



Red Hat Advanced Cluster Management for Kubernetes 2.10

API

API

法律通告

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

查看可用于创建和管理应用程序资源、频道、订阅和查询信息的 API 列表。

目录

第1章 API	3
1.1. CLUSTERS API	3
1.2. CLUSTERSETS API (VIBETA2)	9
1.3. CLUSTERSETBINDINGS API (VIBETA2)	14
1.4. CLUSTERVIEW API (V1ALPHA1)	19
1.5. CHANNELS API	25
1.6. SUBSCRIPTIONS API	33
1.7. PLACEMENTRULES API (已弃用)	43
1.8. APPLICATIONS API	50
1.9. HELM API	61
1.10. POLICY API	69
1.11. OBSERVABILITY API	77
1.12. 搜索查询 API	87
1.13. MULTICLUSTERHUB API	90
1.14. 放置 API (VIBETA1)	105
1.15. PLACEMENTDECISIONS API (VIBETA1)	112
1.16. DISCOVERYCONFIG API	117
1.17. DISCOVEREDCLUSTER API	123
1.18. ADDONDEPLOYMENTCONFIG API (V1ALPHA1)	131
1.19. CLUSTERMANAGEMENTADDON API (V1ALPHA1)	136
1.20. MANAGEDCLUSTERADDON API (V1ALPHA1)	142
1.21. MANAGEDCLUSTERSET API (VIBETA2)	148
1.22. KLUSTERLETCONFIG API (V1ALPHA1)	153
1.23. 策略合规历史记录 (技术预览)	162

第 1 章 API

您可以使用 API 来创建和管理应用程序资源、频道、订阅和查询信息。

用户需要的访问权限：您只能执行已分配角色的操作。了解 [基于角色的访问控制](#) 文档中的访问要求。

您还可以从集成控制台访问所有 API。在 **local-cluster** 视图中，进入到 **Home > API Explorer** 以探索 API 组。

如需更多信息，请参阅以下每个资源的 API 文档：

- [Clusters API](#)
- [ClusterSets API \(v1beta2\)](#)
- [ClusterSetBindings API \(v1beta2\)](#)
- [Channels API](#)
- [Subscriptions API](#)
- [PlacementRules API \(已弃用\)](#)
- [Applications API](#)
- [Helm API](#)
- [Policy API](#)
- [Observability API](#)
- [搜索查询 API](#)
- [MultiClusterHub API](#)
- [Placements API \(v1beta1\)](#)
- [PlacementDecisions API \(v1beta1\)](#)
- [DiscoveryConfig API](#)
- [DiscoveredCluster API](#)
- [AddOnDeploymentConfig API \(v1alpha1\)](#)
- [ClusterManagementAddOn API \(v1alpha1\)](#)
- [ManagedClusterAddOn API \(v1alpha1\)](#)
- [ManagedClusterSet API](#)
- [KlusterletConfig API \(v1alpha1\)](#)
- [策略合规 API \(技术预览\)](#)

1.1. CLUSTERS API

1.1.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的集群资源相关的 API 信息。集群资源有 4 个可用的请求：create、query、delete 和 update。**ManagedCluster** 代表受管集群的所需状态和当前状态。**ManagedCluster** 是一个集群范围的资源。

1.1.1.1. 版本信息

版本：2.10.0

1.1.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.1.1.3. Tags

- cluster.open-cluster-management.io：创建和管理集群

1.1.2. 路径

1.1.2.1. 查询所有集群

GET /cluster.open-cluster-management.io/v1/managedclusters

1.1.2.1.1. 描述

查询集群以获取更多详细信息。

1.1.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.1.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.1.2.1.4. 使用

- **cluster/yaml**

1.1.2.1.5. Tags

- cluster.open-cluster-management.io

1.1.2.2. 创建集群

POST /cluster.open-cluster-management.io/v1/managedclusters

1.1.2.2.1. 描述

创建集群

1.1.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建集群的参数。	Cluster

1.1.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.1.2.2.4. 使用

- **cluster/yaml**

1.1.2.2.5. Tags

- cluster.open-cluster-management.io

1.1.2.2.6. HTTP 请求示例

1.1.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1",
  "kind": "ManagedCluster",
  "metadata": {
    "labels": {
      "vendor": "OpenShift"
    },
    "name": "cluster1"
  },
  "spec": {
    "hubAcceptsClient": true,
    "managedClusterClientConfigs": [
      {
        "caBundle": "test",
        "url": "https://test.com"
      }
    ]
  },
  "status": {}
}
```

1.1.2.3. 查询单个集群

```
GET /cluster.open-cluster-management.io/v1/managedclusters/{cluster_name}
```

1.1.2.3.1. 描述

查询单个集群以获取更多详细信息。

1.1.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	cluster_name <i>必需</i>	要查询的集群的名称。	字符串

1.1.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.1.2.3.4. Tags

- cluster.open-cluster-management.io

1.1.2.4. 删除集群

DELETE /cluster.open-cluster-management.io/v1/managedclusters/{cluster_name}

DELETE /hive.openshift.io/v1/{cluster_name}/clusterdeployments/{cluster_name}

1.1.2.4.1. 描述

删除单个集群

1.1.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	cluster_name 必需	要删除的集群的名称。	字符串

1.1.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容

HTTP 代码	描述	模式
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.1.2.4.4. Tags

- cluster.open-cluster-management.io

1.1.3. 定义

1.1.3.1. Cluster

Name	描述	模式
apiVersion <i>必需</i>	ManagedCluster 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	ManagedCluster 的元数据。	对象
spec <i>必需</i>	ManagedCluster 的规格。	spec

spec

Name	描述	模式
hubAcceptsClient <i>必需</i>	指定 hub 是否可以与受管集群中的 klusterlet 代理建立连接。默认值为 false ，且只能在 hub 集群中配置了 RBAC 规则时更改为 true ，供您为 managedclusters/accept 的虚拟子资源进行更新。	bool
managedClusterClientConfigs <i>可选</i>	列出受管集群的 apiserver 地址。	managedClusterClientConfigs 数组

Name	描述	模式
leaseDurationSeconds 可选	指定受管集群中 klusterlet 代理的租期更新时间间隔。默认情况下，Klusterlet 代理每 60 秒更新其租期。	integer (int32)
taint 可选	防止在调度过程中将受管集群分配给一个或多个受管集群集。	污点 数组

managedClusterClientConfigs

Name	描述	模式
URL 必需		字符串
CABundle 可选	Pattern : <pre>"^(?:[A-Za-z0-9+]{4})*(?:[A-Za-z0-9+]{2}=[A-Za-z0-9+]{3}=)?\$" </pre>	字符串 (字节)

taint

Name	描述	模式
key 必需	应用到集群的污点键。	字符串
value 可选	与污点键对应的污点值。	字符串
effect 可选	污点对于不容许污点的放置的影响。有效值为 NoSelect 、 PreferNoSelect 和 NoSelectIfNew 。	字符串

1.2. CLUSTERSETS API (V1BETA2)

1.2.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 ClusterSet 资源相关的 API 信息。ClusterSet 资源有 4 个可用的请求：create、query、delete 和 update。ManagedClusterSet 定义了一组 ManagedClusters。您可以通过在引用 ManagedClusterSet 的 ManagedCluster 上添加名为 **cluster.open-cluster-management.io/clusterSet** 的标签，将 ManagedCluster 分配给特定的

ManagedClusterSet。当您有一个 RBAC 规则时，您只能在 ManagedCluster 上添加或删除此标签，允许对 **managedclustersets/join** 的虚拟子资源创建权限。您必须在源和目标 ManagedClusterSets 上具有此权限才能更新此标签。

1.2.1.1. 版本信息

版本: 2.10.0

1.2.1.2. URI scheme

BasePath: /kubernetes/apis

Schemes: HTTPS

1.2.1.3. Tags

- cluster.open-cluster-management.io : 创建和管理 Clustersets

1.2.2. 路径

1.2.2.1. 查询所有集群集 (clusterset)

GET /cluster.open-cluster-management.io/v1beta2/managedclustersets

1.2.2.1.1. 描述

查询 Clustersets 以获取更多详细信息。

1.2.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证: Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.2.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.2.2.1.4. 使用

- **clusterset/yaml**

1.2.2.1.5. Tags

- cluster.open-cluster-management.io

1.2.2.2. 创建一个 clusterset

POST /cluster.open-cluster-management.io/v1beta2/managedclustersets

1.2.2.2.1. 描述

创建 Clusterset。

1.2.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 clusterset 的参数。	Clusterset

1.2.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.2.2.2.4. 使用

- **clusterset/yaml**

1.2.2.2.5. Tags

- cluster.open-cluster-management.io

1.2.2.2.6. HTTP 请求示例

1.2.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1beta2",
  "kind": "ManagedClusterSet",
  "metadata": {
    "name": "clusterset1"
  },
  "spec": {
    "clusterSelector": {
      "selectorType": "ExclusiveClusterSetLabel"
    }
  },
  "status": {}
}
```

1.2.2.3. 查询单个集群集

```
GET /cluster.open-cluster-management.io/v1beta2/managedclustersets/{clusterset_name}
```

1.2.2.3.1. 描述

查询单个集群集以获取更多详细信息。

1.2.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	clusterset_name <i>必需</i>	要查询的集群集的名称。	字符串

1.2.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.2.2.3.4. Tags

- cluster.open-cluster-management.io

1.2.2.4. 删除集群集

DELETE /cluster.open-cluster-management.io/v1beta2/managedclustersets/{clusterset_name}

1.2.2.4.1. 描述

删除单个集群集。

1.2.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	clusterset_name 必需	要删除的集群集的名称。	字符串

1.2.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.2.2.4.4. Tags

- cluster.open-cluster-management.io

1.2.3. 定义

1.2.3.1. Clusterset

Name	模式
apiVersion 必需	字符串
kind 必需	字符串
metadata 必需	对象

1.3. CLUSTERSETBINDINGS API (V1BETA2)

1.3.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 ClusterSetBinding 资源相关的 API 信息。ClusterSetBinding 资源有 4 个可用的请求：create、query、delete 和 update。ManagedClusterSetBinding 将 ManagedClusterSet 项目到一个特定命名空间中。在一个命名空间中创建一个 ManagedClusterSetBinding，如果您有一个 RBAC 规则，允许您在 **managedclustersets/bind** 的虚拟子资源上创建它。

1.3.1.1. 版本信息

版本：2.10.0

1.3.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.3.1.3. Tags

- cluster.open-cluster-management.io：创建和管理 clustersetbindings

1.3.2. 路径

1.3.2.1. 查询所有 clustersetbindings

```
GET /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclustersetbindings
```

1.3.2.1.1. 描述

查询 clustersetbindings 以获取更多详细信息。

1.3.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.3.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.3.2.1.4. 使用

- **clustersetbinding/yaml**

1.3.2.1.5. Tags

- cluster.open-cluster-management.io

1.3.2.2. 创建 clustersetbinding

POST /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclustersetbindings

1.3.2.2.1. 描述

创建 clustersetbinding。

1.3.2.2.2. 参数

类型	Name	描述	模式
----	------	----	----

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default 。	字符串
Body	body <i>必需</i>	描述要创建的 clustersetbinding 的参数。	Clustersetbinding

1.3.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.3.2.2.4. 使用

- **clustersetbinding/yaml**

1.3.2.2.5. Tags

- `cluster.open-cluster-management.io`

1.3.2.2.6. HTTP 请求示例

1.3.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1beta2",
  "kind": "ManagedClusterSetBinding",
  "metadata": {
    "name": "clusterset1",
    "namespace": "ns1"
  },
  "spec": {
    "clusterSet": "clusterset1"
  }
}
```

```

    },
    "status" : {}
  }
}

```

1.3.2.3. 查询单个 clustersetbinding

GET /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclusterbindings/{clusterbinding_name}

1.3.2.3.1. 描述

查询单个 clusterbinding 获取更多详细信息。

1.3.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default 。	字符串
路径	clusterbinding_name 必需	要查询的 clusterbinding 的名称。	字符串

1.3.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.3.2.3.4. Tags

- cluster.open-cluster-management.io

1.3.2.4. 删除 clusterbinding

DELETE /cluster.open-cluster-management.io/v1beta2/managedclustersetbindings/{clustersetbinding_name}

1.3.2.4.1. 描述

删除单个 clustersetbinding。

1.3.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default 。	字符串
路径	clustersetbinding_name 必需	要删除的 clustersetbinding 的名称。	字符串

1.3.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.3.2.4.4. Tags

- cluster.open-cluster-management.io

1.3.3. 定义

1.3.3.1. Clustersetbinding

Name	描述	模式
apiVersion 必需	ManagedClusterSetBinding 的版本化模式。	字符串
kind 必需	代表 REST 资源的字符串值。	字符串
metadata 必需	ManagedClusterSetBinding 的元数据。	对象
spec 必需	ManagedClusterSetBinding 的规格。	<code>spec</code>

spec

Name	描述	模式
clusterSet 必需	要绑定的 ManagedClusterSet 的名称。它必须与 ManagedClusterSetBinding 的实例名称匹配，并在创建后无法更改。	字符串

1.4. CLUSTERVIEW API (V1ALPHA1)

1.4.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 **clusterview** 资源相关的 API 信息。**clusterview** 资源提供了一个 CLI 命令，可让您查看您可以访问的受管集群和受管集群集的列表。三个可能的请求有：`list`、`get` 和 `watch`。

1.4.1.1. 版本信息

版本：2.10.0

1.4.1.2. URI scheme

BasePath：/kubernetes/apis
Schemes：HTTPS

1.4.1.3. Tags

- `clusterview.open-cluster-management.io`：查看 ID 可访问的受管集群列表。

1.4.2. 路径

1.4.2.1. 获取受管集群

-

GET /managedclusters.clusterview.open-cluster-management.io

1.4.2.1.1. 描述

查看您可以访问的受管集群列表。

1.4.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.4.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.4.2.1.4. 使用

- **managedcluster/yaml**

1.4.2.1.5. Tags

- clusterview.open-cluster-management.io

1.4.2.2. 列出受管集群

LIST /managedclusters.clusterview.open-cluster-management.io

1.4.2.2.1. 描述

查看您可以访问的受管集群列表。

1.4.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 可选	要列出受管集群的用户 ID 的名称。	字符串

1.4.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.4.2.2.4. 使用

- **managedcluster/yaml**

1.4.2.2.5. Tags

- `clusterview.open-cluster-management.io`

1.4.2.2.6. HTTP 请求示例

1.4.2.2.6.1. 请求正文

```
{
  "apiVersion": "clusterview.open-cluster-management.io/v1alpha1",
  "kind": "ClusterView",
  "metadata": {
    "name": "<user_ID>"
  },
  "spec": {},
  "status": {}
}
```

1.4.2.3. 观察受管集群集

```
WATCH /managedclusters.clusterview.open-cluster-management.io
```

1.4.2.3.1. 描述

观察您可以访问的受管集群。

1.4.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	clusterview_name 可选	要监视的用户 ID 的名称。	字符串

1.4.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.4.2.4. 列出受管集群集。

```
GET /managedclustersets.clusterview.open-cluster-management.io
```

1.4.2.4.1. 描述

列出您可以访问的受管集群。

1.4.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	clusterview_name 可选	要监视的用户 ID 的名称。	字符串

1.4.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.4.2.5. 列出受管集群集。

LIST /managedclustersets.clusterview.open-cluster-management.io

1.4.2.5.1. 描述

列出您可以访问的受管集群。

1.4.2.5.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	clusterview_name 可选	要监视的用户 ID 的名称。	字符串

1.4.2.5.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.4.2.6. 观察受管集群集。

WATCH /managedclustersets.clusterview.open-cluster-management.io

1.4.2.6.1. 描述

观察您可以访问的受管集群。

1.4.2.6.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	clusterview_name 可选	要监视的用户 ID 的名称。	字符串

1.4.2.6.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.5. CHANNELS API

1.5.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的频道资源相关的 API 信息。频道资源有 4 个可用的请求：create、query、delete 和 update。

1.5.1.1. 版本信息

版本：2.10.0

1.5.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.5.1.3. Tags

- channels.apps.open-cluster-management.io：创建和管理可部署性资源（deployables）

1.5.2. 路径

1.5.2.1. 创建频道

POST /apps.open-cluster-management.io/v1/namespaces/{namespace}/channels

1.5.2.1.1. 描述

创建频道

1.5.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间，如 default。	字符串
Body	body 必需	描述要创建的 deployable 的参数。	Channel

1.5.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.5.2.1.4. 使用

- **application/yaml**

1.5.2.1.5. Tags

- channels.apps.open-cluster-management.io

1.5.2.1.6. HTTP 请求示例

1.5.2.1.6.1. 请求正文

```
{
  "apiVersion": "apps.open-cluster-management.io/v1",
  "kind": "Channel",
  "metadata": {
    "name": "sample-channel",
    "namespace": "default"
  },
  "spec": {
    "configMapRef": {
      "kind": "configmap",
      "name": "info-resource-filter-configmap"
    },
    "pathname": "https://charts.helm.sh/stable",
    "type": "HelmRepo"
  }
}
```

1.5.2.2. 为目标命名空间查询所有频道

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/channels
```

