

Alibaba Cloud Container Service

Developer Guide

Issue: 20190414

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.








1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>swich {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Use Container Service by using CLI.....	1
1.1 Overview.....	1
1.2 View all clusters.....	2
1.3 View a cluster.....	3
1.4 Create a cluster.....	3
1.5 Expand a cluster.....	8
1.6 Add existing ECS instances to a cluster.....	9
1.7 Remove a node from a cluster.....	10
1.8 View image list.....	11
1.9 Delete a cluster.....	11
1.10 Obtain cluster certificate.....	12
2 Swarm API reference.....	13
2.1 Introduction.....	13
2.2 API overview.....	13
2.3 Update history.....	15
2.4 Status table.....	15
2.5 Cluster API call mode.....	16
2.5.1 Overview.....	16
2.5.2 Request Structure.....	16
2.5.3 Public parameters.....	17
2.5.4 Returned results.....	19
2.5.5 Signature mechanism.....	20
2.6 Triggers.....	25
2.6.1 Triggers.....	25

1 Use Container Service by using CLI

1.1 Overview

Alibaba Cloud Command Line Interface (CLI) is a tool to manage Alibaba Cloud resources. It is written in Go and built based on Alibaba Cloud APIs. After downloading and configuring this tool, you can use multiple Alibaba Cloud products by using command lines.

For more information about the introduction and installation of the Alibaba Cloud CLI, see [Alibaba Cloud Command Line Interface](#).

Container Service APIs are RESTful APIs. Currently, Container Service supports two scheduling modes: swarm and Kubernetes. The list of APIs that Container Service supports is as follows.



Note:

For more information about the APIs that Alibaba Cloud Container Service supports, see [Container Service API reference](#).

API	Description	Scope
View all clusters	View all the clusters you have created in Container Service, including swarm clusters and Kubernetes clusters.	Swarm clusters and Kubernetes clusters.
View a cluster	View the cluster details according to the cluster ID.	Swarm clusters and Kubernetes clusters.
Create a cluster	Create a cluster with a specified number of new nodes.	Swarm clusters.
Expand a cluster	Increase the number of nodes in the cluster.	Swarm clusters.
Add existing ECS instances to a cluster	Add existing Elastic Compute Service (ECS) instances to a cluster.	Swarm clusters and Kubernetes clusters.

API	Description	Scope
Remove a node from a cluster	Remove a node from a cluster according to the cluster ID and node IP address.	Swarm clusters.
View image list	View the list of supported images in the currently supported regions.	Swarm clusters.
Reset a node	Reset a node in the cluster.	Swarm clusters.
Delete a cluster	Delete a cluster according to the cluster ID and release all node resources of the cluster.	Swarm clusters and Kubernetes clusters.
Obtain cluster certificate	Obtain the cluster certificate according to the cluster ID.	Swarm clusters.

1.2 View all clusters

View all the clusters you have created in Container Service, including swarm clusters and Kubernetes clusters. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

```
aliyun cs GET / clusters
```

Response results

```
" 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 View a cluster

View the cluster details according to the cluster ID. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

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

Response results

```
{
  " Agent_version ": " string ",
  " cluster_id ": " string ",
  " created ": " datetime ",
  " external_loadbalance_r_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 Create a cluster

Create a cluster with a specified number of new nodes. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

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

Parameter descriptions:

- `-- header` Specify Content - Type as applicatio n / json .
- `-- body` is the body to be sent to the server, which can be read from a local file and must be in the valid JSON format. The `create . json` contents are as follows:

Swarm clusters

```
{
  " password ": " password used to log on to the Elastic
  Compute Service ( ECS ) instance by using SSH ",
  " region_id ": " region ID ",
  " instance_t ype ": " ECS instance type ",
  " name ": " cluster name ",
  " size ": " number of nodes ",
  " network_mo de ": " network type , currently only supports
  Virtual Private Cloud ( VPC )",
  " vpc_id ": " VPC_ID ",
  " vswitch_id ": " VSwitch ID of the VPC instance ",
  " subnet_cid r ": " container Classless Inter - Domain Routing
  ( CIDR ) block ",
  " data_disk_ category ": " data disk type ",
  " data_disk_ size ": " data disk size ",
  " need_slb ": " whether or not to create the Server Load
  Balancer instance by default ",
  " io_optimiz ed ": " whether or not to be I / O optimized
  , which is yes by default in the VPC environmen t
  ",
  " ecs_image_ id ": " image ID ",
  " release_ei p_flag ": " whether to release Elastic IP ( EIP
  ) after configurin g the cluster "
}
```

