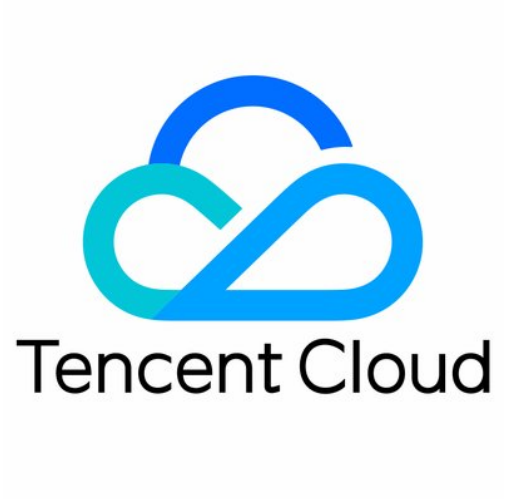
Tencent Container Registry

Operation Guide

Product Documentation



Contents

Operation Guide
Create an enterprise instance

Last updated : 2020-05-29 16:00:27

Scenario

This document describes how to create a TCR Enterprise Edition instance in Tencent Container Registry (TCR).

Prerequisites

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

- Sign up for a Tencent Cloud account, and complete identity verification.
- Submit an application, and obtain the qualification for the TCR beta test.
- Activate the TCR-dependent cloud product, Cloud Object Storage (COS).
- If you need to access the instance through a Virtual Private Network (VPC), activate the VPC service.
- Activate the TCR service on the console and grant certain operation permissions to your COS and VPC resources.

Procedure

1. Log in to the Tencent Cloud console and choose **Products** -> **Compute** -> **Tencent Container Registry** to go to the TCR console.

2. Select **Instance List** in the left sidebar to go to the "Instance List" page, and click **Create**.

3. In the "Create Instance" window, configure the following information to create an instance. See the figure below.

   - **Instance Name**: enter a custom instance name. This document uses `demo-tcr` as an example. The name is globally unique and cannot be identical with an existing instance name of your own or other users. This name is used to access the domain name of this TCR instance. **The name**

cannot be modified after creation. 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 changed after the instance is created.**. If possible, select a region where your main business is located. More regions to choose from will be gradually be made available. If you have special requirements, submit a ticket.
- **Instance Specification**: select the instance specifications you wish to purchase. Different instance specifications have different instance performance and quotas and apply to different application scenarios and business scales. You can purchase only a **Standard** edition instance in the beta phase.
- **Instance Domain Name**: the instance domain name that is automatically generated. Its prefix is the same as that of 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.

4. Click **OK** to create an instance. This process takes about 1 minute.
   You can check the instance creation progress on the "Instance List" page. If the instance status changes to "Running" as shown in the figure below, the instance was successfully created and is running properly.

> If it takes too long to create an instance or the displayed status is abnormal, submit a ticket.

Manage namespaces

Last updated : 2020-07-30 14:18:19

Scenario

In Tencent Container Registry (TCR), 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 tasks:

- Create an Enterprise Edition instance.
- If you are using a sub-account, grant the sub-account operation permissions for the corresponding instance in advance. For more information, see Examples of Enterprise Edition Authorization Schemes.

Procedure

Creating a namespace

1. Log in to the TCR console and select **Namespace** in 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 Name** 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 in the figure below.
   - **Associated Instance**: currently selected instance, to which the created namespace belongs.
   - **Name**: name of the namespace. 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.
   - **Access Level**: you can select either "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 **Confirm** to create the namespace.
   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. See the figure below.

Changing the access level

In the "Access Level" area of a specified namespace, click  in front of "Public" or "Private" to change the public or private attribute of the namespace.

> 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.**

-  : the access level of the namespace is private.
-  : the access level of the namespace is public.

Changing the security scan mode

Click  under "Security Scan" of a specified namespace to change the security scan mode for container images in this namespace. You can set the security scan mode to **Manual** or **Automatic**.

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

-  : the security scan mode is set to **Manual**. To perform a security scan on a specified container image and view the result, go to the "Image Repository" page, select this image, and click **Scan** on the **Tag Management** tab.
-  : the security scan mode is set to **Automatic**. An automatic security scan is triggered when a new image is pushed to any image repository in the current namespace.

Deleting a namespace

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

Manage Image Repository
Manage Image Registry

Last updated : 2020-08-03 15:29:43

Scenario

In Tencent Container Registry (TCR), 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 attributes and security scan triggering mode from its namespace.

The image repository is the minimum unit for permission management in TCR. The instance admin can grant the management or read-only permission to a sub-user. 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 on other permission management and authorization methods, see Examples of TCR Enterprise Edition Authorization Schemes. This document describes how to create and manage an image repository in a TCR Enterprise Edition instance.

Prerequisites

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

- You have created a TCR Enterprise Edition instance.
- If you are using a sub-account, ensure that you have granted the sub-account operation permissions for the corresponding instance in advance. For more information on how to grant the permissions, see Examples of TCR Enterprise Edition Authorization Schemes.

