列表，应该可以访问为此规则选择的 pod。此列表中的项目使用逻辑 OR 操作组合。如果此字段为空或缺失，则该规则匹配所有源（不受源限制的流量）。如果此字段存在并至少包含一个项目，则此规则仅在流量与 from 列表中的至少一个项目匹配时才允许流量。
from[]	对象	NetworkPolicyPeer 描述允许流量到/来自的对等点。只允许某些字段组合

属性	类型	描述
ports	数组	端口是应在为此规则选择的 pod 上访问的端口列表。此列表中的每个项目都使用逻辑 OR 合并。如果此字段为空或缺失，则该规则匹配所有端口（不受端口限制的流量）。如果存在此字段并至少包含一个项目，则此规则仅在流量与列表中至少匹配一个端口时才允许流量。
ports[]	对象	NetworkPolicyPort 描述了允许流量的端口

18.4.1.11. .spec.ingress[].from

描述

from 是源列表，应该可以访问为此规则选择的 pod。此列表中的项目使用逻辑 OR 操作组合。如果此字段为空或缺失，则该规则匹配所有源（不受源限制的流量）。如果此字段存在并至少包含一个项目，则此规则仅在流量与 from 列表中的至少一个项目匹配时才允许流量。

类型

array

18.4.1.12. .spec.ingress[].from[]

描述

NetworkPolicyPeer 描述允许流量到/来自的对等点。只允许某些字段组合

类型

object

属性	类型	描述
ipBlock	对象	IPBlock 描述了特定的 CIDR (Ex). "192.168.1.0/24","2001:db8::/64" ，允许与 NetworkPolicySpec 的 podSelector 匹配的 pod。之外的条目描述了此规则中不应包含的 CIDR。

属性	类型	描述
namespaceSelector	labelSelector	<p>namespaceSelector 使用集群范围的标签选择命名空间。此字段遵循标准标签选择器语义；如果存在但为空，它会选择所有命名空间。</p> <p>如果还设置了 podSelector，则 NetworkPolicyPeer 作为整个设置会选择与 namespaceSelector 选择的命名空间中与 podSelector 匹配的 pod。否则，它会选择 namespaceSelector 选择的命名空间中的所有 pod。</p>
podSelector	labelSelector	<p>podSelector 是一个选择 pod 的标签选择器。此字段遵循标准标签选择器语义；如果存在但为空，它会选择所有 pod。</p> <p>如果还设置了 namespaceSelector，则 NetworkPolicyPeer 作为整个策略选择的命名空间中与 podSelector 匹配的 pod。否则，它会在策略自己的命名空间中选择匹配的 podSelector。</p>

18.4.1.13. .spec.ingress[].from[].ipBlock

描述

IPBlock 描述了特定的 CIDR (Ex)。"192.168.1.0/24","2001:db8::/64")，允许与 NetworkPolicySpec 的 podSelector 匹配的 pod。之外的条目描述了此规则中不应包含的 CIDR。

类型

object

必填

- cidr

属性	类型	描述
cidr	string	CIDR 是一个代表 IPBlock Valid 示例的字符串，即 "192.168.1.0/24" 或 "2001:db8::/64"

属性	类型	描述
except	数组（字符串）	除了 是不应包含在 IPBlock Valid 示例中的 CIDR 的片段为 "192.168.1.0/24" 或 "2001:db8::/64" Except 值（如果它们不在 cidr 范围外），则会被拒绝

18.4.1.14. .spec.ingress[].ports

描述

端口是应在为此规则选择的 pod 上访问的端口列表。此列表中的每个项目都使用逻辑 OR 合并。如果此字段为空或缺失，则该规则匹配所有端口（不受端口限制的流量）。如果存在此字段并至少包含一个项目，则此规则仅在流量与列表中至少匹配一个端口时才允许流量。

类型

array

18.4.1.15. .spec.ingress[].ports[]

描述

NetworkPolicyPort 描述了允许流量的端口

类型

object

属性	类型	描述
endPort	整数	endPort 表示策略应允许端口的范围从端口到 endPort。如果没有定义 port 字段，或者 port 字段定义为 named（字符串）端口，则无法定义此字段。endPort 必须等于或大于 port。
port	IntOrString	port 代表给定协议上的端口。这可以是 pod 上的数字或指定端口。如果没有提供此字段，这将匹配所有端口名称和数字。如果存在，则仅匹配指定协议和端口上的流量。
protocol	string	protocol 代表流量必须匹配的协议 (TCP、UDP 或 SCTP)。如果没有指定，此字段默认为 TCP。 可能的值有： - "SCTP" 是 SCTP 协议。 - "TCP" 是 TCP 协议。 - "UDP" 是 UDP 协议。

18.4.1.16. .status

描述

NetworkPolicyStatus 描述了 NetworkPolicy 的当前状态。

类型

object

属性	类型	描述
conditions	数组(Condition)	conditions 包含一组 metav1.Condition，用于描述 NetworkPolicy 的状态。当前服务状态

18.4.2. API 端点

可用的 API 端点如下：

- **/apis/networking.k8s.io/v1/networkpolicies**
 - **GET**: 列出或监视类型为 NetworkPolicy 的对象
- **/apis/networking.k8s.io/v1/watch/networkpolicies**
 - **GET**: 观察单个对 NetworkPolicy 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies**
 - **DELETE** : 删除 NetworkPolicy 集合
 - **GET**: 列出或监视类型为 NetworkPolicy 的对象
 - **POST** : 创建 NetworkPolicy
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies**
 - **GET**: 观察单个对 NetworkPolicy 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}**
 - **DELETE** : 删除 NetworkPolicy
 - **GET**: 读取指定的 NetworkPolicy
 - **PATCH** : 部分更新指定的 NetworkPolicy
 - **PUT** : 替换指定的 NetworkPolicy
- **/apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies/{name}**
 - **GET**: 观察对类型为 NetworkPolicy 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}/status**

- **GET**: 指定 NetworkPolicy 的读取状态
- **PATCH** : 部分更新指定 NetworkPolicy 的状态
- **PUT** : 替换指定 NetworkPolicy 的状态

18.4.2.1. /apis/networking.k8s.io/v1/networkpolicies

表 18.68. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视 NetworkPolicy 类型的对象

表 18.69. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicyList 模式
401 - Unauthorized	空

18.4.2.2. /apis/networking.k8s.io/v1/watch/networkpolicies

表 18.70. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 NetworkPolicy 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 18.71. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.4.2.3. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies

表 18.72. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.73. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除 NetworkPolicy 集合

表 18.74. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 18.75. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 18.76. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 NetworkPolicy 类型的对象

表 18.77. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 18.78. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicyList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 NetworkPolicy

表 18.79. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.80. 主体参数

参数	类型	描述
正文 (body)	NetworkPolicy 模式	

表 18.81. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
201 - Created	NetworkPolicy 模式
202 - Accepted	NetworkPolicy 模式
401 - Unauthorized	空

18.4.2.4. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies

表 18.82. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.83. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 NetworkPolicy 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 18.84. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.4.2.5. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}

表 18.85. 全局路径参数

参数	类型	描述
name	字符串	NetworkPolicy 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 18.86. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 NetworkPolicy

表 18.87. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 18.88. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 18.89. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 NetworkPolicy

表 18.90. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>NetworkPolicy</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 NetworkPolicy

表 18.91. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 18.92. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 18.93. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
201 - Created	NetworkPolicy 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 NetworkPolicy

表 18.94. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.95. 主体参数

参数	类型	描述
正文 (body)	NetworkPolicy 模式	

表 18.96. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
201 - Created	NetworkPolicy 模式
401 - Unauthorized	空

18.4.2.6. /apis/networking.k8s.io/v1/watch/namespaces/{namespace}/networkpolicies/{name}

表 18.97. 全局路径参数

参数	类型	描述
name	字符串	NetworkPolicy 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 18.98. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

监视对 kind NetworkPolicy 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 18.99. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

18.4.2.7. /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}/stat

表 18.100. 全局路径参数

参数	类型	描述
name	字符串	NetworkPolicy 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 18.101. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 NetworkPolicy 的读取状态

表 18.102. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 NetworkPolicy 的部分更新状态

表 18.103. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 18.104. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 18.105. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
201 - Created	NetworkPolicy 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 NetworkPolicy 的状态

表 18.106. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 18.107. 主体参数

参数	类型	描述
正文 (body)	NetworkPolicy 模式	

表 18.108. HTTP 响应

HTTP 代码	响应正文
200 - OK	NetworkPolicy 模式
201 - Created	NetworkPolicy 模式
401 - Unauthorized	空

第 19 章 节点 API

19.1. 节点 API

19.1.1. RuntimeClass [node.k8s.io/v1]

描述

RuntimeClass 定义集群中支持的容器运行时。RuntimeClass 用于决定使用哪个容器运行时来运行 pod 中的所有容器。runtimeclasses 由用户或集群置备程序手动定义，并在 PodSpec 中引用。Kubelet 负责在运行 pod 前解析 RuntimeClassName 引用。如需了解更多详细信息，请参阅 <https://kubernetes.io/docs/concepts/containers/runtime-class/>

类型

对象

19.2. RUNTIMECLASS [NODE.K8S.IO/V1]

描述

RuntimeClass 定义集群中支持的容器运行时。RuntimeClass 用于决定使用哪个容器运行时来运行 pod 中的所有容器。runtimeclasses 由用户或集群置备程序手动定义，并在 PodSpec 中引用。Kubelet 负责在运行 pod 前解析 RuntimeClassName 引用。如需了解更多详细信息，请参阅 <https://kubernetes.io/docs/concepts/containers/runtime-class/>

类型

object

必填

- handler

19.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
handler	string	handler 指定 CRI 实施将用于处理此类 pod 的底层运行时和配置。可能的值特定于节点和 CRI 配置。假设每个节点上都可用所有处理程序，并且同一名称的处理程序都相当于每个节点。例如，名为 "runc" 的处理程序可以指定 runc OCI 运行时（使用原生 Linux 容器）将用于在 pod 中运行容器。处理程序必须是小写，符合 DNS 标签(RFC 1123)要求，且不可变。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
开销	对象	开销结构代表与运行 pod 关联的资源开销。
调度	对象	调度指定支持 RuntimeClass 的节点的调度限制。

19.2.1.1. .overhead

描述

开销结构代表与运行 pod 关联的资源开销。

类型

object

属性	类型	描述
podFixed	对象（数量）	podFixed 代表与运行 pod 关联的固定资源开销。

19.2.1.2. .scheduling

描述

调度指定支持 RuntimeClass 的节点的调度限制。

类型

object

属性	类型	描述
nodeSelector	对象 (字符串)	nodeSelector 列出在支持此 RuntimeClass 的节点上必须存在的标签。使用此 RuntimeClass 的 Pod 只能调度到与此选择器匹配的节点。RuntimeClass nodeSelector 与 pod 的现有 nodeSelector 合并。任何冲突将导致 pod 在准入中被拒绝。
容限 (tolerations)	数组(Toleration)	容限在准入过程中将 (不包括重复) 附加到使用此 RuntimeClass 运行的 pod, 从而有效地取消 pod 和 RuntimeClass 容许的节点集合。

19.2.2. API 端点

可用的 API 端点如下：

- **/apis/node.k8s.io/v1/runtimeclasses**
 - **DELETE** : 删除 RuntimeClass 的集合
 - **GET**: 列出或监视类型为 RuntimeClass 的对象
 - **POST**: 创建 RuntimeClass
- **/apis/node.k8s.io/v1/watch/runtimeclasses**
 - **GET**: 观察单个对 RuntimeClass 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/node.k8s.io/v1/runtimeclasses/{name}**
 - **DELETE** : 删除 RuntimeClass
 - **GET**: 读取指定的 RuntimeClass
 - **PATCH** : 部分更新指定的 RuntimeClass
 - **PUT** : 替换指定的 RuntimeClass
- **/apis/node.k8s.io/v1/watch/runtimeclasses/{name}**
 - **GET**: 观察对 RuntimeClass 类型的对象的更改。已弃用：使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

19.2.2.1. /apis/node.k8s.io/v1/runtimeclasses

表 19.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RuntimeClass 的集合

表 19.2. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时, 应设置 continue 选项。由于这个值是定义的服务器, 因此客户端只能使用之前查询结果中的 continue 值, 并带有相同的查询参数 (除 continue 值除外), 因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效, 无论是因为过期时间 (通常为 5 到十五分钟) 还是服务器上的配置更改, 服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表, 则必须重启其列表, 而无需 continue 字段。否则, 客户端可能会发送另一个带有 410 错误的令牌的列表请求, 服务器将使用从下一个密钥开始的列表进行响应, 但从最新的快照开始, 从上一个列表结果 (创建、修改或删除) 后, 创建、修改或删除第一个列表请求将包含在响应中, 只要它们的键位于"下一密钥"后。</p> <p>当监视为 true 时, 不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视, 而不错过任何修改。</p>
dryRun	字符串	<p>出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器, 用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 19.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 19.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 RuntimeClass 的对象

表 19.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 19.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	RuntimeClassList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 RuntimeClass

表 19.7. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 19.8. 主体参数

参数	类型	描述
正文 (body)	RuntimeClass 模式	

表 19.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	RuntimeClass 模式
201 - Created	RuntimeClass 模式
202 - Accepted	RuntimeClass 模式
401 - Unauthorized	空

19.2.2.2. /apis/node.k8s.io/v1/watch/runtimeclasses

表 19.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 RuntimeClass 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 19.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

19.2.2.3. /apis/node.k8s.io/v1/runtimeclasses/{name}

表 19.12. 全局路径参数

参数	类型	描述
name	字符串	RuntimeClass 的名称

表 19.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RuntimeClass

表 19.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 19.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 19.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 RuntimeClass

表 19.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	RuntimeClass 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 RuntimeClass

表 19.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 19.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 19.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	RuntimeClass 模式
201 - Created	RuntimeClass 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 `RuntimeClass`

表 19.21. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
<code>fieldValidation</code>	string	<code>fieldValidation</code> 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 <code>BadRequest</code> 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 19.22. 主体参数

参数	类型	描述
正文 (body)	<code>RuntimeClass</code> 模式	

表 19.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>RuntimeClass</code> 模式
201 - Created	<code>RuntimeClass</code> 模式
401 - Unauthorized	空

19.2.2.4. `/apis/node.k8s.io/v1/watch/runtimeclasses/{name}`

表 19.24. 全局路径参数

参数	类型	描述
name	字符串	RuntimeClass 的名称

表 19.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind RuntimeClass.deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 19.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 20 章 策略 API

20.1. 策略 API

20.1.1. 驱除 [policy/v1]

描述

驱除根据某些策略和安全限制从节点驱除 pod。这是 Pod 的子资源。通过 POST 到 `.../pods/<pod name>/evictions` 来创建导致此类驱除的请求。

类型

对象

20.1.2. PodDisruptionBudget [policy/v1]

描述

PodDisruptionBudget 是一个对象，用于定义可导致 pod 集合的最大中断

类型

对象

20.2. 驱除 [POLICY/V1]

描述

驱除根据某些策略和安全限制从节点驱除 pod。这是 Pod 的子资源。通过 POST 到 `.../pods/<pod name>/evictions` 来创建导致此类驱除的请求。

类型

对象

20.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
deleteOptions	DeleteOptions	可以提供 DeleteOptions

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta 描述了正在被驱除的 pod。

20.2.2. API 端点

可用的 API 端点如下：

- **/api/v1/namespaces/{namespace}/pods/{name}/eviction**
 - **POST**：创建 Pod 的驱除

20.2.2.1. /api/v1/namespaces/{namespace}/pods/{name}/eviction

表 20.1. 全局路径参数

参数	类型	描述
name	字符串	Eviction 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 20.2. 全局查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

POST

描述

创建 Pod 的驱除

表 20.3. 主体参数

参数	类型	描述
正文 (body)	eviction 模式	

表 20.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	eviction 模式
201 - Created	eviction 模式
202 - Accepted	eviction 模式
401 - Unauthorized	空

20.3. PODDISRUPTIONBUDGET [POLICY/V1]

描述

PodDisruptionBudget 是一个对象，用于定义可导致 pod 集合的最大中断

类型

对象

20.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	对象	PodDisruptionBudgetSpec 是 PodDisruptionBudget 的描述。
status	对象	PodDisruptionBudgetStatus 代表 PodDisruptionBudget 状态的信息。状态可能会跟踪系统的实际状态。

20.3.1.1. .spec

描述

PodDisruptionBudgetSpec 是 PodDisruptionBudget 的描述。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
maxUnavailable	IntOrString	如果在驱除后由 "selector" 选择的大部分 "maxUnavailable" pod 都不可用，例如，即使没有被驱除的 pod，也被允许。例如，可以通过指定 0 来防止所有自愿驱除。这是与 "minAvailable" 相互排斥的设置。
minAvailable	IntOrString	如果驱除后至少会存在 "selector" 选择的 "minAvailable" pod，如没有被驱除的 pod 仍可用。例如，您可以通过指定 "100%" 来防止所有自愿驱除。
selector	labelSelector	对驱除由中断预算管理的 pod 进行标签查询。null 选择器不匹配任何 pod，而空({})选择器会选择命名空间中的所有 pod。

属性	类型	描述
unhealthyPodEvictionPolicy	string	<p>unhealthyPodEvictionPolicy 定义不健康 pod 应该考虑驱除的条件。当前实施认为健康的 pod，作为带有 type="Ready",status="True" 的 status.conditions 项的 pod。</p> <p>有效策略为 IfHealthyBudget 和 AlwaysAllow。如果没有指定策略，则使用默认为，对应于 IfHealthyBudget 策略。</p> <p>IfHealthyBudget 策略意味着运行 pod (status.phase="Running")，但只有在保护的应用程序没有中断时(status.currentHealthy 至少等于 status.desiredHealthy)，但还没有处于健康状态。健康的 pod 会受到 PDB 进行驱除的影响。</p> <p>AlwaysAllow 策略意味着所有正在运行的 pod (status.phase="Running")，但还没有处于健康状态，无论是否满足 PDB 中的条件，都可能被驱除。这意味着，运行中断的应用程序的 pod 可能无法处于健康状态。健康的 pod 会受到 PDB 进行驱除的影响。</p> <p>以后可能会添加其他策略。如果客户端在此字段中遇到未识别的策略，则应禁止驱除不健康的 pod。</p> <p>此字段是 beta 级。当功能门 PDBUnhealthyPodEvictionPolicy 被启用（默认启用）时，驱除 API 会使用此字段。</p> <p>可能枚举值： - "AlwaysAllow" 策略意味着所有正在运行的 pod (status.phase="Running")，但还没有处于健康状态，无论是否满足 PDB 中的条件，都可以被驱除。这意味着，运行中断的应用程序的 pod 可能无法处于健康状态。健康的 pod 会受到 PDB 进行驱除的影响。 - "IfHealthyBudget" 策略意味着运行 pod (status.phase="Running")，但还没有处于健康状态，只有在保护的应用程序没有被破坏时 (status.currentHealthy 至少等于</p>

属性	类型	描述
		status.desiredHealthy)。健康的 pod 会受到 PDB 进行驱逐的影响。

20.3.1.2. .status

描述

PodDisruptionBudgetStatus 代表 PodDisruptionBudget 状态的信息。状态可能会跟踪系统的实际状态。

类型

object

必填

- **disruptionsAllowed**
- **currentHealthy**
- **desiredHealthy**
- **expectedPods**

属性	类型	描述
conditions	数组(Condition)	条件包含 PDB 的条件。disruption 控制器设置 DisruptionAllowed 条件。以下是 reason 字段的已知值（将来可能会添加其他原因）：- SyncFailed：控制器遇到错误，且无法计算允许中断的数量。因此，不允许中断，条件的状态为 False。- InsufficientPods: pod 的数量可以是 at 或 below PodDisruptionBudget 所需的数量。不允许中断，条件的状态为 False。- SufficientPods: PodDisruptionBudget 所需的 pod 数量多。该条件为 True，允许中断的数量由 disruptionsAllowed 属性提供。
currentHealthy	整数	当前健康的 pod 数量
desiredHealthy	整数	所需健康 pod 的最低数量

属性	类型	描述
disruptedPods	对象(Time)	DisruptedPods 包含由 API 服务器驱除子资源处理程序处理但未被 PodDisruptionBudget 控制器观察到的 pod 的信息。当 API 服务器处理驱除请求时，pod 将处于此映射中，当 pod 被 PDB 控制器认为删除（或超时时）时，pod 将处于这个映射中。映射中的键是 pod 的名称，值是 API 服务器处理驱除请求的时间。如果删除没有发生，并且 pod 仍然会被 PodDisruptionBudget 控制器自动从列表中删除。如果一切顺利，该映射在大部分时间内都应为空。映射中的大量条目可能表示 pod 删除问题。
disruptionsAllowed	整数	当前允许的 pod 中断预算。
expectedPods	整数	此中断预算计算的 pod 总数
observedGeneration	整数	更新此 PDB 状态时观察到的最最近生成。只有 observedGeneration 等于 PDB 对象生成时，DisruptionsAllowed 和其他状态信息才有效。

20.3.2. API 端点

可用的 API 端点如下：

- **/apis/policy/v1/poddisruptionbudgets**
 - **GET**: 列出或监视类型为 PodDisruptionBudget 的对象
- **/apis/policy/v1/watch/poddisruptionbudgets**
 - **GET**: 观察单个对 PodDisruptionBudget 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets**
 - **DELETE**：删除 PodDisruptionBudget 的集合
 - **GET**: 列出或监视类型为 PodDisruptionBudget 的对象
 - **POST**：创建一个 PodDisruptionBudget
- **/apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets**
 - **GET**: 观察单个对 PodDisruptionBudget 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

- **/apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}**
 - **DELETE** : 删除 PodDisruptionBudget
 - **GET** : 读取指定的 PodDisruptionBudget
 - **PATCH** : 部分更新指定的 PodDisruptionBudget
 - **PUT** : 替换指定的 PodDisruptionBudget
- **/apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets/{name}**
 - **GET**: 观察对类型为 PodDisruptionBudget 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
- **/apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/status**
 - **GET** : 指定 PodDisruptionBudget 的读取状态
 - **PATCH** : 部分更新指定 PodDisruptionBudget 的状态
 - **PUT** : 替换指定 PodDisruptionBudget 的状态

20.3.2.1. /apis/policy/v1/poddisruptionbudgets

表 20.5. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 PodDisruptionBudget 的对象

表 20.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudgetList 模式
401 - Unauthorized	空

20.3.2.2. /apis/policy/v1/watch/poddisruptionbudgets

表 20.7. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PodDisruptionBudget 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 20.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

20.3.2.3. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets

表 20.9. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 20.10. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PodDisruptionBudget 的集合

表 20.11. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 20.12. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 20.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PodDisruptionBudget 的对象

表 20.14. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 20.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudgetList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 PodDisruptionBudget

表 20.16. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 20.17. 主体参数

参数	类型	描述
正文 (body)	PodDisruptionBudget 模式	

表 20.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
201 - Created	PodDisruptionBudget 模式
202 - Accepted	PodDisruptionBudget 模式
401 - Unauthorized	空

