CODING Continuous Deployment

FAQs

Product Documentation

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice

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

©2013-2024 Tencent Cloud. All rights reserved.

FAQs

General Questions

Last updated：2024-12-02 16:58:14

What types of artifacts does continuous deployment support?

Continuous deployment supports Docker images, Generic files, and WAR packages.

What types of clusters does continuous deployment support?

Continuous deployment supports CVM (Linux OS), TKE, and SCF clusters.

How can I protect sensitive information when configuring continuous deployment processes?

To protect tokens, SSH keys, Kubernetes certificates, and other confidential information, select **Project Settings** > **Developer Options** > **Credentials Management** in a project in CODING DevOps for Web to open the credentials management page. You can also set the continuous deployment release processes that can use the information.

How do I release source code?

Many common dynamic programming languages involve no compilation and build processes. You can configure Git repository in the application's artifact settings to specify the file path and release the source code.

How can I configure an approval process?

You can configure an approval process in two steps:

1. Open CODING DevOps for Web and select a project. In the project, select **Settings** > **Fields and Processes** > **Approval Process Settings** to go to the "Approval Process Settings" page. You can add multiple approval processes for a single project.
2. Go to **Continuous Deployment** > **Release Process** > **Associate Approval** to associate the release process with an approval process. Then, when you submit a release order, the system will automatically determine which approval process is executed in advance.

Do release orders have to go through an approval process?

No, if you do not associate any approval process with a release process, the release process requires no approval. In this case, the release order is executed immediately after it is submitted.

©2013-2024 Tencent Cloud. All rights reserved.

How to pull images from a private repository when deploying Kubernetes resources

Last updated：2024-01-03 12:00:38

This document describes how to pull images from a private repository when deploying Kubernetes resources.

Prerequisites

You must activate the CODING DevOps service for your Tencent Cloud account before you can use Coding Continuous Deployment (CD).

Open Project

1. Log in to the CODING console and click the team domain name to enter the CODING page.
2. On the Workspace homepage, click

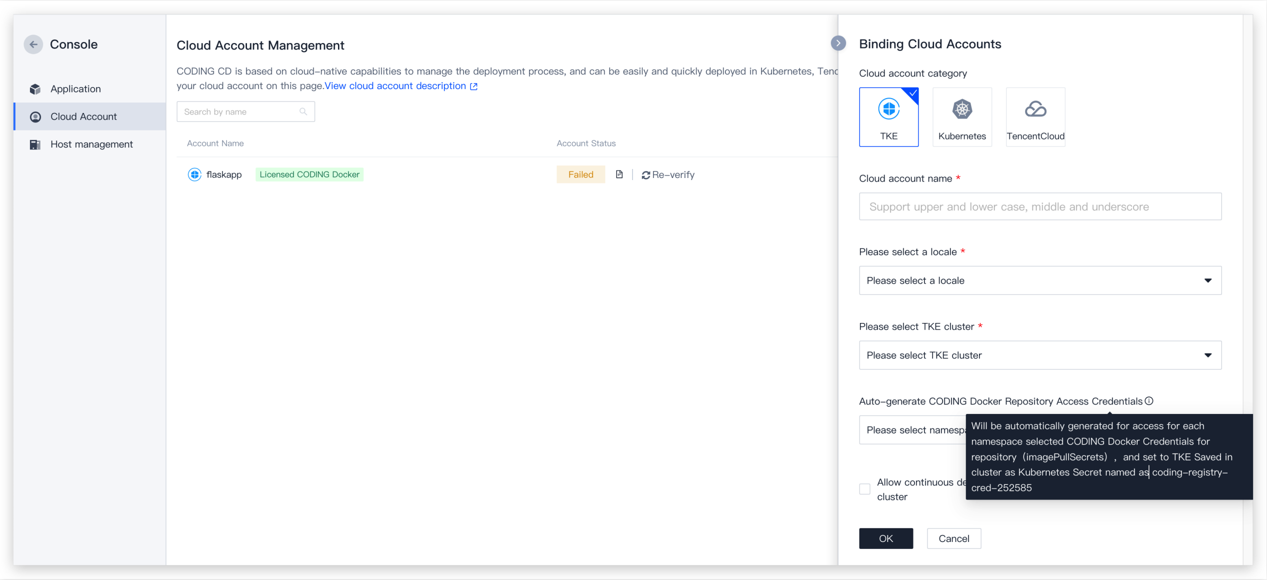
on the left to go to the Continuous Deployment console.

