Container Service

API Reference

API Reference

Introduction

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

Note: Before using these interfaces, be sure that you fully understand the Container Service instructions, use agreements, and billing methods.

Terms

Term	Description
Cluster	Your container cluster. Multiple applications can be deployed in one cluster.
Node	One node in your container cluster. Currently, only ECS instances are supported.
Project	A complex project is composed of multiple services. A simplest project may have only one container.
Service	A group of containers defined with identical images and configurations can function as a scalable microservice.
Container	A running instance deployed from the Docker image.

API overview

The Container Service provides three types of APIs:

- Cluster management interfaces
- Application management interfaces
- Triggers

Cluster management interfaces

The Container Service provides some interfaces for cluster management, for example, cluster creation and deletion.

List of cluster management interfaces is as follows.

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

Application management interfaces

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

List of application interfaces is as follows.

API	Description
List Projects	View the list of application instances.
Create project	Create an application instance.
Retrieve project	View an application instance.
Start project	Start an application instance.
Stop project	Stop an application instance.
Kill project	Terminate an application instance.
Update project	Update an application instance.
Delete project	Delete an application instance.
List Services	View the list of service instances.
Retrieve service	View a service instance.
Start service	Start a service instance.
Stop service	Stop a service instance.
Kill service	Terminate a service instance.
Scale service	Scale a service instance.

Trigger

A trigger is an API provided by the Container Service for simple, fast and persistent deployment. For details, refer to Description.

Update history

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

Cluster API call method

Overview

Calling of the Container Service API interfaces are 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

Request structure

Service address

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

Communication protocol

The system supports request communication through the HTTP or HTTPS channel. It is recommended that you send requests over HTTPS for enhanced 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 unique request parameter for the specified operation.

Request encoding

Requests and returned results are encoded using the UTF-8 character set.

Public parameters

Public request headers

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

Parameter name	Description	Required?
Authorization	Authentication information used to verify the validity of a request, in the format of AccessKeyId:Signature.	Required
Content-Length	Content length of an HTTP request, which is defined in RFC 2616.	Required
Content-Type	Content type of an HTTP request, which is defined in RFC 2616.	Required
Content-MD5	Base64-encoded 128-bit MD5 hash value of the HTTP message body. It is	Required

	recommended that Content-MD5 be added to all requests to prevent request tampering.	
Date	Construction time of a request. At present, 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	Host access value, for example, diku.aliyuncs.com.	Required
Accept	Return-value type required by the client. Options: application/json and application/xml.	Required
x-acs-version	API version. The current version is 2015-12-15.	Required
x-acs-region-id	A region indicates the physical location of an ECS instance. For more details, refer to Region.	Required
x-acs-signature-nonce	Unique random number, used to prevent replay attacks. Different random numbers must be used for different requests.	Required
x-acs-signature-method	Signature method, which can only be HMAC-SHA1.	Required

Example

```
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 ACSbW2iBbyX0Pk9N:mp/6Wdr18V3qAc5gbgsqMe/iB0c
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 return headers

Each time you send a request to call an interface, the system will return a unique identification code (RequestId) to you, 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-AC049166F216</RequestId>
| <!—Return result data-->
</Interface name+Response>
```

JSON example:

```
{
"RequestId": "4C467B38-3910-447D-87BC-AC049166F216"
/* Return result data*/
}
```

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 interfaces are successfully called, data can be returned in XML (default) and JSON formats. When a request is sent, an external system can input a parameter to define the format of returned data.

In this document, examples of returned results are formatted in a way that is easier for you to view. The actual results returned are not formatted with line breaks, indentation, and so on.

Signature mechanism

Description

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 website). The Access Key ID indicates your identity. The Access Key Secret is the key used to encrypt a signature string and to verify the signature string on the server side. You must keep it confidential.

The Container Service verifies each access request it receives; therefore, all requests sent to the Container Service must contain signature information. The Container Service performs symmetric encryption 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. If they are different, the Container Service rejects the request and returns the HTTP 403 error.

You can add the Authorization header carrying signature information to the HTTP request, so as to indicate that the message has been authorized.

The Container Service requires that the signature be included 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"
+ CanonicalizedHeaders + "\n"
+ CanonicalizedResource))
```

- VERB indicates the HTTP method (for example, PUT).
- Accept indicates the return-value type required by the client, which can be application/json or application/xml.
- Content-MD5 indicates the MD5 value of the request.
- Content-Type indicates the content type of the request.
- Date indicates the operation time, which cannot be null. At present, 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 along with an error message and error code. (In the following example, the operation time is Wed, 16 Dec 2015 12:20:18 GMT.)
- CanonicalizedHeaders indicates a combination of fields prefixed with x-acs- in the HTTP request. (For details, refer to “NOTE”.)
- CanonicalizedResource indicates the uniform resource identifier (URI) of the resource in the HTTP request. (In the example, the URI is /clusters?name=my-clusters&resource=new.)

Note 1:

Process CanonicalizedHeaders (headers prefixed with x-acs-) in accordance with the following conventions before signature verification.

Convert the names of all HTTP request headers prefixed 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 specification. However, it is recommended that such names use only lowercase letters here.

If the value section of a public request header is too long, replace the \t, \n, \r, and \f separators with spaces.

Sort all HTTP request headers that are obtained from the preceding step and compliant with Alibaba Cloud specification in the lexicographically ascending order.

Delete any space at either side of a separator between each request header and content. For example, convert x-acs-meta-name: TaoBao,Alipay to x-acs-meta-name:TaoBao,Alipay.

Separate all headers and content with the \n separator to form the final CanonicalizedHeaders.

Note 2:

CanonicalizedResource format specification:CanonicalizedResource is the standard description of the resource to be accessed. Sort sub-resources along with query in the lexicographically ascending order and separate them 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 should be:

```
/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. It is recommended that you use your OpenAPI call program to calculate the signature string in the following example. Then compare your signature string to the result shown by the example.

