

# 阿里云 容器服务

开发指南

文档版本：20190321

# 法律声明

---

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

## 通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 <b>禁止：</b> 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 <b>警告：</b> 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 <b>说明：</b> 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	单击 <b>确定</b> 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[ ]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand   slave}</code>

# 目录

---

法律声明.....	I
通用约定.....	I
<b>1 通过 CLI 使用容器服务.....</b>	<b>1</b>
1.1 概述.....	1
1.2 查看所有集群实例.....	2
1.3 查看集群实例.....	2
1.4 创建集群实例.....	3
1.5 扩容集群.....	6
1.6 添加已有实例.....	7
1.7 移除集群内节点.....	8
1.8 查看镜像列表.....	8
1.9 重置节点.....	9
1.10 删除集群实例.....	9
1.11 获取集群证书.....	10
<b>2 Swarm API参考.....</b>	<b>11</b>
2.1 简介.....	11
2.2 API 概述.....	11
2.3 更新历史.....	13
2.4 状态表.....	13
2.5 集群API调用方式.....	14
2.5.1 概述.....	14
2.5.2 请求结构.....	14
2.5.3 公共参数.....	14
2.5.4 返回结果.....	16

# 1 通过 CLI 使用容器服务

## 1.1 概述

阿里云命令行工具是用 Go 语言编写的，基于阿里云 OpenAPI 打造的，用于管理阿里云资源的工具。通过下载和配置该工具，您可以在一个命令行方式下使用多个阿里云产品。

关于阿里云命令行工具的详细介绍以及安装说明，参见 [阿里云 CLI 文档](#)。

容器服务是 RESTful 风格的 API。目前容器服务支持 Swarm 和 Kubernetes 两种调度方式，下面是目前容器服务开放的 API 列表。



说明：

关于阿里云容器服务已支持的 API，参见 [容器服务 API 参考](#)。

API 接口	解释	适用范围
<a href="#">查看所有集群实例</a>	查看您在容器服务中创建的所有集群（包括 Swarm 和 Kubernetes 集群）。	Swarm 集群和 Kubernetes 集群
<a href="#">查看集群实例</a>	根据集群 ID，查看集群的详细信息。	Swarm 集群和 Kubernetes 集群
<a href="#">创建集群实例</a>	创建一个新的集群实例，并新建指定数量的节点。	Swarm 集群和 Kubernetes 集群
<a href="#">扩容集群</a>	增加集群中节点的数量。	Swarm 集群和 Kubernetes 集群
<a href="#">添加已有实例</a>	添加已有实例到集群。	Swarm 集群和 Kubernetes 集群
<a href="#">移除集群内节点</a>	根据集群 ID，以及节点 IP 从容器集群中移除节点。	Swarm 集群
<a href="#">查看镜像列表</a>	查看目前所支持的地域下支持的镜像列表。	Swarm 集群
<a href="#">重置节点</a>	重置集群中的某个节点。	Swarm 集群
<a href="#">删除集群实例</a>	根据集群 ID，删除集群实例，并释放集群所有节点资源。	Swarm 集群和 Kubernetes 集群
<a href="#">获取集群证书</a>	根据集群 ID，获取集群的证书信息。	Swarm 集群

## 1.2 查看所有集群实例

查看您在容器服务中创建的所有集群，包括 Swarm 和 Kubernetes 集群。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

### API请求响应

#### 请求格式

```
aliyun cs GET /clusters
```

#### 响应结果

```
[
  {
    "agent_version": "string",
    "cluster_id": "string",
    "created": "datetime",
    "external_loadbalancer_id": "string",
    "master_url": "string",
    "name": "string",
    "network_mode": "string",
    "region_id": "string",
    "security_group_id": "string",
    "size": "numbers",
    "state": "string",
    "updated": "datetime",
    "vpc_id": "string",
    "vswitch_id": "string"
  }
]
```

## 1.3 查看集群实例

根据集群 ID，查看集群的详细信息。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

### API请求响应

#### 请求格式

```
aliyun cs GET /clusters/<cluster_id>
```

#### 响应结果

```
{
```

```
"agent_version": "string",
"cluster_id": "string",
"created": "datetime",
"external_loadbalancer_id": "string",
"master_url": "string",
"name": "string",
"network_mode": "string",
"region_id": "string",
"security_group_id": "string",
"size": "numbers",
"state": "string",
"updated": "datetime",
"vpc_id": "string",
"vswitch_id": "string"
}
```

