

流水线

# 用户指南

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。

# 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

---

|   |           |
|---|-----------|
| <b>1 流水线(CodeArts Pipeline)使用流程</b> ..... | <b>1</b>  |
| <b>2 开通并授权使用 CodeArts Pipeline</b> .....  | <b>3</b>  |
| <b>3 访问 CodeArts Pipeline 首页</b> .....    | <b>12</b> |
| <b>4 新建流水线</b> .....                      | <b>13</b> |
| 4.1 新建流水线（图形化）.....                       | 13        |
| 4.2 新建流水线（YAML 化）.....                    | 16        |
| <b>5 配置流水线</b> .....                      | <b>20</b> |
| 5.1 编排流水线阶段.....                          | 20        |
| 5.2 编排流水线任务.....                          | 25        |
| 5.3 配置流水线参数.....                          | 28        |
| 5.4 配置流水线执行计划.....                        | 34        |
| 5.5 配置流水线权限.....                          | 39        |
| 5.6 配置流水线通知订阅.....                        | 40        |
| <b>6 分组管理流水线</b> .....                    | <b>46</b> |
| <b>7 执行流水线</b> .....                      | <b>48</b> |
| <b>8 查看流水线</b> .....                      | <b>50</b> |
| <b>9 配置微服务变更流水线</b> .....                 | <b>54</b> |
| <b>10 管理流水线扩展插件</b> .....                 | <b>61</b> |
| 10.1 流水线插件概述.....                         | 61        |
| 10.2 流水线官方插件.....                         | 62        |
| 10.3 图形化新建流水线自定义插件.....                   | 64        |
| 10.4 代码化新建流水线自定义插件.....                   | 72        |
| 10.5 执行镜像命令.....                          | 82        |
| <b>11 新建 CodeArts 服务扩展点</b> .....         | <b>84</b> |
| <b>12 查看审计日志</b> .....                    | <b>93</b> |
| <b>13 参考</b> .....                        | <b>94</b> |
| 13.1 流水线上下文.....                          | 94        |
| 13.1.1 流水线上下文.....                        | 94        |

|   |            |
|---|------------|
| 13.1.2 配置流水线执行条件表达式.....                  | 99         |
| 13.1.3 配置通过流水线上下文获取构建任务的产物信息.....         | 102        |
| 13.2 YAML 语法规范.....                       | 104        |
| 13.2.1 流水线触发方式 on.....                    | 104        |
| 13.2.2 流水线参数 env.....                     | 108        |
| 13.2.3 流水线任务 jobs.....                    | 108        |
| <b>14 发布管理(CodeArts Release)使用指南.....</b> | <b>114</b> |
| 14.1 发布管理概述.....                          | 114        |
| 14.2 新建发布管理环境.....                        | 115        |
| 14.3 配置发布管理环境变量.....                      | 117        |
| 14.4 配置发布管理发布策略.....                      | 120        |
| 14.5 通过云原生发布插件进行发布.....                   | 126        |
| 14.6 查看发布管理发布结果.....                      | 128        |

# 1 流水线(CodeArts Pipeline)使用流程

---

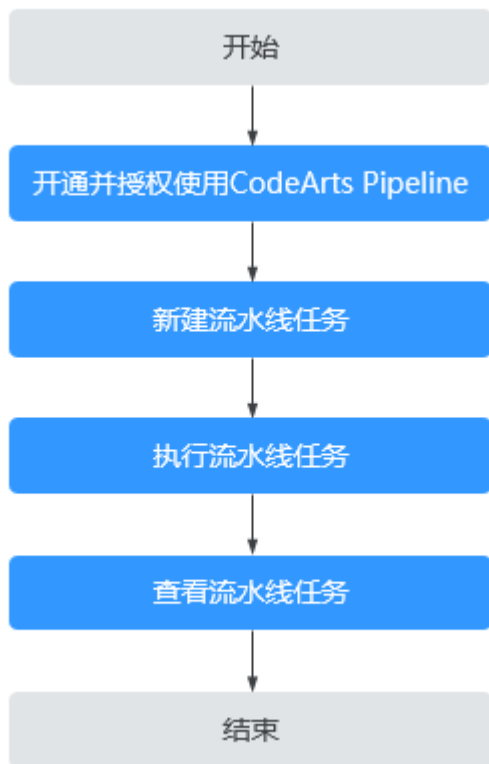
流水线服务本质上是一个可视化的自动化任务调度平台，需配合CodeArts中[编译构建](#)、[代码检查](#)、[测试计划](#)、[部署](#)等服务的自动化任务使用。

根据实际需要，如开发测试环境应用部署、生产环境应用部署等，对这些自动化任务进行自定义编排，一次配置后就可以一键自动化触发调度执行，避免频繁低效的手工操

在[软件开发生产线](#)解决方案中，流水线属于其中一个子服务，具体位置可参考[产品架构](#)

## 流水线服务基本操作流程

图 1-1 流水线服务基本操作流程用例



# 2 开通并授权使用 CodeArts Pipeline

## 前提条件

已[注册华为账号并开通华为云](#)。

## 开通流水线服务

流水线不支持单独购买，需配合CodeArts其他服务使用，开通任意CodeArts套餐即可使用。

**步骤1** 进入[流水线控制台](#)。

**步骤2** 页面提示前往购买CodeArts套餐，单击“购买”。

**步骤3** 进入购买开通CodeArts页面，请根据需要购买相应套餐，购买开通方式请参考[购买CodeArts](#)。

----结束

## 授权流水线服务

流水线权限从上到下可以分成三个层级，分别控制不同模块不同粒度的用户行为。

表 2-1 流水线权限

| 层级                    | 模块                       | 描述   |
|-----------------------|--------------------------|--|
| <a href="#">租户级权限</a> | 扩展插件、策略（租户级）、租户级规则、流水线模板 | 控制租户账号下的模块资源，在统一身份认证服务中进行配置，生效范围为整个租户下的全部项目。 |
| <a href="#">项目级权限</a> | 流水线、策略（项目级）、微服务、变更、环境    | 控制某一具体项目下的模块资源，在项目中配置，生效范围为整个项目下的全部资源。       |
| <a href="#">资源级权限</a> | 流水线任务                    | 控制某一具体流水线任务的操作权限，在流水线任务中进行配置，生效范围为当前流水线任务。   |

● 租户级权限

通过统一身份认证服务，为指定用户配置流水线服务中租户级策略、租户级规则、扩展插件及流水线模板的权限。

- 使用租户账号或其他有权限的账号登录CodeArts。
- 单击右上角的用户名，选择“子账号管理”，进入统一身份认证服务控制台。
- 单击左侧导航“用户组”，进入用户组设置页面，新建一个用户组或选择已有用户组，单击“授权”，开始对用户组授权。

图 2-1 授权用户组



云服务选择“流水线服务”，即可查看“流水线服务”相关策略列表，包括以下四个策略：

表 2-2 流水线系统策略

| 策略名称   | 描述  |
|--|---|
| CloudPipeline Tenant Rules FullAccess          | CodeArts租户规则全部权限，控制用户是否具有流水线服务租户级规则的全部权限。<br><ul style="list-style-type: none"> <li>规则权限对应统一身份认证服务中的“cloudpipeline:rule:update”，管理员可以通过“CloudPipeline Tenant Rules FullAccess”系统策略或自定义策略对用户进行授权。</li> <li>普通用户可以查看全部租户级规则，有权限的用户可以查看和管理全部租户级规则。</li> </ul>                  |
| CloudPipeline Tenant Rule Templates FullAccess | CodeArts租户策略全部权限，控制用户是否具有流水线服务租户级策略的全部权限。<br><ul style="list-style-type: none"> <li>策略权限对应统一身份认证服务中的“cloudpipeline:ruletemplate:update”，管理员可以通过“CloudPipeline Tenant Rule Templates FullAccess”系统策略或自定义策略对用户进行授权。</li> <li>普通用户可以查看全部租户级策略，有权限的用户可以查看和管理全部租户级策略。</li> </ul> |



| 策略名称   | 描述  |
|--|---|
| CloudPipeline Tenant Extensions FullAccess         | CodeArts租户扩展插件全部权限，控制用户是否具有流水线服务扩展插件的全部权限。 <ul style="list-style-type: none"> <li>扩展插件权限对应统一身份认证服务中的“cloudpipeline:extensions:update”，管理员可以通过“CloudPipeline Tenant Extensions FullAccess”系统策略或自定义策略对用户进行授权。</li> <li>普通用户可以查看租户下全部插件，有权限的用户可以查看和管理租户下全部插件。</li> </ul>                                       |
| CloudPipeline Tenant Pipeline Templates FullAccess | CodeArts租户流水线模板全部权限，控制用户是否具有流水线服务流水线模板的全部权限。 <ul style="list-style-type: none"> <li>流水线租户模板权限对应统一身份认证服务中的“cloudpipeline:pipeline:template:update”，管理员可以通过“CloudPipeline Tenant Pipeline Templates FullAccess”系统策略或自定义策略对用户进行授权。</li> <li>普通用户可以创建模板、查看全部模板，但只能对自己创建的模板进行管理，有权限的用户可以查看和管理租户下全部模板。</li> </ul> |


- d. 勾选所需策略，单击“下一步”，设置最小授权范围，完成对用户组的授权。
- e. 最后，将指定用户加入用户组，即可完成对用户的授权。

#### 说明

除了使用系统策略，租户还可以通过[创建自定义策略](#)的方式进行授权。

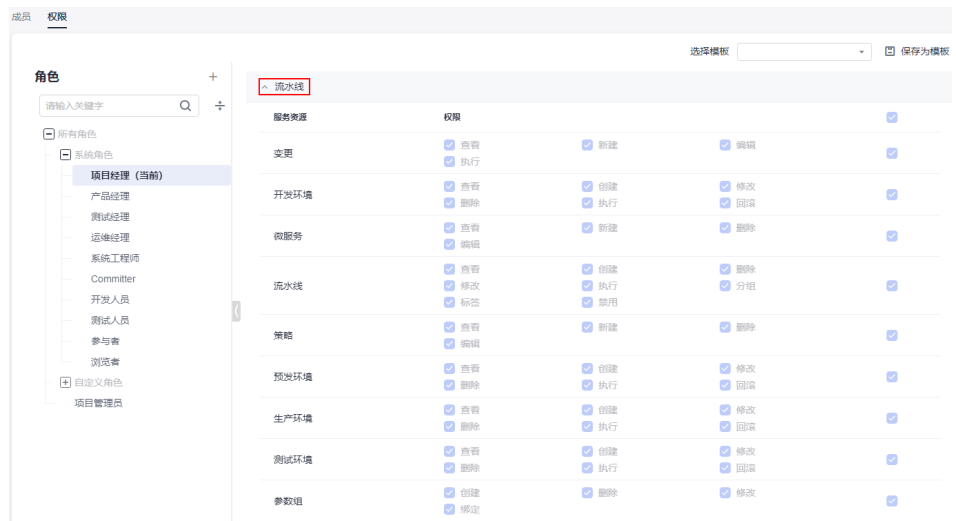
- **项目级权限**

CodeArts支持在项目下统一配置项目各角色对当前项目下流水线资源的默认操作权限。

- a. [登录华为云控制台页面](#)。
- b. 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”。
- c. 单击“立即使用”，进入流水线服务首页。
- d. 单击顶部导航“首页”，进入CodeArts首页。
- e. 单击项目名称，进入项目。
- f. 单击左侧导航“设置 > 通用设置”，进入“服务权限管理”页面。

流水线相关资源集中在流水线服务下，包括流水线、策略（项目级）、微服务、变更、环境、参数组。

图 2-2 流水线项目级权限



**说明**

默认情况下，有编辑或执行权限就有查看权限。

**流水线权限**

初始状态下，项目下各角色具有的流水线权限如下表所示：

表 2-3 项目级流水线权限

| 角色        | 查看 | 创建 | 执行 | 修改 | 删除 | 分组 |
|-----------|----|----|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  | √  | √  |
| 开发人员      | √  | √  | √  | ×  | ×  | ×  |
| 测试经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 测试人员      | √  | ×  | ×  | ×  | ×  | ×  |
| 参与者       | √  | ×  | ×  | ×  | ×  | ×  |
| 浏览者       | √  | ×  | ×  | ×  | ×  | ×  |
| 运维经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 产品经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 系统工程师     | √  | √  | √  | √  | √  | √  |
| Committer | √  | √  | √  | ×  | ×  | ×  |

- 复制流水线需要同时具有创建流水线及修改源流水线的权限。
- 流水线的角色权限默认继承项目下各角色的流水线权限，并与项目下各角色的流水线权限保持联动，直到用户在流水线中修改了该流水线的角色权限为止。
- 流水线创建者默认拥有其创建的流水线的全部权限。

### 策略权限

初始状态下，项目下各角色具有的项目级策略权限如下表所示：

表 2-4 项目级策略权限

| 角色        | 查看 | 创建 | 修改 | 删除 |
|-----------|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  |
| 开发人员      | √  | √  | √  | √  |
| 测试经理      | √  | ×  | ×  | ×  |
| 测试人员      | √  | ×  | ×  | ×  |
| 参与者       | √  | ×  | ×  | ×  |
| 浏览者       | √  | ×  | ×  | ×  |
| 运维经理      | √  | ×  | ×  | ×  |
| 产品经理      | √  | ×  | ×  | ×  |
| 系统工程师     | √  | √  | √  | √  |
| Committer | √  | √  | √  | √  |

其中，复制策略需要同时具有创建策略及修改源策略的权限。

### 微服务权限

初始状态下，项目下各角色具有的微服务权限如下表所示：

表 2-5 项目级微服务权限

| 角色    | 查看 | 创建 | 修改 | 删除 |
|-------|----|----|----|----|
| 项目创建者 | √  | √  | √  | √  |
| 项目经理  | √  | √  | √  | √  |
| 开发人员  | √  | ×  | ×  | ×  |
| 测试经理  | √  | ×  | ×  | ×  |
| 测试人员  | √  | ×  | ×  | ×  |
| 参与者   | √  | ×  | ×  | ×  |

| 角色        | 查看 | 创建 | 修改 | 删除 |
|-----------|----|----|----|----|
| 浏览者       | √  | ×  | ×  | ×  |
| 运维经理      | √  | ×  | ×  | ×  |
| 产品经理      | √  | ×  | ×  | ×  |
| 系统工程师     | √  | √  | √  | √  |
| Committer | √  | ×  | ×  | ×  |

### 变更权限

初始状态下，项目下各角色具有的变更权限如下表所示：

表 2-6 项目级变更权限

| 角色        | 查看 | 创建 | 修改 | 执行 |
|-----------|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  |
| 开发人员      | √  | √  | √  | √  |
| 测试经理      | √  | ×  | ×  | ×  |
| 测试人员      | √  | ×  | ×  | ×  |
| 参与者       | √  | ×  | ×  | ×  |
| 浏览者       | √  | ×  | ×  | ×  |
| 运维经理      | √  | ×  | ×  | ×  |
| 产品经理      | √  | ×  | ×  | ×  |
| 系统工程师     | √  | √  | √  | √  |
| Committer | √  | √  | √  | √  |

### 环境权限

初始状态下，项目下各角色具有的发布管理不同环境权限如下表所示。

表 2-7 项目级开发环境权限

| 角色    | 查看 | 创建 | 修改 | 删除 | 执行 | 回滚 |
|-------|----|----|----|----|----|----|
| 项目创建者 | √  | √  | √  | √  | √  | √  |
| 项目经理  | √  | √  | √  | √  | √  | √  |

| 角色        | 查看 | 创建 | 修改 | 删除 | 执行 | 回滚 |
|-----------|----|----|----|----|----|----|
| 开发人员      | √  | √  | √  | √  | √  | √  |
| 测试经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 测试人员      | √  | ×  | ×  | ×  | ×  | ×  |
| 参与者       | √  | ×  | ×  | ×  | ×  | ×  |
| 浏览者       | √  | ×  | ×  | ×  | ×  | ×  |
| 运维经理      | √  | √  | √  | √  | √  | √  |
| 产品经理      | √  | √  | √  | √  | √  | √  |
| 系统工程师     | √  | √  | √  | √  | √  | √  |
| Committer | √  | √  | √  | √  | √  | √  |

表 2-8 项目级测试环境权限

| 角色        | 查看 | 创建 | 修改 | 删除 | 执行 | 回滚 |
|-----------|----|----|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  | √  | √  |
| 开发人员      | √  | ×  | ×  | ×  | ×  | ×  |
| 测试经理      | √  | √  | √  | √  | √  | √  |
| 测试人员      | √  | √  | √  | √  | √  | ×  |
| 参与者       | √  | ×  | ×  | ×  | ×  | ×  |
| 浏览者       | √  | ×  | ×  | ×  | ×  | ×  |
| 运维经理      | √  | √  | √  | √  | √  | √  |
| 产品经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 系统工程师     | √  | ×  | ×  | ×  | ×  | ×  |
| Committer | √  | √  | √  | √  | √  | √  |

表 2-9 项目级预发环境权限

| 角色        | 查看 | 创建 | 修改 | 删除 | 执行 | 回滚 |
|-----------|----|----|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  | √  | √  |
| 开发人员      | √  | ×  | ×  | ×  | ×  | ×  |
| 测试经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 测试人员      | √  | ×  | ×  | ×  | ×  | ×  |
| 参与者       | ×  | ×  | ×  | ×  | ×  | ×  |
| 浏览者       | ×  | ×  | ×  | ×  | ×  | ×  |
| 运维经理      | √  | √  | √  | √  | √  | √  |
| 产品经理      | √  | ×  | ×  | ×  | ×  | ×  |
| 系统工程师     | √  | ×  | ×  | ×  | ×  | ×  |
| Committer | √  | √  | √  | √  | √  | √  |

表 2-10 项目级生产环境权限

| 角色        | 查看 | 创建 | 修改 | 删除 | 执行 | 回滚 |
|-----------|----|----|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  | √  | √  |
| 开发人员      | ×  | ×  | ×  | ×  | ×  | ×  |
| 测试经理      | ×  | ×  | ×  | ×  | ×  | ×  |
| 测试人员      | ×  | ×  | ×  | ×  | ×  | ×  |
| 参与者       | ×  | ×  | ×  | ×  | ×  | ×  |
| 浏览者       | ×  | ×  | ×  | ×  | ×  | ×  |
| 运维经理      | √  | √  | √  | √  | √  | √  |
| 产品经理      | ×  | ×  | ×  | ×  | ×  | ×  |
| 系统工程师     | √  | ×  | ×  | ×  | ×  | ×  |
| Committer | √  | √  | √  | √  | √  | √  |

### 参数组权限

初始状态下，项目下各角色具有的参数组权限如下表所示：

表 2-11 项目级参数组权限

| 角色        | 创建 | 删除 | 修改 | 绑定 |
|-----------|----|----|----|----|
| 项目创建者     | √  | √  | √  | √  |
| 项目经理      | √  | √  | √  | √  |
| 开发人员      | √  | √  | √  | √  |
| 测试经理      | ×  | ×  | ×  | ×  |
| 测试人员      | ×  | ×  | ×  | ×  |
| 参与者       | ×  | ×  | ×  | ×  |
| 浏览者       | ×  | ×  | ×  | ×  |
| 运维经理      | ×  | ×  | ×  | ×  |
| 产品经理      | ×  | ×  | ×  | ×  |
| 系统工程师     | √  | √  | √  | √  |
| Committer | √  | √  | √  | √  |

- **资源级权限**

支持对单条流水线按照角色和人员两种维度进行权限配置，配置方法请参考[配置流水线权限](#)。

#### 角色权限配置

- 项目创建者、流水线创建者和项目经理可以修改流水线的角色权限。
- 默认情况下，流水线的角色权限和项目下角色的流水线权限保持一致，且会在项目下角色的流水线权限变动时联动更新。
- 用户修改流水线的角色权限后，该流水线的角色权限将不再与项目下角色的流水线权限联动，且以流水线的角色权限为准，即流水线的角色权限优先级高于项目下角色的流水线权限。

#### 人员权限配置


- 项目创建者、流水线创建者和项目经理可以修改流水线的人员权限。
- 默认情况下，流水线的人员权限和角色权限保持一致，且会在流线下角色权限变动时联动更新。
- 用户修改流水线的人员权限后，该流水线的人员权限将不再与角色权限联动，且以流水线的人员权限为准，即流水线的人员权限优先级高于角色权限。

# 3 访问 CodeArts Pipeline 首页

介绍进入流水线服务的两种方式：首页入口和项目入口。

## 首页入口

**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，即可开始使用流水线服务。

单击页面左上方 ，可根据需要切换区域。

----结束

## 项目入口

**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“首页”，进入CodeArts首页。

**步骤5** 单击项目名称，进入项目。

**步骤6** 单击左侧导航“持续交付 > 流水线”，进入项目下流水线任务列表，即可开始使用流水线服务。

单击页面左上角 ，可根据需要切换区域。

----结束



# 4 新建流水线

## 4.1 新建流水线（图形化）

### 准备工作

- [新建项目](#)。
- 使用CodeArts Repo流水线源：[新建代码仓库](#)。
- [新建CodeArts服务扩展点](#)：用于提升CodeArts Repo代码库的操作权限，或连接非CodeArts的第三方代码库。

### 新建流水线

**步骤1** [访问CodeArts Pipeline首页](#)。

**步骤2** 单击“新建流水线”，进入“基本信息”页面，参考[表4-1](#)配置流水线基本信息。

表 4-1 流水线基本信息参数说明

| 参数项  | 说明   |
|------|--|
| 名称   | 流水线的名称，自定义。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 所属项目 | 流水线归属项目。 <ul style="list-style-type: none"><li>• 从流水线服务首页入口新建流水线，请根据实际需要选择已创建的项目。</li><li>• 从项目下流水线入口新建流水线，所属项目默认为当前项目，不可更改。</li></ul> |

| 参数项          | 说明   |
|--------------|--|
| 流水线源         | <p>流水线关联的源类型：</p> <ul style="list-style-type: none"><li>● CodeArts代码源<br/>Repo：提供企业代码托管的全方位服务，为软件开发者提供基于Git的在线代码托管服务。</li><li>● 第三方代码源<ul style="list-style-type: none"><li>- 码云：连接码云账号之后可以获取该账号下的仓库、分支等信息。</li><li>- Github：连接Github账号后可以获取该账号下的仓库、分支等信息。</li><li>- Gitcode：连接Gitcode账号后可以获取该账号下的仓库、分支等信息。</li><li>- 通用Git：连接第三方通用Git仓库后可以获取该账号下的仓库、分支等信息。</li></ul></li><li>● 制品源<br/>SWR：使用华为云容器镜像作为流水线源运行流水线，并生成制品名称、制品下载地址和制品版本号系统参数，供需要的任务插件使用。</li></ul> <p><b>说明</b></p> <ul style="list-style-type: none"><li>● GitCode流水线源当前仅支持“北京四”和“广州”局点白名单用户。如果您需要开通此功能，请新建工单或拨打客服热线进行咨询。</li><li>● 如果用户新建的流水线任务不需要关联代码仓，也可以选择“暂不选择”，不关联代码仓时，如果添加了需要关联代码仓的任务，执行时会提示错误，具体问题请参考<a href="#">常见问题</a>。</li></ul> |
| 创建方式         | <p>关联Repo代码源时，可以选择图形化编排和YAML化编排两种方式。</p> <ul style="list-style-type: none"><li>● 图形化编排：以图形化界面方式轻松配置流水线，清晰展示任务串并行关系。</li><li>● YAML化编排：以代码化YAML方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。请参考<a href="#">新建流水线（YAML化）</a>。</li></ul>  |
| 服务扩展点        | <p>关联第三方代码源时，需通过服务扩展点实现与对应仓库的连接。可选择<a href="#">准备工作中</a>提前创建的扩展点，也可单击“新建服务扩展点”进行创建，具体操作请参考<a href="#">新建CodeArts服务扩展点</a>。</p>   |
| 代码仓          | <p>流水线关联的代码仓库。</p>   |
| 默认分支         | <p>仓库默认分支，手工或定时执行流水线时默认使用的分支。</p>  |
| Repo HTTPS授权 | <p>关联Repo代码源时，可配置授权扩展点以提升代码库操作权限，当前主要用于流水线服务的微服务变更功能模块及其相关插件。可选择<a href="#">准备工作中</a>提前创建的扩展点，也可单击“新建授权”进行创建，具体操作请参考<a href="#">新建CodeArts服务扩展点</a>。</p>   |
| 别名           | <p>仓库别名，设置别名后可基于别名生成仓库对应的系统参数，如：“别名_REPOSITORY_NAME”，表示仓库名称。生成的参数可在参数设置页面查看，并在流水线中通过“\${参数名}”进行引用。</p> <p>支持输入大小写英文字母、数字、“_”，不超过128个字符。</p>  |
| 描述           | <p>不超过1024个字符。</p>   |

| 参数项   | 说明   |
|-------|--|
| 组织    | 关联SWR流水线源时，选择SWR组织。组织用于隔离镜像，每个组织可对应一个公司或部门，将其拥有的镜像集中在该组织下。   |
| 镜像名   | 关联SWR流水线源时，选择组织下的镜像。   |
| 指定版本  | 关联SWR流水线源时，选择镜像的版本。  |
| 制品源别名 | 关联SWR流水线源时，需设置制品源别名，设置别名后可基于别名生成对应的系统参数，如：“别名_ARTIFACT_NAME”，表示制品名称。生成的参数可在参数设置页面查看，并在流水线中通过“\${参数名}”进行引用。<br>支持输入中文、大小写英文字母、数字、“-”、“_”、“.”，不超过128个字符。 |

