CODING Continuous Deployment

Getting Started

Product Documentation

Getting Started

Last updated：2024-01-03 11:38:30

This document describes the basic operations in CODING Continuous Deployment.

Prerequisites

You must activate the CODING DevOps service for your Tencent Cloud account before you can use Coding project management.

Open Project

1. Log in to the CODING console and click **Use Now** to go to the CODING page.
2. On the Workspace homepage, click

on the left to go to the Continuous Deployment console.

Function Overview

CODING-CD is used to manage the project release, deployment, and delivery processes after build. It can seamlessly connect to upstream Git repositories and downstream artifact repositories to achieve automated deployment. Based on a stable technical architecture and Ops tools, it enables blue/green deployment, grayscale release (canary release), rolling release, and fast rollback.
The following Demo project shows how to use the CODING-CD console to release an application to a Tencent Cloud cluster.

Preparation

Configure the permissions required for operations in CODING-CD.
Prepare a Kubernetes cluster that is accessible to CODING-CD. Learn how to apply for Tencent Cloud Standard Clusters.

Import the sample code repository.

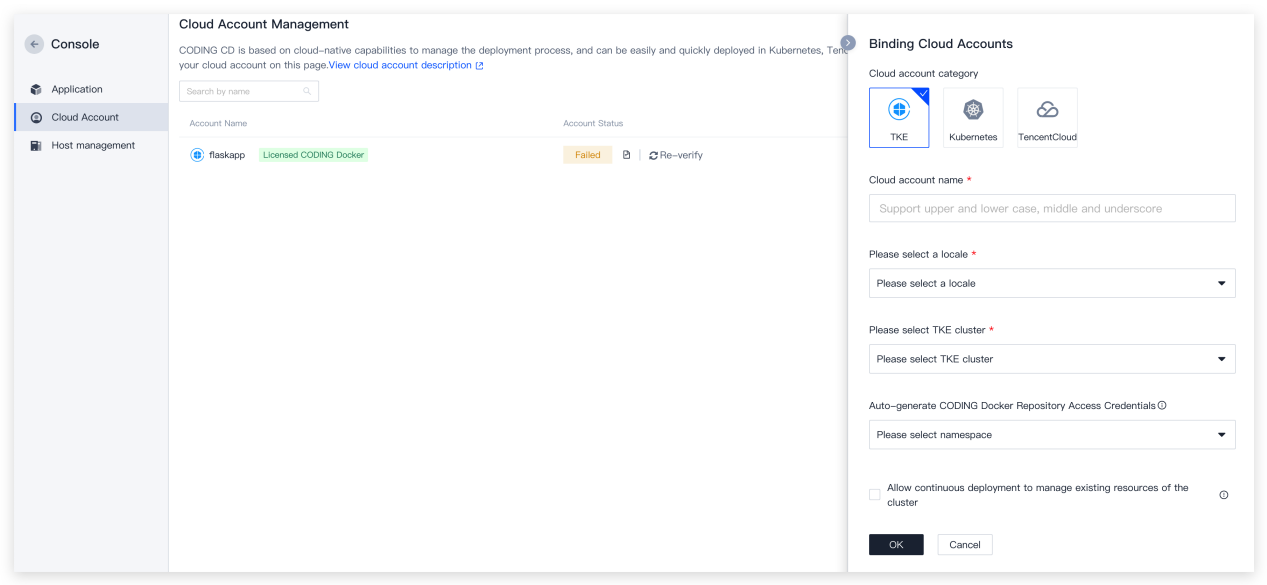
Prepare a Docker artifact repository. Learn how to use Docker Artifact Repositories in a project.

Procedure

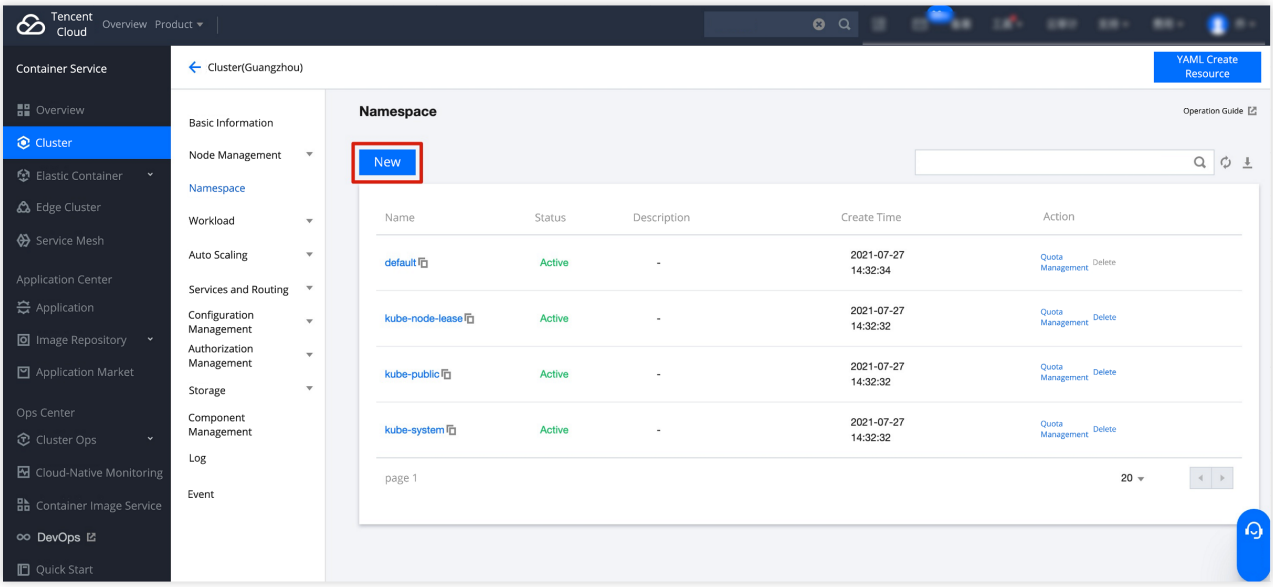
Step 1. Obtain and associate a cloud account

Because Tencent Kubernetes Engine (TKE) is used, a deployed application is released to the cluster. The team account used in the example has been associated with the Tencent Cloud account in **Team Management** > **Service Integration**.

