

ALIBABA CLOUD

阿里云

金融分布式架构
研发效能

文档版本：20220710

 阿里云

法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

- 1.产品简介 06
 - 1.1. 概述 06
 - 1.2. Smartunit 介绍 08
- 2.功能总览 09
- 3.项目协作 10
 - 3.1. 什么是项目协作 10
 - 3.2. 项目管理 10
 - 3.3. 工作项管理 11
 - 3.4. 冲刺管理 13
 - 3.5. 设置 15
- 4.持续交付 19
 - 4.1. 什么是持续交付 19
 - 4.2. 工作台 19
 - 4.3. 研发迭代 20
 - 4.3.1. 应用管理 20
 - 4.3.2. 研发迭代管理 22
 - 4.3.3. 架构域管理 27
 - 4.3.4. 发布管理 30
 - 4.3.5. 配置变更 34
 - 4.3.6. 流水线管理 37
 - 4.3.7. 流水线组件 38
 - 4.4. 代码服务 51
 - 4.4.1. 仓库管理 51
 - 4.4.2. 分支管理 52
 - 4.4.3. 合并请求 53
 - 4.4.4. 密钥管理 53

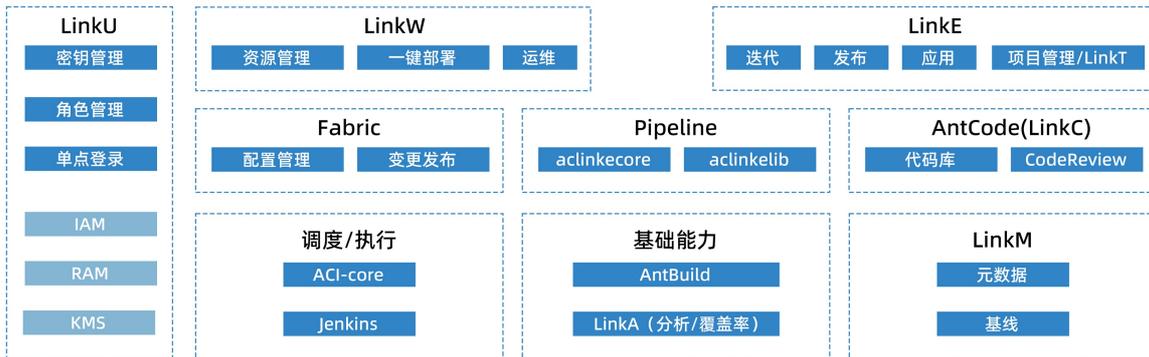
4.4.5. 权限/成员管理	54
4.4.6. 群组管理	55
4.5. 研发环境	55
4.5.1. 管理应用服务	55
4.5.2. 查看操作记录	57
4.6. 流程中心	58
4.7. 管理员配置	60
4.7.1. 部署配置	60
4.7.2. 流水线配置	61
4.7.3. 迭代配置	62
4.7.4. 构建配置	62
4.7.5. 角色配置	66
4.8. 最佳实践	71
4.8.1. 合并请求	71
4.8.2. 自定义构建发布	75

1. 产品简介

1.1. 概述

研发效能平台，源于蚂蚁集团多年互联网金融领域研发背景和工程实践的深厚沉淀，为企业 提供多样化、低成本、可信赖的解决方案，可高质量、高效率地解决企业研发过程中的常见问题，在让您享受极简研发流程的同时，支撑企业产品高可用和研发效能的持续提升。目前，研发效能分为持续交付、项目协作和脚手架三个模块。

系统架构

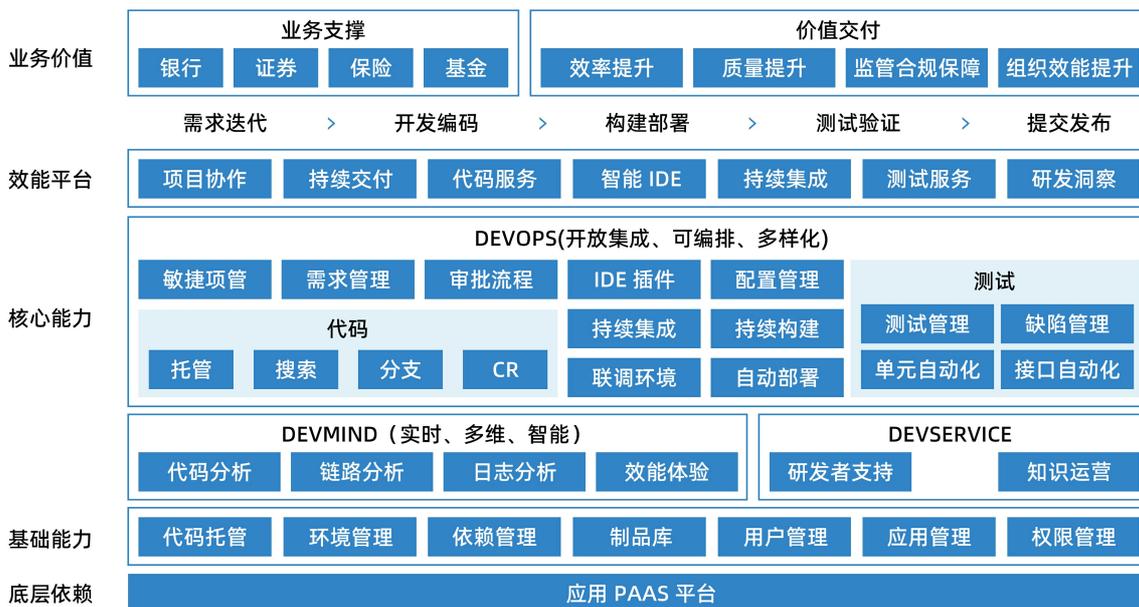


产品优势

- 安全稳定**
 基于 SOFAStack, LinkE 使用高可用、可动态扩展的服务框架体系，构建稳定的分布式代码托管服务，实现多中心、去存储、可靠的运维监控及全面的自动化质量保证，保障研发流程的顺畅。
- 智能研发体系**
 LinkE 集成多种工具平台，既能让您感受编写代码的极速与代码运行的稳定，又能提供对代码的实时监控和智能分析，帮助研发团队全方位立体式的进行代码透析，提前规避研发风险。
- 多样化管理**
 LinkE 具备灵活弹性的配置能力，能够根据企业的多种形态特性，提供适用于不同企业诉求的多样化研发流程和质量标准。
- 一站式服务**
 LinkE 提供端到端的工具服务，覆盖研发全生命周期，平台功能更加全面，极大地简化应用的部署、发布流程，轻松实现应用快速上云，让产品开发变得更加简单。
- 热拔插组件**
 可编排引擎，轻松支持插入定制化的持续集成（Continuous Integration，简称 CI）和持续交付（Continuous Delivery，简称 CD）组件，基于可配置的规则，适应研发节点上下文，自动化推进流程的检测流水线（Inspection Pipeline）。
- 数据驱动**
 数据研发流程体系化，数据质检手段工具化、多样化，数据环境稳定多样，从规划、设计、实现、部署、维护、下线六个阶段监控和度量服务数据，更高效、稳定地进行架构治理。
- 快速构建应用**
 脚手架模块提供集成中间件依赖的 SOFABoot 工程快速创建及下载，开发者只需专注业务开发，简化了研发的难度，提升研发效率，帮助开发者快速构建及管理分布式应用。

功能特点

功能总览



- 研发容器**
 集中统一管理配置项，只需使用简单的配置文件与命令，即能一键式快速适配多环境自动同步。
- 持续交付**
 基于 Gitflow 工作流的最佳实践，组件可灵活编排，只需简单点击操作，即可在几分钟内触发并完成代码扫描、代码评审（code review，简称 CR）、自动化测试、自动编译部署等核心功能。
- 智能 IDE**
 基于 Git 分支的轻量级管理，集成阿里开发规约，支持对合并（merge request，简称 MR）源分支全量代码的实时扫描。提交预扫支持更丰富的代码差异对比（简称 diff）呈现模式，以及智能 QuickFix 等效率工具。
- 代码托管**
 采用支撑百万用户规模的 GitLab 架构，进一步提升性能、确保数据安全、实现单元化部署，形成更安全、稳定、高效、智能、集成、开放的代码生态系统。多样化分支策略支持高扩展、分布式、响应式、大文件、按需下载，更有流控、安全、数据化、单元化等诸多强大提升。
- 流程中心**
 流程规则可按业务需求灵活配置，也可根据软件开发协作状况自由定制，支持规则自定义、加权自定义，支持多成员协作，具有强大的适应性。
- 研发协作**
 专为软件研发团队打造的简洁、易用、整合的一站式协作平台，减少学习成本，以产品、项目为维度，多维度集成管理需求、任务、缺陷、迭代、看板、文档，轻量级支持敏捷开发等多种实践方法。
- SOFABoot 工程构建**
 脚手架提供了一个简单易用的 Web 界面，开发者只需在配置页填写应用基本信息，选择需要引入的中间件依赖，即可快速创建并下载一个集成了相应中间件的 SOFABoot 工程。

应用场景

研发效能平台可为多种类型的用户提供针对各种场景的解决方案，常见应用场景如下：

中小企业

- 缺乏研发和项目管理经验，进度很难有效管控。
- 缺乏代码规范，代码质量低。

- 缺少功能完备的研发工具平台，基于开源平台拼凑的工具无法形成完整的支撑和数据链路。
- 缺少配套易用的发布管理平台，环境部署和维护等与价值创造无关的活动耗费大量时间。

互联网企业

- 经常涉及多团队协同合作，但各研发团队步调时常出现不一致状况。
- 经常涉及多模块协同研发，但内部研发协调经常出现问题。
- 经常需要并行研发，但代码冲突频繁，版本管理混乱。
- 经常需要快速迭代，但缺少涵盖全链路的一站式研发部署平台。

互联网金融企业

- 核心系统上线需要稳定可靠万无一失，上层业务需要快速上线适应市场抢占先机。
- 敏捷交付，持续交付，每一次变更，都需要稳妥的发布风险管控和资金风险防控。

术语

- **项目 (Project)**：具有独特的过程，有开始和结束日期，由一系列相互协调和受控的活动组成。过程的实施是为了达到规定的目标，包括满足时间、费用和资源等约束条件。
- **工作项 (Work items)**：研发项目中需求、缺陷、任务的集合。
- **研发迭代 (iteration)**：为实现需求或修复缺陷而进行的一次研发活动，一个研发迭代可能包含一个或多个应用。
- **元数据信息 (Metadata info)**：应用的各种基础信息、研发配置集合。
- **发布窗口 (Release window)**：控制和协调多个迭代进行统一发布的实体，一个发布窗口可以关联一个或多个迭代。
- **仓库 (Repository)**：特指代码仓库。

SmartUnit 介绍

PinPoint 介绍

1.2. Smartunit 介绍

SmartUnit 是由蚂蚁集团打造的智能单测用例生成平台，可以帮助开发人员提升基础代码覆盖率，同时提供断言自动生成、辅助发现程序逻辑问题。

Smartunit 产品介绍

Smartunit Demo 演示

2.功能总览

研发效能平台，源于蚂蚁集团多年互联网金融领域研发背景和工程实践的深厚沉淀，为企业提供多样化、低成本、可信赖的解决方案。

平台主要功能模块如下：



项目协作

- 支持集中式管理项目架构
- 按需定制工作项模板和流程
- 支持项目冲刺规划，快速交付业务价值



研发迭代

- 创建应用开始研发活动
- 可按业务需求灵活变更迭代配置
- 通过流水线来编排持续集成



代码服务

- 提供基于 Git 的代码托管服务
- 支持灵活强大的分支管理策略
- 权限管理保证代码数据安全



研发环境

- 提供高性能的容器集群管理方案
- 低成本查看和追踪操作历史



流程中心

- 按业务需求灵活配置研发流程规则
- 配置质量标准并进行卡点管控



管理员配置

- 支持跨云租户的环境部署
- 支持租户级别的构建配置
- 丰富的角色与权限选择满足企业研发需求

3.项目协作

3.1. 什么是项目协作

项目协作是一款专为软件研发项目团队打造的简洁、易用、整合的一站式协作平台，以项目为切入点，多维度集成管理需求、任务、缺陷、冲刺、看板，提供轻量级支持敏捷开发等多种项目管理的实践方法。项目协作贯穿软件研发全生命周期，如需求管理、项目管理、冲刺管理、工作项管理、风险管理等环节，让沟通随时进行，提高项目管理透明度，激发团队和组织热情，建设更好的跨团队高效协作文化。

项目管理

支持项目集、冲刺、工作项、风险、里程碑等主要项目管理功能，集中式管理项目架构，能完整清晰地追踪整体进度、阻塞模块、风险矩阵。项目成员可轻松实现任务分解、分派和跟踪，完成工作计划、执行、汇总透视等事项，团队在项目空间内得以高效协作，项目管理有条不紊尽在掌握。

工作项管理

统一需求、任务、缺陷的管理模型，按需定制工作项模板和流程，化繁为简，高效有序工作；支持多维视图展示（列表视图、树状视图和看板视图），简单友好的看板模式助力研发团队更高效的实践敏捷站会。

冲刺管理

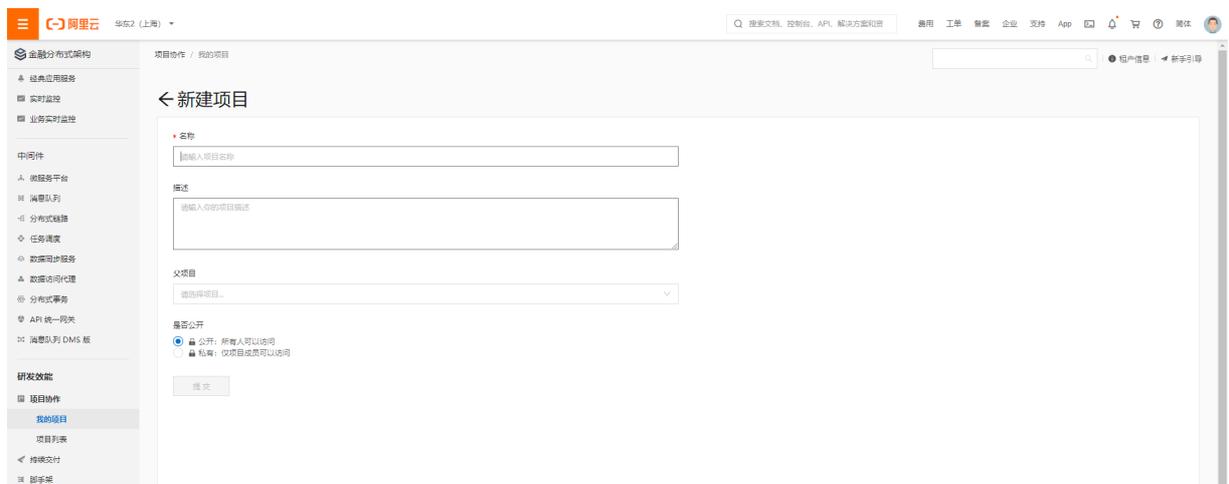
高效进行冲刺规划、锁定、执行、复盘等活动，融合敏捷最佳实践，可将一个项目规划为多个冲刺快速交付业务价值；更有燃尽图等管理工具一目了然地呈现研发进度，让项目节奏更稳当有序。

3.2. 项目管理

本文主要介绍项目协作中项目的新建、修改、归档、废弃相关功能。

新建项目

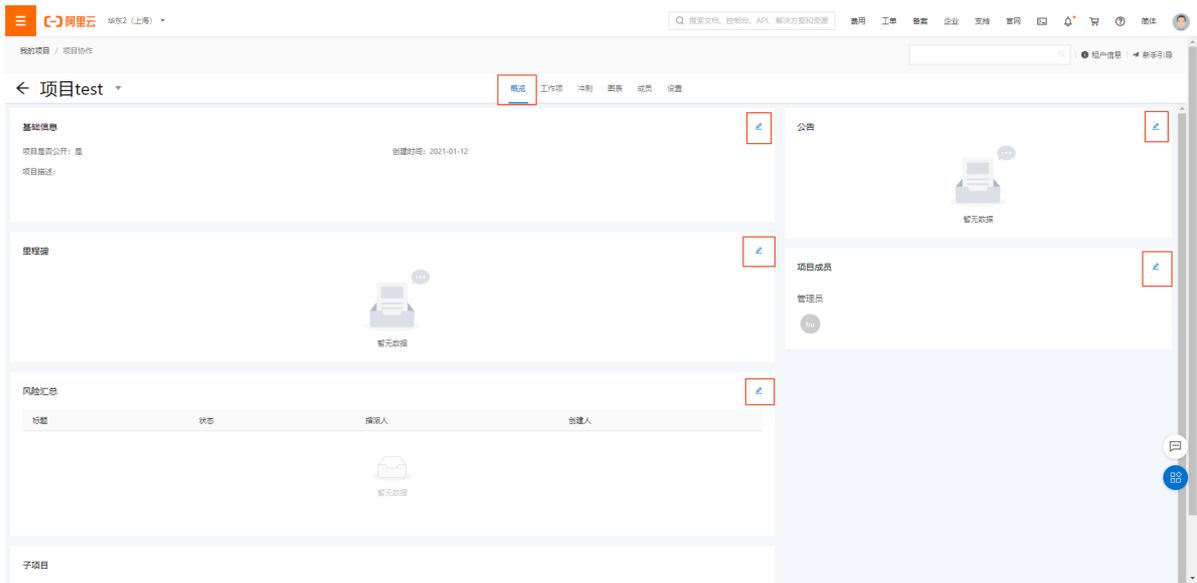
1. 登录 [金融分布式架构控制台](#)。
2. 在左侧导航栏单击 [研发效能](#) > [项目协作](#)。
3. 在 [我的项目](#) 或 [项目列表](#) 中单击 [新建项目](#)，进入项目创建页面。
4. 填写项目名称、项目描述，设置项目是否公开以及项目的父项目后，单击 [提交](#) 按钮，即可完成项目的创建。



