

流水线

最佳实践

文档版本 01
发布日期 2024-07-29



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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 通过微服务变更流水线修复项目 BUG 并快速发布.....	1
2 配置准出条件并对代码检查结果进行校验.....	7
3 通过流水线参数串联编译构建服务和部署服务.....	13
4 基于 Kubernetes 原生 Service 的场景完成微服务蓝绿发布.....	21
5 HE2E DevOps 实践：配置流水线.....	29

1 通过微服务变更流水线修复项目 BUG 并快速发布

方案概述

CodeArts Pipeline为企业提供了一种微服务模型，将企业大军团的作战方式转变为小团队精英作战方式，支持每个微服务独立进行开发、验证、部署及上线发布，缩短需求发布周期。同时支持企业按照业务功能组织团队，优化管理模型，使运作更精细化，有效提升团队运作效率。

在微服务中，用户可以创建流水线资源，并将其设置为变更流水线，用于与微服务下变更资源进行联动，通过变更流水线发布一个或多个变更来实现项目的快速交付。

操作流程

本文介绍如何通过微服务变更流水线模拟修复项目下的一个BUG并快速发布，基本操作流程如下：

步骤1：新建流水线微服务

步骤2：新建变更流水线

步骤3：新建微服务变更

步骤4：执行变更流水线

表 1-1 操作流程说明

流程	说明
新建流水线微服务	在项目下新建一个微服务，用于管理某个特定的业务功能。
新建变更流水线	在微服务下新建一条变更流水线，用于发布微服务下的变更。
新建微服务变更	在微服务下新建一个变更，承载项目下修复BUG的开发活动。
执行变更流水线	执行流水线，发布变更代码。

准备工作

- **新建项目**（选择“Scrum”模板，命名为“Project01”）。
- 在项目下**新建工作项**（以修复一个BUG单为例，命名为“BUGFIX”）。
- 在项目下**新建代码仓库**（选择“Java Maven Demo”模板，命名为“Repo01”）。
- **新建CodeArts Repo HTTPS服务扩展点**（命名为“HttpsEndpoint01”）。

步骤 1：新建流水线微服务

步骤1 **登录华为云控制台页面。**

步骤2 单击页面左上角，在服务列表中选择“开发与运维 > 流水线 CodeArts Pipeline”。

步骤3 单击“立即使用”，进入流水线服务首页。

步骤4 单击顶部导航栏“首页”，在首页搜索**准备工作**中已创建的项目，单击项目名称，进入项目。

步骤5 在左侧导航栏选择“持续交付 > 流水线”，进入项目下流水线列表页面。

步骤6 切换到“微服务”页签，进入微服务列表页面。

步骤7 单击“新建微服务”，进入“新建微服务”页面，配置以下信息。

表 1-2 新建微服务

参数项	说明
所属项目	微服务所在的项目，不可更改。
微服务名称	微服务的名称，输入“Microservice01”。
代码源	微服务关联的源码仓库，仅支持“Repo”。
代码仓	选择 准备工作 中已创建的代码仓“Repo01”。
默认分支	默认使用的分支，选择“master”。
语言	指定微服务开发语言，选择“Java”。
描述	对微服务的介绍说明，选填。

步骤8 单击“确定”，完成微服务的创建。

----结束

步骤 2：新建变更流水线

步骤1 在微服务列表页面，单击微服务名称，进入微服务“概览”页面。

步骤2 切换到“流水线”页签，进入微服务流水线列表页面。

步骤3 单击“新建流水线”，进入“新建流水线”页面，配置以下信息。

表 1-3 新建变更流水线

参数项	说明
所属项目	变更流水线所在的项目，不可更改。
名称	变更流水线的名称，使用自动生成的即可。
代码源	变更流水线关联的源码类型，自动和微服务保持一致。
代码仓	变更流水线关联的源码仓库，自动和微服务保持一致。
默认分支	默认使用的分支，自动和微服务保持一致。
Repo https授权	打开“基于变更触发”后需配置“Repo HTTPS授权”，配置授权扩展点以提升代码仓操作权限。选择 准备工作 中已创建的授权扩展点“HttpsEndpoint01”。
别名	输入别名，会生成该仓库对应的相关系统参数，这里不设置。
基于变更触发	打开“基于变更触发”，将当前流水线设置为微服务变更流水线。
描述	对变更流水线的介绍说明，选填。

📖 说明

微服务变更流水线仅支持基于变更触发运行，且单个微服务下只能存在一条微服务变更流水线。

步骤4 单击“下一步”，选择“Maven构建”模板，自动生成阶段和任务，使用默认配置即可。

步骤5 单击“保存”，完成微服务变更流水线的创建。

----结束

步骤 3：新建微服务变更

步骤1 重新进入微服务“概览”页面。

步骤2 切换到“变更”页签，进入微服务变更列表页面。

步骤3 单击“新建变更”，进入“新建变更”页面，配置以下信息。

表 1-4 新建变更

参数项	说明
变更标题	变更的名称，输入“fix-a-bug”。
代码仓	变更关联的源码仓库，自动和微服务保持一致。
代码分支	变更新需求的开发分支，变更通过流水线发布成功后，会将代码分支自动合并至微服务的默认分支。选择“从默认分支拉取新分支”并输入分支名称“bugfix”。

