

Alibaba Cloud Container Service

FAQ

Issue: 20180906

Legal disclaimer

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








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

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

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

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 General questions.....	1
1.1 What is Container Service?.....	1
1.2 What types of containers does Container Service support.....	1
1.3 Does Container Service charge fees.....	1
1.4 What are the application and service in Container Service.....	1
1.5 Why do I need to create an application when I want to start a container?.....	2
1.6 Does Container Service support external Docker images?.....	2
2 Swarm FAQs.....	3
2.1 How to deploy services in an orchestration template to each node of a cluster?.....	3
2.2 What are the differences between redeployment and rescheduling?.....	3
2.3 What if the application redeployment does not take effect?.....	4
2.4 How to expose or modify application external ports?.....	4
2.5 How do Container Service network models realize cross-host container communicat ion?.....	5
2.6 FAQs about routing service.....	7
2.7 FAQs about custom Server Load Balancer.....	12
2.8 How to troubleshoot access link issues?.....	14
2.9 How does Container Service isolate containers of different users?.....	19
2.10 "Invalid input for user ram ak or ak secret" is displayed when you create an applicatio n and add custom Server Load Balancer instances.....	19
2.11 FAQs about changing application configurations.....	19
2.12 Common errors leading to cluster creation failure.....	20
2.13 Configure DNS options in containers and optimize DNS resolution.....	22
2.14 How to increase data disks for Container Service Docker?.....	23
2.15 How to troubleshoot log issues?.....	25
2.16 Use Nginx + FPM in Container Service.....	27
2.17 Node exception.....	27
2.18 FAQs about the operating system and kernel of Container Service.....	29
2.19 Failed to pull the image.....	29
2.20 Does Container Service support granting permissions to sub-accounts in RAM console?.....	31

1 General questions

1.1 What is Container Service?

Container Service provides the high-performance and scalable container application management service, which enables you to manage the lifecycle of containerized applications by using Docker and Kubernetes. Container Service provides multiple application release methods and the continuous delivery ability, and supports microservice architecture. By simplifying the setup of container management cluster and integrating with the Alibaba Cloud abilities of virtualization, storage, network, and security, Container Service makes an ideal running cloud environment for containers.

1.2 What types of containers does Container Service support

Container Service currently supports Docker and Kubernetes.

1.3 Does Container Service charge fees

Billing method

Currently, Container Service is free of charge. However, you have to pay for the resources you used, such as Elastic Compute Service (ECS) instances and Server Load Balancer instances.

In Container Service, the automatically created or manually added ECS instances and Server Load Balancer instances are billed according to the ECS and Server Load Balancer prices respectively. For more information, see [ECS billing method](#) and [Server Load Balancer billing method](#).

1.4 What are the application and service in Container Service

You can divide an application into different microservices. Each microservice is composed of a group of containers with the same images and configurations, provides the atomic function, and connects with each other.

The preceding microservice is known as a service in Container Service. One or more services constitute an application in Container Service.

To manage applications by using containers, create an application with an image or an orchestration template first. The application created with an image is composed of one service. The application created with an orchestration template is composed of one or more services specified by the orchestration template.

1.5 Why do I need to create an application when I want to start a container?

An application is a management concept exposed by Container Service to users, and is created by using an image or an orchestration template.

We recommend that you divide a complex application into different components and Container Service helps you manage the component properties and the connection between components.

Each component is composed of a group of containers with the same images and configurations, which is known as a service in Container Service.

1.6 Does Container Service support external Docker images?

Container Service does not limit the Docker image sources, provided that it is permitted by the security policy of your server.

2 Swarm FAQs

2.1 How to deploy services in an orchestration template to each node of a cluster?

You have two methods to deploy services in a template to each node according to orchestration templates used for creating applications.

Compose V1/V2: Use the extension capability label `global` provided by Alibaba Cloud Container Service.

Compose V1/V2

You can set the service as a global service by using the extension capability label `global` provided by Alibaba Cloud Container Service.

If a service is set as `aliyun.global: true`, this service is deployed to each node of the cluster. A container is automatically deployed to nodes that are newly added to the cluster.



Note:

For more information about the `global` label, see [Label description](#).

Example:

```
monitor:
  image: sample
  labels:
    aliyun.global: true
```

2.2 What are the differences between redeployment and rescheduling?

Redeployment

Redeployment is to redeploy an application by using the application image.

You can redeploy an application when:

- You update the application image after deploying the application and want to deploy the application according to the updated image.
- You want to activate or recreate the stopped or deleted containers. Container Service activates the stopped containers and recreates the deleted containers when you redeploy the application.

Rescheduling

Rescheduling is to rebalance the number of containers running on each node, move containers from nodes with high loads to newly added nodes or nodes with low loads so as to rebalance the cluster load.

**Note:**

Rescheduling only changes the distribution of containers on nodes and does not pull the image to redeploy the application.

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

2.3 What if the application redeployment does not take effect?

Procedure

1. View image sha256 to check whether or not the image after the redeployment is the latest one.
Follow these steps to view the image sha256: Log on to the [Container Service console](#). Click **Applications** in the left-side navigation pane. Select the cluster in which the application resides from the Cluster list. Click the application name. Click the **Containers** tab. View the image. If the image is the latest one, the redeployment is successful.
2. Make sure whether or not you have mounted a data volume to the host. Redeployment will not update the data volume and the old data volume on the host will still be used. Therefore, any data volume configuration changes made in the new image will not take effect after the application is redeployed.

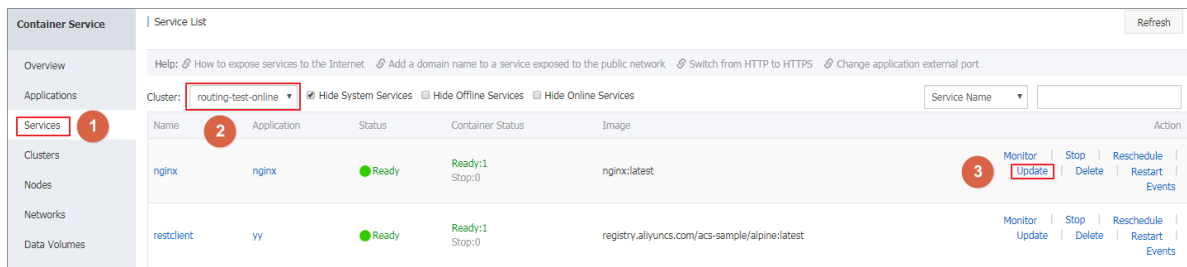
**Note:**

For more information about redeployment, see [Redeploy an application](#).

2.4 How to expose or modify application external ports?

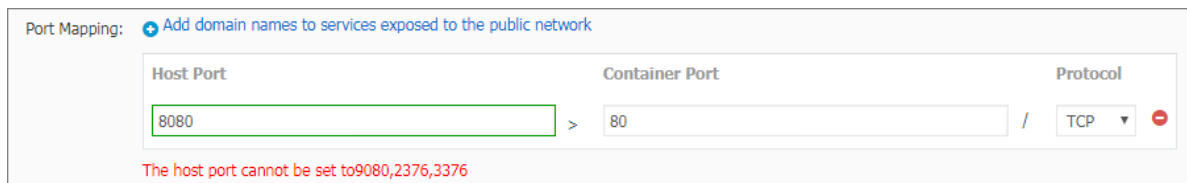
Procedure

1. Log on to the [Container Service console](#).
2. Click **Services** in the left-side navigation pane.
3. Select the cluster from the Cluster list.
4. Click **Update** at the right of the service (nginx in this example).



5. Enter the host port to be mapped in **Port Mapping**.

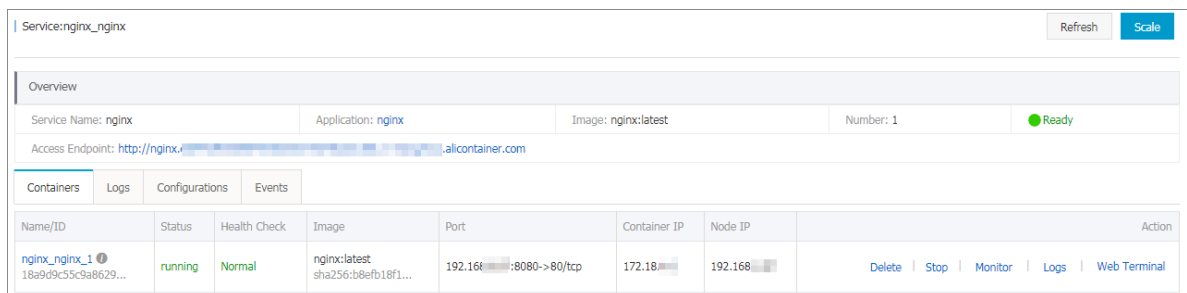
To expose multiple ports, click the plus icon and enter multiple host ports and container ports.



