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, consequent
ial, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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.
## Document conventions

<table>
<thead>
<tr>
<th>Style</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="danger" /></td>
<td>A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.</td>
<td><img src="image" alt="danger" /> Danger: Resetting will result in the loss of user configuration data.</td>
</tr>
<tr>
<td><img src="image" alt="warning" /></td>
<td>A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.</td>
<td><img src="image" alt="warning" /> Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.</td>
</tr>
<tr>
<td><img src="image" alt="caution" /></td>
<td>A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.</td>
<td><img src="image" alt="caution" /> Notice: If the weight is set to 0, the server no longer receives new requests.</td>
</tr>
<tr>
<td><img src="image" alt="note" /></td>
<td>A note indicates supplemental instructions, best practices, tips, and other content.</td>
<td><img src="image" alt="note" /> Note: You can use Ctrl + A to select all files.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Closing angle brackets are used to indicate a multi-level menu cascade.</td>
<td>Click Settings &gt; Network &gt; Set network type.</td>
</tr>
<tr>
<td><strong>Bold</strong></td>
<td>Bold formatting is used for buttons, menus, page names, and other UI elements.</td>
<td>Click OK.</td>
</tr>
<tr>
<td><strong>Courier font</strong></td>
<td>Courier font is used for commands.</td>
<td>Run the <code>cd /d C:/window</code> command to enter the Windows system folder.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic formatting is used for parameters and variables.</td>
<td><code>bae log list --instanceid Instance_ID</code></td>
</tr>
<tr>
<td>[] or [a</td>
<td>b]</td>
<td>This format is used for an optional value, where only one item can be selected.</td>
</tr>
<tr>
<td>Style</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>{} or {a</td>
<td>b}</td>
<td>This format is used for a required value, where only one item can be selected.</td>
</tr>
</tbody>
</table>
Contents

Legal disclaimer........................................................................................................... I
Document conventions............................................................................................... I
1 Authorizations........................................................................................................ 1
  1.1 Role authorization............................................................................................... 1
  1.2 Upgrade sub-account policy............................................................................... 5
  1.3 Create custom authorization policies............................................................... 8
2 Clusters.................................................................................................................. 12
  2.1 Cluster introduction........................................................................................... 12
  2.2 Cluster lifecycle................................................................................................. 13
  2.3 Create a cluster................................................................................................ 14
  2.4 Cluster parameter configurations...................................................................... 22
  2.5 Add an existing ECS instance........................................................................... 26
  2.6 Manage cross-zone nodes................................................................................ 30
  2.7 Bind and unbind a Server Load Balancer instance........................................... 33
  2.8 Set the root domain name of a cluster.............................................................. 35
  2.9 Download cluster certificate............................................................................ 38
  2.10 Expand a cluster............................................................................................... 40
  2.11 Migrate a cluster............................................................................................. 41
  2.12 Search for a cluster......................................................................................... 42
  2.13 Delete a cluster............................................................................................... 43
  2.14 Clean up a cluster disk.................................................................................... 43
  2.15 Log on to image repository.............................................................................. 44
  2.16 Upgrade Agent................................................................................................. 46
  2.17 Upgrade Docker daemon.................................................................................. 47
  2.18 Upgrade system services.................................................................................. 49
3 Nodes..................................................................................................................... 51
  3.1 View containers running on a node................................................................... 51
  3.2 Update a node certificate.................................................................................. 52
4 Images and templates............................................................................................. 54
  4.1 Update an orchestration template..................................................................... 54
5 Service orchestrations............................................................................................ 56
  5.1 routing............................................................................................................... 56
6 Applications........................................................................................................... 59
  6.1 Create an application......................................................................................... 59
  6.2 Schedule an application to specified nodes..................................................... 68
7 Configurations.......................................................................................................... 72
  7.1 Implement multiple environments by using configurations............................ 72
8 Data volumes........................................................................................................... 78
8.1 Overview............................................................................................... 78
8.2 Create an OSSFS data volume.................................................................78
8.3 Create cloud disk data volumes..............................................................82
8.4 View and delete data volumes.............................................................. 86
8.5 Use third-party data volumes............................................................... 87
8.6 FAQ........................................................................................................92

9 Logs........................................................................................................... 93
  9.1 Enable Log Service............................................................................... 93

10 DevOps.................................................................................................98
  10.1 Jenkins-based continuous delivery.................................................... 98

11 Service discovery and load balancing............................................... 110
  11.1 Routing and Server Load Balancer between services in a cluster........110
  11.2 Custom routing - simple sample....................................................... 113
  11.3 Custom routing - Supports TCP......................................................... 123
  11.4 Custom routing - supports multiple HTTPS certificates.................... 126
1 Authorizations

1.1 Role authorization

Grant the system default roles AliyunCSDefaultRole and AliyunCSClusterRole to the service account when you activate Container Service. Only after the roles are correctly granted, Container Service can normally call the services such as Elastic Compute Service (ECS), Object Storage Service (OSS), NAS, and Server Load Balancer (SLB), create clusters, and store logs.

Instructions

- If you have used Container Service before 15 January 2018, the system completes the role authorization by default. For the detailed granted permissions, see the following Default role permissions section. If you used Container Service with a Resource Access Management (RAM) user before, upgrade the authorization policy for the RAM user. For more information, see Upgrade sub-account policy.

- On 15 January 2018, Container Service is fully accessed to the cross-service authorization. New users who use the primary account can use Container Service only after having the cross-service authorization completed. If new users need to authorize RAM users to use Container Service, go to the RAM console to authorize the RAM users. For more information, see #unique_6.
Procedure

1. If you have not granted the default roles to the service account correctly, the Cloud Resource Access Authorization page appears after you log on to the Container Service console. Click Confirm Authorization Policy.

   ![Cloud Resource Access Authorization](image)

   **Note:** Container Service has configured the default role permissions. To modify the role permissions, go to the User Management page of the RAM console. Note that incorrect configurations might cause Container Service cannot obtain the required permissions.

2. After completing the authorization, refresh the Container Service console and then perform the operations.

   To view the policy details of the roles AliyunCSDefaultRole and AliyunCSClusterRole, log on to the RAM console.

Default role permissions

For more information about permissions of each role, see the API documents of each product.

AliyunCSDefaultRole permissions

The default role AliyunCSDefaultRole contains the following main permissions:

- ECS-related permissions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecs:RunInstances</td>
<td>Query ECS instance information.</td>
</tr>
<tr>
<td>ecs:RenewInstance</td>
<td>Renew ECS instances.</td>
</tr>
</tbody>
</table>
### Action | Description
--- | ---
ecs:Create* | Create ECS-related resources, such as instances and disks.
ecs:AllocatePublicIpAddress | Allocate public IP addresses.
ecs:AllocateEipAddress | Allocate Elastic IP (EIP) addresses.
ecs:Delete* | Delete ECS instances.
ecs:StartInstance | Start ECS-related resources.
ecs:StopInstance | Stop ECS instances.
ecs:RebootInstance | Restart ECS instances.
ecs:Describe* | Query ECS-related resources.
ecs:AuthorizeSecurityGroup | Configure inbound security group rules.
ecs:RevokeSecurityGroup | Revoke security group rules.
ecs:AuthorizeSecurityGroupEgress | Configure outbound security group rules.
ecs:AttachDisk | Add disks.
ecs:DetachDisk | Clean up disks.
ecs:AddTags | Add tags.
ecs:ReplaceSystemDisk | Change system disks of ECS instances.
ecs:ModifyInstanceAttribute | Modify ECS instance attributes.
ecs:JoinSecurityGroup | Add ECS instances to specified security groups.
ecs:LeaveSecurityGroup | Remove ECS instances from specified security groups.
ecs:UnassociateEipAddress | Unbind EIP addresses.
ecs:ReleaseEipAddress | Release EIP addresses.

- **Virtual Private Cloud (VPC)-related permissions**

| Action | Description |
--- | ---
vpc:Describe* | Query information of VPC-related resources.
vpc:DescribeVpcs | Query VPC information.
vpc:AllocateEipAddress | Allocate EIP addresses.
vpc:AssociateEipAddress | Associate with EIP addresses.
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpc:UnassociateEipAddress</td>
<td>Do not associate with EIP addresses.</td>
</tr>
<tr>
<td>vpc:ReleaseEipAddress</td>
<td>Release EIP addresses.</td>
</tr>
<tr>
<td>vpc:CreateRouteEntry</td>
<td>Create router interfaces.</td>
</tr>
<tr>
<td>vpc:DeleteRouteEntry</td>
<td>Delete router interfaces.</td>
</tr>
</tbody>
</table>

- **SLB-related permissions**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slb:Describe*</td>
<td>Query information related to Server Load Balancer.</td>
</tr>
<tr>
<td>slb:CreateLoadBalancer</td>
<td>Create Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:DeleteLoadBalancer</td>
<td>Delete Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:RemoveBackendServers</td>
<td>Unbind Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:StartLoadBalancerListener</td>
<td>Start specified listeners.</td>
</tr>
<tr>
<td>slb:StopLoadBalancerListener</td>
<td>Stop specified listeners.</td>
</tr>
<tr>
<td>slb:CreateLoadBalancerTCPLoader</td>
<td>Create TCP-based listening rules for Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:AddBackendServers</td>
<td>Add backend servers.</td>
</tr>
</tbody>
</table>

- **AliyunCSClusterRole permissions**

  The default role AliyunCSClusterRole contains the following main permissions:

  - **OSS-related permissions**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oss:PutObject</td>
<td>Upload files or folders.</td>
</tr>
<tr>
<td>oss:GetObject</td>
<td>Retrieve files or folders.</td>
</tr>
<tr>
<td>oss:ListObjects</td>
<td>Query file list information.</td>
</tr>
</tbody>
</table>

  - **NAS-related permissions**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nas:Describe*</td>
<td>Return NAS-related information.</td>
</tr>
<tr>
<td>nas:CreateAccessRule</td>
<td>Create permission rules.</td>
</tr>
</tbody>
</table>
### SLB-related permissions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slb:Describe*</td>
<td>Query information related to Server Load Balancer.</td>
</tr>
<tr>
<td>slb:CreateLoadBalancer</td>
<td>Create Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:DeleteLoadBalancer</td>
<td>Delete Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:RemoveBackendServers</td>
<td>Unbind Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:StartLoadBalancerListener</td>
<td>Start specified listeners.</td>
</tr>
<tr>
<td>slb:StopLoadBalancerListener</td>
<td>Stop specified listeners.</td>
</tr>
<tr>
<td>slb:CreateLoadBalancerTCPListener</td>
<td>Create TCP-based listening rules for Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:AddBackendServers</td>
<td>Add backend servers.</td>
</tr>
<tr>
<td>slb:DeleteLoadBalancerListener</td>
<td>Delete listening rules of Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:CreateVServerGroup</td>
<td>Create VServer groups and add backend servers.</td>
</tr>
<tr>
<td>slb:ModifyVServerGroupBackendServers</td>
<td>Change backend servers in VServer groups.</td>
</tr>
<tr>
<td>slb:CreateLoadBalancerHTTPListener</td>
<td>Create HTTP-based listeners for Server Load Balancer instances.</td>
</tr>
<tr>
<td>slb:SetBackendServers</td>
<td>Configure backend servers and set the weight for a group of ECS instances at the Server Load Balancer instance backend.</td>
</tr>
<tr>
<td>slb:AddTags</td>
<td>Add tags for Server Load Balancer instances.</td>
</tr>
</tbody>
</table>

### 1.2 Upgrade sub-account policy

Container Service comprehensively upgrades the security authorization management on January 15 2018, and provides cross-service authorization based on STS to provide you with more secure services. If you have used Container Service before 15 January 2018, the system completes the authorization by default. For more information about the granted permissions, see [Role authorization](#) If you
used Container Service with a sub-account before, grant the sub-account the permissions to use Container Service again.

Container Service can automatically upgrade the sub-account policy. With this feature, Container Service automatically grants your sub-accounts the AliyunCSRe adOnlyAccess permission. You can also select to manually grant permissions to your sub-accounts in the Resource Access Management (RAM) console.

Upgrade sub-account policy

1. Use the primary account to log on to the Container Service console.
2. Click Upgrade sub account policy in the upper-right corner on the Overview page.
3. Click OK in the displayed dialog box.

Container Service will grant your sub-accounts the corresponding roles when the sub-account policy is being upgraded.

If the upgrade fails, a dialog box listing the sub-accounts that fail to be upgraded appears.

Click Upgrade sub account policy to try to upgrade again or go to the RAM console to manually grant permissions to sub-accounts.

Grant permissions to sub-accounts in RAM console

1. Use the primary account to log on to the Container Service console.
2. Click Users in the left-side navigation pane.
3. Click Authorize at the right of the sub-account.
4. Select the authorization policy and click 1 to add the policy to the Selected Authorization Policy Name. Click OK.

Container Service provides two system authorization policies:

- AliyunCSFullAccess: Provides full access to Container Service.
- AliyunCSReadOnlyAccess: Provides read-only access to Container Service.

You can also create custom authorization policies as per your needs and grant the policies to the sub-accounts. For more information, see Create custom authorization policies.

1.3 Create custom authorization policies

The authorization granularity of the system authorization policies provided by Container Service is coarse. If these authorization policies with coarse granularity cannot satisfy your requirements, create the custom authorization policies. For example, to control the permissions to a specific cluster, you must use the custom authorization policy to meet the requirements with fine granularity.
Create custom authorization policies

Get to know the basic structure and syntax of the authorization policy language before creating custom authorization policies. For more information, see [Authorizaton policy language descriptions](#).

This document introduces how to grant Resource Access Management (RAM) users permissions to query, expand, and delete clusters.

**Procedure**

1. Log on to the **RAM console** with the primary account.
2. Click Policies in the left-side navigation pane. Click Create Authorization Policy in the upper-right corner.
3. Select a template. Enter the authorization policy name and the policy content.

![Create Authorization Policy](image)

"Statement": [
  "Action": [
    "cs:Get",
    "cs:ScaleCluster",
    "cs:DeleteCluster"
  ],
  "Effect": "Allow",
  "Resource": [
    "acs:cs:*cluster/ch2f6e000000000000"
  ],
  "Version": "1"
]
Wherein:

- **Action**: Enter the permission that you want to grant.

- **Resource** supports the following configuration methods.

