

软件开发生产线

快速入门

文档版本 01
发布日期 2025-01-20



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 入门指引.....	1
2 使用 CodeArts 快速搭建基于 ECS 部署的代码开发流水线.....	3
3 使用 CodeArts 快速搭建基于 CCE 部署的代码开发流水线.....	11

1 入门指引

表 1-1 服务快速入门指引

服务	入门指引
整体流程类	<ul style="list-style-type: none">使用CodeArts快速搭建基于ECS部署的代码开发流水线使用CodeArts快速搭建基于CCE部署的代码开发流水线
需求管理	<ul style="list-style-type: none">创建Scrum项目并新建工作项创建IPD系统设备类项目并新建工作项
软件建模	软件建模快速入门
代码托管	<ul style="list-style-type: none">完成一次Scrum项目下的JAVA代码开发管理员配置CodeArts Repo代码仓库的策略设置
流水线	通过流水线生成软件包并部署到主机
代码检查	快速检查CodeArts Repo代码仓库中的代码质量
编译构建	<ul style="list-style-type: none">使用编译构建服务Ant构建并上传软件包至软件发布库使用编译构建服务的Cmake构建并上传软件包至软件发布库使用编译构建服务的Maven构建上传软件包和推送镜像
制品仓库	<ul style="list-style-type: none">上传软件包到CodeArts Artifact软件发布库上传私有组件到CodeArts Artifact中的Maven私有依赖库
部署	通过部署服务创建Tomcat应用并部署到ECS
测试计划	快速执行一次测试计划(CodeArts TestPlan)并查看测试报告
性能测试	性能测试快速入门
漏洞管理服务	如何使用漏洞管理服务
CodeArts IDE Online	<ul style="list-style-type: none">CodeArts IDE Online入门流程5分钟创建并启动IDE实例
CodeArts IDE	CodeArts IDE快速入门

服务	入门指引
Codearts 盘古助手	使用CodeArts 盘古助手生成代码及对应单元测试
开源镜像站	快速上手开源镜像站
开源治理服务	快速创建一个二进制成分分析任务
联接	<ul style="list-style-type: none">• 使用空模板快速开始• 使用组合应用模板快速开始

2 使用 CodeArts 快速搭建基于 ECS 部署的代码开发流水线

本文基于CodeArts内置代码仓库，介绍如何使用CodeArts完成项目的开发、构建与部署，实现持续交付。

本文采用的部署方式为ECS部署，适用于传统软件包部署场景。

如果您希望使用容器化部署方法，请参考[使用CodeArts快速搭建基于CCE部署的代码开发流水线](#)。

准备工作

1. 已注册华为云并实名认证。如果您还没有华为账号，请参考以下步骤创建。
 - a. [注册华为账号并开通华为云](#)。
 - b. 完成[实名认证](#)。
2. 已[购买弹性云服务器](#)，购买时的必要配置可参考下表，表中未列出的配置可根据实际情况选择。完成购买后，参考[配置安全组规则](#)添加端口22及8080的入方向规则。

表 2-1 弹性云服务器配置

配置分类	配置项	配置建议
基础配置	计费模式	选择“按需计费”。
实例	CPU架构	选择“x86计算”。
	规格	选择2vCPUs 4GiB或以上规格。
操作系统	镜像	选择“公共镜像 > CentOS > CentOS 7.6 64bit(10GiB)”。
公网访问	弹性公网IP	选择“现在购买”。
	公网带宽	选择“按宽带计费”。

配置分类	配置项	配置建议
云服务器管理	登录凭证	选择“密码”。
	密码	输入自定义密码。

开通 CodeArts 体验版

- 步骤1 进入[购买CodeArts套餐页面](#)。
 - 步骤2 选择“体验版”，勾选同意声明，单击“立即开通”。
 - 步骤3 开通成功，返回“软件开发生产线”页面，列表中显示已开通套餐记录。
- 结束

新建项目

项目是使用CodeArts各服务的基础，创建项目后才能完成后续操作。

- 步骤1 在CodeArts控制台单击“立即使用”。
 - 步骤2 单击“新建项目”，选择“Scrum”，输入项目名称“Demo”，单击“确定”。
- 结束

新建代码仓库

代码仓库用于项目代码的版本管理，本文使用服务内置的模板“Java Web Demo”创建代码仓库。

- 步骤1 单击导航栏“代码 > 代码托管”，进入代码托管服务。
 - 步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。
 - 步骤3 选择模板“Java Web Demo”，单击“下一步”。
 - 步骤4 输入代码仓库名称“Web-Demo”，单击“确定”。
- 结束

