

代码检查

用户指南

文档版本

01

发布日期

2026-09-30



版权所有 © 华为云计算技术有限公司 2026。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

目录

1 流水线(CodeArts Pipeline)使用流程.....	1
2 开通并授权使用 CodeArts Pipeline.....	2
2.1 开通 CodeArts Pipeline.....	2
2.2 配置 CodeArts Pipeline 权限.....	2
3 访问 CodeArts Pipeline 首页.....	13
4 新建流水线.....	14
4.1 新建流水线.....	14
4.2 新建流水线（图形化）.....	14
4.3 新建流水线（YAML 化）.....	22
5 新建流水线（新版）.....	26
5.1 新建流水线.....	26
5.2 新建流水线（内置模式）.....	26
5.3 新建流水线（远端模式）.....	30
6 配置流水线.....	33
6.1 编排流水线阶段.....	33
6.2 编排流水线任务.....	41
6.3 配置流水线插件.....	51
6.3.1 Build 构建.....	51
6.3.2 Check 代码检查.....	52
6.3.3 云原生发布.....	53
6.3.4 Deploy 部署.....	54
6.3.5 TestPlan 接口测试.....	54
6.3.6 子流水线.....	55
6.3.7 延时执行.....	56
6.3.8 人工审核.....	57
6.3.9 创建仓库 Tag.....	59
6.3.10 Jenkins 任务.....	59
6.3.11 挂起流水线.....	60
6.3.12 执行镜像命令.....	61
6.3.13 第三方接口调用 v2.....	62
6.3.14 Repo 代码仓分支合入.....	63

6.4 配置流水线参数.....	64
6.5 配置流水线触发设置.....	72
6.6 配置流水线权限.....	81
6.7 配置流水线通知订阅.....	82
7 配置流水线（YAML 化）.....	88
7.1 配置基本信息.....	88
7.2 配置流水线源.....	89
7.3 流水线任务编排.....	92
7.4 配置流水线参数.....	103
7.5 配置事件触发.....	105
7.6 配置其他触发.....	107
7.7 配置执行方案.....	108
7.8 配置定时任务.....	109
8 分组管理流水线.....	111
9 执行流水线.....	115
10 查看流水线.....	117
11 查看统计看板.....	121
12 流水线预警设置.....	124
13 配置微服务变更流水线.....	126
14 管理流水线扩展插件.....	135
14.1 流水线插件概述.....	135
14.2 流水线官方插件.....	136
14.3 图形化新建流水线自定义插件.....	138
14.4 代码化新建流水线自定义插件.....	147
14.5 执行镜像命令.....	159
14.6 添加创建仓库 Tag 插件.....	160
14.7 添加第三方接口调用 v2 插件.....	161
15 管理 CodeArts 服务扩展点.....	163
16 查看审计日志.....	170
17 参考.....	171
17.1 流水线上下文.....	171
17.1.1 流水线上下文.....	171
17.1.2 配置流水线执行条件表达式.....	177
17.1.3 配置通过流水线上下文获取构建任务的产物信息.....	180
17.2 YAML 语法规范.....	182
17.2.1 流水线触发方式 on.....	182
17.2.2 流水线参数 env.....	186
17.2.3 流水线任务 jobs.....	186

18 发布管理(CodeArts Release)使用指南..... 192

18.1 发布管理概述..... 192

18.2 新建发布管理环境..... 192

18.3 配置发布管理环境变量..... 194

18.4 配置发布管理发布策略..... 197

18.5 配置发布管理环境的发布..... 204

18.6 查看发布管理发布结果..... 206

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作。

在[华为云码道](#)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](#)。

表 1-1 CodeArts Pipeline 操作流程说明

流程	说明
开通并授权使用CodeArts Pipeline	为您介绍如何 开通流水线服务 和 授权流水线服务 。
新建流水线任务	为您介绍 新建流水线（图形化） 和 新建流水线（YAML化） 的操作指导，以及如何配置流水线任务的 参数 、 触发设置 、 单任务角色权限 和 事件通知 。
执行流水线任务	流水线任务可通过手动触发或者定时任务触发执行，本节为您介绍在CodeArts Build服务页面 执行流水线任务 。
查看流水线任务	为您介绍如何 查看流水线任务信息以及流水线任务的执行结果 。

2 开通并授权使用 CodeArts Pipeline

2.1 开通 CodeArts Pipeline

前提条件

已[注册华为云账号](#)。

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

2.2 配置 CodeArts Pipeline 权限

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
租户级权限	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。

层级	模块	描述
项目级 权限	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，在项目中进行配置，生效范围为整个项目下的全部资源。
资源级 权限	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。
 - a. 使用租户账号或其他有权限的账号登录CodeArts。
 - b. 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。
 - c. 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。 ● 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。 ● 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。

策略名称	描述
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。 - 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。 - 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。
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。 - 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。 - 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。
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。 - 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。 - 普通用户可以创建模板、查看当前账号下所有子用户的模板，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。

- 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。
- 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。

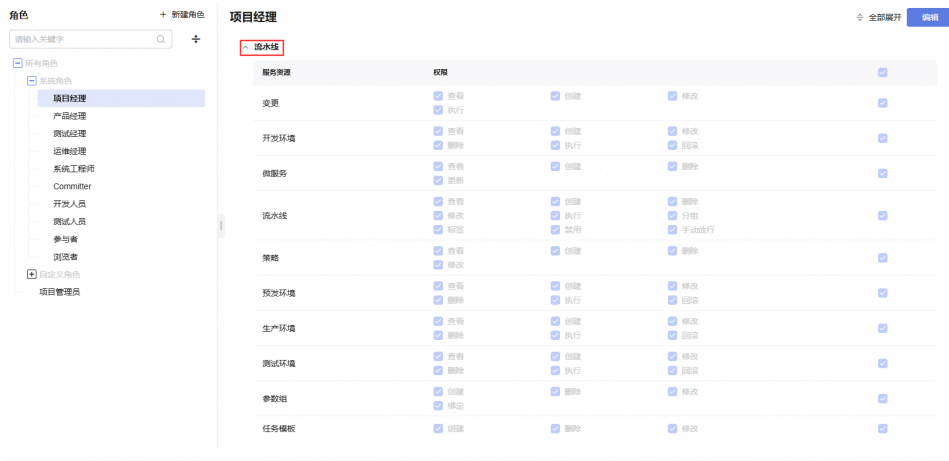
- **项目级权限**

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。

- [登录华为云控制台页面](#)。
- 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”。
- 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 > 权限管理”，进入“权限管理”页面。
- 流水线相关资源集中在流水线服务下，包括变更、开发环境、微服务、流水线、策略（项目级）、预发环境、生产环境、测试环境、参数组和任务模板。

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	标签	禁用	手动放行
项目管理员	√	√	√	√	√	√	√	√	√
项目经理	√	√	√	√	√	√	√	√	√
开发人员	√	√	√	×	×	×	×	×	×
测试经理	√	×	×	×	×	×	×	×	×
测试人员	√	×	×	×	×	×	×	×	×
参与者	√	×	×	×	×	×	×	×	×
浏览者	√	×	×	×	×	×	×	×	×

角色	查看	创建	执行	修改	删除	分组	标签	禁用	手动放行
运维经理	√	×	×	×	×	×	×	×	×
产品经理	√	×	×	×	×	×	×	×	×
系统工程师	√	√	√	√	√	√	√	√	×
Committer	√	√	√	×	×	×	×	×	×
CI/CD 工程师	√	√	√	√	×	×	×	×	×

- 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或人员权限为止。
- 流水线创建者默认拥有其创建的流水线的全部权限。
- 配置手动放行权限后，执行流水线时，若某一阶段任务执行失败，单击失败任务卡片上的，可跳过此阶段，继续执行下一阶段。
- 子流水线下执行手动放行时，不会影响父流水线下子流水线任务的状态。若父流水线需执行手动放行，在父流水线下所配置的子流水线任务上执行。

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
项目管理员	√	√	√	√
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	√	×	×	×

角色	查看	创建	修改	删除
系统工程师	√	√	√	√
Committer	√	√	√	√
CI/CD工程师	√	√	√	×

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	更新	删除
项目管理员	√	√	√	√
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	√	×	×	×
CI/CD工程师	√	√	√	×

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
项目管理员	√	√	√	√
项目经理	√	√	√	√
开发人员	√	√	√	√

角色	查看	创建	修改	执行
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	√	√	√	√
CI/CD工程师	√	√	√	√

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
项目管理员	√	√	√	√	√	√
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	√	√	√	√	√	√
CI/CD工程师	√	√	√	×	√	×

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
项目管理员	√	√	√	√	√	√
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	√	√	√	√	√	√
CI/CD工程师	√	√	√	×	√	×

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
项目管理员	√	√	√	√	√	√
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	√	×	×	×	×	×

角色	查看	创建	修改	删除	执行	回滚
Committer	√	√	√	√	√	√
CI/CD工程师	√	√	√	×	√	×

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
项目管理员	√	√	√	√	√	√
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	√	√	√	√	√	√
CI/CD工程师	√	√	√	×	√	×

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
项目管理员	√	√	√	√
项目经理	√	√	√	√
开发人员	√	√	√	√

角色	创建	删除	修改	绑定
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	√	√	√	√
CI/CD工程师	√	×	√	×

任务模板权限

初始状态下，项目下各角色具有的任务模板权限如下表所示：

表 2-12 任务模板权限

角色	创建	删除	修改
项目管理员	√	√	√
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	√	√	√
CI/CD工程师	√	×	√

- 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

- 项目管理员、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**
- 项目管理员、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，且会在流水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。
- **继承项目权限配置**
- 开启继承项目权限配置后，角色权限与项目设置中角色的流水线权限一致。
 - 关闭继承项目权限配置后，可自定义该流水线的角色权限。
 - 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。

前提条件

已[开通并授权使用CodeArts Pipeline](#)。

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

图形化创建流水线界面简单明确，易于理解和操作，可快速进行流水线的配置，对于新手用户更加友好。代码化创建流水线，可以将其纳入版本控制系统中，实现对流水线配置的版本管理。这样可以轻松地回滚到之前的版本或者查看历史变更记录，同时确保流水线的创建过程是可重复的，减少人为错误的可能性。

YAML化流水线易于修改和测试，团队可以快速迭代和优化流水线，从而提高开发和部署的效率，便于团队协作和复杂项目的长期维护，能够显著提高开发和部署的自动化水平和效率。

约束与限制

- 初始状态下，当用户角色为以下角色时，可创建流水线。
 - 项目管理员。
 - 项目经理。
 - 开发人员。
 - 系统工程师。
 - Committer。
- 项目管理员可修改其他角色的权限，权限配置请参考[授权流水线服务](#)。

4.2 新建流水线（图形化）

准备工作

- 已[开通并授权使用CodeArts Pipeline](#)。
- [新建项目](#)。命名为project01。
- 使用CodeArts Repo流水线源：[新建代码仓库](#)。命名为Repo01
- [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。

约束与限制

- 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。
- 新建流水线自定义模板。
 - 系统模板只能用来复制或生成流水线，不能编辑和删除。
 - 模板中无需配置流水线源。

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	取值样例
名称	流水线的名称，支持自定义名称。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。	输入“pipeline01”。
委托URN	IAM的委托唯一标识。未配置时，使用默认委托。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务。	-
所属项目	流水线归属的项目。 • 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。 • 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。	默认选择的项目，此处默认为“project01”。
流水线分组	流水线的分组，可在新建流水线时选择分组，同时也可新建或编辑分组。	-

参数项	说明	取值样例
流水线源	流水线关联的源类型。 - CodeArts代码源 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。 - 第三方代码源 - 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。 - GitHub：连接GitHub账号后可以获取该账号下的仓库、分支等信息。 - GitCode：连接GitCode账号后可以获取该账号下的仓库、分支等信息。 - 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。 - Artifact：连接Artifact账号后可以获取该制品仓库的仓库类型和版本信息。 - 其他项目Repo：从其他项目的代码托管中拉取代码，请选择已有的项目、该项目下已经创建的代码仓以及默认分支。 - 制品源 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。 说明 - 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，具体问题请参考常见问题。	选择“Repo”。
创建方式	创建流水线的方式。关联本项目Repo代码源时，可以选择图形化编排和YAML化编排两种方式。 - 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。 - YAML化编排：以代码化方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。请参考新建流水线（YAML化）。	选择“图形化编排”。
服务扩展点	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。可选择准备工作中提前创建的扩展点，也可单击“新建服务扩展点”进行创建，具体操作请参考新建CodeArts服务扩展点。	-
项目	当流水线源选择“其他项目Repo”时，所要拉取代码的代码托管所属项目。	-
代码仓	流水线关联的代码仓库，可选择提前新建好的仓库，也可单击“新建仓库”进行创建，新建仓库可参考新建代码仓库。	选择“Repo01”。

参数项	说明	取值样例
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。	选择“master”。
Repo HTTPS 授权	该参数为非必填。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相关插件。可选择 准备工作中 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 新建CodeArts服务扩展点 。	-
别名	该参数为非必填。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。 仅支持输入大小写英文字母、数字、“_”，长度不超过128个字符。	-
描述	该参数为非必填。对流水线的描述，长度不超过1024个字符。	-
来源制品仓库类型	当流水线关联的源类型为Artifact流水线源时，选择来源制品仓库类型。可选择“软件发布库”或“Docker库”。	-
默认版本	运行流水线时，默认使用的版本。可选择“Latest”或“Specific Version”。当流水线关联的源类型为Artifact流水线源时，需填写该参数。	-
制品	软件开发过程产生的软件包。当流水线关联的源类型为Artifact流水线源时需填写该参数。	-
组织	当流水线关联的源类型为SWR流水线源时，选择SWR组织。 当流水线关联的源类型为Artifact流水线源时，选择当前项目下的组织。 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	当流水线关联的源类型为SWR流水线源时，选择组织下的镜像。 当流水线关联的源类型为Artifact流水线源时，选择当前项目组织下的镜像。	-
指定版本	当流水线关联的源类型为SWR流水线源时，选择镜像的版本。	-

参数项	说明	取值样例
制品源别名	当流水线关联的源类型为SWR流水线源或Artifact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。 仅支持输入中文、大小写英文字母、数字、“-”、“_”、“.”，长度不超过128个字符。	-

- 步骤3** 单击“下一步”，进入“模板选择”页面。此处以“空模板”为例创建。
- 选择系统模板或自定义模板，即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](#)。
 - 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”，完成流水线的创建。

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。

图 4-1 任务编排



----结束

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，在企业内完成复用和发布流程标准化。

表 4-2 流水线的常用系统模板

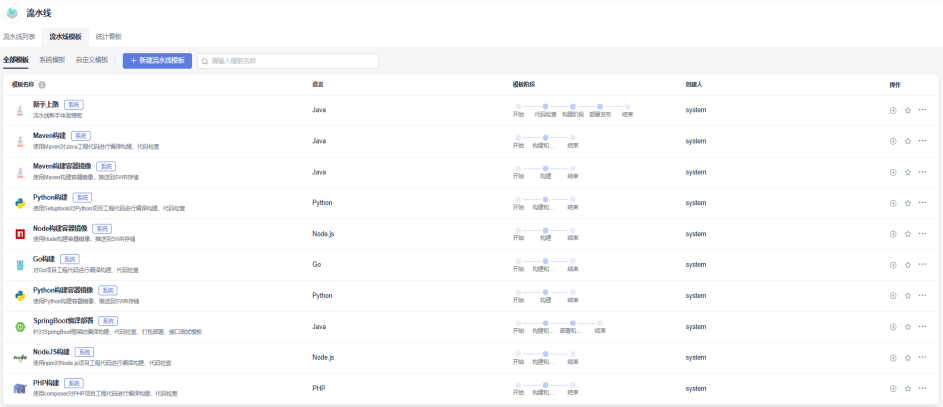
模板名称	模板解释
Maven构建	- 该模板使用Maven模板进行构建。 - 编程语言：JAVA。 - 编译工具：Maven3.5.3、JDK1.8。
Node构建容器镜像	- 该模板使用NPM模板进行构建。 - 编程语言：JAVASCRIPT。 - 编译工具：nodejs12.7.0。

模板名称	模板解释
Maven构建容器镜像	- 该模板使用Maven模板进行构建。 - 编程语言：JAVA。 - 编译工具：Maven3.5.3、JDK1.8。
Python构建	- 该模板使用Setuptools对Python项目工程代码进行编译构建、代码检查。 - 编程语言：Python。 - 编译工具：python3.6。
Go构建	- 该模板使用Go模板进行构建。 - 编程语言：Go。 - 编译工具：go-1.10.3。
Python构建容器镜像	- 该模板使用Setuptools模板进行构建，推送到SWR存储。 - 编程语言：Python。 - 编译工具：python3.6。
SpringBoot编译部署	- 该模板使用Maven模板进行构建。 - 使用语言：JAVA。 - 编译工具：Maven3.5.3、JDK1.8。
NodeJS构建	- 该模板使用NodeJS模板进行构建。 - 编程语言：JAVASCRIPT。 - 编译工具：nodejs12.7.0。
PHP构建	- 该模板使用PHP模板进行构建。 - 编程语言：PHP。 - 编译工具：php7.3.3。

- **流水线模板列表页入口**

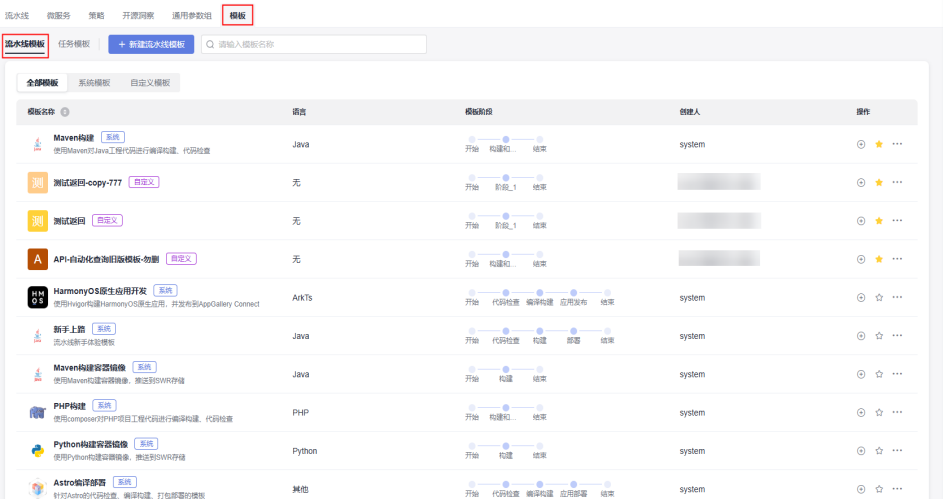
- 首页入口：进入流水线服务首页，默认展示“流水线列表”页面，单击“流水线模板”，进入“流水线模板”页面。

图 4-2 流水线模板列表



- 项目入口：进入项目下流水线页面，单击“模板”，进入“流水线模板”页面。

图 4-3 流水线模板



流水线模板列表操作项说明如下：

表 4-3 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★，再次单击可以取消收藏。

操作项	说明
...	- 单击后选择“编辑”，进入“任务编排”页面，可编辑模板。 - 单击后选择“复制”，进入“任务编排”页面，可复制模板。 - 单击后选择“删除”，弹出“删除模板”弹窗，可根据提示删除模板。

- 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”页面。
 - 根据需要配置模板基本信息、任务编排、参数设置等。
 - 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、ArkTs，默认选择“无”）、模板描述信息。

图 4-4 基本信息

名称

pipeline模板01

语言

无

描述

请填写模板描述

0/1,024

- 配置阶段的准入类型。
 - 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。
 - 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。
 - 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。
- 配置准出条件。
 - 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”。
 - 填写准出条件名称，选择策略。

 **标准策略准出条件**
选择一个标准插件策略用于门禁拦截。

* 名称

标准策略/准出条件

* 策略 [策略管理](#)

系统策略

CodeArtsCheck问题数

检查项	比较	阈值
问题总数	<=	10

3) 单击“确定”，即可添加一个准出条件。

- 任务编排：流水线模板支持添加所有的插件，包括官方插件、第三方插件和自定义插件，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时，会同步创建相应类型的任务。
- 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，自定义参数类型包括字符串类型、自增长类型和枚举类型，配置方法请参见[配置流水线自定义参数](#)。

d. 单击“保存”，弹出“创建模板成功”提示，完成流水线模板的创建。

4.3 新建流水线（YAML 化）

准备工作

- 已[开通并授权使用CodeArts Pipeline](#)。
- [新建项目](#)。命名为project01。
- [新建代码仓库](#)。命名为Repo02。
- 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](#)。

