

软件开发生产线

快速入门

文档版本 03
发布日期 2023-05-08



版权所有 © 华为云计算技术有限公司 2024。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。

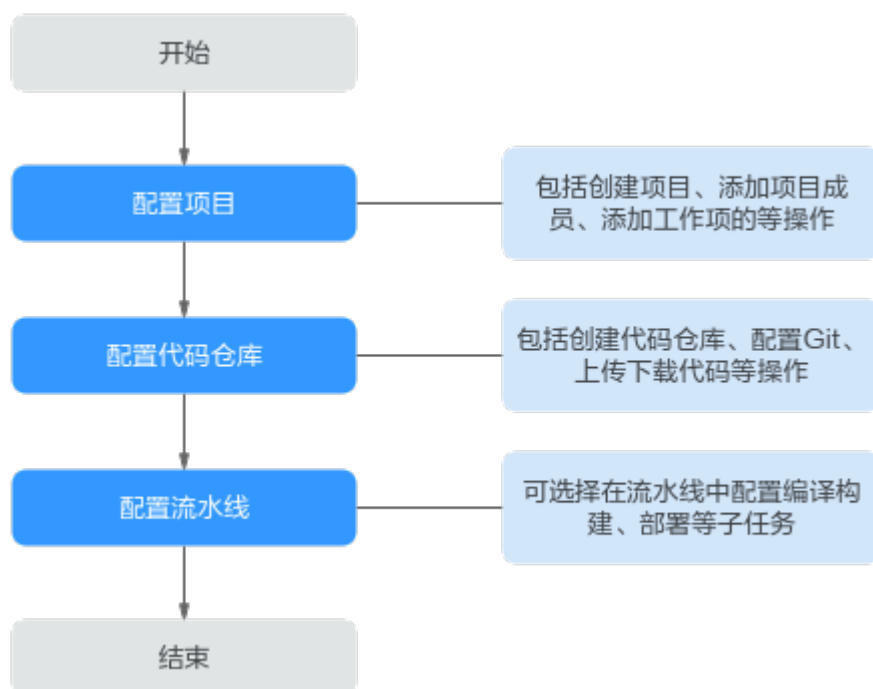
目录

1 软件开发生产线（CodeArts）使用流程.....	1
2 使用 CodeArts 快速搭建项目（ECS 篇）.....	6
3 使用 CodeArts 快速搭建项目（CCE 篇）.....	14

1 软件开发生产线（CodeArts）使用流程

本章节为您介绍软件开发生产线（CodeArts）的基本操作流程。

图 1-1 基本操作流程



前提条件

1. 已[购买CodeArts](#)。
2. 若需要向主机中部署应用，则需要准备一台具有弹性IP的主机，可以使用已有主机，也可以[购买华为云弹性云服务器](#)。

配置项目

需求管理服务是使用CodeArts各服务的基础，首先需要创建项目、添加项目成员，然后可以根据项目规划添加工作项。

步骤1 创建项目。

1. [登录软件开发生产线控制台](#)。
2. 单击📍，选择区域。
3. 单击“立即使用”
4. 单击“新建项目”。
5. 单击“Scrum”，输入项目名称，单击“确定”。

步骤2 添加项目成员。

1. 进入已创建的项目，单击导航栏“设置 > 通用设置 > 成员管理”。
在列表右上方选择“添加成员”，或者“通过链接邀请”。
其中，添加成员有包括以下选择，请根据实际情况选择。
 - 从本企业用户
 - 从其他项目导入用户
 - 邀请其他企业用户
 - 从委托中导入用户

图 1-2 添加成员



步骤3 创建工作项。

1. 进入已创建的项目，单击导航栏“工作”，选择“工作项”页签。
2. 单击“新建”，选择工作项类型，根据实际情况输入工作项标题、优先级、处理人等信息。

---结束

需求管理服务更多操作请参见[需求管理用户指南](#)。

配置代码仓库

代码仓库用于项目代码的版本管理，与本地的交互基于Git，因此在使用代码仓库之前需要在本地安装Git客户端。

步骤1 安装并配置Git客户端。

1. 通过[Git官网](#)下载安装包，并使用默认配置安装Git客户端到本地电脑。
2. 运行Git Bash，Git Bash中输入以下命令行配置用户名和邮箱。

```
git config --global user.name "您的名字"  
git config --global user.email "您的邮箱"
```

3. 输入以下命令行，生成一对SSH密钥。生成的密钥通常保存在“~/ssh/id_rsa.pub”中。

```
ssh-keygen -t rsa -C "您的邮箱"
```
4. 输入以下命令可显示密钥内容。

```
cat ~/.ssh/id_rsa.pub
```

