



OpenShift Container Platform 4.14

模板 API

模板 API 的参考指南

法律通告

Copyright © 2025 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

本文档描述了 OpenShift Container Platform 模板 API 对象及其详细规格。

目录

第 1 章 模板 API	3
1.1. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	3
1.2. PODTEMPLATE [V1]	3
1.3. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]	3
1.4. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	3
第 2 章 BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	4
2.1. 规格	4
2.2. API 端点	5
第 3 章 PODTEMPLATE [V1]	22
3.1. 规格	22
3.2. API 端点	222
第 4 章 TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]	245
4.1. 规格	245
4.2. API 端点	247
第 5 章 TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	271
5.1. 规格	271
5.2. API 端点	277

第 1 章 模板 API

1.1. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

描述

BrokerTemplateInstance 包含与 TemplateInstance 关联的服务代理相关状态。

BrokerTemplateInstance 是实验性 API 的一部分。

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

1.2. PODTEMPLATE [V1]

描述

PodTemplate 描述了用于创建预定义 pod 副本的模板。

类型

object

1.3. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]

描述

模板包含生成配置所需的输入。

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

1.4. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

描述

TemplateInstance 请求并记录模板的实例化。TemplateInstance 是实验性 API 的一部分。

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

第 2 章 BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

描述

BrokerTemplateInstance 包含与 TemplateInstance 关联的服务代理相关状态。

BrokerTemplateInstance 是实验性 API 的一部分。

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**

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	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	object	BrokerTemplateInstanceSpec 描述了 BrokerTemplateInstance 的状态。

2.1.1. .spec

描述

BrokerTemplateInstanceSpec 描述了 BrokerTemplateInstance 的状态。

类型

object

必填

- **templateInstance**
- **secret**

属性	类型	描述
bindingIDs	数组（字符串）	bindingids 是对模板服务代理连续绑定调用过程中提供的 'binding_id' 列表。
secret	ObjectReference	Secret 是对位于命名空间中的 Secret 对象的引用，其中包含必要的模板参数。
templateInstance	ObjectReference	TemplateInstance 是驻留于命名空间中的 TemplateInstance 对象的引用。

2.2. API 端点

可用的 API 端点如下：

- **/apis/template.openshift.io/v1/brokertemplateinstances**
 - **DELETE:** 删除 BrokerTemplateInstance 集合
 - **GET:** list 或 watch kind BrokerTemplateInstance 的对象
 - **POST :** 创建 BrokerTemplateInstance
- **/apis/template.openshift.io/v1/watch/brokertemplateinstances**
 - **GET:** 观察单个对 BrokerTemplateInstance 列表的更改。已弃用：改为使用 'watch' 参数及列表操作。
- **/apis/template.openshift.io/v1/brokertemplateinstances/{name}**
 - **DELETE:** 删除 BrokerTemplateInstance
 - **GET :** 读取指定的 BrokerTemplateInstance
 - **PATCH:** 部分更新指定的 BrokerTemplateInstance
 - **PUT :** 替换指定的 BrokerTemplateInstance

- `/apis/template.openshift.io/v1/watch/brokertemplateinstances/{name}`
 - **GET**: 观察对类型为 `BrokerTemplateInstance` 的对象的更改。已弃用：使用带列表操作的 `'watch'` 参数，而是过滤成带有 `'fieldSelector'` 参数的单个项目。

2.2.1. /apis/template.openshift.io/v1/brokertemplateinstances

表 2.1. 全局查询参数

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

HTTP 方法

DELETE

描述

删除 `BrokerTemplateInstance` 集合

表 2.2. 查询参数

参数	类型	描述
<code>继续</code>	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 <code>410 ResourceExpired</code> 错误。如果客户端需要一致的列表，它必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 <code>410</code> 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 <code>"next key"</code> 后。</p> <p>当 <code>watch</code> 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，而不丢失任何修改。</p>
<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：- <code>All</code> : 所有预演阶段都将被处理
<code>fieldSelector</code>	字符串	一个选择器，用于限制其字段返回的对象列表。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。值为零表示立即删除。如果这个值是 nil，则会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，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。
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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。

表 2.3. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 2.4. 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 BrokerTemplateInstance 的对象

表 2.5. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

表 2.6. HTTP 响应

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

HTTP 方法**POST****描述**

创建 BrokerTemplateInstance

表 2.7. 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 2.8. 主体参数

参数	类型	描述
正文 (body)	BrokerTemplateInstance schema	

表 2.9. HTTP 响应

HTTP 代码	响应正文
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
202 - Accepted	BrokerTemplateInstance schema
401 - Unauthorized	空

2.2.2. /apis/template.openshift.io/v1/watch/brokertemplateinstances

表 2.10. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下, 观察流将以复合事件开头, 以生成集合中对象的当前状态。发送完所有这些事件后, 将发送合成 "Bookmark" 事件。书签将报告与对象集合对应的 ResourceVersion (RV), 并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后, 监视流将照常进行, 从而向监视的对象发送与更改对应的监视事件 (按 RV 排序)。 当设置 sendInitialEvents 选项时, 我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下: - resourceVersionMatch = NotOlderThan 解释为 "数据至少为提供的 resourceVersion ", 当状态同步到 ' resourceVersion ' 时, 如果状态同步到 ListOptions 提供的 ' resourceVersion ' 时, 会发送书签事件。如果未设置 resourceVersion , 则这被解释为 "consistent read", 当请求开始处理时, 将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" (出于向后兼容原因) 和 false, 则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间, 无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改, 并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

HTTP 方法

GET

描述

观察单个对 BrokerTemplateInstance 列表的更改。已弃用：使用带列表操作的 'watch' 参数。

表 2.11. HTTP 响应

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

2.2.3. /apis/template.openshift.io/v1/brokertemplateinstances/{name}

表 2.12. 全局路径参数

参数	类型	描述
name	string	BrokerTemplateInstance 的名称

表 2.13. 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 BrokerTemplateInstance

表 2.14. 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。值为零表示立即删除。如果这个值是 nil, 则会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	弃用: 请使用 PropagationPolicy, 此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false, 则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy, 但不能同时设置。

参数	类型	描述
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。

表 2.15. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 2.16. 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 BrokerTemplateInstance

表 2.17. HTTP 响应

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

HTTP 方法

PATCH

描述

部分更新指定的 BrokerTemplateInstance

表 2.18. 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(application/apply-patch)需要此字段，但对非应用补丁类型(JsonPatch、MergePatch、MergePatch)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	强制要"强制"应用请求。这意味着用户将重新分配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 2.19. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 2.20. HTTP 响应

HTTP 代码	响应正文
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
401 - Unauthorized	空

HTTP 方法

PUT

描述

替换指定的 BrokerTemplateInstance

表 2.21. 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 2.22. 主体参数

参数	类型	描述
正文 (body)	BrokerTemplateInstance schema	

表 2.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
401 - Unauthorized	空

2.2.4. /apis/template.openshift.io/v1/watch/brokertemplateinstances/{name}

表 2.24. 全局路径参数

参数	类型	描述
name	string	BrokerTemplateInstance 的名称

表 2.25. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

观察对类型为 `BrokerTemplateInstance` 的对象的更改。已弃用：使用带有 `list` 操作的 `'watch'` 参数，而是过滤成带有 `'fieldSelector'` 参数的单个项目。

表 2.26. HTTP 响应

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

第 3 章 PODTEMPLATE [V1]

描述

PodTemplate 描述了用于创建预定义 pod 副本的模板。

类型

对象

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
模板	object	PodTemplateSpec 描述了从模板创建 pod 应具有的数据

3.1.1. .template

描述

PodTemplateSpec 描述了从模板创建 pod 应具有的数据

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。

3.1.2. .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	可选持续时间（以秒为单位） pod 可以在相对于 StartTime 的节点上处于活跃状态，然后系统主动尝试将其标记为失败并终止关联的容器。值必须是正整数。
关联性	object	关联性是一组关联性调度规则。
automountServiceAccountToken	布尔值	AutomountServiceAccountToken 表示是否应自动挂载服务帐户令牌。
containers	array	属于 pod 的容器列表。当前无法添加或删除容器。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 除了从 DNSPolicy 生成的外，还定义了 pod 的 DNS 参数。

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 中指定的 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 应该首先使用集群 DNS，除非 hostNetwork 为 true，如果可用，则回退到默认值（由 kubelet 确定）。 - "ClusterFirstWithHostNet" 表示 pod 应首先使用集群 DNS，如果可用，则回退到默认值（由 kubelet 确定）。 kubelet) DNS 设置。 - "Default" 表示 pod 应使用默认值（由 kubelet 决定） DNS 设置。 - "None" 表示 pod 应使用空的 DNS 设置。应该通过 DNSConfig 定义 DNS 参数（如名称服务器和搜索路径）。</p>
enableServiceLinks	布尔值	<p>EnableServiceLinks 指明了有关服务的信息是否应注入到 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，且无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod，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，您可以添加到现有 Pod 中，用于用户发起的活动，如调试。临时容器没有资源或调度保证，它们不会在 Pod 退出或重启 Pod 时重启。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。临时容器可能无法删除或重启。</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。这仅对非主机网络 pod 有效。
hostAliases[]	对象	HostAlias 包含 IP 和主机名之间的映射，这些映射将作为 pod 的主机文件中的条目注入。
hostIPC	布尔值	使用主机的 ipc 命名空间。可选：默认为 false。
hostNetwork	布尔值	为这个 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。默认值为 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 需要功能仅适用于主机用户命名空间时，比如使用 CAP_SYS_MODULE 加载内核模块。当设置为 false 时，会为 pod 创建一个新的 usersns。设置 false 有助于减少容器中漏洞，甚至允许用户以 root 用户身份运行其容器，而无需实际对主机具有 root 特权。此字段是 alpha-level，且仅被启用 UserNamespacesSupport 功能的服务器实现。</p>

属性	类型	描述
hostname	字符串	指定没有指定 Pod 的主机名，pod 的主机名将设置为系统定义的值。
imagePullSecrets	array	imagePullSecrets 是对同一命名空间中的 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 包含足够信息，供您在同一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 容器在启动容器之前按顺序执行。如果任何 init 容器失败，pod 被视为失败，并根据 restartPolicy 处理。init 容器或普通容器的名称对于所有容器来说都必须是唯一的。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。通过查找每种资源类型的最高请求/限制，然后使用该值的最大值或普通容器的总和，在调度期间考虑 init 容器的 resourceRequirements。限制以类似的方式应用到 init 容器。当前无法添加或删除 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。如果不是空的，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 适合某个节点。选择器必须与节点的标签匹配，以便 pod 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。RuntimeClass 准入控制器将在准入时自动填充此字段。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 准入控制器将拒绝已设置开销的 Pod 创建请求。如果在 PodSpec 中配置并选择 RuntimeClass，则 Overhead 将被设置为相应 RuntimeClass 中定义的值，否则它将保留为未设置并被视零。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 pod 的策略。Never 之一 PreemptLowerPriority。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 中填充此字段。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，代表前一个优先级最高。必须通过创建具有该名称的 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 就绪评估所有就绪度。当所有容器都就绪且在就绪的 gates 中指定的所有条件都等于 "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates 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 Pod 允许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。 此字段不可变。
resourceClaims[]	object	PodResourceClaim 通过 ClaimSource 只引用一个 ResourceClaim。它添加一个名称，用于唯一标识 Pod 中的 ResourceClaim。需要访问 ResourceClaim 的容器使用此名称来引用它。

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 之一 Never。在某些上下文中，只允许这些值的子集。默认为 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，该对象应该用于运行此 pod。如果没有与命名类匹配的 RuntimeClass 资源，则不会运行 pod。如果未设置或为空，则使用"传统" RuntimeClass，这是一个具有使用默认运行时处理程序的空定义的隐式类。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 会被默认调度程序分配。</p>
schedulingGates	数组	<p>SchedulingGates 是不透明的值列表，如果指定，它将阻止调度 pod。如果 schedulingGates 不为空，pod 将保持在 SchedulingGated 状态，调度程序不会尝试调度 pod。</p> <p>SchedulingGates 只能在 pod 创建时设置，之后才能被删除。</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 的描述别名。弃用：使用 serviceAccountName 替代。
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 hostname 的 registry 值设置为 FQDN。如果 pod 没有 FQDN，则这无效。默认值为 false。
shareProcessNamespace	布尔值	在 pod 中所有容器间共享单个进程命名空间。当设置容器时，可以从同一 pod 中的其他容器查看和信号进程，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。

属性	类型	描述
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 将根本没有 domainname。
terminationGracePeriodSeconds	整数	pod 安全终止所需的可选持续时间（以秒为单位）。可以在删除请求中减少。值必须是非负整数。值为零表示立即通过终止信号停止（没有关闭的机会）。如果这个值是 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和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。默认值为 30 秒。
容限 (tolerations)	array	如果指定，pod 的容限。
tolerations[]	对象	此 Tolerantion 附加到 pod，以使用匹配的 operator <operator> 容许与 triple <key,value,effect> 匹配的任何污点。
topologySpreadConstraints	array	topologySpreadConstraints 描述了一组如何在拓扑域间分布的 pod。调度程序将按约束来调度 pod。所有 topologySpreadConstraints 都是 AND。
topologySpreadConstraints[]	对象	TopologySpreadConstraint 指定如何在给定的拓扑中分布匹配的 pod。
卷	array	可以通过属于 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 pod 中的指定卷，可以被 pod 中的任何容器访问。

3.1.3. .template.spec.affinity

描述

关联性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。

3.1.4. .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	调度程序更喜欢将 pod 调度到满足此字段指定的关联性表达式的节点，但可能会选择违反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、必需关联表达式等）的每个节点，计算出一个总和总和总和总和（如果节点与相应的 matchExpressions 匹配），则计算总和最高的节点是最首选的。
preferredDuringSchedulingIgnoredDuringExecution[]	对象	空白的首选调度术语与带有隐式权重 0（例如，它的一个 no-op）的所有对象匹配。空首选调度术语不匹配任何对象（例如，也是 no-op）。
requiredDuringSchedulingIgnoredDuringExecution	object	节点选择器代表一组节点上一个或多个标签查询的结果的 union；即，它代表节点选择器术语所代表的选择器或选择器。

3.1.5. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

描述

调度程序更喜欢将 pod 调度到满足此字段指定的关联性表达式的节点，但可能会选择违反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、必需关联表达式等）的每个节点，计算出一个总和总和总和总和（如果节点与相应的 matchExpressions 匹配），则计算总和最高的节点是最首选的。

类型

数组

3.1.6. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE**描述**

空白的首选调度术语与带有隐式权重 0（例如，它的一个 no-op）的所有对象匹配。空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 或空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 nodeSelectorTerm 关联的权重，范围为 1-100。

3.1.7. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE**描述**

null 或空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	按节点标签划分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，其中包含与键和值相关的值、键和值。

属性	类型	描述
matchFields	array	按节点字段划分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，其中包含与键和值相关的值、键和值。

3.1.8. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

描述

按节点标签划分的节点选择器要求列表。

类型

数组

3.1.9. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

描述

节点选择器要求是一个选择器，其中包含与键和值相关的值、键和值。

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。 可能枚举值： - "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。如果运算符是 Gt 或 Lt，则值数组必须具有单个元素，它将解释为整数。这个数组会在策略性合并补丁中被替换。

3.1.10. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

描述

按节点字段划分的节点选择器要求列表。

类型

数组

3.1.11. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

描述

节点选择器要求是一个选择器，其中包含与键和值相关的值、键和值。

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。 可能枚举值： - "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。如果运算符是 Gt 或 Lt，则值数组必须具有单个元素，它将解释为整数。这个数组会在策略性合并补丁中被替换。

3.1.12. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

节点选择器代表一组节点上一个或多个标签查询的结果的 union；即，它代表节点选择器术语所代表的选择器或选择器。

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。术语为 ORed。
nodeSelectorTerms[]	对象	null 或空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。

3.1.13. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

必需。节点选择器术语列表。术语为 ORed。

类型

数组

3.1.14. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

null 或空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	按节点标签划分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，其中包含与键和价值相关的值、键和价值。
matchFields	array	按节点字段划分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，其中包含与键和价值相关的值、键和价值。

3.1.15. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

按节点标签划分的节点选择器要求列表。

类型

数组

3.1.16. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

节点选择器要求是一个选择器，其中包含与键和值相关的值、键和值。

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。 可能枚举值：- "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。如果运算符是 Gt 或 Lt，则值数组必须具有单个元素，它将解释为整数。这个数组会在策略性合并补丁中被替换。

