Alibaba Cloud Aliyun Container for Kubernetes

Best Practices

Issue: 20190613

MORE THAN JUST CLOUD | C-J Alibaba Cloud

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 con	ventions
---------------------	----------

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

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

Contents

Legal disclaimer I
Generic conventionsI
1 Cluster
1.1 ECS instance selection and cluster configurations
1.1.1 Select ECS instances
1.1.2 Recommended Kubernetes cluster configurations to run highly reliable applications
1.2 Update expired certificates of a Kubernetes cluster
1.3 Update the Kubernetes cluster certificates that are about to expire
1.4 Plan Kubernetes CIDR blocks under a VPC
2 Network
2.1 Deploy a highly reliable Ingress controller
3 Storage
3.1 Use a static cloud disk when creating a stateful service
3.2 Use a dynamic cloud disk when creating a stateful service
3.3 Use a StatefulSet service
3.4 Use a NAS file system when creating a stateful service
3.5 Use an OSS bucket when creating a stateful service
4 Release
4.1 Implement Layer-4 canary release by using Alibaba Cloud Server Load
Balancer in a Kubernetes cluster57
Balancer in a Kubernetes cluster
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.65
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.68
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.71
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.76
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingress in a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.79
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.80
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.80
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingress614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.88
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingress614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.805.2 Implement Istio distributed tracking in Kubernetes.885.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes.99
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.885.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes.995.4 Use Istio to deploy application services across Kubernetes and ECS
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingress in a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.885.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes.995.4 Use Istio to deploy application services across Kubernetes and ECS instances.107
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingressin a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.885.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes and ECS995.4 Use Istio to deploy application services across Kubernetes and ECS1075.5 Use the Canary method that uses Istio to deploy a service.115
Balancer in a Kubernetes cluster.574.2 Implement a gray release and a blue/green deployment through Ingress in a Kubernetes cluster.614.2.1 Gray releases and blue/green deployment.614.2.2 Gray release limits.644.2.3 Annotation.654.2.4 Step 1: Deploy a service.684.2.5 Step 2: Release the latest version of a service.714.2.6 Step 3: Remove the earlier version of a service.764.3 Application.795 Istio.805.1 Use Istio to implement intelligent routing in Kubernetes.885.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes.995.4 Use Istio to deploy application services across Kubernetes and ECS instances.107

7 Operation and maintenance142
7.1 Check a Kubernetes cluster to troubleshoot exceptions
8 Serverless
8.1 Use a GPU container instance146
8.2 Associate an EIP address with a pod149
9 Auto Scaling153
9.1 Deploy an Ingress application on a virtual node153
10 DevOps158
10.1 Deploy Jenkins in a Kubernetes cluster and perform a pipeline build 158
10.2 Use GitLab CI to run a GitLab runner and activate a pipleline in a
Kubernetes cluster169
10.3 Deploy Jenkins in a serverless Kubernetes cluster and perform an
application pipeline build180
10.4 Use Bamboo to deploy a remote agent and run a build plan188

1 Cluster

1.1 ECS instance selection and cluster configurations

1.1.1 Select ECS instances

This topic describes the recommend ECS instances for creating a Kubernetes cluster.

Overall cluster ECS instance selection

Low performance ECS instances have the following disadvantages:

- The Worker nodes that run on low performance ECS instances can use only a limited number of network resources.
- If one container consumes most of the resources provided by a low performance ECS instance, the remaining resources become idle because they are insufficient for operations such as creating new containers or restoring failed containers. If you set multiple low performance ECS instances, an excessive amount of resources will be wasted.

High performance ECS instances have the following advantages:

- Large network bandwidth is available. For applications that require large bandwidth, resource usage is high.
- More container communication occurs within one ECS instance, reducing data transmission over networks.
- Images can be more efficiently pulled. For a cluster that uses high performance ECS instances, it only requires one attempt to pull an image and the pulled image then can be used by multiple containers. By contrast, for a cluster that uses low performance ECS instances, multiple attempts must be made to pull an image. Furthermore, scaling a cluster that uses low performance ECS instances takes much longer to perform.

Select the Master node specification

For Kubernetes clusters created through Alibaba Cloud Container Service, core components such as etcd, kube-apiserver, and kube-controller run on Mater nodes. These core components are critical for ensuring cluster stability. Generally, large clusters have higher requirements on the Master node specification.



Note:

You can determine your cluster size by considering the following factors: the number of nodes, the number of pods, deployment frequency, and the number of visits. In this topic, only the number of nodes is used to determine the size of a cluster.

To select the Master node specification of a cluster of the standard size, see the following table. However, you can select the lower performance Master nodes for clusters in a test environment. The specifications recommended in the following table are designed to keep Master node loads low.

Number of nodes	Master node specification
1 to 5	4 cores, 8 GiB (We recommend that you do not select 2 cores with 4 GiB.)
6 to 20	4 cores, 16 GiB
21 to 100	8 cores, 32 GiB
100 to 200	16 cores, 64 GiB

Select the Worker node specification

• Determine the number of cores required by the cluster and the allowed core failure ratio.

For example, assume a cluster has 160 cores in total. If the allowed core failure ratio is 10%, you must select at least ten 16-core ECS instances and ensure that the upper limit of the cluster load is 160*90%=144 cores. If the allowed core failure ratio is 20%, you must select at least five 32-core ECS instances and ensure that the upper limit of the cluster load is 160*80%=128 cores. In either of these two cases, if one ECS instance fails, the remaining ECS instances can still support the cluster services.

• Determine the CPU:memory ratio. If you run applications that consume large amount of memory resource, for example, Java applications, we recommend that you select an ECS instance with a CPU:memory ratio of 1:8.

Select the ECS Bare Metal Instance

We recommend that you select an ECS Bare Metal (EBM) Instance in the following two scenarios:

- You cluster requires 1000 cores for daily operation. In this case, you can use about ten or eleven EBM instances to build your cluster because one EBM instance has a minimum of 96 cores.
- You want to quickly scale out a large number of containers. For example, assume that you are prepared for a popular E-commerce product promotion. To handle the expected large amount of traffic, you can add EBM instances to your cluster because a single EBM instance can run multiple containers.

EBM instances provide the following benefits to your cluster:

- Ultra-high network performance. Remote Direct Memory Access (RDMA) technology is used. Furthermore, the Terway plugin is designed for you to get the most from your hardware and provides a container bandwidth higher than 9 Gbit/s across hosts.
- Zero jitter computing performance. EBM instances use chips developed by Alibaba Cloud to replace Hypervisor, meaning virtualization overhead or resource preemption concerns are no longer issues.
- High security. EBM instances use physical level encryption, support the Intel SGX encryption, provide a reliable computing environment, and support blockchain applications.

1.1.2 Recommended Kubernetes cluster configurations to run highly reliable applications

To help you guarantee that your applications stably and reliably run in Kubernetes, this topic describes the recommended Kubernetes cluster configurations.

Set the disk type and size

Select the disk type

- We recommend that you select the SSD disk type.
- For Worker nodes, we recommend that you select the Attach Data Disk check box when you create a cluster. This disk is provided exclusively for the / var / lib / docker file to store local images. It is designed to allow the root disk to store a massive number of images. After your cluster has run for a period, many images you no longer require remain stored. To quickly solve this, we recommend that you take the machine offline, rebuild this disk, and then bring the machine back online.

Set the disk size

Kubernetes nodes require a large disk space because the Docker images, system logs , and application logs are stored in the disk. When creating a Kubernetes cluster, you need to consider the number of pods on each node, the log size of each pod, the image size, the temporary data size, and the space required for system reserved values.

We recommend that you reserve a space of 8 GiB for the ECS instance operation system because the operation system requires a disk space of at least 3 GiB. Kubernetes resource objects then use the remaining disk space.

Whether to build Worker nodes when creating your cluster

When you create a cluster, you can select either of the following Node Type:

- Pay-As-You-Go, indicates that you can build Worker nodes when creating a cluster.
- Subscription, indicates that you can purchase ECS instances as needed and add the instances to your cluster after you create you cluster.

Configure your cluster network settings

- If you want to connect your cluster with services outside Kubernetes, for example , Relational Database Service (RDS), we recommend that you use an existing VPC , rather than create a VPC. This is because VPCs are logically isolated. You can create a VSwitch and add the ECS instances that run Kubernetes to the VSwitch.
- You can select the Terway network plugin or the Flannel network plugin when creating a Kubernetes cluster. For more information, see *Do I select the Terway or Flannel plugin for my Kubernetes cluster network*?.
- We recommend that you do not set a small CIDR block of the pod network that only supports a minimal number of nodes. The CIDR block setting of the pod network is associated with the Pod Number for Node setting in Advanced Config. For example, if you set the CIDR block of the pod network to X.X.X.X/16, it means that the number of IP addresses assigned to your cluster is 256*256. Additionally, if you set the number of pods on each node to 128, it means that the maximum number of nodes supported by your cluster is 512.

Use multiple zones

Alibaba Cloud supports multiple regions and each region supports multiple zones. Zones are physical areas that have independent power grids and networks within a region. Using multiple zones enables disaster recovery across areas, but increases network latency. When creating a Kubernetes cluster, you can choose to create a multi-zone cluster. For more information, see *Create a multi-zone Kubernetes cluster*.

Claim resources for each pod

When you use a Kubernetes cluster, a common problem is that too many pods are scheduled to one node. This scheduling of pods overloads the node, making it unable to provide services.

We recommend that you specify the resource request parameter and the resource limit parameter when configuring a pod in Kubernetes. This recommended configuration enables Kubernetes to select a node with sufficient resources according to the pod resource requirements during the pod deployment. The following example claims that the Nginx pod uses 1-core CPU and 1024 MiB memory, and the pod cannot use more than 2-core CPU or 4096 MiB memory.

```
apiVersion :
                 v1
kind : Pod
metadata :
           nginx
  name :
spec :
  containers :
    name : nginx
image : nginx
     Resources : #
                                      claim .
                       Resource
       requests :
          memory : " 1024Mi "
cpu : " 1000m "
       limits :
          memory : " 4096Mi "
cpu : " 2000m "
```

Kubernetes uses a static resource scheduling method, which means that instead of using the resources that have been used to calculate the remaining resources on each node, it uses allocated resources. Its calculation method is: the remaining

resources = the total resources - the resources that have been allocated. If you manually run a resource-consuming program, Kubernetes is not aware of the resources that are being used by the program.

Therefore, you must claim resources for all pods. For the pods that do not have resource claims, after they are scheduled to a node, Kubernetes assumes that the resources used by them on the corresponding node are still available. Therefore, too many pods may be scheduled to this node.

Configure cluster operation and maintenance settings

• Enable Log Service

When creating a cluster, select the Using Log Service check box.

· Configure cluster monitoring

Alibaba Cloud Container Service is integrated with CloudMonitor. By configurin g monitoring on nodes, you can implement real-time monitoring. By adding monitoring alarm rules, you can quickly locate the issues that cause abnormal resource usage.

When you create a Kubernetes cluster through Container Service, two application groups are automatically created in CloudMonitor: one for Master nodes and one for Worker nodes. You can add alarm rules under these two groups and these rules apply to all machines in the groups. When subsequent nodes are added to the corresponding group, the alarm rules in the group are automatically applied.

This means that you only need to configure alarm rules for the ECS resources.



 To monitor ECS instances, you need to set alarm rules for resources such as CPU, memory, and disk. We recommend that you set the / var / lib / docker file on an exclusive disk.

Set an application to wait for its dependent application after it starts

Some applications may have some external dependencies. For example, an application may need to read data from a database (DB) or access the interface of another service. However, when the application starts, the DB or the interface may not be available. In traditional manual O&M, if the external dependencies of an application are unavailable when the application starts, the application exits directly. This is known as failfast . This strategy is not applicable for Kubernetes, because O&M in Kubernetes is automated and does not require manual intervention. For example, when you deploy an application, you do not need to manually select a node or start the application on the node. If the application fails, Kubernetes

automatically restarts it. Additionally, automatic capacity increase is supported through HPA when large loads occur.

For example, assume that application A depends on application B, and these two applications run on the same node. After the node restarts, application A starts, but application B has not started. In this case, the dependency of application A is unavailable. According to the strategy of failfast, application A exists and will not start even after application B starts. In this case, application A must be started manually.

In Kubernetes, you can set the system to check the dependency of the application during startup, and to implement polling to wait until the dependency is available. This can be implemented through*lnit Container*.

Set the pod restart policy

When a bug in the code or excessive memory consumption causes application processes to fail, the pod in which the processes reside also fails. We recommend that you set a restart policy for the pod so that the pod can automatically restart after failure.

```
apiVersion : v1
kind : Pod
metadata :
    name : tomcat
spec :
    containers :
        name : tomcat
        image : tomcat
        restartPol icy : OnFailure #
```

Available values of the restart policy parameter are:

- Always : indicates to always restart the pod automatically.
- OnFailure : indicates to automatically restart the pod when the pod fails (the exiting status of the process is not 0).
- Never : indicates to never restart the pod.

Configure the liveness probe and readiness probe

A running pod may not necessarily be able to provide services because processes in the pod may be locked. However, Kubernetes does not automatically restart the pod because the pod is still running. Therefore, you must configure the liveness probe in each pod to determine whether the pod is alive, and whether it can provide services. Then, Kubernetes restarts the pod when the liveness probe detects any exception.

The readiness probe is used to detect whether the pod is ready to provide services. It takes some time for an application to initialize during startup. During the initializa tion, the application cannot provide services. The readiness probe can determine when the pod is ready to receive traffic from Ingress or Service. When the pod is faulty, the readiness probe stops new traffic being forwarded to the pod.

```
apiVersion :
              v1
kind : Pod
metadata :
         tomcat
  name :
spec :
  containers :
   name : tomcat
    image : tomcat
    livenessPr obe :
      httpGet :
        path : / index . jsp
        port : 8080
      initialDel aySeconds :
                               3
      periodSeco nds : 3
    readinessP robe :
      httpGet :
        path : / index . jsp
        port : 8080
```

Set one process to run in each container

Users who are new to the container technology tend to use containers as virtual machines and put multiple processes into one container, such as monitoring process, log process, sshd process, and even the whole systemd. This causes the following two problems:

- It becomes complex to determine the resource usage of the pod as a whole, and it becomes difficult for the resource limit that you set to take effect.
- If only one process runs in a container, the container engine can detect process failures and it restarts the container upon each process failure. However, if multiple processes are put into a container, the container engine cannot determine the failure of any single process. Therefore, the engine does not restart the container when a single process fails even though the container does not work normally.

If you want to run multiple processes simultaneously, Kubernetes can help you easily implement that. For example, nginx and php-fpm communicate with each other through a Unix domain socket. You can use a pod that contains two containers, and put the Unix socket into a shared volume of the two containers.

Avoid Single Point of Failure (SPOF)

If an application uses only one ECS instance, the application is unavailable during the period when Kubernetes restarts the instance upon an instance failure. This issue also occurs when you release an updated version of the application. Therefore , we recommend that you do not directly use pods in Kubernetes. Instead, deploy Deployment or StatefulSet applications and set more than two pods for each application.

1.2 Update expired certificates of a Kubernetes cluster

When cluster certificates expire, communication with the cluster API server by using kubectl or calling APIs is disabled, and the expired certificates on cluster nodes cannot be updated automatically through template deployment. To update the certificates, you can log on to each cluster node and run the container stating commands, docker run.

Update the expired certificates on a Master node

- 1. Log on to a Master node with the root permission.
- 2. Run the following command in any directory to update the expired certificates on the Master node:

```
$ docker run - it -- privileged = true - v /:/ alicoud - k8s
- host -- pid host -- net host \
registry . cn - hangzhou . aliyuncs . com / acs / cert - rotate :
v1 . 0 . 0 / renew / upgrade - k8s . sh -- role master
```

3. Repeat the preceding steps on each cluster Master node to update all the expired certificates.

Update the expired certificates on a Worker node

1. Log on to a Master node with the root permission.

2. Run the following command to obtain the cluster rootCA private key:

```
$ cat / etc / kubernetes / pki / ca . key
```

- 3. Run either of the following commands to obtain the cluster root private key encoded through base64:
 - If the cluster rootCA private key has a blank line, run the following command:

\$ sed ' 1d ' / etc / kubernetes / pki / ca . key | base64 - w
0

• If the cluster rootCA private key does not have any blank line, run the following command:

\$ cat / etc / kubernetes / pki / ca . key | base64 - w 0

- 4. Log on to a Worker node with the root permission.
- 5. Run the following command in any directory to update the expired certificates on the Worker node.

Note:

In step 3, you have obtained \${base64CAKey}, which is the cluster root private key encoded through base64.

6. Repeat the preceding steps on each cluster Worker node to update all the expired certificates.

1.3 Update the Kubernetes cluster certificates that are about to expire

This topic describes how to update the Kubernetes cluster certificates that are about to expire. You can use one of three methods to update the cluster certificates. You can update the cluster certificates in the Container Service console, update all the certificates by running a single command, or update Master and Worker node certificates separately by running different commands.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Kubernetes cluster through kubectl. For more information, see *Connect to a Kubernetes cluster by using kubectl*.

Updates all certificates through the Container Service console

In the Container Service console, click the Update Certificate prompt of the target cluster. For more information, see *Update the Kubernetes cluster certificates that are about to expire*.

Run a command to update all certificates

Log on to a Master node and run the following command:

```
$ curl http://aliacs - k8s - cn - hangzhou . oss - cn - hangzhou
. aliyuncs . com / public / cert - update / renew . sh | bash
```

Verify the results

1. Run the following command to view the status of Master nodes and Worker nodes:

[root@	<pre>~]# kubectl get</pre>	nodes		
NAME	STATUŠ	ROLES	AGE	VERSION
cn-hangzhou.	Ready	<none></none>	23d	v1.11.2
cn-hangzhou.	Ready	<none></none>	23d	v1.11.2
cn-hangzhou.	Ready	master	47d	v1.11.2
cn-hangzhou.	Ready	master	47d	v1.11.2
cn-hangzhou.	Ready	master	47d	v1.11.2
cn-hangzhou.	Ready	<none></none>	47d	v1.11.2
cn-hangzhou.	Ready	<none></none>	47d	v1.11.2
[root@	~]#			

\$ kubectl get nodes

2. Run the following command. When the value of the SUCCESSFUL parameter of each Master node is 1, and the value of the SUCCESSFUL parameter of

each Worker node meets the number of cluster Worker nodes, all certificates are updated.

\$ kubectl get job – nkube – system

[root@	~]# kube	ctl get job	-nkube-system
NAME	DESIRED	SUCCESSFUL	AGE
aliyun-cert-renew-master-1	1	1	6m
aliyun-cert-renew-master-2	1	1	5m
aliyun-cert-renew-master-3	1	1	5m
aliyun-cert-renew-worker	4	4	4m
cert-job-2	1	1	22h
cert-job-3	1	1	22h
cert-job-4	1	1	22h
cert-node-2	4	4	19h

Manually update the certificates of each Master node

1. Copy the following code and paste it into any path to create a *job* - *master*.

```
yml file:
```

```
apiVersion : batch / v1
kind : Job
metadata :
  name : ${ jobname }
  namespace : kube - system
spec :
  backoffLim it : 0
  completion s : 1
  parallelis m : 1
  template :
    spec :
      activeDead lineSecond s: 3600
      affinity :
        nodeAffini ty :
           requiredDu ringSchedu lingIgnore dDuringExe cution :
             nodeSelect orTerms :
            - matchExpre ssions :
                 key : kubernetes . io / hostname
                 operator : In
                 values :
                - ${ hostname }
      containers :
        command :
        - / renew / upgrade - k8s . sh
        - -- role
        - master
       image : registry . cn - hangzhou . aliyuncs . com / acs /
rotate : v1 . 0 . 0
imagePullP olicy : Always
cert -
        name : ${ jobname }
        securityCo ntext :
           privileged : true
        volumeMoun ts :
  mountPath : / alicoud - k8s - host
      name : ${ jobname }
hostNetwor k : true
```

```
hostPID : true
restartPol icy : Never
schedulerN ame : default - scheduler
securityCo ntext : {}
toleration s :
- effect : NoSchedule
key : node - role . kubernetes . io / master
volumes :
- hostPath :
    path : /
    type : Directory
```

name : \${ jobname }

- 2. Obtain the number of Master nodes in the cluster and the hostname of each Master node.
 - · Method 1

Run the following commands:

\$ kubectl get nodes

[root@]# kubectl get	nodes		
NAME	STATUS	ROLES	AGE	VERSION
cn-hangzhou.i	Ready	<none></none>	22d	v1.11.2
cn-hangzhou.i	Ready	<none></none>	22d	v1.11.2
cn-hangzhou.i	Ready	master	46d	v1.11.2
cn-hangzhou.i	Ready	master	46d	v1.11.2
cn-hangzhou.i	Ready	master	46d	v1.11.2
cn-hangzhou.i	Ready	<none></none>	46d	v1.11.2
cn-hangzhou.i	Ready	<none></none>	46d	v1.11.2
[root@~~]#			

- Method 2
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane under Kubernetes, click Clusters.

Container Service - Kubernetes +		Cluster List			You can creat	e up to 5 clust	ers and can ad	d up to 40 nodes in each	cluster. Refresh	Create Kubernetes Cluster
Overview		Help: & Create cluster & Create GPU creation failures & Authorization manage	clusters 🔗 Scale clust ement	er 🔗 Connect to I	Kubernetes cluster	via kubectl	🖉 Manage app	lications with commands	Oluster planning	${\mathscr S}$ Troubleshoot cluster
Clusters Clusters	l	Name V								
Nodes	l	Cluster Name/ID	Cluster Type	Region (All) +	Network Type	Cluster Status	Number of Nodes	Time Created	Kubernetes Version	Action
Volumes Namespace		test-Terway	Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp1nr6ohb0c	Running	6	11/20/2018,17:27:07	1.11.2	Manage View Logs Dashboard Scale Cluster More -
 Application Deployment 		k8s-managed-cluster	ManagedKubernetes	China East 1 (Hangzhou)	VPC vpc- bp1kd7yn4qn	Running	3	11/01/2018,11:21:13	1.11.2	Manage View Logs Dashboard Scale Cluster More -
StatefulSet Job		test-mia	Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp1lkyevdjj	Running	7	09/17/2018,11:37:55	1.11.2	Manage View Logs Dashboard Scale Cluster More -
CronJob										

c. Click the target cluster name, and then click Node List in the left-side navigation pane to view the number of Master nodes and the hostname of each Master node.

<	Node List			Refresh Label	Management Scal	e Cluster Add Existing Instance
Basic Information	Help: 🖉 Postpay instance to Prepay 🛛 & Node ex	ception 🔗 Node monitoring and alarms		is information		
Node List	Clusters test-mia v Filter by	abels -				
Event List	IP Address Role Instance ID/Name	Configuration Pods(Allocated) C	CPU(Request Limit) M	lemory(Request Limit)	Update Time	Action
	Worker	Pay-As-You-Go 25 1 ecs.n1.large 25 1	13.68%	2.60%	09/17/2018,11:49:00	Scheduling Settings Monitor More+
	Master	Pay-As-You-Go 8 2 ecs.n1.large 8 2	24.05%	.11% 90.27 %	09/17/2018,11:39:00	Scheduling Settings Monitor More+
	Master	Pay-As-You-Go ecs.n1.large 13 2	29.05%	.90% 89.03 %	09/17/2018,11:38:00	Scheduling Settings Monitor More+
Ξ	Master	Pay-As-You-Go 9 2 ecs.n1.large 2	25.30% 6.53 % 6.	.39% 89.57 %	09/17/2018,11:41:00	Scheduling Settings Monitor More+
	Worker	Pay-As-You-Go 27 1 ecs.n1.large 1	17.40% 📥 🛌 11.30 % 2	3.73%	09/17/2018,11:49:00	Scheduling Settings Monitor More+
	Worker	Pay-As-You-Go 17 4 ecs.e3.medium	46.10%	1.24%	10/11/2018,09:48:00	Scheduling Settings Monitor More-
	Worker	Pay-As-You-Go ecs.gn5i- 11 6 c2g1.large	58.10%	2.22%	10/11/2018,09:42:00	Scheduling Settings Monitor More-

3. Run the following command to specify the \${ jobname } and \${ hostname }
variables in the job - master . yml file:

\$ sed 's /\${ jobname }/ cert - job - 2 / g ; s /\${ hostname }/ hostname / g ' job - master . yml > job - master2 . yml

In this code line:

- \${ jobname } is the Job and pod name. In this example, this variable is set to
 cert job 2.
- \${ hostname } is the Master name. In this example, hostname is set to a Master name obtained in step 2.
- 4. Run the following command to create a Job:

\$ kubectl create - f job - master2 . yml

- 5. Run the following command to view the Job status. When the value of the SUCCESSFUL parameter is 1 , the certificates of this Master node have been updated.
 - \$ kubectl get job nkube system
- 6. Repeat step 3 to step 5 to update the certificates of the remaining Master nodes in the cluster.

[root@		~]#	kubectl	get	job	-nkube-system
NAME	DESIRED	SUCCESSFUL	AGE			
cert-job-2	1	1	22m			
cert-job-3	1	1	2m			
cert-job-4	1	1	1m			
[root@		~]#				

file:

Manually update Worker node certificates

1. Copy the following code and paste it into any path to create a job - node . yml

```
apiVersion :
              batch / v1
kind : Job
metadata :
  name : ${ jobname }
  namespace : kube - system
spec :
  backoffLim it : 0
  completion s : ${ nodesize }
  parallelis m : ${ nodesize }
  template :
    spec :
      activeDead lineSecond s :
                                    3600
      affinity :
        podAntiAff inity :
          requiredDu ringSchedu lingIgnore dDuringExe cution :
            labelSelec tor :
              matchExpre ssions :
                key : job - name
                operator : In
                values :
               - ${ jobname }
            topologyKe y : kubernetes . io / hostname
      containers :
       command :
       - / renew / upgrade - k8s . sh
       - -- role
         node
       - -- rootkey
       - ${ key }
        image : registry . cn - hangzhou . aliyuncs . com / acs /
       rotate : v1 . 0 . 0
imagePullP olicy :
cert -
                             Always
        name : ${ jobname }
        securityCo ntext :
          privileged : true
        volumeMoun ts :
- mountPath : / alicoud - k8s - host
          name : ${ jobname }
      hostNetwor
                 k: true
      hostPID : true
      restartPol icy : Never
      schedulerN ame : default - scheduler
      securityCo ntext : {}
      volumes :
        hostPath :
          path : /
type : Directory
        name : ${ jobname }
```

Note:

If a Worker node has a taint, you need to add toleration s for the taint in the *job - node*. *yml* file. More specifically, you need to add the following code

between securityCo ntext : {} and volumes : (If the number of Worker nodes that have taints is n, you need to add the following code n times):

```
toleration s :
- effect : NoSchedule
  key : ${ key }
  operator : Equal
  value : ${ value }
```

The method to obtain \${ name } and \${ value } is as follows:

a. Copy the following code and paste it into any path to create a taint . tml

file:

b. Run the following command to view the values of \${ name } and \${ value }

for the Worker nodes that have taints:

\$ kubectl get nodes - o go - template - file =" taint .
 tml "

[root@	~]# kubectl get nodes -o go-template-file="taint.tml"
Node	Taint
cn-hangzhou.i-	key1=value1:NoSchedule
cn-hangzhou.i-	<pre>node-role.kubernetes.io/master=<no value="">:NoSchedule</no></pre>
cn-hangzhou.i-	<pre>node-role.kubernetes.io/master=<no value="">:NoSchedule</no></pre>
cn-hangzhou.i-	<pre>node-role.kubernetes.io/master=<no value="">:NoSchedule</no></pre>

2. Run the following command to obtain the cluster CAKey:

 $\$ sed $\$ ' 1d $\$ / etc / kubernetes / pki / ca . key \mid base64 - w $_0$

3. Run the following command to specify the \${ jobname }, \${ nodesize }, and \${

```
key } variables in the job - node . yml file:
```

\$ sed ' s /\${ jobname }/ cert - node - 2 / g ; s /\${ nodesize }/
nodesize / g ; s /\${ key }/ key / g ' job - node . yml > job node2 . yml

In this code line:

- \${ jobname } is the Job and pod name. In this example, this variable is set to
 cert node 2.
- \${ nodesize } is the number of Worker nodes. For how to obtain this value, see step 2 in Manually update the certificates of each Master node. In this example, the nodesize variable is replaced with the number of the Worker nodes in the cluster.
- \${ key } is the cluster CAKey. In this example, the key variable is replaced with the CAKey obtained in step 3 of *Manually update Worker node certificates*.
- 4. Run the following command to create a Job:

\$ kubectl create - f job - node2 . yml

5. Run the following command to view the Job status. When the value of the SUCCESSFUL parameter is equal to the number of the cluster Worker nodes, all certificates have been updated.

get job - nkube - system

[root@		~]# kı	ubectl get	job	-nkube-system
NAME	DESIRED	SUCCESSFUL	AGE		
cert-job-2	1	1	1h		
cert-job-3	1	1	47m		
cert-job-4	1	1	46m		
cert-node-2	4	4	1m		
[root@		~]#			

1.4 Plan Kubernetes CIDR blocks under a 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

\$

kubectl

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 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

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. 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 10.0.0/8.

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. pod address of Kubernetes clusters in IDC, configure the route table to leased line virtual border router (VBR) in IDC.

2 Network

2.1 Deploy a highly reliable Ingress controller

Ingress is a set of rules that authorize external access to the services in a Kubernetes cluster, providing Layer-7 Server Load Balancer capabilities. You can configure Ingress to provide externally accessible URLs, SLB, SSL, and name-based virtual hosts. Ingress requires high reliability because Ingress functions as the access layer through which external traffic goes into a cluster. This topic describes how to deploy a high-performance, highly reliable Ingress access layer.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Master node by using SSH. For more information, see Access Kubernetes clusters by using SSH.

Highly reliable deployment architecture

To achieve high reliability, you must first resolve any SPOFs. Deploying multiple replicas is the general solution for this problem. Specifically, you can use the multinode deployment architecture to deploy a highly reliable Ingress access layer in a Kubernetes cluster. We also recommend that you configure exclusive Ingress nodes to prevent service applications from competing for resources with the Ingress service because Ingress functions as the traffic access port of a cluster.



As shown in the preceding figure, multiple exclusive Ingress instances constitute an access layer that processes the inbound traffic to the cluster. Furthermore, the number of Ingress nodes can be scaled according to the traffic amount required by the backend services. If your cluster is of a moderate size, you can also deploy the Ingress service and other service applications in a hybrid way. However, we recommend that you limit the number of resources and isolate them for the Ingress and corresponding applications.

View the cluster pod replicas deployed by default and the Internet SLB address

After you create a cluster, a set of Nginx Ingress controller services that have two pod replicas are deployed within the cluster by default. The frontend of this set of services is mounted to an Internet SLB instance.

Run the following command to view the pods on which the Nginx Ingress controller services are deployed:

```
$ kubectl - n kube - system get pod | grep nginx - ingress
- controller
nginx - ingress - controller - 8648ddc696 - 2bshk
1 / 1 Running 0 3h
```

```
nginx - ingress - controller - 8648ddc696 - jvbs9
1 / 1 Running 0 3h
```

Run the following command to view the Internet SLB address corresponding to the nginx-ingress-lb service:

```
$ kubectl - n
                 kube - system
                                get
                                           nginx - ingress - lb
                                      SVC
NAME
                    TYPE
                                     CLUSTER - IP
                                                      EXTERNAL - IP
       PORT (S)
                                       AGE
nginx - ingress - lb
                       LoadBalanc er
                                          172 . xx . x . xx
                                                                118
 . xxx . xxx . xx 80 : 32457 / TCP , 443 : 31370 / TCP
                                                            21d
```

To guarantee the high performance and availability of the cluster access layer for a growing cluster, you need to expand the Ingress access layer. You can use either of the following two methods:

Method 1: Expand the number of replicas

You can quickly scale the Ingress access layer by changing the number of the replicas of the Nginx Ingress controller deployment.

Run the following command to scale out the number of pod replicas to three:

```
$ kubectl - n kube - system scale -- replicas = 3 deployment
/ nginx - ingress - controller
deployment . extensions / nginx - ingress - controller scaled
```

Run the following command to view the pods on which the Nginx Ingress controller services are deployed:

```
$ kubectl - n
                 kube - system
                                 get
                                               grep
                                                      nginx - ingress
                                       pod
- controller
nginx - ingress - controller - 8648ddc696 - 2bshk
  1 / 1
              Running
                                       3h
                         0
nginx - ingress - controller - 8648ddc696 - jvbs9
  1 / 1
              Running 0
                                       3h
nginx - ingress - controller - 8648ddc696 - xqmfn
  1 / 1
              Running
                          0
                                       33s
```

Method 2: Deploy the Ingress service on a specified node

If you want the Nginx Ingress controller to run on target nodes of advanced configurat ions only, you can label the target nodes.