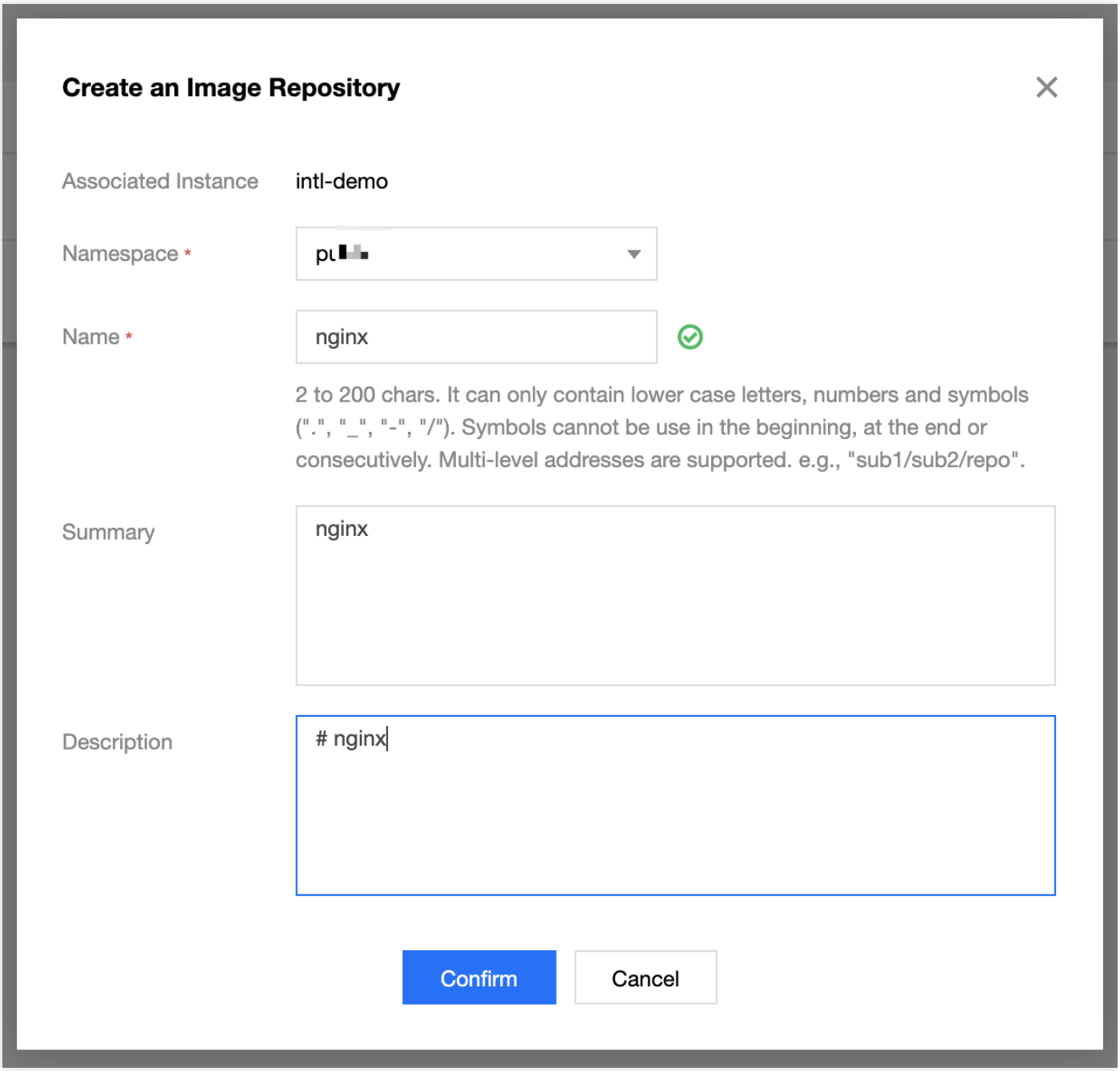
Directions

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, at the top of the page, select the desired instance name from the "Instance Name" drop-down list.

2. Click **Create**. In the "Create an Image Repository" window that appears, configure the image repository by referring to the following field description.



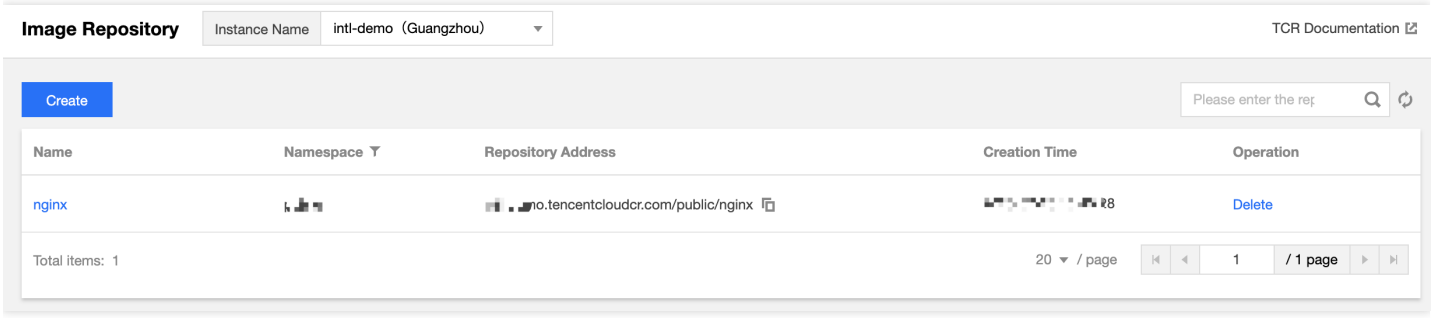
- **Associated Instance**: indicates the currently selected instance, to which the created image repository belongs.
- **Namespace**: indicates the namespace to which the image repository belongs. If the list is empty, first create a namespace in the instance.

- **Name**: indicates the name of the image repository. Its value must be 2 to 200 characters in length and can only contain lowercase letters, numbers, and separators including periods (`.`), underscores (`_`), hyphens (`-`), and slashes (`/`). This parameter cannot start or end with a separator or contain several consecutive separators. In addition, this name can be a cascaded path, such as `team-01/front/nginx` . You can set the name based on your business requirements.
- **Image source**: supports "Local image push" and "Platform image building".
- **Summary**: indicates the brief description of the image repository. Its value is a string of up to 100 characters. You can edit the summary again after the image repository is created.
- **Description**: indicates the detailed description of the image repository. This parameter supports the Markdown syntax. Its value is a string of up to 1,000 characters. You can modify the description after the image repository is created.