20.3.2.4. /apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets

表 20.19. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 20.20. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 PodDisruptionBudget 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 20.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

20.3.2.5. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}

表 20.22. 全局路径参数

参数	类型	描述
name	字符串	PodDisruptionBudget 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 20.23. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PodDisruptionBudget

表 20.24. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 20.25. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 20.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 PodDisruptionBudget

表 20.27. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>PodDisruptionBudget</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 PodDisruptionBudget

表 20.28. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 20.29. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 20.30. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
201 - Created	PodDisruptionBudget 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 PodDisruptionBudget

表 20.31. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 20.32. 主体参数

参数	类型	描述
正文 (body)	PodDisruptionBudget 模式	

表 20.33. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
201 - Created	PodDisruptionBudget 模式
401 - Unauthorized	空

20.3.2.6. /apis/policy/v1/watch/namespaces/{namespace}/poddisruptionbudgets/{name}

表 20.34. 全局路径参数

参数	类型	描述
name	字符串	PodDisruptionBudget 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 20.35. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind PodDisruptionBudget. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 20.36. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

20.3.2.7. /apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets/{name}/status

表 20.37. 全局路径参数

参数	类型	描述
name	字符串	PodDisruptionBudget 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 20.38. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定的 PodDisruptionBudget 的读取状态

表 20.39. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 PodDisruptionBudget 的部分更新状态

表 20.40. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为： <ul style="list-style-type: none"> - Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 20.41. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 20.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
201 - Created	PodDisruptionBudget 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 PodDisruptionBudget 的状态

表 20.43. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为： <ul style="list-style-type: none"> - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为 - Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 20.44. 主体参数

参数	类型	描述
正文 (body)	PodDisruptionBudget 模式	

表 20.45. HTTP 响应

HTTP 代码	响应正文
200 - OK	PodDisruptionBudget 模式
201 - Created	PodDisruptionBudget 模式
401 - Unauthorized	空

第 21 章 RBAC API

21.1. RBAC API

21.1.1. ClusterRole [rbac.authorization.k8s.io/v1]

描述

ClusterRole 是一个集群级别，逻辑分组的 PolicyRules，可通过 RoleBinding 或 ClusterRoleBinding 引用为单元。

类型

对象

21.1.2. ClusterRoleBinding [rbac.authorization.k8s.io/v1]

描述

ClusterRoleBinding 引用一个 ClusterRole，但不包含它。它可以引用全局命名空间中的 ClusterRole，并通过 Subject 添加哪些信息。

类型

对象

21.1.3. Role [rbac.authorization.k8s.io/v1]

描述

role 是一个命名空间的逻辑分组的 PolicyRules，可通过 RoleBinding 引用为单元。

类型

对象

21.1.4. RoleBinding [rbac.authorization.k8s.io/v1]

描述

RoleBinding 引用角色，但不包含它。它可以引用同一命名空间中的角色，或全局命名空间中的 ClusterRole。它通过 Subjects 和 namespace 信息添加它所在的命名空间的信息。给定命名空间中的 rolebindings 仅在该命名空间中生效。

类型

对象

21.2. CLUSTERROLE [RBAC.AUTHORIZATION.K8S.IO/V1]

描述

ClusterRole 是一个集群级别，逻辑分组的 PolicyRules，可通过 RoleBinding 或 ClusterRoleBinding 引用为单元。

类型

对象

21.2.1. 规格

属性	类型	描述
aggregationRule	对象	AggregationRule 描述了如何找到要聚合到 ClusterRole 的 ClusterRole
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。
rules	array	规则包含此 ClusterRole 的所有 PolicyRules
rules[]	object	PolicyRule 包含描述策略规则的信息，但不包含有关规则应用到谁或规则应用到哪个命名空间的信息。

21.2.1.1. .aggregationRule

描述

AggregationRule 描述了如何找到要聚合到 ClusterRole 的 ClusterRole

类型

object

属性	类型	描述
clusterRoleSelectors	数组(LabelSelector)	ClusterRoleSelectors 包含用于查找 ClusterRole 并创建规则的选择器列表。如果有任何选择器匹配，则会添加 ClusterRole 的权限

21.2.1.2. .rules

描述

规则包含此 ClusterRole 的所有 PolicyRules

类型

array

21.2.1.3. .rules[]

描述

PolicyRule 包含描述策略规则的信息，但不包含有关规则应用到谁或规则应用到哪个命名空间的信息。

类型

object

必填

- **verbs**

属性	类型	描述
apiGroups	数组（字符串）	APIGroups 是包含资源的 APIGroup 的名称。如果指定了多个 API 组，则允许针对任何 API 组中的一个枚举资源请求的任何操作。"" 代表核心 API 组，"*"代表所有 API 组。
nonResourceURLs	数组（字符串）	NonResourceURLs 是用户应该有权访问的部分 url 的集合。* 被允许，但只有在路径 Since 非资源 URL 中的完整、最后一个步骤，此字段仅适用于从 ClusterRoleBinding 引用的 ClusterRole。规则可以应用到 API 资源（如 "pods" 或 "secrets"）或非资源 URL 路径（如 "/api"），但不能同时应用。
resourceNames	数组（字符串）	resourceNames 是一个可选的规则应用到的名称白名单。空集表示所有都会被允许。
resources	数组（字符串）	resources 是此规则应用到的资源列表。'*' 代表所有资源。
verbs	数组（字符串）	verbs 是一个 Verbs 列表，适用于此规则中包含的 ALL ResourceKinds。'*' 代表所有动词。

21.2.2. API 端点

可用的 API 端点如下：

- **/apis/rbac.authorization.k8s.io/v1/clusterroles**
 - **DELETE** : 删除 ClusterRole 的集合
 - **GET**: 列出或监视类型为 ClusterRole 的对象
 - **POST** : 创建 ClusterRole
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterroles**
 - **GET**: 观察单个对 ClusterRole 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/clusterroles/{name}**
 - **DELETE** : 删除 ClusterRole
 - **GET** : 读取指定的 ClusterRole
 - **PATCH** : 部分更新指定的 ClusterRole
 - **PUT** : 替换指定的 ClusterRole
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterroles/{name}**
 - **GET**: 观察对类型为 ClusterRole 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

21.2.2.1. /apis/rbac.authorization.k8s.io/v1/clusterroles

表 21.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ClusterRole 集合

表 21.2. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 21.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ClusterRole 的对象

表 21.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 21.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 ClusterRole

表 21.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.8. 主体参数

参数	类型	描述
正文 (body)	ClusterRole 模式	

表 21.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRole 模式
201 - Created	ClusterRole 模式
202 - Accepted	ClusterRole 模式
401 - Unauthorized	空

21.2.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles

表 21.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ClusterRole 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.2.2.3. /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}

表 21.12. 全局路径参数

参数	类型	描述
name	字符串	ClusterRole 的名称

表 21.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ClusterRole

表 21.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 21.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

读取指定的 ClusterRole

表 21.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRole 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 ClusterRole

表 21.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 21.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 21.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRole 模式
201 - Created	ClusterRole 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ClusterRole

表 21.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.22. 主体参数

参数	类型	描述
正文 (body)	ClusterRole 模式	

表 21.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRole 模式
201 - Created	ClusterRole 模式
401 - Unauthorized	空

21.2.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterroles/{name}

表 21.24. 全局路径参数

参数	类型	描述
name	字符串	ClusterRole 的名称

表 21.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法**GET****描述**

观察对类型为 ClusterRole 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 21.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.3. CLUSTERROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]

描述

ClusterRoleBinding 引用一个 ClusterRole，但不包含它。它可以引用全局命名空间中的 ClusterRole，并通过 Subject 添加哪些信息。

类型

object

必填

- **roleRef**

21.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。

属性	类型	描述
roleRef	对象	roleRef 包含指向正在使用的角色的信息
主题	array	subjects 包含角色应用到的对象的引用。
subjects[]	对象	subject 包含对角色绑定应用到的对象或用户身份的引用。这可以保存直接 API 对象引用，也可以是非对象（如用户和组名称）的值。

21.3.1.1. .roleRef

描述

roleRef 包含指向正在使用的角色的信息

类型

object

必填

- **apiGroup**
- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

21.3.1.2. .subjects

描述

subjects 包含角色应用到的对象的引用。

类型

array

21.3.1.3. .subjects[]

描述

subject 包含对角色绑定应用到的对象或用户身份的引用。这可以保存直接 API 对象引用，也可以是非对象（如用户和组名称）的值。

类型

object

必填

- kind
- name

属性	类型	描述
apiGroup	字符串	apiGroup 包含引用主题的 API 组。对于 ServiceAccount 主题，默认为 ""。对于 User 和 Group 主题，默认为 "rbac.authorization.k8s.io"。
kind	字符串	被引用的对象种类。此 API 组定义的值有 "User", "Group", 和 "ServiceAccount"。如果 Authorizer 无法识别 kind 值，Authorizer 应该会报告错误。
name	字符串	被引用的对象的名称。
namespace	字符串	所引用对象的命名空间。如果对象类型不是命名空间，如 "User" 或 "Group"，则这个值不为空，则 Authorizer 应该报告错误。

21.3.2. API 端点

可用的 API 端点如下：

- **/apis/rbac.authorization.k8s.io/v1/clusterrolebindings**
 - **DELETE** : 删除 ClusterRoleBinding 的集合
 - **GET**: 列出或监视类型为 ClusterRoleBinding 的对象
 - **POST**: 创建 ClusterRoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings**
 - **GET**: 观察单个对 ClusterRoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}**
 - **DELETE** : 删除 ClusterRoleBinding
 - **GET**: 读取指定的 ClusterRoleBinding
 - **PATCH** : 部分更新指定的 ClusterRoleBinding

- **PUT** : 替换指定的 ClusterRoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings/{name}**
 - **GET**: 观察对类型为 ClusterRoleBinding 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

21.3.2.1. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings

表 21.27. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ClusterRoleBinding 的集合

表 21.28. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>

参数	类型	描述
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 21.29. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.30. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ClusterRoleBinding 的对象

表 21.31. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 21.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleBindingList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 ClusterRoleBinding

表 21.33. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.34. 主体参数

参数	类型	描述
正文 (body)	ClusterRoleBinding 模式	

表 21.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleBinding 模式
201 - Created	ClusterRoleBinding 模式
202 - Accepted	ClusterRoleBinding 模式
401 - Unauthorized	空

21.3.2.2. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings

表 21.36. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ClusterRoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.37. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.3.2.3. /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}

表 21.38. 全局路径参数

参数	类型	描述
name	字符串	ClusterRoleBinding 的名称

表 21.39. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ClusterRoleBinding

表 21.40. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 21.41. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 ClusterRoleBinding

表 21.43. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleBinding 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 ClusterRoleBinding

表 21.44. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 21.45. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 21.46. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleBinding 模式
201 - Created	ClusterRoleBinding 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ClusterRoleBinding

表 21.47. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.48. 主体参数

参数	类型	描述
正文 (body)	ClusterRoleBinding 模式	

表 21.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	ClusterRoleBinding 模式
201 - Created	ClusterRoleBinding 模式
401 - Unauthorized	空

21.3.2.4. /apis/rbac.authorization.k8s.io/v1/watch/clusterrolebindings/{name}

表 21.50. 全局路径参数

参数	类型	描述
name	字符串	ClusterRoleBinding 的名称

表 21.51. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind ClusterRoleBinding. deprecated 的对象的更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 21.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.4. ROLE [RBAC.AUTHORIZATION.K8S.IO/V1]

描述

role 是一个命名空间的逻辑分组的 PolicyRules，可通过 RoleBinding 引用为单元。

类型**对象****21.4.1. 规格**

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。
rules	array	规则包含此角色的所有 PolicyRules
rules[]	object	PolicyRule 包含描述策略规则的信息，但不包含有关规则应用到谁或规则应用到哪个命名空间的信息。

21.4.1.1. .rules**描述**

规则包含此角色的所有 PolicyRules

类型

array

21.4.1.2. .rules[]**描述**

PolicyRule 包含描述策略规则的信息，但不包含有关规则应用到谁或规则应用到哪个命名空间的信息。

类型

object

必填

- verbs

属性	类型	描述
apiGroups	数组 (字符串)	APIGroups 是包含资源的 APIGroup 的名称。如果指定了多个 API 组, 则允许针对任何 API 组中的一个枚举资源请求的任何操作。"" 代表核心 API 组, "*"代表所有 API 组。
nonResourceURLs	数组 (字符串)	NonResourceURLs 是用户应该有权访问的部分 url 的集合。* 被允许, 但只有在路径 Since 非资源 URL 中的完整、最后一个步骤, 此字段仅适用于从 ClusterRoleBinding 引用的 ClusterRole。规则可以应用到 API 资源 (如 "pods" 或 "secrets") 或非资源 URL 路径 (如 "/api"), 但不能同时应用。
resourceNames	数组 (字符串)	resourceNames 是一个可选的规则应用到的名称白名单。空集表示所有都会被允许。
resources	数组 (字符串)	resources 是此规则应用到的资源列表。'*' 代表所有资源。
verbs	数组 (字符串)	verbs 是一个 Verbs 列表, 适用于此规则中包含的 ALL ResourceKinds。'*' 代表所有动词。

21.4.2. API 端点

可用的 API 端点如下：

- **/apis/rbac.authorization.k8s.io/v1/roles**
 - **GET**: 列出或监视类型为 Role 的对象
- **/apis/rbac.authorization.k8s.io/v1/watch/roles**
 - **GET**: 观察单个对角色列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles**
 - **DELETE**：删除角色集合

- **GET**: 列出或监视类型为 Role 的对象
- **POST** : 创建角色
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles**
 - **GET**: 观察单个对角色列表的更改。已弃用 : 改为使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}**
 - **DELETE** : 删除角色
 - **GET** : 读取指定的角色
 - **PATCH** : 部分更新指定的角色
 - **PUT** : 替换指定的角色
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles/{name}**
 - **GET**: 观察对类型为 Role 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

21.4.2.1. /apis/rbac.authorization.k8s.io/v1/roles

表 21.53. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签, 由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回, 也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视, 则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时, 应设置 continue 选项。由于这个值是定义的服务器, 因此客户端只能使用之前查询结果中的 continue 值, 并带有相同的查询参数 (除 continue 值除外), 因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效, 无论是因为过期时间 (通常为 5 到十五分钟) 还是服务器上的配置更改, 服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表, 则必须重启其列表, 而无需 continue 字段。否则, 客户端可能会发送另一个带有 410 错误的令牌的列表请求, 服务器将使用从下一个密钥开始的列表进行响应, 但从最新的快照开始, 从上一个列表结果 (创建、修改或删除) 后, 创建、修改或删除第一个列表请求将包含在响应中, 只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时, 不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视, 而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 Role 的对象

表 21.54. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleList 模式
401 - Unauthorized	空

21.4.2.2. /apis/rbac.authorization.k8s.io/v1/watch/roles

表 21.55. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Role 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.56. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.4.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles

表 21.57. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.58. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法**DELETE****描述**

删除角色集合

表 21.59. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 21.60. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.61. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 Role 的对象

表 21.62. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 21.63. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建角色

表 21.64. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.65. 主体参数

参数	类型	描述
正文 (body)	Role 模式	

表 21.66. HTTP 响应

HTTP 代码	响应正文
200 - OK	Role 模式
201 - Created	Role 模式
202 - Accepted	Role 模式
401 - Unauthorized	空

21.4.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles

表 21.67. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.68. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 Role 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.69. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.4.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}

表 21.70. 全局路径参数

参数	类型	描述
name	字符串	角色的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.71. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除角色

表 21.72. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 21.73. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 21.74. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的角色

表 21.75. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Role</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的角色

表 21.76. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 21.77. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 21.78. HTTP 响应

HTTP 代码	响应正文
200 - OK	Role 模式
201 - Created	Role 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的角色

表 21.79. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.80. 主体参数

参数	类型	描述
正文 (body)	Role 模式	

表 21.81. HTTP 响应

HTTP 代码	响应正文
200 - OK	Role 模式
201 - Created	Role 模式
401 - Unauthorized	空

21.4.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/roles/{name}

表 21.82. 全局路径参数

参数	类型	描述
name	字符串	角色的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 21.83. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。 - 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind Role. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 21.84. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.5. ROLEBINDING [RBAC.AUTHORIZATION.K8S.IO/V1]

描述

RoleBinding 引用角色，但不包含它。它可以引用同一命名空间中的角色，或全局命名空间中的 ClusterRole。它通过 Subjects 和 namespace 信息添加它所在的命名空间的信息。给定命名空间中的 rolebindings 仅在该命名空间中生效。

类型

object

必填

- **roleRef**

21.5.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。
roleRef	对象	roleRef 包含指向正在使用的角色的信息
主题	array	subjects 包含角色应用到的对象的引用。
subjects[]	对象	subject 包含对角色绑定应用到的对象或用户身份的引用。这可以保存直接 API 对象引用，也可以是非对象（如用户和组名称）的值。

21.5.1.1. .roleRef

描述

roleRef 包含指向正在使用的角色的信息

类型

object

必填

- **apiGroup**
- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 是被引用资源的组
kind	字符串	kind 是被引用的资源类型
name	字符串	name 是被引用的资源的名称

21.5.1.2. .subjects

描述

subjects 包含角色应用到的对象的引用。

类型

array

21.5.1.3. .subjects[]

描述

subject 包含对角色绑定应用到的对象或用户身份的引用。这可以保存直接 API 对象引用，也可以是非对象（如用户和组名称）的值。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
apiGroup	字符串	apiGroup 包含引用主题的 API 组。对于 ServiceAccount 主题，默认为 ""。对于 User 和 Group 主题，默认为 "rbac.authorization.k8s.io"。

属性	类型	描述
kind	字符串	被引用的对象种类。此 API 组定义的值有 "User", "Group", 和 "ServiceAccount"。如果 Authorizer 无法识别 kind 值, Authorizer 应该会报告错误。
name	字符串	被引用的对象的名称。
namespace	字符串	所引用对象的命名空间。如果对象类型不是命名空间, 如 "User" 或 "Group", 则这个值不为空, 则 Authorizer 应该报告错误。

21.5.2. API 端点

可用的 API 端点如下：

- **/apis/rbac.authorization.k8s.io/v1/rolebindings**
 - **GET**: 列出或监视类型为 RoleBinding 的对象
- **/apis/rbac.authorization.k8s.io/v1/watch/rolebindings**
 - **GET**: 观察单个对 RoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings**
 - **DELETE** : 删除 RoleBinding 的集合
 - **GET**: 列出或监视类型为 RoleBinding 的对象
 - **POST** : 创建 RoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings**
 - **GET**: 观察单个对 RoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}**
 - **DELETE** : 删除 RoleBinding
 - **GET**: 读取指定的 RoleBinding
 - **PATCH** : 部分更新指定的 RoleBinding
 - **PUT** : 替换指定的 RoleBinding
- **/apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings/{name}**
 - **GET**: 观察对类型为 RoleBinding 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

21.5.2.1. /apis/rbac.authorization.k8s.io/v1/rolebindings

表 21.85. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 RoleBinding 的对象

表 21.86. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleBindingList 模式
401 - Unauthorized	空

21.5.2.2. /apis/rbac.authorization.k8s.io/v1/watch/rolebindings

表 21.87. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 RoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.88. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.5.2.3. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings

表 21.89. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.90. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RoleBinding 集合

表 21.91. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 21.92. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 21.93. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 RoleBinding 的对象

表 21.94. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 21.95. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleBindingList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 RoleBinding

表 21.96. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.97. 主体参数

参数	类型	描述
正文 (body)	RoleBinding 模式	

表 21.98. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleBinding 模式
201 - Created	RoleBinding 模式
202 - Accepted	RoleBinding 模式
401 - Unauthorized	空

21.5.2.4. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings

表 21.99. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.100. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 RoleBinding 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 21.101. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

21.5.2.5. /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

表 21.102. 全局路径参数

参数	类型	描述
name	字符串	RoleBinding 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 21.103. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RoleBinding

表 21.104. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 21.105. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 21.106. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 RoleBinding

表 21.107. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>RoleBinding</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 RoleBinding

表 21.108. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 21.109. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 21.110. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleBinding 模式
201 - Created	RoleBinding 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 RoleBinding

表 21.111. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 21.112. 主体参数

参数	类型	描述
正文 (body)	RoleBinding 模式	

表 21.113. HTTP 响应

HTTP 代码	响应正文
200 - OK	RoleBinding 模式
201 - Created	RoleBinding 模式
401 - Unauthorized	空

21.5.2.6. /apis/rbac.authorization.k8s.io/v1/watch/namespaces/{namespace}/rolebindings/{r

表 21.114. 全局路径参数

参数	类型	描述
name	字符串	RoleBinding 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 21.115. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind RoleBinding. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 21.116. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 22 章 网络 API

22.1. 网络 API

22.1.1. Route [route.openshift.io/v1]

描述

通过路由，开发人员可以通过 HTTP (S) 了解负载均衡和代理层通过公共 DNS 条目公开服务。路由可以进一步指定 TLS 选项和证书，或者指定路由器也应该接受 HTTP 和 HTTPS 流量的公共 CNAME。管理员通常将其路由器配置为在集群防火墙外可见，还可在服务内容上添加额外的安全性、缓存或流量控制。路由器通常直接与服务端点通信。创建路由后，可能不会更改 **host** 字段。通常，路由器在解析冲突时使用带有给定主机的最旧的路由。路由器受到额外的自定义，可以通过 annotations 字段支持其他控制。由于管理员可以配置多个路由器，因此路由状态字段用于向客户端返回每个路由器下路由的名称和状态的信息。如果客户端选择重复的名称，例如，则使用路由状态条件来指示无法选择路由。要在路由上启用 HTTP/2 ALPN，它需要自定义（非通配符）证书。这可防止客户端的连接并发，特别是 Web 浏览器。由于连接重新使用的风险，我们不支持使用默认证书的 HTTP/2 ALPN。没有其自身自定义证书的路由在前端或后端上都不会是启用了 HTTP/2 ALPN 的路由。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

对象

22.2. ROUTE [ROUTE.OPENSIFT.IO/V1]

描述

通过路由，开发人员可以通过 HTTP (S) 了解负载均衡和代理层通过公共 DNS 条目公开服务。路由可以进一步指定 TLS 选项和证书，或者指定路由器也应该接受 HTTP 和 HTTPS 流量的公共 CNAME。管理员通常将其路由器配置为在集群防火墙外可见，还可在服务内容上添加额外的安全性、缓存或流量控制。路由器通常直接与服务端点通信。创建路由后，可能不会更改 **host** 字段。通常，路由器在解析冲突时使用带有给定主机的最旧的路由。路由器受到额外的自定义，可以通过 annotations 字段支持其他控制。由于管理员可以配置多个路由器，因此路由状态字段用于向客户端返回每个路由器下路由的名称和状态的信息。如果客户端选择重复的名称，例如，则使用路由状态条件来指示无法选择路由。要在路由上启用 HTTP/2 ALPN，它需要自定义（非通配符）证书。这可防止客户端的连接并发，特别是 Web 浏览器。由于连接重新使用的风险，我们不支持使用默认证书的 HTTP/2 ALPN。没有其自身自定义证书的路由在前端或后端上都不会是启用了 HTTP/2 ALPN 的路由。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

object

必填

- spec

22.2.1. 规格

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	spec 是路由的所需状态
status	object	Status 是路由的当前状态