Kubernetes clusters (a single zone)

```
{
  " disable_ro llback ": " whether or not to roll back if
  the cluster fails to be created ",
  " name ": " cluster name ",
  " timeout_mi ns ": " timeout for creating the cluster ",
  " cluster_ty pe ": " Kubernetes ",
  " region_id ": " region ",
  " vpcid ": " Virtual Private Cloud ( VPC ) ID ",
  " zoneid ": " zone ",
  " vswitchid ": " VSwitch ID ",
  " container_ cidr ": " pod Classless Inter - Domain Routing (
  CIDR )",
  " service_ci dr ": " service CIDR ",
```

```

" ssh_flags ": " whether or not to enable SSH access for
  Internet ",
" cloud_moni tor_flags ":" whether or not to install the
  cloud monitoring plug - in ",
" login_pass word ": " password used to log on to the
  node by using SSH . Use either this parameter or the
  key_pair .",
" key_pair ":" key pair name . Use either this parameter
  or login_pass word .",
" master_ins tance_char ge_type ":" master instance payment
  type , which includes postpaid and prepaid ",
" master_per iod_unit ":" subscripti on unit , which includes
  month and year , and takes effect only for the
  prepaid type ",
" master_per iod ":" subscripti on period , which takes
  effect only for the prepaid type ",
" master_aut o_renew ":" whether master nodes auto renew ",
" master_aut o_renew_pe riod ":" master node renew period ",
" master_ins tance_type ": " master instance type ",
" master_sys tem_disk_c ategory ": " system disk type of
  master nodes ",
" master_sys tem_disk_s ize ":" system disk size of master
  nodes ",
" master_dat a_disk ":" Whether the master node has data
  disks mounted ",
" master_dat a_disk_cat egory ":" data disk type of master
  nodes ",
" master_dat a_disk_siz e ":" data disk size of master
  nodes ",
" worker_ins tance_char ge_type ":" worker node payment type
  , which includes prepaid and postpaid ",
" worker_per iod_unit ":" subscripti on unit , which includes
  month and year , and takes effect only for the
  prepaid type .
" worker_per iod ":" subscripti on period , which takes
  effect only for the prepaid type ",
" worker_aut o_renew ":" worker nodes auto renew , which can
  be true or false ",
" worker_aut o_renew_pe riod ":" worker node renew period ",
" worker_ins tance_type ": " worker instance type ",
" worker_sys tem_disk_c ategory ": " system disk type of
  worker nodes ",
" worker_sys tem_disk_s ize ":" system disk size of worker
  node ",
" worker_dat a_disk ":" whether the worker node has data
  disks mounted ",
" worker_dat a_disk_cat egory ":" data disk type of worker
  nodes ",
" worker_dat a_disk_siz e ":" data disk size of worker
  nodes ",
" num_of_nod es ":" number of worker nodes ",
" snat_entry ": " whether or not to configure the
  SNATEntry ",
" public_slb ":" whether to create the SLB correspond ing
  to the public network API server "
}

```

Kubernetes clusters (multiple zones)

```

{
" disable_rol lback ": " whether or not to roll back if
  the cluster fails to be created ",
" name ": " cluster name ",

```

```

" timeout_mins ": " timeout for creating the cluster ",
" cluster_type ": " Kubernetes ",
" region_id ": " region ",
" multi_az ": true,
" vpcid ": " VPC ID ",
" container_cidr ": " container Classless Inter - Domain
Routing ( CIDR )",
" service_cidr ": " service CIDR ",
" vswitch_id_a ": " switch ID of the first zone ",
" vswitch_id_b ": " switch ID of the second zone ",
" vswitch_id_c ": " switch ID of the third zone ",
" master_instance_type_a ": " instance type of master
nodes in the first zone ",
" master_instance_type_b ": " instance type of master
nodes in the second zone ",
" master_instance_type_c ": " instance type of master
nodes in the third zone ",
" master_instance_charge_type ": " master instance payment
type , which includes prepaid and postpaid ",
" master_period_unit ": " subscripti on unit , which includes
month and year , and takes effect only for the
prepaid type ",
" master_period ": " subscripti on period , which takes
effect only for the prepaid type ",
" master_auto_renew ": " whether master nodes auto renew ",
" master_auto_renew_period ": " master node renew period ",
" master_system_disk_category ": " system disk type of
master nodes ",
" master_system_disk_size ": " system disk size of master
nodes ",
" master_data_disk ": " Whether the master node has data
disks mounted ",
" master_data_disk_category ": " data disk type of master
nodes ",
" master_data_disk_size ": " data disk size of master
nodes ",
" worker_instance_type_a ": " instance type of worker
nodes in the first zone ",
" worker_instance_type_b ": " instance type of worker
nodes in the second zone ",
" worker_instance_type_c ": " instance type of worker
nodes in the third zone ",
" worker_instance_charge_type ": " worker node payment type
, which includes postpaid and prepaid ",
" worker_period_unit ": " subscripti on unit , which includes
month and year , and takes effect only for the
prepaid type .
" worker_period ": " subscripti on period , which takes
effect only for the prepaid type ",
" worker_auto_renew ": " worker nodes auto renew , which can
be true or false ",
" worker_auto_renew_period ": " worker node renew period ",
" worker_system_disk_category ": " system disk type of
worker nodes ",
" worker_system_disk_size ": " system disk size of worker
nodes ",
" worker_data_disk ": " whether the worker node has data
disks mounted ",
" worker_data_disk_category ": " data disk type of worker
nodes ",
" worker_data_disk_size ": " data disk size of worker
nodes ",
" num_of_nodes_a ": " number of worker nodes in the
first zone ",