1.5.2.2.1. 描述

查询您的频道以获得更详细的信息。

1.5.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.5.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.5.2.2.4. 使用

- **application/yaml**

1.5.2.2.5. Tags

- channels.apps.open-cluster-management.io

1.5.2.3. 查询命名空间的单个频道

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/channels/{channel_name}
```

1.5.2.3.1. 描述

查询单个频道以了解更多详情。

1.5.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	channel_name <i>必需</i>	您要查询的 deployable 的名称。	字符串
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.5.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.5.2.3.4. Tags

- channels.apps.open-cluster-management.io

1.5.2.4. 删除频道

```
DELETE /apps.open-cluster-management.io/v1/namespaces/{namespace}/channels/{channel_name}
```

1.5.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	channel_name <i>必需</i>	要删除的频道名称。	字符串
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.5.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.5.2.4.3. Tags

- channels.apps.open-cluster-management.io

1.5.3. 定义

1.5.3.1. Channel

Name	模式
apiVersion 必需	字符串
kind 必需	字符串
metadata 必需	对象
spec 必需	spec

spec

Name	描述	模式
configMapRef 可选	ObjectReference 包含足够信息供您检查或修改引用的对象。	configMapRef
gates 可选	ChannelGate 定义推广到频道的条件	gates
pathname 必需		字符串

Name	描述	模式
secretRef <i>可选</i>	ObjectReference 包含足够信息供您检查或修改引用的对象。	secretRef
sourceNamespaces <i>可选</i>		enum (Namespace, HelmRepo, ObjectBucket, Git, namespace, helmrepo, objectbucket, github) array

configMapRef

Name	描述	模式
apiVersion <i>可选</i>	引用的 API 版本。	字符串
fieldPath <i>可选</i>	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 <code>desiredState.manifest.containers[2]</code> 。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似： <code>"spec.containers{name}"</code> （其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 <code>"spec.containers[2]"</code> （使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。Todo: 这个设计不是最终设计的，且该字段在将来会有所改变。	字符串
kind <i>可选</i>	引用的类型。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/	name <i>可选</i>
引用的名称。更多信息： 名称	字符串	namespace <i>可选</i>
引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/	字符串	resourceVersion <i>可选</i>

Name	描述	模式
指定构成这个引用的 resourceVersion (如果存在)。更多信息： https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency	字符串	uid 可选

gates

Name	描述	模式
annotations 可选	k8s 的典型注解	annotations
labelSelector 可选	标签选择器，即一组资源的标签查询。matchLabels 和 matchExpressions 的逻辑关系是 AND。空标签选择器匹配所有对象。null 标签选择器不匹配任何对象。	labelSelector
name 可选		字符串

annotations

Name	模式
key 可选	字符串
value 可选	字符串

labelSelector

Name	描述	模式
matchExpressions 可选	matchExpressions 是标签选择器要求列表。要求的逻辑关系是 AND。	matchExpressions 数组

Name	描述	模式
matchLabels 可选	matchLabels 是 {key,value} 对的映射。MatchLabels 映射中的单个 {key,value} 等同于 matchExpressions 的一个元素，其 key 字段是 "key"，运算符是 "In"，值数组仅包含 "value"。要求的逻辑关系是 AND。	字符串、字符串映射

matchExpressions

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

secretRef

Name	描述	模式
apiVersion 可选	引用的 API 版本。	字符串
fieldPath 可选	如果引用的是对象的一部分而非整个对象，则该字符串应包含有效的 JSON/Go 字段访问声明，如 desiredState.manifest.containers[2]。例如，如果对象引用是一个 pod 中的容器，它应该还以下类似："spec.containers{name}"（其中 "name" 是指触发事件的容器的名称），如果不使用容器名称，则应类似于 "spec.containers[2]"（使用此 pod 中索引值为 2 的容器）。选择这个语法的原因是，为了有一些指向对象部分的正确定义方法。Todo: 这个设计不是最终设计的，且该字段在将来会有所改变。	字符串
kind 可选	引用的类型。更多信息： https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds	字符串
name 可选	引用的名称。更多信息： 名称	字符串

Name	描述	模式
namespace 可选	引用的命名空间。更多信息： https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/	字符串
resourceVersion 可选	指定构成这个引用的 resourceVersion（如果存在）。更多信息： https://git.k8s.io/community/contributors/devel/api-conventions.md#concurrency-control-and-consistency	字符串
uid 可选	引用的 UID。更多信息： UUID	字符串

1.6. SUBSCRIPTIONS API

1.6.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的订阅资源相关的 API 信息。订阅资源有 4 个可用的请求：`create`、`query`、`delete` 和 `update`。已弃用：`PlacementRule`

1.6.1.1. 版本信息

版本：2.10.0

1.6.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.6.1.3. Tags

- `subscription.apps.open-cluster-management.io`：创建和管理订阅

1.6.2. 路径

1.6.2.1. 创建订阅

POST /apps.open-cluster-management.io/v1/namespaces/{namespace}/subscriptions

1.6.2.1.1. 描述

创建订阅

1.6.2.1.2. 参数

类型	Name	描述	模式
----	------	----	----

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串
Body	body <i>必需</i>	描述要创建的订阅的参数。	Subscription

1.6.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.6.2.1.4. 使用

- **subscription/yaml**

1.6.2.1.5. Tags

- subscriptions.apps.open-cluster-management.io

1.6.2.1.6. HTTP 请求示例

1.6.2.1.6.1. 请求正文

```
{
  "apiVersion": "apps.open-cluster-management.io/v1",
  "kind": "Subscription",
  "metadata": {
    "name": "sample_subscription",
    "namespace": "default",
    "labels": {
      "app": "sample_subscription-app"
    }
  },
  "annotations": {
```

```

    "apps.open-cluster-management.io/git-path" : "apps/sample/",
    "apps.open-cluster-management.io/git-branch" : "sample_branch"
  }
},
"spec" : {
  "channel" : "channel_namespace/sample_channel",
  "packageOverrides" : [ {
    "packageName" : "my-sample-application",
    "packageAlias" : "the-sample-app",
    "packageOverrides" : [ {
      "path" : "spec",
      "value" : {
        "persistence" : {
          "enabled" : false,
          "useDynamicProvisioning" : false
        },
        "license" : "accept",
        "tls" : {
          "hostname" : "my-mcm-cluster.icp"
        },
        "sso" : {
          "registrationImage" : {
            "pullSecret" : "hub-repo-docker-secret"
          }
        }
      }
    }
  ]
},
"placement" : {
  "placementRef" : {
    "kind" : "PlacementRule",
    "name" : "demo-clusters"
  }
}
}
}
}

```

1.6.2.2. 查询所有订阅

GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/subscriptions

1.6.2.2.1. 描述

查询您的订阅以获取更多详情。

1.6.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.6.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.6.2.2.4. 使用

- **subscription/yaml**

1.6.2.2.5. Tags

- subscriptions.apps.open-cluster-management.io

1.6.2.3. 查询单个订阅

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/subscriptions/{subscription_name}
```

1.6.2.3.1. 描述

查询单个订阅以了解更多详情。

1.6.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

类型	Name	描述	模式
路径	subscription_name 必需	要查询的订阅名称。	字符串

1.6.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.6.2.3.4. Tags

- subscriptions.apps.open-cluster-management.io

1.6.2.4. 创建一个订阅：

```
DELETE /apps.open-cluster-management.io/v1/namespaces/{namespace}/subscriptions/{subscription_name}
```

1.6.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间，如 default。	字符串
路径	subscription_name 必需	要删除的订阅的名称。	字符串

1.6.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.6.2.4.3. Tags

- `subscriptions.apps.open-cluster-management.io`

1.6.3. 定义

1.6.3.1. Subscription

Name	模式
<code>apiVersion</code> 必需	字符串
<code>kind</code> 必需	字符串
<code>metadata</code> 必需	metadata
<code>spec</code> 必需	spec
<code>status</code> 可选	status

metadata

Name	模式
<code>annotations</code> 可选	对象
<code>labels</code> 可选	对象

Name	模式
name 可选	字符串
namespace 可选	字符串

spec

Name	模式
channel 必需	字符串
name 可选	字符串
overrides 可选	overrides 数组
packageFilter 可选	packageFilter
packageOverrides 可选	packageOverrides 数组
placement 可选	placement
timewindow 可选	timewindow

overrides

Name	模式
clusterName 必需	字符串
clusterOverrides 必需	对象数组

packageFilter

Name	描述	模式
annotations <i>可选</i>		字符串、字符串映射
filterRef <i>可选</i>		filterRef
labelSelector <i>可选</i>		labelSelector
version <i>可选</i>	Pattern: <code>"((\.[0-9])(\.) (\.[0-9])?(\.[xX]))\$"</code>	字符串

filterRef

Name	模式
name <i>可选</i>	字符串

labelSelector

Name	模式
matchExpressions <i>可选</i>	matchExpressions 数组
matchLabels <i>可选</i>	字符串、字符串映射

matchExpressions

Name	模式
key <i>必需</i>	字符串
operator <i>必需</i>	字符串
values <i>可选</i>	字符串数组

packageOverrides

Name	模式
packageAlias <i>可选</i>	字符串
packageName <i>必需</i>	字符串
packageOverrides <i>可选</i>	对象数组

placement

Name	模式
clusterSelector <i>可选</i>	clusterSelector
clusters <i>可选</i>	clusters 数组
local <i>可选</i>	布尔值
placementRef <i>可选</i>	placementRef

clusterSelector

Name	模式
matchExpressions <i>可选</i>	matchExpressions 数组
matchLabels <i>可选</i>	字符串、字符串映射

matchExpressions

Name	模式
key <i>必需</i>	字符串
operator <i>必需</i>	字符串

Name	模式
values <i>可选</i>	字符串数组

clusters

Name	模式
name <i>必需</i>	字符串

placementRef

Name	模式
apiVersion <i>可选</i>	字符串
fieldPath <i>可选</i>	字符串
kind <i>可选</i>	字符串
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

timewindow

Name	模式
daysofweek <i>可选</i>	字符串数组
hours <i>可选</i>	hours 数组

Name	模式
location <i>可选</i>	字符串
windowType <i>可选</i>	enum (active、blocked、Active、Blocked)

hours

Name	模式
end <i>可选</i>	字符串
start <i>可选</i>	字符串

status

Name	模式
lastUpdateTime <i>可选</i>	字符串 (date-time)
message <i>可选</i>	字符串
phase <i>可选</i>	字符串
reason <i>可选</i>	字符串
statuses <i>可选</i>	object

1.7. PLACEMENTRULES API (已弃用)

1.7.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 PlacementRule 资源相关的 API 信息。PlacementRule 资源有 4 个可用的请求：create、query、delete 和 update。

1.7.1.1. 版本信息

版本：2.10.0

1.7.1.2. URI scheme

BasePath : /kubernetes/apis

Schemes : HTTPS

1.7.1.3. Tags

- `placementrules.apps.open-cluster-management.io` : 创建和管理放置规则

1.7.2. 路径

1.7.2.1. 创建放置规则

POST /apps.open-cluster-management.io/v1/namespaces/{namespace}/placementrules

1.7.2.1.1. 描述

创建放置规则。

1.7.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串
Body	body <i>必需</i>	描述要创建的放置规则的参数。	PlacementRule

1.7.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.7.2.1.4. 使用

- **application/yaml**

1.7.2.1.5. Tags

- placementrules.apps.open-cluster-management.io

1.7.2.1.6. HTTP 请求示例

1.7.2.1.6.1. 请求正文

```
{
  "apiVersion": "apps.open-cluster-management.io/v1",
  "kind": "PlacementRule",
  "metadata": {
    "name": "towhichcluster",
    "namespace": "ns-sub-1"
  },
  "spec": {
    "clusterConditions": [ {
      "type": "ManagedClusterConditionAvailable",
      "status": "True"
    } ],
    "clusterSelector": { }
  }
}
```

1.7.2.2. 查询所有放置规则

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/placementrules
```

1.7.2.2.1. 描述

查询放置规则以获取更多详细信息。

1.7.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串

1.7.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.7.2.2.4. 使用

- **application/yaml**

1.7.2.2.5. Tags

- placementrules.apps.open-cluster-management.io

1.7.2.3. 查询单个放置规则

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/placementrules/{placementrule_name}
```

1.7.2.3.1. 描述

查询单个放置规则以了解更多详细信息。

1.7.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串
路径	placementrule_name <i>必需</i>	要查询的放置规则的名称。	字符串

1.7.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.7.2.3.4. Tags

- placementrules.apps.open-cluster-management.io

1.7.2.4. 删除放置规则

```
DELETE /apps.open-cluster-management.io/v1/namespaces/{namespace}/placementrules/{placementrule_name}
```

1.7.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串
路径	placementrule_name <i>必需</i>	要删除的放置规则的名称。	字符串

1.7.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容

HTTP 代码	描述	模式
500	内部服务错误	无内容
503	服务不可用	无内容

1.7.2.4.3. Tags

- `placementrules.apps.open-cluster-management.io`

1.7.3. 定义

1.7.3.1. PlacementRule

Name	模式
<code>apiVersion</code> 必需	字符串
<code>kind</code> 必需	字符串
<code>metadata</code> 必需	对象
<code>spec</code> 必需	spec

spec

Name	模式
<code>clusterConditions</code> 可选	clusterConditions 数组
<code>clusterReplicas</code> 可选	整数
<code>clusterSelector</code> 可选	clusterSelector
<code>clusters</code> 可选	clusters 数组

Name	模式
policies <i>可选</i>	policies 数组
resourceHint <i>可选</i>	resourceHint
schedulerName <i>可选</i>	字符串

clusterConditions

Name	模式
status <i>可选</i>	字符串
type <i>可选</i>	字符串

clusterSelector

Name	模式
matchExpressions <i>可选</i>	matchExpressions 数组
matchLabels <i>可选</i>	字符串、字符串映射

matchExpressions

Name	模式
key <i>可选</i>	字符串
operator <i>可选</i>	字符串
values <i>可选</i>	字符串数组

clusters

Name	模式
name <i>可选</i>	字符串

policies

Name	模式
apiVersion <i>可选</i>	字符串
fieldPath <i>可选</i>	字符串
kind <i>可选</i>	字符串
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

resourceHint

Name	模式
order <i>可选</i>	字符串
type <i>可选</i>	字符串

1.8. APPLICATIONS API

1.8.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的应用程序资源相关的 API 信息。应用程序资源有 4 个可用的请求：create、query、delete 和 update。

1.8.1.1. 版本信息

版本: 2.10.0

1.8.1.2. URI scheme

BasePath: /kubernetes/apis

Schemes: HTTPS

1.8.1.3. Tags

- Applications.app.k8s.io : 创建和管理应用程序

1.8.2. 路径

1.8.2.1. 创建应用程序

POST /app.k8s.io/v1beta1/namespaces/{namespace}/applications

1.8.2.1.1. 描述

创建应用程序。

1.8.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证: Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串
Body	body 必需	描述要创建的应用程序的参数。	Application

1.8.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.8.2.1.4. 使用

- **application/yaml**

1.8.2.1.5. Tags

- applications.app.k8s.io

1.8.2.1.6. HTTP 请求示例

1.8.2.1.6.1. 请求正文

```
{
  "apiVersion": "app.k8s.io/v1beta1",
  "kind": "Application",
  "metadata": {
    "labels": {
      "app": "nginx-app-details"
    },
    "name": "nginx-app-3",
    "namespace": "ns-sub-1"
  },
  "spec": {
    "componentKinds": [ {
      "group": "apps.open-cluster-management.io",
      "kind": "Subscription"
    } ]
  },
  "selector": {
    "matchLabels": {
      "app": "nginx-app-details"
    }
  },
  "status": { }
}
```

1.8.2.2. 查询所有应用程序

```
GET /app.k8s.io/v1beta1/namespaces/{namespace}/applications
```

1.8.2.2.1. 描述

查询您的应用程序以获取更多详情。

1.8.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.8.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.8.2.2.4. 使用

- **application/yaml**

1.8.2.2.5. Tags

- applications.app.k8s.io

1.8.2.3. 查询单个应用程序

```
GET /app.k8s.io/v1beta1/namespaces/{namespace}/applications/{application_name}
```

1.8.2.3.1. 描述

查询单个应用程序以获取更多详细信息。

1.8.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	<code>application_name</code> <i>必需</i>	您要查询的应用程序的名称。	字符串
路径	<code>namespace</code> <i>必需</i>	要使用的命名空间，如 default。	字符串

1.8.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.8.2.3.4. Tags

- `applications.app.k8s.io`

1.8.2.4. 删除应用程序

```
DELETE /app.k8s.io/v1beta1/namespaces/{namespace}/applications/{application_name}
```

1.8.2.4.1. 参数

类型	Name	描述	模式
标头	<code>COOKIE</code> <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	<code>application_name</code> <i>必需</i>	要删除的应用程序的名称。	字符串

