



Software Repository for Container

User Guide

Issue 01

Date 2018-06-26

Copyright © Huawei Technologies Co., Ltd. 2018. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://www.huawei.com>

Email: support@huawei.com

Contents

1 Overview.....	1
1.1 What Is SWR?.....	1
1.2 Relationships Between SWR and Other Services.....	1
1.3 Permission Description.....	3
2 Basis Knowledge of Docker.....	4
2.1 Installing Docker.....	4
2.2 Preparing a Docker Image.....	4
2.3 Creating an Image Package.....	5
3 Uploading an Image to an Image Repository.....	7
4 How to Automatically Create an Image.....	11
4.1 Binding an Account of a Source Code Repository.....	11
4.2 Creating an Image.....	14
5 How to Update the Image Version of an Application.....	18
6 Granting Permissions to Users.....	21
7 Obtaining a Long-Term Valid Docker Login Command.....	22
8 Sharing a Private Image.....	25
9 Using a Public Image.....	27
10 Basic Operations on SWR.....	28
10.1 Uploading an Image Through a Client.....	28
10.2 Uploading an Image Through a Page.....	30
10.3 Editing the Attributes of an Image.....	31
10.4 Obtaining the Download Address of an Image.....	32
10.5 Creating a Containerized Application Using a Container Image.....	32
10.6 Adding an Image to Favorites.....	33
10.7 Creating an Organization.....	33
10.8 Adding Permissions.....	34

1 Overview

[1.1 What Is SWR?](#)

[1.2 Relationships Between SWR and Other Services](#)

[1.3 Permission Description](#)

1.1 What Is SWR?

Software Repository for Container (SWR) provides easy, secure, and reliable management of container images throughout their lifecycles, facilitating quick deployment of containerized services.

Basic Concepts

- **Image:** A Docker image is a special file system that includes all the programs, libraries, resources, and configuration files required to run containers. It also includes some configuration parameters (such as anonymous volumes, environment variables, and users) required for running. An image does not contain any dynamic data, and its content remains unchanged after being built.
- **Container:** A container is an entity that runs Docker images, which are static definitions. The relationship between an image and a container is similar to that between a class and an instance in the object-oriented program design. A container can be created, started, stopped, deleted, or suspended.
- **Organization:** An organization is used to isolate images and assign access permissions (read, edit, and manage) to different users.

1.2 Relationships Between SWR and Other Services

- **OBS**
Object Storage Service (OBS) is an object-based storage service that provides customers with numerous, secure, reliable, and cost-effective data storage capabilities, such as bucket creation, modification, and deletion as well as object upload, download, and deletion.
SWR uses OBS to store private images of users.

- **CCE**
Cloud Container Engine (CCE) is a high-performance, high-reliability service through which enterprises can manage containerized applications. CCE supports native Kubernetes applications and tools, allowing you to easily set up a container runtime environment on the cloud.
SWR supports seamless interconnection with CCE, so you can directly access images in SWR through CCE.
- **CTS**
Cloud Trace Service (CTS) is a professional log audit service. It allows you to collect, store, and query cloud resource operation records and use these records for security analysis, compliance auditing, resource tracking, and fault locating.
With CTS, you can record operations associated with SWR for later query, audit, and backtrack operations.

Table 1-1 SWR operations recorded by CTS

Operation	Resource Type	Transaction Name
Creating a repository	SWR	CreateRepository
Modifying a repository	SWR	UpdateRepository
Deleting a repository	SWR	DeleteRepository
Creating a version	SWR	CreateVersion
Modifying a version	SWR	UpdateVersion
Deleting a version	SWR	DeleteVersion
Creating an image repository	SWR	CreateImageRepository
Modifying an image repository	SWR	UpdateImageRepository
Deleting an image repository	SWR	DeleteImageRepository
Uploading an image package	SWR	UploadImagePackage
Deleting an image tag	SWR	DeleteImageTag
Creating a repository permission	SWR	CreateUserRepositoryAuth
Modifying a repository permission	SWR	UpdateUserRepositoryAuth
Deleting a repository permission	SWR	DeleteUserRepositoryAuth
Creating an organization	SWR	CreateUserNamespace
Deleting an organization	SWR	DeleteUserNamesapce

Operation	Resource Type	Transaction Name
Creating an organization permission	SWR	CreateUserNamespaceAuth
Modifying an organization permission	SWR	UpdateUserNamespaceAuth
Deleting an organization permission	SWR	DeleteUserNamespaceAuth

1.3 Permission Description

By default, the system supports user and resource management permissions. User management refers to the management of users, user groups, and user group permissions. Resource management refers to the control over the operations performed by users on cloud service resources.

For details about SWR user permissions, see [Permission Description](#).

2 Basis Knowledge of Docker

This section describes the basic knowledge of Docker and the background information about Docker and containers. For details, see [Docker Documentation](#).

[2.1 Installing Docker](#)

[2.2 Preparing a Docker Image](#)

[2.3 Creating an Image Package](#)

2.1 Installing Docker

Docker can nearly be installed on all operating systems (OSs). You can select the Docker version to be installed as required. For more information, see <https://docs.docker.com/engine/installation/>.

NOTE

- SWR uses Docker 1.11.2 or later to upload images.
- When installing Docker, connect your PC to the Internet. For a server on a private network, bind an elastic IP address (EIP).

You can run the following commands to quickly install Docker on a PC running Linux.

```
curl -fsSL get.docker.com -o get-docker.sh  
sh get-docker.sh
```

2.2 Preparing a Docker Image

This section describes how to use Dockerfile to prepare a Docker image for a web application. Dockerfile is a text file and contains different instructions. Each instruction is used to build a layer. That is, each instruction describes how to build a layer.

Context

If you use a Nginx image to create a containerized application, the Nginx welcome page is displayed by default. A Nginx image is used as an example to describe how to customize the welcome page of the Nginx image, that is, to change the welcome message to **Hello, SWR!**.

 **NOTE**

To download Docker Hub images in China, access registry.docker-cn.com. The images are hosted on mainland China, so users in China can enjoy faster download speeds and stronger stability. For details, see <https://www.docker-cn.com/registry-mirror>.

Prerequisites

You have installed Docker. For details about how to install Docker, see [2.1 Installing Docker](#).

Procedure

Step 1 Log in to the device running Docker as a **root** user.