1. Click **Deployment Console** on the left of the homepage, and bind the Tencent Cloud account in **Cloud Accounts**. You can customize your cloud account name. After selecting a region, you will automatically get the corresponding cluster.

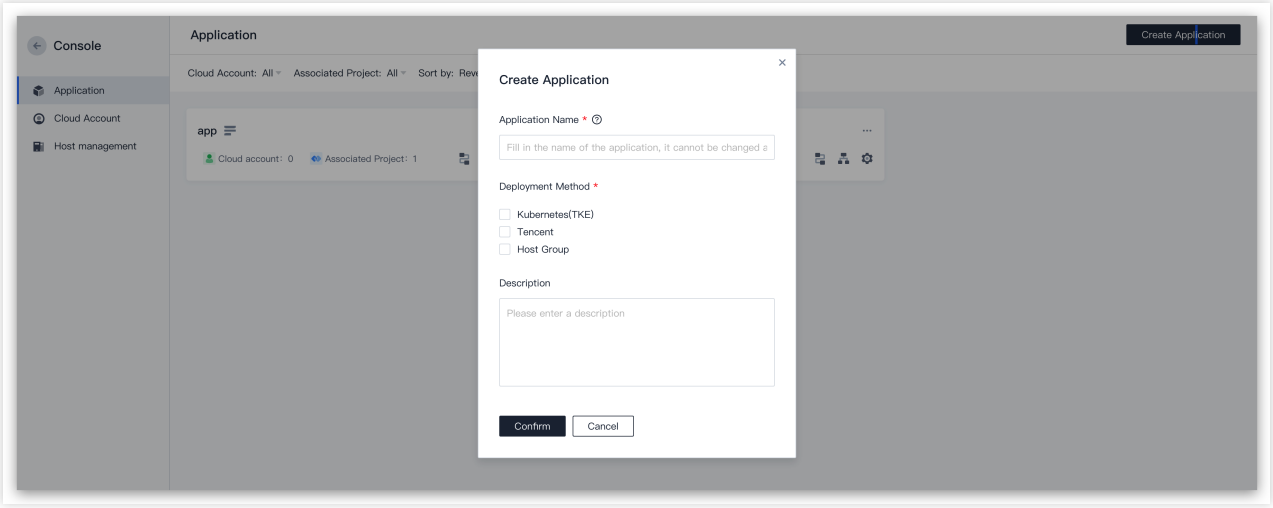


2. Automatically generated artifact repository access credentials are stored in **Namespace**. You can create new credentials in the Tencent Cloud console.

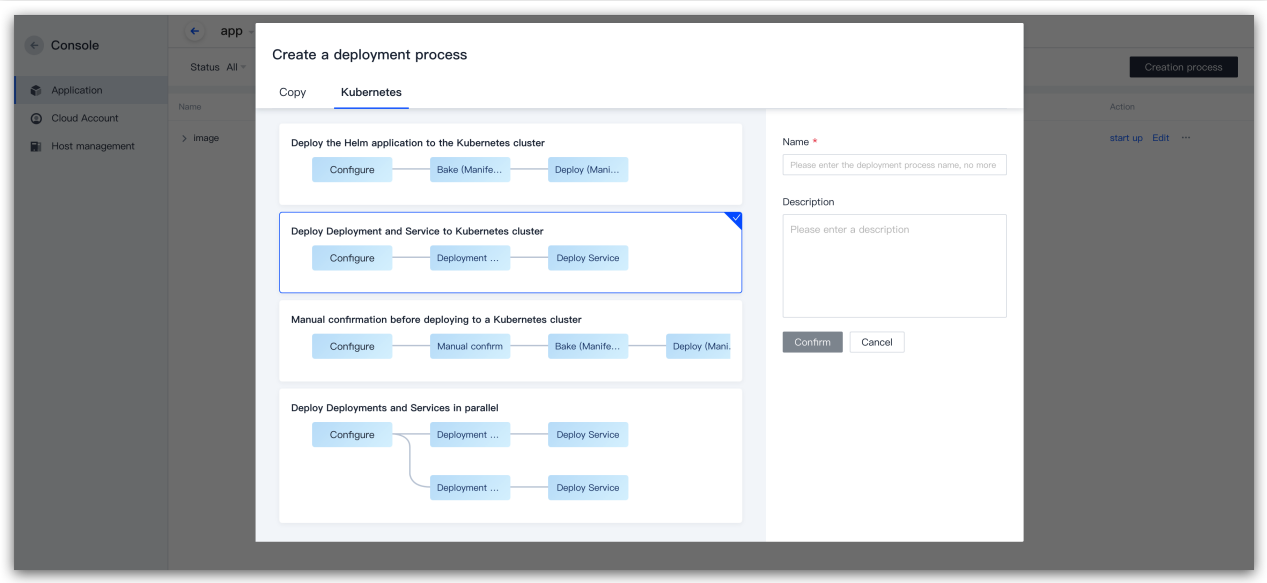


Step 2. Configure an application

1. After adding a cloud account, go to the deployment console and click **Create Application**. Then, enter the application name and select a deployment method.

