

# 持续部署 快速入门 产品文档



腾讯云

**【 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 】**

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

1. 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

文件 提交 分支 标签

**克隆仓库** 已复制

HTTPS

master / 输入以查找文件 ...

文件	操作者	提交信息	时间
更新文件 service.yaml	管理员		最后提交 01c504d016 于 3 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](#)。

更新时间

版本号

仓库

概览

镜像历史

命令

ADD file

CMD ["/

MAINTA

ENV NO

COPY .

WORKD

RUN /bi

<nil>

[帮助中心](#)

**操作指引**

配置凭据

**拉取**

镜像源加速

**拉取**

输入以下拉取相关信息，生成拉取命令：

制品名称:

制品版本:

请在命令行执行以下命令进行拉取：

```
pull StrayBirds-docker.pkg.coding.net/flask-demo/cd-d...ld:latest
```

推送信息

推送人 账号 主账号

推送时间 2 天前

**其他**

大小 61.62 MB

hash sha256:44d5f1eb23a6227332267f13a52dd501f98

系统架构 linux/amd64

## 步骤2：获取并关联云账号

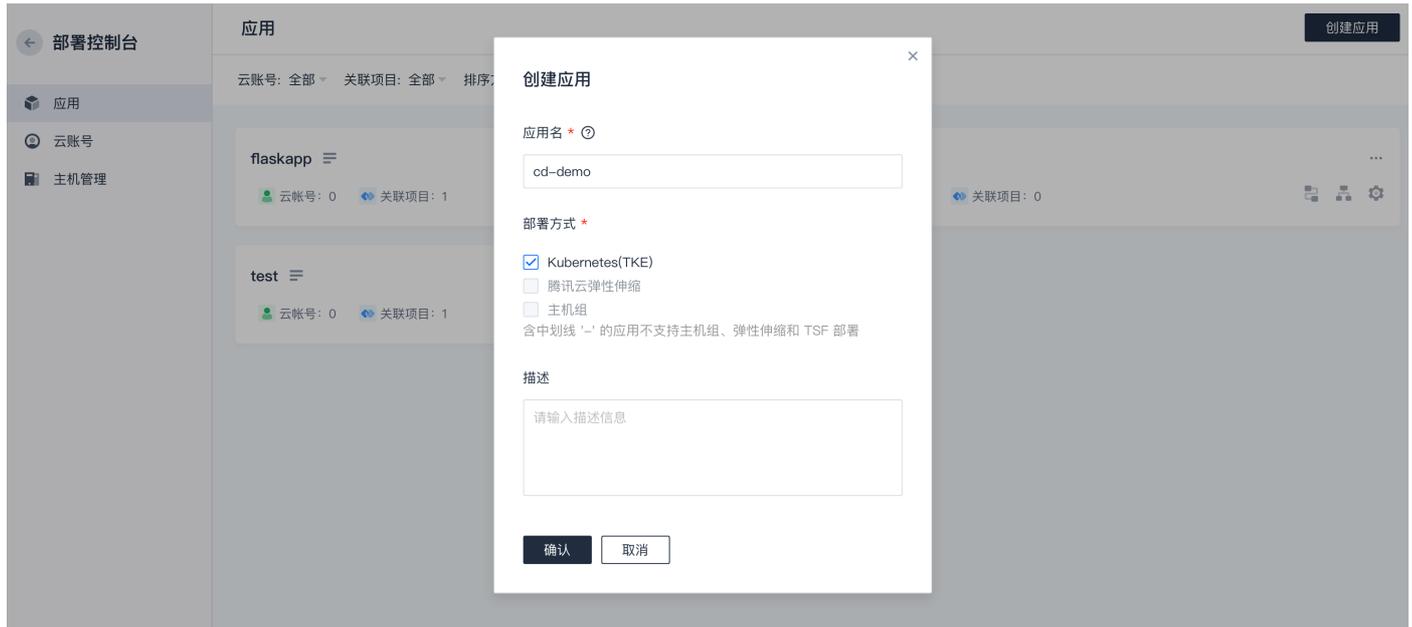
因使用了腾讯云容器服务，部署后应用将发布至集群，本示例使用的团队账号已在[团队管理](#)>[服务集成](#)中关联 [腾讯云账号](#)。