```

```

" num_of_nod es_b ": " number of worker nodes in the
second zone ",
" num_of_nod es_c ": " number of worker nodes in the
third zone ",
" ssh_flags ": " whether or not to enable SSH access for
Internet ",
" login_pass word ": " password used to log on to the
node by using SSH ",
" cloud_moni tor_flags ": " whether or not to install the
cloud monitoring plug - in ",
" public_slb ": " whether to create the SLB correspond ing
to the public network API server "
}

```

Managed Kubernetes clusters

```

{
" disable_rol lback ": " whether or not to roll back if
the cluster fails to be created ",
" name ": " cluster name ",
" timeout_mi ns ": " timeout for creating the cluster ",
" cluster_ty pe ": " ManagedKub ernetes ",
" region_id ": " region . Currently , only cn - beijing and cn
- hangzhou are supported ",
" vpcid ": " VPC ID ",
" zoneid ": " zone ",
" vswitchid ": " VSwitch ID ",
" container_ cidr ": " pod Classless Inter - Domain Routing (
CIDR )",
" service_ci dr ": " service CIDR ",
" cloud_moni tor_flags ": " whether or not to install the
cloud monitoring plug - in ",
" login_pass word ": " password used to log on to the
node by using SSH . Use either this parameter or the
key_pair .",
" key_pair ": " key pair name . Use either this parameter
or login_pass word .",
" worker_ins tance_char ge_type ": " worker node payment type
, which includes prepaid and postpaid ",
" worker_per iod_unit ": " subscripti on unit , which includes
month and year , and takes effect only for the
prepaid type .
" worker_per iod ": " subscripti on period , which takes
effect only for the prepaid type ",
" worker_aut o_renew ": " worker nodes auto renew , which can
be true or false ",
" worker_aut o_renew_pe riod ": " worker node renew period ",
" worker_ins tance_ty pe ": " worker instance type ",
" worker_sys tem_disk_c ategory ": " system disk type of
worker nodes ",
" worker_sys tem_disk_s ize ": " system disk size of worker
nodes ",
" worker_dat a_disk ": " use true or false to determine
whether or not to mount data disks ",
" worker_dat a_disk_cat egory ": " data disk type ",
" worker_dat a_disk_siz e ": " data disk size ",
" num_of_nod es ": " number of worker nodes ",
" snat_entry ": " whether or not to configure the
SNATEntry ",
} ntry ": whether or not to configure the SNATEntry ,

```

```
}

```

Response results

```
{
  " cluster_id ": " c61cf53052 4474386a7a b5a1c192a0 d57 ",
  " request_id ": " 348D4C9C - 9105 - 4A1B - A86E - B58F0F8755 75
",
  " task_id ": " T - 5ad724ab94 a2b109e800 0004 "
}
```

1.5 Expand a cluster

Increase the number of nodes in the cluster. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

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

Parameter descriptions:

- `-- header` Specify Content - Type as `applicatio n / json`.
- `-- body` is the body to be sent to the server, which can be read from a local file and must be in the valid JSON format. The `scale . json` contents are as follows:

Swarm clusters

```
{
  " password ": " password used to log on to the
Elastic Compute Service ( ECS ) instance by using SSH ",
  " instance_t ype ": " ECS instance type ",
  " size ": " number of nodes after expansion ",
  " data_disk_ category ": " disk category ",
  " data_disk_ size ": " disk size ",
  " io_optimiz ed ": " whether or not to be I / O
optimized , which is yes by default in the VPC
environmen t ",
  " ecs_image_ id ": " image ID ",
  " release_ei p_flag ": " whether or not to release
Elastic IP ( EIP ) after configurin g the cluster "
}
```



```
}

```

Kubernetes clusters

```
{ "disable_rollback": "whether or not to roll back if
  the cluster fails to be scaled out or in",
  "timeout_mins": "timeout for creating the cluster",
  "worker_instance_type": "instance type of worker nodes",
  "login_password": "password used to log on to the
  node by using SSH",
  "num_of_nodes": "number of worker nodes"
}
```

Response results

```
{
  "cluster_id": "c61cf53052 4474386a7a b5a1c192a0 d57",
  "request_id": "348D4C9C - 9105 - 4A1B - A86E - B58F0F8755 75",
  "task_id": "T - 5ad724ab94 a2b109e800 0004"
}
```

1.6 Add existing ECS instances to a cluster

Add existing Elastic Compute Service (ECS) instances to a cluster. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

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

Parameter description:

- Specify `Content-Type` as `application/json` in `--header`.
- `--body` is the body to be sent to the server, which can be read from a local file and must be in the valid JSON format. The `attach.json` contents are as follows:

```
{ "password": "password used to log on to the
  ECS instance by using SSH",
  "instances": "the ECS instance array to be
  added",
}
```

```
" ecs_image_id ": " image ID ",
" release_instance_flag ": " whether or not to release
Elastic IP ( EIP ) after configuring the cluster "
```

Response results

```
" list ": [
  " code ": " 200 ",
  " instanceId ": " i - 2zee3oiwcy oz7kwdo8bt ",
  " message ": " successful "

  " code ": " 200 ",
  " instanceId ": " i - 2ze0lgm3y6 iylcbtcypf ",
  " message ": " successful "

" task_id ": " T - 5a544aff80 282e39ea00 0039 "
```

1.7 Remove a node from a cluster

Remove a node from a cluster according to the cluster ID and node IP address. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters.

