Tencent Container Registry

Operation Guide

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

Creating an Enterprise Edition Instance

Last updated：2024-08-01 15:36:46

Overview

This document introduces how to purchase a Tencent Container Registry (TCR) Enterprise Edition instance on the TCR purchase page.

Prerequisites

Before purchasing a TCR Enterprise Edition instance, complete the following tasks:

Sign up for a Tencent Cloud account and complete identity verification.

Activate Cloud Object Storage (COS), which is used to store image data.

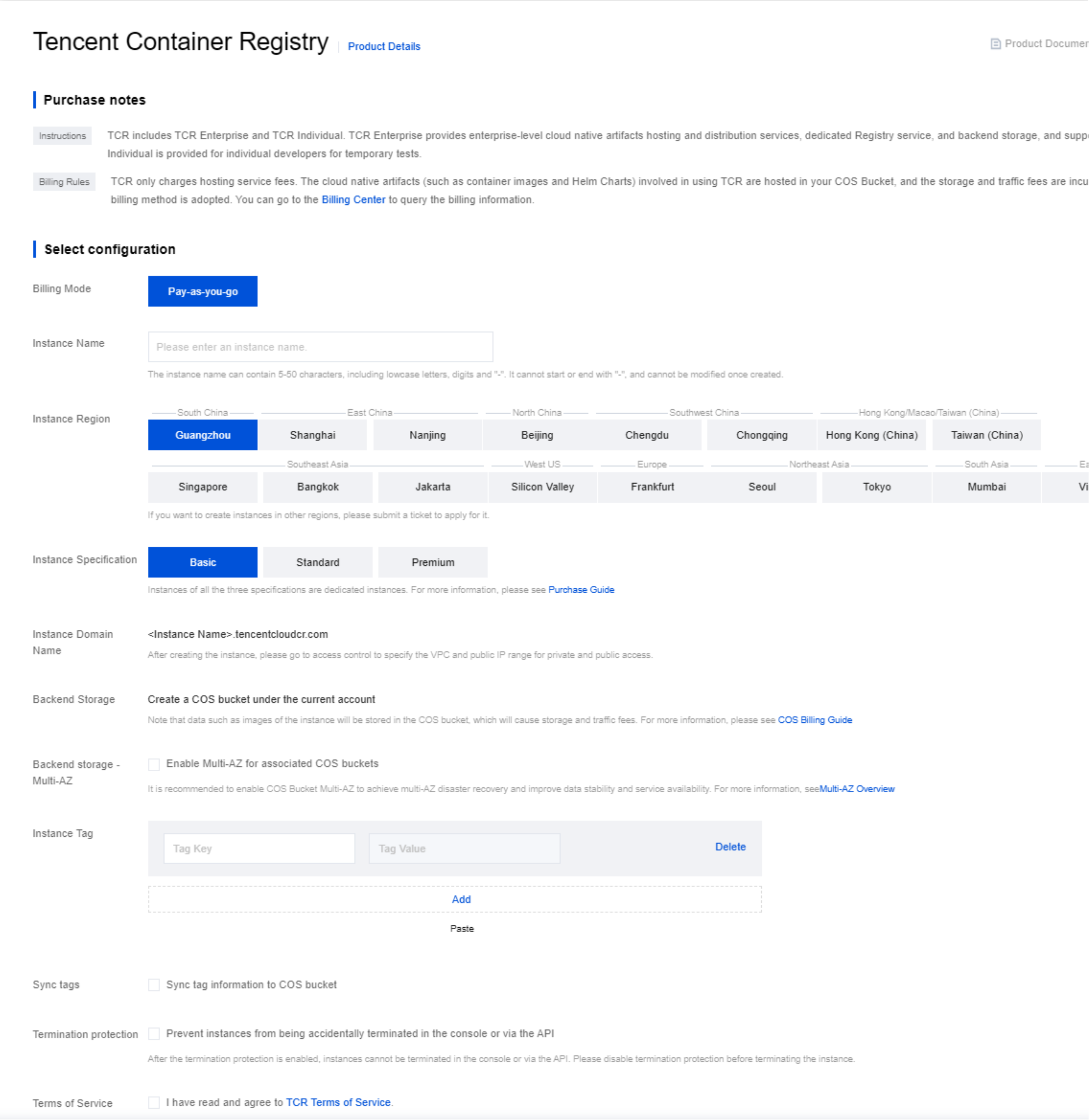
Activate Virtual Private Cloud (VPC) and Private DNS, which are used for image push and pull in a VPC.

Activate the TCR service in the console and grant required operation permissions on your COS and VPC resources.

Directions

Creating via the console

1. Log in to the Tencent Cloud console, choose **Products** > **Basic** > **Container** > TCR**, and click** Buy Now to go to the TCR console.
2. Click **Instance management** in the left sidebar to go to the **Instance management** page and click **Create**.
3. On the **TCR Purchase** page, purchase an instance. You can use the following information for reference:



Tencent Container Registry | Product Details

Product Documer

Purchase notes

Instructions: TCR includes TCR Enterprise and TCR Individual. TCR Enterprise provides enterprise-level cloud native artifacts hosting and distribution services, dedicated Registry service, and backend storage, and supp Individual is provided for individual developers for temporary tests.

Billing Rules: TCR only charges hosting service fees. The cloud native artifacts (such as container images and Helm Charts) involved in using TCR are hosted in your COS Bucket, and the storage and traffic fees are incu billing method is adopted. You can go to the Billing Center to query the billing information.

Select configuration

Billing Mode: Pay-as-you-go

Instance Name: Please enter an instance name.
The instance name can contain 5-50 characters, including lowcase letters, digits and "-". It cannot start or end with "-", and cannot be modified once created.

Instance Region:

South China	East China		North China	Southwest China		Hong Kong/Macao/Taiwan (China)	
Guangzhou	Shanghai	Nanjing	Beijing	Chengdu	Chongqing	Hong Kong (China)	Taiwan (China)

Southeast Asia			West US	Europe	Northeast Asia		South Asia	Ea
Singapore	Bangkok	Jakarta	Silicon Valley	Frankfurt	Seoul	Tokyo	Mumbai	Vi

If you want to create instances in other regions, please submit a ticket to apply for it.

Instance Specification: Basic | Standard | Premium
Instances of all the three specifications are dedicated instances. For more information, please see Purchase Guide

Instance Domain Name: <Instance Name>.tencentcloudcr.com
After creating the instance, please go to access control to specify the VPC and public IP range for private and public access.

Backend Storage: Create a COS bucket under the current account
Note that data such as images of the instance will be stored in the COS bucket, which will cause storage and traffic fees. For more information, please see COS Billing Guide

Backend storage - Multi-AZ: □ Enable Multi-AZ for associated COS buckets
It is recommended to enable COS Bucket Multi-AZ to achieve multi-AZ disaster recovery and improve data stability and service availability. For more information, seeMulti-AZ Overview

Instance Tag: Tag Key | Tag Value | Delete
Add
Paste

Sync tags: □ Sync tag information to COS bucket

Termination protection: □ Prevent instances from being accidentally terminated in the console or via the API
After the termination protection is enabled, instances cannot be terminated in the console or via the API. Please disable termination protection before terminating the instance.

Terms of Service: □ I have read and agree to TCR Terms of Service.

Billing Mode: TCR is billed in pay-as-you-go mode. For more information, see Billing Overview.

Instance Name: Enter a custom instance name. The name must be globally unique and cannot be identical with an existing instance name of your own or another user. This name is used as the access domain name of this TCR instance. **The name cannot be modified after the instance is purchased.** We recommend that you use an abbreviation that combines the company name and instance region or project as the instance name.

Instance Region: Select a region where you want to deploy the instance. **The region cannot be modified after the instance is purchased**. Select the region based on the location of the container cluster resources.

Instance Specification: Select the instance specifications that you want to purchase. Different instance specifications have different instance performance levels and quotas. Make your choices based on the specification comparison on the page.

Instance Domain Name: The instance domain name that is automatically generated. Its prefix is the same as the instance name. **The instance domain name cannot be modified after the instance is purchased.** This domain name is used when you run the `docker login` command to log in to the instance.

Backend Storage: When an instance is purchased, a Tencent Cloud COS bucket is automatically purchased and associated under the current account. Images and other data in the instance will be stored in the bucket, and storage and traffic costs will be generated. For more information, see COS Billing Guide. After instance purchase, you can go to the COS console to view the bucket. Avoid mistakenly deleting the bucket because data such as images hosted in the instance cannot be recovered after the bucket is deleted.

COS multi-AZ: Optional. COS provides the multi-AZ storage architecture. We recommend that you enable the COS multi-AZ feature to achieve multi-AZ disaster recovery for higher data reliability and service availability. This feature incurs higher storage costs. For more information, see MAZ Feature Overview.

Instance Tag: Bind the newly created instance to a Tencent Cloud tag. You can also bind and edit tags on the instance details page after you purchase the instance.

4. Read and agree to the TCR Service Agreement.

Enterprise Edition instances are billed differently based on their region and specifications. Confirm the selected specifications and configuration fees after configuring the basic information.

5. After checking the selected option, click **Buy Now** to purchase the Enterprise Edition instance you have selected and configured.

6. You can view the instance purchase progress on the **Instance management** page. If the instance status is **Running**, the instance is purchased and is running properly.

Note:

If it takes an unusually long time to purchase an instance or the displayed status is **Abnormal**, please submit a ticket.

Creating via API

You can also create an instance by calling the `CreateInstance` API. For more information, see CreateInstance.

Overview

If you purchase Enterprise Edition instances in pay-as-you-go mode, fees will be generated on an hourly basis after an instance is created, and the specific fees will be displayed on the purchase page. You can go to the Billing Center to view the fees generated by this service. If you have any doubt about the fees, please submit a ticket to contact us.

Access Configuration

Credential Access Control

Obtaining an Instance Access Credential

Last updated：2022-01-17 18:37:51

Overview

This document describes how to obtain an access credential for a TCR Enterprise Edition instance in Tencent Container Registry (TCR). To push and pull container images, you must first run the `docker login` command on the access client and enter your username and password (the credential) to log in to the instance.

TCR Enterprise Edition instances support both a long-term access credential and a temporary login command, in which:

- **Long-term access credential**: is permanently valid after being generated. It can be disabled and deleted. Long-term access credentials can be applied in scenarios such as early-stage testing, CICD assembly lines, and container cluster image pull.

  > Note：
  > Please keep the access credential properly after it is generated. If it is lost, disable or delete it promptly.

- **Temporary login token**: is valid for 1 hour and cannot be disabled or terminated after being generated. It can be applied in scenarios such as temporary use and one-time external authorization. Production clusters with high security requirements can also use this method through regular refreshing.

Prerequisite

Before obtaining an access credential for a TCR Enterprise Edition instance, you must complete the following preparations.

- Purchasing Instances.
- To obtain the access credential through an API, you need to obtain the API key for calling API 3.0.

Directions

Obtaining a long-term access credential

1. Log in to the TCR console and click **Access Credential** in the left sidebar.
2. On the "Access Credential" page, choose a region and an instance. Click **Create**.
3. In the "Create Access Credential" window that pops up, complete the following steps to obtain an access credential:
    i. In the "Create Access Credential" step, enter the usage of the credential in **Usage Description**, and click **Next**.
    ii. In the "Save Access Credential" step, click **Save Access Credential** to download the credential. **Please save the access credential properly. You will not be able to get it again.**
4. After the creation, you can view, disable, or delete the credential on the **Access Credential** page.

Obtaining a temporary login token

1. Log in to the TCR console and click **Access Credential** in the left sidebar.
2. On the "Access Credential" page, choose a region and an instance. Click **Generate Temp Login Token**.
3. In the "Temp login token" window that pops up, click **Copy login token** to obtain the temporary access credential.

Creating via API

You can also create an instance access credential via CreateInstanceToken API.

Subsequent Operations

Refer to Logging in to the TCR instance to log in to the TCR Enterprise Edition instance.

Notes

A long-term access credential will be created automatically in some scenarios:

1. When you install the TCR plugin in a TKE cluster, a long-term access credential will be created automatically for the selected instance. This credential will not be terminated automatically when the plugin is deleted. If you do not want to use it any more, you need to delete it manually.
2. When you use an image to build or deliver the pipeline feature, a dedicated access credential will be auto-created and provided to CODING DevOps service to push the auto-built images. Do not delete the access credential directly, otherwise, it will cause the failure of existing image building configuration.

Managing Service Accounts

Last updated：2023-04-25 16:23:30

Overview

To push/pull container images, you need to log in to the instance first with the access credential. TCR supports credentials of user accounts and service accounts. This document describes how to manage service accounts, which is applicable to CI/CD automation scenarios.

A user account is bound with your Tencent Cloud account. The username must be the same as the Tencent Cloud account ID, and the password is generated randomly. The permission of the user account is controlled by the CAM permission of the associated Tencent Cloud account. When the associated Tencent Cloud account is deleted or disabled, the user account goes invalid. This can cause image push/pull failures in Kubernetes clusters or CI/CD scenarios. For more information, see Managing User Accounts.

For CI/CD scenarios or you want to configure permissions on the namespace level, we recommend using the service account. Service Account supports the following features:

Custom username and password

Namespace-specific read/write permission configuration

Custom validity period. You can disable a service account temporarily.

Note:

1. TCR is unable to verify the individual identity of the service account owner. If you need to track and audit the image pull/push events, please use the user account.

2. The permission configuration of a service account prevails the CAM permissions. It means that service account can perform namespace-specific operations that do not allowed by the associated Tencent Cloud account. This brings the risk of broken access control. We recommend only assign the service account to the administrators of the instance.

Prerequisites

Purchase a TCR Enterprise Edition instance.

To obtain the access credential via API, obtain the API key for calling API 3.0.

Note:

Access Credential is now only available to beta users. To try it out, submit a ticket.

Directions

Creating a service account

1. Log in to the TCR console and choose **Access credential** > **Service accounts** in the left sidebar.
2. On the **Service accounts** page, select a region and an instance, and click **Create**.
3. On the **Create service account** page, set the parameters as instructed below:

Name (Required): Custom name of the account. It supports [a-z], [0-9] and [._-], and must start with a letter or digit. The prefix `tcr$` is automatically added to the name to mark it as a service account. For example, if you enter `robot-demo` , the actual username is `tcr$robot-demo` .

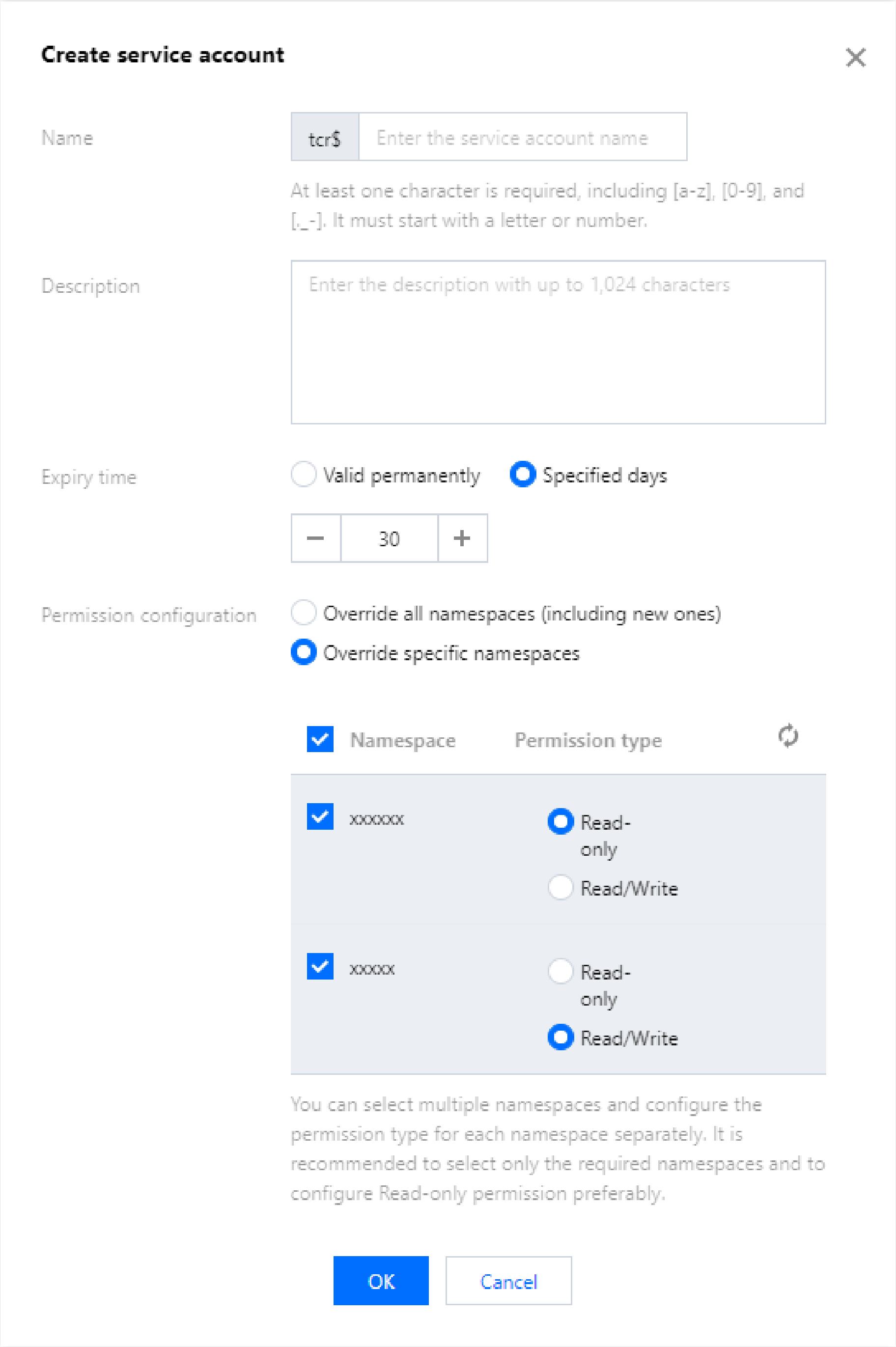
Description: Enter the account description.

Validity: Select **Permanent** or specify a validity period (in days). The default value is 30 days.

Permission configuration: Configure the namespace-specific permission. Select namespaces based on the principle of least privilege.

Namespace: Select target namespaces

Permission type: Select **Read-only** or **Read/Write**. In the **Read-only** mode, image push is not supported.



Create service account

Name: tcr$ Enter the service account name

At least one character is required, including [a-z], [0-9], and [._-]. It must start with a letter or number.

Description: Enter the description with up to 1,024 characters

Expiry time: ○ Valid permanently ● Specified days

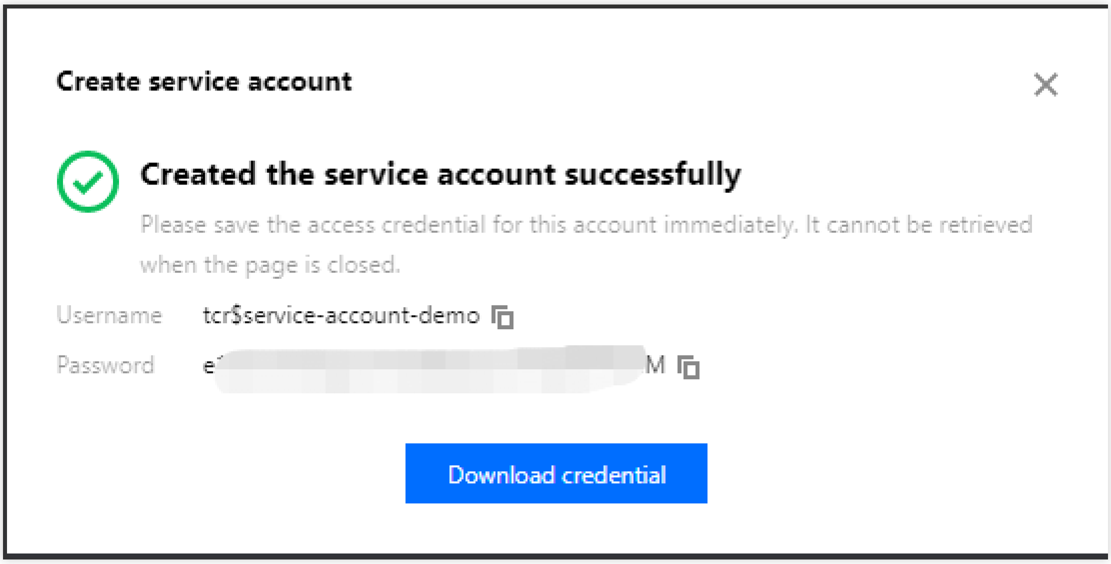
− 30 +

Permission configuration: ○ Override all namespaces (including new ones) ● Override specific namespaces

☑ Namespace	Permission type
☑ xxxxxx	● Read-only ○ Read/Write
☑ xxxxx	○ Read-only ● Read/Write

You can select multiple namespaces and configure the permission type for each namespace separately. It is recommended to select only the required namespaces and to configure Read-only permission preferably.

4. Note down the username and password immediately after the account is created. This page will be displayed only once and the credential information cannot be retrieved after the page is closed.



Create service account

Created the service account successfully

Please save the access credential for this account immediately. It cannot be retrieved when the page is closed.

Username tcr$service-account-demo

Password e[redacted]M

Managing service accounts

1. Log in to the TCR console and choose **Access credential** > **Service accounts** in the left sidebar.
2. On the **Service account** page, select the region and instance name.

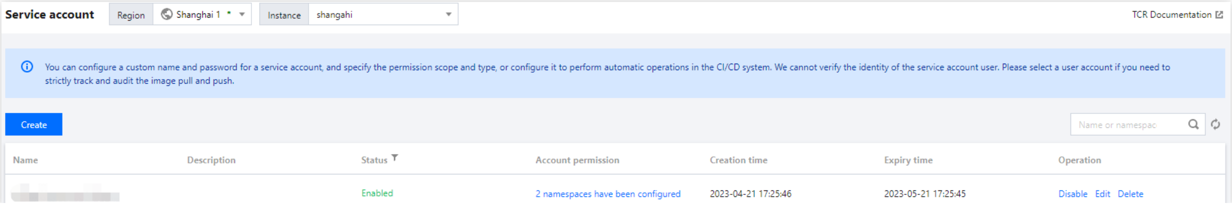
Check existing service accounts

Check the permissions of service accounts

Modify the service account configuration (except the account name)

Enable/Disable service accounts. Note that after an account is disabled, you cannot use it to push or pull images.

Delete service accounts. Note that after an account is deleted, you cannot use it to push or pull images.



Service account Region Shanghai 1 Instance shangahi

You can configure a custom name and password for a service account, and specify the permission scope and type, or configure it to perform automatic operations in the CI/CD system. We cannot verify the identity of the service account user. Please select a user account if you need to strictly track and audit the image pull and push.

Name	Description	Status	Account permission	Creation time	Expiry time	Operation
[redacted]		Enabled	2 namespaces have been configured	2023-04-21 17:25:46	2023-05-21 17:25:45	Disable Edit Delete

Managing an Access Credential

Last updated：2023-05-08 15:44:59

Overview

To push/pull container images, you need to log in to the instance first with the access credential. TCR supports credentials of user accounts and service accounts. This document describes how to manage user accounts.

If you want to use multiple sub-accounts to manage and operate your TCR Enterprise Edition instance, such as pushing or pulling images, you must first log in as the account admin and grant each sub-account permissions. For more information, see TCR Enterprise Authorization Management. When a user logs in to the TCR console by using a sub-account, a user account is generated. This user account is a Docker Registry access credential associated with the user's identity, because the username of the access credential is the same as the ID of the Tencent Cloud sub-account. The user uses this user account to log in to the repository, and push and pull images. Operations of this user account on images are recorded and can be traced back to the user for internal auditing.

When you create a user account, you can create a temporary login token or long-term access credential. We recommend that you create a temporary login token for temporary image push or pull to avoid data security risks due to unexpected credential leakage.

Long-term access credential: A long-term access credential is permanently valid, and can be disabled or deleted. You can use the long-term access credential in scenarios such as early-stage testing, continuous integration and continuous deployment (CI/CD), and image pull in a container cluster.

Notes

Keep the access credential properly. If it is lost, disable or delete it promptly.

Temporary login token: A temporary login token is valid for 1 hour and cannot be disabled or terminated. You can use the temporary login token in scenarios such as one-time external authorization, or in a production cluster with high security requirements by regular refreshing.

Prerequisites

You have purchased a TCR Enterprise Edition instance.

To obtain the access credential through an API, you must obtain the API key that is required for calling v3.0 APIs.

Directions

Obtaining a long-term access credential

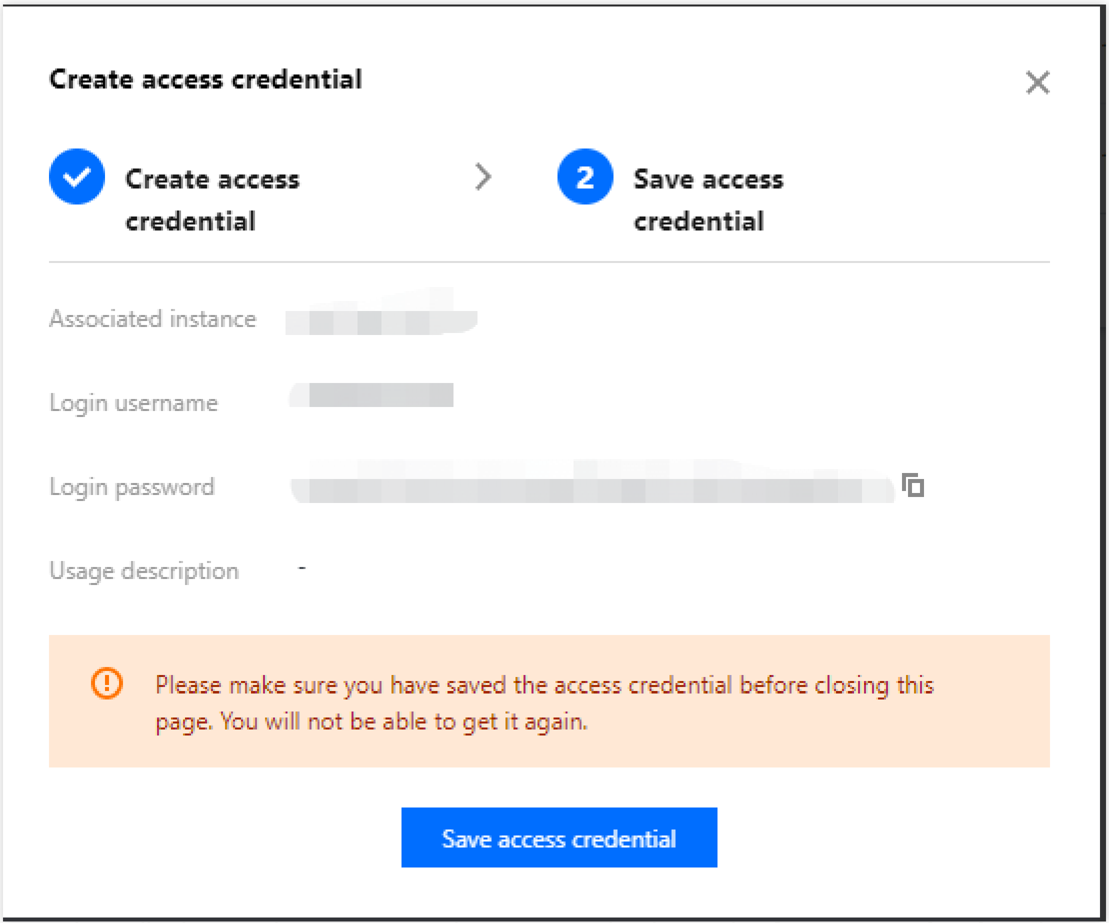
1. Log in to the TCR console and choose **Access credential** > **User accounts** in the left sidebar.

2. On the **User accounts** page, select a region and an instance, and click **Create**.

3. On the **Create Access Credential** page, perform the following steps:

3.1 In the **Create Access Credential** step, specify the purpose of the credential in **Usage Description**, and click **Next**.

3.2 In the **Save Access Credential** step, click **Save Access Credential** to download the access credential. **This is your only chance to download the access credential. Save it properly.**



Create access credential

Associated instance [redacted]

Login username [redacted]

Login password [redacted]

Usage description -

Please make sure you have saved the access credential before closing this page. You will not be able to get it again.

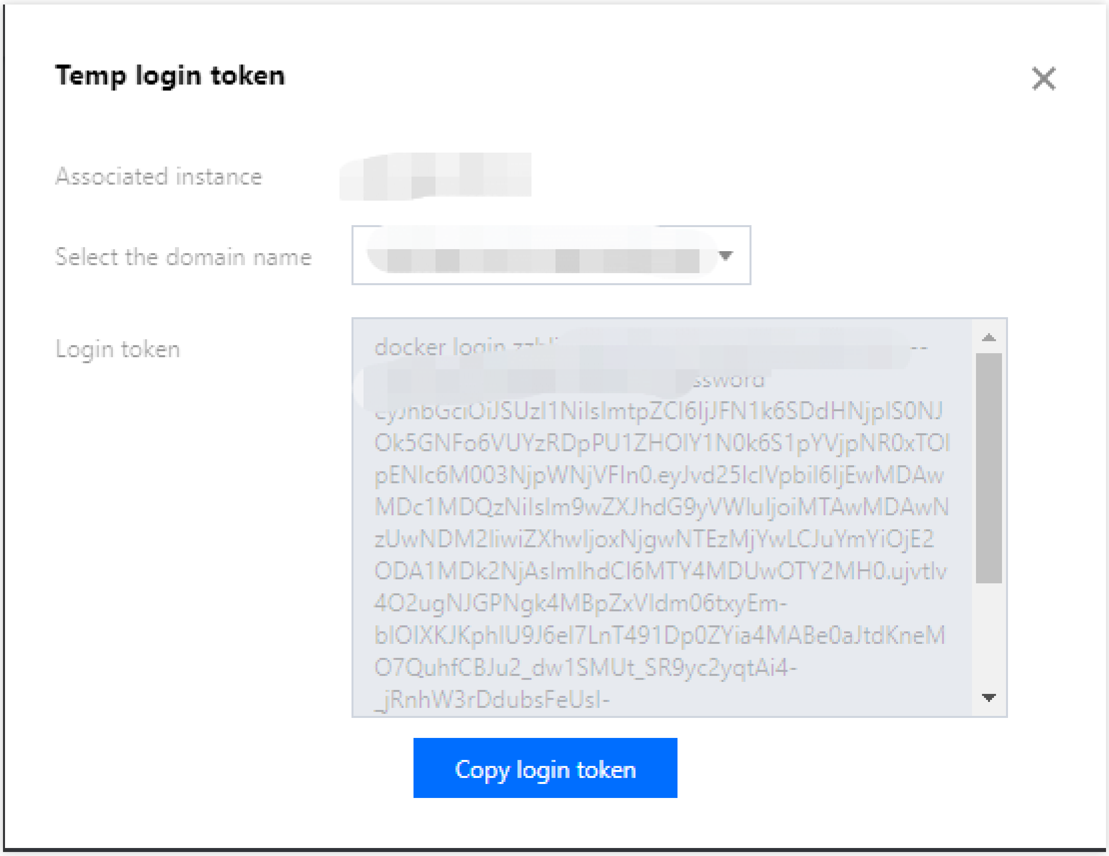
4. You can view, disable, or delete a created access credential on the **Access credential** tab.

Obtaining a temporary login token

1. Log in to the TCR console and choose **Access credential** > **User accounts** in the left sidebar.

2. On the **User accounts** page, select a region and an instance. Click **Generate Temp Login Token**.

3. On the **Temp login token** page, click **Copy login token** to obtain a temporary access credential.



Temp login token

Associated instance [redacted]

Select the domain name [redacted]

Login token

```
docker login [redacted] --
ssword
eyJhbGciOiJSUzI1NiIsImtpZCI6IjJFN1k6SDdHNjpIS0NJ
Ok5GNFo6VUYzRDpPU1ZHOIY1N0k6S1pYVjpNR0xTOI
pENIc6M003NjpWNjVFIn0.eyJvd25lcIVpbil6IjEwMDAw
MDc1MDQzNiIsIm9wZXJhdG9yVWluIjoiMTAwMDAwN
zUwNDM2IiwiZXhwIjoxNjgwNTEzMjYwLCJuYmYiOjE2
ODA1MDk2NjAsImlhdCI6MTY4MDUwOTY2MH0.ujvtlv
4O2ugNJGPNgk4MBpZxVIdm06txyEm-
bIOIXKJKphIU9J6eI7LnT491Dp0ZYia4MABe0aJtdKneM
O7QuhfCBJu2_dw1SMUt_SR9yc2yqtAi4-
_jRnhW3rDdubsFeUsI-
```

Creating via API

You can also create an instance access credential by calling the `CreateInstanceToken` API. For more information, see CreateInstanceToken.

Related Operations

Log in to the TCR Enterprise Edition instance. For more information, see Logging in to the TCR instance.

Overview

A long-term access credential will be created automatically in some scenarios:

1. When you install the TCR add-on in a TKE cluster, a long-term access credential is automatically created for the selected instance. This credential will not be automatically terminated when the add-on is deleted. If you do not want to use it any more, you need to manually delete it.
2. When you use the image building or delivery pipeline feature, a dedicated access credential is automatically created and provided to the CODING DevOps service to push the auto-built images. Do not delete the access credential directly. Otherwise, the build configurations of the existing images become invalid.

Network Access Control

Network Access Control Overview

Last updated：2022-05-09 12:41:23

Tencent Container Registry (TCR) supports network access control for TCR Enterprise instances. To ensure data security of your image repositories and Helm Charts, the public and private network access entries are disabled for the newly created TCR Enterprise instances by default. This means that all external access requests are denied.

Based on your application requirements, you can configure public and private network access control policies to minimize the application clients that can access the instance and pull/push images. For more information, please see:

- Public network access control
- Private network access control

Configuring Public Network Access Control

Last updated：2022-05-09 12:41:24

Overview

Tencent Container Registry (TCR) Enterprise supports public network access control. An allowlist can be configured to restrict instance access by clients through the Internet, ensuring instance data privacy and security. When a TCR Enterprise instance is created, the public network access entry is disabled by default. This means that you cannot use a development test server to push or pull images over the internet.

This document describes how to configure public network access control for a TCR Enterprise instance.

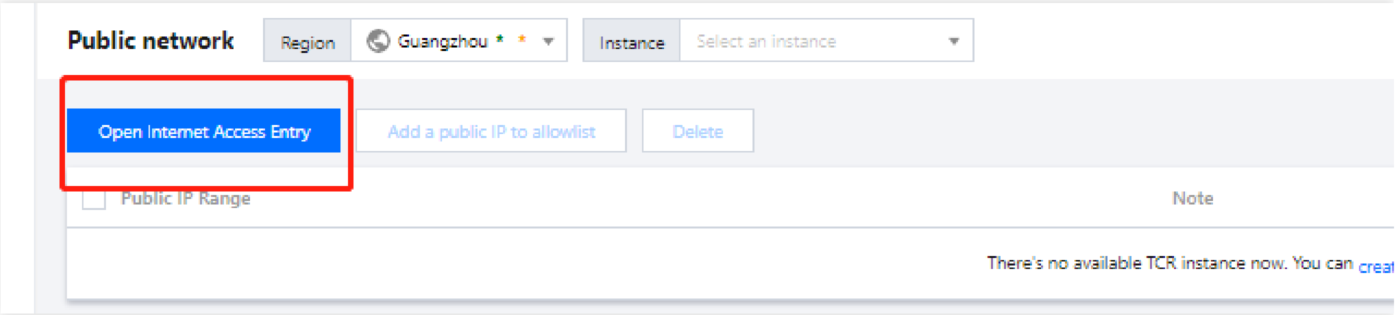
Prerequisites

Before configuring public network access control for a TCR Enterprise instance, you must create a TCR Enterprise instance.

Directions

Enabling the public network access entry

1. Log in to the TCR console and select **Network ACL** > **Public network** in the left sidebar.
2. To change the instance, select the region where the desired instance is deployed and the desired instance name from the **Instance name** drop-down list at the top of the "Public network" page.
3. Select **Open internet access entry** to enable the public network access entry.
   When the status of the button changes from **Enabling** to **Close internet access entry**, and **Add a public IP to allowlist** is in a selectable status, it indicates that the entry is enabled, as shown in the following figure.
   After the entry is enabled, all internet access requests are still denied.



There's no available TCR instance now. You can creat

Configuring the access policy

1. On the “Public network” page, click **Add a public IP to allowlist**. Configure the public IP range and notes in the pop-up window.
   - **Associated instance**: The target instance, for which the public network access policy is configured. To change the instance, select the desired instance name from the "Instance name" drop-down list at the top of the "Public network" page.
   - **Method**: You can manually configure public IP addresses or import the public IP list from an existing security group.
     - **Manually configure**: The public IP address range that is allowed to access the instance. The value can be a single IPv4 address or CIDR block, for example, `192.168.0.0/24` . We do not recommend that you enter `0.0.0.0/0` to accept all internet access requests to the instance.
     - **Import existing**: All allowed addresses in the inbound rules will be deduplicated and imported to the allowlist. After importing, you can further modify the list manually. Please note that the modified security group configurations cannot be synced automatically, and you need to re-import them again.
   - **Note**: Notes about the access policy. This parameter is optional.
2. Click **OK**. The public network access allowlist policy is added and takes effect.

If the public network access entry cannot be enabled or the allowlist policy cannot be created, you can submit a ticket.

Private Network Access Control

Last updated：2023-05-08 16:07:51

Overview

Tencent Container Registry (TCR) Enterprise Edition supports private network access control. A private network access linkage can be used to restrict instance access by clients in a Virtual Private Cloud (VPC). In actual production scenarios involving container computing, pulling container images through a VPC can effectively improve the pulling speed and reduce public network bandwidth costs. TCR allows users to connect their VPCs to a TCR Enterprise Edition instance to implement private network access and access control.

This document describes how to configure private network access control for a TCR enterprise edition instance. After completing the following configuration, you can use a Cloud Virtual Machine (CVM) in a specified VPC to pull images from a TCR instance through the private network, or pull container images in TKE and other container clusters through the cluster private network. For more information, see TKE Clusters Use the TCR Addon to Enable Secret-free Pulling of Container Images via Private Network.

Prerequisites

Make sure that the following conditions are met before configuring private network access control for a TCR Enterprise Edition instance:

You have purchased a TCR Enterprise Edition instance.

If you are using a sub-account, the sub-account must have obtained operation permissions on the corresponding instance. For more information, see TCR Enterprise Authorization Management.