## 1.4 创建集群实例

创建一个新的集群实例，并新建指定数量的节点。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

### API请求响应

#### 请求格式

```
aliyun cs POST /clusters --header "Content-Type=application/json" --
body "$(cat create.json)"
```

#### 参数说明：

- `--header` 需要指定 `Content-Type` 为 `application/json`。
- `--body` 是要发送给服务端的 `body` 内容，可以从本地文件读取，需要是有效的 JSON 格式。  
`create.json` 的内容如下所示。

#### swarm 集群

```
{
  "password": "ECS机器SSH密码",
  "region_id": "地域ID",
  "instance_type": "实例规格",
  "name": "集群名称",
  "size": 节点数量,
  "network_mode": "网络类型，目前仅支持vpc",
  "vpc_id": "VPC示例ID",
  "vswitch_id": "VPC下的交换机ID",
  "subnet_cidr": "容器CIDR",
  "data_disk_category": "数据盘类型",
  "data_disk_size": 数据盘大小,
  "need_slb": 是否默认创建SLB,
  "io_optimized": "是否IO优化，VPC下目前默认为IO优化",
  "ecs_image_id": "镜像ID",
  "release_eip_flag": "是否需要在集群配置完成后释放EIP"
```

```
}

```

### Kubernetes集群(单可用区)

```
{
  "disable_rollback": "失败是否回滚",
  "name": "集群名称",
  "timeout_mins": "集群创建超时时间",
  "cluster_type": "Kubernetes",
  "region_id": "地域",
  "vpcid": "VPC ID",
  "zoneid": "可用区",
  "vswitchid": "交换机ID",
  "container_cidr": "容器POD CIDR",
  "service_cidr": "服务CIDR",
  "ssh_flags": "是否开放公网SSH登陆",
  "cloud_monitor_flags": "是否安装云监控插件",
  "login_password": "节点SSH登陆密码, 和key_pair二选一",
  "key_pair": "keypair名称, 和login_password 二选一",
  "master_instance_charge_type": "Master实例付费类型, PostPaid|PrePaid",
  "master_period_unit": "包年包月单位, Month, Year, 只有在PrePaid下生效",
  "master_period": "包年包月时长, 只有在PrePaid下生效",
  "master_auto_renew": "Master节点是否自动续费",
  "master_auto_renew_period": "Master节点续费周期",
  "master_instance_type": "Master实例规格",
  "master_system_disk_category": "Master系统盘类型",
  "master_system_disk_size": "Master节点系统盘大小",
  "master_data_disk": "Master节点是否挂载数据盘",
  "master_data_disk_category": "Master节点数据盘类型",
  "master_data_disk_size": "Master节点数据盘大小",
  "worker_instance_charge_type": "Worker节点付费类型PrePaid|PostPaid",
  "worker_period_unit": "包年包月单位, Month, Year, 只有在PrePaid下生效",
  "worker_period": "包年包月时长, 只有在PrePaid下生效",
  "worker_auto_renew": "Worker节点自动续费true|false",
  "worker_auto_renew_period": "Worker节点续费周期",
  "worker_instance_type": "Worker实例规格",
  "worker_system_disk_category": "Worker系统盘类型",
  "worker_system_disk_size": "Worker节点系统盘大小",
  "worker_data_disk": "Worker节点是否挂载数据盘",
  "worker_data_disk_category": "Worker节点数据盘类型",
  "worker_data_disk_size": "Worker节点数据盘大小",
  "num_of_nodes": "Worker节点数",
  "snat_entry": "是否配置SNATEntry",
  "public_slb": "是否创建公网API Server对应的SLB"
}
```

### Kubernetes集群(多可用区)

```
{
  "disable_rollback": "失败是否回滚",
  "name": "集群名称",
  "timeout_mins": "集群创建超时时间",
  "cluster_type": "Kubernetes",
  "region_id": "地域",
  "multi_az": true,
  "vpcid": "VPC ID",
  "container_cidr": "容器 CIDR",
  "service_cidr": "服务 CIDR",
  "vswitch_id_a": "第一个可用区交换机ID",
  "vswitch_id_b": "第二个可用区交换机ID",
  "vswitch_id_c": "第三个可用区交换机ID",
  "master_instance_type_a": "第一个可用区Master节点实例规格",