1. Run the following command to view the cluster nodes:

```
$
  kubectl
            get
                   node
NAME
                                        STATUS
                                                    ROLES
                                                               AGE
  VERSION
cn - hangzhou . i - bp11bcmsna 8d4bpf17bc
                                                Ready
                                                            master
  21d
          v1 . 11 . 5
cn - hangzhou . i - bp12h6biv9
                                 bg24lmdc2o
                                                 Ready
                                                           < none >
  21d
        v1 . 11 . 5
```

cn – hangzhou . i – bp12h6biv9 21d v1 . 11 . 5	bg24lmdc2p	Ready	< none >
cn – hangzhou . i – bp12h6biv9	bg24lmdc2q	Ready	< none >
21d v1 . 11 . 5 cn – hangzhou . i – bp181pofzy	yksie2ow03	Ready	master
21d v1 . 11 . 5 cn - hangzhou . i - bp1cbsg6rf	3580z6uyo7	Ready	master
21d v1 . 11 . 5			

2. Run the following commands to add the label node - role . kubernetes . io

/ ingress =" true " to the Ingress node cn - hangzhou . i - bp12h6biv9
bg24lmdc2o and the Ingress node cn - hangzhou . i - bp12h6biv9

bg24lmdc2p :

Note:

- The number of the labeled nodes must be greater than or equal to the number of the cluster pod replicas so that multiple pods do not run on one node.
- We recommend that you label Worker nodes only to deploy the Ingress service.

```
$ kubectl label nodes cn – hangzhou .i – bp12h6biv9
bg24lmdc2o node – role .kubernetes .io / ingress =" true "
node / cn – hangzhou .i – bp12h6biv9 bg24lmdc2o labeled
```

```
$ kubectl label nodes cn - hangzhou . i - bp12h6biv9
bg24lmdc2p node - role . kubernetes . io / ingress =" true "
node / cn - hangzhou . i - bp12h6biv9 bg24lmdc2p labeled
```

3. Run the following command to update your deployment and add the nodeSelector setting:

```
$ kubectl - n kube - system patch deployment nginx -
ingress - controller - p '{" spec ": {" template ": {" spec ": {"
nodeSelect or ": {" node - role . kubernetes . io / ingress ": "
true "}}}'
deployment . extensions / nginx - ingress - controller patched
```

Result:

Run the following command to verify that the Ingress pods are deployed on the cluster nodes that are labeled by node - role . kubernetes . io / ingress ="

```
true ":
```

```
$ kubectl - n kube - system
                                get
                                      pod – o
                                                wide
                                                       grep
nginx - ingress - controller
nginx - ingress - controller - 8648ddc696 - 2bshk
  1 / 1
              Running
                         0
                                      3h
                                              172 . 16 . 2 . 15
  cn - hangzhou . i - bp12h6biv9 bg24lmdc2p
                                              < none >
nginx - ingress - controller - 8648ddc696 - jvbs9
  1 / 1
                                      3h
                       0
                                              172 . 16 . 2 . 145
             Running
  cn - hangzhou . i - bp12h6biv9 bg24lmdc2o
                                              < none >
```

3 Storage

3.1 Use a static cloud disk when creating a stateful service

This topic describes typical scenarios in which a static cloud disk is needed for creating a stateful service, and the procedure for how to use one.

Scenarios and method

Scenarios for using cloud disks:

- You want to create applications that demand high disk I/O performance and do not require shared data. For example, MySQL, Redis, and other data storage services.
- You want logs to be written at high speed.
- You want your stored data to exist persistently. That is, the data still exist when the life cycle of the pod ends.

Scenario for using static cloud disks:

You have purchased a cloud disk.

Method of using static cloud disks:

Manually create a Persistent Volume (PV) and a Persistent Volume Claim (PVC).

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have created a cloud disk. For more information, see *Create a cloud disk*.
- You have connected to the Kubernetes cluster by using kubectl, see *Connect to a Kubernetes cluster by using kubectl*.

Limits

- Cloud disks are the non-shared storage devices provided by the Alibaba Cloud Storage Team. Each cloud disk can be mounted to only one pod.
- In a Kubernetes cluster, a cloud disk can be mounted only to a node that resides in the same zone as the cloud disk.
Create a PV

```
1. Create a pv - static . yaml file.
```

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : < your - disk - id >
  labels :
    alicloud - pvname : < your - disk - id >
    failure - domain . beta . kubernetes . io / zone : < your -
zone >
    failure - domain . beta . kubernetes . io / region : < your -
region >
spec :
  capacity :
    storage : 20Gi
  accessMode s :
    ReadWrite0 nce
  flexVolume :
    driver : " alicloud / disk "
    fsType : " ext4 "
    options :
      volumeId : "< your - disk - id >"
```

Note:

- alicloud pvname : < your disk id >: indicates the PV name. This parameter must be set to the same value as that of the volumeID parameter, namely, the cloud disk ID.
- failure domain . beta . kubernetes . io / zone : < your zone
 >: indicates the zone in which the cloud disk resides. For example, cn hangzhou b .
- failure domain . beta . kubernetes . io / region : < your -
 region >: indicates the region in which the cloud disk resides. For example,
 cn hangzhou .

If you use a Kubernetes cluster that has multiple zones, you must set the failure - domain . beta . kubernetes . io / zone parameter and the failure - domain . beta . kubernetes . io / region parameter so that you can guarantee that your pod can be scheduled to the zone in which the cloud disk resides.

2. Run the following command to create a PV:

```
$ kubectl create - f pv - static . yaml
```

```
Result
```

In the left-side navigation pane under Kubernetes, choose Clusters > Volumes, and select the target cluster to see the created PV.

Container Service - Kubernetes 👻	Volumes and Volumes Claim									
Overview	Volumes Volumes Claim									
▼ Clusters	Clusters k8s-test	•							Refresh	Create
Clusters	Name	Capacity	Access Mode	Reclaim Policy	Status	Storage Class Name	Binding Volume Claim	Time Created		Action
Nodes	1. Construction of the	20Gi	ReadWriteOnce	Retain	Bound		Namespace: default	12/18/2018,14:44:35	Edit Labels Yi	AML Delete
Volumes							Name:pvc-disk			
Namespace										

Create a PVC

Create a PVC for the cloud disk. Specifically, you need to set the selector field to filter for the created PV so that you can associate the PVC with the correct PV.

1. Create a pvc - static . yaml file.

```
kind : Persistent VolumeClai m
apiVersion : v1
metadata :
    name : pvc - disk
spec :
    accessMode s :
        - ReadWriteO nce
    resources :
        requests :
        storage : 20Gi
selector :
        matchLabel s :
        alicloud - pvname : < your - disk - id >
```

2. Run the following command to create a PVC:

\$ kubectl create - f pvc - static . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Volumes Claim, and select the target cluster and namespace to see the created PVC.

Container Service - Kubernetes 🔻		Volumes a	nd Volumes Cl	aim											
 Application 	•	Volumes	Volumes Clair	m											
Deployment		Clusters k	Bs-test	v	Namespac	e default	•						Re	fresh	Create
StatefulSet		Name	Capacity	Access Mo	de	Status	Storag	e Class Name	Relat	te Volume		Time Created			Action
DaemonSet	i.	pvc-disk	20Gi	ReadWrite	Once	Bound			1.0		-	12/18/2018,14:44:59		YAI	ML Delete
Job															
CronJob	-														
Pods															6
Volumes Claim	L														

Create an application

```
1. Create a static . yaml file.
    apiVersion : apps / v1
    kind : Deployment
    metadata :
      name : nginx - static
      labels :
        app :
               nginx
    spec :
      selector :
        matchLabel s :
          app : nginx
      template :
        metadata :
          labels :
            app : nginx
        spec :
          containers :
         - name : nginx
image : nginx
            volumeMoun ts :
                name : disk - pvc
                mountPath : "/ data "
          volumes :
              name : disk - pvc
              persistent VolumeClai m :
                claimName : pvc - disk
```

2. Run the following command to create a deployment:

\$ kubectl create - f static . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Deployment, and select the target cluster and namespace to see the created deployment.

Container Service - Kubernetes 👻	Deployment						Refresh Create b	y Image Create by Templa	ate
 Application 	Help: & How to us Container monitoring	e private images & Ø Blue-green rele	Create applications ase	Schedule a po	d to the specified node	𝔗 Create a Layer-4 Ingress	𝔗 Create a Layer-7 Ingress	${\mathscr S}$ Configure pod auto scaling	8
Deployment	Clusters k8s-test	,	Namespace de	fault 🔻					
	Name	Tag	PodsQuantity	Image	Time Created			A	Action
DaemonSet	nginx-static	app:nginx	1/1	nginx	12/31/2018,16:29:3	8	Details	Edit Scale Monitor Mo	lore 🕶

Persistent data storage on the static cloud disk

- 1. Run the following command to view the pod in which the created deployment resides:
 - \$ kubectl get pod | grep static

nginx - static - 78c7dcb9d7 - g9lll 2 / 2 Running 0 32s

2. Run the following command to check whether the new cloud disk is mounted to the / data path:

\$ kubectl exec nginx - static - 78c7dcb9d7 - g9lll df |
grep data
/ dev / vdf 20511312 45080 20449848 1 % / data

3. Run the following command to view the file in the / data path:

\$ kubectl exec nginx - static - 78c7dcb9d7 - g9lll ls / data lost + found

4. Run the following command to create a file named static in the / data path:

```
$ kubectl exec nginx - static - 78c7dcb9d7 - g9lll touch /
data / static
```

5. Run the following command to view the files in the / data path:

```
$ kubectl exec nginx - static - 78c7dcb9d7 - g9lll ls /
data
static
lost + found
```

6. Run the following command to remove the pod named nginx - static -

78c7dcb9d7 - g9lll :

\$ kubectl delete pod nginx - static - 78c7dcb9d7 - g9lll pod " nginx - static - 78c7dcb9d7 - g9lll " deleted

7. Open another kubectl interface and run the following command to view the process in which the preceding pod is removed and a new pod is created by Kubernetes:

```
kubectl
                             - l
                                    app = nginx
$
             get
                   pod – w
NAME
                                    READY
                                               STATUS
                                                           RESTARTS
  AGE
nginx - static - 78c7dcb9d7 - g9lll
                                          2 / 2
                                                       Running
                                                                    0
          50s
nginx - static - 78c7dcb9d7 - g9lll
                                          2 / 2
                                                     Terminatin
                                                                  g
0
         72s
nginx - static - 78c7dcb9d7 - h6brd
                                          0 / 2
                                                     Pending
                                                                  0
   0s
nginx - static - 78c7dcb9d7 - h6brd
                                          0 / 2
                                                     Pending
                                                                  0
   0s
                                          0 / 2
nginx - static - 78c7dcb9d7 - h6brd
                                                     Init : 0 / 1
0
         0s
                                          0 / 2
                                                     Terminatin g
nginx -
        static - 78c7dcb9d7 - g9lll
0
         73s
        static - 78c7dcb9d7 - h6brd
                                          0 / 2
                                                     Init : 0 / 1
nginx -
0
         5s
```

nginx - static - 78c7dcb9d7 0 78s	′ - g9lll	0 / 2	Terminatin	g
nginx - static - 78c7dcb9d7 0 78s	′- g9lll	0 / 2	Terminatin	g
nginx - static - 78c7dcb9d7	′– h6brd	0 / 2	PodInitial	izing
nginx - static - 78c7dcb9d7 8sg 0 8s	′– h6brd	2 / 2	Running	0

8. Run the following command to view the new pod created by Kubernetes:

<pre>\$ kubectl NAME</pre>	get	pod		READY		STATUS	RESTART	S
AGE nginx - sta	itic - 14s	78c7dcb9d7	- h6br	d	2 /	2	Running	0

9. Run the following command to verify that the created file named static in the / data path has not been removed, indicating that data in the static cloud disk can

be stored persistently:

```
$ kubectl exec nginx - static - 78c7dcb9d7 - h6brd ls /
data
static
lost + found
```

3.2 Use a dynamic cloud disk when creating a stateful service

This topic describes typical scenarios in which a dynamic cloud disk is needed for creating a stateful service, and the procedure for how to use one.

Scenarios and method

Scenario for using dynamic cloud disks:

You want to configure the system to automatically purchase cloud disks when you deploy an application, rather than manually purchase cloud disks before deploying the application.

Method of using a dynamic cloud disk:

- 1. Manually create a PVC and claim a specific StorageClass in the PVC.
- 2. Use the StorageClass to enable the system to automatically create a PV when you deploy an application.

Prerequisites

• You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.

- You have connected to the Kubernetes cluster by using kubectl, see Connect to a Kubernetes cluster by using kubectl.
- You have installed the provisioner plugin in the Kubernetes cluster. The plugin automatically creates a cloud disk according to a specific StorageClass.

Provisioner plugin

When you create a cluster through Alibaba Cloud Container Service for Kubernetes, the provisioner plugin is installed in the cluster by default.

Create a StorageClass

By default, Alibaba Cloud Container Service for Kubernetes creates four StorageClasses for a cluster during the cluster initialization, and the StorageClasses use the default settings. Furthermore, the four default StorageClasses are created only for a cluster that has a single zone. For a cluster that has multiple zones, you need to manually create a StorageClass. The following are the four StorageClasses created by default:

- alicloud disk common indicates to automatically create a basic cloud disk.
- alicloud disk efficiency indicates to automatically create an Ultra cloud disk.
- alicloud disk ssd indicates to automatically create an SSD cloud disk.
- alicloud disk available indicates a systematic method of disk selection. Specifically, the system first attempts to create an Ultra cloud disk. If the Ultra cloud disks in the specified zone are sold out, the system tries to create an SSD cloud disk. If the SSD cloud disks are sold out, the system tries to create a basic cloud disk.
- 1. Create a storagecla ss . yaml file.

```
kind : StorageCla ss
apiVersion : storage . k8s . io / v1beta1
metadata :
    name : alicloud - disk - ssd - hangzhou - b
provisione r : alicloud / disk
reclaimPol icy : Retain
parameters :
    type : cloud_ssd
    regionid : cn - hangzhou
    zoneid : cn - hangzhou - b
    fstype : " ext4 "
```

```
readonly : " false "
```

Parameter setting

• provisione r : Set this parameter to alicloud/disk to specify that the StorageClass creates an Alibaba Cloud cloud disk by using the provisioner plugin.

• reclaimPol icy : Set a policy to reclaim the cloud disk. Available values of this parameter are Delete and Retain . The default setting is Delete .

Note:

If you maintain the default setting, namely, Delete, the data on the cloud disk cannot be restored after you remove the PVC because the cloud disk is also removed.

- type : Specify a cloud disk type by using one the following values: cloud , cloud_effi ciency , cloud_ssd , and available .
- regionid : (optional) Set the region in which the cloud disk is automatically created. This region must be the same as the region in which your cluster resides.
- · zoneid : (optional) Set the zone in which a cloud disk is automatically created.
 - If you set this parameter for a single-zone cluster, the value must be the same as the zone in which the cluster resides.
 - If you set this parameter for a multi-zone cluster, multiple values can be set. For example,

```
zoneid : cn - hangzhou - a , cn - hangzhou - b , cn - hangzhou - c
```

- fstype : (optional) Set the type of the file system used for automatic cloud disk creation. The default setting is ext4 .
- readonly : (optional) Set whether the automatically created cloud disk is read only. If you set this parameter to true, the cloud disk can only be read. If you set this parameter to false, the cloud disk can be read and written. The default setting is false.
- encrypted : (optional) Set whether to encrypt the automatically created cloud disk. If you set this parameter to true, the cloud disk is encrypted. If you set

this parameter to false, the cloud disk is not encrypted. The default setting is false.

2. Run the following command to create a StorageClass:

\$ kubectl create - f storagecla ss . yaml

Create a PVC

1. Create a pvc - ssd . yaml file.

```
kind : Persistent VolumeClai m
apiVersion : v1
metadata :
    name : disk - ssd
spec :
    accessMode s :
    - ReadWriteO nce
    storageCla ssName : alicloud - disk - ssd - hangzhou - b
    resources :
        requests :
        storage : 20Gi
```

- 2. Run the following command to create a PVC:
 - \$ kubectl create f pvc ssd . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Volumes Claim, and select the target cluster and namespace to see that the storage class name associated to the PVC is alicloud – disk – ssd – hangzhou – b specified in the StorageCla ss , and the PVC is associated with the volume.

Container Service - Kubernetes 🔻		Volumes a	nd Volumes Clai	m					
 Application 	^	Volumes	Volumes Claim						
Deployment		Clusters te	st-mia	v Names	oace defau	ilt 💌			Refresh Create
StatefulSet							d sub-accounts from accessing cluster resou complete the cluster resource authorization.		
DaemonSet									
Job		Name	Capacity	Access Mode	Status	Storage Class Name	Relate Volume	Time Created	Action
CronJob	U	disk-ssd	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd-hangzhou-b	10.000	12/19/2018,14:01:19	YAML Delete
Pods	h	pvc-disk	20Gi	ReadWriteOnce	Bound		10000	12/18/2018,16:08:32	YAML Delete
Volumes Claim	1	pvc-disk-02	20Gi	ReadWriteOnce	Bound	disk	contract or later	12/18/2018,17:19:15	YAML Delete
Release	l	pvc-disk-03	20Gi	ReadWriteOnce	Bound	disk	Apple Constant	12/18/2018,19:40:16	YAML Delete

Create an application

1. Create a pvc - dynamic . yaml file.

```
apiVersion : apps / v1
kind : Deployment
```

```
metadata :
  name : nginx - dynamic
  labels :
    app : nginx
spec :
  selector :
    matchLabel s :
      app : nginx
  template :
    metadata :
      labels :
        app : nginx
    spec :
     containers :
     – name : nginx
        image : nginx
        volumeMoun ts :
    - name : disk - pvc
            mountPath : "/ data "
      volumes :
          name : disk - pvc
          persistent VolumeClai m :
            claimName : disk - ssd
```

2. Run the following command to create a deployment:

\$ kubectl create - f nginx - dynamic . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Deployment, and select the target cluster and namespace to see the created deployment.

Container Service - Kubernetes -	D	eployment				R	tefresh Create by Image	Create by Template
Application	H	lelp: 🔗 How to use itainer monitoring	private images Ø Blue-green rel	Oreate application construction of the second se	ions \mathscr{S} Schedule a pod to the specified no	ode 🔗 Create a Layer-4 Ingress 🔗 Cre	eate a Layer-7 Ingress 🔗 Configure	pod auto scaling 🔗
Deployment	Clu	sters test-mia		Namespace	default 🔻			
StatefulSet					cy in the near future, prohibiting unauthorize the "Sub-account Authorization" function to			
DaemonSet								
Job	N	ame	Tag	PodsQuantity	Image	Time Created		Action
CronJob		ginx-dynamic	app:nginx	1/1	nginx	12/19/2018,14:05:23	Details Edit Sca	le Monitor More -
Pods	ng	ginx-static	app:nginx	1/1	nginx	12/18/2018,16:09:17	Details Edit Sca	ile Monitor More - 🥃
Volumes Claim	ng	ginx-static-02	app:nginx	1/1	nginx	12/18/2018,17:20:41	Details Edit Sca	ile Monitor More -

Persistent storage for a dynamic cloud disk

1. Run the following command to view the pod in which the created deployment resides:

\$ kubectl get pod | grep dynamic

nginx - dynamic - 5c74594ccb - zl9pf 2 / 2 Running 0 3m

2. Run the following command to check whether a new cloud disk is mounted to the / *data* path:

3. Run the following command to view the file in the / data path:

\$ kubectl exec nginx - dynamic - 5c74594ccb - zl9pf ls /
data
lost + found

4. Run the following command to create a file named dynamic in the / data path:

```
$ kubectl exec nginx - dynamic - 5c74594ccb - zl9pf touch /
data / dynamic
```

5. Run the following command to view the files in the / data path:

```
$ kubectl exec nginx-dynamic-5c74594ccb-zl9pf ls /
data
dynamic
lost+found
```

6. Run the following command to remove the pod named nginx - dynamic -

78c7dcb9d7 - g9lll :

\$ kubectl delete pod nginx - dynamic - 5c74594ccb - zl9pf pod " nginx - dynamic - 5c74594ccb - zl9pf " deleted

7. Open another kubectl interface and run the following command to view the process in which the preceding pod is removed and a new pod is created by Kubernetes:

```
$
  kubectl
             get
                   pod – w – l
                                    app = nginx
NAME
                                       READY
                                                  STATUS
                                                              RESTARTS
     AGE
nginx - dynamic - 5c74594ccb - zl9pf
                                              2 / 2
                                                          Running
0
              6m48s
nginx - dynamic - 5c74594ccb - zl9pf
                                           2 / 2
                                                      Terminatin
                                                                   g
  0
           7m32s
                                                      Pending
nginx - dynamic - 5c74594ccb - 45sd4
                                           0 / 2
                                                                   0
   0s
nginx - dynamic - 5c74594ccb - 45sd4
                                           0 / 2
                                                      Pending
                                                                   0
   0s
nginx - dynamic - 5c74594ccb - 45sd4
                                           0 / 2
                                                      Init : 0 / 1
0
         0s
nginx - dynamic - 5c74594ccb - zl9pf
                                           0 / 2
                                                      Terminatin
                                                                  g
  0
           7m32s
nginx - dynamic - 5c74594ccb - zl9pf
                                           0 / 2
                                                      Terminatin
                                                                  g
  0
           7m33s
```

nginx - dynamic - 5c74594ccb - zl9pf	0 / 2	Terminatin g
0 7m33s nginx – dynamic – 5c74594ccb – 45sd4	0/2	PodInitial
izing 0 5s nginx - dynamic - 5c74594ccb - 45sd4	2 / 2	Running 0
22s	- / -	

8. Run the following command to view the pod newly created by Kubernetes:

```
$ kubectl get pod
NAME READY STATUS
RESTARTS AGE
nginx - dynamic - 5c74594ccb - 45sd4 2 / 2 Running
0 2m
```

9. Run the following command to verify that the created file named dynamic in the

/ data path has not been removed, indicating that data in the dynamic cloud disk
can be stored persistently:

```
$ kubectl exec nginx - dynamic - 5c74594ccb - 45sd4 ls /
data
dynamic
lost + found
```

3.3 Use a StatefulSet service

This topic describes the typical scenarios in which a StatefulSet is needed for creating a stateful service, and the procedure for how to use one.

Background information

A StatefulSet with N replicas is typically used for applications that require one or more of the following conditions:

- A stable deployment order. Pods are deployed or expanded sequentially. That is, pods are deployed in the defined order of 0 to N-1. Before a new pod is deployed, all its predecessors must have been in Running and Ready status.
- A stable scaling order. Pods are deleted in the defined order of N-1 to 0. Before a pod is deleted, all its predecessors must be all Running and Ready.
- Stable and unique network identifiers. After a pod is rescheduled to any other node , its PodName and HostName remain unchanged.
- Stable and persistent storage implemented through a PVC. After a pod is rescheduled, it can still access the same persistent data.

Method of using a StatefulSet service

Set volumeClai mTemplates to enable the system to automatically create a PVC and a PV.

This topic describes how to:

- Deploy a StatefulSet service
- Scale a StatefulSet service
- Remove a StatefulSet service
- · Persistent storage of a StatefulSet service

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Master node of the Kubernetes cluster. For more information, see *Connect to a Kubernetes cluster by using kubectl*.

Deploy a StatefulSet service



volumeClai mTemplates : indicates a template of PVCs of the same type. If you set this field, the system creates PVCs according to the number of the replicas that are set for the StatefulSet service. That is, the number of the PVCs and that of the replicas are the same. Furthermore, these PVCs share the same settings except for names.

1. Create a statefulse t . yaml file.

Note:

You need to set the storageCla ssName parameter to alicloud - disk - ssd , indicating that an Alibaba Cloud SSD cloud disk is used.

```
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 : StatefulSe t
metadata :
 name : web
spec :
  selector :
    matchLabel s :
     app : nginx
  serviceNam e : " nginx "
  replicas : 2
  template :
    metadata :
      labels :
       app : nginx
    spec :
     containers :
       name : nginx
       image : nginx
       ports :
        containerP ort : 80
         name : web
       volumeMoun ts :
       - name : disk - ssd
         mountPath : / data
  volumeClai mTemplates :
   metadata :
      name : disk - ssd
    spec :
      accessMode s : [ " ReadWrite0 nce " ]
     storageCla ssName : " alicloud - disk - ssd "
      resources :
        requests :
         storage : 20Gi
```

2. Run the following command to deploy a StatefulSet service:

\$ kubectl create - f statefulse t . yaml

3. Open another kubectl interface and run the following command to check that the pods are deployed in order:

\$ kubectl NAME		– l app = ATUS RES	nginx TARTS	AGE
web - 0 web - 0 web - 0 0s	0 / 1 0 / 1 0 / 1 0 / 1	Pending Pending ContainerC	0 0	0s 0s 0
web - 0 web - 1 web - 1 web - 1	1 / 1 0 / 1 0 / 1 0 / 1	Running Pending Pending ContainerC	0 0 0 reating	20s 0s 0s 0
0s web – 1	1 / 1	Running	0	7s

4. Run the following command to view the deployed pod:

\$ kubectl	get	pod			
NAME	-		READY	STATUS	RESTARTS
AGE					

web - 0	1 / 1	Running	Θ	
6m web - 1 6m	1 / 1	Running	0	

5. Run the following command to view the PVCs:

\$ kubectl	get	pvc		
NAME		STATUS	VOLUME	CAPACITY
ACCESS	MODES	STORAGECL	A SS AGE	
disk – ssd	- web		d – 2zegw7et6x	
20Gi	RWO		alicloud - disk - ssd	7m -
disk – ssd	- web	- 1 Bound	d – 2zefbrqggv	kd10xb523h
20Gi	RWO		alicloud - disk - ssd	6m

Scale a StatefulSet service

Scale out a StatefulSet service

1. Run the following command to scale out the StatefulSet service to three pods:

\$ kubectl scale sts web -- replicas = 3
statefulse t . apps / web scaled

2. Run the following command to view the pods:

\$ kubectl NAME AGE	get	pod	READY	STATUS	RESTARTS
web - 0			1 / 1	Running	Θ
34m web - 1			1 / 1	Running	Θ
33m web – 2			1 / 1	Running	Θ
26m				C C	

3. Run the following command to view the PVCs:

\$ kubectl	get	pvc				
NAME		STATU	s vo	LUME		CAPACITY
ACCESS			RAGECLA	SS	AGE	
disk – ssd	- web	- 0			- 2zegw7et6x	c96nbojuoo
20Gi	RWO		al	icloud	- disk - ssd	35m
disk – ssd	- web	- 1			- 2zefbrqggv	kd10xb523h
20Gi	RWO		al	icloud	- disk - ssd	34m
disk – ssd	- web	- 2			- 2ze4jx1zym	
20Gi	RWO		al	icloud	- disk - ssd	27m

Scale in a StatefulSet service

1. Run the following command to scale in the StatefulSet service to two pods:

\$ kubectl scale sts web -- replicas = 2

statefulse t . apps / web scaled

2. Run the following command to view the pod and verify that the number of pods is

```
reduced to two:
```

<pre>\$ kubectl NAME</pre>	get	pod	READY	STATUS	RESTARTS
AGE web - 0				Running	
38m			1 / 1	0	Θ
web - 1 38m			1 / 1	Running	Θ

3. Run the following command to view the PVCs and verify that the number of PVCs and PVs remains unchanged after the number of pods is changed:

\$ kubectl get pvc VOLUME STATUS CAPACITY NAME ACCESS MODES STORAGECLA SS AGE d - 2zegw7et6x disk - ssd - web - 0 Bound c96nbojuoo RWO alicloud - disk - ssd 20Gi 39m disk - ssd - web - 1 d – 2zefbrqggv Bound kd10xb523h alicloud - disk - ssd 20Gi RWO 39m disk - ssd - web - 2 d – 2ze4jx1zym Bound n4n9j3pic2 alicloud - disk - ssd 20Gi RWO 31m

Rescale out a StatefulSet service

1. Run the following command to scale out the StatefulSet service to three pods:

\$ kubectl scale sts web -- replicas = 3
statefulse t . apps / web scaled

2. Run the following command to view the pods:

\$ kubectl NAME AGE	get	pod	READY	STATUS	RESTARTS
web - 0			1 / 1	Running	Θ
1h web – 1			1 / 1	Running	Θ
1h web - 2			1 / 1	Running	Θ
8s			_ , _		·

3. Run the following command to view the PVCs and verify that the newly created pods still use the original PVCs and PVs after the StatefulSet service is scaled out:

\$ kubectl	get	pvc			
NAME		STATUS	VOLUME		CAPACITY
ACCESS	MODES		CLA SS	AGE	
disk – ssd	– web	– 0 Bou		- 2zegw7et6x	c96nbojuoo
20Gi	RWO			d – disk – ssd	1h
disk – ssd	– web	- 1 Bou		- 2zefbrqggv	kd10xb523h
20Gi	RWO		aliclou	d – disk – ssd	1h

disk – ssd	- web - 2	Bound	d – 2ze4jx1zym	n4n9j3pic2
20Gi	RWO	alio	cloud - disk - ssd	1h

Remove a StatefulSet service

1:

1. Run the following command to view the PVC that is used by the pod named web -

\$ kubectl describe pod web - 1 | grep ClaimName ClaimName : disk - ssd - web - 1

2. Run the following command to remove the pod named web - 1 :

```
$ kubectl delete pod web - 1
pod " web - 1 " deleted
```

3. Run the following command to view the pods and verify that the recreated pod shares the same name with the removed pod:

\$ kubectl NAME	get	pod	READY	STATUS	RESTARTS
AGE web - 0 1h			1 / 1	Running	Θ
web - 1 25s			1 / 1	Running	Θ
web - 2 9m			1 / 1	Running	Θ

4. Run the following command to view the PVCs and verify that the recreated pod uses the same PVC as removed the pod:

\$ kubectl	get	pvc				
NAME	-	STAT	US VO	LUME		CAPACITY
ACCESS					AGE	
disk – ssd	- web				- 2zegw7et6x	
20Gi	RWO		al	icloud	- disk - ssd	1h
disk – ssd	- web	- 1			- 2zefbrqggv	
20Gi	RWO		al	icloud	- disk - ssd	1h
disk – ssd	- web	- 2			- 2ze4jx1zym	
20Gi	RWO		al	icloud	- disk - ssd	1h

5. Open a new kubectl interface and run the following command to view the process of pod removal and pod recreation:

\$ kubectl	get po	od - w - l	app = ng	inx	
NAME	READY	STATUS R	ESTARTS	AGE	
web – 0	1 / 1	Running	Θ	102m	
web – 1	1 / 1	Running	Θ	69s	
web - 2	1 / 1	Running	Θ	10m	
web - 1	1 / 1	Terminatin	g 0	89s	
web - 1	0 / 1	Terminatin	g O	89s	
web - 1	0 / 1	Terminatin	g 0	90s	
web - 1	0 / 1	Terminatin	g 0 0	90s	
web - 1	0/1	Pending	õ	0s	
web - 1	0 / 1	Pending	Θ	0s	
web - 1	0/1	ContainerC	reating	Θ	0s

web - 1 1 / 1 Running 0 20s

Persistent storage of a StatefulSet service

1. Run the following command to view the file in the / data path:

\$ kubectl exec web - 1 ls / data
lost + found

- 2. Run the following command to create a statefulse t file in the / data path:
 - \$ kubectl exec web 1 touch / data / statefulse t
- 3. Run the following command to view the files in the / data path:

```
$ kubectl exec web - 1 ls / data
lost + found
statefulse t
```

4. Run the following command to remove the pod named web - 1 :

```
$ kubectl delete pod web - 1
pod "web - 1 " deleted
```

5. Run the following command to view the files in the / data path and verify that the created file named statefulse t has not been removed, indicating that data in the cloud disk can be stored persistently:

```
$ kubectl exec web - 1 ls / data
lost + found
statefulse t
```

3.4 Use a NAS file system when creating a stateful service

This topic describes typical scenarios in which a NAS file system is needed for creating a stateful service, and the procedure for how to use one.

Scenarios and method

If a NAS file system is mounted to multiple pods, the pods share the data in the NAS file system. After a pod modifies the data stored in the NAS file system, the applicatio n supported by the pods is required to automatically update the modified data for the other pods.

Scenarios for using a NAS file system

- You want to create or run applications that demand high disk I/O performance.
- You need a storage service that has higher read and write performance than OSS.

• You want to share files across hosts. For example, you want to use a NAS file system as a file server.

Method of using a NAS file system:

- 1. Manually create a NAS file system and add a mount point to it.
- 2. Manually create a PV and a PVC.

This topic describes how to use Alibaba Cloud NAS services in the PV/ PVC mode by using the *flexvolume* plugin provided by Alibaba Cloud.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Kubernetes cluster by using kubectl, see Connect to a Kubernetes cluster by using kubectl.
- You have created a NAS file system in the NAS console. For more information, see *Create a file system*. You must make sure that the NAS file system and your Kubernetes cluster are in the same zone.
- You have added a mount point for your Kubernetes cluster in the created NAS file system. For information, see *Add a mount point*. You must make sure that the NAS file system and your cluster are in the same VPC.

Create a PV

1. Create a pv - nas . yaml file.

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : pv - nas
  labels :
   alicloud - pvname : pv - nas
spec :
  capacity :
   storage : 5Gi
  accessMode s :
     ReadWriteM any
  flexVolume :
   driver : " alicloud / nas "
   options :
      server : "***-**. cn - hangzhou . nas . aliyuncs . com "
 //// Replace this value
                              with your
                                            mount
                                                   point .
     path : "/ k8s1 "
```

vers : " 4 . 0 "

Parameter description

- · alicloud pvname : indicates a PV name.
- server : indicates a NAS mount point. To view your mount point, log on to the NAS console, click File System List in the left-side navigation pane, select the target file system, and click Manage in the Action column to view the Mount Address in the Mount Point area. The mount address is the mount point of your NAS file system.

<	02	101								
File System Details	Basic Infor	mation							Delete File System	^
	File System	ID: 02aa4494fd		Region: China East 1 (Hangzhou)			Zone: Chir	na East 1 Zone G		
	Storage Typ	ge Type: SSD performance-type			tocol Type: NFS (NFSv3 and NFSv4.0)			n Usage: OB		
	Created On:	Dec 20, 2018, 5:26:13 PM	1							
	Charmen De									
	Storage Pa	аскаде								^
	ID: Buy Pa	ckage	Capacity:		Started At:		Valid Until:			8
										intact
	Mount Poir	nt						How to mount	Add Mount Point	~ 5
	Mount Point Type 🕈	VPC	VSwitch 🗢	Mount Address]	Permission Group	Status 🕈			Action
	VPC 🎝	vpc-	vsw- bp149pxcw4v9tdwzclr83	02aa4494fd-ag hangzhou.nas.		VPC default permission group (Available	Modify Permi	ssion Group Ac Disable	ttivate Delete

- path : indicates the NAS mount directory. You can mount a NAS sub-directory to your cluster. If the NAS sub-directory specified by you does not exist, the system automatically creates the NAS sub-directory and mounts it to your cluster.
- vers : (optional) indicates the version number of the NFS mount protocol. NFS file system V3.0 and V4.0 are available. The default is V4.0.
- mode : (optional) indicates the access permission to the mount directory. By default, this parameter is not set.

Note:

- Access permission to the root directory of the NAS file system cannot be set.
- If you set the model parameter for a NAS file system that stores a large amount of data, the process of mounting the NAS file system to a cluster may take an excessive amount of time or even fail. We recommend that you do not set this parameter.

2. Run the following command to create a PV:

\$ kubectl create - f pv - nas . yaml

Result

In the left-side navigation pane under Kubernetes, choose Clusters > Volumes, and select the target cluster to view the created PV.