Step 2 Create a file named **Dockerfile**.

```
mkdir mynginx
```

```
cd mynginx
```

```
touch Dockerfile
```

Step 3 Edit the **Dockerfile** file.

```
vim Dockerfile
```

New file content:

```
FROM nginx
RUN echo '<h1>Hello,SWR!</h1>' > /usr/share/nginx/html/index.html
```

The following describes the Dockerfile instruction. For details, see <https://hub.docker.com/>.

- **FROM** statement: indicates that the Nginx image is used as the basic image. In a Dockerfile, the FROM instruction must exist and must be the first instruction.
- **RUN** statement: indicates that the echo command is executed to display **Hello, SWR!**.

Step 4 Build a Docker image.

```
docker build -t nginx:v3 .
```

Format of the image build command: `docker build [option] <context path>`

- **-t nginx:v3**: indicates the image name and version.
- **.**: indicates the context path. When the image build command is run, all contents in this path are packaged and sent to a Docker engine to build an image.

Step 5 Run the following command to view the Nginx image. The version is v3.

```
docker images
```

```
----End
```

2.3 Creating an Image Package

This section describes how to compress a Docker image into a .tar or .tar.gz package.

Procedure

Step 1 Log in to the device running Docker as a **root** user.

Step 2 Run the following command to view images:

```
docker images
```

View the image to be exported and its tag.

Step 3 Run the following command to compress the image into a package:

```
docker save [OPTIONS] IMAGE [IMAGE...]
```

OPTIONS: can be set to **--output** or **-o**, indicating that the image is exported to a file.

----End

Example:

```
$ docker save nginx:latest > nginx.tar
$ ls -sh nginx.tar
108M nginx.tar

$ docker save php:5-apache > php.tar.gz
$ ls -sh php.tar.gz
372M php.tar.gz

$ docker save --output nginx.tar nginx
$ ls -sh nginx.tar
108M nginx.tar

$ docker save -o nginx-all.tar nginx
$ docker save -o nginx-latest.tar nginx:latest
```

3 Uploading an Image to an Image Repository

This chapter uses the 2048 application as an example to describe how to create an image based on the Dockerfile file and upload it to an image repository. You can compile a Dockerfile file and use `alpine:3.7` as the basic image to build an image for the 2048 application.

Prerequisites

- You have installed Docker. For details about how to install Docker, see [2.1 Installing Docker](#).
- You have obtained the 2048 application and download its image to a local PC.

Creating an Image

Step 1 Log in to a virtual machine (VM).

Step 2 Create the **2048-demo** directory.

```
mkdir 2048-demo
```

Step 3 Download the source code of the 2048 application to the **myAlpine** directory.

Source code address of the application: <https://github.com/gabrielecirulli/2048.git>

Step 4 Create a file named **Dockerfile**.

```
cd myAlpine
```

```
touch Dockerfile
```

Step 5 Edit the Dockerfile file.

```
vi Dockerfile
```

```
FROM alpine:3.7
RUN apk --update add nginx
COPY 2048 /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- FROM: Specify the basic image `alpine:3.7`.

- RUN: Install the Nginx process.
- COPY: Copy the source code of the 2048 application to the `/usr/share/nginx/html` directory in the container.
- EXPOSE: Expose port 80 of the container.
- CMD: Specify the default command to run the container.

Save the settings and exit.

Step 6 Run the following command to create an image:

```
docker build -t 2048-demo:v1 .
```

The image format is as follows: `docker build [option] <context path>`. For details, see <https://docs.docker.com/engine/reference/commandline/build/>.

- **-t 2048-demo:v1**: Specify the image name and tag.
- **.**: Specify the directory where the Dockerfile file is located. An image will be created based on the content of the Dockerfile file.

Step 7 Run the following command to query details. The **2048-demo** image with tag `v1` is successfully created.