**步骤3** 配置完基本信息，单击“下一步”，进入“选择模板”页面。

- 您可以选择系统模板或自定义模板，在模板基础上快速新建流水线，并自动生成模板预置的任务，模板来源请参考[管理流水线模板](#)。
- 也可以选择“空模板”，即不使用模板，新建空任务流水线。

**步骤4** 选择完模板，单击“确定”，完成流水线创建。

页面自动进入流水线“任务编排”页面，您可以继续[配置流水线](#)，也可以单击“取消”返回流水线列表。

----结束




## 管理流水线模板

流水线服务预置了一系列常用系统模板，同时支持创建自定义模板，帮助您基于模板快速创建持续交付流水线，在企业内完成复用，标准化企业发布流程。

- 进入流水线模板列表**
  - 首页入口：进入流水线服务首页，默认展示“流水线列表”页，单击“流水线模板”，进入“流水线模板”页。
  - 项目入口：进入项目下流水线，单击右上角“更多 > 模板管理”，进入“流水线模板”页。

模板列表操作项说明如下：

**表 4-2** 模板列表操作项

| 操作项   | 说明   |
|---|--|
|  | 生成流水线。单击后进入“新建流水线”页面，可快速生成对应模板的流水线。  |
|  | 收藏模板。单击可收藏模板，收藏后图标变为  ，再单击可以取消收藏。 |

| 操作项 | 说明  |
|-----|---|
| ... | <ul style="list-style-type: none"><li>单击后选择“编辑”，进入“任务编排”页面，可编辑模板。</li><li>单击后选择“复制”，进入“任务编排”页面，可复制模板。</li><li>单击后选择“删除”，可根据提示删除模板。</li></ul> <p><b>说明</b><br/>系统模板只能用来复制或生成流水线，不能编辑和删除。</p> |

- **新建流水线自定义模板**

- 进入流水线模板列表页。
- 单击“新建流水线模板”，进入“任务编排”页。
- 根据需要配置模板基本信息、模板阶段/任务、模板参数等。
  - **基本信息**：可编辑模板的名称（必填）、语言（支持Java、Python、Node.js、Go、.Net、C++、PHP，默认选择“无”）、模板描述信息。
  - **任务编排**：流水线模板目前可以添加流水线阶段及部分插件，将构建、代码检查、部署、接口测试等任务配置在模板中后，通过模板创建流水线的时候会同步创建相应类型的任务。

**说明**

- 模板中无需配置流水线源。
  - 模板编排暂不支持配置阶段准入。
  - **参数设置**：可根据需要为模板添加参数，流水线模板参数包括自定义参数和系统预定义参数两种，其中，自定义参数类型包括：字符串类型、枚举类型、自增长类型，配置方法请参见[配置流水线自定义参数](#)。
- 单击“保存”，完成模板创建。

## 4.2 新建流水线（YAML化）

### 准备工作

- [新建项目](#)。
- [新建代码仓库](#)。
- 准备YAML文件。

通过YAML化方式创建流水线，可以在新建过程中创建一个新的YAML文件，也可以提前编排好YAML文件。YAML化流水线通常由触发方式on、参数env、任务jobs等组成，详见[YAML语法规则](#)。

### YAML 文件示例

为了让您快速理解YAML化流水线，以下通过流水线串联编译构建任务、代码检查任务、部署任务，并在编译构建任务中引用流水线参数的YAML文件为您介绍YAML整体结构。

```
env: # 以键值对的形式定义环境变量，定义后可在流水线的任意任务中引用
  image_version: 1.0.0

jobs: # 定义组成流水线的任务
```

```
build: # 任务ID, 定义任务的唯一标识
  name: maven build # 任务的名称, 在图形化界面中会展示该名称
  steps: # 定义组成任务的步骤
    - name: My build step # 步骤的名称, 在图形化界面中会展示该名称
      uses: CodeArtsBuild # 指定步骤运行的插件
      with: # 以键值对的形式定义插件的运行参数, 可引用env中定义的变量。
        jobId: 878b4d13cb284d9e8f33f988a902f57c # 任务id。获取方法: 在任务详情页, 拷贝浏览器URL末尾的32位数字、字母组合的字符串, 即为任务ID。
        artifactIdentifier: my_image
        version: ${ env.image_version }
  check:
    name: code check
    steps:
      - name: My check step
        uses: CodeArtsCheck
        with:
          jobId: 43885d46e13d4bf583d3a648e9b39d1e
          checkMode: full
  deploy:
    name: cce deploy
    needs: # 使任务在指定的任务结束后才开始运行
      - build
      - check
    if: ${ completed() } # 指定任务的运行条件
    steps:
      - name: My deploy step
        uses: CodeArtsDeploy
        with:
          jobId: 9c5a5cda6ffa4ab583380f5a014b2b31
          version: ${ env.image_version }
```

## 代码化新建流水线

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 单击“新建流水线”，进入“基本信息”页面，参考[表4-3](#)配置流水线基本信息。

表 4-3 流水线基本信息参数说明

| 参数项  | 说明   |
|------|--|
| 名称   | 流水线的名称，自定义。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 所属项目 | 流水线归属项目。 <ul style="list-style-type: none"><li>从流水线服务首页入口新建流水线，请根据实际需要选择已创建的项目。</li><li>从项目下流水线入口新建流水线，所属项目默认为当前项目，不可更改。</li></ul> |
| 流水线源 | 选择“Repo”：即CodeArts Repo，提供企业代码托管的全方位服务，为软件开发者提供基于Git的在线代码托管服务。<br><b>说明</b><br>YAML化流水线仅支持使用CodeArts Repo代码仓。                      |
| 创建方式 | 选择“YAML化编排”：以代码化YAML方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。  |
| 代码仓  | 流水线关联的代码仓库。  |
| 默认分支 | 仓库默认分支，手工或定时执行流水线时默认使用的分支。   |

| 参数项          | 说明   |
|--------------|--|
| 配置方式         | <ul style="list-style-type: none"><li>新建yaml文件：创建一个新的YAML文件。</li><li>已有yaml文件：基于已有YAML文件进行编排，编排后的内容会覆盖原YAML文件。YAML文件编写请参考<a href="#">YAML语法规则</a>。</li></ul> |
| 选择yaml文件     | 配置方式选择“已有yaml文件”时，需设置此参数。<br>选择提前准备好并存储在代码仓库的YAML文件：选择对应分支并输入YAML文件的相对路径。  |
| Repo HTTPS授权 | 配置授权扩展点以提升代码库操作权限，当前主要用于流水线服务的微服务变更功能模块及其相关插件。可选择 <a href="#">准备工作中</a> 提前创建的扩展点，也可单击“新建授权”进行创建，具体操作请参考 <a href="#">新建CodeArts服务扩展点</a> 。                    |
| 别名           | 仓库别名，设置别名后可基于别名生成仓库对应的系统参数，如：“别名_REPOSITORY_NAME”，表示仓库名称。生成的参数可在参数设置页面查看，并在流水线中通过“\${参数名}”进行引用。  |
| 描述           | 不超过1024个字符。  |

**步骤3** 配置完基本信息，单击“确定”，进入“任务编排”页面。

- 左侧为YAML编辑区域，请参考[YAML语法规则](#)编写YAML。
- 右侧为流水线插件列表，请根据需要添加步骤yaml。

编排过程支持语法规则校验；单击“预览”，可切换至图形化界面预览效果。

**步骤4** 编排完任务，单击“保存”，填写提交信息，选择推送分支。

### 保存 ×

\* Commit Message

  
 推送到“master”分支  
 推送到新分支

确定取消

- 选项一表示推送到已有分支：如果是“新建yaml文件”方式新建流水线，已有分支为默认分支；如果是基于“已有yaml文件”方式新建流水线，已有分支则为YAML文件所在的分支。
- 选项二表示推送到新分支：基于上述已有分支创建一个新分支，勾选“创建合并请求”，则会创建一个从新分支到已有分支的合并请求。

**步骤5** 单击“确定”，完成流水线创建。

----结束

# 5 配置流水线

## 5.1 编排流水线阶段

阶段是流水线上的基本组成部分，主要用于对流水线纳管的任务做业务流程上的分类编排管理，方便聚合管理业务关联性强的执行任务，让流水线上的业务流更加直观。





### 编排流水线阶段

**步骤1** 访问CodeArts Pipeline首页。


**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，单击 $\oplus$ 或“新建阶段”，可以为流水线添加新的阶段，添加完阶段后，可根据实际需要编辑、复制、删除、移动阶段、配置阶段准入类型。

表 5-1 管理阶段

| 操作项  | 说明   |
|------|--|
| 编辑阶段 | 单击  ，弹出“编辑阶段”侧滑框，可以配置阶段名称和阶段是否总是运行。 <ul style="list-style-type: none"><li>总是运行：选择“是”，表示流水线执行时，该阶段下的任务默认选中必须执行且不可取消。</li><li>总是运行：选择“否”，表示流水线执行时，该阶段下的任务默认选中但可以取消。</li></ul> |
| 复制阶段 | 单击  ，可以复制流水线阶段。   |
| 删除阶段 | 单击  ，根据删除提示确认是否删除阶段。  |
| 排序阶段 | 单击  不松开，可以移动阶段调整阶段顺序。   |



| 操作项  | 说明   |     |    |    |      |    |    |
|------|--|-----|----|----|------|----|----|
| 阶段准入 | <p>通过阶段准入校验，控制流水线阶段的执行。单击，弹出“准入设置”窗口，可以配置阶段准入类型。</p> <ul style="list-style-type: none"> <li>● 自动：默认执行方式，流水线执行时自动进入下一阶段继续运行。</li> <li>● 手动：流水线执行时需手动确认后才能进入下一阶段继续运行。</li> <li>● 时间窗：流水线执行时需配置好的时间段才能进入下一阶段继续执行。</li> </ul>   |     |    |    |      |    |    |
| 准出条件 | <p>流水线服务提供基于规则和策略的准出条件管理能力，是指在流水线阶段中配置的用于自动化看护的质量指标卡点，通过配置规则和策略，来实现阶段准出校验，并最终控制流水线的执行。</p> <ul style="list-style-type: none"> <li>● 规则：基于扩展插件的输出阈值，设置比较关系和阈值条件，供策略使用，并最终应用到流水线准出条件中，作为准出条件的判断依据。配置方法请参考<a href="#">配置流水线规则</a>。</li> <li>● 策略：策略是一系列规则的集合，每个规则对应了一个步骤插件的输出指标值的条件模板，通过预定义策略，可以很方便地在多条不同流水线中应用同样的准出条件。策略实行分层管理，可分为<a href="#">租户级策略</a>和<a href="#">项目级策略</a>。用户可根据需要将策略资源应用到流水线作为准出条件来控制流水线的执行，保障产品交付质量。</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>● 当前仅支持“标准策略准出条件”，可以选择当前项目或租户下创建好的策略。</li> <li>● 流水线可以在各阶段单独设置准出条件，准出条件只对当前阶段有效。</li> <li>● 同一阶段中可以配置多个不同的准出条件。</li> </ul> <ol style="list-style-type: none"> <li>1. 单击阶段下方“准出条件”，弹出“准出条件”侧滑框，将鼠标移动到准出条件卡片，单击“添加”。</li> <li>2. 填写准出条件名称，选择策略。           <div data-bbox="603 1305 1469 1776" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>标准策略准出条件</b><br/>选择一个标准插件策略用于门禁拦截。</p> <p>* 名称<br/> <input type="text" value="标准策略准出条件"/></p> <p>* 策略<br/> <input style="float: right; color: #00aaff; font-size: small; margin-left: 10px;" type="text" value="系统策略"/>策略管理</p> <p>CodeArtsCheck问题数</p> <p>CodeArtsCheck问题数</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">检查项</th> <th style="width: 20%;">比较</th> <th style="width: 20%;">阈值</th> </tr> </thead> <tbody> <tr> <td>问题总数</td> <td style="text-align: center;">&lt;=</td> <td style="text-align: center;">10</td> </tr> </tbody> </table> </div> </li> <li>3. 单击“确定”，即可添加一个准出条件。</li> </ol> | 检查项 | 比较 | 阈值 | 问题总数 | <= | 10 |
| 检查项  | 比较   | 阈值  |    |    |      |    |    |
| 问题总数 | <=   | 10  |    |    |      |    |    |

**步骤4** 阶段编排完成后，保存流水线。

----结束

## 配置流水线规则

规则是一种租户级别的资源，可在当前租户下所有的租户级策略或项目级策略中使用。

**步骤1** 访问CodeArts Pipeline首页。

**步骤2** 单击右上角头像，选择“租户设置”。

**步骤3** 单击左侧导航“策略管理 > 规则”，进入规则列表页面。

**步骤4** 单击“新建规则”，进入“新建规则”页面，填写参数信息。

图 5-1 新建规则



| 检查项  | 关系 | 默认阈值 | 阈值开关                                |
|------|----|------|-------------------------------------|
| 致命   | =  | 数值 0 | <input checked="" type="checkbox"/> |
| 严重   | =  | 数值 0 | <input checked="" type="checkbox"/> |
| 一般   | =  | 数值 0 | <input checked="" type="checkbox"/> |
| 提示   | =  | 数值 0 | <input checked="" type="checkbox"/> |
| 问题总数 | =  | 数值 0 | <input checked="" type="checkbox"/> |

表 5-2 规则参数说明

| 参数项  | 说明  |
|------|---|
| 名称   | 规则名称，默认根据当前时间生成。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 类型   | 规则类型，与插件类型对应。支持构建、代码检查、测试三种类型插件。 <ul style="list-style-type: none"> <li>构建：用于代码构建的执行插件。</li> <li>代码检查：用于检查代码质量的插件。</li> <li>测试：用于测试接口性能的插件。</li> </ul>  |
| 选择插件 | 插件类型下的所有插件。 <ul style="list-style-type: none"> <li>构建类型插件：通过设置相关构建结果的阈值，对构建阶段结果进行判断。如：Build构建插件，对Maven单元测试返回的结果进行判断。</li> <li>代码检查类型插件：通过设置相关检查项的阈值，对代码检查结果进行判断。如：Check代码检查插件，对代码检查缺陷进行判断。</li> <li>测试类型插件：通过设置相关测试结果阈值，对测试阶段结果进行判断。如：TestMan接口测试插件，对测试套件中测试用例通过率进行判断。</li> </ul> |

| 参数项  | 说明  |
|------|---|
| 插件版本 | 插件的所有版本。仅支持配置了阈值的插件版本。  |
| 阈值配置 | <p>根据选中的插件版本自动获取的插件输出阈值配置，根据需要为插件输出设置比较关系。规则阈值将用于策略，并最终配置到流水线中作为流水线阶段准出的判断条件。</p> <p><b>说明</b><br/>文本类型一般用于“包含”和“不包含”比较关系中；检查项为通过率时，数值取值范围为0~1。</p> |

**步骤5** 单击“确定”，即可在规则列表生成一条规则。

- 在规则列表页面，单击“操作”列 ，可编辑规则。
  - 规则类型不允许修改。
  - 编辑规则后，所有引用了该规则的策略内容都将自动同步修改。
- 在规则列表页面，单击“操作”列 ，弹出删除警告对话框，确认后单击“删除”可删除规则。规则删除后，所有引用了该规则的策略都将自动同步去除引用。

----结束

## 配置流水线租户级策略

租户级策略是一种租户级别的资源，可应用于当前租户下所有流水线的准出条件。租户级策略默认存在一条“系统策略”，用户可查看和使用该策略，但不能编辑和删除。


**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 单击右上角头像，选择“租户设置”。






**步骤3** 单击左侧导航“策略管理 > 策略”，进入租户级策略列表页面。

**步骤4** 单击“新建策略”，进入“新建策略”页面，填写参数信息。

表 5-3 租户级策略参数信息

| 参数项 | 说明  |
|-----|---|
| 名称  | 策略名称，默认根据当前时间生成。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 规则  | <p>选中的规则会在页面右侧展示，可对每条规则进行如下操作：</p> <ul style="list-style-type: none"> <li>编辑：单击规则右上方“详情”，可以查看规则详情，继续单击右上方“编辑”，可修改规则。</li> <li>开关：单击规则右上方 ，可打开/关闭规则，关闭后，该规则在应用到流水线准出条件中会自动关闭并不做拦截校验。</li> </ul> <p><b>说明</b><br/>单个策略中最多可配置20个规则。</p> |

**步骤5** 单击“确定”，即可在策略列表生成一条策略。

- 在策略列表页面，单击列表右侧，可编辑策略。
- 在策略列表页面，单击列表右侧，可查看策略，继续单击右上方“编辑”，可编辑策略。
- 在策略列表页面，单击列表右侧，选中“复制”，可基于当前策略快速复制一个新策略。
- 在策略列表页面，单击列表右侧，选中“删除”，弹出删除警告对话框，确认后单击“删除”，可删除策略。删除策略时，系统会提示当前有多少流水线引用了该策略，删除后相关流水线调度运行时可能会运行失败。
- 在策略列表页面，单击列表右侧，可以打开/关闭策略。关闭策略时，如果策略有被引用，系统会提示当前有多少流水线引用了该策略，关闭后相关流水线调度运行时可能会忽略该策略中的准出条件。

----结束

## 配置流水线项目级策略


项目级策略是一种项目级别的资源，可应用于当前项目下所有流水线的准出条件。

**步骤1** 访问[CodeArts Pipeline](#)首页，进入项目下流水线。





**步骤2** 单击“策略”，进入项目级策略列表页面。

**步骤3** 单击“新建策略”，进入“新建策略”页面，填写参数信息。





表 5-4 项目级策略参数信息

| 参数项 | 说明  |
|-----|---|
| 名称  | 策略名称，默认根据当前时间生成。支持输入中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 规则  | 选中的规则会在界面右侧展示，可对每条规则进行如下操作： <ul style="list-style-type: none"><li>• 编辑：单击规则右上方“详情”，可查看规则详情，继续单击右上方“编辑”，可修改规则。</li><li>• 开关：单击规则右上方，可打开/关闭规则，关闭后，该规则在应用到流水线准出条件中会自动关闭并不做拦截校验。</li></ul> <p><b>说明</b><br/>单个策略中最多可配置20个规则。</p> |

**步骤4** 单击“确定”，即可在策略列表生成一条项目级策略。

- 在策略列表页面，单击列表右侧，可编辑策略。
- 在策略列表页面，单击列表右侧，可查看策略，继续单击右上方“编辑”，可编辑策略。
- 在策略列表页面，单击列表右侧，选中“复制”，可基于当前策略快速复制一个策略。
- 在策略列表页面，单击列表右侧，选中“删除”，弹出删除警告对话框，确认后单击“删除”，可删除策略。

删除策略时，系统会提示当前有多少流水线引用了该策略，删除后相关流水线调度运行时可能会运行失败。


- 在策略列表页面，单击列表右侧 ，可以打开/关闭策略，关闭策略时，如果策略有被引用，系统会提示当前有多少流水线引用了该策略，关闭后相关流水线调度运行时可能会忽略该策略中的准出条件。
- 在策略列表页面，单击页面右上方“租户策略”，打开“租户策略”侧滑框，可以查看、复制和继承租户级策略。
  - 查看：单击操作列 ，可查看租户级策略，继续单击右上方“编辑”，可编辑租户级策略。
  - 复制：单击操作列 ，可基于当前租户级策略复制一个项目级策略。
  - 继承：单击操作列 ，可基于当前租户级策略继承一个项目级策略，通过继承生成的项目级策略，继承的规则实时和父租户级策略保持一致。。

---结束

## 5.2 编排流水线任务


任务是流水线上可管理的最小执行单元，可以在流水线阶段中进行纳管和串并行编排。

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 ，选择“编辑”。



**步骤3** 进入流水线“任务编排”页面，单击阶段下“新建任务”，弹出“新建任务”侧滑框。

### 说明

- 单击任务下方的 ，可以添加一个和该任务串行编排的任务，即按顺序执行，如：构建任务和部署任务应该按顺序先后执行。
- 单击“并行任务”，可以添加一个和已有任务并行编排的任务，即同时执行，如：代码检查任务和构建任务可以同时执行。

**步骤4** 为任务配置插件及任务信息，操作说明如下：

表 5-5 配置任务

| 操作项  | 说明   |
|------|--|
| 添加插件 | <p>流水线插件分为构建插件、代码检查插件、部署插件、测试插件和通用插件5种类型，可以根据不同类型进行过滤或搜索，插件来源请参考<a href="#">管理流水线扩展插件</a>。</p> <p>将鼠标移动到插件卡片，单击“添加”，即可将插件添加到任务中，请根据需要配置插件相关信息：</p> <ul style="list-style-type: none"> <li>• 填写插件名称。</li> <li>• 选择需要调用的任务。支持搜索过滤，如果找不到合适的任务，请根据界面提示新建任务。</li> <li>• 如果调用的任务有参数，参数也会显示出来，请根据需要配置相应参数。</li> <li>• 插件名称后有对应标记，单个任务中最多只能添加一个带“任务”标记的插件；带“草稿”标记的插件表示当前用户发布为草稿的自定义插件。</li> <li>• “挂起流水线”插件只能添加在不含并行任务的阶段中。</li> </ul> |
| 删除插件 | <p>将鼠标移动到已经添加的插件卡片，单击，选择“删除”，可以删除当前插件。</p>  |
| 更换插件 | <p>将鼠标移动到已经添加的插件卡片，单击，选择“更换”，或者单击插件上方“更换插件”，可以返回插件列表，更换当前插件。</p>  |
| 排序插件 | <p>单击插件卡片不松开，可以移动插件调整插件顺序。</p>   |



| 操作项  | 说明   |
|------|--|
| 任务配置 | <p>配置任务ID、执行主机和执行条件。</p> <ul style="list-style-type: none"> <li>● 任务ID：流水线阶段任务的ID，同一条流水线上的任务ID不能重复。支持大小写英文字母、数字、“-”、“_”，长度范围为1~128个字符。</li> <li>● 执行主机：包括内置执行机和自定义执行机。 <ul style="list-style-type: none"> <li>- 内置执行机：服务提供的执行主机，用户无需配置开箱即用。</li> <li>- 自定义执行机：支持用户自行配置工具和运行环境，以满足不同任务的需求。选择自定义执行机需提前在资源池添加代理资源池，详细操作请参见<a href="#">资源池管理</a>。</li> </ul> </li> </ul> <p><b>说明</b><br/>执行主机仅针对非任务级插件。</p> <ul style="list-style-type: none"> <li>● 任务选择 <ul style="list-style-type: none"> <li>- 必须选择：必须执行，执行流水线时该任务默认勾选且无法修改。</li> <li>- 禁止选择：禁止执行，执行流水线时该任务禁止勾选。</li> <li>- 默认选择：执行流水线时该任务默认勾选但可按需修改。</li> <li>- 默认不选择：执行流水线时该任务默认不勾选但可按需修改。</li> </ul> </li> <li>● 执行条件：即流水线阶段任务执行时需要满足的前提条件，通过配置执行条件，可以控制阶段任务是否执行。 <ul style="list-style-type: none"> <li>- 前置任务未选择仍执行：上一个任务状态执行成功或未选择，都继续执行当前任务。</li> <li>- 前置任务成功时执行：上一个任务执行成功才执行当前任务。</li> <li>- 前置任务失败时执行：上一个任务执行失败才执行当前任务。</li> <li>- 始终执行：始终执行当前任务，上一个任务状态是失败、成功、中止或忽略执行，都继续执行当前任务。</li> <li>- 表达式：当前置任务到达终态，包括：成功（COMPLETED）、失败（FAILED）、中止（CANCELED）、忽略执行（IGNORED），且表达式结果为“true”时，执行当前任务，否则不执行。表达式形式为“\${{}}”，由流水线上上下文、操作符、函数或常量任意组合而成。</li> </ul> <p>示例：<br/>在执行条件中定义不管前置任务（任务ID为“job_1”）运行成功或失败，均执行当前任务，表达式语法如下：<br/> <pre> \${{ jobs.job_1.status == 'COMPLETED'    jobs.job_1.status == 'FAILED' }} </pre> 更多表达式编写请参考<a href="#">配置流水线执行条件表达式</a>。</p> </li> </ul> |

**步骤5** 配置完任务，单击“确定”，完成任务添加，可根据实际需要编辑、复制、删除、移动任务。

**表 5-6** 管理任务

| 操作项  | 说明               |
|------|------------------|
| 编辑任务 | 单击任务卡片，可以编辑当前任务。 |



| 操作项  | 说明  |
|------|---|
| 复制任务 | 将鼠标移动到任务卡片，单击  ，可以复制一个和该任务串行编排的任务。 |
| 删除任务 | 将鼠标移动到任务卡片，单击  ，根据删除提示确定是否删除任务。    |
| 排序任务 | 单击任务卡片不松开，可以移动任务调整任务顺序。<br><b>说明</b><br>任务并行执行时不支持调整顺序。   |

**步骤6** 任务编排完成后，保存流水线。

----结束

## 5.3 配置流水线参数

流水线参数可以传递给各个任务使用，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、接口测试等任务的数据拉通。流水线参数包括以下三种。

- 系统预定义参数：系统默认提供的参数，无需定义、不能删除、不能修改。
- 自定义参数：根据需要自定义添加，参数类型包括字符串类型、枚举类型、自增长类型。最多可以新建100个自定义参数。
- 通用参数组：项目下通用流水线参数，所有流水线都可以关联并使用通用参数组。最多可以新建5个通用参数组，每个参数组最多可以添加20个自定义参数。

