CODING DevOps

持续部署

腾讯云

文档目录

持续部署
快速开始

最近更新时间：2025-03-17 15:33:52

> **说明：**
> CODING DevOps 于2025年9月1日起更新 CODING 订购方案，取消原标准版套餐，下线部分功能（**制品安全扫描、测试管理、测试协同、仪表盘、研发度量**），新注册团队用户界面无**持续部署、应用管理**功能，为确保您的使用权益和资产数据安全，请及时关注并处理， 了解更多详情 。

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，请参见 权限详情 。
- 导入 示例代码库 。点击了解如何 导入或关联外部仓库 。
- Docker 制品仓库，单击了解如何使用项目中的 Docker 制品仓库 。
- 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 。

> **说明：**
> 集群的推荐配置为 8 核 32 GB，需开启公网访问权限。

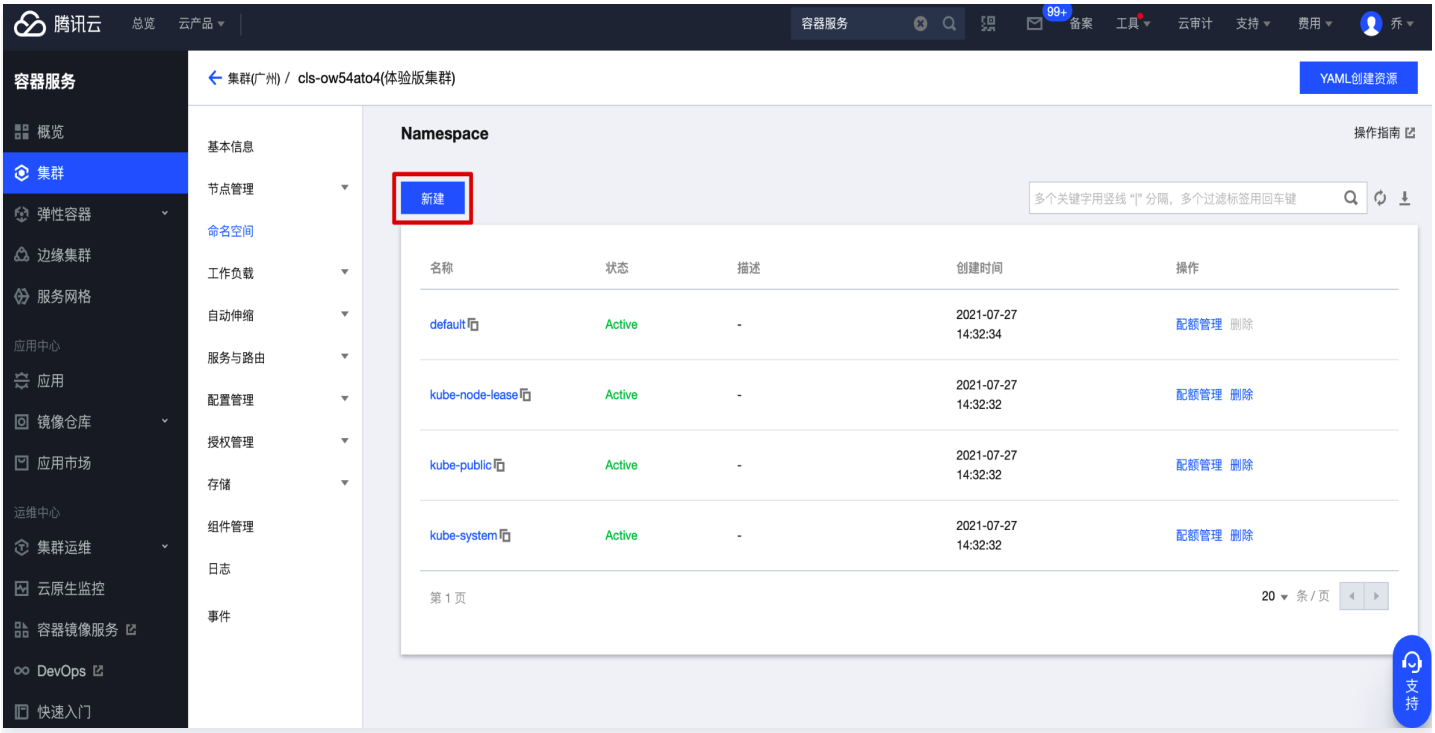
操作步骤

步骤1：获取并关联云账号

> **说明：**
> 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**团队设置中心 > 第三方应用**中关联 腾讯云账号 。

单击首页左侧的**基础设施**，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。

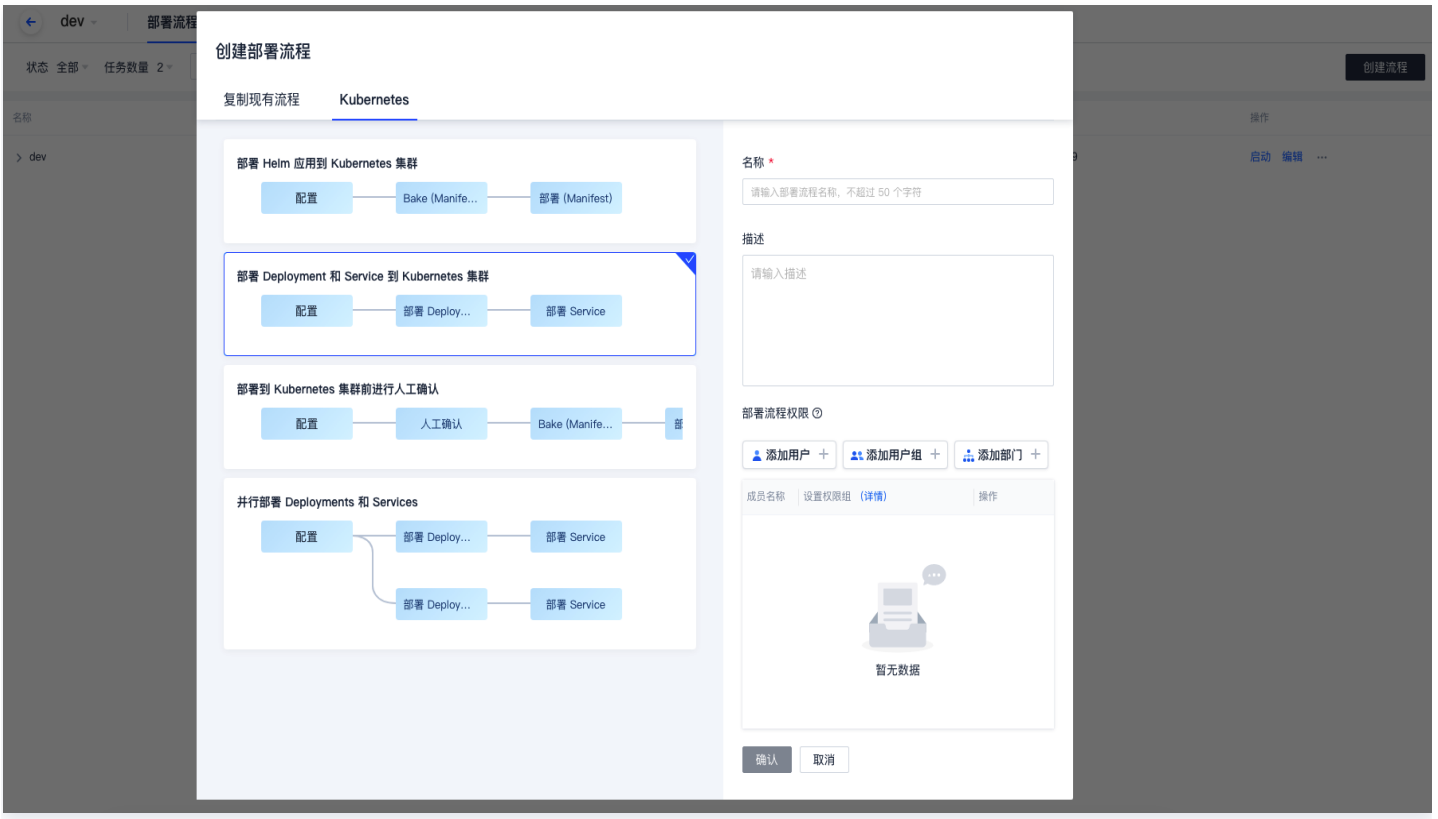
在集群中新建 `命名空间（Namespace）` 用于存储自动生成的制品仓库访问凭证，本文中所使用的集群命名为：cd-demo。



步骤2：配置应用

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。有关于**应用**的更多介绍，详细说明请参见 应用管理 。

