

# Alibaba Cloud Container Service

## User Guide

Issue: 20190911

## Legal disclaimer

---

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

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

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








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





# Generic conventions

Table -1: Style conventions

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

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



# Contents

---

Legal disclaimer.....	I
Generic conventions.....	I
1 Kubernetes cluster.....	1
1.1 Overview.....	1
1.2 Alibaba Cloud Kubernetes vs. self-built Kubernetes.....	2
1.3 Clusters.....	5
1.3.1 Create a cluster.....	5
1.3.2 Access Kubernetes clusters by using SSH.....	15
1.3.3 Access Kubernetes clusters by using SSH key pairs.....	17
1.3.5 Add an existing ECS instance.....	18
1.3.6 Scale out or in a cluster.....	23
1.3.7 Upgrade a cluster.....	25
1.3.8 Delete a cluster.....	26
1.3.9 View cluster overview.....	28
1.4 Application Management.....	29
1.4.1 Create an application in Kubernetes dashboard.....	29
1.4.2 Create an application by using an image.....	33
1.4.3 Create an application by using an orchestration template.....	38
1.4.4 Simplify Kubernetes application deployment by using Helm.....	43
1.4.5 Manage applications by using commands.....	51
1.4.6 Create a service.....	52
1.4.7 Schedule a pod to a specified node.....	56
1.4.8 Service scaling.....	60
1.4.9 View services.....	62
1.4.10 Delete a service.....	63
1.4.11 View pods.....	64
1.4.12 Change container configurations.....	67
1.5 Namespaces.....	68
1.5.1 Create a namespace.....	68
1.5.2 Configure resource quotas for namespaces.....	70
1.5.3 Update a namespace.....	73
1.5.4 Delete a namespace.....	75
1.6 Config map.....	76
1.6.1 Create a config map.....	77
1.6.2 Use a config map in a pod.....	81
1.6.3 Update a config map.....	85
1.7 Secrets.....	89
1.7.1 Create a secret.....	89
1.7.2 View secret details.....	91
1.7.3 Update a secret.....	92

1.7.4 Delete a secret.....	93
1.8 Manage a release.....	94
1.9 App catalog.....	99
1.9.1 App catalog overview.....	99
1.9.2 View app catalog list.....	100
1.10 Plan Kubernetes CIDR blocks under VPC.....	101
1.11 Server Load Balancer.....	105
1.11.1 Overview.....	105
1.11.2 Access services by using Server Load Balancer.....	105
1.11.3 Configure Ingress monitoring.....	112
1.11.4 Support for Ingress.....	115
1.12 Storage.....	120
1.12.1 Overview.....	120
1.12.2 Use Alibaba Cloud cloud disks.....	121
1.12.3 Use Alibaba Cloud NAS.....	127
1.12.4 Use Alibaba Cloud OSS.....	135
1.13 Storage claim management.....	140
1.13.1 Create a persistent storage volume claim.....	140
1.13.2 Using persistent storage volume claim.....	143
1.14 Logs.....	146
1.14.1 Application log management.....	146
1.14.2 View cluster logs.....	146
1.14.3 Collect Kubernetes logs.....	147
1.14.4 Configure Log4jAppender for Kubernetes and Log Service.....	153
1.14.5 A solution to log collection problems of Kubernetes clusters by using log-pilot, Elasticsearch, and Kibana.....	158
1.15 Security.....	165
1.16 FAQ.....	167
1.16.1 Collect Kubernetes diagnosis information.....	167
1.16.2 FAQ about storage volumes.....	167
1.16.3 Failed to create a Kubernetes cluster.....	171
1.16.4 How to use private images in Kubernetes clusters.....	172
1.16.5 Upgrade Helm manually.....	173
<b>2 Authorizations.....</b>	<b>174</b>
2.1 Role authorization.....	174
2.2 Upgrade sub-account policy.....	178
2.3 Create custom authorization policies.....	181
<b>3 Clusters.....</b>	<b>185</b>
3.1 Cluster introduction.....	185
3.2 Cluster lifecycle.....	186
3.3 Create a cluster.....	187
3.4 Cluster parameter configurations.....	194
3.5 Add an existing ECS instance.....	198
3.6 Manage cross-zone nodes.....	203

3.7 Bind and unbind a Server Load Balancer instance.....	206
3.8 Set the root domain name of a cluster.....	208
3.9 Download cluster certificate.....	211
3.10 Expand a cluster.....	213
3.11 Migrate a cluster.....	214
3.12 Search for a cluster.....	215
3.13 Delete a cluster.....	216
3.14 Clean up a cluster disk.....	216
3.15 Log on to image repository.....	217
3.16 Upgrade Agent.....	219
3.17 Upgrade Docker daemon.....	220
3.18 Upgrade system services.....	222
<b>4 Nodes.....</b>	<b>224</b>
4.1 View containers running on a node.....	224
4.2 Update a node certificate.....	225
<b>5 Images and templates.....</b>	<b>227</b>
5.1 Update an orchestration template.....	227
<b>6 Service orchestrations.....</b>	<b>229</b>
6.1 routing.....	229
<b>7 Applications.....</b>	<b>232</b>
7.1 Create an application.....	232
7.2 Schedule an application to specified nodes.....	241
<b>8 Configurations.....</b>	<b>245</b>
8.1 Implement multiple environments by using configurations.....	245
<b>9 Data volumes.....</b>	<b>251</b>
9.1 Overview.....	251
9.2 Create an OSSFS data volume.....	251
9.3 Create cloud disk data volumes.....	255
9.4 View and delete data volumes.....	259
9.5 Use third-party data volumes.....	260
9.6 FAQ.....	265
<b>10 Logs.....</b>	<b>266</b>
10.1 Enable Log Service.....	266
<b>11 DevOps.....</b>	<b>271</b>
11.1 Jenkins-based continuous delivery.....	271
<b>12 Service discovery and load balancing.....</b>	<b>283</b>
12.1 Routing and Server Load Balancer between services in a cluster.....	283
12.2 Custom routing - simple sample.....	286
12.3 Custom routing - Supports TCP.....	297
12.4 Custom routing - supports multiple HTTPS certificates.....	300

# 1 Kubernetes cluster

---

## 1.1 Overview

Kubernetes is a popular open-source container orchestration technology. To allow you to use Kubernetes to manage container applications in Alibaba Cloud, Alibaba Cloud Container Service provides support for Kubernetes clusters.

You can create a safe and high-availability Kubernetes cluster in the Container Service console. The Kubernetes cluster integrates with the virtualization, storage, network, and security capabilities of Alibaba Cloud to provide scalable, high-performance container application management, simplify cluster creation and expansion, and focus on the development and management of containerized applications.

Kubernetes supports the deployment, expansion, and management of containerized applications, and provides the following features:

- Elastic expansion and self-reparation.
- Service discovery and server load balancing.
- Service release and rollback.
- Secret and configuration management.

### Limits

- Currently, Kubernetes clusters only support Linux containers. The support for Kubernetes Windows containers is in the works.
- Currently, Kubernetes clusters only support Virtual Private Cloud (VPC). You can select to create a VPC or use an existing VPC when creating a Kubernetes cluster.

### Related open-source projects

- Alibaba Cloud Kubernetes Cloud Provider: <https://github.com/AliyunContainerService/kubernetes>.
- Alibaba Cloud VPC network drive for Flannel: <https://github.com/coreos/flannel/blob/master/Documentation/alicloud-vpc-backend.md>.

If you have any questions or suggestions regarding a specific project, you are welcome to raise an issue or pull a request in the community.

## 1.2 Alibaba Cloud Kubernetes vs. self-built Kubernetes

### Advantages of Alibaba Cloud Kubernetes

#### Convenient

- Supports creating Kubernetes clusters with one click in the Container Service console.
- Supports upgrading Kubernetes clusters with one click in the Container Service console.

You may have to deal with self-built Kubernetes clusters of different versions at the same time, including version 1.8.6, 1.9.4, and 1.10 in the future. Upgrading clusters each time brings you great adjustments and Operation & Maintenance (O&M) costs. Container Service upgrade solution performs rolling update by using images and uses the backup policy of complete metadata, which allows you to conveniently roll back to the previous version.

- Supports expanding or contracting Kubernetes clusters conveniently in the Container Service console.

Container Service Kubernetes clusters allow you to expand or contract the capacity vertically with one click to respond to the peak of the data analysis business quickly.

#### Strong

Function	Description
Network	<ul style="list-style-type: none"><li>· High-performance Virtual Private Cloud (VPC) network plug-in.</li><li>· Supports network policy and flow control.</li></ul> <p>Container Service can provide you with continuous network integration and the best network optimization.</p>



Function	Description
Server Load Balancer	<p>Supports creating Internet or intranet Server Load Balancer instances.</p> <p>If your self-built Kubernetes clusters are implemented by using the self-built Ingress, publishing the business frequently may cause the Ingress to have pressure about configuration and higher error probabilities. The Server Load Balancer solution of Container Service supports Alibaba Cloud native high-availability Server Load Balancer, and can automatically modify and update the network configurations. This solution has been used by a large number of users for a long time, which is more stable and reliable than self-built Kubernetes.</p>
Storage	<p>Container Service integrates with Alibaba Cloud cloud disk, NAS, and EBS , and provides the standard FlexVolume drive.</p> <p>Self-built Kubernetes clusters cannot use the storage resources on the cloud . Alibaba Cloud Container Service provides the best seamless integration.</p>
O&M	<ul style="list-style-type: none"><li>· Integrates with Alibaba Cloud Log Service and CloudMonitor.</li><li>· Supports auto scaling.</li></ul>

Function	Description
Image repository	<ul style="list-style-type: none"><li>• High availability. Supports high concurrency.</li><li>• Supports speeding up the pull of images.</li><li>• Supports P2P distribution.</li></ul> <p>The self-built image repository may crash if you pull images from millions of clients at the same time. Enhance the reliability of the image repository by using the Apsara Stack version of Container Service image repository, which reduces the O&amp;M burden and upgrade pressure.</p>
Stable	<ul style="list-style-type: none"><li>• Special teams to guarantee the stability of containers.</li><li>• Each Linux version and Kubernetes version are provided to you after the strict test.</li></ul> <p>Container Service provides the Docker CE to reveal all the details and promotes the repair capabilities of Docker. If you have issues such as Docker Engine hang, network problems, and kernel compatibility, Container Service provides you with the best practices.</p>
High availability	<ul style="list-style-type: none"><li>• Supports multiple zones.</li><li>• Supports backup and disaster recovery.</li></ul>

Function	Description
Technical support	<ul style="list-style-type: none"><li>· Provides the Kubernetes upgrade capabilities. Supports upgrading a Kubernetes cluster to the latest version with one click.</li><li>· Alibaba Cloud container team is responsible for solving problems about containers in your environment.</li></ul>

### Costs and risks of self-built Kubernetes

- Building clusters is complicated

You must manually configure the components, configuration files, certificates, keys, plug-ins, and tools related to Kubernetes. It takes several days or weeks for professional personnel to build the cluster.

- For public cloud, it takes you significant costs to integrate with cloud products.

You must devote your own money to integrate with other products of Alibaba Cloud, such as Log Service, monitoring service, and storage management.

- The container is a systematic project, involving network, storage, operating system, orchestration, and other technologies, which requires the devotion of professional personnel.
- The container technology is continuously developing and the version iteration is fast, which requires continuous upgrade and test.

## 1.3 Clusters

### 1.3.1 Create a cluster

You can create a Kubernetes cluster quickly and easily in the Container Service console.

#### Instructions

During cluster creation, the Container Service performs the following operations:

- Create Elastic Compute Service (ECS) instances and configure to log on to other nodes from management nodes with the SSH public key. Install and configure the Kubernetes cluster by using CloudInit.

- Create a security group. This security group allows the Virtual Private Cloud (VPC) inbound access of all the ICMP ports.
- Create a new VPC and VSwitch if you do not use the existing VPC, and then create SNAT for the VSwitch.
- Create VPC routing rules.
- Create NAT gateway and Elastic IP (EIP).
- Create a Resource Access Management (RAM) user and the AccessKey. This RAM user has the permissions of querying, creating, and deleting ECS instances, adding and deleting cloud disks, and all the permissions of Server Load Balancer instances, CloudMonitor, VPC, Log Service, and NAS. Kubernetes clusters dynamically create the Server Load Balancer instances, cloud disks, and VPC routing rules according to your configurations.
- Create an intranet Server Load Balancer instance and expose the port 6443.
- Create an Internet Server Load Balancer instance and expose the ports 6443, 8443, and 22. (If you select to enable the SSH login for Internet when creating the cluster, port 22 is exposed. Otherwise, port 22 is not exposed.)

### Prerequisites

Activate the following services: Container Service, Resource Orchestration Service (ROS), and RAM.

Log on to the [Container Service console](#), [ROS console](#), and [RAM console](#) to activate the corresponding services.



#### Note:

The deployment of Container Service Kubernetes clusters depends on the application deployment capabilities of Alibaba Cloud ROS. Therefore, activate ROS before creating a Kubernetes cluster.

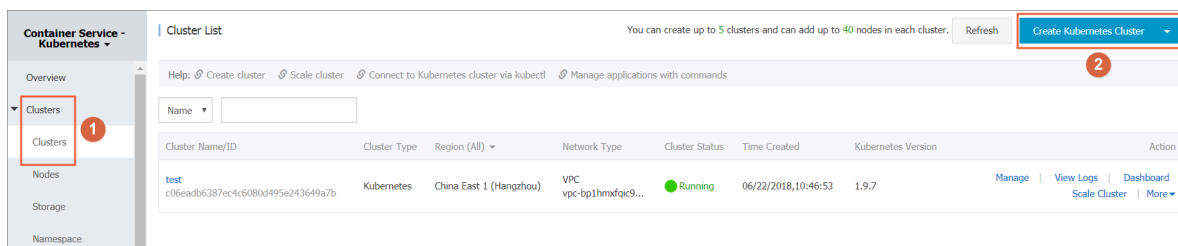
### Limits

- The Server Load Balancer instance created with the cluster only supports the Pay-As-You-Go billing method.
- Kubernetes clusters only support the network type VPC.

- 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 40 worker nodes to each cluster. To create more clusters or nodes, open a ticket. To create more clusters or nodes, open a ticket.
- 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.
- Limits for ECS instances are as follows:
  - Only support the CentOS operating system.
  - Creating Pay-As-You-Go and Subscription ECS instances is supported.

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane to enter the Cluster List page.
3. Click Create Kubernetes Cluster in the upper-right corner.



4. Enter the cluster name.

The cluster name can be 1–63 characters long and contain numbers, Chinese characters, English letters, and hyphens (-).

5. Select the region and zone in which the cluster resides.



6. Set the cluster network type. Kubernetes clusters only support the VPC network type.

You can select **Auto Create** to create a Virtual Private Cloud (VPC) together with the Kubernetes cluster or **Use existing** to use an existing VPC. With **Use Existing** selected, choose the VPC and VSwitch from the appeared drop-down list.

- With **Auto Create** selected, the system automatically creates a NAT gateway for your VPC when the cluster is created.
- With **Use Existing** selected, if the selected VPC already has a NAT gateway, Container Service uses the existing NAT gateway. Otherwise, the system automatically creates a NAT gateway by default. If you do not want the system to automatically create a NAT gateway, clear the **Configure SNAT for VPC** check box.



**Note:**

If you select to not automatically create a NAT gateway, configure the NAT gateway on your own to implement the VPC public network environment with secure access, or manually configure the SNAT. Otherwise, instances in the VPC cannot access public network normally, which leads to cluster creation failure.

VPC

Auto Create Use Existing

VPC123 (vpc-2zercq4pyanzsxficlyl) VSwitch123 (vsw-2zeydhl5uwh1ej522lauo) ZoneA

7. Configure the node type, Pay-As-You-Go and Subscription types are supported.

8. Configure the master nodes.

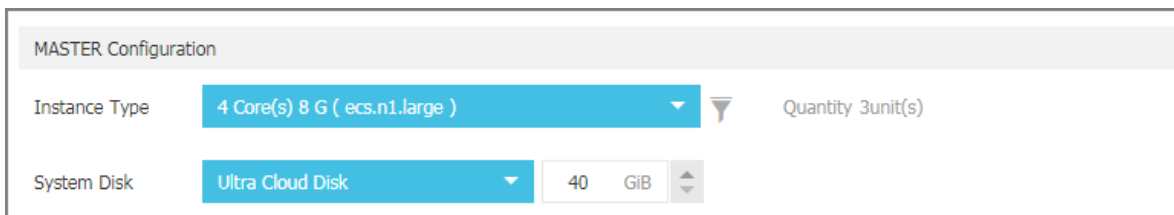
Select the generation, family, and type for the master nodes.



**Note:**

- Currently, master nodes only support CentOS operating system.
- Currently, you can only create three master nodes.

- Supports mounting system disks for the master node, SSD and high-efficiency cloud disks are supported.



MASTER Configuration

Instance Type **4 Core(s) 8 G ( ecs.n1.large )** Quantity 3 unit(s)

System Disk **Ultra Cloud Disk** 40 GiB

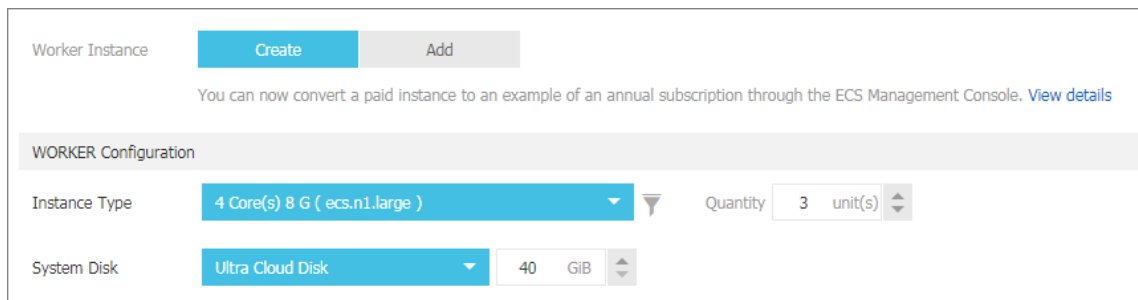
9. Configure the worker nodes. Select whether to create a worker node or add an existing ECS instance as the worker node.



**Note:**

- Currently, worker nodes only support the CentOS operating system.
- Each cluster can contain up to 37 worker nodes. To create more nodes, open a ticket.
- Supports mounting system disks for the worker node, SSD, high-efficiency, and basic cloud disks are supported.

- a. If you want to add an instance, you must generation, family, and type for the worker node., and number for the worker nodes (in this example, select to create one worker node).



Worker Instance **Create** **Add**

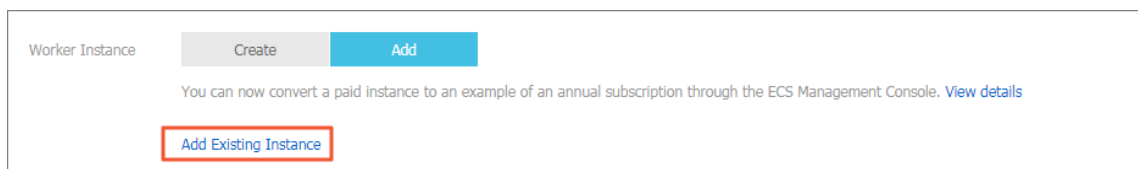
You can now convert a paid instance to an example of an annual subscription through the ECS Management Console. [View details](#)

WORKER Configuration

Instance Type **4 Core(s) 8 G ( ecs.n1.large )** Quantity **3** unit(s)

System Disk **Ultra Cloud Disk** 40 GiB

- b. To add an existing ECS instance as the worker node, you must create an ECS instance in the current region in advance.



Worker Instance **Create** **Add**

You can now convert a paid instance to an example of an annual subscription through the ECS Management Console. [View details](#)

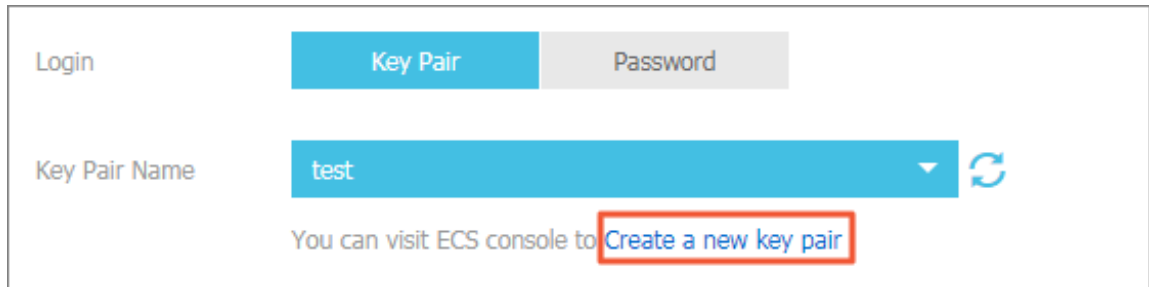
**Add Existing Instance**

## 10. Configure the logon mode.

- Set the secret.

Select the key pair logon mode when creating the cluster, click New Key Pair.

Go to the ECS console, and create a key pair, see [#unique\\_8](#). After the key pair is created, set the key pair as the credentials for logging on to the cluster.



- Set the password.
  - Logon Password: Configure the node logon password.
  - Confirm Password: Confirm your node logon password.

## 11. Configure the Pod Network CIDR and Service CIDR.



Note:

This option is available when you select to use an existing VPC.

Specify the Pod Network CIDR and Service CIDR. Both of them cannot overlap with the Classless Inter-Domain Routing (CIDR) block used by VPC and the existing Kubernetes clusters in VPC, and you cannot modify the values after the cluster is created. Service address segment cannot be repeated with the Pod address segment. Besides, the service CIDR block cannot overlap with the pod CIDR block. For more information about how to plan the Kubernetes CIDR blocks, see [#unique\\_9](#).

## 12. Set whether to configure a SNAT gateway for a private network.



Note:

SNAT must be configured if you select Auto Create VPC. If you select Use existing VPC, you can select whether to automatically configure SNAT gateway. If you select not to configure SNAT automatically, you can configure the NAT gateway



to implement VPC security access to the public network. You can also configure SNAT manually. Otherwise, the VPC cannot access the public network.

Configure SNAT ☒ Configure SNAT for VPC

If the VPC you choose does not have access to Internet, NAT gateway and EIP will be used to configure SNAT for the VPC. During this period, NAT gateway, EIP, and other resources may be created.

### 13. Select whether to enable SSH login for Internet.

- With this check box selected, you can access the cluster by using SSH.
- If this check box is not selected, you cannot access the cluster by using SSH or connect to the cluster by using kubectl. To access the cluster by using SSH, manually bind EIP to the ECS instance, configure security group rules, and open the SSH port (22). For more information, see [#unique\\_10](#).

SSH Login ☐ Enable SSH access for Internet

If you choose not to open it, please refer to [SSH access to Kubernetes cluster](#) to manually enable SSH access.

### 14. Sets whether the cloud monitoring plug-in is enabled.

You can select to install the cloud monitoring plug-in on the ECS instance and then view the monitoring information of the created ECS instance in the CloudMonitor console.

Monitoring Plug-in ☒ Install cloud monitoring plug-in on your ECS.

Installing a cloud monitoring plug-in on the node allows you to view the monitoring information of the created ECS instance in the CloudMonitor console

### 15. Select to add the IP addresses of the ECS instances to the RDS instance whitelist.

It facilitates the ECS instances to access the RDS instances.



#### Note:

This option is available if you are using an existing VPC. The ECS instance must be in the same region and same VPC environment 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.
- b. The Add to RDS instance whitelist dialog box appears. Select the RDS instances and then click OK.

## 16. Select whether to enable the advanced configurations.

- a. Enable the network plug-ins, Flannel and Terway network plug-ins are supported.
  - Flannel: The Flannel cni plug-in for simple and stable communities.
  - Terway: Alibaba Cloud Container Service self-developed network plug-in, which supports Alibaba Cloud flexible network card to be distributed to the container, and supports Kubernetes NetworkPolicy to define the inter-container access policy. Supports bandwidth limiting for the separate containers. Currently it is in the public beta.
- b. Set the number of nodes pod, which is the maximum number of pods that can be run by a single node. We recommend to maintain the default value.

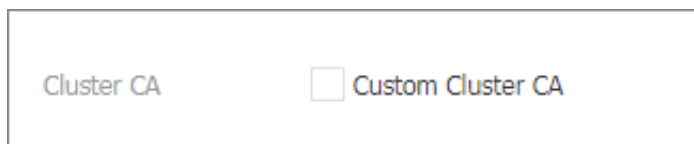


Pod Number for Node 128 ▼

- c. Select whether or not to use the custom image. The ECS instance installs the default CentOS version if no custom image is selected.

Currently, you can only select an image based on CentOS to deploy the environment you need quickly. For example, the image deployed and tested based on the CentOS 7.2 LAMP.

- d. Select whether to use custom cluster CA. With this check box selected, the CA certificate can be added to the Kubernetes cluster, which enhances the security of information exchange between server and client.



Cluster CA ☐ Custom Cluster CA

## 17. Click Create cluster to start the deployment.



### Note:

Creating a Kubernetes cluster with multiple nodes lasts more than 10 minutes.

## Subsequent operations

After the cluster is successfully created, you can view the cluster in the Kubernetes Cluster List of the Container Service console.

Container Service - Kubernetes		Cluster List						You can create up to 5 clusters and can add up to 40 nodes in each cluster.		Refresh	Create Kubernetes Cluster
Overview		Help: Create cluster Scale cluster Connect to Kubernetes cluster via kubectl Manage applications with commands									
Clusters		Name <input type="text"/>									
Clusters											
Nodes											
Storage											
		Cluster Name/ID	Cluster Type	Region (All)	Network Type	Cluster Status	Time Created	Kubernetes Version	Action		
		test c06eadb6387ec4c6080d495e243649a7b	Kubernetes	China East 1 (Hangzhou)	VPC vpc-b0c1tme4gq9...	Running	06/22/2018,10:46:53	1.9.7	Manage	View Logs	Dashboard Scale Cluster More

Click View Logs at the right of the cluster to view the cluster logs. To view more detailed information, click Stack Events.

Detailed resource deployment logs: Stack Events	
Time	Information
06/22/2018,11:13:50	c06eadb6387ec4c6080d495e243649a7b   Start to DescribeK8sUserCertConfig
06/22/2018,11:03:39	c06eadb6387ec4c6080d495e243649a7b   Set up k8s DNS configuration successfully
06/22/2018,11:02:30	c06eadb6387ec4c6080d495e243649a7b   Stack CREATE completed successfully:o
06/22/2018,11:02:30	c06eadb6387ec4c6080d495e243649a7b   Start describeStackInfo
06/22/2018,11:02:29	c06eadb6387ec4c6080d495e243649a7b   Start describeStackInfo
06/22/2018,10:46:55	c06eadb6387ec4c6080d495e243649a7b   Successfully to CreateStack
06/22/2018,10:46:55	c06eadb6387ec4c6080d495e243649a7b   Start to wait stack ready
06/22/2018,10:46:53	c06eadb6387ec4c6080d495e243649a7b   Start to create cluster task
06/22/2018,10:46:53	c06eadb6387ec4c6080d495e243649a7b   Start to CreateK8sCluster
06/22/2018,10:46:53	c06eadb6387ec4c6080d495e243649a7b   Start to CreateStack
06/22/2018,10:46:50	c06eadb6387ec4c6080d495e243649a7b   Start to validateCIDR
06/22/2018,10:46:41	c06eadb6387ec4c6080d495e243649a7b   Start create cluster certificate

You can also click Manage at the right of the cluster to view the basic information and connection information of this cluster.

Basic Information	
Cluster ID: c8b6e9f6b3872e34e5880c9495a343b46a7b	VPC
	Running
	Region: China East 1 (Hangzhou)
Connection Information	
API Server Internet endpoint	https://193.37.114.5:6443
API Server Intranet endpoint	https://193.188.1.227:6443
Master node SSH IP address	204.207.114.5
Service Access Domain	*c8b6e9f6b3872e34e5880c9495a343b46a7b.cn-hangzhou.alicontainer.com
Cluster resource	
ROS	k8s-for-cs-c8b6e9f6b3872e34e5880c9495a343b46a7b
Internet SLB	lb-ludg0e585d6c7planorj
VPC	vpc-bp0t4emofar0p4emw02aefu
NAT Gateway	ngw-bp0t4emw02aefu
<b>Connect to Kubernetes cluster via kubectl</b> 1. Download the latest kubectl client from the <a href="#">Kubernetes Edition page</a> . 2. Install and set up the kubectl client. For more information, see <a href="#">Installing and Setting Up kubectl</a> 3. Configure the cluster credentials: <div> <span>KubeConfig</span> <span>SSH</span> </div>	

### In the Connection Information section:

- **API Server Internet endpoint:** The address and port used by the Kubernetes API server to provide the service for the Internet. You can use kubectl or other tools on the user terminal by means of this service to manage the cluster.
- **API Server Intranet endpoint:** The address and port used by the Kubernetes API server to provide the service for the intranet. This IP address is the address of the Server Load Balancer instance, and three master nodes in the backend are providing the service.
- **Master node SSH IP address:** You can directly log on to the master nodes by using SSH to perform routine maintenance for the cluster.
- **Service Access Domain:** Provides the service in the cluster with access domain name for testing. The suffix of the service access domain name is `< cluster_id >.< region_id >.alicontainer.com`.

For example, you can log on to the master nodes by using SSH, and run the `kubectl get node` to view the node information of the cluster.

```

login as: root
root@iZbp1d7yvpa3j183u0ur11Z's password:

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iZbp1d7yvpa3j183u0ur11Z ~]# kubectl get node
NAME                                STATUS    ROLES    AGE      VERSION
cn-hangzhou.i-04b6e7f30d0bca4e7a1156879ed384e44    Ready    <none>   17m      v1.8.4
cn-hangzhou.i-04b6e7f30d0bca4e7a1156879ed384e44    Ready    master   19m      v1.8.4
cn-hangzhou.i-04b6e7f30d0bca4e7a1156879ed384e44    Ready    master   24m      v1.8.4
cn-hangzhou.i-04b6e7f30d0bca4e7a1156879ed384e44    Ready    master   22m      v1.8.4
[root@iZbp1d7yvpa3j183u0ur11Z ~]#

```

As shown in the preceding figure, the cluster has four nodes, including three master nodes and one worker node configured when creating the cluster.

### 1.3.2 Access Kubernetes clusters by using SSH

If you select not to enable SSH access for Internet when creating the Kubernetes cluster, you cannot access the Kubernetes cluster by using SSH or connect to the Kubernetes cluster by using `kubectl`. To access the cluster by using SSH after creating the cluster, manually bind Elastic IP (EIP) to the Elastic Compute Service (ECS) instance, configure security group rules, and open the SSH port (22).

#### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster.
4. In Cluster resource, click the ID of the Internet SLB. Then, you are redirected to the Instance Details page of your Internet Server Load Balancer instance.

Basic Information			
Cluster ID: c0b6e7f30d0bca4e7a1156879ed384e44	VPC	●Running	Region: China East 1 (Hangzhou)
Connection Information			
API Server Internet endpoint	<a href="https://47.87.228.5:443">https://47.87.228.5:443</a>		
API Server Intranet endpoint	<a href="https://192.168.223.178:443">https://192.168.223.178:443</a>		
Master node SSH IP address	47.87.228.5		
Service Access Domain	*c0b6e7f30d0bca4e7a1156879ed384e44.cn-hangzhou.alicontainer.com		
Cluster resource			
ROS	<a href="#">ali-ros-cs-c0b6e7f30d0bca4e7a1156879ed384e44</a>		
Internet SLB	<a href="#">lb-2ackoev0l0oww49v4gm3</a>		
VPC	<a href="#">vpc-6gcmrta2gplu1rwmvka</a>		
NAT Gateway	<a href="#">nagw-4p12frt1haz17v0me0d0c</a>		

5. Click Listeners in the left-side navigation pane and then click Add Listener in the upper-right corner.

6. Add the SSH listening rule.
  - a. Front-end Protocol [Port]: Select TCP and enter 22.
  - b. Backend Protocol [Port]: Enter 22.
  - c. Turn on the Use Server Group switch and select VServer Group.
  - d. Server Group ID: Select sshVirtualGroup.
  - e. Click Next and then click Confirm to create the listener.

The screenshot shows a configuration window for an SSH listening rule. The interface is divided into several sections with labels and input fields. The 'Front-end Protocol [Port]:\*' section has a dropdown menu set to 'TCP' and a text box containing '22'. Below it, a note states 'Port range is 1-65535.'. The 'Backend Protocol [Port]:\*' section also has a dropdown menu set to 'TCP' and a text box containing '22', with the same 'Port range is 1-65535.' note. The 'Peak Bandwidth:' section shows 'No Limits' with a 'Configure' link and a note: 'Instances charged by traffic are not limited by peak bandwidth. Peak bandwidth range is 1-5000.'. The 'Scheduling Algorithm:' section has a dropdown menu set to 'Weighted f'. The 'Use Server Group:' section has a green toggle switch turned on. The 'Server Group Type:' section has two radio buttons: 'VServer Group' (selected) and 'Master-Slave Server Group'. The 'Server Group ID:' section has a dropdown menu set to 'sshVirtualGn'. The 'Automatically Enable Listener After Creation:' section has a green toggle switch turned on and the word 'Enable'. At the bottom left, there is a checkbox labeled 'Show Advanced Options'. At the bottom right, there are two buttons: 'Next' (blue) and 'Cancel' (gray).

Front-end Protocol [Port]:*	TCP	:	22
Port range is 1-65535.			
Backend Protocol [Port]:*	TCP	:	22
Port range is 1-65535.			
Peak Bandwidth:	No Limits <a href="#">Configure</a>		
Instances charged by traffic are not limited by peak bandwidth. Peak bandwidth range is 1-5000.			
Scheduling Algorithm:	Weighted f		
Use Server Group:	<input checked="" type="checkbox"/>		
Server Group Type:	<input checked="" type="radio"/> VServer Group <input type="radio"/> Master-Slave Server Group		
Server Group ID:	sshVirtualGn		
Automatically Enable Listener After Creation:	<input checked="" type="checkbox"/> Enable		
<input type="checkbox"/> <a href="#">Show Advanced Options</a>			

[Next](#) [Cancel](#)

7. Then, you can use the Server Load Balancer instance IP address to access your cluster by using SSH.

Basic Information	
Server Load Balancer ID: <a href="#">lb-12345678901234567890</a>	Status: <span style="color: green;">● Running</span>
Server Load Balancer Name: <a href="#">KubernetesCluster...</a>	Region: China East 1 (Hangzhou)
Instance IP Type: Public IP	Zone: cn-hangzhou-b(Master)/cn-hangzhou-d(Slave)
Network Type: Classic Network	
Billing Information	
Billing Method: Pay by Traffic	Created At: 2018-01-24 11:13:01
Instance IP Address: <a href="#">114.55.188.25</a> (Public IP)	Automatic Release Time: -

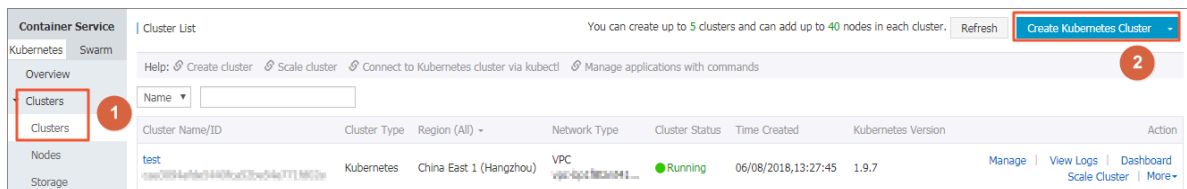
### 1.3.3 Access Kubernetes clusters by using SSH key pairs

Alibaba Cloud Container Service allows you to log on to clusters by using SSH key pairs, which guarantees the security of SSH remote access.

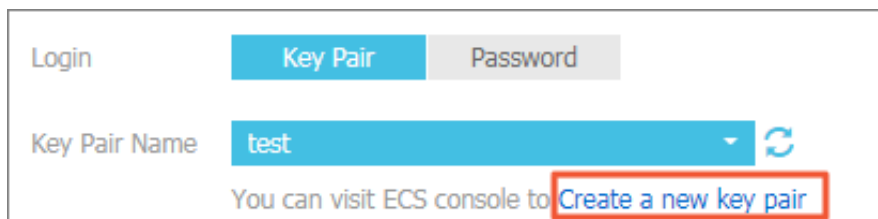
#### Context

#### Procedure

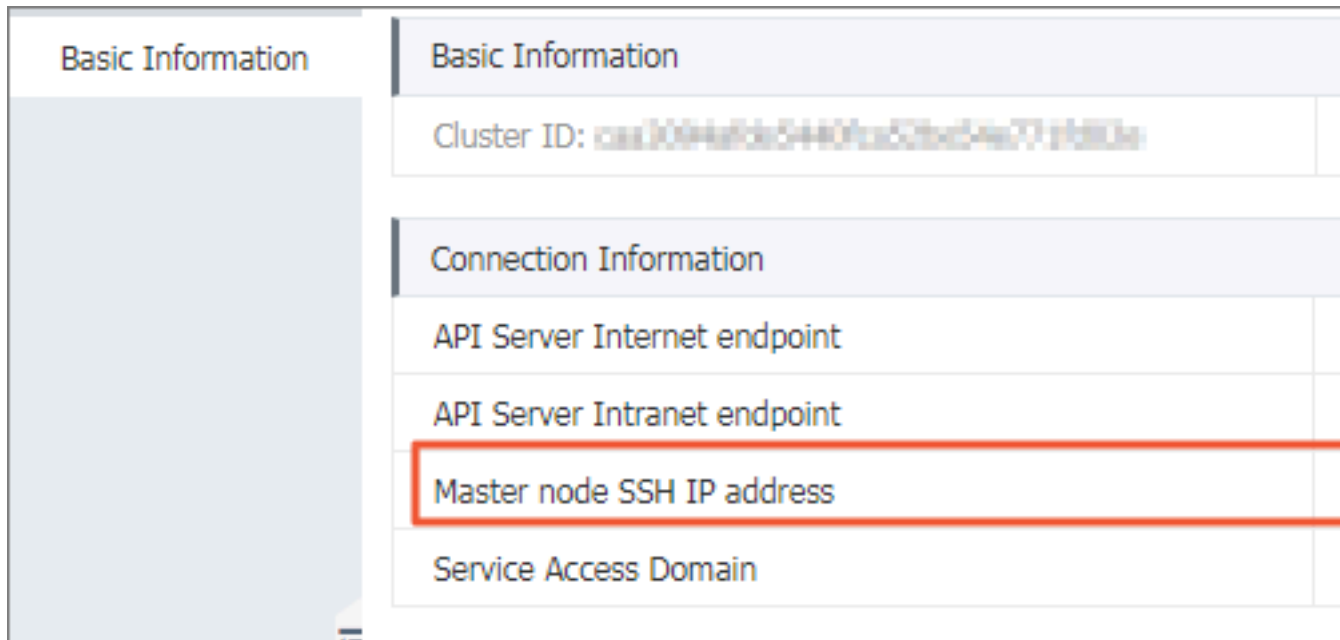
1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Click Create Kubernetes Cluster in the upper-right corner.



4. Select Key Pair in the Login field. Complete the other configurations. For more information, see [#unique\\_13](#). Then, click Create.
  - a. If you have created key pairs in the Elastic Compute Service (ECS) console, select a key pair from the Key Pair Name drop-down list.
  - b. If you have no key pair, click Create a new key pair to create one in the ECS console. For more information, see [#unique\\_8](#).



5. After the cluster is created, click Manage at the right of the cluster on the Cluster List page. View the Master node SSH IP address under Connection Information.



6. Download the `.pem` private key file. Complete the configurations based on your local operating system environment, such as Windows or Linux. For more information, see [#unique\\_14](#). Take Linux as an example.

- a) Find the path where your downloaded `.pem` private key file is stored on your local machine. For example, `/ root / xxx . pem`.

- b) Run the following command to modify the attributes of the private key file:

```
chmod 400 [ path where the .pem private key file
is stored on the local machine ]. For example, chmod 400
/ root / xxx . pem .
```

- c) Run the following command to connect to the cluster: `` ssh - i [ path where the .pem private key file is stored on the local machine ] root @[ master - public - ip ]`. Wherein, master-public-ip is the master node SSH IP address. For example, `ssh - i / root / xxx . pem root @ 10 . 10 . 10 . 100`.

### 1.3.5 Add an existing ECS instance

You can add existing Elastic Compute Service (ECS) instances to a created Kubernetes cluster. Currently, Kubernetes clusters only support adding worker nodes.

#### Prerequisites



- If you have not created a cluster before, create a cluster first. For how to create a cluster, see [#unique\\_13](#).
- Add the ECS instance to the security group of the Kubernetes cluster first.

## Context

### Instructions

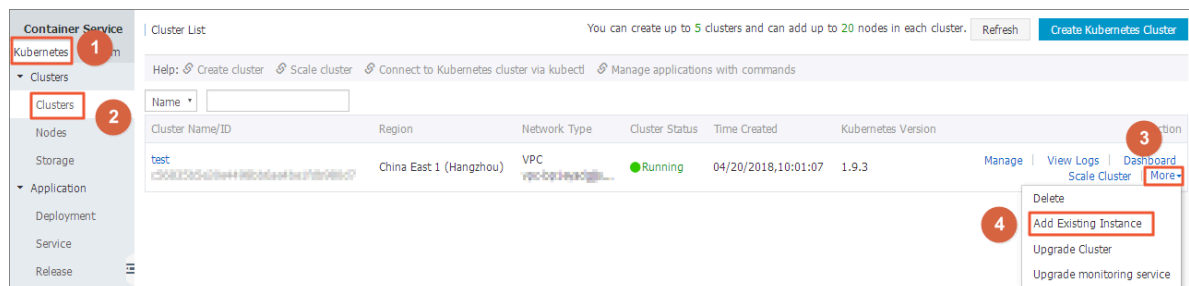
- By default, each cluster can contain up to 40 nodes. To add more nodes, open a ticket.
- The ECS instance to be added must be in the same Virtual Private Cloud (VPC) region as the cluster.
- When adding an existing instance, make sure that your instance has an Elastic IP (EIP) for the VPC network type, or the corresponding VPC is already configured with 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.
- Only nodes with a CentOS operating system are supported.

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Select the target cluster and click More > Add Existing Instance.

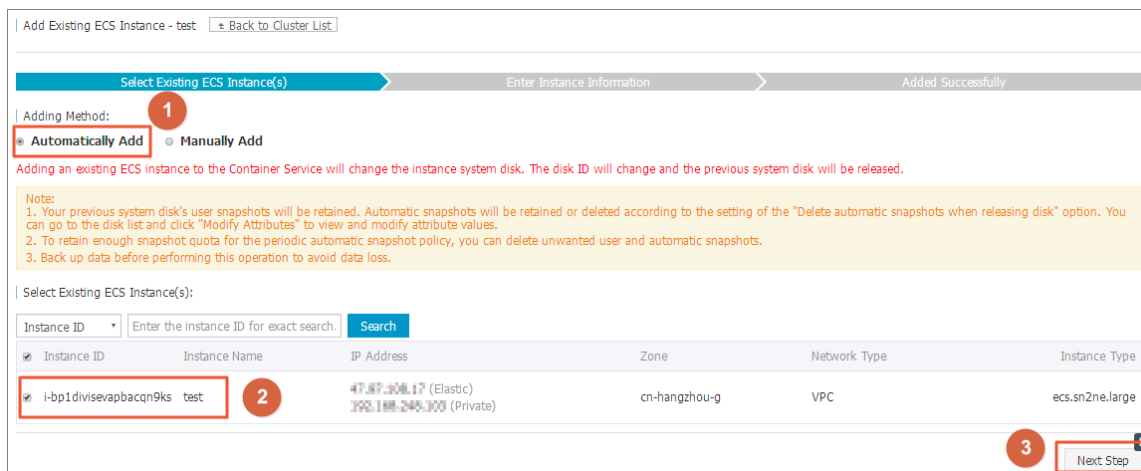
On the Add Existing ECS Instance page and you can automatically or manually add an existing instance.

If Automatically Add is selected, select the ECS instances to add them to the cluster automatically. If Manually Add is selected, you must obtain the command and then log on to the corresponding ECS instance to add the ECS instance to this cluster. You can only add one ECS instance at a time.



#### 4. Select Automatically Add to add multiple ECS instances at a time.

- a) In the list of existing cloud servers, select the target ECS instance, and then click Next Step.



Add Existing ECS Instance - test [Back to Cluster List](#)

Select Existing ECS Instance(s) Enter Instance Information Added Successfully

Adding Method: **1**

☒ Automatically Add ☐ Manually Add

Adding an existing ECS instance to the Container Service will change the instance system disk. The disk ID will change and the previous system disk will be released.

Note:

1. Your previous system disk's user snapshots will be retained. Automatic snapshots will be retained or deleted according to the setting of the "Delete automatic snapshots when releasing disk" option. You can go to the disk list and click "Modify Attributes" to view and modify attribute values.
2. To retain enough snapshot quota for the periodic automatic snapshot policy, you can delete unwanted user and automatic snapshots.
3. Back up data before performing this operation to avoid data loss.

Select Existing ECS Instance(s):

Instance ID  Enter the instance ID for exact search. [Search](#)

Instance ID	Instance Name	IP Address	Zone	Network Type	Instance Type
<input checked="" type="checkbox"/> i-bp1divisevapbacqn9ks test <b>2</b>		47.87.106.17 (Elastic) 192.168.245.100 (Private)	cn-hangzhou-g	VPC	ecs.sn2ne.large

**3** [Next Step](#)

- b) Enter the instance information, set the logon password, and then click Next Step.



选择已有云服务器实例 填写实例信息 添加完成

集群ID/名称:  / test-gpu

当前要添加的集群信息

登录方式: ☒ 设置密码

\* 密码:

密码为8-30个字符，必须同时包含三项（大、小写字母，数字和特殊符号），不支持`两个符号

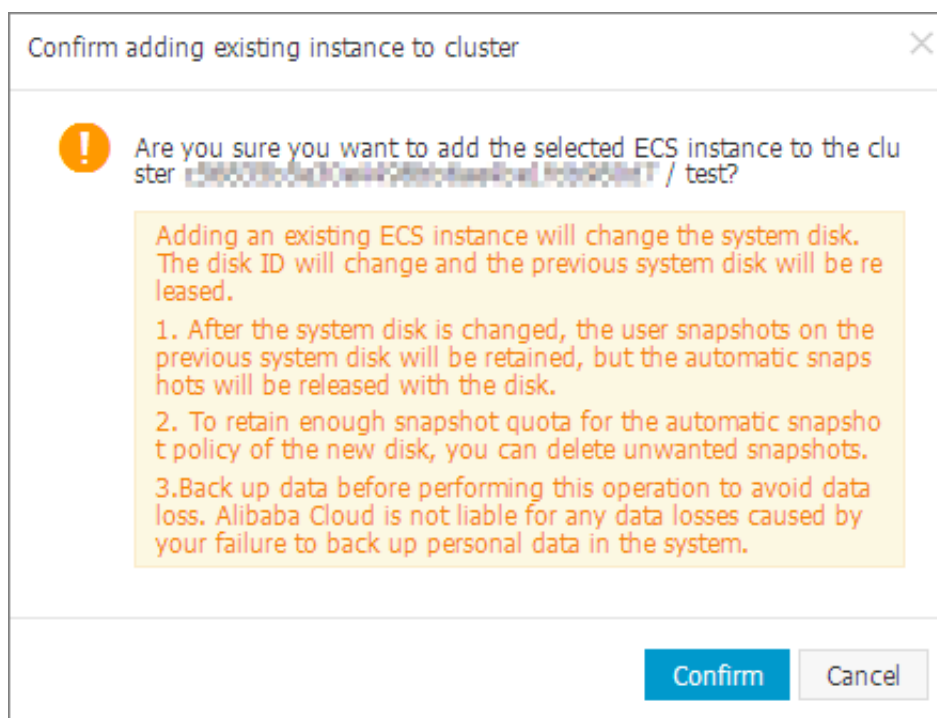
\* 确认密码:

实例信息:

实例ID	实例名称
<input checked="" type="checkbox"/> i-bp1divisevapbacqn9ks test	test
<input type="checkbox"/> i-bp1divisevapbacqn9ks shukun-ECS	shukun-ECS

[上一步](#) **3** [下一步](#)

- c) In the displayed dialog box, click OK, the selected ECS instance is automatically added to the cluster.



5. You can also select **Manually Add** to manually add an existing ECS instance to the cluster.

a) Select the ECS instance to be added and then click **Next Step**. You can only add one ECS instance at a time.

Add Existing ECS Instance - test [← Back to Cluster List](#)

Select Existing ECS Instance(s) Enter Instance Information Added Successfully

Adding Method:  
☐ Automatically Add ☒ **Manually Add** 1

Select Existing ECS Instance(s):

To manually add existing nodes, you can only select one ECS instance at a time.

Instance ID  Enter the instance ID for exact search.

<input checked="" type="checkbox"/>	Instance ID	Instance Name	IP Address	Zone	Network Type	Instance Type
<input checked="" type="checkbox"/>	i-bp1divisevapbacqn9ks	test	192.168.1.100 (Elastic) 192.168.1.100 (Private)	cn-hangzhou-g	VPC	ecs.sn2ne.large

3

b) Confirm the information and then click **Next Step**.

Add Existing ECS Instance - test [← Back to Cluster List](#)

Select Existing ECS Instance(s) Enter Instance Information Added Successfully

Cluster ID/Name : **cn-hangzhou-k8s-17090911 / test**  
 Information of the cluster to which to add the ECS instance(s).

Instance Information :

Instance ID	Instance Name
i-bp1divisevapbacqn9ks	test

c) Go to the **Add Existing ECS Instance** page and copy the command.

Add Existing ECS Instance - test [← Back to Cluster List](#)

Select Existing ECS Instance(s) Enter Instance Information Added Successfully

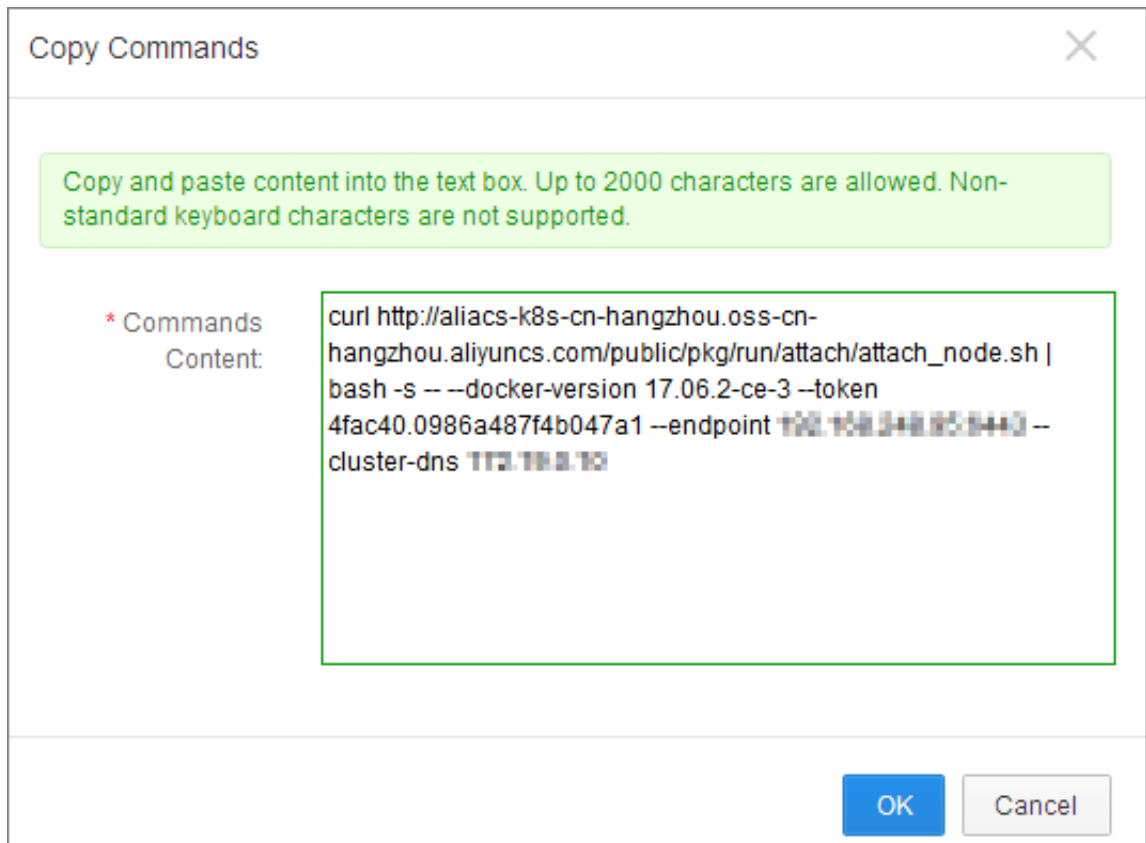
Only supports adding nodes in the same VPC with CentOS operating system

Log in to the node you want to add, execute the following command:

```
curl http://aliacs-k8s-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/public/pkg/run/attach/attach_node.sh | bash -s -- --docker-version 17.06.2-c
e-3 --token 4fac40.0986a487f4b047a1 --endpoint 192.168.1.100 --cluster-dns 172.17.0.1
```

d) Log on to the [ECS console](#). Select the region in which the cluster resides.

e) Click **Connect** at the right of the ECS instance to be added. The **Enter VNC Password** dialog box appears. Enter the VNC password and then click **OK**. Enter the copied command and then click **OK** to run the script.



- f) After the script is successfully run, the ECS instance is added to the cluster.
- You can click the cluster ID on the Cluster List page to view the node list of the cluster and check if the ECS instance is successfully added to the cluster.

### 1.3.6 Scale out or in a cluster

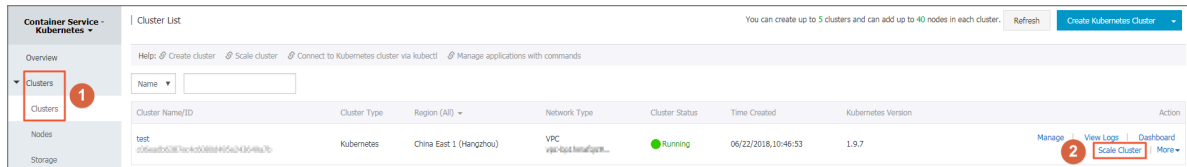
In the Container Service console, you can scale out or scale in the worker nodes of a Kubernetes cluster according to your actual business requirements.

#### Context

- Currently, Container Service does not support scaling in and out the master nodes in a cluster.
- Container Service only supports scaling in the worker nodes that are created when you create the cluster or added after you scale out the cluster. The worker nodes that are added as existing [#unique\\_17](#) when you create the cluster cannot be scaled in.
- When you scale in a cluster, the worker nodes are removed from the cluster in the order that they are added after you scale out the cluster.
- You must have more than 1 node that is not manually added to perform scaling in.

#### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Click Scale Cluster at the right of the cluster.



4. Select Scale out or Scale in in the Scale field and then configure the number of worker nodes.

In this example, scale out the cluster to change the number of worker nodes from one to four.

5. Enter the logon password of the node.



#### Note:

Make sure this password is the same as the one you entered when creating the cluster because you have to log on to the Elastic Compute Service (ECS) instance to copy the configuration information in the upgrade process.

6. Click Submit.

### What's next

After scaling is complete, go to the Kubernetes Clusters Node List page to view that the number of worker nodes changes from one to four.

### 1.3.7 Upgrade a cluster

You can upgrade the Kubernetes version of your cluster in the Container Service console.

View the Kubernetes version of your cluster in the Kubernetes cluster list.

Name	Cluster Name/ID	Region	Network Type	Cluster Status	Time Created	Kubernetes Version	Action
test	cs-20180424095828-1e503e07ad1154e44	China East 1 (Hangzhou)	VPC vpc-20180424095828-1e503e07ad1154e44	Running	04/24/2018,09:58:28	1.9.3	Manage   View Logs   Dashboard Scale Cluster   More

#### Instructions

- To upgrade the cluster, make sure your machine can access the Internet to download the necessary software packages.
- The upgrade may fail. We recommend that you back up snapshots before upgrading the cluster to guarantee your data security. For how to create a snapshot, see [#unique\\_19](#).
- During the upgrade, your applications are not affected, but we recommend that you do not manage the cluster by using kubectl or the Container Service console. The upgrade lasts 5–15 minutes. The cluster status changes to Running after the upgrade.

#### Prerequisites

Check the health status of the cluster before upgrading the cluster. Make sure the cluster is healthy.

Log on to the master node. For more information, see [#unique\\_10](#) and [#unique\\_20](#).

1. Run the command `kubectl get cs`. Make sure all the modules are healthy.

```
NAME      STATUS    MESSAGE    ERROR
scheduler Healthy    ok
controller-manager Healthy    ok
etcd - 0   Healthy    {"health": "true"}
etcd - 1   Healthy    {"health": "true"}
etcd - 2   Healthy    {"health": "true"}
```

2. Run the command `kubectl get nodes`. Make sure all the nodes are in the Ready status.

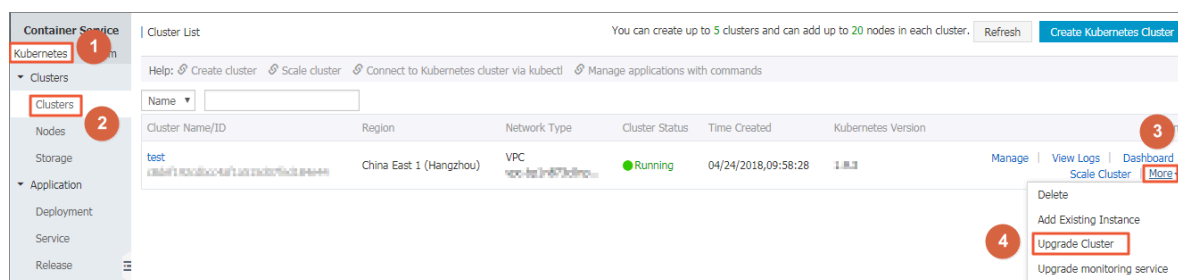
```
kubectl get nodes
NAME                STATUS    ROLES    AGE    VERSION
cn-shanghai-1       Ready     master   38d    v1.9.3
cn-shanghai-2       Ready     master   38d    v1.9.3
cn-shanghai-3       Ready     master   38d    v1.9.3
cn-shanghai-4       Ready     master   38d    v1.9.3
cn-shanghai-5       Ready     master   38d    v1.9.3
```

```
cn - shanghai . i - xxxxxx    Ready    master    38d    v1 . 9 . 3
```

If nodes are abnormal, you can fix them by yourself or open a ticket to ask Alibaba Cloud engineers to fix them.

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Click More > Upgrade Cluster at the right of the cluster.



4. Click Upgrade in the displayed dialog box.

The system starts to upgrade the Kubernetes version.

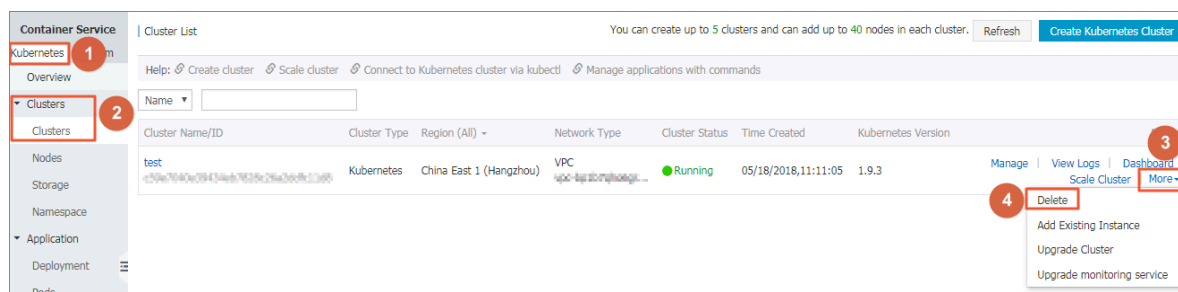
After the upgrade, you can check the Kubernetes version of this cluster in the Kubernetes cluster list to make sure whether or not the upgrade is successful.

## 1.3.8 Delete a cluster

In the Container Service console, you can delete clusters that are no longer in use.

## Procedure

1. Log on to the [Container Service console](#).
2. Click Kubernetes > Clusters in the left-side navigation pane.
3. Click More at the right of the cluster and then select > Delete.



## What's next

### Failed to delete a cluster



If you manually add some resources under the resources created by Resource Orchestration Service (ROS), ROS does not have permissions to delete these manually added resources. For example, manually add a VSwitch under the Virtual Private Cloud (VPC) created by ROS. ROS fails to process this VPC when deleting the Kubernetes resources and then the cluster fails to be deleted.

Container Service allows you to force delete the cluster. You can force delete the cluster record and ROS stack if the cluster fails to be deleted. However, you must release the manually created resources manually.

The cluster status is Failed to delete if the cluster fails to be deleted.

Cluster List

You can create up to 5 clusters and can add up to 40 nodes in each cluster.

Refresh

Create Kubernetes Cluster

Help: [Create cluster](#) [Scale cluster](#) [Connect to Kubernetes cluster via kubectl](#) [Manage applications with commands](#)

Name

Cluster Name/ID	Cluster Type	Region (All)	Network Type	Cluster Status	Time Created	Kubernetes Version	Action
<a href="#">test</a>	Kubernetes	China East 1 (Hangzhou)	VPC	Failed to delete	05/18/2018,11:11:05	1.9.3	<a href="#">Manage</a>   <a href="#">View Logs</a>   <a href="#">Dashboard</a> <a href="#">Scale Cluster</a>   <a href="#">More</a>

Click More at the right of the cluster and then select > Delete. In the displayed dialog box, you can see the resources that failed to be deleted. Select the Force Delete check box and then click OK to delete the cluster and ROS stack.



#### Note:

You must manually release the resources that failed to be deleted. To find these resources, see [#unique\\_22](#).

Delete Cluster - test

Are you sure to delete the cluster test ?

☒ **Force Delete** Delete the cluster record and stack only, you need to manually release the following resources

Resource ID	Resource Type	Status	Updated At
vpc-l2t3rjhaagj...	ALIYUN::ECS::VPC	Delete Failed	2018-05-18 13:07:24

OK Cancel

### 1.3.9 View cluster overview

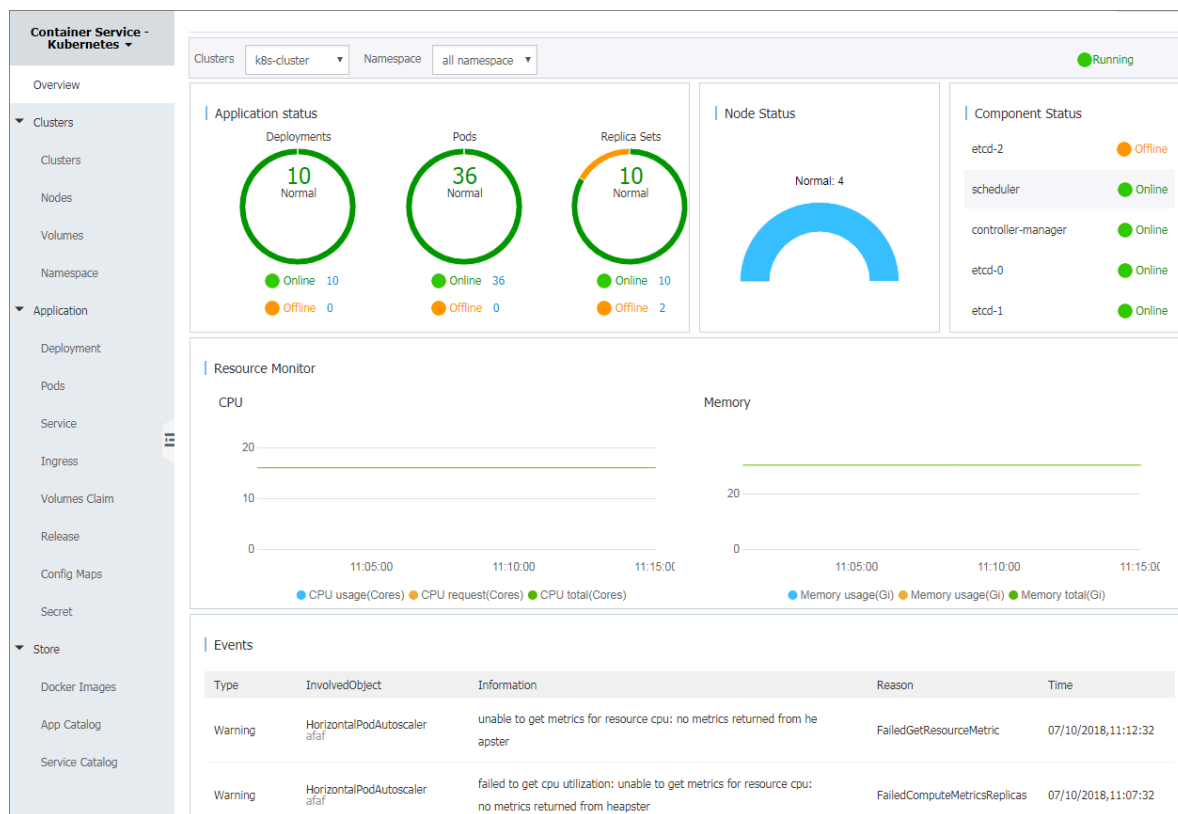
You can view the application status, component status, and resource monitoring charts on the Overview page of Alibaba Cloud Container Service Kubernetes clusters, which allows you to quickly understand the health status of clusters.

#### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Overview in the left navigation bar to enter the Kubernetes cluster overview page.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. You can view the application status, component status, and resource monitoring charts.
  - **Application status:** The status of deployments, pods, and replica sets that are currently running. Green indicates the normal status and orange indicates an exception.
  - **Node status:** Displays the node status of the current cluster.
  - **Component status:** The components of Kubernetes clusters are generally deployed under the kube-system namespace, including the core components such as scheduler, controller-manager, and etcd.
  - **Resource monitor:** Provides the monitoring charts of CPU and memory. CPU is measured in cores and is accurate to three decimal places. The minimum unit is millicores, that is, one thousandth of one core. Memory is measured in G and

is accurate to three decimal places. For more information, see [Meaning of CPU](#) and [Meaning of memory](#).

- **Event:** Displays event information of the cluster, such as warnings and error events.



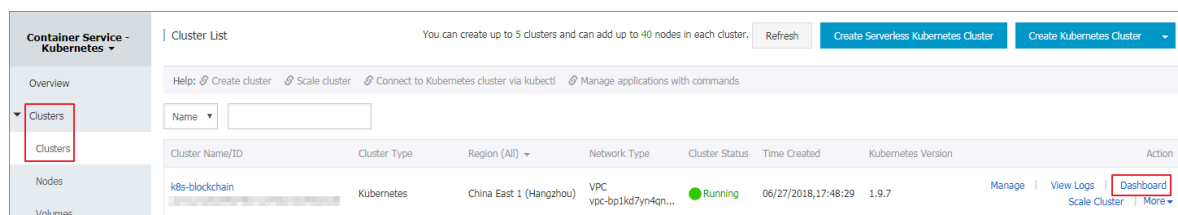
## 1.4 Application Management

### 1.4.1 Create an application in Kubernetes dashboard

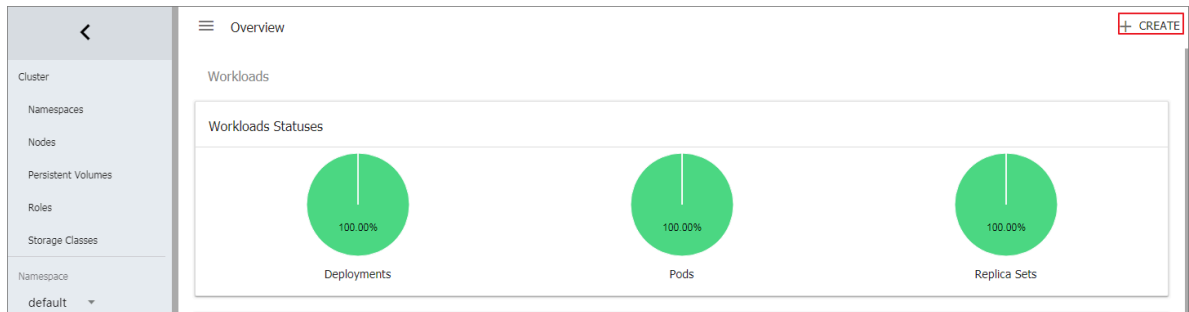
You can create an application in the Kubernetes dashboard.

#### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Clusters** in the left-side navigation pane.
3. Click **Dashboard** at the right of the cluster to enter the Kubernetes dashboard.



4. In the Kubernetes dashboard, click **CREATE** in the upper-right corner to create an application.



## 5. The Resource creation page appears. Configure the application information.

Create an application in any of the following three ways:

- **CREATE FROM TEXT INPUT:** Directly enter the orchestration codes in the YAML or JSON format to create an application. You must know the corresponding orchestration format.

Resource creation + CREATE

CREATE FROM TEXT INPUT CREATE FROM FILE CREATE AN APP

Enter YAML or JSON content specifying the resources to deploy to the currently selected namespace. [Learn more](#)

```

1 apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
2 kind: Deployment
3 metadata:
4   name: nginx-deployment-basic
5   labels:
6     app: nginx
7 spec:
8   replicas: 2
9   selector:
10    matchLabels:
11      app: nginx
12 template:
13   metadata:
14     labels:
15       app: nginx
16   spec:
17     # nodeSelector:
18     #   env: test-team
19     containers:
20     - name: nginx
21       image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
  
```

UPLOAD CANCEL

- **CREATE AN APP:** Complete the following configurations to create an application.
  - **App name:** Enter the name of the application you are about to create. In this example, enter `nginx - test`.
  - **Container image:** Enter the URL of the image to be used. In this example, use Docker [Nginx](#).
  - **Number of pods:** Configure the number of pods for this application.
  - **Service:** Select External or Internal. External indicates to create a service that can be accessed from outside the cluster. Internal indicates to create a service that can be accessed from within the cluster.
  - **Advanced options:** To configure the information such as labels and environment variables, click SHOW ADVANCED OPTIONS. This configuration distributes the traffic load evenly to three pods.

CREATE FROM TEXT INPUT    CREATE FROM FILE    **CREATE AN APP**

App name \*  
nginx-test 10 / 24  
An 'app' label with this value will be added to the Deployment and Service that get deployed. [Learn more](#)

Container Image \*  
nginx  
Enter the URL of a public image on any registry, or a private image hosted on Docker Hub or Google Container Registry. [Learn more](#)

Number of pods \*  
3  
A Deployment will be created to maintain the desired number of pods across your cluster. [Learn more](#)

Service \*  
None  
Optionally, an internal or external Service can be defined to map an incoming Port to a target Port seen by the container. The internal DNS name for this Service will be: nginx-test. [Learn more](#)

[SHOW ADVANCED OPTIONS](#)

**DEPLOY**    CANCEL


- **CREATE FROM FILE:** Upload an existing YAML or JSON configuration file to create an application.