6. Click **Update**.

7. Click the service name (nginx in this example).

The host port maps to the container port and the port connection using Telnet is successful.



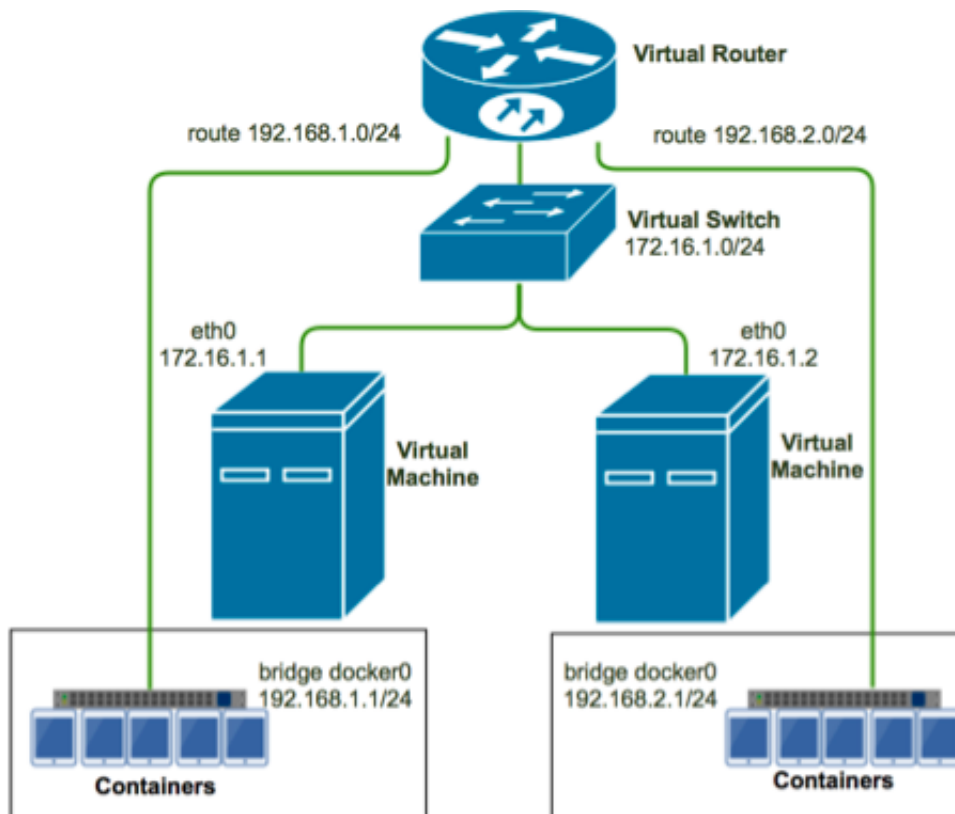
2.5 How do Container Service network models realize cross-host container communication?

Interworking between containers

Container Service provides an independent IP address that is reachable within the cluster for each container in the cluster. Containers can communicate with each other by using the independent IP addresses instead of being exposed to the host port by means of Network Address Translation (NAT). Therefore, the dependency on the IP address of the host is removed, avoiding port conflict issues among multiple containers when configuring NAT. The following section describes how to realize cross-host container communication under different network models.

In a Virtual Private Cloud (VPC):

VPC helps you build an isolated network environment based on Alibaba Cloud. You can have full control over your own virtual network, including a free IP address range, Classless Inter-Domain Routing (CIDR) block division, and the configurations of route table and gateway. By configuring the VPC route table, Container Service forwards inter-container access requests to the Elastic Compute Service (ECS) instances corresponding to the container IP address range. See as follows.



Start Docker daemon on a cluster node (172.16.1.1) and set the default IP address range of the bridge network to 192.168.1.0/24. Start Docker daemon on another node (172.16.1.2) and set the default IP address range of the bridge network to 192.168.2.0/24. Set the corresponding routing rule in the VRouter route table under the VPC to forward access requests from 192.168.1.0/24 to the node 172.16.1.1. Set a similar routing rule for the other node.

Then, if a container with the IP address 192.168.1.2 on node 1 accesses a container with the IP address 192.168.2.2 on node 2, the access request is forwarded by means of the route table to a corresponding machine. The access request is then forwarded to the bridge of Docker0 according to the routing rule created by Docker. Finally, the request is forwarded to the container with the IP address 192.168.2.2.

Besides, Container Service assigns independent CIDR blocks and route entries for containers in the VPC. This helps avoid conflicting with original VSwitch CIDR block, route table entries, and IP route table on the machine. Otherwise, the access request might not be forwarded to the correct container.

In a classic network:

Docker 1.9 and later versions support a native [cross-host container network](#) based on the VXLAN protocol. In a classic network, Container Service creates a network environment for inter-container communication in one cluster based on Docker Overlay Network. The multi-host container network virtualized from the Docker Overlay Network is the same virtualized subnet, so containers can communicate with each other across hosts.

Cross-node link

In a multi-container application, [link](#) is often used to describe the dependency between containers. For example, WordPress Web service depends on the MySQL database service. Then, when a WordPress container is started, a series of parameters of the MySQL container, including the IP addresses and ports for database connection, can be obtained by using a link.

However, the Docker link only supports container connection on the same host node, while Container Service supports cross-node container connection. When the container IP address is changed, the container alias in the link is also changed. These actions are consistent with those on the link used on a single node.

Access from a container to a virtual machine

Containers in Container Service retain routes for external network access. Therefore, if a container needs to access the services or IP address of a virtual machine, the IP address or domain name of the virtual machine can be used directly.

References

- [Alibaba Cloud VPC service](#)
- [Get started with multi-host networking](#)
- [Understand Docker container networking](#)
- [Docker container links](#)

2.6 FAQs about routing service

Q: How to deploy an accessible Web container in Container Service in a simple and fast way?

A: See [Create an Nginx webserver from an image](#) and [Create WordPress with an orchestration template](#) for how to deploy an application in a simple and fast way.

Q: How to use the routing service?

A : See [Simple routing - supports HTTP and HTTPS](#) .

Q: How to use the routing label?

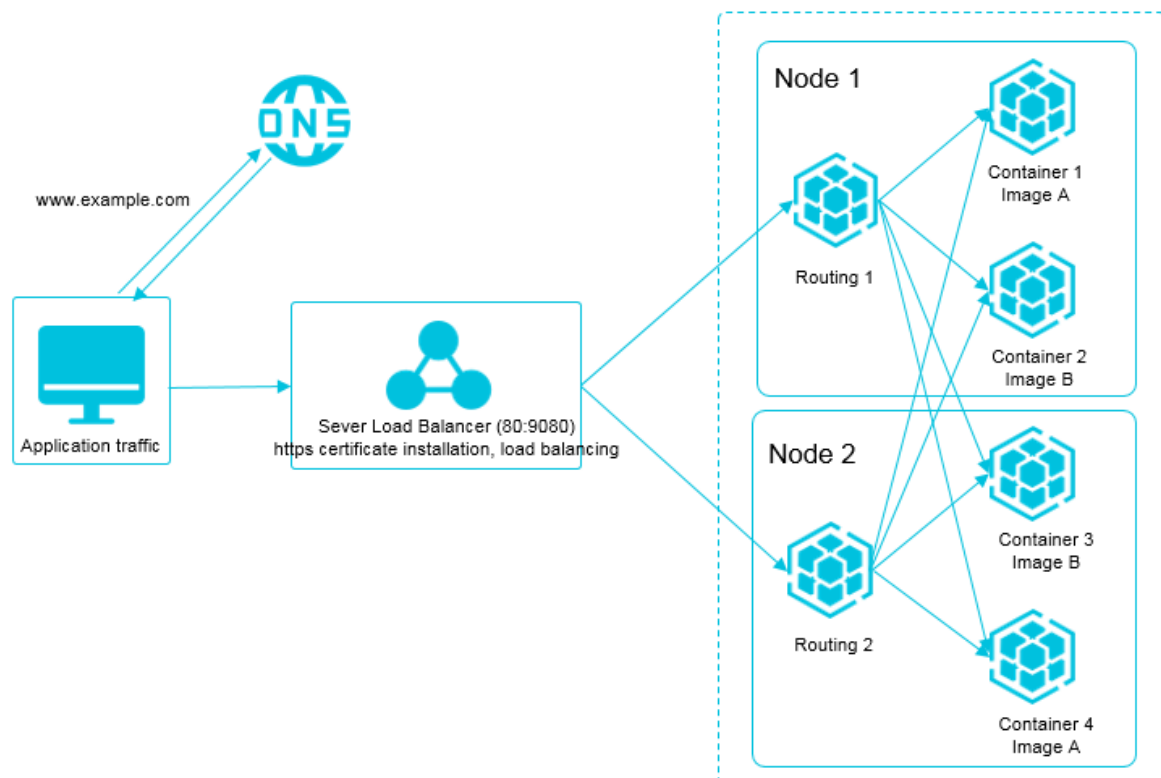
A: See the service orchestration document [Label description](#).