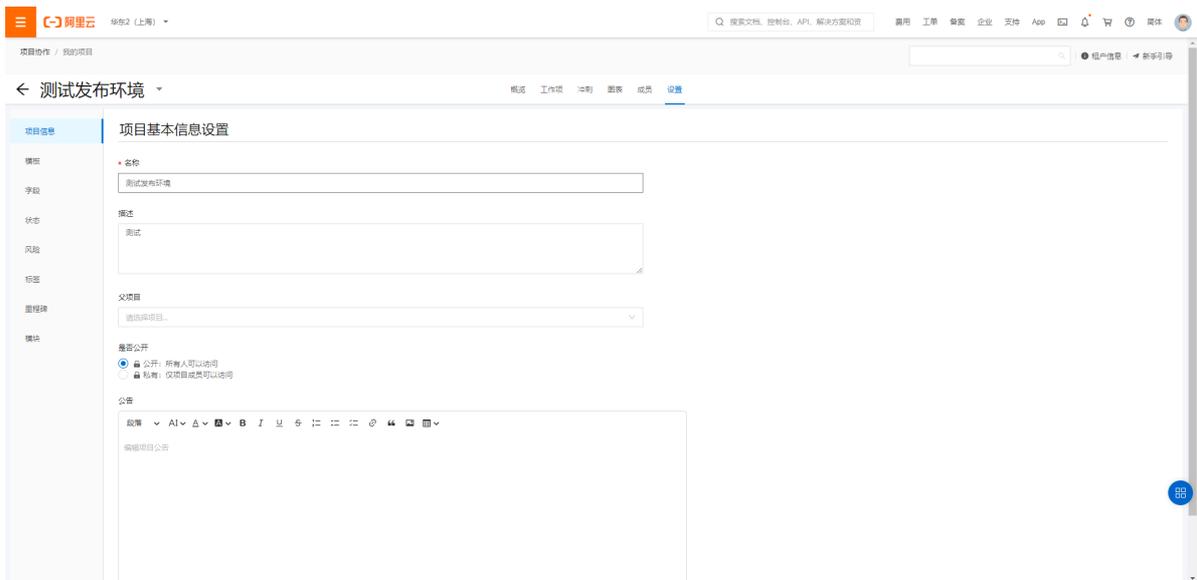
修改项目信息

您可以通过以下两种路径方式对项目的基础信息、公告、里程碑、风险汇总、项目成员等信息进行修改。

- 单击目标项目名称，然后在 **概览** 页面单击目标项目信息右侧的 **编辑** 按钮进入编辑页面。



- 单击目标项目名称，然后在 **设置** 页面选择相应的项目信息进行修改。



归档项目

1. 在 **我的项目** 或 **项目列表** 中，选择需要归档的项目，单击目标项目名称进入项目详情页。
2. 切换到 **设置** 标签页并进入项目详情菜单。
3. 在页面下方，单击 **归档** 或 **取消归档** 按钮分别进行对应的操作。

废弃项目

1. 在 **我的项目** 或 **项目列表** 中，选择需要归档的项目，单击目标项目名称进入项目详情页。
2. 切换到 **设置** 标签页并进入项目详情菜单。
3. 在页面下方，单击 **废弃项目** 按钮将当前项目废弃。

3.3. 工作项管理

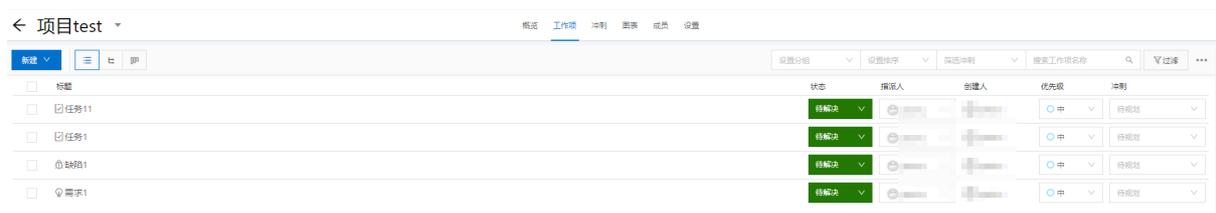
工作项的设置可以帮助您对整个项目进行细分，从而更加清晰的了解整个项目的进度。本文主要介绍工作项的管理，包括工作项列表的功能以及如何新建工作项。

新建工作项

1. 进入项目协作控制台，在 **我的项目** 或 **项目列表** 中单击目标项目名称，进入项目详情页。
2. 在项目详情页，切换至 **工作项** 标签页。
3. 单击 **新建** 按钮，选择工作项类型。工作项分为需求、任务、缺陷三类。
4. 在工作项编辑窗口中，根据工作项类型填写相应的工作项信息，单击 **确认** 完成创建。

工作项列表

工作项列表提供了基本的工作项过滤、分组、编辑能力。工作项列表整体视图如下：



编辑工作项

在工作项列表中，单击任一工作项名称。在打开的工作项编辑窗口中可进行如下操作：

- 编辑工作项基本信息（标题、描述、指派人、优先级等）。
- 上传附件。
- 新建子工作项或关联子工作项。
- 规划迭代。
- 提交评论。
- 查看操作记录。
- 对于需求类型的工作项，支持提交需求评审和查看修订。

视图

工作列表的视图有三种呈现方式：

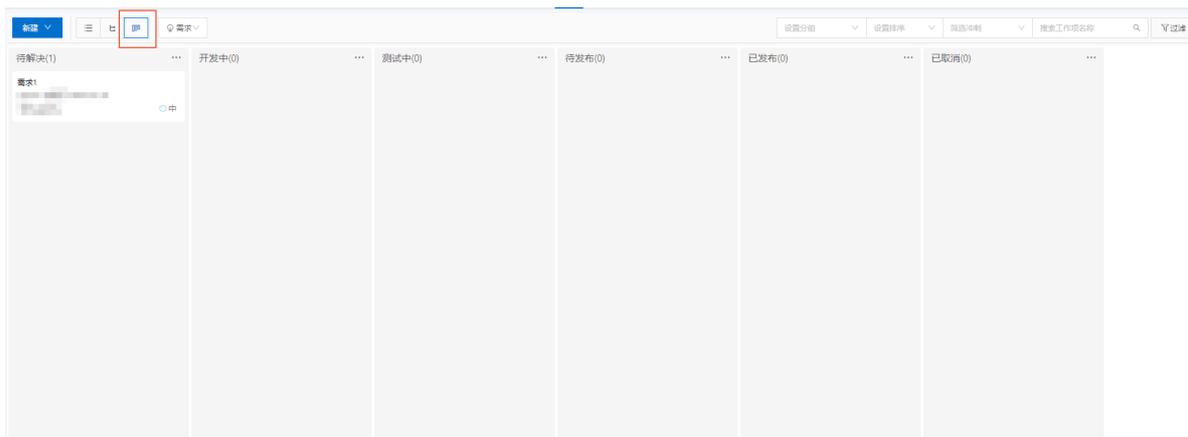
- 列表视图，可以直观的看到所有已经创建的工作项。



- 树状视图，可以直观的看到所创建工作项之间的关系。



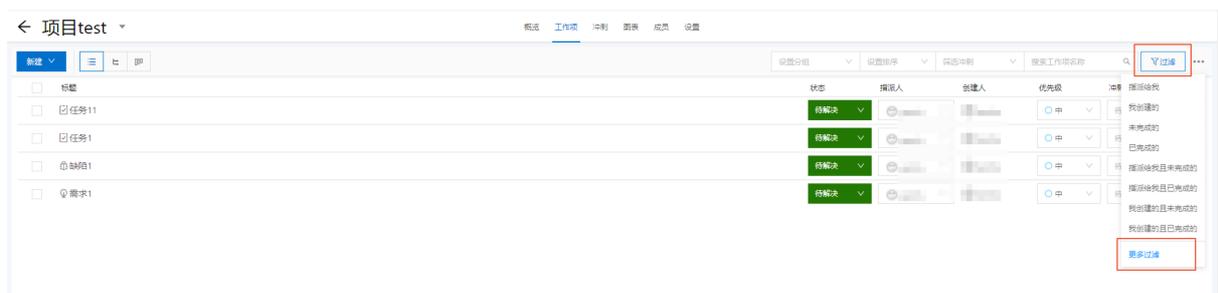
- 看板视图，可以直观的看到所创建工作项的状态。



说明：在看板视图下，您可以通过鼠标拖动工作项的方式直接更改工作项的状态，针对工作项状态的设置请参见 [设置](#)。

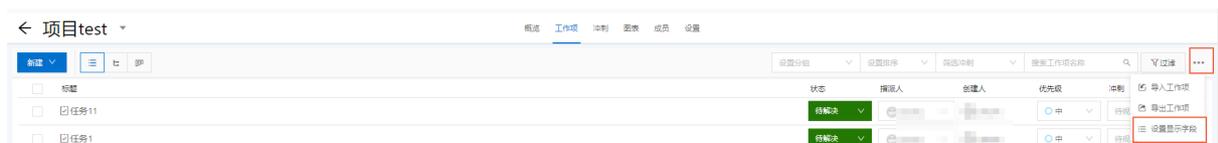
过滤

在项目详情页右侧，您可以单击 **过滤**，对所有工作项进行过滤，也可以单击 **更多过滤** 来添加过滤条件。



设置显示字段

如果你想更改工作项列表上显示的字段，您可以在项目详情页最右侧，单击 **过滤** 按钮右侧按钮，然后单击 **设置显示字段** 进行操作。



导入导出工作项

在项目详情页最右侧，单击 **过滤** 按钮右侧按钮，然后单击 **导入工作项** 或者 **导出工作项** 进行操作。



3.4. 冲刺管理

冲刺（Sprint）的设置主要是为了应对项目中比较紧急或重要的需求，本文主要介绍项目协作中冲刺的新建、修改、规划等相关功能。

新建冲刺

1. 进入项目协作控制台，在 **我的项目** 或 **项目列表** 中单击目标项目名称，进入项目详情页。

2. 切换至 **冲刺** 标签页，单击 **新建冲刺** 按钮。



3. 在 **新建冲刺** 窗口中，输入名称、负责人，并设置冲刺起止时间，单击 **确认**，完成冲刺创建。

修改冲刺

1. 在项目详情页，切换至 **冲刺** 标签页。
2. 在冲刺列表中，选择要编辑的冲刺，单击 **编辑** 按钮，在 **编辑冲刺** 弹窗中修改冲刺信息，包括 **起止时间**、**名称**、**负责人**。



3. 修改完成后，单击 **确认** 保存修改。

删除冲刺

1. 在项目详情页，切换至 **冲刺** 标签页。
2. 在冲刺列表中，选择要编辑的冲刺，单击 **删除** 按钮。



规划冲刺

冲刺规划的入口有两个：

- 在冲刺列表视图内单击各冲刺对应的 **规划** 按钮。



- 在冲刺下的工作项视图内，单击右侧的 **规划** 按钮。



冲刺规划视图如下图所示，可通过选择工作项并进行左右移动来规划冲刺。同时，冲刺规划视图也支持对冲刺进行过滤。



说明

您可以通过规划冲刺的方式来为目标冲刺添加需求、任务等信息，也可以直接在冲刺中新建需求、任务等信息。

冲刺详情

通过 **冲刺详情** 页面您可以把冲刺与持续交付研发迭代进行关联，同时冲刺的规划会议、复盘会议等需要记录的信息，可速记在 **冲刺详情** 页面。



3.5. 设置

为了保证项目信息更加符合需求，在项目详情页设置页面下，您可以自定义项目中的一些信息。

项目信息

在项目详情页下可执行如下操作：

- 编辑项目基本信息，包括项目名、项目描述、项目是否公开、父项目。
- 归档或取消归档项目。
- 废弃项目。



模板

工作项基于模板提供了自定义能力，工作项模板分为 **需求**、**缺陷**、**任务** 三种类型，每种工作项模板对应了一套 workflow 和若干自定义字段。

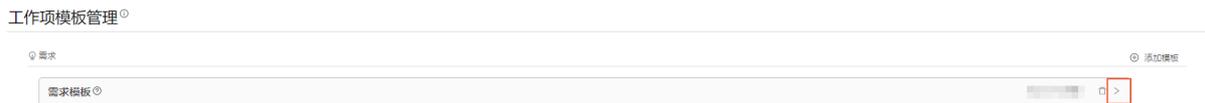
说明

在项目创建时，平台会在该项目下为每种工作项类型生成一套默认模板。

模板设置页面整体视图如下：

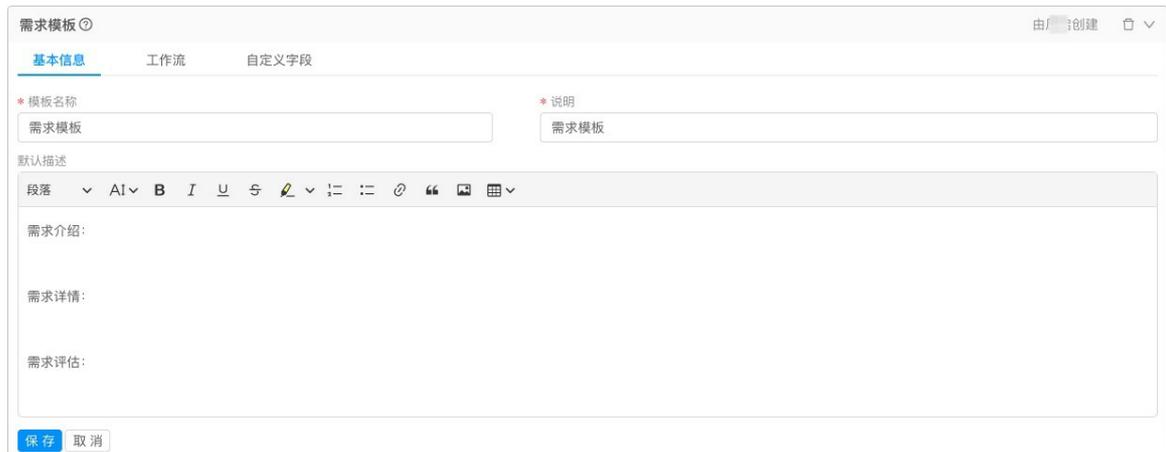


单击模板右侧 > 按钮进入设置页面。



模板设置页面提供的能力包括：

- 编辑模板
 - 您可编辑模板的基本信息，包括模板名、模板说明、模板默认描述。



- 您可编辑模板对应的工作流，包括新增、删除、修改、移动工作流状态。



- 您可编辑或新增模板下的自定义字段，设置自定义字段是否必填，编辑自定义字段在工作项中的显示顺序。



- 添加模板
 - 从全局需求模板导入。
 - 基于空白模板新建。

字段

在字段设置页面右上角单击 **新建自定义字段** 来新建字段。



说明
 字段新建完成后，字段类型不可修改。

风险

风险设置页面显示了当前项目下的所有风险类型的工作项，您可新建风险、编辑或删除已有的风险。

标签

在标签设置页面右上角单击 **新建标签**，然后填写名称并选择标签颜色，完成后单击 **确认**。

创建标签 ×

请填写标签名称...

