

持续部署 常见问题 产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

常见问题

一般性问题

部署 Kubernetes 资源时如何拉取私有库镜像

Kubernetes 云账号的最小权限要求

如何将 Kubeconfig 中的证书文件转为证书数据

常见问题

一般性问题

最近更新时间：2022-02-23 15:32:42

持续部署支持哪些制品类型？

持续部署支持 Docker 镜像、Generic 文件、War 包。

持续部署支持哪些集群类型？

持续部署支持 CVM（Linux 操作系统）、TKE、SCF。

如何在配置持续部署过程的时候保护敏感信息？

对于 Token、SSH 私钥、Kubernetes 证书等保密信息，可以在 CODING DevOps 网页端的任一项目选择 [项目设置 > 开发者选项 > 凭证管理](#)，进入“凭证管理”页面中设置，并可以在此设定哪些持续部署发布流程可用。

如何发布源代码？

很多常见的动态语言是没有编译和构建过程的，可以直接在应用的制品设置中配置 Git 代码仓库指明文件路径后发布。

如何配置审批流程？

审批流程的配置分两步：

1. 进入 CODING DevOps 网页端，选择任一项目，在项目中选择 [设置 > 属性与流程 > 审批流程设置](#)，进入“审批流程设置”页面，可以给本项目添加若干个审批流程。
2. 在 [持续部署 > 发布流程 > 关联审批](#) 中可以设置发布流程与审批流程的关系。设置完毕后，在提发布单进行发布的时候，系统会自动判别出来具体要预先执行哪一个审批流程。