检查代码

通过代码检查服务，可以对代码进行静态检查，管控代码质量。

- 步骤1 单击导航栏“代码 > 代码检查”，进入代码检查服务。页面中显示自动创建的任务“Web-Demo-check”。
 - 步骤2 单击任务所在行的▶启动任务。
 - 步骤3 当页面中显示时，表示任务执行成功。单击任务名称，进入任务的“概览”页面查看检查结果。
- 如果任务执行失败，请根据页面弹出报错提示排查修改。
- 结束

构建并归档软件包

通过编译构建服务，可将软件的源代码编译成目标文件，并把配置文件和资源文件等打包并归档到软件发布库中。

- 步骤1** 单击导航栏“持续交付 > 编译构建”，进入编译构建服务。页面中显示自动创建的任务“Web-Demo-build”。
- 步骤2** 单击任务所在行的▶启动任务。如果出现弹窗，请确认参数设置准确后，单击“确定”。
- 步骤3** 当页面中显示✅时，表示任务执行成功。单击任务名称，进入构建历史页面，在列表中找到最新一次构建的构建编号，记录该编号。

如果构建失败，请根据失败步骤信息与日志中的报错信息排查。

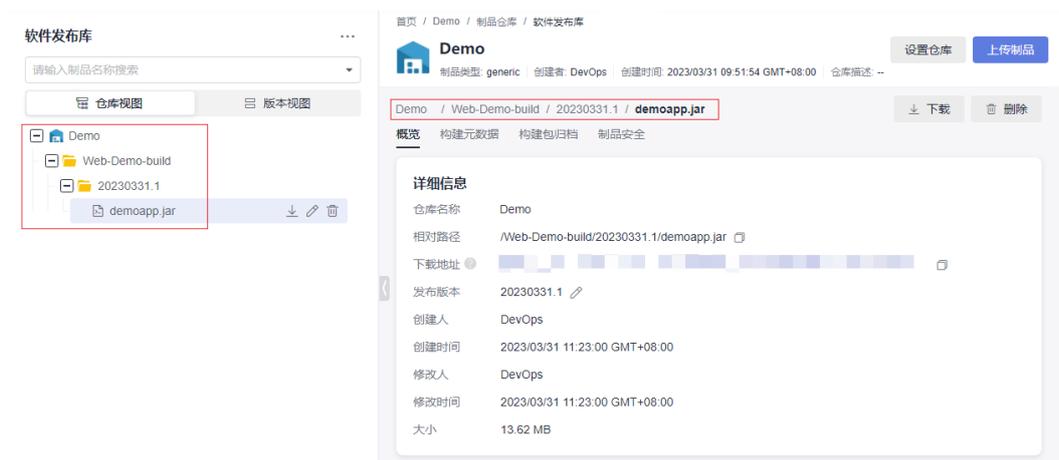
图 2-1 构建编号



- 步骤4** 单击导航栏“制品仓库 > 软件发布库”，进入软件发布库。

在与项目名称同名的仓库中，依次进入与构建任务同名的文件夹、与构建编号同名的文件夹，即可找到生成的软件包“demoapp.jar”。

图 2-2 查看软件包



---结束

部署构建包

通过部署服务，可将软件发布库中的软件包部署到虚拟机，并启动运行。

- 步骤1** 配置目标主机。

1. 单击导航栏“设置 > 通用设置 > 基础资源管理”。
2. 单击“新建主机集群”，配置以下信息，单击“保存”。

表 2-2 新建主机集群

配置项	配置建议
集群名称	输入“host-group”。
操作系统	选择“Linux”。
主机联通方式	选择“直连模式”。
执行主机	选择“官方资源池”。

3. 单击“添加或导入主机”，配置以下信息，单击“确定”。

表 2-3 添加主机

配置项	配置建议
选择添加方式	选择“通过IP手动添加”。
主机名	建议与在 准备工作 中购买的ECS的名称保持一致。
IP	输入在 准备工作 中购买的ECS的弹性公网IP。
认证方式	选择“密码”。
用户名	输入“root”。
密码	输入在 准备工作 中购买ECS时设置的密码。
ssh端口	输入“22”。