3. Click **OK** to create the image repository.

Image repository operations

After the image repository is created, you can view the image repository on the "Image Repository" page. Then, you can perform the following operations to manage the image repository, as shown in the following figure:



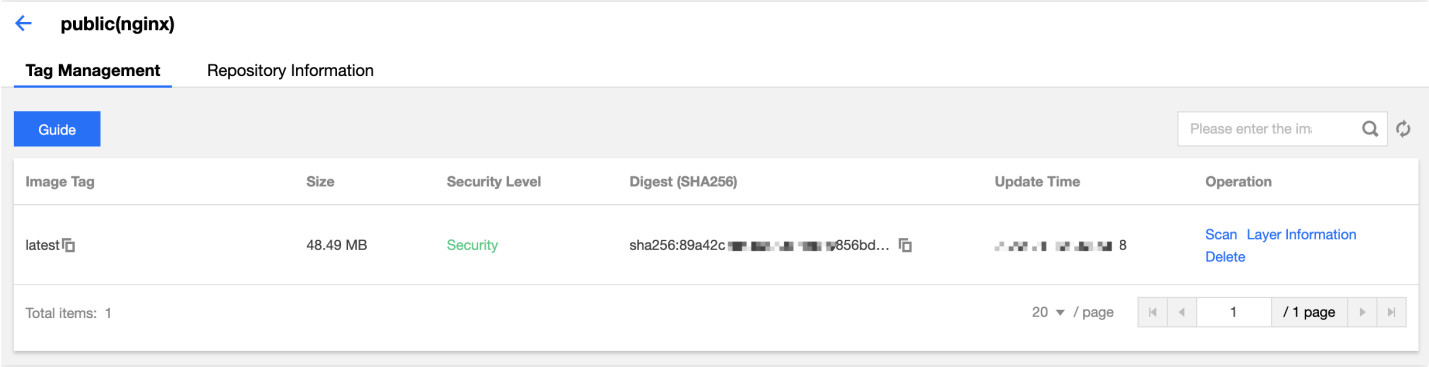
- **Filtering namespaces**
  Select 命名空间 from the "Image Repository" list for filtering. Then, you can select a namespace that you want to view from the drop-down list.
- **Viewing details of a repository**
  Click the name of a specified image repository. The repository details page appears, where you can manage the image tag and edit the basic information of the image repository.
- **Deleting an image repository**
  You can click **Delete** for an image repository to delete it. Exercise caution before confirming the deletion to prevent important data from being deleted by mistake.

> ⚠ **Note :**

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

Managing image tags

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



- **Filtering image tags**
  In the search box in the upper-right corner of the tag list, you can enter an image tag to search for this tag. Fuzzy search is supported.
- **Obtaining a pull command**
  You can click **Pull Command** for the target image tag to copy the pull command of the image tag.
- **Performing a security scan**
  You can click **Scan** for the target image tag to proactively trigger a security scan. When the scanning result is output for the corresponding "Security Level", click to view the detailed result.
- **Viewing the image layer information**
  You can click **Layer Information** for a target image repository to view the layer information for this image in a pop-up window.
- **Deleting an image tag**
  You can click **Delete** for a target image tag to delete this image tag. Excercise caution before confirming the deletion to prevent important data from being deleted by mistake.

> ⚠ **Note :**
>
> When a specified image tag is deleted, other image tags with the same image ID as the deleted image tag may also be deleted. If this is the case, these image tags will become

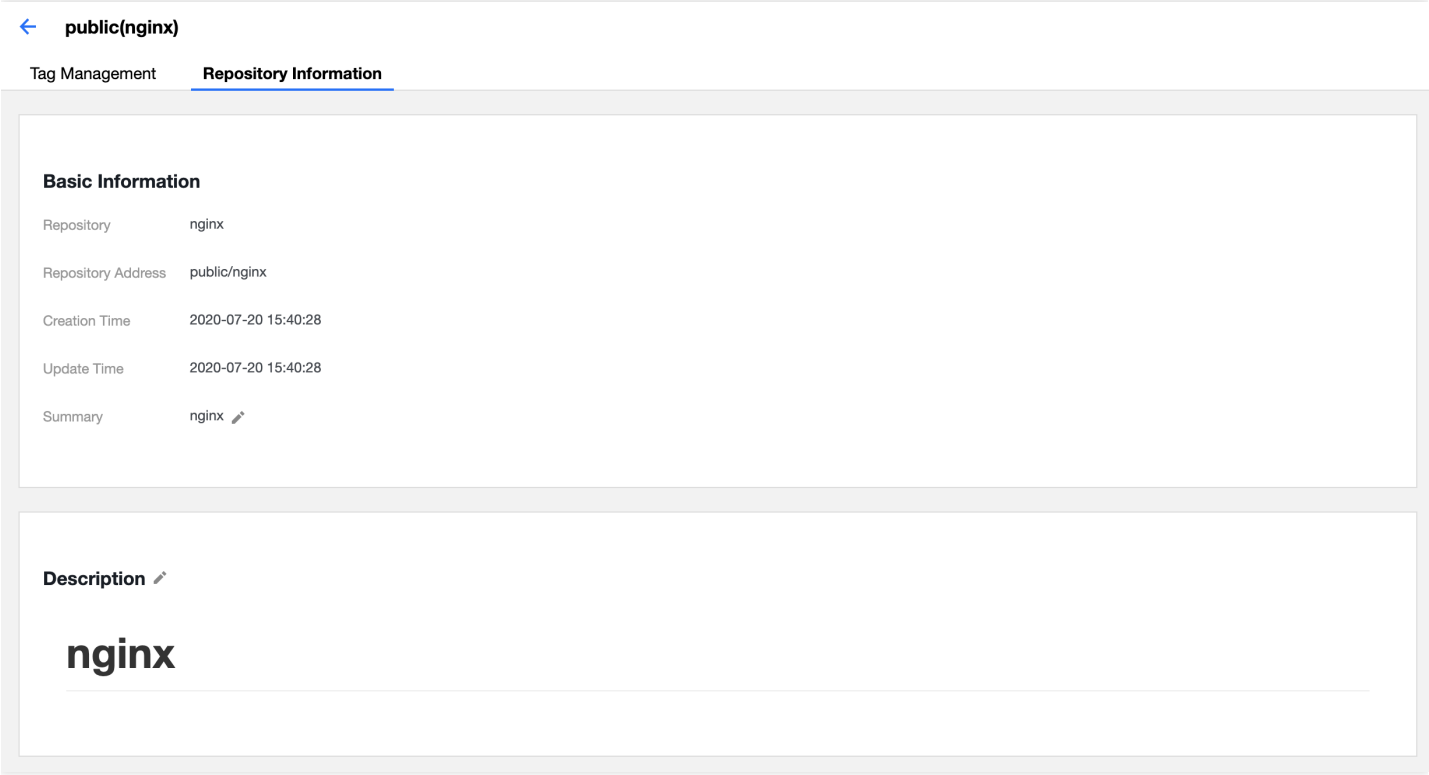
unavailable.

