

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



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

FAQs

General Questions

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

Minimum permissions required for Kubernetes cloud accounts

How to convert certificate files in Kubeconfig to certificate data

FAQs

General Questions

Last updated : 2022-02-23 15:32:41

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.

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



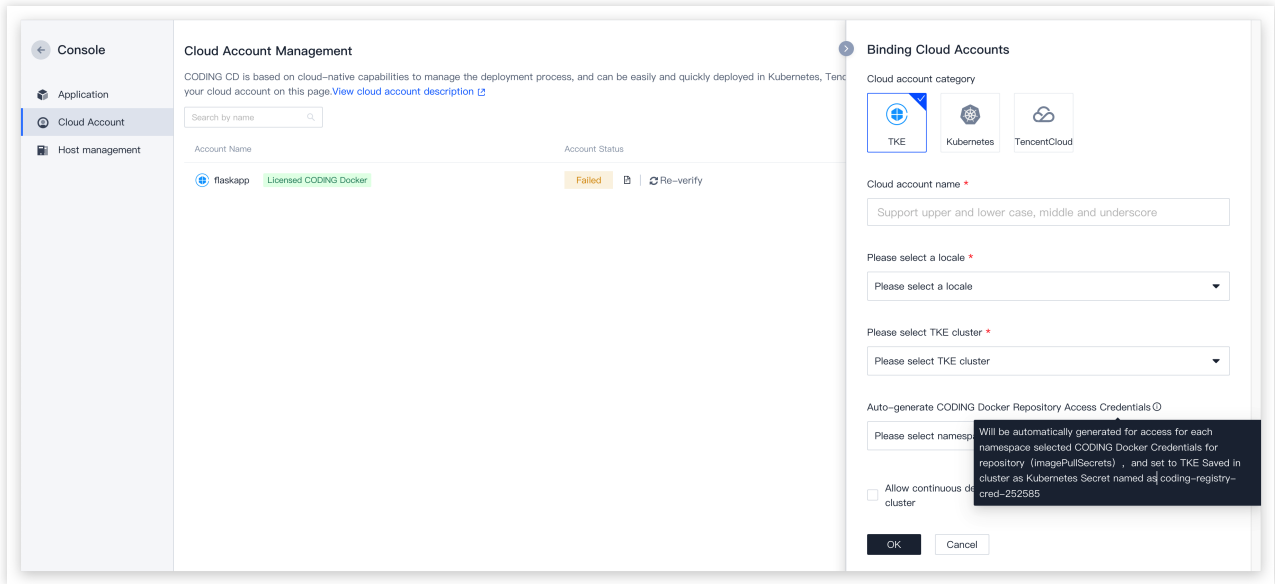
on the left to go to the Continuous Deployment console.

Feature Overview

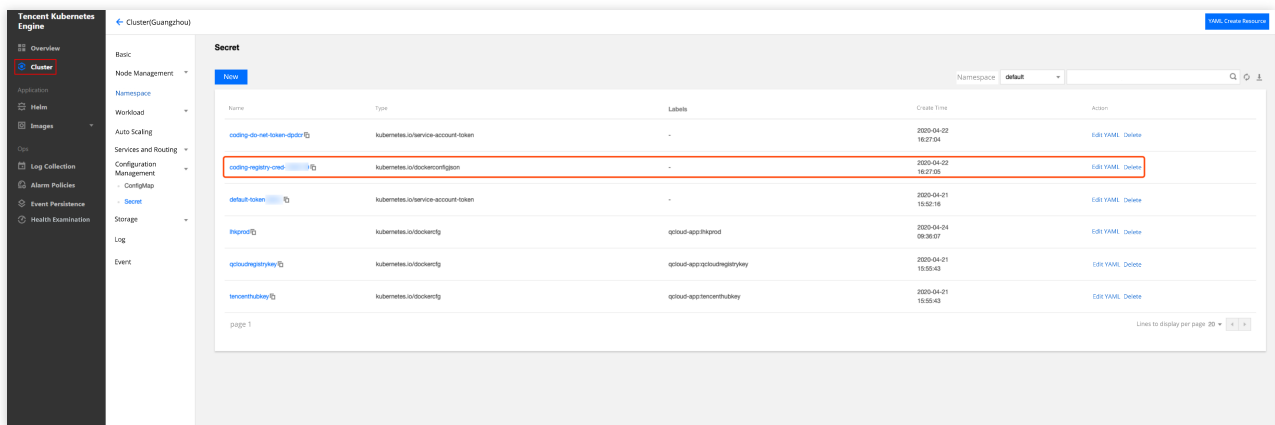
When deploying Kubernetes resources, if the images referenced by manifest are stored in a private repository, you need to configure `imagePullSecrets` in manifest to pull the images.

