

持续部署 常见问题 产品文档





【版权声明】

©2013-2023 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许可,任何主体不得以任何 形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责任。

【商标声明】

🔗 腾讯云

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。未经腾讯云及 有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权利人商标权的侵犯,腾 讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 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默示 的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100。



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 云账号相关问题 常见错误码



常见问题

一般性问题

最近更新时间: 2021-08-26 10:37:33

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:

- 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。
- 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: 2023-06-15 15:57:00

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,详情请参见 开通服务。

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。

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,则需要创建 K8s Secret,并在 manifest 中配置 imagePullSecrets 才能成功拉取镜像。

检查 K8s Secret

首先检查 K8s 中是否存在以 coding 开头的 Secret,命令:

kubectl get secrets



或通过云厂商的网页查看:

於 時 讯 云	产品 🖌 网站备案	+		
容器服务	← 集群(广州) / cls-m	nmocjsvn(test)		
器 概 览	基本信息	Se	ecret	
◎ 集群	节点管理	-	新建	
应用中心	命名空间			
🔅 Helm应用	工作负载	-	名称	类型
◎ 镜像仓库 🚽 🔻	自动伸缩		coding-do-net-token-dpdcr	kubernetes.io/service-account-token
运维中心	服务与路由	*		
□ 日志采集	配置管理	-	coding-registry-cred-7906719	kubernetes.io/dockerconfigjson
	ConfigMap			
◇ 事件持久化	 Secret 		default-token-j86wv 🗖	kubernetes.io/service-account-token
⑦ 健康检查	存储	•	lblogad (=	kubarantan in (denkarafa
	日志		Inkprodi	kubernetes.io/dockercig
	事件		qcloudregistrykey 🗗	kubernetes.io/dockercfg
			tencenthubkey	kubernetes.io/dockercfg
			第1页	
		_		

如果使用腾讯云账号登录 CODING,在设置 CODING CD 时,会在 TKE 集群中自动创建名为 coding-registry-cred-\$(user_id)的 Secret:

← 部署控制台	云账号管理	0	绑定云账号
	CODING CD 基于云原生的能力管理部署过程,可以方便快速部署于 Kubernetes 、腾讯云弹性伸缩、腾讯云 TKE 三种环境。在此页面可以管理您的云账号	. 1	云账号类别
应用	按名称提索 Q		
③ 云账号			
■ 主机管理	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■		期代工作性 Kubernetes 期代工 云紙号名称・ Test 请选择地域・ 请选择地域・ 请选择 TKE 集群・ 请选择 TKE 集群・ 请选择 TKE 集群・ ・ 请选择 TKE 集群・ ・ 请选择 TKE 集群・ ・ 请选择 TKE 集群・ ・ 请选择 a S空间 ・ 介片持续部署管理集群已有资源 ① ・ 微定 取消

创建项目令牌



如果上一步检查发现没有 K8s Secret,则在 CODING 项目设置 > 开发者选项 > 项目令牌 页面创建一个具有读取制品库权限的令牌:

CODING Workshop -					
← 项目设置	项目设置 / 项目令	牌 / 新建项目令牌			
或目与成员项目协同	新建项目令牌				
☑ 项目公告	令牌名称			过期时间	
开发者选项	pull-artifacts			2046-04-06	
℃ 研发规范 beta	制品仓库权限 统一配置所有制品仓 	席权限 〇 指定制品合	的库配置权限		
	制品仓库名称		访问权限	操作权限	
	* 项目内所有制品库		 ✓ 读取 拉取制品库 	○ 读写 拉取、推送制品库	
	新建	取消			
项目令牌创建完成。					
← 项目设置	开发者选项	项目令牌(5)			
北 项目与成员	接口与事件	项目令牌只用于操作	F项目内的功能模块,	只对当前项目有效。 不能跟个人	、令牌通用
☑ 项目协同	兰联合库管理				
□ 项目公告	大职已件自注				
▶ 开发者选项</th <th>坝日安碑</th> <th>令牌名称</th> <th>用尸名</th> <th>密码</th> <th></th>	坝日安碑	令牌名称	用尸名	密码	
℃ 研发规范 beta	Service Hook 凭据管理	pull-artifacts	ptjkppb0m6gr	32c4********f815	5

创建 K8s Secret

把上一步获得的项目令牌创建为 K8s Secret,命令:

kubectl create secret docker-registry coding-docker \ --docker-server=仓库域名 --docker-username=项目令牌用户名 --docker-password=项目令牌密码 ? 说明:



您可以在 CODING 制品仓库中的配置凭据中找到仓库域名。

制品仓库	全部制品 仓库管理◆ 操作指引	× 配置访问令牌 输入密码后系统将自动生成访问令牌,并填入指引命令:		◆设置仓库	创建制品仓库版本覆盖策略
Docker	配置凭据	请输入密码 生成个人令牌作为凭据			
	拉取	设置凭证 请在命令行执行一下命令登陆仓库:			操作指引
	镜像源加速 🎸	in -u l:	版本数	操作	
				每页显示行数	: 15 💌 📘 1

或通过云厂商的网页创建:

🕗 腾讯云	总览	云产品 🗸 📄 网站备案	+ 婦 ()
容器服务		← 新建Secret	
〓 概览			
◎ 集群		名称	请输入名称
			最长63个字符,只能包含小写字母、数字及分隔符("-"),且必须以小写字母开头,数字或小写字母结尾
应用中心		Secret类型	Opaque Dockercfo
🔆 Helm应用			opuluo bookoong
② 镜像仓库		生效范围	● 存量所有命名空间(不包括kube-system、kube-public和后续增量命名空间)
运维中心			○ 指定命名空间
🗖 日志采集		合库域名	语输入博文亦(P
□ 告警设置		6,4-%	141411/ V-94/Tal 246/1
◇ 事件持久化		用户名	请输入第三方仓库的用户名
③ 健康检查		密码	请输入第三方仓库的登陆密码

使用 K8s Secret

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





参考资料

• 从私有仓库拉取镜像 - Kubernetes 官方文档



Kubernetes 云账号的最小权限要求

最近更新时间: 2021-08-09 11:31:54

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,详情请参见 开通服务。

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", "serviceaccounts", "pods/logs"]
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: 2023-06-15 15:56:35

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,详情请参见 开通服务。

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 Kubenetes 云账号时,可以选择 Kubeconfig 或 Service Account 认证方式:

← 部署控制台	云账号管理	0	绑定云账号
	CODING CD 基于云原生的能力管理部署过程,可以方便快速部署于 Kubernetes 、腾讯云弹性	伸缩、腾讯云 TKE 三种环境。在此页面可以管理您的云账号。重	云账号类别
₩ 四用	技名称搜索		
② 云账号	账号名称	账号状态	
■ 主机管理	● flaskapp EBR CODING Docker 包括	▶ 正失败 ▷ ○ 重新验证	成比使metes 勝抗芸 乙紙等名称。 支持大小写字母、中划线和下划线 ③ 提示 ③ 提示 请确保您的 Kubernetes 集群已开放公网访问,并将 CODING 持续部署的公网 IP 段添加到集群 访问控制列表自名单。 CODING 持续部署的公网 IP 段 212.64:10.50/24 212.129.144.0/24
«			送程认证方式 *

