

Tencent Container Registry

Quick Start

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

Quick Start

Quick Start

TCR Individual Getting Started

Quick Start

Quick Start

Last updated : 2024-07-01 18:01:40

This document describes how to purchase a Tencent Container Registry (TCR) Enterprise Edition instance, configure a network access policy, and push and pull container images. To use TCR Individual, see [TCR Individual Getting Started](#).

Step 1: Signing Up for a Tencent Cloud Account

[Sign up for a Tencent Cloud account](#) and complete [identity verification](#). If you already have a Tencent Cloud account, ignore this step.

Step 2: Activating TCR Service

Log in to the [Tencent Cloud console](#) and choose **Tencent Cloud services > Tencent Container Registry** to go to the TCR console. Then, activate and authorize TCR as prompted. If you have already authorized TCR, skip this step.

Step 3: Purchasing an Enterprise Edition Instance

1. Log in to the [TCR console](#) and go to the **Instance management** page.
2. Click **Create**. On the **TCR Purchase** page, purchase an instance. You can use the following information for reference:

Tencent Container Registry

Product Details

Product Documentation Billing Instructions Console

Important: TCR is officially recommended, providing enterprise-level secure hosting and processing on specialized services for cloud-native artifacts, and featuring the capability of large-scale disaster image distribution.

Purchase Notes

Overview: TCR provides you with enterprise-level hosting and distribution services for cloud-native artifacts. Fees of service hosting, data storage and access will be generated during use. If you need to use free services, please visit "Tencent Kubernetes Engine - Image Repository".

Billing Mode: For now, TCR charges only pay-as-you-go (prepaid) billing mode. We will support the monthly subscription (postpaid) billing mode soon. By then, you can convert existing instances to monthly subscription.

Billing Rule: TCR only charges hosting service fees. The cloud-native artifacts (such as container images and Helm charts) included in using this service are hosted in your COS Bucket and generate storage and traffic fees based on actual usage. The COS billing method is adopted. You can go to the Billing Center to query billing information.

Price Calculator

Specification: Guangzhou Basic

Configuration Fee: 0.14 USD/hour

☐ I have read and agree to "TCR Service Protocol"

Buy Now

Select Configuration

Instance Name:

The instance name can contain 5-33 characters, including lowercase letters, digits and '-', it cannot start or end with '-', and cannot contain two consecutive dashes.

Instance Region:

Region	Guangzhou	Shenzhen Phoenix	Shanghai	Beijing	Shanghai Phoenix	Beijing	Beijing Phoenix
China	Guangdong	Guangdong	Shanghai	Beijing	Shanghai	Beijing	Beijing
Overseas	Hong Kong (China)	Singapore	Bangkok	Seoul	Silicon Valley	London	Virginia

Instance Specification:

	Basic	Enhanced	Premium
Outdated registry service	✓	✓	✓
Outdated works access domain name	✓	✓	✓
Outdated data storage backend	✓	✓	✓
Supports high-speed network acceleration	✓	✓	✓
Multi-tenant repository strategy	✓	✓	✓
Image client hosting	✓	✓	✓
Push/pull quota	10	100	300
Image repository quota	1000	5000	10000
Push/pull quota	1000	5000	10000
Private instance access control	✓	✓	✓
VPC access control	✓	✓	✓
VPC access quota	5	5	10
Image vulnerability scanning	✓	✓	✓
Operation log retention	7 days	15 days	30 days
Multiple regions replication for single instance (separate server license)	✓	✓	✓
Global instance (cross-region) automatic synchronization	✓	✓	✓
Multi-tenant repository in the same city	✓	✓	✓
Standard Tagging Type	✓	✓	✓
Container image compilation and building	✓	✓	✓
Cloud-native delivery workflow	✓	✓	✓
HTTP accelerated distribution of images	✓	✓	✓

Instance Domain Name:

After creating the instance, please go to access control to specify the VPC and put it in target for access control to ensure.

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 Rules.

Instance Tag:

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 created.** 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 created.** 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. For more information, see [Purchase Guide](#).

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 created.** This domain name is used when you run the `docker login` command to log in to the instance.

Backend Storage: when an instance is created, a Tencent Cloud COS bucket will be automatically created 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 creation, 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.

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 instance creation.

3. 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.

4. After checking the selected option, click **Buy Now** to purchase the enterprise edition instance you have selected and configured.