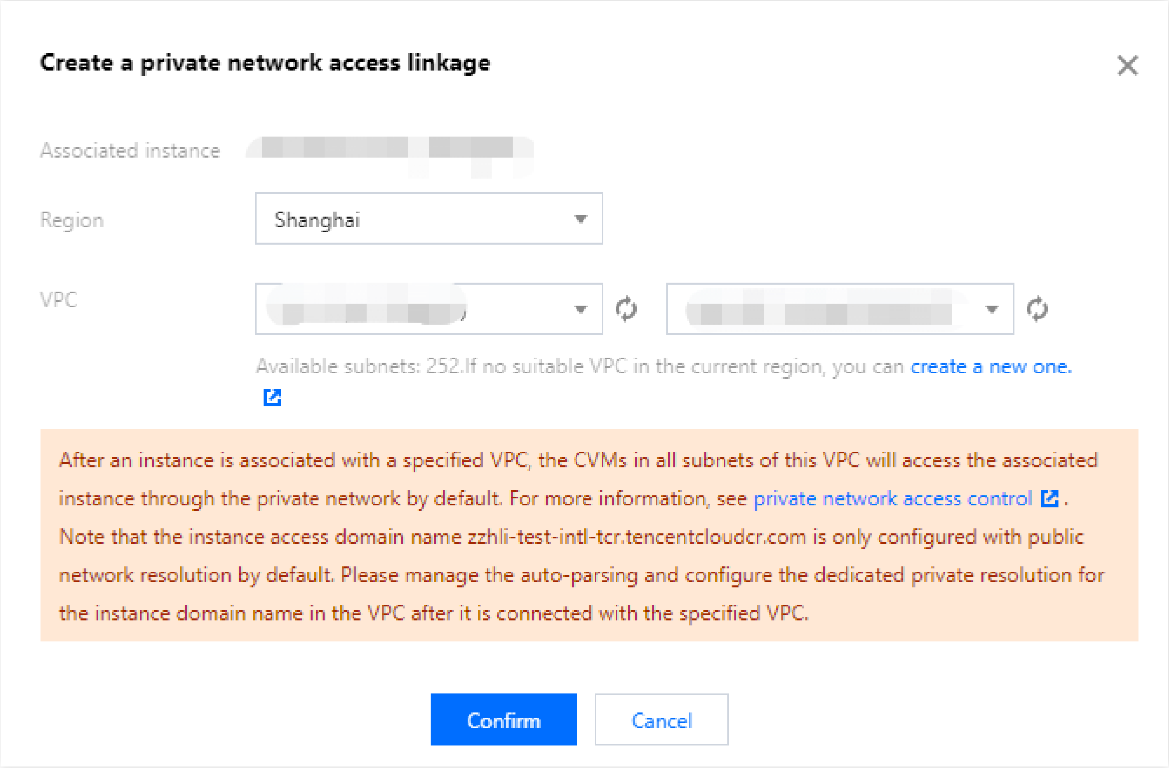
You have activated the VPC service and created a VPC and a subnet in the region where the TCR Enterprise Edition instance is deployed.

You have activated the Private DNS service.

Directions

Creating an access linkage

1. Log in to the TCR console and choose **Access Control** > **Private Network Access** in the left sidebar.
2. On the **Private Network Access** page, click **Create**.
3. In the **Create a private network access linkage** pop-up window, specify a VPC and a subnet, as shown in the figure below:



Create a private network access linkage

Associated instance [redacted]

Region Shanghai

VPC [redacted] [redacted]

Available subnets: 252.If no suitable VPC in the current region, you can create a new one.

After an instance is associated with a specified VPC, the CVMs in all subnets of this VPC will access the associated instance through the private network by default. For more information, see private network access control . Note that the instance access domain name zzhli-test-intl-tcr.tencentcloudcr.com is only configured with public network resolution by default. Please manage the auto-parsing and configure the dedicated private resolution for the instance domain name in the VPC after it is connected with the specified VPC.

Associated Instance: Target instance for which the private network access policy is configured. To change the instance, select another instance name from the **Instance Name** drop-down list at the top of the **Private Network Access** page.

Region: Region where the VPC to access resides, which is the same as the region where the current instance is deployed by default. If the multi-region replication feature is enabled for the current instance and replicas are configured for the instance in multiple regions, you can select the region where a replica is deployed to access the VPC of the replica. For more information, see Configuring Instance Replication.

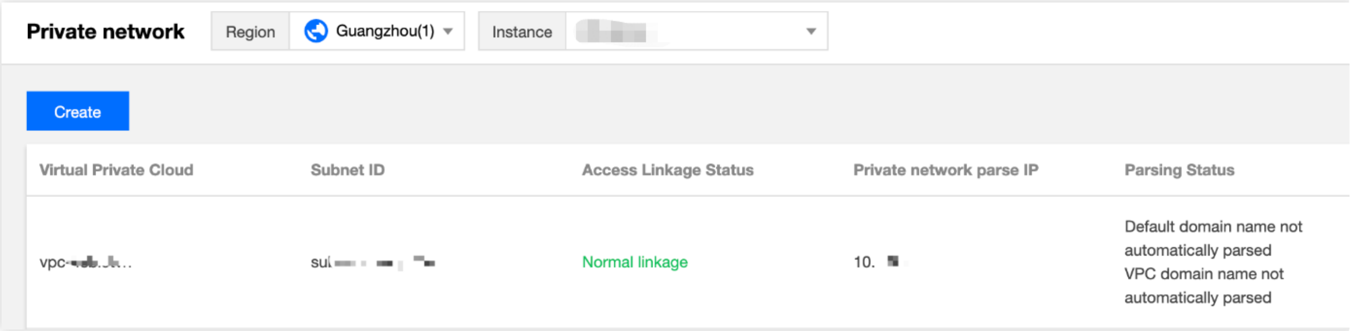
Virtual Private Cloud:

First, select the VPC that you want to connect to. The drop-down list displays all available VPCs in the region of the current instance.

Then, select a subnet with usable private IPs in the VPC. Creating a private network access linkage occupies a private IP in the subnet. The IP is also used as the destination IP for private network resolution of the instance domain name. The subnet is only used to assign private network access addresses. After the linkage is created, CVMs in subnets of the VPC can access the TCR Enterprise Edition instance through the linkage.

4. Click **OK** to start creating the private network access linkage.

If the **access linkage status** changes to **Normal linkage** and the **private network parsing IP** is not empty, the private network access linkage was successfully created.



Private network Region Guangzhou(1) Instance [redacted]

Create

Virtual Private Cloud	Subnet ID	Access Linkage Status	Private network parse IP	Parsing Status
vpc-[redacted]	su[redacted]	Normal linkage	10.[redacted]	Default domain name not automatically parsed VPC domain name not automatically parsed

Note

By default, only resolution for public network access is configured for the access domain name of the instance. After connecting the instance to the specified VPC and obtaining the private network parsing IP, click **Manage Auto-parsing** to configure the dedicated private network parsing for the instance domain name in the VPC.

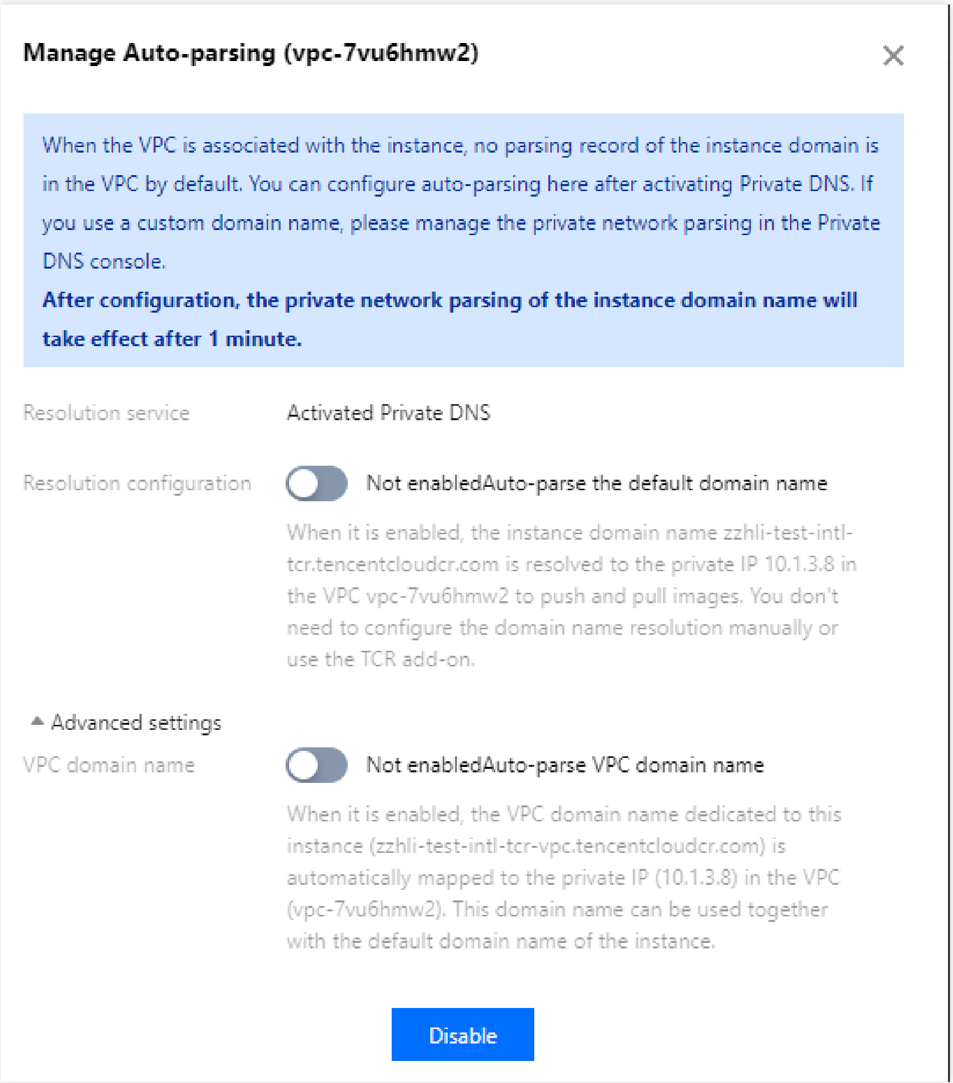
Managing private network parsing

After the private network access linkage is established, CVMs in the associated VPC access the instance through the private network by accessing the private network parsing IP. By default, the default domain name of the instance (for example, tcr-demo.tencentcloudcr.com) and private network domain name (for example, tcr-demo-vpc.tencentcloudcr.com) will not be automatically resolved to the private network parsing IP in the VPC. You need to use the **Manage Auto-parsing** feature to configure private network parsing, or use an external DNS service to manage parsing.

Using Private DNS for automatic configuration (default method)

By default, Tencent Cloud Private DNS is used to configure domain name resolution in VPCs. You need to activate the service before using this feature.

1. Log in to the TCR console and choose **Access Control** > **Private Network Access** in the left sidebar.
2. On the **Private Network Access** page, click **Manage Auto-parsing** next to the created private network access linkage.
3. In the **Manage Auto-parsing** pop-up window that appears, configure domain name resolution for the linkage, as shown in the figure below:



Manage Auto-parsing (vpc-7vu6hmw2)

When the VPC is associated with the instance, no parsing record of the instance domain is in the VPC by default. You can configure auto-parsing here after activating Private DNS. If you use a custom domain name, please manage the private network parsing in the Private DNS console.

After configuration, the private network parsing of the instance domain name will take effect after 1 minute.

Resolution service Activated Private DNS

Resolution configuration Not enabledAuto-parse the default domain name

When it is enabled, the instance domain name zzhli-test-intl-tcr.tencentcloudcr.com is resolved to the private IP 10.1.3.8 in the VPC vpc-7vu6hmw2 to push and pull images. You don't need to configure the domain name resolution manually or use the TCR add-on.

▲ Advanced settings

VPC domain name Not enabledAuto-parse VPC domain name

When it is enabled, the VPC domain name dedicated to this instance (zzhli-test-intl-tcr-vpc.tencentcloudcr.com) is automatically mapped to the private IP (10.1.3.8) in the VPC (vpc-7vu6hmw2). This domain name can be used together with the default domain name of the instance.

DNS Service: Indicates whether Private DNS is activated. If the service is not activated, click **Activate Service** to activate it. No additional fees will be charged on the products using Private DNS.

Resolution Configuration: This feature is disabled by default. You can turn on the switch to enable automatic resolution of the default domain name. After this feature is enabled, the instance domain name will be resolved to the private IP in the VPC instead of being resolved to a public IP of the instance. The instance will be used to push and pull images in the private network without the need to configure the domain name resolution manually or by using the TCR add-on.

Advanced Configuration: You can configure the auto-parsing of VPC domain names. VPC domain names are dedicated domain names in VPCs. You can use a VPC domain name in VPCs to distinguish it from the default domain name used in the public network. By default, an image repository provides the access address of the default domain name and related operation instructions. If you use a dedicated VPC domain name, modify the access address configuration of the image repository.

4. Click **OK**.

Using the TCR add-on for automatic configuration

This solution is applicable to the scenarios where Private DNS has not provided services in the region where the TKE cluster is located. This solution is not recommended by default.
If you are using TKE, refer to TCR to install the TCR add-on in the TKE cluster and select **Enable Private Network Parsing** in the **TCR Component Parameter Setting** window. For nodes in the cluster, this add-on can automatically configure private network resolution for the associated TCR instance. This enables secret-free pulling of images in the instance through the private network.

Manually configuring a CVM host

This solution is applicable to CVMs or nodes located in self-built Kubernetes clusters in VPCs that require temporary access to TCR Enterprise Edition instances. It is not recommended by default.
Here, a Linux CVM is used as an example. Log in to the CVM and run the following command:

```
echo '172.16.1.95 techo-demo.tencentcloudcr.com' >> /etc/hosts
```

Replace `172.21.17.69` and `demo.tencentcloudcr.com` with the private network parsing IP and TCR instance domain name that you use.

Access Permission Configuration Overview

Last updated：2020-07-28 15:55:33

Introduction to Cloud Access Management

Cloud access management (CAM) is a web service provided by Tencent Cloud. It helps users securely manage the permissions for accessing resources under their Tencent Cloud accounts. CAM allows you to create, manage, or terminate users (groups) and controls who can use Tencent Cloud resources through identity management and policy management.

When you use CAM, you can associate a policy with a user or a user group. The policy authorizes or refuses users to use the specified resource to complete the specified task. For more information on CAM policies, refer to Policy Syntax. For more information on how to use CAM policies, refer to Policies.

If you do not need to perform access management of TCR resources for sub-accounts, you can skip this section. This does not affect your understanding and use of other sections of this document.

CAM-based Resource-level Access Control of TCR

Resource-level permissions refer to the capabilities that can specify and allow users to perform specific operations on specific resources. TCR supports resource-level access control of CAM and controls the granularity to the repository level, that is, you can authorize sub-accounts to perform operations on resources in only the specified image repository or the Helm Chart repository by configuring the CAM policy.

Types of resources that can be authorized by TCR in CAM:

Resource Type	Resource Description Method in Authorization Policy
Enterprise edition instance	`qcs::tcr:$region:$account:instance/*`
Enterprise edition repository	`qcs::tcr:$region:$account:repository/*`
Personal edition repository	`qcs::tcr:$region:$account:repo/*`

- `$region` : the region information. For example, `ap-guangzhou` indicates the region of Guangzhou. If the value is null, the field indicates all regions. For the specific list of regions and abbreviations, refer to Regions and Availability Zones.

- `$account` : the root account of the resource owner. The value is expressed as `uin/${uin}` , for example, `uin/12345678` . If the value is null, the field indicates the root account of the CAM user who creates the policy.

  For details on resource description in the authorization policy, refer to Resource Description.

CAM APIs for Enterprise Edition

Last updated：2021-04-08 10:41:06

Instance Management APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateInstance Creating an instance	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstanceStatus Querying the instance status	instance	`qcs::tcr:$region:$account:instance/*` `qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstances Querying the instance information	instance	`qcs::tcr:$region:$account:instance/*` `qcs::tcr:$region:$account:instance/$instanceid`
CreateInstanceToken Creating an instance access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DeleteInstanceToken Deleting a long-term access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
ModifyInstanceToken Updating the instance’s long-term access credential	instance	`qcs::tcr:$region:$account:instance/$instanceid`
DescribeInstanceToken Querying the long-term access credential information	instance	`qcs::tcr:$region:$account:instance/$instanceid`

Namespace APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateNamespace Creating a namespace	repository	`qcs::tcr:$region:$account:repository/$instanceId/$nar`

DeleteNamespace Deleting a namespace	repository	`qcs::tcr:$region:$account:repository/$instanceId/$nar`
ModifyNamespace Updating the namespace information	repository	`qcs::tcr:$region:$account:repository/$instanceId/$nar`
DescribeNamespaces Querying the namespace information	repository	`qcs::tcr:$region:$account:repository/$instanceId/*` `qcs::tcr:$region:$account:repository/$instanceId/$nar`

Image Repository APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateRepository Creating an image repository	repository	`qcs::tcr:$region:$account:repository/$instanceId/$namesp`
DeleteRepository Deleting an image repository	repository	`qcs::tcr:$region:$account:repository/$instanceId/$namesp`
ModifyRepository Updating the image repository information	repository	`qcs::tcr:$region:$account:repository/$instanceId/$namesp`
DescribeImages Querying the container image information	repository	`qcs::tcr:$region:$account:repository/$instanceId/$namesp`
DescribeImages Querying the image repository information	repository	`qcs::tcr:$region:$account:repository/$instanceId/$namesp` `qcs::tcr:$region:$account:repository/$instanceId/$namesp`

TCR Enterprise Authorization Management

Last updated：2023-04-21 16:00:04

This document describes how to enable sub-accounts to view and use the TCR Enterprise related resources through the CAM policy, including specific operation steps and common policy configuration examples.

Note:

If you need the permissions of other Tencent Cloud services when using some features in TCR console such as VPC, CloudAudit, Tag, please see the corresponding CAM Guide in CAM-Enabled Products.

Directions

This document takes the example of "granting the sub-account the read-only permission of an image repository" to introduce how to create a policy.

Instance ID: tcr-xxxxxxxx

Namespace: team-01

Image repository: repo-demo

Creating by policy generator (recommended)

Creating by policy syntax

1. Log in to the CAM console.
2. In the left sidebar, click **Policies** to go to the policy management page.
3. Click **Create custom policy** in the upper-left corner.
4. In the pop-up window, click **Create by policy generator** to go to the **Edit policy** page.
5. Select the service in the Visual Policy Generator, enter the following information, and edit an authorization statement.

Effect: Select **Allow** or **Deny**. Here, we select **Allow**.

Service: Select the service you want to authorize. Here, we select **Tencent Container Registry (tcr)**.

Action: Select the operations you want to authorize. Here, we select **Read**.

Resource: Select all resources or specific resources you want to authorize. Here, we select **Specific resources**, and add the following six-segment resource to restrict the access.

repository: Select the region where the repository resides, and enter the resource path of the repository, for example, `tcr-xxxxxxxx/team-01/repo-demo/*` . You can get the resource path in Image Repository.

repo: It is left empty.

instance: Select the region where the repository resides, and enter the ID of the instance to which the repository belongs, for example, `tcr-xxxxxxxx` . You can get the instance ID in the Instance List.

Condition: It is left empty.

6. Click **Next** to go to the **Associate users/user groups** page.

7. On the **Associate users/user groups** page, add the policy name and description, and you can associate users or user groups for quick authorization at the same time.
8. Click **Complete** to complete the custom policy creation.

1. Log in to the CAM console.
2. In the left sidebar, click **Policies** to go to the policy management page.
3. Click **Create custom policy** in the upper-left corner.
4. In the selection window that pops up, click **Create by policy syntax** to go to the **Select policy template** page.
5. In **Select a template type** section, select **Blank template**.
6. Click **Next** to go to the **Edit policy** page.
7. In the **Edit policy** page, enter the policy name and description, and add the following policy content.

```
{
    "version": "2.0",
    "statement": [{
            "action": [
                "tcr:DescribeRepositories",
                "tcr:PullRepository",
                "tcr:DescribeNamespaces"
            ],
            "resource": [
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01/repo-demo/*"
            ],
            "effect": "allow"
        },
        {
            "action": [
                "tcr:DescribeInstance*"
            ],
            "resource": [
                "qcs::tcr:::instance/tcr-xxxxxxxx"
            ],
            "effect": "allow"
        }
    ]
}
```

8. Click **Complete** to complete the custom policy creation.

Common Policy Configuration

If you need to customize the policy JSON, please see CAM APIs for TCR Enterprise and Syntax Logic.

Preset policy configuration

QcloudTCRFullAccess: Full read/write permission of TCR.
After the policy is bound to a sub-account, the sub-account has all operation permissions for all TCR resources, including TCR Enterprise and TCR Individual.

```
    }]
}
```

```
{
    "version": "2.0",
    "statement": [{
        "action": [
            "tcr:*"
        ],
        "resource": "*",
        "effect": "allow"
    }]
}
```

QcloudTCRReadOnlyAccess: Read-only permission of TCR.

After the policy is bound to a sub-account, the sub-account has the read-only permission for all TCR resources, including the TCR Enterprise and TCR Individual.

```
{
    "version": "2.0",
    "statement": [{
        "action": [
            "tcr:Describe*",
            "tcr:PullRepository*"
        ],
        "resource": "*",
        "effect": "allow"
    }]
}
```

Policy configuration in typical scenarios

Note:

The following scenario policies are only used for TCR Enterprise use cases. For the policies used for TCR Individual, please see Example of Authorization Solution of TCR Individual.

Grant a sub-account all read/write operation permissions for all resources in TCR Enterprise.

```
{
    "version": "2.0",
    "statement": [{
        "action": [
            "tcr:*"
        ],
        "resource": [
            "qcs::tcr:::instance/*",
            "qcs::tcr:::repository/*"
        ],
        "effect": "allow"
        }]
    }
```

Grant a sub-account the read-only permission for all resources in TCR Enterprise.

```
    {
        "version": "2.0",
        "statement": [{
            "action": [
                "tcr:Describe*",
                "tcr:PullRepository*"
            ],
            "resource": [
                "qcs::tcr:::instance/*",
                "qcs::tcr:::repository/*"
            ],
            "effect": "allow"
        }]
    }
```

Grant a sub-account permissions to manage the specified instance, for example, dev-guangzhou, whose instance ID is tcr-xxxxxxxx.

```
{
    "version": "2.0",
    "statement": [{
        "action": [
            "tcr:*"
        ],
        "resource": [
            "qcs::tcr:::instance/tcr-xxxxxxxx",
            "qcs::tcr:::repository/tcr-xxxxxxxx/*"
        ],
        "effect": "allow"
    }]
}
```

Grant a sub-account permissions to manage the specified namespace in the specified instance, for example, team-01 under the instance tcr-xxxxxxxx.

```
{
    "version": "2.0",
    "statement": [{
            "action": [
                "tcr:*"
            ],
            "resource": [
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01",
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01/*"
            ],
            "effect": "allow"
        },
        {
            "action": [
                "tcr:DescribeInstance*"
            ],
            "resource": [
                "qcs::tcr:::instance/tcr-xxxxxxxx"
            ],
            "effect": "allow"
        }
    ]
}
```

Grant a sub-account the read-only permission of an image repository, which means that the sub-account can only pull the images in the image repository instead of deleting a repository, modifying repository attributes, or pushing images, for example, repo-demo in the namespace team-01 under the instance tcr-xxxxxxxx.

```
{
    "version": "2.0",
    "statement": [{
            "action": [
                "tcr:Describe*",
                "tcr:PullRepository"
            ],
            "resource": [
                "qcs::tcr:::instance/tcr-xxxxxxxx",
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01",
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01/repo-demo",
                "qcs::tcr:::repository/tcr-xxxxxxxx/team-01/repo-demo/*"
            ],
            "effect": "allow"
        }
    ]
}
```

Access Domain Name Configuration

Configuring Custom Domain Name

Last updated：2021-12-21 16:42:56

Overview

Tencent Container Registry (TCR) Enterprise Edition allows you to configure and use custom domain names, which facilitates the use of the domain access service uniformly planned by your company. In addition, you can continue using the original domain name after migrating from another image registry service to TCR, which helps maintain the service continuity.

TCR Enterprise Edition instances with any specifications all support configuring multiple domain names without affecting the normal use of existing default domain names of the instances. To use a custom domain name, you need to provide the SSL certificate associated with the domain name and access the instance over HTTPS. This document describes how to use a custom domain name to access a TCR Enterprise Edition instance.

Concepts

Domain name

A domain name is a string of characters separated by dots. A domain name in TCR Enterprise Edition is used to access the instance service and directly determines the access address of an image repository.

SSL certificate

An SSL certificate is used for compliance with the HTTPS protocol, so that TCR Enterprise Edition can implement encrypted transfer and identity verification over the HTTPS protocol, ensuring the transfer security.

DNSPod

DNSPod can route the access traffic to a custom domain name to the corresponding IP address of a TCR Enterprise Edition instance.

Prerequisites

Before configuring and using a custom domain name, you need to complete the following:

- Get a domain name. You can register one in Tencent Cloud Domains.

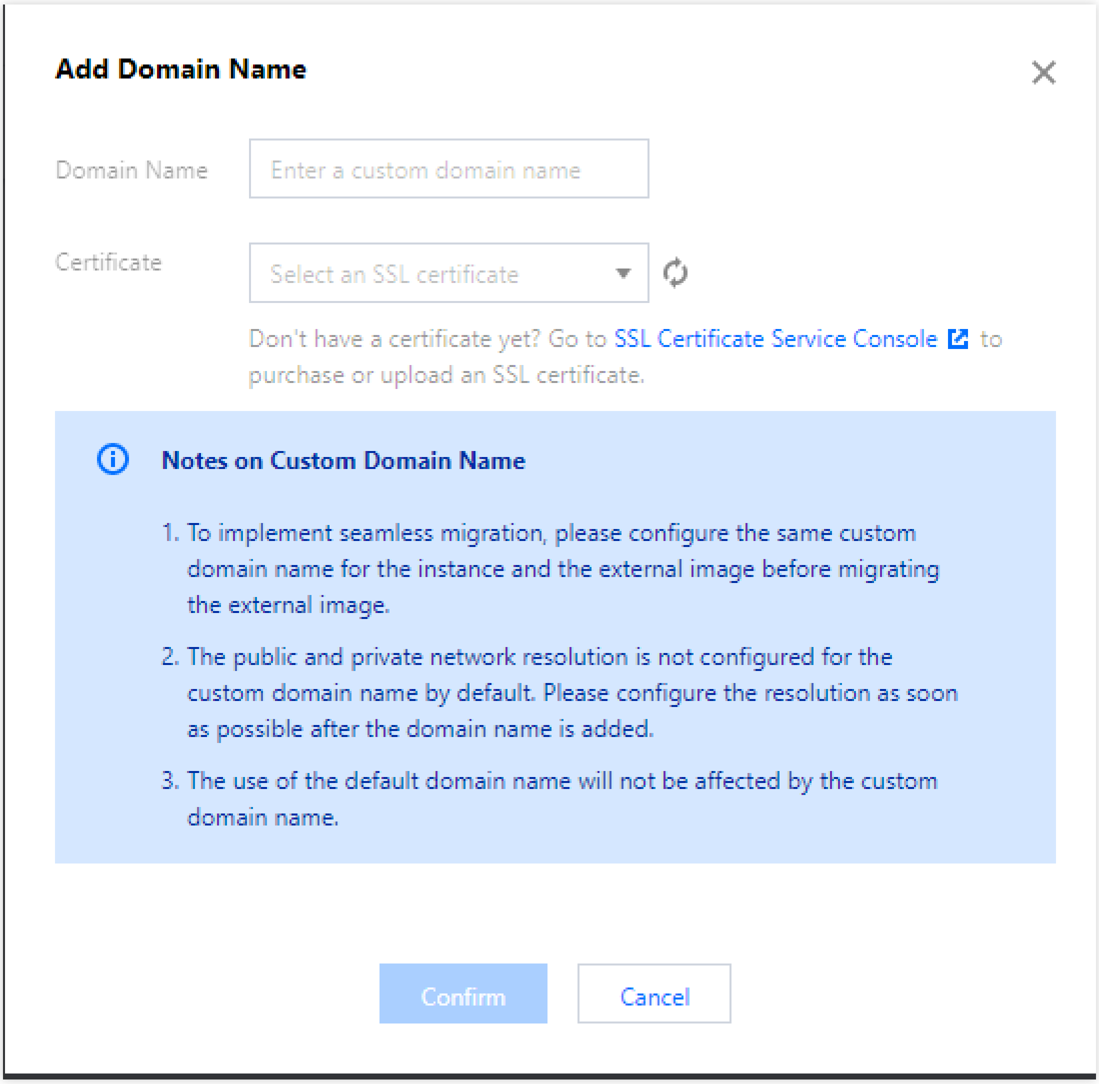
> Note
> - Get an ICP filing for you domain name if you want to use it in a public network environment.
> - You do not need to get an ICP filing if your TCR Enterprise Edition instance is outside the Chinese mainland.

- Get a certificate for the domain name. You can purchase a certificate in SSL Certificates and confirm that it has been bound to the custom domain name to be used by the instance.
- Activate DNSPod. For more information, see Private DNS.

Directions

Creating custom domain name

1. Log in to the TCR console and select **Domain Name Management** on the left sidebar.
2. On the **Domain Name Management** page, select the region and ID of the instance for which you want to add a custom domain name.
3. Click **Add Domain Name*. In the *Add Domain Name** pop-up window, configure the domain name and certificate information as prompted as shown below:



Add Domain Name

Domain Name: Enter a custom domain name

Certificate: Select an SSL certificate

Don't have a certificate yet? Go to SSL Certificate Service Console to purchase or upload an SSL certificate.

Notes on Custom Domain Name

1. To implement seamless migration, please configure the same custom domain name for the instance and the external image before migrating the external image.
2. The public and private network resolution is not configured for the custom domain name by default. Please configure the resolution as soon as possible after the domain name is added.
3. The use of the default domain name will not be affected by the custom domain name.

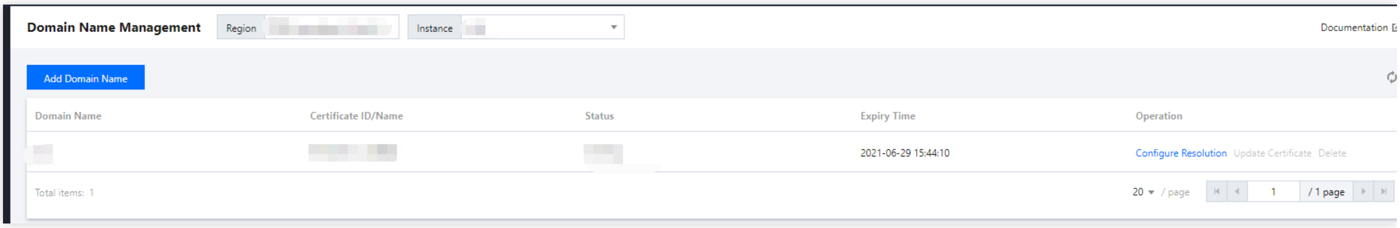
- **Domain Name**: enter the custom domain name to be used. We recommend you use a common domain suffix.
- **Certificate**: select a certificate bound to the custom domain name. Only a certificate managed in SSL Certificates can be selected.

> Note
> If your custom domain name has been filed with MIIT, and a DNS record has been added in the Domains console, you can enter it in the **Domain Name** input box and select a certificate.

4. Click **OK**.

After adding a custom domain name successfully, you can view it on the **Domain Name Management** page.

Then, you can perform the following operations to manage the custom domain name as shown below:



Domain Name	Certificate ID/Name	Status	Expiry Time	Operation
[redacted]	[redacted]	[redacted]	2021-06-29 15:44:10	Configure Resolution Update Certificate Delete

Setting access control and DNS

You can use the custom domain name in the public network or VPC. We recommend you use a VPC to access the instance preferably.

- Private network access
- Public network access

Configuring private network access control

Connect a VPC to the instance and confirm that the private network access IP is generated normally as instructed in Private Network Access Control.

Configuring Private DNS

Go to the Private DNS console, create a private domain with the added custom domain name, and associate it with the connected VPC. Use an A record to configure DNS in the private domain, and use the private network access IP of the created private network access linkage as the record value. For more information, see Private DNS.

Updating domain certificate

If you need to update the certificate bound to your custom domain name for reasons such as certificate expiration or upgrade, go to the **Domain Name Management** page, click **Update Certificate** on the right of the row of the custom domain name and select an SSL certificate again. Certificate update requires redelivery of the SSL certificate information, during which the custom domain name can still be accessed normally.

Deleting custom domain name

On the **Domain Name Management** page, click **Delete** on the right of the row of the specified custom domain name to delete it. Doing so may invalidate the existing container image pull configuration and thus affect application update. Therefore, do so with caution.

Manage Image Repository

Managing Namespaces

Last updated：2021-12-21 16:42:56

Overview

In Tencent Container Registry (TCR) Enterprise Edition, a namespace is used to manage multiple associated image repositories and Helm charts. It does not directly store container images or Helm charts, but can map to teams, product projects, or individuals in an enterprise.

TCR Enterprise Edition instances are exclusive for an enterprise. Therefore, you do not need to worry that a namespace may be occupied by other users when creating the namespace. However, if you create a namespace in a TCR Personal Edition instance, the name of the namespace must be different from that of any existing namespaces. This document describes how to create and manage a namespace in a TCR Enterprise Edition instance.

Prerequisites

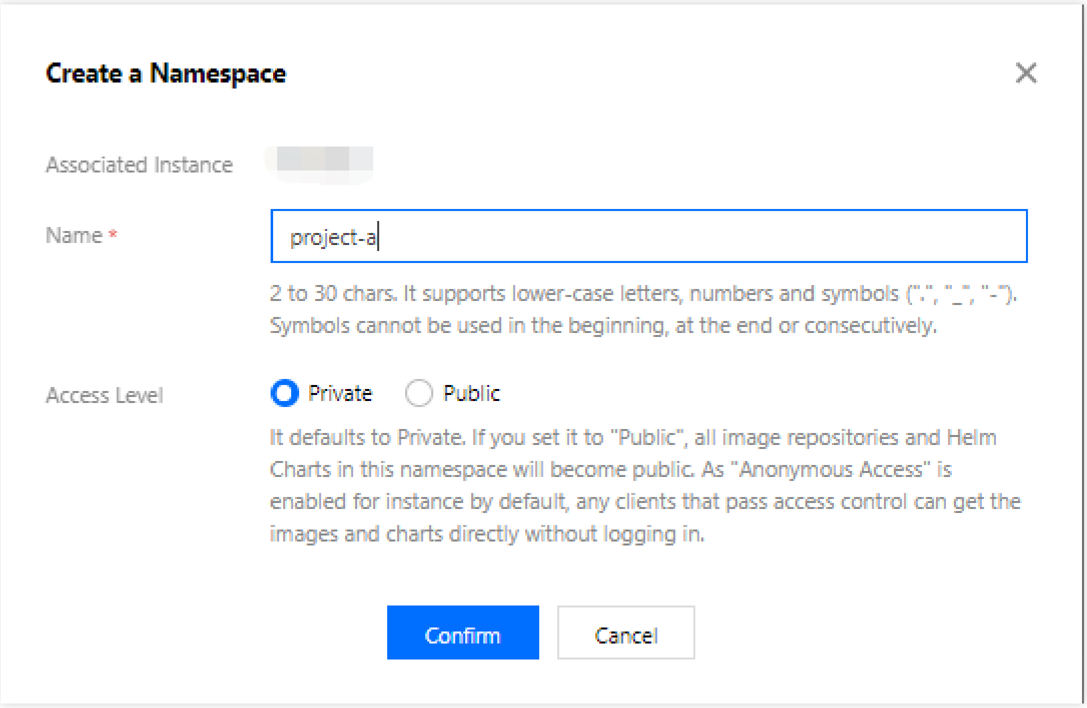
Before creating and managing a namespace in a TCR Enterprise Edition instance, complete the following preparations:

- Purchasing Instances.
- If you are using a sub-account, you must have granted the sub-account required permissions for the instance. For more information, see Example of Authorization Solution of the Enterprise Edition.

Directions

Creating a namespace

1. Log in to the TCR console and select **Namespace** on the left sidebar.
2. On the **Namespace** page, you can view the namespace list of the current instance. To change the instance, select the desired instance name from the **Instance** drop-down list at the top of the page.
3. Click **Create**. In the **Create a Namespace** window, configure the name and access level of the namespace as shown below:



Create a Namespace

Associated Instance [redacted]

Name * project-a

2 to 30 chars. It supports lower-case letters, numbers and symbols (".", "_", "-"). Symbols cannot be used in the beginning, at the end or consecutively.

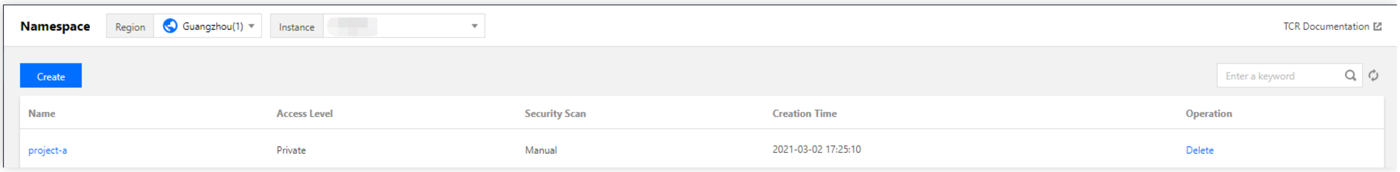
Access Level ● Private ○ Public

It defaults to Private. If you set it to "Public", all image repositories and Helm Charts in this namespace will become public. As "Anonymous Access" is enabled for instance by default, any clients that pass access control can get the images and charts directly without logging in.

- **Associated Instance**: the currently selected instance, to which the created namespace belongs.
- **Name**: the namespace name. It is a string of 2-30 characters. The name can only contain lowercase letters, number, and separators, which are periods (.), underscores (_), and hyphens (-). It cannot start or end with a separator or contain several consecutive separators. We recommend that you set this parameter to the name of an enterprise team or product project. You can also set this parameter to a personal name and use this namespace for personal testing. The namespace names must be unique in an instance.
- **Access Level:** you can select **Private** or **Public**. The default value is **Private**.

  If you set this parameter to "Public", all image repositories and Helm charts in the namespace are public repositories. If anonymous access is also enabled for this instance (which is enabled by default), any clients in the allowlist can pull images and Helm charts without having to log in. You can modify **Access Level** after the namespace is created.