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: 6U4ALMkKSj0PYbeQSHqgmA==
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:/uA9QF5CHrr1FK3siBA4xLMTWE0=

{"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: length of the body content (there is no space or line feed at the beginning of the body)

```
body: {"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"}

Content-Length: 210
```

Content-MD5: MD5 calculation process

```
body: {"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"}

# Calculate the MD5 value of the body.
md5(body): e94e002cc90a4a3d0f61b790487aa098

# Convert the MD5 value to a byte array. Convert every two hexadecimal symbols of the MD5 value to a byte.
# For example, e9 -> 11111111111111111111111111101001 -> -23
bytes(md5(body)): {[-23], [78], [0], [44], [-55], [10], [74], [61], [15], [97], [-73], [-112], [72], [122], [-96], [-104]}

# Convert the byte array into a Base64 string.
base64(bytes(md5(body))): 6U4ALMkKSj0PYbeQSHqgmA==

Content-MD5: 6U4ALMkKSj0PYbeQSHqgmA==
```

Process CanonicalizedHeaders

```
# List all headers prefixed with 'x-acs-'.

x-acs-version: 2015-12-15
```

```
x-acs-signature-nonce: ca480402-7689-43ba-acc4-4d2013d9d8d4
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 each request header and content.
# NOTE: There must be no line feed in the last line.

x-acs-region-id:cn-beijing
x-acs-signature-method:HMAC-SHA1
x-acs-signature-nonce:fbf6909a-93a5-45d3-8b1c-3e03a7916799
x-acs-signature-version:1.0
x-acs-version:2015-12-15
```

Calculate CanonicalizedResource

In the example, the length of CanonicalizedResource should be 27. An \n line feed is located at the end of the first line.

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

Calculate Signature

Assemble SignatureString. In the example, the length of the signature string is 307. An \n line feed is located at the end of all lines except the last line.

```
POST
application/json
6U4ALMkKSj0PYbeQSHqgmA==
application/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-3e03a7916799
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 Access Key Secret is access_key_secret.
hmac-sha1(SignatureString): fee03d405e421ebaf514adec881038c4b313584d

# Convert the encrypted string to a byte array, similar to the Content-MD5 calculation method.
# Convert the byte array into a Base64 string to get the final signature string.
base64(bytes(hmac-sha1(SignatureString))):
ZmVlMDNkNDA1ZTQyMWViYWY1MTRhZGVjODgxMDM4YzRiMzEzNTg0ZA==

Signature: ZmVlMDNkNDA1ZTQyMWViYWY1MTRhZGVjODgxMDM4YzRiMzEzNTg0ZA==
```

Finish

After the above processing, add other header information to construct the final HTTP request.

```
POST http://cs.aliyuncs.com/clusters?param1=value1&param2=value2 HTTP/1.1
Accept-Encoding: identity
Content-Length: 210
Content-MD5: 6U4ALMkKSj0PYbeQSHqgmA==
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:/uA9QF5CHrr1FK3siBA4xLMTWE0=

{"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"}
```

Cluster API list

View all cluster instances

Description

Views all clusters you have created through the Container Service.

If the name parameter is input, the system only returns the details of the specified cluster.

Request information

Request line (RequestLine)

```
GET /clusters?name={name} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description

name	string	No	Name of a cluster

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

```
[
{
"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"
}
]
```

Response body explanation

Cluster format

Name	Type	Description

agent_version	string	Agent version number.
cluster_id	String	ID that uniquely identifies a cluster.
created	string	Time of cluster creation.
external_loadbalancer_id	String	ID of the Server Load Balancer instance deployed for the cluster, for example.
master_url	string	Master address of the cluster, which is used to connect to the cluster to perform operations. For more details, refer to Connect to a cluster using Docker tools.
name	string	Name of the cluster, which is the name you specified when creating the cluster and is unique among the names of your clusters.
network_mode	String	Network mode of the cluster (options: **classic**, indicating a classic network, and **vpc**, indicating a VPC network).
region_id	String	ID of the region where the cluster is located. For more details, refer to Region.
security_group_id	String	ID of a security group.
size	String	Number of nodes.
state	String	Status of the cluster. For more details, refer to Lifecycle of a cluster.
updated	string	Last update time.
vpc_id	string	VPC ID.
vswitch_id	string	vSwitch ID.

Example

Request example

```
GET /clusters HTTP/1.1
<Public request headers>
```

Return example

```
HTTP/1.1 200 OK
<Public response header>

[
{
"agent_version": "0.5-e56dab3",
"cluster_id": "c978ca3eaacd3409a9437db07598f1f69",
"created": "2015-12-11T03:52:40Z",
"external_loadbalancer_id": "1518f2b7e4c-cn-beijing-btc-a01",
"master_url": "https://182.92.245.56:17589",
"name": "my-python-cluster-039de960",
"network_mode": "classic",
"region_id": "cn-beijing",
"security_group_id": "sg-25yqjuxhz",
"size": 5,
"state": "running",
"updated": "2015-12-15T15:01:58Z",
"vpc_id": "",
"vswitch_id": ""
},
{
"agent_version": "0.5-e56dab3",
"cluster_id": "c1eb19e0093204cbb86c3a80334d2129e",
"created": "2015-12-15T14:26:58Z",
"external_loadbalancer_id": "151a6099de1-cn-beijing-btc-a01",
"master_url": "https://182.92.245.56:11905",
"name": "my-test-cluster-002b3f3d",
"network_mode": "classic",
"region_id": "cn-beijing",
"security_group_id": "sg-25rg2ws9f",
"size": 1,
"state": "running",
"updated": "2015-12-15T14:43:55Z",
"vpc_id": "",
"vswitch_id": ""
}
]
```

Create a cluster instance

Description

Creates a cluster instance with the specified quantity of nodes.

Request information

Request line (RequestLine)

```
POST /clusters HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

```
ClusterCreation {
"name": "string",
"size": "number",
"network_mode": "string",
"subnet_cidr": "string",
"instance_type": "string",
"vpc_id": "string",
"vswitch_id": "string",
"password": "string",
"data_disk_size": "number",
"data_disk_category": "number"
"ecs_image_id": "string",
"io_optimized": "string",
}
```

Explanation on the request body

Name	Type	Required?	Description
name	String	Yes	Name of a cluster, which can contain uppercase and lowercase letters, Chinese characters, numbers, and dashes.
size	int	Yes	Quantity of ECS nodes in the cluster.
instance_type	string	Yes	Code indicating the ECS instance type. For more details,

			refer to Instance type table.
network_mode	String	Yes	Network mode of the cluster (options: **classic**, indicating a classic network, and **vpc**, indicating a VPC network.
subnet_cidr	String	Yes	Network address block that can be used by the cluster, for example, 192.168.24.0/22. This field needs to be set only when the network mode is VPC.
vpc_id	string	Yes	ID of a VPC network. This field needs to be set only when the network mode is VPC. For more information on VPC network, refer to VPC network overview.
vswitch_id	string	Yes	Switch ID of the VPC network. This field needs to be set only when the network mode is VPC.
password	string	Yes	Logon password of the root account.
data_disk_category	string	Yes	Disk category used by ECS. For more details, refer to Disk category table.
data_disk_size	number	Yes	Node-shared disk size.
ecs_image_id	string	No	ID of the system image used by ECS.
io_optimized	string	No	Depends the ECS instance type. Its value can be none or optimized.

ecs_image_id list

Region	System	ecs_image_id

Beijing	64-bit Ubuntu 14.04	m-25i16eaw8
Beijing	64-bit CentOS 7.0	m-25ksqahve
Hangzhou	64-bit Ubuntu 14.04	m-2313ixw4g
Hangzhou	64-bit CentOS 7.0	m-23i32bks6

If you need to define the ECS image ID of the cluster, make sure the ECS image meets the following requirements:

- OS: Ubuntu or CentOS.
- Linux Kernel version ≥ 3.18, used to support overlayfs and overlay network.
- The /etc/docker/key.json file is deleted from the image.