```
docker images | grep 2048-demo
```

```
[root@ecs-4dbc myAlpine]# docker images | grep 2048-demo
2048-demo          v1                6df6f67430b3      44 seconds ago
7.97MB
```

----End

Creating an Organization

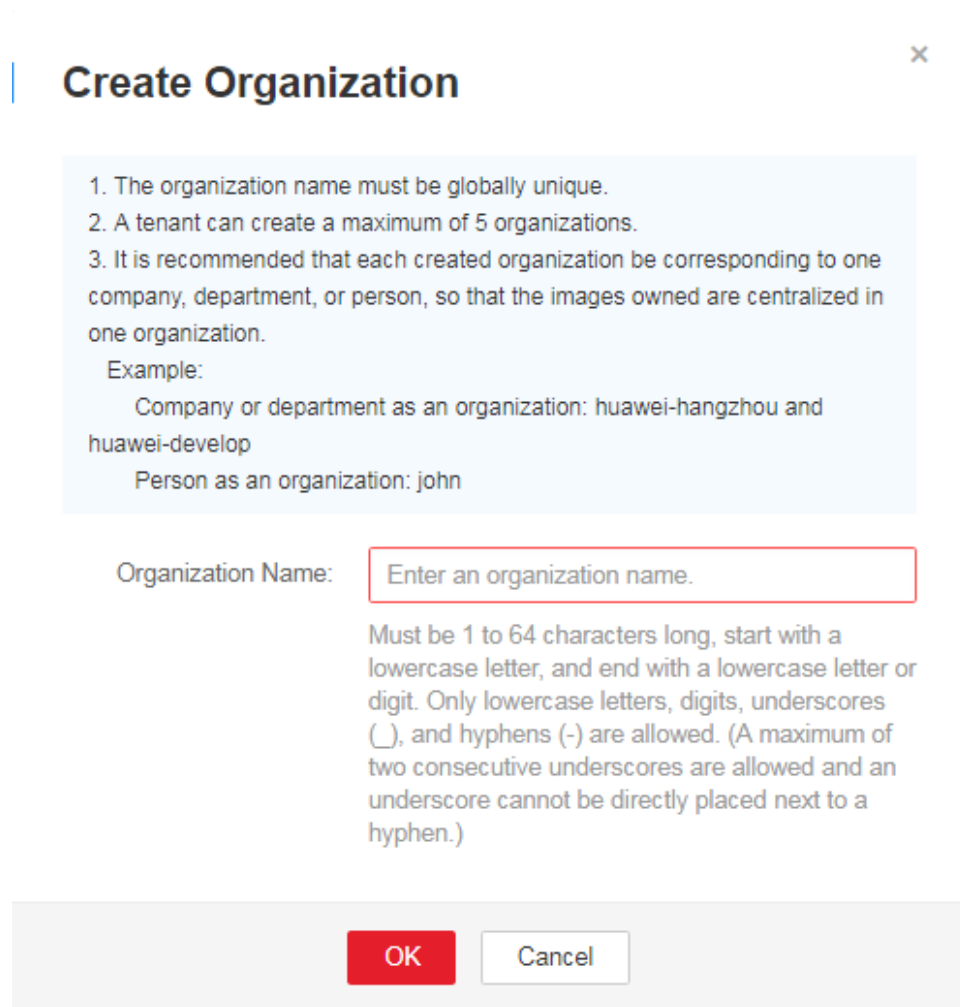
An organization is used to isolate images and assign access permissions (read, edit, and manage) to different users.

- Read: A user can only download images. Image upload is not supported.
- Edit: A user can download or upload images, edit image attributes, and create triggers.
- Manage: A user can download or upload images, delete images or versions, edit image attributes, grant permissions, add triggers, and share images with other users.

Step 1 Log in to the SWR console.

Step 2 In the navigation pane, choose **Organization Management** and click **Create Organization**. On the page that is displayed, specify **Organization Name** and click **OK**.

Figure 3-1 Creating an organization



----End

Uploading an Image Through a Client

You can upload an image to an image repository of SWR through the Docker client by running commands.

This chapter uses the **2048-demo:v1** image as an example to describe how to upload an image. Images that have been successfully uploaded are displayed on the **My Images** page.

NOTICE

- When you use the client to upload images, the size of each layer of an image cannot exceed 10 GB.
- The Docker client used to upload images must be in 1.11.2 or later.

Step 1 Access SWR.

1. Log in to the SWR console.


2. In the navigation pane, choose **My Images** and then click **Upload Through Client**. On the page that is displayed, click **Generate a Temporary Docker Login Command** and click  to copy the **docker login** command. Then, record the domain name (that is, the address of the current image repository) at the end of the **docker login** command.

Figure 3-2 Obtaining the **docker login** command

Step 2. Obtain the permission to log in to Docker, and copy the Docker login command to the node for execution.
Click [Generate a Temporary Docker Login Command](#) ([Obtain a Long-term Valid Docker Login Command](#))

 **NOTE**

The **docker login** command obtained in this step is valid within 16 hours. For details about how to obtain a long-term valid **docker login** command, see [7 Obtaining a Long-Term Valid Docker Login Command](#).

3. Run the copied **docker login** command on the device running Docker.
After you successfully log in to the Docker client, **login succeeded** is displayed.

Step 2 Add a tag to the **2048-demo:v1** image on the device where Docker is installed.

docker tag *Image name:version number Image repository address/Organization name/Image name:version number*

Example:

docker tag *2048-demo:v1 swr.cn-north-1.myhuaweicloud.com/group/2048-demo:v1*

In the preceding information:

- *swr.cn-north-1.myhuaweicloud.com* indicates the image repository address of SWR.
- *group* indicates the organization name. If no organization is created, SWR automatically creates an organization based on the organization name.
- *2048-demo:v1* indicates the image name and version number.

Step 3 Upload the image to the image repository:

docker push *Image repository address/Organization name/Image name:version number*

Example:

docker push *swr.cn-north-1.myhuaweicloud.com/group/2048-demo:v1*

If the following information is displayed on the terminal, the image is successfully pushed.