✓

取消确认

里程碑

在里程碑设置页面右上角单击 **新建里程碑**，然后填写相关信息，完成后单击 **确认**

新建里程碑 ×

*** 标题**

请输入标题

*** 期望完成日期**

请选择期望完成日期 📅

*** 负责人**

请输入负责人 ▼

描述

请填写里程碑描述

取消确认

4. 持续交付

4.1. 什么是持续交付

基于 Gitflow 工作流的最佳实践，通过组件灵活编排，轻松实现代码扫描、代码评审、自动化测试、自动编译部署等核心功能，快速定位问题，更快速完成发布更新，以持续交付实践不断提高研发效率。

优势

- **可扩展**
灵活的插件式架构，可轻松扩展流水线，简化持续集成的配置，具备多平台支撑与多语言支撑能力，减少对持续集成的管理。同时，提供对测试和构建的图像化的支持，使持续交付流动状态一目了然。
- **可编排**
在分布式计算中，组件灵活可编排，可以支持各种规模的动态部署与服务。通过检测容器状态，分析整体数据和资源状况，合理分配和支持不同业务、不同规模的产品服务，实现高效的持续交付。
- **多样化**
多平台支撑，支持多种分支策略，支持多种技术栈，支持多种开发语言。

功能

持续交付封装了以下核心功能模块，覆盖软件研发的各个阶段：

- **研发迭代**
支持更为灵活强大的分支管理策略，各环节均可进行自动代码扫描、代码评审、自动化测试以及自动编译部署，无论是分支切换，还是新增、删除，都有质检环节卡点，通过有效的自动化风险过滤，使项目分支随时处于高质量的稳定状态。
- **代码服务**
采用支撑百万用户规模的 GitLab 架构，进一步提升性能、确保数据安全、实现单元化部署，形成更安全、稳定、高效、智能、集成、开放的代码生态系统。多样化分支策略支持高扩展、分布式、响应式、大文件、按需下载，更有流控、安全、数据化、单元化等诸多强大提升。
- **研发环境**
提供高性能的容器集群管理方案，集中统一管理配置项，只需使用简单的配置文件与命令，即能一键式自动同步，快速适配到多个环境。以镜像作为核心，在研发流中进行高效流转，镜像状态迁移驱动持续交付过程，轻松实现基于容器的持续交付实践。
- **流程中心**
流程规则可按业务需求灵活配置，也可根据软件开发协作状况自由定制，支持规则自定义、加权自定义，支持多成员协作，具有强大的适应性。
- **管理员配置**
为管理员提供配置部署环境、迭代、流水线、构建的统一入口。

4.2. 工作台

工作台是持续交付的总览页面，也是用户查询和创建迭代、发布、应用、任务的快速入口。

工作台主要分为以下四块：

- **最近参与的研发迭代**：展示当前用户近期所参与的研发迭代，用户可查看当前工作空间下的全部研发迭代，并新建迭代。
- **最近参与的发布**：展示当前用户近期所参与的发布，用户可查看当前工作空间下的全部发布，并新建发布。
- **最近常用的应用**：展示当前用户最近常用的应用，用户可查看当前工作空间下的全部应用，并新建应用。

- 待我处理的任务：全量展示平台中待当前用户处理的代码评审、合并请求、流程审批等任务。

4.3. 研发迭代

4.3.1. 应用管理

应用作为研发活动的载体，研发效能平台支持以应用的方式进行持续交付。本文主要介绍在研发效能平台中如何进行新建应用、导入应用以及配置应用信息的操作。

新建应用

说明

- 在新建应用前，您可以按照需求自定义创建架构域，详情请参见：[架构域管理](#)。
- LinkE 的应用信息和 [运维管理](#) 处的应用信息是同步的，如果您需要在 LinkE 中使用 [运维管理](#) 处的应用信息，在 [我的应用](#) 页面单击 [初始化](#) 即可。

1. 从左侧导航栏进入 [研发迭代](#) > [我的应用](#)。
2. 在应用管理页面，单击 [新建应用](#)。
3. 在新建应用页面，根据提示填写应用相关信息，包括应用名称、技术栈、应用等级、二级架构域、开发负责人、测试负责人，并绑定代码仓库。
4. 如当前环境下无代码仓库可选，可单击 [新建代码仓库](#) 添加。新建代码仓库的群组信息与二级架构域绑定，如果无法获取请联系架构师。可前往 [代码服务](#) > [群组](#) 设置集成服务、成员权限，查看变更记录等。具体操作，参见 [群组管理](#)。
5. 设置完毕后，单击 [提交](#) 完成创建。

搜索应用

1. 从左侧导航栏进入 [研发迭代](#) > [我的应用](#)。
2. 在应用管理页面，可按架构域、是否为应用负责人（如果要搜索的是您作为负责人的应用，单击 [我的应用](#) 即可）进行搜索，或者直接在应用搜索框中输入应用名称、ID 搜索应用。

查看应用详情

应用详情由以下 7 部分内容组成：



- **迭代**：显示与该应用关联的迭代，分为进行中、已完成迭代。更多内容，参见 [研发迭代管理](#)。
- **元数据信息**：包括该应用的基本信息。部分元数据信息可针对研发需求进行修改。各项说明详见 [元数据详情](#)。
- **质量规则**：系统将根据应用质量规则在持续交付过程中计算质量分并判断应用质量是否达标。质量是否达标的评判标准则可以自定义。各项说明详见 [质量规则](#)。
- **操作日志**：显示针对该应用的信息变更。

- **待办任务**：显示与应用相关的待办任务，如紧急发布合并冲突。
- **持续集成**：可以在指定分支上设置定时任务执行持续集成。
- **研发配置**：可以自定义 Pipeline。具体参见 [流水线组件](#)。

应用负责人有权限修改以上所有属性。

本文仅对元数据信息和质量规则展开说明，其他内容不在此详述。

元数据详情

应用元数据详情由 **应用基础信息**、**研发设置**、**成员** 三部分组成。具体说明如下：

基础信息

- **应用名**：不可修改。
- **中文名**：不可修改。
- **编译技术栈**：决定构建方式。
- **架构域**：可以修改。

研发设置

- **窗口发布**：在迭代中必须关联一个日常发布窗口才能推进迭代阶段。
- **锁定项目分支**：锁定项目分支状态下通过 MR 触发 CI；非锁定项目分支下通过 MR/PUSH 触发 CI。
- **自动部署 Dev/Sit 服务器**：决定提交 MR 时自动部署按钮的状态，状态可在提交 MR 时变更。
- **发布完成同步沙箱环境**：开启后，当租户中有沙箱环境时，发布完成后沙箱环境会部署本应用。
- **主干提交代码必须关联需求或缺陷**：开启后，通过 MR 提交代码到主干时，必须关联需求或者缺陷。
- **创建迭代时必须关联需求**：开启后，创建迭代时必须关联需求。

成员

- 应用负责人
- 运维负责人
- 测试负责人
- 质量负责人
- 域架构师
- 迭代成员

质量规则

研发效能平台将根据应用质量规则在持续交付过程中计算质量分并判断应用质量是否达标。质量是否达标的评判标准可以由用户自己定义。

② 说明

- LinkE 公有云环境支持邮件通知功能，如果应用负责人在 **成员与角色** 中配置了邮箱地址，当该应用的质量规则发生改变后，应用负责人会收到质量规则变化的说明邮件。
- LinkE 专有云环境邮件通知功能需要开通 SMTP 服务，详情请参见 [SMTP 服务](#)。

开启非下降质量分

应用采用租户默认的质量规则检查

指标项	指标	说明
质量分	大于等于 <input type="text" value="40"/>	质量分计算规则请见 帮助文档
PMD-1	小于等于 <input type="text" value="0"/>	PMD规则block问题, 推荐值0
PMD-2	小于等于 <input type="text" value="0"/>	PMD规则critical问题, 推荐值0
安全问题	小于等于 <input type="text"/>	STC扫描问题, 默认为0, 如需修改, 请找守容、童彦、问何
行覆盖率	大于等于 <input type="text" value="80"/>	包括手工测试, 单元测试, 集成测试
变更行覆盖率	大于等于 <input type="text" value="90"/>	变更代码行的覆盖率
变更行覆盖率(自动)	大于等于 <input type="text"/>	如果无集成测试以单元测试为准
接口注释率	大于等于 <input type="text" value="0"/>	接口注释率
代码重复度	小于等于 <input type="text"/>	重复行数/总行数
自动化执行时间	小于等于 <input type="text"/>	集成测试自动化执行时间, 单位毫秒
行覆盖率(自动)	大于等于 <input type="text"/>	如果无集成测试以单元测试为准
分支覆盖率(自动)	大于等于 <input type="text"/>	如果无集成测试以单元测试为准
测试通过率	大于等于 <input type="text"/>	自动化测试通过率, 包括单元测试、集成测试
集成测试通过率	大于等于 <input type="text" value="100"/>	集成测试通过率
单元测试通过率	大于等于 <input type="text"/>	单元测试通过率
精准通过率(精准)	大于等于 <input type="text"/>	精准通过率(精准)
变更覆盖率(精准)	大于等于 <input type="text"/>	变更覆盖率(精准)
分支覆盖率(全量)	大于等于 <input type="text"/>	分支覆盖率(全量)

后续操作

当您完成应用的创建后, 就可以针对该应用进行研发迭代工作, 详情请参见 [研发迭代管理](#)。

4.3.2. 研发迭代管理

开发者在研发迭代中完成代码提交、代码扫描、集成测试、构建、部署、提交发布等操作。

在开始软件开发之前, 需要先简单了解一下 [迭代类型](#) 及迭代流程。一个完整的迭代流程主要包括以下三大环节:

- 新建迭代。**
以标准迭代为例, 迭代创建完成后进入开发阶段。迭代创建后, 可 [查看迭代详情](#)。
- 针对迭代关联的应用, 完成应用的 [提交代码变更](#)、[提交配置变更](#)、[新建应用服务和申请部署资源](#) 等操作。
- 推进迭代** 到下一阶段。
若一个迭代下有多个应用, 可以通过迭代整体推进把该迭代下的所有应用推进到下一阶段。当迭代推动到发布阶段并发布成功 (详见 [发布管理](#)), 迭代完成。

在迭代过程中，可以修改迭代信息，还可以 **完结和废弃迭代**。

迭代类型

迭代类型可根据租户需求进行定制。通常情况下，在租户初始化时会定义 2 种迭代类型：标准迭代和紧急发布。

- 标准迭代：主要是用于常规的开发发布。标准迭代一般包含开发阶段、集成阶段、预发阶段、发布阶段四个阶段，采用分支开发主干发布的发布策略。
- 紧急发布：主要用于常规发布外的紧急发布。紧急发布包含开发阶段、预发阶段、发布阶段三个阶段，采用分支开发分支发布的分支策略。

说明

迭代类型及包含的阶段都可以在管理员配置中修改和定义。

新建迭代

1. 通过以下任一入口新建迭代：

- 入口 1：在 **持续交付 > 工作台 > 最近参与的迭代** 区域，单击 **新建迭代**，选择迭代类型。
- 入口 2：进入 **持续交付 > 研发迭代 > 我的迭代**，单击 **新建迭代**，选择迭代类型。

2. 在新建迭代页面，输入以下信息：

- **迭代名称**：填写迭代名称。
- **关联应用**：填写涉及的应用名，其中由“独立发布”应用创建的迭代可以不关联“发布窗口”就可以直接发布；由“日常发布”应用创建的迭代必须关联“发布窗口”才能发布。若无应用可选，可新建或导入应用。
- **对应分支**（当选定应用后会出现）：对应开发阶段进行研发使用的项目分支。可以自定义分支名和分支拉流使用的 Tag。
- **添加应用**：添加更多应用到迭代中。
- **关联项目**（选填）：关联 **项目协作** 中的项目。关联项目之前，确保 **项目协作** 组件已开通。
- **关联需求/缺陷**（选填）：关联项目中的工作项。
- **迭代负责人**：默认为迭代创建者，可增加其他成员。
- **迭代成员**：默认为迭代创建者，可增加其他成员。

说明

新建迭代完成后，将会以邮件的方式通知迭代负责人。

搜索迭代

从左侧导航栏进入 **持续交付 > 研发迭代 > 我的迭代**，即可查看迭代列表。

迭代的搜索方式包括简易搜索和高级搜索两种。

- 简易搜索：可根据您是否为迭代创建人、是否为迭代成员、迭代是否已经完成以及迭代名称关键字进行搜索。
- 高级搜索：搜索条件如下图所示。

我的迭代

迭代ID: <input type="text" value="请输入迭代ID"/>	迭代名称: <input type="text" value="请输入迭代关键字"/>	关联项目: <input type="text" value="请输入项目关键字"/>
Release: <input type="text" value="请输入发布关键字"/>	迭代创建人: <input type="text" value="请选择迭代创建人"/>	迭代成员: <input type="text" value="请选择迭代成员"/>
状态、阶段: <input type="text" value="请选择迭代状态、阶段"/>	发布时间: <input type="text" value="开始日期"/> ~ <input type="text" value="结束日期"/>	<input type="button" value="搜索"/> <input type="button" value="清空"/> <input type="button" value="简易搜索"/>

搜索结果包括迭代 ID、迭代名称、负责人、创建时间、关联发布、发布时间、关联项目、状态、阶段。

修改迭代信息

修改基本信息

1. 从左侧导航栏单击 **持续交付 > 研发迭代 > 我的迭代**，在迭代列表中，单击迭代名称进入迭代详情页面。
2. 单击顶部导航栏上的 **设置** 标签。
3. 在设置页面修改迭代信息、添加迭代成员、添加删除应用等。

添加应用

可通过以下任一入口新增关联应用：

- 在迭代详情页面，单击顶部导航栏上的**当前应用名 > 添加应用**。



- 在迭代详情页面，单击顶部导航栏上的**设置**图标，单击设置页面 **应用管理** 栏中的 **添加应用**。

删除应用

在迭代详情页面，单击顶部导航栏上的**设置**图标，单击设置页面 **应用管理** 栏中的 **删除**。



查看迭代详情

研发迭代创建完成后，在迭代列表中，单击迭代名称即可进入迭代内应用的研发活动页面。



提交代码变更

开发阶段代码提交有两种方式：

- **集中工作流方式（Push 模式）**：所有修改都直接提交到目标分支上。首先克隆仓库，然后在工作副本中编辑和提交修改，修改保存在本地。当修改完成后，将本地分支的修改推送到远程仓库，推送后触发 Pipeline。
- **分支工作流方式（MR 模式）**：该模式以集中工作流为基础，不同的是为各个新功能分配一个专门的分支来开发。这样可以在把新功能集成到正式项目前，用 MR 的方式触发 Pipeline 完成代码审核及验证工作。

在开发阶段，平台默认支持的是 Push 模式，如果要改成 MR 模式，则需要应用元数据中设置（仅应用负责人和架构师有权限修改）。如果应用开启了 锁定项目分支，新建迭代拉流后，系统会自动锁项目分支，阻止直接 Push 代码到项目分支。

在进入集成阶段以后，平台提供的是 MR 模式，如果要修改主干代码，只能通过 MR 模式向主干提交。操作步骤如下：

1. 在页面上方的常用操作栏中单击 **提交 MR**。
2. 在新建 MR 页面，输入以下信息：
 - 源分支：要发布到远程的个人开发分支。
 - 目标分支：不可变更。
 - 标题：输入 MR 标题。
 - 描述：输入 MR 描述。
 - 需要评审：若选择 **是**，还需要填写评审人、评审通过票数。

提交配置变更

单击 **配置变更**。当完成配置变更后，请按需手动触发 Pipeline，详情请参见：[配置变更](#)。

新建应用服务和申请部署资源

可通过新建或导入应用服务，申请不同研发环境的部署资源，通过手动触发 Pipeline 将最新的代码部署完成后测试。操作步骤如下：

1. 在迭代详情页上方的常用操作栏中，单击 **新建应用服务**。
2. 在新建应用服务页面，输入以下信息：
 - **环境（必填）**：对应代码部署环境，包括开发、集成等线下环境。
 - **服务器信息（必填）**：填写应用、技术栈版本等信息，选择可用服务器，然后单击 **新建**。

说明

- 如果未申请部署资源、pipeline 执行到部署组件时，会自动跳过。
- 如果未申请部署资源、pipeline 执行到部署组件时，会自动跳过。如果已经在 **运维管理** 中完成开发、测试部署资源的创建并已分配给应用，可通过 **研发环境 > 应用服务 > 导入应用服务** 将应用服务和部署资源自助导入 **持续交付** 组件。**持续交付** 暂不支持创建 AKS 容器应用服务，只能自助导入。
- **持续交付** 组件中只能管理开发、测试等线下环境的应用服务，预发、生产等线上环境的应用服务管理请在 **运维管理** 组件中操作。

部署

在迭代详情页上方的常用操作栏中，单击 **触发 Pipeline > 部署当前环境**，即可触发编译部署 Pipeline 流水线运行。



推进迭代

阶段推进

迭代一般会配置几个阶段，以默认的标准迭代为例，有开发阶段 > 集成阶段 > 预发布阶段 > 发布阶段，通过单击操作栏中的完成某操作阶段（如完成开发阶段）完成阶段推进。

在迭代的推进过程中，研发效能平台会展示出当前迭代的基本信息供推进时做决策。展现的内容包括：

- 质量门禁（代码质量检查，如果有质量数据未达标，标红提示加签审批）。
- 审批（审批流程状态检查）。
- 质量数据展示。

阶段推进检查结果

● 阶段推进检查成功

● 已通过门禁检查
共1项检查 已通过1项

> 质量门禁

审批

● 所有任务已完成

名称	处理人	状态
AKS迭代 测试验证-20200115		已完成

质量

质量分详情

质量分	PMD-1	PMD-2	代码行	测试通过率	集成测试通过率	接口注释率	代码重复度	行覆盖率	变更行覆盖率	自动化执行时间	quality.test_fail	quality.test_skip	quality.test_total
38.33	0	0	9	100.00	100.00	0.0	0.0	33.33	100.0	4m 21s	0	0	1

迭代整体推进

单击页面顶部的 迭代推进 标签，进入迭代推进视图。此视图的作用是方便迭代管理员和项目经理等角色进行迭代整体推进。如果需要单独推进某应用的阶段，请按上一章节操作。

迭代推进 视图顶部会显示迭代当前阶段。如果有多个应用，此处显示的是进度最慢的那个应用的所处阶段。

迭代推进视图展示以下内容：

- 应用信息列表，显示应用及所处阶段，处于当前阶段的应用阶段为蓝色，不处于当前阶段的应用阶段为灰色。默认选择全部应用，单击应用列表可切换当前的应用相关信息。
- 当前阶段检查的所有卡点概要信息，必须在相关阶段检查卡点都完成后才可整体推进阶段。