3.1.17. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

按节点字段划分的节点选择器要求列表。

类型

数组

3.1.18. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

描述

节点选择器要求是一个选择器，其中包含与键和值相关的值、键和值。

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。 可能枚举值： - "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。如果运算符是 Gt 或 Lt，则值数组必须具有单个元素，它将解释为整数。这个数组会在策略性合并补丁中被替换。

3.1.19. .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
----	----	----

属性	类型	描述
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	调度程序更喜欢将 pod 调度到满足此字段指定的关联性表达式的节点，但可能会选择违反一个或多个表达式的节点。最首选的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的节点 (resource request、requiredDuringScheduling 关联性表达式等)的每个节点，计算并迭代此字段元素，并在节点具有与对应 podAffinityTerm 匹配的 pod 时，将"weight"添加到总和（如果节点具有最高 podAffinityTerm）、具有最高总和最高的节点是首选。
<code>preferredDuringSchedulingIgnoredDuringExecution[]</code>	对象	所有匹配的 WeightedPodAffinityTerm 字段的权重添加到每个节点，以查找最首选节点。
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时没有满足此字段指定的关联性要求，则 pod 不会被调度到该节点上。如果此字段指定的关联性要求在 pod 执行期间某一点满足（例如，由于 pod 标签更新），则系统可能或可能无法最终从其节点驱除 pod。当有多个元素时，与每个 podAffinityTerm 对应的节点列表都会被交集，例如必须满足所有术语。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

3.1.20. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringE:

描述

调度程序更喜欢将 pod 调度到满足此字段指定的关联性表达式的节点，但可能会选择违反一个或多个表达式的节点。最首选的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的节点 (resource request、requiredDuringScheduling 关联性表达式等)的每个节点，计算并迭代此字段元

素，并在节点具有与对应 podAffinityTerm 匹配的 pod 时，将"weight"添加到总和（如果节点具有最高 podAffinityTerm）、具有最高总和和最高的节点是首选。

类型

数组

3.1.21. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

描述

所有匹配的 WeightedPodAffinityTerm 字段的权重添加到每个节点，以查找最首选节点。

类型

object

必填

- weight
- podAffinityTerm

属性	类型	描述
podAffinityTerm	object	定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。
weight	整数	与对应的 podAffinityTerm 关联的权重，范围为 1-100。

3.1.22. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

描述

定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

类型

object

必填

- topologyKey

属性	类型	描述
----	----	----

属性	类型	描述
labelSelector	LabelSelector	对一组资源进行标签查询，本例中为 pod。
namespaceSelector	LabelSelector	术语适用的命名空间集合上的标签查询。术语应用于此字段选定的命名空间的 union，以及 namespaces 字段中列出的命名空间。null selector 和 null 或空命名空间列表表示"此 pod 的命名空间"。空选择器({})匹配所有命名空间。
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。术语应用到此字段中列出的命名空间的 union，namespaceSelector 选择的命名空间列表和 null namespaceSelector 表示 " this pod 的命名空间"。
topologyKey	字符串	此 pod 应该在指定命名空间中与 labelSelector 匹配的 pod 共存（反关联性），其中 co-located 定义为在运行所选 pod 的节点的值 running 的节点上运行。不允许为空 topologyKey。

3.1.23. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

描述

如果在调度时没有满足此字段指定的关联性要求，则 pod 不会被调度到该节点上。如果此字段指定的关联性要求在 pod 执行期间某一点满足（例如，由于 pod 标签更新），则系统可能或可能无法最终从其节点驱除 pod。当有多个元素时，与每个 podAffinityTerm 对应的节点列表都会被交集，例如必须满足所有术语。

类型

数组

3.1.24. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

描述

定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	LabelSelector	对一组资源进行标签查询，本例中为 pod。
namespaceSelector	LabelSelector	术语适用的命名空间集合上的标签查询。术语应用于此字段选定的命名空间的 union，以及 namespaces 字段中列出的命名空间。null selector 和 null 或空命名空间列表表示"此 pod 的命名空间"。空选择器({})匹配所有命名空间。
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。术语应用到此字段中列出的命名空间的 union，namespaceSelector 选择的命名空间列表和 null namespaceSelector 表示 " this pod 的命名空间"。
topologyKey	字符串	此 pod 应该在指定命名空间中与 labelSelector 匹配的 pod 共存（反关联性），其中 co-located 定义为在运行所选 pod 的节点的值 running 的节点上运行。不允许为空 topologyKey。

3.1.25. .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	调度程序更喜欢将 pod 调度到满足此字段指定的反关联性表达式的节点，但可能会选择违反一个或多个表达式的节点。最首选的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，计算通过此字段元素来计算总和总和（如果节点具有与对应的 podAffinityTerm 匹配的 pod）的总和。

属性	类型	描述
<code>preferredDuringSchedulingIgnoredDuringExecution[]</code>	对象	所有匹配的 <code>WeightedPodAffinityTerm</code> 字段的权重添加到每个节点，以查找最首选节点。
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	array	如果在调度时没有满足此字段指定的反关联性要求，则 pod 不会被调度到该节点上。如果此字段指定的反关联性要求在 Pod 执行期间某一点满足（例如，由于 pod 标签更新），则系统可能或可能无法最终从其节点驱除 pod。当有多个元素时，与每个 <code>podAffinityTerm</code> 对应的节点列表都会被交集，例如必须满足所有术语。
<code>requiredDuringSchedulingIgnoredDuringExecution[]</code>	对象	定义一组 pod（即，与给定命名空间相对 <code>labelSelector</code> 匹配的 <code>labelSelector</code> ），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 <code>co-located</code> 在具有键 <code><topologyKey></code> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

3.1.26. `.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

调度程序更喜欢将 pod 调度到满足此字段指定的反关联性表达式的节点，但可能会选择违反一个或多个表达式的节点。最首选的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，计算通过此字段元素来计算总和总和（如果节点具有与对应的 `podAffinityTerm` 匹配的 pod）的总和。

类型

数组

3.1.27. `.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

描述

所有匹配的 `WeightedPodAffinityTerm` 字段的权重添加到每个节点，以查找最首选节点。

类型

object

必填

- `weight`
- `podAffinityTerm`

属性	类型	描述
podAffinityTerm	object	定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。
weight	整数	与对应的 podAffinityTerm 关联的权重，范围为 1-100。

3.1.28. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

描述

定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

类型

object

必填

- **topologyKey**

属性	类型	描述
labelSelector	LabelSelector	对一组资源进行标签查询，本例中为 pod。
namespaceSelector	LabelSelector	术语适用的命名空间集合上的标签查询。术语应用于此字段选定的命名空间的 union，以及 namespaces 字段中列出的命名空间。null selector 和 null 或空命名空间列表表示"此 pod 的命名空间"。空选择器({})匹配所有命名空间。
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。术语应用到此字段中列出的命名空间的 union，namespaceSelector 选择的命名空间列表和 null namespaceSelector 表示 " this pod 的命名空间"。

属性	类型	描述
topologyKey	字符串	此 pod 应该在指定命名空间中与 labelSelector 匹配的 pod 共存（反关联性），其中 co-located 定义为在运行所选 pod 的节点的值 running 的节点上运行。不允许为空 topologyKey。

3.1.29. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDurin

描述

如果在调度时没有满足此字段指定的反关联性要求，则 pod 不会被调度到该节点上。如果此字段指定的反关联性要求在 Pod 执行期间某一点满足（例如，由于 pod 标签更新），则系统可能或可能无法最终从其节点驱除 pod。当有多个元素时，与每个 podAffinityTerm 对应的节点列表都会被交集，例如必须满足所有术语。

类型

数组

3.1.30. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDurin

描述

定义一组 pod（即，与给定命名空间相对 labelSelector 匹配的 labelSelector），该 pod 应该与 pod 共同定位（反关联性）或没有共同定位（反关联性）的节点上运行，其中 co-located 在具有键 <topologyKey> 标签的节点上运行，它与其上运行一组 pod 的 pod 匹配。

类型

object

必填

- topologyKey

属性	类型	描述
labelSelector	LabelSelector	对一组资源进行标签查询，本例中为 pod。
namespaceSelector	LabelSelector	术语适用的命名空间集合上的标签查询。术语应用于此字段选定的命名空间的 union，以及 namespaces 字段中列出的命名空间。null selector 和 null 或空命名空间列表表示"此 pod 的命名空间"。空选择器({})匹配所有命名空间。

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。术语应用到此字段中列出的命名空间的 union, namespaceSelector 选择的命名空间列表和 null namespaceSelector 表示 " this pod 的命名空间"。
topologyKey	字符串	此 pod 应该在指定命名空间中与 labelSelector 匹配的 pod 共存（反关联性），其中 co-located 定义为在运行所选 pod 的节点的价值为 running 的节点上运行。不允许为空 topologyKey。

3.1.31. .template.spec.containers

描述

属于 pod 的容器列表。当前无法添加或删除容器。Pod 中必须至少有一个容器。无法更新。

类型

数组

3.1.32. .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 \$(VAR_NAME) 使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 \$，它允许转义 \$(VAR_NAME) 语法：即 "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会扩展，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。没有在 shell 中执行。如果未提供此容器镜像，则使用容器镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME) 使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 \$，它允许转义 \$(VAR_NAME) 语法：即 "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会扩展，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	容器中要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	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（如 Deployments 和 StatefulSets）中的容器镜像。
imagePullPolicy	字符串	镜像拉取(pull)策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则则默认为 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 始终尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示如果镜像在磁盘上不存在，kubelet 会拉取。如果镜像不存在且拉取失败，则容器将失败。- "Never" 表示 kubelet 永不拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
lifecycle	object	lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序，容器块的管理直到操作完成，除非容器进程失败，在这种情况下，处理程序会被中止。
livenessProbe	object	probe 描述要针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。
名称	字符串	指定为 DNS_LABEL 的容器名称。pod 中的每个容器都必须具有唯一的名称(DNS_LABEL)。无法更新。

属性	类型	描述
ports	array	从容器公开的端口列表。这里没有指定端口不会阻止公开该端口。所有正在侦听容器内默认"0.0.0.0"地址的端口均可从网络访问。使用策略合并补丁修改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 代表单个容器中的网络端口。
readinessProbe	object	probe 描述要针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。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源重新定义策略。
resources	对象	ResourceRequirements 描述了计算资源要求。
securityContext	object	securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时，SecurityContext 中的值将具有优先权。
startupProbe	object	probe 描述要针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	此容器是否应该在容器运行时中为 stdin 分配缓冲区。如果没有设置，则容器中的 stdin 读取将始终会导致 EOF。默认为 false。

属性	类型	描述
stdinOnce	布尔值	容器运行时是否应在由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中保持打开状态。如果 stdinOnce 设为 true，则 stdin 会在容器启动时打开，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，此时 stdin 已关闭并保持关闭，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。默认为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件的路径挂载到容器的文件系统中。编写的消息应当是最终状态，如断言失败消息。如果超过 4096 字节，节点将截断。所有容器的总消息长度将限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明应当如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功或失败时填充容器状态消息。如果终止消息文件为空，则 FallbackToLogsOnError 将使用容器日志输出的最后块，并且容器退出并显示错误。日志输出限制为 2048 字节或 80 行，以较小者。默认为 File。无法更新。 可能枚举值：- "FallbackToLogsOnError" 将在容器退出时读取容器状态消息的最新内容，并且 terminationMessagePath 没有任何内容。- "File" 是默认行为，并在容器退出时将容器状态消息设置为容器的 terminationMessagePath 的内容。
tty	布尔值	此容器是否应为自己分配 TTY，也要求 'stdin' 为 true。默认为 false。
volumeDevices	array	volumeDevices 是容器要使用的块设备列表。

属性	类型	描述
volumeDevices[]	对象	volumeDevice 描述了容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统的 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了容器内卷挂载。
workingDir	字符串	容器的工作目录。如果未指定，则将使用容器运行时的默认值，该默认值可能在容器镜像中配置。无法更新。

3.1.33. .template.spec.containers[].env

描述

容器中要设置的环境变量列表。无法更新。

类型

数组

3.1.34. .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。

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以及任何服务环境变量进行扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 $\$$ ，它允许转义 $\$(VAR_NAME)$ 语法：即 $"(VAR_NAME)"$ 将生成字符串 literal $\$(VAR_NAME)$ 。转义的引用永远不会扩展，无论变量是否存在。默认为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的值的源。

3.1.35. .template.spec.containers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的值的源。

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。

3.1.36. .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

3.1.37. .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	模式的版本是按术语编写的 FieldPath，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

3.1.38. .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：卷需要，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

3.1.39. .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 还是其密钥

3.1.40. .template.spec.containers[].envFrom

描述

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

3.1.41. .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 字段的内容将作为环境变量表示键值对。
prefix	字符串	要添加到 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 Data 字段的内容将作为环境变量表示键值对。

3.1.42. .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 字段的内容将作为环境变量表示键值对。

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

3.1.43. .template.spec.containers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的 Data 字段的内容将作为环境变量表示键值对。

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
optional	布尔值	指定是否必须定义 Secret

3.1.44. .template.spec.containers[].lifecycle

描述

lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序，容器块的管理直到操作完成，除非容器进程失败，在这种情况下，处理程序会被中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。
preStop	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

3.1.45. .template.spec.containers[].lifecycle.postStart

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

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 Get 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

3.1.46. .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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.47. .template.spec.containers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值：- "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.48. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.49. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.50. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.51. .template.spec.containers[].lifecycle.preStop

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

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 Get 请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

3.1.52. .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
----	----	----

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，而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, 它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.53. .template.spec.containers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.54. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.55. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.56. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.57. .template.spec.containers[].livenessProbe

描述

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.58. .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，而命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' '等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.59. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.60. .template.spec.containers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值：- "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.61. .template.spec.containers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.62. .template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.63. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.64. .template.spec.containers[].ports

描述

从容器公开的端口列表。这里没有指定端口不会阻止公开该端口。所有正在侦听容器内默认"0.0.0.0"地址的端口均可从网络访问。使用策略合并补丁修改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

3.1.65. .template.spec.containers[].ports[]

描述

containerPort 代表单个容器中的网络端口。

类型

object

必填

- containerPort

属性	类型	描述
containerPort	整数	pod 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。

属性	类型	描述
hostPort	整数	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，它必须与 ContainerPort 匹配。大多数容器都不需要这样做。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，在 pod 中唯一。pod 中的每个命名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"。 可能枚举值：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

3.1.66. .template.spec.containers[].readinessProbe

描述

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。
initialDelaySeconds	整数	容器启动存活度探测前的秒数。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

属性	类型	描述
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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.67. .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
----	----	----

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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.68. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.69. .template.spec.containers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.70. .template.spec.containers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.71. .template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.72. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.73. .template.spec.containers[].resizePolicy

描述

容器的资源重新定义策略。

类型

数组

3.1.74. .template.spec.containers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源重新定义策略。

类型

object

必填

- **resourceName**

- **restartPolicy**

属性	类型	描述
resourceName	string	此资源重新定义策略应用到的资源的名称。支持的值有：cpu、memory。
restartPolicy	string	在指定资源调整大小时，重启策略会被应用。如果没有指定，则默认为 NotRequired。

3.1.75. .template.spec.containers[].resources

描述

ResourceRequirements 描述了计算资源要求。

类型

object

属性	类型	描述
声明	数组	claim 列出此容器使用的 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。 此字段不可变。它只能为容器设置。
claims[]	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/

属性	类型	描述
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/

3.1.76. .template.spec.containers[].resources.claims

描述

claim 列出此容器使用的 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。

此字段不可变。它只能为容器设置。

类型

数组

3.1.77. .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
name	string	name 必须与使用此字段的 Pod 的 pod.spec.resourceClaims 中的一个条目的名称匹配。它允许容器内部使用该资源。

3.1.78. .template.spec.containers[].securityContext

描述

securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时, SecurityContext 中的值将具有优先权。

类型

object

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以比其父进程获得更多特权。此 bool 直接控制容器进程上是否设置了 no_new_privs 标志。当容器是 :1) 作为 CAP_SYS_ADMIN 具有 CAP_SYS_ADMIN 时，allowPrivilegeEscalation 始终为 true，请注意，当 spec.os.name 是 windows 时，不能设置此字段。
功能	object	从正在运行的容器中添加和移除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程本质上等同于主机上的 root。默认为 false。请注意，当 spec.os.name 是 windows 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，它将容器运行时默认值用于只读路径和屏蔽的路径。这要求启用 ProcMountType 功能标记。请注意，当 spec.os.name 是 windows 时无法设置此字段。 可能枚举值：- "Default" 对 /proc 的 readonly 和 masked 路径使用容器运行时默认值。大多数容器运行时屏蔽 /proc 中的特定路径以避免意外出现特殊设备或信息的安全。- "Unmasked" 会绕过容器运行时的默认行为，并确保新创建的 /proc 容器保留在不做任何修改的情况下。
readOnlyRootFilesystem	布尔值	此容器是否具有只读 root 文件系统。默认为 false。请注意，当 spec.os.name 是 windows 时无法设置此字段。