```
6d6b9812c8ae: Pushed
695da0025de6: Pushed
fe4c16cbf7a4: Pushed
v1: digest:
sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948
```

You can return to **My Images** and refresh the page to view the information about the uploaded image.

----End

4 How to Automatically Create an Image

SWR can connect to source code hosting websites like GitHub and automatically creates images when code is updated.

This chapter describes how to automatically create an image. You need to bind an account of a source code repository, authorize SWR to download source code, create a build task, and download code from a source code hosting website to create an image.

[4.1 Binding an Account of a Source Code Repository](#)

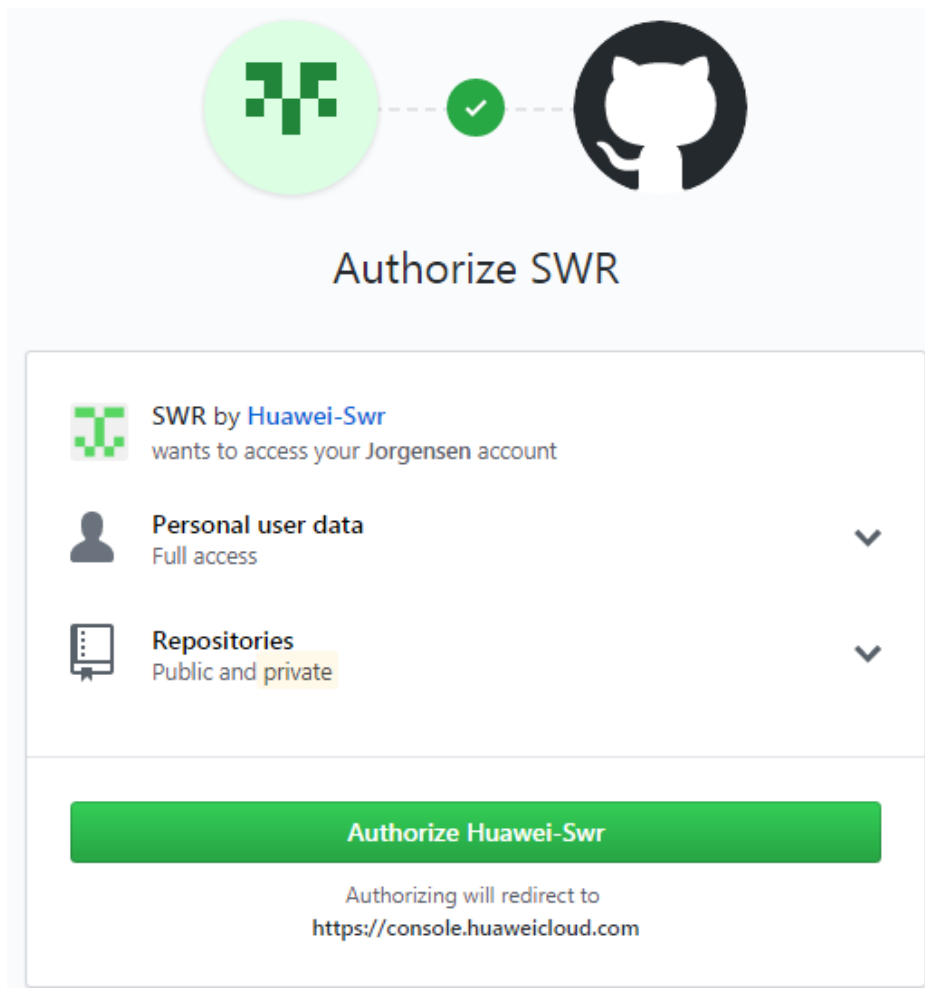
[4.2 Creating an Image](#)

4.1 Binding an Account of a Source Code Repository

Before creating an image, you need to bind an account of a source code hosting repository and authorize SWR to download source code. Currently, SWR supports source code hosting websites GitHub, GitLab, and Gitee.

Binding a GitHub Account

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **Image Build > Source Code Hosting**.
- Step 3** Click **Bind** in the row of GitHub.
- Step 4** Log in to GitHub.
- Step 5** Click **Authorize Huawei-Swr** to complete the binding.

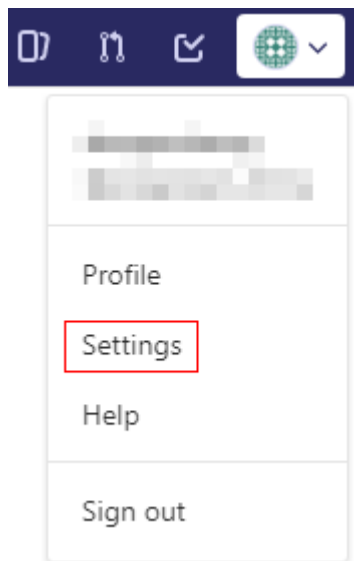


----End

Binding a GitLab Account

Step 1 Obtain the GitLab access token.

1. Log in to GitLab.
2. Click the icon in the upper right corner and select **Settings**.



3. Choose **Access Tokens** on the left side, enter a name, select **read_registry**, and click **Create personal access token** to create a token.

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Add a personal access token

Pick a name for the application, and we'll give you a unique personal access token.

Name

build_image

Expires at

Scopes

api Access the authenticated user's API

Full access to GitLab as the user, including read/write on all their groups and projects

read_user Read the authenticated user's personal information

Read-only access to the user's profile information, like username, public email and full name

read_registry Read Registry

Read Registry

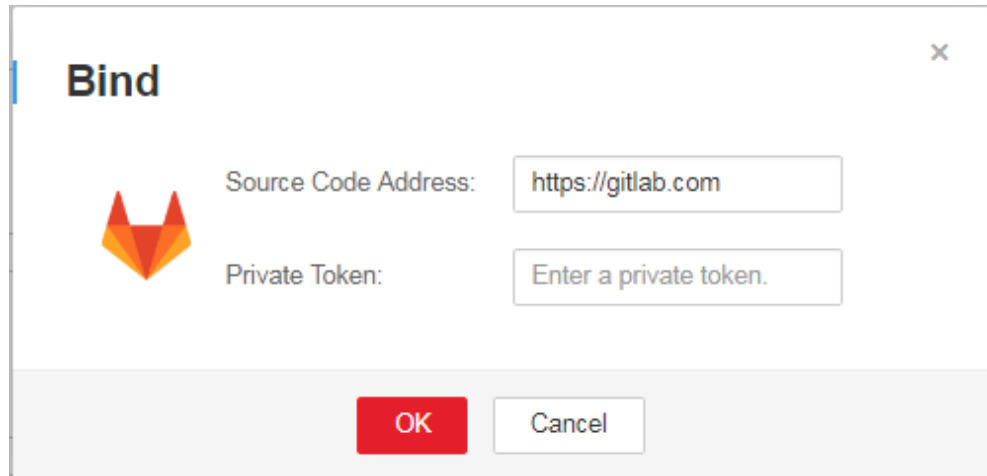
Create personal access token

4. Copy and save the token.

Step 2 Log in to the SWR console.

Step 3 In the navigation pane, choose **Image Build > Source Code Hosting**.

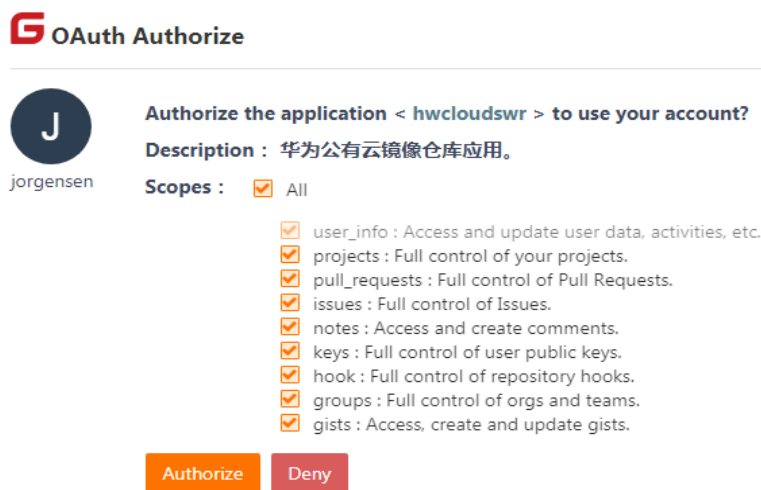
Step 4 Click **Bind** next to GitLab, enter the token obtained in **Step 1**, and click **OK**.



----End

Binding a Gitee Account

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **Image Build > Source Code Hosting**.
- Step 3** Click **Bind** next to Gitee.
- Step 4** Log in to Gitee.
- Step 5** Click **Authorize** to complete the binding.



----End

4.2 Creating an Image

After binding an account of a source code repository, you need to create a build task and download code from a source code hosting website to automatically create an image.

Prerequisites

The source code must contain the Dockerfile file.

Procedure

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **Image Build > Build Task**. On the page that is displayed, click **Create Build Task**.
- Step 3** Set the build task parameters. For details, see [Table 4-1](#).

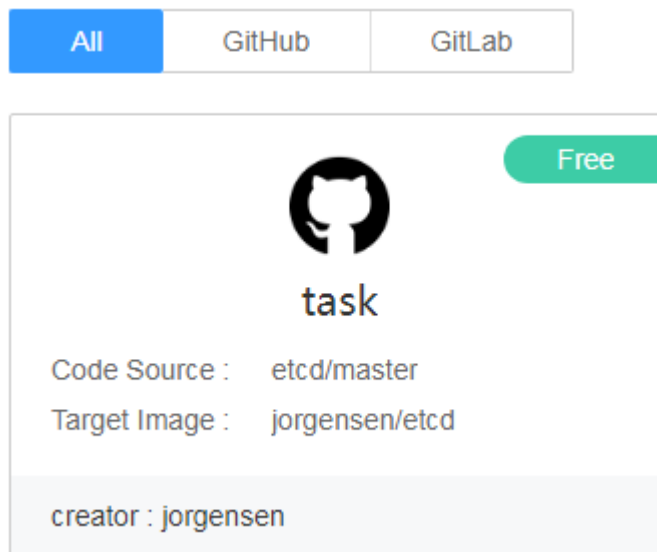
Table 4-1 Parameter description

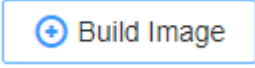
Parameter	Description
*Task Name	Name of a build task.
Description	Description of a build task. The Markdown format is supported.
*Code Source	Select a code repository. If you have not bound an account of a source code hosting website, click Bind Account . For details, see 4.1 Binding an Account of a Source Code Repository .
*Build Settings	<ul style="list-style-type: none">● Code Branch/Tag: Select a code branch to be used.● Dockerfile Directory: Enter the path of the Dockerfile file in the source code.<ul style="list-style-type: none">- If the Dockerfile file is in the root directory and the file name is Dockerfile, enter /Dockerfile here.● Custom Build Path: path of a build project, which is customized by users. The path or its sub-paths must contain the Dockerfile file.<ul style="list-style-type: none">- If the Dockerfile file is in the root directory, leave this parameter blank.- If the Dockerfile file is in the /src/a directory and the file name is Dockerfile, enter /src/a/Dockerfile in Dockerfile Directory, and enter /, /src, or /src/a in Custom Build Path.