发布单可以不审批执行么？

可以，在发布流程的关联审批页面，不进行任何设置，则此发布流程就不需要审批，对应的发布单将在提单后直接开始执行。

部署 Kubernetes 资源时如何拉取私有库镜像

最近更新时间：2024-01-03 12:00:25

本文为您详细介绍在 CODING 里部署 Kubernetes 资源时如何拉取私有库镜像。

前提条件

使用 CODING 持续部署的前提是，您的腾讯云账号需要开通 CODING DevOps 服务。

进入项目

1. 登录 CODING 控制台，单击团队域名进入 CODING 使用页面。
2. 单击工作台首页左侧的



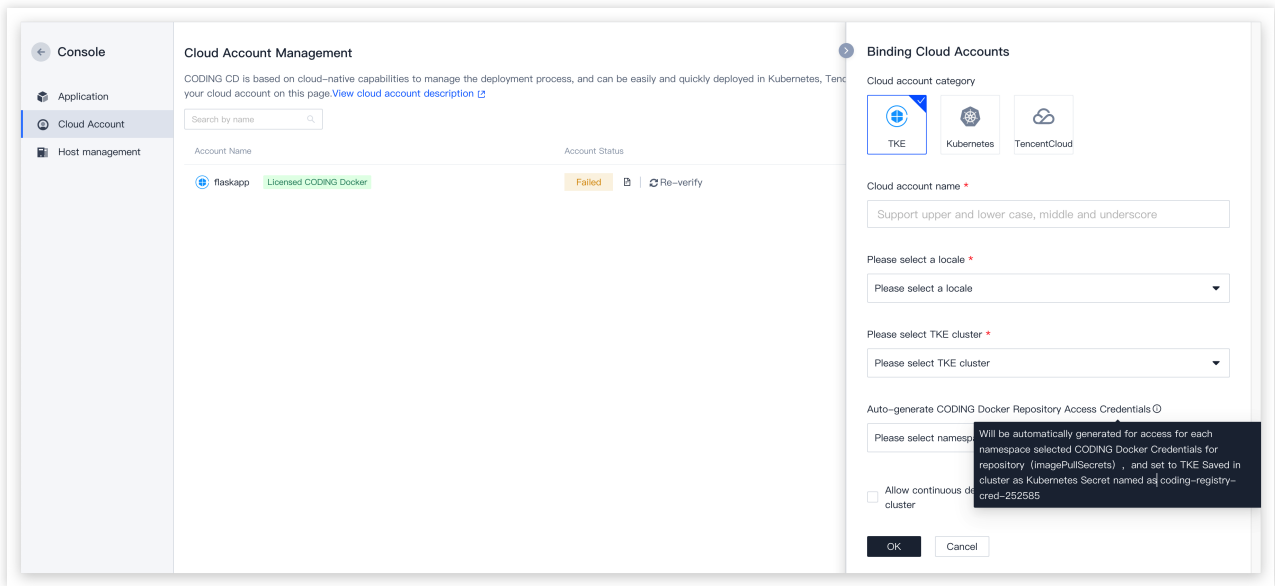
，进入持续部署控制台。

功能介绍

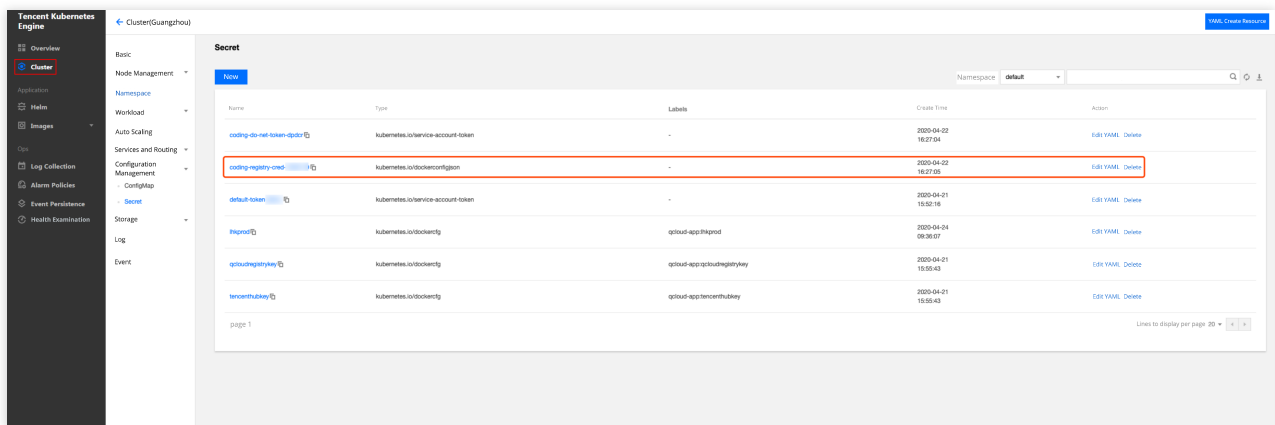
在部署 Kubernetes 资源时，如果 manifest 引用的镜像是存储于私有库的镜像，则需要在 manifest 中配置 `imagePullSecrets` 才能成功拉取镜像。

以下将会分云账号类型来讨论 `imagePullSecrets` 配置场景：

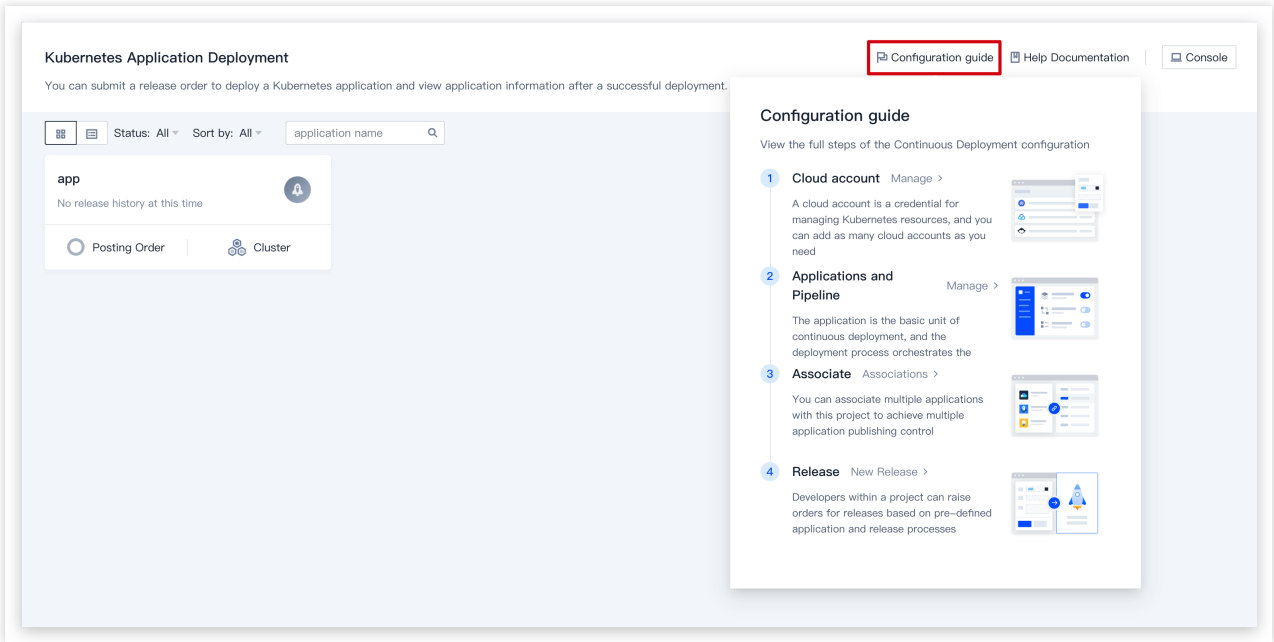
腾讯云 TKE



如上图所示，CODING CD 会在 TKE 集群中生成名为 `coding-registry-cred-$(user_id)` 的 Secret，此 Secret 信息可以在 TKE 控制台查看：

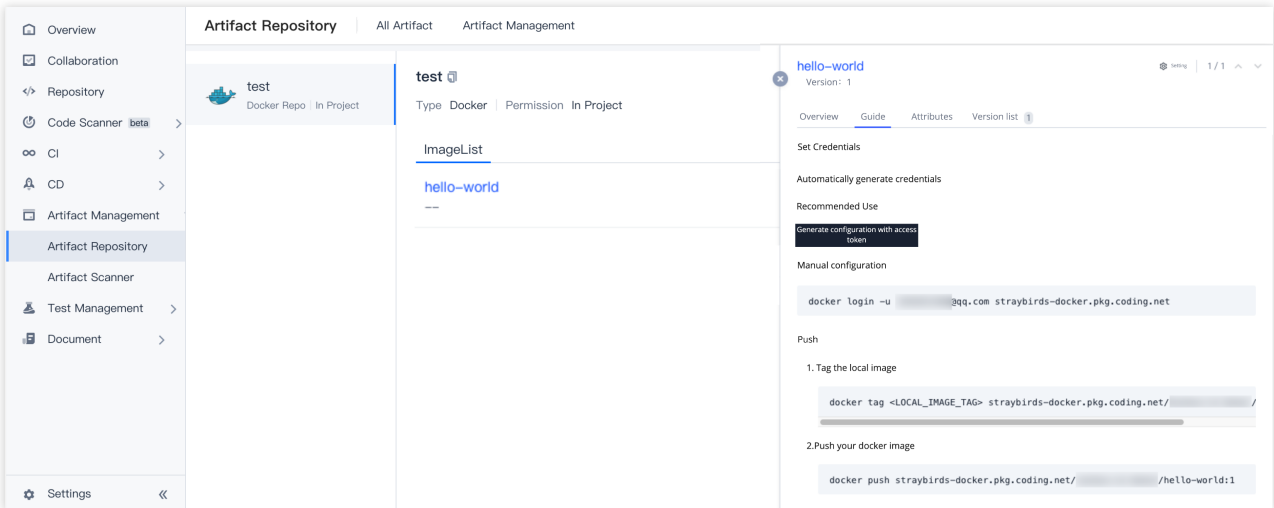


云账号添加成功后，可以查看示例用法：



Kubernetes 云账号 (非 TKE 集群)