### 说明

- 新建流水线时如果设置了别名，可基于别名生成仓库对应的系统参数，未设置别名则以仓库名作为别名，生成系统参数，如：“别名\_TAG”，表示仓库标签名称。
- 当同一条流水线关联多个参数组，且多个参数组存在同名参数时，以后关联的参数组参数取值为准。
- 参数引用形式：\${参数名}，在参数输入框输入“\$”符号后会自动出现参数列表。

## 流水线系统预定义参数

表 5-7 流水线系统预定义参数

| 参数信息            | 说明           |
|-----------------|--------------|
| PROJECT_ID      | 当前流水线所属项目ID。 |
| PIPELINE_ID     | 当前流水线ID。     |
| PIPELINE_NUMBER | 流水线执行编号。     |



| 参数信息                   | 说明   |
|------------------------|--|
| COMMIT_ID              | 执行时代码最后一次提交的ID。  |
| COMMIT_ID_SHORT        | 执行时代码最后一次提交的短ID。   |
| TIMESTAMP              | 流水线执行时间戳。例如：20211222124301。  |
| PIPELINE_TRIGGER_TYPE  | 流水线触发类型。包括：Manual、Scheduler、RollBack以及WebHook类型（CreateTag、Note、Issue、MR、Push）。 |
| PIPELINE_NAME          | 流水线名称。   |
| REPO_URL               | 代码仓地址（HTTPS）。  |
| EXECUTE_USER           | 执行人。   |
| EXECUTE_USER_ID        | 执行人ID。   |
| EXECUTE_USER_NAME      | 执行人名称。   |
| EXECUTE_USER_NICKNAME  | 执行人昵称。   |
| PASS_CONDITIONS_LINK   | 流水线执行详情链接。   |
| PIPELINE_RUN_ID        | 流水线执行ID。   |
| MERGE_ID               | 合并请求ID。  |
| WEBHOOK_PAYLOAD        | Webhook请求负载信息。   |
| Repo01_REPOSITORY_NAME | 仓库名称。  |

| 参数信息                   | 说明              |
|------------------------|-----------------|
| Repo01_SOURCE_BRANCH   | 仓库运行源分支名称。      |
| Repo01_TARGET_BRANCH   | 仓库运行目标分支名称。     |
| Repo01_TAG             | 仓库标签名称。         |
| Repo01_COMMIT_ID       | 执行时代码最后一次提交的ID。 |
| Repo01_COMMIT_ID_SHORT | 执行时代码最后一次提交的短ID |
| Repo01_REPO_URL        | 代码仓地址（HTTPS）。   |

## 配置流水线自定义参数

介绍流水线自定义参数新建和配置。

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“参数设置”页面。



**步骤4** 单击参数列表下的“立即新建”（已存在自定义参数，则单击“新建参数”），填写参数信息。

表 5-8 流水线自定义参数说明

| 参数信息 | 说明   |
|------|--|
| 名称   | 自定义参数名称，不可与系统预定义参数名称相同。支持大小写英文字母、数字、“_”，长度不超过128个字符。 |
| 类型   | 参数可选类型：字符串（默认类型）、自增长、枚举。                             |

| 参数信息  | 说明  |
|-------|---|
| 默认值   | <p>自定义参数的默认取值。</p> <ul style="list-style-type: none"> <li>字符串：长度不超过8192个字符，可以为空。</li> <li>自增长：长度不超过8192个字符。自增长类型参数如果被引用，全新执行一次后，下次执行时参数值末位如果是数字则会自动+1。</li> <li>枚举：仅支持中文、大小写英文字母、数字、“-”、“_”、“.”、“/”，长度不超过8192个字符。<br/>选择枚举类型后，在弹出的对话框中添加可选取值，添加完成后在“默认值”下拉框中选择一个默认值，支持输入关键词搜索过滤。</li> </ul> |
| 私密参数  | 参数为私密参数时，系统会将输入参数做加密存储，使用时解密，同时在运行日志里不可见。   |
| 运行时设置 | 打开“运行时设置”开关，在执行流水线时支持修改该参数的值。   |
| 参数描述  | 不超过512个字符。  |

### 📖 说明

单击“操作”列 ，可以新增参数；单击“操作”列 ，可以删除参数。

**步骤5** 配置完成后，保存流水线。

----结束

## 配置流水线通用参数组

**步骤1** 通过项目入口方式[访问CodeArts Pipeline首页](#)。

**步骤2** 切换至“通用参数组”页面，单击“新建参数组”。

**步骤3** 进入“新建通用参数组”页面，填写参数信息。

图 5-2 新建通用参数组

表 5-9 通用参数组参数说明

| 基本信息    | 说明  |
|---------|---|
| 所属项目    | 通用参数组归属项目，不可更改。                                       |
| 参数组名称   | 支持大小写英文字母、数字、“_”，长度不超过128个字符。                         |
| 描述      | 长度不超过512个字符。  |
| 自定义参数列表 | 单击“立即新建”，可根据需要添加自定义参数，参数配置方法同 <a href="#">自定义参数</a> 。 |

**步骤4** 单击“确定”，新建完成。

**步骤5** 编辑需要关联通用参数组的流水线，进入“参数设置 > 通用参数组”页面。

**步骤6** 单击“立即关联”，选择参数组，然后单击“确定”，即可将参数组关联到当前流水线。


- 展开通用参数组，可以查看参数详情。
- 单击操作列 ，可以解除关联参数组。

图 5-3 关联参数组

**步骤7** 配置完成后，保存流水线。

----结束

## 流水线参数应用示例

本文以通过流水线统一配置releaseversion参数并传递到构建任务为例，介绍流水线参数使用方法。

**步骤1 新建构建任务。**

**步骤2** 在“参数设置”页面添加releaseversion参数并设置好默认值，打开“运行时设置”。

图 5-4 新建构建任务参数

| 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 | 1.0.0  | <input type="checkbox"/> | <input checked="" type="checkbox"/> |              | 前  |

**步骤3** 在“构建步骤”页面，选择“上传软件包到软件发布库”，将“发布版本号”配置成引用参数形式。在参数输入框输入“\$”符号后会自动出现参数列表，选择上一步已经新建好的releaseversion参数。

图 5-5 引用构建任务参数

上传软件包到软件发布库  
上传软件包到软件发布库。 [查看操作指南](#)

\* 步骤显示名称  
上传软件包到软件发布库

\* 构建包路径 ?  
\*\*/target/\*.?ar

发布版本号 ?  
\$(releaseversion)

包名 ?

### 说明

使用“\$”符号引用通用参数组的参数时不会自动生成提示。

**步骤4** 配置完成后，保存构建任务。

**步骤5** 新建空白模板流水线，在流水线阶段添加一个新的任务，然后在该任务中添加“Build构建”插件并调用刚创建的构建任务，即可看到刚新建好的构建任务参数releaseversion。

图 5-6 配置构建任务参数

\* 参数名称: releaseversion  
设置为流水线参数，支持任务中引用该参数  
设置为流水线参数

\* 1.0.1

**步骤6** 将鼠标移动到releaseversion参数上，可以一键将该参数设置为流水线参数并能直接引用该参数；或直接单击“确定”，然后切换到流水线“参数设置”页面，新建流水线参数releaseversion，类型可以根据需求选择自增长或字符串等，设置默认值并打开运行时设置。

图 5-7 新建流水线参数

| 名称             | 类型  | 默认值   | 私密参数                     | 运行时设置                               | 参数描述 | 操作  |
|----------------|-----|-------|--------------------------|-------------------------------------|------|-----|
| releaseversion | 字符串 | 1.0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> |      | + 删 |

**步骤7** 再切回到流水线“任务编排”页面，编辑刚添加的构建任务，将流水线的releaseversion参数通过“\$”符号引用到构建任务的releaseversion参数内。

图 5-8 引用流水线参数

\*releaseversion

#### 说明

- 该处只显示构建任务中开启“运行时设置”的文本类型参数。
- 鼠标移至参数名称，可以快速将该参数设置为流水线参数并能直接引用该参数。

**步骤8** 完成参数引用后，保存信息，然后单击“保存并执行”，弹出“执行配置”对话框，可查看“运行参数配置”。

运行时参数值默认为添加参数时为参数设置的默认值，可根据需要进行修改。流水线运行时在构建任务内使用的就是此时输入的参数值。

**步骤9** 设置完参数，单击“执行”，开始使用配置的参数执行流水线。

----结束

## 5.4 配置流水线执行计划

流水线支持配置事件触发、定时任务、Webhook触发和并发策略四种执行计划。通过事件触发和定时任务，实现自动化执行；通过Webhook触发，用户可以通过任意第三方系统触发流水线执行；通过并发策略，可以更细粒度地为用户分配并发资源。

### 配置流水线事件触发

事件触发包括代码提交时触发、合并请求时触发、创建标签时触发。码云代码仓还支持Pull Request、新建issue、评论issue等其他事件。

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“执行计划”页面，在“事件触发”勾选触发事件进行配置。

- 代码提交时触发（代码托管/码云）

监听代码提交触发事件，可以组合分支和具体路径的包含/排除范围，当在关联的代码仓库中有代码提交且所在的分支和变更文件的路径满足包含/排除策略，会自动触发关联流水线执行。

- 分支过滤：可以通过设置包含或排除分支，来监听触发事件所在的目标分支。
- 路径过滤：可以通过设置包含或排除路径，来监听触发事件所在的变更文件路径。

图 5-9 配置代码提交时触发

配置代码提交时触发的界面截图。顶部有复选框“代码提交时触发”且已勾选。下方分为“分支过滤”和“路径过滤”两部分。每部分都有“包含”和“排除”两个下拉菜单，以及对应的输入框。在“分支过滤”部分，“包含”框内已输入“master”，“排除”框内提示“请选择分支或输入正则表达式”。在“路径过滤”部分，两个框内均提示“请输入路径或正则表达式”。每部分下方都有一个“+ 新增”按钮。

- 合并请求时触发（代码托管/码云）

监听合并请求触发事件，可以组合分支和具体路径的包含/排除范围，对新建、合并、更新和重新打开事件设置触发策略，当在关联的代码仓库中触发合并请求事件且分支满足包含/排除策略，会自动触发关联流水线执行。

事件说明：

- 新建：合并请求创建时触发。
- 合并：合并请求合入时触发，该事件会同时触发代码提交事件。
- 重新打开：合并请求重新打开时触发。
- 更新：合并请求内容更新、设置更新或者源代码更新时触发，打开“仅代码更新触发”，则仅源代码更新时触发。

分支说明：

- 分支过滤：可以通过设置包含或排除分支，来监听触发事件所在的目标分支。
- 路径过滤：可以通过设置包含或排除路径，来监听触发事件所在的变更文件路径。

图 5-10 配置合并请求时触发

配置合并请求时触发的界面截图。顶部有复选框“合并请求时触发”且已勾选。下方有“事件”和“分支过滤/路径过滤”两部分。“事件”部分有“新建”、“合并”、“重新打开”、“更新”四个复选框，其中“更新”复选框下方有一个“仅代码更新触发”的开关，当前处于开启状态。下方是“分支过滤”和“路径过滤”部分，结构与图 5-9 一致。

- 创建标签时触发（代码托管）

监听创建标签触发事件，可以设置标签包含/排除范围，当在关联的代码仓库创建标签且满足标签包含/排除策略，会自动触发关联流水线执行。

图 5-11 配置创建标签时触发



- 其他事件（码云）
  - 评论Pull Request：评论Pull Request时触发流水线执行。
  - 新建issue：新建issue时触发流水线运行。
  - 评论issue：评论issue时触发流水线运行。

#### 📖 说明

- 除了上述事件外，码云代码仓还支持开启事件触发成功后自动评论。
- 先匹配分支，再匹配路径，都匹配成功，则触发流水线。
- 路径排除优先于路径包含，即先匹配排除的路径，如果变更文件没有全部在排除范围内，则继续匹配包含的路径。若未配置包含路径，则匹配成功，触发流水线；若配置了包含路径，并且变更文件在路径排除范围外有任意一个在包含范围内，则匹配成功，触发流水线。
- 标签排除优先于标签包含，即标签同时在包含和排除中时，则匹配失败，不触发流水线。
- 路径匹配范围：单次提交的前300个变更文件，变更文件数超过300建议拆分。

**步骤4** 配置完成后，保存流水线。

----结束

## 配置流水线定时执行

配置流水线在指定的时间自动执行。

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“执行计划”页面。

**步骤4** 单击“定时任务”区域的“创建定时任务”，增加一个定时任务，打开“启用定时执行”开关（默认开启），配置如下信息：



图 5-12 配置定时任务

定时任务\_1   

按照配置定时触发流水线运行

---

\*启用定时执行

\*执行日

周日  周一  周二  周三  周四  周五  周六

\*时间段

00:00  至 00:00  (UTC+08:00) 北京, 重庆, 香港特别行政区, 乌... 



间隔时间

00:15:00 

表 5-10 定时任务

| 参数项  | 说明                   |
|------|----------------------|
| 执行日  | 选择执行日期。              |
| 时间段  | 选择执行时间段和时区。          |
| 间隔时间 | 设置间隔时间，即每间隔多长时间触发一次。 |

#### 说明

- 最多可以设置10个定时任务。
- 单击右上角 ，可删除定时任务；单击右上角 ，可复制定时任务。


步骤5 配置完成后，保存流水线。

----结束

## Webhook 触发

通过Webhook触发，用户可以通过任意第三方系统触发流水线执行。

步骤1 访问[CodeArts Pipeline](#)首页。

步骤2 在流水线列表页搜索目标流水线，单击操作列，选择“编辑”。

步骤3 进入流水线“任务编排”页面，切换至“执行计划”页面。

步骤4 打开“Webhook”开关（默认关闭），参数说明如[表5-11](#)，保存流水线后即可生效。

图 5-13 配置 Webhook 触发



表 5-11 配置 Webhook 触发

| 参数项         | 说明   |
|-------------|--|
| Webhook 触发源 | 将该地址复制到第三方系统触发器，使用Post方法调用，即可运行流水线。  |
| IAM认证       | <ul style="list-style-type: none"><li>启用IAM认证，需要将用户IAM Token加入接口请求头，调用示例如下：<br/><code>curl --header "Content-Type: application/json" --header 'x-auth-token: XXXX ( IAM Token )' --request POST --data "{}" <b>Webhook触发源</b></code></li><li>不启用IAM认证，调用示例如下：<br/><code>curl --header "Content-Type: application/json" --request POST --data "{}" <b>Webhook触发源</b></code></li></ul> |

---结束

## 配置流水线并发策略

默认单条流水线最大执行并发数为5，超过并发数后新的触发忽略不执行。可以通过并发策略为单条流水线配置同时处于运行中和暂停状态的流水线最大实例个数，即并发数，更细粒度地为用户分配并发资源。


- 步骤1** 访问[CodeArts Pipeline](#)首页。
- 步骤2** 在流水线列表页搜索目标流水线，单击操作列，选择“编辑”。
- 步骤3** 进入流水线“任务编排”页面，切换至“执行计划”页面。
- 步骤4** 打开“并发策略”开关（默认关闭），设置并发数和超过并发后执行策略。

图 5-14 配置并发策略

### 并发策略

本条流水线允许同时处于运行中和暂停时的流水线最大实例个数

\* 并发个数 

\* 超过并发后执行策略

- 排队等待  
 忽略不执行

表 5-12 配置并发策略

| 参数项       | 说明   |
|-----------|--|
| 并发个数      | 单条流水线最大执行并发数，受套餐和购买并发数限制。  |
| 超过并发后执行策略 | 超过并发数后可选择如下两种处理方式： <ul style="list-style-type: none"><li>● 排队等待：超过并发数后新的触发排队等待执行，可进入流水线详情页查看排队记录。<ul style="list-style-type: none"><li>- 单条流水线最多支持100条排队记录。</li><li>- 单条排队记录排队时长超过24小时后会放弃执行。</li><li>- 排队记录支持手动取消排队，即放弃执行。</li><li>- 排队记录执行时，配置信息以入队时为准。</li></ul></li><li>● 忽略不执行：超过并发数后新的触发忽略不执行。</li></ul> |

步骤5 配置完成后，保存流水线。

----结束

## 5.5 配置流水线权限

流水线支持对单条流水线按照角色和人员两种维度进行权限配置，包括角色权限和人员权限。

- 默认情况下，流水线的角色权限和项目中的角色权限保持一致。
- 项目创建者和流水线创建者权限不能更改。
- 人员权限优先级高于角色权限：默认情况下，用户修改角色权限后会自动同步人员权限，如果修改了人员权限，则以人员权限为准。
- 默认情况下，有编辑或执行权限就有查看权限。

## 配置流水线权限

**步骤1** 访问CodeArts Pipeline首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“权限管理”页面，关闭“继承项目权限配置”，即可为流水线配置角色权限和人员权限。

- **配置角色权限**

通过勾选/取消勾选，指定角色对当前流水线的查看、执行、编辑、删除操作权限。

**图 5-15** 配置角色权限

| 角色        | 查看                                  | 执行                                  | 编辑                                  | 删除                                  |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 项目创建者     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 流水线创建者    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 项目经理      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 开发人员      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 测试经理      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 测试人员      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 参与者       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 浏览者       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 运维经理      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 产品经理      | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| 系统工程师     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Committer | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |

- **配置人员权限**

通过勾选/取消勾选，指定人员对当前流水线的查看、执行、编辑、删除操作权限。

**图 5-16** 配置人员权限

| 人员 | 所属角色   | 昵称 | 企业用户 | 查看                                  | 执行                                  | 编辑                                  | 删除                                  |
|----|--------|----|------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
|    | 流水线创建者 | kk |      | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

----结束

## 5.6 配置流水线通知订阅

流水线支持配置事件通知，包括服务动态通知、飞书通知、企业微信通知和钉钉通知，请根据需要选择通知方式。

### 配置流水线服务动态通知

通过邮件或服务动态的方式推送通知。流水线被删除、执行失败、执行成功以及配置被更新时，可以配置给创建人、执行人、收藏人发送服务动态或邮件。

**步骤1** 访问CodeArts Pipeline首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列...，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“通知订阅”页面。

**步骤4** 单击“官方通知”下“服务动态”，根据需要勾选/取消勾选通知方式。


- 默认情况下，服务动态通知都发送，邮件通知都不发送。
- 在流水线首页右上角单击，会弹出“通知”侧滑框，即可查看通知消息。

图 5-17 配置服务动态通知

| 名称       | 通知方式   | 默认用户  |
|----------|--|---|
| 流水线被删除   | <input checked="" type="checkbox"/> 服务动态 <input type="checkbox"/> 邮件 | <input checked="" type="checkbox"/> 创建人 <input checked="" type="checkbox"/> 执行人 <input checked="" type="checkbox"/> 收藏人 |
| 流水线执行失败  | <input checked="" type="checkbox"/> 服务动态 <input type="checkbox"/> 邮件 | <input checked="" type="checkbox"/> 创建人 <input checked="" type="checkbox"/> 执行人 <input checked="" type="checkbox"/> 收藏人 |
| 流水线执行成功  | <input checked="" type="checkbox"/> 服务动态 <input type="checkbox"/> 邮件 | <input checked="" type="checkbox"/> 创建人 <input checked="" type="checkbox"/> 执行人 <input checked="" type="checkbox"/> 收藏人 |
| 流水线配置被更新 | <input checked="" type="checkbox"/> 服务动态 <input type="checkbox"/> 邮件 | <input checked="" type="checkbox"/> 创建人 <input checked="" type="checkbox"/> 执行人 <input checked="" type="checkbox"/> 收藏人 |

**步骤5** 配置完成后，保存流水线。

----结束

## 配置流水线飞书通知

将流水线信息推送到飞书。

**步骤1** 访问CodeArts Pipeline首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列...，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“通知订阅”页面。

**步骤4** 单击“第三方通知”下“飞书”，根据需要填写通知信息，填写完成后即可激活通知。

图 5-18 配置飞书通知



飞书  
流水线信息推送到飞书

\* Webhook地址

.....

加签密钥 ?

.....

\* 事件类型 ?

已完成

通知内容

流水线名称

@user\_id ?

afewaf

确定

表 5-13 配置飞书通知

| 操作项       | 说明   |
|-----------|--|
| Webhook地址 | 填写飞书机器人通知URL。  |
| 加签密钥      | 飞书自定义机器人可以开启安全设置，若未使用加签，无需填写加签密钥。                        |
| 事件类型      | 选择要通知的事件，可选择已完成、失败、暂停、挂起、忽略五种事件类型。                       |
| 通知内容      | 选择要通知的内容，可选择流水线名称、流水线执行描述、执行人、触发信息、项目名称、执行时间、运行状态七种通知内容。 |
| @user_id  | 填写要接受通知的飞书的user_id，多个user_id之间以逗号分隔。                     |

**步骤5** 配置完成后，保存流水线。

----结束

## 配置流水线企业微信通知

将流水线信息推送到企业微信。

- 步骤1** 访问CodeArts Pipeline首页。
- 步骤2** 在流水线列表页搜索目标流水线，单击操作列...，选择“编辑”。
- 步骤3** 进入流水线“任务编排”页面，切换至“通知订阅”页面。
- 步骤4** 单击“第三方通知”下“企业微信”，根据需要填写通知信息，填写完成后即可激活通知。

图 5-19 配置企业微信通知

表 5-14 配置企业微信通知

| 操作项       | 说明   |
|-----------|--|
| Webhook地址 | 填写企业微信机器人通知URL。  |
| 事件类型      | 选择要通知的事件，可选择已完成、失败、暂停、挂起、忽略五种事件类型。                       |
| 通知内容      | 选择要通知的内容，可选择流水线名称、流水线执行描述、执行人、触发信息、项目名称、执行时间、运行状态七种通知内容。 |
| @user_id  | 填写要接受通知的企业微信的user_id，多个user_id之间以逗号分隔。                   |

**步骤5** 配置完成后，保存流水线。

---结束

## 配置流水线钉钉通知

将流水线信息推送到钉钉。

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页搜索目标流水线，单击操作列 $\dots$ ，选择“编辑”。

**步骤3** 进入流水线“任务编排”页面，切换至“通知订阅”页面。

**步骤4** 单击“第三方通知”下“钉钉”，根据需要填写通知信息，填写完成后即可激活通知。

图 5-20 配置钉钉通知



**钉钉**  
流水线信息推送到钉钉

\* Webhook地址

请输入钉钉机器人通知URL

加签密钥  $\textcircled{?}$

若钉钉机器人安全设置开启加签，请填写加签密钥

\* 事件类型  $\textcircled{?}$

请选择事件类型

通知内容  $\textcircled{?}$

请选择通知内容

@user\_id  $\textcircled{?}$

请输入你要@的成员user\_id

确定

表 5-15 配置钉钉通知

| 操作项       | 说明            |
|-----------|---------------|
| Webhook地址 | 填写钉钉机器人通知URL。 |



| 操作项      | 说明   |
|----------|--|
| 加签密钥     | 钉钉自定义机器人可以开启安全设置，若未使用加签，无需填写加签密钥。                        |
| 事件类型     | 选择要通知的事件，可选择已完成、失败、暂停、挂起、忽略五种事件类型。                       |
| 通知内容     | 选择要通知的内容，可选择流水线名称、流水线执行描述、执行人、触发信息、项目名称、执行时间、运行状态七种通知内容。 |
| @user_id | 填写要接受通知的飞书的user_id，多个user_id之间以逗号分隔。                     |

**步骤5** 配置完成后，保存流水线。

----结束

# 6 分组管理流水线

## 操作场景

在项目开发过程中，一般会涉及多个流水线，为方便管理和归类，可通过流水线分组功能，将具有相同特性的流水线进行归类统一管理，提升操作效率。例如，可以将流水线按环境级别划分为生产环境流水线、测试环境流水线等多个类别；或按照研发业务划分为定时构建、开发自测、集成测试、生产部署等多个业务场景。

## 约束与限制

项目创建者和项目经理才能管理分组。

## 分组管理流水线








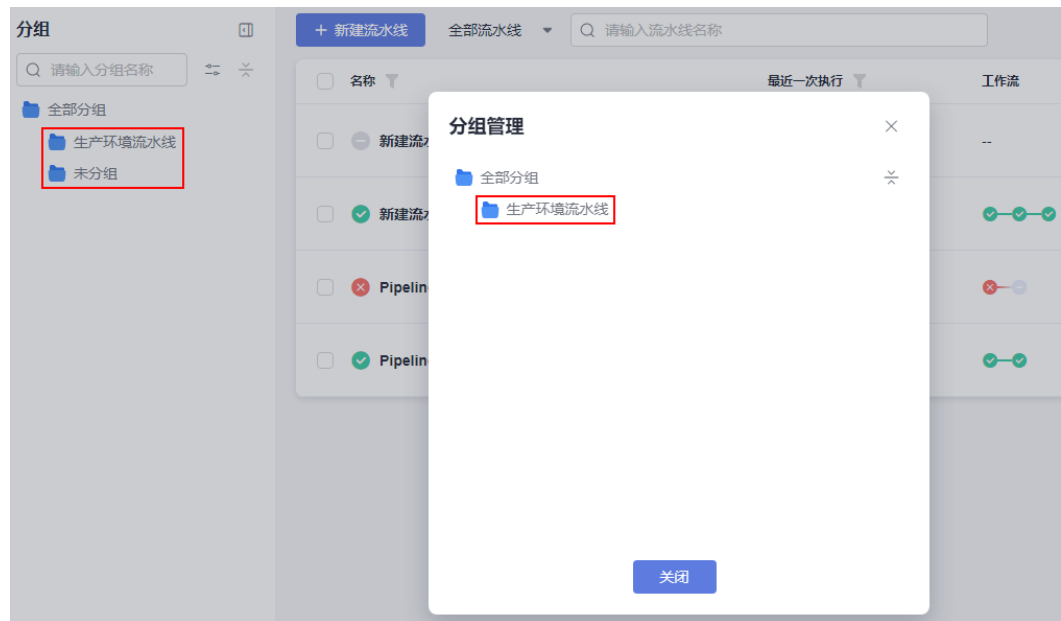
- 步骤1** 通过项目入口方式[访问CodeArts Pipeline首页](#)。
- 步骤2** 单击“全部分组”，展开流水线分组面板。
- 步骤3** 单击，弹出“分组管理”对话框。
- 步骤4** 将鼠标移动到“全部分组”所在行，单击，新增分组。
- 步骤5** 设置分组名称，单击确定创建分组，单击取消创建分组。分组创建完成后，可进行如下操作：
  - 单击分组所在行，可创建子分组，最多支持嵌套三级。
  - 单击分组所在行，可修改分组名称。
  - 单击分组所在行，可移动、删除分组。

