Tencent Container Registry
Operation Guide

Contents

Operation Guide
Creating an Enterprise Edition Instance

Last updated: 2023-09-13 16:03:04

Scenario

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

Preparations

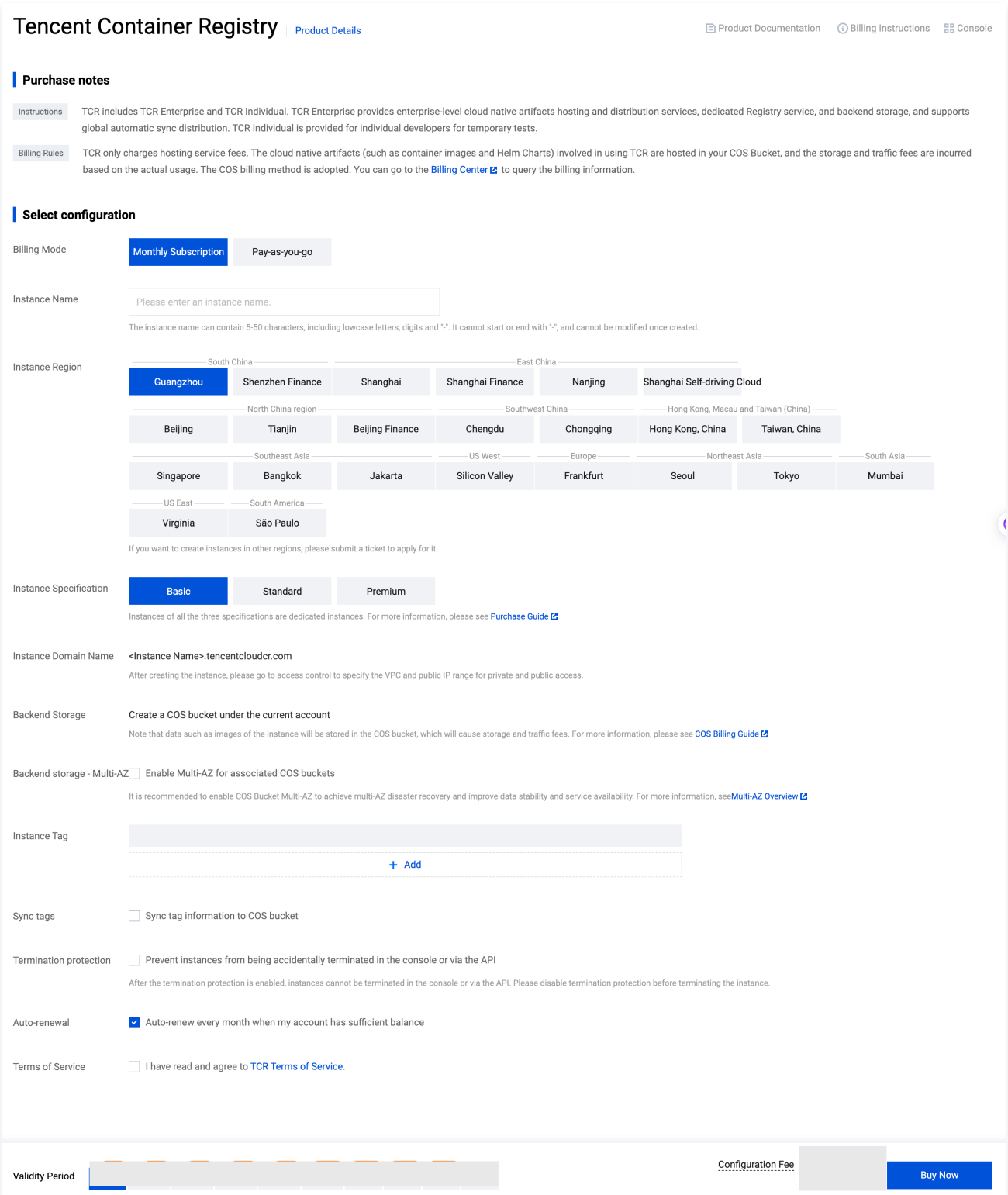
Before purchasing a TCR Enterprise Edition instance, complete the following tasks:

- Sign up for a Tencent Cloud account and complete identity verification.
- Activate the required cloud product for the container image service, Object Storage, which is used for storing 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 certain operation permissions to your COS and VPC resources.

Instructions

Creating via the console

1. Log in to the Tencent Cloud official website, select Container Registry, and click **Buy Now** to enter the Container Registry console.
2. Select **Instance Management** from the left navigation bar, navigate to the "Instance Management" page, and click on **Create**.
3. In the "Container Registry Purchase Page", refer to the following information to purchase an instance, as shown in the figure below:



- **Billing Mode**: Tencent Container Registry (TCR) offers two billing methods: annual/monthly subscription and pay-as-you-go. For more information, see the Billing Overview of TCR.
- **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 purchasing an instance, a Tencent Cloud Object Storage (COS) bucket will be automatically purchased and associated with your account. The instance's images and other data will be stored in this bucket, incurring storage and traffic fees. For details, see COS Billing Guide. After purchasing the instance, you can view the bucket in the COS console. Avoid accidentally deleting the bucket, as the images and other data hosted in the instance will be irretrievable.
- **Backend Storage – Multi-AZ Feature:** Optionally enable the associated COS bucket's Multi-AZ feature. It is recommended to enable the COS bucket Multi-AZ feature for multi-Availability Zone disaster recovery, higher data reliability, and service availability. However, this results in relatively higher storage capacity costs. For more information, refer to Object Storage COS Official Documentation – Multi-AZ 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 purchasing the instance.

4. Read and consent 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 changes to "Running", the instance was successfully purchased and is running properly.

> **Note**
> If the instance purchase takes an unusually long time or the status displayed is abnormal, you can contact us online.

Creating via API

You can also create an instance using the CreateInstance interface. For more information, refer to the Create Instance API documentation.

Supports and Limits

If you choose to purchase an Enterprise Edition instance using the pay-as-you-go billing mode, fees will be generated hourly after the instance is created, with the specific amount displayed on the purchase page. You can visit the Billing Center to check the fees generated by this service. If you have any questions about the charges, please contact us online.

Access Configuration
Credential Access Control
Managing User-Level Account

Last updated: 2023-09-13 16:02:23

Scenario

To push and pull container images, you must first log in to the instance using the credential information by executing the `docker login` command and entering the username and password in the Docker client. This username and password are only used for logging in and authentication of this instance and cannot be used in other scenarios. This document describes how to manage user-level accounts associated with Tencent Cloud accounts in TCR Enterprise Edition instances.
After purchasing a TCR Enterprise Edition instance, if you want multiple sub-accounts to manage and use it simultaneously, such as pushing/pulling images, the account administrator can first configure permissions for each sub-account (see TCR Enterprise Authorization Management for details). After logging in to the product console, sub-accounts can generate user-level accounts, which are Docker Registry access credentials associated with their identity (the username of the access credential is the same as the Tencent Cloud sub-account ID). These credentials can be used to log in to the repository and push/pull images. When using a user-level account associated with a sub-account to operate images, read and write actions will be recorded and traceable to the account holder, which can be used for internal audits.
When creating a user-level account, you can choose to create a temporary access credential or generate a long-term access credential. It is recommended to use temporary access credentials for daily temporary image push/pull operations to avoid data security risks caused by credential leaks.

- **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.

> ⚠ **Note**
> Please keep the access credential properly after it is generated. 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.

Preparations

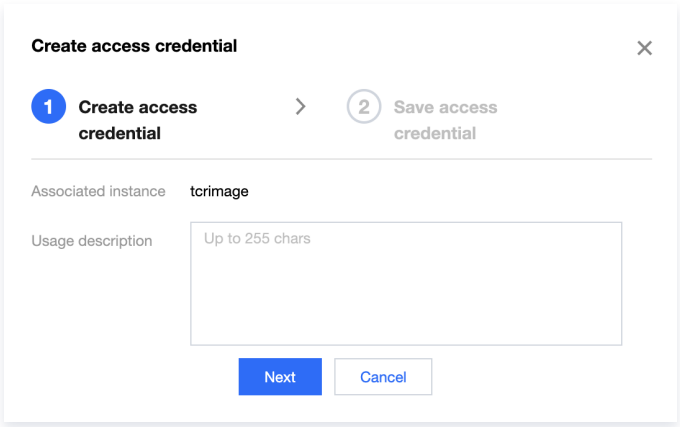
Before obtaining an access credential for a TCR Enterprise Edition instance, you must complete the following preparations.

- You have purchased an 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.

Instructions

Obtaining a long-term access credential

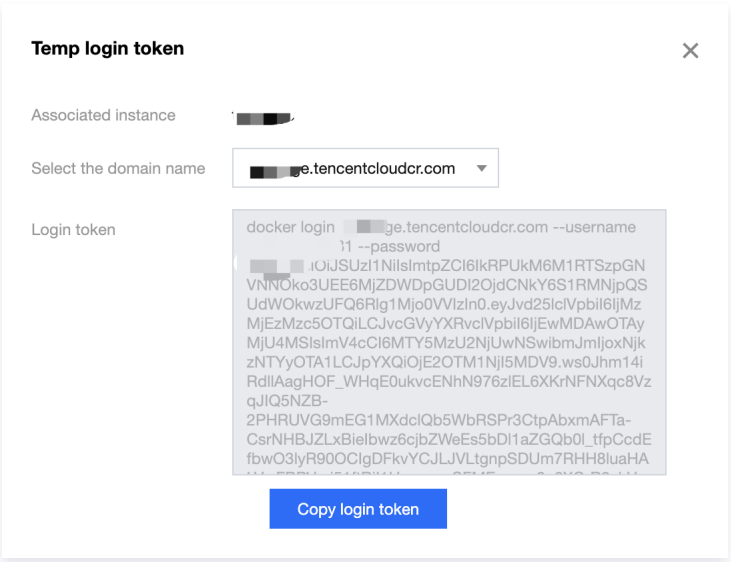
1. Log in to the TCR console and choose **Access credential** > **User accounts** in the left sidebar.
2. On the **User accounts** page, select a region and an instance, and click **Create**.
3. On the **Create access credential** page, perform the following steps:
   3.1 In the **Create access credential** step, enter a description for the credential's purpose and click **Next**.
   3.2 In the **Save access credential** step, click **Save access credential** to download the credential information. **Please keep the access credential safe, as you only have one chance to save it.**



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.



Creating via API

You can also use the CreateInstanceToken interface to create instance access credentials. For more information, please refer to Create Instance Access Credential.

See Also

Please refer to Logging in to the TCR instance for logging in to the TCR Enterprise Edition instance.

Supports and Limits

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 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: 2023-09-13 16:13:41

Scenario

To push and pull container images, you must first log in to the instance using the credential information, i.e., execute the `docker login` command in the Docker client and enter the username and password. This username and password are only used for logging in and authentication of this instance and cannot be used in other scenarios. This document describes how to manage custom service-level accounts in TCR Enterprise Edition instances for use in automated scenarios such as CI/CD.
Upon purchasing an Enterprise Edition instance, you can create user-level accounts bound to your Tencent Cloud account. The username is the same as the account ID, and the password is a randomly generated long string. For more information, refer to User-Level Account Management. User-level accounts are associated with Tencent Cloud accounts, allowing for fine-grained permission configuration based on Cloud Access Management (CAM) and traceability of repository logins, image pushes, and pulls. However, when the corresponding sub-account is deleted or disabled (e.g., when the account user is transferred or leaves the company), the associated user-level account will also become invalid. If this user-level account is used in automated systems such as CI/CD or configured in a Kubernetes cluster, it may cause image push and pull failures, affecting the business. Moreover, managing permissions based on CAM can be complex, and configuring CAM policies for different teams according to namespace dimensions can be challenging.
If you encounter the aforementioned issues, you can opt to use the service-level account feature. This feature supports:

- Custom username and password
- Namespace-specific read/write permission configuration
- Custom validity period. You can disable a service account temporarily.

> **Note:**
> 1. Service-level accounts support operation auditing, i.e., the service-level account used for upload and download operations will be recorded in the CloudAudit logs. However, the platform cannot verify or trace the actual user identity of the service-level account, so please be cautious when distributing service-level accounts externally. If you need to strictly audit the operators or account holders of image pull/push actions, please choose user-level accounts.
> 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.

Preparations

Complete the following tasks before using a TCR Enterprise Edition service account:

- You have purchased an 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.

Instructions

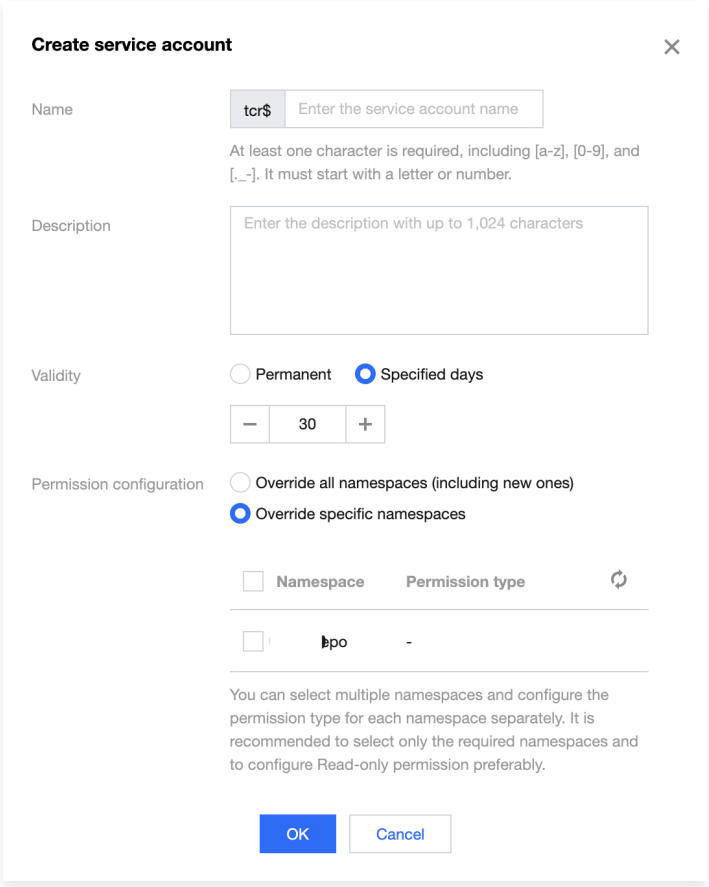
Creating a service account

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. In the **Create service account** page, follow these steps for configuration:
   - **Name**: Composed of lowercase letters, digits, and ._- characters, with at least one character and starting with a letter or digit. Note that a prefix will be automatically added to indicate that the account is a service-level account. For example, if you enter "robot-demo," the actual username will be "tcr$robot-demo."

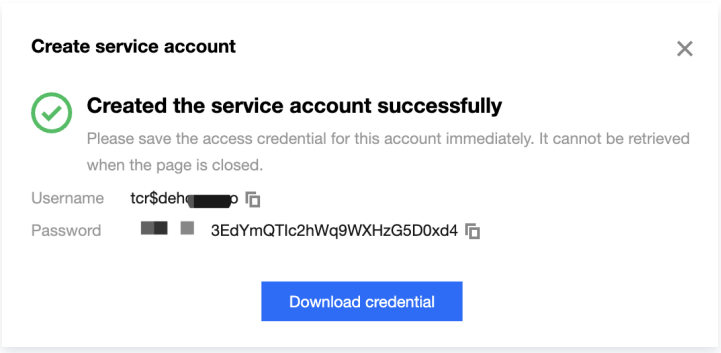
     > **Note:**
     > Some open-source CI/CD platforms may not correctly handle the tcr$ prefix. If you encounter any issues, you can replace it with the tcr@ prefix.

   - **Description**: Enter the account description.

- **Validity**: Choose between never expiring or expiring after a specified number of days, with the default being 30 days.
- **Permission configuration**: Supports selecting multiple namespaces and independently configuring permission types for each namespace; it is recommended to follow the principle of least privilege, selecting only necessary namespaces and prioritizing read-only permissions.
  - **Namespace**: Supports selecting multiple namespaces.
  - **Permission type**: Supports read-only and read-write configurations. Image push is not supported in read-only mode.



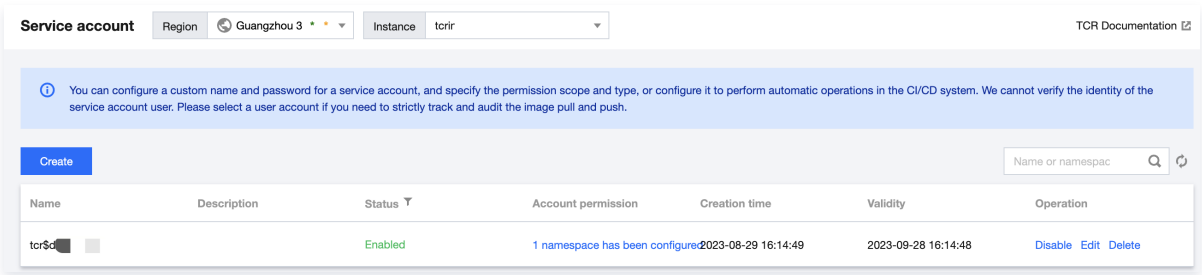
4. Note down the username and password immediately after the account is created. This page will be displayed only once and the credential information cannot be retrieved after the page is closed.