参数项	说明
关联工作项	变更关联的工作项，选择 准备工作中 中已创建的工作项“BUGFIX”。

步骤4 单击“确定”，完成变更的创建。

变更创建完成后，进入代码开发阶段，系统基于微服务默认分支自动创建出本次变更的特性分支，开发人员可以基于该特性分支进行代码开发，问题修复后，模拟向该分支提交代码。

----结束

步骤 4：执行变更流水线

代码更新完成后，即可通过变更流水线执行变更。

步骤1 在微服务变更列表页面，单击变更名称，进入“变更详情”页面。

步骤2 单击右上角“提交发布”，弹出“提交发布”确认框，提示是否确认将变更提交至变更流水线。

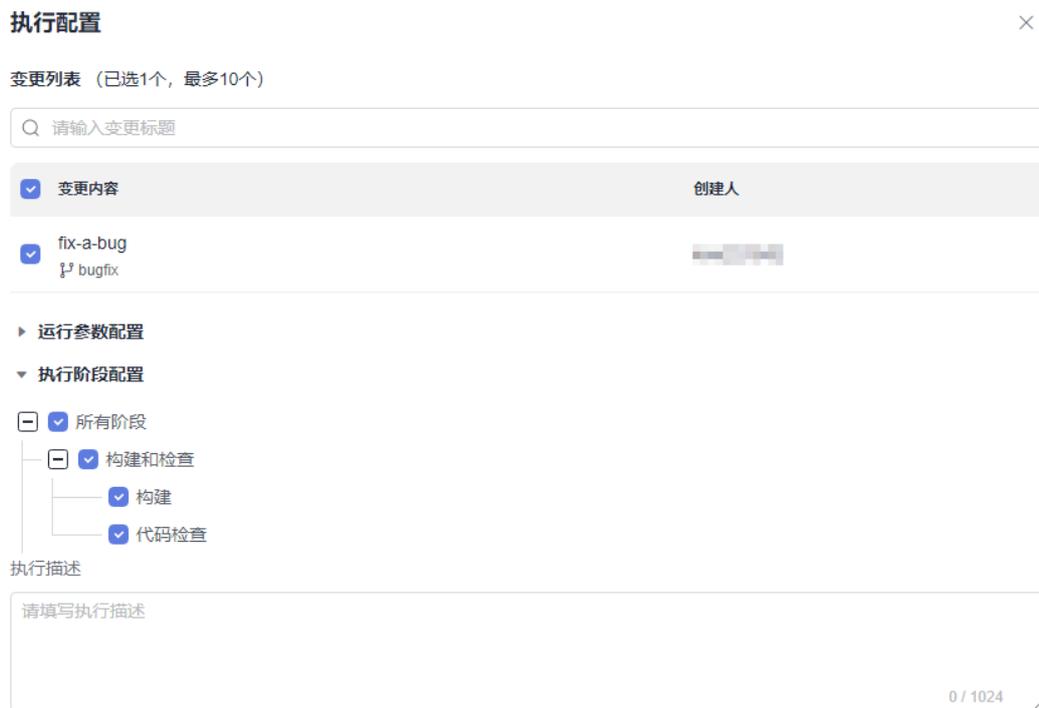
图 1-1 提交发布



步骤3 单击“确定”，完成变更提交，页面跳转至对应变更流水线发布列表页面。

步骤4 单击右上角“执行”，弹出“执行配置”侧滑框，选择刚提交的变更，其他配置保持默认。

图 1-2 执行配置



步骤5 单击“执行”，启动变更流程。

微服务变更流水线在运行过程中，会自动增加“合并成分支”和“合并主干”阶段，并将本次变更关联的特性分支合并到该成分支上。

代码检查、构建任务执行成功后，系统自动进入“合并主干”阶段，并弹出“手动确认”对话框。

图 1-3 执行流水线



步骤6 单击“继续执行”，“合并主干”阶段执行完成后，系统将自动完成以下事项。

- 将本次发布的变更状态更新为“已发布”。
- 将本次发布的变更关联的工作项状态更新为“已关闭”。
- 将本次发布使用的成分支代码内容合并到微服务默认分支，实现代码自动合并。

至此，即完成了一次基本的微服务变更发布流程。

----结束

2 配置准出条件并对代码检查结果进行校验

方案概述

传统软件质量保障以人工验收测试为主，效率低。当前软件能力难以支持业务需要，大量测试验收需要人工手动进行，自动化覆盖率低，且需要人工筛查识别满足验收条件的版本进行后续发布等处理流程，整体测试效率低下。

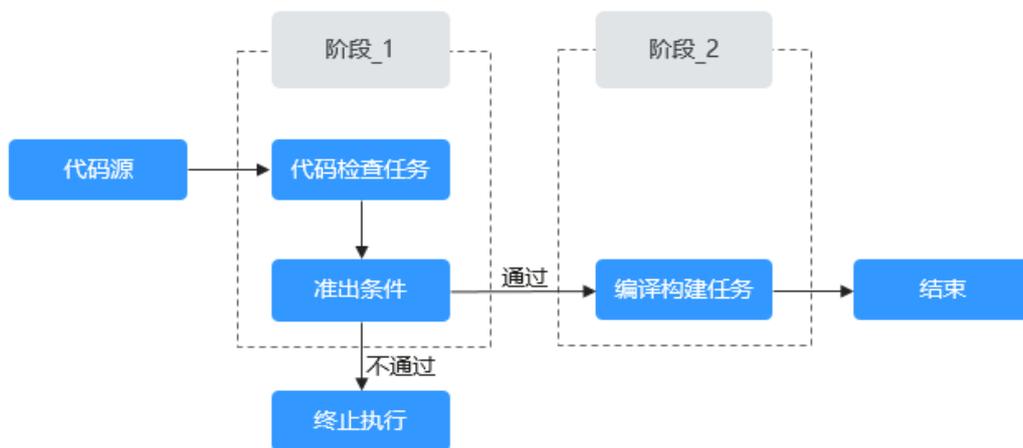
CodeArts Pipeline提供统一的准出条件管理能力，通过配置规则和策略，实现阶段准出校验，并最终控制流水线的执行。用户根据实际需要将策略资源应用到流水线作为准出条件，帮助高效管理项目，保障产品高质量交付。