Container Service - Kubernetes 🕶	Volumes and Volumes Clai	m								
Overview	Volumes Volumes Claim									
 Clusters 	Clusters kubernetes-test	•							Refresh	Create
Clusters	Name	Capacity	Access Mode	Reclaim Policy	Status	Storage Class Name	Binding Volume Claim	Time Created		Action
Nodes	0	20Gi	ReadWriteOnce	Retain	Bound	alicloud-disk-ssd-hangzhou-g	Namespace: default Name:disk-ssd	12/19/2018,14:48:49	Edit Labels	YAML Delete
Namespace	0	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-ssd	Namespace: default Name:disk-ssd-web-1	12/20/2018,10:24:20	Edit Labels	YAML Delete
Authorization Application	0	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-ssd	Namespace: default Name:disk-ssd-web-0	12/20/2018,10:24:05	Edit Labels	YAML Delete
Deployment	pv-nas	5Gi	ReadWriteMany	Retain	Bound		Namespace: default Name:pvc-nas	12/20/2018,19:16:00	Edit Labels	YAML Delete

Create a PVC

Create a PVC for the NAS file system. Specifically, you need to set the selector field to filter for the created PV so that you can associate the PVC with the correct PV.

```
1. Create a pvc - nas . yaml file.
```

```
kind : Persistent VolumeClai m
apiVersion : v1
metadata :
    name : pvc - nas
spec :
    accessMode s :
        - ReadWriteM any
    resources :
        requests :
        storage : 5Gi
    selector :
        matchLabel s :
        alicloud - pvname : pv - nas
```

2. Run the following command to create a PVC:

\$ kubectl create - f pvc - nas . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Volumes Claim, and select the target cluster and namespace to view the created PVC.

Container Service - Kubernetes -	Volumes and V	olumes Claim						
Application	Volumes V	olumes Claim						
Deployment	Clusters kubern	etes-test	 Namespace 	default	Ŧ			Refresh Create
StatefulSet						p-accounts from accessing cluster resound the cluster resource authorization.	rces.	
DaemonSet	Please C	ontact the main	account in time to use tr	le Sub-acco	unt Authorization Tunction to comp	siete the cluster resource authorization.		
dot	Name	Capacity	Access Mode	Status	Storage Class Name	Relate Volume	Time Created	Action
CronJob	disk-ssd	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd-hangzhou-g	(supply being an	12/19/2018,14:48:40	YAML Delete
Pods	disk-ssd-web-0	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd	10.000	12/20/2018,10:23:57	YAML Delete
Volumes Claim	disk-ssd-web-1	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd	and the first section.	12/20/2018,10:24:13	YAML Delete
Release	pvc-nas	5Gi	ReadWriteMany	Bound		pv-nas	12/20/2018,19:23:02	YAML Delete

Create an application

1. Create a nas . yaml .

```
apiVersion : apps / v1
kind : Deployment
metadata :
  name : nas - static
  labels :
    app : nginx
spec :
  replicas : 2
  selector :
    matchLabel s :
     app : nginx
  template :
    metadata :
      labels :
       app : nginx
    spec :
     containers :
       name : nginx
       image : nginx
       ports :
         containerP ort : 80
       volumeMoun ts :
           name : pvc - nas
           mountPath : "/ data "
      volumes :
         name : pvc - nas
         persistent VolumeClai m :
           claimName : pvc - nas
```

2. Run the following command to create a deployment:

\$ kubectl create - f nas . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Deployment, and select the target cluster and namespace to view the created deployment.

Container Service - Kubernetes -		Deployment					Refresh	Create by	Image	Crea	ate by Tem	plate
 Application 	^	Help: & How to use Container monitoring			ns \mathscr{S} Schedule a pod to the specified node	🖉 Create a Layer-4 Ingress	🖉 Create a Laye	er-7 Ingress	8 Config	ure pod a	auto scaling	9
Deployment		Clusters kubernetes	-test 🔻	Namespace	default 🔻							
StatefulSet					r in the near future, prohibiting unauthorized s ne "Sub-account Authorization" function to com							
DaemonSet												
Job		Name	Tag	PodsQuantity	Image	Time Create	d					Action
CronJob	œ	nas-static	app:nginx	2/2	nginx	12/20/2018	19:31:40	Details	Edit	Scale	Monitor	More 🗸
Pods		new-nginx	run:new-nginx	1/1	registry.cn-hangzhou.aliyuncs.com/xianlu/n	ew-nginx 12/29/2018	10:28:03	Details	Edit	Scale	Monitor	More - C

Verify that the NAS file system is shared by pods

1. Run the following command to view the pods in which the created deployment resides:

```
$ kubectl
                   pod
             get
NAME
                                     READY
                                               STATUS
                                                            RESTARTS
  AGE
nas - static - f96b6b5d7 - rcb2f
                                                                    0
                                           1 / 1
                                                        Running
           9m
nas - static - f96b6b5d7 - wthmb
                                           1 / 1
                                                        Running
                                                                    0
           9m
```

2. Run the following commands to view the files in the / data path of each pod:

\$ kubectl exec nas - static - f96b6b5d7 - rcb2f ls / data
\$ kubectl exec nas - static - f96b6b5d7 - wthmb ls / data

Note:

The two / data paths are empty.

- 3. Run the following command to create file *nas* in the / *data* path of one pod:
 - \$ kubectl exec nas static f96b6b5d7 rcb2f touch / data
 / nas
- 4. Run the following commands to view the files in the / data path of each pod:

```
$ kubectl exec nas - static - f96b6b5d7 - rcb2f ls / data
nas
$ kubectl exec nas - static - f96b6b5d7 - wthmb ls / data
```

Note:

nas

After you create the file in the / data path of one pod, the file then exists in both the / data paths of the two pods. This means that the two pods share the NSA file system.

Verify that data on the NAS file system are stored persistently

1. Run the following command to remove all the pods of the created application:

\$ kubectl delete pod nas - static - f96b6b5d7 - rcb2f nas static - f96b6b5d7 - wthmb pod " nas - static - f96b6b5d7 - rcb2f " deleted pod " nas - static - f96b6b5d7 - wthmb " deleted

2. Open another kubectl interface and run the following command to view the process in which the original pods are removed and new pods are created by Kubernetes:

\$ kubectl ge [.] NAME AGE	t pod - w - l		TATUS R	ESTARTS
	f96b6b5d7 - rcb2f	1 / 1	Runni	ng O
	f96b6b5d7 - wthmb	1 / 1	Runni	ng O
nas - static - 28m	f96b6b5d7 - rcb2f	1 / 1	Terminatin	g 0
nas - static - 0s	f96b6b5d7 – wnqdj	0 / 1	Pending	0
nas - static - 0s	f96b6b5d7 – wnqdj	0 / 1	Pending	0
nas - static - 0 0s	f96b6b5d7 – wnqdj	0 / 1	ContainerC	reating
	f96b6b5d7 - wthmb	1 / 1	Terminatin	g 0
	f96b6b5d7 – nwkds	0 / 1	Pending	0
nas - static - 0s	f96b6b5d7 – nwkds	0 / 1	Pending	0
	f96b6b5d7 – nwkds	0 / 1	ContainerC	reating
nas – static – 28m	f96b6b5d7 - rcb2f	0 / 1	Terminatin	g 0
	f96b6b5d7 - wthmb	0 / 1	Terminatin	g 0
nas – static – 28m	f96b6b5d7 - rcb2f	0 / 1	Terminatin	g 0
	f96b6b5d7 - rcb2f	0 / 1	Terminatin	g 0
	f96b6b5d7 - wnqdj	1 / 1	Running	0
	f96b6b5d7 - wthmb	0 / 1	Terminatin	g O
	f96b6b5d7 - wthmb	0 / 1	Terminatin	g O
	f96b6b5d7 - nwkds	1 / 1	Running	Θ

3. Run the following command to view the new pods created by Kubernetes:

\$ kubectl NAME	get	pod	READY	STATUS	RESTART	S
	c - f9 1s	6b6b5d7 - nwkds	1 /	1	Running	0

```
nas - static - f96b6b5d7 - wnqdj 1 / 1 Running 0
21s
```

4. Run the following commands to view the files in the / data path of each pod:

```
$ kubectl exec nas - static - f96b6b5d7 - nwkds ls / data
$ kubectl exec nas - static - f96b6b5d7 - wnqdj ls / data
nas
```

Note:

The created file, namely, file nas has not been removed. This means that data in the NAS file system can be stored persistently.

3.5 Use an OSS bucket when creating a stateful service

This topic describes typical scenarios in which an Object Storage Service (OSS) bucket is needed for creating a stateful service, and the procedure for how to use the bucket.

Scenarios and method

Alibaba Cloud OSS provides massive, secure, low-cost, and highly reliable cloud storage services. An OSS bucket can be mounted to multiple pods.

Scenarios:

- Disk I/O performance requirements are low.
- · Shared services such as files, figures, and short videos are to be configured.

Method:

- 1. Manually create a bucket.
- 2. Obtain the AccessKey ID and AccessKey Secret pair.
- 3. Manually create a Persistent Volume (PV) and a Persistent Volume Claim (PVC).

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Kubernetes cluster by using kubectl, see *Connect to a Kubernetes cluster by using kubectl*.
- You have created a bucket in the OSS console, see Create a bucket.

Precautions

- Upgrading a Kubernetes cluster of Alibaba Cloud Container Service causes kubelet and the OSSFS driver to restart. As a result, the OSS directory becomes unavailabl e. In this case, the pods that use the OSS bucket must be recreated. We recommend that you add health check settings in the YAML file of your application. If you add health check settings for your application, the pods will be automatically restarted to remount the OSS bucket when the OSS directory within your container becomes unavailable.
- If you use a Kubernetes cluster of the latest version, the preceding issue does not affect you.

Create a PV

1. Create a pv - oss . yaml file.

```
apiVersion : v1
kind : Persistent Volume
metadata :
  name : pv - oss
  labels :
    alicloud - pvname : pv - oss
spec :
  capacity :
    storage : 5Gi
  accessMode s :
     ReadWriteM any
  storageCla ssName : oss
  flexVolume :
    driver : " alicloud / oss "
    options :
      bucket : " docker "
                                                    //// Replace
              with your
this
       value
                              bucket
                                        name .
      url : " oss - cn - hangzhou . aliyuncs . com "
ce this value with your URL .
                                                             ////
                                your
Replace
      akId : "***"
                                                  //// Replace
                                                                  this
                  your
                                      ID .
  value with
                         AccessKey
      akSecret : "***"
                                                  //// Replace
                                                                  this
  value with your AccessKey
                                      Secret .
```

```
otherOpts : "- o max_stat_c ache_size = 0 - o
allow_othe r " //// Replace this value with your
specified otherOpts value.
```

Parameter description

- alicloud pvname : indicates a PV name. This parameter value must be used in the selector field of the PVC associated with the PV.
- bucket : indicates a bucket name. Only buckets can be mounted to a Kubernetes cluster. The sub-directories or files in a bucket cannot be mounted to any Kubernetes cluster.
- url : indicates a domain name used to access the OSS bucket, namely, an endpoint. For more information, see *Regions and endpoints*. You can also view the endpoint of the created OSS bucket in the OSS console. That is, log on to the OSS console, select the target bucket, and view Endpoint in the Domain Names area.
- akId : indicates your AccessKey ID. In the Container Service console, click



in the upper-right corner. For a primary account, select accesskeys.

For a RAM user, select AccessKey. Then, you can create your AccessKey ID and AccessKey Secret.

- akSecret : indicates your AccessKey Secret. Use the same method to obtain this parameter value as that to obtain the value of the akId parameter.
- otherOpts : indicates custom parameters for mounting the OSS bucket. Set this parameter in the format of o *** o ***. For more information, see FAQ.
- 2. Run the following command to create a PV:

\$ kubectl create - f pv - oss . yaml

Result

In the left-side navigation pane under Kubernetes, choose Clusters > Volumes, and select the target cluster to view the created PV.

Container Service - Kubernetes 🕶	Volumes and Volumes Clair	n								
Overview	Volumes Volumes Claim									
 Clusters 	Clusters kubernetes-test	*							Refresh	reate
Clusters	Name	Capacity	Access Mode	Reclaim Policy	Status	Storage Class Name	Binding Volume Claim	Time Created		Action
Nodes	0	20Gi	ReadWriteOnce	Retain	Bound	alicloud-disk-ssd-hangzhou-g	Namespace: default Name:disk-ssd	12/19/2018,14:48:49	Edit Labels YAML	Delete
Namespace	0	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-ssd	Namespace: default Name:disk-ssd-web-1	12/20/2018,10:24:20	Edit Labels YAML	Delete
Authorization Application	0	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-ssd	Namespace: default Name:disk-ssd-web-0	12/20/2018,10:24:05	Edit Labels YAML	Delete
Deployment	pv-nas	5Gi	ReadWriteMany	Retain	Bound		Namespace: default Name:pvc-nas	12/20/2018,19:16:00	Edit Labels YAML	Delete
StatefulSet	pv-oss	5Gi	ReadWriteMany	Retain	Bound	oss	Namespace: default Name:pvc-oss	12/21/2018,16:01:55	Edit Labels YAML	Delete

Create a PVC

Create a PVC for the OSS bucket. Specifically, you need to set the selector field to filter for the created PV so that you can associate the PVC with the correct PV. Set the storageCla ssName parameter to associate the PVC with only the PV of the OSS type.

```
1. Create a pvc - oss . yaml file.
```

```
kind : Persistent VolumeClai
                               m
apiVersion : v1
metadata :
 name : pvc - oss
spec :
 accessMode s :

    ReadWriteM any

  storageCla ssName : oss
  resources :
   requests :
     storage : 5Gi
  selector :
   matchLabel s :
     alicloud - pvname :
                          pv - oss
```

2. Run the following command to create a PVC:

\$ kubectl create - f pvc - oss . yaml

Result

In the left-side navigation pane under Kubernetes, choose Application > Volumes Claim, and select the target cluster and namespace to view the created PVC.

Container Service - Kubernetes 🔻		Volumes and Volu	umes Claim							
 Application 	•	Volumes Volu	mes Claim							
Deployment		Clusters kubernete	is-test	 Namespace 	default	Ŧ			Refresh	Create
StatefulSet DaemonSet						future, prohibiting unauthorized sub-acc unt Authorization" function to complete		s.		
Job	l	Name	Capacity	Access Mode	Status	Storage Class Name	Relate Volume	Time Created		Action
CronJob	U	disk-ssd	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd-hangzhou-g	1.0.04070.000000	12/19/2018,14:48:40	Y	AML Delete
Pods	h	disk-ssd-web-0	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd	and the second second	12/20/2018,10:23:57	Y	AML Delete
Volumes Claim	1	disk-ssd-web-1	20Gi	ReadWriteOnce	Bound	alicloud-disk-ssd	No. Roll International Con-	12/20/2018,10:24:13	Y	AML Delete
Release	I	pvc-nas	5Gi	ReadWriteMany	Bound		pv-nas	12/20/2018,19:23:02	Y	AML Delete
 Discovery and Load B 		pvc-oss	5Gi	ReadWriteMany	Bound	055	pv-oss	12/21/2018,16:02:06	Y	AML Delete

Create an application

```
1. Create an oss - static . yaml file.
```

```
apiVersion : apps / v1
kind : Deployment
metadata :
  name : oss - static
  labels :
    app : nginx
spec :
  replicas : 1
  selector :
    matchLabel s :
     app : nginx
  template :
    metadata :
      labels :
        app : nginx
    spec :
     containers :
     - name : nginx
image : nginx
        ports :
       - containerP ort : 80
        volumeMoun ts :
    - name : pvc - oss
         mountPath : "/ data "
- name : pvc - oss
            mountPath : "/ data1 "
        livenessPr obe :
          exec :
            command :
           - sh
           - - c
           - cd / data
          initialDel aySeconds : 30
          periodSeco nds : 30
      volumes :
          name : pvc - oss
          persistent VolumeClai m :
            claimName : pvc - oss
```

For more information about livenessPr obe , see Use Alibaba Cloud OSS volumes.

2. Run the following command to create a deployment:

\$ kubectl create - f oss - static . yaml d

Result

In the left-side navigation pane under Kubernetes, choose Application > Deployment, and select the target cluster and namespace to view the created deployment.

Container Service - Kubernetes 🗸	Deploy	ment							Refresh	Create by	/ Image	Cre	ate by Ter	nplate
Application	Help: Ø Container	How to use privat Monitoring	te images 🛛 🔗 🕻 Ilue-green releas	Create application	s 🔗 Schedul	e a pod to the specified node	🔗 Create a Laye	er-4 Ingress 🛛 🖉	Create a Laye	r-7 Ingress	8 Config	ure pod	auto scalin	g <i>Ø</i>
Deployment	Clusters	kubernetes-test	٣	Namespace	default	Ŧ								
StatefulSet						re, prohibiting unauthorized su Authorization" function to com								
DaemonSet														
Job	Name	Та	ag	PodsQuantity	Image			Time Created						Action
CronJob	nas-stati	ic ap	p:nginx	2/2	nginx			12/20/2018,19:3	1:40	Details	Edit	Scale	Monitor	More 🕶
Pods	new-ngir	nx rur	n:new-nginx	1/1	registry.cn-ha	ngzhou.aliyuncs.com/xianlu/ne	w-nginx	12/29/2018,10:2	8:03	Details	Edit	Scale	Monitor	More - C
Volumes Claim	nginx-dy	mamic ap	p:nginx	1/1	nginx			12/19/2018,17:4	0:53	Details	Edit	Scale	Monitor	More - nlad
Release	old-ngin:	x rur	n:old-nginx	2/2	registry.cn-ha	ngzhou.aliyuncs.com/xianlu/ok	d-nginx	12/29/2018,10:2	9:27	Details	Edit	Scale	Monitor	More -
 Discovery and Load B 	oss-stati	c ap	p:nginx	1/1	nginx			12/21/2018,16:0	2:15	Details	Edit	Scale	Monitor	More 🗸

Verify that data in the OSS bucket are stored persistently

1. Run the following command to view the pod in which the created deployment resides:

```
$ kubectl get pod
NAME READY STATUS RESTARTS
AGE
oss - static - 66fbb85b67 - dqbl2 1 / 1 Running 0
1h
```

2. Run the following command to view the files in the / data path:

```
$ kubectl exec oss - static - 66fbb85b67 - dqbl2 ls / data
| grep tmpfile
```

The / data path is empty.

3. Run the following command to create a file named tmpfile in the / data path:

```
$ kubectl exec oss - static - 66fbb85b67 - dqbl2 touch /
data / tmpfile
```

4. Run the following command to view the file in the / data path:

\$ kubectl exec oss - static - 66fbb85b67 - dqbl2 ls / data
| grep tmpfile

tmpfile

5. Run the following command to remove the pod named oss - static -

```
66fbb85b67 - dqbl2 :
```

\$ kubectl delete pod oss - static - 66fbb85b67 - dqbl2 pod " oss - static - 66fbb85b67 - dqbl2 " deleted

6. Open another kubectl interface and run the following command to view the process in which the preceding pod is removed and a new pod is created by Kubernetes:

\$ kubectl NAME AGE	get	pod	– w	- l	•••		TUS RE	STARTS	;
oss - stati	c - 66 3m	fbb85b	67 -	dqbl2	1 /	1	Runnin	g	0
oss – stati 78m		fbb85b	67 -	dqbl2	1 / 1		Terminatin	g	Θ
oss - stati < invalid >	c - 66	fbb85b	67 -	zl∨mw	0 / 1		Pending	0	
oss - stati	c - 66	fbb85b	67 -	zl∨mw	0 / 1		Pending	0	
oss - stati			67 -	zl∨mw	0 / 1		ContainerC	reati	ng
oss – stati 78m			67 -	dqbl2	0 / 1		Terminatin	g	0
oss - stati 78m	c - 66	fbb85b	67 -	dqbl2	0 / 1		Terminatin	g	Θ
oss - stati 78m	c - 66	fbb85b	67 -	dqbl2	0 / 1		Terminatin	g	0
oss - stati < invalid >	c - 66	fbb85b	67 -	zl∨mw	1 / 1		Running	Θ	

7. Run the following command to view the pod created by Kubernetes:

\$ kubectl	get	pod			CTATUC	DECTAD	
NAME				READY	STATUS	RESTART	5
AGE							
oss - stati	c - 66	fbb85b67 – zl	l∨mw	1 /	1	Running	0
4	0s					U	

8. Run the following command to verify that the created file named *tmpfile* in the

/ data path has not been removed, indicating that data in the OSS bucket can be stored persistently:

```
$ kubectl exec oss - static - 66fbb85b67 - zlvmw ls / data
| grep tmpfile
tmpfile
```

4 Release

4.1 Implement Layer-4 canary release by using Alibaba Cloud Server Load Balancer in a Kubernetes cluster

In a Kubernetes cluster, Layer-7 Ingress cannot properly implement gray release for services accessed by using TCP/UDP. This document introduces how to implement Layer-4 canary release by using Server Load Balancer.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *#unique_34*.
- You have connected to the master node by using SSH. For more information, see *#unique_35*.

Step 1 Deploy the old version of the service

- 1. Log on to the Container Service console.
- 2. Click Application > Deployment in the left-side navigation pane.
- 3. Click Create by template in the upper-right corner.
- 4. 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.

In this example, an nginx orchestration that exposes the service by using SLB.

```
apiVersion : extensions / v1beta1
 kind : Deployment
metadata :
   labels :
    run : old - nginx
  name : old - nginx
 spec :
   replicas :
               1
   selector :
     matchLabel s :
       run : old - nginx
   template :
     metadata :
       labels :
         run : old - nginx
         app : nginx
     spec :
```

```
containers :
        image : registry . cn - hangzhou . aliyuncs . com /
xianlu / old - nginx
        imagePullP olicy :
                             Always
        name : old - nginx
        ports :
          containerP ort :
                             80
          protocol : TCP
       restartPol icy : Always
 apiVersion : v1
kind : Service
metadata :
   labels :
    run : nginx
  name : nginx
 spec :
  ports :
   port :
           80
    protocol : TCP
     targetPort : 80
   selector :
    app : nginx
   sessionAff inity : None
   type : LoadBalanc er ## Expose
                                      the
                                                          using
                                            service
                                                     by
         Cloud
Alibaba
                 SLB .
```

- 5. Click Application > Deployment and Application > Service in the left-side navigation pane to check the deployment and service.
- 6. Click the external endpoint at the right of the service to go to the Nginx default welcome page. In this example, old is displayed on the Nginx welcome page, which indicates that the currently accessed service corresponds to the backend old-nginx container.

To easily display the results of multiple releases , we recommend that you log on to the master node and execute the curl command to view the deployment results.

```
#
  bash
#
           in { 1 .. 10 };
                             do
                                   curl
                                          EXTERNAL - IP ; done
 for
        Х
 ## EXTERNAL - IP is
                      the
                              external
                                        endpoint
                                                of the
service .
old
old
old
old
old
old
old
old
old
```

old

Step 2 Bring new deployment version online

- 1. Log on to the Container Service console.
- 2. Click Application > Deployment in the left-side navigation pane.
- 3. Click Create by template in the upper-right corner.
- 4. 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.

In this example, create a new version of nginx deployment that contains the app : nginx label. The label is used to use the same nginx service as that of the old version of deployment to bring the corresponding traffic.

The orchestration template in this example is as follows:

```
apiVersion : extensions / v1beta1
 kind : Deployment
metadata :
   labels :
     run : new - nginx
  name : new - nginx
 spec :
   replicas : 1
   selector :
    matchLabel s :
       run : new - nginx
   template :
     metadata :
       labels :
         run : new - nginx
         app : nginx
     spec :
      containers :
        image : registry . cn - hangzhou . aliyuncs . com /
xianlu / new - nginx
         imagePullP olicy :
                              Always
         name : new - nginx
         ports :
           containerP
                      ort :
                              80
           protocol : TCP
       restartPol icy : Always
```

5. Click Deployment in the left-side navigation pane. The deployment of new-nginx is displayed on the Deployment page.

6. Log on to the master node and execute the curl command to view the service

access.

```
bash
#
        x in \{1...10\};
                             do
                                   curl EXTERNAL - IP; done
#
  for
 ## EXTERNAL - IP is
                      the
                                        endpoint of the
                              external
service .
new
new
new
old
new
old
new
new
old
old
```

You can see that the old service and new service are accessed for five times respectively. This is mainly because the service follows the Server Load Balancer policy of average traffic to process traffic requests, and the old deployment and new deployment are the same pod, which makes their traffic ratio as 1:1.

Step 3 Adjust traffic weight

You must adjust the number of pods in the backend to adjust the corresponding weight for the canary release based on Server Load Balancer. For example, to make the new service to have higher weight, you can adjust the number of new pods to four

Note:

If the old application version and new application version coexist, the results returned after executing the curl command of a sample do not conform to the configured weight strictly. In this example, to obtain the approximate effect, execute the curl command for 10 times to observe more samples.

- 1. Log on to the Container Service console.
- 2. Under Kubernetes, click Application > Deployment in the left-side navigation pane.
- 3. Select the cluster and namespace from the Clusters and Namespace drop-down lists. Click Update at the right of the deployment.

4. In the displayed dialog box, set the number of pods to four.

Note:

The default update method of Kubernetes deployment resources is rollingUpdate. Therefore, during the update process, the minimum number of containers that provide the service is guaranteed and this number can be adjusted in the template.

5. After the deployment, log on to the master node and execute the curl command to view the effect.

You can see the new service is requested for eight times and the old service is requested twice among the 10 requests.

You can dynamically adjust the number of pods to adjust the weights of the new service and old service and implement the canary release.

4.2 Implement a gray release and a blue/green deployment through Ingress in a Kubernetes cluster

4.2.1 Gray releases and blue/green deployment

This topic describes how to implement a gray release and a blue/green deployment by using the Ingress function provided by Alibaba Cloud Container Service for Kubernetes.

Background information

With a gray release or a blue/green deployment, you can create two identical production environments for the latest version of the target software and an earlier

version. Then you can apply specific rules to reroute traffic from the earlier version to the latest version without affecting the software of the earlier version. After the software of the latest version has run without exceptions for a specified period, you can reroute all traffic from the earlier version to the latest version.

A/B testing is a type of comparative and incremental gray release. Specifically, with A/B testing, you can keep some users using the service of an earlier version, and reroute traffic of other users to the service of the latest version. If the service of the latest version runs without exceptions for the specified period of time, then you can gradually reroute all user traffic to the service of the latest version.

Scenarios

Scenario 1

For example, assume that Service A already runs online to provide an externally accessible Layer-7 service, and a new version of this service with new features, namely, Service A', is developed. You want to release Service A', but you do not want it to directly replace Service A at once. Additionally, you want the client requests of which the request headers contain foo = bar or the cookies contain foo = bar to be forwarded to Service A'. Then, after Service A' has run without exceptions for a specified period, you want to reroute all traffic from Service A to Service A', and then smoothly bring Service A offline.


Scenario 2

For example, assume that an earlier version of a service, named Service B, is running online to provide an externally accessible Layer-7 service. However, it has known problems. A new version, namely Service B' is developed with the problems fixed and you want to release this latest version. However, you initially want to reroute only 20% of all client traffic to Service B' . Then, after Service B' has run without exceptions for a period, you want to reroute all traffic from Service B to Service B' , and then smoothly bring Service B offline.



To meet the preceding application release requirements, Alibaba Cloud Container Service for Kubernetes uses the Ingress function to provide the following four methods of traffic distribution:

In A/B testing

- Distribute traffic according to the request header
- · Distribute traffic according to the cookie
- $\cdot\;$ Distribute traffic according to the Query Param

In a blue/green deployment

· Distribute traffic according to the service weight

4.2.2 Gray release limits

This topic describes the limits for a gray release that is implemented by the Ingress function provided by Alibaba Cloud Container Service for Kubernetes.

The Ingress controller of Alibaba Cloud Container Service for Kubernetes must be V 0.12.0-5 or later.

To view the version number of the Ingress controller, run either of the following commands as required.

• For a cluster in which applications are deployed by using the Deployment method,

run:

```
kubectl - n kube - system get deploy nginx - ingress -
controller - o yaml | grep - v ' apiVersion ' | grep '
aliyun - ingress - controller '
```

 \cdot For a cluster in which applications are deployed by using the DaemonSet method,

run:

```
kubectl - n kube - system get ds nginx - ingress -
controller - o yaml | grep - v ' apiVersion ' | grep
aliyun - ingress - controller '
```

If your Ingress Controller is earlier than $0 \cdot 12 \cdot 0 - 5$, you can upgrade it by running either of the following commands as required.

• For a cluster in which applications are deployed by using the Deployment method,

run:

```
kubectl - n kube - system set image deploy / nginx -
ingress - controller nginx - ingress - controller = registry . cn
- hangzhou . aliyuncs . com / acs / aliyun - ingress - controller :
0 . 12 . 0 - 5
```

• For a cluster in which applications are deployed by using the DaemonSet method,

run:

```
kubectl - n kube - system set image ds / nginx - ingress
- controller nginx - ingress - controller = registry . cn -
hangzhou . aliyuncs . com / acs / aliyun - ingress - controller : 0
. 12 . 0 - 5
```

4.2.3 Annotation

This topic describes the annotation used when you implement a gray release by using the Ingress function provided by Alibaba Cloud Container Service for Kubernetes.

To support a gray release, the Ingress function of Alibaba Cloud Container Service for Kubernetes provides the following annotation: routing rules set by using nginx.

```
ingress . kubernetes . io / service - match and service weight set by using
nginx . ingress . kubernetes . io / service - weight .
```



If you set routing rules by using nginx . ingress . kubernetes . io / service - match and service weight by using nginx . ingress . kubernetes . io / service - weight , the system first determines whether the routing rules set by
using nginx . ingress . kubernetes . io / service - match are matched
when receiving a request:

- If no routing rules are matched, the system forwards the request to the application of the earlier version.
- If the routing rules are matched, the system forwards the request according to the service weight that you set by using nginx . ingress . kubernetes . io / service weight .

Routing rules set by using nginx . ingress . kubernetes . io / service - match

This annotation is used to set the routing rules for the service of the latest version. The annotation format is as follows:

Parameter description

service - name : service name. The requests that meet the requirements of the route matching rules are routed to this service.

```
match - rule : matching rules of routes.
```

- Matching types:
 - header : based on the request header. This matching type supports regular expression matches and full expression matches.
 - cookie : based on the cookie. This matching type supports regular expression matches and full matches.
 - query : based on the queried parameter. This matching type supports regular expression matches and full matches.
- · Matching methods:
 - The format of a regular expression match is /{ Regular Expression }/.
 - The format of a full match is "{ exact expression }"

Configuration examples

```
# If the request header of a request meets the
requiremen ts of the regular expression of foo and
^ bar $, the request is forwarded to the new - nginx
service .
new - nginx : header (" foo ", /^ bar $/)
```

request , if fully # In the request header of a foo request will be matches bar, the forwarded to the new - nginx service . new - nginx : header (" foo ", " bar ") In the cookie of a request , if regular expression ^ sticky -.+\$, the foo matches the regular expression ^ sticky -.+\$, the forwarded to the new - nginx service request be will new - nginx : cookie (" foo ", /^ sticky -.+\$/) the query param of a request, if foo es bar, the request will be forwarded to fullv # Tn matches bar, the the new - nginx service . new - nginx : query (" foo ", " bar ")

Service weight set by using nginx . ingress . kubernetes . io / service - weight

This annotation is used to set the traffic weights for the service of the latest version and the service of the earlier version. The annotation format is as follows:

Parameter description

new - svc - name : name of the service of the latest version. new - svc - weight : weight of the service of the latest version. old - svc - name : name of the service of the earlier version. old - svc - weight : weight of the service of the earlier version.

Configuration examples

```
nginx . ingress . kubernetes . io / service - weight : |
new - nginx : 20 , old - nginx : 60
```

Note:

- Service weights are calculated by using relative values. In the preceding example, the service of the latest version is set to 20 weight and the service of the earlier version is set to 60 weight. Therefore, the weight percentage of the latest version service is 25% and the weight percentage of the earlier version service is 75%.
- In a service group that is composed of services that have the same host and path in an Ingress YAML, the default service weight is 100.

4.2.4 Step 1: Deploy a service

This topic describes how to deploy a service.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Kubernetes cluster by using kubectl. For more information, see *Connect to a Kubernetes cluster by using kubectl*.

Procedure

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Kubernetes, choose Application > Deployment.
- 3. In the upper-right corner, click Create by Template.

Container Service - Kubernetes 🕶	Deployment				Refresh		/ Template
Overview	How to use p	rivate images 🔗 Create	applications 🔗 Schedule a pod to	the specified node 🔗 Create a Layer-4 Ingress	𝔅 Create a Layer-7 Ingress 𝔅 Con	figure pod auto scaling	2
▼ Clusters	S Container mo	nitoring 🔗 Blue-green r	release				
Clusters	Clusters k8s-te	st v Na	mespaces default 🔻			Search By Name	Q
Nodes	Name	Tag	PodsQuantity	Image	Time Created		Action
Volumes				Could not find any record that met th	e condition.		
Namespaces	-						
Authorization							Conta
Applications Deployments							act Us

4. Select the target cluster and namespace, select a sample template or customize a template, and then click DEPLOY.

Clusters	test-mia	v
Namespace	default	•
Resource Type	Custom	٣
Template	<pre>1 apiVerSion: extensions/vlbeta1 2 kind: Deployment 3 metadata: name: old-nginx 5 spec: 6 replicas: 2 7 selector: 8 matchlabels: 9 run: old-nginx 10 template: 11 metadata: 12 labels: 13 run: old-nginx 14 spec: 15 containers: 16 - image: registry.cn-hangzhou.aliyuncs.com/xianlu/old-nginx 17 imagePullPolicy: Always 18 name: old-nginx 19 ports: 20 - containerPort: 80 21 protocol: TCP 22 restartPolicy: Always 23 24 apiVerSion: v1 25 kind: Service 26 metadata: 27 name: old-nginx 28 spec: 29 ports: 30 - port: 80 31 protocol: TCP 32 targetPort: 80</pre>	
	Save Template	DEPLOY

In this example, a template is orchestrated to deploy an Nginx application that contains the required deployment, the target service, and an Ingress. The deployment exposes its port through NodePort. The Ingress provides externally accessible services. The orchestration template is as follows:

```
apiVersion : extensions / v1beta1
kind : Deployment
metadata :
 name : old - nginx
spec :
  replicas :
             2
  selector :
   matchLabel s :
     run : old - nginx
  template :
   metadata :
     labels :
        run : old - nginx
    spec :
     containers :
     - image : registry . cn - hangzhou . aliyuncs . com / xianlu
/ old - nginx
```

```
imagePullP olicy : Always
       name : old - nginx
       ports :
        containerP ort :
                            80
         protocol : TCP
      restartPol icy : Always
apiVersion : v1
kind : Service
metadata :
 name : old - nginx
spec :
 ports :
   port : 80
 -
   protocol : TCP
   targetPort : 80
 selector :
   run : old - nginx
  sessionAff ity : None
 type : NodePort
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
   name : gray - release
spec :
  rules :
   host : www . example . com
   http :
     paths :
    # earlier
                 version
                          of
                                а
                                   service
       path : /
       backend :
         serviceNam e :
                          old - nginx
         servicePor
                     t :
                          80
```

5. In the left-side navigation pane, choose Application > Ingress.

You can see that the virtual host name points to old-nginx.