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: de

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: LS0tLS1CRUdJTiBDRVJUSUZADQFURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ0lCQVRBTkJna3Foa2IHOXcwQk FRc0ZBREFWTVJNd0VRWURWUVFERXdwdGFXNXAKYTNWaVpVTkJNQjRYRFRJd01ESXhNREEwTURJMU1Wb1hEVE13TURJd09EQTBNREk xTVZvd0ZURVRNQkVHQTFVRQpBeE1LYldsdWFXdDFZbVZEUVRDQ0FTSXdEUVIKS29aSWh2Y05BUUVCQIFBRGdnRVBBRENDQVFvQ2dn RUJBTUc1CjYwU0txcHVXeE1mWIVabmVFakM5bjFseHFQSzdUTIVQbzROejFWcWxaQkt6NzJDVVErZjBtVGNQLy9oS3BQUVAKaG9pNndya XJRUmVERTErRFIrOTZHVDIrSGZ3L2VHQTI5ZmErNS80UG5PWIpTUEVpS3MxVVdhc0VqSDJVZG4xTwpEejVRZk1ESkFjZlBoTzV0eUZFaGZ Na2hid0Y2QkJQNnh5RmJJdXl4OThmZGx5SWJJUnpLSml6VWZQcUx2WUZoCmFQbjF4WFZyT2QyMnFtblgzL2VxZXM4aG51SmpJdIVPbWR DRIhjQVRYdE00Wmw2bERvWUs2VS9vaEFzM0x4VzAKWUV4ZkcxMzFXdjIrR0t4WWV2Q0FuMitSQ3NBdFpTZk9zcVljMmorYS9FODVqdzc ySIFkNGd6eGIHMCszaU14WApWaGhpcWFrY10wZIRCc0FtZHY4Q0F3RUFBYU5DTUVBd0RnWURWUjBQQVFIL0JBUURBZ0trTUIwR0ExVW RKUVFXCk1CUUdDQ3NHQVFVRkJ3TUNCZ2dyQmdFRkJRY0RBVEFQQmd0VkhSTUJBZjhFQIRBREFRSC9NQTBHQ1NxR1NJYjMKRFFFQkN3 VUFBNEICQVFDKzFuU2w0dnJDTEV6eWg0VWdXR3ZWSIdtV2ItM2dBWFFJU1R2WG56NXZq0XE3Z0JYSwpCRVUyakVHTFF2UEJQWUZwUj hmZIICZCtqT2xtYS9IdU9ISmw0RUxhaHJKbnIwaU9YcytoeVIpV0ZUKzZ2R05RCmY4QnAvNTIkYzY1ejVVMnIUQjd4VkhMcGYzRTRZdUN2N mZhdy9PZTNUUzZUbThZdFBXREgxNDBOR2ZKMHIWRIYKSzZsQnI5THMwMzZzT1V5ZUJpcEdu0UxyKytvb09mTVZIU2dpaEJIcEI3ZVVvYk 05YU1ram1Hb2VjNk5HTUN3NwpkaFNWTmdMNGxMSnRvRktoVDdTZHFjMmk2SWIwbkJrdUIHUWRJUFIQnF6MkN5eVMyRkZmeEJsV2Im NmcxMTFTClphSUIpQ0ILbXNqeDJvTFBhOUdNSjR6bERNR1hLY1ZyNnhhVQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg== server: https://*****



name: demo

contexts:

- context:

cluster: demo

user: demo

name: demo current-context: demo

preferences: {}

users:

- name: demo

user:

client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURBRENDQWVpZ0F3SUJBZ0lCQWpBTkJna3Foa2lHOXcwQkFRc 0ZBREFWTVJNd0VRWURWUVFERXdwdGFXNXAKYTNWaVpVTkJNQjRYRFRJd01ESXhNekV5TXpreU5sb1hEVEl4TURJeE16RXINemt5Tmxv d01URVhNQlVHQTFVRQpDaE1PYzNsemRHVnRPbTFoYzNSbGNuTXhGakFVQmdOVkJBTVREVzFwYm1scmRXSmxMWFZ6WlhJd2dnRWlN QTBHCkNTcUdTSWIzRFFFQkFRVUFBNEICRHdBd2dnRUtBb0ICQVFDdEp5MThYOGVBaVIJc1g3Z0xQazBOZEFuRINJU0kKeXNGMGIzS21CT k9VclRIcGtnaUFwRldmZDJaVXNZR3Iwd0VBL0FIWm9PMUxPUHdqYzEyb2o1bGlwWlNNWTIMaQpVeW9UQ3huREZIVDJIQnIPNCtGRk5pVC tXTU5FTURURWxERXhlano1WVIwb1pZVjN2V2Z5T3I2SIBna1dFU29ICIVVcnVvQmRRbU5LUzhCeXhIUmFvcnFJUFRVRGkzUFJIbTIGZERKV 1INTWpnbDZmbmdHWkRQMnpjTINFZjRGNzYKc2FUK0VhMU1kbjV5akRCNXczaHJvZXBBclc0QVUyR3NTRFZyTHY2UDFiSWV0RDd0NjZrT 1BBNklIUIRKbUNLLwp2aEJrbGVPMGFwcERzTERDT3hpMkc2Q1BRNDZVYVZUOEhZMk9sQU5nSmtRRDNwYjFXTnlhQIVpRkFnTUJBQUdqC IB6QTINQTRHQTFVZER3RUIvd1FFQXdJRm9EQWRCZ05WSFNVRUZqQVVCZ2dyQmdFRkJRY0RBUVJJS3dZQkJRVUgKQXdJd0RBWURWUjB UQVFIL0JBSXdBREFOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBTHVrZ1dxWnRtbkZITIcxYgpqL012ekcxTmJyUEtWYXVobmI5RzRWWnRZ YzR1Uk5iSGhicEhEdThyamc2dVB1Y0xMdHAzbIU2UGw4S2J2WFpiCmpINmJQR2xvV3VBcFIrVW9KRFQ2VEpDK2o2Qm5CSXpWQkNOL21I SWVPQ0hEK1k5L2dtbzRnd2Q4c2F3U0Z1bjMKZTFVekF2cHBwdTVZY05wcU92aUkxT2NjNGdxNTd2V1h1MFRIdUJkM0vtQ2JZRXUzYXhOL 25IdnhOYnYxbDFRSQovSzRaOWw3MXFqaEp3SVIBaHUzek5pTWpCU1VTRJjkZnd2NmFnclhSUnN6b1Z4ejE5Mm9qM2pWU215cXZxeVFr CmZXckpsc3VhY1NDdTIKUE440UQrVXkwVnZXZmhPdmp4cXVRSktwUW9hMzlQci81Q3YweXFKUkFIMkk5Wk1IZEYKNkJQRVBRPT0KLS0 tLS1FFKQqQ0VSVEIGSUNBVEUtLS0tLQo=