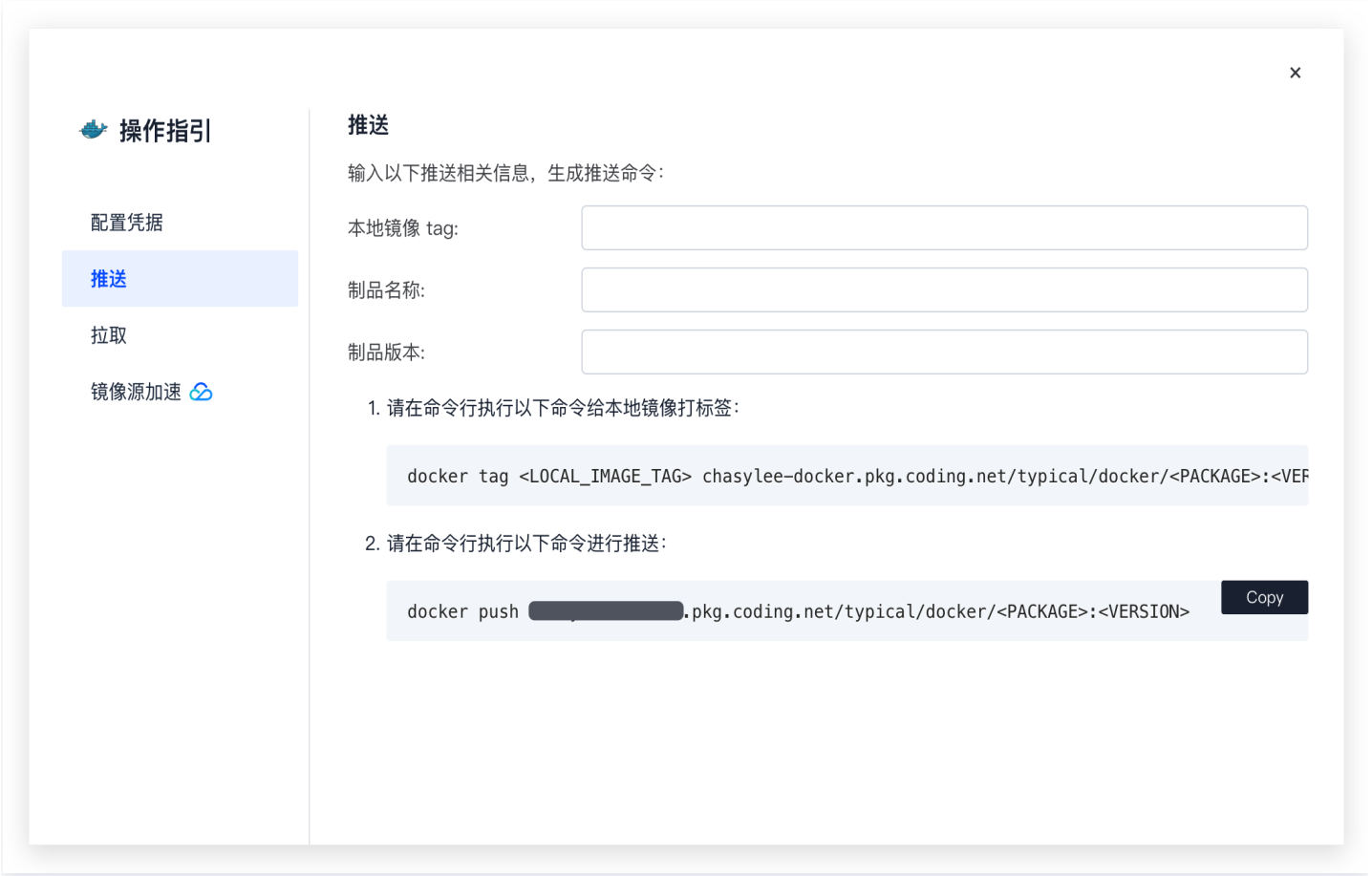
选择**部署到 Kubernetes 集群**模板，填写名称与描述后完成创建。



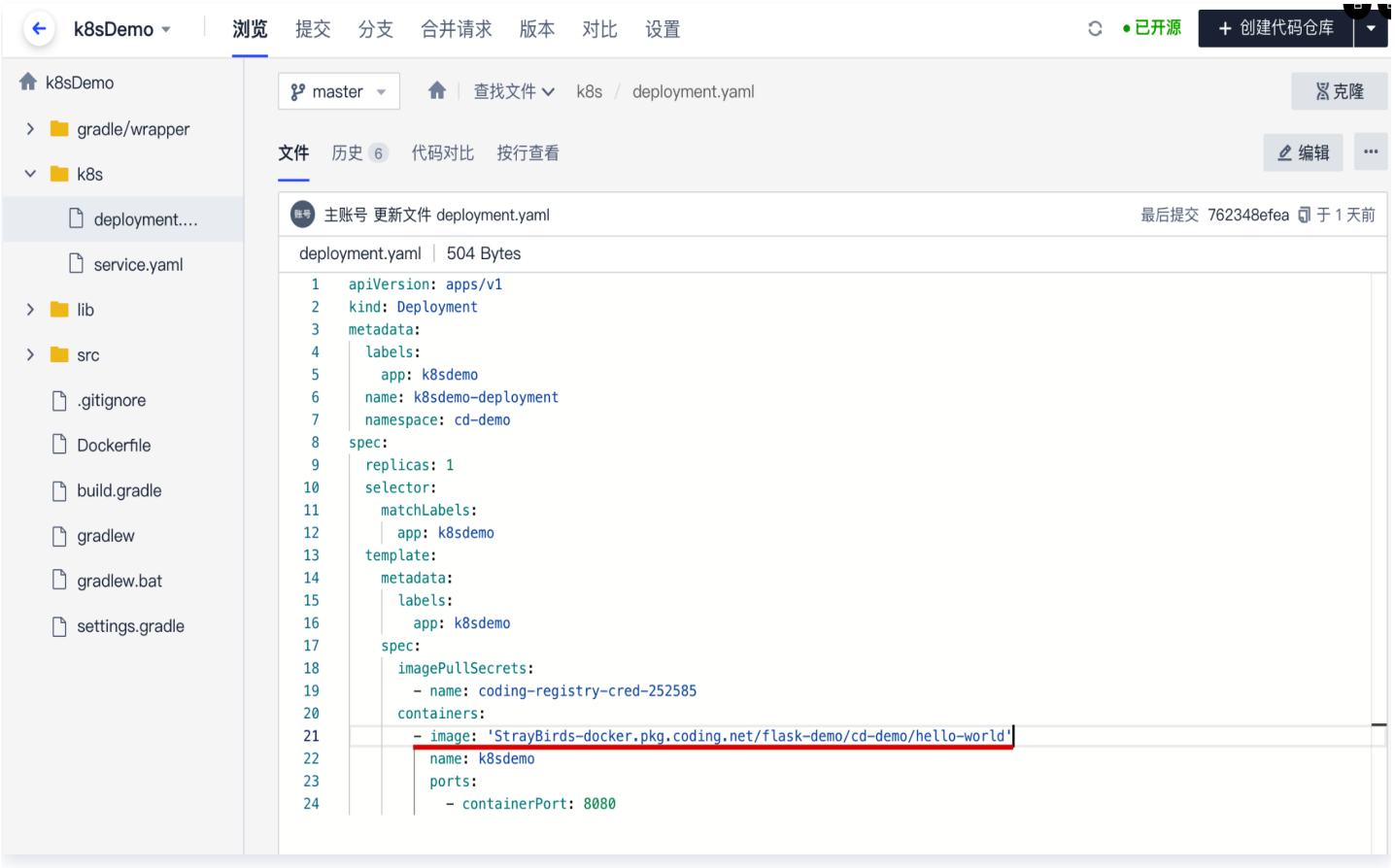
步骤3：初始化项目

此步骤主要用于配置持续部署所涉及的项目文件，默认您已完成前置准备中的导入示例代码库及创建 Docker 制品库。

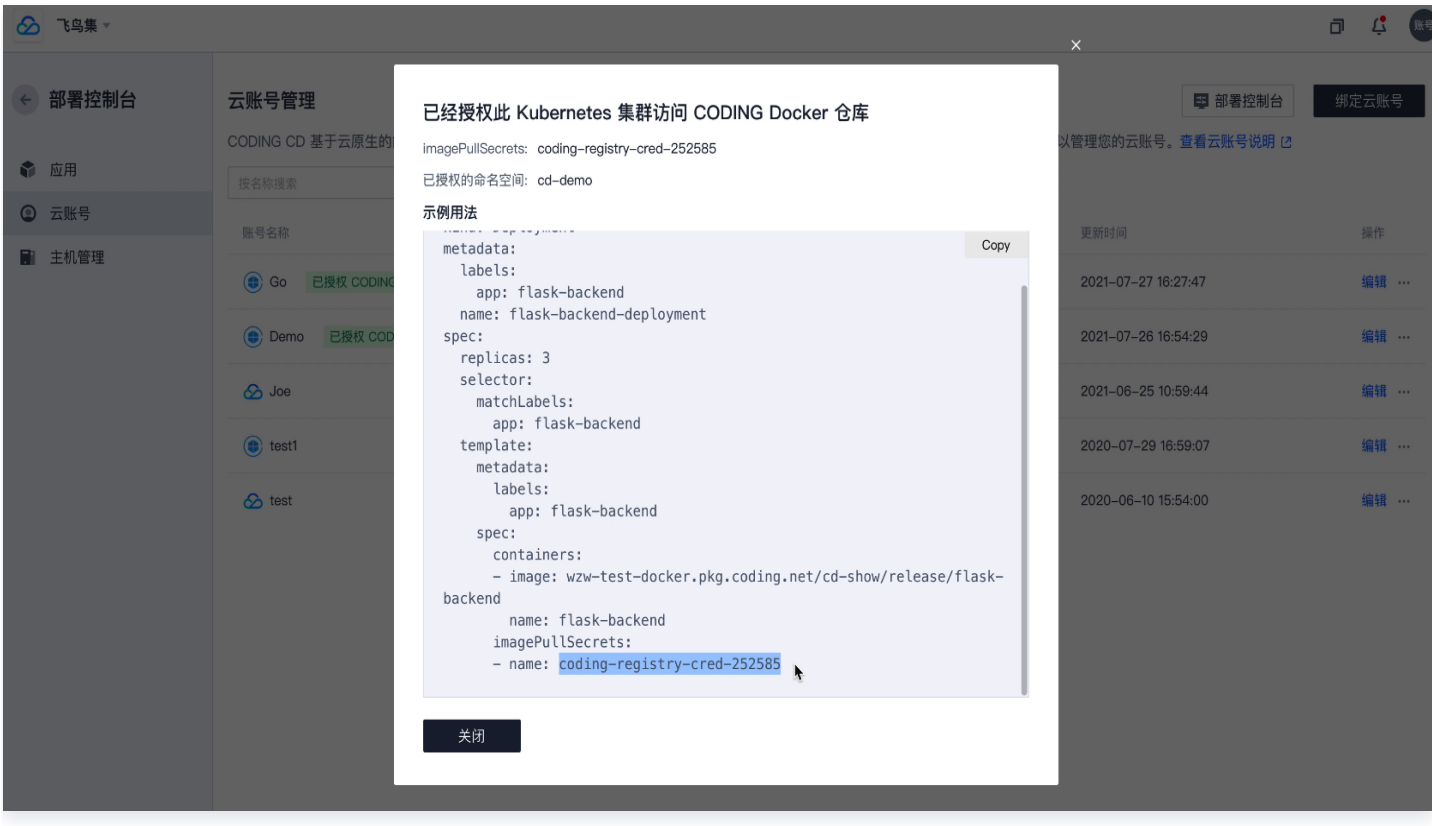
首先，将本地拟发布的 Docker 制品推送至 CODING 制品仓库，具体操作可参考 Docker 。



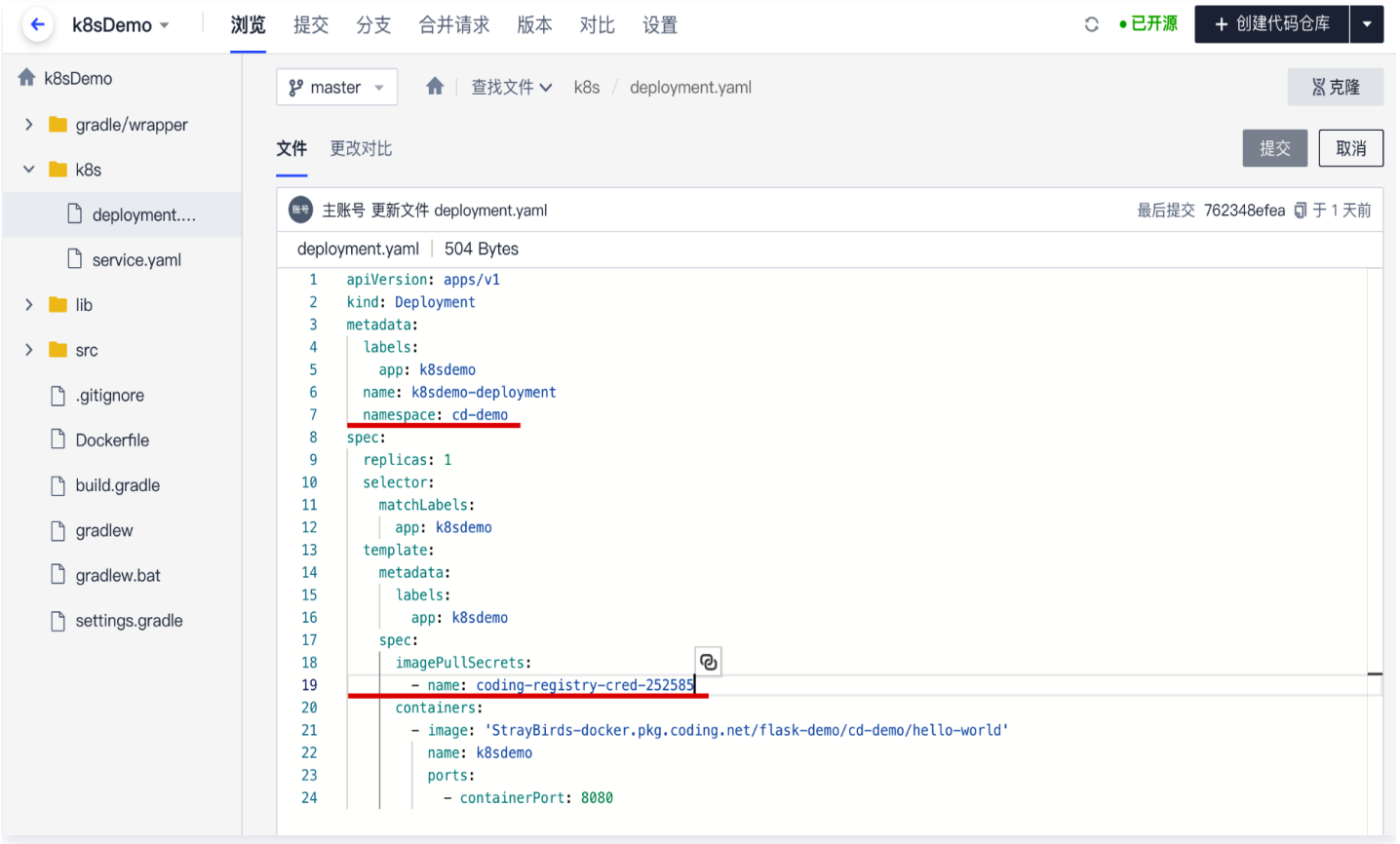
推送至制品仓库后，获取制品的拉取地址（即“操作指引” > “拉取”命令中的镜像路径）并填写至代码仓库中 `/k8s/deployment.yaml` 中的 image 地址。



接下来需导入云账号的 imagePullSecrets 至代码仓库中。在基础设施 > 云账号中单击查看详情后，复制名称。



粘贴至代码仓库中的 `deployment.yaml` 文件中，同时在 namespace 参数一栏中填写在上文中所创建的命名空间 cd-demo。



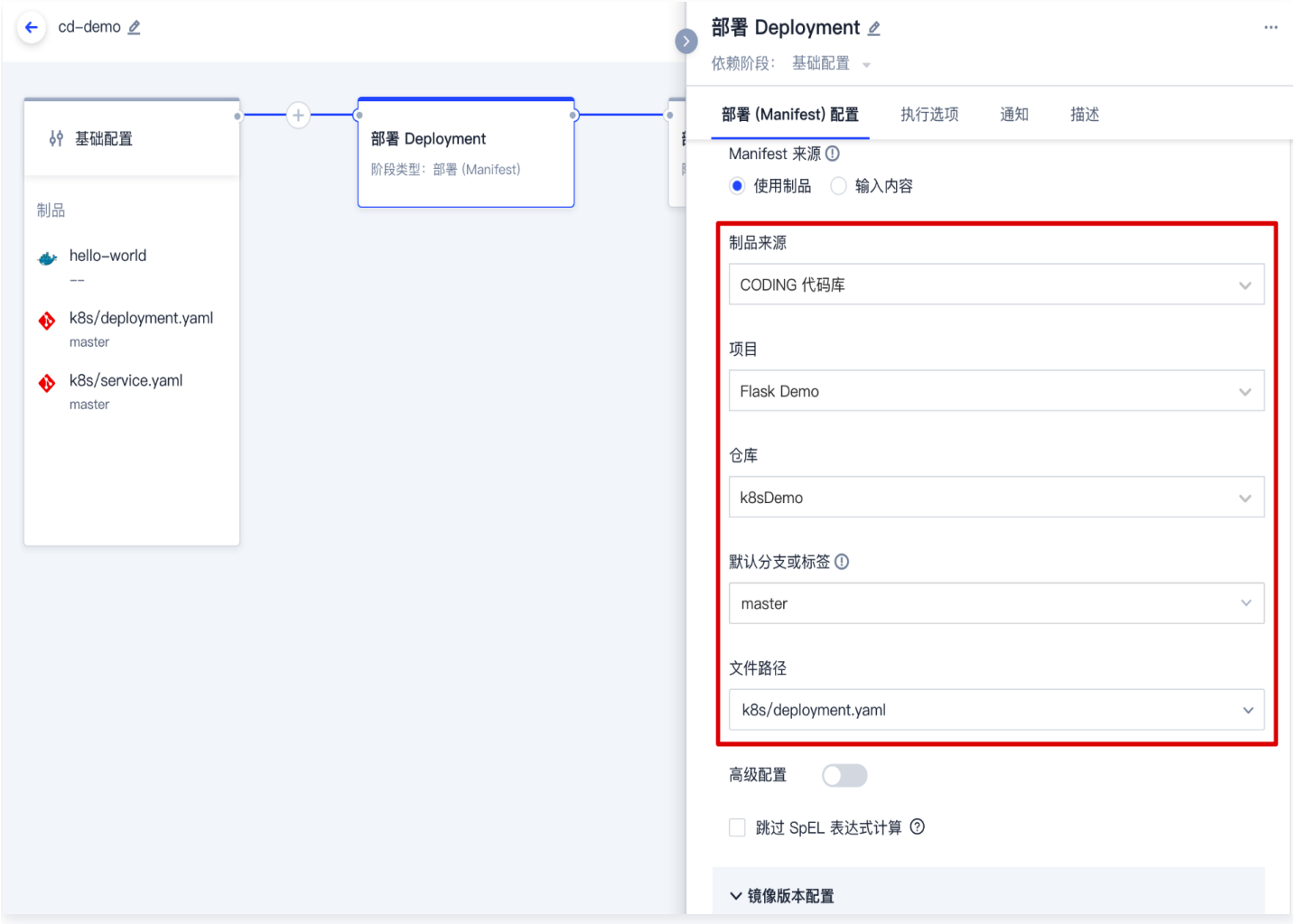
同一层级的 service.yaml 文件中的 namespace 内容也需保持一致。

步骤4：部署流程

在应用中心找到上述步骤 2 创建的应用，点击其名称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。

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。



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

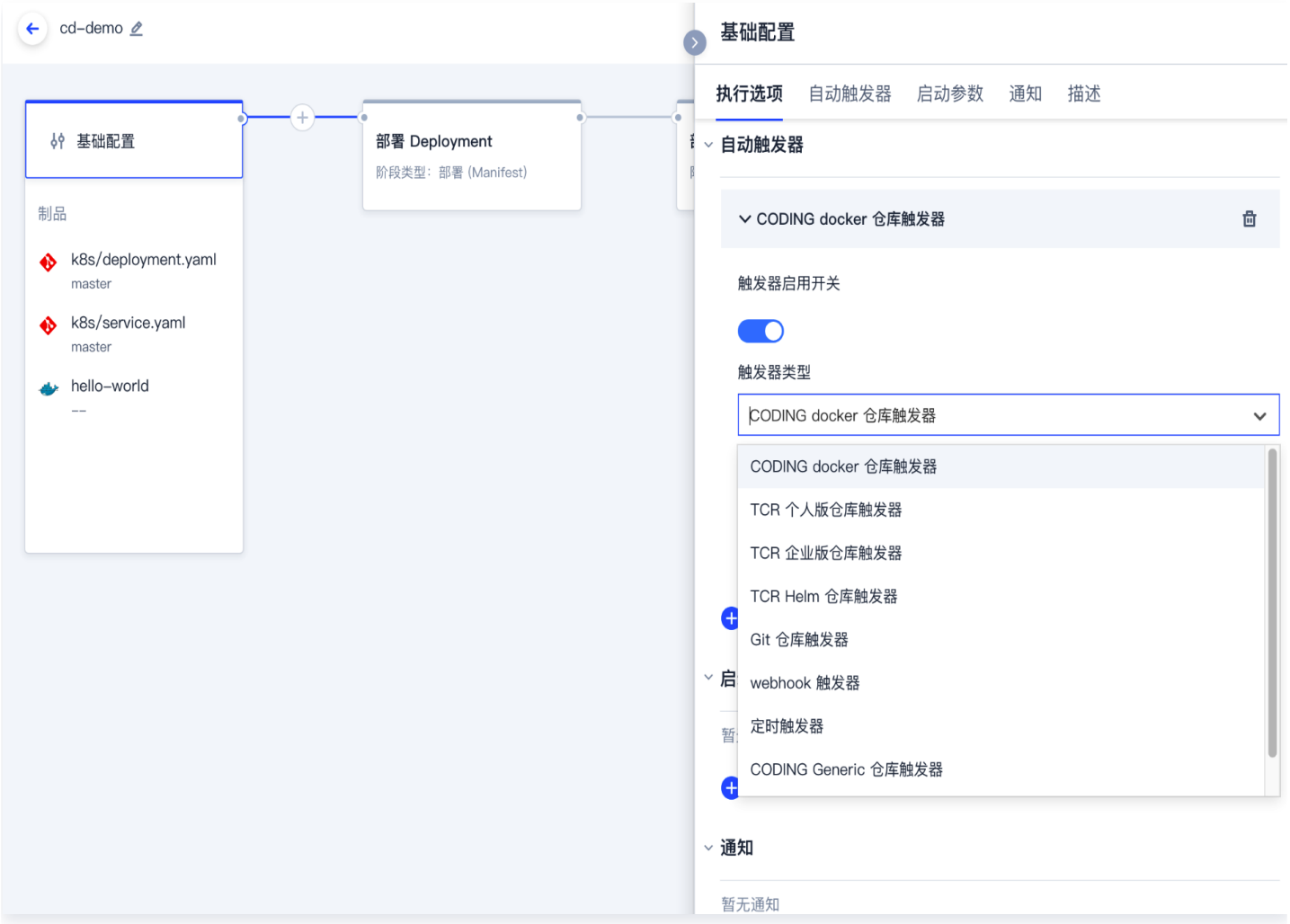
镜像版本配置默认选择自动获取镜像来源。若设置自定义版本规则，将仅传送特定的 image 版本信息号至集群中。

