Tencent Container Registry

Practical Tutorial

Product Documentation

©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice

Tencent Cloud

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

Service Statement

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

Contents

©2013-2022 Tencent Cloud. All rights reserved.

Practical Tutorial
TCR Personal migration
Migration from TCR Individual to TCR Enterprise

Last updated：2023-02-17 11:03:04

Scenarios

Currently, TCR provides the Personal Edition and Enterprise Edition services at the same time. The Personal Edition service is for individual developers and only provides basic features for container image storage and distribution. While the Enterprise Edition service is for enterprise users and can provide a secure, dedicated, and high-performance cloud native artifacts hosting and distribution service. For the differences between the two editions, please see TCR Specifications.

This document describes how enterprise customers can migrate container image data and business configuration from the Personal Edition to the Enterprise Edition to achieve smooth business migration.

Prerequisites

To migrate from the Personal Edition service to the Enterprise Edition, you need to confirm and complete the following preparations:

- You have activated the Personal Edition service and your account has the permission to pull all the images in the image repository of the Personal Edition. For how to grant full read/write permissions of the Personal Edition to the sub-account, please see Example of Authorization Solution of the Personal Edition.
- You have purchased the Enterprise Edition Instance, and your account has the permission to push image to this instance. For how to grant the sub-account the permission for pushing the container image and Helm Chart of the corresponding instance, please see Example of Authorization Solution of the Enterprise Edition. It is recommended to grant the full read/write permissions of TCR to the sub-account that configures synchronization.
- You have configured the running environment of the migration tool. It is recommended to perform the migration task in a VPC to increase the migration speed and avoid the cost of public network traffic.

©2013-2022 Tencent Cloud. All rights reserved.

- Run the migration tool in the VPC: add the VPC where the server of the migration tool is running in the private network access of the target Enterprise Edition instance. For more information, please see Private Network Access Control.

- Run the migration tool in the public network: enable the public network access entry of the target Enterprise Edition instance and allow the access sources. For more information, please see Public Network Access Control.

Data Migration

Preparing the basic running environment

1. Purchase an Enterprise Edition instance in the region where the container business is deployed.

2. In the same region, prepare a CVM for image migration, and try to choose a model with high CPU/memory and high private network bandwidth. After the CVM is started up, install the latest Docker client, or select an operating system with Docker installed.

3. Connect the VPC where the CVM is located to the Enterprise Edition instance, and enable automatic private network resolution. For more information, see Private Network Access Control.

4. Obtain the access credentials of the instance, run `Docker Login` on the CVM for migration, and ensure that you can use the CVM to access the instance successfully via the private network.

Preparing migration configuration

1. Obtain the access credential of the image repository.

   - Access credential of Personal Edition image repository: the username is the UIN of the Tencent Cloud account to which the image to be migrated belongs, and the password is the one set during Personal Edition initialization. If you forgot the password, go to the TCR console > **Instance List**, select your Personal Edition instance, and click **More** > **Reset Login Password** to reset the password.

   - Access credential of Enterprise Edition image repository: you can go to TCR console and select **Access Credential** on the left sidebar. On the **Access Credential** page, select an existing Enterprise Edition instance and click **Create** to generate a long-term access credential dedicated for migration, including the username and password. Note that the user who generates this access credential needs to have full read and write permissions for the instance.

2. Obtain the API calling credential.
You need to call Tencent Cloud API to automatically create the namespace and image repository in the Enterprise Edition instance during the image migration. You can go to the CAM console and select **Access Key** > **API Keys** to create a key or view the existing keys. Please keep the key information carefully.

Downloading and running the migration tool

Run the following command to download the dedicated container image for migration:

```
docker pull ccr.ccs.tencentyun.com/tcrimages/image-transfer:ccr2tcr
```

Run the following command to view the instructions of the tool:

```
docker run --network=host --rm ccr.ccs.tencentyun.com/tcrimages/image-transfer:ccr2tcr /run --help
```

Run the following command to modify the configuration and execute the migration task:

```
docker run --network=host --rm ccr.ccs.tencentyun.com/tcrimages/image-transfer:ccr2tcr /run --tcrName image-transfer --ccrRegionName ap-guangzhou --tcrRegionName ap-shanghai --ccrAuth username:password --tcrAuth username:password --ccrSecretId sid- example --ccrSecretKey skey-example --tcrSecretId sid-example --tcrSecretKey skey-example --tagNum 50
```

Field description

Parameter	Description
tcrName	Name of the target Enterprise Edition instance
ccrRegionName, tcrRegionName	Region of the instance. In the Chinese mainland, the CCR instance region defaults to ap-guangzhou, and the TCR instance region must be set to the actual region such as ap-shanghai.
ccrAuth, tcrAuth	Access credential of the image repository in the format of username:password. If the username and password contain special characters, the special characters need to be escaped. For example, "?" should be written as "\?".
ccrSecretId, ccrSecretKey,	Tencent Cloud API calling key. If the migration is under the same account, the calling keys of CCR and TCR are the same.

tcrSecretId, tcrSecretKey	
tagNum	Specifies that only the latest N tags of images in the image repository will be migrated.

Viewing and confirming the running result

Because the Personal Edition is fully migrated to the Enterprise Edition by default, the migration time is directly related to the number and size of the image repositories in the current Personal Edition. Please wait patiently for the migration. If the following code is displayed, it means that the full migration is successful. Otherwise, re-run the migration tool to try again or submit a ticket for assistance.

```
################ Finished, 0 transfer jobs failed, 0 normal urlPair generate fai
led, 0 jobs generate failed ################
```

Business Migration

Switching the network environment

TCR Personal Edition instance does not support the network ACL. By default, all servers in Chinese mainland regions can access the Personal Edition instance via private network. While TCR Enterprise Edition instance supports the network ACL. You need to connect the VPC of the cluster to the instance and allow the access via the private network.

- Network configuration of the Personal Edition instance: no configuration is required. By default, you can access the instance and upload/download images via all network environments.

- Network configuration of the Enterprise Edition instance:

  - Access via private network (recommended): connect the VPC of the cluster or EKS cluster to the target instance, and enable the automatic resolution in the VPC or manually manage the DNS resolution. For more information, see Private Network Access Control.

  - Public network access: we recommend that you enable the public network access only when testing or distributing images for the external teams, and the access allowlist must be configured.

Reusing the TCR Individual image configuration

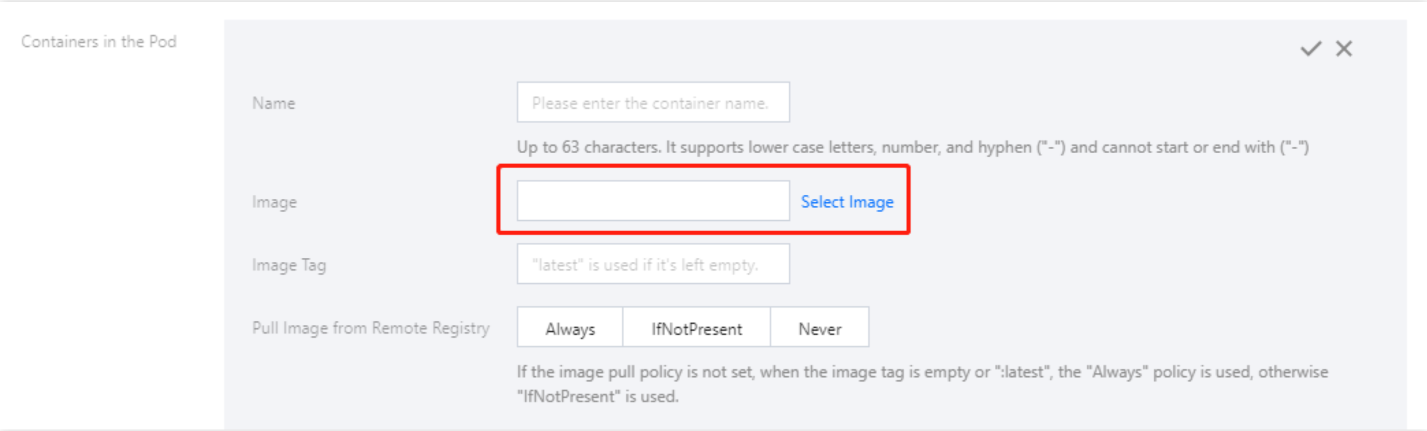
For details, see the document **Using a TCR Individual Domain to Access a TCR Enterprise Instance**. This feature eliminates the need to modify the TKE cluster and the existing image repository address and access credential configuration of the CI/CD platform, allowing customers to switch to TCR Enterprise with minimum cost.

For long-term use, it is recommended to gradually switch the TCR Individual access configuration to the TCR Enterprise configuration to avoid access exceptions caused by subsequent changes in the TCR Individual service.

Using TCR Enterprise image configuration

Switching the access address

Enter the details page of a cluster or EKS cluster, select **Workload** in the left sidebar, and select to create or update a workload. In **Containers in the Pod**, enter or select an image address, or directly modify the **image** parameter in YAML, as shown below:



Containers in the Pod

Name: Please enter the container name.
Up to 63 characters. It supports lower case letters, number, and hyphen ("-") and cannot start or end with ("-")

Image: Select Image

Image Tag: "latest" is used if it's left empty.

Pull Image from Remote Registry: Always | IfNotPresent | Never
If the image pull policy is not set, when the image tag is empty or ":latest", the "Always" policy is used, otherwise "IfNotPresent" is used.

- Image address of Personal Edition instance: defaults to `ccr.ccs.tencentyun.com/namespace/repo:tag` . The default regions cover the AZs in Chinese mainland except Hong Kong, such as Beijing, Shanghai, and Guangzhou. The prefix domain name of Hong Kong is hkccr. Multi-level paths are not supported.

- Image address of Enterprise Edition instance: you can customize the instance name `user-define` , such as `user-define.tencentcloudcr.com/namespace/repo:tag` . You can also customize the domain name, such as `xxx-company.com/namespace/repo:tag` . Multi-level path is supported, and the image address can be `xxx-company.com/ns/sub01/sub02/repo:tag` .

Switching the access credential

Enter the details page of a cluster or EKS cluster, select **Workload** in the left sidebar, and select to create or update a workload. Switch the access credential in **Image Access Credential**, or directly modify the **imagePullSecret** parameter in YAML, as shown below:

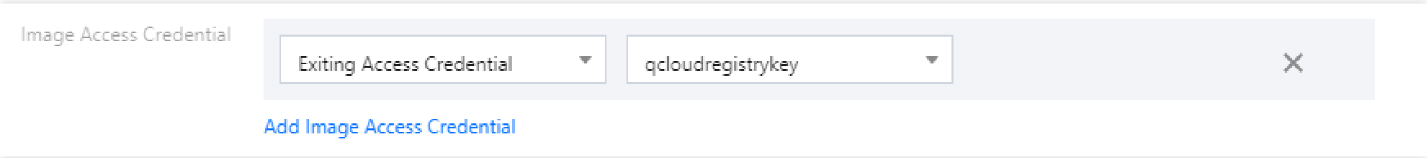


Image Access Credential

Exiting Access Credential

qcloudregistrykey

- Access credential of the Personal Edition instance: when you create a namespace, a Personal Edition access credential **qCloudregistryKey** will be delivered by default.

- Access credential of Enterprise Edition instance:

  - Recommended scheme: use the special plug-in for TCR Enterprise Edition to automatically deliver and configure access credential to achieve secret-free pulling. You do not need to configure the image access credential. Please leave this parameter empty. (This is only available for TKE)

  - Manual configuration: you can configure a long-term access credential in the TCR Enterprise Edition instance, and deliver it to the namespace, or create the image access credential when creating the workload.

Best Practices for TCR Enterprise Instance

This document provides some best practices for migration from the Personal Edition to the Enterprise Edition.

Multi-instance planning

Based on the actual business needs, you can create one or more instances in multiple regions, configure synchronous replication policies, and use the custom domain names to uniformly manage instance access addresses. For more information, see Configuring Instance Synchronization, Configuring Instance Replication, and Synchronizing Images to TCR Enterprise Edition from External Harbor.

Security and Compliance

- Operation compliance

  - We recommend that you grant the minimum permissions to sub-accounts using the Enterprise Edition. You can configure the repository-level permissions, that is, you can specify that a sub-account can only pull or manage the specified image repository, or you can configure more fine-grained API calling permissions. For more information, see Example of Authorization Solution of the Enterprise Edition.

©2013-2022 Tencent Cloud. All rights reserved.

  - You can create dedicated long-term access credentials for different usage scenarios. All long-term access credentials created by an account are only used for accessing the instance.
  - We recommend that you use the temporary login token (valid for 1 hour) for temporary instance access. For more information, see Obtaining a temporary login token.
- Network security
  - Please do not enable the public network access entry to prevent the external unauthorized objects from accessing the instance, or avoid incurring public network access fees for downloading the images of your business via the public network.
  - You can bind a custom domain name to the instance, and manage the public network and VPC resolution of the domain name.

Image management

- Repository planning
  - We recommend that you use the namespace to isolate the services, teams, etc., to facilitate permission management and synchronization rule configuration.
  - Multi-level paths are supported. You can create multi-level repositories as needed to avoid creating too many namespaces.
  - You can push images to automatically create the repositories, and configure the default attributes such as public/private at the namespace level.
  - **Note**: the namespaces and image repositories are path markers. The backend data storage is not isolated.
- Tag naming
  - Please do not use latest to update the image tag in a production environment, which may affect service updates and rollbacks.
  - We recommend that you use the CI tool for the automatic naming of images to facilitate subsequent tag management and image synchronization.
  - **Note**: if you delete an image of a specified tag, the same digest image will be deleted as well.

©2013-2022 Tencent Cloud. All rights reserved.

CI/CD

- The CI/CD service comes with the Enterprise Edition.
  - CI/CD capabilities of the Enterprise Edition are completely based on CODING DevOps, so you need to activate it and complete basic configuration first.
  - Image building is supported. The optional code sources include GitHub, GitLab, TGit, Gitee, and CODING.
  - Delivery pipeline is supported, which enables you to automatically deploy images in clusters, elastic clusters, and edge clusters.
- External services can be connected.
  - You can use the webhook feature to connect to an existing CI/CD system, so image push will automatically trigger a webhook notification whose message body contains the basic image information. For more information, see Managing Triggers.
  - You can directly use the complete service of CODING DevOps, which has built-in TCR templates.

Using Personal Edition Domain Name to Access Enterprise Edition Instance

Last updated：2023-02-28 16:35:24

Overview

Customers who have been using the shared Personal Edition service in production environments and want to upgrade to the exclusive Enterprise Edition instance will need to import the image data from Personal Edition into the Enterprise Edition instance, and change the image address configuration in the existing build and release systems to access the Enterprise Edition instance. In a practical production scenario, image addresses will be used in multiple links in the Kubernetes cluster, such as the build platform, the release platform, and the application YAML definition, so it is costly to modify image addresses uniformly.

In light of the above scenarios and issues, the Enterprise Edition introduces the Personal Edition domain name compatibility feature. This feature allows customers to push and pull Enterprise Edition instance images using the existing Personal Edition image address and access credentials, and supports intelligent origin-pull. If there is no corresponding image in the Enterprise Edition, the system automatically origin-pull requests the corresponding image in the Personal Edition. This minimizes the burden of image repository migration on customer Ops and R&D, accelerating customer migration to the more stable, powerful, high-performance Enterprise Edition as soon as possible.