通过流水线准出条件-Check代码检查规则策略自动化看护，拦截70%+问题，提高测试效率的同时，提高软件质量。

操作流程

本文基于流水线准出条件-Check代码检查规则策略，通过添加代码检查任务所在阶段的准出条件，对代码检查任务的检查结果进行自动化校验，只有检查结果满足准出条件，才可以进入下一个阶段。

图 2-1 操作流程示意图



基本操作流程如下：

- **步骤1: 新建规则并配置阈值**
- **步骤2: 新建策略并添加规则**
- **步骤3: 配置流水线**
- **步骤4: 执行流水线验证准出条件**

表 2-1 操作流程说明

流程	说明
新建规则并配置阈值	新建一个代码检查类型规则，基于代码检查插件的输出结果，设置比较关系和阈值条件，供策略使用，作为准出条件的判断依据。
新建策略并添加规则	新建一个策略，并添加如上代码检查类型规则。
配置流水线	在流水线阶段准出条件添加如上策略，根据设置的规则条件，来控制流水线执行。
执行流水线验证准出条件	执行流水线，验证准出条件拦截结果。 <ul style="list-style-type: none">• 代码检查插件满足规则条件，通过准出条件校验，流水线继续执行。• 代码检查插件如果不满足规则条件，没有通过准出条件校验，流水线终止执行。

准备工作

- **新建项目**（选择“Scrum”模板，命名为“Project01”）。
- 在项目下**新建代码仓库**（选择“Java Maven Demo”模板，命名为“Repo01”）。
新建代码仓库时会自动创建与代码仓库同名的代码检查任务，将代码检查任务名称修改为“CheckTask01”，操作方法请参考[配置代码检查任务基本信息](#)。
- 基于“Repo01”仓库**新建构建任务**（选择“Maven”模板，命名为“BuildTask01”）。
- 基于已创建的代码仓库**新建流水线**（选择空模板，命名为“Pipeline01”）。

步骤 1: 新建规则并配置阈值

步骤1 [登录华为云控制台页面](#)。

步骤2 单击页面左上角 ，在服务列表中选择“开发与运维 > 流水线 CodeArts Pipeline”。

步骤3 单击“立即使用”，进入流水线服务首页。

步骤4 单击右上角头像图标，在下拉菜单中选择“租户设置”，进入租户设置页面。

步骤5 在左侧导航栏选择“策略管理 > 规则”，进入规则管理页面。

步骤6 单击“新建规则”，进入“新建规则”页面，配置以下信息。

图 2-2 新建规则

表 2-2 配置规则信息

参数项	说明
名称	规则名称，输入“校验代码检查结果”。
类型	规则类型，选择“代码检查”。
选择插件	规则绑定的插件名称，选择“Check代码检查”。
插件版本	规则绑定的插件版本，选择“0.0.1”。
阈值配置	根据选中的插件版本自动获取的插件输出阈值配置，这里使用默认值即可。

步骤7 单击“确定”，完成规则创建。

----结束

步骤 2：新建策略并添加规则

策略实行分层管理，可分为租户级策略和项目级策略。其中，租户级策略可以应用到当前租户下所有项目的流水线，项目级策略可以应用到当前项目下所有流水线。本节以租户级策略为例进行介绍。

步骤1 单击左侧导航栏“策略”，进入策略管理页面。

📖 说明

租户级策略中默认存在一条“系统策略”，用户可查看和使用该策略，但不能进行编辑和删除。

步骤2 单击“新建策略”，进入“新建策略”页面，输入策略名称，勾选**步骤1**中新建好的规则。

图 2-3 新建策略



步骤3 单击“确定”，完成策略创建。

----结束

步骤 3: 配置流水线

步骤1 单击顶部导航栏“首页”，返回软件开发生产线首页。

步骤2 找到**准备工作**中已创建的项目，单击项目名称，进入项目。

步骤3 在左侧导航栏选择“持续交付 > 流水线”，进入项目下流水线列表页面。

步骤4 搜索**准备工作**中已创建的流水线，单击“操作”列...，选择“编辑”，进入“任务编排”页面。

步骤5 单击“阶段_1”的+ **新建任务**，添加**准备工作**中已创建的代码检查任务，选择“Full”检查模式。

图 2-4 添加代码检查任务



步骤6 单击“阶段_1”的 **准出条件**，在弹出的侧滑框里添加“标准策略准出条件”，并选择**步骤2**中新建好的策略。

图 2-5 添加准出条件

标准策略准出条件
选择一个标准插件策略用于门禁拦截。

* 名称
标准策略准出条件

* 策略 [策略管理](#)
代码检查任务门禁

✓ 校验代码检查结果

检查缺陷统计

检查项	比较	阈值
致命	=	0
严重	=	0
一般	=	0
提示	=	0
问题总数	=	0

步骤7 单击  或者 **+ 新建阶段**，为流水线添加一个新阶段，并在新阶段添加**准备工作中**中已创建的构建任务，选择构建任务关联的仓库。

图 2-6 添加构建任务

← 更换插件

Build构建  ② 温馨提示
调用CodeArts Build能力进行构建。编译构建服务（CodeArts Build）为开发者提供配置简单... [展开](#)

* 名称
Build构建

* 请选择需要调用的任务  [找不到合适的任务?点击创建](#) | [刷新](#)
BuildTask01

* 仓库
Repo01

产物标识 
请输入

----结束

步骤 4：执行流水线验证准出条件

步骤1 配置完流水线后，单击“保存并执行”，等待流水线执行。

步骤2 执行完成后，查看准出条件拦截结果。

- 如果代码检查任务的检查结果通过准出条件校验，则顺利执行到下一阶段。

图 2-7 执行流水线



- 如果代码检查任务的检查结果没有通过准出条件校验，则终止执行，单击准出条件，可查看详细校验结果。



----结束

3 通过流水线参数串联编译构建服务和部署服务

方案概述

使用编译构建生成软件包并通过部署服务部署时，手动配置获取软件包容易导致软件包不一致，且页面跳转多操作繁琐。通过流水线参数串联编译构建服务和部署服务，将参数传递给构建任务和部署任务，实现任务间数据流动的统一，只需要配置好流水线参数，即可完成构建、部署任务的数据拉通。