Container Service - Kubernetes -	Ingress		Ingress Overview Refresh Create
Overview	${\mathscr S}$ Ingress log analysis and monitoring ${\mathscr S}$ Blue-green release		
Clusters	Clusters k8s-test v Namespaces default v		Search By Name Q
 Applications 	Name Endpoint Rule	Time Created	Action
 Service Mesh 	gray-release www.example.com/ -> old-nginx	05/30/2019,11:01:47	Details Update View YAML Delete
Virtual Services			
Discovery and Load B			_
Services			a a a a a a a a a a a a a a a a a a a
Ingresses			actus
 Configuration 			

6. Log on to the Master node and run the curl command to view the Ingress.



You can obtain the value of <EXTERNAL_IP> by using either of the following two methods:

• Run the following command:

kubectl get ingress

• Under the Kubernetes menu, choose Application > Ingress, and view the endpoint information of the target Ingress.

Container Service - Kubernetes -	Ingress			Ingress Overview Refresh Create
Overview	S Ingress log analysis and monitoring	g 🔗 Blue-green release		
Clusters	Clusters k8s-test •	Namespaces default •		Search By Name Q
Applications	Name Endpoint	Rule	Time Created	Act
 Service Mesh 	gray-release	www.example.com/ -> old-nginx	05/30/2019,11:01:47	Details Update View YAML Dele
Virtual Services				
Discovery and Load B				
Services				
Ingresses				
Configuration				



4.2.5 Step 2: Release the latest version of a service

This topic describes how to release the latest version of a service by using the Ingress function provided by Alibaba Cloud Container Service for Kubernetes.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have connected to the Kubernetes cluster by using kubectl. For more information, see *Connect to a Kubernetes cluster by using kubectl*.

Procedure

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Kubernetes, choose Application > Deployment.

3. In the upper-right corner, click Create by Template.

Container Service - Kubernetes 🔻	Deployment				Refresh Create b	y Image Create by Template
Overview	Help: & How to use private Container monitoring & Blu	images 🔗 Create applications ie-green release	Schedule a pod to the sp	ecified node 🛛 🔗 Create a Layer-4 Ingres	s 🔗 Create a Layer-7 Ingress	${\mathscr O}$ Configure pod auto scaling ${\mathscr O}$
 Clusters 	Clusters test-mia	v Namespace d	lefault 🔻			
Clusters	Name Tag	PodsQuantity	Image Time (reated		Action
Nodes	busybox run:busyb	ox 1/1	busybox 12/26	2018,17:48:46	Details	Edit Scale Monitor More -
Volumes						
Namespace						
Authorization						
 Application 						
Deployment						

4. Select the target cluster and namespace, select a sample template or customize a template, and then click DEPLOY.



Deploy an Nginx application of the latest version that contains the required deployment, the target service, and an Ingress. The orchestration template that contains the deployment and service is as follows:

```
apiVersion : extensions / v1beta1
kind : Deployment
```

```
metadata :
  name : new - nginx
spec :
  replicas : 1
  selector :
   matchLabel s :
     run : new - nginx
  template :
   metadata :
      labels :
        run : new - nginx
    spec :
     containers :
       image : registry . cn - hangzhou . aliyuncs . com / xianlu
/ new - nginx
       imagePullP olicy :
                            Always
       name : new - nginx
       ports :
        containerP ort :
                            80
         protocol : TCP
      restartPol icy : Always
apiVersion : v1
kind : Service
metadata :
  name : new - nginx
spec :
 ports :
 - port : 80
    protocol : TCP
    targetPort : 80
  selector :
    run : new - nginx
  sessionAff inity : None
  type : NodePort
```

The following are Ingress orchestration templates of different annotation settings:

Note:

If you do not set service - match or service - weight in the annotations field of an Ingress template, the Ingress controller forwards client requests evenly to the latest and earlier services in a random manner.

• Ingress template used to specify only the client requests that meet the requirement of the regular expression foo=bar to be routed to the latest version of the service

```
apiVersion : extensions / vlbetal
kind : Ingress
metadata :
    name : gray - release
    annotation s :
        nginx . ingress . kubernetes . io / service - match : |
    # Only if the request header of a request meets
    the requiremen ts of the regular expression foo =
bar , can the request be routed to the new - nginx
    service .
```

```
new - nginx : header (" foo ", /^ bar $/)
spec :
 rules :
   host :
          www . example . com
   http :
     paths :
                 version of the
     #
       Earlier
                                      service
       path : /
       backend :
         serviceNam e :
                          old - nginx
         servicePor t:
                          80
    #
       Latest
                version
                          of
                               the service
       path : /
       backend :
                          new - nginx
         serviceNam e :
         servicePor t:
                          80
```

• Ingress template used to specify the proportion of requests that can be routed to the latest version of the service

Note:

In this example, the latest version of the service and the earlier version version of the service are weighted at 50% each.

```
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
 name : gray - release
  annotation s:
     nginx . ingress . kubernetes . io / service - weight : |
# Set
        50 % of traffic to
                                be routed
                                             to
                                                   the
                                                         new -
nginx
       service .
         new - nginx : 50 , old - nginx :
                                            50
spec :
 rules :
   host :
          www . example . com
   http :
     paths :
                 version of the
     #
       Earlier
                                     service
       path : /
       backend :
                         old - nginx
         serviceNam e :
         servicePor t:
                         80
                              the service
    #
       Latest
                version
                         of
       path : /
       backend :
         serviceNam e :
                         new - nginx
         servicePor t:
                          80
```

• Ingress template used to specify that only 50% of the client request traffic that meets the requirements of foo=bar will be routed the latest version of the service

```
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
   name : gray - release
```

annotation s : nginx . ingress . kubernetes . io / service - match : | Only if the request header of a request meets # meets the requiremen ts of the regular expression foo = be bar, can the request routed the to new – nginx service . new - nginx : header (" foo ", /^ bar \$/)
nginx . ingress . kubernetes . io / service - weight : | #
Only 50 % of the client request traffic that meet meets of the preceding matching the new-nginx service. the requiremen ts rule can be routed to new - nginx : 50, old - nginx : 50spec : rules : host : www . example . com http : paths : # Earlier version of the service path : / backend : serviceNam e :
servicePor t : old - nginx 80 the # service Latest version of path : / backend : new - nginx serviceNam e : servicePor t: 80

5. In the left-side navigation pane, choose Application > Ingress.

You can see that the virtual host name points to old-nginx.

Container Service - Kubernetes +	Ingress	Refresh Create			
Overview	Help: 🔗 Blue-greer	n release			
Clusters	Clusters test-mia				
Application	Name	Endpoint	Rule	Time Created	Action
 Discovery and Load B 	gray-release		www.example.com/ -> old-nginx	12/29/2018,10:51:33	Details Update View YAML Delete
Service	gray-release-01	10123	www.example.com-01/ -> old-nginx www.example.com-01/ -> new-nginx-01	12/29/2018,10:54:23	Details Update View YAML Delete
Configuration	gray-release-02		www.example.com-02/ -> old-nginx www.example.com-02/ -> new-nginx-02	12/29/2018,10:56:01	Details Update View YAML Delete
Config Maps	gray-release-03	*****	www.example.com-03/ -> old-nginx www.example.com-03/ -> new-nginx-03	12/29/2018,10:57:55	Details Update View YAML Delete
Secret					

- 6. Log on to the Master node and run the following curl commands to view the Ingress access of the following settings:
 - Only the client requests that meet the requirements of the regular expression foo=bar can be routed to the latest version of the service.

```
# curl - H " Host : www . example . com " - H " foo : bar "
http ://< EXTERNAL_I P >
```

```
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example1.com" -H "foo: bar" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example1.com" -H "foo: bar" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example1.com" -H "foo: bar" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example1.com" -H "foo: bar" http://
```

 Requests of a specified proportion can be routed to the latest version of the service.

```
# curl - H " Host : www . example . com " http ://<
    EXTERNAL_I P >
```

```
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example2.com" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example2.com" http://
old
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example2.com" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example2.com" http://
old
```

• Only 50% of the client request traffic that meets the requirements of the regular express foo=bar can be routed to the latest version of the service.

```
# curl - H " Host : www . example . com " - H " foo : bar "
http ://< EXTERNAL_I P >
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example3.com" -H "foo: bar" http://
old
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example3.com" -H "foo: bar" http://
new
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example3.com" -H "foo: bar" http://
old
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example3.com" -H "foo: bar" http://
```

4.2.6 Step 3: Remove the earlier version of a service

This topic describes how to remove the earlier version of a service when the latest version of the service (which has been released through a gray release) has run without exceptions for a specified period of time.

Prerequisites

• You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.

- You have connected to the Kubernetes cluster by using kubectl, see *Connect to a Kubernetes cluster by using kubectl*.
- You have deployed an earlier version of the service. For more information, see *Step 1: Deploy a service*. You have also released a later version of the service through a gray release. For more information, see *Step 2: Release the latest version of a service*.

Run a command

1. Run the following command to edit the YAML file deployed by *Step 2: Release the latest version of a service* to remove the earlier version of the service:



Use the Container Service console

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Kubernetes, choose Application > Ingress.
- 3. Select the target cluster and namespace, select the target Ingress, and click Update in the action column.

Container Service - Kubernetes +	Ingress					
Overview	Help: 🔗 Blue-green	release				
Clusters	Clusters test-mia		Namespace default v			
Application	Name	Endpoint	Rule	Time Created	Action	
 Discovery and Load B 	gray-release	10.01214	www.example.com/ -> old-nginx	12/29/2018,10:51:33	Details Update View YAML Delete	
Service	gray-release-01	100.2	www.example.com-01/ -> old-nginx www.example.com-01/ -> new-nginx-01	12/29/2018,10:54:23	Details Update View YAML Delete	
Tingress Configuration	gray-release-02	1011.4	www.example.com-02/ -> old-nginx www.example.com-02/ -> new-nginx-02	12/29/2018,10:56:01	Details Update View YAML Delete	
Config Maps	gray-release-03		www.example.com-03/ -> old-nginx www.example.com-03/ -> new-nginx-03	12/29/2018,10:57:55	Details Update View YAML Delete	
Secret						

4. In the displayed dialog box, modify the Ingress as follows:

a. In the Rule > Service area, remove the earlier version of the service rule.

Update		\times
Name: Rule:	gray-release-02	
	Domain www.example.com-02 Select *. Custom path / Service • Add	
	NamePortWeightPercent of Weightold-nginx8010050.0%Image: Constraint of the second s	
	new-nginx-02 ▼ 80 ▼ 100 50.0% ● □ EnableTLS	
Service weight:	✓ Enable	
Grayscale release:	Add After the gray rule is set, the request meeting the rule will be routed to the new service. If you set a weight other than 100, the request to satisfy the gamma rule will continue to be routed to the new and old version services according to the weights.	
annotation:	• Add rewrite annotation	
Tag:	O Add	
	Update Car	ncel

b. Click Update.

Result

1. Return to the Ingress page. Here, you can see that only one Ingress rule points to the new-nginx service.

Container Service - Kubernetes 🔻	Ingress	Ingress					
Overview	Help: 🔗 Blue-green	release					
 Clusters 	Clusters test-mia		▼ Namespace default ▼				
Application	Name	Endpoint	Rule	Time Created	Action		
 Discovery and Load B 	gray-release		www.example.com/ -> old-nginx	12/29/2018,10:51:33	Details Update View YAML Delete		
Service	gray-release-01	1000	www.example.com-01/ -> old-nginx www.example.com-01/ -> new-nginx-01	12/29/2018,10:54:23	Details Update View YAML Delete		
Configuration	gray-release-02		www.example.com-02/ -> new-nginx-02	12/29/2018,11:20:32	Details Update View YAML Delete		
Config Maps	gray-release-03	100.20	www.example.com-03/ -> old-nginx www.example.com-03/ -> new-nginx-03	12/29/2018,10:57:55	Details Update View YAML Delete		

2. Log on to the Master node and run the curl command to view the Ingress access.

```
$ curl - H " Host : www . example2 . com " http ://<
EXTERNAL_I P >
ubuntu-mia@ubuntumia-VirtualBox:~$ curl -H "Host: www.example2.com" http://
new
```

Now, all requests are routed to the latest version of the service, which means you have completed the gray release deployment cycle. You can also remove the deployment and service of the earlier version.

4.3 Application

5 Istio

5.1 Use Istio to implement intelligent routing in Kubernetes

Alibaba Cloud Container Service for Kubernetes supports one-click deployment of Istio and multiple functions expanded on Istio. This topic describes how to implement intelligent routing through Istio. For information about Istio official documents, see *Intelligent Routing*.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have deployed Istio. For more information, see *Deploy Istio*.

Note:

Istio used in this topic is V 1.0.2.

- You have a local Linux environment in which you have configured the kubectl tool and used the tool to connect to the cluster. For more information, see *Connect to a Kubernetes cluster by using kubectl*.
- You have downloaded the project code of an Istio version and run the relevant commands in the Istio file directory. See https://github.com/istio/istio/releases.

Install the Istio official sample application

Install the Istio official sample application, Bookinfo. For more information, see *https* ://istio.io/docs/guides/bookinfo.

Quickly deploy the Bookinfo sample application

1. Label the default namespace with the istic - injection = enabled tag.



Kubernetes clusters running on Alibaba Cloud Container Service support oneclick deployment of Istio and automatic sidecar injection.

\$ kubectl label namespace default istio - injection =
enabled

2. Run the following kubectl command to deploy the Bookinfo sample application:

```
$ kubectl apply - f samples / bookinfo / platform / kube /
bookinfo . yaml
```

The preceding command starts all four microservices. All three Reviews service versions (v1, v2, and v3) are also started.

3. Run the following command to vefiry that all services and pods are properly defined and started:

\$ kubectl get svc , pods

4. You need to access the application from the outside of your Kubernetes cluster, for example, a browser. You need to create an *Istio Gateway*. Define the ingress gateway for the application.

```
$ kubectl apply - f samples / bookinfo / networking / bookinfo
- gateway . yaml
```

Run the following command to verify that the gateway has been created:

\$ kubectl get gateway
NAME AGE

```
bookinfo - gateway 32s
```

5. Run the following command to check the IP address of istio - ingressgat

```
eway .
```

```
$ kubectl get svc istio - ingressgat eway - n istio -
system
```

You can also log on to the Container Service console to view the IP address of istio - ingressgat eway . Specifically, choose Application > Service in the left-side navigation pane, select the target cluster and the Istio-system namespace.

Container Service -	Service List						Refresh Create
Kubernetes *	- Scivice List						INGI (GALE)
Overview	Clusters managed-cluster	Namespace istio-syste	m 🔻 3				
 Clusters 		ill optimize the security policy current cluster authorization in	in the near future, prohibiting unau nformation immediately	thorized sub-accounts from a	ccessing cluster resources.		
Clusters	Name	Туре	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint	Action
Nodes	grafana	ClusterIP	10/10/2018,11:34:40	172.21.12.155	grafana:3000 TCP		Details Update View YAML Delete
Volumes Namespace	istio-citadel	ClusterIP	10/10/2018,11:34:40	172.21.15.160	istio-citadel:8060 TCP istio-citadel:9093 TCP		Details Update View YAML Delete
Application Deployment	istio-egressgateway	ClusterIP	10/10/2018,11:34:40	172.21.14.36	istio-egressgateway:80 TCP istio-egressgateway:443 TCP		Details Update View YAML Delete
StatefulSet	istio-galley	ClusterIP	10/10/2018,11:34:40	172.21.12.142	istio-galley:443 TCP istio-galley:9093 TCP	-	Details Update View YAML Delete
Pods Service 2 Ingress Volumes Claim Helm Release Config Maps	E Isto-Ingresspateway	LoadBalancer	10/10/2018,11:34:40	172.21.15.225	Isb-operagications, 01 TCP isb-operagications(130 TCP isb-operagication(130 TCP) isb-operagication(130 TC	122,132 p0 122,132,440 132,132,440 122,132,1401 132,132,1401 122,132,1401 132,132,1401 122,132,2453 132,132,1503 132,132,1503 132,132,1503	Details Update View YAML Delete

6. Access the BookInfo home page. The access address is http://{ EXTERNAL - IP

}/ productpag e .

← → C ① 不安全 ■■■■■ //productpage	a 🗴 🗟 O 0 🖯 :
Bookinfo Sample	Sign in
The Comedy Summary: WKipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is his shortest and one addition to puns and word play. Book Details Type: paperback Pages: 200 Publisher: Publis	

If you refresh the browser, different versions of the reviews are displayed on the productpage in a round-robin manner (starting with a red star, to a black star, to no star). This indicates that Istio is currently not being used to control the version routing.

Set a route for requests

You need to set a default route because three Reviews service versions are deployed for the BookInfo sample application. Otherwise, if you access the application multiple times, you will notice that sometimes the book review output contains star ratings and other times it does not. This is because you have not set a default route for the rating service versions, and Istio then randomly routes requests to all available versions in a round robin fashion.

You need to define available versions in the destination routing rule before using Istio to control the route to the service versions of the BookInfo application.

Create the default destination routing rule for the BookInfo service.

· If you do not want to enable bidirectional TLS, run the following command:

\$ kubectl apply - f samples / bookinfo / networking /
destinatio n - rule - all . yaml

· If you want to enable bidirectional TLS, run the following command:

\$ kubectl apply - f samples / bookinfo / networking / destinatio n - rule - all - mtls . yaml

Wait for a few seconds until the destination routing rule takes effect. Run the following command to view the destination routing rule:

\$ kubectl get destinatio nrules - o yaml

Set the default version of all microservices to v1

Run the following command to set the default version of all microservices to v1:

```
$ kubectl apply - f samples / bookinfo / networking / virtual -
service - all - v1 . yaml
```

Run the following command to display all the created routing rules:

kubectl get virtualser vices - o yaml

It takes a period of time for the routing rule to be synchronized to all pods because the routing rule is distributed to the proxy in an asynchronized manner. Therefore, we recommend that you wait for a few seconds before accessing the application.

Open the URL of the Bookinfo application in your browser: http ://{ EXTERNAL - IP }/ productpag e .

On the product page of the BookInfo application, the displayed content does not contain the reviews with starts. This is because the reviews:v1 service does not access the ratings service.

BookInfo Sample	Sign in
The Comedy Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakesp with a major part of the humour coming from slapstick and mistaken identity, in ac Book Details Type: paperback Pages: 200 Publisher: PublisherA Language: English ISBN-10: 1234567890 ISBN-13: 123-1234567890	eare's early plays. It is his shortest and one of his most farcical comedies,

Route the requests from a specific user to reviews:v2

Run the following command to route requests from the test user named jason to reviews:v2 to enable the ratings service:

\$ kubectl apply - f samples / bookinfo / networking / virtual service - reviews - test - v2 . yaml

Run the following command to check whether routing rules are created:

```
get
$ kubectl
                  virtualser vice
                                      reviews
                                                     yaml
                                              - o
apiVersion : networking . istio . io / v1alpha3
kind : VirtualSer vice
metadata :
  name : reviews
  . . .
spec :
  hosts :

    reviews

  http :
 - match :
   - headers :
         end - user :
          exact : jason
    route :
      destinatio n :
        host : reviews
         subset : v2
    route :
      destinatio n :
        host : reviews
```

subset : v1

After you confirm that the routing rule is created, open the URL of the BookInfo application in your browser: http://{EXTERNAL - IP }/ productpag e.

Log on to the product page as the jason user to verify that the rating information is displayed under each review record.



Both the logon account name and password for are jason ..

← → C ① 不安全 2/productpage	아 🗟 🌣 🕒 😫
Bookinfo Sample	L Jason (sign out)
The Comedy Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is his shortest and or addition to puns and word play.	e of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in
Book Details Type: paperback Pages: 200 Publisher: PublisherA Language: English 1284567890 1281-13: 123-1234567890	Book Reviews An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!



In this example, two request routing rules have been changed. Firstly, all requests are routed to the v1 version of the Reviews service provided by the BookInfo application. Then, a new routing rule is set to route specific requests to the v2 version of the Reviews service according to the header of a request (for example, the user cookie).

Inject faults

To test the resiliency of the microservices application, namely, BookInfo, inject a 7-second delay between the reviews : v2 microservices and the ratings microservices for the jason user. Note that the reviews:v2 service has a 10-second hard-coded connection timeout for calls to the ratings service. Therefore, you can still expect the end-to-end flow to continue without any errors even you have set the 7second delay .

Inject an HTTP delay fault

Create a fault injection rule to delay traffic coming from the jason user.

samples / bookinfo / networking / virtual kubectl apply - f \$ service ratings - test - delay vaml

After you confirm that the rule is created, open the URL of the BookInfo application in your browser: http ://{ EXTERNAL - IP }/ productpag e .

Log on to the productpage as the jason user to view the following.

BookInfo Sample	▲ jason kign out)
The Comed Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is his shortest and addition to puns and word play.	
Book Details	Error fetching product reviews!
Type: paperback Pages: 200 Publisher: PublisherA Language: English 1SBN-10: 1234567890 123-1234567890	Sorry, product reviews are currently unavailable for this book.



Note: The reviews service fails because the timeout between the productpage and reviews services is shorter than the timeout between the reviews and ratings services, that is, (3 seconds + 1 retry = 6 seconds) is shorter than 10 seconds. Bugs like this can occur in typical enterprise applications where different teams develop different microservices independently. Istio's fault injection rules help you identify such anomalies without impacting end users.

Inject an HTTP abort fault

Create a fault injection rule to send an HTTP abort

samples / bookinfo / networking / virtual kubectl apply – f service - ratings - test - abort . yaml

After you confirm that the rule is created, open the URL of the BookInfo application in your browser: http ://{ EXTERNAL - IP }/ productpag e.

Log on to the productpage as the jason user to view the following.



Migrate traffic

In addition to the content-based routing rule, Istio also supports the weight-based routing rule.

Run the following command to route all traffic to the v1 version of all microservices.

\$ kubectl replace - f samples / bookinfo / networking / virtual - service - all - v1 . yaml

Run the following command to route 50% of traffic from the reviews v1 service to the reviews v3 service:

\$ kubectl replace - f samples / bookinfo / networking / virtual - service - reviews - 50 - v3 . yaml

Refresh the productpage for multiple times in the browser. You have a 50% probabilit y to see the review content marked with red stars on the page.



Note:

Note that this method is completely different from using the deployment feature of the container orchestration platform for version migration. The container orchestration platform uses the instance scaling method to manage the traffic. With istio, two versions of the reviews service can expand and shrink capacity independently, without affecting the distribution of traffic between the two versions of services. Assuming you decide that the reviews : v3 microservice is stable, you can route 100% of the traffic to reviews : v3 to implement a gray release by running the following command:

\$ kubectl replace - f samples / bookinfo / networking / virtual - service - reviews - v3 . yaml

Conclusion

You can use Alibaba Cloud Container Service for Kubernetes to quickly build the open platform, that is, Istio, to connect, manage, and secure microservices, and to introduce and configure multiple relevant services for applications. This topic uses a sample application from Istio to detail how to use Istio functions such as traffic rouging, fault injection, and traffic migrating. We recommend that you use Alibaba Cloud Container Service for Kubernetes to quickly build Istio, an open management platform for microservices, and integrate Istio with the microservice development of your project.

5.2 Implement Istio distributed tracking in Kubernetes

Background

Microservice is a focus in the current era. More and more IT enterprises begin to embrace the microservices. The microservice architecture splits a complex system into several small services and each service can be developed, deployed, and scaled independently. As a heaven-made match, the microservice architecture and containers (Docker and Kubernetes) further simplify the microservice delivery and strengthen the flexibility and robustness of the entire system.

When monolithic applications are transformed to microservices, the distributed application architecture composed of a large number of microservices also increases the complexity of operation & maintenance, debugging, and security management . As microservices grow in scale and complexity, developers must be faced with complex challenges such as service discovery, Server Load Balancer, failure recovery , indicator collection, monitoring, A/B testing, throttling, access control, and end-to-end authentication, which are difficult to resolve.

In May 2017, Google, IBM, and Lyft published the open-source service network architecture Istio, which provides the connection, management, monitoring, and security protection of microservices. Istio provides an infrastructure layer for services to communicate with each other, decouples the issues such as version management, security protection, failover, monitoring, and telemetry in applicatio n logics and service access. Being unrelated to codes, Istio attracts enterprises to transform to microservices, which will make the microservice ecology develop fast.

Architecture principle of Istio

In Kubernetes, a pod is a collection of close-coupled containers, and these containers share the same network namespace. With the extension mechanism of Initializer in Kubernetes, an Envoy container is automatically created and started for each business pod, without modifying the deployment description of the business pod. The Envoy takes over the inbound and outbound traffic of business containers in the same pod. Therefore, the microservice governance functions, including the traffic management, microservice tracking, security authentication, access control, and strategy implementation, are realized by operating on the Envoy.

Volumes Kestest Kubernetes China East 1 (Hangzhou) Vpc bjsrlal45z Running 6 10/12/2018,16:24:34 1.11.2 Data Scale Cluster Application managed-cluster ManagedKubernetes China North 2 (Beijing) VPC vpc- 2zef2e2y7vc Running 3 10/09/2018,11:20:00 1.11.2 Delete Add Existing In Upgrade Cluster ManagedKubernetes China North 2 (Beijing) VPC vpc- 2zef2e2y7vc Running 3 10/09/2018,11:20:00 1.11.2 Upgrade Cluster	Overview	*	Help: & Create cluster & S creation failures & Authorizat	Scale cluster 🔗 Connect to Kube tion management	rnetes cluster via	a kubecti 🔗 Ma	nage applicatio	ns with com	mands 🔗 Cluster plan	nning 🔗 Create (GPU clusters 🔗 Troubleshoot cluster
Nodes Cluster Name/ID Cluster Type Region (All) Cluster Number Number Kubernetes Volumes Namespace Kas-test Kubernetes China East 1 VPC vpc- top: P Running 6 10/12/2018/16:24:34 1.11.2 Manage View Scale Cluster Application managed-cluster ManagedKubernetes China East 1 VPC vpc- vpc- vgc/22e/22v/v P Running 3 10/09/2018/11:20:00 1.11.2 Delete Addom Users test-mia Kubernetes China East 1 VPC vpc- vgc/22e/2v/v Running 3 10/09/2018/11:20:00 1.11.2 Addom Users	 Clusters 	ų	Name 🔻								
Volumes K8s-test Kubernetes China East 1 (Hangzhou) Vpc- bptsr1al45z Running 6 10/12/2018,16:24:34 1.11.2 Data Namespace Application managed-cluster ManagedKubernetes China North 2 (Beijing) VPC vpc- 2zef2e2y7vc Running 3 10/09/2018,11:20:00 1.11.2 Delete Add Existing In Upgrade Cluster ManagedKubernetes China North 2 (Beijing) VPC vpc- 2zef2e2y7vc Running 3 10/09/2018,11:20:00 1.11.2 Add Existing In Upgrade Cluster		ł	Cluster Name/ID	Cluster Type		Network Type			Time Created		Action
Application Deployment StatefulSet test-mia Kubernetes China Bast 1 VPC VpC		U	k8s-test	Kubernetes		vpc-	Running	6	10/12/2018,16:24:34	1.11.2	Dashboard Scale Cluster More -
stateruiset			managed-cluster	ManagedKubernetes		vpc-	Running	3	10/09/2018,11:20:00	1.11.2	Delete Add Existing Instance Upgrade Cluster
Job (Hangzhou) bp1lkyevdjj Deploy Isto			test-mia	Kubernetes	China East 1 (Hangzhou)	vpc-	Running	7	09/17/2018,11:37:55	1.11.2	Automatic Scaling Addon Upgrade Deploy Istio

An Istio service mesh is logically split into a data plane and a control plane.

- The data plane is composed of a collection of intelligent proxies (Envoys) deployed as sidecars that mediate and control all network communication between microservices.
- The control plane is used to manage and configure the proxies to route traffic, and enforce polices at the runtime.

An Istio is mainly composed of the following components:

 Envoy: The Envoy is used to mediate all the inbound and outbound traffic for all the services in the service mesh. Functions such as dynamic service discovery, Server Load Balancer, fault injection, and traffic management are supported. The Envoy is deployed as a sidecar to the pods of related services.

- Pilot: The Pilot is used to collect and verify the configurations and distribute the configurations to all kinds of Istio components.
- Mixer: The Mixer is used to enforce the access control and usage policies in the service mesh, and collect telemetry data from Envoy proxies and other services.
- Istio-Auth: Istio-Auth provides strong service-to-service and end user authentication.

For more information about Istio, see the Istio official document.

Install Istio

Use an Alibaba Cloud Container Service Kubernetes cluster as an example.

Alibaba Cloud Container Service has enabled the Initializers plug-in by default for Kubernetes clusters if the cluster version is later than 1.8. No other configurations are needed.

📕 Note:

After you deploy the Istio, a sidecar is injected to each pod to take over the service communication. Therefore, we recommend that you verify this in the independent test environment.

Create a Kubernetes cluster

- 1. Log on to the Container Service console.
- 2. Under Kubernetes, click Clusters in the left-side navigation pane, and click Create Kubernetes cluster in the upper-right corner.
- 3. Configure the parameters to create a cluster. For how to create a Kubernetes cluster, see *Create a Kubernetes cluster*.

4. After the cluster is created, click Manage at the right of the cluster when the cluster status is changed to Running.

	_								
Container Service - Kubernetes 👻	1	Service List						Refresh	Create
Overview	•	Help: Ø Layer-4 canary releas	le						
 Clusters 		Clusters k8s-istio •	Namespace istio-system	m 🔻					
Clusters		Name	Туре	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint		Acti
Nodes		grafana	ClusterIP	10/15/2018,10:26:35	172.19.9.0	grafana:3000 TCP		Details Update View YAML	Delet
Volumes	Ľ	istio-citadel	ClusterIP	10/15/2018,10:26:35	172.19.1.199	istio-citadel:8060 TCP istio-citadel:9093 TCP		Details Update View YAML	Delet
Namespace	Ľ	istio-egressgateway	ClusterIP	10/15/2018,10:26:35	172.19.11.106	istio-egressgateway:80 TCP istio-egressgateway:443 TCP		Details Update View YAML	Delet
Application	Ľ	istio-galley	ClusterIP	10/15/2018,10:26:34	172.19.15.222	istio-galley:443 TCP istio-galley:9093 TCP	-	Details Update View YAML	Dele
Deployment StatefulSet Job Pods Service Ingress Volumes Claim		istio-ingresspiteway	LoadBalancer	10/15/2018,10:26-35	172.19.12.210	All-Approximations (1) TO this - hyperspectrum (1) the (1) this - hyperspectrum (1) the (1) the (1) the (1) t	40 45:443 48:1400 48:1511 48:1531 48:15300 48:15000 48:15001	Defails Update View Y201L	Dele

5. On the cluster Basic Information page, you can configure the corresponding connection information based on the page information. You can connect to the cluster either by using *Connect to a Kubernetes cluster by using kubectl* or *Access*

Kubernetes clusters by using SSH.

Container Service - Kubernetes - Overview Custers Custers Nodes	* isto-ingressgateway	LoadBalancer	10/15/2018,10:26:35	172.19.12.210	Istic-regression and the second secon	0 13460 10011 6860 10011 10000 10001	Details Update View VAML Delete
Volumes Namespace Authorization	istio-pilot	ClusterIP	10/15/2018,10:26:35	172.19.4.204	istio-pilot:15010 TCP istio-pilot:15011 TCP istio-pilot:8080 TCP istio-pilot:9093 TCP		Details i Update i View YAML i Delete
Application	istio-policy	ClusterIP	10/15/2018,10:26:35	172.19.14.150	istio-policy:9091 TCP istio-policy:15004 TCP istio-policy:9093 TCP		Details Update View YAML Delete
Deployment	istio-sidecar-injector	ClusterIP	10/15/2018,10:26:35	172.19.1.255	istio-sidecar-injector:443 TCP		Details Update View YAML Delete
Job	istio-statsd-prom-bridge	ClusterIP	10/15/2018,10:26:35	172.19.14.221	istio-statsd-prom-bridge:9102 TCP Istio-statsd-prom-bridge:9125 UDP		Details Update View YAML Delete
Pods Service	istio-telemetry	ClusterIP	10/15/2018,10:26:35	172.19.4.78	istio-telemetry:9091 TCP istio-telemetry:15004 TCP istio-telemetry:9093 TCP istio-telemetry:42422 TCP	-	Details Update View YAML Delete
Ingress	prometheus	ClusterIP	10/15/2018,10:26:35	172.19.10.115	prometheus:9090 TCP		Details Update View YAML Delete
Volumes Claim Helm	servicegraph	ClusterIP	10/15/2018,10:26:35	172.19.6.34	servicegraph:8088 TCP	-	Details Update View YAML Delete
Release Config Maps	tracing-on-sis-agent	ClusterIP	10/15/2018,10:26:35	172.19.10.85	tracing-on-sis-agent:5775 UDP tracing-on-sis-agent:6831 UDP tracing-on-sis-agent:6832 UDP tracing-on-sis-agent:5778 TCP		Details Update View YAML Delete
Secret	tracing-on-sis-collector	ClusterIP	10/15/2018,10:26:35	172.19.3.201	tracing-on-sis-collector:14267 TCP tracing-on-sis-collector:14268 TCP tracing-on-sis-collector:9411 TCP		Details Update View YAML Delete
Docker Images	tracing-on-sis-query	LoadBalancer	10/15/2018,10:26:35	172.19.2.255	tracing-on-sis-quary:80 TCP tracing-on-sis-quary:30258 TCP	1:80	Details Update View YAML Delete

Deploy Istio release version

Log on to the master node and run the following command to get the latest Istio installation package.

curl - L https :// git . io / getLatestI stio | sh -

Run the following command:

cd istio - 0 . 4 . 0 working directory to Istio ## Change the

export PATH =\$ PWD / bin :\$ PATH ## Add the istioctl client to PATH environmen t variable

Run the following command to deploy Istio.

```
kubectl
          apply – f
                       install / kubernetes / istio . yaml
  ##
      Deploy
               Istio
                       system
                                components
          apply - f
kubectl
                       install / kubernetes / istio - initialize
                                                                   r.
yaml
          ## Deploy
                       Istio
                               initialize r
                                                plug - in
```

After the deployment, run the following command to verify if the Istio components are successfully deployed.

```
pod
                                      istio - systemNAME
             get
Ś
  kubectl
                    svc ,
                                 'n
                                                             TYPE
                 EXTERNAL - IP PORT (S) AGEsvc / istio - ingress
 CLUSTER - IP
                     172 . 21 . 10 . 18
                                           101 . 37 . 113 . 231
   LoadBalanc
              er
                                                                     80 :
 30511 / TCP , 443 : 31945 / TCP
                                     1msvc / istio - mixer
                                                               ClusterIP
   172 . 21 . 14 . 221
                           9091 / TCP , 15004 / TCP , 9093 / TCP , 9094
  TCP , 9102 / TCP , 9125 / UDP , 42422 / TCP 1msvc / is
ClusterIP 172 . 21 . 4 . 20 15003 / TCP , 443 / TCP
                                                    1msvc / istio - pilot
                                                                   1mNAME
                             AGEpo / istio - ca - 55b954ff7 - crsjq
                   RESTARTS
 READY
         STATUS
         Running
                        1mpo / istio - ingress - 948b746cb - 4t24c
 1 / 1
                    0
 / 1
       Running
                 0
                      1mpo / istio - initialize r - 6c84859cd - 8mvfj
           Running
   1 / 1
                      0
                          1mpo / istio - mixer - 59cc756b48 - tkx6c
                                                                          3
       Running
                      1mpo / istio - pilot - 55bb7f5d9d - wc5xh
                                                                      2 / 2
  3
                  0
   Running
             0
                  1m
```

After all the pods are in the running status, the Istio deployment is finished.