约束与限制

YAML流水线仅支持使用CodeArts Repo代码仓。

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 }}
```

新建 YAML 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4-4配置流水线基本信息。

表 4-4 流水线基本信息参数说明

参数项	说明	取值样例
名称	流水线的名称，支持自定义。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。	输入“pipeline02”。
委托 URN	IAM的委托唯一标识。未配置时，使用默认委托。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务。	-
所属项目	流水线归属的项目。 - 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。 - 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。	默认选择的项目，此处默认为“project01”。
流水线分组	流水线的分组，可在新建流水线时选择分组，同时也可新建或编辑分组。	-
流水线源	流水线关联的源类型，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。	选择“Repo”。

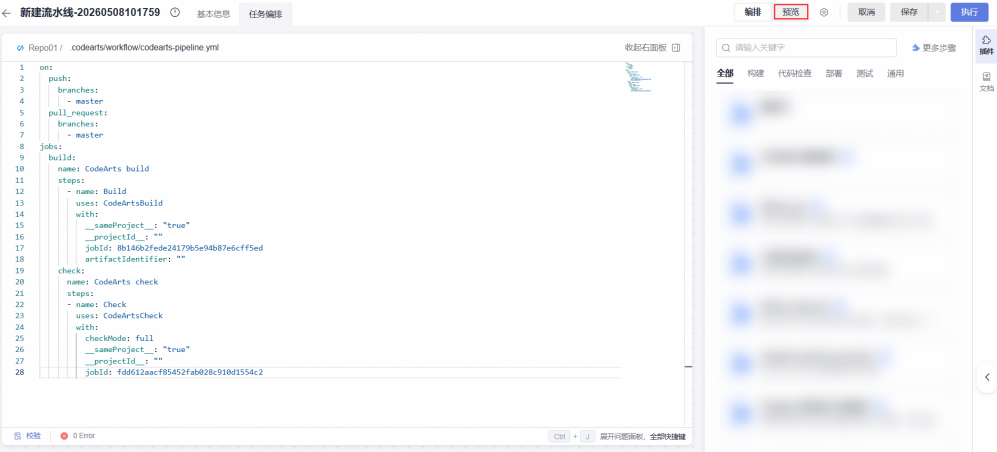
参数项	说明	取值样例
创建方式	创建流水线的方式。选择“YAML化编排”：以代码化YAML方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。	选择“YAML化编排”。
代码仓	流水线关联的代码仓库，可选择提前新建好的仓库，也可单击“新建仓库”进行创建，新建仓库可参考 新建代码仓库 。	选择“Repo02”。
默认分支	仓库默认分支，手工或定时执行流水线时默认使用的分支。	选择“master”。
配置方式	- 新建yaml文件：在编排流水线时，创建一个新的YAML文件。 - 已有yaml文件：基于已有YAML文件进行编排，编排后的内容会覆盖原YAML文件。YAML文件编写请参考YAML语法规则。	选择“新建yaml文件”。
选择yaml文件	配置方式选择“已有yaml文件”时，需配置此参数，此参数为必填项。 选择提前准备好并存储在代码仓库中的YAML文件，选择对应分支并输入YAML文件的相对路径。	-
Repo HTTPS 授权	该参数为非必填。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相关插件。可选择 准备工作 中提前创建的扩展点，也可单击“新建授权”进行创建，具体操作请参考 新建CodeArts服务扩展点 。	-
别名	该参数为非必填。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。 仅支持输入大小写英文字母、数字、“_”，长度不超过128个字符。	-
描述	该参数为非必填。对流水线的描述，长度不超过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，此处以串联编译构建任务、代码检查任务为例。
- 右侧为流水线插件列表，请根据需要进行添加。

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。

图 4-5 任务编排



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

更新.codeartspipeline/pipeline.yml

☒ 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”，弹出“保存信息成功”提示，完成流水线的创建。

----结束

5 新建流水线（新版）

5.1 新建流水线

图形化创建流水线界面简单明确，易于理解 and 操作，可快速进行流水线的配置，对于新手用户更加友好。代码化创建流水线，可以将其纳入版本控制系统中，实现对流水线配置的版本管理。这样可以轻松地回滚到之前的版本或者查看历史变更记录，同时确保流水线的创建过程是可重复的，减少人为错误的可能性。

YAML化流水线易于修改和测试，团队可以快速迭代和优化流水线，从而提高开发和部署的效率，便于团队协作和复杂项目的长期维护，能够显著提高开发和部署的自动化水平和效率。

约束与限制

- 初始状态下，当用户角色为以下角色时，可创建流水线。
 - 项目管理员。
 - 项目经理。
 - 开发人员。
 - 系统工程师。
 - Committer。
- 项目管理员可修改其他角色的权限，权限配置请参考[授权流水线服务](#)。

5.2 新建流水线（内置模式）

准备工作

- 已[开通并授权使用CodeArts Pipeline](#)。
- [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。

约束与限制

- 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。

新建流水线

- 步骤1 [访问CodeArts Pipeline首页](#)。
- 步骤2 单击“新建流水线”，弹出“新建流水线”弹窗，参考[表5-1](#)配置流水线基本信息。

表 5-1 流水线基本信息参数说明

参数项	说明	取值样例
名称	流水线的名称，支持自定义名称。支持中文、大小写英文字母、数字、中划线、下划线，长度不超过128个字符。	输入 “pipeline01” 。
流水线保存方式	创建并保存流水线的方式。 - 内置模式：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。 - 远端模式：以代码化方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。流水线配置通过YAML文件保存在Git仓库。请参考新建流水线（YAML化）。	选择“内置模式”
流水线分组	流水线的分组，可在新建流水线时选择分组，同时也可新建或编辑分组。	-

参数项	说明	取值样例
流水线源	流水线关联的源类型。 ● CodeArts代码源 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。 ● 第三方代码源 - 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。 - GitHub：连接GitHub账号后可以获取该账号下的仓库、分支等信息。 - GitCode：连接GitCode账号后可以获取该账号下的仓库、分支等信息。 - 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。 - Artifact：连接Artifact账号后可以获取该制品仓库的仓库类型和版本信息。 - 其他项目Repo：从其他项目的代码托管中拉取代码，请选择已有的项目、该项目下已经创建的代码仓以及默认分支。 ● 制品源 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。 说明 ● 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，具体问题请参考常见问题。	选择“Repo”。
服务扩展点	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 准备工作 中提前创建的扩展点，也可单击“新建服务扩展点”进行创建，具体操作请参考 管理CodeArts服务扩展点 。	-
代码仓	流水线关联的代码仓库，可选择提前新建好的仓库，也可单击“新建仓库”进行创建，新建仓库可参考 新建代码仓库 。	选择“Repo01”。
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。	选择“master”。
Repo HTTPS 授权	该参数为非必填。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相关插件。可选择 准备工作 中提前创建的扩展点，也可单击“新建授权”进行创建，具体操作请参考 新建CodeArts服务扩展点 。	-

参数项	说明	取值样例
别名	该参数为非必填。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。 仅支持输入大小写英文字母、数字、“_”，长度不超过128个字符。	-
来源制品仓库类型	当流水线关联的源类型为Artifact流水线源时，选择来源制品仓库类型。可选择“软件发布库”或“Docker库”。	-
默认版本	运行流水线时，默认使用的版本。可选择“Latest”或“Specific Version”。当流水线关联的源类型为Artifact流水线源时，需填写该参数。	-
制品	软件开发过程产生的软件包。当流水线关联的源类型为Artifact流水线源时需填写该参数。	-
组织	当流水线关联的源类型为SWR流水线源时，选择SWR组织。 当流水线关联的源类型为Artifact流水线源时，选择当前项目下的组织。 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	当流水线关联的源类型为SWR流水线源时，选择组织下的镜像。 当流水线关联的源类型为Artifact流水线源时，选择当前项目组织下的镜像。	-
指定版本	当流水线关联的源类型为SWR流水线源时，选择镜像的版本。	-
制品源别名	当流水线关联的源类型为SWR流水线源或Artifact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。 仅支持输入中文、大小写英文字母、数字、“-”、“_”、“.”，长度不超过128个字符。	-
委托URN	IAM的委托唯一标识。未配置时，使用默认委托。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务。	-
描述	该参数为非必填。对流水线的描述，长度不超过1024个字符。	-

步骤3 单击“新建”，进入“任务编排”页面。此处以“空模板”为例创建。

图 5-1 任务编排



----结束

5.3 新建流水线（远端模式）

准备工作

- 已[开通并授权使用CodeArts Pipeline](#)。
- [新建项目](#)。
- [新建代码仓库](#)。

前提条件

- 请确保此项目已配置委托URN。
- 请确保此代码仓库中的成员已包含所需的委托URN。查看方法如下。
 - a. 访问流水线服务首页。
 - b. 在左侧导航栏中，单击“代码 > 代码托管”，进入“代码托管”页面。
 - c. 单击需要查看仓库的名称，进入仓库的“代码”页面。
 - d. 单击“成员”，查看成员列表中是否包含所需的委托URN。
- 请确保用户具有将YAML文件推送到仓库分支的权限。查看方法如下（此处以默认分支为例）。
 - a. 访问流水线服务首页。
 - b. 在左侧导航栏中，单击“代码 > 代码托管”，进入“代码托管”页面。
 - c. 单击需要查看仓库的名称，进入仓库的“代码”页面。
 - d. 单击“设置 > 策略设置 > 保护分支”，将鼠标移动至“推送”下的，查看权限信息。
 - e. 若需配置权限，单击“操作”列的，根据需要配置相关权限。

新建 YAML 流水线

- 步骤1 [访问CodeArts Pipeline首页](#)。
- 步骤2 单击“新建流水线”，弹出“新建流水线”弹窗，参考[表5-2](#)配置流水线基本信息。

表 5-2 流水线基本信息参数说明

参数项	说明	取值样例
名称	流水线的名称，支持自定义。支持中文、大小写英文字母、数字、中划线、下划线，长度不超过128个字符。	输入“pipeline02”。
流水线保存方式	创建并保存流水线的方式。 - 内置模式：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。 - 远端模式：以代码化方式编排流水线，支持语法自动补齐和校验，方便多流水线批量配置。流水线配置通过YAML文件保存在GIT仓库。请参考新建流水线（YAML化）。	选择“远端模式”
流水线分组	流水线的分组，可在新建流水线时选择分组，同时也可新建或编辑分组。	-
远端仓库类型	流水线关联的仓库类型，包含Repo或GitCode两种类型。	-
服务扩展点	关联GitCode仓库类型时，需通过服务扩展点实现与对应仓库的连接。可单击“新建服务扩展点”进行创建，也可以提前创建好扩展点。具体操作请参考 管理CodeArts服务扩展点 。	-
代码仓	流水线关联的代码仓库，可选择提前新建好的仓库，也可单击“新建仓库”进行创建，新建仓库可参考 新建代码仓库 。	选择“Repo02”。
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。	选择“master”。
YAML文件	yaml配置文件路径必须在.codearts/workflow目录下。 - 新建yaml文件：在编排流水线时，创建一个新的YAML文件。 - 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文件编写请参考配置流水线（YAML化）。	选择“新建yaml文件”。
提交信息	配置方式选择“新建yaml文件”时，需配置此参数，此参数为必填项。 新建yaml文件时的命名信息。	填写“新建.codearts/codearts-pipeline.yml”
委托URN	IAM的委托唯一标识。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务，否则使用运行人的身份和权限。	填写本项目下已配置的委托URN。
描述	该参数为非必填。对流水线的描述，长度不超过1024个字符。	-

步骤3 配置完基本信息，单击“新建”，进入“任务编排”页面。

- 左侧为YAML编辑区域，请参考[配置流水线（YAML化）](#)编写YAML。
- 右侧为流水线插件列表，请根据需要进行添加。

步骤4 编排完任务，单击“保存”或“保存”下拉框中的“保存并返回”，弹出“保存信息成功”提示，完成YAML流水线的创建与保存。

单击“执行”，弹出“执行配置”弹窗，参考[执行流水线](#)配置相关信息，单击“确定”，完成YAML流水线的创建与执行。

----结束

6 配置流水线

6.1 编排流水线阶段

什么是流水线阶段

流水线阶段是流水线上的基本组成部分，主要用于对流水线的任务做业务流程上的分类编排管理，每个阶段都有明确的输入、处理逻辑和输出，并且按照一定的顺序依次执行，让流水线上的业务流更加直观。用户可参考本章节进行配置。

约束与限制

- 配置流水线的准出条件：
 - 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。
 - 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。
 - 同一阶段中可以配置多个不同的准出条件。
- 一条流水线最多可配置16个阶段（包含后处理阶段）。

费用说明

规则与策略功能需升级CodeArts套餐至基础版才可使用，开通套餐可参考[CodeArts计费模式概述](#)。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以编排流水线阶段，权限配置请参考[授权流水线服务](#)。

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。

图 6-1 任务编排



表 6-1 管理阶段

操作项	说明
编辑阶段	编辑流水线的阶段，单击，弹出“编辑阶段”弹窗，可以配置阶段名称和阶段是否总是运行。 - 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。 - 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。 - 快速停止：选择“是”，表示流水线执行时，在并行执行的同一阶段内，若任一子任务失败，则立即停止该阶段所有并行任务，并标志本阶段为已停止。 - 快速停止：选择“否”，表示流水线执行时，在并行执行的同一阶段内，若任一子任务失败，则不会立即停止该阶段所有并行任务。
复制阶段	复制流水线的阶段，单击，可以实现复制阶段。
删除阶段	删除流水线的阶段，单击，根据删除提示确认是否删除阶段。
排序阶段	对流水线的阶段进行排序，单击不松开，可以移动阶段，调整阶段顺序。
阶段准入	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，可以配置阶段的准入类型。 - 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。 - 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。 - 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。

操作项	说明						
准出条件	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。 ● 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。配置方法请参考配置流水线规则。 ● 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，可分为租户级策略和项目级策略。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。 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”。 2. 填写准出条件名称，选择策略。  标准策略准出条件 选择一个标准插件策略用于门禁拦截。 * 名称 标准策略准出条件 * 策略 系统策略 策略管理 CodeArtsCheck问题数 <table><tr><th>检查项</th><th>比较</th><th>阈值</th></tr><tr><td>问题总数</td><td><=</td><td>10</td></tr></table> 3. 单击“确定”，即可添加一个准出条件。	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 （可选）[配置流水线后处理阶段](#)。
- 步骤5 阶段编排完成后，单击“保存”，弹出“保存信息成功”提示，流水线保存成功。



----结束

配置流水线后处理阶段

在流水线任务编排页面，可根据需要配置后处理阶段。配置后，执行流水线时无论前置阶段成功与否，后处理阶段均会执行。支持用户按需添加插件，也可配置自定义开发插件，实现资源清理、第三方系统回调、消息通知等功能。

- 步骤1 访问CodeArts Pipeline首页。
- 步骤2 选择目标流水线，单击，选择编辑，进入流水线编排页面。
- 步骤3 在后处理阶段下，配置任务插件，配置方法同配置流水线任务。



- 一个流水线至多只能添加一个“后处理阶段”，并且该阶段位置必须在最后。
- 默认生成的后处理阶段，总是运行默认选择“是”，也可以手动修改。
- 如果配置了后处理阶段，并且运行时候选择了该阶段，那么执行流水线时，这个阶段始终会被执行到。

----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 6-2 新建规则



表 6-2 规则参数说明

参数项	说明	取值样例
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。	输入“Test-1”。
类型	规则类型，与插件类型对应。支持构建、代码检查、测试、部署和通用五种类型插件。 - 构建：用于代码构建的执行插件。 - 代码检查：用于检查代码质量的插件。 - 测试：用于测试接口性能的插件。 - 部署：用于发布应用到目标环境的插件。 - 通用：用于扩展服务功能的插件。	选择“代码检查”。
选择插件	插件类型下的所有插件。包括构建类型插件、代码检查类型插件、测试类型插件、部署类型插件和通用类型插件。 - 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。 - 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。 - 测试类型插件：通过设置相关测试结果的阈值，对测试阶段结果进行判断。如：Testplan接口测试插件，对测试套件中测试用例通过率进行判断。	选择“Check代码插件”。
插件版本	插件的所有版本。仅支持配置了阈值的插件版本。	选择“0.0.3”。
阈值配置	根据选中的插件版本自动获取的插件输出阈值配置，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。 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。	-
代码度量	代码度量可准确的反映提交代码的质量，助力用户及时发现并修改问题，提升研发效能。代码度量包括以下两种类型的问题： - 平均圈复杂度：代码的复杂度。 - 代码重复率：代码检查服务通过重复块、重复行、重复率来识别被测代码的重复情况。 详情请参见 查看代码度量 。	-

步骤5 单击“确定”，弹出“创建规则成功”提示，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，确认删除后，单击“确定”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

配置流水线策略

流水线策略包括租户级策略和项目级策略。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。**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。**且租户级策略和项目级策略下，单个策略中最多可配置20个规则。**

配置流水线租户级策略

- 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 6-3 租户级策略参数信息

参数项	说明	取值样例
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。	输入“Test01”。
规则	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： - 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。 - 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。	勾选“Test-1”。

步骤5 单击“确定”，弹出“新建策略成功”提示，即生成一条策略。



- 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

- 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。

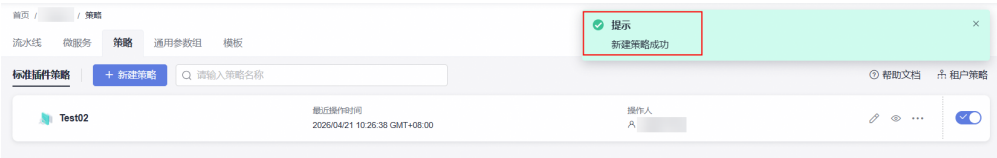
图 6-3 新建策略



表 6-4 项目级策略参数信息

参数项	说明	取值样例
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。	输入 “Test02”。
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： - 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。 - 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。	勾选 “Test-1”。

- 步骤4 单击“确定”，弹出“新建策略成功”提示，即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

6.2 编排流水线任务

什么是流水线任务

流水线任务是实现流水线功能的基本组成部分，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，根据所配置的执行顺序和时间逐步运行。用户可参考本章节进行配置。

约束与限制

- 单租户最多可创建200个执行方案。
- 修改任务模板后，所有关联了该模板的流水线都将被更新，且此操作不可逆。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以编排流水线任务，权限配置请参考[授权流水线服务](#)。

配置流水线任务

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”，可根据需要选择从“空任务新建”或“从模板新建”。使用空模板新建流水线时，“阶段_1”的“新建任务”默认为空任务，无法选择任务模板。



- 单击“从空任务新建”，弹出“从空任务新建”侧滑框，参考表1配置任务信息。
- 单击“从模板新建”，弹出“从模板新建”侧滑框，选用所需要的模板或前往任务模板创建，任务模板相关操作请参考表3配置任务模板。
 - 单击“复制”，进入所选用模板的任务配置界面，可根据需要修改参数信息。
 - 单击“关联模板”，进入所选用模板的任务配置界面，直接引用该任务模板，参数信息不可修改。

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 6-5 配置任务

操作项	说明
添加插件	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，插件来源请参考管理流水线扩展插件。 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： ● 填写插件名称。 ● 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。 ● 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。 ● 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。 ● “挂起流水线”插件只能添加在不含并行任务的阶段中。
删除插件	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。
更换插件	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“更换步骤”，可以返回插件列表，更换当前插件。
排序插件	单击插件卡片不松开，可以移动插件，调整插件顺序。

操作项	说明
任务配置	配置任务ID、资源池高级设置、任务选择和执行条件。 ● 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。 ● 资源池高级设置：流水线任务支持三种类型的运行环境，包括默认资源池、专属资源池和自定义资源池。资源池高级设置仅针对非任务级插件。 - 默认资源池：系统默认的多租户共享资源池，基于容器隔离任务运行环境。 - 专属资源池：系统提供的租户专享资源池，支持通过内网访问用户私有VPC网络。 - 自定义资源池：用户将自有服务器资源，通过网络代理的方式接入，并由CodeArts纳管。详细操作请参见资源池管理。 说明 仅“华北-北京一”、“华北-北京四”、“华南-广州”局点支持资源池类型选择“专属资源池”。 ● 任务选择 - 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。 - 禁止选择：禁止执行，执行流水线时该任务禁止勾选。 - 默认选择：执行流水线时该任务默认勾选但可按需修改。 - 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。 ● 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。 - 前置任务未选择仍执行：上一个串行任务状态执行成功或未选择，都继续执行当前任务。 - 前置任务成功时执行：上一个串行任务执行成功才执行当前任务。 - 前置任务失败时执行：上一个串行任务执行失败才执行当前任务。 - 始终执行：始终执行当前任务，上一个串行任务状态是失败、成功、中止或忽略执行，都继续执行当前任务。 - 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。 ■ 表达式标签模式：单击标签模式下的输入框，可在弹出面板中，点选对应上下文内容、函数或操作符，也可直接在输入框内手动输入表达式，表达式编写请参考配置流水线执行条件表达式。 手动输入表达式示例： 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：

操作项	说明
	jobs.job_1.status == 'COMPLETED' jobs.job_1.status == 'FAILED' ▪ 表达式代码模式：标签模式下点选参数解析后的表达式字符串格式，不支持手动输入。

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 6-6 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。
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。 任务并行执行时不支持调整顺序。

步骤6 任务编排完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。
配置流水线任务过程中遇到的问题请参考[常见报错解决方法](#)。

----结束

配置执行方案

流水线执行方案是对流水线中各项任务如何执行，以及在执行过程中如何进行资源分配、监控和管理等方面进行详细规划和设计的方案，是确保流水线能够高效、稳定运行，实现预期目标的重要依据，流水线支持提前配置好执行方案并预置好执行配置，可在定时任务和手动执行时选择。

- 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“执行方案”。
- 步骤3** 单击“新建执行方案”，填写执行方案名称，配置流水线源信息、配置运行参数、选择执行阶段、填写执行描述，配置方法同[执行流水线](#)。

图 6-4 新建执行方案

新建执行方案

执行方案名称

请输入执行方案名称

流水线源

代码源

Maven-Build-Demo

分支/Tag/CommitId

分支

master

运行参数配置

执行阶段配置

所有阶段

阶段_1

新建任务

执行描述

请填写执行描述

0/1,024

保存

取消

步骤4 单击“保存”，弹出“新建执行方案成功”提示，完成执行方案的新建。可根据需要进行编辑、复制、删除。

提示
新建执行方案成功

执行方案

+ 新建执行方案

请输入执行方案名称

执行方案名称	创建人	创建时间	更新时间	操作
plan01	A	2026/02/02 15:11:47 GMT+08:00	2026/02/02 15:11:47 GMT+08:00	编辑 复制 删除

----结束

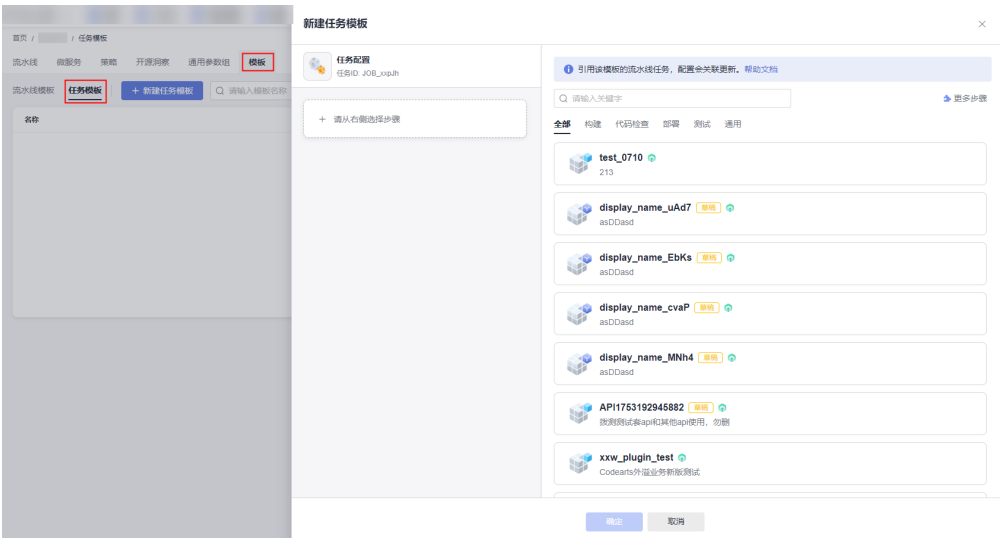
配置任务模板

流水线任务模板是一种预先定义好的任务流程模型，用于在流水线中快速创建和执行具有特定功能和步骤的任务，包含了任务的各种配置信息和执行逻辑，能够提高任务创建的效率和一致性，减少重复工作。

- 步骤1 访问CodeArts Pipeline首页。

步骤2 单击“模板”，进入模板页面，单击“任务模板”，进入“任务模板”页面。

步骤3 单击“新建任务模板”，弹出“新建任务模板”侧滑框，配置相关参数。



步骤4 为任务配置插件及任务信息，操作说明如下：

表 6-7 配置任务模板

操作项	说明	取值示例
添加插件	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，插件来源请参考管理流水线扩展插件。 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： - 填写插件名称。 - 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。 - 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。 - 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。	以添加“Build构建”插件为例，提前新建构建任务build01，添加插件时参数配置如下： - 插件名称：build构建。 - 项目配置：本项目。 - 请选择需要调度的任务：build01。 - 仓库：来自流水线
删除插件	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。	-
更换插件	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“更换步骤”，可以返回插件列表，更换当前插件。	-
排序插件	单击插件卡片不松开，可以移动插件，调整插件顺序。	-

操作项	说明	取值示例
任务配置	配置模板配置、任务ID、资源池高级设置、任务选择和执行条件。 ● 模板配置：包括模板名称和描述。 - 模板名称：流水线新建任务模板的名称。 - 描述：流水线新建任务模板的相关描述，长度不超过1024个字符。 ● 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、中划线和下划线，长度范围为1~128个字符。 说明 任务模板中，仅任务ID在编排流水线时是可编辑的。 ● 资源池高级设置：流水线任务支持三种类型的运行环境，包括默认资源池、专属资源池和自定义资源池。资源池高级设置仅针对非任务级插件。 - 默认资源池：系统默认的多租户共享资源池，基于容器隔离任务运行环境。 - 专属资源池：系统提供的租户专享资源池，支持通过内网访问用户私有VPC网络。 - 自定义资源池：用户将自有服务器资源，通过网络代理的方式接入，并由CodeArts纳管。详细操作请参见资源池管理。 说明 仅“华北-北京一”、“华北-北京四”、“华南-广州”局点支持资源池类型选择“专属资源池”。 ● 任务选择 - 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。 - 禁止选择：禁止执行，执行流水线时该任务禁止勾选。 - 默认选择：执行流水线时该任务默认勾选但可按需修改。 - 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。 ● 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。 - 前置任务未选择仍执行：上一个串行任务状态执行成功或未选择，都继续执行当前任务。 - 前置任务成功时执行：上一个串行任务执行成功才执行当前任务。 - 前置任务失败时执行：上一个串行任务执行失败才执行当前任务。	以添加“build构建任务模板”为例，参数配置如下： ● 模板名称：build构建任务模板。 ● ID：JOB_KJbee。 ● 任务选择：默认选择。 ● 执行条件：前置任务未执行仍选择。

操作项	说明	取值示例
	- 始终执行：始终执行当前任务，上一个串行任务状态是失败、成功、中止或忽略执行，都继续执行当前任务。 - 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。 ■ 表达式标签模式：单击标签模式下的输入框，可在弹出面板中，点选对应上下文内容、函数或操作符，也可直接在输入框内手动输入表达式，表达式编写请参考配置流水线执行条件表达式。 手动输入表达式示例： 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： <pre>jobs.job_1.status == 'COMPLETED' jobs.job_1.status == 'FAILED'</pre> ■ 表达式代码模式：标签模式下点选参数解析后的表达式字符串格式，不支持手动输入。	

步骤5 配置完任务，单击“确定”，跳转至模板页面，完成任务模板的新建。



- 单击，弹出“编辑任务模板”侧滑框，可对该任务模板进行编辑。
编辑完任务模板，单击“确定”，弹出“编辑任务模板”弹窗，若确认修改，单击“一键输入”，单击“确定”，完成对任务模板的修改。

编辑任务模板

×

⚠ 修改当前任务模板后，所有关联的流水线都会进行修改。这是一个不可恢复的操作，请谨慎对待

名称	流水线ID
新建流水线-20251024153605	1d64983b34424e839eb18d84c90dd657

共 1 条

如您确定要进行操作，请输入 YES [一键输入](#)

YES

确定

取消

- 单击 ，弹出“删除任务模板”弹窗，单击“确定”可删除该任务模板。
如果该任务模板已被流水线引用，删除任务模板时，单击 ，会弹出“删除任务模板”对话框。若确认删除，单击“一键输入”，然后再次单击“确定”以完成任务模板的删除。

删除任务模板

×

ℹ 当前已有1条流水线引用该任务模板

即将删除任务模板 新建任务模板-90953

名称	创建人
新建任务模板-90953	

如您确定要删除，请输入 DELETE [一键输入](#)

DELETE

确定

取消

----结束

6.3 配置流水线插件

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线插件，权限配置请参考[授权流水线服务](#)。

6.3.1 Build 构建

调用编译构建能力进行构建。编译构建（CodeArts Build）基于云端大规模并发加速，为客户提供高速、低成本、配置简单的混合语言构建能力，帮助客户缩短构建时间，提升构建效率。详情请参见[编译构建CodeArts Build](#)。

图形化插件

在[编排流水线任务](#)时，添加“Build构建”插件，参数配置如[表6-8](#)。

表 6-8 “Build 构建” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128个字符。
项目配置	可选择本项目或其他项目。
项目选择	项目配置参数选择为“其他项目”时，选择所要调用的其他项目。
请选择需要调用的任务	选择本项目下或其他项目下和当前流水线源配置代码仓相同或构建来源配置为来自流水线的构建任务。
仓库	选择对应的代码仓。
产物标识	用于标识此构建任务的产物，例如：构建任务的ID为build_job，其中定义产物标识为demo，则后续任务中可以“通过\${{ jobs.build_job.artifacts.demo }}"获取产物信息。

代码化插件

在[编排流水线任务](#)时，添加“Build构建”插件，yaml语法配置如下。

- 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。

6.3.2 Check 代码检查

调用CodeArts Check能力进行代码检查。CodeArts Check是基于云端实现的代码检查服务，为用户提供代码风格、通用质量与网络安全等方面的检查能力，专业的扫描报告、便捷的问题闭环处理流程。详情请参见[代码检查CodeArts Check](#)。

图形化插件

在[编排流水线任务](#)时，添加“Check代码检查”插件，参数配置如[表1 “Check代码检查” 插件参数说明](#)。

表 6-9 “Check 代码检查” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 • 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 • 字符长度范围为1-128。
项目配置	可选择本项目或其他项目。
项目选择	单选框参数选择为“其他项目”时，选择所要调用的其他项目。
请选择需要调用的任务	选择本项目下和当前流水线源配置代码仓相同的代码检查任务。
仓库	选择对应的代码仓。
检查模式	Check代码检查的模式，根据实际需求设置检查模式，提高检查效率。

代码化插件

在[编排流水线任务](#)时，添加“Check代码检查”插件，yaml语法配置如下。

- **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。

6.3.3 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）。

图形化插件

在编排流水线任务时，添加“云原生发布”插件，参数配置如表6-10。

表 6-10 “云原生发布”插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 • 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 • 字符长度范围为1-128。
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，详见新建发布管理环境。
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}。

代码化插件

在编排流水线任务时，添加“云原生发布”插件，yaml语法配置如下。

- 云原生发布