4. 页面显示一条主机记录，当“连通性验证”列的值显示为“成功”，表示主机添加完成。

如果主机添加失败，请根据失败详情排查主机配置。

步骤2 单击导航栏“持续交付 > 部署”，进入部署服务。页面中显示自动创建的应用“Web-Demo-deploy”。

步骤3 单击**，选择“编辑”，进入编辑页面。

步骤4 选择“环境管理”页签，配置主机环境。

1. 单击“新建环境”，配置以下信息，单击“保存”。

表 2-4 新建环境

配置项	配置建议
环境名称	输入“demo-env”。
资源类型	选择“主机”。

配置项	配置建议
操作系统	选择“Linux”。

- 单击“导入主机”，在弹框中选择**步骤1**中配置的主机集群和主机，单击“导入”。
- 页面提示导入成功，关闭此窗口。

步骤5 选择“部署步骤”页签，配置部署步骤。

- 安装JDK：确认jdk版本为“openjdk-1.8.0”。
- 选择部署来源：参考下表进行配置。

表 2-5 部署来源配置

配置项	值
选择源类型	选择“构建任务”。
请选择构建任务	选择“Web-Demo-build”。
下载到主机的部署目录	输入“/usr/local/\${package_name}/”。

- 停止SpringBoot服务：首次执行时，由于目标主机上还未有服务，执行该步骤会失败，因此建议禁用此步骤，单击步骤卡片上的“...”，选择“禁用”。

图 2-3 禁用“停止 SpringBoot 服务”



- URL健康测试：此步骤为可选步骤，本文中选择不启用此步骤。

步骤6 选择“参数设置”页签，参考下表配置参数。

参数名	参数值
host_group	选择 步骤4 中添加的环境名称“demo-env”。
package_url	无需此参数，单击对应行 删除。
service_port	输入“8080”。
package_name	输入“demoapp”。

步骤7 单击“保存并部署”。如果出现弹窗，请确认参数设置准确后，单击“确定”。

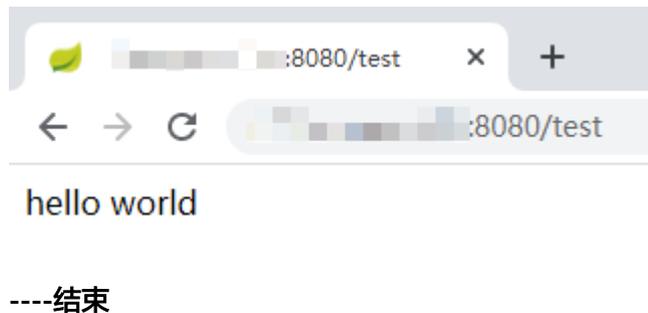
待页面显示 **部署成功**。如果部署失败，请根据失败步骤信息与日志中的报错信息排查。

