SoftWare Repository for Container(Enterprise Edition)

User Guide

Issue	01
Date	2025-09-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are the property 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 Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 IAM-based Permissions Management

1.1 Creating a User and Granting Permissions

Scenarios

System-defined permissions in role/policy-based authorization provided by **Identity and Access Management (IAM)** let you control access to your SWR resources. With IAM, you can:

- Create IAM users or user groups for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing SWR resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust other Huawei Cloud account or cloud service to perform efficient O&M on your SWR resources.

If your Huawei Cloud account does not require individual IAM users for permissions management, you can skip this section.

SWR **system-defined policies** are preset in IAM. If these policies cannot meet your requirements, you can **create custom policies**.

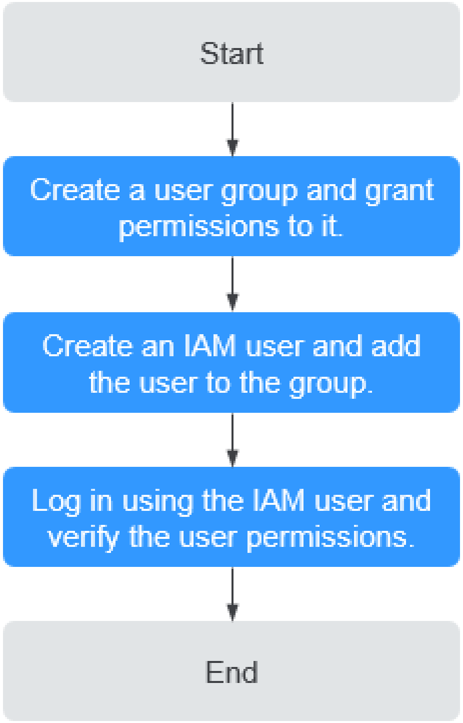
This section takes the **SWR FullAccess** policy as an example to describe how to grant permissions to an IAM user.

Prerequisites

Learn about the permissions (see **SWR Enterprise Edition Permissions**) supported by SWR Enterprise Edition and choose policies or roles as needed. For the system permissions of other services, see **System Permissions**.

Process

Figure 1-1 Process for granting SWR Enterprise Edition permissions


Start
Create a user group and grant permissions to it.
Create an IAM user and add the user to the group.
Log in using the IAM user and verify the user permissions.
End

1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and assign the **SWR FullAccess** policy to the group.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

   NOTE

   If an account is frozen (for example, due to arrears, violation, or before deregistration), you can only view and delete the account and cannot create or update the account. **Unfreeze the account** or complete real-name authentication in a timely manner.

3. **Log in** and verify permissions.

   Log in to the **SWR console** as the created user, switch to the authorized region, and verify the permissions. Click **Create Repository** in the upper right corner of the page. If you can purchase a repository of the Enterprise edition, the permissions are set successfully.

1.2 Custom Policies for SWR Enterprise Edition

Scenarios

Custom policies can be created to supplement system-defined policies. You can add actions in custom policies as needed. For details about supported actions, see **Table 1-1**.

To create a custom policy, choose either visual editor or JSON.

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a policy in the JSON format from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**.

Example Custom Policies

- Example 1: Create a policy to allow users to create, update, view, or delete a namespace.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "swr:repository:getNamespace",
                "swr:repository:listNamespaces",
                "swr:repository:createNamespace",
                "swr:repository:updateNamespace",
                "swr:repository:deleteNamespace"
            ]
        }
    ]
}
```

- Example 2:

  A policy with only Deny permissions must be used in conjunction with other policies to take effect. If the policies assigned to a user contain both Allow and Deny actions, **the Deny actions take precedence**.

  If you want to assign the **SWR FullAccess** policy to a user but do not want this user to have permission to delete repositories, create a custom policy that denies repository deletion. Then, attach both the policies to the group that the user belongs to. In this way, the user can perform all operations on repositories except deleting the repositories. The following is an example of a deny policy:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "swr:instance:delete"
            ]
        }
    ]
}
```

Common SWR Operations Supported by Each System-defined Policy

Table 1-1 SWR Enterprise Edition operations supported by system-defined policies

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Listing artifacts	swr:repository:listArtifacts	√	√	√
Querying artifact details	swr:repository:getArtifact	√	√	√
Deleting artifacts	swr:repository:deleteArtifact	√	√	×
Listing artifact accessories	swr:repository:listAccessories	√	√	√
Querying additional information about an artifact	swr:repository:getArtifactAddition	√	√	√
Querying policies of an Enterprise Edition instance	swr:instance:getPolicy	√	√	√
Updating policies of an Enterprise Edition instance	swr:instance:updatePolicy	√	×	×
Querying configurations of an Enterprise Edition instance	swr:instance:getConfigurations	√	√	√
Updating configurations of an Enterprise Edition instance	swr:instance:updateConfigurations	√	×	×
Listing the instances that use a resource	swr:instance:listResourceInstances	√	√	√
Querying the number of instances that use a resource	swr:instance:getResourceInstancesCount	√	√	√
Creating resource tags in batches	swr:instance:createResourceTags	√	×	×

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Deleting resource tags in batches	swr:instance:deleteResourceTags	√	×	×
Querying project tags	swr:instance:getProjectTags	√	√	√
Querying tags of a resource	swr:instance:getResourceTags	√	√	√
Creating an Enterprise Edition instance	swr:instance:create	√	×	×
Listing Enterprise Edition instances	swr:instance:list	√	√	√
Querying details about an Enterprise Edition instance	swr:instance:get	√	√	√
Deleting Enterprise Edition instances	swr:instance:delete	√	×	×
Querying audit logs of an Enterprise Edition instance	swr:instance:getAuditLogs	√	√	√
Querying statistics on Enterprise Edition instances	swr:instance:getStatistics	√	√	√
Listing tasks	swr:instance:listJobs	√	√	√
Querying task details	swr:instance:getJobs	√	√	√
Deleting tasks	swr:instance:deleteJob	√	×	×
Creating a namespace	swr:repository:createNamespace	√	√	×
Listing namespaces	swr:repository:listNamespaces	√	√	√
Querying namespace details	swr:repository:getNamespace	√	√	√
Modifying a namespace	swr:repository:updateNamespace	√	√	×

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Deleting namespaces	swr:repository:deleteNamespace	√	√	×
Listing artifact repositories	swr:repository:listRepositories	√	√	√
Querying details about an artifact repository	swr:repository:getRepository	√	√	√
Modifying an artifact repository	swr:repository:updateRepository	√	√	×
Deleting artifact repositories	swr:repository:deleteRepository	√	√	×
Listing artifact tags	swr:repository:listTags	√	√	√
Querying details about an artifact tag	swr:repository:getTag	√	√	√
Deleting artifact tags	swr:repository:deleteTag	√	√	×
Querying additional information about an artifact tag	swr:repository:getTagAddition	√	√	√
Creating a tag retention policy	swr:repository:createRetentionPolicy	√	√	×
Listing tag retention policies	swr:repository:listRetentionPolicies	√	√	√
Querying details about a tag retention policy	swr:repository:getRetentionPolicy	√	√	√
Modifying a tag retention policy	swr:repository:updateRetentionPolicy	√	√	×
Deleting tag retention policies	swr:repository:deleteRetentionPolicy	√	√	×
Executing tag retention policies	swr:repository:executeRetentionPolicy	√	√	×
Listing tag retention records	swr:repository:listRetentionPolicyExecutions	√	√	√

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Listing tag retention tasks	swr:repository:listRetentionPolicyExecTasks	√	√	√
Listing tag retention subtasks	swr:repository:listRetentionPolicyExecSub-Tasks	√	√	√
Creating a trigger	swr:repository:createWebhook	√	√	×
Listing triggers	swr:repository:listWebhooks	√	√	√
Querying trigger details	swr:repository:getWebhook	√	√	√
Modifying a trigger	swr:repository:updateWebhook	√	√	×
Deleting triggers	swr:repository:deleteWebhook	√	√	×
Listing triggering records	swr:repository:listWebhookJobs	√	√	√
Creating a destination registry	swr:instance:createRegistry	√	×	×
Listing destination registries	swr:instance:listRegistries	√	√	√
Querying details about a destination registry	swr:instance:getRegistry	√	√	√
Modifying a destination registry	swr:instance:updateRegistry	√	×	×
Deleting destination registries	swr:instance:deleteRegistry	√	×	×
Creating a replication policy	swr:instance:createReplicationPolicy	√	×	×
Listing replication policies	swr:instance:listReplicationPolicies	√	√	√

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Querying details about a replication policy	swr:instance:getReplicationPolicy	√	√	√
Modifying a replication policy	swr:instance:updateReplicationPolicy	√	×	×
Deleting replication policies	swr:instance:deleteReplicationPolicy	√	×	×
Executing replication policies	swr:instance:executeReplicationPolicy	√	√	×
Stopping replication tasks	swr:instance:stopReplicationPolicyExecution	√	×	×
Listing replication records	swr:instance:listReplicationPolicyExecutions	√	√	√
Listing replication tasks	swr:instance:listReplicationPolicyExecTasks	√	√	√
Listing replication subtasks	swr:instance:listReplicationPolicyExecSubTasks	√	√	√
Creating a sign policy	swr:repository:createSignPolicy	√	√	×
Listing sign policies	swr:repository:listSignPolicies	√	√	√
Querying details about a sign policy	swr:repository:getSignPolicy	√	√	√
Modifying a sign policy	swr:repository:updateSignPolicy	√	√	×
Deleting sign policies	swr:repository:deleteSignPolicy	√	√	×
Executing sign policies	swr:repository:executeSignPolicy	√	√	×
Listing signing records	swr:repository:listSignPolicyExecutions	√	√	√
Listing signing tasks	swr:repository:listSignPolicyExecTasks	√	√	√

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Listing signing subtasks	swr:repository:listSignPolicyExecSubTasks	√	√	√
Creating a scan policy	swr:repository:createScanPolicy	√	√	×
Listing scan policies	swr:repository:listScanPolicies	√	√	√
Querying details about a scan policy	swr:repository:getScanPolicy	√	√	√
Modifying a scan policy	swr:repository:updateScanPolicy	√	√	×
Deleting scan policies	swr:repository:deleteScanPolicy	√	√	×
Executing scan policies	swr:repository:executeScanPolicy	√	√	×
Listing scanning records	swr:repository:listScanPolicyExecutions	√	√	√
Listing scanning tasks	swr:repository:listScanPolicyExecTasks	√	√	√
Creating a block policy	swr:repository:createBlockPolicy	√	√	×
Listing block policies	swr:repository:listBlockPolicies	√	√	√
Querying details about a block policy	swr:repository:getBlockPolicy	√	√	√
Modifying a block policy	swr:repository:updateBlockPolicy	√	√	×
Listing blocking records	swr:repository:listBlockPolicyRecords	√	√	√
Updating the whitelist for public network access	swr:instance:updateEndpointPolicy	√	×	×
Updating the whitelist status for public network access	swr:instance:updateEndpointPolicyStatus	√	×	×

Operation	Action	SWR FullAccess	SWR OperateAccess	SWR ReadOnlyAccess
Querying the whitelist for public network access	swr:instance:getEndpointPolicy	√	√	√
Allowing a connection from the intranet	swr:instance:createInternalEndpoint	√	×	×
Querying details about an allowed connection from the intranet	swr:instance:getInternalEndpoint	√	√	√
Denying a connection from the intranet	swr:instance:deleteInternalEndpoint	√	×	×
Listing allowed connections from the intranet	swr:instance:listInternalEndpoints	√	√	√
Uploading artifacts	swr:repository:uploadArtifact	√	√	×
Downloading artifacts	swr:repository:downloadArtifact	√	√	√
Creating a temporary access credential	swr:instance:createTempCredential	√	√	√
Creating a long-term access credential	swr:instance:createLTCredential	√	×	×
Enabling or disabling long-term access credentials	swr:instance:updateLTCredential	√	×	×
Listing long-term access credentials	swr:instance:listLTCredentials	√	√	√
Deleting long-term access credentials	swr:instance:deleteLTCredential	√	×	×

1.3 SWR Enterprise Edition Resources

A resource is an object that exists within a service. In SWR Enterprise Edition, resources include repositories, instances, charts. When creating a policy, you can select a resource by specifying its path.

Table 1-2 SWR resources and their paths

Resource	Resource Name	Path
repository	Image repository	[Format] SWR:*:*:repository:*image repository name* The first * is **regionid**, and the second * is **domainid**. [Note] For image repository resources, IAM automatically generates the resource path prefix (**SWR:*:*:repository:**). For the path of a specific image repository, add the image repository name to the end. You can also use a wildcard character (*) to indicate any image repository. Example: **SWR:*:*:repository/*** indicates any image repository. swr:*:*:repository:test/nginx*: image repository whose name starts with **nginx** in the **test** namespace swr:*:*:repository:test/nginx: image repository whose name starts with **nginx** in the **test** namespace

Resource	Resource Name	Path
instance	SWR Enterprise Edition instance	[Format] SWR:*:*:instance: SWR Enterprise Edition instance The first * is **regionid**, and the second * is **domainid**. [Notes] For SWR Enterprise Edition instances, IAM automatically generates the resource path prefix (**SWR:*:*:instance:**). For the path of a specific SWR Enterprise Edition instance, add the *instance name* to the end. You can also use a wildcard character (*) to indicate any instance. Example: SWR:*:*:instance:example-instance indicates the SWR Enterprise Edition instance named **example-instance**.
chart	Chart repository	[Format] SWR:*:*:chart:*chart repository name* The first * is **regionid**, and the second * is **domainid**. [Notes] For chart repository resources, IAM automatically generates the resource path prefix (**SWR:*:*:chart:**). For the path of a specific chart repository, add the *chart repository name* to the end. You can also use a wildcard character (*) to indicate any chart repository. Example: SWR:*:*:chart:* indicates any chart repository.

For example, to allow users to perform operations only on the instance named **example-instance**, you can define the YAML file as follows:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "swr:instance:*"
            ],
            "Resource": [
                "SWR:*:*:instance:example-instance"
            ]
        }
    ]
}
```

1.4 Tag-based Fine-Grained Authorization

Scenarios

After creating a custom policy for the SWR Enterprise Edition on the IAM console, you can add tags for namespaces and repositories. Use policies and tags together can implement fine-grained authorization on resources of the SWR Enterprise Edition, ensuring controllable and secure resource permissions.

Prerequisites

You have created **namespace tags**.

Procedure

Step 1 **Create one or more policies** on the IAM console.

For example, you can create a policy named **policy77r463**. **Table 1-3** describes the parameters.

Table 1-3 Example policy configuration

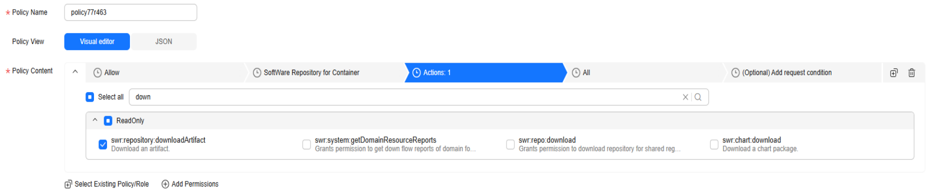
Parameter	Description	Example Value
Policy type	You can select **Allow** or **Deny**.	Allow
Cloud services	Cloud services that the current policy will be applied to	SWR
Action	Actions that the current policy will be applied to. You can select one or more actions.	swr:repository:downloadArtifact
Resource type	● You can select **Specific** or **All**. ● For details about specific resources, see **Resource Type**.	If you select **Specific**, click **Specify resource path** and configure the resource path **SWR:*:*:repository:*/{namespace-name}**.

Parameter	Description	Example Value
Request condition	- Tag of the current policy. A tag is a key-value pair. - For details about request conditions, see **Request Conditions**.	This policy is created for SWR, so select **Service-level condition keys** for **Condition Key**. Configure the parameters as follows: TagKey: test Operator: StringEquals Value: aaa

The following policy is in JSON format:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "swr:repository:downloadArtifact"
            ],
            "Resource": [
                "SWR:*:*:repository:*/{namespace-name}"
            ],
            "Condition": {
                "StringEquals": {
                    "g:ResourceTag/test": [
                        "aaa"
                    ]
                }
            }
        }
    ]
}
```

Figure 1-2 Creating a policy



* Policy Name: policy77r463

Policy View: Visual editor | JSON

* Policy Content: Allow | SoftWare Repository for Container | Actions: 1 | All | (Optional) Add request condition

■ Select all: down

■ ReadOnly

☑ swr:repository:downloadArtifact
Download an artifact.

□ swr:system:getDomainResourceReports
Grants permission to get down flow reports of domain fo...

□ swr:repo:download
Grants permission to download repository for shared reg...

□ swr:chart:download
Download a chart package.

This policy applies to SWR. You can attach this policy to a user or user group of this service. After the policy is applied, the user or user group can download the artifacts from a repository in the **namespace-name** namespace with the **test=aaa** tag.

NOTE

If a user or user group wants to download an image from the image repository, SWR Enterprise Edition will extract the tag of the user or user group and verifies it with **test=aaa**. If the tag matches, the user or user group is allowed to perform the operation. Otherwise, the operation will fail.

Step 2 Attach **policy77r463** generated in **Step 1** to a user or user group. For details, see **Creating a User Group and Granting Permissions**. Add the **test=aaa** tag to a namespace, for example, the **{namespace-name}** namespace. For details, see **Adding a Tag to a Namespace**.

Step 3 Verify that the user or user group in **Step 2** can download artifacts in the **{namespace-name}** namespace.

----End

1.5 SWR Custom Policies

Custom policies can be created to supplement system-defined policies of SWR.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

  For details, see **Creating a Custom Policy**. This section illustrates common custom policies.

Example SWR custom policies

- Example 1: Allowing a user to upload and download images in the **test-swr** SWR Enterprise Edition instance in the **test-namespace** namespace

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "swr:repository:downloadArtifact",
        "swr:repository:uploadArtifact"
      ],
      "Resource": [
        "swr:*:*:repository:test-swr/test-namespace/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "swr:instance:createTempCredential"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

- Example 3: Denying image replication from region A to region B

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "swr:instance:createReplicationPolicy"
      ],
      "Resource": [
        "swr:*:*:instance:*"
      ],
      "Condition": {
        "StringEquals": {
```

```
     "swr:TargetRegion": [
      "${region-b}"
     ],
     "swr:SourceRegion": [
      "${region-a}"
     ]
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": [
    "swr:instance:createReplicationPolicy"
   ],
   "Resource": [
    "swr:*:*:instance:*"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "swr:SourceUrls": [
      "All repository addresses in region a"
     ],
     "swr:TargetUrls": [
      "All repository addresses in region b"
     ]
    }
   }
  }
 ]
}
```

2 Control Policies Supported by SWR

2.1 Control Policy Overview

SWR supports multiple control policies, including IAM-based access control, SCP-based access control, RCP-based access control, NCP-based access control, and VPC Endpoint policy-based access control. You can use different control policies based on security requirements. The following describes several policies.

IAM-based Access Control

Identity and Access Management (IAM) provides permissions management for secure access to your Huawei Cloud services and resources. For details about how to use IAM to control access to SWR, see **IAM-based Permissions Management**.

SCP-based Access Control

Service Control Policies (SCPs) are guardrail policies provided by Organizations. The management account can use SCP to limit the permissions that can be assigned to member accounts in an organization. You can attach an SCP to your organization, OUs, or member accounts. Any SCP attached to an organization or OU affects all the accounts within the organization or under the OU. For details, see **SCP Introduction**.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR.

RCP-based Access Control

Resource Control Policies (RCPs) are guardrail policies provided by Organizations. RCPs limit the maximum permissions allowed for a resource. Access to resources of an organization member account is restricted by RCPs. An organization administrator can set RCPs in an organization to meet the security and compliance requirements for access control of resources in organization member accounts.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR.

NCP-based Access Control

Network Control Policies (NCPs) are guardrail policies provided by Organizations. An NCP policy limits the maximum permissions allowed for access from a VPC endpoint. NCP policies restrict requests initiated from a VPC endpoint created by the member accounts of an organization. An organization administrator can set NCPs in an organization to meet the security and compliance requirements for controlling the access initiated from the VPC endpoints created by member accounts of an organization.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR.

VPC Endpoint Policy-based Access Control

VPC endpoint policies are a type of resource-based policies. You can configure a policy to control which principals can use the VPC endpoint to access VPC endpoint services. For details, see **Managing the Policy of a VPC Endpoint**.

Virtual Private Cloud (VPC) is used to control the network border security. If the API access point of a resource is within the VPC of your account, the access is within the VPC and security is controllable (the VPC can be considered as a network security domain). If the API access point is on a public network, the network attack surface is large and security is hard to control.

NOTE

After a control policy is configured, anonymous download of public images is also controlled by the control policy.

2.2 SCPs

Example: Forbid an account to download images from an SWR Enterprise Edition instance in a namespace.

The following describes how to configure an SCP to forbid an account to download images from the SWR Enterprise Edition instance named **test-swr** in the **test-namespace** namespace.

Configuration method

Step 1 Log in to the management console as the organization administrator or using the management account, and navigate to the Organizations console.

Step 2 On the **Policies** page, click **Service control policies** and then **Create Policy**.

Step 3 Enter the policy name and description. On the left of the policy content, you can copy and paste the following JSON policy content: Click **Save**.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "swr:*:*:repository:test-swr/test-namespace"
      ]
    }
  ]
}
```

Step 4 Bind the policy to an OU or account of the organization to apply the policy.

1. Log in to Huawei Cloud as the organization administrator or using the management account, navigate to the Organizations console, and access the **Organization** page.
2. Select the OU or account you want to attach the SCP to.
3. On the details page, click the **Policies** tab. On the displayed tab, expand **Service control policies** and click **Attach**.
4. Select the policy to be added and enter "Confirm" in the text box. Then, click **Attach**.

----End

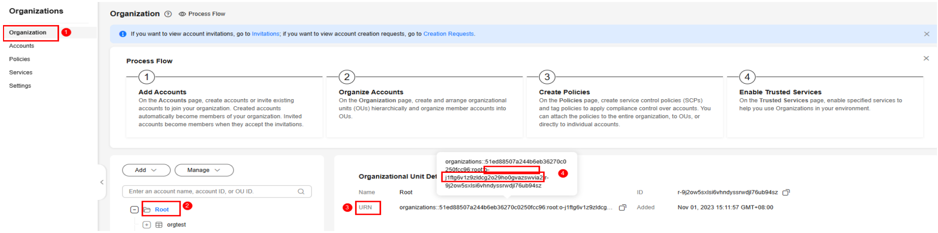
2.3 RCPs

Example 1: Images in an organization can only be downloaded by accounts in that organization.

The following policy means that images in the OU or account bound to the policy cannot be downloaded by accounts outside the **o-j1ftg6v1z9zldcg2o29ho0gvazswvia2** organization. They can only be downloaded by accounts in the organization.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR. To obtain the ID of the organization, follow the steps in the figure.



If you want to view account invitations, go to Invitations; if you want to view account creation requests, go to Creation Requests.

1 Add Accounts
On the Accounts page, create accounts or invite existing accounts to join your organization. Created accounts automatically become members of your organization. Invited accounts become members when they accept the invitations.