Q: How to access the contents of a deployed Web container?

A: A container is a process or a group of processes. You can access contents in a container only when the process exposes a port. The container abstracts a container port and maps the container port to the host port. You can access the container through the host port that is mapped to the container port.

Q: For high availability purpose, only one access endpoint is provided for multiple containers with the same functions. How is this implemented in Container Service?

A: As shown in the following figure, a routing service is provided, which can be configured by **Service > Update > Web Routing**. By default, the routing service deploys a routing container on each node in the cluster (the routing container is displayed in the container list after the cluster is created and belongs to the acsrouting application). A Server Load Balancer instance is created by default after a cluster is created. All access requests go through the frontend port 80 of the cluster Server Load Balancer > the node 9080 port > the port 80 of the routing container. The underlying implementation of the routing container is an HAProxy Server Load Balancer software, which is similar to Nginx, and provides the Server Load Balancer function. The routing container forwards the access requests to different container backends based on the domain names specified by the "HOST" header in HTTP (containers in the same cluster are interworking). During routing configuration, pay attention to the differences and relations among the Server Load Balancer ports, the node virtual machine (VM) ports, and the container ports.



Q: How to add a domain name to a service exposed to the public network and enable the service to support HTTP?

A: See [Simple routing - configure domain names](#).

Q: How to change the protocol from HTTP to HTTPS?

A: See [Simple routing - change HTTP to HTTPS](#).

Q: How to redirect an HTTP request to an HTTPS request?

A: Distinguish whether the request protocol is HTTP or HTTPS. If the protocol is HTTP, the request is redirected to HTTPS by 301 or 302. However, when the HTTP request and HTTPS request are forwarded to the same backend port (for example, port 9080) by using the ports 80 and 443 of Server Load Balancer respectively, the backend container cannot distinguish whether the request protocol is HTTP or HTTPS. To solve this issue, expose an additional port (for example, port 8080) to make the HTTP request go through the following link: the frontend port 80 of the Server Load Balancer > port 8080 of the node host VM > a container dedicated for redirect, for example, the Nginx port 80 > redirected to HTTPS by 302.

Q: Why are exceptions reported on the HTTP port of Server Load Balancer?

A: Exceptions are reported when the health check on the HTTP port of Server Load Balancer fails. The health check principle is to send an HTTP HEAD request, which is similar to a GET request, but only the response header needs to be returned. The domain name needs to be configured for the HTTP request, and the default value is the IP address. Server Load Balancer considers the health check as successful when the request returns the status code 200. Bypass Server Load Balancer and check whether the request returns the status code 200 by running the [curl](#) command directly on your host node.

Q: Why are exceptions reported on the HTTPS port of Server Load Balancer?

A: Exceptions are reported when the health check on the HTTPS port of Server Load Balancer fails. The health check principle is to send an HTTP HEAD request, which is similar to a GET request, but only the response header needs to be returned. The domain name needs to be configured for the HTTP request, and the default value is the IP address. Server Load Balancer considers the health check as successful when the request returns the status code 200. For an HTTPS port, the request domain name must be configured when you configure the health check. Otherwise, the health check fails by default (the request of the default IP address is forwarded to the routing container, but the routing container does not know to which backend the request is to be forwarded, and error 503 is returned). Bypass Server Load Balancer and check whether the request returns the status code 200 by running the [curl](#) command directly on your host node. Check the application validity if the status code 200 is not returned.

Q: Do clusters support binding or unbinding intranet Server Load Balancer instances?

A: The cluster can bind at most one Server Load Balancer instance and the Server Load Balancer instance can be unbound from the cluster. For more information, see [Bind and unbind a Server Load Balancer instance](#).

Q: Do clusters support binding multiple cluster-level Server Load Balancer instances?

A: Not supported currently. You can manually create a Server Load Balancer instance and then bind it to port 9080 of the cluster node. When the node is expanded, you must maintain the backend server of your created Server Load Balancer instance on your own, for example, increasing or reducing the number of backend servers.

Q: How do containers communicate with each other in a cluster?

A: The container name can be used as the internal domain name when a container accesses another container in the same cluster.

Q: How are service discovery and Server Load Balancer between containers in the same cluster implemented?

A: The routing service proxy is used for the forwarding and discovery. See [Routing and Server Load Balancer between services in a cluster](#).

Q: How to troubleshoot problems about routing service access links?

A: See [How to troubleshoot access link issues?](#).

Q: The Web routing rule can be set by using the `aliyun.routing.port_${container_port}` in the orchestration template or by updating the service configurations. What is the difference between these two methods?

A: The two methods are essentially the same. The Web routing rule set by using the `aliyun.routing.port_${container_port}` in the orchestration template can be reflected in the Web routing rule on the **Update Service** page. However, the Web routing rule set by updating the service configurations cannot be reflected in the orchestration template. Configuring the Web routing rule by **Update Service** facilitates you to operate in the console, troubleshoot the issues, and check the errors. This Web routing rule configuration form is converted into a label of the orchestration template and then used to update the service configurations.

Q: What if the default routing service does not meet corner cases?

A : A custom proxy image registry.aliyuncs.com/acs/proxy can be a good solution. The image is based on HAProxy and supports parameter configurations that define HAProxy. This image also supports the dynamic service discovery, that is, the service is routed to a healthy container based on the service health status.

Q: How to obtain the real IP address of the client after using the simple routing?

A: For all the requests that use simple routing, Container Service adds x-forwarded-for information in the request headers.

```
x-forwarded-for: <Client IP address>  
x-forwarded-for: <Proxy server IP>
```



Note:

The header may contain multiple lines. The real IP address of the client can be obtained from the x-forwarded-for of the first line.

2.7 FAQs about custom Server Load Balancer

Q: What are the scenarios of custom Server Load Balancer?

A: The custom Server Load Balancer can be used in the following scenarios:

- In Layer-7 protocol Server Load Balancer, a route is customized for each service. Services of non-container clusters access the services of containers in container clusters when a traditional architecture is migrated to a container architecture.
- In Layer-4 protocol Server Load Balancer, a route is customized for each service. Services of non-container clusters access the services of containers in container clusters when a traditional architecture is migrated to a container architecture.
- Intranet Server Load Balancer instances are used for communication in Container Service.

Q: How to use custom Server Load Balancer?

A: See [Server Load Balancer routing](#).

Q: How to use the labels of custom Server Load Balancer?

A: See **lb** in the service orchestration document [Label description](#).

Q: How to configure ECS to support Server Load Balancer?