Auto Build	Set automatic build. Currently, automatic build can be triggered in two modes. The code is automatically pulled to build an image when either of the following conditions is met: <ul style="list-style-type: none">● A new tag is added.● Code is submitted to a branch. In this example, an image is built when code is submitted to a branch.
Build Cache	If Cache Construction Information is selected, such information can be cached to improve the speed to build images in the future.
*Organization	Organization to which the output image belongs.

Parameter	Description
*Image Name	Name of an image. The image name must be a string of 1 to 128 characters, and start and end with a lowercase letter or digit. Only lowercase letters, digits, underscores (_), and hyphens (-) are allowed. A maximum of two consecutive underscores are allowed and an underscore cannot be directly placed next to a hyphen.
*Image Version Prefix	For example, if the prefix is set to version , the version number of the image successfully built is version.n . <i>n</i> indicates the number of successful builds. It starts from 1 and is incremented by 1 each time a build task is successful. The prefix must be a string of 1 to 16 characters and start with a letter, digit, or underscore (_). Only letters, digits, periods (.), underscores (_), and hyphens (-) are allowed.

Step 4 Click **Create**.

After you successfully create a build task, this task will be displayed on the build task page, as shown in the following figure.



Step 5 Click the build task. On the details page that is displayed, click  on the upper right corner, select **Commit ID**, and click **Build**.

Build Image [Close]

Source: github

Branch: master

User-defined commitID Select a commit ID.

Due to international network congestion, interaction with foreign sites may be unstable, which affects the build.

Build Cancel

After the build task starts, detailed build information is displayed on the **Build History** tab page.

After successfully building an image, you can view this image on the **My Images** page.

---End

5 How to Update the Image Version of an Application

SWR can be used together with **CCE**. When the version of an image is updated, the application that is deployed using the image can be automatically updated.

You only need to set a trigger for the image. With the trigger, the system automatically performs an update operation, for example, updates the application that uses the image each time when the image version is updated.

Prerequisites

Ensure that a containerized application has been created and its image has been deployed on CCE before updating the image version.

If no application is created, log in to the CCE console to create one. For details about how to create an application, see *Cloud Container Engine User Guide*.

Creating a Trigger

Step 1 Log in to the SWR console.

Step 2 In the navigation pane, choose **My Images** and click an image to enter its details page.


Step 3 Click the **Trigger** tab. On the page that is displayed, click , set parameters based on the following table, and click **OK**.

Table 5-1 Trigger

Parameter	Description
Trigger Name	The value must be a string of 1 to 64 characters that start with a letter. It cannot end with an underscore (_) or a hyphen (-). Only letters, digits, underscores, and hyphens are allowed, but consecutive underscores or hyphens are prohibited.

Parameter	Description
Triggering Condition	<p>Application deployment is triggered upon the release of a new image version. The following trigger conditions are supported:</p> <ul style="list-style-type: none"> ● All Trigger: triggers deployment when a new image version is generated or an image version is updated. ● Specified Tag Trigger: triggers deployment when the specified image version is generated or updated. ● Regular Trigger: triggers deployment when an image version that satisfies regular expression conditions is generated or updated. <p>In this example, All Trigger is selected. In this case, deployment is triggered when a new image version is generated.</p>
Trigger Action	Currently, only container images can be updated. You need to specify the desired application and the container image of the application.
Trigger Status	Select Enable .
Select Container	Select the container for which the image is to be updated.

----End

Verification Operations

Step 1 Upload the new image version to the image repository of SWR.

In this example, **All Trigger** is selected. In this case, deployment is triggered when a new image version is generated or updated. Upload the new image version to SWR to verify the

triggering result. For other verification scenarios, for example, image version update, see *SWR Best Practice*.

Step 2 View deployment results.

An image is automatically built. After the image is built, you can view the new image version on the image details page.

