Container Service

Quick Start

Quick Start

Create an Nginx webserver from an image

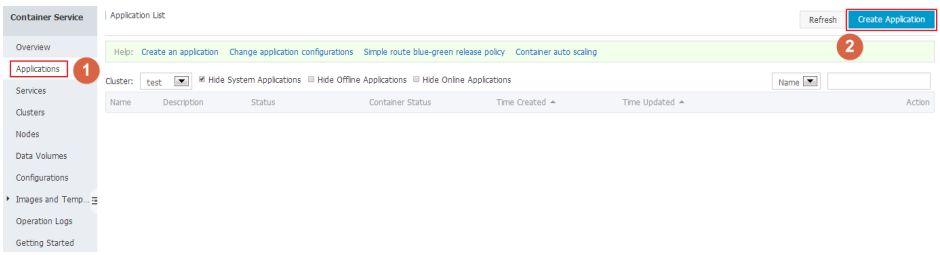
Prerequisite

An existing cluster is required. If you do not have an existing cluster, refer to **Create a cluster** on how to create one.

Operating procedure

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

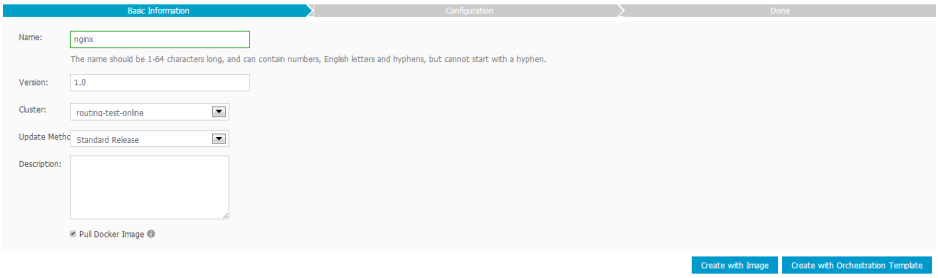
Click **Applications** in the left-side navigation bar, and click **Create Application** in the upper-right corner.



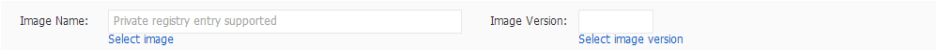
Enter the application information, and click **Create with Image**.

- **Name:** The name of the application to be created. In this example, the application name is **nginx**.
- **Version:** The version of the application to be created. By default, the version is 1.0.
- **Cluster:** The cluster which the application will be deployed to.
- **Update Method:** The release method of the application. You can select **Standard Release** or **Blue-Green Release**. Refer to **Instructions on Release Strategies**.
- **Description:** Information of the application. This information will be displayed in the **Application List** page.
- **Pull Docker Image:** When selected, Container Service pulls the latest Docker image in the registry to create the application, even when the tag of the image does not

change.
In order to improve efficiency, Container Service caches the image; and at deployment, if the tag of the image is consistent with that of the local cache, Container Service uses the cached image instead of pulling the image from the registry. Therefore, if you modify your code and image but do not modify the image tag, Container Service will use the old image cached locally to deploy the application. When this option is selected, Container Service ignores the cached image and re-pulls the image from the registry no matter whether the tage of the image is consistent with that of the cached image, ensuring that the latest image and code are always used.