client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVQSLS0tLQpNSUIFcEFJQkFBS0NBUUVBcINjdGZGL0hnSW1DTEYrNEN6NU5EWF FKeFVpRWINckIkRzI5cGdUVGxLMHq2WkIICmdLUIZuM2RtVkxHQnE5TUJBUHdCMmFEdFN6ajhJM05kcUkrWlc5R1VqR1BTNGxNcUV3c1p 3eFIwOWh3Y2p1UGgKUIRZay9sakRSREEweEpReE1YbzgrV0VkS0dXRmQ3MW44anNyeVQ0SkZoRXFCMUZLN3FBWFVKalNrdkFjc1IwVw HJFZzFheTcrajlXeUhyUSt6ZXVwRGp3T2ICMFV5Wmdpdjc0UVpKWGp0R3FhUTdDd3dqc1kKdGh1Z2owT09sR2xVL0IwdGpwUURZQ1pFQ Tk2VzIWamNtZ1ZJaFFJREFRQUJBb0lCQUdDazVGTnVGaWtkRndYegphd01EZy9oRIV3ckZIZ3hIdHRCcFFBRi80aVF5d3hBT0RTYIIFbDVPUT FSME900FBoNWpvRDVSTHFRWjZTT2owCmhFc0gwMTRYVFNWS3RqTFNua0pBeU9GRWNyL0hFdjjDSFINRzVJRCtSQWEwTFUrbk13bmRv MWpCcG9IY21uRXAKeTNHOUt3Ukkxc04xVXhNQWdhVk12NWFocGE2UzRTdENpalh3VGVVWUxpc1pSZGp5UGljUWlQN0xaSnhBcjRLTgp 5ZUp3OFZiMWtOekRzMSs4YzA4ViNhCkkzNHluOHpjZm1WTkZOUm1ZS1FMK1FGbnZ3ZXM4NFRleGVBb1hGY2FOcVZTNDNVV2JHRW02 ZThDZ1IFQXpINFkKeUxYQ2pWNVppaiFsUTdkQ2FweEQvL0dCT2JsemRocXRuSiI10WxyVkQ4Y3RvUEVKYVkwVFFWc3FaNVhCdTNtVApLb 0w4bmZjWHg4cWR3Z3gvZFRHa2c3d00rZFkrUTFuVDNOWnRFSVdVUkR3T0hLT1N1Tm5kdnU1a1Q4aXRrdHhhCktrYVISMINOT25iMIFzb 1ZHa3F5OS9QK0xWL0FyeWdScmtaYXVNc0NnWUVBaGpUTkdUYzltaXNxeTV2d0FHTzkKM1NFSG9YRIJmbWgvakM2RFAxMUdMUE9iT21 qRIREbzFCS0F5d3JBSm1RWUsyUkpzdUh4L2dGY3JJY1F6bCtqaQpvRGRWaDM1a0tEUTIFd00vaIE0TIIIdW1XOVhITjVvWmNMbTFISmNnL3 Bsd1pzVkxFNFFVaHVzT1IUZUs2TVgyCkU0OS8rcHJBSFVEOG5oNIpuWGN4U1BjQ2dZRUFub0lQcDZab1MwSjliMi9VbTJ2YS9vNnJ0TDBpO Tlpc2JCTWEKNFR6RFAzTXBlc3owYIRZN1JYaW1ncDcybytiY3IUNUtMZVhISnB3RVBPL1R3SUs0Tk8veUxzZzN3TExOR0RCegphRC9Ra1hBU eXM4TFpHekFNTWdmN1VRQ0dVZERrT2dJRHdzd0dxVVIxQ2ZLcnlGYVdCOUJvSi8KMnJMVmVYNDVXTnFpa0tCMlgvckdRbGFIK25YalRBa XA0S0JFVkp3L1E9PQotLS0tLUVORCBSU0EgUFJJVkFURSBLRVktLS0tLQo=