步骤8 查看部署结果。

打开一个新的浏览器页面，输入访问地址“http://IP:8080/test”，其中“IP”为**准备工作**中购买的ECS的弹性公网IP。

如果出现以下访问结果，表示部署成功。

图 2-4 部署结果



配置流水线

通过流水线可以将代码检查、编译构建、部署等任务串联起来，当代码发生改变时，自动触发流水线执行，实现持续交付。

步骤1 单击导航栏“持续交付 > 流水线”，进入流水线服务。在“流水线”页签中显示自动创建的流水线“Web-Demo-pipeline”。

步骤2 单击**，选择“编辑”。

步骤3 选择“任务编排”页签，配置流水线。

1. 本文暂不涉及接口测试，因此需要将接口测试任务从流水线中移除。
单击任务“测试”对应🗑️，在弹框中单击“确定”。

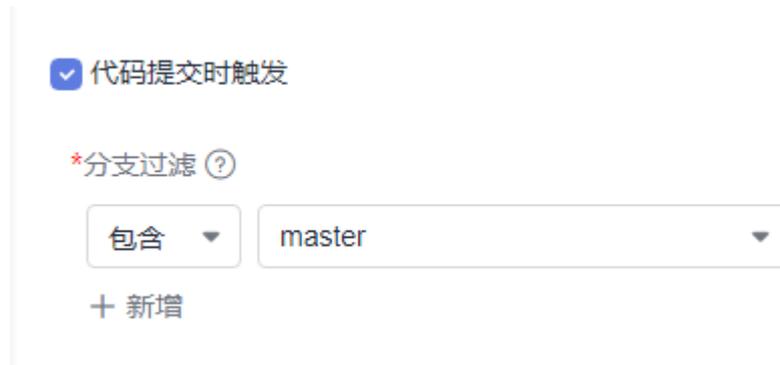
图 2-5 删除任务



2. 单击任务“部署”，关联构建任务选择“构建”，其它配置项的值与**部署构建包**中的参数设置保持一致。

步骤4 选择“执行计划”页签，勾选“代码提交时触发”，在分支过滤下拉列表中勾选分支“master”。

图 2-6 配置执行计划



步骤5 单击“保存”，退出编辑。

步骤6 进入部署服务，编辑部署步骤，启用步骤“停止SpringBoot服务”。

步骤7 进入代码仓库，搜索并打开文件“TestController.java”。

单击, 将“hello world”修改为“hello world again”，输入提交信息后单击“确定”。

图 2-7 修改代码

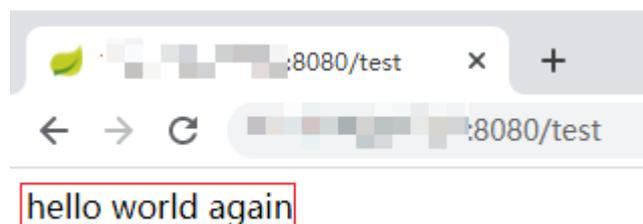
```
8 public class TestController {
9
10
11     @RequestMapping
12     public String index() {
13         return "hello world again";
14     }
15
16 }
17
```

步骤8 返回流水线页面，可看到流水线正在运行中。

当页面显示时，重新访问页面“http://IP:8080/test”，访问结果将变为下图显示内容。

如果任务执行失败，请于执行失败处检查失败原因，可打开步骤详情查看任务日志，根据日志进行排查。

图 2-8 流水线执行结果



----结束

后续操作

为了避免不必要的费用产生，完成本示例体验后，可以释放部分不再使用的资源。可以释放的资源如下。

表 2-6 释放资源

资源名称	操作步骤
CodeArts项目	进入项目的“设置 > 通用设置 > 基本信息”页面，单击“删除项目”，根据页面提示完成删除操作。
ECS	登录ECS控制台，在列表中找到待删除的ECS，单击“更多 > 删除”，根据页面提示完成删除操作。

须知

资源释放后无法恢复，请谨慎操作。

相关信息

本文采用的代码检查、构建、部署、流水线等任务均为代码仓库模板关联的内置任务。

在实际开发项目时可参考以下指导自主创建任务。

表 2-7 创建任务参考

服务	操作方法
代码检查	参考 新建代码检查任务 。
编译构建	参考 新建构建任务 。
部署	参考 新建应用 。
流水线	参考 新建流水线 。

3 使用 CodeArts 快速搭建基于 CCE 部署的代码开发流水线

本文基于CodeArts内置代码仓库，介绍如何使用CodeArts完成项目的开发、构建与部署，实现持续交付。

本文采用的部署方式为CCE部署，适用于容器化部署场景。

如果您希望使用传统软件包部署方法，请参考[使用CodeArts快速搭建基于ECS部署的代码开发流水线](#)。

准备工作

1. 已注册华为云并实名认证。如果您还没有华为账号，请参考以下步骤创建。
 - a. [注册华为账号并开通华为云](#)。
 - b. 完成[实名认证](#)。
2. 已[购买CCE集群](#)，购买时的必要配置请参考[表3-1](#)与[表3-2](#)，表中未列出的配置保持默认即可。

表 3-1 集群购买配置

配置分类	配置项	配置建议
基础配置	集群类型	选择“CCE Standard 集群”。
	计费模式	选择“按需计费”。
	集群名称	输入自定义名称。
	集群版本	建议选择最新版本。
网络配置	虚拟私有云	选择已有的虚拟私有云，如果列表中没有合适的选项，单击“新建虚拟私有云”完成创建。
	默认节点子网	选择已有的子网，如果列表中没有合适的选项，单击“新建子网”完成创建。
	容器网络模型	选择“VPC网络”。