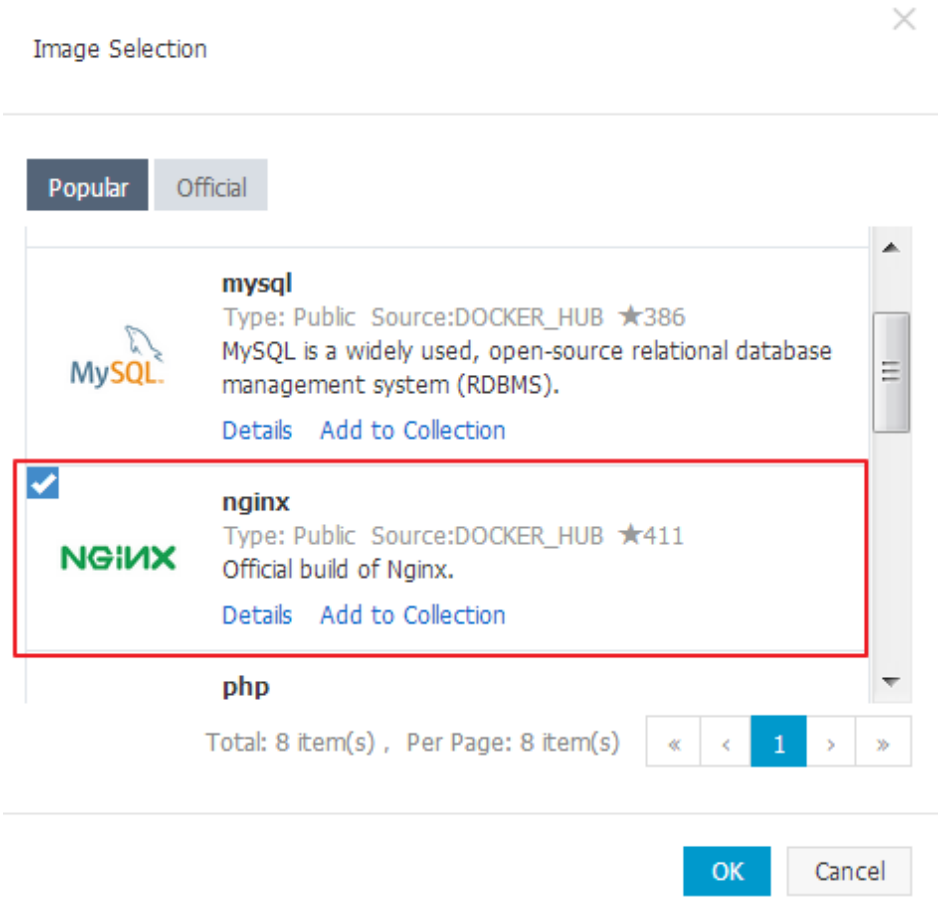


Click **Select image**.



Select **nginx** and click **OK**.

By default, the Container Service uses the latest version of the image. If you want to use another image version, click **Select image version**, then click the desired version, and click **OK**.



Port Mapping shows that the container will listen to Port 80 and Port 443. To enable access to the Nginx server inside the container through public network, you need to configure **Web Routing**.

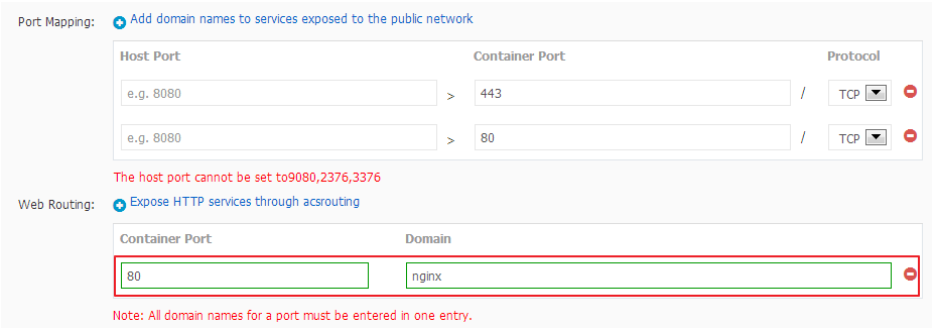
Click the plus icon next to **Web Routing**.

Enter **80** in the **Container Port** field, indicating access to Port 80 on the Nginx container.

Enter **nginx** in the **Domain Name** field. Here, only the domain name prefix is entered. If the domain name prefix is XXX, you get the domain name XXX.$cluster_id.$region_id.alicontainer.com used for testing. In this example, you get the test domain name nginx.c2818a77aac20428488694c0cd1600e6e.cn-shenzhen.alicontainer.com.

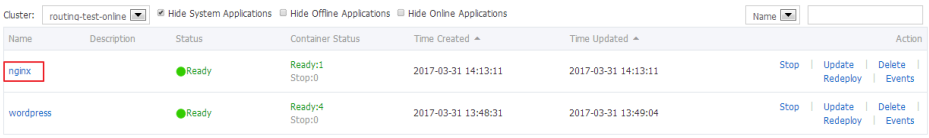
Note: You can also enter your own domain name. For instructions, refer to **Add domain names to services exposed to the public network**. For how to configure the container port and HTTP service domain name for the routing rule, refer to the **routing** label. For information about how the routing service

forwards requests to the container, refer to Expose HTTP services through acsrouting.