云账号相关问题

最近更新时间: 2023-06-15 15:53:19

使用 TKE 云账号进行部署时报错怎么办?

使用 TKE 云账号部署时报错:

Deploy failed: error: unable to recognize "STDIN": no matches for kind "Deployment" in version "apps/v1"

这是因为以下两个资源在 istio 已经废弃,需要手动在集群删除资源。

kubectl delete apiservice vlalphal.rbac.istio.io kubectl delete apiservice vlalpha2.config.istio.io

提示"获取临时密钥失败"怎么办?

若在关联 TKE 集群时提示"获取临时密钥失败",需检查团队是否绑定了 <mark>腾讯云账号</mark>。确认已绑定后,联系团队所有者或管理员前往腾讯云控制台的**访问管 理**页添加 CODING_QCSRole 角色。

🔗 腾讯云	产品▼						Q. 🕜 小程序	「二」 集团账号	▼ 备案			费用 ▼	👥 -
访问管理	角色											CAM角色使	用说明 🛙
語 概览 ら 用户	0	为什么我的账户出现了新角色? 在云服务中完成特定操作(如授 或者、如果您在某项服务开始考	权创建服务角色) 时,云服务会向用户 "持服务相关角色之前已在使用该服务	发送创建服务角色的授权请求,您同意并授权后,会自动创建脂 通过维任等方式告知您后,则会自动在她的联户中创建养鱼色	《务角色并关联相关策略。 1.								
2) 用户组 回 策略	s ti k			, ADLEM (1197) (2011) ANNOULL, 3012 (1997) (2020) 3000 (110)	2 8				CODING	QCSRole		Ø	Q ¢
12 角色													
回 身份提供商	fi c	i色名称 CODING QCSRole	角色ID	角色载体 产品服务 - coding	角色描述 当前角色为CODING DevOps服务角色。	该角色将在已关联策略的机	R	会话最大报 云 2 小时	操时间	创建时间 2020-01-07	10:00:10	操作	
(1)访问密钥	, ,	50项		·				1	0 ▼ 条/页	H 4	1	/1页 →	H

各项云账号的区别是什么?

• TKE

如果您是从 腾讯云控制台 中登录 CODING DevOps 平台,才会在云账号中显示此类账号,支持使用 K8s 部署。

Kubernetes

支持 Kubeconfig 和 Service Account 两个常用凭据与 K8s 部署。

• 腾讯云账号

即腾讯云 API 密钥,不支持 K8s 部署,常用于弹性伸缩组部署与 tsf 部署。

提示云账号已存在怎么办?

云账号支持自定义命名,您可以选用新的云账号名称以避免冲突。

无法删除云账号如何解决?

删除云账号时提示:云账号已被应用使用,无法改名或删除,若需删除,请先解除关联关系。 出现此问题的原因是应用与云账号相互关联,删除云账号前需删除该账号所关联的应用资源。您可以前往 **部署控制台 > 应用 > Kubernetes 集群** 删除已绑定



的应用资源。							
← 部署控制台	← ccj-k8s-test000 - <u>部間:高厚</u> Ku I作負載 服务 云景号:全部 ◆ 命名空時:	bernetes 集群	· 提示 Q	0	nginx-deplo	oyment-854bb79cb6-ssd92	操作 ▼
✿ 应用	名称	命名空间	云账号	健康社	基础信息		制除编辑
 ② 云账号 圖 主机管理 	eployment nginx-deployment	default	1	-	创建时间 云账号	2021-08-12 19:23:03 ccj-k8s-test	
	V001 nginx:1.14.2 & Lord Balancer: service nginx:	IVC			命名空间 资源类型 QOS Class	default pod BestEffort	
	eployment prinx-deployment	default	ccj-k8s-test	_	资源类型 日志	pod 点击查看	
	Voor nginx:1.14.2 - A Load Balancer: service nginx-	IVC			状态		
					 Initialized Ready 大杉 	大约 1 天 22 小时 59 分钟 以前 5 1 天 22 小时 58 分钟 以前	
					Containers	Ready 大约1天22小时58分钟以前	
					PodSchedul	led 大约 1 天 22 小时 59 分钟 以前	