The following describes how to configure `imagePullSecrets` for different types of cloud accounts:

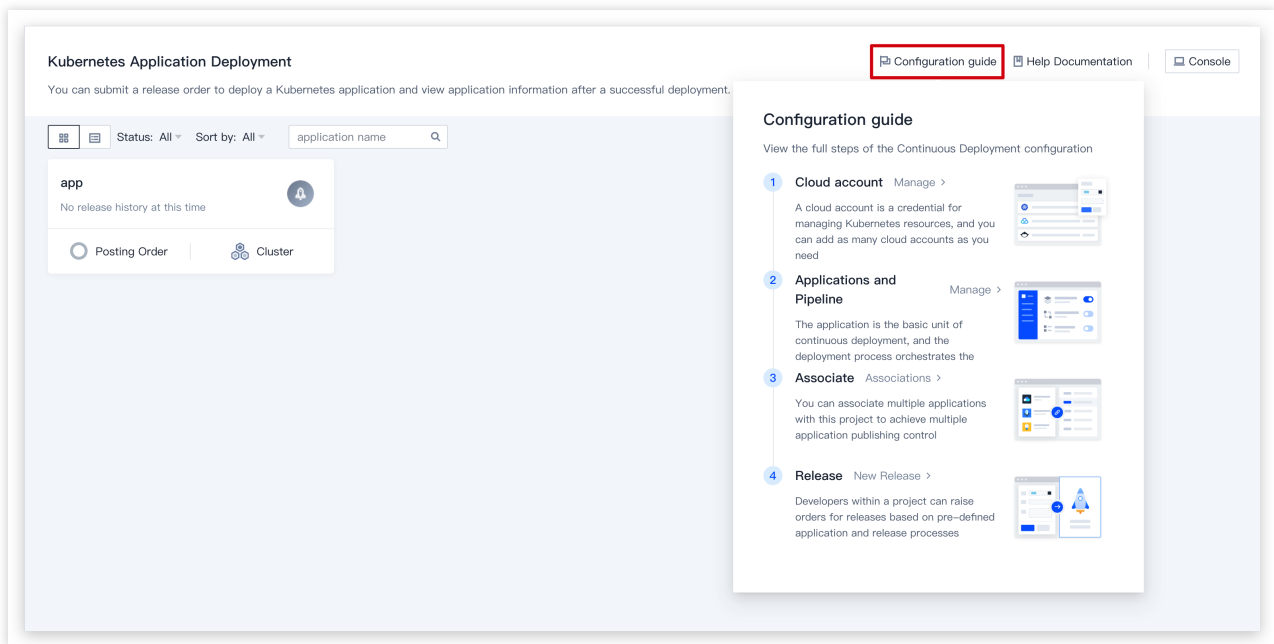
Tencent Cloud TKE



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:

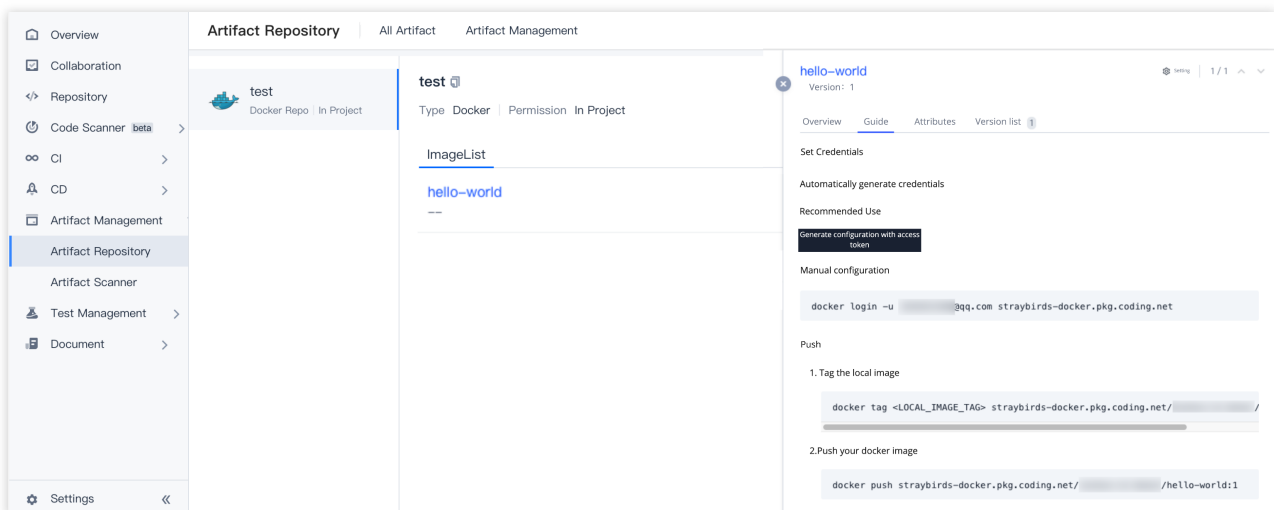


After a cloud account is added, you can view the sample usage :



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):



If you directly reference images from the private repository in manifest 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).

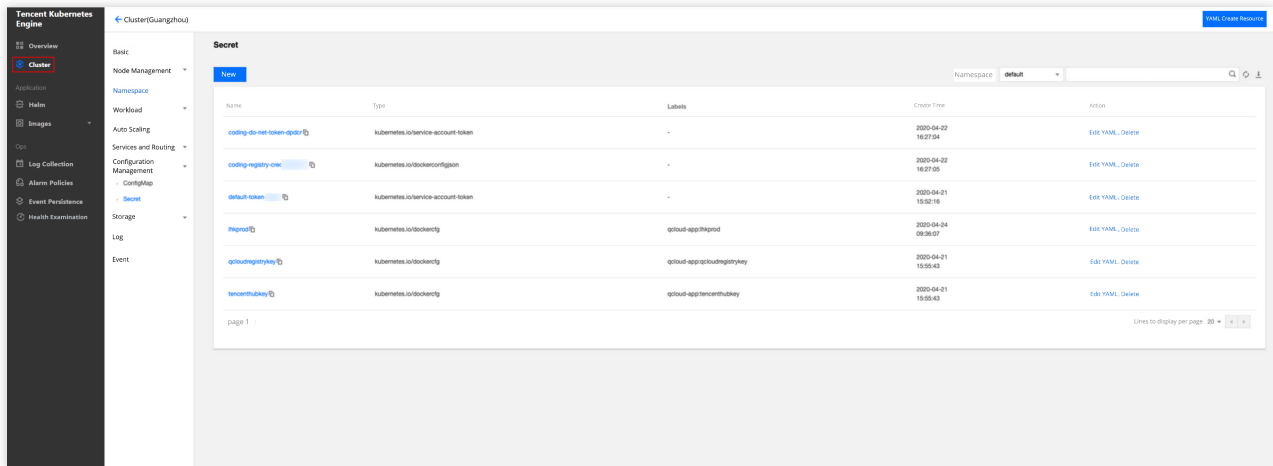


```
apiVersion: apps/v1
kind: Deployment
...
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
  imagePullSecrets:
  - name: coding-regcred
```