6. Click **UPLOAD** or **DEPLOY** to deploy the containers and services.

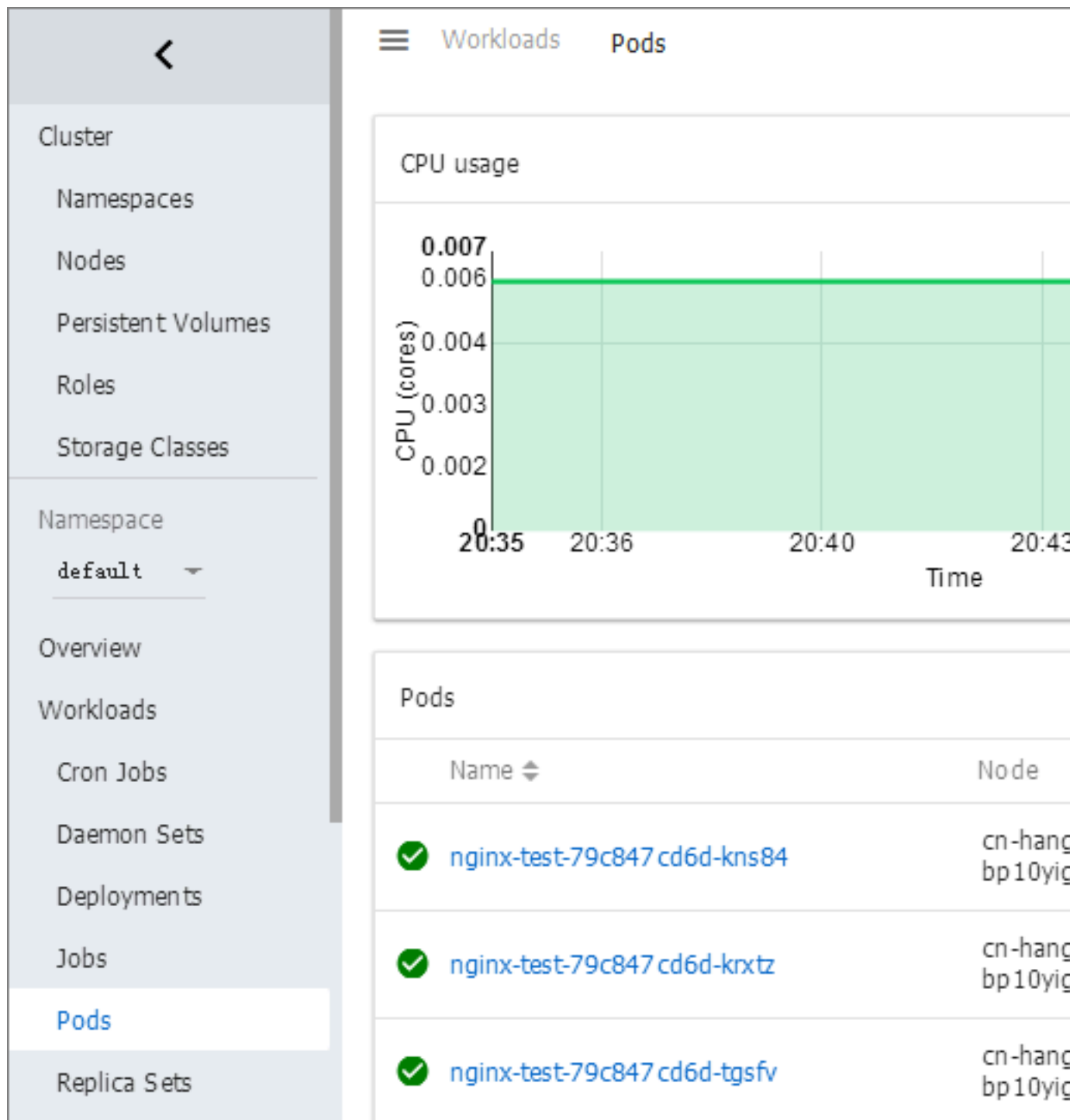
You can also click **SHOW ADVANCED OPTIONS** to configure more parameters.

### What's next

After clicking **UPLOAD** or **DEPLOY**, you can view the services and containers of the application.

Click **Pods** in the left-side navigation pane. You can check the status of each Kubernetes object according to the icon on the left.  indicates the object is still

being deployed.  indicates the object has completed the deployment.



## 1.4.2 Create an application by using an image

### Prerequisites

Create a Kubernetes cluster. For more information, see [#unique\\_13](#).

### Procedure

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

2. Click **Kubernetes > Application > > Deployment** in the left-side navigation pane. Click **Create by image** in the upper-right corner.
3. Enter the application Name, and then select the Cluster and Namespace. Click **Next** to go to the Configuration step.

By default, the system uses the default namespace if the namespace is not configured.

4. Configure the general settings for the application.

- **Image Name:** You can click **Select image** to select the image in the displayed dialog box and then click **OK**. In this example, the image name is `nginx`.

You can also enter the private registry in the format of `domainname / namespace / imagename : tag`.

- **Image Version:** Click **Select image version** to select the image version. If the image version is not specified, the system uses the latest version by default.
- **Scale:** Specify the number of containers. In this example, only one container is in the pod. If multiple containers are specified, the same number of pods will be started.

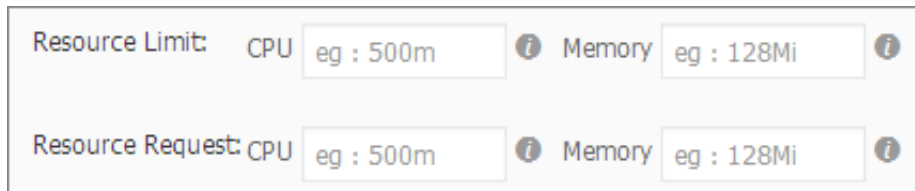
5. Configure the resource limit and resource reserve for the container.

- **Resource Limit:** Specify the upper limit for the resources (CPU and memory) that can be used by this application to avoid occupying excessive resources.
- **Resource Request:** Specify how many resources (CPU and memory) are reserved for the application, that is, these resources are exclusive to the container. Other services or processes will compete for resources when the resources



are insufficient. By specifying the Resource Request, the application will not become unavailable because of insufficient resources.

CPU is measured in millicores (one thousandth of one core). Memory is measured in bytes, which can be Gi, Mi, or Ki.

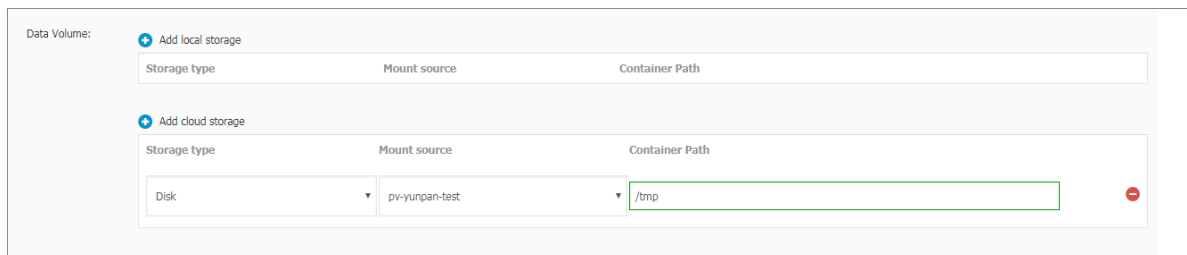


The screenshot shows two rows of configuration fields. The first row is labeled 'Resource Limit:' and contains two input fields: 'CPU' with the value 'eg : 500m' and 'Memory' with the value 'eg : 128Mi'. The second row is labeled 'Resource Request:' and also contains two input fields: 'CPU' with the value 'eg : 500m' and 'Memory' with the value 'eg : 128Mi'. Each input field has an information icon (i) to its right.

## 6. Configure the data volumes.

The hostPath data volumes can be configured. The hostPath data volumes mount the files or directories in the host file system to the pod. For more information, see hostPath in [volumes](#).

In this example, configure a hostPath data volume named data. Use the host directory `/tmp` and mount the data volume data to the `var / lib / docker` directory in the container.



The screenshot shows the 'Data Volume' configuration interface. It has two sections: 'Add local storage' and 'Add cloud storage'. The 'Add cloud storage' section is active and shows a table with three columns: 'Storage type', 'Mount source', and 'Container Path'. The 'Storage type' is set to 'Disk', the 'Mount source' is 'pv-yunpan-test', and the 'Container Path' is '/tmp'. There is a red minus icon in the bottom right corner of the table.

## 7. Configure the environment variable.

You can configure the environment variable for the pod in the format of key-value pairs to add the environment label or pass the configurations for the pod. For more information, see [Pod variable](#).

## 8. Configure the container.

You can configure the Command, Args, and Container Config for the container running in the pod.

- **Command and Args:** If not configured, the default settings of the image are used. If configured, the default settings of the image are overwritten. If only the Args is configured, the default command will run the new arguments when

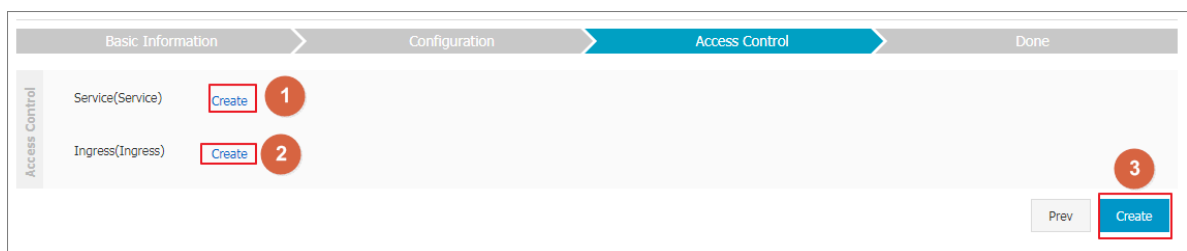
the container is started. Command and Args cannot be modified after the pod is created.

- **Container Config:** Select the stdin check box to enable standard input for the container. Select the tty check box to assign a virtual terminal to send signals to the container. These two options are usually used together, which indicates to bind the terminal (tty) to the container standard input (stdin). For example, an interactive program obtains standard input from you and then displays the obtained standard input in the terminal.

#### 9. Select whether or not to enable the Auto Scaling.

To meet the demands of applications under different loads, Container Service supports the container auto scaling, which automatically adjusts the number of containers according to the container CPU and memory usage.

#### 10. Click Next after completing the configurations. In the Access Control step, configure a service to bind with the backend pods.



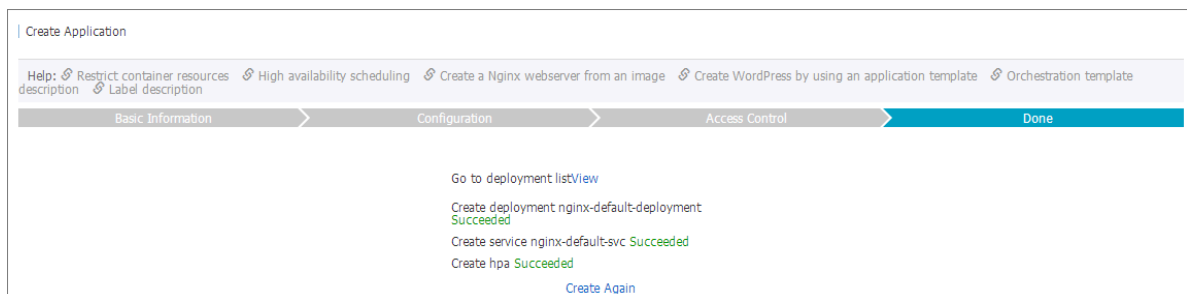
- **Service:** Select None to not create a service, or select a service type as follows:
  - **ClusterIP:** Exposes the service by using the internal IP address of your cluster. With this type selected, the service is only accessible from within the cluster.
  - **NodePort:** Exposes the service by using the IP address and static port (NodePort) on each node. A ClusterIP service, to which the NodePort service is routed, is automatically created. You can access the NodePort service from outside the cluster by requesting `< NodeIP > : < NodePort >`.
  - **Server Load Balancer:** Exposes the service by using Server Load Balancer, which is provided by Alibaba Cloud. Select public or inner to access the

service by using the Internet or intranet. Server Load Balancer can route to the NodePort and ClusterIP services.

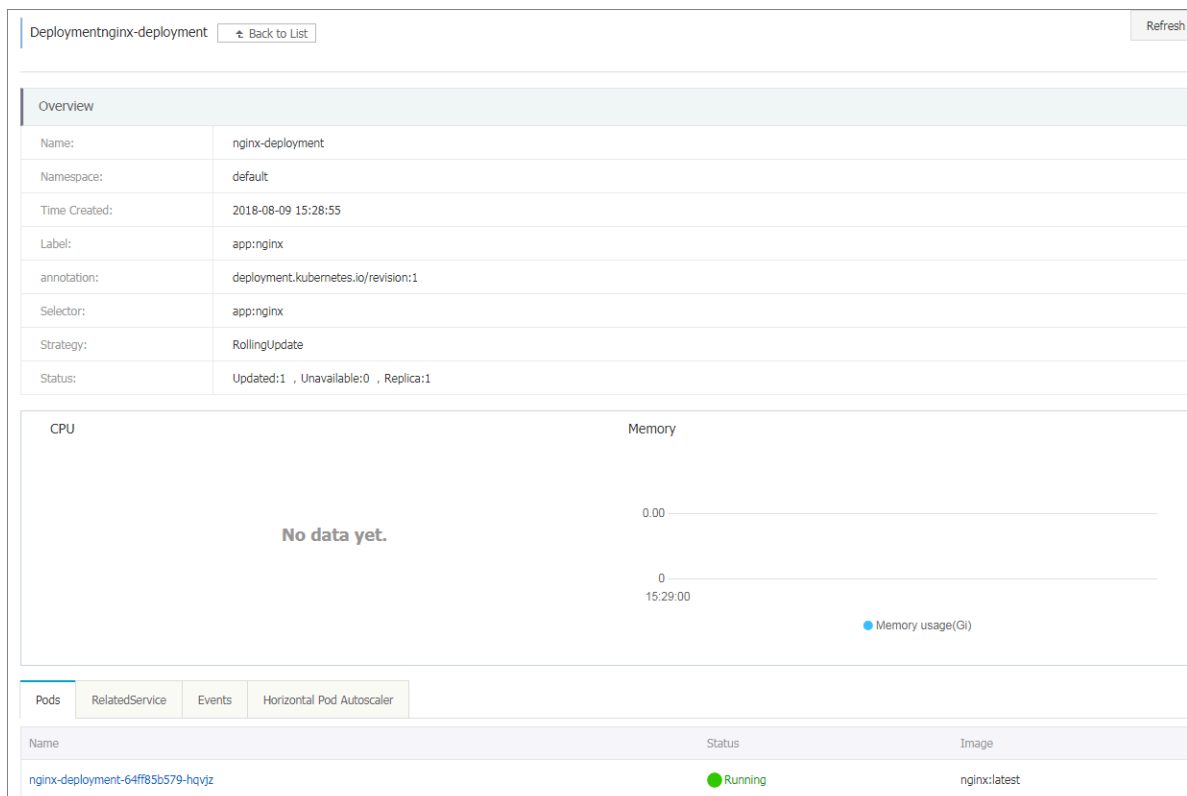
- **Name:** By default, a service name composed of the application name and the suffix svc is generated. In this example, the generated service name is nginx-default-svc. You can modify the service name as needed.
- **Port Mapping:** You must add the service port and the container port. If NodePort is selected as the service type, you must configure the node port to avoid the port conflict. Select TCP or UDP as the Protocol.

11. Click Create after the access control configurations.

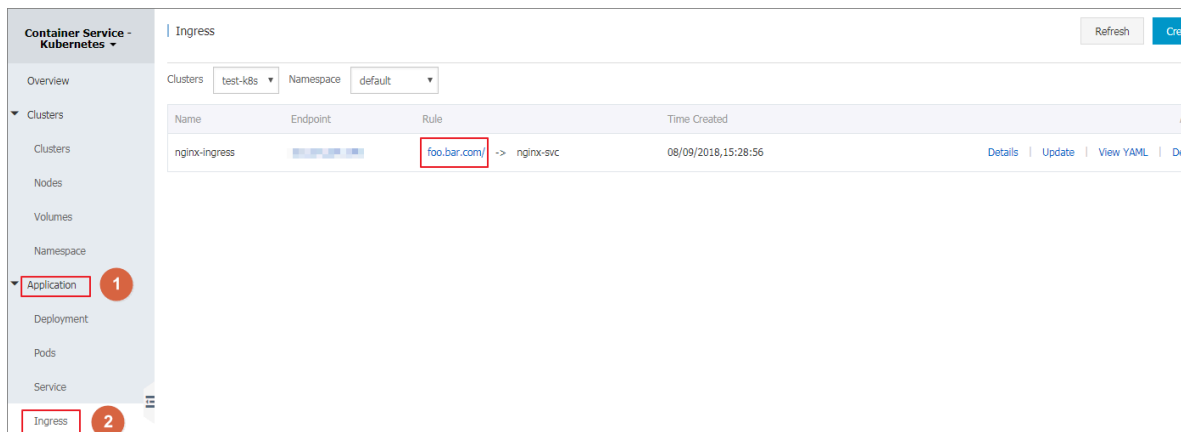
The Done step indicating the successful creation appears. The objects contained in the application are displayed. You can click View to view the deployment list.



12. The newly created deployment nginx-default-deployment is displayed on the Deployment page.



13. Click **Application > > Service** in the left-side navigation pane. The newly created service **nginx-default-svc** is displayed on the Service List page.



14. Access the external endpoint in the browser to access the Nginx welcome page.



### 1.4.3 Create an application by using an orchestration template

#### Prerequisites

Create a Kubernetes cluster. For more information, see [#unique\\_13](#).

#### Context

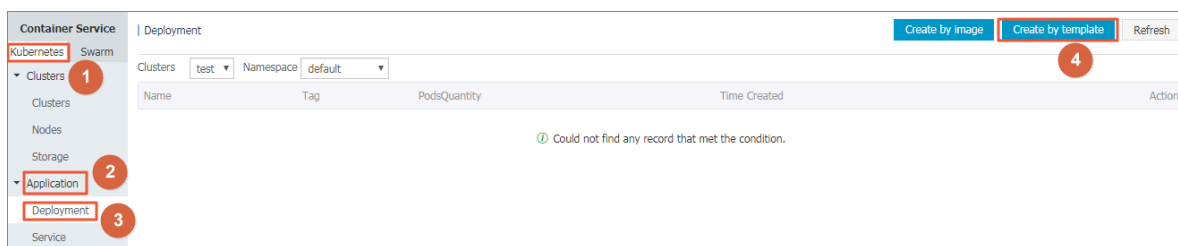
In the Container Service Kubernetes orchestration template, you must define a resource object required for running an application, and combine the resource objects into a complete application by using label selector.

Create an Nginx application in this example. Firstly, create a backend pod resource object by creating the deployment. Then, deploy the service to bind it to the backend pod, forming a complete Nginx application.

#### Procedure

1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Application > > Deployment** in the left-side navigation pane.

### 3. Click Create by template in the upper-right corner.



### 4. Configure the template and then click DEPLOY.

- **Clusters:** Select the cluster in which the resource object is to be deployed.
- **Namespace:** Select the namespace to which the resource object belongs. default is selected by default. Except for the underlying computing resources such as nodes and persistent storage volumes, most of the resource objects must act on a namespace.
- **Resource Type:** Alibaba Cloud Container Service provides Kubernetes YAML sample templates of many resource types for you to deploy resource objects quickly. You can write your own template based on the format requirements

of Kubernetes YAML orchestration to describe the resource type you want to define.

Deploy templates

Only Kubernetes versions 1.8.4 and above are supported. For clusters of version 1.8.1, you can perform "upgrade cluster" operation in the cluster list

Clusters: test
Namespace: default
Resource Type: Resource - basic Deployment

1

Template

```

1 apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
2 kind: Deployment
3 metadata:
4   name: nginx-deployment-basic
5   labels:
6     app: nginx
7 spec:
8   replicas: 2
9   selector:
10    matchLabels:
11      app: nginx
12   template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       # nodeSelector:
18       # env: test-team
19     containers:
20     - name: nginx
21       image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
22       ports:
23     - containerPort: 80

```

2

DEPLOY

The deployment sample orchestration of an Nginx application is as follows. By using this orchestration template, you can create a deployment that belongs to an Nginx application quickly.

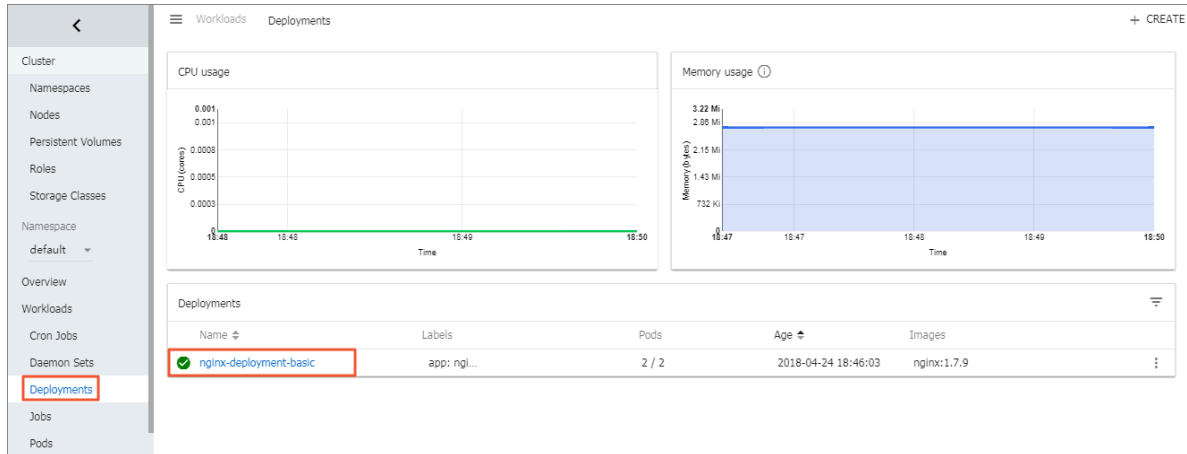
```

apiVersion : apps / v1beta2 # for versions before 1 . 8 .
0 use apps / v1beta1
kind : Deployment
metadata :
  name : nginx - deployment
  labels :
    app : nginx
spec :
  replicas : 2
  selector :
    matchLabel s :
      app : nginx
  template :
    metadata :
      labels :
        app : nginx
    spec :
      containers :
        - name : nginx
          image : nginx : 1 . 7 . 9 # replace it with your
          exactly < image_name : tags >
          ports :

```

```
- containerPort : 80
```

- After you click **DEPLOY**, a message indicating the deployment status is displayed. After the successful deployment, click **Kubernetes Dashboard** in the message to go to the dashboard and check the deployment progress.



- Go back to the Deploy templates page and deploy a service resource object.

Container Service provides a service sample template of an Nginx application. In this example, modify the template a little by changing the access type to LoadBalancer, and then you can create a service bound to the backend pod, which allows you to access the service in the browser.



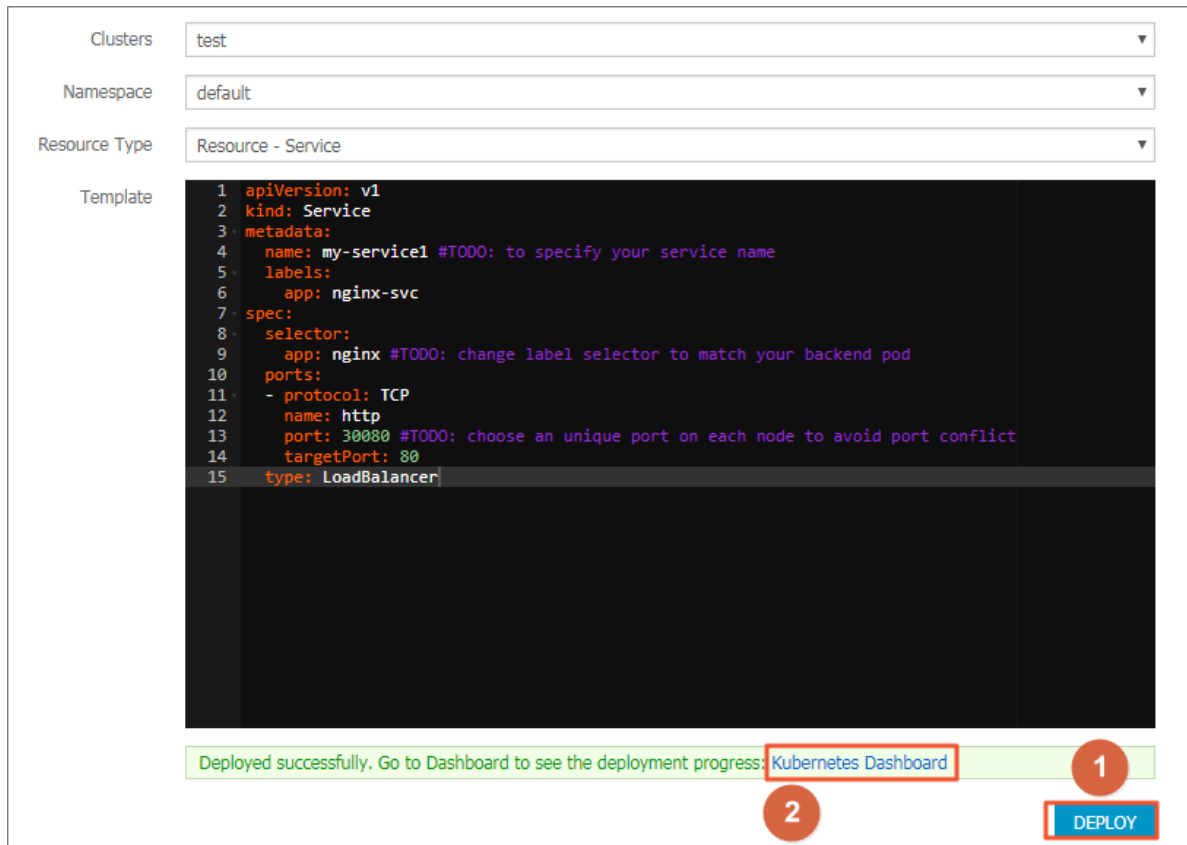
#### Note:

In this example, the selector values in the pod orchestration and service orchestration are both `nginx`, so no modification is required. Make the corresponding modifications according to your actual situations.

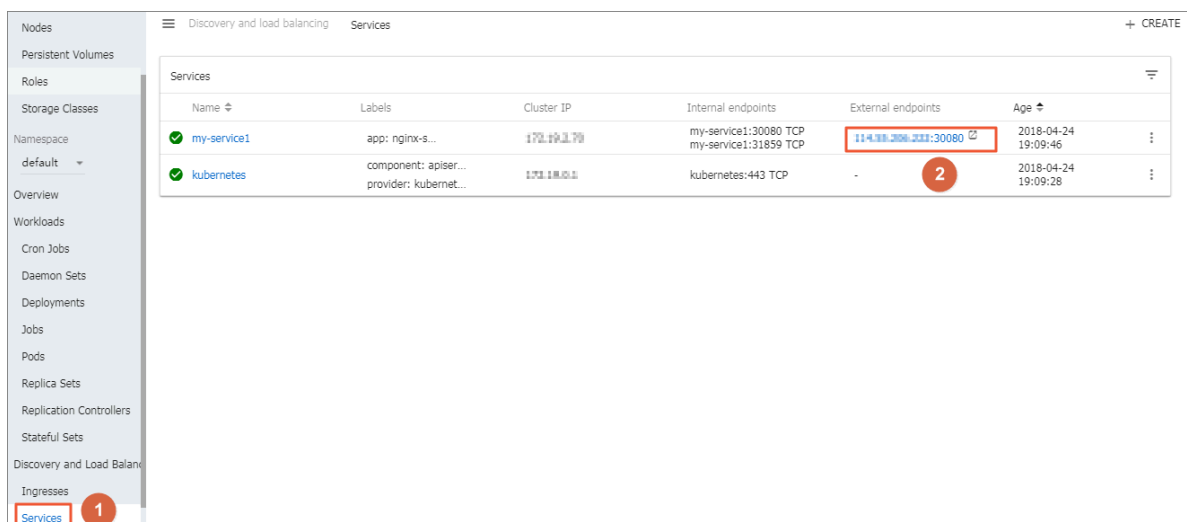
```
apiVersion : v1 # for versions before 1.8.0 use
apps / v1beta1
kind : Service
metadata :
  name : my-service1 # TODO : to specify your service
  name
  labels :
    app : nginx
spec :
  selector :
    app : nginx # TODO : change label selector to match
your backend pod
  ports :
    - protocol : TCP
      name : http
      port : 30080 # TODO : choose an unique port on
each node to avoid port conflict
      targetPort : 80
```

```
type : LoadBalancer ## In this example , change the
type from NodePort to LoadBalancer .
```

7. Enter the preceding orchestration contents in the Template field and then click **DEPLOY**. A message indicating the deployment status is displayed after you click **DEPLOY**. After the successful deployment, click **Kubernetes Dashboard** in the message to go to the dashboard and check the deployment progress of the service.



8. In the Kubernetes dashboard, you can see the service my-service1 is successfully deployed and exposes the external endpoint. Click the access address under **External endpoints**.





9. You can access the Nginx service welcome page in the browser.



## 1.4.4 Simplify Kubernetes application deployment by using Helm

In Kubernetes, app management is the most challenging and in demand field. The Helm project provides a uniform software packaging method which supports version control and greatly simplifies Kubernetes app distribution and deployment complexity.

Alibaba Cloud Container Service integrates the app catalog management function with the Helm tool, extends the functions, and supports official repository, allowing you to deploy the application quickly. You can deploy the application in the Container Service console or by using command lines.

This document introduces the basic concepts and usage of Helm and demonstrates how to use Helm to deploy the sample applications WordPress and Spark on an Alibaba Cloud Kubernetes cluster.

### Basic concepts of Helm

Helm is an open-source tool initiated by Deis and helps to simplify the deployment and management of Kubernetes applications.

You can understand Helm as a Kubernetes package management tool that facilitates discovery, sharing and use of apps built for Kubernetes. It involves several basic concepts.

- **Chart:** A Helm package containing the images, dependencies, and resource definitions required for running an application. It may also contain service definitions in a Kubernetes cluster, similar to the formula of Homebrew, the dpkg of APT, or the rpm file of Yum.
- **Release:** A chart running on a Kubernetes cluster. A chart can be installed multiple times on the same cluster. A new release will be created every time a chart is

installed. For example, to run two databases on the server, you can install the MySQL chart twice. Each installation will generate its own release with its own release name.

- **Repository:** The repository for publishing and storing charts.

## Helm components

Helm adopts a client/server architecture composed of the following components:

- **Helm CLI** is the Helm client and can be run locally or on the master nodes of the Kubernetes cluster.
- **Tiller** is the server component and runs on the Kubernetes cluster. It manages the lifecycles of Kubernetes applications.
- **Repository** is the chart repository. The Helm client accesses the chart index files and packages in the repository by means of the HTTP protocol.

## Use Helm to deploy applications

### Prerequisites

- Before using Helm to deploy an application, create a Kubernetes cluster in Alibaba Cloud Container Service. For more information, see [#unique\\_13](#).

Tiller is automatically deployed to the cluster when the Kubernetes cluster is created. Helm CLI is automatically installed on all the master nodes and the configuration points to the Alibaba Cloud chart repository.

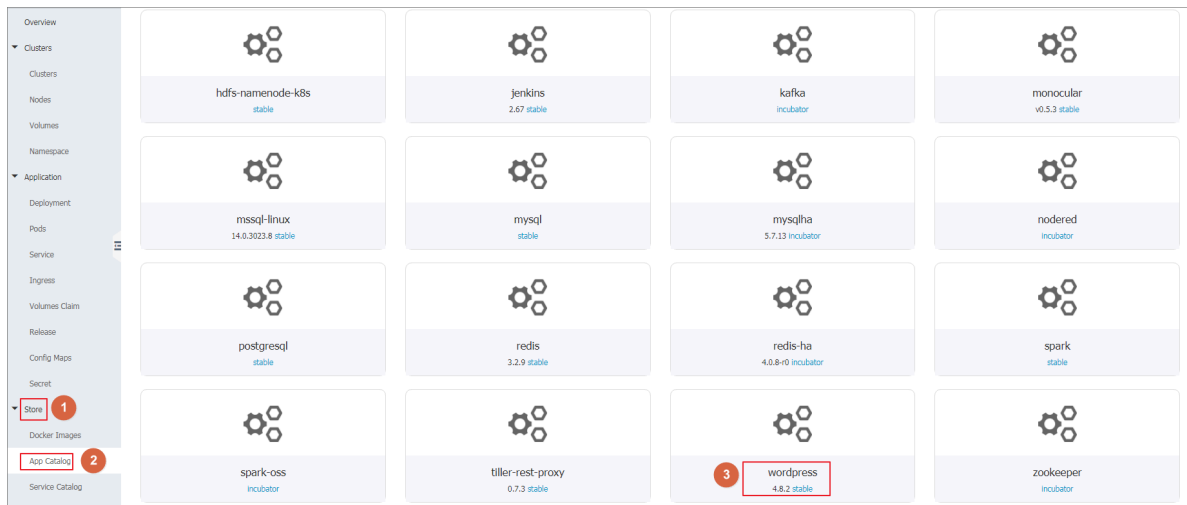
- Check the Kubernetes version of your cluster.

Only clusters whose Kubernetes version is 1.8.4 or later are supported. For clusters whose Kubernetes version is 1.8.1, upgrade the cluster on the Cluster List page.

## Deploy applications in Container Service console

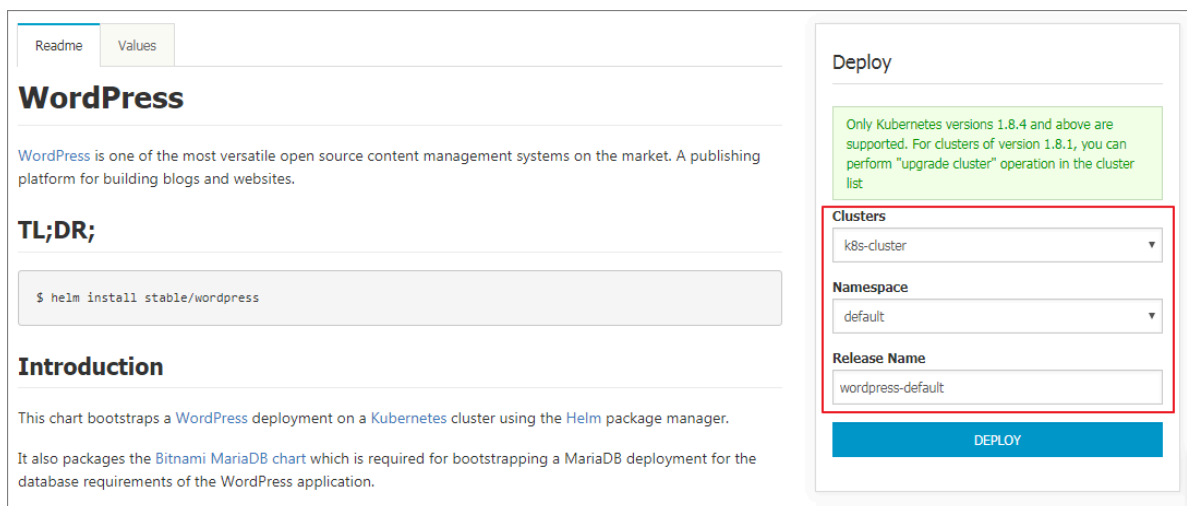
1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Store > App Catalog in the left-side navigation pane.

3. On the App Catalog page, click a chart (WordPress in this example) to enter the chart details page.



4. Enter the basic information for the deployment on the right.

- **Clusters:** Select the cluster in which the application is to be deployed.
- **Namespace:** Select the namespace. default is selected by default.
- **Release Name:** Enter the release name for the application. Enter test in this example.



5. Click the Values tab to modify the configurations.

In this example, bind dynamic data volumes of the cloud disk to a persistent storage volume claim (PVC). For more information, see [#unique\\_29](#).



**Note:**

You need to create a persistent storage volume (PV) of cloud disk in advance. The capacity of the PV cannot be less than the value defined by the PVC.

Readme
Values

```

72  ##
73  mariadbDatabase: bitnami_wordpress
74
75  ## Create a database user
76  ## ref: https://github.com/bitnami/bitnami-docker-mariadb/blob/master/README.md#creating-a
  -database-user-on-first-run
77  ##
78  mariadbUser: bn_wordpress
79
80  ## Password for mariadbUser
81  ## ref: https://github.com/bitnami/bitnami-docker-mariadb/blob/master/README.md#creating-a
  -database-user-on-first-run
82  ##
83  # mariadbPassword:
84
85  ## Enable persistence using Persistent Volume Claims
86  ## ref: http://kubernetes.io/docs/user-guide/persistent-volumes/
87  ##
88  persistence:
89    enabled: true
90    ## mariadb data Persistent Volume Storage Class
91    ## If defined, storageClassName: <storageClass>
92    ## If set to "-", storageClassName: "", which disables dynamic provisioning
93    ## If undefined (the default) or set to null, no storageClassName spec is
94    ## set, choosing the default provisioner. (gp2 on AWS, standard on
95    ## GKE, AWS & OpenStack)
96    ##
97    storageClass: "alicloud-disk-efficiency"
98    accessMode: ReadWriteOnce
99    size: 20Gi
100
101  ## Kubernetes configuration
102  ## For minikube, set this to NodePort, elsewhere use LoadBalancer
103  ##
104  serviceType: LoadBalancer
105

```

Deploy

Only Kubernetes versions 1.8.4 and above are supported. For clusters of version 1.8.1, you can perform "upgrade cluster" operation in the cluster list

Clusters  
k8s-cluster

Namespace  
default

Release Name  
wordpress-default

DEPLOY

Version

0.6.13

Project Homepage

6. Click DEPLOY after completing the configurations. After the successful deployment, you are redirected to the release page of this application.

Container Service - Kubernetes

Overview
Clusters
Nodes
Volumes
Namespace
Application
Deployment
Pods
Service
Ingress
Volumes Claim
Release
Config Maps

Release List - wordpress-default

Refresh

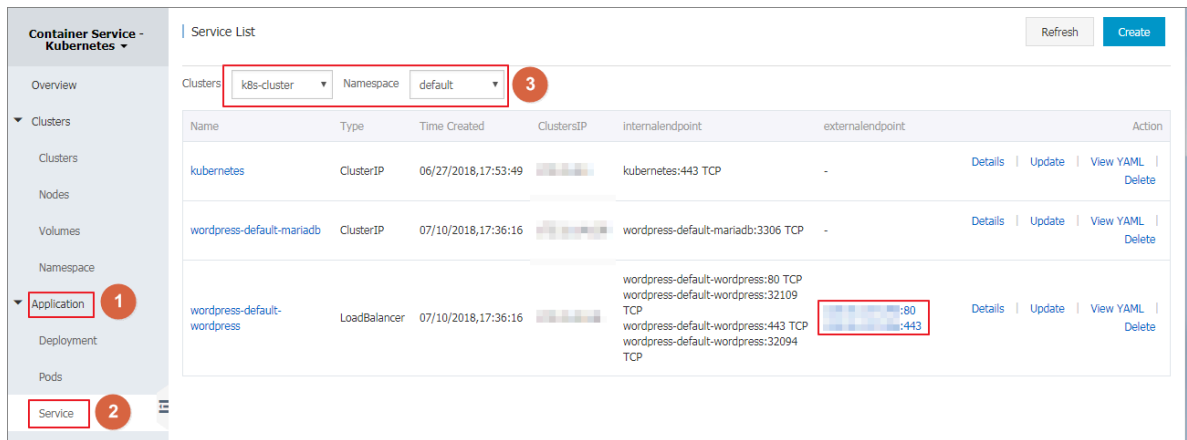
Current Version

Release Name : wordpress-default
Namespace : default
Deployed at : 07/10/2018,17:11:24

Current Version : 1
Time Updated : 07/10/2018,17:11:24

Resource	Kind	Values
wordpress-default-mariadb	Secret	<a href="#">View YAML</a>
wordpress-default-wordpress	Secret	<a href="#">View YAML</a>
wordpress-default-mariadb	ConfigMap	<a href="#">View YAML</a>
wordpress-default-mariadb	PersistentVolumeClaim	<a href="#">View YAML</a>
wordpress-default-wordpress	PersistentVolumeClaim	<a href="#">View YAML</a>
wordpress-default-mariadb	Service	<a href="#">View YAML</a>
wordpress-default-wordpress	Service	<a href="#">View YAML</a>
wordpress-default-mariadb	Deployment	<a href="#">View YAML</a>
wordpress-default-wordpress	Deployment	<a href="#">View YAML</a>

- Click **Application > Service** in the left-hand navigation pane. Select the target cluster and namespace and find the corresponding service. You can obtain the HTTP/HTTPS external endpoint address.



- Click the preceding access address to enter the WordPress blog publishing page.

## Deploy applications by using command lines

You can use SSH to log on to the master node of the Kubernetes cluster when deploying applications by using command lines (Helm CLI is automatically installed and has configured the repository). For more information, see [#unique\\_10](#). You can also install and configure the kubectl and Helm CLI locally.

In this example, install and configure the kubectl and Helm CLI locally and deploy the applications WordPress and Spark.

## Install and configure kubectl and Helm CLI

- Install and configure kubectl on a local computer.

For more information, see [#unique\\_20](#).

To view information of the target Kubernetes cluster, enter the command `kubectl cluster - info`.

- Install Helm on a local computer.

For the installation method, see [Install Helm](#).

- Configure the Helm repository. Here the charts repository provided by Alibaba Cloud Container Service is used.

```
helm init --client-only --stable-repo-url https://aliacs-app-catalog.oss-cn-hangzhou.aliyuncs.com/charts/
helm repo add incubator https://aliacs-app-catalog.oss-cn-hangzhou.aliyuncs.com/charts-incubator/
```

```
helm repo update
```

### Basic operations of Helm

- To view the list of charts installed on the cluster, enter the following command:

```
helm list
```

Or you can use the abbreviated version:

```
helm ls
```

- To view the repository configurations, enter the following command:

```
helm repo list
```

- To view or search for the Helm charts in the repository, enter one of the following commands:

```
helm search
helm search repository name # For example , stable or
incubator .
helm search chart name # For example , wordpress or
spark .
```

- To update the chart list to get the latest version, enter the following command:

```
helm repo update
```

For more information about how to use Helm, see [Helm document](#).

### Deploy WordPress by using Helm

Use Helm to deploy a WordPress blog website.

Enter the following command.

```
helm install --name wordpress --test stable / wordpress
```



#### Note:

The Alibaba Cloud Kubernetes service provides the support for dynamic storage volumes of block storage (cloud disk). You need to create a storage volume of cloud disk in advance.

The result is as follows:

```
NAME : wordpress -- test
LAST DEPLOYED : Mon Nov 20 19 : 01 : 55 2017
NAMESPACE : default
STATUS : DEPLOYED
```

```
...
```

Use the following command to view the release and service of WordPress.

```
helm list
kubectl get svc
```

Use the following command to view the WordPress related pods and wait until the status changes to Running.

```
kubectl get pod
```

Use the following command to obtain the WordPress access address:

```
echo http://${kubectl get svc wordpress - test - wordpress
-o jsonpath='{. status . loadBalancer . ingress [ 0 ]. ip }'}
```

Access the preceding URL in the browser, and you can see the familiar WordPress website.

You can also follow the chart instructions and use the following command to obtain the administrator account and password of the WordPress website:

```
echo Username : user
echo Password : $(kubectl get secret -- namespace default
wordpress - test - wordpress - o jsonpath="{. data . wordpress -
password }" | base64 -- decode )
```

To completely delete the WordPress application, enter the following command:

```
helm delete -- purge wordpress - test
```

## Deploy Spark by using Helm

Use Helm to deploy Spark for processing big data.

Enter the following command:

```
helm install -- name myspark stable / spark
```

The result is as follows:

```
NAME : myspark
LAST DEPLOYED : Mon Nov 20 19 : 24 : 22 2017
NAMESPACE : default
STATUS : DEPLOYED
...
```

Use the following commands to view the release and service of Spark.

```
helm list
```

```
kubectl get svc
```

Use the following command to view the Spark related pods and wait until the status changes to Running. Pulling images takes some time because the Spark related images are large.

```
kubectl get pod
```

Use the following command to obtain the Spark Web UI access address:

```
echo http://$(kubectl get svc myspark - webui - o  
jsonpath='{. status . loadBalanc er . ingress [ 0 ]. ip }'): 8080
```

Access the preceding URL in the browser, and you can see the Spark Web UI, on which indicating currently three worker instances exist.

Then, use the following command to use Helm to upgrade the Spark application and change the number of worker instances from three to four. The parameter name is case sensitive.

```
helm upgrade myspark -- set " Worker . Replicas = 4 " stable /  
spark
```

The result is as follows:

```
Release " myspark " has been upgraded . Happy Helming !  
LAST DEPLOYED : Mon Nov 20 19 : 27 : 29 2017  
NAMESPACE : default  
STATUS : DEPLOYED  
...
```

Use the following command to view the newly added pods of Spark and wait until the status changes to Running.

```
kubectl get pod
```

Refresh the Spark Web UI in the browser. The number of worker instances changes to four.



To completely delete the Spark application, enter the following command:

```
helm delete --purge myspark
```

### Use third-party chart repository

Besides the preset Alibaba Cloud chart repository, you can also use the third-party chart repository (make sure the network is accessible). Add the third-party chart repository in the following command format:

```
helm repo add repository name repository URL
helm repo update
```

For more information about the Helm related commands, see [Helm document](#).

### References

Helm boosts the growth of communities. More and more software providers, such as Bitnami, have begun to provide high-quality charts. You can search for and discover existing charts at <https://kubernetes.io/docs/concepts/containers/kubernetes-deployments/>.

## 1.4.5 Manage applications by using commands

You can create applications or view containers in applications by using commands.

### Prerequisites

Before using commands to manage applications, [#unique\\_20](#).

## Create an application by using commands

Run the following statements to run a simple container (a Nginx Web server in this example).

```
root @ master # kubectl run -it nginx -- image = registry .
aliyuncs . com / spacexnice / netdia : latest
```

This command creates a service portal for this container. Specify `-- type =`

`LoadBalancer` and an Alibaba Cloud Server Load Balancer route will be created to the Nginx container.

```
root @ master # kubectl expose deployment nginx -- port = 80
-- target - port = 80 -- type = LoadBalancer
```

## View containers by using commands

Run the following command to list all the running containers in the default namespaces.

```
root @ master # kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx - 2721357637 - dvwq3    1 / 1     Running    1      9h
```

### 1.4.6 Create a service

A Kubernetes service, which is generally called a microservice, is an abstraction which defines a logical set of pods and a policy by which to access them. The set of pods accessed by a Kubernetes service is usually determined by a Label Selector.

Kubernetes pods are created and deleted in a short time even if they have their own IP addresses. Therefore, using pods directly to provide services externally is not a solution of high availability. The service abstraction decouples the relationship between the frontend and the backend. Therefore, the loose-coupling microservice allows the frontend to not care about the implementations of the backend.

For more information, see [Kubernetes service](#).

## Prerequisites

You have created a Kubernetes cluster successfully. For how to create a Kubernetes cluster, see [#unique\\_13](#).

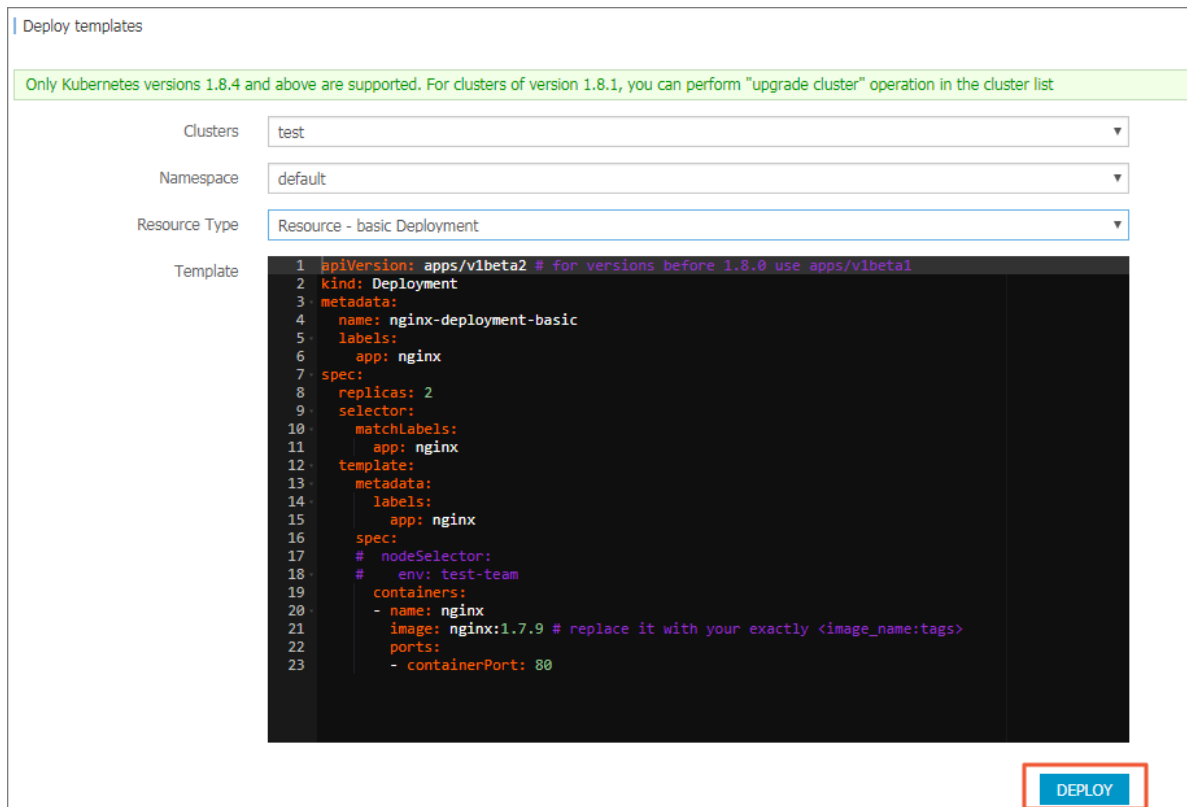
## Step 1 Create a deployment

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

2. Log on to the [Container Service console](#).
3. Click **Kubernetes > Application > > Deployment** in the left-side navigation pane. Click **Create by template** in the upper-right corner.



4. Select the cluster and namespace to create the deployment. In the Resource Type drop-down list, select Custom to customize the template or a sample template. Then, click **DEPLOY**.



In this example, the sample template is an Nginx deployment.

```

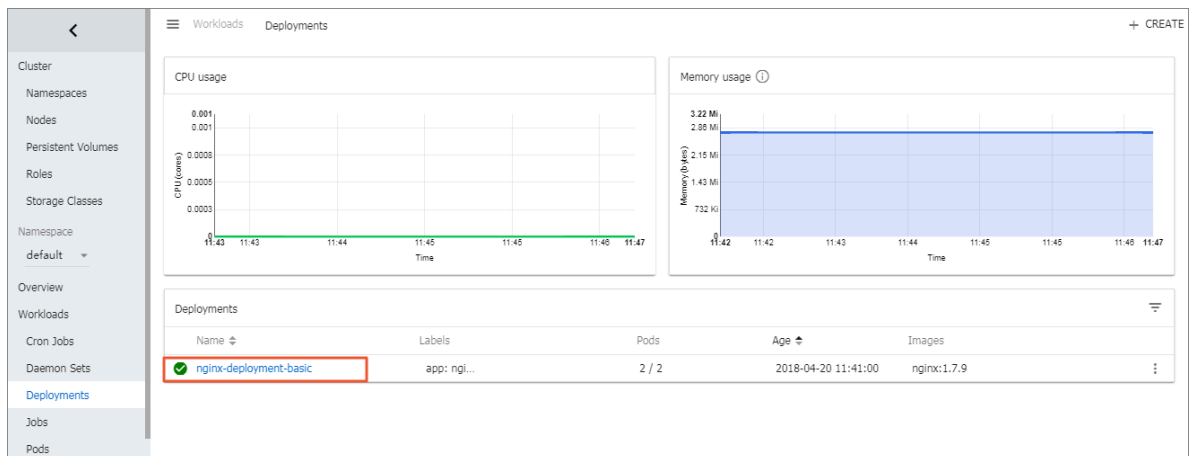
apiVersion : apps / v1beta2 # for versions before 1 . 8 .
0 use apps / v1beta1
kind : Deployment
metadata :
  name : nginx - deployment - basic
  labels :
    app : nginx
spec :
  replicas : 2
  selector :
    matchLabel s :
      app : nginx
  
```

```

template :
  metadata :
    labels :
      app : nginx
  spec :
    containers :
      - name : nginx
        image : nginx : 1 . 7 . 9 # replace it with your
        exactly < image_name : tags >
        ports :
          - containerP ort : 80 ## You must expose this
            port in the service .

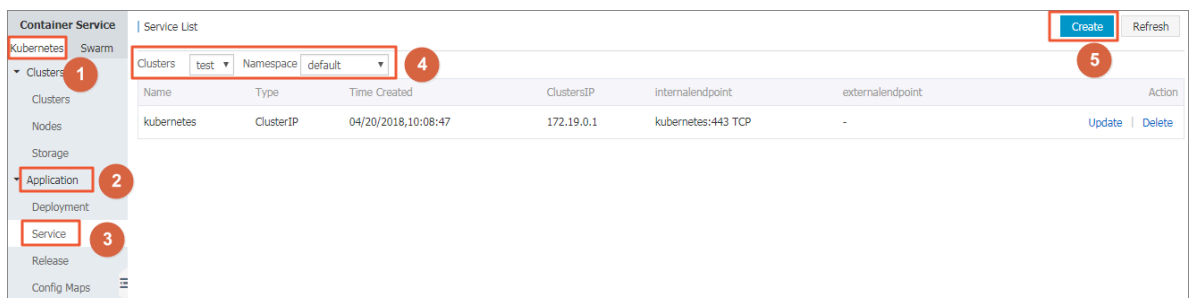
```

5. Go to the Kubernetes dashboard to view the running status of this deployment.



## Step 2 Create a service

1. Log on to the [Container Service console](#).
2. Log on to the [Container Service console](#).
3. Click **Kubernetes > Application > > Service** in the left-side navigation pane.
4. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click **Create** in the upper-right corner.



## 5. Complete the configurations in the displayed Create Service dialog box.

**Create Service**

Name:

Type:

Related deployment:

Port Mapping: +Add

service port	Container Port	Protocol
<input type="text" value="8080"/>	<input type="text" value="80"/>	<input type="text" value="TCP"/>

Create Cancel

- Name: Enter the service name. In this example, enter nginx-svc.
- Type: Select the service type, namely, the access method of the service.
  - ClusterIP: Exposes the service by using the internal IP address of your cluster . With this type selected, the service is only accessible from within the cluster . This type is the default service type.
  - NodePort: Exposes the service by using the IP address and static port (NodePort) on each node. A ClusterIP service, to which the NodePort service is routed, is automatically created. You can access the NodePort service from outside the cluster by requesting `< NodeIP > : < NodePort >`.
  - Server Load Balancer: Exposes the service by using Server Load Balancer , which is provided by Alibaba Cloud. Select public or inner to access the service by using the Internet or intranet. Alibaba Cloud Server Load Balancer can route to the NodePort and ClusterIP services.
- Related deployment: Select the backend object to bind with this service. In this example, select nginx-deployment-basic, the deployment created in the preceding step. The corresponding Endpoints object is not created if no

deployment is selected here. You can manually map the service to your own endpoints. For more information, see [Services without selectors](#).

- **Port Mapping:** Add the service port and container port. The container port must be the same as the one exposed in the backend pod.

6. Click Create. The nginx-svc service is displayed on the Service List page.

Container Service		Service List					Create	Refresh
Kubernetes	Swarm							
Clusters		Clusters	test	Namespace	default			
Clusters		Name	Type	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint	Action
Nodes		kubernetes	ClusterIP	04/20/2018,10:08:47	172.17.0.1	kubernetes:443 TCP	-	Update   Delete
Storage		nginx-svc	LoadBalancer	04/20/2018,14:06:08	172.17.0.128	nginx-svc:8080 TCP nginx-svc:30138 TCP	47.97.246.68:8080	Update   Delete
Application								

7. View the basic information of the service. Access the external endpoint of the nginx-svc service in the browser.



Then, you have created a service that is related to a backend deployment and accessed the Nginx welcome page successfully.

## 1.4.7 Schedule a pod to a specified node

You can add a node label and then configure the `nodeSelector` or `nodeSelector` to schedule a pod to a specified node. For more information about the implementation principle of `nodeSelector`, see [nodeSelector](#).

For business scenario needs, to deploy a service used for management and control to a master node, or deploy services to a machine with an SSD disk, you can use this method to schedule pods to specified nodes.

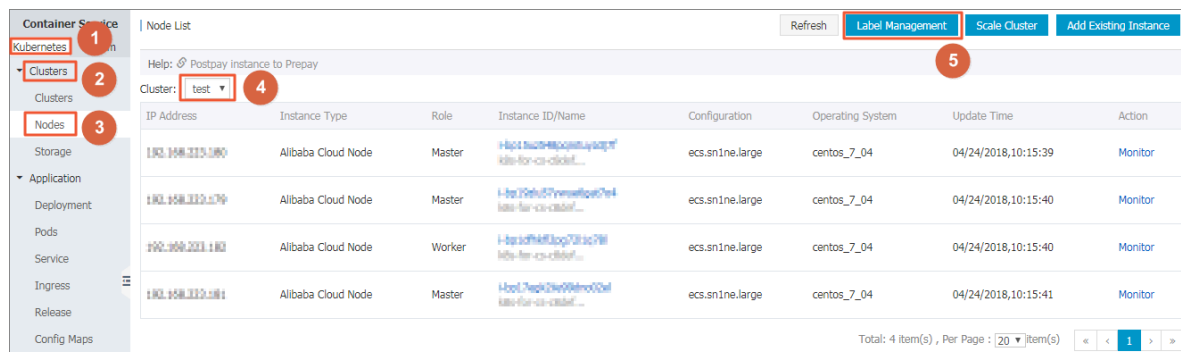
### Prerequisites

You have successfully created a Kubernetes cluster. For more information, see [#unique\\_13](#).

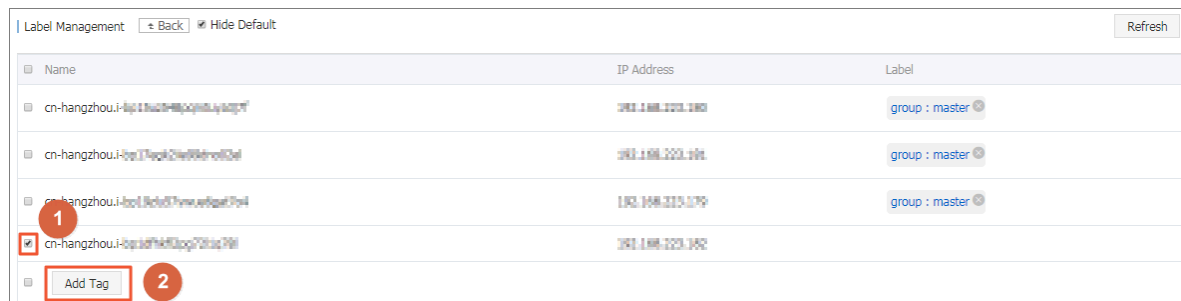
### Step 1 Add a node label

1. Log on to the [Container Service console](#).
2. Under the Kubernetes menu, click Clusters > Nodes in the left-side navigation pane.

3. Select the cluster from the Cluster drop-down list and then click Label Management in the upper-right corner.



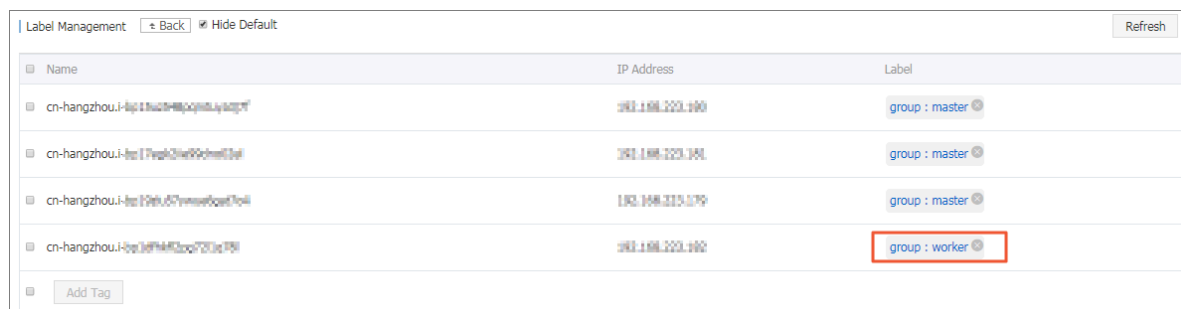
4. Select one or more nodes by selecting the corresponding check boxes and then click Add Tag. In this example, select a worker node.



5. Enter the name and value of the label in the displayed dialog box and then click OK.



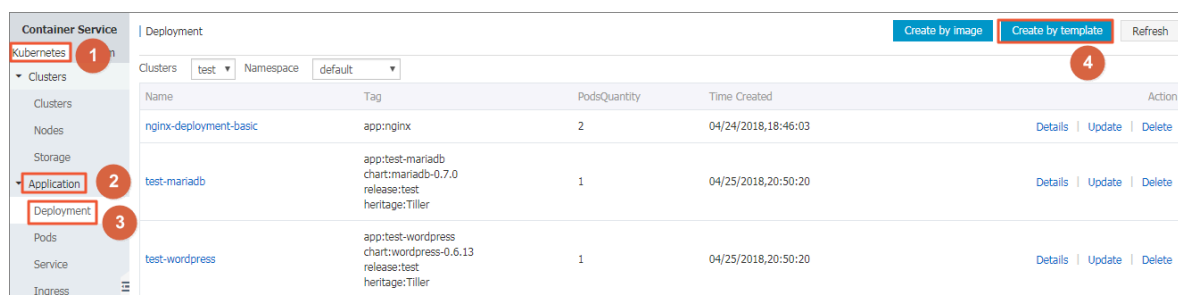
The node label `group : worker` is displayed on the Label Management page.



You can also add a node label by running the command `kubectl label nodes < node - name > < label - key >=< label - value >.`

## Step 2 Deploy a pod to a specified node

1. Log on to the [Container Service console](#).
2. Under the Kubernetes menu, click Applications > Deployment in the left-side navigation pane.
3. Click Create by template in the upper-right corner.





#### 4. Configure the template to deploy a pod. After completing the configurations, click **DEPLOY**.

- **Clusters:** Select a cluster.
- **Namespace:** Select the namespace to which the resource object belongs. In this example, use default as the namespace.
- **Resource Type:** Select Custom in this example.