Prerequisites

You have created an Enterprise Edition instance. For operation details, see Creating an Enterprise Edition Instance.

You have migrated data in the Personal Edition instance to the Enterprise Edition instance. For operation details, see Importing Personal Edition Instance Image to the Enterprise Edition Instance.

Currently, only users on the allowlist can use the feature of accessing the Enterprise Edition instance with a Personal Edition domain name. You need to submit a ticket to apply for the feature.

Basics

The domain names supported by Personal Edition and Enterprise Edition instances are as follows:

Domain names supported by Personal Edition instances

Do not distinguish between public and private domain names. VPC access uses the private linkage by default.

For main service regions (Guangzhou, Shanghai, Nanjing, Beijing, Chengdu, Chongqing): ccr.ccs.tencentyun.com

For other regions: independent service and domain names, for example, hkccr.ccs.tencentyun.com for Hong Kong (China)

Domain names supported by Enterprise Edition instances

Distinguish between public and private domain names, and support custom domain names.

Default domain name: {Enterprise Edition instance name}.tencentcloudcr.com。

Dedicated private domain name: {Enterprise Edition instance name}-vpc.tencentcloudcr.com。

Custom domain name: Supports registering custom domains.

Take migrating the `nginx:latest` image in the `team-a` namespace of the Personal Edition service of the Guangzhou region to the Enterprise Edition instance `company-a` as an example:

Personal Edition access address: ccr.ccs.tencentyun.com/team-a/nginx:latest

Enterprise Edition access address: company-a.tencentcloudcr.com/team-a/nginx:latest

How It Works

When an Enterprise Edition instance is created, the system will issue a certificate that supports Personal Edition domain names by default and supports handling Personal Edition authentication requests.

In a VPC environment, you only need to parse the Personal Edition domain name to the private network access entry of the Enterprise Edition (see Configuring Private Network Access Control). Then, when you access the Personal Edition image repository address in the VPC, you can automatically access the Enterprise Edition service and use the username and password of the Personal Edition service for verification and authentication. You can choose to use Private DNS or manage the cluster node host configure yourself to implement domain name parsing.

When you no longer need to use your Personal Edition domain name to access your Enterprise Edition instance, simply cancel the forced parsing in your VPC and you can access your Personal Edition service.

Use Limits

1. Within one region, the Personal Edition domain name compatibility feature can be enabled for only one Enterprise Edition instance.

2. If you are using a public environment, you need to manually configure parsing the domain name to the Enterprise Edition instance access entry.

3. When using the domain name of the Personal Edition to access the service of the Enterprise Edition, the image repository in the corresponding namespace of the Enterprise Edition will be requested first. Therefore, please avoid using special namespace names such as **library**, **tke**, and **public** in the Enterprise Edition instance; otherwise, the TKE cluster will not be able to access the official image of the product, causing basic service exceptions.

Directions

Confirming the basic environment

1. You have activated and used the Personal Edition service.
2. You have purchase the Enterprise Edition service and synchronized some images of the Personal Edition service to the Enterprise Edition instance.
3. There is a TKE cluster (including the TKE Serverless cluster), and VPC where the cluster resides has been connected to the Enterprise Edition instance. For more information, see Configuring Private Network Access Control.
4. It has been verified that the Personal Edition and Enterprise Edition images can be pulled normally over the internal network in the TKE cluster.

Configuring Private DNS

1. Go to the Private DNS console.
2. Create a private domain.
2.1 Domain name: tencentyun.com
2.2 Associated VPC: Select a VPC connected to the Enterprise Edition.
2.3 Subdomain Recursive Parsing: Keep it enabled.
2.4 Retain the default values of other parameters.
3. Configure Private DNS.
3.1 Click the created private domain to go to its details page.
3.2 Add a record with the following configuration to **DNS Record**:
3.2.1 Host record: For the main service region, enter **ccr.ccs**. For other regions, enter the corresponding domain name prefix, such as **hkccr.ccs**.
3.2.2 Record type: Select **CNAME**.
3.2.3 Record value: Domain name of the Enterprise Edition instance. Use the default or custom domain name. Check that auto parsing has been configured for the default or custom domain name in the product console.
3.2.4 Retain the default values of other settings.
3.3 Retain the default values of other parameters.

Verifying the access effect

Once the preceding configuration is completed, you can verify the effect by using the Personal Edition domain name to access the Enterprise Edition instance.

Scenario 1: Using the Personal Edition domain name to pull an image that has been migrated to the Enterprise Edition instance

1. Use a synchronization tool or manually push an image to the Enterprise Edition instance. Take the following image as an example: `company-a.tencentcloudcr.com/team-a/nginx:latest` , whose corresponding Personal Edition image repository address is `ccr.ccs.tencentyun.com/team-a/nginx:latest` .

2. Log in to the cluster node and manually pull the image, or create new workload to execute image pull. Note that:
2.1 The image address remains `ccr.ccs.tencentyun.com/team-a/nginx:latest` .
2.2 The access credentials remain the configured Personal Edition access credentials.
3. Verify that the cluster can pull the image successfully.

Scenario 2: Using the Personal Edition domain name to pull an image that has not been migrated to the Enterprise Edition instance

1. The Personal Edition already contains the image `ccr.ccs.tencentyun.com/team-b/apache:latest` , and the image has not been synchronized to the Enterprise Edition.
2. Log in to the cluster node and manually pull the image, or create new workload to execute image pull. Note that:
2.1 The image address remains `ccr.ccs.tencentyun.com/team-b/apache:latest` .
2.2 The access credentials remain the configured Personal Edition access credentials.
3. Verify that the cluster can pull the image successfully.

Scenario 3: Using the Personal Edition domain name to push the image to the Enterprise Edition instance

1. Use the Docker CLI or CI platform to push the image and use the Personal Edition address `ccr.ccs.tencentyun.com/team-a/nginx:latest` .
2. If the Enterprise Edition instance already contains a `team-a` namespace, the push will be successful; otherwise, a push failure will be reported.

Recommendations

Use cases

The domain name compatibility feature is recommended for the following scenarios:
Personal Edition images are widely used in environment building, project code, and application deployment, and switching to the Enterprise Edition independent domain name is costly.
The image building and distribution environment is fixed and the cost of one-time configuration of domain name parsing is low.
If the image distribution scenario is complex and it is necessary to support third-party users to access Enterprise Edition images in complex network scenarios, it is not recommended to use the Personal Edition domain name compatibility feature to avoid disrupting the deployment of the production environment.

Canary switching

When initially using the Personal Edition domain name to access the images in the Enterprise Edition instance, it is recommended to push the images to both the Personal Edition and Enterprise Edition so that the cluster can still temporarily switch to access the Personal Edition service when a parsing configuration exception occurs. Later, you

©2013-2022 Tencent Cloud. All rights reserved.

can gradually adjust the existing Personal Edition image address configuration to the Enterprise Edition address and eventually stop using the domain name compatibility feature.

©2013-2022 Tencent Cloud. All rights reserved.

Using the Delivery Assembly Line to Implement Container DevOps

Last updated：2023-03-03 16:37:15

Overview

The cloud-native era has witnessed the popularity of the DevOps concept and its implementation thanks to the rise and wide spread of container technologies. Continuous integration and continuous deployment based on container DevOps can significantly speed up application creation and delivery, thereby enhancing enterprise competitiveness. This document describes how to coordinate the TCR delivery pipeline feature, TKE, and CODING DevOps to offer easy-to-use container DevOps capabilities and enable automatic triggering of image build and application deployment after code push or automatic triggering of deployment after local image push.

Prerequisites

You have purchased a TCR Enterprise Edition instance and created an image repository as instructed in Creating an Enterprise Edition Instance and Basic Image Repository Operations.

You have created a TKE cluster and deployed a container application. For more information, see Creating a Cluster.

You have activated the CODING DevOps service.

Note:

Currently, you can use a TCR Enterprise image to create a workload in the TKE console. In addition, you can install TCR-dedicated add-ons for general TKE clusters to pull images from TCR Enterprise over the private network without a secret. For more information, see Using a Container Image in a TCR Enterprise Instance to Create a Workload.

Directions

Use case 1. Automatic triggering of image build and application deployment after code push

You can configure the pipeline to automatically build the image and trigger automatic deployment to the container platform after code changes.

Configuring delivery pipeline

1. Log in to the TCR console and select **Delivery Pipeline** in the left sidebar.
2. On the **Delivery Pipeline** page, click **Create**.
3. In **Basic Info**, configure the following parameters and click **Next: Image Configuration**.

Pipeline Name: Set the delivery pipeline name.

Pipeline Description: Add description information for the delivery pipeline, which can be modified later.

4. In **Image Configuration**, configure the following parameters and click **Next: Application Deployment**.

Image Repository: Select the image repository associated with the delivery pipeline to automatically configure and push the image build for hosting application deployment.

Image Version Filtering: Impose limitations on the version of images in the execution delivery pipeline to filter out unnecessary ones for execution deployment.

Deploy any version: Any version of the image pushed to the image repository will be deployed.

Deploy specified version only: Specify the image tag and separate multiple versions with commas. Versions not specified will not be deployed.

Deploy specified rule version only: You need to enter a regular expression.

Image Source: Supports images built on the platform and locally pushed. Here, the first kind is used as an example.

Image built on the platform: Allows you to associate code repositories from different code hosting platforms and automatically triggers the delivery pipeline when code changes for the automatic build, image push, and application deployment.

Image pushed locally: When images are manually pushed, application deployment is triggered.

Code Source and Code Repository: Select the code repository used to build the image, and the pipeline will pull the source code of the repository for compilation and build. Authorization is required during first use. Currently, GitHub, public GitLab, private GitLab, Gitee, and TGit code hosting platforms are supported.

Trigger Rule: Rule for triggering automatic image build. Currently, four rules are supported:

Upon pushing to a specified branch: You need to specify a branch.

Upon pushing a new tag: The build is triggered when a tag is created and pushed.

Upon pushing to a branch: You don't need to specify a branch, as the build is triggered upon push to any branch.

Upon matching branch or tag rules: You need to enter a regular expression, for example, `^refs/heads/master$` , to match the master branch for triggering.

Dockerfile Path: The image build is based on a Dockerfile in the code repository, and you need to specify the path to this file. If it is not specified, the file named `Dockerfile` in the root directory of the code repository is used by default.

Build Directory: The working directory where the image build is executed, that is, the context. By default, it is the root directory of the code repository.

Version Rule: Define the name of the image generated by the build, that is, the image tag. You can use custom prefixes and add `branch/tag` , `update time` , and `commit number` environment variables. Here, `update time` is the system time to build the service by running the `docker tag` command.

5. In **Application Deployment**, configure the following parameters and click **OK**.

Platform: The delivery pipeline supports TKE, EKS, and TKE Edge. In this use case, TKE is used as an example.

Region: Region of the target cluster. Select the region of the created general TKE cluster.

Cluster: Target cluster. Select the created general TKE cluster.

Deployment Method: Currently, only **Update existing workloads** is supported.

Namespace: Namespace of the deployed application.

Workload: The workload associated with the deployed application.

Pod Container: Pod container within the workload of the deployed application. It uses the image from the image repository associated in the previous step.

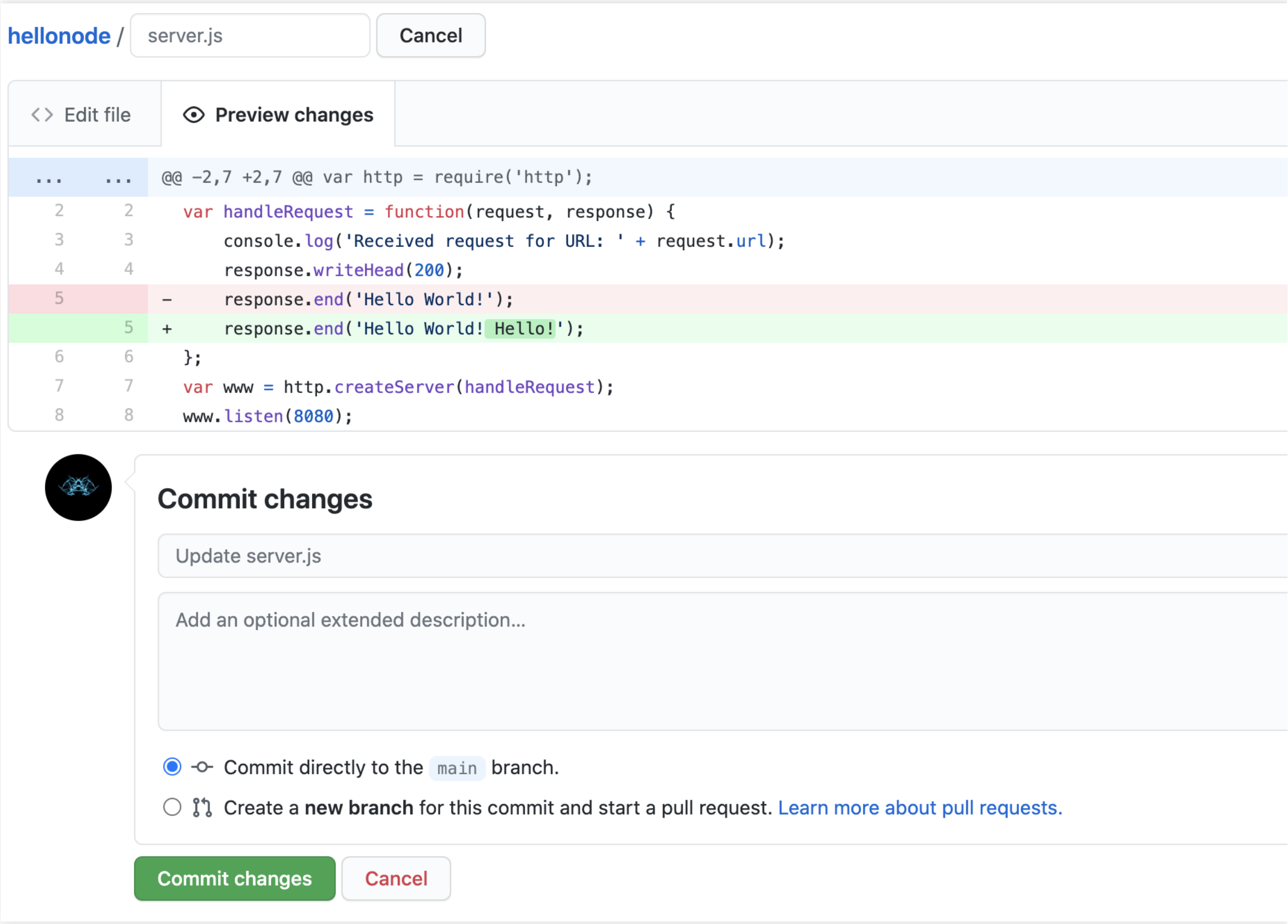
6. After completing the above configuration, you can view the created pipeline on the **Delivery Pipeline** list page.

Updating container application

After completing the above configuration, the system can automatically trigger the image build, push, and application update after the application code is updated.

1. Update the source code.

Update the source code and commit it to the remote code repository as shown below:



```
@@ -2,7 +2,7 @@ var http = require('http');
  var handleRequest = function(request, response) {
      console.log('Received request for URL: ' + request.url);
      response.writeHead(200);
-     response.end('Hello World!');
+     response.end('Hello World! Hello!');
  };
  var www = http.createServer(handleRequest);
  www.listen(8080);
```

Commit changes

Update server.js

Add an optional extended description...