属性	类型	描述
runAsGroup	整数	用于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。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则 SecurityContext 中指定的值具有优先权。请注意，当 spec.os.name 是 windows 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，以确保它不作为 UID 0 (root) 运行，如果容器这样做，则无法启动容器。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 SecurityContext 中指定的值具有优先权。
runAsUser	整数	用于运行容器进程的入口点的 UID。如果未指定，则默认为在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 SecurityContext 中指定的值具有优先权。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/容器的 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。

3.1.79. .template.spec.containers[].securityContext.capabilities

描述

从正在运行的容器中添加和移除 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

3.1.80. .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。

3.1.81. .template.spec.containers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/容器的 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
----	----	----

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示应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。必须是一个降序路径，相对于 kubelet 配置的 seccomp 配置集位置。只有 type 为 "Localhost" 时，才应设置。
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文件。RuntimeDefault - 应使用容器运行时默认配置集。 unconfined - 不应应用配置集。 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)。

3.1.82. .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	GMSACredentialSpec 是 GMSA 准入 Webhook (https://github.com/kubernetes-sigs/windows-gmsa)内联由 <code>GMSACredentialSpec</code> 命名的 GMSA 凭证规格的内容。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证规格的名称。

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应作为"主机进程"容器运行。此字段是 alpha-level, 仅包含启用 WindowsHostProcessContainers 功能标记的组件。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。所有 Pod 的容器都必须具有相同的有效的 HostProcess 值（不允许混合 HostProcess 容器和非主机进程容器）。另外, 如果 HostProcess 为 true, 则 HostNetwork 也必须设置为 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, 则默认为在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 SecurityContext 中指定的值具有优先权。

3.1.83. .template.spec.containers[].startupProbe

描述

probe 描述要针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, 探测被视为失败的最小连续失败。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 Get 请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.84. .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，而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, 它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.85. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.86. .template.spec.containers[].startupProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型**object****必填**

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.87. .template.spec.containers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.88. .template.spec.containers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.89. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.90. .template.spec.containers[].volumeDevices

描述

volumeDevices 是容器要使用的块设备列表。

类型

数组

3.1.91. .template.spec.containers[].volumeDevices[]

描述

volumeDevice 描述了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 是该设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中 persistentVolumeClaim 的名称匹配

3.1.92. .template.spec.containers[].volumeMounts

描述

要挂载到容器文件系统的 Pod 卷。无法更新。

类型

数组

3.1.93. .template.spec.containers[].volumeMounts[]

描述

VolumeMount 描述了容器内卷挂载。

类型

object

必填

- 名称
- **mountPath**

属性	类型	描述
mountPath	字符串	应挂载卷的容器中的路径。不得包含 '!'

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定挂载如何从主机传播到容器以及反向传播。如果没有设置，则使用 MountPropagationNone。此字段在 1.10 中是 beta。</p> <p>可能枚举值： - "双向" 意味着容器中的卷将从主机或其他容器接收新挂载，并且自己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卷中的所有挂载("rshared"在 Linux 术语中)。 - "HostToContainer" 表示容器中的卷将从主机或其他容器接收新挂载，但容器中挂载的文件系统不会传播到主机或其他容器中。请注意，这个模式会递归应用到卷中的所有挂载("在 Linux 术语中")。 - "None" 表示容器中的卷不会从主机或其他容器接收新挂载，挂载到容器内的文件系统不会传播到主机或其他容器。请注意，这个模式在 Linux 术语中对应于"私有"。</p>
名称	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读方式挂载，否则读写(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 root）。
subPathExpr	字符串	在应该挂载容器卷的卷中扩展路径。行为与 SubPath 类似，但环境变量引用 \${VAR_NAME} 使用容器的环境扩展。默认为 ""（卷的 root）。SubPathExpr 和 SubPath 是互斥的。

3.1.94. .template.spec.dnsConfig

描述

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

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中。重复的名称服务器将被删除。
options	array	DNS 解析器选项列表。这将与 DNSPolicy 生成的基本选项合并。删除重复的条目。Options 中提供的解析选项将覆盖基本 DNSPolicy 中显示的选项。
options[]	对象	PodDNSConfigOption 定义 pod 的 DNS 解析器选项。
搜索	数组（字符串）	用于 host-name 查找的 DNS 搜索域列表。这将附加到从 DNSPolicy 生成的基本搜索路径中。删除重复的搜索路径。

3.1.95. .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。删除重复的条目。Options 中提供的解析选项将覆盖基本 DNSPolicy 中显示的选项。

类型

数组

3.1.96. .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	

3.1.97. .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，且无法通过更新 pod 规格来修改它。要将临时容器添加到现有 pod，请使用 pod 的 ephemeralcontainers 子资源。

类型

数组

3.1.98. .template.spec.ephemeralContainers[]

描述

EphemeralContainer 是一个临时容器，您可以添加到现有 Pod 中，用于用户发起的活动，如调试。临时容器没有资源或调度保证，它们不会在 Pod 退出或重启 Pod 时重启。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。临时容器可能无法删除或重启。

类型

object

必填

- name

属性	类型	描述
args	数组（字符串）	入口点的参数。如果未提供该镜像，则使用镜像的 CMD。变量引用 \$(VAR_NAME) 使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 \$，它允许转义 \$(VAR_NAME) 语法：即 "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会扩展，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.没有在 shell 中执行。如果未提供该镜像，则使用镜像的 ENTRYPOINT。变量引用 \$(VAR_NAME)使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 \$，它允许转义 \$(VAR_NAME)语法：即 "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会扩展，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	容器中要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	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(pull)策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签, 则默认为 Always, 否则则默认为 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 始终尝试拉取最新的镜像。如果拉取失败, 则容器将失败。- "IfNotPresent" 表示如果镜像在磁盘上不存在, kubelet 会拉取。如果镜像不存在且拉取失败, 则容器将失败。- "Never" 表示 kubelet 永不拉取镜像, 但只使用本地镜像。如果镜像不存在, 容器将失败</p>
lifecycle	object	lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序, 容器块的管理直到操作完成, 除非容器进程失败, 在这种情况下, 处理程序会被中止。
livenessProbe	object	probe 描述要针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。
名称	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	array	临时容器不允许使用端口。
ports[]	对象	containerPort 代表单个容器中的网络端口。
readinessProbe	object	probe 描述要针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。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源重新定义策略。

属性	类型	描述
resources	对象	ResourceRequirements 描述了计算资源要求。
securityContext	object	securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时，SecurityContext 中的值将具有优先权。
startupProbe	object	probe 描述要针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	此容器是否应该在容器运行时中为 stdin 分配缓冲区。如果没有设置，则容器中的 stdin 读取将始终会导致 EOF。默认为 false。
stdinOnce	布尔值	容器运行时是否应在由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中保持打开状态。如果 stdinOnce 设为 true，则 stdin 会在容器启动时打开，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，此时 stdin 已关闭并保持关闭，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。默认为 false
targetContainerName	字符串	<p>如果设置，则来自此临时容器目标的 PodSpec 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。如果运行时不支持命名空间目标，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的 路径挂载到容器的文件系统中。 编写的消息应当是最终状态，如断 言失败消息。如果超过 4096 字 节，节点将截断。所有容器的总消 息长度将限制为 12kb。默认为 /dev/termination-log。无法更 新。
terminationMessagePolicy	字符串	指明应当如何填充终止消息。文件 将使用 terminationMessagePath 的内容在成功或失败时填充容器状 态消息。如果终止消息文件为空， 则 FallbackToLogsOnError 将使用 容器日志输出的最后块，并且容器 退出并显示错误。日志输出限制为 2048 字节或 80 行，以较小者。 默认为 File。无法更新。 可能枚举值：- "FallbackToLogsOnError" 将 在容器退出时读取容器状态消息的 最新内容，并且 terminationMessagePath 没有任 何内容。- "File" 是默认行为，并 在容器退出时将容器状态消息设置 为容器的 terminationMessagePath 的内 容。
tty	布尔值	此容器是否应为自己分配 TTY，也 要求 'stdin' 为 true。默认为 false。
volumeDevices	array	volumeDevices 是容器要使用的块 设备列表。
volumeDevices[]	对象	volumeDevice 描述了容器中原始 块设备的映射。
volumeMounts	array	要挂载到容器文件系统的 Pod 卷。 临时容器不允许使用 subPath 挂 载。无法更新。
volumeMounts[]	对象	VolumeMount 描述了容器内卷挂 载。

属性	类型	描述
workingDir	字符串	容器的工作目录。如果未指定，则将使用容器运行时的默认值，该默认值可能在容器镜像中配置。无法更新。

3.1.99. .template.spec.ephemeralContainers[].env

描述

容器中要设置的环境变量列表。无法更新。

类型

数组

3.1.100. .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以及任何服务环境变量进行扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 \$，它允许转义 \$(VAR_NAME) 语法：即 "(VAR_NAME)" 将生成字符串 literal "\$(VAR_NAME)"。转义的引用永远不会扩展，无论变量是否存在。默认为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的值的源。

3.1.101. .template.spec.ephemeralContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的值的源。

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。

3.1.102. .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

3.1.103. .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
<code>apiVersion</code>	字符串	模式的版本是按术语编写的 FieldPath，默认为 "v1"。
<code>fieldPath</code>	字符串	在指定 API 版本中选择的字段路径。

3.1.104. `.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
<code>containerName</code>	字符串	容器名称：卷需要，对于 env vars 是可选的
<code>divisor</code>	数量	指定公开资源的输出格式，默认为 "1"
<code>resource</code>	字符串	必需：要选择的资源

3.1.105. `.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
<code>key</code>	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 还是其密钥

3.1.106. .template.spec.ephemeralContainers[].envFrom

描述

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

3.1.107. .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 字段的内容将作为环境变量表示键值对。
prefix	字符串	要添加到 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 Data 字段的内容将作为环境变量表示键值对。

3.1.108. .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 字段的内容将作为环境变量表示键值对。

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

3.1.109. .template.spec.ephemeralContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的 Data 字段的内容将作为环境变量表示键值对。

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 Secret

3.1.110. .template.spec.ephemeralContainers[].lifecycle

描述

lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序，容器块的管理直到操作完成，除非容器进程失败，在这种情况下，处理程序会被中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

属性	类型	描述
preStop	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

3.1.111. .template.spec.ephemeralContainers[].lifecycle.postStart

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

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 Get 请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

3.1.112. .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，而命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.113. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.114. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeader:

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.115. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeader:

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.116. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.117. .template.spec.ephemeralContainers[].lifecycle.preStop

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

类型

object

属性	类型	描述
exec	object	ExecAction 描述了 "run in container" 操作。

属性	类型	描述
httpGet	object	HTTPGetAction 描述了基于 HTTP Get 请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

3.1.118. .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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.119. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。

属性	类型	描述
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.120. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.121. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.122. .template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

描述

TCP SocketAction 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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.123. .template.spec.ephemeralContainers[].livenessProbe**描述**

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。
initialDelaySeconds	整数	容器启动存活度探测前的秒数。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

属性	类型	描述
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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.124. .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
----	----	----

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，而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，它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.125. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.126. .template.spec.ephemeralContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.127. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.128. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.129. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.130. .template.spec.ephemeralContainers[].ports

描述

临时容器不允许使用端口。

类型

数组

3.1.131. .template.spec.ephemeralContainers[].ports[]

描述

containerPort 代表单个容器中的网络端口。

类型

object

必填

- **containerPort**

属性	类型	描述
containerPort	整数	pod 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	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，它必须与 ContainerPort 匹配。大多数容器都不需要这样做。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，在 pod 中唯一。pod 中的每个命名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"。 可能枚举值：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

3.1.132. .template.spec.ephemeralContainers[].readinessProbe

描述

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.133. .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，而命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.134. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.135. .template.spec.ephemeralContainers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.136. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.137. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.138. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.139. .template.spec.ephemeralContainers[].resizePolicy

描述

容器的资源重新定义策略。

类型

数组

3.1.140. .template.spec.ephemeralContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源重新定义策略。

类型

object

必填

- **resourceName**

- restartPolicy

属性	类型	描述
resourceName	string	此资源重新定义策略应用到的资源的名称。支持的值有：cpu、memory。
restartPolicy	string	在指定资源调整大小时，重启策略会被应用。如果没有指定，则默认为 NotRequired。

3.1.141. .template.spec.ephemeralContainers[].resources

描述

ResourceRequirements 描述了计算资源要求。

类型

object

属性	类型	描述
声明	数组	claim 列出此容器使用的 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。 此字段不可变。它只能为容器设置。
claims[]	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/

属性	类型	描述
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/

3.1.142. .template.spec.ephemeralContainers[].resources.claims

描述

claim 列出此容器使用的 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。

此字段不可变。它只能为容器设置。

类型

数组

3.1.143. .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 中的一个条目的名称匹配。它允许容器内部使用该资源。

3.1.144. .template.spec.ephemeralContainers[].securityContext

描述

securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时, SecurityContext 中的值将具有优先权。

类型

object

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以比其父进程获得更多特权。此 bool 直接控制容器进程上是否设置了 no_new_privs 标志。当容器是 :1) 作为 CAP_SYS_ADMIN 具有 CAP_SYS_ADMIN 时，allowPrivilegeEscalation 始终为 true，请注意，当 spec.os.name 是 windows 时，不能设置此字段。
功能	object	从正在运行的容器中添加和移除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程本质上等同于主机上的 root。默认为 false。请注意，当 spec.os.name 是 windows 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，它将容器运行时默认值用于只读路径和屏蔽的路径。这要求启用 ProcMountType 功能标记。请注意，当 spec.os.name 是 windows 时无法设置此字段。 可能枚举值：- "Default" 对 /proc 的 readonly 和 masked 路径使用容器运行时默认值。大多数容器运行时屏蔽 /proc 中的特定路径以避免意外出现特殊设备或信息的安全。- "Unmasked" 会绕过容器运行时的默认行为，并确保新创建的 /proc 容器保留在不做任何修改的情况下。
readOnlyRootFilesystem	布尔值	此容器是否具有只读 root 文件系统。默认为 false。请注意，当 spec.os.name 是 windows 时无法设置此字段。

属性	类型	描述
runAsGroup	整数	用于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。也可以在PodSecurityContext中设置。如果在SecurityContext和PodSecurityContext中设置，则SecurityContext中指定的值具有优先权。请注意，当spec.os.name是windows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，以确保它不作为UID 0 (root)运行，如果容器这样做，则无法启动容器。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SecurityContext中指定的值具有优先权。
runAsUser	整数	用于运行容器进程的入口点的UID。如果未指定，则默认为在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SecurityContext中指定的值具有优先权。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/容器的 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。

3.1.145. .template.spec.ephemeralContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和移除 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

3.1.146. .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。
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。

3.1.147. .template.spec.ephemeralContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/容器的 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
----	----	----

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示应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。必须是一个降序路径，相对于 kubelet 配置的 seccomp 配置集位置。只有 type 为 "Localhost" 时，才应设置。
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文件。RuntimeDefault - 应使用容器运行时默认配置集。 unconfined - 不应应用配置集。 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)。

3.1.148. .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)内联由 GMSACredentialSpec 命名的 GMSA 凭证规格的内容。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证规格的名称。

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应作为"主机进程"容器运行。此字段是 alpha-level, 仅包含启用 WindowsHostProcessContainers 功能标记的组件。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。所有 Pod 的容器都必须具有相同的有效的 HostProcess 值 (不允许混合 HostProcess 容器和非主机进程容器)。另外, 如果 HostProcess 为 true, 则 HostNetwork 也必须设置为 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, 则默认为在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 SecurityContext 中指定的值具有优先权。

3.1.149. .template.spec.ephemeralContainers[].startupProbe

描述

probe 描述要针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, 探测被视为失败的最小连续失败。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 Get 请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.150. .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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.151. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.152. .template.spec.ephemeralContainers[].startupProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型**object****必填**

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.153. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.154. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.155. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.156. .template.spec.ephemeralContainers[].volumeDevices

描述

volumeDevices 是容器要使用的块设备列表。

类型

数组

3.1.157. .template.spec.ephemeralContainers[].volumeDevices[]

描述

volumeDevice 描述了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 是该设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中 persistentVolumeClaim 的名称匹配

3.1.158. .template.spec.ephemeralContainers[].volumeMounts

描述

要挂载到容器文件系统的 Pod 卷。临时容器不允许使用 subPath 挂载。无法更新。

类型

数组

3.1.159. .template.spec.ephemeralContainers[].volumeMounts[]

描述

VolumeMount 描述了容器内卷挂载。

类型

object

必填

- 名称
- **mountPath**

属性	类型	描述
mountPath	字符串	应挂载卷的容器中的路径。不得包含 '!'