2 Organize Accounts
On the Organization page, create and arrange organizational units (OUs) hierarchically and organize member accounts into OUs.

3 Create Policies
On the Policies page, create service control policies (SCPs) and tag policies to apply compliance control over accounts. You can attach the policies to the entire organization, to OUs, or directly to individual accounts.

4 Enable Trusted Services
On the Trusted Services page, enable specified services to help you use Organizations in your environment.

Enter an account name, account ID, or OU ID.

organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg2o29ho0gvazswvia2/r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

Name: Root
ID: r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz
URN: organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg...
Added: Nov 01, 2023 15:11:57 GMT+08:00

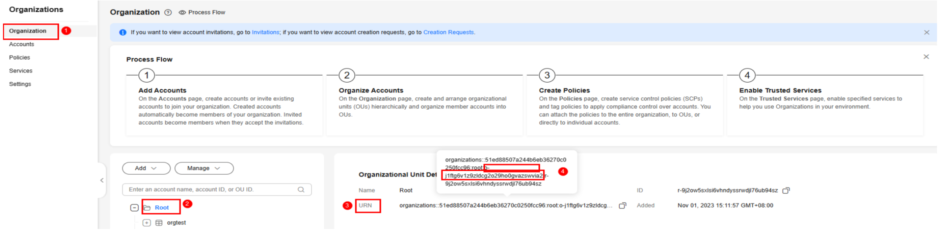
```
{
 "Version": "5.0",
 "Statement": [
  {
    "Effect": "Deny",
    "Principal": "*",
    "Action": [
      "swr:repository:downloadArtifact"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "StringNotEquals": {
        "g:PrincipalOrgId": [
          "o-j1ftg6v1z9zldcg2o29ho0gvazswvia2"
        ]
      },
      "Bool": {
        "g:PrincipalIsService": [
          "false"
        ]
      }
    }
  }
 ]
}
```

Example 2: Images in an organization can only be downloaded by accounts in that organization, except public images.

The following policy means that private images in the OU or account bound to the policy cannot be downloaded by accounts outside the **o-j1ftg6v1z9zldcg2o29ho0gvazswvia2** organization. They can only be downloaded by accounts in the organization. Public images can be downloaded by any account.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR. To obtain the ID of the organization, follow the steps in the figure.



If you want to view account invitations, go to Invitations; if you want to view account creation requests, go to Creation Requests.

Add Accounts
On the Accounts page, create accounts or invite existing accounts to join your organization. Created accounts automatically become members of your organization. Invited accounts become members when they accept the invitations.

Organize Accounts
On the Organization page, create and arrange organizational units (OUs) hierarchically and organize member accounts into OUs.

Create Policies
On the Policies page, create service control policies (SCPs) and tag policies to apply compliance control over accounts. You can attach the policies to the entire organization, to OUs, or directly to individual accounts.

Enable Trusted Services
On the Trusted Services page, enable specified services to help you use Organizations in your environment.

organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg2o29ho0gvazswvia2/r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

ID r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

URN organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg...

Added Nov 01, 2023 15:11:57 GMT+08:00

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringNotEquals": {
          "g:PrincipalOrgId": [
            "o-j1ftg6v1z9zldcg2o29ho0gvazswvia2"
          ]
        },
        "Bool": {
          "g:PrincipalIsService": [
            "false"
          ],
          "swr:RepositoryIsPublic": [
            "false"
          ]
        }
      }
    }
  ]
}
```

NOTE

The configuration method is the same as that described in **SCPs**.

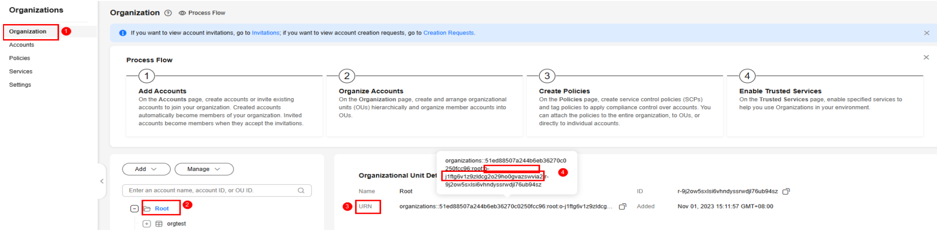
2.4 NCPs

Example 1: Accounts in an organization can only download private images in that organization through VPC Endpoint, but they can download any public images.

The following policy means that images in the OU or account bound to the policy cannot be downloaded by accounts outside the **o-j1ftg6v1z9zldcg2o29ho0gvazswvia2** organization through VPC Endpoint. They can only be downloaded by accounts in the organization.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR. To obtain the ID of the organization, follow the steps in the figure.



If you want to view account invitations, go to Invitations; if you want to view account creation requests, go to Creation Requests.

1 Add Accounts
On the Accounts page, create accounts or invite existing accounts to join your organization. Created accounts automatically become members of your organization. Invited accounts become members when they accept the invitations.

2 Organize Accounts
On the Organization page, create and arrange organizational units (OUs) hierarchically and organize member accounts into OUs.

3 Create Policies
On the Policies page, create service control policies (SCPs) and tag policies to apply compliance control over accounts. You can attach the policies to the entire organization, to OUs, or directly to individual accounts.

4 Enable Trusted Services
On the Trusted Services page, enable specified services to help you use Organizations in your environment.

organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg2o29ho0gvazswvia2:r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

Name	Root	ID	r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz
URN	organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg...	Added	Nov 01, 2023 15:11:57 GMT+08:00

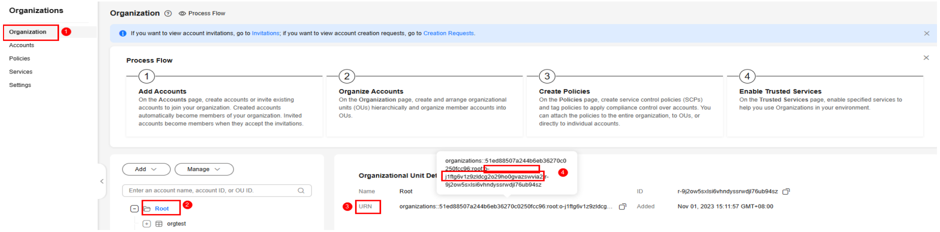
```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "g:PrincipalIsService": [
            "false"
          ]
        },
        "StringNotEquals": {
          "g:ResourceOrgId": [
            "o-j1ftg6v1z9zldcg2o29ho0gvazswvia2"
          ]
        }
      }
    }
  ]
}
```

Example 2: Accounts in an organization can only download private images in that organization through VPC Endpoint, and they can download any public images.

The following policy means that private images in the OU or account bound to the policy cannot be downloaded by accounts outside the **o-j1ftg6v1z9zldcg2o29ho0gvazswvia2** organization through VPC Endpoint. They can only be downloaded by accounts in the organization. Public images can be downloaded by any account.

NOTE

The organization here refers to the organization in the Organizations service, not the organization in SWR. To obtain the ID of the organization, follow the steps in the figure.



If you want to view account invitations, go to Invitations; if you want to view account creation requests, go to Creation Requests.

Process Flow

1 Add Accounts
On the Accounts page, create accounts or invite existing accounts to join your organization. Created accounts automatically become members of your organization. Invited accounts become members when they accept the invitations.

2 Organize Accounts
On the Organization page, create and arrange organizational units (OUs) hierarchically and organize member accounts into OUs.

3 Create Policies
On the Policies page, create service control policies (SCPs) and tag policies to apply compliance control over accounts. You can attach the policies to the entire organization, to OUs, or directly to individual accounts.

4 Enable Trusted Services
On the Trusted Services page, enable specified services to help you use Organizations in your environment.

organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg2o29ho0gvazswvia2:r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

URN: organizations::51ed88507a244b6eb36270c0250fcc96:root:o-j1ftg6v1z9zldcg...

ID: r-9j2ow5sxlsi6vhndyssrwdjl76ub94sz

Added: Nov 01, 2023 15:11:57 GMT+08:00

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "g:PrincipalIsService": [
            "false"
          ],
          "swr:RepositoryIsPublic": [
            "false"
          ]
        },
        "StringNotEquals": {
          "g:ResourceOrgId": [
            "o-j1ftg6v1z9zldcg2o29ho0gvazswvia2"
          ]
        }
      }
    }
  ]
}
```

NOTE

The configuration method is the same as that described in **SCPs**.

2.5 VPC Endpoint Policies

Images can be uploaded to and downloaded from SWR Enterprise Edition through VPC endpoints. You can configure policies to control image upload and download. For details about how to create a VPC endpoint, see **Access Through VPC Endpoint**. For details, see **Managing the Policy of a VPC Endpoint**.

Example 1: Configure a VPC endpoint policy to allow the upload or download of only specified images.

The following policy only allows servers in VPC1 to upload images to or download images from the SWR Enterprise Edition instance named **test-swr** in the **test-namespace** namespace.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Action": [
        "swr:repository:uploadArtifact",
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "swr:*:*:repository:test-swr/test-namespace/*"
      ],
      "Effect": "Allow",
      "Principal": "*"
    }
  ]
}
```

Example 2: Configure a VPC endpoint policy to allow the download of only specified private images and all public images.

The following policy only allows servers in VPC1 to download images from the SWR Enterprise Edition instance named **test-swr** in the **test-namespace** namespace, and public images are not restricted.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "swr:*:*:repository:test-swr/test-namespace/*"
      ],
      "Effect": "Allow",
      "Principal": "*"
    },
    {
      "Action": [
        "swr:repository:downloadArtifact"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Principal": "*",
      "Condition": {
        "Bool": {
          "swr:RepositoryIsPublic": [
            "true"
          ]
        }
      }
    }
  ]
}
```

3 Repository Management

3.1 Image Repository Overview

Scenarios

An image repository manages container images. You can push and pull images to and from a repository and view the image build history.

Prerequisites

Before using an image repository, ensure that:

- You have **purchased a repository**.
- You have access to repositories. For details, see **Access Control Overview**.
- You have **created an access credential**.

Pushing an Image

Step 1 Prepare a server that meets the following requirements:

- The container engine version must be later than 1.13.1.
- The server can be used within the network access range defined in **Access Control**.

Step 2 Log in to the server as **root**.

Step 3 Use the access credential obtained in **Access Credentials** to log in to the registry and access a repository.

The message **Login Succeeded** will be displayed upon a successful login.

Step 4 Run the following command to tag the image:

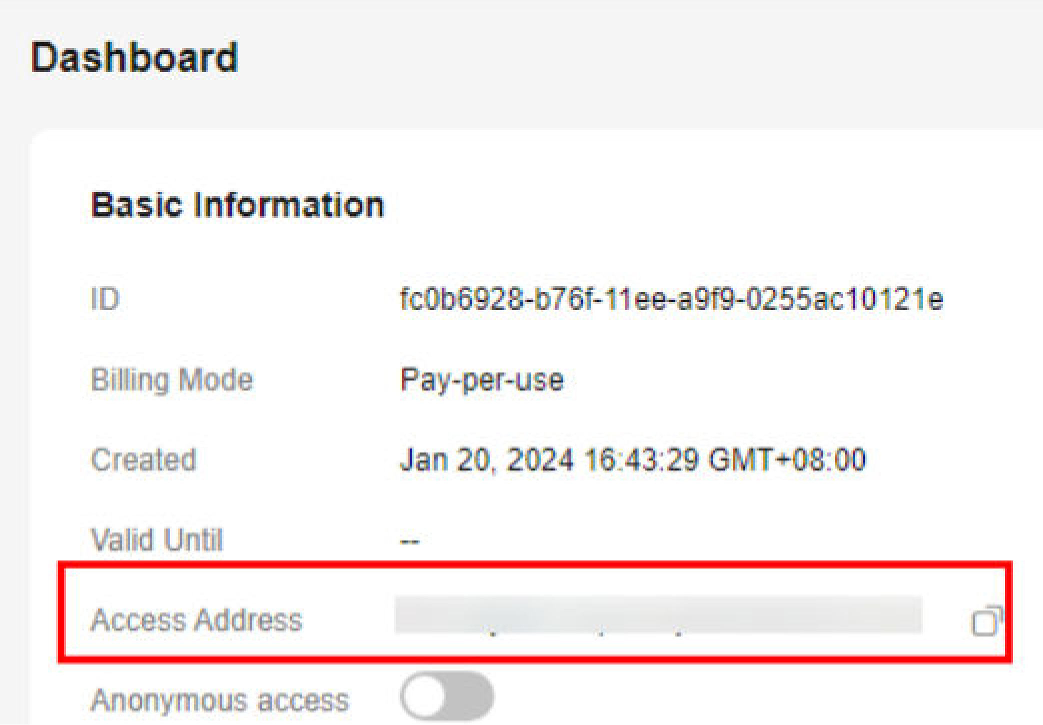
docker tag *[image-name-1:tag-1] [repository-address]/ [namespace-name]/ [image-name-2:tag-2]*

In the preceding command:

- *[image-name-1:tag-1]*: name and tag of the image to be pushed.
- *[repository-address]*: address for accessing the repository where the image is stored. To obtain the address, perform the following operations:

  Log in to the **SWR console**. In the upper left corner, switch to your region. Click the repository name. On the **Dashboard** page, obtain the access address, as shown in **Figure 3-1**.

  Figure 3-1 Repository access address

  Dashboard

  Basic Information

  ID fc0b6928-b76f-11ee-a9f9-0255ac10121e

  Billing Mode Pay-per-use

  Created Jan 20, 2024 16:43:29 GMT+08:00

  Valid Until --

  Access Address [redacted]

  Anonymous access

- *[namespace-name]*: namespace you created in **Creating a Namespace**.
- *[image-name-2:tag-2]*: new name and tag for the image.

Example:

docker tag nginx:latest test-01-2v8iom.swr.cn-east-3.myhuaweicloud.com/library/nginx:1.1.1

Step 5 Push the image to a repository.

docker push *[repository-address]***/** *[namespace-name]***/** *[image-name:tag-name]*

Example:

docker push test-01-2v8iom.swr.cn-east-3.myhuaweicloud.com/library/nginx:1.1.1

The following information will be returned upon a successful push:

```
fbce26647e70: Pushed
fb04ab8effa8: Pushed
8f736d52032f: Pushed
009f1d338b57: Pushed
678bbd796838: Pushed
d1279c519351: Pushed
f68ef921efae: Pushed
v1: digest: sha256:0cdfc7910db531bfa7726de4c19ec556bc9190aad9bd3de93787e8bce3385f8d size: 1780
```

To view the image information, go to the repository details page and choose **Image Repositories** from the navigation pane.

NOTE

After an image is pushed, you can use it to create a workload on the CCE console.

----End

Obtaining an Image Pull Address

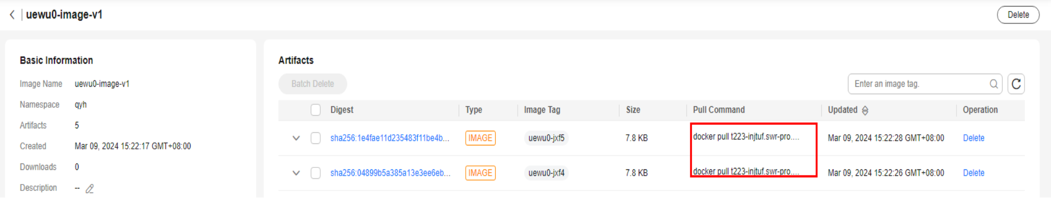
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 In the navigation pane, choose **Image Repositories**.

Step 3 Click the name of the target image to go to the image details page.

Step 4 Locate a desired image tag and obtain the image pull command in the **Pull Command** column.

Figure 3-2 Image pull command



< | uewu0-image-v1 Delete

Basic Information

Image Name	uewu0-image-v1
Namespace	qyh
Artifacts	5
Created	Mar 09, 2024 15:22:17 GMT+08:00
Downloads	0
Description	--

Artifacts

Batch Delete   Enter an image tag.

	Digest	Type	Image Tag	Size	Pull Command	Updated	Operation
□	sha256:1e4fae11d235483f11be4b...	IMAGE	uewu0-jxf5	7.8 KB	docker pull t223-injtuf.swr-pro....	Mar 09, 2024 15:22:28 GMT+08:00	Delete
□	sha256:04899b5a385a13e3ee6eb...	IMAGE	uewu0-jxf4	7.8 KB	docker pull t223-injtuf.swr-pro....	Mar 09, 2024 15:22:26 GMT+08:00	Delete

----End

Other Operations

- Searching for an image

  Search for an image by namespace or name.

  Figure 3-3 Searching for an image

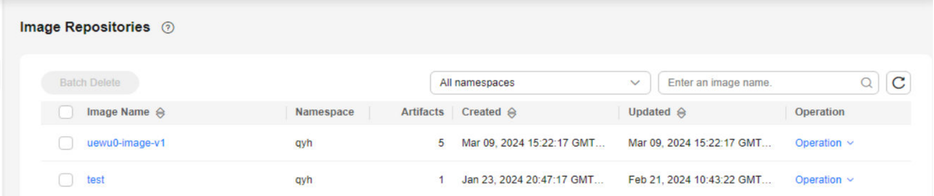
  

  Image Repositories

  Batch Delete   All namespaces   Enter an image name.

□	Image Name	Namespace	Artifacts	Created	Updated	Operation
□	uewu0-image-v1	qyh	5	Mar 09, 2024 15:22:17 GMT...	Mar 09, 2024 15:22:17 GMT...	Operation
□	test	qyh	1	Jan 23, 2024 20:47:17 GMT...	Feb 21, 2024 10:43:22 GMT...	Operation

- Deleting an image

  To delete an image, locate the image and click **Delete**. To avoid deleting important data by mistake, you need to enter **DELETE** to confirm the deletion.

  CAUTION

  Deleting an image will delete all its tags.

- Deleting an image tag

  To delete an image tag, click the desired image name to go to its details page. Locate the target image tag, and click **Delete**. To avoid deleting important data by mistake, you need to enter **DELETE** to confirm the deletion.

Follow-up Operations

After images are pushed to a repository, you can:

- Configure an image signing policy so that images can be automatically signed. For details, see **Signing an Image**.
- Configure an image replication policy so that images can be replicated to another registry automatically. For details, see **Replicating an Image to Other Regions**.
- Configure an image retention policy to automatically delete unnecessary images. For details, see **Image Retention**.

3.2 Purchasing a Repository

Scenarios

To use SWR Enterprise Edition, you first need to buy a repository. SWR Enterprise Edition provides enterprise-class, secure hosting services for container images and other cloud native artifacts that comply with the Open Container Initiative (OCI) specifications.

⚠ CAUTION

- By default, access to new repositories is blocked to ensure data security.
- Repositories are regional resources. If you need to use a repository in multiple regions, purchase it in each region. SWR Enterprise Edition is only available in regions CN East-Shanghai1, CN North-Ulanqab1, CN North-Beijing4, CN South-Guangzhou, CN Southwest-Guiyang1, CN East 2, CN Northwest-Karamay, CN-Hong Kong, AP-Singapore, AF-Johannesburg, TR-Istanbul, and AP-Jakarta.

Prerequisites

- You can access the Virtual Private Cloud (VPC), Object Storage Service (OBS), Key Management Service (KMS), and VPC Endpoint (VPCEP) services.
- SWR Enterprise Edition has been authorized to access VPC, OBS, and other related resources.

Procedure

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 In the upper right corner, click **Create Repository**. Configure the parameters as follows:

- **Billing Mode**: Only **pay-per-use** is available.
- **Project**: Select the region or project where the repository is. The region or project cannot be changed after repository purchase.
- **Repository Name**: Enter a repository name. The name will be used as part of the access address of the repository and cannot be changed after repository purchase.
- **Package Specifications**: Select specifications for the repository. The repository capabilities and quotas vary with different specifications.
- **VPC**: Select the VPC where the repository is. If there is no VPC available, create one by referring to **Creating a VPC**.
- **Subnet**: Select the subnet where the repository is.
- **Custom OBS Bucket**: Enabling this option allows you to select an OBS bucket from the list. You are advised to select a 3-AZ bucket for high availability.
- **OBS Bucket Encryption** (encryption at rest): Key Management Service (KMS) keys are used to automatically encrypt images uploaded to OBS buckets. This will improve data security.

  NOTE

  OBS bucket encryption may affect repository performance.
- **SM Encryption**: If you enable this option, SM algorithms will be used to secure image push, image signatures, and login passwords.
- **Tag**: Tags can be used to categorize cloud resources for easier resource management.
- **Description**: Describe the repository.

Step 3 Click **Next**.

Step 4 On the repository management page, check the creation progress. If the repository status is **Running**, the repository creation is complete.

NOTE

If the repository stays **Creating** or is not displayed in the list, click **Operation Records** in the upper left corner to view the failure cause. If the fault cannot be located, **submit a service ticket**.

----End

3.3 Deleting a Repository

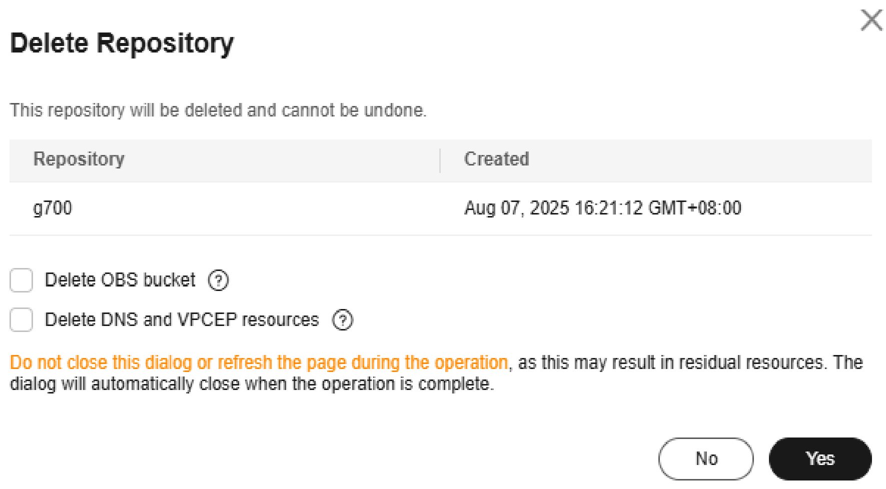
Scenarios

If you no longer need a repository, you can delete it. Deleted repositories cannot be recovered.

Procedure

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Locate the repository and click **Delete**. You can choose whether to delete the OBS bucket and DNS and VPC Endpoint resources associated with this repository.



Delete Repository

This repository will be deleted and cannot be undone.

Repository	Created
g700	Aug 07, 2025 16:21:12 GMT+08:00

□ Delete OBS bucket ⓘ

□ Delete DNS and VPCEP resources ⓘ

Do not close this dialog or refresh the page during the operation, as this may result in residual resources. The dialog will automatically close when the operation is complete.

Step 3 Click **Yes**.

----End

⚠ CAUTION

- A deleted repository cannot be restored.
- Do not close the **Delete Repository** dialog box or refresh the page during the deletion, or residual resources may be left. The dialog box will be automatically closed when the deletion is complete.