步骤5：触发器配置

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

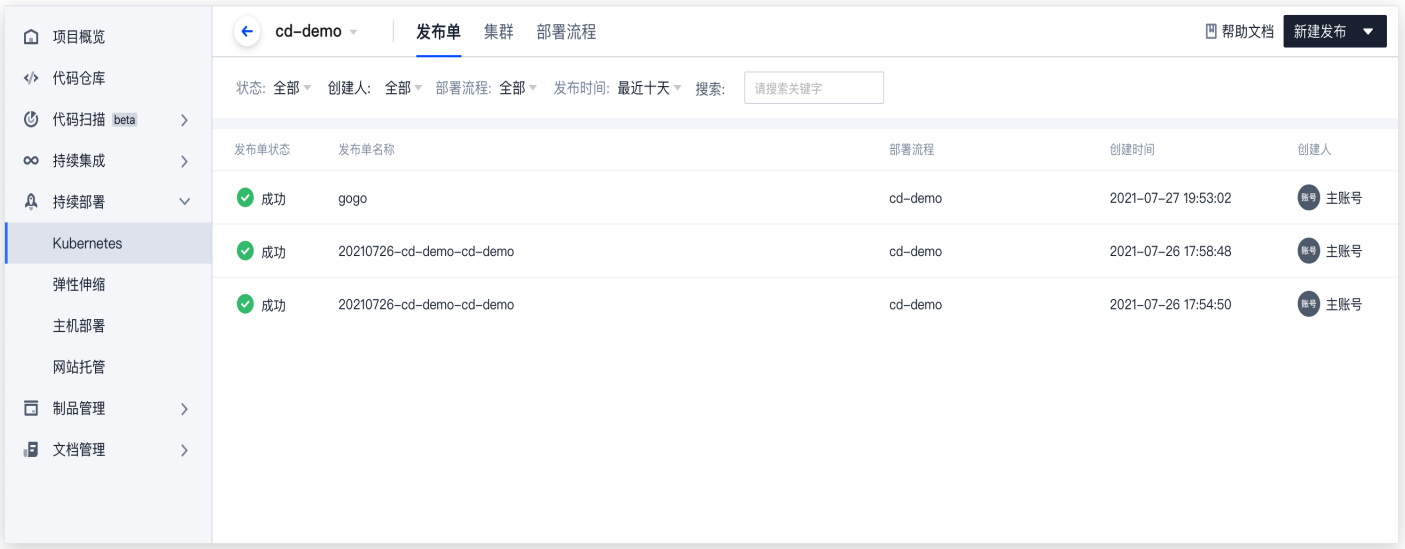
在**基础配置**中点选触发器类型，选择 Docker 仓库触发器。当开发人员更新代码仓库并使用 CI 将镜像打包推送至制品库后，Docker 镜像版本的更新将自动触发部署流程并将应用发布至 Kubernetes（TKE）集群，完成后可以在应用中心查看并确认应用是否发布成功。



手动提交发布单

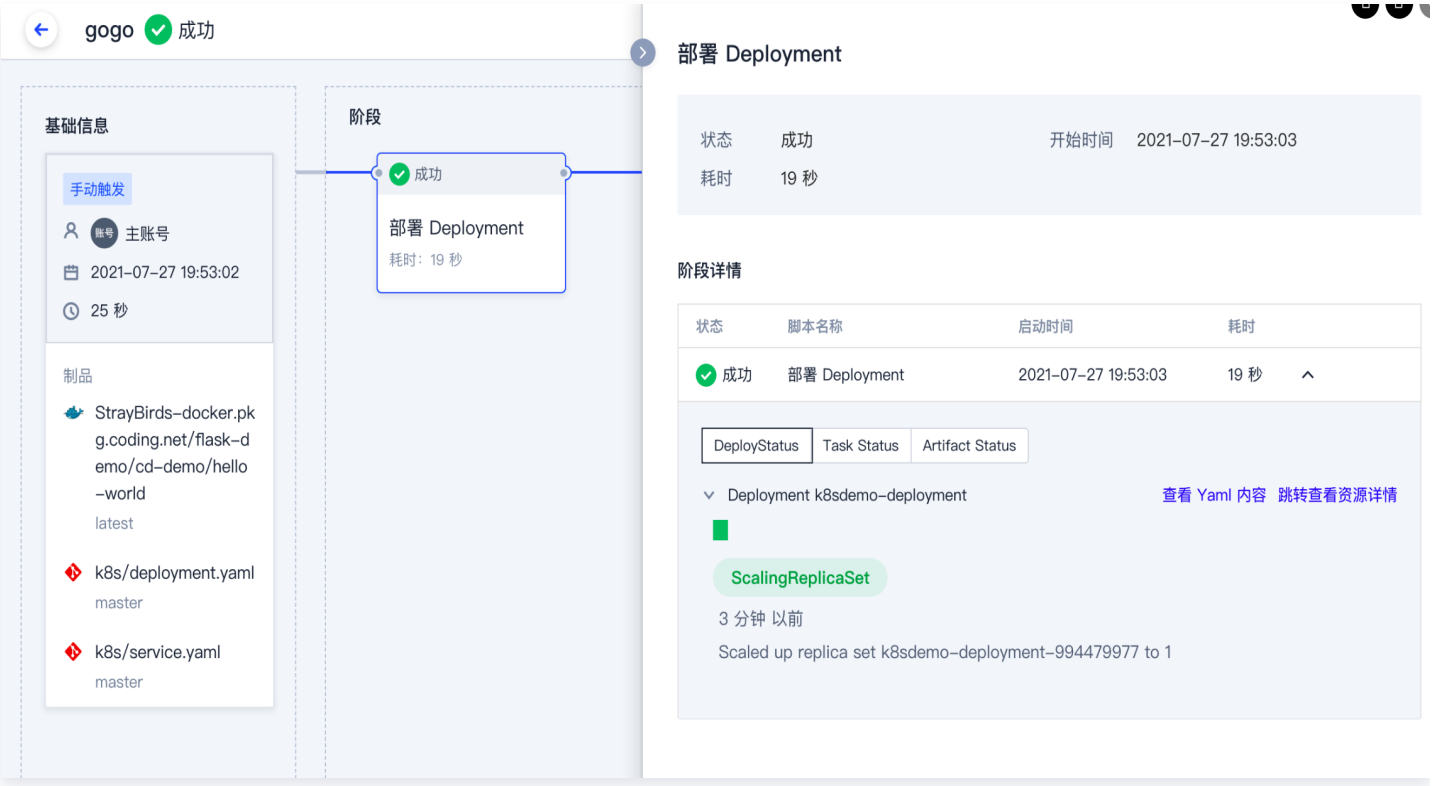
若希望通过手动提交发布单的形式触发部署流程，那么可以将应用（例如本范例的 flaskapp）与项目关联。在应用中心搜索项目名称进行关联：