Deploy templates

Only Kubernetes versions 1.8.4 and above are supported. For clusters of version 1.8.1, you can perform "upgrade cluster" operation in the cluster list

Clusters
test

Namespace
default

Resource Type
Custom

Template

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    labels:
5      name: hello-pod
6      name: hello-pod
7  spec:
8    containers:
9      - image: nginx
10      imagePullPolicy: IfNotPresent
11      name: hello-pod
12      ports:
13        - containerPort: 8080
14          protocol: TCP
15      resources: {}
16      securityContext:
17        capabilities: {}
18        privileged: false
19      terminationMessagePath: /dev/termination-log
20      dnsPolicy: ClusterFirst
21      restartPolicy: Always
22      nodeSelector:
23        group: worker
24  status: {}

```

DEPLOY

The orchestration template in this example is as follows:

```

apiVersion : v1
kind : Pod
metadata :
  labels :
    name : hello - pod
    name : hello - pod
spec :
  containers :
    - image : nginx
      imagePullPolicy : IfNotPresent
      name : hello - pod
      ports :
        - containerPort : 8080
          protocol : TCP
      resources : {}
      securityContext :
        capabilities : {}
        privileged : false
      terminationMessagePath : / dev / termination - log
      dnsPolicy : ClusterFirst
      restartPolicy : Always

```

```
nodeSelect or :
  group : worker ## The same as the node label
  configured in the preceding step .
status :{}
```

5. A message indicating the deployment status is displayed after you click **DEPLOY** .  
After the successful deployment, click **Kubernetes Dashboard** in the message to go to the dashboard and check the deployment status.

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
hello-pod	cn-hangzhou-1-b0t4t4k3log7211a791	Running	0	2018-04-27 17:04:18	0	1.445 Mi
test-mariadb-9bb8f87dd-fjm2m	cn-hangzhou-1-b0t4t4k3log7211a791	Running	0	2018-04-25 20:50:20	0.002	217.309 Mi
test-wordpress-5b74dcf48c-r8j9h	cn-hangzhou-1-b0t4t4k3log7211a791	Running	0	2018-04-25 20:50:20	0.004	192.145 Mi
nginx-deployment-basic-6c54bd5869-wg2t5	cn-hangzhou-1-b0t4t4k3log7211a791	Running	0	2018-04-25 12:11:48	0	1.344 Mi
nginx-deployment-basic-6c54bd5869-krpf7	cn-hangzhou-1-b0t4t4k3log7211a791	Running	0	2018-04-24 18:46:03	0	1.395 Mi

6. Click the pod name to view the pod details.

You can view the information such as the pod label and node ID, which indicates the pod is successfully deployed to a node with the label `group : worker` .



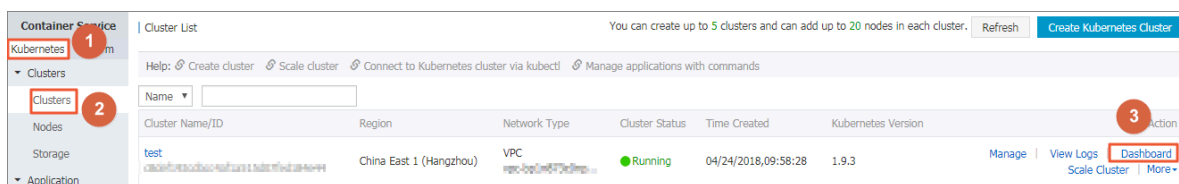
## 1.4.8 Service scaling

After an application is created, you can scale out or in the services as per your needs.

### Procedure

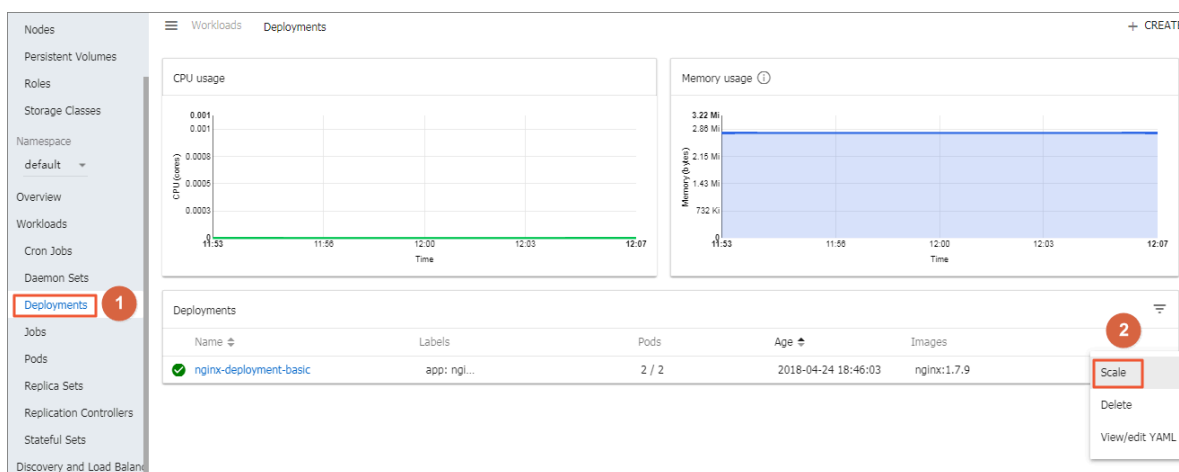
1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Clusters** in the left-side navigation pane.

### 3. Click Dashboard at the right of the cluster to enter the Kubernetes dashboard.



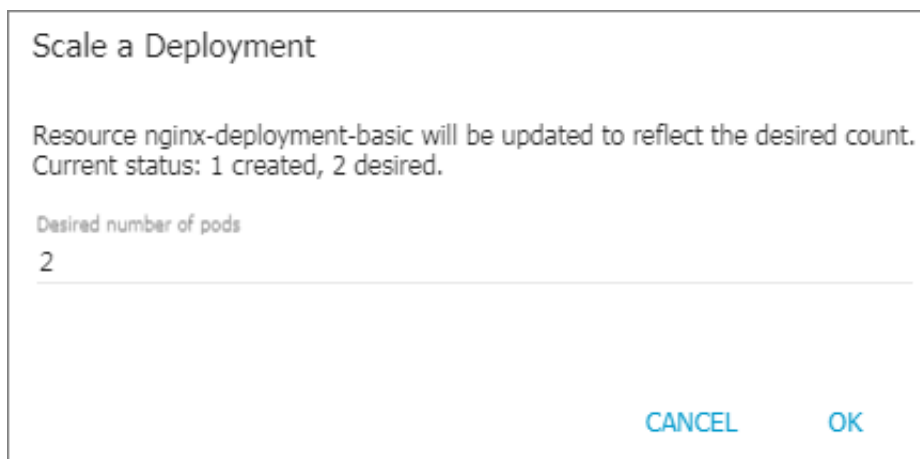
### 4. In the Kubernetes dashboard, click Deployments in the left-side navigation pane to view the created deployments.

### 5. Click the icon at the right of the deployment and then select Scale.





### 6. The Scale a Deployment dialog box appears. Modify the value of Desired number of pods to 2 and then click OK.

Then, a pod is added by expansion and the number of replicas rises to 2.



## What's next

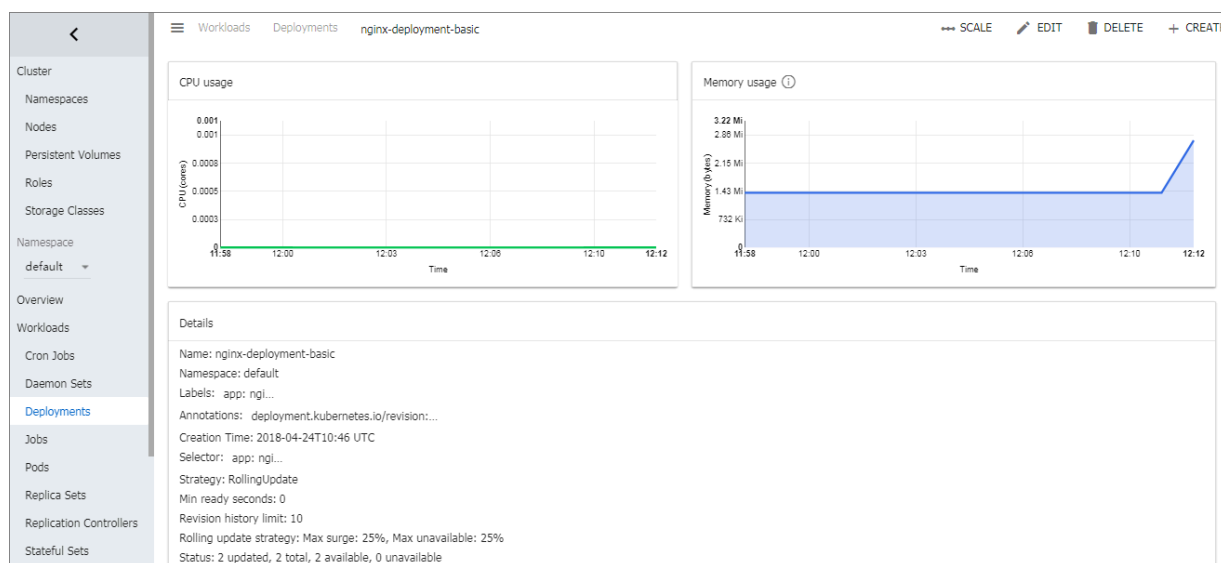
You can check the status of each Kubernetes object according to the icon on the left.  indicates the object is being deployed.  indicates the object has completed the deployment.

After the application completes the deployment, you can click a deployment name to view the details of the running Web service. You can view the replica sets in the deployment, and the CPU usage and memory usage of these replica sets. You can also click Pods in the left-side navigation pane, open a pod, and click LOGS in the upper-right corner to view the container logs.



### Note:

Wait a few minutes if you cannot view any resources.



## 1.4.9 View services

### Context

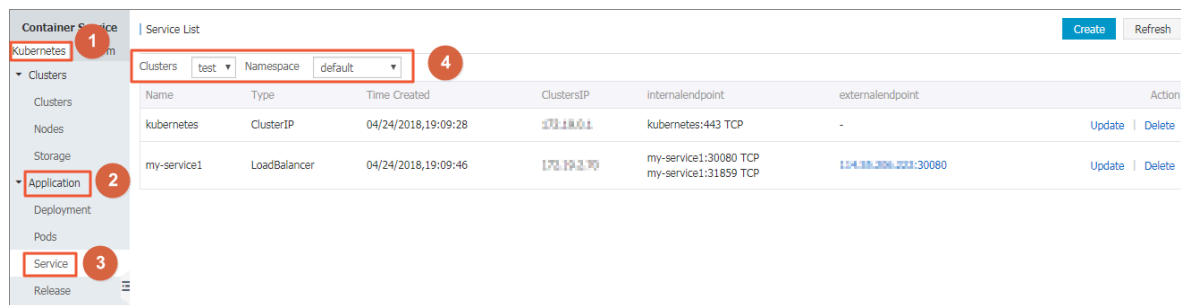
If the external service is configured when you create the application, in addition to running containers, Kubernetes dashboard creates the external services for pre-assigning the Server Load Balancer to bring traffic to the containers in the cluster.

### Procedure

1. Log on to the [Container Service console](#).
2. Click Kubernetes > Application > > Service in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists to view the deployed services.

You can view the name, type, created time, cluster IP address, internal endpoint, and external endpoint of a service. In this example, you can view the external

endpoint (IP address) assigned to the service. Click the IP address to access the Nginx welcome page.



You can also enter the Kubernetes dashboard of the cluster and click Services in the left-side navigation pane to view the services.

## 1.4.10 Delete a service

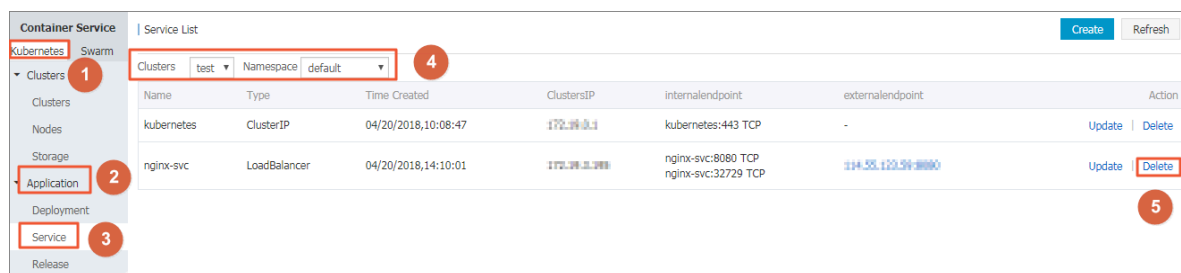
You can delete a Kubernetes service in the Container Service console.

### Prerequisites

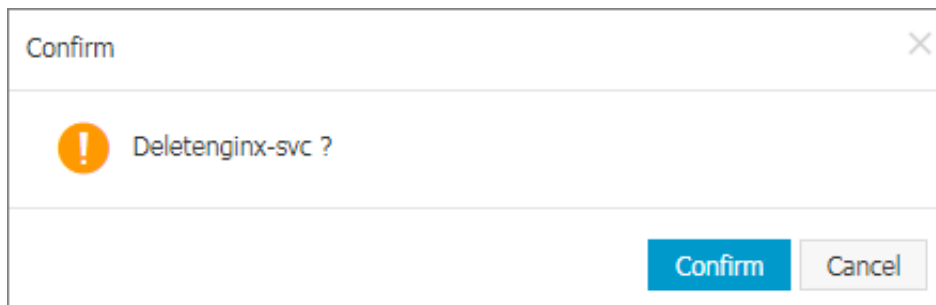
- You have created a Kubernetes cluster successfully. For more information, see [#unique\\_13](#).
- You have created a service successfully. For more information, see [#unique\\_36](#).

### Procedure

- Log on to the [Container Service console](#).
- Click **Kubernetes > Application > Service** in the left-side navigation pane.
- Select the cluster and namespace from the **Clusters** and **Namespace** drop-down lists. Click **Delete** at the right of the service (nginx-svc in this example).



- Click **Confirm** in the displayed dialog box. Then, the service is removed from the **Service List** page.



### 1.4.11 View pods

You can view the pods of a Kubernetes cluster in the Container Service console or in the Kubernetes dashboard.

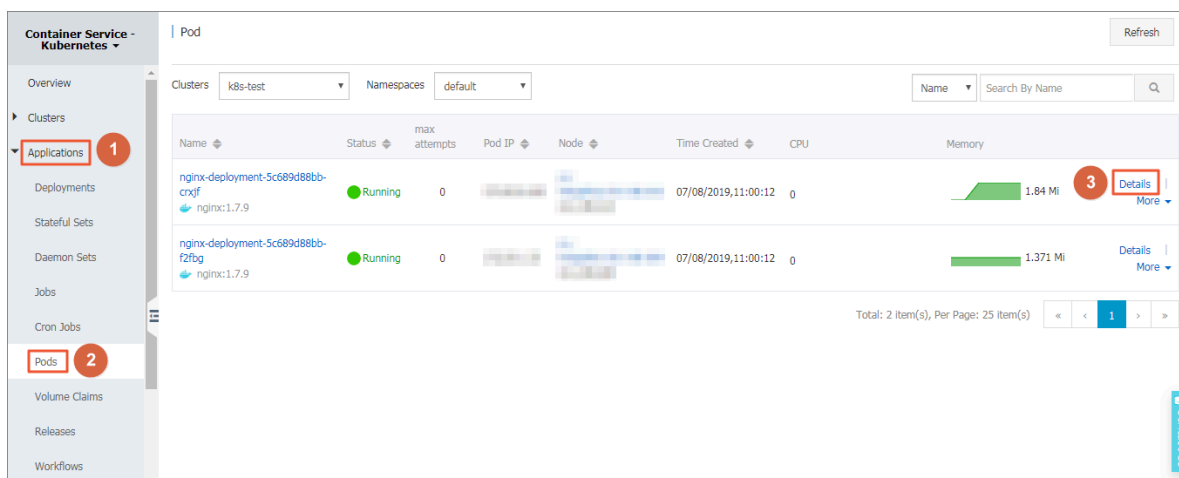
View pods in Container Service console

- Log on to the [Container Service console](#).
- Log on to the [Container Service console](#).
- Click **Kubernetes > Application > > Pods** in the left-side navigation pane.
- Select the cluster and namespace from the **Clusters** and **Namespace** drop-down lists. Click **Detail** at the right of the pod.

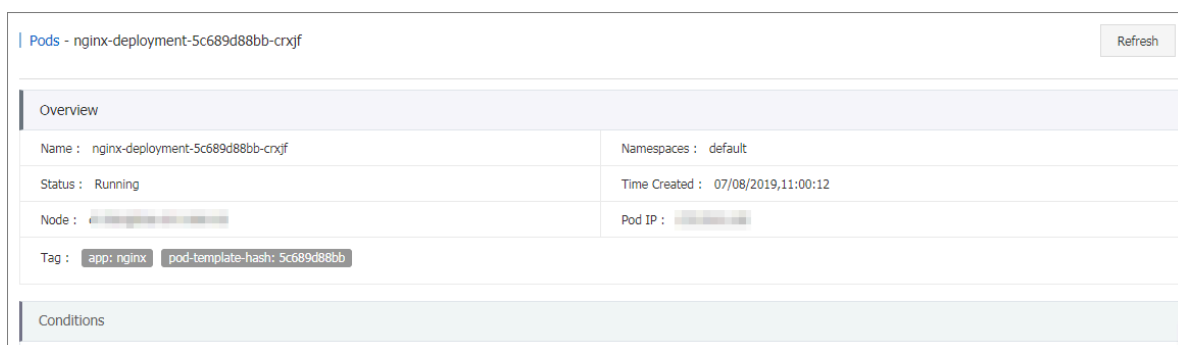


**Note:**

You can update or delete a pod. For pods created by using deployments, we recommend that you manage these pods by using deployments.

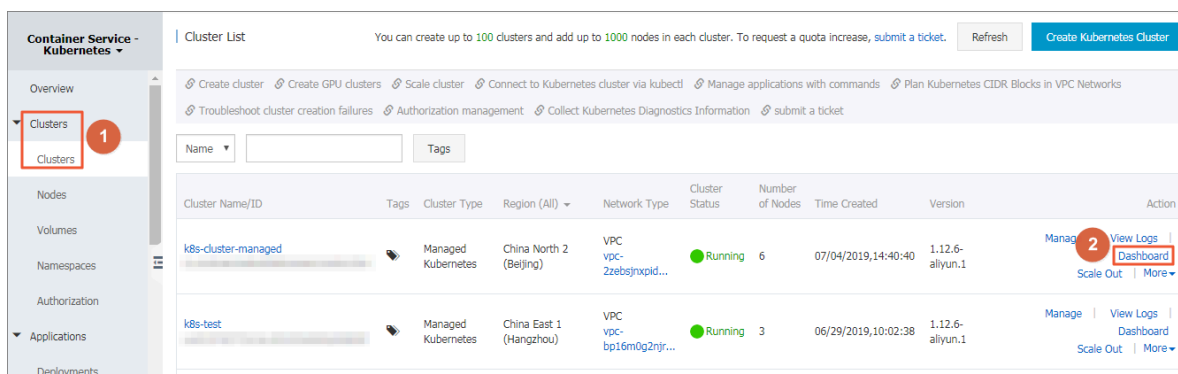


## 5. View the pod details.



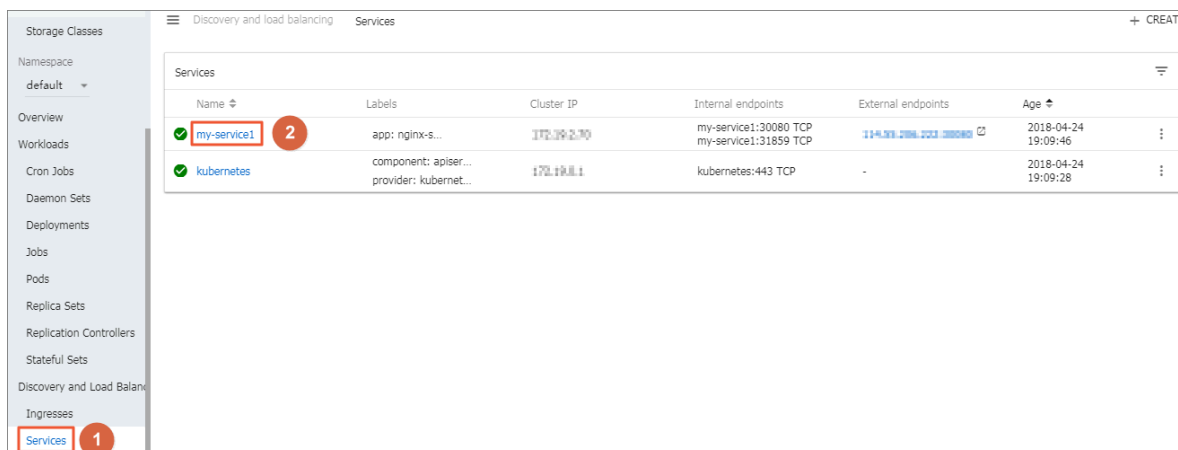
### View pods in Kubernetes dashboard



1. Log on to the [Container Service console](#).
2. Log on to the [Container Service console](#).
3. Click **Kubernetes > Clusters** in the left-side navigation pane.
4. Click **Dashboard** at the right of the cluster to enter the Kubernetes dashboard.

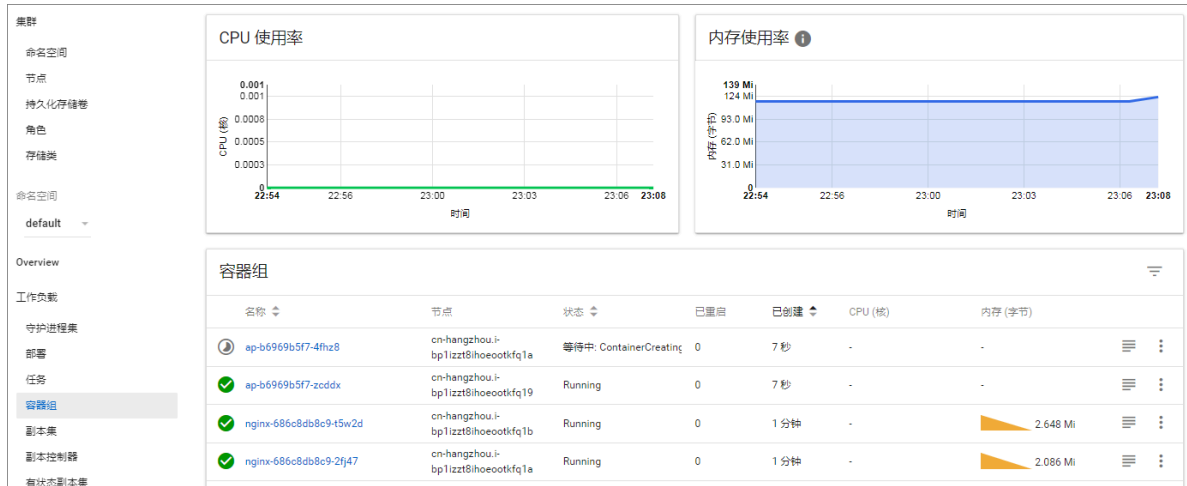


5. In the Kubernetes dashboard, click **Pods** in the left-side navigation pane to view the pods in the cluster.

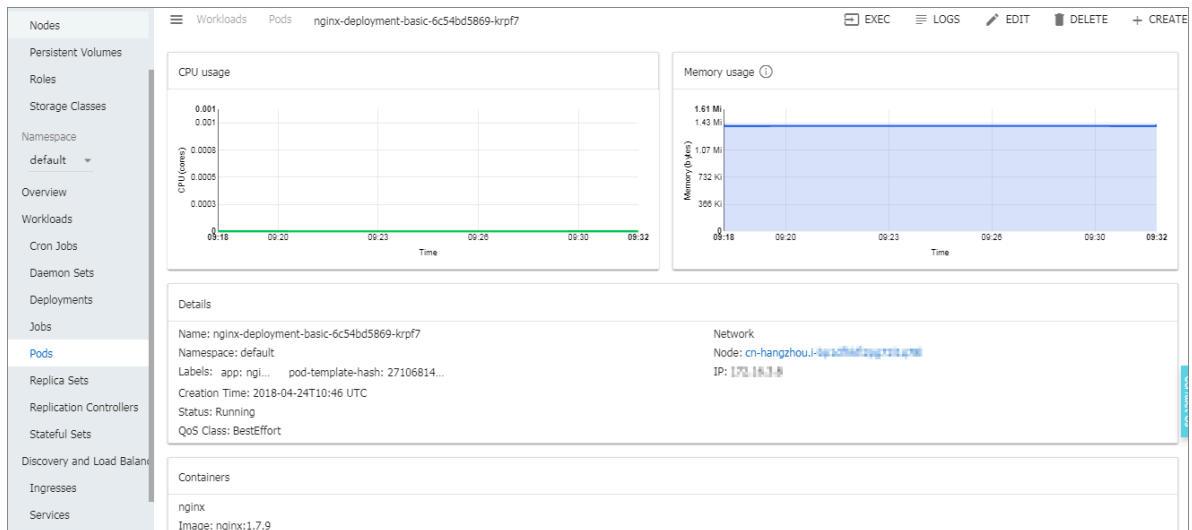
You can also click **Services** in the left-side navigation pane and then click the service name to view the pods in this service.



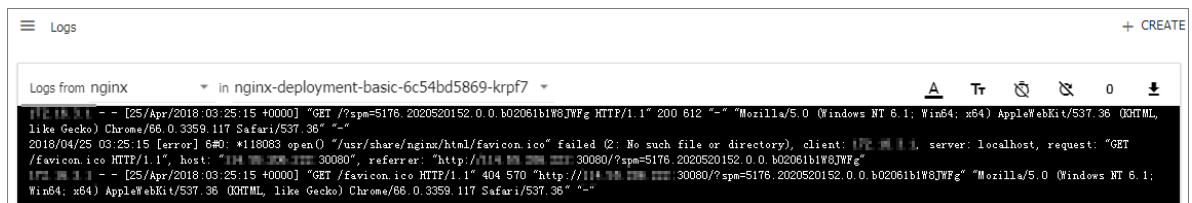
6. You can check the status of each Kubernetes object according to the icon on the left.  indicates the object is being deployed.  indicates the object has completed the deployment.



7. Click the pod name to view the details, CPU usage, and memory usage of the pod.

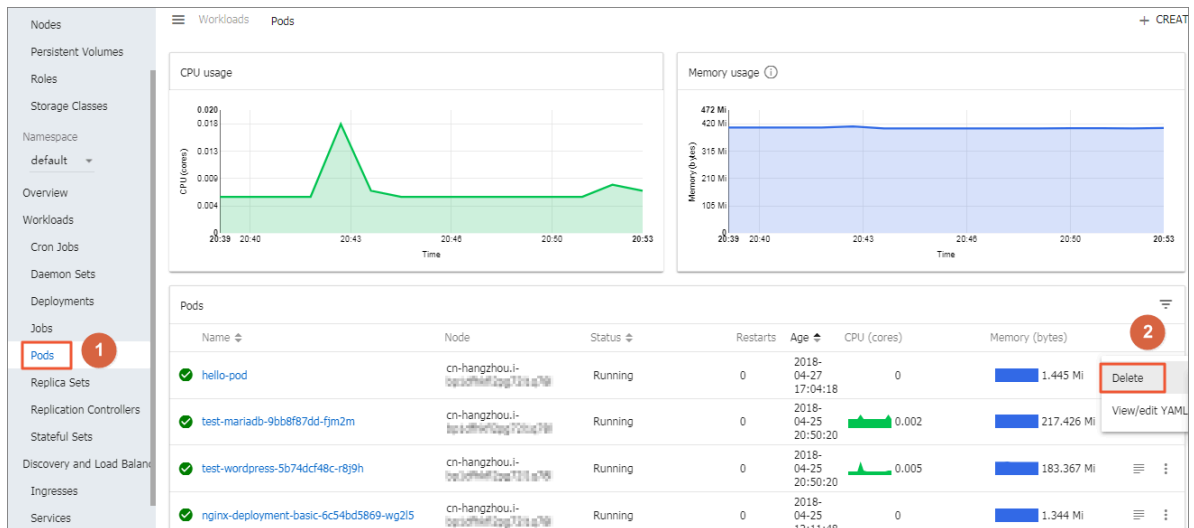


8. Click LOGS in the upper-right corner to view the pod logs.





9. You can also click the icon at the right of the pod and then select Delete to delete the pod.

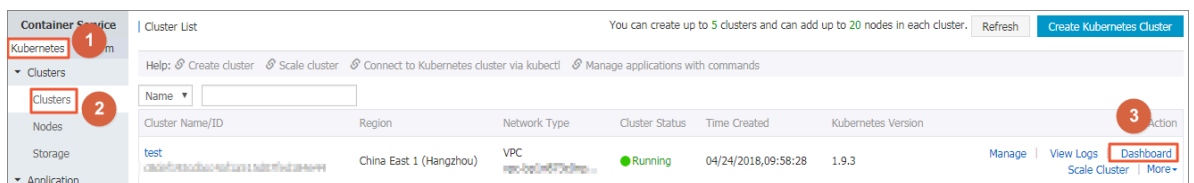


## 1.4.12 Change container configurations

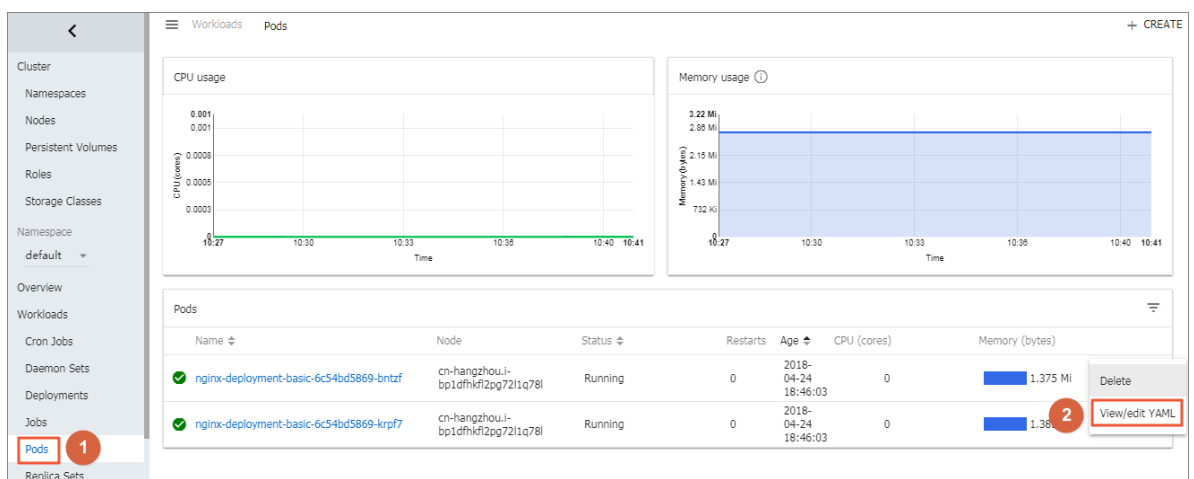
You can change the container configurations in the Container Service console.

### Procedure

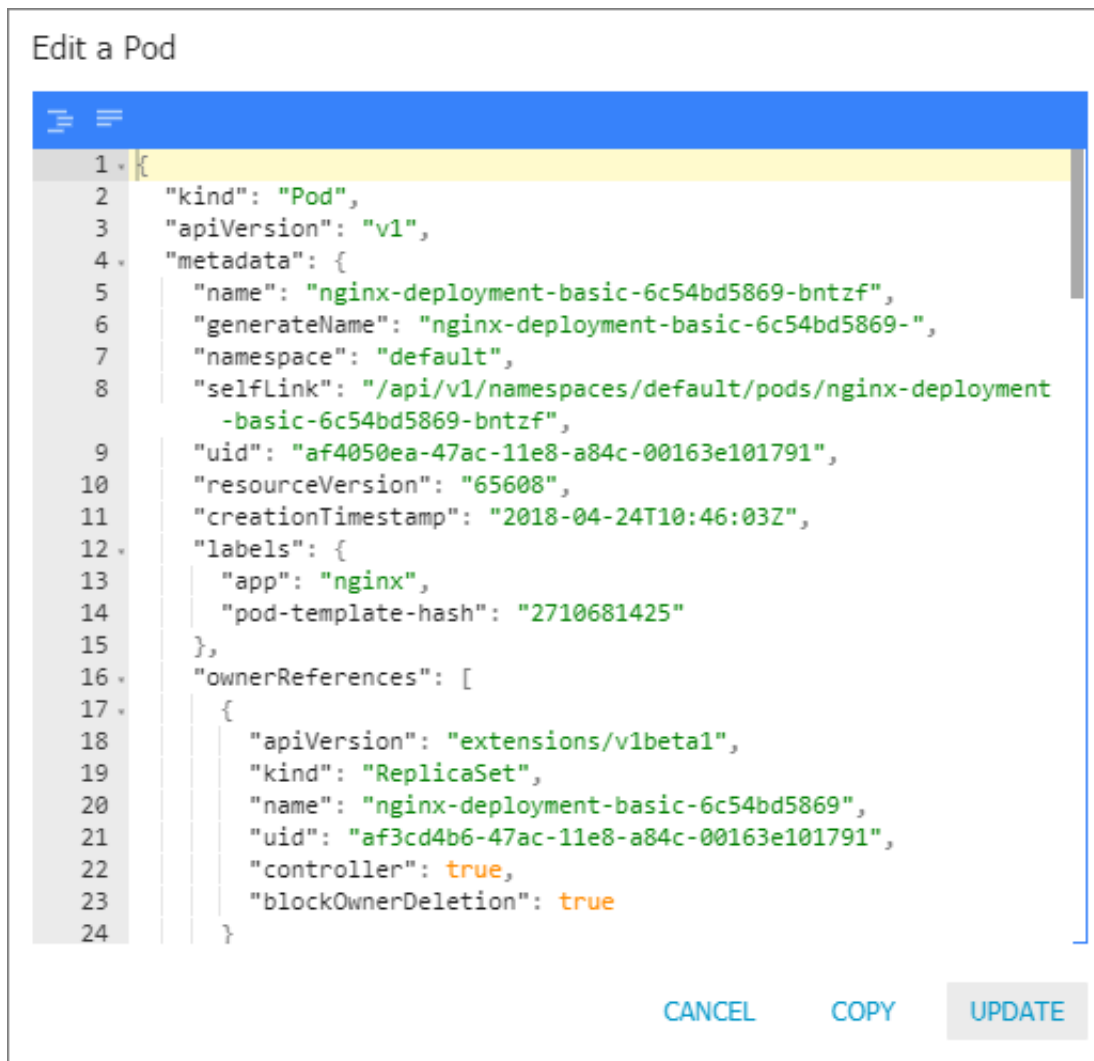
1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Clusters** in the left-side navigation pane.
3. Click **Dashboard** at the right of the cluster to enter the Kubernetes dashboard.



4. In the Kubernetes dashboard, click **Pods** in the left-side navigation pane.
5. Click the icon at the right of the pod and then select **View/edit YAML**.



6. The Edit a Pod dialog box appears. Change the container configurations and then click UPDATE.



## 1.5 Namespaces

### 1.5.1 Create a namespace

#### Prerequisites

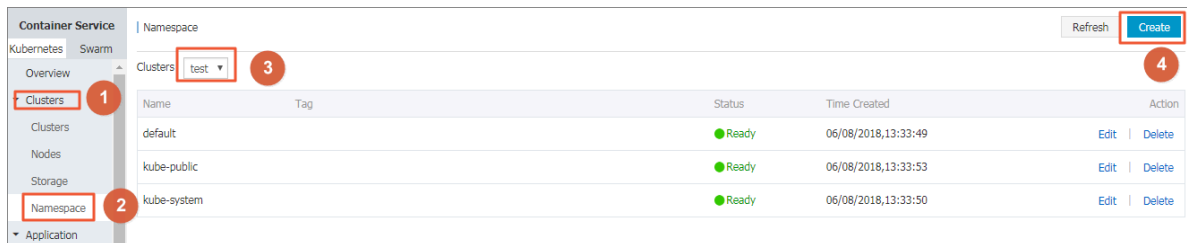
You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).

#### Context

In Kubernetes clusters, you can use the Namespace function to create multiple virtual spaces. When the number of users in one cluster is large, multiple namespaces are used to divide the workspaces effectively and the cluster resources into different purposes. The namespace resources are assigned by using the [resource-quotas](#).

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Clusters** > **Namespace** in the left-side navigation pane.
3. Select the cluster from the Clusters drop-down list and then click **Create** in the upper-right corner.



4. Configure the namespace in the displayed dialog box.

**Create Namespace**

Name:

1-63 characters, can only contain numbers, lower case letters, and "-", and can only be letters or numbers at the beginning and end

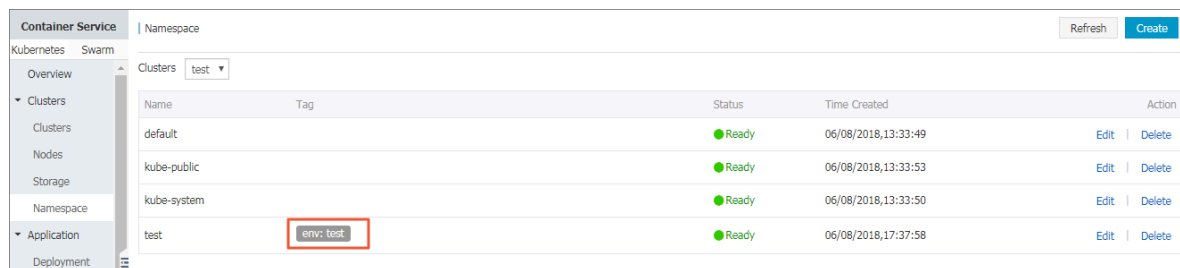
Tags:

Variable Name	Variable Value	Action
env	test	Edit   Delete

- **Name:** Enter the namespace name, which is 1–63 characters long, can only contain numbers, letters, and hyphens (-), and must start and end with a letter or number. In this example, enter test as the name.
- **Tags:** Add one or more tags for the namespace to identify the characteristics of the namespace, for example, to identify this namespace to be used for the test environment.

You can enter the variable name and variable value, and then click **Add** on the right to add a tag for the namespace.

5. Click OK after completing the configurations.
6. The namespace test is successfully created and displayed in the namespace list.



Container Service		Namespace		Refresh	Create
Kubernetes		Clusters test			
Overview	Clusters	Name	Tag	Status	Time Created
▼ Clusters	Clusters	default		Ready	06/08/2018,13:33:49
Nodes	Nodes	kube-public		Ready	06/08/2018,13:33:53
Storage	Storage	kube-system		Ready	06/08/2018,13:33:50
Namespace	Namespace	test	test	Ready	06/08/2018,17:37:58
▼ Application	Application				
Deployment	Deployment				

## 1.5.2 Configure resource quotas for namespaces

### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a namespace test. For more information, see [#unique\\_42](#).
- Connect to the master node SSH IP address of the cluster. For more information, see [#unique\\_10](#).

### Context

By default, a running pod can use the CPU and memory of nodes unlimitedly, which means any pod can use the computing resources of the cluster unlimitedly, and the pods of a namespace may use up the cluster resources.

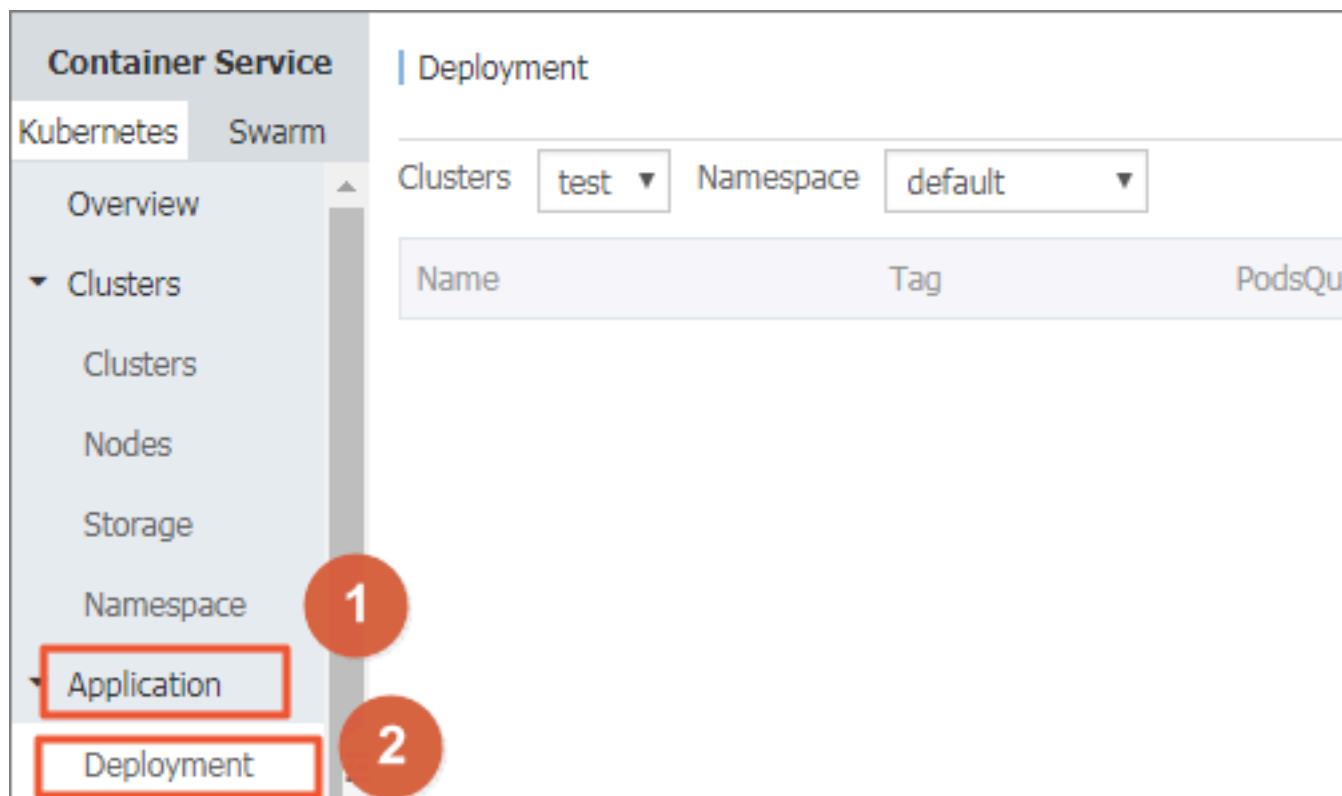
One of the important functions of namespaces is to act as a virtual cluster for multiple purposes and meeting the requirements of multiple users. Therefore, configuring the resource quotas for a namespace is a kind of best practice.

You can configure the resource quotas for a namespace, including CPU, memory, and number of pods. For more information, see [Resource Quotas](#).

### Procedure

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

2. Under Kubernetes, click **Application** > **Deployment** in the left-side navigation pane.  
Click **Create by template** in the upper-right corner.



3. On the Deploy templates page, select the cluster and namespace (test in this example) from the Clusters and Namespace drop-down lists. Use a custom template or the example template Resource – ResourceQuota.

Deploy templates

Only Kubernetes versions 1.8.4 and above are supported. For clusters of version 1.8.1, you

Clusters

test

Namespace

test

Resource Type

Resource - ResourceQuota

Template

```
1  apiVersion: v1
2  kind: ResourceQuota # restrict resource
   roller, pods, service, secret, configma
3  metadata:
4    name: quota
5    # namespace: users-namespace # specif
6  spec:
7    hard:
8      cpu: "2" # adjust limits of cpu fo
9      memory: 4Gi # adjust memory upper
10     requests.storage: 1024G # adjust r
11     persistentvolumeclaims: "50" # adj
12     pods: "50" #adjust number of Pod i
13     replicationcontrollers: "10" # adj
14     services: "10" # adjust number of
15     secrets: "100" # adjust number of
16     configmaps: "100" # adjust number
```

You must configure the ResourceQuota template according to your cluster resources and the plan for the namespace resources. In this example, the template is as follows:

```
apiVersion : v1
kind : ResourceQuota # restrict resource quota for
cpu , memory , storage , pvc , replicationcontroller ,
pods , service , secret , configmap
metadata :
  name : quota
# namespace : users - namespace # specify your namespace
to apply resource quota
spec :
  hard :
    cpu : " 2 " # adjust limits of cpu for your
namespace
    memory : 4Gi # adjust memory upper limits for
your namespace
    requests . storage : 1024G # adjust request of
storage size for your namespace
    persistent volumeclaims : " 50 " # adjust number of
pvc for your namespace
    pods : " 50 " # adjust number of Pod in your
namespace
    replicationcontrollers : " 10 " # adjust number of
ReplicationController in your namespace
    services : " 10 " # adjust number of service for
your namespace
    secrets : " 100 " # adjust number of secrets for
your namespace
    configmaps : " 100 " # adjust number of configmap for
your namespace
```

4. You have configured the resource quotas for this namespace. Connect to the master node SSH IP address and run the following command to view the resource quotas and usage of this namespace.

```
# kubectl describe quota quota -- namespace = test
Name : quota
Namespace : test
Resource      Used      Hard
-----
configmaps    0        100
cpu           0         2
memory        0        4Gi
persistent volumeclaims 0        50
pods          0        50
replicationcontrollers 0        10
requests . storage 0      1024G
secrets       1        100
services      0         10
```

### 1.5.3 Update a namespace

#### Prerequisites

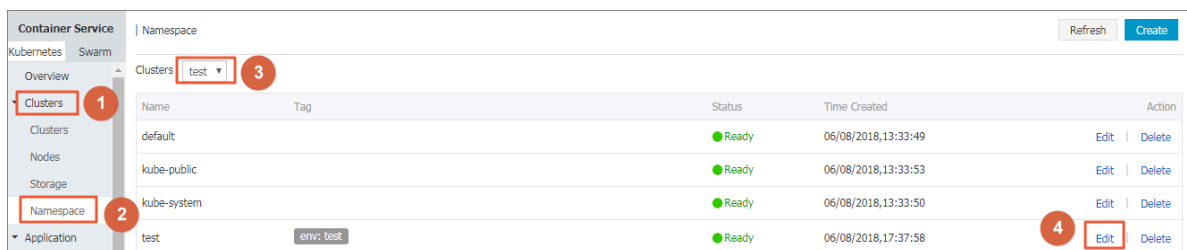
- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a namespace test. For more information, see [#unique\\_42](#).

## Context

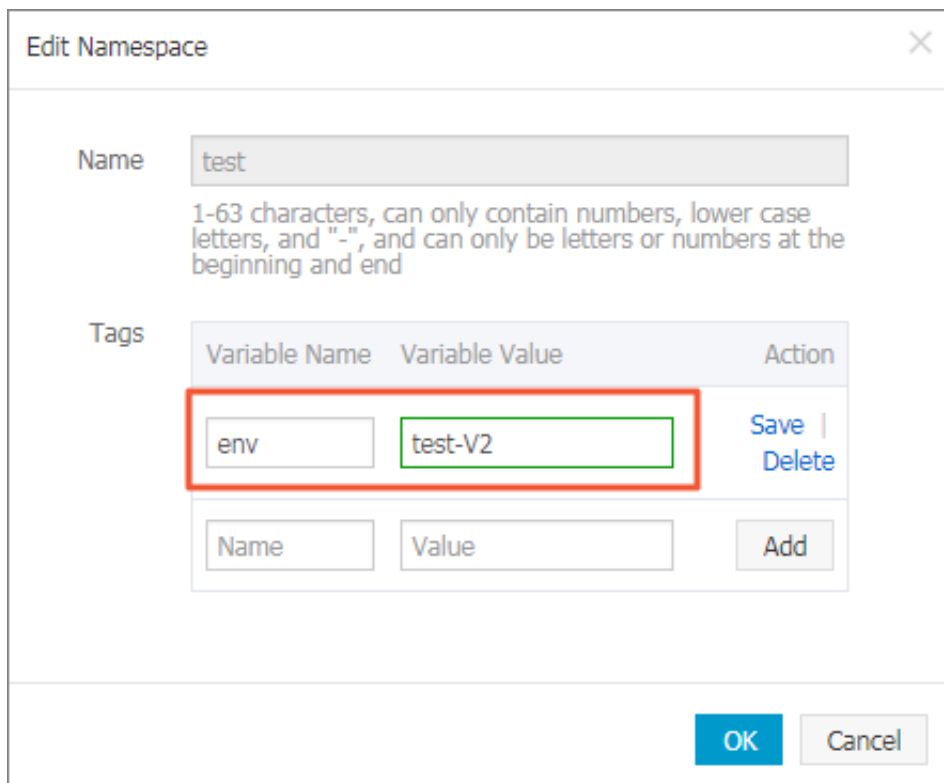
You can update a namespace to add, modify, or delete the namespace tags.

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Clusters > Namespace** in the left-side navigation pane.
3. Select the cluster from the Clusters drop-down list and click **Edit** at the right of the cluster.

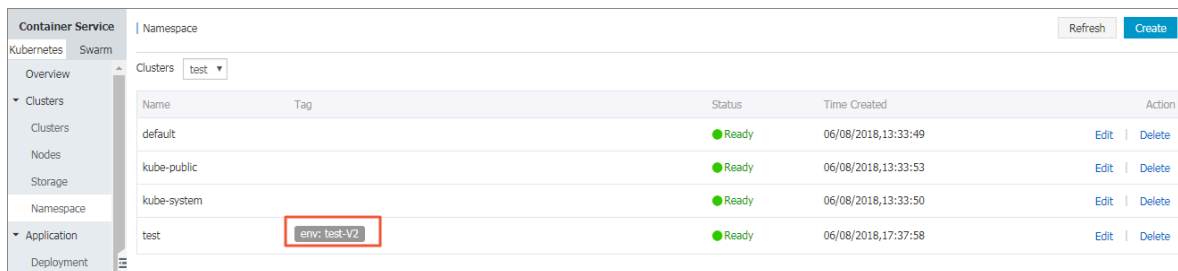


4. Update the namespace tags in the displayed dialog box. For example, change the tag to `env : test - V2`.





5. Click Save on the right and then click OK. The updated namespace tag is displayed in the namespace list.



Name	Tag	Status	Time Created	Action
default		Ready	06/08/2018,13:33:49	Edit   Delete
kube-public		Ready	06/08/2018,13:33:53	Edit   Delete
kube-system		Ready	06/08/2018,13:33:50	Edit   Delete
test	env: test-V2	Ready	06/08/2018,17:37:58	Edit   Delete

## 1.5.4 Delete a namespace

You can delete the namespaces that are no longer in use.

### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a namespace test. For more information, see [#unique\\_42](#).

### Context



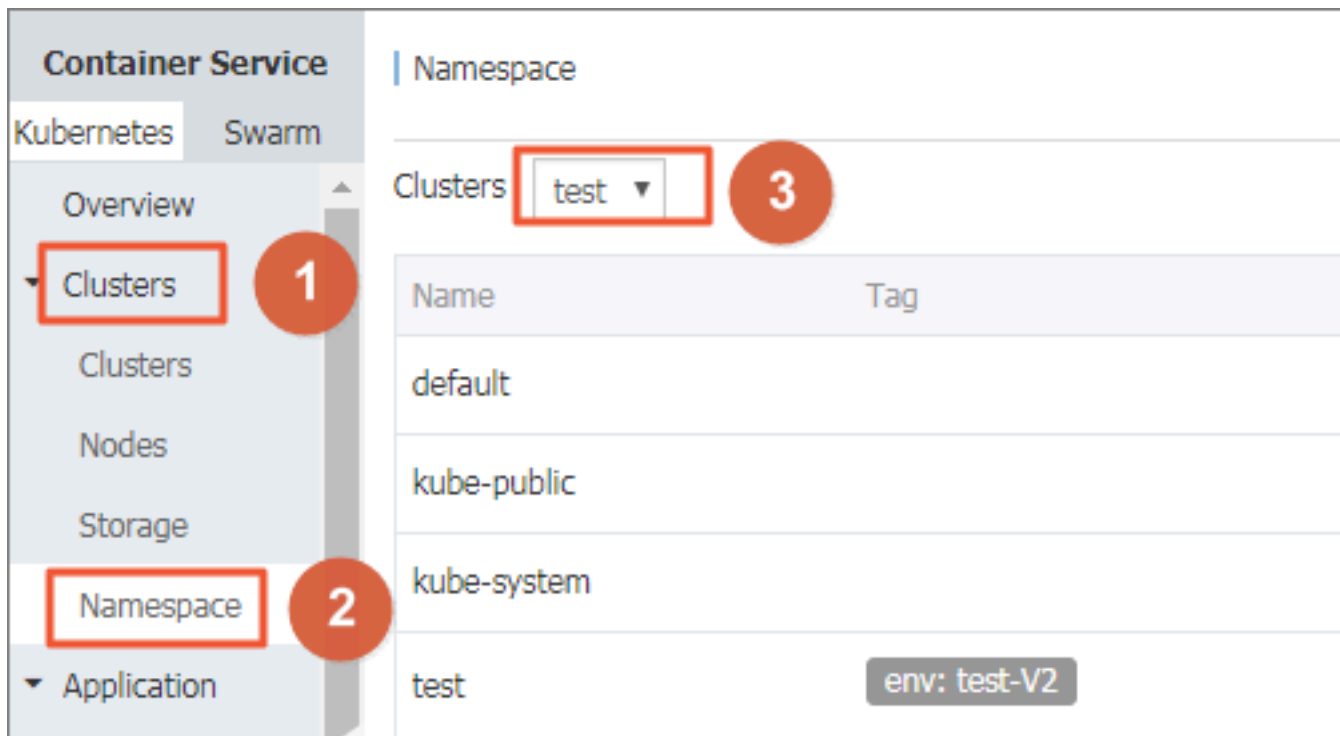
#### Note:

Deleting a namespace also deletes all of its resource objects, so proceed with caution.

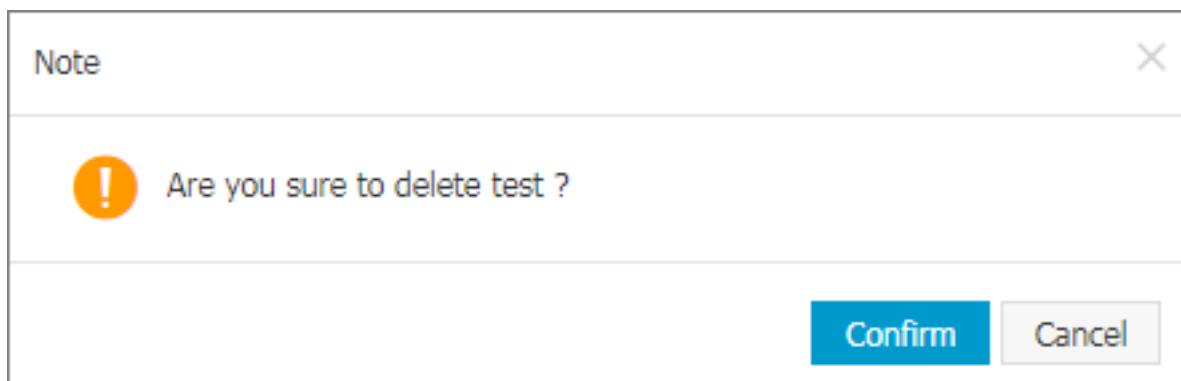
### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters > Namespace in the left-side navigation pane.

3. Select the cluster from the Clusters drop-down list and click Delete at the right of the cluster.



4. Click Confirm in the displayed dialog box.



5. The namespace is deleted from the namespace list and its resource objects are also deleted.

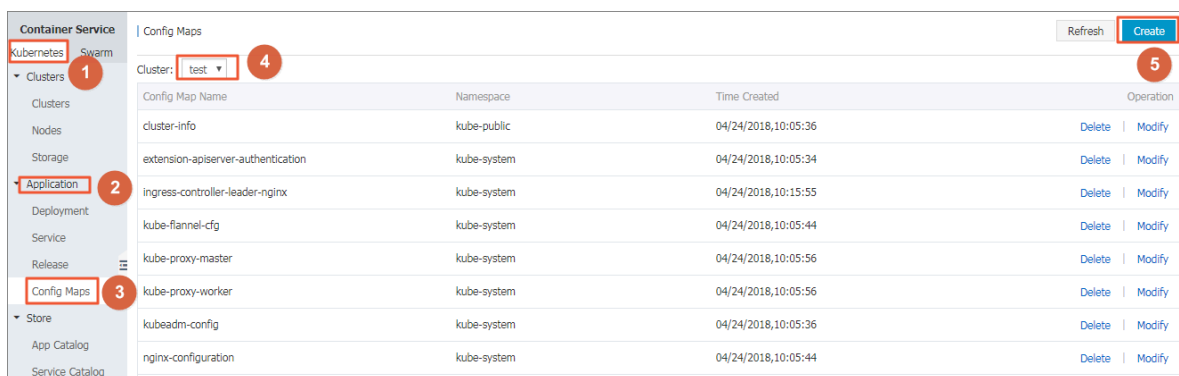
## 1.6 Config map

## 1.6.1 Create a config map

In the Container Service console, you can create a config map on the Config Maps page or by using a template.

Create a config map on Config Maps page

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > Config Maps in the left-side navigation pane.
3. Select the cluster from the Cluster drop-down list. Click Create in the upper-right corner.



4. Complete the settings and then click OK.

- **Namespace:** Select the namespace to which the config map belongs. The config map is a Kubernetes resource object and must act on a namespace.
- **Config Map Name:** Enter the config map name, which can contain lowercase letters, numbers, hyphens (-), and periods (.). The name cannot be empty.

Other resource objects must reference the config map name to obtain the configuration information.

- **Configuration:** Enter the Variable Name and the Variable Value. Then, click Add on the right. You can also click Edit YAML file to set the configurations in the displayed dialog box, and then click OK.

\* Namespace:

default

\* Config Map Name:

test-config

Name must consist of lowercase alphanumeric characters, '-' or '.'. Name cannot be empty.

Configuration:

Variable Name	Variable Value	Action
enemies	aliens	Edit   Delete
lives	3	Edit   Delete

Name

Value

Add

Variable key must be unique. Variable key and value cannot be empty.

Edit YAML file

OK

Cancel

In this example, configure the variables enemies and lives to pass the parameters aliens and 3 respectively.

YAML format

```
1 data:
2   enemies: aliens
3   lives: '3'
4 metadata:
5   name: test-config
6   namespace: default
7
```

\* Configuration must be in YAML format.

OK

Cancel

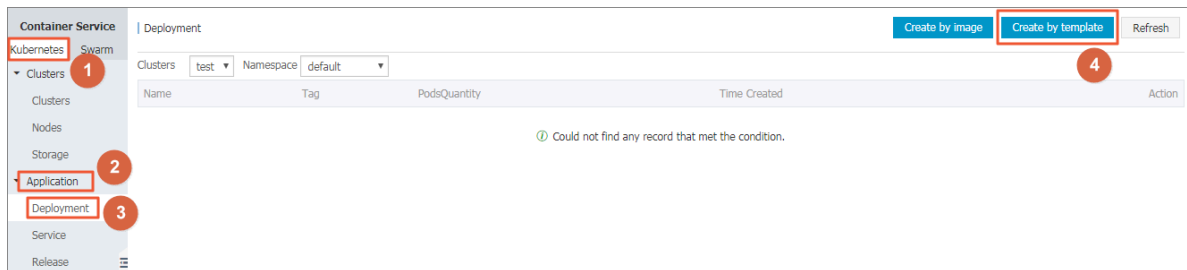
5. You can view the config map test-config on the Config Maps page after clicking OK.

Config Maps				Refresh	Create
Cluster: test					
Config Map Name	Namespace	Time Created	Operation		
test	default	2018-02-09 03:30:31	Delete	Modify	
test-config	default	2018-02-09 05:56:47	Delete	Modify	

## Create a config map by using a template

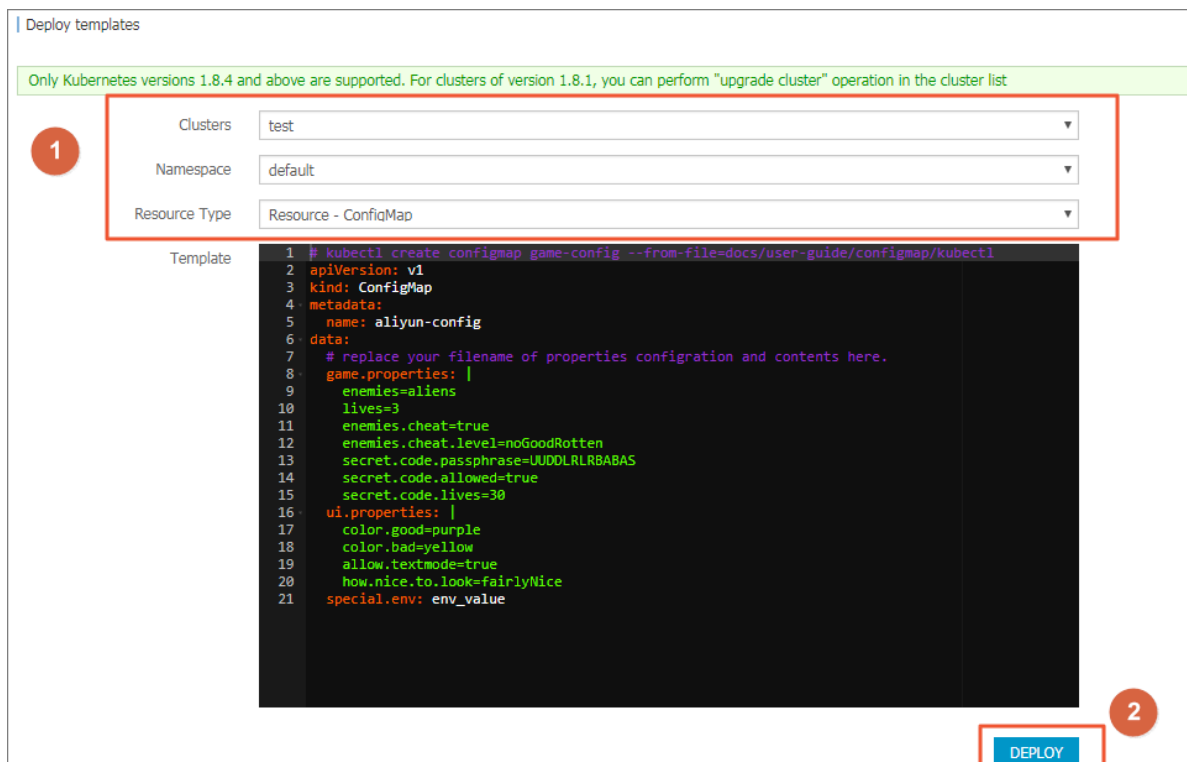
1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > > Deployment in the left-side navigation pane.

### 3. Click Create by template in the upper-right corner.



### 4. On the Deploy templates page, complete the settings and then click DEPLOY.

- **Clusters:** Select the cluster in which the config map is to be created.
- **Namespace:** Select the namespace to which the config map belongs. The config map is a Kubernetes resource object and must act on a namespace.
- **Resource Type:** You can write your own config map based on the Kubernetes YAML syntax rules, or select the sample template Resource - ConfigMap. In the sample template, the config map is named as aliyun-config and includes two variable files `game . properties` and `ui . properties`. You can make modifications based on the sample template. Then, click DEPLOY.



5. After the successful deployment, you can view the config map `aliyun-config` on the Config Maps page.

Config Maps				Refresh	Create
Cluster: test					
Config Map Name	Namespace	Time Created		Operation	
aliyun-config	default	04/24/2018,15:41:32		Delete	Modify

## 1.6.2 Use a config map in a pod

You can use a config map in a pod in the following scenarios:

- Use a config map to define the pod environment variables.
- Use a config map to configure command line parameters.
- Use a config map in data volumes.

For more information, see [Configure a pod to use a ConfigMap](#).

### Limits

To use a config map in a pod, make sure the config map and the pod are in the same cluster and namespace.

### Create a config map

In this example, create a config map `special-config`, which includes two key-value pairs: `SPECIAL_LE VEL : very` and `SPECIAL_TY PE : charm`.

#### Create a config map by using an orchestration template

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Application > > Deployment** Click **Create by template** in the upper-right corner.
3. Select the cluster and namespace from the **Clusters** and **Namespace** drop-down lists. Select a sample template or **Custom** from the **Resource Type** drop-down list. Click **DEPLOY**.

You can use the following **YAML** sample template to create a config map.

```
apiVersion : v1
kind : ConfigMap
metadata :
  name : special - config
  namespace : default
data :
  SPECIAL_LE VEL : very
```

SPECIAL\_TY PE : charm

Create a config map on Config Maps page

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > > Configuration item in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click Create in the upper-right corner.
4. Enter the Config Map Name. Enter the Variable Name and the Variable Value. Then, click Add on the right. Click OK after completing the configurations.

Container Service | Config Map

Kubernetes Swarm

Overview

Clusters

Nodes

Storage

Application

Deployment

Pods

Service

Ingress

Release

Config Maps

Store

App Catalog

Clusters

Namespace

default

\* Config Map Name: special-config

Name must consist of lowercase alphanumeric characters, '-' or '.', Name cannot be empty.

Configuration:

Variable Name	Variable Value	Action
SPECIAL_LEVEL	very	Edit   Delete
SPECIAL_TYPE	charm	Edit   Delete

Name Value Add

Variable key must be unique. Variable key and value cannot be empty.

Edit YAML file

OK Cancel

Use a config map to define pod environment variables

Use config map data to define pod environment variables

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > > Deployment Click Create by template in the upper-right corner.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Select a sample template or Custom from the Resource Type drop-down list. Click DEPLOY.

You can define the environment variables in a pod. Use `valueFrom` to reference the value of SPECIAL\_LEVEL to define the pod environment variables.

See the following orchestration example:

```
apiVersion : v1
kind : Pod
metadata :
  name : config - pod - 1
spec :
  containers :
```



```

- name : test - container
  image : busybox
  command : [ "/ bin / sh ", "- c ", " env " ]
  env :
    - name : SPECIAL_LE VEL_KEY
      valueFrom : ## Use valueFrom to specify env to
reference the value of the config map .
      configMapKeyRef :
        name : special - config ## The referenced
config map name .
        key : SPECIAL_LE VEL ## The referenced config
map key .
  restartPolicy : Never

```

Similarly, to define the values of multiple config maps to the environment variable values of the pod, add multiple env parameters in the pod.

Configure all key-value pairs of a config map to pod environment variables

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > > DeploymentClick Create by template in the upper-right corner.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Select a sample template or Custom from the Resource Type drop-down list. Click DEPLOY.

To configure all the key-value pairs of a config map to the environment variables of a pod, use the envFrom parameter. The key in a config map becomes the environment variable name in the pod.

See the following orchestration example:

```

apiVersion : v1
kind : Pod
metadata :
  name : config - pod - 2
spec :
  containers :
    - name : test - container
      image : busybox
      command : [ "/ bin / sh ", "- c ", " env " ]
      envFrom : ## Reference all the key - value pairs
in the config map special - config .
    - configMapRef :
      name : special - config
      restartPolicy : Never

```

Use a config map to configure command line parameters

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Application > > DeploymentClick Create by template in the upper-right corner.

3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Select a sample template or Custom from the Resource Type drop-down list. Click DEPLOY.

You can use the config map to configure the commands or parameter values in the container by using the environment variable replacement syntax `$( VAR_NAME )`.

See the following orchestration example:

```
apiVersion : v1
kind : Pod
metadata :
  name : config - pod - 3
spec :
  containers :
    - name : test - container
      image : busybox
      command : [ "/ bin / sh ", "- c ", " echo  $( SPECIAL_LE
VEL_KEY ) $( SPECIAL_TY PE_KEY )" ]
      env :
        - name : SPECIAL_LE VEL_KEY
          valueFrom :
            configMapK eyRef :
              name : special - config
              key : SPECIAL_LE VEL
        - name : SPECIAL_TY PE_KEY
          valueFrom :
            configMapK eyRef :
              name : special - config
              key : SPECIAL_TY PE
      restartPol icy : Never
```

The output after running the pod is as follows:

```
very  charm
```

### Use a config map in data volumes

1. Log on to the [Container Service console](#).
2. Under the Kubernetes menu, click Application Deployment in the left-side navigation pane. Click Create by template in the upper-right corner.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Select a sample template or Custom from the Resource Type drop-down list. Click DEPLOY.

You can also use a config map in data volumes. Specifying the config map name under volumes stores the key-value pair data to the mountPath directory (`/ etc /`

`config` in this example). Then, the configuration file with `key` as the name and `value` as the contents is generated.

See the following orchestration example:

```
apiVersion : v1
kind : Pod
metadata :
  name : config - pod - 4
spec :
  containers :
    - name : test - container
      image : busybox
      command : [ "/ bin / sh ", "- c ", " ls / etc / config /" ]
  ## List the file names under this directory .
  volumeMounts :
    - name : config - volume
      mountPath : / etc / config
  volumes :
    - name : config - volume
      configMap :
        name : special - config
  restartPolicy : Never
```

Keys of the config map are output after running the pod.