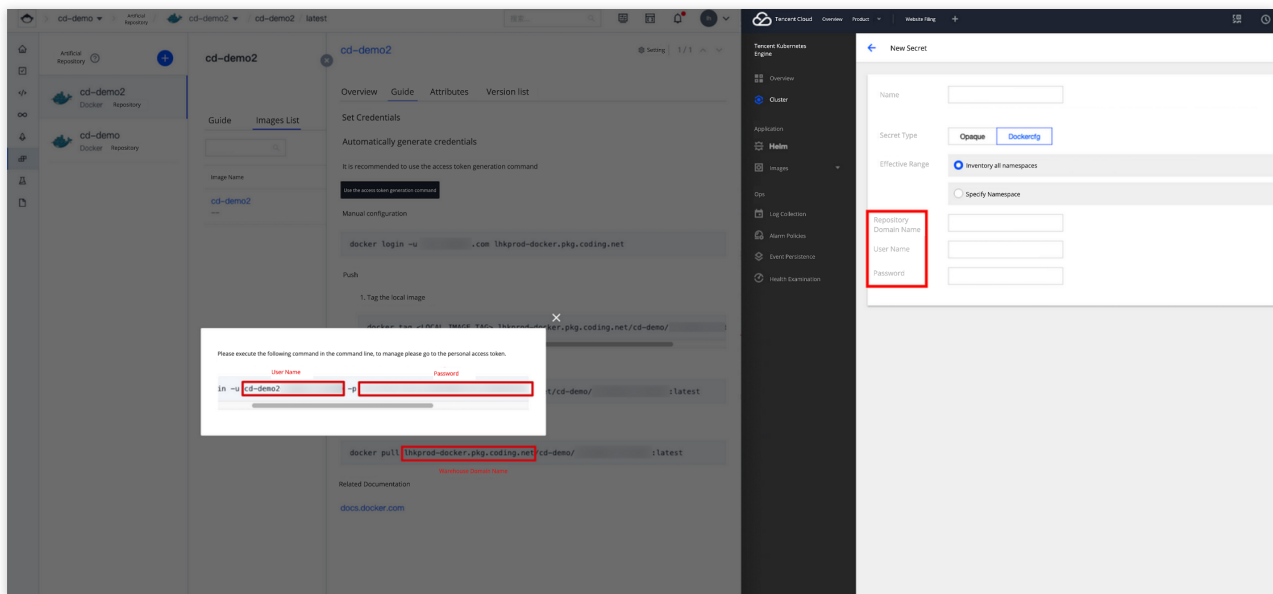
For more information, see [How to pull images from a private repository in Kubernetes](#)

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**:



Fill in the form as follows:



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`.