3.4 Anonymous Access to a Repository

Scenarios

If you enable anonymous access to a repository and set a namespace of the repository to public, all images in the namespace are public. These images can be downloaded more conveniently.

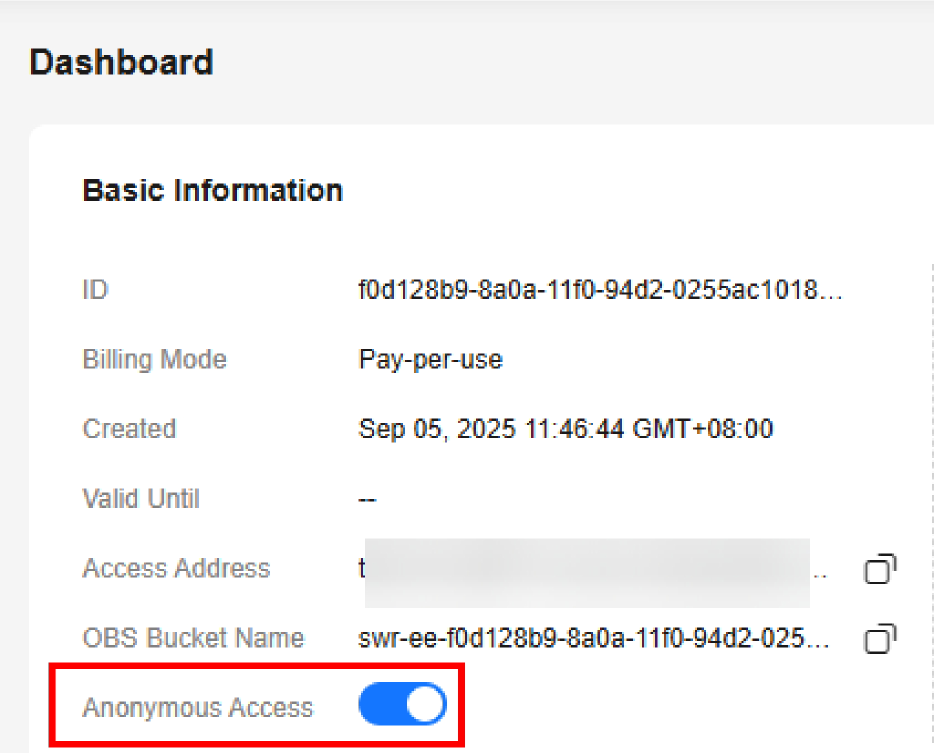
- If anonymous access is enabled, you can download public images without logging in to Docker.
- If anonymous access is disabled,
  - Public images: You can download public images after logging in to Docker. No additional permissions are required.
  - Private images: You can download private images after you log in to Docker and are granted the download permission (the corresponding action is **swr:repository:downloadArtifact**). For details, see **Table 1 SWR Enterprise Edition operations supported by system-defined policies**.

Procedure

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Click the repository name to go to the **Dashboard** page.

Step 3 Enable anonymous access in the **Basic Information** area.



Dashboard

Basic Information

ID f0d128b9-8a0a-11f0-94d2-0255ac1018...

Billing Mode Pay-per-use

Created Sep 05, 2025 11:46:44 GMT+08:00

Valid Until --

Access Address t[redacted]..

OBS Bucket Name swr-ee-f0d128b9-8a0a-11f0-94d2-025...

Anonymous Access ●

----End

3.5 Tag Management

3.5.1 Tag Overview

What Is a Tag?

A tag is an identifier you assign to a cloud resource. When you have many cloud resources, you can use tags to categorize them in different ways (for example, by purpose, owner, or environment).

In SWR Enterprise Edition, you can use tags to identify repositories or namespaces so that you can find and manage them easier.

Application Scenarios

You can use tags to facilitate the following operations:

- **Central management of resources**

  If you have a lot of cloud resources, you can use tags to quickly identify resources of the same type to check, modify, or delete them.

- **Resource migration**

You can define a tag to identify the resources to be migrated. This improves migration efficiency and avoids errors caused by repeatedly creating tags.

- **Custom billing**

  In a billing system, to collect and analyze bills faster and more precisely, you can query resources with specific tags.

Naming Rules

Each tag consists of a key and a value. For each resource, their tag keys must be unique, and each tag key can have only one tag value. If the tag value you add is the same as an existing one for the resource, the new value overwrites the old one.

Table 3-1 Key and value

Parameter	Rule	Example
Key	● Cannot be omitted. ● Cannot start with **_sys_**. ● Contains 1 to 128 characters. ● Consists of letters, digits, underscores (_), and hyphens (-). ● Can contain UTF-8 letters, digits, spaces, and the following characters: _.:=+-@	Test Department
Value	● Can be omitted. ● Cannot be empty or null for a predefined tag. ● Contains 0 to 255 characters. ● Consists of letters, digits, underscores (_), and hyphens (-). ● Can contain UTF-8 letters, digits, spaces, and the following characters: _.:/=+-@	Shanghai

3.5.2 Adding a Repository Tag

Constraints

Table 3-2 Maximum number of tags allowed for a single repository

Item	Quota
Number of tags for a single repository	20

Adding a Tag When Purchasing a Repository

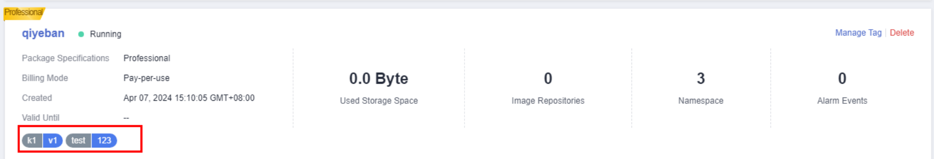
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the upper right corner, click **Create Repository**.

Step 2 On the repository purchase page, click + to add a tag. Enter a key and value as instructed in **Naming Rules**.

Step 3 Click **Next**.

Step 4 After the purchase is complete, check the new repository with tags on the repository management page.

Figure 3-4 Repository with tags



qiyeban

Package Specifications Professional
Billing Mode Pay-per-use
Created Apr 07, 2024 15:10:05 GMT+08:00
Valid Until --

0.0 Byte Used Storage Space
0 Image Repositories
3 Namespace
0 Alarm Events

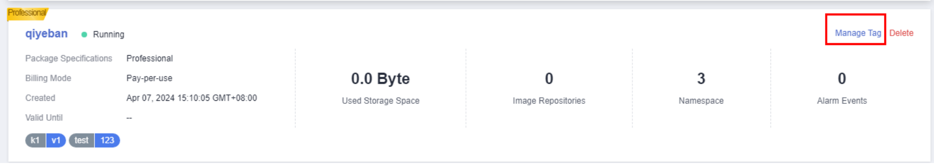
----End

Adding a Tag After Purchasing a Repository

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 On the repository management page, locate the repository you want to add a tag for and click **Manage Tag**.

Figure 3-5 Adding a tag for an existing repository



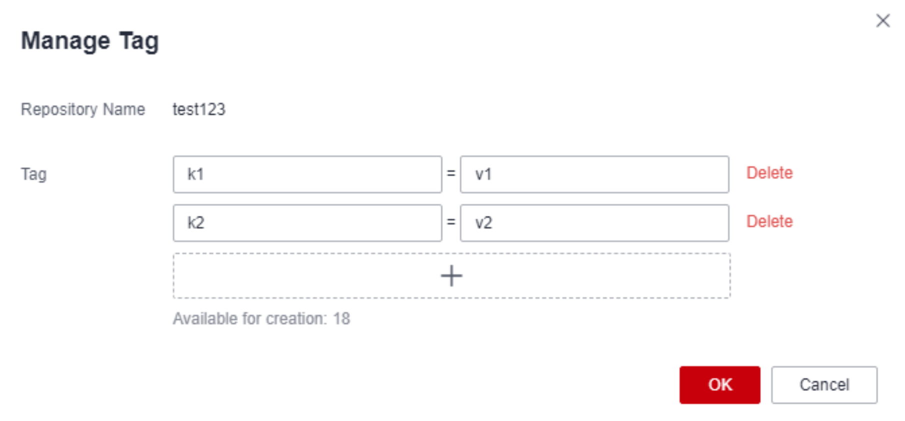
qiyeban

Package Specifications Professional
Billing Mode Pay-per-use
Created Apr 07, 2024 15:10:05 GMT+08:00
Valid Until --

0.0 Byte Used Storage Space
0 Image Repositories
3 Namespace
0 Alarm Events

Step 3 In the **Manage Tag** dialog box, click + . Enter a key and a value.

Figure 3-6 Adding a tag



Manage Tag

Repository Name test123

Tag			
k1	=	v1	Delete
k2	=	v2	Delete

----End

3.5.3 Deleting a Repository Tag

You can delete tags on the SWR or TMS console. There are two methods for you to delete tags:

- **Deleting a Tag on the SWR Console**
- **Deleting Tags in a Batch on the TMS Console**

Deleting a Tag on the SWR Console

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 On the repository management page, locate the repository whose tag needs to be deleted and click **Manage Tag**.

Step 3 In the **Manage Tag** dialog box, locate the tag to be deleted and click **Delete**.

----End

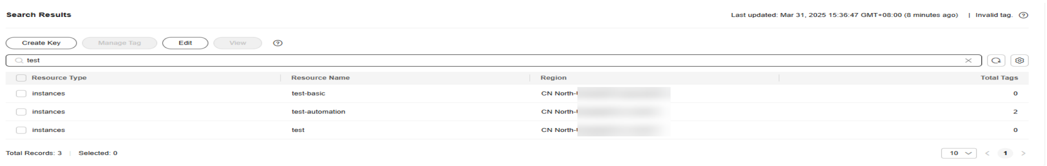
Deleting Tags in a Batch on the TMS Console

Step 1 Log in to the TMS console.

Step 2 Choose **Resource Tags** > **Tag Management**, select the target region, set **Resource Type** to **SWR**, and click **Search**. All SWR resources in this region will be returned.

Step 3 Locate the repositories whose tags need to be deleted. Click **Manage Tag** above the list.

Figure 3-7 Tag search result



Search Results

Last updated: Mar 31, 2025 15:36:47 GMT+08:00 (8 minutes ago) | Invalid tag.

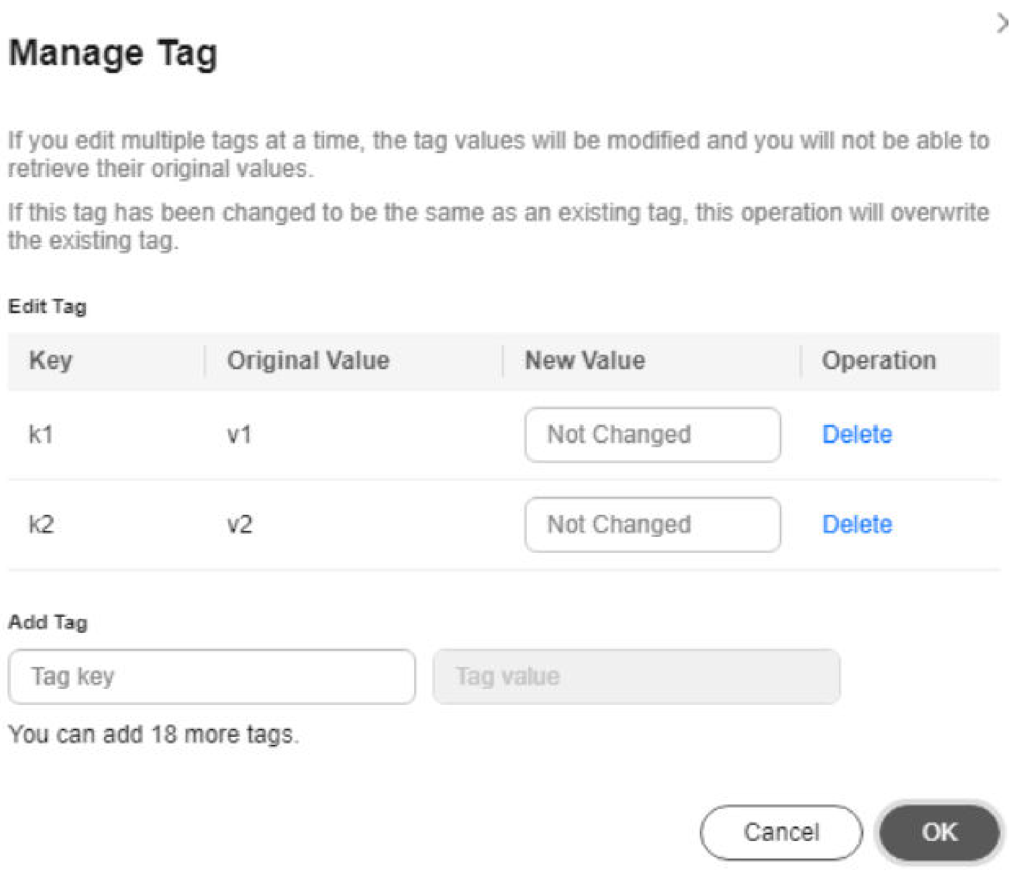
test

Resource Type	Resource Name	Region	Total Tags
instances	test-basic	CN North-[redacted]	0
instances	test-automation	CN North-[redacted]	2
instances	test	CN North-[redacted]	0

Total Records: 3 | Selected: 0

Step 4 Locate each tag to be deleted, click **Delete** in the **Operation** column, and click **OK**.

Figure 3-8 Managing tags



Manage Tag

If you edit multiple tags at a time, the tag values will be modified and you will not be able to retrieve their original values.

If this tag has been changed to be the same as an existing tag, this operation will overwrite the existing tag.

Edit Tag

Key	Original Value	New Value	Operation
k1	v1	Not Changed	Delete
k2	v2	Not Changed	Delete

Add Tag

Tag key Tag value

You can add 18 more tags.

Step 5 (Optional) Click in the upper right corner of the **Search Result** area.

The tag list is refreshed.

----End

3.5.4 Modifying a Repository Tag

You can modify tags on the SWR or TMS console.

Modifying a Tag on the SWR Console

Modifying Tags in a Batch on the TMS Console

Modifying a Tag on the SWR Console

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 On the repository management page, locate the repository you want to modify a tag for and click **Manage Tag**.

Step 3 In the **Manage Tag** dialog box, locate the tag to be modified and enter a new key and value.

----End

Modifying Tags in a Batch on the TMS Console

Step 1 Log in to the TMS console.

Step 2 Choose **Resource Tags** > **Tag Management**, select the target region, set **Resource Type** to **SWR**, and click **Search**. All SWR resources in this region will be returned.

Step 3 Locate the repositories whose tags need to be modified. Click **Manage Tag** above the list.

Step 4 In the **New Value** column, set new values for the tags. Click **OK**.

----End

3.5.5 Querying Repositories by Tag

You can quickly query repositories by tag on the SWR or TMS console.

Querying Repositories on the SWR Console

Querying Repositories on the TMS Console

Querying Repositories on the SWR Console

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 On the repository management page, select one or more tags from the drop-down list on the right to search for the repositories associated with any of these tags.

----End

Querying Repositories on the TMS Console

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Choose **Resource Tags** > **Tag Management**, select the target region, set **Resource Type** to **SWR**, and click **Search**. All SWR resources in this region will be returned.

----End

3.5.6 Managing Namespace Tags

Scenarios

A namespace is used to group container images into a category instead of storing them. A namespace is usually created for a project or department of an enterprise. You can add tags for namespaces to facilitate the search and management.

Prerequisites

A namespace has been created.

Constraints

Table 3-3 Maximum number of tags allowed for a single namespace

Item	Quota
Number of tags for a namespace	20

Adding a Tag to a Namespace

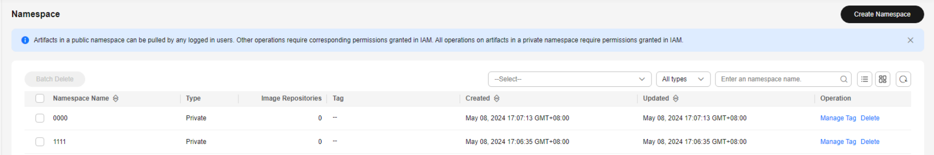
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Locate the repository you want to add a namespace tag for and click the repository name. The repository details page is displayed.

Step 3 In the navigation pane, choose **Namespaces**. Click ☰ in the upper right corner of the page. The namespaces are listed.

Step 4 Locate the namespace you want to add a tag for and click **Manage Tag** in the **Operation** column.

Figure 3-9 Namespace management

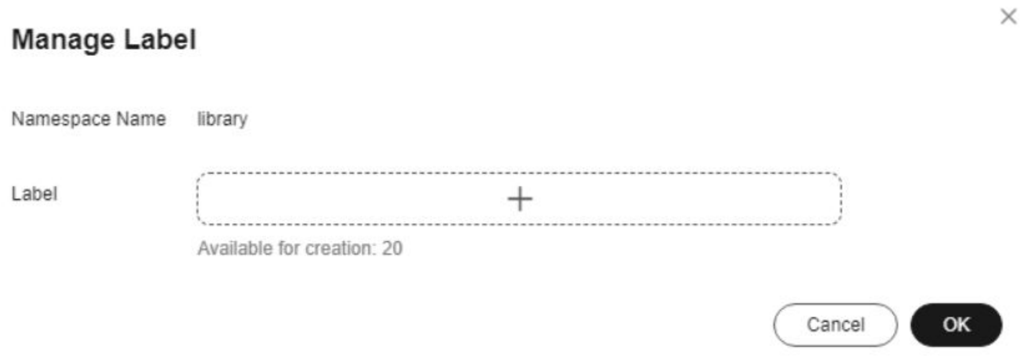


Artifacts in a public namespace can be pulled by any logged in users. Other operations require corresponding permissions granted in IAM. All operations on artifacts in a private namespace require permissions granted in IAM.

□	Namespace Name	Type	Image Repositories	Tag	Created	Updated	Operation
□	0000	Private	0	--	May 08, 2024 17:07:13 GMT+08:00	May 08, 2024 17:07:13 GMT+08:00	Manage Tag Delete
□	1111	Private	0	--	May 08, 2024 17:06:35 GMT+08:00	May 08, 2024 17:06:35 GMT+08:00	Manage Tag Delete

Step 5 In the **Manage Tag** dialog box, click + to add a tag.

Figure 3-10 Tag management



Namespace Name library

Label +

Available for creation: 20

Step 6 Enter a tag key and value.

----End

Modifying a Namespace Tag

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Locate the repository you want to modify a namespace tag for and click the repository name. The repository details page is displayed.

Step 3 In the navigation pane, choose **Namespaces**.

Step 4 Locate the namespace you want to modify a tag for and click **Manage Tag** in the **Operation** column.

Step 5 Enter one or more new keys or values.

----End

Deleting a Namespace Tag

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Locate the repository whose namespace tag needs to be deleted and click the repository name. The repository details page is displayed.

Step 3 In the navigation pane, choose **Namespaces**.

Step 4 Locate the namespace whose tag needs to be deleted and click **Manage Tag** in the **Operation** column.

Step 5 Click **Delete** on the right of the tag.

----End

Querying Namespaces by Tag

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Locate the repository you want to query the namespaces of and click the repository name. The repository details page is displayed.

Step 3 In the navigation pane, choose **Namespaces**.

Step 4 Configure one or more search filters. The search result will be displayed in the list below.

----End

4 Namespace Management

Scenarios

A namespace is used to group container images into a category instead of storing them. A namespace is usually created for a project or department of an enterprise.

NOTE

After a repository is created, a public namespace **library** will be automatically created for it.

Creating a Namespace

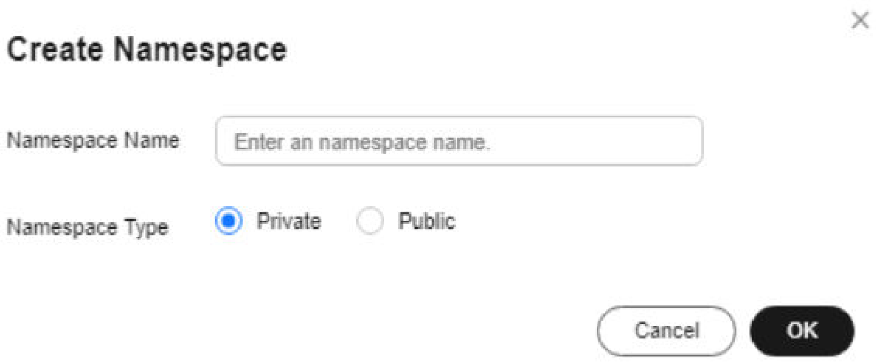
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the navigation pane, choose **Repositories**. Click your repository name.

Step 2 In the navigation pane, choose **Namespaces**.

Step 3 Click **Create Namespace** in the upper right corner.

Step 4 Enter a namespace name and select a namespace type.

Figure 4-1 Creating a namespace



Create Namespace

Namespace Name: Enter an namespace name.

Namespace Type: ● Private ○ Public

- **Public**: Any user can pull artifacts from the namespace after login. If other operations on the artifacts are required, authorize users on the IAM console.
- **Private**: Only users authorized on the IAM console can perform operations on artifacts in the namespace.

Step 5 Click **OK**.

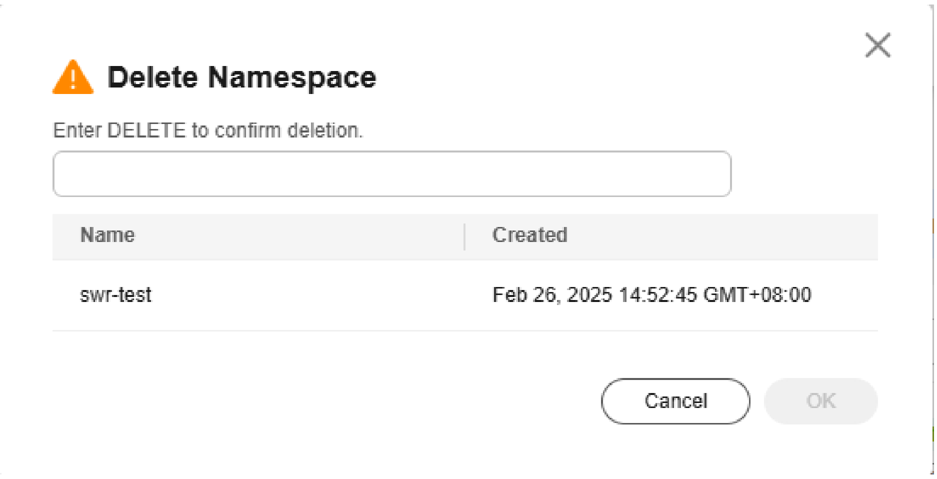
After a namespace is created, you can check its details in the list or card view. Click ☰ or ▦ in the upper right corner to switch the view.

----End

Deleting a Namespace

- List view: Select a namespace and click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** and click **OK**.
- Card view: Select a namespace and click 🗑. In the displayed dialog box, enter **DELETE** and click **OK**.

Figure 4-2 Deleting a namespace



Delete Namespace

Enter DELETE to confirm deletion.

Name	Created
swr-test	Feb 26, 2025 14:52:45 GMT+08:00

NOTE

To avoid deleting important data by mistake, namespaces containing container images cannot be deleted. You need to delete the images first before deleting the namespaces.