Feature Overview

When deploying Kubernetes resources, if the images referenced by manifest are stored in a private repository, you need to configure `imagePullSecrets` in manifest to pull the images.
The following describes how to configure `imagePullSecrets` for different types of cloud accounts:

Tencent Cloud TKE

©2013-2024 Tencent Cloud. All rights reserved.



Console

- Application
- Cloud Account
- Host management

Cloud Account Management

CODING CD is based on cloud-native capabilities to manage the deployment process, and can be easily and quickly deployed in Kubernetes, Tenc your cloud account on this page.View cloud account description

Search by name

Account Name	Account Status
flaskapp Licensed CODING Docker	Failed Re-verify

Binding Cloud Accounts

Cloud account category: TKE, Kubernetes, TencentCloud

Cloud account name *

Support upper and lower case, middle and underscore

Please select a locale *

Please select a locale

Please select TKE cluster *

Please select TKE cluster

Auto-generate CODING Docker Repository Access Credentials

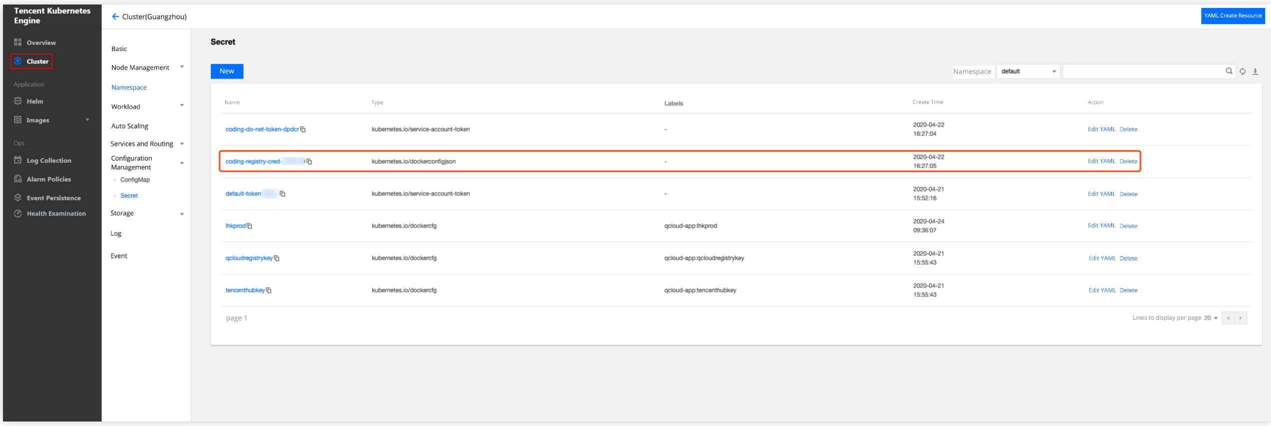
Please select namesp

Will be automatically generated for access for each namespace selected CODING Docker Credentials for repository（imagePullSecrets），and set to TKE Saved in cluster as Kubernetes Secret named as coding-registry-cred-252585

□ Allow continuous de cluster

OK Cancel

As shown above, CODING-CD generates a Secret named `coding-registry-cred-$(user_id)` in the TKE cluster. You can view the Secret information in the TKE console:



Tencent Kubernetes Engine

Cluster(Guangzhou)