22.2.1.1. .spec

描述

spec 是路由的所需状态

类型

object

必填

- 至

属性	类型	描述
----	----	----

属性	类型	描述
alternateBackends	array	alternateBackends 允许最多 3 个额外的后端分配给路由。只允许 Service kind，它将默认为 Service。使用 RouteTargetReference 对象中的 weight 字段指定相对的首选项。
alternateBackends[]	对象	RouteTargetReference 指定解析为端点的目标。只允许 'Service' kind。使用 "weight" 字段加重其他字段。
主机	字符串	主机是一个指向该服务的别名/DNS。可选。如果没有指定路由名称，则通常会自动选择。必须遵循 DNS952 子域惯例。
path	字符串	路由器监视的路径，将流量路由到该服务的流量。选填
port	object	如果指定，路由器要使用的端口。大多数路由器默认使用服务公开的所有端口 - 设置此值以指示要使用的端口的路由器。
subdomain	字符串	子域是在入口控制器的域（作为子域）中请求的 DNS 子域。如果设置 host，则忽略此字段。入口控制器可能会选择忽略这个建议的名称，在这种情况下，控制器会在 status.ingress 数组中报告分配的名称，或拒绝接受该路由。如果设置了这个值，且服务器不支持此字段，主机则会自动填充。否则主机将留空。该字段可能有多个以点分开的部分，但并非所有入口控制器都遵守请求。创建后无法更改此字段，但具有更新路由/自定义主机权限的用户除外。示例：子域 frontend 会自动接收路由器子域 apps.mycluster.com ，以获得完整的主机名 frontend.apps.mycluster.com 。
tls	object	tls 字段提供为路由配置证书和终止的功能。

属性	类型	描述
至	object	to 是路由应用作主后端的对象。只允许 Service kind，它将默认为 Service。如果 weight 字段(0-256 默认 100)设为零，则不会将流量发送到此后端。
wildcardPolicy	字符串	用于路由的通配符策略。目前只允许 'Subdomain' 或 'None'。

22.2.1.2. .spec.alternateBackends

描述

alternateBackends 允许最多 3 个额外的后端分配给路由。只允许 Service kind，它将默认为 Service。使用 RouteTargetReference 对象中的 weight 字段指定相对的首选项。

类型

array

22.2.1.3. .spec.alternateBackends[]

描述

RouteTargetReference 指定解析为端点的目标。只允许 'Service' kind。使用 "weight" 字段加重其他字段。

类型

object

必填

- **kind**
- **name**

属性	类型	描述
kind	字符串	路由引用的目标种类。目前，只允许 'Service'
name	字符串	要引用的服务/目标的名称，如服务的名称
weight	整数	weight 作为 0 到 256 之间的整数，默认 100 指定针对其他目标引用对象的目标相对权重。0 会阻止请求进入此后端。

22.2.1.4. .spec.port

描述

如果指定，路由器要使用的端口。大多数路由器默认使用服务公开的所有端点 - 设置此值以指示要使用的端口的路由器。

类型

object

必填

- **targetPort**

属性	类型	描述
targetPort	integer-or-string	

22.2.1.5. .spec.tls

描述

tls 字段提供为路由配置证书和终止的功能。

类型

object

必填

- **termination**

属性	类型	描述
caCertificate	字符串	caCertificate 提供证书颁发机构证书内容
certificate	字符串	证书提供证书内容。这应该是单一服务证书，而不是证书链。不要包括 CA 证书。
destinationCACertificate	字符串	destinationCACertificate 提供最终目的地的 ca 证书的内容。使用重新加密终止时，应提供此文件，以便让路由器将它用于安全连接上的健康检查。如果没有指定此字段，路由器可能会提供自己的目标 CA，并使用短服务名称 (service.namespace.svc) 执行主机名验证，它允许基础架构生成的证书自动验证。

属性	类型	描述
insecureEdgeTerminationPolicy	字符串	insecureEdgeTerminationPolicy 表示路由不安全连接所需的行为。虽然每个路由器可能会对要公开的端口做出自己的决定，但通常这是端口 80。* allow - 流量发送到不安全端口上的服务器（仅用于边缘/重新加密终止）（默认）。* None - 在不安全的端口上不允许流量。* Redirect - 客户端被重定向到安全端口。
key	字符串	密钥提供密钥文件内容
termination	字符串	termination 表示终止类型。* edge - TLS 终止由路由器完成，http 用于与后端通信（默认）= 透传 - 在不提供 TLS 终止的路由器时直接发送到目的地，TLS 终止由路由器完成，https 用于与后端通信

22.2.1.6. .spec.to

描述

to 是路由应用作主后端的对象。只允许 Service kind，它将默认为 Service。如果 weight 字段(0-256 默认 100)设为零，则不会将流量发送到此后端。

类型

object

必填

- kind
- name

属性	类型	描述
kind	字符串	路由引用的目标种类。目前，只允许 'Service'
name	字符串	要引用的服务/目标的名称，如服务的名称
weight	整数	weight 作为 0 到 256 之间的整数，默认 100 指定针对其他目标引用对象的目标相对权重。0 会阻止请求进入此后端。

22.2.1.7. .status

描述

Status 是路由的当前状态

类型

object

属性	类型	描述
ingress	array	Ingress 描述了可公开路由的位置。ingress 点列表可能包含重复的主机或 RouterName 值。路由在就绪后被视为 Ready
ingress[]	对象	RouteIngress 包含有关公开路由的位置的信息。

22.2.1.8. .status.ingress

描述

Ingress 描述了可公开路由的位置。ingress 点列表可能包含重复的主机或 RouterName 值。路由在就绪后被视为 **Ready**

类型

array

22.2.1.9. .status.ingress[]

描述

RouteIngress 包含有关公开路由的位置的信息。

类型

object

属性	类型	描述
conditions	array	条件是路由的状态，可能为空。
conditions[]	对象	RouteIngressCondition 包含特定路由器上此路由的当前条件的详情。
主机	字符串	host 是公开路由的主机字符串；需要这个值
routerCanonicalHostname	字符串	CanonicalHostname 是路由器的外部主机名，可用作为此路由请求的主机的 CNAME。这个值是可选的，可能不会在所有情况下设置。

属性	类型	描述
routerName	字符串	名称是路由器选择以标识自身的名称；需要这个值
wildcardPolicy	字符串	通配符策略是允许公开此路由的通配符策略。

22.2.1.10. .status.ingress[].conditions

描述

条件是路由的状态，可能为空。

类型

array

22.2.1.11. .status.ingress[].conditions[]

描述

RoutelIngressCondition 包含特定路由器上此路由的当前条件的详情。

类型

object

必填

- **status**
- **type**

属性	类型	描述
lastTransitionTime	string	当条件最后转换时的 RFC 3339 日期和时间
message	字符串	人类可读的消息，指示最后一次转换的详细信息。
reason	字符串	(简洁) 条件最后一次转换的原因，通常是机器和人类可读的常量。
status	字符串	status 是条件的状态。可以是 True, False, Unknown。
type	字符串	type 是条件的类型。目前只有 Admitted。

22.2.2. API 端点

可用的 API 端点如下：

- **/apis/route.openshift.io/v1/routes**
 - **GET** : 列出类型为 Route 的对象
- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes**
 - **DELETE** : 删除路由的集合
 - **GET** : 列出类型为 Route 的对象
 - **POST** : 创建路由
- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}**
 - **DELETE** : 删除路由
 - **GET** : 读指定的路由
 - **PATCH** : 部分更新指定的路由
 - **PUT** : 替换指定的路由
- **/apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status**
 - **GET** : 指定 Route 的读取状态
 - **PATCH** : 部分更新指定路由的状态
 - **PUT** : 替换指定路由的状态

22.2.2.1. /apis/route.openshift.io/v1/routes

表 22.1. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出类型为 Route 的对象

表 22.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	RouteList 模式
401 - Unauthorized	空

22.2.2.2. /apis/route.openshift.io/v1/namespaces/{namespace}/routes

表 22.3. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 22.4. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除路由集合

表 22.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 22.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出类型为 Route 的对象

表 22.7. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 22.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	RouteList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建路由

表 22.9. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 22.10. 主体参数

参数	类型	描述
正文 (body)	Route 模式	

表 22.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
201 - Created	Route 模式
202 - Accepted	Route 模式
401 - Unauthorized	空

22.2.2.3. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}

表 22.12. 全局路径参数

参数	类型	描述
name	字符串	路由的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 22.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除路由

表 22.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 22.15. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 22.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的路由

表 22.17. 查询参数

参数	类型	描述
<code>resourceVersion</code>	字符串	<code>resourceVersion</code> 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 22.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Route</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的路由

表 22.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 22.20. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 22.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的路由

表 22.22. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 22.23. 主体参数

参数	类型	描述
正文 (body)	Route 模式	

表 22.24. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
201 - Created	Route 模式
401 - Unauthorized	空

22.2.2.4. /apis/route.openshift.io/v1/namespaces/{namespace}/routes/{name}/status

表 22.25. 全局路径参数

参数	类型	描述
name	字符串	路由的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 22.26. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定路由的读取状态

表 22.27. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 22.28. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定路由的部分更新状态

表 22.29. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 22.30. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 22.31. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定路由的状态

表 22.32. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 22.33. 主体参数

参数	类型	描述
正文 (body)	Route 模式	

表 22.34. HTTP 响应

HTTP 代码	响应正文
200 - OK	Route 模式
201 - Created	Route 模式
401 - Unauthorized	空

第 23 章 调度 API

23.1. 调度 API

23.1.1. PriorityClass [scheduling.k8s.io/v1]

描述

PriorityClass 定义从优先级类名称到优先级整数值映射。该值可以是任何有效的整数。

类型

对象

23.2. PRIORITYCLASS [SCHEDULING.K8S.IO/V1]

描述

PriorityClass 定义从优先级类名称到优先级整数值映射。该值可以是任何有效的整数。

类型

object

必填

- value

23.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
description	字符串	description 是一个任意字符串，通常会提供有关何时应使用此优先级类的指南。

属性	类型	描述
globalDefault	布尔值	globalDefault 指定此 PriorityClass 是否应该被视为没有优先级类的 pod 的默认优先级。只能将一个 PriorityClass 标记为全局默认。但是，如果存在多个 PriorityClasses ，其 globalDefault 字段设为 true ，则此类全局默认 PriorityClasses 的最小值将用作默认优先级。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
preemptionPolicy	string	preemptionPolicy 是抢占优先级较低的 pod 的策略。Never, PreemptLowerPriority 之一。如果未设置，则默认为 PreemptLowerPriority。 可能枚举值： - "Never" 表示 pod 不会抢占优先级较低的其他 pod。 - "PreemptLowerPriority" 表示 pod 可以抢占优先级较低的其他 pod。
value	整数	value 代表此优先级类的整数值。这是 pod 在 pod 规格中具有此类名称时接收的实际优先级。

23.2.2. API 端点

可用的 API 端点如下：

- **/apis/scheduling.k8s.io/v1/priorityclasses**
 - **DELETE** : 删除 PriorityClass 集合
 - **GET**: 列出或监视类型为 PriorityClass 的对象
 - **POST** : 创建一个 PriorityClass
- **/apis/scheduling.k8s.io/v1/watch/priorityclasses**
 - **GET**: 观察单个对 PriorityClass 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/scheduling.k8s.io/v1/priorityclasses/{name}**
 - **DELETE** : 删除 PriorityClass
 - **GET** : 读取指定的 PriorityClass
 - **PATCH** : 部分更新指定的 PriorityClass
 - **PUT** : 替换指定的 PriorityClass
- **/apis/scheduling.k8s.io/v1/watch/priorityclasses/{name}**
 - **GET**: 观察对类型为 PriorityClass.deprecated 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

23.2.2.1. /apis/scheduling.k8s.io/v1/priorityclasses

表 23.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PriorityClass 集合

表 23.2. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 23.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 23.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 PriorityClass 的对象

表 23.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 23.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityClassList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 PriorityClass

表 23.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 23.8. 主体参数

参数	类型	描述
正文 (body)	PriorityClass 模式	

表 23.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityClass 模式
201 - Created	PriorityClass 模式
202 - Accepted	PriorityClass 模式
401 - Unauthorized	空

23.2.2.2. /apis/scheduling.k8s.io/v1/watch/priorityclasses

表 23.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察个别对 PriorityClass.deprecated 的列表的更改：改为使用 'watch' 参数和 list 操作。

表 23.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

23.2.2.3. /apis/scheduling.k8s.io/v1/priorityclasses/{name}

表 23.12. 全局路径参数

参数	类型	描述
name	字符串	PriorityClass 的名称

表 23.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 PriorityClass

表 23.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 23.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 23.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 PriorityClass

表 23.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityClass 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 PriorityClass

表 23.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 23.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 23.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityClass 模式
201 - Created	PriorityClass 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 PriorityClass

表 23.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 23.22. 主体参数

参数	类型	描述
正文 (body)	PriorityClass 模式	

表 23.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	PriorityClass 模式
201 - Created	PriorityClass 模式
401 - Unauthorized	空

23.2.2.4. /apis/scheduling.k8s.io/v1/watch/priorityclasses/{name}

表 23.24. 全局路径参数

参数	类型	描述
name	字符串	PriorityClass 的名称

表 23.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind PriorityClass. deprecated 的对象更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 23.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

第 24 章 安全 API

24.1. 安全 API

24.1.1. SecurityContextConstraints [security.openshift.io/v1]

描述

SecurityContextConstraints 监管发出会影响到容器的 SecurityContext 的请求的能力。由于历史原因，SCC 在核心 Kubernetes API 组下公开。该风险已弃用，并将在以后的发行版本中删除 - 用户应使用 security.openshift.io 组来管理 SecurityContextConstraints。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

对象

24.2. SECURITYCONTEXTCONSTRAINTS [SECURITY.OPENSIFT.IO/V1]

描述

SecurityContextConstraints 监管发出会影响到容器的 SecurityContext 的请求的能力。由于历史原因，SCC 在核心 Kubernetes API 组下公开。该风险已弃用，并将在以后的发行版本中删除 - 用户应使用 security.openshift.io 组来管理 SecurityContextConstraints。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

object

必填

- allowHostDirVolumePlugin
- allowHostIPC
- allowHostNetwork
- allowHostPID
- allowHostPorts
- allowPrivilegedContainer
- readOnlyRootFilesystem

24.2.1. 规格

属性	类型	描述
allowHostDirVolumePlugin	布尔值	AllowHostDirVolumePlugin 决定策略允许容器是否使用 HostDir 卷插件

属性	类型	描述
allowHostIPC	布尔值	allowHostIPC 确定策略是否允许容器中的 ipc 主机。
allowHostNetwork	布尔值	allowHostNetwork 决定策略是否允许在 pod 规格中使用 HostNetwork。
allowHostPID	布尔值	allowHostPID 确定策略是否允许在容器中主机 pid。
allowHostPorts	布尔值	allowHostPorts 决定策略是否允许容器中的主机端口。
allowPrivilegeEscalation	``	allowPrivilegeEscalation 决定 pod 是否请求允许特权升级。如果未指定，则默认为 true。
allowPrivilegedContainer	布尔值	allowPrivilegedContainer 决定容器是否可以请求以特权方式运行。
allowedCapabilities	``	allowedCapabilities 是可请求添加到容器的功能列表。在 pod 作者可自由裁量添加此字段中的功能。您不能列出 AllowedCapabilities 和 RequiredDropCapabilities 中的功能。要允许所有功能，您可以使用 '*'。
allowedFlexVolumes	``	AllowedFlexVolumes 是允许的 Flexvolumes 的白名单。empty 或 nil 表示可以使用所有 Flexvolumes。只有在 "Volumes" 字段中允许使用 Flexvolumes 时，此参数才有效。
allowedUnsafeSysctls	``	allowedUnsafeSysctls 是明确允许的不安全 sysctl 列表，默认为 none。每个条目都是普通 sysctl 名称，或以 "" 结尾，在这种情况下，它被视为允许的 sysctl 的前缀。单个 * 意味着允许所有不安全 sysctl。kubelet 必须明确将所有允许的不安全 sysctl 列入白名单，以避免拒绝。示例：e.g. "foo/" 允许 "foo/bar", "foo/baz" 等。例如 "foo prerequisites" allow "foo.bar", "foo.baz" 等。

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
defaultAddCapabilities	``	defaultAddCapabilities 是默认的能力集合，将添加到容器中，除非 pod 规格专门丢弃了该功能。您不能列出 DefaultAddCapabilities 和 RequiredDropCapabilities 中的功能。
defaultAllowPrivilegeEscalation	``	defaultAllowPrivilegeEscalation 控制进程是否可以超过父进程的默认设置。
forbiddenSysctls	``	forbiddenSysctls 是一个显式禁止的 sysctl 列表，默认为 none。每个条目都是普通 sysctl 名称，或以 "" 结尾，在这种情况下，它被视为禁止的 sysctl 的前缀。单个 * 意味着禁止所有 sysctl。示例：e.g. "foo/" forbids "foo/bar", "foo/baz" 等。例如，如 "foo.bar" forbids "foo.bar", "foo.baz" 等。
fsGroup	``	fsGroup 是指定 SecurityContext 使用什么 fs 组的策略。
groups	``	有权使用此安全性上下文约束的组
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
priority	``	在评估哪个 SCC 根据 Users and Groups 字段中的访问尝试给定 Pod 请求时，优先级会影响 SCC 的排序顺序。int 越大，优先级越高。unset 值被视为 0 优先级。如果多个 SCC 的分数相等，它们将根据限制性最强到最低限制进行排序。如果优先级和限制性都相等，则 SCC 将根据名称进行排序。
readOnlyRootFilesystem	布尔值	设置为 true 时 readOnlyRootFileSystem 将强制容器使用只读根文件系统运行。如果容器特别要求使用非只读根文件系统运行，则 SCC 应该拒绝该 Pod。如果设置为 false，则容器可以使用只读根文件系统运行（如果希望），但不会强制使用它。
requiredDropCapabilities	``	requiredDropCapabilities 是将从容器中丢弃的功能。需要丢弃它们，且无法添加。
runAsUser	``	runAsUser 是策略，它将指定 SecurityContext 中使用什么 RunAsUser。
seLinuxContext	``	seLinuxContext 是策略，它将在 SecurityContext 中设置哪些标签。
seccompProfiles	``	seccompProfiles 列出了可为 pod 或容器的 seccomp 注解设置的允许的配置集。unset (nil) 或空值意味着 pod 或容器不能决定任何配置集。通配符 "*" 可用于允许所有配置集。用于在为 pod 生成值时，第一个非通配符配置集将用作默认值。
supplementalGroups	``	supplementalGroups 是策略，它将指定 SecurityContext 使用了哪些补充组。

属性	类型	描述
users	``	有权使用此安全性上下文约束的用户
卷	``	卷是允许的卷插件的白名单。fstype 直接与 VolumeSource 的字段名称对应(azureFile, configMap, emptyDir)。要允许所有卷都使用 "*"。要不允许卷, 设置为 ["none"]。

24.2.2. API 端点

可用的 API 端点如下：

- **/apis/security.openshift.io/v1/securitycontextconstraints**
 - **DELETE** : 删除 SecurityContextConstraints 的集合
 - **GET**: 列出类型为 SecurityContextConstraints 的对象
 - **POST** : 创建 SecurityContextConstraints
- **/apis/security.openshift.io/v1/securitycontextconstraints/{name}**
 - **DELETE** : 删除 SecurityContextConstraints
 - **GET** : 读取指定的 SecurityContextConstraints
 - **PATCH** : 部分更新指定的 SecurityContextConstraints
 - **PUT** : 替换指定的 SecurityContextConstraints

24.2.2.1. /apis/security.openshift.io/v1/securitycontextconstraints

表 24.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 SecurityContextConstraints 的集合

表 24.2. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 24.3. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出类型为 SecurityContextConstraints 的对象

表 24.4. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 24.5. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecurityContextConstraintsList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 SecurityContextConstraints

表 24.6. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 24.7. 主体参数

参数	类型	描述
正文 (body)	SecurityContextConstraints 模式	

表 24.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecurityContextConstraints 模式
201 - Created	SecurityContextConstraints 模式
202 - Accepted	SecurityContextConstraints 模式
401 - Unauthorized	空

24.2.2.2. /apis/security.openshift.io/v1/securitycontextconstraints/{name}

表 24.9. 全局路径参数

参数	类型	描述
name	字符串	SecurityContextConstraints 的名称

表 24.10. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 SecurityContextConstraints

表 24.11. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 24.12. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 24.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 SecurityContextConstraints

表 24.14. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 24.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecurityContextConstraints 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 SecurityContextConstraints

表 24.16. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 24.17. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 24.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecurityContextConstraints 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 SecurityContextConstraints

表 24.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 24.20. 主体参数

参数	类型	描述
正文 (body)	SecurityContextConstraints 模式	

表 24.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	SecurityContextConstraints 模式
201 - Created	SecurityContextConstraints 模式
401 - Unauthorized	空

第 25 章 SECURITY-INTERNAL API

25.1. 安全内部 API

25.1.1. RangeAllocation [security.internal.openshift.io/v1]

描述

使用 RangeAllocation，以便我们可以轻松公开类型为安全组的 RangeAllocation，它是一个内部 API，不适用于外部使用。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

object

25.2. RANGEALLOCATION [SECURITY.INTERNAL.OPENSIFT.IO/V1]

描述

使用 RangeAllocation，以便我们可以轻松公开类型为安全组的 RangeAllocation，它是一个内部 API，不适用于外部使用。兼容性级别 1：在主发行版本中至少提供 12 个月或 3 个次版本（以更长的时间为准）。

类型

对象

25.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
data	string	数据是一个字节阵列，代表范围分配的序列化状态。这是一个位图，每个位设置为代表一个范围。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
range	string	范围是一个字符串，代表一系列 uids 的唯一标签 "10000000000-20000000000/10000"。

25.2.2. API 端点

可用的 API 端点如下：

- **/apis/security.internal.openshift.io/v1/rangeallocations**
 - **DELETE** : 删除 RangeAllocation 的集合
 - **GET** : 列出类型为 RangeAllocation 的对象
 - **POST** : 创建一个 RangeAllocation
- **/apis/security.internal.openshift.io/v1/rangeallocations/{name}**
 - **DELETE** : 删除 RangeAllocation
 - **GET** : 读取指定的 RangeAllocation
 - **PATCH** : 部分更新指定的 RangeAllocation
 - **PUT** : 替换指定的 RangeAllocation

25.2.2.1. /apis/security.internal.openshift.io/v1/rangeallocations

表 25.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RangeAllocation 的集合

表 25.2. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 25.3. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出类型为 RangeAllocation 的对象

表 25.4. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 25.5. HTTP 响应

HTTP 代码	响应正文
200 - OK	RangeAllocationList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 RangeAllocation

表 25.6. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 25.7. 主体参数

参数	类型	描述
正文 (body)	RangeAllocation 模式	

表 25.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	RangeAllocation 模式
201 - Created	RangeAllocation 模式
202 - Accepted	RangeAllocation 模式
401 - Unauthorized	空