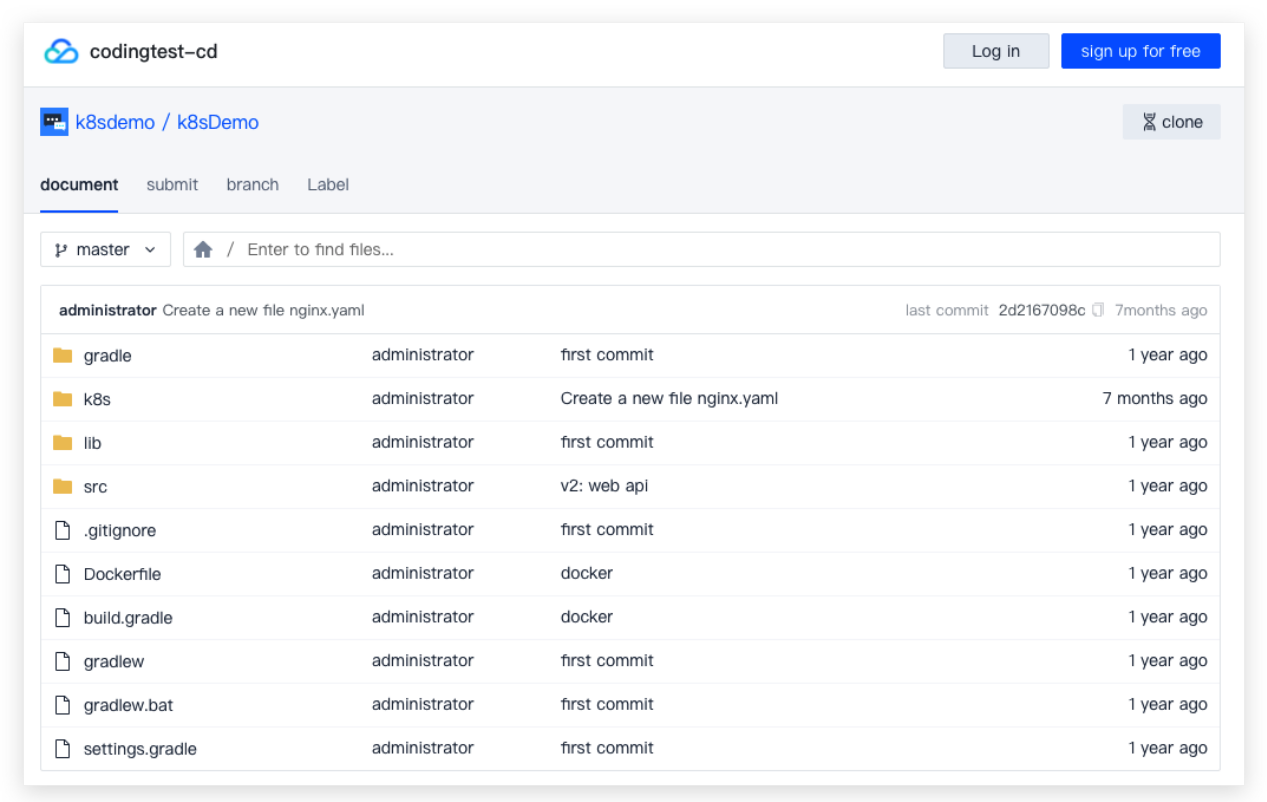


2. Select **Deploy to Kubernetes Cluster** template, and then enter the name and description to create the application.

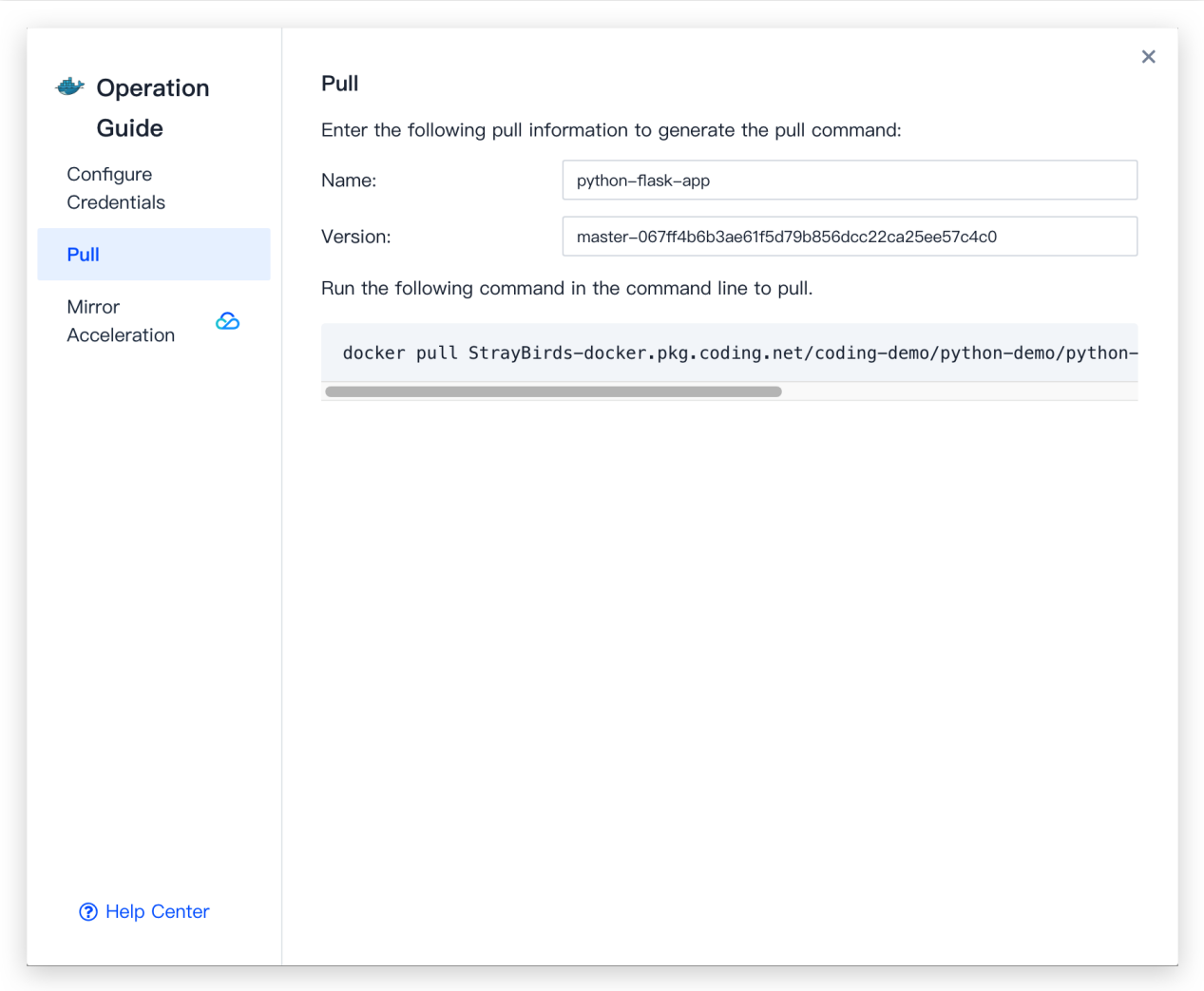


Step 3. Initialize project

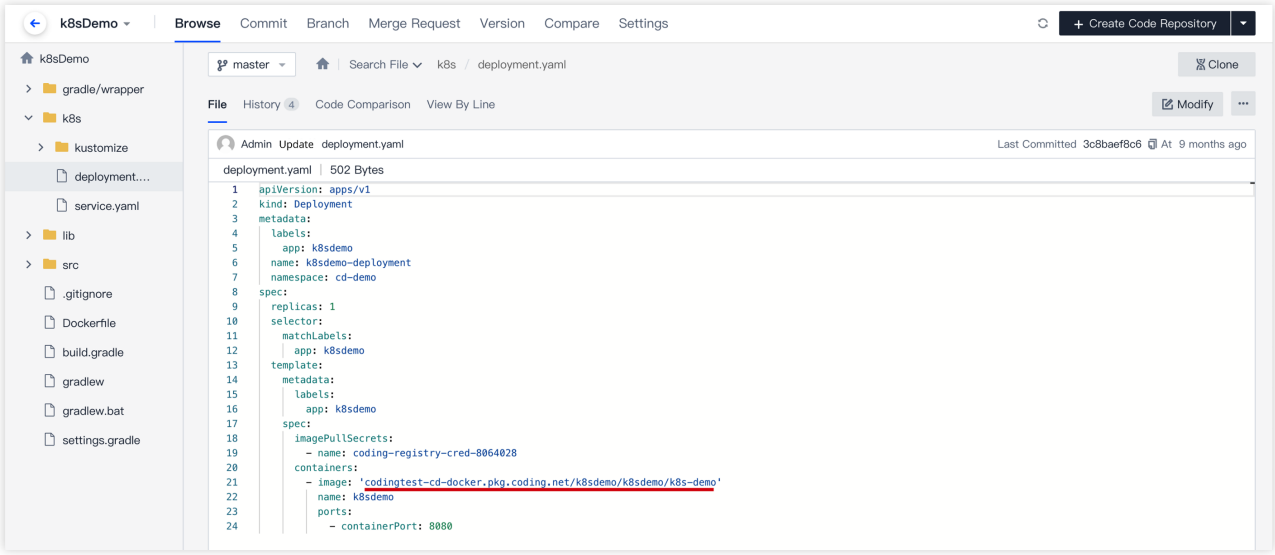
1. This step configures the code and artifact repositories involved in continuous