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

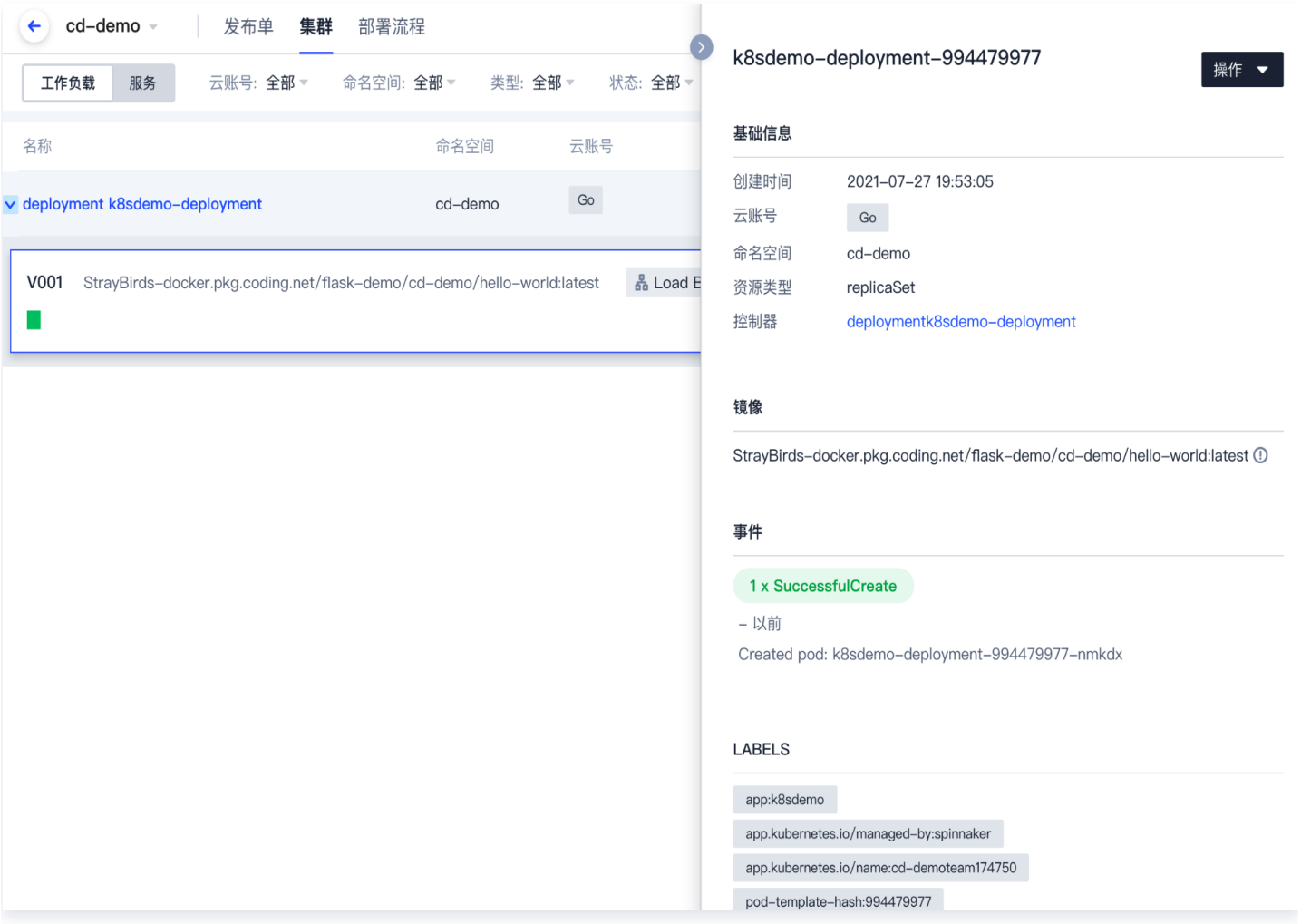


步骤6：发布完成

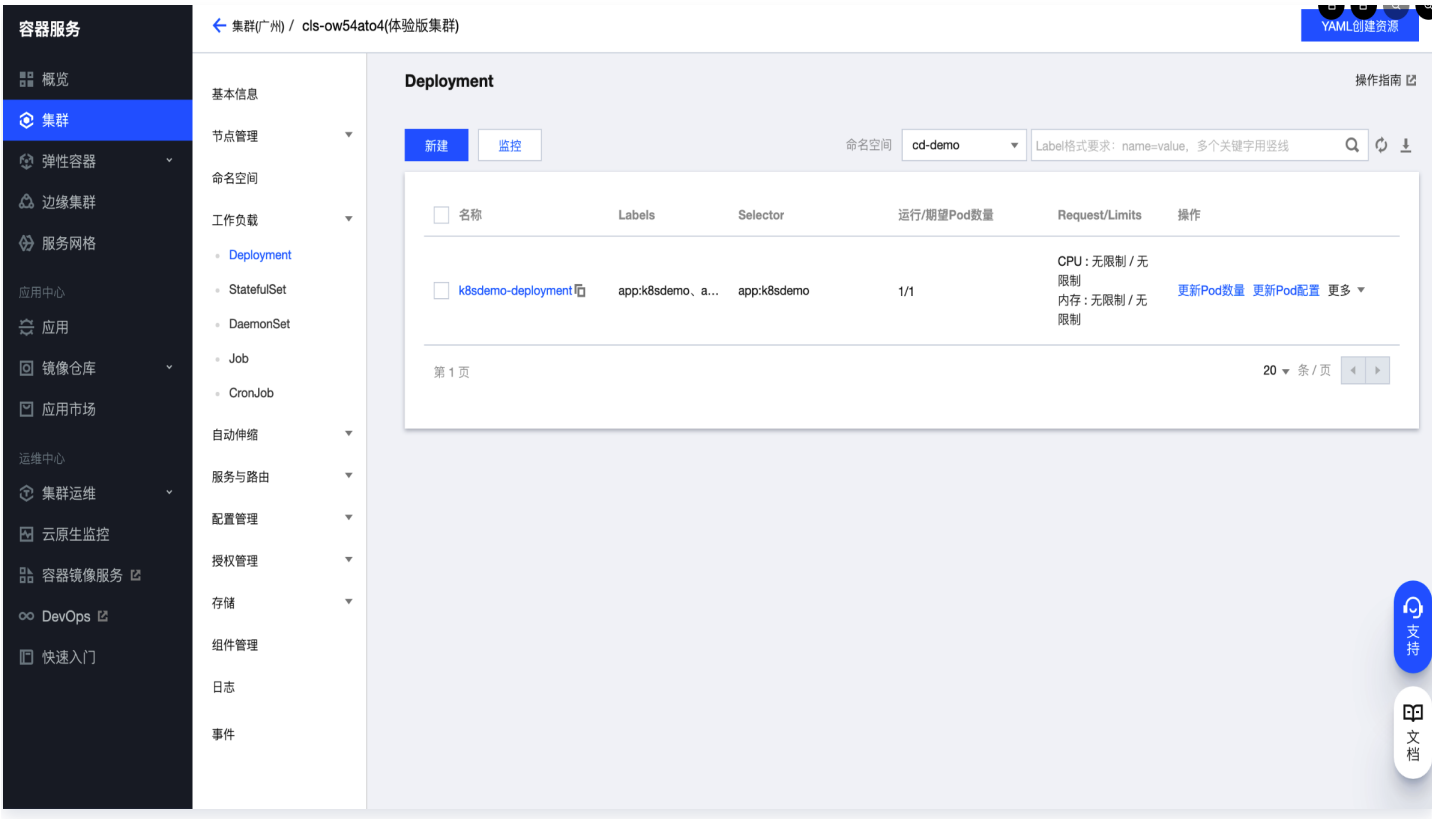
发布成功后，可以查看发布的制品及启动参数及阶段执行详情等信息。



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）。



在腾讯云的容器服务中查看工作负载。

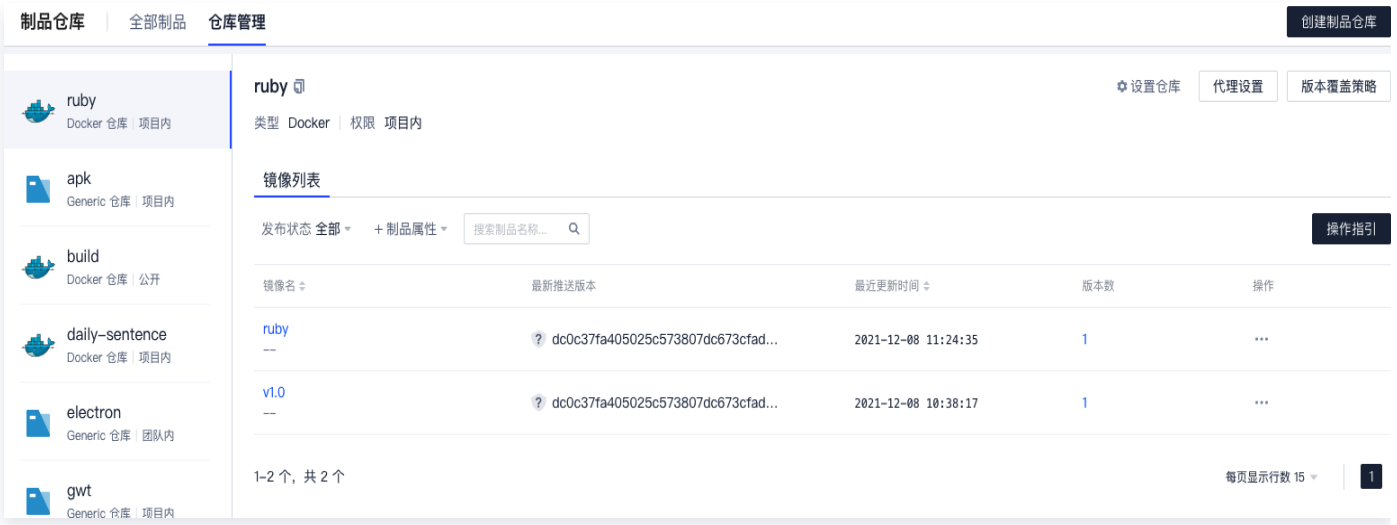


配置自动化构建流程

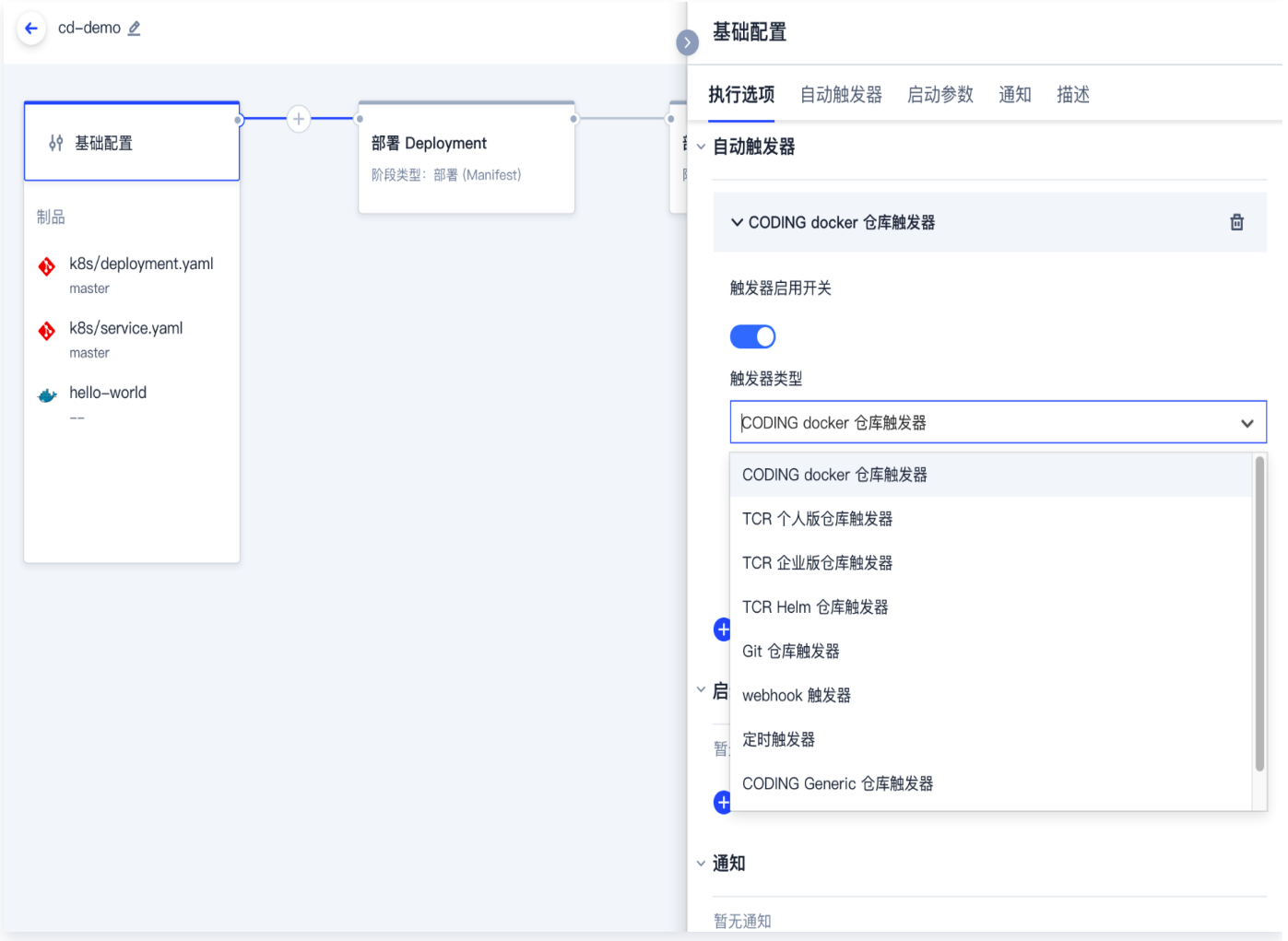
若在后期需要频繁开发项目，能否仅依靠代码推送就能够自动完成发布呢？答案是肯定的。经典的持续交付工作流设计思路如下：

1. 配置持续集成任务，设置监听代码更新后触发。

2. 将项目代码打包成制品后，发布至制品仓库。



3. 持续部署监听到制品版本号更新后，自动发布至集群。



云账号

最近更新时间：2023-09-11 16:05:27

云账号是访问云资源的凭证，只有配置了云账号，持续部署控制台才能实现对云资源的部署管理和基础设施管理。目前支持三种云账号类型：

- 腾讯云 TKE：使用腾讯云容器服务中的集群。
- Kubernetes：支持 Kubeconfig 和 Service Account 两个常用凭据。
- 腾讯云账号：即腾讯云 API 密钥。

单击团队首页左侧的基础设施，进入**云账号**管理页，单击右上方的**绑定云账号**。

腾讯云 TKE

1. 云账号类型选择腾讯云 TKE，按照指引完成与云账号名下的集群绑定。若没有集群请前往 腾讯云 TKE 创建集群。

2. 选择拟部署的集群，单击**确定**后会自动验证该账号名下的集群并完成互联。

Kubernetes

Kubernetes 云账号支持 Kubeconfig 和 Service Account 两种常用凭据。以 Kubeconfig 为例：
登录云计算网页控制台，复制 Kubeconfig，并将 CODING IP 段添加至集群外网访问控制列表（白名单）。

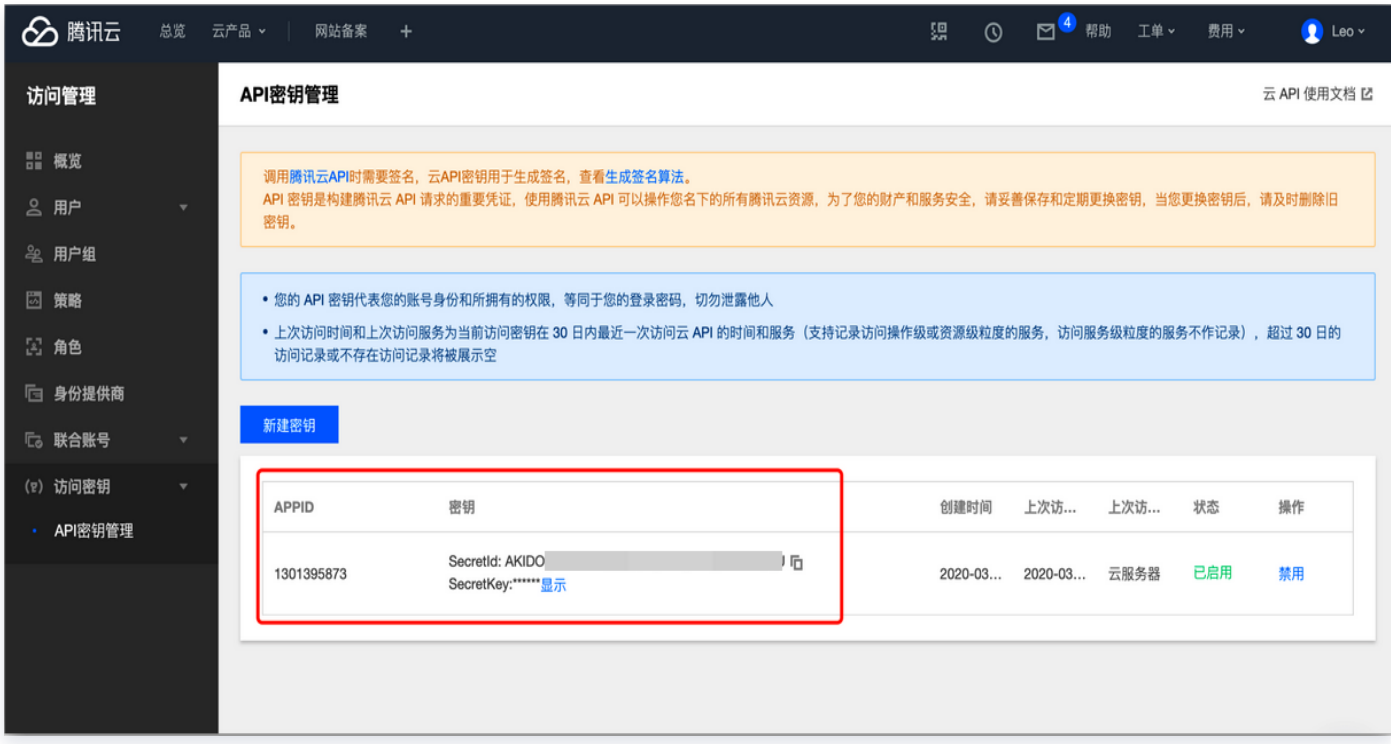
> **说明：**
> CODING 持续部署的公网 IP 段：
> 212.64.105.0/24
> 212.129.144.0/24
> 49.234.127.0/24
> 49.235.224.0/24
> 49.234.65.0/24
> 81.69.101.0/24

将 Kubeconfig 粘贴到云账号中，选择 Cluster Context 完成云账号添加。

腾讯云账号

1. 云账号类型选择腾讯云账号，输入云账号名称，并选择区域。支持多选区域，CODING 持续部署将获得勾选区域的腾讯云资源管理权限。

2. 在腾讯云 访问管理控制台 页面中拷贝 API 密钥信息。



3. 将拷贝的 SecretID 和 SecretKey 粘贴到对应的文本框，单击**确定**完成云账号添加。

部署流程管理

最近更新时间：2024-08-02 15:23:01

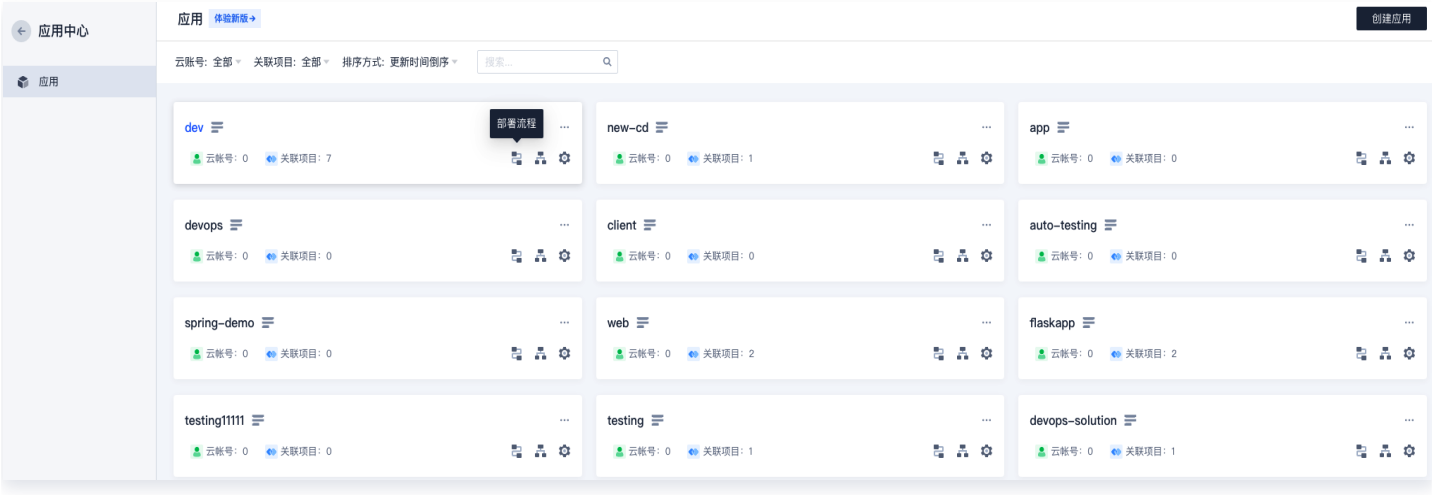
部署流程是实现持续部署最核心的模块。其优势在于支持阶段以任意的顺序组合，这样的能力让部署流程具备出色的灵活性、一致性和可重复性。

- 灵活性：支持串行、并行控制。
- 一致性：支持多种部署策略，回滚能力，确保发布结果符合预期。
- 可重复性：部署流程可重复执行，阶段可被其他部署流程复制使用。