25.2.2.2. /apis/security.internal.openshift.io/v1/rangeallocations/{name}

表 25.9. 全局路径参数

参数	类型	描述
name	string	RangeAllocation 的名称

表 25.10. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 RangeAllocation

表 25.11. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 25.12. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 25.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 RangeAllocation

表 25.14. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 25.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	RangeAllocation 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 RangeAllocation

表 25.16. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 25.17. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 25.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	RangeAllocation 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 RangeAllocation

表 25.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 25.20. 主体参数

参数	类型	描述
正文 (body)	RangeAllocation 模式	

表 25.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	RangeAllocation 模式
201 - Created	RangeAllocation 模式
401 - Unauthorized	空

第 26 章 快照 API

26.1. 快照 API

26.1.1. VolumeSnapshot [snapshot.storage.k8s.io/v1]

描述

VolumeSnapshot 是用户创建持久性卷的时间点快照或绑定到预先存在的快照的请求。

类型

object

26.1.2. VolumeSnapshotClass [snapshot.storage.k8s.io/v1]

描述

VolumeSnapshotClass 指定在创建卷快照时底层存储系统使用的参数。通过在 VolumeSnapshot 对象中指定名称来使用特定的 VolumeSnapshotClass。VolumeSnapshotClasses 是非命名空间

类型

object

26.1.3. VolumeSnapshotContent [snapshot.storage.k8s.io/v1]

描述

VolumeSnapshotContent 代表底层存储系统中实际"on-disk"快照对象

类型

object

26.2. VOLUMESNAPSHOT [SNAPSHOT.STORAGE.K8S.IO/V1]

描述

VolumeSnapshot 是用户创建持久性卷的时间点快照或绑定到预先存在的快照的请求。

类型

object

必填

- **spec**

26.2.1. 规格

属性	类型	描述
----	----	----

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	spec 定义用户请求的快照所需的特征。更多信息： https://kubernetes.io/docs/concepts/storage/volume-snapshots#volumesnapshots 必需。
status	object	status 代表快照的当前信息。用户必须验证 VolumeSnapshot 和 VolumeSnapshotContent 对象之间的绑定是否成功（通过验证 VolumeSnapshot 和 VolumeSnapshotContent 点相互验证），然后才能使用这个对象。

26.2.1.1. .spec

描述

spec 定义用户请求的快照所需的特征。更多信息：

<https://kubernetes.io/docs/concepts/storage/volume-snapshots#volumesnapshots> 必需。

类型

object

必填

- **source**

属性	类型	描述
source	object	Source 指定将从其创建快照的位置。此字段在创建后不可变。必需。
volumeSnapshotClassName	string	volumeSnapshotClassName 是 VolumeSnapshot 请求的 VolumeSnapshotClass 的名称。volumeSnapshotClassName 可能保留为 nil，以指示应使用默认的 mvapich。一个给定集群可能有多个默认卷 mvapiches：每个 CSI Driver 一个默认卷。如果 VolumeSnapshot 没有指定 mvapich，则会检查 VolumeSnapshotSource 来找出关联的 CSI 驱动程序是什么，并使用与该 CSI 驱动程序关联的默认 VolumeSnapshotClass。如果给定 CSI Driver 并有多个 VolumeSnapshotClass 被标记为默认值，则 CreateSnapshot 将失败并生成事件。此字段不允许空字符串。

26.2.1.2. .spec.source

描述

Source 指定将从其创建快照的位置。此字段在创建后不可变。必需。

类型

object

属性	类型	描述
persistentVolumeClaimName	string	PersistentVolumeClaimName 指定代表应该创建快照的卷的 PersistentVolumeClaim 对象的名称。此 PVC 被认为与 VolumeSnapshot 对象位于同一个命名空间中。如果快照不存在并且需要创建，则应设置此字段。此字段是不可变的。

属性	类型	描述
volumeSnapshotContentName	string	volumeSnapshotContentName 指定代表现有卷快照的预先存在的 VolumeSnapshotContent 对象的名称。如果快照已存在且只需要 Kubernetes 中的表示，则应设置此字段。此字段是不可变的。

26.2.1.3. .status

描述

status 代表快照的当前信息。用户必须验证 VolumeSnapshot 和 VolumeSnapshotContent 对象之间的绑定是否成功（通过验证 VolumeSnapshot 和 VolumeSnapshotContent 点相互验证），然后才能使用这个对象。

类型

object

属性	类型	描述
boundVolumeSnapshotContentName	string	boundVolumeSnapshotContentName 是这个 VolumeSnapshot 对象要绑定到的 VolumeSnapshotContent 对象的名称。如果没有指定，这表示 VolumeSnapshot 对象还没有成功绑定到 VolumeSnapshotContent 对象。注意：为了避免可能的安全问题，用户必须在使用这个对象前验证 VolumeSnapshot 和 VolumeSnapshotContent 对象之间的绑定是否成功（通过验证 VolumeSnapshot 和 VolumeSnapshotContent 点）。
creationTime	string	creationTime 是底层存储系统获取点快照时的时间戳。在动态快照创建情形中，快照控制器将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "creation_time" 值。对于预先存在的快照，如果驱动程序支持，则此字段将填充从 CSI "ListSnapshots" gRPC 调用返回的 "creation_time" 值。如果未指定，这可能表示快照的创建时间未知。

属性	类型	描述
错误	object	error 是快照创建过程中最后观察到的错误（若有）。此字段对上级控制器（如应用程序控制器）很有用，以确定它们是否应该根据报告的错误类型继续等待快照创建。快照控制器会在创建快照期间发生错误时保留重试。成功后，这个错误字段将被清除。
readyToUse	布尔值	readyToUse 表示快照是否已准备好用于恢复卷。在动态快照创建情形中，快照控制器将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "ready_to_use" 值。对于预先存在的快照，此字段将填充从 CSI "ListSnapshots" gRPC 调用返回的 "ready_to_use" 值（如果驱动程序支持），否则此字段将被设置为 "True"。如果没有指定，这表示快照的就绪情况未知。
restoreSize	integer-or-string	restoreSize 代表从这个快照创建卷所需的最小卷大小。在动态快照创建情形中，快照控制器将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "size_bytes" 值。对于预先存在的快照，如果驱动程序支持，则此字段将被填充从 CSI "ListSnapshots" gRPC 调用返回的 "size_bytes" 值。从这个快照恢复卷时，如果指定卷的大小必须小于 restoreSize，否则恢复将失败。如果没有指定，这表示大小未知。

26.2.1.4. .status.error

描述

error 是快照创建过程中最后观察到的错误（若有）。此字段对上级控制器（如应用程序控制器）很有用，以确定它们是否应该根据报告的错误类型继续等待快照创建。快照控制器会在创建快照期间发生错误时保留重试。成功后，这个错误字段将被清除。

类型

object

属性	类型	描述
----	----	----

属性	类型	描述
message	string	Message 是一个字符串，详细描述了在快照创建过程中遇到的错误（如果指定）。注意：可以记录消息，并且不应包含敏感信息。
time	string	时间是遇到错误时的时间戳。

26.2.2. API 端点

可用的 API 端点如下：

- **/apis/snapshot.storage.k8s.io/v1/volumesnapshots**
 - **GET** : 列出 VolumeSnapshot 类型的对象
- **/apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots**
 - **DELETE** : 删除 VolumeSnapshot 的集合
 - **GET** : 列出 VolumeSnapshot 类型的对象
 - **POST** : 创建 VolumeSnapshot
- **/apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots/{name}**
 - **DELETE** : 删除 VolumeSnapshot
 - **GET** : 读取指定的 VolumeSnapshot
 - **PATCH** : 部分更新指定的 VolumeSnapshot
 - **PUT** : 替换指定的 VolumeSnapshot
- **/apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots/{name}/status**
 - **GET** : 指定 VolumeSnapshot 的读取状态
 - **PATCH** : 部分更新指定 VolumeSnapshot 的状态
 - **PUT** : 替换指定 VolumeSnapshot 的状态

26.2.2.1. /apis/snapshot.storage.k8s.io/v1/volumesnapshots

表 26.1. 全局查询参数

参数	类型	描述
----	----	----

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出 VolumeSnapshot 类型的对象

表 26.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotList 模式
401 - Unauthorized	空

26.2.2.2. /apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots

表 26.3. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 26.4. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshot 的集合

表 26.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。

参数	类型	描述
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出 VolumeSnapshot 类型的对象

表 26.7. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 VolumeSnapshot

表 26.9. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.10. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshot 模式	

表 26.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
201 - Created	VolumeSnapshot 模式
202 - Accepted	VolumeSnapshot 模式
401 - Unauthorized	空

26.2.2.3. /apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots/{name}

表 26.12. 全局路径参数

参数	类型	描述
name	string	VolumeSnapshot 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 26.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshot

表 26.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。

参数	类型	描述
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 26.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 26.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 VolumeSnapshot

表 26.17. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 26.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 VolumeSnapshot

表 26.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 26.20. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 26.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 VolumeSnapshot

表 26.22. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.23. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshot 模式	

表 26.24. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
201 - Created	VolumeSnapshot 模式
401 - Unauthorized	空

26.2.2.4. /apis/snapshot.storage.k8s.io/v1/namespaces/{namespace}/volumesnapshots/{name}

表 26.25. 全局路径参数

参数	类型	描述
name	string	VolumeSnapshot 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 26.26. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 VolumeSnapshot 的读取状态

表 26.27. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 26.28. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

指定 VolumeSnapshot 的部分更新状态

表 26.29. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

参数	类型	描述
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 26.30. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 26.31. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 VolumeSnapshot 的状态

表 26.32. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.33. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshot 模式	

表 26.34. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshot 模式
201 - Created	VolumeSnapshot 模式
401 - Unauthorized	空

26.3. VOLUMESNAPSHOTCLASS [SNAPSHOT.STORAGE.K8S.IO/V1]

描述

VolumeSnapshotClass 指定在创建卷快照时底层存储系统使用的参数。通过在 VolumeSnapshot 对象中指定名称来使用特定的 VolumeSnapshotClass。VolumeSnapshotClasses 是非命名空间

类型

object

必填

- **deletionPolicy**
- **driver**

26.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
deletionPolicy	string	deletionPolicy 决定在删除绑定 VolumeSnapshotClass 时是否应该删除通过 VolumeSnapshotClass 创建的 VolumeSnapshotContent。支持的值有 "Retain" 和 "Delete"。"包含"意味着保留底层存储系统上的 VolumeSnapshotContent 及其物理快照。"delete" 表示删除底层存储系统上的 VolumeSnapshotContent 及其物理快照。必需。
driver	string	driver 是处理此 VolumeSnapshotClass 的存储驱动程序程序的名称。必需。
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
parameters	对象 (字符串)	参数是包含存储驱动程序特定参数的键值映射，用于创建快照。这些值对 Kubernetes 不透明。

26.3.2. API 端点

可用的 API 端点如下：

- **/apis/snapshot.storage.k8s.io/v1/volumesnapshotclasses**
 - **DELETE** : 删除 VolumeSnapshotClass 的集合
 - **GET**: 列出 VolumeSnapshotClass 类型的对象
 - **POST** : 创建 VolumeSnapshotClass
- **/apis/snapshot.storage.k8s.io/v1/volumesnapshotclasses/{name}**
 - **DELETE** : 删除 VolumeSnapshotClass
 - **GET** : 读取指定的 VolumeSnapshotClass
 - **PATCH** : 部分更新指定的 VolumeSnapshotClass
 - **PUT** : 替换指定的 VolumeSnapshotClass

26.3.2.1. /apis/snapshot.storage.k8s.io/v1/volumesnapshotclasses

表 26.35. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshotClass 的集合

表 26.36. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.37. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET**描述**

列出 VolumeSnapshotClass 类型的对象

表 26.38. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.39. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotClassList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 VolumeSnapshotClass

表 26.40. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.41. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshotClass 模式	

表 26.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotClass 模式
201 - Created	VolumeSnapshotClass 模式
202 - Accepted	VolumeSnapshotClass 模式
401 - Unauthorized	空

26.3.2.2. /apis/snapshot.storage.k8s.io/v1/volumesnapshotclasses/{name}

表 26.43. 全局路径参数

参数	类型	描述
name	string	VolumeSnapshotClass 的名称

表 26.44. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshotClass

表 26.45. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 26.46. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 26.47. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 VolumeSnapshotClass

表 26.48. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 26.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotClass 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 VolumeSnapshotClass

表 26.50. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 26.51. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 26.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotClass 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 VolumeSnapshotClass

表 26.53. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.54. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshotClass 模式	

表 26.55. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotClass 模式
201 - Created	VolumeSnapshotClass 模式
401 - Unauthorized	空

26.4. VOLUMESNAPSHOTCONTENT [SNAPSHOT.STORAGE.K8S.IO/V1]

描述

VolumeSnapshotContent 代表底层存储系统中实际"on-disk"快照对象

类型

object

必填

- **spec**

26.4.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	spec 定义底层存储系统创建的 VolumeSnapshotContent 的属性。必需。
status	object	status 代表快照的当前信息。

26.4.1.1. .spec

描述

spec 定义底层存储系统创建的 VolumeSnapshotContent 的属性。必需。

类型

object

必填

- **deletionPolicy**

- driver
- source
- volumeSnapshotRef

属性	类型	描述
deletionPolicy	string	deletionPolicy 决定在删除绑定 VolumeSnapshot 时是否应该删除底层存储系统上的这个 VolumeSnapshotContent 及其物理快照。支持的值有 "Retain" 和 "Delete"。"包含"意味着保留底层存储系统上的 VolumeSnapshotContent 及其物理快照。"delete" 表示删除底层存储系统上的 VolumeSnapshotContent 及其物理快照。对于动态置备的快照，CSI 快照 sidecar 将自动填充相应 VolumeSnapshotClass 中定义的 "DeletionPolicy" 字段。对于预先存在的快照，用户在创建 VolumeSnapshotContent 对象时必须指定此字段。必需。
driver	string	driver 是用于在底层存储系统上创建物理快照的 CSI 驱动程序名称。这必须与该驱动程序的 CSI GetPluginName () 调用返回的名称相同。必需。
source	object	source 指定快照是（或应该）动态置备或已存在，只需要 Kubernetes 对象表示。此字段在创建后不可变。必需。
sourceVolumeMode	string	SourceVolumeMode 是生成快照的卷模式。可以是 "Filesystem" 或 "Block"。如果没有指定，这表示源卷的模式未知。此字段是不可变的。此字段是一个 alpha 字段。
volumeSnapshotClassName	string	创建此快照的 VolumeSnapshotClass 的名称。请注意，在置备后，可以使用不同值集删除或重新创建 VolumeSnapshotClass，因此不应引用后续快照。

属性	类型	描述
volumeSnapshotRef	object	volumeSnapshotRef 指定这个 VolumeSnapshotContent 对象绑定到的 VolumeSnapshot 对象。VolumeSnapshot.Spec.VolumeSnapshotContentName 字段必须引用此 VolumeSnapshotContent 的名称，以便双向绑定有效。对于已存在的 VolumeSnapshotContent 对象，需要提供 VolumeSnapshot 对象的名称和命名空间，以便进行绑定。此字段在创建后不可变。必需。

26.4.1.2. .spec.source

描述

source 指定快照是（或应该）动态置备或已存在，只需要 Kubernetes 对象表示。此字段在创建后不可变。必需。

类型

object

属性	类型	描述
snapshotHandle	string	SnapshotHandle 指定在底层存储系统上预先存在的快照的 CSI "snapshot_id", Kubernetes 对象表示（或应该）创建 Kubernetes 对象表示。此字段是不可变的。
volumeHandle	string	volumeHandle 指定应该从中动态获取快照的卷的 CSI "volume_id"。此字段是不可变的。

26.4.1.3. .spec.volumeSnapshotRef

描述

volumeSnapshotRef 指定这个 VolumeSnapshotContent 对象绑定到的 VolumeSnapshot 对象。VolumeSnapshot.Spec.VolumeSnapshotContentName 字段必须引用此 VolumeSnapshotContent 的名称，以便双向绑定有效。对于已存在的 VolumeSnapshotContent 对象，需要提供 VolumeSnapshot 对象的名称和命名空间，以便进行绑定。此字段在创建后不可变。必需。

类型

object

属性	类型	描述
apiVersion	字符串	引用的 API 版本。
fieldPath	字符串	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。Todo: 这个设计不是最终设计的，且该字段在将来会有所改变。
kind	字符串	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	字符串	引用的名称。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
namespace	字符串	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
resourceVersion	字符串	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency
uid	字符串	引用的 UID。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids

26.4.1.4. .status

描述

status 代表快照的当前信息。

类型

object

属性	类型	描述
creationTime	整数	creationTime 是底层存储系统获取点快照时的时间戳。在动态快照创建的情况下，CSI 快照 sidecar 将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "creation_time" 值。对于预先存在的快照，如果驱动程序支持，则此字段将填充从 CSI "ListSnapshots" gRPC 调用返回的 "creation_time" 值。如果没有指定，这表示创建时间未知。此字段的格式是以 int64 编码的 Unix 纳秒时间。在 Unix 上，命令 date +%s%N 返回自 1970-01-01 00:00:00 UTC 起的纳秒的当前时间。
错误	object	error 是快照创建过程中最后观察到的错误（若有）。重试成功后，这个错误字段将被清除。
readyToUse	布尔值	readyToUse 表示快照是否已准备好用于恢复卷。在动态快照创建的情况下，CSI snapshotter sidecar 将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "ready_to_use" 值。对于预先存在的快照，此字段将填充从 CSI "ListSnapshots" gRPC 调用返回的 "ready_to_use" 值（如果驱动程序支持），否则此字段将被设置为 "True"。如果没有指定，这表示快照的就绪情况未知。

属性	类型	描述
restoreSize	整数	restoreSize 代表快照的完整大小（以字节为单位）。在动态快照创建的情况下，CSI snapshotter sidecar 将填充从 CSI "CreateSnapshot" gRPC 调用返回的 "size_bytes" 值。对于预先存在的快照，如果驱动程序支持，则此字段将被填充从 CSI "ListSnapshots" gRPC 调用返回的 "size_bytes" 值。从这个快照恢复卷时，如果指定卷的大小必须小于 restoreSize，否则恢复将失败。如果没有指定，这表示大小未知。
snapshotHandle	string	SnapshotHandle 是底层存储系统上快照的 CSI "snapshot_id"。如果没有指定，这表示动态快照创建失败，或者仍在进行中。

26.4.1.5. .status.error

描述

error 是快照创建过程中最后观察到的错误（若有）。重试成功后，这个错误字段将被清除。

类型

object

属性	类型	描述
message	string	Message 是一个字符串，详细描述了在快照创建过程中遇到的错误（如果指定）。注意：可以记录消息，并且不应包含敏感信息。
time	string	时间是遇到错误时的时间戳。

26.4.2. API 端点

可用的 API 端点如下：

- **/apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents**
 - **DELETE** : 删除 VolumeSnapshotContent 的集合
 - **GET**: 列出 VolumeSnapshotContent 类型的对象
 - **POST**: 创建 VolumeSnapshotContent
- **/apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents/{name}**
 - **DELETE** : 删除 VolumeSnapshotContent

- **DELETE** : 删除 VolumeSnapshotContent
- **GET**: 读取指定的 VolumeSnapshotContent
- **PATCH** : 部分更新指定的 VolumeSnapshotContent
- **PUT** : 替换指定的 VolumeSnapshotContent
- **/apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents/{name}/status**
 - **GET**: 指定 VolumeSnapshotContent 的读取状态
 - **PATCH**: 部分更新指定 VolumeSnapshotContent 的状态
 - **PUT** : 替换指定 VolumeSnapshotContent 的状态

26.4.2.1. /apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents

表 26.56. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshotContent 的集合

表 26.57. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.58. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出 VolumeSnapshotContent 类型的对象

表 26.59. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 26.60. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContentList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 VolumeSnapshotContent

表 26.61. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.62. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshotContent schema	

表 26.63. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
201 - Created	VolumeSnapshotContent schema
202 - Accepted	VolumeSnapshotContent schema
401 - Unauthorized	空

26.4.2.2. /apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents/{name}

表 26.64. 全局路径参数

参数	类型	描述
name	string	VolumeSnapshotContent 的名称

表 26.65. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeSnapshotContent

表 26.66. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 26.67. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 26.68. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 VolumeSnapshotContent

表 26.69. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 26.70. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 VolumeSnapshotContent

表 26.71. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 26.72. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 26.73. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 VolumeSnapshotContent

表 26.74. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.75. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshotContent schema	

表 26.76. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
201 - Created	VolumeSnapshotContent schema
401 - Unauthorized	空

26.4.2.3. /apis/snapshot.storage.k8s.io/v1/volumesnapshotcontents/{name}/status

表 26.77. 全局路径参数

参数	类型	描述
----	----	----

参数	类型	描述
name	string	VolumeSnapshotContent 的名称

表 26.78. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

指定 VolumeSnapshotContent 的读取状态

表 26.79. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 26.80. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

指定 VolumeSnapshotContent 的部分更新状态

表 26.81. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 26.82. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 26.83. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 VolumeSnapshotContent 的状态

表 26.84. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 26.85. 主体参数

参数	类型	描述
正文 (body)	VolumeSnapshotContent schema	

表 26.86. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeSnapshotContent schema
201 - Created	VolumeSnapshotContent schema
401 - Unauthorized	空

第 27 章 存储 API

27.1. 存储 API

27.1.1. CSIDriver [storage.k8s.io/v1]

描述

CSIDriver 捕获集群中部署的 Container Storage Interface (CSI) 卷驱动程序的信息。Kubernetes 附加分离控制器使用此对象来确定是否需要附加。kubelet 使用此对象来确定是否需要在挂载时传递 pod 信息。CSIDriver 对象不是命名空间。

类型

object

27.1.2. CSINode [storage.k8s.io/v1]

描述

CSINode 包含有关节点上安装的所有 CSI 驱动程序的信息。CSI 驱动程序不需要直接创建 CSINode 对象。只要它们使用 node-driver-registrar sidecar 容器，kubelet 将自动为 CSI 驱动程序填充 CSI 驱动程序的 CSINode 对象作为 kubelet 插件注册的一部分。CSINode 的名称与节点相同。如果缺少对象，这意味着节点上没有 CSI 驱动程序，或者 Kubelet 版本足够低，它不创建此对象。CSINode 有一个 OwnerReference，指向对应的节点对象。

类型

object

27.1.3. CSIStorageCapacity [storage.k8s.io/v1]

描述

CSIStorageCapacity 存储一个 CSI GetCapacity 调用的结果。对于给定的 StorageClass，这描述了特定拓扑片段中的可用容量。在考虑实例化新的 PersistentVolume 的位置时，可以使用它。

例如，这可以表达如下内容： - StorageClass "standard" has "1234 GiB" in "topology.kubernetes.io/zone=us-east1" - StorageClass "localssd" has "10 GiB" in "kubernetes.io/hostname=knode-abc123"