```
uses: official_microservice_release
with:
  categories: 8dc56cd6c2cf44029181f04025ed173a
  env_id: 39c94e52fb0e4c03af97e3252c68e2a3
  artifact: swr.example.com/demo/springboot-helloworld:v1.1
```

 - categories: 任务发布的环境类别。
 - env_id: 需要发布的环境。
 - artifact: 部署发布时使用的镜像地址。

6.3.4 Deploy 部署

调用CodeArts Deploy能力进行部署。部署（原CloudDeploy）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。详情请参见[部署CodeArts Deploy](#)。

图形化插件

在[编排流水线任务](#)时，添加“Deploy部署”插件，参数配置如[表6-11](#)。

表 6-11 “Deploy 部署” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
请选择需要调用的任务	选择本项目下和当前流水线源配置代码仓相同的部署任务。

代码化插件

在[编排流水线任务](#)时，添加“Deploy部署”插件，yaml语法配置如下。

- Deploy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。

6.3.5 TestPlan 接口测试

调用CodeArts TestPlan能力进行测试。测试计划（CodeArts TestPlan）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。详情请参见[测试计划CodeArts TestPlan](#)。

图形化插件

在[编排流水线任务](#)时，添加“TestPlan接口测试”插件，参数配置如[表6-12](#)。

表 6-12 “TestPlan 接口测试” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
任务类型	测试任务的类型，选择“接口自动化”。
请选择需要调用的任务	选择本项目下和当前流水线源配置代码仓相同的接口自动化测试套件。
参数来源	- 使用选中环境的全局参数：支持选择多种任务类型中配置的不同环境，并使用对应环境中的参数。 - 使用全新执行参数：支持直接填写并覆盖选定测试任务类型中使用的全局参数。
环境	参数来源为“使用选中环境的全局参数”时，选择“默认环境”。

代码化插件

在编排流水线任务时，添加“TestPlan接口测试”插件，yaml语法配置如下。

- TestPlan接口测试**

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

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

6.3.6 子流水线

配置并调用项目下其他流水线任务。

图形化插件

在编排流水线任务时，添加“子流水线”插件，参数配置如表6-13。

表 6-13 “子流水线” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
请选择需要调用的任务	选择需要调用的本项目下其他的流水线。
分支/标签	设置子流水线的分支/标签，可选择使用子流水线默认分支/标签、“使用父流水线运行分支/标签”或“自定义分支”。
请输入子流水线启动分支	当“分支”选择“自定义分支”时，子流水线的启动分支，自定义设置。

代码化插件

在编排流水线任务时，添加“子流水线”插件，yaml语法配置如下。

- 子流水线**

```
uses: SubPipeline
with:
  pipelineId: 80ea2d9ffb94c20b9a0a0be47d3a0d8
  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 }}。

6.3.7 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。

图形化插件

在编排流水线任务时，添加“延时执行”插件，参数配置如表1 “延时执行插件” 插件参数说明。

表 6-14 “延时执行插件” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
选择延时方式	延时执行的方式，支持延时时长和定时时间。
延时时长	延时执行流水线的时长，可进行手动配置。
定时时间	定时执行流水线的时间，可进行手动配置。
时区	区域内的标准时间。

代码化插件

在编排流水线任务时，添加“延时执行”插件，yaml语法配置如下。

- 延时执行**

```
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。

6.3.8 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。

图形化插件

在编排流水线任务时，添加“人工审核”插件，参数配置如表1 “人工审核插件” 插件参数说明。

表 6-15 “人工审核插件” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
审核来源	配置审核人员的来源，可选择“项目成员”或“项目角色”。
审核人	当“审核来源”选择“项目成员”时，配置审核的人员。
审核模式	当“审核来源”选择“项目成员”时，配置审核的模式，可选择“全部人审核”或“任一人审核”。
审核角色	当“审核来源”选择“项目角色”时，选择审核角色，IPD类型项目中，项目创建者不会作为项目经理角色参与审核。
超时处理	配置超时未审核时的处理方式。
审核时长	配置审核时长，审核时长不能超过3天。
审核说明	对该审核任务的说明。

代码化插件

在编排流水线任务时，添加“人工审核”插件，yaml语法配置如下。

- 人工审核

```
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：按成员审核。
 - roles：按角色审核。
 - approvers：审核模式为“members”时的审核人员用户id。
 - roles：审核模式为roles时的角色。
 - checkStrategy：审核模式为“members”时的审核策略。
 - all：所有人员审核通过。
 - any：任一人员审核通过。
 - timeout：审核超时时长。
 - timeoutStrategy：审核超时时策略。

- reject：中止。
- pass：通过。
- comment：对审核的描述。

6.3.9 创建仓库 Tag

创建仓库Tag插件，为代码仓创建Tag并推送。

图形化插件

在编排流水线任务时，添加“创建仓库Tag”插件，参数配置如表6-16。

表 6-16 “创建仓库 Tag” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 • 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 • 字符长度范围为1-128。
Tag名	Tag名称。
仓库	选择对应的代码仓。

- 单击，开启“Tag存在时继续”，若创建的Tag在代码仓中已存在，则不报错，继续运行流水线，否则执行失败。

代码化插件

在编排流水线任务时，添加“创建仓库Tag”插件，yaml语法配置如下。

- **创建仓库Tag**

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

 - tagName：Tag名称。

6.3.10 Jenkins 任务

调用jenkins任务以扩展并实现自定义动作。

图形化插件

在编排流水线任务时，添加“Jenkins任务”插件，参数配置如表6-17。

表 6-17 “Jenkins 任务” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
Jenkins实例	Jenkins服务扩展点，请参考 Jenkins 新建Jenkins服务扩展点。
jobName	jenkins的job名称。
params	启动任务传递的参数（json格式）。
异步	是否异步执行。
描述	连接到终端的说明。

代码化插件

在[编排流水线任务](#)时，添加“Jenkins任务”插件，yaml语法配置如下。

- Jenkins任务**

```
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。

6.3.11 挂起流水线

配置以挂起当前流水线。

图形化插件

在[编排流水线任务](#)时，添加“挂起流水线”插件，参数配置如[表6-18](#)。

表 6-18 “挂起流水线” 插件参数说明

参数项	说明
名称	插件名称，可自定义修改。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。

代码化插件

在编排流水线任务时，添加“挂起流水线” 插件，yaml语法配置如下。

- 挂起流水线
uses: SuspendPipeline

6.3.12 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。

图形化插件

在编排流水线任务时，添加“执行镜像命令” 插件，参数配置如表6-19。

表 6-19 “执行镜像命令” 插件参数说明

参数项	说明
名称	插件名称。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
镜像属性	当前只支持公开镜像。
SWR镜像地址	待下载的SWR镜像的地址。获取方式如下： - 登录容器镜像服务。 - 在导航栏单击“我的镜像”，单击“镜像名称”进入镜像详情页面。 - 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。
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。

代码化插件

在编排流水线任务时，添加“执行镜像命令”插件，yaml语法配置如下。

- 执行镜像命令**

```
uses: official_docker_executor
with:
  attribute: public
  image: swr.example.com/demo/springboot-helloworld:v1.1
  command: sh start.sh
```

 - attribute：镜像属性。
 - image：SWR镜像地址。
 - command：容器启动命令。

6.3.13 第三方接口调用 v2

第三方接口调用插件，支持配置Web API调用第三方API。

- 此插件需要证书校验，不支持自定义证书校验和无证书校验。
- 接口返回码如果不是2XX，则默认为运行失败。

图形化插件

在编排流水线任务时，添加“第三方接口调用v2”插件，参数配置如表6-20。

表 6-20 “第三方接口调用 v2” 插件参数说明

参数项	说明
名称	插件名称。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 至少输入一个字符。
请求方式	可选择GET、POST、PATCH、PUT或DELETE。
请求路径	能被公网调用的接口，需要证书校验。
请求头	请求头，添加键值对。
请求参数	请求参数，添加键值对。
请求体	如果请求方式是POST、PATCH或PUT，则输入json格式的body体。

参数项	说明
输出	根对象名是response，可输出多个output。 ● 如果要拿到具体的请求返回码，并且输出的key是code，示例如下。 输出 ⓘ code response.status • ● 如果返回的请求头中有date字段，并且需要拿到date时间的值，则输出处添加示例如下。 输出 ⓘ myheaderField response.headers.date • ● 如果要输出整个body，示例如下 输出 ⓘ body response.data • ● 如果返回的json体如下，并且要取field对应的value值，示例如下。 <pre>{ "result": { "field": "value", "field1": "value1" } }</pre> 输出 ⓘ fieldKey response.data.result.field •

代码化插件

在编排流水线任务时，添加“第三方接口调用v2”插件，yaml语法配置如下。

- 第三方接口调用v2

```
uses: official_third_part_invoke
with:
  API_METHOD: GET
  API_URL: https://demo
  API_HEADER: '[{"key":"Content-Type","value":"application/json"}]'
  API_PARAM: '[{"key":"name","value":"value"}]'
  API_BODY: "{}"
  API_OUTPUT: '[{"key":"code","value":"response.status"}]'
```

6.3.14 Repo 代码仓分支合入

“Repo代码仓分支合入”插件提供CodeArts Repo代码仓分支合入功能。

图形化插件

在编排流水线任务时，添加“Repo代码仓分支合入”插件，参数配置如表6-21。

表 6-21 “Repo 代码仓分支合入” 插件参数说明

参数项	说明
名称	插件名称。 - 仅支持中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、斜杠、英文括号、中文括号及空格。 - 字符长度范围为1-128。
Repo Url	Repo服务代码仓https地址。
仓库名	代码仓库的名称。
访问令牌	Repo服务访问令牌（在“个人设置 > 代码托管 > 访问令牌”中生成）。
目标分支	源分支在合并时被合并至的分支。
源分支	需要被合并的分支。

代码化插件

在编排流水线任务时，添加“Repo代码仓分支合入”插件，yaml语法配置如下。

- Repo代码仓分支合入

```
jobs:
  JOB_msbBf:
    name: Repo Repository Merg
    needs: []
    select: selected_by_default
    steps:
      - name: Repo Repository Merg
        uses: official_repo_merge
        with:
          repository: "${REPO_URL}"
          repository_name: aass
          access_token: xxssxxss
          target_branch: master
          source_branch: dev
          identifier: 178235170957838b8f854-5b2f-4650-aaa8-b354d928c06a
        if: "${{ default() }}"
    runs-on:
      - default
```

6.4 配置流水线参数

什么是流水线参数

流水线参数是在流水线运行过程中使用的可配置值，用于控制流水线的行为、流程和输出结果，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10个通用参数组，每个参数组最多可以添加100个自定义参数。

约束与限制

- 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Tag名称。
- 当存在同名参数时，参数优先级为：系统预定义参数 > 自定义参数 > 通用参数组。
- 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，以最后关联的参数组的参数取值为准。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线参数，权限配置请参考[授权流水线服务](#)。

配置流水线参数

介绍流水线预定义参数、自定义参数和通用参数组。

流水线系统预定义参数

流水线系统预定义参数请参见[表6-22](#)，参数引用形式：**`${参数名}`**，在参数输入框输入“**`$`**”符号后会自动出现系统预定义参数列表。

表 6-22 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。
COMPONENT_ID	Component ID。
PIPELINE_ID	当前流水线ID。
PIPELINE_NUMBER	流水线执行编号。
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）。

参数信息	说明
PIPELINE_NAME	流水线名称。
REPO_URL	代码仓地址（HTTPS）。
MERGE_REQUEST_ACTION	MR/PR触发的action。action包括新建、合并、重新打开、更新和评论。
MERGE_REQUEST_TOPIC	MR/PR触发的topic。topic是指配置代码仓库合并请求时，变更请求里的变更主题。
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。
EXTERNAL_SYSTEM	触发流水线的外部系统。此处的外部系统指的是除Repo流水线源之外的其他流水线源。
EXTERNAL_USER_ID	外部系统触发流水线时的用户ID。此处的外部系统指的是除Repo流水线源之外的其他流水线源。
EXTERNAL_USER_NAME	外部系统触发流水线时的用户名称。此处的外部系统指的是除Repo流水线源之外的其他流水线源。
PIPELINE_RUN_ID	流水线执行ID。
MERGE_ID	合并请求ID。
MERGE_REQUEST_IID	合并请求IID。
WEBHOOK_PAYLOAD	Webhook请求负载信息。
COMPONENT_CHANGE_IDS	变更ID列表(<ID1>, <ID2>, ...<IDn>)。若基于变更运行则为运行的变更ID列表，若基于代码源运行则为微服务中关联对应代码仓的变更ID列表。
Repo01_REPOSITORY_NAME	仓库名称。Repo01为新建流水线前所创建的代码仓库。
Repo01_SOURCE_BRANCH	合并请求触发流水线时的仓库运行源分支名称。Repo01为新建流水线前所创建的代码仓库。

参数信息	说明
Repo01_TARGET_BRANCH	仓库运行目标分支名称。Repo01为新建流水线前所创建的代码仓库。
Repo01_TAG	仓库Tag名称。Repo01为新建流水线前所创建的代码仓库。
Repo01_COMMIT_ID	仓库执行时代码最后一次提交的ID。Repo01为新建流水线前所创建的代码仓库。
Repo01_COMMIT_ID_SHORT	执行时代码最后一次提交的短ID。Repo01为新建流水线前所创建的代码仓库。
Repo01_REPO_URL	代码仓地址（HTTPS）。Repo01为新建流水线前所创建的代码仓库。

配置流水线自定义参数

介绍流水线自定义参数的新建和配置。参数引用形式：**`${参数名}`**，在参数输入框输入“**`$`**”符号后会自动出现自定义参数列表。

- 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“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。

图 6-5 新建自定义参数



表 6-23 流水线自定义参数说明

参数信息	说明	取值样例
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、“ <code>_</code> ”，长度不超过128个字符。	输入“test01”。
类型	参数可选类型：字符串、自增长、枚举，默认选择字符串类型。	选择“字符串”。

参数信息	说明	取值样例
默认值	自定义参数的默认取值。 - 字符串：长度不超过8192个字符，可以为空。 - 自增长：长度不超过8192个字符。不能为空。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加1。 - 枚举：长度不超过1024个字符，仅支持中文、大小写英文字母、数字、“-”、“_”、“.”、“/”。 选择枚举类型后，在弹出的对话框中添加可选取值，可选取值不能为空。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。	输入“releaseversion”。
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，并在使用时自动解密。该参数在运行日志中不可见，且在流水线复制时不会被同步复制。 仅字符串类型参数可设置为私密参数。	-
运行时设置	打开“运行时设置”开关，在手动执行流水线时支持修改该参数的值。	单击  打开“运行时设置”。
参数描述	对该自定义参数的相关描述，长度不超过512个字符。	-
操作	单击“操作”列  ，可以新增参数。 单击“操作”列  ，可以删除参数。 当参数类型设置为“自增长”时，单击“操作”列中的  ，单击  ，可以开启“校验严格递增”功能。开启后，在执行流水线时将检查该参数是否严格递增。	-

步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，保存流水线成功。



----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，参考表6-24填写参数信息。

图 6-6 新建通用参数组

• 所属项目

project01

• 参数组名称

group01

描述

请填写描述

0/512

⚠ 参数删除后，请重新编辑保存引用该参数的任务，否则可能导致任务失败。

名称	类型	默认值	私有参数	参数描述	操作
 暂无自定义参数立即新建					

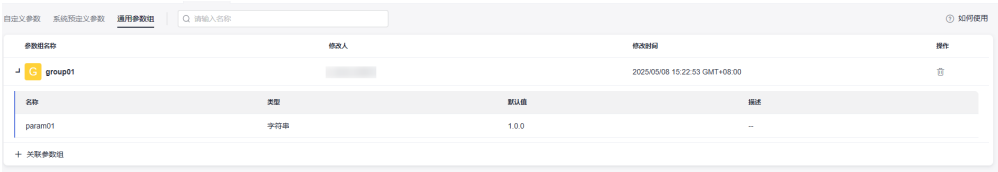
表 6-24 通用参数组参数说明

基本信息	说明	取值样例												
所属项目	通用参数组归属项目，不可更改。	默认为“project01”。												
参数组名称	仅支持大小写英文字母、数字、“_”，长度不超过128个字符。	输入“group01”。												
描述	对所新建的通用参数组的描述，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 自定义参数 。	配置示例如下图： <table><tr><th>ID</th><th>名称</th><th>类型</th><th>默认值</th><th>私有参数</th><th>描述</th></tr><tr><td>group01</td><td>group01</td><td>String</td><td>123</td><td>否</td><td></td></tr></table>	ID	名称	类型	默认值	私有参数	描述	group01	group01	String	123	否	
ID	名称	类型	默认值	私有参数	描述									
group01	group01	String	123	否										

- 步骤4** 单击“确定”，完成新建通用参数组。
- 步骤5** 找到需要关联通用参数组的流水线，单击操作列 ，选择“编辑”，单击“参数设置”，单击“通用参数组”，进入“通用参数组”页面。
- 步骤6** 单击“立即关联”，选择需要关联的参数组，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 6-7 关联参数组



步骤7 配置完成后，单击“保存”，弹出“操作成功”提示，保存成功。



----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 单击“参数设置”，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 6-8 新建构建任务参数



- 步骤3 单击“构建步骤”，切换至“构建步骤”页面，单击“点击添加构建步骤”，选择“上传软件包到软件发布库”，将鼠标移动至插件卡片，单击“添加”，将“发布版本号”配置为引用参数形式。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 6-9 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 ?

- 步骤4 配置完成后，单击“保存”，保存构建任务。
- 步骤5 **新建流水线**，选择“空白模板”，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 6-10 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，单击“设置为流水线参数”，可以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”，单击“参数设置”，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 6-11 新建流水线参数

名称	类型	默认值	私密参数 ①	运行时设置 ②	参数描述	操作
releaseversion	字符串	1.0.1	☐	☒		+ 删

- 步骤7 单击“任务编排”，切换至“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 6-12 引用流水线参数

*releaseversion

`\${releaseversion}`

- 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”弹窗，可查看“运行参数配置”。

运行参数配置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releaseversion参数执行流水线。

----结束

6.5 配置流水线触发设置

什么是流水线触发设置

流水线支持配置事件触发、定时任务、Webhook触发、并发策略和抢占策略五种触发设置。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；通过抢占策略，用户可以通过新触发的运行实例强行停止当前流水线中相同事件触发的其他所有运行实例。

约束与限制

- 配置流水线事件触发时。
 - 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。
 - Tag排除优先于Tag包含，即Tag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。
- 配置流水线定时执行时，定时任务最多可以设置10个。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线触发设置，权限配置请参考[授权流水线服务](#)。

配置流水线事件触发

事件触发包括代码提交事件、合并请求事件、创建Tag。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，选择“编辑”。

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

- 代码提交事件（代码托管/码云）

监听代码提交事件，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 6-13 配置代码提交事件



- 合并请求事件（代码托管/码云）

监听合并请求事件，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、重新打开和仅代码更新触发事件设置触发策略，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。
- 仅代码更新触发：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 6-14 配置合并请求事件

合并请求事件

在代码托管Repo中开启流水线门禁以支持代码自动合入

前往设置

不再提示 ×

事件 ?

☒新建

☒合并

☒重新打开

☒更新

仅代码更新触发

*分支过滤 ?

包含

master

包含

请选择分支或输入正则表达式

🗑

+ 新增

路径过滤 ?

包含

请输入路径或正则表达式

包含

请输入路径或正则表达式

🗑

+ 新增

- 评论Pull Request：评论Pull Request时触发流水线执行。
- 创建Tag（代码托管）
监听创建Tag事件，可以设置Tag包含/排除范围，当在关联的代码仓库创建Tag且满足Tag包含/排除策略，会自动触发关联流水线执行。

图 6-15 配置创建 Tag

创建Tag

*Tag过滤 ?

包含

*

×

包含

请输入Tag或正则表达式

🗑

+ 新增

- 其他事件（码云）
 - 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。

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。

步骤4 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----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...，选择“编辑”。
- 步骤3 进入流水线“任务编排”页面，切换至“触发设置”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 6-16 配置定时任务

定时任务_1

按照配置定时触发流水线运行。[帮助文档](#)

* 启用定时执行

☒

* 启用定时执行

☒ 表单配置

☐ cron表达式 ?

* 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

流水线触发时间总览

每周一、周二、周三、周四、周五

00:00

执行方案

刷新

全部执行

表 6-25 定时任务

参数项	说明
启用定时执行	选择是否开启定时执行。
配置方式	选择“表单配置”或“cron表达式”。“cron表达式”请参考 cron表达式 填写。
执行日	选择定时执行的执行日期。
时间段	选择定时执行的执行时间段和时区。
间隔时间	设置定时执行的间隔时间，即每间隔多长时间触发一次。
执行方案	选择定时执行的执行方案，也可新建执行方案，新建执行方案可参考 配置执行方案 。

 说明

单击右上角，可删除定时任务；单击右上角，可复制定时任务。

- cron表达式由6个部分组成，每个部分使用空格隔开。表达式中的六个部分从左到右代表的含义如下：秒 分 时 日 月 周。

表 6-26 Cron 参数说明

名称	是否必填	可选数值	可选特殊符号
秒	是	数字0-59。	, - * /
分	是	数字0-59。	, - * /
时	是	数字0-23。	, - * /
日	是	每月中的一天，数字1-31。注意2月没有30日和31日，4月、6月、9月、11月没有31日。	, - * ? / L W
月	是	数字1-12，或字符串。数字与字符串的对应关系请参见表6-28。 假如您想在月参数中填写1月，数字“1”或字符串“JAN”都符合Quartz Cron表达式的约定。	, - * /
周	是	数字1-7，或字符串。数字与字符串的对应关系请参见表6-29，Quartz Cron表达式约定周六为每周的最后一天。 假如您想在周参数中填写周一，数字“2”或字符串“MON”都符合Quartz Cron表达式的约定。	, - * ? / L #

表 6-27 特殊符号含义

特殊符号	说明
-	表示范围。例如在“分”使用“5-20”，表示从5分到20分钟每分钟触发一次。
,	表示列表值。例如：在“分”使用“5,20”，则表示在每小时的第5分和20分触发一次定时任务。
*	表示匹配该域的任意值。假如在“分”使用“*”，即表示每分钟都会触发事件。
/	表示起始时间开始触发，然后每隔固定时间触发一次。例如在“分”使用“5/20”，则表示每小时的第5、25、45分钟分别触发定时任务。
?	只能用在“日（每月中的一天）”和“周”两个参数中。它也匹配参数的任意值，但实际不会。因为“日（每月中的一天）”和“周”会相互影响。例如想在每月的20日触发调度，不管20日是星期几，则只能使用如下写法：13 13 15 20 * ?，其中最后一位只能用“?”，而不能使用“*”，如果使用“*”表示不管星期几都会触发，实际上并不是这样。
L	表示最后，只能出现在“日（每月中的一天）”和“周”参数中。如果在“周”域使用“5L”表示在最后的一个星期四触发。
W	表示有效工作日(周一到周五)，只能出现在“日（每月中的一天）”参数中，系统将在离指定日期的最近的有效工作日触发事件。例如：在“日（每月中的一天）”使用“5W”，如果5日是星期六，则将在最近的工作日：星期五，即4日触发。如果5日是星期天，则在6日(周一)触发；如果5日在星期一到星期五中的一天，则就在5日触发。W的最近寻找不会跨过月份。
#	用于确定每个月第几个星期几，只能出现在“日（每月中的一天）”域。例如在4#2，表示某月的第二个星期三。
LW	这两个字符可以连用，表示在某个月最后一个工作日，即最后一个星期五。

表 6-28 月参数的数字与字符串对应关系

数字	字符串
1	JAN
2	FEB
3	MAR
4	APR

数字	字符串
5	MAY
6	JUN
7	JUL
8	AUG
9	SEP
10	OCT
11	NOV
12	DEC

表 6-29 每周数字与字符串对应表

数字	字符串	含义
1	SUN	周日
2	MON	周一
3	TUE	周二
4	WED	周三
5	THU	周四
6	FRI	周五
7	SAT	周六

 说明

最多可以设置10个定时任务。

配置示例：