5 Access Management

5.1 Access Credentials

Scenarios

Image repositories can only be accessed after you have obtained an access credential. Access credentials can be long-term valid or temporary.

- Long-term credentials: permanently valid after being created and can be disabled or deleted. A long-term credential can be used for preliminary tests, CI/CD pipelines, and image pull to container clusters.

  ⚠ CAUTION

  - Keep long-term credentials safe after they are created. If they are lost, disable or delete them in a timely manner.
  - Federated users cannot create or use long-term credentials.

- Temporary credentials: valid for 24 hours and cannot be disabled or deleted after being created. A temporary credential can be used for temporary use, one-time authorization, or other purposes. For example, it can also be used in production clusters that require high security, if it is periodically refreshed.

Creating a Long-Term Credential

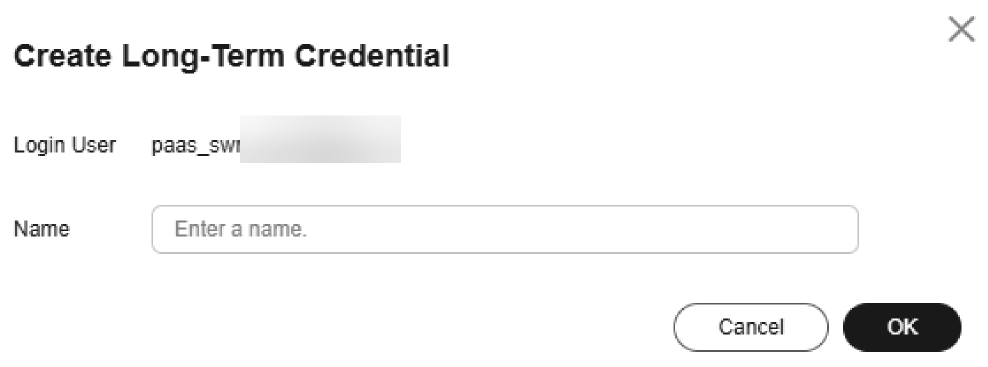
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the navigation pane, choose **Repositories**. Click your repository name.

Step 2 In the navigation pane, choose **Access** > **Access Credentials**.

Step 3 On the **Long-Term Credentials** tab page, click **Create Long-Term Credential**.

Step 4 In the displayed dialog box, enter a credential name.

Figure 5-1 Creating a long-term credential



Create Long-Term Credential

Login User paas_swr[redacted]

Name Enter a name.

Step 5 Click **OK**.

A long-term credential in .csv format will be automatically downloaded.

For container images, a credential is used by the container engine to access image repositories. For details about how to use an image repository, see **Image Management Overview**.

----End

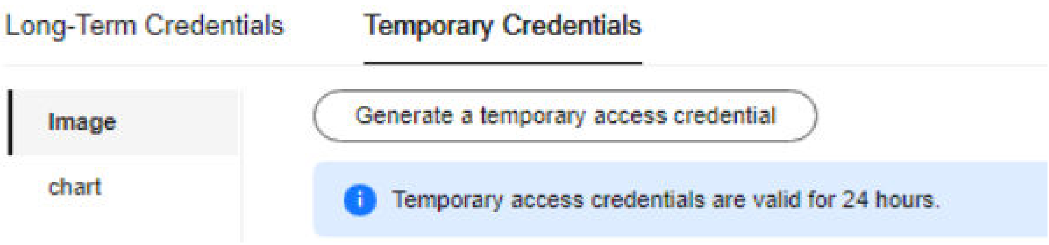
Creating a Temporary Credential

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 In the navigation pane, choose **Access** > **Access Credentials**. Click the **Temporary Credentials** tab.

Step 3 Choose **Image** or **chart** and click **Generate a temporary access credential**.

Figure 5-2 Generating a temporary access credential



Temporary access credentials are valid for 24 hours.

The generated credential is displayed on the current page. You can copy and use it.

For container images, a credential is a Docker command that is used to access image repositories. For details about how to use an image repository, see **Image Management Overview**.

----End

Follow-up Operations

- **Image Management Overview**

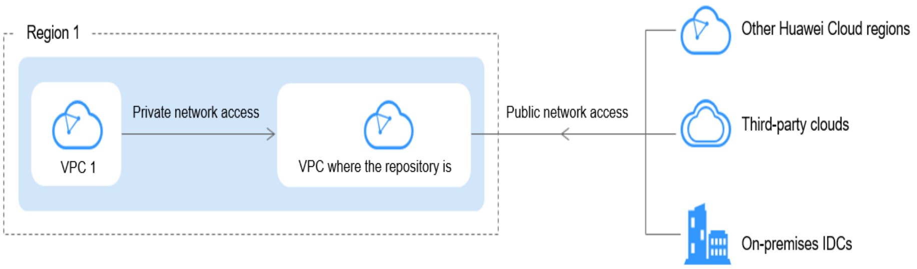
5.2 Access Control

5.2.1 Access Control Overview

By default, access to new SWR Enterprise Edition repositories is blocked for data security. You can configure control policies to allow only required access to repositories.

You can access repositories from the public network or a private network. The permissions are granted separately.

Figure 5-3 Accessing a repository



- Public network access: A whitelist is used to control which IP address CIDR blocks can access repositories.
- Private network access: You can access a repository from any VPC in the region where the repository is. For example, if a repository is in CN East-Shanghai1, you can access it from any VPC in CN East-Shanghai1.

  By default, you can access a repository from a VPC where the repository is. On the **Access Control** > **Private Network Access** page, you can see a default rule to allow the access.

For more information, see:

- **Public Network Access**
- **Private Network Access**

Constraints

To obtain the subnet list of a VPC, IAM users must have the **VPC ReadOnlyAccess** permission. Use your account to log in to IAM and grant this permission to IAM users.

5.2.2 Public Network Access

Scenarios

By default, new repositories cannot be accessed through the Internet. You can configure a whitelist to allow access to a repository through the Internet.

Constraints

You can add a maximum of 300 IP addresses or CIDR blocks to the whitelist when creating a public network access rule.

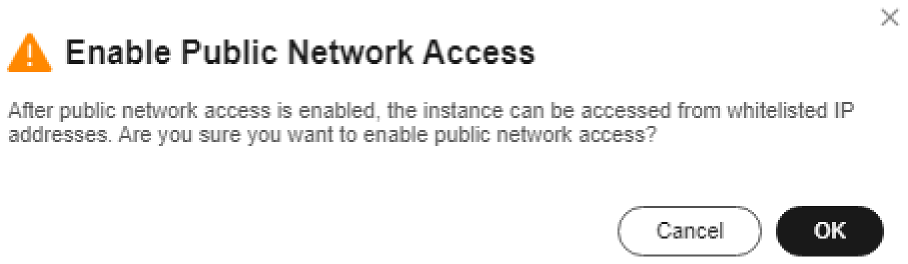
Procedure

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the navigation pane, choose **Repositories**. Click your repository name.

Step 2 In the navigation pane, choose **Access** > **Access Control**.

Step 3 Click the **Public Access** tab and click **Enable Public Network Access**. Read the message in the dialog box and click **OK**.

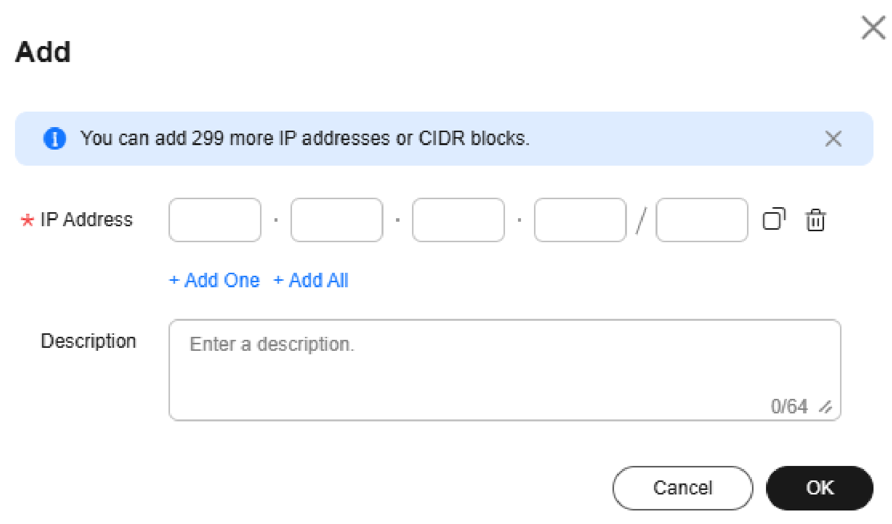
Figure 5-4 Enabling public network access



Enable Public Network Access

After public network access is enabled, the instance can be accessed from whitelisted IP addresses. Are you sure you want to enable public network access?

Step 4 Click **Create Public Network Access Rule** in the upper right corner. In the displayed dialog box, enter or paste the copied CIDR block. Alternatively, click any IP address text box to paste the copied CIDR block. If you need to add multiple CIDR blocks in a batch, click **Add One** for many times. If you want to allow all IP addresses to access the repository, click **Add All**. SWR will automatically add two CIDR blocks (0.0.0.0/1 and 128.0.0.0/1) for you.



You can add 299 more IP addresses or CIDR blocks.

NOTE

- To reduce the risk of attacks, you are advised to add IP addresses one by one instead of adding a CIDR block.
- For each repository, only one rule that allows all IP addresses to access the repository can be added.

Step 5 Click **OK**.

NOTE

The whitelist cannot be modified. You can only delete it and create a new one.

----End

Follow-up Operations

To access a repository, you also need to create an access credential. For details, see **Access Credentials**.

5.2.3 Private Network Access

Scenarios

You can configure a rule to allow certain access to a repository through a private network.

This section describes how to configure private network access for a repository. Once private network access is configured, you can use an ECS in the specified VPC to pull images from the repository over the private network.

After a private network access rule is created, a VPC endpoint will be created in the VPC Endpoint service. You will be billed based on how long you have used the VPC endpoint.

NOTE

By default, you can access a repository from a VPC where the repository is. On the **Access Control** > **Private Network Access** page, you can see a default rule to allow the access.

Constraints

You can configure three private domain names for this VPC endpoint. Ensure that the quota of DNS record sets for private domain names is sufficient.

Procedure

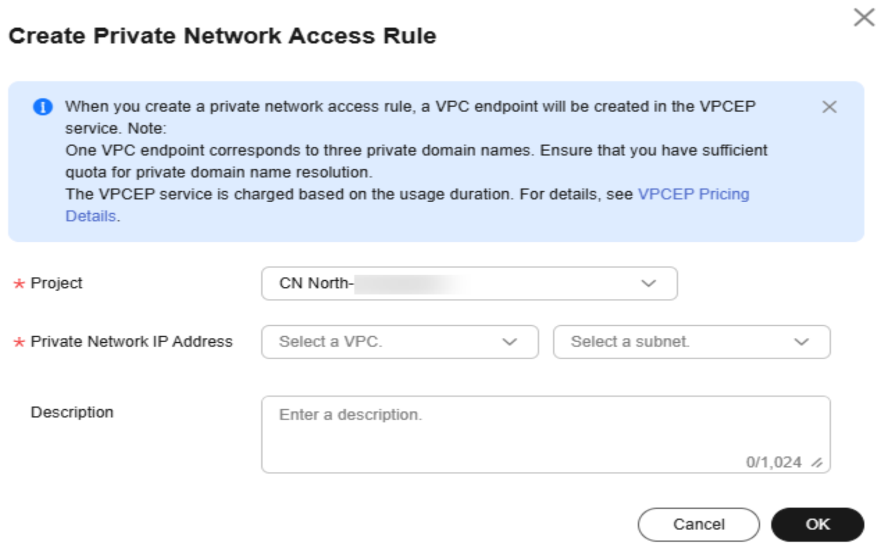
Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the navigation pane, choose **Repositories**. Click your repository name.

Step 2 In the navigation pane, choose **Access** > **Access Control**.

Step 3 Click the **Private Network Access** tab, and click **Create Private Network Access Rule** in the upper right corner.

Step 4 In the displayed dialog box, select a project, VPC, and subnet.

Figure 5-5 Creating a private network access rule



Create Private Network Access Rule

When you create a private network access rule, a VPC endpoint will be created in the VPCEP service. Note:
One VPC endpoint corresponds to three private domain names. Ensure that you have sufficient quota for private domain name resolution.
The VPCEP service is charged based on the usage duration. For details, see VPCEP Pricing Details.

* Project: CN North-[redacted]

* Private Network IP Address: Select a VPC. Select a subnet.

Description: Enter a description. 0/1,024

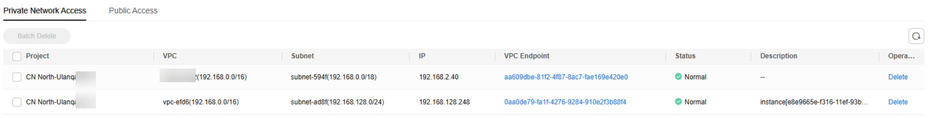
NOTE

If the project you select is not the default one, you need to switch to the project and authorize access to required services in this project before you can continue to create the rule.

Step 5 Click **OK**.

If the **status** changes to **Normal** and there are IP addresses displayed, the private network access rule has been created.

Figure 5-6 Private network access



Project	VPC	Subnet	IP	VPC Endpoint	Status	Description	Opera...
□ CN North-Ulanq[redacted]	[redacted](192.168.0.0/16)	subnet-594f(192.168.0.0/18)	192.168.2.40	aa609dbe-81f2-4f87-8ac7-fae169e420e0	Normal	--	Delete
□ CN North-Ulanq[redacted]	vpc-efd6(192.168.0.0/16)	subnet-ad8f(192.168.128.0/24)	192.168.128.248	0aa0de79-fa1f-4276-9284-910e2f3b88f4	Normal	instance[e8e9665e-f316-11ef-93b...	Delete

Then, you can access the repository from any IP address within the CIDR block of the subnet you selected.

When you create a private network access rule, a VPC endpoint will be created in VPCEP. Do not delete that VPC endpoint.

----End

Follow-up Operations

To access a repository, you also need to create an access credential. For details, see **Access Credentials**.

5.3 Domain Names

There are two types of domain names for SWR Enterprise Edition:

- Default domain name: It is automatically created for each new repository.
- Custom domain name: It is created by a user.

You can create custom domain names when:

- You want to use the domain names planned by your company.
- Repositories are migrated from other registry services and you need to continue to use their original domain names for service continuity.

A repository can have multiple custom domain names in addition to its default domain name. To use a custom domain name, you need to provide the SSL certificate associated with it and access the repository over HTTPS. This section describes how to use a custom domain name to access a repository.

NOTE

A repository can have a maximum of five custom domain names. After a domain name is added or deleted, it takes 60s to 90s to take effect.

Prerequisites

- Domain Name Service (DNS) and Cloud Certificate Manager (CCM) cloud services have been enabled.
- You must have permission to query a certificate list (**scm:cert:list**) and permission to export certificates (varying depending on the IAM console edition).
  - New IAM console: **scm:cert:export**
  - Old IAM console: **scm:cert: download**
- You have a domain name.
- A certificate has been issued for the domain name. You can purchase a certificate using the CCM service and associate the certificate with the domain name.

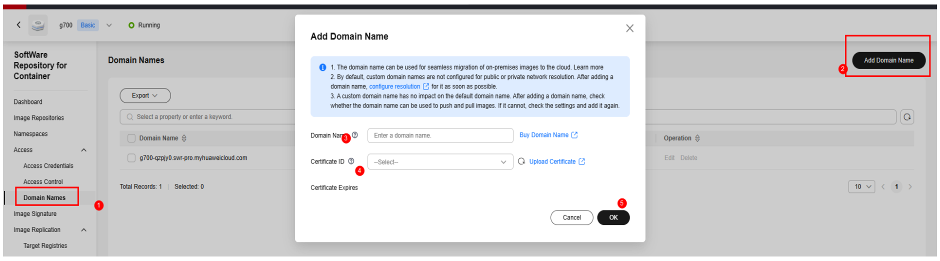
Adding a Domain Name

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. In the navigation pane, choose **Repositories**. Click your repository name.

Step 2 In the navigation pane, choose **Access** > **Domain Names**.

Step 3 Click **Add Domain Name**.

Step 4 In the displayed dialog box, enter a domain name, select the certificate issued for it, and click **OK**.



Add Domain Name

1. The domain name can be used for seamless migration of on-premises images to the cloud. Learn more
2. By default, custom domain names are not configured for public or private network resolution. After adding a domain name, configure resolution for it as soon as possible.
3. A custom domain name has no impact on the default domain name. After adding a domain name, check whether the domain name can be used to push and pull images. If it cannot, check the settings and add it again.

Domain Name: Enter a domain name. Buy Domain Name

Certificate ID: --Select-- Upload Certificate

Certificate Expires

Cancel OK

----End

Updating a Domain Name Certificate

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Click your repository name.

Step 3 In the navigation pane, choose **Access** > **Domain Names**.

Step 4 Locate a domain name, click **Edit** in the **Operation** column.

Step 5 Select the certificate to be updated and click **OK**.

----End

Deleting a Custom Domain Name

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region.

Step 2 Click your repository name.

Step 3 In the navigation pane, choose **Access** > **Domain Names**.

Step 4 Locate a domain name, click **Delete** in the **Operation** column.

Step 5 Enter **DELETE** and click **OK**.

----End

Configuring Domain Name Resolution

- **Public network access**

You can configure **access control** and domain name resolution to access a repository through the Internet using a custom domain name. The following describes how to configure domain name resolution.

Step 1 Log in to the DNS console.

Step 2 In the navigation pane, select **Public Zones**.

Step 3 (Optional) If there is no public domain name with a custom suffix, click **Create Public Zone** in the upper right corner, enter a domain name, and click **OK**.

Step 4 Click your domain name to go to its details page.

Step 5 Click **Add Record Set**. Set parameters and click **OK**.

Table 5-1 Parameters for adding a record set

Parameter	Description
Name	Enter the prefix of the domain name to be resolved.
Type	Type of the record set. Select **CNAME**.
Line	Resolution line. It indicates whether the DNS server will return resolution results based on visitors' carrier networks or geographical locations. **Default** means that, if no lines are matched, the default resolution result will be returned.
TTL	Cache duration of the record set. A shorter TTL is useful for domains whose records change frequently. The default value is 5 minutes.
Value	Set it to the default domain name of the repository.

----End

- **Private network access**

You can configure access control and domain name resolution to pull images from a repository over VPC. The following describes how to configure domain name resolution.

Step 1 Log in to the DNS console.

Step 2 In the navigation pane, select **Private Zones**.

Step 3 (Optional) If there is no private zone with a custom suffix, click **Create Private Zone** in the upper right corner to create one. Enter a domain name, select a region and VPC, and click **OK**.

Step 4 Click your domain name to go to its details page.

Step 5 Click **Add Record Set**. Set parameters and click **OK**.

Table 5-2 Parameters for adding a record set

Parameter	Description
Name	Enter the prefix of the domain name to be resolved.
Type	Type of the record set. Select **CNAME**.
Line	Resolution line. It indicates whether the DNS server will return resolution results based on visitors' carrier networks or geographical locations. **Default** means that, if no lines are matched, the default resolution result will be returned.
TTL	Cache duration of the record set. A shorter TTL is useful for domains whose records change frequently. The default value is 5 minutes.
Value	Set it to the default domain name of the repository.

----End

6 Image Management

6.1 Image Management Overview

SoftWare Repository for Container (SWR) provides easy, secure, and reliable management of container images throughout their lifecycle, facilitating the deployment of containerized applications. You can purchase image repositories of different specifications as needed.

- **Pushing images**: Pushing images (also called uploading images) helps you push local images to an SWR image repository, so that you can manage images more conveniently. You can use either a container engine client or the SWR console to push your images. Currently, there are two types of container engine clients: **Docker** and **containerd**. The supported image artifact types are Docker Image Manifest V2 Schema 2 and Open Container Initiative (OCI).
- **Pulling images**: Pulling images (also called downloading images) is the process of obtaining images from an image repository. Then, you can use this image to deploy containerized applications in CCE or CCI.
- **Adding image triggers**: SWR often works with CCE or CCI to enable automatic application updates. You can add a trigger to automatically update the application that uses the image when the image tag is updated.
- **Replicating images**: After images are uploaded, you can synchronize images of the latest tags to the image repository in another region. Both manual replication and automatic replication are supported.
- **Managing image tag immutability**: To ensure end-to-end trustworthiness and prevent existing images from being overwritten after access credentials are leaked, SWR provides image tag immutability. You can create image tag immutability policies for images in a namespace to ensure that image tags will not be overwritten.
- **Image signing and verification**: To ensure image consistency during distribution and deployment and prevent man-in-the-middle attacks, unauthorized image updates, and unauthorized images, SWR provides the image signing function. After an image is uploaded, it is automatically signed based on the signing rules. When services such as CCE use the image to deploy applications, the image is verified to ensure security.

- **Adding image retention policies**: After images are pushed, you can add retention policies to automatically delete any unused images. There are policies based on the number of image retention days and policies based on the number of image tags.

6.2 Pushing an Image Artifact to an Image Repository

Scenarios

SWR allows you to push (or upload) local image artifacts to an SWR image repository through a container engine client for easier image artifact management.

Pushing an image through a container engine client is to run the **docker** or **ctr** commands on the server where the container engine is installed. If a Docker container engine client is used, run the **docker push** command. If a containerd container engine client is used, run the **ctr push** command.

NOTE

You can push image artifacts over a private network or the public network. For details, see **Configuring Network Access**.

Prerequisites

Before using an image repository, ensure that:

- You have purchased the repository by following the instructions in **Purchasing a Repository** and have permissions to access the repository. For details about the permissions, see **Access Control Overview**.
- You have created an access credential by following the instructions in **Access Credentials**.
- You have created a namespace by following the instructions in **Creating a Namespace**.
- You have prepared a container engine client, which can be used within the network access range defined in **Access Control**.

Constraints

- If a Docker container engine client is used to push images, the Docker version is 18.06 or later.
- If a containerd container engine client is used to push images, the containerd version is 1.5.0 or later.
- The size of each image layer cannot exceed 10 GB.
- A maximum of 160 images can be pushed to a repository of the Enterprise Edition concurrently.

Pushing an Image Using a Container Engine Client

You can refer to the following operations to push image using a Docker or containerd container engine client.

Docker

1. Log in to the server where the container engine client is installed as user **root**.
2. Obtain the temporary or long-term access credential by referring to **Access Credentials** and log in to the container engine client to access the image repository.

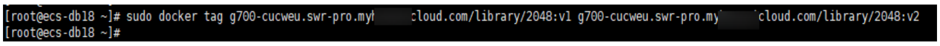
   The message **Login Succeeded** will be displayed upon a successful login.

   CAUTION

   Temporary access credentials are valid for 24 hours after they are generated. Long-term credentials do not expire and can be used permanently.