● Commit directly to the `main` branch.

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

2. Execute the pipeline.

After the source code is pushed, the pipeline execution will be triggered if the image build trigger conditions in the image configuration are met. You can click a pipeline to view its execution history and progress.

Checkout: Check out the code.

Docker Build: Build the image based on the image build configuration and tag the generated image with the specified rule, for example, `v-{tag}-{date}-{commit}` .

Docker Push: The system automatically pushes the image to the associated image repository.

Deploy to TKE: Use the latest pushed image to update the associated workload and the image with the same name in the Pod.

3. Check the application update status.

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

3.2 Click the ID of the target cluster to enter the **Workload** page.

3.3 On the **Deployment** tab, click the **Instance Name** to enter the instance details page.

3.4 On the **Update History** tab, view the statuses of application updates.

You can also access the application service to check whether the update is completed, specifically, over the public network address exposed by the Service.

Use case 2. Automatic triggering of deployment after local image push

In some use cases, if you don't need to have an image automatically built, you can still use TCR to have the locally pushed image automatically deployed to a container platform via a trigger.

Configuring delivery pipeline

1. Log in to the TCR console and select **Delivery Pipeline** in the left sidebar.

2. On the **Delivery Pipeline** page, click **Create**.

3. In **Basic Info**, configure the following parameters and click **Next: Image Configuration**.

Pipeline Name: Set the delivery pipeline name.

Pipeline Description: Add description information for the delivery pipeline, which can be modified later.

4. In **Image Configuration**, configure the following parameters and click **Next: Application Deployment**.

Image Repository: Select the image repository associated with the delivery pipeline to automatically configure and push the image build for hosting application deployment.

Image Version Filtering: Impose limitations on the version of images in the execution delivery pipeline to filter out unnecessary ones for execution deployment.

Deploy any version: Any version of the image pushed to the image repository will be deployed.

Deploy specified version only: Specify the image tag and separate multiple versions with commas. Versions not specified will not be deployed.

Deploy specified rule version only: You need to enter a regular expression.

Image Source: Supports images built on the platform and locally pushed. Here, the second kind is used as an example.

Image built on the platform: Allows you to associate code repositories from different code hosting platforms and automatically triggers the delivery pipeline when code changes for the automatic build, image push, and application deployment.

Image pushed locally: When images are manually pushed, application deployment is triggered.

5. In **Application Deployment**, configure the following parameters and click **OK**.

Platform: The delivery pipeline supports TKE, EKS, and TKE Edge. In this use case, TKE is used as an example.

©2013-2022 Tencent Cloud. All rights reserved.

Region: Region of the target cluster. Select the region of the created general TKE cluster.

Cluster: Target cluster. Select the created general TKE cluster.

Deployment Method: Currently, only **Update existing workloads** is supported.

Namespace: Namespace of the deployed application.

Workload: The workload associated with the deployed application.

Pod Container: Pod container within the workload of the deployed application. It uses the image from the image repository associated in the previous step.

Updating container application

After completing the above configuration, you can push the image locally by running commands to trigger automatic deployment.

1. Push the image locally.

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

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

2. Click **Shortcuts** on the right of the instance to view the shortcuts in the pop-up window.

3. Execute the pipeline.

After the image is pushed locally, the pipeline execution will be triggered if the image build trigger conditions in the image configuration are met. As the image is ready, the pipeline only needs to perform automatic deployment.

4. Check the application update status.

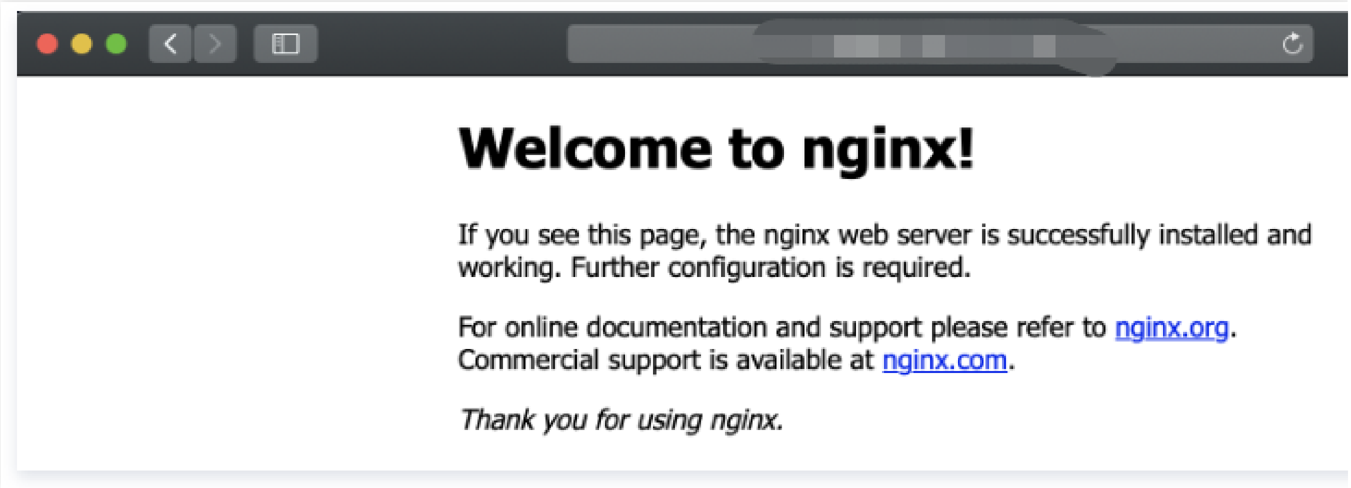
4.1 Log in to the TKE console and select **Cluster** in the left sidebar.

4.2 Click the ID of the target cluster to enter the **Workload** page.

4.3 On the **Deployment** tab, click the **Instance Name** to enter the instance details page.

4.4 On the **Update History** tab, you can view the application update status.

You can also access the application service to check whether the update is completed, specifically, over the public network address exposed by the Service, as shown below:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

TKE Clusters Use the TCR Addon to Enable Secret-free Pulling of Container Images via Private Network

Last updated：2023-02-09 16:05:17

Overview

This document describes how to use the TCR plug-in in Tencent Kubernetes Engine (TKE) to enable secret-free pulling of container images in an Enterprise Edition instance through the private network and to create workloads.

Prerequisites

Before using a private image hosted in TCR Enterprise Edition to deploy applications in TKE, complete the following operations:

Create an Enterprise Edition instance.

Create a TKE cluster.

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

If you are using an existing TKE cluster, ensure that the sub-account has required permissions for the cluster. For more information, see TKE Cluster Permission Management.

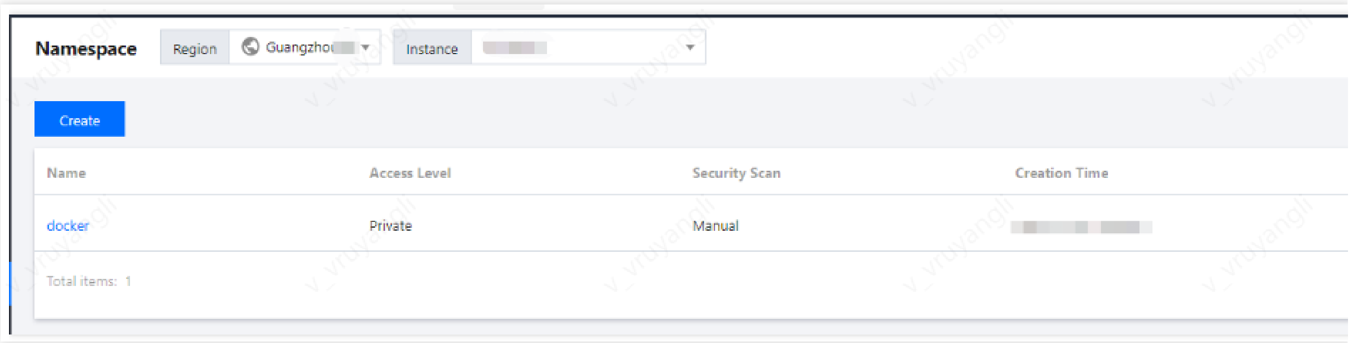
Directions

Preparing a container image

Step 1: Creating a namespace

A new TCR Enterprise Edition instance does not have a default namespace, and a namespace cannot be automatically created through the pushed image. Therefore, create a namespace as required. For more information, see Manage namespaces.

We recommend that the namespace be named based on the project or team name. In this document, `docker` is used as an example. The following page appears after the namespace is created.



Name	Access Level	Security Scan	Creation Time
docker	Private	Manual	[redacted]

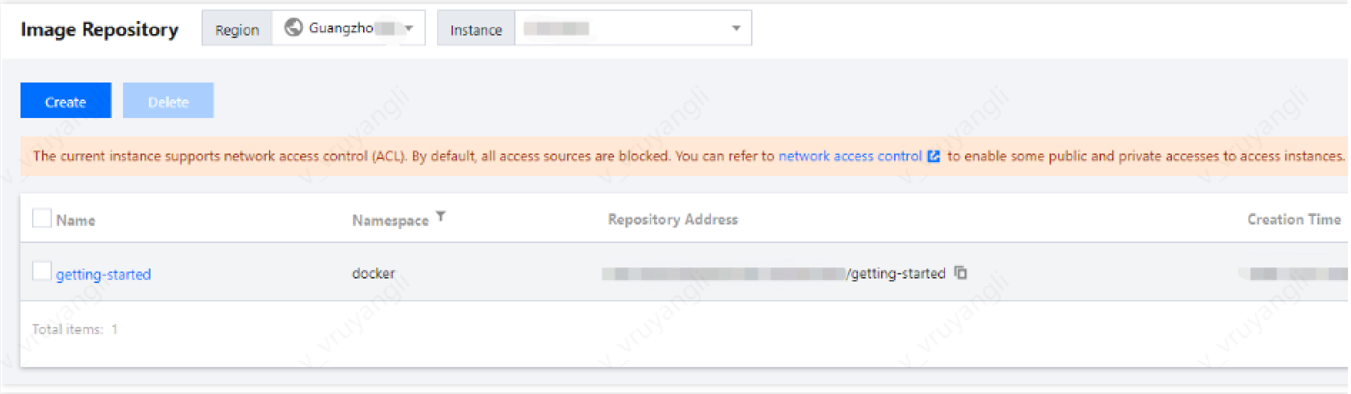
Total items: 1

Step 2: (Optional) Creating an image repository

Container images are hosted in specific image repositories. Create an image repository as required. For more information, see Managing Image Registry. Set the image repository name to the name of the container image to be deployed. In this document, `getting-started` is used as an example. The following page appears after the image repository is created.

Note:

Use Docker CLI or another image tool, such as jenkins, to push the image to the TCR Enterprise Edition instance. If no image repository exists, an image repository will be automatically created. You do not need to create one in advance.



The current instance supports network access control (ACL). By default, all access sources are blocked. You can refer to network access control to enable some public and private accesses to access instances.

□ Name	Namespace	Repository Address	Creation Time
□ getting-started	docker	[redacted]/getting-started	[redacted]

Total items: 1

Step 3: Pushing container images

You can use Docker CLI or another image building tool, such as jenkins, to push an image to a specific image repository. Here, the Docker CLI is used to push images. In this step, you need to use a CVM or CPM with Docker installed and ensure that the target client is in the public or private network access allowlist defined in Network Access Control Overview.

1. Obtain an access credential for the TCR Enterprise Edition instance and run the Docker login command to log in to the instance. For more information on how to obtain an instance access credential, see Obtaining an Instance Access Credential.

2. After successful login, create a container image on the local server or obtain a public image from Docker Hub for testing.

This document uses the latest Nginx image on the official Docker Hub website as an example. In the command-line tool, run the following commands sequentially to push this image. Note to replace `demo-tcr` , `docker` , and `getting-started` with the actual instance, namespace, and image repository names that you created.

```
docker tag getting-started:latest demo-tcr.tencentcloudcr.com/docker/getting-starte
docker push demo-tcr.tencentcloudcr.com/docker/getting-started:latest
```

After the image is pushed, you can go to the Image Repository page in the TCR console and click the name of a repository to view its details.

Configure TKE cluster access TCR instance

TCR Enterprise Edition instances support network access control and deny all external access by default. You can select public network or private network access for a TKE cluster to access a specific instance and pull the container image based on the network configuration of the TKE cluster. If the TKE cluster and TCR instance are deployed in the same region, we recommend that the TKE cluster pull the container image through the private network to accelerate pulling and reduce public network traffic costs.

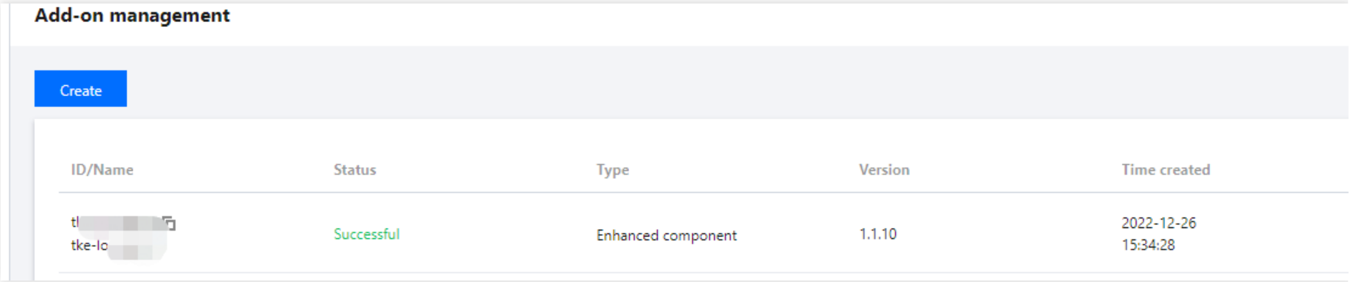
Step 1: Associating the VPC where the cluster is located to the TCR instance

For the data security, the new TCR instance denies all external access by default. To allow the specified TKE cluster to access the TCR instance to pull the image, you need to associate the VPC where the cluster is located to the TCR instance, and configure the corresponding private network domain resolutions.

1. Create a private network access linkage.
2. Configure domain name private network resolution.

Step 2: Installing the TCR add-on in the TKE cluster

If you are using TKE, refer to TCR to install the TCR add-on in the TKE cluster and select **Enable Private Network Parsing** in the **TCR Add-on Parameter Settings** window. For nodes in the cluster, this plug-in can automatically configure private network resolution for the associated TCR instance. This enables secret-free pulling of images in the instance through the private network.
After the add-on is installed, the cluster can pull images from the associated instance without needing a password through a private network.



Add-on management

ID/Name	Status	Type	Version	Time created
tl[redacted] tke-lo[redacted]	Successful	Enhanced component	1.1.10	2022-12-26 15:34:28

Note:
Currently, the TCR add-on only supports clusters in Kubernetes1.12, 1.14, 1.16, 1.18 and 1.20. If you are using another cluster version, manually configure the access method.