Response information

Response line (ResponseLine)

```
HTTP/1.1 202 Accepted
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

```
{
"cluster_id":"string",
"request_id":"string"
}
```

Example

Request example

```
POST /clusters HTTP/1.1
<Public request headers>

{
"password": "Just$test",
"instance_type": "ecs.s2.small",
"name": "my-cluster-001",
"size": 2,
```

```
"network_mode": "classic",
"data_disk_category": "cloud",
"data_disk_size": 20,
"charge_type": "PayByTraffic",
"ecs_image_id":"m-xx251ll"
}
```

Return example

```
HTTP/1.1 202 Accepted
<Public response header>

{
"cluster_id":"c01994faf5f364cc79b6ffb604061fa4a",
"request_id":"b44afc3c-46a4-4087-a215-c333a1218316"
}
```

Delete a cluster instance

Description

Deletes a cluster instance based on a cluster ID, and releases all node resources of the cluster.

Request information

Request line (RequestLine)

```
DELETE /clusters/{cluster_id} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
cluster_id	string	Yes	ID of a cluster

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 202 Accepted
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

None

Example

Request example

```
DELETE /clusters/Cccfd68c474454665ace07efce924f75f HTTP/1.1
<Public request headers>
```

Return example

```
HTTP/1.1 202 Accepted
<Public response header>
```

View a cluster instance

Description

Views cluster details based on a cluster ID.

Request information

Request line (RequestLine)

```
GET /clusters/{cluster_id} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
cluster_id	string	Yes	ID of a cluster

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

```
{
"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"
}
```


Response body explanation

Cluster format

Name	Type	Description
agent_version	string	Agent version number.
cluster_id	String	ID that uniquely identifies a cluster.
created	string	Time of cluster creation.
external_loadbalancer_id	String	ID of the Server Load Balancer instance deployed for the cluster.
master_url	string	Master address of the cluster, which is used to connect to the cluster to perform operations. For details, refer to Connect to a cluster using Docker tools.
name	string	Name of the cluster, which is the name you specified when creating the cluster and is unique among the names of your clusters.
network_mode	String	Network mode of the cluster (options: **classic**, indicating a classic network, and **vpc**, indicating a VPC network).
region_id	String	ID of the region where the cluster is located. For detailed information of region, refer to Region.
security_group_id	String	ID of a security group.
size	String	Number of nodes.
state	String	Status of the cluster. For details, refer to Lifecycle of a cluster.
updated	string	Last update time.
vpc_id	string	VPC ID.

vswitch_id	string	vSwitch ID.

Example

Request example

```
GET /clusters/C5b5e80b0b64a4bf6939d2d8fbbc5ded7 HTTP/1.1
<Public request headers>
```

Return example

```
HTTP/1.1 200 Ok
<Public response header>

{
"agent_version": "0.5-e56dab3",
"cluster_id": "c978ca3eaacd3409a9437db07598f1f69",
"created": "2015-12-11T03:52:40Z",
"external_loadbalancer_id": "1518f2b7e4c-cn-beijing-btc-a01",
"master_url": "https://182.92.245.56:17589",
"name": "my-python-cluster-039de960",
"network_mode": "classic",
"region_id": "cn-beijing",
"security_group_id": "sg-25yqjuxhz",
"size": 5,
"state": "running",
"updated": "2015-12-15T15:01:58Z",
"vpc_id": "",
"vswitch_id": ""
}
```

Obtain cluster certificate information

Description

Obtains the certificate of a cluster based on a cluster ID. Before you operate a cluster through Docker Client, you need to obtain the cluster certificate to access the cluster. Refer to Certificate configuration.

Request information

Request line (RequestLine)

```
GET /clusters/{cluster_id}/certs HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
cluster_id	string	Yes	ID of a cluster

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

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

ResponseBody parsing

Name	Type	Description
ca	string	Certificate issued by the Certification Authority (CA),

		ca.perm.
cert	string	Your public key certificate, cert.perm.
key	string	Your private key certificate, key.perm.

Example

Request example

```
GET /clusters/Cccfd68c474454665ace07efce924f75f/certs HTTP/1.1
<Public request headers>
```

Return example

```
HTTP/1.1 200 OK
<Public response header>

{
"ca": "-----BEGINCERTIFICATE-----ca content-----ENDCERTIFICATE-----\n",
"cert": "-----BEGINCERTIFICATE-----cert content-----ENDCERTIFICATE-----\n",
"key": "-----BEGINRSAPRIVATEKEY-----key content-----ENDRSAPRIVATEKEY-----\n"
}
```

Expand a cluster

Description

Updates the quantity of nodes in a cluster.

Request information

Request line (RequestLine)

```
PUT /clusters/{cluster_id} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
cluster_id	string	Yes	ID of a cluster

Special request header (RequestHead)

None. Refer to Public request headers.

Request body (RequestBody)

```
{
"password": "string",
"instance_type": "string",
"size": "number",
"data_disk_category": "number"
"data_disk_size": "number",
"ecs_image_id": "string",
}
```

Request body parsing

Name	Type	Required?	Description
password	string	No	ID of a cluster.
instance_type	String	No	Code indicating the ECS instance type. For more details, refer to Instance type table..
size	int	No	Quantity of nodes in the cluster.
data_disk_category	String	No	Disk category used by ECS. For more details, refer to Disk category table.
data_disk_size	Number	No	Node-shared disk size (unit: GB).
ecs_image_id	String	No	ID of the system image used by ECS.

ecs_image_id list

Region	System	ecs_image_id

Beijing	64-bit Ubuntu 14.04	m-25i16eaw8
Beijing	64-bit CentOS 7.0	m-25ksqahve
Hangzhou	64-bit Ubuntu 14.04	m-2313ixw4g
Hangzhou	64-bit CentOS 7.0	m-23i32bks6

Note:
If you need to define the ECS image ID of the cluster, make sure the ECS image meets the following requirements:

- OS: Ubuntu or Centos.
- Linux Kernel version ≥ 3.18, used to support overlayfs and overlay network
- The /etc/docker/key.json file is deleted from the image.

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None. Refer to Public return headers.

Response body (ResponseBody)

None

Example

Request example

```
PUT /clusters/Cccfd68c474454665ace07efce924f75f HTTP/1.1
<Public request headers>