API request and response

Request format

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

Parameter description:

- `releaseInstance` : Whether or not to release the Elastic Compute Service (ECS) instance when removing the node.

Response results

None.

1.8 View image list

View the list of supported images in the currently supported regions. For more information about the API description, see [Container Service API reference](#).

API request and response

Request format

```
aliyun  cs  GET  / images
```

Response results

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

1.9 Delete a cluster

Delete a cluster according to the cluster ID and release all node resources of the cluster. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters and Kubernetes clusters.

API request and response

Request format

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

Response results

None.

1.10 Obtain cluster certificate

Obtain the cluster certificate according to the cluster ID. For more information about the API description, see [Container Service API reference](#).

Scope

Swarm clusters.

API request and response

Request format

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

Response results

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

2 Swarm API reference

2.1 Introduction

You can use the APIs introduced in this document to perform relevant operations on Container Service.



Note:

Before using these interfaces, make sure that you fully understand the instructions, use agreement, and billing method of Container Service.

Terms

Term	Chinese	Description
Cluster	Cluster	Your container cluster. Multiple applications can be deployed in one cluster.
Node	Node	One node in your container cluster. Currently, only Elastic Compute Service (ECS) instances are supported.
Project	Application	A complex application is composed of multiple services. A simplest application might contain only one container.
Service	Service	A group of containers based on identical images and configurations function as a scalable microservice.
Container	Container	A runtime instance of Docker container.

2.2 API overview

Container Service APIs are mainly divided into three parts:

- Cluster interfaces
- Application interfaces
- Triggers

Cluster interfaces

Container Service provides some interfaces to manage clusters, for example, the interfaces used to create or delete clusters.

The list of cluster interfaces is as follows.

API	Description
GetClusterList	View all clusters.
CreateCluster	Create clusters.
DeleteCluster	Delete clusters.
GetClusterById	View clusters.
GetClusterCerts	Obtain the cluster certificate.
UpdateClusterSizeById	Update the number of cluster nodes.

Application interfaces

The application interfaces are compatible with the [Docker Remote API](#). You can operate your Docker cluster in the same way as operating a single Docker Engine.

The list of application interfaces is as follows.

API	Description
List projects	View the application list.
Create project	Create applications.
Retrieve project	View applications.
Start project	Start applications.
Stop project	Stop applications.
Kill project	Terminate applications.
Update project	Update applications.
Delete project	Delete applications.
List services	View the service list.

API	Description
Retrieve service	View services.
Start service	Start services.
Stop service	Stop services.
Kill service	Terminate services.
Scale service	Scale services.
Create volume	Create data volumes.
View volume	View data volumes.
List volumes	View the data volume list.
Delete volume	Delete data volumes.

Triggers

A trigger is an API provided by Container Service for simple, fast, and continuous deployment. For more information, see [Triggers](#).

2.3 Update history

Release date	Update	Description
2015-12-15	Determined the first version.	The basic cluster management interfaces were provided.
2016-02-04	Added application-related interfaces.	The basic application management interfaces were provided.

2.4 Status table

Status	Description
Launching	The cluster is applying for corresponding cloud resources.
Running	The cluster is running.
Failed	The cluster fails to apply for cloud resources.
Starting	The cluster is starting.

Status	Description
Stopping	The cluster is being stopped.
Stopped	The cluster is stopped.
Restarting	The cluster is restarting.
Updating	The cluster is upgrading the server.
Scaling	Change the number of cluster nodes.
Deleting	The cluster is being deleted.
Deleted	The cluster is successfully deleted.

2.5 Cluster API call mode

2.5.1 Overview

The call to Container Service API interfaces is performed by sending HTTP requests to the server address of the Container Service APIs and adding corresponding request parameters to the requests according to the interface instructions. The system returns the results according to the processing results of the requests.

1. [Request structure](#)
2. [Public parameters](#)
3. [Returned results](#)
4. [Signature mechanism](#)

2.5.2 Request Structure

Endpoint

The access address of Alibaba Cloud Container Service API is cs.aliyuncs.com.

Communication protocol

The system supports request communication by using the HTTP or HTTPS channel. We recommend that you use the HTTPS channel to send requests for more security.

Request methods

Use HTTP methods such as PUT, POST, GET, and DELETE to send different requests.

Request Parameters

Each request must contain the public request parameters and the request parameters unique to specified operations.

Request encoding

Both requests and returned results are encoded by using the UTF-8 character set.

2.5.3 Public parameters

Public request headers

Public request parameters are the request parameters that each interface must use.