3. Tag the image.

   docker tag *[image-name-1:tag-1] [repository-address]***/** *[namespace-name]***/** *[image-name-2:tag-2]*

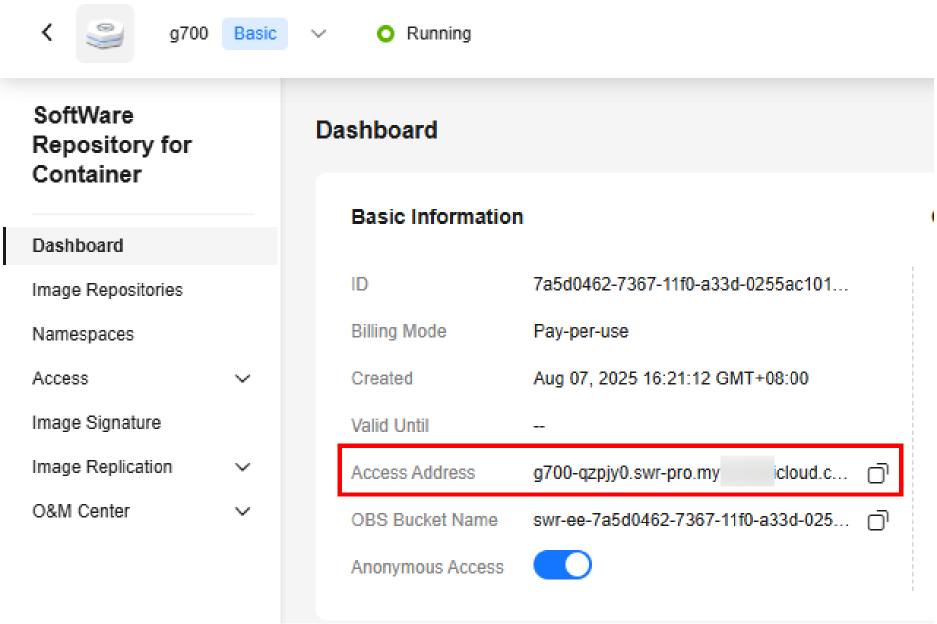
   ```
   [root@ecs-db18 ~]# sudo docker tag g700-cucweu.swr-pro.myh[redacted]cloud.com/library/2048:v1 g700-cucweu.swr-pro.my[redacted]cloud.com/library/2048:v2
   [root@ecs-db18 ~]#
   ```

   In the command:

   – *[image-name-1:tag-1]*: name and tag of the image to be pushed.

   – *[repository-address]*: address for accessing the repository where the image is stored. To obtain the address, perform the following operations:

     Log in to the **SWR console**, switch to the target region in the upper left corner of the page, and choose **Enterprise Edition** in the navigation page. On the displayed page, click the name of the target repository to go to the repository details page. In the **Basic Information** area of the **Dashboard** page, obtain the access address.

   Dashboard

   Basic Information

   ID 7a5d0462-7367-11f0-a33d-0255ac101...

   Billing Mode Pay-per-use

   Created Aug 07, 2025 16:21:12 GMT+08:00

   Valid Until --

   Access Address g700-qzpjy0.swr-pro.my[redacted]cloud.c...

   OBS Bucket Name swr-ee-7a5d0462-7367-11f0-a33d-025...

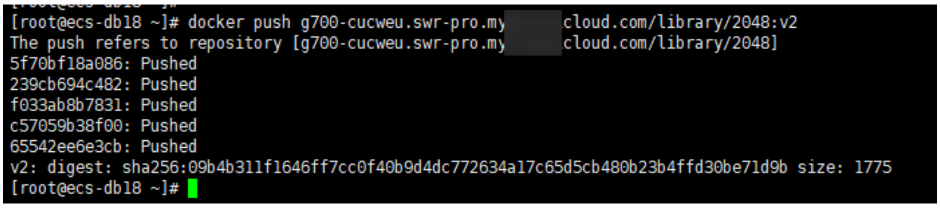
   Anonymous Access

   – *[namespace-name]*: namespace you created in **Creating a Namespace**.

   – *[image-name-2:tag-2]*: new name and tag for the image.

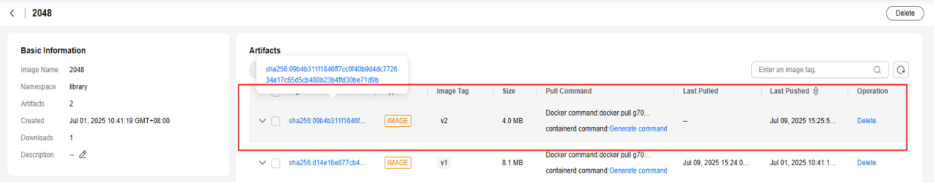
4. Push the image to a repository.

docker push *[repository-address]*/*[namespace-name]*/*[image-name:tag-name]*



```
[root@ecs-db18 ~]# docker push g700-cucweu.swr-pro.my[redacted]cloud.com/library/2048:v2
The push refers to repository [g700-cucweu.swr-pro.my[redacted]cloud.com/library/2048]
5f70bf18a086: Pushed
239cb694c482: Pushed
f033ab8b7831: Pushed
c57059b38f00: Pushed
65542ee6e3cb: Pushed
v2: digest: sha256:09b4b311f1646ff7cc0f40b9d4dc772634a17c65d5cb480b23b4ffd30be71d9b size: 1775
[root@ecs-db18 ~]#
```

5. View the image artifact information in the image artifact list.



< | 2048 Delete

Basic Information

Image Name 2048
Namespace library
Artifacts 2
Created Jul 01, 2025 10:41:19 GMT+08:00
Downloads 1
Description --

Artifacts

sha256:09b4b311f1646ff7cc0f40b9d4dc7726
34a17c65d5cb480b23b4ffd30be71d9b

Enter an image tag.

		Image Tag	Size	Pull Command	Last Pulled	Last Pushed	Operation
sha256:09b4b311f1646f...	IMAGE	v2	4.0 MB	Docker command:docker pull g70... containerd command:Generate command	--	Jul 09, 2025 15:25:5...	Delete
sha256:d14e16e877cb4...	IMAGE	v1	8.1 MB	Docker command:docker pull g70... containerd command:Generate command	Jul 09, 2025 15:24:0...	Jul 01, 2025 10:41:1...	Delete

NOTE

After an image is pushed, you can use it to create a workload on the CCE console.

containerd

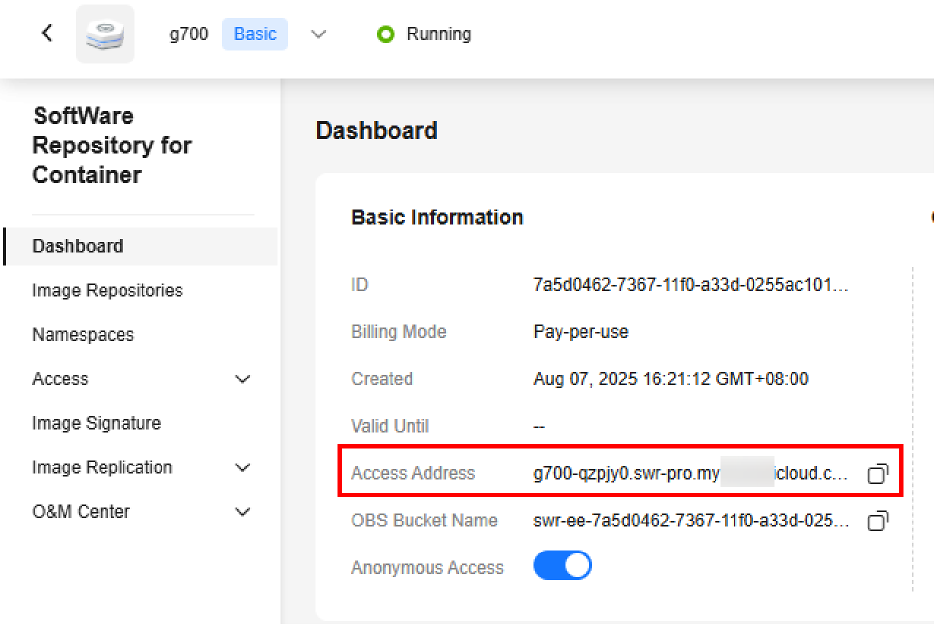
1. Log in to the **SWR console**.
2. In the navigation pane, choose **Enterprise Edition**. On the **Repositories** page, click the name of the target repository to go to the repository details page.
3. In the navigation pane, choose **Image Repositories**. Locate the target image and click **View Pull/Push Commands** in the **Operation** column.
4. On the **containerd** tab, click the copy button next to **ctr -n k8s.io image tag** to copy the tagging command.
5. Log in to the server where the containerd engine is installed as user **root** and run the command copied in **4** to tag the image. Replace the parameters in the command with the actual values before running the command.

   ctr -n k8s.io image tag *[image-name-1:tag-1] [repository-address]*/*[namespace-name]*/*[image-name-2:tag-2]*

   In the preceding command:

   - *[image-name-1:tag-1]*: name and tag of the image to be pushed.
   - *[repository-address]*: address for accessing the repository where the image is stored. To obtain the address, perform the following operations:

     Log in to the **SWR console**, switch to the target region in the upper left corner of the page, and choose **Enterprise Edition** in the navigation page. On the displayed page, click the name of the target repository to go to the repository details page. In the **Basic Information** area of the **Dashboard** page, obtain the access address.



Dashboard

Basic Information

ID 7a5d0462-7367-11f0-a33d-0255ac101...

Billing Mode Pay-per-use

Created Aug 07, 2025 16:21:12 GMT+08:00

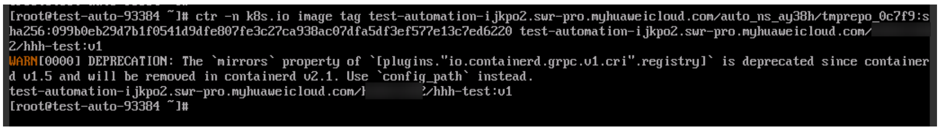
Valid Until --

Access Address g700-qzpjy0.swr-pro.my[redacted]icloud.c...

OBS Bucket Name swr-ee-7a5d0462-7367-11f0-a33d-025...

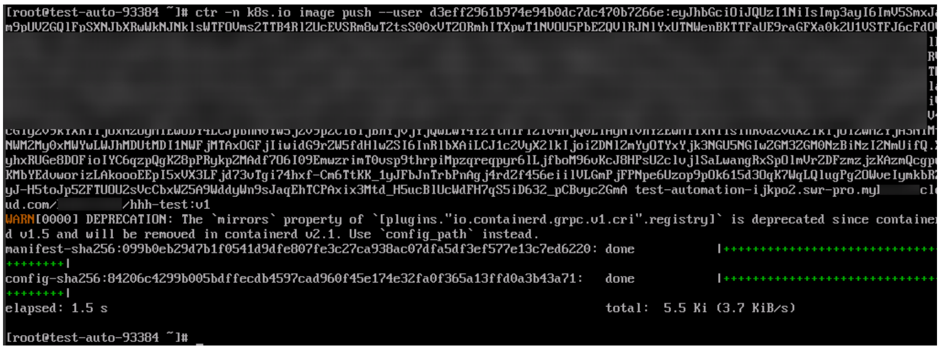
Anonymous Access

- *[namespace-name]*: namespace you created in **Creating a Namespace**.
- *[image-name-2:tag-2]*: new name and tag for the image.



```
[root@test-auto-93384 ~]# ctr -n k8s.io image tag test-automation-ijkpo2.swr-pro.myhuaweicloud.com/auto_ns_ay38h/tmprepo_0c7f9:s
ha256:099b0eb29d7b1f0541d9dfe807fe3c27ca938ac07dfa5df3ef577e13c7ed6220 test-automation-ijkpo2.swr-pro.myhuaweicloud.com/[redacted]
2/hhh-test:v1
WARN[0000] DEPRECATION: The `mirrors` property of `[plugins."io.containerd.grpc.v1.cri".registry]` is deprecated since container
d v1.5 and will be removed in containerd v2.1. Use `config_path` instead.
test-automation-ijkpo2.swr-pro.myhuaweicloud.com/[redacted]2/hhh-test:v1
[root@test-auto-93384 ~]#
```

6. On the **containerd** tab, click the copy button next to **ctr -n k8s.io image push** to copy the push command and change the tag to that in **5**.

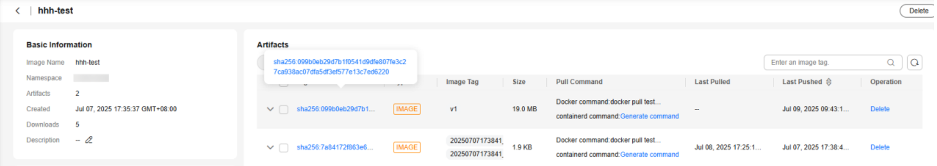


```
[root@test-auto-93384 ~]# ctr -n k8s.io image push --user d3eff2961b974e94b0dc7dc470b7266e:eyJhbGciOiJQUzI1NiIsImp3ayI6ImV5SmxJa
m9pUVZGQ1FpSXNJbXRuWkNJNklsWTFOVms2TTB4R1ZUcEVSRm8wT2tsS00xVTZORmhITXpwT1NVOU5PbEZQV1RJN1YxUTNWenBKTTFaUE9raGFXa0k2U1VSTFJ6cFdO
[redacted]
NWM2My0xMWYwLWJhMDUtMDI1NWFjMTAxOGFjIiwidG9rZW5fdHlwZSI6InRlbXAiLCJ1c2VyX2lkIjoiZDNlZmYyOTYxYjk3NGU5NGIwZGM3ZGM0NzBiNzI2NmUifQ.
yhxRUGe8DOFioIYC6qzpQgKZ8pPRykpZMAdf706I09EmwzrimT0vsp9thrpiMpzqreqpyr6lLjfboM96vKcJ8HPsU2clvjlSaLwangRxSpOlmVrZDFzmzjzKAzmQcgp
KMbYEdvworizLAkoooEEpI5xVX3LFjd73vTgi74hxf-Cm6TtKK_1yJFbJnTrbPnAgj4rd2f456eiilVLGmPjFPNpe6Uzop9pOk615d30qK7WqLQlugPg20WveIymkbR
yJ-H5toJp5ZFTUOU2sVcCbxWZ5A9WddyWn9sJaqEhTCPAxix3Mtd_H5ucBlUcWdFH7qS5iD632_pCBvyc2GmA test-automation-ijkpo2.swr-pro.myl[redacted]cl
ud.com/[redacted]/hhh-test:v1
WARN[0000] DEPRECATION: The `mirrors` property of `[plugins."io.containerd.grpc.v1.cri".registry]` is deprecated since containe
d v1.5 and will be removed in containerd v2.1. Use `config_path` instead.
manifest-sha256:099b0eb29d7b1f0541d9dfe807fe3c27ca938ac07dfa5df3ef577e13c7ed6220: done          |+++++++++++++++++++++++++++++
++++++++|
config-sha256:84206c4299b005bdffecdb4597cad960f45e174e32fa0f365a13ffd0a3b43a71:   done          |+++++++++++++++++++++++++++++
++++++++|
elapsed: 1.5 s                                                                    total:  5.5 Ki (3.7 KiB/s)

[root@test-auto-93384 ~]# _
```

NOTE

The command is only valid for 24 hours after it is generated. To obtain a push command that will remain valid for a long term, see **Access Credentials**.

7. Verify that the image has been pushed.



Basic Information

Image Name hhh-test

Namespace [redacted]

Artifacts 2

Created Jul 07, 2025 17:35:37 GMT+08:00

Downloads 5

Description --

Artifacts

sha256:099b0eb29d7b1f0541d9dfe807fe3c27ca938ac07dfa5df3ef577e13c7ed6220

		Image Tag	Size	Pull Command	Last Pulled	Last Pushed	Operation
sha256:099b0eb29d7b1...	IMAGE	v1	19.0 MB	Docker command:docker pull test... containerd command:Generate command	--	Jul 09, 2025 09:43:1...	Delete
sha256:7a84172f863e6...	IMAGE	20250707173841_ 20250707173841_	1.9 KB	Docker command:docker pull test... containerd command:Generate command	Jul 08, 2025 17:25:1...	Jul 07, 2025 17:38:4...	Delete

6.3 Pulling an Image Artifact to a Local Host

Scenarios

To use an image stored in a repository, you need to pull (or download) it from the repository first. Then, you can use the image to deploy containerized applications in CCE or CCI.

Images are either public or private. If the namespace is public, all images in the namespace are public. If the namespace is private, all images in the namespace are private. Public and private images are different in the following aspects:

- Public images can be downloaded without log into Docker. To improve image management security, SWR also supports permission control over the download of public images through **related control policies**.
- Private images can be downloaded only after you log in to Docker and are granted the download permission (the corresponding action is **swr:repository:downloadArtifact**). For details, see **Table 1 SWR Enterprise Edition operations supported by system-defined policies**.

You can use Docker or containerd to pull images from SWR.

Prerequisites

- Your network is normal.
- You have prepared a container engine client, which can be used within the network access range defined in **Access Control**.

Constraints

If a Docker container engine client is used to pull images, the Docker version is 18.06 or later.

Procedure

You can refer to the following operations to pull image using a Docker or containerd container engine client.

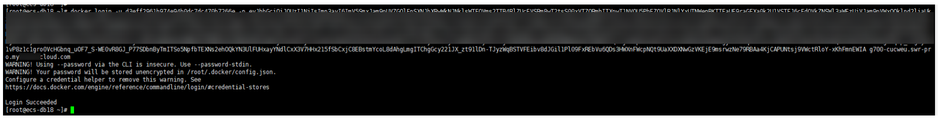
Docker

1. Obtain and copy the temporary access credential.

   ⚠ CAUTION

   Temporary access credentials are valid for 24 hours after they are generated. Long-term credentials do not expire and can be used permanently.

2. Log in to the server where Docker is installed as user **root** and run the command obtained in **1**.



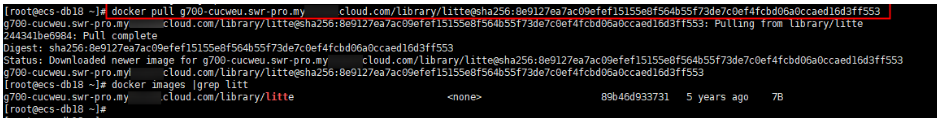
```
[redacted]
1vP8z1clgro0VcHGbnq_uOF7_S-WE0vR8GJ_P77SDbnByTmITSo5NpfbTEXNs2ehOQkYN3UlFUHxayYNdlCxX3V7HHx215fSbCxjC8EBstmYcoL8dAhgLmgITChgGcy22lJX_zt91lDn-TJyzWqBSTVFEibv8dJGil1Pl09FxREbVu6QDs3HWXnFWcpNQt9UaXXDXNwGzVKEjE9msrwzNe79RBAa4KjCAPUNtsj9VWctRloY-xKhFmnEWIA g700-cucweu.swr-pr
o.my[redacted]cloud.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
[root@ecs-db18 ~]#
```

3. Log in to the **SWR console**.
4. In the navigation pane, choose **Enterprise Edition**. On the **Repositories** page, click the name of the target repository to go to the repository details page.
5. In the navigation pane, choose **Image Repositories**. Click the image name to go to the image details page.
6. In the **Artifacts** area on the right, click ⧉ next to **Docker command** in the **Pull Command** column to copy the command.

   NOTE

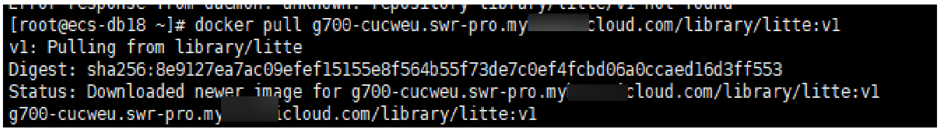
   The command is only valid for 24 hours after it is generated. To obtain a pull command that will remain valid for a long term, see **Access Credentials**.

7. Run the pull command copied in **6** on the server where Docker is installed as user **root**.

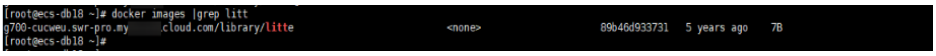
```
[root@ecs-db18 ~]# docker pull g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte@sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553
g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte@sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553: Pulling from library/litte
244341be6984: Pull complete
Digest: sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553
Status: Downloaded newer image for g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte@sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553
g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte@sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553
[root@ecs-db18 ~]# docker images |grep litt
g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte                    <none>                89b46d933731   5 years ago    7B
[root@ecs-db18 ~]#
```

   You can also replace the "at" sign (@) in the pull command copied in **6** with a colon (:) and replace the digest of the image artifact with the image tag.

```
[root@ecs-db18 ~]# docker pull g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte:v1
v1: Pulling from library/litte
Digest: sha256:8e9127ea7ac09efef15155e8f564b55f73de7c0ef4fcbd06a0ccaed16d3ff553
Status: Downloaded newer image for g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte:v1
g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte:v1
```

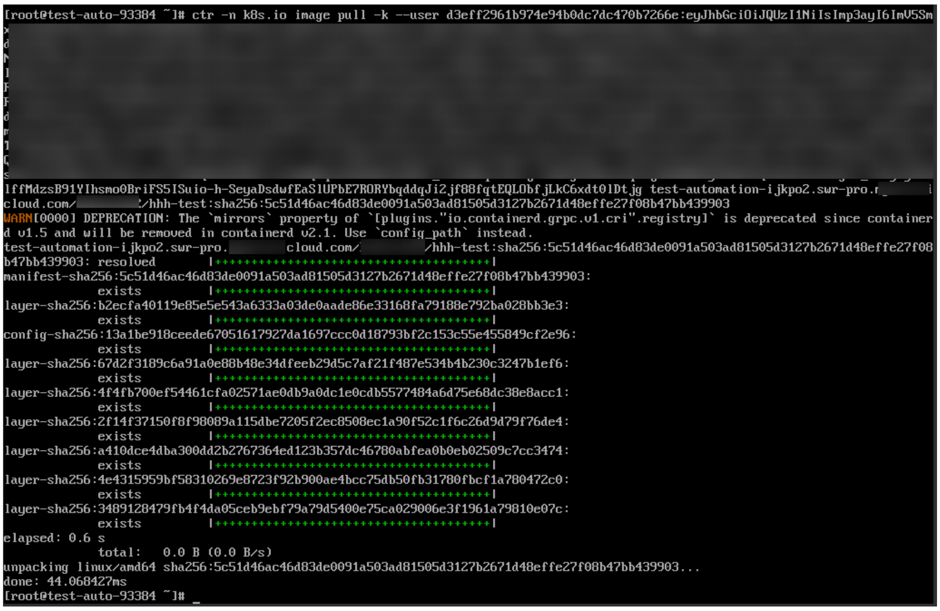
8. Run the **docker images** command to check whether the image is successfully pulled.

```
[root@ecs-db18 ~]# docker images |grep litt
g700-cucweu.swr-pro.my[redacted]cloud.com/library/litte                    <none>                89b46d933731   5 years ago    7B
[root@ecs-db18 ~]#
```

containerd

1. Log in to the **SWR console**.
2. In the navigation pane, choose **Enterprise Edition**. On the **Repositories** page, click the name of the target repository to go to the repository details page.
3. In the navigation pane, choose **Image Repositories**. Click the image name to go to the image details page.
4. In the **Artifacts** area on the right, click **Generate command** next to **containerd command** in the **Pull Command** column. In the displayed dialog box, copy the command.
5. Log in to the server running containerd as user **root**.
6. Run the command copied in **4**.