类型	Name	描述	模式
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.8.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.8.2.4.3. Tags

- applications.app.k8s.io

1.8.3. 定义

1.8.3.1. Application

Name	模式
apiVersion <i>必需</i>	字符串
kind <i>必需</i>	字符串
metadata <i>必需</i>	对象
spec <i>必需</i>	spec

spec

Name	模式
assemblyPhase <i>可选</i>	字符串
componentKinds <i>可选</i>	对象数组
descriptor <i>可选</i>	descriptor
info <i>可选</i>	info 数组
selector <i>可选</i>	对象

descriptor

Name	模式
description <i>可选</i>	字符串
icons <i>可选</i>	icons 数组
keywords <i>可选</i>	字符串数组
links <i>可选</i>	links 数组
maintainers <i>可选</i>	maintainers 数组
notes <i>可选</i>	字符串
owners <i>可选</i>	owners 数组
type <i>可选</i>	字符串
version <i>可选</i>	字符串

icons

Name	模式
size <i>可选</i>	字符串
src <i>必需</i>	字符串
type <i>可选</i>	字符串

links

Name	模式
description <i>可选</i>	字符串
url <i>可选</i>	字符串

maintainers

Name	模式
email <i>可选</i>	字符串
name <i>可选</i>	字符串
url <i>可选</i>	字符串

owners

Name	模式
email <i>可选</i>	字符串
name <i>可选</i>	字符串

Name	模式
url <i>可选</i>	字符串

info

Name	模式
name <i>可选</i>	字符串
type <i>可选</i>	字符串
value <i>可选</i>	字符串
valueFrom <i>可选</i>	valueFrom

valueFrom

Name	模式
configMapKeyRef <i>optional</i>	configMapKeyRef
ingressRef <i>可选</i>	ingressRef
secretKeyRef <i>可选</i>	secretKeyRef
serviceRef <i>可选</i>	serviceRef
type <i>可选</i>	字符串

configMapKeyRef

Name	模式
apiVersion <i>可选</i>	字符串
fieldPath <i>可选</i>	字符串
key <i>可选</i>	字符串
kind <i>可选</i>	字符串
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

ingressRef

Name	模式
apiVersion <i>可选</i>	字符串
fieldPath <i>可选</i>	字符串
host <i>可选</i>	字符串
kind <i>可选</i>	字符串
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串

Name	模式
path <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

secretKeyRef

Name	模式
apiVersion <i>可选</i>	字符串
fieldPath <i>可选</i>	字符串
key <i>可选</i>	字符串
kind <i>可选</i>	字符串
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

serviceRef

Name	模式
apiVersion <i>可选</i>	字符串

Name	模式
fieldPath 可选	字符串
kind 可选	字符串
name 可选	字符串
namespace 可选	字符串
path 可选	字符串
port 可选	integer (int32)
resourceVersion 可选	字符串
uid 可选	字符串

1.9. HELM API

1.9.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 HelmRelease 资源相关的 API 信息。HelmRelease 有 4 个可用的请求：create、query、delete 和 update。

1.9.1.1. 版本信息

版本：2.10.0

1.9.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.9.1.3. Tags

- helmreleases.apps.open-cluster-management.io：创建和管理 helmreleases

1.9.2. 路径

1.9.2.1. 创建 helmrelease

POST /apps.open-cluster-management.io/v1/namespaces/{namespace}/helmreleases

1.9.2.1.1. 描述

创建 helmrelease。

1.9.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串
Body	body 必需	描述要创建的 helmrelease 的参数。	HelmRelease

1.9.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.9.2.1.4. 使用

- **application/yaml**

1.9.2.1.5. Tags

- helmreleases.apps.open-cluster-management.io

1.9.2.1.6. HTTP 请求示例

1.9.2.1.6.1. 请求正文

```

{
  "apiVersion": "apps.open-cluster-management.io/v1",
  "kind": "HelmRelease",
  "metadata": {
    "name": "nginx-ingress",
    "namespace": "default"
  },
  "repo": {
    "chartName": "nginx-ingress",
    "source": {
      "helmRepo": {
        "urls": [ "https://kubernetes-charts.storage.googleapis.com/nginx-ingress-1.26.0.tgz" ]
      },
      "type": "helmrepo"
    },
    "version": "1.26.0"
  },
  "spec": {
    "defaultBackend": {
      "replicaCount": 3
    }
  }
}

```

1.9.2.2. 查询所有 helmreleases

GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/helmreleases

1.9.2.2.1. 描述

查询您的 helmreleases 获取更多详细信息。

1.9.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间，如 default。	字符串

1.9.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容

HTTP 代码	描述	模式
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.9.2.2.4. 使用

- **application/yaml**

1.9.2.2.5. Tags

- helmreleases.apps.open-cluster-management.io

1.9.2.3. 查询单个 helmrelease

```
GET /apps.open-cluster-management.io/v1/namespaces/{namespace}/helmreleases/{helmrelease_name}
```

1.9.2.3.1. 描述

查询单个 helmrelease 获取更多详细信息。

1.9.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	helmrelease_name 必需	要查询的 helmrelease 的名称。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.9.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容

HTTP 代码	描述	模式
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.9.2.3.4. Tags

- helmreleases.apps.open-cluster-management.io

1.9.2.4. 删除 helmrelease

DELETE /apps.open-cluster-management.io/v1/namespaces/{namespace}/helmreleases/{helmrelease_name}

1.9.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	helmrelease_name 必需	要删除的 helmrelease 的名称。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.9.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.9.2.4.3. Tags

- helmreleases.apps.open-cluster-management.io

1.9.3. 定义

1.9.3.1. HelmRelease

Name	模式
apiVersion <i>必需</i>	字符串
kind <i>必需</i>	字符串
metadata <i>必需</i>	对象
repo <i>必需</i>	repo
spec <i>必需</i>	对象
status <i>必需</i>	status

repo

Name	模式
chartName <i>可选</i>	字符串
configMapRef <i>可选</i>	configMapRef
secretRef <i>可选</i>	secretRef

Name	模式
source 可选	source
version 可选	字符串

configMapRef

Name	模式
apiVersion 可选	字符串
fieldPath 可选	字符串
kind 可选	字符串
name 可选	字符串
namespace 可选	字符串
resourceVersion 可选	字符串
uid 可选	字符串

secretRef

Name	模式
apiVersion 可选	字符串
fieldPath 可选	字符串
kind 可选	字符串

Name	模式
name <i>可选</i>	字符串
namespace <i>可选</i>	字符串
resourceVersion <i>可选</i>	字符串
uid <i>可选</i>	字符串

source

Name	模式
github <i>可选</i>	github
helmRepo <i>可选</i>	helmRepo
type <i>可选</i>	字符串

github

Name	模式
branch <i>可选</i>	字符串
chartPath <i>可选</i>	字符串
urls <i>可选</i>	字符串数组

helmRepo

Name	模式
urls <i>可选</i>	字符串数组

status

Name	模式
conditions <i>必需</i>	<code>conditions</code> 数组
deployedRelease <i>可选</i>	<code>deployedRelease</code>

conditions

Name	模式
lastTransitionTime <i>可选</i>	字符串 (date-time)
message <i>可选</i>	字符串
reason <i>可选</i>	字符串
status <i>必需</i>	字符串
type <i>必需</i>	字符串

deployedRelease

Name	模式
manifest <i>可选</i>	字符串
name <i>可选</i>	字符串

1.10. POLICY API**1.10.1. 概述**

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的策略 (Policy) 资源相关的 API 信息。Policy 资源有 4 个可用的请求：create、query、delete 和 update。

1.10.1.1. 版本信息

版本: 2.10.0

1.10.1.2. URI scheme

BasePath: /kubernetes/apis

Schemes: HTTPS

1.10.1.3. Tags

- policy.open-cluster-management.io/v1: 创建和管理策略

1.10.2. 路径

1.10.2.1. 创建策略

POST /policy.open-cluster-management.io/v1/v1alpha1/namespaces/{namespace}/policies/{policy_name}

1.10.2.1.1. 描述

创建策略

1.10.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证: Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串
Body	body 必需	描述要创建策略的参数。	policy

1.10.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.10.2.1.4. 使用

- **application/json**

1.10.2.1.5. Tags

- `policy.open-cluster-management.io`

1.10.2.1.6. HTTP 请求示例

1.10.2.1.6.1. 请求正文

```
{
  "apiVersion": "policy.open-cluster-management.io/v1",
  "kind": "Policy",
  "metadata": {
    "name": "test-policy-swagger",
    "description": "Example body for Policy API Swagger docs"
  },
  "spec": {
    "remediationAction": "enforce",
    "namespaces": {
      "include": [
        "default"
      ],
      "exclude": [
        "kube*"
      ]
    },
    "policy-templates": {
      "kind": "ConfigurationPolicy",
      "apiVersion": "policy.open-cluster-management.io/v1",
      "complianceType": "musthave",
      "metadataComplianceType": "musthave",
      "metadata": {
        "namespace": null,
        "name": "test-role"
      },
      "selector": {
        "matchLabels": {
          "cloud": "IBM"
        }
      },
      "spec": {
        "object-templates": {
          "complianceType": "musthave",
          "metadataComplianceType": "musthave",
          "objectDefinition": {
```

```
"apiVersion": "rbac.authorization.k8s.io/v1",
"kind": "Role",
"metadata": {
  "name": "role-policy",
},
"rules": [
  {
    "apiGroups": [
      "extensions",
      "apps"
    ],
    "resources": [
      "deployments"
    ],
    "verbs": [
      "get",
      "list",
      "watch",
      "delete"
    ]
  },
  {
    "apiGroups": [
      "core"
    ],
    "resources": [
      "pods"
    ],
    "verbs": [
      "create",
      "update",
      "patch"
    ]
  },
  {
    "apiGroups": [
      "core"
    ],
    "resources": [
      "secrets"
    ],
    "verbs": [
      "get",
      "watch",
      "list",
      "create",
      "delete",
      "update",
      "patch"
    ]
  },
],
},
},
},
},
},
```

1.10.2.2. 查询所有策略

```
GET /policy.open-cluster-management.io/v1/namespaces/{namespace}/policies/{policy_name}
```

1.10.2.2.1. 描述

查询您的策略以获得更详细的信息。

1.10.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要应用策略的命名空间, 如 default。	字符串

1.10.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.10.2.2.4. 使用

- **application/json**

1.10.2.2.5. Tags

- policy.open-cluster-management.io

1.10.2.3. 查询单个策略

```
GET /policy.open-cluster-management.io/v1/namespaces/{namespace}/policies/{policy_name}
```

1.10.2.3.1. 描述

查询单个策略以获取更多详细信息。

1.10.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串
路径	policy_name <i>必需</i>	要查询的策略的名称。	字符串
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.10.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.10.2.3.4. Tags

- policy.open-cluster-management.io

1.10.2.4. 删除策略

```
DELETE /policy.open-cluster-management.io/v1/namespaces/{namespace}/policies/{policy_name}
```

1.10.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串
路径	policy_name <i>必需</i>	要删除的策略名称。	字符串

类型	Name	描述	模式
路径	namespace <i>必需</i>	要使用的命名空间，如 default。	字符串

1.10.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.10.2.4.3. Tags

- `policy.open-cluster-management.io`

1.10.3. 定义

1.10.3.1. policy

Name	描述	模式
apiVersion <i>必需</i>	策略的版本化模式（schema）。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	描述定义策略的规则。	对象

spec

Name	描述	模式
remediationAction <i>可选</i>	代表资源中定义的处理违反情况的值。	字符串

Name	描述	模式
namespaceSelector <i>必需</i>	代表策略要应用到的命名空间的值。	字符串

policy-templates

Name	描述	模式
apiVersion <i>必需</i>	策略的版本化模式（schema）。	字符串
kind <i>可选</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	描述定义策略的规则。	对象
complianceType	用于列出必须被评估或应用到受管集群的角色的预期行为和任何 Kubernetes 对象。	字符串
metadataComplianceType <i>optional</i>	为用户提供处理对象标签和注解的方式与其他字段不同。参数值默认为 ComplianceType 参数的值相同。	字符串
clusterConditions <i>可选</i>	定义标签的部分。	字符串
rules <i>可选</i>		字符串

clusterConditions

Name	描述	模式
matchLabels <i>可选</i>	策略应用到一个命名空间所需的标签。	对象
cloud <i>可选</i>	策略应用到一个云供应商所需的标签。	字符串

rules

Name	描述	模式
apiGroups 必需	适用于该规则的 API 列表。	字符串
resources 必需	资源类型列表。	对象
verbs 必需	动词列表。	字符串

1.11. OBSERVABILITY API

1.11.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 MultiClusterObservability 资源相关的 API 信息。MultiClusterObservability 资源有 4 个可用的请求：create、query、delete 和 update。

1.11.1.1. 版本信息

版本：2.10.0

1.11.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.11.1.3. Tags

- Observability.open-cluster-management.io：创建和管理多集群的观察性

1.11.2. 路径

1.11.2.1. 创建 multiclusterobservability 资源

POST /apis/observability.open-cluster-management.io/v1beta2/multiclusterobservabilities

1.11.2.1.1. 描述

创建 MultiClusterObservability 资源。

1.11.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
Body	body 必需	描述要创建的 MultiClusterObservability 资源的参数。	MultiClusterObservability

1.11.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.11.2.1.4. 使用

- **application/yaml**

1.11.2.1.5. Tags

- observability.apps.open-cluster-management.io

1.11.2.1.6. HTTP 请求示例

1.11.2.1.6.1. 请求正文

```
{
  "apiVersion": "observability.open-cluster-management.io/v1beta2",
  "kind": "MultiClusterObservability",
  "metadata": {
    "name": "example"
  },
  "spec": {
    "observabilityAddonSpec": {}
    "storageConfig": {
      "metricObjectStorage": {
        "name": "thanos-object-storage",
        "key": "thanos.yaml"
      }
      "writeStorage": {
        - "key": "",
          "name": ""
        - "key": "",
```

```

    "name": " "
  }
}
}
}

```

1.11.2.2. 查询所有 multiclusterobservabilities

GET /apis/observability.open-cluster-management.io/v1beta2/multiclusterobservabilities

1.11.2.2.1. 描述

查询 MultiClusterObservability 资源以获取更多详细信息。

1.11.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.11.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.11.2.2.4. 使用

- **application/yaml**

1.11.2.2.5. Tags

- observability.apps.open-cluster-management.io

1.11.2.3. 查询单个 multiclusterobservability

GET /apis/observability.open-cluster-management.io/v1beta2/multiclusterobservabilities/{multiclusterobservability_name}

1.11.2.3.1. 描述

查询单个 MultiClusterObservability 资源以获取更多详细信息。

1.11.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	multiclusterobservability_name 必需	要查询的 multiclusterobservability 的名称。	字符串

1.11.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.11.2.3.4. Tags

- observability.apps.open-cluster-management.io

1.11.2.4. 删除一个 multiclusterobservability 资源

```
DELETE /apis/observability.open-cluster-management.io/v1beta2/multiclusterobservabilities/{multiclusterobservability_name}
```

1.11.2.4.1. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	multiclusterobservability_name 必需	要删除的 multiclusterobservability 的名称。	字符串

1.11.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.11.2.4.3. Tags

- observability.apps.open-cluster-management.io

1.11.3. 定义

1.11.3.1. MultiClusterObservability

Name	描述	模式
apiVersion 必需	版本化的 MultiClusterObservability schema。	字符串
kind 必需	代表 REST 资源、MultiClusterObservability 的字符串值。	字符串
metadata 必需	描述定义策略的规则。	对象

spec

Name	描述	模式
<code>enableDownsampling</code> <i>可选</i>	启用或禁用 <code>downsample</code> 。默认值为 true 。如果没有 <code>downsample</code> 数据，则查询不可用。	布尔值
<code>imagePullPolicy</code> <i>可选</i>	MultiClusterObservability 镜像的 pull 策略。默认值为 Always 。	<code>corev1.PullPolicy</code>
<code>imagePullSecret</code> <i>optional</i>	MultiClusterObservability 镜像的 pull secret。默认值为 multiclusterhub-operator-pull-secret	字符串
<code>nodeSelector</code> <i>可选</i>	节点选择器规格。	<code>map[string]string</code>
<code>observabilityAddonSpec</code> <i>required</i>	所有安装了可观察附加组件的受管集群的全局设置。	observabilityAddonSpec
<code>storageConfig</code> <i>必需</i>	指定可观察性要使用的存储配置。	<code>StorageConfig</code>
<code>tolerations</code> <i>可选</i>	提供了所有组件可以容忍任何污点的功能。	<code>[]corev1.Toleration</code>
<code>advanced</code> <i>可选</i>	可观察性的高级配置设置。	advanced
<code>resources</code> <i>可选</i>	MultiClusterObservability 所需的计算资源。	<code>corev1.ResourceRequirements</code>
<code>replicas</code> <i>可选</i>	MultiClusterObservability 副本。	整数

storageConfig

Name	描述	模式
<code>alertmanagerStorageSize</code> <i>可选</i>	应用到 <code>alertmanager</code> 有状态集的存储量。默认值为 1Gi 。	字符串
<code>compactStorageSize</code> <i>可选</i>	应用于 <code>thanos</code> 紧凑有状态集合的存储量。默认值为 100Gi 。	字符串