5. You can check the instance purchase progress on the **Instance List** page. When the instance status changes to **Running**, the instance has been successfully purchased and is available. You can complete the following steps to configure the access control policy of the instance and log in to the instance to push and pull images.

Step 4: Configuring the Network Access Policy

To protect your data security, all public and private network access requests are denied by default after the instance is created. Before you log in to the instance, push, and pull images, you must configure the network access policy.

In the console, select **Access Control** in the left sidebar, select **Private Network Access** or **Public Network Access** as needed, and configure the corresponding access policy.

Private network access (recommended)

Public network access

Note:

Both TCR Individual and TCR Enterprise do not support classic network access. If you need to use this service, we recommend that you switch to VPC as soon as possible and access the service over the private network.

For more information about how to use this service in TKE, see [Using a Container Image in a TCR Enterprise Edition Instance to Create a Workload](#).

We recommend that you push and pull container images through private network access because it can significantly accelerate the push and pull speeds and reduce public network traffic costs. In addition, you can manage private

network access linkages to specify the VPCs that are allowed to access your image data and improve data security. Follow the steps below:

1. In the upper part of the **Private Network Access** page, select the created instance.
2. Click **Create**. In the **Create a private network access linkage** pop-up window, configure the VPC and subnet information. See the figure below:

Create a private network access linkage

Associated Instance intl-demo (Guangzhou)

Virtual Private Cloud doc-te sub-te

If no suitable VPC in the current region, you can [create a new one](#).

You are associating the private network with a TCR instance. After the association, a bridge link is established between the private network and the TCR instance, occupying an IP address of the private network as the access entry. In this way, the cloud server connected to the private network can access the associated instance by accessing the private network IP address. If the private network is connected to other private network or IDC network through cloud networking or Direct Connect, server connected to the associated network can also access the TCR instance resources via the private network. After the private network access link is established, you can use the self-owned DNS, or configure the host on the server, to parse the instance domain name to a specific IP address of the private network. For more information, please see [private network access control](#).

Confirm Cancel

Select the VPC where the container cluster that need to access the image repository is located and select any subnet in this VPC that has usable private IP addresses.

3. After the private network access linkage is successfully established, the parsing of the instance domain name is not configured in the connected VPC by default. Click **Manage Auto-parsing** to enable the parsing for private network domain names. This feature is based on Tencent Cloud Private DNS. Activate this service before proceeding. For more information, see Private Network Access Control.

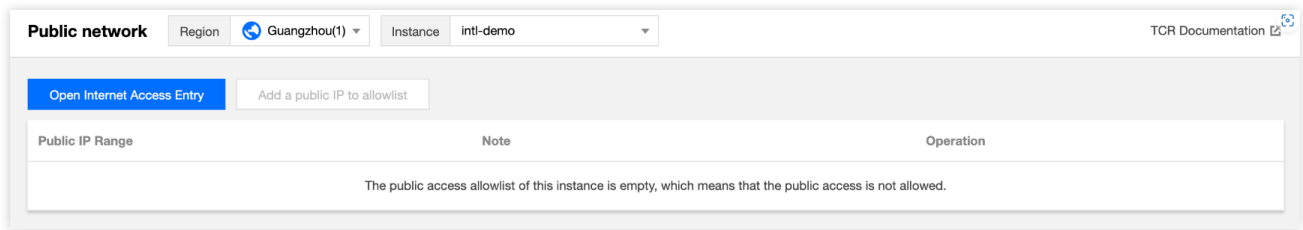
Note:

Enabling the Internet access entry opens your dedicated instance in the public network environment. We recommend that you disable the Internet access entry as soon as possible after completing private network access configuration.

Follow the steps below:

1. In the upper part of the **Public Network Access** page, select the created instance.
2. Click **Open internet access entry** in the upper left corner. The button status changes to **Enabling**, as shown in the figure below:

After Internet access is enabled, the Docker client can access the image repositories through the Internet.



3. When the button status changes from **Enabling** to **Close Internet Access Entry**, Internet access is enabled. Then, click **Add a public IP to allowlist** in the upper left of the list to add the public IP addresses that are allowed to access the image repositories.

4. In the **Create Public Network Access Allowlist** pop-up window, add the public IP addresses or IP ranges that are allowed to access the image repositories, or import the configurations of the existing security groups, and add remarks for this rule (optional), as shown in the figure below.

We recommend that you do not add `0.0.0.0/0`, which allows all Internet access. Alternatively, delete this rule before formally activating the instance.