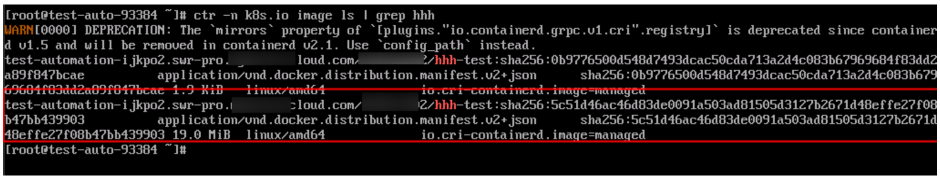


```
[root@test-auto-93384 ~]# ctr -n k8s.io image pull -k --user d3eff2961b974e94b0dc7dc470b7266e:eyJhbGciOiJQUzI1NiIsImp3ayI6ImV5Sm
[redacted]
lffMdzsB91YIhsmo0BriFS5ISuio-h-SeyaDsdwfEaS1UPbE7RORYbqddqJi2jf88fqtEQLObfjLkC6xdt0lDtjg test-automation-ijkpo2.swr-pro.r[redacted]
cloud.com/[redacted]/hhh-test:sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d48effe27f08b47bb439903
WARN[0000] DEPRECATION: The `mirrors` property of `[plugins."io.containerd.grpc.v1.cri".registry]` is deprecated since container
d v1.5 and will be removed in containerd v2.1. Use `config_path` instead.
test-automation-ijkpo2.swr-pro.[redacted]cloud.com/[redacted]/hhh-test:sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d48effe27f08
b47bb439903: resolved       |++++++++++++++++++++++++++++++++++++++|
manifest-sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d48effe27f08b47bb439903:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:b2ecfa40119e85e5e543a6333a03de0aade86e33168fa79188e792ba028bb3e3:
                exists      |++++++++++++++++++++++++++++++++++++++|
config-sha256:13a1be918ceede67051617927da1697ccc0d18793bf2c153c55e455849cf2e96:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:67d2f3189c6a91a0e88b48e34dfeeb29d5c7af21f487e534b4b230c3247b1ef6:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:2f14f37150f8f98089a115dbe7205f2ec8508ec1a90f52c1f6c26d9d79f76de4:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:a410dce4dba300dd2b2767364ed123b357dc46780abfea0b0eb02509c7cc3474:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:4e4315959bf58310269e8723f92b900ae4bcc75db50fb31780fbcf1a780472c0:
                exists      |++++++++++++++++++++++++++++++++++++++|
layer-sha256:3489128479fb4f4da05ceb9ebf79a79d5400e75ca029006e3f1961a79810e07c:
                exists      |++++++++++++++++++++++++++++++++++++++|
elapsed: 0.6 s
                total:   0.0 B (0.0 B/s)
unpacking linux/amd64 sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d48effe27f08b47bb439903...
done: 44.068427ms
[root@test-auto-93384 ~]# _
```

NOTE

The command is only valid for 24 hours after it is generated. To obtain a pull command that will remain valid for a long term, see **Access Credentials**.

7. Verify that the image has been pulled.



```
[root@test-auto-93384 ~]# ctr -n k8s.io image ls | grep hhh
WARN[0000] DEPRECATION: The `mirrors` property of `[plugins."io.containerd.grpc.v1.cri".registry]` is deprecated since container
d v1.5 and will be removed in containerd v2.1. Use `config_path` instead.
test-automation-ijkpo2.swr-pro.[redacted]loud.com/[redacted]2/hhh-test:sha256:0b9776500d548d7493dcac50cda713a2d4c083b67969684f83dd2
a89f847bcae          application/vnd.docker.distribution.manifest.v2+json      sha256:0b9776500d548d7493dcac50cda713a2d4c083b679
69684f83dd2a89f847bcae 1.9 KiB   linux/amd64             io.cri-containerd.image=managed
test-automation-ijkpo2.swr-pro.[redacted]cloud.com/[redacted]2/hhh-test:sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d48effe27f08
b47bb439903          application/vnd.docker.distribution.manifest.v2+json      sha256:5c51d46ac46d83de0091a503ad81505d3127b2671d
48effe27f08b47bb439903 19.0 MiB  linux/amd64             io.cri-containerd.image=managed
[root@test-auto-93384 ~]#
```

6.4 Image Signatures

6.4.1 Signing an Image

Scenarios

You can use keys created in Data Encryption Workshop (DEW) to sign images. This will ensure image consistency during distribution and deployment and prevent man-in-the-middle (MITM) attacks or unauthorized image use and updates. An image can be automatically signed based on a policy after it is pushed. Before signing images, create an asymmetric key in Data Encryption Workshop (DEW). Then, create a signature rule, and set parameters. Images will be manually or automatically signed based on the rule.

Constraints

- Only V1.23 and later clusters are supported.
- Only key algorithms listed in **Table 6-1** can be used.
- A repository can have a maximum of 100,000 image tags and a maximum of 300 image tags can be signed per minute. After the verification plug-in is

installed, the signatures of a maximum of 300 image tags can be verified per minute.

Prerequisites

You have **purchased a repository**.

You have **purchased a CCE cluster**.

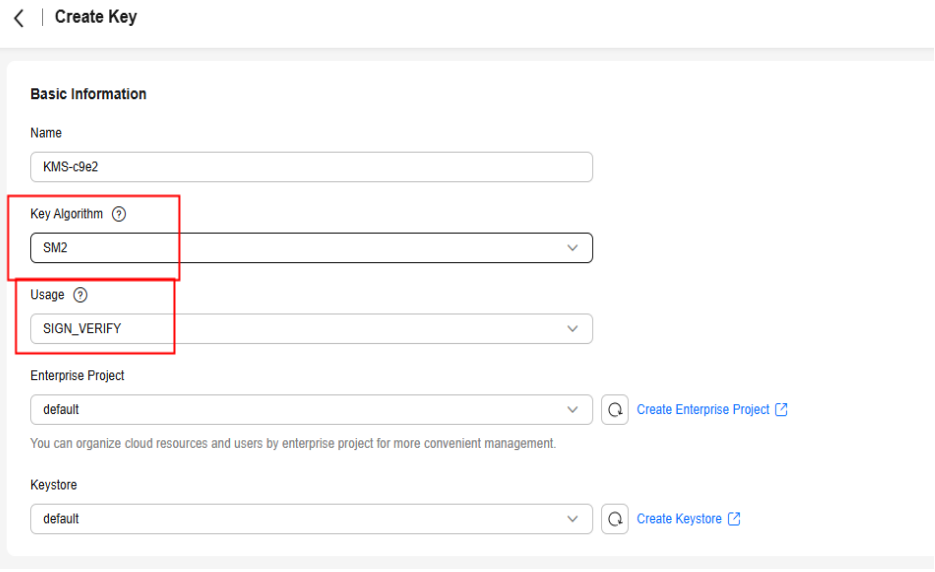
Creating an Asymmetric Key

Step 1 Log in to the DEW console.

Step 2 In the navigation pane, choose **Key Management Service**. Click **Create Key** in the upper right corner.

Step 3 In the displayed dialog box, configure the parameters and click **OK**.

Asymmetric key algorithms are required by image signatures. So, select an ECC or SM2 algorithm for **Key Algorithm** and **SIGN_VERIFY** for **Usage**. For details, see **Table 6-1**. Configure other parameters based on site requirements. For details, see **Creating a Key**.



Create Key

Basic Information

Name: KMS-c9e2

Key Algorithm: SM2

Usage: SIGN_VERIFY

Enterprise Project: default

You can organize cloud resources and users by enterprise project for more convenient management.

Keystore: default

Table 6-1 Key algorithms supported by SWR

Key	Algorithm	Specifications	Description	Used For
Asymmetric	ECC	• EC_P256 – ECDSA_SHA_256 • EC_P384 – ECDSA_SHA_384	NIST Elliptic Curve Cryptography (ECC)	Creating digital signatures
Asymmetric	SM2	SM2	SM2 asymmetric key	Encrypting and decrypting a small amount of data, or creating digital signatures

----End

Creating a Signing Policy

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 2 In the navigation pane, choose **Image Signature**.

Step 3 Click **Create Signing Policy** in the upper right corner.

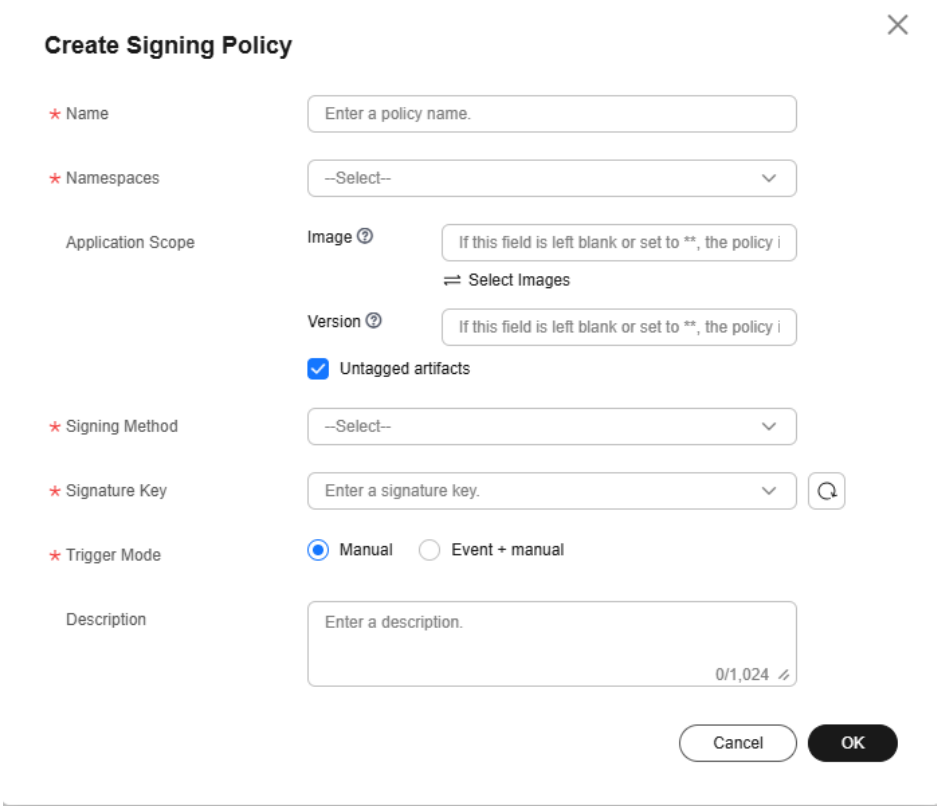
Step 4 In the displayed dialog box, configure the parameters.

Table 6-2 Parameter description

Parameter	Description	Example
Name	Policy name.	SignatureRule
Namespace	Select the namespace where the image is.	library

Parameter	Description	Example
Application Scope	**Image**: Image name. By default, a regular expression is used to match images. Alternatively, you can click ⇆ Select Images to select images. The regular expression can be **nginx-*** or **{***repo1*, *repo2***}**. ● *****: matches any field that does not contain the path separator **/**. ● ******: matches any field that contains the path separator **/**. ● **?**: matches any single character except **/**. ● **{***option 1*, *option 2*, ...**}**: matches any of the options. **Tag**: image tag. A regular expression is used.	**nginx-***: matches images starting with **nginx-**.
Signing Method	Select **KMS**.	KMS
Signature Key	Select the key created in **Creating an Asymmetric Key**. Note the following: ● If the algorithm is not supported, it cannot be selected. ● If the key usage is not **SIGN_VERIFY**, the key cannot be selected. ● If the key status is not **Enabled**, the key cannot be selected.	key1
Trigger Mode	● **Manual**: You need to manually trigger image signing. ● **Event + manual**: When a new image is pushed to a repository and the image matches the regular expression, image signing will be triggered.	Event + manual
Description	Enter a description for the policy.	-

Figure 6-1 Creating a signing policy



Create Signing Policy

* Name: Enter a policy name.

* Namespaces: --Select--

Application Scope
Image: If this field is left blank or set to **, the policy i
⇄ Select Images
Version: If this field is left blank or set to **, the policy i
☑ Untagged artifacts

* Signing Method: --Select--

* Signature Key: Enter a signature key.

* Trigger Mode: ● Manual ○ Event + manual

Description: Enter a description. 0/1,024

Step 5 Click **OK**.

----End

Verifying Image Signing

Log in to the SWR console. In the navigation pane, choose **Enterprise Edition**. Click a repository name to go to its details page. Choose **Image Signature**. Create a signing policy and execute it. After the execution is successful, go to the **Image Repositories** page. Click the signed image. The attachment in the **Artifacts** area is the signature file of the image.

6.4.2 Verifying an Image Signature

Scenarios

To verify image signatures, you need to install the swr-cosign add-on. This section describes how to install the add-on.

Installing swr-cosign

Step 1 Log in to the CCE console.

Step 2 In the navigation pane, choose Add-ons.

Step 3 In the search box, enter **cosign**.

Step 4 Locate the **Container Image Signature Verification** add-on in the search result and click **Install**.

Step 5 Set the following parameters:

- **Cluster**: Select the cluster where the image will be used. Only K8s V1.23 or later clusters are supported.

NOTICE

Before verifying image signatures in a namespace of a cluster, you need to add the **policy.sigstore.dev/include:true** label for the namespace.

- **Version**: Select an add-on version.
- **Specifications**:
  - **Single**: The add-on can be used only in one repository.
  - **HA**: The add-on can be used in two repositories.
  - **Custom**: You can customize the number of repositories, CPU quota, and container quota.

Table 6-3 swr-cosign specifications

Parameter	Description
Add-on Specifications	The value can be **Single**, **HA**, or **Custom**.
Pods	Number of pods that will be created to match the selected add-on specifications. If you selected **Custom** for **Specifications**, you can adjust the number of pods as needed.
Containers	If you selected **Custom** for **Specifications**, you can adjust the container specifications as needed.

- **Parameters**
  - **KMS Key**: Select a key created in **Creating an Asymmetric Key**.
  - **Signature Verification Image**: Click + and select the images whose signatures need to be verified.

Table 6-4 swr-cosign parameters

Parameter	Description
KMS Key	Select a key. Only EC_P256, EC_P384, and SM2 keys are supported. You can create a key using KMS.

Parameter	Description
Signature Verification Image	Enter a regular expression. For example, if you enter **docker.io/****, the signatures of all the images in the **docker.io** repository will be verified. To verify the signatures of all images, enter ****.

Step 6 Click **Install**.

After the installation is complete, select the cluster and click **Add-ons** in the navigation pane. On the displayed page, you can see the installed swr-cosign.

----End

Verifying an Image Signature

Log in to the CCE console and click the name of a cluster where swr-cosign has been installed. In the navigation pane, choose **Workloads** and click **Create Workload**. Select a namespace with the **policy.sigstore.dev/include:true** label and an unsigned image. Select an image access credential and continue to create the workload. The image will fail the signature verification because it has no signature.

6.5 Replicating an Image to Other Regions

Scenarios

You can replicate images between registries. In this way, images in one registry can be used in other registries for quick container deployment and updates globally. You can replicate artifacts between SWR Enterprise Edition and:

- SWR Shared Edition
- An SWR Enterprise Edition registry in another region or a private registry built based on open-source Harbor

You can create a policy to customize a replication. For example, you can customize the artifact type (images, Helm charts, or all), source images and tags (using a regular expression), and whether to overwrite existing artifacts.

You can also access **IAM** and choose **Permissions** > **Policies/Roles** to create a custom policy to determine whether images in a region can be replicated to other regions. After a custom policy is created, if you create an image replication policy in the source or destination region that is not allowed by the custom policy, the replication policy will fail to be created. For details about custom policies, see **SWR Custom Policies**.

6.5.1 Target Registries

Adding a Target Registry

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 2 In the navigation pane, choose **Image Replication** > **Target Registries**.

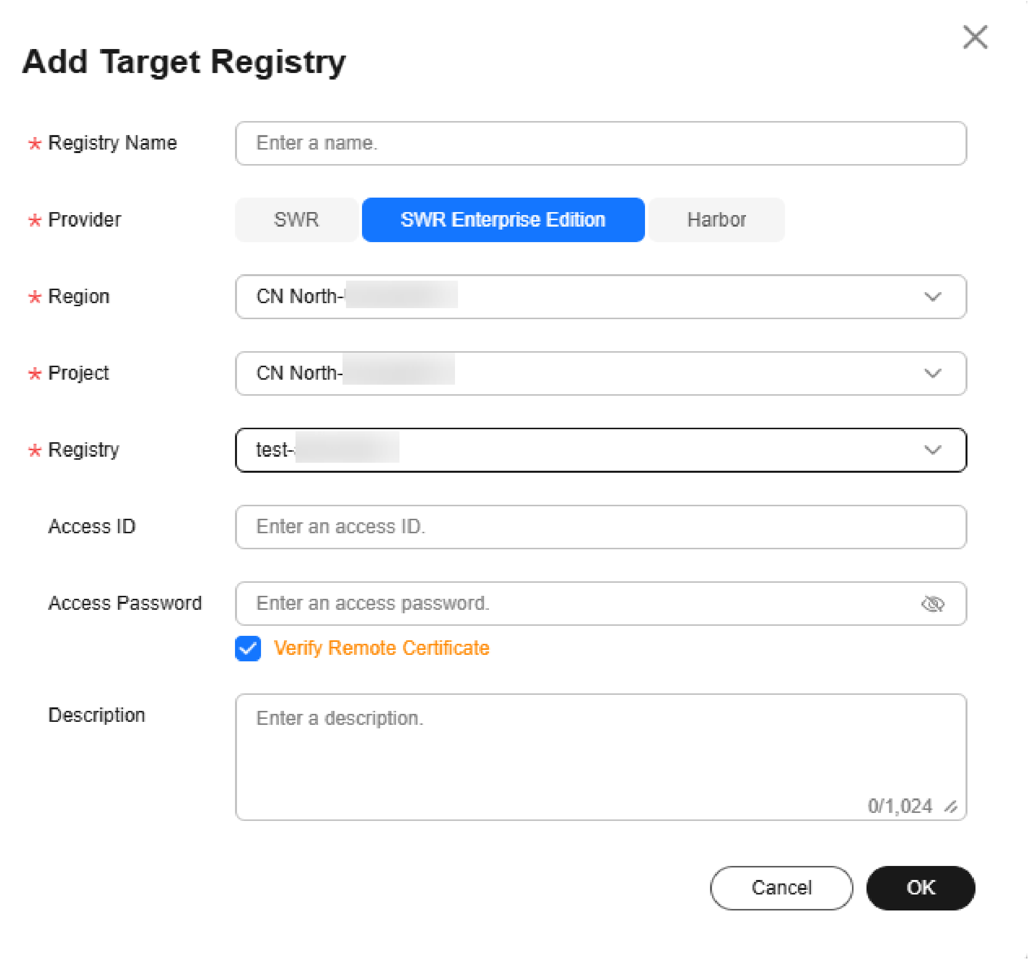
Step 3 In the upper right corner, click **Add Target Registry**.

Table 6-5 Parameter description

Parameter	Description	Example
Registry Name	Target registry name.	remote-registry
Provider	Location of the target registry. The value can be: ● **SWR**: SWR Shared Edition ● **SWR Enterprise Edition**: **Huawei Cloud** indicates SWR Enterprise Edition in another region and **Other** indicates other registry provider. ● **Harbor**: image registry built using **Harbor**.	SWR Enterprise Edition
Registry Address	Target registry address.	swr.cn-east-3.myhuaweicloud.com
Access ID Access Password	ID and password used to access the target registry. The ID and password are the user name and password in the docker login command.	-
Verify Remote Certificate	If you select this option, the system will check whether the remote certificate is released by an authorized organization. If you do not, it will not be checked.	-
Region	Region of the target registry. This parameter is available when the provider is **SWR Enterprise Edition**.	CN East-Shanghai1
Project	Project of the target registry. This parameter is available when the provider is **SWR Enterprise Edition**.	CN East-Shanghai1
Registry	Repository name. This parameter is available when the provider is **SWR Enterprise Edition**.	-
Hosts	This parameter is available only when the provider is Harbor. The backend service can only resolve the public domain name of the current site. If other domain names are involved, set this parameter, for example, to the repository domain name and OBS bucket domain name.	-

Parameter	Description	Example
Description	Describe the target registry.	-

Figure 6-2 Adding a target registry



Add Target Registry

* Registry Name	Enter a name.
* Provider	SWR / **SWR Enterprise Edition** / Harbor
* Region	CN North-[redacted]
* Project	CN North-[redacted]
* Registry	test-[redacted]
Access ID	Enter an access ID.
Access Password	Enter an access password.
	☑ Verify Remote Certificate
Description	Enter a description. 0/1,024

Step 4 Click **OK**.

You can check the health status in the target registry list and modify target registries.

----End

6.5.2 Replication Policies

Creating a Replication Policy

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 2 In the navigation pane, choose **Image Replication** > **Replication Policies**.

Step 3 Click **Add Replication Policy** in the upper right corner.

Step 4 In the displayed dialog box, configure the parameters.