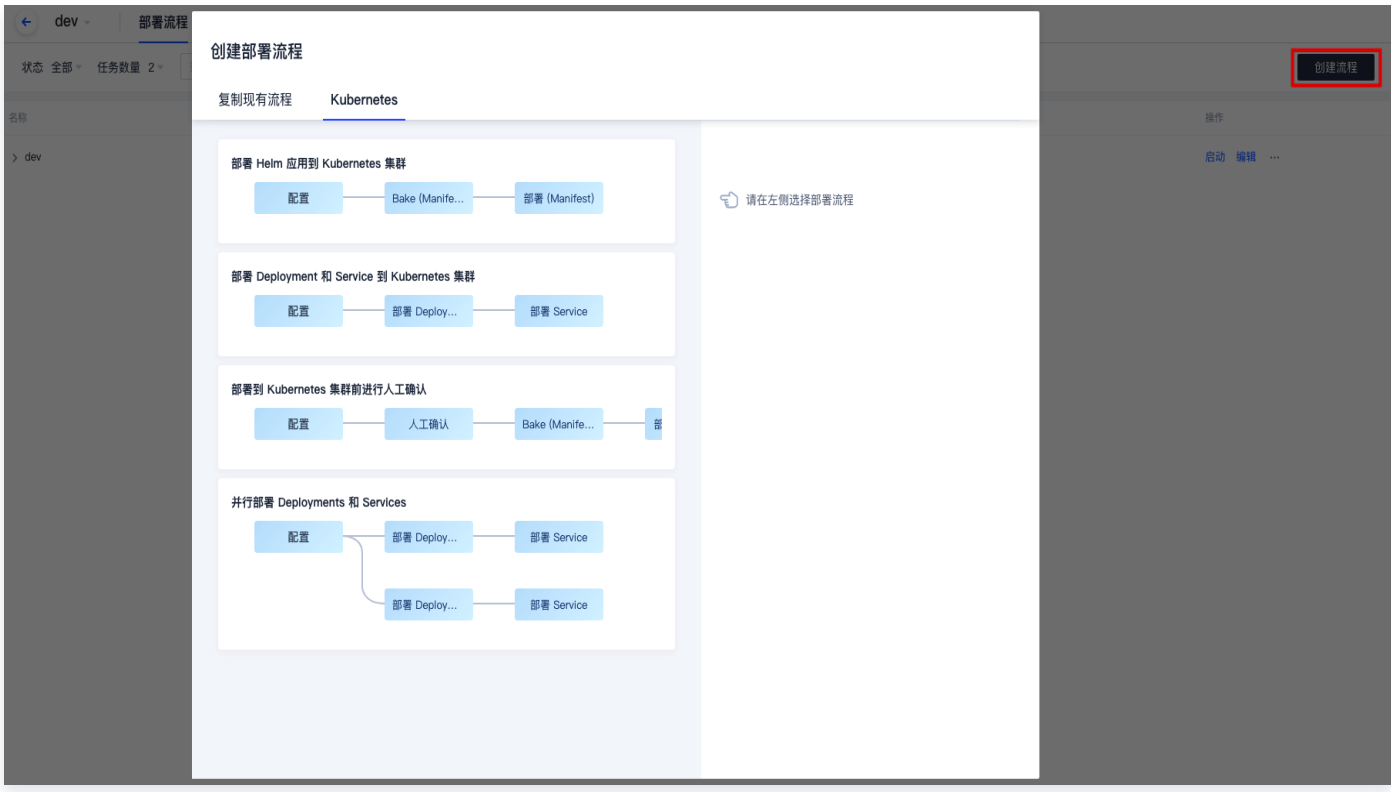
用户可以配置完全自动化的部署流程，也可以在某些阶段加入手工判断条件。此外部署流程支持多种事件的自动化触发，例如 Webhook 触发、由其他部署流程触发等。