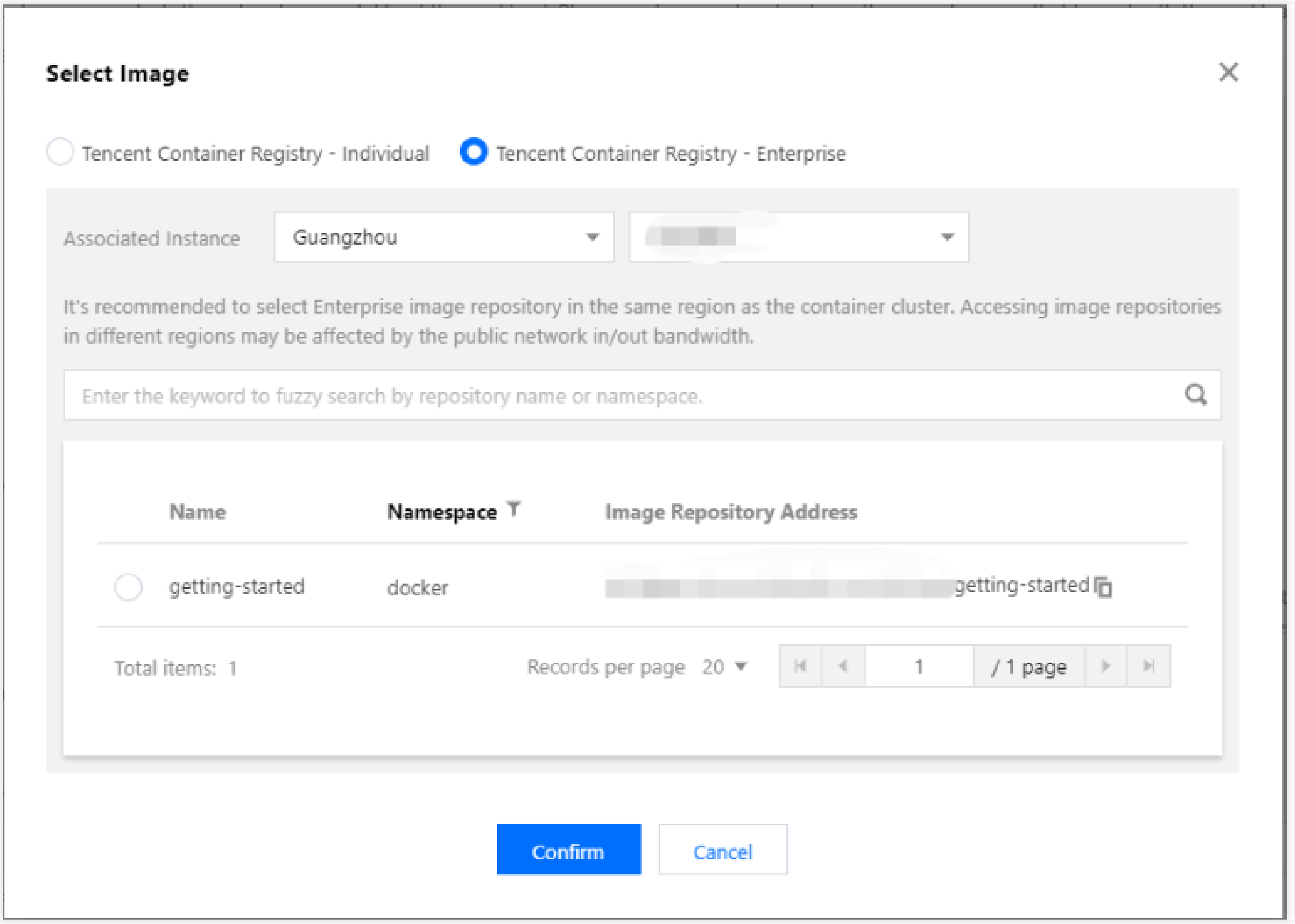
Using the container image in the TCR instance to create a workload

1. Log in to the TKE console and select **Clusters** in the left sidebar.
2. Select the cluster ID that you want to create the workload to go to the cluster details page.
3. On the cluster details page, choose **Workload** > **Deployment** in the left sidebar.
4. On the "Deployment" page, click **Create**.
5. On the “Create Workload” page, create a workload. The main parameters are as follows:

Namespace: select as needed. Make sure that when you install the TCR add-on, the namespace configured to support secret-free pulling already contains the namespace required at this time.

Containers in the Pod：

Image: click **Select Image**, and select **Tencent Container Registry - Enterprise** in the pop-up. Select region, instance and image repository as needed. See the figure below:



Select Image

○ Tencent Container Registry - Individual ● Tencent Container Registry - Enterprise

Associated Instance: Guangzhou [redacted]

It's recommended to select Enterprise image repository in the same region as the container cluster. Accessing image repositories in different regions may be affected by the public network in/out bandwidth.

Enter the keyword to fuzzy search by repository name or namespace.

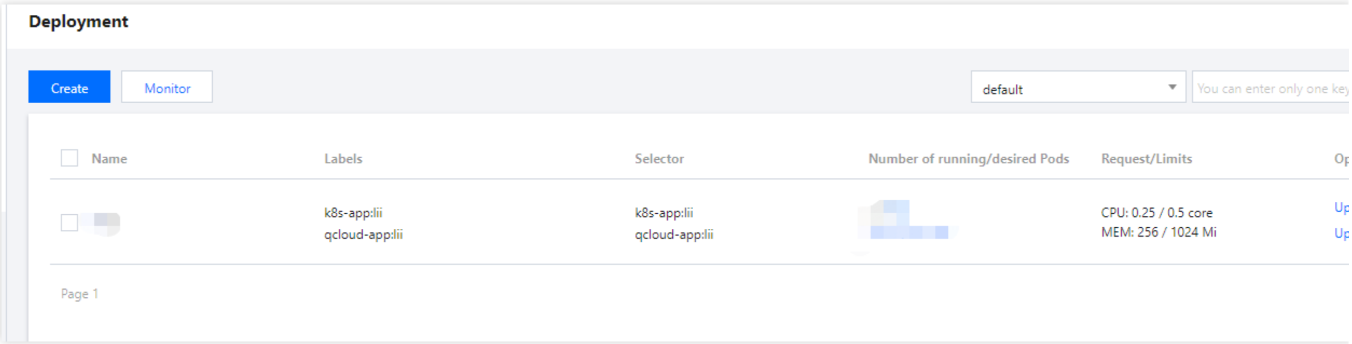
	Name	Namespace	Image Repository Address
○	getting-started	docker	[redacted]getting-started

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

Image Tag: after you select an image, click **Select Image Tag**, and select a tag for the image repository based on your needs in the pop-up. If you do not select, the **latest** will be used by default.

Image Access Credential: if the TCR add-on has been installed in the cluster, explicit configuration is not required. **Please do not select other access credentials, otherwise, this workload cannot load the secret-free pulling configuration of TCR add-on**.

6. After configuring other parameters, click **Create Workload** and view the workload deployment progress. After the workload is deployed, "Number of Running/Desired Pods" for the workload becomes "1/1" on the "Deployment" page.



□ Name	Labels	Selector	Number of running/desired Pods	Request/Limits	Op
□ [redacted]	k8s-app:lii qcloud-app:lii	k8s-app:lii qcloud-app:lii	[redacted]	CPU: 0.25 / 0.5 core MEM: 256 / 1024 Mi	Up Up

Page 1

Synchronizing Images to TCR Enterprise Edition from External Harbor

Last updated：2023-02-09 11:56:52

Overview

When you migrate a container cluster in the customer IDC to TCR, you can also migrate the external container image hosting service to TCR for hosting. After the external image repositories are migrated to TCR, it can not only reduce the O&M costs of building and maintaining the image repository, but also provide professional and stable cloud hosting services and technical supports. In addition, you can use the container service in Tencent Cloud, enjoying a consistent cloud container experience, and use the container clusters to pull images via the private network, reducing the cost of using public network bandwidth.

Harbor is an open source enterprise-class Docker Registry project of VMware. Based on the open source Docker distribution capability, Harbor extends capabilities such as RBAC, secure image scanning, and image synchronization. At present, it has become the preference of creating container image hosting and distribution services. This document describes how to synchronize a container image or Helm Chart in Harbor that is built on the IDC or the cloud server to the TCR Enterprise Edition.

Prerequisites

To synchronize data in the external Harbor to a TCR Enterprise Edition instance, you need to confirm and complete the following preparations first:

- You have built the Harbor service with V1.8.0 and later versions.
- The external Harbor can access TCR through a private line, public network, or VPC.
- You have purchased instances in the region of the cloud container cluster or in an adjacent region.
- If you are using a sub-account, you must have granted the sub-account the permission to push TCR and Helm Chart to corresponding instance. For more information, see Example of Authorization Solution of the Enterprise Edition. We recommend that you grant the full TCR access permissions to the sub-account used for synchronization configuration.

Directions

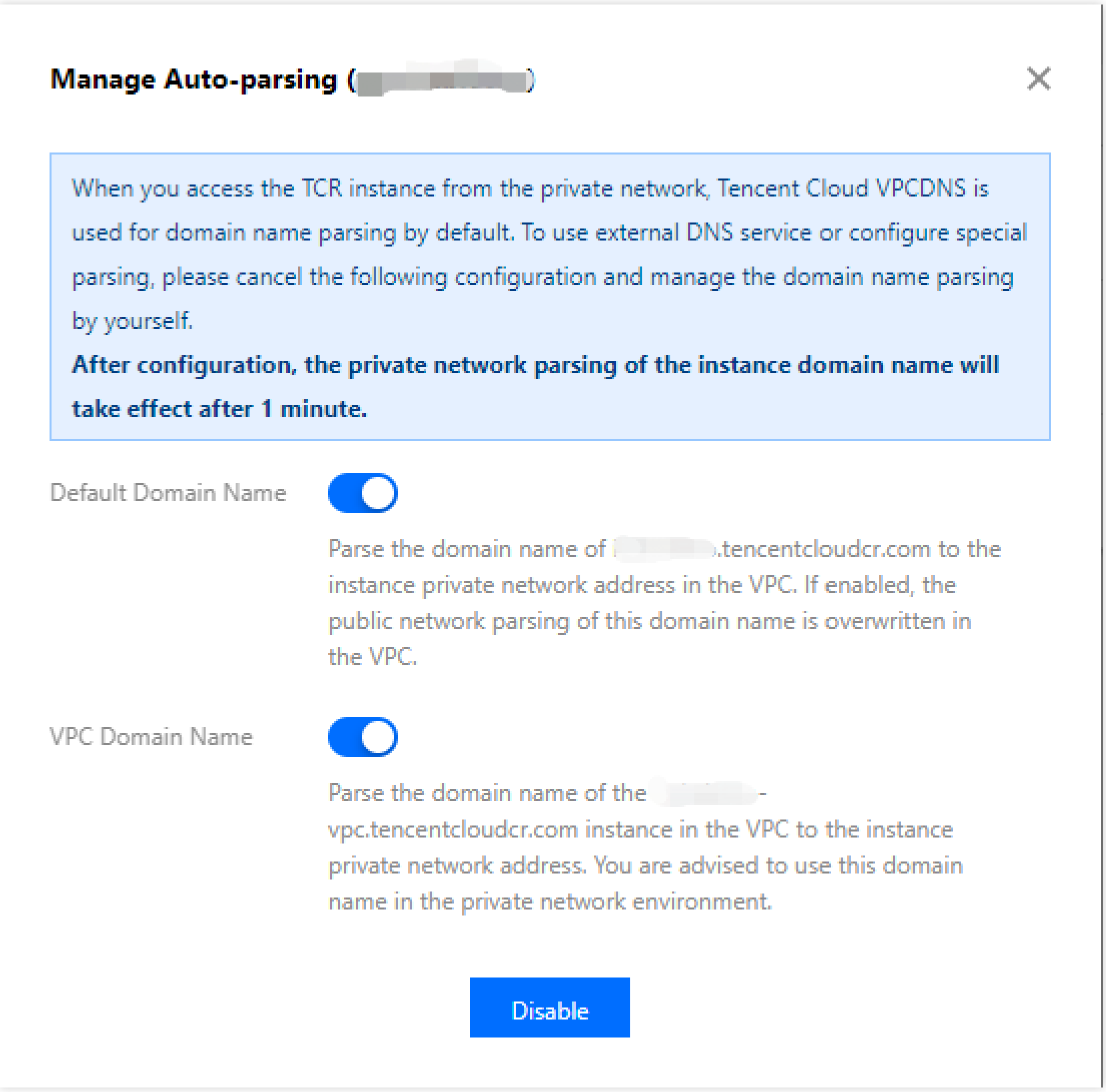
Configuring access to the TCR Enterprise Edition instance for Harbor

Based on the actual network of Harbor, you can select access via Tencent Cloud VPC or access via public network to configure access to the TCR Enterprise Edition instance.

- Accessing via Tencent Cloud VPC
- Accessing via public network

If the current Harbor is deployed in Tencent Cloud VPC or has been connected to the Tencent Cloud VPC through a private line, you can synchronize data through the private network, which can increase the speed of data synchronization and reduce the cost of public network traffic.

1. Log in to the TCR console and select **Network ACL** > **Private network** on the left sidebar.
2. In the **Instance** drop-down list at the top of the page, select an instance for data synchronization.
3. Click **Create**, and in the **Create a private network access linkage** window that is displayed, configure a new private access linkage to allow Harbor to access the instance through the private network.
   - **Associated Instance**: the currently selected instance for data synchronization.
   - **Virtual Private Cloud**: the VPC where the Harbor is located or the VPC connected through Direct Connect.
   - **Subnet**: the created private access linkage occupies a private IP address of the selected VPC. Select a subnet under the VPC to assign the subnet of the private IP address.
4. After completing the above configuration, you can obtain the target access IP address of the private access linkage. To resolve the domain name of the instance into the private IP address in the VPC, please manage the automatic resolution of the private network access linkage and enable the automatic resolution of the default domain name. For more information, see Managing Private Network Resolution.



Manage Auto-parsing ([redacted])

When you access the TCR instance from the private network, Tencent Cloud VPCDNS is used for domain name parsing by default. To use external DNS service or configure special parsing, please cancel the following configuration and manage the domain name parsing by yourself.
After configuration, the private network parsing of the instance domain name will take effect after 1 minute.

Default Domain Name

Parse the domain name of [redacted].tencentcloudcr.com to the instance private network address in the VPC. If enabled, the public network parsing of this domain name is overwritten in the VPC.

VPC Domain Name

Parse the domain name of the [redacted]-vpc.tencentcloudcr.com instance in the VPC to the instance private network address. You are advised to use this domain name in the private network environment.

You can also configure the host on the CVM where the Harbor is located. If you choose manual configuration, you can run the following command on CVM to configure the host. If an independent DNS is currently in use, you can also configure the host in the DNS.

```
echo x.x.x.x harbor-sync.tencentcloudcr.com >> /etc/hosts
```

Creating an access credential for the TCR Enterprise Edition instance

You can create and manage multiple access credentials in TCR Enterprise Edition. We recommend that you create an independent access credential for data synchronization and delete it in time after the data synchronization to avoid the leakage of the instance's access permission.

1. Log in to the TCR console and select **Instance** on the left sidebar.
2. On the "Instance List" page, select an instance for data synchronization to go to its details page.
3. Select the **Access Credential** tab and click **Create** at the top of the instance list.
4. In the pop-up **Create Access Credential** window, perform the following steps:

i. In the “Create Access Credential” section, enter the usage description of the credential and click **Next**. For example, “Dedicated for data synchronization of external Harbor”.

ii. In the “Save Access Credential” section, click **Save Access Credential** to download the credential. **Please save the access credential properly. You will not be able to get it again.**
After the access credential is created, you can view it in the **Access Credential** tab. Please disable and delete the access credential in time after the data synchronization.