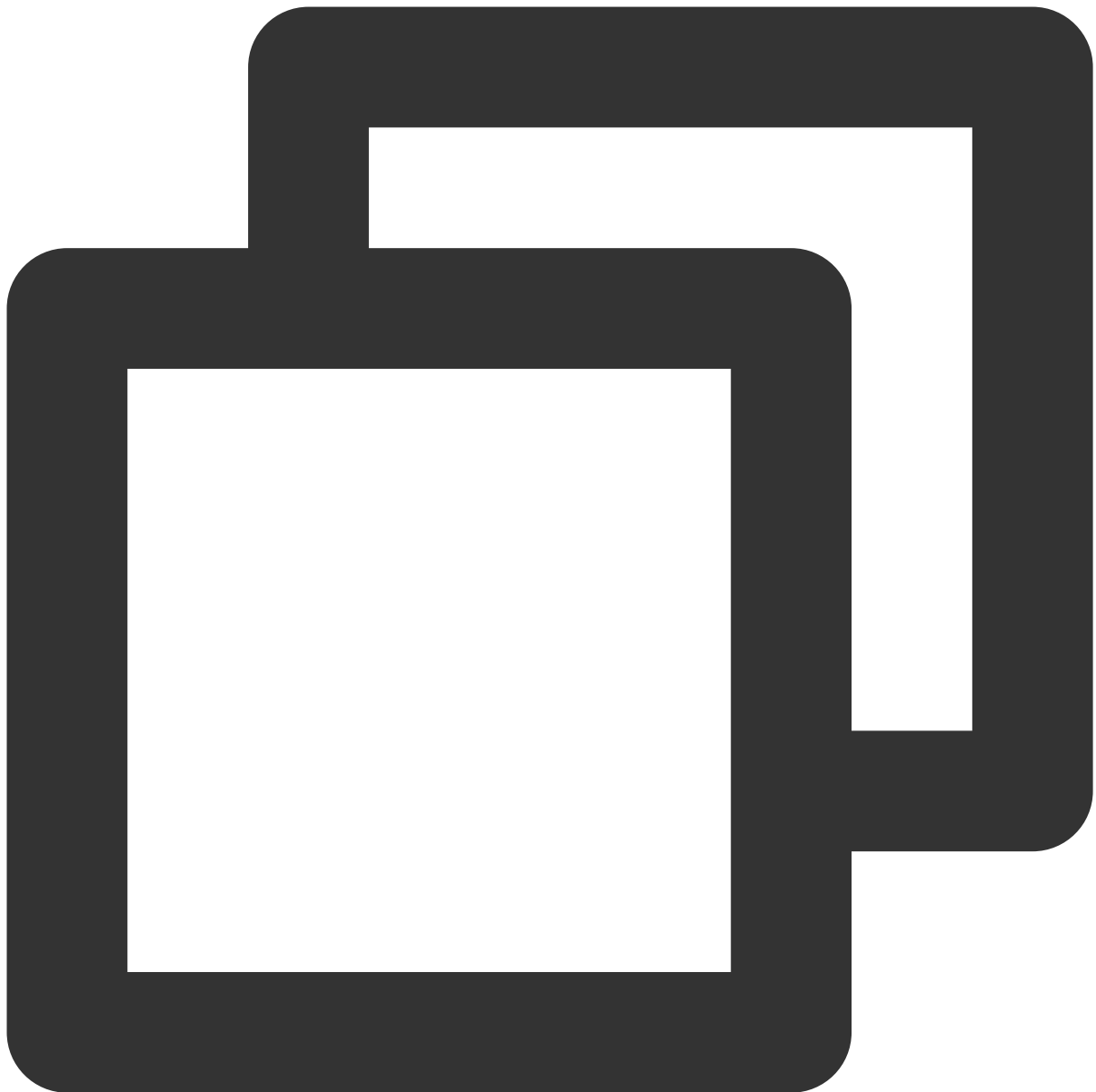


```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: coding-cd-role
rules:
- apiGroups: [""]
  resources: ["namespaces", "configmaps", "events", "replicationcontrollers", "serv
  verbs: ["get", "list"]
- apiGroups: [""]
  resources: ["pods", "pods/portforward", "services", "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

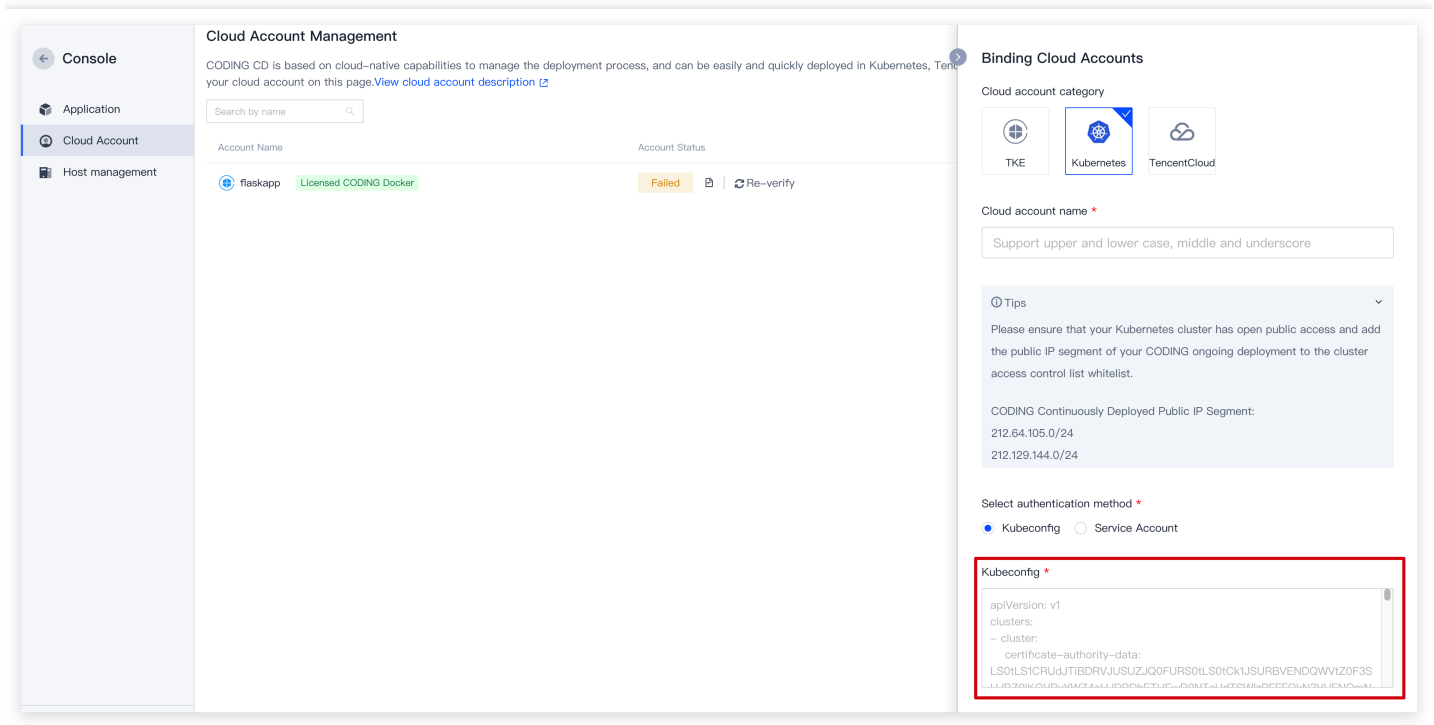
You must activate the CODING DevOps service for your Tencent Cloud account before you can use Coding Continuous Deployment (CD).

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:



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: LS0tLS1CRUdJTiBDRVJUSUZADQFURS0tLS0tCk1JSUM1ekNDQWMrZ
0F3SUJBZ0lCQVRBTKJna3Foa2lHOXcwQkFRc0ZBREFTVjNnd0VRWURWUVFERXdwdGFjXNkYTNWVpVTk
JNQjRFRFRjd0ESXhNREEwTURJMU1Wb1hEVE13TURJd09EQTBREkxTVZvd0ZURVRNbkVHQTFVRQpBeE1
LYlZdsdWFXdDFZbVZEUVRDQ0FTSXdEUVlks29aSWWh2Y05BUUVCQlFBRGdnRVBIBRENDQVFvQ2dnRUJBTUc1
CjYwU0txcHVXeE1mWlVabmVFakM5bjFseHFQSzdUTlVQbzROejFWcWxaQkt6NzJDVVErZjBtVGNQLy9oS
3BQUVAKAg9pNndyaXJRUmVERTErRFIROTZHVDIrrSGZ3L2VHQTI5ZmErNS80UG5PWlpTUEVpS3MxVVdhc0
VqSDJVZG4xTwpEejVRZk1ESkFjZlBoTzV0eUZFaGZNa2hid0Y2QkJONnh5RmJjdXl4OThmZGx5SWJUNp
LSml6VWZQcUx2WUZoCmFQbjF4WFZyT2QyMnFtblgzL2VxZXM4aG51SmpJdlVPbWRDRlhjQVRYdE00Wmw2
bERvWUs2VS9vaEFzM0x4VzAKWUV4ZkcXZmFXdjIrrR0t4WWV2Q0FuMitSQ3NBdFpTZk9zcVljMmorYS9FO
DVqdzcySlFkNGd6eG1HMCszaU14WApWaGhpcWFfY1owZlRCC0FtZHY4Q0F3RUFBU5DTUVBd0RnWURWUj
```

```