- 当前阶段检查的卡点的具体内容，包含所有待办任务、质量信息、以及其它阻塞迭代推进的事项。

当阶段检查卡点全部完成后，即可进行迭代整体推进，单击 **批量推进阶段** 按钮即可。迭代当前阶段下的所有应用都将会被推进到迭代的下一阶段，如当前阶段是开发阶段，此阶段的相关应用都会被推进到集成阶段。

完结和废弃迭代

当迭代推进到发布阶段并发布成功，即表示迭代完成。当有应用推进到发布阶段，迭代则不能废弃。如果代码已经合并到了主干，那么废弃迭代之后，系统不会自动从主干扣除代码，这些代码会在下次发布时一并发布掉。所以，如果废弃迭代的同时也不需要已经合并到主干的代码，需要自行对代码进行处理。在迭代详情页面的顶部操作栏中，单击 **设置 > 废弃迭代**。

4.3.3. 架构域管理

研发效能平台除了默认架构域外，还支持用户自行创建架构域来满足实际开发中的应用需求。

新建架构域

1. 登录 **研发效能控制台**。
2. 在左侧导航栏单击 **持续交付 > 研发迭代 > 我的架构域**。
3. 在 **我的架构域** 页面，单击 **新建**，配置架构域信息。

持续交付 / 研发迭代 / 我的架构域

← 新建架构域

✔ 新建应用的代码库分组来源于架构域中关联的代码库分组

*中文名称:

描述:

*域架构师(主):

域架构师(备):

*质量负责人(主):

质量负责人(备):

*架构域等级:

父架构域:

保存

参数说明如下：

参数	说明
中文名称	必填，输入架构域名称信息。
描述	选填，输入架构域描述信息。
域架构师（主）	必选，选择主架构师人员。
域架构师（备）	选填，选择备选架构师。
质量负责人（主）	必选，选择主质量负责人员。
质量负责人（备）	选填，选择备选质量负责人。
架构域等级	<p>必选，选择架构域等级。</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p>? 说明</p> <ul style="list-style-type: none"> ◦ 目前仅支持到二级架构域。 ◦ 只有等级为一级架构域才能够被选为父架构域。 ◦ 在创建应用时，只能选择二级架构域。 </div>
父架构域	当架构域等级选为二级架构域时，可以选择该架构域所属的一级架构域。

4. 完成后单击 **保存**，完成架构域的创建。

编辑架构域

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 **持续交付 > 研发迭代 > 我的架构域**。
3. 在架构域列表中，单击目标架构域右侧 **编辑**，配置架构域信息。

编辑架构域
✕

✔ 新建应用的代码库分组来源于架构域中关联的代码库分组

*中文名称:

描述:

*域架构师(主):

域架构师(备):

*质量负责人(主):

质量负责人(备):

*架构域等级:

? 说明
 信息配置同 [新建架构域](#)。

删除架构域

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 [持续交付](#) > [研发迭代](#) > [我的架构域](#)。
3. 在架构域列表中，单击目标架构域右侧 [删除](#)，即可删除架构域。

我的架构域

名称	等级	描述	域架构师	操作
ifhrmqax	一级架构域		son	编辑 删除
testrole	一级架构域		don	编辑 删除
测试	一级架构域		tan	编辑 删除
dudu	二级架构域		tan	编辑 删除

其他操作

完成架构域的创建后，您可以通过单击架构域名称来查看该架构域绑定的应用信息以及迭代信息。

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 [持续交付](#) > [研发迭代](#) > [我的架构域](#)。
3. 在架构域列表中，单击目标架构域名称。
4. 在架构域详情页查看其绑定的应用以及迭代信息。
 - 单击 [应用](#)，查看该架构域下绑定的应用信息，支持单击应用名称查看应用详情。



- 单击 **迭代**，查看该架构域下应用绑定的迭代信息，支持单击迭代名称查看迭代详情。



说明
在 **迭代** 页签下，您可以单击 **新建迭代** 进行创建迭代的操作，详情请参见 **研发迭代管理**。

4.3.4. 发布管理

通过本文，您将了解研发效能平台上的发布模式，以及如何创建、查看发布窗口，关联发布窗口与迭代。研发效能平台支持以下两种发布模式：

- 单迭代发布：迭代创建之后，会有一个默认的发单（快速发布）存在，如果不和其他迭代一同发布，则不需要再额外创建发布窗口（即窗口发布）。
- 发布窗口（窗口发布）：适用于多个迭代在同一个发单中进行发布。在这种场景下，将迭代和发布窗口进行关联即可。

新建发布窗口

- 从左侧导航栏进入 **研发迭代 > 我的发布**，单击 **新建发布**。
- 在新建发布窗口页面，输入以下信息：
 - 发布名称**：发布名称信息。
 - 发布接口人**：只有发布接口人才能对发布窗口实施提交发布。
 - 合并截至时间**：将开发阶段分支合并至主干的截至时间，若延期将会提示。
 - 提交预发截至时间**：进入预发阶段的截至时间，若延期将会提示。
 - 提交发布截至时间**：延期将会提示。
 - 预计发布时间**：预计发布的时间。
- 单击 **新建** 完成创建。

关联迭代

可通过以下两种方式关联迭代：

发布窗口关联迭代

- 从左侧导航栏进入 **研发迭代 > 我的发布**，在发布列表中，单击具体发布名称，进入发布详情页。
- 单击 **关联迭代** 标签下的 **关联迭代** 按钮。
- 选择要关联的迭代，单击 **确认** 按钮完成关联。

说明

- 您可以在 **关联迭代** 标签下，迭代列表的右侧单击 **踢出** 来解绑迭代和发布，如果迭代负责人在 **成员与角色** 中配置了邮箱地址，当解绑迭代后，迭代负责人会收到发布与迭代解绑的说明邮件。



- LinkE 专有云环境邮件通知功能需要开通 SMTP 服务，详情请参见 **SMTP 服务**。

迭代关联发布窗口

- 从左侧导航栏进入 **研发迭代 > 我的迭代**，在迭代列表中，单击具体迭代名称，进入迭代详情页。
- 在顶部导航栏上单击 **发布配置 标签 > 关联发布窗口**。
- 选择要关联的发布，单击 **确认** 按钮完成关联。

查看发布窗口

- 从左侧导航栏进入 **研发迭代 > 我的发布**，进入发布列表页面。
- 按发布关系人（**我参与的、我创建的、所有发布**）、发布状态（**所有状态、提交发布、发布中、发布完成**）、关键字（发布名称、发布 ID）进行发布搜索。
- 单击发布名称，即可进入发布窗口详情页面，该页面整体分为 6 个部分：
 - 发布信息**：包括发布状态、发布基础配置、发布依赖、发布应用清单。
 - 发布内容**：包括代码变更、jar 包。
 - 关联迭代**：发布窗口内关联的迭代信息。
 - 相关任务**：发布窗口内关联的任务信息，包括任务状态。
 - 应用信息**：发布窗口内包含的应用信息。

说明

- 此处的应用信息是该发布已关联迭代中的应用信息，您可以在发布中直接解绑应用，但是此操作不可逆，会同时在迭代中删除该应用，若需要再发布该应用，需要重新创建新的迭代进行绑定发布。
 - LinkE 公有云环境支持邮件通知功能，如果应用负责人在 **成员与角色** 中配置了邮箱地址，当解绑应用后，应用负责人会收到发布与应用解绑的说明邮件。
 - LinkE 专有云环境邮件通知功能需要开通 SMTP 服务，详情请参见 **SMTP 服务**。
- 报告**：发布窗口内应用的质量信息。
 - 操作日志**：按时间节点显示该发布中进行了哪些操作。



更新/取消发布

当迭代由于某种原因发布失败后，为防止发布单长期处于发布中的状态，LinkE 现支持在迭代详情页对 灰度 和 发布 两个阶段的迭代进行更新/取消发布的操作。

灰度阶段

当迭代推进至灰度阶段时，您可以通过单击 **更新发布** 或 **重置发布** 进行更新/取消发布的操作，不同的操作和使用场景说明如下：



- **重置发布**

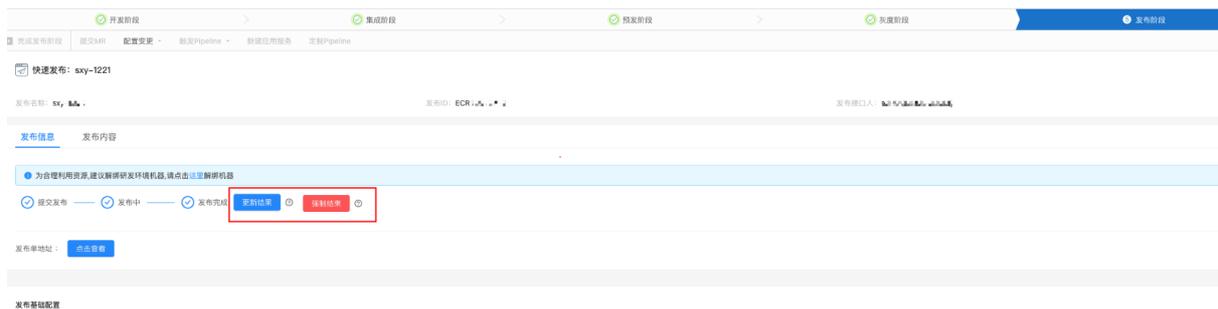
- 当您未单击发布时，如果单击 **更新发布**，系统会提示 **未提交发布**。
- 当您单击 **提交灰度发布** 后，再单击 **更新发布**，系统会提示您登录 PaaS 平台进行发布。
- 开始发布后，单击 **更新发布**，如果 PaaS 平台仍然在发布中，则反馈发布中，否则将发布单更新到完成。

- **重置发布**

单击 **重置发布** 后刷新页面，该迭代将退回至待提交发布阶段，可以再次提交灰度发布。

发布阶段

当迭代推进至发布阶段时，您可以通过单击 **更新结果** 或 **强制结束** 进行更新/取消发布的操作，不同的操作和使用场景说明如下：



更新结果

- 当您未单击发布时，如果单击更新结果，系统会提示未提交发布。
- 当您单击提交发布后，再单击更新结果，系统会提示您登录 PaaS 平台进行发布。
- 开始发布后，单击更新发布，如果 PaaS 平台仍然在发布中，则反馈发布中；如果 PaaS 平台已经发布完成，则反馈更新结束。

强制结束

单击强制结束后，迭代页面显示发布取消。

说明
发布结束和发布取消两种状态都代表该迭代已经完成。

生产发布说明

1. 将迭代推进到发布阶段后，如果发布下所有迭代及应用都已经到达发布阶段，会显示提交发布按钮。
2. 单击提交发布，系统会将发布单提交到经典发布运维中，并显示发布单地址。
3. 单击发布单地址后，可以跳转到经典发布运维平台进行发布。
4. 当检测到对应发布已经发布完成后，系统会自动结束发布阶段，迭代状态显示为发布完成。

说明
因为部分流水线组件如构建、filex 上传、部署等，都需要配置部署环境才能执行，所以在手动触发流水线前，您可以选择是否进行流水线预检，避免流水线因组件配置失误导致发布失败，如下图所示。



4.3.5. 配置变更

在迭代详情页面，用户可以对当前研发迭代进行配置变更。

技术栈配置

技术栈配置是从 SOFAShield 的应用服务同步获取，由于目前仅技术栈 2.0 提供查询接口，所以只有升级到技术栈 2.0 的应用才支持配置。服务器申请、预发部署和生产发布都需要用到技术栈配置。

操作步骤

1. 从左侧导航栏单击 **研发迭代 > 我的迭代**，在迭代列表中，单击迭代名称进入迭代详情页面。
2. 在页面顶部的常用操作栏中单击 **配置变更 > 软件/新应用配置**。
3. 如果需要修改技术栈版本，从版本下拉框中选择目标版本，然后单击 **保存**，完成配置。

更多关于技术栈的内容，参见 [技术栈管理](#) 相关文档。

技术栈配置生命周期

如果是在线下环境进行技术栈版本变更，变更的配置会在集成流水线进行发布部署时自动生效；如果是在线上环境进行变更，变更会随发布生效。

CI 配置

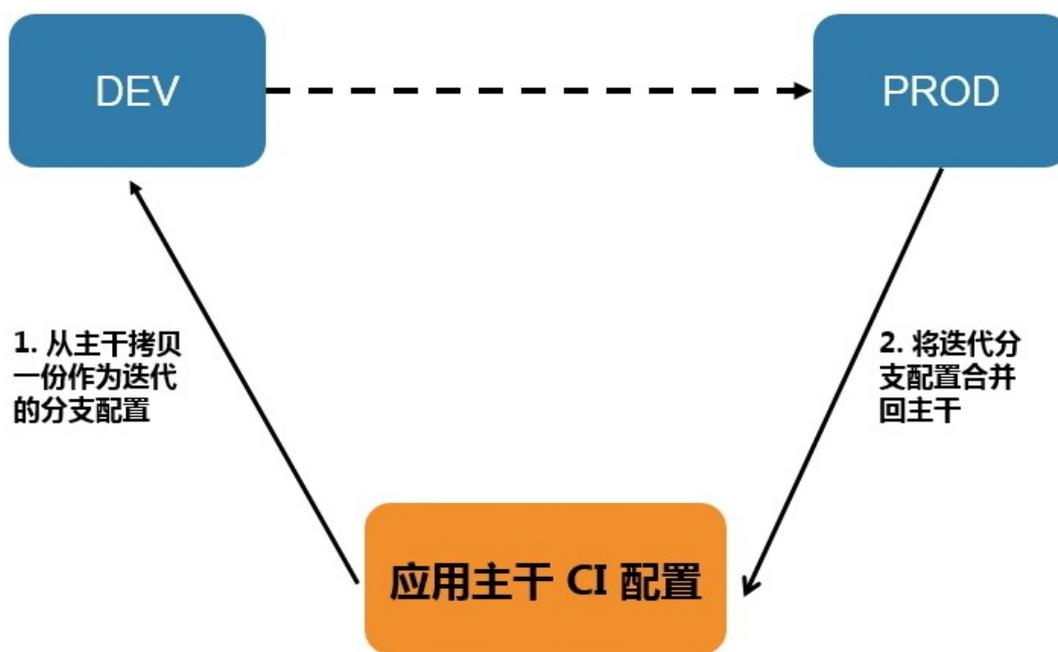
CI（持续集成）配置是在执行 CI 组件（比如 PMD 扫描，单元测试等）的时候需要的一些工作空间与参数配置，所有与 CI 相关的内容都在 CI 配置页面进行配置。

操作步骤

1. 从左侧导航栏单击 **研发迭代 > 我的迭代**，在迭代列表中，单击迭代名称进入迭代详情页面。
2. 在页面顶部的常用操作栏中单击 **配置变更 > CI 配置**，在配置页面中按需进行相应的配置。
3. 单击 **保存** 完成配置。

CI 配置生命周期

1. 当您新建迭代时，系统会从该主干配置中获取一份拷贝作为该迭代的分支配置。
2. 当该迭代提交发布后，系统会将迭代的分支配置合并回应用的主干配置，包括这次迭代对相关 CI 配置值的修改，从而使下一次迭代可以继承上次修改的内容。



常见问题

问题描述

新建了多个迭代，在一个迭代中修改 CI 配置，其他迭代也会重复修改或者没有自动同步。

问题分析

此问题的原因是没有做迭代隔离，如 [CI 配置](#) 章节所述，在并行迭代中，迭代彼此之间的 CI 配置是相互独立的，只有某一个迭代发布后配置内容合并回主干，那么再一次新建迭代才会基于新的主干内容。这样做是考虑到，如果全局各迭代共用 CI 配置，没有做迭代隔离，那么一个迭代升级 CE 版本，就会导致其他没有升级的迭代的配置也会受影响。

消息队列配置

在使用消息配置变更前，需要先了解 [消息](#)。

目前，消息变更支持：

- 消息类型变更（增加、删除、修改）
- 消息订阅变更（增加、删除、修改）

从左侧导航栏进入 [研发迭代 > 我的迭代](#)，在迭代列表中，单击迭代名称进入迭代详情页面。在页面顶部的常用操作栏中单击 [配置变更 > 消息配置](#)，进入消息配置页面。下面对消息配置的主要功能进行说明。

新增消息类型

1. 在 [消息类型配置](#) 下，单击 [新增](#)。
2. 在配置窗口中，输入 [消息主题](#)、[消息码](#)、[描述](#) 信息，然后单击 [保存](#) 即可。

修改或删除消息类型

1. 在 [消息类型配置](#) 下，单击 [修改/删除](#)，在配置窗口中，选择变更类型为 [修改](#) 或 [删除](#)。
2. 在 [检索条件](#) 输入框中，根据 [消息主题](#) 或 [消息码](#) 搜索并选择消息类型。
3. 选定的消息类型会出现在变更列表中，单击 [修改](#) 进行配置调整，或单击 [删除](#)，然后单击 [保存](#) 完成修改或删除。