A: In principle, no special configuration is required for the Elastic Compute Service (ECS) instances added to the Server Load Balancer instance backend. For the ECS instances on Linux that associate with the Layer-4 protocol (TCP) Server Load Balancer, if you cannot access them normally, make sure that the values of the following three parameters in the system configuration file `/etc/sysctl.conf` are zero:

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.eth0.rp_filter = 0
```

If the ECS instances deployed in the same intranet segment cannot communicate with each other, check whether or not the following parameters are set correctly:

```
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.all.arp_announce = 2
```

Run the `sysctl -p` command to update the parameter settings.

Q: What are the benefits of custom Server Load Balancer?

A: The custom Server Load Balancer can automatically remove routes of the backend containers that are not running when you are updating the service configurations, and when the container is stopped or fails to be deployed. You must maintain the other settings of Server Load Balancer.

Q: What are the limits for custom Server Load Balancer?

A: Currently, the limits for custom Server Load Balancer are as follows:

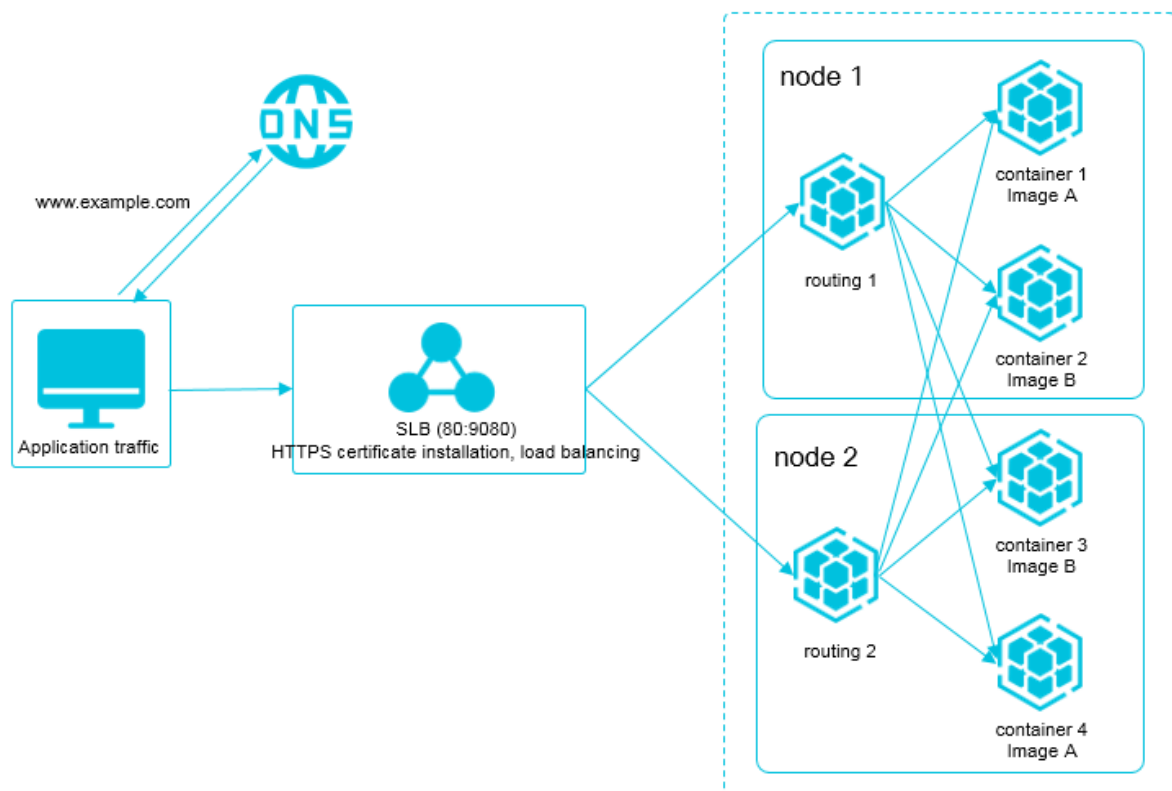
- Create a Server Load Balancer instance, name it, and create the corresponding listening port. Then, provide the Server Load Balancer instance name `$slb_name` or `$slb_id`, the port to be exposed, the used protocol `$scheme` (possible values include `tcp`, `http`, `https`, and `udp`), the mapped container port `$container_port` by using extension labels, and specify the frontend port `$front_port` of the Server Load Balancer instance.
- You must specify the host:container port mapping of the service port to be exposed and then use the standard Dockerfile label `ports` to specify the port mapping. You must specify the host port and this port cannot conflict with the host ports mapped by other services. Server Load Balancer uses the host port to bind the backend ECS instance.
- A service can only use one or more Server Load Balancer instances to expose the service port. Services cannot share the same Server Load Balancer instance because they are distributed in different ECS instance backends.
- The host that has the service with Server Load Balancer NAT mapping deployed uses the same host:container port mapping. Therefore, these services only have one instance on each ECS instance.
- The supported Server Load Balancer protocol `$scheme` includes `tcp`, `http`, `https`, and `udp`.
- Create a listening port on your own in the Alibaba Cloud Server Load Balancer console.
- Log on to the Server Load Balancer console to modify the configurations for the Server Load Balancer instance used in Container Service, such as bandwidth limitation, on your own.
- The value of the `lb` label is that the backend is bound automatically after you configure the corresponding labels, without binding the backend ECS instance of Server Load Balancer by yourself. Therefore, except for binding the Server Load Balancer backend, set and modify the Server Load Balancer instances on your own in the Alibaba Cloud Server Load Balancer console.
- Container Service helps you generate a Resource Access Management (RAM) user (you must activate RAM). This account has some Server Load Balancer permissions, but does not have the permission to create or delete Server Load Balancer instances. Use this account to help

you manage the Server Load Balancer instances used in Container Service, for example, binding some nodes in the cluster as the service backend.

2.8 How to troubleshoot access link issues?

Context

When a web container is set up in Container Service and routing is used to forward requests to this server, the request link is as follows: client > DNS resolution > Server Load Balancer VIP > an acsrouting container in the cluster > forwarded to the web container. This is shown in the following figure.

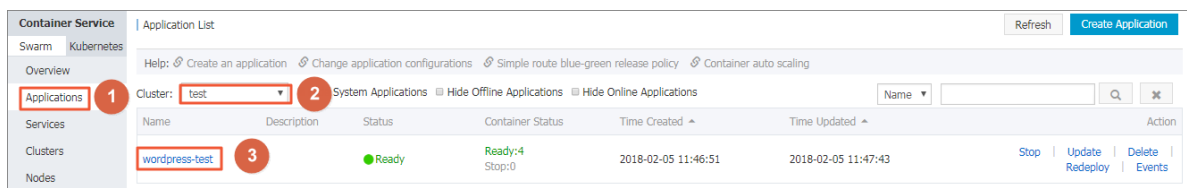


If a problem occurs at any stage in this process, user requests may not be correctly routed to the web container. Troubleshoot the access link issues as follows, starting from the health checks of the developers' web containers, where issues are always located.

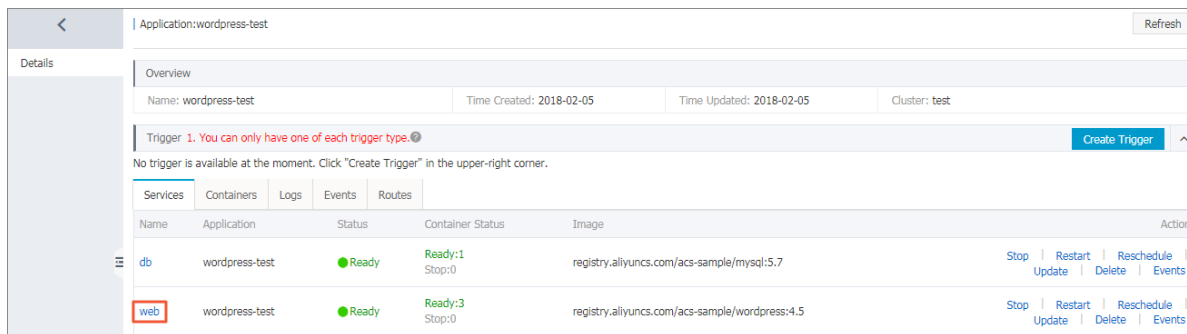
Procedure

1. Check whether or not the container is running.

Log on to the [Container Service console](#). Click **Applications** in the left-side navigation pane. Select the cluster from the Cluster list. Click the application name (wordpress-test in this example).

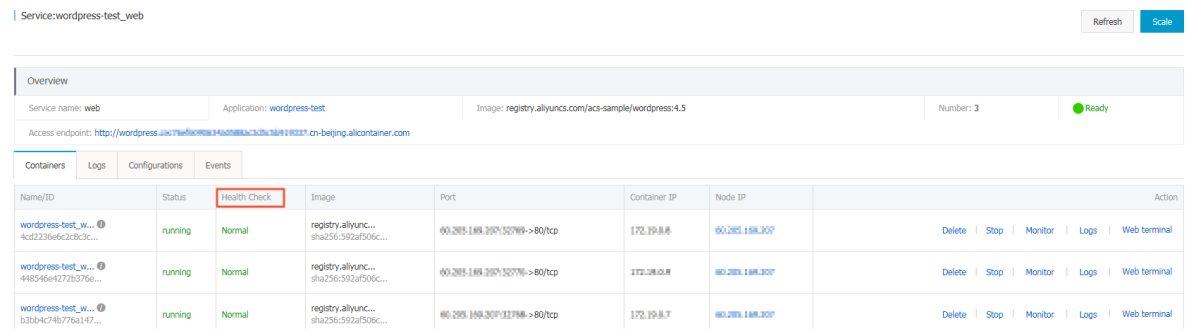


2. Click the name of the service (web in this example) that provides the web container.



3. Check the health check status of the container that provides the web service.

Under the **Containers** tab, check whether or not all of the containers have **Normal** displayed for **Health Check**. If not, click the **Logs** tab to check the error message and click the **Events** tab to check if any deployment exception occurs. If the **health check** is set for the application, you must confirm that the health check page returns the status code 200 to make sure the health check status is normal. See the following figure.



4. Check whether or not the web container page responds normally.

If the health check status of the container is normal, you must bypass the routing service and check the accessibility of the web container directly. As shown in the preceding figure, you can view the container IP of a web container. Log on to the routing container of a machine in the cluster and use the container IP to request the web container page. If the returned HTTP status code is less than 400, the web container page is normal. In the following example, `docker`

`exec -it f171110f2fe2 sh` is used. Here, `f171110f2fe2` is the container ID of the container `acsrouting_routing_1` and the IP address `172.19.0.7` in `curl -v 172.19.0.7`

is the container IP address of a web service. The request then returns the status code 302, indicating that the web container can be accessed normally.