Building images

You can use source code hosted in GitHub, GitLab.com, Gitee.com, and CODING to perform compilation and building.

Editing the repository information

On the details page of the image repository, you can click the **Repository Information** tab to view and edit the basic information about the image repository, as shown in the following figure:



- **Editing the summary**
  Click for "Summary" to activate editing. After editing the summary, click **Save** to save the change.
- **Editing the description**
  Click for "Description" to activate editing. After editing the description, click **Save** to save the change. "Description" supports the Markdown syntax. You can view the rendered text after saving the change.

Obtaining an Instance Access Credential

Last updated : 2020-07-28 15:55:33

Scenario

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 command**: 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.

Prerequisites

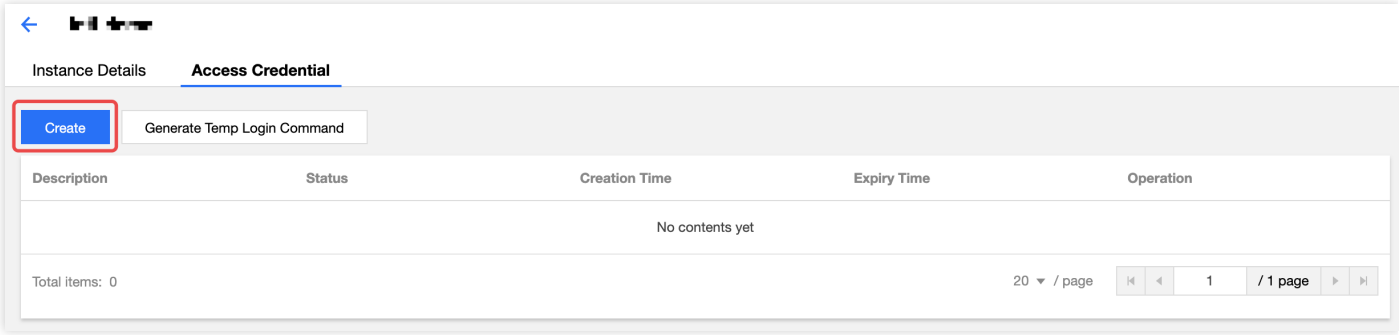
Before obtaining an access credential for a TCR Enterprise Edition instance, you must complete the following preparations.

- You have created a TCR Enterprise Edition instance.
- 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

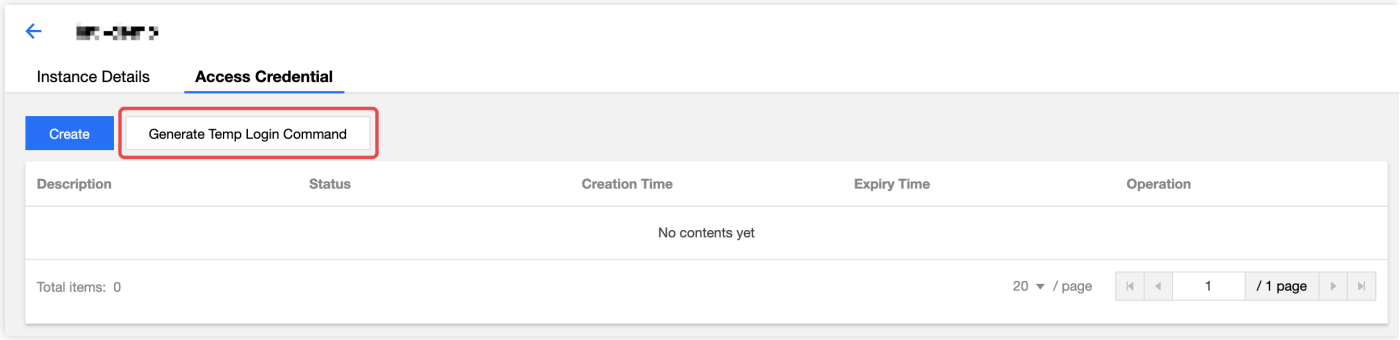
1. Log in to the TCR console and click **Instance List** in the left sidebar.
2. On the "Instance List" page, choose an instance name to go to the instance details page.
3. Click the **Access Credential** tab and click **Create** above the instance list, as shown in the following figure:



4. In the "Create an Access Credential" window that appears, complete the following steps to obtain an access credential:
   i. In the "Create an access credential" step, enter the purpose of the credential in "Purpose", and click **Next**.
   ii. In the "Save the access credential" step, click **Save the access credential** to download the credential. **Store the access credential properly, as it can be saved only once.**
   After the creation, you can view the credential on the **Access Credential** tab page. You can also disable or delete the credential now.