步骤2 创建代码仓库。

1. 进入已创建的项目，单击导航栏“代码 > 代码托管”。
2. 单击“新建仓库”。
3. 代码托管服务提供三种仓库类型：“普通仓库”、“模板仓库”或“导入仓库”，根据需要选择仓库类型，单击“下一步”。
4. 按照页面提示输入仓库名称等基本信息，单击“确定”。
5. 代码仓库创建成功后，在仓库里列表页面单击仓库名称，可查看仓库内的文件。

步骤3 克隆/推送代码。

1. 单击页面右上角用户名，在下拉列表中选择“个人设置”。
2. 单击导航“代码托管 > SSH密钥管理”。
3. 单击“添加SSH密钥”，输入标题，将负责[安装并配置Git客户端](#)中生成的密钥粘贴到文本框中，勾选同意声明，单击“确定”。
4. 返回代码托管服务页面，单击待克隆的仓库名称，进入仓库。
5. 单击“克隆/下载”，复制仓库SSH下载链接。
6. 运行Git Bash，输入以下命令行，克隆云端仓库到本地。

```
git clone 您仓库的SSH下载链接
```
7. 在本地完成代码的编辑后，在Git Bash中依次输入以下命令行，保存代码并推送到代码仓库。

```
git add .  
git commit -m "您对本次提交备注的信息"  
git push origin master
```
8. 返回代码仓库，即可查看更新后的文件。

---结束

代码托管服务更多操作请参见[代码托管用户指南](#)。

配置流水线

流水线集成代码检查、编译构建、部署等任务，可根据需要灵活配置流水线中的任务。流水线为非必须操作，根据需要配置即可。

- 代码检查服务可以对代码进行静态检查和安全检查，代码检查为可选操作，根据需要配置即可。
- 编译构建服务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。构建为可选操作，对于一些项目，例如PHP、Node.js前端代码，无需配置构建任务。
- 部署服务将软件包或代码部署到虚拟机或者容器。部署为可选操作，对于一些项目，例如手机APP开发，无需此类部署方式。

步骤1 创建代码检查任务。

1. 进入已创建的项目，单击导航栏“代码 > 代码检查”。
2. 单击“新建任务”，在页面中选择需要检查的代码仓库，单击仓库名称后的“新建任务”。

3. 任务创建成功后，进入“概览”页面，单击“开始检查”启动任务。
4. 待任务执行成功后，可以查看检查结果、问题列表、修改建议等。

代码检查服务更多操作请参考[代码检查用户指南](#)。

步骤2 创建构建任务。

1. 进入已创建的项目，单击导航栏“持续交付 > 编译构建”
2. 单击“新建任务”，根据需要配置任务信息。
 - a. 基本信息：配置以下信息，单击“下一步”。

表 1-1 构建任务基本信息

配置项	配置建议
任务名称	输入自定义名称。
源码源	选择“Repo”。
源码仓库	选择在 配置代码仓库 中创建的代码仓库。
分支	根据需要选择仓库分支。

- b. 构建模板：可选择内置模板，也可以选择“空白构建模板”，单击“下一步”。
3. 根据需要配置构建步骤、参数、执行计划，单击“新建并执行”。
4. 待任务执行完成后，可以在页面中查看构建结果与构建日志。

编译构建服务更多操作指导请参考[编译构建用户指南](#)。

步骤3 创建部署应用。

1. 进入已创建的项目，在页面导航栏选择“设置 > 通用设置 > 基础资源管理”，新建主机集群，并将[前提条件](#)中准备的主机加入集群。
2. 进入已创建的项目，在页面导航栏选择“持续交付 > 部署”。
3. 单击“新建应用”，根据需要配置应用信息。
 - a. 基本信息：输入自定义应用名称，单击“下一步”。
 - b. 选择部署模板：可选择内置模板，也可以选择“空白模板”，单击“确定”。
4. 根据需要配置部署步骤、参数、环境等信息，单击“保存并部署”。
5. 待应用部署成功后，可以在页面中查看部署结果、部署日志、报错信息等。
部署服务更多操作指导请参考[部署用户指南](#)。

步骤4 配置流水线。

1. 进入已创建的项目，单击导航栏“持续交付 > 流水线”。
2. 选择“流水线”页签，单击“新建流水线”，根据实际情况配置流水线信息。
 - a. 基本信息：配置以下信息，单击“下一步”。