```
root@c68a460635b8c405e83c052b7c2057c7b-node2:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b403ea045fa1 registry.aliyuncs.com/acs-sample/wordpress:4.5 "/
entrypoint.sh apach" 13 seconds ago Up 11 seconds 0.0.0.0:32768->80/
tcp w_web_2
025f7967cec3 registry.aliyuncs.com/acs-sample/mysql:5.7 "/
entrypoint.sh mysql" About a minute ago Up About a minute 3306/tcp
w_db_1
2f247b8a76e5 registry.aliyuncs.com/acs/ilogtail:0.9.9 "/bin/sh -c '
sh /usr/" 31 minutes ago Up 31 minutes acslogging_logtail_1
42b75bee6cd8 registry.aliyuncs.com/acs/monitoring-agent:latest "acs
-mon-run.sh --hel" 31 minutes ago Up 31 minutes acsmonitoring_acs-
monitoring-agent_2
0a9afa527f03 registry.aliyuncs.com/acs/volume-driver:0.7-252cb09
"acs-agent volume_exe" 31 minutes ago Up 31 minutes acsvolumed
river_volumedriver_2
3c1440fd114c registry.aliyuncs.com/acs/logspout:0.1-41e0e21 "/bin/
logspout" 32 minutes ago Up 32 minutes acslogging_logspout_1
f171110f2fe2 registry.aliyuncs.com/acs/routing:0.7-staging "/opt/
run.sh" 32 minutes ago Up 32 minutes 127.0.0.1:1936->1936/tcp, 0.0.0
.0:9080->80/tcp acsrouting_routing_1
0bdeb8464c14 registry.aliyuncs.com/acs/agent:0.7-bfe8bdf "acs-agent
join --nod" 33 minutes ago Up 33 minutes acs-agent
ba32a0e9e7fe registry.aliyuncs.com/acs/tunnel-agent:0.21 "/acs/
agent -config=c" 33 minutes ago Up 33 minutes tunnel-agent
root@c68a460635b8c405e83c052b7c2057c7b-node2:~# docker exec -it
f171110f2fe2 sh
/ # curl -v 172.19.0.7
* Rebuilt URL to: 172.19.0.7/
* Trying 172.19.0.7...
* Connected to 172.19.0.7 (172.19.0.7) port 80 (#0)
> GET / HTTP/1.1
> Host: 172.19.0.7
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 302 Found
< Date: Mon, 09 May 2016 03:19:47 GMT
< Server: Apache/2.4.10 (Debian) PHP/5.6.21
< X-Powered-By: PHP/5.6.21
< Expires: Wed, 11 Jan 1984 05:00:00 GMT
< Cache-Control: no-cache, must-revalidate, max-age=0
< Pragma: no-cache
< Location: http://172.19.0.7/wp-admin/install.php
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 172.19.0.7 left intact
```

5. Verify the validity of acsrouting.

[Upgrade routing](#) to the latest version. Log on to each machine in the cluster (any machine might receive requests, no matter on which machine the application container is deployed), and request the routing health check page.

```
root@c68a460635b8c405e83c052b7c2057c7b-node2:~# curl -Ss -u admin:admin 'http://127.0.0.1:1936/haproxy?stats' &> test.html
```

Copy the page `test.html` to a machine with a browser and use the browser to open the local file `test.html`. Check the corresponding web service and container backend. The first part is the stats information, providing routing statistics. The second part is the frontend statistics. The third part, which provides backend information, is essential to view. Here, **w_web_80_servers** indicates the information for the port 80 backend servers of the service web under the application w. In total, three backend servers exist, namely, the backend has three containers that provide web service. Green indicates that the routing container can connect to the three containers and the system works properly. Any other color indicates an exception.

file:///Users/tanlin/na/test.html

HAProxy

Statistics Report for pid 33

> General process information

active UP

active UP going down

active DOWN, going up

active or backup DOWN

active or backup DOWN for maintenance (MAINT)

active or backup SOFT STOPPED for maintenance

backup UP

backup UP going down

backup DOWN, going up

not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

Scope:

Hide DOWN servers

Refresh now

CSV export

External resources:

Primary site

Updates (v1.9)

Online manual

stats

	Queue			Session rate			Sessions			LbTot	Last	Bytes			Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrle
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit			In	Out	Req												
Frontend	1	1	-	1	1	1	2 000	1				0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Backend	0	0		0	0		0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

name_resolver_http

	Queue			Session rate			Sessions			LbTot	Last	Bytes			Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrle
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit			In	Out	Req												
Frontend	1	1	-	0	1	1	2 000	145				15 805	21 170	0	0	0	0	0	0	0	0	0	0	0	0	

w_web_80_servers

	Queue			Session rate			Sessions			LbTot	Last	Bytes			Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrle
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit			In	Out	Req												
server_539e28216ef533707c34ce9e1aa724da18cb80c2230b6048a13d54795a8308_80	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
server_95f0501a9b6437c7d73c482e2af4e7787e9e27c7900860efa94aa63f1cbe_80	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
server_b403ea045fa1dc2a2c64ea9d258da2542e4c743fab9b40c3d7e7d4568bcb66_80	0	0	-	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Backend	0	0		0	0		0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

6. Check whether or not the Server Load Balancer VIP is forwarding data correctly and the health check status is normal.

a) Find the Server Load Balancer VIP of the cluster. Click **Clusters** in the left-side navigation pane in the [Container Service console](#).

Container Service | Cluster List

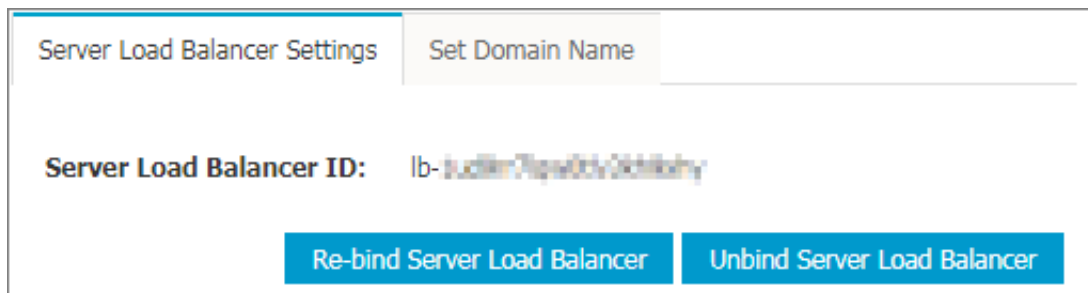
You can create up to 5 clusters and can add up to 20 nodes in each cluster. Refresh Create Cluster

Help: Create cluster How to add existing ECS instances Cross-zone node management Log Service integration Connect to cluster through Docker Client

Cluster Name/ID	Cluster Type	Region	Network Type	Cluster Status	Node Status	Number of Nodes	Time Created	Docker Version	Action
test	Alibaba Cloud Cluster	China East 1 (Hangzhou)	VPC	Running	Healthy	2	2018-02-05 09:44:57	17.06.2-ce	Manage View Logs Delete Monitor More

b) Click **Manage** at the right of the cluster (test in this example). Click **Load Balancer Settings** in the left-side navigation pane. View and copy the Server Load Balancer ID. Click Products

> Server Load Balancer to go to the [Server Load Balancer console](#). Click **Manage** at the right of the Server Load Balancer instance to enter the instance details page.



c) View the IP address of the Server Load Balancer instance.

Basic Information		^
Server Load Balancer ID: lb-5ucll8m7p9u00k0z0cl0u0u0y	Status: Running	
Server Load Balancer Name: lb-5ucll8m7p9u00k0z0cl0u0u0y	Region: China East 1 (Hangzhou)	
Instance IP Type: Public IP	Zone: cn-hangzhou-f(Master)/cn-hangzhou-e(Slave)	
Network Type: Classic Network		
Billing Information		^
Billing Method: Pay by Traffic	Created At: 2018-01-03 18:10:55	Billing Details Release
Instance IP Address: 47.85.118.20(Public IP)	Automatic Release Time: -	

d) View the health status of the Server Load Balancer port. Click **Listeners** in the left-side navigation pane. The **Running status** indicates the port works properly.

Instance Details

Listeners

Add Listener

Refresh

Listeners

Servers

Backend Servers

VServer Groups

Master-Slave Se...

Monitor

Front-end Protocol/Port

Backend Protocol/Port

Status

Forwarding Rules

Session Persistence

Health Check

Peak Bandwidth

Server Group

Actions

TCP: 80

TCP: 9080

Normal

Weighted Round Robin

Disable

Enable

No Limits

-

Configure

Details

More

Start

Stop

Delete

e) Check the status of the backend servers mounted to Server Load Balancer. Click **Servers** > **Backend Servers** in the left-side navigation pane. Make sure the **Health Check** status is **Normal**.

Instance Details		Load Balancer Server Pool Region: China East 1 (Hangzhou) Zone: cn-hangzhou-e (Master) /cn-hangzhou-f (Slave)							
Listeners									
Servers									
Backend Servers									
VServer Groups									
Master-Slave Se...									
Monitor									
		Servers Added		Servers Not Added					
		Enter the name of the ECS instance				Search			
						Refresh			
		ECS Instance ID/Name	Zone	Public/Internal IP Address	Status(All)	Network Type(All)	Health Check	Weight	Actions
		hpc-lb-5ucll8m7p9u00k0z0cl0u0u0y	cn-hangzhou-e	47.86.178.40 (EIP) 202.38.118.124 (Private)	Running	VPC (vpc-b0t4p3t3p9b0c1ggk9l0u)	Normal	100	Remove
		hpc-lb-5ucll8m7p9u00k0z0cl0u0u0y	cn-hangzhou-e	47.86.180.240 (EIP) 202.38.118.123 (Private)	Running	VPC (vpc-b0t4p3t3p9b0c1ggk9l0u)	Normal	100	Remove

7. Check whether or not the domain name is correctly resolved to the Server Load Balancer VIP. For example, use the `ping` or `dig` command to view the resolution result. The domain name

must be resolved and directed to the Server Load Balancer VIP address found in the previous step .