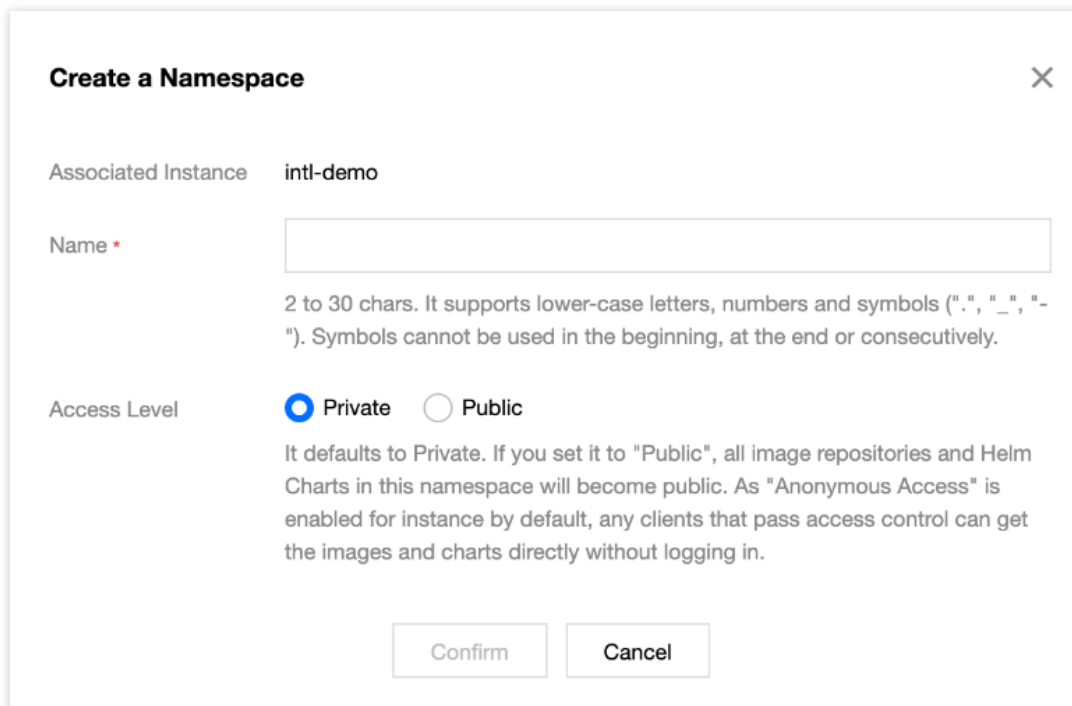
Step 5: Creating a Namespace

1. Select **Namespace** in the left sidebar. On the **Namespace** page, click **Create**.

Note:

Namespaces are used to manage image repositories in the instance. They do not directly store container images, but can map to teams, product projects, or other custom layers in an enterprise.

2. In the **Create a Namespace** pop-up window, configure the namespace information and click **Confirm**, as shown in the figure below.



Create a Namespace ✕

Associated Instance intl-demo

Name *

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.

Name: We recommend that you set this parameter to the name of an enterprise team or product project. Namespace names must be unique in an instance.

Access Level: you can select either **Private** or **Public**. Image repositories and Helm chart repositories in the namespace will inherit this attribute. You can modify this attribute after creating the namespace.

Step 6: (Optional) Creating an Image Repository

Note:

After creating a namespace, you can use the Docker client to push images to the namespace, and the corresponding image repository will be automatically created.

1. Click **Image Repository** in the left sidebar to go to the **Image Repository** list page.
2. Click **Create**. In the **Create an Image Repository** pop-up window, configure the image repository information and click **Confirm**, as shown in the figure below.

From the **Namespace** drop-down list, you can select a created namespace. The name cannot be a multi-level path, and the Markdown syntax is supported in the description.

Create an Image Repository

Associated Instanceintl-demo

Namespace *public

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".

Image sourceLocal

After creating the image repository, you can use Docker client or CI tool to push existing images to this repository.

Brief descriptionnginx repo

Detailed Description# Nginx
Support Markdown

ConfirmCancel

Step 7: Pushing and Pulling an Image

After completing the preceding steps, you have created an instance and image repository. Next, you can perform the following operations to push an image to or pull an image from the image repository.

Note:

In this step, you need to use a CVM or CPM instance with Docker installed and ensure that the target client is in the public or private network access allowlist defined in [Configuring the Network Access Policy](#).