4. Click **OK**.
5. After the namespace is created, you can view the namespace on the "Namespace" page. Then, you can perform the following operations to manage the namespace, as shown in the figure below.



Namespace | Region Guangzhou(1) | Instance [redacted]

Name	Access Level	Security Scan	Creation Time	Operation
project-a	Private	Manual	2021-03-02 17:25:10	Delete

Changing the access level

1. Click the namespace that you want to change the access level and go to its "Basic Information" page.

2. On the "Basic Information" page, click ✎ next to the **Access Level**. Change the public or private attribute of the namespace in the pop-up window.

> Note：
> After the access level is changed, all the image repositories and Helm charts in the namespace immediately inherit this attribute. **Do not change a private namespace to a public namespace unless necessary**.

Changing the security scan mode

1. Click the namespace that you want to change the access level and go to its "Basic Information" page.
2. Click ✎ next to the **Security Scan** to change the security scan mode for container images in this namespace. You can set the security scan mode to **Manually Scan** or **Auto-scan**.

> Note：
> Changing the security scan mode does not affect the existing security scan results.

- **Manually Scan**: to perform a security scan on a specified container image and view the result, you need to go to the **Image Repository** page, select the image, and click **Scan** on the **Tag Management** tab.
- **Auto-scan**: an automatic security scan is triggered when a new image is pushed to any image repository in the current namespace.

Configuring deployment security

An Enterprise Edition instance can block the deployment of high-risk images. For more information, see High-Risk Image Deployment Blocking.

Deleting a namespace

To delete a namespace, select it and click **Delete** next to it. To prevent important data from being deleted by mistake, a namespace that still contains image repositories or Helm charts cannot be deleted.

Basic Image Repository Operations

Last updated：2024-07-01 18:08:02

Overview

In Tencent Container Registry (TCR) Enterprise Edition, an image repository is used to manage container images. A single image repository may contain container images with different tags. An image repository belongs to a namespace and inherits the public or private attribute and security scan triggering mode from its namespace.

An image repository is the minimum unit for permission management in TCR. The instance admin can grant image repository management or read-only permissions to a sub-account. For example, the instance admin can grant the tom sub-account only the permission to pull images from the project-a-frontend image repository, but disallow the sub-account to push or delete images. For more information about other permission management and authorization methods, see TCR Enterprise Authorization Management. This document describes how to create and manage an image repository in a TCR Enterprise Edition instance.

Prerequisites

Make sure that the following conditions are met before creating and managing an image repository in a TCR Enterprise Edition instance:

You have purchased a TCR Enterprise Edition instance.

If you are using a sub-account, the sub-account must have obtained operation permissions on the corresponding instance. For more information, see TCR Enterprise Authorization Management.

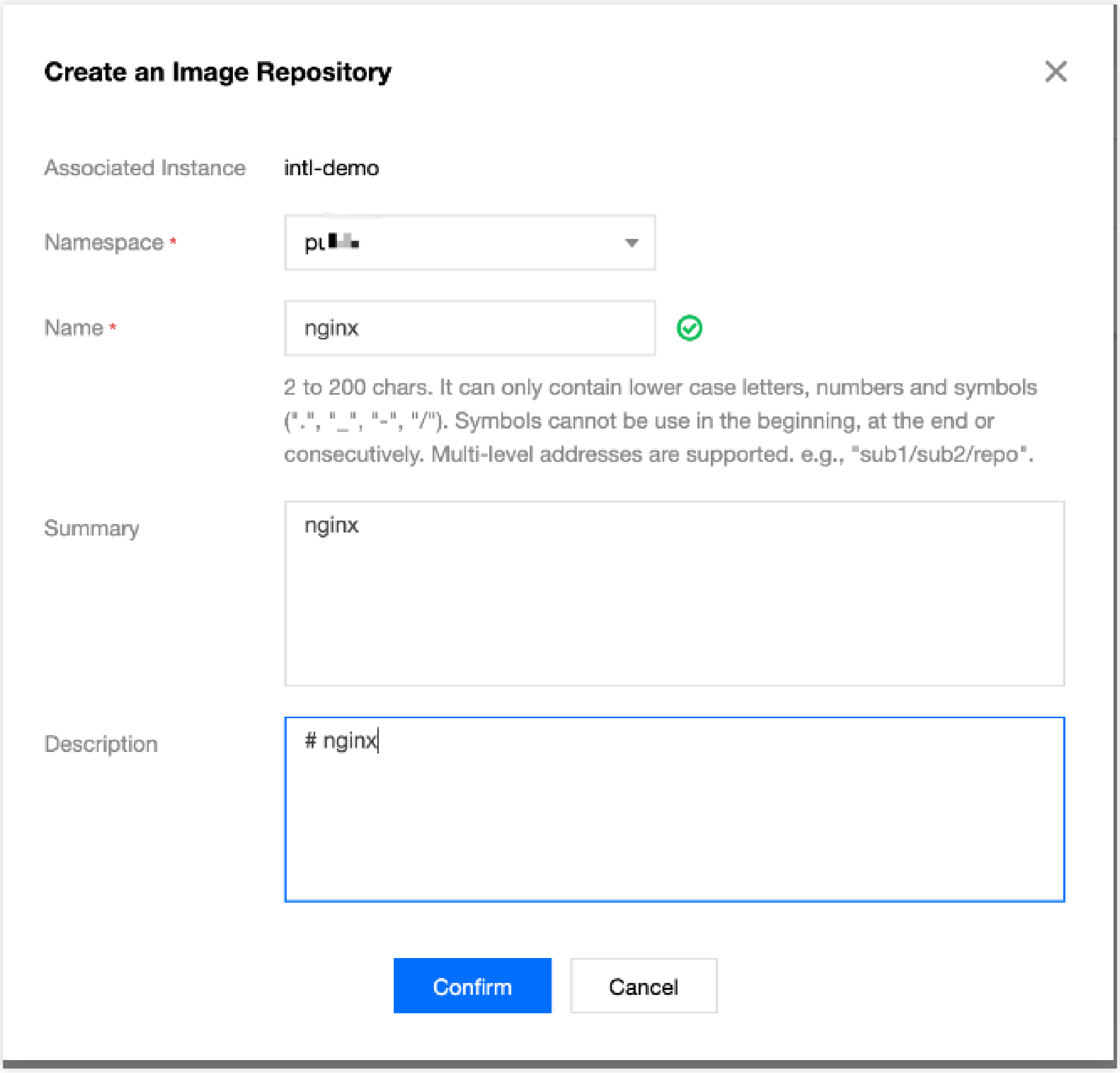
Directions

Creating an image repository

1. Log in to the TCR console and click **Image Repository** in the left sidebar.

On the **Image Repository** page, you can view the image repository list of the current instance. To change the instance, select the required instance name from the **Instance Name** drop-down list at the top of the page.

2. Click **Create**. In the **Create repository** pop-up window, configure the image repository, as shown in the figure below:



Create an Image Repository

Associated Instance intl-demo

Namespace * pu[redacted]

Name * nginx

2 to 200 chars. It can only contain lower case letters, numbers and symbols (".", "_", "-", "/"). Symbols cannot be use in the beginning, at the end or consecutively. Multi-level addresses are supported. e.g., "sub1/sub2/repo".

Summary nginx

Description # nginx

Associated Instance: Currently selected instance, to which the image repository belongs.

Namespace: Namespace to which the image repository belongs. If the list is empty, create a namespace in the instance.

Name: Name of the image repository. The value must be 2 to 200 characters in length and can only contain lowercase letters, numbers, and separators including periods (.), underscores (_), hyphens (-), and slashes (/). It cannot start or end with a separator or contain several consecutive separators. The name can be a multi-level path, such as `team-01/front/nginx` . You can set the name flexibly based on your business requirements.

Image Source: Source of the image. You can select **Local** or **Platform**.

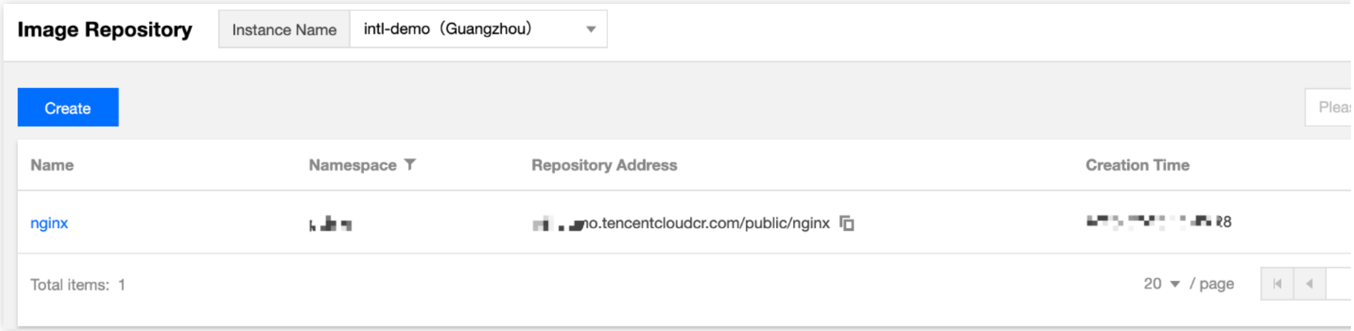
Brief Description: Brief description of the image repository. It is a string that can be up to 100 characters in length. You can edit the description after the image repository is created.

Detailed Description: Detailed description of the image repository. This parameter supports the Markdown syntax. It is a string that can be up to 1000 characters in length. You can edit the description after the image repository is created.

3. Click **OK**.

Performing basic operations on the created image repository

You can view the created image repository on the **Image Repository** page and perform the following basic operations on the image repository:



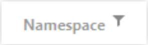
Name	Namespace	Repository Address	Creation Time
nginx	[redacted]	[redacted]o.tencentcloudcr.com/public/nginx	[redacted]

Total items: 1

20 / page

Filtering namespaces

On the **Image Repository** page, you can click



and select the target namespace from the drop-down list.

Viewing image repository details

Click the name of the image repository. The details page is displayed, where you can manage image tags and edit the basic information of the image repository.

Deleting the image repository

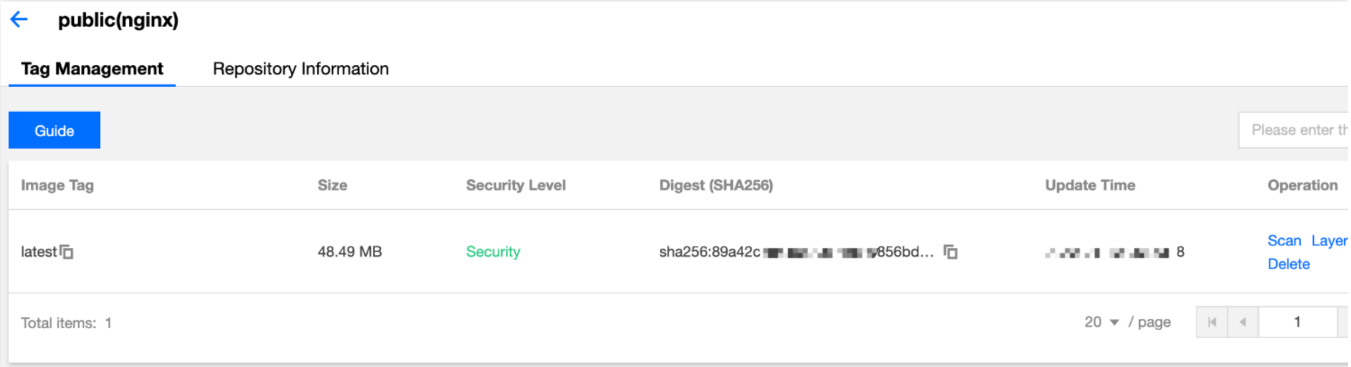
You can click **Delete** next to the image repository to delete it. Carefully confirm the deletion before deleting to prevent important data from being deleted by mistake.

Note:

After the image repository is deleted, **all container images in the image repository are deleted**.

Managing image tags

Click the name of a specified image repository. The repository details page is displayed, and the **Tag Management** tab is selected by default. On this tab, you can manage all the image tags in the repository, perform security scans, and view the layer information, as shown in the figure below:



public(nginx)

Image Tag	Size	Security Level	Digest (SHA256)	Update Time	Operation
latest	48.49 MB	Security	sha256:89a42c[redacted]856bd...	[redacted] 8	Scan Layer Delete

Total items: 1 20 / page 1

Filtering image tags

In the search box in the upper-right part of the tag list, you can enter an image tag to search for this image tag. Fuzzy search is supported.

Obtaining the pulling command

You can click **Copy command** next to a target image tag to copy the pulling command of the image tag.

Performing a security scan

You can click **Scan** for a target image tag to perform a security scan. After the scan result of the **security level** property appears, click

👁 to view the details.

Viewing the image layer information

You can click **Layer Information** next to a target image repository to view the image layer information in the pop-up window.

Deleting an image tag

You can click **Delete** next to a target image tag to delete this image tag. Carefully confirm the deletion before deleting to prevent important data from being deleted by mistake.

Note:

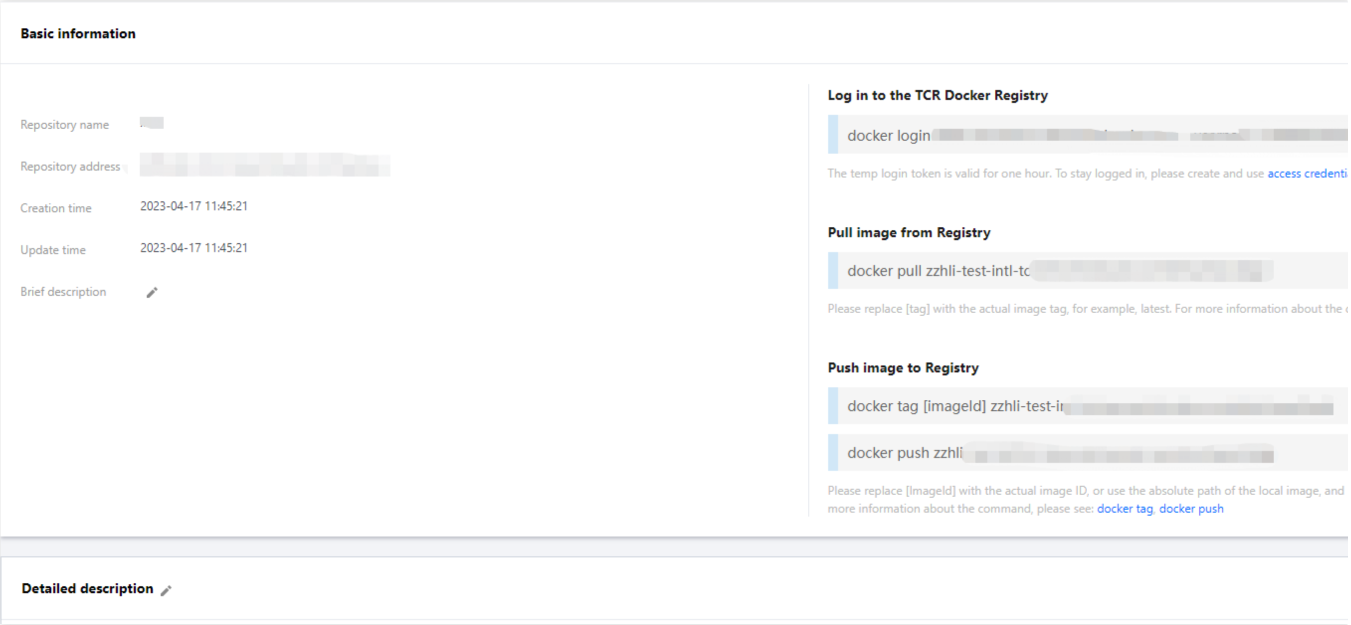
When a specified image tag is deleted, other image tags that have the same image ID as the deleted image tag may also be deleted. Consequently, these image tags will become unavailable.

Building images

You can compile the source code managed on GitHub, GitLab.com, private Gitlab, Gitee, TGit, or CODING to build images.

Editing the repository information

On the details page of an image repository, select the **Repository Information** tab. On this tab, you can view and edit the basic information of the image repository, as shown in the figure below:



Basic information

Repository name [redacted]

Repository address [redacted]

Creation time 2023-04-17 11:45:21

Update time 2023-04-17 11:45:21

Brief description ✎

Log in to the TCR Docker Registry

```
docker login [redacted]
```

The temp login token is valid for one hour. To stay logged in, please create and use access credenti

Pull image from Registry

```
docker pull zzhli-test-intl-tc [redacted]
```

Please replace [tag] with the actual image tag, for example, latest. For more information about the

Push image to Registry

```
docker tag [imageId] zzhli-test-in [redacted]
docker push zzhli [redacted]
```

Please replace [ImageId] with the actual image ID, or use the absolute path of the local image, and more information about the command, please see: docker tag, docker push

Detailed description ✎

Editing the brief description

Click ✎ next to **Brief Description**, edit the brief description, and then click **Save**.

Editing the detailed description

Click ✎ next to **Detailed Description**, edit the detailed description, and then click **Save**. The Markdown syntax is supported for the detailed description. After you save the description, you can view the effect of text rendering.

Image Distribution

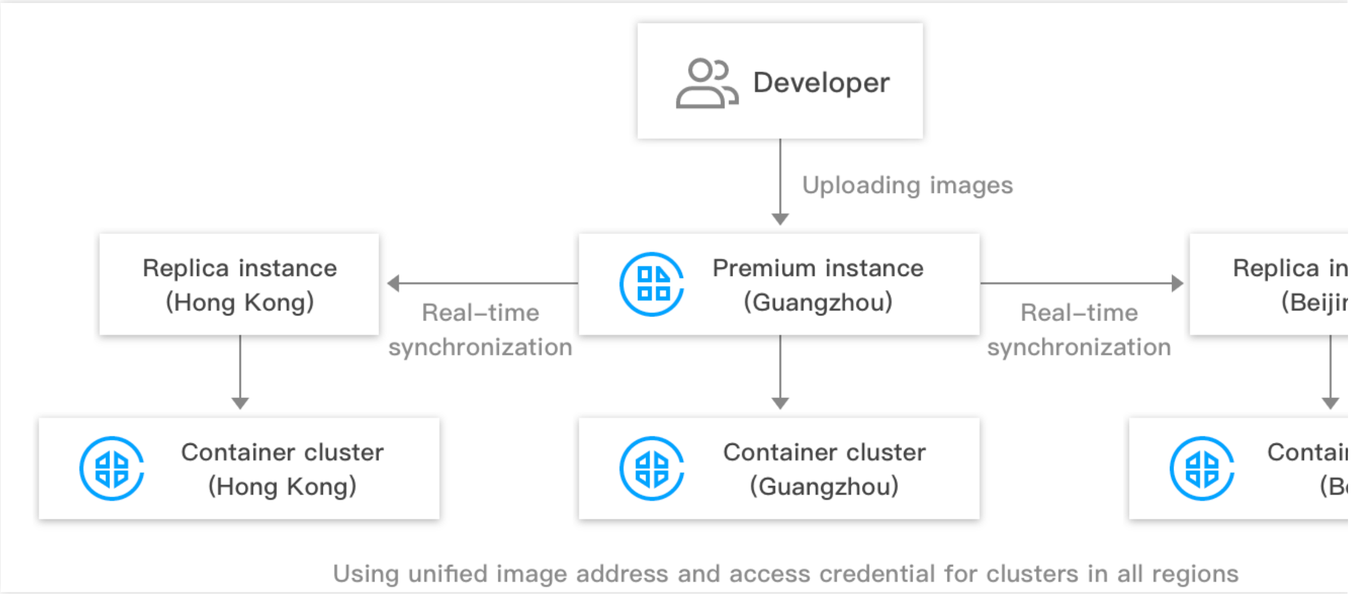
Intra-Instance Multi-Region Image Replication

Last updated：2023-05-08 15:44:59

Overview

Tencent Container Registry (TCR) allows users to create replicas of a premium instance in multiple regions with the same access domain name and credential as the original premium instance. It realizes the single-region upload, multi-region high-speed real-time synchronization, and image pulling from the nearest region over the private network. Compared with the cross-instance synchronization, this feature can unify the publish configuration of multi-regional clusters, improve the cross-region synchronization speed of cloud native artifacts, and help users realize the global synchronization update of service applications.

The instance replication feature allows users to create replicas of a premium instance in multiple regions. The replica instances will synchronize data with the primary instance in real time. Users can use the domain name and access credentials of the primary instance to access the replica instances through the private network. By using the instance replication feature, users can uniformly manage the application images of the multi-regional services, and do not need to purchase multiple Enterprise Edition instances, which can reduce the usage cost, increase the speed of container image distribution, and simplify the deployment and configuration.


Developer
Uploading images
Replica instance (Hong Kong)
Premium instance (Guangzhou)
Replica in (Beijir
Real–time synchronization
Real–time synchronization
Container cluster (Hong Kong)
Container cluster (Guangzhou)
Contai (B
Using unified image address and access credential for clusters in all regions

Prerequisites

Make sure that the following conditions are met before creating and managing the replica instances of a TCR Enterprise Edition instance:

You have purchased an Enterprise Edition instance with premium specification.

If you are using a sub-account, the sub-account must have obtained operation permissions on the corresponding instance. For more information, see TCR Enterprise Authorization Management.

Use Limits

1. You can pull images from, but not push images to, instances. To push and pull images across regions at the same time, use the cross-instance (account) image synchronization feature, which requires you to create an instance in each region and configure a synchronization rule.
2. The instance replication feature relies on the support of the underlying networks and is subject to security compliance requirements.

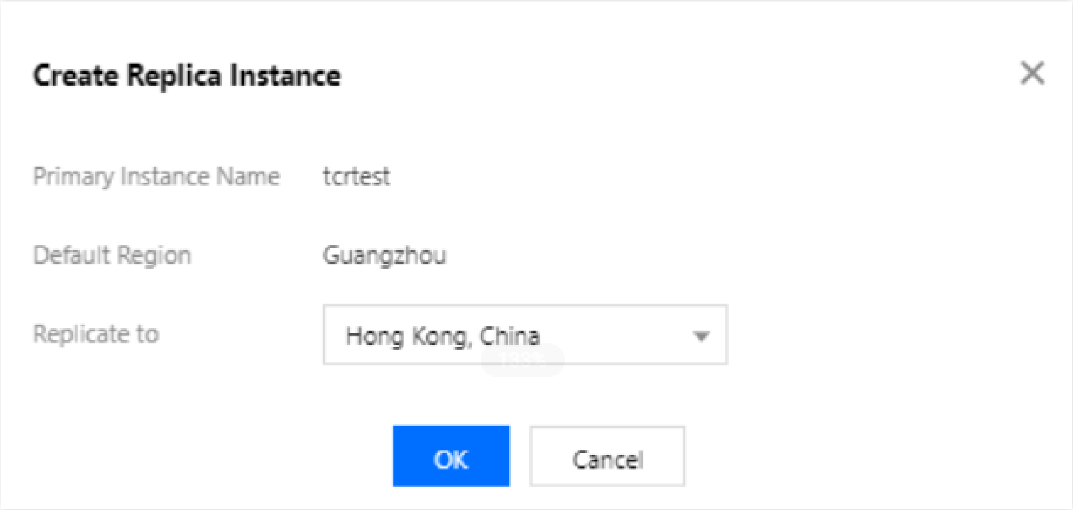
Cross-border instance replication is not supported. For example, if your primary instance is in the South China (Guangzhou) region, you cannot create a replica that belongs to an overseas region.

The instance replication feature is not supported in the Taiwan (China) region.

Directions

Creating and managing a replica instance

1. Log in to the TCR console and choose **Synchronization and Replication** > **Replication** in the left sidebar.
2. On the **Replication** page, select a region and an instance, and click **Create**.
3. In the **Create Replica Instance** window, complete the following configurations:



Create Replica Instance

Primary Instance Name: tcrtest

Default Region: Guangzhou

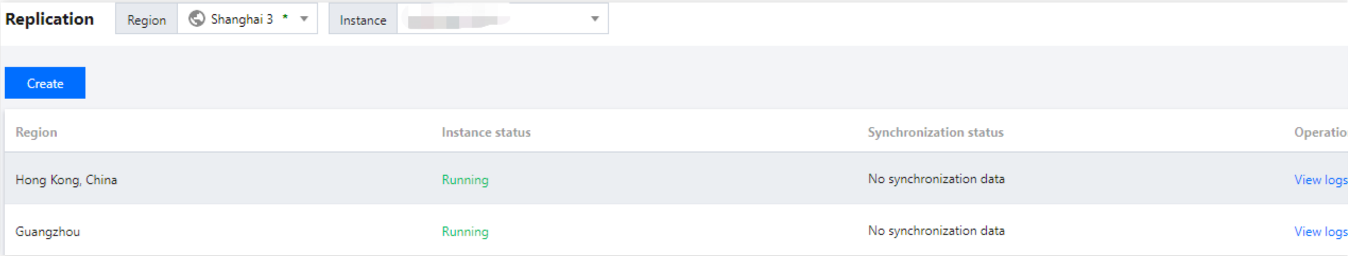
Replicate to: Hong Kong, China

Primary Instance Name: Name of the currently selected premium instance.

Default Region: Region where the currently selected premium instance is located.

Replicate to: Region where the replica instance is located, which cannot be the same as the region of the current primary instance.

4. Click **OK**.



Replication　Region　Shanghai 3　Instance [redacted]

Create

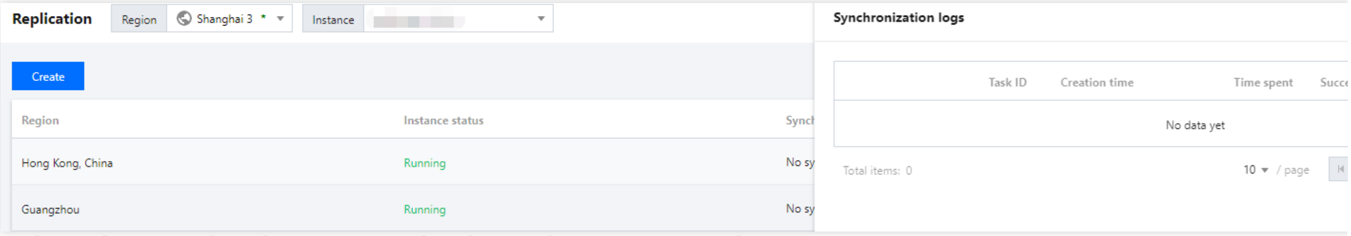
Region	Instance status	Synchronization status	Operatio
Hong Kong, China	Running	No synchronization data	View logs
Guangzhou	Running	No synchronization data	View logs

Note

You can delete the replication instances that are no longer needed in the specified region. If you have configured replica instances for a premium instance and you want to delete the premium instance, you need to delete all the replica instances first before deleting the premium instance.

Viewing image replication logs

After configuring a replica instance, if you push an image to the primary instance, the image data will be automatically replicated to the replica instance.



Replication　Region　Shanghai 3　Instance [redacted]

Create

Region	Instance status	Synch
Hong Kong, China	Running	No sy
Guangzhou	Running	No sy

Synchronization logs

Task ID	Creation time	Time spent	Succe
No data yet			

Total items: 0　10 / page

Accessing the replica instance via the private network

1. To ensure that the container clusters or CVMs in the replication region can access the replica instance through the private network, you need to connect the VPC in the replication region to the instance. Please refer to Configuring Private Network Access Control and choose the VPC in the replication region.

2. After the above configuration is completed, the container clusters or CVMs in the replication region can access the instance through the private network. The image access address and access credentials are the same as those of the premium instance in the original region.

References

You can also use the `CreateReplicationInstance` API to create a replica instance. For more information, see CreateReplicationInstance.

Cross-Tenant Synchronization

Last updated：2022-01-17 18:41:48

Overview

Tencent Container Registry (TCR) supports the synchronization of container images and Helm Charts among instances in different regions. It also supports single-instance image pushing and worldwide automatic image data synchronization and distribution, helping enterprises quickly deploy and update the container service in multiple regions worldwide.

The instance synchronization feature allows you to customize rules to synchronize specified resources in an instance to a specified location of another instance. In particular, you can select the synchronized resource type (container image, Helm Chart, or both), filter the synchronized resource paths, and use a regular expression to filter repositories and tags. You can also select whether to overwrite existing images with the same names to prevent the loss of historical data due to overwriting. Currently, the instance synchronization feature supports cross-tenant synchronization. You can create a synchronization rule by specifying the instance ID, account ID and access credential of the synchronization target.

Prerequisites

Before creating and managing the synchronization configuration of a TCR Enterprise instance, you need to complete the following tasks:

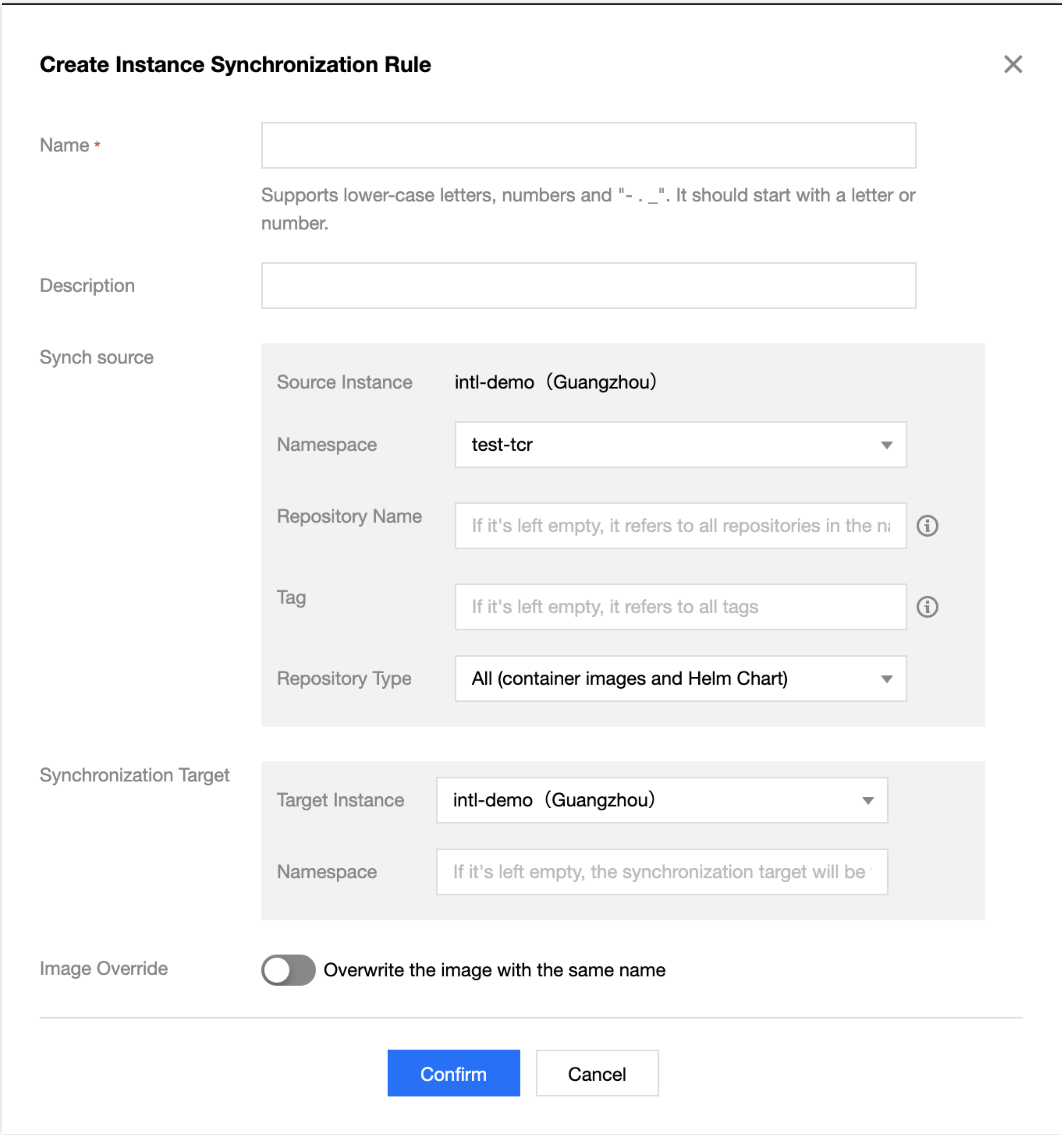
- You have purchased a TCR Enterprise instance with standard or premium specification.
- If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of the Enterprise Edition.

Directions

Creating a synchronization rule

1. Log in to the TCR console and select **Synchronization and Replication** > **Synchronization** in the left sidebar.
2. On the **Synchronization** page, select a region and an instance, and click **Create**.

3. In the **Create Instance Synchronization Rule** window, configure the rule as instructed below.



Create Instance Synchronization Rule

Name *

Supports lower-case letters, numbers and "- . _". It should start with a letter or number.

Description

Synch source

Source Instance: intl-demo（Guangzhou）

Namespace: test-tcr

Repository Name: If it's left empty, it refers to all repositories in the na

Tag: If it's left empty, it refers to all tags

Repository Type: All (container images and Helm Chart)

Synchronization Target

Target Instance: intl-demo（Guangzhou）

Namespace: If it's left empty, the synchronization target will be

Image Override: Overwrite the image with the same name

- **Name**: rule name, which is a string of lowercase letters, numbers, and special characters ([-._]) and must start with a letter or number.
- **Description**: rule description.
- **Sync Source**:

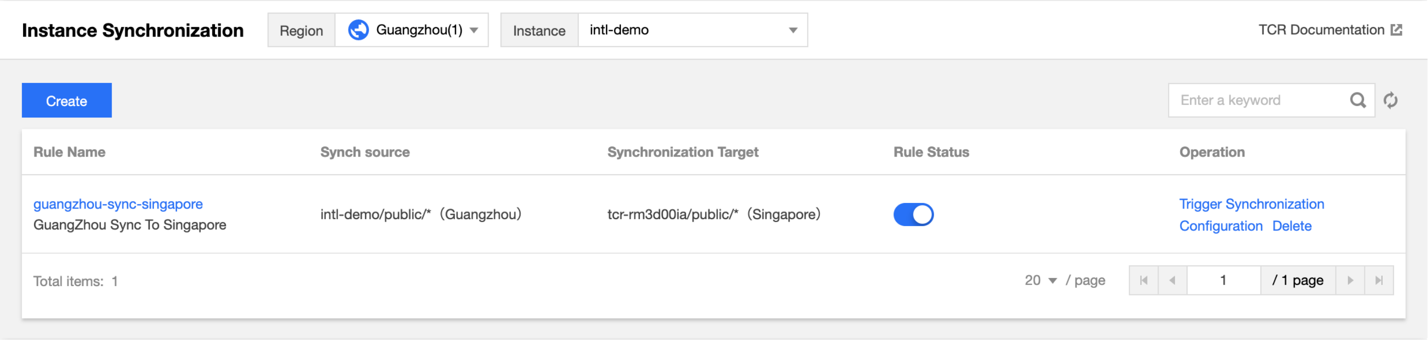
        - **Source Instance**: the current instance is the source instance. You can return to the "Synchronization" page to change the source instance.
        - **Namespace**: namespace with which the current instance needs to synchronize. Currently, you cannot select all namespaces.
        - **Repository Name**: the repository to be synchronized. If you do not enter the repository name, the system will select all repositories in the namespace by default.
        - **Tag**: the tag to be synchronized. If you do not enter the tag, the system will select all tags in the qualified repositories by default.
        - **Repository Type**: type of the resource to be synchronized. You can choose both **Container Image** and **Helm Chart**, or only one of them.
    - **Synchronization Target**: you can specify whether to enable cross-tenant synchronization.
        - Disabled
        - Enabled

      If **Cross-tenant sync** is disabled, the synchronization is between instances under the same tenant. You need to complete the following fields.

        - **Target Instance**: target instance for data synchronization. You can select any instance in the root account.
        - **Namespace**: namespace where the repository is located after it is synchronized to the target instance. If this parameter is not specified, it is set to the namespace with the same name as that in the source instance by default. If such a namespace does not exist, a namespace is created.
    - **Image Override**: this indicates whether the container image with the same name in the target instance is overwritten. We recommend that you do not overwrite images with the same name.
4. Click **OK**.

Managing synchronization rules

After a synchronization rule is created, you can check it on the **Synchronization** page. Then, you can perform the operations shown in the figure below to manage synchronization rules.



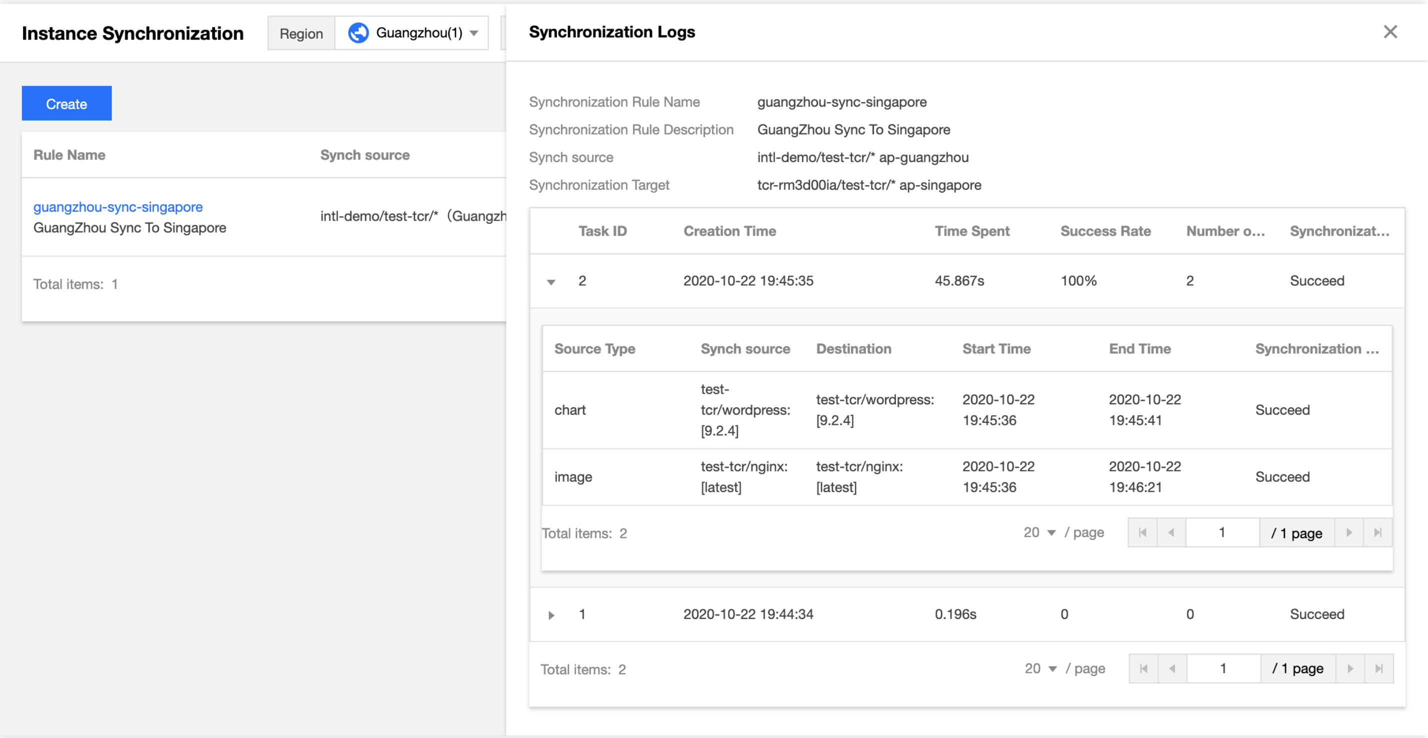
Rule Name	Synch source	Synchronization Target	Rule Status	Operation
guangzhou-sync-singapore GuangZhou Sync To Singapore	intl-demo/public/*（Guangzhou）	tcr-rm3d00ia/public/*（Singapore）	●	Trigger Synchronization Configuration Delete

- **View Synchronization Logs**: click an instance rule to view triggering logs of the rule. For more information, see Viewing synchronization logs.