以下三种情况都意味着，没有可用于特定组合的容量： - 没有带有合适的拓扑和存储类名称的对象存在 - 此类对象存在，但容量未设置 - 这种对象存在，但容量为零

这些对象的制作人可以决定哪些方法更合适。

当 CSI 驱动程序选择使用 CSIDriverSpec.StorageCapacity 时，kube-scheduler 会消耗它们。调度程序将 MaximumVolumeSize 与请求的待处理卷大小进行比较，以过滤出 unsuitable 节点。如果未设置 MaximumVolumeSize，它将回退到与不太精确容量进行比较。如果这也未设置，调度程序会假定容量不足，并尝试其他节点。

类型

object

27.1.4. StorageClass [storage.k8s.io/v1]

描述

StorageClass 描述了可以动态置备 PersistentVolume 的存储类的参数。

StorageClasses 是非命名空间；根据 etcd 的存储类的名称位于 ObjectMeta.Name 中。

类型

object

27.1.5. VolumeAttachment [storage.k8s.io/v1]

描述

VolumeAttachment 捕获将指定卷附加到指定节点的意图。

VolumeAttachment 对象不是命名空间。

类型

object

27.2. CSIDRIVER [STORAGE.K8S.IO/V1]

描述

CSIDriver 捕获集群中部署的 Container Storage Interface (CSI) 卷驱动程序的信息。Kubernetes 附加分离控制器使用此对象来确定是否需要附加。kubelet 使用此对象来确定是否需要挂载时传递 pod 信息。CSIDriver 对象不是命名空间。

类型

object

必填

- **spec**

27.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ObjectMeta	<p>标准对象 metadata.</p> <p>metadata.Name 表示此对象所指定的 CSI 驱动程序名称，它必须是该驱动程序的 CSI GetPluginName () 调用返回的相同名称。驱动程序名称必须是 63 个字符或更少，以字母数字字符([a-z0-9A-Z])开头和结尾，带有短划线(-)、句点(.)和字母数字字符。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</p>
spec	object	CSIDriverSpec 是 CSIDriver 的规格。

27.2.1.1. .spec

描述

CSIDriverSpec 是 CSIDriver 的规格。

类型

object

属性	类型	描述
attachRequired	布尔值	<p>attachRequired 表示此 CSI 卷驱动程序需要一个 attach 操作（因为它实现了 CSI ControllerPublishVolume () 方法），Kubernetes attach detach 控制器应调用附加卷接口来检查 volumeattachment 状态并等待卷被附加后再继续挂载。CSI external-attacher 与 CSI 卷驱动程序协调，并在 attach 操作完成后更新 volumeattachment 状态。如果启用了 CSIDriverRegistry 功能门，且值指定为 false，则会跳过附加操作。否则，将调用 attach 操作。</p> <p>此字段是不可变的。</p>

属性	类型	描述
fsGroupPolicy	string	<p>fsGroupPolicy 定义底层卷是否支持在挂载前更改卷的所有权和权限。如需了解更多详细信息，请参阅特定的 FSGroupPolicy 值。</p> <p>此字段是不可变的。</p> <p>默认为 <code>ReadWriteOnceWithFSType</code>，这将检查每个卷以确定 Kubernetes 是否应该修改卷的所有权和权限。使用默认策略时，只有在定义了 <code>fstype</code> 且卷的访问模式包含 <code>ReadWriteOnce</code> 时，才会应用定义的 <code>fsGroup</code>。</p>

属性	类型	描述
podInfoOnMount	布尔值	<p>podInfoOnMount 表示此 CSI 卷驱动程序在挂载操作过程中需要额外的 pod 信息（如 podName、podUID 等）。如果设置为 false，则不会在挂载时传递 pod 信息。默认值为 false。</p> <p>CSI 驱动程序指定 podInfoOnMount 作为驱动程序部署的一部分。如果为 true，Kubelet 会在 CSI NodePublishVolume () 调用中将 pod 信息作为 VolumeContext 传递。CSI 驱动程序负责解析和验证作为 VolumeContext 传递的信息。</p> <p>如果将 podInfoOnMount 设置为 true，则将传递以下 VolumeContext。此列表可能会增加，但将使用前缀。</p> <pre>"csi.storage.k8s.io/pod.name": pod.Name "csi.storage.k8s.io/pod.namespace": pod.Namespace "csi.storage.k8s.io/pod.uid": string (pod.UID) "csi.storage.k8s.io/ephemeral": "true"，如果卷是临时卷定义的临时卷，则为 "falseSource"</pre> <p>"CSI.storage.k8s.io/ephemeral" 是 Kubernetes 1.16 中的新功能。只有支持 "Persistent" 和 "Ephemeral" VolumeLifecycleMode 的驱动程序才需要它。其他驱动程序可能会禁用 pod 信息，/或忽略此字段。因为 Kubernetes 1.15 不支持此字段，因此驱动程序只能在这样的集群中部署时支持一个模式，因此部署决定了哪个模式，例如通过驱动程序的命令行参数。</p> <p>此字段是不可变的。</p>

属性	类型	描述
requiresRepublish	布尔值	<p>requiresRepublish 表示 CSI 驱动程序需要定期调用 NodePublishVolume 来反映挂载卷中的任何可能的更改。此字段默认为 false。</p> <p>注：成功调用初始 NodePublishVolume 后，对 NodePublishVolume 的后续调用应该只更新卷的内容。正在运行的容器不会看到新的挂载点。</p>
seLinuxMount	布尔值	<p>seLinuxMount 指定 CSI 驱动程序是否支持 "-o context" 挂载选项。</p> <p>当 "true" 时，CSI 驱动程序必须确保此 CSI 驱动程序提供的所有卷都可以使用不同的 -o context 选项单独挂载。这通常用于作为文件系统在块设备或独立共享卷上提供卷的存储后端。当挂载一个在设置了 SELinux 上下文的 Pod 中使用的 ReadWriteOncePod 卷时，Kubernetes 会使用 "-o context=xyz" 挂载选项调用 NodeStage / NodePublish。未来，它可以扩展到其他卷 AccessModes。无论哪种情况，Kubernetes 都会确保该卷只挂载一个 SELinux 上下文。</p> <p>当 "false" 时，Kubernetes 不会将任何特殊的 SELinux 挂载选项传递给驱动程序。这通常用于代表更大共享文件系统的子目录的卷。</p> <p>默认为 "false"。</p>

属性	类型	描述
storageCapacity	布尔值	<p>StorageCapacity 表示 CSI 卷驱动程序希望 pod 调度通过创建带有容量信息的 CSIStorageCapacity 对象来考虑驱动程序部署的存储容量（如果设为 true）。</p> <p>在部署驱动程序时，可以立即启用检查。在这种情况下，使用更新的绑定置备新卷将暂停，直到驱动程序部署发布一些合适的 CSIStorageCapacity 对象。</p> <p>或者，可以使用未设置或 false 字段部署驱动程序，稍后当存储容量信息被发布时，可以撤销它。</p> <p>此字段在 Kubernetes 1.22 中是不可变的，现在是可变的。</p>
tokenRequests	数组	<p>tokenRequests 表示 CSI 驱动程序需要 pod 的服务帐户令牌来挂载卷来进行必要的身份验证。kubelet 在 CSI NodePublishVolume 调用中传递 VolumeContext 中的令牌。CSI 驱动程序应该解析并验证以下 VolumeContext:</p> <pre>"csi.storage.k8s.io/serviceAccount.tokens": { "<audience>": { "token": <token>, "expirationTimestamp": <expiration timestamp in RFC3339>, }, ... }</pre> <p>注：每个 TokenRequest 中的 Audience 应该不同，最多一个令牌是空字符串。要在到期后接收新令牌，可以使用 RequiresRepublish 定期触发 NodePublishVolume。</p>
tokenRequests[]	object	TokenRequest 包含服务帐户令牌的参数。

属性	类型	描述
volumeLifecycleModes	数组（字符串）	<p>volumeLifecycleModes 定义此 CSI 卷驱动程序支持的卷类型。如果列表为空，则默认为 "Persistent"，这是 CSI 规格定义的使用量，并通过常见的 PV/PVC 机制在 Kubernetes 中实施。</p> <p>其他模式为 "Ephemeral"。在这个模式中，卷会在带有 CSIVolumeSource 的 pod 规格中定义，其生命周期与该 pod 的生命周期相关联。必须了解这个驱动程序，因为它只会为此类卷获取 NodePublishVolume 调用。</p> <p>有关实现这个模式的更多信息，请参阅 https://kubernetes-csi.github.io/docs/ephemeral-local-volumes.html。A 驱动程序可以支持一个或多个这些模式，并将来可能会添加更多模式。</p> <p>此字段是 beta。此字段是不可变的。</p>

27.2.1.2. .spec.tokenRequests

描述

tokenRequests 表示 CSI 驱动程序需要 pod 的服务帐户令牌来挂载卷来进行必要的身份验证。kubelet 在 CSI NodePublishVolume 调用中传递 VolumeContext 中的令牌。CSI 驱动程序应该解析并验证以下 VolumeContext: "csi.storage.k8s.io/serviceAccount.tokens": { "<audience>": { "token": <token>, "expirationTimestamp": <expiration timestamp in RFC3339>, }, ... }

注：每个 TokenRequest 中的 Audience 应该不同，最多一个令牌是空字符串。要在到期后接收新令牌，可以使用 RequiresRepublish 定期触发 NodePublishVolume。

类型

数组

27.2.1.3. .spec.tokenRequests[]

描述

TokenRequest 包含服务帐户令牌的参数。

类型

object

必填

- 受众

属性	类型	描述
受众	string	audience 是令牌在"TokenRequestSpec"中的预期受众。它将默认为 kube apiserver 的受众。
expirationSeconds	整数	expirationSeconds 是"TokenRequestSpec" 中令牌的有效性持续时间。它在"TokenRequestSpec" 中具有相同的默认值 "ExpirationSeconds"。

27.2.2. API 端点

可用的 API 端点如下：

- **/apis/storage.k8s.io/v1/csidrivers**
 - **DELETE** : 删除 CSIDriver 的集合
 - **GET**: 列出或监视类型 CSIDriver 的对象
 - **POST** : 创建一个 CSIDriver
- **/apis/storage.k8s.io/v1/watch/csidrivers**
 - **GET**: 观察单个对 CSIDriver 列表的更改。已弃用：使用 'watch' 参数和一个 list 操作。
- **/apis/storage.k8s.io/v1/csidrivers/{name}**
 - **DELETE** : 删除 CSIDriver
 - **GET**: 读取指定的 CSIDriver
 - **PATCH** : 部分更新指定的 CSIDriver
 - **PUT** : 替换指定的 CSIDriver
- **/apis/storage.k8s.io/v1/watch/csidrivers/{name}**
 - **GET**: 观察对类型 CSIDriver 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

27.2.2.1. /apis/storage.k8s.io/v1/csidrivers

表 27.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE**描述**

删除 CSIDriver 集合

表 27.2. 查询参数

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 27.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型 CSIDriver 的对象

表 27.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 27.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriverList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 CSIDriver

表 27.7. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.8. 主体参数

参数	类型	描述
正文 (body)	CSIDriver 模式	

表 27.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriver 模式
201 - Created	CSIDriver 模式
202 - Accepted	CSIDriver 模式
401 - Unauthorized	空

27.2.2.2. /apis/storage.k8s.io/v1/watch/csidrivers

表 27.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CSIDriver 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 27.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.2.2.3. /apis/storage.k8s.io/v1/csidrivers/{name}

表 27.12. 全局路径参数

参数	类型	描述
name	string	CSIDriver 的名称

表 27.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CSIDriver

表 27.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。

参数	类型	描述
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 27.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriver 模式
202 - Accepted	CSIDriver 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 CSIDriver

表 27.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriver 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 CSIDriver

表 27.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriver 模式
201 - Created	CSIDriver 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 CSIDriver

表 27.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.22. 主体参数

参数	类型	描述
正文 (body)	CSIDriver 模式	

表 27.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIDriver 模式
201 - Created	CSIDriver 模式
401 - Unauthorized	空

27.2.2.4. /apis/storage.k8s.io/v1/watch/csidrivers/{name}

表 27.24. 全局路径参数

参数	类型	描述
name	string	CSIDriver 的名称

表 27.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型 CSIDriver 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 27.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.3. CSINODE [STORAGE.K8S.IO/V1]

描述

CSINode 包含有关节点上安装的所有 CSI 驱动程序的信息。CSI 驱动程序不需要直接创建 CSINode

对象。只要它们使用 node-driver-registrar sidecar 容器，kubelet 将自动为 CSI 驱动程序填充 CSI 驱动程序的 CSINode 对象作为 kubelet 插件注册的一部分。CSINode 的名称与节点相同。如果缺少对象，这意味着节点上没有 CSI 驱动程序，或者 Kubelet 版本足够低，它不创建此对象。CSINode 有一个 OwnerReference，指向对应的节点对象。

类型

object

必填

- **spec**

27.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象的元数据。 metadata.name 必须是 Kubernetes 节点名称。
spec	object	CSINodeSpec 包含有关节点上安装的所有 CSI 驱动程序规格的信息

27.3.1.1. .spec

描述

CSINodeSpec 包含有关节点上安装的所有 CSI 驱动程序规格的信息

类型

object

必填

API 规范

- 驱动程序

属性	类型	描述
驱动程序	数组	驱动程序是节点上现有所有 CSI 驱动程序的信息列表。如果卸载列表中的所有驱动程序，这可能会为空。
<code>drivers[]</code>	<code>object</code>	<code>CSINodeDriver</code> 包含有关在节点上安装的一个 CSI 驱动程序规格的信息

27.3.1.2. `.spec.drivers`

描述

驱动程序是节点上现有所有 CSI 驱动程序的信息列表。如果卸载列表中的所有驱动程序，这可能会为空。

类型

数组

27.3.1.3. `.spec.drivers[]`

描述

`CSINodeDriver` 包含有关在节点上安装的一个 CSI 驱动程序规格的信息

类型

`object`

必填

- `name`
- `nodeID`

属性	类型	描述
<code>allocatable</code>	<code>object</code>	<code>VolumeNodeResources</code> 是一组用于调度卷的资源限值。
<code>name</code>	<code>string</code>	<code>name</code> 代表此对象引用的 CSI 驱动程序的名称。这个 <code>MUST</code> 与该驱动程序的 <code>CSI GetPluginName ()</code> 调用返回的名称相同。

属性	类型	描述
nodeID	string	从驱动程序的角度来看，节点的 nodeid。此字段可让 Kubernetes 与不为节点共享相同的 nomenclature 的存储系统通信。例如，Kubernetes 可能会将一个给定节点指代为"node1"，但存储系统可能会引用与"nodeA"相同的节点。当 Kubernetes 向存储系统发送一个命令来将卷附加到特定节点时，可以使用此字段来引用使用存储系统将理解的 ID 的节点名称，如 "nodeA" 而不是 "node1"。此字段是必需的。
topologyKeys	数组（字符串）	topologyKeys 是驱动程序支持的键列表。当在集群中初始化驱动程序时，它会提供一组拓扑键，用于理解（例如："company.com/zone", "company.com/region"）。当驱动程序在节点上初始化时，它会提供相同的拓扑键以及值。kubelet 会将这些拓扑键公开为其自身的节点对象上的标签。当 Kubernetes 进行拓扑感知置备时，它可以使用此列表来决定它应该从节点对象检索的标签，并传递给驱动程序。不同的节点可以使用不同的拓扑密钥。如果驱动程序不支持拓扑，则可以为空。

27.3.1.4. .spec.drivers[].allocatable

描述

VolumeNodeResources 是一组用于调度卷的资源限值。

类型

object

属性	类型	描述
数量	整数	count 表示节点上可以使用的 CSI 驱动程序管理的最大唯一卷数量。在节点上附加和挂载的卷被视为一次，而不是两次。相同的规则适用于同一节点上的多个 pod 共享的唯一卷。如果没有指定此字段，则此节点上支持的卷数量会绑定。

27.3.2. API 端点

可用的 API 端点如下：

- **/apis/storage.k8s.io/v1/csinodes**
 - **DELETE** : 删除 CSINode 集合
 - **GET**: 列出或监视类型 CSINode 的对象
 - **POST** : 创建一个 CSINode
- **/apis/storage.k8s.io/v1/watch/csinodes**
 - **GET**: 观察单个对 CSINode. deprecated 列表的更改：使用 'watch' 参数和一个 list 操作。
- **/apis/storage.k8s.io/v1/csinodes/{name}**
 - **DELETE** : 删除 CSINode
 - **GET**: 读取指定的 CSINode
 - **PATCH** : 部分更新指定的 CSINode
 - **PUT** : 替换指定的 CSINode
- **/apis/storage.k8s.io/v1/watch/csinodes/{name}**
 - **GET**: 观察对类型 CSINode. deprecated 的对象的更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

27.3.2.1. /apis/storage.k8s.io/v1/csinodes

表 27.27. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CSINode 集合

表 27.28. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 27.29. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.30. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型 CSINode 的对象

表 27.31. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 27.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSNodeList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 CSNode

表 27.33. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.34. 主体参数

参数	类型	描述
正文 (body)	CSNode 模式	

表 27.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSINode 模式
201 - Created	CSINode 模式
202 - Accepted	CSINode 模式
401 - Unauthorized	空

27.3.2.2. /apis/storage.k8s.io/v1/watch/csinodes

表 27.36. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CSINode.deprecated 列表的更改：使用 'watch' 参数和 list 操作。

表 27.37. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.3.2.3. /apis/storage.k8s.io/v1/csinodes/{name}

表 27.38. 全局路径参数

参数	类型	描述
name	string	CSINode 的名称

表 27.39. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CSINode

表 27.40. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 27.41. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSINode 模式
202 - Accepted	CSINode 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 CSINode

表 27.43. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSINode 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 CSINode

表 27.44. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.45. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.46. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSINode 模式
201 - Created	CSINode 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 CSINode

表 27.47. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.48. 主体参数

参数	类型	描述
正文 (body)	CSINode 模式	

表 27.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSINode 模式
201 - Created	CSINode 模式
401 - Unauthorized	空

27.3.2.4. /apis/storage.k8s.io/v1/watch/csinodes/{name}

表 27.50. 全局路径参数

参数	类型	描述
name	string	CSINode 的名称

表 27.51. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

监视对 kind CSINode.deprecated 的对象更改：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 27.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.4. CSISTORAGECAPACITY [STORAGE.K8S.IO/V1]

描述

CSISStorageCapacity 存储一个 CSI GetCapacity 调用的结果。对于给定的 StorageClass，这描述了特定拓扑片段中的可用容量。在考虑实例化新的 PersistentVolume 的位置时，可以使用它。

例如，这可以表达如下内容： - StorageClass "standard" has "1234 GiB" in "topology.kubernetes.io/zone=us-east1" - StorageClass "localssd" has "10 GiB" in "kubernetes.io/hostname=knode-abc123"

以下三种情况都意味着，没有可用于特定组合的容量： - 没有带有合适的拓扑和存储类名称的对象存在 - 此类对象存在，但容量未设置 - 这种对象存在，但容量为零

这些对象的制作人可以决定哪些方法更合适。

当 CSI 驱动程序选择使用 CSIDriverSpec.StorageCapacity 时，kube-scheduler 会消耗它们。调度程序将 MaximumVolumeSize 与请求的待处理卷大小进行比较，以过滤出 unsuitable 节点。如果未设置 MaximumVolumeSize，它将回退到与不太精确容量进行比较。如果这也未设置，调度程序会假定容量不足，并尝试其他节点。

类型

object

必填

- **storageClassName**

27.4.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
capacity	数量	<p>capacity 是 CSI 驱动程序在 GetCapacityResponse 中报告的值，它带有与上字段匹配的拓扑和参数的 GetCapacityResponse。</p> <p>语义目前(CSI spec 1.2)定义为：可用于置备卷的存储的可用容量（以字节为单位）。如果没有设置，则此信息当前不可用。</p>
kind	字符串	<p>kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</p>
maximumVolumeSize	数量	<p>maximumVolumeSize 是 CSI 驱动程序在 GetCapacityResponse 中为与上字段匹配的 topology 和 parameter 的 GetCapacityResponse 报告的值。</p> <p>这是因为 CSI spec 1.4.0 作为 CreateVolumeRequest.capacity_range.required_bytes 字段中使用的最大大小，创建一个与 GetCapacityRequest 中参数相同的卷。Kubernetes API 中的对应值是卷声明中的 ResourceRequirements.Requests。</p>

属性	类型	描述
metadata	ObjectMeta	<p>标准对象元数据。名称没有特定含义。它必须是 DNS 子域（允许，253 个字符）。为确保集群中没有与其他 CSI 驱动程序冲突，建议使用 <code>csisc-<uuid></code>、生成的名称或以唯一 CSI 驱动程序名称结尾的反向域名。</p> <p>对象是拥有命名空间的。</p> <p>更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</p>
nodeTopology	labelSelector	<p><code>nodeTopology</code> 定义了哪些节点有权访问报告容量的存储。如果没有设置，则无法从集群中的任何节点访问存储。如果为空，则可以从所有节点访问存储。此字段是不可变的。</p>
storageClassName	string	<p><code>storageClassName</code> 代表报告容量应用到的 <code>StorageClass</code> 的名称。它必须满足与 <code>StorageClass</code> 对象的名称相同的要求（非空、DNS 子域）。如果这个对象不再存在，<code>CSISStorageCapacity</code> 对象已过时，应该被其创建者删除。此字段是不可变的。</p>

27.4.2. API 端点

可用的 API 端点如下：

- **/apis/storage.k8s.io/v1/csistoragecapacities**
 - **GET**: 列表或监视类型 `CSISStorageCapacity` 的对象
- **/apis/storage.k8s.io/v1/watch/csistoragecapacities**
 - **GET**: 观察单个对 `CSISStorageCapacity` 列表的更改。已弃用：使用 'watch' 参数和一个 list 操作。
- **/apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities**
 - **DELETE**：删除 `CSISStorageCapacity` 的集合
 - **GET**: 列表或监视类型 `CSISStorageCapacity` 的对象

- **POST**: 创建一个 CSIStorageCapacity
- **/apis/storage.k8s.io/v1/watch/namespaces/{namespace}/csistoragecapacities**
 - **GET**: 观察单个对 CSIStorageCapacity 列表的更改。已弃用：使用 'watch' 参数和一个 list 操作。
- **/apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities/{name}**
 - **DELETE** : 删除 CSIStorageCapacity
 - **GET**: 阅读指定的 CSIStorageCapacity
 - **PATCH**: 部分更新指定的 CSIStorageCapacity
 - **PUT** : 替换指定的 CSIStorageCapacity
- **/apis/storage.k8s.io/v1/watch/namespaces/{namespace}/csistoragecapacities/{name}**
 - **GET**: 观察对类型 CSIStorageCapacity.deprecated 的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

27.4.2.1. /apis/storage.k8s.io/v1/csistoragecapacities

表 27.53. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存​​在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型 CSISStorageCapacity 的对象

表 27.54. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSISStorageCapacityList schema
401 - Unauthorized	空

27.4.2.2. /apis/storage.k8s.io/v1/watch/csisstoragecapacities

表 27.55. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CSISStorageCapacity 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 27.56. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.4.2.3. /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities

表 27.57. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 27.58. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CSISStorageCapacity 的集合

表 27.59. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 27.60. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.61. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型 CSISStorageCapacity 的对象

表 27.62. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 27.63. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSIStorageCapacityList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 CSIStorageCapacity

表 27.64. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.65. 主体参数

参数	类型	描述
正文 (body)	CSIStorageCapacity 模式	

表 27.66. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSISStorageCapacity 模式
201 - Created	CSISStorageCapacity 模式
202 - Accepted	CSISStorageCapacity 模式
401 - Unauthorized	空

27.4.2.4. /apis/storage.k8s.io/v1/watch/namespaces/{namespace}/csistoragecapacities

表 27.67. 全局路径参数

参数	类型	描述
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 27.68. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>

参数	类型	描述
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 CSIStorageCapacity 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 27.69. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.4.2.5. /apis/storage.k8s.io/v1/namespaces/{namespace}/csistoragecapacities/{name}

表 27.70. 全局路径参数

参数	类型	描述
name	string	CSIStorageCapacity 的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 27.71. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 CSIStorageCapacity

表 27.72. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则"orphan"终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。

表 27.73. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 27.74. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Status</code> 模式
202 - Accepted	<code>Status</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 `CSISStorageCapacity`

表 27.75. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>CSISStorageCapacity</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `CSISStorageCapacity`

表 27.76. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.77. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.78. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSStorageCapacity 模式
201 - Created	CSStorageCapacity 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 CSStorageCapacity

表 27.79. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.80. 主体参数

参数	类型	描述
正文 (body)	CSISStorageCapacity 模式	

表 27.81. HTTP 响应

HTTP 代码	响应正文
200 - OK	CSISStorageCapacity 模式
201 - Created	CSISStorageCapacity 模式
401 - Unauthorized	空

27.4.2.6. /apis/storage.k8s.io/v1/watch/namespaces/{namespace}/csistoragecapacities/{na

表 27.82. 全局路径参数

参数	类型	描述
name	string	CSISStorageCapacity 的名称
namespace	字符串	对象名称和身份验证范围，如团队 (team) 和项目 (project)

表 27.83. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind CSIStorageCapacity.deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 27.84. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.5. STORAGECLASS [STORAGE.K8S.IO/V1]

描述

StorageClass 描述了可以动态置备 PersistentVolume 的存储类的参数。

StorageClasses 是非命名空间；根据 etcd 的存储类的名称位于 ObjectMeta.Name 中。

类型

object

必填

- Provisioner

27.5.1. 规格

属性	类型	描述
allowVolumeExpansion	布尔值	allowVolumeExpansion 显示存储类是否允许卷扩展。
allowedTopologies	数组(TopologySelectorTerm)	allowedTopologies 限制可以动态置备卷的节点拓扑。每个卷插件定义自己的支持的拓扑规格。空 TopologySelectorTerm list 表示没有拓扑限制。此字段仅被启用 VolumeScheduling 功能的服务器所接受。
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
mountOptions	数组（字符串）	mountOptions 控制此存储类的动态置备 PersistentVolume 的 mountOptions。例如 ["ro", "soft"]。Not validated - 如果 PV 无效，则挂载将失败。
parameters	对象（字符串）	参数包含置备程序的参数，应该创建此存储类的卷。
Provisioner	string	provisioner 表示置备程序类型。
reclaimPolicy	string	reclaimPolicy 控制此存储类的动态置备 PersistentVolume 的 reclaimPolicy。默认为 Delete。 可能枚举值：- "Delete" 表示该卷将从其声明中的 Kubernetes 中删除。卷插件必须支持 Deletion。 - "Recycle" 意味着卷将被重新回收到其声明中未绑定持久性卷的池。卷插件必须支持 Recycling。 - "Retain" 表示该卷将保留在其当前阶段(Released)中，供管理员手动回收。默认策略为 Retain。
volumeBindingMode	string	volumeBindingMode 指明如何置备和绑定 PersistentVolumeClaim。如果未设置，则使用 VolumeBindingImmediate。此字段仅被启用 VolumeScheduling 功能的服务器所接受。 可能枚举值：- "Immediate" 表示 PersistentVolumeClaims 应该立即置备和绑定。这是默认模式。 - "WaitForFirstConsumer" 表示在创建引用 PersistentVolumeClaim 的第一个 Pod 前，不应置备和绑定 PersistentVolumeClaim。卷置备和绑定将在 Pod 调度过程中发生。

27.5.2. API 端点

可用的 API 端点如下：

- `/apis/storage.k8s.io/v1/storageclasses`
 - **DELETE** : 删除 StorageClass 的集合

- **GET**: 列出或监视类型为 StorageClass 的对象
- **POST** : 创建一个 StorageClass
- **/apis/storage.k8s.io/v1/watch/storageclasses**
 - **GET**: 观察单个对 StorageClass 列表的更改。已弃用 : 使用 'watch' 参数和 list 操作。
- **/apis/storage.k8s.io/v1/storageclasses/{name}**
 - **DELETE** : 删除 StorageClass
 - **GET** : 读取指定的 StorageClass
 - **PATCH** : 部分更新指定的 StorageClass
 - **PUT** : 替换指定的 StorageClass
- **/apis/storage.k8s.io/v1/watch/storageclasses/{name}**
 - **GET**: 观察对类型为 StorageClass 的对象的更改。已弃用 : 使用带有列表操作的 'watch' 参数, 而是过滤到带有 'fieldSelector' 参数的单个项目。

27.5.2.1. /apis/storage.k8s.io/v1/storageclasses

表 27.85. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 StorageClass 的集合

表 27.86. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 <code>continue</code> 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 <code>continue</code> 值，并带有相同的查询参数（除 <code>continue</code> 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 <code>continue</code> 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 <code>ResourceExpired</code> 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 <code>true</code> 时，不支持此字段。客户端可以从服务器返回的最后一个 <code>resourceVersion</code> 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 <code>nil</code>，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"'（用于向后兼容的原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 27.87. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.88. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 StorageClass 类型的对象

表 27.89. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 27.90. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClassList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 StorageClass

表 27.91. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.92. 主体参数

参数	类型	描述
正文 (body)	StorageClass 模式	

表 27.93. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClass 模式
201 - Created	StorageClass 模式
202 - Accepted	StorageClass 模式
401 - Unauthorized	空

27.5.2.2. /apis/storage.k8s.io/v1/watch/storageclasses

表 27.94. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 StorageClass 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 27.95. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.5.2.3. /apis/storage.k8s.io/v1/storageclasses/{name}

表 27.96. 全局路径参数

参数	类型	描述
name	string	StorageClass 的名称

表 27.97. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 StorageClass

表 27.98. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 27.99. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.100. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClass 模式
202 - Accepted	StorageClass 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 StorageClass

表 27.101. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClass 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 StorageClass

表 27.102. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.103. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.104. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClass 模式
201 - Created	StorageClass 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 StorageClass

表 27.105. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.106. 主体参数

参数	类型	描述
正文 (body)	StorageClass 模式	

表 27.107. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageClass 模式
201 - Created	StorageClass 模式
401 - Unauthorized	空

27.5.2.4. /apis/storage.k8s.io/v1/watch/storageclasses/{name}

表 27.108. 全局路径参数

参数	类型	描述
name	string	StorageClass 的名称

表 27.109. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法**GET****描述**

观察对类型为 StorageClass 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 27.110. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.6. VOLUMEATTACHMENT [STORAGE.K8S.IO/V1]

描述

VolumeAttachment 捕获将指定卷附加到指定节点的意图。

VolumeAttachment 对象不是命名空间。

类型

object

必填

- **spec**

27.6.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	VolumeAttachmentSpec 是 VolumeAttachment 请求的规格。
status	object	VolumeAttachmentStatus 是 VolumeAttachment 请求的状态。

27.6.1.1. .spec

描述

VolumeAttachmentSpec 是 VolumeAttachment 请求的规格。

类型

object

必填

- **attacher**
- **source**
- **nodeName**

属性	类型	描述
attacher	string	attacher 表示需要处理此请求的卷驱动程序的名称。这是 GetPluginName () 返回的名称。
nodeName	string	nodeName 代表卷应附加到的节点。
source	object	VolumeAttachmentSource 代表应该附加的卷。现在，只有 PersistenVolumes 才能通过外部附加程序附加，我们以后也可以允许 pod 中的内联卷。可以设置正好一个成员。

27.6.1.2. .spec.source

描述

VolumeAttachmentSource 代表应该附加的卷。现在，只有 PersistentVolumes 才能通过外部附加程序附加，我们以后也可以允许 pod 中的内联卷。可以设置正好一个成员。

类型

object

属性	类型	描述
inlineVolumeSpec	PersistentVolumeSpec	inlineVolumeSpec 包含附加 pod 内联 VolumeSource 定义的持久性卷所需的所有信息。此字段只针对 CSIMigration 功能填充。它包含 pod 的内联 VolumeSource 中的转换字段到 PersistentVolumeSpec。此字段是 beta 级别，仅适用于启用了 CSIMigration 功能的服务器。
persistentVolumeName	string	persistentVolumeName 代表要附加的持久性卷的名称。

27.6.1.3. .status

描述

VolumeAttachmentStatus 是 VolumeAttachment 请求的状态。

类型

object

必填

- **Attached**

属性	类型	描述
attachError	object	VolumeError 捕获卷操作过程中遇到的错误。
Attached	布尔值	attached 表示该卷已成功附加。此字段只能由完成附加操作的实体设置，即 external-attacher。
attachmentMetadata	对象（字符串）	attachmentMetadata 填充附加操作在成功附加后返回的任何信息，这些信息必须传递给后续的 WaitForAttach 或 Mount 调用。此字段只能由完成附加操作的实体设置，即 external-attacher。
detachError	object	VolumeError 捕获卷操作过程中遇到的错误。

27.6.1.4. .status.attachError

描述

VolumeError 捕获卷操作过程中遇到的错误。

类型

object

属性	类型	描述
message	string	Message 代表 Attach 或 Detach 操作过程中遇到的错误。可以记录此字符串，因此不应包含敏感信息。
time	时间	time 代表遇到错误的时间。

27.6.1.5. .status.detachError

描述

VolumeError 捕获卷操作过程中遇到的错误。

类型

object

属性	类型	描述
message	string	Message 代表 Attach 或 Detach 操作过程中遇到的错误。可以记录此字符串，因此不应包含敏感信息。
time	时间	time 代表遇到错误的时间。

27.6.2. API 端点

可用的 API 端点如下：

- **/apis/storage.k8s.io/v1/volumeattachments**
 - **DELETE** : 删除 VolumeAttachment 的集合
 - **GET**: 列出或监视类型为 VolumeAttachment 的对象
 - **POST** : 创建一个 VolumeAttachment
- **/apis/storage.k8s.io/v1/watch/volumeattachments**
 - **GET**: 观察单个对 VolumeAttachment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。
- **/apis/storage.k8s.io/v1/volumeattachments/{name}**

- **DELETE** : 删除 VolumeAttachment
 - **GET** : 阅读指定的 VolumeAttachment
 - **PATCH** : 部分更新指定的 VolumeAttachment
 - **PUT** : 替换指定的 VolumeAttachment
- **/apis/storage.k8s.io/v1/watch/volumeattachments/{name}**
 - **GET**: 观察对类型为 VolumeAttachment 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。
 - **/apis/storage.k8s.io/v1/volumeattachments/{name}/status**
 - **GET**: 指定 VolumeAttachment 的读取状态
 - **PATCH** : 部分更新指定 VolumeAttachment 的状态
 - **PUT** : 替换指定 VolumeAttachment 的状态

27.6.2.1. /apis/storage.k8s.io/v1/volumeattachments

表 27.111. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 VolumeAttachment 集合

表 27.112. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 27.113. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.114. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 VolumeAttachment 的对象

表 27.115. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 27.116. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachmentList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建一个 VolumeAttachment

表 27.117. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.118. 主体参数

参数	类型	描述
正文 (body)	VolumeAttachment 模式	

表 27.119. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
201 - Created	VolumeAttachment 模式
202 - Accepted	VolumeAttachment 模式
401 - Unauthorized	空

27.6.2.2. /apis/storage.k8s.io/v1/watch/volumeattachments

表 27.120. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 VolumeAttachment 列表的更改。已弃用：改为使用 'watch' 参数和 list 操作。

表 27.121. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.6.2.3. /apis/storage.k8s.io/v1/volumeattachments/{name}

表 27.122. 全局路径参数

参数	类型	描述
name	string	VolumeAttachment 的名称

表 27.123. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除卷附加

表 27.124. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 27.125. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 27.126. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
202 - Accepted	VolumeAttachment 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

阅读指定的 VolumeAttachment

表 27.127. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 VolumeAttachment

表 27.128. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.129. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.130. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
201 - Created	VolumeAttachment 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 VolumeAttachment

表 27.131. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.132. 主体参数

参数	类型	描述
正文 (body)	VolumeAttachment 模式	

表 27.133. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
201 - Created	VolumeAttachment 模式
401 - Unauthorized	空

27.6.2.4. /apis/storage.k8s.io/v1/watch/volumeattachments/{name}

表 27.134. 全局路径参数

参数	类型	描述
name	string	VolumeAttachment 的名称

表 27.135. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 VolumeAttachment 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 27.136. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

27.6.2.5. /apis/storage.k8s.io/v1/volumeattachments/{name}/status

表 27.137. 全局路径参数

参数	类型	描述
name	string	VolumeAttachment 的名称

表 27.138. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定 VolumeAttachment 的读取状态

表 27.139. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 VolumeAttachment 的部分更新状态

表 27.140. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 27.141. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 27.142. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
201 - Created	VolumeAttachment 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 VolumeAttachment 的状态

表 27.143. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 27.144. 主体参数

参数	类型	描述
正文 (body)	VolumeAttachment 模式	

表 27.145. HTTP 响应

HTTP 代码	响应正文
200 - OK	VolumeAttachment 模式
201 - Created	VolumeAttachment 模式
401 - Unauthorized	空

第 28 章 存储版本迁移 API

28.1. 存储版本迁移 API

28.1.1. StorageVersionMigration [migration.k8s.io/v1alpha1]

描述

StorageVersionMigration 代表存储的数据迁移到最新存储版本。

类型

object

28.2. STORAGEVERSIONMIGRATION [MIGRATION.K8S.IO/V1ALPHA1]

描述

StorageVersionMigration 代表存储的数据迁移到最新存储版本。

类型

对象

28.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	迁移规格。

属性	类型	描述
status	object	迁移的状态。

28.2.1.1. .spec

描述

迁移规格。

类型

object

必填

- **resource**

属性	类型	描述
continueToken	string	列表选项中使用的令牌获取要迁移的下一个对象块。当 <code>.status.conditions</code> 表示迁移为 "Running" 时，用户可以使用此令牌来检查迁移的进度。
resource	object	正在迁移的资源。migrator 将请求发送到为资源提供服务的端点。不可变。

28.2.1.2. .spec.resource

描述

正在迁移的资源。migrator 将请求发送到为资源提供服务的端点。不可变。

类型

object

属性	类型	描述
group	string	组的名称。
resource	string	资源名称。
version	string	版本的名称。

28.2.1.3. .status

描述

迁移的状态。

类型

object

属性	类型	描述
conditions	数组	迁移当前状态的最新可用影响。
conditions[]	object	描述迁移在特定时间点上的状态。

28.2.1.4. .status.conditions

描述

迁移当前状态的最新可用影响。

类型

array

28.2.1.5. .status.conditions[]

描述

描述迁移在特定时间点上的状态。

类型

object

必填

- **status**
- **type**

属性	类型	描述
lastUpdateTime	string	最后一次更新此条件的时间。
message	字符串	人类可读的消息，指示有关转换的详细信息。
reason	字符串	条件最后一次转换的原因。
status	字符串	条件的状态，True, False, Unknown 之一。
type	string	条件的类型。

28.2.2. API 端点

可用的 API 端点如下：

- **/apis/migration.k8s.io/v1alpha1/storageversionmigrations**

- **DELETE** : 删除 StorageVersionMigration 的集合
- **GET**: 列出类型为 StorageVersionMigration 的对象
- **POST**: 创建 StorageVersionMigration
- **/apis/migration.k8s.io/v1alpha1/storageversionmigrations/{name}**
 - **DELETE** : 删除 StorageVersionMigration
 - **GET**: 读取指定的 StorageVersionMigration
 - **PATCH** : 部分更新指定的 StorageVersionMigration
 - **PUT** : 替换指定的 StorageVersionMigration
- **/apis/migration.k8s.io/v1alpha1/storageversionmigrations/{name}/status**
 - **GET**: 指定 StorageVersionMigration 的读取状态
 - **PATCH** : 部分更新指定 StorageVersionMigration 的状态
 - **PUT** : 替换指定 StorageVersionMigration 的状态

28.2.2.1. /apis/migration.k8s.io/v1alpha1/storageversionmigrations

表 28.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 StorageVersionMigration 的集合

表 28.2. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>

参数	类型	描述
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 28.3. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法
GET

描述

列出类型为 StorageVersionMigration 的对象

表 28.4. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 28.5. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigrationList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 StorageVersionMigration

表 28.6. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 28.7. 主体参数

参数	类型	描述
正文 (body)	StorageVersionMigration 模式	

表 28.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
201 - Created	StorageVersionMigration 模式
202 - Accepted	StorageVersionMigration 模式
401 - Unauthorized	空

28.2.2.2. /apis/migration.k8s.io/v1alpha1/storageversionmigrations/{name}

表 28.9. 全局路径参数

参数	类型	描述
name	string	StorageVersionMigration 的名称

表 28.10. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 StorageVersionMigration

表 28.11. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 28.12. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 28.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的 StorageVersionMigration

表 28.14. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 28.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 StorageVersionMigration

表 28.16. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 28.17. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 28.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 StorageVersionMigration

表 28.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 28.20. 主体参数

参数	类型	描述
正文 (body)	StorageVersionMigration 模式	

表 28.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
201 - Created	StorageVersionMigration 模式
401 - Unauthorized	空

28.2.2.3. /apis/migration.k8s.io/v1alpha1/storageversionmigrations/{name}/status

表 28.22. 全局路径参数

参数	类型	描述
----	----	----

参数	类型	描述
name	string	StorageVersionMigration 的名称

表 28.23. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法**GET****描述**

指定 StorageVersionMigration 的读取状态

表 28.24. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 28.25. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

指定 StorageVersionMigration 的部分更新状态

表 28.26. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 28.27. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 28.28. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 StorageVersionMigration 的状态

表 28.29. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 28.30. 主体参数

参数	类型	描述
正文 (body)	StorageVersionMigration 模式	

表 28.31. HTTP 响应

HTTP 代码	响应正文
200 - OK	StorageVersionMigration 模式
201 - Created	StorageVersionMigration 模式
401 - Unauthorized	空

第 29 章 TOPOLVM API

29.1. TOPOLVM API

29.1.1. LogicalVolume [topolvm.io/v1]

描述

logicalvolume 是 logicalvolumes API 的 Schema

类型

object

29.2. LOGICALVOLUME [TOPOLVM.IO/V1]

描述

logicalvolume 是 logicalvolumes API 的 Schema

类型

对象

29.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

属性	类型	描述
spec	object	LogicalVolumeSpec 定义 LogicalVolume 的所需状态
status	object	LogicalVolumeStatus 定义 LogicalVolume 的观察状态

29.2.1.1. .spec

描述

LogicalVolumeSpec 定义 LogicalVolume 的所需状态

类型

object

必填

- **name**
- **nodeName**
- **size**

属性	类型	描述
accesstype	string	'accesstype' 指定用户计划如何使用快照逻辑卷。在创建快照时，设置为"ro"，在恢复快照或创建克隆时设置为"rw"。只有在 LogicalVolume 有源时才会填充此字段。
deviceClass	string	
name	string	
nodeName	string	
size	integer-or-string	
source	string	'source' 指定源的 logicalvolume 名称；如果存在。只有在 LogicalVolume 有源时才会填充此字段。

29.2.1.2. .status

描述

LogicalVolumeStatus 定义 LogicalVolume 的观察状态

类型

object

属性	类型	描述
code	整数	Code 是一个未签名的 32 位错误代码，如 gRPC spec 中定义的。
currentSize	integer-or-string	
message	string	
volumeID	string	INSERT ADDITIONAL STATUS FIELD - 定义在集群 Important: 在修改此文件后运行"make"以重新生成代码

29.2.2. API 端点

可用的 API 端点如下：

- **/apis/topolvm.io/v1/logicalvolumes**
 - **DELETE** : 删除 LogicalVolume 的集合
 - **GET** : 列出类型为 LogicalVolume 的对象
 - **POST** : 创建 LogicalVolume
- **/apis/topolvm.io/v1/logicalvolumes/{name}**
 - **DELETE** : 删除 LogicalVolume
 - **GET**: 读取指定的 LogicalVolume
 - **PATCH** : 部分更新指定的 LogicalVolume
 - **PUT** : 替换指定的 LogicalVolume
- **/apis/topolvm.io/v1/logicalvolumes/{name}/status**
 - **GET** : 指定 LogicalVolume 的读取状态
 - **PATCH** : 部分更新指定 LogicalVolume 的状态
 - **PUT** : 替换指定 LogicalVolume 的状态

29.2.2.1. /apis/topolvm.io/v1/logicalvolumes

表 29.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 LogicalVolume 的集合

表 29.2. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 29.3. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出类型为 LogicalVolume 的对象

表 29.4. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 29.5. HTTP 响应

HTTP 代码	响应正文
200 - OK	LogicalVolumeList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 LogicalVolume

表 29.6. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 29.7. 主体参数

参数	类型	描述
正文 (body)	logicalvolume 模式	

表 29.8. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
201 - Created	logicalvolume 模式
202 - Accepted	logicalvolume 模式
401 - Unauthorized	空

29.2.2.2. /apis/topolvm.io/v1/logicalvolumes/{name}

表 29.9. 全局路径参数

参数	类型	描述
name	string	LogicalVolume 的名称

表 29.10. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 LogicalVolume

表 29.11. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 29.12. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 29.13. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

读取指定的 LogicalVolume

表 29.14. 查询参数

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 29.15. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 LogicalVolume

表 29.16. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 29.17. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 29.18. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 LogicalVolume

表 29.19. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 29.20. 主体参数

参数	类型	描述
正文 (body)	logicalvolume 模式	

表 29.21. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
201 - Created	logicalvolume 模式
401 - Unauthorized	空

29.2.2.3. /apis/topolvm.io/v1/logicalvolumes/{name}/status

表 29.22. 全局路径参数

参数	类型	描述
name	string	LogicalVolume 的名称

表 29.23. 全局查询参数

参数	类型	描述
<code>pretty</code>	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

GET

描述

指定的 LogicalVolume 的读取状态

表 29.24. 查询参数

参数	类型	描述
<code>resourceVersion</code>	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置

表 29.25. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>logicalvolume</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

指定 LogicalVolume 的部分更新状态

表 29.26. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时, 表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 29.27. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 29.28. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定 LogicalVolume 的状态

表 29.29. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 29.30. 主体参数

参数	类型	描述
正文 (body)	logicalvolume 模式	

表 29.31. HTTP 响应

HTTP 代码	响应正文
200 - OK	logicalvolume 模式
201 - Created	logicalvolume 模式
401 - Unauthorized	空