Configuring the synchronization repository and rule of Harbor

You can add a third party Registry and configure the data replication rules in Harbor. This document takes Harbor V2.1.2 as an example.

1. Log in to Harbor console with an admin account. You can view and perform **Administration**.
2. Select **Administration** > **Registries** on the left sidebar to go to the **Registries** page.
3. On the **Registries** page, click **NEW ENDPOINT**. Refer to the following information to add a TCR Enterprise Edition instance.
   - **Provider**: select “tencent-tcr”.
   - **Name**: custom the endpoint name for the synchronization such as tencent-tcr.
   - **Description**: the description of the synchronization endpoint.
   - **Endpoint URL**: the access domain name of the TCR Enterprise Edition instance, for example `https://harbor-sync.tencentcloudcr.com` .
   - **Access ID**: enter the `SecretId` obtained from **Access Keys** > **Manage API Key**.
   - **Access Secret**: enter the `SecretKey` obtained from **Access Key** > **Manage API Key**.
   - **Verify Remote Cert**: keep the default setting.
4. Click **TEST CONNECTION**.
   - If “Connection tested successfully” is displayed, the current Harbor can access the TCR Enterprise Edition instance normally.
   - If “Failed to ping endpoint” is displayed, make sure that you have configured access to the TCR Enterprise Edition instance for the external Harbor.
5. Click **OK** to create the registry endpoint.

> Note：
> If the Harbor is an earlier version, there is no "tencent-tcr" option for the **Provider**. Please select "Docker Registry" for **Provider** when creating a registry endpoint. For **Access ID** and **Access Password**, respectively enter the **Login Username** and **Login password** of the long-term access credential of the image repository obtained in **Instance** > **Access Credential** page. In this case, the namespace cannot be automatically created in the TCR.

©2013-2022 Tencent Cloud. All rights reserved.

6. Select **Administration** > **Replications** on the left sidebar and click **NEW REPLICATION RULE**. Refer to the following information to create the rule.
   - **Name**: the rule name. You can enter a name based on the actual usage scenario.
   - **Description**: the rule description.
   - **Replication mode**: defaults to **Push-based**. Only when “tencent-tcr” is selected for **Provider** and the version of Harbor is V2.1.2 or later, you can select **Pull-based**. **Push-based** means to synchronize the new image in the Harbor to the TCR, while **Pull-based** means to synchronize the new image in the TCR to the Harbor.
   - **Source resource filter**: you can filter resources to synchronize. If you do not set a filter, all container images and Helm Chart resources in the Harbor are selected by default.
   - **Destination registry**: select the repository endpoint created in Step 3.
   - **Destination namespace**: specify the destination namespace. If it is left empty, the resources will be put under the same namespace as the source. The default setting is recommended.
   - **Trigger Mode**: defaults to **Manual**. If you need automatic synchronization for the newly pushed container image or Helm Chart, please select “Event Based”. We recommend that you do not select “Delete remote resources when locally deleted”.
   - **Override**: defaults to override the resources at the destination if a resource with the same name exists.

Triggering synchronization and viewing the synchronization log

If you select **Event Based** for the **Trigger Mode**, when container images and Helm Chart are pushed to the Harbor, they will be automatically synchronized to the TCR Enterprise Edition instance. You can select the replication rule to view the synchronization log, and access the TCR Enterprise Edition instance console to check whether synchronization is successful. This document takes the example of manually pushing the container image `nginx:latest` to the Harbor and triggering the synchronization.

1. Push the container image and view it in Harbor.
   Use the docker client to push the local container image `nginx:latest` to Harbor, and then log in to the Harbor console to view the pushed image.
2. View the synchronization record and progress.
   Select **Administration** > **Replications** on the left sidebar and select the replication rule created in Step 6 to view the replication task of the rule.
3. View the synchronized image in TCR console.
   Log in to the TCR console and select **Image Repository** on the left sidebar. In the **Image Repository** page, select the instance used for synchronizing with the Harbor to view the successfully synchronized container image.

©2013-2022 Tencent Cloud. All rights reserved.

TKE Serverless Clusters Pull TCR Container Images

Last updated：2023-05-08 16:19:30

Overview

This document describes how to pull container images in a Tencent Container Registry (TCR) Enterprise Edition instance in a Tencent Kubernetes Engine (TKE) Serverless cluster and to create workloads.

Prerequisites

Before using a private image hosted in TCR Enterprise Edition to deploy applications in TKE, complete the following operations:

You have purchased a TCR Enterprise Edition instance.

You have created a TKE Serverless cluster.

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

Directions

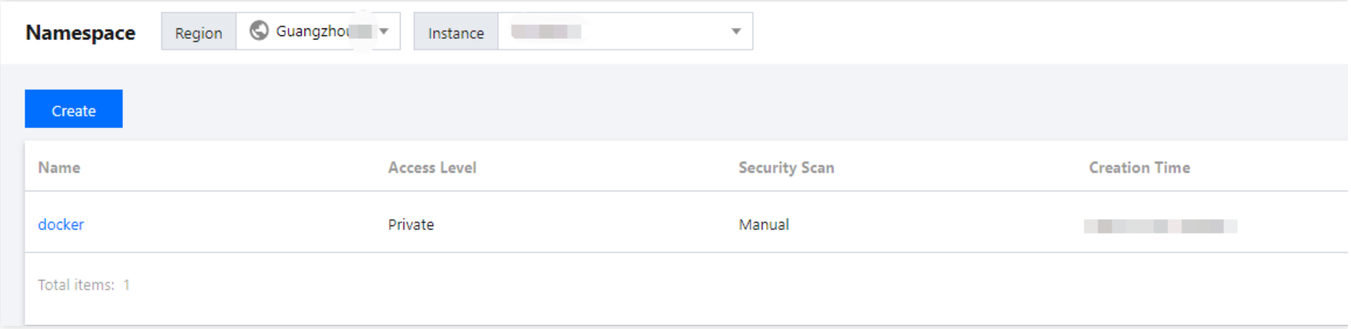
Preparing a container image

Step 1: Creating a namespace

A new TCR Enterprise Edition instance does not have a default namespace, and a namespace cannot be automatically created through the pushed image. Therefore, you must manually create a namespace as needed. For more information, see Managing Namespaces.

We recommend that you name the namespace based on the project or team name. In this document, `docker` is used as an example. The following page appears after the namespace is created.

©2013-2022 Tencent Cloud. All rights reserved.



Namespace Region Guangzhou[redacted] Instance [redacted]

Create

Name	Access Level	Security Scan	Creation Time
docker	Private	Manual	[redacted]

Total items: 1

Step 2: (Optional) Creating an image repository

Container images are hosted in specific image repositories. You can create an image repository as needed. For more information, see Creating an image repository. Set the image repository name to the name of the container image to be deployed. In this document, `getting-started` is used as an example. The following page appears after the image repository is created.

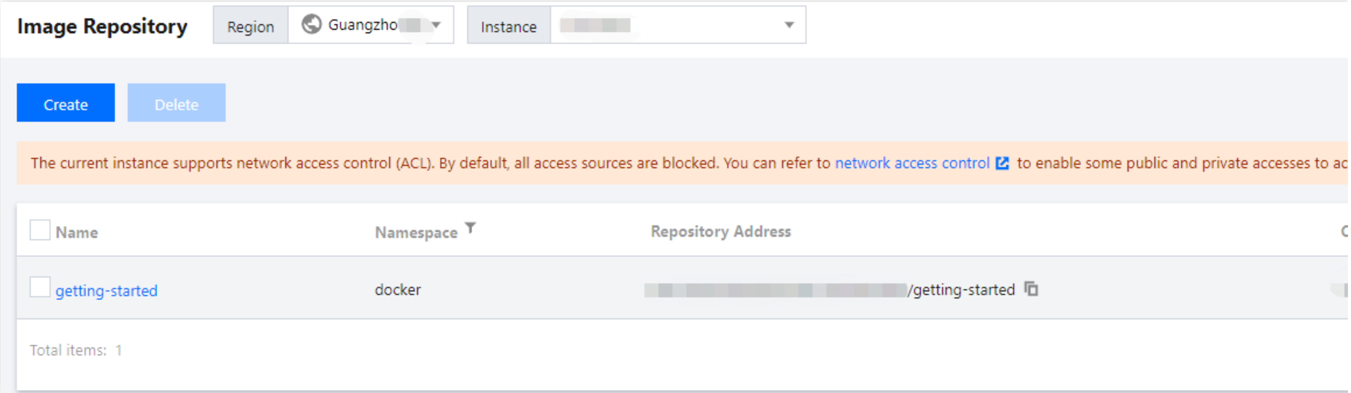


Image Repository Region Guangzho[redacted] Instance [redacted]

Create Delete

The current instance supports network access control (ACL). By default, all access sources are blocked. You can refer to network access control to enable some public and private accesses to ac

□ Name	Namespace	Repository Address	C
□ getting-started	docker	[redacted]/getting-started	

Total items: 1

Note

Use Docker CLI or another image tool, such as Jenkins, to push the image to the TCR Enterprise Edition instance. If no image repository exists, an image repository will be automatically created. You do not need to create one in advance.

Step 3: Pushing container images

1. You can use Docker CLI or another image tool, such as Jenkins, to push an image to a specific image repository. Here, Docker CLI is used to push images. To push a container image, you need to use a CVM or CPM instance with Docker installed and ensure that the client is allowed to access the instance. For more information, see Network Access Control Overview.
2. Obtain an access credential for the TCR Enterprise Edition instance and run the `docker login` command to log in to the instance. For more information about how to obtain an instance access credential, see Obtaining an Instance Access Credential.
3. Create a container image on the local server or obtain a public image from Docker Hub for testing.
This document uses the latest Nginx image on the official Docker Hub website as an example. In the command-line tool, run the following commands sequentially to push this image. Note to replace `demo-tcr` , `docker` , and `getting-started` with the actual instance, namespace, and image repository names that you created.

```
docker tag getting-started:latest demo-tcr.tencentcloudcr.com/docker/getting-starte
```

©2013-2022 Tencent Cloud. All rights reserved.

```
docker push demo-tcr.tencentcloudcr.com/docker/getting-started:latest
```

4. After the image is pushed, you can go to the Image Repository page in the TCR console and click the name of a repository to view its details.

Configuring a TKE Serverless cluster to access a TCR instance

For your data security, TCR and TKE Serverless deny all public and private access requests by default. Therefore, you must configure the network access policies before deploying the TCR image to TKE Serverless.

©2013-2022 Tencent Cloud. All rights reserved.

TCR Enterprise Edition instances support network access control. You can select public network or private network access for a TKE Serverless cluster to access a specific instance and pull the container image based on the network configuration of the TKE Serverless cluster. If the TKE Serverless cluster and TCR instance are deployed in the same region, we recommend that the TKE Serverless cluster pulls the container image through the private network to accelerate pulling and reduce public network traffic costs.

This document describes how to access a TCR instance through the private network. For more information about how to access a TCR instance through the public network, see Accessing Internet through NAT Gateway.

Step 1: Associating the VPC where the cluster is located to the TCR instance

For your data security, the new TCR instance denies all external access requests by default. To allow the specified TKE Serverless cluster to access the TCR instance to pull the image, you must associate the VPC where the cluster resides to the TCR instance, and configure the corresponding private network domain parsing service.

1. Creating an access link
2. Managing private network parsing

Step 2: Obtaining a TCR instance access credential

Before pulling container images from a TCR instance, you need to log in to the instance with the credential. For more information, see Obtaining an Instance Access Credential. Keep the long-term access credential of this instance for later configuration and deployment of TCR images.

Using the container image in the TCR instance to create a workload

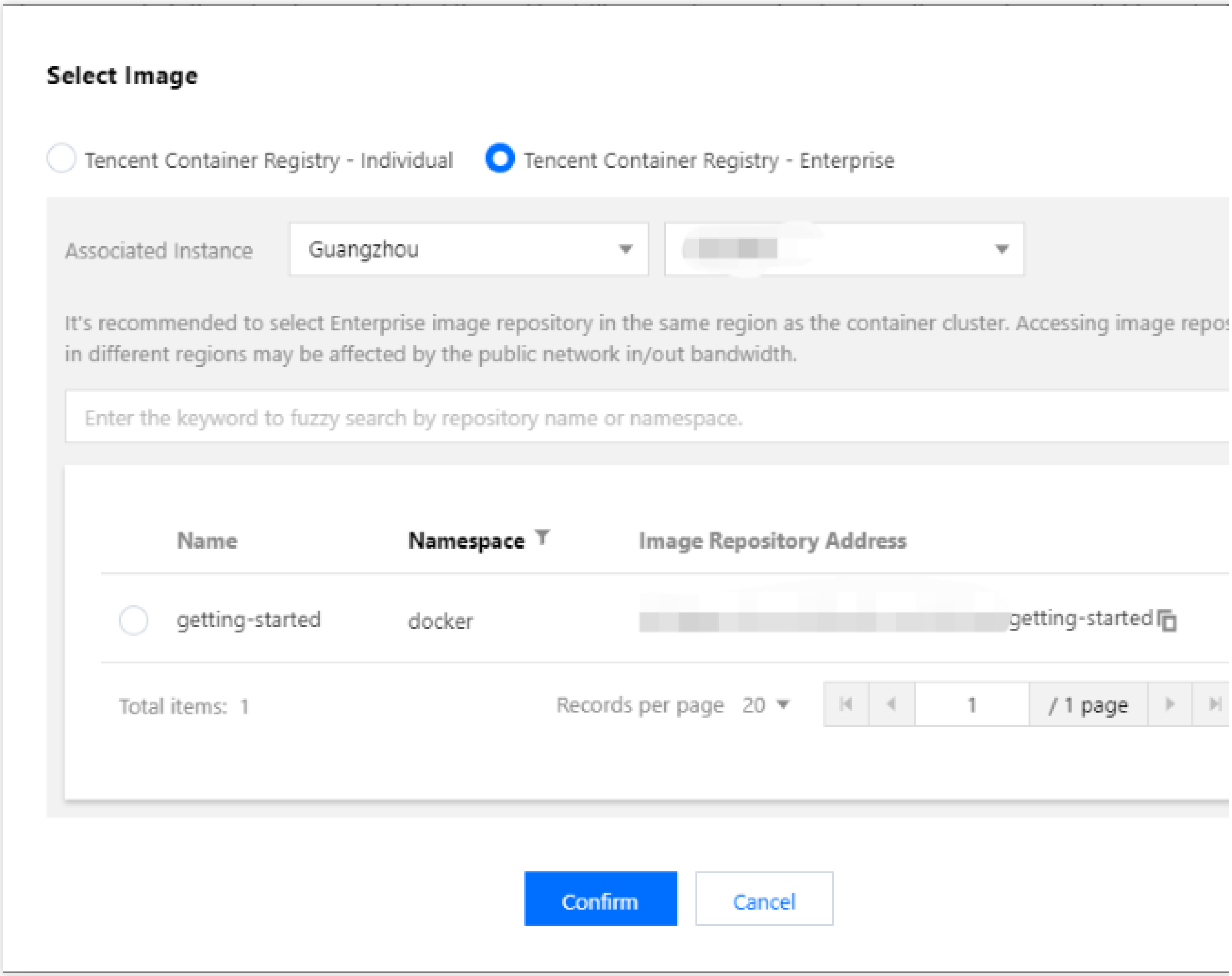
1. Log in to the TKE console.
2. On the cluster list page, click the ID of the target Serverless cluster to go to the cluster details page.
3. On the cluster details page, choose **Workload** > **Deployment** in the left sidebar.
4. On the **Deployment** page, click **Create**.
5. On the **New deployment** page, specify the following parameters to create a workload:

Namespace: Select a namespace in the cluster as needed.