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定挂载如何从主机传播到容器以及反向传播。如果没有设置，则使用 MountPropagationNone。此字段在 1.10 中是 beta。</p> <p>可能枚举值：- "双向" 意味着容器中的卷将从主机或其他容器接收新挂载，并且自己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卷中的所有挂载("rshared"在 Linux 术语中)。- "HostToContainer" 表示容器中的卷将从主机或其他容器接收新挂载，但容器中挂载的文件系统不会传播到主机或其他容器中。请注意，这个模式会递归应用到卷中的所有挂载("在 Linux 术语中")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，挂载到容器内的文件系统不会传播到主机或其他容器。请注意，这个模式在 Linux 术语中对应于"私有"。</p>
名称	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读方式挂载，否则读写(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 root）。
subPathExpr	字符串	在应该挂载容器卷的卷中扩展路径。行为与 SubPath 类似，但环境变量引用 \$(VAR_NAME)使用容器的环境扩展。默认为 ""（卷的 root）。SubPathExpr 和 SubPath 是互斥的。

3.1.160. .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。这仅对非主机网络 pod 有效。

类型

数组

3.1.161. .template.spec.hostAliases[]

描述

HostAlias 包含 IP 和主机名之间的映射，这些映射将作为 pod 的主机文件中的条目注入。

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。

3.1.162. .template.spec.imagePullSecrets**描述**

imagePullSecrets 是对同一命名空间中的 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

3.1.163. .template.spec.imagePullSecrets[]**描述**

LocalObjectReference 包含足够信息，供您在同一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

3.1.164. .template.spec.initContainers**描述**

属于 pod 的初始化容器列表。Init 容器在启动容器之前按顺序执行。如果任何 init 容器失败，pod 被视为失败，并根据 restartPolicy 处理。init 容器或普通容器的名称对于所有容器来说都必须是唯一的。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。通过查找每种资源类型的最高请求/限制，然后使用该值的最大值或普通容器的总和，在调度期间考虑 init 容器的 resourceRequirements。限制以类似的方式应用到 init 容器。当前无法添加或删除 init 容器。无法更新。更多信息：<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

类型

数组

3.1.165. .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> 使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：即 <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会扩展，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。没有在 shell 中执行。如果未提供此容器镜像，则使用容器镜像的 ENTRYPOINT。变量引用 <code>\$(VAR_NAME)</code> 使用容器的环境扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 <code>\$</code> ，它允许转义 <code>\$(VAR_NAME)</code> 语法：即 <code>"(VAR_NAME)"</code> 将生成字符串 literal <code>\$(VAR_NAME)</code> 。转义的引用永远不会扩展，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	容器中要设置的环境变量列表。无法更新。

属性	类型	描述
<code>env[]</code>	对象	EnvVar 代表容器中存在的环境变量。
<code>envFrom</code>	array	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。
<code>envFrom[]</code>	对象	EnvFromSource 代表一组 ConfigMap 的源
<code>image</code>	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（如 Deployments 和 StatefulSets）中的容器镜像。
<code>imagePullPolicy</code>	字符串	镜像拉取(pull)策略。Always, Never, IfNotPresent 之一。如果指定了 :latest 标签，则默认为 Always，否则则默认为 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 始终尝试拉取最新的镜像。如果拉取失败，则容器将失败。- "IfNotPresent" 表示如果镜像在磁盘上不存在，kubelet 会拉取。如果镜像不存在且拉取失败，则容器将失败。- "Never" 表示 kubelet 永不拉取镜像，但只使用本地镜像。如果镜像不存在，容器将失败
<code>lifecycle</code>	object	lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序，容器块的管理直到操作完成，除非容器进程失败，在这种情况下，处理程序会被中止。

属性	类型	描述
livenessProbe	object	probe 描述要针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。
名称	字符串	指定为 DNS_LABEL 的容器名称。pod 中的每个容器都必须具有唯一的名称(DNS_LABEL)。无法更新。
ports	array	从容器公开的端口列表。这里没有指定端口不会阻止公开该端口。所有正在侦听容器内默认"0.0.0.0"地址的端口均可从网络访问。使用策略合并补丁修改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 代表单个容器中的网络端口。
readinessProbe	object	probe 描述要针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。
resizePolicy[]	object	ContainerResizePolicy 代表容器的资源重新定义策略。
resources	对象	ResourceRequirements 描述了计算资源要求。
securityContext	object	securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时，SecurityContext 中的值将具有优先权。
startupProbe	object	probe 描述要针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。

属性	类型	描述
stdin	布尔值	此容器是否应该在容器运行时中为 stdin 分配缓冲区。如果没有设置，则容器中的 stdin 读取将始终会导致 EOF。默认为 false。
stdinOnce	布尔值	容器运行时是否应在由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中保持打开状态。如果 stdinOnce 设为 true，则 stdin 会在容器启动时打开，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，此时 stdin 已关闭并保持关闭，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。默认为 false
terminationMessagePath	字符串	可选：将容器终止消息写入的文件的路径挂载到容器的文件系统中。编写的消息应当是最终状态，如断言失败消息。如果超过 4096 字节，节点将截断。所有容器的总消息长度将限制为 12kb。默认为 /dev/termination-log。无法更新。
terminationMessagePolicy	字符串	指明应当如何填充终止消息。文件将使用 terminationMessagePath 的内容在成功或失败时填充容器状态消息。如果终止消息文件为空，则 FallbackToLogsOnError 将使用容器日志输出的最后块，并且容器退出并显示错误。日志输出限制为 2048 字节或 80 行，以较小者。默认为 File。无法更新。 可能枚举值：- "FallbackToLogsOnError" 将在容器退出时读取容器状态消息的最新内容，并且 terminationMessagePath 没有任何内容。- "File" 是默认行为，并在容器退出时将容器状态消息设置为容器的 terminationMessagePath 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，也要求 'stdin' 为 true。默认为 false。
volumeDevices	array	volumeDevices 是容器要使用的块设备列表。
volumeDevices[]	对象	volumeDevice 描述了容器中原始块设备的映射。
volumeMounts	array	要挂载到容器文件系统的 Pod 卷。无法更新。
volumeMounts[]	对象	VolumeMount 描述了容器内卷挂载。
workingDir	字符串	容器的工作目录。如果未指定，则将使用容器运行时的默认值，该默认值可能在容器镜像中配置。无法更新。

3.1.166. .template.spec.initContainers[].env

描述

容器上要设置的环境变量列表。无法更新。

类型

数组

3.1.167. .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 $\$(VAR_NAME)$ 使用容器中之前定义的环境变量以及任何服务环境变量进行扩展。如果无法解析变量，则输入字符串中的引用将保持不变。双引号被缩减为单个 $\$$ ，它允许转义 $\$(VAR_NAME)$ 语法：即 $\$(VAR_NAME)$ 将生成字符串 literal $\$(VAR_NAME)$ 。转义的引用永远不会扩展，无论变量是否存在。默认为 ""。
valueFrom	object	EnvVarSource 代表 EnvVar 的值的源。

3.1.168. .template.spec.initContainers[].env[].valueFrom

描述

EnvVarSource 代表 EnvVar 的值的源。

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。

3.1.169. .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

3.1.170. .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	模式的版本是按术语编写的 FieldPath，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

3.1.171. .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：卷需要，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

3.1.172. .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 还是其密钥

3.1.173. .template.spec.initContainers[].envFrom

描述

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

3.1.174. .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 字段的内容将作为环境变量表示键值对。
prefix	字符串	要添加到 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 Data 字段的内容将作为环境变量表示键值对。

3.1.175. .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 字段的内容将作为环境变量表示键值对。

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

3.1.176. .template.spec.initContainers[].envFrom[].secretRef

描述

SecretEnvSource 选择一个 Secret 来填充环境变量。
目标 Secret 的 Data 字段的内容将作为环境变量表示键值对。

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
optional	布尔值	指定是否必须定义 Secret

3.1.177. .template.spec.initContainers[].lifecycle

描述

lifecycle 描述了管理系统响应容器生命周期事件应执行的操作。对于 PostStart 和 PreStop 生命周期处理程序，容器块的管理直到操作完成，除非容器进程失败，在这种情况下，处理程序会被中止。

类型

object

属性	类型	描述
postStart	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。
preStop	object	LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

3.1.178. .template.spec.initContainers[].lifecycle.postStart

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

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 Get 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

3.1.179. .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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.180. .template.spec.initContainers[].lifecycle.postStart.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值：- "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.181. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.182. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.183. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.184. .template.spec.initContainers[].lifecycle.preStop

描述

LifecycleHandler 定义应在生命周期 hook 中执行的操作。除 TCPSocket 外，只能指定其中一个字段。

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 Get 请求的操作。
tcpSocket	object	TCPSocketAction 描述了基于打开套接字的操作

3.1.185. .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，而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, 它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.186. .template.spec.initContainers[].lifecycle.preStop.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.187. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.188. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]**描述**

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.189. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.190. .template.spec.initContainers[].livenessProbe

描述

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.191. .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，而命令的工作目录是容器文件系统中的 root ('/')。命令只是 exec'd，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.192. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.193. .template.spec.initContainers[].livenessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- **port**

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值：- "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.194. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.195. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.196. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.197. .template.spec.initContainers[].ports

描述

从容器公开的端口列表。这里没有指定端口不会阻止公开该端口。所有正在侦听容器内默认"0.0.0.0"地址的端口均可从网络访问。使用策略合并补丁修改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

3.1.198. .template.spec.initContainers[].ports[]

描述

containerPort 代表单个容器中的网络端口。

类型

object

必填

- containerPort

属性	类型	描述
containerPort	整数	pod 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。

属性	类型	描述
hostPort	整数	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，它必须与 ContainerPort 匹配。大多数容器都不需要这样做。
name	字符串	如果指定，这必须是 IANA_SVC_NAME，在 pod 中唯一。pod 中的每个命名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"。 可能枚举值：- "SCTP" 是 SCTP 协议。- "TCP" 是 TCP 协议。- "UDP" 是 UDP 协议。

3.1.199. .template.spec.initContainers[].readinessProbe

描述

probe 描述要针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，探测被视为失败的最小连续失败。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 Get 请求的操作。
initialDelaySeconds	整数	容器启动存活度探测前的秒数。更多信息： https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

属性	类型	描述
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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.200. .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
----	----	----

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，而命令的工作目录是容器文件系统中的 root ('/'). 命令只是 exec'd, 它不会在 shell 中运行，因此传统的 shell 指令('!' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.201. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.202. .template.spec.initContainers[].readinessProbe.httpGet

描述

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型

object

必填

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.203. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.204. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.205. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.206. .template.spec.initContainers[].resizePolicy

描述

容器的资源重新定义策略。

类型

数组

3.1.207. .template.spec.initContainers[].resizePolicy[]

描述

ContainerResizePolicy 代表容器的资源重新定义策略。

类型

object

必填

- **resourceName**

- **restartPolicy**

属性	类型	描述
resourceName	string	此资源重新定义策略应用到的资源的名称。支持的值有：cpu、memory。
restartPolicy	string	在指定资源调整大小时，重启策略会被应用。如果没有指定，则默认为 NotRequired。

3.1.208. .template.spec.initContainers[].resources

描述

ResourceRequirements 描述了计算资源要求。

类型

object

属性	类型	描述
声明	数组	claim 列出此容器使用的 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。 此字段不可变。它只能为容器设置。
claims[]	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/

属性	类型	描述
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/

3.1.209. .template.spec.initContainers[].resources.claims

描述

claim 列出此容器使用的 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。

此字段不可变。它只能为容器设置。

类型

数组

3.1.210. .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 中的一个条目的名称匹配。它允许容器内部使用该资源。

3.1.211. .template.spec.initContainers[].securityContext

描述

securityContext 包含要应用到容器的安全配置。SecurityContext 和 PodSecurityContext 中都存在一些字段。当两者都被设置时, SecurityContext 中的值将具有优先权。

类型

object

属性	类型	描述
allowPrivilegeEscalation	布尔值	allowPrivilegeEscalation 控制进程是否可以比其父进程获得更多特权。此 bool 直接控制容器进程上是否设置了 no_new_privs 标志。当容器是 :1) 作为 CAP_SYS_ADMIN 具有 CAP_SYS_ADMIN 时，allowPrivilegeEscalation 始终为 true，请注意，当 spec.os.name 是 windows 时，不能设置此字段。
功能	object	从正在运行的容器中添加和移除 POSIX 功能。
privileged	布尔值	以特权模式运行容器。特权容器中的进程本质上等同于主机上的 root。默认为 false。请注意，当 spec.os.name 是 windows 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，它将容器运行时默认值用于只读路径和屏蔽的路径。这要求启用 ProcMountType 功能标记。请注意，当 spec.os.name 是 windows 时无法设置此字段。 可能枚举值： - "Default" 对 /proc 的 readonly 和 masked 路径使用容器运行时默认值。大多数容器运行时屏蔽 /proc 中的特定路径以避免意外出现特殊设备或信息的安全。 - "Unmasked" 会绕过容器运行时的默认行为，并确保新创建的 /proc 容器保留在不做任何修改的情况下。
readOnlyRootFilesystem	布尔值	此容器是否具有只读 root 文件系统。默认为 false。请注意，当 spec.os.name 是 windows 时无法设置此字段。

属性	类型	描述
runAsGroup	整数	用于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。也可以在 PodSecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则 SecurityContext 中指定的值具有优先权。请注意，当 spec.os.name 是 windows 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，以确保它不作为 UID 0 (root) 运行，如果容器这样做，则无法启动容器。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 SecurityContext 中指定的值具有优先权。
runAsUser	整数	用于运行容器进程的入口点的 UID。如果未指定，则默认为在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 SecurityContext 中指定的值具有优先权。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/容器的 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。

3.1.212. .template.spec.initContainers[].securityContext.capabilities

描述

从正在运行的容器中添加和移除 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

3.1.213. .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。
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。

3.1.214. .template.spec.initContainers[].securityContext.seccompProfile

描述

SeccompProfile 定义 pod/容器的 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
----	----	----

属性	类型	描述
localhostProfile	字符串	localhostProfile 表示应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。必须是一个降序路径，相对于 kubelet 配置的 seccomp 配置集位置。只有 type 为 "Localhost" 时，才应设置。
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文件。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用配置集。 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)。

3.1.215. .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)内联由 <code>GMSACredentialSpec</code> 命名的 GMSA 凭证规格的内容。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证规格的名称。

属性	类型	描述
hostProcess	布尔值	HostProcess 确定容器是否应作为"主机进程"容器运行。此字段是 alpha-level, 仅包含启用 WindowsHostProcessContainers 功能标记的组件。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。所有 Pod 的容器都必须具有相同的有效的 HostProcess 值（不允许混合 HostProcess 容器和非主机进程容器）。另外, 如果 HostProcess 为 true, 则 HostNetwork 也必须设置为 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, 则默认为在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 SecurityContext 中指定的值具有优先权。

3.1.216. .template.spec.initContainers[].startupProbe

描述

probe 描述要针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, 探测被视为失败的最小连续失败。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 Get 请求的操作。

属性	类型	描述
initialDelaySeconds	整数	容器启动存活度探测前的秒数。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，探测至少连续成功会被视为成功。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	pod 在探测失败时安全终止所需的可选持续时间（以秒为单位）。宽限期是 pod 中运行的进程发送终止信号和进程被强制停止使用 kill 信号的时间（以秒为单位）。为您的进程设置这个值超过预期清理时间。如果这个值是 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。值为零表示立即通过终止信号停止（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

3.1.217. .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，而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，它不会在 shell 中运行，因此传统的 shell 指令(' ' 等)将无法工作。要使用 shell，您需要明确调用该 shell。0 的退出状态被视为 live/healthy，非零不健康。

3.1.218. .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。number 必须在 1 到 65535 之间。
service	string	service 是要放入 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，则默认行为由 gRPC 定义。

3.1.219. .template.spec.initContainers[].startupProbe.httpGet**描述**

HTTPGetAction 描述了基于 HTTP Get 请求的操作。

类型**object****必填**

- port

属性	类型	描述
主机	字符串	要连接到的主机名，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 允许重复的标头。
httpHeaders[]	对象	HTTPHeader 描述了在 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	用于访问容器的端口的名称或编号。number 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。 可能枚举值： - "HTTP" 表示使用的方案为 http:// - "HTTPS" 表示使用的方案是 https://

3.1.220. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders

描述

在请求中设置的自定义标头。HTTP 允许重复的标头。

类型

数组

3.1.221. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

描述

HTTPHeader 描述了在 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，因此理解 case-variant 名称与相同的标头一样。
value	字符串	标头字段值

3.1.222. .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	用于访问容器的端口的编号或名称。number 必须在 1 到 65535 之间。名称必须是 IANA_SVC_NAME。

3.1.223. .template.spec.initContainers[].volumeDevices

描述

volumeDevices 是容器要使用的块设备列表。

类型

数组

3.1.224. .template.spec.initContainers[].volumeDevices[]

描述

volumeDevice 描述了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 是该设备要映射到的容器内的路径。
name	字符串	name 必须与 pod 中 persistentVolumeClaim 的名称匹配

3.1.225. .template.spec.initContainers[].volumeMounts

描述

要挂载到容器文件系统的 Pod 卷。无法更新。

类型

数组

3.1.226. .template.spec.initContainers[].volumeMounts[]

描述

VolumeMount 描述了容器内卷挂载。

类型

object

必填

- 名称
- **mountPath**

属性	类型	描述
mountPath	字符串	应挂载卷的容器中的路径。不得包含 '!'