图 6-1 配置流水线分组



### 📖 说明

第一个分组创建完成后，会自动生成一个“未分组”组别，未分组的流水线会自动划入该组。

**步骤6** 创建完所有分组，单击“关闭”，返回流水线列表页面。

**步骤7** 选择需要操作的流水线，可进行以下批量操作。

图 6-2 批量操作流水线



- 单击“移动到”，弹出“移动分组”对话框，选择某分组，单击“确定”，即可批量将流水线移动到该分组。
- 单击“执行”，弹出“批量执行”对话框，单击“确定”即可批量执行流水线。
- 单击“权限管理”，弹出“权限管理”对话框，可批量修改流水线权限。
- 单击“更多 > 设置标签”，弹出“设置标签”对话框，可批量为流水线设置标签。
- 单击“更多 > 删除”，弹出“批量删除”对话框，确认后输入提示信息，单击“确认”，即可批量删除流水线，一次最多可删除20条流水线。


----结束

# 7 执行流水线

流水线支持一键执行，执行过程可视化，可实时查看执行进度、日志、结果等信息，提供健康度分析。

## 执行流水线

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 在流水线列表页找到待执行的流水线，单击操作列。

**步骤3** 弹出“执行配置”对话框，配置如下信息：

- 流水线源：选择代码源的分支或标签。
- 运行参数配置：如果流水线配置了运行时参数，请根据需要设置参数值。运行参数配置及使用请参见[配置流水线参数](#)。
- 执行阶段配置：流水线支持选择任务执行，可以选择一个或多个，默认执行全部任务。

### 说明

流水线阶段如果在配置时选择了“总是运行”，则该阶段的任务默认选中且不能取消。

- 执行描述：填写本次执行的信息说明。

**步骤4** 单击“执行”，进入流水线执行详情页面，可实时查看流水线执行进展及任务执行状态。

图 7-1 执行流水线



- 单击“停止”，可中止任务执行。
- 单击“编辑”，可修改流水线配置信息。
- 流水线支持并发执行，继续单击“执行”，可以再次执行流水线，单条流水线最大支持同时执行的次数由购买套餐决定（体验版：1次；基础版：5次；专业版：10次；企业版：20次）。

**步骤5** 执行完成后，即可查看流水线执行结果。执行过程遇到的问题请参考[流水线常见问题](#)。

----结束

# 8 查看流水线

查看流水线列表信息、执行历史记录、执行详情、排队等待情况等。

## 约束与限制

默认只有项目经理或创建者（项目创建者和流水线创建者）才可以删除流水线，可通过配置权限给不同角色添加不同操作权限。

## 查看流水线

**步骤1** [访问CodeArts Pipeline首页](#)。

流水线列表页展示当前用户下的所有流水线，列表项说明如下：

表 8-1 流水线列表

| 参数项       | 说明  |
|-----------|---|
| 名称        | 流水线名称和流水线所属项目名称。<br><b>说明</b><br>项目下流水线列表不显示项目名称。 |
| 最近一次执行    | 最近一次的执行信息：执行方式、分支、最近一次代码提交的CommitID、执行人。          |
| 工作流       | 流水线的调度流程及执行状态（已完成、失败、运行中、已停止等）。                   |
| 开始时间&执行时长 | 最近一次执行开始时间和持续时长。                                  |

| 参数项     | 说明  |    |    |         |  |
|---------|---|----|----|---------|--|
| 操作      | <ul style="list-style-type: none"> <li>单击▶，可执行流水线。</li> <li>单击☆，可关注流水线，关注成功后图标变为★，再次单击可取消关注。</li> </ul> <p><b>说明</b><br/>收藏流水线后，刷新页面或下次进入流水线列表时，该流水线会在流水线列表中置顶显示，收藏多条流水线会按流水线最后一次执行时间降序排列，若未执行过，则按创建时间降序排列。</p> <ul style="list-style-type: none"> <li>单击⋮，选择“编辑”，可修改流水线。</li> <li>单击⋮，选择“复制”，可基于当前流水线快速创建一条新的流水线。</li> <li>单击⋮，选择“预览”，可预览流水线。</li> <li>单击⋮，选择“操作历史”，可查看流水线操作历史记录（新建记录、编辑记录）和触发失败记录。</li> <li>单击⋮，选择“设置标签”，弹出“设置标签”对话框。             <ul style="list-style-type: none"> <li>单击+，为流水线设置标签，最多可选择5个标签。</li> <li>单击“标签管理”，可新建标签、编辑标签、删除标签。</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p style="text-align: center;"><b>标签管理</b></p> <div style="display: flex; justify-content: space-between; align-items: center;"> <span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">+ 新建标签</span> <input style="border: 1px solid #ccc; border-radius: 3px; width: 150px;" type="text" value="请输入标签名称搜索"/> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="text-align: left;">标签</th> <th style="text-align: right;">操作</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">● tag01</td> <td style="text-align: right; padding: 5px;"> <span style="font-size: 1em;">✎</span> <span style="font-size: 1em;">🗑️</span> </td> </tr> </tbody> </table> </div> <p><b>说明</b><br/>单租户最多可创建100个标签。</p> <ul style="list-style-type: none"> <li>单击⋮，选择“禁用”，可禁用流水线。</li> <li>单击⋮，选择“删除”，可删除流水线。</li> </ul> | 标签 | 操作 | ● tag01 | <span style="font-size: 1em;">✎</span> <span style="font-size: 1em;">🗑️</span> |
| 标签      | 操作  |    |    |         |  |
| ● tag01 | <span style="font-size: 1em;">✎</span> <span style="font-size: 1em;">🗑️</span>  |    |    |         |  |

- 流水线任务列表默认所有用户均可以查看。
- 单击“全部流水线”的下拉框，可以根据“全部流水线”、“我新建的”及“最后一次记录由我执行的”进行过滤。
- 支持在搜索栏中输入流水线名称关键字进行搜索。
- 单击右上方“设置”，可以根据需要自定义设置流水线列表信息。

**步骤2** 单击流水线名称，进入流水线“执行历史”页面，展示流水线执行记录。

**📖 说明**

首次执行时执行历史页面为空，执行一次后才有记录。


表 8-2 流水线执行历史

| 参数项  | 说明  |
|------|---|
| 执行信息 | 展示执行序号、执行分支、分支最新Commit信息以及分支最新CommitID。                 |
| 状态   | 流水线执行状态，包括已完成、运行中、失败、已停止、暂停、挂起、忽略。                      |
| 触发方式 | 触发流水线运行的方式，包括手动触发、定时任务触发、MR触发、Push触发、Webhook触发、子流水线触发等。 |
| 工作流  | 流水线的调度流程及执行状态（已完成、失败、运行中、已停止等）。                         |
| 执行时间 | 流水线开始执行的时间。   |
| 执行时长 | 流水线执行的持续时长。   |

- 单击时间筛选器，可以选择时间段进行过滤，支持查看最近90天执行记录，默认展示最近31天的执行记录，支持最近7天、14天、31天、90天快捷切换。
- 单击右上角“设置”，可以根据需要自定义设置流水线执行历史列表信息。

**步骤3** 单击执行编号，进入“流水线详情”页面，查看对应执行记录详情。


表 8-3 流水线详情页操作

| 操作项  | 说明   |
|------|--|
| 重试   | 如果任务执行失败，单击页面右上角“重试”，可从失败处继续执行流水线。   |
| 编辑   | 单击“编辑”，可修改流水线配置信息。   |
| 执行   | 单击“执行”，可按最新的配置重新执行流水线，并生成一条新的执行记录。   |
| 下载   | 单击页面上方产出信息，可下载通过流水线自动构建出来的构建包。<br><b>说明</b> <ul style="list-style-type: none"> <li>• 只有构建类型的任务才有构建包。</li> <li>• 存在多个构建包时，单击“全部下载”，可以下载全部构建包。</li> <li>• 只展示最新构建的10个包，如需下载其他构建包，请切换到发布库页面自行下载。</li> </ul> |
| 查看日志 | 单击任务卡片，可查看对应任务的任务日志和任务结果。<br><b>说明</b> <ul style="list-style-type: none"> <li>• “延时执行”和“挂起流水线”任务没有日志。</li> <li>• 因未满足任务执行条件而未执行的任务，单击任务卡片会展示未执行的原因。</li> </ul>   |
| 更多操作 | 单击页面右上角  ，可复制流水线、查看流水线操作历史、预览流水线、禁用流水线、删除流水线。   |



**步骤4** 切换到“排队列表”页，可查看流水线排队记录。

如果流水线配置了并发策略，且超过并发后执行策略为排队等待，在该页面可以查看排队等待执行的记录。

- 单条流水线最多支持100条排队记录。
- 单条排队记录排队时长超过24小时后会被放弃执行。
- 单击操作列, 可手动取消排队记录，即放弃执行。
- 排队记录执行时，配置信息以入队时为准。

----结束

# 9 配置微服务变更流水线

微服务是一种软件治理架构，一个大型的复杂软件项目，可以由一个或多个微服务组成，系统中各个微服务之间是松耦合的，各微服务并行独立开发、构建、验证及部署上线。变更是一次研发交付活动的载体，项目的需求研发和漏洞修复都可以放在变更上来实现，一个变更只能属于一个微服务。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，用于与微服务下的变更资源进行联动，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。

微服务具备以下优点：

- 边界清晰：每一个微服务只关注一个特定的业务功能，开发和维护单个微服务都相对简单。
- 支持独立部署：单个微服务独立部署更新，不会对整个业务运行造成影响，保证业务7\*24小时在线。
- 技术多样化：微服务可以使用不同的技术栈来实现，不同服务之间通过restful API通信，每个服务可以使用不同的技术框架和不同的存储库来实现。

与普通流水线相比，变更流水线具有以下特点：

- 单个微服务下仅允许存在一条基于变更触发的流水线。
- 变更流水线基于变更触发，运行时会自动创建集成分支，并在运行成功后将集成分支内容合并至主干分支。
- 变更流水线运行成功后会自动更新发布的变更状态。
- 变更流水线仅支持单实例运行。
- 变更流水线不支持事件触发或定时任务触发。

## 新建流水线微服务

**步骤1** 访问[CodeArts Pipeline](#)首页，进入项目下流水线。

**步骤2** 单击“微服务”，进入微服务列表页面。

**步骤3** 单击“新建微服务”，进入“新建微服务”页面，配置相关参数。

表 9-1 新建微服务

| 参数项   | 说明  |
|-------|---|
| 所属项目  | 微服务所在项目，不可更改。   |
| 微服务名称 | 微服务的名称。支持大小写英文字母、数字、“_”，不超过128个字符。  |
| 代码源   | 代码仓库来源，当前仅支持Repo。<br><b>说明</b><br>可暂不选择代码源，微服务创建成功后，可根据需要通过单击微服务名称，进入微服务概览页再进行关联。 |
| 代码仓   | 微服务关联的代码仓库，选择已创建的代码仓库。<br><b>说明</b><br>一个代码仓只能关联一个微服务。                            |
| 默认分支  | 微服务关联的默认分支，执行微服务流水线时默认使用的分支。<br><b>说明</b><br>变更流水线执行完成后，所有变更特性分支都将会合入默认分支。        |
| 语言    | 微服务的开发语言，支持Java、Python、Node.js、Go、.Net、C++、PHP。                                   |
| 描述    | 长度不超过1024个字符。   |

**步骤4** 单击“确定”，完成微服务创建并进入微服务“概览”页。





概览页展示微服务的创建人、创建时间、代码源等信息，可修改微服务开发语言、关联的代码仓库和描述信息。

#### 说明

更换代码仓库时，如果微服务下存在没有关闭的变更或运行中的流水线，页面会弹出“数据处理”侧滑框，需先关闭所有变更并停止所有运行中的流水线。

**步骤5** 返回微服务列表，可以查看已创建的微服务，列表项说明如下。

表 9-2 微服务列表

| 列表项  | 说明  |
|------|---|
| 微服务  | 微服务的名称。   |
| 创建人  | 创建微服务的用户名。  |
| 创建时间 | 创建微服务的时间，将鼠标悬浮至“创建时间”列，单击  ，可以根据创建时间排序微服务。   |
| 状态   | 展示微服务当前状态信息，创建后会自动处于激活状态。   |
| 操作   | 单击  可收藏该微服务，收藏成功后图标变为  ，再次单击可取消收藏；单击  ，可删除该微服务。删除微服务，微服务内所有变更记录和流水线都将会被删除，请谨慎操作。 |


- 微服务列表展示项目下的所有微服务。
- 支持在搜索栏中输入微服务名称关键字进行搜索。

----结束

## 新建变更流水线

- 步骤1** 访问CodeArts Pipeline首页，进入项目下流水线。
- 步骤2** 单击“微服务”，进入微服务列表页面。
- 步骤3** 单击微服务名称，进入微服务“概览”页面。
- 步骤4** 单击“流水线”，进入微服务流水线列表页面。
- 步骤5** 单击“新建流水线”，进入“基本信息”页面，配置相关参数。

表 9-3 新建流水线

| 参数项           | 说明  |
|---------------|---|
| 所属项目          | 微服务所在的项目。   |
| 名称            | 流水线的名称，默认根据当前时间生成。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          | 流水线代码来源，微服务下流水线仅支持Repo。   |
| 代码仓           | 微服务关联的代码仓库。<br><b>说明</b><br>修改微服务的代码仓库，微服务下所有流水线配置的代码仓库会自动更改。   |
| 默认分支          | 微服务关联的默认分支。<br><b>说明</b><br>修改微服务的默认分支，微服务下所有流水线配置的默认分支会自动更改。   |
| Repo HTTPS 授权 | 配置授权扩展点以提升代码库操作权限，主要用于微服务变更流水线和部分仓库操作插件。单击“新建授权”，可添加CodeArts Repo HTTPS服务扩展点，配置方法请参考 <a href="#">新建CodeArts服务扩展点</a> 。<br><b>说明</b><br>新建CodeArts Repo HTTPS服务扩展点时，如果用户名密码填写错误，会授权失败并导致流水线执行失败。具体问题请参考 <a href="#">常见问题</a> 。 |
| 别名            | 仓库别名。支持大小写英文字母、数字、“_”，长度不超过128个字符。<br>设置别名后可生成仓库对应的系统参数，如：“别名_REPOSITORY_NAME”，表示仓库名称。生成的参数可在参数设置页面查看，并在流水线中通过“\${参数名}”进行引用。  |
| 基于变更触发        | 开启“基于变更触发”开关的流水线即为变更流水线，在微服务流水线列表中，变更流水线会被标识为  。<br><b>说明</b><br>单个微服务下仅允许存在一条基于变更触发的流水线。  |

| 参数项 | 说明            |
|-----|---------------|
| 描述  | 长度不超过1024个字符。 |

**步骤6** 单击“下一步”，进入“选择模板”页面，根据需要选择对应模板，在模板基础上快速新建任务；也可以选择“空白模板”，即不使用模板，直接新建。

**步骤7** 单击“确定”，根据需要**编排流水线**，然后单击“保存”。

----结束

## 新建微服务变更

管理微服务下的变更信息。

**步骤1** 访问[CodeArts Pipeline](#)首页，进入项目下流水线。

**步骤2** 单击“微服务”，进入微服务列表页面。

**步骤3** 单击微服务名称，进入微服务“概览”页面。

**步骤4** 单击“变更”，进入微服务变更列表页面。

展示微服务下所有变更信息，单击“所有变更”，选择“我的变更”可以筛选出当前登录用户创建的变更信息。

**步骤5** 单击“新建变更”，进入“基本信息”页面，配置相关参数。

表 9-4 新建变更

| 参数项   | 说明   |
|-------|--|
| 变更标题  | 输入变更标题。长度不超过256个字符。  |
| 代码仓   | 微服务关联的代码仓库，不可更改。   |
| 代码分支  | 本次变更新需求的开发分支，可从默认分支拉取新分支，也可关联其他已有分支。<br><b>说明</b><br>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。 |
| 关联工作项 | 支持关联需求管理中开始态和进行态的工作项。  |

**步骤6** 单击“确定”，完成变更创建并进入“变更详情”页。

页面展示变更概览信息、关联工作项、操作历史，还可以对变更执行提交发布、退出发布、撤销变更操作。

图 9-1 变更详情



### 说明

- 变更的发布生命周期为：开发中 > 待发布 > 发布中 > 已发布。
- 阶段状态为“开发中”的变更，单击“编辑工作项”，可以重新选择工作项。

以下详细介绍提交发布、退出发布、撤销变更具体操作：

#### • 提交发布

针对阶段状态为“开发中”的变更，单击“提交发布”，弹出“提交变更”对话框。

- 如果微服务下不存在变更流水线，请根据提示[新建变更流水线](#)。
- 如果微服务下存在变更流水线，直接单击“确定”，可将变更提交到变更流水线的发布列表。

提交发布后，变更状态由“开发中”变为“待发布”。

#### • 退出发布

针对阶段状态为“待发布”或“发布中”的变更，单击“退出发布”，可将变更从变更流水线的发布列表退出，变更状态变为“开发中”。

### 说明

阶段状态为“发布中”的变更，如果变更流水线正在运行，则无法退出发布。

#### • 撤销变更

针对阶段状态为“开发中”的变更，单击“撤销变更”，弹出“撤销变更”对话框。

单击“确定”，即可将变更状态变为“已撤销”，同时变更被删除。

----结束

## 执行变更流水线

- 步骤1** 访问[CodeArts Pipeline](#)首页，进入项目下流水线。
- 步骤2** 单击“微服务”，进入微服务列表页面。
- 步骤3** 单击微服务名称，进入微服务“概览”页面。
- 步骤4** 单击“流水线”，进入微服务流水线列表页面。

**步骤5** 单击流水线名称，进入流水线“执行历史”页面。

**步骤6** 单击右上角“执行”，弹出“执行配置”对话框，进行如下配置。

**图 9-2** 配置变更流水线执行信息



- 变更列表：变更列表展示阶段状态为“待发布”和“发布中”的变更，运行变更流水线时需要选择一个或多个变更。
- 运行参数配置：如果流水线配置了运行时参数，请根据需要设置参数值并保存，可参考[流水线参数应用示例](#)。
- 执行阶段配置：流水线支持选择任务执行，可以选择一个或多个，默认执行全部任务。
- 执行描述：填写本次运行的调试信息。

**步骤7** 配置完成后，单击“执行”，进入变更流水线执行详情页面。

**图 9-3** 执行变更流水线




变更流水线运行时默认添加“合并集成分支”和“合并主干”阶段。

- 合并集成分支：变更流水线会自动从主干分支拉出一条新分支，并将本次运行变更的所有特性分支集成到该新分支，然后通过这条集成分支来运行流水线。

- 合并主干：将集成分支合并回主干分支。

**步骤8** 执行完成后，可查看流水线执行结果。

变更流水线执行成功时，会将本次运行选择的所有变更状态改为“已发布”状态。

- 单击流水线名称，进入“流水线详情”页面
  - 单击流水线源卡片的“查看更多”，弹出运行变更侧滑框，展示本次流水线运行时选择的变更信息。
  - 单击列表内“变更标题”可以跳转到变更详情页面。
- 单击“发布列表”页签，切换到“发布列表”页面
  - 发布列表页面展示所有阶段状态为“待发布”和“发布中”的变更。
  - 列表支持在搜索栏中输入变更名称关键字进行搜索。
  - 单击“操作”列，弹出“退出发布”对话框，单击“确认”会将变更退出变更流水线的发布列表，变更状态重置为“开发中”。

#### 说明

阶段状态为“发布中”的变更，如果变更流水线正在运行，则需要等待变更流水线运行完成或停止变更流水线后可以退出发布。

----结束



# 10 管理流水线扩展插件

## 10.1 流水线插件概述

流水线内置了一系列常用的插件，覆盖构建、检查、部署、测试全领域，供用户在流水线编排时使用。同时提供一套标准的扩展插件接入方式，支持企业将已有工具链接入插件平台，或者基于自身业务需求快速开发、发布插件，并在企业内进行共享和复用，加速企业上云。通过打造可视化、低代码、开放的插件市场，充分利用企业内的开发能力及需求打磨插件生态，实现高复用、低定制DevOps插件市场能力。

### 流水线扩展插件入口

- 入口一
  - a. [访问CodeArts Pipeline首页](#)。
  - b. 进入流水线服务首页，单击“服务 > 扩展插件”，即可进入扩展插件页面。
- 入口二
  - a. [访问CodeArts Pipeline首页](#)。
  - b. 进入项目下流水线，新建或编辑流水线。
  - c. 在“任务编排”页，新建或编辑任务，单击右上方“更多插件”，也可跳转至扩展插件页面。

扩展插件页面展示全部可使用的插件，单击插件卡片可以查看插件的详细信息。

### 流水线扩展插件使用场景

- 使用者可通过扩展插件对接使用各类云服务。如使用CodeArts Pipeline提供的官方插件，如Kubernetes发布。
- 开发者利用官方工具镜像进行插件开发，自行实现业务逻辑。当前支持开发者使用主流多语言进行业务脚本编写，如：shell/nodejs/python/java等。针对业务的特殊环境要求，支持使用自定义执行机配合基础插件（执行容器镜像）提供更加多样的执行方式。
- 使用者可开发扩展插件对接已有的第三方CI/CD工具。

## 10.2 流水线官方插件

流水线当前内置的官方插件如[表10-1](#)。

表 10-1 流水线官方插件

| 插件类型 | 插件名              | 插件使用介绍  |
|------|------------------|---|
| 构建   | Build构建          | 调用编译构建能力进行构建。编译构建服务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。支持编译构建任务一键创建、配置和执行，实现获取代码、构建、打包等活动自动化，实时监控构建状态，让您更加快速、高效地进行云端编译构建。 <a href="#">了解更多</a> 。     |
|      | Build构建（模板）      | 仅支持在流水线模板中配置，该插件会在通过模板生成流水线时，依照当前配置的构建任务模板，自动创建构建任务并配置到所生成的流水线中。  |
| 测试   | TestPlan接口测试     | 调用CodeArts TestPlan能力进行测试。测试计划是面向软件开发者提供的一站式云端测试平台，覆盖测试管理、接口测试，融入DevOps敏捷测试理念，帮助您高效管理测试活动，保障产品高质量交付。 <a href="#">了解更多</a> 。   |
|      | TestPlan接口测试（模板） | 仅支持在流水线模板中配置，该插件会在通过模板生成流水线时，自动创建接口测试任务并配置到所生成的流水线中。  |
| 部署   | Deploy部署         | 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，支持并行部署和流水线无缝集成，实现部署环境标准化和部署过程自动化。 <a href="#">了解更多</a> 。 |
|      | Deploy部署（模板）     | 仅支持在流水线模板中配置，该插件会在通过模板生成流水线时，依照当前配置的部署任务模板，自动创建部署任务并配置到所生成的流水线中。  |
|      | Kubernetes发布     | 将容器镜像部署到云容器引擎（CCE）或原生Kubernetes集群中。支持滚动发布和基于集群Service的蓝绿发布。   |
|      | 云原生发布            | 云原生发布提供了环境级别的服务发布能力，支持对环境进行自定义编排发布（例如：滚动发布和灰度发布）。   |
| 代码检查 | Check代码检查        | 调用CodeArts Check能力进行代码检查。代码检查服务（CodeArts Check）是基于云端实现代码质量管理的服务，软件开发者可在编码完成后执行多语言的代码静态检查和安全检查，获取全面的质量报告，并提供缺陷的改进建议和趋势分析，有效管控代码质量，降低成本。 <a href="#">了解更多</a> 。           |

| 插件类型 | 插件名           | 插件使用介绍   |
|------|---------------|--|
|      | Check代码检查（模板） | 仅支持在流水线模板中配置，该插件会在通过模板生成流水线时，自动创建代码检查任务并配置到所生成的流水线中。 |
|      | 分支变动检查        | 指定目标分支，若当前流水线运行分支落后于指定分支，则流水线运行失败。                   |
| 通用   | 创建仓库标签        | 为代码仓创建标签并推送。   |
|      | 子流水线          | 配置并调用项目下其他流水线任务。                                     |
|      | Jenkins任务     | 调用Jenkins任务以扩展并实现自定义动作。                              |
|      | 挂起流水线         | 配置以挂起当前流水线。  |
|      | 延时执行          | 暂停流水线运行一段时间或暂停至指定时间点，用户可以手动继续、终止流水线运行或进行至多3次的额外延时。   |
|      | 人工审核          | 以单签或会签的方式配置人工审核任务，配置后将以邮件通知的方式提醒审核人审核。               |
|      | 下载仓库          | 下载流水线源中配置的代码库，可结合执行shell命令、Maven构建等插件使用。             |
|      | 执行shell       | 用于执行shell命令。   |
|      | 钉钉机器人         | 在流水线上配置钉钉机器人的webhook，向机器人发送自定义内容。                    |
|      | 飞书机器人         | 在流水线上配置飞书机器人的webhook，向机器人发送自定义内容。                    |
|      | 微信机器人         | 在流水线上配置企业微信机器人的webhook，向机器人发送自定义内容。                  |
|      |               | 上传文件至OBS   |
|      | OBS文件下载       | 从OBS下载文件至本地。   |
| 微服务  | 创建集成分支        | 基于微服务默认分支创建集成分支，用于微服务变更流水线运行使用，由变更流水线自动配置，不支持手动配置。   |
|      | 合并集成分支        | 在微服务变更流水线运行期间，将变更特性分支合并到集成分支，由变更流水线自动配置，不支持手动添加。     |
|      | 合并主干          | 在微服务变更流水线运行期间，将集成分支合并回微服务默认分支，由变更流水线自动配置，不支持手动添加。    |
| 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。                                    |