- **Modify Rule Status**: indicates that a rule is enabled, and indicates that a rule is disabled. A newly created synchronization rule is enabled by default, and you can change the status as needed.
- **Trigger Synchronization**: manually trigger synchronization. All repositories in the instance that match the rule are scanned and synchronized.
- **Configuration**: re-configure an instance synchronization rule. You can configure all parameters.
- **Deletion**: delete the instance synchronization rule.

Viewing synchronization logs

Click the name of an instance synchronization rule to go to the "Synchronization Logs" page of this rule, as shown in the figure below.



Rule Name	Synch source
guangzhou-sync-singapore GuangZhou Sync To Singapore	intl-demo/test-tcr/*（Guangzh

Total items: 1

Synchronization Rule Name guangzhou-sync-singapore
Synchronization Rule Description GuangZhou Sync To Singapore
Synch source intl-demo/test-tcr/* ap-guangzhou
Synchronization Target tcr-rm3d00ia/test-tcr/* ap-singapore

Task ID	Creation Time	Time Spent	Success Rate	Number o...	Synchronizat...
2	2020-10-22 19:45:35	45.867s	100%	2	Succeed

Source Type	Synch source	Destination	Start Time	End Time	Synchronization ...
chart	test-tcr/wordpress:[9.2.4]	test-tcr/wordpress:[9.2.4]	2020-10-22 19:45:36	2020-10-22 19:45:41	Succeed
image	test-tcr/nginx:[latest]	test-tcr/nginx:[latest]	2020-10-22 19:45:36	2020-10-22 19:46:21	Succeed

Total items: 2 20 / page 1 / 1 page

Task ID	Creation Time	Time Spent	Success Rate	Number o...	Synchronizat...
1	2020-10-22 19:44:34	0.196s	0	0	Succeed

Total items: 2 20 / page 1 / 1 page

- **Task ID**: synchronization task ID, which is unique in the instance.
- **Creation Time**: time when the synchronization task was created.
- **Time Spent**: time consumed to complete all the synchronization tasks.
- **Success Rate**: resource synchronization completion ratio. Multiple repositories may be synchronized concurrently in the same synchronization task.
- **Number of Synced Repositories**: number of repositories that need to be synchronized in the current task.
- **Synchronization Status**: task status. If a large number of container images and Helm Charts need to be synchronized in a task, the task may remain in the InProgress status for a long time.

References

You can also use the `ManageReplication` API to manage instance sync.

Loading Container Images on Demand

Last updated：2023-05-08 15:44:59

Overview

When using container images to deploy and update business applications, the traditional scheme is to download and decompress the full amount of container image data. On the one hand, it takes a long time to start the container. On the other hand, it may also cause large network and disk read and write pressure due to the large scale of the cluster and the download and decompression process, which leads to large-scale container startup, failing to meet deployment expectations. In fact, container startup may use only part of the data inside the container image.

TCR Enterprise supports on-demand loading of container images. During business deployment, you can use converted accelerated image tags to download image data without full loading and decompress the data online, greatly improving application distribution efficiency for ultimate elastic experience. This document introduces how to load container images on demand.

Prerequisites

You have created a container cluster. Currently, the on-demand loading feature is available only for Tencent Cloud TKE clusters that meet the following requirements:

The cluster Kubernetes version is 1.16 or later.

The cluster runtime add-on is Containerd v1.4.3. You can modify the runtime configuration of an existing cluster to Containerd v1.4.3, so that the nodes added after the modification will default to this version.

The cluster OS is Ubuntu, TencentOS, or CentOS. If the cluster OS is CentOS, you need to run the `yum install -y fuse` command on the cluster nodes to install FUSE.

You have purchased a TCR Enterprise Edition instance with premium specification. The feature of loading container images on demand can be enabled only for a **premium instance**.

The VPC of the container cluster has been connected to the TCR Enterprise Edition instance, and the cluster nodes can access the images in the instance via the private network. For information about the configuration details, see Configuring Private Network Access Control.

Preparing accelerated images

Enabling image acceleration

1. Log in to the TCR console and click **Image Acceleration** in the left sidebar.

2. On the **Image Acceleration** page, select the region and name of the instance for which image acceleration is to be enabled, and you can view the status of the current instance image acceleration and the list of image acceleration rules.

3. Click **Enable Image Acceleration**. In the **Activate Image Acceleration** window that pops up, read the notes carefully.

Once image acceleration is enabled, a new OCI-compatible accelerated image is generated after you upload a container image that complies with the acceleration rules.

Note that after this feature is enabled and used, storing both general and accelerated images will incur additional image storage costs.

4. Click **Conform**.

Adding an image acceleration rule

1. Click **Add Image Acceleration Rule**. In the **Create Image Acceleration Rule** window that pops up, configure a rule as prompted.

Name: Rule name.

Description: Rule description.

Triggering Rule:

Triggered Instance: The currently selected instance is the triggered instance.

Namespace: Namespace whose distribution needs to be accelerated within the current instance. Currently, you cannot select all namespaces.

Repository Name: Accelerated repository. You can use a regular expression to filter repositories. If this parameter is not specified, all repositories in the namespace are selected by default.

Tag: Accelerated tag. You can use a regular expression to filter tags. If this parameter is not specified, all tags in the repositories that meet the requirements are selected by default.

Validate Rule: Enter the address of the image to be accelerated to check whether the image under the current triggering rule meets the acceleration rule.

2. Click **OK**.

Pushing the image and automatically converting it

Check the added image acceleration rule on the **Image Acceleration** page. If the rule is enabled, push the new container image to the image repository that meets the rule. This will automatically trigger the image format conversion, and an accelerated image with the -apparate suffix will be generated. The default image artifact type is Docker-Image. After the conversion, the image artifact type is OCI-Image-v1.

Deploying an Acceleration Image

Tencent Kubernetes Engine (TKE) is a Kubernetes managed service that works closely with TCR. You can install the TCR acceleration application in a TKE cluster and deploy an acceleration image to increase the business startup speed.

Configuring cluster nodes

Cluster nodes do not support acceleration images by default. To enable a cluster node to use an acceleration image with priority, add the image acceleration label to the cluster node via the CLI or TKE console.
Adding the Image Acceleration Label via the CLI
Adding the Image Acceleration Label via the TKE Console
Run the following command to add the image acceleration label to a cluster node:

```
kubectl label node xxx cloud.tencent.com/apparate=true
```

1. Log in to the TKE console and select **Cluster** in the left sidebar.

2. On the **Cluster Management** page, click the ID of the cluster that requires image acceleration for distribution to go to the cluster details page.

3. Select the ID/name of the cluster for which to set the node label to go to the cluster details page.

4. In the left sidebar, choose **Node Management** > **Nodes** to go to the **Node List** page.

5. Choose **More** > **Edit Label** on the right of the target node.

6. In the **Edit Label** window that pops up, edit the label as `cloud.tencent.com/apparate=true` and click **Submit**.

Installing the acceleration application

A cluster does not support the use of acceleration images by default, so you need to install the TCR acceleration suite application on the cluster. After the TCR acceleration suite application is installed, the nodes that have been labeled as supporting the deployment of acceleration images will automatically deploy the DaemonSet process and can load the acceleration images normally.
After the TCR acceleration suite application is installed, if you add a node and label it `cloud.tencent.com/apparate=true` , the node will also automatically deploy the DaemonSet process and can load acceleration images normally.

Installing the TCR acceleration suite application via the CLI

1. Install the Helm V3 CLI. For more information, see Using the Helm client to upload and download Helm Charts.
2. Add the Helm repository and pull the TCR acceleration application Chart package.

```
helm repo add tcr-helm-public https://helmhub.tencentcloudcr.com/chartrepo/public
helm pull tcr-helm-public/apparate --version 1.0.0
```

3. Decompress the downloaded Chart package and modify `values.yaml` .

```
tar -xzvf apparate-1.0.0.tgz
vim apparate/values.yaml
```

Configure the following parameters:

3.1 `imagePullSecretsCrs` : This configuration is used for pulling acceleration images. You need to set `dockerUsername` , `dockerPassword` , and `dockerServer` to specify the TCR Enterprise Edition instance username, password, and access domain respectively.

3.2 `image` : Retain the default value. This configuration is used for pulling basic images during application installation on a cluster. If the cluster is not deployed in the Chinese mainland, change the value to the access domain name of the TCR Individual image repository in the corresponding region.

4. Build the Chart package again and install it to the specified cluster.

```
helm package apparate/
helm install apparate apparate-1.0.0.tgz
```

Before running the `helm install` command, you need to configure the cluster access credentials locally in advance. For more information, see Connecting to a Cluster Using the Local Helm Client.

5. Go to the cluster application page and confirm the application's installation status and configuration.

Deploying an acceleration image

When creating a workload, select an image within the current instance. Only when the following conditions are met, the cluster loads the image on demand to quickly start the container:
The container image specified for the workload is a converted acceleration image, such as `nginx:latest-apparate`, and its artifact type is OCI-Image-v1.
The image acceleration label `cloud.tencent.com/apparate=true` is added to the node to which the workload Pod is scheduled.
Therefore, when creating a workload, select an accelerated image tag, add nodeSelector, and set `cloud.tencent.com/apparate=true` so that the workload will be scheduled to a node that supports accelerated images to implement accelerated startup.

FAQs

Can I delete regular and accelerated images?

Yes. When both regular and accelerated images exist in the repository, deleting one will not affect the pull and deployment of the other.

What should I do if no accelerated image is generated automatically after image push?

Check if the image matches the existing acceleration rules. If you are sure that the image meets the acceleration rules in the enabled state, you can submit a ticket for help.

Image Security

Container Image Security Scanning

Last updated：2023-05-08 15:44:59

Overview

The Tencent Container Registry (TCR) Enterprise Edition supports security scanning of managed container images and can generate scanning reports, expose potential security vulnerabilities in container images, and provide suggestions for fixing them. Container image security is an important part of cloud native application delivery security. Timely security scanning of uploaded container images and blocking application deployment based on the scanning results can effectively reduce the risk of vulnerabilities in the production environment.

The image security scanning feature is a built-in feature of image repositories. You can actively trigger the security scanning of the container image of the specified version after uploading the container image. Also, you can configure automatic scanning at the namespace level, so that newly pushed images in the namespace will be scanned automatically after upload. The current image security scanning service is based on the open-source Clair solution, and the relevant vulnerability information is from the official CVE vulnerability library and synchronized on a regular basis.

Prerequisites

Make sure that the following conditions are met before using the image security scanning feature,:

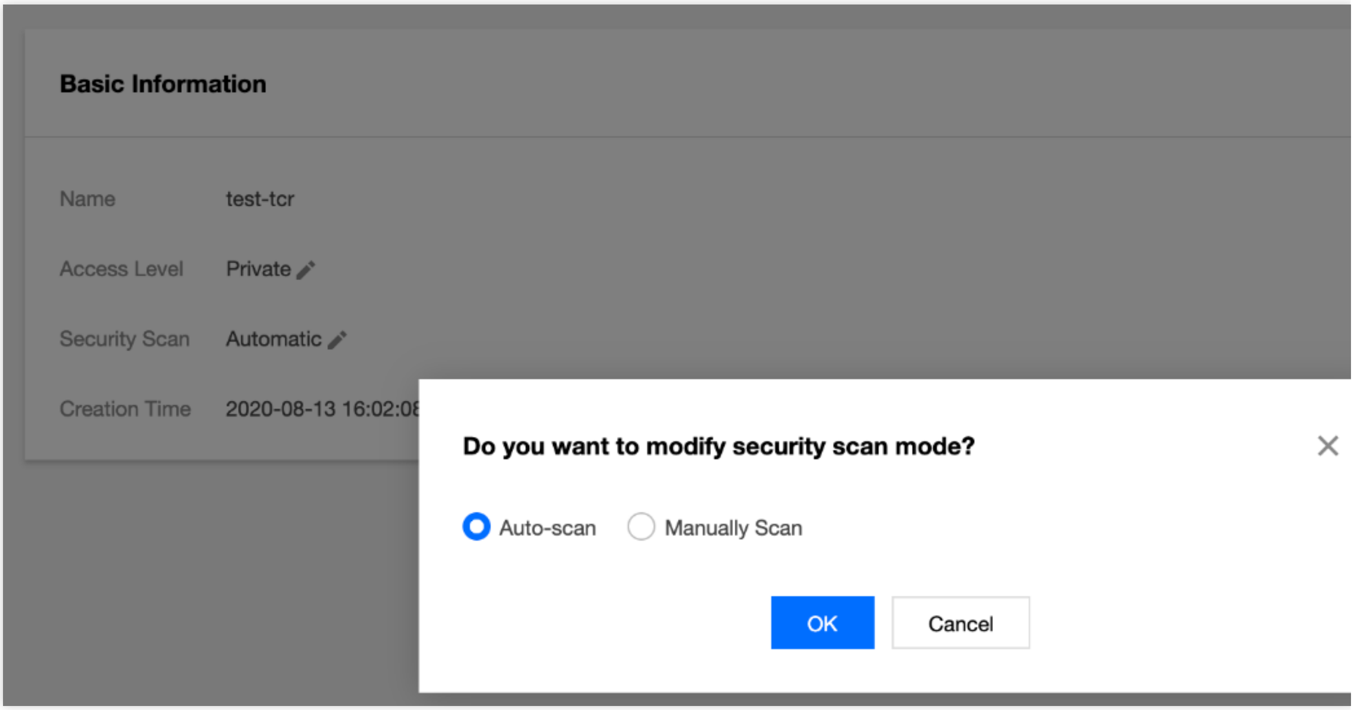
You have purchased a TCR Enterprise Edition instance.

If you are using a sub-account, the sub-account must have obtained operation permissions on the corresponding instance. For more information, see TCR Enterprise Authorization Management.

Directions

Configuring the scanning policy

1. Log in to the TCR console and click **Namespace** in the left sidebar.
2. On the **Namespace** page, click the name of the instance for which you want to enable the image security scanning feature to go to the namespace details page.
3. On the details page, select **Auto-scan** for security scanning, as shown in the figure below:



Basic Information

Name	test-tcr
Access Level	Private
Security Scan	Automatic
Creation Time	2020-08-13 16:02:08

Do you want to modify security scan mode?

● Auto-scan ○ Manually Scan

Manually triggering scanning

Step 1: Prepare a container image

Refer to Basic Image Repository Operations to upload a container image and view the image on the version management page of the corresponding image repository.

Step 2: Trigger image scanning

Select a specific image version in the image repository and click **Scan** to trigger image scanning. At this time, the security level is displayed as **Scanning**, as shown in the figure below:

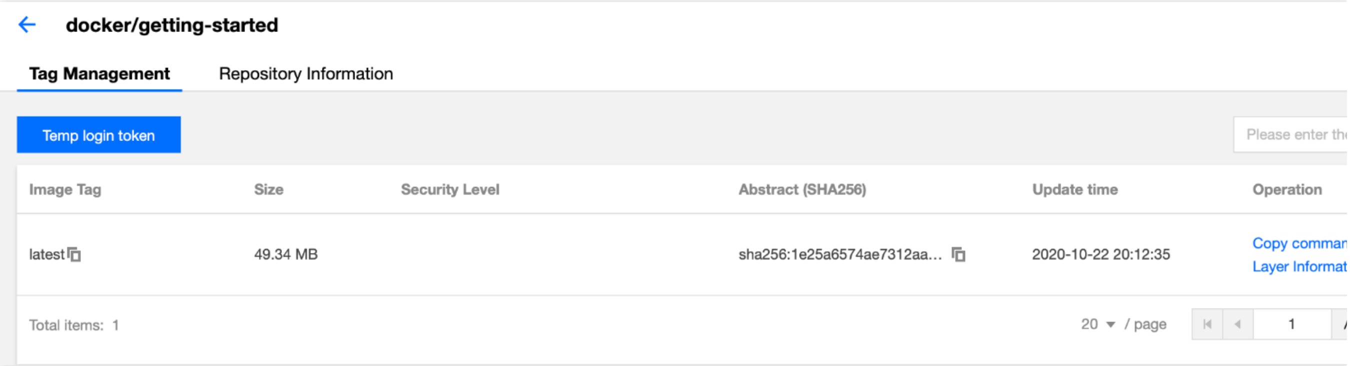
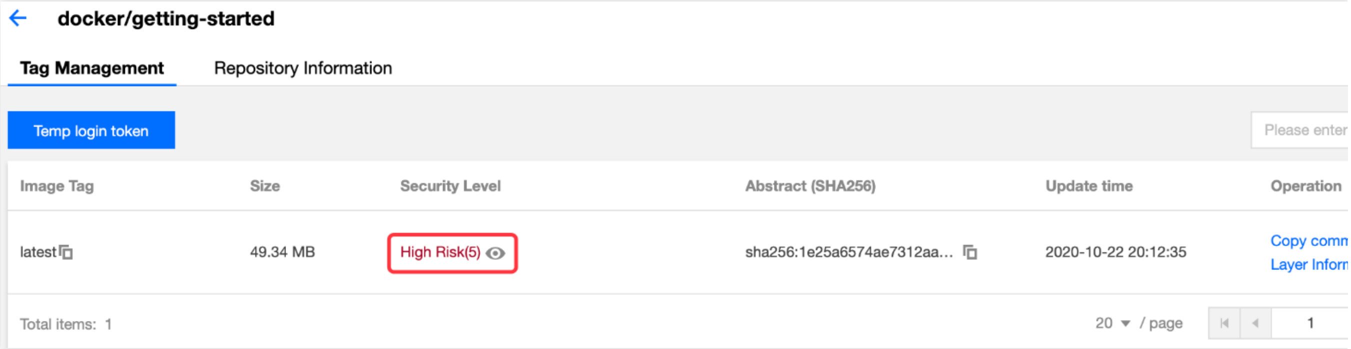


Image Tag	Size	Security Level	Abstract (SHA256)	Update time	Operation
latest	49.34 MB		sha256:1e25a6574ae7312aa...	2020-10-22 20:12:35	Copy comman Layer Informat

Total items: 1

Step 3: View the scanning results

After the security scanning is complete, the highest level and the number of vulnerabilities in the current image are displayed in the security level section. You can view the vulnerability details, as shown in the figure below:



docker/getting-started

Tag Management | Repository Information

Temp login token

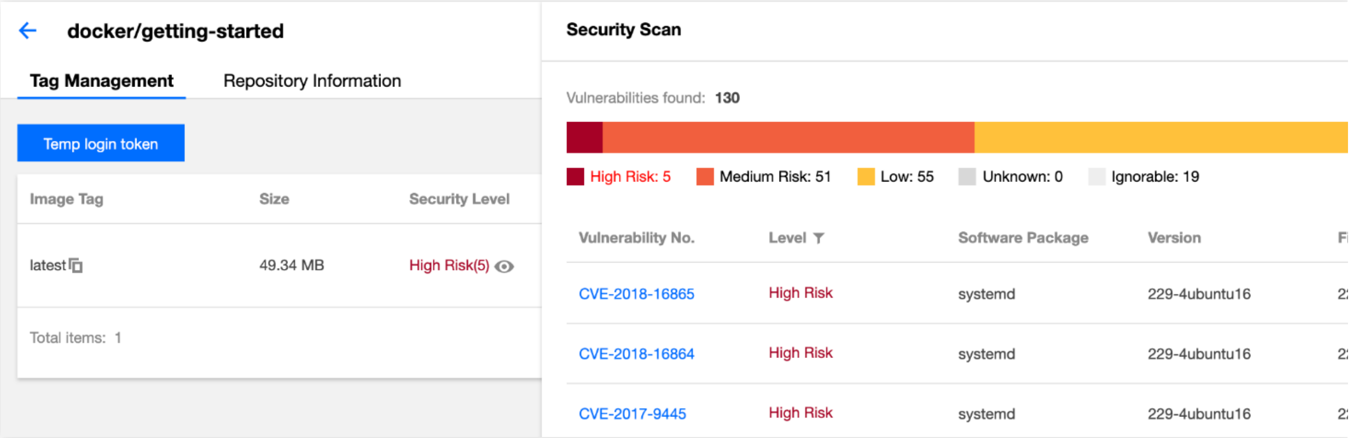
Please enter

Image Tag	Size	Security Level	Abstract (SHA256)	Update time	Operation
latest	49.34 MB	High Risk(5)	sha256:1e25a6574ae7312aa...	2020-10-22 20:12:35	Copy comm Layer Inform

Total items: 1

20 / page 1

When viewing the details of vulnerabilities, you can click a specific vulnerability ID to redirect to the details of the vulnerability so that you can assess its actual impact on business, as shown in the figure below:



docker/getting-started

Tag Management | Repository Information

Temp login token

Image Tag	Size	Security Level
latest	49.34 MB	High Risk(5)

Total items: 1

Security Scan

Vulnerabilities found: 130

High Risk: 5 | Medium Risk: 51 | Low: 55 | Unknown: 0 | Ignorable: 19

Vulnerability No.	Level	Software Package	Version	Fi
CVE-2018-16865	High Risk	systemd	229-4ubuntu16	2
CVE-2018-16864	High Risk	systemd	229-4ubuntu16	2
CVE-2017-9445	High Risk	systemd	229-4ubuntu16	2

Step 4: Re-trigger scanning

As the vulnerability library is updated regularly, you can refer to Step 2: Trigger image scanning to re-trigger the security scanning of the specified image and obtain the latest scanning results.

Configuring Image Tag Immutability

Last updated：2022-04-01 16:53:44

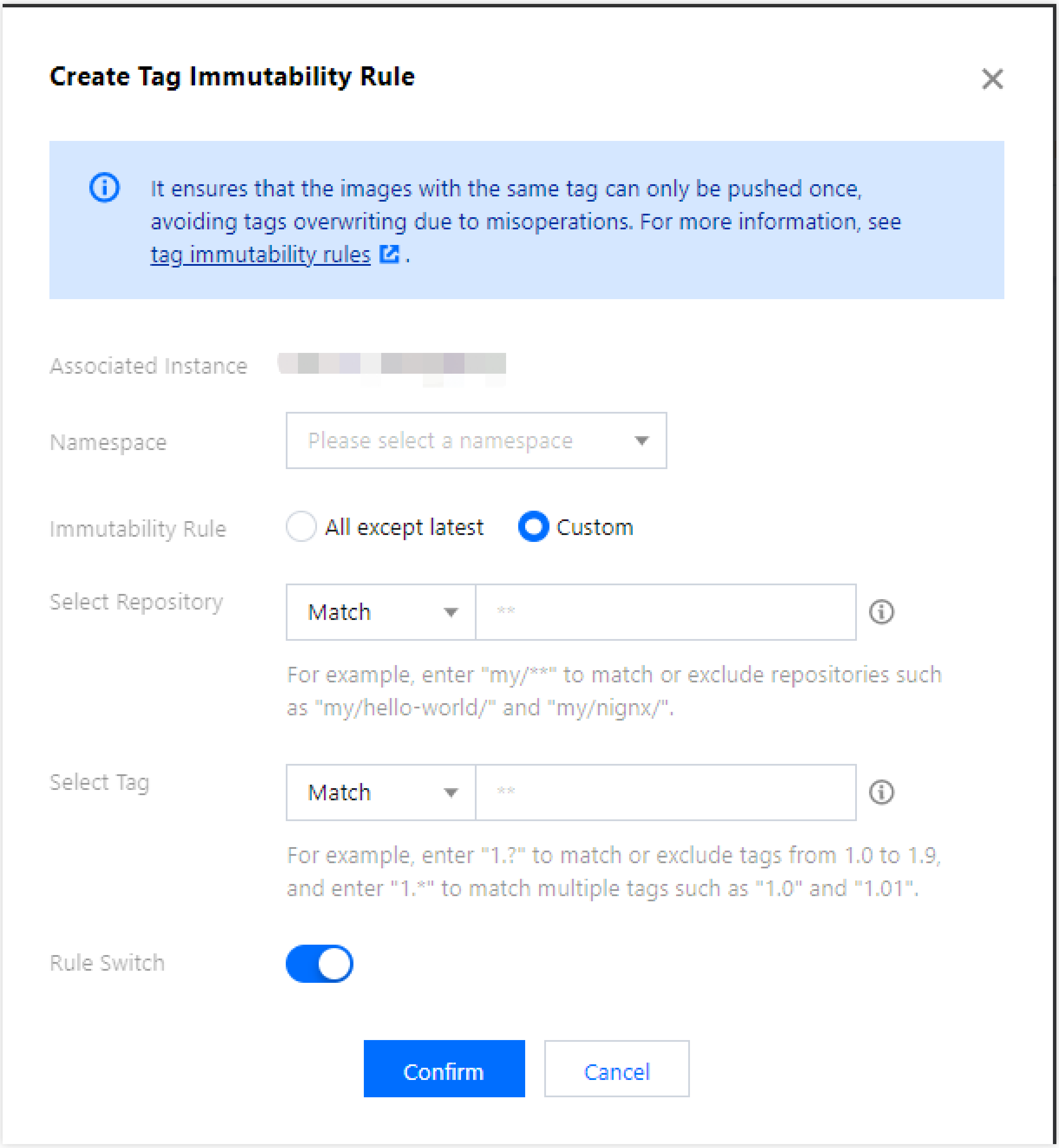
Overview

Tencent Container Registry (TCR) Enterprise Edition supports protection for the hosted container image tags. Container image security is a key part of cloud-native application delivery. It enables tag immutability feature for the images hosted in TCR, which ensures the images of the same tag will only be successfully pushed once, thus effectively reduce the risk of tag overwriting caused by misoperation in the production environment. TCR supports tag protection at the namespace level. Users can fine-grainly define the repositories and image tags covered by the feature according to service demands.

Directions

Creating tag immutability rule

1. Log in to the TCR console and select **Tag Management** > **Tag Immutability** on the left sidebar.
2. Select the region where the instance is located and the instance name on the “Tag Immutability” page.
3. Click **Create Rule**. In the **Create Tag Immutability Rule** window, configure the rule based on the following information. See the figure below:



Create Tag Immutability Rule

It ensures that the images with the same tag can only be pushed once, avoiding tags overwriting due to misoperations. For more information, see tag immutability rules .

Associated Instance

Namespace: Please select a namespace

Immutability Rule: ○ All except latest ● Custom

Select Repository: Match **

For example, enter "my/**" to match or exclude repositories such as "my/hello-world/" and "my/nignx/".

Select Tag: Match **

For example, enter "1.?" to match or exclude tags from 1.0 to 1.9, and enter "1.*" to match multiple tags such as "1.0" and "1.01".

Rule Switch

Configuration Item	Description
Associated instance	The instance which has been selected currently.
Namespaces	The current instance needs to enable the namespace for tag protection. Only a rule can be created in a single namespace.
Immutability rule	**latest**: in all repositories in the current namespace, all image tags are not allowed to be overwritten except the latest tag.
	Custom: customize the configuration of the repository and image tag that need to be

	matched. ○ **Repository matching**: select filter type for the image repository, and enter the name of the repository which needs to be filtered. ○ **Tag matching**: select filter type for the image tag, and enter the name of the tag which needs to be filtered.
Rule switch	The rule is effective as of creation by default. Note Enabling means the rule takes effect. You can enable/disable the rule in the configuration.

iv. Click **Confirm** to create the rule.

Managing tag immutability rule

You can view the rules on the “Tag Immutability” page after creation, and take the following actions to manage the rules.

- **Configuration**: you can reconfigure the instance tag immutability rule but cannot modify the namespace for which it takes effect.
- **Delete**: delete the tag immutability rule under the instance.

Blocking the Deployment of High-Risk Images

Last updated：2022-05-09 12:41:24

Overview

The Tencent Container Registry (TCR) Enterprise supports security scanning of managed container images and can generate scanning reports, expose potential security vulnerabilities in container images, and provide suggestions for fixing them. Container image security is an important part of cloud native application delivery security. Timely security scanning of uploaded container images and blocking application deployment based on the scanning results can effectively reduce the risk of vulnerabilities in the production environment.

The image deployment blocking feature is configured at the namespace level. You can enable it, and configure the blocking policy and the ignorable image vulnerabilities. Then, if a container client is trying to pull images that meet the blocking policy, the pulling will be denied and error reports will be returned.

Prerequisites

Before using the image deployment blocking feature, you need to perform the following operations:

- Create a TCR Enterprise Instance.
- If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of TCR Enterprise.

Directions

Configuring the blocking policy

1. Log in to the TCR console and select **Namespace** in the left sidebar.
2. On the "Namespace" page, click the name of the instance for which you want to configure the blocking policy to go to the namespace details page.
3. On the "Deployment security" page, enable the blocking and configure the vulnerability level to be blocked.

Configure the allowlist of vulnerabilities

After enabling the blocking, you can configure the allowlist of vulnerabilities. Enter one or more CVE IDs and separate them with commas. Then, if the image ID is contained in the security scanning result, it will be ignored by the blocking policy. That means, if there is a high-risk vulnerability in the image, and the vulnerability is configured in the allowlist, the image can be pulled normally even if it is configured to block the pulling of images with high-risk vulnerabilities.

Container Image Signature

Last updated：2024-07-01 18:10:09

Image signing and signature verification can avoid man-in-the-middle attacks and the update and running of invalid images, ensuring image consistency across the entire linkage ranging from distribution to deployment. TCR Enterprise supports namespace-level automatic image signing. When an image is pushed to the repository, it will be automatically signed according to the matched signing policy to ensure image content trustworthiness in your repository.

Prerequisites

Before using the image signing feature, you need to perform the following operations:

Create a TCR Enterprise Instance.

If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of TCR Enterprise.

Key Management Service (KMS) has been activated.

Directions

Creating an asymmetric signature verification key

1. Log in to the KMS console.
2. Choose **Key Management** > **Customer Managed CMK** and click **Create**.
3. In the **Create Key** window that pops up, set key parameters and click **OK**. The container signature feature requires that the KMS key usage be set to **Asymmetric Signature Verification** and the encryption algorithm be set to **RSA_2048**. For the settings of the other parameters, see Creating a Key.

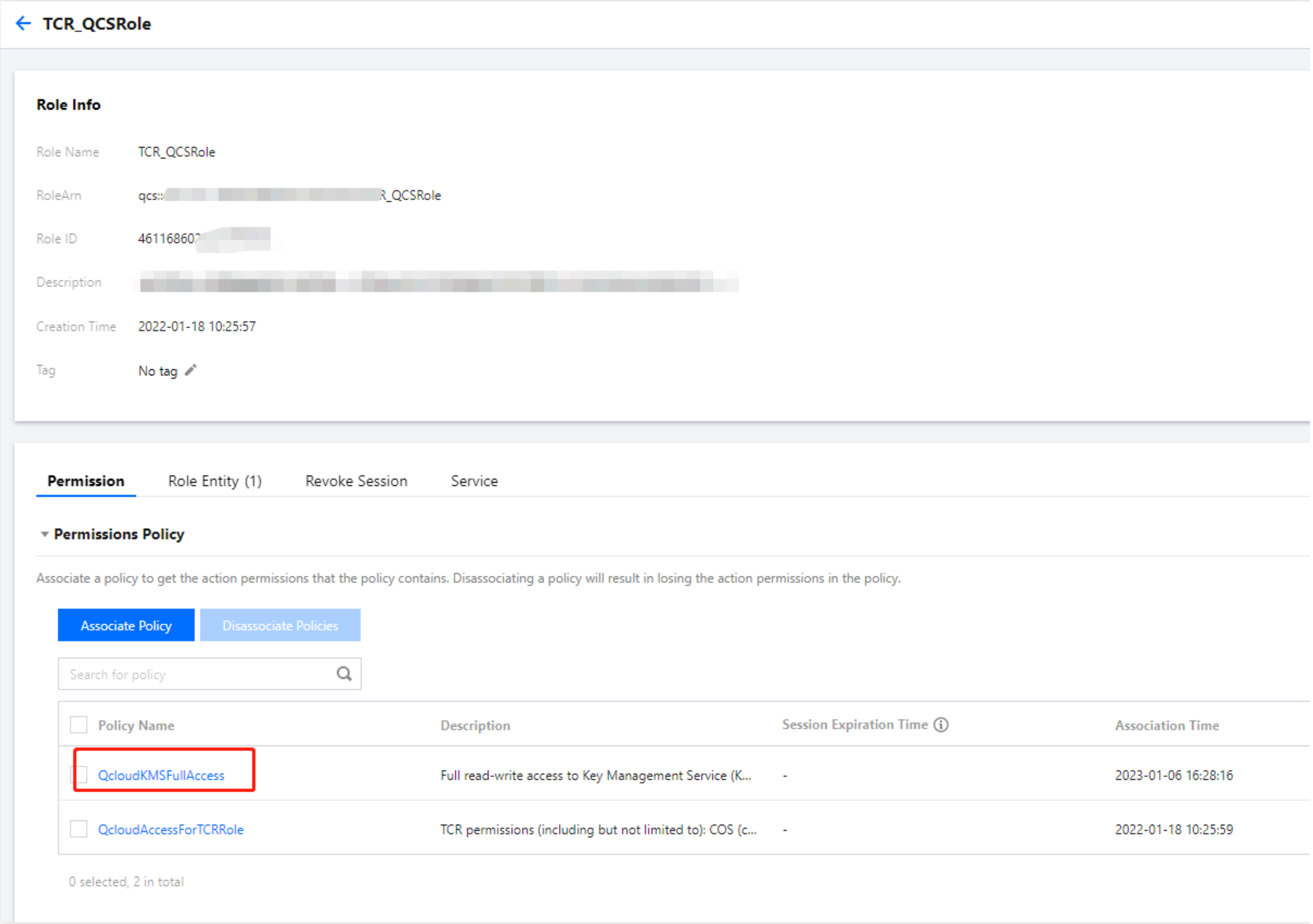
Note:

TCR supports obtaining user keys in all regions of the KMS service. To reduce the cross-region communication overhead, it is recommended that the KMS user key and the image repository instance be located in the same region.

Authorizing TCR to use the KMS key

To enable TCR to read the asymmetric signature verification key under your account, you need to configure a policy as follows under your account:

1. Log in to the CAM console.
2. On the **Role** page, click **TCR_QCSRole**.
3. On the TCR_QCSRole details page, associate the preset policy **QcloudKMSFullAccess**.



TCR_QCSRole

Role Info

Role Name	TCR_QCSRole
RoleArn	qcs::[redacted]R_QCSRole
Role ID	46116860[redacted]
Description	[redacted]
Creation Time	2022-01-18 10:25:57
Tag	No tag

Permissions Policy

Associate a policy to get the action permissions that the policy contains. Disassociating a policy will result in losing the action permissions in the policy.

Policy Name	Description	Session Expiration Time	Association Time
QcloudKMSFullAccess	Full read-write access to Key Management Service (K...	-	2023-01-06 16:28:16
QcloudAccessForTCRRole	TCR permissions (including but not limited to): COS (c...	-	2022-01-18 10:25:59

0 selected, 2 in total

Creating an image signing policy

1. Log in to the TCR console.
2. On the instance management page, select a target image repository instance.
3. Select **Image Security** in the left sidebar to go to the image signing details page.
4. Click **Create**. In the signing policy creation window that pops up, set parameters as instructed.

Policy Name: Image signing policy name. The value must be 2 to 50 characters in length and can contain only lowercase letters, numbers, and separators, including periods (.), underscores (_), hyphens (-), and slashes (/). It can neither start or end with a separator nor contain consecutive separators.

Namespace: Namespace where the image signing policy takes effect. Only one signing policy is supported per namespace.

KMS Key: KMS customer managed CMK that supports signing. Only a key that is used for RSA2048 asymmetric key verification can be loaded.

Domain Name: Domain name used to access the repository instance service.

5. Click **OK**.

Note:

Once created, the signing policy takes effect for new images immediately. That is, when an image is pushed to the repository, it will be automatically signed according to the matched signing policy.
An enabled signing policy does not take effect for images that already exist in the repository. For existing images, you need to manually trigger signing on the **Image Repository** > **Tag Management** page in the console.

Viewing image signing status

You can check whether the signing policy is enabled on the **Namespace** page.
You can check whether the signing policy is enabled for images on the **Image Repository** > **Tag Management** page. For images that have been pushed to the image repository before the signing policy is enabled, you can manually trigger signing in the **Operation** column.

Deleting an image signing policy

On the **Image Signature** page, select the signing policy to delete and click **Delete**. In the window that pops up, click **OK**.
Note:
Deleting the signing policy will also delete the image signing information in the existing namespace, which may cause signature verification failure.

Synchronization and Replication

Configuring Instance Synchronization

Last updated：2021-11-23 15:36:14

Overview

Tencent Container Registry (TCR) supports the synchronization of container images and Helm Charts among instances in different regions. It also supports single-instance image pushing and worldwide automatic image data synchronization and distribution, helping enterprises quickly deploy and update the container service in multiple regions worldwide.

The instance synchronization feature allows you to customize rules to synchronize specified resources in an instance to a specified location of another instance. In particular, you can select the synchronized resource type (container image, Helm Chart, or both), filter the synchronized resource paths, and use a regular expression to filter repositories and tags. You can also select whether to overwrite existing images with the same names to prevent the loss of historical data due to overwriting. Currently, the instance synchronization feature supports cross-tenant synchronization. You can create a synchronization rule by specifying the instance ID, account ID and access credential of the synchronization target.