Managing service accounts

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 manage existing service-level accounts with the following operations:
   - 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. Please exercise caution when performing this operation.
- Delete specified service-level accounts. Note that once deleted, they cannot be recovered. After deletion, the account cannot be used for pushing or pulling images, so please proceed with caution.



Network Access Control
Network Access Control Overview

Last updated: 2023-09-13 16:00:24

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.
You can configure public and private network access control policies according to your specific business requirements, allowing the minimal scope of access for clients to interact with the instance, push, or pull images. For more information, please refer to:

- Configuring Public Network Access Control
- Private Network Access Control

Configuring Public Network Access Control

Last updated: 2023-09-13 16:00:10

Scenario

Tencent Container Registry (TCR) Enterprise Edition supports public network access control, which can restrict access to instances from public network environments based on a whitelist policy, ensuring data privacy and security within the instance. By default, newly created TCR Enterprise Edition instances do not have public network access enabled, meaning that development and testing servers in public network environments cannot directly push or pull images.

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

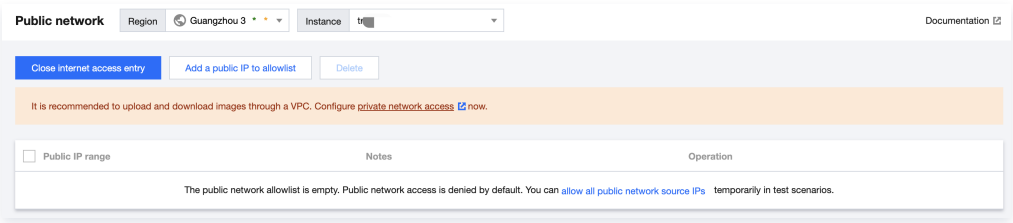
Preparations

Before configuring public network access control for a TCR Enterprise Edition instance, you need to successfully purchase an Enterprise Edition instance.

Instructions

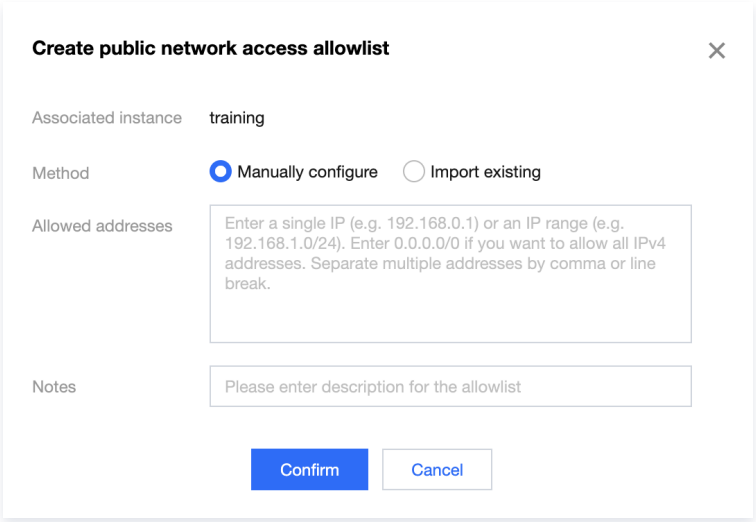
Enabling the public network access entry

1. Log in to the TCR console and choose **Access Control** > **Public Network Access** in the left sidebar.
2. On the "Public Network" page, if you need to switch instances, select the instance's region at the top of the page and choose from the **Instance Name** dropdown list.
3. Select **Open internet access entry** to begin enabling the entry.
   Wait for the button status to change from **Enabling** to **Close internet access entry**, and **Add a public IP to allowlist** becomes selectable, indicating that the entry is open, as shown in the figure below:
   After the entry is opened, all public access sources are still blocked by default.



Configuring Access Policy

1. On the "Public Networks" page, click **Add a public IP to allowlist** and configure the public IP address range and notes in the pop-up window, as shown in the following figure:



   - **Associated instance**: The instance for which the public network access policy is being configured. This can be modified in

the "Instance Name" dropdown at the top of the "Public Network Access" page.

- **Method**: Supports manually configuring public IP addresses or importing public IP lists from existing security groups.
  - **Manually configure**: Allow public IP address ranges. Supports single IPV4 addresses or CIDR, such as `192.168.0.0/24`. It is not recommended to enter `0.0.0.0/0` to allow unrestricted public network access.
  - **Import existing**: All allowed addresses in the inbound rules will be deduplicated and imported to the allowlist. You can import security groups multiple times, and manage the public network allowlist for the instance manually after importing. Please note that modifying the security group configuration again will not automatically synchronize; you need to re-import.
- **Notes**: Optional remarks for the access policy, supporting Chinese input.

2. After completing the configuration, click **Confirm**. The public network access allowlist policy will be added and take effect.

If you are unable to enable public network access or create a whitelist policy, please contact us for online assistance.

Access Permission Configuration
Access Control Management Overview

Last updated: 2023-09-13 15:57:20

Tencent Container Registry (TCR) offers two access control management modes for Enterprise Edition instances, enabling instance administrators to assign separate access credentials to development and operations personnel, as well as internal CI/CD and other automated systems. This fine-grained user permission management ensures data security within the instance.

Use Cases

Personnel Permission Management

After purchasing an Enterprise Edition instance, an organization typically has multiple business teams using the instance, involving various roles such as development, operations, and testing. To ensure data security, different teams and roles should be configured with the least privileges necessary to meet their business requirements, minimizing the risk of issues such as accidental image deletion or image leakage.
Specifically, individual accounts should be assigned to designated personnel for authentication purposes, and separate permissions should be configured for operational authorization. In the context of using an image repository, this means providing designated personnel with access credentials (username and password) for logging into the image repository (docker login) and configuring the account's permissions, such as allowing them to push and pull images only within Repository A or only pull images from Repository B.
Additionally, organizations need to audit personnel operations on image repositories and trace any abnormal activities for accurate identification.

System Permission Management

The production and deployment of container images also involve automated systems, such as CI/CD pipelines. In automated scenarios, using a designated person's account could have severe consequences. For example, if an operations staff member's access credentials are configured in the release system for pulling images in a cluster, and the staff member's account is accidentally deleted or the employee changes roles or leaves the company, the image access credentials will become invalid, leading to deployment issues. Therefore, in automated scenarios, access credentials unrelated to specific individuals should be used, and account permissions should be managed independently.

Product Features

To address these two typical scenarios, the product offers two access control management features: user-level accounts and service-level accounts. You can use these features in combination to meet your organization's permission management requirements.

User account

User-level accounts are directly associated with Tencent Cloud accounts and managed through Cloud Access Management (CAM). To grant permissions to designated personnel within an organization, you need to create a dedicated sub-account for them under the Tencent Cloud master account and associate the sub-account with permission policies in CAM. For more information, please refer to Managing Sub-account Permissions with CAM.
The user logs into the product console using a sub-account and navigates to the **Access Credentials - User Accounts** feature page to create a dedicated access credential. This access credential is associated with the sub-account and is only visible and manageable by the sub-account. The user can use this access credential to log in to the image repository and push or pull images, with actual operations controlled by the associated permission policies.

> **Note**
> Please note that when a sub-account is disabled or deleted, the associated access credentials will also become invalid. Therefore, avoid configuring access credentials associated with sub-accounts in automated systems.

For detailed usage of user-level accounts, please refer to User-Level Accounts.

Service Accounts

Service accounts are not directly associated with Tencent Cloud accounts and are considered instance-specific resources, with permission management independent of the CAM (Cloud Access Management). If you need to configure access credentials for automated systems, such as CI/CD pipelines or Kubernetes clusters, to automatically push and pull images, you can create a service account within the instance and configure its permission scope.

> **Note**
> As service accounts are resources within an instance, all sub-accounts with relevant API permissions for service accounts can query and manage them. However, since the account permissions are independent of the CAM (Cloud Access Management) permission system, there is a risk of broken access control. Please strictly control the authorization scope of related APIs.

For more information on using service-level accounts, please see Service-Level Accounts.

Managing Sub-account Permissions with CAM

Last updated: 2023-09-13 15:57:09

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 external permissions when using certain features in the TCR console, such as private networks, cloud audits, or cloud tags, please refer to the access management guide documentation for the corresponding products in Products Supported by CAM.

Cloud Access Management (CAM) Introduction

Cloud Access Management (CAM) is a web-based Tencent Cloud service that primarily assists users in securely managing access permissions to resources under their Tencent Cloud accounts. With CAM, you can create, manage, and terminate users (groups) and control who can use which Tencent Cloud resources through identity management and policy management.
When using CAM, you can associate a policy with a user or user group to allow or deny them to use specified resources to complete specified tasks. For more information on CAM policies, see Policy Syntax. For more information on how to use CAM policies, see Policy.
If you do not need to manage TCR-related resource access for sub-accounts, you can skip this section without affecting your understanding and use of the remaining parts of the documentation.

TCR Resource-Level Access Control Based on CAM

Resource-level permission refers to the ability to specify which resources users are allowed to perform operations on. Tencent Container Registry (TCR) supports resource-level access control based on CAM, with control granularity down to the repository level. This means that users can configure CAM policies to authorize sub-accounts to operate only on specified image repositories or Helm Chart repository resources.
Resource types in TCR that can be authorized through CAM:

ResourceType	Resource Description Method in Authorization Policy
Enterprise Edition Instance-related	`qcs::tcr:$region:$account:instance/*`
Enterprise Repository Related	`qcs::tcr:$region:$account:repository/*`
Personal Repository Related	`qcs::tcr:$region:$account:repo/*`

- `$region`: Describes region information, such as `ap-guangzhou` representing the Guangzhou region. If the value is empty, it means all regions. For a detailed list of regions and abbreviations, please refer to Regions and Availability Zones.
- `$account`: Describes the root account information of the resource owner, represented as `uin/${uin}`, for example, `uin/12345678`. If the value is empty, it refers to the root account to which the CAM user creating the policy belongs.
  For more information on resource description methods in authorization policies, please refer to Resource Description Methods.

Instructions

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

Create Using Policy Generator (Recommended)

1. Log in to the Cloud Access Management console.
2. In the left sidebar, click **Policies** to enter the policy management page.

3. Click on **Create Custom Policy** in the top left corner.
4. In the pop-up window for selecting the creation method, click **Create by Policy Generator** to enter the policy editing page.
5. Select the service in the Visual Policy Generator, enter the following information, and edit an authorization statement.
   - **Effect**: Choose between 'Allow' or 'Deny'. In this instance, we select **'Allow'**.
   - **Service**: Select the product to be authorized; in this case, we choose **Tencent Container Registry (tcr)**.
   - **Action**: Select the operation that requires authorization; in this case, we choose **Read operation**.
   - **Resource**: Choose either all resources or specific resources you want to authorize. In this case, we select **specific resources** and add the following six-segment resource description to restrict access.
     - **Repository**: Select the region of the repository and enter its resource path, such as `tcr-xxxxxxxx/team-01/repo-demo/*`. You can obtain the resource path in the Image Repository.
     - **repo**: Leave this field empty.
     - **Instance**: Select the region of the repository and enter the instance ID of the repository's associated instance, such as `tcr-xxxxxxxx`. You can obtain the instance ID from the Instance List.
   - **Condition**: Leave this field empty.
6. Click **Next** to proceed to the Associate User/User Group 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 finish creating a custom policy using the policy generator.

Create by Policy Syntax

1. Log in to the Cloud Access Management console.
2. In the left sidebar, click **Policies** to enter the policy management page.
3. Click on **Create Custom Policy** in the top left corner.
4. In the pop-up window for selecting the creation method, click on **Create by Policy Syntax** to navigate to the "Select Policy Template" page.
5. In the template type selection, choose **Blank Template**.
6. Click **Next** to proceed 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 finish creating a custom policy based on policy syntax.

Common Policy Configuration

If you need to customize and edit the policy JSON, please refer to the Enterprise Edition CAM API List and Policy Syntax.

Preset policy configuration

- **QcloudTCRFullAccess**: Full read and write permissions for Tencent Container Registry (TCR).
  After binding this policy to a sub-account, it will have full operation permissions for all TCR resources, including Enterprise Edition and Personal Edition.

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

- **QcloudTCRReadOnlyAccess**: TCR Read-Only Permission.
  After binding this policy to a sub-account, it will have read-only access to all resources of the container image service, including both Enterprise and Personal editions.

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

Policy Configuration in Typical Scenarios

> **Note**
> The following policy scenarios are applicable only to TCR Enterprise use cases. For TCR Individual policy scenarios, please refer to TCR Individual Authorization Solution Examples.

- 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"
  }]
}
```

- Authorize a sub-account to manage a specific instance, such as dev-guangzhou, with an instance ID of tcr-xxxxxxxx.

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

- Authorize a sub-account to manage a specific namespace within a designated instance, such as the team-01 namespace under the tcr-xxxxxxxx instance.

```
{
  "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"
      }
  ]
}
```

- Authorize a sub-account with read-only access to a specific image repository, allowing it to pull images from the repository but not delete the repository, modify its properties, or push images. For example, the repo-demo in the team-01 namespace under the tcr-xxxxxxxx instance.

```
{
  "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: 2023-09-13 15:56:44

Scenario

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.
In TCR Enterprise Edition instances, all instance types support configuring multiple custom domain names without affecting the normal use of existing default domain names. To use a custom domain name, you need to provide an SSL certificate associated with the domain name and access the instance through the HTTPS protocol. This article explains how to access a TCR Enterprise Edition instance via a custom domain name.

Concepts

Domain name

Domain name refers to a series of characters separated by dots. In the TCR Enterprise Edition, domain names are used to access instance services and directly affect the access address of the image repository.

SSL certificate

SSL Certificate is used to comply with the HTTPS protocol, enabling TCR Enterprise Edition instances to perform encrypted transmission and identity authentication through HTTPS, ensuring the security of the transmission process.

DNSPod

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

Preparations

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

- You can register a domain name through Tencent Cloud Domain Service. For more information, see Domain Registration.

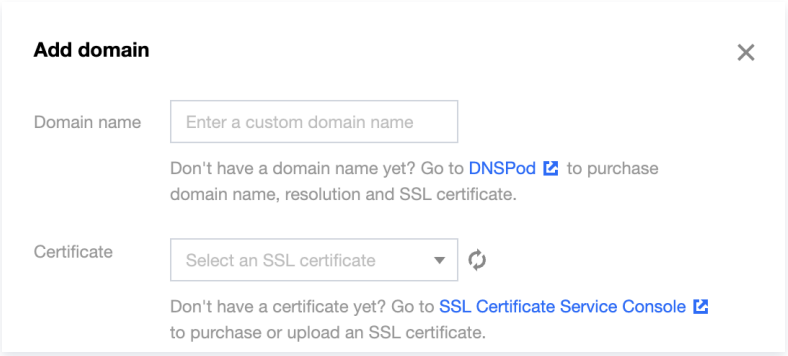
> ⚠ Note
> - If you wish to use a custom domain name in a public network environment, your domain name must be ICP filed.
> - You do not need to get an ICP filing if your TCR Enterprise Edition instance is outside the Chinese mainland.

- You have obtained a certificate for the domain name. You can purchase a certificate through Tencent Cloud SSL Certificate Service and ensure that the custom domain name required for the instance is bound.
- You have activated Tencent Cloud DNSPod service. For more information, please refer to DNS Resolution and Private DNS Resolution.

Instructions

Creating custom domain name

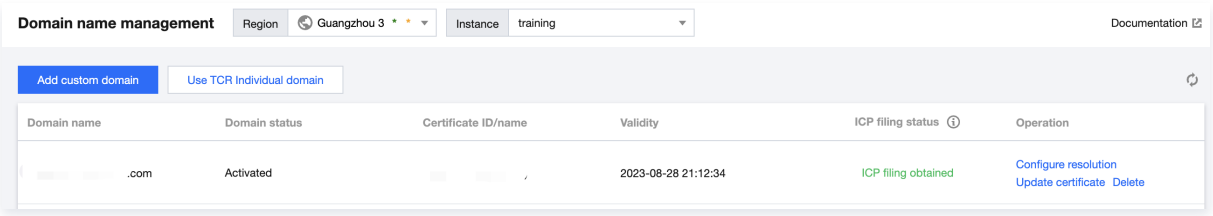
1. Log in to the Container Registry console and click **Domain Management** in the left sidebar.
2. On the **Domain name management** page, select the region and ID of the instance for which you want to add a custom domain name.
3. Click **Add Domain**. In the "Add domain" pop-up window, configure the domain name and certificate information according to the following tips, as shown in the image below:



- **Domain name:** To use a custom domain name, it is recommended to use common domain name suffixes.
- **Certificate:** A certificate bound to a custom domain name, which only supports selecting certificates hosted within Tencent Cloud SSL Certificate Service.