Parameter name	Description	Option
Authorization	The authentication information used to verify the validity of a request. Format: <code>AccessKeyId : Signature</code> .	Required
Content - Length	The content length of an HTTP request, which is defined in RFC 2616.	Required
Content - Type	The content type of an HTTP request, which is defined in RFC 2616.	Required
Content - MD5	The Base64-encoded results converted from 128-bit MD5. We recommend that you add this message to all requests to prevent requests from being tampered. hash value of the HTTP message body. We recommend that you add this message to all requests to prevent requests from being tampered.	Required

Parameter name	Description	Option
Date	The construction time of a request. Currently , only the GMT format is supported. If the difference between the construction time and the MNS server time exceeds 15 minutes, invalid request is returned.	Required
Host	The host access value, for example, diku.aliyuncs.com.	Required
Accept	The return type required by the client. application / json and application / xml are supported.	Required
x - acs - version	The API version. The current version is 2015-12-15.	Required
x - acs - region - id	A region indicates the physical location of an Elastic Compute Service (ECS) instance. an Elastic Compute Service (ECS) instance.	Required
x - acs - signature - nonce	The unique random number used to prevent network replay attacks. Different random numbers must be used for different requests.	Required
x - acs - signature - method	The method of user signature. Currently only HMAC-SHA1 is supported.	Required

Example

```
GET / clusters HTTP / 1 . 1
Host : cs . aliyuncs . com
Accept : applicatio n / 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 -
ea8fde61ea e3
Authorization : acs < yourAccess KeyId >:< 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
```

Public response headers

Each time you send a request to call an interface, the system returns a unique identifier (RequestId), no matter the request is successful or not.

Example

XML example:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 " ? >
<!-- Result root node -->
< Interface name + Response >
| <!-- Return request tag -->
| < RequestId > 4C467B38 - 3910 - 447D - 87BC - AC049166F2 16 </
RequestId >
| <!-- Return result data -->
</ Interface name + Response >
```

JSON example:

```
" RequestId " : " 4C467B38 - 3910 - 447D - 87BC - AC049166F2 16 "
/* Return result data */
```

2.5.4 Returned results

After the API service is called, data is returned in a unified format. The returned HTTP status code `2xx` indicates that the call is successful. The returned HTTP status code `4xx` or `5xx` indicates that the call fails. When the call is successful, data can be returned mainly in two formats: XML and JSON. When a request is sent, an external system can pass in a parameter to define the format of the returned data. The default format is XML.

Examples of returned results in this document are formatted for ease of viewing. The actual results returned are not formatted with line breaks or indentation.

2.5.5 Signature mechanism

Introduction

The Access Key ID and Access Key Secret are officially issued to you by Alibaba Cloud (you can apply for and manage them on the Alibaba Cloud official website). The Access Key ID is used to identify your identity. The Access Key Secret is the key used to encrypt the signature string and verify the signature string on the server side. You must keep the Access Key Secret confidential. Only you and Alibaba Cloud can know it.

Container Service verifies each access request it receives. Therefore, all requests sent to Container Service must contain signature information. Container Service performs symmetric encryption by using the Access Key ID and Access Key Secret to verify the identity of request senders. If the calculated verification code is the same as the one provided, the request is considered as valid. Otherwise, Container Service rejects the request and returns the HTTP 403 error.

You can add the authorization header in the HTTP request to contain the signature information, indicating that the message has been authorized.

Container Service requires to contain the signature in the HTTP header in the format of `Authorization : acs [Access Key ID]:[Signature]`.

The `Signature` calculation method is as follows:

```
Signature = base64 ( hmac - sha1 ( VERB + "\ n "
+ ACCEPT + "\ n " +
+ Content - MD5 + "\ n "
+ Content - Type + "\ n "
+ Date + "\ n "
+ Canonicali zedHeaders + "\ n "
+ Canonicali zedResourc e ) )
```

- `VERB` indicates the HTTP method, for example, `PUT`.
- `Accept` indicates the return type required by the client, which can be `application/json` or `application/xml`.
- `Content - MD5` indicates the MD5 value of the requested content.
- `Content - Type` indicates the type of the requested content.
- `Date` indicates the operation time, which cannot be null. Currently, only the GMT format is supported. If the difference between the request time and the CAS server time exceeds 15 minutes, CAS considers the request as invalid and returns error

400. For more information, see the 5th section. For example, `Thu , 17 Mar 2012 18 : 49 : 58 GMT .`

- `CanonicalizedHeaders` indicates a combination of fields started with `x - acs -` in the HTTP request.
- `CanonicalizedResource` indicates the uniform resource identifier (URI) of the resource in the HTTP request. For example, `/ clusters ? name = my - clusters & resource = new .`



Note:

Conform to the following specifications for `CanonicalizedHeaders` (headers started with `x - acs -`) before signature verification:

1. Convert the names of all HTTP request headers started with `x - acs -` to lowercase letters. For example, convert `X - ACS - Meta - Name : TaoBao` to `x - acs - meta - name : TaoBao` . The names of request headers are case-insensitive according to Alibaba Cloud specifications. However, we recommend that you use the lowercase letters.
2. If the value part of a public request header is too long, replace the `\ t` , `\ n` , `\ r` , and `\ f` separators with spaces.
3. Sort all HTTP request headers that are obtained from the preceding step and compliant with Alibaba Cloud specifications in the lexicographic ascending order.
4. Delete any space at either side of a separator between request header and content. For example, convert `x - acs - meta - name : TaoBao , Alipay` to `x - acs - meta - name : TaoBao , Alipay` .
5. Separate all headers and contents with the `\ n` separator to form the final `CanonicalizedHeaders` .



Note:

The format specification for `CanonicalizedResource : CanonicalizedResource` indicates the standard description of the resource you want to access. Sort sub-resources and query in the lexicographically ascending order

and separate them by using the & separator to generate a sub-resource string (all parameters after ?).