deployment. In the **Code Repository** field, choose to import an external repository. Go to the sample repository and clone the repository address.



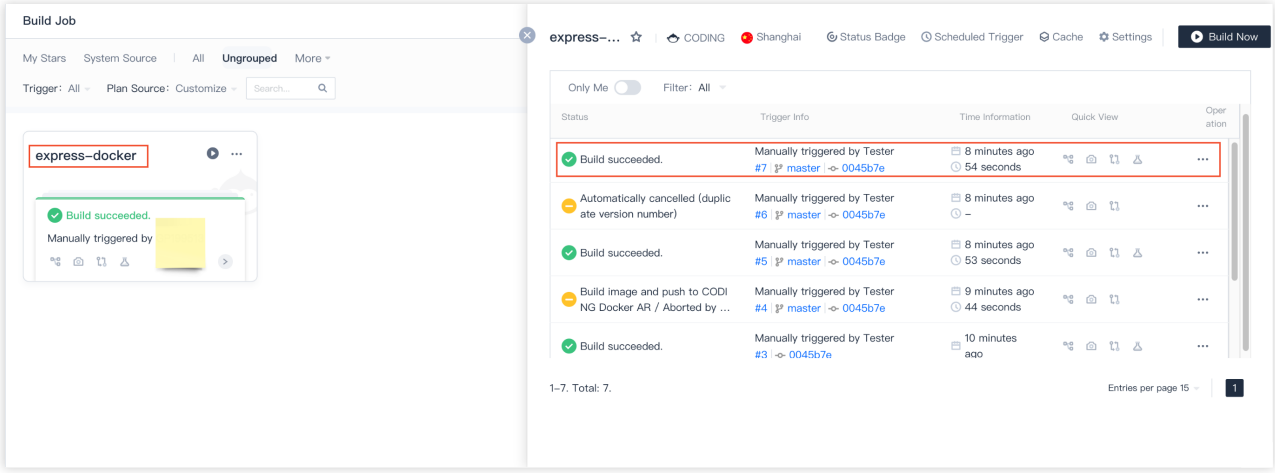
2. After the import, start to manage artifacts. Host the to-be-released Docker artifacts in the CODING artifact repository. For more information, see Docker Artifacts.