Logging in to the TCR instance

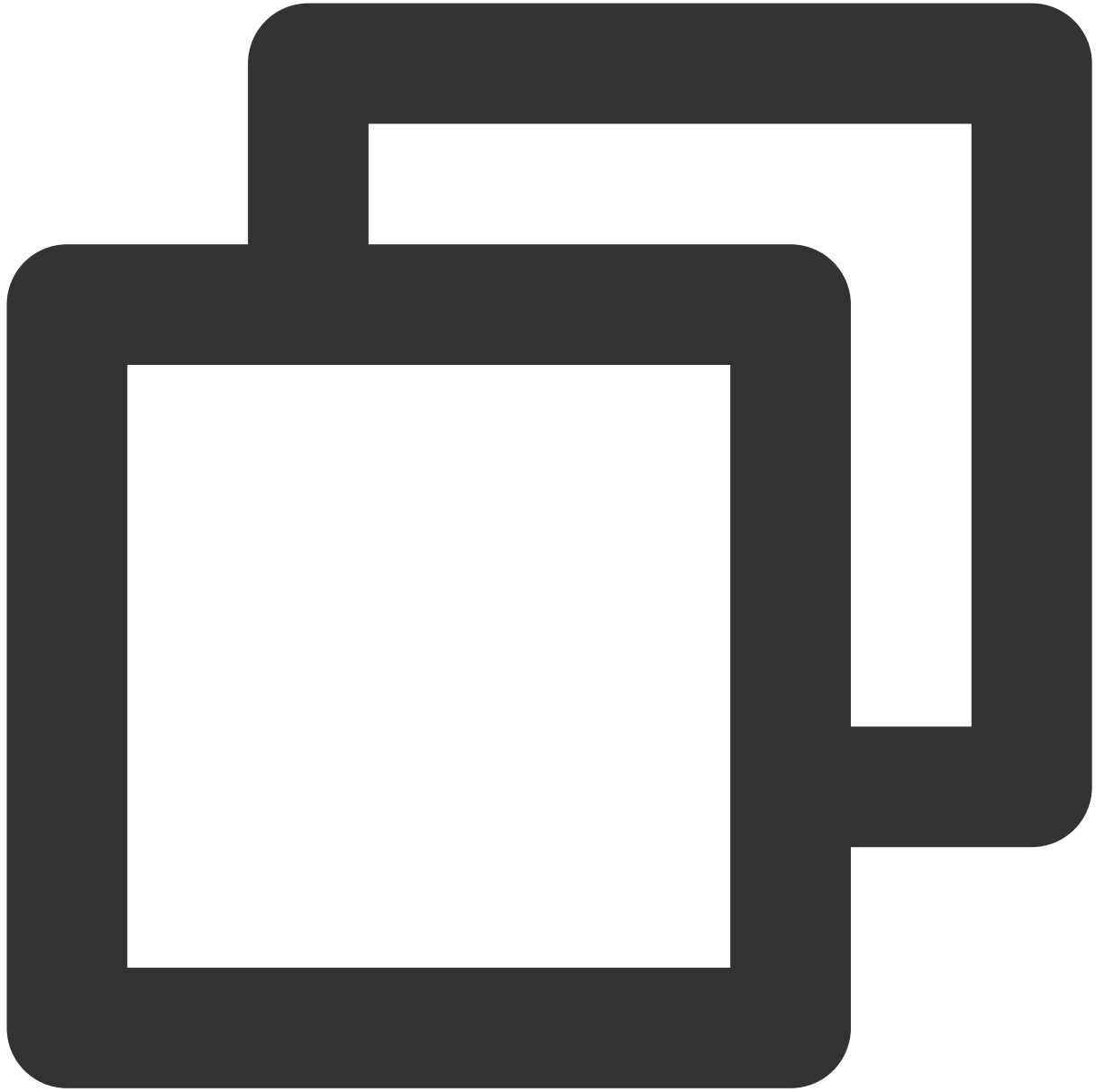
1. Click **Access Credential** in the left sidebar to go to the **Access Credential** list page. Select the newly created instance, and click **Generate Temp Login Token**.

Note:

In this document, a temporary login token for the instance is used as an example. You can also [obtain a long-term access credential](#).