BQQVFIL0JBUURBZ0trTUIwR0ExVWRKUVFXck1CUUdDQ3NHQVFVRkJ3TUNCZ2dyQmdFRkJRY0RBVEFQQmd
OVkhSTUJBZjhFQ1RBREFRSC9NQTbHQ1NxR1NJYjMKRFFQkN3VUFBNELCQVFDKzFuU2w0dnJDTEV6eWg0
VWdXR3ZWSldtV2ltM2dBWFFJU1R2WG56NXZqOXE3Z0JYSwpCRVUyakVHTFF2UEJQUWZwUjhmZllCZCtqT
2xtYS9IdU9ISmw0RUxhaHJKbnIwaU9YcytoeVlpV0ZUKzZ2R05RCmY4QnAvNTlkYzY1ejVVMnlUQjd4Vk
hMcGYzRTRZdUN2NmZhdy9PZTNUUzZUbThZdFBXREgxNDBOR2ZKMH1WR1YKSzZsQnl5THMwMzZzT1V5ZUJ
pcEduOUxyKytvb09mTVZIU2dpaEJlcEl3ZVVvYk05YU1ram1Hb2VjNk5HTUN3NwpkaFNWTmdMNGxMSnRv
RktoVDdTZHFjMmk2SWlwbkJrdUlHUWRJUFliQnF6MkN5eVMyRkZmeEJsV2lmNmcxMTFTClphSUlpQ0lLb
XNqeDjvTFBhOUdNSjR6bERNR1hLY1ZyNnhhVQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
server: https://*****
name: demo
contexts:
- context:
cluster: demo
user: demo
name: demo
current-context: demo
preferences: {}
users:
- name: demo
user:
client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURBRENDQWVpZ0F3
SUJBZ0lCQWpBTkna3Foa2lhOXcwQkFRc0ZBREFTVJND0VRWURWUVFERXdwdGFXXAKYTNWaVpVTkJNQ
jRYRFRJd01ESXhNekV5TXpreU5sb1hEVEl4TURJeE16RX1Nemt5Tmxvd01URVhNQlVHQTFVRQpDaE1PYz
NsemRHVnRPbTFoYzNSbGNuTXhGakFVQmdOVKBjBTVREVzFwYm1scmRXSmxMWFZ6Wlhd2dnRwlnNQTBHCkN
TcUdTzIzRFFQkFRVUFBNELCRHdBd2dnRUtBb0lCQVFDdEp5MThYOGVBaVlJc1g3Z0xQazBOZEFuR1NJ
U0kKeXNGMGIzS21CTk9VclRlCgtnaUFwRlZmZDJaVXN3R3Iwd0VBL0FIWm9PMUxPUHdqYzEyb2o1bGIwW
lNNTWlMaQpVeW9UQ3huREZlVDJlQn1PNCtGrk5pVctXTU5FTURURWxERXh1ano1WVIwb1pZVjN2V2Z5T3
l2SlBna1dFU29lClVVCnVvQmRRbU5LUzhCeXhIUmfvcnFJUFVRGkzUFJlTlZGZERV1lNTWpnbDZmbmd
HWkRQMnpjTlNFZjRGNzYkC2FUK0VhMU1kbjV5akRCNXcZaHJvZXBbc1c0QVUyR3NTRFZyTHY2UDFiSWV0