> **Note**
> If your custom domain name has been registered with the Ministry of Industry and Information Technology (recorded) and has been resolved in the Domain Name Service Console, simply enter your custom domain name in the "Domain Name" input box and select the certificate.

4. Click **OK** to add a custom domain name.
   After successfully adding a custom domain name, you can view it on the "Domain name management" page. At this point, you can follow the steps below to manage your custom domain name, as shown in the following image:



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

Configuring private network access control

Please refer to Configuring Private Network Access Control to configure the VPC to connect to the instance and ensure that the private network access IP has been generated successfully.

Configuring Private DNS

Go to the PrivateDNS console, use the added custom domain name to create a private zone, associate it with the connected VPC, and configure the resolution within the private zone.

1. Use CNAME with the record value set to the standard access domain name of the instance. Please note that when configuring the VPC access within TCR, you need to enable auto-parsing.
2. Use an A record with the record value set to the private access IP of the created private network access link. Please note that the private access IP may change if the VPC is reconnected in the future.

For more information, please refer to Private DNS.

Public network access

Configuring public network access control

Please refer to Public Network Access Control to enable the public network access entry and allow public network access addresses.

Configuring public network DNS

Please go to the DNSPod console and configure the cloud resolution for the added custom domain name. Select "CNAME" as the record type and enter the default domain name of the instance as the record value. For more information, refer to Quickly Add Domain Name Resolution.

Updating domain certificate

If you need to update the certificate bound to a custom domain name due to certificate expiration or upgrading, you can click **Update Certificate** on the right side of the custom domain name row in the "Domain Management" page and reselect the SSL certificate. Updating the certificate requires reissuing the SSL certificate; during this period, the custom domain name can still be accessed normally.

Deleting custom domain name

On the "Domain Management" page, select **Delete** on the right side of the row where the specified custom domain name is located to delete it. Deleting a custom domain name may cause existing container image pull configurations in the container cluster to become unusable, which may affect application updates. Please proceed with caution.

Image Creation
Managing Namespaces

Last updated: 2023-09-12 18:25:51

Scenario

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.
In TCR Enterprise Edition instances, since each enterprise has exclusive access to the instance, there is no need to worry about the desired namespace being occupied by other customers. However, in Individual Edition instances, it is still necessary to avoid naming conflicts with existing namespaces. This document explains how to create and manage namespaces within TCR Enterprise Edition instances.

Preparations

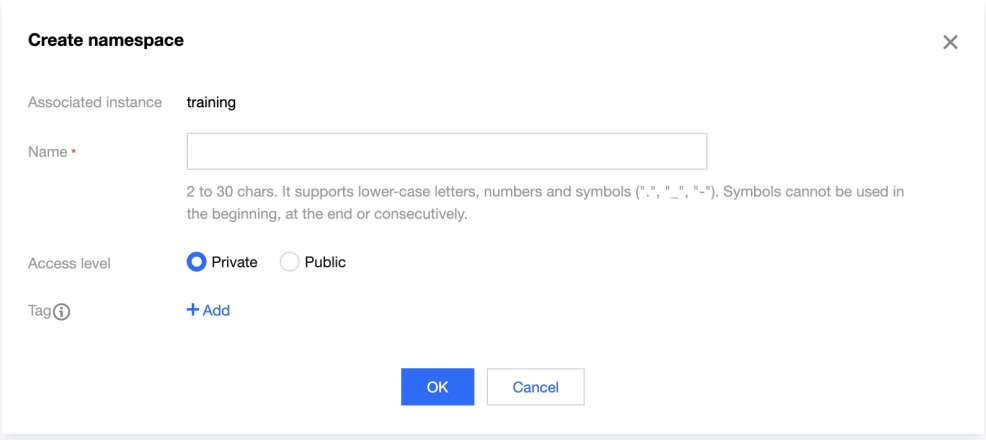
Before creating and managing a namespace in a TCR Enterprise Edition instance, complete the following preparations:

- You have purchased an Enterprise Edition 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.

Instructions

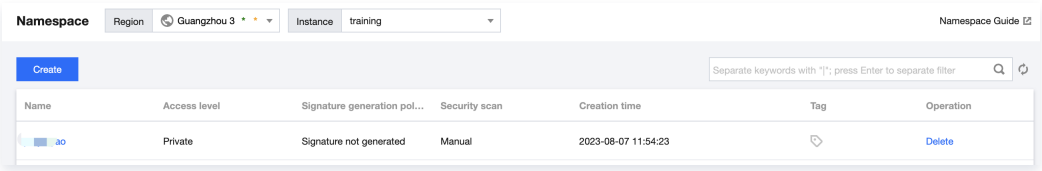
Creating namespace

1. Log in to the TCR console and click **Namespace** in the left sidebar.
2. On the "Namespace" page, you can view the list of namespaces in the current instance. If you need to switch instances, select from the **Instance Name** dropdown list at the top of the page.
3. Click **Create**. In the "Create namespace" window, configure the namespace name and access level according to the following prompts, as shown in the figure below:



   - **Associated instance:** Currently selected instance, to which the new namespace will belong.
   - **Name:** Namespace name, 2 – 30 characters in length, can only contain lowercase letters, numbers, and separators `.`, `_`, `-`, and cannot start, end, or be consecutive with separators. It is recommended to use organization teams, product projects, or personal names for naming, or as a personal testing space. Namespace names must be unique within a single instance.
   - **Access level:** You can choose between **Private** or **Public**, with the default setting being Private.
     If set to Public, all image repositories and Helm charts within the namespace become public repositories. If the instance also enables anonymous access (enabled by default), any client that passes access control can pull images and charts without logging in. Please exercise caution when setting this attribute, as it can be modified again after creation.
   - **Tag:** You can bind cloud tags for categorizing resources at the namespace level. If the existing tags do not meet your requirements, go to the console Manage Tags.
4. Click **OK** to create the namespace.

After successful creation, you can view it on the "Namespace" page, where you can manage the namespace following the steps below, as shown in the figure:



Changing the access level

1. Click the namespace that you want to change the access level and go to its “Basic Information” page.
2. In "Basic Information", click in "Access Level" to switch between public and private attributes for the namespace in the pop-up window.

> **Note**
> Upon switching, all image repositories and Helm charts in the namespace will immediately inherit the change. **Please exercise caution when changing a private namespace to a public one.**

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. In "Basic Information", click in "Security Scanning" to toggle the security scanning attribute for container images in the namespace. It can be set to **manual** or **automatic** mode:

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

- **Manual Scanning**: To perform a security scan on a specific container image and view the results, navigate to the "Image Repository" page, select the image, and click **Scan** in the **Tag Management** tab.
- **Scheduled Scan**: When a new image is pushed to any image repository within the namespace, an automatic security scan will be triggered.

Configuring deployment security

TCR Enterprise Edition supports blocking the deployment of high-risk images. Please refer to Blocking High-Risk Image Deployment.

Deleting namespace

To delete a namespace, click **Delete** on the right side of the specified namespace row. To prevent accidental deletion of important data, namespaces containing image repositories and Helm charts cannot be deleted.

Managing Image Repository

Last updated: 2023-09-13 16:07:00

Scenario

In Tencent Container Registry (TCR) Enterprise Edition, an image repository is used to manage container images directly. A single image repository may contain container images with different versions. An image repository belongs to a namespace and inherits the public or private attribute and security scan triggering mode from its namespace.
Image repositories are the smallest units for permission management in TCR. Instance administrators can grant sub-users management or read-only permissions for image repositories. For example, granting sub-account tom permission to pull images from the project-a-frontend repository, but not to push or delete images. For more information on permission management and authorization methods, please refer to Enterprise Edition Authorization Scheme Examples. This document describes how to create and manage image repositories within a TCR Enterprise Edition instance.

Preparations

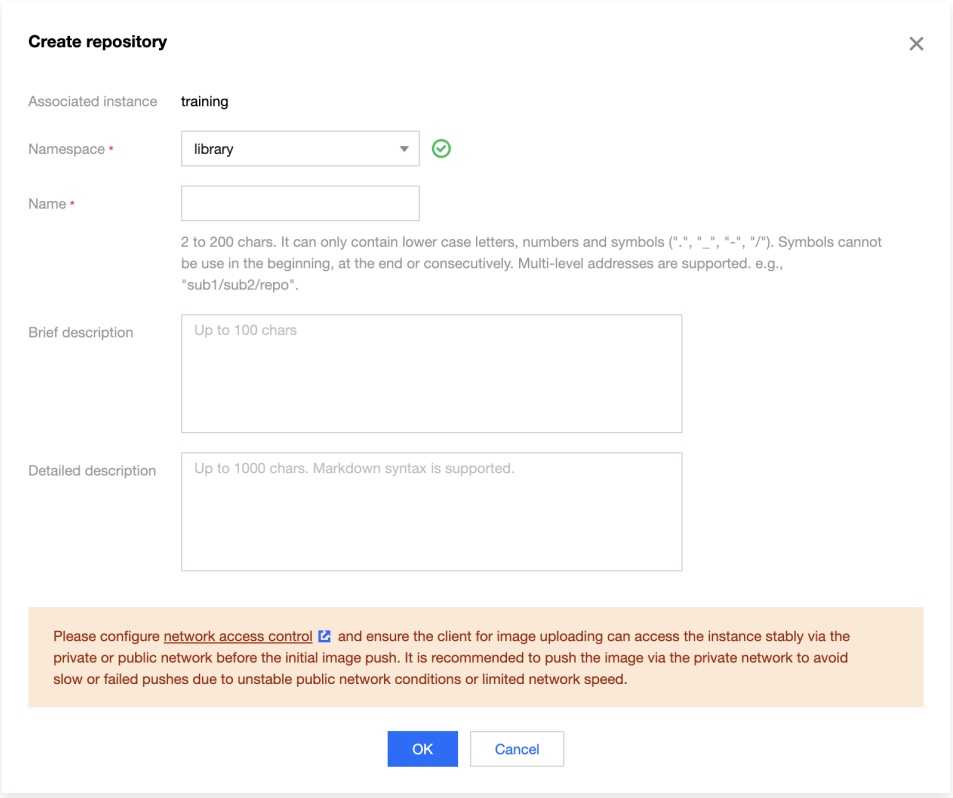
Make sure that the following conditions are met before creating and managing an image repository in a TCR Enterprise Edition instance:

- You have purchased an Enterprise Edition 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.

Instructions

Create an image repository

1. Log in to the Container Image Service console and select **Image Repository** from the left navigation bar.
   On the "Image Repository" page, you can view the list of image repositories in the current instance. If you need to switch instances, select from the "Instance Name" dropdown list at the top of the page.
2. Click **Create**. In the "Create repository" window, configure the image repository according to the following tips. As shown in the figure below:



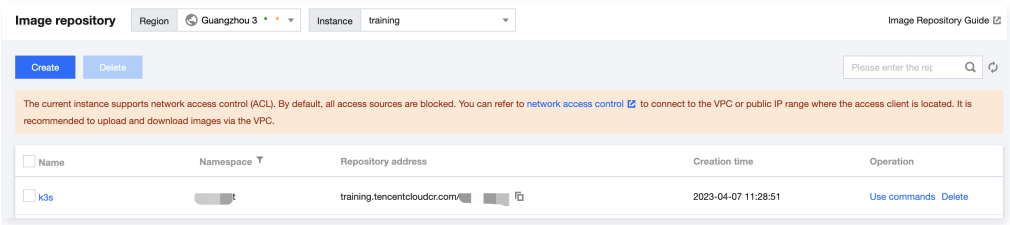
- **Associated Instance**: Currently selected instance, to which the new namespace will belong.

- **Namespace**: Namespace to which the image repository belongs. If the list is empty, create a namespace in the instance.
- **Name**: The image repository name must be 2 – 200 characters in length and can only contain lowercase letters, numbers, and separators (`.` , `_` , `-` , `/`). It cannot start or end with a separator or have consecutive separators. The name supports multi-level paths, such as `team-01/front/nginx` , which can be flexibly set according to business needs.
- **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**: Provide a detailed description of the image repository. Supports Markdown, with a maximum length of 1000 characters. You can continue editing the description after the repository is created.

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

Basic image repository operations

Upon successful creation, you can view it on the "Image Repository" page, where you can perform the following operations to manage the image repository, as illustrated below:



- **Filtering namespaces**

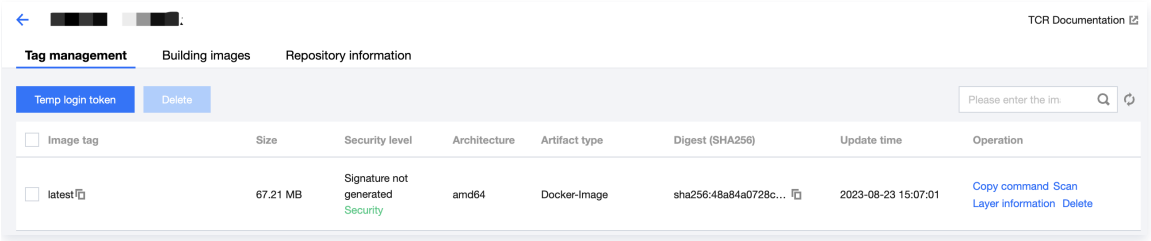
  In the "Image Repository" list, click Namespace to filter and select the desired namespace from the drop-down list.

- **Viewing image repository details**
  Click the name of the specified image repository to access its details page, where you can manage image versions and edit the basic information of the image repository.
- **Deleting the image repository**
  Click **Delete** next to the specified image repository to remove it. To avoid accidentally deleting important data, a second confirmation is required when deleting an image repository.

> **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:



- **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.
- **Perform Security Scan**
  To actively trigger a security scan, select **Scan** on the right side of the row where the specified image version is located. Once the scan result appears in the "Security Level" attribute, you can click to view the detailed results.

- **Viewing the image layer information**
  Click **Layer Information** next to the desired 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. For more information, see Configuring Image Building.

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.

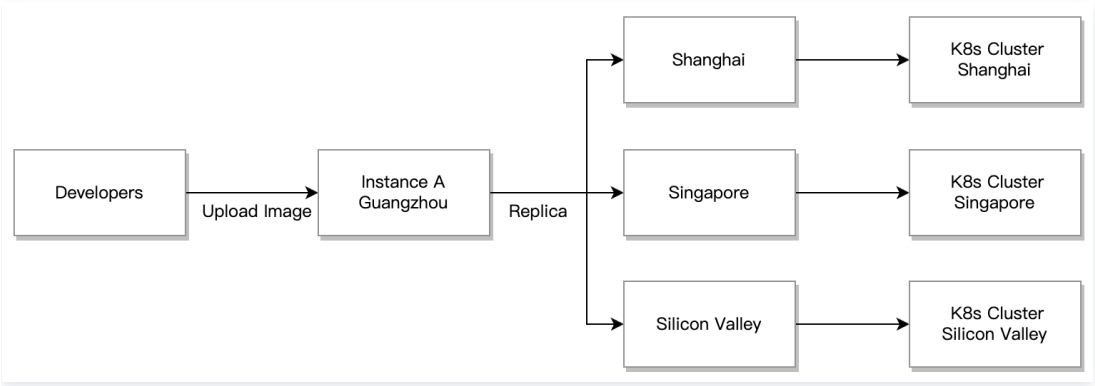
- **Edit Short Description**
  After selecting "Short Description", click on the to enter the editing mode. Once the changes are made, click on **Save** to finish editing.
- **Edit Detailed Description**
  After selecting "Detailed Description", click on the to enter the editing mode. Once the changes are made, click **Save** to finish editing. The detailed description supports Markdown syntax, and you can view the rendered text effect after saving.

Image Distribution
Intra-Instance Multi-Region Image Replication

Last updated: 2023-09-13 15:50:19

Scenario

Tencent Container Registry (TCR) supports the configuration of replication instances in other regions worldwide for advanced instances. With a unified domain name and access credentials, it enables single-region uploads, multi-region high-speed real-time synchronization, and nearby private network pulls. Compared to cross-instance synchronization, this feature unifies the deployment configuration across multiple regions and improves the cross-region synchronization speed of cloud-native artifacts, helping customers achieve synchronized updates for global business 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.



Preparations

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

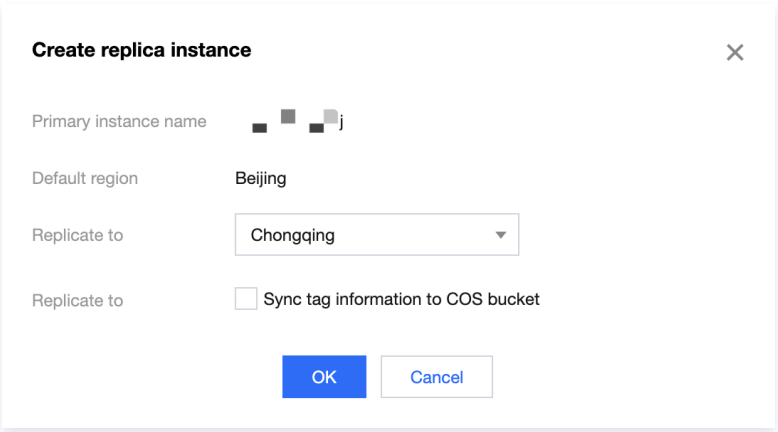
Usage Limits

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.