Secret

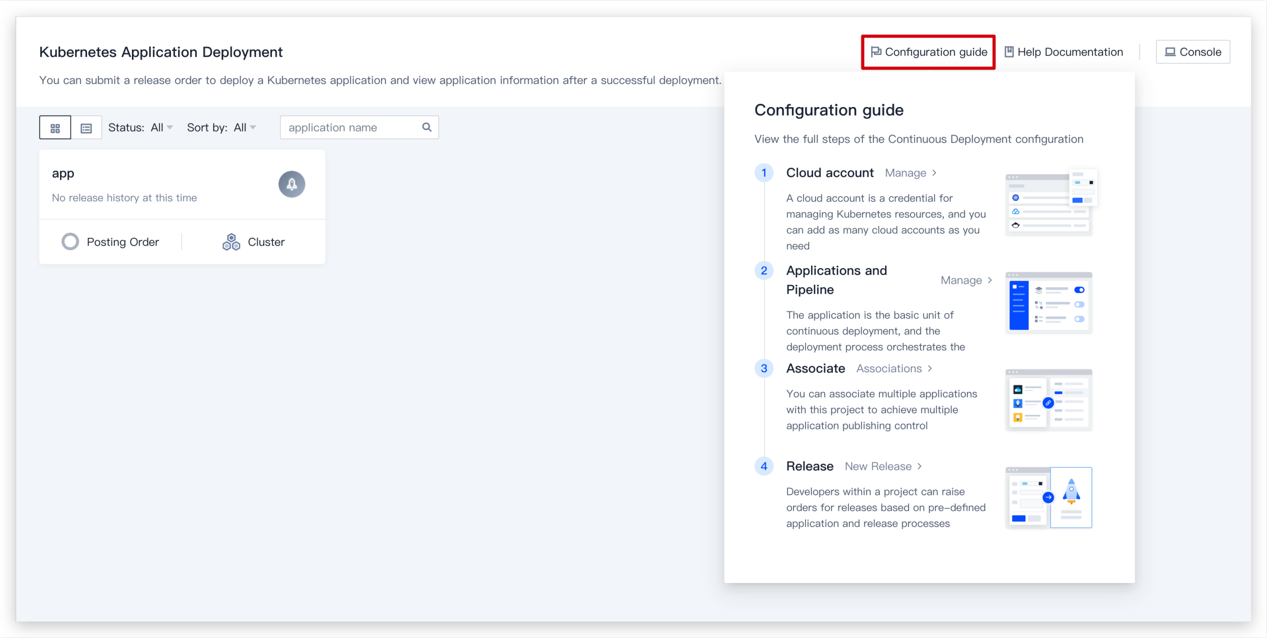
Namespace: default

Name	Type	Labels	Create Time	Action
coding-do-net-token-dpdcr	kubernetes.io/service-account-token	-	2020-04-22 16:27:04	Edit YAML Delete
coding-registry-cred-[redacted]	kubernetes.io/dockerconfigjson	-	2020-04-22 16:27:05	Edit YAML Delete
default-token[redacted]	kubernetes.io/service-account-token	-	2020-04-21 15:52:16	Edit YAML Delete
lhkprod	kubernetes.io/dockercfg	qcloud-app:lhkprod	2020-04-24 09:36:07	Edit YAML Delete
qcloudregistrykey	kubernetes.io/dockercfg	qcloud-app:qcloudregistrykey	2020-04-21 15:55:43	Edit YAML Delete
tencenthubkey	kubernetes.io/dockercfg	qcloud-app:tencenthubkey	2020-04-21 15:55:43	Edit YAML Delete

page 1

Lines to display per page 20

After a cloud account is added, you can view the sample usage：



Kubernetes Application Deployment

You can submit a release order to deploy a Kubernetes application and view application information after a successful deployment.

Configuration guide

View the full steps of the Continuous Deployment configuration

1 Cloud account Manage >

A cloud account is a credential for managing Kubernetes resources, and you can add as many cloud accounts as you need

2 Applications and Pipeline Manage >

The application is the basic unit of continuous deployment, and the deployment process orchestrates the

3 Associate Associations >

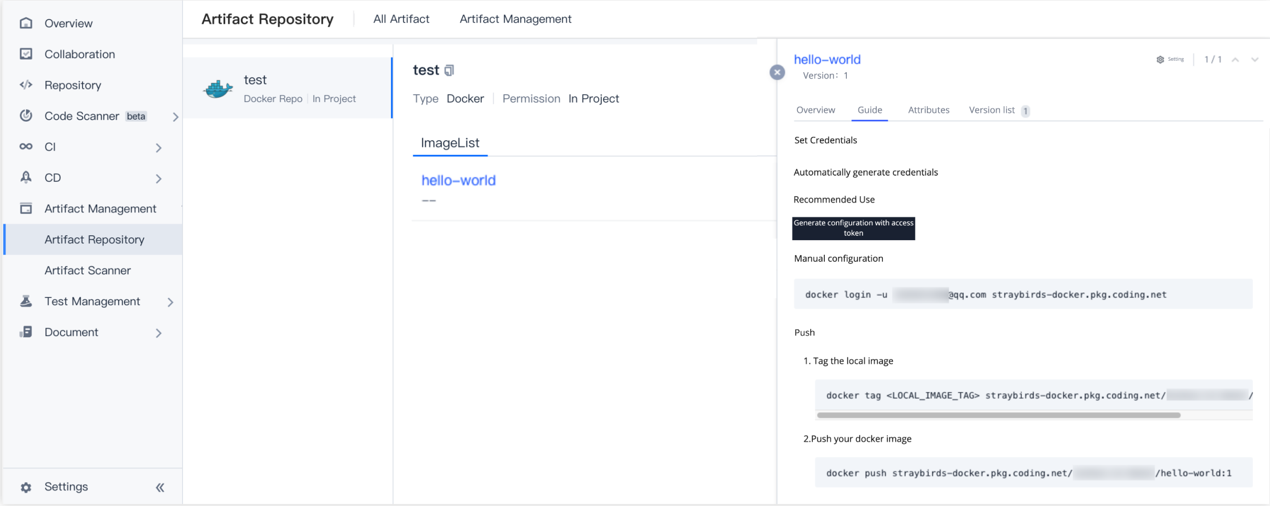
You can associate multiple applications with this project to achieve multiple application publishing control