RDdONjZrT1BBNk1lU1RkbnUNLlWp2aEJrbGVPMGFwcERzTERDT3hpMkc2Q1BRNDZVYVZUOEhZMk9sQU5nS
mtRRDNwYjFXTnlhQlVpRkFnTUJBQUdqClB6QTlNQTRHQTFVZER3RUIvd1FFQXdJRm9EQWRCZ05WSFNVRU
ZqQVVVCZ2dyQmdFRkJRY0RBVU1JS3dZQkJRUVgKQXdJd0RBWURWUjBUQVFIL0JBSXdbREF0QmdrcWhraUc
5dzBCQVZzRkFBT0NBUEVBTbVrZ1dxWnRtbkZITlcxYgpqL012ekcxTmJyUEtWYXVobml5RzRWwNRYZzR1
Uk5iSGhicEhEdThyamd2dVB1Y0xMdHAzblU2UGw4S2J2WFpiCmplNmJQR2xvV3VBcFIRvW9KRfQ2VEpDK
2o2Qm5CSXpWQkNOL21lSWVPQ0hEK1k5L2dtbzRnd2Q4c2F3U0Z1bjMKZTFVekF2cHBwdTVZy05wcU92aU
kxT2NjNGdxNTd2V1h1MFRlIdUJkM0VtQ2JZR XUzYXhOL25ldnhOYnYxbDFRSQovSzRaOWw3MXFqaEp3SVl
BaHUzek5pTWpCU1VTRjJkZnd2NmFnc1hSUnN6b1Z4eje5Mm9qm2pWU215cXZxeVFrCmZXckpsc3VhY1ND
dTlKUE44OUQrVXkwVnZXZmhPdmp4cXVRSktwUW9hMz1Qci81Q3YweXFKUkFIMkk5Wk1IZEYKkNkQJRVBRP
TOKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
client-key-data: LS0tLS1CRUdJTiBSU0EgUUFJJVkfURSBURVQSL0tLQpNSU1FcEFJQkFBS0NBUEVB
c1NjdGZGL0hnSW1DTEYrNEN6NU5EWFkKfVpRw1NckJkRz15cGdUVGxLMHg2WklJCmduLULzUm2RtVkhHQ
nE5TUJBUHdCMmFedFN6ajhJM05kcUkrWlc5R1VqR1BTNGxNcUV3c1p3eFIwOwh3Y2p1UGgKU1RZay9sak
RSREWEepReE1YbzgrV0Vks0dXRmQ3MW44anNyeVQ0SkZoRXCfMUZLN3FBWFVKalNrdkFjc1IwVwpxSzZ
pRDaxQTR0ejBSNXZSWFF5Vm1EREk0SmVuNTRCbVf6OXmzRFVosCtCZStyr2svaEd0VEhaK2Nvd3dlY040
CmE2SHFRSzf1QUZOahJFZzFheTcrajlXeUhyUst6ZxVwRgP3T21CMFV5Wmdpdc0UVPKWGp0R3FhUTdDd
3dq1kKdGh1Z2owT09sR2xVL0IwdGpwUURZQ1pFQTK2Vz1WamNtZ1ZJaFFJREFRQUJBb0lCQUdDazVGTn
VGaWtkRndYegphd01Ezy9or1V3ckZIZ3hIdHRCcFFBRi80avF5d3hBT0RTY1lFbdVPUTFSME90OFBoNwP
```