Istio distributed service tracking case

Deploy and test the application BookInfo

BookInfo is an application similar to an online bookstore, which is composed of several independent microservices compiled by different languages. The application BookInfo is deployed in the container mode and does not have any dependencies on Istio. All the microservices are packaged together with an Envoy sidecar. The Envoy sidecar intercepts the inbound and outbound call requests of services to demonstrate the distributed tracking function of Istio service mesh.

For more information about BookInfo, see Bookinfo guide.

Clusters								
	•							
Namespace								
istio-system								
Release Name								
istio								
Version								
1.0.3								
Enable Prometheus for metrics/logs collection								
Enable Grafana for metri	cs display							
🖉 Enable automatic Istio S	idecar injection							
Enable the Kiali Visualization Service Grid								
Enable Log Service(SLS)	and Jaeger							
* Endpoint	cn-hangzhou.log.aliyuncs.com							
* Project								
* Logstore	AND THE REPORT OF A							
* AccessKeyID								
* AccessKeySecret								

Run the following command to deploy and test the application Bookinfo.

kubectl apply - f samples / bookinfo / kube / bookinfo . yaml

In the Alibaba Cloud Kubernetes cluster environment, every cluster has been configured with the Server Load Balancer and Ingress. Run the following command to obtain the IP address of Ingress.

```
$ kubectl get ingress - o wide
NAME HOSTS ADDRESS PORTS AGE
```

gateway * 101.37.xxx.xxx 80 2m

If the preceding command cannot obtain the external IP address, run the following command to obtain the corresponding address.

```
export GATEWAY_UR L =$( kubectl get ingress - o wide - o
jsonpath ={. items [ 0 ]. status . loadBalanc er . ingress [ 0 ]. ip
})
```

The application is successfully deployed if the following command returns 200.

```
curl - o / dev / null - s - w "%{ http_code }\ n " http ://${ GATEWAY_UR L }/ productpag e
```

You can open http://\${ GATEWAY_UR L }/ productpag e in the browser to access the application. GATEWAY_URL is the IP address of Ingress.

aeger UI Lookup by Trace ID Sea	ch Dependencies	About Jaeger 🗸
Find Traces		
productpage	500ms	
Operation (4)		Tic
all	04.26:40 pm 04.35:00 pm 04.43:20 pm	
Tags 🕐	10 Traces	Sort: Most Recent V
http.status_code=200 error=true		
Lookback	productpage: productpage.default.svc.cluster.local:9080/productpage	45.89ms
Last Hour	Today 4:49:50 pm 8 minutes ago	
Min Duration		
e.g. 1.2s, 100ms, 500us	productpage: productpage.default.svc.cluster.local:9080/productpage	39.26ms
Max Duration	Today 4:49:49 pm 8 minutes ago	
e.g. 1.1s		
Limit Results	productpage: productpage.default.svc.cluster.local:9080/productpage	26.33ms
20	Today 4:49:48 pm 8 minutes ago	
Find Traces	productpage: productpage default svc.cluster.local:9080/productpage	43.62ms
	Today 4:49:47 pm 8 minutes apo	

Deploy Jaeger tracking system

Distributed tracking system helps you observe the call chains between services and is useful when diagnosing performance issues and analyzing system failures.

Istio ecology supports different distributed tracking systems, including *Zipkin* and *Jaeger*. Use the Jaeger as an example.

Istio version 0.4 supports Jaeger. The test method is as follows.

```
kubectl apply - n istio - system - f https :// raw .
githubuser content . com / jaegertrac ing / jaeger - kubernetes /
master / all - in - one / jaeger - all - in - one - template . yml
```

After the deployment is finished, if you connect to the Kubernetes cluster by using kubectl, run the following command to access the Jaeger control panel by using port mapping and open http://localhost:16686 in the browser.

```
kubectl port - forward - n istio - system $( kubectl get
pod - n istio - system - l app = jaeger - o jsonpath ='{.
items [0]. metadata . name }') 16686 : 16686 &
```

If you connect to the Alibaba Cloud Kubernetes cluster by using SSH, run the following command to check the external access address of jaeger-query service.

```
$ kubectl
                               istio - system
                   svc – n
             get
NAME
                       TYPE
                                        CLUSTER - IP
                                                             EXTERNAL -
IΡ
           PORT (S)
            AGE
jaeger - agent
                         ClusterIP
                                                            < none >
                                          None
        5775 / UDP , 6831 / UDP , 6832 / UDP
                 1h
 jaeger - collector
                         ClusterIP
                                          172 . 21 . 10 . 187
                                                                  < none
              14267 / TCP , 14268 / TCP , 9411 / TCP
                        1h
                                            172 . 21 . 10 . 197
 jaeger - query
                         LoadBalanc er
                                                                      114
 . 55 . 82 . 11
                       80 : 31960 / TCP
                                              ## The
                                                       external
                                                                  access
   address
                  114 . 55 . 82 . 11 : 80 .
             is
                       ClusterIP
                                                          < none >
 zipkin
                                        None
      9411 / TCP
```

Record the external access IP address and port of jaeger-query and then open the application in the browser.

By accessing the application BookInfo for multiple times and generating the call chain information, we can view the call chain information of services clearly.

Trace Start: October 15, 2018 4:18 PM Duration: 1.02s				700.40-		
Oms	254.04ms	508.08ms		762.13ms		1.0
		_				×
Service & Operation	Oms	254.04ms	508.08ms		762.13ms	1.0
istio-ingressgateway productpage.default.svc.cluster.local						
productpage productpage.default.svc.cluster.local:9080						
productpage async outbound[9091][istio-policy.istio	. 1.27ms					
V istio-policy Check	0.97ms					
v istio-mixer /istio.mixer.v1.Mixer/Check	1 0.23ms					
istio-mixer kubernetes:handler.kubernete	. 1 0.07ms					
productpage details.default.svc.cluster.local:9080/*	6.46ms					
✓ details details.default.svc.cluster.local:9080/*	5.66ms					
details async outbound[9091][istio-policy.istio	- I 1.73ms					
V istio-policy Check	I 1ms					
 istio-mixer /istio.mixer.v1.Mixer/Check 	I 0.28ms					
istio-mixer kubernetes:handler.ku	. I 0.07ms					
productpage reviews.default.svc.cluster.local:9080/*	ns 🗖					
✓ reviews reviews.default.svc.cluster.local:9080/*	ns 🗲					
reviews async outbound[9091][istio-policy.isti	I 1.98ms					
V istio-policy Check	I 1.07ms					
✓ istio-mixer /istio.mixer.v1.Mixer/Check	I 0.3ms					
istio-mixer kubernetes:handler.ku	. I 0.08ms					
reviews ratings.default.svc.cluster.local:9080/*						7.4ms 📒
✓ ratings ratings.default.svc.cluster.local:90						5.76ms 🛢
ratings async outbound[9091][istio-pol						2.66ms
V istio-policy Check						1.71ms
istio-mixer /istio.mixer.v1.Mix						0.36ms
istio-mixer kubernetes:ha						0.11ms

Click a specific Trace to view the details.



You can also view DAG.



Implementation principle of Istio distributed tracking

The kernel of Istio service mesh is the Envoy, which is a high-performance and opensource Layer-7 proxy and communication bus. In Istio, each microservice is injected with an Envoy sidecar and this instance is responsible for processing all the inbound and outbound network traffic. Therefore, each Envoy sidecar can monitor all the API calls between services, record the time required by each service call, and record whether each service call is successful or not.

Whenever a microservice initiates an external call, the client Envoy will create a new span. A span represents the complete interaction process between a collection of microservices, starting from a caller (client) sending a request to receiving the response from the server.

In the service interaction process, clients record the request start time and response receipt time, and the Envoy on the server records the request receipt time and response return time.

Each Envoy distributes their own span view information to the distributed tracking system. When a microservice processes requests, other microservices may need

to be called, which causes the creation of a causally related span and then forms the complete trace. Then, an application must be used to collect and forward the following Headers from the request message:

- x request id
- x b3 traceid
- x b3 spanid
- x b3 parentspan id
- x b3 sampled
- x b3 flags
- x ot span context

Envoys in the communication links can intercept, process, and forward the corresponding Headers.



For specific codes, see the Istio document https://istio.io/docs/tasks/telemetry/distributed-tracing.html.
Conclusion

Istio is accelerating the application and popularization of service mesh by using the good expansion mechanism and strong ecology. In addition to those mentioned in the preceding sections, Weave Scope, Istio Dashboard, and Istio-Analytics projects provide abundant call link visualization and analysis capabilities.

5.3 Integrate Istio with Alibaba Cloud Log Service on Kubernetes

With Alibaba Cloud Container Service for Kubernetes, you can integrate Istio with Log Service. This topic uses an official example to describe how to integrate Istio with the distributed tracing system based on Alibaba Cloud Log Service.

Prerequisites

- You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.
- You have a local Linux environment in which you have configured the kubectl tool and connected to the cluster with kubectl. For more information, see *Connect to a Kubernetes cluster by using kubectl*.
- You have downloaded the project codes of an Istio version and run the relevant commands in the Istio file directory. For more information, see https://github.com/istio/istio/releases.
- You have deployed the test application, BookInfo. For more information, seeInstall the Istio official sample application or Bookinfo Application .

Introduction to OpenTracing

The OpenTracing specification has been created to address API incompatibility between different distributed tracing systems. OpenTracing is a lightweight and standardized layer that is located between applications/class libraries and tracing or log analysis programs. OpenTracing has joined the Cloud Native Computing Foundation (CNCF) and provided unified concepts to provide data standards for global distributed tracing systems. It provides platform-independent and vendorindependent APIs. With OpenTracing, developers can easily add or replace tracing systems. Jaeger is an open source distributed tracing system released by CNCF, and is compatible with OpenTracing APIs.

Alibaba Cloud Log Service and the distributed tracing system Jaeger

Log Service (LOG for short, formerly called SLS) is a one-stop service for real-time data. It has been used extensively by Alibaba Group in big data scenarios. Log service helps you collect, consume, ship, query, and analyze massive log data with improved processing capabilities.

Jaeger is an open source distributed tracing system released by Uber for microservice scenarios. It is used to analyze the calling process of multiple servers and display the service calling tracks with graphs. It is a powerful tool for diagnosing performance and analyzing system failures.

Jaeger on Aliyun Log Service is a distributed tracing system developed on Jaeger. With this system, you can store the collected tracing data to Alibaba Cloud Log Service permanently, and query and present the data through the native interfaces of Jaeger.



Jaeger on Aliyun Log Service

Component	Description
Jaeger client	The Jaeger client provides SDKs that conform to the OpenTracing standard for different languages. Applications write data to the client through APIs. The client library delivers the traces to the Jaeger agent according to the sampling policy specified by the applications. The tracing data is serialized by Thrift and transmitted through UDP.
Jaeger agent	The Jaeger agent is a network daemon that listens to spans sent over UDP and sends data in batches to the Jaeger collector. The Jaeger agent is designed as an infrastructure component that is deployed to all hosts. The Jaeger agent decouples the Jaeger client and Jaeger collector, and makes the collectors invisible to the client. Collectors are then no longer routed or discovered by the client.
Jaeger collector	The Jaeger collector receives traces from the Jaeger agent and writes the traces into the backend storage system . The backend storage is a pluggable component. Jaeger on Aliyun Log Service supports Alibaba Cloud Log Service.
Alibaba Cloud Log Service	The Jaeger collector stores the received spans to Log Service permanently. Query retrieves data from Log Service.
Query & UI	Query is a service that receives query requests, retrieves traces from the backend storage system, and hosts a UI to display traces.

Step 1: Deploy Istio with a single click through the Web interface

1. Log on to the Container Service console.

2. In the left-side navigation pane, choose Clusters, select a cluster, and click More > Deploy Istio.



In this example, you need to enable Log Service (SLS) and Jaeger. For more information, see *Deploy Istio*. This example uses Istio V1.0.

Container Service - Kubernetes +		Cluster List			You can create u	p to 5 clusters	and can ad	d up to 40 nodes in each	cluster. Refresh	Create Kubernetes Cluster
Overview	•	Help: & Create cluster & Scale cluster & Connect to Kubernetes cluster via kubect & Manage applications with commands & Cluster planning & Create GPU clusters & Troubleshoot cluster creation failures & Authorization management								
Clusters	l	Name 🔻								
Clusters	l	Cluster Name/ID	Cluster Type	Region (All)	Network Type	Cluster Status	Number of Nodes	Time Created	Kubernetes Version	Action
Volumes Namespace	-	k8s-test c9a8fedbf98d44aa6be1a46b8aa9cf862	Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp1sr1al45z	Running	6	10/12/2018,16:24:34	1.11.2	Manage View Logs Dashboard Scale Cluster More -
 Application Deployment 		managed-cluster c8b9a7701ede14c0eb19e60bf37b63951	ManagedKubernetes	China North 2 (Beijing)	VPC vpc- 2zef2e2y7vc	Running	3	10/09/2018,11:20:00	1.11.2	Delete Add Existing Instance Upgrade Cluster
StatefulSet Job		test-mia cad7e15c0784848a5be02443e9186ccb7	Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp1kyevdjj	Running	7	09/17/2018,11:37:55	1.11.2	Automatic Scaling Addon Upgrade Deploy Istio
Pods										

3. On the Istio deployment page, select Log Service(SLS) and Jaeger.

Container Service for Kubernetes is integrated with Log Service. Its distributed tracing data is saved to the Logstore of Log Service.

Clusters						
-	•					
Namespace						
istio-system						
Release Name						
istio						
Version						
1.0.3						
 Enable Grafana for metrics display Enable automatic Istio Sidecar injection Enable the Kiali Visualization Service Grid 						
Enable Log Service(SLS)	and Jaeger					
* Endpoint	cn-hangzhou.log.aliyuncs.com					
* Project						
* Logstore						
* AccessKeyID						
* AccessKeySecret						

The descriptions of Log Service parameters are as follows.

Name	Parameter	Description
Endpoint	storage . aliyun_sls . endpoint	Specifies the endpoint in which the Project resides to store spans.

Name	Parameter	Description
Project	storage . aliyun_sls . project	Specifies the Project that stores spans. The Project name must be a string of 3 to 63 bytes and must contain only lowercase letters, numbers, and hyphens (-). In addition, the name must start and end with lowercase letters or numbers.
Logstore	storage . aliyun_sls . logstore	Specifies the Logstore that stores spans. The Logstore name must be a string of 3 to 63 bytes that must contain lowercase letters, numbers, hyphens (-), and underscores (_). The name must start and end with lowercase letters or numbers. A Logstore name must be unique in the Project to which it belongs.
AccessKeyID	storage . aliyun_sls . accesskey . id	Specify the user Access Key ID.
AccessKeySecret	storage . aliyun_sls . accesskey . secret	Specify the user Access Key Secret.

Note:

If you specify a Project that does not exist, the system automatically creates and initializes a new Project.

- 4. Click Deploy Istio. After a few minutes, the Istio integrated with Alibaba Cloud Log Service is created.
- 5. In the left-side navigation pane, when you choose Application > Service, the Istio services, including the integrated Log Service, are displayed on the right.

Step 2: View Jaeger on Aliyun Log Service

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane, click Application > Service, click a cluster and the istio-system namespace, and then click the external endpoint of the tracing-on-sls-query service.
- 3. On the left side, select a service from the Service drop-down list, set other parameters, and click Find Traces. The scatter diagram of time and duration displayed in the upper-right corner displays the results and provides drill-down capability.



In this example, the Bookinfo sample application is used to view the productpage service calling.

Jaeger UI Lookup by Trace ID	Search Dependencies	About Jaeger v
Find Traces Service (8) productpage Operation (4) all	V 500ms 09:41:40 am 09:43:20 am	09:45:00 am
Tags ⑦ http.status_code=200 error=true	20 Traces	Sort: Most Recent
Lookback	productpage: productpage.default.svc.cluster.local:9080/productpage	20.15ms
Last Hour	V Today 9:46:05 am 14 minutes ago	
e.g. 1.2s, 100ms, 500us		45.65ms
o.g. 1.23, 100113, 00005	productpage: productpage.default.svc.cluster.local:9080/productpage	40.00115
Max Duration e.g. 1.1s	Today 9:46:01 am 14 minutes ago	
Limit Results	productpage: productpage.default.svc.cluster.local:9080/productpage	27.68ms

4. You can select from multiple views that display trace results in different ways. For example, you can select a histogram within a tracing period, or the service duration time of the tracing process.

istio-ingressgateway: oductpage.default.svc.c	luster.local:9080/pro	oductoage		æ	View Op	otions v Sea	arch
e Start: July 31, 2018 9:46 AM Duration: 20.76	•						
	5.19ms	10.38ms		15.	57ms		20.7
			_				
vice & Operation	Oms	5.19ms		10.38ms		15.57ms	20.7
stio-ingressgateway productpage.default.svc.cluster							
productpage productpage.default.svc.cluster.local:	. ns C						
v productpage details.default.svc.cluster.local:90			4.5ms				
details details.default.svc.cluster.local:9080/*			3.11ms				
	details.default.svc.cluste	er.local:9080/*		Servi	ce: details	Duration: 3.11ms	Start Time: 5.44
	<pre>> Tags: guid:x-request-id=b0e8544 > Process: ip=-1408236812</pre>	0b-e260-9118-9c99-ad3d5609	e2cd respons	e_flags = - span.kind = serv	er node_id	=sidecar~172.16.2	244~details-v1-7
v productpage reviews.default.svc.cluster.local:9				6.08ms			•
reviews reviews.default.svc.cluster.local:9080/*				5.27ms			
	reviews.default.svc.clus	ter local 9080/*	Service	reviews	Juration: 5.27ms	Start Time: 12.46	

Step 3: View Log Service

- 1. Log on to the Log Service console.
- 2. Select a Project and click the Project name.
- 3. Click Search on the right of a Logstore.
- 4. Select a log query time range, and then click Search & Analysis.

With Log Service, you can use query and analysis statements and other extended functions to optimize your query. For example, you can use analysis charts.

The following is a statistical chart.

Conclusion

Alibaba Cloud Container Service for Kubernetes can integrate Istio with Log Service . This topic uses an official example to describe how to integrate Istio with the distributed tracing system based on Alibaba Cloud Log Service.

We recommend that you use Alibaba Cloud Container Service for Kubernetes to quickly build Istio, an open management platform for microservices, and integrate Istio with the microservice development of your project.

5.4 Use Istio to deploy application services across Kubernetes and ECS instances

Starting from v0.2, Istio provides mesh expansion. With this feature, you can integrate non-Kubernetes services that typically run on VMs or bare metal hosts with the Istio service mesh that runs on your Kubernetes cluster.

Alibaba Cloud Container Service for Kubernetes supports the Istio mesh expansion capabilities. This topic uses an example from the Istio official website to details how to use Istio to deploy application services across Kubernetes and ECS instances.

Mesh expansion

Mesh expansion is a method based on the Istio service mesh deployed on Kubernetes. With this method, you can integrate VMs or bare metal hosts into the service mesh.

Mesh expansion is suitable for when you need to migrate your applications from your local system to cloud services. In a microservices system, not all workloads can run in Kubernetes. This means you may encounter scenarios in which you can only operate and maintain some services in Kubernetes, while other services run on VMs or bare metal hosts.

With the Istio control plane, you can manage services across Kubernetes and VMs or bare metal hosts, and ensure that all your services can continue to run normally.

Create a Kubernetes cluster and install Istio

Alibaba Cloud Container Service for Kubernetes 1.11.5 is now available. You can quickly create a Kubernetes cluster through the Container Service console. For more information, see *Create a Kubernetes cluster*.



You must make sure that you can connect to your Kubernetes cluster by using kubectl. For more information, see *Connect to a Kubernetes cluster by using kubectl*.

Deploy Istio through the app catalog. Create the istio - system namespace through a command or the console.

- 1. Log on to the Container Service console.
- In the left-side navigation pane, choose Store > App Catalog, and click ack istio on the right side.

3. On the displayed page, select istio - system from the namespace drop-down list, and click Values. You can edit parameters to customize your Istio.

Note:

The readme document on the page provides the installation and removal information, including common questions about Custom Resource Definition (CRD) versions.

Install the sample application in your Kubernetes cluster

Run the following commands or use the console to create the bookinfo namespace, and then deploy the modified application. In the modified application, the details component is removed and ingressgat eway is defined.

To obtain the files used in this example, see Istio multi-cluster sample files.

```
bookinfo
kubectl
         create
                  ns
                 namespace bookinfo
                                       istio - injection = enabled
kubectl
         label
kubectl
         apply – n
                      bookinfo - f ./ bookinfo / bookinfo -
without - details . yaml
                      bookinfo - f ./ bookinfo / bookinfo -
kubectl
         apply – n
gateway . yaml
```

Both the details and the database components of the application deployment run on the ECS instance that is outside the Kubernetes system.



Access the / productpag e page through the address exposed by ingressgat eway and verify that the details part cannot be displayed.

BookInfo Sample	Sign in
The Come Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare part of the humour coming from slapstick and mistaken identity, in addition to puns ar	
Error fetching product details!	Book Reviews
Sorry, product details are currently unavailable for this book.	An extremely entertaining play by Shakespeare. The slapstick humour is refreshing! - Reviewer1 ★ ★ ★ ★ ★
	Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.

– Reviewer2
★ ★ ★ ★ ☆

Configure your Kubernetes

1. If you have not set internal load balancers for Kube DNS, Pilot, Mixer, and Citadel when you install Istio, you need to run the following command:

kubectl apply - f ./ mesh - expansion . yaml

As shown in the following figure, the four services are created.

```
ali-1c36bbed0b91:meshexpansion wangxn$ kubectl apply -f ./mesh-expansion.yaml
service "istio-pilot-ilb" created
service "dns-ilb" created
service "mixer-ilb" created
service "citadel-ilb" created
```

2. Generate the cluster . env Istio configuration file and the kubedns DNS

 $configuration \ file \ both \ of \ which \ are \ to \ be \ deployed \ in \ the \ VMs. \ The \ \ cluster \ .$

env file contains the range of the cluster IP addresses that will be intercepted.

The *kubedns* file contains the cluster service names that can be resolved by the

applications on the VMs and then will be intercepted and forwarded by the sidecar.

To generate the configuration files, run the following command:

./ setupMeshE x . sh generateCl usterEnvAn dDnsmasq

Configuration file cluster . env



Configuration file kubedns



Set the ECS instance

Configure your working environment to communicate with the ECS instance.

Generate an SSH key and assign it to the ECS instance. You can run the ssh root

```
@< ECS_HOST_I P > command to check if you can connect to the ECS instance.
```

To generate a public key, run the following command:

ssh - keygen - b 4096 - f ~/.ssh / id_rsa - N ""

Note:

To ensure that the ECS instance and Kubernetes are mutually accessible over the Internet, you need to add them to the same security group.

With Alibaba Cloud Container Service for Kubernetes, you can quickly configure an ECS instance by running the following script:

```
export SERVICE_NA MESPACE = default
./ setupMeshE x . sh machineSet up root @< ECS_HOST_I P >
```

Run the following command to check the running process:

ps	aux		grep	istio	
root@remotev	vm:~# ps aux	lgrep i	stio		
root 194	460 0.0 0.0	0 52284	3404 ?	Ss 11:59	0:00 su -s /bin/bash -c INSTANCE_IP=192.168.3.70 POD_NAME=remotevm POD_NAMESPACE=default exec /u
sr/local/bi	n/pilot-agent	t proxy	se	rviceCluster rawvr	ndiscoveryAddress istio-pilot.istio-system:8080controlPlaneAuthPolicy MUTUAL_TLS
2> /var/log/	/istio/istio	.err.log	> /var/l	og/ <mark>istio/istio</mark> .log	j istio-proxy
istio-p+ 195	508 0.0 0.0	0 45276	4592 ?	Ss 11:59	0:00 /lib/systemd/systemduser
istio-p+ 19	510 0.0 0.0	0 61324	2064 ?	S 11:59	0:00 (sd-pam)
istio-p+ 19	516 0.0 0.2	2 31204	17252 ?	Ssl 11:59	0:00 /usr/local/bin/pilot-agent proxyserviceCluster rawvmdiscoveryAddress istio-pilot.isti
-system:808	80control	PlaneAut	hPolicy M	UTUAL_TLS	
istio-p+ 19	537 7.1 0.5	5 133208	41572 ?	Sl 11:59	0:02 /usr/local/bin/envoy -c /etc/istio/proxy/envoy-rev1.jsonrestart-epoch 1drain-time-s 2
parent-sl	hutdown-time	-s 3s	ervice-cl	uster rawvmserv	/ice-node sidecar~192.168.3.70~remotevm.default~default.svc.cluster.localmax-obj-name-len 189 -l
warnv2-o	config-only				

Run the following command to check if the node agent authenticated by Istio is running in a healthy status:

sudo systemctl status istio - auth - node - agent

Run services on the ECS instance

As shown in the preceding deployment figure, two services run on the ECS instance: one is the Details service, the other one is the Database service.

Run the Details service on the ECS instance

Run the following commands to simulate (by using Docker only) the Details service, run the service on the ECS instance, and expose port 9080 for the service.

docker pull istio / examples - bookinfo - details - v1 : 1 . 8 .
0
docker run - d - p 9080 : 9080 -- name details - on - vm
istio / examples - bookinfo - details - v1 : 1 . 8 . 0

Configure the sidecar to intercept the port. You need to configure this in the / var

/ lib / istio / envoy / sidecar . env path and use the ISTIO_INBO

UND_PORTS environment variable.

Run the following command on the VM in which the service runs:

```
echo " ISTIO_INBO UND_PORTS = 9080 , 8080 " > / var / lib / istio /
envoy / sidecar . env
systemctl restart istio
```

Register the Details service with Istio

Run the following command to view the IP address of the VM so that you can add it to the service mesh:

hostname – I

Manually configure a selector-less service and endpoints. The selector-less service is used to host services that are not backed by Kubernetes pods. For example, run the following command to register the Details service on a server that has the permissions to modify Kubernetes services and supports isticctl commands:

```
istioctl - n bookinfo register details 192 . 168 . 3 . 202
http : 9080
```

Access the / productpag e page again to verify that the details part is displayed as shown in following figure.

Sign in

BookInfo Sample

The Comedy of Errors



Update the Ratings service to the version that can access a database

By default, the Ratings service cannot access any database. Run the following command to update the service version so that the service can access the database:

```
kubectl apply - f ./ bookinfo / bookinfo - ratings - v2 - mysql -
vm . yaml
kubectl apply - f ./ bookinfo / virtual - service - ratings -
mysql - vm . yaml
```

Access the / productpag e page to verify that the Ratings part cannot be displayed as shown in the following figure. Then, you need to build a database service on the ECS instance and add the service to Istio.

BookInfo Sample	👤 jason (sign out)
The Comedy Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is slapstick and mistaken identity, in addition to puns and word play. Book Details	
Type: paperback Pages: 200 Publisher: PublisherA Language: English ISBN-10: 1234567890 ISBN-13: 123-1234567890	An extremely entertaining play by Shakespeare. The slapstick humour is refreshing! Reviewert Ratings service is currently unavailable Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare. Reviewer2 Ratings service is currently unavailable

Run a database service on the ECS instance

On the VM, run MariaDB as the backend for the Ratings service, and set MariaDB to be remotely accessible.

apt - get update && apt - getinstall - y mariadb - server

```
sed - i ' s / 127 \. 0 \. 0 \. 1 / 0 \. 0 \. 0 \. 0 \ g ' /
etc / mysql / mariadb . conf . d / 50 - server . cnf
sudo mysql
# Grant the root permission .
GRANT ALL PRIVILEGES ON *. * T0 ' root '@' localhost '
IDENTIFIED BY ' password ' WITHGRANTO PTION ;
quit ;
sudo systemctl restart mysql
```

Run the following command to initialize the Ratings database on the VM:

```
curl - q https :// raw . githubuser content . com / istio / istio
/ master / samples / bookinfo / src / mysql / mysqldb - init . sql
| mysql - u root - ppassword
```

To view different outputs of the Bookinfo application, run the following command to modify the rating records to generate different rating data that are displayed on the page:

mysql - u root - ppassword test - e " select * from ratings ;" mysql - u root - ppassword test - e " update ratings set rating = 2 ; select * from ratings ;"

Register the database service into Istio

Configure the sidecar to intercept the port. You need to configure this in the / var

/ lib / istio / envoy / sidecar . env path and use the ISTIO_INBO

UND_PORTS environment variable.

Run the following command on the VM in which the service runs:

```
echo " ISTIO_INBO UND_PORTS = 3306 , 9080 , 8080 " > / var / lib /
istio / envoy / sidecar . env
systemctl restart istio
```

Run the following command to register the database service on a server that has the permissions to modify Kubernetes services and supports isticctl commands:

istioctl - nbookinfor egistermys qldb 192 . 168 . 3 . 202 3306

Now Kubernetes pods and other servers included by mesh expansion can access the database service running on this server.

Access the / productpag e page to verify that both the Details and Ratings parts can be displayed and these two services are provided by the ECS instance.

BookInfo Sample		💄 jason (sign out)
Summary: Wikipedia Summary: The Cor slapstick and mistaken identity, in additi	nedy of Errors is one of William Shakespear	he Comedy of Errors re's early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming fro
	Book Details	Book Reviews
Iype: baperback Pages: 200 Publisher: ublisherA anguage: English SBN-10: 1234567890 SBN-13: 123-1234567890		An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!

Conclusion

Alibaba Cloud Container Service for Kubernetes provides the Istio mesh expansion capabilities. This topic uses a sample application from the Istio official website to details how to use Istio to deploy application services across Kubernetes and ECS instances.

We recommend that you use Alibaba Cloud Container Service for Kubernetes to quickly build Istio, an open management platform for microservices, and integrate Istio with the microservice development of your project.

5.5 Use the Canary method that uses Istio to deploy a service

This topic describes how to use the Canary method (that uses Istio) to deploy a service through Alibaba Cloud Container Service for Kubernetes (ACK).

Background information

Istio is a service mesh that can be used to meet the requirements of the distribute d application architectures that involve microservices such as application O&M , debugging, and security management. You also can use Istio for microservice network scenarios such as load balancing, service-to-service authentication, and monitoring.

By using Istio, ACK allows you to manage services of an application. With ACK, you can create and manage multiple versions of one service for an application, manage the distribution of traffic destined for the service, and implement a Canary release for the service.

Prerequisites

- A Kubernetes cluster is created. For more information, see *Create a Kubernetes cluster*.
- Istio is deployed for this Kubernetes cluster. For more information, see *Deploy Istio*.

Procedure

- 1. Enable automatic sidecar injection.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane under Container Service-Kubernetes, choose Clusters > Namespaces.
 - c. Click Create.
 - d. In the displayed dialog box, set Name to demo, and set Tag to follows:
 - The variable name of the tag is set to istio injection .
 - $\cdot \,$ The variable value of the tag is set to $\,$ enabled .

Create Names	pace			\times
Name	demo]
Tags	1-63 characters, can o can only be letters or Variable Name		er case letters, and "-", and and end Action	
	istio-injection	enabled	Add	1
			2 ок	Cancel

e. Click Add, and then click OK.

Then, the newly created namespace is displayed in the namespace list.