表 1-2 流水线基本信息

配置项	配置建议
名称	输入自定义名称。
流水线源	选择“Repo”。
代码库名称	选择在 配置代码仓库 中创建的代码仓库。
默认分支	根据需要选择仓库分支。

- b. 模板：可选择内置模板，也可以选择“空白模板”，单击“确定”。
3. 根据需要配置任务编排、参数、执行计划，单击“保存并运行”。
4. 待任务执行成功后，可单击各任务查看其执行详情。

流水线服务更多操作指导请参考[流水线用户指南](#)。

----结束

2 使用 CodeArts 快速搭建项目（ECS 篇）

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ECS部署，若需了解CCE部署方法，请参考[使用CodeArts快速搭建项目（CCE篇）](#)。

准备工作

1. 已[购买CodeArts](#)。
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”。
	CPU架构	选择“x86计算”。若无此配置项，请忽略。
	规格	选择2核4G或以上规格。
	镜像	选择“公共镜像 > CentOS > CentOS 7.6 64bit(40GB)”。
网络配置	弹性公网IP	选择“现在购买”。
	公网带宽	选择“按带宽计费”。
高级配置	登录凭证	选择“密码”。
	密码	输入自定义密码。

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 登录软件开发生产线控制台。

步骤2 单击 ，选择区域。

步骤3 单击“立即使用”

步骤4 单击“新建项目”。

步骤5 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”。

步骤3 根据需要选择“模板仓库”，单击“下一步”

步骤4 选择模板“Java Web Demo”，单击“下一步”。

步骤5 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-codecheck”。

说明


本文使用的是代码仓库模板关联的内置任务。

实际开发项目需要单击“新建任务”，根据业务场景自主创建任务，详细操作请参考[新建代码检查任务](#)。

步骤2 单击任务所在行的“立即执行”启动任务。

说明

如果代码检查任务已被执行，则单击任务名称进入概览页，单击“开始检查”重新执行任务。

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。

若任务执行失败，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-cloudbuild”。

📖 说明

本文使用的是代码仓库模板关联的内置任务。

实际开发项目需要单击“新建任务”，根据业务场景自主创建任务，详细操作请参考[新建构建任务](#)。

步骤2 单击任务所在行的▶启动任务。若出现弹窗，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。

若构建失败，请根据失败步骤信息与日志中的报错信息排查。

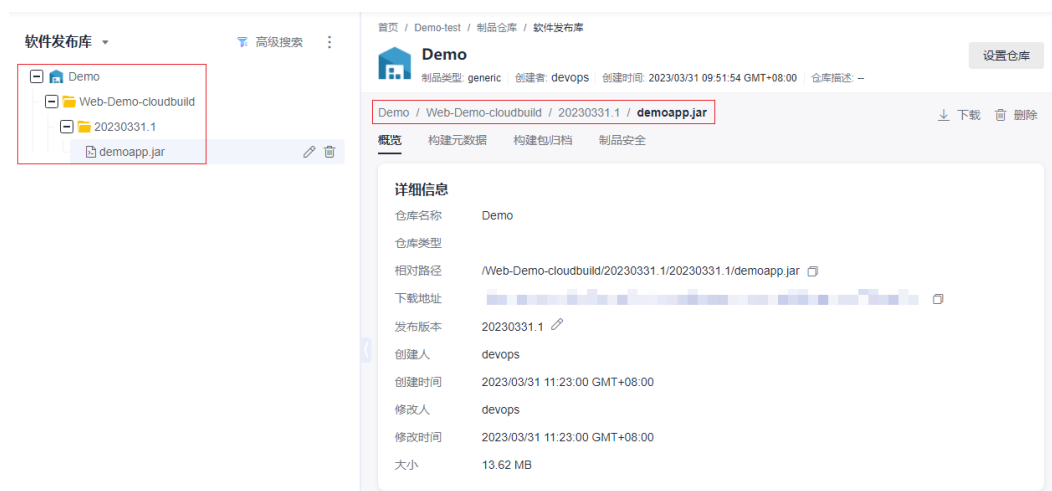
图 2-1 构建编号



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”，输入集群名称“hosts”、选择操作系统“Linux”，单击“保存”。
3. 单击“新增目标主机”，在弹框中配置以下信息，勾选同意声明后，单击“添加”。

表 2-2 新增目标主机

配置项	配置建议
主机名	输入自定义主机名称。为了方便辨认，可与在 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IP。
用户名	输入“root”。
密码	输入在 准备工作中 购买的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。
若主机添加失败，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”。