```
SPECIAL_TY PE
SPECIAL_LE VEL
```

### 1.6.3 Update a config map

You can modify the configurations of a config map.



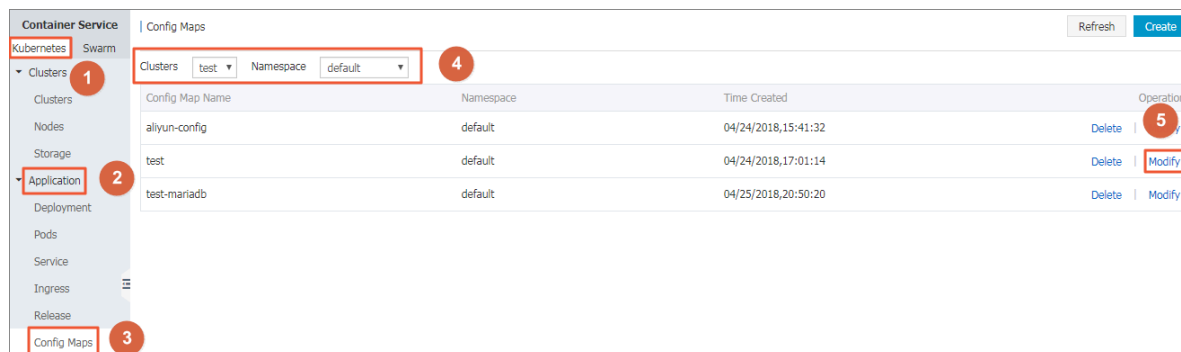
**Note:**

Updating a config map affects applications that use this config map.

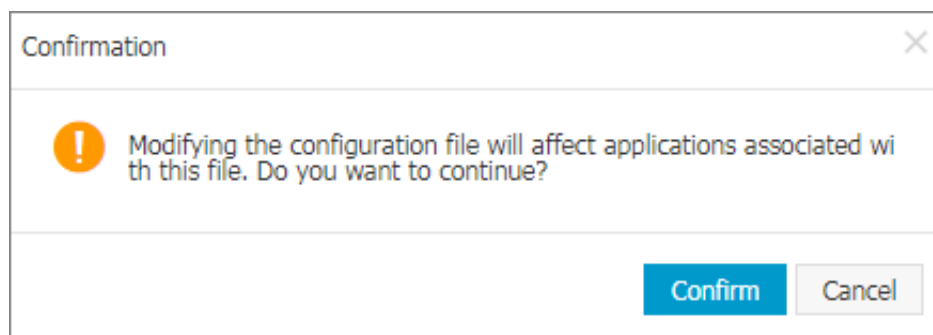
Update a config map on Config Maps page

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Application >> Config Maps** in the left-side navigation pane.

3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click Modify at the right of the config map.

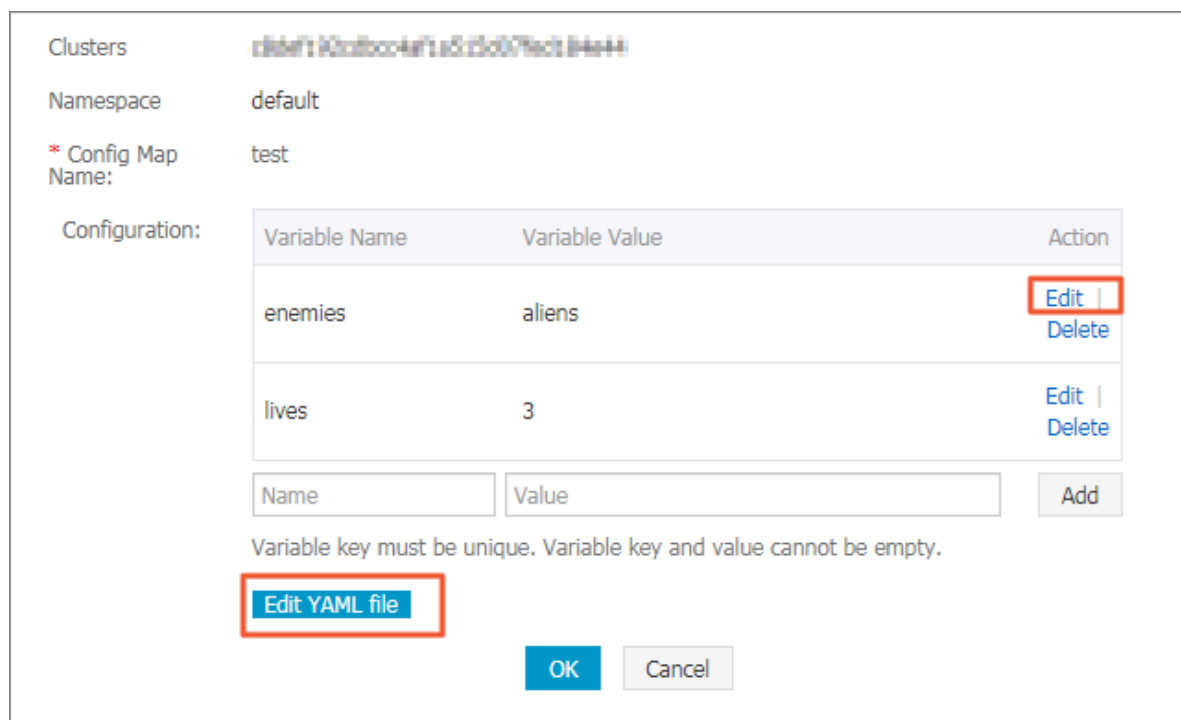


4. Click Confirm in the displayed dialog box.



5. Modify the configurations.

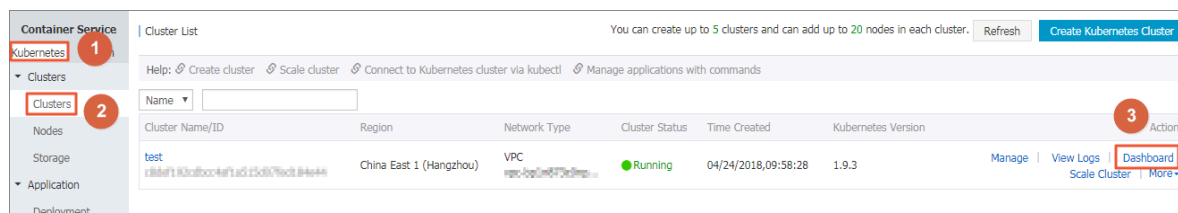
- Click Edit on the right of the configuration you want to modify. Update the configuration and then click Save.
- You can also click Edit YAML file. Click OK after making the modifications.



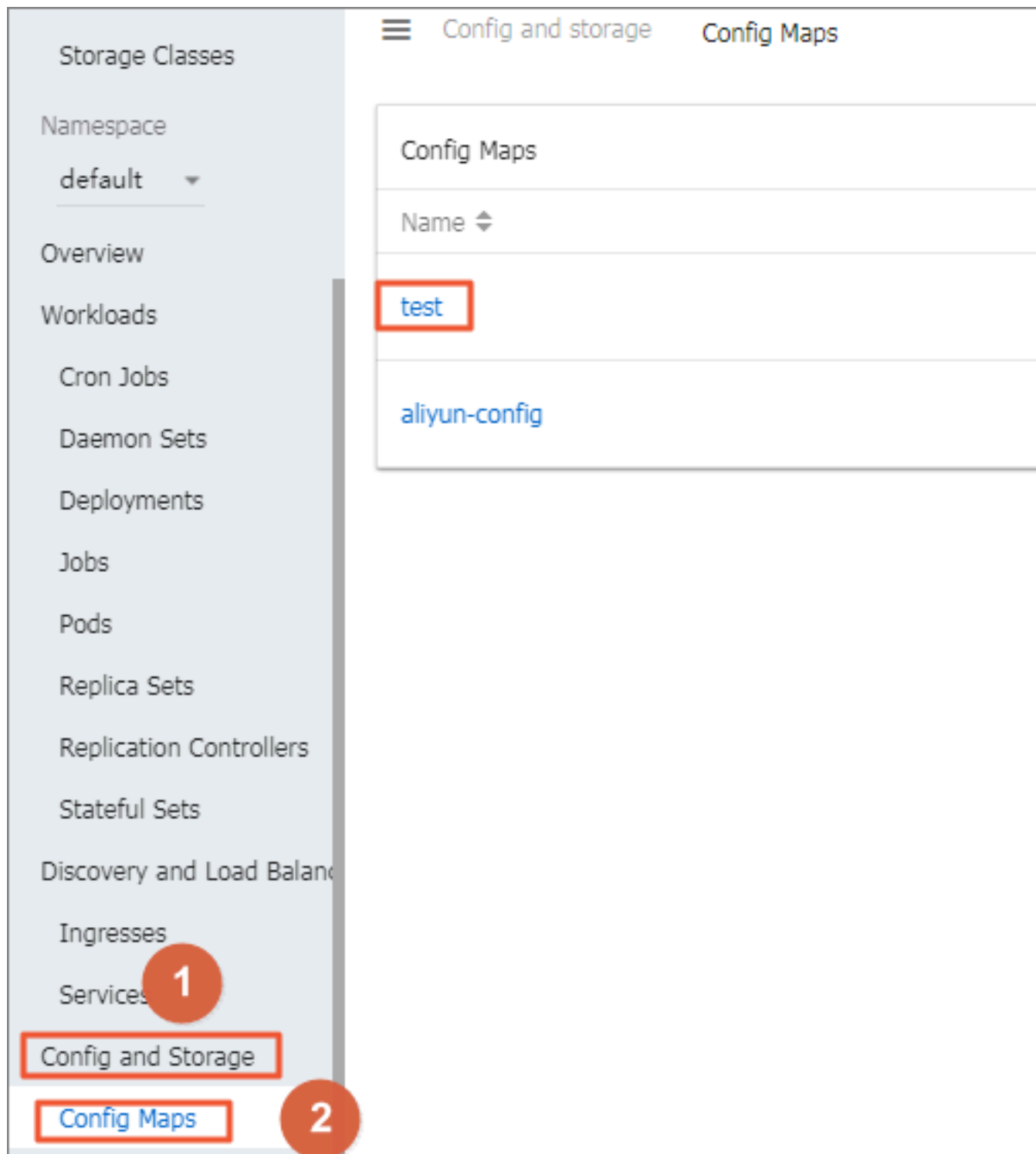
6. After modifying the configurations, click OK.

## Update a config map in Kubernetes dashboard

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click Clusters in the left-side navigation pane.
3. Click Dashboard at the right of the cluster.



4. In the Kubernetes dashboard, click **Config and Storage** > > **Config Maps** in the left-side navigation pane. Click the icon at the right of the config map and then select > **View/edit YAML**.



5. The Edit a Config Map dialog box appears. Modify the configurations and then click UPDATE.



## 1.7 Secrets

### 1.7.1 Create a secret

#### Prerequisites

You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).

#### Context

We recommend that you use secrets for sensitive configurations in Kubernetes clusters, such as passwords and certificates.

Secrets have many types. For example:

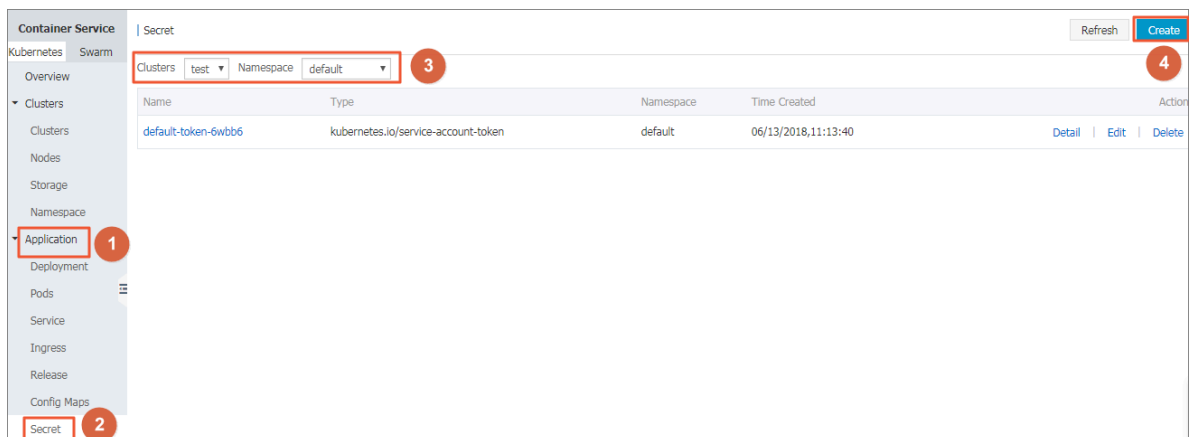
- **Service Account:** Automatically created by Kubernetes, which is used to access Kubernetes APIs and is automatically mounted to the pod directory `/run/secrets/kubernetes.io/serviceaccount`.
- **Opaque:** Secret in the base64 encoding format, which is used to store sensitive information such as passwords and certificates.

By default, you can only create secrets of the Opaque type in the Container Service console. Opaque data is of the map type, which requires the value to be in the base64 encoding format. Alibaba Cloud Container Service supports creating secrets with one click and automatically encoding the clear data to base64 format.

You can also create secrets manually by using command lines. For more information, see [Kubernetes secrets](#).

## Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Application > Secret** in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click **Create** in the upper-right corner.



4. Complete the configurations to create a secret.



**Note:**



To enter the clear data of the secret, select the Encode data values using Base64 check box.

Namespace default

\* Name  1  
Name must consist of lowercase alphanumeric characters, '-' or '.'. Name cannot be empty.

\* Data

Name	Value
<input type="text" value="username"/> 2	<input type="text" value="admin"/>
<input type="text" value="password"/> 3	<input type="text" value="1f2d1e2e67df"/>

Names can only contain numbers, letters, '-', '\_' and '.'.

☒ Encode data values using Base64

- Name: Enter the secret name, which must be 1–253 characters long, and can only contain lowercase letters, numbers, hyphens (-), and dots (.).
  - Configure the secret data. Click the add icon next to Name and enter the name and value of the secret, namely, the key-value pair. In this example, the secret contains two values: `username : admin` and `password : 1f2d1e2e67 df`.
  - Click OK.
5. The Secret page appears. You can view the created secret in the secret list.

Secret					Refresh	Create
Clusters	test	Namespace	default			
Name	Type	Namespace	Time Created	Action		
account	Opaque	default	06/13/2018,11:39:06	Detail	Edit	Delete
default-token-6wbb6	kubernetes.io/service-account-token	default	06/13/2018,11:13:40	Detail	Edit	Delete

## 1.7.2 View secret details

You can view the details of a created secret in the Container Service console.

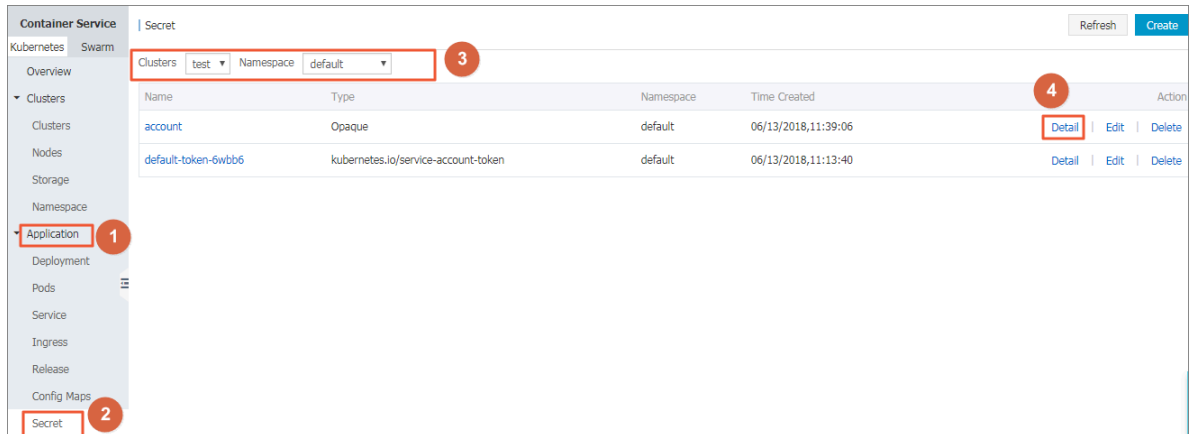
### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a secret. For more information, see [#unique\\_52](#).

## Context

### Procedure

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Application** > **Secret** in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click **Detail** at the right of the secret.



4. You can view the basic information of the secret, and the data that the secret contains.

Click the icon at the right of the data name under **Detail** to view the clear data.



## 1.7.3 Update a secret

You can update an existing secret directly in the Container Service console.

### Prerequisites

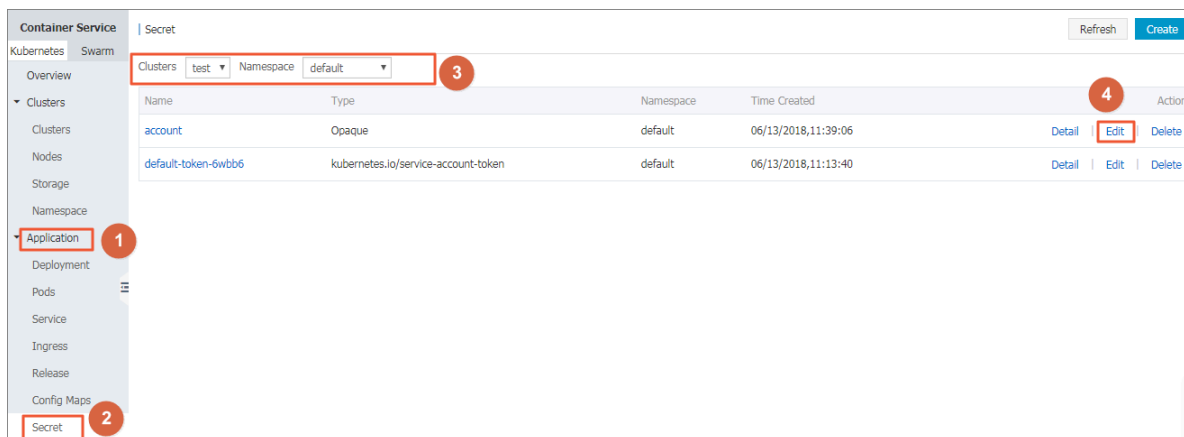
- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a secret. For more information, see [#unique\\_52](#).

## Context

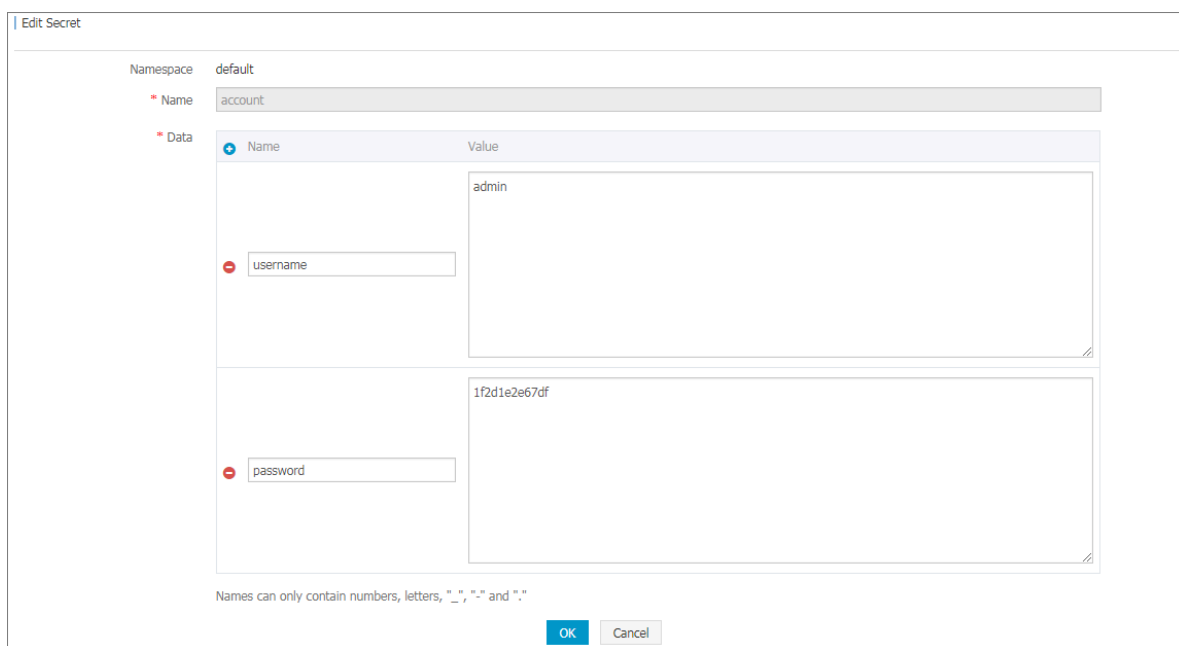
### Procedure

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

2. Under Kubernetes, click **Application** > **Secret** in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click **Edit** at the right of the secret.



4. Update the secret data on the Edit Secret page.



5. Click **OK**.

## 1.7.4 Delete a secret

You can delete an existing secret directly in the Container Service console.

### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- You have created a secret. For more information, see [#unique\\_52](#).

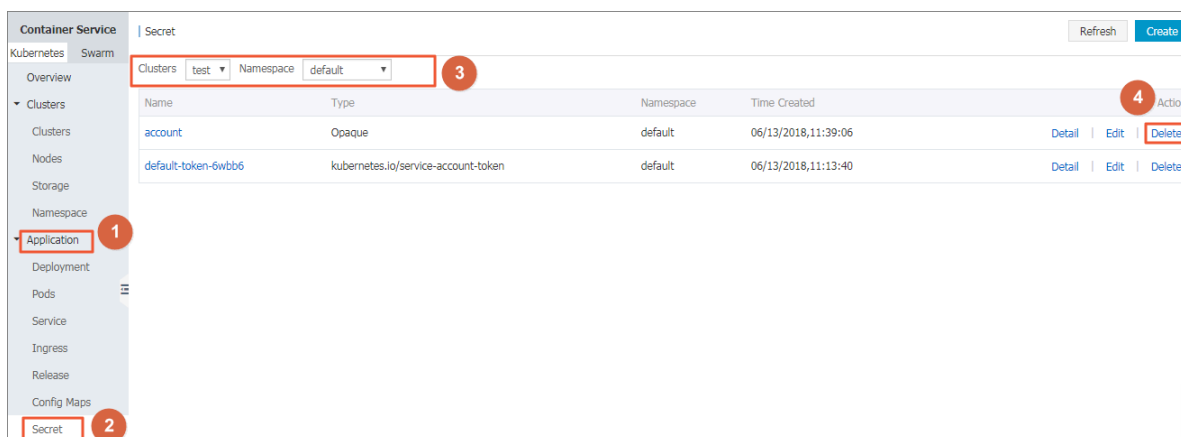
### Context

**Note:**

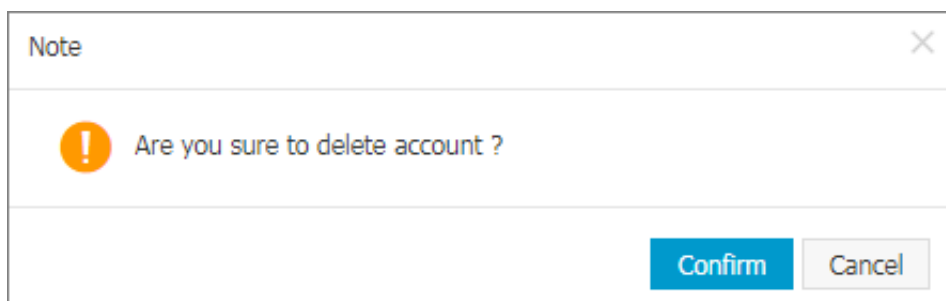
Do not delete the secret generated when the cluster is created.

**Procedure**

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Application > Secret** in the left-side navigation pane.
3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click **Delete** at the right of the secret.



4. Click **Confirm** in the displayed dialog box to delete the secret.



## 1.8 Manage a release

The Alibaba Cloud Kubernetes service enriches the cloud marketplace, in which the app catalog and service catalog functions integrate with the Helm package management tool and allow you to build the application in the cloud quickly. A chart can be released multiple times, which requires you to manage the release versions. Therefore, the Alibaba Cloud Kubernetes service provides the release function, which allows you to manage the applications released by using Helm in the Container Service console.

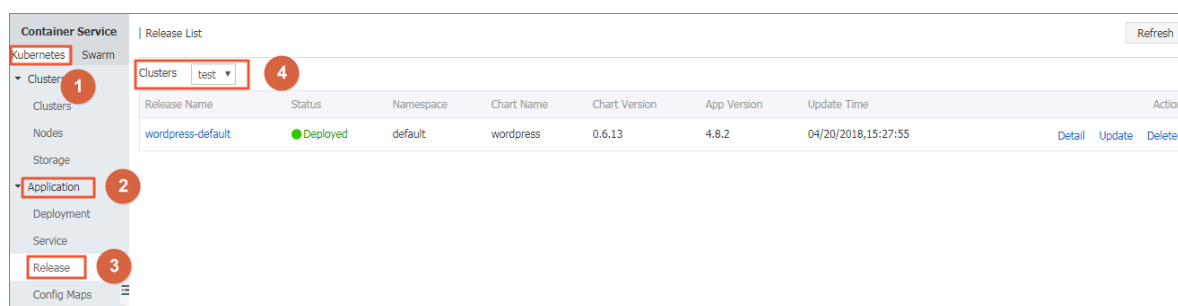
## Prerequisites

- You have created a Kubernetes cluster successfully. For how to create a Kubernetes cluster, see [#unique\\_13](#).
- You have used the app catalog function or service catalog function to install a Helm application. For more information, see [#unique\\_56](#). In this document, use the wordpress-default application as an example.

## View release details

1. Log on to the [Container Service console](#).
2. Click Kubernetes > Application > Release in the left-side navigation pane. Select the cluster from the Clusters drop-down list.

You can view the applications released by using the Helm package management tool and their services in the selected cluster.



3. Take the wordpress-default as an example. Click Detail at the right of the release to view the release details,

such as the current version and history version of this release (in this example, the current version is 1 and no history version exists). You can also view the resource information of wordpress-default, such as the resource name and resource type, and view the YAML information.



Note:

Click the resource name and you are redirected to the Kubernetes dashboard to view the detailed running status of this resource.

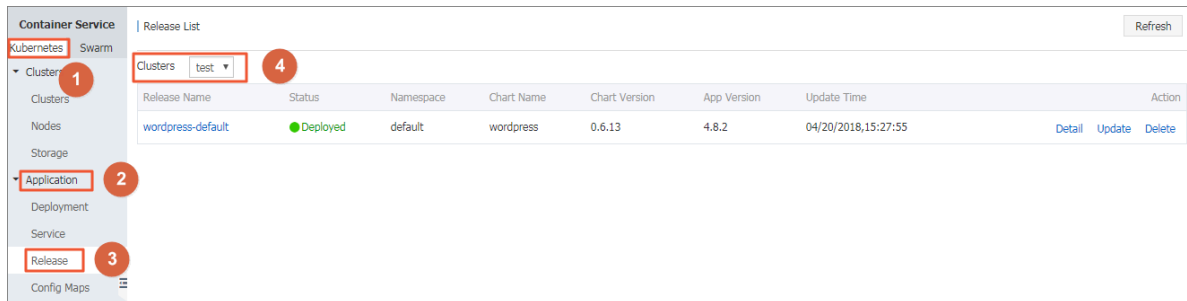
4. Click the Values tab to view the parameter configurations for installing the Helm package.

## Update a release version

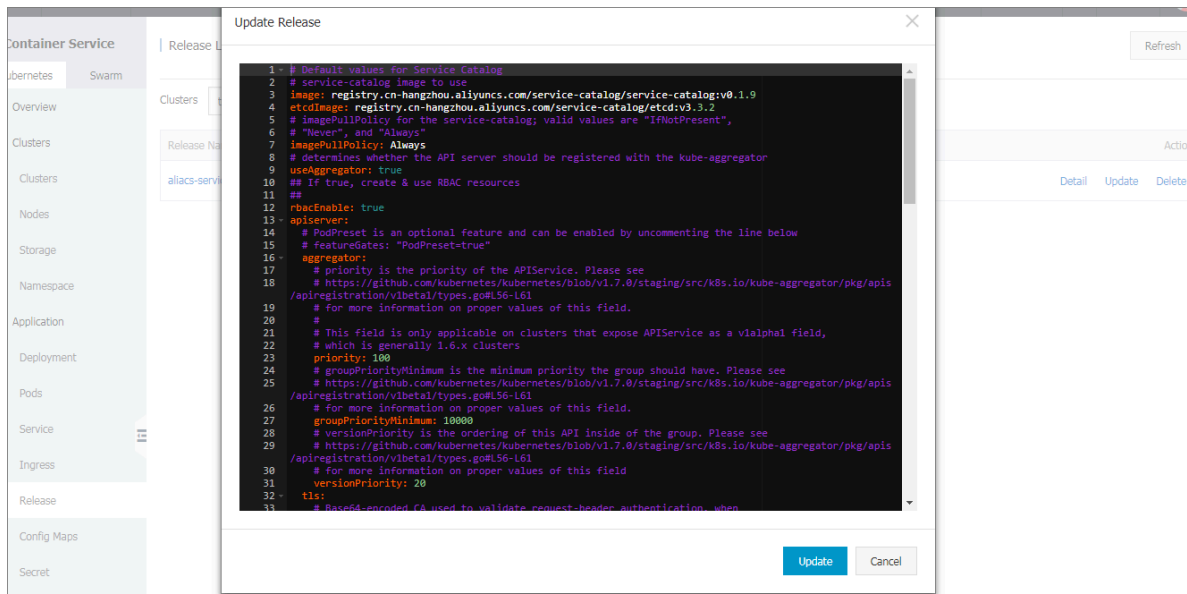
1. Log on to the [Container Service console](#).

- Click **Kubernetes > Application > > Release** in the left-side navigation pane. Select the cluster from the **Clusters** drop-down list.

You can view the applications released by using the Helm package management tool and their services in the selected cluster.



- Take the **wordpress-default** as an example. Click **Update** at the right of the release to update the release. The **Update Release** dialog box appears.



#### 4. Modify the parameters and then click Update.

Update Release

```

1 ## Bitnami WordPress image version
2 ## ref: https://hub.docker.com/r/bitnami/wordpress/tags/
3 ##
4 image: bitnami/wordpress:4.8.2-r0
5
6 ## Specify a imagePullPolicy
7 ## ref: http://kubernetes.io/docs/user-guide/images/#pre-pulling-images
8 ##
9 imagePullPolicy: IfNotPresent
10
11 ## User of the application
12 ## ref: https://github.com/bitnami/bitnami-docker-wordpress#environment
13 ##
14 wordpressUsername: user
15
16 ## Application password
17 ## Defaults to a random 10-character alphanumeric string if not set
18 ## ref: https://github.com/bitnami/bitnami-docker-wordpress#environment
19 ##
20 # wordpressPassword:
21
22 ## Admin email
23 ## ref: https://github.com/bitnami/bitnami-docker-wordpress#environment
24 ##

```

Update

Cancel

The current version is changed to 2 and you can find version 1 under History Version. To roll back the release, click Rollback.

Current Version

Release Name : wordpress-default

Namespace : default

Deployed at : 04/20/2018,17:45:35

Current Version : 2

Time Updated : 04/20/2018,17:45:46

Resource	Kind	Values
wordpress-default-mariadb	Secret	<a href="#">View YAML</a>
wordpress-default-wordpress	Secret	<a href="#">View YAML</a>
wordpress-default-mariadb	ConfigMap	<a href="#">View YAML</a>
wordpress-default-mariadb	PersistentVolumeClaim	<a href="#">View YAML</a>
wordpress-default-wordpress	PersistentVolumeClaim	<a href="#">View YAML</a>
wordpress-default-mariadb	Service	<a href="#">View YAML</a>
wordpress-default-wordpress	Service	<a href="#">View YAML</a>
wordpress-default-mariadb	Deployment	<a href="#">View YAML</a>
wordpress-default-wordpress	Deployment	<a href="#">View YAML</a>

History Version

Version : 1

Rollback

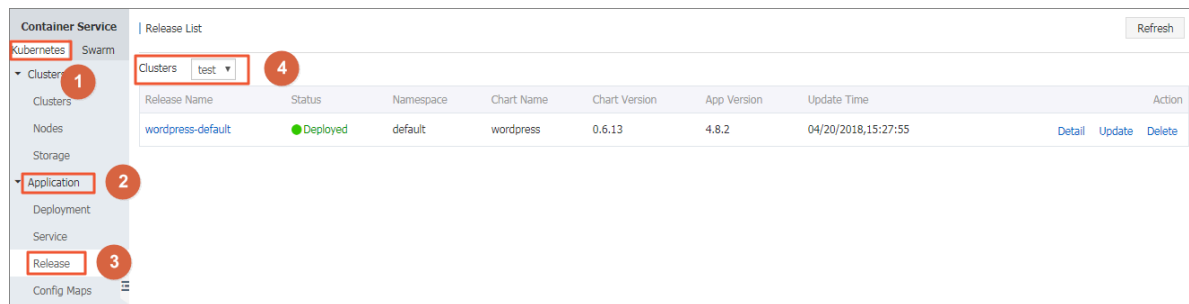
Time Updated : 04/20/2018,17:45:35



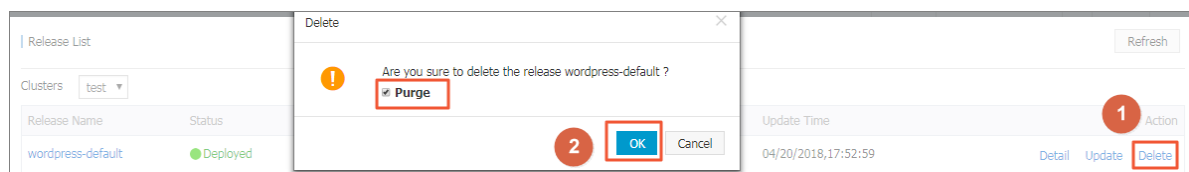
## Delete a release

1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Application > Release** in the left-side navigation pane. Select the cluster from the Clusters drop-down list.

You can view the applications released by using the Helm package management tool and their services in the selected cluster.



3. Take the wordpress-default as an example. Click Delete at the right of the release to delete the release. The Delete dialog box appears.



4. Select the Purge check box to clear the release records if necessary. Click OK to delete the wordpress-default application and its resources such as the services and deployments.

## 1.9 App catalog

### 1.9.1 App catalog overview

Microservice is the theme of container era. The application microservice brings great challenge to the deployment and management. By dividing a large single application into several microservices, the microservice can be independently deployed and extended so as to realize the agile development and fast iteration. Microservice brings great benefits to us. However, developers have to face the management issues of the microservices, such as the resource management, version management, and configuration management. The number of microservices is large because an application is divided into many components that correspond to many microservices.

For the microservice management issues under Kubernetes orchestration, Alibaba Cloud Container Service introduces and integrates with the Helm open-source project to help simplify the deployment and management of Kubernetes applications.

Helm is an open-source subproject in the Kubernetes service orchestration field and a package management tool for Kubernetes applications. Helm supports managing and controlling the published versions in the form of packaging softwares, which simplifies the complexity of deploying and managing Kubernetes applications.

#### Alibaba Cloud app catalog feature

Alibaba Cloud Container Service app catalog feature integrates with Helm, provides the Helm-related features, and extends the features, such as providing graphic interface and Alibaba Cloud official repository.

The chart list on the App Catalog page includes the following information:

- **Chart name:** A Helm package corresponding to an application, which contains the image, dependencies, and resource definition required to run an application.
- **Version:** The version of the chart.
- **Repository:** The repository used to publish and store charts, such as the official repository stable and incubator.

The information displayed on the details page of each chart may be different and include the following items:

- Chart introduction
- Chart details
- Prerequisites for installing chart to the cluster, such as pre-configuring the persistent storage volumes (pv)
- Chart installation commands
- Chart uninstallation commands
- Chart parameter configurations

Currently, you can deploy and manage the charts in the app catalog by using the Helm tool. For more information, see [#unique\\_56](#).

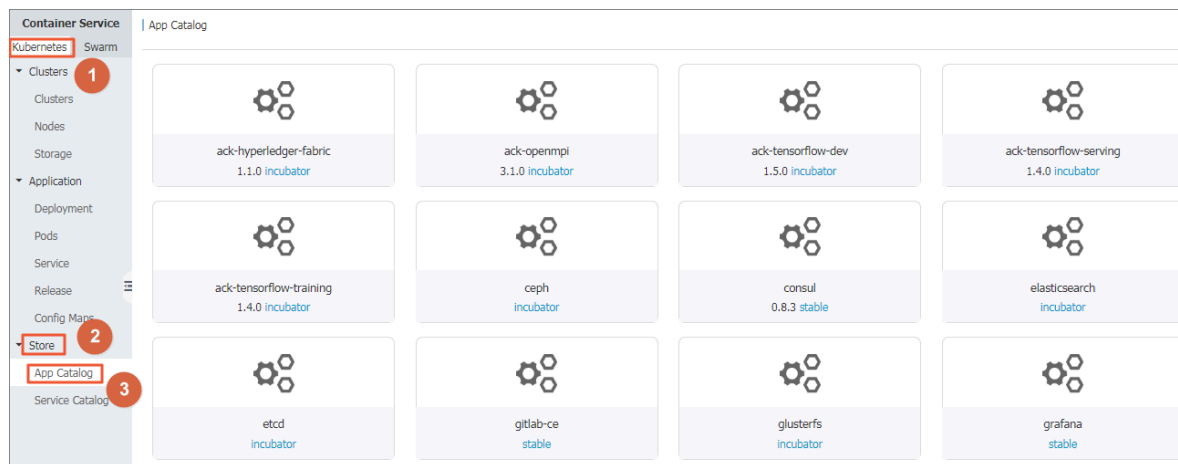
## 1.9.2 View app catalog list

### Procedure

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

2. Click Kubernetes > Store > > App Catalog in the left-side navigation pane.

View the charts on the App Catalog page, each of which corresponds to an application, containing some basic information such as the application name, version, and source repository.



## What's next

You can click to enter a chart and get to know the detailed chart information. Deploy the application according to the corresponding information by using the Helm tool. For more information, see [#unique\\_56](#).

## 1.10 Plan Kubernetes CIDR blocks under VPC

Generally, you can select to create a Virtual Private Cloud (VPC) automatically and use the default network address when creating a Kubernetes cluster in Alibaba Cloud. In some complicated scenarios, plan the Elastic Compute Service (ECS) address, Kubernetes pod address, and Kubernetes service address on your own. This document introduces what the addresses in Kubernetes under Alibaba Cloud VPC environment are used for and how to plan the CIDR blocks.

### Basic concepts of Kubernetes CIDR block

The concepts related to IP address are as follows:

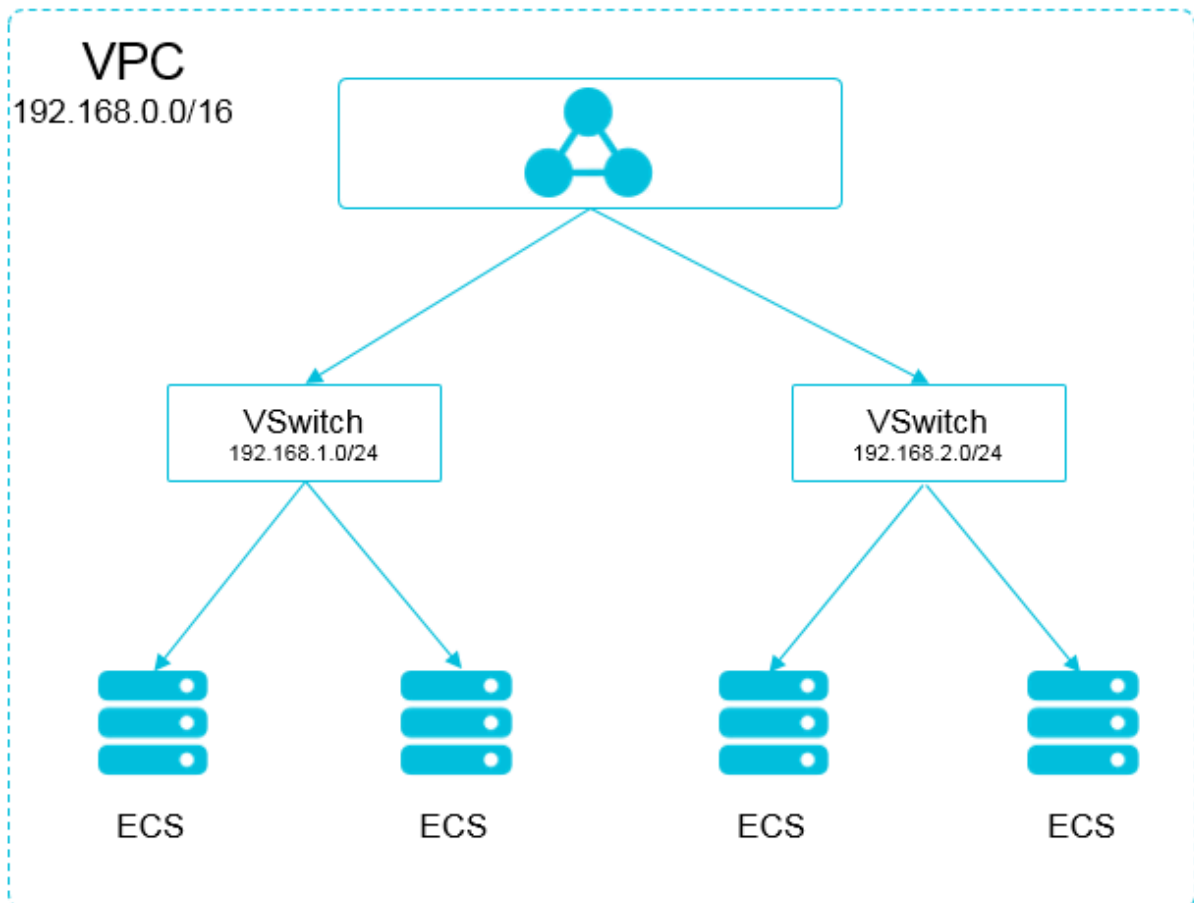
#### VPC CIDR block

The CIDR block selected when you create a VPC. Select the VPC CIDR block from 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.

#### VSwitch CIDR block

The CIDR block specified when you create a VSwitch in VPC. The VSwitch CIDR block must be the subset of the current VPC CIDR block, which can be the same as the VPC CIDR block but cannot go beyond that range. The address assigned to the ECS instance under the VSwitch is obtained from the VSwitch CIDR block. Multiple VSwitches can be created under one VPC, but the VSwitch CIDR blocks cannot overlap .

The VPC CIDR block structure is as follows.



### Pod CIDR block

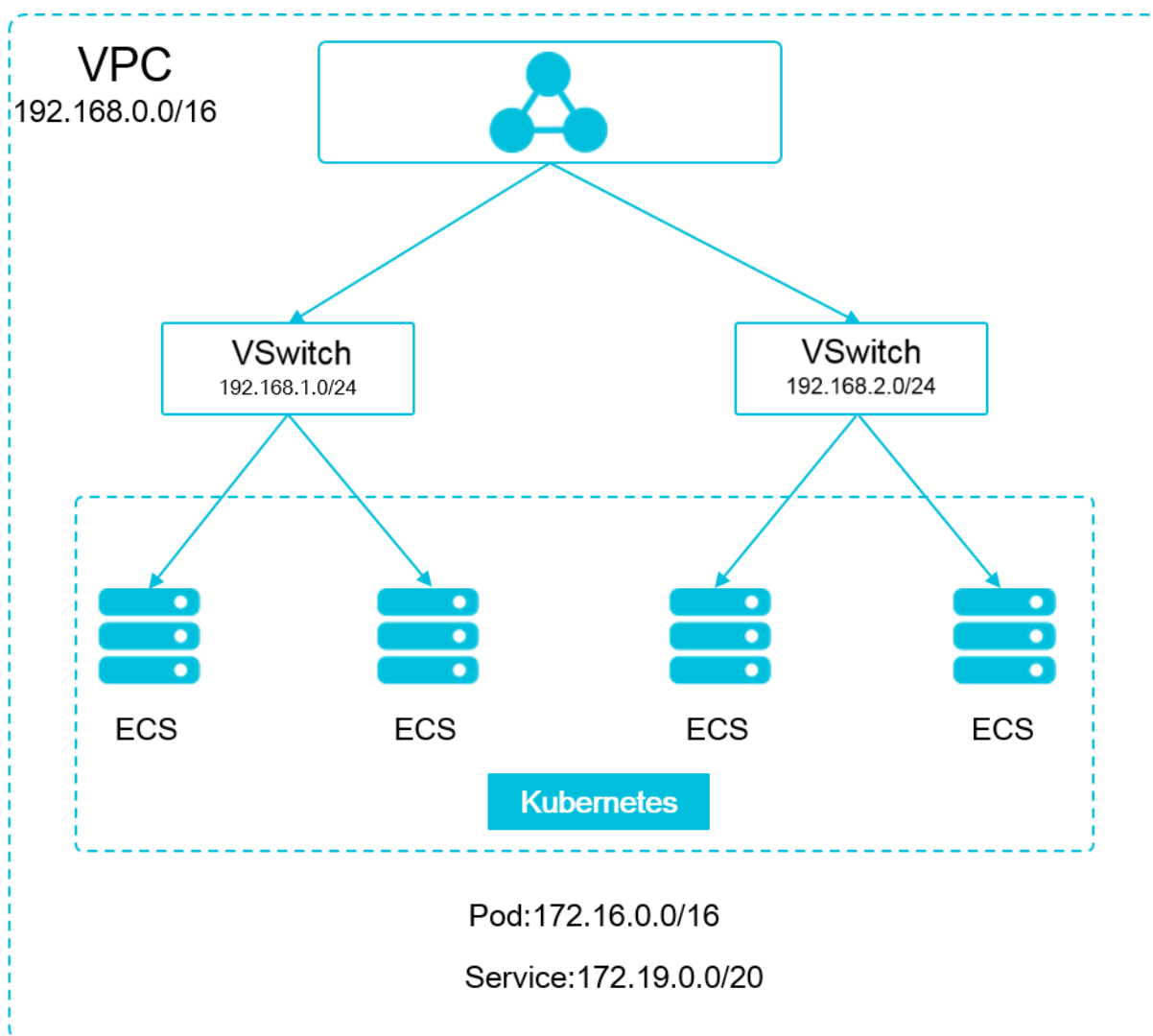
Pod is a concept in Kubernetes. Each pod has one IP address. You can specify the pod CIDR block when creating a Kubernetes cluster in Alibaba Cloud Container Service and the pod CIDR block cannot overlap with the VPC CIDR block. For example, if the VPC CIDR block is 172.16.0.0/12, then the pod CIDR block of Kubernetes cannot use 172.16.0.0/16, 172.17.0.0/16, or any address that is included in 172.16.0.0/12.

### Service CIDR block

Service is a concept in Kubernetes. Each service has its own address. The service CIDR block cannot overlap with the VPC CIDR block or pod CIDR block. The service

address is only used in a Kubernetes cluster and cannot be used outside a Kubernetes cluster.

The relationship between Kubernetes CIDR block and VPC CIDR block is as follows.



### How to select CIDR block

#### Scenario of one VPC and one Kubernetes cluster

This is the simplest scenario. The VPC address is determined when the VPC is created. Select a CIDR block different from that of the current VPC when creating a Kubernetes cluster.

#### Scenario of one VPC and multiple Kubernetes clusters

Create multiple Kubernetes clusters under one VPC. In the default network mode (Flannel), the pod message needs to be routed by using VPC, and Container Service automatically configures the route table to each pod CIDR block on the VPC route.

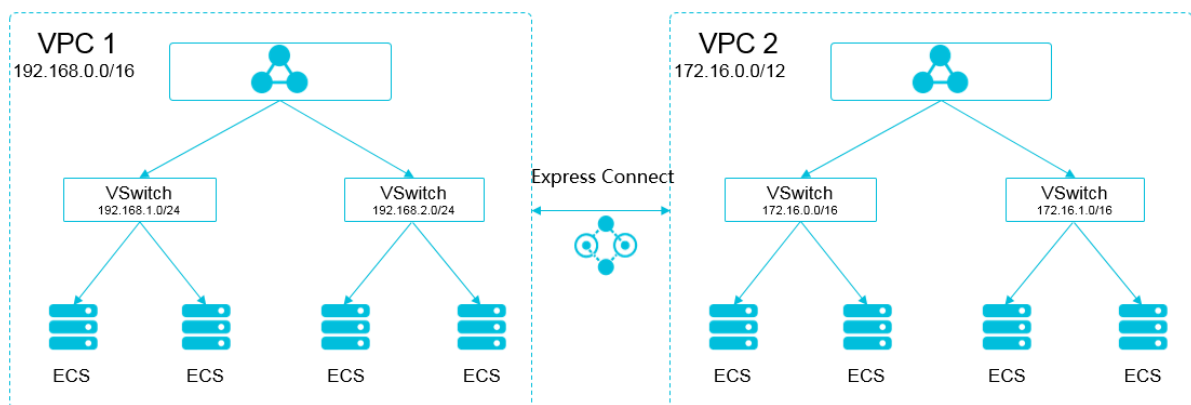
The pod CIDR blocks of all the Kubernetes clusters cannot overlap, but the service CIDR blocks can overlap.

The VPC address is determined when the VPC is created. Select a CIDR block that does not overlap with the VPC address or other pod CIDR blocks for each Kubernetes cluster when creating a Kubernetes cluster.

In such a situation, parts of the Kubernetes clusters are interconnected. The pod of one Kubernetes cluster can directly access the pod and ECS instance of another Kubernetes cluster, but cannot access the service of another Kubernetes cluster.

#### Scenario of VPC interconnection

You can configure what messages are to be sent to the opposite VPC by using route tables when two VPCs are interconnected. Take the following scenario as an example: VPC 1 uses the CIDR block 192.168.0.0/16 and VPC 2 uses the CIDR block 172.16.0.0/12. By using route tables, specify to send the messages of 172.16.0.0/12 in VPC 1 to VPC 2.



In such a situation, the CIDR block of the Kubernetes cluster created in VPC 1 cannot overlap with VPC 1 CIDR block or the CIDR block to be routed to VPC 2. The same applies to the scenario when you create a Kubernetes cluster in VPC 2. In this example, the pod CIDR block of the Kubernetes cluster can select a sub-segment under 10.0.0.0/8.



#### Note:

The CIDR block routing to VPC 2 can be considered as an occupied address. Kubernetes clusters cannot overlap with an occupied address.

To access the Kubernetes pod of VPC 1 in VPC 2, configure the route to the Kubernetes cluster in VPC 2.

## Scenario of VPC to IDC

Similar to the scenario of VPC interconnection, if parts of the CIDR blocks in VPC route to IDC, the pod address of Kubernetes clusters cannot overlap with those addresses. To access the pod address of Kubernetes clusters in IDC, configure the route table to leased line virtual border router (VBR) in IDC.

## 1.11 Server Load Balancer

### 1.11.1 Overview

Kubernetes clusters provide a diversity of approaches to access container applications, and support accessing internal services and realizing load balancing by means of Alibaba Cloud Server Load Balancer or Ingress.

### 1.11.2 Access services by using Server Load Balancer

You can access services by using Alibaba Cloud Server Load Balancer.



#### Note:

If cloud-controller-manager of your cluster is in v 1.9.3 or later versions, when you specify an existing SLB, the system does not process listeners for this SLB by default. You have to manually configure listeners for this SLB.

To view the version of cloud-controller-manager, execute the following command:

```
root @ master # kubectl get po -n kube-system -o
yaml | grep image : | grep cloud-controller-manager | uniq

image : registry-vpc.cn-hangzhou.aliyuncs.com/acs/cloud-controller-manager-amd64:v1.9.3
```

#### Operate by using command line

##### 1. Create an Nginx application by using command line.

```
root @ master # kubectl run nginx --image=registry.aliyuncs.com/acs/netdia:latest
root @ master # kubectl get po
```

NAME	READY	STATUS
RESTARTS	AGE	

```

nginx - 2721357637 - dvwq3          1 / 1      Running
1                               6s

```

2. Create Alibaba Cloud Server Load Balancer service for the Nginx application and specify `type = LoadBalancer` to expose the Nginx service to the Internet.

```

root @ master # kubectl expose deployment nginx -- port =
80 -- target - port = 80 -- type = LoadBalancer
root @ master # kubectl get svc
NAME                                CLUSTER - IP          EXTERNAL - IP
PORT ( S )                        AGE
nginx                              172 . 19 . 10 . 209    101 . 37 . 192 . 20
80 : 31891 / TCP                    4s

```

3. Visit `http://101.37.192.20` in the browser to access your Nginx service.

### Operate by using Kubernetes dashboard

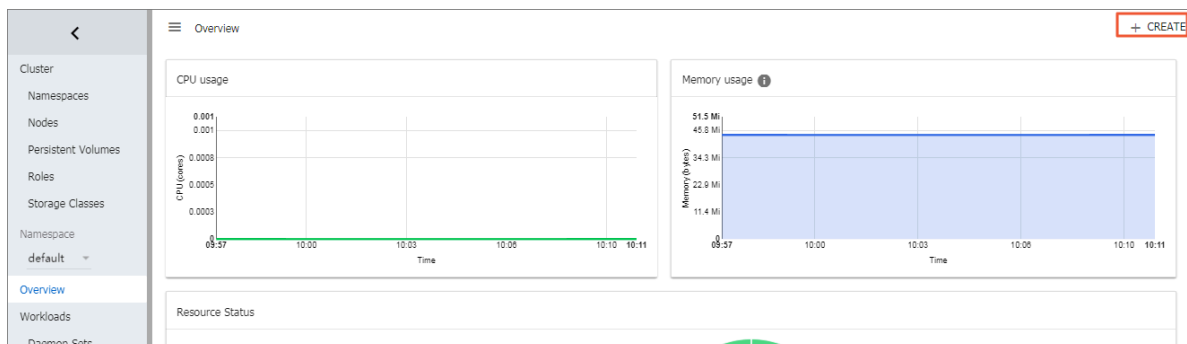
1. Save the following yml codes to the `nginx - svc . yml` file.

```

apiVersion : v1
kind : Service
metadata :
  labels :
    run : nginx
  name : http - svc
  namespace : default
spec :
  ports :
    - port : 80
      protocol : TCP
      targetPort : 80
  selector :
    run : nginx
  type : LoadBalancer

```

2. Log on to the [Container Service console](#). Click Dashboard at the right of a cluster.
3. Click CREATE in the upper-right corner to create an application.



4. Click the CREATE FROM FILE tab. and then upload the `nginx - svc . yml` file you saved.

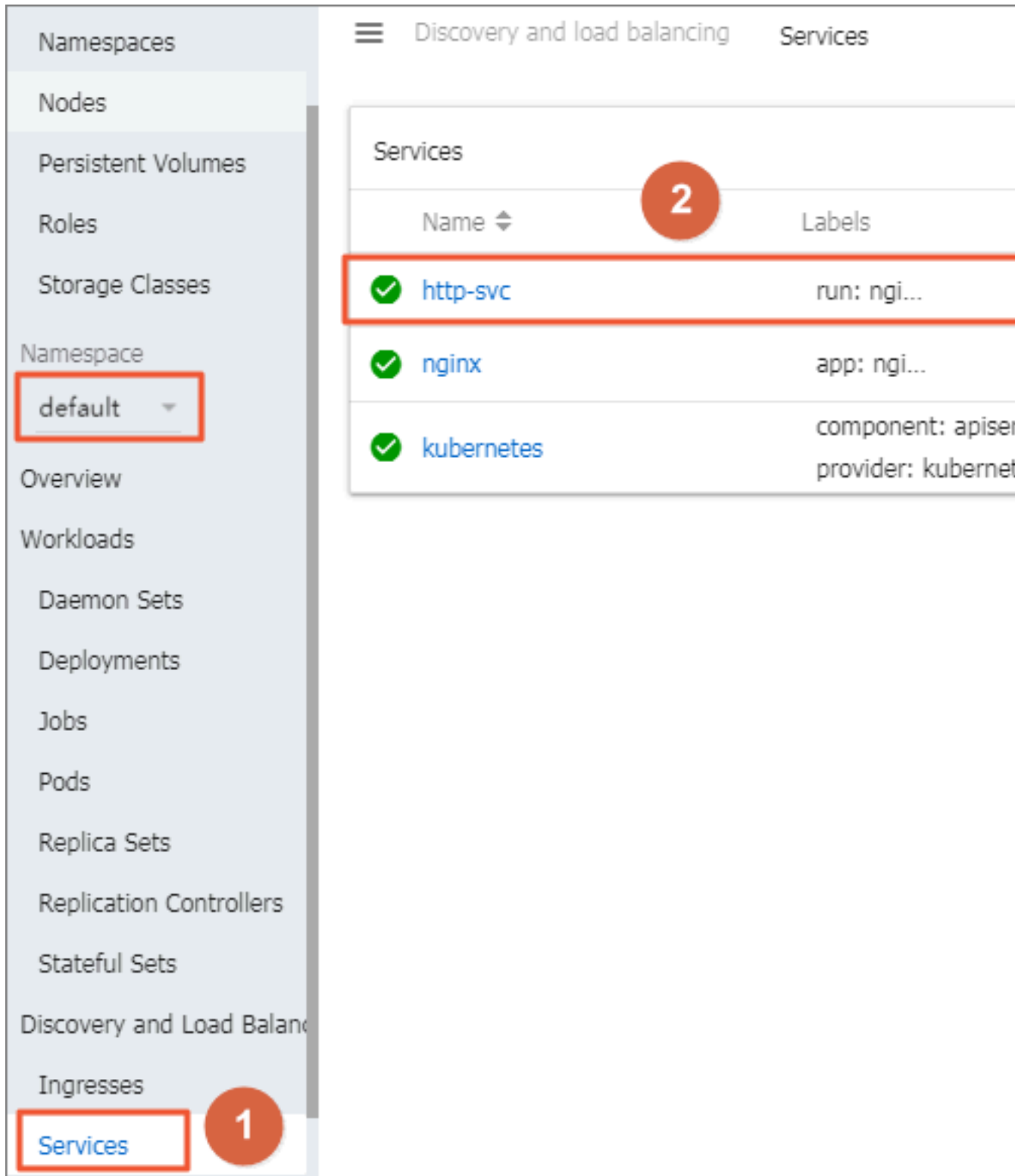


**5. Click UPLOAD.**

A Nginx application specified by Alibaba Cloud Server Load Balancer instance is created. The service name is `http - svc` .

6. Select default under Namespace in the left-side navigation pane. Click Services in the left-side navigation pane.

You can view the created Nginx service `http - svc` and the Server Load Balancer address `http :// 114 . 55 . 79 . 24 : 80`.



7. Copy the address to the browser to access the service.

## More information

Alibaba Cloud Server Load Balancer also supports parameter configurations such as health check, billing method, and load balancing. For more information, see [Server Load Balancer configuration parameters](#).

## Annotations

Alibaba Cloud supports a lot of Server Load Balancer features by using annotations.

Use existing intranet Server Load Balancer instance

You must specify two annotations. Replace with your own Server Load Balancer instance ID.

```
apiVersion : v1
kind : Service
metadata :
  annotation s :
    service . beta . kubernetes . io / alicloud - loadbalanc er -
address - type : intranet
    service . beta . kubernetes . io / alicloud - loadbalanc er - id
: your - loadbalanc er - id
  labels :
    run : nginx
    name : nginx
    namespace : default
spec :
  ports :
    - name : web
      port : 80
      protocol : TCP
      targetPort : 80
    selector :
      run : nginx
    sessionAff inity : None
    type : LoadBalanc er
```

Save the preceding contents as `slb.svc` and then run the command `kubectl apply`

```
- f slb . svc .
```

Create an HTTPS type Server Load Balancer instance

Create a certificate in the Alibaba Cloud console and record the cert-id. Then, use the following annotation to create an HTTPS type Server Load Balancer instance.

```
apiVersion : v1
kind : Service
metadata :
  annotation s :
    service . beta . kubernetes . io / alicloud - loadbalanc er -
cert - id : your - cert - id
    service . beta . kubernetes . io / alicloud - loadbalanc er -
protocol - port : " https : 443 "
  labels :
```

```

    run : nginx
    name : nginx
    namespace : default
  spec :
    ports :
    - name : web
      port : 443
      protocol : TCP
      targetPort : 443
    selector :
      run : nginx
    sessionAffinity : None
    type : LoadBalancer

```

**Note:**

Annotations are case sensitive.

Annotation	Description	Default value
service.beta.kubernetes.io/alibabacloud-loadbalancer-protocol-port	Use commas (,) to separate multiple values. For example, https:443,http:80.	N/A
service.beta.kubernetes.io/alibabacloud-loadbalancer-address-type	The value is Internet or intranet.	internet
service.beta.kubernetes.io/alibabacloud-loadbalancer-slb-network-type	Server Load Balancer network type. The value is classic or VPC.	classic
service.beta.kubernetes.io/alibabacloud-loadbalancer-charge-type	The value is paybytraffic or paybybandwidth.	paybybandwidth
service.beta.kubernetes.io/alibabacloud-loadbalancer-id	The Server Load Balancer instance ID. Specify an existing Server Load Balance with the loadbalancer-id, and the existing listener is overwritten. Server Load Balancer is not deleted when the service is deleted.	N/A
service.beta.kubernetes.io/alibabacloud-loadbalancer-backend-label	Use label to specify which nodes are mounted to the Server Load Balancer backend.	N/A

Annotation	Description	Default value
service.beta.kubernetes.io/alibabacloud-loadbalancer-region	The region in which Server Load Balancer resides.	N/A
service.beta.kubernetes.io/alibabacloud-loadbalancer-bandwidth	Server Load Balancer bandwidth.	50
service.beta.kubernetes.io/alibabacloud-loadbalancer-cert-id	Authentication ID on Alibaba Cloud. Upload the certificate first.	“”
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-flag	The value is on or off.	The default value is off. No need to modify the TCP parameters because TCP enables health check by default and you cannot configure it.
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-type	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-uri	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-connect-port	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-healthy-threshold	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-unhealthy-threshold	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-interval	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	
service.beta.kubernetes.io/alibabacloud-loadbalancer-health-check-connect-timeout	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	

Annotation	Description	Default value
service.beta.kubernetes.io/alibaba-loadbalancer-health-check-timeout	See <a href="#">../SP_23/DNSLB11870158/EN-US_TP_4205.dita#doc_api_Slb_CreateLoadBalancerTCPListener</a> .	

### 1.11.3 Configure Ingress monitoring

You can view the Ingress monitoring data by enabling the default VTS module of Ingress.

Enable VTS module by running commands

1. Modify the Ingress ConfigMap configuration to add the configuration item `enable`

```
- vts - status : " true ".
```

```
root @ master # kubectl edit configmap nginx - configurat
ion - n kube - system
configmap " nginx - configurat ion " edited
```

After the modification, the contents of the Ingress ConfigMap are as follows:

```
apiVersion : v1
data :
  enable - vts - status : " true " # Enable VTS module
  proxy - body - size : 20m
kind : ConfigMap
metadata :
  annotation s :
    kubectl . kubernetes . io / last - applied - configurat ion :
|

  creationTimestamp : 2018 - 03 - 20T07 : 10 : 18Z
  labels :
    app : ingress - nginx
  name : nginx - configurat ion
  namespace : kube - system
  selfLink : / api / v1 / namespaces / kube - system / configmaps /
nginx - configurat ion
```

2. Verify if Ingress Nginx has enabled the VTS module normally.

```
root @ master # kubectl get pods -- selector = app = ingress
- nginx - n kube - system
NAME READY STATUS RESTARTS AGE
nginx - ingress - controller - 79877595c8 - 78gq8 1 / 1 Running
0 1h
root @ master # kubectl exec - it nginx - ingress -
controller - 79877595c8 - 78gq8 - n kube - system -- cat
/ etc / nginx / nginx . conf | grep vhost_traf fic_status
_display
vhost_traf fic_status _display ;
```

```
vhost_traf fic_status _display_f ormat html ;
```

### 3. Locally access the Ingress Nginx monitoring console.



#### Note:

By default, the VTS port is not opened for security considerations. Here use the port-forward method to access the console.

```
root @ master # kubectl port - forward nginx - ingress -
controller - 79877595c8 - 78gq8 - n kube - system 18080
Forwarding from 127 . 0 . 0 . 1 : 18080 -> 18080
Handling connection for 18080
```

### 4. Use `http :// localhost : 18080 / nginx_stat us` to access the VTS monitoring console.

## Nginx Vhost Traffic Status

### Server main

Host	Version	Uptime	Connections				Requests				Shared memory			
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode
nginx-ingress-controller-79877595c8-78gq8	1.13.7	32m 41s	7	0	1	6	93566	93566	1428	1	vhost_traffic_status	10.0 MiB	2.4 KiB	1

### Server zones

Zone	Requests			Responses					Traffic					Cache								
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarce	Total
-	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0
*	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0

### Upstreams

#### upstream-default-backend

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses						Traffic						
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s			
172.16.3.6:8080	up	0ms	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

update interval: 1 sec

[JSON](#) | [GITHUB](#)

## Enable VTS module by using the Kubernetes dashboard

1. Log on to the [Container Service console](#).
2. On the Cluster List page of Kubernetes clusters, click Dashboard at the right of a cluster to enter the Kubernetes dashboard page.
3. Select kube-system under Namespace in the left-side navigation pane. Click Config Maps in the left-side navigation pane. Click the icon at the right of nginx-configuration and then select View/edit YAML. Edit the config map to add the configuration item `enable - vts - status : " true "`.

The contents of the saved Ingress ConfigMap are as follows:

```
" kind ": " ConfigMap ",
" apiVersion ": " v1 ",
" metadata ": {
  " name ": " nginx - configurat ion ",
```

```

" namespace ": " kube - system ",
" selfLink ": "/ api / v1 / namespaces / kube - system /
configmaps / nginx - configurat ion ",
" creationTimestamp ": " 2018 - 03 - 20T07 : 10 : 18Z ",
" labels ": {
  " app ": " ingress - nginx "

  " annotation s ": {
    " kubectl . kubernetes . io / last - applied - configurat
ion ": "{\n apiVersion \":\n v1 \",\n data \":{\n proxy - body -
size \":\n 20m \",\n kind \":\n ConfigMap \",\n metadata \":{\n
annotation s \":{\n labels \":{\n app \":\n ingress - nginx \",
\n name \":\n nginx - configurat ion \",\n namespace \":\n kube -
system \"}\n}"
" data ": {
  " proxy - body - size ": " 20m ",
  " enable - vts - status ": " true "

```

#### 4. Locally access the Ingress Nginx monitoring console.



##### Note:

By default, the VTS port is not opened for security considerations. Here use the port-forward method to access the console.

```

root @ master # kubectl port - forward nginx - ingress -
controller - 79877595c8 - 78gq8 - n kube - system 18080
Forwarding from 127 . 0 . 0 . 1 : 18080 -> 18080
Handling connection for 18080

```

#### 5. Use `http://localhost:18080/nginx_status` to access the VTS monitoring console.

## Nginx Vhost Traffic Status

### Server main

Host	Version	Uptime	Connections				Requests				Shared memory			
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	usedSize	usedNode
nginx-ingress-controller-79877595c8-78gq8	1.13.7	32m 41s	7	0	1	6	93566	93566	1428	1	vhost_traffic_status	10.0 MiB	2.4 KiB	1

### Server zones

Zone	Requests			Responses					Traffic					Cache								
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Scarc	Total
-	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0
*	660	1	0ms	0	660	0	0	0	660	1.7 MiB	145.4 KiB	1.1 KiB	503 B	0	0	0	0	0	0	0	0	0

### Upstreams

#### upstream-default-backend

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Traffic		
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd
172.16.3.6:8080	up	0ms	1	0	0	0	0	0ms	0	0	0	0	0	0	0 B	0 B

update interval: 1 sec

[JSON](#) | [GITHUB](#)



## 1.11.4 Support for Ingress

In Kubernetes clusters, Ingress is a collection of rules that authorize inbound connection to the cluster services and provides you with Layer-7 Server Load Balancer capabilities. You can provide the Ingress configuration with externally accessible URL, Server Load Balancer, SSL, and name-based virtual host.

### Prerequisites

To test the complex routing service, create an Nginx application in this example. You must create the Nginx deployment and multiple services in advance to observe the routing effect. Replace with your own service in the actual test. In the actual test enter your own service.

```
root @ master # kubectl run nginx -- image = registry . cn -
hangzhou . aliyuncs . com / acs / netdia : latest

root @ master # kubectl expose deploy nginx -- name = http -
svc -- port = 80 -- target - port = 80
root @ master # kubectl expose deploy nginx -- name = http -
svc1 -- port = 80 -- target - port = 80
root @ master # kubectl expose deploy nginx -- name = http -
svc2 -- port = 80 -- target - port = 80
root @ master # kubectl expose deploy nginx -- name = http -
svc3 -- port = 80 -- target - port = 80
```

### Simple routing service

Create a simple Ingress service by using the following commands. All the accesses to the `/ svc` path are routed to the Nginx service. `nginx . ingress . kubernetes . io / rewrite - target : /` redirects the path `/ svc` to the path `/` that can be recognized by backend services.

```
root @ master # cat << EOF | kubectl create - f -
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : simple
  annotation s :
    nginx . ingress . kubernetes . io / rewrite - target : /
spec :
  rules :
  - http :
    paths :
    - path : / svc
      backend :
        serviceNam e : http - svc
        servicePor t : 80
EOF
root @ master # kubectl get ing
NAME HOSTS ADDRESS PORTS AGE
```

```
simple      *      101 . 37 . 192 . 211      80
lls
```

Now visit `http://101.37.192.211/svc` to access the Nginx service.

### Simple fanout routing based on domain names

If you have multiple domain names providing different external services, you can generate the following configuration to implement a simple fanout effect based on domain names:

```
root @ master # cat << EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
  - host: foo.example.com
    http:
      paths:
      - path: /film
        backend:
          serviceName: http-svc3
          servicePort: 80
EOF
root @ master # kubectl get ing
NAME          HOSTS          ADDRESS          PORTS          AGE
simple-fanout  *              101.37.192.211  80
```

Then, you can access the `http-svc1` service by using `http://foo.bar.com/foo`, access the `http-svc2` service by using `http://foo.bar.com/bar`, and access the `http-svc3` service by using `http://foo.example.com/film`.



#### Note:

- In a production environment, point the domain name to the preceding returned address `101.37.192.211`.

- In a testing environment, you can modify the `hosts` file to add a domain name mapping rule.

```
101 . 37 . 192 . 211    foo . bar . com
101 . 37 . 192 . 211    foo . example . com
```

### Default domain name of simple routing

It does not matter if you do not have the domain name address. Container Service binds a default domain name for Ingress service. You can use this default domain name to access the services. The domain name is in the format of `*.[ cluster - id ].[ region - id ].alicontainer.com`. You can obtain the address on the cluster Basic Information page in the console.

Use the following configuration to expose two services with the default domain name.

```
root @ master # cat << EOF | kubectl create -f -
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : shared - dns
spec :
  rules :
    - host : foo .[ cluster - id ].[ region - id ]. alicontainer .
      com ## Replace with the default service access domain
        name of your cluster .
      http :
        paths :
          - path : /
            backend :
              serviceName : http - svc1
              servicePort : 80
    - host : bar .[ cluster - id ].[ region - id ]. alicontainer .
      com ## Replace with the default service access domain
        name of your cluster .
      http :
        paths :
          - path : /
            backend :
              serviceName : http - svc2
              servicePort : 80
EOF
root @ master # kubectl get ing
NAME          HOSTS          ADDRESS          PORTS          AGE
shared - dns   foo .[ cluster - id ].[ region - id ]. alicontainer .
er . com , bar .[ cluster - id ].[ region - id ]. alicontainer .
com           47 . 95 . 160 . 171      80              40m
```

Then, you can access the `http - svc1` service by using `http :// foo .[ cluster - id ].[ region - id ]. alicontainer . com` /and access the `http - svc2` service by using `http :// bar .[ cluster - id ].[ region - id ]. alicontainer . com` .

## Configure a safe routing service

Management of multiple certificates is supported to provide security protection for your services.

### 1. Prepare your service certificate.

If no certificate is available, generate a test certificate in the following method:



**Note:**

The domain name must be consistent with your Ingress configuration.

```
root @ master # openssl req -x509 -nodes -days 365
- newkey rsa : 2048 - keyout tls . key - out tls . crt -
subj "/ CN = foo . bar . com / O = foo . bar . com "
```

The above command generates a certificate file `tls . crt` and a private key file `tls . key`.

Create a Kubernetes secret named `foo . bar` using the certificate and private key. The secret must be referenced when you create the Ingress.

```
root @ master # kubectl create secret tls foo . bar --
key tls . key -- cert tls . crt
```

### 2. Create a safe Ingress service.

```
root @ master # cat << EOF | kubectl create -f -
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : tls - fanout
spec :
  tls :
  - hosts :
    - foo . bar . com
    secretName : foo . bar
  rules :
  - host : foo . bar . com
    http :
      paths :
      - path : / foo
        backend :
          serviceName : http - svc1
          servicePort : 80
      - path : / bar
        backend :
          serviceName : http - svc2
          servicePort : 80
EOF
root @ master # kubectl get ing
NAME                                HOSTS                                ADDRESS                                PORTS
AGE
```

```
tls - fanout      *      101 . 37 . 192 . 211      80
  11s
```

3. Follow the notes in Simple fanout routing based on domain names to configure the `hosts` file or set the domain name to access the TLS service.

You can access the `http - svc1` service by using `http :// foo . bar . com / foo` and access the `http - svc2` service by using `http :// foo . bar . com / bar`.

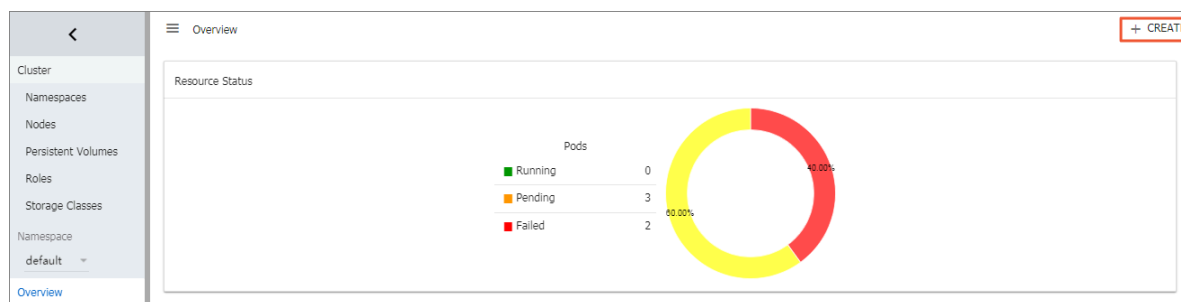
You can also access the HTTPS service by using HTTP. By default, Ingress redirects HTTP access configured with HTTPS to the HTTPS address. Therefore, access to `http :// foo . bar . com / foo` will be automatically redirected to `https :// foo . bar . com / foo`.

## Deploy Ingress in Kubernetes dashboard

1. Save the following yml code to the `nginx - ingress . yml` file.

```
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : simple
spec :
  rules :
  - http :
    paths :
    - path : / svc
      backend :
        serviceName : http - svc
        servicePort : 80
```

2. Log on to the [Container Service console](#). Under Kubernetes, click Clusters in the left-side navigation pane. Click Dashboard at the right of the cluster to enter the Kubernetes dashboard.
3. Click CREATE in the upper-right corner to create an application.



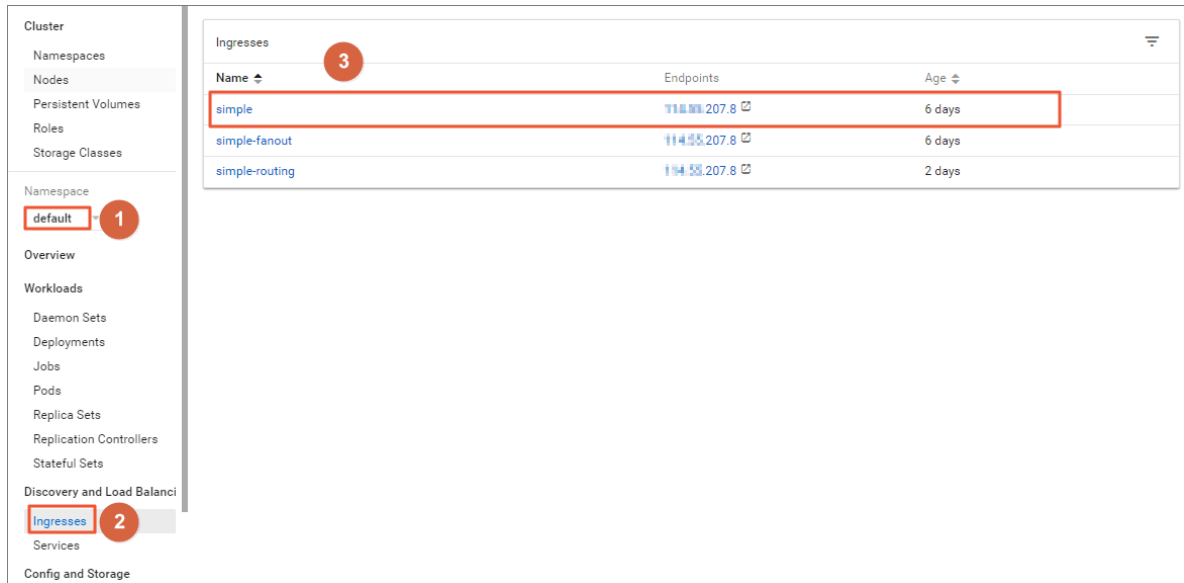
4. Click the CREATE FROM FILE tab. Select the `nginx - ingress . yml` file you saved.

## 5. Click UPLOAD.

Then an Ingress Layer-7 proxy route will be created to the `http - svc` service.

## 6. Click default under Namespace in the left-side navigation pane. Click Ingresses in the left-side navigation pane.

You can view the created Ingress resource and its access address `http :// 118 . 178 . 174 . 161 / svc .`



## 7. Enter the address in the browser to access the created `http - svc` service.

# 1.12 Storage

## 1.12.1 Overview

Container Service supports automatically binding Kubernetes pods to Alibaba Cloud cloud disks, NAS, and Object Storage Service (OSS).

Currently, static storage volumes and dynamic storage volumes are supported.

See the following table for how each type of data volumes supports the static data volumes and dynamic data volumes.

Alibaba Cloud storage	Static data volume	Dynamic data volume
Alibaba Cloud cloud disk	<p>You can use the cloud disk static storage volumes by:</p> <ul style="list-style-type: none"> <li>Using the volume method.</li> <li>Using PV/PVC.</li> </ul>	Supported.

Alibaba Cloud storage	Static data volume	Dynamic data volume
Alibaba Cloud NAS	You can use the NAS static storage volumes by: <ul style="list-style-type: none"> <li>· Using flexvolume plug-in.</li> <li>- Using the volume method.</li> <li>- Using PV/PVC.</li> <li>· Using NFS drive of Kubernetes.</li> </ul>	Supported.
Alibaba Cloud OSS	You can use the OSS static storage volumes by: <ul style="list-style-type: none"> <li>· Using the volume method.</li> <li>· Using PV/PVC.</li> </ul>	Not supported.

## 1.12.2 Use Alibaba Cloud cloud disks

You can use the Alibaba Cloud cloud disk storage volumes in Alibaba Cloud Container Service Kubernetes clusters.

Currently, Alibaba Cloud cloud disk provides the following two Kubernetes mount methods:

- [Static storage volumes](#)

You can use the cloud disk static storage volumes by:

- [Using the volume method](#)
- [Using PV/PVC](#)

- [Dynamic storage volumes](#)



**Note:**

The following requirements are imposed on the created cloud disk capacity:

- Basic cloud disk: Minimum 5Gi
- Ultra cloud disk: Minimum 20Gi
- SSD cloud disk: Minimum 20Gi

## Static storage volumes

You can use Alibaba Cloud cloud disk storage volumes by using the volume method or PV/PVC.

### Prerequisites

Before using cloud disk data volumes, you must create cloud disks in the Elastic Compute Service (ECS) console. For how to create cloud disks, see [#unique\\_70](#).

### Instructions

- The cloud disk is not a shared storage and can only be mounted by one pod at the same time.
- Apply for a cloud disk and obtain the disk ID before using cloud disk storage volumes. See [#unique\\_70](#).
- `volumeId`: The disk ID of the mounted cloud disk, which must be the same as `volumeName` and PV Name.
- Only the cluster node that is in the same zone as the cloud disk can mount the cloud disk.

### Use volume method

Use the `disk - deploy . yaml` file to create the pod.

```
apiVersion : extensions / v1beta1
kind : Deployment
metadata :
  name : nginx - disk - deploy
spec :
  replicas : 1
  template :
    metadata :
      labels :
        app : nginx
    spec :
      containers :
        - name : nginx - flexvolume - disk
          image : nginx
          volumeMounts :
            - name : " d - bp1j17ifxf asvts3tf40 "
              mountPath : "/" data "
          volumes :
            - name : " d - bp1j17ifxf asvts3tf40 "
              flexVolume :
                driver : " alicloud / disk "
                fsType : " ext4 "
                options :
                  volumeId : " d - bp1j17ifxf asvts3tf40 "
```

### Use PV/PVC



## Step 1. Create a cloud disk type PV

You can create the cloud disk type PV in the Container Service console or by using the yaml file.

### Create PV by using yaml file

Use the `disk - pv . yaml` file to create the PV.



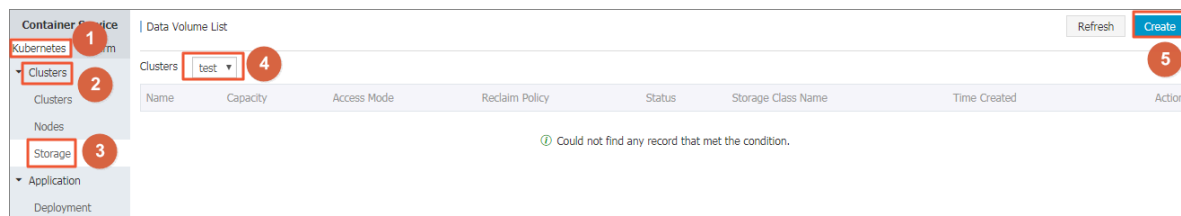
#### Note:

The PV name must be the same as the Alibaba Cloud cloud disk ID.

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : d - bp1j17ifxf asvts3tf40
  labels :
    failure-domain.beta.kubernetes.io/zone : cn-hangzhou-b
    failure-domain.beta.kubernetes.io/region : cn-hangzhou
spec :
  capacity :
    storage : 20Gi
  storageClassName : disk
  accessModes :
    - ReadWriteOnce
  flexVolume :
    driver : "alicloud/disk"
    fsType : "ext4"
    options :
      volumeId : "d - bp1j17ifxf asvts3tf40"
```

### Create cloud disk data volumes in Container Service console

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Clusters** > **Storage** in the left-side navigation pane.
3. Select the cluster from the Clusters drop-down list and then click **Create** in the upper-right corner.



4. The Create Data Volume dialog box appears. Configure the data volume parameters.

- **Type:** cloud disk in the example.
- **Name:** The name of the created data volume. The data volume name must be the same as the Cloud Disk ID.
- **Access Mode:** ReadWriteOnce by default.
- **Cloud Disk ID:** Select the cloud disk to be mounted and is in the same region and zone as the cluster.
- **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 ext4 is selected by default.
- **Tag:** Click Add Tag to add tags for this data volume.

5. After the preceding settings, click Create.

## Step 2. Create PVC

Use the `disk - pvc . yaml` file to create the PVC.

```
kind : Persistent VolumeClaim
apiVersion : v1
metadata :
  name : pvc - disk
spec :
  accessModes :
    - ReadWriteOnce
  storageClassName : disk
```

```
resources :
  requests :
    storage : 20Gi
```

### Step 3. Create a pod

Use the `disk - pod . yaml` file to create the pod.

```
apiVersion : v1
kind : Pod
metadata :
  name : " flexvolume - alicloud - example "
spec :
  containers :
    - name : " nginx "
      image : " nginx "
      volumeMounts :
        - name : pvc - disk
          mountPath : "/" data "
  volumes :
    - name : pvc - disk
      persistent VolumeClaim :
        claimName : pvc - disk
```

## Dynamic storage volumes

Dynamic storage volumes require you to manually create a Storage Class and specify the target type of cloud disk in the PVC by storage Class Name.

### Create a StorageClass

```
kind : StorageClass
apiVersion : storage . k8s . io / v1beta1
metadata :
  name : alicloud - disk - common - hangzhou - b
provisioner : alicloud / disk
parameters :
  type : cloud_ssd
  regionid : cn - hangzhou
  zoneid : cn - hangzhou - b
```

#### Parameters:

- **provisioner:** configured as Alibaba Cloud/disk, the identifier is created using the Alibaba Cloud Provisioner plug-in.
- **type:** identifies the cloud disk type, supports cloud, cloud\_efficiency, cloud\_ssd, and available types. To improve efficiency, tries to create SSD, and common cloud disk until it is created successfully.
- **regionid:** the region where cloud disk is to be created.
- **zoneid:** the zone where cloud disk is to be created.

## Create a service

```
kind : Persistent VolumeClaim
apiVersion : v1
metadata :
  name : disk - common
spec :
  accessModes :
    - ReadWriteOnce
  storageClassName : alicloud - disk - common - hangzhou - b
  resources :
    requests :
      storage : 20Gi
---
kind : Pod
apiVersion : v1
metadata :
  name : disk - pod - common
spec :
  containers :
    - name : disk - pod
      image : nginx
      volumeMounts :
        - name : disk - pvc
          mountPath : "/mnt"
      restartPolicy : "Never"
  volumes :
    - name : disk - pvc
      persistentVolumeClaim :
        claimName : disk - common
```

## Default options

By default, the cluster provides the following StorageClasses, which can be used in a single AZ cluster.

- **alicloud-disk-common**: basic cloud disk.
- **alicloud-disk-efficiency**: high-efficiency cloud disk.
- **alicloud-disk-ssd**: SSD disk.
- **alicloud-disk-available**: provides highly available options, first attempts to create a high-efficiency cloud disk. If the corresponding AZ's efficient cloud disk resources are sold out, tries to create an SSD disk. If the SSD is sold out, tries to create a common cloud disk.