新建部署流程

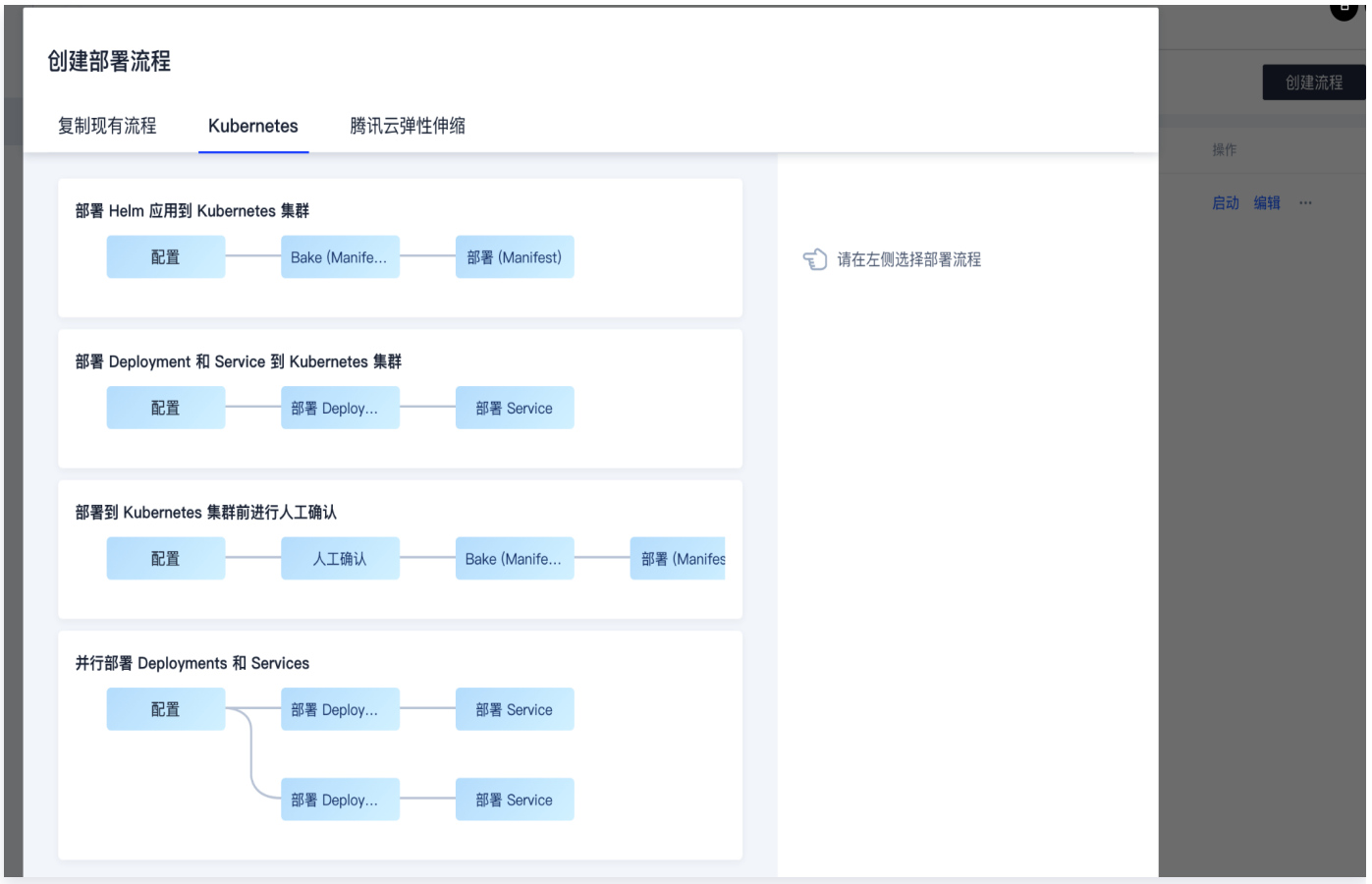
前往应用中心，单击应用卡片右下方的部署流程按钮。



1. 单击右上角的**创建流程**。

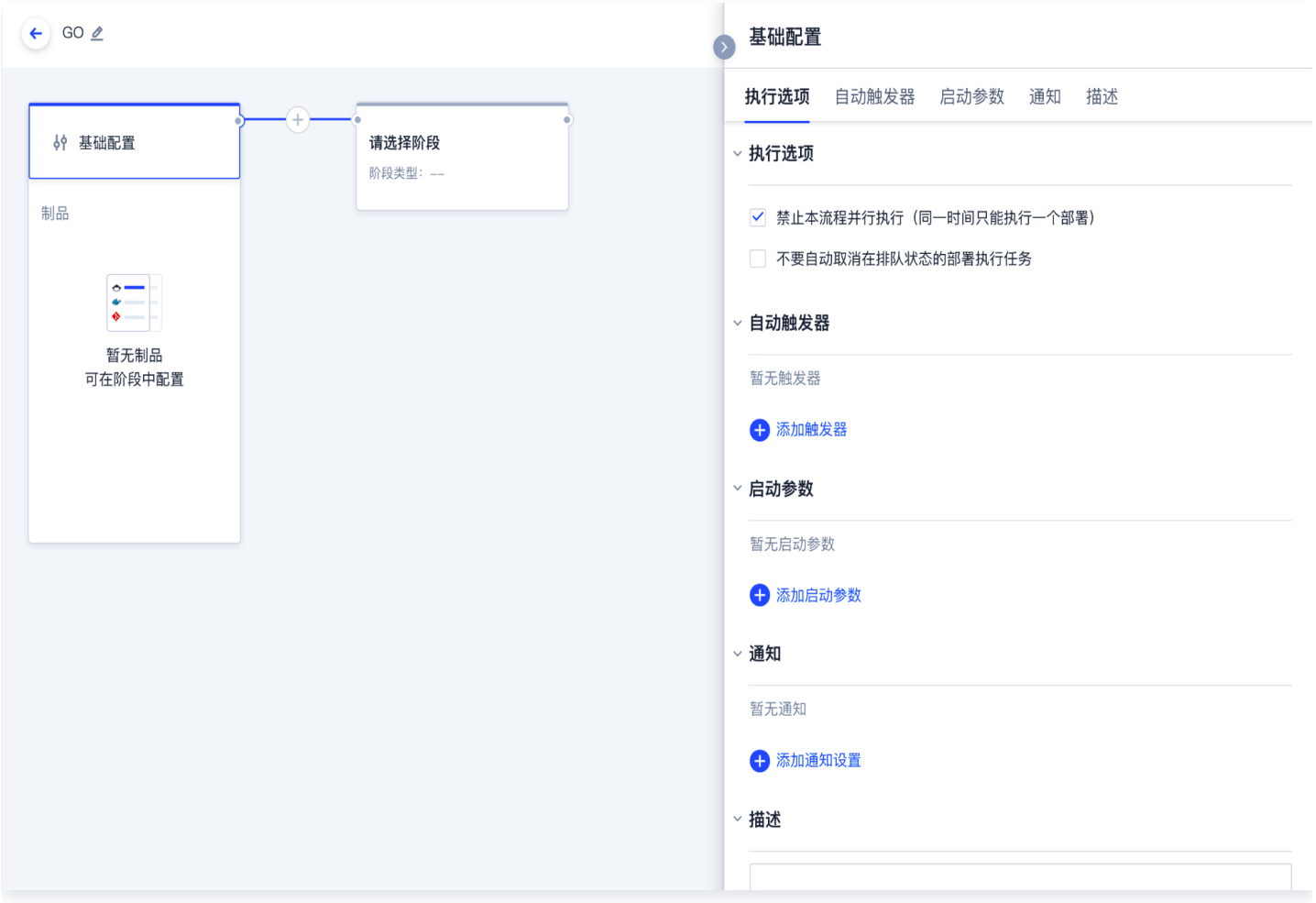


2. 您可以复制在其他应用中创建的流程，或通过空白流程自行创建。



基础配置

应用的基础配置可以理解为构建整体的初始环节，既可以设置触发条件，也可以配置部署流程的通知方式等。

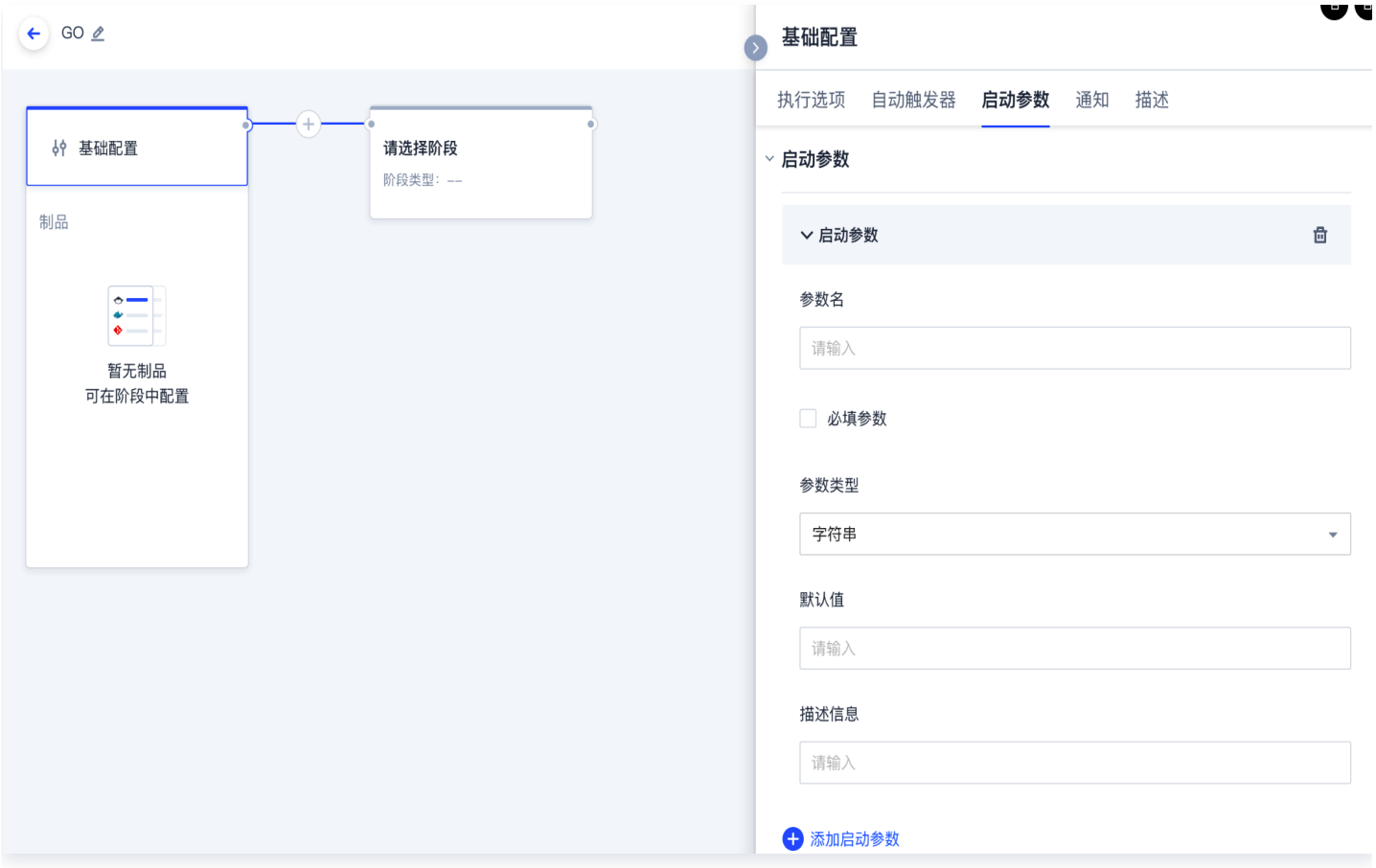


自动触发器

自动触发器支持 CODING Docker 制品仓库、TCR 个人版仓库触发器、Git 仓库触发器等触发条件。

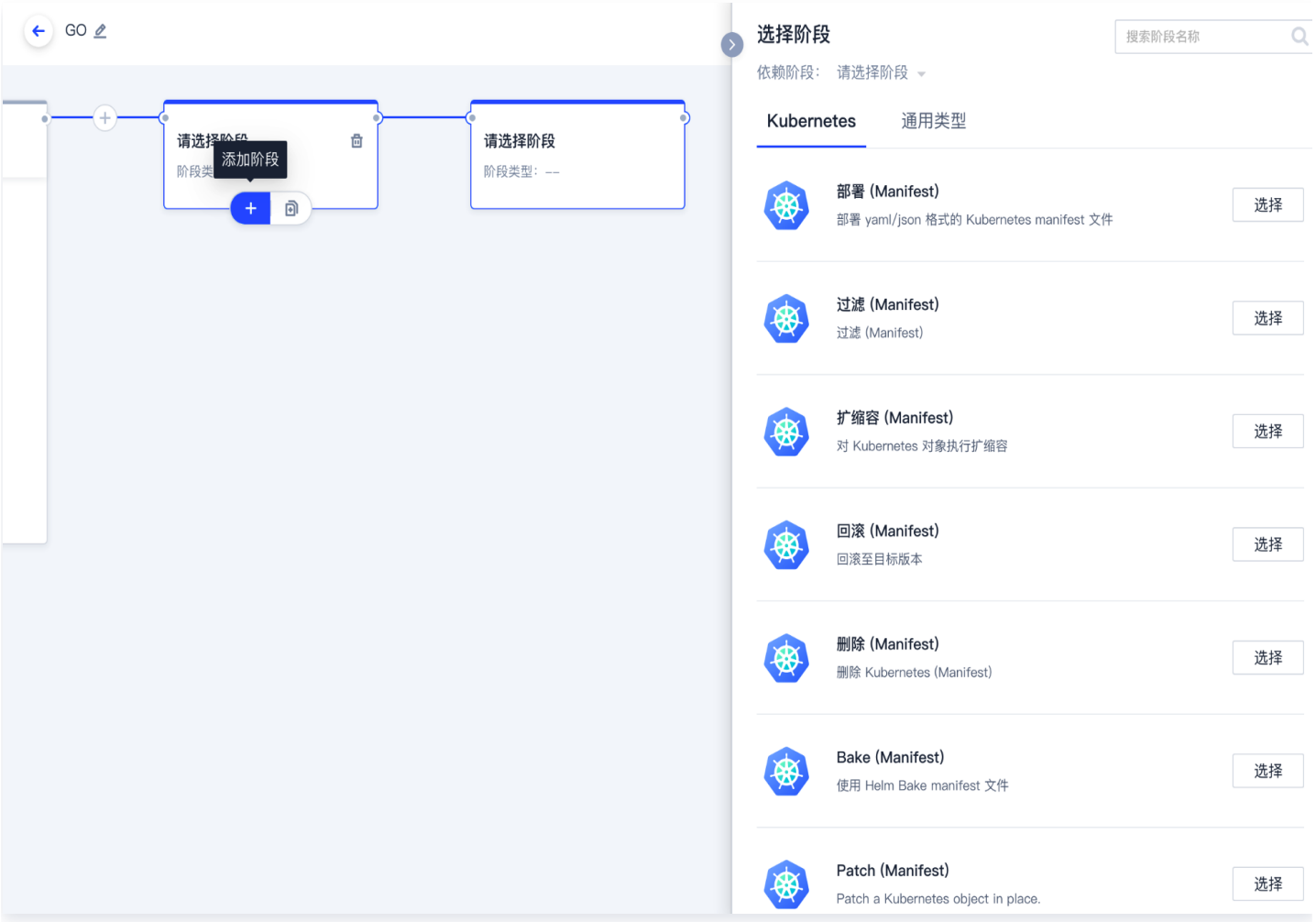
添加部署流程参数

在部署流程配置页面，单击**添加参数**，即可开始填写参数。



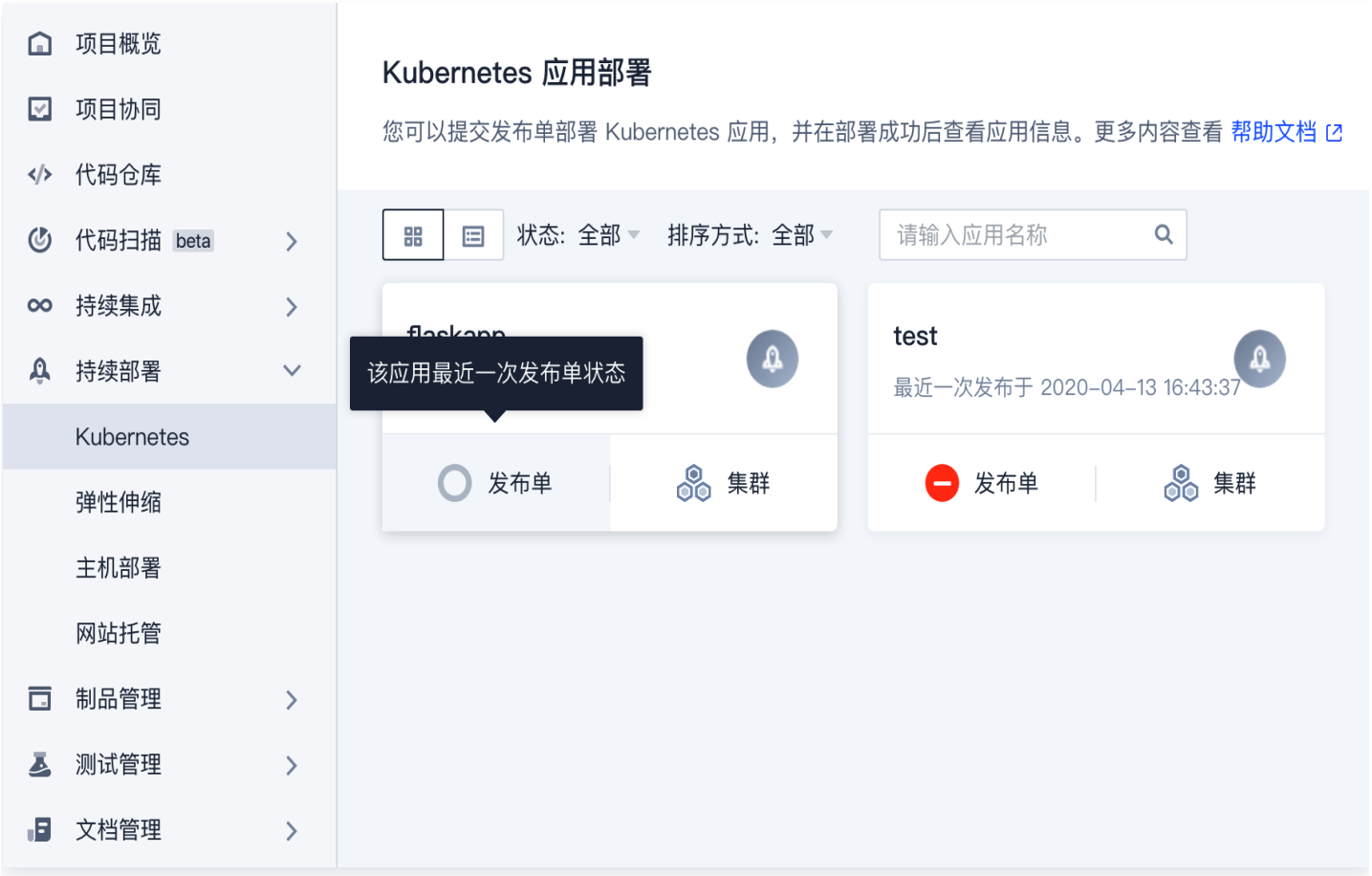
添加阶段

在部署流程配置页面单击加号即可添加新的阶段，右侧列表中支持选择阶段类型。



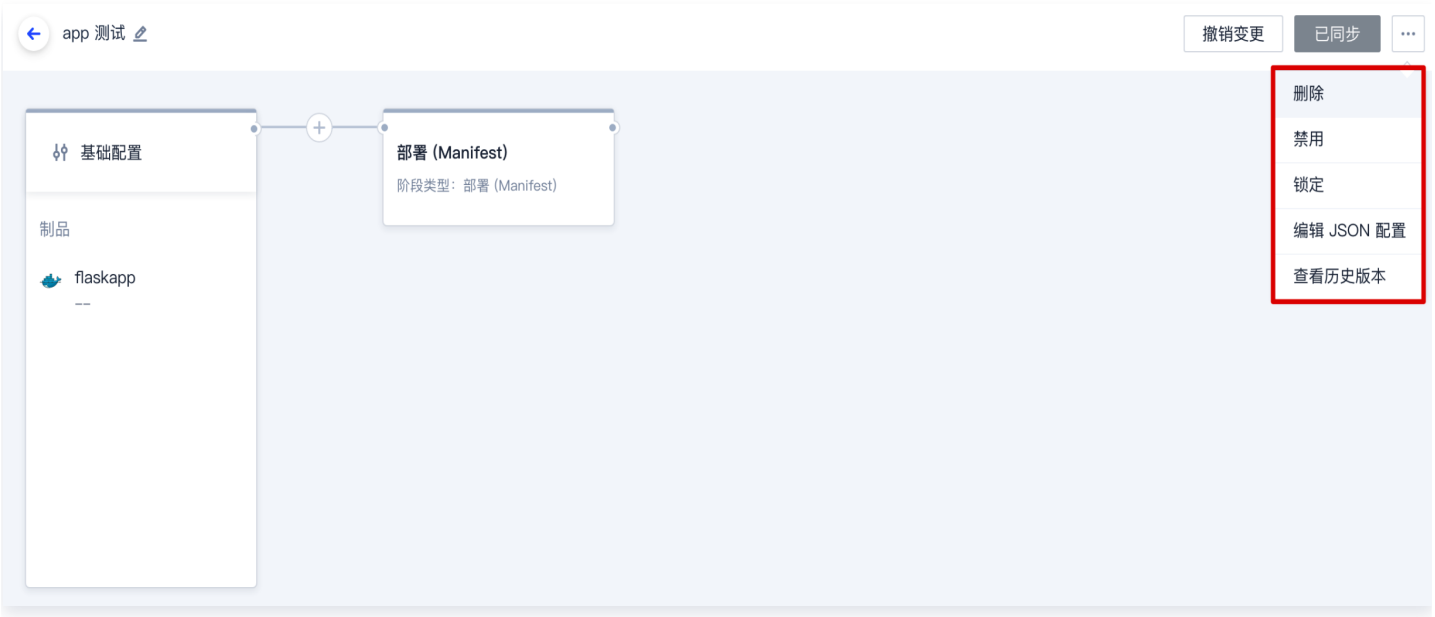
执行部署流程

部署流程配置完成后，您可以通过设置好的触发器依提交自动执行，或在持续部署中提交发布单手动触发部署流程。



部署流程配置

部署流程支持删除、禁用、锁定、查看历史版本与编辑 JSON 配置。



删除部署流程

设置后，将删除此部署流程。

禁用部署流程

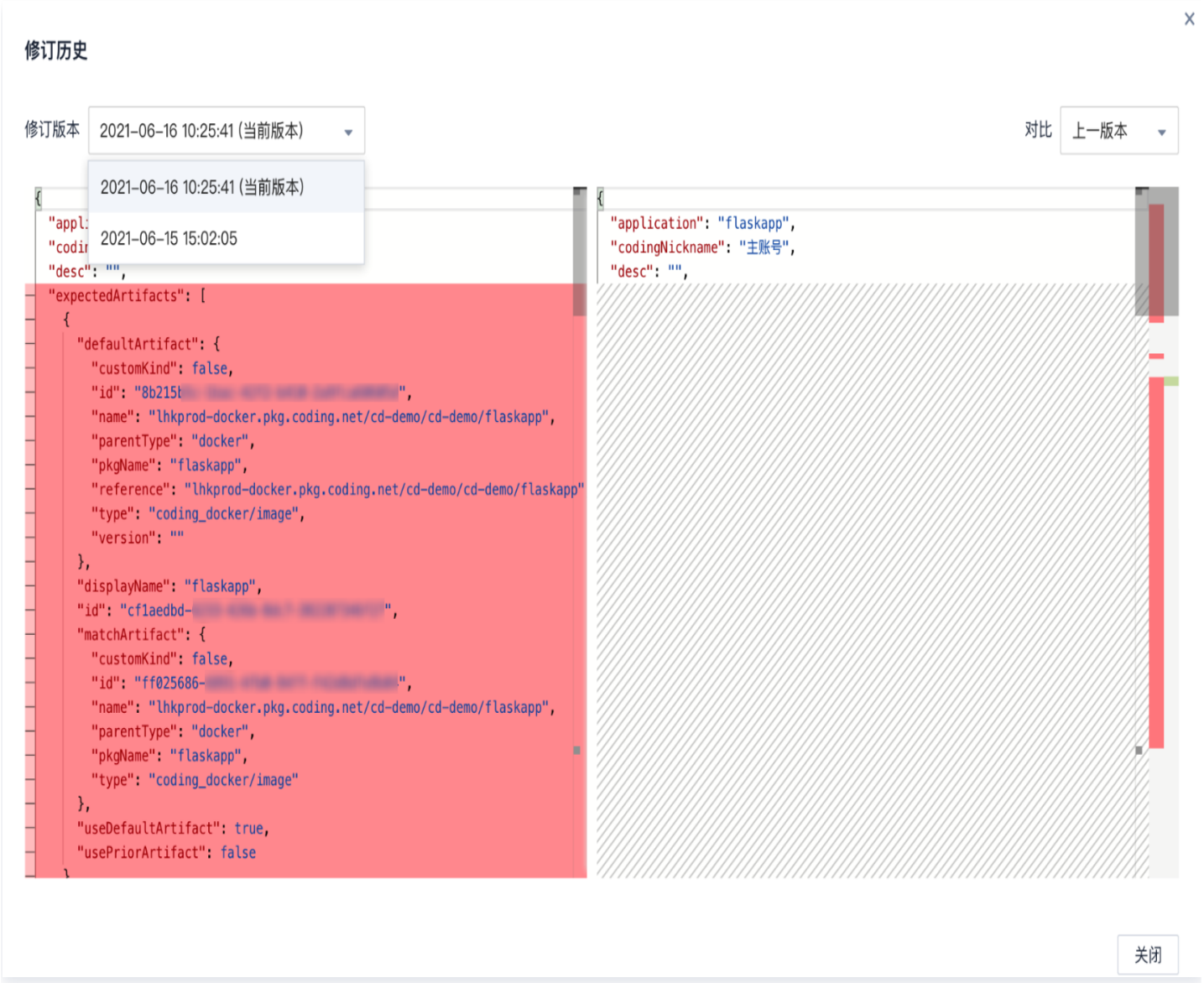
设置后，将禁止任意触发器启动部署流程，包括手动触发。可以选择在团队内整体禁用或仅在项目内禁用。