配置分类	配置项	配置建议
	容器网段	勾选“自动设置网段”。

表 3-2 节点配置

配置分类	配置项	配置建议
节点配置	计费模式	选择“按需计费”。
	节点类型	选择“弹性云服务器-虚拟机”。
	节点规格	选择2vCPUs 4GiB或以上规格。
	容器引擎	选择“Docker”。
	操作系统	选择“公共镜像 > CentOS 7.6”。
	登录方式	选择“密码”。
	密码	输入自定义密码。
网络配置	节点IP	选择“自动分配”。
	弹性公网IP	选择“暂不使用”。

- 已在容器镜像服务中创建名称为“web-demo”的组织，操作方法请参考[创建组织](#)。

📖 说明

如果页面提示“组织已存在”，请自定义其它名称。

开通 CodeArts 体验版

步骤1 进入[购买CodeArts套餐页面](#)。

步骤2 选择“体验版”，勾选同意声明，单击“立即开通”。

步骤3 开通成功，返回“软件开发生产线”页面，列表中显示已开通套餐记录。

---结束

新建项目

项目是使用CodeArts各服务的基础，创建项目后才能完成后续操作。

步骤1 在CodeArts控制台单击“立即使用”。

步骤2 单击“新建项目”，选择“Scrum”，输入项目名称“Demo”，单击“确定”。

---结束

新建代码仓库

代码仓库用于项目代码的版本管理，本文使用服务内置的模板“Java Web Demo”创建代码仓库。

步骤1 单击导航栏“代码 > 代码托管”，进入代码托管服务。

步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择模板“Java Web Demo”，单击“下一步”。

步骤4 输入代码仓库名称“Web-Demo”，单击“确定”。

----结束

准备 Dockerfile

Dockerfile是用于构建镜像的文本文件，其中包含了构建镜像所需的指令和说明。了解 Dockerfile更详细说明，请参见[Docker官网](#)。

步骤1 单击仓库名称，进入代码仓库。

步骤2 单击文件列表上方“新建”，在下拉列表中选择“新建文件”。

图 3-1 新建文件



步骤3 输入文件名Dockerfile，文件详情输入以下代码行。

```
FROM openjdk:8-alpine
ADD target /demo
COPY ./target/demoapp.jar /demo
CMD ["java","-jar","/demo/demoapp.jar"]
```

步骤4 输入备注信息，单击“提交”。

----结束

构建并推送镜像

通过编译构建任务将软件的源代码编译成镜像，并把镜像推送归档到容器镜像服务（SWR）中。

步骤1 单击导航栏“持续交付 > 编译构建”，进入编译构建服务。

步骤2 单击“新建任务”，配置任务信息。

1. 基本信息：配置以下信息，单击“下一步”。

表 3-3 构建任务基本信息

配置项	配置建议
名称	输入“Web-Demo-docker”。
代码源	选择“Repo”。
代码仓	选择“Web-Demo”。
默认分支	选择“master”。

2. 构建模板：选择“空白构建模板”，单击“确定”。

步骤3 配置构建步骤。

1. 单击“点击添加构建步骤”，在步骤列表中找到“Maven构建”，单击“添加”。
2. 单击“添加步骤”，在步骤列表中找到“制作镜像并推送到SWR仓库”，单击“添加”。
3. 参照下表配置步骤“制作镜像并推送到SWR仓库”（表中未涉及的字段保持默认配置即可）。

表 3-4 配置镜像信息

配置项	配置建议
组织	输入在 准备工作 中创建的组织名称（“web-demo”）。
镜像标签	v1.0.0

步骤4 完成配置，单击“保存并执行”。

当页面中显示时，表示任务执行成功。如果构建失败，请根据失败步骤信息与日志中的报错信息排查。

步骤5 登录容器镜像服务控制台，在页面左侧导航栏中选择“我的镜像”。

页面中有一条镜像名称为“demo”，所属组织为“web-demo”的记录。

单击镜像名称查看详情，镜像版本为“v1.0.0”。

图 3-2 查看镜像



----结束

创建负载

在云容器引擎（CCE）中创建无状态负载（Deployment），用于加载运行demo镜像。

- 步骤1** 登录云容器引擎控制台，单击在[准备工作](#)中购买的集群，进入详情页。
- 步骤2** 在页面左侧导航选择“工作负载”，单击“创建工作负载”。
- 步骤3** 参考下表完成配置，单击“创建工作负载”。