新增消息订阅

1. 在消息订阅配置下，单击**新增**。
2. 在配置窗口中，输入**消息主题、消息码、组、订阅关系类型**信息，选择是否持久化，然后单击**保存**即可。

修改或删除消息订阅

1. 在消息订阅配置下，单击**修改/删除**，在配置窗口中，选择变更类型为**修改**或**删除**。
2. 勾选需要变更的订阅关系，选定的消息订阅会出现在变更列表中，单击**修改**进行配置调整，或单击**删除**，然后单击**保存**完成修改或删除。

同步和生效消息配置

单击消息配置页面右下角的**同步**，将消息类型和订阅变更同步到当前的工作空间，避免在研发效能平台的每个环境都重复添加。

同步操作只是在平台内部各环境间拷贝配置，并没有将配置生效到阿里云工作空间。如要使消息配置变更生效，单击**生效**即可。

生效结果展示

消息生效任务列表显示各环境（生产环境除外）消息配置变更的生效情况。生产环境配置会跟随发布单一起提交，其他环境消息配置支持单独生效，所以这里不展示生产环境的生效任务情况。

消息配置生命周期

- 非生产环境消息配置，由开发人员自主控制生效，平台提供生效按钮。
- 生产环境消息配置，随发布单一并提交，其中消息类型作为发布的前置任务，消息订阅作为发布的后置任务。

动态配置

在使用微服务-动态配置（DRM）配置变更前，需要先了解 **DRM**。

目前，研发效能平台仅支持**新增**和**修改** DRM 配置，不支持删除操作。

注意

- 首次使用前，建议仔细阅读页面右上方的提示内容。
- 填写 DRM 配置表单需符合各字段规范。资源标识需符合全限定类名规范；DRM 属性字段名需符合变量名规范。
- DRM 配置变更的只是元数据，不涉及线上环境中 DRM 值的变更，需要生效实际环境中的 DRM，需要进行 DRM 推送。

从左侧导航栏单击**研发送代 > 我的迭代**，在迭代列表中，单击迭代名称进入迭代详情页面。在页面顶部的常用操作栏中单击**配置变更 > DRM 配置**，进入 DRM 配置页面。

新增 DRM

1. 在 DRM 配置页面，单击**添加变更 > 新增添加**。
2. 在配置窗口中，输入**所属应用、资源域、资源标识、资源名称**等信息，然后单击**保存**即可。

修改 DRM

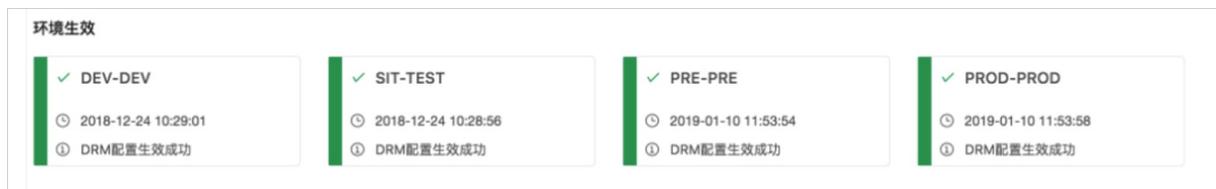
1. 在 DRM 配置页面，单击**添加变更 > 修改已有配置**。
2. 从生产环境的已有 DRM 列表中选择一条进行修改，后续操作同**新增 DRM**。

同步和生效 DRM 配置

单击 DRM 配置页面右下角的 **环境同步** 按钮，选择需要同步的环境。单击 **生效** 则会将 DRM 配置生效到阿里云环境。注意，此操作只是将 DRM 的元数据配置生效到 SOFAShield DRM 控制台。

生效结果展示

各环境的生效结果如下图所示：



DRM 配置生命周期

DRM 配置变更与研发流程关联。

支持在研发流程中随时提交 DRM 配置变更。所有的配置会保存到数据库中，如果需要生效到对应环境，可进行 **同步和生效 DRM 配置** 操作。

4.3.6. 流水线管理

本文主要介绍研发过程中流水线（Pipeline）的作用，介绍 Pipeline 的触发方式、相关操作及 CI 配置。

什么是 Pipeline

研发效能平台通过 Pipeline 来编排持续集成。

例如，研发人员提交了一次代码后，一条典型的 Pipeline 会帮助开发人员完成以下任务：

1. 代码扫描（PMD/FindBugs/STC）
2. 编译
3. 部署
4. 集成测试

以上的这些任务按照一定顺序执行，并且在一定条件下被触发。例如部署不可能在编译之前执行，直接在迭代中推送了代码到项目分支或针对项目分支提交了 MR 则会触发 Pipeline。

Pipeline 的触发方式

自动触发

代码提交会自动触发 Pipeline，代码提交包括：

- 提交 MR 到项目分支
- 直接推送代码到项目分支
研发效能平台支持提交 MR 和直接推送两种模式。平台默认支持 MR 模式。在新建迭代拉流后，系统会自动锁定项目分支，阻止直接推送代码到项目分支。如果需要支持推送模式，则需要应用元数据中进行设置（仅应用 Owner 和架构师有权限修改）。



- 提交主干代码：如果在应用进入到集成阶段后，需要修改主干代码，只能通过 MR 模式向主干提交。

手动触发

可以在配置变更完成后和其他需要的时机手动触发 Pipeline。

说明

因为部分流水线组件如构建、filex 上传、部署等，都需要配置部署环境才能执行，所以在手动触发流水线前，您可以选择是否进行流水线预检，避免流水线因组件配置失误导致执行失败，如下图所示。



Pipeline 的操作

取消和重新执行

- 取消：可随时取消执行中的 Pipeline。
- 重新执行：可重新执行处于暂停状态或者失败状态的 Pipeline。

跳过组件

如果 Pipeline 组件执行失败，您可以临时跳过该组件，继续执行后续组件。鼠标悬停在 Pipeline 组件上方，点击弹出的 **跳过** 链接即可。



Pipeline 相关的 CI 配置

Pipeline 在执行的过程中依赖一些 CI（持续集成）配置，这些配置不同于发布时的配置，仅用于 CI 阶段。例如，运行测试用例时的 MVN 版本，统计 PMD 及覆盖率问题时的排除类，部署开发服务器时是否打开调试端口。这些配置都可以在迭代详情页面的 **配置变更 > CI 配置** 中修改。

4.3.7. 流水线组件

持续交付支持的 Pipeline 组件包括冲突检测、缺陷查找、代码检测、代码合并、编译、功能测试、部署、交付内容传输等组件。

冲突检测

冲突检测组件用于检测 MR 是否存在合并冲突。

问题处理

如果发生冲突导致合并失败，可尝试先合并一次目标分支，例如当从 source_branch 合并到 target_branch 时产生冲突，可尝试使用以下方法解决：

```
git fetch #更新代码
git checkout source_branch #这里的 source_branch 指来源分支，根据实际替换
git merge target_branch #合并目标分支，并解决冲突，这里的 target_branch 指目标分支，根据实际替换
git push origin source_branch #合并后的代码推送到远端
```



缺陷查找 (Findbugs)

Findbugs 组件用于扫描代码中的缺陷。

问题处理

- 如果扫描失败，会显示出日志链接，可根据日志内容来进行问题排查。
- 针对扫描出的问题，需要根据具体内容进行解决。

说明
流水线配置先已支持连接 Jenkins 组件和 Pinpoint 代码扫描组件。

代码检测 (PMD)

PMD (Programming Mistake Detector) 组件提供代码扫描功能，支持查看扫描结果详情。

问题处理

- 根据扫描结果进行代码修复，解决对应的问题。
- 对于不需要扫描的代码，提前在 CI 配置页面进行配置。



代码合并 (PMD)

代码合并组件用于合并代码。

问题处理

如果合并失败，需要检查代码是否有冲突等情况。



编译

编译组件提供代码编译功能，从代码编译出可以部署的包或图片。

问题处理

如果编译失败，会显示出日志链接，可根据日志内容来进行问题排查。

The table shows the build and deployment process:

- 编译 (Build):** 编译 (Build) with a green checkmark and 00:05:18 duration.
- 部署 (Deploy):** 部署 (Deploy) with a green checkmark and 00:00:05 duration.

 Below is a table of build records:

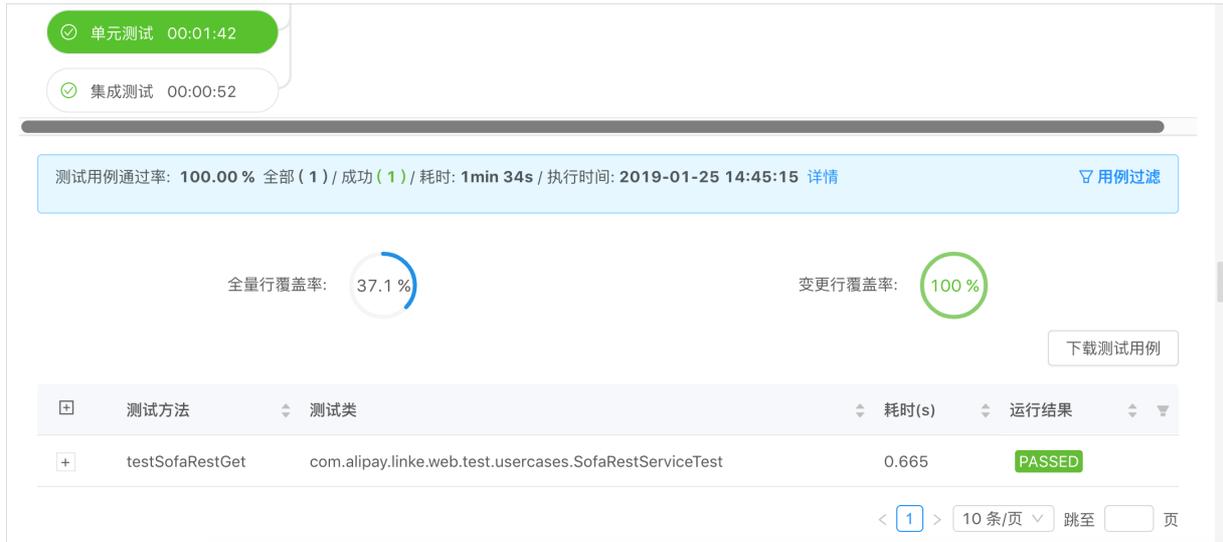
执行时间	耗时	Package	编译日志	编译状态
2019/1/25 下午2:43:27	5min 17s	imageId: imageSha: undefined imageTag: undefined imageUrl: undefined	日志详情	Success

功能测试

功能测试组件一般分为单元测试和集成测试组件。

问题处理

- 如果测试失败，会显示出日志链接，可根据日志内容来进行问题排查。
- 如果是环境原因导致用例测试失败，可以尝试 **重跑失败用例** 功能。



部署

部署组件可以将编译组件的产出部署到目标机器上。

问题处理

如果部署失败，会显示出日志链接，可根据日志内容来进行问题排查。

交付内容传输（Filex）

Filex 组件用来传输交付物。

问题处理

当传输失败时，可尝试重试以解决问题。

Jenkins 组件

Jenkins 是基于 Java 开发的一种持续集成工具，可以满足用户的定制化需求。LinkE 现支持自定义 Jenkins 组件的添加，用户只需要通过与 Jenkins 联调，就可以将自有平台的能力集成到 LinkE 中，减轻调试的成本。

操作步骤

1. 在 Jenkins 集群中配置好对应的任务，如下图：



配置完成后可以先自行运行一遍以确保配置无误。可参见 [Jenkins Pipeline 示例](#)。

The screenshot shows the Jenkins web interface for a pipeline named 'testjob'. The breadcrumb path is 'Jenkins > All > testjob >'. On the left sidebar, the 'Build with Parameters' button is highlighted with a red box. The main header area displays 'Pipeline testjob' with 'testjob' boxed in red. Below this, the text '运行 Jenkins Pipeline 任务' and 'Jenkins JobName' is visible. A red box highlights the '上次成功的成品' (Last successful build) section, which lists artifacts 'result.json' (93 B) and 'stdout.log' (16 B), and a '最近变更' (Recent changes) section with the text '任务执行结果产物' (Task execution result artifact). The 'Build History' section shows a table with one entry: '#1' at '2021-7-1 下午3:14', with the text '任务执行记录: BuidNumber' next to it. The 'Stage View' section shows a 'Declarative: Tool Install' stage with an average time of 50ms. The '相关链接' (Related links) section contains four links: 'Last build(#1), 8.6 秒之前', 'Last stable build(#1), 8.6 秒之前', 'Last successful build(#1), 8.6 秒之前', and 'Last completed build(#1), 8.6 秒之前'.

2. 在 LinkE 中配置含有 Jenkins 组件的流水线模板。具体步骤请参见 [新建流水线模板](#)。

自定义组件 ✕

基本信息

分组 名称

测试 v ✔ []

描述

请描述组件的主要功能和用途

类型

Jenkins 组件 v

Jenkins 账号

Jenkins Master URL 地址

[]

用户名 密码

[] [] 👁

选择 Jenkins Job

Job Name Build Number

[] 留空则为 NextBuildNumber

执行配置

超时配置

10 分钟

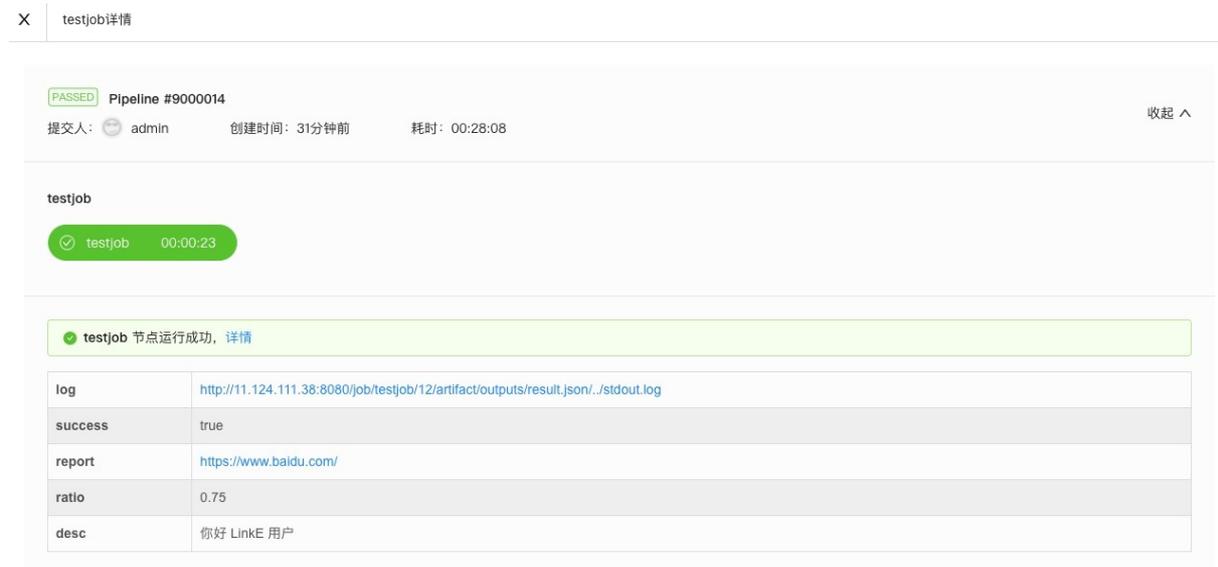
参数说明如下：

参数	说明
分组	选择自定义组件所属分组，目前仅支持代码、扫描、测试、构建和部署五种分组。
名称	输入自定义组件的名称信息。
描述	输入自定义组件的描述信息。

参数	说明
类型	选择自定义组件的类型，目前仅支持 Jenkins 组件，且无法修改。
Jenkins 账号	输入 Jenkins Master URL 地址信息，以及对应的用户名和密码。
选择 Jenkins Job	输入 Jenkins 集群中已经创建好的 Job 和对应的 Build Number（版本号）。
超时配置	配置 Jenkins 超时时间，默认为 10 分钟。
是否检查执行结果数据	选择是否检查执行结果数据，如果选中，Jenkins 组件任务运行成功的标志不只是 Jenkins Job 正常运行完成，还需要解析相应的结果文件。
结果文件	当 是否检查执行结果数据 状态为选中时，需要配置结果文件以验证 Jenkins 组件任务是否运行成功。