```



```

"master_instance_type_b": "第二个可用区Master节点实例规格",
"master_instance_type_c": "第三个可用区Master节点实例规格",
"master_instance_charge_type": "Master实例付费类型, PostPaid|PrePaid",
"master_period_unit": "包年包月单位, Month, Year, 只有在PrePaid下生效",
"master_period": "包年包月时长, 只有在PrePaid下生效",
"master_auto_renew": "Master节点是否自动续费",
"master_auto_renew_period": "Master节点续费周期",
"master_system_disk_category": "Master节点系统盘类型",
"master_system_disk_size": "Master节点系统盘大小",
"master_data_disk": "Master节点是否挂载数据盘",
"master_data_disk_category": "Master节点数据盘类型",
"master_data_disk_size": "Master节点数据盘大小",
"worker_instance_type_a": "第一个可用区Worker节点实例规格",
"worker_instance_type_b": "第二个可用区Worker节点实例规格",
"worker_instance_type_c": "第三个可用区Worker节点实例规格",
"worker_instance_charge_type": "Worker节点付费类型PrePaid|PostPaid",
"worker_period_unit": "包年包月单位, Month, Year, 只有在PrePaid下生效",
"worker_period": "包年包月时长, 只有在PrePaid下生效",
"worker_auto_renew": "Worker节点自动续费true|false",
"worker_auto_renew_period": "Worker节点续费周期",
"worker_system_disk_category": "Worker节点系统盘类型",
"worker_system_disk_size": "Worker节点系统盘大小",
"worker_data_disk": "Worker节点是否挂载数据盘",
"worker_data_disk_category": "Worker节点数据盘类型",
"worker_data_disk_size": "Worker节点数据盘大小",
"num_of_nodes_a": "第一个可用区Worker节点数",
"num_of_nodes_b": "第二个可用区Worker节点数",
"num_of_nodes_c": "第三个可用区Worker节点数",
"ssh_flags": "是否开放公网SSH 登陆",
"login_password": "SSH 登陆密码",
"cloud_monitor_flags": "是否安装云监控插件",
"public_slb": "是否创建公网API Server对应的SLB"
}

```

## 托管Kubernetes集群

```

{
"disable_rollback": "失败是否回滚",
"name": "集群名称",
"timeout_mins": "集群创建超时时间",
"cluster_type": "ManagedKubernetes",
"region_id": "地域, 目前仅支持北京和杭州地域(cn-beijing, cn-hangzhou)",
"vpcid": "VPC ID",
"zoneid": "可用区",
"vswitchid": "交换机ID",
"container_cidr": "容器POD CIDR",
"service_cidr": "服务CIDR",
"cloud_monitor_flags": "是否安装云监控插件",
"login_password": "节点SSH登陆密码, 和key_pair二选一",
"key_pair": "keypair名称, 和login_password 二选一",
"worker_instance_charge_type": "Worker节点付费类型PrePaid|PostPaid",
"worker_period_unit": "包年包月单位, Month, Year, 只有在PrePaid下生效",
"worker_period": "包年包月时长, 只有在PrePaid下生效",
"worker_auto_renew": "Worker节点自动续费true|false",
"worker_auto_renew_period": "Worker节点续费周期",
"worker_instance_type": "Worker实例规格",
"worker_system_disk_category": "Worker系统盘类型",
"worker_system_disk_size": "Worker节点系统盘大小",
"worker_data_disk": "是否挂载数据盘 true|false",
"worker_data_disk_category": "数据盘类型",
"worker_data_disk_size": "数据盘大小",
"num_of_nodes": "Worker节点数",
"snat_entry": "是否配置SNATEntry",

```

```
}ntry": 是否配置SNATEntry,  
}
```

### 响应结果

```
{  
  "cluster_id": "c61cf530524474386a7ab5a1c192a0d57",  
  "request_id": "348D4C9C-9105-4A1B-A86E-B58F0F875575",  
  "task_id": "T-5ad724ab94a2b109e8000004"  
}
```

