

软件开发生产线

快速入门

文档版本

03

发布日期

2025-07-01



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..... 2

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	- 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	- 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
代码托管	- 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	- 通过编译构建服务完成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	- 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

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](#)。

操作流程

操作步骤	说明
准备工作	注册华为账号并开通华为云，为账户充值，购买ECS。
步骤一：开通CodeArts体验版	CodeArts支持包年/包月计费，使用CodeArts前需要完成购买，开通体验版即可完成本文中的操作流程。
步骤二：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。
步骤三：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。
步骤四：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。
步骤五：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。
步骤六：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。
步骤七：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。
释放资源	按需计费的资源会产生持续计费，为了避免不必要的费用产生，完成体验后，可以释放不再使用的资源。

准备工作

- 已[注册华为账号并开通华为云](#)。
- 参考[自定义购买ECS](#)，购买满足以下配置的ECS。
 - 计费模式：建议选择按需计费。
 - CPU架构：x86计算。
 - 节点规格：2vCPUs | 4GiB或以上规格，系统盘≥80GiB。
 - 操作系统：公共镜像-CentOS 7.6
 - 弹性公网IP：自动创建。

完成购买后，参考[配置安全组规则](#)添加两条入方向规则。

- 协议为“TCP”、端口为“22”、源地址为“0.0.0.0/0”。
- 协议为“TCP”、端口为“8080”、源地址为“0.0.0.0/0”。

步骤一：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”。

待开通成功，在“软件开发生产线”页面中显示已开通套餐记录。

----结束

步骤二：新建项目

步骤1 在CodeArts控制台单击“进入工作台”。

步骤2 在CodeArts首页单击“新建 > 新建项目”，进入新建项目页面。

步骤3 选择模板“Scrum”。

步骤4 输入项目名称（项目名称不超过128个字符，本文中使用“Demo”），单击“确定”。

新建项目成功，页面自动跳转至“工作项”页面。

----结束

步骤三：新建代码仓库

步骤1 在项目中单击导航栏“代码 > 代码托管”，进入代码托管服务。

步骤2 单击“新建仓库”，进入新建仓库页面。

步骤3 选择“按模板新建”，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（代码仓库名称需以大小写字母、数字、下划线开头，可包含大小写字母、数字、中划线、下划线、英文句点，但不能以.git、.atom或.结尾，本文中使用“Web-Demo”），单击“确定”。

新建代码仓库成功，页面跳转至仓库内，显示仓库的代码文件列表。

----结束

步骤四：检查代码

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**，选择“设置”。

步骤3 单击导航“规则集”，在规则中关闭“CSS”、“HTML”后的开关。

图 2-1 编辑规则集



步骤4 单击“开始检查”，启动任务。

当页面中显示 **检查成功** 时，表示任务执行成功。

如果任务执行失败，请根据报错提示，参考[代码检查常见问题](#)排查处理。

步骤5 单击“代码问题”页签查看问题列表。

由于无致命或严重问题，本文中暂不修改代码。

----结束

步骤五：构建并归档软件包

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”。

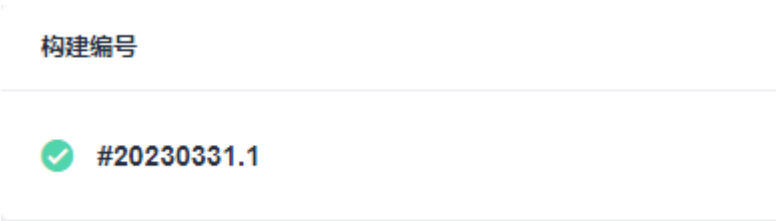
步骤2 单击任务所在行的▶启动任务。如果出现弹窗，请确认参数设置准确后，单击“确定”。