Organization/Image Name:	full/game2048	Category:	Other
Used Versions/Number of Versions:	2/100	Downloads:	7
Created:	03/29/2018 16:45:17 GMT+08:00	Space Used:	9.4MB

Image Version	Size	Updated	External Download Command	Internal Download Command	Operation
Version 2	4.7MB	04/24/2018 18:39:42 GMT+08:00	docker pull swr.cn-north-1.myhuaweicloud.com/full/game2048:version-2	docker pull 100.125.0.198:20202/full/game2048:version-2	View Manifest Delete
Version 1	4.7MB	03/29/2018 16:45:17 GMT+08:00	docker pull swr.cn-north-1.myhuaweicloud.com/full/game2048:version-1	docker pull 100.125.0.198:20202/full/game2048:version-1	View Manifest Delete

The image is updated, further triggering the update of the containerized application.

Application Name:	game2048	Type:	Stateless application
Status:	Running	Namespace:	default
Instances (Normal/All):	1/1	Application Group Name:	default
Created:	04/24/2018 18:13:34 GMT+08:00	External Access Address:	117.78.27.59:32249
Description:	--	Label:	Label Management

Instance (Pod)	Status	Latest Event	Requested CPUs	Requested Memo...	Homing Node	Runtime
game2048-427...	Running	--	0.25	0.50	192.168.0.160	6 minutes

On the application details page of the CCE console, you can view that an application instance is being created. After the instance is created, the application has been updated.

应用名称:	game2048	类型:	无状态应用
状态:	运行中	应用命名空间:	default
实例个数(正常全部):	1/1	应用组名称:	默认
创建时间:	2018/03/29 17:00:33 GMT+08:00	外部访问地址:	117.78.35.15:30430
描述:	--	标签:	标签管理

实例(Pod)	状态	最新事件	CPU申请量 (core)	内存申请量 (GiB)	所在节点	运行时长
game2048-845216050...	运行中	--	--	--	192.168.0.213	22分钟

Switch to **Access Mode** and click the access link IP address to access the application details page. The updated application information is displayed.

----End

6 Granting Permissions to Users

After creating an image, you can grant permissions to a user so that the user has the permissions to read, edit, and manage the image.

- **Read:** A user can only download images. Image upload is not supported.
- **Edit:** A user can download or upload images, edit image attributes, and create triggers.
- **Manage:** A user can download or upload images, delete images or versions, edit image attributes, grant permissions, add triggers, and share images with other users.

Step 1 Log in to the SWR console.

Step 2 In the navigation pane, choose **My Images**. On the page that is displayed, click the name of the image to be edited.

Step 3 On the image details page, click the **Rights Management** tab.

Step 4 Click **Add Permission**, select the user name under the tenant, and add the **Read**, **Edit**, or **Manage** permission. After a permission is added, the user has the corresponding permission.

Step 5 You can also modify and delete the user permissions.

----End

7 Obtaining a Long-Term Valid Docker Login Command

This chapter describes how to obtain a long-term valid **docker login** command.

Procedure

Step 1 Obtain an image repository address and the name of a regional project.

1. Visit <https://console.huaweicloud.com/iam/#/myCredential>.
2. On the **Project List** tab page, view the project corresponding to the current region.

The **cn-north-1** project corresponds to **CN North-Beijing1**, as shown in the following figure.

Figure 7-1 Regions and projects

Region	Project
AP-Hong Kong	ap-southeast-1
CN East-Shanghai2	cn-east-2
CN North-Beijing1	cn-north-1
CN Northeast-Dalian	cn-northeast-1
CN South-Guangzhou	cn-south-1

The image repository address is in the format of *swr.name of the regional project.myhuaweicloud.com*. For example, the image repository address corresponding to CN North-Beijing1 is **swr.cn-north-1.myhuaweicloud.com**.

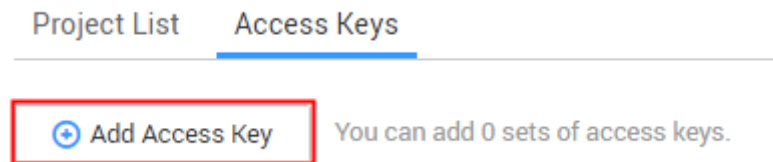
Step 2 Obtain the Access Key ID/Secret Access Key (AK/SK).

 **NOTE**

If the AK/SK is available, you can directly use it.

1. Visit <https://console.huaweicloud.com/iam/#/myCredential>.
2. On the **Access Keys** tab page, click **Add Access Key** to create an access key.

Figure 7-2 Adding an access key



3. Enter the login password of the current user and verify the password by email or mobile phone.

 **NOTE**

For the accounts created in Identity and Access Management (IAM), if no email address or mobile phone is filled during account creation, you only need to verify the login password.

4. Click **OK** to download an access key.

Properly keep the downloaded key. You need to upload the key when creating a cluster. Otherwise, the cluster cannot be created.

 **NOTE**

Keep the access key confidential to prevent information leakage.

Step 3 Log in to a Linux PC and run the following command to obtain the login key:

```
printf "$AK" | openssl dgst -binary -sha256 -hmac "$SK" | od -An -vtx1 | sed 's/[ \n]//g' | sed 'N;s/\n/'
```

In the preceding information, **\$AK** and **\$SK** respectively indicate the AK and SK obtained in [Step 2](#).

Example:

```
[root@SZV1000258977 ~]# printf "DKAKX9J60BEVMARHLBQM" | openssl dgst -binary -sha256 -hmac "0uDrd9HcRhmngEhAXo6SQiflN1UqufLF531jifkX" | od -An -vtx1 | sed 's/[ \n]//g' | sed 'N;s/\n/'  
7ca3582173f52caa98fcf87389e9cc26d007a2e4b2f6231006a301568f2e1ef8
```

Step 4 Splice the **docker login** command in the following format:

```
docker login -u [Name of the regional project]@[AK] -p [Login key] [Image repository address]
```

In the preceding information, the name of the regional project and image repository address are obtained in [Step 1](#), the AK is obtained in [Step 2](#), and the login key is the execution result in [Step 3](#).

Example:

```
docker login -u cn-north-1@DKAKX9J60BEVMARHLBQM -p  
7ca3582173f52caa98fcf87389e9cc26d007a2e4b2f6231006a301568f2e1ef8 swr.cn-  
north-1.myhuaweicloud.com
```

----End

8 Sharing a Private Image

After an image is uploaded, you can share the image with other users and grant them the permission to operate the image.

Prerequisites

Only tenants and users with the private image management permission can share images. Shared users can only read and download images.

Procedure

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **My Images**. On the page that is displayed, click the name of the image to be edited.
- Step 3** On the image details page, click the **Sharing** tab.
- Step 4** Click **Shared Image**. In the dialog box that is displayed, set the shared account, deadline, and permission, and click **OK**.

×

Shared Image

* Share To :

* Deadline : × |

Permanently Valid

Description :
0/1000

* access :

OK Cancel

Step 5 After the configuration is complete, go back to the **Owned Image** tab page, and select **My Shared Image** to view all the shared images.

Owned Image Other Sharing

My Shared Image All organizations

Image Name	Organization	Number of Versions	Updated
alpine	dingshujie	12	04/20/2018 09:36:36 GMT+08:00

Step 6 Shared users can log in to the SWR console and choose **My Images > Other Sharing** to view the image information.

----End

9 Using a Public Image

To use a public image, you can log in to the VM and run the **docker pull** command to obtain the image from Docker Hub.

10 Basic Operations on SWR

- [10.1 Uploading an Image Through a Client](#)
- [10.2 Uploading an Image Through a Page](#)
- [10.3 Editing the Attributes of an Image](#)
- [10.4 Obtaining the Download Address of an Image](#)
- [10.5 Creating a Containerized Application Using a Container Image](#)
- [10.6 Adding an Image to Favorites](#)
- [10.7 Creating an Organization](#)
- [10.8 Adding Permissions](#)

10.1 Uploading an Image Through a Client

You can upload an image to SWR through the Docker client.

The **nginx:v3** image is used as an example. Images that have been successfully uploaded are displayed on the **My Images** page.



NOTICE

When you use the client to upload images, the size of each layer of an image cannot exceed 10 GB.

Prerequisites

The Docker client used to upload images must be in 1.11.2 or later.

Procedure

Step 1 Access SWR.


1. Log in to the SWR console.
2. In the navigation pane, choose **My Images** and then click **Upload Through Client**. On the page that is displayed, click **Generate a Temporary Docker Login Command** and click  to copy the **docker login** command. Then, record the domain name (that is, the address of the current image repository) at the end of the **docker login** command.

Figure 10-1 Obtaining the **docker login** command

Step 2. Obtain the permission to log in to Docker, and copy the Docker login command to the node for execution.

Click [Generate a Temporary Docker Login Command](#) ([Obtain a Long-term Valid Docker Login Command](#))

 **NOTE**

The **docker login** command obtained in this step is valid within 16 hours. For details about how to obtain a long-term valid **docker login** command, see [7 Obtaining a Long-Term Valid Docker Login Command](#).

3. Run the copied **docker login** command on the device running Docker.
After you successfully log in to the Docker client, **login succeeded** is displayed.

Step 2 Add a tag to the **nginx:v3** image.

1. View the image name:

docker images

2. Add a tag to the **nginx:v3** image.

docker tag *Image name:version number Image repository address/Organization name/ Image name:version number*

Example:

docker tag *nginx:v3 swr.cn-north-1.myhuaweicloud.com/group/nginx:v3*

In the preceding information:

- *swr.cn-north-1.myhuaweicloud.com* indicates the image repository address of SWR.
- *group* indicates the organization name. If no organization is created, SWR automatically creates an organization based on the organization name.
- *nginx:v3* indicates the image name and version number.

Step 3 Upload the image to the image repository:

docker push *Image repository address/Organization name/Image name:version number*

Example:

docker push *swr.cn-north-1.myhuaweicloud.com/group/nginx:v3*

If the following information is displayed on the terminal, the image is successfully pushed.

