









【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。



快速入门

最近更新时间: 2023-06-05 17:51:29

本文为您详细介绍在 CODING 中持续部署的基本操作。

前提条件

使用 CODING 项目管理的前提是,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**立即使用**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

CODING 持续部署用于把控构建之后的项目发布与部署交付流程。能够无缝对接上游 Git 仓库、制品仓库以实现 全自动化部署。在稳定的技术架构、运维工具等基础上,具备蓝绿发布,灰度发布(金丝雀发布),滚动发布,快 速回滚等能力。

下文将以一个简单的 Demo 项目为例,演示如何使用 CODING 持续部署控制台将应用发布至腾讯云集群。

前置准备

- 开启持续部署设置权限,单击了解 权限详情。
- 一个可被 CODING 持续部署访问的 Kubernetes 集群,单击了解如何申请 腾讯云标准集群。
- 导入 示例代码库。
- Docker 制品仓库,单击了解如何使用项目中的 Docker 制品仓库。

操作步骤

步骤1:初始化项目

 此步骤主要用于配置持续部署所涉及代码仓库与制品仓库。在代码仓库中点选导入外部仓库,访问 示例仓库并克 隆仓库地址。



🔗 codingtest-cd				15
				 克隆
文件 提交 分支 标签		克隆仓库		已复制
P master ~ 介 / 输入	以查找文件	HTTPS ▼ https://e.codin	g.net/codingtest-cd/k{)emo.git	Ū
管理员 更新文件 service.yaml			最后提交 01c504d016 🗍 🗧	F3个月前
gradle	管理员	first commit		8 个月前
k8s	管理员	更新文件 service.yaml		3 个月前
lib	管理员	first commit		8 个月前
src	管理员	v2:web api		6 个月前
] .gitignore	管理员	first commit		8 个月前
Dockerfile	管理员	docker		8 个月前
🗋 build.gradle	管理员	docker		8 个月前
🗋 gradlew	管理员	first commit		8 个月前
🗋 gradlew.bat	管理员	first commit		8 个月前
settings.gradle	管理员	first commit		8 个月前

2. 代码仓库导入后,前往制品库创建 Docker 制品仓库,获取下图中的制品拉取地址。关于如何创建制品仓库,详 情请参见 Docker 制品库。

 ← ? ● ●<th>◆ 操作指引</th><th>拉取 输入以下拉取相关信息,生成</th><th>艾拉取命令:</th><th>×</th><th></th><th> │ ¢设置</th>	◆ 操作指引	拉取 输入以下拉取相关信息,生成	艾拉取命令:	×		│ ¢设置
仓库	配置凭据	制品名称:	hello-world			
概览	拉取	制品版本:	latest			操作指引
镜像历史 命令 ADD file CMD ["/ MAINTAI ENV NO COPY ., WORKDI RUN /bit <nil></nil>	镜像源加速	请在命令行执行以下命令进行 pull <mark>StrayBirds-docke</mark>	7拉取: r.pkg.coding.net/flask-demo/cd-d しd:latest		推送信息 推送人 推送时间 大小 hash	 ● 主账号 2 天前 61.62 MB sha256:44d5f1eb23a6227 332267f1 3352dd5 01f98 linux/amd64
	③ 帮助中心					



步骤2: 获取并关联云账号

因使用了腾讯云容器服务,部署后应用将发布至集群,本示例使用的团队账号已在**团队管理>服务集成**中关联 腾讯 云账号。

- 1. 单击首页左侧的部署控制台,在云账号中绑定腾讯云账号。云账号名称可以自拟,选择地域后将自动获取对应集
 - 群。

← 功能设置	持续部署设置	云账号管理	0	绑定云账号
 ・ 功能设置 ジ 项目协同 ジ 代码扫描 beta ・ 持续集成 ・ 持续部署 ・ 公开资源 ・ MD 模板 	持续部署设置 云原生邮署 云账号 主机邮署 堡垒机 主机组	云账号管理 CODING CD 基于云原生的能力管理部署过程,可以方便快速 查看云账号说明 ② 按名称搜索 低 Go 已授収 CODING Docker 仓库 ④ Demo 已授収 CODING Docker 仓库 ④ Joe ④ test	志部署于 Kub 账号状态 已验证 已验证 已验证 已验证	第二日日 第二日 (1) (1) (1) <t< td=""></t<>
				目动生成 CODING Docker 仓库访问凭业① 请选择命名空间 ▼
				 允许持续部署管理集群已有资源① 确定 取消