Instructions

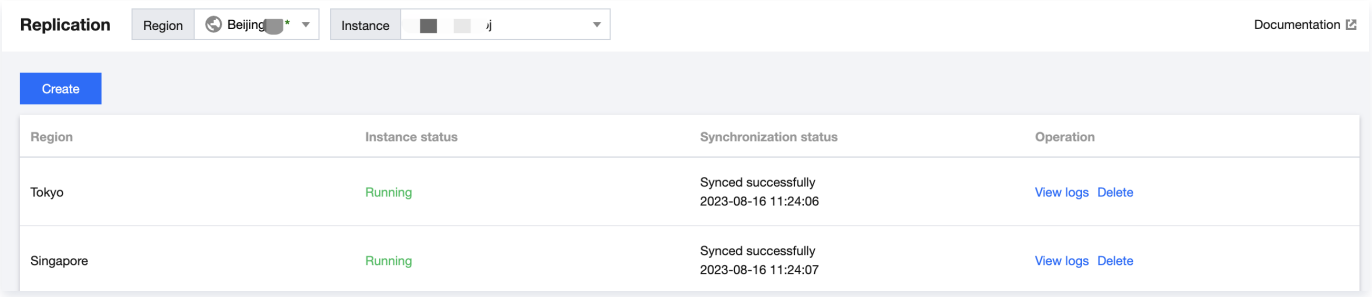
Creating and managing a replica instance

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



- **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** to create a replica instance.



> **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. To confirm the image synchronization status, you can select the specified replica instance and view the replication log.

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.

Documentation

You can also use the CreateReplicationInstance interface to create a replica instance. For more information, please refer to the Create Replica Instance API documentation.

Cross-Tenant Synchronization

Last updated: 2023-09-13 15:48:58

Scenario

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 users to create custom synchronization rules, specifying which resources within an instance should be synchronized to a designated location in another instance. For example, users can choose the type of resources to synchronize (container images, Helm Charts, or both), filter resource paths, use regular expressions to filter repositories and versions, and decide whether to overwrite existing images with the same name to prevent data loss. The instance synchronization feature now supports cross-master account synchronization, allowing users to create synchronization rules based on the target user's instance ID, account ID, and access credentials.

Preparations

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

- You have purchased an Enterprise Edition instance with either standard or premium specifications.
- 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.

Usage Limits

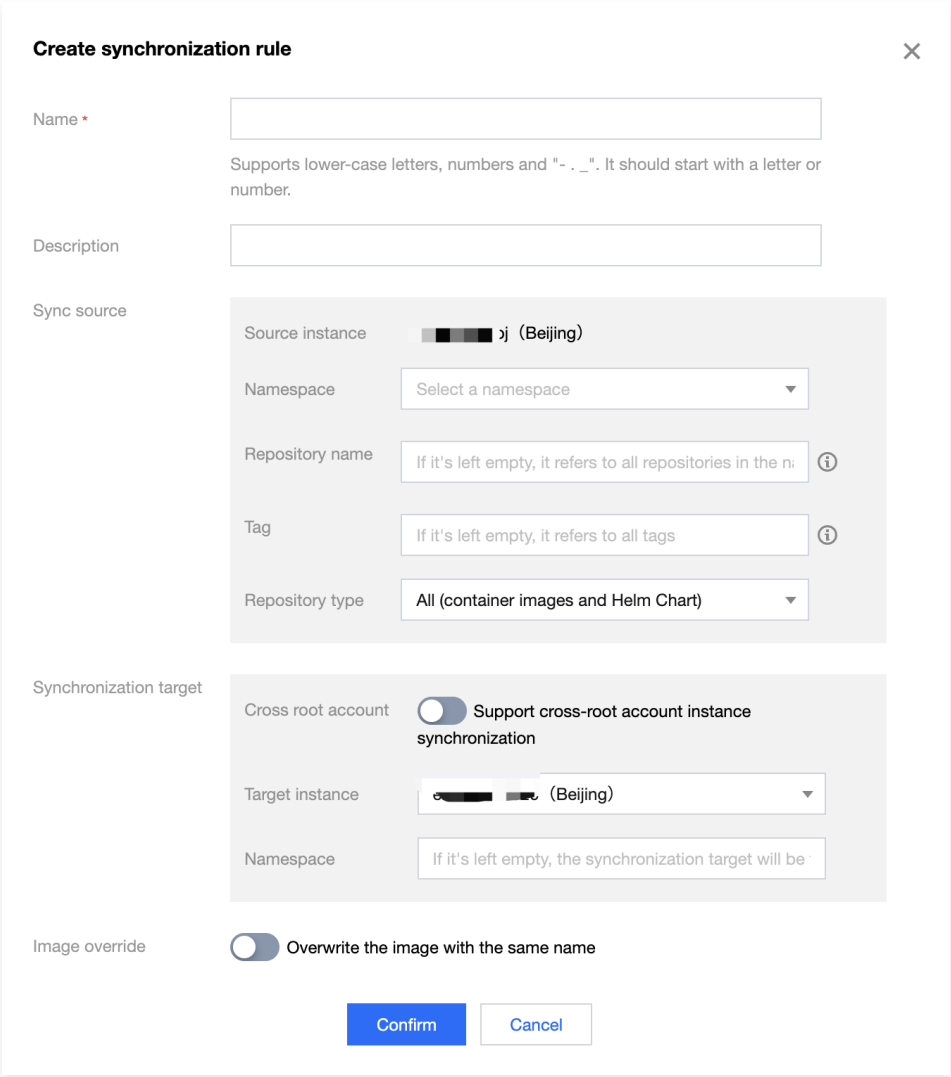
The instance replication feature relies on the support of the underlying networks and is subject to security compliance requirements.

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

Instructions

Creating a synchronization rule

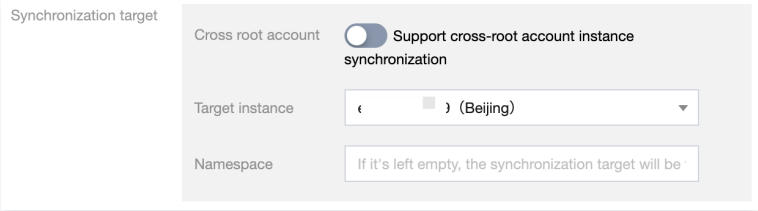
1. Log in to the TCR console and choose **Synchronization and Replication** > **Instance Synchronization** in the left sidebar.
2. In the "Instance Synchronization" page, select the region and instance name, then click on "Create New".
3. In the "Create synchronization rule" pop-up window, configure the rule as per the following instructions. Refer to the figure below:



- **Name**: Instance rule name, supports lowercase letters, digits, and three symbols (`-` , `.` , `_`), and must start with a letter or digit.
- **Description**: Rule description.
- **Sync source**:
  - **Source instance**: The currently selected instance is the source instance. You can return to the "Instance Synchronization" page to modify it.
  - **Namespace**: The namespace that needs to be synchronized within the current instance. Currently, selecting all namespaces is not supported.
  - **Repository name**: The repository to be synchronized. If left empty, all repositories within the namespace will be synchronized by default.
  - **Tag**: The version to be synchronized. If not specified, all versions in the repositories that meet the conditions will be synchronized by default.
  - **Repository type**: The type of resources to synchronize, which can be both container images and Helm Charts, or just one of them.
- **Synchronization target**: Choose whether to enable cross-master account instance synchronization.

Closed

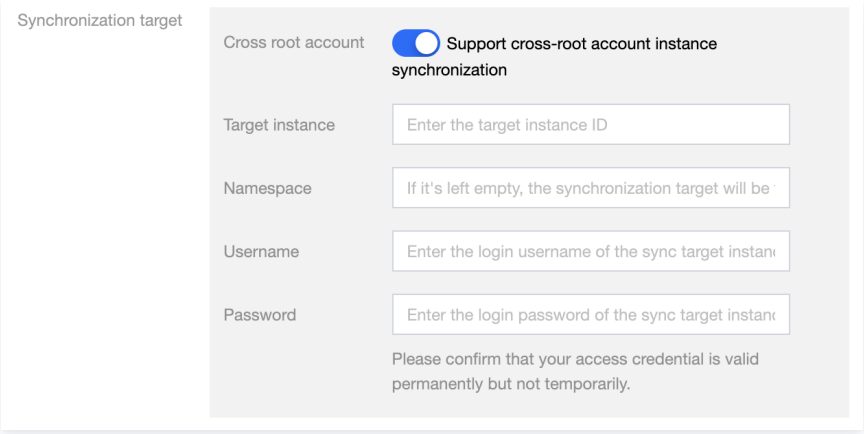
If **Cross root account** is disabled, the synchronization is between instances under the same tenant. You need to complete the following fields. shown below:



- **Target Instance**: The destination instance for data synchronization, which can be any instance within the master account.
- **Namespace**: The namespace where the repository will be located after synchronization to the target instance. If left empty, the default is the namespace with the same name as in the source instance. If there is no namespace, a new one will be created.

Enabled

If **Cross root account** is enabled, you can synchronize between instances under different tenants. The following fields are required.



- **Target instance**: The target instance ID for data synchronization can be obtained from the Instance Management page.
- **Namespace**: The namespace where the repository will be located after synchronization to the target instance. If left empty, the default is the namespace with the same name as in the source instance. If there is no namespace, a new one will be created.
- **Username**: The account ID of the synchronization target can be obtained by going to Account Information.
- **Password**: Access credentials provided by the target account. The target user can refer to Obtaining Instance Access Credentials for guidance.

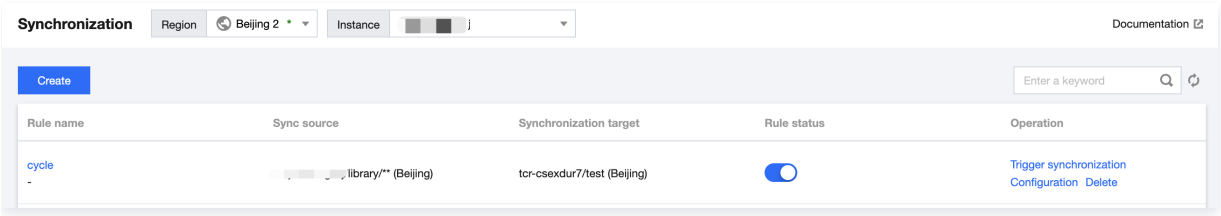
> **Note**
> 1. Please use a long-term access credential to avoid synchronization rule unavailability. The lifecycle of the synchronization rule is consistent with the lifecycle of the latest added access credential for the synchronization rule under the target account.
> 2. Please be aware of the risk of unauthorized access. The permissions of the access credentials are consistent with the permissions of the sub-account that created them. It is recommended that the target account creates a new sub-account dedicated to cross-master account instance synchronization and grants this sub-account only the permission to push images. For more information, see Example of Authorization Solution of TCR Enterprise.

- **Image overwrite**: You can choose whether to overwrite existing container images with the same name in the target instance. It is recommended not to overwrite.

4. Click **Confirm** to create the synchronization rule.

Managing synchronization rules

After successful creation, you can view the created synchronization rules on the "Instance Synchronization" page. You can manage the synchronization rules by performing the following operations, as shown in the figure below:



- **View Synchronization Logs**: You can click the instance rule name to view its triggering logs. For more information, see Viewing synchronization 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.
- **Trigger synchronization**: Manually trigger synchronization, scanning all repositories within the instance that meet the rules and performing synchronization.
- **Configuration**: You can reconfigure all parameters of the instance synchronization rule.
- **Delete**: Remove the instance synchronization rule.

Viewing synchronization logs

Click the name of the target instance synchronization rule to view the triggering logs of the rule.

- **Task ID**: Unique synchronization task ID within the instance.
- **Creation Time**: The time when the synchronization task was created.
- **Time Spent**: Time consumed to complete all synchronization tasks.
- **Success Rate**: The proportion of resources successfully synchronized, as a single synchronization task may involve multiple repositories.
- **Number of Repositories to Synchronize**: The number of repositories required for the current synchronization task.
- **Synchronization Status**: Status of task completion. If a single synchronization task requires a large number of container images and Helm Charts, the synchronization status may remain in the "InProgress" state for an extended period.

Documentation

You can also use the ManageReplication interface to manage instance synchronization. For more information, please refer to the Manage Instance Synchronization API documentation.

Loading Container Images on Demand

Last updated: 2023-09-13 15:46:33

Scenario

When deploying and updating business applications using container images, the traditional approach involves downloading the full container image data and decompressing it. This not only results in longer container startup times but may also cause significant network and disk read/write pressure due to large cluster sizes, leading to large-scale container deployments that do not meet expectations. In reality, only a portion of the container image data may be used during container startup.

TCR Enterprise Edition supports on-demand loading, allowing you to use the accelerated image version in your business deployments. This eliminates the need for full image data downloads and online decompression, significantly improving application distribution efficiency and providing an exceptional elastic experience. This article explains how to load container images on-demand.

Preparations

- 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 component is containerd with version 1.4.3. Existing clusters can modify the runtime configuration to containerd 1.4.3, and the nodes added after the adjustment will use this version by default.
  - The cluster operating system is Ubuntu, TencentOS, or CentOS. If using CentOS, you need to execute `yum install -y fuse` on the cluster nodes to install the fuse application.
- You have purchased an Enterprise Edition instance. The on-demand loading of container images is currently available only for **premium instances**.
- The container cluster's VPC is connected to the TCR Enterprise Edition instance, allowing cluster nodes to access images within the instance over the private network. For specific configuration, refer to 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 Service" window, please read the related prompts carefully.
   - Once image acceleration is enabled, a new OCI format 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 "New Image Acceleration Rule" window, configure the rule according to the following instructions.
   - **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, supporting regular expression filtering. If not specified, all repositories within

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.

- **Validation Rule**: Enter the address of the image to be accelerated to verify whether the image under the current rule meets the acceleration criteria.

2. Click **OK** to add an image acceleration rule for the current instance.

Pushing the image and automatically converting it

After successfully adding a rule, you can view the added image acceleration rule on the "Image Acceleration" page. Ensure that the rule status is enabled, and push a new container image to the image repository that meets the rule. This will automatically trigger the accelerated image format conversion, generating an accelerated image with the -apparate suffix. The default image artifact type is Docker-Image, and the converted 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 image acceleration labels via the command line

Run the following command to add the image acceleration label to a cluster node:

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

Adding an image acceleration label through the console

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, select **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" pop-up window, set the Label to `cloud.tencent.com/apparate=true` and click **Submit**.

Installing the acceleration application

By default, clusters do not support the use of accelerated images. You need to install the TCR Acceleration Suite application in the cluster. After installing the TCR Acceleration Suite application, nodes marked to support accelerated image deployment will automatically deploy the daemonset and can load accelerated images normally.

When installing the TCR Acceleration Suite application and adding new nodes with the cloud.tencent.com/apparate=true label, the nodes will also automatically deploy the daemonset and can deploy accelerated images normally.

Installing the TCR acceleration suite application via 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 accelerated images. Please modify the dockerUsername, dockerPassword, and dockerServer fields, filling in the Enterprise Edition instance's username, password, and access domain, respectively.

3.2 Image: Keep the default setting, which is used for pulling basic images when installing applications in the cluster. If the cluster is deployed outside of mainland China, you can change it to the access domain name of the Personal Edition 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 executing helm install, you need to pre-configure the cluster's access credentials locally. For more information, refer to: Connecting the Local Helm Client to the Cluster.

5. Go to the Cluster Applications 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.

Hence, when creating a workload, please select the accelerated image version and add a nodeSelector with the value `cloud.tencent.com/apparate=true`. This will ensure that the workload is scheduled on nodes that support accelerated images, enabling faster startup.

FAQs Overview

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?

First, 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 consult Smart Customer Service or Online Consultation for help.

Image security
Container Image Security Scanning

Last updated: 2023-09-13 15:45:29

Scenario

Tencent Container Registry (TCR) Enterprise Edition supports security scanning of hosted container images, generating scan reports, exposing potential security vulnerabilities within container images, and providing remediation suggestions. Container image security is a crucial aspect of cloud-native application delivery. Timely security scanning of uploaded container images and blocking application deployment based on scan results can effectively reduce vulnerability risks in production environments.
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.

Preparations

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

- You have purchased an 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.

Instructions

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 **Basic Information** page, set the security scanning configuration to **Automatic Scanning**.

Manually triggering scanning

Step 1: Preparing 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.