属性	类型	描述
mountPropagation	字符串	<p>mountPropagation 决定挂载如何从主机传播到容器以及反向传播。如果没有设置，则使用 MountPropagationNone。此字段在 1.10 中是 beta。</p> <p>可能枚举值：- "双向" 意味着容器中的卷将从主机或其他容器接收新挂载，并且自己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卷中的所有挂载("rshared"在 Linux 术语中)。- "HostToContainer" 表示容器中的卷将从主机或其他容器接收新挂载，但容器中挂载的文件系统不会传播到主机或其他容器中。请注意，这个模式会递归应用到卷中的所有挂载("在 Linux 术语中")。- "None" 表示容器中的卷不会从主机或其他容器接收新挂载，挂载到容器内的文件系统不会传播到主机或其他容器。请注意，这个模式在 Linux 术语中对应于"私有"。</p>
名称	字符串	这必须与卷的 Name 匹配。
readOnly	布尔值	如果为 true，则以只读方式挂载，否则读写(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 root）。
subPathExpr	字符串	在应该挂载容器卷的卷中扩展路径。行为与 SubPath 类似，但环境变量引用 \$(VAR_NAME)使用容器的环境扩展。默认为 ""（卷的 root）。SubPathExpr 和 SubPath 是互斥的。

3.1.227. .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。当前支持的值有 linux 和 windows。以后可以定义额外的值，可以是： https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration 客户端应该预期处理额外的值，并将此字段中未识别的值视为 os: null

3.1.228. .template.spec.readinessGates

描述

如果指定，则会针对 pod 就绪评估所有就绪度。当所有容器都就绪且在就绪的 gates 中指定的所有条件都等于 "True" More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>时，pod 已就绪：<https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

类型

数组

3.1.229. .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。

3.1.230. .template.spec.resourceClaims

描述

ResourceClaims 定义在 Pod 允许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。

此字段不可变。

类型

数组

3.1.231. .template.spec.resourceClaims[]

描述

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

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。 应设置这些字段中的一个。此类型的消费者必须将空对象视为未知值。

3.1.232. .template.spec.resourceClaims[].source

描述

ClaimSource 描述了对 ResourceClaim 的引用。
应设置这些字段中的一个。此类型的消费者必须将空对象视为未知值。

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>该模板将用于创建新的 ResourceClaim，它将绑定到此容器集。删除此 pod 时，ResourceClaim 也会被删除。ResourceClaim 的名称将是 <pod name>-<resource name>，其中 <resource name> 是 PodResourceClaim.Name。如果串联的名称对 ResourceClaim（例如太长）无效，则 Pod 验证将拒绝 pod。</p> <p>具有名称不归 pod 所有的现有 ResourceClaim 将不会用于 pod，以避免错误地使用不相关的资源。然后，调度和 pod 启动会被阻止，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 进行任何更改。</p>

3.1.233. .template.spec.schedulingGates

描述

SchedulingGates 是不透明的值列表，如果指定，它将阻止调度 pod。如果 schedulingGates 不为空，pod 将保持在 SchedulingGated 状态，调度程序不会尝试调度 pod。SchedulingGates 只能在 pod 创建时设置，之后才能被删除。

这是 PodSchedulingReadiness 功能门启用的 beta 功能。

类型

数组

3.1.234. .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	调度最低要求的名称。每个调度最低要求都必须有一个唯一的 name 字段。

3.1.235. .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由 pod 所有：</p> <p>1.拥有的 GID 为 FSGroup 2。设置 setgid 位（卷中创建的新文件将归 FSGroup 所有）3。权限位是 OR'd with rw-rw---</p> <p>如果未设置，Kubelet 不会修改任何卷的所有权和权限。请注意，当 spec.os.name 是 windows 时无法设置此字段。</p>

属性	类型	描述
fsGroupChangePolicy	字符串	<p>fsGroupChangePolicy 定义在 Pod 内公开卷之前更改卷的所有权和权限的行为。此字段将只适用于支持基于 fsGroup 的所有权（和权限）的卷类型。它对临时卷类型没有影响，如 secret、configmap 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 windows 时无法设置此字段。</p> <p>可能枚举值：- "Always" 表示卷的所有权和权限在 Pod 中挂载时应始终改变。这个默认行为 - "OnRootMismatch" 表示只有在根目录的权限和所有权与卷中预期权限不匹配时，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>
runAsGroup	整数	<p>用于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。还可能会在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则 SecurityContext 中指定的值优先于该容器。请注意，当 spec.os.name 是 windows 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，以确保它不作为 UID 0 (root) 运行，如果容器这样做，则无法启动容器。如果未设置或 false，则不会执行这样的验证。还可能会在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则 SecurityContext 中指定的值具有优先权。</p>

属性	类型	描述
runAsUser	整数	用于运行容器进程的入口点的 UID。如果未指定，则默认为在镜像元数据中指定的用户。还可能会在 SecurityContext 中设置。如果在 SecurityContext 和 PodSecurityContext 中设置，则 SecurityContext 中指定的值优先于该容器。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/容器的 seccomp 配置集设置。只能设置一个配置文件源。
supplementalGroups	数组（整数）	应用到每个容器中的第一个进程的组列表，以及容器的主 GID、fsGroup（如果指定）以及容器镜像中定义的组成员资格，用于容器进程的 uid。如果未指定，则不会将其他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，即使它们没有包含在此列表中。请注意，当 spec.os.name 是 windows 时无法设置此字段。
sysctls	array	sysctl 包含用于 pod 的命名空间 sysctl 列表。带有不支持 sysctl（容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是 windows 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。

3.1.236. .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。

3.1.237. .template.spec.securityContext.seccompProfile

描述

SeccompProfile 定义 pod/容器的 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 表示应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。必须是一个降序路径，相对于 kubelet 配置的 seccomp 配置集位置。只有 type 为 "Localhost" 时，才应设置。

属性	类型	描述
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文件。RuntimeDefault - 应使用容器运行时默认配置集。unconfined - 不应应用配置集。 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)。

3.1.238. .template.spec.securityContext.sysctls

描述

sysctl 包含用于 pod 的命名空间 sysctl 列表。带有不支持 sysctl（容器运行时）的 Pod 可能无法启动。请注意，当 spec.os.name 是 windows 时无法设置此字段。

类型

数组

3.1.239. .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

3.1.240. .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)内联由 GMSACredentialSpec 命名的 GMSA 凭证规格的内容。
gmsaCredentialSpecName	字符串	GMSACredentialSpecName 是要使用的 GMSA 凭证规格的名称。
hostProcess	布尔值	HostProcess 确定容器是否应作为"主机进程"容器运行。此字段是 alpha-level, 仅包含启用 WindowsHostProcessContainers 功能标记的组件。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。所有 Pod 的容器都必须具有相同的有效的 HostProcess 值 (不允许混合 HostProcess 容器和非主机进程容器)。另外, 如果 HostProcess 为 true, 则 HostNetwork 也必须设置为 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, 则默认为在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 SecurityContext 中指定的值具有优先权。

3.1.241. .template.spec.tolerations

描述

如果指定, pod 的容限。

类型

数组

3.1.242. .template.spec.tolerations[]

描述

此 Toleration 附加到 pod，以使用匹配的 operator <operator> 容许与 triple <key,value,effect> 匹配的任何污点。

类型

object

属性	类型	描述
effect	字符串	<p>effect 表示要匹配的污点效果。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 any already-running pod 不容许该污点。目前，由 NodeController 强制执行 - "NoSchedule" 不允许新 pod 调度到该节点上，除非容许污点，但允许所有提交至 Kubelet 的 pod 而无需通过调度程序启动，并允许所有已在运行的 pod 继续运行。由调度程序强制执行。 - "PreferNoSchedule" Like TaintEffectNoSchedule, 但调度程序不会将新 pod 调度到该节点上，而不是阻止新 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，则必须存在运算符；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。默认值为 Equal。exists 等同于值的通配符，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（必须生效 NoExecute, 否则此字段将被忽略）可以容忍污点。默认情况下，它不会被设置，这意味着容许任何污点（不要驱除）。零值和负值将被视为 0（立即删除）。</p>

属性	类型	描述
value	字符串	value 是容限匹配的污点值。如果运算符是 Exists，则该值应该为空，否则仅是一个常规字符串。

3.1.243. .template.spec.topologySpreadConstraints

描述

topologySpreadConstraints 描述了一组如何在拓扑域间分布的 pod。调度程序将按约束来调度 pod。所有 topologySpreadConstraints 都是 AND。

类型

数组

3.1.244. .template.spec.topologySpreadConstraints[]

描述

TopologySpreadConstraint 指定如何在给定的拓扑中分布匹配的 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。

属性	类型	描述
matchLabelKeys	数组（字符串）	<p>matchLabelKeys 是一组 pod 标签键，用于选择要在其中计算分布的 pod。这些键用于从传入的 pod 标签中查找值，这些键值标签由 labelSelector 匹配，以选择现有 pod 的组，用于为传入的 pod 计算。匹配键被禁止在 MatchLabelKeys 和 LabelSelector 中。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 或空列表表示仅与 labelSelector 匹配。</p> <p>这是一个 beta 字段，需要启用 MatchLabelKeysInPodTopologySpread 功能门（默认启用）。</p>
maxSkew	整数	<p>maxSkew 描述了 pod 可能没有被均匀分布的程度。当 whenUnsatisfiable=DoNotSchedule 时，它是目标拓扑中匹配 pod 数量和全局最小值之间允许的最大区别。如果符合条件的域数量小于 MinDomains，则全局最小值是合格域中匹配 pod 的最小数量。例如，在一个 3 区集群中，MaxSkew 被设置为 1，并且具有与 2/2/1 相同的 labelSelector 的 pod：在这种情况下，全局最小值为 1。 zone1 zone2 zone3 P P P P P P P - 如果 MaxSkew 是 1，则传入的 pod 只能调度到 zone3 变为 2/2/2；将它调度到 zone1 (zone2) 上，使 ActualSkew (3-1) on zone1 (zone2) 违反 MaxSkew (1)。- 如果 MaxSkew，则 pod 可以调度到任何区(zone2)。当 whenUnsatisfiable=ScheduleAnyway 时，它用于为满足它的拓扑赋予更高的优先级。这是必填字段。默认值为 1，不允许使用 0。</p>

属性	类型	描述
minDomains	整数	<p>MinDomains 表示合格的域数量。当具有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 的计算。当匹配拓扑键等于或大于 minDomains 的合格域数量时，这个值不会影响调度。因此，当有资格的域数量小于 minDomains 时，调度程序不会将超过 maxSkew Pod 的调度到这些域。如果值为 nil，则约束的行为就像 MinDomains 等于 1 一样。有效值为大于 0 的整数。如果值不是 nil，WhenUnsatisfiable 必须是 DoNotSchedule。</p> <p>例如，在一个 3 个区集群中，MaxSkew 设置为 2，MinDomains 被设置为 5，且带有与 2/2/2: zone1 zone2 zone2 zone2 zone3 P P P P P P P P P P P 数量的域的数量小于 5 (MinDomains)，因此"global minimum"被视为 0。在这种情况下，如果新 Pod 调度到任何三个区域，则无法调度具有相同 labelSelector 的新 pod，因为 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：只有与 nodeAffinity/nodeSelector 匹配的节点才会包含在计算中。- Ignore: nodeAffinity/nodeSelector 会被忽略。所有节点都包含在计算中。</p> <p>如果这个值是 nil，则行为等同于 Honor 策略。这是 NodeInclusionPolicyInPodTopologySpread 功能默认启用的 beta 级别功能。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布偏移时使用此调度指令 - "Ignore" 在计算 pod 拓扑分布偏移时忽略这个调度指令。</p>
nodeTaintsPolicy	string	<p>NodeTaintsPolicy 指示在计算 pod 拓扑分布偏移时如何对待节点污点。选项为：- Honor: 没有污点的节点，以及传入的 pod 具有容限的污点节点。- Ignore: node taint are ignored.所有节点都包括在内。</p> <p>如果这个值是 nil，则行为等同于 Ignore 策略。这是 NodeInclusionPolicyInPodTopologySpread 功能默认启用的 beta 级别功能。</p> <p>可能枚举值：- "Honor" 表示在计算 pod 拓扑分布偏移时使用此调度指令 - "Ignore" 在计算 pod 拓扑分布偏移时忽略这个调度指令。</p>

描述

可以通过属于 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

3.1.246. .template.spec.volumes[]**描述**

卷代表 pod 中的指定卷，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 中的持久磁盘资源。 在挂载到容器前，必须存在 AWS EBS 磁盘。磁盘也必须与 kubelet 位于同一个 AWS 区中。AWS EBS 磁盘只能挂载为读/写一次。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	代表持续 pod Cephfs 卷生命周期的 Ceph 文件系统挂载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。卷也必须与 kubelet 位于同一个区域。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，该文件使用 Data 字段中的密钥作为文件名，除非 items 元素使用到路径的特定映射填充。ConfigMap 卷支持所有权管理和 SELinux 重新标记。</p>
csi	object	代表要挂载的卷的源位置，由外部 CSI 驱动程序管理
downwardAPI	object	<p>DownwardAPIVolumeSource 代表包含 Downward API 信息的卷。</p> <p>Downward API 卷支持所有权管理和 SELinux 重新标记。</p>
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 卷只能挂载为读/写一次。Fibre Channel 卷支持所有权管理和 SELinux 重新标记。
flexVolume	对象	FlexVolume 代表使用基于 exec 的插件置备的/附加的通用卷资源。
flocker	object	代表 Flocker 代理挂载的 Flocker 卷。应该设置一个 datasetName 和 datasetUUID 之一。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>挂载到容器前，GCE PD 必须存在。磁盘还必须位于与 kubelet 相同的 GCE 项目和区中。GCE PD 只能挂载为读/写一次或多次只读。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 存储库卷不支持所有权管理。Git 存储库卷支持 SELinux 重新标记。 DEPRECATED: GitRepo 已被弃用。要使用 git 存储库置备容器，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	代表持续 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 卷只能挂载为读/写一次。ISCSI 卷支持所有权管理和 SELinux 重新标记。
名称	string	卷的名称。必须是 DNS_LABEL，且必须在 pod 中唯一。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	代表持续 pod 生命周期的 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 基本上是一个由其他人（系统）拥有的另一个类型的卷的打包程序。
photonPersistentDisk	object	代表 Photon Controller 持久磁盘资源。
portworxVolume	object	PortworxVolumeSource 代表 Portworx 卷资源。

属性	类型	描述
projected	object	代表投射卷源
quobyte	object	代表持续 pod 生命周期的 Quobyte 挂载。quobyte 卷不支持所有权管理或 SELinux 重新标记。
rbd	object	代表持续 pod 生命周期的 Rados 块设备挂载。RBD 卷支持所有权管理和 SELinux 重新标记。
scaleIO	object	ScaleIOVolumeSource 代表持久的 ScaleIO 卷
secret	object	将 Secret 调整为卷。 目标 Secret 的 Data 字段的内容将作为文件显示在卷中，该文件使用 Data 字段中的密钥作为文件名。机密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。