2. 自动生成的制品仓库访问凭证将存储于命名空间,您可以在腾讯云控制台中新建。

於 ■ 2 ○ 於 ■ 2 ○ ☆ 前讯云 ■ 点覧 ■ 2 ○ ○ ○ □ □	5产品 ▼				容	路服务 🛛 🛚) Q		1		• ••• •	•
容器服务	← 集群(/广州) / cls	-ow54ato	94(体验版集群)								YAM	L创建资源
器 概 览	基本信息		Namespace									操作指南 🖸
◎ 集群	节点管理	•	新建					多个关键字用	竖线 " " 分隔,	多个过滤标签用	回车键	α¢±
 □ 弹性谷器 ◇ ◇ ⇒ ⇒	命名空间											
↔ 服务网格	工作负载	Ť	名称	状态	描述		创建时间		3	操作		
应用中心	服务与路由	•	default 🕞	Active	-		14:32:34		1	記额管理删除		
☆ 应用 □ 链像☆ 広	配置管理	*	kube-node-lease 🗖	Active	-		2021-07-27 14:32:32		ı	記额管理 删除		
 回 硯像包库 · ·<	授权管理	Ŧ	kube-public 🖻	Active	-		2021-07-27		1	記额管理 删除		
运维中心	存储	Ŧ					2021.07.27					
⑦ 集群运维 ~	组件官理 日志		kube-system 🛅	Active	-		14:32:32		ī	記额管理 删除		
	事件		第1页							:	20 - 条/页	• •
谷谷現像服労 ഥ ∞ DevOps ☑												- 0
E 快速入门												支持

步骤3:配置应用



1. 成功添加云账号后,在部署控制台中单击创建应用,填写应用名与选择部署方式。

< ● 部署控制台	应用			创建应用
◎ 应用	云账号:全部▼ 关联项目:全部▼ 排序;	创建应用	×	
③ 云账号	flaskapp <i>≡</i>	应用名 * ⑦		
主机管理	▲ 云帐号: 0 ◆ 关联项目: 1	cd-demo 部署方式 *	◆ 关联项目: 0	8 A Ø
	test ☴ 2 云张号: 0 🔹 关联项目: 1	 Kubernetes(TKE) 時讯云弹性伸缩 主机组 主机组 合中划线 '-' 的应用不支持主机组、弹性伸缩和 TSF 部署 		
		描述 请输入描述信息		
		确认 取消		

2. 选择部署到 Kubernetes 集群模板,填写名称与描述后完成创建。

← 部署控制台	创建部署流程	创建流程
✿ 应用	复制现有流程 Kubernetes 腾讯云弹性伸缩	
② 云账号		
■ 主机管理	部署 Helm 应用到 Kubernetes 集群 名称 *	
	配置 Bake (Manife 部署 (Manifest) cd-demo	
	描述	
	部署 Deployment 和 Service 到 Kubernetes 集群 请输入描述	
	配置 部署 Deploy 部署 Service	
	部署到 Kubernetes 集群前进行人工确认 确认 取消 配置 人工确认 Bake (Manife 部署 (Manifes	
	并行部署 Deployments 和 Services	
	配置 部署 Deploy 部署 Service	
	部署 Deploy 部署 Service	

步骤4:修改镜像地址与命名空间



1. 创建制品仓库后,	将上文中获取的制品拉取地址填写至代码仓库中 /k8s/deployment.yaml /	中的 image 地址。
-------------	--	---------------------