## 1.5 扩容集群

增加集群中节点的数量。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

### API 请求响应

#### 请求格式

```
aliyun cs PUT /clusters/<cluster_id> --header "Content-Type=  
application/json" --body "$(cat scale.json)"
```

#### 参数说明：

- `--header` 需要指定 Content-Type 为 application/json。
- `--body` 是要发送给服务端的 body 内容，可以从本地文件读取，需要是有效的 JSON 格式。  
`scale.json` 的内容如下所示。

#### Swarm 集群

```
{  
  "password": "ECS 机器 SSH 密码",  
  "instance_type": "实例规格",  
  "size": 节点数量,  
  "data_disk_category": "数据盘类型",  
  "data_disk_size": 数据盘大小,  
  "io_optimized": "是否 IO 优化, VPC 下目前默认为 IO 优化",  
  "ecs_image_id": "镜像 ID",  
  "release_eip_flag": "是否需要在集群配置完成后释放 EIP"  
}
```

#### Kubernetes 集群

```
{ "disable_rollback": "失败是否回滚",  
  "timeout_mins": 集群创建超时时间,  
  
  "worker_instance_type": "Worker实例规格",  
  "login_password": "节点SSH登录密码",  
  "num_of_nodes": "Worker节点数"
```

```
}
```

### 响应结果

```
{
  "cluster_id": "c61cf530524474386a7ab5a1c192a0d57",
  "request_id": "348D4C9C-9105-4A1B-A86E-B58F0F875575",
  "task_id": "T-5ad724ab94a2b109e8000004"
}
```

## 1.6 添加已有实例

添加已有实例到集群。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

### API 请求响应

#### 请求格式

```
aliyun cs POST /clusters/<cluster_id>/attach --header "Content-Type=application/json" --body "$(cat attach.json)"
```

#### 参数说明：

- `--header` 需要指定 Content-Type 为 application/json。
- `--body` 是要发送给服务端的 body 内容，可以从本地文件读取，需要是有效的 JSON 格式。`attach.json` 的内容如下所示。

```
{
  "password": "ECS 机器 SSH 密码",
  "instances": "ECS 示例数组",
  "ecs_image_id": "镜像 ID",
  "release_eip_flag": "是否需要在集群配置完成后释放 EIP"
}
```

### 响应结果

```
{
  "list": [
    {
      "code": "200",
      "instanceId": "i-2zee3oiwcyoz7kwdo8bt",
      "message": "successful"
    },
    {
      "code": "200",
      "instanceId": "i-2ze0lgm3y6iylcbtcypf",
      "message": "successful"
    }
  ],
  "task_id": "T-5a544aff80282e39ea000039"
}
```

```
}
```

## 1.7 移除集群内节点

根据集群 ID，以及节点 IP 从容器集群中移除节点。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群。

### API 请求响应

#### 请求格式

```
aliyun cs DELETE /clusters/<cluster_id>/ip/<ip>?releaseInstance=true
```

#### 参数说明：

- releaseInstance 设置移除节点同时是否释放 ECS。

#### 响应结果

无。

## 1.8 查看镜像列表

查看目前所支持的地域下支持的镜像列表。具体的 API 描述，参见 [容器服务 API 参考](#)。

### API 请求响应

#### 请求格式

```
aliyun cs GET /images
```

#### 响应结果

```
{
  "<RegionID>": {
    "items": [
      {
        "text": "<ImageName>",
        "value": "<ImageID>"
      },
      {
        "text": "<ImageName>",
        "value": "<ImageID>"
      }
    ]
  }
}
```

```
}
```

## 1.9 重置节点

重置集群中的某个节点。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群。

### API请求响应

#### 请求格式

```
aliyun cs POST /clusters/<cluster_id>/instances/<instance_id>/reset  
--header "Content-Type=application/json" --body "$(cat reset.json)"
```

#### 参数说明：

- `--header` 需要指定 Content-Type 为 application/json。
- `--body` 是要发送给服务端的 body 内容，可以从本地文件读取，需要是有效的 JSON 格式。  
`reset.json` 的内容如下所示。

```
{  
  "password": "ECS 机器 SSH 密码",  
  "ecs_image_id": "镜像 ID",  
  "release_eip_flag": "是否需要在集群配置完成后释放 EIP"  
}
```