## 10.3 图形化新建流水线自定义插件

### 新建流水线自定义插件

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 进入流水线服务首页，单击“服务 > 扩展插件”。

**步骤3** 单击 ，开始注册插件。

**步骤4** 填写基本信息，相关参数说明如表10-2所示。

表 10-2 自定义插件基本信息

| 参数项  | 说明   |
|------|--|
| 插件图标 | 插件封面图，不上传将会自动生成系统图标。支持png、jpeg、jpg格式，文件大小不超过512KB，建议128*128px。 |
| 名称   | 插件在插件市场和流水线显示的名称。支持空格、中文、大小写英文字母、数字、“-”、“_”、“.”，长度不超过50字符。     |
| 唯一标识 | 插件的唯一标识，设置后不可修改，建议设置成具有实际含义的内容。支持大小写英文字母、数字、“-”、“_”，长度不超过50字符。 |
| 插件类型 | 插件的类型，支持构建、代码检查、测试、部署、通用共5种类型，设置后不可修改。                         |
| 插件描述 | 插件描述将展示在插件市场中，介绍插件主要用途和功能，设置后可修改。长度不超过1000字符。                  |

**步骤5** 单击“下一步”，进入“版本信息”页面，填写插件版本和版本描述。

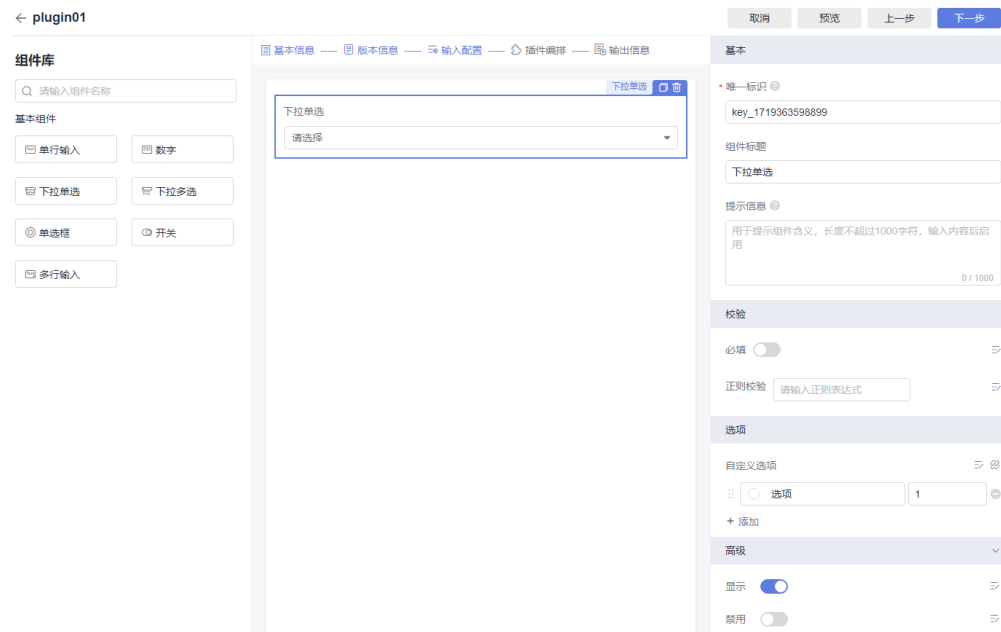
#### 说明

- 插件版本号，支持填写3组0-99的数字。
- 该页面填写的信息将作为插件版本信息展示，发布后不支持修改，请谨慎填写。

**步骤6** 单击“下一步”，进入“输入配置”页面，根据需要编排组件。

插件平台支持低代码化生成前端页面，拖拽组件生成可视化表单，打通流水线上下文数据，减少插件开发成本。平台默认提供了单行输入、数字、下拉单选、下拉多选、单选框、开关、多行输入等多款组件。

图 10-1 编排组件



将左侧组件拖拽到中间布局区域，单击具体组件，右侧区域展示组件参数配置，参数说明如下：

表 10-3 组件参数说明

| 类别 | 参数名   | 说明   | 适用控件                        |
|----|-------|--|-----------------------------|
| 基本 | 唯一标识  | 组件唯一标识，用于插件业务逻辑实现过程中获取组件输入时使用。支持大小写英文字母、数字、“-”、“_”，长度不超过200字符。 | 全部控件。                       |
|    | 组件标题  | 用于标识组件，该标题会展示在流水线任务编排页面，长度不超过140字符。                            | 全部控件。                       |
|    | 提示信息  | 组件标题旁的提示信息，如：组件的含义。长度不超过1000字符。                                | 全部控件。                       |
|    | 预设提示语 | 组件输入框内的提示信息，如：输入规范。  | 单行输入。                       |
|    | 精度    | 设置控件数值支持的小数位，输入范围为0~4。   | 数字。                         |
|    | 默认值   | 用于设置组件的默认值。  | 单行输入、数字、开关、多行输入、Metrics。    |
| 校验 | 必填    | 设置控件内容是否必填，支持设置错误提示信息。   | 单行输入、数字、下拉单选、下拉多选、单选框、多行输入。 |





| 类别 | 参数名   | 说明   | 适用控件                    |
|----|-------|--|-------------------------|
|    | 正则校验  | 设置控件输入内容校验，支持设置错误提示信息。   | 单行输入、数字、下拉单选、下拉多选、多行输入。 |
|    | 字符数限制 | 用于设置组件的最大字符数限制。  | 多行输入。                   |
| 选项 | 自定义选项 | <p>设置控件的选项范围。单击 <b>+</b> 添加，新增选项，单击 <b>-</b>，删除选项。</p> <ul style="list-style-type: none"> <li>选项名称：插件配置时的界面显示内容。</li> <li>选项值：插件运行时实际下发的值。</li> </ul> <p>除手动配置外，还支持通过以下方式设置可选范围：</p> <ul style="list-style-type: none"> <li>通过接口获取选项：配置webapi以获取选项内容。单击右侧 ，弹出“API接口”对话框，启用后即可配置，如图10-2，API接口参数说明请参考表10-4。</li> <li>通过上下文获取选项：数据源当前支持代码源和构建任务，分别获取流水线运行时的代码源url和当前插件之前的所有构建任务ID信息。单击“自定义选项”旁的 ，弹出“上下文”对话框，启用后即可配置。</li> </ul> | 下拉单选、下拉多选、单选框。          |
| 高级 | 显示    | 设置控件是否可见（默认显示），单击右侧  ，可在弹窗中配置按条件显示。   | 全部控件                    |
|    | 禁用    | 设置控件是否禁用（默认不禁用），单击右侧  ，可在弹窗中配置按条件禁用。  | 全部控件                    |

图 10-2 通过接口获取选项

### API接口 ×

启用

联动属性

请选择 ▼

\* URL

GET ▼ 请输入URL

返回数据路径

选项值 ?      选项名称 ?

选项值 (value) , 输入内容后启用      选项名称 (label) , 输入内容后启用

Params参数

+ 添加

Header参数

+ 添加

远程搜索

远程搜索字段

远程搜索传参

Params参数     Body参数

表 10-4 API 接口参数说明

| 参数项  | 说明   |
|------|--|
| 启用   | 控制是否通过接口获取选项。  |
| 联动属性 | 建立其他所选组件和接口的联动关系，可以进行参数传递，当组件的值更新时，会使用新值作为参数对接口进行重新调用。 |

| 参数项      | 说明   |
|----------|--|
| URL      | 仅支持HTTPS协议。  |
| 返回数据路径   | 使用的控件需为列表数据。如以下响应体示例的返回数据路径为“result.parameters”。<br><pre> {   "result": {     "total": 2,     "parameters": [       {         "id": 3353753,         "name": "parameters01"       },       {         "id": 3353697,         "name": "parameters02"       }     ]   },   "status": "success" } </pre> |
| 选项值      | 填写返回数据路径对应列表中的字段，插件运行时实际下发的对应值。  |
| 选项名称     | 填写返回数据路径对应列表中的字段，插件配置时的界面显示内容。   |
| Params参数 | API接口请求体的Params参数。   |
| Header参数 | API接口请求体的Header参数。   |
| 远程搜索     | 开启后，可添加远程搜索字段，使用插件时在输入框中填写值，会使用所填的值作为远程搜索字段的值对接口进行重新调用。 <ul style="list-style-type: none"> <li>Params参数：搜索字段的参数类型为API接口请求体的Params参数。</li> <li>Body参数：搜索字段的参数类型为API接口请求体的Body参数。</li> </ul>   |

**步骤7** 单击“下一步”，进入“插件编排”页面，可添加“下载制品产物”插件和“执行Shell”插件。

- 执行Shell：用于执行用户输入的Shell命令。  
在插件命令中填入Shell命令，流水线调用插件时即可执行该命令。

#### 📖 说明

- 插件命令是插件实际业务逻辑实现的过程，更多输入输出配置，请参见[开发流水线自定义插件Shell命令](#)。
- 下载制品产物：用于下载构建生成的制品产物。  
填写待下载的制品产物地址，格式为：\${jobs.build\_job.artifacts.demo}，用于标识构建任务的产物。
    - jobs：固定目录，保持不变。
    - build\_job：Build构建插件对应的任务ID，需要手动替换，如[图10-3](#)。
    - artifacts：固定目录，保持不变。
    - demo：对应Build构建插件的“产物标识”，如[图10-4](#)。



图 10-3 Build 构建任务 ID



图 10-4 Build 构建产物标识



**步骤8** 单击“下一步”，进入“输出信息”页面，单击“新增配置”，即可新增一条输出信息，包括output、metric、link和table四种类型，流水线执行完成后，在详情页面进入任务结果页面，查看对应插件的任务结果，会根据类型展示相应输出信息。

- output: 在任务结果页面的“其他”卡片展示，结合Shell命令用于数据输出。
- metric: 在任务结果页面的“其他”卡片展示，用于输出指标解析，输出信息可以在规则中通过该插件进行引用，并最终应用到流水线。
- link: 在任务结果页面的“链接”卡片展示，其值是一个链接，单击可跳转至对应页面。
- table: 在任务结果页面的“表格数据”卡片展示，其值是一个对象数组，以表格形式展示数组信息。

**步骤9** 配置完成后，单击“发布草稿”或“发布”，完成插件新建流程。

- 发布草稿

单击“发布草稿”，作为测试版本进行发布。

- 发布草稿后，创建者可在流水线配置该草稿版本进行插件调试，测试完毕后，草稿版本可以发布为正式版本，供当前租户账号下其他成员使用。
- 草稿版本均带有“草稿”标识。
- 只允许存在一个草稿版本，且存在草稿版本时，不允许新建版本，必须将草稿版本发布为正式版本或删除后才可新建。

- 正式发布

单击“发布”，作为正式版本进行发布，正式版本发布后，会有一个独立的版本号，当前租户账号下所有成员都可在流水线中使用该插件版本。

---结束

## 开发流水线自定义插件 Shell 命令

注册插件或新建插件版本时，支持用户在插件命令输入Shell命令以实现插件业务逻辑，编写命令过程中，通常会涉及和流水线执行过程中的各种数据进行交互。本节从插件的代码开发输入输出，介绍插件业务的代码实现。

- 自定义插件的数据输入

插件代码开发过程中获取的外部信息主要包括三部分：低代码界面配置输入、流水线运行参数、其他外部信息。

- 低代码界面配置输入：在插件命令中，使用环境变量获取低代码界面输出，如：echo \${控件唯一标识}。
- 流水线运行参数：流水线的部分运行参数会下发到环境变量中，当前已提供的环境变量如下：

**表 10-5** 流水线环境变量

| 变量              | 描述            |
|-----------------|---------------|
| STEP_NAME       | 当前流水线的运行步骤名称。 |
| STEP_ID         | 当前流水线的运行步骤ID。 |
| PLUGIN_VERSION  | 当前自定义插件版本。    |
| PIPELINE_ID     | 流水线ID。        |
| PIPELINE_RUN_ID | 流水线执行ID。      |
| PLUGIN_NAME     | 插件名称。         |
| PROJECT_ID      | 当前项目ID。       |
| JOB_ID          | 当前流水线运行任务ID。  |

| 变量              | 描述   |
|-----------------|--|
| RESULT_MSG_PATH | 插件执行结果文件存放目录，在该目录下写入\${STEP_ID}_result.json和\${STEP_ID}_metrics.json以上报执行结果到流水线。 |

- 其他外部信息：通过git、wget、curl等方式自主与外部数据进行交互，以获取所需要的数据或信息。
- 自定义插件的数据输出  
业务执行完毕后，自定义插件可以读取指定路径文件信息，获取插件的指标数据输出，具体实现步骤如下：
  - a. 在插件的低代码界面，通过配置输出信息，用于配置插件的输出阈值。
  - b. 在插件的代码开发过程中，通过向指定路径输出“\${STEP\_ID}\_result.json”和“\${STEP\_ID}\_metrics.json”文件，实现插件运行完成后，对相应的指标值进行解析。

表 10-6 输出文件说明

| 输出文件   | 使用说明   |
|--|--|
| \$<br>{RESULT_MSG_PATH}/\${STEP_ID}_result.json  | 输出 {"par1":123, "par2":456} 格式的文本文件，流水线运行完成后会将该结果展示在相应任务的任务结果当中。<br><b>说明</b><br>当前仅“代码检查”类型的插件支持将该结果进行展示。   |
| \$<br>{RESULT_MSG_PATH}/\${STEP_ID}_metrics.json | 输出 {"par1":123, "par2":456} 格式的文本文件，配合插件“输入配置”中的metrics控件使用。流水线在插件步骤运行完成后，会根据该插件配置的metrics控件中的阈值和“\${STEP_ID}_metrics.json”文件中的实际内容，进行比对和解析，并最终用于流水线阶段准出的拦截依据。比对解析逻辑如下： <ul style="list-style-type: none"> <li>• 解析时，服务会忽略未配置在metrics控件中的key值。</li> <li>• 当metrics控件中配置的key值无法在“\${STEP_ID}_result.json”文件中找到对应值时，对应阈值会用value值进行填充。</li> </ul> |

示例介绍：实现对外输出“par1”和“par2”指标用于任务结果展示，输出“par3”和“par4”指标用于准出拦截，示例代码如下：

```
# Optionally, construct the extension output.
echo '{"par1":100,"par2":200}' > ${RESULT_MSG_PATH}/${STEP_ID}_result.json
echo '{"par3":300,"par4":400}' > ${RESULT_MSG_PATH}/${STEP_ID}_metrics.json
```

- c. 流水线在插件步骤运行完毕后，单击相应步骤卡片，即可查看自定义插件的输出结果。

| 红线   |     |
|------|-----|
| 检查项  | 结果  |
| par3 | 300 |
| par4 | 400 |

如果当前插件配置了相应的策略，并应用到流水线准出条件中，还可以单击准出条件查看阶段拦截情况。

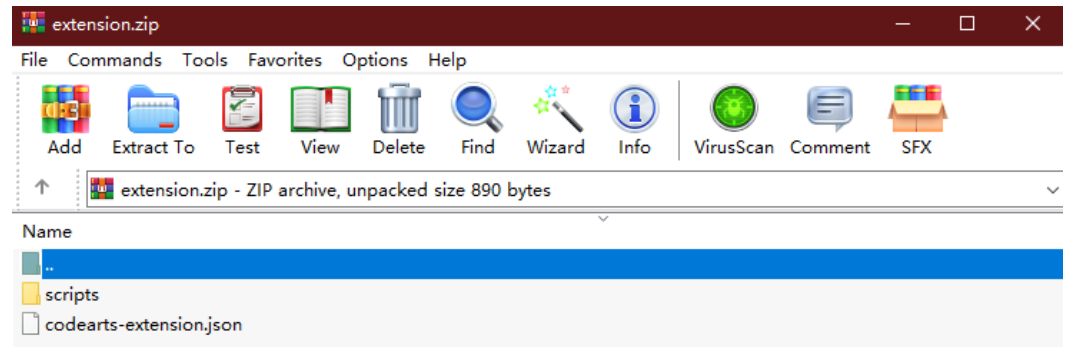
| Group |      |     |    |     |
|-------|------|-----|----|-----|
| 检查项   | 状态   | 当前值 | 比较 | 阈值  |
| par1  | ● 成功 | 100 | =  | 100 |
| par2  | ● 成功 | 200 | =  | 200 |

## 10.4 代码化新建流水线自定义插件

### 准备自定义插件包

#### 插件包结构

图 10-5 插件包结构



#### 文件结构

```
extension.zip      # 插件zip包
|-- scripts       # (可选)脚本文件夹,用于存放包含插件执行逻辑的脚本
| |-- xxx        # 包含插件执行逻辑的脚本
| |-- i18n       # (可选)多语言,存放所有文案信息
| |-- zh-cn      # 中文环境内容
|   |-- resources.json # 对应语言的国际化资源
|   |-- en-us    # 英文环境内容
|   |-- resources.json # 对应语言的国际化资源
|-- codearts-extension.json # (必选)插件执行定义文件,Json格式,包括扩展插件的基本信息+input+execution
```

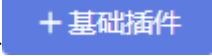
#### 注意事项

- 插件包必须为.zip后缀。
- 插件包根目录下必须包含元数据文件codearts-extension.json，codearts-extension.json文件介绍请参考[codearts-extension.json文件定义](#)。
- resources.json文件只能使用utf-8进行编码，不支持中文标点符号与其他编码形式，若输入则为乱码展示。

## 新建流水线自定义插件

**步骤1** 访问[CodeArts Pipeline](#)首页。

**步骤2** 进入流水线服务首页，单击“服务 > 扩展插件”。

**步骤3** 单击 ，开始注册插件。

**步骤4** 填写基本信息，相关参数说明如[表10-7](#)所示。

表 10-7 自定义插件基本信息

| 参数项  | 说明   |
|------|--|
| 插件图标 | 插件封面图，不上传将会自动生成系统图标，支持png、jpeg、jpg格式，文件大小不超过512KB，建议128*128px。   |
| 名称   | 插件在插件市场和流水线显示的名称。支持空格、中文、大小写英文字母、数字、“-”、“_”、“.”，长度不超过50字符。   |
| 唯一标识 | 自定义插件的唯一标识，需与插件包中codearts-extension.json文件的name字段值匹配，设置后不可修改，建议设置成具有实际含义的内容。支持大小写英文字母、数字、“-”、“_”，长度不超过50字符。<br>插件类型和category字段值对应关系如下： <ul style="list-style-type: none"><li>• 构建：Build。</li><li>• 代码检查：Gate。</li><li>• 部署：Deploy。</li><li>• 测试：Test。</li><li>• 通用：Normal。</li></ul> |
| 插件类型 | 插件的类型，支持构建、代码检查、测试、部署、通用共5种类型，一经设置，不可修改。   |
| 插件描述 | 插件描述将展示在插件市场中，介绍插件主要用途和功能，后期可修改。长度不超过1000字符。   |

**步骤5** 基本信息填写完成后，单击“下一步”，进入“版本管理”页面。


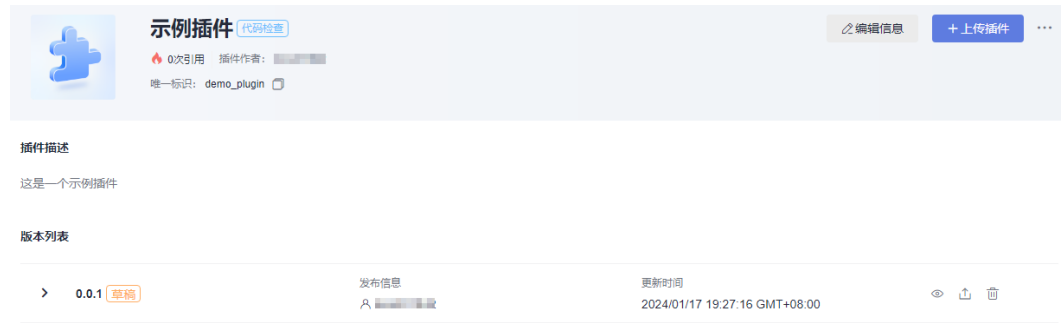
**步骤6** 单击 ，弹出“上传插件”对话框，选择已准备好的插件（插件中已包含输入定义、业务执行脚本等内容），然后上传。上传成功后可以看到带“草稿”标识的版本。

图 10-6 上传插件



**步骤7** 调试插件。

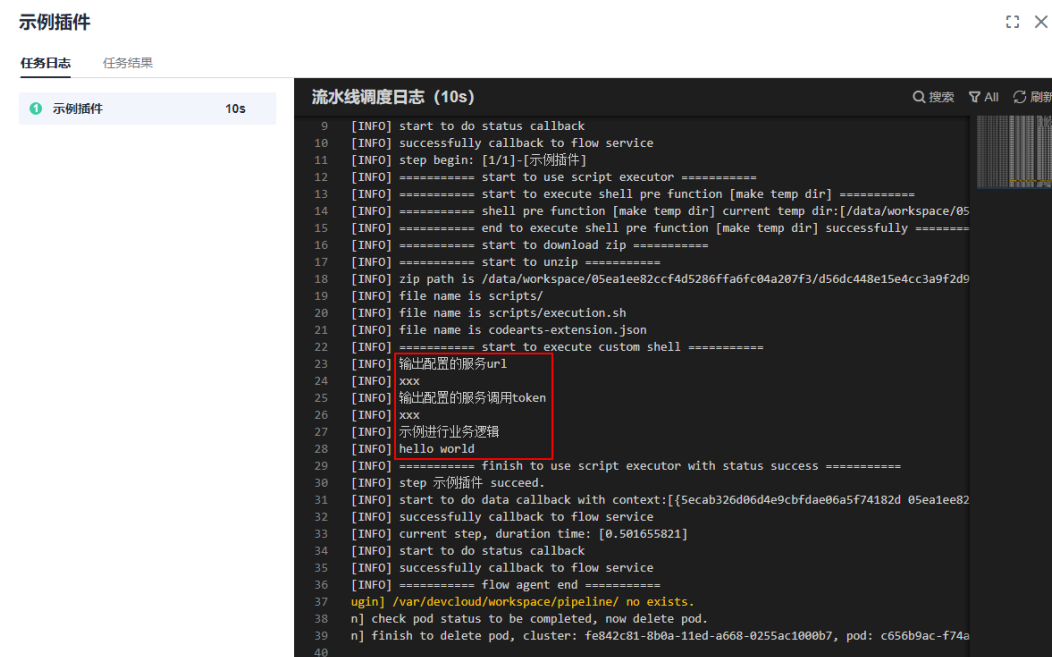
新建流水线任务，在“任务编排”页面新建任务，添加已注册的基础插件，填写参数信息。

图 10-7 调试插件




**步骤8** 保存并执行流水线，执行完成后，单击插件名称，查看执行结果。

图 10-8 查看插件执行结果



步骤9 (可选) 业务逻辑调试无误后, 建议将插件发布为正式版本。

1. 返回到扩展插件页面。
2. 单击刚注册的基础插件, 进入插件“版本管理”页面。
3. 单击版本列表对应插件版本右侧的 , 将此版本发布为正式版本。

草稿版本可以同版本多次覆盖, 但正式版本不可重复更新, 只可以单击右上角“上传插件”重新上传插件新增版本。如下图: 0.0.1版本为正式版本, 0.0.2版本为重新上传后的草稿版本。

图 10-9 发布正式版本



----结束

## codearts-extension.json 文件定义

codearts-extension.json文件示例:

```

{
  "type": "Task",
  "name": "demo_plugin",

```

```

"friendlyName": "示例插件",
"description": "这是一个示例插件",
"category": "Gate",
"version": "0.0.2",
"versionDescription": "0.0.1初始化版本",
"dataSourceBindings": [],
"inputs": [
  {
    "name": "samplestring",           # 如插件业务脚本中使用${samplestring}获取运行者在流
    "type": "input",                 # 不同类型信息对应不同展示功能, 可选项见下文
    "description": "Sample String",  # input条目的描述信息, 用于描述其值含义
    "defaultValue": "00",           # 默认值, required属性为false时, 如不在流水线上重新输
    "required": true,                # true则流水线编辑时必须重新填值, false则不填使用默认值
    "label": "测试输入框",          # input条目在流水线编辑页面显示的名称信息
    "validation": {
      "requiredMessage": "请输入值", # (可选) 如required字段为true, 未填写时的提示信息
      "regex": "[a-zA-Z0-9-_\u4e00-\u9fa5]{1,32}$", # (可选)可填写正则校验的内容
      "regexMessage": "类型错误"    # (可选) 如正则校验失败的提示信息
    }
  }
],
"execution": {
  "type": "Shell",
  "target": "scripts/execution.sh"
},
"outputs": [{
  "name": "okey",                   # 输出名称
  "type": "output",                 # 输出类型, output/metrics
  "description": "描述",
  "prop": {
    "defaultValue": "123"          # 默认值
  }
},
{
  "name": "mkey",
  "type": "metrics",
  "description": "描述",
  "prop": {
    "defaultValue": "213",
    "group": "213"                 # 对应准出规则中的分组名称
  }
}
]
}