  - **Grant permissions of a single cluster**

    "Resource": [
    "acs:cs:*:*:cluster/cluster ID"
    ]

  - **Grant permissions of multiple clusters**

    "Resource": [
    "acs:cs:*:*:cluster/cluster ID",
    "acs:cs:*:*:cluster/cluster ID"
    ]

  - **Grant permissions of all your clusters**

    "Resource": [

    You must replace `cluster ID` with your actual cluster ID.

4. Click Create Authorization Policy after completing the configurations.

Table 1-1: Container Service RAM action

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateCluster</td>
<td>Create clusters.</td>
</tr>
<tr>
<td>AttachInstances</td>
<td>Add existing Elastic Compute Service (ECS) instances to clusters.</td>
</tr>
<tr>
<td>ScaleCluster</td>
<td>Expand clusters.</td>
</tr>
<tr>
<td>GetClusters</td>
<td>View cluster list.</td>
</tr>
<tr>
<td>GetClusterById</td>
<td>View cluster details.</td>
</tr>
<tr>
<td>ModifyClusterName</td>
<td>Modify cluster names.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>DeleteCluster</td>
<td>Delete clusters.</td>
</tr>
<tr>
<td>UpgradeClusterAgent</td>
<td>Upgrade cluster Agent.</td>
</tr>
<tr>
<td>GetClusterLogs</td>
<td>View cluster operation logs.</td>
</tr>
<tr>
<td>GetClusterEndpoint</td>
<td>View cluster access point.</td>
</tr>
<tr>
<td>GetClusterCerts</td>
<td>Download cluster certificate.</td>
</tr>
<tr>
<td>RevokeClusterCerts</td>
<td>Revoke cluster certificate.</td>
</tr>
<tr>
<td>BindSLB</td>
<td>Bind Server Load Balancer instances to clusters.</td>
</tr>
<tr>
<td>UnBindSLB</td>
<td>Unbind Server Load Balancer instances from clusters.</td>
</tr>
<tr>
<td>ReBindSecurityGroup</td>
<td>Rebind security groups to clusters.</td>
</tr>
<tr>
<td>CheckSecurityGroup</td>
<td>Check existing security group rules of clusters.</td>
</tr>
<tr>
<td>FixSecurityGroup</td>
<td>Fix cluster security group rules.</td>
</tr>
<tr>
<td>ResetClusterNode</td>
<td>Reset cluster nodes.</td>
</tr>
<tr>
<td>DeleteClusterNode</td>
<td>Delete cluster nodes.</td>
</tr>
<tr>
<td>CreateAutoScale</td>
<td>Create node auto scaling rules.</td>
</tr>
<tr>
<td>UpdateAutoScale</td>
<td>Update node auto scaling rules.</td>
</tr>
<tr>
<td>DeleteAutoScale</td>
<td>Delete node auto scaling rules.</td>
</tr>
<tr>
<td>GetClusterProjects</td>
<td>View applications in clusters.</td>
</tr>
<tr>
<td>CreateTriggerHook</td>
<td>Create triggers for applications.</td>
</tr>
<tr>
<td>GetTriggerHook</td>
<td>View application trigger list.</td>
</tr>
<tr>
<td>RevokeTriggerHook</td>
<td>Delete application triggers.</td>
</tr>
<tr>
<td>CreateClusterToken</td>
<td>Create tokens.</td>
</tr>
</tbody>
</table>
2 Clusters

2.1 Cluster introduction

A cluster is a collection of cloud resources that are required to run containers. It is associated with several Elastic Compute Service (ECS) nodes, Server Load Balancer, and other cloud resources.

Create a cluster

You can create a cluster by using the following methods:

Method 1: Create a cluster and several ECS instances.

You can directly create a cluster with several new ECS instances by using Container Service.

For more information, see Create a cluster.

The ECS instances created using this method are all Pay-As-You-Go instances. If you want to use monthly or yearly subscription ECS instances, buy them separately and then follow Method 2.

Method 2: Create a zero-node cluster and add existing ECS instances to the cluster.

1. Create a zero-node cluster.

   If you have purchased several ECS instances from the ECS service, create a zero-node cluster in Container Service. Method 1 except that you need to select Do not Add when creating the cluster to add existing ECS instances instead of creating some new ones.

   The operations are the same as Method 1 except that you need to select Do not Add when creating the cluster to add existing ECS instances instead of creating some new ones.
2. Add existing ECS instances.

You can add an existing ECS instance to Container Service in the following ways:

- Reset the image of the ECS instance and add the ECS instance to the cluster automatically.

  As this method will reset the image and system disk of the ECS instance, proceed with caution. However, ECS instances added by using this method are cleaner.

- Run scripts on the ECS instance and manually add the ECS instance to the cluster.

  This method is applicable to images that do not require a reset of the ECS instance.

Add an existing ECS instance.

Manage a cluster

You can search for, expand, connect to, clean up, or delete a cluster. For more information, see the following documents:

- Search for a cluster
- Expand a cluster
- Download cluster certificate
- Clean up a cluster disk
- Delete a cluster

2.2 Cluster lifecycle

Table 2-1: A complete cluster lifecycle includes the following statuses.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inactive</td>
<td>The successfully created cluster does not contain any node.</td>
</tr>
<tr>
<td>initial</td>
<td>The cluster is applying for corresponding cloud resources.</td>
</tr>
<tr>
<td>running</td>
<td>The cluster successfully applied for the cloud resources.</td>
</tr>
<tr>
<td>updating</td>
<td>The cluster is upgrading the Agent.</td>
</tr>
</tbody>
</table>
### Status Flow

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scaling</td>
<td>Change the number of cluster nodes.</td>
</tr>
<tr>
<td>failed</td>
<td>The cluster application for cloud resources failed.</td>
</tr>
<tr>
<td>deleting</td>
<td>The cluster is being deleted.</td>
</tr>
<tr>
<td>delete_failed</td>
<td>The cluster failed to be deleted.</td>
</tr>
<tr>
<td>deleted (invisible to users)</td>
<td>The cluster is successfully deleted.</td>
</tr>
</tbody>
</table>

#### Figure 2-1: Cluster status flow

2.3 Create a cluster

You can specify the configurations and the number of Elastic Compute Service (ECS) instances when creating clusters. You can also create a zero-node cluster, and then bind it with other ECS instances.

Note:
The zero-node cluster is in the Inactive status after the creation and is activated with the Running status after you add ECS instances to it. For how to add existing ECS instances to the cluster, see Add an existing ECS instance.

Instructions

Container Service performs the following operations when creating a cluster:

- Create a Server Load Balancer instance with 80:9080 configured as the listener if the Automatically Create Server Load Balancer check box is selected.
- Create a security group. The security group rules are as follows.

Virtual Private Cloud (VPC) inbound

- Create a Resource Access Management (RAM) user if you have activated the RAM service.
- Create the ECS instances and distribute the Internet IP address to the ECS instances if you select Add in the Add Node field. (If the Network Type is VPC, distribute the Elastic IP (EIP) to the ECS instances and create the corresponding routing rules.)
- Use the configured Logon Password to configure the ECS instances.

Note:

Container Service does not save this password.

- If the VPC node configuration fails, Container Service collects the standard output of the node creation and initialization. You can view the information in the cluster logs.

Limits

- Server Load Balancer instances created with clusters are only available in Pay-As-You-Go mode.
- By default, each account has a certain quota for the cloud resources they can create. The cluster fails to be created if the quota is exceeded. Make sure you
have enough quota before creating the cluster. To increase your quota, open a ticket.

- By default, each account can create at most five clusters in all regions and add up to 20 nodes to each cluster.
- By default, each account can create at most 100 security groups.
- By default, each account can create at most 60 Pay-As-You-Go Server Load Balancer instances.
- By default, each account can create at most 20 EIPs.

Procedure

1. Log on to the Container Service console.

2. Click Swarm > Clusters in the left-side navigation pane. Click Create Cluster in the upper-right corner.

3. Complete the following configurations.

   • Cluster Name: Enter the name of the cluster. It can be 1‒63 characters long and contain numbers, Chinese characters, English letters, and hyphens (-).

   Note:
   The cluster name must be unique under the same account and the same region.

   • Region: Select the region in which the cluster is to be deployed.

   • Zone: Select the zone for the cluster.

   Note:
You can select the region and zone according to the distribution of your servers.

Cluster Name: [Input Field]
The cluster name should be 1-63 characters long, and can contain numbers, Chinese characters, English letters and hyphens.

Region:
- China North 1 (Qingdao)
- China North 2 (Beijing)
- China East 1 (Hangzhou)
- China East 2 (Shanghai)
- China South 1 (Shenzhen)
- Asia Pacific NT 1 (Tokyo)
- US West 1 (Silicon Valley)
- Asia Pacific SE 1 (Singapore)
- Asia Pacific SE 2 (Sydney)
- Asia Pacific SE 3 (Kuala Lumpur)
- EU Central 1 (Frankfurt)
- US East 1 (Virginia)
- Hong Kong
- China North 3

Zone: [Selector] East China 1 Zone G

4. Select the network type of the cluster. Currently, Container Service only supports VPC.

Complete the corresponding configurations.

Network Type: [Dropdown] VPC

Initial CIDR Block: 172.20.0.0/16

VPC enables you to build an isolated network environment based on Alibaba Cloud. You can have full control over your own virtual network, including a free IP address range, Classless Inter-Domain Routing (CIDR) block division, and the configurations of route table and gateway.

Specify a VPC, a VSwitchId, and the initial CIDR block of containers (the subnet CIDR block to which the Docker containers belong. For ease of IP management, containers of different virtual machines belong to different CIDR blocks, and container subnet CIDR block cannot conflict with virtual machine CIDR block). We recommend that you build your own VPC/VSwitchld for the cluster to prevent issues such as network conflicts.
5. Select whether or not to add nodes.

You can create a cluster with several new ECS instances, or create a zero-node cluster and then add existing ECS instances to the cluster. For how to add existing ECS instances to the cluster, see Add an existing ECS instance.

- Add
  
  a. Select the operating system for the node.

  ![Operating System Selection](image)

  Currently, Ubuntu 14.04/16.04 64bit and CentOS 7.4 64bit are supported.

  b. Configure the ECS instance specifications.

  ![Instance Specification Configuration](image)

  You can select the generation, family, type, and quantity of the instance, disk type and capacity (the ECS instance has a 20 GB system disk by default), and logon password. Container Service uses the configured Logon Password to configure the ECS instances when creating the cluster, but does not save this password.

  ![Note](image)
- The data disk is mounted to the \texttt{/var/lib/docker} directory and used for the storage of Docker images and containers if you select the Attach Data Disk check box.
- In terms of performance and management, we recommend that you mount an independent data disk to the host and manage the persistent data of containers by using Docker volumes.

- Do not Add

You can click Add Existing Instance to add existing ECS instances to the cluster, or click Add Existing Instances on the Cluster List page to add existing ECS instances to the cluster after the cluster is created. For more information, see \textit{Add an existing ECS instance}.

6. Select whether or not to configure public EIP.

If you select VPC as the network type, Container Service configures an EIP for each ECS instance in the VPC environment by default. If this is not required, select the Do not Configure Public EIP check box and then configure the SNAT gateway.

\textbf{Note:}

You can apply for up to 20 EIPs per account. To use VPC and create EIP automatically when creating a cluster, the cluster fails to be created if the number of EIPs under your account reaches its quota.

7. Select whether or not to create a Server Load Balancer instance.

The \textbf{Automatically Create Server Load Balancer} check box is selected by default. With this check box selected, a Server Load Balancer instance is created after the cluster is created. You can access the container applications in the cluster by
means of this Server Load Balancer instance. The created Server Load Balancer instance is in the Pay-As-You-Go mode.

8. Select whether or not to install cloud monitoring plug-in on your ECS instances.

To view the monitoring information of the created ECS instances in the CloudMonitor console, select the Install cloud monitoring plug-in on your ECS check box.

9. You can select to add the IP addresses of the ECS instances to the RDS instance whitelist.

Adding the IP addresses of the ECS instances to the RDS instance whitelist facilitates the ECS instances to access the RDS instances.

Note:

- We recommend that you configure the RDS Whitelist when Add is selected for Add Node.
- If Do not Add is selected for Add Node and you want to configure the RDS Whitelist, add the existing ECS instances on the Create Cluster page. The RDS Whitelist cannot be configured if you create a zero-node cluster and add existing ECS instances after the cluster creation.
The ECS instance must be in the same region as the RDS instance so that the IP address of the ECS instance can be added to the RDS instance whitelist.

a. Click Select RDS Instances. The Add to RDS instance whitelist dialog box appears.

b. Select the RDS instances and then click OK.

10. Click Create Cluster.

After the cluster is successfully created, you can configure the ECS instance or Server Load Balancer instance in the corresponding console.

Subsequent operations

On the Cluster List page, you can click View Logs at the right of the cluster to view the creation process logs of the cluster.

You can create applications in the created cluster. For more information, see Create an application.

References

If the cluster fails to be created, see #unique_19 for troubleshooting.
2.4 Cluster parameter configurations

This document aims to help you understand what the parameters on the page mean when you create a cluster. Then, you can configure the parameters smoothly. For some parameters, some documents are provided for your reference.

Cluster Name

Configure the cluster name.

- The name can be 1–63 characters long and contain numbers, Chinese characters, English letters, and hyphens (-), but cannot start with a hyphen (-).
- You can modify the cluster name on the Cluster List page after creating the cluster.

Region and Zone

Container Service authorizes to create the region and zone of the Elastic Compute Service (ECS) instances. Currently, the regions and zones supported by Container Service belong to the subset of ECS product. For more information, see Regions and zones.

Network Type

Select VPC as the network type of the ECS instances. Alibaba Cloud Virtual Private Cloud (VPC) allows you to create a custom VPC. Layer-2 logical isolation exists between different VPCs. You can plan the Classless Inter-Domain Routing (CIDR) block of each cluster flexibly. VPC is applicable to a scenario with large-scale container clusters and provides higher security and flexibility. To better guarantee the system security and the support of hybrid cloud business, Container Service does not support creating clusters whose network type is classic network or with non-I/O optimized instance since January 1, 2018.

Initial CIDR Block of Container Service

Configure this parameter only when you select VPC. When planning the CIDR block, make sure the container initial CIDR block does not overlap with the VPC CIDR block.