```

0 0 12 * * ? //每天中午12点执行一次。
0 * 14 * * ? //在每天下午2点到下午2:59期间的每1分钟执行一次。
0 0/30 9-19 * * ? //上午9点到下午19:59期间每半小时执行一次。
0 0 12 ? * WED //每个星期三中午12点执行一次。
0 0/30 9-19 ? * MON-FRI //周一到周五，早上9点至下午19:59点期间，每半小时执行一次。
0 15 10 ? * MON-FRI //周一到周五每天上午10:15执行一次。
0 0 2 1 * ? //在每月的1日的凌晨2点执行一次。
0 15 10 ? * 6#3 //每月的第三个星期五上午10:15执行一次。
0 15 10 ? * 6L //每月的最后一个星期五上午10:15执行一次。

```

步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----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“触发设置”页面。
- 步骤4 打开“Webhook”开关（默认关闭），参考表6-30配置相关参数。

图 6-17 配置 Webhook 触发



表 6-30 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	- 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： curl --header "Content-Type: application/json" --header 'x-auth-token: XXXX (IAM Token)' --request POST --data "{}" Webhook触发源 - 不启用IAM认证，调用示例如下： curl --header "Content-Type: application/json" --request POST --data "{}" Webhook触发源

- 步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。
- 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“触发设置”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 6-18 配置并发策略

并发策略

本条流水线允许同时处于运行中的流水线最大实例个数。[帮助文档](#)

* 并发个数

* 超过并发后执行策略

☐ 排队等待

☒ 忽略不执行

表 6-31 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，详情请参考 流水线规格特性差异 。
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： - 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。 - 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72小时后会放弃执行。 - 排队记录支持手动取消排队，即放弃执行。 - 排队记录执行时，配置信息以入队时为准。 - 忽略不执行：超过并发数后新的触发忽略不执行。

- 步骤5** 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存
- 结束

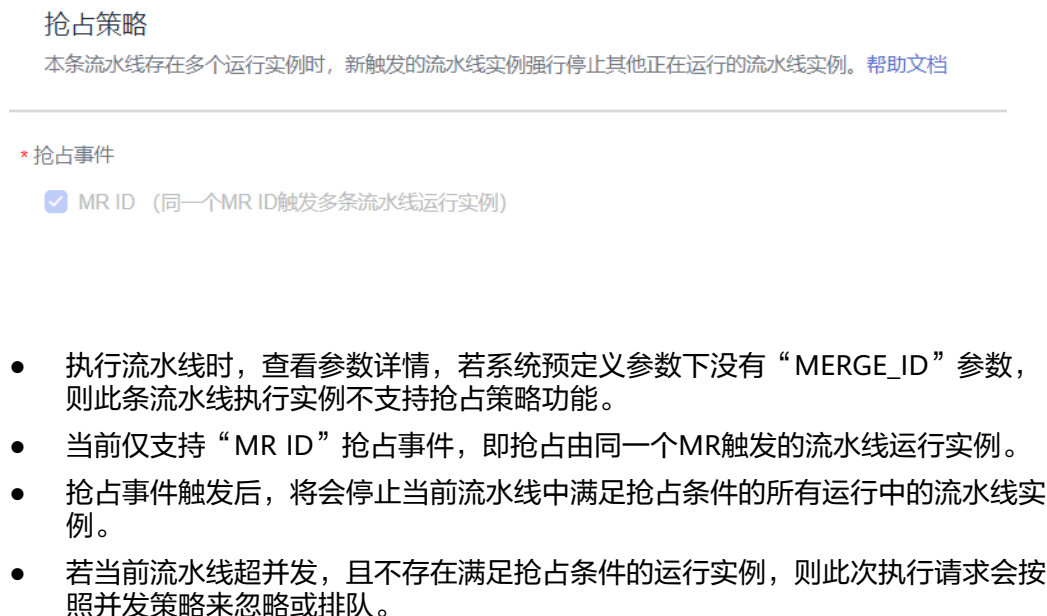
配置流水线抢占策略

同一条流水线存在多个运行实例时，新触发的运行实例会强行停止当前流水线中相同事件触发的其他所有运行实例。

- 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“触发设置”页面。

步骤4 打开“抢占策略”开关，选择抢占事件。

图 6-19 配置抢占策略



步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----结束

6.6 配置流水线权限

什么是流水线权限

流水线权限是指在流水线系统中，对不同用户或角色在流水线相关操作和资源访问上所设置的限制和许可，它是保障流水线安全、稳定运行，以及确保数据和操作的合法性、合规性的重要机制。

本章节介绍如何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，用户可参考本章节进行配置。

约束与限制

- 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。
- 项目管理员权限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。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线权限，权限配置请参考[授权流水线服务](#)。

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...，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。开启“继承项目权限配置”，可直接继承项目的权限配置。
 - 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 6-20 配置角色权限

角色权限配置

人员权限配置

使用项目权限配置 温馨提示

角色	查看	执行	编辑	删除
项目创建者	☒	☒	☒	☒
流水线创建者	☒	☒	☒	☒
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	☒	☒	☐	☐
CI/CD工程师	☒	☒	☒	☐

- 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 6-21 配置人员权限

角色权限配置

人员权限配置

使用项目权限配置 温馨提示

请输入人员、昵称或企业用户进行搜索

人员	所属角色	昵称	企业用户	查看	执行	编辑	删除
	项目经理, 项目创建者, ...			☒	☒	☒	☒

----结束

6.7 配置流水线通知订阅

什么是流水线通知订阅

流水线通知订阅是一种在流水线系统中，让用户或系统组件能够接收与流水线相关事件通知的机制。通过订阅特定的流水线事件，相关人员或系统可以及时了解流水线的

运行状态、任务执行结果等信息。流水线支持配置事件通知，包括站内消息通知、邮件通知、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。

约束与限制

- 默认情况下，服务动态通知除预警外都发送，邮件通知都不发送。
- 关于流水线服务的预警通知功能，所支持的地域（Region）信息可通过控制台查询详情。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线通知订阅，权限配置请参考[授权流水线服务](#)。

配置流水线服务动态通知

通过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、配置被更新以及预警触发时，可以配置给创建人、执行人、收藏人发送服务动态。

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 $\cdots$ ，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“系统消息”下“站内消息通知”，根据需要开启/关闭通知。

- 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 6-22 配置服务动态通知



步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----结束

配置流水线邮件消息通知

通过邮件通知的方式推送通知。流水线执行成功、执行失败、被删除、配置被更新以及预警触发时，可以配置给创建人、执行人、收藏人发送邮件。

- 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“系统消息”下“邮件通知”，根据需要开启/关闭通知。

图 6-23 配置邮件通知



- 步骤5** 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。
- 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 6-24 配置企业微信通知

企业微信

流水线信息推送到企业微信。具体配置指导，可查看[用户指南](#)。

☐ 使用项目通知配置 ① 温馨提示

* URL ②

* 事件类型 ②

☒ 全选

☒ 已完成

☒ 失败

☒ 暂停

☒ 挂起

☒ 忽略

* 通知内容

☒ 全选

☒ 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

☒ 分支

☒ Commit Id

@user_id ②

请输入你要@的成员user_id，多个user_id之间以分隔。

确定

表 6-32 配置企业微信通知

操作项	说明
URL	填写企业微信机器人通知URL，只能以http://或https://开头，例如https://xxxxx。
事件类型	选择要通知的事件，可全选或选择已完成、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，可全选或选择流水线名称、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、分支、Commit Id九种通知内容。
@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，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----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...，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 6-25 配置钉钉通知

 **钉钉**

流水线信息推送到钉钉。具体配置指导，可查看[用户指南](#)。

☐ 使用项目通知配置 ① 温馨提示

* URL ②

* 事件类型 ②

☒ 全选 ☒ 已完成 ☒ 失败 ☒ 暂停 ☒ 挂起 ☒ 忽略

* 通知内容 ②

☒ 全选

☒ 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

☒ 分支

☒ Commit Id

@user_id ②

@手机号 ②

加签密钥 ②

确定

表 6-33 配置钉钉通知

操作项	说明
URL	填写钉钉机器人通知URL，只能以http://或https://开头，例如https://xxxxx。
事件类型	选择要通知的事件，可全选或选择已完成、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，可全选或选择流水线名称、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、分支、Commit Id九种通知内容。
@user_id	填写要接受通知的钉钉的user_id，多个user_id之间以逗号分隔。
@手机号	填写要接受通知的手机号，多个手机号之间以逗号分隔。
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。

步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----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 6-26 配置飞书通知

 **飞书**
流水线信息推送到飞书。具体配置指导，可查看[用户指南](#)。

☐ 使用项目通知配置 ① 温馨提示

* URL ②

* 事件类型 ②

☒ 全选

☒ 已完成

☒ 失败

☒ 暂停

☒ 挂起

☒ 忽略

* 通知内容

☒ 全选

☒ 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

☒ 分支

☒ Commit Id

@user_id ②

加签密钥 ②

确定

表 6-34 配置飞书通知

操作项	说明
URL	填写飞书机器人通知URL，只能以http://或https://开头，例如https://xxxxx。
事件类型	选择要通知的事件，可全选或选择已完成、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，可全选或选择流水线名称、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、分支、Commit Id九种通知内容。
@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。
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。

- 步骤5 配置完成后，单击“保存”，弹出“保存信息成功”提示，完成流水线的保存。

----结束

7配置流水线（YAML 化）

7.1 配置基本信息

- 步骤1** [新建流水线（远端模式）](#)。
- 步骤2** 默认进入“任务编排”页面，单击“基本信息”，跳转至“基本信息”页面。
- 步骤3** 可根据需要修改基本信息中的名称、委托URN、标签和描述等。参数详情请参见[表 7-1](#)。

表 7-1 基本信息

参数项	说明
名称	流水线的名称，支持自定义名称。支持中文、大小写英文字母、数字、“-”、“_”，长度不超过128个字符。
委托URN	IAM的委托唯一标识。未配置时，使用默认委托。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务。
所属项目	流水线归属的项目。不可更改。
流水线保存方式	创建并保存流水线的方式。YAML流水线采用远端模式，此处不可更改。
远端仓库类型	流水线关联的源类型。在创建时选定，此处不可更改。
代码仓	流水线关联的代码仓库，在创建时选定，此处不可更改。
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。在创建时选定，此处不可更改。
YAML文件	创建YAML流水线时，新建或导入的YAML文件，此处不可更改。
标签	为流水线设置标签，详情可参考 查看流水线 。
描述	非必填。对流水线的描述，长度不超过1024个字符。

步骤4 填写完基本信息后，单击“保存”，完成基本信息的保存。也可单击其他页签，进行参数配置。

----结束

7.2 配置流水线源

通过使用sources字段，可以指定流水线的源。在同一条流水线中，可以配置多个源。详情请参考[表7-2](#)。

表 7-2 流水线源类型

源类型	仓库类型	配置属性
代码源	codehub	codehub_id: 代码仓库id。 repo_project_id: 代码源为“其他项目Repo”时，所填写的项目id。
	github	endpoint_id: 扩展点id，获取方式请参考 管理服务扩展点 。
	general_git: 通用git。	
	gitlab	
	gitee: 码云。	
	gerrit	
	tfs_git: TFS。	
	custom_gitlab: 自建GitLab。	
	gitcode	repo_id: 代码仓唯一标识id。
	bitbucket	endpoint_id: 扩展点id。 workspace_id: bitbucket上的项目空间id。
产物源	artifact_generic: 软件发布库。	version_type: 产物源默认版本策略。 specific: 指定版本；latest: 最新版本。 version: 当version_type配置为specific时，需配置版本号。 name: 产物包名。 directory: 产物包所在目录。

源类型	仓库类型	配置属性
	artifact_docker: Docker 仓库。	version_type: 产物源默认版本策略。 specific: 指定版本; latest: 最新版本。 version: 当version_type配置为specific时, 需配置版本号。 org: 镜像所属组织。 image: 镜像地址。
	artifact_swr: SWR源。	version: 版本号 organization: 所属组织。 name: 产物名称。 path: 产物路径。

YAML 文件示例

```
sources:
  phonix_sample:
    type: codehub
    name: phineoxsample
    branch: master
    git_url: https://test/test/phineoxsample.git
    ssh_git_url: git@test:test/phineoxsample.git
    codehub_id: 123
    #界面上别名
    #仓库/制品类型
    #代码源名称
    #默认分支
    #仓库http地址
    #仓库ssh地址
  test_bit:
    type: bitbucket
    name: test_bit
    branch: master
    git_url: https://test/test/test_bit.git
    ssh_git_url: git@test:test/test_bit.git
    endpoint_id: test
    workspace_id: xxx
    #界面上别名,源唯一标识
    #git仓库类型
    #代码源名称
    #默认分支
    #仓库http地址
    #仓库ssh地址
    #扩展点id
  artigen_source1:
    type: artifact_generic
    version_type: specific
    version: 1.0.0
    directory: xxx
    name: test
    #别名, 源唯一标识
    #产物类型
    #产物源默认版本策略
    #当version_type为specific时, 需配置版本号
    #产物包所在目录
    #产物包名
  swr_javadocer:
    type: artifact_docker
    org: dev_x
    image: java
    version_type: specific
    version: test
    #别名, 源唯一标识
    #产物类型
    #组织名
    #镜像名
    #产物源默认版本策略
    #当version_type为specific时, 需配置版本号
  _devcloud_devcloud_demo:
    type: artifact_swr
    organization: org
    name: demo
    version: v1.1
    path: swr.cn-north-7.example.com/org/demo:v1.1
    #别名, 源唯一标识
    #组织
    #镜像名
    #指定版本
```

表 7-3 参数解释

参数名	是否必填	说明
phonix_sample	是	流水线源的唯一标识，仅支持大小写英文字母、数字、下划线，长度不超过128个字符。
type	是	仓库类型。当前支持的仓库类型有codehub、github、general_git、gitlab、gitee、gerrit、tfs_git、gitcode、custom_gitlab、bitbucket。
name	是	代码源名称。
branch	是	默认分支，若启动时未指定分支，则使用默认分支执行。
git_url	是	仓库http地址，例如https://test/test/phineoxsample.git。
ssh_git_url	是	仓库ssh地址，例如git@test:test/phineoxsample.git。
codehub_id	否	仓库id。
endpoint_id	否	扩展点id，非codehub源时必填，需提供扩展点，用以访问配置的仓库。
workspace_id	否	仓库类型为bitbucket时，必填，bitbucket上的项目空间id。
version_type	是	产物类型为artifact_generic时，产物源默认的版本策略。 • specific：指定版本。 • latest：最新版本。
version	是	产物类型为artifact_generic，version_type配置为specific时，需配置版本号。
directory	是	产物类型为artifact_generic。version_type配置为specific时，产物包所在目录。
name	是	产物类型为artifact_generic，version_type配置为specific时，产物包名。
org	是	产物类型为artifact_docker时，镜像所属组织名。
image	是	产物类型为artifact_docker时，镜像名。
version_type	是	产物类型为artifact_docker时，产物源默认的版本策略。 • specific：指定版本。 • latest：最新版本。
version	是	产物类型为artifact_docker，version_type配置为specific时，需配置版本号。
organization	是	产物类型为artifact_swr时，选择的SWR组织。

参数名	是否必填	说明
name	是	产物类型为artifact_swr时，SWR组织下的镜像。
version	是	产物类型为artifact_swr时，镜像的版本。
path	是	产物类型为artifact_swr时，镜像的地址。

7.3 流水线任务编排

配置YAML流水线，需在yaml文件中配置“stage-job-step”结构。

YAML 文件示例

```
stages:                                #流水线阶段务
test2:                                #阶段的唯一标识
  name: 阶段_1                        #阶段的名称
  select: always                      #阶段是否总是运行
  jobs:                               #流水线任务
    job_1:                            #任务的唯一标识
      name: 任务名称                  #任务名称
      if: ${ default() }              #流水线阶段任务执行时需要满足的前提条件，默认值为${ default() }
      runs-on: default
      select: selected_by_default     #任务选择的执行方式，默认为selected_by_default
      needs: [ ]                     #依赖任务的唯一标识，依赖的任务需存在
      steps:                          #定义组成任务的步骤
        - name: step1                #步骤的名称
          uses: official_devcloud_cloudBuild #插件信息，格式为插件名@插件版本。
          env:                         #插件中的环境变量
            test: test
            test2: test2
          with:                        #插件中输入的参数
            key1: value1
            key2: value2
    job_2:
      name: 任务名称2
      if: ${ default() }
      runs-on: [ self-hosted, test2 ] #流水线任务支持的运行环境
      select: selected_by_default
      needs: [ job_1 ]
    job_3:
      name: 任务名称2
      if: ${ default() }
      runs-on: [ dedicated, arm_linux ]
      select: selected_by_default
      needs: [ job_2 ]
  pre:
    type: auto                        #阶段的准入类型
  post:                              #配置阶段的准出任务
    - name: step1
      type: gate
      with:
        name: 创建策略-20250331212921
        level: tenant
    - name: step1
      type: gate
      with:
        name: xxw-chqtest-test
        level: project
post:                                 #后处理阶段的唯一标识
  name: 阶段名称                      #后处理阶段的名称
  select: always                      #后处理阶段是否总是运行，默认值为always
```

```
jobs:
  job1:
    name: 任务名称
    if: ${ default() }
    select: selected_by_default
    needs: [ ]
    steps:
      - name: step1
        uses: official_devcloud_cloudBuild
        env:
          test: test
          test2: test2
        with:
          key1: value1
          key2: value2
```

编排流水线阶段

YAML流水线包含两种阶段：常规阶段和后置阶段。

- **常规阶段：**编排常规阶段时，从编排的顶层结构，由一个stages键开始，stages下最多包含16个阶段

```
stages:
  test2:
    name: 阶段_1
    select: selected_by_default
    jobs:
      pre:
        - type: time_window
          with:
            timeZone: Pacific SA Standard Time
            timeRanges: "[[\"00:00\", \"00:00\"]]"
    post: # 阶段的准出任务，可以配置多个
```

表 7-4 阶段的参数解释

参数名	说明
test2	阶段的唯一标识，仅支持大小写英文字母、数字、中划线或下划线，长度为1-128子字符串。
name	流水线阶段名称。
select	阶段是否总是运行。默认值为selected_by_default。 ● always：阶段总是运行。 ● selected_by_default：阶段默认运行，但不总是运行。
jobs	流水线任务，每个阶段下最多配置100个任务。
type	阶段的准入类型，当前支持三种准入类型。 ● auto：自动准入。 ● manual：手动准入。 ● time_window：时间窗准入，需配置时区和时间段。 - timezone：指定时区。 - timeRanges：指定时间段范围。

参数名	说明
post	配置阶段的准出任务，目前仅支持门禁插件。配置时需指定策略，支持租户策略和项目策略。 若使用租户策略，则需配置level: tenant；若使用项目策略，则需配置level: project。

- **后处理阶段：**编排后处理阶段时，从编排的顶层结构下的post中配置。参数解释请参见表7-4。