2. In the **Temp login token** pop-up window, click **Copy login token**.

3. In the command-line tool, run the login token that you have obtained to log in to the instance. Sample token:



```
sudo docker login demo-tcr.tencentcloudcr.com --username 1xxx1019xxxx --password ey
```

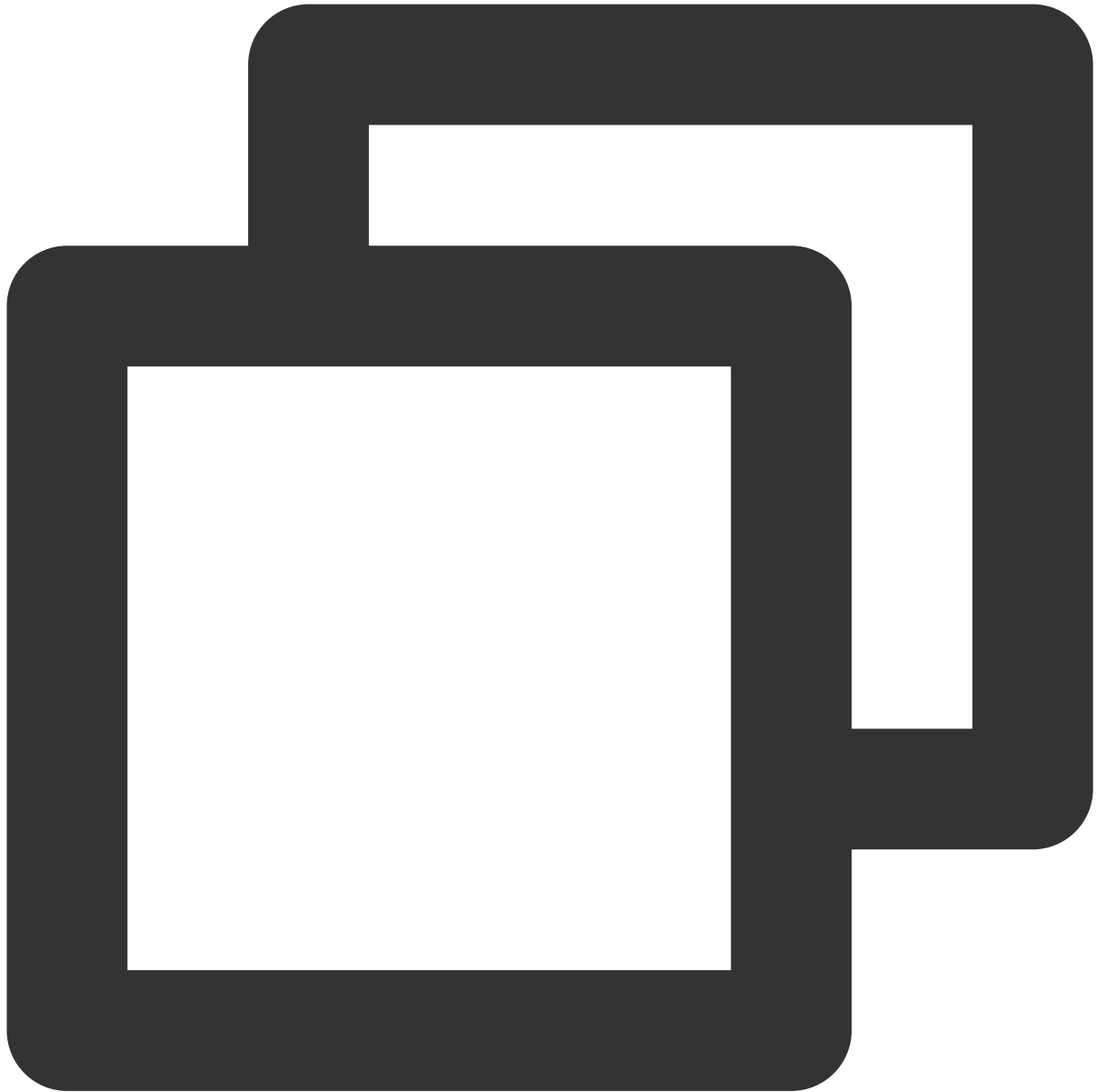
If `Login Succeeded` is displayed in the command line tool, you have logged in to the instance successfully.