Name	描述	模式
metricObjectStorage 必需	用于为指标配置 secret 的对象存储。	metricObjectStorage
receiveStorageSize 可选	应用到 thanos 接收有状态集的存储量。默认值为 100Gi 。	字符串
ruleStorageSize 可选	应用到 thanos 规则有状态集的存储量。默认值为 1Gi 。	字符串
storageClass 可选	指定 storageClass 有状态的集合。如果您的操作系统配置了 metricObjectStorage 来创建存储，则此存储用于对象存储。默认值为 gp2 。	字符串
storeStorageSize 可选	应用到 thanos 存储有状态集的存储量。默认值为 10Gi 。	字符串
writeStorage 可选	端点访问信息列表。	[] WriteStorage

writeStorage

Name	描述	模式
name 必需	带有端点访问信息的 secret 名称。	字符串
key 必需	要从中选择的 secret 的密钥。	字符串

metricObjectStorage

Name	描述	模式
key 必需	要从中选择的 secret 的密钥。必须是一个有效的 secret 密钥。请参阅 Thanos 文档 。	字符串
name 必需	metricObjectStorage 的名称。如需更多信息，请参阅 Kubernetes 名称 。	字符串

observabilityAddonSpec

Name	描述	模式
enableMetrics <i>可选</i>	指示 Observability 附加组件是否将指标发送到 hub 集群。默认值为 true 。	布尔值
interval <i>可选</i>	Observability 附加组件将指标发送到 hub 集群的间隔。默认值为 300 秒 (300s)。	整数
resources <i>可选</i>	指标收集器资源要求的资源。默认 CPU 请求为 100m ，内存请求为 100Mi 。	corev1.ResourceRequirements

advanced

Name	描述	模式
retentionConfig <i>可选</i>	指定供可观察性使用的数据保留配置。	RetentionConfig
rbacQueryProxy <i>可选</i>	指定 rbac-query-proxy 部署的副本和资源。	CommonSpec
grafana <i>可选</i>	指定 grafana 部署的副本和资源	CommonSpec
alertmanager <i>可选</i>	指定 alertmanager statefulset 的副本和资源。	CommonSpec
observatoriumAPI <i>可选</i>	指定 observatorium-api 部署的副本和资源。	CommonSpec
queryFrontend <i>可选</i>	为 query-frontend 部署指定副本和资源。	CommonSpec
query <i>可选</i>	指定查询部署的副本和资源。	CommonSpec
receive <i>可选</i>	指定接收 statefulset 的副本和资源。	CommonSpec
rule <i>可选</i>	指定规则 statefulset 的副本和资源。	CommonSpec
store <i>可选</i>	指定存储 statefulset 的副本和资源。	CommonSpec

Name	描述	模式
CompactSpec <i>可选</i>	指定紧凑 statefulset 的资源。	<code>compact</code>
storeMemcached <i>可选</i>	指定 store-memcached 的副本、资源等。	<code>storeMemcached</code>
queryFrontendMemcached <i>可选</i>	指定 query-frontend-memcached 的副本、资源等。	<code>CacheConfig</code>

retentionConfig

Name	描述	模式
blockDuration <i>可选</i>	阻塞时间序列数据库 (TSDB) 块持续时间的长度。默认值为 2h 。	字符串
cleanupInterval <i>可选</i>	清理部分上传块的频率，以及清理启用了 <code>--wait</code> 的删除标记的块的频率。默认值为 5m 。	字符串
deleteDelay <i>可选</i>	从存储桶中删除标记要删除的块前的时间。默认值为 48h 。	字符串
retentionInLocal <i>可选</i>	从本地存储保留原始样本的时间长度。默认值为 24h 。	字符串
retentionResolutionRaw <i>可选</i>	在存储桶中保留原始分辨率样本的时间。默认值为 30 天 (30d)	字符串
retentionResolution5m <i>可选</i>	在存储桶中保留分辨率 1 样本 (5 分钟) 的时间长度。默认值为 180 天 (180d)。	字符串
retentionResolution1h <i>可选</i>	在存储桶中保留分辨率 2 样本 (1 小时) 的时间长度。默认值为 0 天 (0d)。	字符串

CompactSpec

Name	描述	模式
resources <i>可选</i>	thanos 紧凑所需的计算资源。	<code>corev1.ResourceRequirements</code>

Name	描述	模式
serviceAccountAnnotations <i>可选</i>	annotations 是一个无结构的键值映射，它存储有紧凑的服务帐户。	map[string]string

storeMemcached

Name	描述	模式
resources <i>可选</i>	MultiClusterObservability 所需的计算资源。	corev1.ResourceRequirements
replicas <i>可选</i>	MultiClusterObservability 副本。	整数
memoryLimitMb <i>optional</i>	以 MB 为单位的 Memcached 内存限值。	整数
maxItemSize <i>可选</i>	Memcached 的最大项目大小。默认值为 1m, min:1k, max:1024m 。	字符串
connectionLimit <i>optional</i>	Memcached 同时连接数上限。默认值为	整数

status

Name	描述	模式
status <i>可选</i>	status 包含 MultiClusterObservability 的不同条件状态。	metav1.Condition

CommonSpec

Name	描述	模式
resources <i>可选</i>	组件所需的计算资源。	corev1.ResourceRequirements
replicas <i>可选</i>	组件的副本。	整数

QuerySpec

Name	描述	模式
CommonSpec <i>可选</i>	指定查询部署的副本和资源。	CommonSpec
serviceAccountAnnotations <i>可选</i>	annotations 是一个无结构的键值映射，它会与查询服务帐户存储。	map[string]string

ReceiveSpec

Name	描述	模式
CommonSpec <i>可选</i>	指定查询部署的副本和资源。	CommonSpec
serviceAccountAnnotations <i>可选</i>	annotations 是一个无结构的键值映射，它会与查询服务帐户存储。	map[string]string

StoreSpec

Name	描述	模式
CommonSpec <i>可选</i>	指定查询部署的副本和资源。	CommonSpec
serviceAccountAnnotations <i>可选</i>	annotations 是一个无结构的键值映射，它会与查询服务帐户存储。	map[string]string

RuleSpec

Name	描述	模式
CommonSpec <i>可选</i>	指定查询部署的副本和资源。	CommonSpec
evalInterval <i>可选</i>	指定规则的评估间隔。	字符串
serviceAccountAnnotations <i>可选</i>	annotations 是一个无结构的键值映射，它会与查询服务帐户存储。	map[string]string

1.12. 搜索查询 API

搜索查询 API 不是 Kubernetes API，因此无法通过 Red Hat OpenShift Container Platform API Explorer 显示。继续阅读以了解搜索查询 API 功能。

1.12.1. 概述

您可以使用路由公开搜索查询 API，并使用 API 解析搜索查询。API 是一个 GraphQL 端点。您可以使用任何客户端，如 curl 或 Postman。

1.12.1.1. 版本信息

版本: 2.10.0

1.12.1.2. URI scheme

basePath : /searchapi/graphql

Schemes : HTTPS

1.12.1.3. 配置 API 访问

使用以下命令，创建路由以访问集群外部的搜索 API：

```
oc create route passthrough search-api --service=search-search-api -n open-cluster-management
```

重要： 您必须配置路由来保护您的环境。如需了解更多详细信息，请参阅 OpenShift Container Platform 文档中的 [Route 配置](#)。

1.12.2. 模式设计

```
input SearchFilter {
  property: String!
  values: [String!]
}
input SearchInput {
  keywords: [String]
  filters: [SearchFilter]
  limit: Int
  relatedKinds: [String]
}
type SearchResult {
  count: Int
  items: [Map]
  related: [SearchRelatedResult]
}
type SearchRelatedResult {
  kind: String!
  count: Int
  items: [Map]
}
```

带有 ! 的参数表示需要字段。

1.12.2.1. 查询输入的描述表

类型	描述	属性
SearchFilter	定义用于过滤结果的键和值。当您为属性提供多个值时，API 会将值解释为"OR"操作。当您提供多个过滤器时，结果会匹配所有过滤器，API 会解释为"AND"操作。	string
SearchInput	输入关键字以接收资源列表。当您提供许多关键字时，API 会将它解析为"AND"操作。	字符串
limit	确定输入查询后返回的最大结果数。默认值为 10,000 。值 -1 表示删除了限制。	整数

1.12.2.2. 模式示例

```
{
  "query": "type SearchResult {count: Int!items: [Map]related: [SearchRelatedResult]} type SearchRelatedResult {kind: String!count: Int!items: [Map]}",
  "variables": {
    "input": [
      {
        "keywords": [],
        "filters": [
          {
            "property": "kind",
            "values": [
              "Deployment"
            ]
          }
        ],
        "limit": 10
      }
    ]
  }
}
```

1.12.3. 通用模式

```
type Query {
  search(input: [SearchInput]): [SearchResult]
  searchComplete(property: String!, query: SearchInput, limit: Int): [String]
  searchSchema: Map
  messages: [Message]
}
```

1.12.4. 支持的查询

继续读取以查看 JSON 格式支持的查询类型。

1.12.4.1. 搜索部署

query :

```
query mySearch($input: [SearchInput]) {
  search(input: $input) {
    items
  }
}
```

变量 :

```
{"input":[
  {
    "keywords":[],
    "filters":[
      {"property":"kind","values":["Deployment"]}],
    "limit":10
  }
]}
```

1.12.4.2. 搜索 pod

query :

```
query mySearch($input: [SearchInput]) {
  search(input: $input) {
    items
  }
}
```

变量 :

```
{"input":[
  {
    "keywords":[],
    "filters":[
      {"property":"kind","values":["Pod"]}],
    "limit":10
  }
]}
```

1.13. MULTICLUSTERHUB API

1.13.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 MultiClusterHub 资源相关的 API 信息。MultiClusterHub 资源有 4 个可用的请求：create、query、delete 和 update。

1.13.1.1. 版本信息

版本: 2.10.0

1.13.1.2. URI scheme

BasePath: /kubernetes/apis

Schemes: HTTPS

1.13.1.3. Tags

- multiclusterhubs.operator.open-cluster-management.io: 创建和管理多集群 hub operator

1.13.2. 路径

1.13.2.1. 创建 MultiClusterHub 资源

POST /operator.open-cluster-management.io/v1beta1/namespaces/{namespace}/mch

1.13.2.1.1. 描述

创建一个 MultiClusterHub 资源来定义 multicluster hub 实例的配置。

1.13.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证: Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串
Body	body 必需	描述要创建的 multicluster hub 的参数。	定义

1.13.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.13.2.1.4. 使用

- **multiclusterhubs/yaml**

1.13.2.1.5. Tags

- multiclusterhubs.operator.open-cluster-management.io

1.13.2.1.6. HTTP 请求示例

1.13.2.1.6.1. 请求正文

```

{
  "apiVersion": "apiextensions.k8s.io/v1",
  "kind": "CustomResourceDefinition",
  "metadata": {
    "name": "multiclusterhubs.operator.open-cluster-management.io"
  },
  "spec": {
    "group": "operator.open-cluster-management.io",
    "names": {
      "kind": "MultiClusterHub",
      "listKind": "MultiClusterHubList",
      "plural": "multiclusterhubs",
      "shortNames": [
        "mch"
      ],
      "singular": "multiclusterhub"
    },
    "scope": "Namespaced",
    "versions": [
      {
        "additionalPrinterColumns": [
          {
            "description": "The overall status of the multicluster hub.",
            "jsonPath": ".status.phase",
            "name": "Status",
            "type": "string"
          }
        ],
        {
          "jsonPath": ".metadata.creationTimestamp",
          "name": "Age",
          "type": "date"
        }
      ],
      "name": "v1",
      "schema": {
        "openAPIV3Schema": {
          "description": "MultiClusterHub defines the configuration for an instance of the multiCluster hub, a central point for managing multiple Kubernetes-based clusters. The deployment of multicluster hub components is determined based on the configuration that is defined in this resource.",
          "properties": {
            "apiVersion": {
              "description": "APIVersion defines the versioned schema of this representation

```


of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources>,

```

    "type": "string"
  },
  "kind": {
    "description": "Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. The value is in CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
    "type": "string"
  },
  "metadata": {
    "type": "object"
  },
  "spec": {
    "description": "MultiClusterHubSpec defines the desired state of MultiClusterHub.",
    "properties": {
      "availabilityConfig": {
        "description": "Specifies deployment replication for improved availability. Options are: Basic and High (default).",
        "type": "string"
      },
      "customCAConfigmap": {
        "description": "Provide the customized OpenShift default ingress CA certificate to {product-title-short}.",
        "type": "string"
      },
      "disableHubSelfManagement": {
        "description": "Disable automatic import of the hub cluster as a managed cluster.",
        "type": "boolean"
      },
      "disableUpdateClusterImageSets": {
        "description": "Disable automatic update of ClusterImageSets.",
        "type": "boolean"
      },
      "hive": {
        "description": "(Deprecated) Overrides for the default HiveConfig specification.",
        "properties": {
          "additionalCertificateAuthorities": {
            "description": "(Deprecated) AdditionalCertificateAuthorities is a list of references to secrets in the 'hive' namespace that contain an additional Certificate Authority to use when communicating with target clusters. These certificate authorities are used in addition to any self-signed CA generated by each cluster on installation.",
            "items": {
              "description": "LocalObjectReference contains the information to let you locate the referenced object inside the same namespace.",
              "properties": {
                "name": {
                  "description": "Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
                  "type": "string"
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "type": "object"
},
  "type": "array"
},
  "backup": {
    "description": "(Deprecated) Backup specifies configuration for backup
      integration. If absent, backup integration is disabled.",
    "properties": {
      "minBackupPeriodSeconds": {
        "description": "(Deprecated) MinBackupPeriodSeconds specifies
          that a minimum of MinBackupPeriodSeconds occurs in between
          each backup. This is used to rate limit backups. This potentially
          batches together multiple changes into one backup. No backups
          are lost for changes that happen during the interval
          that is queued up, and results in a backup once
          the interval has been completed.",
        "type": "integer"
      },
      "velero": {
        "description": "(Deprecated) Velero specifies configuration for the Velero backup
          integration.",
        "properties": {
          "enabled": {
            "description": "(Deprecated) Enabled dictates if the Velero backup integration is enabled. If not
              specified, the default is disabled.",
            "type": "boolean"
          }
        },
        "type": "object"
      },
      "type": "object"
    },
    "externalDNS": {
      "description": "(Deprecated) ExternalDNS specifies configuration for external-dns if it is to be
        deployed by Hive. If absent, external-dns is not deployed.",
      "properties": {
        "aws": {
          "description": "(Deprecated) AWS contains AWS-specific settings for external DNS.",
          "properties": {
            "credentials": {
              "description": "(Deprecated) Credentials reference a secret that is used to authenticate with AWS
                Route53. It needs permission to manage entries in each of the managed domains for this cluster.
                Secret should have AWS keys named 'aws_access_key_id' and 'aws_secret_access_key'.",
              "properties": {
                "name": {
                  "description": "Name of the referent. More info:
                    https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
                  "type": "string"
                }
              },
              "type": "object"
            }
          },
          "type": "object"
        }
      },
      "type": "object"
    }
  },
}

```

```

"type": "object"
},
"gcp": {
"description": "(Deprecated) GCP contains Google Cloud Platform specific settings for external
DNS.",
"properties": {
"credentials": {
"description": "(Deprecated) Credentials reference a secret that is used to authenticate with GCP
DNS. It needs permission to manage entries in each of the managed domains for this cluster. Secret
should have a key names 'osServiceAccount.json'. The credentials must specify the project to use.",
"properties": {
"name": {
"description": "Name of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
"type": "string"
}
},
"type": "object"
}
},
"type": "object"
},
"failedProvisionConfig": {
"description": "(Deprecated) FailedProvisionConfig is used to configure settings related to handling
provision failures.",
"properties": {
"skipGatherLogs": {
"description": "(Deprecated) SkipGatherLogs disables functionality that attempts to gather full logs
from the cluster if an installation fails for any reason. The logs are stored in a persistent volume for up
to seven days.",
"type": "boolean"
}
},
"type": "object"
},
"globalPullSecret": {
"description": "(Deprecated) GlobalPullSecret is used to specify a pull secret that is used globally by
all of the cluster deployments. For each cluster deployment, the contents of GlobalPullSecret are
merged with the specific pull secret for a cluster deployment(if specified), with precedence given to
the contents of the pull secret for the cluster deployment.",
"properties": {
"name": {
"description": "Name of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
"type": "string"
}
},
"type": "object"
},
"maintenanceMode": {
"description": "(Deprecated) MaintenanceMode can be set to true to disable the Hive controllers
in situations where you need to ensure nothing is running that adds or act upon finalizers on Hive
types. This should rarely be needed. Sets replicas to zero for the 'hive-controllers' deployment to