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。

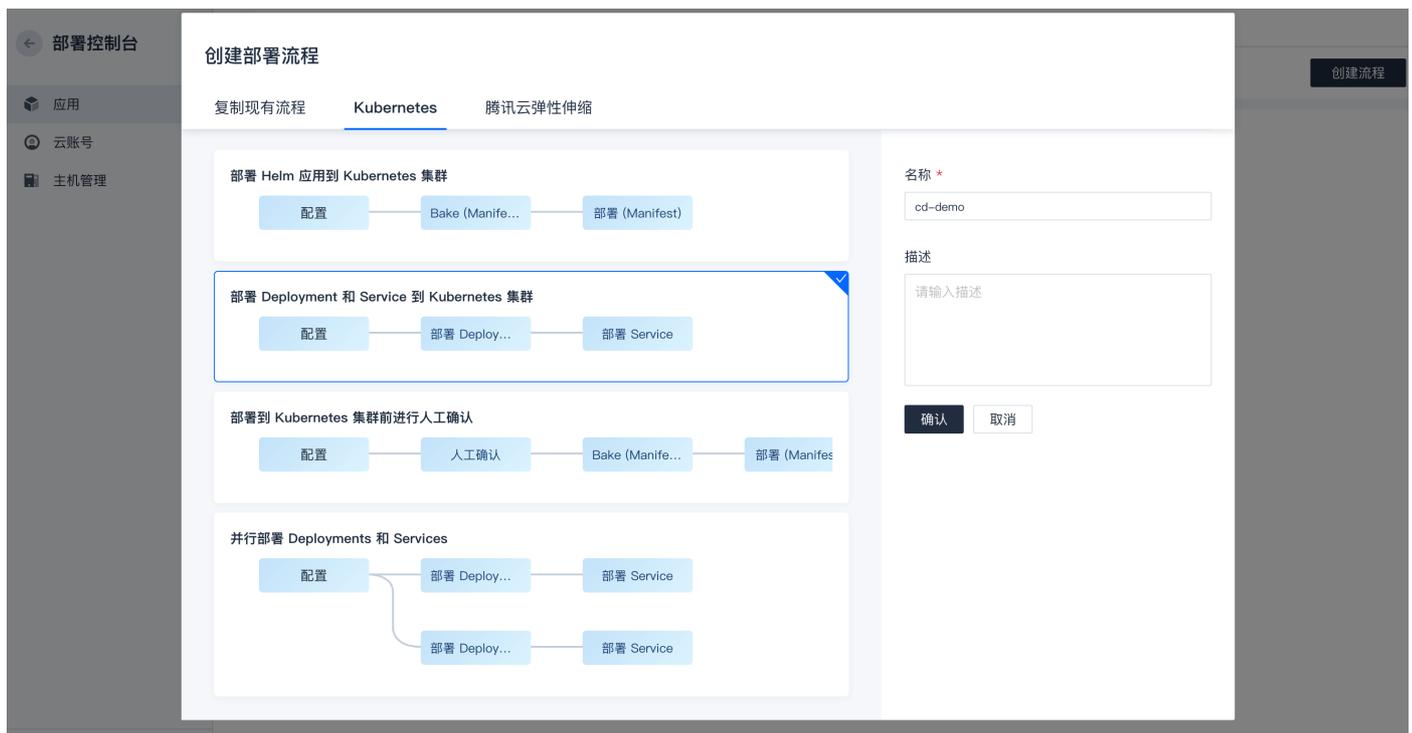
2. 自动生成的制品仓库访问凭证将存储于命名空间，您可以在腾讯云控制台中新建。