Containers in the Pod:

Image: Click **Select Image**, select **Tencent Container Registry - Enterprise** in the pop-up window, and select the region, instance, and image repository based on your needs. See the figure below:

©2013-2022 Tencent Cloud. All rights reserved.



Select Image

○ Tencent Container Registry - Individual ● Tencent Container Registry - Enterprise

Associated Instance: Guangzhou | [redacted]

It's recommended to select Enterprise image repository in the same region as the container cluster. Accessing image repos in different regions may be affected by the public network in/out bandwidth.

Enter the keyword to fuzzy search by repository name or namespace.

	Name	Namespace	Image Repository Address
○	getting-started	docker	[redacted]getting-started

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

Image Tag: Click **Select Image Tag**, and select a tag for the image repository based on your needs in the pop-up window. If you do not select one, `latest` is used by default.

Image Access Credential: Click **Add Image Access Credential**, and select **Use New Access Credential** from the drop-down list. See the figure below:

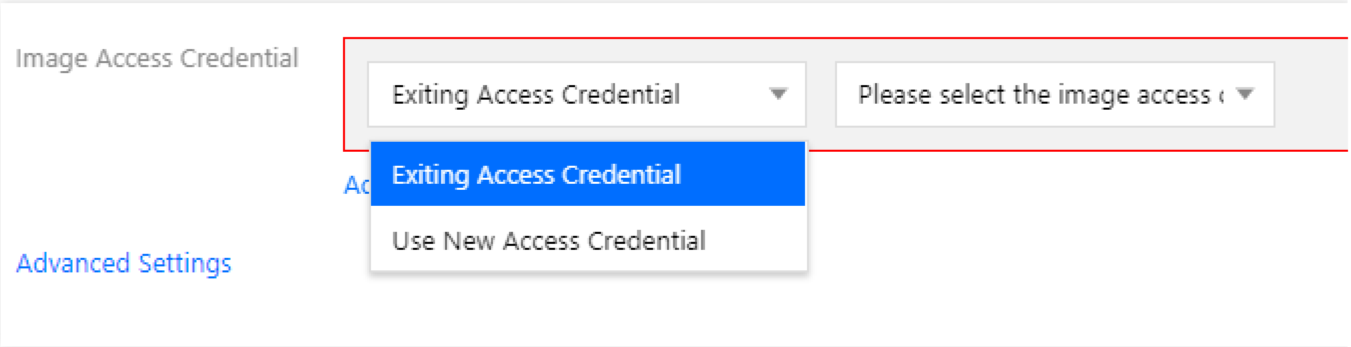


Image Access Credential: Exiting Access Credential | Please select the image access c

Click **Configure Access Credential Information**, and enter the repository domain name, username, and password for the image in the pop-up window.

Repository Domain Name: Log in to the TCR console and click **Image Repository** in the left sidebar to get the repository address of the required image.

Username: Go to Account Info to get the account ID. The account ID is your username.

Password: The access credential obtained in Step 2 is the password.

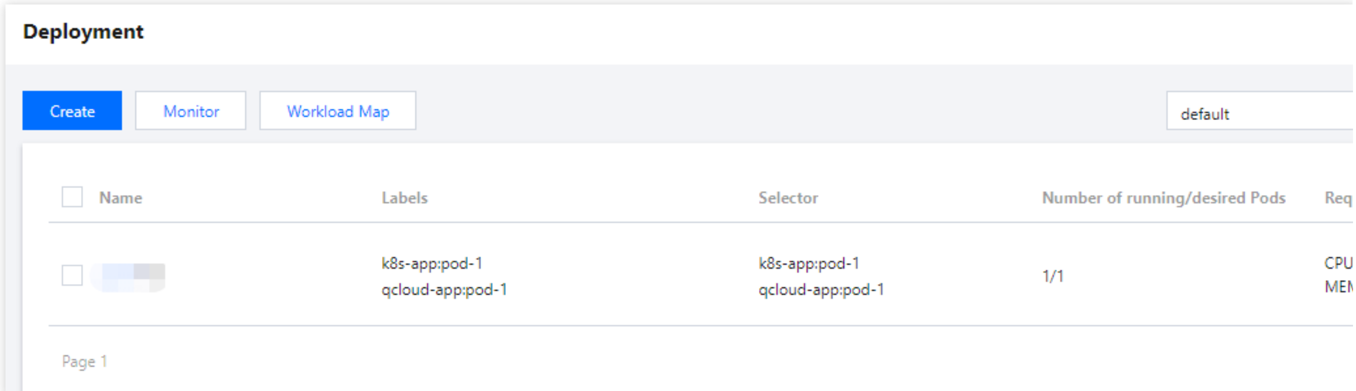
Access Settings (Service): You can deploy various containers in Kubernetes. Some of them provide layer-7 network service over HTTP or HTTPS, and others provide layer-4 network service over TCP or UDP. Service resources defined by Kubernetes are used to manage the service access for layer-4 network in the cluster. Specify the following parameters to complete access settings:

Service: Select **Enable**.

Service Access: Select **Via VPC**.

6. Click **Create Deployment** and view the deployment progress.

After the workload is deployed, "Number of Running/Desired Pods" for the workload becomes "1/1" on the **Deployment** page, as shown in the figure below:



☐ Name	Labels	Selector	Number of running/desired Pods	Req
☐ [redacted]	k8s-app:pod-1 qcloud-app:pod-1	k8s-app:pod-1 qcloud-app:pod-1	1/1	CPU MEI

Page 1

Image Data Synchronization and Replication Between Multiple Platforms in Hybrid Cloud

Last updated：2023-02-13 15:12:40

Overview

During development and OPS, you may need to use multiple container image registries across tenants, regions, borders, and platforms. You can manually push and distribute tasks between instances, but problems such as high OPS costs, slow sync, and difficult management may exist.

For such scenarios, TCR currently provides synchronization and replication features and an open-source image migration tool.

- The instance synchronization feature allows you to **sync** instance images **as needed** based on the configured rules. For more information, see Configuring Instance Synchronization.
- The instance replication feature allows you to **replicate the full** image data from the primary instance to a replica instance. For more information, see Configuring Instance Replication.
- The image migration tool supports migration of Docker image data between **multiple image registry services**. For more information, see Image Migration Tool: image-transfer.
- In addition, when you migrate your data from another image registry service to TCR, you can also configure a custom domain name for your TCR instance or use the original domain name to maintain the service continuity. For more information, see Configuring Custom Domain Name.

This document describes how to sync and replicate image data between different image registries in a hybrid cloud.

Prerequisites

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

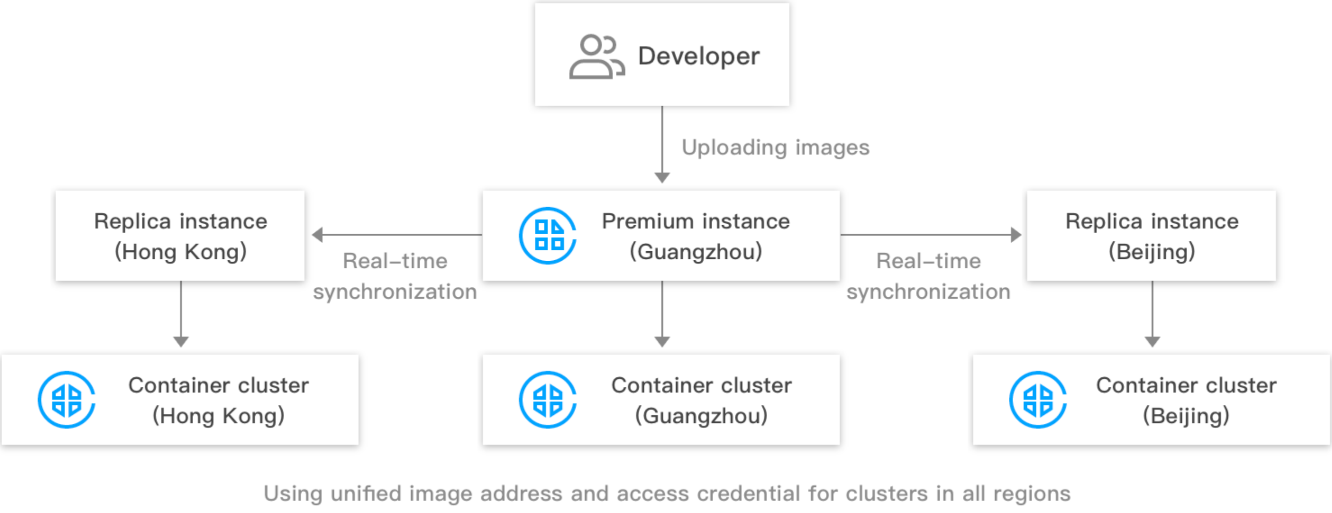
- You have purchased a TCR Enterprise Edition instance with standard specification (for the instance synchronization feature) or premium specification (for the instance synchronization or replication feature).
- If you are using a sub-account, you must have granted the sub-account operation permissions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

Directions

Scenario 1: cross-region TCR instance replication

Cross-region instance replication

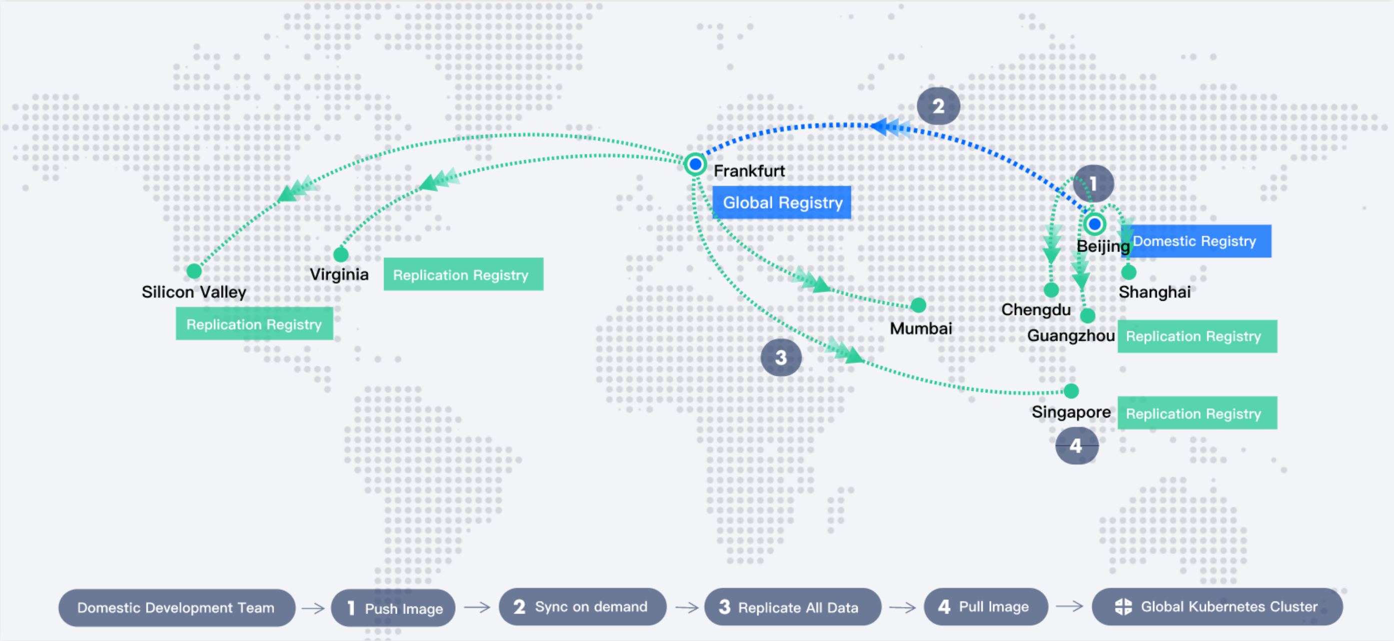
If you have a cross-region business, you can use the instance replication feature to implement single-region upload, multi-region high-speed real-time synchronization, and nearby pull over private network. Compared with the instance synchronization feature, it can unify the release configuration in clusters in multiple regions and improve the cross-region synchronization speed of cloud native artifacts.


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

Cross-border cross-region instance synchronization and replication

If your cross-region business is also cross-border, you also need to use the instance synchronization feature. For security compliance considerations, cross-border instance replication is not supported currently.

1. Purchase a domestic and an overseas premium TCR instance and create a synchronization rule to sync data across borders as needed.
2. Create and manage replica instances in both instances to implement single-region upload, multi-region real-time replication, and nearby pull over private network domestically and overseas.



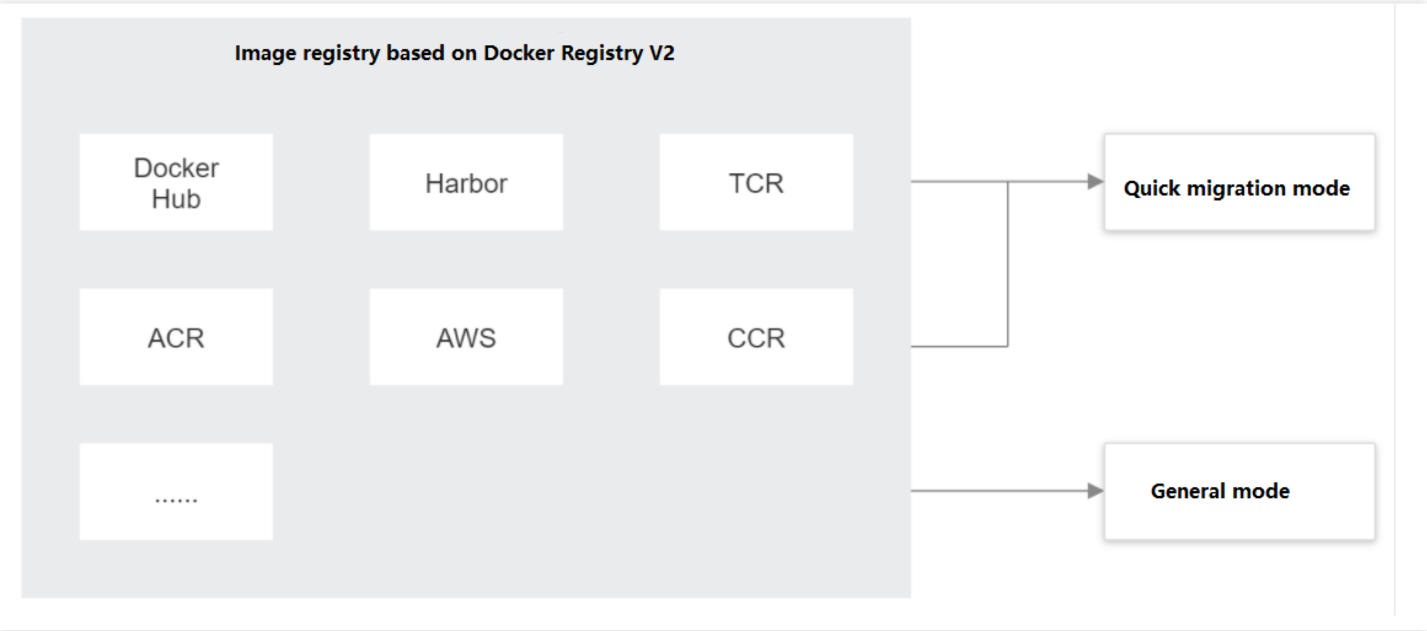
> Note：
> To implement nearby pull over private network, you need to manually connect the VPC in the replication region to the instance. Refer to Private Network Access Control and select the VPC in the replication region.