操作流程

本文以版本号参数为例，介绍如何将流水线参数传递给编译构建任务和部署应用，基本操作流程如下：

步骤1：新建编译构建任务

步骤2：新建部署应用

步骤3：新建并执行流水线

步骤4：查看编译构建和部署结果

表 3-1 操作流程说明

流程	说明
新编译构建任务	新建编译构建任务，新增版本号参数，并在构建步骤引用该参数。
新建部署应用	新建部署任务，新增软件包参数，并在部署步骤引用该参数。
新建并执行流水线	新建流水线，新增版本号参数，并在流水线上添加编译构建任务及部署应用。 <ul style="list-style-type: none">在编译构建任务的版本号参数引用流水线版本号参数。在部署应用软件包参数引用流水线版本号参数。

流程	说明
查看编译构建和部署结果	分别查看编译构建结果和部署结果。 <ul style="list-style-type: none"> 编译构建包版本号为流水线传递的动态参数。 部署任务成功获取软件包。

准备工作

- **新建项目**（选择“Scrum”模板，命名为“Project01”）。
- 在项目下**新建代码仓库**（选择“Java Maven Demo”模板，命名为“Repo01”）。
- 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**。

步骤 1：新建编译构建任务

步骤1 **登录华为云控制台页面。**

步骤2 单击页面左上角 ，在服务列表中选择“开发与运维 > 流水线 CodeArts Pipeline”。

步骤3 单击“立即使用”，进入流水线服务首页。

步骤4 单击顶部导航栏“首页”，在首页搜索**准备工作中**已创建的项目，单击项目名称，进入项目。

步骤5 在左侧导航栏选择“持续交付 > 编译构建”，进入编译构建页面。

步骤6 单击“新建任务”，填写基本信息。

表 3-2 构建任务基本信息

参数项	说明
名称	构建任务名称，输入“BuildTask01”。
所属项目	构建任务所在的项目，不可更改。
代码源	构建任务关联的代码源类型，选择“Repo”。
代码仓	选择 准备工作中 已创建的代码仓“Repo01”。
默认分支	默认使用的分支，选择“master”。

步骤7 单击“下一步”，选择Maven系统模板，单击“确定”。

步骤8 在“参数设置”页面，新建releaseversion参数并打开运行时设置。

图 3-1 新建构建参数

名称	类型	默认值	私密参数	运行时设置	参数描述	操作
codeBranch	字符串	master	<input type="checkbox"/>	<input checked="" type="checkbox"/>	代码分支,系统预定义参数	
releaseversion	字符串		<input type="checkbox"/>	<input checked="" type="checkbox"/>		

步骤9 在“构建步骤”页面，单击“上传软件包到软件发布库”。发布版本号配置为参数引用形式“`${releaseversion}`”，引用`releaseversion`参数，其他参数保持默认。

图 3-2 配置构建步骤



步骤10 单击“保存”，完成构建任务的创建。

----结束

步骤 2: 新建部署应用

步骤1 在左侧导航栏选择“设置 > 通用设置 > 基础资源管理”，新建主机集群，并将已购买的主机加入集群。

步骤2 在左侧导航栏选择“持续交付 > 部署”，进入部署页面。

步骤3 单击“新建应用”，填写任务名称“DeployTask01”，单击“下一步”，选择“空白模板”，单击“确定”。

步骤4 在“参数设置”页面，新建`package_url`参数并打开运行时设置。

图 3-3 新建部署参数

名称	类型	默认值	私密参数	运行时设置	参数描述	操作
package_url	字符串		<input type="checkbox"/>	<input checked="" type="checkbox"/>		+ 删除

步骤5 在“环境管理”页面，单击“新建环境”，命名为“Environment01”，并将集群中的主机导入环境。

步骤6 在“部署步骤”页面，添加“选择部署来源”步骤，配置步骤信息。

图 3-4 配置部署步骤

选择部署来源

选择软件包路径或者根据构建记录，将制品库中对应的软件包下载到目标主机集群中。
[查看操作指南](#)

* 步骤显示名称

* 选择源类型

制品仓库

构建任务

* 环境 管理 | 新建

Environment01
▼
🔄

* 选择软件包

`${package_url}`
⋮

* 下载到主机的部署目录

`/usr/local`

软件包将会下载到该路径下,且该路径必须对当前用户具有可写权限:
(e.g. Linux: /usr/local/tomcat/apache-tomcat-8.5.38/webapps ; Windows: C:/tomcat/apache-tomcat-8.5.38/webapps)

▲ 控制选项

失败后继续运行

使用sudo权限执行

表 3-3 配置部署步骤

参数项	说明
步骤显示名称	步骤名称。保持默认即可。
选择源类型	软件包来源。选择“制品仓库”
环境	部署环境。选择“Environment01”。
选择软件包	待部署的软件包。这里将获取构建任务上传至软件发布库的构建包，配置为参数引用形式“ <code>\${package_url}</code> ”，引用 package_url 参数。
下载到主机的部署目录	软件包下载到目标主机的路径。输入“ <code>/usr/local</code> ”。
控制选项	保持默认。

步骤7 单击“保存”，完成部署应用的创建。

----结束

步骤 3：新建并执行流水线

步骤1 在左侧导航栏选择“持续交付 > 流水线”，进入流水线页面。

步骤2 单击“新建流水线”，根据需要配置流水线信息。

1. 基本信息：配置以下信息，单击“下一步”。

表 3-4 流水线基本信息

配置项	配置建议
名称	流水线名称，输入“Pipeline01”。
代码源	流水线关联的代码源类型，选择“Repo”。
代码仓	选择 准备工作 中已创建的代码仓“Repo01”。
默认分支	默认使用的分支，选择“master”。