Prerequisites

Before creating and managing the synchronization configuration of a TCR Enterprise instance, you need to complete the following tasks:

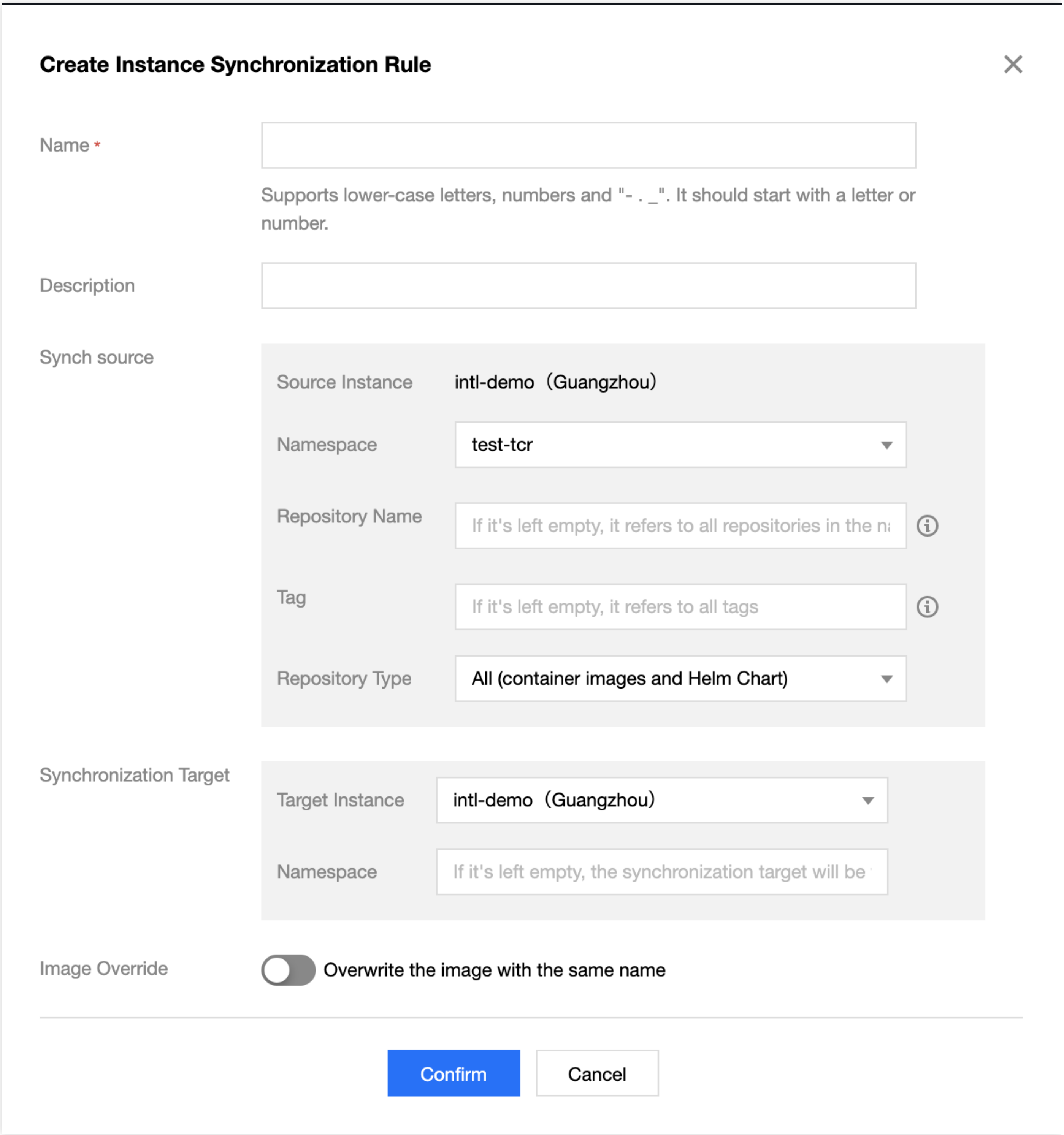
- You have purchased an TCR Enterprise instance with standard or premium specification.
- If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of the Enterprise Edition.

Directions

Creating a synchronization rule

1. Log in to the TCR console and select **Synchronization and Replication** > **Synchronization** in the left sidebar.
2. On the **Synchronization** page, select a region and an instance, and click **Create**.

3. In the **Create Instance Synchronization Rule** window, configure the rule as instructed below.



Create Instance Synchronization Rule

Name *:

Supports lower-case letters, numbers and "- . _". It should start with a letter or number.

Description:

Synch source

Source Instance: intl-demo（Guangzhou）

Namespace: test-tcr

Repository Name: If it's left empty, it refers to all repositories in the n

Tag: If it's left empty, it refers to all tags

Repository Type: All (container images and Helm Chart)

Synchronization Target

Target Instance: intl-demo（Guangzhou）

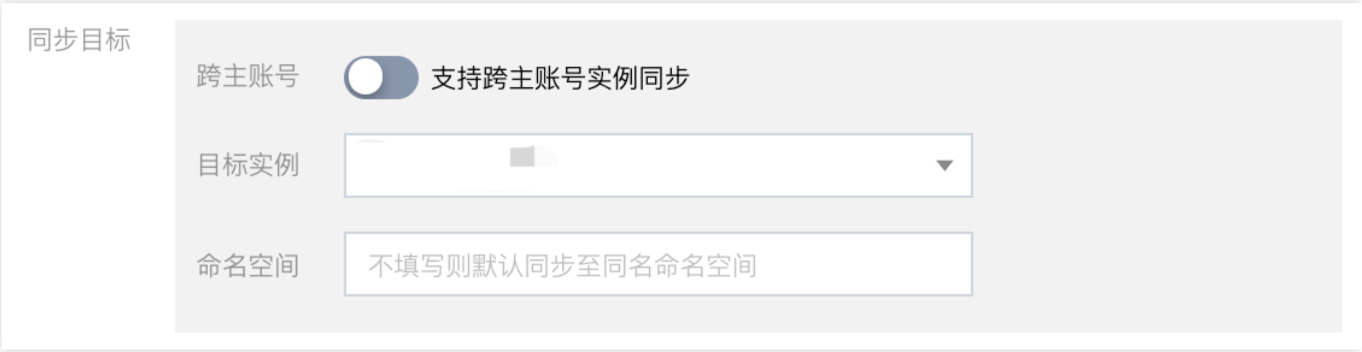
Namespace: If it's left empty, the synchronization target will be

Image Override: Overwrite the image with the same name

- **Name**: rule name, which is a string of lowercase letters, numbers, and special characters ([-._]) and must start with a letter or number.
- **Description**: rule description.
- **Sync Source**:

    - **Source Instance**: the current instance is the source instance. You can return to the "Synchronization" page to change the source instance.
    - **Namespace**: namespace with which the current instance needs to synchronize. Currently, you cannot select all namespaces.
    - **Repository Name**: the repository to be synchronized. You can use regular expression to filter. If you do not enter the repository name, the system will select all repositories in the namespace by default.
    - **Tag**: the tag to be synchronized. You can use regular expression to filter. If you do not enter the tag, the system will select all tags in the qualified repositories by default.
    - **Repository Type**: type of the resource to be synchronized. You can choose both **Container Image** and **Helm Chart**, or only one of them.
  - **Synchronization Target**: you can specify whether to enable cross-tenant synchronization.
    - Disabled
    - Enabled

  If **Cross-tenant sync** is disabled, the synchronization is between instances under the same tenant. You need to complete the following fields.



同步目标

跨主账号 支持跨主账号实例同步

目标实例

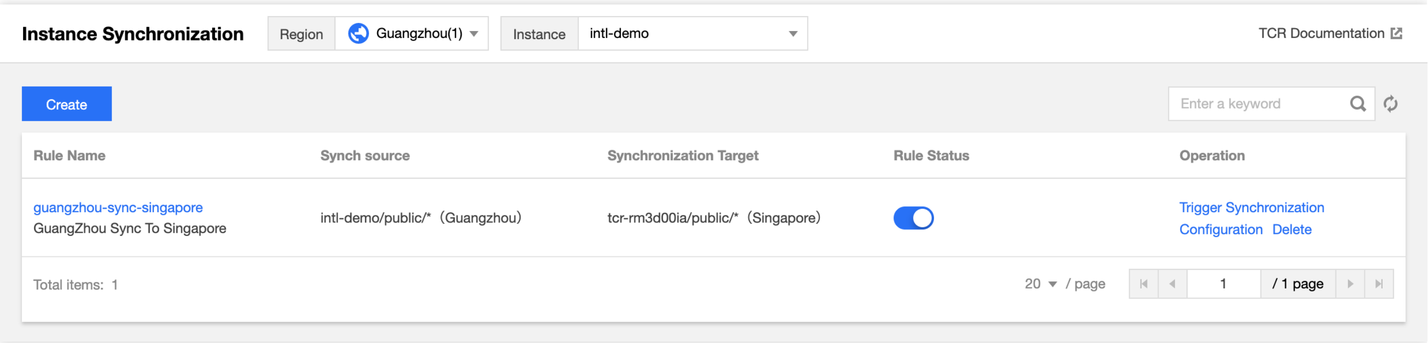
命名空间 不填写则默认同步至同名命名空间

    - **Target Instance**: target instance for data synchronization. You can select any instance in the master account.
    - **Namespace**: namespace where the repository is located after it is synchronized to the target instance. If this parameter is not specified, it is set to the namespace with the same name as that in the source instance by default. If such a namespace does not exist, a namespace is created.
  - **Image Override**: this indicates whether the container image with the same name in the target instance is overwritten. We recommend that you do not overwrite images with the same name.

4. Click **OK**.

Managing synchronization rules

After a synchronization rule is created, you can check it on the **Synchronization** page. Then, you can perform the operations shown in the figure below to manage synchronization rules.



Instance Synchronization Region Guangzhou(1) Instance intl-demo TCR Documentation

Create Enter a keyword

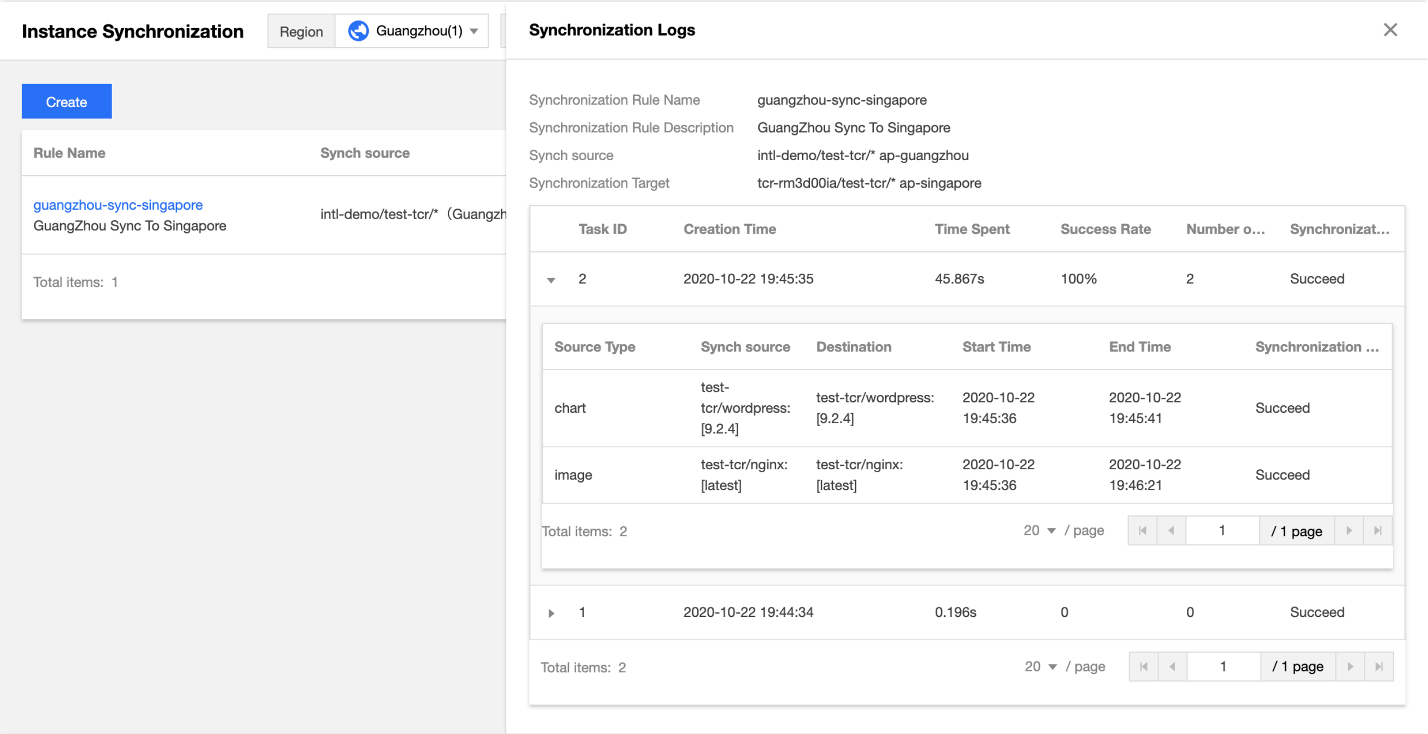
Rule Name	Synch source	Synchronization Target	Rule Status	Operation
guangzhou-sync-singapore GuangZhou Sync To Singapore	intl-demo/public/*（Guangzhou）	tcr-rm3d00ia/public/*（Singapore）	●	Trigger Synchronization Configuration Delete

Total items: 1 20 / page 1 / 1 page

- **View Synchronization Logs**: click an instance rule to view triggering logs of the rule. For more information, see Viewing synchronization logs.
- **Modify Rule Status**: ● indicates that a rule is enabled, and ○ indicates that a rule is disabled. A newly created synchronization rule is enabled by default, and you can change the status as needed.
- **Trigger Synchronization**: manually trigger synchronization. All repositories in the instance that match the rule are scanned and synchronized.
- **Configuration**: re-configure an instance synchronization rule. You can configure all parameters.
- **Deletion**: delete the instance synchronization rule.

Viewing synchronization logs

Click the name of an instance synchronization rule to go to the "Synchronization Logs" page of this rule, as shown in the figure below.



Instance Synchronization Region Guangzhou(1)

Create

Rule Name	Synch source
guangzhou-sync-singapore GuangZhou Sync To Singapore	intl-demo/test-tcr/*（Guangzh

Total items: 1

Synchronization Logs

Synchronization Rule Name guangzhou-sync-singapore
Synchronization Rule Description GuangZhou Sync To Singapore
Synch source intl-demo/test-tcr/* ap-guangzhou
Synchronization Target tcr-rm3d00ia/test-tcr/* ap-singapore

Task ID	Creation Time	Time Spent	Success Rate	Number o...	Synchronizat...
2	2020-10-22 19:45:35	45.867s	100%	2	Succeed

Source Type	Synch source	Destination	Start Time	End Time	Synchronization ...
chart	test-tcr/wordpress:[9.2.4]	test-tcr/wordpress:[9.2.4]	2020-10-22 19:45:36	2020-10-22 19:45:41	Succeed
image	test-tcr/nginx:[latest]	test-tcr/nginx:[latest]	2020-10-22 19:45:36	2020-10-22 19:46:21	Succeed

Total items: 2 20 / page 1 / 1 page

Task ID	Creation Time	Time Spent	Success Rate	Number o...	Synchronizat...
1	2020-10-22 19:44:34	0.196s	0	0	Succeed

Total items: 2 20 / page 1 / 1 page

- **Task ID**: synchronization task ID, which is unique in the instance.
- **Creation Time**: time when the synchronization task was created.
- **Time Spent**: time consumed to complete all the synchronization tasks.
- **Success Rate**: resource synchronization completion ratio. Multiple repositories may be synchronized concurrently in the same synchronization task.
- **Number of Synced Repositories**: number of repositories that need to be synchronized in the current task.
- **Synchronization Status**: task status. If a large number of container images and Helm Charts need to be synchronized in a task, the task may remain in the InProgress status for a long time.

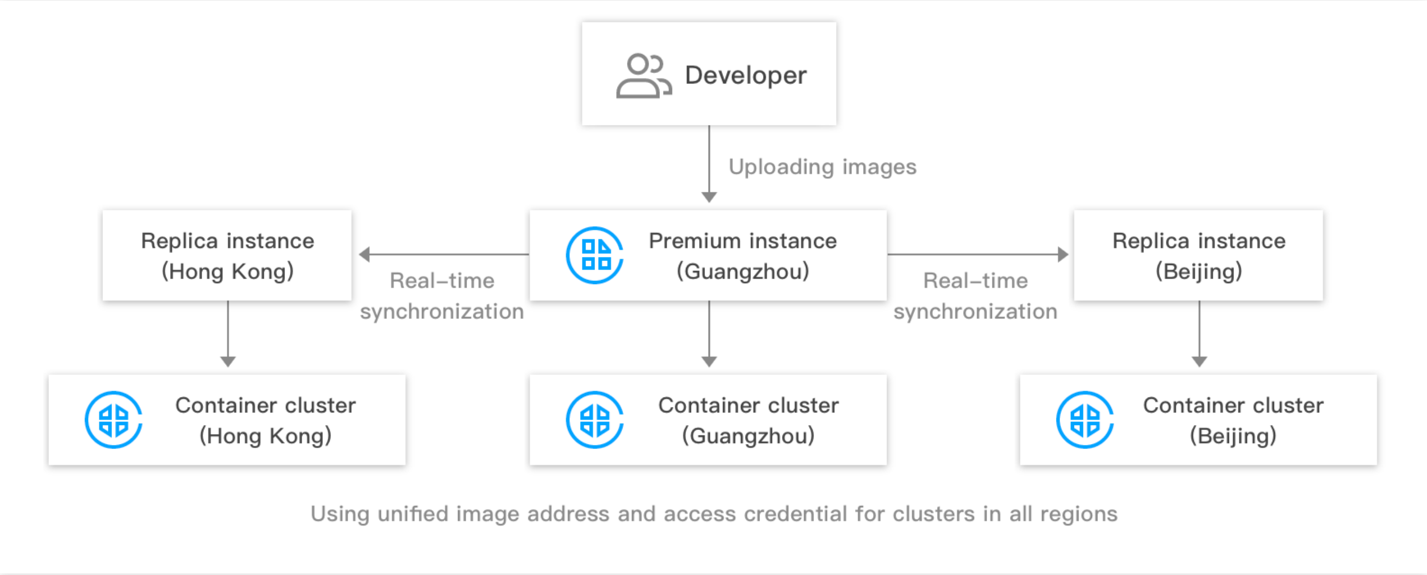
Configuring Instance Replication

Last updated：2021-08-11 17:39:31

Overview

Tencent Container Registry (TCR) allows users to create replicas of a premium instance in multiple regions with the same access domain name and credential as the original premium instance. It realizes the single-region upload, multi-region high-speed real-time synchronization, and image pulling from the nearest region over the private network. Compared with the cross-instance synchronization, this feature can unify the publish configuration of multi-regional clusters, improve the cross-region synchronization speed of cloud native artifacts, and help users realize the global synchronization update of service applications.

The instance replication feature allows users to create replicas of a premium instance in multiple regions. The replica instances will synchronize data with the primary instance in real time. Users can use the domain name and access credentials of the primary instance to access the replica instances through the private network. By using the instance replication feature, users can uniformly manage the application images of the multi-regional services, and do not need to purchase multiple Enterprise Edition instances, which can reduce the usage cost, increase the speed of container image distribution, and simplify the deployment and configuration.


Developer
Uploading images
Replica instance (Hong Kong)
Real–time synchronization
Premium instance (Guangzhou)
Real–time synchronization
Replica instance (Beijing)
Container cluster (Hong Kong)
Container cluster (Guangzhou)
Container cluster (Beijing)
Using unified image address and access credential for clusters in all regions

Prerequisites

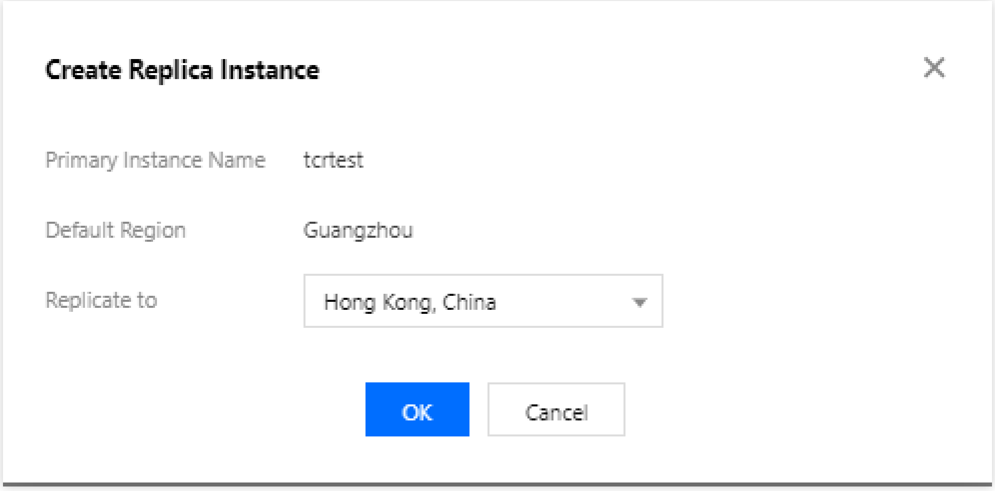
Before creating and managing the replica instance of a TCR Enterprise Edition instance, complete the following preparations:

- You have purchased an Enterprise Edition instance with premium specification.
- If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of the Enterprise Edition.

Directions

Creating and managing a replica instance

1. Log in to the TCR console and select **Synchronization and Replication** > **Instance Replication** in the left sidebar.
2. On the **Instance Replication** page, select a region and an instance, and click **Create**.
3. In the **Create Replica Instance** window, complete the following configurations:



Create Replica Instance

Primary Instance Name tcrtest

Default Region Guangzhou

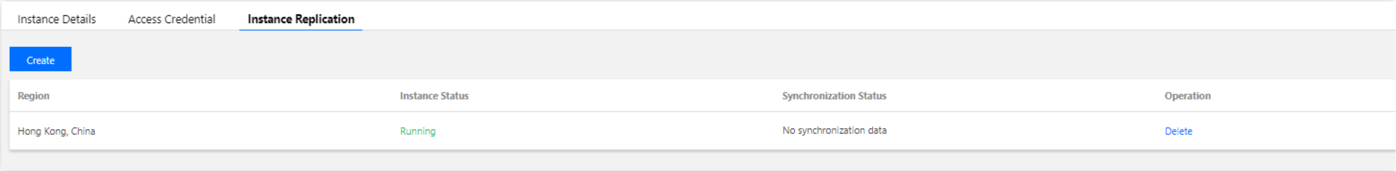
Replicate to Hong Kong, China

   - **Primary Instance Name**: the name of the currently selected premium instance.
   - **Default Region**: the region where the currently selected premium instance is located.
   - **Replicate to**: the region where the replica instance is located, which cannot be the same as the region of the current primary instance.
4. Click **Confirm** to complete the creation of the replica instance.

> Note
> You can delete the replication instances that are no longer needed in the specified region. If a premium instance has configured the replica instances, you need to delete all the replica instances first to delete the premium instance.

Accessing the replica instance via the private network

1. After completing the instance replication, you can push images to this instance through public network or private network. The container images will be replicated to the replication region in real-time, as shown in the figure below:



Region	Instance Status	Synchronization Status	Operation
Hong Kong, China	Running	No synchronization data	Delete

2. To ensure that the container clusters or CVMs in the replication region can access the instance through the private network, you need to connect the VPCs in the replication region to the instance. Please refer to Private Network Access Control, and choose the VPC in the replication region.
3. After the above configuration is completed, the container clusters or CVMs in the replication region can access the instance through the private network. The image access address and access credentials are the same as those of the premium instance in the original region.

Reference

You can also use the `CreateReplicationInstance` API to create the replica instance.

Configuring Image Tag Retention

Last updated：2021-02-26 17:46:38

Operation Scenario

Tencent Container Registry (TCR) supports the hosting and distribution of container images and provides an image building feature to enable image building, push, and hosting to be automatically triggered by code changes. If users need to quickly iterate their business apps and adopt an automated assembly line to generate images, large numbers of image tags will be generated continuously, rendering most old image tags unnecessary. If a single image repository contains too many image tags, the burden of tag management is huge, and the quota of image tags in the repository will be used up. Therefore, TCR provides the image tag retention feature to allow users to create custom rules for tag retention. Such rules can be triggered periodically to automatically clear image tags that fall outside the retention scope.

Tag retention rules support two types of retention policies: retaining the latest # tags pushed and retaining the tags pushed within # days, and simulated execution is supported.

Prerequisites

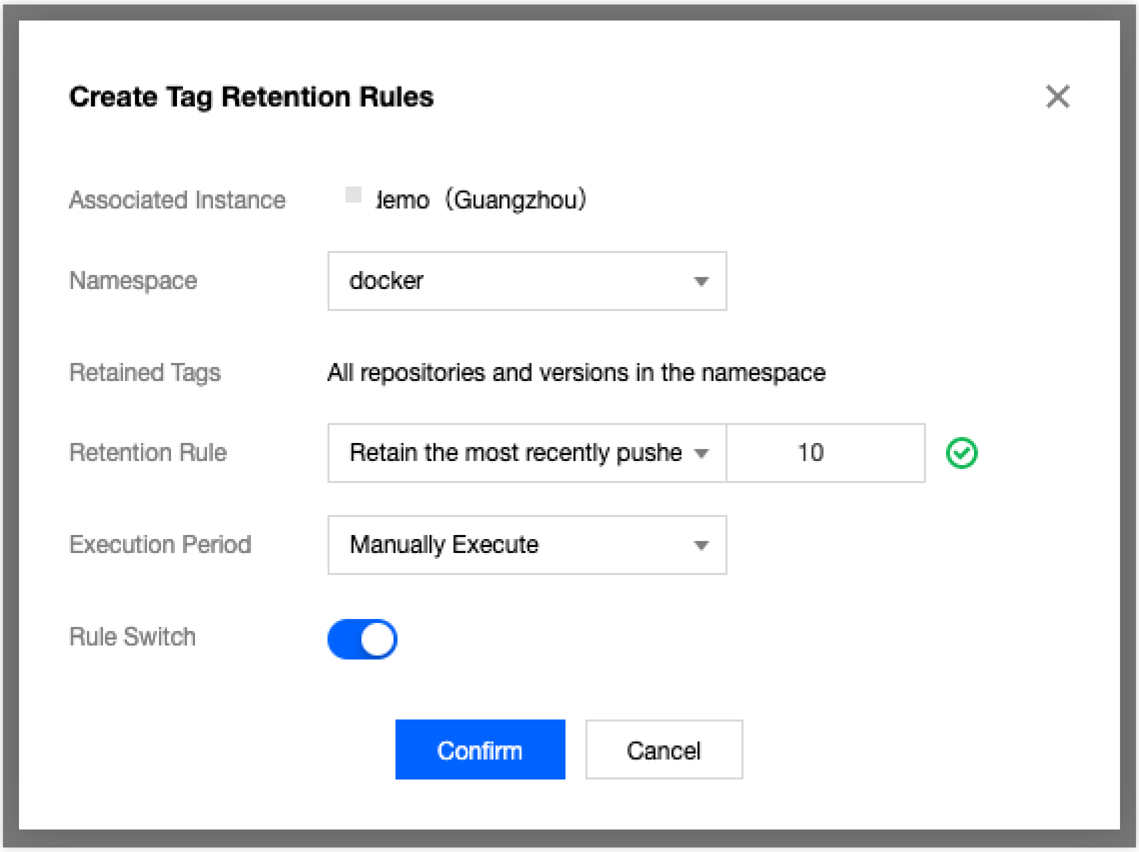
Before creating and managing an image repository for a TCR Enterprise Edition instance, complete the following tasks:

- Purchasing Instances.
- If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of the Enterprise Edition.

Directions

Creating tag retention rules

1. Log in to the TCR console and select **Tag Retention** in the left sidebar.
   On the "Tag Retention" page, you can view the list of tag retention rules for the current instance. To change the instance, select the desired instance name from the "Instance Name" drop-down list at the top of the page.
2. Click **Create**. In the "Create Tag Retention Rule" window, configure the rule based on the following information. See the figure below.



Create Tag Retention Rules

Associated Instance: demo（Guangzhou）

Namespace: docker

Retained Tags: All repositories and versions in the namespace

Retention Rule: Retain the most recently pushe | 10

Execution Period: Manually Execute

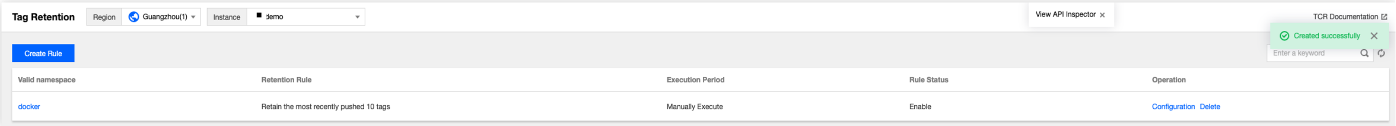
Rule Switch: on

- **Instance**: the current instance selected.
- **Namespace**: the namespace for which the tag retention rule takes effect. Currently, only one rule can be created for a single namespace.
- **Retained Tags**: by default, all repositories and tags in the namespace are retained and no filter is applied.
- **Retention Policy**: you can choose between retaining the most recent # tags pushed and retaining the tags pushed within # days and specify the number of tags or days accordingly.
- **Execution Cycle**: the cycle for executing the tag retention rule: manual, daily, weekly, or monthly execution.
- **Enable Rule**: by default, the rule is enabled.

3. Click **OK** to create the tag retention rule.

Managing tag retention rules

1. After successfully creating tag retention rules, you can view the created tag retention rules on the "Tag Retention" page. You can also perform the following operations to manage the rules. See the figure below:



Tag Retention — Region: Guangzhou(1) — Instance: demo

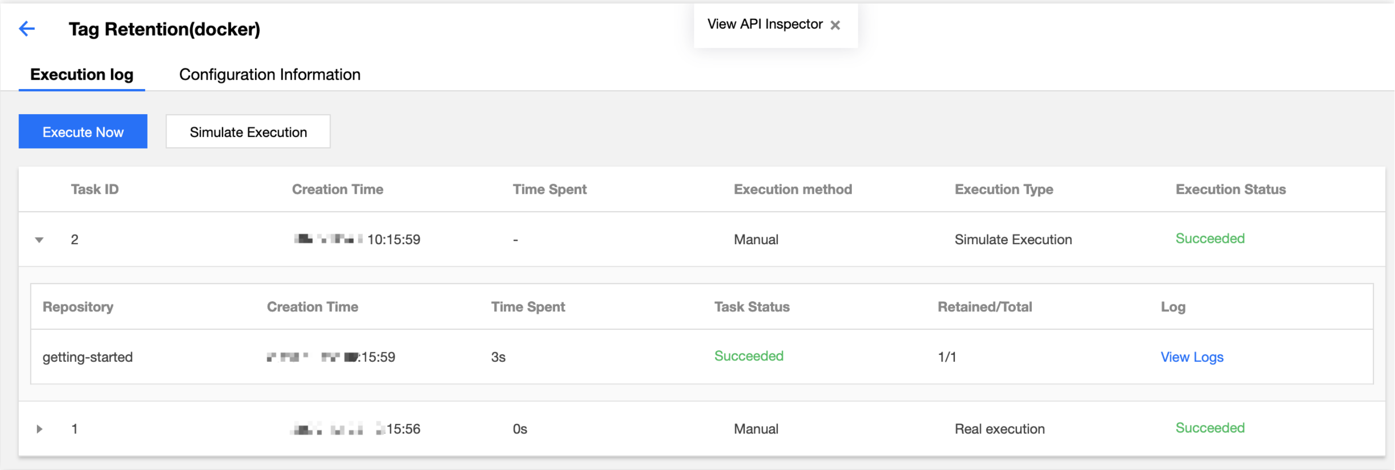
Created successfully

Valid namespace	Retention Rule	Execution Period	Rule Status	Operation
docker	Retain the most recently pushed 10 tags	Manually Execute	Enable	Configuration Delete

- **View the rule execution log**: you can click the name of a rule to view its triggering log. For more information, see Viewing the execution log.
- **Configure**: you can reconfigure a tag retention rule but cannot modify the namespace for which it takes effect.
- **Delete**: you can delete a tag retention rule.

Checking the execution log

1. Click the name of a tag retention rule to view the triggering log for the rule, as shown in the figure below:



	Task ID	Creation Time	Time Spent	Execution method	Execution Type	Execution Status
▾	2	[redacted] 10:15:59	-	Manual	Simulate Execution	Succeeded

Repository	Creation Time	Time Spent	Task Status	Retained/Total	Log
getting-started	[redacted]:15:59	3s	Succeeded	1/1	View Logs

	Task ID	Creation Time	Time Spent	Execution method	Execution Type	Execution Status
▸	1	[redacted] 15:56	0s	Manual	Real execution	Succeeded

   - **Task ID**: ID of a tag retention task, unique within the instance
   - **Creation Time**: time when a tag retention task was created
   - **Time Spent**: time consumed to complete all the tag retention tasks
   - **Execution Mode**: manual or automatic. You can click "Execute Now" or "Simulate Execution" for manual execution. Automatic execution is based on the cycle specified in the tag retention rule.
   - **Execution Type**: real execution or simulated execution. Simulated execution can be used to confirm whether a rule is effective, but it does not actually clear image tags.
   - **Execution status**: status of task completion

2. You can click a task ID to view the task details and click a specific repository to view its execution log.

Image Cleanup

Auto-Deleting Image Tags

Last updated：2023-05-08 16:15:16

Overview

Tencent Container Registry (TCR) supports the hosting and distribution of container images and provides the image building feature to enable image building, push, and hosting to be automatically triggered by code changes. If customers need to quickly iterate their applications, they can adopt an automated pipeline to generate images. Large number of image tags will be generated continuously, and the old image tags will no longer be used. If a single image repository contains too many image tags, the burden of tag management is huge, and the quota of image tags in the repository will be used up. Therefore, TCR provides the image tag retention feature to allow users to create custom rules for tag retention. Such rules can be triggered periodically to automatically delete the image tags that fall outside the retention scope.

Tag retention rules support two types of retention policies: retaining the latest # tags pushed and retaining the tags pushed within # days, and simulated execution is supported.

Notes

The tag retention feature only allows users to delete the image tags that are no longer in use based on rules.

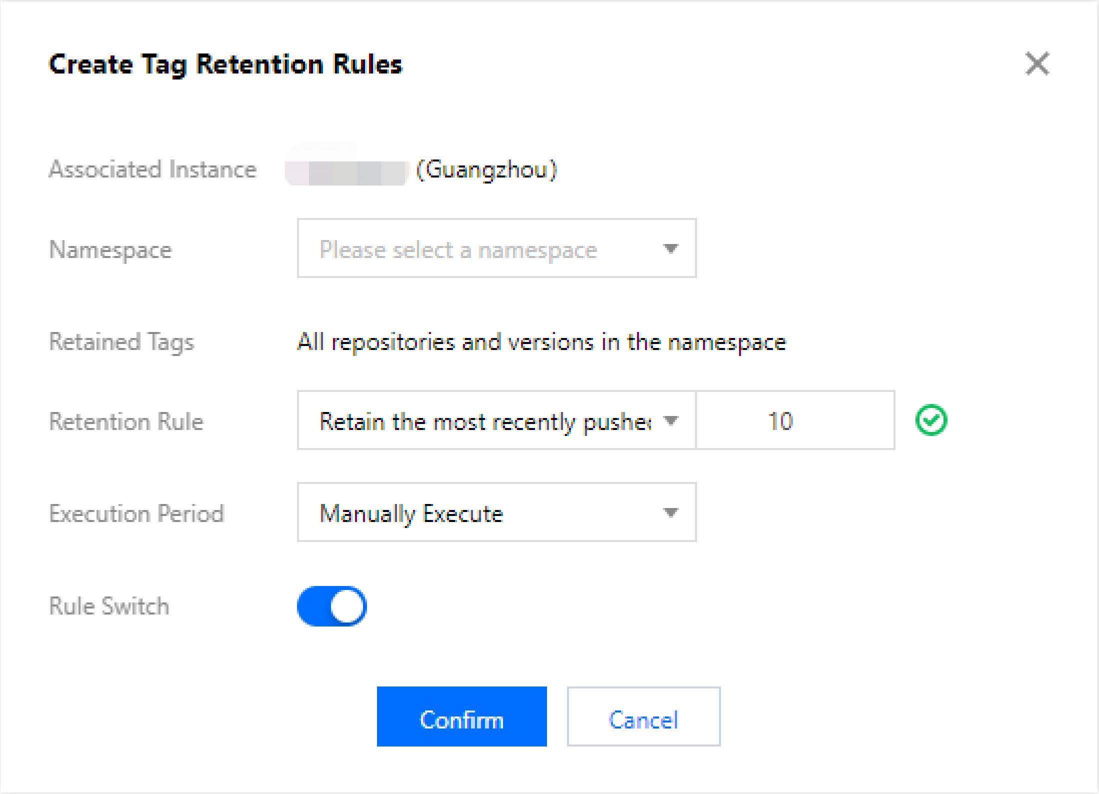
Directions

Creating tag retention rules

1. Log in to the TCR console and click **Tag Retention** in the left sidebar.

On the **Tag Retention** page, you can view the list of tag retention rules for the current instance. To change the instance, select the required instance name from the "Instance Name" drop-down list at the top of the page.

2. Click **Create Rule**. In the **Create tag retention rules** pop-up window, configure the rule, as shown in the figure below:



Create Tag Retention Rules

Associated Instance	[redacted] (Guangzhou)
Namespace	Please select a namespace
Retained Tags	All repositories and versions in the namespace
Retention Rule	Retain the most recently pushe 10
Execution Period	Manually Execute
Rule Switch	●

Associated Instance: Currently selected instance.

Namespace: Namespace for which the tag retention rule will take effect. Currently, only one rule can be created for a single namespace.

Retained Tags: By default, all repositories and tags in the namespace are retained and no filter is applied.

Retention Rule: You can choose between **Retain the most recently pushed # tags** and **Retain tags pushed within the last # days** and specify the number of tags or days accordingly.