常见错误码

最近更新时间: 2023-08-10 11:14:18

Deployment exceeded its progress deadline

错误截图如下:



这是由于 deployment 的 pod 没有成功运行所导致的错误。可以通过以下方式查看 pod 事件详情,需要根据 pod 的事件排查问题。

方法一

在产品页面单击 方块 查看事件。

事件 …						,* 全屏				
首次出现时间	类型	最后出现时间	内容	详细	信息			开始时间 2021-0	9-10 20:02:2	5
2021-09-10 19:52:18	Warning	2021-09-10 20:02:31	FailedScheduling	0/1 (nod pod	nodes are available: le.kubernetes.io/unr didn't tolerate.	1 node(s) had taint eachable: }, that the				
关闭								启动时间	#EB3	
								2021-09-10 20:02:25	6秒	^
加設大 🕒 已終止 🗐	耗时 00分06秒					DeployStatus Task Status	Artifact Status Manifest To Sta	bilize)		□ 源数据
						'deployment test' in ogress deadline	'default' for acc	ount iptestteam15085: De	ployment e	ceeded its pr
	ccj-	k8s-poc		◎ 未启用		Deployment test		放 着	Yami 内容 i	洗转查看资源详情
波 🕑 成功 总耗制	00分02秒					ScalingReplicaSet				
						11 天 18 小时 36 分钟 L Scaled up replica set to	人前 est-b7dfd78d6	i to 1		

方法二 前往您的集群查看 pod 事件。 例如:



• 在 TKE 界面查看 pod 的事件:

← 集群(广州) / cls-mqSu0shv(ccitest) / Deploym Pod管理 修订历史 事件 日	rent:nginx-deployment1(default) 志 详情 YAML						
1032 (1992) (1992)							國际Evicted Pod
实例名称	状态	实例所在节点P	实例P	运行时间 ①	01891/0	重启次数 ①	操作
Inginx-deployment1-7df5bf9b4b-86sc	Running	10.0.0.17	172.17.0.38	7d 21h 12m	2021-09-14 17:17:25	0次	销股重建 远程登录
Inginv deployment 1-7df5bf9b4b-gidmi Inginv deployment 1-7df5bf9b4b-gidmi	Running	10.0.0.17個	172.17.0.37	7d 21h 12m	2021-09-14 17:17:25	0次	纳税而建 远程使果
Inginu deployment 1-7dfSbf9b4b-zfh9j R	Running	10.0.0.17	172.17.0.36	7d 21h 12m	2021-09-14 17:17:25	0次	转股而建 远程登录

• 使用命令行查看 pod 的事件:

kubectl describe pod [pod name] -n [pod location] kubectl describe pod nginxtest -n test

Failed to pull image xxxxxx

这是因为再 yaml 文件里配置了拉取 CODING docker 镜像,但没有配置 CODING docker 仓库的凭据,需要在 yaml 文件添加相关的 imagePullSecrets。

若要解决此问题,需要将 CODING 登录凭据添加到您的 kubernetes 集群的 secret,请参见 部署 Kubernetes 资源时如何拉取私有库镜像?

Deploy failed: error: unable to recognize "STDIN"

Ø	部署 Dep	loyment			
_	状态 耗时	已终止 5 秒	开始时间	2021-09-14 14:53:41	
	阶段详情				
	状态	脚本名称	启动时间	耗时	
	😑 已终止	部署 Deployment	2021-09-14 14:53	8:41 5秒	^
	DeploySt	atus Task Status Artifact Status			□ 源数据
	S Exc De ers	eption (Monitor Deploy) ploy failed: error: unable to recog ion "v1"	nize "STDIN": no n	natches for kind "Deple	oyment" in v

出现此报错的原因一般为 yaml 文件格式问题。建议在本地集群验证 yaml 文件,确保能够在本地集群正确执行之后,再粘贴至 CODING 上使用。

ConfigMap/Secret

问题详情:

部署完成后, ConfigMap/Secret 参数中新增了 v00x 版本号。

解决办法:

在 configmap/secret 的 annotations 参数中加入 strategy.spinnaker.io/versioned: 'false' 命令行。