```

```

accomplish this.",
  "type": "boolean"
},
"required": [
  "failedProvisionConfig"
],
"type": "object"
},
"imagePullSecret": {
  "description": "Override pull secret for accessing MultiClusterHub operand and endpoint images.",
  "type": "string"
},
"ingress": {
  "description": "Configuration options for ingress management.",
  "properties": {
    "sslCiphers": {
      "description": "List of SSL ciphers enabled for management ingress. Defaults to full list of supported ciphers.",
      "items": {
        "type": "string"
      },
      "type": "array"
    }
  },
  "type": "object"
},
"nodeSelector": {
  "additionalProperties": {
    "type": "string"
  },
  "description": "Set the node selectors..",
  "type": "object"
},
"overrides": {
  "description": "Developer overrides.",
  "properties": {
    "imagePullPolicy": {
      "description": "Pull policy of the multicluster hub images.",
      "type": "string"
    }
  }
},
"type": "object"
},
"separateCertificateManagement": {
  "description": "(Deprecated) Install cert-manager into its own namespace.",
  "type": "boolean"
},
"type": "object"
},
"status": {
  "description": "MultiClusterHubStatus defines the observed state of MultiClusterHub.",
  "properties": {
    "components": {
      "additionalProperties": {

```

```

    "description": "StatusCondition contains condition information.",
    "properties": {
      "lastTransitionTime": {
        "description": "LastTransitionTime is the last time the condition changed from one status to
another.",
        "format": "date-time",
        "type": "string"
      },
      "message": {
        "description": "Message is a human-readable message indicating\ndetails about the last status
change.",
        "type": "string"
      },
      "reason": {
        "description": "Reason is a (brief) reason for the last status change of the condition.",
        "type": "string"
      },
      "status": {
        "description": "Status is the status of the condition. One of True, False, Unknown.",
        "type": "string"
      },
      "type": {
        "description": "Type is the type of the cluster condition.",
        "type": "string"
      }
    },
    "type": "object"
  },
  "description": "Components []ComponentCondition `json:\`\"manifests,omitempty\`\"",
  "type": "object"
},
  "conditions": {
    "description": "Conditions contain the different condition statuses for the MultiClusterHub.",
    "items": {
      "description": "StatusCondition contains condition information.",
      "properties": {
        "lastTransitionTime": {
          "description": "LastTransitionTime is the last time the condition changed from one status to
another.",
          "format": "date-time",
          "type": "string"
        },
        "lastUpdateTime": {
          "description": "The last time this condition was updated.",
          "format": "date-time",
          "type": "string"
        },
        "message": {
          "description": "Message is a human-readable message indicating details about the last status
change.",
          "type": "string"
        },
        "reason": {
          "description": "Reason is a (brief) reason for the last status change of the condition.",
          "type": "string"
        }
      },
    }
  },

```

```

    "status": {
      "description": "Status is the status of the condition. One of True, False, Unknown.",
      "type": "string"
    },
    "type": {
      "description": "Type is the type of the cluster condition.",
      "type": "string"
    }
  },
  "type": "object"
},
"type": "array"
},
  "currentVersion": {
    "description": "CurrentVersion indicates the current version..",
    "type": "string"
  },
  "desiredVersion": {
    "description": "DesiredVersion indicates the desired version.",
    "type": "string"
  },
  "phase": {
    "description": "Represents the running phase of the MultiClusterHub",
    "type": "string"
  }
},
  "type": "object"
}
},
  "type": "object"
}
},
"served": true,
"storage": true,
"subresources": {
  "status": {}
}
}
]
}
}
}

```

1.13.2.2. 查询所有 MultiClusterHubs

```
GET /operator.open-cluster-management.io/v1beta1/namespaces/{namespace}/operator
```

1.13.2.2.1. 描述

查询您的 multicluster hub operator 以获取更多详细信息。

1.13.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.13.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.13.2.2.4. 使用

- **operator/yaml**

1.13.2.2.5. Tags

- multiclusterhubs.operator.open-cluster-management.io

1.13.2.3. 查询 MultiClusterHub operator

```
GET /operator.open-cluster-management.io/v1beta1/namespaces/{namespace}/operator/{multiclusterhub_name}
```

1.13.2.3.1. 描述

查询单个 multicluster hub operator 以获取更多详细信息。

1.13.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	<code>application_name</code> <i>必需</i>	要查询的应用程序的名称。	字符串
路径	<code>namespace</code> <i>必需</i>	要使用的命名空间，如 default。	字符串

1.13.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.13.2.3.4. Tags

- `multicenterhubs.operator.open-cluster-management.io`

1.13.2.4. 删除 MultiClusterHub operator

```
DELETE /operator.open-cluster-management.io/v1beta1/namespaces/{namespace}/operator/{multicenterhub_name}
```

1.13.2.4.1. 参数

类型	Name	描述	模式
标头	<code>COOKIE</code> <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	<code>application_name</code> <i>必需</i>	要删除的 multicenter hub Operator 的名称。	字符串

类型	Name	描述	模式
路径	namespace 必需	要使用的命名空间，如 default。	字符串

1.13.2.4.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.13.2.4.3. Tags

- `multiclusterhubs.operator.open-cluster-management.io`

1.13.3. 定义

1.13.3.1. multicluster hub operator

Name	描述	模式
apiVersion 必需	MultiClusterHub 的版本化模式。	字符串
kind 必需	代表 REST 资源的字符串值。	字符串
metadata 必需	描述定义资源的规则。	对象
spec 必需	资源规格。	<code>spec</code>

`spec`

availabilityConfig <i>可选</i>	指定部署复制以提高可用性。默认值为 High 。	字符串
customCAConfigmap <i>可选</i>	为 Red Hat Advanced Cluster Management 提供自定义的 OpenShift 默认入口 CA 证书。	字符串
disableHubSelfManagement <i>可选</i>	禁用自动导入 hub 集群作为受管集群。	布尔值
disableUpdateClusterImageSets <i>optional</i>	禁用 ClusterImageSets 的自动更新。	布尔值
hive <i>可选</i>	(已弃用) 用于覆盖默认 HiveConfig 规格的对象。	hive
imagePullSecret <i>optional</i>	覆盖用于访问 MultiClusterHub 操作对象和端点镜像的 pull secret。	字符串
ingress <i>可选</i>	入口管理的配置选项。	ingress
nodeSelector <i>可选</i>	设置节点选择器。	字符串
separateCertificateManagement <i>可选</i>	(已弃用) 将 cert-manager 安装到其自己的命名空间中。	布尔值

hive

additionalCertificateAuthorities <i>可选</i>	(已弃用) 到 hive 命名空间中的 secret 的引用列表，它包含了在与目标集群通信时要使用的额外证书颁发机构 (CA)。除了安装时每个集群生成的任何自签名 CA 外，这些证书颁发机构也被使用。	对象
backup <i>可选</i>	(已弃用) 指定备份集成的配置。如果没有，则禁用备份集成。	backup
externalDNS <i>可选</i>	(已弃用) 如果要由 Hive 部署，指定 external-dns 的配置。如果不存在，则不会部署 external-dns 。	对象

<code>additionalCertificateAuthorities</code> 可选	(已弃用) 到 <code>hive</code> 命名空间中的 <code>secret</code> 的引用列表，它包含了在与目标集群通信时要使用的额外证书颁发机构 (CA)。除了安装时每个集群生成的任何自签名 CA 外，这些证书颁发机构也被使用。	对象
<code>failedProvisionConfigRequired</code>	(已弃用) 用于配置与处理调配失败相关的设置。	failedProvisionConfig
<code>globalPullSecret</code> optional	(已弃用) 用于指定供所有集群部署全局使用的 pull secret。对于每个集群部署， globalPullSecret 的内容都会与集群部署的特定 pull secret 合并 (如果指定)，优先于集群部署的 pull secret 的内容。	对象
<code>maintenanceMode</code> 可选	(已弃用) 在您需要确保没有运行任何会在 Hive 类型的 finalizer 上进行添加或操作时，可以使用它来禁用 hive 控制器。这应该很少需要。将 hive-controllers 部署的 <code>replicas</code> 设置为 0 来完成此操作。	布尔值

ingress

<code>sslCiphers</code> 可选	为管理入口启用的 SSL 密码列表。默认为所支持密码的完整列表。	字符串
-------------------------------	----------------------------------	-----

backup

<code>minBackupPeriodSeconds</code> 可选	(已弃用) 指定在每个备份之间的最小 MinBackupPeriodSeconds 。这用于对备份进行比率限制。这可能会将多个更改组合到一个备份中。因为在这一时间段内发生的更改会被放入排队，在备份间隔时间后这些队列中的更改会被备份，因此不会丢失备份数据。	整数
<code>velero</code> 可选	(已弃用) Velero 指定 Velero 备份集成的配置。	对象

failedProvisionConfig

<code>skipGatherLogs</code> <i>可选</i>	(已弃用) 当安装因为任何原因失败时，禁用尝试从集群收集完整日志的功能。日志会存储在持久卷中，并最多存储七天。	布尔值
--	---	-----

status

<code>components</code> <i>可选</i>	状态配置的组件。	对象
<code>conditions</code> <i>可选</i>	包含 multicluster hub 的不同条件。	conditions
<code>desiredVersion</code> <i>可选</i>	表示所需的版本。	字符串
<code>phase</code> <i>可选</i>	代表 MultiClusterHub 资源的活跃阶段。用于此参数的值包括： Pending 、 Running 、 Installing 、 Updating 、 Uninstalling	字符串

conditions

<code>lastTransitionTime</code> <i>可选</i>	条件从一个状态变为另一个状态的最后一次的时间。	字符串
<code>lastUpdateTime</code> <i>可选</i>	最后一次更新此条件的时间。	字符串
<code>message</code> <i>必需</i>	Message 是人类可读的消息，表示关于上次状态更改的详细信息。	字符串
<code>reason</code> <i>必需</i>	条件发生状态更改的原因。	字符串
<code>status</code> <i>必需</i>	条件的状态。	字符串
<code>type</code> <i>必需</i>	集群条件的类型。	字符串

StatusConditions

kind 必需	代表此状态的资源 kind。	字符串
available 必需	指明此组件是否在正确运行。	布尔值
lastTransitionTime 可选	条件从一个状态变为另一个状态的最后一次的时间。	metav1.time
lastUpdateTime 可选	最后一次更新此条件的时间。	metav1.time
message 必需	Message 是人类可读的消息，表示关于上次状态更改的详细信息。	字符串
reason 可选	条件发生状态更改的原因。	字符串
status 可选	条件的状态。	字符串
type 可选	集群条件的类型。	字符串

1.14. 放置 API (V1BETA1)

1.14.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的放置（Placement）资源相关的 API 信息。放置资源有 4 个可用的请求：create、query、delete 和 update。Placement 定义了一个规则，从绑定到放置命名空间的 ManagedClusterSets 中选择一组 ManagedClusters。会创建一个带有标签 **cluster.open-cluster-management.io/placement={placement name}** 的 PlacementDecisions 片段，以表示由这个放置选择的 ManagedClusters。

1.14.1.1. 版本信息

版本：2.10.0

1.14.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.14.1.3. Tags

- cluster.open-cluster-management.io：创建和管理放置

1.14.2. 路径

1.14.2.1. 查询所有放置

GET /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placement

1.14.2.1.1. 描述

查询您的放置以获取更多详细信息。

1.14.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.14.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.14.2.1.4. 使用

- **placement/yaml**

1.14.2.1.5. Tags

- cluster.open-cluster-management.io

1.14.2.2. 创建一个放置

POST /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placements

1.14.2.2.1. 描述

创建一个放置。

1.14.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的放置绑定的参数。	Placement

1.14.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.14.2.2.4. 使用

- **placement/yaml**

1.14.2.2.5. Tags

- cluster.open-cluster-management.io

1.14.2.2.6. HTTP 请求示例

1.14.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1beta1",
  "kind": "Placement",
  "metadata": {
    "name": "placement1",
    "namespace": "ns1"
  },
  "spec": {
    "predicates": [
      {
        "requiredClusterSelector": {
          "labelSelector": {
            "matchLabels": {
              "vendor": "OpenShift"
            }
          }
        }
      }
    ]
  }
}
```

```

    }
  }
}
]
},
"status" : {}
}

```

1.14.2.3. 查询单个放置

GET /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placements/{placement_name}

1.14.2.3.1. 描述

查询单个放置以获取更多详细信息。

1.14.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	placement_name 必需	要查询的放置名称。	字符串

1.14.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.14.2.3.4. Tags

- cluster.open-cluster-management.io

1.14.2.4. 删除一个放置

```
DELETE /cluster.open-cluster-
management.io/v1beta1/namespaces/{namespace}/placements/{placement_name}
```

1.14.2.4.1. 描述

删除一个放置。

1.14.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	placement_name 必需	要删除的放置名称。	字符串

1.14.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.14.2.4.4. Tags

- cluster.open-cluster-management.io

1.14.3. 定义

1.14.3.1. Placement

Name	描述	模式
apiVersion 必需	放置的版本化模式。	字符串

Name	描述	模式
kind 必需	代表 REST 资源的字符串值。	字符串
metadata 必需	放置的元数据。	对象
spec 必需	放置的规格。	spec

spec

Name	描述	模式
clusterSets 可选	从中选择 ManagedClusters 的 ManagedClusterSets 子集。如果 ManagedClusterSet 为空，则从绑定到 Placement 命名空间的 ManagedClusterSets 中选择 ManagedClusters 。如果 ManagedClusterSet 包含 ManagedClusters ， ManagedClusters 会从这个子集的交集中选择。所选 ManagedClusterSets 绑定到放置命名空间。	字符串数组
numberOfClusters 可选	要选择的 ManagedClusters 数量。	integer (int32)
predicates 可选	选择 ManagedClusters 的集群 predicates 的子集。条件逻辑为 OR。	clusterPredicate 数组
prioritizerPolicy 可选	优先级器的策略。	prioritizerPolicy
tolerations 可选	允许但不是必需的值，通过放置具有匹配容限的放置来选择具有特定污点的受管集群。	容限数组

clusterPredicate

Name	描述	模式
<code>requiredClusterSelector</code> 可选	选择带有标签和集群声明的 ManagedClusters 的集群选择器。	clusterSelector

clusterSelector

Name	描述	模式
<code>labelSelector</code> 可选	按标签的 ManagedClusters 选择器。	对象
<code>claimSelector</code> 可选	按声明的 ManagedClusters 选择器。	clusterClaimSelector

clusterClaimSelector

Name	描述	模式
<code>matchExpressions</code> 可选	集群声明选择器要求的子集。条件逻辑是 <i>AND</i> 。	< object > 数组

prioritizerPolicy

Name	描述	模式
<code>mode</code> 可选	Exact 、 Additive 、或 ""。"" 的默认值为 Additive 。	字符串
<code>configurations</code> 可选	配置优先级。	prioritizerConfig 数组

prioritizerConfig

Name	描述	模式
<code>scoreCoordinate</code> 必需	配置优先级和分数源。	scoreCoordinate
<code>weight</code> 可选	优先级分数的权重。该值的范围： [-10,10]。	int32

scoreCoordinate

Name	描述	模式
type 必需	优先级分数的类型。有效值为 "BuiltIn" 或 "AddOn"。	字符串
builtIn 可选	来自以下选项的 BuiltIn prioritizer 的名称：1) Balance（平衡）：平衡集群之间的决策。2) Steady（稳定）：确保现有决策稳定。3) ResourceAllocatableCPU 和 ResourceAllocatableMemory: 根据可分配资源排序集群。4) Spread（分发）：将工作负载平均到拓扑。	字符串
addOn 可选	当类型为 AddOn 时， AddOn 定义资源名称和分数名称。	对象

容限 (tolerations)

Name	描述	模式
key 可选	容限应用到的污点键。空表示与所有污点键匹配。	字符串
operator 可选	键与值的关系。有效的运算符为 Exists 和 Equal 。默认值为 Equal 。	字符串
value 可选	匹配容限的污点值。	字符串
effect 可选	要匹配的污点效果。空意味着匹配所有污点效果。指定后，允许的值是 NoSelect 、 PreferNoSelect 和 NoSelectIfNew 。	字符串
tolerationSeconds 可选	容许污点的时间长度，之后污点不会被容许。默认值为 nil，这表示污点的容许时间没有时间限制。	int64

1.15. PLACEMENTDECISIONS API (V1BETA1)

1.15.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 PlacementDecision 资源相

关的 API 信息。PlacementDecision 资源有 4 个可用的请求：create、query、delete 和 update。PlacementDecision 表示来自放置的决定。PlacementDecision 使用标签 **cluster.open-cluster-management.io/placement={placement name}** 来引用特定的放置。

1.15.1.1. 版本信息

版本：2.10.0

1.15.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.15.1.3. Tags

- cluster.open-cluster-management.io：创建和管理放置Decisions。

1.15.2. 路径

1.15.2.1. 查询所有 PlacementDecisions

GET /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placementdecisions

1.15.2.1.1. 描述

查询您的 PlacementDecisions 获取更多详细信息。

1.15.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.15.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.15.2.1.4. 使用

- **placementdecision/yaml**

1.15.2.1.5. Tags

- cluster.open-cluster-management.io

1.15.2.2. 创建 PlacementDecision

POST /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placementdecisions

1.15.2.2.1. 描述

创建 PlacementDecision。

1.15.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 PlacementDecision 的参数。	PlacementDecision

1.15.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.15.2.2.4. 使用

- **placementdecision/yaml**

1.15.2.2.5. Tags

- cluster.open-cluster-management.io

1.15.2.2.6. HTTP 请求示例

1.15.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1beta1",
  "kind": "PlacementDecision",
  "metadata": {
    "labels": {
      "cluster.open-cluster-management.io/placement": "placement1"
    },
    "name": "placement1-decision1",
    "namespace": "ns1"
  },
  "status": {}
}
```