如果通过 Kubeconfig 或 Service Account 凭据添加的 Kubernetes 云账号，在 manifest 引用私有库的镜像时，需要先在 Kubernetes 集群中创建 Secret，以 CODING 制品库为例：



在 manifest 中直接引用此私有库的镜像会失败，需要在集群中生成 Secret：



```
kubectl create secret docker-registry coding-regcred \\
--docker-server=您的团队域名 --docker-username=您的邮箱 --docker-password=$(passwd)
```

创建 Secret 后，在 manifest 中通过 `imagePullSecrets` 配置此 Secret 用于拉取镜像（最后两行）：

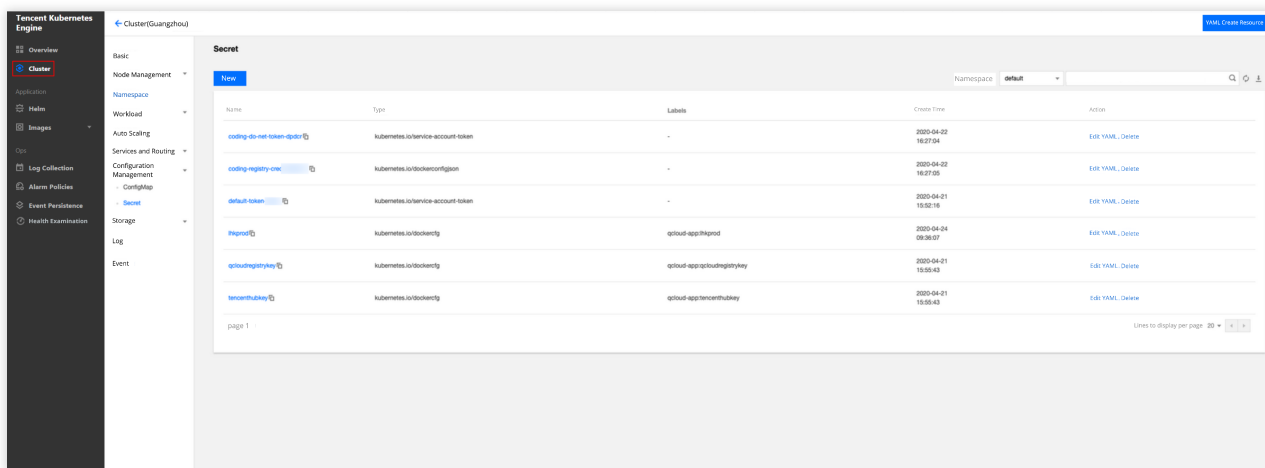


```
apiVersion: apps/v1
kind: Deployment
...
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
  imagePullSecrets:
  - name: coding-regcred
```

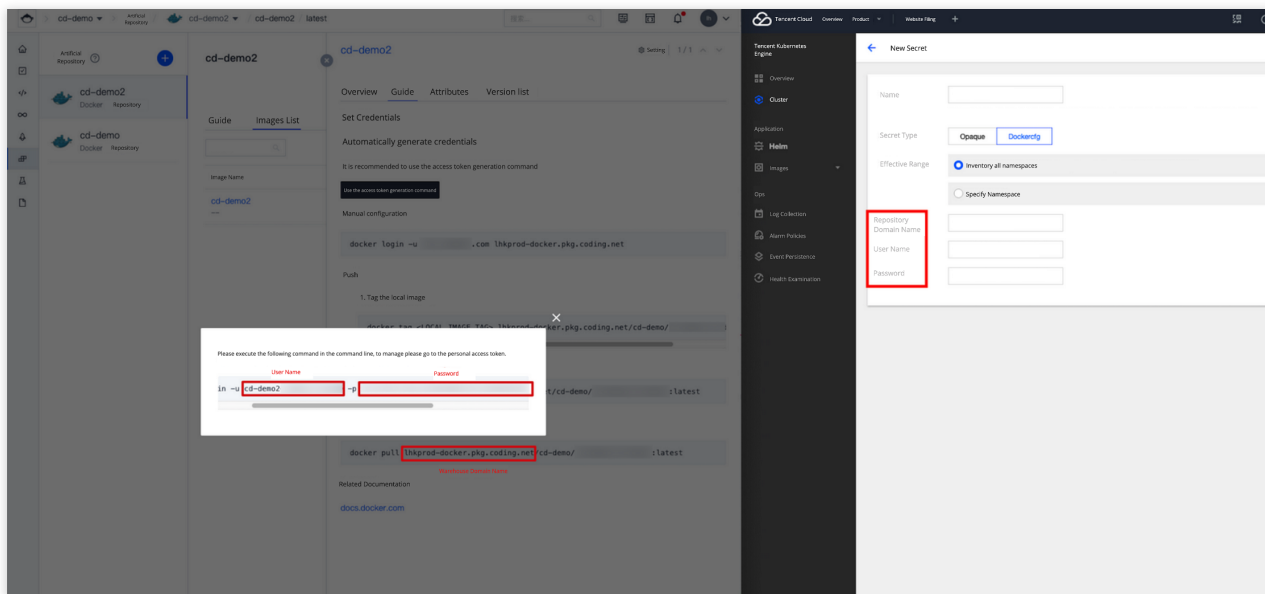
阅读更多：[Kubernetes 如何从私有仓库拉取镜像](#)