刘贤	【 · 提交 · 分支 · 合并请求 · 版本 · 对比 · 设置	○ • 已开源 + 创建代码仓库 -
R states	😵 master 👻 🛖 ≜ 查找文件 🗸 k8s / deployment.yaml	※ 克隆
> in gradle/wrapper	文件 历史 6 代码对比 按行查看	∠ 编辑 …
🗋 deployment	1999 主账号 更新文件 deployment.yaml	最后提交 🗊 于 1 天前
🗋 service.yaml	deployment.yaml 504 Bytes	
> 📕 lib	2 kind: Deployment 3 metadata:	
> 🦰 src	4 labels: 5 app:	
🗋 .gitignore	6 name: k loyment 7 namespace: cd-demo	
Dockerfile	8 spec: 9 replicas: 1	
🗋 build.gradle	10 selector: 11 matchLabels:	
🗋 gradlew	12 app: 13 template:	
🗋 gradlew.bat	14 metadata: 15 labels:	
Settings.gradle	16 app: 17 spec: 18 imagePullSecrets: 19 - name: coding-registry-cred- 20 containers: 21 - image: 'StrayBirds-docker.pkg.coding.net/flask-demo/cd-demo/hello-world' 22 name: 'C' 23 ports: 24 - containerPort:	

2. 接下来需导入云账号的 imagePullSecrets 至代码仓库中。在部署控制台 > 云账号中单击查看详情后,复制名

 称。					
③ 飞鸟集 ▼				×	
← 部署控制台	云账号管理 CODING CD 基于云原生的	已经授权此 Kubernetes 集群访问 CODING Docker 仓库 imagePullSecrets: coding-registry-cred-252585		部署控制台 以管理您的云账号。查看云账号说明 (2)	绑定云账号
▲ 应用	按名称搜索	已授权的命名空间: cd-demo			
◎ 云账号		示例用法		77.000.007	
■ 主机管理	账号名称	metadata:	Сору	更新时间	
	● Go 已授权 CODINC	labels: app: flask-backend name: flask-backend-deployment		2021-07-27 16:27:47	编辑 …
	Demo 已授权 COD	spec: replicas: 3		2021–07–26 16:54:29	编辑 …
	Soe	selector: matchLabels: app: flask-backend		2021-06-25 10:59:44	编辑 …
	e test1	template: metadata:		2020-07-29 16:59:07	编辑 …
	⊗ test	<pre>tabets: app: flask-backend spec: containers: - image: wzw-test-docker.pkg.coding.net/cd-show/release/fla backend name: flask-backend imagePullSecrets: - name: coding-registry-cred-252585 关闭</pre>	ask-	2020-06-10 15:54:00	編辑 …

3. 粘贴至代码仓库中的 deployment.yaml 文件中,同时需注意 namespace 内容是否与上文中指定的命名空间一致。



kitsDemo =	浏览 提交 分支 合并请求 版本 对比 设置	○ ● 已开源 + 创建代码仓库 ▼
ft shilers	😵 master 👻	※ 克隆
> 📄 gradle/wrapper		
✓ k8s	文件 更改对比	提交 取消
🗋 deployment	🐵 主账号 更新文件 deployment.yaml	最后提交 🗊 于 1 天前
	deployment.yaml 504 Bytes	
	1 apiVersion: apps/v1	
> 📔 lib	2 kind: Deployment 3 metadata:	
> 📄 src	4 labels:	
	5 app: 1	
gitignore	7 names epicoymetric	
	8 spec:	
	9 replicas: 1	
🗋 build.gradle	10 selector:	
	11 matchLabels:	
🗋 gradlew	12 app:	
	13 template:	
☐ gradlew.bat	14 metadat	
	17 spec:	
	18 imagePullSecrets:	
	19 - name: coding-registry-cred-252585	
	20 containers:	
	21 - image: 'StrayBirds-docker.pkg.coding.net/flask-demo/cd-demo/hello-wor	ld'
	22 name:	
	23 ports	
	24 – c	