{
"password": "password",
"instance_type": "ecs.s3.large",
"size": 2,
"data_disk_category": "cloud_ssd"
"data_disk_size": 500,
"ecs_image_id": "ubuntu1404_64_20G_aliaegis_20150130.vhd",
}
```

Return example

```
HTTP/1.1 202 Accepted
<Public response header>
```

Application API call method

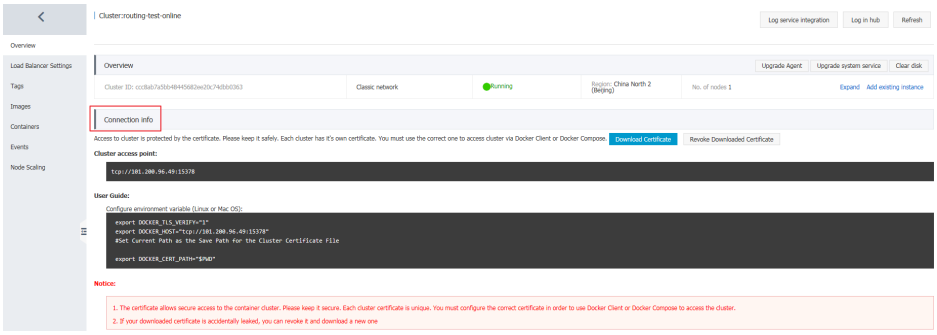
Call methods

The REST API of application management will be accessed from the cluster’ s management endpoint. And it is required HTTPS communication with the cluster certificate.

Obtain the management endpoint and certificates of the cluster

Console

Access the Container Service management console and select the connection information of the specified cluster to download the certificates. For details, refer to Connect to a cluster using Docker tools.



If you use API, change tcp in the endpoint to https.

Programming

You need to obtain the following items through the API to describe cluster.

1. Value of the master_url field. For details, refer to View a cluster instance.

2. The cluster certificate. For details, refer to Obtain cluster certificate information.

Results returned from the API:

```
{
"ca": "string", ##Certificate issued by the Certification Authority, ca.pem
"cert": "string", ##Your public key certificate, cert.pem
"key": "string" ##Your private key certificate, key.pem
}
```

It is recommended to save the three strings in the returned results as three files, namely, ca.pem, cert.pem, and key.pem, in one directory. The HTTPS certificate files can be used in tools or programming frameworks.

Call the application management API

Assume that your cluster name is ClusterName, and the above three certificates are saved to the ~/.docker/aliyun/ClusterName directory.

Assume that the obtained master_url address is https://123.123.123.123:1234.

Example of listing applications

The following describes how to list applications with REST API (the context path is /projects/).

For details, refer to View application instances.

CURL

```
# TIP: You may need to upgrade your CURL to the correct version.
curl --insecure --cert ~/.docker/aliyun/ClusterName/cert.pem --key ~/.docker/aliyun/ClusterName/key.pem
https://123.123.123.123:1234/projects/
```

PHP

```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://123.123.123.123:1234/projects/");
curl_setopt($ch, CURLOPT_SSLKEY, "~/.docker/aliyun/ClusterName/key.pem");
curl_setopt($ch, CURLOPT_CAINFO, "~/.docker/aliyun/ClusterName/ca.pem");
curl_setopt($ch, CURLOPT_SSLCERT, "~/.docker/aliyun/ClusterName/cert.pem");
$result=curl_exec($ch);
echo $result;
curl_close($ch);
?>
```

Python

```
import requests


res = requests.get('https://123.123.123.123:1234/projects/', verify='~/.docker/aliyun/ClusterName/ca.pem',
cert=('~/.docker/aliyun/ClusterName/cert.pem', '~/.docker/aliyun/ClusterName/key.pem'))
print res.content
```

Java

Add maven dependency

```
        <dependency>
<groupId>org.apache.httpcomponents</groupId>
<artifactId>httpclient</artifactId>
<version>4.5.1</version>
</dependency>

<dependency>
<groupId>org.bouncycastle</groupId>
<artifactId>bcpkix-jdk15on</artifactId>
<version>1.52</version>
</dependency>
```

Code sample

```
        import java.nio.file.Path;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.KeyFactory;
import java.security.KeyStore;
import java.security.PrivateKey;
import java.security.cert.Certificate;
import java.security.cert.CertificateFactory;
import java.security.spec.PKCS8EncodedKeySpec;
import javax.net.ssl.SSLContext;
import org.bouncycastle.openssl.PEMKeyPair;
import org.bouncycastle.openssl.PEMParser;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.ssl.SSLContexts;
import org.apache.http.util.EntityUtils;


public class Test {
```

```
public static void main(String[] argc) throws Exception {


final char[] KEY_STORE_PASSWORD = "".toCharArray();

//Obtain the certificate address
Path caCertPath = Paths.get("~/.docker/aliyun/ClusterName/ca.pem");
Path clientCertPath = Paths.get("~/.docker/aliyun/ClusterName/cert.pem");
Path clientKeyPath = Paths.get("~/.docker/aliyun/ClusterName/key.pem");

final CertificateFactory cf = CertificateFactory.getInstance("X.509");
final Certificate caCert = cf.generateCertificate(Files.newInputStream(caCertPath));
final Certificate clientCert = cf.generateCertificate(
Files.newInputStream(clientCertPath));
final PEMKeyPair clientKeyPair = (PEMKeyPair) new PEMParser(
Files.newBufferedReader(clientKeyPath,
Charset.defaultCharset()))
.readObject();
final PKCS8EncodedKeySpec spec = new PKCS8EncodedKeySpec(
clientKeyPair.getPrivateKeyInfo().getEncoded());
final KeyFactory kf = KeyFactory.getInstance("RSA");
final PrivateKey clientKey = kf.generatePrivate(spec);

//Set the trusted certificate
final KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
trustStore.load(null, null);
trustStore.setEntry("ca", new KeyStore.TrustedCertificateEntry(caCert), null);

//Set the private key
final KeyStore keyStore = KeyStore.getInstance(KeyStore.getDefaultType());
keyStore.load(null, null);
keyStore.setCertificateEntry("client", clientCert);
keyStore.setKeyEntry("key", clientKey, KEY_STORE_PASSWORD, new Certificate[]{clientCert});

SSLContext sslContext = SSLContexts.custom()
.loadTrustMaterial(trustStore, null)
.loadKeyMaterial(keyStore, KEY_STORE_PASSWORD)
.build();
SSLConnectionSocketFactory sslsf = new SSLConnectionSocketFactory(
sslContext,
SSLConnectionSocketFactory.getDefaultHostnameVerifier());

//httpclient connection
CloseableHttpClient httpclient = HttpClients.custom()
.setSSLSocketFactory(sslsf)
.build();
try {