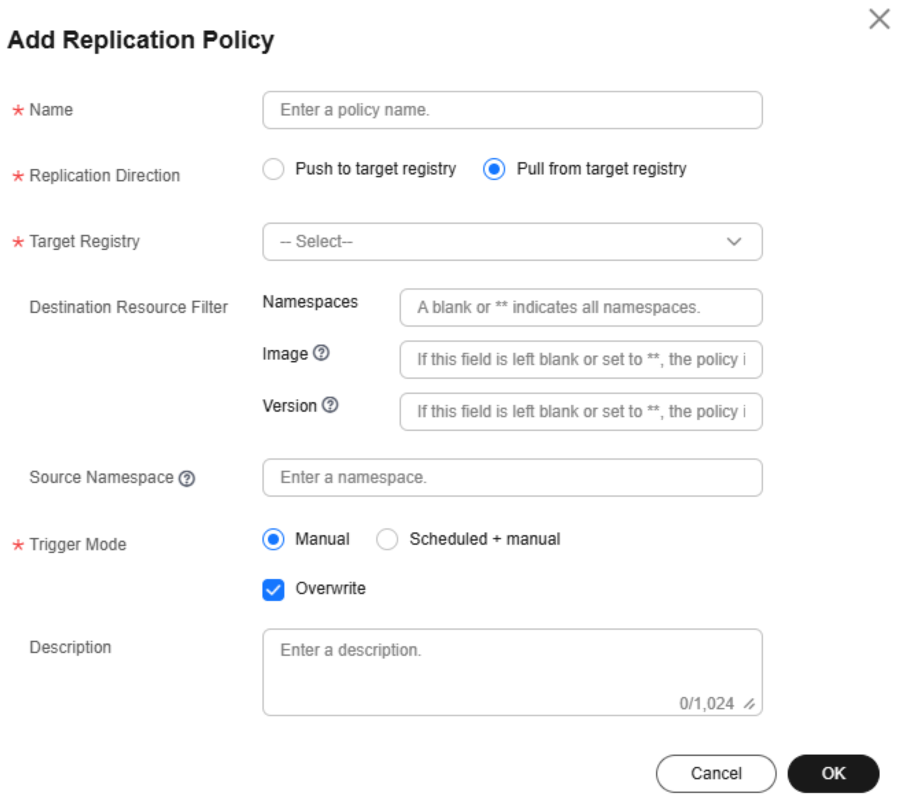
Table 6-6 Parameter description

Parameter	Description	Example
Name	Replication policy name.	SyncRule
Replication Direction	● **Push to target registry**: Push images to the target registry. ● **Pull from target registry**: Pull images from the target registry.	Push to target registry
Target Registry	Select the target registry added in **Adding a Target Registry**.	-
Destination Namespace (for push to target registry)	Namespace that images will be pushed to. A namespace may be called a project on other clouds. If you omit it here, images will be pushed to the same namespace as in the source registry by default. If no such a namespace exists at the destination, replication may fail.	library1
Destination Namespace (for pull from target registry)	Namespace that images will be pulled to. A namespace may be called a project on other clouds. If you omit it here, images will be pulled to the same namespace as in the source registry by default. If no such a namespace exists at the destination, replication may fail.	library1

Parameter	Description	Example
Source Resource Filter (for push to target registry)	**Namespace**: Select a namespace. **Image**: Image name. By default, a regular expression is used to match images. Alternatively, you can click ⇆ Select Images to select images. The regular expression can be **nginx-*** or *{repo1, repo2}*. ● *****: matches any field that does not contain the path separator **/**. ● ******: matches any field that contains the path separator **/**. ● **?**: matches any single character except **/**. ● *{option 1, option 2, ...}*: matches any of the options. **Tag**: Image tag. You can use a regular expression to specify tags. The matching rules are the same as those for images. **NOTE** This parameter is available only when the replication direction is **Push to target registry**.	library2 nginx-* **
Source Resource Filter (for pull from target registry)	**Namespace**: You can use a regular expression to specify namespaces. **Image**: Image name. By default, a regular expression is used to match images. The regular expression can be **nginx-*** or *{repo1, repo2}*. ● *****: matches any field that does not contain the path separator **/**. ● ******: matches any field that contains the path separator **/**. ● **?**: matches any single character except **/**. ● *{option 1, option 2, ...}*: matches any of the options. **Tag**: Image tag. You can use a regular expression to specify tags. The matching rules are the same as those for images. **NOTE** This parameter is available only when the replication direction is **Pull from target registry**.	library2 nginx-* **

Parameter	Description	Example
Trigger Mode	- **Manual**: You need to manually trigger image replication. - **Event + manual**: Image replication is triggered when a new image is pushed or pulled and the image meets the regular expression. - **Scheduled + manual**: **Scheduled** means image replication is triggered periodically.	Scheduled + manual
Scheduled	This parameter is available only when **Trigger Mode** is set to **Scheduled + manual**.	-
Overwrite	Whether to overwrite images at the destination with the same name.	-
Description	Enter a description for the policy.	-

Figure 6-3 Creating a replication policy



Add Replication Policy

* Name: Enter a policy name.

* Replication Direction: ○ Push to target registry ● Pull from target registry

* Target Registry: -- Select--

Destination Resource Filter
Namespaces: A blank or ** indicates all namespaces.
Image: If this field is left blank or set to **, the policy i
Version: If this field is left blank or set to **, the policy i

Source Namespace: Enter a namespace.

* Trigger Mode: ● Manual ○ Scheduled + manual

☑ Overwrite

Description: Enter a description. 0/1,024

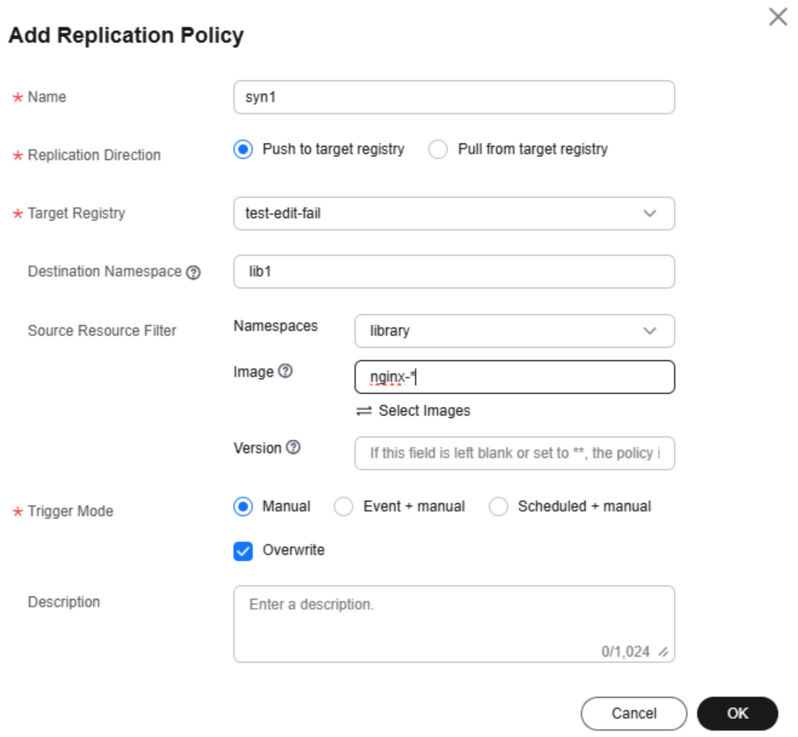
Step 5 Click **OK**.

----End

Replication Policy Examples

- Push to target registry

  Push all images starting with **nginx-in** from the **library** namespace of the local repository to the **lib1** namespace of the target repository **test-edit-fail**. The replication needs to be triggered manually and images with the same name will be overwritten.



Add Replication Policy

* Name: syn1

* Replication Direction: ● Push to target registry ○ Pull from target registry

* Target Registry: test-edit-fail

Destination Namespace: lib1

Source Resource Filter
Namespaces: library
Image: nginx-*
Version: If this field is left blank or set to **, the policy i

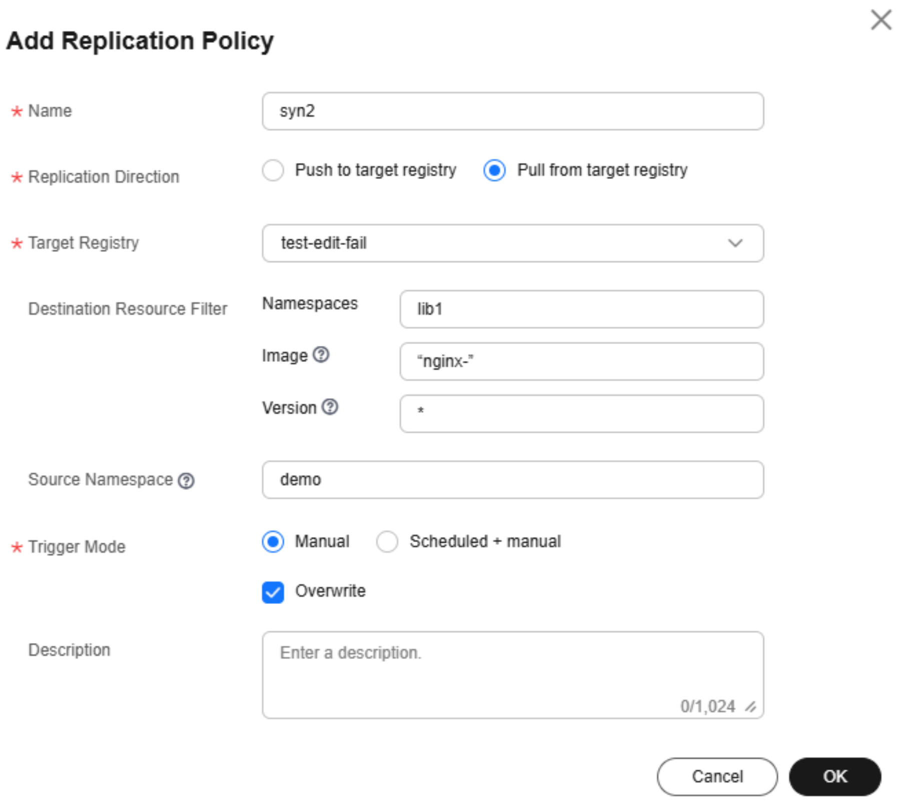
* Trigger Mode: ● Manual ○ Event + manual ○ Scheduled + manual

☑ Overwrite

Description: Enter a description. 0/1,024

- Pull from target registry

  Pull all images starting with **nginx-in** from the **lib1** namespace of the target repository **test-edit-fail** to the **library1** namespace of the local repository. The replication needs to be triggered manually and images with the same name will be overwritten.



Add Replication Policy

* Name: syn2

* Replication Direction: ○ Push to target registry ● Pull from target registry

* Target Registry: test-edit-fail

Destination Resource Filter
Namespaces: lib1
Image: “nginx-”
Version: *

Source Namespace: demo

* Trigger Mode: ● Manual ○ Scheduled + manual

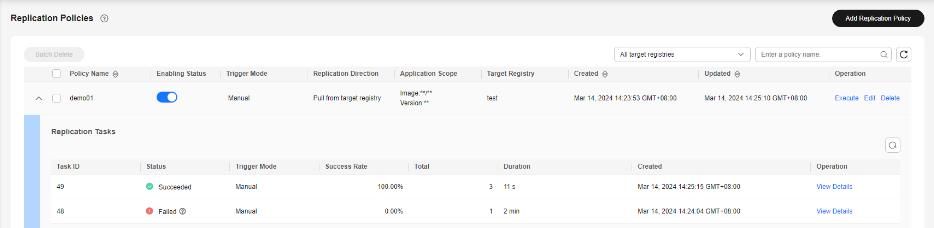
☑ Overwrite

Description: Enter a description. 0/1,024

Managing Replication Policies

You can manage your replication policies as follows:

Figure 6-4 Replication policies



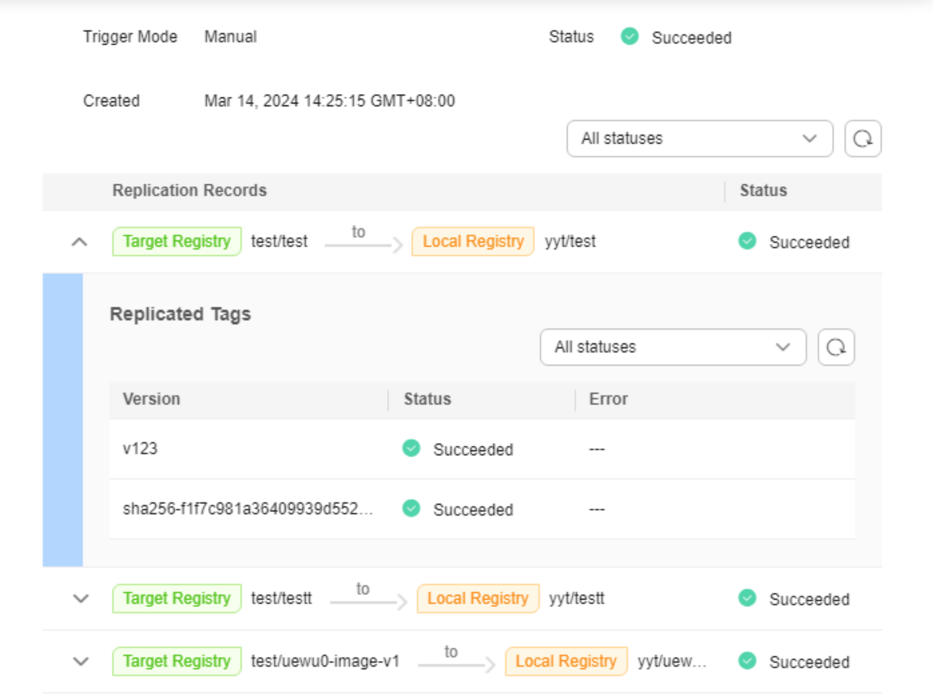
Replication Policies

Policy Name	Enabling Status	Trigger Mode	Replication Direction	Application Scope	Target Registry	Created	Updated	Operation
demo01	●	Manual	Pull from target registry	Image:**/** Version:**	test	Mar 14, 2024 14:23:53 GMT+08:00	Mar 14, 2024 14:25:10 GMT+08:00	Execute Edit Delete

Replication Tasks

Task ID	Status	Trigger Mode	Success Rate	Total	Duration	Created	Operation
49	Succeeded	Manual	100.00%	3	11 s	Mar 14, 2024 14:25:15 GMT+08:00	View Details
48	Failed	Manual	0.00%	1	2 min	Mar 14, 2024 14:24:04 GMT+08:00	View Details

Figure 6-5 Task details



Trigger Mode Manual

Status ● Succeeded

Created Mar 14, 2024 14:25:15 GMT+08:00

All statuses

Replication Records	Status
Target Registry test/test to Local Registry yyt/test	● Succeeded

Replicated Tags

All statuses

Version	Status	Error
v123	● Succeeded	---
sha256-f1f7c981a36409939d552...	● Succeeded	---

Replication Records	Status
Target Registry testt/testt to Local Registry yyt/testt	● Succeeded
Target Registry test/uewu0-image-v1 to Local Registry yyt/uew...	● Succeeded

- Enable or disable a replication policy. indicates a policy is enabled and indicates the policy is disabled. A new policy is enabled by default.
- Manually execute a replication policy.
- Modify a replication policy.
- Delete a replication policy.
- View a replication task. When a replication policy is triggered, the images that meet the policy will be replicated. The following table describes details about a replication task.

Table 6-7 Replication task parameters

Parameter	Description
Task ID	Unique ID of a replication task for a repository.
Status	Task status.
Trigger Mode	The value is **Manual** or **Automatic**. If you click **Execute**, the trigger mode is **Manual**. If the replication is executed periodically based on a schedule, the trigger mode is **Automatic**.
Success Rate	The percentage of images that are successfully replicated to the total number of images that need to be replicated.

Parameter	Description
Total	Total number of images to be replicated in the current task.
Duration	Time required to complete a task.
Created	Time when a replication task was triggered.
Operation	**View Details**: You can check the replicated images in the right pane after clicking this button.

6.5.3 Replicating Images

Procedure

Step 1 Purchase a repository. For details, see **Purchasing a Repository**.

Step 2 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 3 In the navigation pane, choose **Image Replication** > **Target Registries**.

Step 4 Configure a target registry as described in **Table 6-5**.

Step 5 In the navigation pane, choose **Image Replication** > **Replication Policies** to create a replication policy. For details, see **Creating a Replication Policy**. Images will be manually or automatically replicated based on the policy.

----End

6.6 Triggers

Scenarios

You can create a trigger to automatically execute the defined HTTP POST requests. For example, when an image is pushed, the CI/CD pipeline will automatically pull and deploy the image to a cluster. In this way, you can quickly connect to the CI/CD pipeline for container DevOps.

Creating a Trigger

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 2 In the navigation pane, choose **O&M Center** > **Triggers**.

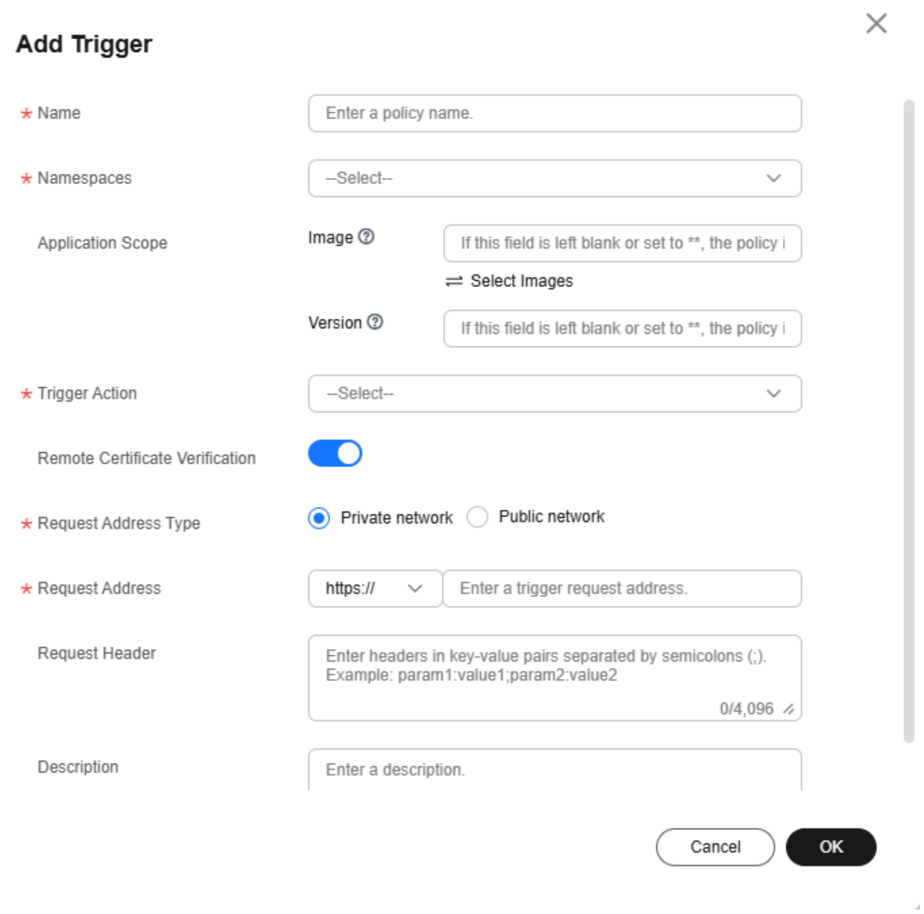
Step 3 Click **Add Trigger** in the upper right corner.

Step 4 In the displayed dialog box, configure the parameters.

Table 6-8 Parameter description

Parameter	Description	Example
Name	Trigger name.	TriggerRule
Namespace	Namespace where a trigger will be created.	library1
Application Scope	**Image**: Image name. By default, a regular expression is used to match images. Alternatively, you can click ⇆ Select Images to select images. The regular expression can be **nginx-*** or *{repo1, repo2}*. ● *: matches any field that does not contain the path separator **/**. ● **: matches any field that contains the path separator **/**. ● **?**: matches any single character except **/**. ● *{option 1, option 2, ...}*: matches any of the options. **Tag**: Image tag. You can use a regular expression to specify tags. The matching rules are the same as those for images.	nginx-*
Trigger Action	You can set the following action as a trigger: ● Pushing an image	Pushing an image
Remote Certificate Verification	If you select this option, the system will check whether the remote certificate is released by an authorized organization. If you do not, it will not be checked.	-
Request Address Type	● Private network ● Public network	Private network
Request Address	IP address the trigger will send a POST request to. **CAUTION** The IP address must fall into the default VPC network CIDR block you specified when you purchased the repository.	-
Request Header	When a trigger sends a POST request, the header information can be in **Key:Value** format. Example: **Authentication:** *xxxxxxx*. Use semicolons (;) to separate multiple headers, for example, *param1***:***value1***;***param2***:***value2*.	-

Figure 6-6 Creating a trigger



Add Trigger

* Name: Enter a policy name.

* Namespaces: --Select--

Application Scope: Image: If this field is left blank or set to **, the policy i
⇄ Select Images
Version: If this field is left blank or set to **, the policy i

* Trigger Action: --Select--

Remote Certificate Verification: ●

* Request Address Type: ● Private network ○ Public network

* Request Address: https:// | Enter a trigger request address.

Request Header: Enter headers in key-value pairs separated by semicolons (;). Example: param1:value1;param2:value2 0/4,096

Description: Enter a description.

Step 5 Click **OK**.

----End

Managing Triggers

You can manage your triggers as follows:

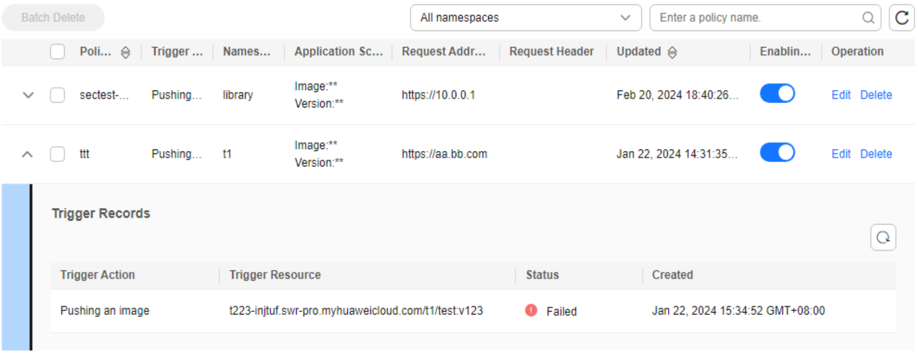
- Enable or disable a trigger. ● indicates a trigger is enabled and ○ indicates the trigger is disabled. A new trigger is enabled by default.
- Modify a trigger. All parameters except **Namespace** and **Request Address** can be modified.
- Delete a trigger.
- View a trigger. When the action specified in a trigger is executed, the trigger will send a request. You can click ∨ to view trigger records.

Table 6-9 Trigger records

Parameter	Description
Trigger Action	Action that triggers a request.
Trigger Resource	Repository resource on which the action was performed.

Parameter	Description
Status	Status of the Webhook request sent by a trigger.
Created	Time when the Webhook request was sent.

Figure 6-7 Managing triggers



	Poli...	Trigger ...	Names...	Application Sc...	Request Addr...	Request Header	Updated	Enablin...	Operation
□	sectest-...	Pushing...	library	Image:** Version:**	https://10.0.0.1		Feb 20, 2024 18:40:26...	●	Edit Delete
□	ttt	Pushing...	t1	Image:** Version:**	https://aa.bb.com		Jan 22, 2024 14:31:35...	●	Edit Delete

Trigger Records

Trigger Action	Trigger Resource	Status	Created
Pushing an image	t223-injtuf.swr-pro.myhuaweicloud.com/t1/test:v123	Failed	Jan 22, 2024 15:34:52 GMT+08:00

6.7 Image Tag Immutability

Scenarios

To ensure end-to-end trust and prevent existing images from being overwritten if a set of access credentials gets leaked, you can configure an immutability policy for images in a namespace. If you attempt to push an image with a tag that is already in the namespace, an error will be returned.

Constraints

Only one immutability policy can be created for each namespace.

Creating an Image Tag Immutability Policy

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click your repository name.

Step 2 In the navigation pane, choose **O&M Center** > **Image Tag Immutability**.

Step 3 Click **Create Immutability Policy** in the upper right corner.

Step 4 In the displayed dialog box, configure the parameters.

Table 6-10 Image tag immutability policies

Parameter	Description
Namespace	Namespace where an immutability policy will be created. It can be a public or private namespace.

Parameter		Description
Application Scope	Image	Select one or more images in the namespace.
	Tag	Specify the image tags that the policy will be applied to. If this parameter is omitted or set to ******, the policy will be applied to all image tags.

NOTE

- For **Image**, you can select one or more images from the list.
- Alternatively, you can enter a regular expression.

  The regular expression can be **nginx-*** or *{repo1, repo2}*.

  - *****: matches any field that does not contain the path separator **/**.
  - ******: matches any field that contains the path separator **/**.
  - **?**: matches any single character except **/**.
  - *{option 1, option 2, ...}*: matches any of the options.