#### 响应结果

```
{  
  "cluster_id": "c61cf530524474386a7ab5a1c192a0d57",  
  "request_id": "348D4C9C-9105-4A1B-A86E-B58F0F875575",  
  "task_id": "T-5ad724ab94a2b109e8000004"  
}
```

## 1.10 删除集群实例

根据集群 ID，删除集群实例，并释放集群所有节点资源。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群和 Kubernetes 集群。

## API 请求响应

### 请求格式

```
aliyun cs DELETE /clusters/<cluster_id>
```

### 响应结果

无。

## 1.11 获取集群证书

根据集群 ID，获取集群的证书信息。具体的 API 描述，参见 [容器服务 API 参考](#)。

### 适用范围

Swarm 集群。

## API 请求响应

### 请求格式

```
aliyun cs GET /clusters/<cluster_id>/certs
```

### 响应结果

```
{
  "ca": "string",
  "cert": "string",
  "key": "string"
}
```

## 2 Swarm API参考

### 2.1 简介

您可以使用本文档介绍的 API 对容器服务进行相关操作。



说明:

请确保在使用这些接口前，您已充分了解了容器服务说明、使用协议和收费方式。

#### 术语表

术语	中文	说明
Cluster	集群	您的容器集群，一个集群可以部署多个应用。
Node	节点	您的容器集群中的某一个节点，目前只支持 ECS 实例。
Project	应用	一个复杂的应用可以由多个服务组合而成，最简单的应用可能只包含一个容器。
Service	服务	一组基于相同镜像和配置定义的容器，作为一个可伸缩的微服务。
Container	容器	Docker 容器运行时实例。

### 2.2 API 概述

容器服务的 API 主要分为3个部分：

- 集群管理接口
- 应用管理接口
- 触发器

#### 集群管理接口

容器服务提供了一些用于集群管理的接口，例如创建集群、删除集群等。

集群管理接口列表：

API	描述
<a href="#">GetClusterList</a>	查看所有集群实例
<a href="#">CreateCluster</a>	创建集群实例
<a href="#">DeleteCluster</a>	删除集群实例
<a href="#">GetClusterById</a>	查看集群实例
<a href="#">GetClusterCerts</a>	获取集群证书
<a href="#">UpdateClusterSizeById</a>	更新集群节点数量

### 应用管理接口

应用管理提供了 [Docker Remote API](#) 兼容接口。您可以像访问单个 Docker Engine 一样，操作您的 Docker 集群。

应用管理接口列表：

API	描述
<a href="#">List Projects</a>	查看应用实例列表
<a href="#">Create project</a>	创建应用实例
<a href="#">Retrieve project</a>	查看应用实例
<a href="#">Start project</a>	启动应用实例
<a href="#">Stop project</a>	停止应用实例
<a href="#">Kill project</a>	终止应用实例
<a href="#">Update project</a>	更新应用实例
<a href="#">Delete project</a>	删除应用实例
<a href="#">List Services</a>	查看服务实例列表
<a href="#">Retrieve service</a>	查看服务实例
<a href="#">Start service</a>	启动服务实例
<a href="#">Stop service</a>	停止服务实例
<a href="#">Kill service</a>	终止服务实例
<a href="#">Scale service</a>	伸缩服务实例
<a href="#">Create volume</a>	创建数据卷
<a href="#">View volume</a>	查看数据卷



API	描述
<a href="#">List volumes</a>	查看数据卷列表
<a href="#">Delete volume</a>	删除数据卷

## 触发器

触发器是容器服务提供的简单快捷地进行持续部署的 API。更多详细信息参见 [说明](#)。

## 相关文档

- [API 资源导航](#)
- [API Explorer](#)
- [API 错误中心](#)

## 2.3 更新历史

发布时间	更新	说明
2015-12-15	第一版确定	提供了集群管理的基本接口
2016-02-04	增加应用相关接口	提供了应用管理的基本接口

## 2.4 状态表

状态	说明
创建中 (Launching)	集群正在申请相应的云资源
运行中 (Running)	集群正在运行
创建失败 (Failed)	集群申请云资源失败
启动中 (Starting)	集群正在启动
停止中 (Stopping)	集群正在停止运行
停止 (Stopped)	集群停止运行
重启中 (Restarting)	集群正在重启
升级中 (Updating)	集群升级服务端版本
集群规模变更中 (Scaling)	变更集群的节点数量
删除中 (Deleting)	正在删除集群
已删除 (Deleted)	集群删除成功