- You can only specify one CIDR block for each VPC. 172.16.0.0/12 is the default VPC CIDR block.
Specify the corresponding container CIRD block when creating a Container Service cluster. Currently, Container Service supports the following container CIDR blocks: 192.168.1.0/24 and 172.16-31.0/24

Add Node

Container Service has two ways to add nodes: create nodes and add existing nodes. If you select Add, Container Service is authorized to automatically create ECS instances when the cluster is created and automatically add the created ECS instances to the created cluster. If you select Do not Add, the existing ECS instances are added to the cluster. You can add the existing ECS instances on the Create Cluster page directly or create a zero-node cluster and then add the existing ECS instances on the Cluster List page. For more information, see Add an existing ECS instance.

Node Type

The node type is Pay-As-You-Go by default. After creating the ECS instances, you can go to the ECS console to change the Pay-As-You-Go ECS instances to monthly or yearly subscription ECS instances.

Operating System

Select the operating system installed in the ECS instances. We recommend that you use Ubuntu 14.04 64 bit and CentOS 7.4 64 bit.

Instance Generation and Instance Family

Different instance generations correspond to different instance families. ECS instances provide you with corresponding computing capabilities based on the instance specifications. ECS instances can be divided into many generations and families according to the business scenarios and usage scenarios. For the specific scenarios for each instance generation and family, see t9548.dita#concept_sx4_lxv_tdb.

Instance Type

ECS instance type defines two basic attributes: the CPU configuration and memory configuration of the instance. However, ECS instances can determine the specific service pattern of an instance only by working together with the disk, image, and network type.
Instance Quantity

The number of the ECS instances to be created. The number of ECS instances in one cluster cannot exceed 20. To enhance the cluster availability, we do not recommend that you create a cluster with one node. 2 sets is the default value in the console.

System Disk Type

Select the cloud disk type of the installation system. Select Ultra Cloud Disk or SSD Cloud Disk according to your requirements on the system performance of the ECS instances. For the performance indicator comparison between these two types of cloud disks, see t9557.dita#concept_ytm_vwj_ydb.

Data disk configurations

Select the type of the data disk that is to be mounted to the container. Select the Attach Data Disk check box and select the data disk capacity. The data disk is mounted to the /var/lib/docker directory of the container to store the image data and container data.

Logon Password and Confirm Password

Enter and confirm the logon password of the ECS instances. The password is 8‒30 characters long and must contain uppercase letters/lowercase letters, numbers, and special characters at the same time. This password is required when you log on to the ECS console or log on to the ECS instance by using SSH.

Note:

- Container Service uses this password to configure the ECS instances when creating the cluster, but does not save this password.
- Keep this password properly for the initialization usage.

EIP

The Elastic IP (EIP) is used to access the Internet. By default, Container Service retains the EIP. If you select to not retain the EIP, the cluster releases the EIP after the instance initialization. You can access the Internet by using the t13979.dita#concept_wpm_kfy_ydb or configuring SNAT for Linux on your own.
Server Load Balancer

An Internet Server Load Balancer instance is created by default if a cluster is created. The billing method is Pay-As-You-Go. The created Server Load Balancer instance is used to distribute the traffic to control the services and implement the service high availability.

Monitoring Plug-in

Select the check box to install the cloud monitoring plug-in on the ECS instances. Then, the operating system-level performance indicators of the ECS instances in the cluster can be monitored.

RDS Whitelist

You can select to add the IP addresses of the created nodes to the RDS instance whitelist, which facilitates the ECS instances to access the RDS instances.

- We recommend that you configure the RDS Whitelist when Add is selected for Add Node.
- If Do not Add is selected for Add Node and you want to configure the RDS Whitelist, add the existing ECS instances on the Create Cluster page. The RDS Whitelist cannot be configured if you create a zero-node cluster and add existing ECS instances after the cluster creation.
- The ECS instance must be in the same region as the RDS instance so that the IP address of the ECS instance can be added to the RDS instance whitelist.

Security Group

Container Service configures the default security group and only sets the inbound security group rules. You can configure the security group according to your business scenarios after the cluster is created successfully. For more information, see #unique_21

- Ports 443 and 80 can be opened or closed as per your needs.
- We recommend that you retain the ICMP rules for communication between nodes and the convenience of troubleshooting. Some tools also depend on ICMP.
- Make sure you open all the ports you need. Otherwise, some services become inaccessible. The port that is accessed by using Server Load Balancer is not required to be opened.
2.5 Add an existing ECS instance

You can add a purchased Elastic Compute Service (ECS) instance to a specified cluster.

Note:
At most 20 ECS instances can be added to a cluster by default. To add more ECS instances, open a ticket.

You can add an existing ECS instance in the following ways:

- Add ECS instances automatically: The image and system disk of the ECS instance are reset by using this method. You can add one or more ECS instances to the cluster at a time.
- Add the ECS instance manually: Manually add the ECS instance by running scripts on the ECS instance. You can only add one ECS instance to the cluster at a time.

Prerequisites

If you have not created a cluster before, create a cluster first. For information about how to create a cluster, see Create a cluster.

Instructions

- The ECS instance to be added must be in the same region and use the same network type (Virtual Private Cloud (VPC)) as the cluster.
- When adding an existing ECS instance, make sure that your ECS instance has an Elastic IP (EIP) for the network type VPC, or the corresponding VPC has configured the NAT gateway. In short, make sure the corresponding node can access public network normally. Otherwise, the ECS instance fails to be added.
- The ECS instance to be added must be under the same account as the cluster.
- If you select to manually add the ECS instance, note that:
  - If you have already installed Docker on your ECS instance, the ECS instance may fail to be added. We recommend that you uninstall Docker and remove
the Docker folders before adding the ECS instance by running the following command:

**Ubuntu:**
```
apt-get remove -y docker-engine, rm -fr /etc/docker/ /var/lib/docker /etc/default/docker
```

**CentOS:**
yum remove -y docker-engine, rm -fr /etc/docker /var/lib/docker

- Container Service nodes have special requirements for the operating system of the ECS instance. We recommend that you use Ubuntu 14.04/16.04 or CentOS 7 as the operating system. We have strictly tested the stability and compatibility of these operating systems.

**Procedure**

1. Log on to the **Container Service console**.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click More at the right of the cluster that you want to add ECS instances and then select Add Existing Instances from the drop-down list.
4. Add ECS instances.

   The ECS instances displayed are filtered and synchronized from your ECS instance list according to the region and network type defined by the cluster.

**Add the ECS instances in the following ways:**

- Add ECS instances automatically.

**Note:**
As this method will reset the image and system disk of the ECS instance, proceed with caution. Create a snapshot to back up your data before adding...
the ECS instance. For information about how to create a snapshot, see #unique_22.

a. Select the ECS instances you want to add to the cluster and click Next Step.
   You can add one or more ECS instances at a time.

b. Configure the instance information. Click Next Step and then click Confirm in the confirmation dialog box.

c. Click Finish.

- Manually add the ECS instance by running scripts on the ECS instance.
  a. Select Manually Add. Select an ECS instance, and then click Next Step.
   You can only add one ECS instance at a time.

  b. Confirm the instance information and click Next Step.

  c. The scripts unique to this ECS instance are displayed. Click log on to the ECS instance xxxxxx.

  d. The VNC connection password is displayed in the dialog box. Copy the password and click Close.

  e. In the dialog box, enter the VNC connection password and click OK.
f. Enter the logon account (root) and password of the ECS instance, and press Enter to log on to the ECS instance.

![Image of ECS instance connection]

g. Click Input Commands. Paste the preceding scripts into the dialog box, click OK and press Enter.

```
curl -Ls http://aliyuncontainerservice.oss-cn-hangzhou.aliyuncs.com/17.03.1-ce/attachNodeScript | sudo bash -s -- instance-id i-12345678
```

The system runs the scripts. Wait until the scripts are successfully run. A success message is displayed. The ECS instance is successfully added.
You can modify the VNC connection password of the ECS instance in the remote terminal connection page. Click Modify Management Terminal Password, enter the new password and click OK in the dialog box.

Note: The modified VNC password will not take effect until the instance is restarted at the console.

2.6 Manage cross-zone nodes

To enhance the high availability of applications, you can distribute multiple nodes in different zones when creating a cluster.
You can create a cluster with one node or a zero-node cluster. Then, add nodes of different zones by expanding the cluster or adding existing Elastic Compute Service (ECS) instances.

**Note:**
- Nodes added by expanding the cluster are Pay-As-You-Go ECS instances.
- Nodes added by adding existing ECS instances can be Pay-As-You-Go ECS instances or monthly/yearly subscription ECS instances.

Add nodes of different zones by expanding the cluster

**Procedure**

1. Log on to the Container Service console.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click More at the right of the cluster that you want to expand and then select Expand. As shown in the following figure.

4. The Expand page appears. Configure the specifications of the new nodes.

You can create nodes of different zones by setting Zone.

5. Click Expand to add the new nodes to the cluster.
6. Repeat the preceding steps to create and add nodes of different zones to the cluster.

Add nodes of different zones by adding existing ECS instances

Prerequisites

To add nodes by using this method, purchase ECS instances from the ECS purchase page first, and select different zones for them during the purchase.

Procedure

1. Log on to the Container Service console.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click More at the right of the cluster that you want to add existing ECS instances and then select > Add Existing Instances. As shown in the following figure.

4. Select ECS instances of different zones and add them manually or automatically to the cluster.

For more information, see Add an existing ECS instance.

5. Repeat the preceding steps to add ECS instances of different zones to the cluster.
2.7 Bind and unbind a Server Load Balancer instance

You can automatically create a Pay-As-You-Go Server Load Balancer instance when creating a cluster, or bind a monthly/yearly subscription or Pay-As-You-Go Server Load Balancer instance to a cluster after creating the cluster.

Container Service supports binding an Internet Server Load Balancer instance, a VPC Server Load Balancer instance, or an intranet Server Load Balancer instance in a classic network to a cluster.

Limits

- You can only bind a Server Load Balancer instance to a cluster of the same region.
- You can only bind a Server Load Balancer instance to a cluster created by the same account.
- A VPC cluster can bind an Internet Server Load Balancer instance or a VPC Server Load Balancer instance.
- One cluster can only bind one Server Load Balancer instance.
- Two clusters cannot share one Server Load Balancer instance.

Prerequisites

You have created a Server Load Balancer instance in the Server Load Balancer console and configured the TCP 9080 port for the instance to listen to backend servers.

For how to create a Server Load Balancer instance, see Create a Server Load Balancer instance.

Bind a Server Load Balancer instance

1. Log on to the Container Service console.

2. Click Manage at the right of the cluster that you want to bind a Server Load Balancer instance. The cluster details page appears.
3. Click Load Balancer Settings in the left-side navigation pane > and then click Bind Server Load Balancer.

4. Select the Server Load Balancer instance that you want to bind to the cluster from the Server Load Balancer ID list and click OK.

Note:
If the selected Server Load Balancer instance has been bound to a backend server, the system will prompt you that “This Server Load Balancer instance is already bound to a backend server”. You need to select another Server Load Balancer instance that has not been bound to any backend server.

Rebind a Server Load Balancer instance

You can change the Server Load Balancer instance bound to your cluster per your needs.

1. Log on to the Container Service console.
2. Click Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster that you want to re-bind a Server Load Balancer instance. The cluster details page appears.
4. Click Load Balancer Settings in the left-side navigation pane, > and click Re-bind Server Load Balancer.

![Image of Load Balancer Settings page]

5. Select the Server Load Balancer instance that you want to bind to the cluster from the Server Load Balancer ID list and click OK.

Unbind a Server Load Balancer instance

You can unbind a Server Load Balancer instance in the Container Service console if the instance is not required.

1. Log on to the Container Service console.
2. Click Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster that you want to unbind a Server Load Balancer instance. The cluster details page appears.

![Image of Cluster Details page]

4. Click Load Balancer Settings in the left-side navigation pane, > and click Unbind Server Load Balancer.

![Image of Unbind Load Balancer page]

2.8 Set the root domain name of a cluster

Context
When you configure the web routing rules, you are only required to enter the domain name prefix nginx. Then, you can obtain the domain name in the format of $cluster_id.$region_id.alicontainer.com. You can replace this domain name by setting a root domain name (51ili.com is used in this example) of the cluster. When you redeploy the application nginx, the domain name changes from nginx.cd5b226071936493b89e75bbe8841664c.cn-hangzhou.alicontainer.com to nginx.51ili.com, which makes it convenient for you to access the cluster applications with your own root domain name.

**Note:**
To guarantee the normal operation of the following example, upgrade the Agent to the latest version first.

**Procedure**

1. **Bind a Server Load Balancer instance.**
   a) Log on to the **Container Service console**.
   b) Log on to the **Container Service console**.
   c) Click Clusters in the left-side navigation pane.
   d) Click Manage at the right of the cluster (**routing-test-online** in this example) that you want to configure.

![Container Service Console](image)

   e) Click Load Balancer Settings in the left-side navigation pane.

   If no Server Load Balancer instance is bound to this cluster, log on to the **Server Load Balancer console** and create a Server Load Balancer instance. Then, return to this page and bind the instance to this cluster.

**Note:**
For more information about how to bind and unbind a Server Load Balancer instance to and from a cluster and the limits in Container Service, see *Bind and unbind a Server Load Balancer instance*.

2. **Set the domain name.**

   a) Click the Set Domain Name tab and enter the root domain name you bought in the Domain Name field. In this example, 51ili.com is entered.

   ![Set Domain Name Example](image)

   b) Click Set.
3. Resolve the domain name to the bound Server Load Balancer instance.
   a) Log on to the Server Load Balancer console. Click Instances in the left-side navigation pane, and then click the ID of the Server Load Balancer instance bound to the cluster routing-test-online.
   b) View the instance details. Find the instance IP address.
   c) Log on to the Alibaba Cloud DNS console and add record A to resolve *.51ili.com to the Server Load Balancer VIP address.

4. Redeploy the nginx application.
   a) Click Redeploy at the right of nginx. The service access endpoint of the application nginx is changed.
      The access endpoint before setting the root domain name.

      The access endpoint after setting the root domain name.

   b) Access the latest access endpoint http://nginx.51ili.com.

2.9 Download cluster certificate

Context

With the downloaded certificate, you can connect to the endpoint exposed from the cluster by using Docker Swarm API or Docker client. For more information, see #unique_27.
Procedure

1. Obtain the access address.
   a) Log on to the Container Service console.
   b) Log on to the Container Service console.
   c) Click Clusters in the left-side navigation pane. On the Cluster List page, click Manage at the right of a cluster.
   d) The cluster details page is displayed, showing the cluster connection information.

2. Download and save the TLS certificate.

   Configure a TLS certificate before you use the preceding access address to access the Docker cluster.