Pushing a container image

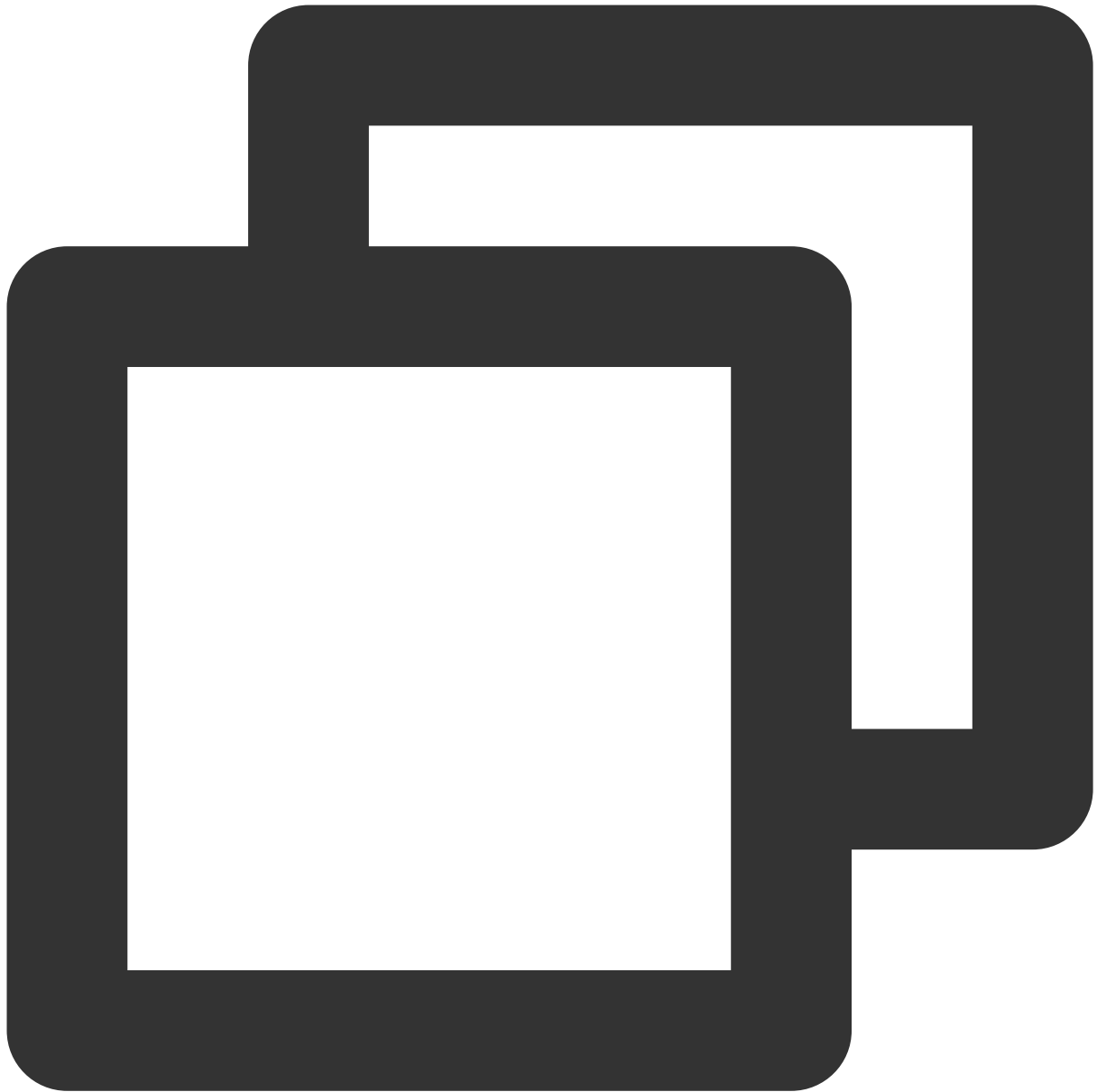
You can create a container image on the local server or obtain a public image from Docker Hub for testing.

This document uses the official and latest Nginx image on Docker Hub as an example. In the command line tool, run

the following commands sequentially to push this image. Replace `demo-tcr` , `project-a` , and `nginx` with the actual instance, namespace, and image repository names you have created.



```
sudo docker tag nginx:latest demo-tcr.tencentcloudcr.com/project-a/nginx:latest
```



```
sudo docker push demo-tcr.tencentcloudcr.com/project-a/nginx:latest
```

Pulling a container image

This document uses the successfully pushed Nginx image as an example. In the command line tool, run the following command to pull this image:



```
sudo docker pull demo-tcr.tencentcloudcr.com/project-a/nginx:latest
```

References

TCR Enterprise Edition provides advanced features such as Helm chart hosting, cross-region instance synchronization, and image security scanning. To use them, refer to the following documents:

[Managing Helm Charts](#)

[Cross-Tenant Synchronization](#)[Managing Triggers](#)[Network Access Control Overview](#)[Access Permission Configuration](#)

What if a problem occurs when I use TCR?