Obtaining a temporary login command

1. Log in to the TCR console and click **Instance List** in the left sidebar.
2. On the "Instance List" page, choose an instance name to go to the instance details page.
3. Click the **Access Credential** tab and click **Generate a temporary login command**, as shown in the following figure:



4. In the "Temporary Login Command" window that appears, click **Copy login command** to obtain the temporary access credential.

Next Steps

Refer to Logging in to a Registry Instance to log in to the TCR Enterprise Edition instance.

Network Access Control
Network Access Control Overview

Last updated : 2020-05-29 16:11:32

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

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

- Public network access control
- Private network access control

Public Network Access Control

Last updated : 2020-07-30 14:19:00

Scenario

The Tencent Container Registry (TCR) Enterprise Edition supports public network access control. An allowlist can be configured to restrict clients' access to the instance through the Internet, ensuring instance data privacy and security. When a TCR Enterprise Edition instance is created, the public network access entry is disabled by default. This means 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 Edition instance.

Prerequisites

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

Procedure

Opening the public network access entry

1. Log in to the TCR console and choose **Access Control** -> **Public network** in the left sidebar.
2. On the "Public network" page, select an instance name from the **Instance Name** drop-down list at the top of the page if you want to change the instance.
3. Select **Open Internet Access Entry** in the upper-right corner to enable the public network access entry.
   When the status of this button changes from **Opening** to **Close Internet Access Entry** and **Add a Public IP Allowlist** becomes selectable, the public network access entry has been successfully enabled. See the figure below.
   After the entry is enabled, all Internet access requests are still denied.

Configuring the access policy

1. On the "Public network" page, click **Add a Public IP Allowlist** and add the public IP address range and note in the displayed window. See the figure below.

- **Associated Instance**: target instance, for which the public network access policy is configured. You can change the instance by selecting another instance name from the "Instance Name" drop-down list at the top of the "Public network" page.
- **Entry**: public IP address range that is allowed to access the instance. The value can be a single IPv4 address or CIDR, 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.
- **Note**: note of the access policy. This parameter is optional.

2. Click **OK**. The public network access allowlist policy is added and takes effect.

- If you need to change the allowlist information, delete the policy and create another one.
- If the public network access entry cannot be enabled or the allowlist policy cannot be created, submit a ticket.

Private network Access control

Last updated : 2020-07-30 14:19:20

Scenario

The Tencent Container Registry (TCR) Enterprise Edition supports private network access control. A Virtual Private Cloud (VPC) access link can be used to restrict instance access by clients in the VPC. In actual production scenarios involving container computing, pulling container images through the 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.

Prerequisites

Before configuring private network access control for a TCR Enterprise Edition instance, complete the following tasks:

- Activate the VPC service and create a VPC and subnet in the region where the TCR Enterprise Edition instance is deployed.
- Activate the Domain Name Service (DNS) for the VPC in the Tencent Cloud DNS console.

Procedure

1. Log in to the TCR console and choose **Access Control** -> **Private network** in the left sidebar.

2. On the "Private network" page, click **Create**.

3. In the "Create Private Network Access Allowlist" window, configure the VPC and subnet information, as shown in the figure below.
   - **Associated Instance**: target instance, for which the private network access policy is configured. You can change the instance by selecting another instance name from the "Instance Name" drop-down list at the top of the "Private network" page.

- **Virtual Private Cloud**: connected VPC. Select the VPC that you want to connect. The drop-down list displays all VPCs available in the region of the current instance.
- **Subnet**: any subnet in the VPC. Select a subnet in the VPC that has usable private IP addresses. Creating a VPC access link will occupy a private IP address and use this IP address as the destination address for private network resolution of the instance domain name.

4. Click **OK** to start creating the VPC access link.
   If "Access Linkage Status" changes to **Normal linkage**, and "Private network parse IP" is not empty, the VPC access link was successfully created.

5. Log in to the Tencent Cloud DNS console and select **VPC** to go to the VPC resolution configuration page. Configure the resolution records of "Instance Domain Name" and "Private network parse IP" as prompted.

   > Currently, VPC resolution is a beta feature of Tencent Cloud DNS. If this feature is not activated, you can configure these resolution records on a VPC node or in your own DNS service.

Management Helm Chart

Last updated : 2020-05-29 16:03:31

Scenario

Tencent Container Registry (TCR) can host Helm charts to meet users' 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 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. That is, 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

Before uploading and managing Helm charts in a TCR Enterprise Edition instance, complete the following tasks:

- Create an Enterprise Edition instance.
- If you are using a sub-account, grant the sub-account operation permissions for the corresponding instance in advance. For more information, see Examples of Enterprise Edition Authorization Schemes.

Procedure

Configuring and installing the Helm client

1. Download the specified Helm client from the official Helm project and install it.
   Note that, if you wish to use Helm in Tencent Kubernetes Engine (TKE), you need to select the v2.10.0 version. You can run the `helm version -c` command to check the version of the installed client.

```
# The Linux platform is selected by default. If you install the client on other platforms, dow
nload the installation package corresponding to this platform.

# Decompress the installation package.
tar -zxvf helm-v2.10.0-linux-amd64.tgz
# Move the installation package to the specified location.
mv linux-amd64/helm /usr/local/bin/helm
```

2. Install the Helm plugin.
   You need to install Git and the Helm-Push plugin before using the Helm client to upload charts.

```
# Install the Helm plugin.
helm plugin install https://github.com/chartmuseum/helm-push
```

3. Initialize Helm.
   - Initialize Helm on the container cluster node. By default, Helm is activated and Tiller is installed.

```
helm init --client-only --skip-refresh
```

   - Tiller is not installed in your Kubernetes cluster.

```
helm init --skip-refresh
```

Adding a Helm repository

4. Obtain the access credentials for the current instance. The access credential is the username plus the temporary password, which is consistent with the credential used for Docker login.