   Click Download Certificate in the cluster details page to download the TLS certificate. The certFiles.zip file is downloaded. The certificate. The certFiles.zip file is downloaded. In the following example, the downloaded certificate is saved to the ~/.acs/certs/ClusterName/ directory. ClusterName indicates the name of your cluster. You can save the certificate to a different directory, but we recommend using the ~/.acs/certs/ClusterName/ directory for easy management.

   mkdir ~/.acs/certs/ClusterName/ #Replace ClusterName with your cluster name
   cd ~/.acs/certs/ClusterName/
   cp /path/to/certFiles.zip .
unzip certFiles.zip

The `certFiles.zip` file contains `ca.pem`, `cert.pem`, and `key.pem`.

2.10 Expand a cluster

Prerequisites

A cluster can contain up to 20 nodes.

Context

You can expand your cluster according to your business needs.

**Note:**

Elastic Compute Service (ECS) instances added by expanding the cluster are Pay-As-You-Go instances.

Procedure

1. Log on to the *Container Service console*.

2. Click Clusters in the left-side navigation pane.

3. On the Cluster List page, click More at the right of the cluster that you want to expand and then select Expand from the list.

4. In the displayed dialog box, configure the specifications of the new node.

   You can select the number and the specifications of the ECS instances you are about to add to the cluster.

5. Click Expand.
2.11 Migrate a cluster

For a Swarm cluster created earlier, you can guarantee the performance and stability of the cluster by migrating the cluster.

Context

- The latest time for migrating a cluster is displayed through SMS, station message, or email. Complete the Swarm cluster migration before the latest time. The system automatically migrates the cluster if you do not migrate the cluster before the latest time.
- Cluster migration rebuilds connections from cluster nodes to the container server without affecting applications deployed in the cluster, nor adding or modifying any data. Make sure that you perform this operation during the low peak period of your business because unpredictable risks might still exist throughout the migration process.

Procedure

1. Log on to the Container Service console.
2. Under the Swarm menu, click Clusters.
3. Click Cluster Migration in the action column at the right of the cluster to be migrated.
4. Click OK in the Prompt dialog box.

Note:
During cluster migration:

- Information query, deployment, upgrade, and other operations cannot be performed in the console.
- The cluster cannot be connected to through the cluster access point API.
- The data and application status in the cluster remain unchanged. Applications deployed on the cluster are still accessible.
The migration process takes about three minutes.

On the Cluster List page, Migrating is displayed in the Cluster Status column.

After cluster migration is completed, on the Cluster List page, Running is displayed in the Cluster Status column.

Note:
- The cluster ID, access point address, and other attributes remain unchanged.
- Please be sure to confirm that your business is running properly.
- During the migration process, if you have any questions, please open a ticket in which you include the cluster ID and state whether your deployed applications are normal.

2.12 Search for a cluster

Procedure
1. Log on to the Container Service console.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Enter the cluster name or keywords of the cluster name in the search box. Clusters with the keywords in their names are displayed. As shown in the following figure.

Note:
2.13 Delete a cluster

Context

You can delete clusters from Container Service. Deleting the cluster also deletes its associated Elastic Compute Service (ECS) instances, Server Load Balancer instance, and other cloud resources, so proceed with caution.

Procedure

1. Log on to the Container Service console.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click Delete at the right of the cluster you are about to delete.
4. In the displayed window, select whether or not to keep the Server Load Balancer instance and click OK.

2.14 Clean up a cluster disk

Context

Cleaning up disk clears the dirty data on each server in your cluster. Dirty data is limited to:

- Docker images downloaded locally but not used.
Volume directory once attached to a container but not cleaned up after the destruction of the container.

Procedure

1. Log on to the Container Service console.

2. Click Clusters in the left-side navigation pane.

3. On the Cluster List page, click Manage at the right of the cluster that you want to clean up the disk.

4. Click Clear Disk on the cluster details page.

2.15 Log on to image repository

Prerequisites

- Prepare an available image repository. Use the Docker Hub official service in this example, which requires you to register a Docker ID and build an available repository in it.
- Configure the independent logon password for the repository. In this example, log on to the Container Registry console to configure or modify the repository logon password. Note that you are configuring the password when you modify the repository logon password for the first time.

Context

You can log on to the image repository in a cluster to provide the related cluster logon information, which facilitates you to manage clusters by using cluster management tools.

Procedure
1. Log on to the Container Service console.

2. Click Clusters in the left-side navigation pane.

3. Click Manage at the right of the cluster you want to configure.

4. Click Log on to Hub.

5. Configure the parameters in the displayed dialog box.

- **Repository Domain Name**: Enter the hub domain name of the image repository. Take the image address `registry.cn-hangzhou.aliyuncs.com/`
acs/agent:0.8 as an example. The repository domain name is registry.cn-hangzhou.aliyuncs.com.

- Username: Enter the username of the image repository. In this example, enter the Docker ID registered in Docker Hub.
- Password: Enter the independent logon password of the image repository. In this example, enter the logon password set when you registered in Docker Hub. Registry's login password is set and modified on the container mirroring Service's console.
- Email: Enter the email set when you registered the image repository. In this example, enter the email set when you registered in Docker Hub.

6. Click OK. You have successfully logged on to the image repository if no error message appears.

2.16 Upgrade Agent

Context

Note:
Your applications are not affected during the upgrade, but you can neither manage the cluster by using the Web interface, nor use Docker client to connect to the cluster access port for about 2 minutes.

The Agent of Container Service, which is installed on each server in the cluster, receives commands issued by the Container Service control system.

New functions are regularly added to Container Service. If you need the latest functions, upgrade the Agent of the cluster.

Procedure

1. Log on to the Container Service console.

2. Click Clusters in the left-side navigation pane.
3. On the Cluster List page, click More at the right of the cluster that you want to upgrade the Agent and then select Upgrade Agent from the list.

4. Click OK in the displayed dialog box.

2.17 Upgrade Docker daemon

Context

Standard Docker daemon is installed on each server in the cluster to manage containers.

Note:

- The cluster Docker daemon upgrade requires that the machine is able to access the Internet to download necessary software packages.
- The cluster Docker daemon upgrade may fail. To guarantee your data security, we recommend that you back up snapshots before upgrading Docker daemon.
- During the cluster Docker daemon upgrade, the services deployed on the cluster are interrupted and you cannot perform operations on the cluster and applications. Make appropriate arrangements before the upgrade. The upgrade lasts 3‒30 minutes. The cluster status changes to Running after the upgrade.

You can view the Docker version of the cluster on the Cluster List page.

Procedure
1. Log on to the *Container Service* console.

2. Under Swarm, click Clusters in the left-side navigation pane.

3. On the Cluster List page, click Upgrade in the Docker Version column, or click More > Upgrade Docker at the right of the cluster.

4. On the Upgrade Docker page, click Upgrade Agent to upgrade the Agent first if your Agent is not in the latest version.

5. If your Agent is in the latest version, upgrade Docker daemon in the following ways:
   - Upgrade Directly
     
     Click Upgrade Directly to enter the Docker Engine upgrade process.
   - Back up Snapshot before Upgrade
     
     We recommend that you back up the snapshots before upgrading Docker daemon. In this way, you can recover Docker daemon by using the snapshots if an error occurs during the upgrade process.

     Click Back up Snapshot before Upgrade, and then the system calls the Elastic Compute Service (ECS) API to take snapshots of the cluster nodes.

     Backing up snapshots may take some time. Wait until the snapshots are backed up, and then the system automatically enters the Docker Engine upgrade process.

     If the snapshots failed to be backed up, you can click Continue or Quit. Click Continue to enter the Docker Engine upgrade process, or click Quit to give up the upgrade.
Return to the Cluster List page and you can see that the cluster you upgraded the Docker daemon is in the Docker-Engine is upgrading status. This may take a while as container data will be backed up during the upgrade of the Docker Engine.

2.18 Upgrade system services

Context

The system services of a cluster, including Log Service `acslogging`, Simple Routing Service `acsrouting`, Monitor Service `acsmonitoring`, and Volume Service `acsvolumedriver`, are used to deal with general services necessary for applications. This document introduces how to upgrade these system services.

Note:

During the upgrade of the cluster system services, your applications or services may be temporarily inaccessible or abnormal, so proceed with caution. We recommend that you upgrade the system services when the access traffic is low or at the maintenance time.

Procedure

1. Log on to the Container Service console.
2. Under Swarm, click Clusters in the left-side navigation pane.
3. On the Cluster List page, click More at the right of the cluster whose system services you want to upgrade and then select Upgrade System Service from the drop-down list. As shown in the following figure.
4. The Upgrade System Service dialog box opens. Select the system services you want to upgrade and click Upgrade.

For example, select Simple Routing Service (corresponding to acsrouting; note that the upgrade will temporarily affect your access to applications) and Volume Service (corresponding to acsvolumedriver; note that the upgrade might temporarily affect the functions of your associated applications).

Click Applications in the left-side navigation pane and select the cluster from the Cluster drop-down list. You can see the system services are being upgraded.

After the upgrade, the affected services and applications resume normal functioning.
3 Nodes

3.1 View containers running on a node

Context

You can view containers running on a node on the Node List page.

Procedure

1. Log on to the Container Service console.
2. Click Swarm > Nodes in the left-side navigation pane.
3. On the Node List page, select a cluster from the Cluster drop-down list.
4. Click the node ID.

You can see the list of containers running on the node.

What's next

In the list, you can view the labels, images, the image SHA256 values, logs, and monitoring information of containers and perform operations on containers, including starting and stopping containers, deleting containers, and operating on containers on a remote terminal.
3.2 Update a node certificate

You can update a node certificate of a Swarm cluster to avoid node certificate expiration.

Prerequisites

1. You have created a swarm cluster, see `Create a cluster`.
2. Updating a node certificate reboots the node Docker Daemon. Make sure that containers on the node are all configured to restart automatically.

   **Note:**
   You can configure a container restart policy when creating an application. When you create an application by using an image, select the Always check box for Restart. When you create an application by using a template, configure a container restart policy in the template `restart: always`.

3. If a node certificate expires within 60 days, a prompt is displayed. You must timely update the node certificate.

Context

Each cluster node has a certificate used to access system control services. Each issued certificate has a valid period. When the valid period of a certificate is about to expire, you must manually renew the certificate. Otherwise, the service of the node is affected.

Procedure

1. Log on to the `Container Service console`.
2. Under the Swarm menu, click Nodes in the left-side navigation pane. The certificate expiration information of each cluster node is displayed.

   **Note:**
   The certificate expiration time is displayed in the status column only if the node certificate expires within 60 days.

3. Select a node in the node list, and click More > Update Certificate on the right to reissue the node certificate.

   **Note:**
We recommend that you upgrade the cluster agent to the latest version before updating the node certificate.

4. Optional: If the system prompts you to upgrade the cluster agent after you click Update Certificate, the current cluster agent does not support this feature. You need to upgrade the cluster agent to the new version first, see Upgrade Agent. If no prompt is displayed, go to the next step.

5. If no prompt is displayed or the cluster agent is updated, click Update Certificate. Confirm updating information and then update the node cluster certificate.

Note:

- When the node certificate update is completed, the Docker Daemon node is automatically restarted about 1 minute later.
- To guarantee that containers on the node can automatically restart, make sure that an automatic restart policy is configured.

6. After the cluster node certificate is updated, the node certificate information is no longer displayed.
4 Images and templates

4.1 Update an orchestration template

Context

You can only edit orchestration templates displayed under My Orchestrations on the Orchestration List page. To edit templates displayed under Sample, save the sample template as your own template and then edit it.

For how to save an orchestration template as a new one, see #unique_38.

Procedure

1. Log on to the Container Service console.

2. In the left-side navigation pane, click Images and Templates >> Orchestration Templates.

3. Click the My Orchestrations tab and then click Details of the orchestration template you want to update.

4. Click Edit in the upper-right corner.
5. Edit the template content.

To modify a service, you can modify the content in the template directly or click Edit to modify the configurations in the appeared Create Service dialog box.

To add another service to the orchestration template, click Add Service. The Create Service dialog box appears. Select an image and complete the other configurations. Click OK. You can modify the content in the template directly or click Delete to delete the service.

6. Click Save in the upper-right corner to save the modifications.
5 Service orchestrations

5.1 routing

The routing label configures the access domain name of a service.

Format:

```
aliyun.routing.port$_container_port: [http://]$domain|$domain_prefix[:$context_path]
```

Field description:

- $container_port: container port. Note: This is not the host port.
- $domain: domain name. Enter a domain name.
- $domain_prefix: domain name prefix. If you enter a domain name prefix, Container Service provides you with a test domain name and the domain name suffix is `<cluster_id>.<region_id>.alicontainer.com`.
- $context_path: requested service path. You can select services according to the requested path.

Domain name selection:

- If the HTTP protocol is used to expose the service, you can use the internal domain name (the top-level domain is alicontainer.com) provided by Container Service for testing, or use your own domain name.
- If the HTTPS protocol is used, you can use only your own domain name. For example, www.example.com. You must modify the DNS settings to assign the domain name to the Server Load Balancer service provided by the container cluster.

Format requirements of the label statement:

- Container Service allocates a subdomain name to each cluster, and you only need to provide the domain name prefix to bind the internal domain name. The domain name prefix only indicates a domain name level and cannot be separated with periods (.).
- If you do not specify scheme, the HTTP protocol is used by default.
The length of the domain name cannot exceed 128 characters. The length of the context root cannot exceed 128 characters.

When you bind multiple domain names to the service, use semicolons (;) to separate them.

A backend service can have multiple ports. These ports are exposed by the container. A port can only be assigned one label. Therefore, a service with multiple ports must be assigned multiple labels.

Example:

Use the routing label.

**Bind the internal domain name** `wordpress.<cluster_id>.<region_id>.alicontainer.com` provided by Container Service and your own domain name `http://wp.sample.com/context` to port 80 of the Web service.

```yaml
web:
  image: wordpress:4.2
  links:
    - db:mysql
  labels:
    - aliyun.routing.port_80: wordpress;http://wp.sample.com/context

db:
  image: mysql
  environment:
    - MYSQL_ROOT_PASSWORD=password
```

The internal domain name that you finally get is `wordpress.cd3dfe269056e4543acb5e19b01c074.cn-beijing.alicontainer.com`.

After starting the Web service, you can access the corresponding Web services by using the URL: `http://wordpress.cd3dfe269056e4543acb5e19b01c074.cn-beijing.alicontainer.com` or `http://wp.sample.com/context`.