2. 选择模板：选择“空模板”，单击“确定”。

步骤3 在“参数设置”页面，新建releaseversion参数，默认值设置为\${TIMESTAMP}，并打开运行时设置。

步骤4 在“任务编排”页面，系统默认生成两个阶段（“代码源”和“阶段_1”），单击“新建阶段”新增一个阶段“阶段_2”。

图 3-5 任务编排



1. 添加编译构建任务

- a. 单击“阶段_1”的“新建任务”，弹出“新建任务”侧滑框。
- b. 单击“构建”分类，找到“Build构建”插件。
- c. 将鼠标移动到插件，单击“添加”，选择**已创建的构建任务**，选择构建任务关联的仓库，releaseversion参数配置为参数引用形式“\${releaseversion}”，引用流水线的releaseversion参数。

图 3-6 添加构建任务

← 更换插件

 **Build构建** ② 温馨提示
官方自研

调用CodeArts Build能力进行构建。编译构建服务（CodeArts Build）为开发者提供配置简单的混合语言构建平台，实现编译构建云端化，支撑企业实现持续交付，缩短交付周期，提升交付效率。支持编译构建... [展开](#)

* 名称

Build构建

* 请选择需要调用的任务 ② 找不到合适的任务? [点击创建](#) | [刷新](#)

BuildTask01

* 仓库

Repo01

* releaseversion

\${releaseversion}

产物标识 ②

请输入

2. 添加部署应用

- a. 单击“阶段_2”的“新建任务”，弹出“新建任务”侧滑框。
- b. 在搜索框输入“部署”关键字或单击“部署”分类，找到“Deploy部署”插件。
- c. 将鼠标移动到插件，单击“添加”，选择**已创建的部署应用**，输入 package_url，并关联上一步的构建任务。

图 3-7 添加部署应用

← 更换插件

 ① 温馨提示

调用CodeArts Deploy能力进行部署。部署服务（CodeArts Deploy）提供可视化、一键式部署服务，支持部署到虚拟机或者容器，提供Tomcat、SpringBoot等模板或者自由组装编排原子步骤进行部署，支持并... [展开](#)

* 名称

Deploy部署

* 请选择需要调用的任务 找不到合适的任务? [点击新建](#) | [刷新](#)

DeployTask01

* package_url

/BuildTask01/\${releaseversion}/javaMavenDemo-1.0.jar

关联构建任务

Build构建

📖 说明

package_url为软件包地址，输入构建包在软件发布库的相对路径，且在路径中引用流水线的releaseversion参数。

步骤5 任务编排完成后，单击“保存并执行”，开始执行流水线。

----结束

步骤 4：查看编译构建和部署结果

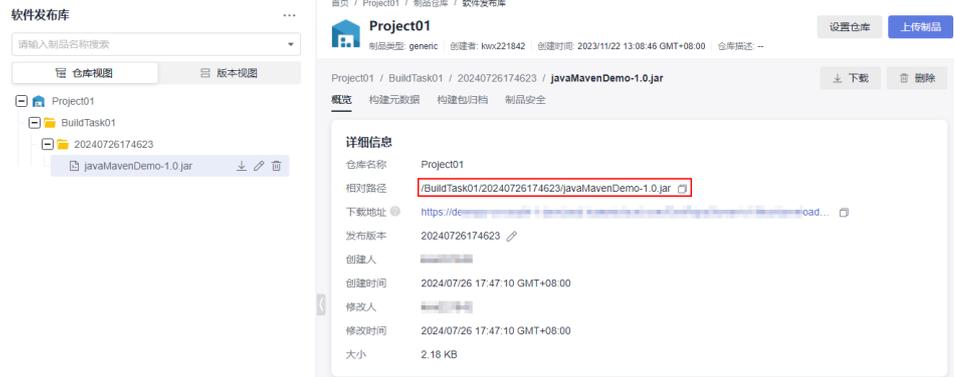
流水线执行成功后，分别查看流水线参数在编译构建和部署任务之间传递的结果。

- **查看编译构建结果**

- 在左侧导航栏选择“制品仓库 > 软件发布库”。
- 展开左侧项目导航树，查看构建成功后上传的软件包。

如图，可以看到相对路径中的版本号即流水线releaseversion参数传递过来的时间戳。

图 3-8 查看软件包



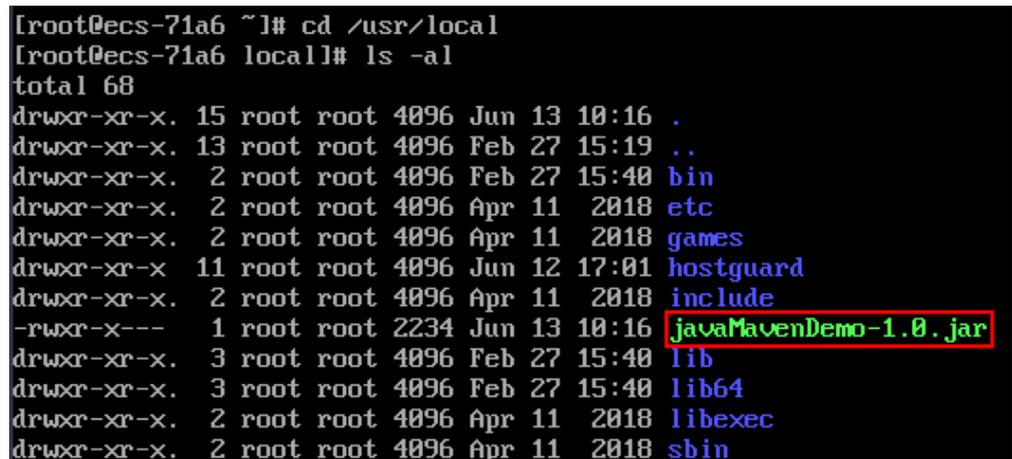
● 查看部署结果

1. 单击右上角用户名。
2. 单击“CodeArts控制台”。
3. 单击左上角, 找到弹性云服务器服务, 单击并进入。
4. 找到本次部署使用的弹性云服务器, 单击“操作”列“远程登录”。
5. 选择“其他方式 > 使用控制台提供的VNC方式登录”, 单击“立即登录”。
6. 输入购买弹性云服务器时配置的用户名和密码, 回车。
7. 输入如下命令, 回车, 进入新建部署应用时配置的部署目录“/usr/local”。

```
cd /usr/local
```
8. 输入如下命令, 回车, 即可查看刚部署成功的软件包, 说明参数正常传递部署应用成功获取到了软件包。

```
ls -al
```