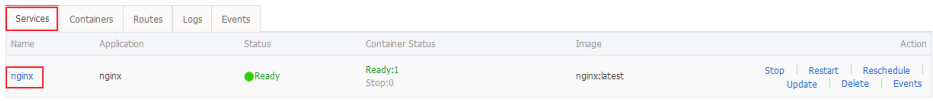


Click **Create**. The Container Service creates the application **nginx** according to the above settings.

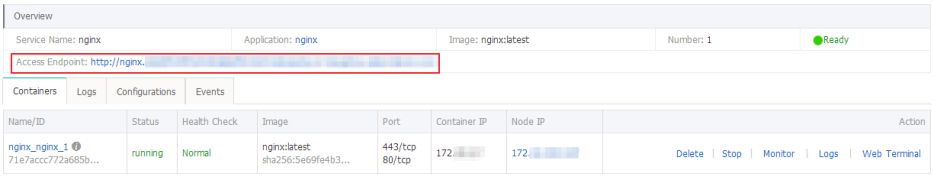
Click **View Application List**, **Back to Application List** or **Applications** in the left navigation pane, and click the application name **nginx** to view the application details.



Click the service name **nginx** in **Services** to view the service details.



Click the access endpoint address of the service **nginx**.



The Nginx server default welcome page is displayed.

Note: If you cannot access this page, refer to Access link troubleshooting.

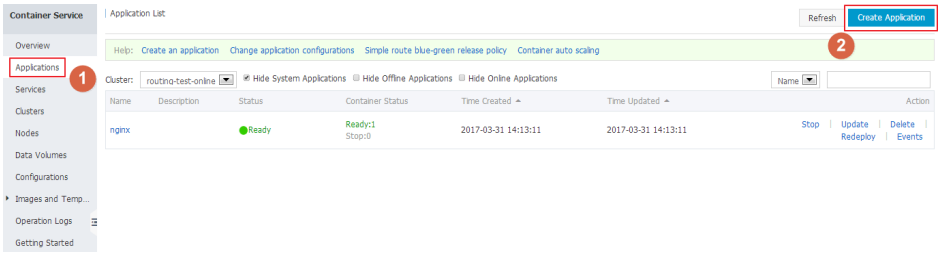
Prerequisite

An existing cluster is required. If you do not have an existing cluster, refer to **Create a cluster** on how to create one.

Operating procedure

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

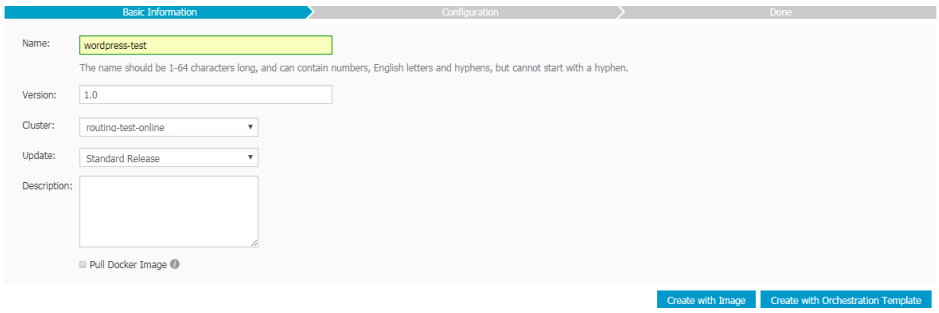
Click **Applications** in the left navigation pane and click **Create Application** in the upper-right corner.