To support the HTTPS service, upload the HTTPS certificate by using the Server Load Balancer console on the Alibaba Cloud website, and then bind the corresponding cluster to access the Server Load Balancer terminal.

**routing/session_sticky**

By using this feature, you can determine whether to maintain session sticky (session persistence) when you set the routing for a routing request. With session persistence, during the session, each request is routed to the same backend container instead of being randomly routed to different containers.
Note:

- The setting takes effect only when you have configured `aliyun.routing.port_$container_port`.
- Simple routing session persistence is based on the Cookie mechanism. By default, the maximum expiration time of Cookie is 8 hours and the idle expiration time is 30 minutes.
- Simple routing session persistence is enabled by default.

The setting methods are as follows:

- **Enable session persistence**
  
  `aliyun.routing.session_sticky: true`

- **Disable session persistence**
  
  `aliyun.routing.session_sticky: false`

**Example of a template orchestration file:**

```yaml
web:
  image: wordpress:4.2
  links:
    - db:mysql
  labels:
    - `aliyun.routing.port_80: wordpress;http://wp.sample.com/context`
    - `aliyun.routing.session_sticky: true`

db:
  image: mysql
  environment:
    - `MYSQL_ROOT_PASSWORD=password`
```
6 Applications

6.1 Create an application

Context

Limits

Swarm clusters only support the compose V1 and compose V2 orchestration templates. The system reports an error if you select to use the compose V3 template.

Note:
In the orchestration template list, compose V3 templates are marked with composev3.

Procedure

1. Log on to the Container Service console.
2. Click Applications in the left-side navigation pane.
3. Click Create Application in the upper-right corner.
4. Complete the basic information for the application you are about to create.

- **Name**: Enter the name of the application. It can be 1–64 characters long and contain numbers, English letters, and hyphens (-), but cannot start with a hyphen (-).
- **Version**: Enter the version of the application. By default, 1.0 is entered.
- **Cluster**: Select the cluster on which the application is to be deployed.
- **Update**: The update method of the application. Select Standard Release or Blue-Green Release. For more information, see \#unique_42.
- **Description**: Enter the information of the application. This field is optional. The entered description cannot exceed 1024 characters, and is displayed on the Application List page.
- **Pull Docker Image**: With this check box selected, Container Service pulls the latest Docker image from the repository to deploy the application, even when the image tag does not change.

To improve efficiency, Container Service caches the image. When deploying an application, Container Service uses the cached image instead of pulling the image from the repository if the image tag is the same as that of the local cache. Therefore, if you modify your codes and image but do not modify the image tag for the convenience of upper business, Container Service uses the old image cached locally to deploy the application. With this check box selected, Container Service ignores the cached image and re-pulls the image from the
repository when deploying the application to make sure the latest image and codes are always used.

5. Click Create with Image.

Click Create with Image. Set the following parameters according to your requirements.

a) In the General section:

- Set the Image Name and Image Version.
  
  You can select an image provided by Container Service or enter your image address in the format of `domainname/namespace/imagename:tag`. To select an image, click Select image, select the image, and then click OK. By default, the Container Service uses the latest image version. To use another version of the image, click Select image version, and then click OK.

- Set the number of containers (Scale).

- Select the Network Mode of the application. Currently, Container Service supports two network modes: Default and host. The Default mode is the bridge network mode. The host network mode allows containers to use
the network stacks of Elastic Compute Service (ECS) instances. For more information, see Docker container networking.

- Set the Restart field.

   The Always check box is selected by default. With the check box selected, the containers are restarted regardless of the exit status code. Docker
daemon restarts the containers unlimitedly. Whatever the container status is, the container tries to be restarted when daemon is started.

When the check box is not selected, the restart policy becomes no, indicating containers are not restarted automatically on exit.

b) In the Container section:

- Set the startup command (Command and Entrypoint) of the container. If configured, the image default configurations are overwritten.

  Command is used to specify the startup command of the container main process. For more information, see Command.

  Entrypoint is used to specify the container startup process and parameter. Used together with command, the cmd contents can be passed to Entrypoint as parameters. For more information, see Entrypoint.

- Set the resource limits (CPU Limit and Memory Limit) of the container.

  Set the resource limit for the CPU and memory to be used by the container. For more information, see #unique_43.

- Set the Capabilities.

  For how to add or drop Linux related privileges for the container, see Capabilities.

- Set the Container Config.

  c) In the Network section:
- Set the Port Mapping. Specify the port mapping for the host and the container, and select TCP or UDP as the protocol.

The port mapping is used for the routing between container and host, and is the precondition of Web Routing and Load Balancer. The container provides external services by means of the configured port mapping.

- Set the Web Routing. The cluster automatically creates the acsrouting application, including the routing service, and provides the simple routing function. A routing service instance is deployed on each node. In a node, the `acsrouting_routing_index` container implements the routing forward in the cluster to route the HTTP or HTTPS service. For more information, see #unique_44.

**Note:**
When exposing the HTTP/HTTPS services, you can use the overlay network or Virtual Private Cloud (VPC) to directly access the container port, without configuring the specific host port.

- Set the Load Balancer. Configure the port mapping before configuring the mapping of container_port$scheme://$[slb_name|slb_id]:$slb_front_port. For how to use the Server Load Balancer label, see #unique_45.

When configuring this parameter, control the routing access path on your own, including the routing mapping of Server Load Balancer front end port > backend host port > container port.

d) Set the Data Volume.

- Create a data volume. Enter the host path or data volume name, the container path, and select RW or RO as the data volume permission. For more information, see volume.

- Configure the volumes_from field. Enter the name and permission parameter of another service or container, such as service_name:ro. If no access permission is specified, RW is the default permission. For more
information, see `volumes_from`. After the configuration, the container is authorized to use volumes of another service or container.

e) Set the Environment variables.

Formats such as array, dictionary, and boolean are supported. For more information, see `Environment variables`.

f) Set the container Labels.

For the extension labels supported by Container Service, see `#unique_46`.

g) In the Deploy section:

- Set whether to enable Smooth Upgrade for containers.
  
  For more information, see `#unique_47`.

- Set the Across Multiple Zones settings for containers.
  
  Select Ensure to deploy containers in two zones. The container creation fails if less than two zones are in the current cluster, or the containers cannot be deployed in two zones because of limited machine resources. Select Try best to try to deploy containers in two zones. The container can still be created even if this condition is not met.

  If this parameter is not configured, Container Service deploys the containers in one zone by default. For more information, see `#unique_48`.

- Set whether or enable the container Auto Scaling.

  For more information, see `#unique_49`.

h) Click Create at the right of the page after completing the settings.
6. Click Create with Orchestration Template.

   ![Create Application interface](image)

   a) Click Use Existing Orchestration Template or write a new template by yourself. The contents of the orchestration template comply with the Docker Compose format.

   b) Click Select next to the template after clicking Use Existing Orchestration Template.

   c) Edit the orchestration template.

   Edit the orchestration template according to your requirements. Make modifications in the template directly, or click Edit to modify the service or Delete to delete the service.

   ![Orchestration template editor](image)

   Click Add Service to add another service to this orchestration template. Select the image and configure the parameters. Then, click OK.
d) Click Create and Deploy after completing the settings.

### 6.2 Schedule an application to specified nodes

To deploy an application to specified nodes, we recommend that you use user tags and the **constraint** keyword to make the deployment configurations.

**Note:**
- The deployment constraint only works for newly created containers. It does not work when existing containers change the configurations.
• After you use a user tag to deploy an application, deleting the user tag does not affect the deployed application, but will affect the next deployment of the application. Proceed with caution when deleting user tags.
Procedure

1. Add user tags for nodes.
   a. Log on to the *Container Service console*.
   b. Click Swarm > Clusters in the left-side navigation pane.
   c. Click Manage at the right of the cluster.
   d. Click User Tags in the left-side navigation pane.
   e. Select the nodes that you want to deploy the application and then click Add Tag.
   f. Enter your tag key and tag value, and then click OK to add user tags for the selected nodes.
2. Create an application by clicking Create with Orchestration Template. Configure the `constraint` keyword in the template.

For information about how to create an application, see Create an application.

```
environment:
  constraint:group==1 #Indicates to deploy the application on all the nodes with the "group:1" tag
```

Delete a user tag

1. Log on to the Container Service console.
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster.
4. Click User Tags in the left-side navigation pane.
5. Select the nodes that you want to delete the user tags and then click Delete Tag.
6. The confirmation dialog box appears. Click OK.
7 Configurations

7.1 Implement multiple environments by using configurations

An application consists of codes and configurations. After an application is containerized, the configurations are usually transmitted by using container environment variables to deploy multiple applications using the same image and different configurations.

Limits

- When associating a configuration file with an application, make sure the configuration file is in the same region as the application.
- Currently, associating a configuration file when creating an application is only available when you create the application by using an orchestration template.

Create an application

1. Log on to the Container Service console.
2. Under Swarm, click Configurations in the left-side navigation pane. Select the region in which you want to create a configuration from the Region list and click Create.
3. Complete the settings and then click OK.

- File Name: It can contain 1–32 characters.
- Description: It can contain up to 128 characters.
- Configuration: You can add up to 50 configurations in a region.

In this example, the `size` variable is set.

4. Under Swarm, click Applications in the left-side navigation pane. Select the cluster in the same region as the created configuration from the Cluster list and click Create Application.

5. Enter the basic information of the application and click Create with Orchestration Template.

6. Enter the following orchestration template and then click Create and Deploy.

   Wherein, `size` is a dynamic variable and will be overwritten by the value in the configuration.

   ```yaml
   busybox:
     image: 'busybox'
     command: 'top -b'
     labels:
   ```
7. The dialog box appears. Select the configuration file to be associated with from the Associated Configuration File drop-down list. Click Replace Variable and click OK.

<table>
<thead>
<tr>
<th>Template Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associated Configuration File:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Contrast</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td></td>
<td>Miss</td>
</tr>
</tbody>
</table>

Description:
- **Same**: The selected configuration file contains this variable and the variable values are the same.
- **Diff**: The selected configuration file contains this variable but the variable values are different.
- **Miss**: The selected configuration file does not contain this variable.

Replace Variable  OK  Cancel

Update an application

If you associated a configuration file when creating an application, you can update the application by modifying the configuration file and redeploying the application.

1. Log on to the Container Service console.

2. Under Swarm, click Configurations in the left-side navigation pane. Select the region in which the configuration you want to modify resides from the Region list, and click Modify at the right of the configuration.

3. Click Confirm in the displayed dialog box.
4. Click Edit (changes to Save after you click it) at the right of the variable you want to modify. Modify the variable value. Click Save and then click OK.

5. Under Swarm, click Applications in the left-side navigation pane. Select the cluster in the same region as the created configuration, and then click Redeploy at the right of the application.

After the application is updated, the number of containers changes to three.
Trigger an update

If you associated a configuration file when creating an application, you can redeploy the application by using the redeployment trigger.

1. Log on to the Container Service console.
2. Under Swarm, click Configurations in the left-side navigation pane. Select the region in which the configuration you want to modify resides from the Region list, and click Modify at the right of the configuration.
3. Click Confirm in the displayed dialog box.
4. Click Edit (changes to Save after you click it) at the right of the variable you want to modify. Modify the variable value. Click Save and then click OK.
5. Create a redeployment trigger.

For how to create a trigger, see #unique_53.

6. Initiate the redeployment trigger.

```bash
curl "https://cs.console.aliyun.com/hook/trigger?triggerUrl= Y2ViZDhkZTIwZGMyMjRmOTM4NDIzMTgwMzI3NmIwM2IxfHRlc3QtZ3JvdXB8c2NhGluZ3wxXZwVzNmOIFtNcwfa==&secret=466242376775654951546d6451656a7a66e7f5b61db6885f8d15aa64826672c2"
```

After the application is updated, the number of containers changes to three.
8 Data volumes

8.1 Overview

The characteristic of Docker determines the containers are non-persistent. Deleting a container also deletes its data. Data volumes provided by Docker can realize persistent storage by attaching to the host directories, but the data volumes in the host have the following limits in the cluster environment:

- Data cannot be migrated when containers are migrated between machines.
- Different machines cannot share data volumes.

To solve these issues, Alibaba Cloud Container Service provides third-party data volumes. By packaging various cloud storage resources as data volumes, these data volumes can be attached to containers directly and automatically reattached when containers are restarted or migrated. Currently, cloud disks and OSSFS are supported.

8.2 Create an OSSFS data volume

OSSFS is a FUSE-based file system provided by Alibaba Cloud (click https://github.com/aliyun/ossfs to view the project homepage). OSSFS data volumes can package Object Storage Service (OSS) buckets as data volumes.

The performance and functions of OSSFS differ from those of local file systems because data must be synchronized to the cloud by the means of network. Do not run databases, I/O-intensive applications, logs and other applications that require constantly writing files to OSSFS. OSSFS is suitable for sharing configuration files across containers, uploading attachments, and other scenarios without rewrite operations.

OSSFS differs from local file systems in the following ways:

- Random write or append write leads to the entire file being overwritten.
- Metadata operations, such as list directory, provide poor performance because the system must remotely access the OSS server.
- The file/folder rename operation is not atomic.
• Coordinate the actions of each client on your own when multiple clients are mounted to the same OSS bucket. For example, avoid multiple clients from writing the same file.
• Hard link is not supported.

Prerequisites

You can only use the data volume function when your cluster meets the following conditions:

• The cluster Agent is of version 0.6 or later.

You can check your Agent version on the Cluster List page. Select the target cluster, and click More > Upgrade Agent on the right.

If your Agent version is earlier than 0.6, upgrade the Agent first. For how to upgrade Agent, see Upgrade Agent.

• Deploy the acsvolumedriver application in the cluster. We recommend that you upgrade the acsvolumedriver application to the latest version.

You can deploy and upgrade the acsvolumedriver application by upgrading the system services. For more information, see Upgrade system services.

Note:
When acsvolumedriver is upgraded or restarted, containers that use OSSFS data volumes are restarted, and your services are also restarted.
Step 1. Create an OSS bucket

1. **Log on to the OSS console.**
2. and create a bucket.

   In this example, create a bucket in the region of China East 1 (Hangzhou).

Step 2. Create an OSSFS data volume

1. **Log on to the Container Service console.**
2. Click Data Volumes in the left-side navigation pane.
3. Select the cluster in which you want to create a data volume from the Cluster list. Click Create in the upper-right corner.
4. The Create Data Volume dialog box appears. Select OSS as the Type, configure the data volume parameters, and then click Create. Container Service creates a data volume with the same name on each cluster node.