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，参考[编译构建常见问题](#)排查处理。

步骤3 单击任务名称，查看任务详情。在“构建历史”页面找到最新一次构建的构建编号，记录该编号。

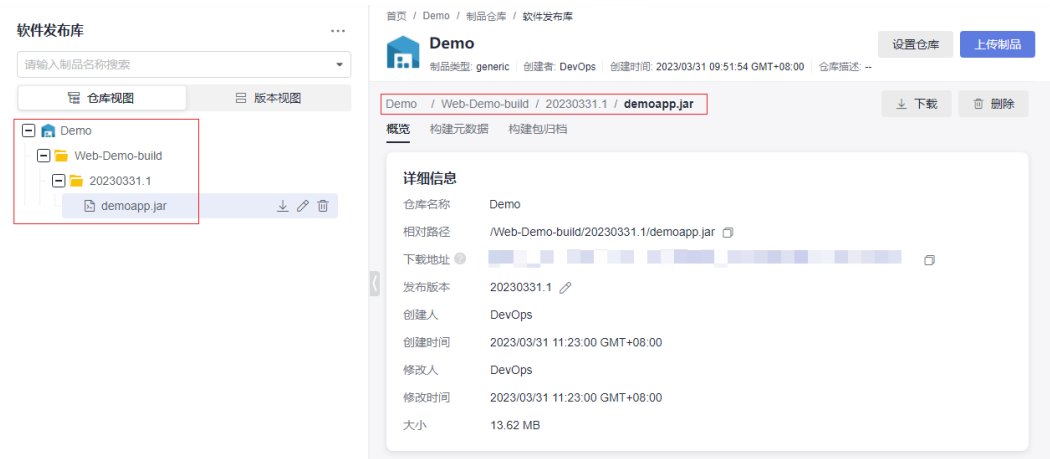
图 2-2 构建编号



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“仓库视图”中找到与项目名称同名的仓库，依次进入文件夹“Web-Demo-build > [步骤3中记录构建编号](#)”，即可找到生成的软件包“demoapp.jar”。

图 2-3 查看软件包



----结束

步骤六：部署构建包

- 步骤1** 配置目标主机。
- 单击导航栏“设置 > 通用设置 > 基础资源管理”，进入基础资源管理页面。
 - 单击“新建主机集群”，配置以下基本信息，单击“保存”。

表 2-1 新建主机集群

配置项	示例	说明
集群名称	host-group	自定义主机集群名称，仅支持3-128位数字、中文、英文字母、“-”、“.”。

配置项	示例	说明
操作系统	Linux	选择加入集群的主机的操作系统类型，可以选择“Linux”或“Windows”。
主机联通方式	直连模式	选择部署服务与目标主机的联通方式，可以选择“直连模式”或“代理模式”。
执行主机	官方资源池	资源池是部署软件包时执行部署命令的物理环境的集合。 可以选择服务提供的“官方资源池”，也可以将自有的服务器作为资源池。

3. 待页面提示新建主机集群成功，在“目标主机”页签中单击“添加或导入主机”，选择添加方式“导入已购ECS”，在列表中找到准备工作
- 中购买的ECS，单击“导入”。

图 2-4 导入 ECS



4. 配置以下信息，单击“确定”。

表 2-2 添加主机

配置项	示例	说明
用户名	root	ECS的登录用户名，Linux主机通常为“root”。
密码	输入在准备工作中购买ECS时设置的密码。	ECS的登录密码。
ssh端口	22	通常为“22”，也可以自定义端口。

5. 页面显示一条主机记录，当“连通性验证”列的值显示为“成功”时，表示主机添加完成。
- 如果主机添加失败，请根据失败详情，参考主机管理常见问题排查处理。

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。
- 单击“新建环境”, 配置以下基本信息, 单击“保存”。

表 2-3 新建环境

配置项	示例	说明
环境名称	demo-env	自定义环境名称, 仅支持3-128位数字、中文、英文字母、“-”、“_”、“.”。
资源类型	主机	环境中的资源类型, 默认为“主机”。
操作系统	Linux	选择即将添加到环境中的主机的操作系统类型, 可以选择“Linux”或“Windows”。

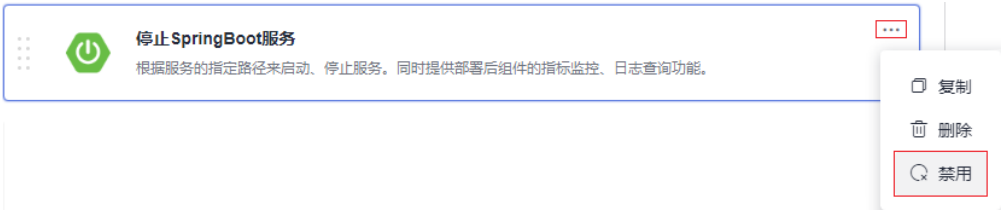
- 待页面提示创建成功, 在“资源列表”页签中单击“导入主机”, 在弹框中选择[步骤1](#)中配置的主机集群和主机, 单击“导入”。
 - 待页面提示导入成功, 关闭此窗口。
- 步骤5** 选择“部署步骤”页签, 配置部署步骤。
- 安装JDK: 确认jdk版本为“openjdk-1.8.0”。
 - 选择部署来源: 参考下表进行配置。