vRDVSTHFRWjZTT2owCmhFc0gwMTRYVFNWS3RqTFNua0pBeU9GRWNYL0hFdjJDSF1NRzVJRctSQWEwTFUr
bk13bmRvMwPccG91Y21uRXAKeTNHOUT3Ukkxc04xVXhNQWdhVk12NWFocGE2UzRTdENpalh3VGVVWUxpc
1pSZGp5UG1jUWlQN0xaSnhBcjRLTgpTUH1DNE1IZTJtV3F3cjm5cnBrMWZ3WkViMTRPMjR2Z3dMYmROTf
JYdVhZSTdicEpOUGRJbEQvRExOQkJSLOFVCjhJYjNdATZwZ2M4dFA4VzJCeW9TQUJVZUNpWDRFM21wQUt
tVytKbzFuU3FwQ1FnM2JGV0RpRjFrKzdEZjJZM3IKc09UT0srVUNnWUVBMTBEb3BtRVcrNnowanZadVFZ
dFlPW1VQQzZwV0dBaDFLW1VHZndYVWVLQ3dnN1NuQW9qRwpuMjR4bWJVdTLzRzBjd2syK0VVcXh3S2IrR
2NJVTdyNVdOaUNXNVZYQzV5ZUp3OFZiMwtQekRzMSs4YzA4VjNhCkzNHluOHpjZm1WTkZOUM1ZS1FMK1
FGbnZ3ZXM4NFRleGVbb1hGY2FQcVZTNDNVV2JHRW02ZThDZ1lFQXplNFkKeUxYQ2pWNVppajFsUTdkQ2F
weEQvL0dCT2JsemRocXRuSjl1OWxyVkQ4Y3RvUEVKYVkwVFFWc3FaNVhCdTntVApLb0w4bmZjWHg4cWR3
Z3gvZFRHa2c3d0rZFkrUTFuVDNOWnRFSVdVUkR3T0hLT1N1Tm5kdnU1a1Q4aXRrdHhhCktrYV1SM1NOT
25iMlFzb1ZHa3F5OS9QK0xWLOFyeWdScmtaYXVnc0NnWUVBaGpUTkdUYzltaxNXeTV2d0FHTzkKM1NFSG
9YRlJmbWgvakM2RFaxMUDmUE9iT21qRlREbzFCS0F5d3JBSm1RWUusyUkpzdUh4L2dGY3JJY1F6bCtqaQp
vRGRWaDM1a0tEUTlFd00valE0Tl1IdW1XOVhITjVvWmNMbTFISmNnL3Bsd1pzVkxFNFFVaHVzT1lUZUs2
TVgyCkU0OS8rchJBSFVEOG5oNlpuWGN4U1BjQ2dZRUFub0lQcDZab1MwSjliMi9VbTJ2YS9vNnJ0TDBpO
Tlpc2JCTWEKNFR6RFAzTXBIc3owYlRZN1JYaW1ncDcybytiY3lUNUtMZVhISnB3RVBPL1R3SUs0Tk8veU
xzZzN3TExOR0RCegphRC9Ra1hBUWNQazg3NFJrc2s1WVpkZS9kTDRHQk00QnhScXpxZmhXME5LeXVUUXR
UQ0NGWTEvMm5OeGdSekp6CmNZNkwxRU1DZ1lCdXpHRkJVeXM4TFpHekFNTWdmN1VRQ0dVZERrT2dJRHdz
d0dxVVlxQ2ZLcnlGYVdCOUJvSi8KMnJMVmVYNDVXTnFpa0tCmlgvckdRbGFIK25YalRBaDlpN0nrWGRyS
UQzeXA1cGJBa0VnYjg3dGo2Y3hONlBOcQo5cnhzOU1lR0NleFhsdjBkUUpMUkownXU2OEVxYm44QORIOX
FRSWZaTml0NXA0S0JFVkp3L1E9PQotLS0tLUVORCBSU0EgUFJJVkfURSBLRVktLS0tLQo=