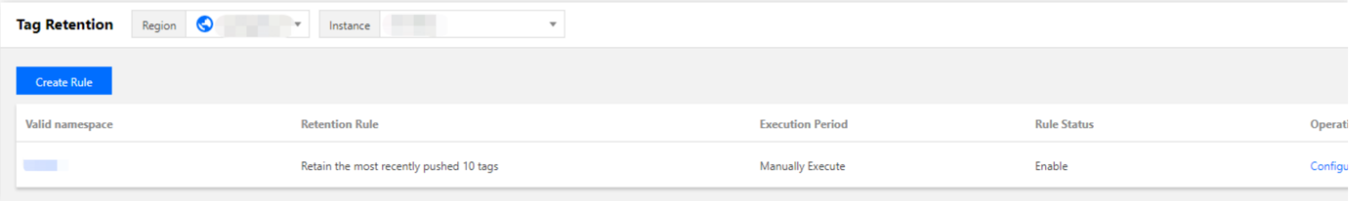
Execution Period: Cycle for executing the tag retention rule. Manual, daily, weekly, and monthly execution are supported.

Rule Switch: By default, the rule is enabled.

3. Click **Confirm** to create the tag retention rule.

Managing tag retention rules

After successfully creating tag retention rules, you can view the created tag retention rules on the **Tag Retention** page. You can also perform the following operations to manage the rules. See the figure below:



Tag Retention Region [redacted] Instance [redacted]

Valid namespace	Retention Rule	Execution Period	Rule Status	Operati
[redacted]	Retain the most recently pushed 10 tags	Manually Execute	Enable	Configu

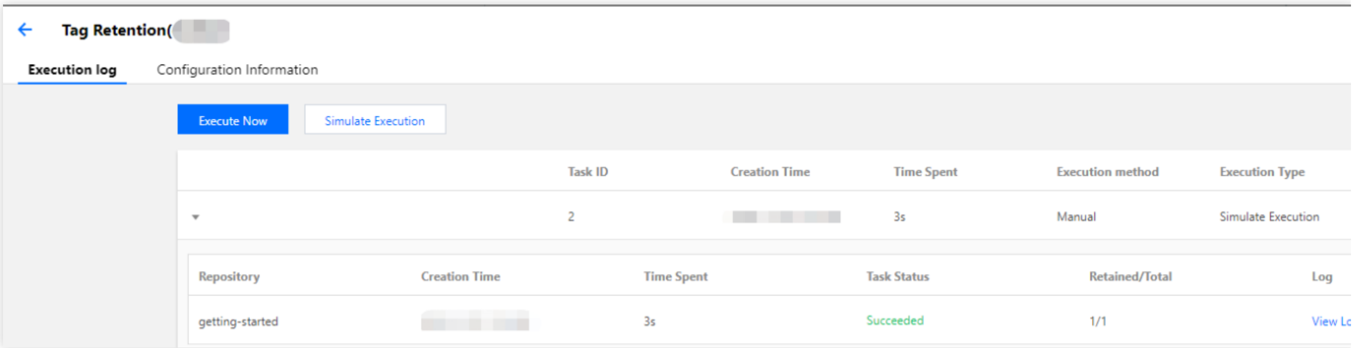
View the rule execution logs: You can click the name of a rule to view its triggering logs. For more information, see Viewing execution logs.

Configure: You can reconfigure a tag retention rule but cannot modify the namespace for which it takes effect.

Delete: You can delete a tag retention rule.

Viewing execution logs

1. Click the name of the target tag retention rule to view the triggering logs of the rule, as shown in the figure below:



	Task ID	Creation Time	Time Spent	Execution method	Execution Type
▾	2	[redacted]	3s	Manual	Simulate Execution

Repository	Creation Time	Time Spent	Task Status	Retained/Total	Log
getting-started	[redacted]	3s	Succeeded	1/1	View Lc

Task ID: ID of a tag retention task, unique within the instance.

Creation Time: Time when a tag retention task was created.

Time Spent: Time consumed to complete all the tag retention tasks.

Execution Method: Manual or automatic. You can click **Execute Now** or **Simulate Execution** for manual execution. Automatic execution is based on the cycle specified in the tag retention rule.

Execution Type: Real execution or simulate execution. Simulate execution can be used to confirm whether the rule is effective, but it does not actually clear image tags.

Execution Status: Status of task completion.

2. You can click a task ID to view the task details and click a specific repository to view its execution log.

References

You can also use the `CreateTagRetentionRule` API to create tag retention rules. For more information, see CreateTagRetentionRule.

Releasing COS Storage Capacity

Last updated：2023-05-08 15:44:59

Overview

TCR allows users to set up rules to remove unused image tags of the Enterprise Edition instances in batch. However, after you delete the image tags, the image data stored in the COS bucket associated with the instance still exists. You can use the garbage collection feature to delete the invalid image tag data in COS bucket to release the occupied storage capacity, and reduce the storage costs. This document describes how to use the garbage collection feature to release the occupied storage capacity of a COS bucket.

Notes

Garbage collection will affect the instance service status and the data in your instance. Please note the following points:

During garbage collection, the instance becomes read-only. You can pull images from image repositories, but cannot push images to the image repositories.

The garbage collection task will clean up the image layer data that is no longer associated with valid image tags in all image repositories of the instance. The deletion operation is irreversible. We recommend that you perform a dry run to evaluate the impacts before performing garbage collection.

The time required for the garbage collection task depends on the image data size stored in the COS bucket and the number of historical image tags. Temporary suspension of tasks is not supported. We recommend that you perform the garbage collection tasks during non-business time, or perform a dry run to estimate the required time.

Directions

Performing a dry run

1. Log in to the TCR console and click **Garbage Collection** in the left sidebar.
2. On the **Garbage Collection** page, view the garbage collection tasks of the current instance. To change the instance, select the required instance name from the **Instance Name** drop-down list at the top of the page.
3. Click **Dry run** and read the notes carefully.

Note

In a dry run, the instance is fully scanned for unused data. You can estimate the cleanup range and the required time. During the dry run, the instance's basic features are not affected. You can still pull and push images. However, the intensive computing tasks of the cleanup may affect the speed of image pulling and push.

4. Click **OK**.

Running garbage collection

1. Log in to the TCR console and click **Garbage Collection** in the left sidebar.
2. On the **Garbage Collection** page, view the garbage collection tasks of the current instance. To change the instance, select the required instance name from the **Instance Name** drop-down list at the top of the page.
3. Click **Run Garbage Collection** and read the notes carefully.

Note

During garbage collection, the instance becomes read-only. You can pull images from image repositories, but cannot push images to the image repositories. The time required for the job depends on the data size and usage duration of the instance.

4. Click **OK**.

DevOps

Managing Triggers

Last updated：2023-05-08 15:44:59

Overview

Tencent Container Registry (TCR) allows users to configure and use the flexible trigger (Webhook) feature. By configuring a proper trigger in an instance, you can quickly integrate existing R&D processes and CI/CD platforms and realize container DevOps scenarios such as image updates automatically triggering application deployment.

The trigger feature allows users to create custom trigger rules and view triggering logs. Trigger actions support the push, pull, and deletion of container images and Helm charts. Triggering rules support flexible regular expression filtering and regular filtering based on specified namespaces in an instance and configured image repositories and tags. This allows the trigger to be triggered by only certain repositories or image tags that use special naming formats.

The custom Header feature allows users to configure the Header for accessing the target URL in the `Key:Value` format, which is applicable to authentication and other scenarios.

Prerequisites

Before creating and managing a trigger in a TCR Enterprise Edition instance, complete the following tasks:

Purchase a TCR Enterprise Edition instance. The trigger feature is supported only in TCR Enterprise Edition.

If you are using a sub-account, you must have granted the sub-account operation permissions for the corresponding instance. For more information, see Example of Authorization Solution of TCR Enterprise.

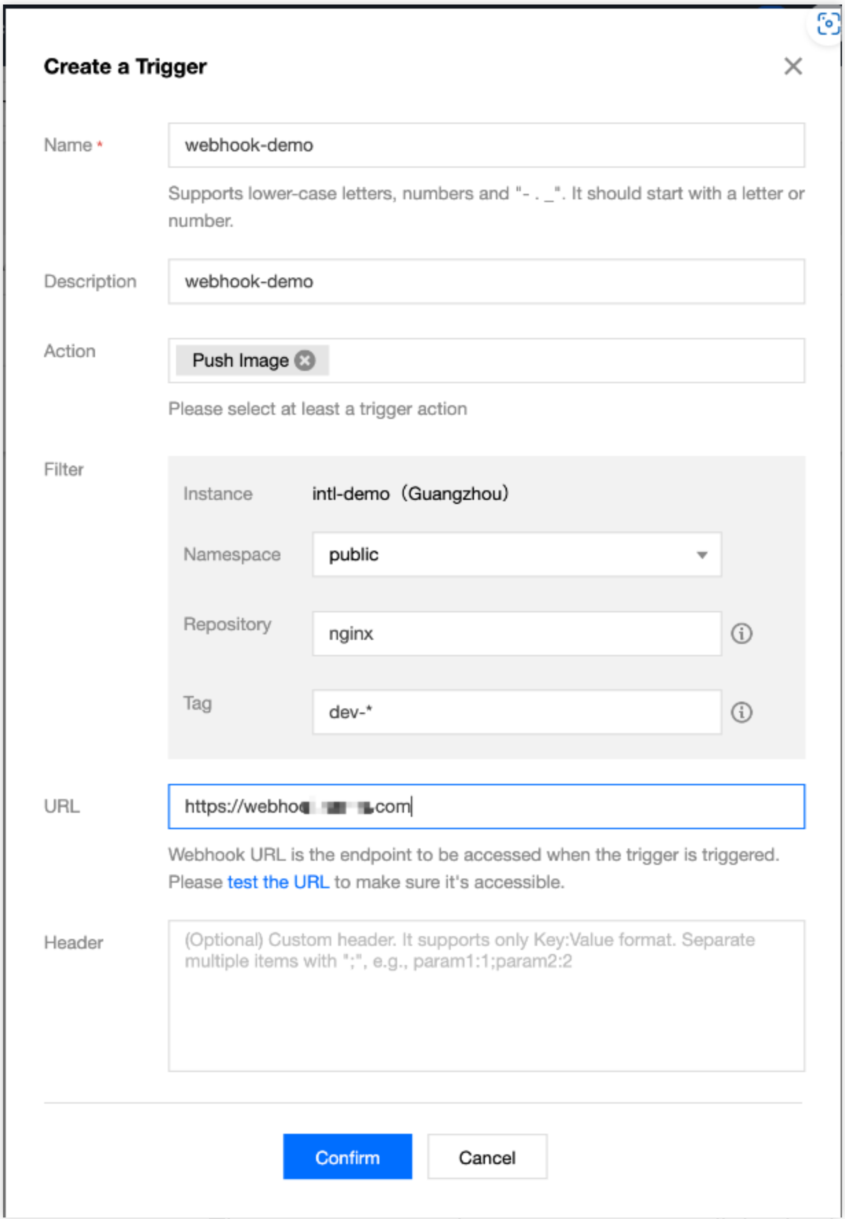
Directions

Creating a trigger

1. Log in to the TCR console and click **Trigger** in the left sidebar.

On the **Trigger** page, you can view the list of trigger rules for the current instance. To change the instance, select the desired instance name from the **Instance Name** drop-down list at the top of the page.

2. Click **Create**. In the **Create Trigger** pop-up window, specify the parameters and the trigger rule. You can use the following figure for reference:



Create a Trigger

Name *	webhook-demo Supports lower-case letters, numbers and "- . _". It should start with a letter or number.
Description	webhook-demo
Action	Push Image Please select at least a trigger action
Filter	Instance: intl-demo（Guangzhou） Namespace: public Repository: nginx Tag: dev-*
URL	https://webho[redacted].com Webhook URL is the endpoint to be accessed when the trigger is triggered. Please test the URL to make sure it's accessible.
Header	(Optional) Custom header. It supports only Key:Value format. Separate multiple items with ";", e.g., param1:1;param2:2

Name: Instance name. The name can contain lowercase letters, digits, hyphens (-), periods (.), and underscores (_), and must start with a letter or digit. In this document, `webhook-demo` is used as an example.

Description: Rule description.

Action: Four trigger actions are supported: push images, delete images, upload Helm charts, and delete Helm charts. During the execution of the webhook, the initiated webhook request contains information about the trigger action.

Triggering Rule:

Triggered Instance: The instance to which the webhook belongs, which is the currently selected instance and cannot be changed.

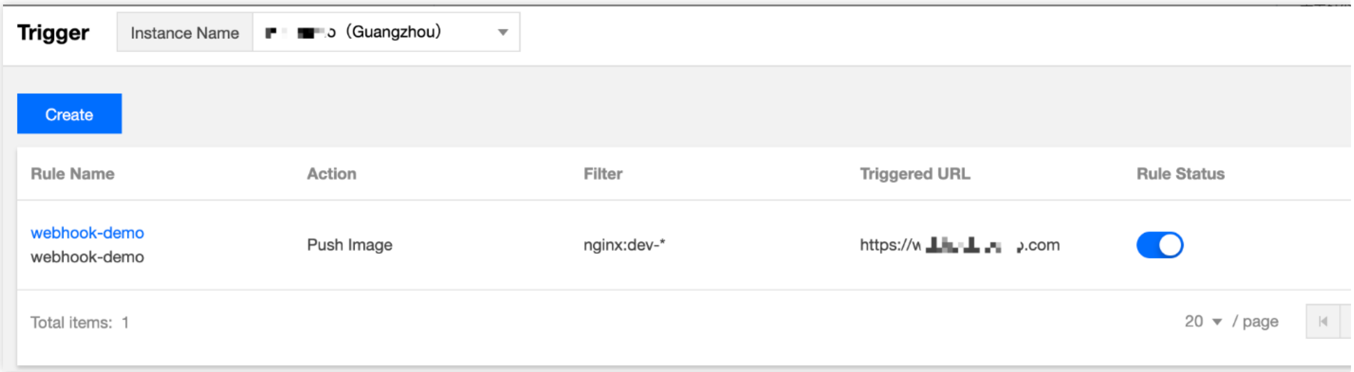
Namespace: The namespace for which the webhook takes effect. If the list is empty, create a namespace in the instance.

Repository Name: The name of the repository for which the webhook takes effect. Regular matching of image repositories and Helm charts is supported.

Tag: The tag for which the webhook takes effect. It supports regular matching. If you want the webhook to take effect for all tags, leave this parameter empty.
URL: The target URL for request initiation after the trigger fires, which is the URL that you specified for the webhook server. The trigger will send a POST request to the URL, and the request body contains the trigger action, trigger rule, and other information.
Header: The Header information in the Key:Value format to be carried in a POST request initiated by the trigger. Example: `Authentication: xxxxxxx` .
3. Click **OK**.

Managing trigger rules

After a trigger rule is created, you can view the trigger rule on the **Trigger** page and perform the following operations to manage the trigger rule:



Instance Name: [redacted] (Guangzhou)

Rule Name	Action	Filter	Triggered URL	Rule Status
webhook-demo webhook-demo	Push Image	nginx:dev-*	https://w[redacted].com	

Total items: 1 20 / page

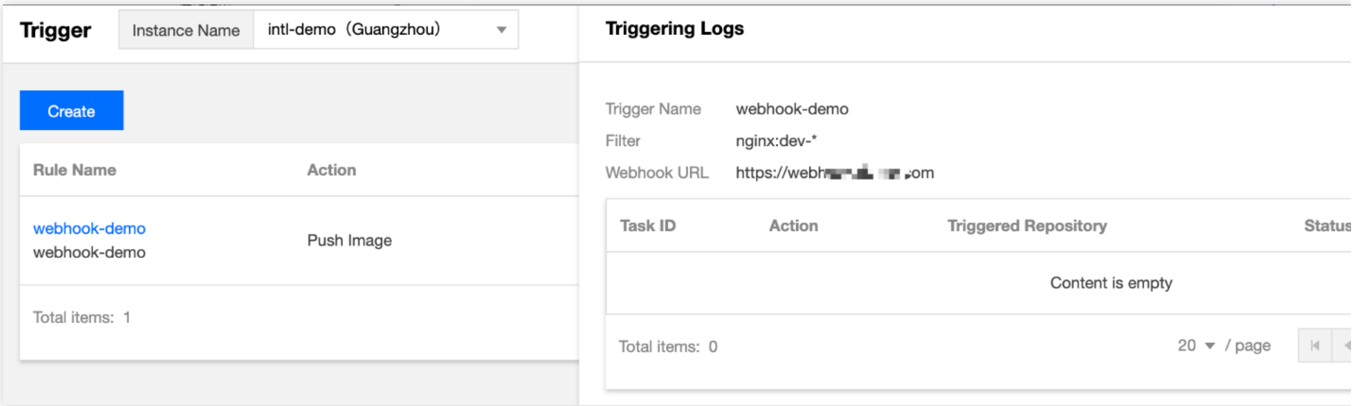
View Trigger Logs: You can click the name of the trigger rule or click **Trigger Logs** on the right side of the trigger rule to view the trigger logs of the rule. For more information, see Viewing trigger logs.
Modify Rule Status:

indicates that the rule is enabled, and

indicates that the rule is disabled. By default, a synchronization rule is enabled after it is created. You can change the status as needed.
Configuration: You can re-configure all parameters of the trigger rule.
Delete: You can delete the trigger rule.

Viewing trigger logs

You can click the name of a trigger rule or click **Trigger Logs** on the right side of the trigger rule to view the trigger logs of the rule. See the figure below:



Rule Name	Action
webhook-demo webhook-demo	Push Image

Total items: 1

Trigger Name webhook-demo
Filter nginx:dev-*
Webhook URL https://webh[redacted]om

Task ID	Action	Triggered Repository	Status
Content is empty			

Total items: 0

A log contains the following information:

Task ID: Trigger task ID, which is unique in the instance.

Action: The action that launched the trigger, such as image push.

Triggered Repository: The repository resources that launched the trigger.

Status: Success status of the trigger in executing the webhook request.

Creation Time: The time when the trigger was launched, which is also the time when the webhook request was initiated.

More Information

Webhook request format for reference

When users perform a relevant action on resources that meet a trigger rule, for example, pushing new images to the specified image repository, the relevant trigger is triggered and sends an HTTP POST request to the URL configured in the rule. The request body contains information such as the trigger action and repository path. The following is the resolved information of a sample request body after the trigger is triggered by image pushing. This sample is for reference in webhook server development.

```
{
  "type": "pushImage",
  "occur_at": 1589106605,
  "event_data": {
    "resources": [
      {
        "digest": "sha256:89a42c3ba15f09a3fbe39856bddacdf9e94cd03df7403cad4fc105xxx
        "tag": "v1.10.0",
        "resource_url": "xxx-bj.tencentcloudcr.com/public/nginx:v1.10.0"
      }
    ],
    "repository": {
      "date_created": 1587119137,
      "name": "nginx",
      "namespace": "public",
      "repo_full_name": "public/nginx",
      "repo_type": "public"
    }
  },
  "operator": "332133xxxx"
}
```

Using regular expressions to create rules

Regular matching rules

The following are the matching rules supported by the regular expression when you enter “repository name” or “version tag”:

`*` : matches all strings of any length that do not contain the path separator (`/`).

`**` : matches all strings of any length that contain the path separator (`/`).

Note:

`**` must be used as a complete relative path. If you use `/path**` , it will be the same as `/path*` and will only match the first-level repositories whose names are prefixed with path. To match all repositories under path, you should use `/path/**` . To match all repositories whose names are prefixed with path, you should use `/path*/**` .

`?` : matches any single character except ‘/’.

`{alt1, alt2, …}` : matches multiple regular expressions at the same time.

Use cases

Matches all repositories in the specified namespace	`**` or leave it empty
Matches all first-level repositories whose names are prefixed with path in the specified namespace	`/path*`
Matches all first-level repositories whose names are prefixed with path1 and path2 in the specified namespace	`/{path1, path2}*`
Matches all repositories under the path1 and path2 directories in the specified namespace	`/{path1, path2}/**`
Matches all repositories whose names are prefixed with path1 and path2 in the specified namespace	`/{path1, path2}*/**`
Matches all 1.x version tags in the specified repositories	`1.?`

Matches all version tags whose names are prefixed with env1 and env2 in the specified repository	`{env1*, env2*}`

OCI Artifacts Management

OCI Artifacts Management Overview

Last updated：2022-05-09 12:41:24

Overview

Tencent Container Registry (TCR) is compatible with OCI standard and supports hosting of multiple cloud-native artifacts including Docker Image, meeting the requirements of advanced users for hosting and distribution of Helm Chart, CNAB and custom OCI artifacts.

Currently, TCR Enterprise and TCR Individual instances support hosting of OCI artifacts. You can push OCI artifacts to image repositories, check the artifact type and pull commands.

For more about OCI artifacts and the usage, please see the official project opencontainers/artifacts on the GitHub.

Prerequisites

You must complete the following preparations before you can upload and manage OCI artifacts in the TCR instances.

- Create a TCR Enterprise instance or initialize the TCR Individual.
- If you are using a sub-account, you must have granted the sub-account required permissions for the instance. For more information, see Example of Authorization Solution of the TCR Enterprise.

Directions

Managing OCI artifacts in the console

1. Log in to the TCR console and select **Image Repository** in the left sidebar.
2. On the page that appears, you can view the list of image repositories in the current instance. It supports hosting of OCI artifacts by default. You can build OCI artifacts using the specific client tools and push the artifacts to the image repository.
3. Click the name of the desired image repository to go to the details page, where you can view the existing artifacts in the image repository.

References

Helm Charts Management

If you want to use Helm Charts, you can push them as OCI artifacts to the image repository for unified management. The Helm V3 tools are required if you choose this management method. Also, you can use the Helm Chart hosting feature provided by TCR Enterprise instances based on the Chart Museum open source project. For more information, see Helm Chart Hosting.

Management Helm Chart

Last updated：2023-05-08 15:44:59

Overview

Tencent Container Registry (TCR) can host Helm Charts to meet requirements for hosting and distribution of cloud-native applications. You can manage both container images and Helm Charts in the same namespace so that cloud-native deliverables of both container images and Helm Charts can be used in a business project.

Currently, only TCR Enterprise Edition instances support Helm Chart hosting and the use of the console or a Helm client to upload and download Helm Charts. Helm Chart repositories inherit the public or private attribute from their namespaces, and no extra configuration is needed. In terms of permission management, Helm Charts and container images share the **repository** resource type. This means the resource description **qcs::tcr:$region:$account:repository/tcr-xxxxxx/project-a**/* contains all image repositories and Helm Charts in the project-a namespace. You can flexibly use these image repositories and Helm Charts during resource permission management.

Prerequisites

Make sure that the following conditions are met before uploading and managing Helm Charts in a TCR Enterprise Edition instance:

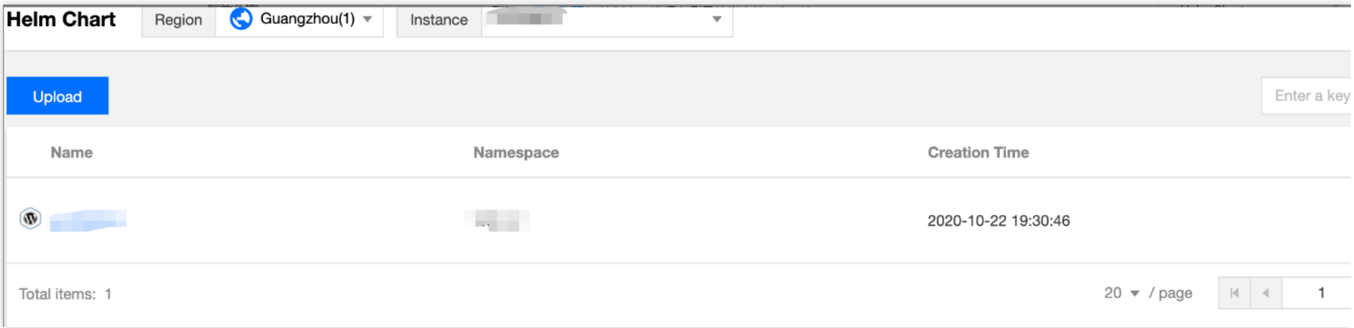
You have purchased a TCR Enterprise Edition instance.

If you are using a sub-account, the sub-account must have obtained operation permissions on the corresponding instance. For more information, see TCR Enterprise Authorization Management.

Directions

Managing Helm Charts in the console

1. Log in to the TCR console and click **Helm Chart** in the left sidebar.
2. On the **Helm Chart** page, view the list of Helm Charts in the current instance. To change the instance, select the required instance name from the **Instance Name** drop-down list at the top of the page, as shown in the figure below:



Name	Namespace	Creation Time
[redacted]	[redacted]	2020-10-22 19:30:46

Total items: 1

The Chart list contains the following information and supported operations:

Name: Helm Chart name. You can click it to enter the Chart details page, where you can view and manage each version of the Chart. You can also view the file details of each version of the Chart package on the **Basic Information** tab.

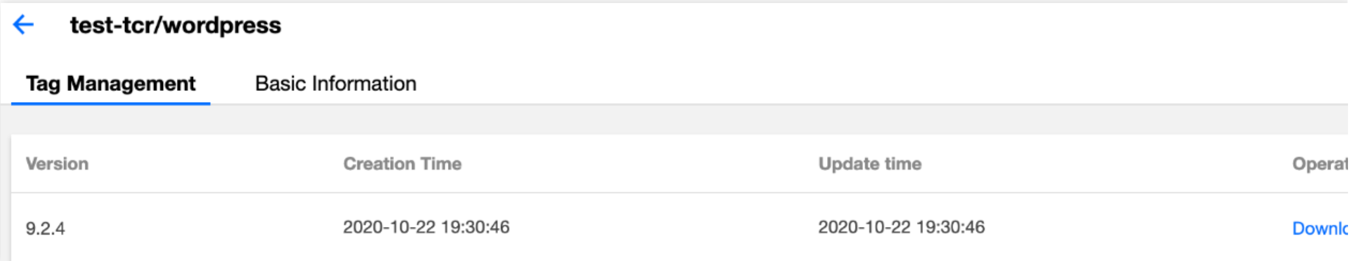
Namespace: Namespace to which a Helm Chart belongs.

Create Time: Time when the Helm Chart was pushed to the repository for the first time.

Operation: Click **Use commands** to obtain the commands for the current repository. For more information, see Using the Helm client to upload and download Helm Charts. Click **Delete** to delete the current repository.

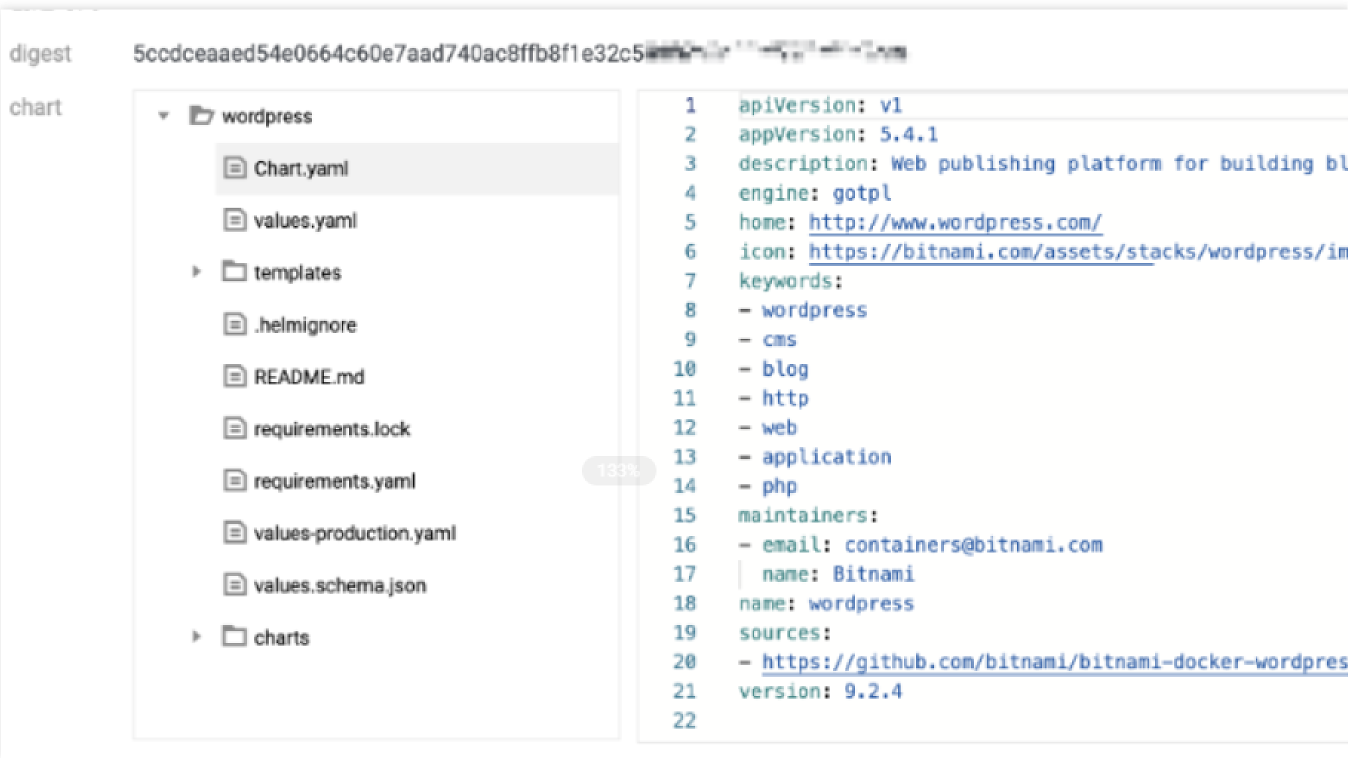
3. Click the name of the specified Helm Chart repository to enter the repository details page.

Version Management: This page displays the existing Chart versions in the current repository, and you can **download** or **delete** the specified versions, as shown in the figure below:



Version	Creation Time	Update time	Operat
9.2.4	2020-10-22 19:30:46	2020-10-22 19:30:46	Downlc

Basic Information: This page displays the Chart version details, such as Chart.yaml, as shown in the figure below:



digest 5ccdceaaed54e0664c60e7aad740ac8ffb8f1e32c5[redacted]

```
apiVersion: v1
appVersion: 5.4.1
description: Web publishing platform for building bl
engine: gotpl
home: http://www.wordpress.com/
icon: https://bitnami.com/assets/stacks/wordpress/im
keywords:
- wordpress
- cms
- blog
- http
- web
- application
- php
maintainers:
- email: containers@bitnami.com
  name: Bitnami
name: wordpress
sources:
- https://github.com/bitnami/bitnami-docker-wordpres
version: 9.2.4

```

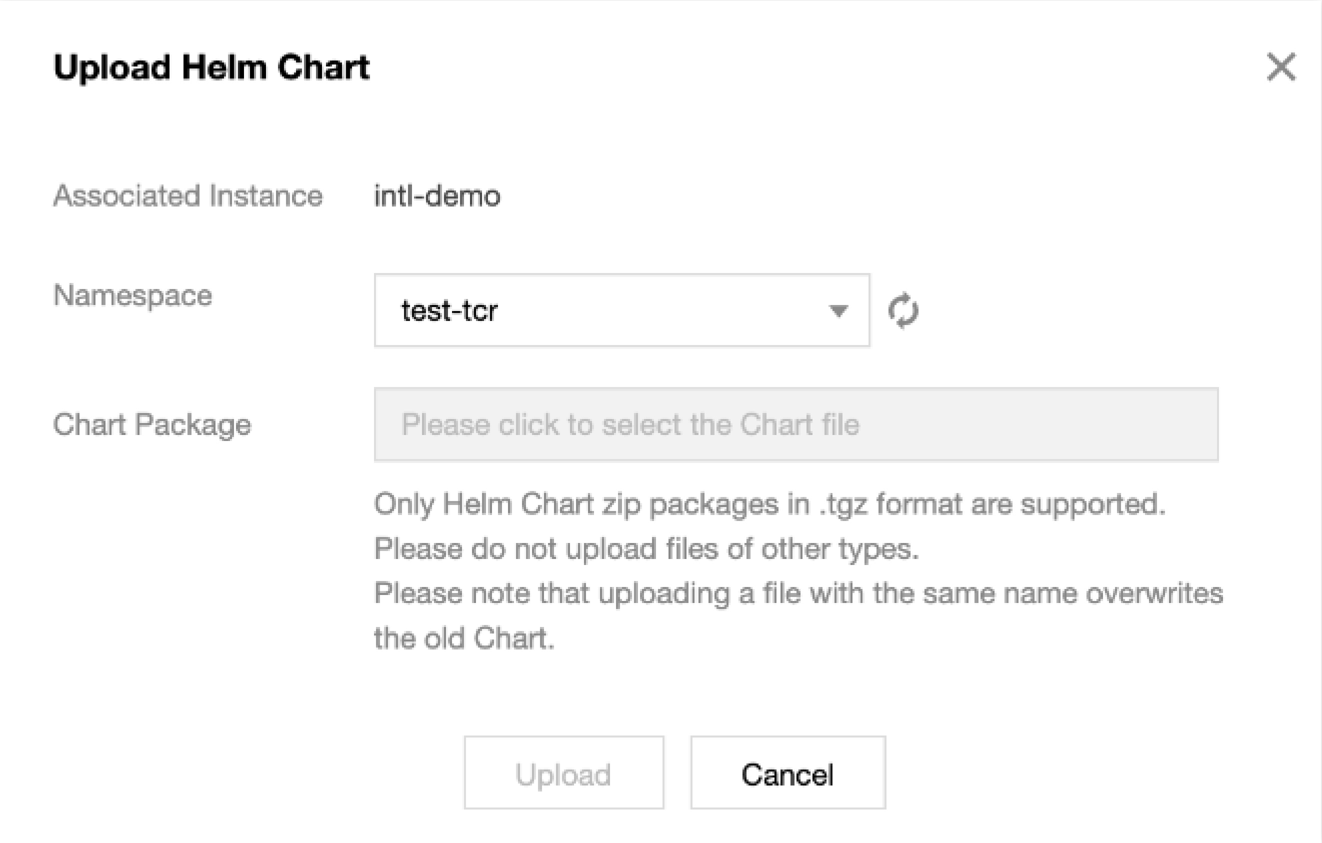
Using the console to upload and download Helm Charts

Uploading a local Helm Chart package

1. Log in to the TCR console and click **Helm Chart** in the left sidebar.

On the **Helm Chart** page, you can view the list of Helm Chart repositories in the current instance. To change the instance, select the required instance name from the **Instance Name** drop-down list at the top of the page.

2. Click **Upload**. In the **Upload Helm Chart** pop-up window, configure settings as shown in the figure below:



Upload Helm Chart

Associated Instance intl-demo

Namespace test-tcr

Chart Package Please click to select the Chart file

Only Helm Chart zip packages in .tgz format are supported. Please do not upload files of other types.
Please note that uploading a file with the same name overwrites the old Chart.

Associated Instance: Current instance selected.
Namespace: Namespace to which the Helm Chart repository belongs. If the list is empty, create a namespace in the instance.
Chart Package: Click to select a Helm Chart package that has been downloaded to the local system.
Note
Only Helm Chart packages in .tgz format are supported. Please avoid uploading other types of files. Note that uploading a file will overwrite the existing Chart with the same name.
3. Click **Upload** to start uploading the Helm Chart package. After uploading, you can view the uploaded Helm Chart on the repository list page. If the uploaded package does not have a corresponding Helm Chart repository, a new repository will be created automatically.

Downloading a Helm Chart package to the local system

1. View the Helm Chart repository list in the current instance on the **Helm Chart** page. Click the specified repository to enter its version management page.
2. Select the specified version in the Chart repository, click **Download** on the right of the row where the version is located, and the Chart package on this version will be automatically downloaded to the local system. Depending on the browser and configuration, you can choose to specify the download path.

Using the Helm client to upload and download Helm Charts

Installing a Helm client

Note

Note that, if you wish to use Helm in Tencent Kubernetes Engine (TKE), you need to select a v3.x.x version. You can run the `helm version -c` command to check the version of the installed client.

This document takes the installation on a Linux node as an example. For installation on other operating systems, download the corresponding installation package.

Run the following commands in sequence to download and install the Helm client. For more information, see Installing Helm.

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scrip
```

```
chmod 700 get_helm.sh
```

```
./get_helm.sh
```

Adding a Helm repository

1. Log in to the TCR console and go to the **Instance list** page. On this page, click the name of the target instance to go to the details page.

2. Obtain the username and password used to log in to the instance. For more information, see Obtaining an Instance Access Credential.

3. Run the following command on the node to add the namespace, which is used to manage Helm Charts, to the local Helm repository.

Note

Ensure that the server running this command is in the Internet allowlist or connected VPC of the corresponding instance. For more information, see Configuring Public Network Access Control and Configuring Private Network Access Control.

```
helm repo add $instance-$namespace https://$instance.tencentcloudcr.com/chartrepo/$
```

`$instance-$namespace` : Name of the Helm repository. We recommend that you use the combination of **instance name + namespace name** for naming so as to distinguish between instances and namespaces.

`https://$instance.tencentcloudcr.com/chartrepo/$namespace` : Remote address of the Helm repository.

`$username` : Username obtained in Step 2.

`$instance-token` : Password obtained in Step 2.

If the add operation is successful, the following message will be prompted.

```
"$instance-$namespace" has been added to your repositories
```

Pushing Helm Charts

1. Install the helm-push plugin.

Note

Install the helm-push plugin of the v0.10.0 or a later version. Otherwise, version incompatibility may cause Helm Chart package pushing to fail.

If the version of the helm-push plugin you use is earlier than v0.10.0, replace `helm cm-push` with the `helm push` command.

To upload Chart packages by using the Helm CLI, you need to install the helm-push plugin. The plugin supports using the `helm push` command to push Helm Charts to the specified repository, as well as uploading directories and compressed packages.