## Creating a multi-instance StatefulSet using cloud disk

Use volume Claim Templates that dynamically creates multiple PVCs and PVs and binds them.

```
apiVersion : v1
kind : Service
metadata :
  name : nginx
  labels :
```

```

    app : nginx
spec :
  ports :
    - port : 80
      name : web
    clusterIP : None
  selector :
    app : nginx
---
apiVersion : apps / v1beta2
kind : StatefulSet
metadata :
  name : web
spec :
  selector :
    matchLabels :
      app : nginx
  serviceName : "nginx"
  replicas : 2
  template :
    metadata :
      labels :
        app : nginx
    spec :
      containers :
        - name : nginx
          image : nginx
          ports :
            - containerPort : 80
              name : web
          volumeMounts :
            - name : disk - common
              mountPath : / data
  volumeClaimTemplates :
    - metadata :
        name : disk - common
      spec :
        accessModes : [ "ReadWriteOnce" ]
        storageClassName : "alicloud - disk - common"
        resources :
          requests :
            storage : 10Gi

```

### 1.12.3 Use Alibaba Cloud NAS

You can use the Alibaba Cloud NAS data volumes in Container Service Kubernetes clusters.

Currently, Alibaba Cloud NAS provides the following two Kubernetes mount methods:

- **Static storage volumes**

You can use the static storage volumes by:

- Using the flexvolume plug-in.
  - Using the volume method.
  - Using PV/PVC.
- Using NFS drive of Kubernetes.

- **Dynamic storage volumes**

#### Prerequisite

Before using NAS data volumes, you must create a file system in the NAS console and add the mount point of a Kubernetes cluster in the file system. The created NAS file system and your cluster must be in the same Virtual Private Cloud (VPC).

#### Static storage volumes

You can use Alibaba Cloud NAS file storage service by using the flexvolume plug-in provided by Alibaba Cloud or the NFS drive of Kubernetes.

#### Use flexvolume plug-in

Use the flexvolume plug-in and then you can use the Alibaba Cloud NAS data volumes by using the volume method or using PV/PVC.



#### Note:

- NAS is a shared storage and can provide shared storage service for multiple pods at the same time.
- **server:** The mount point of the NAS data disk.
- **path:** The mount directory for connecting to the NAS data volumes. You can mount NAS data volumes to a NAS sub-directory. The system automatically creates the sub-directory if the sub-directory does not exist and mounts the NAS data volumes to the created sub-directory.
- **vers:** Defines the version number of NFS mount protocol. 4.0 is supported.
- **mode:** Defines the access permission of the mount directory. The mount permission cannot be configured if you mount the NAS data volumes to the NAS root directory. If the NAS disk contains a huge amount of data, configuring the mode leads to the slow mounting or even the mounting failure.

Using the volume method.

Use the `nas - deploy . yaml` file to create the pod.

```
apiVersion : v1
kind : Pod
metadata :
  name : " flexvolume - nas - example "
spec :
  containers :
    - name : " nginx "
      image : " nginx "
      volumeMounts :
        - name : " nas1 "
          mountPath : "/" data "
  volumes :
    - name : " nas1 "
      flexVolume :
        driver : " alicloud / nas "
        options :
          server : " 0cd8b4a576 - grs79 . cn - hangzhou . nas .
aliyuncs . com "
          path : "/" k8s "
          vers : " 4 . 0 "
```

Use PV/PVC

Step 1 Create PV

You can create NAS data volumes in the Container Service console or by using the YAML file.

- Create PV by using YAML file

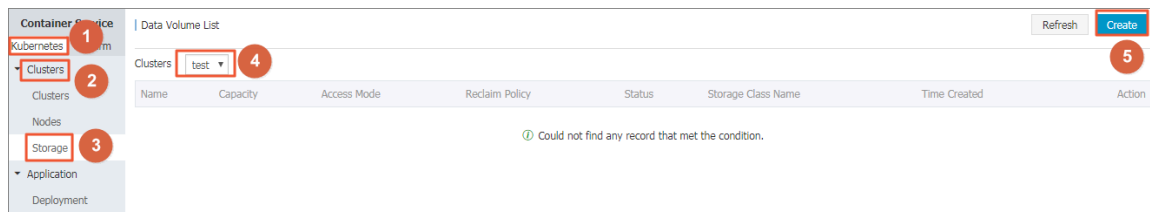
Use the `nas - pv . yaml` file to create the PV.

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : pv - nas
spec :
  capacity :
    storage : 5Gi
  storageClassName : nas
  accessModes :
    - ReadWriteMany
  flexVolume :
    driver : " alicloud / nas "
    options :
      server : " 0cd8b4a576 - uih75 . cn - hangzhou . nas .
aliyuncs . com "
      path : "/" k8s "
```

```
vers : " 4 . 0 "
```

- Create NAS data volumes in Container Service console

1. Log on to the [Container Service console](#).
2. Under Kubernetes, click **Cluster** > **Storage** in the left-side navigation pane.
3. Select the cluster from the Clusters drop-down list and then click **Create** in the upper-right corner.

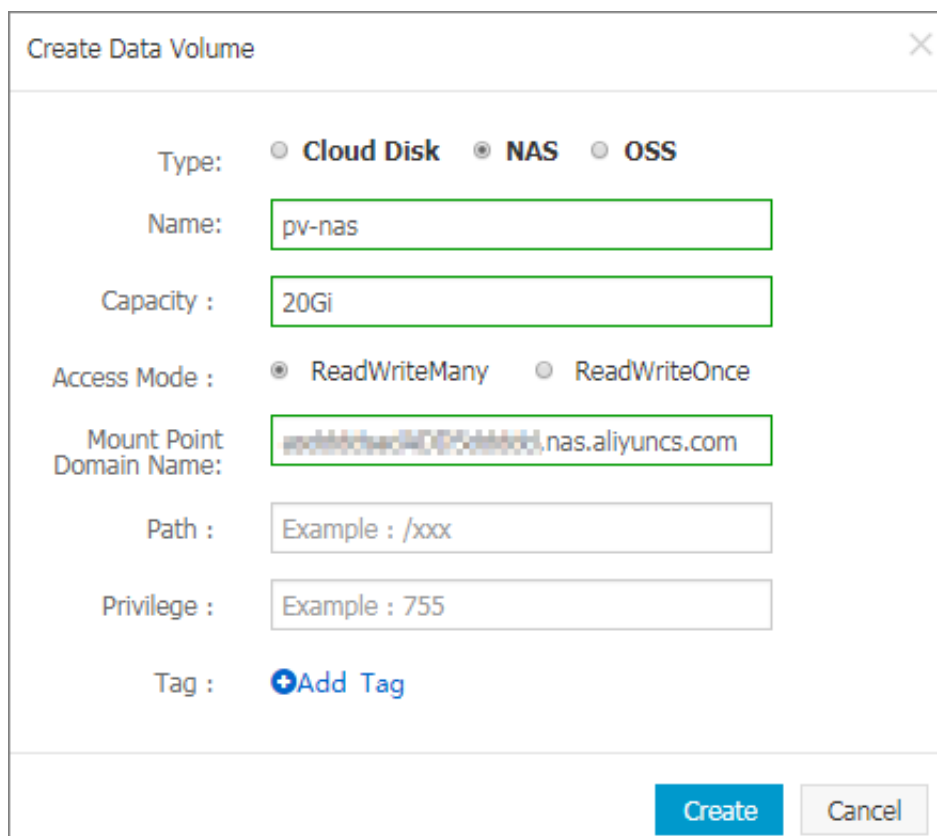


4. The Create Data Volume dialog box appears. Configure the data volume parameters.

- **Type:** Select NAS in this example.
- **Name:** Enter the name of the data volume you are about to create. The data volume name must be unique in the cluster. In this example, enter pv-nas.
- **Capacity:** Enter the capacity of the data volume to be created. Make sure the capacity cannot exceed the disk capacity.
- **Access Mode:** ReadWriteMany is selected by default.
- **Mount Point Domain Name:** Enter the mount address of the mount point in the NAS file system for the cluster.
- **Path:** The sub-directory under the NAS path, which starts with a forward slash ( / ). The data volume is mounted to the specified sub-directory after being created.
  - If this sub-directory does not exist in the NAS root directory, the data volume is mounted after the sub-directory is created by default.
  - If this field is left empty, the data volume is mounted to the NAS root directory by default.
- **Privilege:** Configure the access permission of the mount directory, such as 755, 644, and 777.
  - You can only configure the privilege when the data volume is mounted to the NAS sub-directory, that is, you cannot configure the privilege if the data volume is mounted to the NAS root directory.
  - If this field is left empty, use the permissions of the NAS files by default.



- **Tag:** Click Add Tag to add tags for this data volume.



The image shows a 'Create Data Volume' dialog box with the following fields and options:

- Type:** Radio buttons for Cloud Disk, **NAS** (selected), and OSS.
- Name:** Text input field containing 'pv-nas'.
- Capacity :** Text input field containing '20Gi'.
- Access Mode :** Radio buttons for **ReadWriteMany** (selected) and ReadWriteOnce.
- Mount Point Domain Name:** Text input field containing 'ecs-4d0f56b0c1.nas.aliyuncs.com'.
- Path :** Text input field containing 'Example : /xxx'.
- Privilege :** Text input field containing 'Example : 755'.
- Tag :** A blue button labeled '+Add Tag'.

At the bottom right, there are two buttons: 'Create' (in blue) and 'Cancel' (in grey).

5. Click Create after the configurations.

## Step 2 Create PVC

Use the `nas - pvc . yaml` file to create the PVC.

```
kind : Persistent VolumeClaim
apiVersion : v1
metadata :
  name : pvc - nas
spec :
  accessModes :
    - ReadWriteMany
  storageClassName : nas
  resources :
    requests :
      storage : 5Gi
```

## Step 3 Create pod

Use the `nas - pod . yaml` file to create the pod.

```
apiVersion : v1
kind : Pod
metadata :
  name : " flexvolume - nas - example "
spec :
  containers :
    - name : " nginx "
```

```

    image : " nginx "
    volumeMounts :
      - name : pvc - nas
        mountPath : "/ data "
  volumes :
    - name : pvc - nas
      persistent VolumeClaim :
        claimName : pvc - nas

```

Using NFS drive of Kubernetes.

### Step 1 Create a NAS file system

Log on to the [NAS console](#) to create a NAS file system.



#### Note:

The created NAS file system and your cluster must be in the same region.

Assume that your mount point is `055f84ad83 - ixxxx . cn - hangzhou . nas . aliyuncs . com`.

### Step 2 Create PV

You can create NAS data volumes in the Container Service console or by using an orchestration template.

- Use an orchestration template

Use the `nas - pv . yaml` file to create the PV.

Run the following commands to create a NAS type PersistentVolume.

```

root@master # cat << EOF | kubectl apply -f -
apiVersion : v1
kind : Persistent Volume
metadata :
  name : nas
spec :
  capacity :
    storage : 8Gi
  accessModes :
    - ReadWriteMany
  persistent VolumeReclaimPolicy : Retain
  nfs :
    path : /
    server : 055f84ad83 - ixxxx . cn - hangzhou . nas . aliyuncs . com
EOF

```

- Create NAS data volumes in Container Service console

For more information, see [Create NAS data volumes in Container Service console](#) in [Use PV/PVC](#).

## Step 2 Create PVC

Create a `PersistentVolumeClaim` to request to bind this `PersistentVolume`.

```
root @ master # cat << EOF | kubectl apply -f -
apiVersion : v1
kind : Persistent VolumeClaim
metadata :
  name : nasclaim
spec :
  accessModes :
    - ReadWriteMany
  resources :
    requests :
      storage : 8Gi
EOF
```

## Step 3 Create pod

Create an application to declare to mount and use this data volume.

```
root @ master # cat << EOF | kubectl apply -f -
apiVersion : v1
kind : Pod
metadata :
  name : mypod
spec :
  containers :
    - name : myfrontend
      image : registry.aliyuncs.com/spacexnice/netdia :
latest
      volumeMounts :
        - mountPath : "/var/www/html"
          name : mypd
  volumes :
    - name : mypd
      persistentVolumeClaim :
        claimName : nasclaim
EOF
```

Then, the NAS remote file system is mounted to your pod application.

## Dynamic storage volumes

To use dynamic NAS storage volumes, you must manually install the drive plug-in and configure the NAS mount point.

### Install the plug-in

```
apiVersion : storage.k8s.io/v1
kind : StorageClass
metadata :
  name : alicloud-nas
provisioner : alicloud/nas
---
apiVersion : v1
kind : ServiceAccount
metadata :
```

```

    name : alicloud - nas - controller
    namespace : kube - system
---
kind : ClusterRole
apiVersion : rbac.authorization.k8s.io / v1beta1
metadata :
  name : run - alicloud - nas - controller
subjects :
  - kind : ServiceAccount
    name : alicloud - nas - controller
    namespace : kube - system
roleRef :
  kind : ClusterRole
  name : alicloud - disk - controller - runner
  apiGroup : rbac.authorization.k8s.io
---
kind : Deployment
apiVersion : extensions / v1beta1
metadata :
  name : alicloud - nas - controller
  namespace : kube - system
spec :
  replicas : 1
  strategy :
    type : Recreate
  template :
    metadata :
      labels :
        app : alicloud - nas - controller
    spec :
      tolerations :
        - effect : NoSchedule
          operator : Exists
          key : node-role.kubernetes.io / master
        - effect : NoSchedule
          operator : Exists
          key : node.cloudprovider.kubernetes.io / uninitialized
      nodeSelector :
        node-role.kubernetes.io / master : ""
      serviceAccount : alicloud - nas - controller
      containers :
        - name : alicloud - nas - controller
          image : registry.cn-hangzhou.aliyuncs.com / acs /
alicloud - nas - controller : v1.8.4
          volumeMounts :
            - mountPath : / persistent volumes
              name : nfs - client - root
          env :
            - name : PROVISIONER_NAME
              value : alicloud / nas
            - name : NFS_SERVER
              value : 0cd8b4a576 - mmi32.cn - hangzhou.nas.aliyuncs.com
            - name : NFS_PATH
              value : /
          volumes :
            - name : nfs - client - root
              nfs :
                server : 0cd8b4a576 - mmi32.cn - hangzhou.nas.aliyuncs.com

```

```
path : /
```

### Use dynamic storage volumes

```
apiVersion : apps / v1beta1
kind : StatefulSet
metadata :
  name : web
spec :
  serviceName : "nginx"
  replicas : 2
  volumeClaimTemplates :
  - metadata :
    name : html
    spec :
      accessModes :
        - ReadWriteOnce
      storageClassName : alicloud - nas
      resources :
        requests :
          storage : 2Gi
  template :
    metadata :
      labels :
        app : nginx
    spec :
      containers :
      - name : nginx
        image : nginx : alpine
        volumeMounts :
        - mountPath : "/usr / share / nginx / html /"
          name : html
```

## 1.12.4 Use Alibaba Cloud OSS

You can use the Alibaba Cloud Object Storage Service (OSS) data volumes in Alibaba Cloud Container Service Kubernetes clusters.

Currently, OSS static storage volumes are supported, while OSS dynamic storage volumes are not supported. You can use the OSS static storage volumes by:

- Using the volume method.
- Using PV/PVC.

### Prerequisites

You must create a bucket in the OSS console before using the OSS static storage volumes.

### Instructions

- OSS is a shared storage and can provide shared storage service for multiple pods at the same time.

- **bucket:** Currently, Container Service only supports mounting buckets and cannot mount the sub-directories or files under the bucket.
- **url:** The OSS endpoint, which is the access domain name for mounting OSS.
- **akId:** Your AccessKey ID.
- **akSecret:** Your AccessKey Secret.
- **otherOpts:** Customized parameter input in the format of `- o *** - o ***` is supported when mounting OSS.

#### Note

If your Kubernetes cluster is created before Feb 6th, 2018, [#unique\\_73](#) before using the data volumes. To use OSS data volumes, you must create the secret and enter the AccessKey information when deploying the flexvolume service.

#### Use OSS static storage volumes

##### Use volume method

Use the `oss - deploy . yaml` file to create the pod.

```
apiVersion : extensions / v1beta1
kind : Deployment
metadata :
  name : nginx - oss - deploy
spec :
  replicas : 1
  template :
    metadata :
      labels :
        app : nginx
    spec :
      containers :
        - name : nginx - flexvolume - oss
          image : nginx
          volumeMounts :
            - name : " oss1 "
              mountPath : "/" data "
          volumes :
            - name : " oss1 "
              flexVolume :
                driver : " alicloud / oss "
                options :
                  bucket : " docker "
                  url : " oss - cn - hangzhou . aliyuncs . com "
                  akId : ***
                  akSecret : ***
                  otherOpts : "- o max_stat_c ache_size = 0 - o
allowOther "

```

Use PV/PVC (currently, dynamic pv is not supported)

#### Step 1 Create PV

You can create the PV in the Container Service console or by using the YAML file.

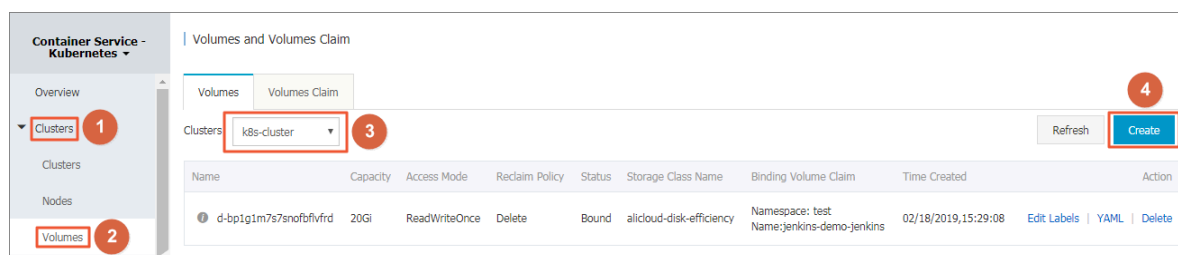
### Create PV by using YAML file

Use the `oss - pv . yaml` file to create the PV.

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : pv - oss
spec :
  capacity :
    storage : 5Gi
  accessModes :
    - ReadWriteMany
  storageClassName : oss
  flexVolume :
    driver : "alicloud / oss"
    options :
      bucket : "docker"
      url : "oss - cn - hangzhou . aliyuncs . com"
      akId : ***
      akSecret : ***
      otherOpts : "-o max_stat_cache_size = 0 -o allow_othe
r "
```

### Create OSS data volumes in Container Service console

1. Log on to the [Container Service console](#).
2. In the left-side navigation pane under Kubernetes, choose Clusters > Storage.
3. Select the cluster from the Clusters drop-down list and then click Create in the upper-right corner.



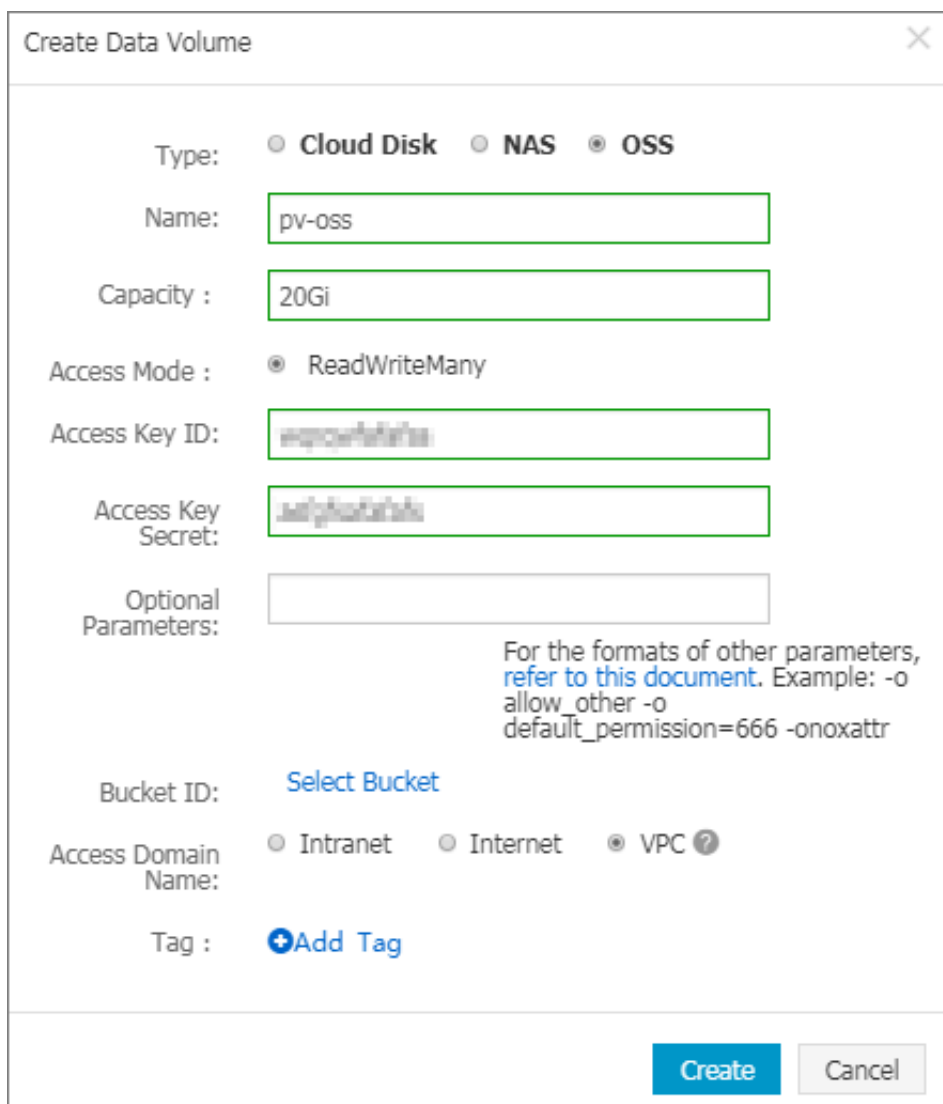
4. The Create Data Volume dialog box appears. Configure the data volume parameters.

- **Type:** Select OSS in this example.
- **Name:** Enter the name of the data volume you are about to create. The data volume name must be unique in the cluster. In this example, enter pv-oss.
- **Capacity:** Enter the capacity of the data volume to be created.
- **Access Mode:** ReadWriteMany by default.
- **Access Key ID/Access Key Secret:** The AccessKey required to access OSS.
- **Bucket ID:** Select the OSS bucket name you want to use. Click Select Bucket. Select the bucket in the displayed dialog box and click Select.
- **Access Domain Name:** If the bucket and Elastic Compute Service (ECS) instance are in different regions, select Internet. If the bucket and ECS instance are in the same region, select Intranet or VPC according to the cluster network type. Select



VPC if the network type is Virtual Private Cloud (VPC) or select Intranet if the network type is classic network.

- Tag: Click Add Tag to add tags for this data volume.



The 'Create Data Volume' dialog box contains the following fields and options:

- Type:** Radio buttons for ☐ Cloud Disk, ☐ NAS, and ☒ OSS.
- Name:** Text input field containing 'pv-oss'.
- Capacity :** Text input field containing '20Gi'.
- Access Mode :** Radio button for ☒ ReadWriteMany.
- Access Key ID:** Text input field containing a placeholder ID.
- Access Key Secret:** Text input field containing a placeholder secret.
- Optional Parameters:** Empty text input field.
- Bucket ID:** Link labeled 'Select Bucket'.
- Access Domain Name:** Radio buttons for ☐ Intranet, ☐ Internet, and ☒ VPC ?.
- Tag :** Link labeled '+Add Tag'.

Below the 'Optional Parameters' field, there is a note: "For the formats of other parameters, refer to this document. Example: -o allow\_other -o default\_permission=666 -onoxattr".

At the bottom right, there are 'Create' and 'Cancel' buttons.

5. Click Create after completing the configurations.

## Step 2 Create PVC

Use the `oss - pvc . yaml` file to create the PVC.

```
kind : Persistent VolumeClaim
apiVersion : v1
metadata :
  name : pvc - oss
spec :
  storageClassName : oss
  accessModes :
    - ReadWriteMany
resources :
  requests :
```

```
storage : 5Gi
```

### Step 3 Create pod

Use the `oss - pod . yaml` file to create the pod.

```
apiVersion : v1
kind : Pod
metadata :
  name : " flexvolume - oss - example "
spec :
  containers :
    - name : " nginx "
      image : " nginx "
      volumeMounts :
        - name : pvc - oss
          mountPath : "/" data "
  volumes :
    - name : pvc - oss
      persistent VolumeClaim :
        claimName : pvc - oss
```

Use OSS dynamic storage volumes

Currently not supported.

## 1.13 Storage claim management

### 1.13.1 Create a persistent storage volume claim

You can create a persistent storage volume claim (PVC) by using the Container Service console.

#### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- If you have already created a storage volume, use a cloud disk to create a cloud storage volume. For more information, see [#unique\\_29](#).

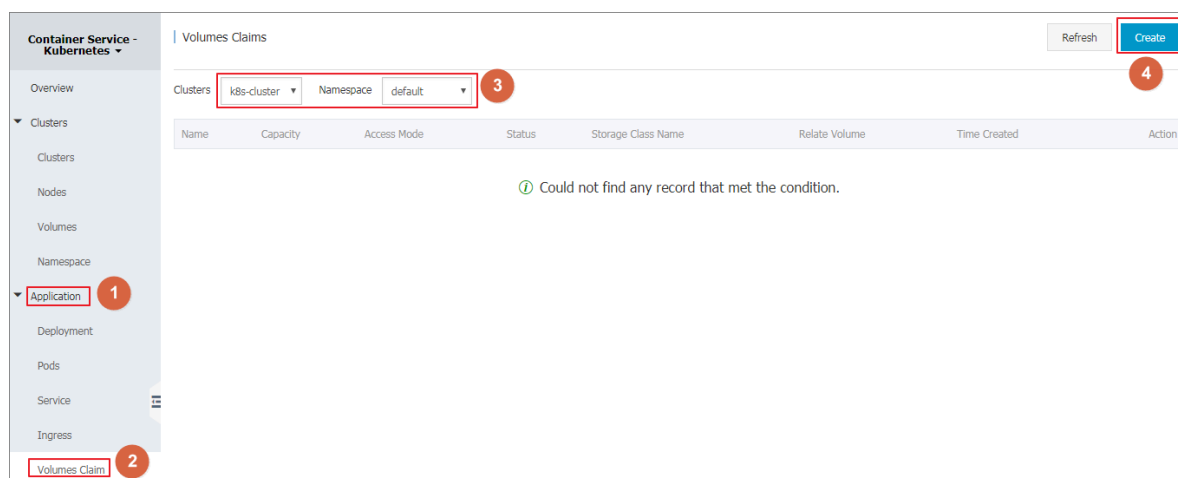
By default, the storage claim is bound to the storage volume depending on the label `alicloud - pvname`. When the data volume is created by using the Container Service console, the storage volume is labeled by default. If the storage volume label does not exist, you must add a label before you select to bound this storage volume.

#### Context

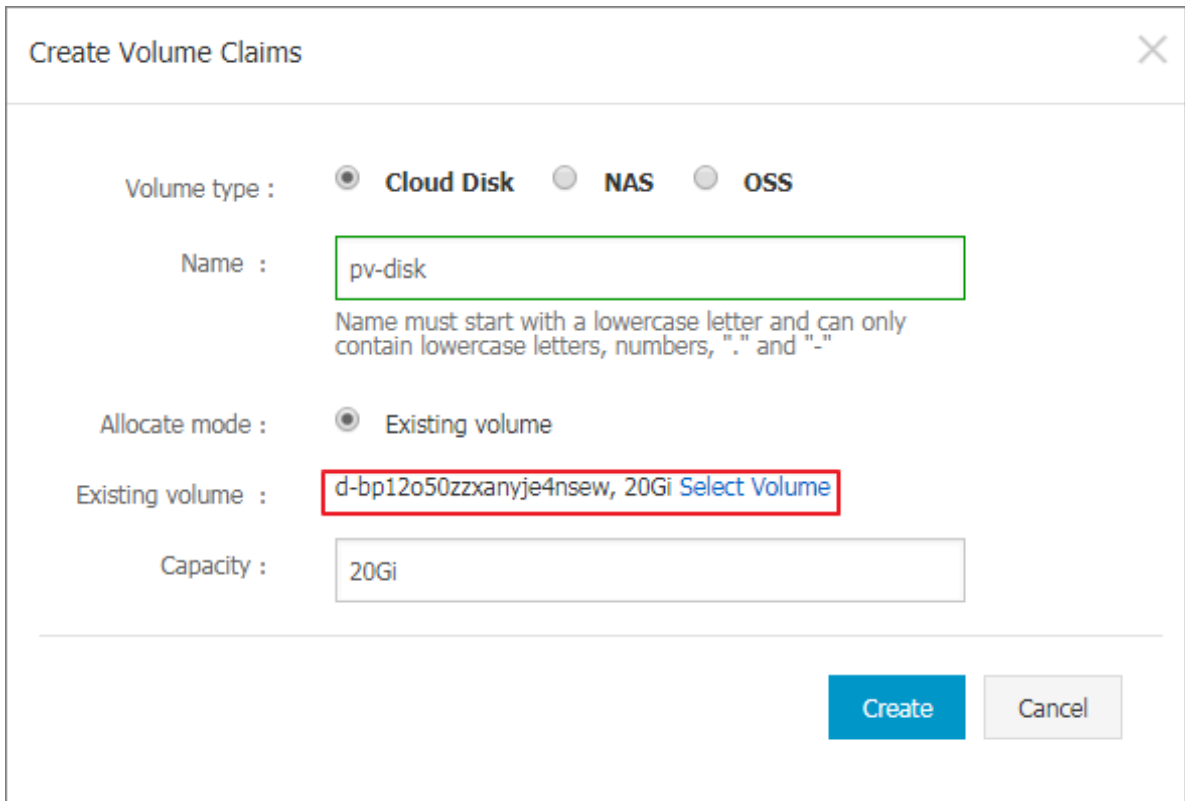
#### Procedure

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

2. Under Kubernetes, click Application > Volumes Claim in the left-side navigation pane to enter the Volumes Claims list page.
3. Select the target cluster and namespace, and click Create in the upper-right corner.



4. Complete the configurations in the Create Volume Claim dialog box, and click Create.



The image shows a 'Create Volume Claims' dialog box with the following fields and options:

- Volume type :** Three radio buttons are present: **Cloud Disk** (selected), **NAS**, and **OSS**.
- Name :** A text input field containing 'pv-disk'. Below the field, a note states: 'Name must start with a lowercase letter and can only contain lowercase letters, numbers, "." and "-"'. The field is highlighted with a green border.
- Allocate mode :** Two radio buttons are present: **Existing volume** (selected) and an unlabeled option.
- Existing volume :** A dropdown menu showing 'd-bp12o50zzxanyje4nsew, 20Gi' with a 'Select Volume' link. The dropdown is highlighted with a red border.
- Capacity :** A text input field containing '20Gi'.

At the bottom right, there are two buttons: **Create** (blue) and **Cancel** (gray).

- Volume claim type: Consistent with storage volume, including cloud, NAS, and OSS types.
- Name: Enter the storage volume claim name.
- Distribution mode: Currently, only existing storage volumes are supported.
- Existing storage volume: Select to bound the storage volume of this type.
- Total: Claim usage, cannot be greater than the total amount of storage volumes.

**Note:**

If a storage volume already exists in your cluster and is not used, but cannot be found in Select Existing Storage Volume, maybe the `alicloud - pvname` label is not defined.

If you cannot find an available storage volume, you can click **Clusters > Volumes** in the left-side navigation pane. Find the target storage volume, click **Label Management** on the right. Add the corresponding label `alicloud - pvname`, the

value is the name of the storage volume. The cloud storage volume defaults to the cloud disk ID as the name of the storage volume.

Name	Value
alicloud-pvname	d-bp1-7550t00c7emx3iv0e
failure-domain.beta.kubernetes.io/zone	cn-hangzhou-g
failure-domain.beta.kubernetes.io/region	cn-hangzhou

5. Return to the Volumes Claims list, you can see that the newly created storage claim appears in the list.

### 1.13.2 Using persistent storage volume claim

On the Container Service console, use an image or a template to deploy an application, so that you can use a persistent storage volume claim. In this example, an image is used to create an application. If you want to use a persistent storage volume claim with the template, see [#unique\\_29](#).

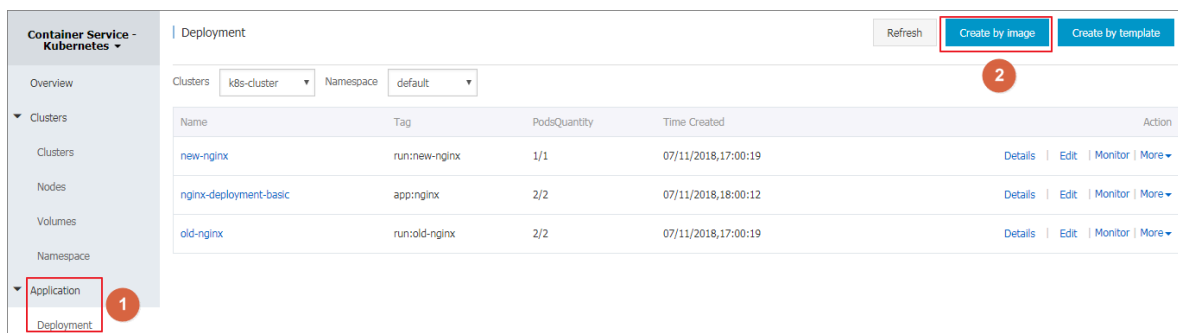
#### Prerequisites

- You have created a Kubernetes cluster. For more information, see [#unique\\_13](#).
- If you have already created a storage volume claim, use the cloud disk to create a cloud disk storage volume claim PVC disk. For more information, see [#unique\\_77](#).

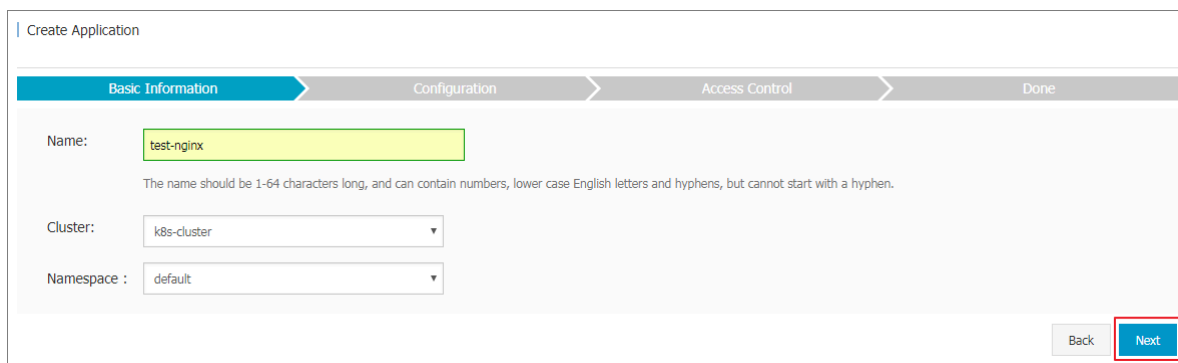
#### Procedure

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

2. Under Kubernetes, click Application > Deployment in the left-side navigation pane. Enter the Deployment List page and click Create by image in the upper-right corner.



3. On the Basic Information page, configure the application name, deploy the cluster, and the namespace. Then click Next.



4. On the Application Configuration page, select Image. Then configure the cloud storage type of data volume, cloud disk, NAS, and OSS types are supported. In this example, use the cloud storage volume claim and click Next.

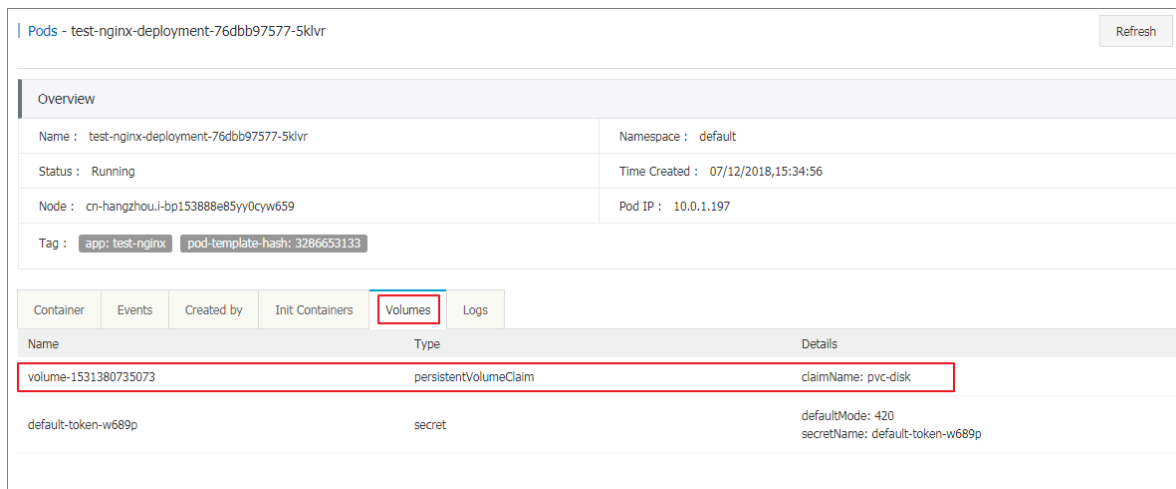
The screenshot shows the 'Application Configuration' page. The 'General' tab is selected, displaying configuration for the 'nginx' image. Below the general settings, the 'Volume' tab is visible, showing a table for 'Data Volume'. A red box highlights the 'Add cloud storage' section, which includes a table with columns 'Storage type', 'Mount source', and 'Container Path'. The table contains one entry: 'Disk' for storage type, 'pvc-disk' for mount source, and '/tmp' for container path.

5. See [#unique\\_36](#) to configure the test-nginx application, and click Create.
6. After the application is created, click Apply > Container Group in the left-side navigation pane. Find the container group to which the application belongs, and click Details.

The screenshot shows the 'Container Service - Kubernetes' page. The left navigation pane has 'Application' (1) and 'Pods' (2) highlighted. The main area shows a table of pods for the 'test-nginx-deployment-76dbb97577-5klvr' pod. The pod is in 'Running' status. A red box highlights the 'Clusters' and 'Namespace' dropdowns (3) and the 'Detail' button (4).

Name	Status	Pod IP	Node	Time Created	CPU	Memory	Actions
test-nginx-deployment-76dbb97577-5klvr	Running			07/12/2018, 15:34:56	0	0	Detail More

7. On the Container Group details page, click Storage to view the container group is properly bound to the PVC disk.



Pods - test-nginx-deployment-76dbb97577-5klvr Refresh

Overview

Name : test-nginx-deployment-76dbb97577-5klvr Namespace : default

Status : Running Time Created : 07/12/2018,15:34:56

Node : cn-hangzhou-l-bp153888e85yy0cyw659 Pod IP : 10.0.1.197

Tag : app: test-nginx pod-template-hash: 3286653133

Container Events Created by Init Containers **Volumes** Logs

Name	Type	Details
volume-1531380735073	persistentVolumeClaim	claimName: pvc-disk
default-token-w689p	secret	defaultMode: 420 secretName: default-token-w689p

## 1.14 Logs

### 1.14.1 Application log management

A Kubernetes cluster that runs on Alibaba Cloud Container Service provides you with multiple methods to manage application logs.

- Following the instructions of [#unique\\_80](#), you can make the best use of the functions provided by Alibaba Cloud Log Service, such as log statistics and analysis.
- With [Log-pilot](#), an open source project provided by Alibaba Cloud Container Service, and [#unique\\_81](#), you can easily build your own application log clusters.

### 1.14.2 View cluster logs

#### Context

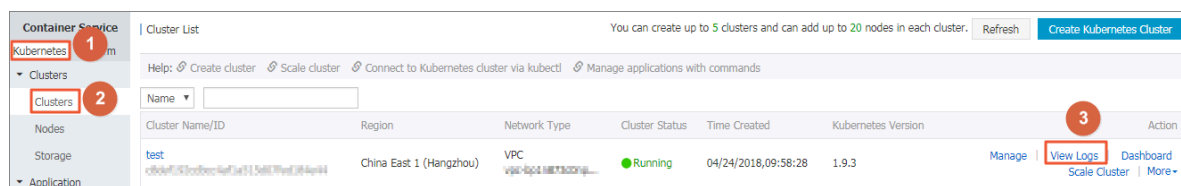
You can view the cluster operation logs by using the simple log service of Container Service.

#### Procedure

- Log on to the [Container Service console](#).
- Click Kubernetes > Clusters in the left-side navigation pane.



### 3. Click View Logs at the right of the cluster.



### View the cluster operation information.

Cluster Logs: test <a href="#">Back to Cluster List</a>		Refresh
Detailed resource deployment logs: <a href="#">Stack Events</a>		
Time	Information	
04/24/2018,13:55:38	c8def192cdbcc4af1a515d07fed184e44   Start to client.DescribeTemplate	
04/24/2018,13:55:35	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,11:27:38	c8def192cdbcc4af1a515d07fed184e44   Start to client.DescribeTemplate	
04/24/2018,11:27:36	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,11:27:08	c8def192cdbcc4af1a515d07fed184e44   Start to client.DescribeTemplate	
04/24/2018,11:27:06	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,11:26:55	c8def192cdbcc4af1a515d07fed184e44   Start to client.DescribeTemplate	
04/24/2018,11:26:54	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,11:25:02	c8def192cdbcc4af1a515d07fed184e44   Start to client.DescribeTemplate	
04/24/2018,11:25:00	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,10:16:42	c8def192cdbcc4af1a515d07fed184e44   Set up k8s DNS configuration successfully	
04/24/2018,10:15:36	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	
04/24/2018,10:15:36	c8def192cdbcc4af1a515d07fed184e44   Stack CREATE completed successfully:	
04/24/2018,10:15:34	c8def192cdbcc4af1a515d07fed184e44   Start describeStackInfo	

## 1.14.3 Collect Kubernetes logs

Log Service enables Logtail to collect Kubernetes cluster logs, and uses the CustomResourceDefinition (CRD) API to manage collection configurations. This document describes how to install and use Logtail to collect Kubernetes cluster logs.

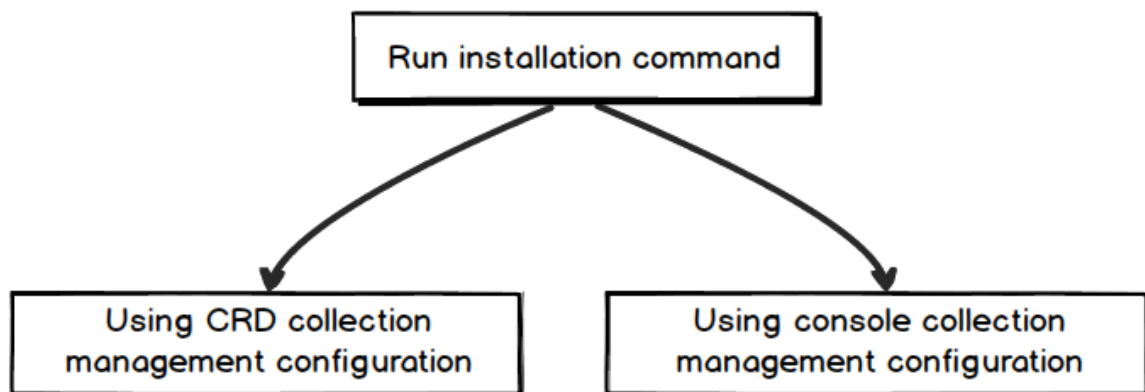
### Collection procedure

#### 1. Install the alibaba-log-controller Helm package.

## 2. Configure the collection.

You can configure the collection in the Log Service console or by using the CRD API as required. To configure the collection in the console, follow these steps:

Figure 1-1: Procedure



### Step 1 Install the package.

1. Log on to the Master node of the Alibaba Cloud Container Service for Kubernetes.

For how to log in, see [#unique\\_84](#).

2. Replace the parameters and run the following command.

`${ your_k8s_c luster_id }` to your Kubernetes cluster ID in the following installation command, and run this command:

```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/alicloud-log-k8s-install.sh -O alicloud-log-k8s-install.sh; chmod 744 ./alicloud-log-k8s-install.sh; sh ./alicloud-log-k8s-install.sh ${ your_k8s_c luster_id }
```

### Installation example

Run the installation command to obtain the following echo:

```
[root@izbp*****biaZ ~]# wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/alicloud-log-k8s-install.sh -O alicloud-log-k8s-install.sh; chmod 744 ./alicloud-log-k8s-install.sh; sh ./alicloud-log-k8s-install.sh c12ba20*****86939f0b
....
....
....
alibaba - cloud - log / Chart . yaml
alibaba - cloud - log / templates /
alibaba - cloud - log / templates / _helpers . tpl
```

```

alibaba - cloud - log / templates / alicloud - log - crd . yaml
alibaba - cloud - log / templates / logtail - daemonset . yaml
alibaba - cloud - log / templates / NOTES . txt
alibaba - cloud - log / values . yaml
NAME : alibaba - log - controller
LAST DEPLOYED : Wed May 16 18 : 43 : 06 2018
NAMESPACE : default
STATUS : DEPLOYED

RESOURCES :
==> v1beta1 / ClusterRoleBinding
NAME AGE
alibaba - log - controller 0s

==> v1beta1 / DaemonSet
NAME DESIRED CURRENT READY UP - TO - DATE AVAILABLE NODE
SELECTOR AGE
logtail 2 2 0 2 0 0s

==> v1beta1 / Deployment
NAME DESIRED CURRENT UP - TO - DATE AVAILABLE AGE
alibaba - log - controller 1 1 1 0 0s

==> v1 / Pod ( related )
NAME READY STATUS RESTARTS AGE
logtail - ff6rf 0 / 1 ContainerC reating 0 0s
logtail - q5s87 0 / 1 ContainerC reating 0 0s
alibaba - log - controller - 7cf6d7dbb5 - qvn6w 0 / 1 ContainerC
reating 0 0s

==> v1 / ServiceAccount
NAME SECRETS AGE
alibaba - log - controller 1 0s

==> v1beta1 / CustomResourceDefinition
NAME AGE
aliyunlogconfigs . log . alibabacloud . com 0s

==> v1beta1 / ClusterRole
alibaba - log - controller 0s

[ SUCCESS ] install helm package : alibaba - log - controller
success .

```

You can use `helm status alibaba - log - controller` to check the current Pod status. The Running status indicates a successful installation.

Then, Log Service creates the project that is named starting with k8s-log. You can search for this project by using the k8s-log keyword in the Log Service console.

**Step 2: Configure the collection.**

To create Logstore and collect standard output (stdout) from all K8s containers, follow these steps:

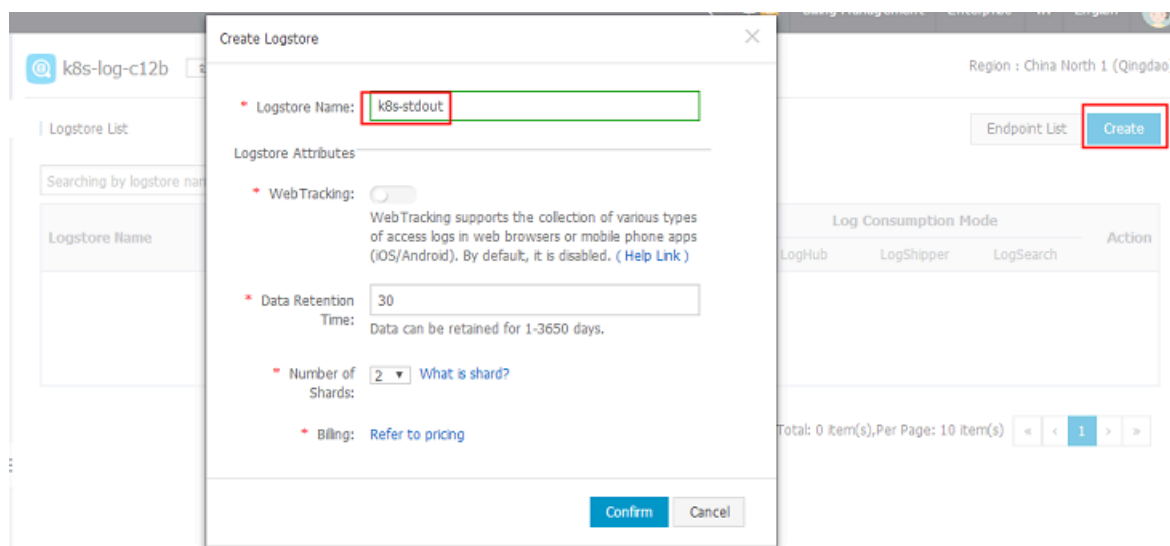
**1. Go to the Logstore List page.**

Click the project created in Step 1 to go to the Logstore List page.

**2. Create Logstore.**

Click Create in the upper-right corner, and in the dialog box that appears, create Logstore.

**Figure 1-2: Creating Logstore**

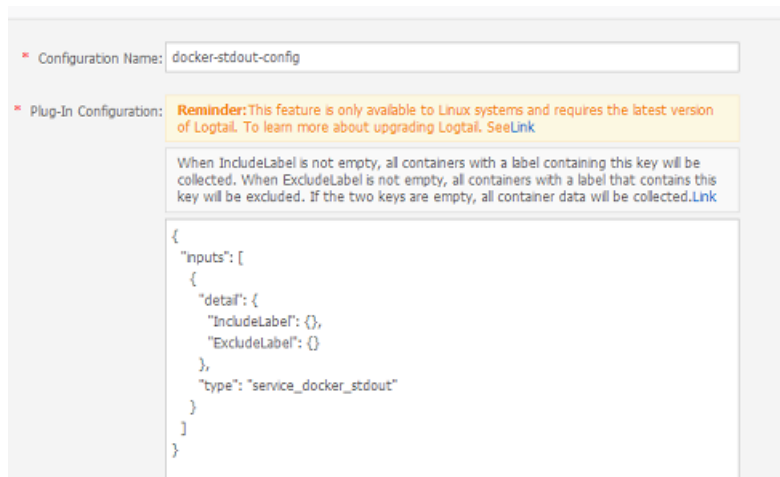


### 3. Configure the collection.

- a. Go to the Data Import Wizard page.
- b. Select Docker Stdout from Third-Party Software.

Click Apply to Machine Group on the configuration pages. Then, you can collect all stdout files from all containers.

Figure 1-3: Docker stdout



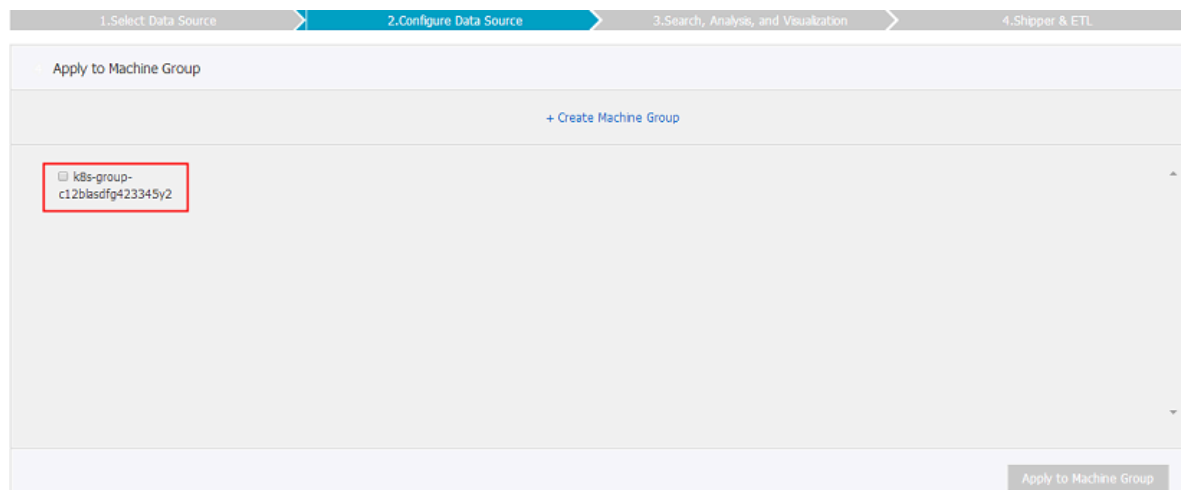
The screenshot shows the configuration interface for Docker Stdout. It includes a text input for 'Configuration Name' with the value 'docker-stdout-config'. Below it, the 'Plug-In Configuration' section contains a yellow reminder box stating: 'Reminder: This feature is only available to Linux systems and requires the latest version of Logtail. To learn more about upgrading Logtail, SeeLink'. A text box below the reminder explains: 'When IncludeLabel is not empty, all containers with a label containing this key will be collected. When ExcludeLabel is not empty, all containers with a label that contains this key will be excluded. If the two keys are empty, all container data will be collected.Link'. At the bottom, a JSON configuration is displayed:

```
{
  "inputs": [
    {
      "detail": {
        "IncludeLabel": {},
        "ExcludeLabel": {}
      },
      "type": "service_docker_stdout"
    }
  ]
}
```

### 4. Apply the configuration to the machine group.

On the Apply to Machine Group page, select a machine group, and click Next.

Figure 1-4: Applying the configuration to the machine group



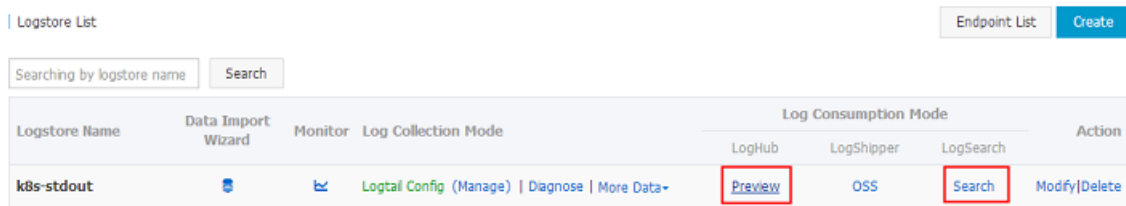
The screenshot shows the 'Apply to Machine Group' page. At the top, there is a progress bar with four steps: '1. Select Data Source', '2. Configure Data Source' (highlighted in blue), '3. Search, Analysis, and Visualization', and '4. Shipper & ETL'. Below the progress bar, the page title 'Apply to Machine Group' is displayed. A '+ Create Machine Group' link is visible. A list of machine groups is shown, with one group selected and highlighted by a red box: 'k8s-group-c12blasdfg423345y2'. At the bottom right, there is an 'Apply to Machine Group' button.

Now you have configured the collection. To configure indexes and log shipping, continue with the follow-up configurations. You can also exit the current page to complete the configuration.

## View collected logs

Based on the collection configuration, Logtail can collect stdout logs one minute after a container in your cluster receives stdout input. On the Logstore List page, click **Preview** to quickly preview collected logs, or click **Search** to customize searching and analysis of these logs.

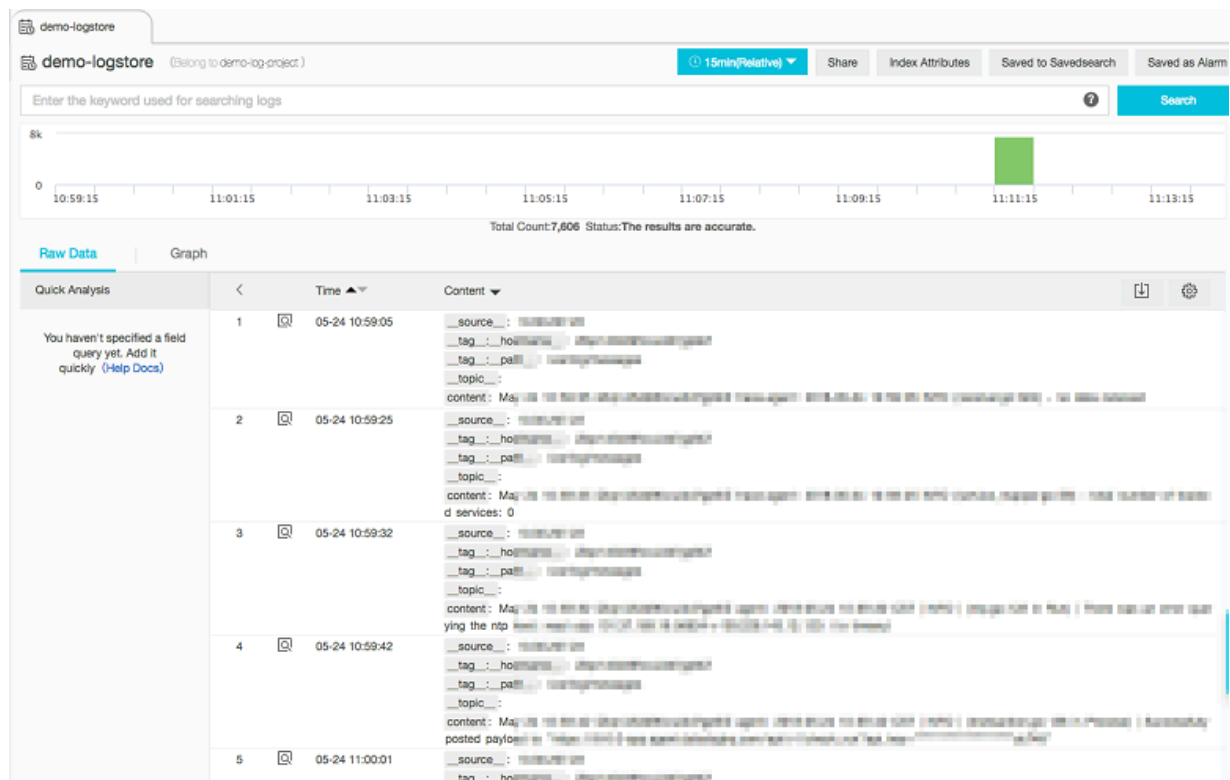
Figure 1-5: Previewing and searching



Logstore Name	Data Import Wizard	Monitor	Log Collection Mode	Log Consumption Mode			Action
				LogHub	LogShipper	LogSearch	
k8s-stdout			Logtail Config (Manage)   Diagnose   More Data-	<b>Preview</b>	OSS	<b>Search</b>	Modify Delete

As shown in the following image of the Search page, click any keyword of a log to start quick searching, or enter the keyword in the search box to search the specified logs.

Figure 1-6: Searching logs



## Other methods for configuring collections

For more information about other methods for configuring collections, see:

## Console Configuration

For more information about Console configuration, see:

- [Container text log \(recommended\)](#)
- [Container standard output \(recommended\)](#)
- [Host text file](#)

By default, the root directory of the host is mounted to the `/ logtail_ho st` directory of the Logtail container. You must add this prefix when configuring the path. For example, to collect data in the `/ home / logs / app_log /` directory of the host, set the log path on the configuration page to `/ logtail_ho st / home / logs / app_log /`.

## CRD Configuration

For more information about CRD(CustomResourceDefinition) configuration, see [#unique\\_88](#).

## 1.14.4 Configure Log4jAppender for Kubernetes and Log Service

Log4j is an open-source project of Apache, which consists of three important components: log level, log output destination, and log output format. By configuring Log4jAppender, you can set the log output destination to console, file, GUI component, socket server, NT event recorder, or UNIX Syslog daemon.

This document introduces how to configure a YAML file to output Alibaba Cloud Container Service Kubernetes cluster logs to Alibaba Cloud Log Service, without modifying the application codes. In this document, deploy a sample API application in the Kubernetes cluster for demonstration.

### Prerequisites

- You have activated Container Service and created a Kubernetes cluster.

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

- Enable AccessKey or Resource Access Management (RAM). Make sure you have sufficient access permissions. Use the AccessKey in this example.

## Step 1 Configure Log4jAppender in Alibaba Cloud Log Service

1. Log on to the [Log Service console](#).
2. On the Project List page, click Create Project in the upper-right corner. Complete the configurations and then click Confirm to create the project.

In this example, create a project named k8s-log4j and select the same region ( China East 1 (Hangzhou)) as the Kubernetes cluster.



### Note:

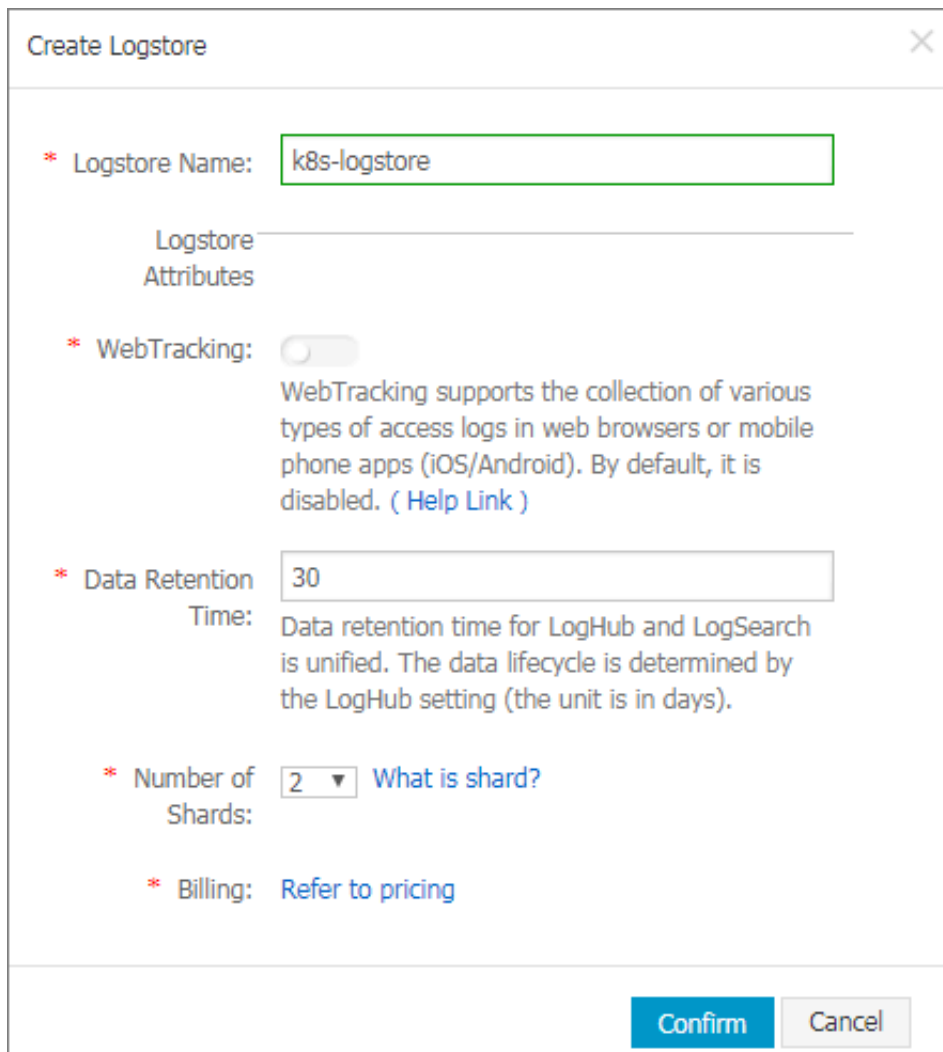
Generally, create a Log Service project in the same region as the Kubernetes cluster. When the Kubernetes cluster and Log Service project are in the same region, log data is transmitted by using the intranet, which saves the Internet bandwidth cost and time of data transmission because of different regions, and implements the best practice of real-time collection and quick query.

3. After being created, the project k8s-log4j is displayed on the Project List page. Click the project name.
4. The Logstore List page appears. Click Create in the upper-right corner.



5. Complete the configurations and then click Confirm.

In this example, create a Logstore named k8s-logstore.

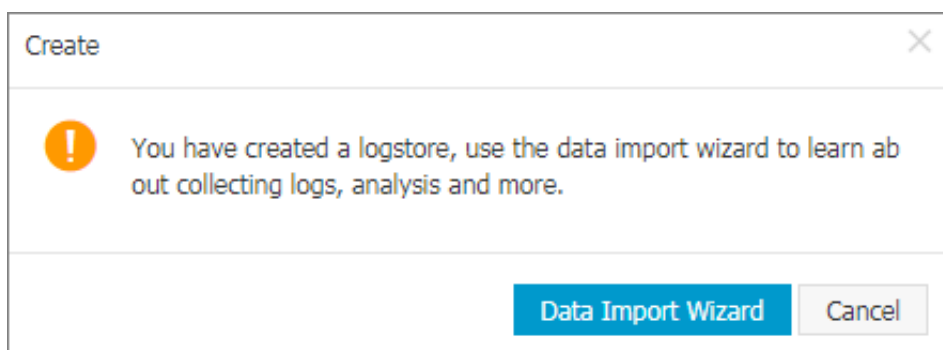


The 'Create Logstore' dialog box contains the following fields and options:

- Logstore Name:** A text input field containing 'k8s-logstore'.
- Logstore Attributes:** A section header.
- WebTracking:** A toggle switch that is currently disabled. Below it, text states: 'WebTracking supports the collection of various types of access logs in web browsers or mobile phone apps (iOS/Android). By default, it is disabled. ( [Help Link](#) )'.
- Data Retention Time:** A text input field containing '30'. Below it, text states: 'Data retention time for LogHub and LogSearch is unified. The data lifecycle is determined by the LogHub setting (the unit is in days).'
- Number of Shards:** A dropdown menu showing '2' and a link 'What is shard?'.
- Billing:** A link 'Refer to pricing'.

At the bottom right, there are two buttons: 'Confirm' (highlighted in blue) and 'Cancel'.

6. Then, a dialog box asking you to use the data import wizard appears.

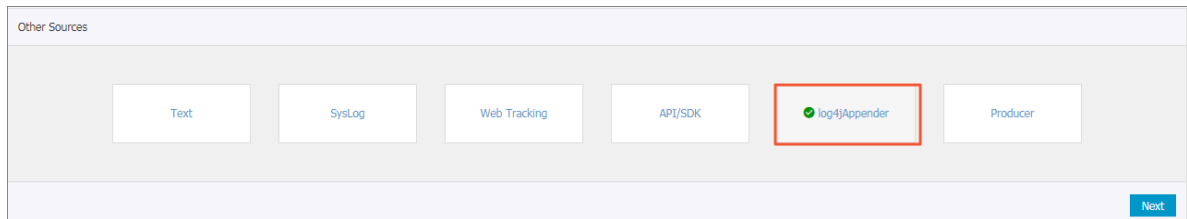


The 'Create' dialog box displays a notification with an orange exclamation mark icon. The text reads: 'You have created a logstore, use the data import wizard to learn about collecting logs, analysis and more.'

At the bottom right, there are two buttons: 'Data Import Wizard' (highlighted in blue) and 'Cancel'.

7. Click Data Import Wizard. In the Select Data Source step, select log4jAppender under Other Sources and then complete the configurations as instructed on the page.

Use the default configurations in this example. Configure the settings according to the specific scenarios of log data.



## Step 2 Configure Log4jAppender in the Kubernetes cluster

In this example, use the sample YAML files [demo-deployment](#) and [demo-service](#) for demonstration.

1. Connect to your Kubernetes cluster.

For more information, see [#unique\\_10](#) or [#unique\\_20](#).

2. Obtain the `demo - deployment . yaml` file and configure the environment variable `JAVA_OPTS` to collect logs from the Kubernetes cluster.

The sample orchestration of the `demo - deployment . yaml` file is as follows:

```
apiVersion : apps / v1beta2
kind : Deployment
metadata :
  name : log4j - appender - demo - spring - boot
  labels :
    app : log4j - appender
spec :
  replicas : 1
  selector :
    matchLabels :
      app : log4j - appender
  template :
    metadata :
      labels :
        app : log4j - appender
    spec :
      containers :
        - name : log4j - appender - demo - spring - boot
          image : registry . cn - hangzhou . aliyuncs . com /
jaegertracing / log4j - appender - demo - spring - boot : 0 . 0 .
2
          env :
            - name : JAVA_OPTS ## Note
              value : "- Dproject ={ your_project } - Dlogstore ={
your_logstore } - Dendpoint ={ your_endpoint } - Daccess_ke
```

```
y_id={ your_access_key_id } - Daccess_key_id={ your_access_key_secret }"
  ports :
  - containerPort : 8080
```

Wherein:

- `Dproject` : The name of the used Alibaba Cloud Log Service project. In this example, it is `k8s-log4j`.
- `Dlogstore` : The name of the used Alibaba Cloud Log Service Logstore. In this example, it is `k8s-logstore`.
- `Dendpoint` : The service endpoint of Log Service. You must configure your service endpoint according to the region where the Log Service project resides. For more information, see [Service endpoint](#). In this example, it is `cn-hangzhou.log.aliyuncs.com`.
- `Daccess_key_id` : Your AccessKey ID.
- `Daccess_key_secret` : Your AccessKey Secret.

3. Run the following command in the command line to create the deployment:

```
kubectl create -f demo - deployment . yaml
```

4. Obtain the `demo - service . yaml` file and run the following command to create the service.

No need to modify the configurations in the `demo - service . yaml` file.

```
kubectl create -f demo - service . yaml
```

### Step 3 Test to generate Kubernetes cluster logs

You can run the `kubectl get` command to view the deployment status of the resource object. Wait until the deployment and the service are successfully deployed. Then, run the `kubectl get svc` command to view the external access IP of the service, that is, the EXTERNAL-IP.

```
$ kubectl get svc
NAME      TYPE      CLUSTER - IP      EXTERNAL - IP      PORT ( S )      AGE
log4j - appender - demo - spring - boot - svc      LoadBalancer      172
. 21 . XX . XX      120 . 55 . XXX . XXX      8080 : 30398 / TCP      1h
```

In this example, test to generate Kubernetes cluster logs by running the `login` command, wherein, `K8S_SERVICE_IP` is the `EXTERNAL - IP`.



Note:

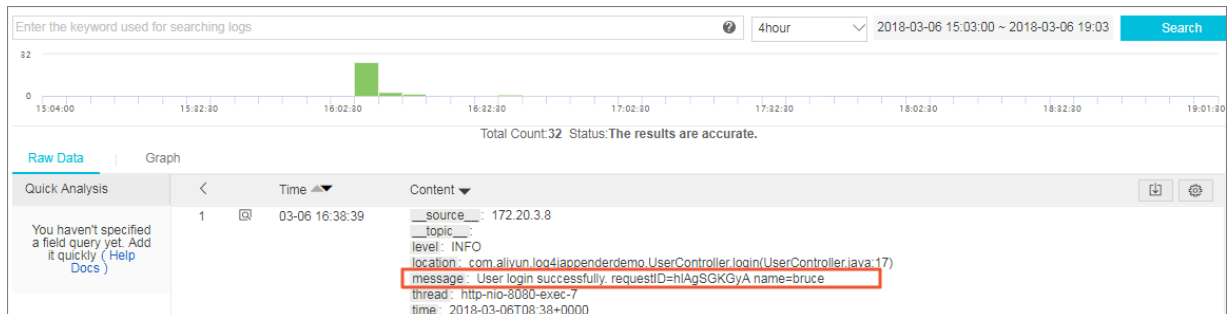
See [GitHub log4j-appender-demo](#) to view the complete collection of APIs.

```
curl http ://${ K8S_SERVIC E_IP }: 8080 / login ? name = bruce
```

#### Step 4 View logs in Alibaba Cloud Log Service

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

Click the project name and click Search at the right of the Logstore k8s-logstore to view the output logs of the Kubernetes cluster.



The output content of the log corresponds to the preceding command. This example demonstrates how to output the logs of the sample application to Alibaba Cloud Log Service. By completing the preceding steps, you can configure Log4JAppender in Alibaba Cloud and implement advanced functions such as collecting logs in real time , filtering data, and querying logs by using Alibaba Cloud Log Service.

### 1.14.5 A solution to log collection problems of Kubernetes clusters by using log-pilot, Elasticsearch, and Kibana

Requirements for logs of distributed Kubernetes clusters always bother developers . This is mainly because of the characteristics of containers and the defects of log collection tools.

- Characteristics of containers:
  - Many collection targets: The characteristics of containers cause the number of collection targets is large, which requires to collect the container logs and container stdout. Currently, no good tool can collect file logs from containers dynamically. Different data sources have different collection softwares. However, no one-stop collection tool exists.
  - Difficulty caused by auto scaling: Kubernetes clusters are in the distributed mode. The auto scaling of services and the environment brings great difficulty to log collection. You cannot configure the log collection path in advance, the

same as what you do in the traditional virtual machine (VM) environment. The dynamic collection and data integrity are great challenges.