3.1.247. .template.spec.volumes[].awsElasticBlockStore

描述

代表 AWS 中的持久磁盘资源。

在挂载到容器前，必须存在 AWS EBS 磁盘。磁盘也必须与 kubelet 位于同一个 AWS 区中。AWS EBS 磁盘只能挂载为读/写一次。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
----	----	----

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，则默认使用卷名称来挂载。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 VolumeMount 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) 中持久磁盘资源的唯一 ID。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

3.1.248. .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
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 Name
diskURI	string	diskURI 是 blob 存储中的数据磁盘的 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"。
kind	string	kind 预期值为 Shared: 每个存储帐户 Dedicated 有多个 blob 磁盘：每个存储帐户 Managed: azure managed data disk (only on managed availability set). 默认为 shared 可能枚举值：- "Dedicated" - "Managed" - "Shared"
readOnly	布尔值	readonly Defaults to false (read/write)。此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。

3.1.249. .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	readOnly 默认为 false（读/写）。此处的 readOnly 将强制 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

3.1.250. .template.spec.volumes[].cephfs

描述

代表持续 pod Cephfs 卷生命周期的 Ceph 文件系统挂载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）	monitor 是必需的：monitor 是 Ceph 监视器更多信息的集合： 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 为 Optional: Defaults to false (read/write)。此处的 readOnly 将强制 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

属性	类型	描述
secretFile	string	secretfile 为 Optional: SecretFile 是用户密钥环的路径，默认为 /etc/ceph/user.secret 更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference 包含足够信息，供您同一命名空间中找到引用的对象。
user	string	User 是可选的：用户是 rados 用户名，默认为 admin 更多信息： https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

3.1.251. .template.spec.volumes[].cephfs.secretRef

描述

LocalObjectReference 包含足够信息，供您同一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

3.1.252. .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。卷也必须与 kubelet 位于同一个区域。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	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	readonly 默认为 false（读/写）。此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference 包含足够信息，供您同一命名空间中找到引用的对象。
volumeID	string	用于识别 cinder 中的卷的 VolumeID。更多信息： https://examples.k8s.io/mysql-cinder-pd/README.md

3.1.253. .template.spec.volumes[].cinder.secretRef

描述

LocalObjectReference 包含足够信息，供您同一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

3.1.254. .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，该文件使用 Data 字段中的密钥作为文件名，除非 items 元素使用到路径的特定映射填充。ConfigMap 卷支持所有权管理和 SELinux 重新标记。

类型

object

属性	类型	描述
defaultMode	整数	defaultMode 是可选的：默认用于在创建的文件上设置权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。默认值为 0644。路径中的目录不受此设置的影响。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
items	数组	如果未指定，则引用的 ConfigMap 的 Data 字段中每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 ConfigMap 中的键，则卷设置将失败，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。
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

3.1.255. .template.spec.volumes[].configMap.items

描述

如果未指定，则引用的 ConfigMap 的 Data 字段中每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 ConfigMap 中的键，则卷设置将失败，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。

类型

数组

3.1.256. .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	模式为可选：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	string	path 是将键映射到的文件的相对路径。可能不是绝对路径。不得包含路径元素 '..'。不得以字符串 '..' 开头。

3.1.257. .template.spec.volumes[].csi

描述

代表要挂载的卷的源位置，由外部 CSI 驱动程序管理

类型

object

必填

- driver

属性	类型	描述
driver	string	driver 是处理此卷的 CSI 驱动程序的名称。请参考您的 admin，了解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，则会将空值传递给关联的 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	object	LocalObjectReference 包含足够信息，供您同一命名空间中找到引用的对象。
readOnly	布尔值	readonly 为卷指定只读配置。默认为 false（读/写）。
volumeAttributes	对象（字符串）	volumeAttributes 存储传递给 CSI 驱动程序的驱动程序特定属性。有关支持的值，请参阅驱动程序的文档。

3.1.258. .template.spec.volumes[].csi.nodePublishSecretRef

描述

LocalObjectReference 包含足够信息，供您同一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

3.1.259. .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：默认在创建的文件上使用模式位。必须是 Optional: 模式位，用于默认对创建的文件设置权限。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。默认值为 0644。路径中的目录不受此设置的影响。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
items	array	items 是 Downward API 卷文件的列表
items[]	对象	DownwardAPIVolumeFile 代表创建包含 pod 字段的文件的信息

3.1.260. .template.spec.volumes[].downwardAPI.items

描述

items 是 Downward API 卷文件的列表

类型

数组

3.1.261. .template.spec.volumes[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表创建包含 pod 字段的文件的信息

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：用于设置此文件权限的模式位，必须是 0000 到 0777 之间的数值，或 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能是绝对的，也不能包含 '..' 路径。必须经过 utf-8 编码。相对路径的第一个项不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

3.1.262. .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	模式的版本是按术语编写的 FieldPath，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

3.1.263. .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

- resource

属性	类型	描述
containerName	字符串	容器名称：卷需要，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

3.1.264. .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 代表存储介质类型应支持这个目录。默认值为 ""，这意味着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 是这个 EmptyDir 卷所需的本地存储总量。大小限制也适用于内存介质。内存用量的最大使用量为此处指定的 SizeLimit 和 pod 中所有容器的内存限值总和。默认值为 nil，这意味着限制未定义。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#emptydir

3.1.265. .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。

3.1.266. .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	可能包含创建 PVC 时将复制到 PVC 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 Source 用于特定于供应商的属性

3.1.267. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

描述

PersistentVolumeClaimSpec 描述了存储设备的通用属性，并允许 Source 用于特定于供应商的属性

类型

object

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

属性	类型	描述
dataSource	对象	TypedLocalObjectReference 包含足够信息，以便您在同一命名空间中找到类型引用的对象。
dataSourceRef	object	<p>如果需要非空卷，则 <code>dataSourceRef</code> 指定要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 <code>PersistentVolumeClaim</code> 对象的任何对象。当指定对象的类型与某些安装的卷填充或动态置备程序匹配时，卷绑定才会成功。此字段将替换 <code>dataSource</code> 字段的功能，如果两个字段都不是空的，则它们必须具有相同的值。为了向后兼容，当命名空间没有在 <code>dataSourceRef</code> 中指定时，如果其中一个为空，则两个字段(<code>dataSource</code> 和 <code>dataSourceRef</code>)都会自动设置为相同的值，另一个是非空的。当在 <code>dataSourceRef</code> 中指定 <code>namespace</code> 时，<code>dataSource</code> 不设置为相同的值，必须为空。<code>dataSource</code> 和 <code>dataSourceRef</code> 之间有三个重要区别：</p> <ul style="list-style-type: none"> * While <code>dataSource</code> 只允许两种特定类型的对象，<code>dataSourceRef</code> 允许任何非核心对象，以及 <code>PersistentVolumeClaim</code> 对象。 * 虽然 <code>dataSource</code> 忽略了禁止的值（丢弃它们），<code>dataSourceRef</code> 保留所有值，并在指定禁止的值时生成错误。 * 虽然 <code>dataSource</code> 只允许本地对象，<code>dataSourceRef</code> 允许任何命名空间中的对象。(beta)使用此字段需要启用 <code>AnyVolumeDataSource</code> 功能门。(alpha)使用 <code>dataSourceRef</code> 的 <code>namespace</code> 字段需要启用 <code>CrossNamespaceVolumeDataSource</code> 功能门。
resources	对象	<code>ResourceRequirements</code> 描述了计算资源要求。
selector	LabelSelector	<code>selector</code> 是要考虑绑定的卷的标签查询。

属性	类型	描述
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 定义声明需要哪些卷类型。当未包含在 claim spec 中时，文件系统的值会被表示。 可能枚举值：- "Block" 表示卷不会格式化为文件系统，并保留原始块设备。- "Filesystem" 表示卷将被格式化为文件系统或使用文件系统格式化。
volumeName	string	volumeName 是对此声明支持的 PersistentVolume 的绑定引用。

3.1.268. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

描述

TypedLocalObjectReference 包含足够信息，以便您在同一命名空间中找到类型引用的对象。

类型

object

必填

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

属性	类型	描述
apiGroup	字符串	APIGroup 是所引用资源的组。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

3.1.269. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef

描述

如果需要非空卷，则 `dataSourceRef` 指定要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。当指定对象的类型与某些安装的卷填充或动态置备程序匹配时，卷绑定才会成功。此字段将替换 `dataSource` 字段的功能，如果两个字段都不是空的，则它们必须具有相同的值。为了向后兼容，当命名空间没有在 `dataSourceRef` 中指定时，如果其中一个为空，则两个字段(`dataSource` 和 `dataSourceRef`)都会自动设置为相同的值，另一个是非空的。当在 `dataSourceRef` 中指定 `namespace` 时，`dataSource` 不设置为相同的值，必须为空。`dataSource` 和 `dataSourceRef` 之间有三个重要区别：
 * While `dataSource` 只允许两种特定类型的对象，`dataSourceRef` 允许任何非核心对象，以及 `PersistentVolumeClaim` 对象。
 * 虽然 `dataSource` 忽略了禁止的值（丢弃它们），`dataSourceRef` 保留所有值，并在指定禁止的值时生成错误。
 * 虽然 `dataSource` 只允许本地对象，`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 是所引用资源的组。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	命名空间是被引用的资源的命名空间，请注意，当指定命名空间时，引用命名空间中需要一个 <code>gateway.networking.k8s.io/ReferenceGrant</code> 对象来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。

3.1.270. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

描述

`ResourceRequirements` 描述了计算资源要求。

类型

object

属性	类型	描述
声明	数组	claim 列出此容器使用的 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。 此字段不可变。它只能为容器设置。
claims[]	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/

3.1.271. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claim

描述

claim 列出此容器使用的 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。

此字段不可变。它只能为容器设置。

类型

数组

3.1.272. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.claim

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 中的一个条目的名称匹配。它允许容器内部使用该资源。

3.1.273. .template.spec.volumes[].fc

描述

代表光纤通道卷。Fibre Channel 卷只能挂载为读/写一次。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 为 Optional: Defaults to false (read/write)。此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。
targetWWNs	数组 (字符串)	targetWWNs 是可选 : FC 目标全球名称(WWN)
wwids	数组 (字符串)	wwids 可选 : FC 卷全局范围标识符(wwids) Either wwids 或 targetWWN 和 lun 的组合必须被设置, 但不能同时设置。

3.1.274. .template.spec.volumes[].flexVolume

描述

FlexVolume 代表使用基于 exec 的插件置备的/附加的通用卷资源。

类型

object

必填

- driver

属性	类型	描述
driver	string	driver 是用于此卷的驱动程序的名 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 认文件系统依赖于 FlexVolume 脚 本。
options	对象 (字符串)	选项为 Optional : 此字段包含额外 的命令选项 (若有)。
readOnly	布尔值	readonly 为 Optional: 默认为 false (读/写)。此处的 readOnly 将强 制 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够 信息, 供您同一命名空间中找到 引用的对象。

3.1.275. .template.spec.volumes[].flexVolume.secretRef

描述

LocalObjectReference 包含足够信息, 供您同一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

3.1.276. .template.spec.volumes[].flocker

描述

代表 Flocker 代理挂载的 Flocker 卷。应该设置一个 datasetName 和 datasetUUID 之一。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 是数据存储为 Flocker dataset 上的 metadata → name 的 Name, 应被视为已弃用
datasetUUID	string	datasetUUID 是 dataset 的 UUID。这是 Flocker 数据集的唯一标识符

3.1.277. .template.spec.volumes[].gcePersistentDisk

描述

代表 Google Compute Engine 中的 Persistent Disk 资源。

挂载到容器前, GCE PD 必须存在。磁盘还必须位于与 kubelet 相同的 GCE 项目和区中。GCE PD 只能挂载为读/写一次或多次只读。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, 则默认使用卷名称来挂载。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 VolumeMount 中的 ReadOnly 设置。默认为false。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

3.1.278. .template.spec.volumes[].gitRepo

描述

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

DEPRECATED: GitRepo 已被弃用。要使用 git 存储库置备容器，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，卷将在带有给定名称的子目录中包含 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。

3.1.279. .template.spec.volumes[].glusterfs

描述

代表持续 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 Glusterfs 拓扑的详细信息端点的名称。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 readOnly 将强制 Glusterfs 卷以只读权限挂载。默认为 false。更多信息： https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

3.1.280. .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，它将遵循指向实际路径的链接。更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath

属性	类型	描述
type	string	<p>HostPath Volume 默认为 "" 更多信息： https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>可能枚举值：- "" 用于向后兼容，如果给定路径上不存在 "BlockDevice" A 块设备，则必须存在于给定路径 - "Directory" A 字符设备，则给定路径 - "Directory" A 目录必须存在于给定路径中 - "DirectoryOrCreate" (如果给定路径中不存在)，则会根据需要创建空目录，并带有文件模式 0755，使用 Kubelet 相同的组和所有权。- "File" A file must the given path - "FileOrCreate" 如果给定路径中没有，则会根据需要创建一个空文件，且文件模式为 0644，具有与 Kubelet 相同的组和所有权。- "Socket" A UNIX socket 必须存在于给定路径中</p>

3.1.281. .template.spec.volumes[].iscsi

描述

代表 iSCSI 磁盘。iSCSI 卷只能挂载为读/写一次。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 会话 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 表示 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 (通常为 TCP 端口 860 和 3260)，则门户是 IP 或 ip_addr:port。
readOnly	布尔值	此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。默认为 false。
secretRef	object	LocalObjectReference 包含足够信息，供您同一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 (通常为 TCP 端口 860 和 3260)，则门户是 IP 或 ip_addr:port。

3.1.282. .template.spec.volumes[].iscsi.secretRef

描述

LocalObjectReference 包含足够信息，供您同一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

3.1.283. .template.spec.volumes[].nfs

描述

代表持续 pod 生命周期的 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

3.1.284. .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 基本上是一个由其他人（系统）拥有的另一个类型的卷的打包程序。

类型

object

必填

- **claimName**

属性	类型	描述
claimName	string	claimName 是与使用这个卷的 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。

3.1.285. .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

3.1.286. .template.spec.volumes[].portworxVolume

描述

PortworxVolumeSource 代表 Portworx 卷资源。

类型

object

必填

- **volumeID**

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	readonly 默认为 false（读/写）。此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。
volumeID	string	VolumeID 唯一标识 Portworx 卷

3.1.287. .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 是默认用于在创建的文件上设置权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。路径中的目录不受此设置的影响。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
sources	数组	sources 是卷投射列表
sources[]	对象	可投射以及其他支持的卷类型的预测

3.1.288. .template.spec.volumes[].projected.sources

描述

sources 是卷投射列表

类型

数组

3.1.289. .template.spec.volumes[].projected.sources[]

描述

可投射以及其他支持的卷类型的预测

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显示在项目中，使用 Data 字段中的键作为文件名，除非 items 元素使用到路径的特定映射填充。请注意，这与没有默认模式的 configmap 卷源相同。
downwardAPI	object	代表项目到投射卷的 Downward API 信息。请注意，这与没有默认模式的 downwardAPI 卷源相同。
secret	object	将 secret 调整为投射卷。 目标 Secret 的 Data 字段的内容将以文件形式显示，以文件形式使用 Data 字段中的键作为文件名。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。此预测可用于将服务帐户令牌插入到 pod 运行时文件系统中，以针对 API (Kubernetes API Server 或其他)。

3.1.290. .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显示在项目中，使用 Data 字段中的键作为文件名，除非 items 元素使用到路径的特定映射填充。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，则引用的 ConfigMap 的 Data 字段中每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 ConfigMap 中的键，则卷设置将失败，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。
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

3.1.291. .template.spec.volumes[].projected.sources[].configMap.items

描述

如果未指定，则引用的 ConfigMap 的 Data 字段中每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 ConfigMap 中的键，则卷设置将失败，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。

类型

数组

3.1.292. .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	模式为可选：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	string	path 是将键映射到的文件的相对路径。可能不是绝对路径。不得包含路径元素 '..'。不得以字符串 '..' 开头。

3.1.293. .template.spec.volumes[].projected.sources[].downwardAPI

描述

代表项目到投射卷的 Downward API 信息。请注意，这与没有默认模式的 downwardAPI 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 代表创建包含 pod 字段的文件的信息

3.1.294. .template.spec.volumes[].projected.sources[].downwardAPI.items

描述

items 是 DownwardAPIVolume 文件列表

类型

数组

3.1.295. .template.spec.volumes[].projected.sources[].downwardAPI.items[]