Step 5 Click **OK**.

----End

Managing Image Tag Immutability Policies

You can manage your immutability policies as follows:

- Enable or disable an immutability policy. 🔵 indicates a policy is enabled and ⚪ indicates the policy is disabled. A new policy is enabled by default.
- Modify an immutability policy. All parameters except **Namespace** can be modified.
- Delete an immutability policy.

6.8 Image Retention

Scenarios

In modern software development, images are generated in pipelines and updated in each iteration. When images of earlier versions are no longer needed, you can delete them by using image retention policies, which can be, manually or periodically triggered. The rules in a policy can be used separately or in a combination.

Constraints

There can only be one retention policy in a given namespace. Each policy has 1 to 15 rules.

Creating an Image Retention Policy

Step 1 Log in to the **SWR console**. In the upper left corner, switch to your region. Click a repository name to go to its details page.

Step 2 In the navigation pane, choose **O&M Center** > **Image Retention**.

Step 3 Click **Add Retention Policy** in the upper right corner.

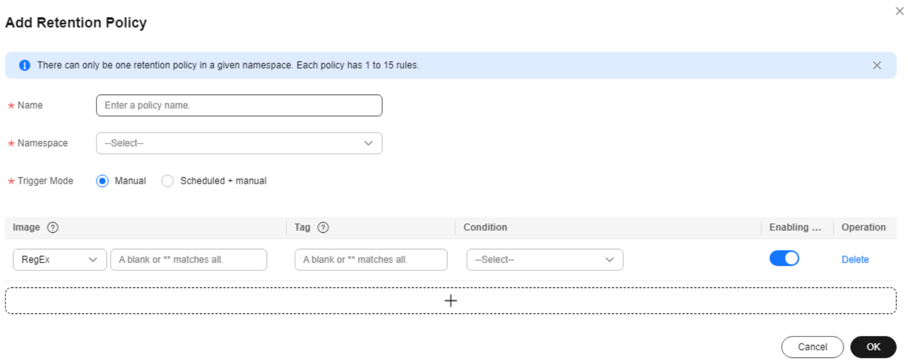
Step 4 In the displayed dialog box, configure the parameters.

Table 6-11 Parameter description

Parameter	Description	Example
Name	Retention policy name.	AgingRule
Namespace	Namespace where the retention policy will be applied.	library1
Trigger Mode	● **Manual**: You need to manually trigger image retention. ● **Scheduled + manual**: **Scheduled** means image retention is triggered periodically.	Scheduled + manual
Scheduled	This parameter is available only when **Trigger Mode** is set to **Scheduled + manual**.	-
Image	You can: ● Enter a regular expression. Example: **nginx-*** or *{repo1, repo2}*. – *****: matches any field that does not contain the path separator **/**. – ******: matches any field that contains the path separator **/**. – **?**: matches any single character except **/**. – *{ repo1, repo2, ...}*: matches any of the options. Note: If this parameter is left blank or set to ******, all images will be matched. ● Select images from a list.	nginx-*

Parameter	Description	Example
Tag	Image tag. Enter a regular expression. Example: **v1*** or *{v1, v2}*. ● *: matches any field that does not contain the path separator **/**. ● **: matches any field that contains the path separator **/**. ● **?**: matches any single character except **/**. ● *{v1, v2}*: matches any of the options.	v1
Condition	Retention condition. The options are as follows: ● Retain the # image tags pushed most recently ● Retain the # image tags pulled most recently ● Retain image tags pushed within the last # days ● Retain image tags pulled within the last # days *#* indicates the number of tags or days.	Retain the 10 image tags pushed most recently
Enable	Whether to enable or disable a retention rule.	-
Operation	You can delete a retention rule.	-

Figure 6-8 Creating an image retention policy



Add Retention Policy

There can only be one retention policy in a given namespace. Each policy has 1 to 15 rules.

* Name: Enter a policy name.

* Namespace: --Select--

* Trigger Mode: ● Manual ○ Scheduled + manual

Image		Tag	Condition	Enabling ...	Operation
RegEx	A blank or ** matches all.	A blank or ** matches all.	--Select--	●	Delete

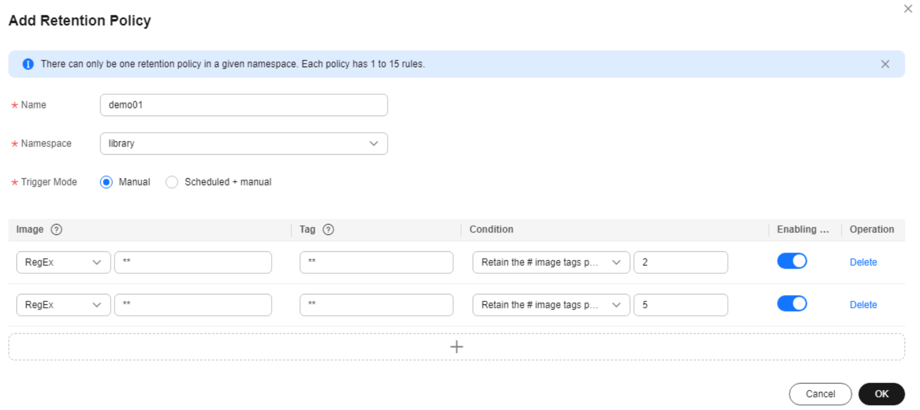
Step 5 Click **OK**.

----End

Retention Policy Examples

- Example 1:

  In the **library** namespace, for all images, retain the 2 most recently pushed and the 5 most recently pulled tags. The policy needs to be manually triggered.

  Add Retention Policy

  There can only be one retention policy in a given namespace. Each policy has 1 to 15 rules.

  * Name: demo01

  * Namespace: library

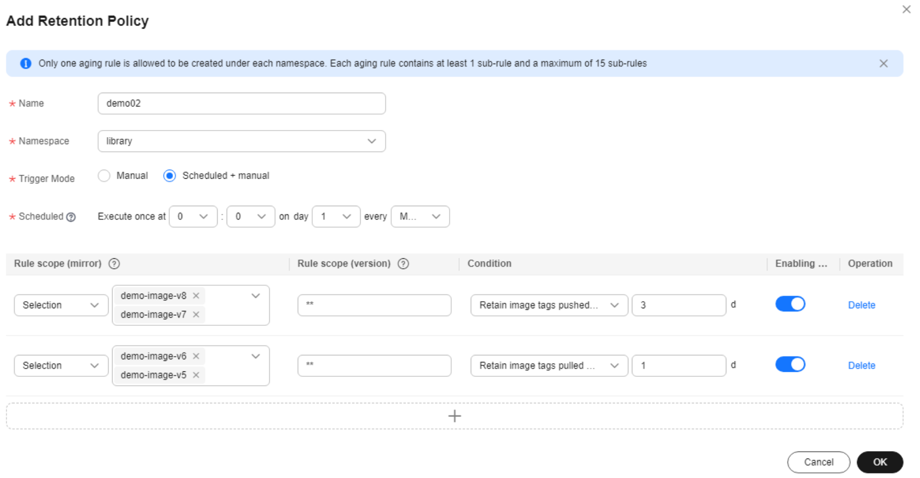
  * Trigger Mode: ● Manual ○ Scheduled + manual

Image		Tag	Condition		Enabling ...	Operation
RegEx	**	**	Retain the # image tags p...	2	●	Delete
RegEx	**	**	Retain the # image tags p...	5	●	Delete

  For example, there are 10 image tags. Image tags 9 and 10 are most recently pushed. Image tags 1 to 5 are most recently pulled. Based on the policy, image tags 6 to 8 will be deleted.

- Example 2:

  In the **library** namespace, retain the tags pushed in the last 3 days for the **demo-image-v8** and **demo-image-v7** images and the tags pulled in the last day for the **demo-image-v6** and **demo-image-v5** images. The policy is executed at 00:00 of the first day every month but can also be triggered manually.

  Add Retention Policy

  Only one aging rule is allowed to be created under each namespace. Each aging rule contains at least 1 sub-rule and a maximum of 15 sub-rules

  * Name: demo02

  * Namespace: library

  * Trigger Mode: ○ Manual ● Scheduled + manual

  * Scheduled: Execute once at 0 : 0 on day 1 every M...

Rule scope (mirror)		Rule scope (version)	Condition			Enabling ...	Operation
Selection	demo-image-v8 demo-image-v7	**	Retain image tags pushed...	3	d	●	Delete
Selection	demo-image-v6 demo-image-v5	**	Retain image tags pulled ...	1	d	●	Delete

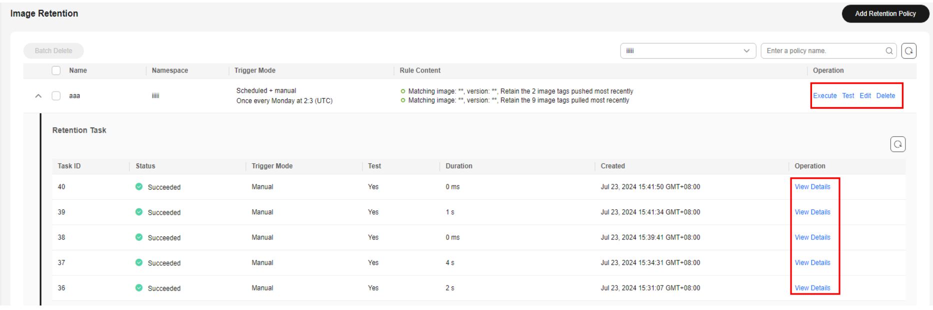
Managing Retention Policies

You can:

- Execute a retention policy. To prevent misoperations, you are advised to test a retention policy before executing it for the first time.
- Test a retention policy. You can use it to check whether a policy is in effect but no image tags will be deleted in the test.

- Modify a retention policy. All parameters except **Namespace** can be modified.
- Delete a retention policy.

Figure 6-9 Managing retention policies



Name	Namespace	Trigger Mode	Rule Content	Operation
aaa	iiiii	Scheduled + manual Once every Monday at 2:3 (UTC)	Matching image: **, version: **, Retain the 2 image tags pushed most recently Matching image: **, version: **, Retain the 9 image tags pulled most recently	Execute Test Edit Delete

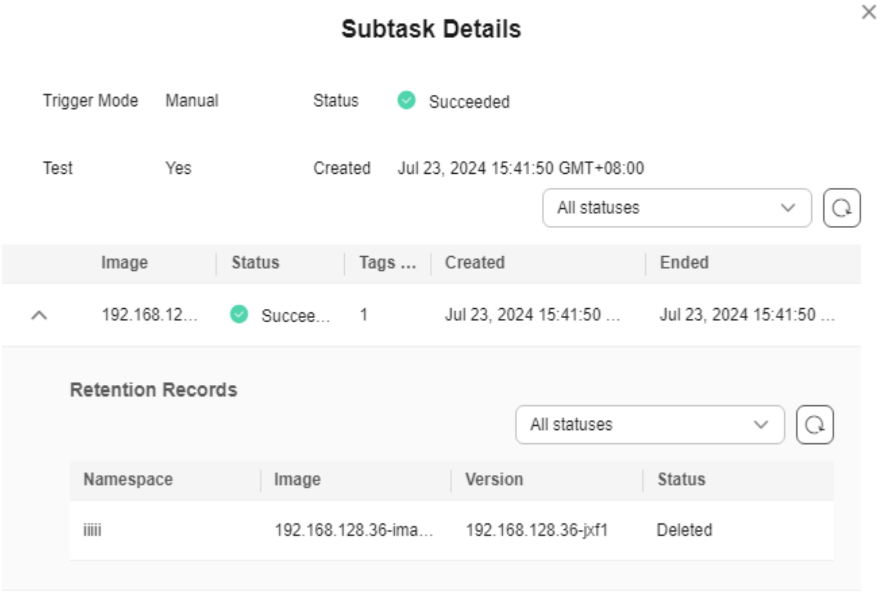
Task ID	Status	Trigger Mode	Test	Duration	Created	Operation
40	Succeeded	Manual	Yes	0 ms	Jul 23, 2024 15:41:50 GMT+08:00	View Details
39	Succeeded	Manual	Yes	1 s	Jul 23, 2024 15:41:34 GMT+08:00	View Details
38	Succeeded	Manual	Yes	0 ms	Jul 23, 2024 15:39:41 GMT+08:00	View Details
37	Succeeded	Manual	Yes	4 s	Jul 23, 2024 15:34:31 GMT+08:00	View Details
36	Succeeded	Manual	Yes	2 s	Jul 23, 2024 15:31:07 GMT+08:00	View Details

- View a retention task. When a retention policy is triggered, only the images that meet the policy will be retained. The following table describes details about a retention task.

Table 6-12 Retention task parameters

Parameter	Description
Trigger Mode	The value is **Manual** or **Automatic**. If you click **Execute** or **Test**, the trigger mode is **Manual**. If the replication is executed periodically based on a schedule, the trigger mode is **Automatic**.
Status	Task status.
Test	The value can be **Yes** or **No**. If you click **Test**, the value is **Yes**. If you click **Execute**, the value is **No**. You can use **Test** to check whether a policy is in effect but no image tags will be deleted in the test.
Tags Deleted	Number of image tags that are deleted based on the policy.
Created	Time when a retention task was triggered.
Ended	Time when a retention task was ended.
Retention Records	Retention records of each image tag, such as the namespace, tag name, and retention results.

Figure 6-10 Task details



Subtask Details

Trigger Mode	Manual	Status	Succeeded
Test	Yes	Created	Jul 23, 2024 15:41:50 GMT+08:00

All statuses

Image	Status	Tags ...	Created	Ended
192.168.12...	Succee...	1	Jul 23, 2024 15:41:50 ...	Jul 23, 2024 15:41:50 ...

Retention Records

All statuses

Namespace	Image	Version	Status
iiiii	192.168.128.36-ima...	192.168.128.36-jxf1	Deleted

7 Using CTS to Audit SWR

7.1 SWR Operations Supported by CTS

Scenarios

Cloud Trace Service (CTS) is a log audit service provided by Huawei Cloud and intended for cloud security. It allows you to collect, store, and query cloud resource operation records and use these records to analyze security, audit compliance, track resources, and locate faults.

With CTS, you can record operations related to SWR for future query, audit, and backtrack.

Key Operations Recorded by CTS

Table 7-1 SWR Enterprise Edition operations recorded by CTS

Operation	Resource Type	Trace Name
Creating an Enterprise Edition instance	instance	createInstance
Listing Enterprise Edition instances	instance	listInstances
Querying the details about an Enterprise Edition instance	instance	getInstance
Deleting an Enterprise Edition instance	instance	deleteInstance
Querying the instance configuration	configuration	getInstanceConfigurations
Modifying the instance configuration	configuration	updateInstanceConfigurations
Querying audit logs	instance	getInstanceAuditLogs

Operation	Resource Type	Trace Name
Querying instance statistics	instance	getInstanceStatistics
Creating a namespace	namespace	createNamespace
Listing namespaces	namespace	listNamespace
Querying the details about a namespace	namespace	getInstanceNamespace
Modifying a namespace	namespace	updateNamespace
Deleting a namespace	namespace	deleteNamespace
Listing repositories	repository	listInstanceRepositories
Listing repositories in an organization	repository	listInstanceRepositories
Querying the details about a repository	repository	getInstanceRepository
Deleting a repository	repository	deleteRepository
Modifying a repository	repository	updateRepository
Listing artifacts	artifact	listInstanceArtifacts
Querying the details about an artifact	artifact	getInstanceArtifact
Deleting an artifact	artifact	deleteArtifact
Listing artifact attachments	artifact	listInstanceAccessories
Querying artifact dependencies	artifact	getInstanceArtifactAddition
Listing artifact tags	tag	listInstanceTags
Querying the details about an artifact tag	tag	getInstanceTag
Deleting an artifact tag	tag	deleteTag
Querying the details about an artifact accessory	tag	getInstanceTagAddition
Creating a retention policy	retentionpolicy	createRetention
Listing retention policies	retentionpolicy	listInstanceRetentionPolicies
Querying the details about a retention policy	retentionpolicy	getInstanceRetentionPolicy
Modifying a retention policy	retentionpolicy	updateRetention
Deleting a retention policy	retentionpolicy	deleteRetention

Operation	Resource Type	Trace Name
Executing a retention policy manually	retentionpolicy	executeRetention
Listing execution records of a retention policy	retention	listInstanceRetentionPolicyExecutions
Listing execution tasks of a retention policy	retention	listInstanceRetentionPolicyExecTasks
Listing execution subtasks of a retention policy	retention	listInstanceRetentionPolicyExecSubTasks
Creating a trigger policy	triggerPolicy	createTriggerPolicy
Listing trigger policies	triggerPolicy	listInstanceWebhooks
Querying the details of a trigger policy	triggerPolicy	getInstanceWebhook
Modifying a trigger policy	triggerPolicy	updateTriggerPolicy
Deleting a trigger policy	triggerPolicy	deleteTriggerPolicy
Listing jobs executed by a trigger policy	triggerPolicy	listInstanceWebhookJobs
Creating an image replication registry	registry	createRegistry
Listing image replication registries	registry	listInstanceRegistries
Querying the details about an image replication registry	registry	getInstanceRegistry
Modifying an image replication registry	registry	updateRegistry
Deleting an image replication registry	registry	deleteRegistry
Creating an image replication policy	replication	createReplicationPolicy
Listing image replication policies	replication	listInstanceReplicationPolicies
Querying the details about an image replication policy	replication	getInstanceReplicationPolicy
Updating an image replication policy	replication	updateReplicationPolicy
Deleting an image replication policy	replication	deleteReplicationPolicy

Operation	Resource Type	Trace Name
Executing an image replication policy	replication	executeReplicationPolicy
Listing execution records of an image replication policy	replication	listInstanceReplicationPolicyExecutions
Stopping an image replication task	replication	stopReplicationExecution
Listing execution tasks of an image replication policy	replication	listInstanceReplicationPolicyExecTasks
Listing execution subtasks of an image replication policy	replication	listInstanceReplicationPolicyExecSubTasks
Listing scan policies	scan	listInstanceScanPolicies
Creating a scan policy	scan	createScanPolicy
Querying the details of a scan policy	scan	getInstanceScanPolicy
Modifying a scan policy	scan	updateScanPolicy
Deleting a scan policy	scan	deleteScanPolicy
Executing a scan policy	scan	executeScanPolicy
Listing the execution records of a scan policy	scan	listInstanceScanPolicyExecutions
Listing the execution tasks of a scan policy	scan	listInstanceScanPolicyExecTasks
Listing image signature policies	signature	listInstanceSignPolicies
Creating an image signing policy	signature	createSignaturePolicy
Querying the details about an image signing policy	signature	getInstanceSignPolicy
Updating an image signing policy	signature	updateSignaturePolicy
Deleting an image signing policy	signature	deleteSignaturePolicy
Executing an image signing policy manually	signature	executeSignaturePolicy
Listing the execution records of an image signing policy	signature	listInstanceSignPolicyExecutions
Listing execution tasks of an image signing policy	signature	listInstanceSignPolicyExecTasks

Operation	Resource Type	Trace Name
Listing the execution subtasks of an image signing policy	signature	listInstanceSignatureExecution-Subtasks
Listing blocking policies	block	listInstanceBlockPolicies
Creating a blocking policy	block	createBlockPolicy
Querying the details about a blocking policy	block	getInstanceBlockPolicy
Modifying a blocking policy	block	updateBlockPolicy
Deleting a blocking policy	block	deleteBlockPolicy
Listing the execution records of a blocking policy	block	listInstanceBlockPolicyRecords
Creating a temporary access credential	TempCredentialAuth	createTempCredentialAuthPolicy
Creating a long-term access credential	LongTermCredentialAuth	createLongTermCredentialAuthPolicy
Listing long-term access credentials	LongTermCredentialAuth	listInstanceLTCredentials
Enabling or disabling a long-term access credential	LongTermCredentialAuth	updateLongTermCredentialAuthPolicy
Deleting a long-term access credential	LongTermCredentialAuth	deleteLongTermCredentialAuthPolicy
Listing jobs	jobs	listInstanceJobs
Querying the details about a job	jobs	getInstanceJobs
Deleting a job	jobs	deleteJob
Listing private network access rules	IntranetEndpoint	listInstanceInternalEndpoints
Creating a private network access rule	IntranetEndpoint	createInternalEndpoint
Querying the details about a private network access rule	IntranetEndpoint	getInstanceInternalEndpoint
Deleting a private network access rule	IntranetEndpoint	deleteInternalEndpoint
Updating the status of the trustlist configuration for public network access	endpointPolicy	enableEndpointPolicy disableEndpointPolicy

Operation	Resource Type	Trace Name
Updating the trustlist configuration for public network access	endpointPolicy	updateEndpointPolicy
Querying the trustlist configuration for public network access	endpointPolicy	getInstanceEndpointPolicy
Listing resource instances	instance	listInstanceResourceInstances
Querying the number of resource instances	instance	getInstanceResourceInstances-Count
Creating resource tags in batches	tms	createResourceTags
Deleting resource tags in batches	tms	deleteResourceTags
Querying project tags	tms	getInstanceProjectTags
Querying resource tags	tms	getInstanceResourceTags
Listing resource instances	tms	listInstanceResourceInstances
Querying the number of resource instances	tms	getInstanceResourceInstances-Count
Creating resource tags in batches	resourceTag	createResourceTags
Deleting resource tags in batches	resourceTag	deleteResourceTags
Querying project tags	tms	getInstanceProjectTags
Querying resource tags	resourceTag	getInstanceResourceTags
Creating an image tag immutability policy	immutableRule	createImmutableRule
Listing image tag immutability policies	immutableRule	listImmutableRules
Updating an image tag immutability policy	immutableRule	updateImmutableRule
Deleting an image tag immutability policy	immutableRule	deleteImmutableRule
Creating a domain name	DomainName	addDomainName
Deleting a domain name	DomainName	deleteDomainName
Updating a domain name	DomainName	updateDomainName
Listing domain names	DomainName	listDomainNames

Operation	Resource Type	Trace Name
Pulling an image	Manifest	GetInstanceManifest
Pushing an image	Manifest	PutInstanceManifest
Creating a repository	Repository	CreateInstanceRepository

7.2 Viewing Logs in CTS

Scenarios

After you enable CTS, the system starts recording operations performed on SWR resources. CTS stores operation records generated within a week.

This section describes how to view the records on the CTS console.

Procedure

Step 1 Log in to the CTS console. In the upper right corner, click **Go to Old Edition**.

Step 2 In the navigation pane, choose **Trace List**.

Step 3 Set the filter criteria and click **Query**.

The following filters are available:

- **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**

  Select the desired filter criteria from the drop-down lists, and set **Trace Type** to **Management** and **Trace Source** to **SWR**.

  If you set **Search By** to **Resource ID**, you need to enter a resource ID. Only whole word match is supported.

- **Operator**: Select a specific operator from the drop-down list.
- **Trace Status**: Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.
- Time range: You can select **Last 1 hour**, **Last 1 day**, **Last 1 week**, or **Customize** in the upper right corner.

Step 4 Locate a record and click ∨ to view its details.

Step 5 Click **View Trace** in the **Operation** column. The trace structure details are displayed.

----End