```

codearts-extension.json文件参数说明如下：

表 10-8 codearts-extension.json 文件参数说明

| 参数项          | 说明                          |
|--------------|-----------------------------|
| type         | 填写固定值“Task”，标识为一个插件类型。      |
| name         | 请与注册插件时页面填写的基本信息“唯一标识”字段一致。 |
| friendlyName | 请与注册插件时页面填写的基本信息“插件名称”字段一致。 |










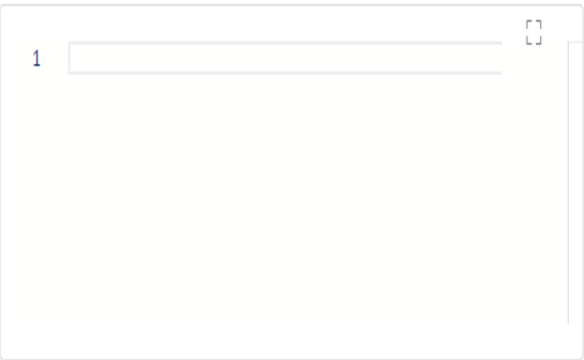

| 参数项                | 说明   |
|--------------------|--|
| category           | <p>请与注册插件时页面填写的基本信息“插件类型”字段一致，包括以下可选值：</p> <ul style="list-style-type: none"> <li>• Build：对应“构建”插件类型。</li> <li>• Test：对应“测试”插件类型。</li> <li>• Gate：对应“代码检查”插件类型。</li> <li>• Normal：对应“通用”插件类型。</li> <li>• Deploy：对应“部署”插件类型。</li> </ul> |
| version            | 插件版本，支持填写3组0-99的数字，如需新增正式版本，请修改此字段。  |
| description        | 插件的描述信息。   |
| versionDescription | 此版本插件的描述信息，建议体现每个版本的差异点。   |
| dataSourceBindings | 此字段暂时未启用，请将值设置为“[]”。   |
| inputs             | 插件输入内容，对应流水线页面插件展示格式，其值可在业务脚本中通过引用环境变量的方式引用。   |
| execution          | 业务插件执行内容，其中type字段为业务脚本语言类型，target字段为执行文件入口，建议放在scripts文件夹下。  |
| outputs            | 插件输出内容，在插件运行结束后写入此处定义值，可对应用作门禁指标metrics，不同的展示结果output。   |

当前支持的inputs类型如下：

表 10-9 全量 inputs 类型

| inputs类型 | 组件名称  | 样例   | extendProp扩展  |
|----------|-------|--|---|
| input    | 单行输入框 | <p>单行输入</p>  | <ul style="list-style-type: none"> <li>• visibleConditions</li> <li>• disabledConditions</li> </ul> |

| inputs类型       | 组件名称  | 样例   | extendProp扩展   |
|----------------|-------|--|--|
| inputNumber    | 数字输入框 | <p>数字</p>      | <ul style="list-style-type: none"> <li>• visibleConditions</li> <li>• disabledConditions</li> </ul>  |
| switch         | 开关    | <p>开关</p>       | <ul style="list-style-type: none"> <li>• visibleConditions</li> <li>• disabledConditions</li> </ul>  |
| singleSelect   | 下拉单选框 | <p>下拉单选</p>    | <ul style="list-style-type: none"> <li>• options</li> <li>• apiType</li> <li>• apiOptions</li> </ul> |
| multipleSelect | 下拉多选框 | <p>下拉多选</p>  | <ul style="list-style-type: none"> <li>• options</li> <li>• apiType</li> <li>• apiOptions</li> </ul> |
| keyValuePair   | 键值对   | <p>键值对</p>   | <ul style="list-style-type: none"> <li>• visibleConditions</li> <li>• disabledConditions</li> </ul>  |
| radio          | 单选框   | <p>单选框</p>   | options  |

| inputs类型   | 组件名称     | 样例   | extendProp扩展   |
|--|----------|--|--|
| timeInterval   | 时间间隔控件   | <p>时间间隔</p>    | <ul style="list-style-type: none"> <li>• visible Conditions</li> <li>• disabledConditions</li> </ul> |
| shell  | shell控件  | <p>Shell</p>  | <ul style="list-style-type: none"> <li>• visible Conditions</li> <li>• disabledConditions</li> </ul> |
| endpoint:<br>:\${XXX}<br>(其中,<br>“XXX”<br>为扩展点<br>“module_id”) | 扩展点下拉单选框 |              | <ul style="list-style-type: none"> <li>• visible Conditions</li> <li>• disabledConditions</li> </ul> |

inputs基础字段说明如下：

表 10-10 inputs 基础字段

| 字段名称          | 含义      | 是否必填 | 备注            |
|---------------|---------|------|---------------|
| name          | 组件唯一标识。 | 是    | 同一基础插件内不允许重复。 |
| label         | 组件标题。   | 是    | -             |
| type          | 组件类型。   | 是    | -             |
| default Value | 初始值。    | 否    | 控件初始默认值，可为空。  |

| 字段名称        | 含义   | 是否必填 | 备注   |
|-------------|--|------|--|
| description | 组件描述。  | 否    | 控件名称右侧问号内描述信息。<br> |
| required    | 是否必填。  | 否    | 带星号为必填。<br>         |
| validation  | 校验信息，是一个对象，包含 requiredMessage、regex、regexMessage 三个属性。<br><pre>{   requiredMessage:   ", // 必填项提示语   regex: ", //   正则校验   regexMessage:   " // 正则校验失败的提示语 }</pre> | 否    |                    |
| extend Prop | 扩展字段<br><pre>{   visibleConditions:   [],   disabledConditions: []   ... }</pre>   | 否    | 针对特殊组件的功能扩展使用，详见表 10-11。   |

extendProp 扩展功能说明如下：

表 10-11 extendProp 扩展功能

| 字段名称               | 含义              | 是否必填 | 备注   |
|--------------------|-----------------|------|--|
| visibleConditions  | 按条件显示。          | 否    | <p>格式如下，可包含多个显示条件：<br/>[{};{};{};...]</p> <p>示例：<br/>[{}comp:'key_001',symbol:'===', value: 'xxx']</p> <p>表示如果当前控件为A，当其他控件中存在唯一标识为key_001的控件B，且控件B的值满足等于xxx这个条件时，则当前控件A显示。</p> <p>symbol包含如下四种：</p> <ul style="list-style-type: none"> <li>• ===：等于</li> <li>• !==：不等于</li> <li>• empty：为空</li> <li>• notEmpty：非空</li> </ul> |
| disabledConditions | 按条件禁用。          | 否    | <p>格式如下，可包含多个禁用条件：<br/>[{};{};{};...]</p> <p>示例：<br/>[{}comp:'key_002',symbol:'!==', value: 'yyy']</p> <p>表示如果当前控件为A，当其他控件中存在唯一标识为key_002控件B，且控件B的值满足不等于yyy这个条件时，则当前控件A禁用。</p> <p>symbol包含如下四种：</p> <ul style="list-style-type: none"> <li>• ===：等于</li> <li>• !==：不等于</li> <li>• empty：为空</li> <li>• notEmpty：非空</li> </ul> |
| options            | 固定下拉选项，类型为list。 | 否    | <p>示例：<br/>[{}label: '选项1',value: 1},{label: '选项2',value: 2}]</p>  |

| 字段名称       | 含义  | 是否必填 | 备注   |
|------------|---|------|--|
| apiType    | <p>下拉选项类型：</p> <ul style="list-style-type: none"> <li>fixed：使用options内的值作为下拉选项。</li> <li>api：通过接口请求，需配置apiOptions。</li> </ul> | 否    | 不填则为fixed。   |
| apiOptions | JSON体，包含api接口使用的各个参数  | 否    | <p>示例：</p> <pre>{   "body":{"xxx":111},"header":{"yyy":222},"linkedFields":   ["key_001"],"method":"POST","params":   {"zzz":333},"remote":true,"remoteName":"xxx","remoteQueryField":   "body","responseUrl":"data","label":"name","value":"id","url":"https://sss/lll/mmm"}</pre> <p>JSON解析后：</p> <pre>{   body: {xxx:111},           // 参数为接口对应传参字段   header: {yyy: 222},       // 请求头字段   params: {zzz: 333},       // 参数为接口对应传参字段   linkedFields: ['key_001], // 为联动其他控件字段，当其他控件值改变时，接口会重新触发调用，并清空选项   method: 'POST',          // 请求方式：POST/GET   remote: true,             // 是否开启远程搜索   remoteName: 'tt',        // 开启远程搜索时搜索字段   remoteQueryField: 'body', // 远程搜索字段传参形式   responseUrl: 'data',     // 解析返回值获取选项list的路径   label: 'name',           // 下拉框控件中显示label对应的参数   value: 'id',             // 下拉框控件中实际value对应的参数   url: 'https://sss/lll/mmm' // 接口url }</pre> |

## 10.5 执行镜像命令

通过“执行镜像命令”插件，可以将用户在SWR上的公开镜像下载到自定义执行机并启动。

### 约束与限制

“执行镜像命令”插件只支持自定义执行机。


## 配置方法

**步骤1** 在**编排流水线任务**时，添加“执行镜像命令”插件。

图 10-10 执行镜像命令



**步骤2** 根据需要配置任务参数，参数说明如下：

| 参数项     | 说明  |
|---------|---|
| 名称      | <p>插件名称。</p> <ul style="list-style-type: none"> <li>支持中文、大小写英文字母、数字、“-”、“_”、“.”、“;”、“:”、“.”、“/”、“(”、“)”、“(”、“)”及空格。</li> <li>字符长度范围为1-128。</li> </ul>   |
| 镜像属性    | 当前只支持公开镜像。  |
| SWR镜像地址 | <p>待下载的SWR镜像的地址。获取方式如下：</p> <ol style="list-style-type: none"> <li>登录<a href="#">容器镜像服务</a>。</li> <li>在导航栏单击“我的镜像”，单击“镜像名称”进入镜像详情页面。</li> <li>单击“下载指令”列的，复制镜像下载指令，其中，“docker pull”后面部分即为镜像地址。</li> </ol> |
| 启动命令    | 容器启动命令。输入Docker命令，以便在容器中运行特定的应用程序或脚本。   |

----结束

# 11 新建 CodeArts 服务扩展点

## 操作场景

服务扩展点是CodeArts的一种扩展插件，为CodeArts提供连接第三方服务的能力。

当CodeArts中的一些任务需要连接到远程第三方服务，进行一些获取数据操作时，例如连接第三方GitHub仓库获取项目源码、连接第三方Jenkins服务执行Jenkins任务等，可以通过新建扩展点来实现与相应服务的连接。

CodeArts支持以下扩展点。

表 11-1 服务扩展点

| 类型                | 使用场景  |
|-------------------|---|
| Docker repository | 用于连接Docker镜像仓库，连接成功后可以通过部署服获取仓库中的Docker镜像。                              |
| Jenkins           | 用于连接Jenkins服务，连接成功后可以在流水线中调用并执行Jenkins服务中所包含的任务。                        |
| Kubernetes        | 用于连接Kubernetes集群，连接成功后可对Kubernetes集群下发部署任务。                             |
| nexus repository  | 用于连接第三方私有Maven仓库，连接成功后可以通过构建任务获取该仓库中文件信息。                               |
| 通用Git             | 用于连接第三方Git仓库，连接成功后可以在流水线、构建等服务中获取该仓库的分支等信息。                             |
| 码云Git             | 用于连接码云Git账号，连接成功后可以在流水线、构建等服务中获取该账号的仓库、分支等信息。                           |
| GitHub            | 用于连接Github账号，连接成功后可以在流水线、构建等服务中获取该账号的仓库、分支等信息。                          |
| IAM账户             | 用于委托自己账号的AK/SK给需要执行任务的账号，在该账号执行部署任务的时候可以通过AK/SK获得被委托的账号的token执行更高权限的任务。 |




| 类型                  | 使用场景  |
|---------------------|---|
| CodeArts Repo HTTPS | 用于授权CodeArts服务对托管的Repo仓库进行代码下载、分支创建、分支合并、代码提交等操作。当前主要用于流水线服务的微服务变更功能模块及其相关插件。 |
| Gerrit              | 用于连接第三方Gerrit仓库，连接成功后可以在流水线、构建等服务中获取该仓库代码。                                    |
| GitCode             | 用于连接GitCode账号，连接成功后可以在流水线、构建等服务中获取该账号的仓库、分支等信息。                               |

## 前提条件

- 在CodeArts项目中拥有扩展点的编辑权限。
- 待接入的第三方服务能通过公网环境无限制地访问。

## 新建 Docker repository 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称，进入项目。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“Docker repository”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-2 新建 Docker repository 服务扩展点


| 配置项  | 是否必填 | 说明                                       |
|------|------|--|
| 连接名称 | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。 |
| 仓库地址 | 是    | 输入待连接的Docker仓库的地址。支持http、https协议地址。      |
| 用户名  | 是    | 输入待连接的Docker仓库的用户名。不超过256个字符。            |
| 密码   | 是    | 输入待连接的Docker仓库的密码。不超过256个字符。             |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建 Jenkins 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录**CodeArts控制台**，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“Jenkins”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-3 新建 Jenkins 服务扩展点


| 配置项        | 是否必填 | 说明  |
|------------|------|---|
| 连接名称       | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。                        |
| Server URL | 是    | 输入待连接的Jenkins服务的地址。支持输入格式为“http://ip:端口号”或者“https://ip:端口号”的地址。 |
| 用户名        | 是    | 输入待连接的Jenkins服务的用户名。不超过300个字符。                                  |
| 密码         | 是    | 输入待连接的Jenkins服务的密码。不超过300个字符。                                   |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建 Kubernetes 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录**CodeArts控制台**，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“Kubernetes”。

**步骤5** 在弹框中配置以下信息，单击“验证并确定”。

表 11-4 新建 Kubernetes 服务扩展点


| 配置项            | 是否必填 | 说明   |
|----------------|------|--|
| 连接名称           | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。             |
| Kubernetes URL | 是    | 输入待连接的集群的server地址，可通过“kubeconfig.json”中搜索“server”获取。 |
| Kubeconfig     | 是    | 输入待连接的集群的“kubeconfig.json”文件的全部内容。                   |

**步骤6** 验证成功，关闭弹窗后页面中显示新建的扩展点名称。

----结束

## 新建 nexus repository 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“nexus repository”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-5 新建 nexus repository 服务扩展点


| 配置项  | 是否必填 | 说明                                       |
|------|------|--|
| 连接名称 | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。 |
| 仓库地址 | 是    | 输入待连接的nexus仓库的地址。支持http、https协议地址。       |
| 用户名  | 是    | 输入待连接的nexus仓库的用户名。不超过300个字符。             |
| 密码   | 是    | 输入待连接的nexus仓库的密码。不超过300个字符。              |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建通用 Git 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“通用Git”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-6 新建通用 Git 服务扩展点


| 配置项             | 是否必填 | 说明                                       |
|-----------------|------|--|
| 连接名称            | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。 |
| Git仓库Url        | 是    | 输入待连接的Git仓库的HTTPS地址。                     |
| 用户名             | 否    | 输入待连接的Git仓库的用户名。不超过300个字符。               |
| 密码或Access Token | 否    | 输入待连接的Git仓库的密码。不超过300个字符。                |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建码云 Git 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“码云Git”。

**步骤5** 在弹框中配置以下信息，单击“授权并确定”。

**表 11-7** 新建码云 Git 服务扩展点


| 配置项  | 是否必填 | 说明   |
|------|------|--|
| 连接名称 | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。   |
| 验证方式 | 是    | 支持两种验证方式。 <ul style="list-style-type: none"> <li>OAuth认证：选择此方式，单击“授权并确定”后，需要登录码云进行手动授权。</li> <li>AccessToken认证：选择此方式，需要输入通过码云获取的私人令牌。获取方法请参考<a href="#">码云官网</a>。</li> </ul> |

**步骤6** 授权成功，关闭弹窗后页面中显示新建的扩展点名称。

---结束

## 新建 GitHub 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“GitHub”。

**步骤5** 在弹框中配置以下信息，单击“授权并确定”。

**表 11-8** 新建 GitHub 服务扩展点


| 配置项  | 是否必填 | 说明  |
|------|------|---|
| 连接名称 | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。  |
| 验证方式 | 是    | 支持两种验证方式。 <ul style="list-style-type: none"> <li>OAuth认证：选择此方式，单击“授权并确定”后，需要登录GitHub进行手动授权。</li> <li>AccessToken认证：选择此方式，需要输入通过GitHub获取的AccessToken。获取方法请参考<a href="#">GitHub官网</a>。</li> </ul> |

**步骤6** 授权成功，关闭弹窗后页面中显示新建的扩展点名称。

----结束

## 新建 IAM 账户服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“IAM账户”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-9 新建 IAM 账户服务扩展点


| 配置项               | 是否必填 | 说明   |
|-------------------|------|--|
| 连接名称              | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。           |
| Access Key Id     | 是    | 输入通过“我的凭证”页面获取的AK。获取方法请参考 <a href="#">新增访问密钥</a> 。 |
| Secret Access Key | 是    | 输入通过“我的凭证”页面获取的SK。获取方法请参考 <a href="#">新增访问密钥</a> 。 |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建 CodeArts Repo HTTPS 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。



**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“CodeArts Repo HTTPS”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

**表 11-10** 新建 CodeArts Repo HTTPS 服务扩展点


| 配置项             | 是否必填 | 说明   |
|-----------------|------|--|
| 连接名称            | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。   |
| CodeArts Repo地址 | 是    | 输入待连接的CodeArts Repo仓库的HTTPS地址。<br>进入目标仓库，单击“克隆/下载”，在弹框中选择“用HTTPS克隆”，即可获取仓库地址。  |
| 用户名             | 否    | 输入待连接的CodeArts Repo仓库的HTTPS用户名。<br>在导航栏中单击用户名  ，选择“个人设置”。在“代码托管 > HTTPS密码”页面可以获取用户名。 |
| 密码              | 否    | 输入待连接的CodeArts Repo仓库的HTTPS密码。<br>在导航栏中单击用户名  ，选择“个人设置”。在“代码托管 > HTTPS密码”页面可以获取密码。   |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建 Gerrit 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“Gerrit”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-11 新建 Gerrit 服务扩展点


| 配置项      | 是否必填 | 说明                                       |
|----------|------|--|
| 连接名称     | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。 |
| Gerrit地址 | 是    | 输入待连接的Gerrit仓库的地址。支持https协议地址。           |
| 用户名      | 否    | 输入待连接的Gerrit仓库的用户名。不超过300个字符。            |
| 密码       | 否    | 输入待连接的Gerrit仓库的密码。不超过300个字符。             |

**步骤6** 新建成功，页面中显示新建的扩展点名称。

----结束

## 新建 GitCode 服务扩展点

**步骤1** 进入CodeArts首页。

1. 登录[CodeArts控制台](#)，单击，选择区域。
2. 单击“前往工作台”。

### 说明

如果当前账号采用的是历史计费模式（详情请参见[历史计费模式说明](#)），则单击“立即使用”。

**步骤2** 单击目标项目名称。

**步骤3** 在导航中依次选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤4** 单击“新建服务扩展点”，在下拉列表中选择“GitCode”。

**步骤5** 在弹框中配置以下信息，单击“确定”。

表 11-12 新建 GitCode 服务扩展点

| 配置项   | 是否必填 | 说明   |
|-------|------|--|
| 连接名称  | 是    | 自定义。支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。               |
| Token | 是    | 输入GitCode获取的个人访问密钥。获取方法请参考 <a href="#">GitCode官网</a> 。 |

**步骤6** 授权成功，关闭弹窗后页面中显示新建的扩展点名称。

----结束



# 12 查看审计日志

云审计服务（Cloud Trace Service，CTS）是华为云安全解决方案中专业的日志审计服务，可以记录CodeArts Pipeline相关的操作事件，便于日后的查询、审计和回溯。

开通云审计服务后，系统开始记录CodeArts Pipeline资源的操作。云审计服务管理控制台保存最近7天的操作记录。

## CTS 支持的 CodeArts Pipeline 操作

表 12-1 云审计服务支持的 CodeArts Pipeline 操作列表

| 操作名称    | 资源类型     | 事件名称   |
|---------|----------|--------|
| 执行流水线任务 | pipeline | run    |
| 修改流水线任务 | pipeline | update |
| 创建流水线任务 | pipeline | create |
| 删除流水线任务 | pipeline | delete |
| 停止流水线任务 | pipeline | stop   |

## 查看审计事件

如果需要在云审计服务管理控制台查询CodeArts Pipeline的相关操作事件，请参考[查看审计事件](#)。

# 13 参考

## 13.1 流水线上下文

### 13.1.1 流水线上下文

流水线上下文存储了流水线运行的实例、变量、任务等信息，每个上下文都是一个包含各种属性的对象。流水线上下文主要包括以下四类：

表 13-1 流水线上下文

| 上下文      | 类型     | 说明                   |
|----------|--------|----------------------|
| pipeline | object | 流水线运行相关的信息。          |
| sources  | object | 流水线运行的源相关的信息。        |
| env      | object | 流水线运行的自定义参数相关的信息。    |
| jobs     | object | 流水线运行的已执行完成的任务相关的信息。 |

### 流水线上下文引用形式

```
`${<context>.<attribute_name>}`
```

其中，context表示流水线上下文，attribute\_name表示属性。

## 流水线上下文属性介绍

表 13-2 流水线上下文属性

| 上下文              | 属性                    | 类型     | 说明   | 示例   |
|------------------|-----------------------|--------|--|--|
| pipeline context | pipeline              | object | 流水线运行相关的所有信息，此对象包含的属性包括：<br>project_id、<br>pipeline_id、<br>run_number、<br>timestamp、<br>trigger_type和<br>run_id。 | <ul style="list-style-type: none"> <li>内容示例<br/>如下示例为一次手动运行的流水线包含的pipeline context信息。<br/> <pre>{   "project_id": "6428c2e2b4b64affa14ec80896695c49",   "pipeline_id": "f9981060660249a3856f46c2c402f244",   "run_number": "168",   "timestamp": "20231016000004",   "trigger_type": "Manual",   "run_id": "c2f507f93510459190b543e47f6c9bec" }</pre> </li> <li>使用示例<br/>如需在某个任务的输入中获取本次流水线运行的触发方式，可使用如下语法来获取。<br/> <pre>{{ pipeline.trigger_type }}</pre> </li> </ul> |
|                  | pipeline.project_id   | string | 当前流水线所属项目ID，同现有 <a href="#">流水线系统预定义参数</a> PROJECT_ID。   |  |
|                  | pipeline.pipeline_id  | string | 当前流水线ID，同现有 <a href="#">流水线系统预定义参数</a> PIPELINE_ID。  |  |
|                  | pipeline.run_number   | string | 流水线执行编号，同现有 <a href="#">流水线系统预定义参数</a> PIPELINE_NUMBER。  |  |
|                  | pipeline.timestamp    | string | 流水线执行时间戳，同现有 <a href="#">流水线系统预定义参数</a> TIMESTAMP，格式为yyyyMMddHHmmss。如：<br>20211222124301。                        |  |
|                  | pipeline.trigger_type | string | 流水线触发类型，同现有 <a href="#">流水线系统预定义参数</a> PIPELINE_TRIGGER_TYPE。  |  |
|                  | pipeline.run_id       | string | 流水线执行ID，同现有 <a href="#">流水线系统预定义参数</a> PIPELINE_RUN_ID。  |  |

| 上下文             | 属性                              | 类型     | 说明   | 示例   |
|-----------------|---------------------------------|--------|--|--|
| sources context | sources                         | object | 流水线运行的源相关的所有信息，此对象包含的属性包括：alias、repo_name、commit_id、commit_id_short、commit_message、repo_url、repo_type、repo_name、ssh_repo_url、tag、merge_id、source_branch和target_branch。 | <ul style="list-style-type: none"> <li>内容示例<br/>如下示例为单代码源的流水线一次手动运行所包含的sources context信息，其中代码源别名为“my_repo”。<br/> <pre>{   "my_repo": {     "commit_id": "dedb73bb9abfdaab7d810f2616bae9d2b6632ecc",     "commit_id_short": "dedb73bb",     "commit_message": "maven0529 更新 pipeline0615.yml",     "repo_url": "https://example.com/clsy00001/maven0529.git",     "repo_type": "codehub",     "repo_name": "maven0529",     "ssh_repo_url": "git@example.com:clsy00001/maven0529.git",     "target_branch": "master"   } }</pre> </li> <li>使用示例<br/>如需在某个任务的输入中获取本次流水线运行的运行分支，可使用如下语法来获取。<br/> <pre>\${{ sources.my_repo.target_branch }}</pre> </li> </ul> |
|                 | sources.<alias>                 | object | 别名为alias对应的流水线源的所有信息，当流水线源定义别名时存在。   |  |
|                 | sources.<repo_name>             | object | 代码仓名称为repo_name的流水线代码源的所有信息，当代码源未定义别名时存在，包含属性同alias。   |  |
|                 | sources.<alias>.commit_id       | string | 流水线执行前最后一次代码提交的CommitID，同现有 <a href="#">流水线系统预定义参数</a> COMMIT_ID。  |  |
|                 | sources.<alias>.commit_id_short | string | 流水线执行前最后一次代码提交的CommitID的前8位，同现有 <a href="#">流水线系统预定义参数</a> COMMIT_ID_SHORT。  |  |
|                 | sources.<alias>.commit_message  | string | 流水线执行前最后一次代码提交的提交信息。   |  |