```
helm plugin install https://github.com/chartmuseum/helm-push
```

2. Run the following command on the node to create a Chart:

```
helm create tcr-chart-demo
```

3. (Optional) Run the following command to directly push the specified directory to the Chart repository:

```
helm cm-push tcr-chart-demo $instance-$namespace
```

Here, `$instance-$namespace` is the name of the added local repository.

4. Run the following commands to compress the specified directory and push it to the Chart repository:

```
tar zcvf tcr-chart-demo-1.0.0.tgz tcr-chart-demo/
```

```
helm cm-push tcr-chart-demo-1.0.0.tgz $instance-$namespace
```

Here, `$instance-$namespace` is the name of the added local repository.

Pulling Helm Charts

1. Run the following command on the node to obtain the latest Chart information:

```
helm repo update
```

2. Run the following command to pull the Helm Chart of the specified version:

```
helm fetch <Local repository name>/<Chart name> --version <Chart version>
```

In the following example, tcr-chart-demo 1.0.0 in the project-a namespace is pulled from the tcr-demo TCR Enterprise Edition instance:

```
helm fetch tcr-demo-project-a/tcr-chart-demo --version 1.0.0
```

Operation Guide for TCR Individual

Resetting the Login Password

Last updated：2024-07-01 18:13:17

Overview

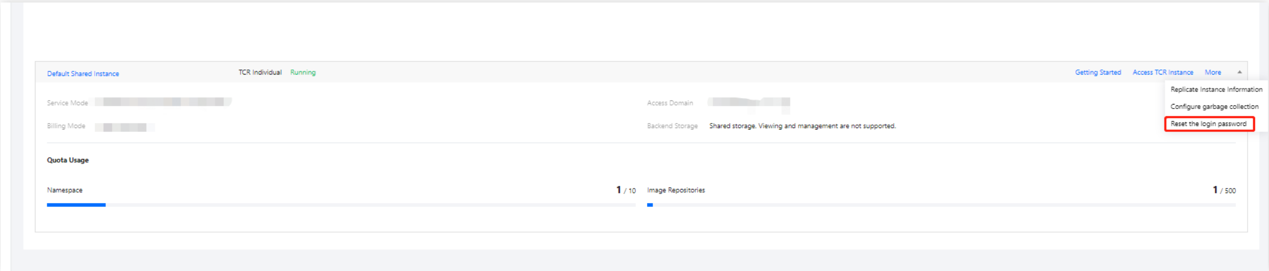
This document describes how to reset the login password for a TCR Individual instance.

Notes

The login password for a TCR Individual instance is a fixed password, which is consistently applied among all regions.

Directions

1. Log in to the TCR console and select **Instance Management** in the left sidebar.
On the page that appears, you can view the list of instances under the current account. Select a TCR Individual instance in any region.



Default Shared Instance TCR Individual Running

Service Mode [redacted]

Billing Mode [redacted]

Access Domain [redacted]

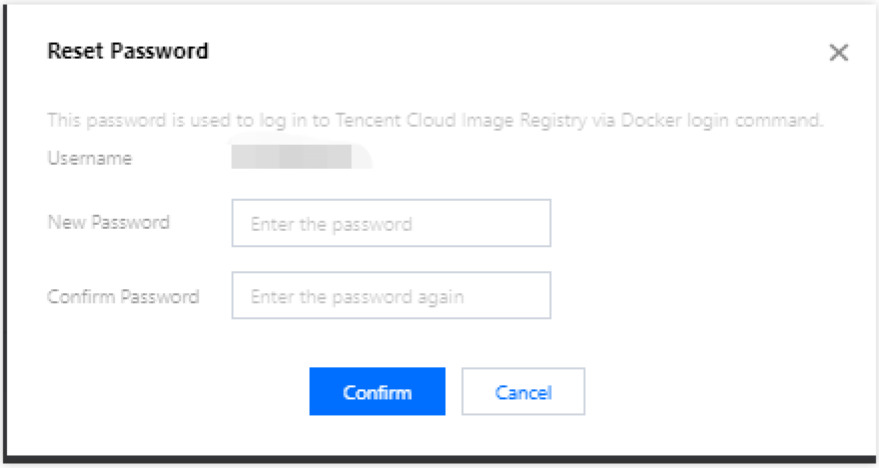
Backend Storage Shared storage. Viewing and management are not supported.

Quota Usage

Namespace 1 / 10

Image Repositories 1 / 500

2. Click **More** > **Reset the login password**, as shown in the following figure.



Reset Password

This password is used to log in to Tencent Cloud Image Registry via Docker login command.

Username

New Password: Enter the password

Confirm Password: Enter the password again

Username: The current Tencent Cloud account ID.

New password: The new password you want to use. It is recommended that you configure a strong password.

Confirm password: Enter the password again.

3. Click **Confirm** to complete the process.

Configuring Access Permission

CAM APIs for Personal Edition

Last updated：2021-04-08 10:41:06

Namespace APIs

APIs and Description	Resource Type	Six-segment Example of Resource
CreateNamespacePersonal Creating a namespace of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace`
DeleteNamespacePersonal Deleting a namespace of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace`

Image Repository APIs

APIs and Description	Resource Type	Six-segment Example of Resource
DescribeRepositoryOwnerPersonal Querying all repositories of Personal Edition	repo	`qcs::tcr:$region:$account:repo/*`
CreateRepositoryPersonal Creating an image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$`
DeleteRepositoryPersonal Deleting an image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$`
BatchDeleteRepositoryPersonal Deleting the image repositories of Personal Edition in batches	repo	`qcs::tcr:$region:$account:repo/$namespace/*`
DeleteImagePersonal Deleting the repository tag of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$`

APIs and Description	Resource Type	Six-segment Example of Resource
BatchDeleteImagePersonal Deleting the repository tags of Personal Edition in batches	repo	`qcs::tcr:$region:$account:repo/$namespace/$`
PullRepositoryPersonal Pulling the images in the image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$`
PushRepositoryPersonal Pushing the images in the image repository of Personal Edition	repo	`qcs::tcr:$region:$account:repo/$namespace/$`

Example of Authorization Solution of TCR Individual

Last updated：2022-05-09 12:41:24

Policy Configuration in Typical Scenarios

> Note：
>
> The following scenario policies are only used for TCR Individual use cases.

- Grant a sub-account the full read/write permissions for all resources in TCR Individual.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
"resource": [
"qcs::tcr:::repo/*"
],
"effect": "allow"
}]
}
```

- Grant a sub-account the read-only permission for all resources in TCR Individual (former Image Repositories in TKE).

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepository*"
],
"resource": [
"qcs::tcr:::repo/*"
],
"effect": "allow"
}]
}
```

- Grant a sub-account permissions to manage the specific namespace in the specific region. For example, the namespace `team-01` in the default region.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:*"
],
"resource": [
"qcs::tcr:::repo/team-01",
"qcs::tcr:::repo/team-01/*"
],
"effect": "allow"
}
]
}
```

- Grant a sub-account the read-only permission for an image repository, which means that the sub-account can only pull the images in the image repository instead of deleting the repository, modifying repository attributes, or pushing images. For example, the image repository `repo-demo` in the namespace `team-01` in the default region.

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:Describe*",
"tcr:PullRepositoryPersonal"
],
"resource": [
"qcs::tcr:::repo/team-01",
"qcs::tcr:::repo/team-01/repo-demo",
"qcs::tcr:::repo/team-01/repo-demo/*"
],
"effect": "allow"
}
]
}
```

Update Guide of Resource Level APIs and Authorization Solution of Personal Edition

Last updated：2021-04-08 10:41:06

Overview

TCR provides container image hosting and distribution services to enterprise users and personal users. The Personal Edition provides users with simple and free basic services, that is, the image repository in TKE.
To provide users with more standardized interface definitions and significantly reduced access delay API services, the APIs of the original Personal Edition image repository (CCR) has been upgraded from version 2.0 to the latest version 3.0, and the API name and authorization solutions have updated accordingly. This document describes the mappings between the new and legacy APIs after the upgrade of the APIs that support resource-level authentication and how to use the new authorization solution.

Mappings Between the v2.0 and v3.0 APIs

API Name of v2.0	API Name of v3.0	Description	Latest Resource Description M
CreateCCRNamespace	CreateNamespacePersonal	Creates a namespace of Personal Edition	`qcs::tcr:$region:$acc`
DeleteUserNamespace	DeleteNamespacePersonal	Deletes a namespace of Personal Edition	`qcs::tcr:$region:$acc`
GetUserRepositoryList	DescribeRepositoryOwnerPersonal	Queries all repositories of Personal Edition	`qcs::tcr:$region:$acc`
CreateRepository	CreateRepositoryPersonal	Creates an image repository of Personal Edition	`qcs::tcr:$region:$acc`

DeleteRepository	DeleteRepositoryPersonal	Deletes an image repository of Personal Edition	`qcs::tcr:$region:$acc`
BatchDeleteRepository	BatchDeleteRepositoryPersonal	Deletes image repositories of Personal Edition in batches	`qcs::tcr:$region:$acc`
DeleteTag	DeleteImagePersonal	Deletes the repository tag of Personal Edition	`qcs::tcr:$region:$acc`
BatchDeleteTag	BatchDeleteImagePersonal	Deletes the repository tags of Personal Edition in batches	`qcs::tcr:$region:$acc`
pull	PullRepositoryPersonal	Pulls the images in the image repository of Personal Edition	`qcs::tcr:$region:$acc`
push	PushRepositoryPersonal	Pushes the images in the image repository of Personal Edition	`qcs::tcr:$region:$acc`

Mappings Between the legacy and new Authorization Solutions

Due to the upgrade and update of the product name and API version, the original resource description methods and actions of the TCR Personal Edition have been updated accordingly. Please use the latest resources and action authorization solutions while using the v3.0 APIs.

During the upgrade of APIs, CAM APIs will be compatible with both the legacy and new resource description methods and actions to ensure that the custom policies are still effective. To make it easier for you to manage the APIs and authorization solutions uniformly, we recommend that you upgrade the authorization solution to the latest version. For more information, please see Example of Authorization Solution of the Personal Edition.

The legacy resource-level authorization solution

- **Action**: use `ccr` as the product prefix, and the API name is version 2.0. For example, create a namespace as `ccr:CreateCCRNamespace` .
- **Resource description**: use `ccr` as the product name, and there is only a `repo` resource type. For example, to describe the image repository `repo-b` under the namespace `namespace-a` , it would be `qcs::ccr:::repo/namespace-a/repo-b` . If `$region` and `$account` are left empty, all regions will be used by default, and the account will be the root account of the CAM user who created the policy by default.
  For more information on authorization solution, see TKE Image Registry Resource-level Permission Settings.

The new resource-level authorization solution

- **Action**: use `tcr` as the product prefix, and the API name is version 3.0. For example, create a namespace of Personal Edition as `tcr:CreateNamespacePersonal` .
- **Resource description**: use `tcr` as the product name, and there are three resource types: `instance` , `repository` and `repo` . Among them, `repo` is a dedicated resource type of the Personal Edition. For example, to describe the image repository `repo-b` under the namespace of Personal Edition `namespace-a` , it would be `qcs::tcr:$region:$account:repo/namespace-a/repo-b` . If `$region` and `$account` are left empty, all regions will be used by default, and the account will be the root account of the CAM user who created the policy by default.
  For more information on authorization solution, see CAM APIs for Personal Edition and Example of Authorization Solution of the Personal Edition.

The compatible example of legacy and new authorization solutions

For example, the authorized sub-account can read the image repository of `repo-b` (an image repository of Personal Edition) under the namespace `namespace-a` in the default region. Then this account can only query the repository information and pull the image in the repository, but cannot modify the repository attributes, push images, and delete the repository.

- **Legacy authorization solution:**

```
{
"version": "2.0",
"statement": [{
```

```
"action": [
"ccr:pull"
],
"resource": "qcs::ccr:::repo/namespace-a/repo-b",
"effect": "allow"
}]
}
```

- **New authorization solution:**