- **Name:** The name of the data volume that must be unique within the cluster.
- **Access Key ID/Access Key Secret:** The AccessKey required to access OSS. You can obtain them from the AccessKey console.
- **Bucket ID:** The name of the OSS bucket to be used. Click Select Bucket in the dialog box, and click Select.
• Access domain name: If the bucket and ECS instances are in different regions, select external domain name. If they are located in the same region, you must select the corresponding cluster network type. For VPC network, select VPC domain name, and for classic network, select intranet domain name respectively.

• File Caching: Select Disable if you want to synchronize the modifications of the same file on multiple machines (for example, modify the file on machine A and read the modified contents on machine B).

Note:
Turning off the File Caching causes `ls` folder to become slow, especially when a lot of files exist in the same folder. Therefore, when there is no such requirement, enable the File Caching and increase the speed of the `ls` command.

You can view the created OSSFS data volumes on the Data Volume List page.

Subsequent operations

After the data volumes are created, you can use the data volumes created in your app. For more information about how to use data volumes in an application, see Use third-party data volumes.

8.3 Create cloud disk data volumes

Cloud disk is a block storage system officially provided by Alibaba Cloud, and an elastic block storage product of distributed storage architecture that Alibaba Cloud provides to Elastic Compute Service (ECS). Cloud disk provides random storage of data block level, features in low latency, persistence, and high reliability, and adopts the distributed mechanism of three copies.

Cloud disk can be used for relational database applications or development and test applications. For more information, see #unique_59.

Limits

• The cloud disk and the ECS instances in the cluster must be in the same region and zone.
• Cloud disk data volumes only support being mounted to a single machine, but does not support the shared mode.
• A cloud disk data volume can be used by only one container at the same time.

Prerequisites

• Create a cloud disk manually in the ECS console before using the cloud disk data volume.
• Upgrade your Agent to the latest version. For more information, see Upgrade Agent.
• Deploy the acsvolumedriver application in the cluster. We recommend that you upgrade the acsvolumedriver application to the latest version.

You can deploy and upgrade the acsvolumedriver application by upgrading the system services. For more information, see Upgrade system services.

Procedure

Step 1 Create a cloud disk

In this example, create a cloud disk that is in the same region and zone as the cluster.

1. Log on to the ECS console.

2. Click Cloud Disks in the left-side navigation pane.

3. On the Disk List page, click Create Cloud Disk in the upper-right corner.

4. Configure the parameters for the cloud disk. Select the corresponding region and zone. Create the cloud disk according to the guidance on the page.

Note:
The purchased cloud disk can be mounted only when you select the same zone as the server. The cloud disk cannot be mounted across zones or regions.

Step 2 Create data volumes by using the cloud disk

1. Log on to the Container Service console.
2. Click Data Volumes in the left-side navigation pane.
3. Select the cluster in which you want to create the data volume from the Cluster list and then click Create in the upper-right corner.
4. In the displayed dialog box, select Cloud Disk as the Type, configure the data volume parameters and then click Create. Container Service will create a data volume with the same name on each cluster node.

- **Name**: The name of the data volume, The data volume name must be unique within the cluster.
- **Cloud Disk ID**: Select the cloud disk to be mounted and is in the same region and zone as the cluster. In this example, select the ID of the cloud disk created in step 1.
- **AccessKey ID and AccessKey Secret**: The AccessKey of your account.

  Note: Please use this function after upgrading to the volume driver to the latest version. You can manage volume drivers through Cluster List > “More” > “Upgrade System Service”.

- **File System Type**: You can select the data type in which data is stored to the cloud disk. The supported types include ext4, ext3, xfs, and vfat.
After the data volume is successfully created, you can view the cloud disk data volume on the Data Volume List page.

![Data Volume List](image)

Subsequent operations

You can manage the cloud disk data volumes, including deleting all the data volumes with the same name and viewing data volume parameters.

8.4 View and delete data volumes

You can view and delete the created data volumes.

Procedure

1. Log on to the Container Service console.

2. Click Data Volumes in the left-side navigation pane and select the target cluster.

   All the data volumes in the selected cluster are displayed on the Data Volume List page, including the local data volumes and third-party data volumes.

   On this page, you can view the containers that reference the data volumes.

   ![Data Volume List](image)

   For local data volumes, the data volume name is in the format of `node_name/volume_name`.

   For third-party data volumes, you can click View under Volume Parameters to view the parameters of the data volumes.

   When you create a third-party data volume, Container Service creates the data volume with the same name on each node in the cluster, allowing containers to
be migrated between nodes. You can also select to Delete all volumes with the same name.

**Note:**
Data volumes referenced by containers cannot be deleted. The Data Volume List page displays the containers that reference the data volume. You must delete the containers that references the data volume before you can delete the data volume.

8.5 Use third-party data volumes

Third-party data volumes are used in the same way as local data volumes.

You can set the data volumes when creating an application or changing the configurations of an existing application.

Prerequisite

You have created a data volume in Container Service console. For details, see [Create an OSSFS data volume](#).

Procedure

**Take the OSSFS data volume test in the test cluster as an example.**

<table>
<thead>
<tr>
<th>Node</th>
<th>Volume Name</th>
<th>Driver</th>
<th>Mount Point</th>
<th>Container</th>
<th>Volume Parameters</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>d9035293282143102e4123b3b2a02...</td>
<td>test</td>
<td>OSS File System</td>
<td>/mnt/wa_mnt/osfs/volum...</td>
<td>View</td>
<td></td>
<td>Delete All Volumes with the Same Name</td>
</tr>
<tr>
<td>d9035293282143102e4123b3b2a02...</td>
<td>test</td>
<td>OSS File System</td>
<td>/mnt/wa_mnt/osfs/volum...</td>
<td>View</td>
<td></td>
<td>Delete All Volumes with the Same Name</td>
</tr>
</tbody>
</table>

Create an application by using an image

1. Log on to the [Container Service console](#).
2. Click Applications in the left-side navigation pane.
3. Click Create Application in the upper-right corner.
4. Enter the basic information for the application you want to create and then click Create with Image. In this example, enter volume as the Name and select test as the Cluster.

**Note:**
The cluster on which the application will be deployed must be the same as the one of the OSSFS data volume that you want to use.

5. Select the image you want to use and complete the other configurations.

Note:
For how to create an application by using an image, see Create an application.

6. Click the plus icon in the Volume section. Enter the data volume name in the Host Path or Data Volume Name field. Enter the Container Path and select RW or RO as the data volume permission.

7. Click Create at the right of the page after completing the settings.

On the Data Volume List page, you can see that the OSSFS data volume test is referenced by the container of the volume application.

Create an application by using an orchestration template

1. Log on to the Container Service console.
2. Click Applications in the left-side navigation pane.
3. Click Create Application in the upper-right corner.
4. Enter the basic information for the application you want to create and then click Create with Orchestration Template. In this example, enter volume as the Name and select test as the Cluster.

**Note:**
The cluster on which the application will be deployed must be the same as the one of the OSSFS data volume that you want to use.

5. Click Use Existing Orchestration Template or use your own orchestration template.

**Note:**
For how to create an application by using an orchestration template, see [Create an application](#).

6. In the volumes section of the template, enter the data volume name, container path, and permission.
7. Click Create and Deploy after completing the settings.

On the Data Volume List page, you can see that the OSSFS data volume test is referenced by the container of the volume application.

Change the configurations of an existing application

1. Log on to the Container Service console.

2. Click Applications in the left-side navigation pane.

3. Select the cluster (the test cluster in this example) in which the application resides from the Cluster list. Click Update next to the application you want to change the configurations.

For how to change the application configurations, see #unique_61.

Note:
Make sure the application and the OSSFS data volume you want to use are in the same cluster.
4. The Change Configuration dialog box appears. In the **volumes** section of the template, enter the data volume name, container path, and permission.

![Change Configuration dialog box](image)

Note: The version of the application must be changed; otherwise, the "OK" button is not available.

5. Click OK after completing the modifications.

On the Data Volume List page, you can see that the OSSFS data volume test is referenced by the container of the volume application.
8.6 FAQ

The container fails to be launched and the system reports an error such as chown /mnt/acs_mnt/ossfs/XXXX: input/output error if you use the third-party data volume in the method of Data volume name: an existing directory in the image (for example, o1:/data, when the /data directory exists in the image).

This error occurs because for named data volumes, Docker copies the existing files in the image to the data volumes and uses chown to set the relevant user permissions. However, Linux prohibits the use of chown for mount points.

To solve this issue, you can use one of the following solutions:

- Upgrade Docker to version 1.11 or later versions. Upgrade Agent to the latest version and specify nocopy in the orchestration template. Docker will not copy the data and thereby, no chown error will occur.

```plaintext
volumes:
  - o1:/data:nocopy
  - /tmp:/bbb
```

- If you need to copy the data, use the mount point path instead of the data volume name to set the data volume. For example, /mnt/acs_mnt/ossfs/XXXX:/data. However, this method bypasses the volume driver. When the machine is restarted, the container might be started before the OSSFS is successfully mounted and the container might be attached to a local data volume. To avoid this issue, use two data volumes at the same time. One is set by the data volume name and the other is set by the mount point path. The data volume set by the data volume name is only used for synchronizing with the volume driver and is not used for storage.

```plaintext
volumes:
  - o1:/nouse
  - /mnt/acs_mnt/ossfs/XXXX:/data
  - /tmp:/bbb
```
9 Logs

9.1 Enable Log Service

Log Service is a platform service for log scenarios. You can collect, distribute, ship, and query logs quickly without development, which is applicable to scenarios such as log transfer, monitoring, performance diagnosis, log analysis, and audit. Container Service integrates with Log Service, which allows you to send the application logs to Log Service.

Note:
On the cluster management page, choose Enable Log Service > OK. After Log Service is successfully enabled, the log index is created for each automatically created Logstore by using the built-in Resource Access Management (RAM) account. With this feature enabled, you are charged for the Alibaba Cloud Log Service usage after configuring the following settings. For more information, see #unique_65. Make sure you know your log volume to avoid large unexpected costs.

Enable Log Service

1. Log on to the Container Service console.
2. Click Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster.
4. Click Enable Log Service in the upper-right corner.
5. In the dialog box, click OK.

Before enabling Log Service in Container Service, activate the RAM service and Log Service first. Click Activate It to activate the RAM service and Log Service if they are not activated yet. The created Log Service project is displayed after Log Service is successfully enabled.

![Enable Log Service - test dialog box](image)

Are you sure you want to enable the Log Service?
The Log Service must be enabled for application log collection. Log data will be stored in Alibaba Cloud Log Service even after a cluster is deleted.

**Help on using Log Service in clusters:** [Alibaba Cloud Log Service Integration](#)

1. The RAM service must be activated before using Log Service. Please activate it if you have not. [Activate It](#)
2. Please activate Log Service if you have not. [Activate It](#)
3. Aliyun log service will have costs, see product price

Check installation result of acslogging service

**Container Service installs the Agent required by Log Service on your machine if this is the first time Log Service is enabled.** You can use Log Service after the application is installed successfully. You can find this application on the Application List page. You can use Log Service after the application is installed successfully.

1. Log on to the **Container Service console**.
2. Click Applications in the left-side navigation pane.
3. Select the cluster from the Cluster list and clear the **Hide System Applications** check box.

The acslogging application is successfully installed.

![Application List](image)
The system creates a corresponding project in Alibaba Cloud Log Service. You can view the project in the Log Service console. The project name contains the Container Service cluster ID.

Use Log Service in orchestration files

Most Docker applications write the logs directly to stdout, now you can do this as well (for the scenarios of writing logs to files, see *Use file logs* in the following section). After enabling Log Service, stdout logs are automatically collected and sent to Alibaba Cloud Log Service.

In the following example a WordPress application is created. It contains two services: WordPress service and MySQL service. Logs are collected to Alibaba Cloud Log Service, which contains two services: WordPress service and MySQL service. Logs are collected to Alibaba Cloud Log Service.

MySQL and WordPress

```yaml
mysql:
  image: mysql
  ports:
  - 80
  labels:
    aliyun.scale: "1"
  environment:
    MYSQL_ROOT_PASSWORD=password

web:
  image: registry.aliyuncs.com/jiangjizhong/wordpress
  ports:
  - 80
  labels:
    aliyun.routing.port_80: wordpress-with-log
    aliyun.log_store_dbstdout: stdout  # Collect stdout logs to the dbstdout Logstore.
    aliyun.log_ttl_dbstdout: 30  # Set the data retention time for the dbstdout Logstore to 30 days.
  links:
  - mysql
```

In the preceding orchestration file:
Container Service

User Guide / 9 Logs

- **aliyun.log_store_dbstdout**: stdout indicates to write the container standard to the Logstore `acslog-wordpress-dbstdout`. The label format is `aliyun.log_store_{name}: {logpath}`. Wherein:
  - `name` is the name of the Alibaba Cloud Log Service Logstore. The actually created Logstore name is `acslog-${app}-${name}`.
  - `app` is the application name.
  - `logpath` is the log path in the container.
  - `stdout` is a special `logpath`, indicating the standard output.

- **aliyun.log_ttl_<logstore_name>** is used to set the data retention time (in days) for the Logstore. The value range is 1–365. If left empty, logs are kept in the Logstore for two days by default.

Note:
The value configured here is the initial configuration value. To modify the data retention time later, modify it in the Log Service console.

You can create an application named `wordpress` in the Container Service console by using the preceding orchestration file. After the application is started, you can find the Logstore `acslog-wordpress-dbstdout` in the Log Service console, in which stores the logs of application `wordpress`.

**View logs in Log Service console**

After deploying an application by using the preceding orchestration file, you can view the collected logs in the Alibaba Cloud Log Service console. Log on to the Log Service console. Find the Log Service project corresponding to the cluster. You can view the Logstore `acs-wordpress-dbstdout` used in the orchestration file.

Click Search at the right of the Logstore to view the logs.
Use file logs

To write the logs directly to files (for example, `/var/log/app.log`) instead of stdout, configure as follows:

```
aliyun.log_store_name: /var/log/app.log
```

*name* is the Logstore name. `/var/log/app.log` is the log path in the container.