## 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。

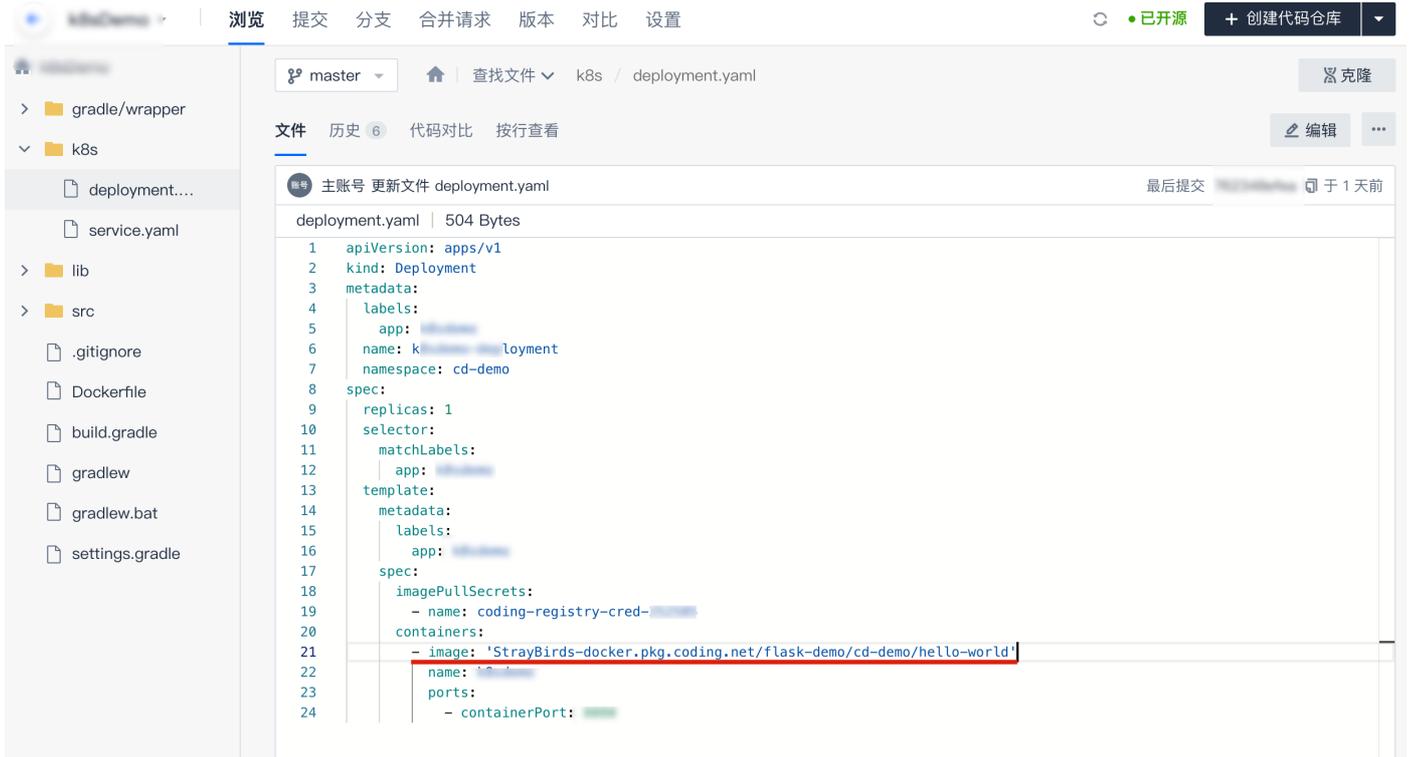


## 2. 选择部署到 **Kubernetes 集群** 模板，填写名称与描述后完成创建。

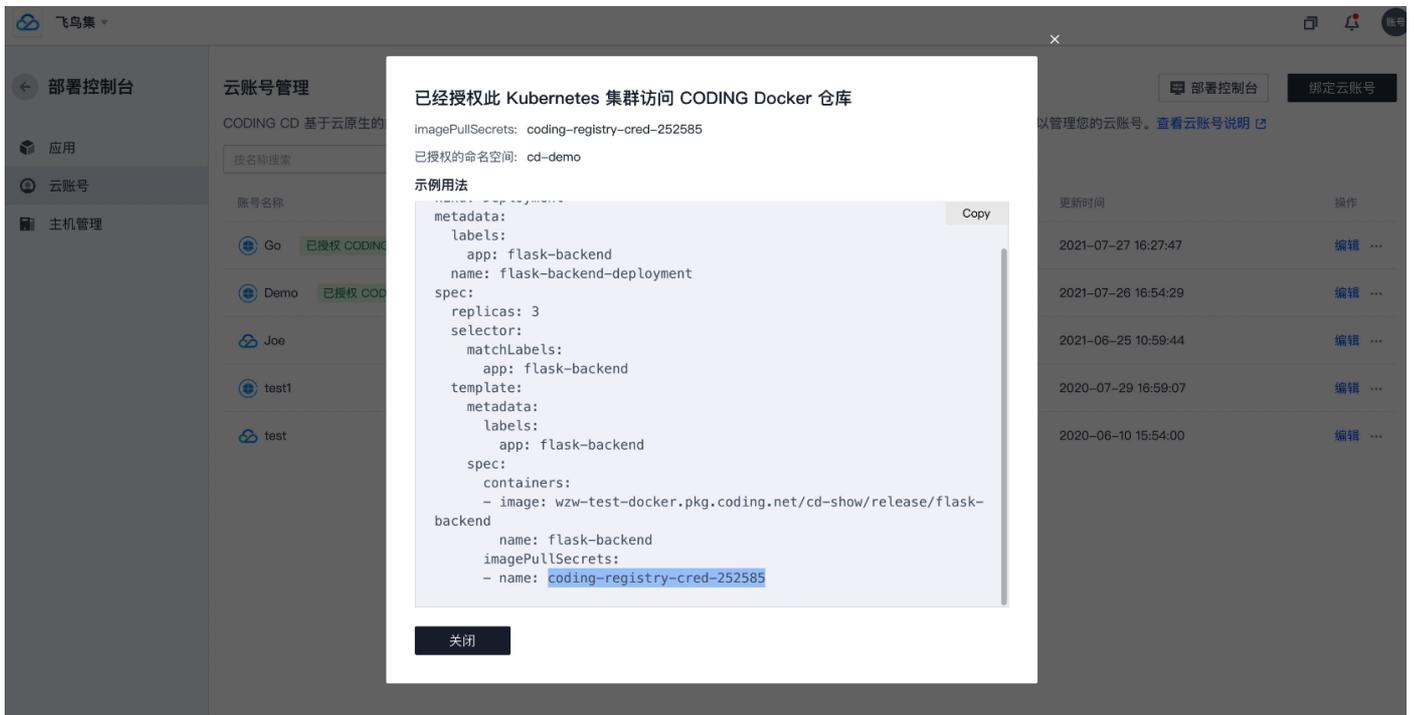