锁定部署流程

锁定部署流程后，不能在应用中心对部署流程进行任何修改。

查看修订历史

保存新的部署流程配置后，旧版本将会添加到修订历史。您可以在修订历史页对比各版本信息，选择并还原到任意历史版本。

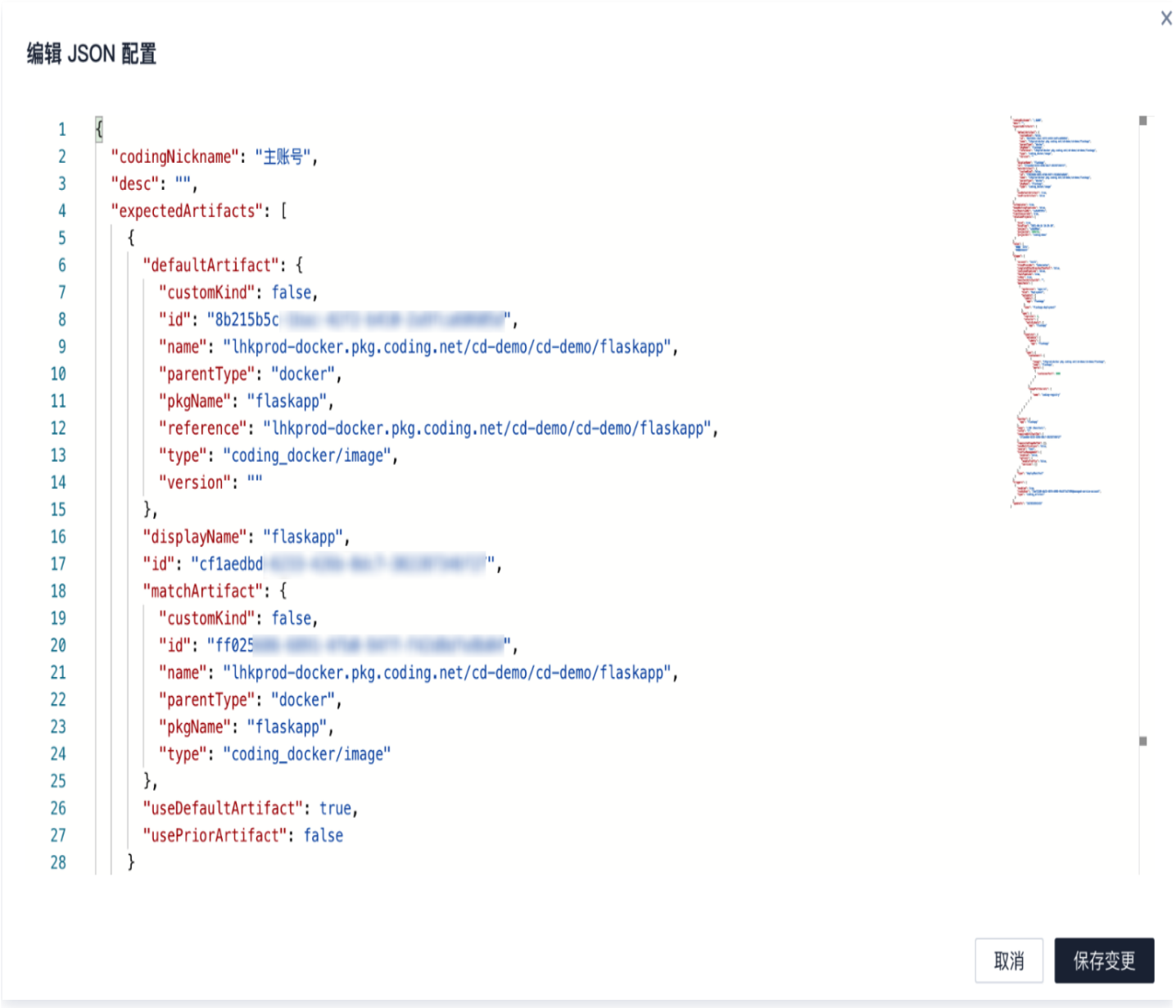


编辑 JSON 配置

在部署控制台中所做的任何更改最终都会以 JSON 格式文件保存，直接编辑部署流程的 JSON 内容可以为部署流程添加新属性或自定义 UI 界面尚未显示的配置项。

> ⚠ **注意：**

此种方式允许用户将在文本框内自由编辑部署流程，但有可能会破坏部署流程的可用性，我们提供了从修订历史中恢复到任意指定版本的能力。



部署方式
自动发布 Docker 制品时触发

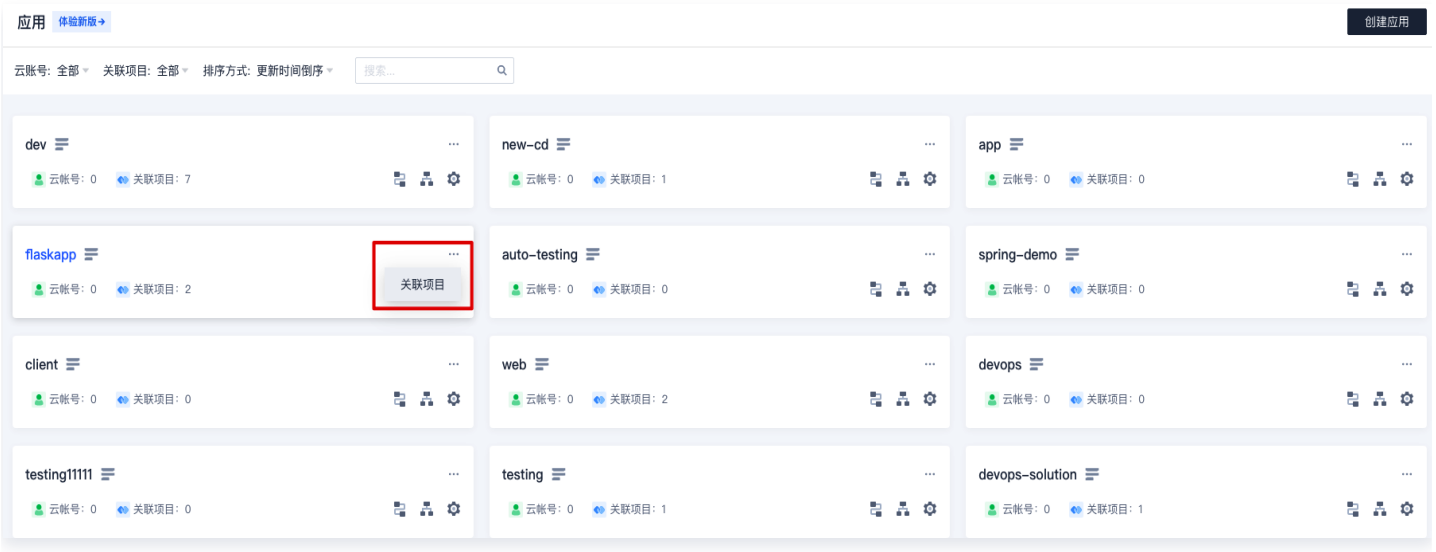
最近更新时间：2023-09-11 16:05:28

CODING 持续部署的一大优势在于能够便捷的集成上下游产品为工作流，下文将演示如何通过三个步骤实现**持续集成任务推送制品 > 制品仓库镜像更新 > 触发部署流程**这一基础的自动化流水线配置。

操作步骤

步骤1：应用与项目关联

部署流程控制台中的**应用**需提前与项目相关联。前往应用中心，单击应用中的**关联项目图标**，选择持续集成配置所在的项目并进行关联。



步骤2：配置持续集成

此步骤使用持续集成将制品推送至制品仓库。您可以通过持续集成计划模板创建，或直接编写 Jenkinsfile 手动增加此阶段。

在持续集成流程中手动增加此阶段：

Jenkinsfile 参考：

```
stage('推送到 CODING Docker 制品库') {
      steps {
        script {
          docker.withRegistry(
```

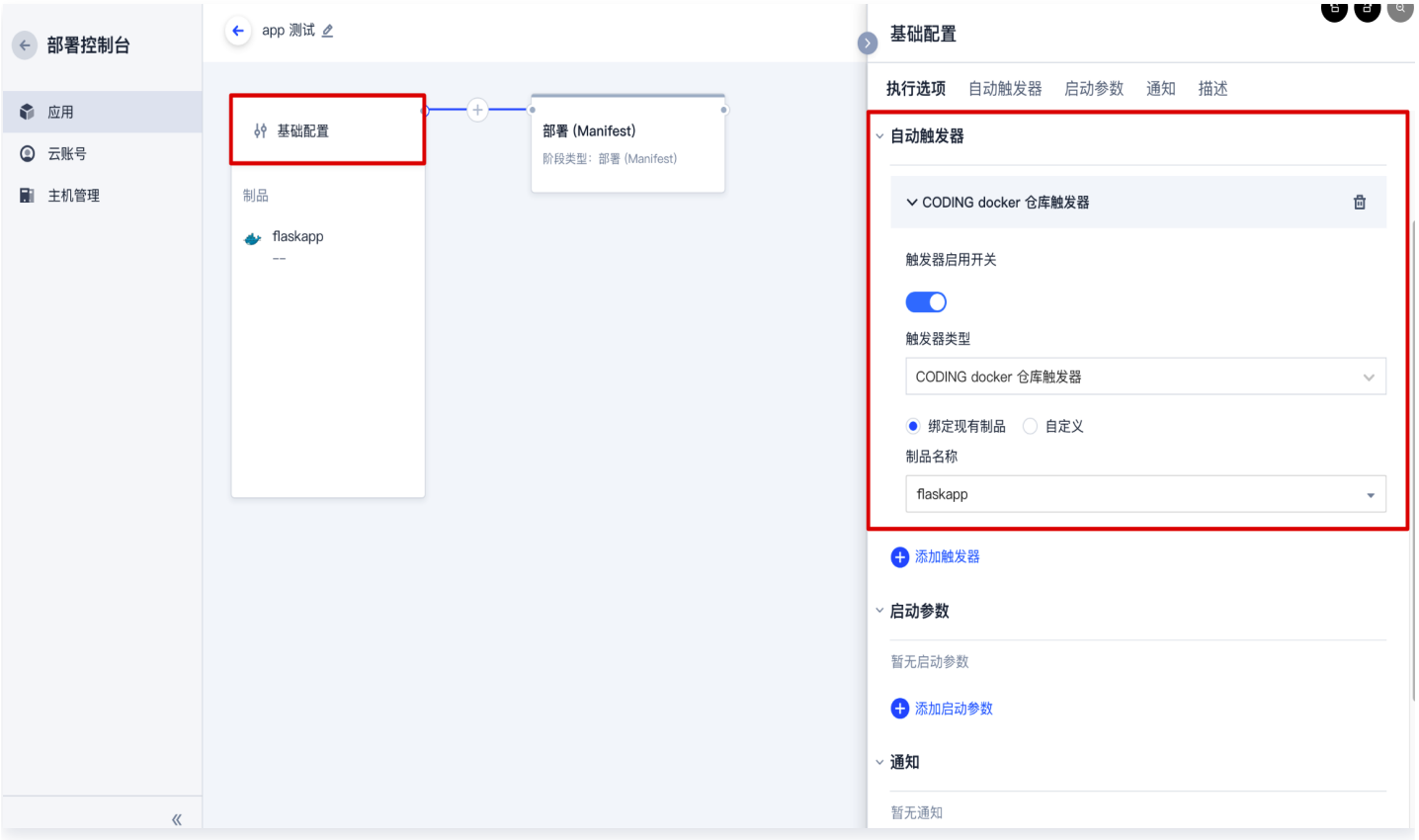
```
"${env.CCI_CURRENT_WEB_PROTOCOL}://${env.CODING_DOCKER_REG_HOST}",
            "${env.CODING_ARTIFACTS_CREDENTIALS_ID}"
          ) {

docker.image("${CODING_DOCKER_IMAGE_NAME}:${env.DOCKER_IMAGE_VERSION}").
push()
          }
        }

      }
```

步骤3：根据制品镜像版本触发

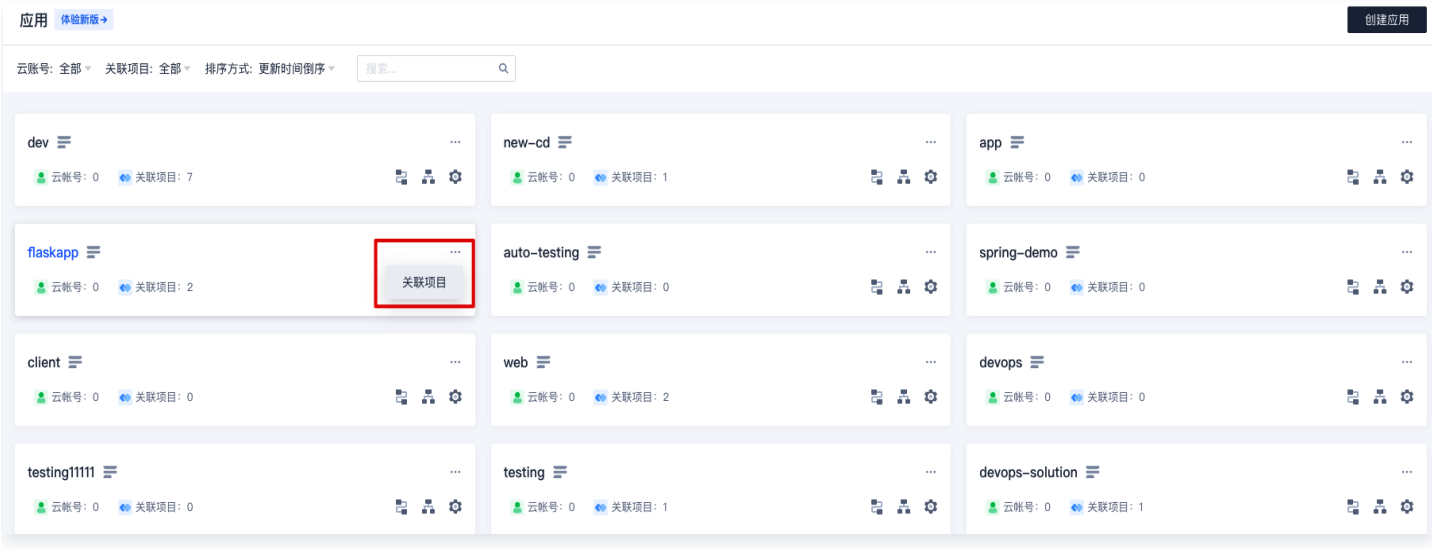
前往持续部署中的应用部署流程，单击**基础配置**中的触发器启用开关。此处选择通过 CODING Docker 制品更新触发，将监听关联项目中制品版本号。若持续集成将制品推送至制品仓库时，将自动触发部署流程；选择**自定义**能够监听其他项目的制品仓库更新情况。
除了通过 CODING Docker 制品更新触发，您还可以通过 Git 仓库或定时器触发此部署流程。