```
$ ping www.example-domain.com
```

```
$ dig www.example-domain.com
```

2.9 How does Container Service isolate containers of different users?

Container Service produces and manages Elastic Compute Service (ECS) instances by means of your authorization. Your containers can only run on the ECS instances that you own.

If the cluster is of the classic network, the access between clusters of different users is isolated by security groups.

If the network type of the cluster is Virtual Private Cloud (VPC), the access between clusters of different users is isolated by VPC.

You can customize the security groups or VPC access permissions of the clusters that you own.

2.10 "Invalid input for user ram ak or ak secret" is displayed when you create an application and add custom Server Load Balancer instances

Procedure

1. Check whether the RAM service is activated or not. If yes, go to step 2. If not, activate the service and try again.
2. Check whether the number of RAM accounts has reached its upper limit or not. If yes, delete an account and try again.
3. If neither of the preceding problems exists, update your RAM authorization information. (Log on to the [Container Service console](#). Click **Clusters** in the left-side navigation pane. Click **More** at the right of the cluster. Select **Update RAM Authorization Information** from the drop-down list. Click **Confirm** in the displayed dialog box.)

2.11 FAQs about changing application configurations

By default, Container Service will restart or recreate the container on the current machine when you change the application configurations. This is to make sure the local data volumes of the service container on the current machine are not lost. Therefore, if you specify to schedule the

container to another machine when changing the configurations, Container Service will ignore your scheduling settings.

If you are sure the service has no local data volumes or the container data in local data volumes can be lost, turn on the **Force Reschedule** switch. Then, Container Service will schedule the container to another machine according to the scheduling settings in the **Template**.

**Note:**

Turning on the Force Reschedule switch to schedule the container to another machine will cause the container data in the local data volumes on the current machine to become lost. So proceed with caution.

Example

Assume that your container is deployed on node1.

You specify to schedule the container to node2 (`constraint:aliyun.node_index==2`) when changing the application configurations as follows:

```
web:
  image: 'nginx:latest'
  restart: always
  environment:
    - 'constraint:aliyun.node_index==2'
  ports:
    - 80
  labels:
    aliyun.scale: 1
```

In this situation:

- If the **Force Reschedule** switch is turned off, Container Service will ignore your scheduling settings and still deploys the container on node1.
- If the **Force Reschedule** switch is turned on, Container Service will schedule the container to node2. The container data in local data volumes on node1 will be lost.

2.12 Common errors leading to cluster creation failure

The cluster creation in Container Service might fail because of some errors. Some common errors and solutions are as follows for your reference:

- **Server Load Balancer error. (You can provide the RequestID to the support staff.)**

```
Config cs AcessRouting failed : Failed to CreateLoadBalancer:Aliyun
API Error: RequestId: 47AE1BFC-AFEA-469C-82F5-1E3BD81897F2 Status
```

```
Code: 500 Code: InternalError Message: The request processing has failed due to some unknown error, exception or failure.
```

- **Docker configuration timeout. (Check the daemon logs.)**

```
Adding tags map[provider:aliyunacs acsversion:1.0 acsclusterid:xxx acsclustername:xxx] to instance xx
```

- **Zones are currently not on sale.**

```
Code: Zone.NotOnSale Message: The specified zone is not on sale.
```

- **Failed to create the cluster master.**

```
Creating Master Region Controller: 500 Internal Server Error
```

- **The security group exceeds the quota.**

```
Config cs ClusterNetwork failed : Aliyun API Error: RequestId : 264E5ADC-0571-44C0-8508-C037096856C7 Status Code: 403 Code: QuotaExceed.SecurityGroup Message: The security group quota exceeds.
```

- **The resource is insufficient.**

```
Failed to create instance: Aliyun API Error: RequestId: CC2FF296-D29E-484E-B095-8905CDA016BA Status Code: 403 Code: OperationDenied Message: The resource is out of usage.
```

- **The number of Server Load Balancer instances exceeds the quota.**

```
Config cs AccessRouting failed : Failed to CreateLoadBalancer:Aliyun API Error: RequestId: A1E5D644-C31A-4142-B722-CBD6EF57A3A7 Status Code: 400 Code: OverQuota Message: The Total is over the quota.
```

- **The number of shared images exceeds the quota.**

```
Fail shared image: Aliyun API Error: RequestId: 693F8B6C-9349-4277-B109-9841BFF1F76C Status Code: 404 Code: InvalidAccount.Forbidden Message: The specified Account does not yourself.
```

- **The Elastic IP (EIP) exceeds the quota.**

```
QuotaExceeded.Eip Message: Elastic IP address quota exceeded
```

- **The Pay-As-You-Go Elastic Compute Service (ECS) instances exceed the quota.**

```
QuotaExceed.AfterpayInstance Message: Living afterpay instances quota exceeded
```

- **Parameters are missing (basically when calling API).**

```
c7904aa1b9e3642fa8fbf440f48dfedb8 | Fail shared image: Aliyun API Error: RequestId: D2518C32-0C2B-4EAD-9F88-43A5C2AAF0C1 Status Code:
```

```
404 Code: InvalidAccount.NotFound Message: The specified parameter "
AddAccount.n" or "RemoveAccount.n" does not exist.
```

- **Mismatched instance type. (This error is mainly from the API users.)**

```
InvalidInstanceType.ValueUnauthorized Message: The specified
InstanceType is not authorized
```

- **The container ClasslessInter-Domain Routing (CIDR) block configured in the Virtual Private Cloud (VPC) environment conflicts with the current route table CIDR block or the CIDR block of the VSwitch under VPC.**

```
InvalidCIDRBlock.Duplicate Message: Specified CIDR block is already
exists
```

2.13 Configure DNS options in containers and optimize DNS resolution

Configure DNS options in a container

You can specify `dns` and `dns_options` in the orchestration template of Container Service to specify the DNS server and DNS options for the container.

For example:

```
testdns:
  image: nginx
  dns:

  dns_options:
    - use-vc
    - no-tld-query
```

The preceding example configures the DNS server and DNS query option for the service container



Note:

Docker embeds a DNS server in each container for service discovery. The DNS server in the `/etc/resolv.conf` file of the container is the built-in DNS server 127.0.0.11 for Docker. Docker listens to DNS requests of built-in servers and forwards the DNS requests to the server configured by `dns`.

Optimize DNS resolution

When requesting a domain name, the DNS resolution might time out or fail, which causes the website to become inaccessible. The operating system generally enables the `nscd` service as the

DNS cache to avoid DNS resolution failure. However, the `nsd` service is generally not configured in container images. You can install the `nsd` service on the container that you often perform DNS resolution to optimize DNS resolution in the container.

Install the `nsd` software package. Then, when the container is started, start the `nsd` service first and then start your processes.