注意
新配置的组件会于 1 分钟后生效，请不要立即执行包含该组件的流水线。

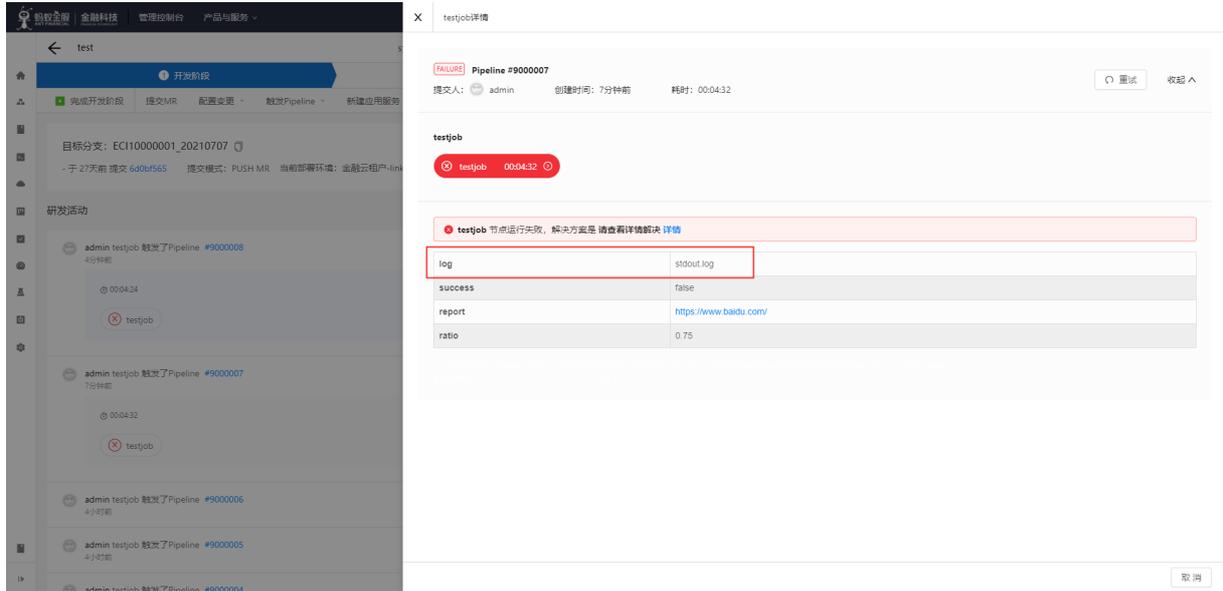
新建完成后，可以在迭代中新增并运行包含 Jenkins 组件的流水线，运行成功后如下图所示：



登录相应的 Jenkins 集群可以看到运行记录。



在配置流水线模板时，若选中 **是否检查执行结果数据**，那么该流水线组件在运行完毕后，还会检查相应路径下的 json 文件，如果不符合预期 LinkE会报错并阻塞流水线的继续运行，如下图：



Jenkins Pipeline 示例

Jenkins Pipeline 实质上是一个 Groovy 脚本：

```
import groovy.json.JsonBuilder

// 可使用 Groovy 语法定义方法

void printJavaInfo() {
    def userDir = System.getProperty("user.dir")
    println("userDir: ${userDir}")
}

void dumpJson(dir, jsonFile) {
    boolean success = true
    // boolean success = false // 设置运行成功检测标志位，测试失败场景
    // Thread.sleep(31 * 1000L) // 休眠 31 秒，测试超时场景

    // others 为自定义 KV 集
    def resultMap = ["success": success,
                    "report": "https://www.baidu.com/",
                    "log": "stdout.log",
                    "others": ["ratio": 0.75, "desc": "你好 LinkE 用户"]]

    def builder = new JsonBuilder(resultMap)
    def jsonStr = builder.toString()
    def jsonFilePath = dir + File.separatorChar + jsonFile
    def newFile = new File(jsonFilePath)
    newFile.write(jsonStr)
}

def resultJsonFile = "./outputs/result.json"
def resultLogFile = "./outputs/stdout.log"
```

```

pipeline {
  agent any

  // 定义 Jenkins 任务运行参数
  parameters {
    string(name: 'ACI_TIMEOUT', defaultValue: '10', description: '超时时间 (分钟)')
    string(name: 'GIT_REPO', defaultValue: '', description: '代码仓库 HTTP 地址')
    string(name: 'LAST_COMMIT', defaultValue: '', description: '最后提交 commit_id')
    string(name: 'GIT_REVISION', defaultValue: '', description: '代码 revision 信息')
    string(name: 'GIT_BRANCH_NAME', defaultValue: '', description: '代码分支名')
    string(name: 'GIT_BRANCH', defaultValue: '', description: '代码 revision / 分支名')
    string(name: 'LINKE_APP', defaultValue: '', description: '应用名称')
    string(name: 'LINKE_ITERATION', defaultValue: '', description: '迭代名称')
    string(name: 'LINKE_TENANT', defaultValue: '', description: '租户名称')
    string(name: 'ACI_JOB_ID', defaultValue: '', description: 'LinkE-aci 模块任务 ID')
    string(name: 'ACI_JOB_NAME', defaultValue: '', description: 'LinkE-aci 任务名称')
    string(name: 'ACI_TAGS', defaultValue: '', description: 'LinkE-aci 任务标签')
    string(name: 'CMD', defaultValue: 'hostname && date +%Y%m%d-%H%M%S', description: '自定义参数 run command')
  }

  options {
    timeout(time: 10, unit: 'MINUTES')
  }

  tools {
    jdk 'jdk8'
    maven '3.2.5'
  }

  stages {

    stage('beforeScript') {
      steps {
        echo '----- beforeScript Start to Excecute -----'
        sh 'mkdir -p ./output'
        echo "ACI_TIMEOUT ${params.ACI_TIMEOUT}"
        echo "GIT_REPO ${params.GIT_REPO}"
        echo "LAST_COMMIT ${params.LAST_COMMIT}"
        echo "GIT_REVISION ${params.GIT_REVISION}"
        echo "GIT_BRANCH_NAME ${params.GIT_BRANCH_NAME}"
        echo "GIT_BRANCH ${params.GIT_BRANCH}"
        echo "LINKE_APP ${params.LINKE_APP}"
        echo "LINKE_ITERATION ${params.LINKE_ITERATION}"
        echo "LINKE_TENANT ${params.LINKE_TENANT}"
        echo "ACI_JOB_ID ${params.ACI_JOB_ID}"
        echo "ACI_JOB_NAME ${params.ACI_JOB_NAME}"
        echo "ACI_TAGS ${params.ACI_TAGS}"
        echo '----- beforeScript End to Excecute -----'
      }
    }

    stage('script') {
      steps {

```

```
echo '----- script Start to Excecute -----'

// 如果探测到上次生成的产物文件夹，则先将其备份
sh '''
    now=$(date +%Y%m%d-%H%M%S)
    echo "now: ${now}, current pwd: $(pwd)"
    if [[ -d ./outputs ]]
    then
        mv ./outputs ./outputs_${now}
        echo "Last build outputs dir has backup to dir outputs_${now}"
    fi
    mkdir -p ./outputs
'''

// 运行预定义方法
printJavaInfo()
dumpJson("${workspace}", resultJsonFile)
sh "${params.CMD} > ${resultLogFile}"

echo '----- script End to Excecute -----'
}
}

stage('afterScript') {
    steps {
        echo '----- afterScript Start to Excecute -----'
        sh params.CMD
        echo '----- afterScript End to Excecute -----'
    }
}

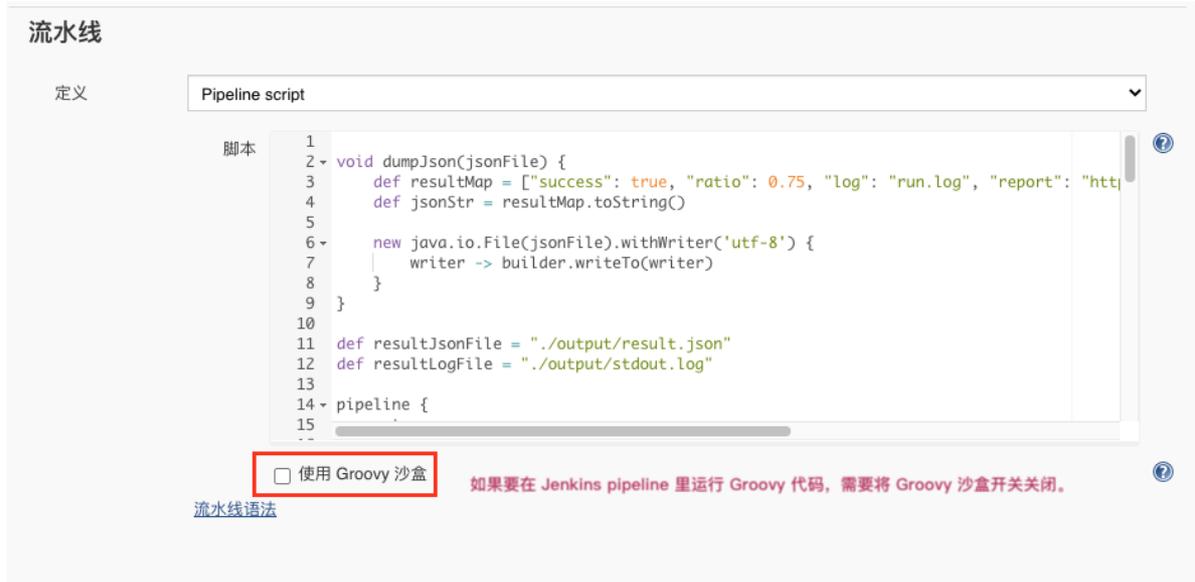
}

post {
    always {
        // 产物收集文件夹
        archiveArtifacts allowEmptyArchive: true, caseSensitive: true, onlyIfSuccessful: false, fingerprint: false, defaultExcludes: true, artifacts: 'outputs/'
    }
}
}
```

如上 pipeline 脚本如示，其定义了 Jenkins 任务的运行参数（LinkE 在执行任务时会自动配置），也给出 Jenkins 任务结果路径 out put /result.json 文件的生成方法。

注意事项

- Jenkins Groovy 脚本的运行配置
在 Jenkins 集群中进行 Groovy 脚本配置时，需要取消 Groovy 沙盒的勾选。



- Jenkins 的系统级访问配置
 - 组件配置页面指定的用户，需要有权限运行指定 Jenkins 任务。
 - Jenkins 系统配置 Jenkins Location 中，需要确认 Jenkins Url 配置正确：

Jenkins Location

Jenkins URL

系统管理员邮件地址

Shell

Shell executable

- 关闭 Jenkins 跨站保护。
需要关闭高版本的 Jenkin 中默认开启的 CSRF 跨站访问保护（关键配置项 jenkins crumb）。

说明
该配置项无法通过配置页面关闭，需要重新启动 Jenkins。

常见问题

LinkE 如何查询并启动 Jenkins 自定义任务？

通过 `JobName` 匹配。需要注意在 Jenkins 中，`JobName` 是任务的唯一索引，不能重名。

如何多次运行 Jenkins 自定义任务并获取结果？

Jenkins 提供 `BuildNumber` 参数，用于版本控制。第 1 次任务启动时其值为 1，后续每再启动一次，`BuildNumber` 的数值加 1。LinkE 会将保存每次自定义组件任务所关联的 JobName 与 BuildNumber 信息，并通过这两个参数查询对应 Jenkins CI 任务是否运行结果。

Jenkins 任务的运行参数

LinkE 在启动自定义 Jenkins 任务时，会为任务配置如下参数，包括应用名称、迭代信息、代码库的相关参数。

Jenkins Job 参数	说明	默认值	示例
ACI_TIMEOUT	超时时间（分钟）	10	10
GIT_REPO	代码仓库 HTTP 地址	空	https://code.sofa4finance.aliyuncs.com/opglpayg/225715210075827836/sxy-aks-ite.git
LINKE_APP	应用名称	空	jzh-test
LINKE_ITERATION	迭代名称	空	ECI10011688
LINKE_TENANT	租户名称	空	OPGLPAYG
LAST_COMMIT	最后一次提交 commit_id	空	e2358bc1dd936d49ff84d746bdf6d8e0f2dc3f44
GIT_BRANCH_NAME	关联 Git 分支名	空	master
GIT_REVISION	关联 Git revision id	空	e2358bc1dd936d49ff84d746bdf6d8e0f2dc3f44
GIT_BRANCH	默认为 GIT_REVISION 值，若其为空则采用分支名	空	e2358bc1dd936d49ff84d746bdf6d8e0f2dc3f44
LAST_COMMIT	最后一次提交 commit_id	空	e2358bc1dd936d49ff84d746bdf6d8e0f2dc3f44
ACI_JOB_ID	LinkE-aci 模块任务 ID	空	60a7259fb27cef10fb160f7d
ACI_JOB_NAME	LinkE-aci 任务名称	空	testjob
ACI_TAGS	LinkE-aci 任务标签	空	v2

LinkE 如何对 Jenkins 任务运行结果进行检测

Jenkins 任务在运行时会产生一个产物文件夹 `outputs`，其中内包含一个 `result.json` 文件。LinkE 流水线在执行时，将尝试解析如下 `json key` 内容：

JSON KEY	类型	说明
success	boolean	Jenkins 任务结果是否符合通过标准。

JSON KEY	类型	说明
report	string	Jenkins 任务生成的报告 URL。
log	string	与 result.json 文件在同一个文件夹下的日志文件。

 说明

上表中最关键的 key 为 success。如果用户在配置时勾选了检测 Jenkins 组件的运行结果，那么若该值不为 true，LinkE 会阻塞流水线的运行。

下表展示了一个测试 Jenkins 任务的 `result.json` 文件结果，里面有三个用于解析的 key (success、report、log) 与两个自定义 key (ratio、desc)。

```
{
  "success": true,
  "report": "https://www.baidu.com/",
  "log": "stdout.log",
  "others": {
    "ratio": 0.75,
    "desc": "你好 LinkE 用户"
  }
}
```

4.4. 代码服务

4.4.1. 仓库管理

持续交付提供基于 Git 的代码托管服务，为此您可以创建并管理一个或多个代码仓库，用于代码托管。该代码服务具有以下特性：

- **隔离性**：独立站点式的访问体验，自由创建分组进行工程管理。
- **多人评审**：支持单人或多人模式的合并评审。

新建仓库

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，单击页面右上方的 **新建仓库** 按钮。
2. 在 **新建仓库** 页面填写以下信息：
 - **群组**：必填，表示仓库归属到哪个群组。选择一个已有群组，或输入一个群组名以创建一个新的群组。群组名只能由英文、数字、点、中横线、下划线组成，同时不得使用 admin、groups、dashboard、help、profile、u 等作为名称。
 - **仓库名称**：必填，具体的仓库名，不同分组下仓库可同名。仓库名只能由英文、数字、点、中横线、下划线组成，并不得以 .git、.wiki 结尾。
 - **描述**：可选，说明仓库的功能。
 - **是否公开**：若设置为公开，则整个租户成员均可访问；若不公开，则仅仓库或分组成员可访问。
 - **是否初始化仓库**：若选择初始化，则会创建一个包含默认分支 master 及一个初始提交的仓库，可直接开始创建分支等操作。若不初始化，则需要通过提交进行初始化。

- **默认编码**：选择默认编码，目前仅支持 UTF-8 和 GBK。

3. 单击 **新建** 按钮进行仓库创建，创建完成后的仓库将会展示在仓库列表中。

下载仓库

研发效能平台提供 HTTP 或 SSH 协议进行 `git clone` 代码下载。

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，单击进入代码仓库详情页面。
2. 单击页面右上方的 **Clone** 按钮。
3. 单击 SSH 或 HTTP 选择用于克隆的协议，然后查看下载地址。
4. 单击复制按钮，获取下载地址链接。

标签管理

标签作为一次发布的标识，通常可使用最新标签作为新建分支的起点。在后续的迭代发布过程中，您可以通过标签来拉取对应的代码仓库。

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，单击进入代码仓库详情页面。
2. 在页面上方，单击 **标签 > 新建标签** 按钮，填写新的标签名，并选择或输入创建来源，可以是其他分支或者版本。
3. （可选）填写标签的描述信息。
4. 单击 **新建** 后，标签创建完成。

🔍 说明

在标签列表页面，单击对应的 **删除** 图标，即可删除对应标签，此操作需要管理员及以上权限。

4.4.2. 分支管理

查看并管理仓库下的分支（branch），包括新建、删除、保护、清理、比较分支。

新建分支

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，单击进入代码仓库详情页面。
2. 在页面上方，单击 **分支 > 新建分支**，填写新的分支名，然后选择或输入创建来源，可以是其他分支、标签或者版本。
3. 单击 **新建** 后，分支创建完成。

删除分支

在分支列表页面，单击需要删除的分支所在行的 **删除** 图标，在弹出确认框中单击 **确认** 后，执行删除。

保护分支

保护分支设置，即将分支锁定，不允许分支删除及重写（force push）操作，仅 Master 及以上权限的用户可删除或重写被保护的分支。

要保护或解锁分支，您可在分支列表页面，单击目标分支所在行的 **保护分支** 开关修改状态。

清理分支

批量删除已合并到默认分支（未变更则为 master 分支）的分支。

1. 单击 **分支清理** 按钮，等待完成查找可清理分支。
2. 若存在可清理分支，则需要勾选要删除的分支记录。

3. 单击 **清理** 后，开始执行选中分支删除。

🔍 说明

对于保护分支，仅 Master 及以上权限的用户可以在解锁后进行删除。

比较分支/提交记录

您可以使用比较（compare）功能，进行两个分支或提交记录间可合并差异的对比。对比结果与使用命令行

```
git diff {ref1}...{ref2}
```

 所得的结果相同。

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，单击进入代码仓库详情页面。
2. 在页面上方，单击 **比较** 页签，选择或输入对比起点（左侧输入框）及对比终点（右侧输入框）。
3. 单击 **比较** 按钮，获取对比结果。
4. 单击输入框左侧的 **交换** 图标，交换对比起点及终点分支/版本。