HttpGet httpget = new HttpGet("https://123.123.123.123:1234/projects/");
CloseableHttpResponse response = httpclient.execute(httpget);
try {
System.out.println("----------------------------------------");
String bodyAsString = EntityUtils.toString(response.getEntity());
System.out.println(bodyAsString);

} finally {
```

```
response.close();
}
} finally {
httpclient.close();
}
}

}
```

Application API list

View application instances

Description

Lists application instances in the container cluster.

Request information

Request line (RequestLine)

```
GET /projects/ HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
q	string	No	Application name.
services	boolean	No	Includes service information or not. The default value is true.
containers	boolean	No	Includes service container information or not. The default value is true.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

```
[
{
"name": "string",
"description": "string",
"template": "string",
"version": "string",
"created": "datetime",
"updated": "datetime",
"desired_state": "string",
"current_state": "string",
"environment": {
"key": "value",
...
}
"services": [
...
]
}
]
```

Response body explanation

Project format

Name	Type	Description

name	string	Name
description	string	Description
template	string	Docker Compose template
version	string	Applicaiton version
created	datetime	Creation time
updated	datetime	Update time
desired_state	string	Desired state
current_state	string	Current state
environment	map	Key/value of environment variable
services	array	List of services

Example

Request example

```
GET /projects HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8

[
{
"name": "test",
"description": "This is a test application",
"template": "...",
"version": "1.0",
"environment": {
"COMPOSE_PROJECT_NAME": "test"
},
"created": "2016-02-02T07:45:13.113833319Z",
"updated": "2016-02-02T07:45:16.03142154Z",
"desired_state": "running",
"current_state": "running",
"services": [
...
]
},
...
]
```

Create an application instance

Description

Creates an application instance.

Request information

Request line (RequestLine)

```
POST /projects/ HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

```
Content-Type: application/json
```

Request body (RequestBody)

JSON object

```
{
"name": "string",
"description": "string",
"template": "string",
"version": "string",
"environment": {
"key": "value",
...
}
}
```

Explanation on the request body

Name	Type	Required?	Description
name	string	Yes	Name of application,

			which can contain uppercase and lowercase letters, numbers, and dashes.
description	string	No	Description.
template	string	Yes	The Docker Compose template in yaml format. It should be encoded as one JSON string.
version	string	No	Application version. The default value is 1.0.
environment	map	No	key/value, used to replace the variable parameter in the Compose template.
latest_image	bool	No	Whether to update the image before creating the application.

Response information

Response line (ResponseLine)

```
HTTP/1.1 201 Created
```

Special response header (ResponseHead)

```
Location /projects/<name>
```

Example

Request example

```
POST /projects HTTP/1.1
Content-Type: application/json

{
"name": "test",
```

```
"description": "This is a test application",
"template": "web:\r\n image: nginx",
"version": "1.0",
"environment": {
"USER": "abc",
"PWD": "password"
}
}
```

Return example

```
HTTP/1.1 201 Created
Location /projects/test
```

Delete an application instance

Description

Deletes an application instance.

Request information

Request line (RequestLine)

```
DELETE /projects/{name}?force={force}&v={volume} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application instance.
force	string	No	Force deletion or not. Options include true (1) and false (0). The default value is false.
volume	string	No	Remove the container volume or not. Options include true (1) and false (0). The default value is

			false.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
DELETE /projects/test HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

View an application instance

Description

View the details of application instance by name.

Request information

Request line (RequestLine)

```
GET /projects/{name} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application instance

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

```
{
"name": "string",
"description": "string",
```

```
"template": "string",
"version": "string",
"created": "datetime",
"updated": "datetime",
"desired_state": "string",
"current_state": "string",
"environment": {
"key": "value",
...
}
"services": [
...
]
}
```

Response body explanation

Format of the application instance

Name	Type	Description
name	string	Name
description	string	Description
template	string	Docker Compose template
version	string	Application version
created	datetime	Creation time
updated	datetime	Update time
desired_state	string	Desired state
current_state	string	Current state
environment	map	Environment variables
services	array	List of services

Example

Request example

```
GET /projects/test HTTP/1.1
```

Return example

```
HTTP/1.1 200 Ok
Content-Type:application/json;charset=UTF-8
```

```
{
"name": "test",
"description": "This is a test application",
"template": "...",
"version": "1.0",
"environment": {
"COMPOSE_PROJECT_NAME": "test"
},
"created": "2016-02-02T07:45:13.113833319Z",
"updated": "2016-02-02T07:45:16.03142154Z",
"desired_state": "running",
"current_state": "running",
"services": [
...
]
}
```

Start an application instance

Description

Starts an application instance. The containers will be started in order of service dependencies.

Request information

Request line (RequestLine)

```
POST /projects/{name}/start HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /projects/test/start HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Stop an application instance

Description

Stops all containers in an application instance. The containers will be stopped in a reverse order of services dependencies.

Request information

Request line (RequestLine)

```
POST /projects/{name}/stop?t={timeout} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application.
timeout	int	No	Timeout time (in seconds) to stop the container. The default value is 10.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /projects/test/stop HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Terminate an application instance

Description

Sends signals to all containers in an application instance. The container will be notified in a reverse order of service dependencies. The default signal is to terminate container.

Request information

Request line (RequestLine)

```
POST /projects/{name}/kill?signal={signal} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application.
signal	string	No	Signal name, the default value is KILL.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /projects/test/kill HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Update an application instance

Description

Updates configurations of an application instance.

Request information

Request line (RequestLine)

```
POST /projects/{name}/update HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
name	string	Yes	Name of the application instance.

Special request header (RequestHead)

```
Content-Type: application/json
```

Request body (RequestBody)

JSON object

```
{
"description": "string",
"template": "string",
"version": "string",
"environment": {
"key": "value",
...
}
}
```

Request body parsing

Name	Type	Required?	Description
description	string	No	Updated description.
template	string	Yes	Updated Docker Compose yaml template, which must be encoded in JSON string.
version	string	No	Updated application version. The new version must be different from the old version; otherwise, HTTP

			code 409 is returned.
environment	map	Environmental variable	key/value pairs, used to replace the environment variables in the Compose template.

Response information

Response line (ResponseLine)

```
HTTP/1.1 202 Accepted
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /projects/test/update HTTP/1.1
Content-Type: application/json


{
"description": "This is a test application",
"template": "web:\r\n image: nginx",
"version": "2.0",
"environment": {
"USER": "abc",
"PWD": "newpwd"
}
}
```

Return example

```
HTTP/1.1 202 Accepted
```

View service instances

Description

View all service instances in the container cluster.

Request information

Request line (RequestLine)