描述

DownwardAPIVolumeFile 代表创建包含 pod 字段的文件的信息

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：用于设置此文件权限的模式位，必须是 0000 到 0777 之间的数值，或 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	字符串	必需：Path 是要创建的文件的路径名称。不能是绝对的，也不能包含 '..' 路径。必须经过 utf-8 编码。路径的第一个项不能以 '..' 开头。
resourceFieldRef	object	ResourceFieldSelector 代表容器资源(cpu、memory)及其输出格式

3.1.296. .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	模式的版本是按术语编写的 FieldPath，默认为 "v1"。
fieldPath	字符串	在指定 API 版本中选择的字段路径。

3.1.297. .template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceF

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：卷需要，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

3.1.298. .template.spec.volumes[].projected.sources[].secret**描述**

将 secret 调整为投射卷。

目标 Secret 的 Data 字段的内容将以文件形式显示，以文件形式使用 Data 字段中的键作为文件名。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，则引用 Secret 的 Data 字段中的每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 Secret 中的密钥，则卷设置将出错，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。
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

属性	类型	描述
optional	布尔值	可选字段指定是否必须定义 Secret 或其密钥

3.1.299. .template.spec.volumes[].projected.sources[].secret.items

描述

如果未指定，则引用 Secret 的 Data 字段中的每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 Secret 中的密钥，则卷设置将出错，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。

类型

数组

3.1.300. .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	模式为可选：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	string	path 是将键映射到的文件的相对路径。可能不是绝对路径。不得包含路径元素 '..'。不得以字符串 '..' 开头。

3.1.301. .template.spec.volumes[].projected.sources[].serviceAccountToken

描述

ServiceAccountTokenProjection 代表投射服务帐户令牌卷。此预测可用于将服务帐户令牌插入到 pod 运行时文件系统中，以针对 API (Kubernetes API Server 或其他)。

类型

object

必填

- **path**

属性	类型	描述
受众	string	受众是令牌的预期使用者。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其自身，否则应拒绝令牌。受众默认为 apiserver 的标识符。
expirationSeconds	整数	expirationSeconds 是服务帐户令牌的请求的有效期。当令牌接近过期时，kubelet 卷插件会主动轮转服务帐户令牌。如果令牌的时间超过 80%，或者令牌早于 24 小时，则 kubelet 将开始尝试轮转令牌。默认为 1 小时，且必须至少为 10 分钟。
path	string	path 是与文件挂载点相关的路径，用于将令牌放入其中。

3.1.302. .template.spec.volumes[].quobyte**描述**

代表持续 pod 生命周期的 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

属性	类型	描述
readOnly	布尔值	此处的 readOnly 将强制使用只读权限挂载 Quobyte 卷。默认为 false。
registry	string	registry 代表单个或多个 Quobyte Registry 服务，它指定为 host:port 对（多个条目用逗号分开），它充当卷的中央 registry
tenant	string	在带有动态置备 Quobyte 卷的后端使用中拥有给定 Quobyte 卷的租户，由插件设置
user	string	用户，将卷访问权限映射到 Defaults 到 serviceaccount 用户
卷	string	volume 是按名称引用已创建 Quobyte 卷的字符串。

3.1.303. .template.spec.volumes[].rbd

描述

代表持续 pod 生命周期的 Rados 块设备挂载。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 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 包含足够信息，供您同一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

3.1.304. .template.spec.volumes[].rbd.secretRef

描述

LocalObjectReference 包含足够信息，供您同一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

3.1.305. .template.spec.volumes[].scaleIO

描述

ScaleIOVolumeSource 代表持久的 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 是配置存储的 ScaleIO 保护域的名称。
readOnly	布尔值	readonly Defaults to false (read/write)。此处的 readOnly 将强制 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够信息，供您在同一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 指示卷的存储是否应 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 ScaleIO 中配置的存储系统的名称。
volumeName	string	volumeName 是在与这个卷源关联的 ScaleIO 系统中创建的卷的名称。

3.1.306. .template.spec.volumes[].scaleIO.secretRef

描述

LocalObjectReference 包含足够信息，供您在同一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

3.1.307. .template.spec.volumes[].secret

描述

将 Secret 调整为卷。

目标 Secret 的 Data 字段的内容将作为文件显示在卷中，该文件使用 Data 字段中的密钥作为文件名。机密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：默认用于设置创建文件的权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。默认值为 0644。路径中的目录不受此设置的影响。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
items	数组	如果未指定项目，则引用 Secret 的 Data 字段中的每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 Secret 中的密钥，则卷设置将出错，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。
items[]	对象	将字符串键映射到卷中的路径。
optional	布尔值	可选字段指定是否必须定义 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

3.1.308. .template.spec.volumes[].secret.items

描述

如果未指定项目，则引用 Secret 的 Data 字段中的每个键值对将投射到卷中，其名称是键，内容是值。如果指定，列出的键将投射到指定的路径中，并且不会出现未列出的键。如果指定了没有存在于 Secret 中的密钥，则卷设置将出错，除非标记为可选。路径必须是相对的，且不能包含 '..' 路径，或者以 '..' 开始。

类型

数组

3.1.309. .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	模式为可选：用于设置此文件权限的模式位。必须是 0000 到 0777 之间的数值，或者 0 到 511 之间的十进制值。YAML 接受八进制和十进制值，JSON 需要模式位的十进制值。如果没有指定，则使用卷 defaultMode。这可能与影响文件模式的其他选项（如 fsGroup）冲突，结果可以是其他模式位。
path	string	path 是将键映射到的文件的相对路径。可能不是绝对路径。不得包含路径元素 '..'。不得以字符串 '..' 开头。

3.1.310. .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	readonly 默认为 false（读/写）。此处的 readonly 将强制 VolumeMount 中的 ReadOnly 设置。
secretRef	object	LocalObjectReference 包含足够信息，供您在同一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。卷名称仅在命名空间中唯一。
volumeNamespace	string	volumeNamespace 指定 StorageOS 中的卷范围。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 名称进行更紧密的集成。将 VolumeName 设置为任何名称，以覆盖默认的行为。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。

3.1.311. .template.spec.volumes[].storageos.secretRef

描述

LocalObjectReference 包含足够信息，供您在同一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

3.1.312. .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 关联的存储策略 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

3.2. 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' 参数及列表操作。
- **/api/v1/namespaces/{namespace}/podtemplates**
 - **DELETE**: 删除 PodTemplate 集合
 - **GET**: 列出或监视类型为 PodTemplate 的对象
 - **POST** : 创建一个 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' 参数及列表操作。
- **/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**: 观察对类型为 PodTemplate 的对象的更改。已弃用：使用带有 list 操作的 'watch' 参

数，而是过滤到带有 'fieldSelector' 参数的单个项目。

3.2.1. /api/v1/podtemplates

表 3.1. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

列出或监视类型为 PodTemplate 的对象

表 3.2. HTTP 响应

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

3.2.2. /api/v1/watch/podtemplates

表 3.3. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的值，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下, 观察流将以复合事件开头, 以生成集合中对象的当前状态。发送完所有这些事件后, 将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV), 并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后, 监视流将照常进行, 从而向监视的对象发送与更改对应的监视事件 (按 RV 排序)。 当设置 sendInitialEvents 选项时, 我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下: - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion ", 当状态同步到 ' resourceVersion 时, 如果状态同步到 ListOptions 提供的 ' resourceVersion 时, 会发送书签事件。如果未设置 resourceVersion , 则这被解释为"consistent read", 当请求开始处理时, 将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" (出于向后兼容原因) 和 false, 则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间, 无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改, 并将它们作为添加、更新和删除通知的流返回。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' 参数及列表操作。

表 3.4. HTTP 响应

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

3.2.3. /api/v1/namespaces/{namespace}/podtemplates

表 3.5. 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 3.6. 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 PodTemplate 集合

表 3.7. 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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</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>一个选择器，用于限制其字段返回的对象列表。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。值为零表示立即删除。如果这个值是 nil，则会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，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。</p>
propagationPolicy	字符串	<p>是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。

表 3.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 3.9. 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 PodTemplate 的对象

表 3.10. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。 当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下： - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion "，当状态同步到 'resourceVersion 时，如果状态同步到 ListOptions 提供的 'resourceVersion 时，会发送书签事件。如果未设置 resourceVersion ，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。 - resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" （出于向后兼容原因）和 false ，则默认为 true 。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

表 3.11. 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 3.12. 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 3.13. 主体参数

参数	类型	描述
正文 (body)	PodTemplate 模式	

表 3.14. 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

3.2.4. /api/v1/watch/namespaces/{namespace}/podtemplates

表 3.15. 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 3.16. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>

参数	类型	描述
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的值，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。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' 参数及列表操作。

表 3.17. HTTP 响应

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

3.2.5. /api/v1/namespaces/{namespace}/podtemplates/{name}

表 3.18. 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 3.19. 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 PodTemplate

表 3.20. 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。值为零表示立即删除。如果这个值是 nil，则会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	弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。

表 3.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 3.22. 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 3.23. 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 3.24. 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(application/apply-patch)需要此字段，但对非应用补丁类型(JsonPatch、MergePatch、MergePatch)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	强制要"强制"应用请求。这意味着用户将重新分配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 3.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 3.26. 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 3.27. 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 3.28. 主体参数

参数	类型	描述
正文 (body)	PodTemplate 模式	

表 3.29. HTTP 响应

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

3.2.6. /api/v1/watch/namespaces/{namespace}/podtemplates/{name}

表 3.30. 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 3.31. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下, 观察流将以复合事件开头, 以生成集合中对象的当前状态。发送完所有这些事件后, 将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV), 并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后, 监视流将照常进行, 从而向监视的对象发送与更改对应的监视事件 (按 RV 排序)。 当设置 sendInitialEvents 选项时, 我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下: - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion ", 当状态同步到 ' resourceVersion 时, 如果状态同步到 ListOptions 提供的 ' resourceVersion 时, 会发送书签事件。如果未设置 resourceVersion , 则这被解释为"consistent read", 当请求开始处理时, 将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" (出于向后兼容原因) 和 false, 则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间, 无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改, 并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 PodTemplate 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤成带有 'fieldSelector' 参数的单个项目。

表 3.32. HTTP 响应

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

第 4 章 TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]

描述

模板包含生成配置所需的输入。

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

- **objects**

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
labels	对象 (字符串)	labels 是一个可选的标签，在 Template 到 Config 转换过程中应用于每个对象。
message	string	Message 是可选的说明消息，将在此模板实例化时显示该消息。此字段应告知用户如何使用新创建的资源。在显示之前，将在消息上执行参数替换，以便输出中包含生成的凭证和其他参数。

属性	类型	描述
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
objects	数组(RawExtension)	对象是要包含在此模板中的一组资源。如果对象中硬编码了命名空间值，它将在模板实例化过程中被删除，但如果命名空间值是或包含，则 <code>\${PARAMETER_REFERENCE}</code> 将遵循参数替换后解析的值，并在该命名空间中创建对象。
parameters	数组	参数是模板期间使用的可选参数数组。
parameters[]	object	参数定义要在模板到配置转换期间处理的名称/值变量。

4.1.1. .parameters

描述

参数是模板期间使用的可选参数数组。

类型

数组

4.1.2. .parameters[]

描述

参数定义要在模板到配置转换期间处理的名称/值变量。

类型

object

必填

- name

属性	类型	描述
description	string	参数的描述。可选。
displayName	string	可选：在 UI 中显示的名称而不是参数 'Name'

属性	类型	描述
from	string	from 是生成器的输入值。可选。
generate	string	<p>generate 指定用于从 From 字段指定的输入值生成随机字符串的生成器。结果字符串存储在 Value 字段中。如果为空，则不使用生成器，从而使结果值保持不变。可选。</p> <p>唯一支持的生成器是 "expression"，它接受一个 "from" 值，格式为包含范围表达式 "[a-zA-Z0-9]"，长度表达式 "a{length}"。</p> <p>示例：</p> <pre>from value ----- ----- "test[0-9] -----x" "test7x" "[0-1]{8}" "01001100" "0x[A-F0-9]{4}" "0xB3AF" "[a-zA-Z0-9]{8}" "hW4yQU5i"</pre>
名称	string	必须设置 name，并且可以使用 <code>\${PARAMETER_NAME}</code> 在模板项目中引用它。必需。
required	布尔值	可选：指示参数必须具有值。默认为 false。
value	string	value 包含参数数据。如果指定了，则忽略生成器。该值替换模板到配置转换期间出现的所有 Parameter <code>\${Name}</code> 表达式。可选。

4.2. API 端点

可用的 API 端点如下：

- **/apis/template.openshift.io/v1/templates**
 - **GET**: 列出或监视类型为 Template 的对象
- **/apis/template.openshift.io/v1/watch/templates**
 - **GET**: 观察单个对模板列表的更改。已弃用：改为使用 'watch' 参数及列表操作。
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templates**
 - **DELETE**：删除模板集合
 - **GET**: 列出或监视类型为 Template 的对象

- **POST** : 创建模板
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates**
 - **GET**: 观察单个对模板列表的更改。已弃用 : 改为使用 'watch' 参数及列表操作。
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templates/{name}**
 - **DELETE** : 删除模板
 - **GET** : 读取指定的模板
 - **PATCH**: 部分更新指定的模板
 - **PUT** : 替换指定的模板
- **/apis/template.openshift.io/v1/namespaces/{namespace}/processedtemplates**
 - **POST** : 创建模板
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates/{name}**
 - **GET**: 观察对类型为 Template 的对象的更改。已弃用 : 使用带有 list 操作的 'watch' 参数, 而是过滤成带有 'fieldSelector' 参数的单个项目。

4.2.1. /apis/template.openshift.io/v1/templates

表 4.1. 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。不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, 它必须重启其列表, 而无需 continue 字段。否则, 客户端可能会发送另一个带有 410 错误接收的令牌的列表请求, 服务器将以从下一个密钥开始开始的列表响应, 但从最新的快照开始, 这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中, 只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视, 而不丢失任何修改。</p>

参数	类型	描述
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

列出或监视类型为 Template 的对象

表 4.2. HTTP 响应

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

4.2.2. /apis/template.openshift.io/v1/watch/templates

表 4.3. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下, 观察流将以复合事件开头, 以生成集合中对象的当前状态。发送完所有这些事件后, 将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV), 并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后, 监视流将照常进行, 从而向监视的对象发送与更改对应的监视事件 (按 RV 排序)。 当设置 sendInitialEvents 选项时, 我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下: - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion ", 当状态同步到 resourceVersion 时, 如果状态同步到 ListOptions 提供的 resourceVersion 时, 会发送书签事件。如果未设置 resourceVersion , 则这被解释为"consistent read", 当请求开始处理时, 将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" (出于向后兼容原因) 和 false , 则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间, 无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改, 并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

HTTP 方法
GET

描述

观察单个对模板列表的更改。已弃用：改为使用 'watch' 参数及列表操作。

表 4.4. HTTP 响应

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

4.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templates

表 4.5. 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 4.6. 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

删除模板集合

表 4.7. 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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</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>一个选择器，用于限制其字段返回的对象列表。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。值为零表示立即删除。如果这个值是 nil，则会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，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	<p>弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。</p>
propagationPolicy	字符串	<p>是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion 时，如果状态同步到 ListOptions 提供的 'resourceVersion 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。

表 4.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 4.9. 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 **Template** 的对象

表 4.10. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。 当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下： - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion "，当状态同步到 resourceVersion 时，如果状态同步到 ListOptions 提供的 resourceVersion 时，会发送书签事件。如果未设置 resourceVersion ，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。 - resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" （出于向后兼容原因）和 false ，则默认为 true 。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

表 4.11. HTTP 响应

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

HTTP 方法

POST

描述

创建模板

表 4.12. 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 4.13. 主体参数

参数	类型	描述
正文 (body)	Template 模式	

表 4.14. HTTP 响应

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

4.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates

表 4.15. 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 4.16. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>

参数	类型	描述
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的值，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

观察单个对模板列表的更改。已弃用：改为使用 'watch' 参数及列表操作。

表 4.17. HTTP 响应

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

4.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templates/{name}

表 4.18. 全局路径参数

参数	类型	描述
name	string	模板的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 4.19. 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

删除模板

表 4.20. 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。值为零表示立即删除。如果这个值是 nil，则会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	弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。

表 4.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 4.22. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Template</code> 模式
202 - Accepted	<code>Template</code> 模式
401 - Unauthorized	空

HTTP 方法

GET

描述

读取指定的模板

表 4.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>Template</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的模板