Scenario 2: cross-platform image migration or synchronization

If you use both a public cloud image registry and a self-built image registry at the same time or use multiple public cloud image registry, you often need to migrate or sync images across platforms. In this case, you can use TCR's custom domain name feature to implement unified access to multiple platforms through the same configuration, so as to ensure service continuity. For more information, see Configuring Custom Domain Name.

Cross-platform image migration

image-transfer is an open-source tool provided by Tencent Cloud for image migration and supports batch image migration between multiple image registry services as long as they are based on Docker Registry V2, such as TCR Personal Edition and Enterprise Edition, Docker Hub, Quay, Alibaba Cloud Container Registry (ACR), and Harbor. It has two use modes: general mode and quick migration mode exclusive to Tencent Cloud as shown below:



- General mode
- Quick migration mode

You can use the general mode of image-transfer to migrate images between multiple image registries. To do so, you only need to configure the authentication file and migration rule file. For more information on how to download, install, and use this tool, see image-transfer.

Cross-platform image synchronization

In cross-platform scenarios, in addition to batch data migration, you often also need to sync images across platforms in real time.

You can sync an image from external Harbor to TCR Enterprise Edition as instructed in Synchronizing Images to TCR Enterprise Edition from External Harbor. To do so, you need to configure the synchronization rules in Harbor. On one hand, migration from an external container image service to TCR reduces the OPS and management costs of building and maintaining the service, and TCR offers professional and stable cloud hosting services and technical support; on the other hand, such migration enables linkage with TKE, so that you can enjoy a consistent user experience of cloud-based containers and pull images over the private network of the container cluster, which reduces the public network bandwidth costs. In addition, you can also configure rules in Harbor to sync images to other third-party registry service platforms. Harbor supports the following image registry services:

- Docker Hub
- Docker Registry
- AWS Elastic Container Registry

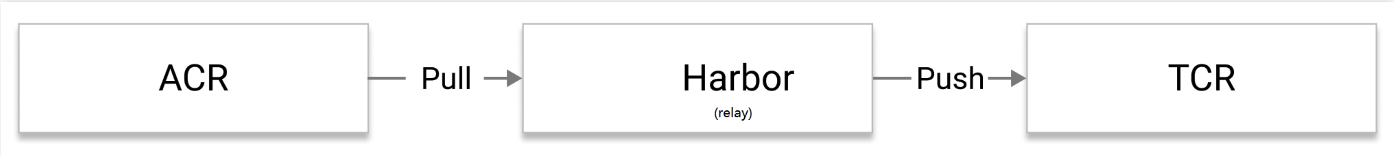
©2013-2022 Tencent Cloud. All rights reserved.

- Azure Container Registry
- Alibaba Cloud Container Registry
- Google Container Registry
- Huawei SWR
- Artifact Hub
- GitLab
- Quay
- Jfrog Artifactory
- Tencent Container Registry

You can also use the external Harbor registry as a relay to sync images between third-party registry service platforms.

The following image takes **real-time image synchronization from ACR to TCR** as an example:

1. Configure a pull-based replication policy in Harbor to pull an image from ACR to Harbor in real time and use Harbor as a relay registry.
2. Configure a push-based replication policy in Harbor to push the image from ACR in Harbor to TCR in real time.



In this way, the image can be synced from ACR to TCR. You can also configure image synchronization between other platforms in the same way.

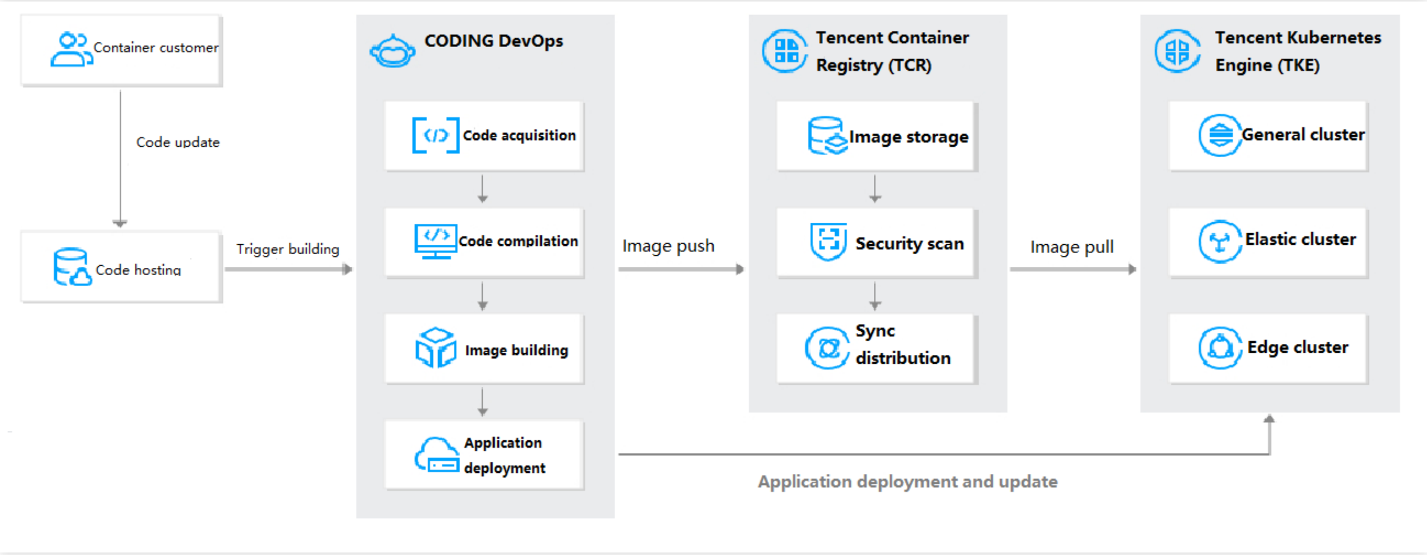
Scenario 3: DevOps image flow

During development and OPS, an application usually needs to undergo multiple steps from development and testing to pre-production and eventual release into the production environment. The corresponding image also needs to flow through multiple steps.

You can use TCR's instance synchronization feature to build a DevOps pipeline for the aforementioned scenario to flow the image. If different root account are used in different environments, enable "Support cross-root account instance synchronization" when configuring an instance synchronization rule.

You can also use the delivery pipeline feature to push code to automatically trigger image building and application deployment or locally push images to automatically trigger deployment.

©2013-2022 Tencent Cloud. All rights reserved.



Note：
Currently, TCR's delivery pipeline feature only supports preconfigured fixed pipelines. If you need more complicated DevOps pipelines, you can use CODING DevOps, which is Tencent Cloud's one-stop DevOps tool. TCR's delivery pipeline feature depends on the continuous integration and deployment features of CODING DevOps.

©2013-2022 Tencent Cloud. All rights reserved.

Nearby Access Through Image Synchronization Between Multiple Global Regions

Last updated：2023-02-09 17:29:57

Overview

When an enterprise expands the container service to multiple regions, it may want that container images can be pulled from the nearest nodes to accelerate pull and reduce the cross-region public network traffic costs; it may also want to implement hot backup across multiple regions and transfer images between multiple image registry services in the same region for cross-team sharing or flow from the development repository to the production repository. In the aforementioned scenarios, the traditional best practice is to create and maintain multiple container image registries (Docker Registry) in one or multiple regions and write scripts to call `docker push` or `docker pull` to implement cross-registry image replication. TCR Enterprise Edition also supports on-demand cross-instance image synchronization as well as single-instance multi-region replication. You can select and combine them flexibly as needed to meet your requirements. The two features have the following strengths:

Instance synchronization

Specified images can be automatically synced between multiple instances in the same region or multiple regions as needed.

- On-demand synchronization: target instances and images to be synced can be precisely matched and selected based on custom rules.
- Automatic synchronization: after the image to be synced is pushed to the source instance, it will be automatically synced to the target instance.
- Cross-tenant synchronization: public images can be synced between instances across multiple root accounts in the same enterprise.
- Helm Chart and container image can be filtered, and you can choose to sync only one type of cloud native artifacts.
- You can query synchronization logs, which can be used with a webhook to implement synchronization success event notification.

Instance replication

TCR enables you to configure replicas (replica instances) in multiple regions for a single instance and supports nearby access.

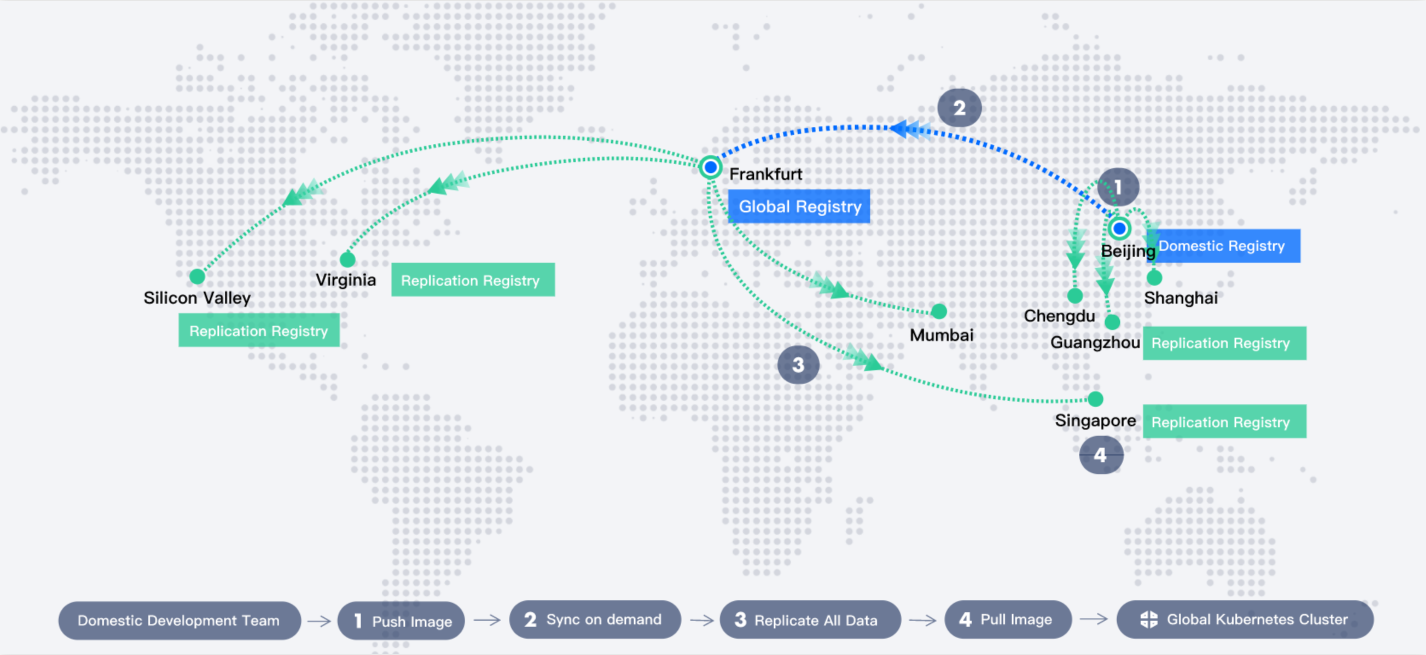
©2013-2022 Tencent Cloud. All rights reserved.

- Single instance: if a single instance needs to be accessed in multiple regions, you can use the same image repository name/tag to maintain consistent container configurations.
- Nearby access: in multi-region deployment, image data in the same region can be pulled to improve the deployment efficiency and stability.
- Reduced costs: nearby image pull over private network can reduce the public network traffic fees or Direct Connect fees incurred in cross-region image registry access.
- Simplified management: you don't need to manage multiple instances, configure synchronization rules between instances, or care about the synchronization status of the specified image.
- Quick synchronization: the image layer data is replicated across regions in real time in a streaming manner; therefore, once pushed to one region, an image can be pulled from multiple regions.

Use Cases

Scenario 1: nearby access in multiple global regions for international business

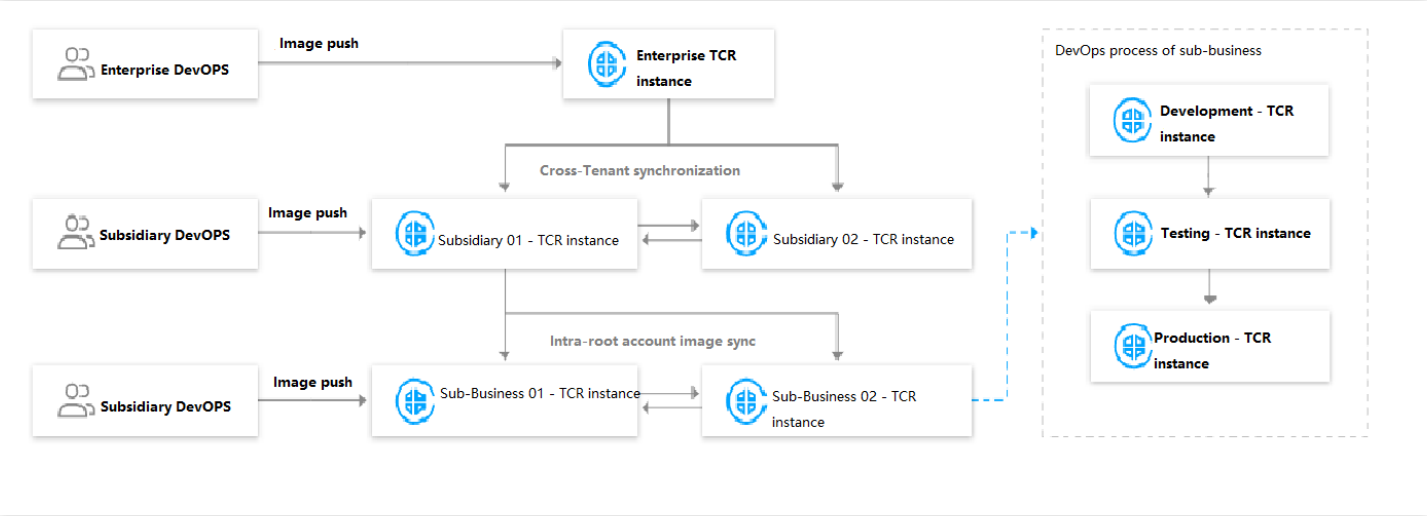
A game developer wants to deploy a containerized gaming business in multiple regions around the globe at the same time and needs to implement multi-region synchronization and nearby access of container images. In addition, as restricted by data compliance regulations, it requires separate management of data in and outside the Chinese mainland for data transfer control. It uses the following solution to implement multi-region data synchronization and nearby access, maintain unified configurations, and greatly improve the business release efficiency and stability.