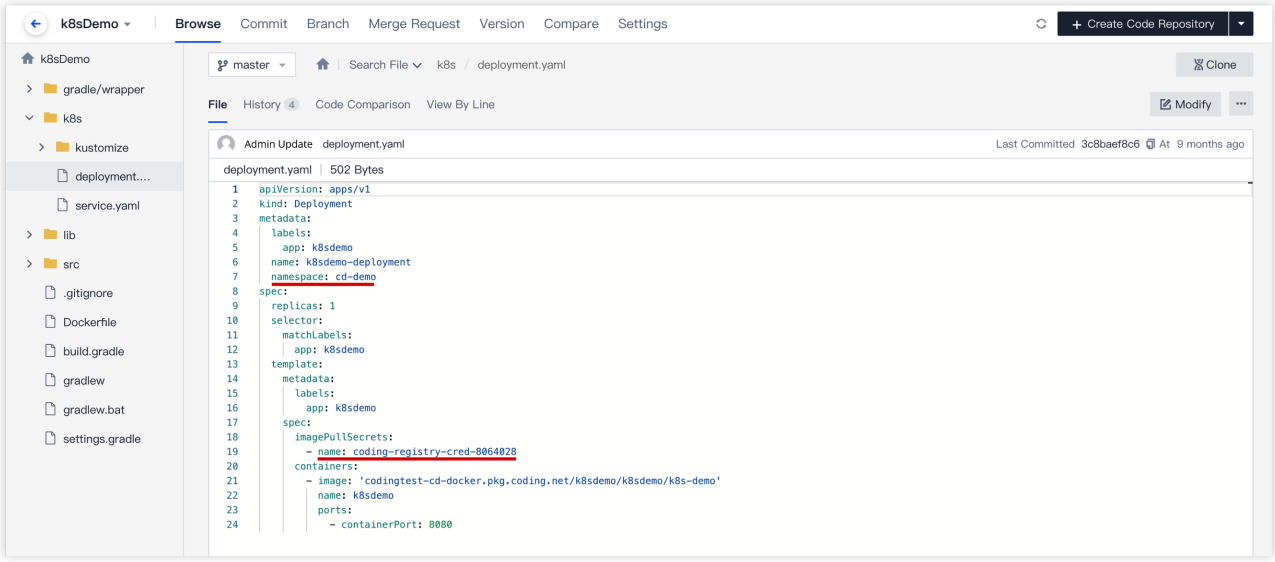
3. After pushing the artifacts to the artifact repository, get the artifact pulling address and enter it as the `image` address in the code repository's `/k8s/deployment.yaml` file.



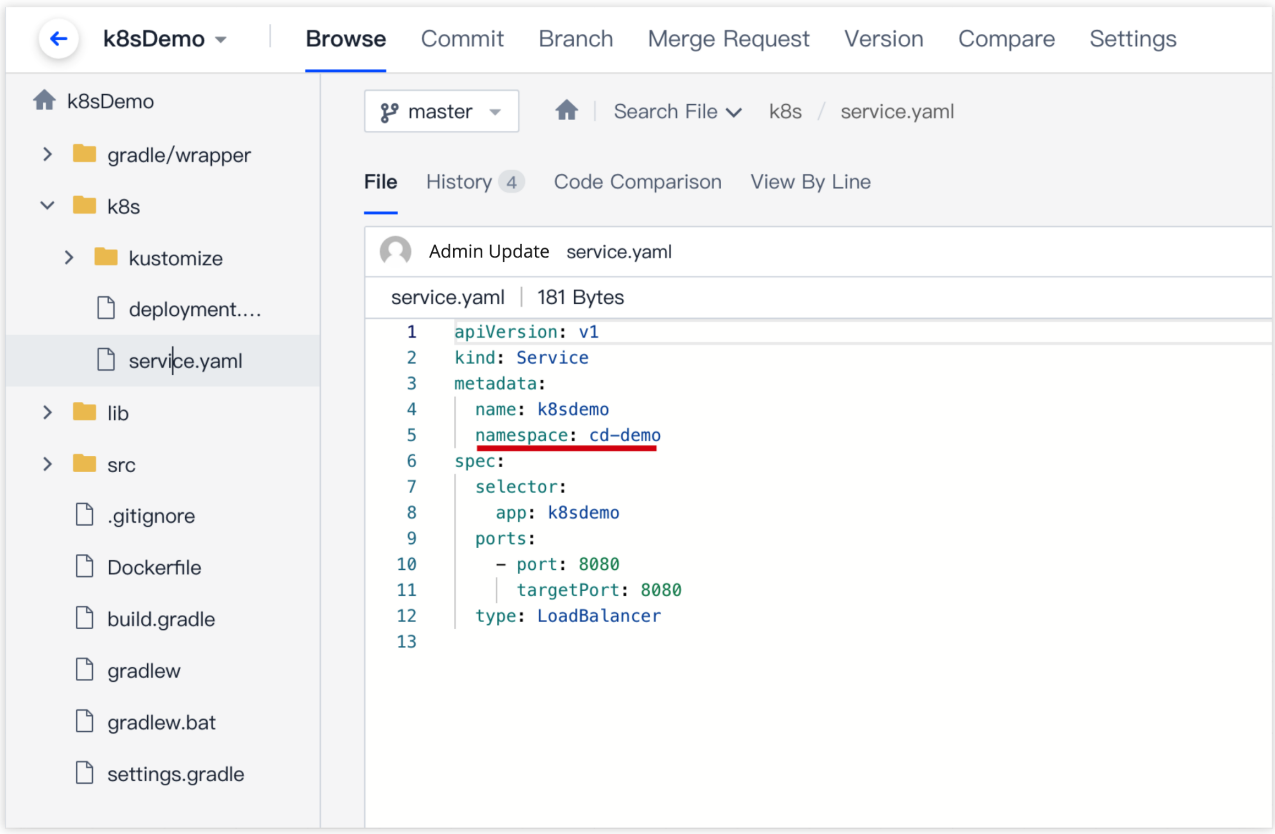
4. Next, import the cloud account's `imagePullSecrets` to the code repository. Go to **Deployment Console** > **Cloud Account**, click "View Details", and copy the name.