1.15.2.3. 查询单个 PlacementDecision

```
GET /cluster.open-cluster-
management.io/v1beta1/namespaces/{namespace}/placementdecisions/{placementdecision_name}
```

1.15.2.3.1. 描述

查询单个 PlacementDecision 获取更多详细信息。

1.15.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	placementdecision_name <i>必需</i>	要查询的 PlacementDecision 的名称。	字符串

1.15.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.15.2.3.4. Tags

- cluster.open-cluster-management.io

1.15.2.4. 删除 PlacementDecision

DELETE /cluster.open-cluster-management.io/v1beta1/namespaces/{namespace}/placementdecisions/{placementdecision_name}

1.15.2.4.1. 描述

删除单个 PlacementDecision。

1.15.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	placementdecision_name 必需	要删除的 PlacementDecision 的名称。	字符串

1.15.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.15.2.4.4. Tags

- cluster.open-cluster-management.io

1.15.3. 定义

1.15.3.1. PlacementDecision

Name	描述	模式
apiVersion <i>必需</i>	PlacementDecision 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	PlacementDecision 的元数据。	对象
status <i>可选</i>	PlacementDecision 的当前状态。	PlacementStatus

PlacementStatus

Name	描述	模式
Decisions <i>必需</i>	根据放置的决策分片。	ClusterDecision 数组

ClusterDecision

Name	描述	模式
clusterName <i>必需</i>	ManagedCluster 的名称。	字符串
reason <i>必需</i>	选择 ManagedCluster 的原因。	字符串

1.16. DISCOVERYCONFIG API

1.16.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 DiscoveryConfig 资源相关的 API 信息。DiscoveryConfig 资源有 4 个可用的请求：create、query、delete 和 update。

1.16.1.1. 版本信息

版本：2.10.0

1.16.1.2. URI scheme

BasePath : /kubernetes/apis

Schemes : HTTPS

1.16.1.3. Tags

- `discoveryconfigs.discovery.open-cluster-management.io` : 创建和管理 DiscoveryConfigs

1.16.2. 路径

1.16.2.1. 创建 DiscoveryConfig

POST /app.k8s.io/v1/namespaces/{namespace}/discoveryconfigs

1.16.2.1.1. 描述

创建 DiscoveryConfig。

1.16.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串
Body	body <i>必需</i>	描述要创建的 DiscoveryConfig 的参数。	DiscoveryConfig

1.16.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.16.2.1.4. 使用

- `discoveryconfigs/yaml`

1.16.2.1.5. Tags

- discoveryconfigs.discovery.open-cluster-management.io

1.16.2.1.5.1. 请求正文

```

{
  "apiVersion": "apiextensions.k8s.io/v1",
  "kind": "CustomResourceDefinition",
  "metadata": {
    "annotations": {
      "controller-gen.kubebuilder.io/version": "v0.4.1",
    },
    "creationTimestamp": null,
    "name": "discoveryconfigs.discovery.open-cluster-management.io",
  },
  "spec": {
    "group": "discovery.open-cluster-management.io",
    "names": {
      "kind": "DiscoveryConfig",
      "listKind": "DiscoveryConfigList",
      "plural": "discoveryconfigs",
      "singular": "discoveryconfig"
    },
    "scope": "Namespaced",
    "versions": [
      {
        "name": "v1",
        "schema": {
          "openAPIV3Schema": {
            "description": "DiscoveryConfig is the Schema for the discoveryconfigs API",
            "properties": {
              "apiVersion": {
                "description": "APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources",
                "type": "string"
              },
              "kind": {
                "description": "Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
                "type": "string"
              },
              "metadata": {
                "type": "object"
              },
              "spec": {
                "description": "DiscoveryConfigSpec defines the desired state of DiscoveryConfig",
                "properties": {
                  "credential": {
                    "description": "Credential is the secret containing credentials to connect to the OCM api on behalf of a user",
                    "type": "string"
                  }
                }
              }
            }
          }
        }
      }
    ]
  }
}

```

```

    },
    "filters": {
      "description": "Sets restrictions on what kind of clusters to discover",
      "properties": {
        "lastActive": {
          "description": "LastActive is the last active in days of clusters to discover, determined
by activity timestamp",
          "type": "integer"
        },
        "openShiftVersions": {
          "description": "OpenShiftVersions is the list of release versions of OpenShift of the form
\"<Major>.<Minor>\"",
          "items": {
            "description": "Semver represents a partial semver string with the major and minor
version in the form \"<Major>.<Minor>\". For example: \"4.13\"",
            "pattern": "^(?:0|[1-9]\\d*)\\.(?:0|[1-9]\\d*)$",
            "type": "string"
          },
          "type": "array"
        }
      },
      "type": "object"
    }
  },
  "required": [
    "credential"
  ],
  "type": "object"
},
"status": {
  "description": "DiscoveryConfigStatus defines the observed state of DiscoveryConfig",
  "type": "object"
}
},
"type": "object"
}
},
"served": true,
"storage": true,
"subresources": {
  "status": {}
}
}
]
},
"status": {
  "acceptedNames": {
    "kind": "",
    "plural": ""
  },
  "conditions": [],
  "storedVersions": []
}
}

```

1.16.2.2. 查询所有 DiscoveryConfigs

GET /operator.open-cluster-management.io/v1/namespaces/{namespace}/operator

1.16.2.2.1. 描述

查询发现配置 operator 以获取更多详细信息。

1.16.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.16.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.16.2.2.4. 使用

- **operator/yaml**

1.16.2.2.5. Tags

- `discoveryconfigs.discovery.open-cluster-management.io`

1.16.2.3. 删除 DiscoveryConfig Operator

DELETE /operator.open-cluster-management.io/v1/namespaces/{namespace}/operator/{discoveryconfigs_name}

1.16.2.3.1. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	application_name <i>必需</i>	要删除的 Discovery Config operator 的名称。	字符串
路径	namespace <i>必需</i>	要使用的命名空间, 如 default。	字符串

1.16.2.3.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.16.2.3.3. Tags

- `discoveryconfigs.operator.open-cluster-management.io`

1.16.3. 定义

1.16.3.1. DiscoveryConfig

Name	描述	模式
apiVersion <i>必需</i>	discoveryconfig 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	描述定义资源的规则。	对象

Name	描述	模式
spec 必需	定义 DiscoveryConfig 所需的状态。	请参阅 specs 列表

1.16.3.2. specs 列表

Name	描述	模式
credential 必需	凭据是包含代表用户连接到 OCM API 的凭证的 secret。	字符串
filters 可选	对要发现的集群类型设置限制。	请参阅 过滤器列表

1.16.3.3. 过滤器列表

Name	描述	模式
lastActive 必需	lastActive 是集群中要发现的最新活跃天数的集群，由活动时间戳决定。	整数
openShiftVersions 可选	OpenShiftVersions 是 OpenShift 的发行版本列表，格式为 "<Major>.<Minor>"	对象

1.17. DISCOVEREDCLUSTER API

1.17.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 DiscoveredCluster 资源相关的 API 资源。DiscoveredCluster 资源有 4 个可用的请求：create、query、delete 和 update。

1.17.1.1. 版本信息

版本：2.10.0

1.17.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.17.1.3. Tags

- DiscoveredClusters.discovery.open-cluster-management.io：创建和管理 DiscoveredClusters

1.17.2. 路径

1.17.2.1. 创建一个 DiscoveredCluster

POST /app.k8s.io/v1/namespaces/{namespace}/discoveredclusters

1.17.2.1.1. 描述

创建一个 DiscoveredCluster。

1.17.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串
Body	body 必需	描述要创建的 DiscoveredCluster 的参数。	DiscoveredCluster

1.17.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.17.2.1.4. 使用

- **discoveredclusters/yaml**

1.17.2.1.5. Tags

- discoveredclusters.discovery.open-cluster-management.io

1.17.2.1.5.1. 请求正文


```

{
  "apiVersion": "apiextensions.k8s.io/v1",
  "kind": "CustomResourceDefinition",
  "metadata": {
    "annotations": {
      "controller-gen.kubebuilder.io/version": "v0.4.1",\
    },
    "creationTimestamp": null,
    "name": "discoveredclusters.discovery.open-cluster-management.io",
  },
  "spec": {
    "group": "discovery.open-cluster-management.io",
    "names": {
      "kind": "DiscoveredCluster",
      "listKind": "DiscoveredClusterList",
      "plural": "discoveredclusters",
      "singular": "discoveredcluster"
    },
    "scope": "Namespaced",
    "versions": [
      {
        "name": "v1",
        "schema": {
          "openAPIV3Schema": {
            "description": "DiscoveredCluster is the Schema for the discoveredclusters API",
            "properties": {
              "apiVersion": {
                "description": "APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources",
                "type": "string"
              },
              "kind": {
                "description": "Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
                "type": "string"
              },
              "metadata": {
                "type": "object"
              },
              "spec": {
                "description": "DiscoveredClusterSpec defines the desired state of DiscoveredCluster",
                "properties": {
                  "activityTimestamp": {
                    "format": "date-time",
                    "type": "string"
                  },
                  "apiUrl": {
                    "type": "string"
                  },
                  "cloudProvider": {
                    "type": "string"
                  }
                }
              }
            }
          }
        }
      }
    ]
  }
}

```

```

"console": {
  "type": "string"
},
"creationTimestamp": {
  "format": "date-time",
  "type": "string"
},
"credential": {
  "description": "ObjectReference contains enough information to let you inspect or modify
the referred object. --- New uses of this type are discouraged because of difficulty describing its
usage when embedded in APIs. 1. Ignored fields. It includes many fields which are not generally
honored. For instance, ResourceVersion and FieldPath are both very rarely valid in actual usage. 2.
Invalid usage help. It is impossible to add specific help for individual usage. In most embedded
usages, there are particular restrictions like, \"must refer only to types A and B\" or \"UID not
honored\" or \"name must be restricted\". Those cannot be well described when embedded. 3.
Inconsistent validation. Because the usages are different, the validation rules are different by usage,
which makes it hard for users to predict what will happen. 4. The fields are both imprecise and overly
precise. Kind is not a precise mapping to a URL. This can produce ambiguity during interpretation
and require a REST mapping. In most cases, the dependency is on the group,resource tuple and
the version of the actual struct is irrelevant. 5. We cannot easily change it. Because this type is
embedded in many locations, updates to this type will affect numerous schemas. Don't make new
APIs embed an underspecified API type they do not control. Instead of using this type, create a
locally provided and used type that is well-focused on your reference. For example,
ServiceReferences for admission registration: https://github.com/kubernetes/api/blob/release-
1.17/admissionregistration/v1/types.go#L533 .",
  "properties": {
    "apiVersion": {
      "description": "API version of the referent.",
      "type": "string"
    },
    "fieldPath": {
      "description": "If referring to a piece of an object instead of an entire object, this string
should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2].
For example, if the object reference is to a container within a pod, this would take on a value like:
\"spec.containers{name}\" (where \"name\" refers to the name of the container that triggered the
event) or if no container name is specified \"spec.containers[2]\" (container with index 2 in this pod).
This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO:
this design is not final and this field is subject to change in the future.",
      "type": "string"
    },
    "kind": {
      "description": "Kind of the referent. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
      "type": "string"
    },
    "name": {
      "description": "Name of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
      "type": "string"
    },
    "namespace": {
      "description": "Namespace of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/",
      "type": "string"
    },
    "resourceVersion": {

```

```

        "description": "Specific resourceVersion to which this reference is made, if any. More
info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-
control-and-consistency",
        "type": "string"
    },
    "uid": {
        "description": "UID of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids",
        "type": "string"
    }
},
"type": "object"
},
"displayName": {
    "type": "string"
},
"isManagedCluster": {
    "type": "boolean"
},
"name": {
    "type": "string"
},
"openshiftVersion": {
    "type": "string"
},
"status": {
    "type": "string"
},
"type": {
    "type": "string"
}
},
"required": [
    "apiUrl",
    "displayName",
    "isManagedCluster",
    "name",
    "type"
],
"type": "object"
},
"status": {
    "description": "DiscoveredClusterStatus defines the observed state of DiscoveredCluster",
    "type": "object"
}
},
"type": "object"
}
},
"served": true,
"storage": true,
"subresources": {
    "status": {}
}
}
]

```

```

    },
    "status": {
      "acceptedNames": {
        "kind": "",
        "plural": ""
      },
    },
    "conditions": [],
    "storedVersions": []
  }
}

```

1.17.2.2. 查询所有 DiscoveredClusters

GET /operator.open-cluster-management.io/v1/namespaces/{namespace}/operator

1.17.2.2.1. 描述

查询发现的集群 operator 以获取更多详细信息。

1.17.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.17.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.17.2.2.4. 使用

- **operator/yaml**

1.17.2.2.5. Tags

- discoveredclusters.discovery.open-cluster-management.io

1.17.2.3. 删除 DiscoveredCluster Operator

```
DELETE /operator.open-cluster-management.io/v1/namespaces/{namespace}/operator/{discoveredclusters_name}
```

1.17.2.3.1. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	application_name 必需	要删除的发现的 Cluster Operator 的名称。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.17.2.3.2. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.17.2.3.3. Tags

- discoveredclusters.operator.open-cluster-management.io

1.17.3. 定义

1.17.3.1. DiscoveredCluster

Name	描述	模式
apiVersion <i>必需</i>	发现集群的版本化 schema。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	描述定义资源的规则。	对象
spec <i>必需</i>	DiscoveredClusterSpec 定义所需的 DiscoveredCluster 状态。	请参阅 <i>specs 列表</i>

1.17.3.2. specs 列表

Name	描述	模式
activityTimestamp <i>可选</i>	DiscoveredClusters 最后可用活动时间戳。	metav1.time
apiUrl <i>必需</i>	DiscoveredClusters API URL 端点。	字符串
cloudProvider <i>可选</i>	发现集群的云供应商。	字符串
console <i>可选</i>	DiscoveredClusters 控制台 URL 端点。	字符串
creationTimestamp <i>可选</i>	DiscoveredCluster 创建时间戳。	metav1.time
credential <i>可选</i>	对从中发现集群的凭证的引用。	corev1.ObjectReference
displayName <i>必需</i>	发现集群的显示名称。	字符串
isManagedCluster <i>必需</i>	如果为 true，集群由 ACM 管理。	布尔值
name <i>必需</i>	discoveredcluster 的名称。	字符串
openshiftVersion <i>可选</i>	发现集群的 OpenShift 版本。	字符串

Name	描述	模式
status 可选	发现集群的状态。	字符串
type 必需	OpenShift 类别（例如，OCP、ROSA 等）。	字符串

1.18. ADDONDEPLOYMENTCONFIG API (V1ALPHA1)

1.18.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 AddOnDeploymentConfig 资源相关的 API 信息。AddOnDeploymentConfig 资源有 4 个可用的请求：create、query、delete 和 update。AddOnDeploymentConfig 代表附加组件的部署配置。

1.18.1.1. 版本信息

版本：2.10.0

1.18.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.18.1.3. Tags

- addon.open-cluster-management.io：创建和管理 AddOnDeploymentConfig

1.18.2. 路径

1.18.2.1. 查询所有 AddOnDeploymentConfig

GET /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/addondeploymentconfigs

1.18.2.1.1. 描述

查询您的 AddOnDeploymentConfig 以获取更多详细信息。

1.18.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串

1.18.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.18.2.1.4. 使用

- **addondeploymentconfig/yaml**

1.18.2.1.5. Tags

- `addon.open-cluster-management.io`

1.18.2.2. 创建 AddOnDeploymentConfig

POST /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/addondeploymentconfigs

1.18.2.2.1. 描述

创建 AddOnDeploymentConfig。

1.18.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 AddOnDeploymentConfig 绑定的参数。	AddOnDeploymentConfig