说明

本文使用的是代码仓库模板关联的内置应用。


实际开发项目需要单击“新建应用”，根据业务场景自主创建应用，详细操作请参考[新建应用](#)。

步骤3 单击“”，在下列表中选择“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”，输入环境名称“host-group”、选择资源类型“主机”、操作系统“Linux”，单击“保存”。
2. 列表中新增一条环境记录，单击环境名称，在滑出的窗口中选择“资源列表”页签。
3. 单击“导入主机”，在弹框中的下拉列表中选择[步骤1](#)中创建的主机集群，并在列表中勾选主机，单击“导入”。
4. 页面提示导入成功，关闭此窗口。

步骤5 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“host-group”。
package_url	无需此参数，单击对应行  删除。

参数名	参数值
service_port	输入“8080”。
package_name	输入“demoapp”。

步骤6 选择“部署步骤”页签，配置部署步骤。


- 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，因此不勾选“启用”选项。
- 安装JDK：修改jdk版本为“openjdk-1.8.0”。
- 选择部署来源：参考下表进行配置。

表 2-3 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-cloudbuild”。
下载到主机的部署目录	输入“/usr/local/\${package_name}/”。

- URL健康测试：此步骤可有可无，根据需要选择是否启用（本文不勾选“启用”选项）。

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

待页面显示  **部署成功**。若部署失败，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IP。

若出现以下访问结果，表示部署成功。

图 2-3 部署结果



----结束

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”页面中显示自动创建的流水线“Web-Demo-pipeline”。

📖 说明

本文使用的是代码仓库模板关联的内置流水线。

实际开发项目需要单击“新建流水线”，根据业务场景自主创建流水线，详细操作请参考[新建流水线](#)。

步骤2 单击**，在下拉列表中选择“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“apitest”对应🗑️，在弹框中单击“确定”。

图 2-4 删除任务



2. 单击任务“codecheck”
3. 单击任务“deploy”，关联构建任务选择“cloudbuild”，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 2-5 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，在“停止SpringBoot服务”中勾选“启用”。

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 2-6 修改代码

```

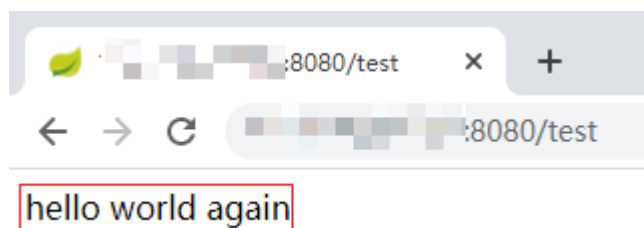
8 public class TestController {
9
10
11     @RequestMapping
12     public String index() {
13         return "hello world again";
14     }
15
16 }
17
    
```

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。

若任务执行失败，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 2-7 流水线执行结果



----结束

释放资源

为了避免不必要的费用产生，完成本示例体验后，可释放以下资源。

须知

资源释放后无法恢复，请谨慎操作。

步骤1 删除项目。

1. 进入项目“设置 > 通用设置 > 基本信息”页面。
2. 单击“删除项目”，在弹框中输入项目名称，单击“删除”。

步骤2 删除ECS。

1. 登录ECS控制台，在列表中找到待删除的ECS，单击“更多”，在下拉列表中选择“删除”。
2. 勾选弹框中的所有选项，单击“是”。

----结束

3 使用 CodeArts 快速搭建项目（CCE 篇）

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。

本文介绍采用CCE部署方式，若需了解ECS部署方法，请参考[使用CodeArts快速搭建项目（ECS篇）](#)。

准备工作

1. 已[购买CodeArts](#)。
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	计费模式	选择“按需计费”。
	集群版本	建议选择最新版本。
网络配置	网络模型	选择“VPC网络”。
	虚拟私有云	选择已有的虚拟私有云，若列表中没有合适的选项，单击“新建虚拟私有云”完成创建。
	控制节点子网	选择已有的子网，若列表中合适的选项，单击“新建子网”完成创建。
	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
计算配置	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”。

配置分类	配置项	配置建议
	节点规格	选择2核8G及以上规格即可。
	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“随机分配”。
	弹性公网IP	选择“暂不使用”。

3. 已在容器镜像服务中[创建组织](#)，本文中创建的组织名称为“web-demo”。

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 [登录软件开发生产线控制台](#)。

步骤2 单击，选择区域。

步骤3 单击“立即使用”

步骤4 单击“新建项目”。

步骤5 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”。

步骤3 根据需要选择“模板仓库”，单击“下一步”