```
GET /services/ HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
q	string	No	Service name.
containers	boolean	No	Includes service container information or not. The default value is true.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```

```

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

```
[
{
"id": "string",
"name": "string",
"project": "string",
"description": "string",
"created": "datetime",
"updated": "datetime",
"desired_state": "string",
"current_state": "string",
"definition": {
"key": "value",
...
},
"extensions": {
"key": "value",
...
},
"containers": {
"key": "value",
...
}
},
...
]
```

Response body explanation

Format of the service instance

Name	Type	Description
id	string	Service ID.
name	string	Service name.
project	string	Application name.
created	datetime	Service creation time.
updated	datetime	Service update time.
desired_state	string	Desired state (If the application is in the intermediate state, the parameter specifies the final

		state of the application).
current_state	string	Current state.
definition	map	Compose service definition key/value.
extensions	map	Expansion key/value of services in the Container Service Compose.
containers	map	Container key (container ID)/value (attribute) in the service.

Example

Request example

```
GET /services/ HTTP/1.1
```

Return example

```
HTTP/1.1 200 Ok
Content-Type:application/json;charset=UTF-8

[
{
"id": "wordpress_db",
"name": "db",
"project": "wordpress",
"definition": {
"environment": [
"MYSQL_ROOT_PASSWORD=password"
],
"image": "mysql:5.7",
"restart": "always"
},
"extensions": {
"scale": 1,
"logs": [
"/var/log/mysql"
]
},
"created": "2016-04-21T13:36:32.440646459Z",
"updated": "2016-04-21T13:36:33.270308958Z",
"desired_state": "running",
"current_state": "running",
"containers": {
"5616f05d27516b3502a391fd2ca9d312cabffa5ad431bf261ea81f4ceabd476e": {
"name": "/wordpress_db_1",
```

```
"node": "10.246.2.3",
"ip": "10.0.0.2",
"running": true,
"status": "running",
"health": "success"
}
}
}
]
```


View a service instance

Description

View the details of a service instance by ID.

Request information

Request line (RequestLine)

```
GET /services/{service_id} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
service_id	string	Yes	Service instance ID, in the format of {project_name}_{service_name}.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

```
{
"id": "string",
"name": "string",
"project": "string",
"description": "string",
"created": "datetime",
"updated": "datetime",
"desired_state": "string",
"current_state": "string",
"definition": {
"key": "value",
...
},
"extensions": {
"key": "value",
...
},
"containers": {
"key": "value",
...
}
}
```

Response body explanation

Format of the application instance

Name	Type	Description
id	string	Service ID.
name	string	Service name.
project	string	Application name.
created	datetime	Service creation time.
updated	datetime	Service update time.
desired_state	string	Desired state (If the application is in the intermediate state, the

		parameter specifies the final state of the application).
current_state	string	Current state.
definition	map	Compose service definition key/value.
extensions	map	Expansion key/value of services in the Container Service Compose.
containers	map	Container key (container ID)/value (attribute) in the service.

Example

Request example

```
GET /services/wordpress_db HTTP/1.1
```

Return example

```
HTTP/1.1 200 Ok
Content-Type:application/json;charset=UTF-8

{
"id": "wordpress_db",
"name": "db",
"project": "wordpress",
"definition": {
"environment": [
"MYSQL_ROOT_PASSWORD=password"
],
"image": "mysql:5.7",
"restart": "always"
},
"extensions": {
"scale": 1,
"logs": [
"/var/log/mysql"
]
},
"created": "2016-04-21T13:36:32.440646459Z",
"updated": "2016-04-21T13:36:33.270308958Z",
"desired_state": "running",
"current_state": "running",
"containers": {
"5616f05d27516b3502a391fd2ca9d312cabffa5ad431bf261ea81f4ceabd476e": {
"name": "/wordpress_db_1",
```

```
"node": "10.246.2.3",
"ip": "10.0.0.2",
"running": true,
"status": "running",
"health": "success"
}
}
}
```


Start a service instance

Description

Starts all containers in a service instance.

Request information

Request line (RequestLine)

```
POST /services/{service_id}/start HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
service_id	string	Yes	Service instance ID, in the format of {project_name}_{service_name}.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /service/test_web/start HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Stop a service instance

Description

Stops all containers in a service instance.

Request information

Request line (RequestLine)

```
POST /services/{service_id}/stop?t={timeout} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
service_id	string	Yes	Service instance ID, in the format of {project_name}_{service_name}.
timeout	int	No	Timeout time (in seconds) to stop the container. The default value is 10.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /services/test_web/stop HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Terminate a service instance

Description

Sends signals to all containers in the service instance. The default signal is to terminate container.

Request information

Request line (RequestLine)

```
POST /services/{service_id}/kill?signal={signal} HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
service_id	string	Yes	Service instance ID, in the format of {project_name}_{service_name}.
signal	string	No	The default value is KILL.

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

```
POST /services/test_web/kill HTTP/1.1
```

Return example

```
HTTP/1.1 200 OK
```

Scale a service instance

Description

Scales the number of containers in a service instance to a specific number.

Request information

Request line (RequestLine)

```
POST /services/{service_id}/scale HTTP/1.1
```

Request line parameter (URI Param)

Name	Type	Required?	Description
service_id	string	Yes	Service instance ID, in the format of {project_name}_{service_name}.

Special request header (RequestHead)

None

Request body (RequestBody)

JSON object

```
{
"type": "string",
"value": "int"
}
```

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Special response header (ResponseHead)

None

Response body (ResponseBody)

None

Example

Request example

Scale containers of the **test web** service to 3

```
POST /services/test_web/scale HTTP/1.1
Content-Type: application/json

{
"type": "scale_to",
"value": 3
}
```

Return example

```
HTTP/1.1 200 OK
```

Create a data volume

Description

Create a data volume in the cluster.

Request information

Request line (RequestLine)

```
POST /volumes/create HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

```
Content-Type: application/json
```

Request body (RequestBody)

JSON object

```
{
"name": "***",
```

```
"driver": "***",
"driverOpts": {
"para1": "value",
"para2": "value",
...
}
}
```

Explanation on the request body

Name	Type	Required	Description
name	string	Yes	The name of the data volume. The name should contain 1~64 characters. It can contain numbers, English letters and hyphens (-) but cannot start with a hyphen (-).
driver	string	Yes	The type of the data volume. Currently, the Container Service supports ossfs data volumes.
driverOpts	DriverOptions	Yes	The configuration parameter options of the data volume. The configuration parameters differs for different types of data volumes. For OSS data volumes, it is OSSOpts.