1.18.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.18.2.2.4. 使用

- **addondeploymentconfig/yaml**

1.18.2.2.5. Tags

- `addon.open-cluster-management.io`

1.18.2.2.6. HTTP 请求示例

1.18.2.2.6.1. 请求正文

```
{
  "apiVersion": "addon.open-cluster-management.io/v1alpha1",
  "kind": "AddOnDeploymentConfig",
  "metadata": {
    "name": "deploy-config",
    "namespace": "open-cluster-management-hub"
  },
  "spec": {
    "nodePlacement": {
      "nodeSelector": {
        "node-dedicated": "acm-addon"
      },
      "tolerations": [
        {
          "effect": "NoSchedule",
          "key": "node-dedicated",
          "operator": "Equal",
          "value": "acm-addon"
        }
      ]
    }
  }
}
```

1.18.2.3. 查询单个 AddOnDeploymentConfig

```
GET /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/addondeploymentconfigs/{addondeploymentconfig_name}
```

1.18.2.3.1. 描述

查询单个 AddOnDeploymentConfig 获取更多详细信息。

1.18.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	addondeploymentconfig_name 必需	要查询的 AddOnDeploymentConfig 的名称。	字符串

1.18.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.18.2.3.4. Tags

- `addon.open-cluster-management.io`

1.18.2.4. 删除 AddOnDeploymentConfig

```
DELETE /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/addondeploymentconfigs/{addondeploymentconfig_name}
```

1.18.2.4.1. 描述

删除单个 AddOnDeploymentConfig。

1.18.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	addondeploymentconfig_name <i>必需</i>	要删除的 AddOnDeploymentConfig 的名称。	字符串

1.18.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.18.2.4.4. Tags

- `addon.open-cluster-management.io`

1.18.3. 定义

1.18.3.1. AddOnDeploymentConfig

Name	描述	模式
apiVersion <i>必需</i>	AddOnDeploymentConfig 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	AddOnDeploymentConfig 的元数据。	object

Name	描述	模式
spec 必需	AddOnDeploymentConfig 的规格。	spec

spec

Name	描述	模式
customizedVariables 可选	当前附加组件部署的 name-value 变量列表。附加组件实施可以使用这些变量呈现其附加组件部署。	customizedVariable 数组
nodePlacement 必需	启用对受管集群中附加组件代理调度的显式控制。	nodePlacement

customizedVariable

Name	描述	模式
name 必需	此变量的名称。	字符串
value 可选	此变量的值。	字符串

nodePlacement

Name	描述	模式
nodeSelector 可选	定义 pod 调度到运行的节点。当 nodeSelector 为空时， nodeSelector 会选择所有节点。	map[string]string
tolerations 可选	应用到 pod，并使用匹配的运算符 (<operator>) 将 pod 调度到与 <key,value,effect> 容限匹配的污点。	[]corev1.Toleration

1.19. CLUSTERMANAGEMENTADDON API (V1ALPHA1)

1.19.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 ClusterManagementAddOn 资源相关的 API 信息。ClusterManagementAddOn 资源有 4 个可用的请求：create、query、delete 和 update。

ClusterManagementAddOn 代表集群管理器的附加组件注册。此资源允许用户发现哪些附加组件可用于集群管理器，并提供有关附加组件的元数据信息。此资源还提供对 ManagedClusterAddOn 的引用，即用于命名空间范围的 ManagedClusterAddOn 资源的 ClusterManagementAddOn 资源的名称。ClusterManagementAddOn 是一个集群范围的资源。

1.19.1.1. 版本信息

版本：2.10.0

1.19.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.19.1.3. Tags

- addon.open-cluster-management.io：创建和管理 ClusterManagementAddOns

1.19.2. 路径

1.19.2.1. 查询所有 ClusterManagementAddOns

GET /addon.open-cluster-management.io/v1alpha1/clustermanagementaddons

1.19.2.1.1. 描述

查询 ClusterManagementAddOns 获取更多详细信息。

1.19.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证：Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.19.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容

HTTP 代码	描述	模式
500	内部服务错误	无内容
503	服务不可用	无内容

1.19.2.1.4. 使用

- `clustermanagementaddon/yaml`

1.19.2.1.5. Tags

- `addon.open-cluster-management.io`

1.19.2.2. 创建 ClusterManagementAddOn

POST /addon.open-cluster-management.io/v1alpha1/clustermanagementaddons

1.19.2.2.1. 描述

创建 ClusterManagementAddOn。

1.19.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 ClusterManagementAddOn 绑定的参数。	ClusterManagementAddOn

1.19.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.19.2.2.4. 使用

- `clustermanagementaddon/yaml`

1.19.2.2.5. Tags

- `addon.open-cluster-management.io`

1.19.2.2.6. HTTP 请求示例

1.19.2.2.6.1. 请求正文

```
{
  "apiVersion": "addon.open-cluster-management.io/v1alpha1",
  "kind": "ClusterManagementAddOn",
  "metadata": {
    "name": "helloworld"
  },
  "spec": {
    "supportedConfigs": [
      {
        "defaultConfig": {
          "name": "deploy-config",
          "namespace": "open-cluster-management-hub"
        },
        "group": "addon.open-cluster-management.io",
        "resource": "addondeploymentconfigs"
      }
    ]
  },
  "status": {}
}
```

1.19.2.3. 查询单个 ClusterManagementAddOn

```
GET /addon.open-cluster-
management.io/v1alpha1/clustermanagementaddons/{clustermanagementaddon_name}
```

1.19.2.3.1. 描述

查询单个 ClusterManagementAddOn 获取更多详细信息。

1.19.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	<code>clustermanagementaddon_name</code> <i>required</i>	要查询的 ClusterManagementAddOn 的名称。	字符串

1.19.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.19.2.3.4. Tags

- `addon.open-cluster-management.io`

1.19.2.4. 删除 ClusterManagementAddOn

DELETE /addon.open-cluster-management.io/v1alpha1/clustermanagementaddons/{clustermanagementaddon_name}

1.19.2.4.1. 描述

删除单个 ClusterManagementAddOn。

1.19.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	<code>clustermanagementaddon_name</code> <i>required</i>	要删除的 ClusterManagementAddOn 的名称。	字符串

1.19.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.19.2.4.4. Tags

- `addon.open-cluster-management.io`

1.19.3. 定义

1.19.3.1. ClusterManagementAddOn

Name	描述	模式
apiVersion <i>必需</i>	ClusterManagementAddOn 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	ClusterManagementAddOn 的元数据。	object
spec <i>必需</i>	ClusterManagementAddOn 的规格。	spec

spec

Name	描述	模式
addOnMeta <i>optional</i>	AddOnMeta 是对附加组件的元数据信息的引用。	addOnMeta
supportedConfigs <i>可选</i>	SupportedConfigs 是附加组件支持的配置类型列表。	configMeta 数组

addOnMeta

Name	描述	模式
displayName <i>可选</i>	表示显示的附加组件名称。	字符串
description <i>可选</i>	代表附加组件的详细描述。	字符串

configMeta

Name	描述	模式
group <i>可选</i>	附加组件配置的组。	字符串
resource <i>必需</i>	附加组件配置的资源。	字符串
defaultConfig <i>必需</i>	代表默认附加组件配置的命名空间和名称。这是所有附加组件都有相同的配置的位置。	configReferent

configReferent

Name	描述	模式
namespace <i>可选</i>	附加组件配置的命名空间。如果没有设置此字段，则配置是 cluster-scope。	字符串
name <i>必需</i>	附加组件配置的名称。	字符串

1.20. MANAGEDCLUSTERADDON API (V1ALPHA1)

1.20.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 ManagedClusterAddOn 资源相关的 API 信息。ManagedClusterAddOn 资源有 4 个可用的请求：create、query、delete 和 update。ManagedClusterAddOn 是包含附加组件当前状态的自定义资源对象。此资源应该在 ManagedCluster 命名空间中创建。

1.20.1.1. 版本信息

版本: 2.10.0

1.20.1.2. URI scheme

BasePath : /kubernetes/apis

Schemes : HTTPS

1.20.1.3. Tags

- `addon.open-cluster-management.io` : 创建和管理 `ManagedClusterAddOns`

1.20.2. 路径

1.20.2.1. 查询所有 `ManagedClusterAddOns`

GET /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/managedclusteraddons

1.20.2.1.1. 描述

查询 `ManagedClusterAddOns` 获取更多详细信息。

1.20.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.20.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.20.2.1.4. 使用

- `managedclusteraddon/yaml`

1.20.2.1.5. Tags

- `addon.open-cluster-management.io`

1.20.2.2. 创建 ManagedClusterAddOn

POST /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/managedclusteraddons

1.20.2.2.1. 描述

创建 ManagedClusterAddOn。

1.20.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 ManagedClusterAddOn 绑定的参数。	ManagedClusterAddOn

1.20.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.20.2.2.4. 使用

- `managedclusteraddon/yaml`

1.20.2.2.5. Tags

- `addon.open-cluster-management.io`

1.20.2.2.6. HTTP 请求示例

1.20.2.2.6.1. 请求正文

```

{
  "apiVersion": "addon.open-cluster-management.io/v1alpha1",
  "kind": "ManagedClusterAddOn",
  "metadata": {
    "name": "helloworld",
    "namespace": "cluster1"
  },
  "spec": {
    "configs": [
      {
        "group": "addon.open-cluster-management.io",
        "name": "cluster-deploy-config",
        "namespace": "open-cluster-management-hub",
        "resource": "addondeploymentconfigs"
      }
    ],
    "installNamespace": "default"
  },
  "status": {}
}

```

1.20.2.3. 查询单个 ManagedClusterAddOn

GET /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/managedclusteraddons/{managedclusteraddon_name}

1.20.2.3.1. 描述

查询单个 ManagedClusterAddOn 获取更多详细信息。

1.20.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	managedclusteraddon_name 必需	要查询的 ManagedClusterAddOn 的名称。	字符串

1.20.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容

HTTP 代码	描述	模式
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.20.2.3.4. Tags

- `addon.open-cluster-management.io`

1.20.2.4. 删除 ManagedClusterAddOn

DELETE /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/managedclusteraddons/{managedclusteraddon_name}

1.20.2.4.1. 描述

删除单个 ManagedClusterAddOn。

1.20.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	managedclusteraddon_name 必需	要删除的 ManagedClusterAddOn 的名称。	字符串

1.20.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容

HTTP 代码	描述	模式
503	服务不可用	无内容

1.20.2.4.4. Tags

- `addon.open-cluster-management.io`

1.20.3. 定义

1.20.3.1. ManagedClusterAddOn

Name	描述	模式
<code>apiVersion</code> <i>必需</i>	ManagedClusterAddOn 的版本化模式。	字符串
<code>kind</code> <i>必需</i>	代表 REST 资源的字符串值。	字符串
<code>metadata</code> <i>必需</i>	ManagedClusterAddOn 的元数据。	object
<code>spec</code> <i>必需</i>	ManagedClusterAddOn 的规格。	spec

spec

Name	描述	模式
<code>installNamespace</code> <i>可选</i>	受管集群上的命名空间，用于安装附加组件代理。如果没有设置，则使用 <code>open-cluster-management-agent-addon</code> 命名空间来安装附加组件代理。	字符串
<code>configs</code> <i>可选</i>	当前附加组件有其自身配置的附加组件配置列表。	addOnConfig 数组

addOnConfig

Name	描述	模式
<code>group</code> <i>可选</i>	附加组件配置的组。	字符串

Name	描述	模式
resource 必需	附加组件配置的资源。	字符串
namespace 可选	附加组件配置的命名空间。如果没有设置此字段，则配置是 cluster-scope。	字符串
name 必需	附加组件配置的名称。	字符串

1.21. MANAGEDCLUSTERSET API (V1BETA2)

1.21.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 ManagedClusterSet 资源相关的 API 信息。ManagedClusterSet 资源有 4 个可用的请求：create、query、delete 和 update。ManagedClusterSet 将两个或多个受管集群分组到可一起工作的集合。属于集合的受管集群可以有类似的属性，如共享使用目的或同一部署区域。

1.21.1.1. 版本信息

版本：2.10.0

1.21.1.2. URI scheme

BasePath：/kubernetes/apis

Schemes：HTTPS

1.21.1.3. Tags

- cluster.open-cluster-management.io：创建和管理 ManagedClusterSets

1.21.2. 路径

1.21.2.1. 查询所有 managedclustersets

```
GET /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclustersets
```

1.21.2.1.1. 描述

查询 managedclustersets 以获取更多详细信息。

1.21.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default。	字符串

1.21.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.21.2.1.4. 使用

- **managedclusterset/yaml**

1.21.2.1.5. Tags

- `cluster.open-cluster-management.io`

1.21.2.2. 创建 managedclusterset

POST /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclustersets

1.21.2.2.1. 描述

创建 managedclusterset.

1.21.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

类型	Name	描述	模式
路径	namespace <i>必需</i>	要使用的命名空间，如 default 。	字符串
Body	body <i>必需</i>	描述要创建的 managedclusterset 的参数。	ManagedClusterSet

1.21.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.21.2.2.4. 使用

- **managedclusterset/yaml**

1.21.2.2.5. Tags

- cluster.open-cluster-management.io

1.21.2.2.6. HTTP 请求示例

1.21.2.2.6.1. 请求正文

```
{
  "apiVersion": "cluster.open-cluster-management.io/v1beta2",
  "kind": "ManagedClusterSet",
  "metadata": {
    "name": "example-clusterset",
  },
  "spec": {
  },
  "status": {}
}
```

1.21.2.3. 查询单个 managedclusterset

GET /cluster.open-cluster-management.io/v1beta2/namespaces/{namespace}/managedclustersets/{managedclusterset_name}

1.21.2.3.1. 描述

查询单个 managedclusterset 获取更多详细信息。

1.21.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace 必需	要使用的命名空间, 如 default 。	字符串
路径	managedclusterset_name 必需	要查询的 managedclusterset 的名称。	字符串

1.21.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.21.2.3.4. Tags

- cluster.open-cluster-management.io

1.21.2.4. 创建 managedclusterset.

DELETE /cluster.open-cluster-management.io/v1beta2/managedclustersets/{managedclusterset_name}

1.21.2.4.1. 描述

删除单个 managedclusterset

1.21.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证：Bearer {ACCESS_TOKEN}； ACCESS_TOKEN 是用户访问令牌。	字符串
路径	namespace <i>必需</i>	要使用的命名空间，如 default 。	字符串
路径	managedclusterset_name <i>必需</i>	要删除的 managedclusterset 的名称。	字符串

1.21.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.21.2.4.4. Tags

- cluster.open-cluster-management.io

1.21.3. 定义

1.21.3.1. ManagedClusterSet

Name	描述	模式
apiVersion <i>必需</i>	ManagedClusterSet 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	ManagedClusterSet 的元数据。	object

Name	描述	模式
spec必需	ManagedClusterSet 的规格。	spec

1.22. KLUSTERLETCONFIG API (V1ALPHA1)

1.22.1. 概述

本文档介绍了与 Red Hat Advanced Cluster Management for Kubernetes 的 `KlusterletConfig` 资源相关的 API 信息。`KlusterletConfig` 资源有 4 个可用的请求：`create`、`query`、`delete` 和 `update`。`KlusterletConfig` 包含有关 `klusterlet` 的配置信息，如 **nodeSelector**、**tolerations** 和 **pullSecret**。`KlusterletConfig` 是一个集群范围的资源，仅适用于 **open-cluster-managemnet-agent** 命名空间中的 `klusterlet pod`。`KlusterletConfig` 不会影响附加部署配置。

1.22.1.1. 版本信息

版本: 2.10.0

1.22.1.2. URI scheme

BasePath: /kubernetes/apis
Schemes: HTTPS

1.22.1.3. Tags

- `config.open-cluster-management.io`: 创建和管理 `KlusterletConfig`

1.22.2. 路径

1.22.2.1. 查询所有 `KlusterletConfig`

GET /config.open-cluster-management.io/v1alpha1/namespaces/{namespace}/klusterletconfigs

1.22.2.1.1. 描述

查询 `KlusterletConfigs` 以获取更多详细信息。

1.22.2.1.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证: Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串

1.22.2.1.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.22.2.1.4. 使用

- `klusterletconfig/yaml`

1.22.2.1.5. Tags

- `config.open-cluster-management.io`

1.22.2.2. 创建 KlusterletConfig

POST /config.open-cluster-management.io/v1alpha1/namespaces/{namespace}/klusterletconfigs

1.22.2.2.1. 描述

创建 KlusterletConfig。

1.22.2.2.2. 参数

类型	Name	描述	模式
标头	COOKIE 必需	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
Body	body 必需	描述要创建的 KlusterletConfig 绑定的参数。	KlusterletConfig

1.22.2.2.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容

HTTP 代码	描述	模式
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.22.2.2.4. 使用

- **klusterletconfig/yaml**

1.22.2.2.5. Tags

- config.open-cluster-management.io

1.22.2.2.6. HTTP 请求示例

1.22.2.2.6.1. 请求正文

```
{
  "apiVersion": "apiextensions.k8s.io/v1",
  "kind": "CustomResourceDefinition",
  "metadata": {
    "annotations": {
      "controller-gen.kubebuilder.io/version": "v0.7.0"
    },
    "creationTimestamp": null,
    "name": "klusterletconfigs.config.open-cluster-management.io"
  },
  "spec": {
    "group": "config.open-cluster-management.io",
    "names": {
      "kind": "KlusterletConfig",
      "listKind": "KlusterletConfigList",
      "plural": "klusterletconfigs",
      "singular": "klusterletconfig"
    },
    "preserveUnknownFields": false,
    "scope": "Cluster",
    "versions": [
      {
        "name": "v1alpha1",
        "schema": {
          "openAPIV3Schema": {
            "description": "KlusterletConfig contains the configuration of a klusterlet including the upgrade strategy, config overrides, proxy configurations etc.",
            "properties": {
              "apiVersion": {
                "description": "APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized"