4 Release New Release >

Developers within a project can raise orders for releases based on pre-defined application and release processes

Kubernetes cloud accounts (non-TKE cluster)

For a Kubernetes cloud account added with Kubeconfig or Service Account credentials, you need to create a Secret in the Kubernetes cluster before manifest can reference images from a private repository (CODING artifact repository is taken as an example here):



test

Type Docker Permission In Project

hello-world

Version: 1

Set Credentials

Automatically generate credentials

Recommended Use

Manual configuration

```
docker login -u [redacted]@qq.com straybirds-docker.pkg.coding.net
```

Push

1. Tag the local image

```
docker tag <LOCAL_IMAGE_TAG> straybirds-docker.pkg.coding.net/[redacted]/
```

2.Push your docker image

```
docker push straybirds-docker.pkg.coding.net/[redacted]/hello-world:1
```

If you directly reference images from the private repository in mainfest without generating a Secret in the cluster, the operation will fail.

```
kubectl create secret docker-registry coding-regcred \\
--docker-server=Your team domain --docker-username=Your email --docker-password=$(p
```

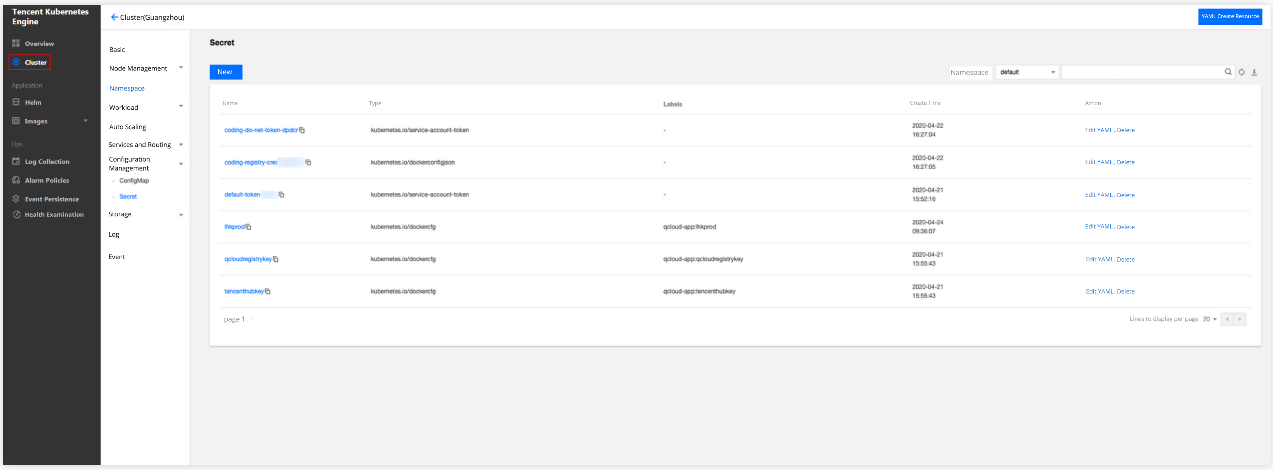
After a Secret is generated, use `imagePullSecrets` in manifest to configure the Secret for pulling images (the last two lines).

```
apiVersion: apps/v1
kind: Deployment
...
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
      imagePullSecrets:
      - name: coding-regcred
```

For more information, see How to pull images from a private repository in Kubernetes