| 上下文 | 属性                            | 类型     | 说明  | 示例 |
|-----|-------------------------------|--------|---|----|
|     | sources.<alias>.repo_url      | string | 代码仓地址（HTTPS），同现有 <a href="#">流水线系统预定义参数REPO_URL</a> 。 |    |
|     | sources.<alias>.repo_type     | string | 代码仓类型。如：codehub、gitlab、github、gitee、general_git。      |    |
|     | sources.<alias>.repo_name     | string | 代码仓名称。  |    |
|     | sources.<alias>.ssh_repo_url  | string | 代码仓地址（SSH）。   |    |
|     | sources.<alias>.tag           | string | 标签触发时的标签名称。   |    |
|     | sources.<alias>.merge_id      | string | 合并请求触发时的合并请求ID。                                       |    |
|     | sources.<alias>.source_branch | string | 合并请求触发时的源分支名称。  |    |
|     | sources.<alias>.target_branch | string | MR时代表目标分支名称，否则代表运行分支名称。                               |    |

| 上下文          | 属性                                  | 类型     | 说明  | 示例  |
|--------------|-------------------------------------|--------|---|---|
| env context  | name                                | string | 自定义参数名称。  | <ul style="list-style-type: none"> <li>内容示例<br/>如下示例为流水线一次运行包含的env context信息，包含了两个自定义参数。<br/> <pre>{   "var_1": "val1",   "var_2": "val2" }</pre> </li> <li>使用示例<br/>如需要在某个任务的输入中获取本次流水线运行的自定义参数“var_1”的值，可使用如下语法来获取。<br/> <pre>\${ env.var_1 }</pre> </li> </ul>   |
|              | value                               | string | 自定义参数值。   |   |
| jobs context | jobs                                | object | 流水线运行的任务相关的信息，此对象包含的属性包括：job_id、status、outputs、output_name、metrics、metric_name。   | <ul style="list-style-type: none"> <li>内容示例<br/>如下示例为流水线一次运行对应的jobs context信息，包含两个执行成功的任务，ID为“check_job”的任务的输出为两项指标，ID为“demo_job”的任务的输出为两项一般输出。<br/> <pre>{   "check_job": {     "status": "COMPLETED",     "metrics": {       "critical": "0",       "major": "0"     }   },   "demo_job": {     "status": "COMPLETED",     "outputs": {       "output1": "val1",       "output2": "val2"     }   } }</pre> </li> <li>使用示例<br/>如需要在后续任务的输入中获取本次运行的“demo_job”的输出“output1”的值，可使用如下语法来获取。<br/> <pre>\${ jobs.demo_job.outputs.output1 }</pre> </li> </ul> |
|              | jobs.<job_id>                       | object | ID为job_id的任务包含的所有信息。  |   |
|              | jobs.<job_id>.status                | string | 任务的运行结果，取值结果为：INIT（初始化）、QUEUED（排队中）、RUNNING（运行中）、CANCELED（已中止）、COMPLETED（成功）、FAILED（失败）、PAUSED（暂停）、IGNORED（忽略）、SUSPEND（挂起）或UNSELECTED（未选择）。 |   |
|              | jobs.<job_id>.outputs               | object | 任务的运行一般输出，即键值对形式的输出。  |   |
|              | jobs.<job_id>.outputs.<output_name> | string | 任务的运行一般输出的数据名称。   |   |

| 上下文 | 属性                                  | 类型     | 说明                             | 示例 |
|-----|-------------------------------------|--------|--------------------------------|----|
|     | jobs.<job_id>.metrics               | object | 任务的运行指标输出。如：代码检查问题数、测试通过率等指标项。 |    |
|     | jobs.<job_id>.metrics.<metric_name> | string | 任务的运行指标输出的名称。                  |    |

## 相关信息

流水线上下文应用广泛，以下常见场景供参考：

- [配置流水线执行条件表达式](#)。
- [配置通过流水线上下文获取构建任务的产物信息](#)。
- [通过流水线生成标签名并通过上下文传递为代码仓库创建标签](#)。

### 13.1.2 配置流水线执行条件表达式

在流水线阶段任务的执行条件中通过表达式引用流水线上下文，用以限制任务执行。表达式由[流水线上下文](#)、操作符、函数或常量组合而成，表达式以编程的方式访问流水线上下文，实现将流水线中运行的实例、变量、任务等信息在流水线之间互相传递。

**步骤1** [新建流水线](#)。

**步骤2** 新建阶段任务或编辑已有阶段任务。

**步骤3** 单击“任务配置”，在“执行条件”参数项选择“表达式”，即可配置流水线阶段任务执行条件的表达式。

#### 说明

如果是新建阶段任务，需添加插件后再单击“任务配置”。

图 13-1 执行条件表达式



示例：

在执行条件中定义任务在指定代码源的运行分支为“master”时才运行，表达式语法如下：

```
${{ sources.my_repo.target_branch == 'master' }}
```

----结束

## 相关参考

- **操作符**

表达式中可以使用如下操作符：

表 13-3 操作符

| 操作符 | 说明  |
|-----|---|
| .   | 属性引用。如：\${{ pipeline.trigger_type }}，表示获取流水线的触发类型。  |
| !   | 非。如：\${{ !startsWith(sources.my_repo.target_branch, 'release') }}，表示判断流水线代码源的分支是否不以release开头。 |
| ==  | 等于。如：\${{ pipeline.trigger_type == 'Manual' }}，表示判断流水线的触发方式是否为手动触发。                           |
| !=  | 不等于。如：\${{ pipeline.trigger_type != 'Manual' }}，表示判断流水线的触发方式是否不为手动触发。                         |



| 操作符 | 说明  |
|-----|---|
| &&  | 且。如: <code>\${{ pipeline.trigger_type == 'Manual' &amp;&amp; sources.my_repo.target_branch == 'master' }}</code> , 表示判断流水线的触发方式为手动触发并且流水线代码源的分支为master。 |
|     | 或。如: <code>\${{ pipeline.trigger_type == 'Manual'    sources.my_repo.target_branch == 'master' }}</code> , 表示判断流水线的触发方式为手动触发或者流水线代码源的分支为master。         |

- **函数**

表达式中可使用如下内置函数:

表 13-4 内置函数

| 函数         | 说明  |
|------------|---|
| contains   | <ul style="list-style-type: none"> <li>• 格式<br/><code>contains(search, item)</code></li> <li>• 含义<br/>如果“search”包含“item”，则函数返回“true”。                             <ul style="list-style-type: none"> <li>- 如果“search”是一个数组，那么如果“item”是数组中的一个元素，则此函数返回“true”。</li> <li>- 如果“search”是一个字符串，那么如果“item”是“search”的子字符串，则函数返回“true”。</li> </ul> </li> <li>• 示例<br/><code>contains('abc', 'bc')</code>，函数将返回“true”。</li> </ul> |
| startsWith | <ul style="list-style-type: none"> <li>• 格式<br/><code>startsWith(searchString, searchValue)</code></li> <li>• 含义<br/>如果“searchString”以“searchValue”开始，则函数返回“true”。</li> <li>• 示例<br/><code>startsWith('abc', 'ab')</code>，函数将返回“true”。</li> </ul>   |
| endsWith   | <ul style="list-style-type: none"> <li>• 格式<br/><code>endsWith(searchString, searchValue)</code></li> <li>• 含义<br/>如果“searchString”以“searchValue”结束，则函数返回“true”。</li> <li>• 示例<br/><code>endsWith('abc', 'bc')</code>，函数将返回“true”。</li> </ul>   |

| 函数    | 说明   |
|-------|--|
| 对象筛选器 | <p>使用 “*” 语法应用筛选器并选择集合中的匹配项。</p> <p>示例：如下是流水线某次运行的jobs上下文</p> <pre>{   "check_job": {     "status": "COMPLETED",     "metrics": {       "critical": "0",       "major": "0"     }   },   "demo_job": {     "status": "FAILED"   } }</pre> <ul style="list-style-type: none"><li>• 示例1: “jobs.*.status”，表示所有任务的状态，所以返回 “[ 'COMPLETED', 'FAILED' ]”。</li><li>• 示例2: 结合contains函数一起使用，如：<br/>contains(jobs.*.status, 'FAILED')， “jobs.*.status” 中包含 “FAILED”，所以返回 “true”。</li></ul> |

### 13.1.3 配置通过流水线上下文获取构建任务的产物信息

在流水线阶段任务的输入框中通过引用流水线上下文获取相关信息。本示例介绍如何在流水线上使用“Build构建”插件生成构建产物，并在后续阶段的“执行Shell”任务中通过流水线上下文引用，获取构建产物信息。

**步骤1** 新建流水线。

**步骤2** 在“阶段\_1”添加“Build构建”插件，获取任务ID，如[图13-2](#)，并将构建产物标识设置为“demo”，如[图13-3](#)。

图 13-2 获取任务 ID

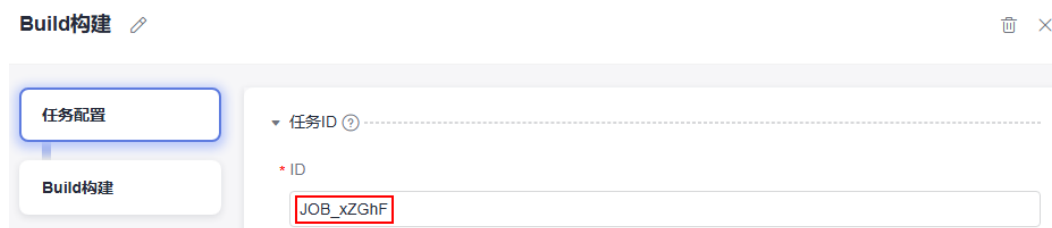


图 13-3 Build 构建插件



**步骤3** 在“阶段\_2”添加“执行Shell”插件，通过构建任务ID，结合流水线上上下文，输入如下命令，即可获取构建产物的相关信息。

```
# 获取构建产物下载地址  
echo ${ jobs.JOB_xZGhF.artifacts.demo.download_url }  
# 获取构建产物所有信息  
echo ${ jobs.JOB_xZGhF.artifacts.demo }
```

图 13-4 执行 Shell



**步骤4** 执行流水线，执行成功后即可在日志中查看打印的产物信息。

图 13-5 查看产物信息

```
[INFO] #####@ [step start] @#####
[INFO] ----- start to use script executor -----
[INFO] ----- start to execute shell pre function [make temp dir] -----
[INFO] ----- shell pre function [make temp dir] current temp dir: [/data/workspace/51e96984e29d48ee9507acbc407b457/accf7caa696647c6a37de95021bbe2d6] -----
[INFO] ----- end to execute shell pre function [make temp dir] successfully -----
[INFO] ----- start to execute custom shell -----
[INFO] https://devrepo.oss-cn-hangzhou.aliyuncs.com/DevRepoServer/v1/files/download?file_id=0be4d509b61742559725a8cef0abbef
[INFO] name:javaMavenDemo-1.0.jar artifact_type:package upload_target:artifact download_url:https://devrepo.oss-cn-hangzhou.aliyuncs.com/DevRepoServer/v1/files/download?file_id=0be4d509b61742559725a8cef0abbef
[INFO] ----- finish to use script executor with status success -----
[INFO] #####@ [step end] @#####
```

----结束

## 13.2 YAML 语法规则

### 13.2.1 流水线触发方式 on

使用on关键字可指定流水线的自动触发方式。可以在on中指定多个代码事件触发流水线的方式，支持通过过滤规则实现仅当满足特定的分支、文件、标签变更时才触发流水线；也可以在on中指定定时任务执行流水线。

#### on.<event\_name>.types

指定触发流水线的代码事件的具体类型，当前仅支持对合并请求事件进行类型过滤：

表 13-5 合并请求事件类型

| 合并请求事件类型    | 说明         |
|-------------|------------|
| opened      | 合并请求被创建。   |
| synchronize | 源分支被更新。    |
| closed      | 合并请求被合入。   |
| reopened    | 合并请求被重新打开。 |

默认情况下，流水线会在合并请求发生opened、synchronize、reopened事件时触发，以下示例表示仅当合并请求被合入时才触发流水线：

```
on:
  pull_request:
    types:
      - closed
```

#### on.<pull\_request>.<branches|branches-ignore>

使用pull\_request事件定义流水线针对特定分支的合并请求运行。

- branches：指定在触发规则中包含某些分支或同时包含和排除某些分支，使用branches进行过滤。
- branches-ignore：指定在触发规则中排除某些分支时，使用branches-ignore进行过滤。

#### 📖 说明

为避免产生歧义，不能在pull\_request事件中同时指定branches和branches-ignore。

branches和branches-ignore支持以Glob模式进行匹配，具体匹配规则如下：

表 13-6 匹配规则

| 匹配符 | 说明                                 | 示例  |
|-----|------------------------------------|---|
| *   | 匹配0到多个任意字符（除“/”外）。                 | dev*: 可匹配dev、develop等，但无法匹配dev/test。  |
| ?   | 匹配任意一个字符。                          | dev?: 可匹配dev1、dev2等，但无法匹配dev12。   |
| **  | 匹配0到多个任意字符。                        | dev**: 可匹配dev、develop、dev/test等。  |
| []  | 匹配括号内指定范围内的一个字符，范围限定在a-z、A-Z、0-9中。 | <ul style="list-style-type: none"> <li>v[a-z].[0-9]*: 可匹配va.1、vb.111等。</li> <li>v[01]: 可匹配v0、v1。</li> </ul> |
| {}  | 匹配括号内的多个字符串。                       | {branch1,branch2}: 可以匹配branch1、branch2。   |
| !   | 在字符串第一位定义，表示匹配除了后续字符串之外的任意字符串。     | !develop: 表示匹配develop之外的所有字符串。  |

## 示例

- **定义包含分支**

```
on:
  pull_request:
    branches:
      - master
      - 'release**'
```

如上定义表示流水线仅在合并请求事件的目标分支满足以下条件之一时运行：

- 分支为master。
- 分支名以release开头，如：release、release-1.0.0、release/1.0.0。

- **定义排除分支**

```
on:
  pull_request:
    branches-ignore:
      - test
      - 'dev**'
```

如上定义表示流水线会在任意分支的合并请求事件触发时运行，除非目标分支满足以下条件之一：

- 分支为test。
- 分支名以dev开头，如dev、develop-1.0.0、develop/1.0.0。

- **同时定义包含和排除分支**

如果需要同时定义包含和排除分支，需在包含分支中定义“!”以排除特定的分支。

```
on:
  pull_request:
    branches:
```

```
- 'release**'  
- '!release/v1**'
```

如上定义表示流水线仅在合并请求的目标分支满足以下全部条件时运行：

- 分支名以release开头，如：release、release-1.0.0、release/1.0.0。
- 分支名不以release/v1/\*\*开头，如：release/v1、release/v1.0、release/v1/1.0。

## on.<push>.<branches|branches-ignore|tags|tags-ignore>

使用push事件定义流水线针对特定分支/tag的代码提交运行。

- branches：指定在触发规则中包含某些分支或同时包含和排除某些分支时，使用branches进行过滤。
- branches-ignore：指定在触发规则中排除某些分支时，使用branches-ignore进行过滤。
- tags：指定在触发规则中包含某些tag或同时包含和排除某些tag时，使用tags进行过滤。
- tags-ignore：指定在触发规则中排除某些tag时，使用tags-ignore进行过滤。

### 📖 说明

- 为避免产生歧义，不能在push事件中同时指定branches和branches-ignore。
- 为避免产生歧义，不能在push事件中同时指定tags和tags-ignore。

### 示例

- **定义包含分支/tag**

```
on:  
  push:  
    branches:  
      - master  
      - 'release**'  
    tags:  
      - v1  
      - 'v2.*'
```

如上定义表示流水线仅在代码提交的分支/tag满足以下条件之一时运行：

- 分支为master。
- 分支名以release开头，如：release、release-1.0.0、release/1.0.0。
- tag为v1。
- tag以v2.开头，如：v2.1、v2.1.1。

- **定义排除分支/tag**

```
on:  
  push:  
    branches-ignore:  
      - test  
      - 'dev**'  
    tags-ignore:  
      - v1  
      - 'v2.*'
```

如上定义表示流水线会在任意代码提交事件时运行，除非分支/tag满足以下条件之一：

- 分支为test。
- 分支名以dev开头，如：dev、develop-1.0.0、develop/1.0.0。

- tag为v1。
- tag以v2.开头，如：v2.1、v2.1.1。
- **同时定义包含和排除分支/tag**  
如果需要同时定义包含和排除分支/tag，需在包含分支/tag中定义“!”以排除特定的分支/tag。

```
on:  
  push:  
    branches:  
      - 'release**'  
      - '!release/v1**'  
    tags:  
      - 'v1**'  
      - '!v1.1'
```

如上定义表示流水线仅在代码提交的分支/tag满足以下所有情况时运行：

- 分支名以release开头，如：release、release-1.0.0、release/1.0.0。
- 分支名不以release/v1/\*\*开头，如：release/v1、release/v1.0、release/v1/1.0。
- tag名以v1开头，如：v1、v1.2。
- tag名不为v1.1。

## on.<push|pull\_request>.<paths|paths-ignore>

使用push和pull\_request事件定义流水线在指定文件发生变更时运行。

- paths：指定在触发规则中包含某些文件或同时包含和排除某些文件时，使用paths进行过滤。
- paths-ignore：指定在触发规则中排除某些文件时，使用paths-ignore进行过滤。

### 📖 说明

为避免产生歧义，不能在push和pull\_request事件中同时指定paths和paths-ignore。

### 示例

- **定义包含文件**

```
on:  
  push:  
    paths:  
      - '**.java'
```

表示流水线在代码提交变更的文件中至少有一个.java文件时运行。

- **定义排除文件**

```
on:  
  push:  
    paths-ignore:  
      - 'docs/**'
```

表示流水线会在任意代码提交事件时运行，除非变更的文件列表全部在docs目录下。

- **同时定义包含和排除文件**

如果需要同时定义包含和排除文件，需在包含文件中定义“!”以排除特定的文件。

```
on:  
  push:  
    paths:  
      - 'src/**'  
      - '!src/docs/**'
```

如上定义表示流水线仅在代码提交的变更文件列表中包含src目录下的文件，但不全是src/docs目录下的文件时运行。

## on.schedule

使用on.schedule通过定义cron表达式使流水线在指定的UTC时间执行。以最后一次保存流水线时YAML内容中的定时任务配置为准，如果需要更新定时触发信息，需重新在页面上编辑YAML内容并保存流水线。定时任务执行时，会使用流水线源的默认分支。

```
on:
  schedule:
    - cron: '0 0 12 * * ?'
    - cron: '0 0 20 * * ?'
```

如上定义表示流水线有两个定时触发任务，分别在UTC时间每天12:00以及20:00触发。

## 13.2.2 流水线参数 env

使用env以键值对的形式定义环境变量，定义后可在流水线的任意任务中引用。

### 示例

```
env:
  version: 1.0.0
```

如上定义可在任意任务中使用“\${version}”或“\${{ env.version }}”的方式进行引用。

推荐使用“\${{ env.version }}”表达式形式，支持搭配多种操作符和函数，功能更丰富，表达式编写请参考[流水线执行条件表达式](#)。

## 13.2.3 流水线任务 jobs

一条流水线由定义在jobs中的多个任务组成。

### jobs.<job\_id>

使用jobs.<job\_id>定义任务的唯一标识，流水线内不能重复，该标识只能由大小写字母、数字、“-”、“\_”组成，最长32位。

```
jobs:
  job1:
    name: first job
  job2:
    name: second job
```

如上定义表示两个唯一标识分别为job1和job2的任务。

### jobs.<job\_id>.name

使用jobs.<job\_id>.name定义任务的名称，在图形化界面中会展示该名称。

```
jobs:
  job1:
    name: first job
  job2:
    name: second job
```

如上定义表示job1和job2任务的名称分别为first job和second job。



## jobs.<job\_id>.needs

使用jobs.<job\_id>.needs可定义任务在指定的任务结束后才开始运行。

```
jobs:
  job1:
    name: first job
  job2:
    needs: [ job1 ]
    name: second job
```

如上定义表示job2会在job1运行成功后开始执行。

## jobs.<job\_id>.if

使用jobs.<job\_id>.if可定义任务的运行条件，表达式编写请参考[流水线执行条件表达式](#)。

```
jobs:
  job1:
    name: first job
  job2:
    needs: [ job1 ]
    if: ${{ always() }}
    name: second job
```

如上定义表示job2会在job1运行结束后开始执行，不论job1的最终状态如何。

## jobs.<job\_id>.steps

每个任务由定义在steps中的多个步骤组成，每个步骤可运行一个指定的插件。

## jobs.<job\_id>.steps<\*>.name

使用jobs.<job\_id>.steps<\*>.name定义步骤的名称，在图形化界面中会展示该名称。

## jobs.<job\_id>.steps<\*>.uses

使用jobs.<job\_id>.steps<\*>.uses可指定步骤运行的插件。

```
jobs:
  demo_job:
    name: simple demo job
    steps:
      - name: simple custom step
        uses: custom_plugin@1.0.0
```

如上定义表示一个名称为custom\_plugin，版本为1.0.0的插件。

## 流水线官方插件 YAML 语法

- **Build构建**

调用编译构建能力进行构建。

```
uses: CodeArtsBuild
with:
  jobId: 878b4d13cb284d9e8f33f988a902f57c
  artifactIdentifier: my_pkg
  customParam: value
```

- jobId: 编译构建任务id。获取方法：在构建任务详情页，复制浏览器URL末尾的32位数字、字母组合的字符串，即为构建任务ID。

- artifactIdentifier: 构建产物标识。
- customParam: 在构建任务定义的支持运行时设置的参数值，可能有0到多个。

- **TestPlan接口测试**

调用CodeArts TestPlan能力进行测试。

```
uses: CodeArtsTestPlan
with:
  jobId: vb180000vnrgoeib
  environmentModel: 1
  environmentId: 7c2eff2377584811b7981674900158e8
```

- jobId: 接口测试任务的id。
- environmentModel: 参数来源（0表示使用全新执行参数，1表示使用选中环境的全局参数）。
- environmentId: environmentModel为1时的环境id。

- **Deploy部署**

调用CodeArts Deploy能力进行部署。

```
uses: CodeArtsDeploy
with:
  jobId: 9c5a5cda6ffa4ab583380f5a014b2b31
  customParam: value
```

- jobId: 部署任务的id。
- customParam: 在部署任务处定义的支持运行时设置的参数值，可能有0到多个。

- **Check代码检查**

调用CodeArts Check能力进行代码检查。

```
uses: CodeArtsCheck
with:
  jobId: 43885d46e13d4bf583d3a648e9b39d1e
  checkMode: full|push_inc_full|push_multi_inc_full
```

- jobId: 代码检查任务的id。
- checkMode: 检查模式。
  - full: 表示对本次commit所在分支的全量代码进行检查。
  - push\_inc\_full: 表示对本次commit的变更文件内容进行检查。
  - push\_multi\_inc\_full: 表示对上一次检查成功的commit到本次commit之间的变更文件内容进行检查。

- **创建仓库标签**

为代码仓创建标签并推送。

```
uses: CreateTag
with:
  tagName: v1
```

tagName为标签名称。

- **子流水线**

配置并调用项目下其他流水线任务。

```
uses: SubPipeline
with:
  pipelineId: 80ea2d9ffba94c20b9a0a0be47d3a0d8
  branch: master
```

- pipelineId: 调用的流水线的id。
- branch: 子流水线运行使用的分支。
  - branch参数可以不定义，表示使用子流水线默认分支运行。
  - 若定义branch参数则表示以指定的分支运行，可以引用参数或上下文，如：使用父流水线源的运行分支，假设代码源别名为“my\_repo”，则引用形式为`${sources.my_repo.target_branch}`。

### • Jenkins任务

调用Jenkins任务以扩展并实现自定义动作。

```
uses: Jenkins
with:
  endpoint: eac965b206e74e2b898a24a4375b6df6
  jobName: job
  params: { \"key\": \"value\" }
  async: true|false
  description: description
```

- endpoint: jenkins扩展点的id。
- jobName: jenkins的job名称。
- params: 启动任务传递的参数（json格式）。
- async: 是否异步执行。
- description: 执行说明。

### • 挂起流水线

配置以挂起当前流水线。

```
uses: SuspendPipeline
```

### • 延时执行

让流水线暂停运行一段时间或暂停至指定时间点，用户可以手动继续、终止流水线运行或进行最多3次的额外延时。

```
uses: Delay
with:
  timerType: delay|scheduled
  delayTime: 300
  scheduledTime: '00:00'
  timeZone: China Standard Time
```

- timerType: 延时类型。delay表示延时一段时间；scheduled表示延时至指定时间。
- delayTime: timerType为delay时的延时时间（单位为秒）。
- scheduledTime: timerType为scheduled时的延时时间点。
- timeZone: 对应时区，可选值见下表。