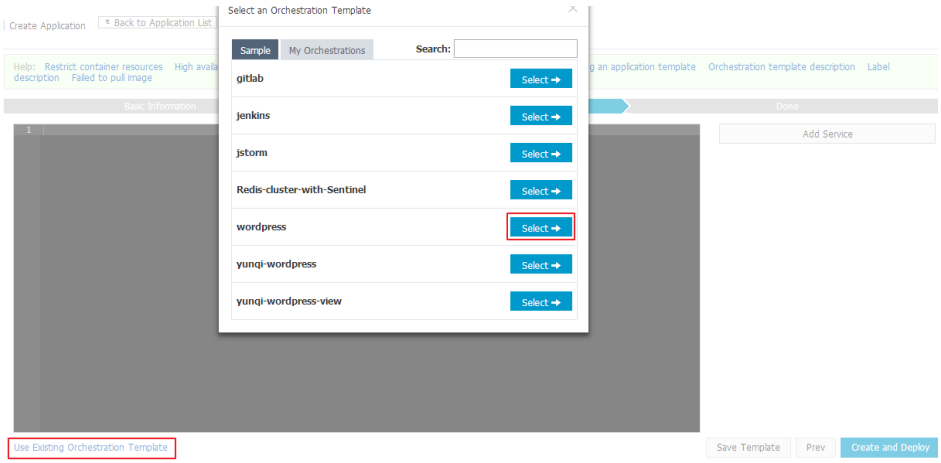
Enter the application information, and click **Create with Orchestration Template**.

- **Name:** The name of the application to be created. In this example, the application name is **wordpress-test**.
- **Version:** The version of the application to be created. By default, the version is 1.0.
- **Cluster:** The cluster which the application will be deployed to.
- **Update Method:** The release method pf the application. You can select **Standard Release** or **Blue-Green Release**. Refer to **Instructions on Release Strategies**.
- **Description:** Information of the application. This information will be displayed in the **Application List** page.
- **Pull Docker Image:** When selected, Container Service pulls the latest Docker image in the registry to create the application, even when the tag of the image does not change.
  In order to improve efficiency, Container Service caches the image; and at deployment, if the tag of the image is consistent with that of the local cache, Container Service uses the cached image instead of pulling the image from the

registry. Therefore, if you modify your code and image but do not modify the image tag, Container Service will use the old image cached locally to deploy the application. When this option is selected, Container Service ignores the cached image and re-pulls the image from the registry no matter whether the tage of the image is consistent with that of the cached image, ensuring that the latest image and code are always used.



Click **Use Existing Application Template**, and click **Select** beside the **wordpress** template.



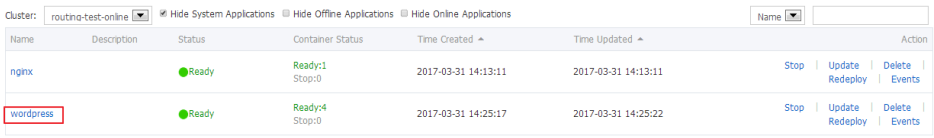
Modify corresponding settings in the template edit box.

You can make modifications in the template, or select the service that you want to modify and click **Edit** to modify the configurations.

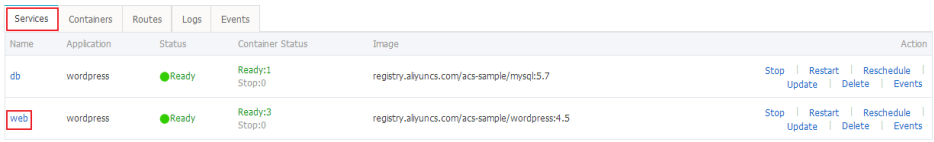
aliyun.routing.port_80: http://wordpress indicates that requests from http://wordpress.$testDomain are forwarded to Port 80 on the container after the container starts running.

Click **Create and deploy**.

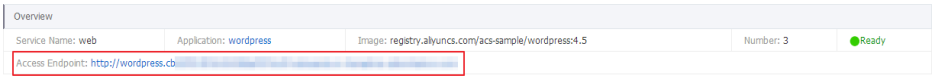
Click **View Application List**, **Back to Application List** or **Applications** in the left navigation pane, and click **wordpress-test** to view the application details.