To output multiple log files to Log Service, configure as follows to put the files under multiple directories:

```
aliyun.log_store_s1: /data/logs/access/access.log
aliyun.log_store_s2: /data/logs/error/error.log
aliyun.log_store_s3: /data/logs/exception/*.log #Wildcards are supported
```

**Note:**
Currently, multiple Logstores cannot correspond to the same log directory. The log files corresponding to the three Logstores s1, s2, and s3 in the preceding example must be under three directories.

Enable timestamp

You can select whether to add timestamp when Docker is collecting logs. Configure timestamp by using the `aliyun.log.timestamp` label in Container Service. The timestamp is added by default.

- **Add timestamp**
  ```
  aliyun.log.timestamp: "true"
  ```

- **Remove timestamp**
  ```
  aliyun.log.timestamp: "false"
  ```
10 DevOps

10.1 Jenkins-based continuous delivery

As an important step in agile development, continuous integration aims to maintain high quality while accelerating product iteration. Every time codes are updated, an automated test is performed to test the codes and function validity. The codes can only be delivered and deployed after they pass the automated test. This document mainly introduces how to integrate Jenkins, one of the most popular continuous integration tools, with Alibaba Cloud Container Service to realize automated test and image building push.

The following example demonstrates how to perform automated test and build a Docker image by using Alibaba Cloud Container Service Jenkins, which realizes high-quality continuous integration.

Background information

Every time codes are submitted to nodejs project in GitHub, Alibaba Cloud Container Service Jenkins will automatically trigger a unit test. If the test is successful, Jenkins continues to build images and then pushes them to a target image repository. Finally, Jenkins notifies you of the results by email.

A general process is as follows.

Slave-nodejs is a slave node used for unit test and building and pushing the image.
Jenkins introduction

Jenkins is an open-sourced continuous integration tool developed on Java. It monitors and triggers continuously repeated work and supports expansion of multiple platforms and plug-ins. Jenkins is an open-sourced tool featuring easy installation and interface-based management. It uses job to describe every work step, and node is a project execution environment. The master node is a default execution environment of a Jenkins job and also the installation environment for Jenkins applications.

Master/slave

Master/slave is equivalent to the server/agent concept. A master provides Web interface with which you manage the job and slave. The job can run on the master or be assigned to the slave. One master can be associated with several slaves to serve different jobs or different configurations of the same job.

Several slaves can be configured to prepare a separate test and building environment for different projects.

Note:
The Jenkins job and project mentioned in this document all refer to a build unit of Jenkins, namely, an execution unit.

Step 1 Deploy Jenkins applications and slave nodes

The building and testing of different applications need different dependencies. The best practice is to use different slave containers with corresponding runtime dependencies and tools to perform the test and building. By using the slave images and sample templates provided by Alibaba Cloud Container Service for different environments such as Python, Node.js, and Go, you can quickly and easily generate Jenkins applications and various slave nodes, configure node information in Jenkins applications, and specify the execution nodes in the build projects so as to implement the entire continuous integration process.

Note:
For images provided by Alibaba Cloud Container Service for developing slave nodes, see https://github.com/AliyunContainerService/jenkins-slaves.

1.1 Create a Jenkins orchestration template
Create a template and create the orchestration based on the following contents.

The labels supported by Alibaba Cloud Container Service Jenkins master are: 1.651.3, 2.19.2, and 2.32.2.

**Note:**

For how to create an orchestration template, see [http://example.com](http://example.com).

```yaml
jenkins:
  image: 'registry.aliyuncs.com/acs-sample/jenkins:1.651.3'
  volumes:
    - /var/lib/docker/jenkins:/var/jenkins_home
  restart: always
  labels:
    aliyun.scale: '1'
    aliyun.probe.url: 'tcp://container:8080'
    aliyun.probe.initial_delay_seconds: '10'
    aliyun.routing.port_8080: jenkins
  links:
    - slave-nodejs

slave-nodejs:
  image: 'registry.aliyuncs.com/acs-sample/jenkins-slave-dind-nodejs'
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  restart: always
  labels:
    aliyun.scale: '1'
```

1.2 Use the template to create Jenkins application and slave node

Use the orchestration template created in the preceding section or the Jenkins sample template provided by Alibaba Cloud Container Service to create the Jenkins application and slave node.

**Note:**

For how to create an application by using an orchestration template, see [Create an application](http://example.com).

After a successful creation, the Jenkins application and slave node are displayed in the service list.
Open the access endpoint provided by Container Service to use the deployed Jenkins application.

Step 2: Realize automated test and automated build and push of image

2.1 Configure the slave container as the slave node of the Jenkins application

Open the Jenkins application. Click Manage Jenkins in the left-side navigation pane. Click Manage Nodes on the right pane. Click New Node in the left-side navigation pane. Enter the node name and then click OK. Then, complete the parameters as follows.
2.2 Create a project to implement automated test

1. Go back to the Jenkins home page. Click New Item in the left-side navigation pane. Enter the item name, select Freestyle project, and then click OK.

2. Enter the project name and select a node for running the project. In this example, enter the slave-nodejs-ut node prepared in the preceding section.
3. Configure the source code management and code branch. In this example, use GitHub to manage source codes.

4. Configure the build trigger. In this example, automatically trigger project execution by combining GitHub Webhooks & services.

5. Add the Jenkins service hook to GitHub to implement automatic triggering.

   On the GitHub project home page, click the Settings. Click Webhooks & services, click Add Service, and then select Jenkins(Git plugin) from the drop list. In the
dialog box of Jenkins hook url, enter ${Jenkins IP}/github-webhook/. For example:

http://jenkins.cd**************.cn-beijing.alicontainer.com/github-webhook/

6. Add a build step of Execute shell type and write shell scripts to perform the test.

The commands in this example are as follows:

```
pwd
ls
cd chapter2
npm test
```

SVN source code example:
Select Subversion in Source Code Management and enter the SVN repository address in the Repository URL field (if the Jenkins master and SVN server are in different time zones, add @HEAD at the end of the repository address). Add the username and password of the SVN server in Credentials.

Configure the build trigger. In this example, Post-commit hook is used to automatically trigger the project execution. Enter your configured token in Token Name.

Log on to the SVN server. Create a post-commit file in the hooks directory of the code repository (svn-java-demo).

```
cd /home/svn/svn-java-demo/hooks
cp post-commit.tmpl post-commit
chmod 755 post-commit
```

Add the curl -u ${Jenkins_account}:${password}

```
${Jenkins_url}/job/svn/build?
token=${token} command
```

in the <g id="1">post-commit</g> file. For example:

```
curl -u test:test
```
2.3 Create a project to automatically build and push images

1. Go back to the Jenkins home page. Click New Item in the left-side navigation pane. Enter the item name, select Freestyle project, and then click OK.
2. Enter the project name and select a node for running the project. In this example, enter the slave-nodejs-ut node prepared in the preceding section.
3. Configure the source code management and code branch. In this example, use GitHub to manage source codes.
4. Add the following trigger and set to automatically build the image only after the unit test is successful.

5. Write the shell script for building and pushing images.

The commands in this example are as follows:

```bash
cd chapter2
sudo docker build -t registry.aliyuncs.com/qinyujia-test/nodejs-demo .
sudo docker login -u ${yourAccount} -p ${yourPassword} registry.aliyuncs.com
sudo docker push registry.aliyuncs.com/qinyujia-test/nodejs-demo
```

Step 3 Automatically redeploy the application

3.1 Deploy the application for the first time
Use the orchestration template to deploy the image created in step 2.3 to Container Service and create the nodejs-demo application.

Example:

```yaml
express:
  image: 'registry.aliyuncs.com/qinyujia-test/nodejs-demo'
  expose:
    - '22'
    - '3000'
  restart: always
  labels:
    aliyun.routing.port_3000: express
```

3.2 Automatic redeployment

1. Select the created application nodejs-demo and create the trigger.

   Note:
   For how to create a trigger, see #unique_53.

2. Add a line to the shell script in 2.3. The address is the trigger link of the created trigger.

   ```sh
```

3. Change the command in the example of 2.3 as follows:

   ```sh
cd chapter2
   sudo docker build -t registry.aliyuncs.com/qinyujia-test/nodejs-demo .
   sudo docker login -u ${yourAccount} -p ${yourPassword} registry.aliyuncs.com
   sudo docker push registry.aliyuncs.com/qinyujia-test/nodejs-demo
```

After pushing the image, Jenkins automatically triggers the redeployment of the nodejs-demo application.

Step 4 Configure email notification of the results

To send the unit test or image building results to relevant developers or project execution initiators by email, perform the following configurations:
1. On the Jenkins homepage, click Manage Jenkins > Configure System, and configure the Jenkins system administrator email.

![Jenkins Location](image1)

2. Install the Extended Email Notification plug-in, configure the SMTP server and other relevant information, and then set the default email recipient list, as shown in the following figure:

![Email Notification](image2)

The preceding example shows the parameter settings of the Jenkins application system. The following example shows the relevant configurations for Jenkins projects whose results are to be pushed by email.

3. Add post-building steps in the Jenkins project, select Editable Email Notification and enter the email recipient list.
4. Add a trigger to send emails.
11 Service discovery and load balancing

11.1 Routing and Server Load Balancer between services in a cluster

Container Service can expose the HTTP service based on domain names by using acsrouting, and work with health check to enable the automatic Server Load Balancer and service discovery. When one container malfunctions, routing will automatically remove the container that failed the health check from the backend, which achieves the automatic service discovery. However, in this way, the service is exposed to the Internet.

Then, how can automatic service discovery and Server Load Balancer be achieved between services in a cluster by using this method? The routing container of Alibaba Cloud Container Service has the function of Server Load Balancer. Use the domain name ending with .local to make the container can only be accessed by the other containers in the cluster, and then work with the external_links label to implement the inter-service discovery and Server Load Balancer in the cluster.

Implementation principle

1. Docker version later than 1.10 supports alias resolution in the container. In the restservice container that depends and loads on the restserver.local, the restserver.local domain name resolves the address of the routing container. When the restclient service initiates a request, the HTTP request is forwarded to the routing container, with HOST as the request header of restserver.local.

2. Routing container monitors the health status of the containers configured with aliyun.routing.port_xxx: restserver.local label and mounts the status to the backend of HAProxy. When HAProxy receives the HTTP request with the restserver.local HOST header, the request can be forwarded to the corresponding container.
Advantages

- Compared with the DNS-based method using link or hostname, the inconsistent handling of DNS cache by different clients will delay service discovery, and the DNS solution which only includes round robin cannot meet the requirements of microservice scenarios.

- Compared with other microservice discovery solutions, this solution provides a mechanism to achieve unrelated service discovery and Server Load Balancer, which can be used without any modification on the server side or client application.

- In decoupling service lifecycle, every microservice can adopt a Docker Compose template for independent deployment and update. Only a virtual domain name is required to achieve dynamic mutual binding.

Orchestration example

In the following orchestration example, add the `aliyun.routing.port_80: restserver.local` label to the restserver service to make sure only the containers in the cluster can access this domain name. Then, configure `external_links` for the restclient service, pointing to the restserver.local domain name. The restclient
service can use this domain name to access the restserver service, and work with health check to implement automatic service discovery.

restserver:  # Simulate the rest service.
  image: nginx
  labels:
    aliyun.routing.port_80: restserver.local  # Use the local domain name and only the containers in the cluster can access this domain name.
    aliyun.scale: "2"  # Expand two instances to simulate the Server Load Balancer.
    aliyun.probe.url: "http://container:80"  # Define the container health check policy as http and the port as 80.
    aliyun.probe.initial_delay_seconds: "2"  # The health check starts two seconds after the container is started.
    aliyun.probe.timeout_seconds: "2"  # The timeout for health check. A container is considered as unhealthy if no result is returned in two seconds.

restclient:  # Simulate the rest service consumer.
  image: registry.aliyuncs.com/acs-sample/alpine:3.3
  command: "sh -c 'apk update; apk add curl; while true; do curl --head restserver.local; sleep 1; done'"  # Access the rest service and test the Server Load Balancer.

  tty: true
  external_links:
    - "restserver.local"  # Specify the link service domain name. Make sure that you set external_links. Otherwise, the access fails.

The following restclient service logs show that the HTTP request of restclient curl is routed to the containers of different rest services. The container ID is 053cb232fd fbcb5405ff791650a0746ab77f26cfe74f9ea2f075c2af55c97f5 and b8c36abca5 25ac7fb02d2a9fcaba8d36641447a774ea956cd93068419f17ee3f.
11.2 Custom routing - simple sample

In this example, an container is deployed, services are exposed by using a Server Load Balancer instance (with the label) externally, and an Nginx server is attached at the backend. This example only shows the Nginx homepage, and other functions will be added based on the basic example.

Note:
Different services cannot share the same Server Load Balancer. Otherwise, the backend machines of Server Load Balancer will be deleted and the services will become unavailable.

Basic example

The compose template is as follows:

```
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each virtual machine (VM).
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
  environment:
    # Indicates the range of backend containers that support route loading. "*" indicates the whole cluster. By default, it indicates the services in applications.
    ADDITIONAL_SERVICES: "*"
  appone:
    expose: # For proxied services, use expose or ports to tell proxy containers which port is to be exposed.
```
After the service is successfully started, the following figure appears.

![Welcome to nginx!](image)

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org). Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

Enable session persistence

```yaml
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each VM.
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
  environment:
    # Indicates the range of backend containers that support route loading. "*" indicates the whole cluster. By default, it indicates the services in applications.
    ADDITIONAL_SERVICES: "*"

appone:
  ports:
    - 80/tcp
    - 443/tcp
  image: 'nginx:latest'
  labels:
    # http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
    # Session persistence is enabled, the cookie method is applied, and the key is CONTAINERID.
    aliyun.proxy.COOKIE: "CONTAINERID insert indirect"
```
Customize 503 page

When the VIP address of the Server Load Balancer instance instead of the domain name is entered, the 503 error page is returned as follows.

```
HTTP/1.0 503 Service Unavailable
Cache-Control: no-cache
Connection: close
Content-Type: text/html;charset=UTF-8
<html><body><h1>503 Service Unavailable</h1>
<h3>No server is available to handle this request.</h3>
<li>If you are the visitor of this application, contact the application maintainer to solve the problem. </li>
<li>If you are the application maintainer, view the following information. </li>
<li>You are using the simple routing service. The request is sent from Server Load Balancer to the acsrouting application container then to your application container. Follow these steps for troubleshooting. </li>
<li>Log on to the Container Service console. Click "Services" in the left-side navigation pane. Select the corresponding cluster on the "Service List" page. Click the name of the service exposed to the public network. View the "Access Endpoint" of the service, and check whether your access domain name is the same as the domain name configured in the corresponding service. </li>
<li>Locate and troubleshoot the problem by following the instructions described in <a href="https://www.alibabacloud.com/help/faq-detail/42660.html">. </a></li>
<li>If the problem persists, open a ticket and contact the technical staff for help. We will serve you faithfully. </li>
</body></html>
```

You can modify the error page as per your needs. The compose template is modified as follows:

```
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
```

After entering the VIP address of the Server Load Balancer instance, the 503 page is displayed as follows.