第 30 章 WEBHOOK API

30.1. WEBHOOK API

30.1.1. MutatingWebhookConfiguration [admissionregistration.k8s.io/v1]

描述

MutatingWebhookConfiguration 描述了接受或拒绝的和 admission Webhook 的配置，并可能会更改对象。

类型

对象

30.1.2. ValidatingWebhookConfiguration [admissionregistration.k8s.io/v1]

描述

ValidatingWebhookConfiguration 描述了接受或拒绝和对象的配置，而不更改它。

类型

对象

30.2. MUTATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

描述

MutatingWebhookConfiguration 描述了接受或拒绝的和 admission Webhook 的配置，并可能会更改对象。

类型

对象

30.2.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

属性	类型	描述
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata 。
Webhook	array	Webhook 是一个 webhook 列表，以及受影响的资源和操作。
webhooks[]	对象	变异webhook 描述了准入 Webhook，以及它应用到的资源和操作。

30.2.1.1. .webhooks

描述

Webhook 是一个 webhook 列表，以及受影响的资源和操作。

类型

array

30.2.1.2. .webhooks[]

描述

变异webhook 描述了准入 Webhook，以及它应用到的资源和操作。

类型

object

必填

- **name**
- **clientConfig**
- **sideEffects**
- **admissionReviewVersions**

属性	类型	描述
admissionReviewVersions	数组（字符串）	admissionReviewVersions 是 Webhook 期望的首选 AdmissionReview 版本的顺序列表。API 服务器将尝试使用其支持的列表中的第一个版本。如果 API 服务器支持的此列表中没有指定的版本，则对此对象验证将失败。如果持久的 Webhook 配置指定了允许的版本，且不包含任何已知的 API 服务器版本，则对 Webhook 的调用将失败，并受到失败策略的影响。
clientConfig	对象	WebhookClientConfig 包含与 Webhook 进行 TLS 连接的信息
failurePolicy	字符串	failurePolicy 定义如何处理准入端点中的未识别错误 - 允许的值为 Ignore 或 Fail。默认为 Fail。 可能枚举值：- "Fail" 表示调用 Webhook 的错误会导致准入失败。- "Ignore" 表示调用 webhook 的错误被忽略。
matchConditions	数组	MatchConditions 是必须满足的条件列表，才能将请求发送到此 webhook。匹配条件过滤请求，它们已与 rules、namespaceSelector 和 objectSelector 匹配。 matchConditions 空列表匹配所有请求。允许最多有 64 个匹配条件。 完全匹配的逻辑是（按顺序排列）：1. 如果 ANY matchCondition 评估为 FALSE，则会跳过 Webhook。2. 如果 ALL matchConditions 评估为 TRUE，则调用 Webhook。3. 如果任何 matchCondition 评估为错误（但没有为 FALSE）：- 如果 failurePolicy=Fail，则拒绝请求 - 如果 failurePolicy=Ignore，则忽略错误并跳过 Webhook 这是一个 alpha 功能，由 AdmissionWebhookMatchConditions 功能门管理。

属性	类型	描述
matchConditions[]	object	MatchCondition 代表必须实现才能使请求发送到 webhook 的条件。
matchPolicy	字符串	<p>matchPolicy 定义如何使用 "rules" 列表来匹配传入的请求。允许的值是 "Exact" 或 "Equivalent"。</p> <p>- exact : 仅在与指定规则完全匹配时才匹配请求。例如, 如果可通过 apps/v1、apps/v1beta1 和 extensions/v1beta1 修改部署, 但 "rules" 只包括 apiGroups: ["apps"], apiVersions: ["v1"], resources: ["deployments"], 对 apps/v1beta1 或 extensions/v1beta1 的请求不会发送到 webhook。</p> <p>- equivalent : 在修改规则中列出的资源时匹配请求, 即使通过另一个 API 组或版本。例如, 如果部署可以通过 apps/v1、apps/v1beta1 和 extensions/v1beta1 修改, 且 "rules" 只包括 apiGroups: ["apps"], apiVersions: ["v1"], resources: ["deployments"], 对 apps/v1beta1 或 extensions/v1beta1 的请求将转换为 apps/v1beta1 并发送到 webhook。</p> <p>默认为 "Equivalent"</p> <p>可能枚举值: - "Equivalent" 表示如果请求通过另一个 API 组或版本修改规则中列出的资源, 则应该向 webhook 发送请求。- "Exact" 表示请求应仅发送到 webhook (如果请求完全与给定规则匹配)。</p>

属性	类型	描述
name	字符串	准入 Webhook 的名称。name 应该是完全限定的，如 imagepolicy.kubernetes.io，其中 "imagepolicy" 是 Webhook 的名称，kubernetes.io 是机构的名称。必需。
namespaceSelector	labelSelector	<p>namespaceSelector 决定是否在对象上运行 webhook，具体取决于该对象的命名空间是否与选择器匹配。如果对象本身是一个命名空间，则匹配会在 object.metadata.labels 上执行。如果对象是另一个集群范围的资源，它永远不会跳过 Webhook。</p> <p>例如，要在命名空间没有与 "0" 或 "1" 关联的 "runlevel" 或 "1" 运行 webhook；您要将选择器设置为如下：</p> <pre>"namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }</pre> <p>如果您只想在命名空间与 "prod" 或 "staging" 关联的 "environment" 或 "staging" 对象上运行 webhook；您要将选择器设置为如下：</p> <pre>"namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }</pre> <p>有关标签选择器的更多示例，请参阅 https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/。</p> <p>默认为空 LabelSelector，它与所有内容匹配。</p>

属性	类型	描述
objectSelector	labelSelector	<p>objectSelector 决定是否根据对象是否有匹配的 labels。objectSelector 运行 webhook。objectSelector 是针对要发送到 webhook 的 oldObject 和 newObject 评估的，并被视为匹配对象是否与选择器匹配。null 对象（如果为 create，则为 newObject，则为 newObject）或没有标签（如 DeploymentRollback 或 PodProxyOptions 对象）的对象不被视为匹配。只有在 Webhook 是 opt-in 时，才使用对象选择器，因为最终用户可以通过设置标签来跳过准入 Webhook。默认为空 LabelSelector，它与所有内容匹配。</p>
reinvocationPolicy	字符串	<p>reinvocationPolicy 指明此 webhook 是否作为单一准入评估的一部分被多次调用。允许的值是 "Never" 和 "IfNeeded"。</p> <p>Never：在单个准入评估中，webhook 不会多次调用。</p> <p>IfNeeded：如果在初始 Webhook 调用后被其他插件修改了其他准入插件，则 webhook 最少会作为准入评估的一部分被调用。指定这个选项的 Webhook 必须 是幂等的，可以处理之前接受的对象。注：* 无法保证额外调用的数量完全相同。* 如果额外的调用结果对对象进行进一步的修改，则无法保证再次调用 Webhook。* 使用此选项的 Webhook 可以重新排序来最小化附加调用的数量。* 要在保证完成所有变异后验证对象，请使用验证准入 Webhook。</p> <p>默认为 "Never"。</p> <p>可能的 enum 值：- "IfNeeded" 表示如果初始 webhook 调用后其他准入插件被其他准入插件修改，则 webhook 必须至少称为 "Never" 作为准入评估的一部分。</p>

属性	类型	描述
rules	array	规则描述了 Webhook 关注的 resources/subresources 的操作。如果与任何规则匹配，webhook 会处理某个操作。但是，为了防止 ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 使集群处于无法完全禁用插件的状态，ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 永远不会调用 ValidatingWebhookConfiguration 和 MutatingWebhookConfiguration 对象的准入请求。
rules[]	对象	RuleWithOperations 是操作和资源的元组。建议确保所有元组扩展都有效。
sideEffects	字符串	<p>SideEffects 指出此 webhook 是否产生副作用。可接受值为：None, NoneOnDryRun (webhooks via v1beta1 创建，也可以指定 some 或 Unknown)。具有副作用的 Webhook 实现了协调系统，因为准入链中的将来步骤可能会拒绝请求，因此需要撤销副作用。具有 dryRun 属性的请求如果与带有 sideEffects == Unknown 或 some 的 webhook 匹配，则会自动拒绝。</p> <p>可能的 enum 值：- "None" 表示调用 Webhook 不会有副作用。- "NoneOnDryRun" 表示调用 Webhook 可能会产生副作用，但如果检查了请求具有 dry-run 属性，则会隐藏副作用。- "Some" 表示调用 Webhook 可能会产生副作用。如果具有 dry-run 属性的请求会触发调用此 webhook，则请求将失败。- "Unknown" 表示没有有关调用 webhook 的副作用的信息。如果具有 dry-run 属性的请求会触发对此 webhook 的调用，则请求将失败。</p>

属性	类型	描述
timeoutSeconds	整数	timeoutSeconds 指定此 webhook 的超时时间。通过超时后，webhook 调用将被忽略，或者 API 调用将根据失败策略失败。超时值必须在 1 到 30 秒之间。默认值为 10 秒。

30.2.1.3. .webhooks[].clientConfig

描述

WebhookClientConfig 包含与 Webhook 进行 TLS 连接的信息

类型

object

属性	类型	描述
caBundle	字符串	cabundle 是一个 PEM 编码的 CA 捆绑包，用于验证 webhook 的服务器证书。如果未指定，则使用 apiserver 中的系统信任 root。
service	对象	ServiceReference 包含对 Service.legacy.k8s.io 的引用
url	字符串	URL 以标准 URL 格式提供 webhook 的位置 (scheme://host:port/path)。必须指定 url 或服务 之一。 主机 不应引用集群中运行的服务；改为使用 service 字段。在一些 apiservers（如 kube-apiserver）中，主机可以通过外部 DNS 解析，因为 kube-apiserver 无法解析集群 DNS，因为这将是分层违反情况。主机 也可以是 IP 地址。 请注意，使用 localhost 或 127.0.0.1 作为主机有风险，除非您特别注意在运行 apiserver 的所有主机上运行这

属性	类型	描述
		<p>个 webhook，这可能需要调用此 webhook。此类安装可能是不可移植的，例如，无法在新集群中轻松打开。</p> <p>方案必须是 "https"；URL 必须以 "https://" 开头。</p> <p>路径是可选的，如果存在，则 URL 中可能存在任何字符串。您可以使用路径将任意字符串传递给 Webhook，例如集群标识符。</p> <p>尝试使用用户或基本身份验证，例如不允许 "user:password@"。不允许片段("#...")和查询参数("?...")。</p>

30.2.1.4. .webhooks[].clientConfig.service

描述

ServiceReference 包含对 Service.legacy.k8s.io 的引用

类型

object

必填

- namespace
- name

属性	类型	描述
name	字符串	name 是服务的名称。必填

属性	类型	描述
namespace	字符串	namespace 是服务的命名空间。必填
path	字符串	path 是一个可选 URL 路径，将在此服务的任何请求中发送。
port	整数	如果指定，托管 Webhook 的服务上的端口。默认为 443，以便向后兼容。 端口 应该是有效的端口号 (1-65535，包含)。

30.2.1.5. .webhooks[].matchConditions

描述

MatchConditions 是必须满足的条件列表，才能将请求发送到此 webhook。匹配条件过滤请求，它们已与 rules、namespaceSelector 和 objectSelector 匹配。matchConditions 空列表匹配所有请求。允许最多有 64 个匹配条件。

完全匹配的逻辑是（按顺序排列）：1. 如果 ANYmatchCondition 评估为 FALSE，则会跳过 Webhook。2. 如果 ALLmatchConditions 评估为 TRUE，则调用 Webhook。3. 如果任何matchCondition 评估为错误（但没有为 FALSE）：- 如果 failurePolicy=Fail，则拒绝请求 - 如果 failurePolicy=Ignore，则忽略错误并跳过 Webhook

这是一个 alpha 功能，由 AdmissionWebhookMatchConditions 功能门管理。

类型

数组

30.2.1.6. .webhooks[].matchConditions[]

描述

MatchCondition 代表必须实现才能使请求发送到 webhook 的条件。

类型

object

必填

- **name**
- **expression**

属性	类型	描述
----	----	----

属性	类型	描述
expression	string	<p>表达式代表 CEL 将评估的表达式。必须评估 bool。CEL 表达式可以访问 AdmissionRequest 和 Authorizer 的内容，被组织为 CEL 变量：</p> <p>'object' - 传入请求中的对象。 DELETE 请求的值是 null。 'oldObject' - 现有对象。对于 CREATE requests，值为 null。 'request' - 准入请求的属性 (/pkg/apis/admission/types.go#AdmissionRequest). 'authorizer' - A CEL Authorizer。可用于对请求的主（用户或服务帐户）执行授权检查。请参阅 https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz 'authorizer.requestResource' - 从 'authorizer' 构建的 CEL ResourceCheck，并使用 request 资源进行配置。CEL 的文档： https://kubernetes.io/docs/reference/using-api/cel/</p> <p>必需。</p>
name	string	<p>name 是此匹配条件的标识符，用于 MatchConditions 的策略合并，并提供用于日志目的的标识符。相关表达式应当有一个好的名称。name 必须是由字母数字字符、'-','_' 或 '.' 组成的合格名称，且必须以字母数字字符开头和结尾（例如：'MyName' 或 'my.name'，或 '123-abc'，regex is '([A-Za-z0-9]([A-Za-z0-9.]*)?[A-Za-z0-9])'），带有可选的 DNS 子域前缀和 '/'（如 'example.com/MyName'）</p> <p>必需。</p>

30.2.1.7. .webhooks[].rules

描述

规则描述了 Webhook 关注的 resources/subresources 的操作。如果与任何规则匹配，webhook 会处理某个操作。但是，为了防止 ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 使集群处于无法完全禁用插件的状态，ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 永远不会调用 ValidatingWebhookConfiguration 和 MutatingWebhookConfiguration 对象的准入请求。

类型

array

30.2.1.8. .webhooks[].rules[]

描述

RuleWithOperations 是操作和资源的元组。建议确保所有元组扩展都有效。

类型

object

属性	类型	描述
apiGroups	数组（字符串）	apiGroups 是资源所属的 API 组。"是所有组。如果 ' 存在，分片的长度必须为一。必需。
apiVersions	数组（字符串）	APIVersions 是资源所属的 API 版本。"是所有版本。如果 ' 存在，分片的长度必须为一。必需。
操作	数组（字符串）	操作是准入 hook 关注的操作 - CREATE、UPDATE、DELETE、CONNECT 或 *（对于所有这些操作）以及添加的任何未来准入操作。如果 '*' 存在，分片的长度必须为一。必需。
资源	数组（字符串）	resources 是一组这个规则应用到资源列表。 例如：'pods' 表示 pod。 'pods/log' 表示 pod 的日志子资源。'表示所有资源，但不是子资源。'pod/' 表示所有 pod 的子资源。'/scale' 表示所有扩展子资源。'Burstable' 表示所有资源及其子资源。 如果存在通配符，验证规则将确保资源不会相互重叠。 根据括起对象，可能不允许子资源。必需。
scope	字符串	scope 指定此规则的范围。有效值为 "Cluster", "Namespaced", 和 "" "Cluster" 表示只有集群范围的资源与此规则匹配。命名空间 API 对象是集群范围的。 "namespaced" 意味着只有命名空间的资源与此规则匹配。"表示没有范围限制。子资源与其父资源的范围匹配。默认为 "*"。

30.2.2. API 端点

可用的 API 端点如下：

- **/apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations**
 - **DELETE** : 删除 MutatingWebhookConfiguration 的集合
 - **GET**: 列出或监视 kind MutatingWebhookConfiguration 的对象
 - **POST**: 创建一个 MutatingWebhookConfiguration
- **/apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations**
 - **GET**: 观察单个对 MutatingWebhookConfiguration 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/{name}**
 - **DELETE** : 删除 MutatingWebhookConfiguration
 - **GET**: 读取指定的 MutatingWebhookConfiguration
 - **PATCH**: 部分更新指定的 MutatingWebhookConfiguration
 - **PUT** : 替换指定的 MutatingWebhookConfiguration
- **/apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations/{name}**
 - **GET**: 观察对 kind MutatingWebhookConfiguration 对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

30.2.2.1. /apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations

表 30.1. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 MutatingWebhookConfiguration 集合

表 30.2. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated : 请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 30.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 30.4. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视 kind MutatingWebhookConfiguration 的对象

表 30.5. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 30.6. HTTP 响应

HTTP 代码	响应正文
200 - OK	MutatingWebhookConfigurationList schema
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 MutatingWebhookConfiguration

表 30.7. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 30.8. 主体参数

参数	类型	描述
正文 (body)	MutatingWebhookConfiguration 模式	

表 30.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	MutatingWebhookConfiguration 模式
201 - Created	MutatingWebhookConfiguration 模式
202 - Accepted	MutatingWebhookConfiguration 模式
401 - Unauthorized	空

30.2.2.2. /apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations

表 30.10. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	<p>如果为 'true'，则输出会经过 pretty print 处理。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 MutatingWebhookConfiguration 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 30.11. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

30.2.2.3. /apis/admissionregistration.k8s.io/v1/mutatingwebhookconfigurations/{name}

表 30.12. 全局路径参数

参数	类型	描述
name	字符串	MutatingWebhookConfiguration 的名称

表 30.13. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

delete a MutatingWebhookConfiguration

表 30.14. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 30.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 30.16. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

读取指定的 MutatingWebhookConfiguration

表 30.17. HTTP 响应

HTTP 代码	响应正文
200 - OK	MutatingWebhookConfiguration 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 MutatingWebhookConfiguration

表 30.18. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 30.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 30.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	MutatingWebhookConfiguration 模式
201 - Created	MutatingWebhookConfiguration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 MutatingWebhookConfiguration

表 30.21. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 30.22. 主体参数

参数	类型	描述
正文 (body)	MutatingWebhookConfiguration 模式	

表 30.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	MutatingWebhookConfiguration 模式
201 - Created	MutatingWebhookConfiguration 模式
401 - Unauthorized	空

30.2.2.4. /apis/admissionregistration.k8s.io/v1/watch/mutatingwebhookconfigurations/{name}

表 30.24. 全局路径参数

参数	类型	描述
name	字符串	MutatingWebhookConfiguration 的名称

表 30.25. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind MutatingWebhookConfiguration 对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 30.26. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

30.3. VALIDATINGWEBHOOKCONFIGURATION [ADMISSIONREGISTRATION.K8S.IO/V1]

描述

ValidatingWebhookConfiguration 描述了接受或拒绝和对象的配置，而不更改它。

类型

对象

30.3.1. 规格

属性	类型	描述
apiVersion	字符串	APIVersion 定义对象的这个表示法的版本化的 schema。服务器应该将识别的模式转换为最新的内部值，并可拒绝未识别的值。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	字符串	kind 是一个字符串值，代表此对象所代表的 REST 资源。服务器可以从客户端向其提交请求的端点推断。无法更新。采用驼峰拼写法 (CamelCase)。更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	标准对象元数据；更多信息： https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata 。

属性	类型	描述
Webhook	array	Webhook 是一个 webhook 列表，以及受影响的资源和操作。
webhooks[]	对象	验证webhook 描述了准入 Webhook，以及它应用到的资源和操作。

30.3.1.1. .webhooks

描述

Webhook 是一个 webhook 列表，以及受影响的资源和操作。

类型

array

30.3.1.2. .webhooks[]

描述

验证webhook 描述了准入 Webhook，以及它应用到的资源和操作。

类型

object

必填

- **name**
- **clientConfig**
- **sideEffects**
- **admissionReviewVersions**

属性	类型	描述
admissionReviewVersions	数组（字符串）	admissionReviewVersions 是 Webhook 期望的首选 AdmissionReview 版本的顺序列表。API 服务器将尝试使用其支持的列表中的第一个版本。如果 API 服务器支持的此列表中没有指定的版本，则对此对象验证将失败。如果持久的 Webhook 配置指定了允许的版本，且不包含任何已知的 API 服务器版本，则对 Webhook 的调用将失败，并受到失败策略的影响。
clientConfig	对象	WebhookClientConfig 包含与 Webhook 进行 TLS 连接的信息

属性	类型	描述
failurePolicy	字符串	<p>failurePolicy 定义如何处理准入端点中的未识别错误 - 允许的值为 Ignore 或 Fail。默认为 Fail。</p> <p>可能枚举值： - "Fail" 表示调用 Webhook 的错误会导致准入失败。 - "Ignore" 表示调用 webhook 的错误被忽略。</p>
matchConditions	数组	<p>MatchConditions 是必须满足的条件列表，才能将请求发送到此 webhook。匹配条件过滤请求，它们已与 rules、namespaceSelector 和 objectSelector 匹配。</p> <p>matchConditions 空列表匹配所有请求。允许最多有 64 个匹配条件。</p> <p>完全匹配的逻辑是（按顺序排列）：1. 如果 ANY matchCondition 评估为 FALSE，则会跳过 Webhook。2. 如果 ALL matchConditions 评估为 TRUE，则调用 Webhook。3. 如果任何 matchCondition 评估为错误（但没有为 FALSE）： - 如果 failurePolicy=Fail，则拒绝请求 - 如果 failurePolicy=Ignore，则忽略错误并跳过 Webhook</p> <p>这是一个 alpha 功能，由 AdmissionWebhookMatchConditions 功能门管理。</p>
matchConditions[]	object	<p>MatchCondition 代表必须实现才能使请求发送到 webhook 的条件。</p>

属性	类型	描述
matchPolicy	字符串	<p>matchPolicy 定义如何使用 "rules" 列表来匹配传入的请求。允许的值是 "Exact" 或 "Equivalent"。</p> <ul style="list-style-type: none"> - exact : 仅在与指定规则完全匹配时才匹配请求。例如, 如果可通过 apps/v1、apps/v1beta1 和 extensions/v1beta1 修改部署, 但 "rules" 只包括 apiGroups: ["apps"], apiVersions: ["v1"], resources: ["deployments"], 对 apps/v1beta1 或 extensions/v1beta1 的请求不会发送到 webhook。 - equivalent : 在修改规则中列出的资源时匹配请求, 即使通过另一个 API 组或版本。例如, 如果部署可以通过 apps/v1、apps/v1beta1 和 extensions/v1beta1 修改, 且 "rules" 只包括 apiGroups: ["apps"], apiVersions: ["v1"], resources: ["deployments"], 对 apps/v1beta1 或 extensions/v1beta1 的请求将转换为 apps/v1beta1 并发送到 webhook。 <p>默认为 "Equivalent"</p> <p>可能枚举值: - "Equivalent" 表示如果请求通过另一个 API 组或版本修改规则中列出的资源, 则应该向 webhook 发送请求。- "Exact" 表示请求应仅发送到 webhook (如果请求完全与给定规则匹配)。</p>
name	字符串	<p>准入 Webhook 的名称。name 应该是完全限定的, 如 imagepolicy.kubernetes.io, 其中 "imagepolicy" 是 Webhook 的名称, kubernetes.io 是机构的名称。必需。</p>