表 13-7 时区

| 可选值                        | 时区        |
|----------------------------|-----------|
| GMT Standard Time          | GMT       |
| South Africa Standard Time | GMT+02:00 |
| SE Asia Standard Time      | GMT+07:00 |

| 可选值                            | 时区        |
|--------------------------------|-----------|
| Singapore Standard Time        | GMT+08:00 |
| China Standard Time            | GMT+08:00 |
| Pacific SA Standard Time       | GMT-04:00 |
| E. South America Standard Time | GMT-03:00 |
| Central Standard Time (Mexico) | GMT-06:00 |
| Egypt Standard Time            | GMT+02:00 |
| Saudi Arabia Standard Time     | GMT+03:00 |

- **人工审核**

以单签或会签的方式配置人工审核任务。

```
uses: Checkpoint
with:
  mode: members|roles
  approvers: 05d8ca972f114765a8984795a8aa4d41
  roles: PROJECT_MANAGER
  checkStrategy: all|any
  timeout: 300
  timeoutStrategy: reject|pass
  comment: comment
```

- mode: 审核模式。members表示按成员审核，roles表示按角色审核。
- approvers: 审核模式为members时的审核人员用户id，多个用户以“,”分隔。
- role: 审核模式为roles时的角色，可选值见[表13-8](#)，多个角色以“,”分隔。
- checkStrategy: 审核模式为members时的审核策略。all表示需所有人员审核才能通过；any表示任一人员审核即可通过。
- timeout: 审核超时时长（单位为秒）。
- timeoutStrategy: 审核超时的策略。reject表示中止；pass表示通过。
- comment: 审核描述。

**表 13-8** 审核角色

| 审核角色  | YAML标识          |
|-------|-----------------|
| 项目创建者 | PROJECT_CREATOR |
| 项目经理  | PROJECT_MANAGER |
| 开发人员  | DEVELOPER       |
| 测试经理  | TESTING_MANAGER |
| 测试人员  | TESTER          |

| 审核角色      | YAML标识            |
|-----------|-------------------|
| 参与者       | PARTICIPANT       |
| 浏览者       | VIEWER            |
| 运维经理      | OPERATION_MANAGER |
| 产品经理      | PRODUCT_MANAGER   |
| 系统工程师     | SYSTEM_ENGINEER   |
| Committer | COMMITTER         |

# 14 发布管理(CodeArts Release)使用指南

## 14.1 发布管理概述

发布管理（CodeArts Release）是CodeArts提供的版本配套、自动化部署上线的E2E解决方案，支撑产品的版本级持续交付。通过发布管理可以保证发布团队在保持现有生产环境完整性的同时，高效交付业务所需的应用程序和升级。使用Codearts产品时，虚拟机部署一般使用Codearts Depoly，若为容器化部署则推荐使用Codearts Release。

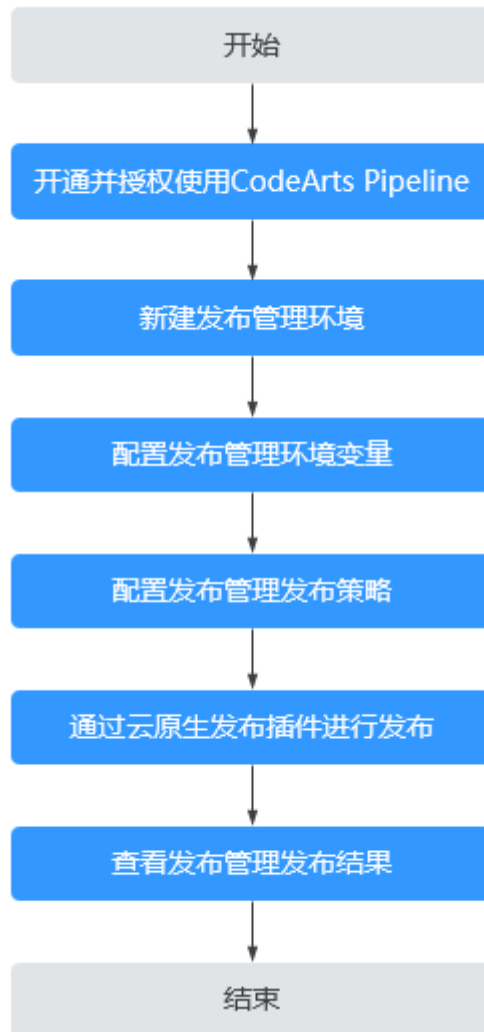
Codearts Release包括以下特性：

- 提供面向解决方案版本基线管理能力，支持微服务、模块级、产品级多维度版本编排，支持多云版本配套。
- 云原生微服务发布管理能力，支持微服务灰度编排发布能力，支持蓝绿、金丝雀等灰度发布策略，基于UCS实现跨云编排

### 发布管理服务基本操作流程

发布管理的基本操作流程包括：[开通并授权使用CodeArts Pipeline](#)、新建发布管理环境、配置发布管理环境变量、配置发布管理发布策略、通过云原生发布插件进行发布、查看发布管理发布结果。


图 14-1 发布管理服务基本操作流程



## 14.2 新建发布管理环境

### 新建发布管理环境

**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 > 软件开发生产线”。

**步骤3** 单击“立即使用”，进入CodeArts首页。

**步骤4** 单击项目名称，进入目标项目。

**步骤5** 单击“持续交付 > 发布管理”，进入发布管理环境列表页面

**步骤6** 单击“新建环境”，进入“新建环境”页面，填写基本信息，相关参数说明如表14-1所示。

表 14-1 参数说明

| 参数项           | 说明  |
|---------------|---|
| 归属项目          | 环境所属项目，不可修改。  |
| 环境名称          | 发布管理环境唯一标识。支持中文、英文字母、数字、“-”、“_”，长度不超过128个字符。  |
| 资源类型          | 承载服务的资源类型，当前支持华为云CCE集群、UCS集群和自建Kubernetes集群，不同类型的资源支持的部署插件不同。 <ul style="list-style-type: none"><li>● CCE：华为云对Kubernetes封装的集群类型，使用华为云CCE资源时选择该类型，<a href="#">了解更多</a>。</li><li>● UCS：华为云对Kubernetes封装的多云集群类型，当用户有多云部署需求时，可选择该类型，<a href="#">了解更多</a>。</li><li>● K8S：业界标准的原生集群类型，当用户使用自建集群或者第三方集群时，可选择该类型。</li></ul> |
| 发布用户          | 可选择当前用户和其他用户。 <ul style="list-style-type: none"><li>● 当前用户：基于当前用户下的集群创建环境。</li><li>● 其他用户：基于其他用户下的集群创建环境。选择其他用户时，需要通过服务扩展点获取该用户的集群权限，具体操作请参考<a href="#">新建CodeArts服务扩展点</a>。</li></ul>  |
| 区域            | 资源类型为CCE时，需设置此参数。<br>选择部署环境的区域。   |
| 集群            | 资源类型为CCE时，需设置此参数。<br>选择在CCE云容器引擎中 <a href="#">已购买的Kubernetes集群</a> 。  |
| 关联类型          | 资源类型为UCS时，需设置此参数。<br>关联的UCS资源粒度，当前仅支持 <a href="#">容器舰队</a> 。  |
| 容器舰队          | 资源类型为UCS时，需设置此参数。<br>选择 <a href="#">已创建的舰队</a> 。  |
| Kubernetes扩展点 | 资源类型为K8S时，需设置此参数。<br>支持以扩展点的方式关联集群资源，以Kubernates扩展点配置的凭证信息访问集群资源，具体操作请参考 <a href="#">新建CodeArts服务扩展点</a> 。  |
| 环境级别          | 环境类型，内置了开发环境、测试环境、预发环境和生产环境四种类型，主要用于区分环境的操作权限，对应环境权限请参考 <a href="#">项目级权限</a>   |
| 描述            | 填写环境的描述信息。不超过200个字符。  |

**步骤7** 填写完所有参数，单击“确定”，完成环境创建并进入“环境信息”页，页面展示环境基本信息。



表 14-2 环境信息

| 参数项   | 说明                    |
|-------|-----------------------|
| 资源类型  | 环境关联的资源类型。            |
| 发布用户  | 当前用户。                 |
| 服务扩展点 | CCE资源的服务扩展点。          |
| 集群区域  | 在CCE云容器引擎中申请的K8S集群区域。 |
| 集群ID  | 在CCE云容器引擎中申请的K8S集群ID。 |
| 变量版本  | 当前环境的环境变量版本号。         |
| 标签    | 环境类型。                 |
| 描述    | 描述信息。                 |

- 单击右上角“编辑”，可修改环境信息。
- 切换页签，可[配置环境变量](#)、[配置发布策略](#)和[查看部署结果](#)。

----结束

## 14.3 配置发布管理环境变量

用户可以在新建/编辑发布策略时使用“\${变量名}”引用环境变量，也可在发布策略关联的YAML配置文件中使“{{变量名}}”引用环境变量。环境变量包括以下两种。

- 自定义变量：根据需要自定义添加，目前仅支持字符串类型的变量。
- 默认变量：系统默认提供的参数，无需定义，不能删除，不能修改。

表 14-3 发布管理环境默认变量

| 参数信息       | 说明                                  |
|------------|-------------------------------------|
| ARTIFACT   | 产物地址。在部署YAML中，使用{{ARTIFACT}}引用构建产物。 |
| TIMESTAMP  | 插件执行时间戳。如：20230401095436。           |
| PROJECT_ID | 当前环境所属项目Id。                         |

### 配置发布管理环境自定义变量

**步骤1** 进入发布管理环境列表页面。


**步骤2** 单击环境名称，进入“环境信息”页面。

**步骤3** 单击“环境变量”页签，切换至“环境变量 > 自定义变量”页面。

**步骤4** 单击“编辑变量”，可新增一条变量，填写参数信息。

表 14-4 自定义变量参数说明


| 参数项  | 说明  |
|------|---|
| 变量名  | 变量名称。支持英文字母、数字、“-”、“_”、“.”，长度不超过128个字符。     |
| 类型   | 仅支持字符串类型。                                   |
| 变量值  | 环境变量当前值（新增变量时为空），长度不超过512个字符。               |
| 本次变更 | 环境变量更新后的值。                                  |
| 变量描述 | 变量说明，长度不超过128个字符。                           |
| 私密变量 | 参数为私密变量时，系统会将输入的参数做加密存储，使用时进行解密，且在运行日志里不可见。 |

- 单击操作列 ，可删除对应变量的。
- 单击 **+ 添加**，可继续添加变量。

**步骤5** 操作完成后，单击“保存”，弹出“保存修改”对话框。

**步骤6** 确认信息无误，填写修订日志，单击“确定”，完成添加。

切换至单击“历史版本”页签，可查看变量历史版本列表。

- 单击版本名称，可查看指定版本的自定义变量内容。
- 单击操作列 ，可将当前版本和指定版本进行对比。

----结束

## 发布管理环境变量使用示例

用户可以在以下两种场景中使用环境变量。

- **配置发布管理发布策略**时，在所编排任务的YAML路径参数中，可以使用“\${变量名}”引用环境变量，如：“滚动升级”任务中的“工作负载YAML路径”参数。


图 14-2 引用环境变量

### 滚动升级

\* 部署方式 

镜像创建  YAML创建

\* 仓库类型

Repo 

\* 代码仓

请选择 

\* 分支

请选择 

\* 工作负载YAML路径 

`#{deployment.yaml}`

- 在发布策略关联的YAML配置文件中，使用“`#{变量名}`”引用环境变量。

图 14-3 引用环境变量

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: rolling-test
5    labels:
6      run: rolling-test
7    namespace: default
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       run: rolling-test
13   template:
14     metadata:
15       labels:
16         run: rolling-test
17     spec:
18       containers:
19         - name: main
20           image: {{ARTIFACT}}
21           ports:
22             - containerPort: 8080
23           env:
24             - name: TIMESTAMP
25               value: {{TIMESTAMP}}
26             - name: PROJECT_ID
27               value: {{PROJECT_ID}}
28             - name: COMPONENT_ID
29               value: {{COMPONENT_ID}}
30           resources:
31             limits:
32               cpu: 250m
33               memory: 512Mi
34             requests:
35               cpu: 250m
36               memory: 512Mi
```

## 14.4 配置发布管理发布策略

### 新建发布管理自定义策略

发布管理环境预置了滚动升级模板、灰度升级模板和空白yaml模板，用户可以基于任一模板，添加原子插件，编辑发布策略。

**步骤1** 进入发布管理环境列表页面。

**步骤2** 单击环境名称，进入“环境信息”页面。

**步骤3** 单击“发布策略”页签，切换至“发布策略”页面。

**步骤4** 单击“自定义策略”旁的 $+$ ，弹出“新建策略”对话框，选择策略模板，单击“确定”。

**步骤5** 在模板的基础上自定义编排插件，并配置相关信息。

基本信息
取消
保存
保存并应用

\* 策略名称

策略描述

编排插件

**灰度起负载**

\* 部署方式 ?

镜像升级     YAML部署

\* 命名空间

\* 服务 ?

自定义灰度版本号

表 14-5 新建策略

| 参数项  | 说明   |
|------|--|
| 策略名称 | 策略的名称。支持中文、英文字母、数字、“-”、“_”，不超过128个字符。  |
| 策略描述 | 长度不超过200个字符。   |
| 编排插件 | <p>选择策略模板后会自动生成相应的插件，可根据需要自定义编排并配置插件信息，插件参数配置请参考<a href="#">配置发布管理原子插件</a>。</p> <ul style="list-style-type: none"> <li>单击 ，可添加插件。</li> <li>单击 ，可复制当前插件。</li> <li>单击 ，可删除当前插件。</li> </ul> |

**步骤6** 配置完成后，单击“保存”。

**步骤7** 在左侧目录找到刚创建的策略，单击策略所在行 ，即可应用该策略，应用后策略会变为“使用中”的状态。

----结束

## 配置发布管理原子插件

发布管理提供**滚动升级插件**、**灰度起负载插件**、**灰度引流插件**、**灰度下线插件**和**人工卡点插件**五种插件，用于滚动升级和灰度升级，以下分别介绍各插件配置方法。

- **滚动升级插件**

滚动升级包括两种部署方式：镜像创建和YAML创建。

- 镜像创建：替换工作负载中容器的镜像。

滚动升级

\* 部署方式 

镜像创建  YAML创建

\* 命名空间

请选择 

\* 工作负载

请选择 

\* 容器

请选择 

表 14-6 参数说明

| 参数项  | 说明             |
|------|----------------|
| 命名空间 | 待升级的服务所在的命名空间。 |
| 工作负载 | 命名空间下的工作负载。    |
| 容器   | 工作负载中需要升级的容器。  |

- YAML创建：使用yaml文件升级或上线工作负载。

**滚动升级**

\* 部署方式 

镜像创建  YAML创建

\* 仓库类型

Repo 

\* 代码仓

请选择 

\* 分支

请选择 

\* 工作负载YAML路径 

deployment.yaml

表 14-7 参数说明

| 参数项        | 说明   |
|------------|--|
| 仓库类型       | 代码仓库的类型，当前仅支持Repo。   |
| 代码仓        | 当前项目下的代码仓。   |
| 分支         | 代码仓库分支。  |
| 工作负载YAML路径 | 升级的工作负载YAML路径，填写YAML文件的相对路径。 <ul style="list-style-type: none"> <li>• 当前目录为代码分支的根目录。</li> <li>• 仅支持单个YAML文件。</li> <li>• YAML路径可以使用“\${变量名}”引用环境变量，YAML文件中可以使用“{{变量名}}”引用环境变量。</li> </ul> |

- **灰度起负载插件**  
灰度起负载包括两种部署方式：镜像升级和YAML部署。
  - 镜像创建：替换工作负载中容器的镜像，创建与线上配置完全一致的工作负载，仅更新构建产物（镜像包）。

灰度起负载

\* 部署方式 ?

镜像创建  YAML创建

\* 命名空间

请选择

\* 服务 ?

请选择

自定义灰度版本号



\* 灰度版本号 ?

请输入

表 14-8 参数说明

| 参数项    | 说明  |
|--------|---|
| 命名空间   | 待升级的服务所在的命名空间。  |
| 服务     | 集群命名空间下的Service，需要确保该Service有且仅有一个工作负载与其关联。   |
| 自定义版本号 | 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。   |
| 灰度版本号  | 灰度版本号将作为新旧负载的引流标识，用于区分正式版本和灰度版本，请按需配置，支持使用\${ENV}引用环境变量。如：\${TIMESTAMP}，表示引用系统时间戳环境变量作为灰度版本号。<br>支持大小写英文字母、数字、“-”、“_”、“.”，长度不超过62个字符。 |

- YAML创建：使用yaml文件上线或升级工作负载。



灰度起负载

\* 部署方式 ?

镜像创建  YAML创建

\* 命名空间

请选择

\* 服务 ?

请选择

\* 仓库类型

Repo

\* 代码仓

请选择

\* 分支

请选择

\* 工作负载YAML路径 ?

deployment.yaml


表 14-9 参数说明

| 参数项            | 说明   |
|----------------|--|
| 命名空间           | 待升级的服务所在的命名空间。   |
| 服务             | 集群命名空间下的服务，需要确保该服务有且仅有一个工作负载与其关联。  |
| 仓库类型           | 代码仓库的类型，当前仅支持Repo。   |
| 代码仓            | 当前项目下的代码仓。   |
| 分支             | 代码仓库分支。  |
| 工作负责<br>YAML路径 | 填写YAML文件的相对路径。 <ul style="list-style-type: none"> <li>• 当前目录为代码分支的根目录。</li> <li>• 仅支持单个YAML文件。</li> <li>• YAML路径可以使用“\${变量名}”引用环境变量，YAML文件中可以使用“{{变量名}}”引用环境变量。</li> </ul> |

- 灰度引流插件

#### 灰度引流

##### \* 引流类型

Service蓝绿引流 

选择引流类型，包括：Service蓝绿引流和ASM灰度引流。

- Service蓝绿引流：基于集群Service将所有流量切换至新负载（灰度负载）。
- ASM灰度引流：基于ASM的VirtualService和DestinationRule配置来控制访问流量，可以基于流量比例、请求Header等进行灰度引流，要求集群内安装ASM（应用服务网格）。

- **灰度下线插件**

灰度下线插件会自动下线服务关联的旧工作负载，无需配置。

- **人工卡点插件**

可以在卡点确认时进行批准或驳回部署策略，运行至人工卡点时流水线将暂停，若审批通过则继续执行，若驳回则终止执行。

#### 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

00  小时 00  分钟

##### 卡点说明

请输入

表 14-10 参数说明

| 参数项  | 说明   |
|------|--|
| 超时处理 | <p>超过等待审批时间后流水线的处理方案。</p> <ul style="list-style-type: none"> <li>● 卡点失败，发布流终止执行：运行至人工卡点时流水线将暂停，若超时不处理则终止执行流水线。</li> <li>● 忽略结果，发布流继续执行：运行至人工卡点时流水线将暂停，若超时不处理则继续执行流水线。</li> </ul> |
| 卡点时长 | 等待审批的时间，时间范围为1分钟至12小时。   |
| 卡点说明 | 卡点内容说明。长度不超过200个字符。  |

## 14.5 通过云原生发布插件进行发布

发布管理环境依赖流水线云原生发布插件进行发布，云原生发布插件主要作用是触发对应环境的发布策略，依照发布策略完成环境的发布。

## 通过云原生发布插件进行发布

**步骤1** 配置流水线。

**步骤2** 在流水线阶段新建任务，添加“云原生发布”插件，参数说明如表14-11。

云原生发布提供环境级别的服务发布能力，支持在CCE集群对环境进行自定义编排发布（如：滚动发布和灰度发布）。

图 14-4 配置云原生发布插件

云原生发布 ⓘ 温馨提示

云原生发布提供了环境级别的服务发布能力，支持对环境进行自定义编排发布（例如：滚动... 展开

\* 名称

云原生发布

\* 环境级别

请选择

\* 环境

请选择

\* 产物地址

请输入

表 14-11 参数说明

| 参数项  | 说明   |
|------|--|
| 名称   | 插件的名称。支持输入中文、大小写英文字母、数字、“-”、“_”、“.”、“/”、“(”、“)”、“(”、“)”、空格，长度不超过128个字符。  |
| 环境级别 | 任务发布的环境类别，对发布管理的环境类型，目前支持开发环境、测试环境、预发环境和生产环境。  |
| 环境   | 需要发布的环境，详见 <a href="#">新建发布管理环境</a> 。  |
| 产物地址 | 部署发布时使用的镜像地址，如：swr.example.com/demo/springboot-helloworld:v1.1，可以通过“\${}”引用流水线参数，如：swr.example.com/demo/springboot-helloworld:\${version}。<br><b>说明</b><br>推荐使用SWR容器镜像服务，可通过编译构建 <a href="#">制作镜像并推送到SWR仓库</a> 。 |

**步骤3** 配置完成后，运行流水线，可实时查看运行进展。

**步骤4** 单击云原生发布插件任务，可查看任务运行情况，包括任务日志和任务结果。

图 14-5 查看云原生发布插件运行结果



- 任务日志：查看实时日志信息和运行状态。
- 任务结果：查看任务基本信息，包括工单名称、工单ID、触发人等。  
单击工单ID或“查看详情”可进入发布管理工单详情页面，工单详情页介绍请参见[查看发布管理发布结果](#)。

----结束

## 14.6 查看发布管理发布结果

- 步骤1** 进入发布管理环境列表页面。
- 步骤2** 单击环境名称，进入“环境信息”页面。
- 步骤3** 单击“部署历史”页签，切换至“部署历史”页面，页面展示该环境的所有工单历史。
- 步骤4** 单击工单名称，可以进入工单的详情页面，详情页面展示本次发布的[发布流信息](#)、[基本信息](#)以及各[原子插件发布单信息](#)。

- **发布流信息**

图 14-6 发布流信息



- 展示工单执行结果、工单类型、触发人、流水线、发布策略模板等信息，单击对应原子插件可展示该插件的发布单信息。
- 取消：在发布过程中，单击“取消”可手动取消本次发布。
- 重试：在发布过程中，如果出现失败或取消运行，单击“重试”可重试发布流。
- 回滚：单击“回滚”弹出提示确认框，确认后将取消本次发布过程，将服务状态恢复至发布前。

### 📖 说明

支持在发布流的任意时刻回滚，可用于当前部署版本不符合预期，需要快速将环境恢复到上一次可用状态的场景。

- **基本信息**  
展示环境名称、策略、服务扩展点、变量版本、镜像、开始时间、结束时间。

● 原子插件发布单信息


展示各原子插件的发布详情，单击  刷新 可以手动刷新发布单详情。

图 14-7 原子插件的发布详情

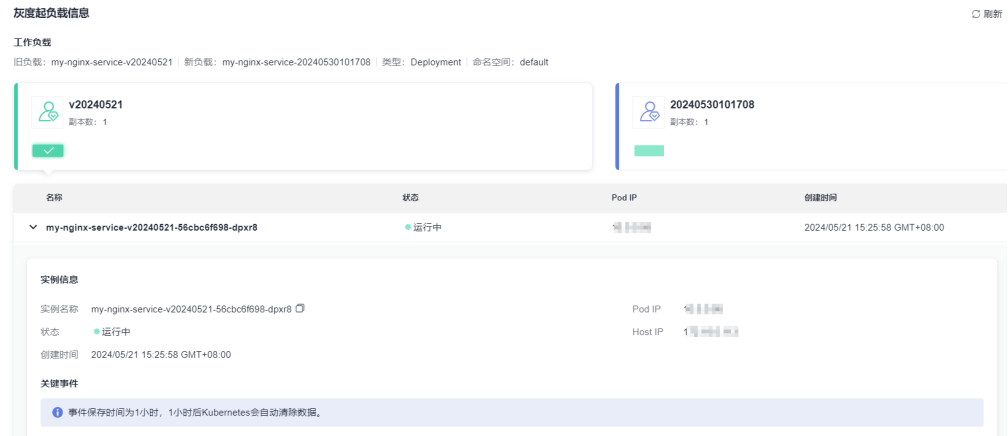


表 14-12 原子插件发布单信息

| 插件类型  | 发布单信息  |
|-------|--|
| 滚动升级  | <p>发布单展示升级的工作负载信息、工作负载对应的实例信息以及关键事件信息。</p> <ul style="list-style-type: none"> <li>- 工作负载信息<br/>负载名称、类型、命名空间及创建时间。</li> <li>- 实例信息<br/>实例名称、实例运行状态、Pod IP、Host IP（Pod所在节点的IP）、实例创建的时间。</li> <li>- 关键事件<br/>查看Pod的关键事件信息，可用于辅助Pod问题定位，包括Kubernetes组件名、事件类型、K8s事件、首次发生时间、最近发生时间。</li> </ul>                           |
| 灰度起负载 | <p>发布单展示升级的工作负载信息、工作负载对应的实例信息以及关键事件信息，单击卡片可以切换新旧负载，查看对应负载信息。</p> <ul style="list-style-type: none"> <li>- 工作负载信息<br/>旧负载名称、新负载名称、负载类型及命名空间。</li> <li>- 实例信息<br/>实例名称、实例运行状态、Pod IP、Host IP（Pod所在节点的IP）、实例创建的时间。</li> <li>- 关键事件<br/>查看Pod的关键事件信息，可用于辅助Pod问题定位，包括Kubernetes组件名、事件类型、K8s事件、首次发生时间、最近发生时间。</li> </ul> |
| 灰度引流  | 发布单展示服务名称、旧版本号、新版本号和命名空间。  |
| 灰度下线  | 发布单展示负载名称、负载类型和命名空间。   |

| 插件类型 | 发布单信息                        |
|------|------------------------------|
| 人工卡点 | 发布单展示卡点时长、操作时间、卡点说明、卡点状态等信息。 |

----结束