```
{
"version": "2.0",
"statement": [{
"action": [
"tcr:PullRepositoryPersonal"
],
"resource": "qcs::tcr:::repo/namespace-a/repo-b",
"effect": "allow"
}]
}
```

Configuring Garbage Collection

Last updated：2022-05-09 12:41:24

Overview

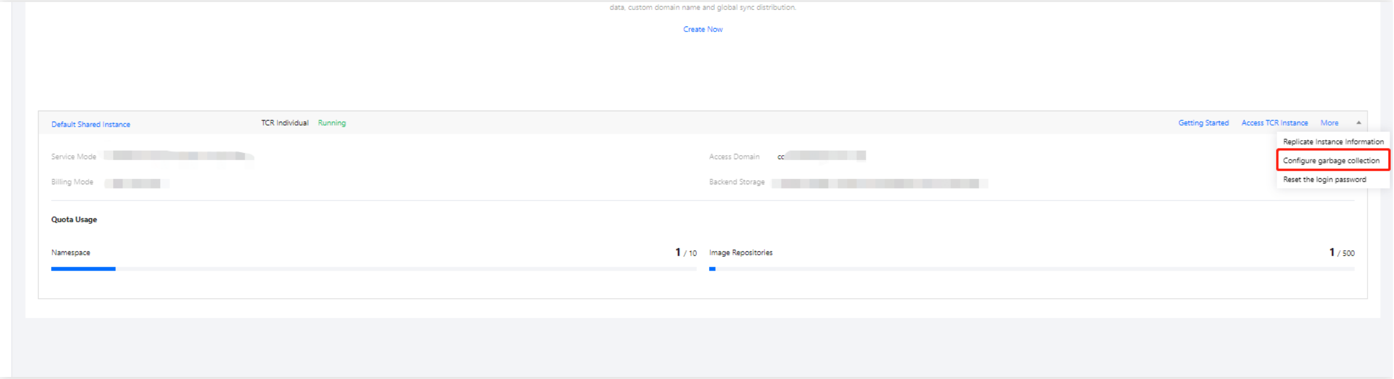
This document describes how to configure/update a garbage collection policy for TCR Individual instances.

Notes

A garbage collection policy for TCR Individual instances is only applicable for the instances in the specified regions. For example, TCR Individual instances deployed in Guangzhou and Silicon Valley regions can be configured with other different garbage collection policies.

Directions

1. Log in to the TCR console and select **Instance Management** in the left sidebar.
   On the page that appears, you can view the list of instances under the current account. Select a TCR Individual instance in any region.



data, custom domain name and global sync distribution.

Default Shared Instance TCR Individual Running

Service Mode [redacted]

Billing Mode [redacted]

Access Domain cc[redacted]

Backend Storage [redacted]

Quota Usage

Namespace 1 / 10

Image Repositories 1 / 500

2. Click **More** > **Configure garbage collection**, as shown in the following figure.

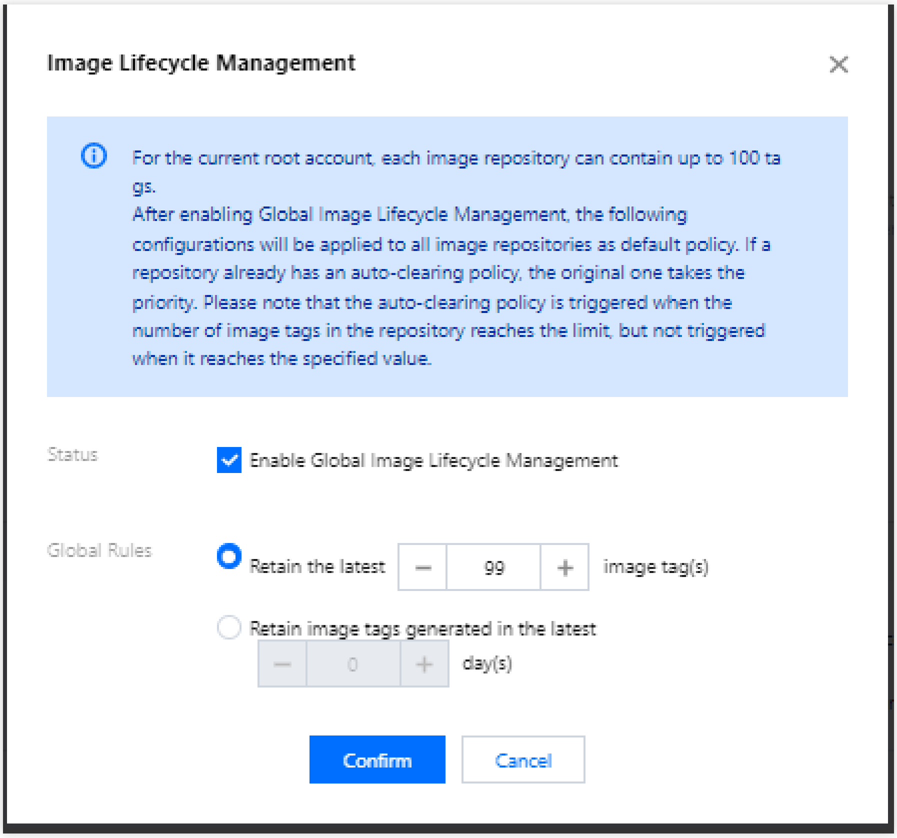


Image Lifecycle Management

For the current root account, each image repository can contain up to 100 ta gs.
After enabling Global Image Lifecycle Management, the following configurations will be applied to all image repositories as default policy. If a repository already has an auto-clearing policy, the original one takes the priority. Please note that the auto-clearing policy is triggered when the number of image tags in the repository reaches the limit, but not triggered when it reaches the specified value.

Status ☑ Enable Global Image Lifecycle Management

Global Rules ● Retain the latest 99 image tag(s)

○ Retain image tags generated in the latest 0 day(s)

- **On/Off**: Select **Enable Global Image Lifecycle Management**.
- **Global rules**:
    - **Retain the latest XX image tag**: Enter a number based on your actual needs. The number cannot exceed the default quota of image tags under the current root account.
    - **Retain the image tags in the last XX day**: Enter a number based on your actual needs.

3. Click **OK**.

Image Lifecycle Management(old)

Last updated：2021-12-06 17:36:32

Overview

Image lifecycle management supports global configuration and independent configuration for a specified image repository. If global image lifecycle management is enabled, the configured global rules are applied to all images of the root account. You can also configure an image tag automatic clearing policy for a specified image repository, and this policy has higher priority than the global configuration.

This document describes how to implement image lifecycle management through the TKE console.

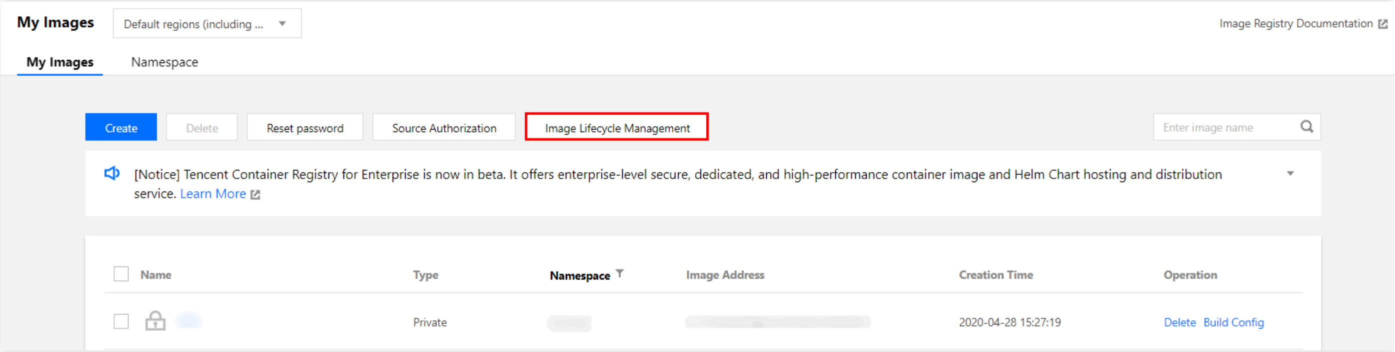
Notes:

- For an image repository, the image tag automatic clearing policy configured specifically for this image repository has **higher priority than** a global image lifecycle management policy.
- After a global image lifecycle management policy is enabled, image tags are automatically cleared only after the number of image tags of the root account reaches the default quota.

Directions

Configuring global image lifecycle management

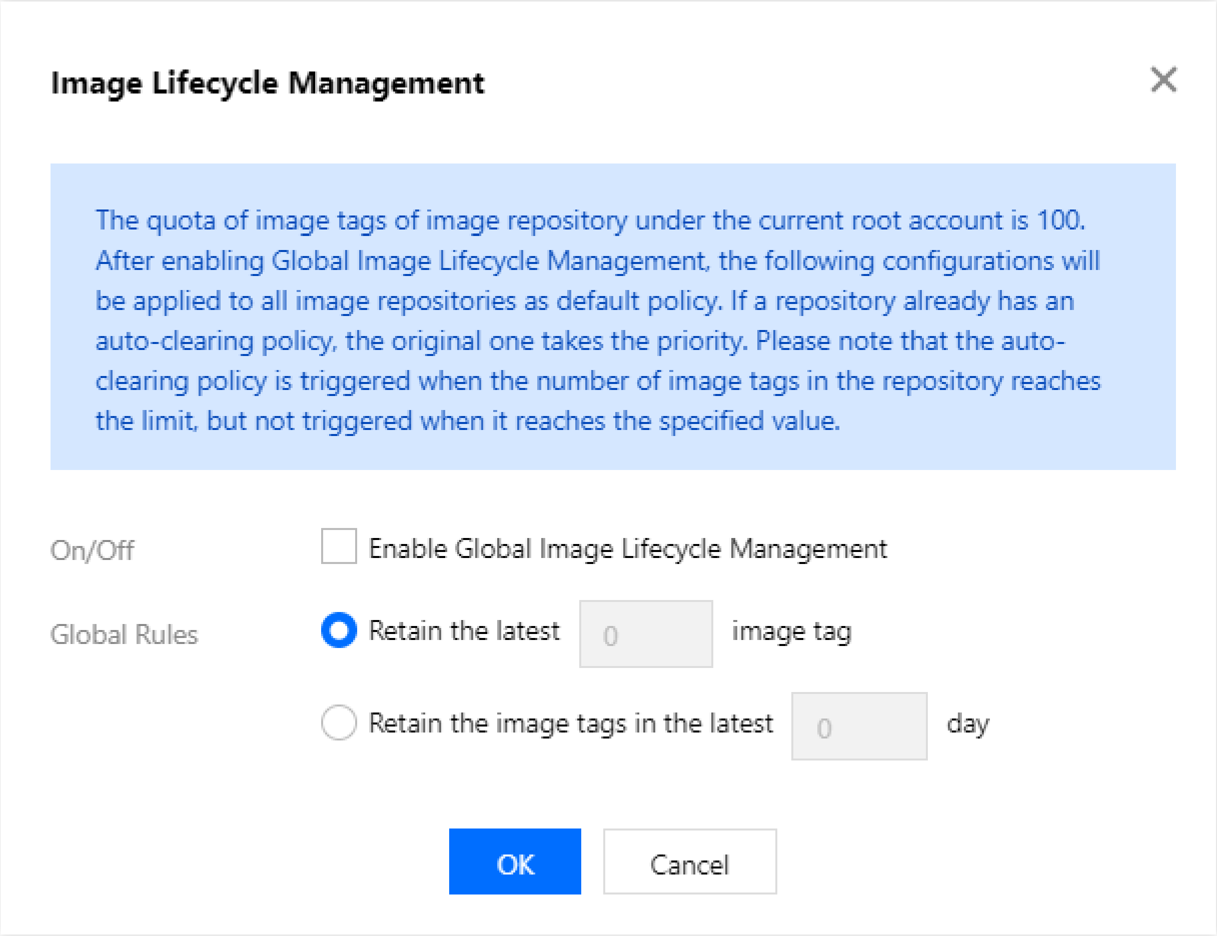
1. Log in to the TKE console, and choose **Image Repositories** -> **My Images** in the left sidebar.
2. On the **My Images** page, click **Image Lifecycle Management**. See the figure below.



[Notice] Tencent Container Registry for Enterprise is now in beta. It offers enterprise-level secure, dedicated, and high-performance container image and Helm Chart hosting and distribution service. Learn More

□	Name	Type	Namespace	Image Address	Creation Time	Operation
□	[redacted]	Private	[redacted]	[redacted]	2020-04-28 15:27:19	Delete Build Config

3. In the **Image Lifecycle Management** window that appears, configure the related parameters. See the figure below.



The quota of image tags of image repository under the current root account is 100. After enabling Global Image Lifecycle Management, the following configurations will be applied to all image repositories as default policy. If a repository already has an auto-clearing policy, the original one takes the priority. Please note that the auto-clearing policy is triggered when the number of image tags in the repository reaches the limit, but not triggered when it reaches the specified value.

On/Off □ Enable Global Image Lifecycle Management

Global Rules ● Retain the latest 0 image tag

○ Retain the image tags in the latest 0 day

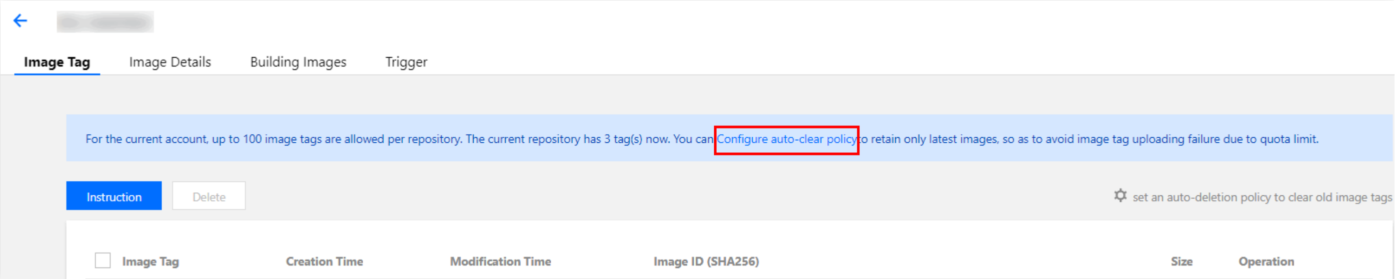
- **Enable Status**: select **Enable image lifecycle management**.
- **Global Rule**:
  - **Reserve the latest XX image tags**: enter a number based on your actual needs. The number cannot exceed the default quota of image tags under the current root account.
  - **Reserve image tags in the last XX days**: enter a number based on your actual needs.

4. Click **OK**.

Configuring an image tag automatic clearing policy for a specified image repository

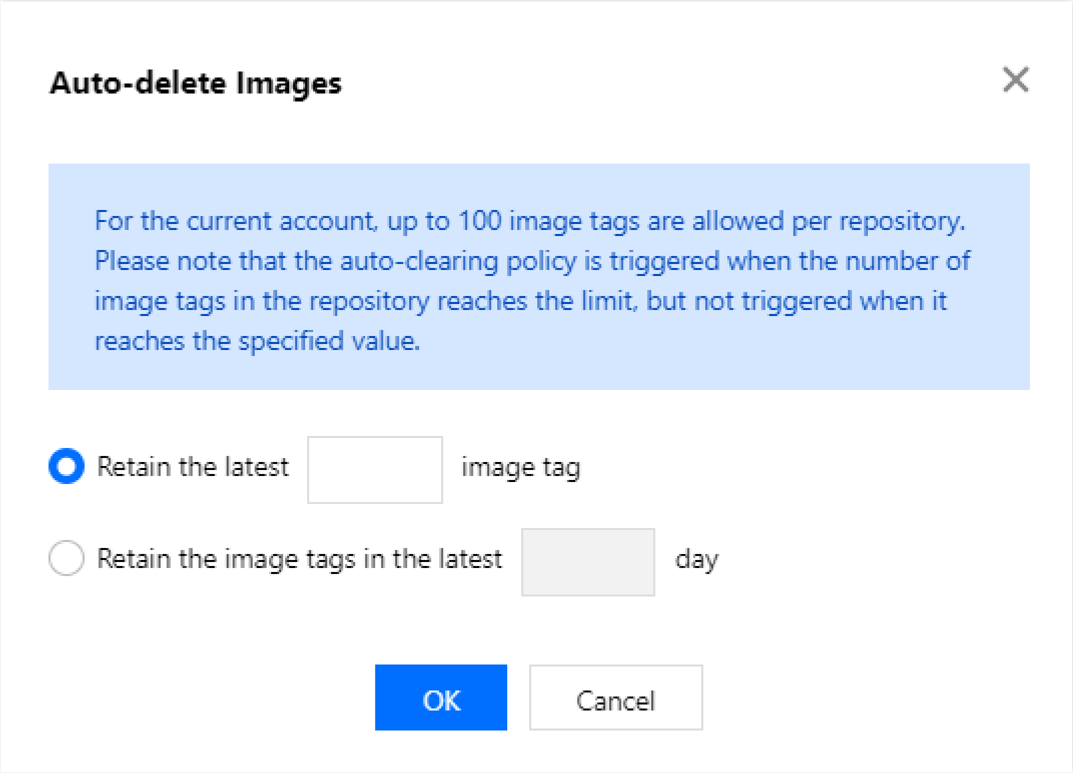
1. Log in to the TKE console, and choose **Image Repositories** -> **My Images** in the left sidebar.
2. On the **My Images** list, click the name of the target image to go to the details page of this image.

3. Click **set an automatic clearing policy** in the prompt above the image tag list. See the figure below.



For the current account, up to 100 image tags are allowed per repository. The current repository has 3 tag(s) now. You can Configure auto-clear policyto retain only latest images, so as to avoid image tag uploading failure due to quota limit.

4. In the **Setting Automatic Clearing Policy** window that appears, configure the related parameters. See the figure below.



Auto-delete Images

For the current account, up to 100 image tags are allowed per repository. Please note that the auto-clearing policy is triggered when the number of image tags in the repository reaches the limit, but not triggered when it reaches the specified value.

● Retain the latest ______ image tag

○ Retain the image tags in the latest ______ day

- **Reserve the latest XX image tags**: enter a number based on your actual needs. The number cannot exceed the default quota of image tags under the current root account.
- **Reserve image tags in the last XX days**: enter a number based on your actual needs.

5. Click **OK**.

Building Images

Image Building Overview

Last updated：2021-03-18 11:22:17

Note：

To provide more stable and powerful code building, image building, container image services, TCR and CODING DevOps have now realized the interconnection of container DevOps related features. Please go to TCR console and CODING DevOps to use them. Creating image building and trigger in Personal Edition are not supported. The existing configurations will be retained and be effective.

Overview

TKE offers continuous integration for containers, enabling users to build container images automatically and manually.

Auto building

Container images can be automatically built based on GitHub or GitLab code repository that **contains Dockerfile files**. You need to register the token of the Github/Gitlab server first. If the code repository is GitLab, **the GitLab server used for the code repository must be accessible through internet**. You can configure an auto building rule for a specific code repository. When you push code to the code repository, if the auto building rule is matched, an container image is automatically built on the TKE Platform and then automatically pushed to the Tencent Cloud TKE image repository.
Configuring auto-building of images:

- Step 1: authorize the source code repository
- Step 2: configure building rules
- Step 3: submit code for auto building

Manual building

You can manually build an image in two ways:

- **Based on GitHub and GitLab code repositories**
  Similar to auto building, the code repository also needs to contain Dockerfile files. If the code repository is in GitLab, GitLab must support internet access. Different from auto building rules in which a container image is automatically built when code is pushed to the repository, manual building requires you to build an image on the console by clicking **Build**.

- **Based on an uploaded Dockerfile**
  On the image repository console, you can upload a Dockerfile based on which the TKE builds a container image.

Build description

- For Dockerfile that is in a Git repository or manually uploaded, if Dockerfile depends on external resource, the external resource must also be accessible through the internet.
- Both manual building and auto building are performed on the TKE platform, so you do not need to provide a build environment or server resources.

Source Code Repository Authorization

Last updated：2020-12-15 12:21:50

Overview

Before using a Git repository to build container images, you need to authorize TKE to access the code source. Currently, TKE supports GitHub repository and GitLab repository.

Prerequisites

You have logged in to the TKE console.

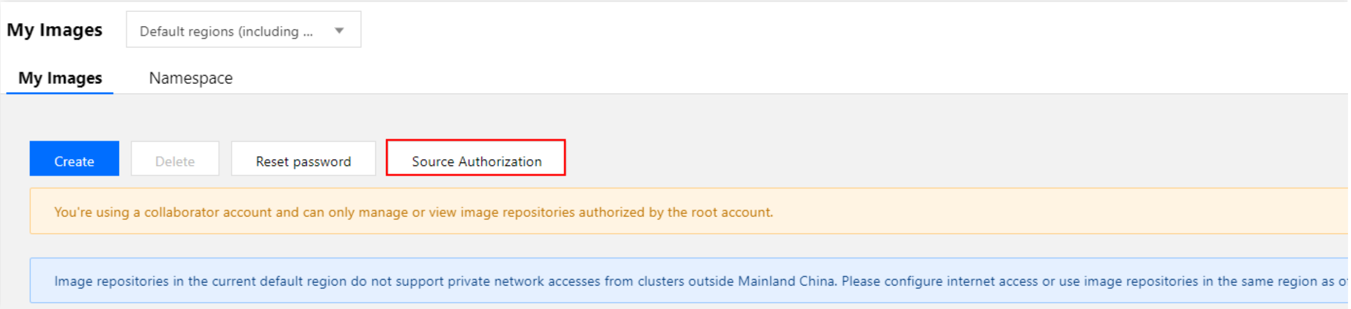
Directions

Selecting a code source

1. Choose **Image Repository** > **My Images** in the left sidebar.
2. On the **My Images** page, click **Source Authorization**, as shown in the following figure.

Note:

“Source Authorization” is only available in China mainland, namely the “Default Region” on the console.



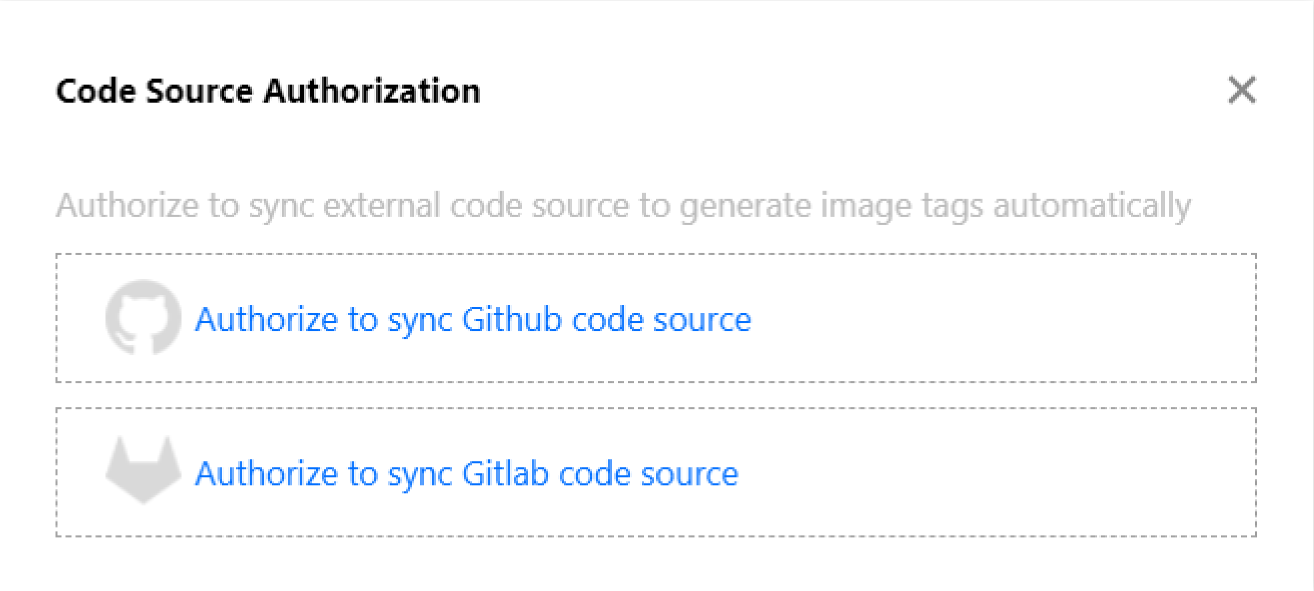
You're using a collaborator account and can only manage or view image repositories authorized by the root account.

Image repositories in the current default region do not support private network accesses from clusters outside Mainland China. Please configure internet access or use image repositories in the same region as o

3. In the **Code Source Authorization** window that appears, select an option as needed, as shown in the following figure.

Note:

A user can authorize both GitHub and GitLab accounts at the same time, but only one account of each type. To change your bound GitHub or GitLab account, you need to unbind the original account.



Code Source Authorization

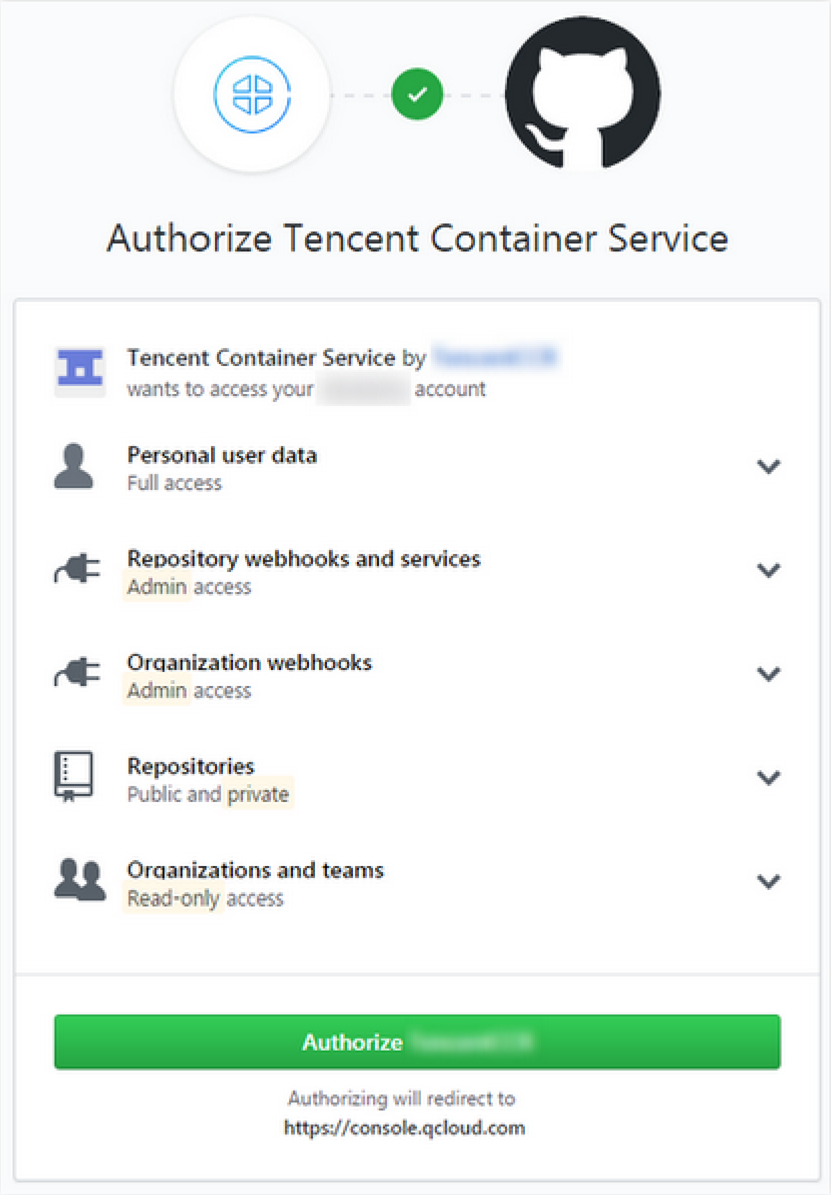
Authorize to sync external code source to generate image tags automatically

×
Authorize to sync Github code source
Authorize to sync Gitlab code source

If your repository is on GitHub, see Authorizing GitHub to complete the authorization.
If your repository is on an on-premises GitLab server or a GitLab managed server, see Authorizing GitLab to complete the authorization.

Github authorization

1. In the **Code Source Authorization** window, click **Authorize to sync Github code source**.
2. For the first time of authorization, you will be directed to the GitHub website and see a prompt stating that the app needs to access your repositories and personal data, as shown in the following figure.



Authorize Tencent Container Service

Tencent Container Service by [redacted] wants to access your [redacted] account

Personal user data
Full access

Repository webhooks and services
Admin access

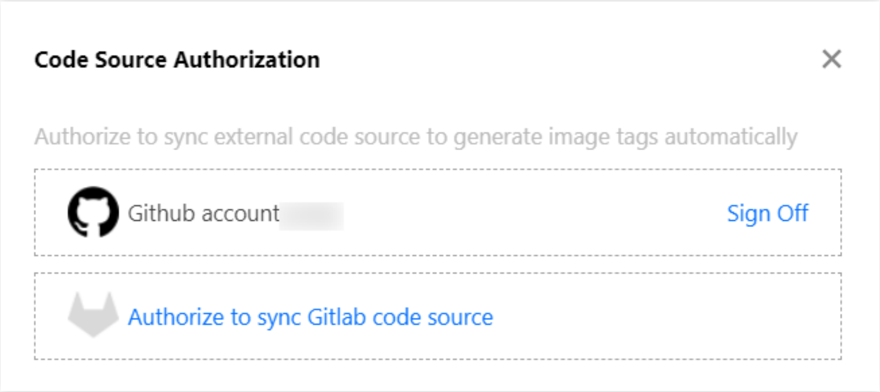
Organization webhooks
Admin access

Repositories
Public and private

Organizations and teams
Read-only access

Authorizing will redirect to
https://console.qcloud.com

3. Click **Authorize** to finish authorizing a GitHub repository, as shown in the following figure.



Code Source Authorization

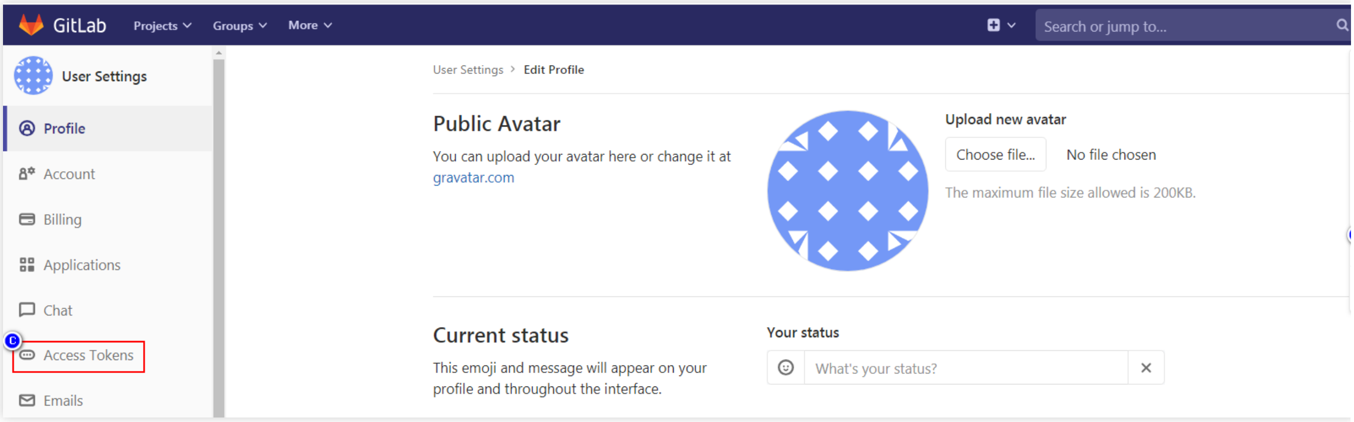
Authorize to sync external code source to generate image tags automatically

Github account [redacted]

Gitlab authorization

Obtaining the Access Token of GitLab

1. Log in to the GitLab website and go to the **Create Access Token** page, as shown in the following figure.



Public Avatar

You can upload your avatar here or change it at gravatar.com

Upload new avatar

The maximum file size allowed is 200KB.

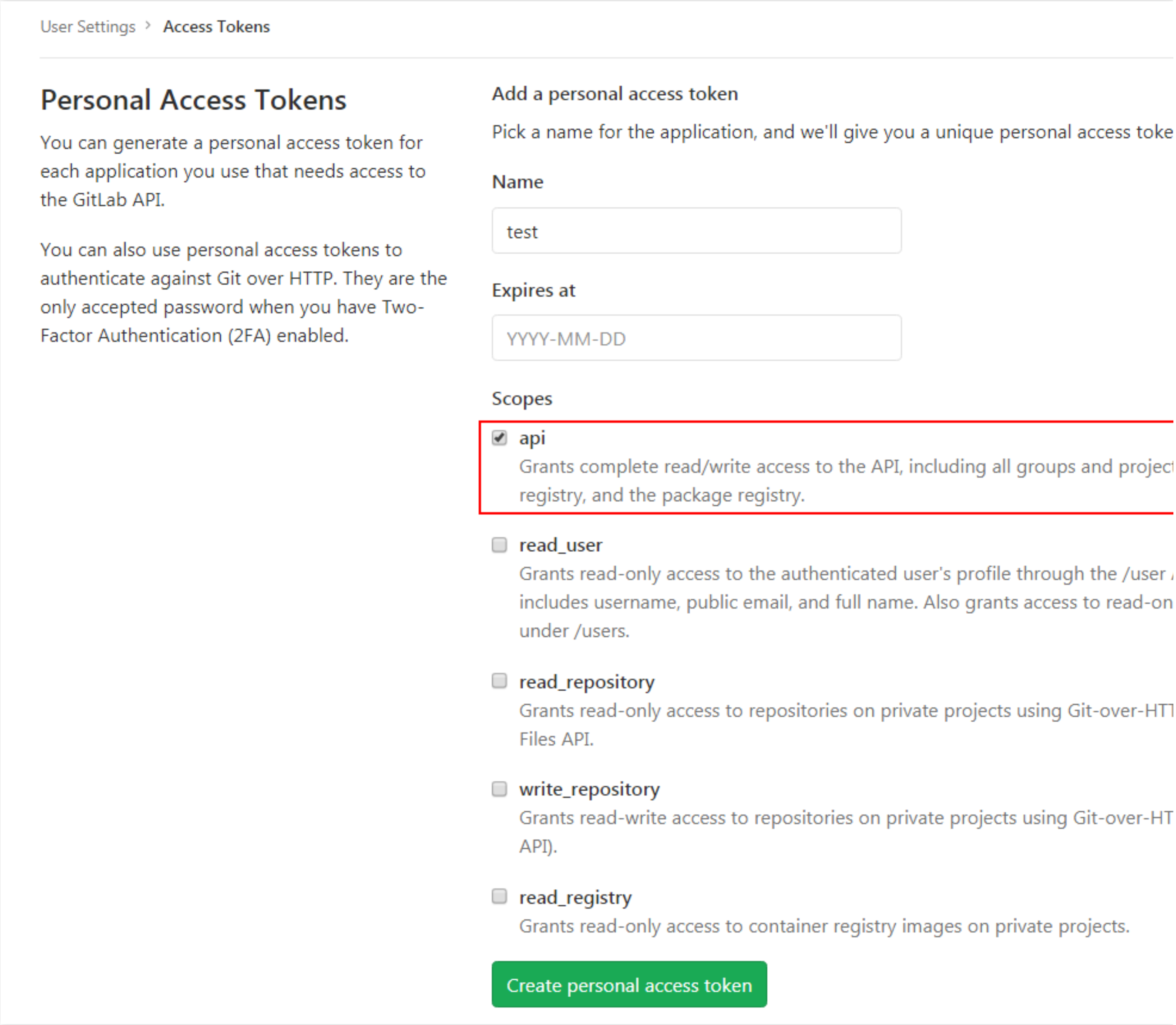
Current status

This emoji and message will appear on your profile and throughout the interface.

2. Click your profile photo. In the drop-down menu, select **Settings**.

3. In the left pane of the "User Settings" page, click **Access Tokens** to go to the **Create Access Token** page.

On the "Create Access Token" page, set the following information as needed:



User Settings > Access Tokens

Personal Access Tokens

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

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

Add a personal access token

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

Name

test

Expires at

YYYY-MM-DD

Scopes

☑ api
Grants complete read/write access to the API, including all groups and project registry, and the package registry.

□ read_user
Grants read-only access to the authenticated user's profile through the /user includes username, public email, and full name. Also grants access to read-on under /users.

□ read_repository
Grants read-only access to repositories on private projects using Git-over-HT Files API.

□ write_repository
Grants read-write access to repositories on private projects using Git-over-HT API).

□ read_registry
Grants read-only access to container registry images on private projects.

Name: enter a name for the new access token.

Expires at: set the expiration date of the new access token.

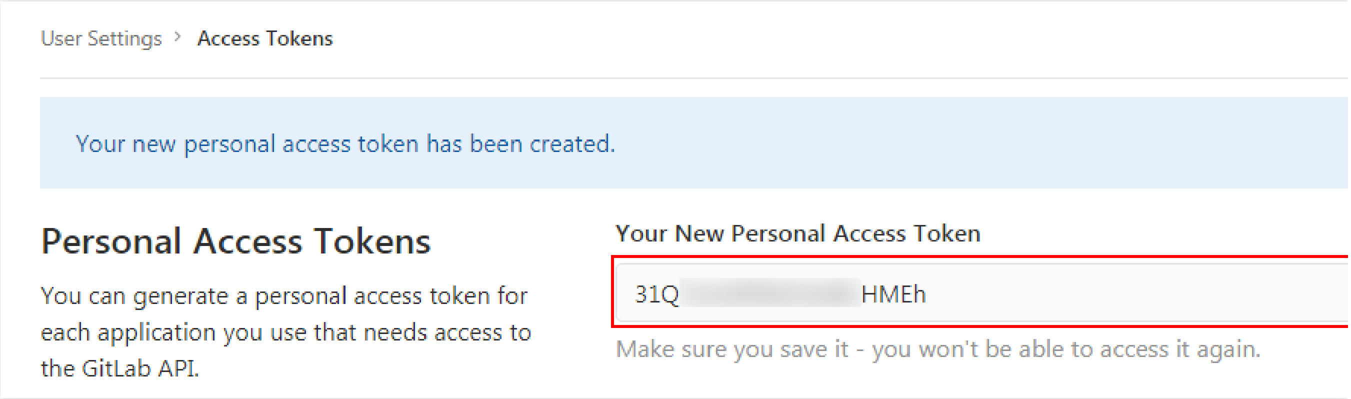
Scopes: set the access scope of the new access token. **The API option is required**.

Note:

The API option is required for Scopes. Otherwise, you will not be able to obtain source repository information or set callback hooks for auto building.

Set a reasonable expiration date for the token to ensure that it is always valid during usage.

4. Click **Create** and save the created access token, as shown in the following figure.



User Settings › Access Tokens

Your new personal access token has been created.

Personal Access Tokens

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

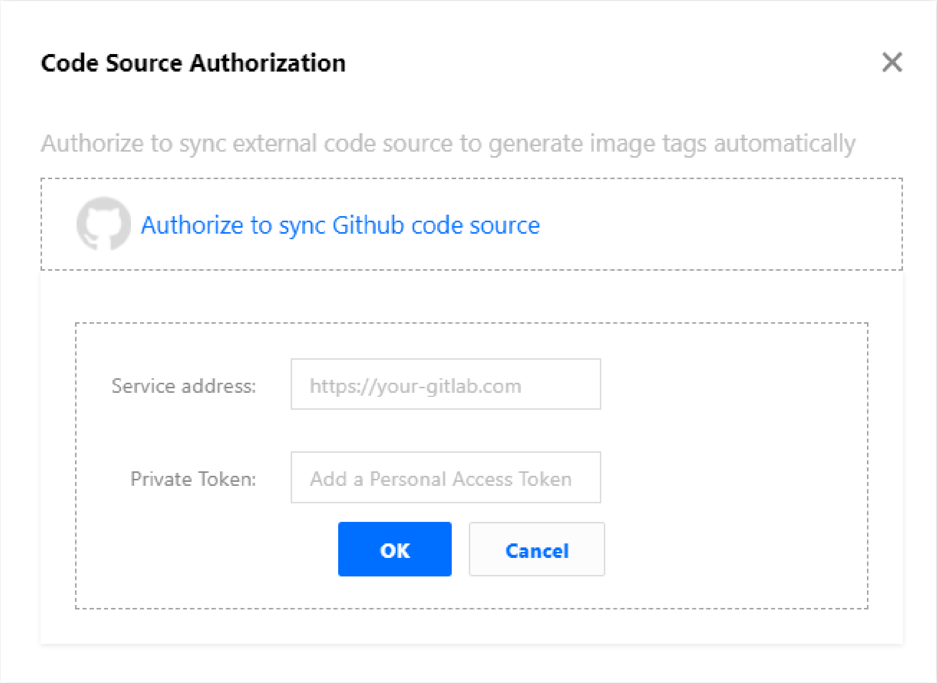
Your New Personal Access Token

31Q[redacted]HMEh

Make sure you save it - you won't be able to access it again.

Authorizing GitLab code source synchronization

1. In the **Code Source Authorization** window, click **Authorize to sync GitLab code source** and enter the following information:



Code Source Authorization

Authorize to sync external code source to generate image tags automatically

Authorize to sync Github code source

Service address: https://your-gitlab.com

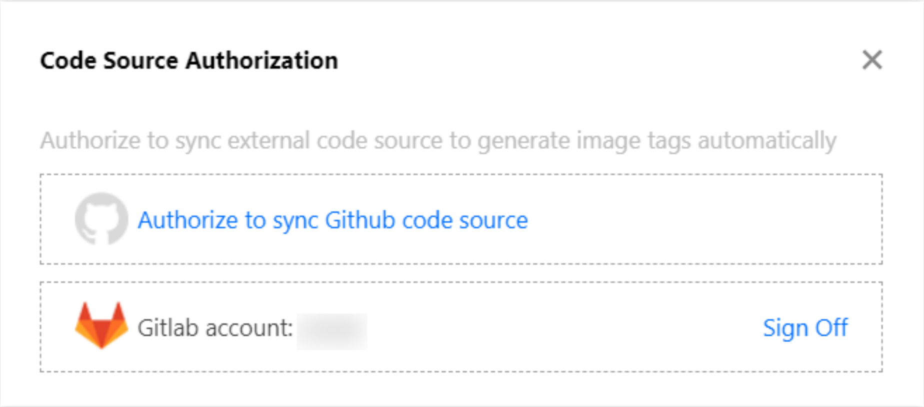
Private Token: Add a Personal Access Token

OK Cancel

Service address: indicates the URL of the GitLab server, which must use the HTTP or HTTPS protocol and can be accessed over the public network. For example, `https://you-gitlab.com` . Do not enter the URL of a specific project or repository.

Private token: must be the Access Token. If you do not have the Access Token yet, see Obtaining a GitLab access token and create one.

2. Click **OK** to finish the authorization, as shown in the following figure.



Code Source Authorization

Authorize to sync external code source to generate image tags automatically

Authorize to sync Github code source
Gitlab account: [redacted]
Sign Off

Build Rules Configuration

Last updated：2020-12-15 12:28:40

Overview

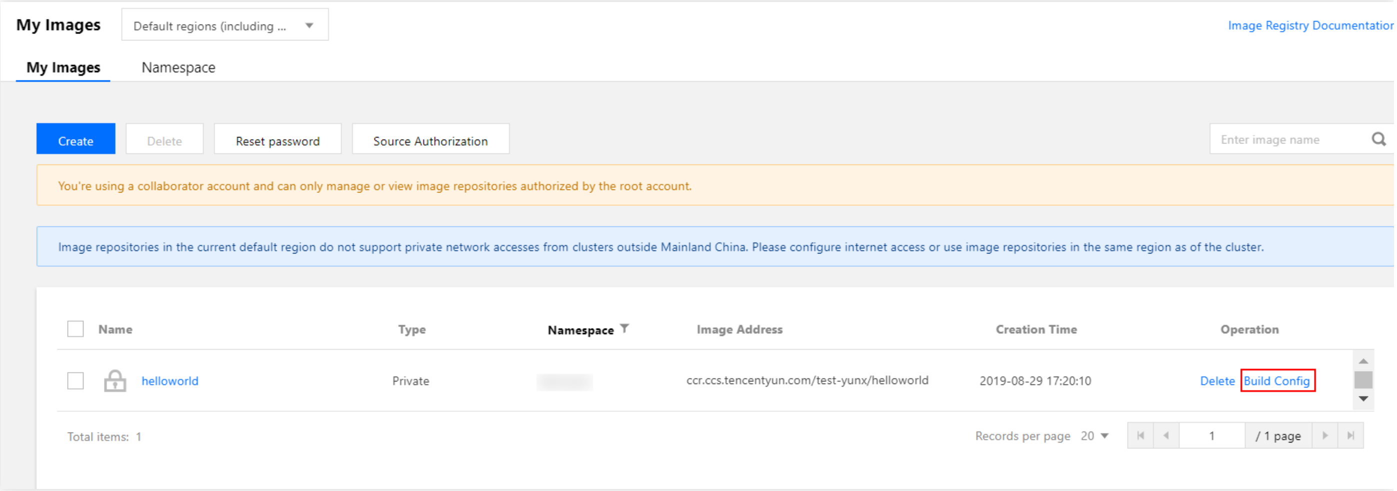
This document describes how to complete auto-building of Github-based images by in TKE console.

Prerequisites

- You have logged in to the TKE console.
- You’ve obtained the access to the source code. For details, see Source Code Repository Authorization.

Directions

1. Select **Image Repository** > **My Images** in the left sidebar to go to the **My Images** page.
2. On the **My Images** page, click **Build Configuration** to the right of the row where the image is located. This is shown in the following figure:



You're using a collaborator account and can only manage or view image repositories authorized by the root account.

Image repositories in the current default region do not support private network accesses from clusters outside Mainland China. Please configure internet access or use image repositories in the same region as of the cluster.

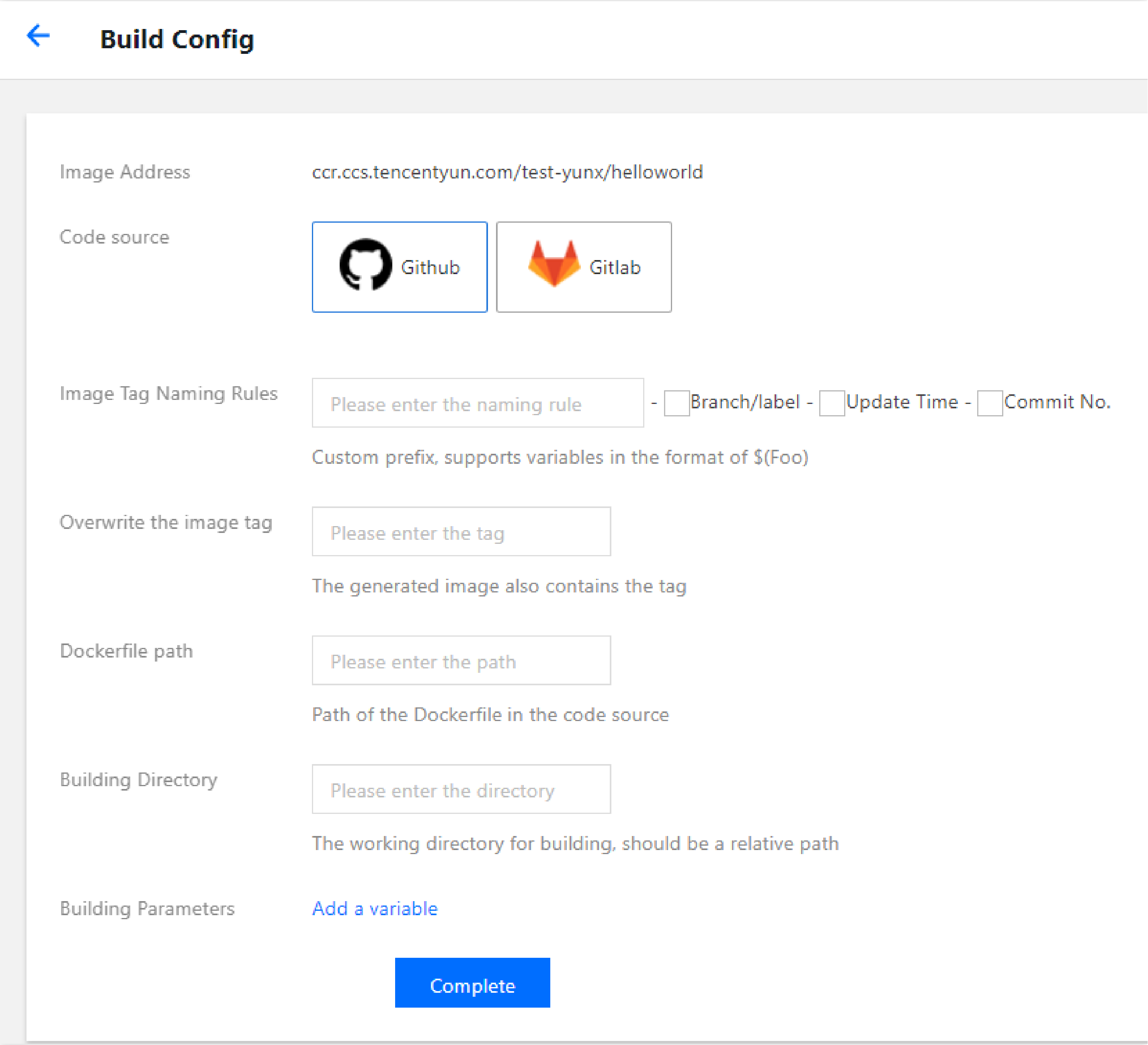
□	Name	Type	Namespace	Image Address	Creation Time	Operation
□	helloworld	Private	[redacted]	ccr.ccs.tencentyun.com/test-yunx/helloworld	2019-08-29 17:20:10	Delete Build Config

Total items: 1 Records per page 20 1 / 1 page

3. Go to the building rule page, and refer to the following steps to configure the related information.

Building configurations for Github-based images

Configure Github build rules according to the following information, and click **Complete**. This is shown in the following figure:



Build Config

Image Address ccr.ccs.tencentyun.com/test-yunx/helloworld

Code source Github Gitlab

Image Tag Naming Rules Please enter the naming rule - □Branch/label - □Update Time - □Commit No.

Custom prefix, supports variables in the format of $(Foo)

Overwrite the image tag Please enter the tag

The generated image also contains the tag

Dockerfile path Please enter the path

Path of the Dockerfile in the code source

Building Directory Please enter the directory

The working directory for building, should be a relative path

Building Parameters Add a variable

Complete

- **Code source**: select **Github**.
- **Organization**: select your organization (usually your Github account), or choose one of the organizations you belong to.
- **Repository**: select a registry through which you need to build a container image.
- **Triggering method**: specifies a condition to build a container image automatically when the code is pushed to a branch or a new Tag. You can also leave it blank and build an image manually on the image building page.
- **Version naming rule**: the container image **Tag naming rule**. An image tag can be formatted.
  - **Branch/tag**: it can contain branch name/repository Tag name.
  - **Update time**: the time when the image is built
  - **commit number**: the latest commit number of the branch/Tag.

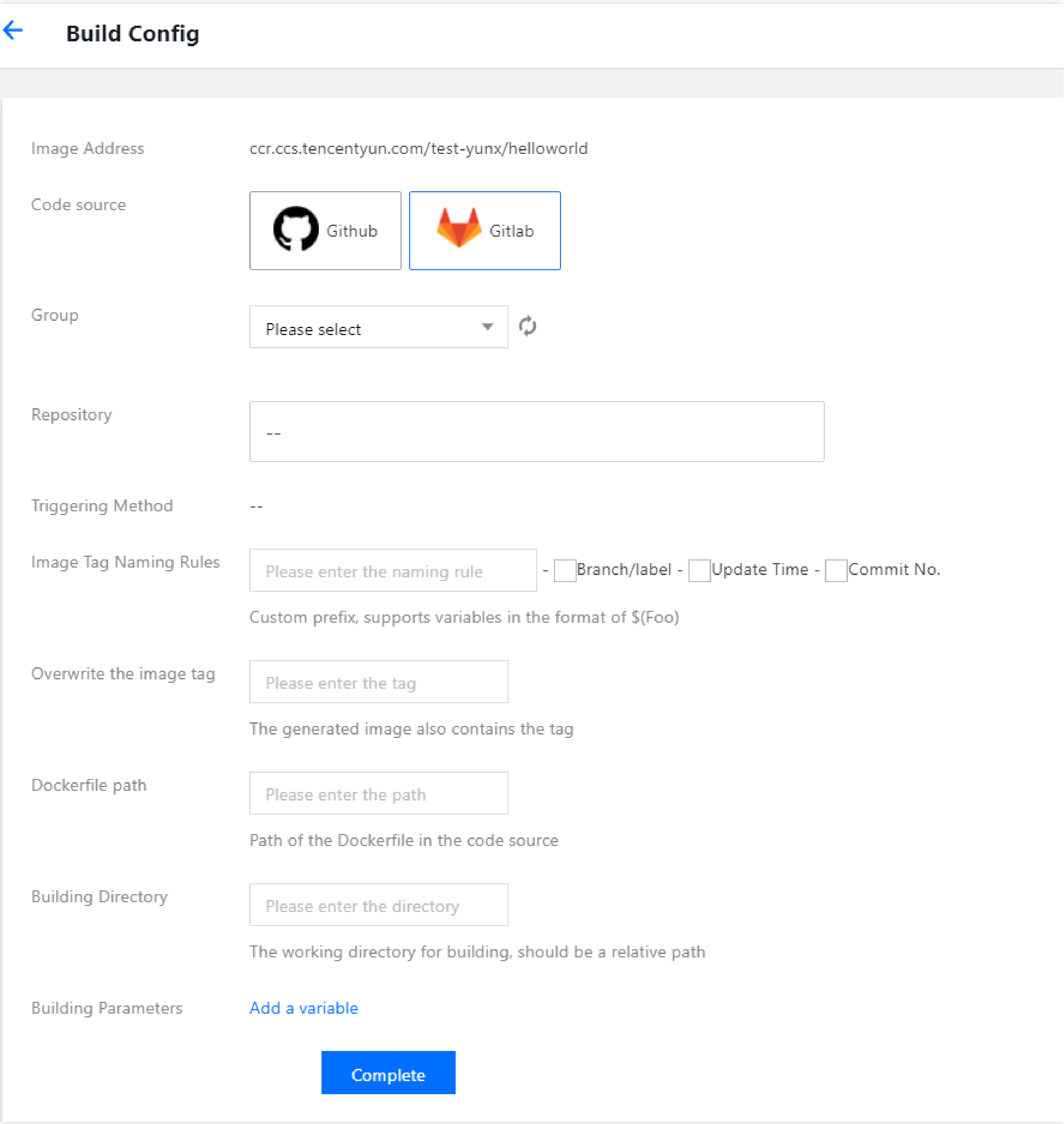
⚠ **Note：**

If an image is automatically built based on a branch or a Tag, and the naming rule contains branch/Tag, the branch or Tag name should be a combination of uppercase/lowercase letters, underscores (`_`), and dashes (`-`), and cannot contain special characters as `/` , `$` , etc.

- **Overwrite the image tag**: after building, an image with the same tag name will be generated, and will overwrite the existing image of the same name.
- **Dockerfile path**: The **Relative Path** of Dockerfile in the registry. It is left empty by default. If it is not specified, Dockerfile is located in the project's root directory. The file name must be "Dockerfile" starting with a capital “D”. If Dockerfile is located in another directory (e.g. the build directory in the registry) and the file name is Dockerfile, the Dockerfile path is: `build/Dockerfile` .
- **Building directory**: the working directory for the build. Dockerfile commands will be executed in this directory.
- **Building parameters**: the parameters that are passed in when the image is built. These can be used to set environment variables.

Building Gitlab-based images

Complete configurations for building Gitlab-based images as instructed below:



Build Config

Image Address ccr.ccs.tencentyun.com/test-yunx/helloworld

Code source Github Gitlab

Group Please select

Repository --

Triggering Method --

Image Tag Naming Rules Please enter the naming rule - □Branch/label - □Update Time - □Commit No.

Custom prefix, supports variables in the format of $(Foo)

Overwrite the image tag Please enter the tag

The generated image also contains the tag

Dockerfile path Please enter the path

Path of the Dockerfile in the code source

Building Directory Please enter the directory

The working directory for building, should be a relative path

Building Parameters Add a variable

- **Code source**: select **Gitlab**.
- **Group**: select a Gitlab Group.
- **Repository**: select a registry through which you need to build a container image.
- **Triggering method**: specifies a condition to build a container image automatically when the code is pushed to a branch or a new Tag. You can also leave it blank and build an image manually on the image building page.

- **Version naming rule**: the container image **Tag naming rule**. An image tag can be formatted.
  - **Branch/tag**: can contain branch/repository Tag name.
  - **Update time**: the time when the image is built
  - **commit number**: the latest commit number of the branch/Tag.

> ⚠ **Note：**
>
> If an image is automatically built based on a branch or a Tag, and the naming rule contains branch/Tag, the branch or Tag name should be a combination of uppercase/lowercase letters, underscores (`_`), and dashes (`-`), and cannot contain special characters as `/` , `$` , etc.

- **Overwrite the image tag**: after building, an image with the same tag name will be generated, and will overwrite the existing image of the same name.
- **Dockerfile path**: The **Relative Path** of Dockerfile in the registry. It is left empty by default. If it is not specified, Dockerfile is located in the project's root directory. The file name must be "Dockerfile" starting with a capital “D”. If Dockerfile is located in another directory (e.g. the build directory in the registry) and the file name is Dockerfile, the Dockerfile path is: `build/Dockerfile` .
- **Building directory**: the working directory for the build. Dockerfile commands will be executed in this directory.
- **Building parameters**: the parameters that are passed in when the image is built. These can be used to set environment variables.

Auto building

Select a triggering method in the configuration, and a container image is automatically built when you submit a new branch or push the code to the specified registry. The entire building process is implemented on the Tencent Cloud TKE platform. After you finish building, a new image is generated according to the version naming rule you defined, and uploaded to the Tencent Cloud TKE image registry.

FAQs

For more information on the source code build Dockerfile file path issue, see Image Build FAQs.

Trigger

Trigger Overview

Last updated：2022-05-09 12:41:24

Note
The entries of TCR Enterprise and TCR Individual in the console are merged. To provide you with more powerful and stable code compilation, image building and image deployment services, the image building of TCR Individual is now supported by CODING DevOps service. You can select the desired image repository in the new console to configure images. You can configure webhook through “Operation Center - Trigger” to realize connection with the external release system. The image building and trigger features in the legacy console are only provided for existing users, and configurations cannot be added.

An image repository trigger can automatically push images. It does not support connecting with external webhook servers.

An image repository trigger contains the following four attributes:

- Trigger name: It indicates the name of the trigger.
- Image repository: It specifies an image repository to be bound with the trigger. Currently, an image repository can be bound with up to 10 triggers.
- Trigger condition: It specifies that a trigger action is performed only if an image that has a specific tag is submitted.
- Trigger action: It only supports triggering updates of TKE.

Triggering Conditions

Currently, Tencent Cloud TKE Image Registry supports three types of tag trigger expressions, which can be used to configure trigger conditions:

- All: The action is triggered whenever a tag is created or updated in the image repository.
- Specified tags: Enter multiple tags and separate them with semicolons (;). The action is triggered when images with the specified tags are created or updated in the image repository.
- Regular expression: The action is triggered when a matching tag is created or updated in the image repository.

Trigger Action

Currently, service update is the only supported trigger action. The configuration of the trigger action includes setting the region, cluster, namespace, service, container image, and other parameters of TKE.
When a trigger condition is met, the specified container image of the service is updated based on the set parameters.

Triggering History

Every time a repository trigger performs an action, a triggering history entry is generated, which contains information such as the trigger name, triggering condition, trigger action, trigger result, and triggered time.

Terminating/Returning Instances

Last updated：2023-05-08 15:44:59

Overview

This document introduces how to terminate or return Enterprise Edition instances in Tencent Container Registry (TCR).

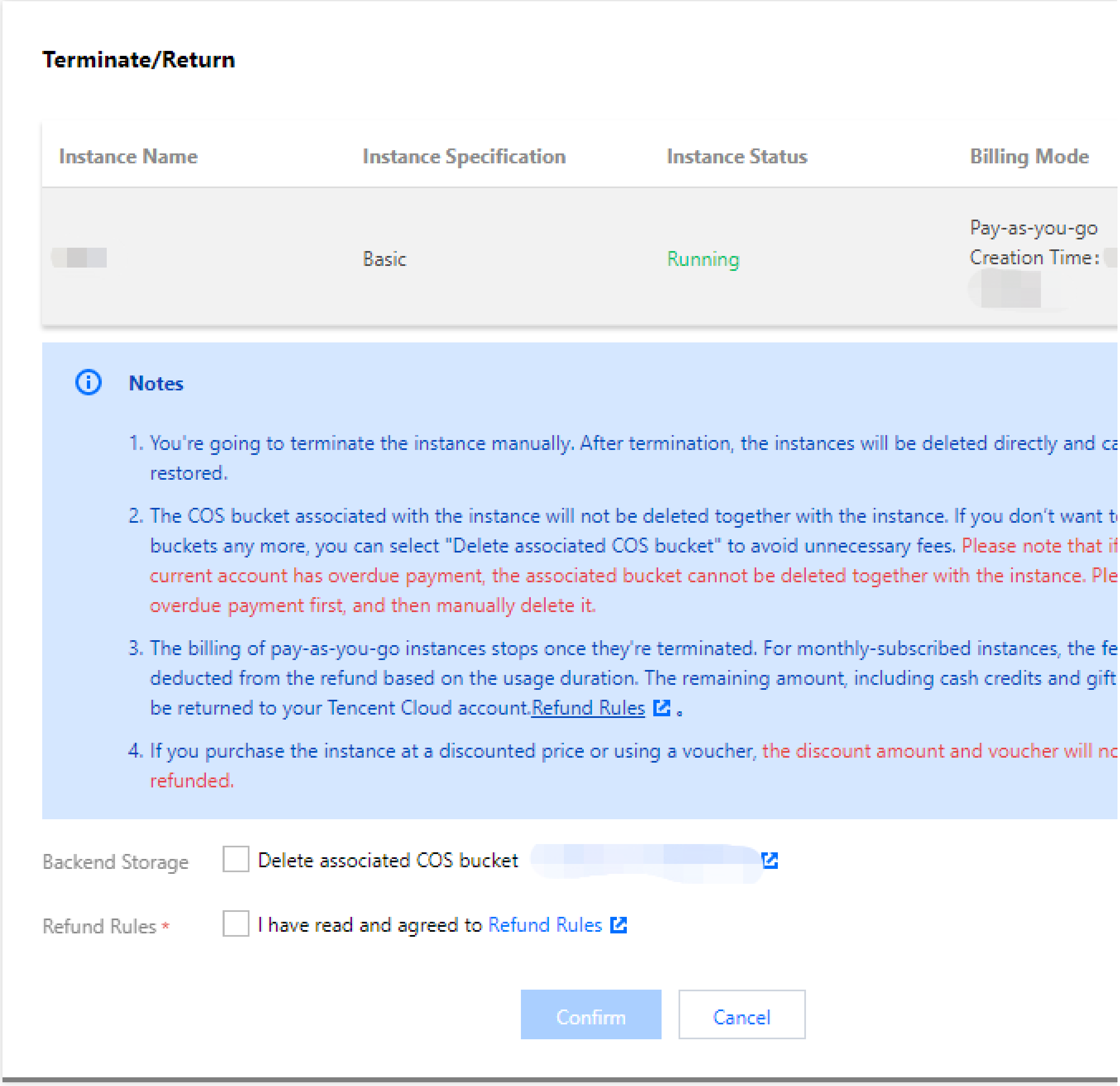
Prerequisites

You have purchased a TCR Enterprise Edition instance, and the current account has the permissions to delete this instance.

Directions

Terminating or returning an instance in the TCR console

1. Log in to the TCR console.
2. Click **Instance management** in the left sidebar to go to the **Instance management** page.
3. Find the instance and choose **More** > **Terminate**/**Return** on the right side.
4. In the confirmation pop-up window, select **Delete associated COS Bucket** based on your needs, as shown in the figure below:



Terminate/Return

Instance Name	Instance Specification	Instance Status	Billing Mode
[redacted]	Basic	Running	Pay-as-you-go Creation Time: [redacted]

Notes

1. You're going to terminate the instance manually. After termination, the instances will be deleted directly and ca restored.
2. The COS bucket associated with the instance will not be deleted together with the instance. If you don't want t buckets any more, you can select "Delete associated COS bucket" to avoid unnecessary fees. Please note that if current account has overdue payment, the associated bucket cannot be deleted together with the instance. Ple overdue payment first, and then manually delete it.
3. The billing of pay-as-you-go instances stops once they're terminated. For monthly-subscribed instances, the fe deducted from the refund based on the usage duration. The remaining amount, including cash credits and gift be returned to your Tencent Cloud account.Refund Rules .
4. If you purchase the instance at a discounted price or using a voucher, the discount amount and voucher will nc refunded.

Backend Storage □ Delete associated COS bucket [redacted]

Refund Rules * □ I have read and agreed to Refund Rules

Notes

Please read the notes for terminating and returning the instance carefully. Deleting an instance will completely and irrecoverably erase the user data and relevant configuration stored during the use of the instance.

If you are sure that you no longer need the container images, Helm charts, and other underlying data stored during the use of TCR Enterprise Edition, you can select **Delete associated COS Bucket** to avoid unnecessary fees.

If the current account has overdue payments, the COS service does not allow you to directly delete the associated COS bucket. In this case, do not select that option. Instead, you need to delete the instance, and then go to the COS console to manage the COS bucket.

The billing of pay-as-you-go instances stops once they are terminated.

5. Select **I have read and agreed to Refund Rules** and click **Confirm** to delete the instance. No fees will be charged for this instance after deletion.

Note

If it takes an unusually long time to delete an instance or the displayed status is **Abnormal**, please submit a ticket.

Terminating or returning an instance by calling an API

You can also call the `DeleteInstance` API to delete an instance. For more information, see DeleteInstance.