Kubernetes cloud accounts (TKE cluster)

If you add a TKE cluster cloud account using a Kubeconfig or Service Account, you can directly create a Secret on the TKE console. Go to the cluster Information page, and select **Configuration Management** > **Secret** > **New**:

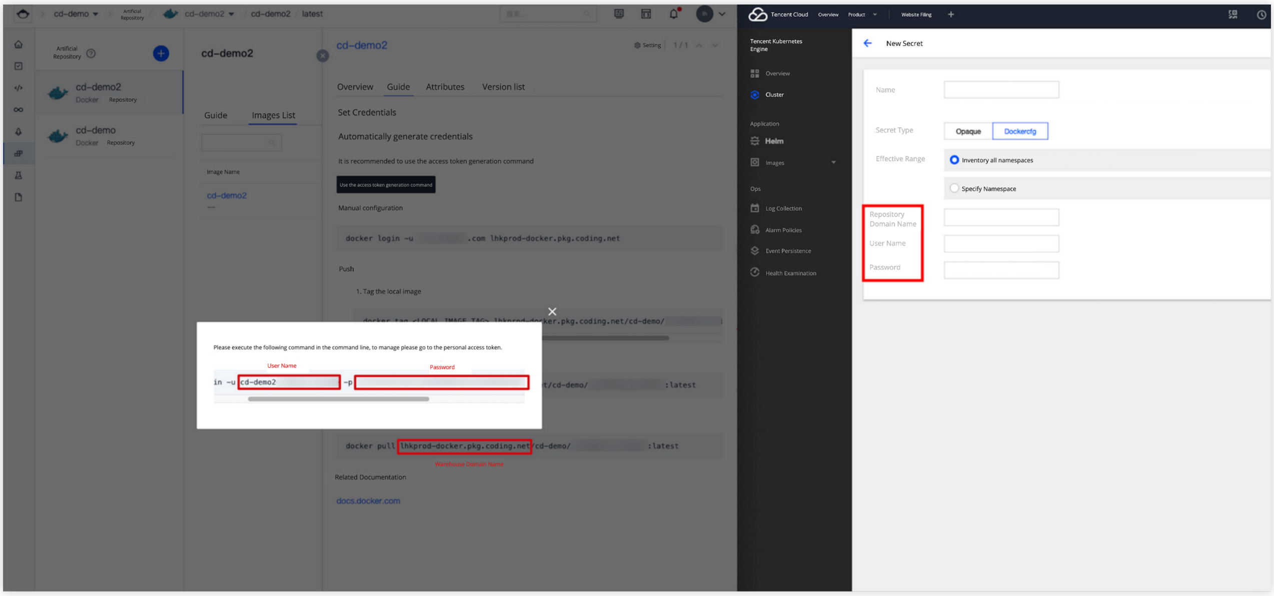


Name	Type	Labels	Create Time	Action
coding-do-net-token-dpdcr	kubernetes.io/service-account-token	-	2020-04-22 16:27:04	Edit YAML, Delete
coding-registry-crec	kubernetes.io/dockerconfigjson	-	2020-04-22 16:27:05	Edit YAML, Delete
default-token-	kubernetes.io/service-account-token	-	2020-04-21 15:52:16	Edit YAML, Delete
lhkprod	kubernetes.io/dockercfg	qcloud-app:lhkprod	2020-04-24 09:36:07	Edit YAML, Delete
qcloudregistrykey	kubernetes.io/dockercfg	qcloud-app:qcloudregistrykey	2020-04-21 15:55:43	Edit YAML, Delete
tencenthubkey	kubernetes.io/dockercfg	qcloud-app:tencenthubkey	2020-04-21 15:55:43	Edit YAML, Delete

page 1
Lines to display per page 20

Fill in the form as follows:

©2013-2024 Tencent Cloud. All rights reserved.



Set Credentials

Automatically generate credentials

It is recommended to use the access token generation command

Use the access token generation command

Manual configuration

```
docker login -u [redacted].com lhkprod-docker.pkg.coding.net
```

Push

1. Tag the local image

```
docker tag <LOCAL_IMAGE_TAG> lhkprod-docker.pkg.coding.net/cd-demo/[redacted]
```

```
t/cd-demo/[redacted]:latest
```

Please execute the following command in the command line, to manage please go to the personal access token.