图 3-9 查看部署结果



4 基于 Kubernetes 原生 Service 的场景完成 微服务蓝绿发布

方案概述

应用场景

在进行服务发布的时候，通常会碰到这样的问题：

- 直接在生产环境发布后再进行测试，可能会带来较大风险，对线上用户造成影响。
- 测试环境虽然和生产环境高度一致，然而实际情况下，由于资源配置、网络环境等，测试环境和生产环境之间总会存在差异，因此难以在测试环境覆盖验证所有场景。

通过灰度发布可以实现在不引发风险的同时，在生产环境进行发布及验证。

方案优势

灰度发布是在生产环境中创建与当前线上服务完全一致的工作负载（灰度负载），仅对其中的包版本（业务代码和配置）进行更新，但是新创建的工作负载不承接任何现网流量，对线上用户没有任何影响，就可以在没有风险的情况下，在生产环境进行测试了。在灰度环境验证无问题之后，就可以逐渐将线上用户的真实访问引流到灰度负载，直至完全引流后，新创建的灰度负载承接所有现网流量，原先的线上负载不承接任何流量，此时就可以安全地删除旧负载，保留新负载，完成一次发布。

按照引流方式的不同，当前常用的灰度发布方式包括：快速完成100%流量切换的蓝绿发布和按流量比例或线上请求内容引流的金丝雀发布。通常情况，蓝绿发布已经足以降低发布风险，蓝绿发布有如下优势：

- 不需要集群安装额外的插件。
- 无风险的进行生产环境灰度验证。
- 快速进行流量切换（从旧负载切换新负载，或从新负载切回旧负载）。
- 在验证阶段发现问题后，可以快速将线上服务恢复（回滚）至发布前的状态。

操作流程

本文介绍在不需要占用集群额外资源安装插件的情况下，基于Kubernetes原生Service的场景完成微服务蓝绿发布，基本操作流程如下：

步骤1：新建发布管理环境**步骤2：配置发布管理策略****步骤3：通过云原生发布插件进行发布**

表 4-1 操作流程说明

流程	说明
新建发布管理环境	新建一个用于灰度发布的测试环境，并配置好资源集群等信息。
配置发布管理环境	为发布环境配置发布策略，编排配置发布流程及插件。
通过云原生发布插件进行发布	通过流水线云原生发布插件，发布环境。

准备工作

- **新建项目**（选择“Scrum”模板，命名为“Project01”）。
- **新建空仓库空模板流水线**（命名为“Pipeline01”）。
- 具有创建部署应用的权限，可参考部署服务[编辑权限管理](#)添加权限。
- 开通购买CCE服务，并在CCE集群中创建服务相关Service和Deployment：**创建Service和Deployment**。
 - Service和Deployment资源相关参考：[服务（Service）](#)、[Deployments](#)。
 - Service需要配置为关联至Deployment而非Pod，并且有且仅有一个Deployment与之关联。

- Deployment示例配置，以下示例将创建名为nginx的工作负载：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
    version: v1
spec:
  replicas: 1
  minReadySeconds: 10
  progressDeadlineSeconds: 60
  strategy:
    rollingUpdate:
      maxSurge: 20%
      maxUnavailable: 20%
  selector:
    matchLabels:
      app: my-nginx
      version: v1
  template:
    metadata:
      labels:
        app: my-nginx
        version: v1
    spec:
      containers:
        - name: nginx
          image: nginx
          imagePullPolicy: IfNotPresent
```

```

startupProbe:
  tcpSocket:
    port: 80
livenessProbe:
  tcpSocket:
    port: 80
readinessProbe:
  tcpSocket:
    port: 80
    
```

- Service示例配置，以下示例将创建名为nginx-service的Service资源：

```

apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  labels:
    app: nginx-service
spec:
  type: NodePort
  selector:
    app: my-nginx
    version: v1
  ports:
    - port: 8000
      targetPort: 80
      nodePort: 32500
    