表 3-5 创建工作负载

配置分类	配置项	配置建议
基本信息	负载类型	选择“无状态负载”。
	负载名称	输入“web-demo”。
	实例数量	选择“1”。
容器配置	镜像名称	单击“选择镜像”，在弹框中勾选“demo”，单击“确定”。
	更新策略	勾选“总是拉取镜像”。
	镜像版本	选择“v1.0.0”。
高级配置	升级策略	升级方式选择“替换升级”。

- 步骤4** 页面提示创建成功，单击“查看工作负载详情”，返回负载详情页面，“实例列表”页签中显示一条记录。

当该记录的状态为“运行中”时，选择“访问方式”页签，单击“创建”，参考下表完成创建服务配置，单击“确定”。

如果实例状态异常，请参考[工作负载异常](#)排查处理。

表 3-6 配置访问方式

配置项	配置建议
Service名称	输入“web-demo”。
访问类型	选择“负载均衡”。
服务亲和	选择“集群级别”。
负载均衡器	<ul style="list-style-type: none"> 选择“共享型 > 自动创建”。 实例名称：输入“web-demo-test”。 弹性公网：选择“自动创建”。
端口配置	<ul style="list-style-type: none"> 协议：TCP 容器端口：8080 服务端口：8080

步骤5 列表中显示一条记录。

当该记录中显示  [web-demo-test](#) 时，将鼠标悬浮在访问类型下的负载均衡器名称处，在弹窗中复制公网地址。

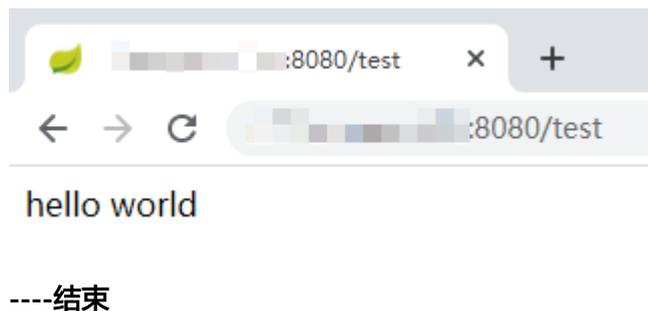
图 3-3 复制访问地址



步骤6 打开新的浏览器页面，输入“http://IP:8080/test”，其中IP为**步骤5**中复制的公网地址。

如果出现以下访问结果，表示负载运行成功。

图 3-4 部署结果



部署镜像

通过在部署服务中创建应用，将镜像部署自动化。

步骤1 返回CodeArts页面，单击导航栏“持续交付 > 部署”，进入部署服务。

1. 单击“新建应用”。
2. 输入应用名称“web-demo-k8s”，单击“下一步”。
3. 选择“空白模板”，单击“确定”。

步骤2 在步骤列表中搜索并添加步骤“Kubernetes快速部署(CCE集群)”，参考下表完成步骤配置。

表 3-7 配置部署步骤

配置项	配置建议
区域	选择集群所在的区域。
集群名称	选择 准备工作 中购买的集群名称。
命名空间名称	选择“default”。
工作负载名称	选择“web-demo”。
容器名称	选择 创建负载 时，容器配置部分的容器名称。

步骤3 单击“保存并部署”。

当页面显示  **部署成功** 时，表示测试通过。如果部署失败，请根据失败步骤信息与日志中的报错信息排查。

----结束

配置流水线实现自动化更新镜像部署

通过配置流水线，将代码仓库、构建、部署串联起来，当代码仓库中发生代码提交动作时，可以自动触发流水线的执行，实现持续交付。

步骤1 单击导航栏“持续交付 > 流水线”，进入流水线服务。

步骤2 单击“新建流水线”，配置流水线。

1. 基本信息：配置以下信息，单击“下一步”。

表 3-8 流水线基本信息

配置项	配置建议
名称	输入“pipeline-web-demo”。
流水线源	选择“Repo”。
代码仓	选择“Web-Demo”。
默认分支	选择“master”。