5. Add the namespace, which is used to manage Helm charts, to the local Helm repository.

```
helm repo add $instance-$namespace https://$instance.tencentcloudcr.com/chartrepo/$namesapce --username $username --password $instance-token
```

"$instance-$namespace" is the name of the Helm repository. We recommend that you name this Helm repository using the following format to differentiate instances and namespaces: instance name + namespace name. `https://$instance.tencentcloudcr.com/chartrepo/$namespace` is the remote address of the Helm repository, where "$instance" and "$namespace" must be replaced by the actual instance name and namespace name. "$username" and "$instance-token" are the

username and temporary password obtained in Step 1.
After the namespace is added, the following prompt is displayed:

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

Pushing Helm charts

The installed Helm-Push plug-in can use the `helm push` command to push Helm charts to a specified repository. Both directories and compressed packages can be uploaded.
In the following example, tcr-chart-demo 1.0.0 is uploaded to the repository added in the previous step.

```
# Create a local chart.
helm create tcr-chart-demo
# Push the chart directory, where "$instance-$namespace" is the name of the added local repositor
y.
helm push tcr-chart-demo $instance-$namespace
# Push the chart package, where "$instance-$namespace" is the name of the added local repository.
helm push tcr-chart-demo-1.0.0.tgz $instance-$namespace
```