## 2.5 集群API调用方式

### 2.5.1 概述

对容器服务 API 接口的调用是通过向容器服务 API 的服务端地址发送 HTTP 请求，并按照接口说明在请求中加入相应请求参数来完成的。根据请求的处理情况，系统会返回处理结果。

1. [请求结构](#)
2. [公共参数](#)
3. [返回结果](#)
4. [签名机制](#)

### 2.5.2 请求结构

#### 服务地址

阿里云容器服务的 Open API 接入地址为 `cs.aliyuncs.com`。

#### 通信协议

支持通过 HTTP 或 HTTPS 通道进行请求通信。为了获得更高的安全性，推荐您使用 HTTPS 通道发送请求。

#### 请求方法

使用 HTTP 的 PUT、POST、GET、DELETE 等 HTTP Method 发送不同的请求。

#### 请求参数

每个请求都需要包含公共请求参数和指定操作所特有的请求参数。

#### 请求编码

请求及返回结果都使用 UTF-8 字符集进行编码。

### 2.5.3 公共参数

#### 公共请求头部

公共请求参数是指每个接口都需要使用到的请求参数。

参数名称	说明	选项
Authorization	用于验证请求合法性的认证信息，采用 AccessKeyId: Signature 的形式。	Required

参数名称	说明	选项
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度。	Required
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型。	Required
Content-MD5	HTTP 协议消息体的 128-bit MD5 散列值转换成 BASE64 编码的结果。为了防止所有请求被篡改，建议所有请求都附加该信息。	Required
Date	请求的构造时间，目前只支持 GMT 格式。如果与 MNS 的服务器时间前后差异超过 15 分钟将返回本次请求非法。	Required
Host	访问 Host 值，例如：diku.aliyuncs.com。	Required
Accept	客户端需要的返回值类型，支持 application/json 和 application/xml。	Required
x-acs-version	API 版本号。目前版本号为 2015-12-15。	Required
x-acs-region-id	地域 (Region) 指的是 ECS 实例所在的物理位置。更多详细信息参见 <a href="#">地域概念</a> 和 <a href="#">ECS查询可用地域列表</a> 。	Required
x-acs-signature-nonce	唯一随机数，用于防止网络重放攻击。您在不同请求间要使用不同的随机数值。	Required
x-acs-signature-method	用户签名方式，目前只支持 HMAC-SHA1。	Required

### 示例

```
GET /clusters HTTP/1.1
Host: cs.aliyuncs.com
Accept: application/json
User-Agent: cs-sdk-python/0.0.1 (Darwin/15.2.0/x86_64;2.7.10)
x-acs-signature-nonce: f63659d4-10ac-483b-99da-ea8fde61eae3
Authorization: acs <yourAccessKeyId>:<yourSignature>
x-acs-signature-version: 1.0
Date: Wed, 16 Dec 2015 11:18:47 GMT
x-acs-signature-method: HMAC-SHA1
```

```
Content-Type: application/json;charset=utf-8
X-Acs-Region-Id: cn-beijing
Content-Length: 0
```

### 公共返回头部

您发送的每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码 RequestId。

### 示例

#### XML 示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!--结果的根结点-->
<接口名称+Response>
| <!--返回请求标签-->
| <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
| <!--返回结果数据-->
</接口名称+Response>
```

#### JSON 示例：

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216"
  /* 返回结果数据 */
}
```

## 2.5.4 返回结果

调用 API 服务后返回数据采用统一格式。返回的 HTTP 状态码为 2xx，代表调用成功；返回的 HTTP 状态码为 4xx 或 5xx，代表调用失败。调用成功返回的数据格式主要有 XML 和 JSON 两种，外部系统可以在请求时传入参数来制定返回的数据格式，默认为 XML 格式。

为了便于您查看，本文档中的返回示例做了格式化处理，实际返回结果是没有换行、缩进等处理的。