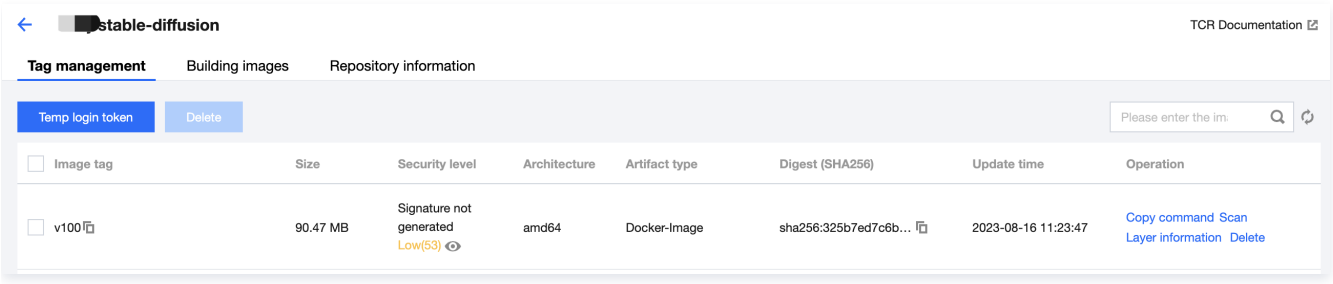
Step 2: Trigger image scanning

Select the specified image version within the image repository and click **Scan** to trigger the image scan. The security level will then display as "Scanning".

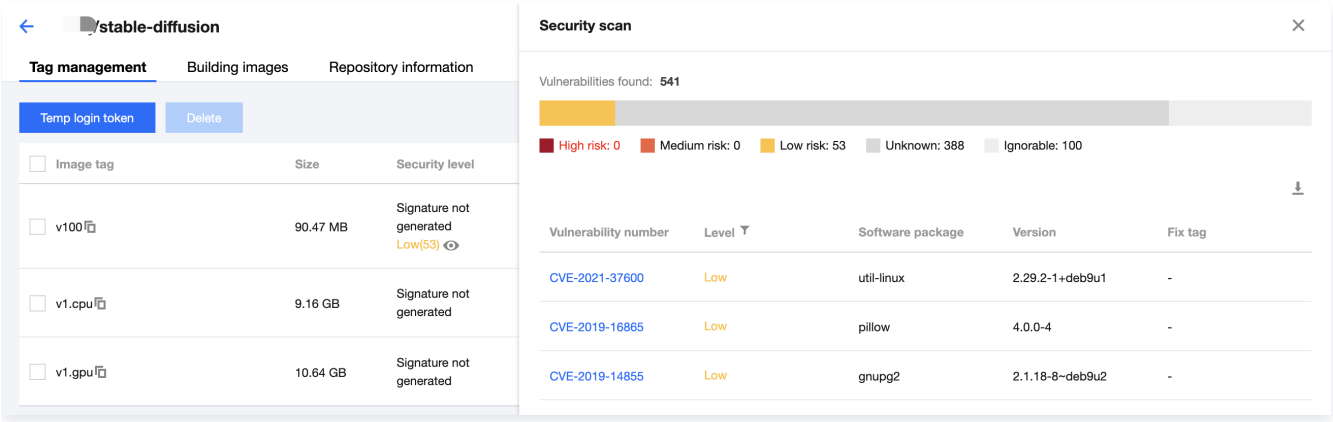
> ⚠ **Note:**
> The image scanning feature is only available for images with the artifact type DockerImage; other artifact types are not currently supported for image scanning.

Step 3: Viewing 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:



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:



Step 4: Re-triggering 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: 2023-09-13 15:44:23

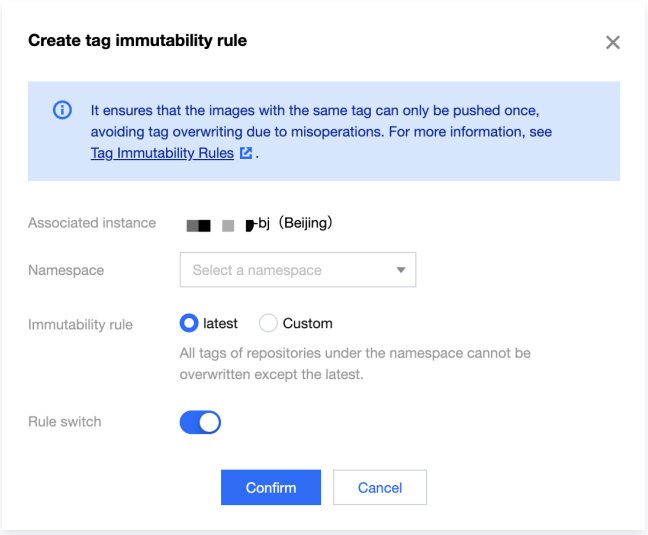
Scenario

Tencent Container Registry (TCR) Enterprise Edition supports protecting hosted container image tags. Container image security is a crucial aspect of cloud-native application delivery. Enabling the immutability feature for images hosted in TCR ensures that the same image tag is pushed only once, effectively reducing the risk of version overwrite caused by inadvertent operations in production environments. TCR supports namespace-level tag protection, allowing users to define the granularity of repositories and image tags covered by this feature based on their business requirements.

Instructions

Creating tag immutability rule

1. Log in to the TCR console and choose **Version Management** > **Tag Immutability** in 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 according to the following guidelines. As shown in the figure below:

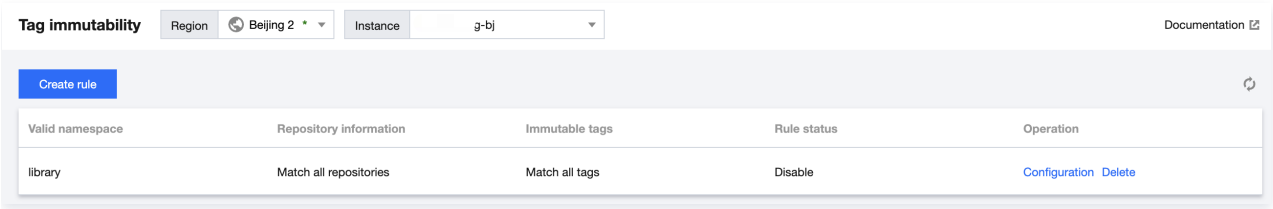


Configuration items	Configuration Notes
Associated instance	The instance which has been selected currently.
Namespace	Select the namespace for which you want to enable tag protection in the current instance. Only one rule can be created per namespace.
Immutability rule	latest: in all repositories in the current namespace, all image tags are not allowed to be overwritten except the latest tag.
	Custom: Configure the repositories and image tags to be matched according to your requirements. Repository Matching: Select the filtering type for the image repository and enter the repository name to be filtered. Tag Matching: Select the filter type for image tags and enter the tag names to be filtered.
Rule switch	The rule is effective as of creation by default. Enabling means the rule takes effect. You can enable/disable the rule in the configuration.

4. 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. shown below:



- **Configuration**: You can reconfigure a tag immutability rule but cannot modify the namespace for which it takes effect.
- **Delete**: Remove the tag immutability rule for this instance.

Blocking the Deployment of High-Risk Images

Last updated: 2023-09-13 15:43:03

Scenario

Tencent Container Registry (TCR) Enterprise Edition supports security scanning of hosted container images, generating scan reports, exposing potential security vulnerabilities within container images, and providing remediation suggestions. Container image security is a crucial aspect of cloud-native application delivery. Timely security scanning of uploaded container images and blocking application deployment based on scan results can effectively reduce vulnerability risks in production environments.
The image deployment blocking feature is built-in at the namespace level, allowing you to enable this feature and configure blocking rules and ignorable image vulnerabilities. Once enabled, if a container client attempts to pull a container image that meets the blocking policy, the action will be blocked and an error message will be returned.

Preparations

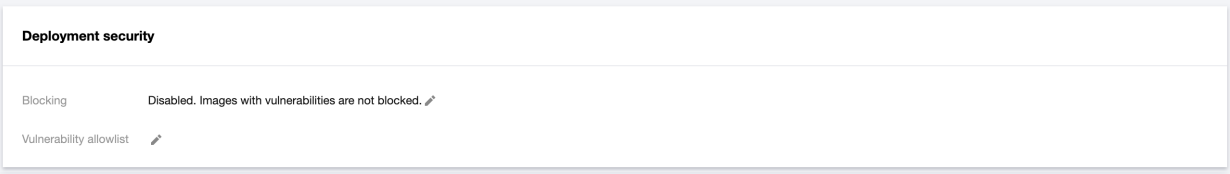
Before using the image deployment blocking feature, you need to perform the following operations:

- You have purchased an Enterprise Edition 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.

Instructions

Configuring the blocking 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 configure the blocking policy to go to the namespace details page.
3. On the "Deployment security" page, enable the deployment blocking feature and configure the vulnerability levels to be blocked, as shown in the figure below:



Configure the allowlist of vulnerabilities

After enabling deployment blocking, you can configure a vulnerability allowlist by entering one or more CVE IDs separated by commas. If the image security scan results include the specified vulnerability ID, it will be ignored by the blocking policy. For example, if an image has a high-risk vulnerability, but the vulnerability is on the allowlist, the image can still be pulled normally even if the policy is set to block images with high-risk vulnerabilities.

Container image signature

Last updated: 2023-09-13 15:41:55

Image signature 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 Edition supports namespace-level automatic image signature. When an image is pushed to the registry, it will be automatically signed according to the matched signature policy to ensure image content trustworthiness in your registry.

Preparations

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

- You have purchased an Enterprise Edition 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.
- You have activated the Key Management Service.

Instructions

Creating an asymmetric signature verification key

1. Log in to the Key Management System (Compliance) console.
2. In **Key Management > User Key**, click **Create**.
3. In the "Create Key" pop-up window, configure the key parameters and click **Confirm**. The container signature feature requires the KMS key purpose to be **"Asymmetric Signature Verification"** and the encryption algorithm to be **"RSA_2048"**. For other parameter configurations, please refer to Create Key.

> **Note**
> TCR supports obtaining user keys from KMS services across all regions. To reduce communication overhead between regions, it is recommended that the KMS user key and the image repository instance are located in the same region.

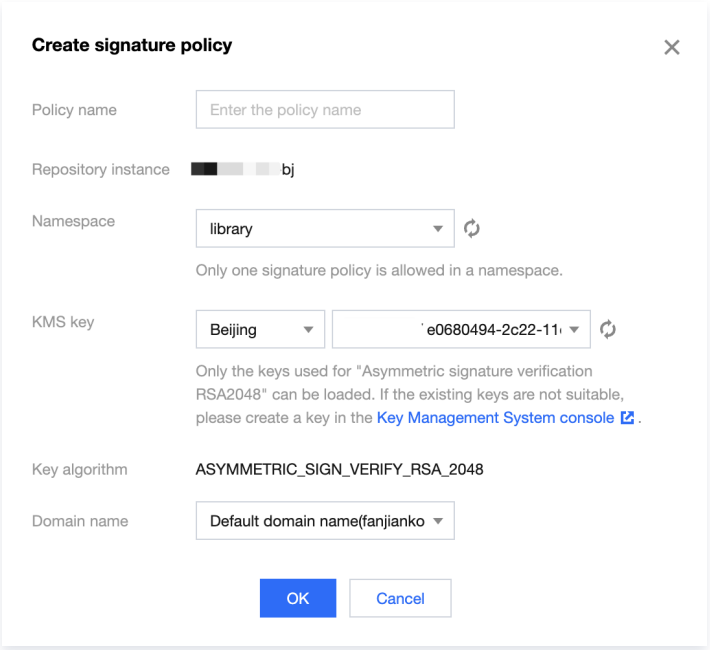
Authorizing TCR to use the KMS key

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 Cloud Access Management console.
2. On the "Roles" page, click **TCR_QCSRole**.
3. On the TCR_QCSRole role details page, associate the preset policy **QcloudKMSFullAccess.**

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. Click **Image Security** in the left sidebar to enter the "Image Signature" details page.
4. Click **Create**. In the "Create signature policy" pop-up window, fill in the following information:



- **Policy name**: The image signature policy name must be 2–50 characters in length and can contain only lowercase letters, digits, and any of the following separators: periods (.), underscores (_), hyphens (-), and slashes (/). The name cannot start or end with a separator, nor have consecutive separators.
- **Namespace**: Namespace for which the image signature policy will take effect. Only one signature policy is allowed in a namespace.
- **KMS Key**: A KMS user key that supports signature operations. Only keys used for "Asymmetric signature verification RSA2048" purposes are supported.
- **Domain name**: Used to access the repository instance service.

5. Click **OK** to complete the creation of the signature policy.

> **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.
> - The signature policy does not apply to existing images in the repository. You need to manually trigger image signing in **Image Repository > Version Management**.

Viewing image signing status

- You can check whether the signing policy is enabled on the **Namespace** page.
- You can check whether the image signing policy is enabled on the **Image Repository > Tag Management** page. For images pushed to the repository before enabling the signing policy, you can manually trigger the signing process under "Actions".

Deleting an image signing policy

On the "Image Signature" page, select the signature policy you want to delete and click **Delete**. In the "Delete Signature Policy" pop-up window, click **Confirm**.

> **Note**
> Deleting a signature policy will also remove the image signature information in the existing namespace, which may result in signature verification failure. Please proceed with caution.

Documentation

You can use the enhanced add-on in TKE clusters for automatic signature verification and set policies to block image deployment when verification fails. For more information, see Container Image Signature Verification.

Image Cleanup
Releasing COS Storage Capacity

Last updated: 2023-09-13 15:40:38

Scenario

Tencent Container Registry (TCR) supports setting custom rules to batch clean image tags within Enterprise Edition instances. However, after deleting image tags, the image data stored in the associated Object Storage COS is still retained. You can use the artifact cleanup feature to delete the invalid image tag-related data in Object Storage COS, freeing up COS storage space and reducing storage costs. This article will guide you on how to use the artifact cleanup feature to clean up COS storage space.

Supports and Limits

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 the image repositories, but cannot push images to image repositories.
- The garbage collection task will clean up the image layer data in all image repositories of the instance that is no longer associated with valid image tags. The deletion operation is irreversible. We recommend that you perform a dry run to evaluate the impacts before performing the 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. The 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.

Instructions

Dry run

1. Log in to the TCR console and click **Garbage Collection** in the left sidebar.
2. On the "Artifact Cleanup" page, you can view the list of artifact cleanup tasks for the current instance. To switch instances, select the desired 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** to perform a dry run of the cleanup task.

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** to run the cleanup task immediately.

Auto-Deleting Image Tags

Last updated: 2023-09-13 15:40:18

Scenario

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. Additionally, in advanced configuration mode, you can filter repositories and tags, and combine them with the two retention policies to achieve more flexible and precise tag management.

Supports and Limits

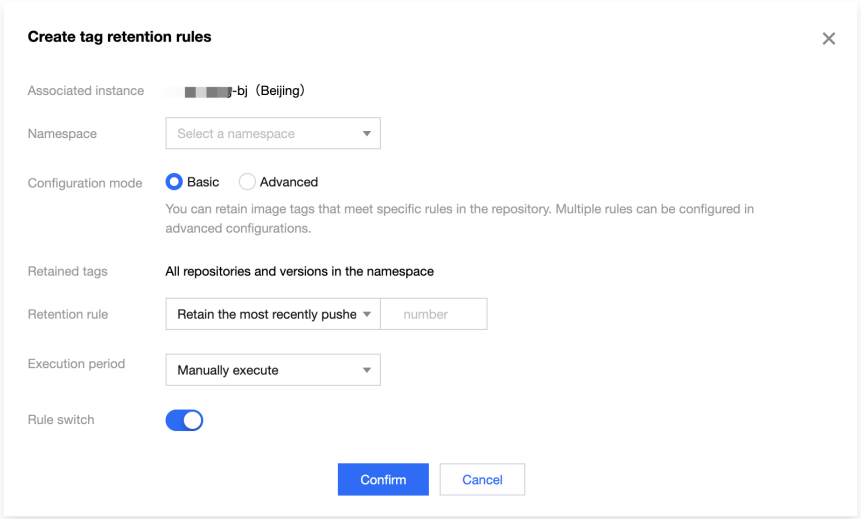
1. The tag retention feature will delete image tags that fall outside the retention rules.
2. The tag retention feature only deletes image tag information, but not the underlying image data. To thoroughly clean up image data, please use the garbage collection feature for COS storage.
3. The advanced configuration mode is currently in beta. To use the advanced configuration mode, please submit a ticket.