Kubernetes 云账号（TKE 集群）

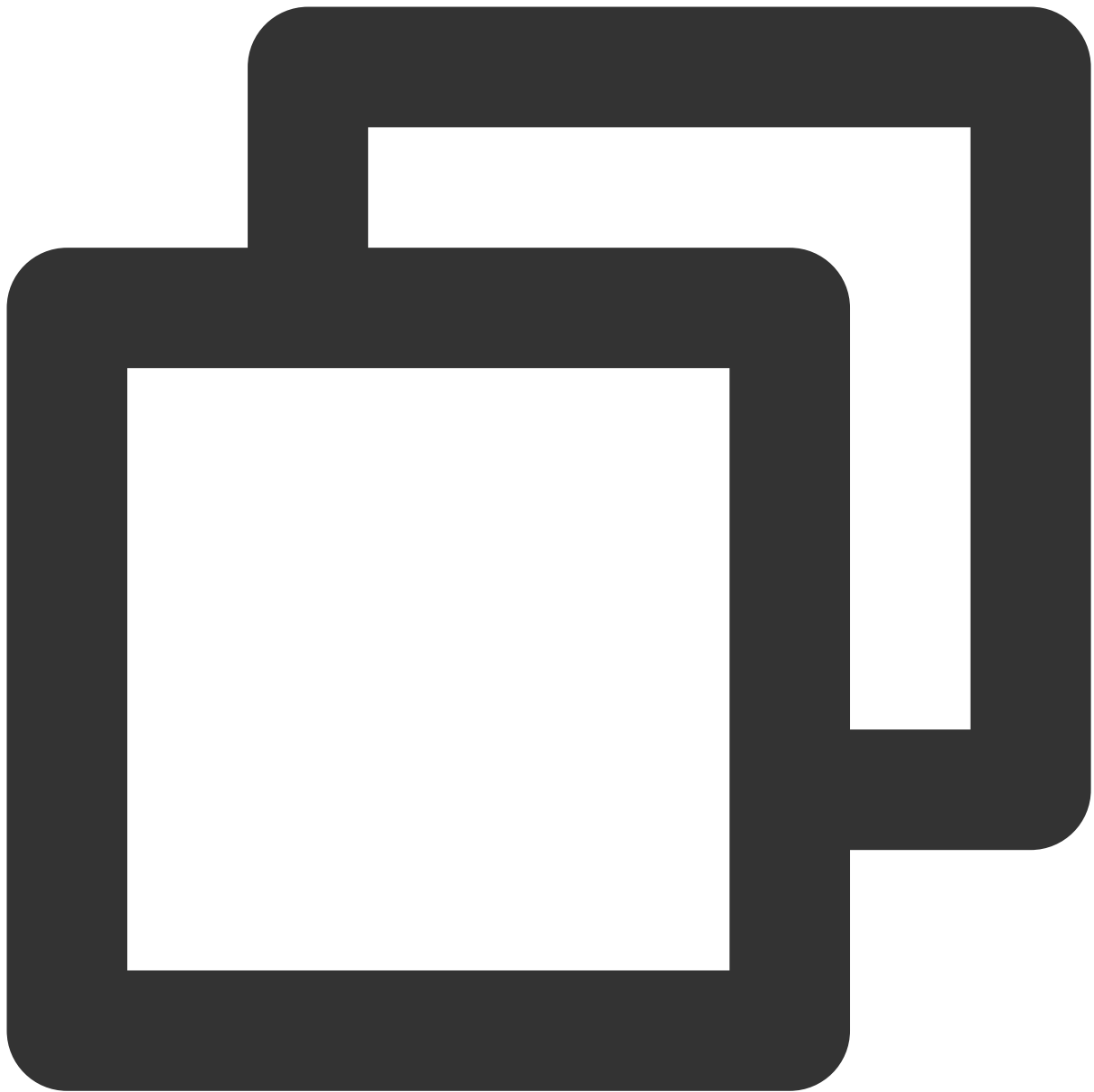
如果您选择通过 Kubeconfig 或 Service Account 添加 TKE 集群云账号，可以直接在 TKE 控制台创建 Secret。进入集群信息页，选择**配置管理 > Secret > 新建**：



需要填写的表单信息对应关系如下：



同样地，创建 Secret 后，在 manifest 中通过 `imagePullSecrets` 配置此 Secret 用于拉取镜像（最后两行）：



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


Kubernetes 云账号的最小权限要求

最近更新时间：2024-01-03 12:00:52

本文为您详细介绍 CODING 里 Kubernetes 云账号的最小权限要求。

前提条件

使用 CODING 持续部署的前提是，您的腾讯云账号需要开通 CODING DevOps 服务。

进入项目

1. 登录 CODING 控制台，单击团队域名进入 CODING 使用页面。
2. 单击工作台首页左侧的



，进入持续部署控制台。

功能说明

若希望在 Kubernetes 场景下（K8s 云账号）完成应用发布，CODING 持续部署需要调用相关的 Kubernetes APIs。CODING 团队不推荐您将 Kubernetes 集群所有权限皆授予 CODING CD；得益于 Kubernetes 的 RBAC（Role Based Access Control）机制，您可以给 CODING CD 配置应用发布所需的最小权限。下文是配置最小权限的指引。