5. Paste the name in the `deployment.yaml` file of the code repository. Make sure that the `namespace` matches the **Namespace** specified above.



6. It must also match the `namespace` in the `service.yaml` file at the same level.



Step 4. Configure deployment pipeline

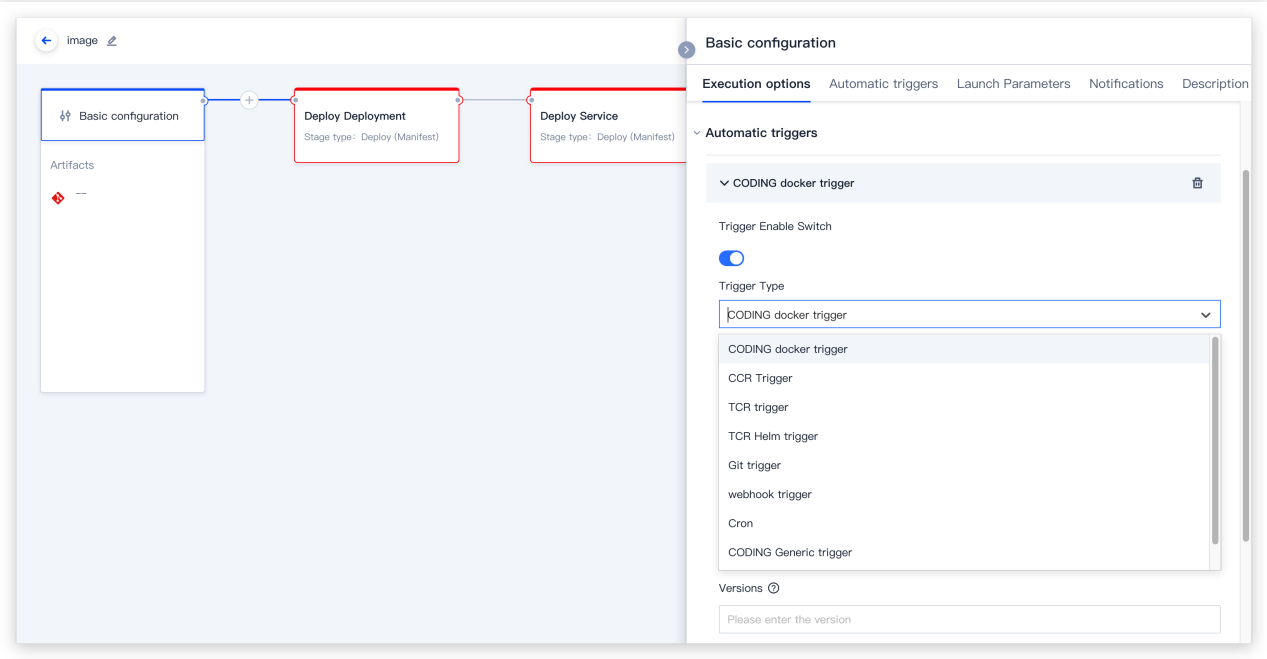
Go to the Deployment Pipeline Configuration page to set:

Pipeline execution options (in this demo, all the default values are retained).

Artifacts needed in the deployment and service deployment stages.

Manual or automatic trigger.

1. Configure the deployment (Manifest) stage. For basic settings, select the cloud account bound, select **CODING Code Repository** for "Manifest Source", enter the relevant path, and choose to automatically get the image version configuration.



2. Configure the service deployment stage by following the same steps as above. You also need to select the file path of the `k8s/service.yaml` file.