Instructions

Creating tag retention rules

1. Log in to the Container Registry console and select **Tag Retention** from the left navigation bar. On the "Tag Retention" page, you can view the list of tag retention rules within the current instance. If you need to switch instances, please select from the "Instance Name" dropdown list at the top of the page.
2. Click on **Create Rule**, in the "Create tag retention rules" window, refer to the following prompts for rule configuration. You can choose between two configuration modes:

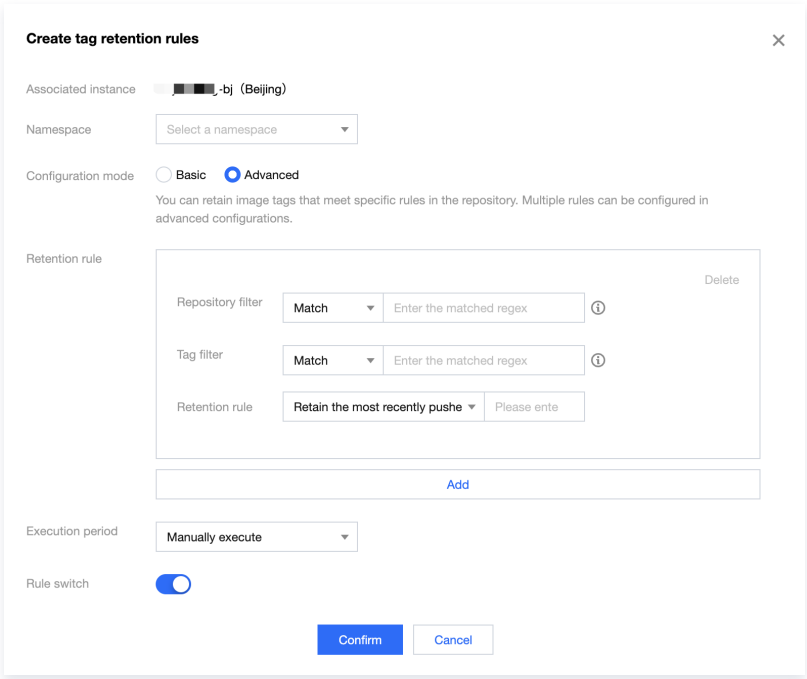
Basic Configurations



- 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.
- Configuration mode: Basic Configuration

- Retained tags: By default, all repositories and tags in the namespace are retained and no filter is applied.
- Retention rules: You can choose between retaining the most recently pushed # tags and retaining the tags pushed within the last # days, and specify the number of tags or days accordingly.
- Execution period: The cycle for implementing the tag retention rule, which supports manual execution, or daily, weekly, and monthly execution. The specific execution times are as follows: daily execution is at midnight, weekly execution defaults to midnight on Monday, and monthly execution is at midnight on the first day of the month.
- Rule switch: By default, the rule is enabled.

Advanced Configurations

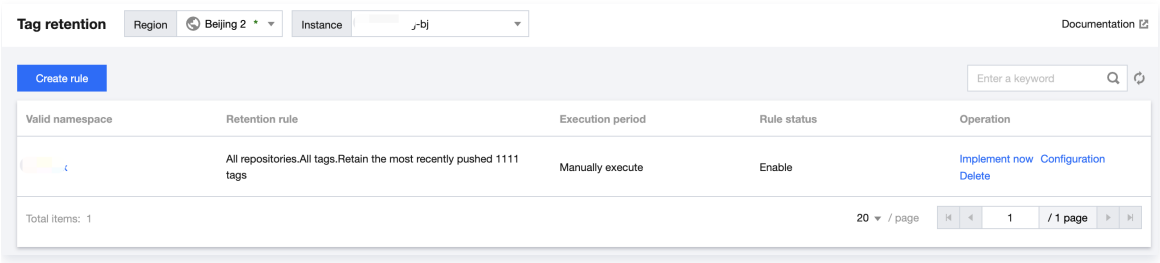


- Associated instance: Currently selected instance.
- Namespace: Namespace for which the tag retention rule will take effect. Currently, only one rule can be created for a single namespace.
- Configuration mode: Advanced Configuration
- Retention rules: Multiple retention rules can be configured, and the rules are combined using the union operation. That is, an image tag will be retained if it meets any of the retention rules.
  - Repository filter: Supports matching or excluding specific repositories by using regex to filter repository names. "*" and "" can match any length of strings, but "* does not support matching multi-level repository names. "?" matches any single character except "/". For more matching rules and scenarios, click here.
  - Tag filter: Supports matching or excluding specific tags within a repository by using regular expressions to filter tag names, following the same rules as repository filtering.
  - Retention rule: You can choose between retaining the most recently pushed # tags and retaining the 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

Once the tag retention rule is successfully created, you can view the created rules on the "Tag retention" page. You can perform the following operations to manage the tag retention rules, as shown in the figure below:



- **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.
- **Configuration**: You can reconfigure a tag retention rule but cannot modify the namespace for which it takes effect.
- **Delete**: Remove the tag retention rule for the instance.

Viewing execution logs

1. Click the name of the target tag retention rule to view the triggering logs of the rule:
   - **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. Clicking "Execute Now" or "Simulate Execution" is the manual method, while automatic execution is performed according to the cycle defined by the 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.

Documentation

You can also use the CreateTagRetentionRule interface to create tag retention rules. For more information, please refer to the Create Tag Retention Rule API documentation.

DevOps
Managing Triggers

Last updated: 2023-09-13 15:37:28

Scenario

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.

Preparations

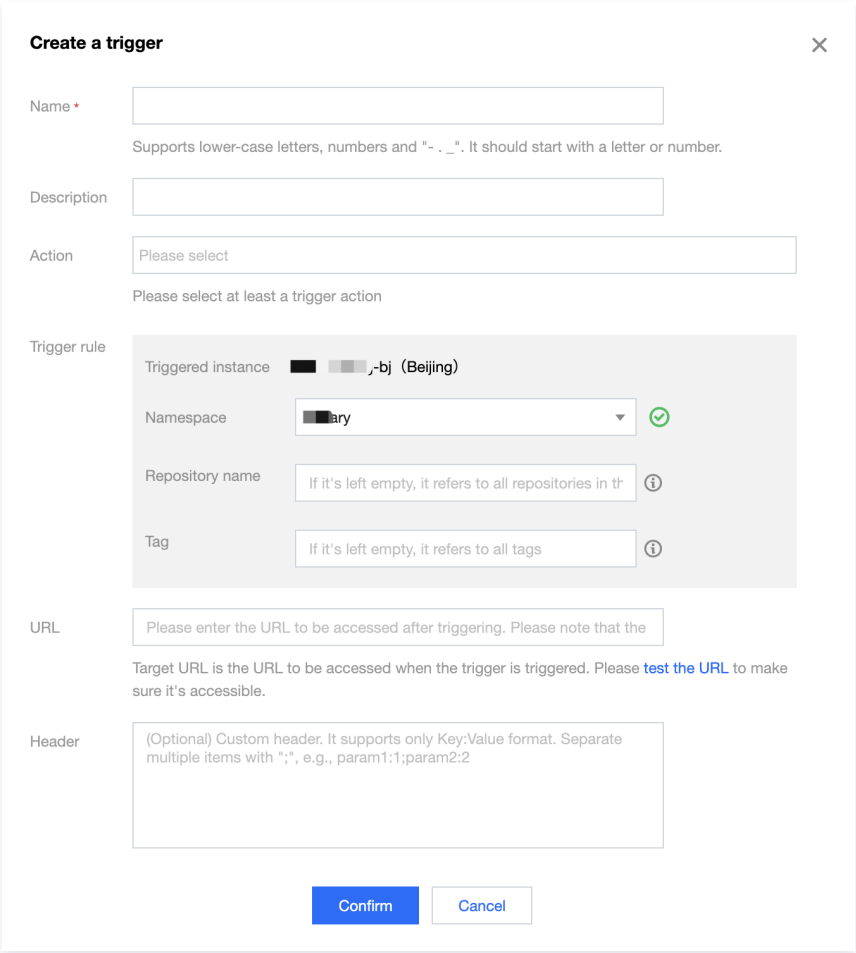
Before creating and managing a trigger in a TCR Enterprise Edition instance, complete the following tasks:

- You have purchased a TCR Enterprise Edition instance. This feature is only applicable to Enterprise Edition 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 TCR Enterprise.

Instructions

Creating Trigger

1. Log in to the Container Registry Service console and select **Trigger** from the left navigation bar. On the "Trigger" page, you can view the list of trigger rules within the current instance. If you need to switch instances, select from the "Instance Name" dropdown list at the top of the page.
2. Click **New**. In the "Create a trigger" window that appears, configure the rules according to the following prompts, as shown in the figure below:

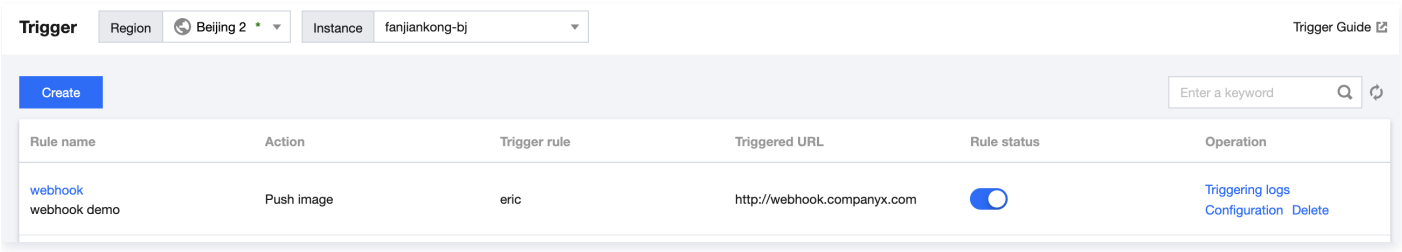


- **Name**: Instance rule name, supporting lowercase letters, digits, and three symbols (`-` , `.` , `_`), and must start with a letter or digit. In this example, we use `webhook-demo` .
- **Description**: Rule description.
- **Action**: Four trigger actions are supported: push images, delete images, upload Helm charts, and delete Helm charts. During the execution of the webhook, the initiated webhook request contains information about the trigger action.
- **Trigger rule**:
  - **Triggered instance**: The instance to which the trigger belongs, which is the currently selected instance and cannot be changed.
  - **Namespace**: The namespace for which the trigger 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 **Confirm** to create the synchronization rule.

Managing trigger rules

Upon successful creation, you can view the created trigger rules on the "Triggers" page. You can perform the following operations to manage the trigger rules, as shown in the figure below:



- **View Triggering Logs**: Click the trigger rule name or **Triggering Logs** on the right side of the trigger rule name to view the trigger logs for that 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**: Remove the trigger rule.

Viewing trigger logs

Click the name of a specific trigger rule or click **Trigger Logs** on the right side of the trigger rule to view the trigger logs of the rule. The information includes:

- **Task ID**: Unique trigger task ID within the instance.
- **Action**: The action that initiated the trigger, such as pushing an image.
- **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.