```
post: # 后处理阶段
  name: 阶段名称
  select: selected_by_default
  jobs:
  pre:
  - type: auto
  post: # 阶段的准出任务，可以配置多个
```

表 7-5 阶段的参数解释

参数名	说明
name	流水线后处理阶段名称。仅支持输入中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、顿号、斜杠、英文括号、中文括号及空格，其中空格不可在名称开头或结尾使用，长度为1-128个字符。
select	阶段是否总是运行。默认值为selected_by_default。 ● always：阶段总是运行。 ● selected_by_default：阶段默认运行，但不总是运行。
jobs	流水线任务，每个阶段下最多配置100个任务。
type	阶段的准入类型，当前支持三种准入类型。 ● auto：自动准入。 ● manual：手动准入。 ● time_window：时间窗准入，需配置时区和时间段。 - timezone：指定时区。 - timeRanges：指定时间段范围。
post	配置阶段的准出任务，目前仅支持门禁插件。配置时需指定策略，支持租户策略和项目策略。 若使用租户策略，则需配置level: tenant，若使用项目策略，则需配置level: project。

配置流水线任务

```
jobs:
  job_1:
    name: 任务名称
    if: ${ default() }
```

```
runs-on: default
select: selected_by_default
needs: [ ]
steps:
  - name: step1
    uses: official_devcloud_cloudBuild
    env:
      test: test
      test2: test2
    with:
      key1: value1
      key2: value2
```

表 7-6 任务的参数解释

参数名	说明
job_1	任务的唯一标识，仅支持大小写英文字母、数字、中划线或下划线，长度为1-128个字符。
name	任务名称，仅支持输入中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、顿号、斜杠、英文括号、中文括号及空格，其中空格不可在名称开头或结尾使用，长度为1-128个字符。
if	流水线任务执行时需要满足的前提条件，需为可解析的表达式，不超过1024个字符。默认值为\${ default() }。
runs-on	流水线任务支持的三种类型的运行环境，包括默认资源池、专属资源池和自定义资源池。 - 官方资源池：default。 - 自定义资源池：配置一个数组。 runs-on: - self-hosted - 00142c2fb7db4701b074d52910f703d5 - self-hosted：资源池的类型。 - 00142c2fb7db4701b074d52910f703d5：自定义资源池ID。 - 专属资源池：配置一个数组。 runs-on: - dedicated - 2132b0af0a014aa1bcc9e82f2c239c77 - dedicated：资源池的类型。 - 2132b0af0a014aa1bcc9e82f2c239c77：专属资源池ID。
select	任务选择的执行方式，包含以下四种方式。 - always：任务必须执行。 - never：任务禁止执行。 - selected_by_default：任务默认执行，但不总是执行。 - not_selected_by_default：任务默认不被执行。
needs	依赖任务的唯一标识。依赖任务需存在。

配置流水线步骤

```
steps:
- name: step1
  uses: official_devcloud_cloudBuild
  env:
    test: test
    test2: test2
  with:
    key1: value1
    key2: value2
```

表 7-7 步骤的参数解释

参数名	说明
steps	定义任务组成的步骤，单个任务下最多可配置16个步骤。
name	步骤的名称，名称仅支持输入中文、大小写英文字母、数字、中划线、下划线、英文逗号、英文分号、英文冒号、英文句号、顿号、斜杠、英文括号、中文括号及空格，其中空格不可在名称开头或结尾使用，长度为1-128个字符。
uses	插件信息，格式为“插件名@插件版本”。插件名仅支持输入大小写英文字母、数字、中划线或下划线，长度不超过50个字符。插件版本的格式为a.b.c，三段都是0-99的数字。 官方插件仅体现插件名。
env	插件中的环境变量，最多可配置20个参数。 参数名仅支持大小写英文字母、数字或下划线，不超过128个字符。 参数值不超过8192字符
with	插件中输入的参数，最多可配置50个参数。

流水线任务 jobs

- **jobs.<job_id>.runs-on**
使用jobs.<job_id>.runs-on定义任务运行的资源池，支持三种类型：默认资源池（codearts-hosted）、自定义资源池（self-hosted）、专属资源池（dedicated-hosted）。如果需要进一步在资源池节点上自定义容器运行时，可配合container字段使用。
 - a. 默认资源池
默认资源池为codearts官方提供的资源类型，您可以通过以下方式快速定义。
 - i. 通过官方默认标签定义。可以通过简写default使用默认资源池。

```
jobs:
  job_name:
    runs-on: default
```
 - ii. 选择指定类型标签定义。默认资源池建议使用三段式格式定义：[{os}-{arch}-{flavor}]。参数解释请参见表7-8。

表 7-8 字段解释

字段	类型	必填	描述
os	String	是	由两段组成{os}-{version}。version代表os所支持的版本，具体信息可参见表7-9。
arch	String	是	架构类型。支持选择“x64”或“arm64”。
flavor	String	是	runner规格。详细规格信息请参见表7-10。

CodeArts默认资源池规格

当前默认资源池os支持的版本列表如表6 os支持的版本列表。支持的系统规格如表7-10。

表 7-9 os 支持的版本列表

OS	支持版本
ubuntu	latest
euler	latest

表 7-10 runner 规格定义

系统(OS)	规格	处理器（CPU）	内存（RAM）	存储
ubuntu/ euler	slim	1	4GB	20GB
ubuntu/ euler	small	2	8GB	50GB
ubuntu/ euler	medium	4	16GB	100GB
ubuntu/ euler	large	8	32GB	200GB
ubuntu/ euler	xlarge	16	64GB	500GB

```
jobs:
  job_name:
    runs-on: [euler-latest,x64,small]
    steps:
      - run: echo Hello world!
```

数组中可以混合使用字符串和变量，例如：

```
jobs:
  job_name:
    runs-on: ["${ inputs.chosen-os }","${ inputs.arch }",small]
    steps:
      - run: echo Hello world!
```



使用变量必须使用""包裹，直接定义字符串不需要使用""包裹。

b. 自定义资源池（ self-hosted ）

自定义资源池通过group和labels对已定义的执行资源进行过滤。使用三段式格式定义：[{type}-{group}-{label1}-{lable2}....]。

```
jobs:
  job_name:
    runs-on: [self-hosted,my-private-pool,x86,region=cn-north-4
```

表 7-11 参数解释

参数	类型	必填	描述
type	String	是	定义资源池的类型。 • self-hosted：自定义资源池。 • dedicated-hosted：专属资源池。 示例： self-hosted。
group	String	是	资源池名称。系统根据此名称自动路由到对应的资源池。 示例： my-private-pool。
labels	多个String	否	环境标签过滤器。用于在指定资源池内进一步筛选具备特定属性的节点。支持配置一个或多个标签。 示例： 1. 简单标签（ label1 ）： ["x86"] 。 2. 键值对标签 （ label2 ）： ["region=cn-north-4"]。

c. 专属资源池（ dedicated-hosted ）

定义同自定义资源池，通过group和labels对已定义的执行资源进行过滤。使用三段式格式定义：[{type}-{group}-{label1}-{lable2}....]。

```
jobs:
  job_name:
    runs-on: [dedicated-hosted,my-dedicated-pool,x86,region=cn-north-4]
```

表 7-12 参数解释

参数	类型	必填	描述
type	String	是	定义资源池的类型。 - self-hosted：自定义资源池。 - dedicated-hosted：专属资源池。 示例：dedicated-hosted
group	String	是	资源池名称。系统根据此名称自动路由到对应的资源池。 示例：my-dedicated-pool。
labels	多个String	否	环境标签过滤器。用于在指定资源池内进一步筛选具备特定属性的节点。支持配置一个或多个标签。 示例： 1. 简单标签（label1）： ["x86"]。 2. 键值对标签（label2）： ["region=cn-north-4"]。

- jobs.<job_id>.container**
镜像配置定义。使用jobs.<job_id>.container在宿主节点上拉取并启动指定的Docker容器作为任务的运行时环境。

表 7-13 规格定义

属性	说明
类型	object。
必填	否。
默认值	无。
约束	- 仅在自定义资源池或专属资源池下生效。 - 必须安装Docker。 - 默认资源池不支持。

-  **注意**
- 此配置仅在自定义资源池 (self-hosted) 或专属资源池 (dedicated-hosted) 场景下生效，并且必须安装 Docker。
 - 默认资源池或CCE接入模式不支持配置此字段（系统将忽略或报错）。

jobs.<job_id>.container用于创建容器，任务下的全量步骤在该容器环境下执行。如果某些步骤同时使用了脚本操作和容器操作，则这些容器操作将作为同级容器在同一网络上运行，并具有相同的卷挂载点。如果不设置，则所有步骤都将直接在runs-on指定的执行机上运行。

- 在容器内运行任务

```
name: CI
on:
  push:
    branches: [ main ]
jobs:
  container-test-job:
    runs-on: ubuntu-latest
    container:
      image: node:18
      env:
        NODE_ENV: development
      ports:
        - 80
      volumes:
        - my_docker_volume:/volume_mount
      options: --cpus 1
      steps:
        - name: Check for dockerev file
          run: (ls /.dockerev && echo Found dockerev) || (echo No dockerev)
```
- 只指定容器镜像。如果只指定容器镜像，则可以省略image关键字。

```
jobs:
  container-test-job:
    runs-on: ubuntu-latest
    container: node:18
```

- a. **jobs.<job_id>.container.image**
使用jobs.<job_id>.container.image定义运行操作时使用的Docker镜像。该值可以是Docker Hub镜像名称或镜像仓库名称。

表 7-14 规格定义

属性	说明
类型	string。
必填	是。
默认值	无。
约束	Docker Hub镜像名称或镜像仓库名称。支持带标签（如node:18）。

- b. **jobs.<job_id>.container.credentials**
如果镜像的容器注册表需要身份验证才能拉取镜像，您可以使用jobs.<job_id>.container.credentials来设置用户名和密码。

表 7-15 规格定义

属性	说明
类型	object。
必填	否。
默认值	无。
约束	包含username和password字段。建议使用secrets引用敏感信息。

■ 为容器注册表定义凭据

```
container:
  image: ghcr.io/owner/image
  credentials:
    username: ${ github.actor }
    password: ${ secrets.github_token }
```

- c. **jobs.<job_id>.container.env**
用于在容器中设置环境变量。

表 7-16 规格定义

属性	说明
类型	object。
必填	否。
默认值	无。
约束	变量名仅支持包含字母、数字、下划线，且不能以数字开头。

- d. **jobs.<job_id>.container.ports**
用于设置容器上要暴露的端口列表。

表 7-17 规格定义

属性	说明
类型	array[number]。
必填	否。
默认值	无。
约束	端口范围为1-65535。

- e. **jobs.<job_id>.container.volumes**

用于设置容器要使用的卷。通过使用卷，可以实现在服务之间或任务中的其他步骤之间共享数据。您可以指定命名Docker卷、匿名Docker卷或主机上的绑定挂载。

表 7-18 规格定义

属性	说明
类型	array[string]。
必填	否。
默认值	无。
约束	格式：<source>:<destinationPath>。 - source：卷名或绝对路径。 - destinationPath：容器内绝对路径。



注意

指定卷需要指定源路径和目标路径。

■ 在容器中挂载卷

```
volumes:  
- my_docker_volume:/volume_mount  
- /data/my_data  
- /source/directory:/destination/directory
```

- f. **jobs.<job_id>.container.options**
用于配置其他Docker容器资源选项。

表 7-19 规格定义

属性	说明
类型	string。
必填	否。
默认值	无。
约束	Docker容器选项，不支持--network和--entrypoint选项。

■ 使用自定义镜像在自定义资源池执行配置示例

```
jobs:  
  job_name:  
    runs-on:  
      group: my-private-pool
```

```
labels: [x86,region=cn-north-4]

container:
  source: swr
  image: my-org/java-build-env:v2.0
  env:
    # 传递给容器的环境变量
    NODE_ENV: production
  ports:
    # 端口映射（较少用，多用于service容器）
    - 8080:80
  volumes:
    # 挂载宿主机的目录到容器
    - my_docker_volume:/volume_mount
  options: --cpus 1
    # 传递给"docker create"额外命令行参数
  credentials:
    # 如果是私有镜像库，需要提供认证信息
    username: ${atomgit.actor}
    password: ${secrets.swr_token}

resources:
  limits:
    # 限制容器最大使用8核16G
    cpu: 8
    memory: 16G
  requests:
    cpu: 4
    memory: 8G

steps:
  - run: mvn clean package
```

7.4 配置流水线参数

YAML流水线参数分为两种类型：自定义参数和通用参数组。私密参数不支持在YAML中配置，需要转换到页面配置中进行设置。

YAML 文件示例

```
inputs:
  #除私密参数之外的自定义参数
  input1:
    #参数名
    type: string
    default: abc
    #参数类型
    manual_override: false
    #默认值
    description: 描述
    #是否运行时设置
  input2:
    type: increment
    default: 1.0.0
    manual_override: false
    is_strict: true
    #是否开启严格递增校验
    description: 描述
  input3:
    type: enum
    default: value1
    manual_override: true
    description: 描述
    values: [ "value1", "value2", "value3" ]
    #枚举值
vars:
  uses:
    #通用参数组的名称
    - test1
    - test2
    - test3
```

自定义参数

自定义参数可通过 "inputs" 进行配置，目前支持字符串、自增长和枚举三种参数类型。yaml文件示例如下。

```
inputs:
  input1:
```

```
type: string
default: abc
manual_override: false
description: 描述
input2:
  type: increment
  default: 1.0.0
  manual_override: false
  is_strict: true
  description: 描述
input3:
  type: enum
  default: value1
  manual_override: true
  description: 描述
  values: [ "value1", "value2", "value3" ]
```

表 7-20 参数解释

参数名	说明
inputs	用于定义自定义参数配置。
input1	自定义参数的名称，仅支持大小写英文字母、数字或下划线，不超过128个字符。
type	定义参数的类型，包含三种类型。 - string：自定义参数为字符串类型。 - increment：自定义参数为自增长类型。 - is_strict：若为true，则执行流水线时会校验该参数是否严格递增。 - latest_value：当前参数的自增长进展。 - enum：自定义参数位枚举类型。values为枚举值列表。
default	参数的默认值。
manual_override	是否运行时设置。若为true，则支持执行流水线时修改该参数的值。 - true：是。 - false：否。
description	对于参数的描述。

通用参数组

```
vars:
  uses: #通用参数组的名称
  - test1
  - test2
  - test3
```

表 7-21 参数解释

参数名	说明
vars	用于定义通用参数组配置。
uses	通用参数组的名称，仅支持大小写字母、数字或下划线，不超过128个字符。

7.5 配置事件触发

YAML流水线事件触发有两种配置方式：一种是[流水线源仅包含一个代码源](#)，另一种是[流水线源包含多个代码源](#)。这两种方式在层级上有所不同，但除此之外没有其他差别。

流水线源仅包含一个代码源

流水线源仅包含一个代码源时，不需要指定具体代码仓，默认关联到唯一的代码源。并且可以直接在on.merge_requests、on.push、on.tag_push中配置事件触发。

```
on:
  merge_requests:                #MR事件触发
    types: [ merge, open, reopen, update ]
    branches: [ 'master' ]
    branches-ignore: [ 'test' ]
    paths: [ 'test' ]
    paths-ignore: [ 'test2' ]
    code-update: true
  push:                          #代码提交事件触发
    branches: [ 'master' ]
    branches-ignore: [ 'test' ]
    paths: [ 'test' ]
    paths-ignore: [ 'test2' ]
    tags: [ 'master' ]           #创建Tag触发
    tags-ignore: [ 'test' ]
```

表 7-22 参数解释

参数名	说明
merge_requests	定义合并请求事件触发配置。
branches	分支包含。branches和branches-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
branches-ignore	分支排除。branches和branches-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
paths	路径包含。
paths-ignore	路径排除。

参数名	说明
types	事件触发类型，共包含以下五种。 • open：新建，合并请求创建时触发。 • merge：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。 • reopen：重新打开，合并请求重新打开时触发。 • update：更新，合并请求内容更新、设置更新或者源代码更新时触发。 - code-update：仅代码更新时触发，types选择update时生效。true表示只有代码变更会触发；false表示任何变更会触发。
push	定义代码提交事件触发。
tags	创建Tag触发。Tag包含。tags和tags-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
tags-ignore	创建Tag触发。Tag排除。tags和tags-ignore的总数不能超过32个，且单个长度不得超过1024个字符。

流水线源包含多个代码源

流水线源包含多个代码源时，需要在on.sources_identifier下配置。identifier为源的唯一标识。

```
on:
  sources:
    phoenix_sample:                #源唯一标识
      merge_requests:              #MR事件触发
        types: [ merge, open, reopen, update ]
        branches: [ 'master' ]
        branches-ignore: [ 'test' ]
        paths-ignore: [ 'test2' ]
      push:                        #代码提交事件触发
        branches: [ 'master' ]
        branches-ignore: [ 'test' ]
        paths-ignore: [ 'test2' ]
      tags: [ 'master' ]           #创建Tag触发
      tags-ignore: [ 'test' ]
```

表 7-23 参数解释

参数名	说明
sources	多代码源场景下事件触发的配置。
phoenix_sample	流水线源唯一标识，和sources下配置的代码源映射，不可配置产物源。仅支持大小写英文字母、数字、下划线，长度不超过128个字符。
merge_requests	定义合并请求事件触发配置。

参数名	说明
branches	分支包含。branches和branches-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
branches-ignore	分支排除。branches和branches-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
paths	路径包含。
paths-ignore	路径排除。
types	事件触发类型，共包含以下五种。 • open：新建，合并请求创建时触发。 • merge：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。 • reopen：重新打开，合并请求重新打开时触发。 • update：更新，合并请求内容更新、设置更新或者源代码更新时触发。 - code-update：仅代码更新时触发，types选择update时生效。true表示只有代码变更会触发；false表示任何变更会触发。
push	定义代码提交事件触发。
tag_push	定义创建Tag触发。
tags	创建Tag触发。Tag包含。tags和tags-ignore的总数不能超过32个，且单个长度不得超过1024个字符。
tags-ignore	创建Tag触发。Tag排除。tags和tags-ignore的总数不能超过32个，且单个长度不得超过1024个字符。

7.6 配置其他触发

并发策略和抢占策略配置示例如下，其功能与图形化页面保持一致。

```
concurrency:      #并发策略
  max: 5          #并发个数
  exceed-action: IGNORE
  enable: true    #是否启用
preemption:      #抢占策略
  enable: true    #是否启用
  events: mr_id   #抢占事件
```

表 7-24 参数解释

参数名	说明
concurrency	定义并发策略触发设置。
max	并发个数，取值范围为1-120。

参数名	说明
exceed-action	执行的流水线超过并发个数后的限制。 - IGNORE：忽略不执行。 - QUEUE：排队等待。
enable	是否启用并发策略。默认为true。 - true：启用。 - false：不启用。
preemption	定义抢占策略触发设置。
enable	是否启用抢占策略。默认为true。 - true：启用。 - false：不启用。
events	抢占事件

7.7 配置执行方案

流水线最多可配置200个执行方案，yaml文件示例如下。

```
policies:                                #执行方案 最多配置200个
test1:
  sources:                              #启动流水线使用的源
    phonix_sample:
      branch: master
      tag: master
    swr_javadoccer:
      version: version1.1
  description: 执行方案1
  inputs:                                #执行的环境变量
    input1: value1
    input2: value2
    input3: value3
  jobs: [ job_1, job_2, job_3 ]         #执行时所选的任务
```

表 7-25 参数解释

参数名	说明
policies	用于定义配置流水线执行方案。
test1	执行方案的名称，仅支持输入中文、大小写英文字母、数字、中划线或下划线，长度不超过128个字符。
sources	执行时流水线时，流水线源的取值。 若为代码源，则需要额外配置分支或标签名；若为产物源，则需要配置版本号
phonix_sample	流水线源的唯一标识，需要和sources下配置的代码源映射。
swr_javadoccer	流水线源的唯一标识，需要和sources下配置的代码源映射。

参数名	说明
branch	分支，不超过1024个字符。branch与tag仅需选择其中一个配置。
tag	tag，不超过1024个字符。branch与tag仅需选择其中一个配置。
description	对执行方案的描述信息。
inputs	新建执行方案时，运行时参数的取值。
jobs	新建执行方案时，所选择执行的任务。

7.8 配置定时任务

定时任务目前仅支持周期性触发模式，在on.schedules下进行配置。yaml文件示例如下。