🔍 说明

对比结果仅展示右侧分支可合并到左侧分支的提交记录及文件变更，并非两个分支的直接对比差异。

4.4.3. 合并请求

当您需要发起一个合并到目标分支的请求时，可通过页面新建合并请求，进行一个交互式的评审过程，并完成最终合并。

操作步骤

1. 在左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，单击进入代码仓库详情页面。
2. 在页面上方，单击 **合并请求 > 新建合并申请**，选择合并的 **来源分支** 及 **目标分支**，单击 **比较**，查看差异。若来源分支及目标分支没有差异，则无需提交。
3. 修改标题，默认使用来源分支作为标题，简单表述修改内容。
4. 在描述信息中，详细描述在分支上进行的变更（支持 Markdown 语法文本）。
5. 开启 **合并后删除来源分支** 开关。开启后，系统将在当前合并请求合并成功后，删除来源分支。
6. 开启 **压制提交** 开关。开启后，系统会将本次变更的所有提交记录，压缩为一个提交记录，合并到目标分支上。
7. 选择合并处理人（必填）。合并处理人负责整个合并请求的跟踪，仅当其有目标分支的写权限时，才可以单击 **合并** 按钮进行合并。
8. 开启 **评审** 开关。开启评审后，您需要选择当前合并请求的评审成员，并设置评审通过所需的票数，当评审通过票数满足条件时，合并处理人才能执行合并。
9. 单击 **提交** 按钮，发起本次合并请求。

4.4.4. 密钥管理

添加您的 SSH 公钥，或设置一个 HTTP 访问令牌，用于下载及提交代码。

SSH 公钥

您可以设置 SSH 密钥以建立电脑和 AntCode 之间的连接。

操作如下：

1. 在左侧导航中，单击 **代码服务 > 个人密钥设置**。

2. 选择子菜单 **SSH 公钥**，可在该页面添加个人公钥信息，以使用 SSH 协议进行交互（推荐）：

i. 使用 `cat ~/.ssh/id_rsa.pub` 查看一个本地已有公钥，或使用

```
ssh-keygen -t rsa -C "您的登录邮箱" 创建一个公私钥对。
```

ii. 将公钥的文本内容，粘贴至 **SSH-RSA 公钥内容** 输入框。

iii. 如果觉得默认读取的公钥名称不够清晰，可以在下方的 **公钥名称** 输入框中进行修改。

iv. 单击 **保存** 按钮，将公钥信息添加到密钥列表当中。

说明

若需要移出某个公钥，则可在密钥列表中，单击对应的 **删除** 图标进行移除。

访问令牌

您可为您使用的每个需要访问 Antcode API 的应用程序生成个人访问令牌。

操作如下：

1. 选择子菜单 **访问令牌**，根据页面提示选择相应的应用程序、失效日期，以及访问范围。
2. 设置完毕后，单击 **创建令牌**。生成的令牌将出现在 **有效访问令牌** 列表中。

私有令牌

您可以使用私有令牌以 HTTP 的方式使用 Git。

选择子菜单 **私有令牌**，单击 **重置私有令牌** 可重新生成令牌。

注意

请妥善保管以上个人密钥信息。

4.4.5. 权限/成员管理

查看并管理仓库或分组的成员，包括添加、删除成员，修改成员权限。

注意

管理仓库成员需要 Master 及以上权限，管理分组成员需要 Owner 权限。

管理成员

从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，点击进入代码仓库详情页面。在页面上方，单击 **权限** 标签进入成员及权限管理页面。

添加成员

在权限管理页面的 **用户** 输入框中，查找您期望加入到该仓库或分组下的成员，并选择需要赋予的角色。

修改成员权限

在成员列表中，通过修改下拉框的角色类型，修改对应成员权限。

删除成员

点击成员所对应的 **删除** 图标，将该成员移出。

角色及权限

每一级资源会继承上一级的权限，即分组成员拥有分组下所有仓库的对应权限；租户成员拥有租户下所有分组及仓库的对应权限。另外，当继承权限与当前权限不一致时，总是以最高的权限为准。研发效能平台中的角色及其对应权限如下：

- Role: ... 选权限

- reporter: 读权限
- Developer: 非保护分支写权限
- Master: 所有分支写权限及仓库设置管理权限
- Owner: 分组管理及仓库创建、删除权限

4.4.6. 群组管理

从左侧导航栏进入 **代码服务 > 群组**，可查看自己当前所在的群组列表，查看群组相关信息、新建群组、添加成员到群组。

查看群组详情

- **概览** 页签页展示群组动态，以及群组下的所有仓库。若无仓库，您可以单击 **新建仓库** 进行添加。具体操作，参见 **仓库管理**。
- **合并请求** 页签页展示当前群组下的合并请求，可根据请求的处理状态进行过滤。
- **权限** 页签页展示当前群组的成员列表，以及成员变更记录，可在该页面添加或删除成员。
- **设置** 页签页展示群组中已添加的一些代码库相关服务，以及用于合并请求及评论的标记。可自行添加服务和标记。

新建群组

1. 在群组页面，单击页面右上方的 **新建群组** 按钮，进入群组创建页面。
2. 输入 **群组路径** 和对应的描述，单击 **新建** 即可。

管理成员

1. 在群组列表中，单击群组路径进入群组详情页，然后单击 **权限** 页签，或直接单击目标群组右侧的成员图标，进入成员添加页面。
2. 在权限管理页面的 **用户** 输入框中，查找您期望加入到该分组下的成员，选择需要赋予其的角色，并设置权限过期时间，然后单击 **添加** 即可。

在群组成员列表中，单击成员所对应的 **删除** 图标，将该成员移出。

4.5. 研发环境

4.5.1. 管理应用服务

效能平台支持经典 ECS 和容器应用服务两种研发模式，统一通过 **应用服务** 来管理开发、集成等线下环境的 ECS 服务器和容器应用服务，并结合 Pipeline 流水线将最新的代码部署至应用服务实例上进行测试。本文介绍应用服务的新建、导入、迭代关联、删除等操作。

新建应用服务

1. 从左侧导航栏进入 **持续交付 > 研发环境 > 应用服务** 页面，单击 **新建应用服务**。

应用服务	迭代/分支	环境	应用	创建时间	修改时间	状态	操作
aks-linkw	迭代: EC110003259	prod[AKS]: dev	helloinke	2020-03-06 20:36:56	2020-03-06 20:41:55	运行中	☆ △ □
helloinke-pre		dev[AKS]:pre	helloinke	2020-01-14 17:40:23	2020-01-19 15:31:09	运行中	☆ △ □

- 2. 新建应用服务也可以在研发迭代详情页面完成：单击常用工具栏的 **新建应用服务**，或者单击顶部导航栏 **环境 > 应用服务列表**，再单击 **新建应用服务**。
- 3. 输入应用服务信息，单击 **新建** 后，可通过 **操作记录** 查看工单详情。



导入应用服务

如果用户在 **运维管理** 组件中已经完成应用服务实例的创建，则可以 **导入应用服务**，然后关联到研发迭代中进行使用。

- 从左侧导航栏进入 **研发环境 > 应用服务** 页面，单击 **导入应用服务**，选择后，单击 **提交** 执行导入操作。
- 导入应用服务也可以在研发迭代详情页面完成，单击顶部菜单栏 **环境 > 应用服务列表**，再单击 **导入应用服务** 进入导入页面，勾选要导入的应用服务，单击 **确定** 后执行导入操作。

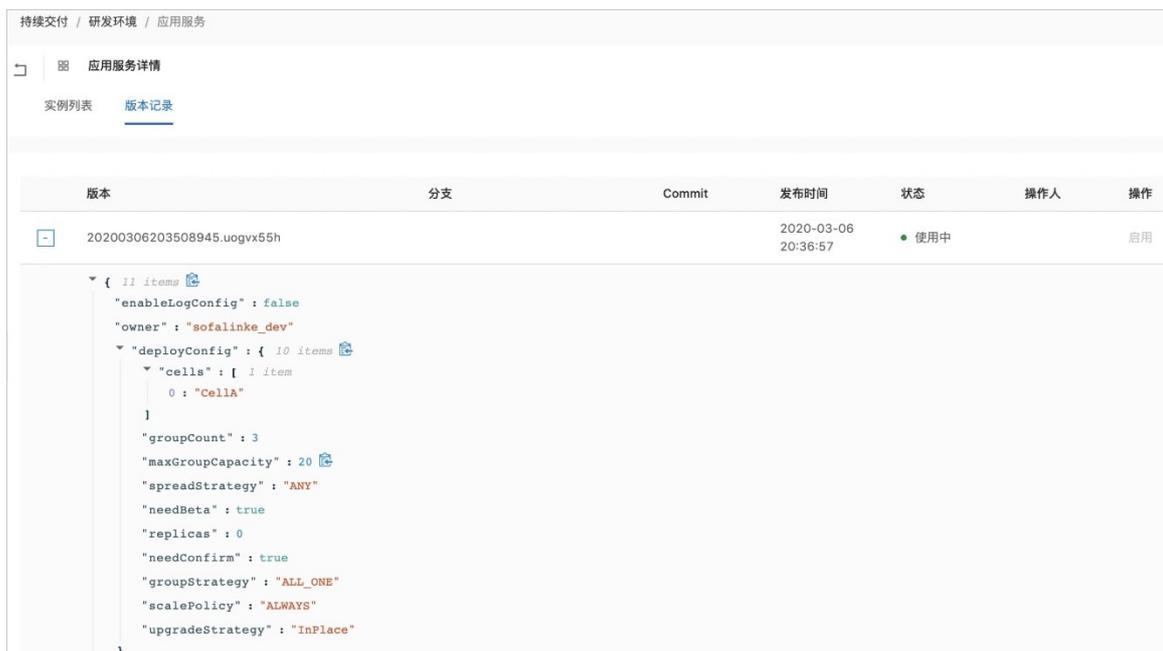


- 在开通效能平台后，推荐通过 **持续交付** 组件来管理（新建、删除）开发、集成等线下环境的应用服务实例。

查看容器应用服务

- 1. 从左侧导航栏单击 **持续交付 > 研发环境 > 应用服务** 进入容器应用服务列表页面，可根据环境和应用信息进行过滤筛选。
- 2. 应用服务列表展示了 **我关注的**应用服务和 **全部**应用服务，用户可以对应用服务进行关注操作，关注后的应用服务会出现在 **我关注的**应用服务 列表中。

3. 单击应用服务详情，可以看到应用服务对应的服务器资源以及版本记录。



应用服务关联迭代和解除关联

1. 新建或导入完成后，如果是开发环境的应用服务，通常需要在迭代详情页面 **环境 > 应用服务列表** 中执行 **关联迭代** 操作，确定在该迭代中开发阶段默认部署的应用服务实例，已支持并行开发和自测。
2. 如果在研发迭代页面中执行新建或导入应用服务后，平台会自动关联当前迭代。



3. 迭代发布后，应用服务自动取消迭代关联，进而可被其他迭代关联使用。

删除应用服务

从左侧导航栏单击 **持续交付 > 研发环境 > 应用服务** 进入容器应用服务列表页面，单击 **删除** 图标，将调用应用 PAAS 平台 **运维管理** 的接口执行应用服务删除操作。

4.5.2. 查看操作记录

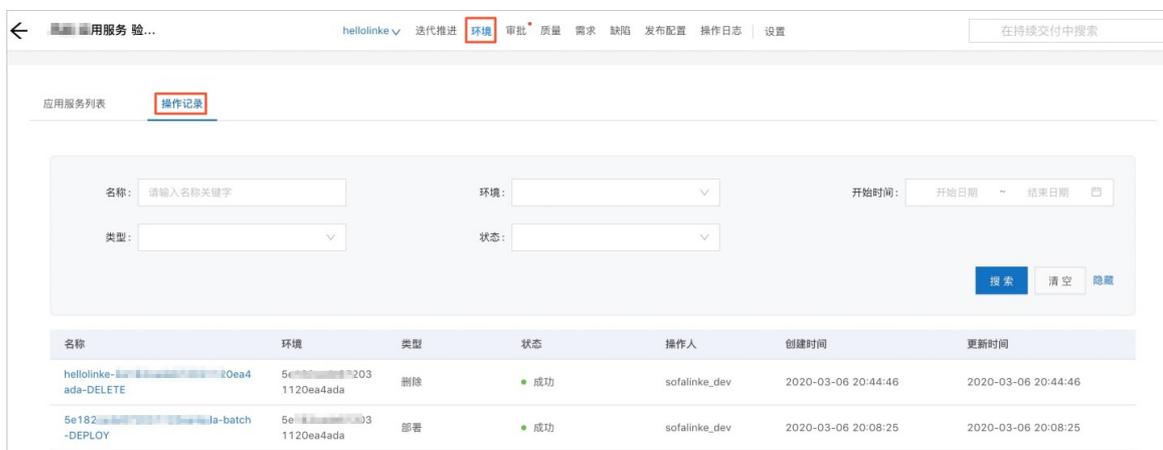
操作记录 模块主要用于记录用户在 **研发环境** 中的所有操作，便于用户查看、追踪操作历史。

操作步骤

1. 从左侧导航栏单击 **持续交付 > 研发环境 > 操作记录**，进入操作记录查询页面。可以根据名称、环境、类型、状态、开始日期、结束日期等条件查询操作记录。



2. 也可以在迭代详情页面的顶部菜单的 **环境** 入口，展示跟迭代相关的操作记录，用户可以根据筛选条件进行筛选，单击名称链接可查看详情。



4.6. 流程中心

持续交付模块支持配置人工审批流程。在研发过程中，您可以按照实际需求，对代码流程、发布流程、验收流程进行审批条件的配置，从而更好的把握各个阶段的研发产出。同时也支持查看您配置的第三方审批流程，如何添加第三方审批流程请参见 [审批配置](#)。

说明

流程审批人会在流程发生变化时收到邮件通知。

测试验证

此类型人工审批适用于测试结果确认，包括但不限于开发测试、集成测试等。

触发条件

进入相关阶段后自动触发。

流程

默认审批人为相应的测试负责人，支持处理人转派。



质量加签

触发条件

研发迭代进行阶段推进时，质量不达标的应用会提示触发质量加签审批。

流程

默认审批人为架构域质量负责人，支持处理人转派。



业务验收

此类型人工审批适用于业务验收，包括但不限于 UAT 验证、预发确认等。

触发条件

进入相关阶段后自动触发。

流程

默认审批人为迭代负责人或预发确认人，支持处理人转派。



发布评审

触发条件

在发布的前一阶段自动触发。

流程

1. 自动触发后，流程进入待提交状态。



2. 处理人完成相关的发布信息输入后，将流程真正推进到待审批状态。
3. 流程提交后，默认生成三个并行的子审批流，分别是架构师审批、质量负责人审批和运维负责人审批。三个子流程分别审批通过，则发布评审流程通过完成。

紧急发布审批

触发条件

创建紧急发布迭代时，自动触发紧急发布审批流程。

流程

1. 自动触发后，流程进入待提交状态。



2. 处理人完成相关的发布信息输入后，将流程真正推进到待审批状态。
3. 流程提交后，默认生成三个并行的子审批流，分别是架构师审批、质量负责人审批和运维负责人审批。三个子流程分别审批通过，则发布评审流程通过完成。

4.7. 管理员配置

4.7.1. 部署配置

开通持续交付组件后，用户可通过研发迭代开展研发工作，要进行开发测试环境的自动部署及生产环境的发布单提交，需要先为迭代的各个阶段配置不同的租户和环境（即工作空间或命名空间）。

说明

仅管理员有权限修改 部署配置。

租户管理

如果研发迭代不同阶段要部署到阿里云或其他跨云的环境里，组件支持用户给不同的租户配置不同的部署环境。

添加租户

编辑租户

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 [持续交付](#) > [管理员配置](#) > [部署配置](#)。
3. 在页面右上角单击 [租户管理](#)。
4. 在租户管理页面可以编辑租户或删除租户。

租户名称	是否跨云发布	网关	操作
默认租户/IF-██████████	否	https://prodapigw.cloud.ali...	编辑
默认租户/C-██████████	是	https://prodapigw.cloud.ali...	编辑 删除

< 1 >

说明

- 无法删除当前用户所属租户。
- 编辑租户信息时，是否跨云、endPoint、企业ID、租户 这三个字段无法修改。

环境配置

配置流程分为以下两种：

经典 ECS 模式

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 [持续交付](#) > [管理员配置](#) > [部署配置](#)。
3. 单击 [经典 ECS 模式](#)，选择要导入的环境、设置环境用途、完成服务器申请等配置，导入后的页面与修改工作空间的页面相同，具体参见页面文案引导。

说明

- 用户如果采用 [经典应用服务](#) 研发模式，请创建、导入和配置 [工作空间](#)，相关概念请参见 [工作空间](#)。
- 在 [持续交付](#) 中导入和修改 [工作空间](#) 之前，需要先在 [全局设置](#) > [工作空间](#) 中创建工作空间，具体操作请参见 [管理工作空间](#)。

镜像化模式

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 [持续交付](#) > [管理员配置](#) > [部署配置](#)。
3. 单击 [镜像化模式](#)，选择要导入的环境、设置环境用途、完成服务器申请等配置，导入后的页面与修改命名空间的页面相同，具体参见页面文案引导。