Relevant 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": "sha256:89a42c3ba15f09a3fbe39856bddacdf9e94cd03df7403cad4fc105xxxx268fc9",
     "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"
}
```

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**`, its effect will be equivalent to `/path*`, and it can only match first-level repositories with the "path" prefix. To match all repositories under "path", use `/path/**`; to match all repositories with the "path" prefix, use `/path*/**`.

- `?`: matches any single character except ‘/’.
- `{alt1, alt2, ...}`: matches multiple regular expressions at the same time.

Typical Use Cases

Matches all repositories in the specified namespace.	** or leave it blank
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*}

Automatic Image Deployment

Last updated: 2023-09-13 15:36:40

Scenario

Tencent Container Registry (TCR) offers image building and delivery pipeline features based on Tencent Cloud CODING DevOps, meeting the needs of container users for quick configuration and application of continuous integration and continuous deployment. For more flexible and powerful continuous compilation, building, and delivery pipeline features, please consider using Container DevOps.
Currently, both the TCR Enterprise Edition and Individual Edition services support the delivery pipeline feature, which is used to automatically deploy container images to specified container clusters. This feature allows you to specify an image repository and, when a new image is pushed into this repository, it filters the image versions that meet the deployment rules and automatically deploys them to the container cluster. It currently supports deployment to Container Service TKE, Serverless Container Service, and Edge Container Service. For a quick start on this feature, please refer to the best practice document Implementing Container DevOps with Delivery Pipeline.

Preparations

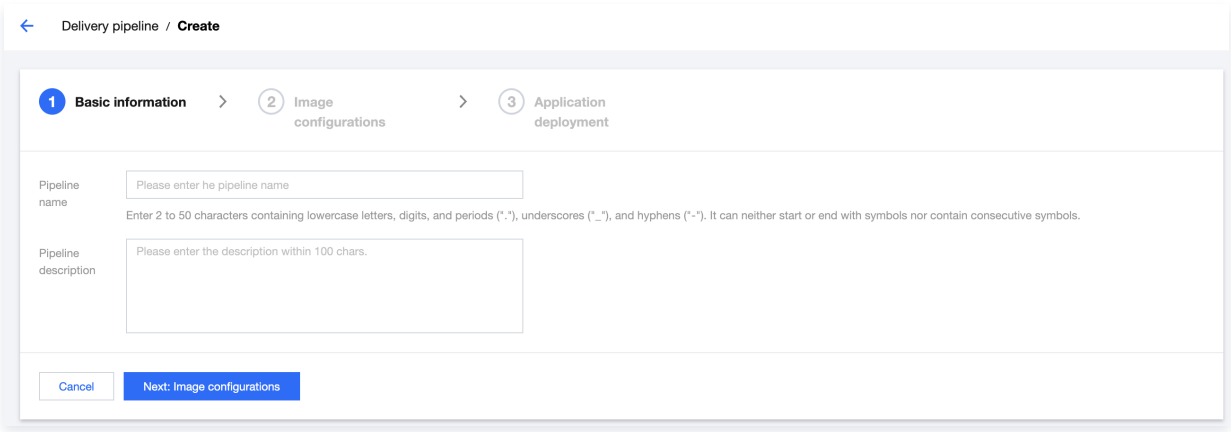
Before configuring automatic image building, make sure that the following conditions are met:

- You have purchased an Enterprise Edition instance or initialized an Individual Edition instance.
- You have activated the CODING DevOps service and completed the authorization process. For more information, please refer to Activating the Service.
- If you are using a sub-account, please refer to Enterprise Edition Authorization Scheme Example or Individual Edition Authorization Scheme Example to grant the sub-account the necessary operation permissions for the corresponding instance.

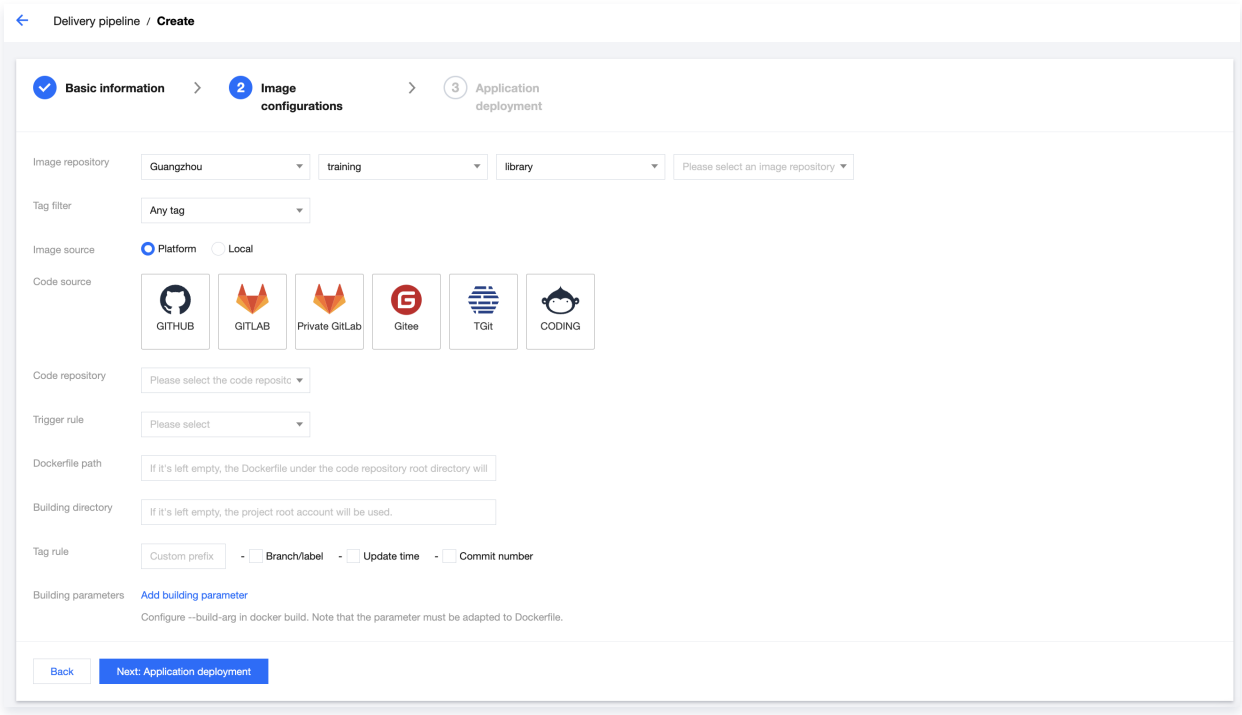
Instructions

Create Delivery Pipeline

1. Log in to the Container Image Service Console and select **Delivery pipeline** from the left navigation bar.
   On the "Delivery pipeline" page, you can view the list of delivery pipelines within the current instance. If you need to switch instances, please select from the "Instance Name" dropdown list at the top of the page.
2. Click **Create** and, on the "Delivery pipeline" page, follow the instructions provided for configuration.
   - **Basic information**: Configure the pipeline name and description. The description supports Chinese input. This information can be edited after creation.



   - **Image configurations**: Set up the bound image repository and the image filtering rules for deployment.



- Image repository: Choose an existing image repository within either an Enterprise Edition or Individual Edition instance.
- Tag filter: Specify the image to be deployed. Options include "Deploy any tag", "Deploy only specified tag names", and "Deploy only tags of the specified rule". The specified rule should be a regular expression input.
- Image source: You can choose between platform-built images or locally pushed images. If the image repository has not yet been configured for automatic image building, you can opt for platform-built images. If you prefer to use a self-built CI service for image building or manually package images, you can select locally pushed images.

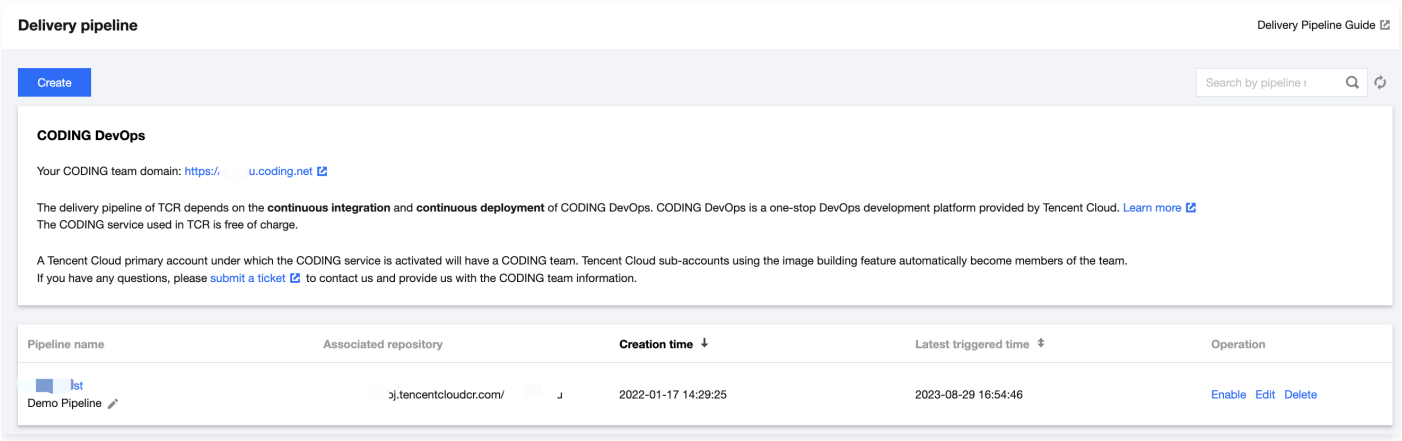
- **Application deployment**: Set up the image deployment environment.
  - Deployment platform: Supports Container Service TKE, Serverless Container Service, and Edge Container Service TKE Edge.
  - Deployment Region: The region where the cluster is located.
  - Deployment Cluster: Select the target cluster.
  - Namespace: The namespace within the cluster.
  - Workload types: Supports Deployment, StatefulSet, and DaemonSet.
  - Workload: Select an existing workload in the specified namespace. Creating a new workload is not supported at the moment.
  - Pod Container: The specified container within the workload will have its image updated.

3. Click **OK** to create the image build rule.

If the above configuration parameters do not meet your requirements, please go directly to the CODING DevOps platform and use the **Continuous Build** feature.

Managing Delivery Pipeline

After completing the creation of the delivery pipeline, you can view the existing pipelines in the **Delivery pipeline** list page. By selecting a specific pipeline, you can perform the following management operations on the current delivery pipeline:



1. List page
   - **Enable**: Manually trigger the pipeline deployment and select a specific image tag.
   - **Edit**: Edit the pipeline configuration.
   - **Delete**: Remove the pipeline.
2. Details page
   - **View Execution Records**
     Enter the "Execution Records" tab to view the current delivery pipeline execution records, detailed logs, and delete records.
   - **Viewing pipeline information**
     Navigate to the "Pipeline Information" tab to view the detailed information of the current delivery pipeline, including basic information, image configuration, and application deployment.

Trigger Deployment and View Details

After configuring the delivery pipeline, automatic image deployment will be triggered when an image push operation meets the trigger rules. You can also manually trigger deployment by selecting a specific image version. Once the deployment task is initiated, click on the delivery pipeline to access the execution record page, view the execution logs, and expand to see the execution details.

Troubleshooting

If you encounter any of the following issues while using the delivery pipeline feature, please refer to the corresponding methods to retry. If the issue persists, submit a ticket for consultation.

Image deployment failure

Please visit the container cluster to view the specific log information for this workload. If the error message indicates image pull failure, check whether the cluster has been properly configured to access the image repository and verify the access credential configuration. For more information, see TKE Clusters Use the TCR Addon to Enable Secret-free Pulling of Container Images via Private Network.

Automatic Image Building

Last updated: 2023-09-13 15:35:11

Scenario

Tencent Container Registry (TCR) offers image building and delivery pipeline features based on Tencent Cloud CODING DevOps, meeting the needs of container users for quick configuration and application of continuous integration and continuous deployment. For more flexible and powerful continuous compilation, building, and delivery pipeline features, please consider using Container DevOps.
Both TCR Enterprise Edition and Personal Edition services support image building features, with shared source code authorization information. The image building feature supports compiling and building source code hosted on GitHub, GitLab.com, private GitLab, Gitee, Tencent Work, and CODING. For users hosting code on CODING.net, it is recommended to configure image building directly within the CODING DevOps product using its "Continuous Integration" feature, which currently includes built-in templates for "Build Image and Push to TCR Enterprise/Personal Edition." TCR supports four types of trigger mechanisms, such as triggering image building by pushing to a specified branch. It allows custom configuration of Dockerfile paths and build directories. Flexible image version naming rules can be configured, such as custom prefix-branch-build time-commit number, to synchronize image version management with code updates.

Preparations

Before configuring automatic image building, make sure that the following conditions are met:

- You have purchased an Enterprise Edition instance or initialized an Individual Edition instance.
- You have activated the CODING DevOps service and completed the authorization process. For more information, please refer to Activating the Service.
- If you are using a sub-account, please refer to Enterprise Edition Authorization Scheme Example or Individual Edition Authorization Scheme Example to grant the sub-account the necessary operation permissions for the corresponding instance.

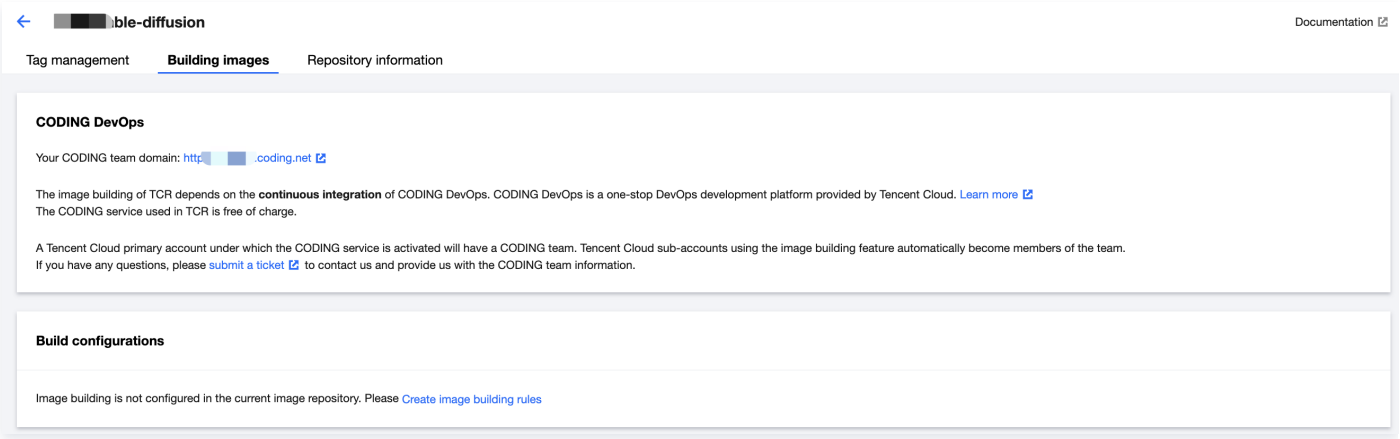
Instructions

Create an image repository

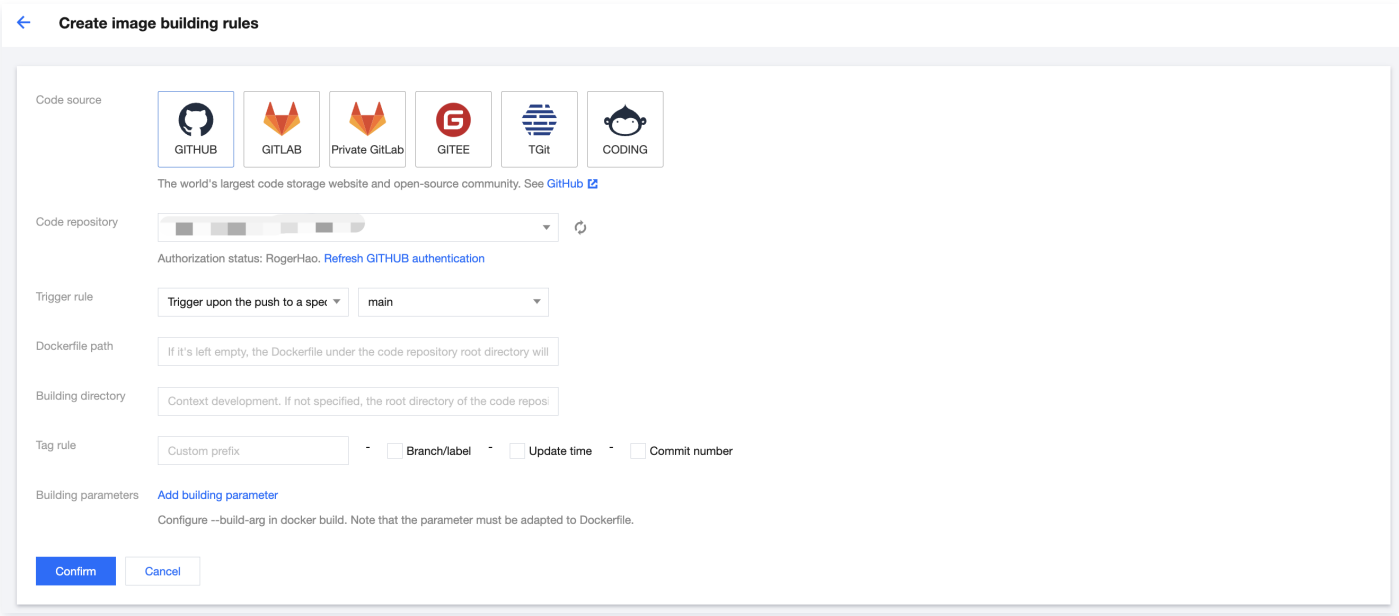
Refer to Create Image Repository to complete the creation of the image repository, selecting "Platform-built Image" as the "Image Source."

Creating Image Building Rules

1. After successfully creating an image repository, the page will automatically redirect to the **Building images** tab of the repository, as shown in the figure below:



2. Click "Create image building rules". In the "Create image building rules" pop-up window, refer to the following information to configure automatic image building and pushing for this image repository, as shown in the figure below:



- **Code source**: The hosting platform for the source code used in image building. Currently supported platforms include GITHUB, GITLAB, private GITLAB, GITEE, TGit, and CODING. It is recommended for CODING users to directly access the image building module within the CODING DevOps platform and use the built-in **Build Image and Push to TCR Enterprise Edition** or **Build Image and Push to TCR Personal Edition** templates to configure image building. The image building service needs to pull the corresponding source code, so please authorize the specified code source for the first-time use. After authorization, you can normally list code repositories, branches, and other information. If you encounter any issues with source code authorization, please submit a ticket for consultation.
- **Code repository**: After specifying the code source and completing its authorization, you can select a code repository under the source code hosting platform. When the image building service is triggered, it will pull the source code from this repository for compilation and building.
- **Trigger rule**: The rule conditions for automatically triggering image building. Currently, the following four scenarios are supported:
  - **Push to Specified Branch Trigger**: A specific branch must be designated.
  - **Trigger build when pushing a new tag**: Triggered when creating and pushing a new tag.
  - **Trigger build on branch push**: Triggered when pushing to any branch, without specifying a particular branch.
  - **Build when the branch or tag rule is matched**: You need to enter a regular expression, such as `^refs/heads/master$`, which can match the master branch to trigger the build.
- **Dockerfile path**: The image building operation is based on the Dockerfile within the code repository, and the path to this Dockerfile file must be specified. If not specified, the default is a file named Dockerfile in the root directory of the code repository.
- **Building directory**: The working directory for image building, also known as the context environment (context), which defaults to the root directory of the code repository.
- **Tag rule**: Define the name of the image generated by the image build, i.e., the image tag. Custom prefixes can be configured, combined with the three environment variables: "Branch/Tag," "Update Time," and "Commit Number." The update time is the system time of the build service when executing the docker tag command.
- **Building parameters**: Configure the --build-arg in docker build, ensuring compatibility with the Dockerfile.

3. Click **Confirm** to create the image build rule.

If the above configuration parameters do not meet your requirements, please go directly to the CODING DevOps platform and use the **Continuous Build** feature.

Managing Image Building Rules

After completing the creation of the image building rules, you can view the build configuration and build logs under the **Image Building** tab of the repository.
You can perform the following management operations on the current build rules:

- **Trigger Build Now**

Manually trigger image building within the console, specifying a branch or code version.

- **Edit**
  You can edit the current build rule, and all parameters that can be configured during the creation process can be re-edited.
- **Delete**
  Remove the current build rule.
- **Automatic Trigger**
  You can enable or disable the automatic trigger for this rule. When the automatic trigger is disabled, pushing code that meets the trigger criteria will not initiate image building, but you can still use the "Build Now" feature for manual triggering.

Triggering image building and viewing build logs

After configuring the image building rules, an automatic build and image push will be triggered when there is a source code operation that meets the trigger rules. This article takes pushing new code updates to a specified branch in a GitHub repository as an example.
For existing build logs, you can view the execution details or delete the log record.

Troubleshooting

If you encounter any of the following issues while using the image building feature, please refer to the corresponding methods to retry. If the issue persists, submit a ticket for assistance.

Source Code Authorization Failed

You can click "Authorize Source Code" again in the "Create Image Build Rule" window, or go to the corresponding source code hosting platform to view third-party authorization, delete "CODING DevOps," and try re-authorizing. For example, on GitHub, the third-party authorization location is: **Settings** > **Applications** > **Authorized OAuth Apps**.

Image Building Failure

You can view the execution details of the build task in the build log and make adjustments based on the error information in the execution details. For example, you can reconfigure the parameters in the image build configuration or modify the Dockerfile.

OCI Artifacts Management
OCI Artifacts Management Overview

Last updated: 2023-09-13 15:29:08

Scenario

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.
Both Enterprise and Individual Edition instances currently support hosting OCI artifacts. You can directly push OCI artifacts to the image repository and view the artifact type and obtain pull commands.
For more information on OCI artifacts and their usage, please refer to the official project on GitHub: opencontainers/artifacts.

Preparations

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

- You have purchased an Enterprise Edition instance or initialized an Individual 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.

Instructions

Managing OCI artifacts in the console

1. Log in to the TCR console and click **Image Repository** in the left sidebar.
2. On the "Image Repository" page, you can view the image repository list of the current instance, which supports hosting OCI artifacts by default. You can use a dedicated client tool for OCI artifacts to build and push them 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.

Documentation

Helm Charts Management

If you need to use Helm Chart, you can choose to push Helm Chart as an OCI artifact to the image repository for unified management. This management method requires the use of Helm V3 tool. Alternatively, you can directly use the Helm Chart hosting feature provided by the Enterprise Edition instance based on the Chart Museum open-source project. For more information, please refer to: Hosting Helm Chart.

Management Helm Chart

Last updated: 2023-09-13 15:04:25

Scenario

Tencent Container Registry (TCR) supports hosting Helm Charts, fulfilling users' needs for managing and distributing cloud-native applications. Users can simultaneously manage container images and Helm Charts within the same namespace, enabling the concurrent use of container images and Helm Chart cloud-native deliverables in business projects.
Currently, only TCR Enterprise Edition instances support hosting Helm Charts, allowing for uploading and downloading Charts using the console or Helm client. Helm Chart repositories inherit the public and private attributes of their respective namespaces without requiring additional configuration. In terms of permission management, Helm Charts share the **repository** resource type with container images. This means that the **qcs::tcr:$region:$account:repository/tcr-xxxxxx/project-a/*** resource description includes all image repositories and Helm Charts within the project-a namespace, providing flexibility when managing resource permissions.

Preparations

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

- You have purchased an Enterprise Edition 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.

Instructions

Managing Helm Charts in the console

1. Log in to the TCR console and click **Helm Chart** in the left sidebar.
2. You can view the list of Helm Charts in the current instance on the "Helm Chart" page. To switch instances, select from the "Instance Name" dropdown list at the top of the page.
   The Chart list includes the following information and 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 **Shortcuts** 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 remove 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.
   - **Basic Information**: This page displays the Chart version details, such as Chart.yaml.

Using the console to upload and download Helm Charts

Uploading a local Helm chart package

1. Log in to the Container Registry console and select **Helm Chart** from the left sidebar.
   On the "Helm Chart" page, you can view the list of Helm Chart repositories in the current instance. To switch instances, select the desired instance name from the "Instance Name" drop-down list at the top of the page.
2. Click **Upload**. In the "Upload Helm Chart" window, configure as per the following instructions.
   - **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 with the same name will overwrite the existing Chart, so proceed with caution.

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 within the Chart repository and click **Download** on the right side of the row. The Chart package for that version will be automatically downloaded to your local system. Depending on your browser and settings, you can choose a specific download path.

Using the Helm client to upload and download Helm Charts

Installing a Helm client

Note

- 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 platforms, please 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/scripts/get-helm-3
```

```
chmod 700 get_helm.sh
```

```
./get_helm.sh
```