- Defects of current log collection tools:
  - Lack the capability to dynamically configure log collection: The current log collection tools require you to manually configure the log collection method and path in advance. These tools cannot dynamically configure the log collection because they cannot automatically detect the lifecycle changes or dynamic migration of containers.
  - Log collection problems such as logs are duplicate or lost: Some of the current log collection tools collect logs by using the tail method. Logs may be lost in this way. For example, the application is writing logs when the log collection tool is being restarted. Logs written during this period may be lost. Generally, the conservative solution is to collect logs of 1 MB or 2 MB previous to the current log by default. However, this may cause the duplicate log collection.
  - Log sources without clear marks: An application may have multiple containers that output the same application logs. After all the application logs are collected to a unified log storage backend, you cannot know a log is generated on which application container of which node when querying logs.

This document introduces log-pilot, a tool to collect Docker logs, and uses the tool together with Elasticsearch and Kibana to provide a one-stop solution to log collection problems in the Kubernetes environment.

## Introduction on log-pilot

Log-pilot is an intelligent tool used to collect container logs, which not only collects container logs and outputs these logs to multiple types of log storage backends efficiently and conveniently, but also dynamically discovers and collects log files from containers.

Log-pilot uses declarative configuration to manage container events strongly and obtain the stdout and file logs of containers, which solves the problem of auto scaling. Besides, log-pilot has the functions of automatic discovery, maintenance of checkpoint and handle, and automatic tagging for log data, which effectively deals with the problems such as dynamic configuration, duplicate logs, lost logs, and log source marking.

Currently, log-pilot is completely open-source in GitHub. The project address is <https://github.com/AliyunContainerService/log-pilot>. You can know more implementation principles about it.

### Declarative configuration for container logs

Log-pilot supports managing container events, can dynamically listen to the event changes of containers, parse the changes according to the container labels, generate the configuration file of log collection, and then provide the file to collection plug-in to collect logs.

For Kubernetes clusters, log-pilot can dynamically generate the configuration file of log collection according to the environment variable `aliyun_log s_ $ name = $ path`. This environment variable contains the following two variables:

- One variable is \$name, a custom string which indicates different meanings in different scenarios. In this scenario, \$name indicates index when collecting logs to Elasticsearch.
- The other is \$path which supports two input modes, stdout and paths of log files within containers, respectively corresponding to the standard output of logs and log files within containers.
  - Stdout indicates to collect standard output logs from containers. In this example, to collect Tomcat container logs, configure the label `aliyun . logs . catalina = stdout` to collect standard output logs of Tomcat.
  - The path of a log file within a container also supports wildcards. To collect logs within the Tomcat container, configure the environment variable `aliyun_log s_access = /usr / local / tomcat / logs / *. log`. To not use the keyword aliyun, you can use the environment variable `PILOT_LOG_PREFIX`, which is also provided by log-pilot, to specify the prefix of your declarative log configuration. For example, `PILOT_LOG_ PREFIX : " aliyun , custom "`.

Besides, log-pilot supports multiple log parsing formats, including none, JSON, CSV, Nginx, apache2, and regexp. You can use the `aliyun_log s_ $ name_format = < format >` label to tell log-pilot to use what format to parse logs when collecting logs.

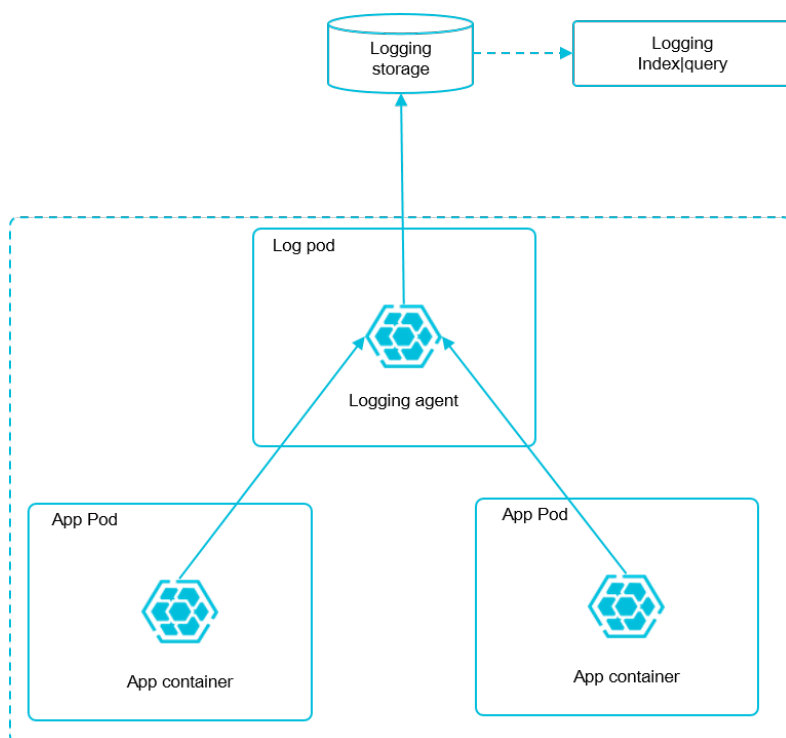
Log-pilot also supports custom tags. If you configure `aliyun_log s_ $ name_tags = " K1 = V1 , K2 = V2 "` in the environment variable, K1=V1 and K2=V2 are collected to log output of the container during the log collection. Custom tags help

you tag the log generation environment for convenient statistics, routing, and filter of logs.

### Log collection mode

In this document, deploy a log-pilot on each machine and collect all the Docker application logs from the machines.

Compared with deploying a logging container on each pod, the most obvious advantage of this solution is less occupied resources. The larger the cluster scale is, the more obvious the advantage is. This solution is also recommended in the community.



### Prerequisites

You have activated Container Service and created a Kubernetes cluster. In this example, create a Kubernetes cluster in China East 1 (Hangzhou).

#### Step 1 Deploy Elasticsearch

1. Connect to your Kubernetes cluster. For more information, see [#unique\\_13](#) or [#unique\\_10](#).
2. Deploy the resource object related to Elasticsearch first. Then, enter the following orchestration template. This orchestration template includes an elasticsearch-api

service, an elasticsearch-discovery service, and a status set of Elasticsearch. All of these objects are deployed under the namespace kube-system.

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/elasticsearch.yml
```

3. After the successful deployment, corresponding objects are under the namespace kube-system. Run the following commands to check the running status:

```
$ kubectl get svc,StatefulSet -n=kube-system
NAME      TYPE      CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
svc / elasticsearch-apiserver ClusterIP      172.17.0.134    <
none > 9200 / TCP    22h
svc / elasticsearch-discovery ClusterIP      172.17.0.91     <
none > 9300 / TCP    22h
...
NAME      DESIRED    CURRENT    AGE
statefulset / elasticsearch-apiserver 3          3          22h
```

## Step 2 Deploy log-pilot and the Kibana service

1. Deploy the log-pilot log collection tool. The orchestration template is as follows:

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/log-pilot.yml
```

2. Deploy the Kibana service. The sample orchestration template contains a service and a deployment.

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/kibana.yml
```

## Step 3 Deploy the test application Tomcat

After deploying the log tool set of Elasticsearch + log-pilot + Kibana, deploy a test application Tomcat to test whether or not logs can be successfully collected, indexed, and displayed.

The orchestration template is as follows:

```
apiVersion : v1
kind : Pod
metadata :
  name : tomcat
  namespace : default
  labels :
    name : tomcat
spec :
  containers :
    - image : tomcat
      name : tomcat-test
      volumeMounts :
        - mountPath : /usr/local/tomcat/logs
          name : accesslogs
      env :
```



```

- name : aliyun_log s_catalina
  value : " stdout " ## Collect standard output logs .
- name : aliyun_log s_access
  value : "/ usr / local / tomcat / logs / catalina . *. log "
## Collect log files within the container .
volumes :
- name : accesslogs
  emptyDir : {}

```

The Tomcat image is a Docker image that both uses stdout and file logs. In the preceding orchestration, the log collection configuration file is dynamically generated by defining the environment variable in the pod. See the following descriptions for the environment variable:

- `aliyun_log s_catalina = stdout` indicates to collect stdout logs from the container.
- `aliyun_log s_access = / usr / local / tomcat / logs / catalina . *. log` indicates to collect all the log files whose name matches `catalina . *. log` under the directory `/ usr / local / tomcat / logs /` from the container.

In the Elasticsearch scenario of this solution, the `$ name` in the environment variable indicates index. In this example, `$ name` is `catalina` and `access`.

#### Step 4 Expose the Kibana service to Internet

The Kibana service deployed in the preceding section is of the NodePort type, which cannot be accessed from the Internet by default. Therefore, create an Ingress in this document to access the Kibana service from Internet and test whether or not logs are successfully indexed and displayed.

1. Create an Ingress to access the Kibana service from Internet. In this example, use the simple routing service to create an Ingress. For more information, see [#unique\\_91](#). The orchestration template of the Ingress is as follows:

```

apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : kibana - ingress
  namespace : kube - system # Make sure the namespace is
the same as that of the Kibana service .
spec :
  rules :
  - http :
    paths :
    - path : /
      backend :

```

```

serviceName : kibana # Enter the name of the
Kibana service .
servicePort : 80 # Enter the port exposed by
the Kibana service .

```

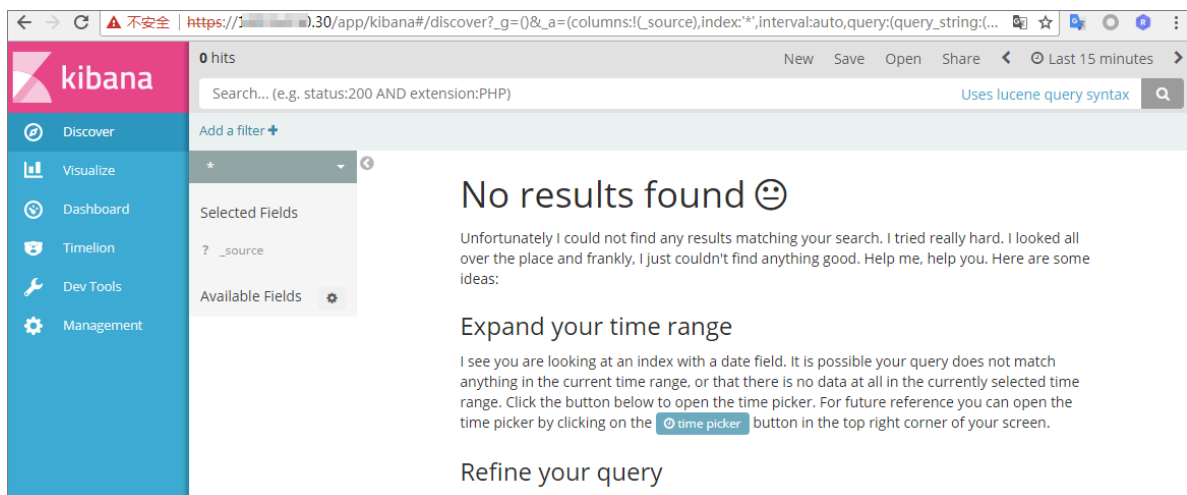
2. After the Ingress is successfully created, run the following commands to obtain the access address of the Ingress:

```

$ kubectl get ingress -n=kube-system
NAME HOSTS ADDRESS PORTS AGE
shared-dns * 120.55.150.30 80 5m

```

3. Access the address in the browser as follows.



4. Click Management in the left-side navigation pane. Then, click Index Patterns > Create Index Pattern. The detailed index name is the `$name` variable suffixed with a time string. You can create an index pattern by using the wildcard `*`. In this example, use `$name*` to create an index pattern.

You can also run the following commands to enter the corresponding pod of Elasticsearch and list all the indexes of Elasticsearch:

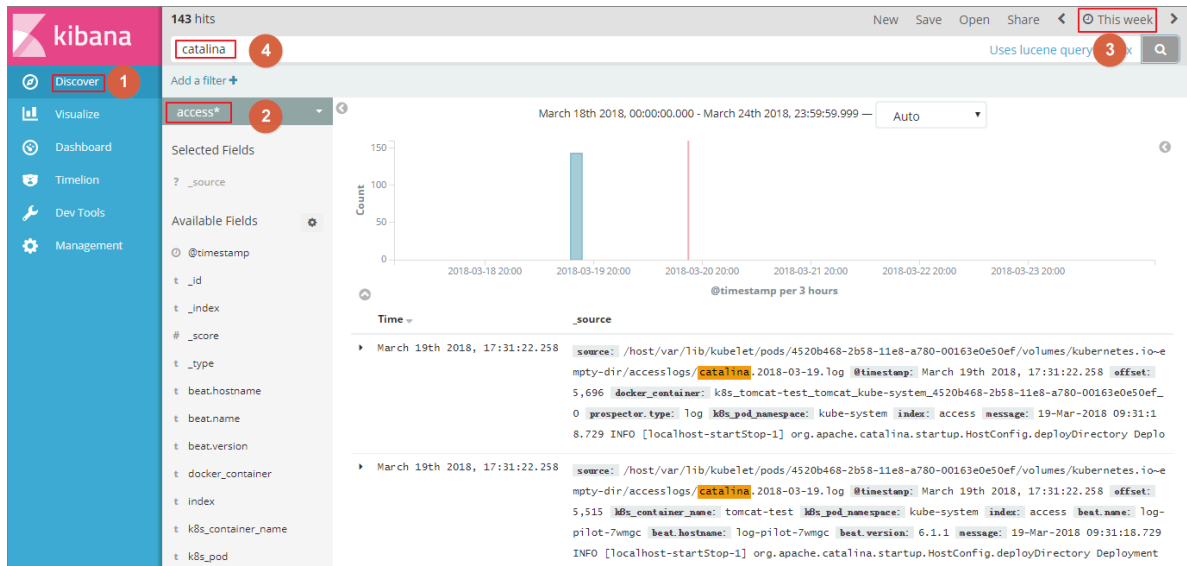
```

$ kubectl get pods -n=kube-system # Find the
corresponding pod of Elasticsearch .
...
$ kubectl exec -it elasticsearch-1 bash # Enter a
pod of Elasticsearch .
...
$ 'curl 'localhost : 9200 / _cat / indices ? v ' ## List all
the indexes .
health status index uuid pri rep docs . count docs .
deleted store . size pri . store . size
green open . kibana x06jj19PS4 C1m6Ajo51P Wg 1 1 4
0 53 . 6kb 26 . 8kb
green open access - 2018 . 03 . 19 txd3tG - NR6 -
guqmMEKKzE w 5 1 143 0 823 . 5kb 411 . 7kb

```

```
green    open    catalina - 2018 . 03 . 19    ZgtWd16FQ7    qqJNNWXxFP
cQ    5    1    143    0    915 . 5kb    457 . 5kb
```

5. After successfully creating the indexes, click Discover in the left-side navigation pane, select the created index and the corresponding time range, and then enter the related field in the search box to query logs.



Then, you have successfully tested the solution to log collection problems of Alibaba Cloud Kubernetes clusters based on log-pilot, Elasticsearch, and Kibana. By using this solution, you can deal with requirements for logs of distributed Kubernetes clusters effectively, improve the Operation and Maintenance and operational efficiencies, and guarantee the continuous and stable running of the system.

## 1.15 Security

### Authorization

The same as swarm clusters, Kubernetes clusters support authorizing RAM users to perform operations on clusters.

For more information, see [#unique\\_93](#).

### Full-link TLS certificates

The following communication links in Container Service Kubernetes clusters are verified by TLS certificates to prevent the communication from being eavesdropped or tampered:

- `kubelet` on worker nodes actively communicates with `apiserver` on master nodes

- `apiserver` on master nodes actively communicates with `kubelet` on worker nodes

During initialization, the master node uses SSH tunnels to connect to the SSH service of other nodes (port 22) for initialization.

#### Native secret & RBAC support

Kubernetes secrets are used to store sensitive information such as passwords, OAuth tokens, and SSH keys. Using plain text to write sensitive information to a pod YAML file or a Docker image may leak the information, while using secrets avoids such security risks effectively.

For more information, see [Secret](#).

Role-Based Access Control (RBAC) uses the Kubernetes built-in API group to drive authorization and authentication, which allows you to use APIs to manage pods that correspond to different roles, and the access permissions of roles.

For more information, see [Using RBAC authorization](#).

#### Network policy

In a Kubernetes cluster, pods on different nodes can communicate with each other by default. In some scenarios, to reduce risks, the network intercommunication among different business services is not allowed and you must introduce the network policy. In Kubernetes clusters, you can use the Canal network driver to implement the support for network policy.

#### Image security scan

Kubernetes clusters can use Container Registry to manage images, which allows you to perform image security scan.

Image security scan identifies the security risks in images quickly and reduces the possibility of applications running on your Kubernetes cluster being attacked.

For more information, see [Image security scan](#).

#### Security group and Internet access

By default, each newly created Kubernetes cluster is assigned a new security group with the minimal security risk. This security group only allows ICMP for the Internet inbound.

By default, you cannot use Internet SSH to access your clusters. To use Internet SSH to connect to the cluster nodes, see [#unique\\_10](#).

The cluster nodes access the Internet by using the NAT Gateway, which further reduces the security risks.

## 1.16 FAQ

### 1.16.1 Collect Kubernetes diagnosis information

1. Download diagnosis script on the master node and add the operation permission.

```
curl -o /usr/local/bin/diagnose_k 8s .sh http://
aliacs-k8s-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com
/public/diagnose/diagnose_k 8s .sh
chmod u+x /usr/local/bin/diagnose_k 8s .sh
```

2. Run the diagnosis script.

```
diagnose_k 8s .sh

+ echo 'please get diagnose_1 514939155 .tar .gz for
diagnostic s' ## The generated log file name is
different every time you run the diagnosis script .
please get diagnose_1 514939155 .tar .gz for
diagnostic s
+ echo 'Upload diagnose_1 514939155 .tar .gz '
Upload diagnose_1 514939155 .tar .gz
```

3. Upload the generated logs.

```
cd /usr/local/bin
ls -ltr | grep diagnose_1 514939155 .tar .gz ## Replace
with the generated log file name .
```

### 1.16.2 FAQ about storage volumes

Storage volumes cannot be mounted

Check if flexvolume is installed.

Execute the following command on the master node:

#	kubectl	get	pod	-n	kube-system		grep	flexvolume
flexvolume	-	4wh8s	1	/ 1	Running	0	8d	
flexvolume	-	65z49	1	/ 1	Running	0	8d	
flexvolume	-	bpc6s	1	/ 1	Running	0	8d	
flexvolume	-	l8pml	1	/ 1	Running	0	8d	
flexvolume	-	mzkpv	1	/ 1	Running	0	8d	
flexvolume	-	wbfhv	1	/ 1	Running	0	8d	

```
flexvolume - xf5cs 1 / 1 Running 0 8d
```

Check if the flexvolume pod status is Running and if the number of running flexvolume pods is the same as the number of nodes.

If not, see [#unique\\_73](#).

If the flexvolume pod status is not running, see the running log analysis of the plug-in .

Check if the dynamic storage plug-in is installed

To use the dynamic storage function of a cloud disk, execute the following command to verify the dynamic storage plug-in is installed:

```
# kubectl get pod -n kube-system | grep alicloud - disk
alicloud - disk - controller - 8679c9fc76 - lq6zb 1 / 1 Running
0 7d
```

If not, see [#unique\\_73](#).

If the dynamic storage plug-in status is not running, see the running log analysis of the plug-in.

How to view types of storage logs?

View flexvolume logs by executing commands on the master1 node

Execute the following get command to view the error pod:

```
# kubectl get pod -n kube-system | grep flexvolume
```

Execute the following log command to view the log for the error pod:

```
# kubectl logs flexvolume - 4wh8s -n kube-system
# kubectl describe pod flexvolume - 4wh8s -n kube-system

# The last several lines in the pod description are
the descriptions of pod running status. You can
analyze pod errors based on the descriptions.
```

View drive logs of the cloud disk, Network Attached Storage (NAS), and Object Storage Service (OSS):

```
# View the persistent logs on the host node ;
# If a pod mount fails, view the address of the
node on which the pod resides :

# kubectl describe pod nginx - 97dc96f7b - xbx8t | grep
Node
```

```

Node : cn - hangzhou . i - bp19myla3u vnt6zihejb / 192 . 168 . 247
. 85
Node - Selectors : < none >

# Log on to the node to view logs :

# ssh 192 . 168 . 247 . 85
# ls / var / log / alicloud / flexvolume *
flexvolume _disk . log flexvolume _nas . log flexvolume _o #
ss . log

You can see logs mounted on the cloud disk , NAS ,
and OSS ;

```

**View provsioner plug-in logs by executing commands on the master1 node**

**Execute the following get command to view the error pod:**

```
# kubectl get pod - n kube - system | grep alicloud - disk
```

**Execute the log command to view the log for the error pod:**

```

# kubectl logs alicloud - disk - controller - 8679c9fc76 - lq6zb
- n kube - system
# kubectl describe pod alicloud - disk - controller -
8679c9fc76 - lq6zb - n kube - system

# The last several lines in the pod descriptio n are
the descriptio ns of pod running status . You can
analyze pod errors based on the descriptio ns .

```

**View Kubelet logs**

```

# If a pod mount fails , view the address of the
node on which the pod resides :

# kubectl describe pod nginx - 97dc96f7b - xbx8t | grep
Node
Node : cn - hangzhou . i - bp19myla3u vnt6zihejb / 192 . 168 . 247
. 85
Node - Selectors : < none >

# Log on to the node to view kubelet logs :

# ssh 192 . 168 . 247 . 85
# journalctl - u kubelet - r - n 1000 &> kubelet . log

# The value of - n indicates the number of log lines
that you expect to see ;

```

The above are methods to obtain error logs of flexvolume, provsioner, and kubelet. If the logs cannot help you to repair the status, contact Alibaba Cloud technical support with the logs.

## FAQ about cloud disks

### Cloud disk mount fails with timeout errors

If the node is added manually, the failure may be caused by problem about Security Token Service (STS) permissions. You need to manually configure Resource Access Management (RAM) permissions: [#unique\\_97](#).

### Cloud disk mount fails with size errors

The following are size requirements for creating a cloud disk:



#### Note:

- Basic cloud disk: Minimum 5Gi
- Ultra cloud disk: Minimum 20Gi
- SSD cloud disk: Minimum 20Gi

### Cloud disk mount fails with zone errors

When the ECS mounts a cloud disk, they must be in the same zone under the same region. Otherwise, the cloud disk cannot be mounted successfully.

After your system is upgraded, the cloud disk sometimes reports input/output error

1. Upgrade flexvolume to v1.9.7-42e8198 or later.
2. Rebuild pods that have already gone wrong.

Upgrading command:

```
# kubectl set image daemonset / flexvolume acs - flexvolume =  
registry.cn-hangzhou.aliyuncs.com / acs / flexvolume : v1.9  
.7 - 42e8198 - n kube - system
```

**Flexvolume version information:** To obtain the latest version of flexvolume, log on to the container image service console, click Image search in the left-side navigation pane, and search for acs/flexvolume.

## FAQ about NAS

### NAS mount time is too long

If the NAS volume contains a large amount of files and the chmod parameter is configured in the mount template, the mount time may be too long. To solve this problem, remove the chmod parameter.

### NAS mount fails with the timeout error



Check if the NAS mount point and the cluster are within the same Virtual Private Cloud (VPC). If not, NAS cannot be mounted.

## FAQ about OSS

### OSS mount fails

Check if the AK used is correct.

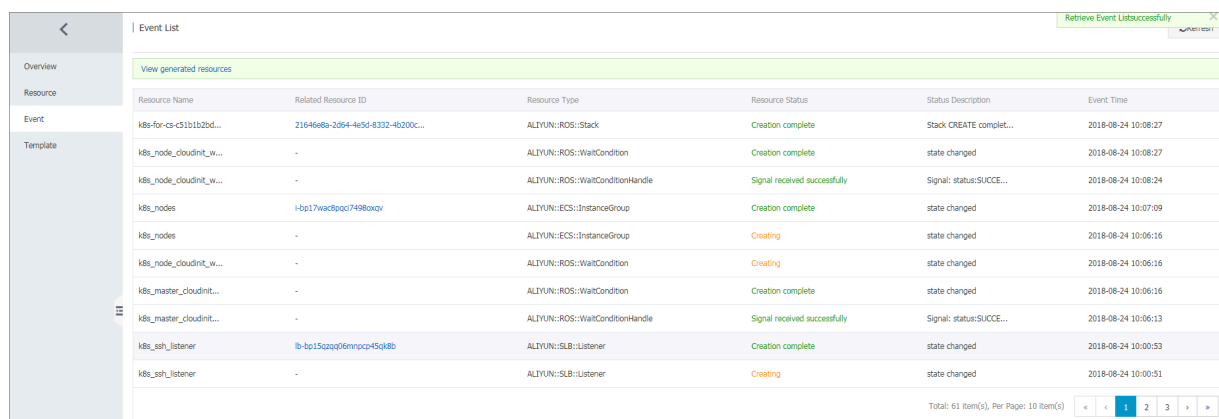
## 1.16.3 Failed to create a Kubernetes cluster

### Check the cause of failure

You can check the cause of cluster creation failure by viewing the cluster creation events.

Log on to the [Resource Orchestration Service \(ROS\) console](#).

Select the region in which the cluster resides. Click Manage at the right of the cluster. Click Event in the left-side navigation pane. Move the cursor over the failed event to view the specific error message of the failure.



Resource Name	Related Resource ID	Resource Type	Resource Status	Status Description	Event Time
k8s-for-cs-c51b1b2bd...	21646e8a-3d94-4e5d-8332-4b200c...	ALTYUN:ROS::Stack	Creation complete	Stack CREATE complet...	2018-08-24 10:08:27
k8s_node_cloudinit_w...	-	ALTYUN:ROS::WaitCondition	Creation complete	state changed	2018-08-24 10:08:27
k8s_node_cloudinit_w...	-	ALTYUN:ROS::WaitConditionHandle	Signal received successfully	Signal: status:SUCCE...	2018-08-24 10:08:24
k8s_nodes	l-bp17wcdpggc7498oazv	ALTYUN:ECS::InstanceGroup	Creation complete	state changed	2018-08-24 10:07:09
k8s_nodes	-	ALTYUN:ECS::InstanceGroup	Creating	state changed	2018-08-24 10:06:16
k8s_node_cloudinit_w...	-	ALTYUN:ROS::WaitCondition	Creating	state changed	2018-08-24 10:06:16
k8s_master_cloudinit...	-	ALTYUN:ROS::WaitCondition	Creation complete	state changed	2018-08-24 10:06:16
k8s_master_cloudinit...	-	ALTYUN:ROS::WaitConditionHandle	Signal received successfully	Signal: status:SUCCE...	2018-08-24 10:06:13
k8s_slb_listener	lb-bp15czzqg06mnpcc45qk8b	ALTYUN:SLB::Listener	Creation complete	state changed	2018-08-24 10:00:53
k8s_slb_listener	-	ALTYUN:SLB::Listener	Creating	state changed	2018-08-24 10:00:51

If the preceding error message is displayed, it means that the cluster creation failed because the number of Virtual Private Cloud (VPC) instances has reached the quota.

### Failure codes and solutions

- **Code:** QuotaExceeded.Eip, **Message:** Elastic IP address quota exceeded  
**Solution:** Release unused EIPs, or open a ticket to raise the EIP quota.
- **The maximum number of SLB instances is exceeded. Code:** ORDER.QUANTITY\_INVALID  
**Solution:** Release unused SLB instances, or open a ticket to raise the SLB quota.

- Resource CREATE failed: `ResponseException: resources.k8s_vpc: VPC quota exceeded. Code: QuotaExceeded.Vpc`

**Solution:** Release unused VPCs, or open a ticket to raise the VPC quota.

- Resource CREATE failed: `ResponseException: resources.k8s_master_1: The specified image does not support cloud-init. Code: ImageNotSupportCloudInit`

**Solution:** When using custom image to create a cluster, the custom image used must be developed based on the latest Centos public cloud image.

- Status Code: 403 Code: `InvalidResourceType.NotSupported` Message: This resource type is not supported;

**Solution:** ECS is out of stock or the type of ECS instances you selected are not supported.

## 1.16.4 How to use private images in Kubernetes clusters

```
kubectl create secret docker-registry regsecret -- docker-
server = registry - internal . cn - hangzhou . aliyuncs . com --
docker-username = abc @ aliyun . com -- docker-password = xxxxxx
-- docker-email = abc @ aliyun . com
```

Where:

- `regsecret`: Specifies the secret key name and the name is customizable.
- `--docker-server`: Specifies the Docker repository address.
- `--docker-username`: Specifies the user name of the Docker repository.
- `--docker-password`: Specifies the logon password of the Docker repository.
- `--docker-email`: Specifies the email address (optional).

Add secret key parameters in the YML file.

```
containers :
- name : foo
  image : registry - internal . cn - hangzhou . aliyuncs . com /
abc / test : 1 . 0
imagePullSecrets :
- name : regsecret
```

Where:

- `imagePullSecrets` declares that a secret key must be specified when you pull the image.
- `regsecret` must be the same as the preceding secret key name.

- The Docker repository name in `image` must be the same as that in `-- docker -server`.

For more information, see the official documentation [Use private repository](#).

## 1.16.5 Upgrade Helm manually

Log on to the master node of the Kubernetes cluster, see [#unique\\_20](#).

Execute the following command:

```
helm init -- tiller - image registry . cn - hangzhou . aliyuncs . com / acs / tiller : v2 . 9 . 1 -- upgrade
```

The image address can use the VPC domain name of the region corresponding to the image. For example, the image address of a machine in the Hangzhou region can be replaced by `registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.9.1`.

Wait for `tiller` passing through health check. Then you can execute `helm version` to view the upgraded version.



### Note:

Only the Helm server version is upgraded here. To use the Helm client, download the corresponding client binary.

Helm 2.9.1 client download address: <https://github.com/kubernetes/helm/releases/tag/v2.9.1>. Currently, the latest version of Helm supported by Alibaba Cloud is 2.9.1.

After the Helm client and server are both upgraded, you can see the following information by executing the `helm version` command:

```
# helm version
Client : & version . Version { SemVer : " v2 . 9 . 1 ", GitCommit : "
a80231648a 1473929271 764b920a8e 346f6de844 ", GitTreeSta te : "
clean " }
Server : & version . Version { SemVer : " v2 . 9 . 1 ", GitCommit : "
a80231648a 1473929271 764b920a8e 346f6de844 ", GitTreeSta te : "
clean " }
```

## 2 Authorizations

---

### 2.1 Role authorization

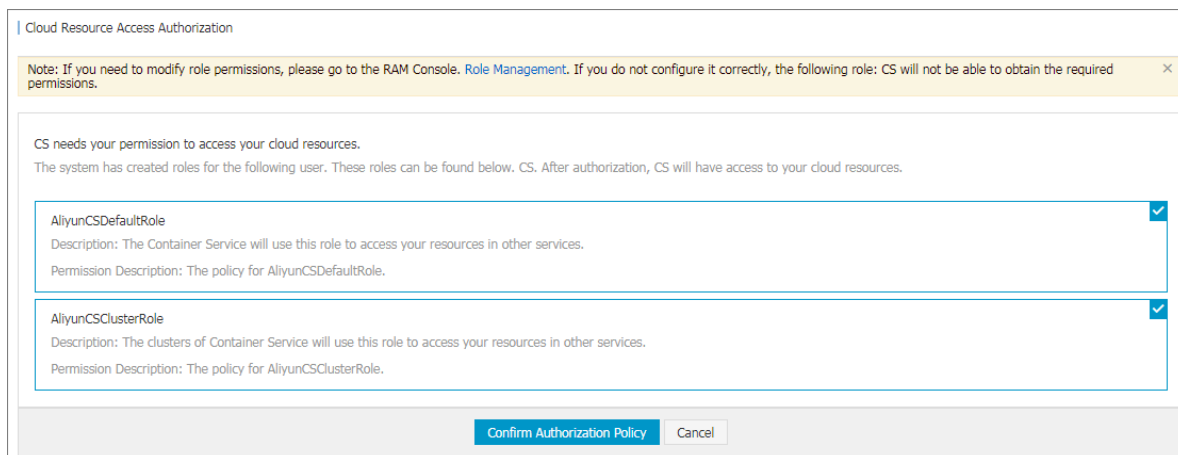
Grant the system default roles AliyunCSDefaultRole and AliyunCSClusterRole to the service account when you activate Container Service. Only after the roles are correctly granted, Container Service can normally call the services such as Elastic Compute Service (ECS), Object Storage Service (OSS), NAS, 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 [#unique\\_103](#).
- 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\\_93](#).

## 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

Note: If you need to modify role permissions, please go to the RAM Console. [Role Management](#). If you do not configure it correctly, the following role: CS will not be able to obtain the required permissions. X

CS needs your permission to access your cloud resources.  
The system has created roles for the following user. These roles can be found below. CS. After authorization, CS will have access to your cloud resources.

AliyunCSDefaultRole Description: The Container Service will use this role to access your resources in other services. Permission Description: The policy for AliyunCSDefaultRole.	✓
AliyunCSClusterRole Description: The clusters of Container Service will use this role to access your resources in other services. Permission Description: The policy for AliyunCSClusterRole.	✓

[Confirm Authorization Policy](#) [Cancel](#)



### 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

Action	Description
ecs:RunInstances	Query ECS instance information.
ecs:RenewInstance	Renew ECS instances.

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.

Action	Description
vpc:UnassociateEipAddress	Do not associate with EIP addresses.
vpc:ReleaseEipAddress	Release EIP addresses.
vpc:CreateRouteEntry	Create router interfaces.
vpc>DeleteRouteEntry	Delete router interfaces.

- SLB-related permissions

Action	Description
slb:Describe*	Query information related to Server Load Balancer.
slb:CreateLoadBalancer	Create Server Load Balancer instances.
slb>DeleteLoadBalancer	Delete Server Load Balancer instances.
slb:RemoveBackendServers	Unbind Server Load Balancer instances.
slb:StartLoadBalancerListener	Start specified listeners.
slb:StopLoadBalancerListener	Stop specified listeners.
slb:CreateLoadBalancerTCPLListener	Create TCP-based listening rules for Server Load Balancer instances.
slb:AddBackendServers	Add backend servers.

### AliyunCSClusterRole permissions

The default role AliyunCSClusterRole contains the following main permissions:

- OSS-related permissions

Action	Description
oss:PutObject	Upload files or folders.
oss:GetObject	Retrieve files or folders.
oss:ListObjects	Query file list information.

- NAS-related permissions

Action	Description
nas:Describe*	Return NAS-related information.
nas:CreateAccessRule	Create permission rules.

· SLB-related permissions

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

## 2.2 Upgrade sub-account policy

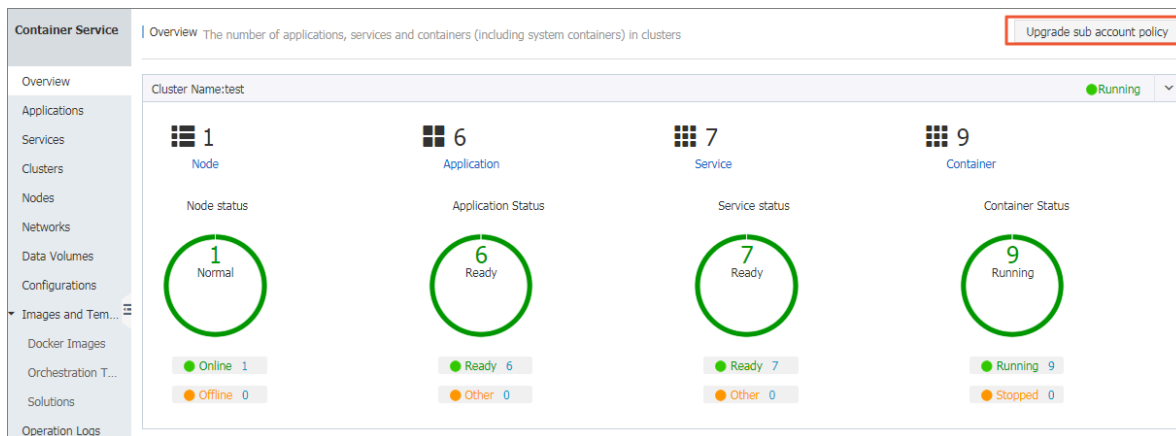
Container Service comprehensively upgrades the security authorization management on January 15 2018, and provides cross-service authorization based on STS to provide you with more secure services. If you have used Container Service before 15 January 2018, the system completes the authorization by default. For more information about the granted permissions, see [#unique\\_105](#) If you used Container Service with a sub-account before, grant the sub-account the permissions to use Container Service again.



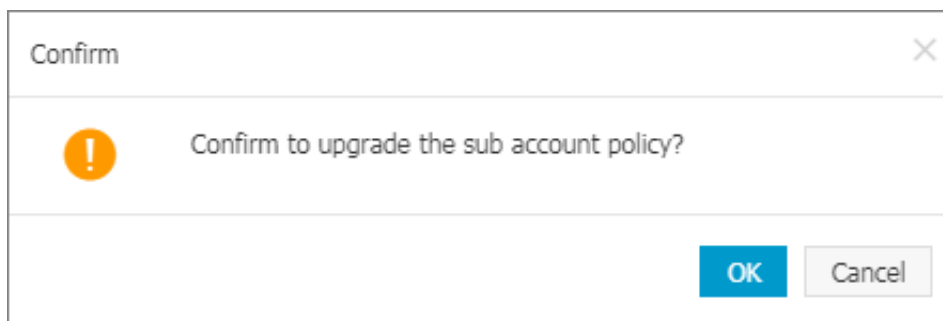
Container Service can automatically upgrade the sub-account policy. With this feature, Container Service automatically grants your sub-accounts the AliyunCSReadOnlyAccess permission. You can also select to manually grant permissions to your sub-accounts in the Resource Access Management (RAM) console.

### 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.

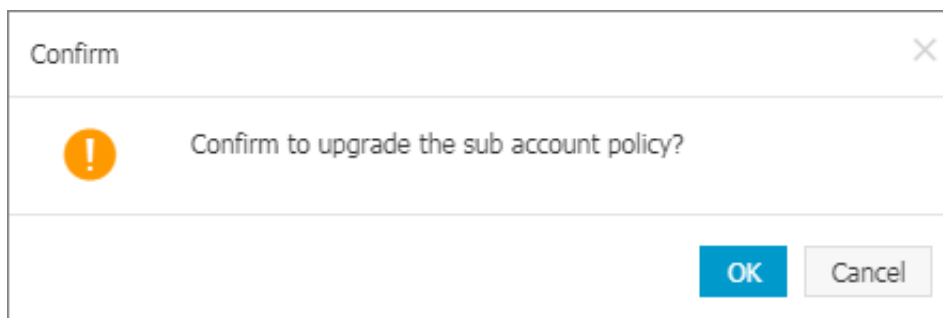


### 3. Click OK in the displayed dialog box.



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

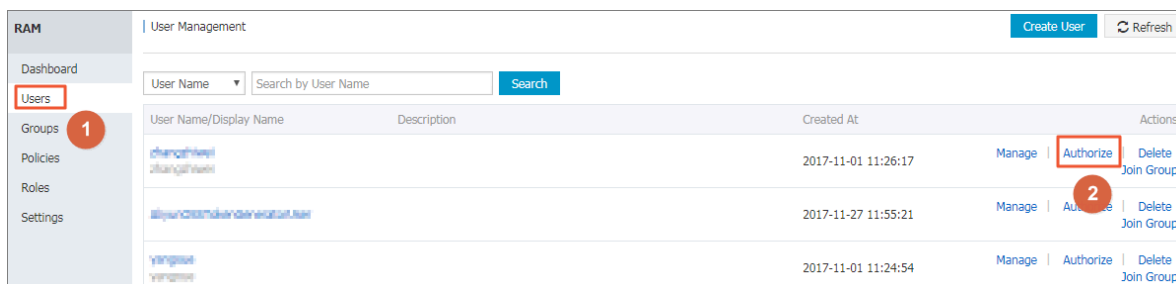
If the upgrade fails, a dialog box listing the sub-accounts that fail to be upgraded appears.



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.

Members added to this group have all the permissions of this group. A member cannot be added to the same group more than once.

Available Authorization Policy Names	Type
CS	
EcsRamRoleDocument...	
AliyunACSResourcesAccess_yangx...	Custom
aliyun container s...	
AliyunCSReadOnlyAccess	System
Provides read-only...	
AliyunCSFullAccess	System
Provides full acce...	

Selected Authorization Policy Name	Type
AdministratorAccess	System
Provides full acce...	
AliyunACSResourcesAccess_xingy...	Custom
aliyun container s...	

OK Close

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 [#unique\\_106](#).

## 2.3 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 [Authorization 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

Step 1: Select an authorization policy | Step 2: Edit permissions and submit. | Policy creation complete.

\* Authorization Policy Name : clusterpolicy  
Names must be 1-128 characters long. They may only contain the letters A-Z, numbers 0-9, and hyphens.

Description :

Policy Content :

```

1 {
2   "Statement": [{
3     "Action": [
4       "cs:Get*",
5       "cs:ScaleCluster",
6       "cs>DeleteCluster"
7     ],
8     "Effect": "Allow",
9     "Resource": [
10      "acs:cs:*:*:cluster/cb2f4c..."
11    ]
12  }],
13  "Version": "1"
14 }

```

[Authorization Policy Format](#)

Previous Create Authorization Policy Cancel

```

" Statement ": [{
  " Action ": [
    " cs : Get *",
    " cs : ScaleClus ter ",
    " cs : DeleteClus ter "
  ]
  " Effect ": " Allow ",

```

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

" Version ": " 1 "
```

#### Wherein:

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



#### Note:

All the Actions support wildcards.

- **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 2-1: Container Service RAM action

Action	Description
CreateCluster	Create clusters.
AttachInstances	Add existing Elastic Compute Service (ECS) instances to clusters.
ScaleCluster	Expand clusters.
GetClusters	View cluster list.
GetClusterById	View cluster details.
ModifyClusterName	Modify cluster names.

Action	Description
DeleteCluster	Delete clusters.
UpgradeClusterAgent	Upgrade cluster Agent.
GetClusterLogs	View cluster operation logs.
GetClusterEndpoint	View cluster access point.
GetClusterCerts	Download cluster certificate.
RevokeClusterCerts	Revoke cluster certificate.
BindSLB	Bind Server Load Balancer instances to clusters.
UnBindSLB	Unbind Server Load Balancer instances from clusters.
ReBindSecurityGroup	Rebind security groups to clusters.
CheckSecurityGroup	Check existing security group rules of clusters.
FixSecurityGroup	Fix cluster security group rules.
ResetClusterNode	Reset cluster nodes.
DeleteClusterNode	Delete cluster nodes.
CreateAutoScale	Create node auto scaling rules.
UpdateAutoScale	Update node auto scaling rules.
DeleteAutoScale	Delete node auto scaling rules.
GetClusterProjects	View applications in clusters.
CreateTriggerHook	Create triggers for applications.
GetTriggerHook	View application trigger list.
RevokeTriggerHook	Delete application triggers.
CreateClusterToken	Create tokens.

## 3 Clusters

---

### 3.1 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 [#unique\\_110](#).

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.

[#unique\\_111](#).

### Manage a cluster

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

- [#unique\\_112](#)
- [#unique\\_113](#)
- [#unique\\_114](#)
- [#unique\\_115](#)
- [#unique\\_116](#)

## 3.2 Cluster lifecycle

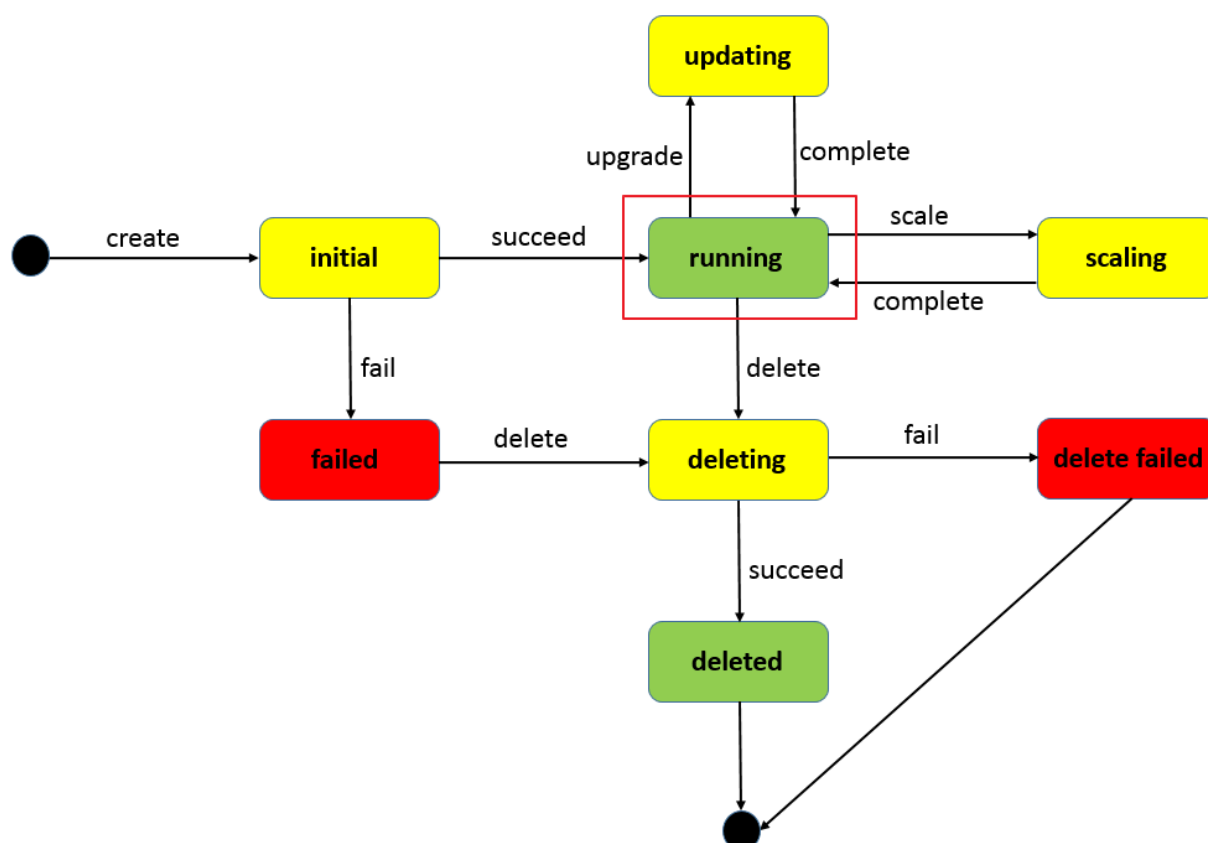
Table 3-1: A complete cluster lifecycle includes the following statuses.

Status	Description
inactive	The successfully created cluster does not contain any node.
initial	The cluster is applying for corresponding cloud resources.
running	The cluster successfully applied for the cloud resources.
updating	The cluster is upgrading the Agent.
scaling	Change the number of cluster nodes.



Status	Description
failed	The cluster application for cloud resources failed.
deleting	The cluster is being deleted.
delete_failed	The cluster failed to be deleted.
deleted (invisible to users)	The cluster is successfully deleted.

Figure 3-1: Cluster status flow



### 3.3 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 [#unique\\_111](#).

## 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

Authorization Policy	Protocol Type	Port Range	Authorization Type	Authorization Object	Description	Priority	Creation time	Operation
Allow	All	-1/-1	Address Field Access	172.22.0.0/16	-	100	2018-05-06 18:36:11	<a href="#">Modify Description</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
Allow	All ICMP	-1/-1	Address Field Access	0.0.0.0/0	-	100	2018-05-06 18:36:10	<a href="#">Modify Description</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
Allow	Custom TCP	80/80	Address Field Access	0.0.0.0/0	-	100	2018-05-06 18:36:09	<a href="#">Modify Description</a>   <a href="#">Clone</a>   <a href="#">Delete</a>
Allow	Custom TCP	443/443	Address Field Access	0.0.0.0/0	-	100	2018-05-06 18:36:09	<a href="#">Modify Description</a>   <a href="#">Clone</a>   <a href="#">Delete</a>

- 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.

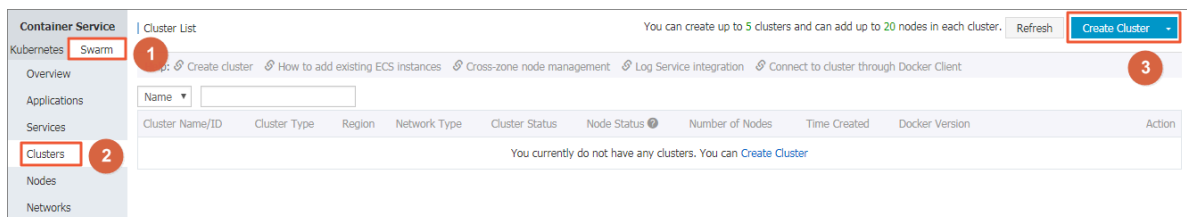
## Limits

- 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:

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)	<b>China East 1 (Hangzhou)</b>	China East 2 (Shanghai)	China South 1 (Shenzhen)	Asia Pacific NE 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 (Zhangjiakou)		

Zone: **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: **VPC**

**vpc-bp-1h4v2vnpqk111111111111** **vsw-bp-1h4v2vnpqk111111111111**

Initial CIDR Block: **172.20.0.0/16** Existing CIDR Block of Container Service ⓘ

Can not be same as the CIDR used by the VPC. Can not be modified after creation.  
Valid value:  
- 192.168.0.0/16  
- 172.19-30.0.0/16  
- 10.0.0.0/16  
System reserved private network address: 172.16/17/18/31.0.0/16  
Maximum number of hosts allowed in the cluster: 256

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/VSwitchId for the cluster to prevent issues such as network conflicts.

## 5. Select whether or not to add nodes.

Add Node :

Add
Do not Add

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 [#unique\\_111](#).

### · Add

#### a. Select the operating system for the node.

Operating System:

CentOS 7.4 64bit
?

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

#### b. Configure the ECS instance specifications.

Add Node

Instance Generation:

Generation III
Generation IV
?

The series III use Intel Broadwell CPU , DDR4 memory, default is I/O optimization instance, high frequency and frequency in the two CPU with a variety of memory ratio, can provide users with better performance and more choices.

Instance Family:

GPU Compute Type gn5
Network Enhanced sn1ne
Network Enhanced sn2ne
Network Enhanced se1ne

I/O Optimized:

IO optimized instance

Instance Type:

8-core, 60GB ( ecs.gn5-....

More instance type, please contact customer service

Instance Quantity:

1
5set(s)
10set(s)
20set(s)
2 set(s)

Each cluster can contain up to 20 ECS instances.

System Disk Type:

Ultra Cloud Disk
SSD Cloud Disk

Data Disk Type:

Ultra Cloud Disk
SSD Cloud Disk

Attach Data Disk:
☐ Attach Data Disk

Login:

Password
Key Pair

\* Logon Password:
?

The password should be 8-30 characters long and contain three types of characters (uppercase/lowercase letters, numbers and special characters).

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:**

- The data disk is mounted to the `/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 [#unique\\_111](#).

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.



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.

EIP:

☐ Do not Configure Public EIP

You must configure the SNAT (refer to the following documents) if a public EIP is not configured. Failure in configuring the SNAT will cause the VPC unable to access the public network. This will affect cluster creation and application deployment.  
Documents for reference: [Configuring SNAT for Linux in a VPC environment to use a server proxy with EIP to access the Internet](#)  
without a public network ECS instance

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

Server Load Balancer: ☒ Automatically Create Server Load Balancer

A public network Server Load Balancer instance is created by default while a cluster is created. The billing method is [Pay-As-You-Go](#).

The 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.

Monitoring Plug-in: Install cloud monitoring plug-in on your ECS.

Installing a cloud monitoring plug-in on the node allows you to view the monitoring information of the created ECS instance in the CloudMonitor console

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.

RDS Whitelist : [Select RDS Instances](#)

b. Select the RDS instances and then click OK.

Add to RDS instance whitelist

<input checked="" type="checkbox"/>	InstanceId	Instance engine version	ZoneId	Network
<input checked="" type="checkbox"/>	rm-1udal1k7o8p2ykv0	MySQL 5.6	cn-hangzhou-b	VPC
<input checked="" type="checkbox"/>	rm-1ud0rifgodbkvmwf9	MySQL 5.7	cn-hangzhou-b	VPC

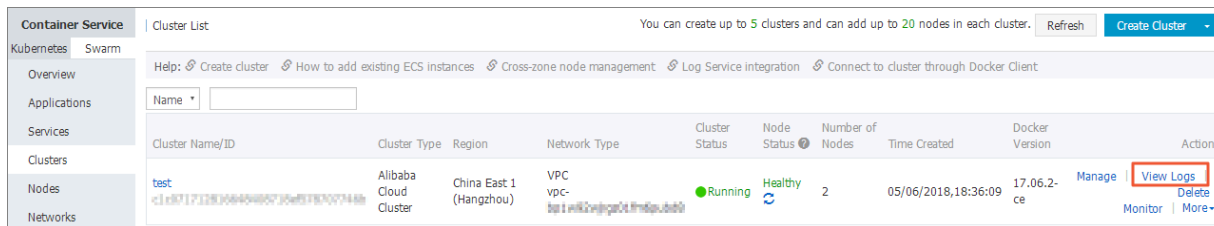
OK Cancel

## 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.



Cluster Name/ID	Cluster Type	Region	Network Type	Cluster Status	Node Status	Number of Nodes	Time Created	Docker Version	Action
test c1e8717128100404000730e8178707746b	Alibaba Cloud Cluster	China East 1 (Hangzhou)	VPC vpc- hnp1-e8178707746b	Running	Healthy	2	05/06/2018, 18:36:09	17.06.2-ce	Manage View Logs Delete Monitor More

You can create applications in the created cluster. For more information, see [#unique\\_119](#).

## References

If the cluster fails to be created, see [#unique\\_120](#) for troubleshooting.

## 3.4 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]. 1.0/24

## 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 [#unique\\_111](#).

## 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 [../SP\\_2/DNECS19100341/EN-US\\_TP\\_9548.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 [../SP\\_2/DNA0011894323/EN-US\\_TP\\_9557.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 [../SP\\_110/DNnat1841475/EN-US\\_TP\\_13979.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\\_122](#)

- 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.

## 3.5 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 [#unique\\_110](#).

## 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:

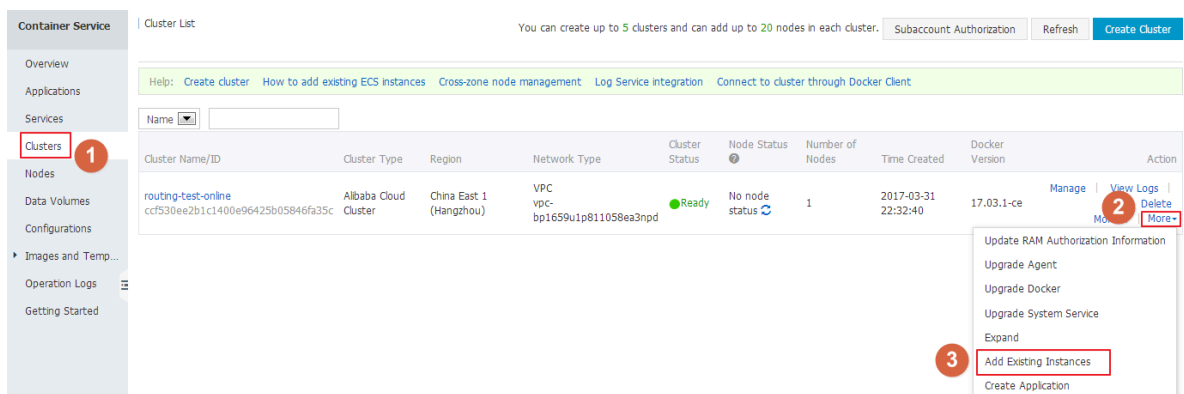
**Ubuntu:** `apt - get remove - y docker - engine , rm - fr / etc / docker / / var / lib / docker / etc / default / docker`

**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\\_19](#).

- 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.

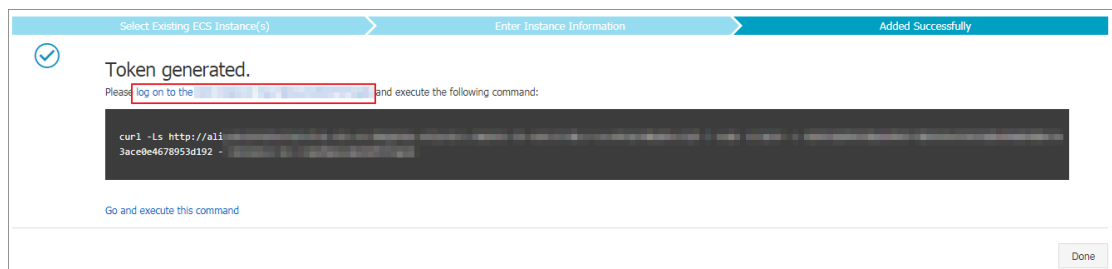
- Manually add the ECS instance by running scripts on the ECS instance.

- a. Select Manually Add. Select an ECS instance, and then click Next Step.

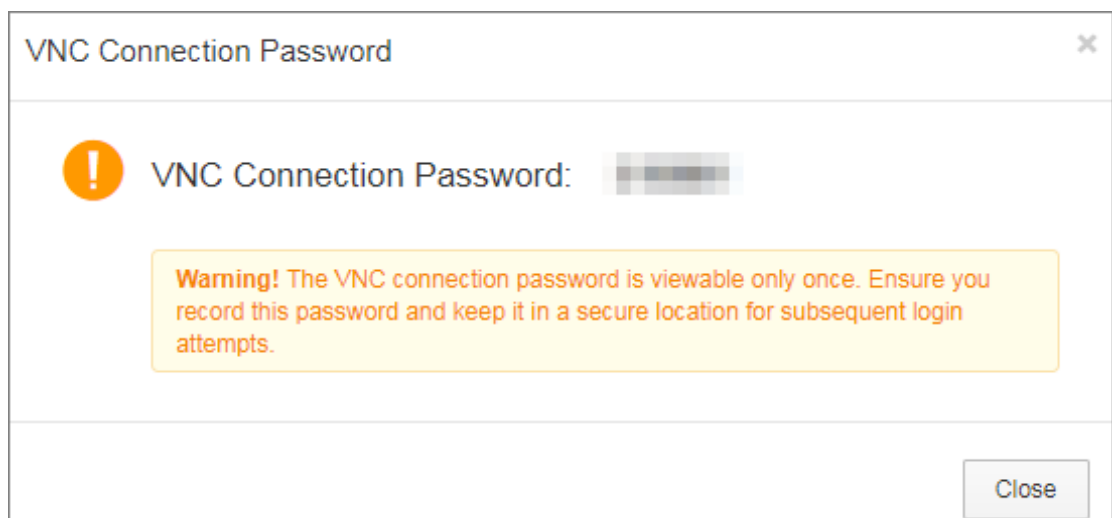
You can only add one ECS instance at a time.

- b. Confirm the instance information and click Next Step.

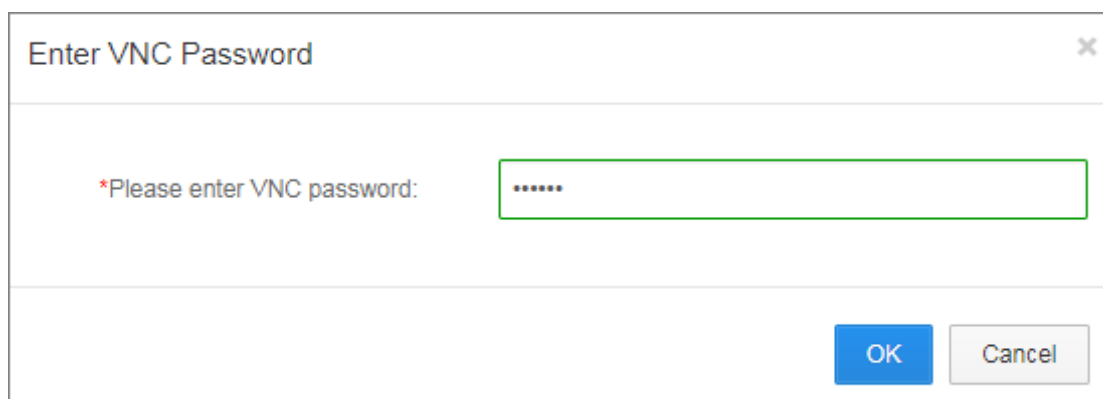
- c. The scripts unique to this ECS instance are displayed. Click log on to the ECS instance xxxxxx.



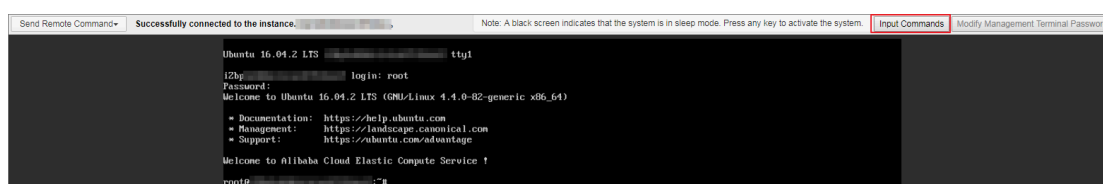
- d. The VNC connection password is displayed in the dialog box. Copy the password and click Close.



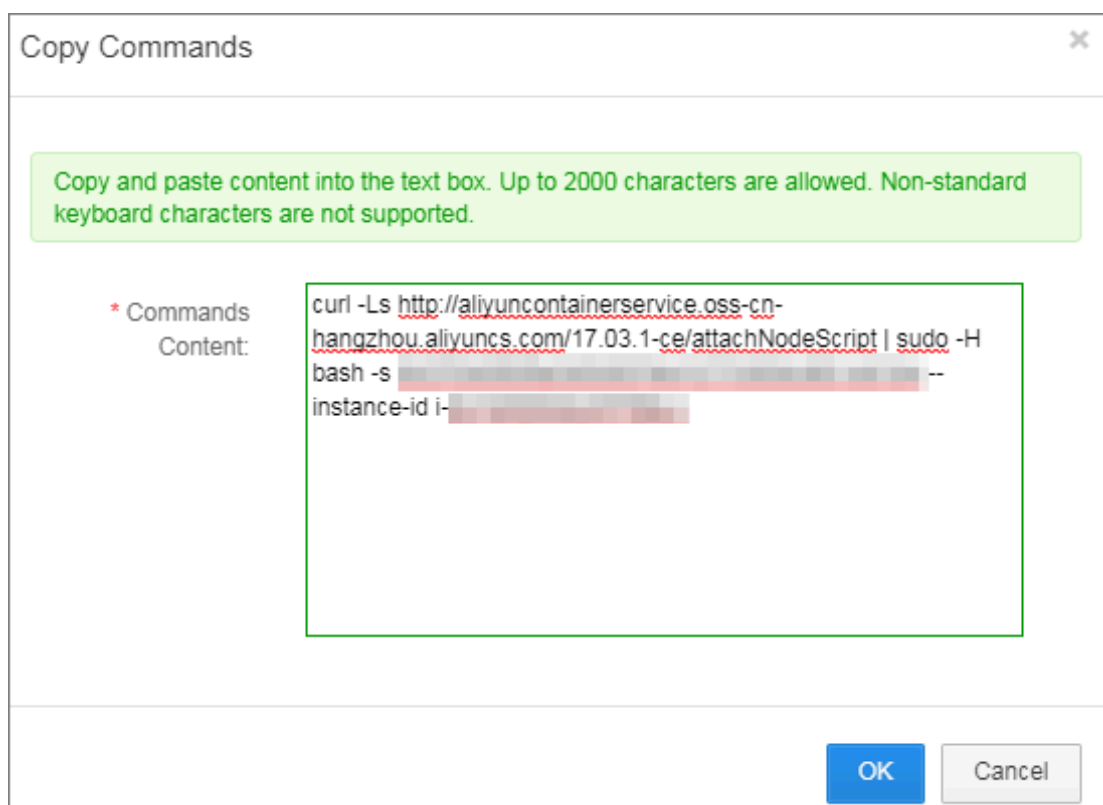
- e. In the dialog box, enter the VNC connection password and click OK.

A dialog box titled "Enter VNC Password" with a close button (X) in the top right corner. The main area contains the text "\*Please enter VNC password:" followed by a password input field with six dots. At the bottom right, there are two buttons: "OK" (blue) and "Cancel" (gray).

- f. Enter the logon account (root) and password of the ECS instance, and press Enter to log on to the ECS instance.



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

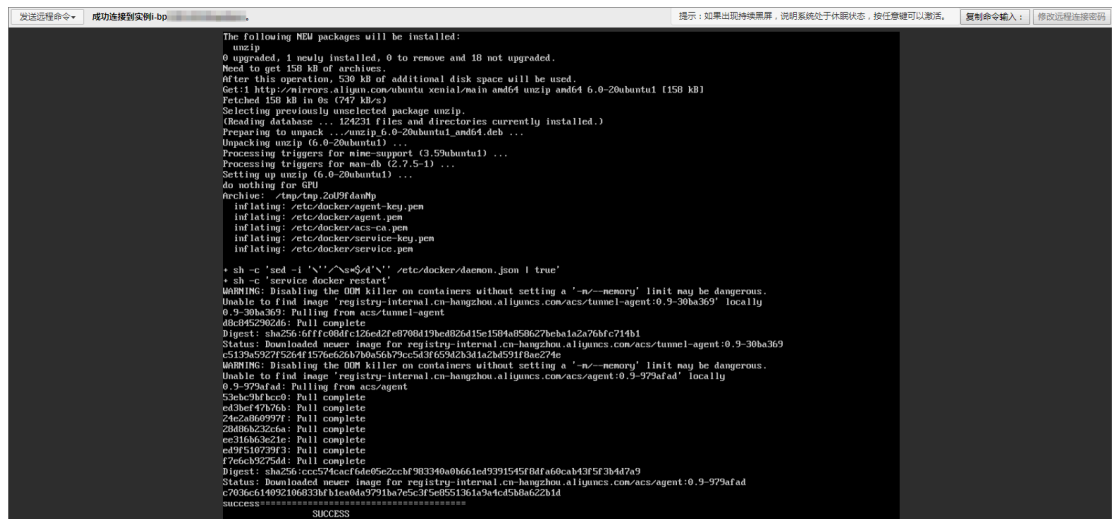
A dialog box titled "Copy Commands" with a close button (X) in the top right corner. The main area contains a green instruction box: "Copy and paste content into the text box. Up to 2000 characters are allowed. Non-standard keyboard characters are not supported." Below this, there is a text area with the label "\* Commands Content:". The text area contains the following commands: 

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

 At the bottom right, there are two buttons: "OK" (blue) and "Cancel" (gray).

The system runs the scripts. Wait until the scripts are successfully run. A success message is displayed. The ECS instance is successfully added.





## Related operation

**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.**

Modify Management Terminal Password

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

\*Please enter a new password:

\*\*\*\*\*

Password character limit is 6 characters. Only uppercase letters, lowercase letters, and numbers are supported.

\*Confirm the new password:

\*\*\*\*\*

OK

Cancel

### 3.6 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.



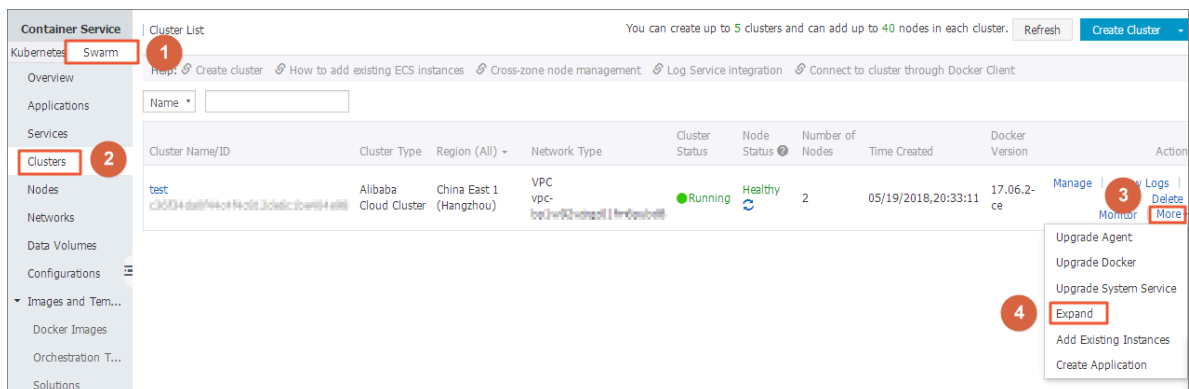
#### Note:

- 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

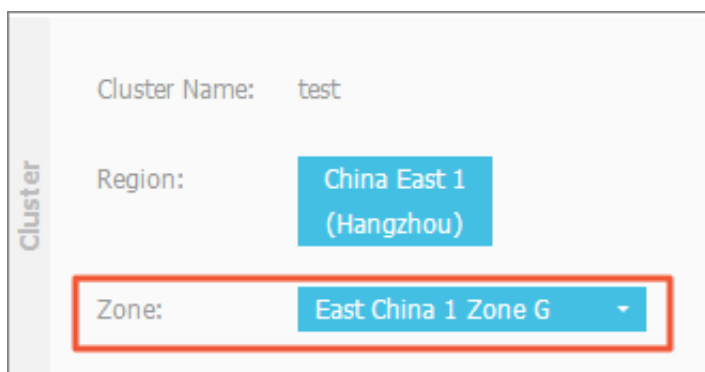
#### Procedure

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

.

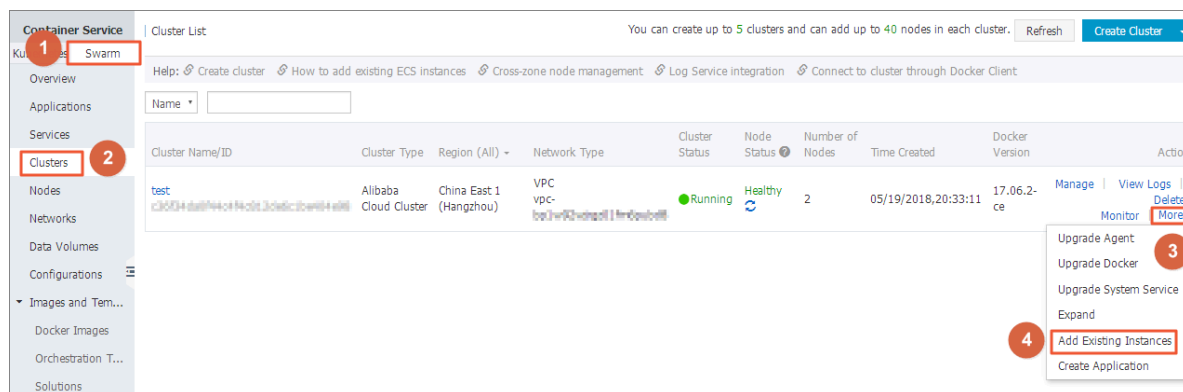
### Add nodes of different zones by adding existing ECS instances

#### 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.

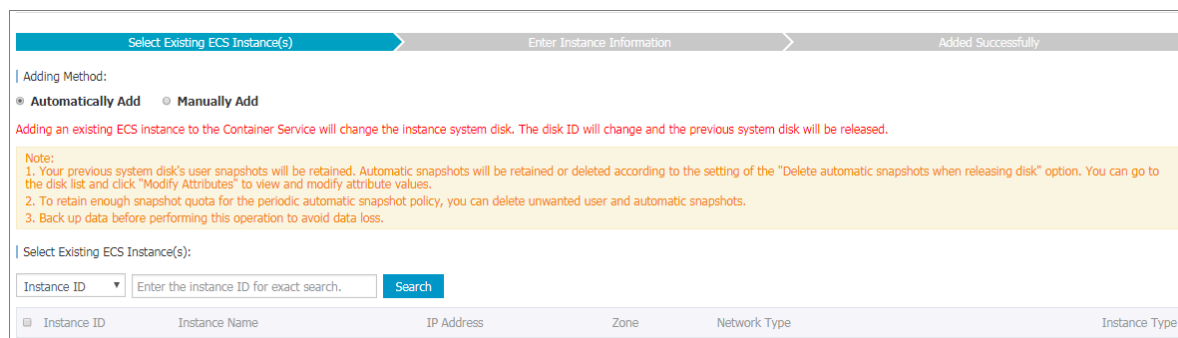
#### Procedure

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 [#unique\\_111](#).