The definitions of OSSOpts type of data volumes are as follows:

OSSOpts:

Name	Type	Required	Description
bucket	string	Yes	The OSS bucket name. You can get it from the OSS Management Console.
ak_id	string	Yes	The Access Key ID required for users to access the OSS resources.
ak_secret	string	Yes	The Access Key

			Secret required for users to access the OSS resources.
url	string	Yes	The domain name provided by the OSS bucket. You can get it from the OSS Management Console.
no_stat_cache	string	Yes	File caching. If you do not need to synchronize the changes of a file among different machines, disable caching.
other_opts	string	Yes	Configuration parameters for connecting OSS.

Response information

Response line (ResponseLine)

```
HTTP/1.1 201 Created
```

Response body (ResponseBody)

JSON object

```
{
"Name": "volume",
"Driver": "***",
"Mountpoint": "/mnt/acs_mnt/**/***",
"Labels": null,
"Scope": ""
}
```

Response body explanation

Name	Type	Description
Name	string	The name of the data volume.
Driver	string	The type of the data volume driver. It can be ossfs.

Mountpoint	string	The mount point of the data volume on the host: /mnt/acs_mnt/**/***.
Labels	map[string]string	The meta data information of the data volume.
Scope	string	The data volume management range. global indicates the cluster level; local indicates inside the host.

OSS data volume example

Request example

```
{
"name": "ossvolume",
"driver": "ossfs",
"driverOpts": {
"bucket": "aliyun-docker",
"ak_id": "****",
"ak_secret": "*****",
"url": "oss-cn-hangzhou.aliyuncs.com",
"no_stat_cache": "true",
"other_opts": "-o allow_other -o default_permission=666"
}
}
```

Return example

```
{
"Name": "ossvolume",
"Driver": "ossfs",
"Mountpoint": "/mnt/acs_mnt/ossfs/aliyun-docker",
"Labels": null,
"Scope": ""
}
```

View a data volume

Description

View the detailed information of a data volume.

Request information

Request line (RequestLine)

```
Get /volumes/{name} HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Response body (ResponseBody)

JSON object

```
{
"Name": "volume",
"Driver": "***",
"Mountpoint": "/mnt/acs_mnt/**/***",
"Labels": null,
"Scope": "",
"Node": "***",
"Refs": []
}
```

Name	Type	Description
Name	string	The name of the data volume.
Driver	string	The type of the data volume

		driver. It can be ossfs.
Mountpoint	string	The mount point of the data volume on the host: /mnt/acs_mnt/**/***.
Labels	map[string]string	The meta data information of the data volume.
Scope	string	The data volume management range. global indicates the cluster level; local indicates inside the host.
Node	string	The name of the host of the data volume.
Refs	[]VolumeRef	The list of containers that use this data volume. For details, refer to the explanation of VolumeRef.

VolumeRef data type:

Name	Type	Description
Name	string	Container name
ID	string	Container ID

Example

Return example

```
{
"Name": "volume",
"Driver": "ossfs",
"Mountpoint": "/mnt/acs_mnt/ossfs/bucket",
"Labels": null,
"Scope": "",
"Node": "asd903233s23q-node1",
"Refs": []
}
```

View data volume list

Description

View all the data volumes under the current cluster.

Request information

Request line (RequestLine)

```
Get /volumes HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 200 OK
```

Response body (ResponseBody)

JSON object

```
{
"Name": "volume",
"Driver": "***",
"Mountpoint": "/mnt/acs_mnt/**/***",
"Labels": null,
"Scope": "",
"Node": "***",
"Refs": []
}
```

Name	Type	Description
Name	string	The name of the data volume.
Driver	string	The type of the data volume driver. It can be ossfs.
Mountpoint	string	The mount point of the data volume on the host: /mnt/acs_mnt/**/***.
Labels	map[string]string	The meta data information of the data volume.
Scope	string	The data volume management range. global indicates the cluster level; local indicates inside the host.
Node	string	The name of the host of the data volume.
Refs	[]VolumeRef	The list of containers that use this data volume. For details, refer to the explanation of VolumeRef.

VolumeRef data type:

Name	Type	Description
Name	string	Container name
ID	string	Container ID

Example

Return example


```
{
"Volumes": [
{
"Name": "docker2",
"Driver": "ossfs",
"Mountpoint": "/mnt/acs_mnt/ossfs/bucket2",
"Labels": null,
"Scope": "",
"Node": "b347bcf0e68b",
"Refs": []
},
{
"Name": "docker1",
```

```
"Driver": "ossfs",
"Mountpoint": "/mnt/acs_mnt/ossfs/bucket1",
"Labels": null,
"Scope": "",
"Node": "b347bcf0e68b",
"Refs": []
}
]
}
```

Delete a data volume

Description

Delete the specified data volume.

Request information

Request line (RequestLine)

```
Delete /volumes/{name} HTTP/1.1
```

Request line parameter (URI Param)

None

Special request header (RequestHead)

None

Request body (RequestBody)

None

Response information

Response line (ResponseLine)