表 2-4 部署来源配置

配置项	示例	说明
选择源类型	构建任务	软件包的来源, 可以选择“制品仓库”或“构建任务”。
请选择构建任务	Web-Demo-build	当“选择源类型”为“构建任务”时, 显示此配置项。
下载到主机的部署目录	/usr/local/\${package_name}/	软件包在目标主机中的保存路径。

- 停止SpringBoot服务: 首次执行部署时, 由于目标主机上还未启动过SpringBoot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 2-5 禁用“停止 SpringBoot 服务”



- 启动SpringBoot服务：保持默认配置即可。
- 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。

表 2-5 配置参数

名称	默认值
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”。

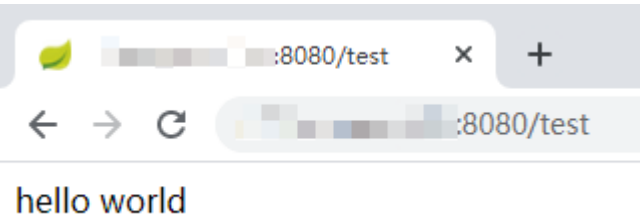
待页面显示 **部署成功**。如果部署失败，请根据失败步骤信息与日志中的报错信息，参考**部署常见问题**排查处理。

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 2-6 部署结果



----结束

步骤七：配置流水线

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”。

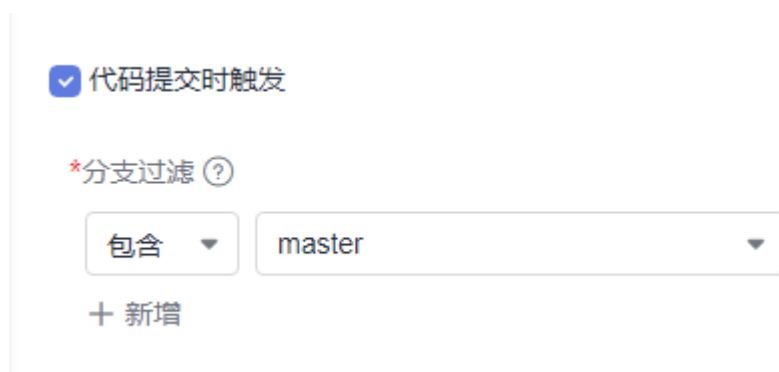
图 2-7 删除任务



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**步骤六：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 2-8 配置执行计划



步骤5 单击“保存”，退出编辑。

保存成功，页面显示更新后的执行计划。

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 2-9 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