在构建计划中添加部署阶段

最近更新时间：2023-09-11 16:05:28

在持续集成中触发部署时请提前前往应用中心，将应用与项目关联。



本文给出了两种设置方法，您可以按照需求有选择性阅读。

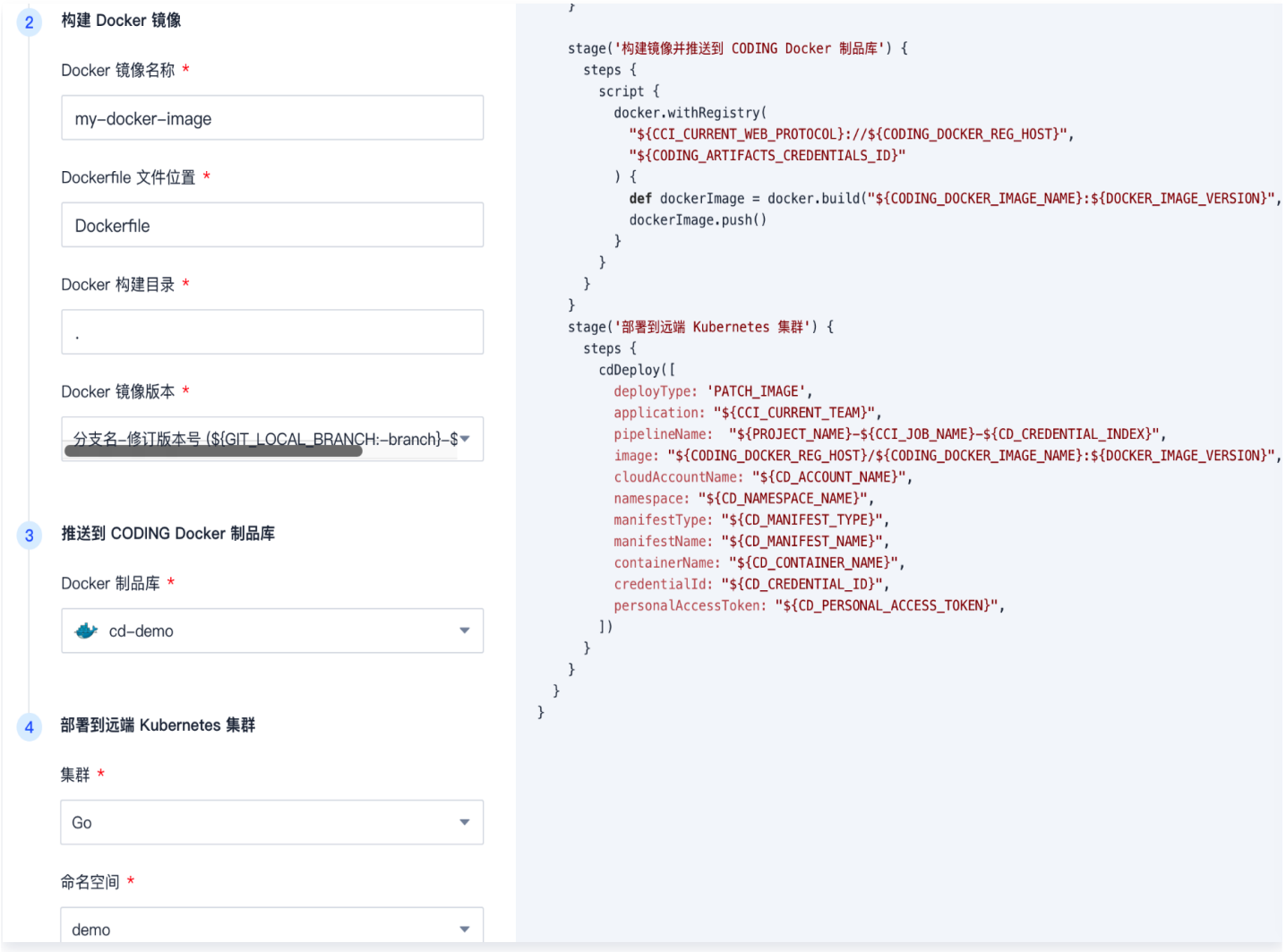
- 直接使用构建计划模板
- 在已有构建计划中添加部署阶段

使用构建计划模板

请在应用中心关联涉及到相关集群的云账号，详情请参见 云账号 。

单击项目内左侧产品栏**持续集成**，右上角创建构建计划，选择**部署**分类下的**推送到 Kubernetes** 模板。

按照模板提示选择相应的制品仓库、远端集群地址等信息，完成后勾选创建后触发构建。



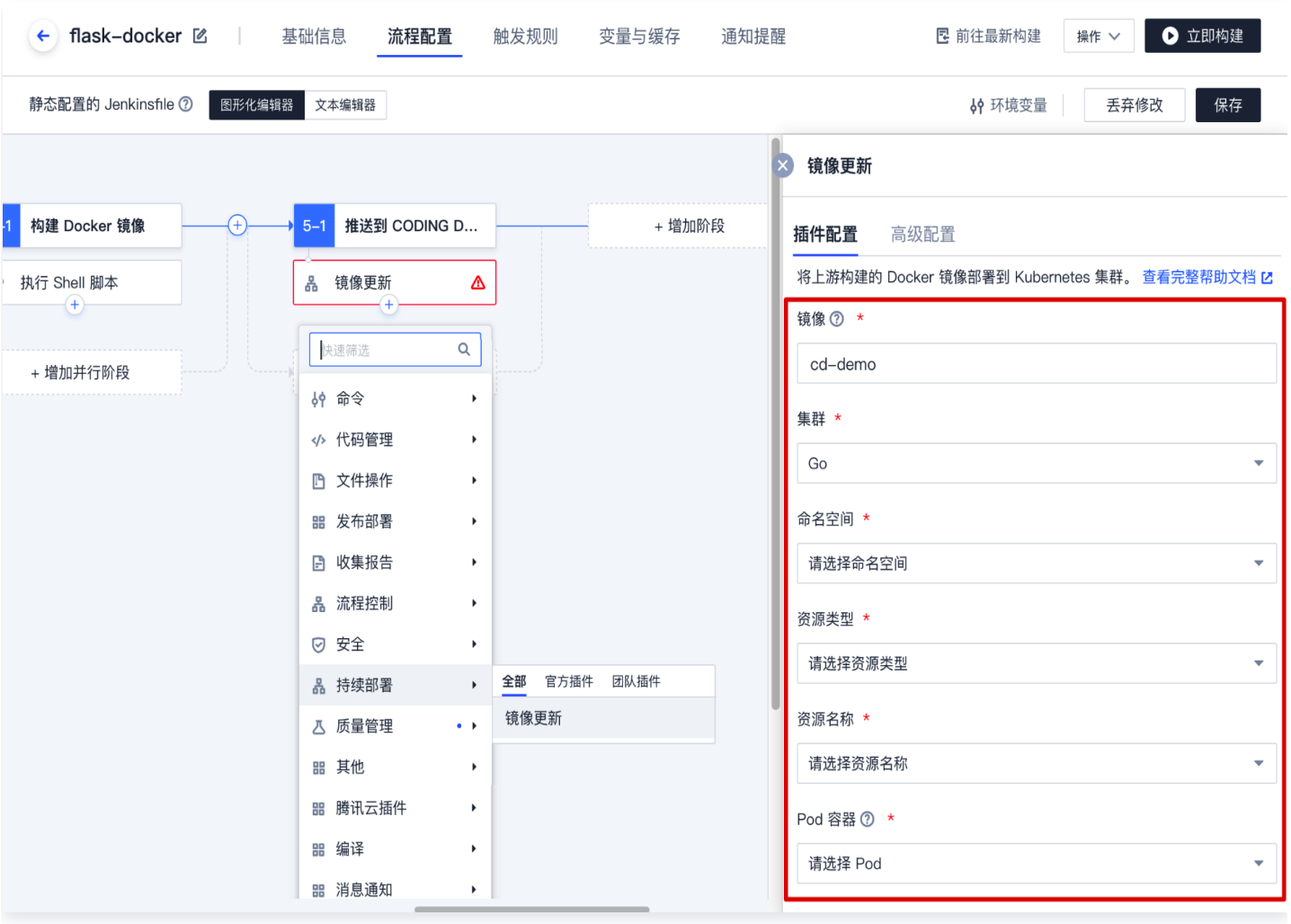
设置完成后，运行持续构建计划即可完成自动发布。

添加部署阶段

在此方法中您可以在**构建计划 > 流程配置**中使用编辑器或填写命令添加部署阶段。

图形化编辑器

在已有构建计划设置中添加**部署**阶段，填写镜像地址、集群与命名空间等关键信息。



Jenkinsfile 参考：

```
stage('部署到远端 Kubernetes 集群') {
      steps {
        cdDeploy([
          deployType: 'PATCH_IMAGE',
          application: "${CCI_CURRENT_TEAM}",
          pipelineName:
"${PROJECT_NAME}-${CCI_JOB_NAME}-${CD_CREDENTIAL_INDEX}",
          image:
"${CODING_DOCKER_REG_HOST}/${CODING_DOCKER_IMAGE_NAME}:${DOCKER_IMAGE_VE
RSION}",
          cloudAccountName: "${CD_ACCOUNT_NAME}",
          namespace: "${CD_NAMESPACE_NAME}",
          manifestType: "${CD_MANIFEST_TYPE}",
          manifestName: "${CD_MANIFEST_NAME}",
          containerName: "${CD_CONTAINER_NAME}",
```

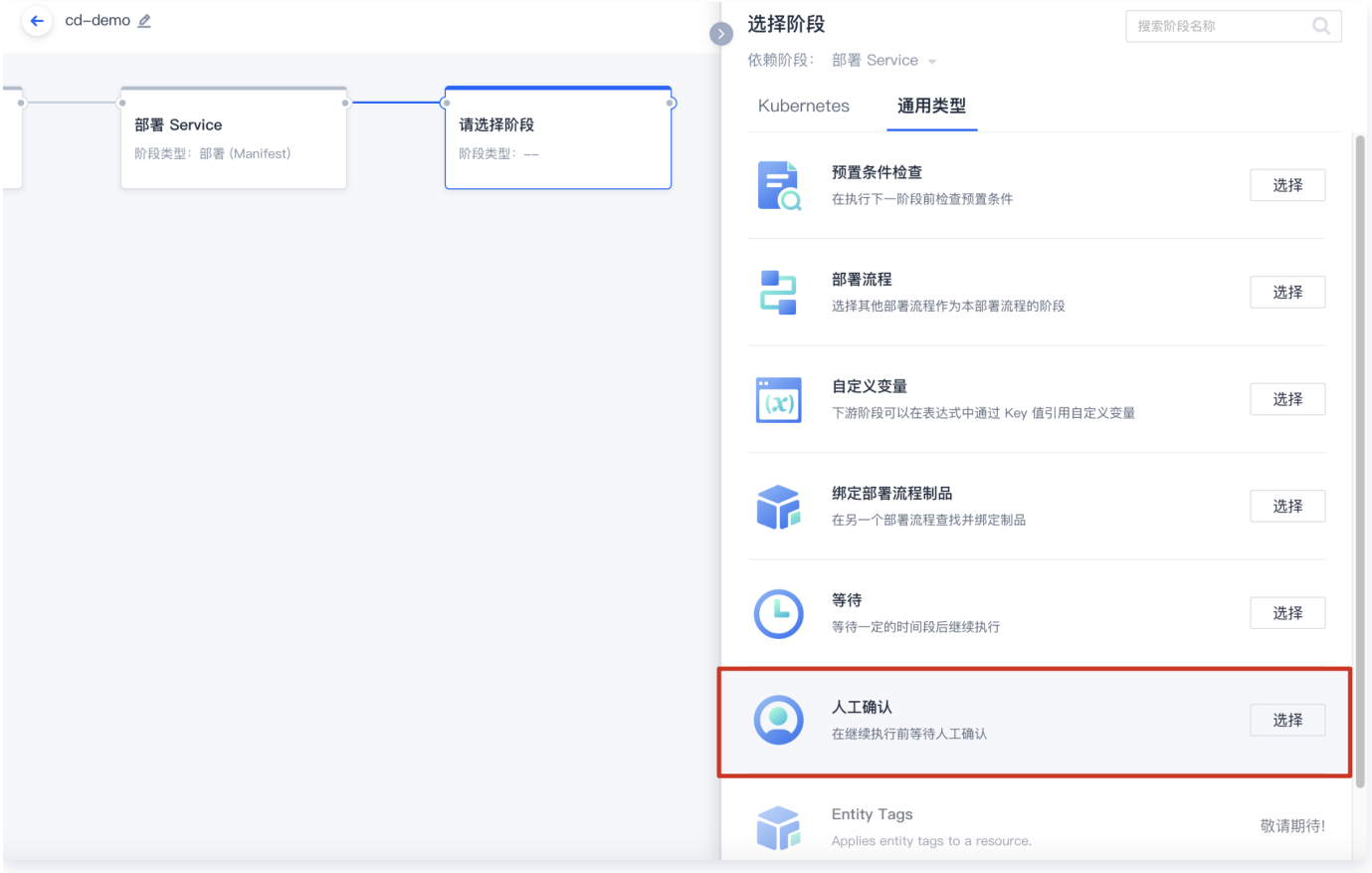
```
            credentialId: "${CD_CREDENTIAL_ID}",
            personalAccessToken: "${CD_PERSONAL_ACCESS_TOKEN}",
          ])
        }
      }
```

手动提交发布单

最近更新时间：2025-04-14 17:03:52

新建发布单

在应用的部署流程中添加人工确认步骤，确保通过发布单发布时是经二次确认的，通过权限控制把控发布质量。



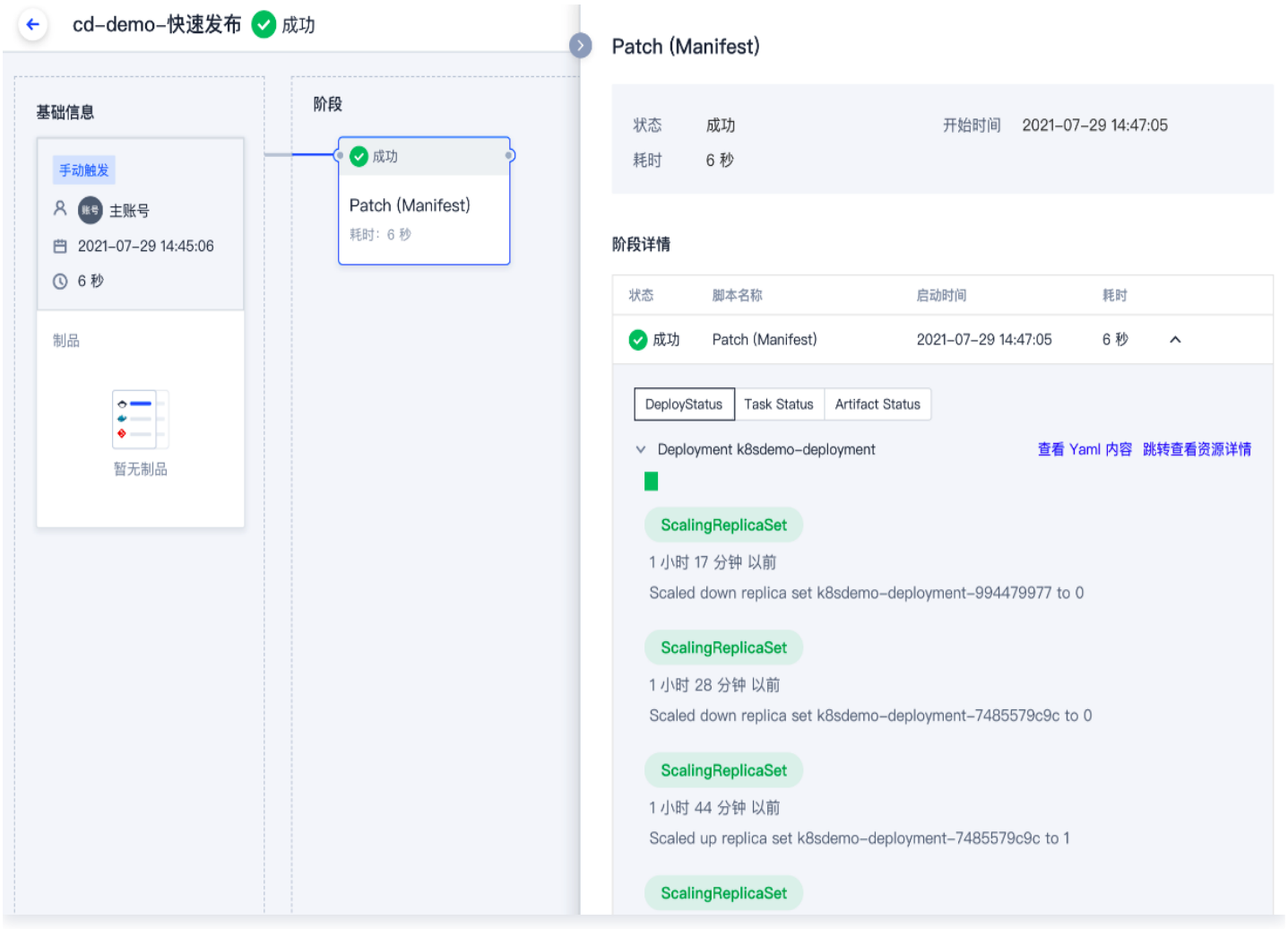
1. 要使用发布单功能，必须将 应用与项目关联 。
2. 关联完成后，进入项目中**持续部署**找到该应用，单击**发布单**进入发布单管理页面。

3. 单击**新建发布单**后，可以运行已有应用及部署流程。

快速发布

若不希望在团队设置复杂的权限限制，需要直接体验持续部署功能，可以使用**快速发布**功能。您无需在控制台中配置部署流程即可将镜像发布至集群中，适用于部署流程更加灵活复杂的场景，例如说临时镜像变更等突发场景，无需快速将制品发布至集群中。

发布完成后可以在持续部署中查看发布详情。


cd-demo-快速发布
成功
基础信息
手动触发
主账号
2021-07-29 14:45:06
6 秒
制品
暂无制品
阶段
成功
Patch (Manifest)
耗时：6 秒
Patch (Manifest)
状态 成功
开始时间 2021-07-29 14:47:05
耗时 6 秒
阶段详情
状态 脚本名称 启动时间 耗时
成功 Patch (Manifest) 2021-07-29 14:47:05 6 秒
DeployStatus
Task Status
Artifact Status
Deployment k8sdemo-deployment
查看 Yaml 内容
跳转查看资源详情
ScalingReplicaSet
1 小时 17 分钟 以前
Scaled down replica set k8sdemo-deployment-994479977 to 0
ScalingReplicaSet
1 小时 28 分钟 以前
Scaled down replica set k8sdemo-deployment-7485579c9c to 0
ScalingReplicaSet
1 小时 44 分钟 以前
Scaled up replica set k8sdemo-deployment-7485579c9c to 1
ScalingReplicaSet