Role

CODING 推荐您在需要开放权限的命名空间中创建 `Role`，并将 `ServiceAccount` 和 `Role` 进行绑定。

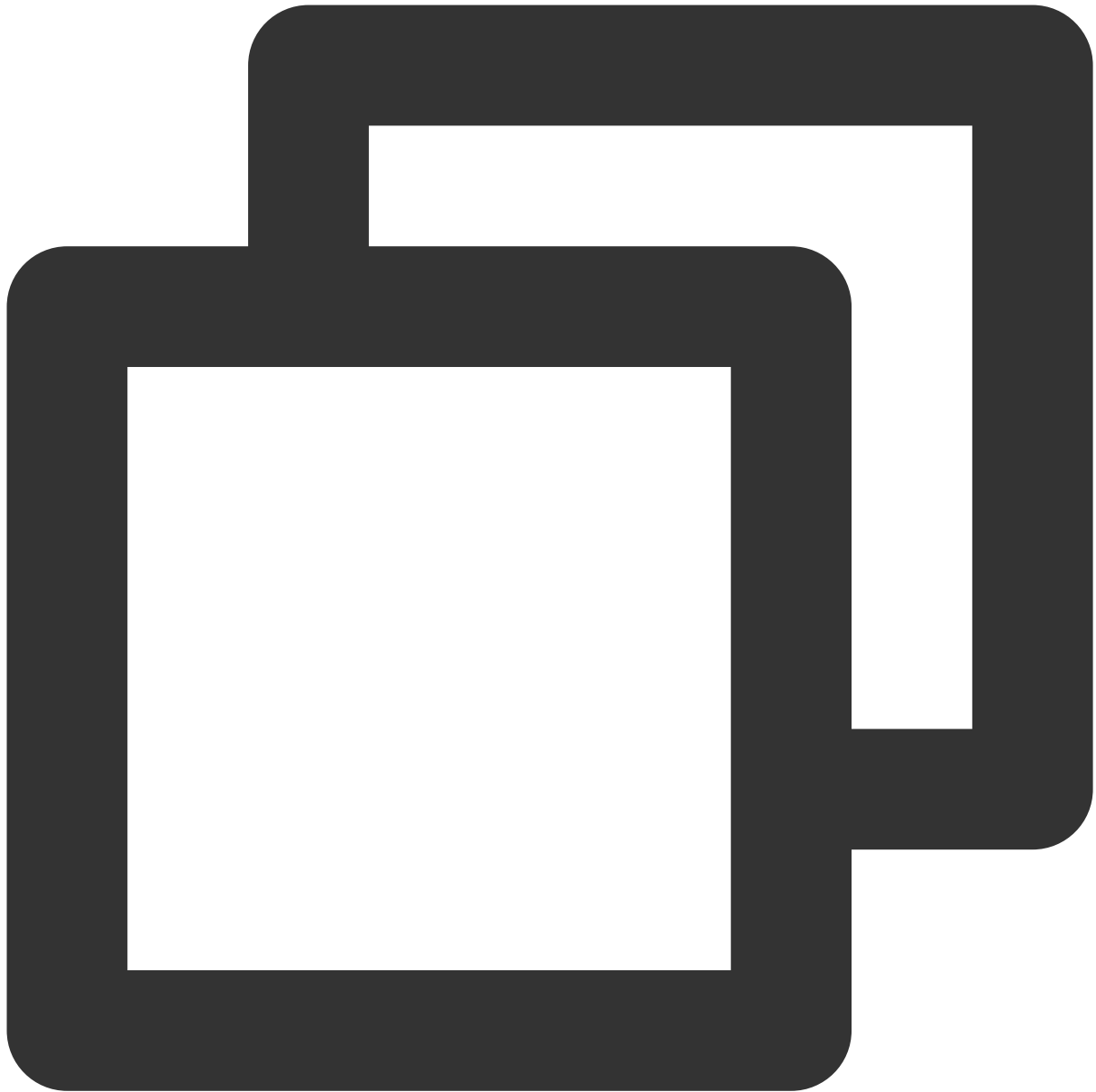


```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: coding-cd-role
rules:
- apiGroups: [""]
  resources: ["namespaces", "configmaps", "events", "replicationcontrollers", "serv
  verbs: ["get", "list"]
- apiGroups: [""]
  resources: ["pods", "pods/portforward", "services",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

下一步是为 CODING CD 创建一个 `Service Account`，持续部署控制台将会用此 `Service Account` 与 Kubernetes 集群交互。您可以使用如下的 `manifest` 创建 `Service Account`。



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

Role Binding

最后，创建一个 `RoleBinding` 将上述的 `coding-cd-role` 与 `coding-cd-service-account` 进行绑定。



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