Container Service - Kubernetes 👻	Na	mespaces				Refr	resh	Create
Overview	He	p: \mathscr{O} Configure resource quotas for namespaces						
 Clusters 	Clust	ers test-01 •						
Clusters	Na	ne Tag	Status	Time Created				Action
Nodes	def	ault	Ready	02/28/2019,14:50:52	ResourceQuota and LimitRang	e	Edit	Delete
Volumes	der	istio-injection: enabled	Ready	04/26/2019,16:39:07	ResourceQuota and LimitRang	e	Edit	Delete
Namespaces	isti	-system name: istio-system	Ready	04/01/2019,16:21:52	ResourceQuota and LimitRang	e I	Edit	Delete
Authorization Applications	kul	e-public	Ready	02/28/2019,14:50:52	ResourceQuota and LimitRang	e	Edit	Delete
Deployments	kul	e-system	Ready	02/28/2019,14:50:52	ResourceQuota and LimitRang	e	Edit	Delete

- 2. Create a server-end application and its virtual service.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane, choose Service Mesh > Virtual Service.
 - c. In the upper-right corner of the page, click

Create.	Container Service - Kubernetes 🔻		Virtual	Service		
	Overview	•	Clusters	test-01	•	Namesp
	 Clusters 		Name			Worklo
	 Applications 					
	 Service Mesh 					
	Virtual Services					
	 Discovery and Load B 	Ę				
	Services					
	Ingresses					
	 Configuration 					

- d. On the Basic Information tab page, set the required parameters.
 - Name: Set the application name.
 - Version: Set the application version.
 - Clusters: Select the cluster where you want to deploy the application.
 - Namespaces: Select the namespace where you want to deploy the application.
 You must select the namespace created in step 1.
 - Replicas: Set the number of replicas, that is, the number of pods.

eate Application	1			
Basic	Information	Container	Service Configuration	Done
Name:	nginx			
	The name should be 1-64 charact	ters long, and can contain numbers, lower ca	se English letters and hyphens, but cannot start with a hyph	en.
Version:	v1			
	The version begins with a letter a	nd can contain numbers, English letters, or "	-", less than 64 characters in length.	
Clusters	test-01	Ŧ		
Namespaces	demo	Ŧ		
Replicas:	1			
				Back Next

e. Configure the container. Set Image Name to nginx, and set Image Version to 1.14. For more information about other container settings, see *Create a deployment application by using an image*. Then, click Next.

Creat	Create Application							
Cont	Basic Informa ainer1 • Add Cont		ner	Service Configuration	Dor	e		
	Image Name:	nginx	Select image		Overview			
	Image Version:	1.14	Select image version		Environment Health Check			
le		Always pull image Image pull secret			Life cycle Volume			
General	Resource Limit:	CPU eg : 500m Core Memory eg	: 128Mi MiB		Log Configuration Prev	Next		
	Resource Request:	CPU 0.25 Core Memory 512	2 MiB OPle	ase set according to actual usage				
	Init Container							

- f. Set the Service parameters.
 - A. In the upper-left corner, click Create. Then, in the displayed dialog box, set the required parameters.

Cr	eate Application				
	Basic Inform	ation	Container	Service Configuration	Done
Control	Services(Service)	Create			
Access Control		Name	service port	Container Port	Protocol
					Prev Create

• Name: Enter the service name.

- Type: Select the ClusterIP service type.
- Port Mapping: Set the service port name, service port number, container port number, and TCP protocol. The format of a service port name must be <protocol>[-<suffix>-]. The protocol part of a service port name can be one of the following: gRPC, HTTP, HTTP/2, HTTPS, Mongo, Redis, TCP, TLS, or UDP. Istio supports a routing service through one of these protocols.



The protocol part of a service port name is case insensitive.

- Annotation: Add an annotation to the service.
- Tag: Add a tag to the service to identify the service.

Create Service		\times
Name: Type:	nginx ClusterIp • Headless Service	
Port Mapping:	 Add Name service port Port http 8080 80 TCP ▼ ● 	
Annotation:	• Add	
Tag:	• Add	
	Create Ca	ncel

B. Click Create.

	Basic Informat	ion	Container	Service Configuration	Done
Services(Service)	Services(Service)	Update Delete			
		Name	service port	Container Port	Protocol
Access Control		http	8080	80	ТСР

g. In the lower-right corner, click Create. Then, the system displays all the created resources if no error occurs. You can click View Details to view the application details.

Create Application				
Basic Information	Container	> Ser	vice Configuration	one
	Create			
		Success		
	Create deployment	nginx	Succeeded	
	Create service	nginx	Succeeded	
	CreateDestinationRule	nginx	Succeeded	
	CreateVirtualService	nginx	Succeeded	
	View Details	Create again		

- 3. Create a client-end application named sleep. This application will be used to send access traffic to the created virtual service.
 - a. Save the following as a file named sleep . yaml .

```
apiVersion : v1
kind : Service
metadata :
  name : sleep
  labels :
    app : sleep
spec :
 ports :
 - port :
            80
    name :
            http
  selector :
    app : sleep
apiVersion : extensions / v1beta1
kind : Deployment
metadata :
  name : sleep
spec :
  replicas : 1
  template :
    metadata :
       labels :
         app : sleep
    spec :
      containers :
        name : sleep
image : pstauffer / curl
command : ["/ bin / sleep ", " 3650d "]
```

imagePullP olicy : IfNotPrese nt

- b. In the demo namespace, run the kubectl apply f sleep . yaml
 - n demo command.
- c. Run the kubectl exec it n demo < podName > bash

command to log on to the pod where the sleep application runs, and then run the following commands to call the nginx service:

```
for i in `seq 1000`
    do
    curl - I http://nginx.demo:8080;
    echo '';
    sleep 1;
    done;
```

If the service is called, the content shown in the following figure is displayed.



d. Access the Kiali console to verify that the virtual service is called.



Before you perform this step, you must enable the Kiali visualized service mesh when deploying Istio for the target Kubernetes cluster.

- A. Run the kubectl port forward svc / kiali n istio
 - system 20001 : 20001 command to open port 20001 in your local host.
- B. Enter http://localhost: 20001 in your browser.



- 4. Create a version of the virtual service for a Canary release.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane under Container Service-Kubernetes, choose Service Mesh > Virtual Service.
 - c. On the right of the target virtual service, click Manage.

Container Service - Kubernetes +	Virtual Service			Refresh Create
Overview	Clusters test-01	 Namespaces 	demo 🔻	Search By Name Q
 Clusters 	Name	Workload	Time Created	Action
 Applications 	nginx	v1 -> nginx	04/26/2019,18:29:32	Delete Manage
 Service Mesh 				
Virtual Services				
Discovery and Load B				

d. Click Add new canary version.

nginx 🔁 Back to Lis	t				Refresh
Overview					
Workload Type:	Deployment				
Time Created:	04/26/2019,18:2	29:32			
Version Management	Add new canary version	dit the canary deployme	nt policy Fault Injection Policy		
Version Workload	Number of instances	Canary Deployment	Time Created		Action
v1 nginx	1		04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic T	raffic Control

e. Set the Version and Replicas, then click Next.

Create Applicati	ion					
Bas	sic Information	Container	>	Canary Deployment	>	Done
Version:	v2					
	The version begins with a letter and	can contain numbers, English letters	, or "-", less th	an 64 characters in length.		
Replicas:	1					
						Back Next

f. Configure a container. Set Image Name to nginx, and set Image Version to
 1.15. For more information about other container configurations, see Create a
 deployment application by using an image. Then, click Next.

Crea	Create Application							
Con	Basic Inform tainer1 • Add Cor		Done					
	Image Name:	Inginx Select image	Overview					
	Image Version:	1.15 Select image version Always pull image Image pull secret	General Environment Health Check Life cycle Volume					
General	Resource Limit:	CPU eg : 500m Core Memory eg : 128Mi MiB	Volume Log Configuration Prev Next					
	Resource Request:	CPU 0.25 Core Memory 512 MiB OPlease set according to actual usage						
	Init Container							

g. Set the Type parameter to select a Canary deployment policy. For example, you can select the Weight based Routing radio button, and then set v2Traffic Weight

to 30 %.

Note:

• If you select the Weight based Routing radio button, then you can set the traffic distribution ratio for versions of the service that is to be deployed.

• You only need to set the v2Traffic Weight parameter. The v1Traffic Weight value is shown through automatic calculation.

Basic Inforn	nation	Container	Canary Deployment	Done
Туре	Weight based Routing	ng \odot Request Content based Routing		
Traffic Weight	v2Traffic Weight 30	% v1Traffic Weight 70 %		

h. Click Create. After a creation success message is displayed by the system, choose Service Mesh > Virtual Services.

Container Service - Kubernetes -	Virtual Service			Refresh	Create
Overview	Clusters test-01	v Namespaces demo	¥	Search By Name	٩
Clusters	Name	Workload	Time Created		Action
 Applications 	nginx	v1 -> nginx v2 -> nginx-v2	04/26/2019,18:29:32	Delete	Manage
 Service Mesh 					
Virtual Services					
Discovery and Load B					

i. On the right of the target service, click Manage.

Container Service - Kubernetes +	nginx	€ Back to Li	st				Refresh
Overview							
Clusters	Overv	iew					
 Applications 	Workl	oad Type:		Deployment			
	Time	Created:		04/26/2019,1	18:29:32		
 Service Mesh 	Version	Management	Add new	canary version	Edit the canary deploy	ment policy Fault Injection Polic	cy
Virtual Services							
 Discovery and Load B 	Version	Workload	Number	of instances	Canary Deployment	Time Created	Action
Services	v1	nginx	1		Traffic Weight 70%	04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic Traffic Control
Services	v2	nginx-v2	1		Traffic Weight 30%	04/28/2019,10:25:37	Manage Workload Delete Take over all the traffic Traffic Control
Ingresses							9
 Configuration 							

j. Run the kubectl exec - it - n demo < podName > bash command to log on to the pod where the sleep application runs, and then run the following commands to call the nginx service:

```
for i in `seq 1000`
    do
    curl - I http://nginx.demo:8080;
    echo '';
    sleep 1;
```

done ;

If the service is called, the content shown in the following figure is displayed.

HTTP/1.1 200 OK content-type: text/html; charset=utf-8 content-length: 5719 server: istio-envoy date: Tue, 02 Apr 2019 08:21:13 GMT x-envoy-upstream-service-time: 24

k. Access the Kiali console to verify that the access traffic of the virtual service are distributed to the two versions according to the distribution ratio that you set.

Note:
Before you perform this step, you must enable the Kiali visualized service mesh
when deploying Istio for the target Kubernetes cluster.
A. Run the kubectl port - forward svc / kiali - n istio -
system 20001 : 20001 command to open port 20001 in your local host.
B. Enter http://localhost: 20001 in your browser.



- 5. Edit the Canary release policy.
 - a. On the page that displays the details of the target virtual service, click Edit the canary deployment policy.

nginx	★ Back to Lis	t				Refresh
Overvie	ew					
Workloa	ad Type:		Deployment			
Time Cr	reated:		04/26/2019,1	3:29:32		
Version	Management	Add new	canary version	Edit the canary deploy	ment policy Fault Injection Policy	
Version	Workload	Number	of instances	Canary Deployment	Time Created	Action
v1	nginx	1		Traffic Weight 70%	04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic Traffic Control
v2	nginx-v2	1		Traffic Weight 30%	04/28/2019,10:25:37	Manage Workload Delete Take over all the traffic Traffic Control

b. Change the traffic distribution ratio for the two versions of the target virtual services. For example, set Version v1 to 20 %, and set Version v2 to 80 %.
 Then, click OK.

Traffic Weight			×
Version v1 Version v2	20	- route: - destination: host: nginx subset: v1 weight: 20 - destination: host: nginx subset: v2 weight: 80	
			OK Cancel

c. In the upper-right corner, click Refresh.

nginx	€ Back to List	t						Refresh	
Over	view								
Work	load Type:		Deployment						
Time	Created:		04/26/2019,	18:2	9:32				
Versio	n Management	Add new	canary version	0	Edit the canary deployr	nent policy	Fault Injection Policy		
Version	Workload	Number	of instances		Canary Deployment	Time Crea	ated	Actio	n
v1	nginx	1			Traffic Weight 20%	04/26/20	19,18:29:31	Manage Workload Delete Take over all the traffic Traffic Contro	J
v2	nginx-v2	1			Traffic Weight 80%	04/28/20:	19,10:25:37	Manage Workload Delete Take over all the traffic Traffic Contro	J

d. Run the kubectl exec - it - n demo < podName > bash command to log on to the pod where the sleep application runs, and then run the following commands to call the nginx service:

```
for i in `seq 1000`
    do
    curl - I http://nginx.demo:8080;
    echo '';
    sleep 1;
    done;
```

If the service is called, the content shown in the following figure is displayed.



e. In your browser, enter http :// localhost : 20001 to log on to the Kiali console to check how the target service is called.



- 6. Use the latest version of the target service to take over all the traffic that is destined for the earlier version.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane, choose Service Mesh > Virtual Service.
 - c. On the right of the target virtual service, click Manage.
 - d. In the Version Management area, find v2, and click Take over all the traffic in the Action column.

nginx	★ Back to List	t				Refresh
Overview	1					
Workload	Туре:		Deployment			
Time Crea	ited:		04/26/2019,18	3:29:32		
Version Ma	anagement	Add new (canary version	Edit the canary deploy	ment policy Fault Injection Policy	
Version	Workload	Number of	of instances	Canary Deployment	Time Created	Actio
v1	nginx	1		Traffic Weight 20%	04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic Traffic Control
v2	nginx-v2	1		Traffic Weight 80%	04/28/2019,10:25:37	Manage Workload Delete Take over all the traffic Traffic Contro

e. In the displayed dialog box, click Confirm. Then, in the Canary Deployment column of v2, Traffic Weight 100% is displayed.

	nginx	€ Back to Lis	t				Refresh
Number Canary Deployment Time Created Version Workload Number of Instances Canary Deployment Time Created V1 nginx 1 Traffic Weight 0% 04/26/2019,18:29:31 Manage Workload Delete Take over all the traffic Traffic	Overvie	ew					
Version Management Add new canary version Edit the canary deployment policy Fault Injection Policy Version Workload Number of instances Canary Deployment Time Created v1 nginx 1 Traffic Weight 0% 04/26/2019,18:29:31 Manage Workload Delete Take over all the traffic Traffic	Workloa	ad Type:		Deployment			
Version Workload Number of instances Canary Deployment Time Created v1 nginx 1 Traffic Weight 0% 04/26/2019,18:29:31 Manage Workload Delete Take over all the traffic Traffic	Time Cr	eated:		04/26/2019,1	8:29:32		
v1 nginx 1 Traffic Weight 0% 04/26/2019,18:29:31 Manage Workload Delete Take over all the traffic Traffic	Version I	Management	Add new	canary version	Edit the canary deployr	nent policy Fault Injection Policy	
	Version	Workload	Number	of instances	Canary Deployment	Time Created	Actio
v2 nginx-v2 1 Traffic Weight 100% 04/28/2019,10:25:37 Manage Workload Delete Take over all the traffic Traffic	v1	nginx	1		Traffic Weight 0%	04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic Traffic Contro
	v2	nginx-v2	1		Traffic Weight 100%	04/28/2019,10:25:37	Manage Workload Delete Take over all the traffic Traffic Contro

f. Run the kubectl exec - it - n demo < podName > bash command to log on to the pod where the sleep application runs, and then run the following commands to call the nginx service:

```
for i in `seq 1000`
    do
    curl - I http://nginx.demo:8080;
    echo '';
    sleep 1;
```

done ;

If the service is called, the content shown in the following figure is displayed.



g. In your browser, enter http://localhost: 20001 to log on to the Kiali console to check how the target service is called.



- 7. Delete the earlier version of the target service.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane under Container Service-Kubernetes, choose Service Mesh > Virtual Service.
 - c. On the right of the target virtual service, click Manage.
 - d. In the Version Management area, find v1, and click Delete in the Action column.

nginx	€ Back to Lis	t			Refresh
Overvie	ew				
Workloa	ad Type:	Deployment			
Time Cr	reated:	04/26/2019,1	8:29:32		
Version	Management	Add new canary version	Edit the canary deploy	ment policy Fault Injection Policy	
Version	Workload	Number of instances	Canary Deployment	Time Created	Action
v1	nginx	1	Traffic Weight 0%	04/26/2019,18:29:31	Manage Workload Delete Take over all the traffic Traffic Control
v2	nginx-v2	1	Traffic Weight 100%	04/28/2019,10:25:37	Manage Workload Delete Take over all the traffic Traffic Control

Then, only the latest version of the target service operates.

nginx	€ Back to Lis	t				Refresh
Overvie	ew.					
Workloa	d Type:		Deployment			
Time Cr	eated:		04/28/2019,1	17:07:16		
Version I	Management	Add new	canary version	Edit the canary deployment	t policy Fault Injection Policy	
Version	Workload	Number	of instances	Canary Deployment	Time Created	Act
v2	nginx	1		Traffic Weight 100%	04/28/2019,17:07:16	Manage Workload Delete Take over all the traffic Traffic Cont

Note:

If you want to release one more version of the target service by using the Canary deployment method, you can repeat the preceding steps.

6 Monitoring

6.1 Use ARMS to monitor an application running in a Kubernetes cluster

This topic describes how to use Application Real-Time Monitoring Service (ARMS) to monitor an application running in Alibaba Cloud Container Service for Kubernetes.

Overview about ARMS

ARMS is a Java application performance management (APM) monitoring product developed

by Alibaba Cloud. If you use ARMS to monitor a Java application, you only need to mount a probe in the application startup script without modifying any code. By comprehensively monitoring the application, the probe helps you quickly locate faulty and slow interfaces, reproduce parameter calling, detect memory leaks, and discover system bottlenecks, more efficiently diagnosing problems online. For more information, see *ARMS*.

ARMS monitors an application as follows:

- · Automatically discovers the application topology
- · Automatically discovers and monitors interfaces
- · Captures exception transactions and slow transactions, and provides SQL analysis
- · Provides a Java exception report
- · Provides trace-based transaction snapshot queries

周用方法	行号	扩展信息	时间轴(单位:毫秒)	
Tomcat Servlet Process				605
▼ StandardHostValve.invoke(org.apache.catalina.connector.Request request, org.apache.c	110		0	
FrameworkServlet.doGet(javax.servlet.http.HttpServletRequest request, javax.servlet	858	异常:		
 ZipkinBraveController.zxMqTwo() 	201		vork.web.util.NestedServletException ng failed; nested exception is	
ProducerImpl.send(com.aliyun.openservices.ons.api.Message message)	105		eException: Tue Aug 14 12:37:31	0
FrameworkServlet.doGet(javax.servlet.http.HttpServletRequest request, javax.servlet	858	CST 2018 Send n	nq message failed. Topic is:zxMqTwo	0

• Provides multidimensional ad-hoc troubleshooting, including multidimensional trace searches and exception trace searches
· Integrates PaaS platforms

Prerequisites

- A Kubernetes cluster is created. For more information, see *Create a Kubernetes cluster*.
- · ARMS is activated. For more information, see Activate ARMS.

Install ARMS components

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane, choose Store > App Catalog. On the App Catalog page, select ack-arms-pilot.
- 3. On the App Catalog ack-arms-pilot page, click DEPLOY.

In the left-side navigation pane, choose Applications > Deployments, and then select the target cluster and namespace. An application named *ack* - *arms* - *pilot* -

default - ack - arms - pilot is displayed on the page.

Container Service - Kubernetes +	-	Deployment				Refresh	Create by Image	Create by Template
Overview	^	$\ensuremath{\mathscr{S}}$ How to use private images		a pod to the specif	fied node 🔗 Create a Layer-4 Ingress	& Create a Layer-7 Ingress & Confi	gure pod auto scaling	
 Clusters Clusters 	l	S Container monitoring S B Clusters k8s-test	v Namespaces default	v			Search By Name	٩
Nodes	l	Name	Tag	PodsQuantity	Image	Time Created		Action
Volumes Namespaces	IJ	ack-arms-pilot-default-ack- arms-pilot	app:ack-arms-pilot-default-ack- arms-pilot chart:ack-arms-pilot-0.1.1 release:ack-arms-pilot-default heritage:Tiller	1/1			05:40 Details	Edit Scale Monitor More→
Authorization								
Deployments								
Stateful Sets								

Grant permission to use ARMS

1. In the left-side navigation pane under Container Service-Kubernetes, choose Clusters > Clusters.



To perform the operations required in this section, you must use the primary account to log on to the Container Service console.

2. Click the target cluster name to view the cluster details.

3. In the Cluster Resources area, click Worker RAM Role.

Cluster Resource	
ROS	And a second
Internet SLB	Contraction of Contraction (Contraction)
VPC	the freedom of the first first first
NAT Gateway	ter french fall fill and
Master RAM Role	Management and the second second second second second
Worker RAM Role	relieve device a factor of the second state of the second state of the



Note:

This topic uses the latest version of the RAM console.

If you use an earlier version of the RAM console, you can modify the target policy document by using one of the following methods:

Method 1

a. In the left-side navigation pane, click Roles, and then enter the Worker RAM Role name in the Role Name box. Click the target Role Name.

Role Management		Create Role Create Role
Role Name 🔻	Search	
Role Name	Created At	Actions
Terrorent Conten-Conten-	2018-12-28 15:46:49	Manage Authorize Delete

b. In the Basic Information area, click Edit Basic Information in the upper-right corner.

Basic Information	Edit Basic Information
Role Name Ku	Description Grant ecs with kubernetes worker role.
Created At 2018-12-28 15:46:49	

Method 2

In the lower-right corner of the RAM dashboard page, click New Version to switch to the latest version of the RAM console. In the Container Service console, click Worker RAM Role to log on to the RAM console.

RAM		RAM Overview						
Dashboard Users Groups		Welcome to Resource A	Access Mana	igement (I	RAM)			
Policies Roles		RAM User Logon Link: http://signin.aliyun.com/containerdoc/login.htm						
Settings		User Overview	Group Overvie	W		Permission Policy Overview	Role Overview	
ActionTrail	Ę	You have 6 users	You have 0 gr	You have 0 groups		You have 15 custom authorization policies	You have 28 roles	
		Operations Guide						
		 Manage authorization policies. Create and authorize groups. Create users and add them to groups. Authorization complete. 		RAM Dashboard Users Groups Policies Roles	Sy 1.Choose	Icy Management Custom Pulay Custom Pulay	No. of times references	

- 4. On the RAM Roles page, click the policy name on the Permissions tab page.
- 5. On the Policies page, click Modify Policy Document on the Policy Document tab

page.

RAM	RAM / Policies / k8	sWorkerRolePolicy-		
Overview	← k8sWor	kerRolePolicy-	75 u896-4625-8500-0745-8576/bak	
Identities ^	Basic Information			
Groups	Policy Name	k8sWorkerRolePolicy	Version Number v2	
Users	Policy Type	Custom Policy	Note	
Settings				
Permissions ^	Policy Document	t Versions References		
Grants	Modify Policy Docu	iment		
Policies	« 1 (1	
RAM Roles		'Version": "1", 'Statement": [
OAuth Applications	4	{ "Action": [
	6	"ecs:AttachDisk",		
	7	"ecs:DetachDisk",		16
	8	"ecs:DescribeDisks",		
	9	"ecs:CreateDisk",		
	10	"ecs:CreateSnapshot",		
	11	"ecs:DeleteDisk",		
	12	"ecs:CreateNetworkInterfa	ace",	
	13	"ecs:DescribeNetworkInter		
	14	"ecs:AttachNetworkInterfa		
	15	"ecs:DetachNetworkInterfa		
	16	"ecs:DeleteNetworkInterfa	ace".	

6. In the Policy Document area, add the following fields and then click OK.

```
{
    " Action ": " arms :*",
    " Resource ": "*",
    " Effect ": " Allow "
```

}



Deploy ARMS monitoring for an application

In a YAML file used to create the target deployment, add the following annotation

s to deploy ARMS application monitoring.

```
annotation s :
    armsPilotA utoEnable : " on "
```

armsPilotC reateAppNa me : "< your - deployment - name >"

- Note:
- In the YAML file, you must add annotation s under metadata of template in the spec field.
- The value of armsPilotC reateAppNa me is the name of the application monitored by ARMS.
- 1. In the left-side navigation pane of the Container Service console, choose Applications > Deployments.
- 2. In the upper-right corner, click Create by Template.
- 3. Select the target cluster and namespace to create a deployment.

Clusters	k8s-cluster 🔻	
Namespace	default	
Resource Type	Custom	
Template	<pre>1 apiVersion: apps/vlbetal # for versions before 1.8.0 use apps/vlbetal 2 kind: Deployment 3 metadata: 4 name: arms-springboot-demo 5 labels: 6 app: arms-springboot-demo 7 spec: 7 replicas: 2 9 selector: 10 metadata: 11 app: arms-springboot-demo 12 template: 13 metadata: 14 annotations: 15 armsPilotCreateAppName: "arms-k8s-demo" 16 armsPilotCreateAppName: "arms-k8s-demo" 17 labels: 18 app: arms-springboot-demo 19 spec: 20 containers: 21 - resources: 22 limits: 23 cpu: 0.5 24 imagePullPolicy: Always 25 name: arms-springboot-demo 27 env: 28 - name: MYSQL_SERVICE_HOST 29 value: "arms-decker_repo/arms-springboot-demo 27 value: "arms-decker_report 31 value: "3306" </pre>	Add Deployment Deploy with exist template
	Save Template DEPLOY	

```
apiVersion : apps / vlbeta1 # for versions before 1.8.
0 use apps / vlbeta1
kind : Deployment
metadata :
    name : arms - springboot - demo
    labels :
        app : arms - springboot - demo
spec :
        replicas : 2
        selector :
        matchLabel s :
        app : arms - springboot - demo
template :
```

```
metadata :
      annotation s:
        armsPilotA utoEnable : " on "
        armsPilotC reateAppNa me : " arms - k8s - demo "
      labels :
        app : arms - springboot - demo
    spec :
      containers :
          resources :
       -
            limits :
              cpu: 0.5
image : registry . cn - hangzhou . aliyuncs . com / arms
- docker - repo / arms - springboot - demo : v0 . 1
          imagePullP olicy : Always
          name : arms - springboot - demo
          env :
             name : MYSQL_SERV ICE_HOST
              value : " arms - demo - mysql "
              name : MYSQL_SERV ICE_PORT
              value : " 3306 "
apiVersion : apps / v1beta1 # for versions
                                                 before
                                                         1.8.
0 use apps / v1beta1
kind : Deployment
metadata :
  name : arms - demo - mysql
labels :
    app : mysql
spec :
  replicas : 1
  selector :
    matchLabel s :
      app : mysql
  template :
    metadata :
      labels :
        app : mysql
    spec :
      containers :
          resources :
            limits :
              cpu: 0.5
          image : registry . cn - hangzhou . aliyuncs . com / arms
- docker - repo / arms - demo - mysql : v0 . 1
          name : mysql
          ports :
             containerP ort: 3306
              name : mysql
apiVersion : v1
kind : Service
metadata :
  labels :
    name : mysql
  name : arms - demo - mysql
spec :
  ports :
           port
                  that this
                                 service
                                           should
   # the
                                                  serve
                                                            on
   - name : arms - mysql - svc
      port : 3306
      targetPort : 3306
   label
          keys and values
                                 that must
                                                match
                                                        in
 #
                                                             order
      receive traffic for
                                 this
                                        service
  to
  selector :
```

app : mysql

Verify the results

- 1. In the left-side navigation pane of the Container Service console, choose Applications > Deployments, and then select the target cluster and namespace to view the created deployment.
- 2. In the Action column of the target deployment, click ARMS console to log on to the ARMS console to view application details such as Application Overview, Interface Invocation, and other information.



If ARMS console is not displayed in the Action column, check whether you have granted Container Service permission to use ARMS. For more information, see *Grant permission to use ARMS*.

Deployment				Refresh Crea	te by Image Create by	Template		
Help: & How to use private images & Create applications & Schedule a pod to the specified node & Create a Layer-4 Ingress & Create a Layer-7 Ingress & Configure pod auto scaling & Container monitoring & Blue-green release								
Clusters k8s-test v Na	amespace default 🔻				Search By Name	Q		
Name	Tag	PodsQuantity	Image	Time Created		Action		
ack-arms-pilot-default-ack-arms- pilot	app:ack-arms-pilot-default-ack- arms-pilot chart:ack-arms-pilot-0.1.1 release:ack-arms-pilot-default heritage:Tiller	1/1	registry.cn-hangzhou.aliyuncs.com/arms-docker-repo/arms- pilot:v1.26	01/29/2019,13:42:46	Details Edit Scale	e Monitor More↓		
nginx-deployment-basic	app:nginx	2/2	nginx:1.7.9	01/29/2019,14:20:19	Details Edit Scale	<u> </u>		

7 Operation and maintenance

7.1 Check a Kubernetes cluster to troubleshoot exceptions

This topic describes how to quickly check a Kubernetes cluster in the Container Service console to troubleshoot exceptions.

Prerequisites

• A Kubernetes cluster is created. For more information, see Create a Kubernetes

cluster.

• The Kubernetes cluster is in the running status.



On the Cluster List page, you can check whether the Kubernetes cluster status is in the Running status.

Procedure

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Container Service-Kubernetes, choose Clusters > Clusters. Find the target Kubernetes cluster, and choose More > Cluster Check in the action column.

Container Service - Kubernetes 🔻		Cluster List You	u can creat	e up to 100 clus	sters and add up t	o 1000 nodes in ea	ich cluster. To r	request a c	uota increase, submit a	ticket.	Refresh Create Kubernetes C	Cluster
Overview		Help: & Create cluster & Create C creation failures & Authorization ma	GPU cluster: nagement	s 🔗 Scale clu	ster 🔗 Connect	to Kubernetes clus	ster via kubectl	🔗 Mana	ge applications with con	nmano	ls 🔗 Cluster planning 🔗 Troubleshoo	ot cluster
Clusters	l	Name 🔻		Tags								
Nodes	l	Cluster Name/ID	Tags	Cluster Type	Region (All) 👻	Network Type	Cluster Status	Number of Nodes	Time Created	Ver	sion	Action
Volumes Namespaces	l	k8s-test	۲	Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp118b4d5g1	Running	5	04/28/2019,14:02:54		un.1	Logs hboard More -
Authorization • Applications		k8s-windows-cluster	۲	Windows Kubernetes	China East 1 (Hangzhou)	VPC vpc- bp1fmdpc9b	Running	3	04/20/2019,15:46:00	1.1 aliy	Delete Add Existing Instance Upgrade Cluster	2
Deployments Stateful Sets	ŀ										Automatic Scaling Addon Upgrade	
Daemon Sets											Deploy Istio	Contad
Jobs											Cluster Check 3 Cluster Audit	Š
Cron Jobs											Use Cloud Shell	
Pods											Collect Kubernetes Diagnostics Informa	ation
Volume Claims												

3. In the left-side navigation pane, choose Inspect > Inspection.

Ê	Note:	
	note:	

You can also click Inspection in the toolbox area.

<	Overview						
Overview Inspect Inspection	Cluster Details Type: 1.12.6-aliyun.1 (Dedicated) Upgrade Running ⑦ Cluster: k8-lest Details Type: 1.12.6-aliyun.1 (Dedicated) Upgrade Running ⑦ China (Hangzhou) (Container Service) Create Time: Apr 28, 2019, 14:02:54 Last check:Unknown						
	Inspection No data This feature can diagnose and provide recommendations for your container cluster. It is recommended that you check and maintain the cluster on a regular basis. Start						
	Toolbox Inspection Quickly locate container cl Quickly locate node issues Quickly locate node issues						

- · Cluster: displays the cluster name, type, status, and the result of the latest check
- Inspection: displays the corresponding time and results for the latest five cluster checks.



•

If you check a Kubernetes cluster for the first time, this area does not display any data except for the Start button. Clicking this button opens the the Inspection page where you can then enable cluster checking.

- Toolbox: displays the cluster check module.
- 4. Click Start. Then, in the displayed page, select the Warning check box, and click Start.



Issue: 20190613

After you enable the cluster checks, the page displays the progress for the cluster checks in real time.

<	Inspection	Inspection	\times
Overview Inspect	Stort 1	• Cluster k8s-test	
Inspection	No data	* Select Checklist	
	This feature can diagnose and provide recommendations for your container cluster. It is recommended that you check and maintain t	Cluster Resources Check whether the cluster resources affect the normal operation of the cluster	
		Cluster Configurations Check if the cluster configuration affects the normal operation of the cluster	
		• Warning 2	
		Some check items will deploy container in your	
		container cluster and collect the check results, please confirm whether it is allowed	
			Contact Us
		3	
		Start Cancel	

5. Click Details to view the results for the cluster resource and configuration checks.

<	Inspection Report	
Inspection Report	The inspection feature is in the beta test. The inspection result is for reference only	
	Start Time May 7, 2019, 14:11:13 Status Completed Result N	rmal
	Cluster Resources Completed	
	Cluster Resources Result	
	LoadBalancer	
	apiserver (Internet)	
	apiserver (Intranet) Normal (1)	
	SSH (Internet)	
	Network	
	VPC Normal (1)	
	VSwitch	

- 6. Perform the action that corresponds to the result returned from the check.
 - Normal: No action is required.
 - Warning: Confirm the severity of the issue. If the issue may result in cluster exceptions, perform the appropriate actions to troubleshoot the issue.
 - · Exception: Perform the required actions to troubleshoot the issue immediately.

<	Inspection Report		Result Instance
Inspection Report	The inspection feature is in the beta te	est. The inspection result is for reference only	✓ Instance Exists
	Start Time May 7, 2019, 14:11:13	Status Completed Re	✓ Instance Status
	Cluster Resources	Cluster Configurations	✓ Instance Listener Configuration
	Completed	Completed	✓ Instance Backends
	Cluster Resources Result		
	All Pending		
	LoadBalancer		
	apiserver (Internet)	Normal (1)	
	apiserver (Intranet)	Normal (1)	
	SSH (Internet)	Normal (1)	Ē



- For each resource or configuration (which contains result and instance information) in the result of a check, you can click the corresponding box icon on the right to view the instance and result information.
- If a resource or configuration is not in the normal state, the result displays the checked items, severity of exceptions, and recommended solutions. You can perform actions to troubleshoot as needed.

8 Serverless

8.1 Use a GPU container instance

This topic describes how to use a GPU container instance by using an example in which Tensorflow is used to identify a picture. ACK serverless Kubernetes provides GPU container instances by integrating with Elastic Container Instance (ECI). As a part of this service, GPU container instances can be used in a serverless Kubernetes cluster or on a virtual node of a dedicated or managed Kubernetes cluster.

Billing

GPU and CPU resources provided by ACK serverless Kubernetes are charged with the Pay-As-You-Go billing method.

GPU instances provided by ACK serverless Kubernetes are charged the same as the ECS GPU instances. You are not charged additional fees.

Overview

In an ACK serverless Kubernetes cluster, you can easily create a pod to which a GPU is attached. Specifically, for the pod configurations, you must set the annotation s parameter to specify the GPU type that you want, and set the limits parameter to specify the number of GPU.