属性	类型	描述
namespaceSelector	labelSelector	<p>namespaceSelector 决定是否在对象上运行 webhook，具体取决于该对象的命名空间是否与选择器匹配。如果对象本身是一个命名空间，则匹配会在 object.metadata.labels 上执行。如果对象是另一个集群范围的资源，它永远不会跳过 Webhook。</p> <p>例如，要在命名空间没有与 "0" 或 "1" 关联的 "runlevel" 或 "1" 运行 webhook；您要将选择器设置为如下：</p> <pre>"namespaceSelector": { "matchExpressions": [{ "key": "runlevel", "operator": "NotIn", "values": ["0", "1"] }] }</pre> <p>如果您只想在命名空间与 "prod" 或 "staging" 关联的 "environment" 或 "staging" 对象上运行 webhook；您要将选择器设置为如下：</p> <pre>"namespaceSelector": { "matchExpressions": [{ "key": "environment", "operator": "In", "values": ["prod", "staging"] }] }</pre> <p>有关标签选择器的更多示例，请参阅 https://kubernetes.io/docs/concepts/overview/working-with-objects/labels。</p> <p>默认为空 LabelSelector，它与所有内容匹配。</p>
objectSelector	labelSelector	<p>objectSelector 决定是否根据对象是否有匹配的 labels。</p> <p>objectSelector 运行 webhook。objectSelector 是针对要发送到 webhook 的 oldObject 和 newObject 评估的，并被视为匹配对象是否与选择器匹配。null 对象（如果为 create，则为 newObject，则为 newObject）或没有标签（如 DeploymentRollback 或 PodProxyOptions 对象）的对象不被视为匹配。只有在 Webhook 是 opt-in 时，才使用对象选择器，因为最终用户可以通过设置标签来跳过准入 Webhook。默认为空 LabelSelector，它与所有内容匹配。</p>

属性	类型	描述
rules	array	规则描述了 Webhook 关注的 resources/subresources 的操作。如果与任何规则匹配，webhook 会处理某个操作。但是，为了防止 ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 使集群处于无法完全禁用插件的状态，ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 永远不会调用 ValidatingWebhookConfiguration 和 MutatingWebhookConfiguration 对象的准入请求。
rules[]	对象	RuleWithOperations 是操作和资源的元组。建议确保所有元组扩展都有效。
sideEffects	字符串	<p>SideEffects 指出此 webhook 是否产生副作用。可接受值为：None, NoneOnDryRun (webhooks via v1beta1 创建，也可以指定 some 或 Unknown)。具有副作用的 Webhook 实现了协调系统，因为准入链中的将来步骤可能会拒绝请求，因此需要撤销副作用。具有 dryRun 属性的请求如果与带有 sideEffects == Unknown 或 some 的 webhook 匹配，则会自动拒绝。</p> <p>可能的 enum 值：- "None" 表示调用 Webhook 不会有副作用。- "NoneOnDryRun" 表示调用 Webhook 可能会产生副作用，但如果检查了请求具有 dry-run 属性，则会隐藏副作用。- "Some" 表示调用 Webhook 可能会产生副作用。如果具有 dry-run 属性的请求会触发调用此 webhook，则请求将失败。- "Unknown" 表示没有有关调用 webhook 的副作用的信息。如果具有 dry-run 属性的请求会触发对此 webhook 的调用，则请求将失败。</p>

属性	类型	描述
timeoutSeconds	整数	timeoutSeconds 指定此 webhook 的超时时间。通过超时后，webhook 调用将被忽略，或者 API 调用将根据失败策略失败。超时值必须在 1 到 30 秒之间。默认值为 10 秒。

30.3.1.3. .webhooks[].clientConfig

描述

WebhookClientConfig 包含与 Webhook 进行 TLS 连接的信息

类型

object

属性	类型	描述
caBundle	字符串	cabundle 是一个 PEM 编码的 CA 捆绑包，用于验证 webhook 的服务器证书。如果未指定，则使用 apiserver 中的系统信任 root。
service	对象	ServiceReference 包含对 Service.legacy.k8s.io 的引用
url	字符串	URL 以标准 URL 格式提供 webhook 的位置 (scheme://host:port/path)。必须指定 url 或服务 之一。 主机 不应引用集群中运行的服务；改为使用 service 字段。在一些 apiservers（如 kube-apiserver）中，主机可以通过外部 DNS 解析，因为 kube-apiserver 无法解析集群 DNS，因为这将是分层违反情况。主机 也可以是 IP 地址。 请注意，使用 localhost 或 127.0.0.1 作为主机有风险，除非您特别注意在运行 apiserver 的所有主机上运行这

属性	类型	描述
		<p>个 webhook，这可能需要调用此 webhook。此类安装可能是不可移植的，例如，无法在新集群中轻松打开。</p> <p>方案必须是 "https"；URL 必须以 "https://" 开头。</p> <p>路径是可选的，如果存在，则 URL 中可能存在任何字符串。您可以使用路径将任意字符串传递给 Webhook，例如集群标识符。</p> <p>尝试使用用户或基本身份验证，例如不允许 "user:password@"。不允许片段("#...")和查询参数("?...")。</p>

30.3.1.4. .webhooks[].clientConfig.service

描述

ServiceReference 包含对 Service.legacy.k8s.io 的引用

类型

object

必填

- namespace
- name

属性	类型	描述
name	字符串	name 是服务的名称。必填

属性	类型	描述
namespace	字符串	namespace 是服务的命名空间。必填
path	字符串	path 是一个可选 URL 路径，将在此服务的任何请求中发送。
port	整数	如果指定，托管 Webhook 的服务上的端口。默认为 443，以便向后兼容。 端口 应该是有效的端口号 (1-65535，包含)。

30.3.1.5. .webhooks[].matchConditions

描述

MatchConditions 是必须满足的条件列表，才能将请求发送到此 webhook。匹配条件过滤请求，它们已与 rules、namespaceSelector 和 objectSelector 匹配。matchConditions 空列表匹配所有请求。允许最多有 64 个匹配条件。

完全匹配的逻辑是（按顺序排列）：1. 如果 ANYmatchCondition 评估为 FALSE，则会跳过 Webhook。2. 如果 ALLmatchConditions 评估为 TRUE，则调用 Webhook。3. 如果任何matchCondition 评估为错误（但没有为 FALSE）：- 如果 failurePolicy=Fail，则拒绝请求 - 如果 failurePolicy=Ignore，则忽略错误并跳过 Webhook

这是一个 alpha 功能，由 AdmissionWebhookMatchConditions 功能门管理。

类型

数组

30.3.1.6. .webhooks[].matchConditions[]

描述

MatchCondition 代表必须实现才能使请求发送到 webhook 的条件。

类型

object

必填

- **name**
- **expression**

属性	类型	描述
----	----	----

属性	类型	描述
expression	string	<p>表达式代表 CEL 将评估的表达式。必须评估 bool。CEL 表达式可以访问 AdmissionRequest 和 Authorizer 的内容，被组织为 CEL 变量：</p> <p>'object' - 传入请求中的对象。 DELETE 请求的值是 null。 'oldObject' - 现有对象。对于 CREATE requests，值为 null。 'request' - 准入请求的属性 (/pkg/apis/admission/types.go#AdmissionRequest). 'authorizer' - A CEL Authorizer。可用于对请求的主（用户或服务帐户）执行授权检查。请参阅 https://pkg.go.dev/k8s.io/apiserver/pkg/cel/library#Authz 'authorizer.requestResource' - 从 'authorizer' 构建的 CEL ResourceCheck，并使用 request 资源进行配置。CEL 的文档： https://kubernetes.io/docs/reference/using-api/cel/</p> <p>必需。</p>
name	string	<p>name 是此匹配条件的标识符，用于 MatchConditions 的策略合并，并提供用于日志目的的标识符。相关表达式应当有一个好的名称。name 必须是由字母数字字符、'-','_' 或 '.' 组成的合格名称，且必须以字母数字字符开头和结尾（例如：'MyName' 或 'my.name'，或 '123-abc'，regex is '([A-Za-z0-9]([A-Za-z0-9.]*)?[A-Za-z0-9])'），带有可选的 DNS 子域前缀和 '/'（如 'example.com/MyName'）</p> <p>必需。</p>

30.3.1.7. .webhooks[].rules

描述

规则描述了 Webhook 关注的 resources/subresources 的操作。如果与任何规则匹配，webhook 会处理某个操作。但是，为了防止 ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 使集群处于无法完全禁用插件的状态，ValidatingAdmissionWebhooks 和 MutatingAdmissionWebhooks 永远不会调用 ValidatingWebhookConfiguration 和 MutatingWebhookConfiguration 对象的准入请求。

类型

array

30.3.1.8. .webhooks[].rules[]

描述

RuleWithOperations 是操作和资源的元组。建议确保所有元组扩展都有效。

类型

object

属性	类型	描述
apiGroups	数组（字符串）	apiGroups 是资源所属的 API 组。"是所有组。如果 ' 存在，分片的长度必须为一。必需。
apiVersions	数组（字符串）	APIVersions 是资源所属的 API 版本。"是所有版本。如果 ' 存在，分片的长度必须为一。必需。
操作	数组（字符串）	操作是准入 hook 关注的操作 - CREATE、UPDATE、DELETE、CONNECT 或 *（对于所有这些操作）以及添加的任何未来准入操作。如果 '*' 存在，分片的长度必须为一。必需。
资源	数组（字符串）	resources 是一组这个规则应用到的资源列表。 例如：'pods' 表示 pod。 'pods/log' 表示 pod 的日志子资源。'表示所有资源，但不是子资源。'pod/' 表示所有 pod 的子资源。'/scale' 表示所有扩展子资源。'Burstable' 表示所有资源及其子资源。 如果存在通配符，验证规则将确保资源不会相互重叠。 根据括起对象，可能不允许子资源。必需。
scope	字符串	scope 指定此规则的范围。有效值为 "Cluster", "Namespaced", 和 "" "Cluster" 表示只有集群范围的资源与此规则匹配。命名空间 API 对象是集群范围的。 "namespaced" 意味着只有命名空间的资源与此规则匹配。"表示没有范围限制。子资源与其父资源的范围匹配。默认为 "*"。

30.3.2. API 端点

可用的 API 端点如下：

- **/apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations**
 - **DELETE** : 删除 ValidatingWebhookConfiguration 的集合
 - **GET**: 列出或监视类型为 ValidatingWebhookConfiguration 的对象
 - **POST**: 创建一个 ValidatingWebhookConfiguration
- **/apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations**
 - **GET**: 观察单个对 ValidatingWebhookConfiguration 列表的更改。已弃用：使用 'watch' 参数和 list 操作。
- **/apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}**
 - **DELETE** : 删除 ValidatingWebhookConfiguration
 - **GET**: 阅读指定的 ValidatingWebhookConfiguration
 - **PATCH**: 部分更新指定的 ValidatingWebhookConfiguration
 - **PUT** : 替换指定的 ValidatingWebhookConfiguration
- **/apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations/{name}**
 - **GET**: 观察对类型为 ValidatingWebhookConfiguration. deprecated 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

30.3.2.1. /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations

表 30.27. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ValidatingWebhookConfiguration 的集合

表 30.28. 查询参数

参数	类型	描述
----	----	----

参数	类型	描述
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
dryRun	字符串	<p>出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理</p>
fieldSelector	字符串	<p>用于按字段限制返回对象列表的选择器。默认为任何内容。</p>
gracePeriodSeconds	整数	<p>应该删除对象前的持续时间（以秒为单位）。值必须是非负整数。值零表示立即删除。如果这个值为 nil，则使用指定类型的默认宽限期。如果没有指定，则默认为每个对象值。零表示立即删除。</p>
labelSelector	字符串	<p>一个选择器，用于按标签限制返回的对象列表。默认为任何内容。</p>

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的值来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>deprecated：请使用 PropagationPolicy，此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false，则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy，但不能同时设置这两个字段。</p>
propagationPolicy	字符串	<p>是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为：'Orphan' - 孤立依赖项；'Background' - 允许垃圾收集器删除后台依赖的依赖项；'Foreground' - 一个级联策略，会删除前台所有依赖的级联策略。</p>
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。

表 30.29. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 30.30. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

列出或监视类型为 ValidatingWebhookConfiguration 的对象

表 30.31. 查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false，则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

表 30.32. HTTP 响应

HTTP 代码	响应正文
200 - OK	ValidatingWebhookConfigurationList 模式
401 - Unauthorized	空

HTTP 方法

POST

描述

创建 `ValidatingWebhookConfiguration`

表 30.33. 查询参数

参数	类型	描述
<code>dryRun</code>	字符串	出现时，表示不应保留修改。无效的或未被识别的 <code>dryRun</code> 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
<code>fieldManager</code>	字符串	<code>fieldmanager</code> 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
<code>fieldValidation</code>	string	<code>fieldValidation</code> 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore : 忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 <code>BadRequest</code> 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 30.34. 主体参数

参数	类型	描述
正文 (body)	ValidatingWebhookConfiguration 模式	

表 30.35. HTTP 响应

HTTP 代码	响应正文
200 - OK	ValidatingWebhookConfiguration 模式
201 - Created	ValidatingWebhookConfiguration 模式
202 - Accepted	ValidatingWebhookConfiguration 模式
401 - Unauthorized	空

30.3.2.2. /apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations

表 30.36. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。

参数	类型	描述
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>
pretty	字符串	如果为 'true'，则输出会经过 pretty print 处理。
resourceVersion	字符串	<p>resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions</p> <p>默认为未设置</p>
resourceVersionMatch	字符串	<p>resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch，详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions。</p> <p>默认为未设置</p>

参数	类型	描述
sendInitialEvents	布尔值	<p>'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下，监视流将以 synthetic 事件开头，以生成集合中对象的当前状态。发送完所有此类事件后，将发送一个合成"Bookmark"事件。书签将报告与一组对象对应的 ResourceVersion (RV)，并标有 "k8s.io/initial-events-end": "true" 注解。之后，监视流会照常进行，将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。</p> <p>当设置了 'sendInitialEvents' 选项时，我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下： - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时，书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion'，则会将其解释为 "consistent read"，并且当状态在请求开始处理时至少同步时，会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。</p> <p>如果 'resourceVersion="" 或 'resourceVersion="0"' (用于向后兼容的原因) 和 false，则默认为 true。</p>
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间，而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改，并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 ValidatingWebhookConfiguration 列表的更改。已弃用：使用 'watch' 参数和 list 操作。

表 30.37. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空

30.3.2.3. /apis/admissionregistration.k8s.io/v1/validatingwebhookconfigurations/{name}

表 30.38. 全局路径参数

参数	类型	描述
name	字符串	ValidatingWebhookConfiguration 的名称

表 30.39. 全局查询参数

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。

HTTP 方法

DELETE

描述

删除 ValidatingWebhookConfiguration

表 30.40. 查询参数

参数	类型	描述
dryRun	字符串	出现时, 表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应, 且请求不会被进一步处理。有效值为: - All: 所有预演阶段都将被处理
gracePeriodSeconds	整数	应该删除对象前的持续时间 (以秒为单位)。值必须是非负整数。值零表示立即删除。如果这个值为 nil, 则使用指定类型的默认宽限期。如果没有指定, 则默认为每个对象值。零表示立即删除。
orphanDependents	布尔值	deprecated : 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。依赖的对象应该被孤立。如果为 true/false, 则 "orphan" 终结器将从对象的终结器列表添加到/删除。可以设置此字段或 PropagationPolicy, 但不能同时设置这两个字段。
propagationPolicy	字符串	是否以及如何执行垃圾回收。可以设置此字段或 OrphanDependents, 但不能同时设置这两个字段。默认策略由 metadata.finalizers 和特定于资源的默认策略中设置的现有终结器决定。可接受值为: 'Orphan' - 孤立依赖项; 'Background' - 允许垃圾收集器删除后台依赖的依赖项; 'Foreground' - 一个级联策略, 会删除前台所有依赖的级联策略。

表 30.41. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 30.42. HTTP 响应

HTTP 代码	响应正文
200 - OK	Status 模式
202 - Accepted	Status 模式
401 - Unauthorized	空

HTTP 方法**GET****描述**

阅读指定的 ValidatingWebhookConfiguration

表 30.43. HTTP 响应

HTTP 代码	响应正文
200 - OK	ValidatingWebhookConfiguration 模式
401 - Unauthorized	空

HTTP 方法**PATCH****描述**

部分更新指定的 ValidatingWebhookConfiguration

表 30.44. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。应用请求需要此字段(application/apply-patch)，但对于非应用补丁类型(JsonPatch、MergePatch、strategicMergePatch)是可选的。

参数	类型	描述
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果从对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。
force	布尔值	强制尝试"强制"应用请求。这意味着用户将重新排序由其他人员拥有的冲突字段。对于非应用补丁请求，必须取消设置 force 标志。

表 30.45. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 30.46. HTTP 响应

HTTP 代码	响应正文
200 - OK	ValidatingWebhookConfiguration 模式
201 - Created	ValidatingWebhookConfiguration 模式
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 ValidatingWebhookConfiguration

表 30.47. 查询参数

参数	类型	描述
dryRun	字符串	出现时，表示不应保留修改。无效的或未被识别的 dryRun 指令将导致错误响应，且请求不会被进一步处理。有效值为：- All: 所有预演阶段都将被处理

参数	类型	描述
fieldManager	字符串	fieldmanager 是与进行这些更改的参与者或实体相关联的名称。该值必须小于或是 128 个字符长，且仅包含可打印的字符，如 https://golang.org/pkg/unicode/#IsPrint 所定义的那样。
fieldValidation	string	fieldValidation 指示服务器如何处理包含了未知或重复字段的请求 (POST/PUT/PATCH) 中的对象。有效值为：- Ignore：忽略从对象中静默丢弃的未知字段，并将忽略所有除最后一个重复字段以外的所有字段。这是 v1.23 版本之前的默认行为- Warn: 这会通过标准的警告响应头为从对象丢弃的每个未知字段以及遇到的每个重复字段，发送警告。如果没有其他错误，请求仍会成功，且只保留重复字段的最后一个。这是 v1.23+ 的默认行为 - Strict: 如果对对象中删除了任何未知字段，或者存在任何重复字段，请求将失败并带有 BadRequest 错误。从服务器返回的错误将包含遇到的，以及重复的字段。

表 30.48. 主体参数

参数	类型	描述
正文 (body)	ValidatingWebhookConfiguration 模式	

表 30.49. HTTP 响应

HTTP 代码	响应正文
200 - OK	ValidatingWebhookConfiguration 模式
201 - Created	ValidatingWebhookConfiguration 模式
401 - Unauthorized	空

30.3.2.4. /apis/admissionregistration.k8s.io/v1/watch/validatingwebhookconfigurations/{name}

表 30.50. 全局路径参数

参数	类型	描述
name	字符串	ValidatingWebhookConfiguration 的名称

表 30.51. 全局查询参数

参数	类型	描述
allowWatchBookmarks	布尔值	allowWatchBookmarks 请求类型为“BOOKMARK”的监视事件。没有实现书签的服务器可能会忽略这个标志和书签，由服务器自由裁量发送。客户端不应假设书签在任何特定间隔返回，也不会假定服务器在会话期间发送任何 BOOKMARK 事件。如果这不是监视，则忽略此字段。
继续	字符串	<p>从服务器检索更多结果时，应设置 continue 选项。由于这个值是定义的服务器，因此客户端只能使用之前查询结果中的 continue 值，并带有相同的查询参数（除 continue 值除外），因此服务器可能会拒绝它无法识别的值。如果指定的 continue 值不再有效，无论是因为过期时间（通常为 5 到十五分钟）还是服务器上的配置更改，服务器将会与继续令牌一起响应 410 ResourceExpired 错误。如果客户端需要一致的列表，则必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误的令牌的列表请求，服务器将使用从下一个密钥开始的列表进行响应，但从最新的快照开始，从上一个列表结果（创建、修改或删除）后，创建、修改或删除第一个列表请求将包含在响应中，只要它们的键位于“下一密钥”后。</p> <p>当监视为 true 时，不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不错过任何修改。</p>
fieldSelector	字符串	用于按字段限制返回对象列表的选择器。默认为任何内容。
labelSelector	字符串	一个选择器，用于按标签限制返回的对象列表。默认为任何内容。
limit	整数	<p>limit 是列表调用返回的最大响应数。如果存在更多项目，服务器会将列表元数据上的 'continue' 字段设置为可用于同一初始查询的值，以检索下一个结果集合。如果过滤了所有请求的对象，设置限制可能会小于请求的项目数量（最多零项），并且客户端应该只使用 continue 字段的存在来确定是否有可用的结果。服务器可能选择不支持 limit 参数，并将返回所有可用结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假设没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用 continue 时返回的对象与在没有限制的情况下发出单个列表调用时返回的对象将相同，即在发出第一个请求后没有创建、修改或删除的对象。这有时被称为一致的快照，并确保使用限制的客户端接收大量结果的较小的块可以确保它们可以看到所有可能的对象。如果在块列表列表期间更新对象，则返回第一个列表结果时存在的对象版本。</p>

参数	类型	描述
pretty	字符串	如果为 'true', 则输出会经过 pretty print 处理。
resourceVersion	字符串	resourceVersion 对可能给请求提供服务的资源版本设置一个约束。详情请查看 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 默认为未设置
resourceVersionMatch	字符串	resourceVersionMatch 决定 resourceVersion 如何应用到列表调用。强烈建议您为设置 resourceVersion 的列表调用设置 resourceVersionMatch, 详情请参阅 https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions 。 默认为未设置
sendInitialEvents	布尔值	'sendInitialEvents=true' 可以与 'watch=true' 一起设置。在这种情况下, 监视流将以 synthetic 事件开头, 以生成集合中对象的当前状态。发送完所有此类事件后, 将发送一个合成 "Bookmark" 事件。书签将报告与一组对象对应的 ResourceVersion (RV), 并标有 "k8s.io/initial-events-end": "true" 注解。之后, 监视流会照常进行, 将与更改对应的监控事件(subsequent to the RV)发送到被监视的对象。 当设置了 'sendInitialEvents' 选项时, 我们还需要设置 'resourceVersionMatch' 选项。监视请求的语义如下: - 'resourceVersionMatch' = NotOlderThan 解释为 "data at the provided as the provided as the provided 'resourceVersion'", 当状态同步到 'resourceVersion' 时, 书签事件至少作为 ListOptions 提供的 "resourceVersion" 提供。如果未设置 'resourceVersion', 则会将其解释为 "consistent read", 并且当状态在请求开始处理时至少同步时, 会发送书签事件。- 'resourceVersionMatch' 设置为任何其他值或取消设置 Invalid 错误。 如果 'resourceVersion="" 或 'resourceVersion="0" (用于向后兼容的原因) 和 false, 则默认为 true。
timeoutSeconds	整数	list/watch 调用的超时。这限制了调用的持续时间, 而不考虑任何活动或不活跃。
watch	布尔值	观察对上述资源的更改, 并将其恢复为添加、更新和删除通知的流。指定 resourceVersion。

HTTP 方法

GET

描述

观察对 kind ValidatingWebhookConfiguration. deprecated 的对象更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤到带有 'fieldSelector' 参数的单个项目。

表 30.52. HTTP 响应

HTTP 代码	响应正文
200 - OK	WatchEvent 模式
401 - Unauthorized	空