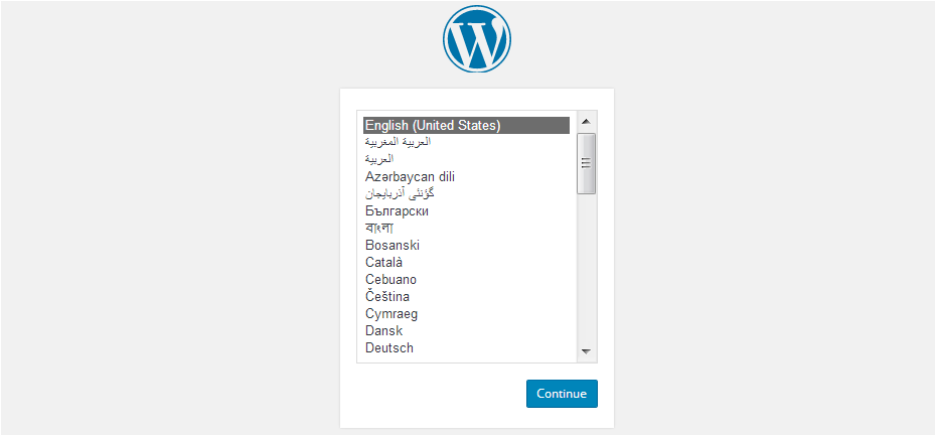
Click the service name **web** to view the service details.



Click the access endpoint address of the service **web**. This address is the domain name for access.



The WordPress page is displayed.



Note:

- The domain name in all preceding examples is only used for testing. You must replace it with your own domain name.
- If you cannot access the endpoint address, refer to Access link troubleshooting.

Connect to a cluster by using Docker tools

The Container Service is fully compatible with the Docker Swarm API. You can access and manage Docker clusters using common Docker tools, such as Docker Client and Docker Compose.

For more information, refer to Docker Swarm and Docker Compose.

Install a certificate

Obtain the access address.

Log on to the Container Service console.

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

Select a cluster in the cluster list and click **Manage**.

The cluster details page is displayed, showing the cluster connection information.

Download and save the certificate.

Configure a TLS certificate before using the preceding service address to access the Docker cluster.

Click **Download Certificate** in the cluster details page to download the certificate which is

contained in the certFile.zip file. In the following example, the downloaded certificate is saved to the ~/.acs/certs/ClusterName/ directory. ClusterName indicates the name of your cluster. You can save the certificate to a different directory, but the ~/.acs/certs/ClusterName/ directory is recommended for easy management.

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

The certFile.zip file contains ca.pem, cert.pem, and key.pem files.

Manage clusters

Use Docker Client to manage clusters

You can use Docker Client to access the container clusters of the Container Service. To do this, you need to configure a certificate and a service address using either of the following two methods.

Configure a certificate using command-line parameters.

```
docker --tlsverify --tlscacert=~/.acs/certs/ClusterName/ca.pem --
tlscert=~/.acs/certs/ClusterName/cert.pem --tlskey=~/.acs/certs/ClusterName/key.pem \
-H=tcp://master4g4.cs-cn-hangzhou.aliyun.com:10351 ps #Replace ClusterName and tcp://master4g4.cs-
cn-hangzhou.aliyun.com:10351 with the actual path and access address
```

Use environment variables.

```
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://master4g4.cs-cn-hangzhou.aliyun.com:10351" #Replace
tcp://master4g4.cs-cn-hangzhou.aliyun.com:10351 with the actual access address
export DOCKER_CERT_PATH=~/.acs/certs/ClusterName #Replace ClusterName with the actual path

docker ps
```

The preceding two examples show how to run the docker ps command in the cluster. You can replace ps with any other Docker command. For example, you can run the docker run command to start a new container.

Use Docker Compose to manage clusters

Docker Compose supports the use of environment variables to declare a service address and a

certificate.

```
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://master4g4.cs-cn-hangzhou.aliyun.com:10351"
export DOCKER_CERT_PATH=~/.acs/certs/ClusterName

docker-compose up
```

Revoke a certificate

In case of accidental disclosure of your certificate during usage, you need to revoke the certificate as soon as possible. Click **Revoke Downloaded Certificate** in the cluster details page to revoke the downloaded certificate. The revoked certificate will then be unavailable, and you can download a new certificate.

> **Note:** Clicking **Revoke Downloaded Certificate** will invalidate the earlier downloaded certificate.