Pulling Helm charts

```
# Pull Helm charts of a 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" Enterprise Edition instance.

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

Managing Helm charts on the console

1. Log in to the TCR console and select **Helm Chart** in the left sidebar.
2. On the "Helm Chart" page, you can view the list of Helm charts in 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.
   The Helm chart list provides the following information and operations:
   - Name: name of a Helm chart. You can click a chart name to go to the details page of this chart.
   - Namespace: namespace to which a Helm chart belongs.
   - Create Time: time when the Helm chart is pushed to the repository for the first time.
   - Operation: you can click **Delete** to delete the current repository.

3. Click a specific chart repository to go to the details page. On the details page, you can view and manage chart versions. In addition, you can view the details of the files contained in each chart version on the **Basic Information** tab.

Configure instance synchronization

Last updated : 2020-05-29 16:00:28

Scenario

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

The instance synchronization feature allows you to customize synchronization rules and 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 versions. You can also select whether to overwrite existing images with the same names to prevent the loss of historical data due to overwriting.

Prerequisites

Before creating and managing a synchronization rule for a TCR Enterprise Edition instance, complete the following tasks:

- Create an Enterprise Edition instance.
- If you are using a sub-account, grant the sub-account operation permissions for the corresponding instance in advance. For more information, see Examples of Enterprise Edition Authorization Schemes.

Procedure

Creating a synchronization rule

1. Log in to the TCR console and select **Instance Synchronization** in the left sidebar.
   On the "Instance Synchronization" page, you can view the list of synchronization 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 Instance Synchronization Rule" window, configure the rule based on the following information. See the figure below.
   - **Name**: name of the instance synchronization rule. It can contain lowercase letters, numbers, hyphens (-), periods (.), and underscores (_), and must start with a letter or number.
   - **Description**: rule description.
   - **Source Address**
   - **Source Instance**: the current instance is the source instance. You can return to the "Instance Synchronization" page to change the source instance.
   - **Namespace**: namespace with which the current instance needs to synchronize. Currently, you cannot select all namespaces.
   - **Repository**: synchronized 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**: synchronized 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.
   - **Repository Type**: synchronized resource type. You can synchronize container images and Helm charts simultaneously, or only one of them.
   - **Target Address**
   - **Target Instance**: target instance for data synchronization. You can select any instance on the platform, including instances in other regions and source instances.
   - **Namespace**: namespace where the repository is located after this repository 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.
   - **Overwrite the image with the same name**: 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.

3. Click **OK** to create the synchronization rule.

Managing synchronization rules

After a synchronization rule is created, you can view the synchronization rule on the "Instance Synchronization" page. Then, you can perform the following operations to manage synchronization rules. See the figure below.

- **Viewing synchronization logs**: you can click a rule name to view the triggering logs of the rule.
- **Modifying the rule status**: a new instance synchronization rule is enabled by default. You can click the toggle button to enable or disable the rule.

- **Triggering synchronization**: you can click "Sync" in the "Operation" column to manually trigger synchronization. Then, all repositories in the instance that match the rule are scanned and synchronized.
- **Configuring a rule**: you can click "Configuration" in the "Operation" column to re-configure all parameters of the instance synchronization rule.
- **Deleting a rule**: you can click "Delete" in the "Operation" column to delete the instance synchronization rule.

Viewing synchronization logs

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

- **Task ID**: synchronization task ID, which is unique in the instance.
- **Create Time**: time that the synchronization task is 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 the number of container images and Helm charts that need to be synchronized in a task is large, the task may remain in the synchronizing state for a long time.

Trigger Management

Last updated : 2020-07-28 15:55:33

Scenario

Tencent Container Registry (TCR) allows users to configure and use the flexible trigger feature. By configuring appropriate triggers in instances to quickly access the existing R&D process and CI/CD platform, users can realize container DevOps scenarios such as automatic triggering of application deployment by image update.