503 Service Unavailable
No server is available to handle this request.

If this page is returned, a problem occurs during the service access process. Take the following steps for troubleshooting:

- If you are the visitor of this application, contact the application maintainer to solve the problem.
- If you are the application maintainer, view the following information.
  - You can use the simpler visiting service. The request is sent from Server Load Balancer to the accounting application container running in the application container. Take the following steps for troubleshooting.
  - Log on to the Container Service Management Console, select "Services" in the left navigation pane and select the corresponding "cluster" in "Service List". Click the "services" exposed to the public network, view the "Access Endpoints" of the services, and check whether your access domain is the same as the domain configured in the services.
  - Locate and solve the problem in accordance with "Sample creating service link manually".
- Refer to Hosting FAQs.

Support extensive domain names

Modify the configurations as follows to enable the backend of Nginx to support extensive domain names (that is, the Nginx homepage can be accessed by using appone.example.com and *.example.com).

lb:

```yaml
- image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
```

appone:

```yaml
ports:
  - 80/tcp
  - 443/tcp
image: 'nginx:latest'
labels:
  # You can specify paths when configuring URLs. In this example, http/https/ws/wss are supported.
  aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
  restart: always
```
Bind a host and enter the domain name `www.example.com`. The Nginx homepage is displayed as follows.

![Welcome to nginx!](image)

Configure default backend

**Remove the URL configuration and modify the configurations as follows to enable access to Nginx at the backend by using an IP address.**

```
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each VM.
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
```

```
environment:
  # Indicates the range of backend containers that support
  # route loading. "*" indicates the whole cluster. By default, it
  # indicates the services in applications.
  ADDITIONAL_SERVICES: "*
  # Specify the error page when 503 is returned.
  EXTRA_FRONTEND_SETTINGS_80: "errorfile 503 /usr/local/etc/
  haproxy/errors/503.http"
  volumes:
  # Mount the error page to the container from the host.
  - /errors/:/usr/local/etc/haproxy/errors/

appone:
  ports:
  - 80/tcp
  - 443/tcp
  image: 'nginx:latest'
  labels:
    # Indicates that the service must be proxied.
    aliyun.proxy.required: "true"
  restart: always

After entering the VIP address of the Server Load Balancer instance, the Nginx homepage is displayed as follows.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Select backend based on URL parameter values

You can use different backend proxies based on different URL parameter values.

The following example shows how to access the appone service, that is, the Nginx homepage, by using http://www.example.com?backend=appone and how to access the apptwo service, that is, the hello world homepage, by using http://www.example.com?backend=apptwo. The application template codes are as follows:

lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
  - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription
    registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each VM.
    aliyun.global: "true"
# A Server Load Balancer instance is bound to the frontend.

```
aliyun.lb.port_80: tcp://proxy_test:80
```

```
environment:
  # Indicates the range of backend containers that support
  # route loading. "*" indicates the whole cluster. By default, it
  # indicates the services in applications.
  ADDITIONAL_SERVICES: "*"
  # Obtain the value of the "backend" parameter in the URL and
  # modify the HOST header to the backend domain name which needs to be
  # matched.
  EXTRA_FRONTEND_SETTINGS_80: " http-request set-header HOST %[urlp(backend)].example.com"
```

```
appone:
  ports:
    - 80/tcp
    - 443/tcp
  image: 'nginx:latest'
  labels:
    # You can specify paths when configuring URLs. In this
    # example, http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
  restart: always

apptwo:
  ports:
    - 80/tcp
  image: 'registry.cn-hangzhou.aliyuncs.com/linhuatest/hello-world:latest'
  labels:
    # You can specify paths when configuring URLs. In this
    # example, http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "http://apptwo.example.com"
  restart: always
```

Bind a host and enter the link `http://www.example.com?backend=appone`. Then, the Nginx homepage for the appone service is displayed as follows.

![Welcome to nginx!](https://via.placeholder.com/150)

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Bind a host and enter the link `http://www.example.com?backend=apptwo`. Then, the hello world homepage for the apptwo service is displayed as follows.
Record access logs

```
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each VM.
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
  environment:
    # Indicates the range of backend containers that support route loading. "*" indicates the whole cluster. By default, it indicates the services in applications.
    ADDITIONAL_SERVICES: "*"
    EXTRA_DEFAULT_SETTINGS: "log rsyslog local0,log global,option httplog"
  links:
    - rsyslog
rsyslog:
  image: registry.cn-hangzhou.aliyuncs.com/linhuatest/rsyslog:latest
  links:
    - rsyslog
appone:
  image: 'nginx:latest'
  labels:
    # http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
  ports:
    - 80/tcp
    - 443/tcp
  restart: always
```

Logs are printed directly to the standard output of the rsyslog container. The access logs of custom routing can be viewed by using `docker logs $rsyslog_container_name`. 
Server Load Balancer between services

The following template creates a Server Load Balancer service `lb` and an application service `appone` to provide services externally with the domain name `appone.example.com`.

```yaml
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  Hostname: proxy # Specify the domain name of the service as proxy, which is resolved to all containers with this image deployed.
  ports:
    - '80:80'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each VM.
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
  environment:
    # Indicates the range of backend containers that support route loading. "*" indicates the whole cluster. By default, it indicates the services in applications.
    ADDITIONAL_SERVICES: "*"

appone:
  ports:
    - 80/tcp
    - 443/tcp
  image: 'nginx:latest'
  labels:
    # http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
  restart: always
```

The following template is used as a client to access the `appone` application service, but the access path is used to request access to the Server Load Balancer service `lb` and then provide a reverse proxy for the `appone` application service.

```yaml
restclient: # Simulate rest service consumers.
  image: registry.aliyuncs.com/acs-sample/alpine:3.3
  command: "sh -c 'apk update; apk add curl; while true; do curl --head http://appone.example.com; sleep 1; done'" # Access the rest service and test Server Load Balancer.
  tty: true
  external_links:
    - "proxy:appone.example.com" # Specify the domain name of the link service and the alias of the domain name.
```

In the containers of the `restclient` service, the `appone.example.com` domain name is resolved to the IP addresses of all containers of the Server Load Balancer service `lb`.

```bash
/ # drill appone.example.com
```
Configure monitoring page

```yaml
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
    - '127.0.0.1:1935:1935' # The port that monitoring page exposes to the public network. Configure the port with due care because of the potential security risk.
  restart: always
  labels:
    aliyun.custom_addon: "proxy"
    aliyun.global: "true"
    aliyun.lb.port_80: tcp://proxy_test:80
  environment:
    ADDITIONAL_SERVICES: "*
    STATS_AUTH: "admin:admin" # The logon account and password used for monitoring, which are customizable.
    STATS_PORT: "1935" # The port used for monitoring, which is customizable.
appone:
  expose:
    - 80/tcp
  image: 'nginx:latest'
  labels:
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
  restart: always
```

Log on to each machine where the custom routing image resides (each machine can receive the request, no matter the application container is on which machine) and request the `acs/proxy` health check page.

Note:
Configure the correct username and password according to the environment variable STATS_AUTH of the application template.

root@c68a460635b8c405e83c052b7c2057c7b-node2:~# curl -Ss -u admin:admin 'http://127.0.0.1:1935/' &> test.html

Copy the page test.html to a machine with browsers and open the local file test.html with the browser. View the stats monitoring statistics page. Green indicates the network from container acs/proxy to backend containers is connected and the container acs/proxy is working normally. Other colors indicate an exception.

11.3 Custom routing - Supports TCP

When Alibaba Cloud Container Service is in use, the following problem may occur to TCP Server Load Balancer: when the client image and server image of an application are deployed on the same Elastic Compute Service (ECS) instance, the application client cannot access the local server by using Server Load Balancer due to the limitation of Server Load Balancer. In this document, take the common TCP-based Redis as an example to describe how to solve the problem by using the custom routing acs/proxy.

Note:
Different services cannot share the same Server Load Balancer instance. Otherwise, the backend machine of the Server Load Balancer is deleted and the services are unavailable.

Solution 1: Deploy client and server containers on different nodes by scheduling containers

The following is a sample application template (the label and swarm filter function are used):

```
redis-master:
  ports:
    - 6379:6379/tcp
  image: 'redis:alpine'
  labels:
    - aliyun.lb.port_6379: tcp://proxy_test:6379
redis-client:
  image: 'redis:alpine'
  links:
    - redis-master
  environment:
    - 'affinity:aliyun.lb.port_6379! =tcp://proxy_test:6379'
  command: redis-cli -h 120.25.131.64
  stdin_open: true
```
Note:

- Follow these steps if the scheduling does not take effect: Log on to the Container Service console. Click Swarm > Services in the left-side navigation pane. Select the cluster in which the service you want to reschedule resides from the Cluster drop-down list. Click Reschedule at the right of the service you want to reschedule. > Select the Force Reschedule check box in the displayed dialog box and then click OK.
- The volumes of existing containers will be lost if you select the Force Reschedule check box. Backup and migrate the data in advance.

Solution 2: Clients inside the container cluster access the server by using links, while clients outside access the server by using Server Load Balancer

The following is a sample application template (the `#unique_45` label is used):

```yaml
redis-master:
  ports:
    - 6379:6379/tcp
  image: 'redis:alpine'
  labels:
    aliyun.lb.port_6379: tcp://proxy_test:6379
redis-client:
  image: 'redis:alpine'
  links:
    - redis-master
  command: redis-cli -h redis-master
  stdin_open: true
  tty: true
```

Solution 3: Clients inside the container cluster access the server by using Custom routing (which is based on HAProxy and serves as a proxy server), while clients outside access the server by using Server Load Balancer

The following is a sample application template (the `#unique_45` label and Custom routing are used):

```yaml
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '6379:6379/tcp'
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each virtual machine (VM).
```

Issue: 20191028
This solution provides a master-slave Redis architecture and balances load by using the **Server Load Balancer between services** to make Container Server become highly available.
11.4 Custom routing - supports multiple HTTPS certificates

Use the `acs/proxy` image `#unique_72/unique_72_Connect_42_section_it1_v2z_xdb` in this example.

Note:

Services cannot use the same Server Load Balancer; otherwise, the backend machine of the Server Load Balancer will be deleted, and the service will be unavailable.

```yaml
lb:
  image: registry.aliyuncs.com/acs/proxy:0.6
  ports:
    - '80:80'
    - '443:443' # HTTPS must expose this port
  restart: always
  labels:
    # Addon allows the proxy image to function as a subscription registry center and dynamically load the service route.
    aliyun.custom_addon: "proxy"
    # A proxy image container is deployed on each virtual machine (VM).
    aliyun.global: "true"
    # A Server Load Balancer instance is bound to the frontend.
    aliyun.lb.port_80: tcp://proxy_test:80
    aliyun.lb.port_443: tcp://proxy_test:443
  environment:
    # Indicates the range of backend containers that support route loading. "*" indicates the whole cluster. By default, it indicates the services in applications.
    ADDITIONAL_SERVICES: "*"

appone:
  expose: # For proxied services, use expose or ports to tell proxy containers which port is to be exposed.
    - 80/tcp
  image: 'nginx:latest'
  labels:
    # You can specify paths when configuring URLs. In this example, http/https/ws/wss are supported.
    aliyun.proxy.VIRTUAL_HOST: "https://appone.example.com"
    # Configure the appone certificate.
    aliyun.proxy.SSL_CERT: "-----BEGIN RSA PRIVATE KEY-----
    ..."
User Guide / 11 Service discovery and load balancing

---BEGIN CERTIFICATE-----
MIIDvDCCAqSgAwIBAgIBATANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJDTjERMA8GA1UEC
BMWmhlamlhbmcxETAPBgNVBAcTCEhhbmdmda69IbMRQwEwYDVQQKEwthbGliYWJhL
mNvbTEvTEVMbMAGUXc5M3d3l3Njv3b3Yu929tMT4XKDIWotIAMA9MgEwRwYDVQQK
EwthbGliYWJhLmNvbTEvTEVMbMAGUXc5M3d3l3Njv3b3Yu929tMT4XKDIWotIAMA9Mg
-----END CERTIFICATE-----

```
.restart: always

apptwo:
  expose: # For proxied services, use expose or ports to tell proxy
  containers which port is to be exposed.
  - 80/tcp
  image: 'registry.cn-hangzhou.aliyuncs.com/linhuatest/hello-world:
latest'
  labels:
    # You can specify paths when configuring URLs. In this example,
    http/s/ww/s are supported.
    aliyun.proxy.VIRTUAL_HOST: "https://apptwo.example.com"

---BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----

```

---END RSA PRIVATE KEY-----

---BEGIN CERTIFICATE-----
MIIDvDCCAqSgAwIBAgIBATANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJDTjER
MA8GA1UEC
BMWmhlamlhbmcxETAPBgNVBAcTCEhhbmdmda69IbMRQwEwYDVQQKEwthbGliYWJhL
mNvbTEvTEVMbMAGUXc5M3d3l3Njv3b3Yu929tMT4XKDIWotIAMA9MgEwRwYDVQQK
EwthbGliYWJhLmNvbTEvTEVMbMAGUXc5M3d3l3Njv3b3Yu929tMT4XKDIWotIAMA9Mg
-----END CERTIFICATE-----

```

```
```

```
Services appone and apptwo use aliyun.proxy.VIRTUAL_HOST to specify the domain names. If you must configure the certificate, set the protocol to https. Then, use aliyun.proxy.SSL_CERT to specify the certificate content. The method of configuring the certificate content is as follows:

Assume that the key.pem is a private key file, and ca.pem is a public key file. Run the following commands (the current directory contains the public key file and private key file).

```
$ cp key.pem cert.pem
$ cat ca.pem >> cert.pem
$ awk 1 ORS='\n' cert.pem
```

Finally, enter the output of the awk command as the value of label aliyun.proxy.SSL_CERT.

SSL_CERT. Use double quotation marks (") for separation. For other information, such as lb label, #unique_45 see the preceding template and the corresponding Custom routing - simple sample.