## 步骤4：修改镜像地址与命名空间

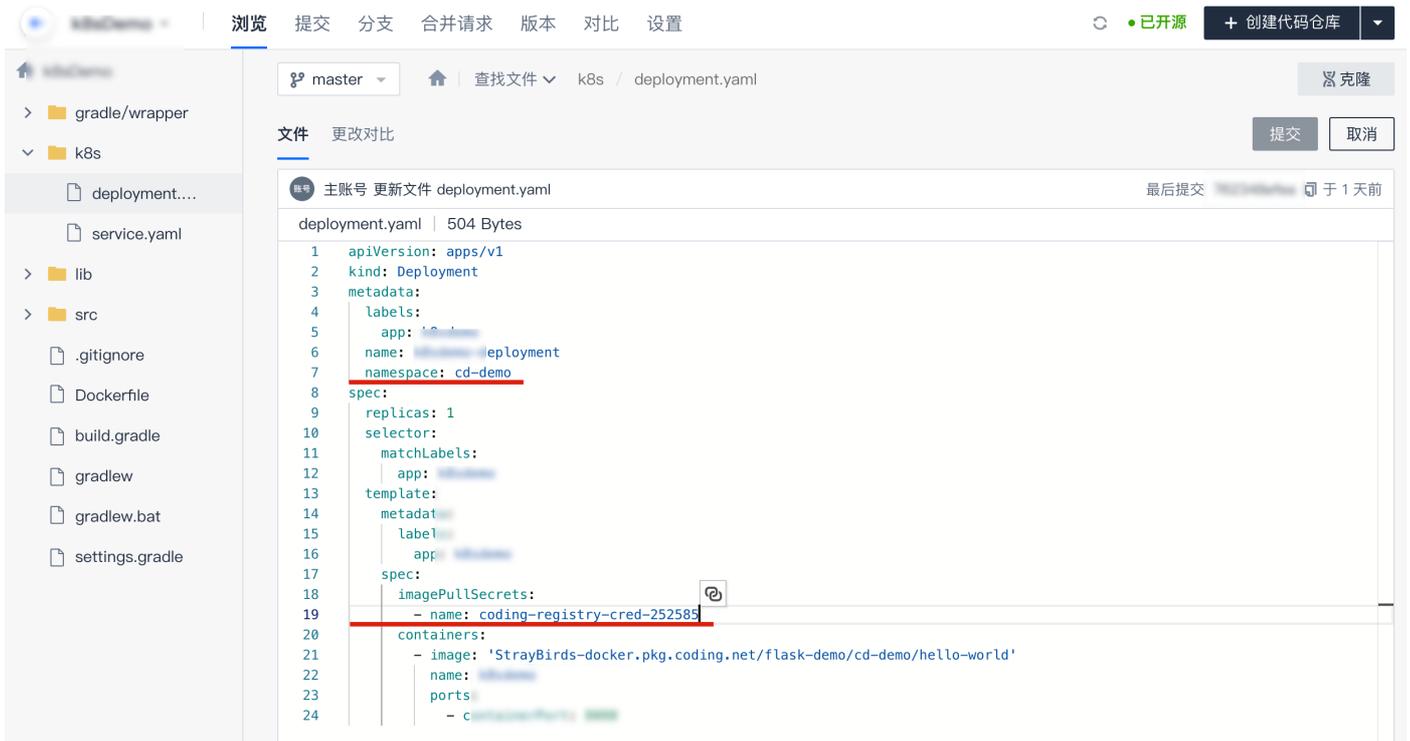
## 1. 创建制品仓库后，将上文中获取的制品拉取地址填写至代码仓库中 /k8s/deployment.yaml 中的 image 地址。