```
FROM registry.aliyuncs.com/acs/ubuntu
RUN apt-get update && apt-get install -y nsd && rm -rf /var/lib/apt/
lists/*
CMD service nsd start; bash
```

2.14 How to increase data disks for Container Service Docker?

Docker data is stored on disks by using the union file system. If the number of containers or images needing to be run on the machines is continuously increased, the disk size may not meet the requirements. In this situation, increase the data disks to expand the storage space for Docker data directory.

Docker data directory

For Docker, the container and image data is stored in the `/var/lib/docker` directory by default. You can check the currently occupied disk size of this directory by running the `du` command. For example:

```
du -h --max-depth=0 /var/lib/docker
7.9G /var/lib/docker
```

Change Docker data disk

Many Docker images are big. Therefore, several images might occupy large disk space, which leads to insufficient disk space. By increasing the data disks for the Docker data directory, the requirements of increasing images or containers continuously can be met.

Purchase ECS data disk and mount to machines needing expansion

1. Log on to the [Elastic Compute Service \(ECS\) console](#) to purchase the cloud disk with corresponding configurations.
2. Click **Instances** in the left-side navigation pane. Select the region and then click the instance name or click **Manage** at the right of the instance. > Click **Instance Disks** in the left-side navigation pane. > Click **Attach Disk** in the upper-right corner. Select the purchased disk and record the mount point `/dev/xvd*` or `/dev/vd*`. Determine the mount point by running the `cd` command. The mount point of the I/O optimized instance is `/dev/vd*`.

Log on to the machine and format the mounted disk

1. Run `ls -l /dev/xvd*` or `ls -l /dev/vd*` on the machine to view the disk ID, which is consistent with your recorded one.
2. Partition the disk by running the `fdisk` command. Then, format the disk by using `mkfs.ext4`.

For more information, see [Format and mount data disks for Linux instances](#). For example:

```
root@izbp16hlijt5er5wempg4sZ:~# ls -l /dev/vd*
brw-rw---- 1 root disk 253, 0 Jan 5 17:44 /dev/vda
brw-rw---- 1 root disk 253, 1 Jan 5 17:44 /dev/vda1
brw-rw---- 1 root disk 253, 16 Jan 5 17:55 /dev/vdb
root@izbp16hlijt5er5wempg4sZ:~# fdisk -S 56 /dev/vdb
Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x44e128c4.
Command (m for help): n
Partition type
   p primary (0 primary, 0 extended, 4 free)
   e extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-41943039, default
41943039):
Created a new partition 1 of type 'Linux' and of size 20 GiB.
Command (m for help): wq
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
root@izbp16hlijt5er5wempg4sZ:~# ll /dev/vd*
brw-rw---- 1 root disk 253, 0 Jan 5 17:44 /dev/vda
brw-rw---- 1 root disk 253, 1 Jan 5 17:44 /dev/vda1
brw-rw---- 1 root disk 253, 16 Jan 5 17:58 /dev/vdb
brw-rw---- 1 root disk 253, 17 Jan 5 17:58 /dev/vdb1 ##Add
partition
root@izbp16hlijt5er5wempg4sZ:~# mkfs.ext4 /dev/vdb1 ##Format
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 5242624 4k blocks and 1310720 inodes
Filesystem UUID: cef1625c-7533-4308-bc44-511580e3edc8
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736,
1605632, 2654208,
    4096000
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Move Docker data to a new disk

1. Stop Docker daemon first to guarantee the data integrity in the process of moving Docker data.
Run the command `service docker stop` to stop the Docker daemon.

2. Move the data in the Docker default data directory to a backup directory. For example, if the backup directory is `/var/lib/docker_data`, run the command `mv /var/lib/docker /var/lib/docker_data`.
3. Mount the newly formatted disk to the `/var/lib/docker` directory. Run the command `echo "/dev/vdb1 /var/lib/docker ext4 defaults 0 0" >>/etc/fstab && mkdir /var/lib/docker && mount -a`.
4. Move the backed up Docker data to the new disk. Run the command `mv /var/lib/docker_data/* /var/lib/docker/`.

Start Docker daemon and check data location

1. Start Docker daemon and run the command `service docker start`.
2. Run the command `df`. You can see `/var/lib/docker` is mounted to the new disk.

```
root@izbp16hlijt5er5wempg4sZ:~# df -h
Filesystem Size Used Avail Use% Mounted on
udev      2.0G  0  2.0G  0% /dev
tmpfs     396M  7.1M  389M  2% /run
/dev/vda1  40G  2.7G  35G  8% /
tmpfs     2.0G  476K  2.0G  1% /dev/shm
tmpfs     5.0M  0  5.0M  0% /run/lock
tmpfs     2.0G  0  2.0G  0% /sys/fs/cgroup
tmpfs     396M  0  396M  0% /run/user/0
/dev/vdb1  20G  2.1G  17G  12% /var/lib/docker ##This directory is
mounted to the new disk.
```

3. Run the command `docker ps` to check if containers are lost. Restart the related containers as required, for example, the containers without configuring the `restart:always` label.

2.15 How to troubleshoot log issues?

If the extension label `label aliyun.log_store_xxx: xxx` is added in the application but no logs are collected to Log Service, follow these steps to troubleshoot the issue.



Note:

Troubleshoot the issue by following these steps and do not skip the steps.

1. Check whether or not Logstore is successfully created

The application is not successfully deployed if Logstore is not created. Check if any error message about deployment is in the application Events.

2. Check the ilogtail version

Run the command `docker ps | grep ilogtail` on the machine and determine the version of the ilogtail image according to the output. If the version is 0.11.6, upgrade the system services to the latest version (currently, the latest version is 0.13.4). After the upgrade, query the logs in the Log Service console after the application generates new logs.

3. Check the ilogtail logs

Run the command `docker exec -it <ilogtail container ID> cat /usr/local/ilogtail/ilogtail.LOG` and determine what the issue is according to the ilogtail logs.

Common possible reasons include:

- Network is not connected. Determine whether the network is connected or not by running the following command:

```
Virtual Private Cloud (VPC):
telnet logtail.cn-<region>-vpc.log.aliyuncs.com 80
Internet:
telnet logtail.cn-<region>.log.aliyuncs.com 80
```

- AccessKey is not configured.

Unauthorized ErrorMessage:no authority, denied by ACL appears in the logs if the primary account does not configure the AccessKey. Create the AccessKey for the primary account first. Check whether or not the primary account configures the AccessKey even if Unauthorized ErrorMessage:no authority, denied by ACL does not appear in the logs.

4. Check whether or not the machine IP is in the Log Service machine group

1. Log on to the [Log Service console](#).
2. Click the name of the Log Service project that corresponds to the cluster. The project naming rule is acslog-project-<first 10 letters of the cluster ID>.
3. Click **Logtail Machine Group** in the left-side navigation pane.
4. Click **Machine Status** at the right of the machine group and check if the IP address of the current machine is in the IP list.

5. Check whether or not the primary account configures the AccessKey

Make sure the primary account has at least one enabled AccessKey.

6. Check whether or not the log file has contents

Enter the business application container and determine whether or not logs are actually generated. For stdout logs, use the `docker logs` command directly.

2.16 Use Nginx + FPM in Container Service

To use Nginx + FPM in Container Service, we recommend that you use the image <https://github.com/ngineered/nginx-php-fpm> as the base image, which has both Nginx and FPM in one image.