A GPU is exclusive to one pod.

Prerequisites

Either a serverless Kubernetes cluster is created, or a virtual node is created for a Kubernetes cluster. For more information, see *Create a serverless Kubernetes cluster* or *Deploy a virtual node*.

Procedure

In the following steps, Tensorflow is used in a serverless Kubernetes cluster to identify the following picture.



- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Deployments. Then, in the upper-right corner, click Create by Template.
- 3. Select the target cluster and namespace, select an example template or customize a template, and then click DEPLOY.

You can use the following YAML template to create a pod:

Note:

In this pod template, the GPU type is set to P4 , and the number of GPU is set to

1.

```
apiVersion :
             v1
kind : Pod
metadata :
  name : tensorflow
  annotation s :
    k8s . aliyun . com / eci - gpu - type : " P4 "
spec :
 containers :
   image : registry - vpc . cn - hangzhou . aliyuncs . com / ack
- serverless / tensorflow
    name : tensorflow
   command :
   - " sh "
   - "- c "
   - " python
               models / tutorials / image / imagenet / classify_i
mage . py "
    resources :
      limits :
        nvidia . com / gpu : " 1 "
  restartPol icy : OnFailure
```

4. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Pods to verify that the pod is created.

Note:

It may take a few minutes to create a pod. If you the pod is not displayed, you can refresh the page.

Container Service - Kubernetes -		Pod												Refresh
Authorization	*	Clusters k8s-cl	luster-serverless	s ▼ Na	mespaces de	fault 🔻						Search By Na	me	Q
 Applications 		To enh	nance data secu	rity, RAM u	sers are not allo	wed to access	clusters without auth	norization.						
Deployments		If you o	cannot access o	ertain reso	urces, contact A	ibaba Cloud a	ccount owner to obta	in authorization. F	or more information, see	Documentatio	ons.			
Stateful Sets		Name 🔶	Status 🗢	max atter	npts Pod IP 🐗	Node 🗢			Time Created 🗢	CPU	Memory			
Daemon Sets	l	tensorflow	Running	0		viking-c		10,000	05/17/2019,18:30:11	0		3.184 Mi	Details	More +
Jobs											Total: 1 item(s), Per Pa	age: 25 item(s)		
Cron Jobs	Œ										Total 2 total by For H	sgor contoni(o)		
Pods														
Volume Claims														

5. Click the target pod, and click the Logs tab to verify that the picture is identified according to the logs marked in the red frame.

	_							
Container Service - Kubernetes +		Ready	True	05/28/20	19,11:24:36	-	-	
Overview	-	Initialized	True	05/28/20	19,11:24:36	-	-	
 Clusters 		PodScheduled	True	05/28/20	19,11:24:36		-	
Clusters		Container Events Created by	Init Containers	Volumes Logs				
Nodes								_
Volumes		Container : tensorflow		Hum of Endo I	100 -		Auto Refresh Re	efresh Download
Namespaces			flow/compiler/xla/servic	e/service.cc:150] XLA serv	rice 0x54e78c0 executing com	putations on platform Host. Devices:		
Authorization		2019-05-15 10:37:26.012067: I tenso 2019-05-15 10:37:26.012393: I tenso						
 Applications 		name: Tesla P4 major: 6 minor: 1 me	aoryClockRate(GHz): 1.113					
Deployments	Ξ	pciBusID: 0000:00:08.0 totalMemory: 7.43GiB freeMemory: 7.3						
Stateful Sets		2019-05-15 10:37:26.012427: I tenso	flow/core/common_runtime					
Daemon Sets		2019-05-15 10:37:26.014409: I tenso 2019-05-15 10:37:26.014441: I tenso			vice interconnect StreamExe O	cutor with strength 1 edge matrix:		
Jobs		2019-05-15 10:37:26.014455: I tenso						
		2019-05-15 10:37:26.014670: I tenso name: Tesla P4, pci bus id: 0000:00			reated TensorFlow device (/	job:localhost/replica:0/task:0/device:GF	'U:O with 7116 MB memory) -> p	whysical GPU (device: 0,
Cron Jobs		2019-05-15 10:37:27.092482: I tenso	flow/stream_executor/dso	loader.cc:152] successful	ly opened CUDA library libr			
Pods		tgz 100.0%>>> Downloading inception-: Successfully downloaded inception-20			12-05.tgz 100.0W>> Download	ling inception-2015-12-05.tgz 100.0%>> Do	wnloading inception-2015-12-0	05.tgz 100.0%
Volume Claims		giant panda, panda, panda bear, com						
Releases		indri, indris, Indri indri, Indri b						
Workflow		lesser panda, red panda, panda, bea custard apple (score = 0.00147)	cat, cat bear, Ailurus	ulgens (score = 0.00296)				
 Service Mesh 		earthstar (score = 0.00117)						
	-							

If you want to use a GPU container instance on a virtual node of a dedicated or managed Kubernetes cluster, follow these steps:

```
Note:
```

A GPU container instance used on a virtual node supports a larger variety of deep learning platforms such as Kubeflow, Arena, or other customized CRD.

1. Create a pod on the virtual node or in a namespace to which the virtual -

```
node - affinity - injection = enabled label tag is added. For more
information, see Deploy a virtual node.
```

2. Replace the template used in step 3 with the following template:

```
apiVersion : v1
kind : Pod
metadata :
```

```
name : tensorflow
  annotation s:
    k8s . aliyun . com / eci - gpu - type : " P4 "
spec :
  containers :
    image : registry - vpc . cn - hangzhou . aliyuncs . com / ack
- serverless / tensorflow
    name : tensorflow
    command :
   - " sh "
- "- c "
   - " python
                 models / tutorials / image / imagenet / classify_i
mage . py "
    resources :
      limits :
        nvidia . com / gpu : " 1 "
  restartPol icy : OnFailure
nodeName : virtual - kubelet
```

8.2 Associate an EIP address with a pod

This topic describes how to associate an EIP address with a pod in a serverless Kubernetes cluster or with a pod on a virtual node of a dedicated or managed Kubernetes cluster.

Benefits

Associating an EIP address with a target pod offers the following benefits:

- You can easily deploy an application that uses serverless containers.
- Services provided by the application can be more quickly accessed.
- Your pod can access the Internet without the need of a VPC NAT gateway.
- Your target application can be accessed through the Internet without the need of an SLB instance.

Prerequisites

- Either a serverless Kubernetes cluster is created, or a virtual node is created in a dedicated or managed Kubernetes cluster. For more information, see *Create a serverless Kubernetes cluster* or *Deploy a virtual node*.
- The corresponding port for the target application that you want to run is enabled in the security group of the target Kubernetes cluster.

Note:

The Nginx application is used as an example. Therefore, port 80 must be enabled in the security group of the target Kubernetes cluster.

Procedure

1. Log on to the VPC console to create an EIP address. For more information, see *Create an EIP*.



You must create an EIP address in the same region where the target Kubernetes cluster is located.

VPC	Elas	tic IP Add	resses												Product Up	pdates
VPCs	🕑 Or	One stop network solution to quickly connect with Alibaba Cloud. To order Smart Access Gateway, click here .														
Route Tables	Crea	ate EIP	Combined Purc	hase	Request Spe	cific EIP	Refresh	Expo	ort Re	new EIP	Custom					
VSwitches											Elastic IP Addr	ess 🗸	Enter a	ID		Q
Internet Shared Band																AF
Shared Data Transfer		Instance ID/Name	IP Address	Monitor	Bandwidth	Connection Type	Charge Type(A	II)	Status(All)	Shared Bandwidth/ Acceleration	Bind Instance	Instanc Type(A	ce NI) 77	Resource Group	Actions	
▼ Elastic IP Addresses										Add to						
Elastic IP Addresses		6	7		1 Mbps Pay By Traffic	BGP	Pay-As-You-Go 05/28/2019, 20:		 Available 	Shared Bandwidth Package	-	2			Bind Unbind	
High Definition Traffic		Pay by tram		ray by Hallic		Created			Add to Global Acceleration					More ∨		

- 2. Log on to the Container Service console.
- 3. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Deployments.
- 4. In the upper-right corner, click Create by Template.

5. Select the target cluster and namespace, select an example template or customize a template, and then click DEPLOY.

Create with Template		
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	managed-cluster v	
Namespaces	kube-system •	
Sample Template	Custom	
Template	<pre>1 aplVersion: v1 2 kind: Pod 3 metadata: 4 name: nginx 5 annotations: 6 "k8s.aliyun.com/eci-eip-instanceid": " 7 spec: 8 containers: 9 - image: registry-vpc.cn-hangzhou.aliyuncs.com/jovi/nginx:alpine 10 inagerulPolicy: Always 11 name: nginx 12 ports: 13 - containerPort: 80 14 name: http 15 protocol: TCP 16 restartPolicy: OnFailure </pre>	Add Deployment Use Existing Template
	Save Template DEPLOY	

You can use the following YAML template to create a pod:



You must replace < youreipIns tanceId > with the EIP ID obtained in step 1.

```
apiVersion : v1
kind : Pod
metadata :
  name : nginx
  annotation s:
" k8s . aliyun . com / eci - eip - instanceid ": "< youreipIns tanceId >"
spec :
  containers :
 - image : registry - vpc . cn - hangzhou . aliyuncs . com / jovi
/ nginx : alpine
    imagePullP olicy : Always
    name : nginx
ports :
      containerP ort :
                         80
      name : http
      protocol : TCP
```

restartPol icy : OnFailure

6. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Pods to view the pod status.

Container Service - Kubernetes +	Pod	Refresh
Overview	Clusters k8s-cluster-serverless v Namespaces default v	Q
Clusters	To enhance data security, RAM users are not allowed to access clusters without authorization. If you cannot access certain resources, contact Alibaba Cloud account owner to obtain authorization. For more information, see Documentations.	
 Applications Deployments 	max Name ⇔ Status ⇔ attempts Pod IP ⇔ Node ⇔ Time Created ⇔ CPU Memory	
Stateful Sets	nginx nginx:alpine 0 0 05/17/2019,18:30:11 0 3.188 Mi	Details More 🗸
Jobs	Total: 1 item(s), Per Page: 25 item(s)	1 > »
Cron Jobs		
Pods		
Volume Claims		5

7. In your browser, enter the URL *http://ip* address to access the welcome page of Nginx.



9 Auto Scaling

9.1 Deploy an Ingress application on a virtual node

This topic describes how to deploy an Ingress application on a virtual node of a Kubernetes cluster. With the virtual node, the Kubernetes cluster can provide the application with greater computing capability without the need for a new node needs to be created for the cluster.

Prerequisites

A virtual node is deployed in your target Kubernetes cluster. For more information, see *Deploy a virtual node*.

The tag virtual - node - affinity - injection : enabled is added to the target namespace. For more information, see Set a namespace tag to create a pod.

Procedure

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Deployments.
- 3. In the upper-right corner, click Create by Template.

4. Select the target cluster and namespace, select an example template or customize a template, and then click DEPLOY.



You can use the following template to customize a YAML template to create an

Ingress application:

```
apiVersion : extensions / v1beta1
kind : Deployment
metadata :
 name : coffee
spec :
  replicas : 2
  selector :
   matchLabel s :
     app : coffee
  template :
    metadata :
      labels :
       app : coffee
    spec :
     containers :
     - name : coffee
       image : nginxdemos / hello : plain - text
       ports :
       - containerP ort : 80
apiVersion : v1
kind : Service
metadata :
  name : coffee - svc
spec :
 ports :
   port: 80
    targetPort : 80
    protocol : TCP
  selector :
    app : coffee
  clusterIP : None
apiVersion : extensions / v1beta1
```

```
kind : Deployment
metadata :
  name : tea
spec :
  replicas : 3
  selector :
   matchLabel s :
     app : tea
  template :
    metadata :
      labels :
       app : tea
    spec :
     containers :
     - name : tea
       image : nginxdemos / hello : plain - text
       ports :
       - containerP ort : 80
apiVersion : v1
kind : Service
metadata :
    name : tea - svc
  labels :
spec :
 ports :
 - port : 80
    targetPort : 80
    protocol : TCP
  selector :
   app : tea
  clusterIP : None
apiVersion : extensions / v1beta1
kind : Ingress
metadata :
  name : cafe - ingress
spec :
  rules :
    host : cafe . example . com
    http :
     paths :
     - path : / tea
        backend :
          serviceNam e : tea - svc
          servicePor t:
                          80
       path : / coffee
        backend :
          serviceNam e : coffee - svc
```

servicePor t: 80

Verify the results

- To verify that the coffee and tea deployments are created, follow these steps:
 - 1. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Deployments.
 - 2. Select the target cluster and namespace.

Container Service - Kubernetes -	[Deploym	ient					Refresh	Create by Image	Create by Templ	late
Overview		🔗 How to	use private images	S Create applications	${\ensuremath{\mathscr{S}}}$ Schedule a pod to the specified node	🔗 Create a Layer-4 Ingress	🖉 Create a Layer-7	Ingress 🔗 Config	jure pod auto scaling		
 Clusters 			ner monitoring 🔗 Bi		vk 🔻				Search By Name		Q
Clusters									Search by Name		~
Nodes	N	Name	Tag	PodsQuantity	Image	Time Created					Action
Volumes		coffee	app:coffee	0/2	nginxdemos/hello:plain-text	05/28/2019,11:36:58		C	etails Edit Sca	le Monitor M	∙lore v
Namespaces	r	nginx	run:nginx	1/1	nginx	05/28/2019,11:24:36		C	etails Edit Sca	le Monitor M	4ore →
Authorization	t	tea	app:tea	0/3	nginxdemos/hello:plain-text	05/28/2019,11:36:58		C	etails Edit Sca	le Monitor M	•lore ↓
 Applications 											
Deployments											

• To verify that all the pods of the Ingress application run on the virtual-kubelet node, follow these steps:

- 1. In the left-side navigation pane under Container Service-Kubernetes, choose Applications > Pods.
- 2. Select the target cluster and namespace.

Container Service - Kubernetes 👻		Pod									Refresh
Overview	-	Clusters managed-cluster	• • • N	lamespaces vk	×				Search By Name		Q
 Clusters 	L	Name 🜲	Status 🜲	max attempts	Pod IP 🌲	Node 🗢	Time Created 🜲	CPU	Memory		
Clusters	L	coffee-56668d6f78-xzbgr	Running	0		virtual-kubelet	05/28/2019,11:36:58	0	0 0	etails	More 🗸
Nodes	L	coffee-56668d6f78-z4inr									
Volumes	L	 hello:plain-text 	Running	0		virtual-kubelet	05/28/2019,11:36:58	0	0 D	Details	More 👻
Namespaces	L	nginx-dbddb74b8-5567q 🗼 nginx	Running	0		virtual-kubelet	05/28/2019,11:24:36	0	0 D)etails	More 🗸
Authorization Applications	U	tea-85f8bf86fd-rrp8k	Running	0		virtual-kubelet	05/28/2019,11:36:58	0	0 C	etails	More 🗸
Deployments Stateful Sets	ł	tea-85f8bf86fd-t5ffl hello:plain-text	Running	0		virtual-kubelet	05/28/2019,11:36:58	0	0 0	Oetails	More 🗸
Daemon Sets		tea-85f8bf86fd-vdq4f	Running	0		virtual-kubelet	05/28/2019,11:36:58	0	0 0	etails	More 🗸
Jobs									Total: 6 item(s), Per Page: 25 item(s)	1	> »
Cron Jobs											
Pods											
Volume Claims											

- · To verify that the target Ingress is created, follow these steps:
 - 1. In the left-side navigation pane under Container Service-Kubernetes, choose Discovery and Load Balancing > Ingresses.
 - 2. Select the target cluster and namespace.

Container Service - Kubernetes +	Ingress		Ingress Overview Refresh Create
Pods	${\mathscr S}$ Ingress log analysis and monitoring ${\mathscr S}$ Blue-green release		
Volume Claims	Clusters managed-cluster v Namespaces vk v		Search By Name C
Releases	Name Endpoint Rule	Time Created	Act
 Service Mesh 	cafe-ingress cafe.example.com/tea -> tea-svc cafe.example.com/coffee -> coffee-svc	05/28/2019,11:36:58	Details Update View YAML Dele
Virtual Services			
 Discovery and Load B 			
Services			
Ingresses			

• To verify that the created Ingress application can be accessed, run the following commands:

```
kubectl - n vk get ing
curl - H " Host : cafe . example . com " < EXTERNAL_I P >/ tea
curl - H " Host : cafe . example . com " < EXTERNAL_I P >/
coffee
```



10 DevOps

10.1 Deploy Jenkins in a Kubernetes cluster and perform a pipeline build

This topic describes how to deploy Jenkins, a continuous integration environment, in an Alibaba Cloud Kubernetes cluster, and how to perform an application pipeline build. The example in this topic details the pipeline build, including how to compile the source code of the application, build and push the application image, and deploy the application.

Prerequisites

You have created a Kubernetes cluster. For more information, see *Create a Kubernetes cluster*.

Deploy Jenkins

- 1. Log on to the Container Service console.
- 2. In the left-side navigation pane, choose Container Service-Kubernetes > Store > App Catalog. Then, click jenkins.



- 3. Click the Values tab.
- 4. Modify the AdminPassw ord field to set a password.



To ensure that your password takes effect, you must remove the pound sign (#)

before AdminPassw ord

If you do not set any password, the system generates a password after Jenkins is deployed. You can run the following command to view the password:

\$ printf \$(kubectl get secret -- namespace ci jenkins ci - jenkins - o jsonpath ="{. data . jenkins - admin - password
}" | base64 -- decode); echo

5. Select the target Cluster and Namespace, enter a Release Name, then click DEPLOY.



Note:

We recommend that you select a custom namespace or the default namespace. In this example, a custome namespace named test is selected.

- 6. In the left-side navigation pane, choose Discovery and Load Balancing > Service.
- 7. Select the cluster and the namespace used for deploying Jenkins.

8. Click the external endpoint of the Jenkins service to log on to Jenkins.

Container Service - Kubernetes ▼	Service							Re	fresh	Create
CronJob	Help: 🔗 Canary release									
Pods	Clusters k8s-cluster	 Namespace 	test 🔻	3				Search By Name		Q
Volumes Claim	Name	Туре	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint				Action
Release	jenkins-demo-jenkins	LoadBalancer	02/18/2019,14:47:08	172.19.8.69	jenkins-demo-jenkins:8080 TCP jenkins-demo-jenkins:30004 TCP		Details	Update Vi	ew YAML	Delete
Discovery and Load B Service 2	jenkins-demo-jenkins-agent	ClusterIP	02/18/2019,14:47:08	172.19.1.20	jenkins-demo-jenkins-agent:50000 TCP	_ 4	Details	Update Vi	ew YAML	Delete
Service 2 Ingress										
 Configuration 										
Config Maps										
Secret										
▼ Store										
Docker Images										

Create a cluster certificate and an image repository certificate, and build and deploy an application

- 1. Create a Kubernetes cluster certificate.
 - a. In the left-side navigation pane, click Manage Jenkins.
 - b. On the Manage Jenkins page, click Configure System.
 - c. In the Cloud area, click Add on the right of Credentials.

Cloud		
Kubernetes		
Name	kubernetes	2
Kubernetes URL	https://kubernetes.default.svc.cluster.local:443	6
Kubernetes server certificate key		
		6
Disable https certificate check	2	
Kubernetes Namespace	test	
Credentials	- none - 🔻 💓 Add 🕶	
	Test Connection	
Jenkins URL	http://jenkins-demo-jenkins:8080	C
Jenkins tunnel	jenkins-demo-jenkins-agent:50000	C
Connection Timeout	0	0
Read Timeout	0	6
Container Cap	10	0
Max connections to Kubernetes API	32	0
Save Apply		

Before adding a credential, you must obtain KubeConfig on the Basic Information tab page of the target Kubernetes cluster.

Connect to Kubernetes cluster via kubect (Use Cloud Shell)						
1. Download the latest kubect client from the Kubernetes Edition page.						
2. Install and set up the kubecti client. For more information, see Installing and Setting Up kubecti						
3. Configure the cluster credentials:						
KubeConfig (Public Access) KubeConfig (Internal Access)						
Copy the following content to your local machine \$HOME/.kube/config						
client-certificate-data:						
Provide and the second s						
The second se						
<pre>client-kay-data:</pre>						

d. In the diaplayed dialog box, set the following parameters:

I I	enkins Credenti	als Provider: Jenkins	
💕 Add	Credentials		
Domain	Global credentials (un	restricted)	•
Kind	Docker Host Certificat	e Authentication	•
	Scope	Global (Jenkins, nodes, items, all child items, etc)	0
	Client Key		
			:
	Client Certificate		
			:
	Server CA Certificate		
			;
	ID		0
	Description		0
Add	Cancel		

- · Kind: Select Docker Host Certificate Authentication.
- Client Key: Paste the copied client-key-data content in KubeConfig.
- Client Certificate: Paste the copied client-certificate-data content in KubeConfig.
- · ID: Enter the certificate ID. In this example, k8sCertAuth is entered.
- Description : Enter description content.
- e. Click Add.
- f. Test connectivity.

Select the added credential in the preceding step from the Credentials dropdown list, and then click Test Connection.

Kubernetes Namespace	test					
Credentials	k8sCertAuth ▼					
	Connection test successful	Test Connection				

g. Set the Kubernetes cluster to dynamically build pods as follows.

Cloud			
Kubernetes			
Name			1_
Name	kubernetes		2
Kubernetes URL	https://kubernetes.default.svc.cluster.local;443		0
Kubernetes server certificate key			1
			0
			2
Disable https certificate check			0
Kubernetes Namespace	test]
Credentials	k8sCertAuth ▼ 🚅 Add ▼		
	Connection test successful	Test Connection	
Jenkins URL	http://jenkins-demo-jenkins:8080		?
Jenkins tunnel	jenkins-demo-jenkins-agent:50000		2
Connection Timeout	0		0
Read Timeout	0		0
Container Cap			1
Container Cap	10		0
Save Apply			

h. Set the Kubernetes pod template.

The slave-pipeline uses four containers to create each corresponding stage of the pipeline.

• Set container jnlp as follows.

	Kubernetes Pod Template					
	Name	slave-pipeline				
	Namespace	ci				
	Labels	slave-pipeline				
	Usage	只允许运行绑定到这台机器的Job		÷		
	The name of the pod template to inherit from					
	Containers					
		Container Template				
		Name	jnlp	0		
		Docker image	registry.cn-beijing.aliyuncs.com/ac	0		
		Always pull image				
		Working directory	/home/jenkins	0		
		Command to run		0		
		Arguments to pass to the command		•		
		Allocate pseudo-TTY	0			
		EnvVars	Add Environment Variable			
			List of environment variables to set in agent pod			
			agent pod			
Save Apply						

Use jenkins-slave-jnlp as a Docker image. The jenkins-slave-jnlp image is used to create the jnlp node to connect the Jenkins master.

```
registry . cn - beijing . aliyuncs . com / acs - sample / jenkins - slave - jnlp : 3 . 14 - 1
```

• Set container kaniko as follows.

Container Template		
Name	kaniko	0
Docker image	registry.cn-beijing.aliyuncs.com/ac	0
Always pull image		
Working directory	/home/jenkins	
Command to run	/bin/sh -c	
Arguments to pass to the command	cat	0
Allocate pseudo-TTY		
EnvVars	Add Environment	
	Variable	
	List of environment variables to set in agent pod	

Use jenkins-slave-kaniko as a Docker image. The jenkins-slave-kaniko image is used to create and push an image.

```
registry . cn - beijing . aliyuncs . com / acs - sample /
jenkins - slave - kaniko : 0 . 6 . 0
```

· Set container kubectl as follows.

Container Template		
Name	kubectl	0
Docker image	registry.cn-beijing.aliyuncs.com/ac	0
Always pull image		
Working directory	/home/jenkins	
Command to run	/bin/sh -c	0
Arguments to pass to the command	cat	
Allocate pseudo-TTY		
EnvVars	Add Environment	
	Variable	
	List of environment variables to set in agent pod	

Use jenkins-slave-kubectl as a Docker image. The jenkins-slave-kubectl image is used to deploy the application.

registry . cn - beijing . aliyuncs . com / acs - sample /
jenkins - slave - kubectl : 1 . 11 . 5

• Set container maven as follows.

Container Template		
Name	maven	0
Docker image	registry.cn-beijing.aliyuncs.com/ac	?
Always pull image		
Working directory	/home/jenkins	0
Command to run	/bin/sh -c	?
Arguments to pass to the command	cat	0
Allocate pseudo-TTY		
EnvVars	Add Environment Variable	
	List of environment variables to set in agent pod	

Use jenkins-slave-maven as a Docker image.The jenkins-slave-maven image is used by mvn to package and build the application.

registry . cn - beijing . aliyuncs . com / acs - sample / jenkins - slave - maven : 3 . 3 . 9 - jdk - 8 - alpine

i. Set image repository permission for kaniko as follows.

Add Cont	ainer 🔻	
List of containe	er in the agent pod	
Environ	ment Variable	
Key	DOCKER_CONFIG	0
Value	/home/jenkins/.docker	?
	Delete Environment Variable	•
Add Envir	ronment Variable 🔻	
List of environr	nent variables to set in all container of the pod	
Secret	/olume	
Secret nar	ne jenkins-docker-cfg	
Mount pat	h /home/jenkins/.docker	0
	Delete Volume	•

- j. Click Save.
- 2. To set image repository permission, use kubectl to create jenkins-docker-cfg secret in the target Kubernetes cluster.

In this example, the Beijing image repository provided by Alibaba Cloud is used.

```
$ docker login - u xxx - p xxx registry . cn - beijing .
aliyuncs . com
Login Succeeded
```

```
$ kubectl create secret generic jenkins - docker - cfg - n
ci -- from - file =/ root /. docker / config . json
```

- 3. Create demo-pipeline and access the application service.
 - a. On the Jenkins home page, click demo-pipeline.

🚱 Jenkins						6 Search		Ø demo log out
Jenkins >								ENABLE AUTO REFRESH
謍 New 任务								Zadd description
العديقة المحمد المحم		所有	•					
Build History		s	w	名称 ↓	Last Success	Last Failure	Last Duration	收藏
			*	demo-freestyle	N/A	N/A	N/A	
鵗 我的视图			*	demo-pipeline	N/A	N/A	N/A	🔊 🕁
🝚 打开 Blue Ocean		Icon: <u>S M</u>	L					
🗞 Lockable Resources						Legend M RSS 1	or all M RSS for failures	SS for just latest builds
条 凭据								
新建视图								
Build Queue								
No builds in the queue.								
Build Executor Status	-							

- b. In the left-side navigation pane, click Build with Parameters.
- c. Modify the parameters according to your image repository information. In this example, the source code repository is master, and the image is registry.cn-beijing.aliyuncs.com/haoshuwei:stable.

🧶 Jenkins			4 Qsearch (1) admin 1 log out
Jenkins → demo-pipeline →			
 ▲ Back to Dashboard Q Status ➢ Changes 			ne demo-pipeline urires parameters:
Build with Parameters		image_tag	stable
S Delete Pipeline		origin_repo	registry.cn-beijing.aliyuncs.com/haoshuwei
Configure		repo	jenkins-java-demo
Q Full Stage View		branch	master
Rename		Build	
Pipeline Syntax			
🔅 Build History	trend ==		
find	x		
🔊 RSS for all 🔊 RS	SS for failures		

- d. Click Build.
- e. Click Build History to check the result. The following figure indicates a success.



f. Log on to the *Container Service console* to view the IP addresses of the services provided by the application.

Click *Here* to obtain the source code repository used in the example.

For more information, see Container Service.

For more information, see kaniko.

10.2 Use GitLab CI to run a GitLab runner and activate a pipleline in a Kubernetes cluster

This topic describes how to install and register a GitLab runner in a Kubernetes cluster and add a Kubernetes executor to build an application. After completing the preceding operations, you can build a Java application source code project. The example in this topic shows a CI/CD pipleline that includes the following actions in order: compiling and building the code, packaging the application image, and deploying the application.

Background

In the following example, a Java software project is built and deployed in a Kubernetes cluster that runs on Alibaba Cloud Container Service for Kubernetes. The example shows how to use GitLab CI to run a GitLab runner, set a Kubernetes executor, and activate a pipeline in an Alibaba Cloud Kubernetes cluster.

Create the GigLab source code project and upload the example code

1. Create the GitLab source code project.

In this example, the address of the created GitLab source code project is as follows:

http://xx.xx.xx. xx/demo/gitlab-java-demo.git

2. Run the following commands to obtain the example code and upload it to GitLab:

```
$ git clone https :// code . aliyun . com / CodePipeli ne /
gitlabci - java - demo . git
$ git remote add gitlab http :// xx . xx . xx . xx / demo /
gitlab - java - demo . git
```
\$ git push gitlab master

Install the GitLab runner in a Kubernetes cluster

- 1. Obtain the GitLab runner registration information.
 - a. Obtain the registration information of the runner specific to the project.
 - A. Log on to GitLab.
 - B. In the top navigation bar, choose Projects > Your projects.
 - C. On the Your projects tab page, select the target project.
 - D. In the left-side navigation pane, choose Settings > CI / CD.
 - E. Click Expand on the right of Runners.



F. Obtain the URL and the registration token.



- b. Obtain the group runner registration information.
 - A. In the top navigation bar, choose Groups > Your groups.
 - B. On the Your groups tab page, select the target group.

C. In the left-side navigation pane, choose Settings > CI / CD.

D. Click Expand on the right of Runners.

✿ Overview	Variables 🕑	Expand
D Issues 0	Variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. You can use variables for passwords, secret keys, or whatever you want.	
11 Merge Requests 0		
👪 Members	Runners 3 Register and see your runners for this group.	Expand
G Kubernetes		
Settings		
General		
Projects		

E. Obtain the URL and the registration token.



c. Obtain the shared runner registration information.



Only the administrator has permission to perform this action.

- A. In the top navigation bar, click
- B. In the left-side navigation pane of the Admin Area page, choose Overview > Runners.
- C. Obtain the URL and the registration token.

🖌 Admin Area	Admin Area > Runners	
Overview Dashboard Projects Users Groups Jobs Runners	A 'Runner' is a process which runs a job. You can set up as many Runners as you need. Runners can be placed on separate users, servers, even on your local machine. Each Runner can be in one of the following states: • ghated - Runner runs jobs from all unassigned projects • group - Runner runs jobs from all unassigned projects • group - Runner runs jobs from assigned projects • locked - Runner runs jobs from assigned projects • locked - Runner runs jobs from assigned to other projects • locked - Runner will not receive any new jobs	Set up a shared Runner manually 1. Install Gitlab Runner 2. Specify the following URL during the Runner setup: http://line 9. 3. Use the following registration token during setup: 8. Reset runners registration token 4. Start the Runner!

2. Run the following command to obtain and modify the Helm Chart of the GitLab runner:

```
$ git clone https :// code . aliyun . com / CodePipeli ne /
gitlab - runner . git
```

Modify the values . yaml file as follows:

```
## GitLab
            Runner
                     Image
##
image : gitlab / gitlab - runner : alpine - v11 . 4 . 0
             an imagePullP olicy
##
   Specify
##
imagePullP
            olicy : IfNotPrese
                                 nt
## Default
             container
                         image
                                 to
                                      use
                                            for
                                                  initcontai ner
init :
   image : busybox
         latest
   tag :
## The GitLab
                                ( with
                                         protocol )
                  Server
                           URL
                                                     that
                                                            want
     register the
to
                     runner
                               against
##
gitlabUrl : http://xx . xx . xx . xx /
##
   The
         Registrati on
                          Token
                                  for
                                        adding
                                                       Runners
                                                                 to
                                                 new
   the
         GitLab Server.
                           This
                                  must
##
        retrieved from
   be
                           your
                                  GitLab
                                           Instance .
##
runnerRegi strationTo ken : " AMvEWrBTBu - d8czEYyfY "
                all
## Unregister
                      runners
                                before
                                        terminatio n
##
unregister Runners : true
##
   Configure
               the
                     maximum
                               number
                                        of
                                                          jobs
                                             concurrent
##
concurrent :
              10
  Defines
                                  often
                                               check
                                                       GitLab
             in
                  seconds
                            how
                                          to
                                                                for
##
  а
      new
            builds
```

```
##
checkInter val: 30
## For RBAC support:
##
rbac :
  create : true
  clusterWid eAccess : false
## Configure integrated
                          Prometheus
                                     metrics
                                               exporter
##
metrics :
  enabled : true
## Configurat ion for
                         the
                              Pods
                                     that
                                           that
                                                 the
                                                       runner
  launches for each
                        new
                             job
##
runners :
 ## Default container
                       image to
                                    use
                                          for
                                               builds
                                                        when
none is specified
 ##
  image : ubuntu : 16 . 04
 ## Specify the tags
                        associated
                                     with
                                            the
                                                 runner .
                  list
Comma - separated
                         of
                            tags .
 ##
  tags : " k8s - runner "
          all containers with the
 ## Run
                                       privileged
                                                   flag
enabled
 ## This
          will allow the docker : dind image
                                                    to
                                                         run
if
          need to run Docker
     you
 ## commands . Please read the docs
                                          before
                                                  turning
this on :
 ##
  privileged : true
 ## Namespace
                   run Kubernetes jobs in (defaults
              to
                                                           to
       same
              namespace of this
  the
                                  release )
 ##
  namespace : gitlab
  cachePath : "/ opt / cache "
  cache : {}
  builds : {}
  services : {}
  helpers : {}
 resources : {}
```

3. Run the following command to install the GitLab runner.

```
$ helm package .
Successful ly packaged chart and saved it to : / root
/ gitlab / gitlab - runner / gitlab - runner - 0 . 1 . 37 . tgz
```

```
$ helm install -- namespace gitlab -- name gitlab - runner
*. tgz
```

Check whether the related deployment/pod has been started. If the related deployment/pod has been started, the GitLab runner that has been registered in GitLab is displayed.