## 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. 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。



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



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

依赖阶段: 基础配置

部署 (Manifest) 配置 执行选项 通知 描述

Manifest 来源

使用制品  输入内容

制品来源

CODING 代码库

项目

Flask Demo

仓库

k8s

默认分支或标签

master

文件路径

k8s/deployment.yaml

高级配置

跳过 SpEL 表达式计算

镜像版本配置

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。

### 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。

发布单状态	发布单名称	部署流程	创建时间	创建人
成功	gogo	cd-demo	2021-07-27 19:53:02	主账号
成功	20210726-cd-demo-cd-demo	cd-demo	2021-07-26 17:58:48	主账号
成功	20210726-cd-demo-cd-demo	cd-demo	2021-07-26 17:54:50	主账号

## 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。

基础信息

手动触发

主账号

2021-07-27 19:53:02

25 秒

制品

- StrayBirds-docker.pkg.coding.net/flask-demo/cd-demo/hello-world latest
- k8s/deployment.yaml master
- k8s/service.yaml master

阶段

成功

部署 Deployment

耗时: 19 秒

部署 Deployment

状态 成功 开始时间 2021-07-27 19:53:03

耗时 19 秒

阶段详情

状态	脚本名称	启动时间	耗时
成功	部署 Deployment	2021-07-27 19:53:03	19 秒

DeploymentStatus Task Status Artifact Status

Deployment [script name] ment [查看 Yaml 内容](#) [跳转查看资源详情](#)

ScalingReplicaSet

3 分钟 以前

Scaled up replica set [script name]-994479977 to 1

### 2. 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）。

cd-demo
发布单 集群 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 状态: 全部

名称	命名空间	云账号
✓ deployer -deployment	cd-demo	Go

V001 StrayBirds-docker.pkg.coding.net/flask-demo/cd-demo/hello-world:latest Load E

### deployment-994479977

**基础信息**

创建时间: 2021-07-27 19:53:05

云账号: Go

命名空间: cd-demo

资源类型: replicaSet

控制器: deployment

**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-nmidx

**LABELS**

app: flask

app: flask-version=0.0.1

app: flask-version=0.0.1

pod: flask

### 3. 在腾讯云的容器服务中查看工作负载。

容器服务

- 概览
- 集群
- 弹性容器
- 边缘集群
- 服务网络
- 应用中心
- 应用
- 镜像仓库
- 应用市场
- 运维中心
- 集群运维
- 云原生监控
- 容器镜像服务
- DevOps
- 快速入门

集群(广州) / cls-ow54ato4(体验版集群)
YAML创建资源

Deployment

节点管理

命名空间

工作负载

- Deployment
- StatefulSet
- DaemonSet
- Job
- CronJob

自动伸缩

服务与路由

配置管理

授权管理

存储

组件管理

日志

事件

新建 监控
命名空间: cd-demo
Label格式要求: name=value, 多个关键字用竖线

名称	Labels	Selector	运行/期望Pod数量	Request/Limits	操作
deployment-994479977	app=flask	app=flask	1/1	CPU: 无限制 / 无限制 内存: 无限制 / 无限制	更新Pod数量 更新Pod配置 更多

第 1 页 20 条 / 页

版权所有：腾讯云计算（北京）有限责任公司

第12 共12页