This image can be used to create a container for Nginx and PHP-FPM. The created container can pull website codes from Git, and push or pull the code changes to or from Git. The container can also update the orchestration file by using the variables passed to Docker so as to update your codes and settings.

This image also supports Let's Encrypt SSL configurations, customizing Nginx configurations, modifying Nginx/PHP configurations, X-Forwarded-For headers, and UID mapping (support local data volumes).

2.17 Node exception

Container Service cannot connect to a node if the node status is **Exception**.

Reason analysis

Node exception occurs mainly because of your heavy node load, including the CPU usage, memory usage, network traffic, and I/O of the node.

Swarm clusters

You can view the monitoring data of your node either in the Container Service console or Alibaba Cloud CloudMonitor console.

- **View node monitoring data in Container Service console**

1. Log on to the [Container Service console](#).
2. Click Swarm > **Clusters** in the left-side navigation pane.
3. Click the cluster name.
4. Click **Monitor** at the right of the node that you want to view.

- **View node monitoring data in Alibaba Cloud CloudMonitor console**

1. Log on to the [CloudMonitor console](#).
2. Click **Cloud Service Monitoring** > **Container Service** in the left-side navigation pane.
3. Click **Node Monitoring** at the right of the cluster in which the node you want to view resides.
4. Click **Monitoring Charts** at the right of the node to view the monitoring data of this node.

**Note:**

To monitor the node load in real time, you can create alarm rules for the node. Click **Create Alarm Rule** in the upper-right corner of the page.

Kubernetes clusters

You can view the monitoring data of your node either in the Container Service console or in the Kubernetes application group.

- **View node monitoring data in Container Service console**

1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Clusters > > Nodes** in the left-side navigation pane.
3. Select the cluster from the Cluster drop-down list. Click **Monitor** at the right of the node that you want to view.

- **View node monitoring data in Kubernetes application group of CloudMonitor console**

1. Log on to the [Container Service console](#).
2. Click **Kubernetes > Clusters** in the left-side navigation pane.
3. Click **More > at the right of the cluster and then select Upgrade monitoring service**. Click OK in the displayed dialog box.
4. Log on to the [CloudMonitor console](#).
5. Click **Application Groups** in the left-side navigation pane.

Solutions

You can solve the node exception by:

- Reducing the number of containers deployed on the node.
- Restricting the resources used by the containers. See [Restrict container resources](#) for swarm clusters.
- Reducing the load to get the node back to normal.
- Expanding the node or cluster.
- Adding monitoring charts and creating alarm rules for the group resources in the cluster, which avoids the node from being overloaded.

2.18 FAQs about the operating system and kernel of Container Service

An error occurs when deleting or updating a container

The error that occurs when deleting or updating a container is similar to the following one:

```
failed to remove root filesystem for xxx: device or resource busy
```

Generally, this error occurs because the kernel version of the node where the container resides is low. Log on to the node where the container resides and run the command `uname -a` to view the kernel version. The error occurs if:

- The kernel version is equal to or earlier than 3.13 for Ubuntu 14.04.
- The kernel version is equal to or earlier than 3.10.0-514 for CentOS 7.

Solutions

You can upgrade the kernel of the node where the container resides to solve the issue.

1. Schedule the application from this node by using the scheduling constraint. For more information, see [Specified nodes scheduling](#).
2. Upgrade the kernel version of the node.

Ubuntu 14.04

```
apt-get update && apt-get install -y linux-generic-lts-xenial
```

CentOS 7

```
yum update -y kernel
```

3. Restart the node after upgrading the kernel to bring the new version of kernel into effect.
4. Schedule the application back to this node by using the scheduling constraint.

Whether or not to configure NTP synchronization for time in containers

The time on Linux is obtained by using the kernel interface, and the kernel is shared by containers on the same node. Therefore, time is consistent. Generally, NTP time synchronization is configured on nodes. No additional configurations for NTP synchronization are required in containers.

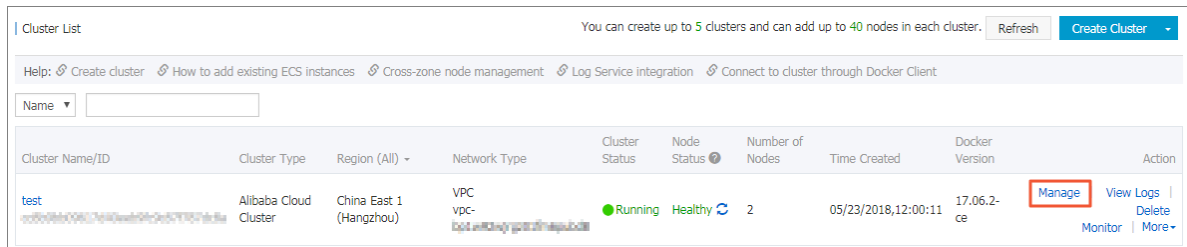
2.19 Failed to pull the image

Context

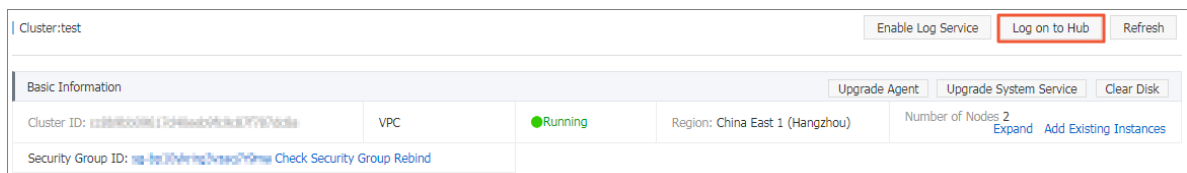
If you failed to pull the image, log on to the image repository again by following these steps:

Procedure

1. On the **Cluster List** page, click **Manage** at the right of the cluster in which the application is to be deployed.

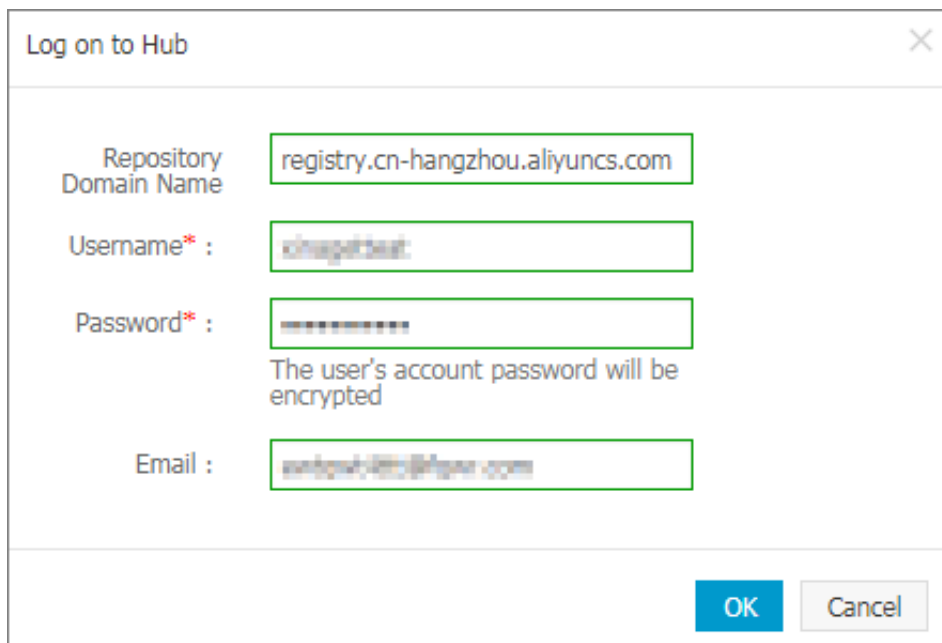


2. Click **Log on to Hub**.



3. In the displayed dialog box, enter the logon information and then click **OK**.

To use the Alibaba Cloud image repository, enter the domain name of the Alibaba Cloud image repository (for example, registry.cn-hangzhou.aliyuncs.com) in the **Repository Domain Name** field, your Alibaba Cloud username in the Username field, and the independent password used to log on to the repository in the Password field.



2.20 Does Container Service support granting permissions to sub-accounts in RAM console?

Currently, Container Service does not support granting permissions to sub-accounts in the Resource Access Management (RAM) console (for cloud products that support RAM, see [Cloud services supporting RAM](#)). However, you can grant permissions to sub-accounts in the [Container Service console](#).

For how to grant permissions to sub-accounts in the Container Service console, see [Use sub-accounts](#).