Adding a Helm repository

1. Log in to the Container Registry Console, select the instance name from the "Instance List" page, and enter the instance 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 in the node to add the namespace used for Helm chart management 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://$instance.tencentcloudcr.com/chartrepo/$namespace --username $username
--password $instance-token
```

- `$instance-$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 using a version prior to v0.10.0, replace the helm cm-push command with the helm push command.

To upload Chart packages using the Helm CLI, you need to install the helm-push plugin. This plugin supports pushing Helm Charts to a specified repository using the `helm push` command and supports uploading both 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. Run the following command to directly push the specified directory to the Chart repository (optional).

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

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

4. Run the following command 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 on 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: 2023-09-13 15:02:54

Scenario

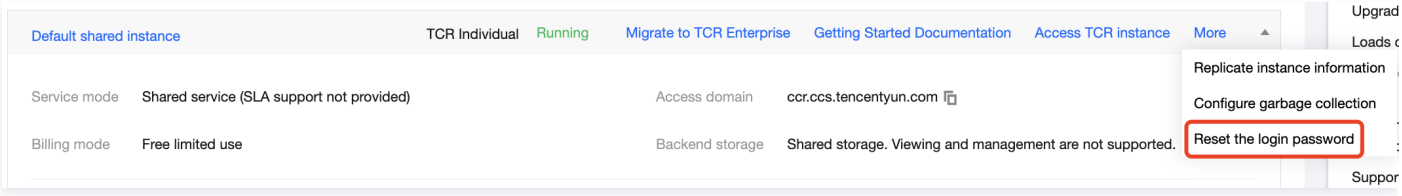
This document describes how to reset the login password for a TCR Individual instance.

Supports and Limits

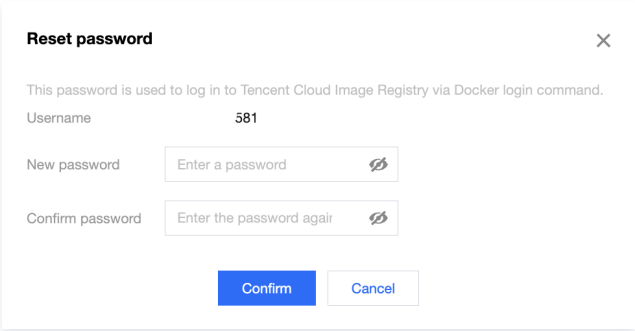
The login password for a TCR Individual instance is a fixed password, which is consistently applied among all regions.

Instructions

1. Log in to the Container Registry Service console and select **Instance Management** from the left navigation pane.
   On the "Instance Management" page, you can view the list of instances under the current account. Select a TCR Individual instance from any region.



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



   - **Username**: The Tencent Cloud account currently logged in.
   - **New password**: Set a new password, preferably a strong one.
   - **Confirm password**: Confirm the password again.
3. Click **Confirm** to reset the password.

Configuring Access Permission CAM APIs for Personal Edition

Last updated: 2023-09-13 15:02:06

Namespace APIs

APIs and Description	ResourceType	Six-segment Example of Resource
CreateNamespacePersonal Create Personal Namespace	repo	qcs::tcr:$region:$account:repo/$namespace
DeleteNamespacePersonal Delete Personal Namespace	repo	qcs::tcr:$region:$account:repo/$namespace

Image Repository APIs

APIs and Description	ResourceType	Six-segment Example of Resource
DescribeRepositoryOwnerPersonal Query All Personal Repositories	repo	qcs::tcr:$region:$account:repo/*
CreateRepositoryPersonal Create Personal Image Repository	repo	qcs::tcr:$region:$account:repo/$namespace/$repo
DeleteRepositoryPersonal Delete Personal Image Repository	repo	qcs::tcr:$region:$account:repo/$namespace/$repo
BatchDeleteRepositoryPersonal Batch Delete Personal Image Repositories	repo	qcs::tcr:$region:$account:repo/$namespace/*
DeleteImagePersonal Delete Personal Repository Tag	repo	qcs::tcr:$region:$account:repo/$namespace/$repo
BatchDeleteImagePersonal Batch Delete Personal Repository Tags	repo	qcs::tcr:$region:$account:repo/$namespace/$repo
PullRepositoryPersonal Pull Images from Personal Image Repository	repo	qcs::tcr:$region:$account:repo/$namespace/$repo
PushRepositoryPersonal Push Image to Personal Image Repository	repo	qcs::tcr:$region:$account:repo/$namespace/$repo

Example of Authorization Solution of TCR Individual

Last updated: 2023-09-13 15:01:37

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: 2023-09-13 15:00:56

Overview

Tencent Container Registry (TCR) provides container image hosting and distribution services to both enterprise customers and individual users. The Personal Edition offers simple, free basic services, which are the current image repositories within Tencent Kubernetes Engine (TKE).

To provide users with more standardized API interface definitions and significantly reduced access latency, the API interface of the original Personal Edition image repository (CCR) has been upgraded from version 2.0 to the latest version 3.0. The interface names and authorization schemes have also been updated accordingly. This document introduces the mapping relationship between the old and new interfaces after the API upgrade, which supports resource-level authentication, and how to use the latest authorization scheme.

Mappings Between the v2.0 and v3.0 APIs

API v2.0 Name	API v3.0 Name	API Description	Latest Resource Description Method
CreateCCRNamespace	CreateNamespacePersonal	Create Personal Namespace	qcs::tcr:$region:$account:repo/$namespace
DeleteUserNamespace	DeleteNamespacePersonal	Delete Personal Namespace	qcs::tcr:$region:$account:repo/$namespace
GetUserRepositoryList	DescribeRepositoryOwnerPersonal	Query All Personal Repositories	qcs::tcr:$region:$account:repo/*
CreateRepository	CreateRepositoryPersonal	Create Personal Image Repository	qcs::tcr:$region:$account:repo/$namespace/$repo
DeleteRepository	DeleteRepositoryPersonal	Delete Personal Image Repository	qcs::tcr:$region:$account:repo/$namespace/$repo
BatchDeleteRepository	BatchDeleteRepositoryPersonal	Batch Delete Personal Image Repositories	qcs::tcr:$region:$account:repo/$namespace/*
DeleteTag	DeleteImagePersonal	Delete Personal Repository Tag	qcs::tcr:$region:$account:repo/$namespace/$repo
BatchDeleteTag	BatchDeleteImagePersonal	Batch Delete Personal Repository Tags	qcs::tcr:$region:$account:repo/$namespace/$repo
pull	PullRepositoryPersonal	Pull Images from Personal Image Repository	qcs::tcr:$region:$account:repo/$namespace/$repo
push	PushRepositoryPersonal	Push Image to Personal Image Repository	qcs::tcr:$region:$account:repo/$namespace/$repo

Mappings Between the legacy and new Authorization Solutions

Due to the product name and API interface version upgrades, the resource description and action methods in TCR Personal Edition have changed accordingly. Please use the latest resource and operation authorization scheme while using the API v3.0 interface.

During the API upgrade, the access management-related interfaces will be compatible with both the old and new resource descriptions and actions to ensure that user-defined policies remain effective. For easier management of API interfaces and

authorization schemes, it is recommended to upgrade the authorization scheme to the latest version. For more information, please refer to the Personal Edition Authorization Scheme Example.

The legacy resource-level authorization solution

- **Action**: With `ccr` as the product prefix, the API interface name is based on the v2.0 version. For example, creating a namespace is `ccr:CreateCCRNamespace`.
- **Resource Description**: Use `ccr` as the product name, with only `repo` as the resource type. For example, to describe the image repository repo-b under the namespace namespace-a, use `qcs::ccr:::repo/namespace-a/repo-b`. If `$region` and `$account` are left empty, they default to all regions and the primary account associated with the CAM user who created the policy, respectively.

  For specific authorization schemes, please refer to: TKE Image Repository Resource-Level Permission Settings.

The new resource-level authorization solution

- **Action**: Using `tcr` as the product prefix, the API interface name is in version 3.0. For example, creating a Personal Edition namespace is `tcr:CreateNamespacePersonal`.
- **Resource Description**: With `tcr` as the product name, there are three resource types: `instance`, `repository`, and `repo`. Among them, `repo` is a resource type exclusive to the Personal Edition. For example, to describe the image repository repo-b under the Personal Edition namespace namespace-a, use `qcs::tcr:$region:$account:repo/namespace-a/repo-b`. If `$region` and `$account` are left empty, they default to all regions, and the account defaults to the primary account to which the CAM user creating the policy belongs.

  For specific authorization schemes, please refer to: API List for Personal Edition Accessing CAM and Personal Edition Authorization Scheme Examples.

The compatible example of legacy and new authorization solutions

For example, if you grant a sub-account read access to the image repository repo-b within the namespace-a in the default region, and the image repository is a Personal Edition, the sub-account can only query the repository information and pull images from it, but cannot delete the repository, modify its properties, or push images.

- **Previous Authorization Scheme:**

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

- **New Authorization Scheme:**

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

Configuring Garbage Collection

Last updated: 2023-09-13 15:00:33

Scenario

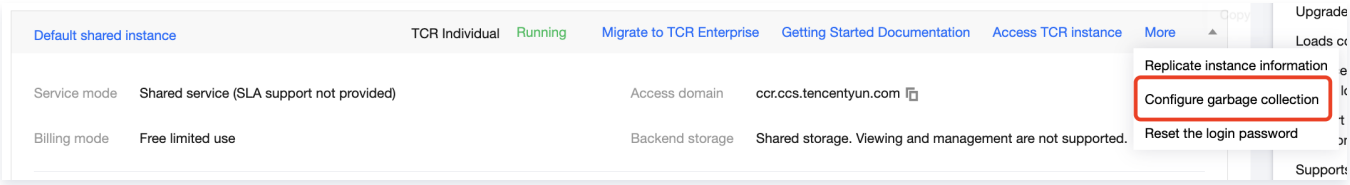
This document describes how to configure/update a garbage collection policy for TCR Individual instances.

Supports and Limits

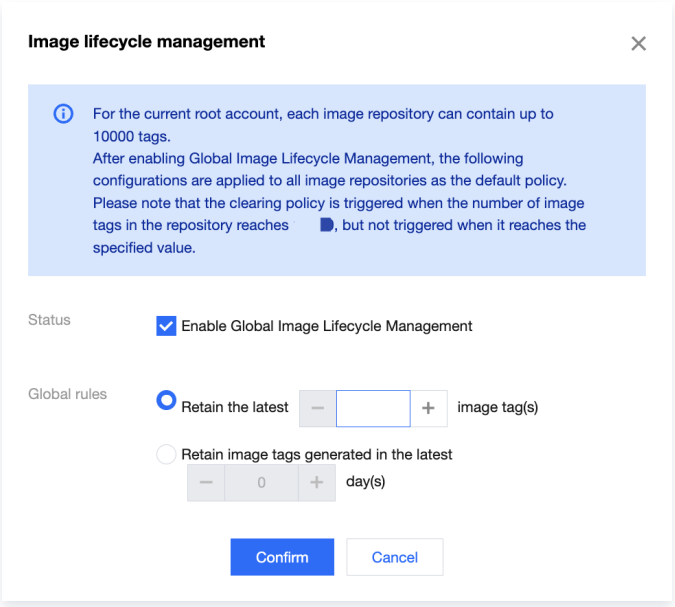
A garbage collection policy for TCR Individual instances is only applicable for the instances in the specified regions. For example, TCR Individual instances deployed in Guangzhou and Silicon Valley regions can be configured with other different garbage collection policies.

Instructions

1. Log in to the TCR console and click **Instance management** in the left sidebar.
2. On the **Instance management** page, you can view the list of instances under your account. Select a TCR Individual instance in any region.



3. Click **More > Configure garbage collection**, as shown below:



   - **Enable Status**: Select "Enable Global Image Lifecycle Management".
   - **Global rules**:
     - **Retain Latest Image tag(s)**: Please set the value according to your needs, but it cannot exceed the default quota for image tags under the current primary account.
     - **Retain image tags generated in the latest**: Please configure as needed.
4. Click **Confirm** to successfully configure.

Terminating/Returning Instances

Last updated: 2023-09-13 14:56:58

Scenario

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

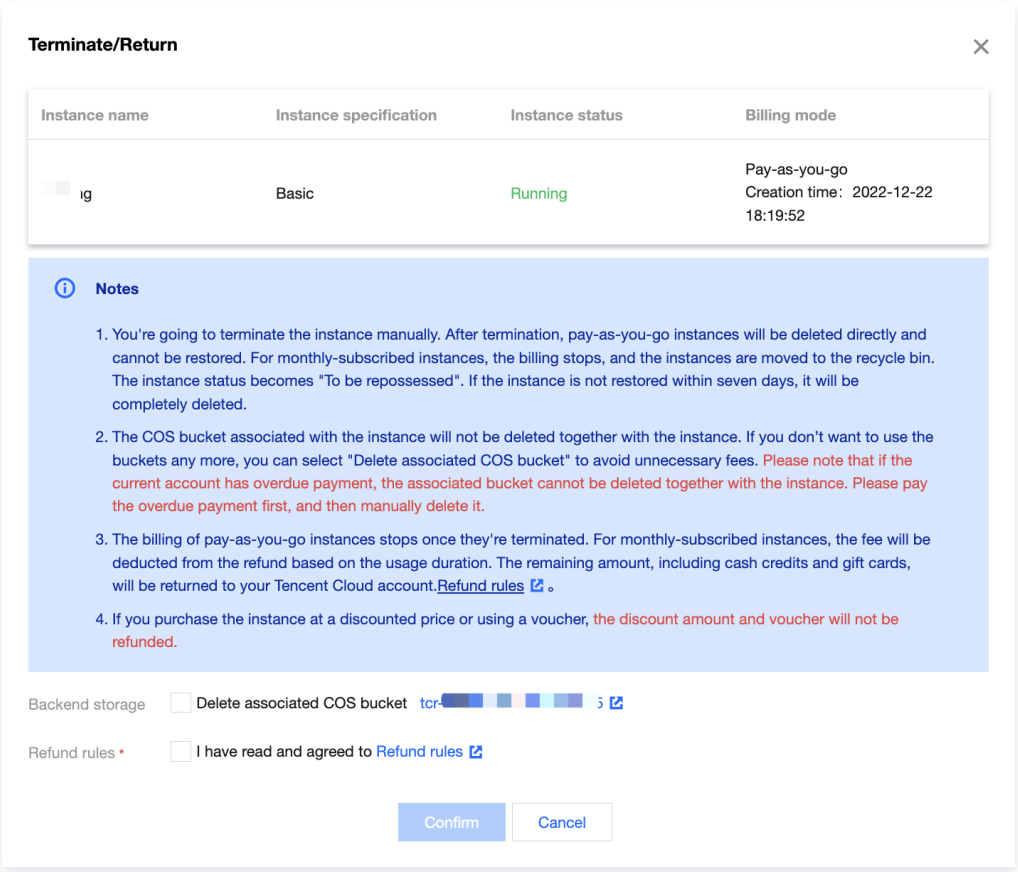
Preparations

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

Instructions

Terminating or returning an instance in the TCR console

1. Log in to the TCR console.
2. Click on **Instance Management** in the left navigation bar to access the "Instance Management" page.
3. Find the instance and choose **More** > **Terminate/Return** on the right side.
4. In the confirmation window that appears, select "Delete associated COS bucket" as needed. As shown in the figure below:



Note

- Please read the notes for terminating and returning the instance carefully. Deleting an instance will completely and irreversibly erase the user data and relevant configuration stored during the use of the instance. Proceed with caution.
- If you are sure that you no longer need the container images, Helm Charts, and other underlying data stored during the use of the 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.
- After the instance is terminated, pay-as-you-go instances will no longer incur fees, and the refund for monthly subscribed instances will be returned to your Tencent Cloud account based on the proportion of cash and voucher

paid during the purchase. For more information, please refer to Refund Policy.

5. After selecting "I have read and agree to the refund policy," click **Confirm** to delete the selected instance. This instance will no longer incur charges.

> **Note**
> If the deletion of the instance takes an unusually long time or the status displayed is abnormal, you may consult us online.

Terminating or returning an instance by calling an API

You can also use the DeleteInstance interface to remove an instance. For more information, please refer to the Delete Instance API documentation.