步骤4 选择模板“Java Web Demo”，单击“下一步”。

步骤5 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，文件详情输入以下命令行。

```
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“确定”。

----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
任务名称	输入自定义名称（本文中为“Web-Demo-docker”）。
源码源	选择“Repo”。
源码仓库	选择“Web-Demo”。
分支	选择“master”。

2. 构建模板：选择“空白构建模板”，单击“下一步”。


步骤3 配置构建步骤。

- 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”。
- 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”。
- 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“新建并执行”。

当页面中显示时，表示任务执行成功。若构建失败，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“创建负载”，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	自定义，本文中为“web-demo”。

配置分类	配置项	配置建议
	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3 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”。

若实例状态异常，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	自定义，本文中为“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	选择“共享型 > 自动创建”，自定义实例名称（本文中为“web-demo-test”），勾选已阅读须知。 说明 若账号下已有负载均衡器，可选择“共享型 > 使用已有”，并选择已存在的负载均衡器名称。
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4 列表中显示一条记录，当服务名称出显示为 ● web-demo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。

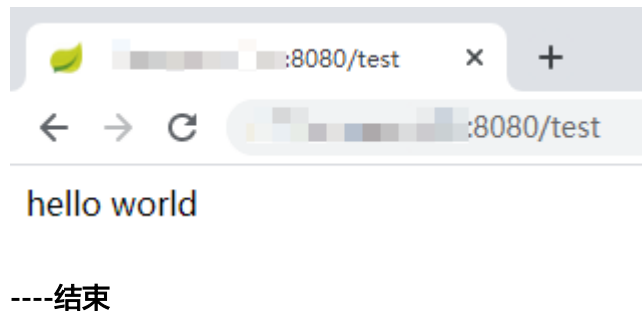
图 3-3 复制访问地址



步骤5 打开新的浏览器页面，输入“http://IP:8080/test”，其中IP为**步骤4**中复制的公网地址。

若出现以下访问结果，表示镜像部署并运行成功。

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”，输入应用名称（本文中为“web-demo-k8s”），单击“下一步”。
 2. 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	本文中选择“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。若部署失败，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”。
代码库名称	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

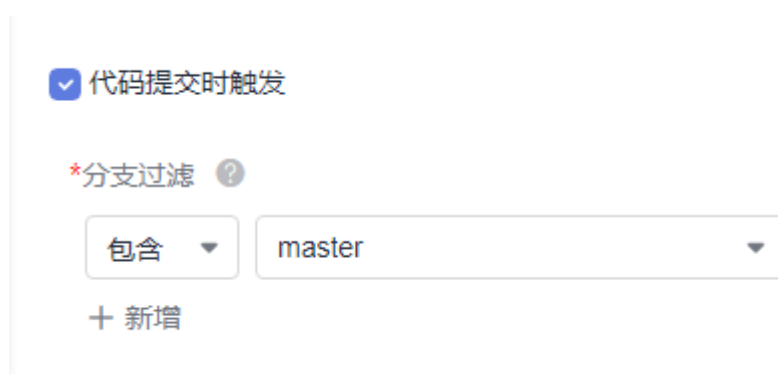
配置项	配置建议
名称	输入自定义名称（本文中保持默认值）。
请选择需要调用的任务	选择“Web-Demo-docker”。

配置项	配置建议
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”，单击“保存”。

图 3-7 配置执行计划



步骤5 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

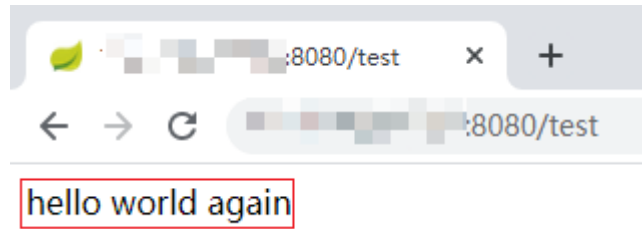
```

步骤6 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。

若任务执行失败，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

释放资源

为了避免不必要的费用产生，完成本示例体验后，可释放以下资源。

须知

资源释放后无法恢复，请谨慎操作。


步骤1 删除项目。

1. 进入项目“设置 > 通用设置 > 基本信息”页面。
2. 单击“删除项目”，在弹框中输入项目名称，单击“删除”。

步骤2 删除组织与镜像。

1. 登录容器镜像服务控制台。
2. 在“我的镜像”页面中，勾选本文中创建的镜像，单击“删除”。在弹框中单击“是”。
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”，在弹框中单击“是”。

步骤3 删除集群。

1. 登录云容器引擎控制台。在列表中找到待删除的集群，单击.
2. 勾选弹框中的所有选项，单击“是”。

----结束