In this solution, the customer creates two independent instances at the same time in Beijing and Frankfurt, configures instance synchronization rules, and syncs the images released for the production environment as needed. In addition, it configures multiple replica instances in both the Beijing and Frankfurt instances. The Beijing instance contains three replica instances: Shanghai, Chengdu, and Guangzhou. When the latest game version needs to be released, the Beijing development team pushes the latest container image tags for the Chinese mainland and the regions outside the Chinese mainland. Here, the tag for regions outside the Chinese mainland will be automatically pushed to the Frankfurt instance and replicated to regions including Silicon Valley, Virginia, Mumbai, and Singapore. When the container cluster in Singapore updates the image to the latest tag, it can access the replica instance in Singapore over the private network for quick and stable production container update.

Scenario 2: image transfer between multiple subsidiaries and businesses in a large enterprise

A large enterprise has many subsidiaries and business groups, among which a subsidiary has multiple businesses and independent IT teams for them. To separately manage the permissions and costs of cloud resources, many subsidiaries and businesses use different Tencent Cloud root accounts. The enterprise uses the following solution to share the basic images within it and share images between multiple businesses:


Enterprise DevOPS
Image push
Enterprise TCR instance
DevOps process of sub-business
Development - TCR instance
Cross-Tenant synchronization
Subsidiary DevOPS
Image push
Subsidiary 01 - TCR instance
Subsidiary 02 - TCR instance
Testing - TCR instance
Production - TCR instance
Intra-root account image sync
Subsidiary DevOPS
Image push
Sub-Business 01 - TCR instance
Sub-Business 02 - TCR instance

In this solution, cross-tenant instance synchronization is configured between instances under multiple subsidiary accounts to share public images. The basic platform admin configures the basic image synchronization between business instances in each subsidiary in a unified manner. Some businesses plan to use multiple instances to separately manage images during business development, testing, and production and automatically transfer business images between each production stages based on image tag.

Note：

Both of the above scenarios are quite complex, and you can choose a certain part of the solutions to meet your requirements in regular businesses, such as cross-region hot backup, multi-region nearby access, and intra-region cross-instance business image flow.

In addition, an Enterprise Edition instance allows you to leverage the custom domain name feature. Together with DNS, you can use the same domain name for multiple instances to implement unified image release configuration and multi-region nearby access over private network, which helps you manage instances more flexibly.

Operation Guide

Prerequisites

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

Configuring instance synchronization

For detailed directions, see Configuring Instance Synchronization.
Instance synchronization supports cross-tenant instance synchronization, and you can configure it when creating a synchronization rule.

Configuring instance replication

For detailed directions, see Configuring Instance Replication.
Instance replication doesn't support replication between regions in and outside the Chinese mainland; for example, you cannot configure a replica instance in the Silicon Valley region for a premium instance in the Beijing region.

Configuring custom domain name

For detailed directions, see Configuring Custom Domain Name.
Custom domain name DNS needs to be configured independently. For Tencent Cloud, you can use Private DNS. For Tencent Cloud International, Private DNS is not supported currently, and we recommend you use a self-built DNS service.

Troubleshooting Guide

1. What should I do if instance synchronization failed?

You can go to the TCR console, view the relevant rules and synchronization logs, and manually trigger synchronization. If the synchronization is slow or still fails, submit a ticket for assistance.

2. What should I do if instance replication failed?

Currently, instance replication doesn't allow you to view the synchronization logs of the specific repository. Wait patiently during instance replication and try accessing the image after the status becomes "Synced successfully". If the instance replication status doesn't become "Synced successfully" or you still cannot get the image after successful synchronization, submit a ticket for assistance.

3. How do I know whether the specified image has been replicated to a replica instance?

Currently, instance replication allows you to view historical synchronization tasks. You can check the task details logs to determine the synchronization status of the specified image repository or image.

4. How do I accelerate cross-region instance synchronization?

Currently, you cannot accelerate cross-region synchronization of the specified account or instance. The synchronization speed is subject to the cross-region CCN bandwidth, which is shared between multiple tenants. If you want a special guarantee, submit a ticket for assistance.

©2013-2022 Tencent Cloud. All rights reserved.

Using Custom Domain Name and CCN to Implement Cross-Region Private Network Access

Last updated：2023-02-28 16:35:24

Overview

TCR Enterprise Edition supports network access control. It allows users to access a specified VPC and allows the Docker clients within the VPC to access image data over the private network. With the popularization and practice of the concept of multi-cloud/distributed cloud, users' container cluster is no longer located in a single VPC in the designated region of Tencent Cloud, but may be distributed in complex networks of multiple cloud vendors and IDCs, and these complex networks may be interoperable through the CCN and Peering Connection network products. In this context, users need to access a single TCR Enterprise Edition instance from multiple regions and VPCs simultaneously for normal private network push and image pull.

This document mainly introduces how an enterprise customer uses a custom domain name together with the CCN, Peering Connection, and Private DNS products to enable multiple VPCs to access a TCR instance simultaneously and distribute container images over the private network.

In particular, if your business is distributed in multiple clouds and multiple regions, in order to realize data disaster recovery backup and nearby access, it is recommended that you refer to the following best practices and choose the most suitable scheme according to your business needs: Image Data Synchronization and Replication Between Multiple Platforms in Hybrid Cloud and Nearby Access Through Image Synchronization Between Multiple Global Regions.

Prerequisites

Check that you have completed the following preparations:

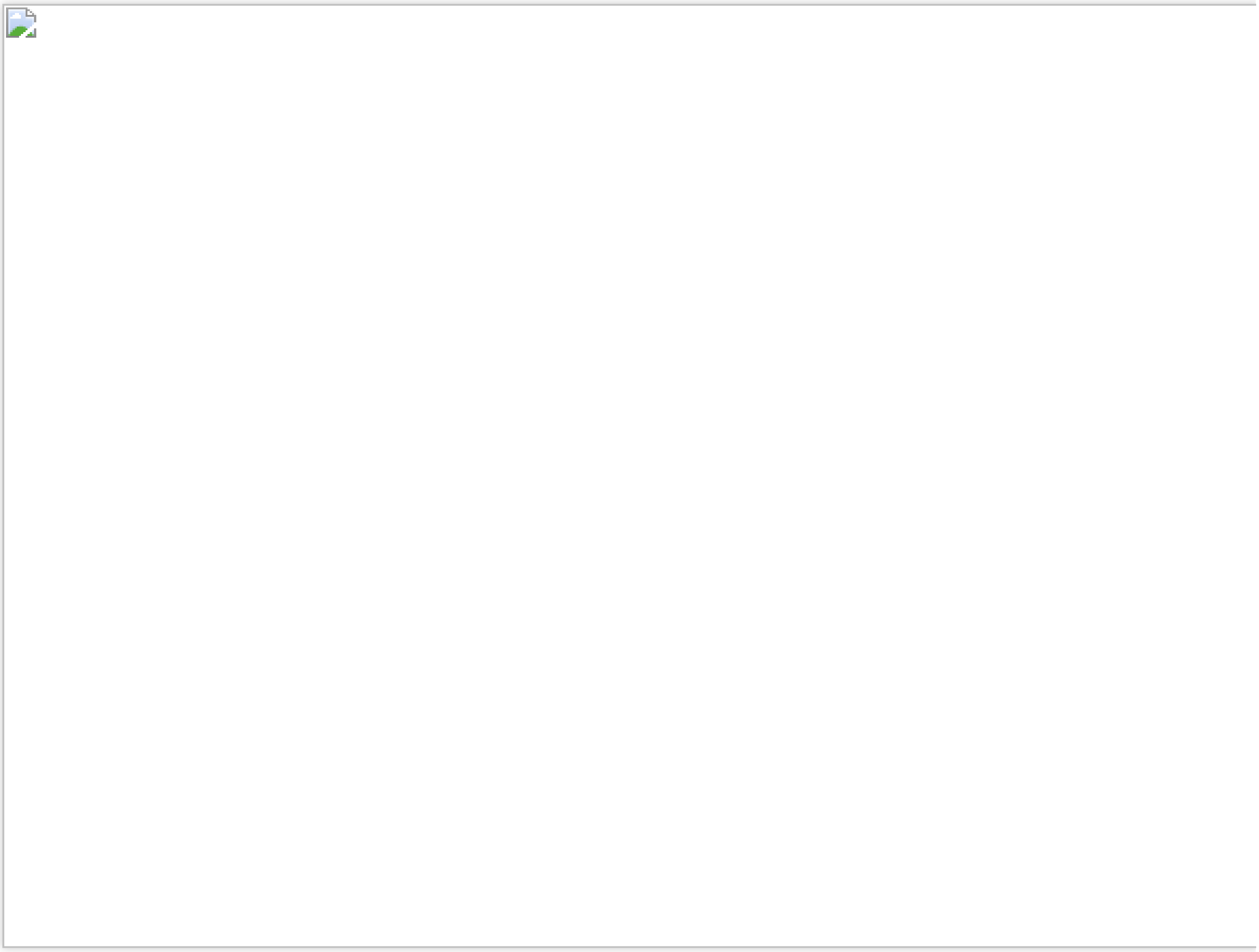
Purchase a TCR Enterprise Edition instance and obtain the instance management permissions such as QcloudTCRFullAccess.

Configure a valid domain name. For more information, see Configuring Custom Domain Name.

Activate services such as CCN and Peering Connection, and access multiple VPCs.

Overall Structure

The customer deployed containerized business in both Guangzhou and Shanghai and used the TCR Enterprise Edition instance in Guangzhou to host and distribute container images.



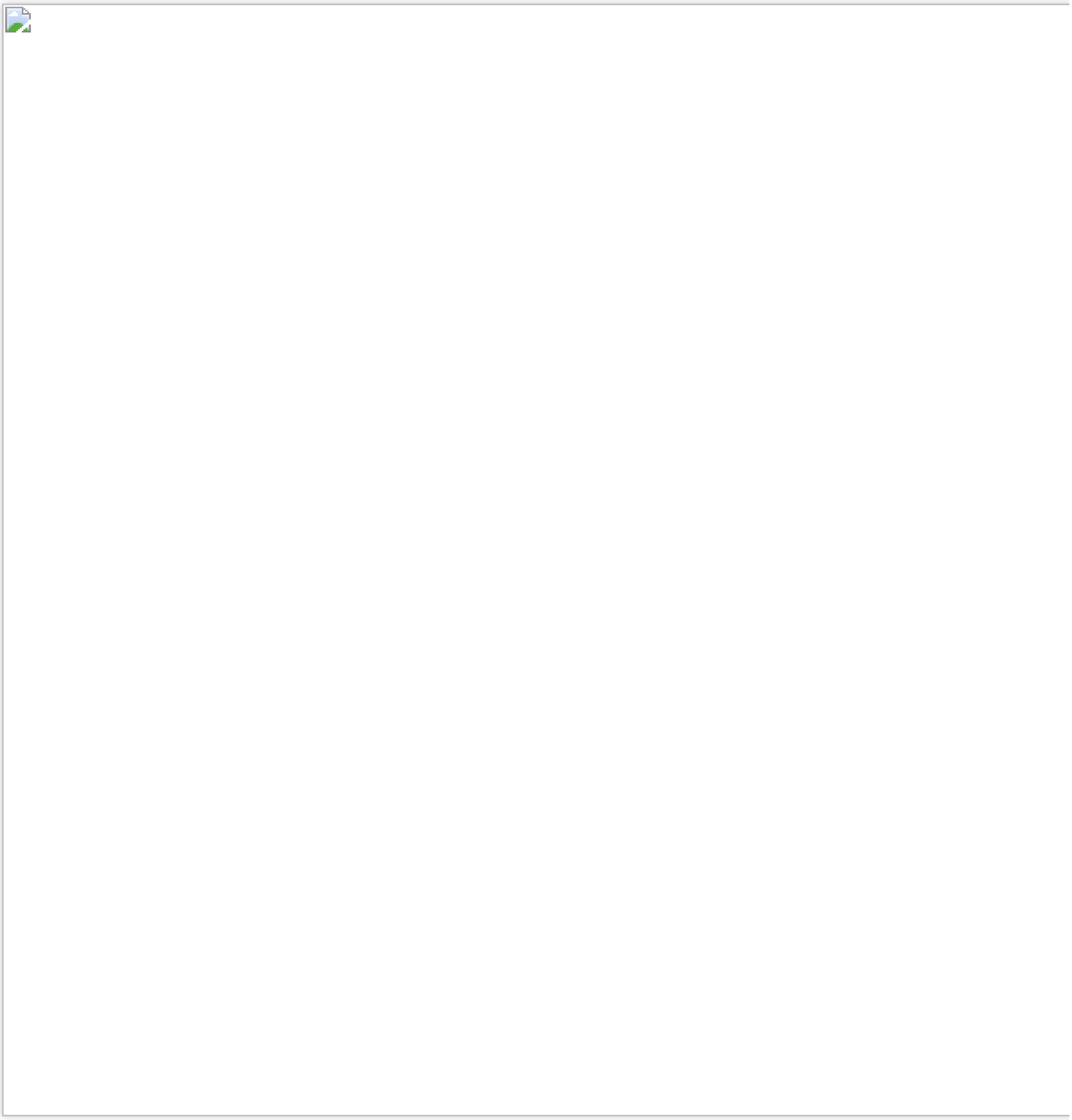
Configuration Details

Creating a TCR Enterprise Edition instance and binding a custom domain name

1. Purchase a TCR Enterprise Edition instance in the region where the container business is deployed. For more information, see Purchasing TCR Enterprise Edition Instance. For this best practice, select Guangzhou (ap-guangzhou, gz).
2. Initialize the instance and upload the first image. For more information, see TCR Enterprise Edition Getting Started. For this best practice, this step is to access the specified VPC vpc-gz-01 and push images over the private network.
3. Configure a custom domain name. For more information, see Configuring Custom Domain Name.

Associating multiple VPCs with CCN

1. Go to the VPC console, create a CCN instance, and associate it with the Guangzhou and Shanghai VPCs.



2. You can choose to use the peering connection feature to associate the VPCs mentioned above.

Configuring Private DNS for the custom domain name

1. Go to the Private DNS console, use the bound custom domain name to create a private zone, and associate it with the VPCs mentioned above.
2. Configure the parsing record: Select **A record**, use **@** to directly parse the main domain name, and configure the record to the private IP corresponding to the accessed VPC.

©2013-2022 Tencent Cloud. All rights reserved.

Scenario Verification

Verifying the VPC connected to the instance

1. In the connected VPC in Guangzhou, create a CVM and install the Docker client.
2. Log in to the CVM and try to pull the image. The following is a reference command, where you need to replace `demo-tcr.cn` with the actual bound custom domain name and replace `demo/nginx:latest` with the actual image address (`demo` is the namespace).

```
# Pull the image in the container cluster in Guangzhou
```

©2013-2022 Tencent Cloud. All rights reserved.

```
docker pull demo-tcr.cn/demo/nginx:latest
```

If the image pull is successful, the VPC connection, custom domain name, and Private DNS are configured properly, and the container cluster of the Guangzhou VPC can use the custom domain name to pull images over the private network.

Verifying the other VPC connected to CCN

1. In the VPC connected to CCN in Shanghai, create a CVM and install the Docker client.
2. Log in to the CVM and try to pull the image. You can use the same path to directly pull the Enterprise Edition instance in Guangzhou.

©2013-2022 Tencent Cloud. All rights reserved.

```
# Pull the image in the container cluster in Shanghai
docker pull demo-tcr.cn/demo/nginx:latest
```

If the image pull is successful, the CCN configuration is normal, and the container cluster of the Shanghai VPC can use the custom domain name to pull the image across regions over the private network.

©2013-2022 Tencent Cloud. All rights reserved.