```
6d6b9812c8ae: Pushed
695da0025de6: Pushed
fe4c16cbf7a4: Pushed
v3: digest:
sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948
```

You can return to **My Images** and refresh the page to view the information about the uploaded image.

----End

10.2 Uploading an Image Through a Page

You can upload an image to SWR through a page.



NOTICE

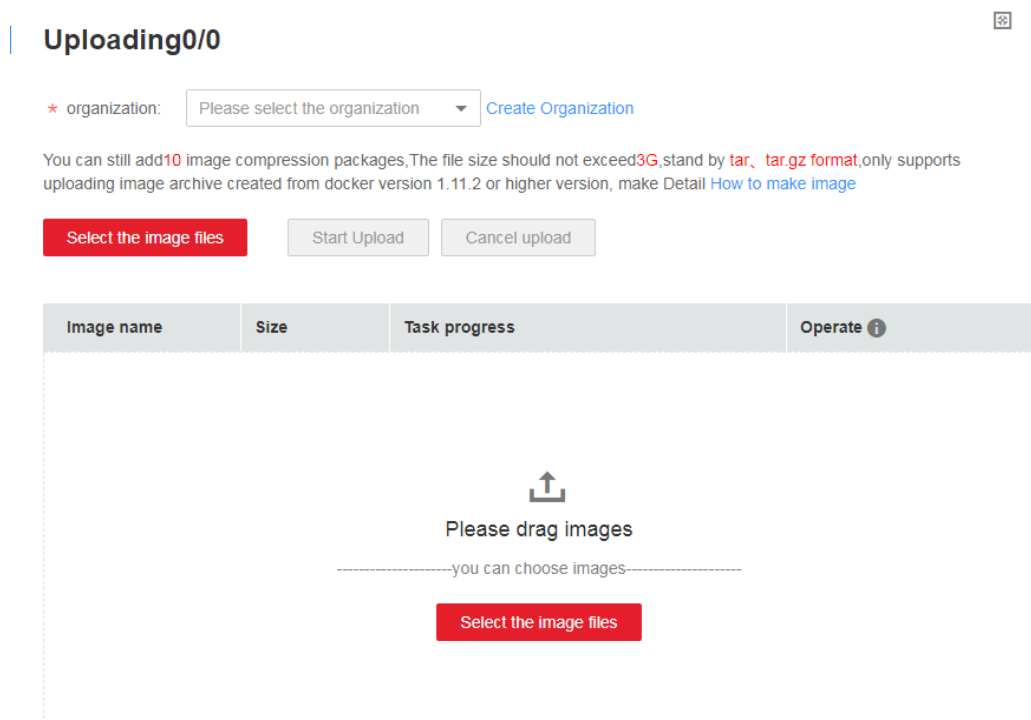
You can upload a maximum of 10 files at a time. The size of a single file, for example, a decompressed file cannot exceed 3 GB. Only image files in .tar or .tar.gz format can be uploaded through the page.

Prerequisites

- You have created an organization. For more information about how to create an organization, see [10.7 Creating an Organization](#).
- You have compressed the image to be uploaded into a .tar or .tar.gz package. For more information about how to create an image package, see [2.2 Preparing a Docker Image](#).
- Only image packages created using Docker 1.11.2 or later can be uploaded.

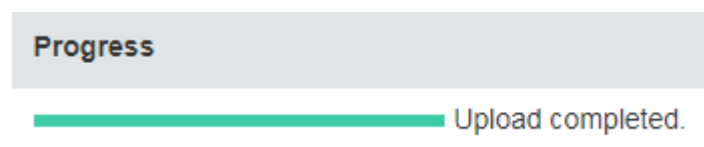
Procedure

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **My Images** and then click **Upload Through Page**.
- Step 3** On the page that is displayed, select an organization to which the image is to be uploaded. Then, click **Select Image File** to upload the desired image file, as shown in the following figure.

Figure 10-2 Uploading an image

Step 4 On the page that is displayed, click **Start Upload**.

Wait until **Upload completed** is displayed, which indicates that the image is successfully uploaded.

Figure 10-3 Upload completed

---End

10.3 Editing the Attributes of an Image

After uploading an image, set the attributes of the image as required, including the image category and description.

Procedure

Step 1 Log in to the SWR console.

Step 2 Choose **My Images** from the main menu and click the name of the image to be edited.

Step 3 On the details page, click **Edit** in the upper right corner. On the page that is displayed, edit the image type, category, and description as required, and click **OK**.

Figure 10-4 Editing the attributes of an image

Edit

Organization: jorgensen

warehouseName: redis-slave

Type: Public Private

Category:


description:
0/30000

----End

10.4 Obtaining the Download Address of an Image

After uploading an image, obtain the image download address and run the **Docker pull** command to download the image.

Procedure

- Step 1** Log in to the SWR console.
- Step 2** In the navigation pane, choose **My Images**. On the page that is displayed, click an image name.
- Step 3** On the image details page, click  on the right of the image version and copy the image download address.

----End

10.5 Creating a Containerized Application Using a Container Image

Deploy an image on CCE to create a containerized application.

For details, see *Cloud Container Engine User Guide*.

10.6 Adding an Image to Favorites

You can add an image to **My Favorites** for ease of use.

Procedure

Step 1 Log in to the SWR console.

Step 2 On the main menu, choose **Image Resource > Official Docker Image**.

Step 3 In the image list, click the icon for saving the image to favorites.

After adding an image to favorites, you can view this image on the **My Favorites** page.

---End

10.7 Creating an Organization

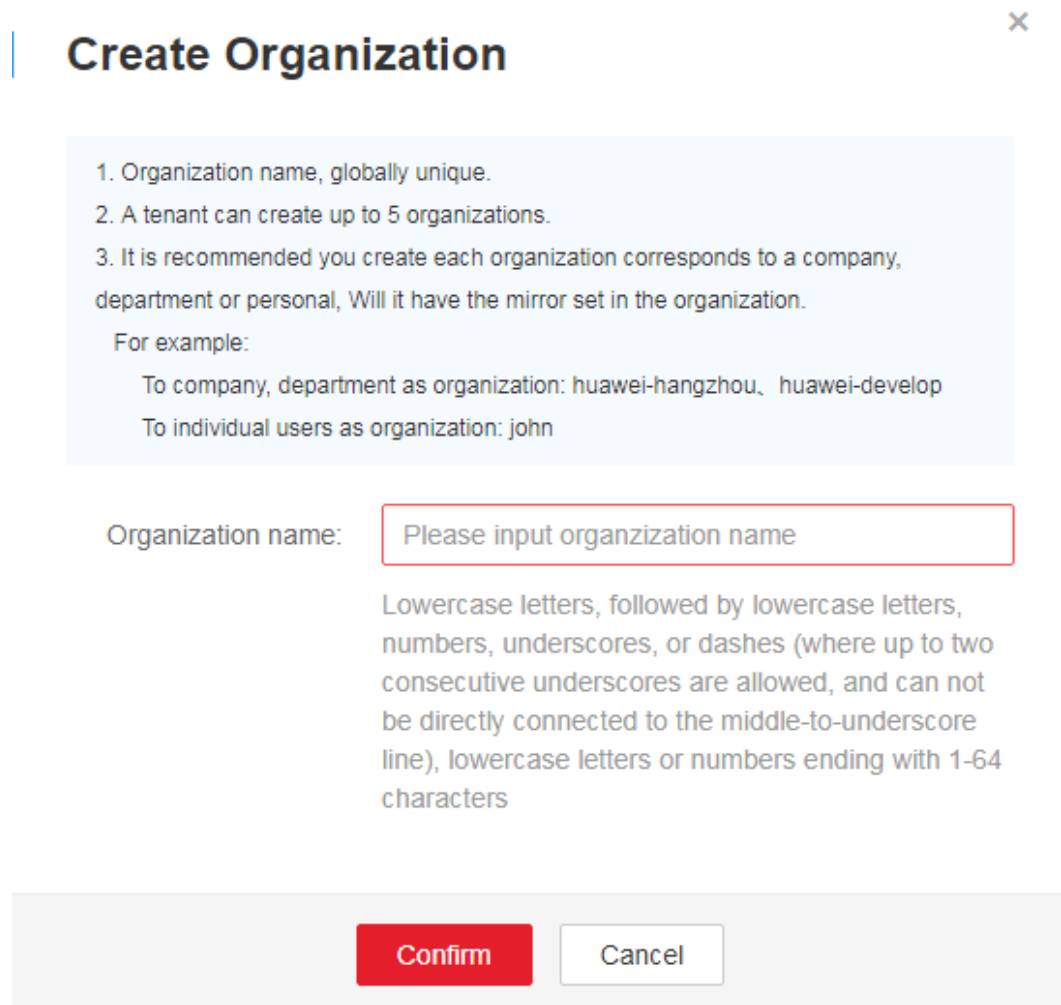
An organization is used to isolate images and assign access permissions (read, edit, and manage) to different users.

Procedure

Step 1 Log in to the SWR console.

Step 2 Choose **Organization Management** from the main menu and click **Create Organization**.
On the page that is displayed, specify **Organization Name** and click **OK**.

Figure 10-5 Creating an organization



Create Organization ×

1. Organization name, globally unique.
2. A tenant can create up to 5 organizations.
3. It is recommended you create each organization corresponds to a company, department or personal, Will it have the mirror set in the organization.

For example:
To company, department as organization: huawei-hangzhou, huawei-develop
To individual users as organization: john

Organization name:

Lowercase letters, followed by lowercase letters, numbers, underscores, or dashes (where up to two consecutive underscores are allowed, and can not be directly connected to the middle-to-underscore line), lowercase letters or numbers ending with 1-64 characters

Confirm

----End

10.8 Adding Permissions

This section describes how to add permissions to users in an organization so that the users can read, edit, and manage all images in the organization.

User permissions can be classified into three types:

- Read: A user can only download images. Image upload is not supported.
- Edit: A user can download or upload images, edit image attributes, and create triggers.
- Manage: A user can download or upload images, delete images or versions, edit image attributes, grant permissions, add triggers, and share images to other users.

Prerequisites

Permissions can be added only to users with the **Manage** permission.

Procedure

- Step 1** Log in to the SWR console.
- Step 2** Click **Organization Management** from the main menu and click the name of the organization.
- Step 3** On the **User** tab page, click **Add Permission**. On the page that is displayed, select permissions for the user and click **OK**.

---End