4. 同一层级的 service.yaml 文件中的 namespace 内容也需保持一致。

 kBuDemo - 	浏览 提交 分支 合并请求 版本 对比 设置	○ ●已开源 + 创建代码仓库 ▼
ft shifters	😵 master 👻	 宏陸
> 📄 gradle/wrapper Y 📄 k8s	文件 历史 4 代码对比 按行查看	⊿ 编辑 …
🗋 deployment	♀ 管理员 更新文件 service.yaml	最后提交 🗊 于 3 个月前
service.yaml	service.yaml 181 Bytes	
 > Iib > src 	1 apiVersion: v1 2 kind: Service 3 metadata: 4 name 5 namespace: cd-demo 6 spec: 7 selector: 8 app: 9 ports. 10 - port 11 targ 12 type: LoadBalancer	
settings.gradle		

步骤5: 配置部署流程

进入部署流程配置页面,可以为此流程设定:

- 流程的执行选项(在此示例中保持默认即可)。
- 部署 Deployment 阶段以及部署 Service 阶段所需制品。
- 手动或自动触发。
- 1. 首先配置部署(Manifest)阶段。基础设置选择已绑定的云账号,在 Manifest 来源选择 CODING 代码库, 填写相应的路径,镜像版本配置选择自动获取。



cd-demo 🖉		部署 Deployment	
∮? 基础配置	+	● 部署 (Manifest) 配置 执行选项 通知 描述	
	阶段类型:部署 (Manifest)	● 使用制品 输入内容	
hello-world		制品来源	
		CODING 代码库	~
x8s/deployment.yaml naster		项目	
k8s/service.yaml		Flask Demo	~
		仓库	
		shilers	~
		默认分支或标签 ①	
		master	~
		文件路径	
		k8s/deployment.yaml	~
		高级配置	
		跳过 SpEL 表达式计算 ⑦	

2. 配置部署 Service 阶段时步骤同上,但在文件路径处选择 k8s/service.yaml 文件。

步骤6: 触发器配置

1. 完成部署阶段配置后,您可以使用自动化触发器、手动提交发布单执行部署。

自动触发

在基础配置中点选触发器类型,选择 Docker 仓库触发器。当开发人员更新代码仓库并使用 CI 将镜像打包推送 至制品库后,Docker 镜像版本的更新将自动触发部署流程并将应用发布至 Kubernetes (TKE) 集群,完成后



可以在基础设施页面查看并确认应用是否发布成功。

← cd-demo 🥖	▶ 基础配置	
(+)	执行选项 自动触发器 启动参数 通知 描述	
↓↑ 基础配置 部署 Deployment 阶段类型:部署 (Manifest)	^ŧ > 自动触发器	
制品	✓ CODING docker 仓库触发器	6
k8s/deployment.yaml master	触发器启用开关	
k8s/service.yaml		
▲ hello-world	触发器类型	
	CODING docker 仓库触发器	~
	CODING docker 仓库触发器	
	TCR 个人版仓库触发器	Т
	TCR 企业版仓库触发器	
	TCR Helm 仓库触发器	
	Git 仓库触发器	
	✓ 启 webhook 触发器	
	暂 定时触发器	
	CODING Generic 仓库触发器	
	~ 通知	
	暂无通知	

手动提交发布单

若希望通过手动提交发布单的形式触发部署流程,那么可以将**应用**(如本范例的 flaskapp)与项目关联。在部 署控制台的**应用列表**中搜索项目名称进行关联:





2. 关联完成后,单击项目中的持续部署 > Kubernetes 手动提交发布单。

í.	〕 项目概览		🧲 cd-de	mo → 发布单 集群 部署流程			新建发布 🔻
?</th <th>▶ 代码仓库</th> <th></th> <th>状态:全部▼</th> <th>创建人: 全部 ♥ 部署流程: 全部 ♥ 发布时间: 最近十天 ▼ 搜索: 荷搜索关键字</th> <th></th> <th></th> <th></th>	▶ 代码仓库		状态:全部▼	创建人: 全部 ♥ 部署流程: 全部 ♥ 发布时间: 最近十天 ▼ 搜索: 荷搜索关键字			
Ċ) 代码扫描 beta	>					
~	• 持续集成	>	发布单状态	发布单名称	部署流程	创建时间	创建人
ŵ	持续部署	~	🕑 成功	gogo	cd-demo	2021-07-27 19:53:02	● 主账号
	Kubernetes		🕑 成功	20210726-cd-demo-cd-demo	cd-demo	2021-07-26 17:58:48	● 建账号
	弹性伸缩 主机部署		🕑 成功	20210726-cd-demo-cd-demo	cd-demo	2021-07-26 17:54:50	₩9 主账号
	网站托管						
	制品管理	>					
ъ	▶ 文档管理	>					

步骤7:发布完成

1. 发布成功后,可以查看发布的制品及启动参数及阶段执行详情等信息。

础信息	阶段	状态 成功 开始时间 2021-07-27 19:53:03
手动触发		耗时 19 秒
▲ 主账号2021-07-27 19:53:02	部署 Deployment 耗时:19秒	阶段详情
③ 25 秒		状态 脚本名称 启动时间 耗时
制品		⊘ 成功 部署 Deployment 2021–07–27 19:53:03 19 秒 ヘ
 StrayBirds-docker.pk g.coding.net/flask-d emo/cd-demo/hello -world latest 		DeployStatus Task Status Artifact Status ✓ Deployment nent 查看 Yaml 内容 跳转查看资源详
k8s/deployment.yaml master		ScalingReplicaSet 3 分钟 以前
k8s/service.yaml master		Scaled up replica set pyment-994479977 to 1

 当需要查看某个资源在集群中的运行状态时,单击集群下的工作负载即可查看详情(例如工作负载的 Pod 实例, 日志等信息)。



	及印字 未矸	部署流程	
工作负载 服务	云账号:全部▼	命名空间:全部 🔻 类型:全部	鄒▼ 状态:全部▼
名称		命名空间	云账号
leploymer	-deployment	cd-demo	Go

deployment-994479977				
2021–07–27 19:53:05				
Go				
cd-demo				
replicaSet				
deployme eployment				
	ployment-994479977 2021-07-27 19:53:05 Go ccd-demo replicaSet deployme eployment	ployment-994479977 操作 2021-07-27 19:53:05 Go cd-demo replicaSet deployme eployment		

镜像

StrayBirds-docker.pkg.coding.net/flask-demo/cd-demo/hello-world:latest ()

事件

1 x SuccessfulCreate

- 以前

Created pod: -deployment-994479977-nmkdx

LABELS

app app app

pod

3. 在腾讯云的容器服务中查看工作负载。

容器服务	← 集群(广州) / cls-ow54	to4(体验版集群) YAML创建资源	
■ 概览	基本信息	Deployment 操作指读	i 12
◎ 集群	节点管理 ▼		
② 弹性容器 →	命名空间	新進 監控 命名空間 cd-demo ▼ Label格式要求: name=value, 多个关键字用竖线 Q Q	+
ふ 边缘集群	⊤作负载 ▼	名称 Labels Selector 运行/期望Pod数量 Request/Limits 操作	
↔ 服务网格	Deployment	CPU: 无限制 / 无	
应用中心	 StatefulSet 	I/1 限制 更新Pod設量 更新Pod設置 更新Pod 認 更新Pod 認 更新Pod 認 DimPod 認 DimPod 認 Dim	
☆ 应用	 DaemonSet 	限制	
◎ 镜像仓库 ◆	- Job	第1页 20 * 条/页 4 >>	
凹 应用市场	CronJob		
运维中心			
② 集群运维 🔶 👻	服务与路田 *		
▶ 云原生监控	能置管理 🔹		
計 容器镜像服务 Ľ	授权管理		
∞ DevOps ⊠	存储 🔹		ら 支
🗈 快速入门	组件管理		持
	日志		œ
	事件		· 文 档