表 4.24. 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(application/apply-patch)需要此字段，但对非应用补丁类型(JsonPatch、MergePatch、MergePatch)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	强制要"强制"应用请求。这意味着用户将重新分配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 4.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 4.26. HTTP 响应

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

HTTP 方法

PUT

描述

替换指定的模板

表 4.27. 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 4.28. 主体参数

参数	类型	描述
正文 (body)	Template 模式	

表 4.29. HTTP 响应

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

4.2.6. /apis/template.openshift.io/v1/namespaces/{namespace}/processedtemplates

表 4.30. 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 4.31. 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

创建模板

表 4.32. 主体参数

参数	类型	描述
正文 (body)	Template 模式	

表 4.33. HTTP 响应

HTTP 代码	响应正文
200 - OK	Template 模式
201 - Created	Template 模式
202 - Accepted	Template 模式

HTTP 代码	响应正文
401 - Unauthorized	空

4.2.7. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates/{n

表 4.34. 全局路径参数

参数	类型	描述
name	string	模板的名称
namespace	字符串	对象名称和身份验证范围，如团队（team）和项目（project）

表 4.35. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

观察对类型为 Template 的对象的更改。已弃用：使用带有列表操作的 'watch' 参数，而是过滤成带有 'fieldSelector' 参数的单个项目。

表 4.36. HTTP 响应

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

第 5 章 TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

描述

TemplateInstance 请求并记录模板的实例化。TemplateInstance 是实验性 API 的一部分。
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**

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	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	object	TemplateInstanceSpec 描述了 TemplateInstance 的所需状态。
status	object	TemplateInstanceStatus 描述了 TemplateInstance 的当前状态。

5.1.1. .spec

描述

TemplateInstanceSpec 描述了 TemplateInstance 的所需状态。

类型

object

必填

- 模板

属性	类型	描述
requester	object	TemplateInstanceRequester 包含请求模板实例化的代理的身份。
secret	LocalObjectReference	Secret 是引用包含所需模板参数的 Secret 对象。
模板	object	模板包含生成配置所需的输入。 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）。

5.1.2. .spec.requester

描述

TemplateInstanceRequester 包含请求模板实例化的代理的身份。

类型

object

属性	类型	描述
extra	object	额外包含验证器提供的其他信息。
extra{}	数组（字符串）	
groups	数组（字符串）	组代表此用户所属的组。
uid	string	UID 是一个在一段时间内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，则它们具有不同的 UID。
username	string	在所有活动用户中唯一标识此用户的用户名。

5.1.3. .spec.requester.extra

描述

额外包含验证器提供的其他信息。

类型

object

5.1.4. .spec.template

描述

模板包含生成配置所需的输入。

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

- **objects**

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
labels	对象 (字符串)	labels 是一个可选的标签，在 Template 到 Config 转换过程中应用于每个对象。

属性	类型	描述
message	string	Message 是可选的说明消息，将在此模板实例化时显示该消息。此字段应告知用户如何使用新创建的资源。在显示之前，将在消息上执行参数替换，以便输出中包含生成的凭证和其他参数。
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
objects	数组(RawExtension)	对象是要包含在此模板中的一组资源。如果对象中硬编码了命名空间值，它将在模板实例化过程中被删除，但如果命名空间值是或包含，则 <code>_\${PARAMETER_REFERENCE}</code> 将遵循参数替换后解析的值，并在该命名空间中创建对象。
parameters	数组	参数是模板期间使用的可选参数数组。
parameters[]	object	参数定义要在模板到配置转换期间处理的名称/值变量。

5.1.5. .spec.template.parameters

描述

参数是模板期间使用的可选参数数组。

类型

数组

5.1.6. .spec.template.parameters[]

描述

参数定义要在模板到配置转换期间处理的名称/值变量。

类型

object

必填

- **name**

属性	类型	描述
description	string	参数的描述。可选。
displayName	string	可选：在 UI 中显示的名称而不是参数 'Name'
from	string	from 是生成器的输入值。可选。
generate	string	<p>generate 指定用于从 From 字段指定的输入值生成随机字符串的生成器。结果字符串存储在 Value 字段中。如果为空，则不使用生成器，从而使结果值保持不变。可选。</p> <p>唯一支持的生成器是 "expression"，它接受一个 "from" 值，格式为包含范围表达式 "[a-zA-Z0-9]"，长度表达式 "a{length}"。</p> <p>示例：</p> <pre>from value ----- ----- "test[0-9] -----x" "test7x" "[0-1]{8}" "01001100" "0x[A-F0-9]{4}" "0xB3AF" "[a-zA-Z0-9]{8}" "hW4yQU5i"</pre>
名称	string	必须设置 name，并且可以使用 <code>#{PARAMETER_NAME}</code> 在模板项目中引用它。必需。
required	布尔值	可选：指示参数必须具有值。默认为 false。
value	string	value 包含参数数据。如果指定了，则忽略生成器。该值替换模板到配置转换期间出现的所有 Parameter <code>#{Name}</code> 表达式。可选。

5.1.7. .status

描述

TemplateInstanceStatus 描述了 TemplateInstance 的当前状态。

类型

object

属性	类型	描述
conditions	数组	条件代表 TemplateInstance 当前状态的最新可用影响。
conditions[]	object	TemplateInstanceCondition 包含 TemplateInstance 的条件信息。
objects	数组	对象引用 TemplateInstance 创建的对象。
objects[]	object	TemplateInstanceObject 引用 TemplateInstance 创建的对象。

5.1.8. .status.conditions

描述

条件代表 TemplateInstance 当前状态的最新可用影响。

类型

array

5.1.9. .status.conditions[]

描述

TemplateInstanceCondition 包含 TemplateInstance 的条件信息。

类型

object

必填

- **type**
- **status**
- **lastTransitionTime**
- **reason**
- **message**

属性	类型	描述
lastTransitionTime	时间	lastTransitionTime 是条件状态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 是最后一次转换的详细信息的人类可读描述，补充原因。
reason	字符串	原因是对条件最后一次转换的简短机器可读说明。
status	string	条件的状态，True、False 或 Unknown 之一。
type	string	条件的类型，当前 Ready 或 InstantiateFailure。

5.1.10. .status.objects

描述

对象引用 TemplateInstance 创建的对象。

类型

数组

5.1.11. .status.objects[]

描述

TemplateInstanceObject 引用 TemplateInstance 创建的对象。

类型

object

属性	类型	描述
ref	ObjectReference	ref 是对创建的对象引用。在 .spec 下使用时，只使用 name 和 namespace；它们可以包含对参数的引用，这些参数将按照常规规则替换。

5.2. API 端点

可用的 API 端点如下：

- **/apis/template.openshift.io/v1/templateinstances**
 - **GET**: 列出或监视类型为 TemplateInstance 的对象
- **/apis/template.openshift.io/v1/watch/templateinstances**
 - **GET**: 观察单个对 TemplateInstance 列表的更改。已弃用：改为使用 'watch' 参数及列表操作。

- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances**
 - **DELETE:** 删除 TemplateInstance 集合
 - **GET:** 列出或监视类型为 TemplateInstance 的对象
 - **POST :** 创建 TemplateInstance
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances**
 - **GET:** 观察单个对 TemplateInstance 列表的更改。已弃用：改为使用 'watch' 参数及列表操作。
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}**
 - **DELETE :** 删除 TemplateInstance
 - **GET :** 读取指定的 TemplateInstance
 - **PATCH:** 部分更新指定的 TemplateInstance
 - **PUT :** 替换指定的 TemplateInstance
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances/{name}**
 - **GET:** 观察对类型为 TemplateInstance 的对象的更改。已弃用：使用带有 list 操作的 'watch' 参数，而是过滤成带有 'fieldSelector' 参数的单个项目。
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}/status**
 - **GET :** 指定 TemplateInstance 的读取状态
 - **PATCH:** 部分更新指定 TemplateInstance 的状态
 - **PUT :** 替换指定 TemplateInstance 的状态

5.2.1. /apis/template.openshift.io/v1/templateinstances

表 5.1. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。 当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下： - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion "，当状态同步到 resourceVersion 时，如果状态同步到 ListOptions 提供的 resourceVersion 时，会发送书签事件。如果未设置 resourceVersion ，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。 - resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" （出于向后兼容原因）和 false ，则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

HTTP 方法

GET

描述

列出或监视类型为 `TemplateInstance` 的对象

表 5.2. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>TemplateInstanceList</code> 模式
401 - Unauthorized	空

5.2.2. /apis/template.openshift.io/v1/watch/templateinstances

表 5.3. 全局查询参数

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。不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，它必须重启其列表，而无需 <code>continue</code> 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 <code>watch</code> 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，而不丢失任何修改。</p>
<code>fieldSelector</code>	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
<code>labelSelector</code>	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

观察单个对 `TemplateInstance` 列表的更改。已弃用：改为使用 'watch' 参数及列表操作。

表 5.4. HTTP 响应

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

5.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances

表 5.5. 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 5.6. 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 TemplateInstance 集合

表 5.7. 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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</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	一个选择器，用于限制其字段返回的对象列表。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。值为零表示立即删除。如果这个值是 nil，则会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
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，则返回第一个列表结果时存在的对象版本。</p>
orphanDependents	布尔值	弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。
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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。

表 5.8. 主体参数

参数	类型	描述
正文 (body)	DeleteOptions 模式	

表 5.9. 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 `TemplateInstance` 的对象

表 5.10. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。 当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下： - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion "，当状态同步到 resourceVersion 时，如果状态同步到 ListOptions 提供的 resourceVersion 时，会发送书签事件。如果未设置 resourceVersion ，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。 - resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" （出于向后兼容原因）和 false ，则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

表 5.11. HTTP 响应

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

HTTP 方法

POST

描述

创建 TemplateInstance

表 5.12. 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 5.13. 主体参数

参数	类型	描述
正文 (body)	TemplateInstance 模式	

表 5.14. HTTP 响应

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

5.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinst:

表 5.15. 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 5.16. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>

参数	类型	描述
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的值，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下，观察流将以复合事件开头，以生成集合中对象的当前状态。发送完所有这些事件后，将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV)，并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后，监视流将照常进行，从而向监视的对象发送与更改对应的监视事件（按 RV 排序）。</p> <p>当设置 sendInitialEvents 选项时，我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下：- resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion"，当状态同步到 'resourceVersion' 时，如果状态同步到 ListOptions 提供的 'resourceVersion' 时，会发送书签事件。如果未设置 resourceVersion，则这被解释为"consistent read"，当请求开始处理时，将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。</p> <p>如果 resourceVersion="" 或 resourceVersion="0"（出于向后兼容原因）和 false，则默认为 true。</p>
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间，无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改，并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion 。

HTTP 方法

GET

描述

观察单个对 `TemplateInstance` 列表的更改。已弃用：改为使用 'watch' 参数及列表操作。

表 5.17. HTTP 响应

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

5.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{

表 5.18. 全局路径参数

参数	类型	描述
name	string	TemplateInstance 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 5.19. 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 TemplateInstance

表 5.20. 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。值为零表示立即删除。如果这个值是 nil，则会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	弃用：请使用 PropagationPolicy，此字段将在 1.7 中弃用。如果依赖对象是孤立的。如果为 true/false，则 "orphan" finalizer 将添加到对象的终结器列表中。可以设置此字段或 PropagationPolicy，但不能同时设置。
propagationPolicy	字符串	是否执行垃圾回收。可以设置此字段或 OrphanDependents，但不能同时设置两者。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。可接受的值有：'Orphan' - 孤立的依赖项；'Background' - 允许垃圾收集器删除依赖项，'Foreground' - 删除前台所有依赖的级联策略。

表 5.21. 主体参数

参数	类型	描述
正文 (body)	<code>DeleteOptions</code> 模式	

表 5.22. 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 `TemplateInstance`

表 5.23. HTTP 响应

HTTP 代码	响应正文
200 - OK	<code>TemplateInstance</code> 模式
401 - Unauthorized	空

HTTP 方法

PATCH

描述

部分更新指定的 `TemplateInstance`

表 5.24. 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(application/apply-patch)需要此字段，但对非应用补丁类型(JsonPatch、MergePatch、MergePatch)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	强制要"强制"应用请求。这意味着用户将重新分配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 5.25. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 5.26. HTTP 响应

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

HTTP 方法

PUT

描述

替换指定的 TemplateInstance

表 5.27. 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 5.28. 主体参数

参数	类型	描述
正文 (body)	TemplateInstance 模式	

表 5.29. HTTP 响应

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

5.2.6. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstanc

表 5.30. 全局路径参数

参数	类型	描述
name	string	TemplateInstance 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 5.31. 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。不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，它必须重启其列表，而无需 continue 字段。否则，客户端可能会发送另一个带有 410 错误接收的令牌的列表请求，服务器将以从下一个密钥开始开始的列表响应，但从最新的快照开始，这与上一个列表结果中不一致 - 在第一个列表请求后将包含在响应中，只要它们的密钥位于 "next key" 后。</p> <p>当 watch 为 true 时不支持此字段。客户端可以从服务器返回的最后一个 resourceVersion 值开始监视，而不丢失任何修改。</p>
fieldSelector	字符串	一个选择器，用于限制其字段返回的对象列表。默认为任何内容。
labelSelector	字符串	一个选择器，用于限制其标签返回的对象列表。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。如果过滤所有请求的对象，则设置限制可能会返回少于请求项目数量（最多为零项），并且客户端只应使用 continue 字段的存在，以确定是否有更多结果可用。服务器可能会选择不支持 limit 参数，并返回所有可用的结果。如果指定了 limit，并且 continue 字段为空，客户端可能会假定没有更多结果可用。如果 watch 为 true，则不支持此字段。</p> <p>服务器保证，在使用时返回的对象将与在没有限制限制的情况下发出单个列表调用相同 - 也就是说，在发出第一个请求后没有创建、修改或删除对象。这有时被称为一致的快照，并确保使用限制来接收非常大结果的较小的块的客户端可确保它们看到所有可能的对象。如果在块列表中更新对象，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 决定如何将 resourceVersion 应用到列表调用。强烈建议您为设置 resourceVersion 的 list 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 可以和 watch=true 一起设置。在这种情况下, 观察流将以复合事件开头, 以生成集合中对象的当前状态。发送完所有这些事件后, 将发送合成"Bookmark"事件。书签将报告与对象集合对应的 ResourceVersion (RV), 并使用 "k8s.io/initial-events-end": "true" 注解进行标记。之后, 监视流将照常进行, 从而向监视的对象发送与更改对应的监视事件 (按 RV 排序)。 当设置 sendInitialEvents 选项时, 我们还需要设置 resourceVersionMatch 选项。监控请求的语义如下: - resourceVersionMatch = NotOlderThan 解释为"数据至少为提供的 resourceVersion ", 当状态同步到 resourceVersion 时, 如果状态同步到 ListOptions 提供的 resourceVersion 时, 会发送书签事件。如果未设置 resourceVersion , 则这被解释为"consistent read", 当请求开始处理时, 将发送书签事件。- resourceVersionMatch 设置为任何其他值或未设置 Invalid 错误。 如果 resourceVersion="" 或 resourceVersion="0" (出于向后兼容原因) 和 false , 则默认为 true。
timeoutSeconds	整数	列表/watch 调用的超时。这限制了调用的持续时间, 无论任何活动或不活跃。
watch	布尔值	观察对所描述资源的更改, 并将它们作为添加、更新和删除通知的流返回。指定 resourceVersion。

HTTP 方法

GET

描述

观察对类型为 `TemplateInstance` 的对象的更改。已弃用：使用带列表操作的 'watch' 参数，而是过滤成带有 'fieldSelector' 参数的单个项目。

表 5.32. HTTP 响应

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

5.2.7. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{

表 5.33. 全局路径参数

参数	类型	描述
name	string	TemplateInstance 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 5.34. 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 `TemplateInstance` 的读取状态

表 5.35. HTTP 响应

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

HTTP 方法

PATCH

描述

部分更新指定 `TemplateInstance` 的状态

表 5.36. 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(application/apply-patch)需要此字段，但对非应用补丁类型(JsonPatch、MergePatch、MergePatch)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	强制要"强制"应用请求。这意味着用户将重新分配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 5.37. 主体参数

参数	类型	描述
正文 (body)	Patch 模式	

表 5.38. HTTP 响应

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

HTTP 方法

PUT

描述

替换指定 TemplateInstance 的状态

表 5.39. 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。
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 5.40. 主体参数

参数	类型	描述
正文 (body)	TemplateInstance 模式	

表 5.41. HTTP 响应

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