```
kind: ServiceAccount  
name: coding-cd-service-account
```

如何将 Kubeconfig 中的证书文件转为证书数据

最近更新时间：2022-03-30 10:23:07

本文为您详细介绍如何将 Kubeconfig 中的证书文件转为证书数据。

前提条件

使用 CODING 持续部署的前提是，您的腾讯云账号需要开通 CODING DevOps 服务。

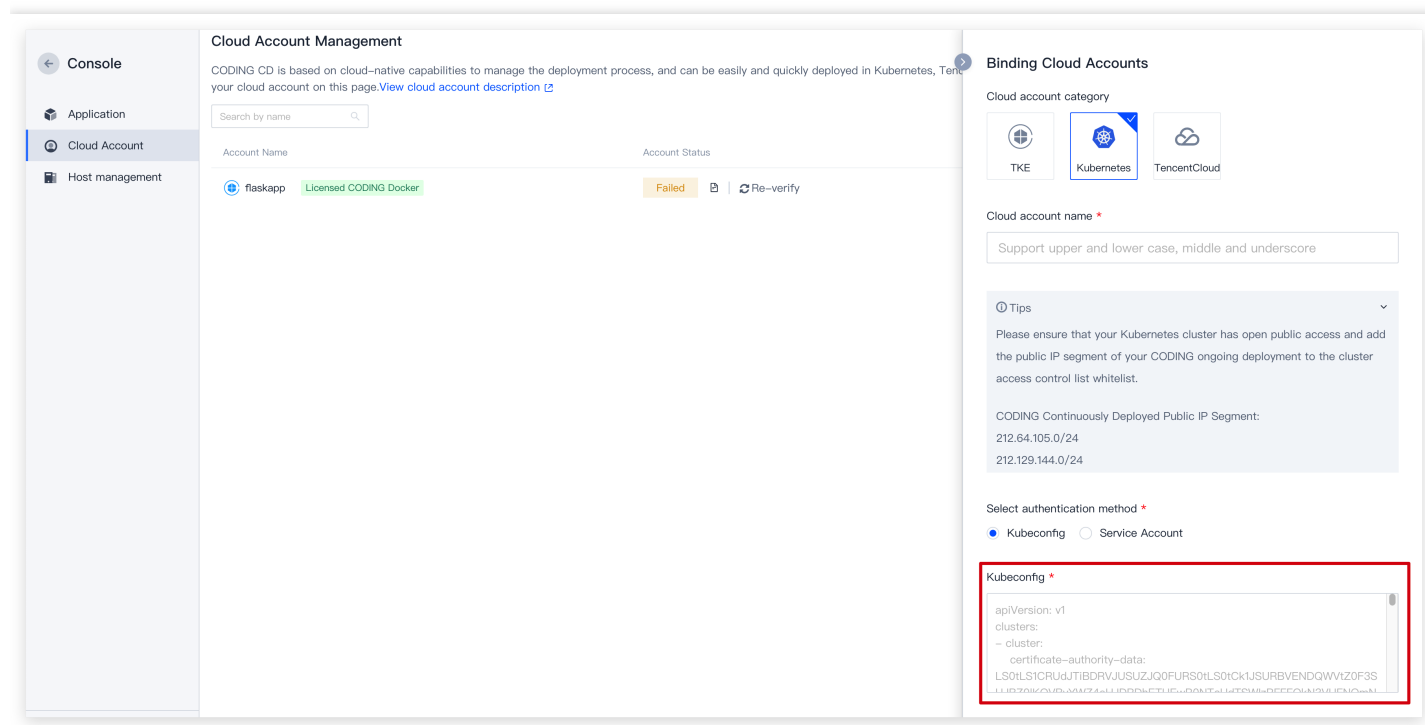
进入项目

1. 登录 CODING 控制台，单击团队域名进入 CODING 使用页面。

2. 单击工作台首页左侧的 ，进入持续部署控制台。

功能介绍

在添加 Kubernetes 云账号时，可以选择 Kubeconfig 或 Service Account 两种认证方式：



如果拿到了 Kubeconfig 指定证书文件认证（以如下的 Kubeconfig 为例：`certificate-authority`、`client-certificate` 和 `client-key` 都指的是某个具体文件，并且文件中包含了证书信息）。那么在添加云账号之前，需要把证书文件转换为 `base64` 编码过后的字符串。

```
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
```

下面将会讲解如何进行转换操作。

获取证书文件的 `base64` 编码

执行命令：

OS X

```
cat "证书文件" | base64
```

Linux

```
cat "证书文件" | base64
```

Windows

```
certutil -f -encode "证书文件" "输出文件"
```

更新 Kubeconfig 文件

1. 将 `certificate-authority` 改为 `certificate-authority-data` , 并且将 `*/ca.crt` 证书文件经 `base64` 编码后的字符串填入该位置。
2. 将 `client-certificate` 改为 `client-certificate-data` , 并且将 `*/client.crt` 证书文件经 `base64` 编码后的字符串填入该位置。
3. 将 `client-key` 改为 `client-key-data` , 并且将 `*/client.key` 证书文件 `base64` 编码后的字符串填入该位置。

最终得到的内容如下：