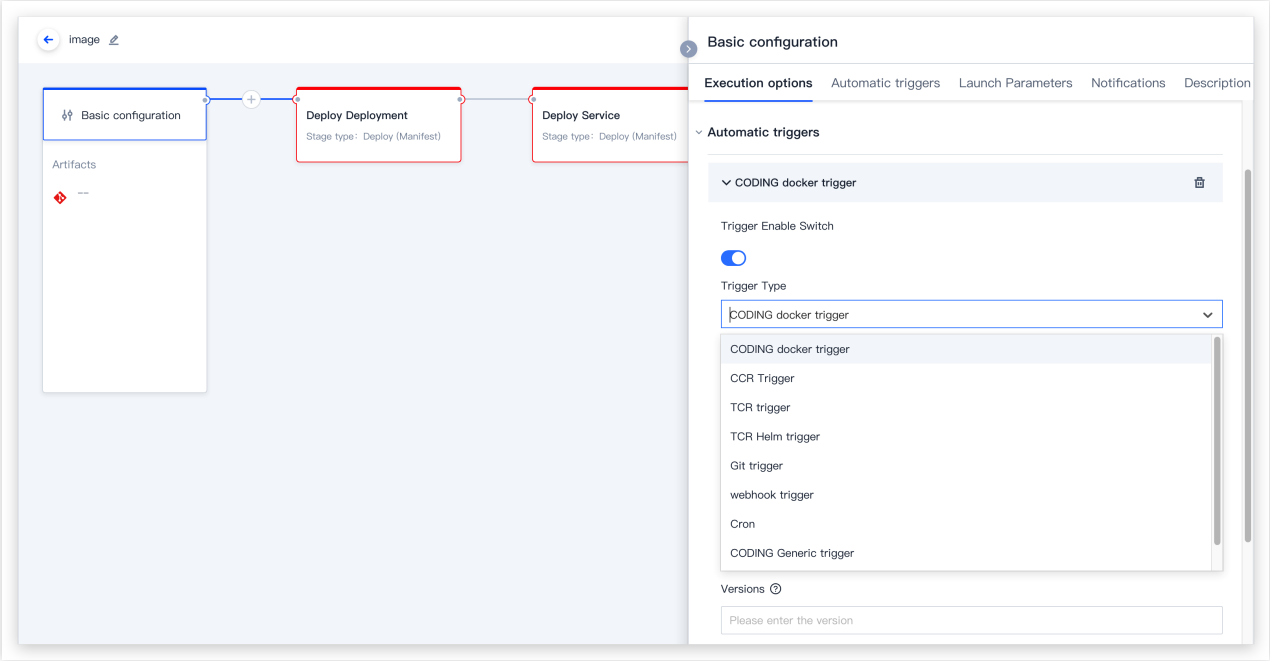
Step 5. Configure the trigger

1. After configuring the deployment stage, you can select "Auto Trigger" or "Manually Submit Release Order" as the deployment method.

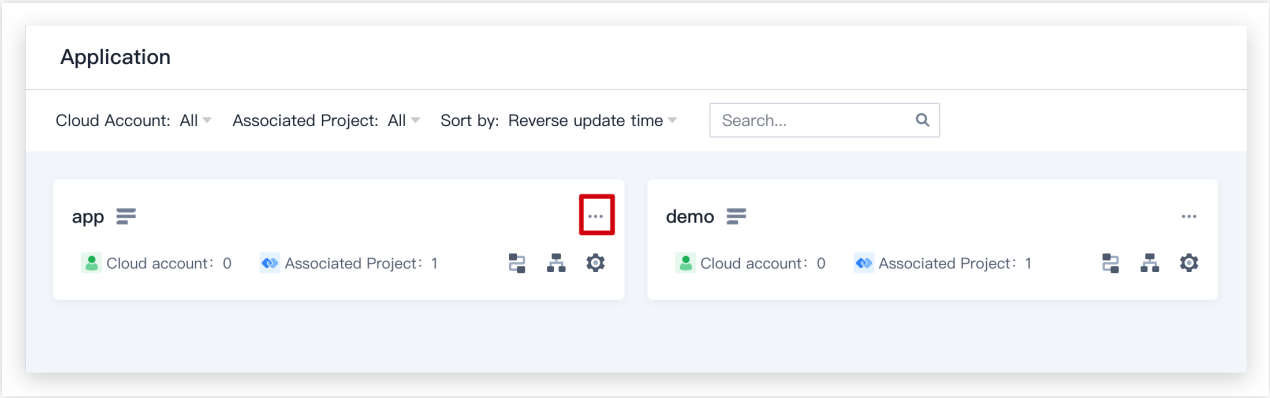
Auto Trigger

Manually Submit Release Order

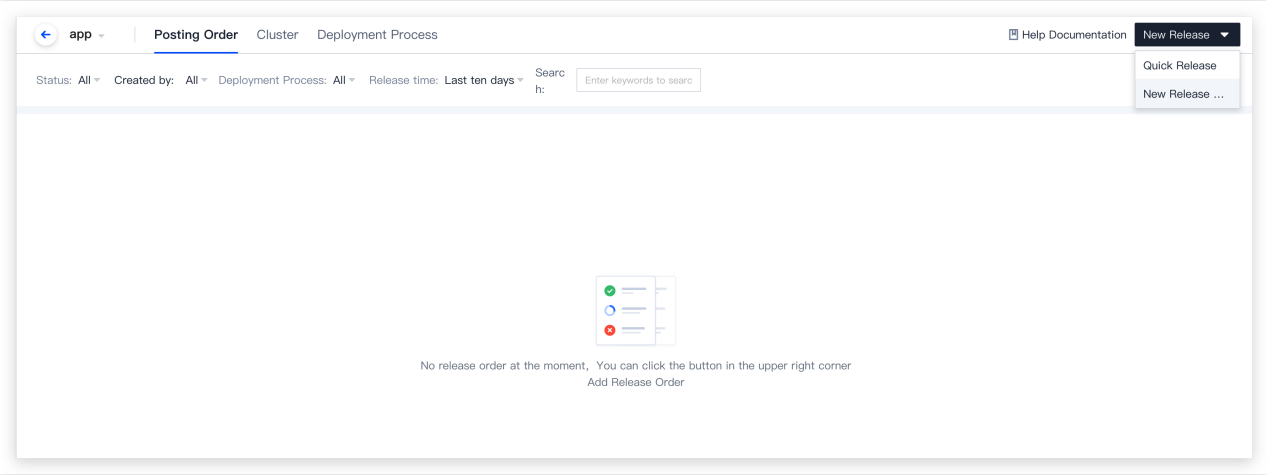
Click the trigger type in **Basic Configuration** and select Docker repository trigger. When a developer updates the code repository and uses CI to package and push the image to the artifact repository, the updates of the Docker image version will automatically trigger the deployment process and release the application to the Kubernetes (TKE) cluster. Then, you can check whether the application has been successfully released in the infrastructure page.



To trigger the deployment process by manually submitting a release order, associate the **application** (such as flaskapp in this example) with a project. Search for the project to be associated in the **App List** in the deployment console.