2. 模板：选择“空模板”，单击“确定”。

步骤3 配置工作流。

1. 单击“阶段_1”后的，在“编辑阶段”窗口中输入阶段名称“构建”，单击“确定”。

图 3-5 编辑阶段名称



2. 单击“新建任务”。
在“新建任务”窗口中，单击“Build构建”插件后的“添加”。

图 3-6 添加任务



3. 参考下表配置任务信息，单击“确定”。

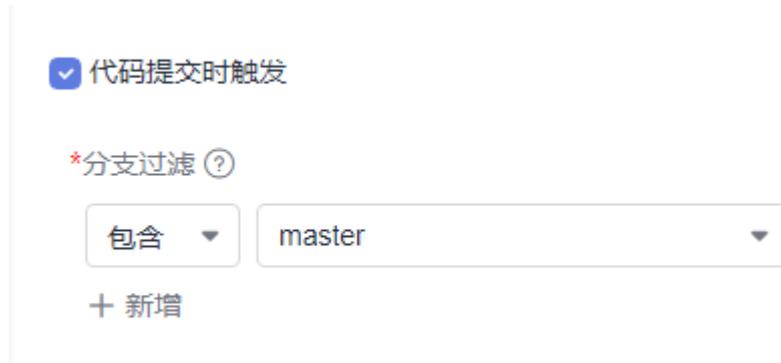
表 3-9 编辑构建任务

配置项	配置建议
名称	保持默认值即可。
请选择需要调用的任务	选择“Web-Demo-docker”。
仓库	选择“Web-Demo”。

4. 单击“新建阶段”，并修改阶段名称为“部署”。
5. 单击“新建任务”，添加插件“Deploy部署”。
6. 选择调用任务“web-demo-k8s”、关联构建任务选择[步骤3.3](#)设置的任务名称。

步骤4 选择“执行计划”页签，勾选“代码提交时触发”，在分支过滤下拉列表中勾选分支“master”。

图 3-7 配置执行计划



步骤5 单击“保存”，退出编辑。

步骤6 进入代码仓库，搜索并打开文件“TestController.java”。

单击, 将“hello world”修改为“hello world again”，输入提交信息后单击“确定”。

图 3-8 修改代码

```

8 public class TestController {
9
10
11     @RequestMapping
12     public String index() {
13         return "hello world again";
14     }
15
16 }
17

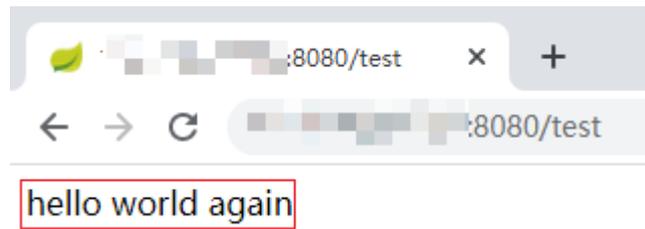
```

步骤7 返回流水线页面，可看到流水线正在运行中。

当页面显示时，重新访问页面“http://IP:8080/test”，访问结果将变为下图显示内容。

如果任务执行失败，请于执行失败处检查失败原因，可打开步骤详情查看任务日志，根据日志进行排查。

图 3-9 流水线执行结果



----结束

后续操作

为了避免不必要的费用产生，完成本示例体验后，可以释放部分不再使用的资源。可以释放的资源如下。

表 3-10 释放资源

资源名称	操作步骤
CodeArts项目	进入项目的“设置 > 通用设置 > 基本信息”页面，单击“删除项目”，根据页面提示完成删除操作。
SWR组织、镜像	<ol style="list-style-type: none"> 1. 登录SWR控制台。 2. 在“我的镜像”页面中，勾选本文中创建的镜像，单击“批量删除”，根据页面提示完成删除操作。 3. 在“组织管理”页面中，单击待删除组织的名称，进入详情页。单击“删除”，根据页面提示完成删除操作。
CCE集群	登录CCE控制台。在列表中找到待删除的集群，单击...，选择“删除集群”，根据页面提示完成删除操作。

须知

资源释放后无法恢复，请谨慎操作。