```
apiVersion: v1
kind: Config
clusters:
- cluster:
certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZADQFURS0tLS0tCk1JSUM1ekNDQWMrZ
0F3SUJBZ0lCQVRBTKJna3Foa2lHOXcwQkFrc0ZBREFWTVJNd0VRWURWUWFERXdwdGFjXNxAkYTNWVpVTk
JNQjRYRFRJd01ESXhNREEwTURJMU1Wb1hEVE13TURJd09EQTBNREkxTVZvd0ZURVRNQkVHQTFVRQpBeE1
LYlksdWFXdDFZbVZEUVRDQ0FTSXdEUVlKS29aSWWh2Y05BUUVCQlFBRGdnRVBIBRENDQVFvQ2dnRUJBTUc1
CjYwU0txcHVXeE1mWlVabmVFakM5bjFseHFQSzdUTlVQbzROejFWcWxaQkt6NzJDVVErZjBtVGNQLy9oS
3BQUVAKaG9pNndyaXRJUmVERTErRFIROTZHVDIrsGZ3L2VHQTI5ZmErNS80UG5PWlpTUEVpS3MxVVdhc0
VqSDJVZG4xTwpEejVRZk1ESkFjZlBoTzV0eUZFaGZNa2hid0Y2QkJonnh5RmJJdXl40ThmZGx5SWJJUnp
LSml6VWZQcUx2WUZoCmFQbjF4WFZyT2QyMnFtblgzL2VxZXM4aG51SmpJdlVPbWRDRlhjQVRYdE00Wmw2
bERvWUs2VS9vaEFzM0x4VzAKWUV4ZkcxMzFXdjlRrR0t4WWV2Q0FuMitSQ3NBdFpTZk9zcVljMmorYS9FO
DVqdzcySlFkNGd6eGlHMCszaU14WApWagHpcWFrY1owZlRCC0FtZHY4Q0F3RUFBYU5DTUVBd0RnWURWUj
BQOVFIL0JBUURBZ0trTUlwR0ExVWRKUVFXck1CUUdDQ3NHQVFVRk1J3TUNC22dyQmdFRk1JRY0RBVEFQQmd
OVkhSTUJBZjhFQlRBREFRSC9NQTBHQ1Nxr1NjYjMkRFRFFQkn3VUFBNElCQVFDKzFuU2w0dnJDTEV6eWg0
VWdXR3ZWSldtV2ltM2dBWFFJU1R2WG56NXZqOXE3Z0JYSwpCRVUyakVHTFF2UEJQUWZwUjhmZllCZCtqT
2xtYS9IdU9ISmw0RUxhaHJKbnIwaU9YcytoeVlpV0ZUKzZ2R05RCmY4QnAvNTlkYzY1ejVVMnlUQjd4Vk
hMcGYzRTRZdUN2NmZhdy9PZTNUUzZUbtThZdFBXREgXNDBOR2ZKMh1WRlYKSzZsQnl5THMwMzZzT1V5ZUJ
pcEduOUxyKytvb09mTVZIU2dpaEJlcE13ZVVvYk05YU1ram1Hb2VjNk5HTUN3NwpkaFNWtmdMNGxMSnRv
RktoVDdTZHFjMmk2SWlwbk1JrdU1HUWRJUFliQnF6MkN5eVMYRkZmeEJsV2lmNmcxMTFTClphSUlpQ0lLb
XNqeD1VTFBhOUdNSjR6bERNR1hLY1ZyNnhhVQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
server: https://*****
name: demo
contexts:
- context:
cluster: demo
user: demo
name: demo
current-context: demo
preferences: {}
users:
- name: demo
user:
client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURBRENDQWVpZ0F3
SUJBZ0lCQWpBTk1na3Foa2lHOXcwQkFrc0ZBREFWTVJNd0VRWURWUWFERXdwdGFjXNxAkYTNWVpVTk1NQj

```