User Name Password

```
in -u cd-demo2[redacted] -p [redacted]
```

```
docker pull lhkprod-docker.pkg.coding.net/cd-demo/[redacted]:latest
```

Warehouse Domain Name

Name

Secret Type: Opaque [✓ Dockercfg]

Effective Range: ● Inventory all namespaces ○ Specify Namespace

Repository Domain Name

User Name

Password

Similarly, after a Secret is generated, use `imagePullSecrets` in manifest to configure the secret for pulling images (the last two lines).

```
apiVersion: apps/v1
kind: Deployment
...
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
      imagePullSecrets:
      - name: coding-regcred
```

Minimum permissions required for Kubernetes cloud accounts

Last updated：2024-01-03 12:01:05

This document describes how to configure the minimum permissions required for Kubernetes cloud accounts in CODING.

Prerequisites

You must activate the CODING DevOps service for your Tencent Cloud account before you can use Coding Continuous Deployment (CD).

Open Project

1. Log in to the CODING console and click the team domain name to enter the CODING page.
2. On the Workspace homepage, click

on the left to go to the Continuous Deployment console.

Feature description

If you want to release apps in a Kubernetes scenario (using a K8s account), CODING-CD must be able to call Kubernetes APIs. We do not recommend you grant all permissions for the Kubernetes cluster to CODING-CD. Using Kubernetes' Role Based Access Control (RBAC), you can grant CODING-CD the minimum permissions required to release apps. The following describes how to configure mini permissions.

Role

We recommend you create a `Role` in the namespace for which you will grant permissions and bind a `ServiceAccount` to the `Role`.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: coding-cd-role
```

```
rules:
- apiGroups: [""]
  resources: ["namespaces", "configmaps", "events", "replicationcontrollers", "serv
  verbs: ["get", "list"]
- apiGroups: [""]
  resources: ["pods", "pods/portforward", "services", "services/proxy", "secrets"]
  verbs: ["*"]
- apiGroups: ["autoscaling"]
  resources: ["horizontalpodautoscalers"]
  verbs: ["list", "get"]
- apiGroups: ["apps"]
  resources: ["controllerrevisions", "statefulsets"]
  verbs: ["list"]
- apiGroups: ["extensions", "app", "apps"]
  resources: ["deployments", "replicasets", "ingresses", "daemonsets"]
  verbs: ["*"]
```

Service Account

Next, create a `Service Account` for CODING-CD. The Continuous Deployment console uses the `Service Account` to interact with the Kubernetes cluster. You can use the following manifest to create a `Service Account` .

```
apiVersion: v1
kind: ServiceAccount
metadata:
 name: coding-cd-service-account
 namespace: default
```

Role Binding

Finally, create a `RoleBinding` to bind the above `coding-cd-role` to `coding-cd-service-account` .

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: coding-cd-role-
 namespace: webapp
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: Role
 name: coding-cd-role
subjects:
- namespace: default
 kind: ServiceAccount
 name: coding-cd-service-account
```

How to convert certificate files in Kubeconfig to certificate data

Last updated：2022-03-30 10:23:07

This document describes how to convert certificate files in Kubeconfig to certificate data.