```
http://cs.aliyuncs.com/clusters?name=my-clusters&resource=new
```

The CanonicalizedResource format is:

```
/clusters?name=my-clusters&resource=new
```

Signature example

Overview

The following example shows the signature process.

In the example, the Access Key ID and Access Key Secret are `access_key_id` and `access_key_secret` respectively. We recommend that you use your own API call program to calculate the signature string in the following example. Then, compare your signature string with the example result.

The request example is as follows:

```
POST http://cs.aliyuncs.com/clusters?param1=value1&param2=value2 HTTP/1.1
Accept-Encoding: identity
Content-Length: 210
Content-MD5: 6U4ALMkKSj0PYbeQSHqg mA ==
x-acs-version: 2015-12-15
Accept: application/json
User-Agent: cs-sdk-python/0.0.1 (Darwin/15.2.0/x86_64;2.7.10)
x-acs-signature-nonce: fbf6909a-93a5-45d3-8b1c-3e03a7916799
x-acs-signature-version: 1.0
Date: Wed, 16 Dec 2015 12:20:18 GMT
x-acs-signature-method: HMAC-SHA1
Content-Type: application/json; charset=utf-8
X-Acs-Region-Id: cn-beijing
Authorization: acs_access_key_id:/ZmVlMDNkND A1ZTQyMWVi YWY1MTRhZG VjODgxMDM4 YzRiMzEzNT g0ZA ==
{"password": "Just $ test", "instance_type": "ecs.m2.medium", "name": "my-test-cluster-97082734", "size": 1, "network_mode": "classic", "data_disk_category": "cloud", "data_disk_size": 10, "ecs_image_id": "m-253llee3l"}
```

Request construction process

Calculate Content-Length and Content-MD5

Content-Length: The length of the body content.

**Note:**

No space or line break is at the beginning of the example body.

```
body : {" password " : " Just $ test "," instance_t ype " : " ecs . m2
. medium "," name " : " my - test - cluster - 97082734 "," size " : 1
," network_mo de " : " classic "," data_disk_ category " : " cloud
"," data_disk_ size " : 10 ," ecs_image_ id " : " m - 253llee3l "}
Content - Length : 210
```

Content - MD5 : The MD5 calculation process.

```
body : {" password " : " Just $ test "," instance_t ype " : " ecs . m2
. medium "," name " : " my - test - cluster - 97082734 "," size " : 1
," network_mo de " : " classic "," data_disk_ category " : " cloud
"," data_disk_ size " : 10 ," ecs_image_ id " : " m - 253llee3l "}
# Calculate the MD5 value of the body .
md5 ( body ) : e94e002cc9 0a4a3d0f61 b790487aa0 98
# Convert the MD5 value to a byte array . Convert
every two hexadecimal symbols of the MD5 value to
a byte .
# For example , e9 -> 11111111 11111111 111111010 01 ->
- 23
bytes ( md5 ( body )) : { [- 23 ], [ 78 ], [ 0 ], [ 44 ], [- 55 ], [ 10
], [ 74 ], [ 61 ], [ 15 ], [ 97 ], [- 73 ], [- 112 ], [ 72 ], [ 122
], [- 96 ], [- 104 ]}
# Convert the obtained byte array to a Base64 string
.
base64 ( bytes ( md5 ( body )) ) : 6U4ALMkKSj 0PYbeQSHqg mA ==
Content - MD5 : 6U4ALMkKSj 0PYbeQSHqg mA ==
```

Process CanonicalizedHeaders

```
# List all headers started with ' x - acs - '.
x - acs - version : 2015 - 12 - 15
x - acs - signature - nonce : ca480402 - 7689 - 43ba - acc4 -
4d2013d9d8 d4
x - acs - signature - version : 1 . 0
x - acs - signature - method : HMAC - SHA1
X - Acs - Region - Id : cn - beijing
# Convert the request name to lowercase letters , delete
the spaces at the beginning and end of each line
, and sort the headers in the lexicographically
ascending order . Delete any space at either side of
a separator between request header and content .
# Note : No line break is in the last line .
x - acs - region - id : cn - beijing
x - acs - signature - method : HMAC - SHA1
x - acs - signature - nonce : fbf6909a - 93a5 - 45d3 - 8b1c -
3e03a79167 99
x - acs - signature - version : 1 . 0
x - acs - version : 2015 - 12 - 15
```

Calculate CanonicalizedResource

In the example, the length of CanonicalizedResource is 27.

**Note:**

An `\n` line break is at the end of the first line.

```
/ clusters ? param1 = value1 & param2 = value2
```

Calculate Signature

Assemble `SignatureS` `tring`. In the example, the length of the signature string is 307. An `\n` line break is at the end of all lines except the last line.