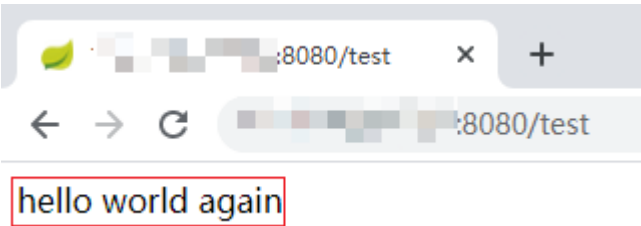
```

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。

如果流水线执行失败，可单击失败原因提示，打开日志，参考[流水线常见问题](#)排查处理。

图 2-10 流水线执行结果



----结束

释放资源



资源释放后无法恢复，请谨慎操作。

本文中涉及的按需计费资源为ECS。

如果完成体验后不再使用ECS，建议参考[删除弹性云服务器](#)释放该资源。

相关文档

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 2-6 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](#)。

操作流程

操作步骤	说明
准备工作	注册华为账号并开通华为云，为账户充值，购买CCE，创建SWR组织。
步骤一：开通CodeArts体验版	CodeArts支持包年/包月计费，使用CodeArts前需要完成购买，开通体验版即可完成本文中的操作流程。
步骤二：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。
步骤三：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。
步骤四：准备Dockerfile	创建用于构建镜像的文本文件“Dockerfile”，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 。
步骤五：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SWR中。
步骤六：创建负载	在CCE中创建无状态负载（Deployment），用于加载运行demo镜像。
步骤七：部署镜像	通过在部署服务中创建应用，将镜像部署自动化。
步骤八：配置流水线	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。

操作步骤	说明
释放资源	按需计费的资源会产生持续计费，为了避免不必要的费用产生，完成体验后，可以释放部分不再使用的资源。

准备工作

- 已[注册华为账号并开通华为云](#)。
- 已购买满足以下配置的CCE集群。

表 3-1 集群配置要求

配置分类	配置要求	参考
集群	建议选择按需计费。 • 集群类型：CCE Standard集群。 • 集群版本：建议选择最新版本。 • 控制节点架构：X86。 • 容器网络模型：VPC网络。 • 容器网段：自动设置网段。	购买Standard/Turbo集群
节点	建议选择按需计费。 • 节点类型：弹性云服务器-虚拟机。 • 节点规格：2vCPUs 8GiB或以上规格。 • 容器引擎：Docker。 • 操作系统：公共镜像-CentOS 7.6。	创建节点

- 已在容器镜像服务中创建名称为“web-demo”的组织（如果页面提示“组织已存在”，请自定义其它名称。），操作方法请参考[创建组织](#)。

步骤一：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”。
- 待开通成功，在“软件开发生产线”页面中显示已开通套餐记录。
- 结束

步骤二：新建项目

- 步骤1 在CodeArts控制台单击“进入工作台”。
- 步骤2 在CodeArts首页单击“新建 > 新建项目”，进入新建项目页面。
- 步骤3 选择模板“Scrum”。

步骤4 输入项目名称（项目名称不超过128个字符，本文中使用“Demo”），单击“确定”。

新建项目成功，页面自动跳转至“工作项”页面。

----结束

步骤三：新建代码仓库

步骤1 在项目中单击导航栏“代码 > 代码托管”，进入代码托管服务。

步骤2 单击“新建仓库”，进入新建仓库页面。

步骤3 选择“按模板新建”，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（代码仓库名称需以大小写字母、数字、下划线开头，可包含大小写字母、数字、中划线、下划线、英文句点，但不能以.git、.atom或.结尾，本文中使用“Web-Demo”），单击“确定”。

新建代码仓库成功，页面跳转至仓库内，显示仓库的代码文件列表。

----结束

步骤四：准备 Dockerfile

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。

```
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

步骤五：构建并推送镜像

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 3-2 构建任务基本信息

配置项	示例	说明
名称	Web-Demo-docker	自定义构建任务名称。支持中英文、数字、下划线“_”、“-”，长度不超过115个字符。
代码源	Repo	选择构建任务的代码来源，支持Repo、GitHub等多种来源。
代码仓	Web-Demo	选择需要编译的代码仓库。
默认分支	master	选择需要编译的代码仓库分支。

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。

- 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 3-3 配置镜像信息

配置项	示例	说明
组织	输入在 准备工作 中创建的组织名称。	选择镜像推送到SWR后所属的组织。
镜像标签	v1.0.0	自定义用来标记镜像的版本。支持大小写字母、数字、“.”、“_”、“-”，不可以“.”或“-”开头，最多可包含128个字符。

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，参考[编译构建常见问题](#)排查处理。

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

步骤六：创建负载

- 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”。
- 配置项详细说明请参考[创建无状态负载](#)。

表 3-4 创建工作负载

配置分类	配置项	示例
基本信息	负载类型	无状态负载
	负载名称	web-demo
	实例数量	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	v1.0.0
高级配置	升级策略	替换升级

- 步骤4 待页面提示创建成功，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”。
- 配置项详细说明请参考[创建负载均衡类型的服务](#)。
- 如果实例状态异常，请参考[工作负载异常](#)排查处理。

表 3-5 配置访问方式

配置项	示例
Service名称	web-demo
访问类型	负载均衡
服务亲和	集群级别
负载均衡器	1. 选择“共享型 > 自动创建”。 2. 配置以下信息： - 实例名称：web-demo-test。 - 弹性公网IP：自动创建。
端口配置	- 协议：TCP。 - 容器端口：8080。 - 服务端口：8080。

创建成功后，列表中显示一条新增记录。

步骤5 刷新页面，查看访问方式列表，当列表中显示  [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



----结束

步骤七：部署镜像

- 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（应用名称仅支持中英文、数字、“-”、“_”，长度为3-128个字符，本文中使用“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 3-6 配置部署步骤