```
on:
  schedules:
    - type: 'periodic'
      name: 定时任务2
      enable: true
      policies: [ 执行方案1 ]
      days: [ 1,2,3,4,5,6,7 ]
      start: 00:00
      end: 00:01
      timezone: Asia/Shanghai
      interval: 900
```

表 7-26 参数解释

参数名	说明
schedules	定义配置定时任务，最多可配置10个定时任务，且名称不可重复。
type	触发定时任务的模式，当前仅支持周期触发模式。可设置为periodic。
name	定时任务的名称，仅支持输入中文、大小写英文字母、数字、中划线或下划线，长度为1-128个字符。
enable	是否启用定时任务。默认为true。 • true：是。 • false：否。
policies	定义定时任务的执行方案，执行方案名称长度为1-128个字符。若不配置，将根据流水线的默认配置执行。
days	配置定时任务的周期，不可为空。长度为1-7，取值范围为1-7。 1：周日；2：周一；3：周二；4：周三；5：周四；6：周五；7：周六。

参数名	说明
start	定时任务开始时间，格式为HH:mm。
end	定时任务结束时间，格式为HH:mm。结束时间必须大于开始时间。
timezone	定时任务的时区，包含以下取值。 • UTC。 • America/Sao_Paulo。 • America/Santiago。 • America/Chicago。 • Africa/Cairo。 • Europe/Moscow。 • Asia/Bangkok。 • Asia/Shanghai。 • Asia/Singapore。
interval	触发定时任务的时间间隔。单位为秒，设置时不得少于900秒。在定时任务开始到结束的时间区间内，每隔interval将触发一次流水线。

8 分组管理流水线

什么是分组管理

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

- 仅项目管理员和项目经理支持在“权限管理”页签下，批量修改流水线权限。
- 批量审核功能仅支持审核配置了人工审核任务的流水线。
- 一个项目下，最多可创建1000个分组。
- 批量删除流水线时，一次最多可删除20条。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置流水线分组管理，权限配置请参考[授权流水线服务](#)。

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 将鼠标移动到“全部分组”所在行，单击，新增分组。



步骤4 将鼠标移至新建的分组上，单击 设置分组名称，设置完成后，单击空白处，完成分组名称的保存。



步骤5 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，可上移、下移和删除分组。



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 在流水线页面，选择需要操作的流水线，可进行以下批量操作。

图 8-1 批量操作流水线



- 单击“移动到”，弹出“移动分组”弹窗，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”弹窗，单击“确定”即可批量执行流水线。
- 单击“审核”，弹出“批量审批”弹窗，可批量审批流水线。
 - 单击“通过”，弹出“批量通过审批”弹窗，确认通过后单击“确定”，弹出“批量审核结果”弹窗，可查看审核结果。
 - 单击“驳回”，弹出“批量驳回审批”弹窗，确认驳回后单击“确定”，弹出“批量审核结果”弹窗，可查看审核结果。
- 单击“权限管理”，弹出“权限管理”弹窗，项目管理员和项目经理可批量修改流水线权限。
 - 开启“继承项目权限配置”，可直接继承项目的权限配置。
 - 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。
- 单击“禁用”，弹出“批量禁用”弹窗，可批量禁用流水线。
- 单击“取消禁用”，弹出“批量取消禁用”弹窗，可批量取消禁用流水线。

- 单击“更多 > 设置委托”，弹出“批量设置委托”弹窗，可批量为流水线设置委托。
- 单击“更多 > 通知订阅”，弹出“批量设置通知订阅”弹窗，可批量为流水线设置通知订阅。配置通知订阅请参考[配置流水线通知订阅](#)。
- 单击“更多 > 设置标签”，弹出“设置标签”弹窗，可批量为流水线设置标签。
- 单击“更多 > 删除”，弹出“批量删除”弹窗，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

9 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。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以执行流水线，权限配置请参考[授权流水线服务](#)。

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”弹窗，选择“自定义执行”，配置如下信息：
 - 代码源：选择代码源的分支、Tag或CommitId。
 - 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：流水线执行阶段支持只执行部分任务，可以选择一个或多个任务执行，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”，进入流水线详情页面，可实时查看流水线执行进展及任务执行状态。

图 9-1 执行流水线



- 单击“重试”，可在失败任务处重新执行。
- 单击失败任务卡片上的，可在此失败任务处重新执行。
- 单击“重新执行”，可按照此次配置重新执行。
- 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，可参考[流水线规格差异特性说明](#)。

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

10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、项目创建者、流水线创建者或系统工程师才可以删除流水线，可通过配置权限给不同角色添加不同操作权限。
- 新建标签时，标签名称最多可设置50个字符。
- 单租户最多可创建2000个标签。
- 查看流水线执行记录时，首次执行时执行历史页面为空，执行一次后才有记录。
- 运行中的流水线无法禁用或删除。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为项目下各成员时，可以查看流水线，权限配置请参考[授权流水线服务](#)。

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 10-1 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。 说明 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。
标签	流水线的标签。

- 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“导入”，通过导入JSON文件创建和更新流水线。
- 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。

表 10-2 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，代码提交时触发，合并请求时触发，创建标签时触发，定时任务触发、Webhook触发、子流水线触发。
执行人	流水线执行人名称。
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）。
执行描述	对流水线执行的具体描述。
任务详情	流水线执行的详细信息。
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。
操作	流水线执行参数详情，单击  可查看。

- 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 10-3 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。 - 只有构建类型的任务才有构建包。 - 存在多个构建包时，单击“全部下载”，可以下载全部构建包。 - 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。
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。 “延时执行”和“挂起流水线”任务没有日志。 - 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。
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、查看执行方案。

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72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

11

查看统计看板

流水线统计看板可视化展示了流水线的各类数据、进度和状态，介绍了流水线总览、执行情况、并发队列情况、项目流水线数量、项目执行情况总览和项目队列情况总览。

前提条件

已[开通并授权使用CodeArts Pipeline](#)。

查看统计看板

- 步骤1 访问[CodeArts Pipeline](#)首页。流水线统计看板页展示了流水线数据的统计情况。
- 步骤2 单击“所有项目”旁边的下拉框，可根据需要选择所有项目或所要查看的项目下流水线数据的统计情况。

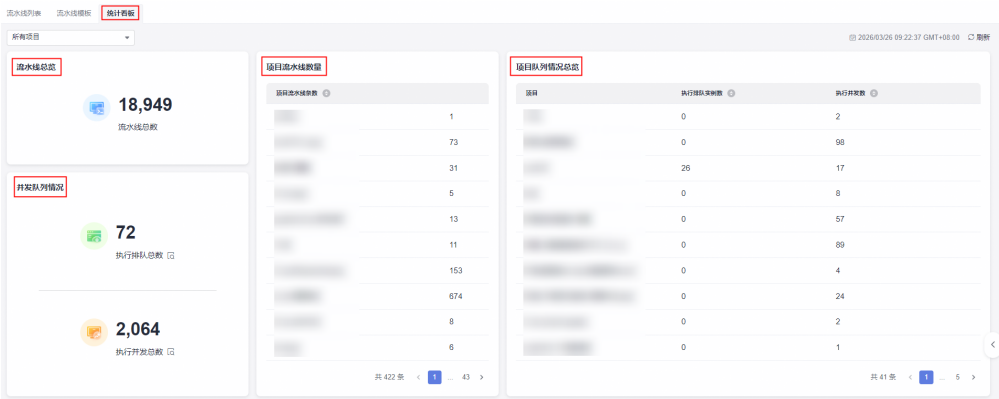


表 11-1 流水线统计看板

参数项	说明
流水线总览	流水线总数。

参数项	说明
并发队列情况	流水线执行排队实例数和执行并发总数。 - 单击执行排队总数右侧，弹出“执行排队实例数”侧滑框，可查看执行排队实例数的流水线和触发信息。 - 单击执行并发总数右侧，弹出“执行并发实例数”侧滑框，可查看执行并发实例数的名称、执行信息、触发方式和执行时间。
项目流水线数量	各项目下的流水线的数量，单击项目名称可进入对应的流水线界面。
项目队列情况总览	各项目下流水线的执行排队实例数和执行并发数。

步骤3 查看指定时间内流水线的执行效率、执行情况、排队和并发动态统计和项目执行情况总览。支持查看最近30分钟、1小时、24小时、7天、30天内流水线的执行数据。



表 11-2 流水线统计看板

参数项	说明
执行效率	查看指定时间内，流水线的“平均执行时长”和“平均排队时长”的数据统计情况。
执行情况	查看指定时间内，流水线执行成功率和执行次数，可查看90天内流水线的成功率和执行次数。
排队和并发动态统计	查看指定时间内，流水线排队实例数和执行并发总数统计。该统计表记录每5分钟时间点的排队实例数与执行并发数。
项目执行情况总览	查看指定时间内，各项目下流水线的执行成功率和执行次数，可查看指定时间内的流水线的信息，单击项目名称可进入对应的流水线界面。

----结束

查看单条流水线统计看板

- 步骤1 访问CodeArts Pipeline首页。
- 步骤2 单击需要查看流水线的名称，进入流水线“执行历史”页面。
- 步骤3 单击“统计看板”，进入“统计看板”页面，查看此流水线的数据统计情况。

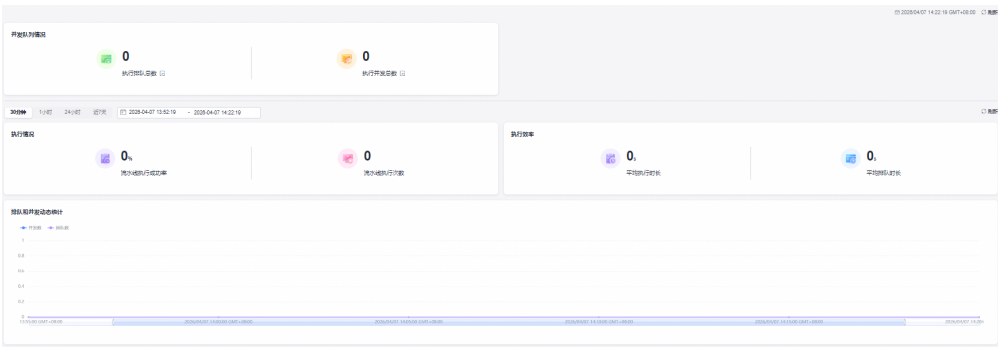


表 11-3 单条流水线统计看板

参数项	说明
并发队列情况	流水线执行排队实例数和执行并发总数。 - 单击执行排队总数右侧，弹出“执行排队实例数”侧滑框，可查看执行排队实例数的流水线和触发信息。 - 单击执行并发总数右侧，弹出“执行并发实例数”侧滑框，可查看执行并发实例数的名称、执行信息、触发方式和执行时间。
执行情况	查看指定时间内，流水线执行成功率和执行次数，可查看90天内流水线的成功率和执行次数。
执行效率	查看指定时间内，流水线的“平均执行时长”和“平均排队时长”的数据统计情况。
排队和并发动态统计	查看指定时间内，流水线排队实例数和执行并发总数统计。

----结束

12 流水线预警设置

流水线预警机制是指在软件开发和部署过程中，为确保流程的稳定性和可靠性而设置的一系列预警措施。该机制能够实时监控流水线各阶段及关键指标，在检测到异常时及时发出警报，并快速响应解决问题。通过实施流水线预警机制，可显著提升软件开发与部署效率，减少故障和中断，提升用户体验。

约束与限制

本页面介绍了流水线服务支持的预警功能，关于此功能支持的地域（Region）信息，可通过控制台查询详情。

配置租户级预警设置

- 步骤1 访问CodeArts Pipeline首页。
- 步骤2 单击“统计看板”，进入“统计看板”页面。单击右上角“预警设置”，弹出“租户级预警设置”侧滑框。
- 步骤3 单击“新建预警”，弹出“新建预警”侧滑框，参考表12-1配置以下信息。

表 12-1 新建预警

参数项	说明
名称	预警的名称，支持自定义设置。
归属项目（默认为租户级（全部项目））	并发饱和度。 - 并发使用百分比：流水线执行并发数达到上限的百分比。 - 严重等级：超过设定值时的严重程度，包含“提示”、“一般”和“严重”，默认为“提示”。
	排队积压告警。 - 流水线排队数：流水线等待执行的数量。 - 严重等级：超过设定值时的严重程度，包含“提示”、“一般”和“严重”，默认为“一般”。

参数项	说明
	连续失败次数。 - 流水线连续失败次数：流水线执行时连续失败的次数。 - 严重等级：超过设定值时的严重程度，包含“提示”、“一般”和“严重”，默认为“严重”。

步骤4 单击，打开并配置参数，配置完成后单击“确定”，弹出“创建成功”提示，预警新建成功。



----结束

查看租户级预警设置

- 步骤1** 访问CodeArts Pipeline首页。
- 步骤2** 单击“统计看板”，进入“统计看板”页面。单击右上角“预警设置”，弹出“租户级预警设置”侧滑框。
- 租户级预警设置页面展示当前租户下的所有预警，列表项说明如下：

表 12-2 租户级预警设置

参数项	说明
名称	预警的名称，支持自定义设置
预警内容	所设置的预警内容，包括并发饱和度、排队积压告警和连续失败次数。
创建时间	创建预警的时间。
操作	- 单击，弹出“编辑预警”弹窗，可修改预警内容。 - 单击，弹出设置本条预警为默认预警的提示框，单击“确定”可设置本条预警为默认预警。 - 单击，弹出“删除预警配置”弹窗。单击“一键输入”或手动输入“DELETE”后，单击“确定”，即可删除本条预警。

----结束

13 配置微服务变更流水线

什么是微服务

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。

约束与限制

- 新建流水线微服务关联流水线源时，可暂不选择流水线源，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。
- 新建流水线微服务关联关联代码仓时，一个代码仓只能关联一个微服务。
- 创建微服务后，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。

前提条件

- 已[开通并授权使用CodeArts Pipeline](#)。
- 当用户角色为“项目管理员”或者“流水线创建者”，可以配置微服务变更流水线，权限配置请参考[授权流水线服务](#)。

新建流水线微服务

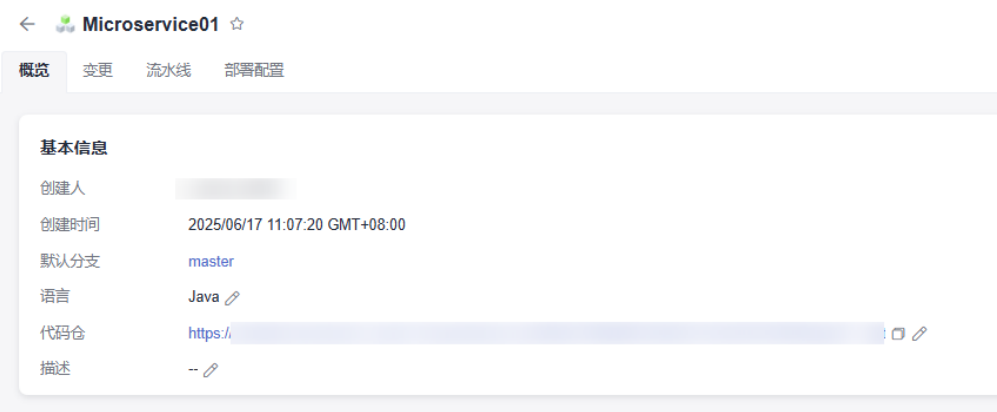
- 步骤1** 访问[CodeArts Pipeline](#)首页，进入项目下流水线。
- 步骤2** 单击“微服务”，进入微服务列表页面。
- 步骤3** 单击“新建微服务”，进入“新建微服务”页面，配置相关参数。

表 13-1 新建微服务

参数项	说明	取值样例
所属项目	微服务所在项目，不可更改。	此处默认为“project01”。
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。	输入“Microservice01”。
流水线源	流水线代码来源，当前仅支持Repo。	选择“Repo”。
代码仓	微服务关联的代码仓库，选择已创建的代码仓库。	选择“Repo01”。
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。 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。	选择“master”。
语言	微服务的开发语言，支持Java、Python、Node.js、Go、.Net、C++、PHP。	选择“Java”。
描述	对微服务的描述，长度不超过1024个字符。	-

- 步骤4** 单击“确定”，弹出“新建微服务成功”提示，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。

图 13-1 微服务概览页



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。

图 13-2 微服务列表



表 13-2 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

新建微服务变更

管理微服务下的变更信息。

准备工作

新建工作项。

操作步骤

- 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。

图 13-3 新建变更

基本信息

* 变更标题

Change01

代码仓

https://

* 代码分支

从默认分支拉取新分支

关联已有分支

master

* 关联工作项

712876189

新建工作项

标题	编号	操作
Bug BUGFIX	712876189	

表 13-3 新建变更

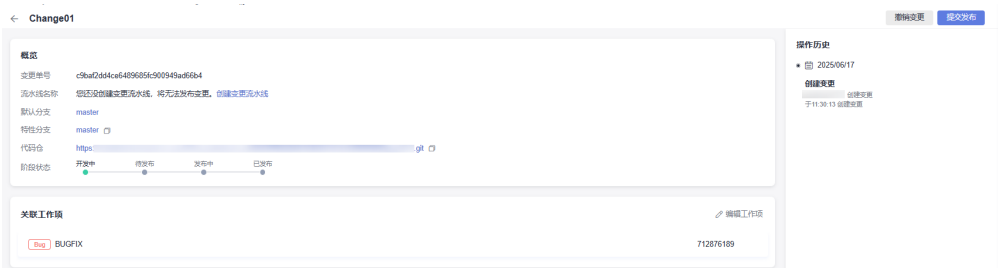
参数项	说明	取值样例
变更标题	输入变更标题。长度不超过256个字符。	输入“Change01”。
代码仓	微服务关联的代码仓库，不可更改。	-
代码分支	本次变更新需求的开发分支，可从默认分支拉取新分支，也可关联其他已有分支。 变更通过流水线发布成功后，会将代码分支自动合并至微服务的默认分支。	选择“关联已有分支”，下拉框选择“master”。

参数项	说明	取值样例
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。	选择 准备工作中 中已创建的工作项“BUGFIX”。

步骤6 单击“确定”，弹出“创建成功”提示，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 13-4 变更详情



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”。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

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 13-4 新建流水线

参数项	说明	取值样例
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。	输入“Pipeline-1”。
委托URN	IAM的委托唯一标识。配置后，执行流水线中任务时使用委托身份和权限访问其他云服务，否则使用运行人的身份和权限。	-
所属项目	微服务所在的项目。	此处默认为“project01”。
流水线源	流水线代码来源，微服务下流水线仅支持Repo。	选择“Repo”。
代码仓	微服务关联的代码仓库。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。	选择“Repo01”。
默认分支	微服务关联的默认分支。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。	选择“master”。
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，配置方法请参考新建CodeArts服务扩展点。 - 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。具体问题请参考常见问题。	新建授权，示例如下。 
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。 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。	-

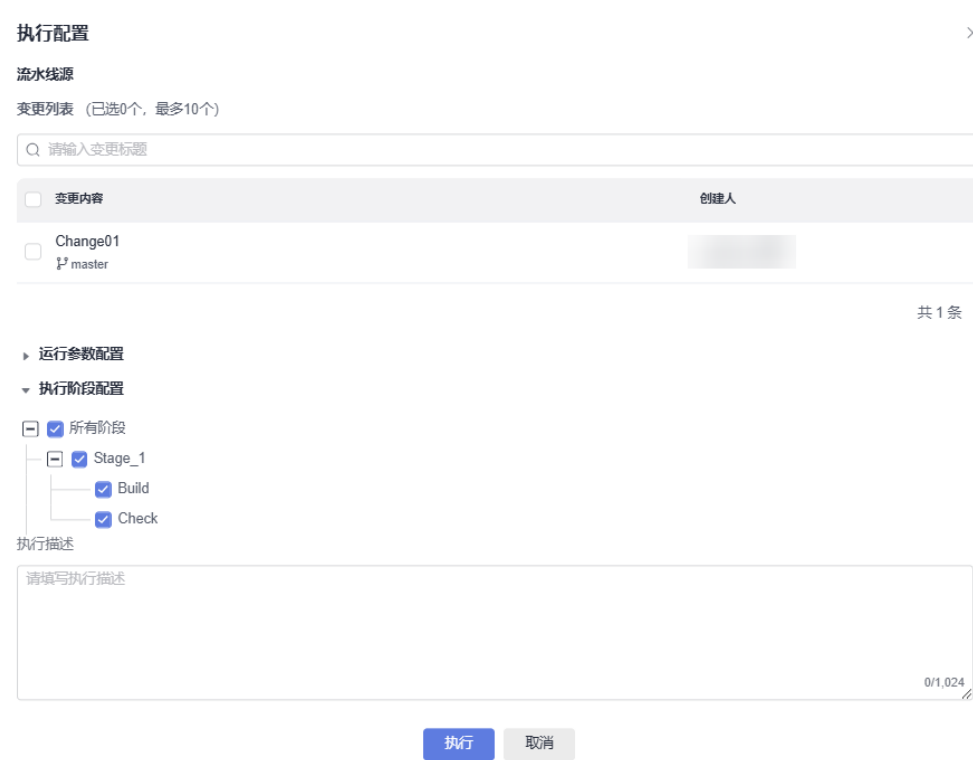
参数项	说明	取值样例
基于变更触发	开启“基于变更触发”开关的流水线即为变更流水线，在微服务流水线列表中，变更流水线会被标识为  。单个微服务下仅允许存在一条基于变更触发的流水线。	单击  开启“基于变更触发”。
描述	对变更流水线的描述，长度不超过1024个字符。	-

- 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”，进入“任务编排”页面，根据需要编排流水线。
- 步骤8** 单击“保存”，弹出“保存信息成功”提示，完成变更流水线的保存。
- 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 13-5 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 13-6 执行变更流水线



变更流水线运行时会默认添加“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，执行过程中遇到的问题请参考[常见问题](#)。

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”。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才可以退出发布。

----结束

14 管理流水线扩展插件

14.1 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“更多步骤”，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。

14.2 流水线官方插件

流水线当前内置的官方插件如[表14-1](#)。

表 14-1 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。 了解更多 。
	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。 了解更多 。
	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。 了解更多 。
	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）。
	开发者联调	开发者联调是基于kubernetes自主研发的云原生研发环境交付平台。提供面向软件开发者的研发环境持续交付、组件联调测试等能力，提供面向企业管理的降低研发环境资源成本解决方案。
	部署执行器	按照微服务环境模型中配置的集群策略执行部署。系统内部调度使用，不支持手动配置。

插件类型	插件名	插件使用介绍
	滚动升级	基于原生Kubernetes实现的滚动升级部署插件，可以实现滚动升级动作；当前仅支持在部署服务内使用，通过与部署服务其他步骤结合，可以实现灵活的自定义发布流程编排。
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。 了解更多 。
	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。
	鲲鹏代码迁移	鲲鹏代码迁移工具是一款可以简化客户应用迁移到基于鲲鹏916/920的服务器的过程的工具。工具可扫描并自动分析待迁移软件，提供可迁移性评估报告；也可对待迁移软件进行源码分析，准确定位需迁移的代码，并给出迁移指导或一键代码替换。
	标准策略准出条件	选择一个标准插件策略用于门禁拦截。
通用	创建仓库Tag	为代码仓创建Tag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。

插件类型	插件名	插件使用介绍
	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	将文件上传至华为云OBS。
	OBS文件下载	从OBS下载文件至本地。
	下载制品产物	用于下载制品产物。
	使用jq命令	对\${SHARE_PATH}下的json文件使用jq命令。
	解压zip文件	用于解压位于\${SHARE_PATH}路径下的zip文件。
	压缩成zip文件	将位于\${SHARE_PATH}路径下的单个文件或文件夹压缩成zip格式。
	鸿蒙应用发布	发布鸿蒙应用app到AGC。
	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。
	执行docker操作	执行指定docker操作，仅支持运行在自定义执行机中。

14.3 图形化新建流水线自定义插件

图形化新建自定义插件通过可视化界面帮助用户创建、配置和管理插件。

约束与限制

- 插件版本号，支持填写3组0-99的数字。
- “版本信息”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。

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“搭建流水线插件”，开始注册插件。

步骤4 填写基本信息，相关参数说明如表14-2所示。

表 14-2 自定义插件基本信息

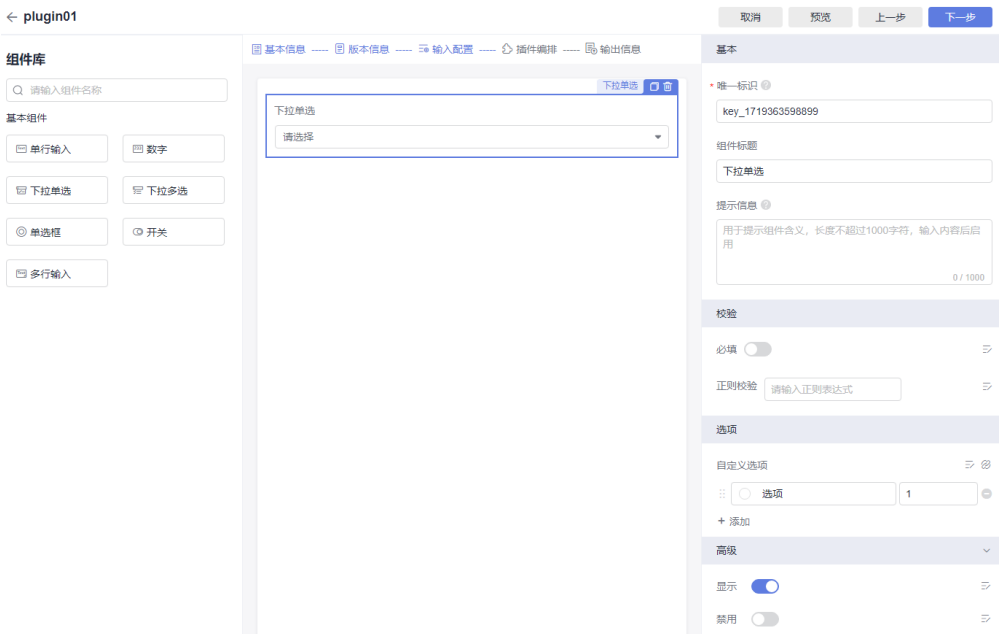
参数项	说明
插件图标	插件封面图，不上传将会自动生成系统图标。支持png、jpeg、jpg格式，文件大小不超过512KB，建议128*128px。
名称	插件在插件市场和流水线显示的名称。支持空格、中文、大小写英文字母、数字、“-”、“_”、“.”，长度不超过50字符。
唯一标识	插件的唯一标识，设置后不可修改，建议设置成具有实际含义的内容。支持大小写英文字母、数字、“-”、“_”，长度不超过50字符。
插件类型	插件的类型，支持构建、代码检查、测试、部署、通用共5种类型，设置后不可修改。
插件描述	插件描述将展示在插件市场中，介绍插件主要用途和功能，设置后可修改。长度不超过1000字符。

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。

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 14-1 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 14-3 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。
	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。

类别	参数名	说明	适用控件
选项	自定义选项	设置控件的选项范围。单击  添加，新增选项，单击 ，删除选项。 - 选项名称：插件配置时的界面显示内容。 - 选项值：插件运行时实际下发的值。 除手动配置外，还支持通过以下方式设置可选范围： - 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14-2，API接口参数说明请参考表14-4。 - 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。	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 14-2 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

搜索提示语 ?

请输入

确定

取消

表 14-4 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。
URL	仅支持HTTPS协议。

参数项	说明
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”。 <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。 - Params参数：搜索字段的参数类型为API接口请求体的Params参数。 - Body参数：搜索字段的参数类型为API接口请求体的Body参数。
搜索提示语	当前组件下拉框中搜索栏下方的提示语，用于描述搜索功能，长度不超过100字符。

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。**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14-3](#)。
 - artifacts：固定目录，保持不变。
 - demo：对应Build构建插件的“产物标识”，如[图14-4](#)。

图 14-3 Build 构建任务 ID

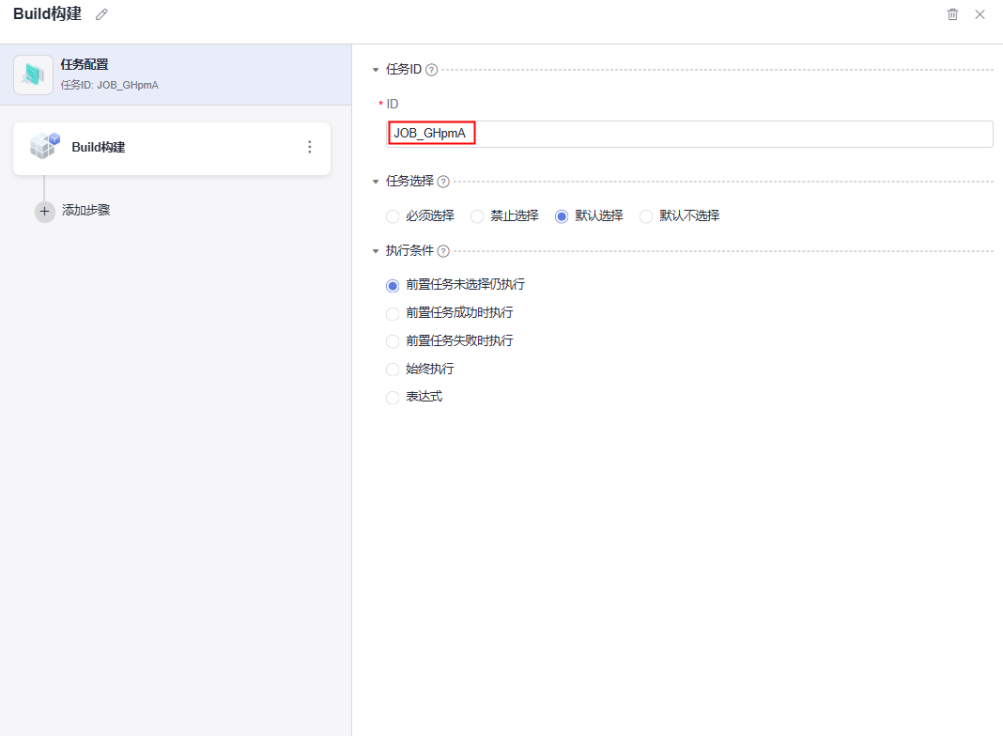
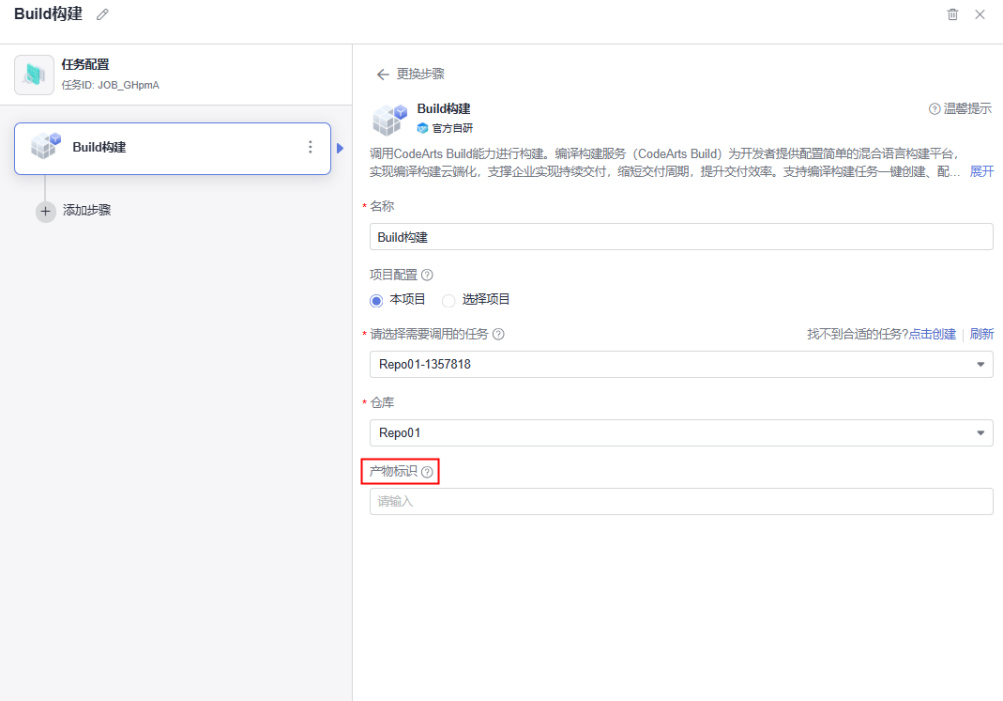


图 14-4 Build 构建产物标识



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 14-5 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。

变量	描述
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。
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 14-6 输出文件说明

输出文件	使用说明
\${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。当前仅“代码检查”类型的插件支持将该结果进行展示。
\${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： • 解析时，服务会忽略未配置在metrics控件中的key值。 • 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。

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。

CustomExtension/RuleExample				
Group				
检查项	状态	当前值	比较	阈值
par1	● 成功	100	=	100
par2	● 成功	200	=	200

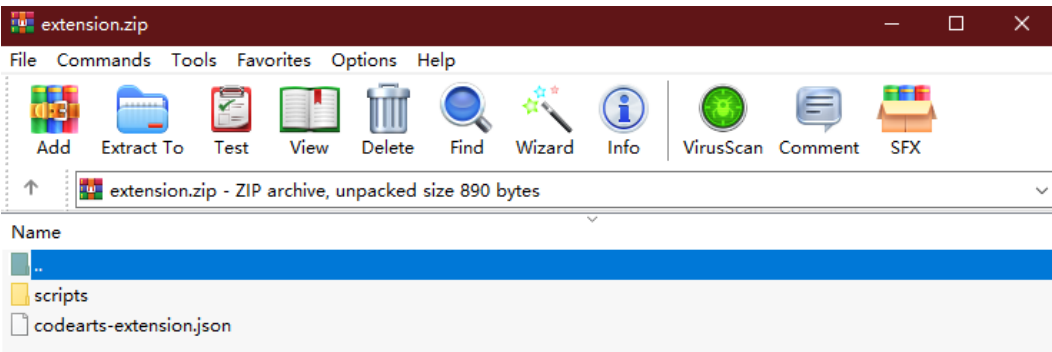
14.4 代码化新建流水线自定义插件

代码化新建流水线自定义插件通过编写代码来创建和配置流水线，代码化创建流水线插件提供了更高的灵活性和可维护性。

准备自定义插件包

插件包结构

图 14-5 插件包结构



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。

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
|       | -- resource.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
```