If you encounter a problem while using TCR, locate and solve the problem by referring to the [FAQs](#). Alternatively, you can [submit a ticket](#), and we will solve the problem for you as soon as possible.

TCR Individual Getting Started

Last updated : 2024-07-01 18:03:13

This document describes how to initialize a Tencent Container Registry (TCR) Individual Edition instance, configure a namespace, and push and pull container images. To use TCR Enterprise Edition, see [Quick Start](#).

Step 1: Signing Up for a Tencent Cloud Account

[Sign up for a Tencent Cloud account](#) and complete [identity verification](#). If you already have a Tencent Cloud account, ignore this step.

Step 2: Activating TCR Service

Log in to the [Tencent Cloud console](#) and choose **Tencent Cloud services > Tencent Container Registry** to go to the TCR console. Then, activate and authorize TCR as prompted. If you have already authorized TCR, skip this step.

Step 3: Initializing TCR Individual Service

1. Log in to the [TCR console](#) and go to the **Instance management** page.
2. Select the region where you want to use the service. Currently, TCR Individual service is deployed only in Guangzhou in the Chinese mainland and supports cross-region access over the private network from regions such as Beijing, Shanghai, and Chengdu. For more information about other supported regions, see [Purchase Guide](#). The actual available regions are subject to the region list in the console. This document takes the TCR Individual Edition instance of the selected region as an example.
3. Check the tab of the TCR Individual Edition instance in the region, and click **Initialize Password** to set the password for accessing TCR Individual service. You can choose **More > Reset the login password** to reset the password.
4. Click **Log In to Instance** to obtain the guidance on login of TCR Individual.



```
docker login ccr.ccs.tencentyun.com --username=xxxxxxxxx
```

"username" is the current Tencent Cloud account ID.

Run this login command in the command line tool, and enter the password. The login is successful if `Login Succeeded` is displayed.

Step 4: Creating a Namespace

1. Click **Namespace** in the left sidebar. On the **Namespace** page, select **TCR Individual Instance** and click **Create**.

Note:

Namespaces are used to manage image repositories in the instance. They do not directly store container images, but can map to teams, product projects, or other custom layers in an enterprise.

2. In the **Create a Namespace** pop-up window, configure the namespace information and click **Confirm**, as shown in the figure below:

Name: We recommend that you use an enterprise team or project name. A TCR Individual Edition instance is a shared instance. You cannot create a namespace with the name that has been used by another user.

Step 5: (Optional) Creating an Image Repository

Note:

After creating a namespace, you can use the Docker client to push images to the namespace, and the corresponding image repository will be automatically created.

1. Click **Image Repository** in the left sidebar to go to the **Image Repository** list page. Select **TCR Individual Instance** at the top of the page.