```

📖 说明

通常默认创建的资源即可满足要求，若有问题，请检查Deployment的spec.selector.matchLabels配置（如上示例加粗部分）和Service的spec.selector配置（如上示例加粗部分），确保标签值一致。

步骤 1：新建发布管理环境

步骤1 [登录软件开发生产线](#)。

步骤2 搜索[准备工作](#)中已创建的项目，单击项目名称，进入项目。

步骤3 在左侧导航栏选择“持续交付 > 发布管理”，进入环境列表页面。

步骤4 单击“新建环境”，弹出“新建环境”对话框，配置以下信息。

表 4-2 新建集群

参数项	说明
环境名称	项目下环境唯一标识，创建后不可修改，输入“灰度发布demo”。
资源类型	承载服务的资源类型，选择“CCE”。
发布用户	选择“当前用户”。
区域	选择 准备工作 中已购买的集群所在的区域。
集群	选择 准备工作 中已购买的集群。
环境级别	环境类型，选择“测试环境”。
描述	可选。输入“这是一个灰度发布测试环境”。

步骤5 单击“确定”，完成灰度发布测试环境的创建。

----结束

步骤 2：配置发布管理策略

步骤1 在环境列表页面，单击刚创建的环境名称，进入“环境信息”页面。

步骤2 单击“发布策略”页签，切换到“发布策略”页面。

步骤3 单击“自定义策略”旁的+，弹出“新建策略”对话框，选择系统提供的“灰度升级模板”。

步骤4 单击“确定”，然后逐步完成插件配置。

图 4-1 配置发布策略

基本信息

* 策略名称

基于Service的灰度蓝绿发布

策略描述

基于Service的灰度蓝绿发布

编排插件

灰度起负载

* 部署方式

镜像创建 YAML创建

* 命名空间

default

* 服务

nginx-service

自定义灰度版本号

* 灰度版本号

`\${TIMESTAMP}`

1. 灰度起负载

参数项	说明
部署方式	选择“镜像创建”，将创建与线上配置完全一致的工作负载，仅更新镜像包版本（业务代码及配置）。
命名空间	选择待升级的服务所在的命名空间。

参数项	说明
服务	待升级的服务，选择 准备工作中 已创建的服务。
自定义灰度版本号	开关关闭，系统将自动随机生成灰度版本号；开关开启，用户可以按需配置灰度版本号。这里选择开启开关。
灰度版本号	灰度版本号将作为新旧负载的引流标识，输入“\${TIMESTAMP}”，引用系统时间戳环境变量作为灰度版本号，例如：20230401095436。

- 基于所选Service关联的Deployment替换新镜像，创建新负载（灰度负载）。
 - 新负载内所有配置，如CPU、内存、副本数等，与旧负载（线上负载）完全一致。
 - 新负载不承接任何流量，可供开发人员进行测试，且对旧负载（线上负载）没有任何影响。
 - 新负载名称为“旧负载名称-灰度版本号”。
2. 灰度引流
选择“Service蓝绿引流”，基于Service将所有流量切换至新负载，完成引流后，旧负载不承接任何流量。
 3. 灰度下线
默认删除线上的旧负载，无需配置。
 4. 人工卡点
为了降低引流过程中的风险，并给运维、测试人员预留充足的测试时间，可以在灰度引流、灰度下线等较高危操作前，添加人工卡点进行验证。

图 4-2 配置人工卡点

人工卡点

* 超时处理

卡点失败，发布流终止执行 忽略结果，发布流继续执行

* 卡点时长

12 小时 00 分钟

卡点说明

引流前对灰度负载执行手工验证

参数项	说明
超时处理	卡点超时后处理方式，选择“卡点失败，发布流终止执行”，即超时未处理则终止执行。
卡点时长	卡点处理时长，可选择最长时间12小时。
卡点说明	自定义描述，如：引流前对灰度负载执行手工验证。

步骤5 单击“保存并应用”，即可将该策略设置为“使用中”策略，至此完成策略配置。

----结束

步骤 3：通过云原生发布插件进行发布

步骤1 单击顶部导航栏“首页”，返回软件开发生产线首页。

步骤2 找到**准备工作**中已创建的项目，单击项目名称，进入项目。

步骤3 在左侧导航栏选择“持续交付 > 流水线”，进入项目下流水线列表页面。

步骤4 搜索**准备工作**中已创建的流水线，单击“操作”列...，选择“编辑”，进入“任务编排”页面。

步骤5 在“阶段_1”单击默认任务，添加“云原生发布”插件并配置相关参数。

图 4-3 配置云原生发布插件



参数项	说明
名称	插件的名称，保持默认即可。
环境级别	任务发布的环境级别，选择“测试环境”。
环境	需要发布的环境，选择 步骤1：新建发布管理环境 中创建的环境“灰度发布demo”。
产物地址	部署时使用的镜像地址，可以使用“\${变量名}”通过环境变量引用构建产物，也可以直接输入产物地址。这里直接填写镜像中心提供的开源nginx镜像“nginx:latest”。

步骤6 配置完成后，单击“保存并执行”，开始执行流水线。

步骤7 单击运行中的“云原生发布”任务，切换至“任务结果”页签，可以查看当前发布策略执行过程。

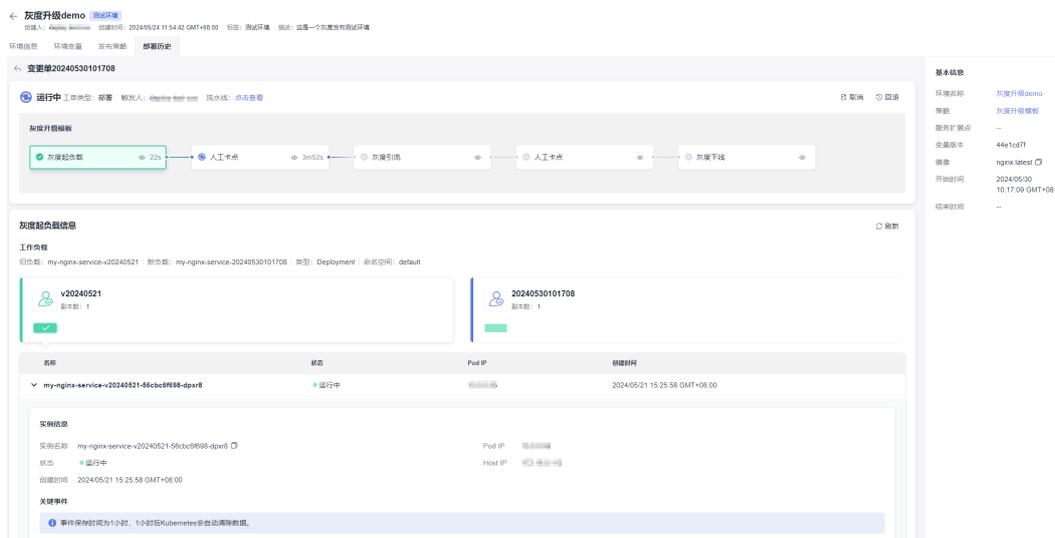
图 4-4 查看执行过程



步骤8 单击右上角“查看详情”，可以跳转到发布策略执行详情。

如下图所示，“灰度起负载”步骤执行成功，可以看到新旧负载的名称、状态、副本数量、集群事件等信息。此时新负载（灰度负载）已启动成功，但线上用户不感知，对线上服务无任何影响，可以在人工卡点的时限内，无风险地对灰度负载进行测试。

图 4-5 查看执行详情