新建流水线自定义插件

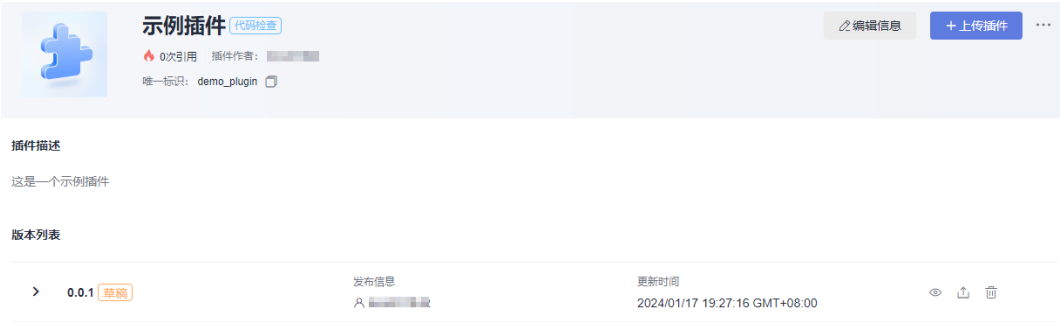
- 步骤1 访问CodeArts Pipeline首页。
- 步骤2 进入流水线服务首页，单击“服务 > 扩展插件”。
- 步骤3 单击“搭建流水线插件”旁边的下拉框，单击“上传流水线插件”，开始注册插件。
- 步骤4 填写基本信息，相关参数说明如表14-7所示。

表 14-7 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。 插件类型和category字段值对应关系如下： - 构建：Build。 - 代码检查：Gate. - 部署：Deploy。 - 测试：Test。 - 通用：Normal。
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 单击“确定”，进入“版本管理”页面。
- 步骤6 单击“上传插件”，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。

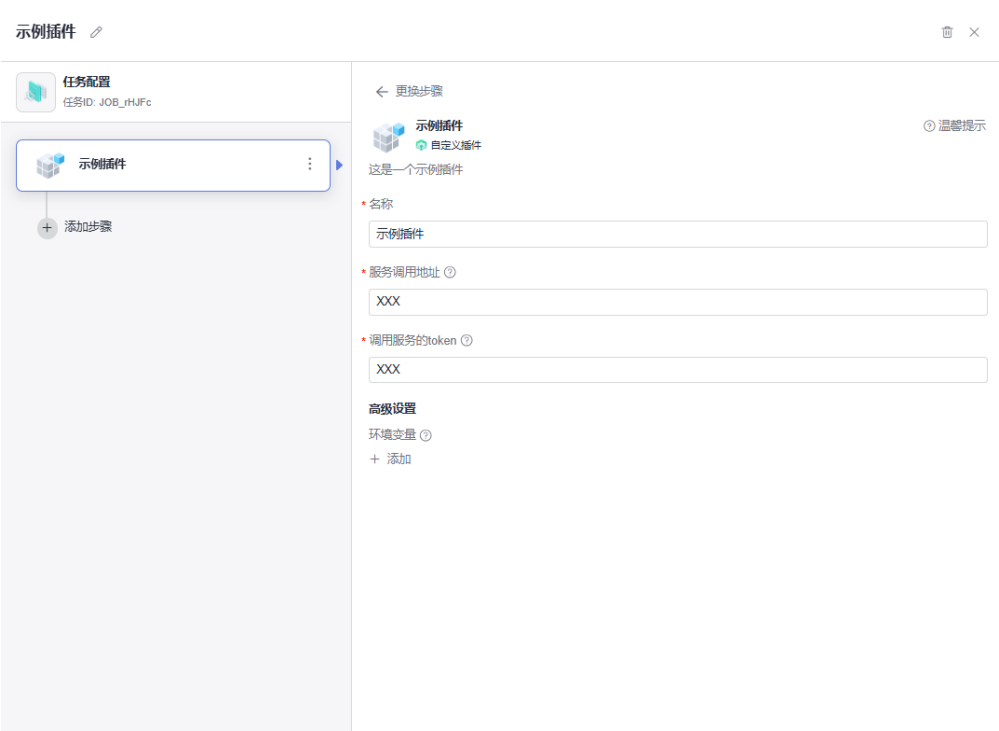
图 14-6 上传插件



步骤7 调试插件。

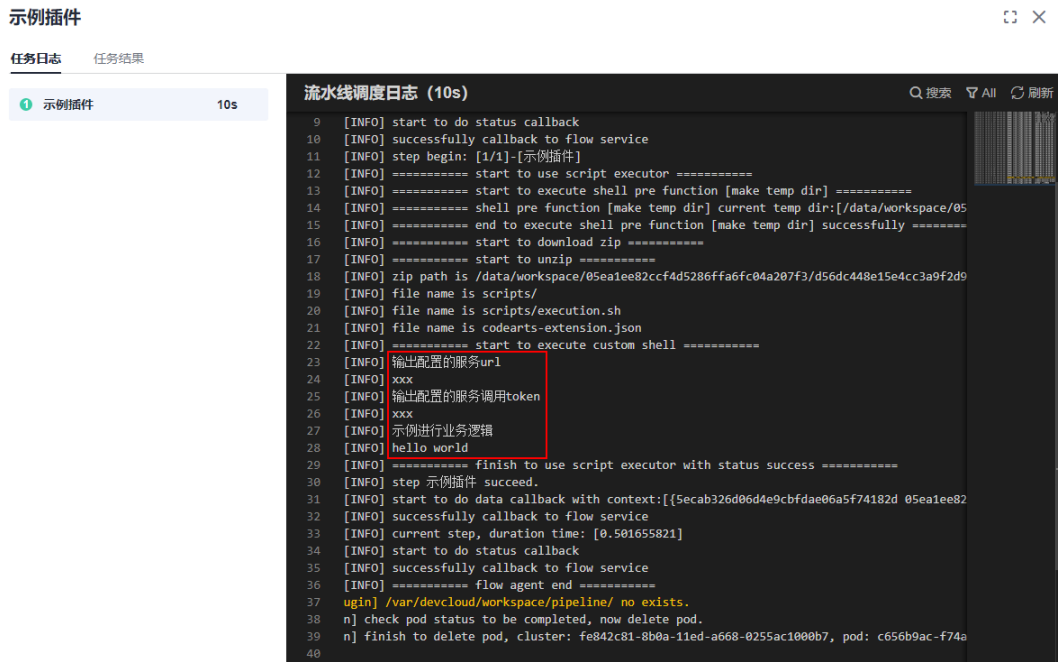
新建流水线任务，在“任务编排”页面新建任务，添加已注册的基础插件，填写参数信息。

图 14-7 调试插件



步骤8 单击“保存并执行”，执行流水线，执行完成后，单击插件名称，查看执行结果。

图 14-8 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 14-9 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
    }
  },
  {
    "name": "mkey",
    "type": "metrics",
    "description": "描述",           # output条目的描述信息，用于描述其值含义
    "prop": {
      "defaultValue": "213",
      "group": "213"                # 对应准出规则中的分组名称
    }
  }
],
"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",           # output条目的描述信息，用于描述其值含义
    "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 14-8 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	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： • Build：对应“构建”插件类型。 • Test：对应“测试”插件类型。 • Gate：对应“代码检查”插件类型。 • Normal：对应“通用”插件类型。 • Deploy：对应“部署”插件类型。
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。 output用于流水线上下文，输出内容可以在后续阶段使用或者在任务运行结果处展示；metrics的数据用于创建准出规则，具体参数信息请参考表14-12。

当前支持的inputs类型如下：

表 14-9 全量 inputs 类型

inputs类型	组件名称	样例	extendProp扩展
input	单行输入框	单行输入 请输入	• visibleConditions • disabledConditions

inputs类型	组件名称	样例	extendProp扩展
inputNumber	数字输入框	数字 请输入	- visibleConditions - disabledConditions
switch	开关	开关 ☐	- visibleConditions - disabledConditions
singleSelect	下拉单选框	下拉单选 请选择	- options - apiType - apiOptions
multipleSelect	下拉多选框	下拉多选 选项1 选项2 ☒ 选项1 ☒ 选项2 ☐ 选项3	- options - apiType - apiOptions
keyValuePair	键值对	键值对 请输入 请输入 + 添加	- visibleConditions - disabledConditions
radio	单选框	单选框 ☐ 选项1 ☐ 选项2 ☐ 选项3	options

inputs类型	组件名称	样例	extendProp扩展
timeInterval	时间间隔控件	时间间隔 00 ▼ 小时 00 ▼ 分钟 00 ▼ 秒	- visibleConditions - disabledConditions
shell	shell控件	Shell 1	- visibleConditions - disabledConditions
endpoint: \${XXX} (其中, “XXX”为扩展点“module_id”)	扩展点下拉单选框	测试扩展点 新建扩展点 刷新 请选择	- visibleConditions - disabledConditions

inputs基础字段说明如下：

表 14-10 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。 
required	是否必填。	否	带星号为必填。 
validation	校验信息，是一个对象，包含 requiredMessage、regex、regexMessage 三个属性。 <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	 
extendProp	扩展字段 <pre>{ visibleConditions: [], disabledConditions: [] ... }</pre>	否	针对特殊组件的功能扩展使用，详见 表 14-11 。

extendProp扩展功能说明如下：

表 14-11 extendProp 扩展功能

字段名称	含义	是否必填	备注
visibleConditions	按条件显示。	否	格式如下，可包含多个显示条件： [{};{};{};...] 示例： [{}comp:'key_001',symbol:'===', value: 'xxx'] 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。 symbol包含如下四种： • ===：等于 • !==：不等于 • empty：为空 • notEmpty：非空
disabledConditions	按条件禁用。	否	格式如下，可包含多个禁用条件： [{};{};{};...] 示例： [{}comp:'key_002',symbol:'!==', value: 'yyy'] 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。 symbol包含如下四种： • ===：等于 • !==：不等于 • empty：为空 • notEmpty：非空
options	固定下拉选项，类型为list。	否	示例： [{}label: '选项1',value: 1},{label: '选项2',value: 2}]

字段名称	含义	是否必填	备注
apiType	下拉选项类型： - fixed：使用options内的值作为下拉选项。 - api：通过接口请求，需配置apiOptions。	否	不填则为fixed。
apiOptions	JSON体，包含api接口使用的各个参数	否	示例： <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> JSON解析后： <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 body/params 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>

outputs基础字段说明如下：

表 14-12 outputs 基础字段

字段名称	含义	是否必填	备注
name	输出名称	是	同一基础插件内不允许重复。

字段名称	含义	是否必填	备注
type	输出类型。 • output: 在任务结果页面的“其他”卡片展示, 用于流水线上下文。 • metrics: 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门禁, 规则配置请参考配置流水线规则。 • 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。	是	-
description	描述信息, 用于描述其值含义, 在type为“metrics”时此处必填。	-	-
prop	在type为“metrics”时, group对应准出规则中的分组名称, 此时必填。	-	-

14.5 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。

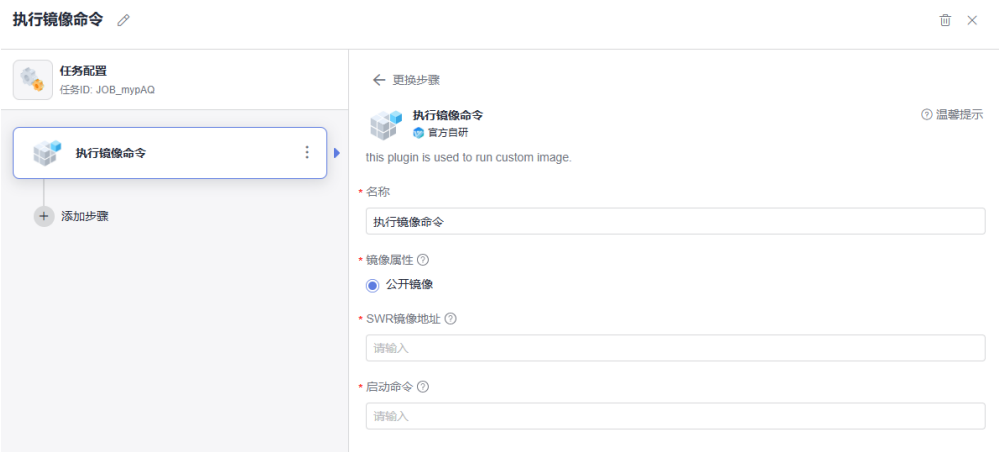
约束与限制

“执行镜像命令”插件只支持自定义执行机，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[资源池管理](#)。

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 14-10 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	插件名称。 - 仅支持中文、大小写英文字母、数字、“-”、“_”、“.”、“/”、“(”、“)”、“'”、“””及空格。 - 字符长度范围为1-128。
镜像属性	当前只支持公开镜像。
SWR镜像地址	待下载的SWR镜像的地址。获取方式如下： - 登录容器镜像服务。 - 在导航栏单击“我的镜像”，单击“镜像名称”进入镜像详情页面。 - 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。

参数项	说明
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

14.6 添加创建仓库 Tag 插件

创建仓库Tag插件，为代码仓创建Tag并推送。

配置方法

步骤1 在编排流水线任务时，添加“创建仓库Tag”插件。

图 14-11 创建仓库标签



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	插件名称。 - 仅支持中文、大小写英文字母、数字、“-”、“_”、“.”、“/”、“(”、“)”及空格。 - 至少输入一个字符。
Tag名	Tag名称。
仓库	选择对应的代码仓。

- 单击 ，开启“Tag存在时继续”，若创建的Tag在代码仓中已存在，则不报错，继续运行流水线，否则执行失败。

----结束

14.7 添加第三方接口调用 v2 插件

“第三方接口调用v2” 插件支持配置Web API调用第三方API。

约束与限制

- 此插件需要证书校验，不支持自定义证书校验和无证书校验。
- 接口返回码如果不是2XX，则默认为运行失败。

配置方法

步骤1 在[编排流水线任务](#)时，添加“第三方接口调用v2” 插件。

图 14-12 第三方接口调用 v2 插件



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	插件名称。 - 仅支持中文、大小写英文字母、数字、“-”、“_”、“.”、“;”、“:”、“.”、“/”、“(”、“)”、“(”、“)”和空格。 - 至少输入一个字符。
请求方式	可选择GET、POST、PATCH、PUT或DELETE。
请求路径	能被公网调用的接口，需要证书校验。
请求头	请求头，添加键值对。
请求参数	请求参数，添加键值对。

参数项	说明
请求体	如果请求方式是POST、PATCH或PUT，则输入json格式的body体。
输出	根对象名是response，可输出多个output。 ● 如果要拿到具体的请求返回码，并且输出的key是code，示例如下。 输出 ⓘ code response.status ● 如果返回的请求头中有date字段，并且需要拿到date时间的值，则输出处添加示例如下。 输出 ⓘ myheaderField response.headers.date ● 如果要输出整个body，示例如下 输出 ⓘ body response.data ● 如果返回的json体如下，并且要取field对应的value值，示例如下。 <pre>{ "result": { "field": "value", "field1": "value1" }}</pre> 输出 ⓘ fieldKey response.data.result.field

步骤3 插件运行完毕，如果想使用输出output，请参考[流水线上下文](#)。

----结束

15 管理 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 15-1 服务扩展点

类型	使用场景
Docker repository	用于连接第三方Docker镜像仓库。连接成功后，部署服务可以获取仓库中的Docker镜像。
Jenkins	用于连接第三方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，可以对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，构建任务可以获取该仓库中的文件信息。
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。
AGC	用于连接AGC Connect API。连接成功后，可以在流水线中使用该服务。

前提条件

- 拥有“插件市场 > 扩展点 > 创建”权限的用户可以新建服务扩展点。[项目成员如何查看/获取需要的操作权限？](#)
- 待接入的第三方服务能通过公网环境无限制地访问。

新建服务扩展点

步骤1 进入CodeArts首页。

- 登录[CodeArts控制台](#)，单击，根据需要选择已开通CodeArts的区域。
 - 单击“前往工作台”。
- 如果当前账号采用的是历史计费模式（详情请参考[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”，在弹框中选择类型。

步骤5 完成扩展点信息的配置。

新建成功后，页面中显示新建的扩展点名称。

----结束

各类服务扩展点的配置参数如下。

Docker repository

表 15-2 新建 Docker repository 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。

参数	说明
仓库地址	待连接的Docker仓库的地址，支持HTTP、HTTPS协议地址。
用户名	待连接的Docker仓库的用户名。
密码	待连接的Docker仓库的密码。

Jenkins

表 15-3 新建 Jenkins 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Server URL	待连接的Jenkins服务的地址。支持HTTP、HTTPS协议地址。
用户名	待连接的Jenkins服务的用户名。
密码	待连接的Jenkins服务的密码。