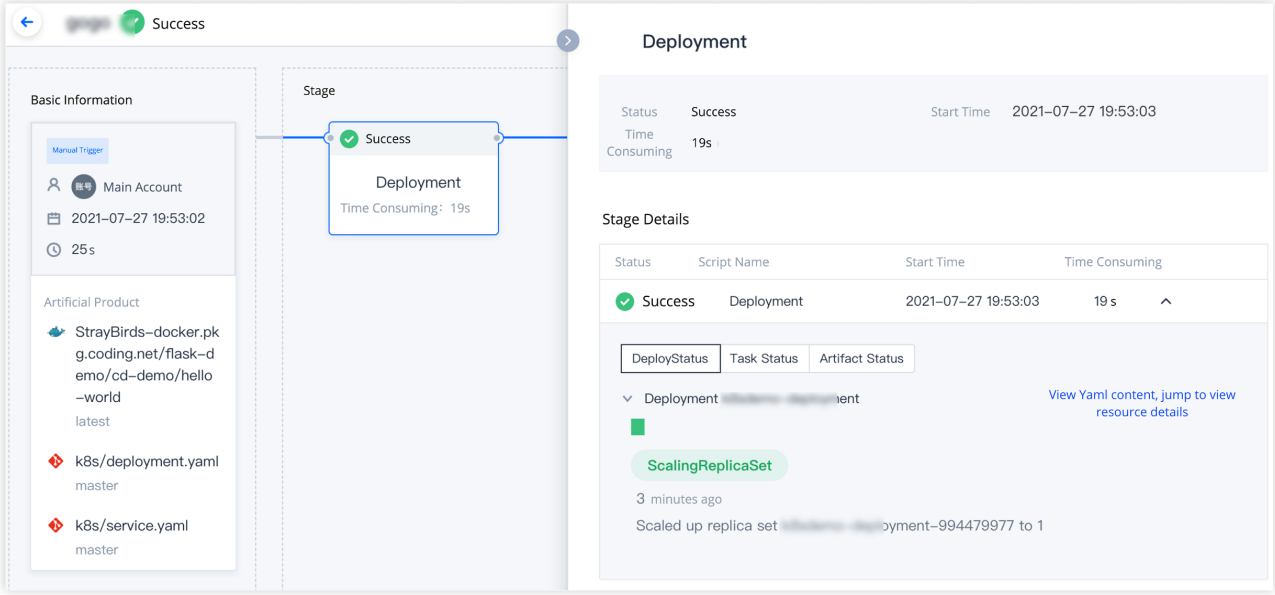


2. After association, click **Continuous Deployment** > **Kubernetes** in the project to manually submit a release order.

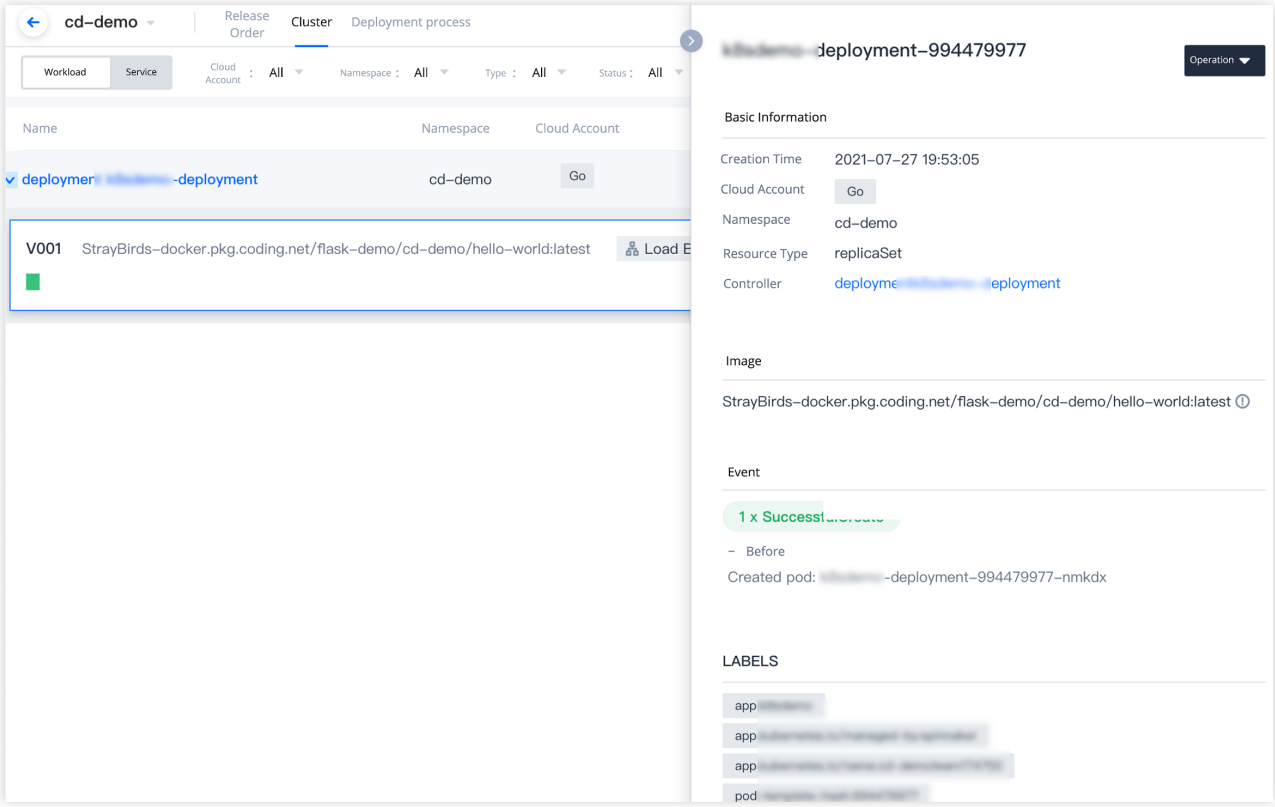


Step 6. Complete the release

1. After a successful release, you can view the released artifacts, launch parameters, and stage execution details.



2. To view the operating status of a resource in the cluster, click the workload under **Cluster** to view details (such as workload's pod instances and logs).



3. View workload in Tencent Kubernetes Engine.