步骤9 进入“人工卡点”确认阶段，进行人工确认。

- 若灰度验证发现问题，可单击“驳回”中断发布，并按实际情况决定是否需要进行回滚，恢复至发布前状态。
- 若验证无问题，可单击“批准”，执行下一步灰度引流。

图 4-6 人工卡点确认



步骤10 灰度引流完成后，所有流量都将被转发至新负载，旧负载不承接任何流量，进入旧负载下线前卡点确认阶段。

- 若验证发现问题，可单击“驳回”中断发布并单击“回滚”，快速将流量切换至旧负载，恢复服务状态。
- 若线上新负载验证无误，可单击“批准”，继续执行“灰度下线”阶段，删除旧负载。

步骤11 执行完成后，旧负载的副本数量变为0，新负载正常对外提供服务。

图 4-7 查看执行结果



说明

在发布完成后的任何时间发现线上服务出现问题，可以随时单击右上角“回滚”，恢复至此次发布之前的状态。

至此，基于Kubernetes原生Service的灰度蓝绿发布相关操作执行完成。

---结束

5 HE2E DevOps 实践：配置流水线

本文以“DevOps全流程示例项目”为例，介绍如何将代码检查、构建、部署任务串联起来，实现持续交付。

开展实践前，需要完成[部署](#)。

预置流水线简介

示例项目中预置以下5个流水线任务，可根据需要查看并使用。

表 5-1 预置流水线任务

预置流水线任务	任务说明
phoenix-workflow	基本的流水线任务。
phoenix-workflow-test	测试环境对应的流水线任务。
phoenix-workflow-work	Worker功能对应的流水线任务。
phoenix-workflow-result	Result功能对应的流水线任务。
phoenix-workflow-vote	Vote功能对应的流水线任务。

配置并执行流水线

一条流水线通常由多个阶段构成，每个阶段中可以添加多个子任务。

步骤1 配置流水线。

1. 进入“凤凰商城”项目，单击导航“持续交付 > 流水线”。
2. 找到流水线“phoenix-workflow”。单击图标**，选择“编辑”。
3. 添加代码检查阶段。

- a. 单击“代码源”与“构建”之间的 \oplus ，添加阶段。
- b. 单击“阶段_1”后的 \pencil ，在“编辑阶段”窗口中输入阶段名称“代码检查”，单击“确定”。

图 5-1 编辑阶段名称



- c. 单击“新建任务”。
- 在“新建任务”窗口中，单击“Check代码检查”插件后的“添加”。
- d. 选择调用任务“phoenix-codecheck-worker”，单击“确定”。

说明

代码检查任务有三种检查模式，本文保持默认值“Full”，可根据需要修改。

- Full：全量检查，扫描代码仓里的所有文件。
- Incremental (last commit)：增量检查，基于最近一次commit文件进行扫描。
- Incremental (last success)：增量检查，基于最近一次门禁通过后的变更文件进行扫描。

4. 配置部署任务。

单击部署任务名称，在窗口中选择关联构建任务“phoenix-sample-ci”，并检查配置项的值。

- 任务“phoenix-sample-standalone”的配置需与部署服务中同名任务的“参数设置”页面内容保持一致。
- 任务“phoenix-cd-cce”的配置需与部署服务中同名任务“参数设置”页面内容保持一致。

说明

部署任务中添加了两个部署任务，如果您在之前的步骤中只选择了一种部署方式，请保留对应的部署任务，将另一个删除。

5. 配置完成，单击“保存”。

步骤2 如果在**步骤1**配置了“phoenix-cd-cce”任务，进入云容器引擎服务。找到目标集群，单击集群名称进入总览页。

在导航中单击“工作负载”，选择“无状态负载”页签，确认列表中无记录。

如果列表中有记录，则勾选全部记录，单击“批量删除”，并在弹框中勾选所有选项，单击“是”，将列表记录清空。

步骤3 返回流水线列表页面，单击“phoenix-workflow”所在行的 \triangleright ，在滑出的窗口单击“执行”，启动流水线。

当页面中显示时，表示任务执行成功。

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

----结束

配置准出条件

为了控制代码的质量，代码必须经过扫描，并且错误数量控制在合理范围内，才允许发布。通过添加质量门禁可以有效的自动化控制流程。

- 步骤1** 在流水线任务“phoenix-workflow”详情页，单击“编辑”。
- 步骤2** 在阶段“代码检查”中，单击“准出条件”。
- 步骤3** 在“准出条件”窗口中，单击“标准策略准出条件”插件后的“添加”。
- 步骤4** 选择“系统策略”，单击“确定”。
- 步骤5** 单击“保存并执行”，启动流水线任务。

如果代码检查问题数未达到准出条件，流水线任务将执行失败。

----结束

配置代码变更自动触发流水线

通过以下配置，可实现代码变更自动触发流水线执行，从而实现项目的持续交付。

- 步骤1** 在流水线任务“phoenix-workflow”详情页，单击“编辑”。
- 步骤2** 选择“执行计划”页签，在“事件触发”目录下勾选“代码提交时触发”开关，在分支过滤下拉列表中勾选分支“master”，单击“保存”。
- 步骤3** 验证配置结果：修改代码并推送至master，即可查看流水线是否自动执行。

----结束