Kubernetes

表 15-4 新建 Kubernetes 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Kubernetes URL	待连接集群的server地址，可通过在集群配置文件“kubeconfig.json”中搜索“server”获取。
Kubeconfig	待连接集群的配置信息，可输入集群配置文件“kubeconfig.json”的全部内容。

nexus repository

表 15-5 新建 nexus repository 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
仓库地址	待连接的Nexus仓库的地址，支持HTTP、HTTPS协议地址。

参数	说明
用户名	待连接的Nexus仓库的用户名。
密码	待连接的Nexus仓库的密码。

通用 Git

表 15-6 新建通用 Git 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Git仓库Url	待连接的Git仓库的HTTPS地址，例如：https://*.*.*/user/repo.git。
用户名	待连接的Git仓库的用户名。 该参数为可选参数，当仓库需要验证用户名时配置。
密码或Access Token	待连接的Git仓库的密码或Access Token。 该参数为可选参数，当仓库需要验证密码或Access Token时配置。

码云 Git

表 15-7 新建码云 Git 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
验证方式	支持两种验证方式。 • 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。 • AccessToken认证：选择此方式，需要输入通过码云获取的私人令牌。获取方法请参考码云官网。

GitHub 服务

表 15-8 新建 GitHub 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
验证方式	支持两种验证方式。 ● 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。 ● AccessToken认证：选择此方式，需要输入通过GitHub获取的access token。获取方法请参考GitHub官网。

IAM 账户

表 15-9 新建 IAM 账户服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Access Key Id	待连接IAM用户的AK。通过“我的凭证”页面获取，获取方法请参考 新增访问密钥 。
Secret Access Key	待连接IAM用户的SK。通过“我的凭证”页面获取，获取方法请参考 新增访问密钥 。

CodeArts Repo HTTPS

表 15-10 新建 CodeArts Repo HTTPS 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
CodeArts Repo地址	待连接的CodeArts Repo仓库的HTTPS地址。 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	待连接的CodeArts Repo仓库的HTTPS用户名，格式为“租户名/IAM用户名”。 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。

参数	说明
密码	待连接的CodeArts Repo仓库的HTTPS密码，不超过300个字符。 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。

Gerrit

表 15-11 新建 Gerrit 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Gerrit地址	待连接的Gerrit仓库的地址，支持HTTPS协议地址。
用户名	待连接的Gerrit仓库的用户名。
密码	待连接的Gerrit仓库的密码。

GitCode

表 15-12 新建 GitCode 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
Token	GitCode的个人访问密钥。获取方法请参考 GitCode官网 。

AGC

表 15-13 新建 AGC 服务扩展点

参数	说明
连接名称	服务扩展点的连接名称，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。
客户端id	待连接的AppGallery Connect API客户端的客户端ID，获取方法请参考 AppGallery Connect官网 。
密钥	待连接的AppGallery Connect API客户端的密钥，获取方法请参考 AppGallery Connect官网 。

管理服务扩展点

- 步骤1 进入CodeArts首页。
- 1. 登录CodeArts控制台，单击📍，根据需要选择已开通CodeArts的区域。
 - 2. 单击“前往工作台”。
- 如果当前账号采用的是历史计费模式（详情请参考[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 在列表中单击目标服务扩展点，根据需要完成以下管理操作。

表 15-14 管理服务扩展点

操作	说明
编辑	单击“编辑”，在弹框中编辑各项参数值，单击“保存”。 修改成功后，页面中将显示修改后的信息。
删除	警告 删除后，将无法恢复，请谨慎操作。 单击“删除”，在弹框中单击“确定”。 删除成功后，页面中将不显示已删除的服务扩展点。

----结束

16 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 16-1 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

相关文档

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](#)。

17 参考

17.1 流水线上下文

17.1.1 流水线上下文

什么是流水线上下文

流水线上下文是指在流水线运行过程中，用于存储和传递与流水线相关的各种信息的一种机制或环境。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。使用流水线时，当一个任务生成了数据或结果，且这些数据或结果需要在后续的任务中使用时，可以基于流水线上下文来实现，流水线上下文可以帮助实现任务间的信息传递和协作，使得整个流水线运作更加灵活和高效。流水线上下文主要包括以下四类：

表 17-1 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> }}
```

- context：流水线上下文。
- attribute_name：属性。

流水线上下文属性介绍

表 17-2 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： project_id、 pipeline_id、 run_number、 timestamp、 trigger_type和 run_id。	- 内容示例 如下示例为一次手动运行的流水线包含的pipeline context信息。<pre>{ "project_id": "6428c2e2b4b64affa14ec80896695c49", "pipeline_id": "f9981060660249a3856f46c2c402f244", "run_number": "168", "timestamp": "20231016000004", "trigger_type": "Manual", "run_id": "c2f507f93510459190b543e47f6c9bec" }</pre> - 使用示例 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。<pre>\${ pipeline.trigger_type }</pre>
	pipeline.project_id	string	当前流水线所属项目ID，同现有 流水线系统预定义参数 PROJECT_ID。	
	pipeline.component_id	string	当前流水线所属微服务ID，同现有 流水线系统预定义参数 COMPONENT_ID。	
	pipeline.pipeline_id	string	当前流水线ID，同现有 流水线系统预定义参数 PIPELINE_ID。	
	pipeline.run_number	string	流水线执行编号，同现有 流水线系统预定义参数 PIPELINE_NUMBER。	
	pipeline.timestamp	string	流水线执行时间戳，同现有 流水线系统预定义参数 TIMESTAMP，格式为yyyyMMddHHmmss。如：20211222124301。	
	pipeline.trigger_type	string	流水线触发类型，同现有 流水线系统预定义参数 PIPELINE_TRIGGER_TYPE。	
	pipeline.run_id	string	流水线执行ID，同现有 流水线系统预定义参数 PIPELINE_RUN_ID。	

上下文	属性	类型	说明	示例
	pipeline.execute_user_name	string	流水线执行人名称，同现有 流水线系统预定义参数 EXECUTE_USER_NAME。	
	pipeline.upstream_status	string	流水线除后处理阶段外的前置阶段的状态。	
	pipeline.external_system	string	触发流水线的外部系统，同现有 流水线系统预定义参数 EXTERNAL_SYSTEM。该参数只在流水线被外部系统触发时才出现。	
	pipeline.external_user_id	string	外部系统触发流水线时的用户ID，同现有 流水线系统预定义参数 EXTERNAL_USER_ID。	
	pipeline.external_user_name	string	外部系统触发流水线时的用户名称，同现有 流水线系统预定义参数 EXTERNAL_USER_NAME。	

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、ssh_repo_url、tag、merge_id、source_branch和target_branch。	- 内容示例 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”。 <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> - 使用示例 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。 \${{ sources.my_repo.target_branch }}
	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 流水线系统预定义参数 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 流水线系统预定义参数 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。	
	sources.<alias>.repo_url	string	代码仓地址（HTTPS），同现有 流水线系统预定义参数 REPO_URL。	

上下文	属性	类型	说明	示例
	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。	
env context	name	string	自定义参数名称。	- 内容示例 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。<pre>{ "var_1": "val1", "var_2": "val2"}</pre> - 使用示例 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。<pre>\${{ env.var_1 }}</pre>
	value	string	自定义参数值。	

上下文	属性	类型	说明	示例
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。	- 内容示例 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。<pre>{ "check_job": { "status": "COMPLETED", "metrics": { "critical": "0", "major": "0" } }, "demo_job": { "status": "COMPLETED", "outputs": { "output1": "val1", "output2": "val2" } }}</pre> - 使用示例 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。<pre>\${{ jobs.demo_job.outputs.output1 }}</pre>
	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，即键值对形式的数据输出。	
	jobs.<job_id>.outputs.<output_name>	string	任务的运行一般输出的数据名称。	
	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](#)。

17.1.2 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。**如果是新建阶段任务，需添加插件后再单击“任务配置”。**

图 17-1 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，表达式代码语法如下：

```
${{ 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 17-3 操作符

操作符	说明
.	属性引用。如： <code>\${{ pipeline.trigger_type }}</code> ，表示获取流水线的触发类型。
!	非。如： <code>\${{ !startsWith(sources.my_repo.target_branch, 'release') }}</code> ，表示判断流水线代码源的分支是否不以release开头。
==	等于。如： <code>\${{ pipeline.trigger_type == 'Manual' }}</code> ，表示判断流水线的触发方式是否为手动触发。
!=	不等于。如： <code>\${{ pipeline.trigger_type != 'Manual' }}</code> ，表示判断流水线的触发方式是否不为手动触发。
&&	且。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 17-4 内置函数

函数	说明
contains	- 格式 contains(search, item) - 含义 如果“search”包含“item”，则函数返回“true”。 - 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”。 - 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”。 - 示例 contains('abc', 'bc')，函数将返回“true”。
startsWith	- 格式 startsWith(searchString, searchValue) - 含义 如果“searchString”以“searchValue”开始，则函数返回“true”。 - 示例 startsWith('abc', 'ab')，函数将返回“true”。

函数	说明
endsWith	• 格式 endsWith(searchString, searchValue) • 含义 如果 “searchString” 以 “searchValue” 结束，则函数返回 “true”。 • 示例 endsWith('abc', 'bc'), 函数将返回 “true”。
format	• 格式 format(string, replaceValue0, replaceValue1, ..., replaceValueN) • 含义 使用replaceValueN的值替换原始字符串string中的值。 • 示例 format('Hello {0} {1}', 'a', 'b')返回Hello a b。
substring	• 格式 substring(string, beginIndex, endIndex) • 含义 截取原始字符串string中下标从beginIndex到endIndex-1的字符串。不写endIndex，则返回beginIndex到末尾的字符串。 • 示例 substring('202412091101', 0, 8)返回20241209。 substring('202412091101', 8)返回1101。
replace	• 格式 replace(string, target, replacement) • 含义 将原始字符串string中与target匹配的每个子字符串替换为replacement。 • 示例 replace('hello a', 'a', 'b')返回hello b。
completed	• 格式 completed(job1, job2, ..., jobN) • 含义 在依赖的任务运行成功(不填写参数)或指定的job运行成功（填写参数）时，返回true。completed()在依赖的任务均运行成功 (COMPLETED)时，返回true。同 “任务配置” 中的 “前置任务成功时执行”。 • 示例 completed('job1', 'job2')在job1和job2均运行成功时，返回 true。

函数	说明
always	• 格式 always() • 含义 在依赖的任务运行结束后，不管它们的状态如何，均执行当前任务。同“任务配置”中的“始终执行”。
canceled	• 格式 canceled() • 含义 在依赖的任务运行被中止时，执行当前任务。
failed	• 格式 failed(job1, job2, ..., jobN) • 含义 在依赖的任务运行失败(不填写参数)或指定的job运行失败(填写参数)时，返回true。failed()在依赖的任务中任意一个运行失败(FAILED)时，返回true。同“任务配置”中的“前置任务失败时执行”。 • 示例 failed('job1', 'job2')在job1和job2中任意一个运行失败时，返回true。
对象筛选器	使用“*”语法应用筛选器并选择集合中的匹配项。 示例：如下是流水线某次运行的jobs上下文 <pre>{ "check_job": { "status": "COMPLETED", "metrics": { "critical": "0", "major": "0" } }, "demo_job": { "status": "FAILED" } }</pre> • 示例1：“jobs.*.status”，表示所有任务的状态，所以返回“['COMPLETED', 'FAILED']”。 • 示例2：结合contains函数一起使用，如： contains(jobs.*.status, 'FAILED')，“jobs.*.status”中包含“FAILED”，所以返回“true”。

17.1.3 配置通过流水线上下文获取构建任务的产物信息

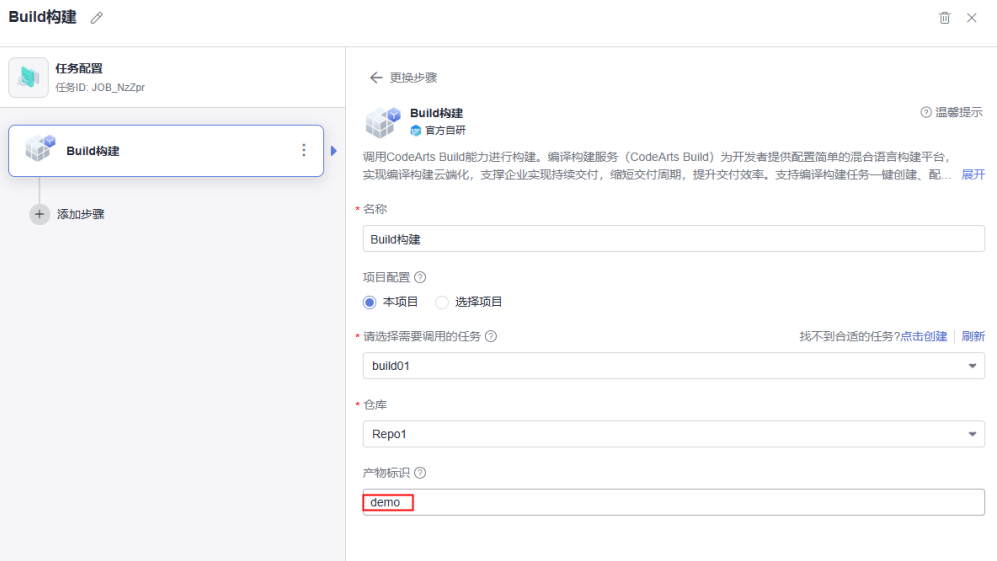
在流水线阶段任务的输入框中通过引用流水线上下文获取相关信息。本示例介绍如何在流水线上使用“Build构建”插件生成构建产物，并在后续阶段的“执行Shell”任务中通过流水线上下文引用，获取构建产物信息。

- 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17-2，并将构建产物标识设置为“demo”，如图17-3。

图 17-2 获取任务 ID



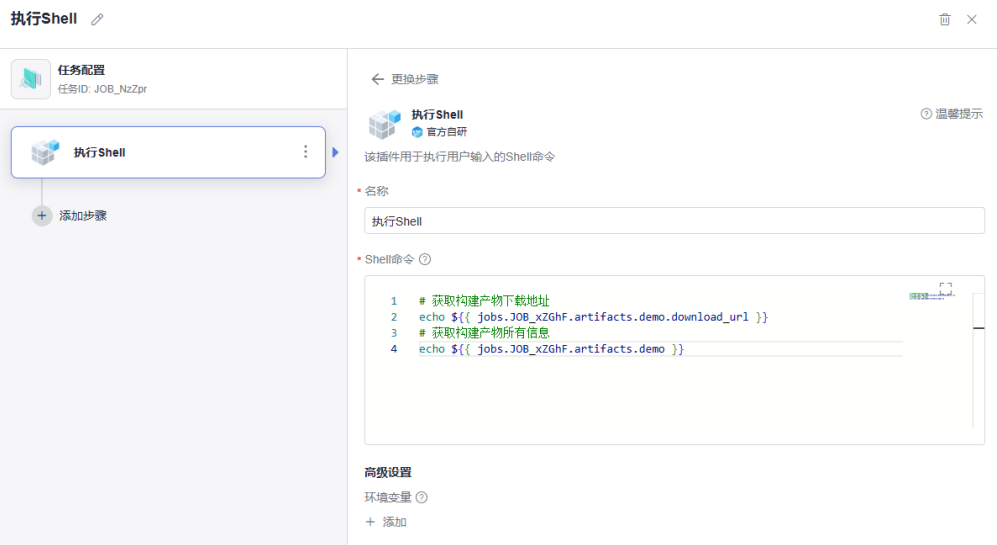
图 17-3 Build 构建插件



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。

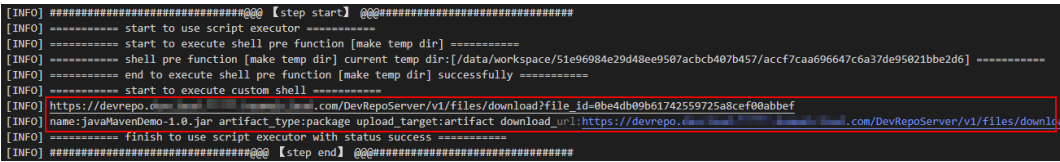
```
# 获取构建产物下载地址
echo ${ jobs.JOB_xZGhF.artifacts.demo.download_url }
# 获取构建产物所有信息
echo ${ jobs.JOB_xZGhF.artifacts.demo }
```

图 17-4 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 17-5 查看产物信息



----结束

17.2 YAML 语法规则

17.2.1 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 17-5 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。为避免产生歧义，不能在pull_request事件中同时指定branches和branches-ignore。

- 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。

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

表 17-6 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。	- v[a-z].[0-9]*: 可匹配va.1、vb.111等。 - v[01]: 可匹配v0、v1。
{}	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。**为避免产生歧义，不能在push事件中同时指定branches、branches-ignore、tags和tags-ignore。**

- 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。

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

如果需要同步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。**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。**

- 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。

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

如果需要同步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。

17.2.2 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](#)。

17.2.3 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Tag**

为代码仓创建Tag并推送。

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

tagName为Tag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 17-7 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
Singapore Standard Time	GMT+08:00

可选值	时区
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: '3'
  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17-8](#)，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：审核超时时策略。reject表示中止；pass表示通过。
- comment：审核描述。

表 17-8 审核角色

审核角色	YAML标识
项目经理	'3'
开发人员	'4'
测试经理	'5'
测试人员	'6'
参与者	'7'
浏览者	'8'

审核角色	YAML标识
运维经理	'9'
产品经理	'10'
系统工程师	'11'
Committer	'12'

18 发布管理(CodeArts Release)使用指南

18.1 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 Deploy，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。

18.2 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 > 华为云码道”。

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18-1所示。

表 18-1 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。 ● 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，了解更多。 ● 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，了解更多。 ● 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。
发布用户	可选择当前用户和其他用户。 ● 当前用户：基于当前用户下的集群创建环境。 ● 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，具体操作请参考新建CodeArts服务扩展点。
区域	资源类型为CCE时，需设置此参数。 选择部署环境的区域。
集群	资源类型为CCE时，需设置此参数。 选择在CCE云容器引擎中 已购买的Kubernetes集群 。
关联类型	资源类型为UCS时，需设置此参数。 关联的UCS资源粒度，当前仅支持 容器舰队 。
容器舰队	资源类型为UCS时，需设置此参数。 选择 已创建的舰队 。
Kubernetes扩展点	资源类型为K8S时，需设置此参数。 支持以扩展点的方式关联集群资源，以Kubernetes扩展点配置的凭证信息访问集群资源，具体操作请参考 新建CodeArts服务扩展点 。
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 项目级权限
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 18-2 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

18.3 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配置文件中使用“{{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 18-3 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 18-4 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。
- 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 18-1 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 18-2 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
```

18.4 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

灰度升级模板

策略描述

基于ASM或Service的灰度发布

编排插件

灰度起负载

灰度引流

灰度下线

灰度起负载

部署方式 ?

☒ 镜像升级 ☐ YAML部署

* 命名空间

请选择

* 服务 ?

请选择

自定义灰度版本号

☐

表 18-5 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	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 配置发布管理原子插件 。 - 单击, 可添加插件。 - 单击, 可复制当前插件。 - 单击, 可删除当前插件。

- 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“默认”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](#)、[通用k8s部署插件](#)、[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 18-6 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 18-7 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。 - 当前目录为代码分支的根目录。 - 仅支持单个YAML文件。 - 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。

- 通用k8s部署插件**
通用k8s部署插件，按照用户配置的代码仓信息及文件路径，取到对应的文件，并将其下发至用户的k8s集群。

通用k8s部署

* 仓库类型

Repo

* 代码仓

CK-1

* 分支

master

* 资源路径 ?

deployment.yaml

表 18-8 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。
资源路径	填写YAML文件的相对路径。 - 当前目录为代码分支的根目录。 - 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。

- 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 18-9 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。 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 18-10 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 YAML路径	填写YAML文件的相对路径。 - 当前目录为代码分支的根目录。 - 仅支持单个YAML文件。 - 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。

- 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 18-11 参数说明

参数项	说明
超时处理	超过等待审批时间后流水线的处理方案。 ● 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。 ● 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。
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。

18.5 配置发布管理环境的发布

发布管理环境可直接发布或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。

发布管理环境的直接发布

- 步骤1 新建发布管理环境
- 步骤2 单击右上角“执行”，弹出“执行配置”侧滑框，填写相关参数。
 - 发布策略：选择配置发布管理发布策略时配置好的发布策略。
 - 产物地址：待发布的镜像产物地址，可直接填写SWR镜像地址，或共享中心镜像（如"nginx:latest"）。
- 步骤3 单击“确定”，完成发布。
- 结束

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18-12。
- 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 18-3 配置云原生发布插件

云原生发布

官方自研

⑦ 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：滚动发布和灰度发布）

* 名称

云原生发布

* 环境级别

请选择

* 环境

请选择

* 产物地址 ⑦

请输入

表 18-12 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。

参数项	说明
环境	需要发布的环境，详见 新建发布管理环境 。
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}。 说明 推荐使用SWR容器镜像服务，可通过编译构建 制作镜像并推送到SWR仓库 。

- 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 18-4 查看云原生发布插件运行结果



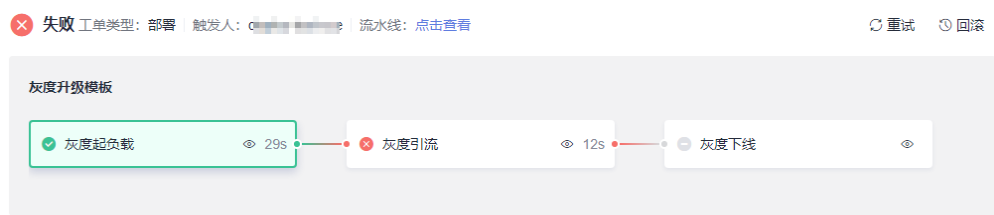
- 任务日志：查看实时日志信息和运行状态。
- 任务结果：查看任务基本信息，包括工单名称、工单ID、触发人等。
单击工单ID或“查看详情”可进入发布管理工单详情页面，工单详情页介绍请参见[查看发布管理发布结果](#)。

----结束

18.6 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 18-5 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

环境直接发布时，部署工单不展示流水线信息。

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 18-6 原子插件的发布详情

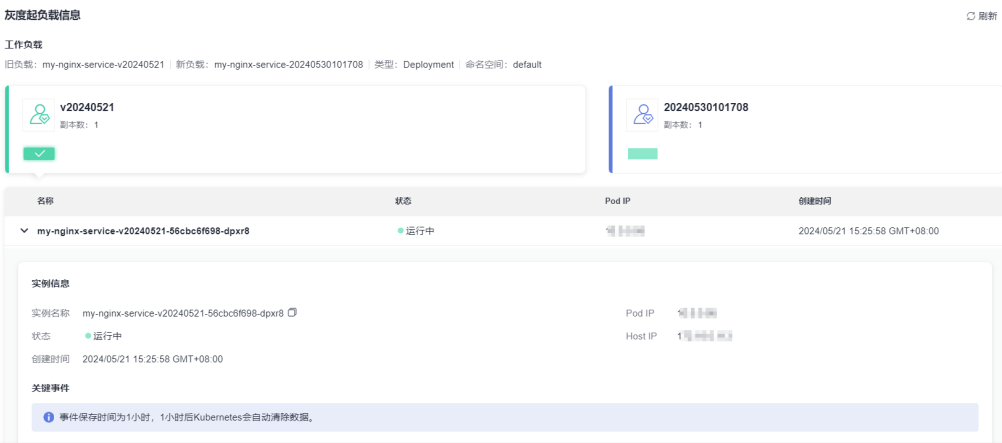


表 18-13 原子插件发布单信息

插件类型	发布单信息
滚动升级	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。 - 工作负载信息 负载名称、类型、命名空间及创建时间。 - 实例信息 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。 - 关键事件 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。

插件类型	发布单信息
灰度起负载	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。 - 工作负载信息 旧负载名称、新负载名称、负载类型及命名空间。 - 实例信息 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。 - 关键事件 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。
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。
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