The trigger feature allows users to create custom trigger rules and view triggering logs. Triggering actions support the push, pull, and deletion of container images and Helm Chart. Trigger rules support flexible filtering by regular expressions. They also support regular filtering rules by specifying namespaces in instances and configuring image repositories and tags. In this way, the feature allows triggers to be launched only by certain repositories or image tags in special naming formats. The Header customization feature supports the configuration of the Header in `Key:Value` format for accessing the destination URL, which can be applied to authentication and other scenarios.

Prerequisites

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

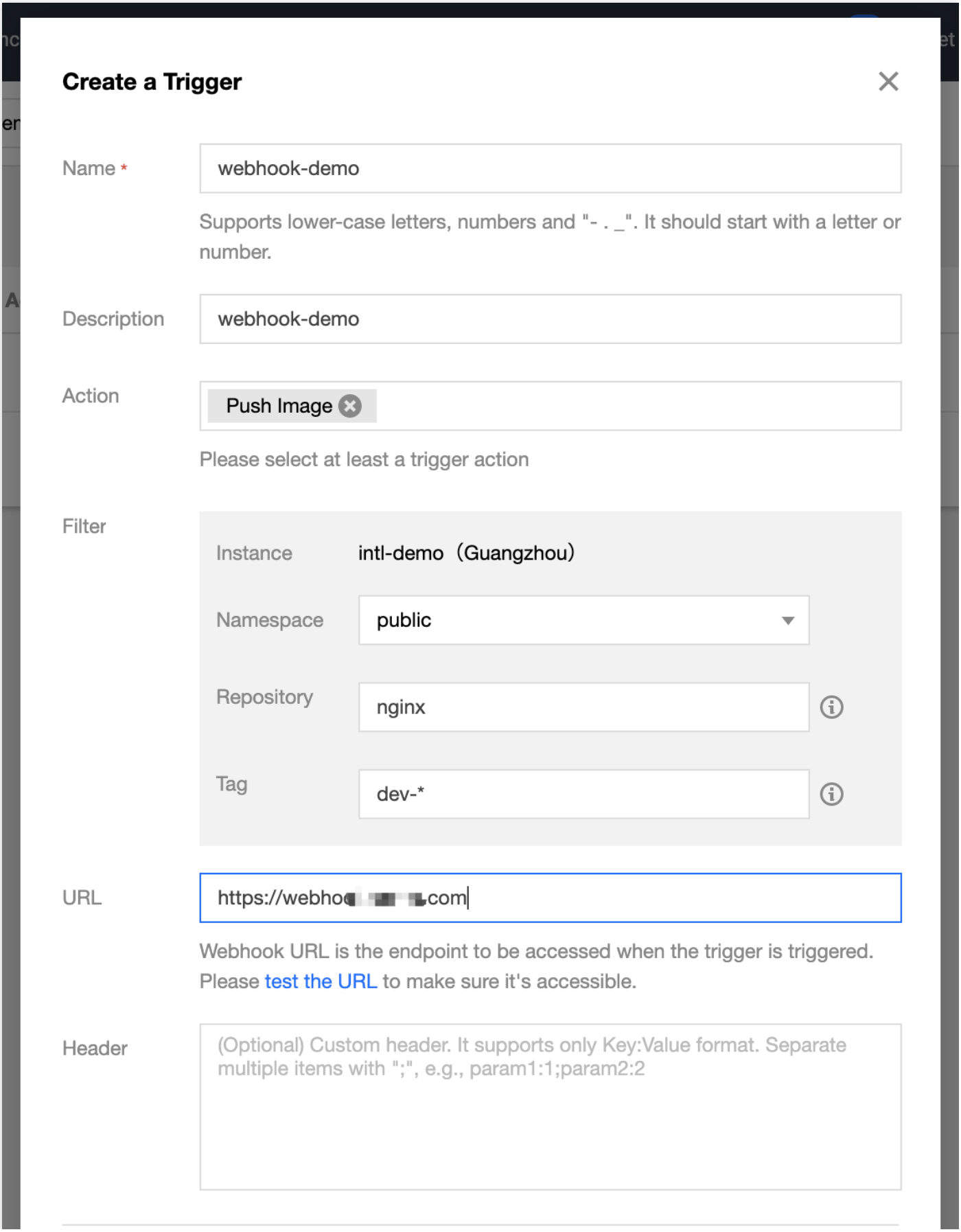
- You have created an TCR Enterprise Edition instance.
- If you are using a sub-account, ensure that you have granted the sub-account operation permissions for the corresponding instance in advance. For more information on how to grant the permissions, see Examples of TCR Enterprise Edition Authorization Schemes.

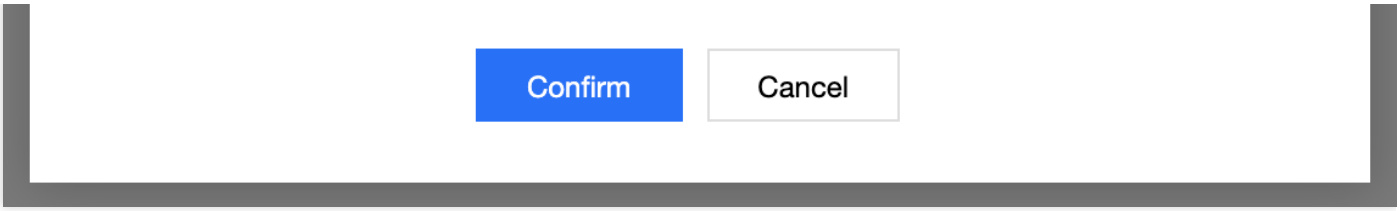
Directions

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, at the top of the page, select the desired instance name from the "Instance Name" drop-down list.

2. Click **Create**. In the "Create a Trigger" window that appears, configure the rule by referring to the following field description.





- **Name**: indicates the instance rule name. It supports lowercase letters, numbers, and three symbols (`-` , `.` , and `_`) and must start with a letter. In this document, `webhook-demo` is the sample instance rule name.
- **Description**: indicates the rule description.
- **Triggering action**: currently, six options are available: image push, image pull, image deletion, Helm Chart upload, Helm Chart download, and Helm Chart deletion. During the execution of a trigger, the webhook request initiated will contain information about the triggering action.
- **Triggering rules**:
  - **Triggering instance**: indicates the instance to which the trigger belongs, that is, the current selected instance. This field cannot be modified.
  - **Namespace**: indicates the namespace where the trigger runs. If the list is empty, first create a namespace in the instance.
  - **Repository name**: indicates the name of the repository where the trigger runs. Regex matching of the image repository and Helm Chart repository is supported.

    Note :

    Regex rules can be specified as `nginx-*` and `{repo1, repo2}` , in which:

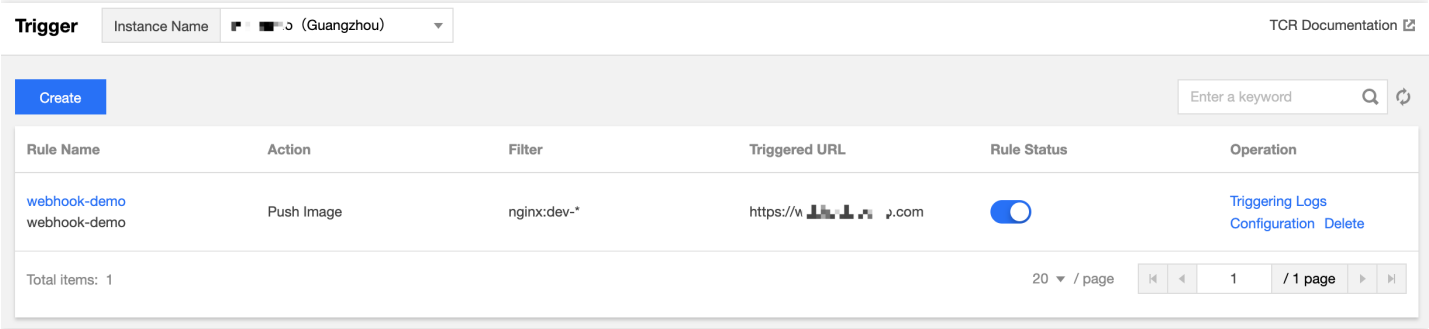
    - `*` : matches any field that does not contain '/'.
    - `**` : matches any field that contains '/'.
    - `?` : matches any single non-'/' character.
    - `{option 1,option 2,...}` : matches multiple options simultaneously.

  - **Tag**: indicates the tag for which the trigger runs. Regex matching is supported. The rule is the same as the repository name rule. To apply the trigger to all tags, leave this parameter unspecified.
- **URL**: indicates the destination URL for the request initiated after the trigger is fired. The trigger will initiate a POST request to the URL, and the request body will contain information such as the triggering action and trigger rule.
- **Header**: indicates the Header information that can be carried when the trigger initiates a POST request. It supports the Key:Value format, such as `Authentication: xxxxxxx` .

3. Click **OK** to create the synchronization rule.

Managing trigger rules

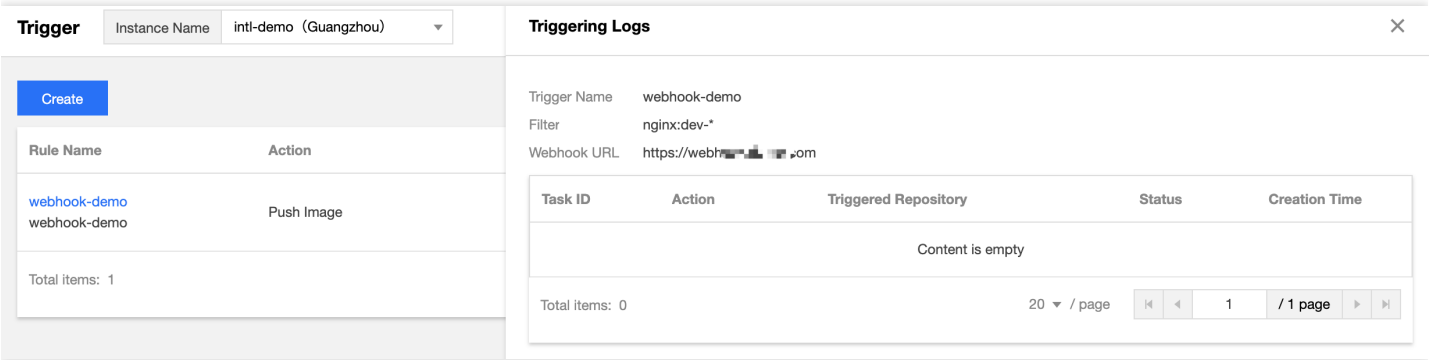
After trigger rules are created, you can view created trigger rules on the "Trigger" page. You can perform the following operations to manage trigger rules, as shown in the following figure:



- **View triggering logs**: you can click the name of a trigger rule to view its triggering log. For more information, see Viewing trigger logs.
- **Modify rule status**: indicates that the rule is enabled. indicates that the rule is disabled. New instance synchronization rules are enabled by default. You can adjust them based on your requirements.
- **Configure**: is to re-configure the trigger rule. You can re-configure all parameters.
- **Delete**: is to delete the trigger rule.

Viewing triggering logs

You can click the name of a trigger rule to view its triggering log, as shown in the following figure:



The log contains the following information:

- **Task ID**: indicates the ID of the trigger task, which is unique in the instance.
- **Triggering action**: indicates the action that fires this trigger, for example, image push.
- **Triggering repository**: indicates the repository that generates this triggering action.
- **Status**: indicates the status when the trigger successfully executes the webhook request.

- **Creation time**: indicates the time when the trigger is launched, that is, when the webhook request is initiated.

Related Information

Webhook request format reference

When users perform an action on resources that meet the rule, for example, pushing a new image to the specified image repository, the corresponding trigger will be fired, and an HTTP POST request will be initiated to the URL configured in the rule. The request body contains information such as the triggering action and repository path. The following request body information is parsed after the triggerring by an image push action. This information can be used as a reference for developing the Webhook server.


```
{
"type": "pushImage",
"occur_at": 1589106605,
"event_data": {
"resources": [
{
"digest": "sha256:89a42c3ba15f09a3fbe39856bddacdf9e94cd03df7403cad4fc105xxxx268fc9",
"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"
}
```