🦊 GitLab Projects 🗸 G	roups ~ Activity Milestones Snippets 🕒 🌮	Search or jump to	۹	0)	n	୯ 🧃) ~
G gitlab-java-demo ☆ Project	 Click the button below to begin the install process by navigating to the Kubernetes page Select an existing Kubernetes cluster or create a 	Disable shared Runners for this project					
Repository Issues 0	new one 3. From the Kubernetes cluster details view, install Runner from the applications list Install Runner on Kubernetes	This GitLab instance does not provide any shared Runners yet. Instance administrators can register shared Runners in the admin area.					
1 Merge Requests		Group Runners					
🦃 CI/CD	Setup a specific Runner manually	GitLab Group Runners can execute code for all the projects in this group. They can be managed using the					
G Operations	1. Install GitLab Runner 2. Specify the following URL during the Runner setup:	Runners API.					
🖱 Wiki	http:// ر م 3. Use the following registration token during setup: م	Disable group Runners for this project					
🕉 Snippets	4. Start the Runner!						
Settings		This group does not provide any group Runners yet. Group maintainers can register group runners in the Group CI/CD					
General	Runners activated for this project	settings					
Members	● b8b9c3e0 Pause Remove Runner						
Integrations	gitlab-runner-gitlab-runner-7c69979868-4fbks #12						
Repository	k8s-runner						
≪ Collapse sidebar	L						

Set the GtiLab runner cache

GitLab runners have a limited cache capacity. Therefore, you need to mount a volume to your GitLab runner so that the volume functions as the GitLab runner cache. In this example, the / opt / cache directory is used as the GitLab runner cache by default. You can modify the runners . cachePath field of the values . yaml file to change the default cache directory.

For example, to create a maven cache, add the MAVEN_OPTS variable to variables and specify a local cache directory as follows:

```
variables :
   KUBECONFIG : / etc / deploy / config
   MAVEN_OPTS : "- Dmaven . repo . local =/ opt / cache /. m2 /
repository "
```

To mount a new volume, modify the following fields in the templates / configmap

```
. yaml file:
cat >>/ home / gitlab - runner /. gitlab - runner / config . toml
<< EOF
     [[ runners . kubernetes . volumes . pvc ]]
     name = " gitlab - runner - cache "
     mount_path = "{{ . Values . runners . cachePath }}"
```

EOF

This means that you need to modify the settings of *config*. *toml* during the period between the time at which the GigLab runner was registered and the time at which the runner has not yet been started.

Set the global variables

- 1. In the top navigation bar, choose Projects > Your projects.
- 2. On the Your projects tab page, select the target project.
- 3. In the left-side navigation pane, choose Settings > CI / CD.
- 4. Click Expand on the right of Runners to add available to the GitLab runner.

 Project Repository Issues 	Auto DevOps Auto DevOps will automatically build, test, and deploy your application based and Delivery configuration. Learn more about Auto DevOps	d on a predefined Continuous Integration
 Merge Requests CI / CD 	Runners Register and see your runners for this project.	
OperationsWiki	Variables 🕝 Variables are applied to environments via the runner. They can be protected	by only exposing them to protected
& Snippets	branches or tags. You can use variables for passwords, secret keys, or whatev Input variable key Input variable key Input variable value	Protected (3
General Members	Save variables Hide values	
Integrations Repository	Pipeline triggers Triggers can force a specific branch or tag to get rebuilt with an API call. The user including their access to projects and their project permissions.	se tokens will impersonate their associated
CI / CD		

In this example, add the following three variables:

- REGISTRY_USERNAME: indicates a registry username.
- · REGISTRY_PASSWORD: indicates a registry password.
- kube_config: indicates a string of KubeConfig code characters.

Run the following command to generate a string of KubeConfig code characters:

```
echo $( cat ~/. kube / config | base64 ) | tr - d " "
```

Edit the . gitlab - ci . yml file

Edit the . *gitlab – ci . yml* file to compile and build the Java demo source code project, push the application image, and deploy the application. For more information, see the .gitlab-ci.yml.example of the gitlabci-java-demo source code project. This section first shows an example of an entire YAML file, and then describes key sections of the file in more detail.

The entire . gitlab - ci . yml file is as follows:

```
image : docker : stable
stages :

    package

   docker_bui ld
 - deploy_k8s
variables :
  KUBECONFIG : / etc / deploy / config
mvn_build_ job :
    image : registry . cn - beijing . aliyuncs . com / codepipeli
ne / public - blueocean - codepipeli ne - slave - java : 0 . 1 -
63b99a20
  stage : package
  tags :
  - k8s - test
  script :
          package – B – DskipTests
   - mvn
- cp
           target / demo . war / opt / cache
docker_bui ld_job :
  image : registry . cn - beijing . aliyuncs . com / codepipeli
ne / public - blueocean - codepipeli ne - slave - java : 0 . 1 -
63b99a20
  stage : docker_bui ld
  services :
  - docker : dind
  variables :
    DOCKER_DRI VER : overlay
    DOCKER_HOS T : tcp :// localhost : 2375
  tags :
   - k8s - test
  script :
  - docker login - u $ REGISTRY_U SERNAME - p $ REGISTRY_P
ASSWORD registry . cn - beijing . aliyuncs . com
   - mkdir target

    - cp / opt / cache / demo . war target / demo . war
    - docker build - t registry . cn - beijing . aliyuncs . com

/ gitlab - demo / java - demo :$ CI_PIPELIN E_ID
- docker push registry . cn - beijing . aliyuncs . com /
gitlab - demo / java - demo :$ CI_PIPELIN E_ID
deploy_k8s _job :
  image : registry . cn - beijing . aliyuncs . com / codepipeli
ne / public - blueocean - codepipeli ne - slave - java : 0 . 1 -
63b99a20
  stage : deploy_k8s
  tags :
   - k8s - test
  script :
   - mkdir - p / etc / deploy
      echo $ kube_confi g | base64 - d > $ KUBECONFIG
   _
      sed - i " s / IMAGE_TAG /$ CI_PIPELIN E_ID / g " deployment
. yaml
            deployment . yaml
      cat
      kubectl apply - f deployment.yaml
```

The. gitlab - ci . yml file defines a pipeline that is executed in three stages.

```
image : docker : stable
                        # If no
                                   image
                                           is
                                                specified
                                                           for
 each step of the
                        pipeline, the docker : stable
                                                         image
           by default.
 is
      used
stages :
                         #

    package

                          package the
                                          soucre
                                                   code
```

- docker_bui ld # build , package , and push the applicatio n image - deploy_k8s # deploy the applicatio n variables : KUBECONFIG : / etc / deploy / config # define the global variable , namely , KUBECONFIG

· Package the maven source code

• Build, package, and push the application image

```
- docker : dind
  variables :
   DOCKER_DRI VER : overlay
   DOCKER_HOS T : tcp :// localhost : 2375 # Connect the
Docker Daemon .
  tags :
                           # GitLab
                                      Runner
                                              tag
  - k8s - test
  script :
  - docker login - u REGISTRY_U SERNAME - p $ REGISTRY_P
ASSWORD registry . cn - beijing . aliyuncs . com # log on
to the registry
   - mkdir target

    - cp / opt / cache / demo . war target / demo . war
    - docker build - t registry . cn - beijing . aliyuncs .

com / gitlab - demo / java - demo :$ CI_PIPELIN E_ID . #
Package the Docker image. The used tag is pipeline ID.
                                                     the
  - docker push registry . cn - beijing . aliyuncs . com /
gitlab - demo / java - demo :$ CI_PIPELIN E_ID  # Push
                                                          the
  Docker
         image .
```

• Deploy the application

script :	
– mkdir – p / etc / deploy	
- echo \$ kube_confi g base64 - d > \$ KUBECONFIG	#
Set the config file that is used to connect	
Kubernetes cluster.	
- sed - i " s / IMAGE_TAG /\$ CI_PIPELIN E_ID / g "	
deployment . yaml # Dynamicall y replace the image	tag
of the deployment file .	U
– kubectl apply – f deployment . yaml	

Activate the pipeline

After you submit the . *gitlab – ci . yml* file, the gitlab-java-demo project automatically detects this file and activate the pipeline.

🦊 GitLab 🛛 Projects 🗸 🤇	Groups – Activity Milestones Snippets	۴ ا	0	 Search or jump to 	 ባ ከ ድ 🌐
G gitlab-java-demo	demo > gitlab-java-demo > Pipelines				
✿ Project	All 5 Pending 0 Running 1	Finished 4 Branches Tage	5	Run Pipeline	Clear Runner Caches Cl Lint
Repository	Status Pipeline	Commit	Stages		
O) Issues O	#85 by 🔅	약master ⊸c66f76ca			_
Merge Requests	running latest	🛞 add .gitlab-ci.yml	.		×
🔗 CI/CD					
Pipelines					
Jobs					
Schedules					
Charts					
G Operations					
🗋 Wiki					
🕉 Snippets					
de Cattinga					
A horas a					
🦊 GitLab 🏾 Projects 🗸 🖓	Groups ~ Activity Milestones Snippets	<u>له</u>	÷	 Search or jump to 	Q D) D C ∰ ✓
G gitlab-java-demo	Groups - Activity Milestones Snippets demo > gitlab-java-demo > Pipelines > #85	۴ ک	C	 Search or jump to 	໑ ຫ ໊ ເ ∰×
	demo > gitlab-java-demo > Pipelines > #85	🖿 🌶	G	 Search or jump to 	ዓ ወ ክ ድ 🌐 ›
G gitlab-java-demo	demo > gitlab-java-demo > Pipelines > #85		C	 Search or jump to 	፝ ር) ሽ ଓ ∰ γ
G gitlab-java-demo ☆ Project	demo > gitlab-java-demo > Pipelines > #85	nutes ago by 🏦 Administrator	C	 Search or jump to 	Q D D E ∰ ✓
 G gitlab-java-demo ☆ Project ᢙ Repository 	demo > gitlab-Jave-demo > Pipelines > #85 @ passed Pipeline #85 triggered 4 mil add .gitlab-ci.yml @ 3 jobs from master (queued for 28 s	nutes ago by 🏦 Administrator	0	 Search or jump to 	Q D) Ϊ) ເ⊭ ∰γ
G gitlab-java-demo 合 Project ⑦ Repository ① Issues 0	demo > gitlab-java-demo > Pipelines > #85	nutes ago by 🏦 Administrator	C	 Search or jump to 	Q D N & 🔀 v
G gitlab-java-demo 合 Project ① Repository D Issues 0 门 Merge Requests 0	demo > gitlab-java-demo > Pipelines > #85 ⓒ passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml ⓒ 3 jobs from master (queued for 28 s ↔ c66f76ca %	nutes ago by 🏦 Administrator	C	 Search or jump to 	Q D N & 🔀 v
G gitlab-java-demo 合 Project ⊡ Repository D Issues 0 i1 Merge Requests 0 14 CI / CD	demo > gitlab-Jave-demo > Pipelines > #85 @ passed Pipeline #85 triggered 4 mil add .gitlab-ci.yml @ 3 jobs from master (queued for 28 s	nutes ago by 🏦 Administrator	6	Search or jump to	Q D N & ∰v
G gitlab-java-demo 合 Project C Repository D Issues 0 IN Merge Requests 0 IV CI / CD Pipelines	demo > gitlab-java-demo > Pipelines > #85 ⊘ passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml Ø 3 jobs from master (queued for 28 s ↓ c66f76ca ···· ♀ Pipeline Jobs 3	nutes ago by 🏦 Administrator	C Deploy_k8s	Search or jump to	Q D N & ∰v
G gitlab-java-demo	demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do	nutes ago by 🎉 Administrator econds)		Search or jump to	Q D N & D D
G gittab-java-demo	demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do	nutes ago by 🎋 Administrator econds) cker_build	Deploy_k8s	Search or jump to	Q D N & D D
gitlab-java-demo Project Repository Issues Issues Issues Merge Requests Image Requests <li< td=""><td>demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do</td><td>nutes ago by 🎋 Administrator econds) cker_build</td><td>Deploy_k8s</td><td>Search or jump to</td><td></td></li<>	demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do	nutes ago by 🎋 Administrator econds) cker_build	Deploy_k8s	Search or jump to	
gittab-java-demo	demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do	nutes ago by 🎋 Administrator econds) cker_build	Deploy_k8s	Search or jump to	
G gitlab-java-demo	demo > gitlab-java-demo > Pipelines > #85 © passed Pipeline #85 triggered 4 mi add .gitlab-ci.yml © 3 jobs from master (queued for 28 s -> c66f76ca - Po Pipeline Jobs 3 Package Do	nutes ago by 🎋 Administrator econds) cker_build	Deploy_k8s	Search or jump to	

Access the application service

If no namespace is specified in the deployment file, the application is deployed in the GitLab namespace by default.

```
$ kubectl - n gitlab get svc
NAME TYPE CLUSTER - IP EXTERNAL - IP
PORT (S) AGE
java - demo LoadBalanc er 172.19.9.252 xx.xx.
xx.xx 80:32349 / TCP 1m
```

Visit xx.xx.xx/demo in your browser to check the result.

For more information, see Container Service and GitLab Cl.

10.3 Deploy Jenkins in a serverless Kubernetes cluster and perform an application pipeline build

This topic describes how to deploy Jenkins, a continuous integration environment, in an Alibaba Cloud serverless Kubernetes cluster, and how to perform an application pipeline build. The example in this topic details the pipeline build, including how to compile the source code of the application, build and push the application image, and deploy the application.

Prerequisites

You have created a serverless Kubernetes cluster. For more information, see *Create a* serverless Kubernetes cluster.

Deploy Jenkins

1. Run the following command to download the Jenkins deployment file:

```
$ git clone https://github.com/AliyunCont ainerServi ce
/ jenkins - on - serverless . git
$ cd jenkins - on - serverless
```

2. Persist the jenkins_home directory.

Serverless Kubernetes clusters do not support cloud disks. To persist the jenkins_home directory, create the nfs volume, then modify the serverless -

```
k8s - jenkins - deploy . yaml file to add the following field annotations and set nfs parameters:
```

```
# volumeMoun ts :
    # - mountPath : / var / jenkins_ho me
    # name : jenkins - home
    ....
```

```
# volumes :
# - name : jenkins - home
# nfs :
# path : /
```

server :

3. Run the following command to deploy Jenkins:

```
$ kubectl apply - f serverless - k8s - jenkins - deploy . yaml
```

- 4. Log on to Jenkins.
 - a. Log on to the Container Service console.
 - b. In the left-side navigation pane, choose Discovery and Load Balancing > Service.
 - c. Click the external endpoint of the Jenkins service to log on to Jenkins.

Container Service - Kubernetes 🔻		Service								Refresh	Create
Application	*	Help: 🔗 Canary release									
Deployment	ł.	Clusters k8s-cluster	 Namespace 	test v				s	earch By Nam	ie	Q
StatefulSet	L	Name	Туре	Time Created	ClustersIP	InternalEndpoint	ExternalEndpoint 3				Action
DaemonSet	L	jenkins-demo-jenkins	LoadBalancer	02/18/2019,15:29:06	1.00	jenkins-demo-jenkins:8080 TCP jenkins-demo-jenkins:31913 TCP	-	Details	Update	View YAML	Delete
Job CronJob		jenkins-demo-jenkins- agent	ClusterIP	02/18/2019,15:29:06	1	jenkins-demo-jenkins-agent:50000 TCP	*	Details	Update	View YAML	Delete
Pods											
Volumes Claim											
Release											
Discovery and Load B	1										
Service 2											

d. On the Jenkins logon page, enter the user name and the password. The default user name and password are admin. We recommend that you modify them after you log on to Jenkins.

Welcome to Jenkins!
Username
Password
Sign in
Keep me signed in

Create a cluster certificate and an image repository certificate, and build and deploy an application

- 1. Set Kubernetes Cloud parameters to dynamically create a slave pod.
 - a. In the left-side navigation pane, click Manage Jenkins.
 - b. Click Configure System.
 - c. In the Cloud area, enter the API server URL in KubeConfig as the Kubernetes URL.

Cloud		
Kubernetes		
Name	kubernetes	0
Kubernetes URL	https://www.integrates.com/6443	0

d. Click Add on the right of Credentials.

Disable https certificate check	×	0
Kubernetes Namespace	test]
Credentials	- none - 🔻 🛀 Add 🕶	
	Test Connection	

Before adding a credential, you must obtain KubeConfig on the Basic

Information tab page of the target Kubernetes cluster.



In the displayed dialog box, set the following parameters:

Jan Star	Jenkins Credentials Provider: Jenkins							
🛁 Add	Credentials							
Domain	Global credentials (un	restricted)	•					
Kind	Docker Host Certificat	e Authentication	•					
	Scope	Global (Jenkins, nodes, items, all child items, etc)	0					
	Client Key							
			8					
	Client Certificate		1					
			*					
	Server CA Certificate		1					
	ID		0					
	Description		0					
Add	Cancel							

- · Kind: Select Docker Host Certificate Authentication.
- Client Key: Paste the copied client-key-data content in KubeConfig.
- Client Certificate: Paste the copied client-certificate-data content in KubeConfig.
- · ID: Enter the certificate ID. In this example, k8sCertAuth is entered.
- Description: Enter description content.
- e. Click Add.
- f. Test connectivity.

Select the added credential in the preceding step from the Credentials dropdown list, and then click Test Connection.

Kubernetes Namespace	default	
Credentials	k8sCertAuth (k8sCertAuth) ♦ 🛛 🥪 Add ◄	
	Connection test successful Tes	t Connection

g. Enter the external endpoint of the jenkins service as Jenkins URL, and enter the external endpoint of the jenkins-agent service as Jenkins tunnel.

Jenkins URL	http://www.awarupeticies.8080	0
Jenkins tunnel	110 1 .184.0214 50000	0
Connection Timeout	0	0
Read Timeout	0	0

- h. Click Save.
- 2. To set image repository permission, use kubectl to create jenkins-docker-cfg secret in the target serverless Kubernetes cluster.

In this example, the Beijing image repository provided by Alibaba Cloud is used.

\$ docker login - u xxx - p xxx registry . cn - beijing .
aliyuncs . com
Login Succeeded

```
$ kubectl create secret generic jenkins - docker - cfg --
from - file =/ root /. docker / config . json
```

- 3. Create demo-pipeline and access the application service.
 - a. On the Jenkins home page, click demo-pipeline.
 - b. In the left-side navigation pane, click Build with Parameters.
 - c. Modify the parameters according to your image repository information. In this example, the source code repository branch is serverless, and the image is registry.cn-beijing.aliyuncs.com/haoshuwei:stable.

Jenkins > demo-pipeline >		
A Back to Dashboard Status	Pipelin	e demo-pipeline
Changes		res parameters:
Build with Parameters	image_tag	stable
O Delete Pipeline	origin_repo	registry.cn-beijing.aliyuncs.com/haoshuwei
Configure	repo	jenkins-java-demo
 ✓ 打开 Blue Ocean 	branch	serverless
Rename	api_server_url	https serverless-a.kubernetes.cn-hangzhou.aliyuncs.com:6443
Pipeline Syntax	Build	
Build History trend =>		
find x		
S RSS for all S RSS for failures		

- d. Click Build.
- e. Click Build History to check the result. The following figure indicates a success.

✓ <u>demo-pipeline</u> 1			Pipeline				১	€ \$	Logout	
Branch: — Commit: —										
	Start		Pro	Deploy to ubernetes I Deploy to oduction En Deploy to ging001 En	End					
Deploy to Kubernetes / Deplo						Q) <u>Restart De</u>	<u>eploy to Kub</u>	ernetes 🔽	3 ₹
 Started by u Running in D PipeLine] m Still waiting Vaiting for i Agent specif * [anis] * [anis] * [anis] * [kanis] re * [kaven] re 	ser Administrator mrability level: MAX_SURVIVABILITY ode g to schedule task nest available executor nest available executor nestionin (Kubernets Pod Template] (slav nestion: nebeijing,aliyuncs.com/acs-samp gistry, cn-beijing,aliyuncs.com/acs-samp gistry, cn-beijing,aliyuncs.com/acs-samp gistry, cn-beijing,aliyuncs.com/acs-samp gistry, cn-beijing,aliyuncs.com/acs-samp lave-pipetine-jo5j4 in /home/jenkins/wo h	e-pipeline): 2/jenkins-slave-jnlp:3.14–1 le/jenkins-slave-kaniko:0.6.0 mple/jenkins-slave-kubectl:1.11.5 le/jenkins-slave-maven:3.3.9-jdk-8-alp	bine							

f. Log on to the *Container Service console* to view the IP addresses of the services provided by the application.

The source code repository used in this topic can be found at https://github.com/ AliyunContainerService/jenkins-demo.

10.4 Use Bamboo to deploy a remote agent and run a build plan

This topic describes how to use Bamboo to deploy a remote agent in a Kubernetes cluster implemented with Alibaba Cloud Container Service for Kubernetes (ACK) and how to use the agent to run a build plan for an application. In this topic, an example application complied in Java is created and deployed to a Kubernetes cluster.

Prerequisites

- A Kubernetes cluster is created by using ACK. For more information, see *Create a Kubernetes cluster*.
- A Bamboo server is created.

Source code of the application to be deployed

The source code of the application created in this topic can be obtained with the following address:

```
https :// github . com / AliyunCont ainerServi ce / jenkins - demo
. git
```

After you access the corresponding GitHub page, you can find the source code in the bamboo branch.

Deploy a remote agent in a Kubernetes cluster

1. Create a kaniko-docker-cfg secret.



This secret is used to set the permissions for accessing the target image repository required by building tasks in the remote agent and using kaniko to push a container image.

a. Log on to your Linux server by using the root account to run the following command to create a / root /. docker / config . json file:

docker login registry . cn - hangzhou . aliyuncs . com

b. Use *Cloud Shell* to connect to the target Kubernetes cluster, and then run the following command to create a kaniko-docker-cfg secret:

kubectl - n bamboo create secret generic kaniko docker - cfg -- from - file =/ root /. docker / config . json

2. Create a Bamboo agent in the target Kubernetes cluster.

ServiceAcc ount and ClusterRol eBinding are created to set the permissions required for kubectl to deploy an application to the target Kubernetes cluster.

a. Create a file bamboo - agent . yaml and copy the following code to the file:

```
apiVersion : v1
kind : ServiceAcc ount
metadata :
  namespace : bamboo
  name : bamboo
apiVersion : rbac . authorizat ion . k8s . io / v1
kind : ClusterRol eBinding
metadata :
  name : bamboo - cluster - admin
subjects :
    kind :
           ServiceAcc ount
    name :
           bamboo
    namespace : bamboo
roleRef :
  kind :
         ClusterRol e
  name : cluster - admin
  apiGroup : rbac . authorizat ion . k8s . io
apiVersion : apps / v1beta2
kind : Deployment
metadata :
  name : bamboo - agent
spec :
  replicas : 1
  selector :
    matchLabel s :
      app : bamboo - agent
  template :
   metadata :
```

labels : app : bamboo - agent spec : serviceAcc ountName : bamboo containers : - name : bamboo - agent env : - name : BAMBOO_SER VER_URL value : http:// xx . xx . xx . xx : 8085
image : registry . cn - hangzhou . aliyuncs . com / haoshuwei / docker - bamboo - agent : v1 imagePullP olicy : Always volumeMoun ts : mountPath : / root /. docker / name : kaniko - docker - cfg volumes : name : kaniko - docker - cfg secret : secretName : kaniko - docker - cfg

- **b.** Run the kubectl n bamboo apply f bamboo agent . yaml command to create the Bamboo agent.
- c. Run the following command to view logs of the agent:

```
kubectl - n bamboo logs - f < bamboo agent pod name
>
Note:
```

4

You must replace < bamboo agent pod name > with the file name that you use.

d. Log on to the Bamboo server you created. Then, in the upper-right corner, click , and select Agent to view the deployed agent.

Samboo My Bamboo	Projects Build - Deploy - Specs - Reports - Cr	eate 🗸		Search	a @ 0 (
Bamboo administra	tion				
BUILD RESOURCES	Agents summary	Add local agent	Enable remote agent authentication	Enable security token verification	Install remote agent
Agents Agent matrix	An agent is a service that executes Bamboo builds and deplo You can also use this matrix to determine which agents can			nts.	
xecutables	Local agents				Server capabiliti
DKs erver capabilities Blobal variables	Local agents run on the Bamboo server. Select: All, None, Idle, Disabled Action: Delete Disable	Enable			
inked repositories	Agent			Status	Operations
nared credentials	Default Agent			🎍 Idle	View Edit
epository settings	Remote agents			Shared remote capabilities Disabl	le remote agent supp
ASTIC BAMBOO	Remote agents run on computers other than the Bamboo set	rver.			
onfiguration	Online remote agents Offline remote agents				
ANS	There is currently 1 remote agent online.				
oncurrent builds uarantine settings	Select: All, None, Idle, Disabled Action: Delete Disable	e Enable			
piry	Agent			Status	Operations
Ik action	bamboo-agent-755cf799bc-2hpm8			🍦 Idle	View Edit
ild monitoring move plans	Jun 5, 2019 10:25:24 AM A remote agent is loading on Jun 5, 2019 10:25:29 AM A remote agent is loading on Jun 5, 2019 10:26:29 AM Remote agent "bamboo-agent-7	bamboo-agent-77b	6dd97c-vcc9d (114.55.243.78).		
ove plans	Jun 5, 2019 11:02:06 AM A remote agent is loading on Jun 5, 2019 11:02:11 AM A remote agent is loading on	bamboo-agent-77b bamboo-agent-77b	6dd97c-cctmz (114.55.243.78). 6dd97c-cctmz (114.55.243.78).		
Bulk edit plan permissions	Jun 5, 2019 11:02:53 AM Remote agent "bamboo-agent-7 Jun 5, 2019 11:08:33 AM A remote agent is loading or Jun 5, 2019 11:08:38 AM A remote agent is loading or	7b96dd97c-cctmz" bamboo-agent-7b4	as registered. 9cb67-gmtbk (114.55.243.78).		
rtifact handlers	Jun 5, 2019 11:09:20 AM Remote agent "bamboo-agent-7 Jun 5, 2019 11:54:06 AM A remote agent is loading on	b469cb67-gmtbk" ha	s registered.		

Powered by a free Atlassian Bamboo evaluation license. Please consider purchasing it today

Configure a build plan

- 1. Create a build plan.
 - a. Log on to the Bamboo server you created, and choose Create > Create plan.
 - b. Select bamboo ack demo from the Project drop-down list, set Plan name, Plan key, and Plan description, select java - demo from the Repository host drop-down list, and then click Configure plan.

reate plan		Configure plan Config	-
Configure plan		How to create a buil	d plan
	everything about your build process. Each plan has a Default job when it is s, and the ability to add more jobs will be available to you after creating this		
Project and build pl	an name		
Project	bamboo-ack-demo 🔹		
	The project the new plan will be created in.		
Plan name [*]	bamboo-ack-demo		
Plan key*	BAM11		
	For example WEB (for a plan named Website)		
Plan description			
Plan access	Allow all users to view this plan. Applies to new project as well.		
Link repository to n	ew build plan		
Repository host*	Previously linked repository		
	java-demo 💌		
	Link new repository		
	None None		
	Configure plan Cancel		
	Conigure plan Cancer		

- 2. Configure a job that contains four required tasks for the build plan.
 - a. Confirm and save the setting of the source code repository.



In the preceding step where you create the build plan, your setting for Repository host specifies the source code repository. You can retain or modify this setting.

A. In the Create tasks area, click Source Code Checkout.

Source Code Checkout (8) Checkout Default Repository	Source Code Checkout configuration	How to use the Sour Code Checkout ta
inal tasks Are always executed even if a previous task fails	Task description	Code Checkout ta
Drag tasks here to make them final	Checkout Default Repository	
Add task	Disable this task	
	You can check out one or more repositories with this Task. You Plan's <i>Default Repository</i> or specify a <i>Specific Repository</i> . You to this Plan via the Plan configuration. Repository*	
	java-demo	•
	Default always points to Plans default repository.	
	Checkout Directory	
	(Optional) Specify an alternative sub-directory to which the co	de will be checked out.
	 Force Clean Build Removes the source directory and checks it out again prior significantly increase build times. Add repository 	to each build. This may

- B. On the Source Code Checkout configuration page, select a new repository from the Repository drop-down list, and then click Save.
- b. Add the command type of mvn to use the mvn tool.
 - A. In the Create tasks area, click Add task. Then, on the displayed Task types page, click Command.



B. On the Command configuration page, set Task description, Executable, and Argument, and then click Save.

Source Code Checkout Checkout Default Repository	Command configuration How to use the Com
Command	Task description
inal tasks Are always executed even if a previous task fail	is mvn
Drag tasks here to make them final	Disable this task
Add task	Executable
	mvn Add new executable
	Argument
	package -B -DskipTests
	Argument you want to pass to the command. Arguments with spaces in them must be
	quoted.
	Environment variables
	Extra environment variables. e.g. JAVA_OPTS="-Xmx256m -Xms128m". You can add mu
	parameters separated by a space. Working subdirectory
	Specify an alternative subdirectory as working directory for the task.

c. Use kaniko to package and push the required container image to the target image repository.

A. In the Create tasks area, click Add task. Then, on the displayed Task types page, click Script.



B. On the Script configuration page, set Task description, and Script location, then click Save.



For the remaining parameters of the script, you can retain the default settings.

Checkout Default Repository	Script configuration	How to use the Script task
Command mvn	Task description	
mvn	kaniko	
# Script	Disable this task	
Final tasks Are always executed even if a previous task fails	Interpreter	
Drag tasks here to make them final	Shell	
Add task	An interpreter is chosen based on the shebang line of your scri	pt.
	Script location	
	Inline	
	Script body*	
	1 kaniko -f `pwd`/Dockerfile -c `pwd`destination	n=registry.cn-han
		li li
	Argument	
	Environment variables	
	Working subdirectory	
	Save Cancel	

In this example, the Script location is set as follows:

```
kaniko – f ` pwd `/ Dockerfile – c ` pwd ` -- destinatio
n = registry . cn – hangzhou . aliyuncs . com / haoshuwei /
bamboo – java – demo : latest
```

- d. Use kubectl to deploy the application on the target Kubernetes cluster.
 - A. In the Create tasks area, click Add task. Then, on the displayed Task types page, click Script.



B. On the Script configuration page, set Task description and Script location, then click Save.



For the remaining parameters of the script, you can retain the default settings.

III Checkout Default Repository	Script configuration now to use the Script task
Command	Task description
Command mvn	kube deploy
Script kaniko	Disable this task
	Interpreter
Script	Shell
Final tasks Are always executed even if a previous task fails	An interpreter is chosen based on the shebang line of your script.
Drag tasks here to make them final	Script location
Add task	Inline T
	mine
	Script body* 1 sed -i 's#IMAGE_URL#registry.cn-hangzhou.aliyuncs.com/haoshuwei/b
	2 kubectl apply -f ./
	4
	Argument
	Environment variables
	Working subdirectory
	Save Cancel

In this example, the Script location is set as follows:

```
sed - i ' s # IMAGE_URL # registry . cn - hangzhou .
aliyuncs . com / haoshuwei / bamboo - java - demo : latest
#' ./*. yaml
```

kubectl apply - f ./

3. Run the build plan.

a. On the Configure Job page, click Create.

ach plan has a default job when it is created. Here, yo ias been created.	ou can configure Tasks for this plan's default job. You can add more jobs to this plan once the pla
vith Docker. Run this job in [*] Agent environment Docker container Create tasks	g system. If you want to run your build in an isolated and controlled environment, you can do it ng directory using an executable. An example of task would be the execution of a script, a shell about tasks.
Command	Task created successfully.
Script kaniko	No task selected
Script kube deploy	Select a task from the list on the left to configure it.
Final tasks Are always executed even if a previous task fails Drag tasks here to make them final Add task	

b. In the upper-right corner of the page, choose Run > Run plan.

🕹 Bamboo My Bamboo Projects Build 🗸 Deploy 🖌 Specs 🗸 Reports 🗸 Create 🗸	Search	۵ 💿 د
build databoard / bamboo-ack-demo bamboo-ack-demo	← Ø → Run plan	Run ♥ ② Actions ♥
Plan summary Recent failures History Tests	Run custor	nized
Plan summary		Showing Last 25 builds

You can click the Logs tab to view the logs.

amboo-ack-demo				
15 was successful – Manual run by <u>bamboo</u>				
ummary Tests Commits Artifacts Logs Metadata				
ogs				
ne following logs have been generated by the jobs in this plan.	Expand all Collapse			
Job	Logs			
O Default Job Default Stage	Download or View			
5-Jun-201 11:58:40 INF0[0008] Skipping paths under /run/secrets/kubernets.ic/serviceaccount, a: 5-Jun-201 11:58:40 INF0[0008] Skipping paths under /var/run, as it is a whitelisted directory 5-Jun-201 11:58:40 INF0[0008] Skipping paths under /var/run, as it is a whitelisted directory 5-Jun-201 11:58:55 INF0[0014] Jusing files from context: [/rost/banboo-agent-hank/nl-data/bulid- 5-Jun-201 11:58:55 INF0[0014] ADD target/demo.war /vsr/local/tometa/websps/demo.war 5-Jun-201 11:58:55 INF0[0014] ADD target/demo.war /vsr/local/tometa/websps/demo.war 5-Jun-201 11:58:55 INF0[0014] Skipping paths under /vsr/local/tometa/skipping paths/Skipping/Sk	-dir/BAM-BAM-JOB1/target/demo.war] 33393b7ed210e99368026cf1cbb7a7e bb868427yab2c9c4cs9b8772da37874 Lef721cfc187b7c2379480f587de53c14 db7131790ec6a941478c5357de53c14 db7143795ac634143982557de53c14 db7143795ac63414398127178b7 db7847755ac6314459825546 db78471759ab7c534e18931778b7 Bd6c4748571759ab6c54379b377c19 Bd7341c9361c7758659a61230b4711 Bd6c47485717598de5c1379b37c19 Bd7341c9361c74786879818212a014 Bd6c432bc65221fd877b352b2c5 Bd304080bccc531cfac63a3c57			

- 4. Access the deployed application.
 - a. Run the kubectl n bamboo get svc command to view the Internet

IP address of the application.

1cz75s kaZ	bamboo]#	kubectl - n
PE	CLUSTER -	IP
RT (S)	AGE	
LoadBalanc	er xx.	xx . xx . xx
80 : 3266	58 / TCP	39m
	PE RT (S) LoadBalanc	1cz75s kaZ bamboo]# PE CLUSTER - RT (S) AGE LoadBalanc er xx. 80 : 32668 / TCP

b. In your browser, enter http:// EXTERNAL - IP to access the application.



Reference

For information about how to create the registry . cn - hangzhou . aliyuncs

. com / haoshuwei / docker - bamboo - agent : v1 image, see Docker bamboo agent.

For more information, see *Bamboo*.