```
jRYRFRJd01ESXhNekV5TXpreU5sb1hEVEl4TURJeE16RXlNemt5Tmxvd01URVhNQLVHQTFVRQpDaE1PYz
NsemRHVnRPbTFoYzNSBgNuTXhGakFVQmdOVkJBTvREVzFwYm1scmRXSmxMWFZ6WlhJd2dnRWlNQTBHCkN
TcUdTswiZrFFFQkFRVUFBNelCRHdBd2dnRUtBb0lCQVFDdEp5MThYOGVBaVlJc1g3Z0xQazBOZEFuRlNJ
U0kKeXNGMGIzS2lCTk9VclRlCgtNaUFRlZmZJaVXNZR3Iwd0VBL0FIWm9PMUxPUhdqYzEyb2o1bGIwW
lNNTWlMaQpVeW9UQ3huREZIVDJIQnlPNcTGrk5pVcTXTU5FTURURWxERXh1ano1WVIwb1pZVjN2V2Z5T3
l2S1Bna1dFU29IClVvcnVvQmRRbU5LUzhCeXhIUmfvcnFJUFRVRGkzUFJlTlGZERKV1lNTWpnbDZmbmd
HWkRQMnpjTlNFZjRGNzYkC2FUK0VhMU1kbjV5akRCNXcZaHJvZXBbc1c0QVUyR3NTRFZyTHY2UDFiSWV0
RDdONjZrT1BBNklIUlRkbnUNLLWp2aEJrbGVPMGFwcERzTERDT3hpMkc2Q1BRNDZVYVZUOEhZMk9sQU5nS
mtRRDNwYjFXTnlhQlVpRkFnTUJBQUdqClB6QTlNQTRHQTFVZER3RUIvd1FFQXdJRm9EQWRCZ05WSFNVRU
ZqQVVCZ2dyQmdFRkYR0RBUVlJS3dZQkJRvUgKQXdJd0RBWURWUjBUQVFIL0JBSXdBREFOQmdrcWhraUc
5dzBCQVZzRkFBT0NBuUVBTHVrZ1dxWnRtbkZITlCXYgpqL012ekcxTmJyUeTwyXVobml5RzRWwNRYZr1
Uk5iSGhicEhEdThyame2dVB1Y0xMdHAzblU2UGw4S2J2WFpiCmplNmJQR2xvV3VBcFIrVW9KRFQ2VEpDK
2o2Qm5CSXpWQkNOL21lSWVPQ0hEK1k5L2dtbzRnd2Q4c2F3U0Z1bjMKZTFVekF2cHBwdTVZY05wcU92aU
kxT2NjNgdxNTd2V1h1MFRIdUJkM0VtQ2JZRUXzYXhOL25ldnhOYnYxbDFRSQovSzRaOWw3MXFqaEp3SVl
BaHUzek5pTWpCU1VTRjJkZnd2NmFnc1hSunN6b1Z4eJE5Mm9qM2pWU215cXZxeVFrCmZXckpsc3VhY1ND
dTlKUE44OUQrVXkwVnZXZmhPdmp4cXVRSktuW9hMz1Qci81Q3YweXFKUkFIMkk5Wk1IZEYKNkjqRVBRP
T0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
client-key-data: LS0tLS1CRUdJTiBSU0EgUUFJJVkfURSBURVQSL0tLQpNSU1FcEFJQkFBS0NBuUVB
c1NjdGZGL0hnSW1DTEYrNEN6NU5EWFfKefVpRWlNckJkRz15cGdUVGxLMHg2WklJCmDLULZuM2RtVkxHQ
nE5TUJBuHdCMmFedFN6ajhJM05kcUkrWlc5R1VqR1BTNGxNcUV3c1p3eFIwOwh3Y2p1UGgKU1RZay9sak
RSREeweEpReE1YbzgrV0Vks0dXRmQ3MW44anNyeVQ0SkZoRXCfMUZLN3FBWFVKalNrdkFjc1IwVwpXszZ
pRDaxQTR0ejBSNXZSWFF5Vm1EREk0SmVuNTRCbVF6OXmZRFVoSctCZStyR2svaEd0VEhaK2Nvd3dlY040
CmE2SHFRSzf1QUZOahJFzZfheTcrajlXeUhyUst6ZXVwRgP3T2lCMFV5Wmdpdc0UvPKWGP0R3FhUTdDd
3dq1kKdGh1Z2owT09sr2xVL0IwdGpwUURZQ1pFQTK2Vz1WamNtZ1ZJaFFJREFRQUJBb0lCQUdDazVGTn
VGaWtkRndYegphd01Ezy9oR1V3ckZIZ3hIdHRCcFFBRi80avF5d3hbT0RTY1lFbDVPuTFsME90OFBoNWp
vRDVSTHFRWjZTT2owCmhFc0gwmTRYVFNWS3RqTFNua0pBeU9GRWNYL0hFdjJDSFLNRzVJRctSQWEwTFur
bk13bmRvMwPcCg91Y21uRXAKeTNHOUT3Ukkxc04xVXhNQWdhVkl2NWFocGE2UzRTdENpalh3VGVVWUxpc
1pSZGp5UGljUWlQN0xaSnhBcjRLTgpTUH1DNE1IZTJtV3F3cjm5cnBrMWZ3WkViMTRPMjR2Z3dMYmROTf
JYdVhZSTdicEpouGRJbEQvRExOQkJSLOFVCjhJYjNdAtZwZ2M4dFA4VzJcEw9TQUJVZUNpWDRFM21wQUt
tVytKbzFuU3FwQ1FnM2JGV0RpRjFrKzdeZjJZM3IKc09UT0srVUNnWUVBMTBEB3BtRVcrNnowanZadVFZ
dFlPWlVQqzZwV0dBaDFLWlVHZndYVWVlQ3dnN1NuQW9qRwpuMjR4bWJVdTlZrZBjd2syK0VvCkXh3S2Irr
2NJVtdyNVdOaUNXNVZYQzV5ZUp3OFZiMwtQekRzMSs4YzA4VjNhCkKzNHluOHpjZm1WTkZOum1ZS1FMK1
FGbnZ3ZXM4NFRleGVbb1hGY2FQcVZTNDNVV2JHRW02ZThDZ1lFQXplNFkKeUxYQ2pWNvppajfSUTdkQ2F
weEQvL0dCT2JsemRocXRuSjl1OWxyVkQ4Y3RvUEVKYVkwVFFWc3FaNVhCdTntVApLb0w4bmZjWHg4cWR3
Z3gvZFRHa2c3d00rZfkrUTFuVDNOWNRFSVdVUkr3T0hLT1N1Tm5kdnU1a1Q4aXRrdHhhCktrYVlSmlNOT
25im1Fzb1ZHa3F5OS9QK0xWL0FyeWdScmtaYXVnc0NnWUVBaGpUTkdUYz1taXNxeTV2d0FHTzkKM1NFSG
9YRlJmbWgvakM2RFAxMUMdMUE9iT21qR1REbzfCFS0F5d3JBSm1RWUesyUkpzdUh4L2dGY3JJY1F6bCtqaQp
vRGRWADM1a0tEUT1Fd00valE0T1lIdW1XOVhitjVvWmNMbTFISmNnL3Bsd1pzVkfNFFVaHvZT1lUZUs2
TVgyCkU0OS8rcHJBSFVEOG5oN1puWGN4U1BjQ2dZRUFub0lQcDZab1MwSjliMi9VbTJ2YS9vNnJ0TDBpO
Tlpc2JCTWEKNFR6RFAzTXBic3owY1RZN1JYaW1ncDcybyt1Y3lUNUtMZVhISnB3RVBPL1R3SUs0Tk8veU
xzZzN3TExor0RCegphRC9Ra1hBUWNQazg3NFJrc2s1WVpkZS9kTDRHQk00QnhScXpxZmhXME5LeXVUUXR
UQ0NGWTEvMm5OeGdSekp6CmNZNkwxRU1DZ1lCdXpHRkJVeXM4TFpHekFNTWdmN1VRQ0dVZERrT2dJRhdz
d0dxVVlxQ2ZLcnlGYVdCOUJvSi8KMnJMVMYNDVXTnFpa0tCMLgvckdRbGFik25Ya1RBaDlpN0nrWGRyS
UQzeXA1cGJBa0VnYjg3dGo2Y3hONlBOcQo5cnhzOU1lR0NleFhsdjBkUUpMUkownXU2OEvxYm44QORIOX
FRSWZaTml0NXA0S0JFVkp3L1E9PQotLS0tLUVORCBSU0EgUUFJJVkfURSBURVktLS0tLQo=
```