```
POST
applicatio n / json
6U4ALMkKSj 0PYbeQSHqg mA ==
applicatio n / json ; charset = utf - 8
Wed , 16 Dec 2015 12 : 20 : 18 GMT
x - acs - region - id : cn - beijing
x - acs - signature - method : HMAC - SHA1
x - acs - signature - nonce : fbf6909a - 93a5 - 45d3 - 8b1c -
3e03a79167 99
x - acs - signature - version : 1 . 0
x - acs - version : 2015 - 12 - 15
/ clusters ? param1 = value1 & param2 = value2
```

Calculate Signature

```
# Use Access Key Secret to encrypt the signature
string . In the example , the accessKeyS ecret is
access_key _secret .
hmac - sha1 ( SignatureS tring ) : fee03d405e 421ebaf514
adec881038 c4b313584d
# Convert the encrypted string to a byte array ,
similar to the Content - MD5 calculatio n method .
# Convert the byte array into a Base64 string to
get the final signature string .
base64 ( bytes ( hmac - sha1 ( SignatureS tring ) ) ) : ZmVlMDNkND
A1ZTQyMWVvi YWY1MTRhZG VjODgxMDM4 YzRiMzEzNT g0ZA ==
Signature : ZmVlMDNkND A1ZTQyMWVvi YWY1MTRhZG VjODgxMDM4
YzRiMzEzNT g0ZA ==
```

Finish

After the preceding processing, add some other header information to construct the final HTTP request as follows:

```
POST http :// cs . aliyuncs . com / clusters ? param1 = value1 &
param2 = value2 HTTP / 1 . 1
Accept - Encoding : identity
Content - Length : 210
Content - MD5 : 6U4ALMkKSj 0PYbeQSHqg mA ==
x - acs - version : 2015 - 12 - 15
Accept : applicatio n / json
User - Agent : cs - sdk - python / 0 . 0 . 1 ( Darwin / 15 . 2 . 0
/ x86_64 ; 2 . 7 . 10 )
x - acs - signature - nonce : fbf6909a - 93a5 - 45d3 - 8b1c -
3e03a79167 99
```