```
HTTP/1.1 204 No Content
```

Response body (ResponseBody)

None

Trigger

Introduction

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

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

Currently, the Container Service provides redeployment trigger and 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 Hub container, so that the container can be automatically redeployed using the new image after the new image is constructed.

Scaling trigger

You can call the scaling trigger to realize container scaling.

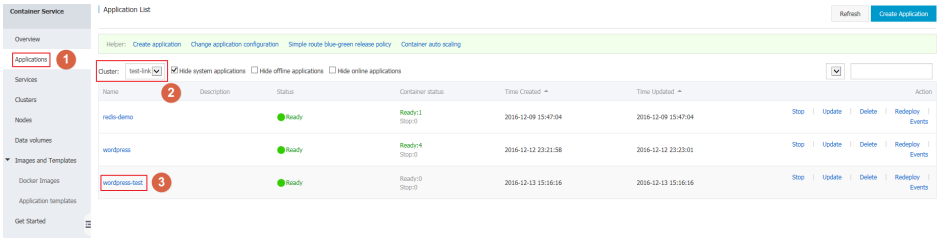
Create a trigger

Procedure

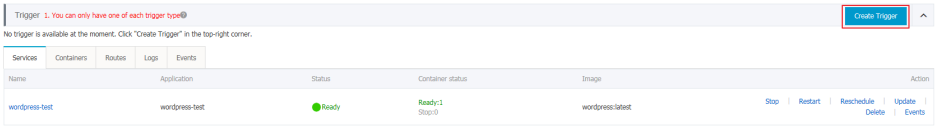
Log on to the Container Service console

Click **Applications** in the left navigation pane.

Select the target cluster and click the name of the application, as shown in the figure below.



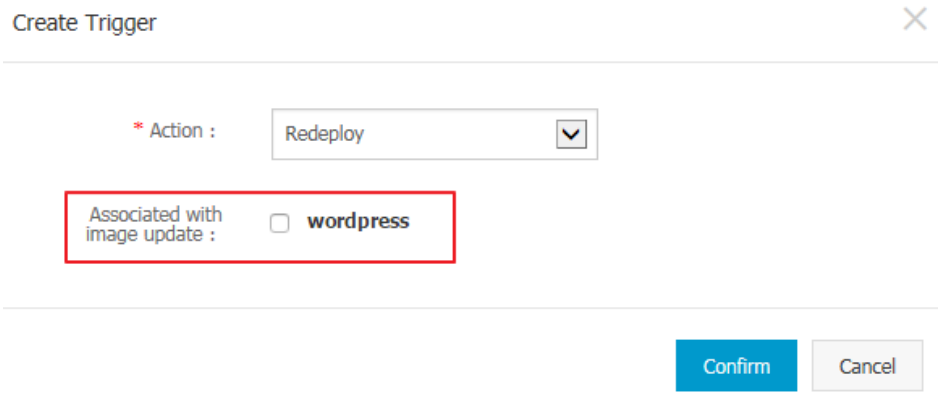
Click **Create Trigger**.



Select **Redeploy** or **Scaling** in **Action** and click **Confirm**.

Redeploy

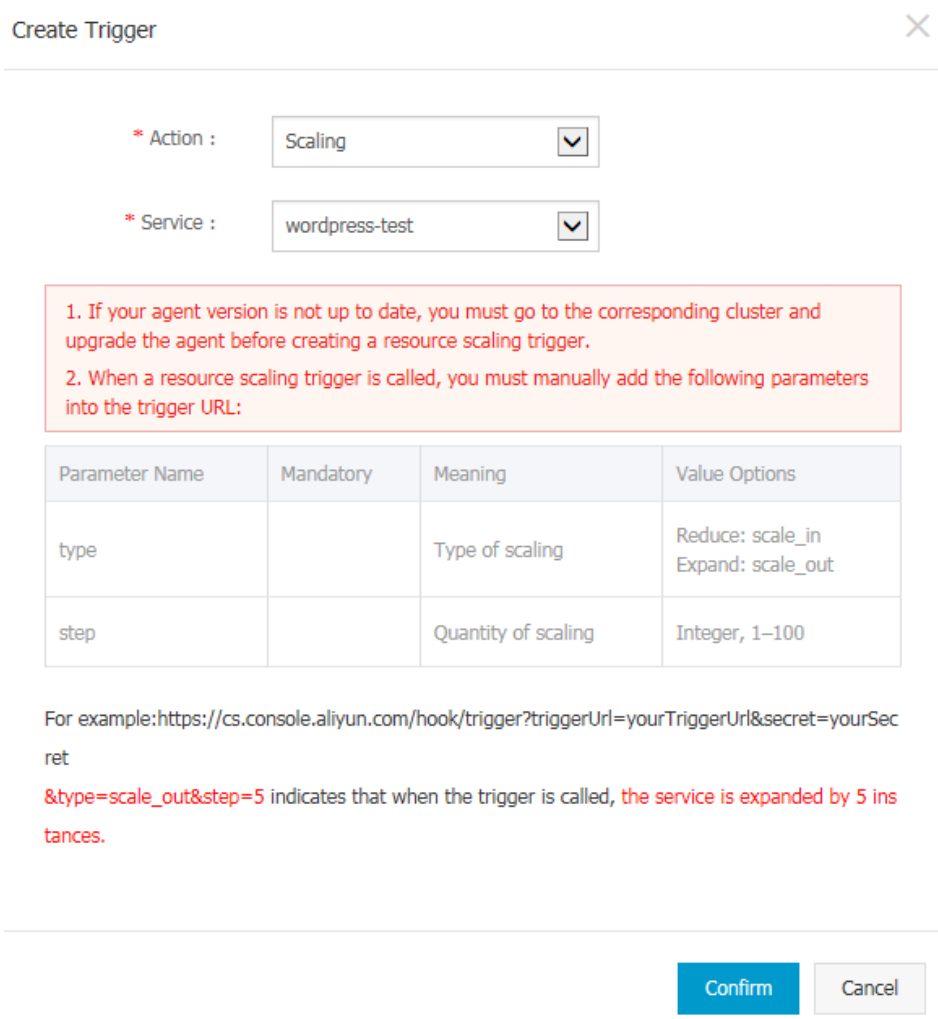
When you have write permission on the image used by the application, you can select **Associated with image update**. Then, the container can be automatically redeployed using the new image after the new image is constructed.



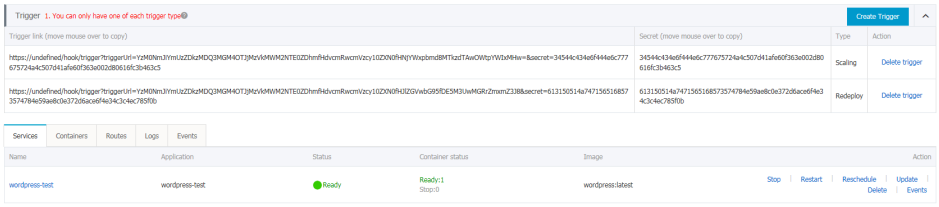
Scaling

Select the desired service in the **Service** drop-down list.

> **Note:** To use the scaling trigger, you need to upgrade the cluster Agent to the latest version.



The trigger IP addresses generated are the API IP addresses.



Subsequent operations

You can call the trigger through a third-party 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=YzI4YTk5NzFkZWZkYzQ2MTJiOWZkNTM1MzY2ZDU1M2NifGNvbGxlY3RkLWJlbmNobWFya3xyZWRlcGxveXwxOGlxbjc1Z25uMmVzfA==&secret=44586c6b466352395143584c3
```

```
970654ff5323d2509d546fdc1b33054b0928da8'
```

Call the scaling trigger:

Note: When calling the scaling trigger, you need to add the following parameters in the URL:

Parameter	Required	Meaning	Value
type	Yes	The type of scaling	Reduce: scale_in; expand: scale_out
step	Yes	The number of containers to be removed or added	Positive integer: 1~100

For example, calling the following trigger will add 5 containers to the service.

```
curl
'https://cs.console.aliyun.com/hook/trigger?triggerUrl=Y2IxZjI5YzhhYjIwMzRlMjBiYjc2OGUzYTlmZDgyNDAyfHdvcmRwcmVzcy10ZXN0fHNjYWxpbmd8MTkzZmEyMXFwZXVwMXw=&secret=53374142724e4e4a626f664a313131556e62c6716cd0d97d096900b3ad42a9ad&type=scale_out&step=5'
```

Appendix

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.
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 nodes of the cluster.
Deleting	The cluster is being deleted.
Deleted	The cluster is successfully deleted.