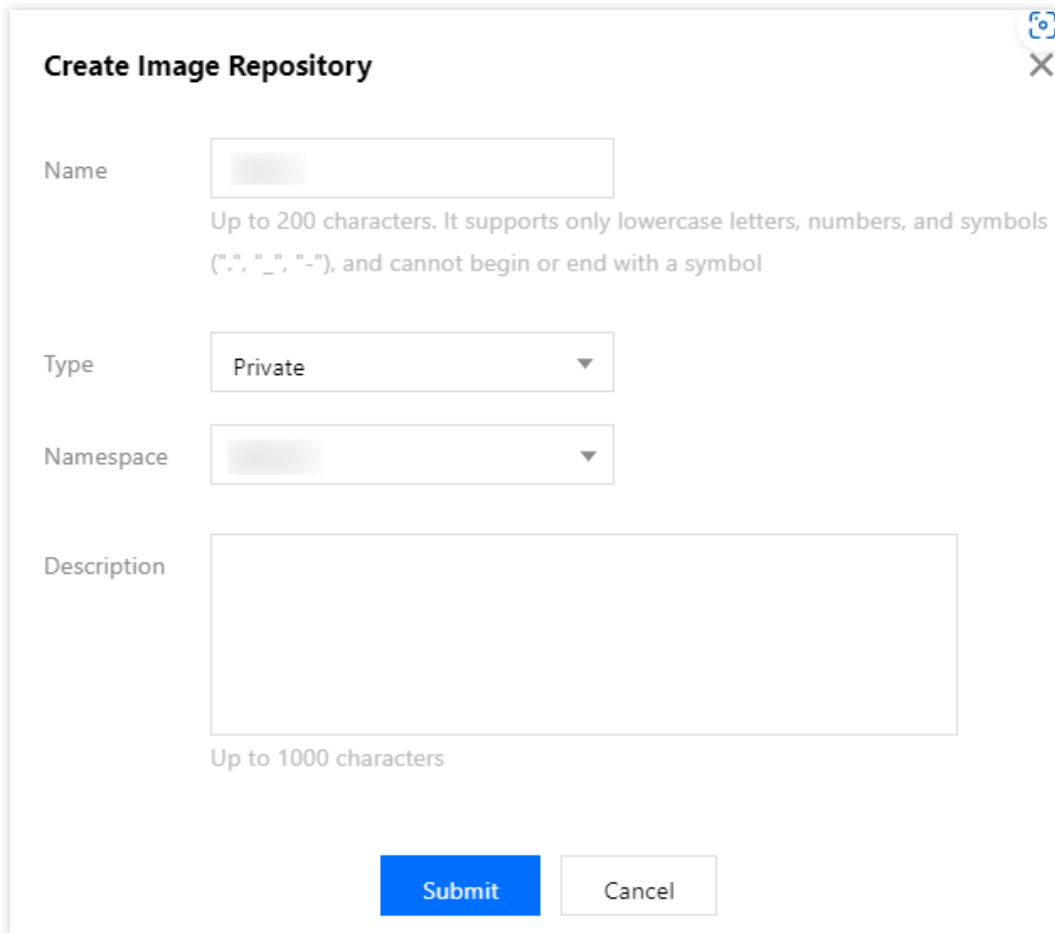
2. Click **Create**. In the **Create an Image Repository** pop-up window, configure the image repository information and click **Confirm**, as shown in the figure below.

2.1 Name: The name must be 200 characters in length and can contain only lowercase letters, digits, and any of the following separators: periods (.), underscores (_), and hyphens (-). The name cannot start or end with a separator, and cannot be a multi-level path.

2.2 Type: Public or private. Like the image types in DockerHub, a public image is visible to all external users and can be pulled anonymously, and a private image is visible only to users with permissions and can be pulled only after login.

2.3 Namespace: Select a created namespace.

2.4 Description: The Markdown syntax is supported.



The image shows a 'Create Image Repository' dialog box. It has a title bar with a close button (X) and a maximize button. The form contains the following fields:

- Name:** A text input field. Below it, a note states: 'Up to 200 characters. It supports only lowercase letters, numbers, and symbols (".", "_", "-"), and cannot begin or end with a symbol'.
- Type:** A dropdown menu currently showing 'Private'.
- Namespace:** A dropdown menu with a blurred selection.
- Description:** A large text area. Below it, a note states: 'Up to 1000 characters'.

At the bottom right, there are two buttons: a blue 'Submit' button and a white 'Cancel' button with a grey border.

Step 6: Pushing and Pulling an Image

After completing the preceding steps, you have created a namespace and image repository. Next, you can perform the following operations to push an image to or pull an image from the image repository.

Note:

You need to use a CVM or CPM instance with Docker installed.

Pushing a container image

You can create a container image on the local server or obtain a public image from Docker Hub for testing.

This document uses the official and latest Nginx image on Docker Hub as an example. In the command line tool, run the following commands sequentially to push this image. Replace `project-a` and `nginx` with the actual namespace and image repository names you have created.



```
sudo docker tag nginx:latest ccr.ccs.tencentyun.com/project-a/nginx:latest
```



```
sudo docker push ccr.ccs.tencentyun.com/project-a/nginx:latest
```

Pulling a container image

This document uses the successfully pushed Nginx image as an example. In the command line tool, run the following command to pull this image:



```
sudo docker pull ccr.ccs.tencentyun.com/project-a/nginx:latest
```

What if a problem occurs when I use TCR?

If you encounter a problem while using TCR, locate and solve the problem by referring to the [FAQs](#). Alternatively, you can [submit a ticket](#), and we will solve the problem for you as soon as possible.