```

```

values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources,
  "type": "string"
},
"kind": {
  "description": "Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
  "type": "string"
},
"metadata": {
  "type": "object"
},
"spec": {
  "description": "Spec defines the desired state of KlusterletConfig",
  "properties": {
    "hubKubeAPIServerProxyConfig": {
      "description": "HubKubeAPIServerProxyConfig holds proxy settings for connections between klusterlet/add-on agents on the managed cluster and the kube-apiserver on the hub cluster. Empty means no proxy settings is available.",
      "properties": {
        "caBundle": {
          "description": "CABundle is a CA certificate bundle to verify the proxy server. It will be ignored if only HTTPProxy is set; And it is required when HTTPSProxy is set and self signed CA certificate is used by the proxy server.",
          "format": "byte",
          "type": "string"
        },
        "httpProxy": {
          "description": "HTTPProxy is the URL of the proxy for HTTP requests",
          "type": "string"
        },
        "httpsProxy": {
          "description": "HTTPSProxy is the URL of the proxy for HTTPS requests HTTPSProxy will be chosen if both HTTPProxy and HTTPSProxy are set.",
          "type": "string"
        }
      },
      "type": "object"
    },
    "nodePlacement": {
      "description": "NodePlacement enables explicit control over the scheduling of the agent components. If the placement is nil, the placement is not specified, it will be omitted. If the placement is an empty object, the placement will match all nodes and tolerate nothing.",
      "properties": {
        "nodeSelector": {
          "additionalProperties": {
            "type": "string"
          },
          "description": "NodeSelector defines which Nodes the Pods are scheduled on. The default is an empty list.",
          "type": "object"
        },
        "tolerations": {
          "description": "Tolerations is attached by pods to tolerate any taint that matches the

```


triple <key,value,effect> using the matching operator <operator>. The default is an empty list.",

```

    "items": {
      "description": "The pod this Toleration is attached to tolerates any taint that matches
the triple <key,value,effect> using the matching operator <operator>.",
      "properties": {
        "effect": {
          "description": "Effect indicates the taint effect to match. Empty means match all
taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.",
          "type": "string"
        },
        "key": {
          "description": "Key is the taint key that the toleration applies to. Empty means
match all taint keys. If the key is empty, operator must be Exists; this combination means to match all
values and all keys.",
          "type": "string"
        },
        "operator": {
          "description": "Operator represents a key's relationship to the value. Valid operators
are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can
tolerate all taints of a particular category.",
          "type": "string"
        },
        "tolerationSeconds": {
          "description": "TolerationSeconds represents the period of time the toleration
(which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is
not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated
as 0 (evict immediately) by the system.",
          "format": "int64",
          "type": "integer"
        },
        "value": {
          "description": "Value is the taint value the toleration matches to. If the operator is
Exists, the value should be empty, otherwise just a regular string.",
          "type": "string"
        }
      },
      "type": "object"
    },
    "type": "array"
  }
},
"type": "object"
},
"pullSecret": {
  "description": "PullSecret is the name of image pull secret.",
  "properties": {
    "apiVersion": {
      "description": "API version of the referent.",
      "type": "string"
    },
    "fieldPath": {
      "description": "If referring to a piece of an object instead of an entire object, this string
should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2].
For example, if the object reference is to a container within a pod, this would take on a value like:
'spec.containers{name}' (where 'name' refers to the name of the container that triggered the
event) or if no container name is specified 'spec.containers[2]' (container with index 2 in this pod).

```

This syntax is chosen only to have some well-defined way of referencing a part of an object. TODO: this design is not final and this field is subject to change in the future.",

```

    "type": "string"
  },
  "kind": {
    "description": "Kind of the referent. More info:
https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds",
    "type": "string"
  },
  "name": {
    "description": "Name of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names",
    "type": "string"
  },
  "namespace": {
    "description": "Namespace of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/",
    "type": "string"
  },
  "resourceVersion": {
    "description": "Specific resourceVersion to which this reference is made, if any. More
info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-
control-and-consistency",
    "type": "string"
  },
  "uid": {
    "description": "UID of the referent. More info:
https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids",
    "type": "string"
  }
},
"type": "object"
},
"registries": {
  "description": "Registries includes the mirror and source registries. The source registry
will be replaced by the Mirror.",
  "items": {
    "properties": {
      "mirror": {
        "description": "Mirror is the mirrored registry of the Source. Will be ignored if Mirror is
empty.",
        "type": "string"
      },
      "source": {
        "description": "Source is the source registry. All image registries will be replaced by
Mirror if Source is empty.",
        "type": "string"
      }
    }
  },
  "required": [
    "mirror"
  ],
  "type": "object"
},
"type": "array"
}

```

```

    },
    "type": "object"
  },
  "status": {
    "description": "Status defines the observed state of KlusterletConfig",
    "type": "object"
  }
}
},
"type": "object"
}
},
"served": true,
"storage": true,
"subresources": {
  "status": {}
}
}
],
},
"status": {
  "acceptedNames": {
    "kind": "",
    "plural": ""
  },
  "conditions": [],
  "storedVersions": []
}
}
}

```

1.22.2.3. 查询单个 KlusterletConfig

GET /config.open-cluster-management.io/v1alpha1/namespaces/{namespace}/klusterletconfigs/{klusterletconfig_name}

1.22.2.3.1. 描述

查询单个 KlusterletConfig 获取更多详细信息。

1.22.2.3.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	klusterletconfig_name <i>required</i>	要查询的 KlusterletConfig 的名称。	字符串

1.22.2.3.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容
500	内部服务错误	无内容
503	服务不可用	无内容

1.22.2.3.4. Tags

- config.open-cluster-management.io

1.22.2.4. 删除 KlusterletConfig

```
DELETE /addon.open-cluster-management.io/v1alpha1/namespaces/{namespace}/klusterletconfigs/{klusterletconfig_name}
```

1.22.2.4.1. 描述

删除单个 klusterletconfig。

1.22.2.4.2. 参数

类型	Name	描述	模式
标头	COOKIE <i>必需</i>	身份验证 : Bearer {ACCESS_TOKEN}; ACCESS_TOKEN 是用户访问令牌。	字符串
路径	klusterletconfig_name <i>required</i>	要删除的 KlusterletConfig 的名称。	字符串

1.22.2.4.3. 响应

HTTP 代码	描述	模式
200	成功	无内容
403	禁止访问	无内容
404	未找到资源	无内容

HTTP 代码	描述	模式
500	内部服务错误	无内容
503	服务不可用	无内容

1.22.2.4.4. Tags

- `config.open-cluster-management.io`

1.22.3. 定义

1.22.3.1. KlusterletConfig

Name	描述	模式
apiVersion <i>必需</i>	KlusterletConfig 的版本化模式。	字符串
kind <i>必需</i>	代表 REST 资源的字符串值。	字符串
metadata <i>必需</i>	KlusterletConfig 的元数据。	object
spec <i>必需</i>	KlusterletConfig 的规格。	spec

spec

Name	描述	模式
registry <i>可选</i>	包括镜像和源 registry。源 registry 被镜像替代。	registry
pullSecret <i>可选</i>	镜像 pull secret 的名称。	object
nodePlacement <i>必需</i>	在受管集群中启用附加代理的调度控制。	nodePlacement
hubKubeAPIServerProxyConfig <i>required</i>	包含受管集群中 klusterlet 或附加代理和 hub 集群上的 kube-apiserver 之间的连接的代理设置。空表示没有可用的代理设置。	kubeAPIServerProxyConfig

nodePlacement

Name	描述	模式
nodeSelector <i>可选</i>	定义 pod 调度到运行的节点。当 nodeSelector 为空时， nodeSelector 会选择所有节点。	map[string]string
tolerations <i>可选</i>	应用到 pod，并使用匹配的运算符 (<operator>) 将 pod 调度到与 <key,value,effect> 容忍匹配的污点。	[]corev1.Toleration

kubeAPIServerProxyConfig

Name	描述	模式
cabundle <i>可选</i>	用于验证代理服务器的 CA 证书捆绑包。如果只设置了 HTTPProxy，则捆绑包将被忽略。当设置 HTTPSProxy 且代理服务器使用自签名 CA 证书时，需要捆绑包。	map[string]string
httpProxy <i>可选</i>	HTTP 请求的代理的 URL	map[string]string
httpsProxy <i>可选</i>	HTTPS 请求的代理的 URL。如果设置了 HTTPProxy 和 HTTPSProxy，则会选择 httpsProxy。	map[string]string

1.23. 策略合规历史记录（技术预览）

1.23.1. 概述

如果您希望以可查询的格式进行 Red Hat Advanced Cluster Management for Kubernetes 策略合规事件的长期存储，策略合规历史记录 API 是一个可选的技术预览功能。您可以使用 API 获取其他详情，如 **spec** 字段审核并排除您的策略，并在策略从集群中删除或从集群中删除时获取合规事件。策略合规历史记录 API 也可以生成以逗号分隔的策略合规事件表，以帮助您进行审计和故障排除。

1.23.1.1. 版本信息

版本：2.10.0

1.23.2. API 端点

1.23.2.1. 列出策略合规事件

/api/v1/compliance-events

这将列出您默认能够访问的所有策略合规事件。响应格式如下，默认按 **event.timestamp** 进行排序：

```

{
  "data": [
    {
      "id": 2,
      "cluster": {
        "name": "cluster1",
        "cluster_id": "215ce184-8dee-4cab-b99b-1f8f29dff611"
      },
      "parent_policy": {
        "id": 3,
        "name": "configure-custom-app",
        "namespace": "policies",
        "categories": ["CM Configuration Management"],
        "controls": ["CM-2 Baseline Configuration"],
        "standards": ["NIST SP 800-53"]
      },
      "policy": {
        "apiGroup": "policy.open-cluster-management.io",
        "id": 2,
        "kind": "ConfigurationPolicy",
        "name": "configure-custom-app",
        "namespace": "",
        // Only shown with `?include_spec`
        "spec": {}
      },
      "event": {
        "compliance": "NonCompliant",
        "message": "configmaps [app-data] not found in namespace default",
        "timestamp": "2023-07-19T18:25:43.511Z",
        "metadata": {}
      }
    },
    {
      "id": 1,
      "cluster": {
        "name": "cluster2",
        "cluster_id": "415ce234-8dee-4cab-b99b-1f8f29dff461"
      },
      "parent_policy": {
        "id": 3,
        "name": "configure-custom-app",
        "namespace": "policies",
        "categories": ["CM Configuration Management"],
        "controls": ["CM-2 Baseline Configuration"],
        "standards": ["NIST SP 800-53"]
      },
      "policy": {
        "apiGroup": "policy.open-cluster-management.io",
        "id": 4,
        "kind": "ConfigurationPolicy",
        "name": "configure-custom-app",
        "namespace": "",

```

```
// Only shown with `?include_spec`
"spec": {}
},
"event": {
  "compliance": "Compliant",
  "message": "configmaps [app-data] found as specified in namespace default",
  "timestamp": "2023-07-19T18:25:41.523Z",
  "metadata": {}
}
},
"metadata": {
  "page": 1,
  "pages": 7,
  "per_page": 20,
  "total": 123
}
}
```

以下可选查询参数被接受。请注意，没有描述的那些只过滤其引用的字段。参数值 **null** 代表没有值。另外，可以使用逗号指定多个值。例如，`?cluster.name=cluster1,cluster2` 用于 "or" 过滤。如果需要，可以使用 `\` 转义逗号。

表 1.1. 查询参数表

查询参数	描述
cluster.cluster_id	
cluster.name	
direction	要排序的方向。默认值为 desc ，这代表降序。支持的值有 asc 和 desc 。
event.compliance	
event.message_includes	包含输入字符串的合规性消息的过滤器。仅支持一个值。
event.message_like	用于合规性消息的 SQL LIKE 过滤器。百分比符号 (%) 代表零个或多个字符的通配符。下划线符号 (_) 代表单个字符的通配符。例如， %configmaps [%my-configmap%]% 匹配引用配置映射 my-configmap 的任何配置策略合规性消息。
event.reported_by	
event.timestamp	
event.timestamp_after	RFC 3339 时间戳，以指示此时间后仅指示合规事件。例如， 2424-02-28T16:32:57Z 。

查询参数	描述
event.timestamp_before	RFC 3339 时间戳，在应当显示此时间前仅指示合规事件。例如， 2424-02-28T16:32:57Z 。
id	
include_spec	在返回值中包含策略的 spec 字段的标志。默认不设置。
page	查询中的页面号。默认值为 1 。
parent_policy.categories	
parent_policy.controls	
parent_policy.id	
parent_policy.name	
parent_policy.namespace	
parent_policy.standards	
per_page	每个页面返回的合规事件数量。默认值为 20 ，且不能大于 100 。
policy.apiGroup	
policy.id	
policy.kind	
policy.name	
policy.namespace	
policy.severity	
sort	要排序的字段。默认为 event.timestamp 。 policy.spec 和 event.metadata 以外的所有字段均可使用点表示法排序。要指定多个排序选项，请使用逗号，如 ?sort=policy.name,policy.namespace 。

1.23.2.2. 选择单一策略合规事件

/api/v1/compliance-events/<id>

您可以通过指定数据库 ID 来选择单个策略合规事件。例如：`/api/v1/compliance-events/1` 选择 ID 为 1 的合规性事件。返回值的格式是以下 JSON：

```
{
  "id": 1,
  "cluster": {
    "name": "cluster2",
    "cluster_id": "415ce234-8dee-4cab-b99b-1f8f29dff461"
  },
  "parent_policy": {
    "id": 2,
    "name": "etcd-encryption",
    "namespace": "policies",
    "categories": ["CM Configuration Management"],
    "controls": ["CM-2 Baseline Configuration"],
    "standards": ["NIST SP 800-53"]
  },
  "policy": {
    "apiGroup": "policy.open-cluster-management.io",
    "id": 4,
    "kind": "ConfigurationPolicy",
    "name": "etcd-encryption",
    "namespace": "",
    "spec": {}
  },
  "event": {
    "compliance": "Compliant",
    "message": "configmaps [app-data] found as specified in namespace default",
    "timestamp": "2023-07-19T18:25:41.523Z",
    "metadata": {}
  }
}
```

1.23.2.3. 生成电子表格

/api/v1/reports/compliance-events

您可以生成以逗号分隔的值(CSV)电子表格，用于审计和故障排除。它输出相同的查询参数，并接受与 `/api/v1/compliance-events` API 端点相同的查询参数。默认情况下，没有设置 `per_page` 限制，并且 `per_page` 查询参数没有最大值。所有 CSV 标头与 `/api/v1/compliance-events` API 端点相同，带有下列划线分隔 JSON 对象。例如，事件时间戳的标头为 `event_timestamp`。

1.23.3. 认证和授权

策略合规历史记录 API 使用 Red Hat Advanced Cluster Management hub 集群用于身份验证和授权的 OpenShift 实例。您必须在 HTTPS 请求的 **Authorization** 标头中提供 OpenShift 令牌。

要查找您的令牌，请运行以下命令：

```
oc whoami --show-token
```

1.23.3.1. 查看审计事件

要查看受管集群的合规事件，您需要访问 Red Hat Advanced Cluster Management hub 集群上的 **ManagedCluster** 对象的 **get verb**。例如，要查看 **local-cluster** 集群的合规事件，您可以使用 **open-cluster-management:view:local-cluster ClusterRole** 或创建自己的资源，如下例所示：

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: local-cluster-view
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  resourceNames:
  - local-cluster
  verbs:
  - get
```

要验证您对特定受管集群的访问，请使用 **oc auth can-i** 命令。例如，要检查是否可以访问 **local-cluster** 受管集群，请运行以下命令：

```
oc auth can-i get managedclusters.cluster.open-cluster-management.io/local-cluster
```

1.23.3.2. 记录合规事件

在对应受管集群命名空间中的 **policies.policy.open-cluster-management.io/status** 资源中具有 补丁 动词访问权限的用户或服务帐户，可以访问记录策略合规事件。受管集群上的 **governance-policy-framework** pod 使用 Red Hat Advanced Cluster Management hub 集群上对应的受管集群命名空间中的 **open-cluster-management-compliance-history-api-recorder** 服务帐户来记录合规事件。每个服务帐户都有 **open-cluster-management:compliance-history-api-recorder ClusterRole** 绑定到受管集群命名空间。限制用户和服务帐户对策略状态的访问，以确保存储在策略合规历史记录 API 中数据的可信性。