Prerequisites

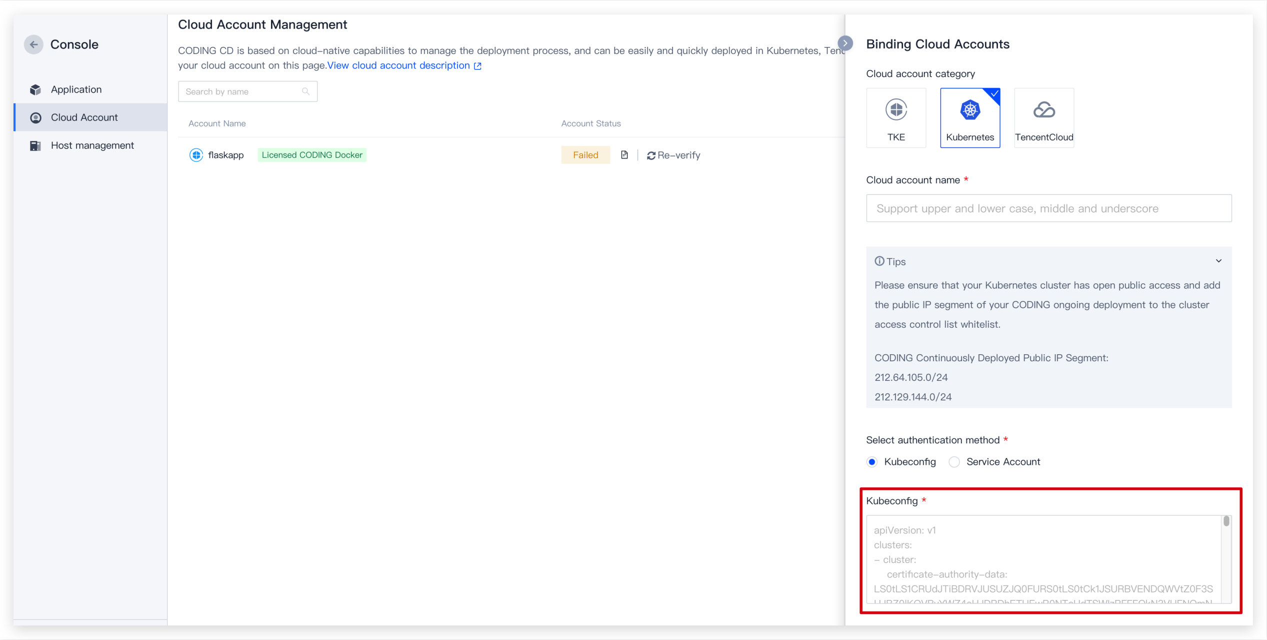
You must activate the CODING DevOps service for your Tencent Cloud account before you can use Coding Continuous Deployment (CD).

Open Project

1. Log in to the CODING console and click the team domain name to enter the CODING page.
2. On the Workspace homepage, click on the left to go to the Continuous Deployment console.

Feature Overview

When adding a Kubernetes cloud account, you can select Kubeconfig or Service Account authentication method:



Cloud Account Management

CODING CD is based on cloud–native capabilities to manage the deployment process, and can be easily and quickly deployed in Kubernetes, Tent
your cloud account on this page.View cloud account description

Search by name

Account Name	Account Status
flaskapp Licensed CODING Docker	Failed Re–verify

Binding Cloud Accounts

Cloud account category

TKE | Kubernetes ✓ | TencentCloud

Cloud account name *

Support upper and lower case, middle and underscore

Tips

Please ensure that your Kubernetes cluster has open public access and add the public IP segment of your CODING ongoing deployment to the cluster access control list whitelist.

CODING Continuously Deployed Public IP Segment:
212.64.105.0/24
212.129.144.0/24

Select authentication method *

● Kubeconfig ○ Service Account

Kubeconfig *

```
apiVersion: v1
clusters:
– cluster:
    certificate–authority–data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURBVENDQWVtZ0F3S
```

For authentication using Kubeconfig certificate files (in the example below, `certificate-authority` , `client-certificate` , and `client-key` are all specific files that contain certificate information), you need to convert the certificate files to base64-encoded strings before adding a cloud account.

```
apiVersion: v1
kind: Config
clusters:
- cluster:
    certificate-authority: */ca.crt
    server: https://******
  name: demo
contexts:
- context:
    cluster: demo
    user: demo
  name: demo
current-context: demo
preferences: {}
users:
- name: demo
  user:
    client-certificate: */client.crt
    client-key: */client.key
```

The section below explains how to perform the conversion.

Get the base64-encoded string of the certificate file

Run the following commands:

OS X

```
cat "certificate file" | base64
```

Linux

```
cat "certificate file" | base64
```

Windows

```
certutil -f -encode "certificate file" "output file"
```

Update Kubeconfig file

1. Change `certificate-authority` to `certificate-authority-data` , and enter the base64-encoded string of the `*/ca.crt` certificate file.
2. Change `client-certificate` to `client-certificate-data` , and enter the base64-encoded string of the `*/client.crt` certificate file.
3. Change `client-key` to `client-key-data` , and enter the base64-encoded string of the `*/client.key` certificate file.

The result is as follows:

```
apiVersion: v1
kind: Config
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZADQFURS0tLS0tCk1JSUM1ekNDQWM
    server: https://******
  name: demo
contexts:
- context:
    cluster: demo
    user: demo
  name: demo
current-context: demo
preferences: {}
users:
- name: demo
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURBRENDQWVpZ0
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVQSLS0tLQpNSUlFcEFJQkFBS0NBUU
```

©2013-2024 Tencent Cloud. All rights reserved.