配置项	示例	说明
区域	选择 准备工作 中购买的集群所在的区域。	选择目标集群所在的区域。
集群名称	选择 准备工作 中购买的集群名称。	选择目标集群的名称。
命名空间名称	default	选择目标集群中的命名空间。
工作负载名称	web-demo	选择需要部署的工作负载。
容器名称	选择 步骤六：创建负载 时，在“容器配置”部分显示的容器名称。	选择需要部署的容器实例名。

- 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，参考[部署常见问题](#)排查处理。

----结束

步骤八：配置流水线

- 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 3-7 流水线基本信息

配置项	示例	说明
名称	pipeline-web-demo	自定义流水线名称。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。
流水线源	Repo	选择流水线的代码来源。支持Repo、通用Git等多种来源。
代码仓	Web-Demo	选择流水线关联的代码仓库名称。
默认分支	master	选择流水线关联的代码仓库分支。

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（阶段名称支持中文、大小写英文字母、数字、“_”、“-”、“.”、“:”、“.”、“.”、“/”、“(”、“)”、“(”、“)”、空格, 其中空格不可在名称开头或结尾使用, 长度为1-128个字符。此处使用“构建”), 单击“确定”。

图 3-5 编辑阶段名称

编辑阶段

×

* 阶段名称

构建

总是运行 ?

☐ 是

☒ 否

确定

取消

- 单击“新建任务”选择“从空任务新建”，页面右侧滑出“从空任务新建”窗口。
- 在列表中找到“Build构建”，单击“添加”。

图 3-6 添加任务

 **Build构建** [任务](#)

调用CodeArts Build能力进行构建。编译构建服务（CodeArts Build）为开发者...

[添加](#)

4. 参考下表配置任务信息，单击“确定”。

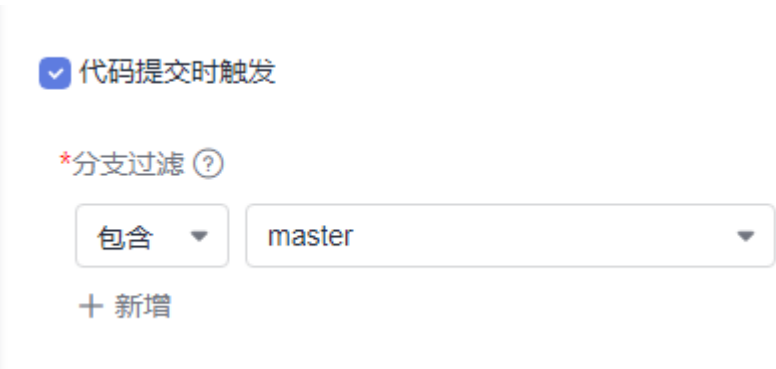
表 3-8 编辑构建任务

配置项	示例	说明
名称	保持默认值即可。	自定义任务名称。支持中文、大小写英文字母、数字、“-”、“_”、“.”、“/”、“(”、“)”、“(”、“)”、空格，其中空格不可在名称开头或结尾使用，长度为1-128个字符。
请选择需要调用的任务	Web-Demo-docker	选择本项目下和当前流水线源配置代码仓相同、或构建来源配置为来自流水线的构建任务。
仓库	Web-Demo	选择构建任务关联的代码仓库。

5. 单击“新建阶段”，并修改阶段名称为“部署”。添加成功后，页面中显示新增的阶段。
6. 单击“新建任务”，添加插件“Deploy部署”。
7. 选择调用任务“web-demo-k8s”、关联构建任务选择步骤3.4设置的任务名称，单击“确定”。

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。

保存成功，页面显示更新后的执行计划。

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

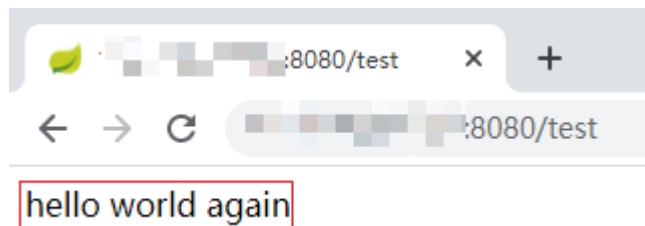
```

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。

如果流水线执行失败，可单击失败原因提示，打开日志，参考[流水线常见问题](#)排查处理。

图 3-9 流水线执行结果



----结束

释放资源

警告

资源释放后无法恢复，请谨慎操作。

本文中涉及的按需计费资源为CCE。

如果完成体验后不再使用CCE，建议参考[删除按需计费的集群](#)，释放该资源。