```
x - acs - signature - version : 1 . 0
Date : Wed , 16 Dec 2015 12 : 20 : 18 GMT
x - acs - signature - method : HMAC - SHA1
Content - Type : applicatio n / json ; charset = utf - 8
X - Acs - Region - Id : cn - beijing
Authorizat ion : acs access_key _id : / ZmVlMDNkND A1ZTQyMWVl
YWY1MTRhZG VjODgxMDM4 YzRiMzEzNT g0ZA ==
{" password " : " Just $ test " , " instance_t ype " : " ecs . m2 .
medium " , " name " : " my - test - cluster - 97082734 " , " size " : 1 , "
network_mo de " : " classic " , " data_disk_ category " : " cloud " , "
data_disk_ size " : 10 , " ecs_image_ id " : " m - 253llee3l " }
```

2.6 Triggers

2.6.1 Triggers

Introduction

A trigger is an API provided by Container Service for simple and fast redeployment and resource scaling.

The strict authentication is needed because standard APIs must guarantee the security. However, in scenarios where an API is integrated with a third-party system (for example, Jenkins or other continuous integration CI/CD system), the required permissions are limited, for example, messaging only. Therefore, to guarantee the security and convenience, APIs that have partial authentication policies and can be flexibly called are widely applied in scenarios requiring continuous integration and delivery.

Currently, Container Service provides redeployment trigger and resource scaling trigger.

- **Redeployment trigger**

You can integrate your APIs with your monitoring system and redeploy your applications when the system has exceptions. You can also integrate your APIs with container Hub, and then the application can be automatically redeployed by using the latest image after the new image is constructed.

- **Resource scaling trigger**

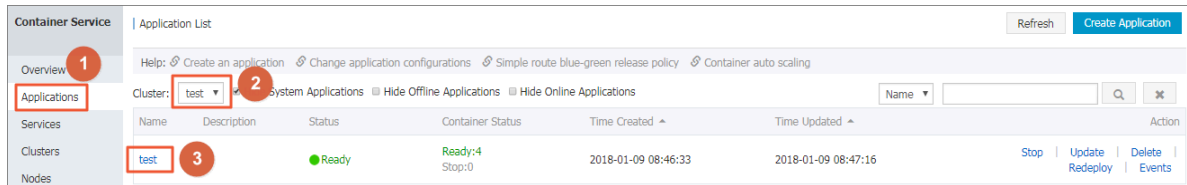
You can call the resource scaling trigger to realize container scaling.

Create a trigger

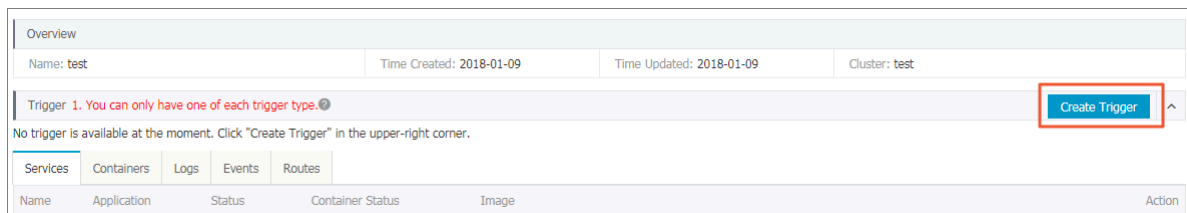
Procedure

1. Log on to the [Container Service console](#).

2. Click Applications in the left-side navigation pane.
3. Select the cluster in which the application resides from the Cluster list.
4. Click the application name. name.



5. Click Create Trigger in the upper right corner.

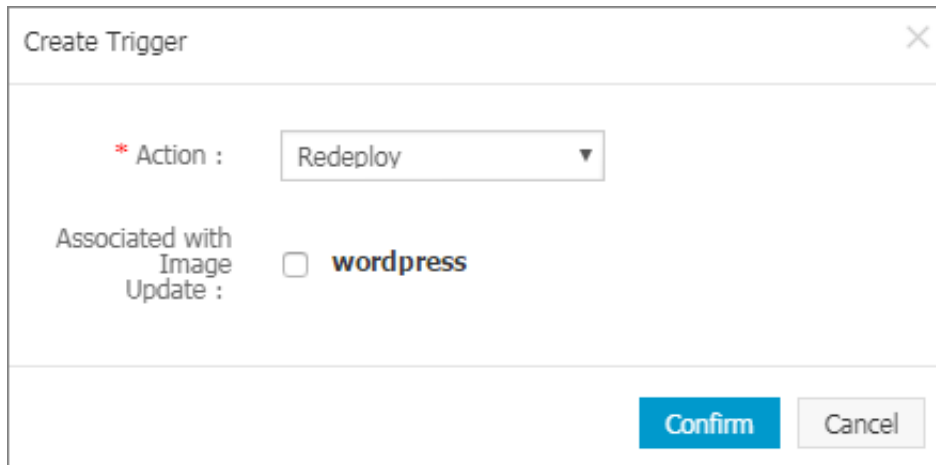


6. The Create Trigger dialog box appears. Select Redeploy or Resource Scaling from the Action list and then click Confirm.

- Redeploy

If you have Write permission to the image used by the application, you can select the Associated with Image Update check box. Then, the application can be automatically redeployed by using the latest image after the new image is constructed. Associated with Image Update check box. Then, the application

can be automatically redeployed by using the latest image after the new image is constructed.



The image shows a 'Create Trigger' dialog box. It has a title bar with 'Create Trigger' and a close button (X). The main area contains a form with the following elements:

- A label '* Action :' followed by a dropdown menu showing 'Redeploy'.
- A label 'Associated with Image Update :' followed by a checkbox and the text 'wordpress'.
- At the bottom right, there are two buttons: 'Confirm' (highlighted in blue) and 'Cancel'.

- **Resource scaling**

Select the service that needs to set the resource scaling trigger from the Service list.



Note:

To use the resource scaling trigger, upgrade the cluster Agent to the latest version.

Create Trigger
✕

* Action : Resource Scaling ▼

* Service : db ▼

1. If your Agent version is not up to date, you must go to the corresponding cluster and upgrade the Agent before creating a resource scaling trigger.

2. When calling a resource scaling trigger, you must manually add the following parameters into the trigger URL:

Parameter Name	Required?	Meaning	Value Options
type		Type of scaling	Reduce: scale_in Expand: scale_out
step		Quantity of scaling	Integer, 1-100

For example: https://cs.console.aliyun.com/hook/trigger?triggerUrl=yourTriggerUrl&secret=yourSecret&type=scale_out&step=5 indicates that when the trigger is called, the Container Service adds 5 instances into the cluster.

Confirm
Cancel

The generated trigger address is the API address.

Trigger Link (move mouse over to copy)	Secret (move mouse over to copy)	Type	Action
https://undefined/hook/trigger?triggerUrl=Yzc1ZGI1NGY1ZmExNjRjZDc4ZTU0MzMyNmZk4MjVjFHRlc3R8c2NhbgZ3wYTF2djlhaz2dubHVnfA==&secret=766634746f376246456c48324433306646cc677d18389f1436233e0b4808436b	766634746f376246456c48324433306646cc677d18389f1436233e0b4808436b	Resource Scaling	Delete Trigger

Subsequent operations


You can call the trigger by using a third-party integration system or a GET or POST request. For example, you can run the `CURL` command to call the trigger.

Call the redeployment trigger:

```
curl 'https://cs.console.aliyun.com/hook/trigger?triggerUrl=YzI4YTk5NzFkZWZkYzQ2MTJiOWZkNTM1MzY2ZDU1M2NifGNvbGxly3RkLWJlbnNobWFya3xyZWRLcGxvexWxOGLxbj'
```

```
c1Z25uMmVz fA ==& secret = 44586c6b46 6352395143 584c397065
4ff5323d25 09d546fdc1 b33054b092 8da8 '
```

Call the resource scaling trigger:

 **Note:**
 When calling the resource scaling trigger, manually add the following parameters to the trigger URL.

Parameter name	Required	Meaning	Optional value
type	Yes	The scaling type.	Contract: scale_in . Expand: scale_out .
step	Yes	The scaling number	Positive integer: 1-100.

For example, calling the following trigger will add five containers to the service.

```
curl 'https://cs.console.aliyun.com/hook/trigger
?triggerUrl=Y2IxZjI5YzhhYjIwMzRlMjBiYjc2OGUzYTlmZDgy
NDYyYzI5YzhhYjIwMzRlMjBiYjc2OGUzYTlmZDgy
NDYyYzI5YzhhYjIwMzRlMjBiYjc2OGUzYTlmZDgy
secret=53374142724e4e4a626f664a313131556e62c6716cd0d97d09
6900b3ad42a9ad&type=scale_out&step=5'
```