5. Repeat the preceding steps to add ECS instances of different zones to the cluster.

## 3.7 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.

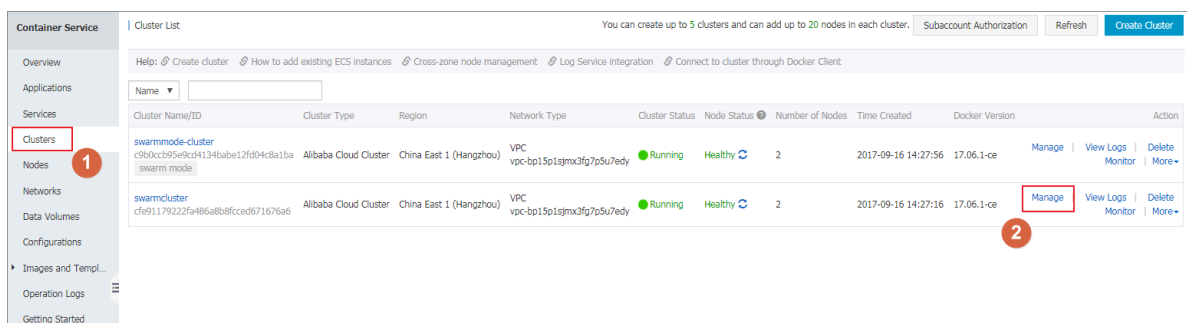
### Prerequisites

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.

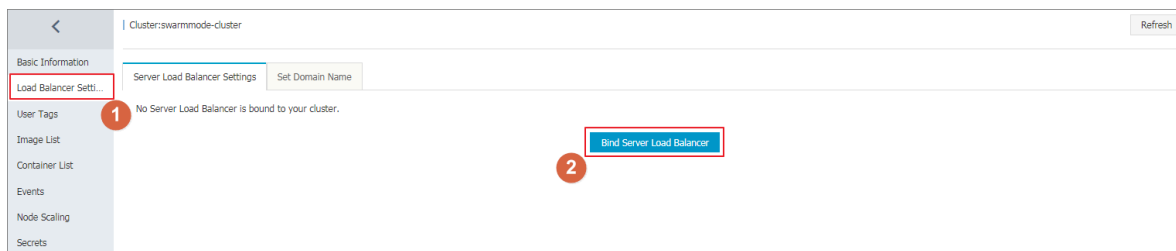
For how to create a Server Load Balancer instance, see [Create a Server Load Balancer instance](#).

### 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.



- Click **Load Balancer Settings** in the left-side navigation pane > and then click **Bind Server Load Balancer**.



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



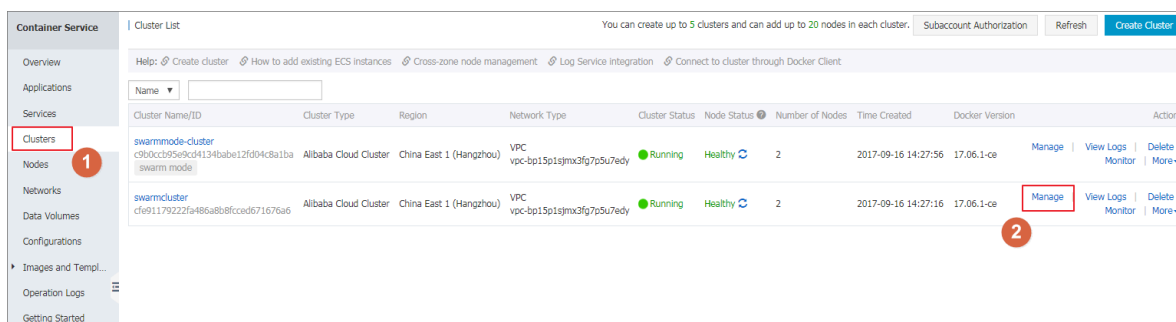
#### Note:

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.

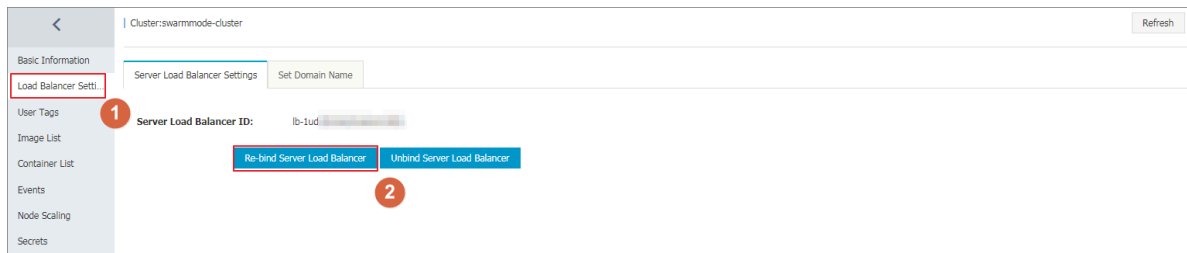
### Rebind a Server Load Balancer instance

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

- Log on to the [Container Service console](#).
- Click **Clusters** in the left-side navigation pane.
- Click **Manage** at the right of the cluster that you want to re-bind a Server Load Balancer instance. The cluster details page appears.



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

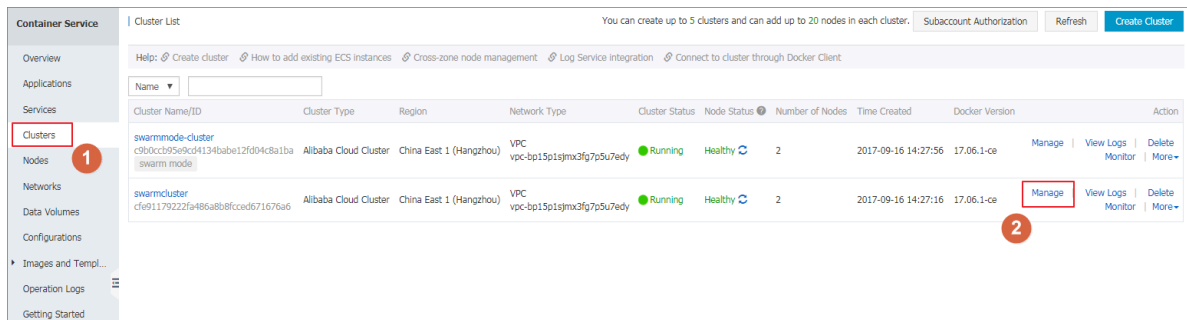


- Select the Server Load Balancer instance that you want to bind to the cluster from the Server Load Balancer ID list and click OK.

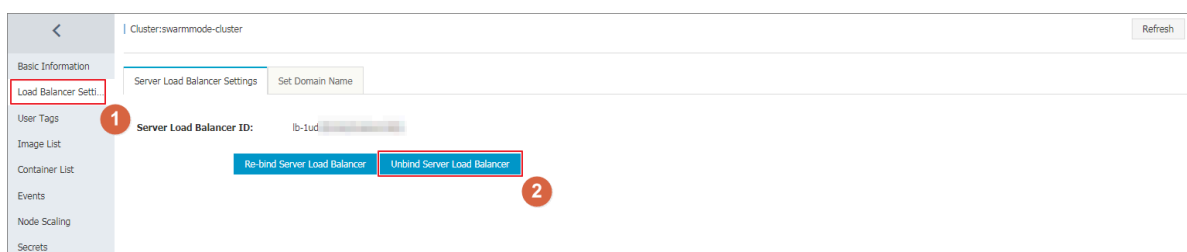
### Unbind a Server Load Balancer instance

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

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



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



## 3.8 Set the root domain name of a cluster

### Context

When you [#unique\\_127](#) and 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.

Name/ID	Region	Network type	Status	Node status	No. of nodes	Time Created	Docker version	Action
routing-test-online cc0ba07a50146f405628e20c74db0363	China North 2(Beijing)	Classic network	Ready	Healthy	1	2016-10-14 15:43:01	1.11.2.1	Manage   Logs   Monitor   Delete
l013he ca9b682b2dc4241449c3d02293a0f419f	China East 1(Hangzhou)	Classic network	Failed	No node status	2	2016-10-13 17:14:58	1.11.2.1	Manage   Logs   Monitor   Delete
l013rg c05a9e674a00e4247a099a81c551896de	China North 2(Beijing)	Classic network	Failed	No node status	2	2016-10-13 14:44:26	1.11.2.1	Manage   Logs   Monitor   Delete

- 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 [#unique\\_128](#).

<

Basic Information

Load Balancer Se...

User Tags

Image List

Container List

Events

Node Scaling

Cluster:test

Server Load Balancer Settings

Set Domain Name

Server Load Balancer ID: lb-1a2b3c4d5e6f7g8h9i0j

Re-bind Server Load Balancer

Unbind Server Load Balancer

**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.

Server Load Balancer Settings

Set Domain Name

Domain Name:

Set

Cancel

- 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.

acs-slb-c9b105e434... [Return to Server Load Balancer List](#) [Restrictions and Notes](#)

Basic Information	
Server Load Balancer ID: <a href="#">b-1u078h9n7uafj1com</a>	Status: <span style="color: green;">Running</span>
Server Load Balancer Name: <a href="#">slb-c9b105e434...</a>	Region: China East 1 (Hangzhou)
Instance IP Type: Public IP	Zone: cn-hangzhou-f(Master)/cn-hangzhou-e(Slave)
Network Type: Classic Network	

Billing Information		<a href="#">Billing Details</a>	<a href="#">Release</a>
Billing Method: Pay by Traffic	Created At: 2017-11-27 09:15:59		
Instance IP Address: <span style="border: 1px solid red; padding: 2px;">47.97.99.153(Public IP)</span>	Automatic Release Time: -		

- 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>.

## 3.9 Download cluster certificate

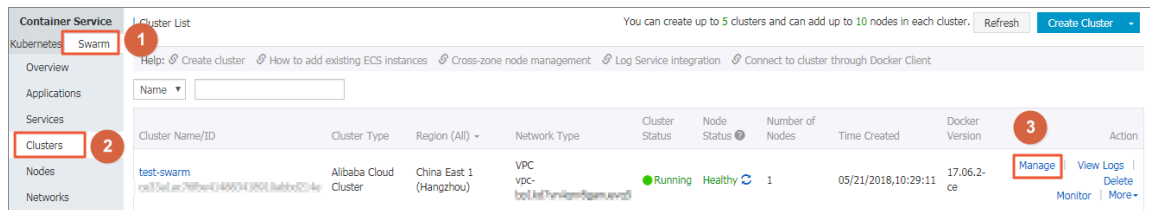
### Context

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\\_130](#).

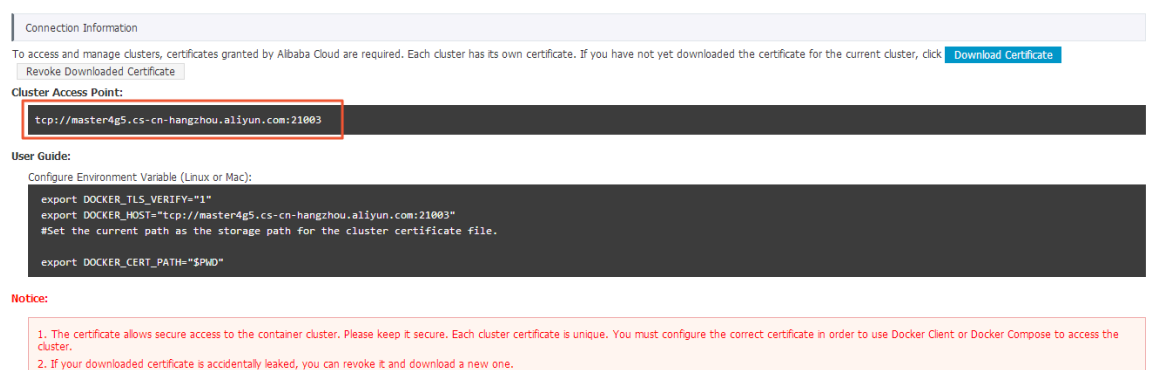
## Procedure

### 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 `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` .

## 3.10 Expand a cluster

### Prerequisites

A cluster can contain up to 20 nodes.

### Context

You can expand your cluster according to your business needs.

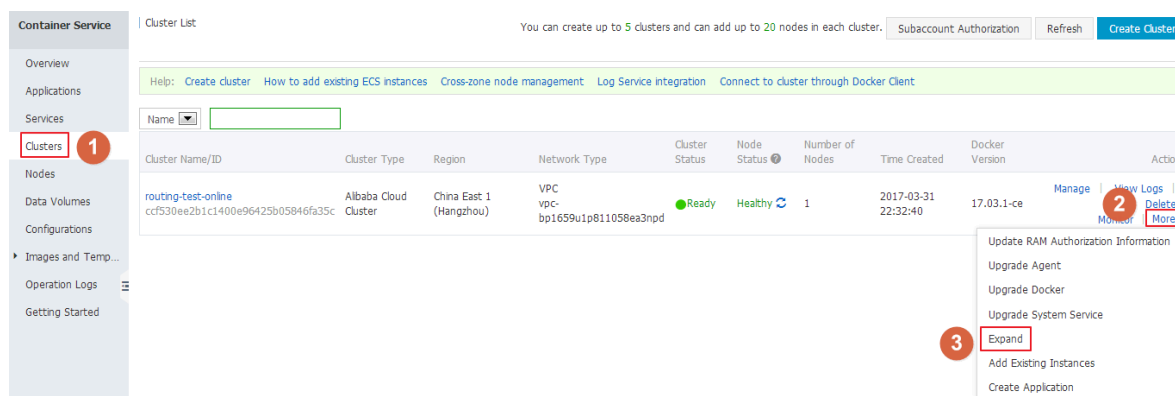


#### Note:

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.

## 3.11 Migrate a cluster

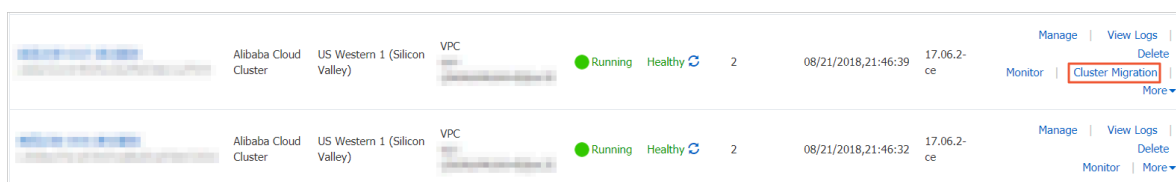
For a Swarm cluster created earlier, you can guarantee the performance and stability of the cluster by migrating the cluster.

### 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.



	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	<span style="color: green;">●</span> Running	<span style="color: blue;">↻</span> Healthy	2	08/21/2018,21:46:39	17.06.2-ce	<a href="#">Manage</a>   <a href="#">View Logs</a>   <a href="#">Delete</a> <a href="#">Monitor</a>   <b><a href="#">Cluster Migration</a></b>   <a href="#">More</a>
	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	<span style="color: green;">●</span> Running	<span style="color: blue;">↻</span> Healthy	2	08/21/2018,21:46:32	17.06.2-ce	<a href="#">Manage</a>   <a href="#">View Logs</a>   <a href="#">Delete</a> <a href="#">Monitor</a>   <a href="#">More</a>

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.

	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	Migrating	Healthy	2	08/21/2018,21:46:47	17.06.2-ce	Manage   View Logs   Delete   Monitor   More▼
	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	Running	Healthy	2	08/21/2018,21:46:39	17.06.2-ce	Manage   View Logs   Delete   Monitor   More▼

## Result

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.

	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	Running	Healthy	2	08/21/2018,21:46:47	17.06.2-ce	Manage   View Logs   Delete   Monitor   More▼
	Alibaba Cloud Cluster	US Western 1 (Silicon Valley)	VPC	Running	Healthy	2	08/21/2018,21:46:39	17.06.2-ce	Manage   View Logs   Delete   Monitor   More▼

## 3.12 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:

The search is case insensitive.

Cluster List You can create up to 5 clusters and can add up to 20 nodes in each cluster. Subaccount Authorization Refresh Create Cluster

Help: [Create cluster](#) [How to add existing ECS instances](#) [Cross-zone node management](#) [Log Service integration](#) [Connect to cluster through Docker Client](#)

Name

Cluster Name/ID	Cluster Type	Region	Network Type	Cluster Status	Node Status	Number of Nodes	Time Created	Docker Version	Action
<a href="#">routing-test-online</a> ccf530ee2b1c1400e96425b05846fa35c	Alibaba Cloud Cluster	China East 1 (Hangzhou)	VPC vpc-bp1659u1p811058ea3npd	Ready	Healthy	1	2017-03-31 22:32:40	17.03.1-ce	<a href="#">Manage</a>   <a href="#">View Logs</a>   <a href="#">Delete</a>   <a href="#">Monitor</a>   <a href="#">More</a>

## 3.13 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.

Container Service | Cluster List You can create up to 5 clusters and can add up to 20 nodes in each cluster. Subaccount Authorization Refresh Create Cluster

Help: [Create cluster](#) [How to add existing ECS instances](#) [Cross-zone node management](#) [Log Service integration](#) [Connect to cluster through Docker Client](#)

Name

Cluster Name/ID	Cluster Type	Region	Network Type	Cluster Status	Node Status	Number of Nodes	Time Created	Docker Version	Action
<a href="#">routing-test-online</a> ccf530ee2b1c1400e96425b05846fa35c	Alibaba Cloud Cluster	China East 1 (Hangzhou)	VPC vpc-bp1659u1p811058ea3npd	Ready	Healthy	1	2017-03-31 22:32:40	17.03.1-ce	<a href="#">Manage</a>   <a href="#">Logs</a>   <a href="#">Delete</a>   <a href="#">Monitor</a>   <a href="#">More</a>

4. In the displayed window, select whether or not to keep the Server Load Balancer instance and click OK.

## 3.14 Clean up a cluster disk

### Context

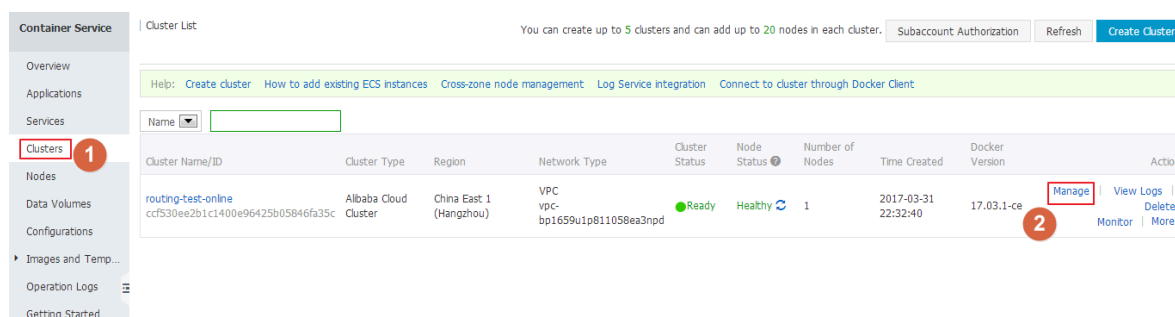
Cleaning up disk clears the dirty data on each server in your cluster. Dirty data is limited to:

- Docker images downloaded locally but not used.

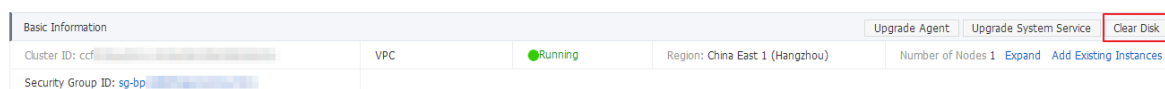
- Volume directory once attached to a container but not cleaned up after the destruction of the container.

## 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.



## 3.15 Log on to image repository

### Prerequisites

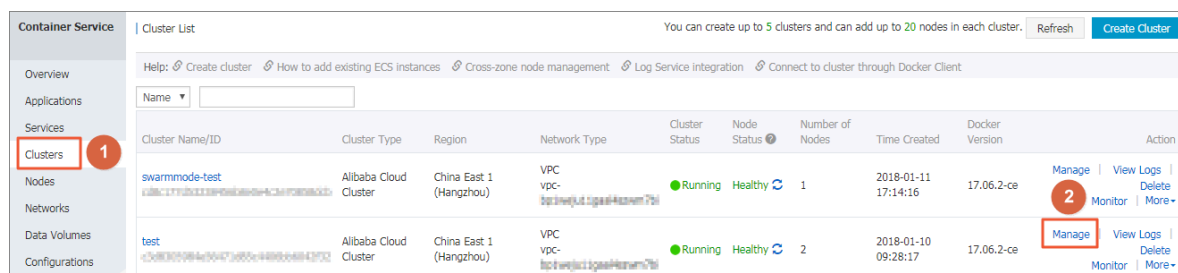
- Prepare an available image repository. Use the Docker Hub official service in this example, which requires you to register a Docker ID and build an available repository in it.
- Configure the independent logon password for the repository. In this example, log on to the [Container Registry console](#) 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

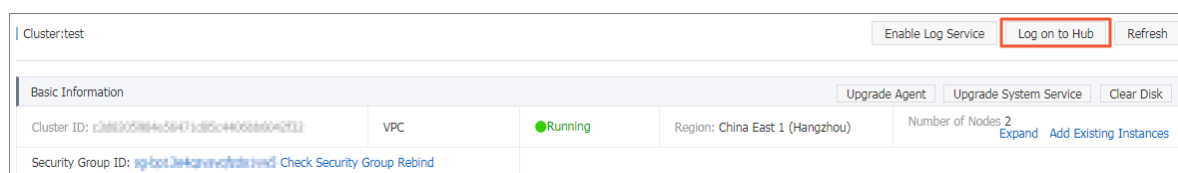
You can log on to the image repository in a cluster to provide the related cluster logon information, which facilitates you to manage clusters by using cluster management tools.

### Procedure

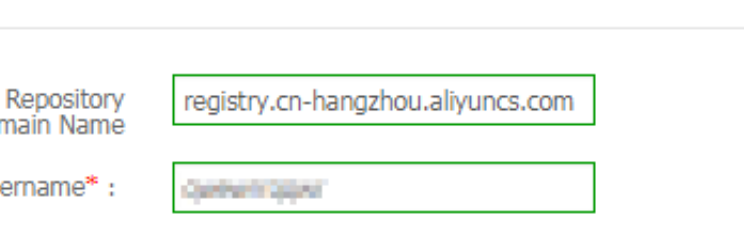
1. Log on to the [Container Service console](#).
2. Click Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster you want to configure.



- 4. Click Log on to Hub.**



- 5. Configure the parameters in the displayed dialog box.**



Log on to Hub

Repository Domain Name: registry.cn-hangzhou.aliyuncs.com

Username\*: [blurred]

Password\*: [masked]

The user's account password will be encrypted

Email: [empty]

OK Cancel

- **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.

## 3.16 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.

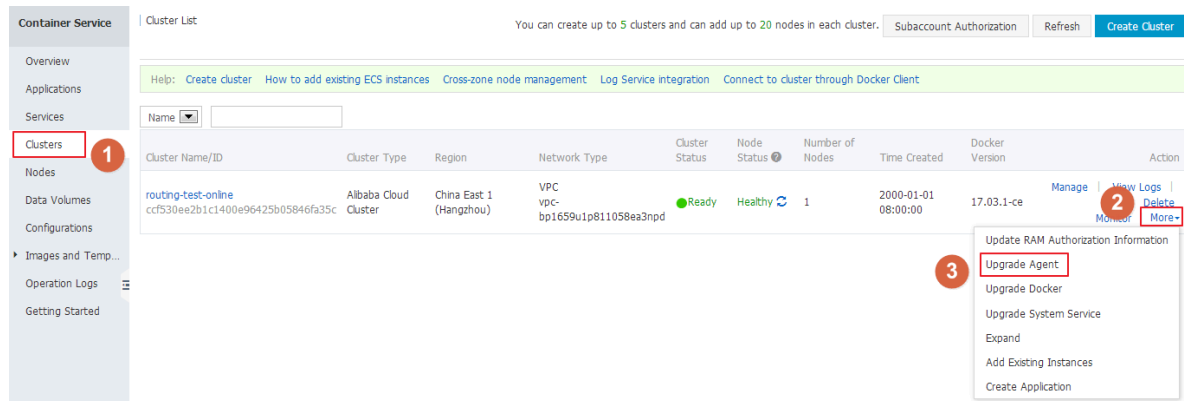
The Agent of Container Service, which is installed on each server in the cluster, receives commands issued by the Container Service control system.

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.

## 3.17 Upgrade Docker daemon

### Context

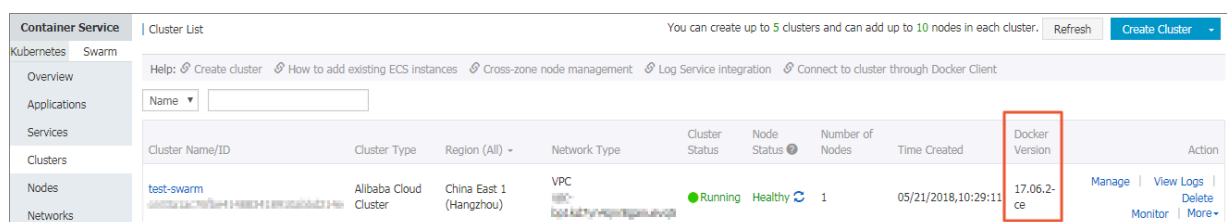
Standard Docker daemon is installed on each server in the cluster to manage containers.



#### Note:

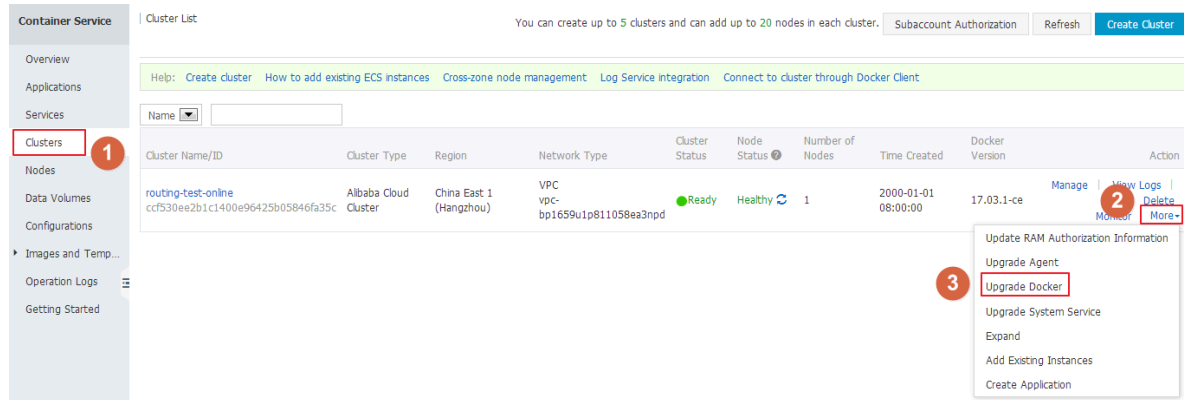
- The cluster Docker daemon upgrade requires that the machine is able to access the Internet to download necessary software packages.
- The cluster Docker daemon upgrade may fail. To guarantee your data security, we recommend that you back up snapshots before upgrading Docker daemon.
- During the cluster 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.

What's next

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.

## 3.18 Upgrade system services

### Context

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

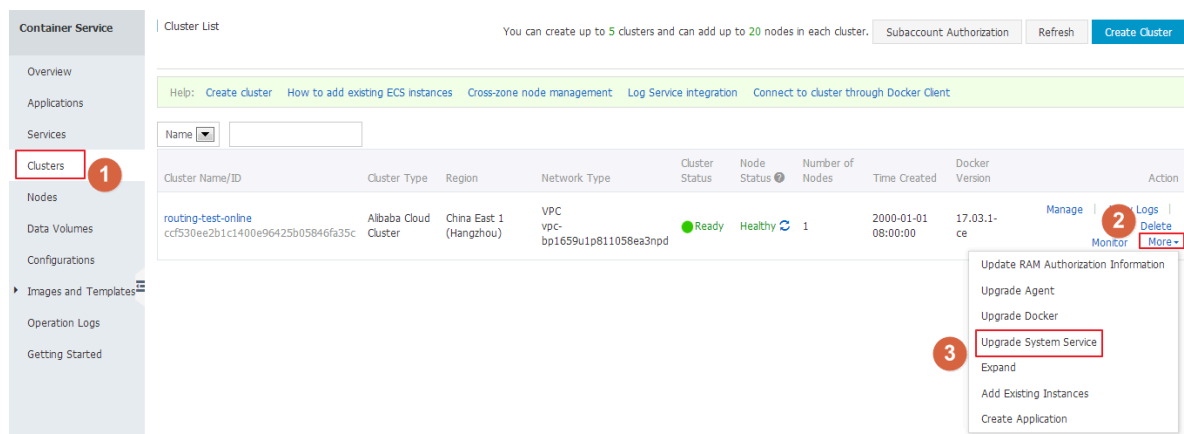


#### Note:

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.

## 4 Nodes

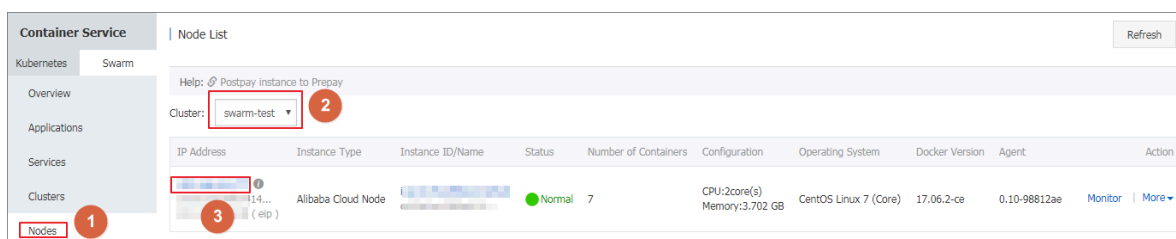
### 4.1 View containers running on a node

#### Context

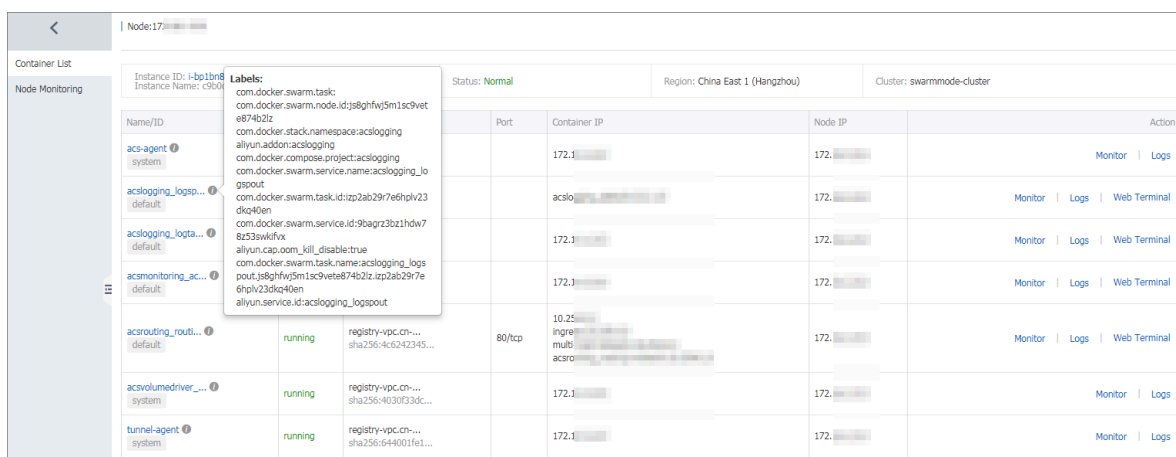
You can view containers running on a node on the Node List page.

#### 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.

## 4.2 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 [#unique\\_110](#).
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 [#unique\\_143](#). 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.



## 5 Images and templates

### 5.1 Update an orchestration template

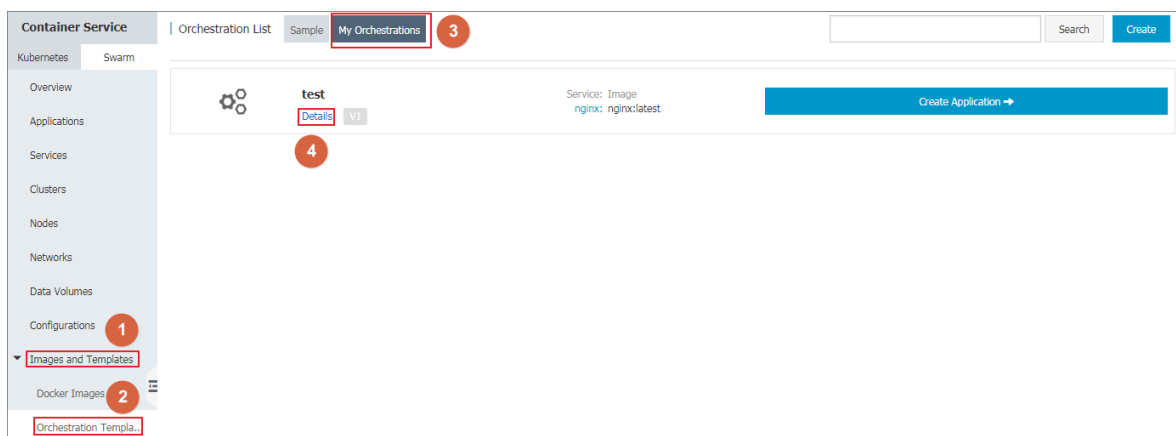
#### Context

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\\_146](#).

#### 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.

Orchestration:test

Back to Orchestration List

Cancel

Save

Name:

test

The name should be 1-64 characters long, and can contain numbers, English letters, Chinese characters and hyphens.

Description:

Template:

```
1 newServiceBlock1510281361729:
2   image: 'nginx:latest'
3   restart: always
4   expose:
5     - 80/tcp
6   labels:
7     aliyun.scale: '1'
```

Service(s) Contained

Service Name:  
newServiceBlock1510281361...  
Image: nginx:latest  
Edit Delete

Add Service

## 6. Click Save in the upper-right corner to save the modifications.

## 6 Service orchestrations

---

### 6.1 routing

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

Format:

```
aliyun . routing . port_ $ container_ port : [ http :// ]$ domain |$  
domain_pre fix [ :$ context_pa th ]
```

Field description:

- `$ container_ port` : container port. Note: This is not the host port.
- `$ domain` : domain name. Enter a domain name.
- `$ domain_pre fix` : 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 >.alicontain er . com`.
- `$ context_pa th` : 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 `alicontain er . 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 > .`

`alicontain er . com` provided by Container Service and your own domain name `http :// wp . sample . com / context` to port 80 of the Web service.

```
web :
  image : wordpress : 4 . 2
  links :
    - db : mysql
  labels :
    aliyun . routing . port_80 : wordpress ; http :// wp . sample .
com / context
db :
  image : mysql
  environmen t :
    - MYSQL_ROOT _PASSWORD = password
```

The internal domain name that you finally get is `wordpress . cd3dfe2690`

`56e4543acb ec5e19b01c 074 . cn - beijing . alicontain er . com .`

After starting the Web service, you can access the corresponding Web services by

using the URL: `http :// wordpress . cd3dfe2690 56e4543acb ec5e19b01c`

`074 . cn - beijing . alicontain er . 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_ $ containr_ 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_st icky : true
```

- Disable session persistence

```
aliyun . routing . session_st icky : false
```

Example of a template orchestration file:

```
web :
  image : wordpress : 4 . 2
  links :
    - db : mysql
  labels :
    aliyun . routing . port_80 : wordpress ; http :// wp . sample .
com / context
    aliyun . routing . session_st icky : true
db :
  image : mysql
  environmen t :
    - MYSQL_ROOT _PASSWORD = password
```

# 7 Applications

## 7.1 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.

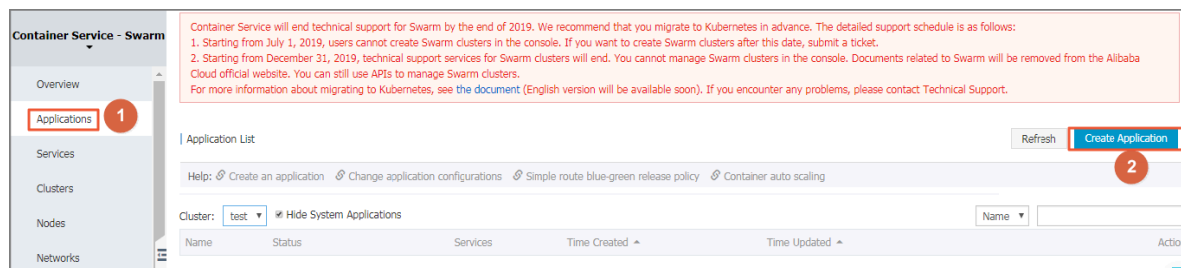


#### Note:

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\\_151](#).
- **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.

Create Application [Back to Application List](#)

Help: [Restrict container resources](#) [High availability scheduling](#) [Create a Nginx webserver from an image](#) [Create WordPress by using an application template](#) [Orchestration template description](#) [Label description](#)

Basic Information Configuration Done

Name:   
The name should be 1-64 characters long, and can contain numbers, English letters and hyphens, but cannot start with a hyphen.

Version:

Cluster:

Update:

Description:

☐ Pull Docker Image

Create with Image Create with Orchestration Template

## 5. Click Create with Image.

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

### a) In the General section:

General

Image Name:  [Select image](#)

Image Version:

Scale:

Network Mode:

Restart: ☒ Always

- 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](#).

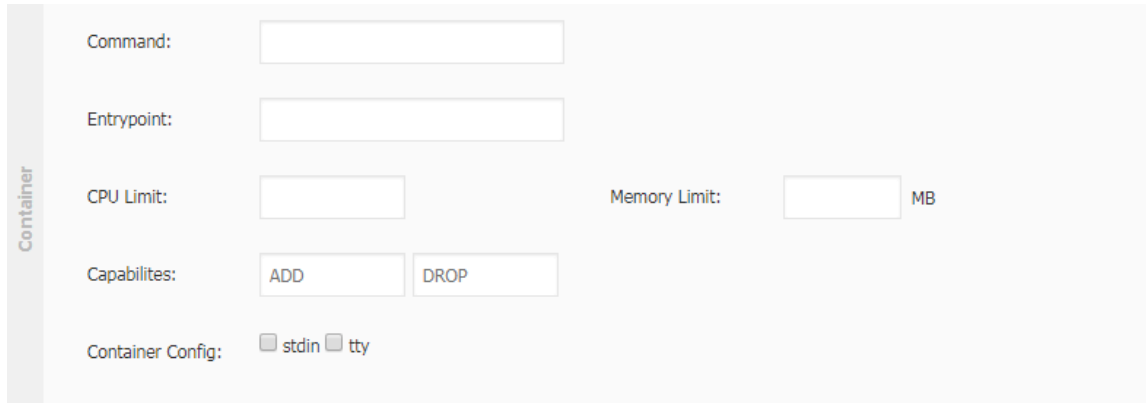
- Set the Restart field.

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:**



The screenshot shows a 'Container' configuration panel. It contains the following elements:

- Command:** A text input field.
- Entrypoint:** A text input field.
- CPU Limit:** A text input field.
- Memory Limit:** A text input field followed by a unit selector dropdown menu currently showing 'MB'.
- Capabilities:** Two buttons labeled 'ADD' and 'DROP'.
- Container Config:** Two checkboxes labeled 'stdin' and 'tty'.

- 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.

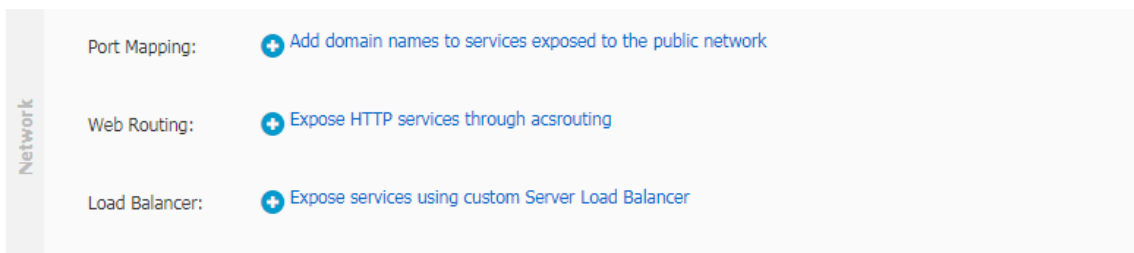
Set the resource limit for the CPU and memory to be used by the container. For more information, see [#unique\\_152](#).

- 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\\_153](#).



#### 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\\_154](#).

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.

The screenshot shows a configuration panel for 'Data Volume'. At the top, there is a link '+ Use third-party data volumes'. Below this is a table with three columns: 'Host Path or Data Volume Name', 'Container Path', and 'Permission'. The first row contains example values: 'e.g. /path or name', 'e.g. /path', and 'RW'. To the right of the 'RW' value is a red minus sign. Below the table is a field labeled 'volumes\_from:' with an empty input box.

Host Path or Data Volume Name	Container Path	Permission
e.g. /path or name	e.g. /path	RW

volumes\_from:

- 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\\_155](#).

g) In the Deploy section:

- Set whether to enable Smooth Upgrade for containers.

For more information, see [#unique\\_156](#).

- 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.

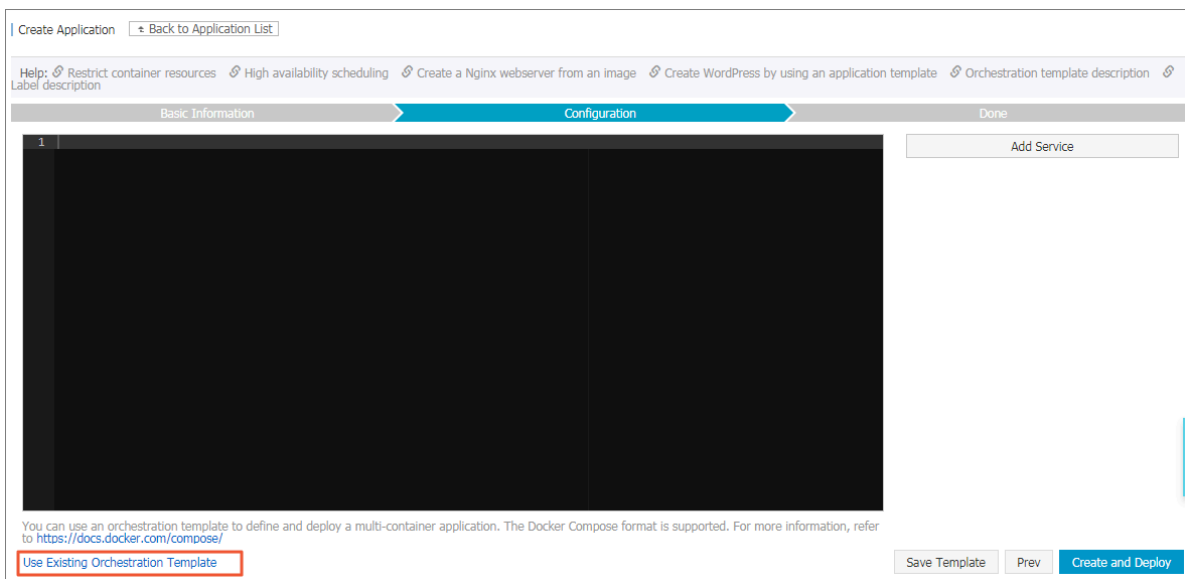
If this parameter is not configured, Container Service deploys the containers in one zone by default. For more information, see [#unique\\_157](#).

- Set whether or enable the container Auto Scaling.

For more information, see [#unique\\_158](#).

h) Click Create at the right of the page after completing the settings.

## 6. Click Create with Orchestration Template.

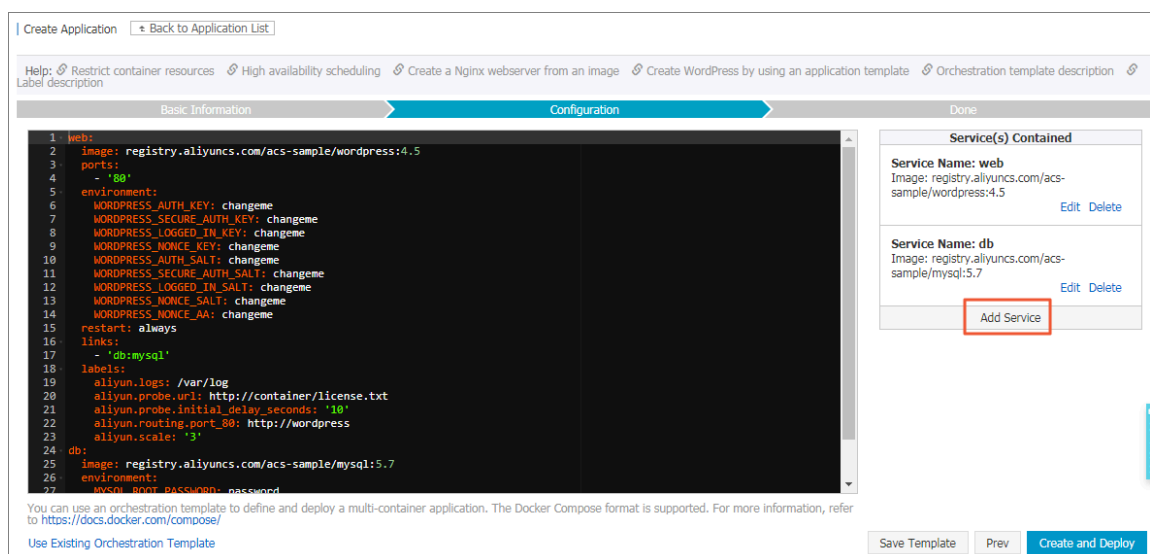


- a) Click Use Existing Orchestration Template or write a new template by yourself.

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.



Click Add Service to add another service to this orchestration template. Select the image and configure the parameters. Then, click OK.

Create Service

Image Name: Private registry entr [Select image](#)

Image Version: [Select image version](#)

Scale: 1

Port Mapping:	Host Port	Container Port	Publish	Protocol	Action
	Host Port	Container Port	<input checked="" type="checkbox"/>		Add

Environment:	Variable Name	Variable Value	Action
	Name	Value	Add

Data Volume:	Host Path or Data Volume Name	Container Path	Permission	Action
	Host Path or Data Volume I	Container Path	Read/Writ	Add

Web Routing:	Container Port	Domain Name:	Action
	Container Port	Domain name: For example: http://[domain nam	Add

Note: All domain names for a port must be entered in one entry.

Restart: ☒

More Settings

d) Click Create and Deploy after completing the settings.

## 7.2 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.

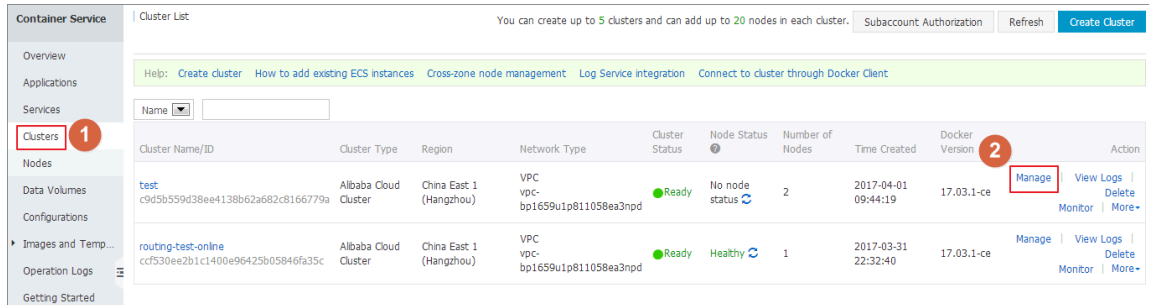
- After you use a user tag to deploy an application, deleting the user tag does not affect the deployed application, but will affect the next deployment of the application. Proceed with caution when deleting user tags.



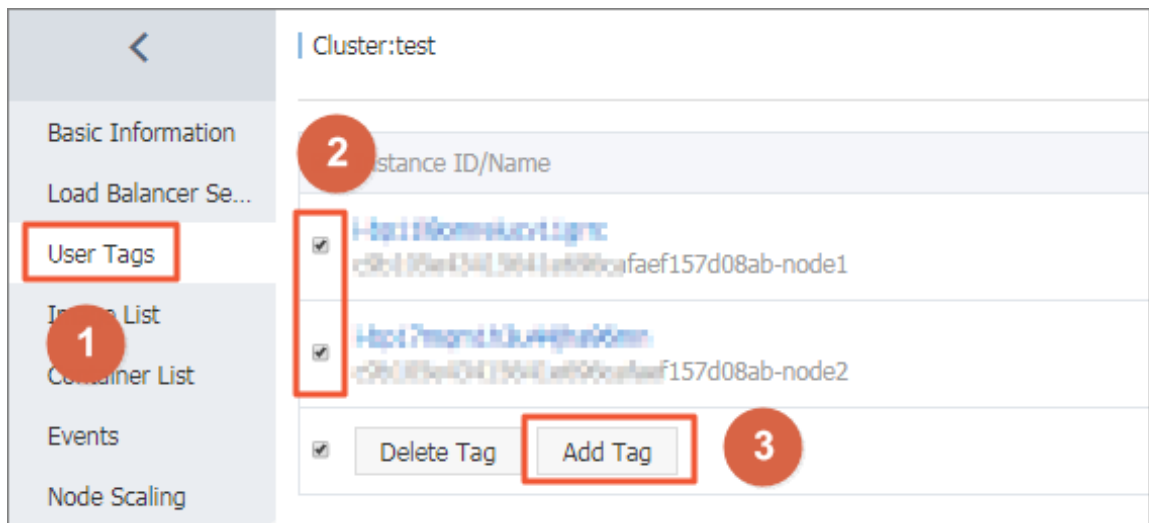
## Procedure

### 1. Add user tags for nodes.

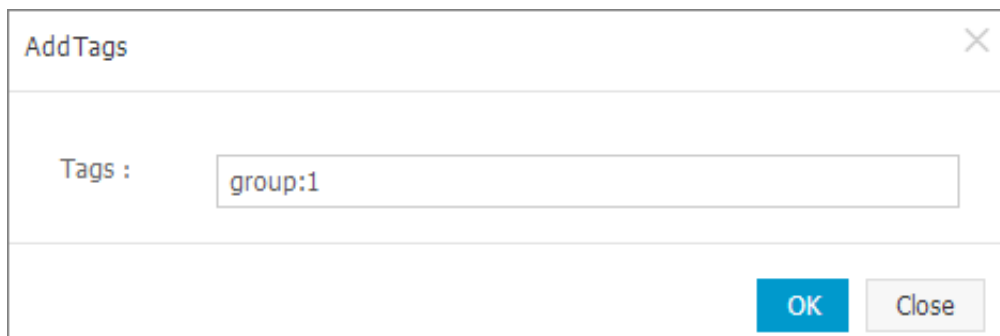
- a. Log on to the [Container Service console](#).
- b. Click Swarm > Clusters in the left-side navigation pane.
- c. Click Manage at the right of the cluster.



- 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.



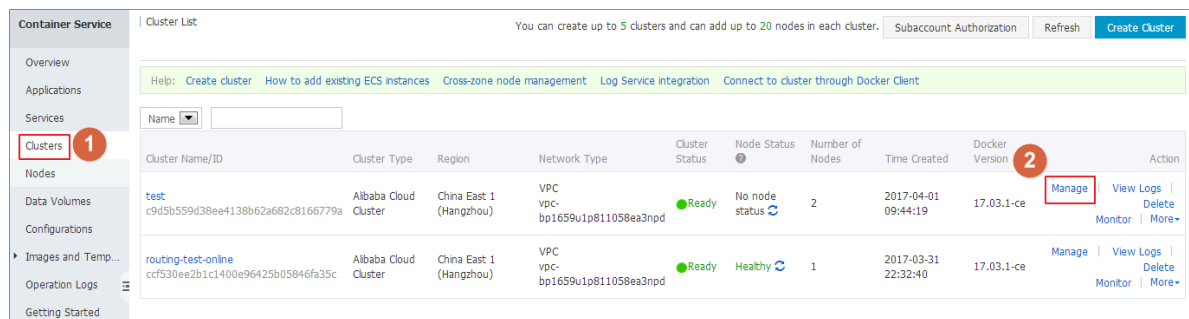
2. Create an application by clicking Create with Orchestration Template. Configure the `constraint` keyword in the template.

For information about how to create an application, see [#unique\\_119](#).

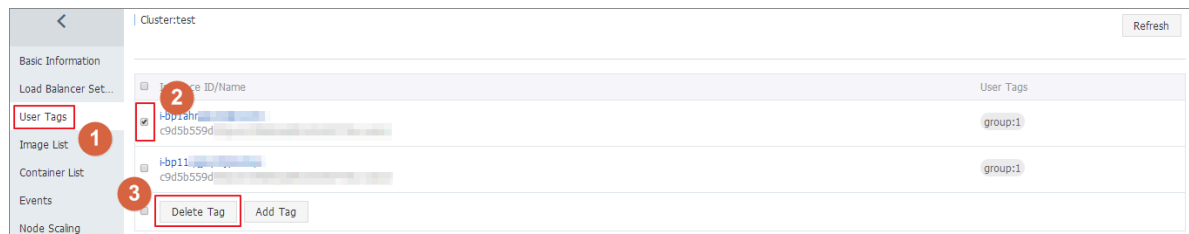
```
environment :
- constraint : group == 1 # Indicates to deploy the
  application on all the nodes with the "group : 1 "
  tag
```

### Delete a user tag

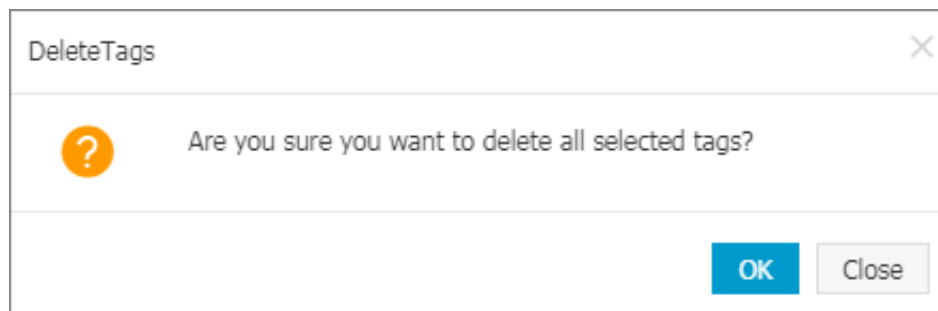
1. Log on to the [Container Service console](#).
2. Click Swarm > Clusters in the left-side navigation pane.
3. Click Manage at the right of the cluster.



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.



## 8 Configurations

### 8.1 Implement multiple environments by using configurations

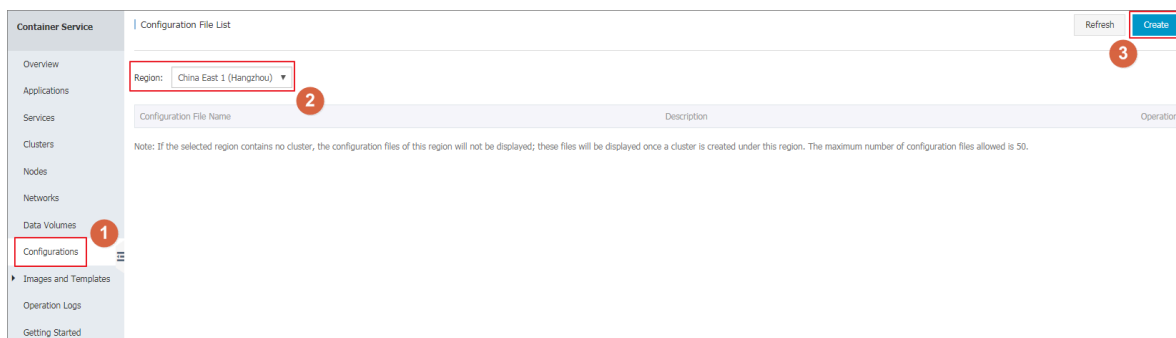
An application consists of codes and configurations. After an application is containerized, the configurations are usually transmitted by using container environment variables to deploy multiple applications using the same image and different configurations.

#### 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.

The screenshot shows a configuration dialog box with the following elements:

- \* File Name:** A text input field containing "test-group". Below it, a message states: "The configuration file name should contain 1 to 32 characters."
- Description:** A text area containing "test group". Below it, a message states: "The description can contain up to 128 characters."
- Configuration:** A section with an "Edit JSON File" button and a table of variables.
 

Variable Name	Variable Value	Action
size	2	Edit   Delete

 Below the table, there are input fields for "Name" and "Value", and an "Add" button. A message below these fields states: "The variable key should contain 1 to 32 characters; the variable value should contain 1 to 128 characters. The variable value must be unique. The variable name and variable value cannot be empty."

At the bottom of the dialog are "OK" and "Cancel" buttons.

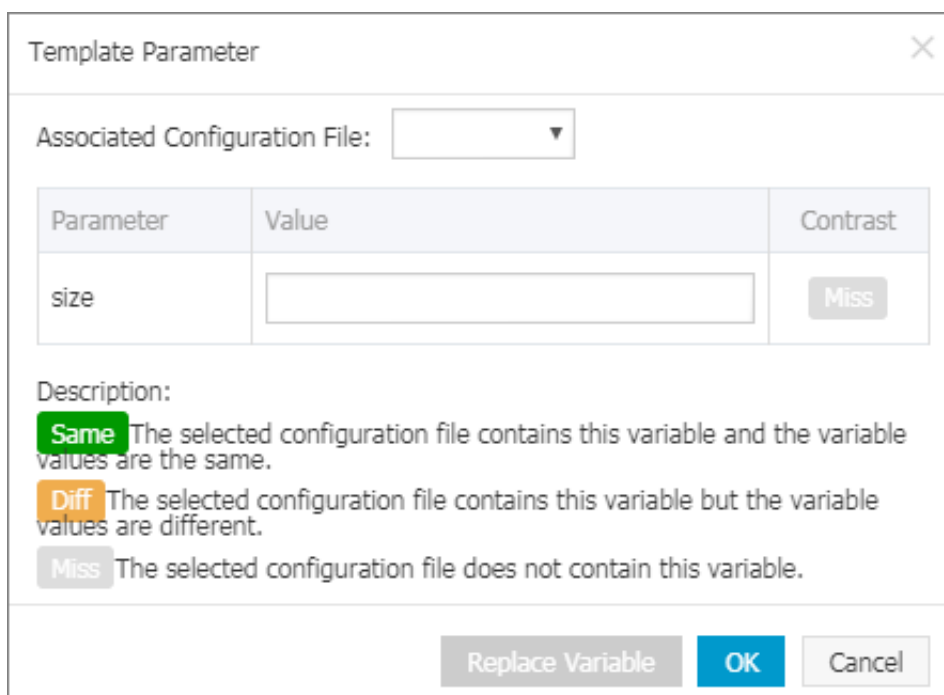
4. Under Swarm, click Applications in the left-side navigation pane. Select the cluster in the same region as the created configuration from the Cluster list and click Create Application.
5. Enter the basic information of the application and click Create with Orchestration Template.
6. Enter the following orchestration template and then click Create and Deploy.

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

```
busybox :
  image : ' busybox '
  command : ' top - b '
  labels :
```

```
aliyun . scale : $ size
```

- 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.



Template Parameter

Associated Configuration File:

Parameter	Value	Contrast
size	<input type="text"/>	<input type="button" value="Miss"/>

Description:

**Same** The selected configuration file contains this variable and the variable values are the same.

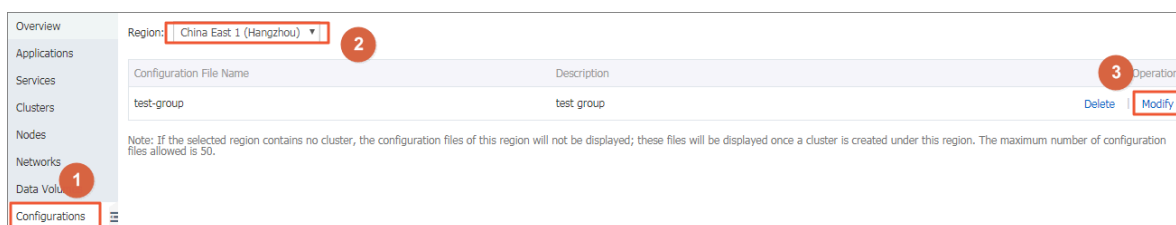
**Diff** The selected configuration file contains this variable but the variable values are different.

**Miss** The selected configuration file does not contain this variable.

## 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.

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



Overview

Applications

Services

Clusters

Nodes

Networks

Data Volume

Configurations

Region:

Configuration File Name	Description	Operation
test-group	test group	<input type="button" value="Delete"/> <input type="button" value="Modify"/>

Note: If the selected region contains no cluster, the configuration files of this region will not be displayed; these files will be displayed once a cluster is created under this region. The maximum number of configuration files allowed is 50.

- Click Confirm in the displayed dialog box.

- 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.

\* File Name: test-group

Description: test group

The description can contain up to 128 characters.

Configuration: [Edit JSON File](#)

Variable Name	Variable Value	Action
size	3	Save   Delete

Name Value Add

The variable key should contain 1 to 32 characters; the variable value should contain 1 to 128 characters. The variable value must be unique. The variable name and variable value cannot be empty.

[OK](#) [Cancel](#)

- 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.

Container Service | Application List

Refresh [Create Application](#)

Heber: [Create application](#) [Change application configuration](#) [Simple route blue-green release policy](#) [Container auto scaling](#)

Cluster: test [Hide system applications](#) [Hide offline applications](#) [Hide online applications](#)

Name	Description	Status	Container status	Time Created	Time Updated	Action
test		Ready	Ready2 Stop:0	2016-12-29 19:08:08	2016-12-29 19:29:57	Stop   Update   Delete   Redeploy   Events

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

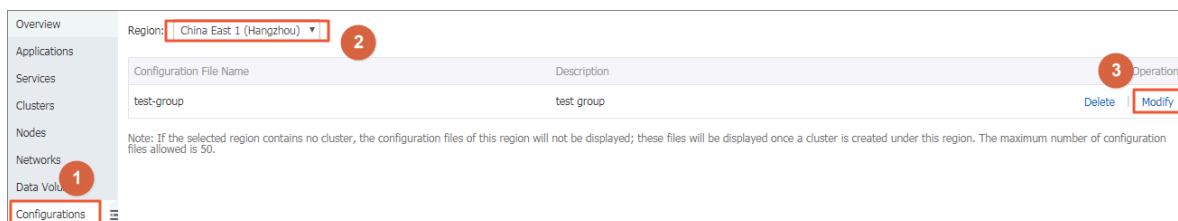
Cluster: test [Hide system applications](#) [Hide offline applications](#)

Name	Description	Status
test		Ready

## 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.

\* File Name: test-group

Description: test group

The description can contain up to 128 characters.

Configuration: [Edit JSON File](#)

Variable Name	Variable Value	Action
size	3	Save   Delete

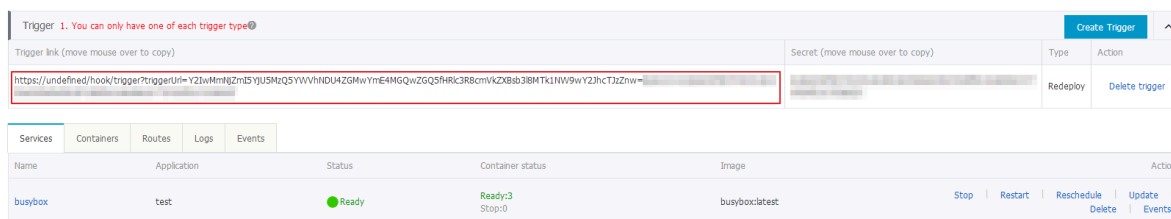
Name Value Add

The variable key should contain 1 to 32 characters; the variable value should contain 1 to 128 characters. The variable value must be unique. The variable name and variable value cannot be empty.

OK Cancel

## 5. Create a redeployment trigger.

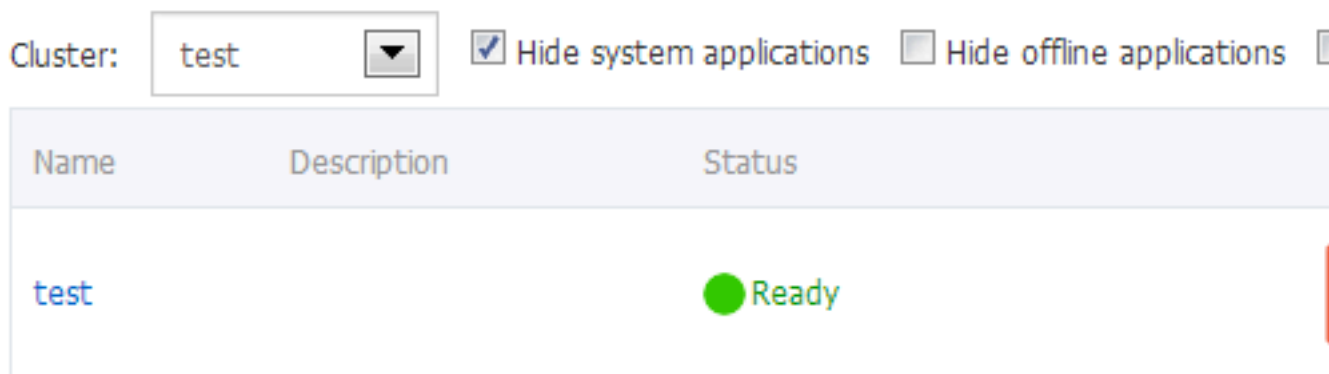
For how to create a trigger, see [#unique\\_162](#).



## 6. Initiate the redeployment trigger.

```
curl "https://cs.console.aliyun.com/hook/trigger?triggerUrl=Y2ViZDhkZT IwZGMzMjRm OTM4NDIzMT gwMzI3NmIw M2Ix fHRlc3 QtZ3JvdXB8 c2NhbGluZ3 wxOXZwYzNm OXFfINTcwfa ==&secret=4662423767 7565495154 6d6451656a 7a66e7f5b6 1db6885f8d 15aa648266 72c2"
```

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





## 9 Data volumes

---

### 9.1 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.

### 9.2 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.

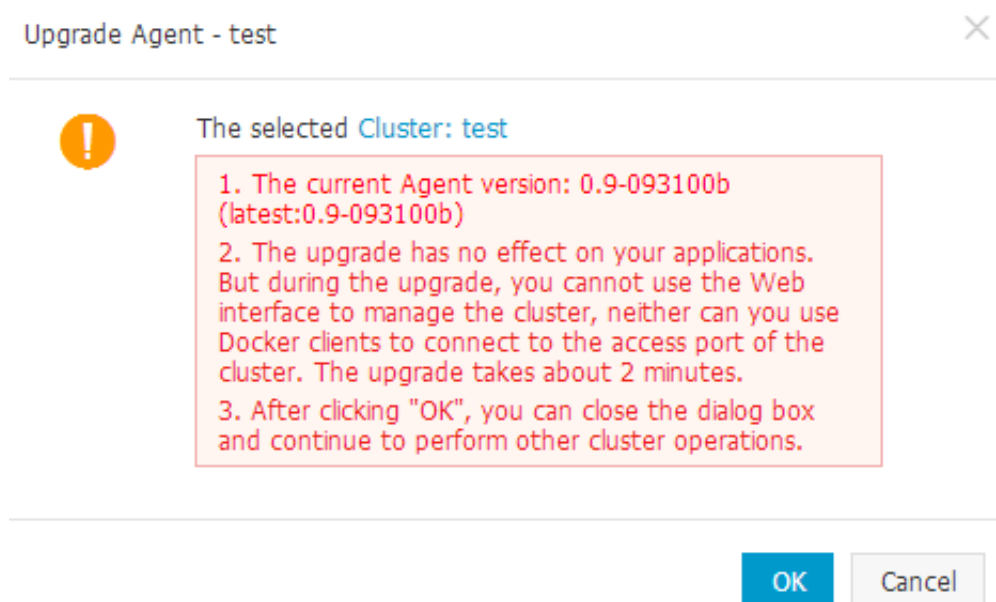
- Coordinate the actions of each client on your own when multiple clients are mounted to the same OSS bucket. For example, avoid multiple clients from writing the same file.
- Hard link is not supported.

### 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 [#unique\\_143](#).

- 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 [#unique\\_166](#).



#### Note:

When acsvolumedriver is upgraded or restarted, containers that use OSSFS data volumes are restarted, and your services are also restarted.

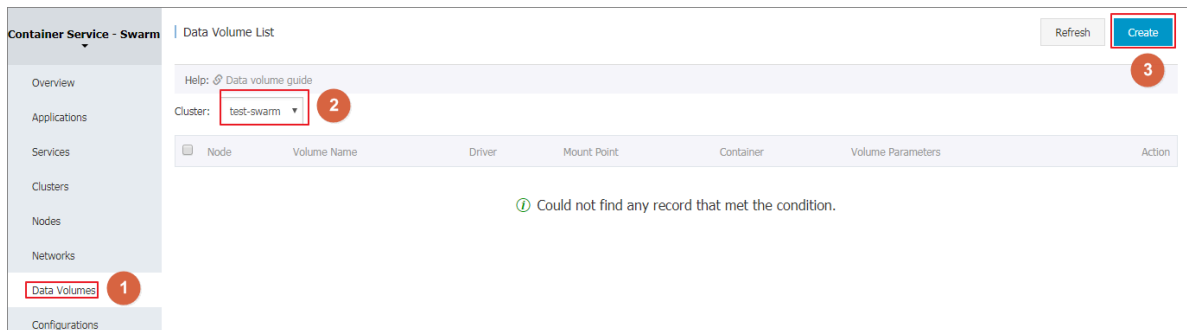
## Step 1. Create an OSS bucket

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.

Create Data Volume

Type: ☒ OSS ☐ Cloud Disk

Name:

Access Key ID:

Access Key Secret:

Optional Parameters: ☐ allow\_other ☐ noxattr

Other Parameters:

For the formats of other parameters, refer to this [document](#). Example: -o allow\_other -o default\_permission=666 -onoxattr

Note: Only clusters with volume driver version 0.7 or above support these parameters. You can go to the application list, find the acsvolumedriver application, and view the volumedriver service's image version in the service list on the application details page. If the image version is lower than 0.7, please upgrade the volumedriver.

Bucket ID: 

Select Bucket

Access Domain Name: ☐ Intranet ☐ Internet ☐ VPC

File Caching: ☐ Enable ☒ Close

Create

Cancel

- 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 [#unique\\_167](#).

## 9.3 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\\_169](#).

#### 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 [#unique\\_143](#).
- 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 [#unique\\_166](#).

#### 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.

**Cloud Disk** [Purchase ECS Instances](#) [Related Products](#)

Choose the Datacenter Region and Zone

Asia Pacific SE 1 (Singapore) Asia Pacific SE 1 Zone B	Asia Pacific SE 2 (Sydney) Asia Pacific SE 2 Zone A	Asia Pacific NE 1 (Tokyo) Asia Pacific NE 1 Zone A	EU Central 1 (Frankfurt) EU Central 1 Zone A	Middle East 1 (Dubai) Middle East 1 Zone A
US East 1 (Virginia) US East 1 Zone A	US West 1 (Silicon Valley) US West 1 Zone B	Hong Kong Hong Kong Zone C	China North 1 (Qingdao) China North 1 Zone C	China North 2 (Beijing) China North 2 Zone E
China North 3 (Zhangjiakou) China North 3 Zone A	<b>China East 1 (Hangzhou) China East 1 Zone F</b>	China East 2 (Shanghai) China East 2 Zone D	China South 1 (Shenzhen) China South 1 Zone B	China North 5 (Huhehaote) China North 5 Zone A
Asia Pacific SE 3 (Kuala Lumpur) Asia Pacific SE 3 Zone A				

Choose Storage

SSD Cloud Disk 20 GB [Create disk with snapshot](#)

Purchase Plan  
Instance Cost  
\$0.006 USD/hour

[Add To Cart](#) [Buy Now](#)

## 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.

Container Service - Swarm | Data Volume List

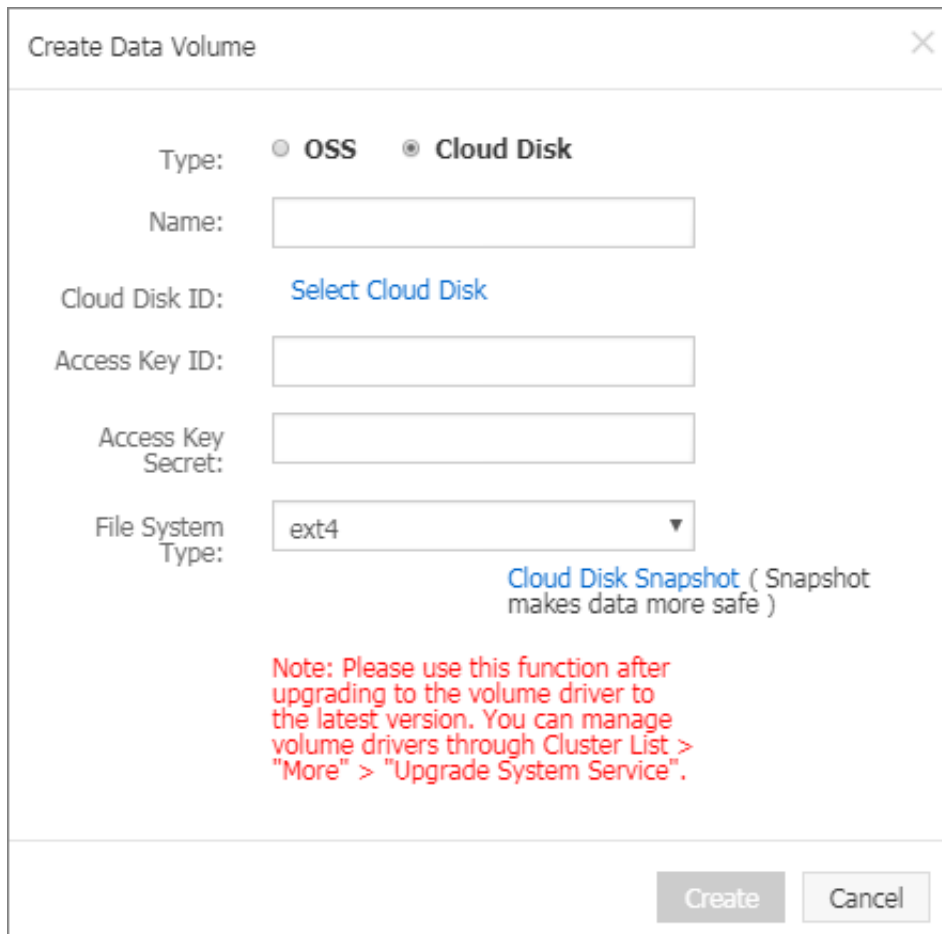
Help: [Data volume guide](#)

Cluster: test

Refresh Create

Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
Could not find any record that met the condition.						

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.



The dialog box titled "Create Data Volume" contains the following fields and options:

- Type:** Radio buttons for **OSS** and **Cloud Disk**. **Cloud Disk** is selected.
- Name:** A text input field.
- Cloud Disk ID:** A button labeled **Select Cloud Disk**.
- Access Key ID:** A text input field.
- Access Key Secret:** A text input field.
- File System Type:** A dropdown menu with **ext4** selected.

Below the fields, there is a link: [Cloud Disk Snapshot](#) ( Snapshot makes data more safe ).

A red note is displayed: **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".**

At the bottom right, there are **Create** and **Cancel** buttons.

- **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.

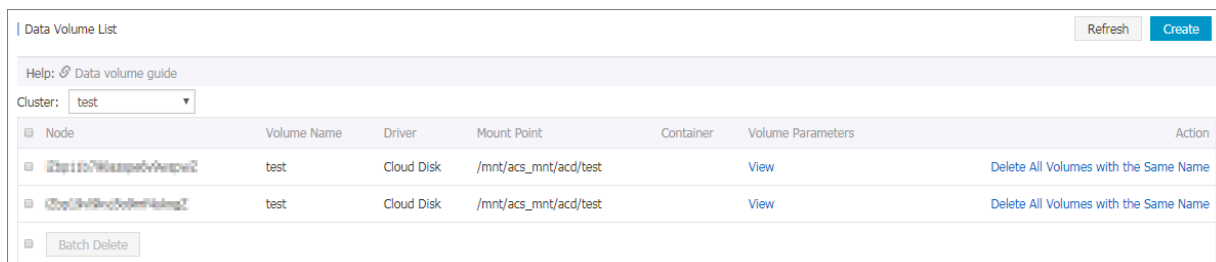
**Note:**

Container Service begins to support the STS Token function since December 5, 2017. If your cluster is created after that, you must enter the AccessKey when you create a cloud disk data volume in the cluster.

- **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.



Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
node1	test	Cloud Disk	/mnt/acs_mnt/acd/test		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>
node2	test	Cloud Disk	/mnt/acs_mnt/acd/test		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>

### 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.

## 9.4 View and delete data volumes

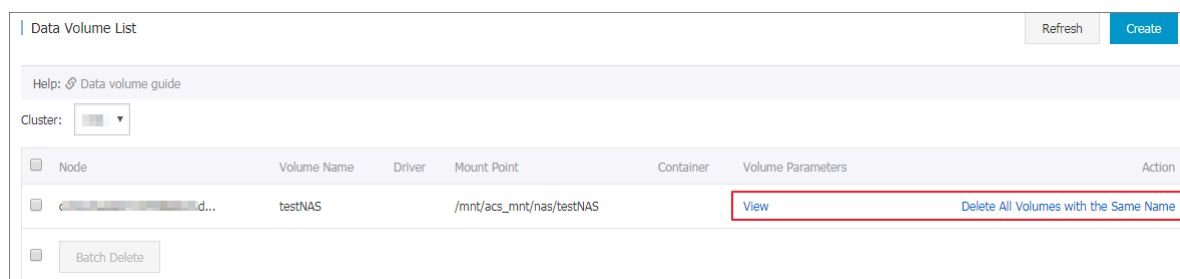
You can view and delete the created data volumes.

### 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.



Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
node1	testNAS		/mnt/acs_mnt/nas/testNAS		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>

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.

## 9.5 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 [#unique\\_172](#).

### Procedure

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

Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
<input type="checkbox"/> c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>
<input type="checkbox"/> c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>
<input type="checkbox"/> <a href="#">Batch Delete</a>						

### 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.

**Basic Information** Configuration Done

Name:   
The name should be 1-64 characters long, and can contain numbers, English letters and hyphens, but cannot start with a hyphen.

Version:

Cluster:

Update:

Description:

☐ Pull Docker Image

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 [#unique\\_119](#).

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.

Data Volume:  [Use third-party data volumes](#)

**Volume**

Host Path or Data Volume Name:  Container Path:  Permission:

volumes\_from:

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.

Cluster:

Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
<input type="checkbox"/> c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...	<input type="text" value="volume_volume_1"/>	<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>
<input type="checkbox"/> c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...		<a href="#">View</a>	<a href="#">Delete All Volumes with the Same Name</a>
<input type="checkbox"/> Batch Delete						

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 [#unique\\_119](#).

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.

Cluster:

Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...	volume_volume_1	View	Delete All Volumes with the Same Name
c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...		View	Delete All Volumes with the Same Name
Batch Delete						

## 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\\_173](#).



### 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

×

Name: volume

\*\*Version: 1.1

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

Description:

Use Latest Image: ☒

Force Reschedule: ☐

?

Release Mode: Standard Release

Template:

```

1 - volume:
2   restart: always
3   expose:
4     - 443/tcp
5     - 80/tcp
6   labels:
7     aliyun.scale: '1'
8   image: 'nginx:latest'
9   volumes:
10    - 'test:/testvolume:rw'

```

Use Existing Orchestration Template

Label description

OK

Cancel

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.

Cluster: test

Node	Volume Name	Driver	Mount Point	Container	Volume Parameters	Action
c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...	volume_volume_1	View	Delete All Volumes with the Same Name
c9d5b559d38ee4138b62a682...	test	OSS File System	/mnt/acs_mnt/ossfs/volum...		View	Delete All Volumes with the Same Name
Batch Delete						

## 9.6 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.

```
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.

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

# 10 Logs

## 10.1 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.

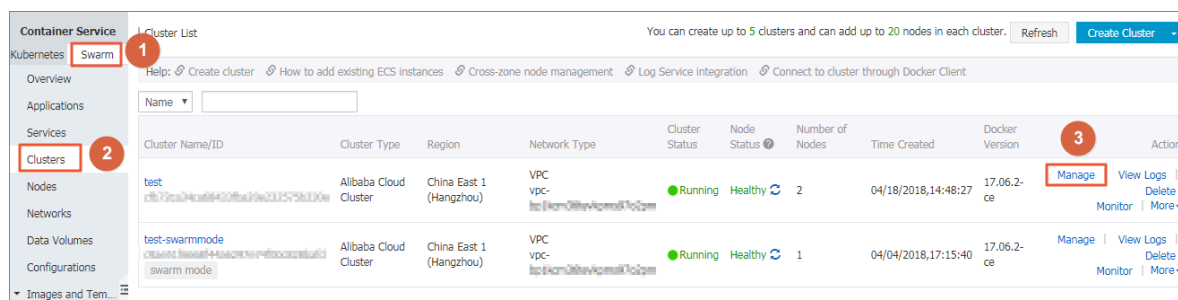


### Note:

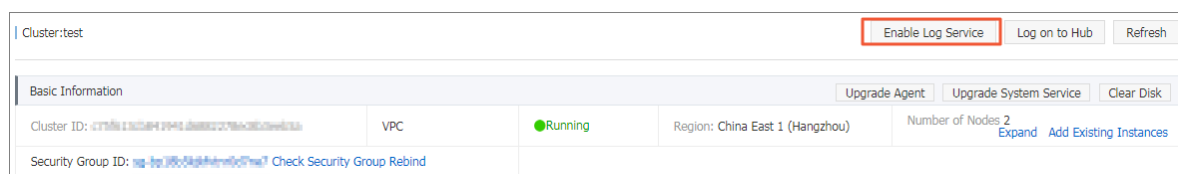
On the cluster management page, choose Enable Log Service > OK. After Log Service is successfully enabled, the log index is created for each automatically created Logstore by using the built-in Resource Access Management (RAM) account. With this feature enabled, you are charged for the Alibaba Cloud Log Service usage after configuring the following settings. For more information, see [#unique\\_177](#). 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.



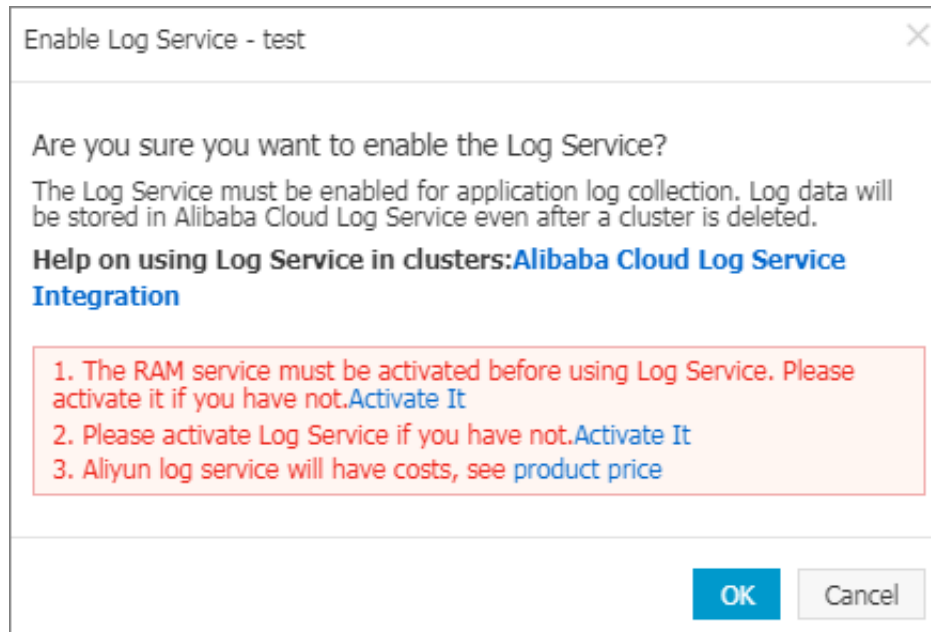
4. Click Enable Log Service in the upper-right corner.





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.

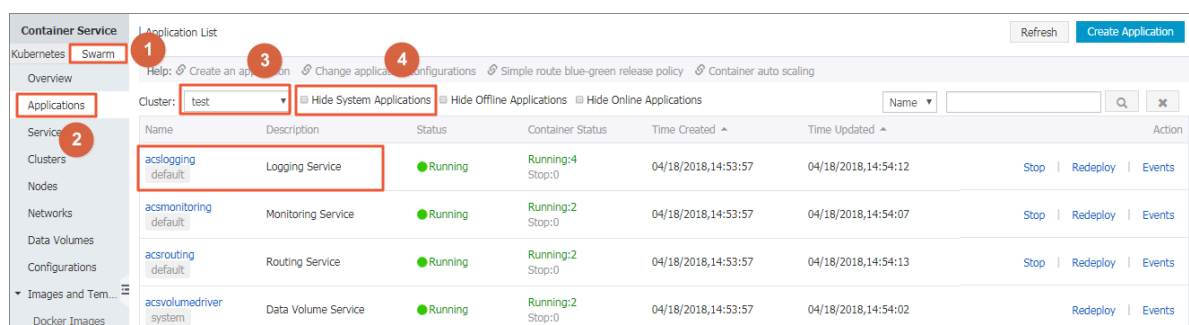


Check installation result of acslogging service

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.



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.

acslog-project-cfb72ca34c-mavbu	cfb72ca34ca68433fba20e2...	China East 1 (Hangzhou)	2018-04-18 15:14:20	<a href="#">Modify</a>   <a href="#">Delete</a>
---------------------------------	----------------------------	-------------------------	---------------------	---

## 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

```
mysql :
  image : mysql
  ports :
    - 80
  labels :
    aliyun . scale : " 1 "
  environmen t :
    - MYSQL_ROOT _PASSWORD = password
web :
  image : registry . aliyuncs . com / jiangjizho ng / 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_db stdout : 30 # Set the data
retention time for the dbstdout Logstore to 30 days .
  links :
    - mysql
```

In the preceding orchestration file:

- `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 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.



Logstore List

Endpoint List

Create

Searching by logstore name

Search

Logstore Name	Data Import Wizard	Monitor	Log Collection Mode	Log Consumption Mode			Action
				LogHub	LogShipper	LogSearch	
acslog-wordpress-dbstdout			Logtail Config (Manage)   Diagnose   More Data	Preview	OSS	<div>Search</div>	Modify   Delete

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 "
```

# 11 DevOps

## 11.1 Jenkins-based continuous delivery

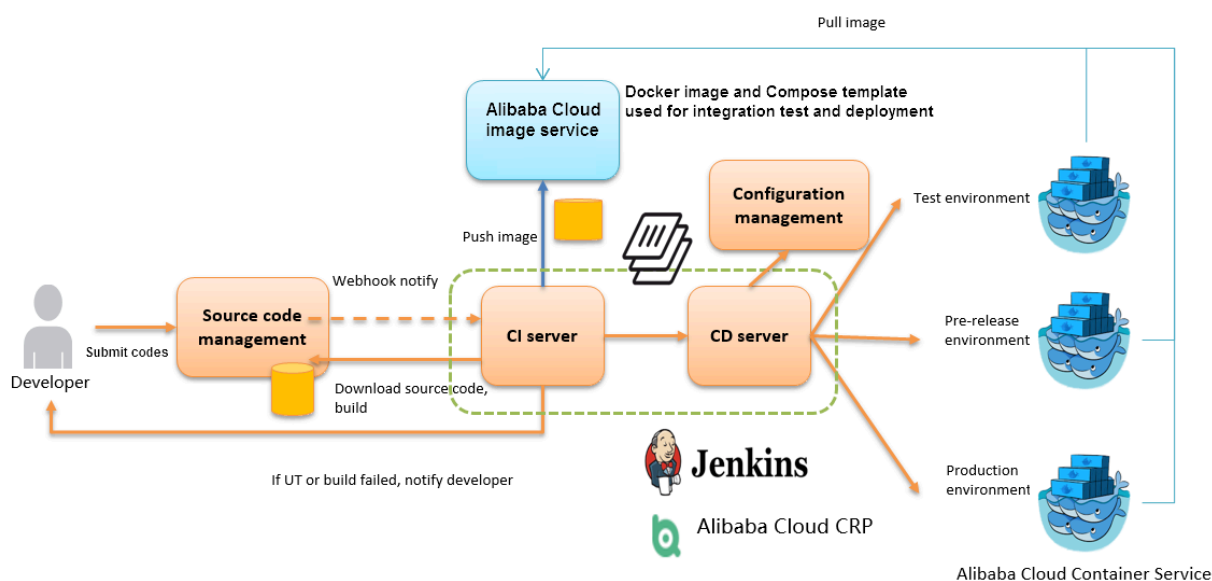
As an important step in agile development, continuous integration aims to maintain high quality while accelerating product iteration. Every time codes are updated, an automated test is performed to test the codes and function validity. The codes can only be delivered and deployed after they pass the 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 [#unique\\_180](#).

```
jenkins :
  image : ' registry . aliyuncs . com / acs - sample / jenkins : 1
. 651 . 3 '
  volumes :
    - / var / lib / docker / jenkins : / var / jenkins_ho me
  restart : always
  labels :
    aliyun . scale : ' 1 '
    aliyun . probe . url : ' tcp : // container : 8080 '
    aliyun . probe . initial_de lay_second s : ' 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 '
```

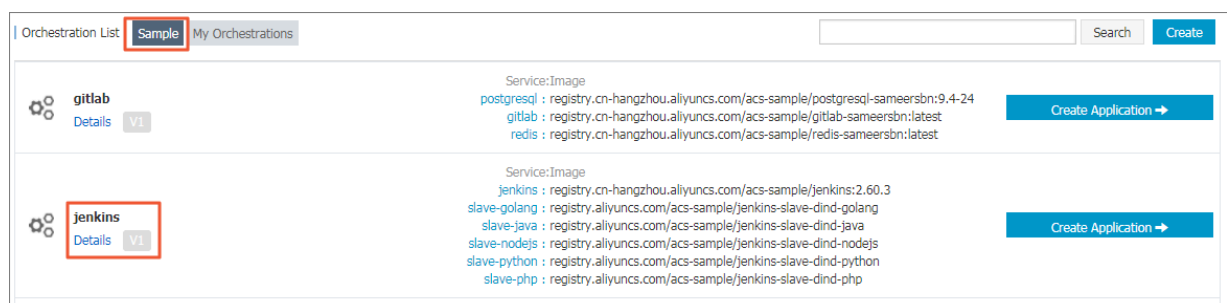
## 1.2 Use the template to create Jenkins application and slave node

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 [#unique\\_119](#).



After a successful creation, the Jenkins application and slave node are displayed in the service list.

Application:jenkins

Refresh

Overview

Name:jenkins

Time Created: 2018-01-16

Time Updated: 2018-01-16

Cluster: test

Trigger 1. You can only have one of each trigger type.

Create Trigger

No trigger is available at the moment. Click "Create Trigger" in the upper-right corner.

Services

Containers

Logs

Events

Routes

Name	Application	Status	Container Status	Image	Action
jenkins	jenkins	<div></div> Ready	Ready:1 Stop:0	registry.cn-hangzhou.aliyuncs.com/acs-sample/jen...	<div>Stop</div> <div>Restart</div> <div>Reschedule</div> <div>Update</div> <div>Delete</div> <div>Events</div>
slave-golang	jenkins	<div></div> Ready	Ready:1 Stop:0	registry.aliyuncs.com/acs-sample/jenkins-slave-d...	<div>Stop</div> <div>Restart</div> <div>Reschedule</div> <div>Update</div> <div>Delete</div> <div>Events</div>
slave-java	jenkins	<div></div> Ready	Ready:1 Stop:0	registry.aliyuncs.com/acs-sample/jenkins-slave-d...	<div>Stop</div> <div>Restart</div> <div>Reschedule</div> <div>Update</div> <div>Delete</div> <div>Events</div>
slave-nodejs	jenkins	<div></div> Ready	Ready:1 Stop:0	registry.aliyuncs.com/acs-sample/jenkins-slave-d...	<div>Stop</div> <div>Restart</div> <div>Reschedule</div> <div>Update</div> <div>Delete</div> <div>Events</div>

Open the access endpoint provided by Container Service to use the deployed Jenkins application.

Service:jenkins\_jenkins

Refresh

Scale

Overview

Service Name: jenkins

Application: jenkins

Image: registry.cn-hangzhou.aliyuncs.com/acs-sample/jenkins:2.60.3

Number: 1

● Ready

Access Endpoint: http://jenkins.8402cbd57131355b...cn-hangzhou.alicontainer.com

Containers

Logs

Configurations

Events

Name/ID	Status	Health Check	Image	Port	Container IP	Node IP	Action
jenkins_jenkins_... 8402cbd57131355b...	running	Normal	registry.cn-hang... sha256:a33929a9c...	8080/tcp 50000/tcp	172.17.0.5	192.168.1.109	Delete   Stop   Monitor   Logs   Web Terminal

## 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.



Name: slave-nodejs-ut

Description: slave-nodejs-ut

# of executors: 1

Remote root directory: /home/jenkins

Labels: slave-nodejs-ut

Usage: Utilize this node as much as possible

Launch method: Launch slave agents on Unix machines via SSH

Host: 172.17.0.1

Credentials: jenkins/\*\*\*\*\* Add

Availability: Keep this slave on-line as much as possible

Node Properties

☒ Environment variables

☒ Tool Locations

Save



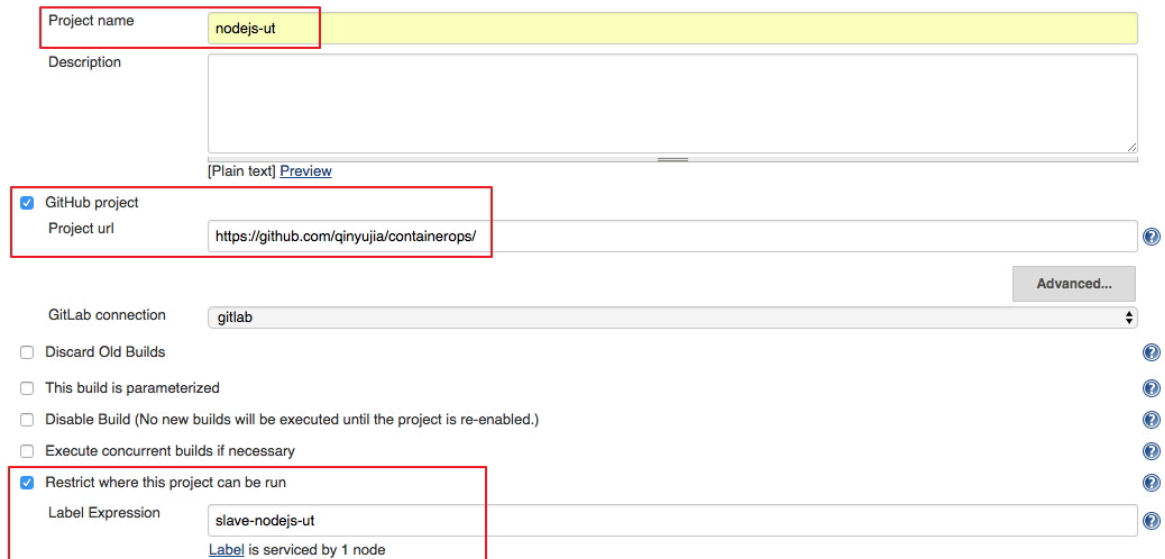
#### Note:

- Label is the unique identifier of the slave.
- The slave container and Jenkins container run on the Alibaba Cloud platform at the same time. Therefore, enter a container node IP address that is inaccessible to the Internet to isolate the test environment.
- When adding the credentials, use the jenkins account and password (the initial password is jenkins) in Dockerfile for the creation of the slave-nodejs image. The image Dockerfile address is [jenkins-slave-dind-nodejs](#).

## 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.



Project name: `nodejs-ut`

Description: [Plain text] [Preview](#)

☒ GitHub project  
Project url: `https://github.com/qinyujia/containerops/`

GitLab connection: `gitlab`

☐ Discard Old Builds

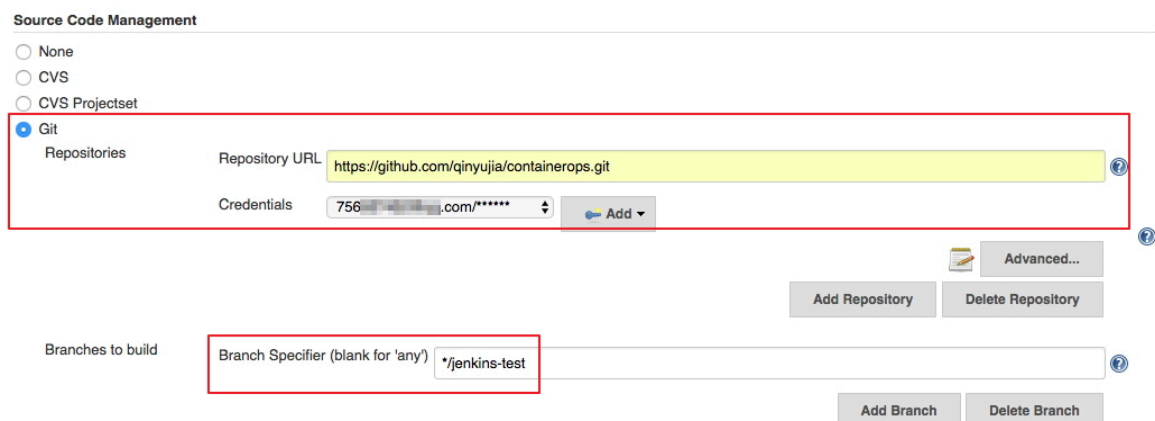
☐ This build is parameterized

☐ Disable Build (No new builds will be executed until the project is re-enabled.)

☐ Execute concurrent builds if necessary

☒ Restrict where this project can be run  
Label Expression: `slave-nodejs-ut`  
[Label](#) is serviced by 1 node

3. Configure the source code management and code branch. In this example, use GitHub to manage source codes.



Source Code Management

☐ None

☐ CVS

☐ CVS Projectset

☒ Git

Repositories

Repository URL: `https://github.com/qinyujia/containerops.git`

Credentials: `756...com/*` [Add](#)

Advanced...

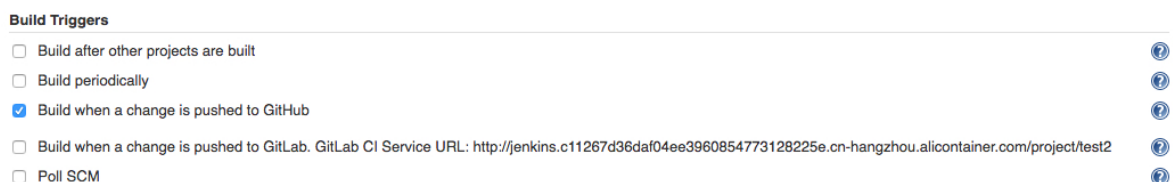
Add Repository Delete Repository

Branches to build

Branch Specifier (blank for 'any'): `*/jenkins-test`

Add Branch Delete Branch

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



Build Triggers

☐ Build after other projects are built

☐ Build periodically

☒ Build when a change is pushed to GitHub

☐ Build when a change is pushed to GitLab. GitLab CI Service URL: `http://jenkins.c11267d36daf04ee3960854773128225e.cn-hangzhou.alicloud.com/project/test2`

☐ Poll SCM

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 /
```

The screenshot shows the 'Services / Add Jenkins (Git plugin)' page in the QiniuJia ContainerOps repository. The left sidebar contains navigation links: Options, Collaborators, Branches, Webhooks (selected), Integrations & services, and Deploy keys. The main content area has a tab for 'Settings'. Under 'Install Notes', it lists requirements for the Git Plugin and the Jenkins URL. The 'Details' section explains the Jenkins service. A red box highlights the 'Jenkins url' input field, which contains 'http://jenkins.c112...-hang', and the 'Active' checkbox, which is checked. Below these fields is a green 'Add service' button.

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

The screenshot shows the 'Build' configuration page in Jenkins. The 'Execute shell' build step is selected. The 'Command' field contains the following shell script:

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

Below the command field is a link: [See the list of available environment variables](#). A red 'Delete' button is located in the bottom right corner.

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 `post-commit` file. For example:

```
curl -u test:test
```

```
http://127.0.0.1:8080/jenkins/job/svn/build?token=qinyujia
```

## 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.

**Build Triggers**

☒ Build after other projects are built

Projects to watch:

☒ Trigger only if build is stable  
☐ Trigger even if the build is unstable  
☐ Trigger even if the build fails

☐ Build periodically  
☐ Build when a change is pushed to GitHub  
☐ Build when a change is pushed to GitLab. GitLab CI Service URL: <http://jenkins.c11267d36daf04ee3960854773128225e.cn-hangzhou.alicontainer.com/project/nodejs-build>  
☐ Poll SCM

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

**Build**

☒ Execute shell

Command:

```
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
```

[See the list of available environment variables](#)

Delete

The commands in this example are as follows:

```
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:

```
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\\_162](#).

Trigger 1. You can only have one of each trigger type. ⓘ				Create Trigger	^
Trigger Link (move mouse over to copy)		Secret (move mouse over to copy)		Type	Action
https://undefined/hook/trigger?triggerUrl=YzkNW1NTkMzhIZTOxMzhINjJhNjYyZDZlZGlmYmtpbnN8cmVkb3B3b3RMTjYTNMTYy		74386f737245553732703738674b7966439e		Redeploy	Delete Trigger

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

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

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

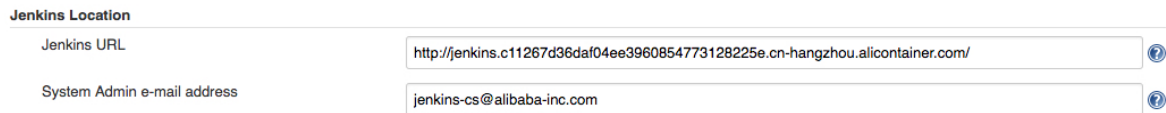
```
cd chapter2
sudo docker build - t registry . aliyuncs . com / qinyujia
- test / nodejs - demo .
sudo docker login - u ${ yourAccoun t } - p ${ yourPasswo
rd } registry . aliyuncs . com
sudo docker push registry . aliyuncs . com / qinyujia - test
/ nodejs - demo
curl ' https :// cs . console . aliyun . com / hook / trigger ?
triggerUrl =***=& secret =***'
```

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**

Jenkins URL

System Admin e-mail address

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:



**E-mail Notification**

SMTP server

Default user e-mail suffix

☒ Use SMTP Authentication

User Name

Password

Use SSL ☒

SMTP Port

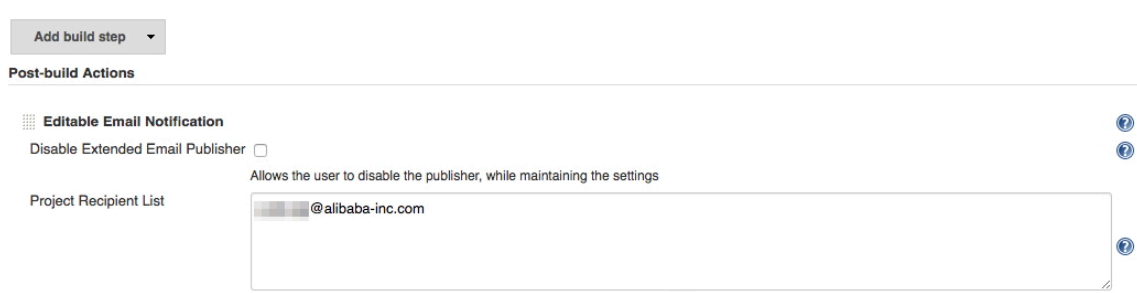
Reply-To Address

Charset

☐ Test configuration by sending test e-mail

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.



**Add build step**

**Post-build Actions**

**Editable Email Notification**

Disable Extended Email Publisher ☐

Project Recipient List

#### 4. Add a trigger to send emails.

Triggers

<div><div></div>Always</div>	<div><div></div>?</div>
Send To	
<div><div></div>Recipient List</div>	<div><div></div>?</div>
	<div>Delete</div>
<div><div></div>Developers</div>	<div><div></div>?</div>
	<div>Delete</div>
<div><div></div>Requestor</div>	<div><div></div>?</div>
	<div>Delete</div>
<div>Add ▼</div>	<div>?</div>
	<div>Advanced...</div>
	<div>Remove Trigger</div>



## 12 Service discovery and load balancing

---

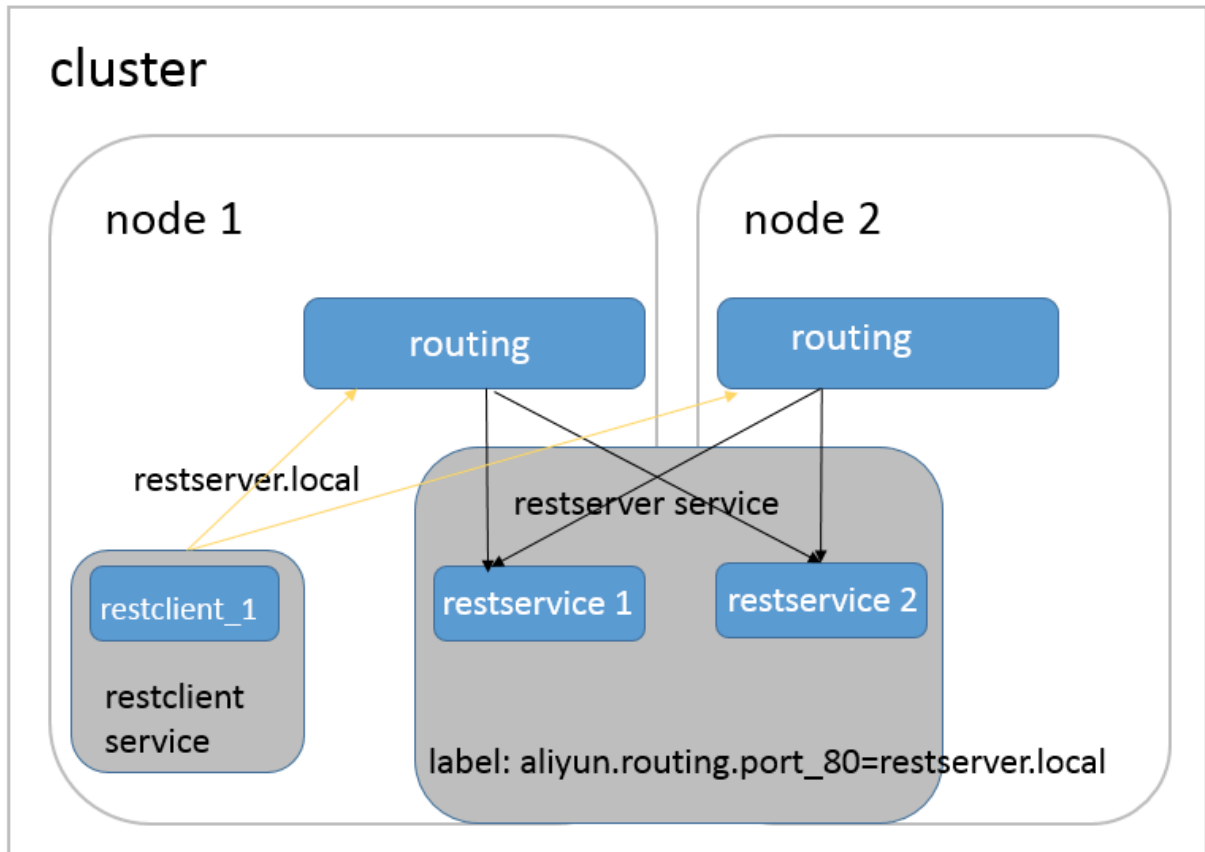
### 12.1 Routing and Server Load Balancer between services in a cluster

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_l inks` 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 fbc5405ff 791650a074 6ab77f26cc e74fea2320 075c2af55c 975f and b8c36abca5 25ac7fb02d 2a9fcaba8d 36641447a7 74ea956cd9 3068419f17 ee3f .

```
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066803626Z Server : nginx / 1 . 11 . 1
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43
: 49 . 066814507Z Date : Fri , 01 Jul 2016 06 : 43 : 49
GMT
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066821392Z Content - Type : text / html
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066829291Z Content - Length : 612
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066835259Z Last - Modified : Tue , 31 May 2016 14 : 40
: 22 GMT
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066841201Z ETag : " 574da256 - 264 "
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
49 . 066847245Z Accept - Ranges : bytes
```

```

internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 :
43 : 49 . 066853137Z Set - Cookie : CONTAINERI D = 053cb232fd
fbc5405ff 791650a074 6ab77f26cc e74fea2320 075c2af55c 975f ;
path =/
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 080502413Z HTTP / 1 . 1 200 OK
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 082548154Z Server : nginx / 1 . 11 . 1
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43
: 50 . 082559109Z Date : Fri , 01 Jul 2016 06 : 43 : 50
GMT
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 082589299Z Content - Type : text / html
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 082596541Z Content - Length : 612
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 082602580Z Last - Modified : Tue , 31 May 2016 14 : 40
: 22 GMT
internal - loadbalanc e_restclie nt_1 2016 - 07 - 01T06 : 43 :
50 . 082608807Z ETag : " 574da256 - 264 "
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 : 43 :
50 . 082614780Z Accept - Ranges : bytes
internal - loadbalanc e_restclie nt_1 | 2016 - 07 - 01T06 :
43 : 50 . 082621152Z Set - Cookie : CONTAINERI D = b8c36abca5
25ac7fb02d 2a9fcaba8d 36641447a7 74ea956cd9 3068419f17 ee3f ;
path =/

```

## 12.2 Custom routing - simple sample

In this example, an [#unique\\_184/unique\\_184\\_Connect\\_42\\_section\\_it1\\_v2z\\_xdb](#) container is deployed, services are exposed by using a Server Load Balancer instance (with the [#unique\\_154](#) 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 subscrip on registry center and dynamical y load
    the service route .

```

```

    aliyun . custom_add on : " 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
    environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    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 :
    # http / https / ws / wss are supported . Use your
    own domain name instead of the test domain name
    provided by Container Service .
    aliyun . proxy . VIRTUAL_HO ST : " http :// appone . example
    . com "
    restart : always

```

After the service is successfully started, the following figure appears.



## Enable session persistence

```

lb :
  image : registry . aliyuncs . com / acs / proxy : 0 . 6
  ports :
    - ' 80 : 80 '
  restart : always
  labels :
    # Addon allows the proxy image to function as
    a subscripti on registry center and dynamical l y load
    the service route .
    aliyun . custom_add on : " 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 applicatio
ns .
    ADDITIONAL _SERVICES : "*"
appone :
    ports :
        - 80 / tcp
        - 443 / tcp
    image : ' nginx : latest '
    labels :
        # http / https / ws / wss are supported .
        aliyun . proxy . VIRTUAL_HO ST : " http :// appone . example
. com "
        # Session persistenc e is enabled , the cookie
method is applied , and the key is CONTAINERI D .
        aliyun . proxy . COOKIE : " CONTAINERI D insert indirect
"
    restart : always

```

## 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.



To add messages to the 503 page, add the `/ errors` folder to the VM where the container resides and add the `/ errors / 503 . http` file with the following content:

```

HTTP / 1 . 0 503 Service Unavailabl e
Cache - Control : no - cache
Connection : close
Content - Type : text / html ; charset = UTF - 8
< html >< body >< h1 > 503 Service Unavailabl e < / h1 >
< h3 > No server is available to handle this request .< /
h3 >
< li > If you are the visitor of this applicatio n
, contact the applicatio n maintainer to solve the
problem . < / li >
< li > If you are the applicatio n maintainer , view the
following informatio n . < / li >
< li > You are using the simple routing service . The
request is sent from Server Load Balancer to
the acsrouting applicatio n container then to your

```

```

    applicatio n container . Follow these steps for
    troublesho ting . </ li >
< li > Log on to the Container Service console . Click
    " Services " in the left - side navigation pane . Select
    the correspond ing 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
    correspond ing service . </ li >
< li > Locate and troublesho t the problem by following
    the instructio ns described in < a href = " https :// www .
    . alibabacloud . com / help / faq - detail / 42660 . html "> . </ li
    >
< li > View Routing FAQs < a href = " https :// www .
    alibabacloud . com / help / zh / faq - detail / 42658 . html "> . </
    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:

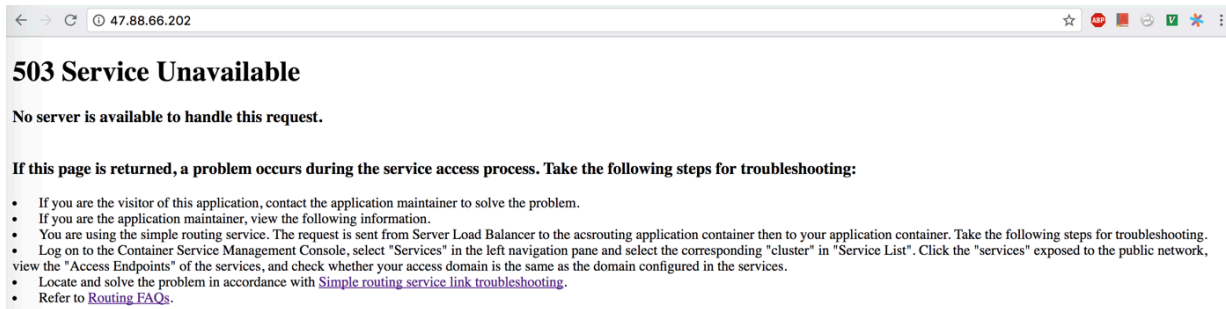
```

lb :
  image : registry . aliyuncs . com / acs / proxy : 0 . 6
  ports :
    - ' 80 : 80 '
  restart : always
  labels :
    # Addon allows the proxy image to function as
    a subscripti on registry center and dynamical y load
    the service route .
    aliyun . custom_add on : " 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    ADDITIONAL _SERVICES : "*"
    EXTRA_FRO NTEND_SETTI NGS_80 : " errorfile 503 / usr /
    local / etc / haproxy / errors / 503 . http "
  volumes :
    - / errors :/ usr / local / etc / haproxy / errors /
  appone :
    ports :
      - 80 / tcp
      - 443 / tcp
    image : ' nginx : latest '
    labels :
      # You can specify paths when configurin g URLs
      . In this example , http / https / ws / wss are supported .
      aliyun . proxy . VIRTUAL_HO ST : " http :// appone . example
      . com "

```

```
restart : always
```

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



## 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

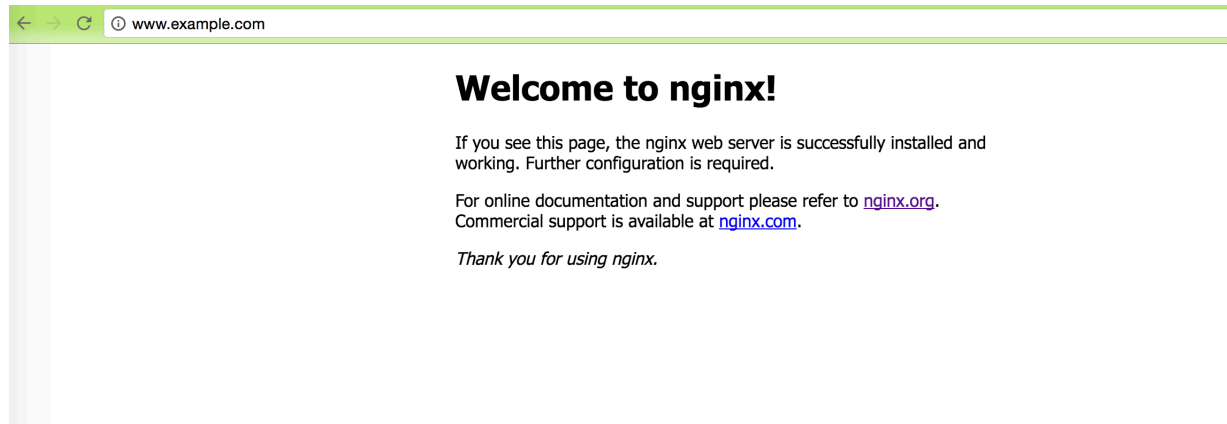
appone . example . com and \*. example . com ).

```
lb :
  image : registry . aliyuncs . com / acs / proxy : 0 . 6
  ports :
    - ' 80 : 80 '
  restart : always
  labels :
    # Addon allows the proxy image to function as
    a subscripti on registry center and dynamical l y load
    the service route .
    aliyun . custom_add on : " 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    ADDITIONAL _SERVICES : "*"
    EXTRA_FRON TEND_SETTI NGS_80 : " errorfile 503 / usr /
    local / etc / haproxy / errors / 503 . http "
  volumes :
    - / errors :// usr / local / etc / haproxy / errors /
  appone :
  ports :
    - 80 / tcp
    - 443 / tcp
  image : ' nginx : latest '
  labels :
    # You can specify paths when configurin g URLs
    . In this example , http / https / ws / wss are supported .
    aliyun . proxy . VIRTUAL_HO ST : " http ://*. example . com
  "
```



```
restart : always
```

Bind a host and enter the domain name `www . example . com` . The Nginx homepage is displayed as follows.



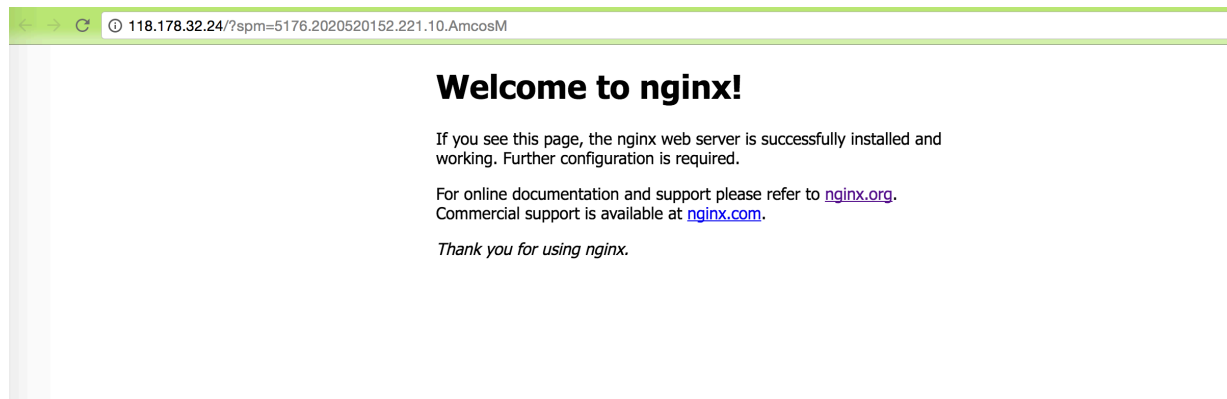
### Configure default backend

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 subscripti on registry center and dynamical ly load
    the service route .
    aliyun . custom_add on : " 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    ADDITIONAL _SERVICES : "*"
    # Specify the error page when 503 is returned .
    EXTRA_FRON TEND_SETTI NGS_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.



### 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:

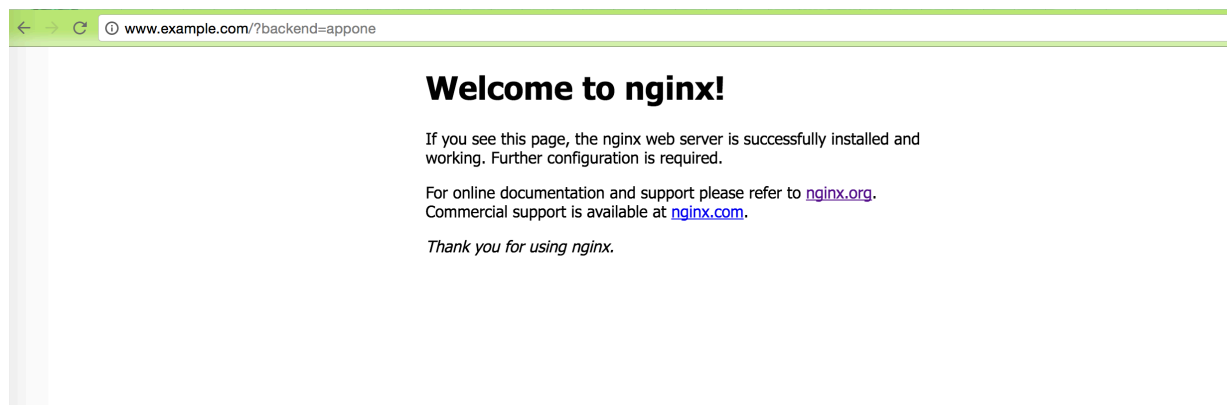
```
lb :
  image : registry . aliyuncs . com / acs / proxy : 0 . 6
  ports :
    - ' 80 : 80 '
  restart : always
  labels :
    # Addon allows the proxy image to function as
    a subscripti on registry center and dynamicall y load
    the service route .
    aliyun . custom_add on : " proxy "
    # 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    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 .
```

```

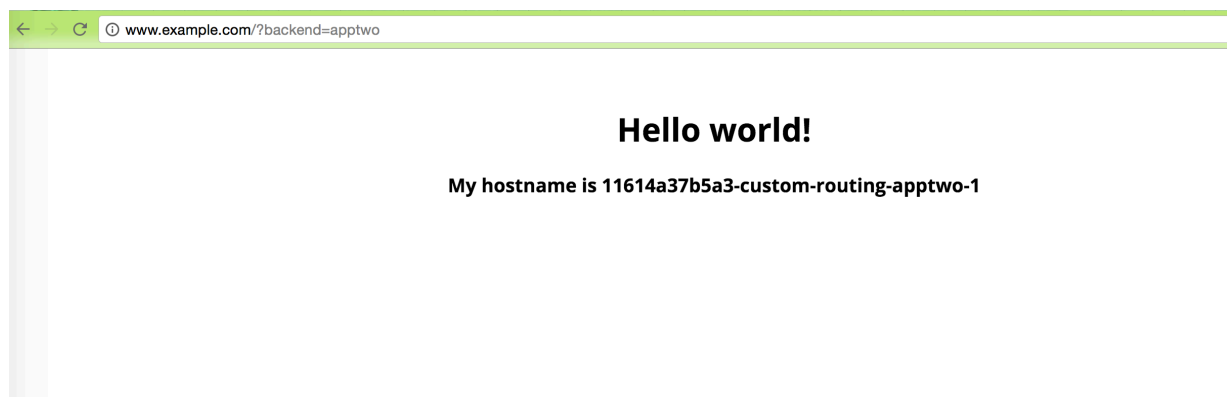
        EXTRA_FROM_TEND_SETTI_NGS_80 : " http - request set -
header HOST %[ urlp ( backend ) ]. example . com "
appone :
    ports :
        - 80 / tcp
        - 443 / tcp
    image : ' nginx : latest '
    labels :
        # You can specify paths when configurin g URLs
        . In this example , http / https / ws / wss are supported .
        aliyun . proxy . VIRTUAL_HO ST : " 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 configurin g URLs
        . In this example , http / https / ws / wss are supported .
        aliyun . proxy . VIRTUAL_HO ST : " 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.



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 subscripti on registry center and dynamicall y load
    the service route .
    aliyun . custom_add on : " 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    ADDITIONAL _SERVICES : "*"
    EXTRA_DEFA ULT_SETTIN GS : " log rsyslog local0 , log
    global , option httplog "
  links :
    - rsyslog : rsyslog
  rsyslog :
    image : registry . cn - hangzhou . aliyuncs . com / linhuatest /
    rsyslog : latest
  appone :
    ports :
      - 80 / tcp
      - 443 / tcp
    image : ' nginx : latest '
    labels :
      # http / https / ws / wss are supported .
      aliyun . proxy . VIRTUAL_HO ST : " http :// appone . example
      . com "
    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_co`  
`ntainer_na me` .

## 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` .

```
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 applicatio
      ns .
      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.

```

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` .

```

/ # drill appone . example . com
;; ->> HEADER <<- opcode : QUERY , rcode : NOERROR , id : 60917
;; flags : qr rd ra ; QUERY : 1 , ANSWER : 3 , AUTHORITY
: 0 , ADDITIONAL : 0
;; QUESTION SECTION :
;; appone . example . com . IN A
;; ANSWER SECTION :

```

```

appone . example . com .      600    IN    A    172 . 18 . 3 . 4
appone . example . com .      600    IN    A    172 . 18 . 2 . 5
appone . example . com .      600    IN    A    172 . 18 . 1 . 5
;; AUTHORITY SECTION :
;; ADDITIONAL SECTION :
;; Query time : 0 msec
;; SERVER : 127 . 0 . 0 . 11
;; WHEN : Mon Sep 26 07 : 09 : 40 2016
;; MSG SIZE rcvd : 138

```

## Configure monitoring page

```

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_add on : " proxy "
    aliyun . global : " true "
    aliyun . lb . port_80 : tcp :// proxy_test : 80
  environmen t :
    ADDITIONAL _SERVICES : "*"
    STATS_AUTH : " admin : admin " # The logon account and
password used for monitoring , which are customizab le
.
    STATS_PORT : " 1935 " # The port used for monitoring
, which is customizab le .
appone :
  expose :
    - 80 / tcp
  image : ' nginx : latest '
  labels :
    aliyun . proxy . VIRTUAL_HO ST : " 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 @ c68a460635 b8c405e83c 052b7c2057 c7b - 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.

## 12.3 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 [#unique\\_184/unique\\_184\\_Connect\\_42\\_section\\_it1\\_v2z\\_xdb](#) 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 [#unique\\_154](#) 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
  tty : 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\\_154](#) label is 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
  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\\_154](#) label and [#unique\\_186](#) are used):

```
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 subscripti on registry center and dynamical l y load
    the service route .
    aliyun . custom_add on : " 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 , and the lb label is used .
    aliyun . lb . port_6379 : tcp :// proxy_test : 6379
    # Indicates that the custom routing must be
    started after the master Redis and slave Redis are
    started , and the custom routing depends on the master
    Redis and slave Redis .
    aliyun . depends : redis - master , redis - slave
```



```

environment :
  # Indicates the range of backend containers that
  support route loading. "*" indicates the whole cluster
  . By default, it indicates the services in applicatio
ns .
  ADDITIONAL_SERVICES : "*"
  EXTRA_DEFAULT_SETTINGS : " log rsyslog local0 , log
global , option httplog "
  # Configures HAProxy to work in TCP mode .
  MODE : " tcp "
  links :
    - rsyslog : rsyslog
  rsyslog :
    image : registry.cn-hangzhou.aliyuncs.com/linhuatest/
rsyslog : latest
  redis - master :
    ports :
      - 6379 / tcp
    image : ' redis : alpine '
    labels :
      # Indicates that the custom routing is to
  expose the port 6379 .
      aliyun.proxy.TCP_PORTS : " 6379 "
      # Indicates that the service route is to be
  added to the custom routing .
      aliyun.proxy.required : " true "
  redis - slave :
    ports :
      - 6379 / tcp
    image : ' redis : alpine '
    links :
      - redis - master
    labels :
      # Indicates that the custom routing is to expose
  the port 6379 .
      aliyun.proxy.TCP_PORTS : " 6379 "
      # Indicates that the service route is to be
  added to the custom routing .
      aliyun.proxy.required : " true "
      # Indicates that the slave Redis depends on the
  master Redis and must be started after the master
  Redis is started .
      aliyun.depends : redis - master
      command : redis - server -- slaveof redis - master 6379
  redis - client :
    image : ' redis : alpine '
    links :
      - lb : www.example.com
    labels :
      aliyun.depends : lb
      command : redis - cli -h www.example.com
      stdin_open : true
      tty : true

```

This solution provides a master-slave Redis architecture and balances load by using the [#unique\\_186/unique\\_186\\_Connect\\_42\\_section\\_wqf\\_v3z\\_xdb](#) to make Container Server become highly available.

## 12.4 Custom routing - supports multiple HTTPS certificates

Use the acs/proxy image [#unique\\_184/unique\\_184\\_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.

```
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 subscripti on registry center and dynamical l y load
    the service route .
    aliyun . custom_add on : " 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
  environmen t :
    # Indicates the range of backend containers that
    support route loading . "*" indicates the whole cluster
    . By default , it indicates the services in applicatio
    ns .
    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 configurin g URLs
      . In this example , http / https / ws / wss are supported .
      aliyun . proxy . VIRTUAL_HO ST : " https :// appone .
    example . com "
    # Configure the appone certificat e .
    aliyun . proxy . SSL_CERT : "----- BEGIN  RSA  PRIVATE
    KEY -----\ nMIIePQIBA AKCAQEAvgn KhephWHKWY DEiBiSjzst
    7nRP0DJxZ5 cIOxyXmncd 2kslr \ nkUIB5qT / MSiJGBL3Lr 4advS6kI /
    JFmxloFrPt wEe2FGkLBf CDXXDrWgxy FhbuPQY \ nBLNueUu94 sffIXg
    + 4u5Mriui7f tind0Af0d2 1PSM9gb / ZUypxIgAd3 RHCE / gtT0h \
    nVCn6FikXy nXLDToDYWC thQHBwSZS8 8HNU + B0T9Yl65Ji Q0mV +
    YF + h3D / c232E6Gp \ nzK + 8ehVB13s5h ecUx3dvdUQ PBUhJYvzsP
    jChgsXSMDR exiN66kbhH 6dJArsrYb8 t \ nEBWxFCZaT cF82wkAsUe /
    fhLGhh97h + 66lh60QQID AQABAoIBA Q C4d8ifNWRI 9vIB \ nbbAZRne7x
    Mm5MCU2GI8 q97Rgm + nAPL5bHinM VsaBnKgaj7 6EH + TQ + relxyiSKwC H
    \ nQ7FidsQqY GwQjy9NncJ ATpAjQ4EPe LWQU2D9Ly + NjnhEKr / u0Ro6LhdA
    + hqt59dS \ nXHvfEP / It5odN62yJ zikDWBmk / hhK0tu28dP YUuPoWswXW
```

```

FMkaNttmFL gZlagiqr \ nYp7rxAFqQ urzctQ2VNw ezekDHQoh8 ounHGEniZ
+ fA6sFtYi83 KTKWkvFom1 chZQr \ nxxPbbgANJ JJlNgtkl6J ZNxj6SYimm
WvzmrrU25k hKg / klP5EtQzIx 6UFhURnuTK u \ nzNgqcIABA oGBAOqUOer
veEUePvsAl ta8CV / p2KKwenv + kUofQ4UpKF XfnHbQHqfr \ nZHS290QiP
xqjVXYLu8g NfLRfKtUNy qV + TDrzJ1elW2 RKc00GHAwP bXxijPhmJ2
fW \ neskn8tLDc yXpvoqWJG3 4896vo4Ibc L0H / eUs0jJo60J
lCQBKXik + t3gxAoGBAM 9k \ nVOTV2caKy rZ4ta0Q1LK qKf0kt0j +
vKz167J5pS LjVKQSUxGM yLnGwiQdDt B4iy6L \ nFcCB / S0HM0UWkJW
hNYAL8kHry 53bVdHtQG0 tuYFYvBJo7 A + Nppsn9MtlV h8KbVu4 \
nhOz / 3MwWbQNNvI VCGK / fSlts1GhTk 4rKL7PjNwM RAoGBALK0n
3bqXj6Rrzs 7FK6c \ na6vLE4PFX Fpv8jF8pcy hMThSdPlSz HsHCE2cn +
3YZSIE +/ FFORZLqBAI XBuzP6Na \ nFyrlqLgto fVCfppUKDP L4QXccjaeZ
DDIBZyPUYP Qzb05WE5t2 WzqNqcUOUV aMEXh \ n + 7uGrM94esp
WXEgbX6aeP 9lRAoGARlJ Q7t8MXuQE5 GZ9w9cnKAX G / 9RkSZ4Gv +
cL \ nKpNQyUmoE 5IbFKJWFZg tkC1CLrIRD 5EdqQ7ql / APFGgYUoQ9
LdPfKzcW7c nHic0W \ nW51rkQ2U U ++ a2 + uhiHB4Y3U6 + WP00CP4gTI
CUhPT05IQC 8vS8M85UZq u41LRA5W \ nqnpq1uECg YEAq + 6KpHhLr +
5h3Y / m0n84yJ0Yu Cmrl7HFRzB MdOcaW3oaY L83rAaq \ n6dJqpAVge
u3HP8AtiGV ZRE78J + n4d2JGYsqg tP2lFFTdF9 HfhcR2P9bU BNYtWols
\ nEs3iw53t8 a4BndLGBwL PA3lklf7J5 stYanRv6Nq aRaLq4FQMx
sw1A0Q =\ n ----- END RSA PRIVATE KEY -----\ n ----- BEGIN
CERTIFICAT E -----\ nMIIDvDCCA qSgAwIBAgI BATANBgkqh kiG9w0BAQU
FADBGMQswC QYDVQQGEWJ DTjER \ nMA8GA1UEC BMIWmhlaml hbmcxETAPB
gNVBAcTEh hbmd6aG91M RQwEgYDVQQ KEwth \ nbGliYWJhL mNvbTEVMBM
GA1UECMMd 3d3LnJvb3Q uY29tMB4XD TE1MDIwOTA 1MzQx \ nOFoXDTE2M
DIwOTA1MzQ xOFowZjELM AKGA1UEBhM CQ04xETAPB gNVBAGTCFp oZWpp
\ nYw5nMRQwE gYDVQQKEwt hbGliYWJhL mNvbTEVMBM GA1UECMMd
3d3LnJvb3Q uY29t \ nMRcwFQYDV QQDEw53d3c ubGluaHVhL mNvbTCCASI
wDQYJKoZIh vcNAQEBBQA DggEP \ nADCCAQoCg gEBAL4JyoX qYVhYlMAxI
gYko87Le50 T9AycWeXCD scl5p3HdpL Ja5FC \ nAeak / zEoiRgS9y6
+ Gnb70pCPyR ZsZaBaz7cB HthRpCwXwg 1lw6loMchY W7j0GASz \
nbnlLveLH3 yMYPuLuTK4 rou37Yp3Tg H9HdtT0jPY G / 2VMqcSIAHd
0Rwnv4LU9I VQp \ n + hYpF8p1yw0 zg2FgrYUBw cEmUvPBzVP gdE
/ WJeuSYkNJl fmBfodw / 3Nt9h0hqcy v \ nvHoVQdd70 YXnFMd3b3V
EDwVISWL87 D4woYLF0jA 0XsYjeupG4 R + nSQK7K2G / LRAV \ nl3wmWk3Bf
NsJALFHV34 ZRoYfe4fuU pYejkECAwE AAaN7MHkwC QYDVR0TBAl wADAs
\ nBglghkgBh vhCAQ0EHxY dT3BlblNTT CBHw5lcmF 0ZWQgQ2Vyd
GlmaWNhdGU wHQYD \ nVR00BBYEF M6ESmkDKrq nqMwBawkje ONKrrMQMB8
GA1UdIwQYM BaAFFUrhN9 ro + Nm \ nrZnl4WQzD pgTbCBhMA0 GCSqGSIB3D
QEBBQUAA4I BAQCQ2D9CR iv8brx3fnr / RZG6 \ nFYPEdxjY / CyfJrAbij0
PdKjzZKk10 67chM10xs2 JhJ6tMqg2s v50bGx4Xmb SPmEe \ nYTJjIXMY
+ jCoJ / Zmk3Xgu4K1 y1LvD25Pah DVhRrPN8H4 WjsYu51pQN shil5E /
3iQ \ n2JoV0r8Qi AsPiiY5 + mNCD1fm + QN1tyUabcz i / DHafgWJxf2
B3M66e3oUd tbza2pf \ nYHR8RveSF rjaBqud08i r + uYcRbRkroY mY5Vm +
4Yp64oetrP pKUPWSYaAZ 0uRtpEL \ nB5DpqXz9G Ebb5m2Q4dK js5Hm6vyFU
ORCzZc04Xe xDhcgdLOH5 qznmh9oMCK 9QvZf \ n ----- END CERTIFICAT
E -----\ n "
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 configurin g URLs
. In this example , http / https / ws / wss are supported .
    aliyun . proxy . VIRTUAL_HO ST : " https :// apptwo .
example . com "
    # Configure the apptwo certificat e .
    aliyun . proxy . SSL_CERT : "----- BEGIN RSA PRIVATE
KEY -----\ nMIIEpQIBA AKCAQEAvgn KhephWHKWY DEiBiSjzst
7nRP0DJxZ5 cIOxyXmncd 2kslr \ nkUIB5qT / MSiJGBL3Lr 4advs6kI /

```

```

JFmxloFrPt wEe2FGkLBf CDXXDrWgxy FhbuPQY \ nBLNueUu94 sffIxcg
+ 4u5Mriui7f tind0Af0d2 1PSM9gb / ZUypxIgAd3 RHCE / gtT0h \
nVCn6FikXy nXLDtODYWC thQHBwSZS8 8HNU + B0T9Yl65Ji Q0mV +
YF + h3D / c232E6Gp \ nzK + 8ehVB13s5h ecUx3dvdUQ PBUhJYvzsP
jChgsXSMDR exiN66kbhH 6dJArsrYb8 t \ nEBWxfCZaT cF82wkAsUe /
fhLGhh97h + 66lh60QQID AQABAoIBAQ C4d8ifNWRI 9vIB \ nbbAZRne7x
Mm5MCU2GI8 q97Rgm + nAPL5bHinM VsaBnKgaj7 6EH + TQ + relxyiSkWc H
\ nQ7FidsQqY GwQjy9NncJ ATpAjQ4EPe LWQU2D9Ly + NjnhEKr / u0Ro6LhdA
+ hqt59dS \ nXHvfEP / It5odN62yJ zikDWBmk / hhK0tu28dP YUuPoWswXW
FMkaNttmFL gZlagiqr \ nYp7rxAFqQ urzctQ2VNw ezekDHQoh8 ounHGENiZ
+ fA6sFtYi83 KTKWkvFom1 chZQr \ nxxPbbgANJ JjLNgtkl6J ZNxj6SYimm
WvzmrrU25k hKg / klP5EtQzIx 6UFhURnuTK u \ nzNgqcIABA oGBA0qUOer
veEUePvsAl ta8CV / p2KKwenv + kUofQ4UpKF XfnHbQHqfr \ nZHS290QiP
xqjVXYLu8g NfLRFktUNy qV + TDrzJ1elW2 RKc00GHAwP bXxijPhmJ2
fW \ neskn8tldc yXpvoqWJG3 4896vo4Ibc L0H / eUs0jJo60J
lCQBKXik + t3gxAoGBAM 9k \ nVOTV2caKy rZ4ta0Q1LK qKf0kt0j +
vKz167J5pS LjVKQSUxGM yLnGwiQdDt B4iy6L \ nFcCB / S0HM0UwkJW
hNYAL8kHry 53bVdHtQG0 tuYFYvBJo7 A + Nppsn9MtlV h8KbVu4 \
nh0z / 3MWwbQNvI VCGK / fSltS1GhTk 4rKL7PjNwM RAoGBALk0n
3bqXj6Rrzs 7FK6c \ na6vLE4PFX Fpv8jF8pcy hMThSdPlSz HSHCe2cn +
3YZSIE + / FF0RZLqBAL XBUZP6Na \ nFyrlqLgto fVCfppUKDP L4QXccjaeZ
DDIBZyPUYP Qzb05WE5t2 WzqNqcUOUV aMEXh \ n + 7uGrM94esp
WXEgbX6aeP 9lRAoGARlJ Q7t8MXuQE5 GZ9w9cnKAX G / 9RkSZ4Gv +
cL \ nKpNQyUmoE 5IbFKJWFZg tkC1CLrIRD 5EdqQ7ql / APFGgYUoQ9
LdPfKzcW7c nHic0W \ nWw51rkQ2U U ++ a2 + uhiHB4Y3U6 + WPO0CP4gtI
CUHPT05IQC 8vS8M85UZq u41LRA5W \ nqnpqluECg YEAq + 6KpHhLR +
5h3Y / m0n84yJ0Yu Cmrl7HFRzB Md0caW3oaY L83rAaq \ n6dJqpAVge
u3HP8AtiGV ZRe78J + n4d2JGYSqg tP2lFFTdF9 HfhcR2P9bU BNYtWols
\ nEs3iw53t8 a4BndLGBwL PA3lklf7J5 stYanRv6Nq aRaLq4FQMx
sW1A0Q = \ n ----- END RSA PRIVATE KEY ----- \ n ----- BEGIN
CERTIFICAT E ----- \ nMIIDvDCCA qSgAwIBAgI BATANBgkqh kiG9w0BAQU
FADBgMQswC QYDVQQGEwJ DTjER \ nMA8GA1UEC BMIWmhlaml hbmcmxETAPB
gNVBACTCEh hbmd6aG91M RQwEgYDVQQ KEWth \ nbGliYWJhL mNvbTEVMBM
GA1UECxMMd 3d3LnJvb3Q uY29tMB4XD TE1MDIwOTA 1MzQx \ n0FoXDTE2M
DIwOTA1MzQ xOFowZjELM AkGA1UEBhM CQ04xETAPB gNVBAGTCFp oZWpp
\ nYW5nMRQwE gYDVQQKEwt hbGliYWJhL mNvbTEVMBM GA1UECxMMd
3d3LnJvb3Q uY29t \ nMRcwFQYDV QQDEw53d3c ubGluaHVhL mNvbTCCASI
wDQYJKoZIh vcNAQEBBQA DggEP \ nADCCAQoCg gEBAL4JyoX qYVhYlmAXI
gYko87Le50 T9AycWeXCD scl5p3HdpL Ja5FC \ nAeak / zEoiRgS9y6
+ Gnb70pCPyR ZsZaBaz7cB HthRpCwXwg 1lw61oMchY W7j0GASz \
nbnlLveLH3 yMYPuLuTK4 rou37Yp3Tg H9HdtT0jPY G / 2VMqcSIAHd
0Rwnv4LU9I VQp \ n + hYpF8p1yw0 zg2FgrYUBw cEmUvPBzVP gdE
/ WJeuSYkNjL fmBfodw / 3Nt9h0hqcY v \ nvHoVQdd70 YXnFMd3b3V
EDwVISWL87 D4woYLF0jA 0XsYjeupG4 R + nSQK7K2G / LRAV \ nl3wmWk3Bf
NsJALFhv34 ZRoYfe4fuU pYejkECAwE AaAn7MHkwC QYDVR0TBAl wADAs
\ nBglghkgBh vhCAQ0EHxY dT3BlblNTT CBHZW5lcmF 0ZWQgQ2Vyd
GlmaWNhdGU wHQYD \ nVR00BBYEF M6ESmkDKrq nqMwBawkje ONKrrMQMB8
GA1UdIwQYM BaAFFUrhN9 ro + Nm \ nrZnl4WQzD pgTbCBhMA0 GCSqGSIb3D
QEBBQUAA4I BAQCQ2D9CR iv8brx3fnr / RZG6 \ nFYPEdxjY / CyfJrAbij0
PdKjzZKk10 67chM10xs2 JhJ6tMqg2s v50bGx4Xmb SPmEe \ nYTJjIXMY
+ jCoJ / Zmk3Xgu4K1 y1LvD25Pah DVhRrPN8H4 WjsYu51pQN shil5E /
3iQ \ n2JoV0r8Qi AsPiiY5 + mNCD1fm + QN1tyUabcz i / DHafgWJxf2
B3M66e3oUd tbzA2pf \ nYHR8RveSF rjaBqud08i r + uYcRbRkroY mY5Vm +
4Yp64oetrP pKUPWSYaAZ 0uRtpeL \ nB5DpqXz9G Ebb5m2Q4dK js5Hm6vyFU
ORCzZc04Xe xDhcgdLOH5 qznmh9oMck 9QvZf \ n ----- END CERTIFICAT
E ----- \ n "
restart : always

```

Services appone and apptwo use `aliyun . proxy . VIRTUAL_H0 ST` 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 in the bash (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`. Use double quotation marks ( “ ” ) for separation. For other information, such as lb label, [#unique\\_154](#) see the preceding template and the corresponding [#unique\\_186](#).