4.7.2. 流水线配置

LinkE 通过流水线进行研发活动的编排持续集成，在使用 LinkE 前，您可以先对流水线模板进行配置，以此满足实际的研发需求。

② 说明

仅管理员有权限进行流水线配置和管理，管理员用户可以新增、删除、修改、启用或停用当前工作空间下的流水线模板。

新建流水线模板

编辑流水线模板

在 **流水线** 页面中，选择要修改的流水线模板，单击 **操作** 列下的 **编辑**，修改模板信息或重新编排流水线。

复用流水线模板

支持基于已有的流水线模板快速创建新模板。选择目标流水线模板，单击 **操作** 列下的 **复制**，修改模板信息或重新编排流水线。

启用/停用流水线模板

只有已发布的流水线模板才能供迭代使用。选择目标流水线模板，单击 **操作** 列下的 **启用/停用** 使模板生效或失效。

删除流水线模板

选择目标流水线模板，单击 **操作** 列下的 **删除** 即可删除所选模板。

4.7.3. 迭代配置

管理员用户可以新增、删除、修改、启用或停用当前工作空间下的迭代模板。

② 说明

仅管理员有权限进行迭代模板配置。

管理员用户可以新增、删除、修改、启用或停用当前工作空间下的迭代模板。

新建迭代模板

启用/停用迭代模板

选择目标迭代模板，单击 **操作** 列下的 **启用/停用** 使模板生效或失效。

删除迭代模板

选择目标迭代模板，单击 **操作** 列下的 **删除** 即可删除所选模板。

4.7.4. 构建配置

如果需要，您可以在租户级别进行构建配置，平台支持自定义产物存储 OSS（用于存储构建包）、设置 Maven 仓库、设置镜像中心、上传构建依赖文件。

操作步骤

1. 登录 [研发效能控制台](#)。
2. 在左侧导航栏单击 **持续交付** > **管理员配置** > **构建配置**。
3. 可以根据需求选择经典 ECS 模式或者镜像化模式。



构建配置有以下两种模式：

经典 ECS 模式

经典 ECS 模式下构建的产物是 tgz 包，部署对象为经典应用服务实例。

单击 **经典 ECS 模式** 旁的编辑图标，在右侧弹出的页面中编辑配置信息。



配置产物存储

开启 **配置产物存储** 开关，填入 OSS 相关信息访问信息即可，包括 OSS endPoint、accessKey、accessId、bucket。

说明

若使用平台公共存储空间，最长保留3天。

Maven 仓库

支持租户构建私有 Maven 仓库，可直接上传配置好仓库地址的 settings.xml 文件。最多能上传一个文件。

说明

默认使用阿里云/金融云官方 Maven 仓库，如使用私有 Maven 仓库，请将配置好仓库地址的 settings 文件上传或存放在代码库的根目录。

镜像化模式

镜像化模式下构建的产物是镜像，部署对象为容器应用服务实例。

单击 **镜像化模式** 旁的编辑图标，在右侧弹出的页面中进行添加或者删除镜像仓库的操作。



说明

- 镜像仓库用于 PUSH 镜像，可以配置多个仓库地址，以支持在环境配置中用于不同环境或阶段。
- 如果不设置 **部署仓库地址**，则默认与 **构建仓库地址** 相同，在镜像仓库列表中显示 **-**。

注意事项

LinkE 会仓库地址进行分析，系统通过拼接 HTTP/HTTPS、监测连接是否超时来判断是否为安全镜像仓库，如判断不安全，则会在地址右侧添加 **!** 图标，如下图所示。



判断方式如下：

- 地址链接末尾是 `.aliyuncs.com` 的仓库为安全镜像仓库。
- LinkE 系统通过拼接 HTTP/HTTPS 请求返回为 200/401 的仓库为安全镜像仓库。

除以上两种情况，其余的仓库都会被判断为 **非安全镜像仓库**。

4.7.5. 角色配置

在企业实际的研发过程中，针对不同的应用场景，需要给企业成员配置不同的角色权限以满足具体的研发需求，本文将介绍如何在研发效能平台中进行成员与角色的配置。



注意

仅管理员可以进行角色配置。

在对成员进行角色配置前，您需要了解以下三个概念：成员、角色和权限。

名称	描述
成员	平台的使用者，管理员通过邀请的方式使企业成员加入到研发效能平台。
角色	用于配置权限后绑定成员，分为全局默认、水平权限和自定义三种类型。
权限	系统根据实际使用过程而设置的权限类型。

您可以创建一个角色，为这个角色配置一些权限类型，然后再把这个角色绑定到某个成员，那么该成员就可以进行权限范围内的相关操作。

成员

1. 登录研发效能控制台。
2. 在左侧导航栏单击 持续交付 > 管理员配置 > 成员与角色。
3. 在 成员与角色 页面下，单击 成员列表 页签，即可进行成员的添加和管理操作。



说明

列表中的 用户名 字段，是成员进入平台后，系统自动生成的一个用于标记该成员的 ID。



说明

鼠标悬停在对应成员的显示名称字段上，会显示该成员在 IAM 中的名称信息以及状态。

- 在职：该用户身份信息仍记录在 IAM 上。
- 离职：该用户身份信息已在 IAM 上删除。

添加成员

在成员列表页签下，单击邀请成员，把对应的邀请链接发送至对方，对方访问该链接后，RAM子账号自动加入。

邀请成员 ×

第一步:请先在RAM添加成员子账号,如果您不了解RAM,请查看[帮助手册](#)。

第二步:将下面链接通过钉钉、邮箱等方式发给对方,对方访问该链接后,RAM子账号自动加入,默认角色为“普通用户”,加入后可按需配置不同角色。

https://[redacted]e

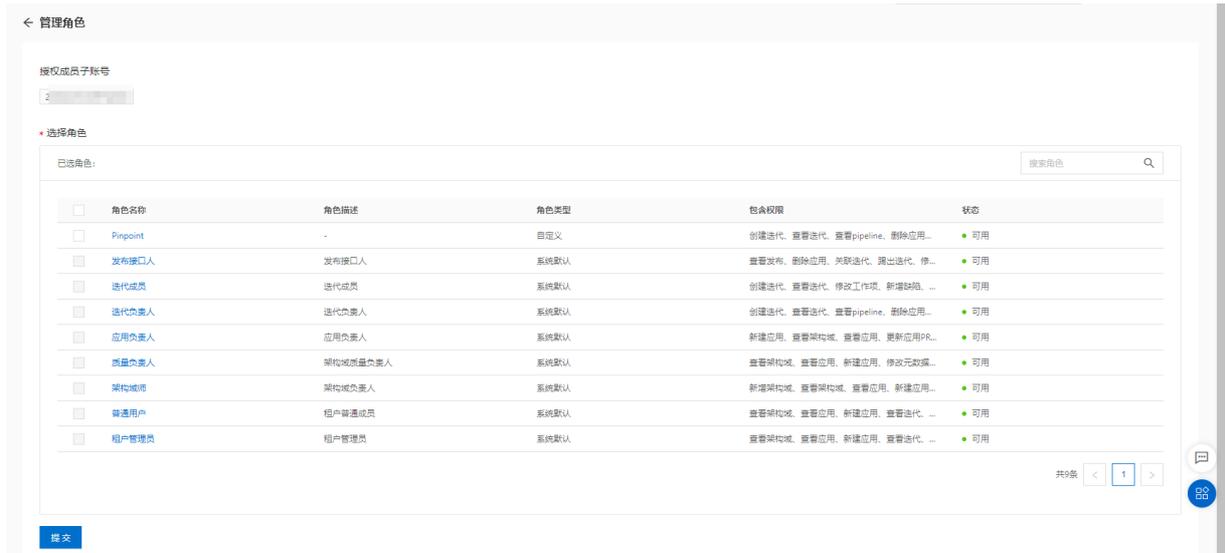
关闭

注意

- 添加成员仅能通过邀请的方式，需要被邀请成员已经事先在 RAM 中配置了子账号，详情请参见[帮助手册](#)。
- 新添加的成员默认角色是普通用户，具有普通用户的角色权限。
- 由于 RAM 是阿里云产品，非阿里云飞天底座输出的环境中可使用蚂蚁 IAM 创建访问密钥（AccessKey）做身份验证。

管理成员

在成员列表右侧，您可以单击管理角色，为目标成员绑定或解绑角色，选中想要为该成员授权的角色，然后单击页面下方提交即可。



说明

- 一个成员可以绑定多个角色。
- 只能为成员绑定自定义类型的角色。

移出成员

在成员列表右侧，您可以单击移出，将目标成员从 Linke 平台删除。



导出成员信息

在成员列表选中需要导出的成员，然后单击批量操作 > 批量导出，即可导出格式为 Excel 的成员信息。



角色

1. 登录研发效能控制台。
2. 在左侧导航栏单击持续交付 > 管理员配置 > 成员与角色。
3. 在成员与角色页面下，单击角色列表页签，即可进行角色的添加和管理操作。

角色权限

成员列表 角色列表 权限一览表

添加角色 批量操作

全部类型 搜索角色

角色名称	角色描述	角色类型	包含成员	包含权限	状态	操作
Pinpoint	-	自定义	1	创建迭代、查看迭代、查看pipelin...	可用	编辑 导出 ...
发布接口人	发布接口人	系统默认	0	查看发布、删除应用、关联迭代、...	可用	编辑 导出 ...
迭代成员	迭代成员	系统默认	0	创建迭代、查看迭代、修改工作项...	可用	编辑 导出 ...
迭代负责人	迭代负责人	系统默认	0	创建迭代、查看迭代、查看pipelin...	可用	编辑 导出 ...
应用负责人	应用负责人	系统默认	0	新建应用、查看架构域、查看应用...	可用	编辑 导出 ...
质量负责人	架构域质量负责人	系统默认	0	查看架构域、查看应用、新建应用...	可用	编辑 导出 ...
架构域师	架构域负责人	系统默认	0	新增架构域、查看架构域、查看应...	可用	编辑 导出 ...
普通用户	租户普通成员	系统默认	0	查看架构域、查看应用、新建应用...	可用	编辑 导出 ...
租户管理员	租户管理员	系统默认	0	查看架构域、查看应用、新建应用...	可用	编辑 导出 ...

共9条

角色类型说明如下表所示：

类型	说明
自定义	用户创建的角色都为自定义类型，且可以进行二次编辑和绑定成员。
水平权限	系统默认创建好的，除管理员和普通用户之外的角色类型，是根据在实际应用过程中的操作设置的，比如：某成员创建了一个迭代，那么该成员就默认是迭代负责人角色，拥有迭代负责人角色的所有权限。
全局默认	<p>系统默认创建好的，只包括管理员和普通用户两种角色类型。</p> <p>说明</p> <ul style="list-style-type: none"> 管理员角色无法进行编辑。 普通用户角色名称无法编辑。

添加角色

在角色列表页签下，单击添加角色，输入相关字段并且为角色添加权限类型，完成后单击提交。

添加角色

角色名称

角色描述

负责人

参考已有角色: 已选权限: 0

迭代

查看迭代 创建迭代 修改迭代信息 删除迭代 创建MR 查看pipeline 触发pipeline 定制pipeline 阶段推进 修改工作项 新增缺陷 发起审批 关联发布 取消关联发布 添加应用 删除应用 删除/添加迭代成员 修改迭代负责人

发布

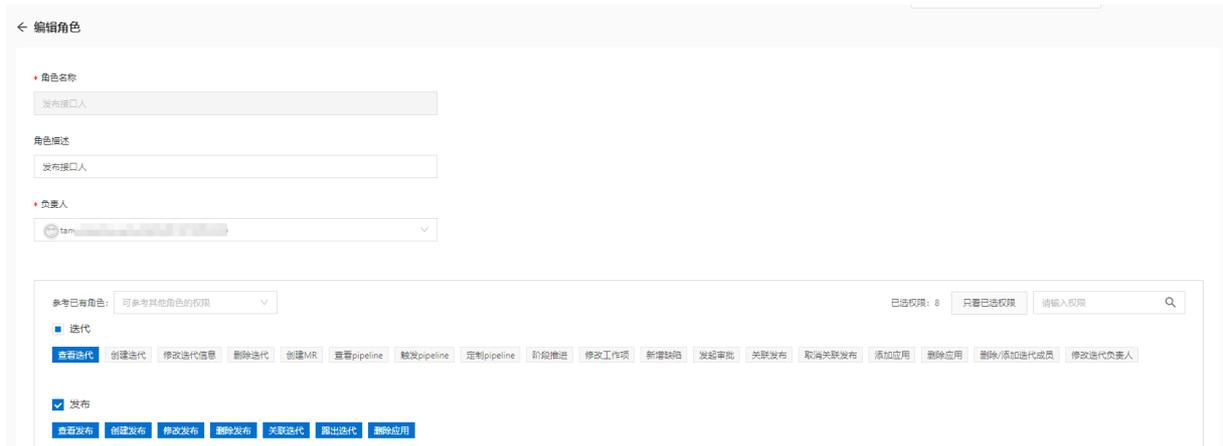
查看发布 创建发布 修改发布 删除发布 关联迭代 退出迭代 删除应用

说明

- 负责人 仅能从已添加的成员中选择。
- 新增角色的类型都为自定义类型，默认不具备任何权限。

编辑角色

在 角色列表 右侧单击 **编辑**，进入角色编辑页面，该页面与添加角色页面中的字段相同。



停用/删除角色

在 角色列表 右侧单击 “...” 按钮，您可以对角色进行停用或者删除操作。



您也可以单击 **批量操作**，来对角色进行批量删除或停用/启用。



说明

只有 **自定义** 类型的角色可以被删除。

查看角色详情

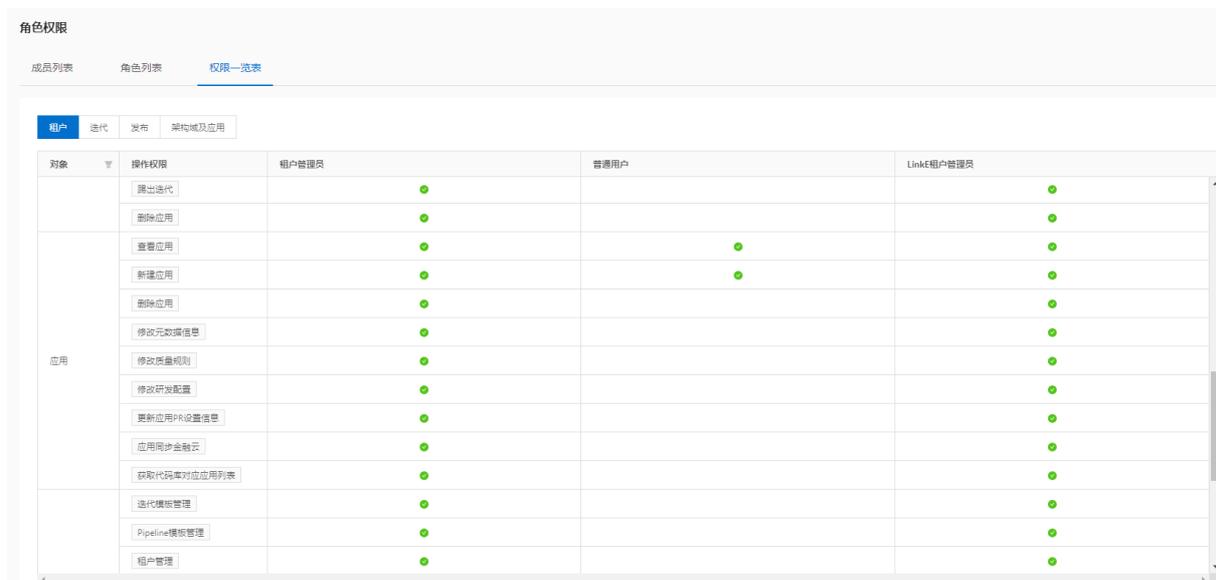
在 角色列表 单击目标角色名称进入角色详情页面，您可以查看目标角色的基本信息，包括名称、类型、负责人以及所包含的权限等，同时也可以对该角色所包含的成员进行添加和移除操作。



说明
只有 **管理员角色** 和 **自定义** 类型的角色可以进行添加或移出成员的操作。

权限

1. 登录研发效能控制台。
2. 在左侧导航栏单击 **持续交付 > 管理员配置 > 成员与角色**。
3. 在 **成员与角色** 页面下，单击 **权限一览表** 页签，即可查看所有权限类型。



说明
目前权限类型为系统定义，不支持增加或删除权限类型。

4.8. 最佳实践

4.8.1. 合并请求

合并请求指从一个分支合并到另一个分支，代码服务的一个重要组成部分，是代码协作的基础。合并请求可能是新需求、优化改造、缺陷修复等。典型的合并请求处理过程涉及如何提交合并请求、如何对合并请求进行评审以便确定是否接受请求、由谁来处理合并、合并后的通知机制等问题。

通过 **合并请求** 功能，您可以：

- 比较两个分支之间的变化。
- 在线查看和评论代码修改，并记录问题状态。
- 设置评审规则，支持多种评审需求，如多人评审等。
- 设置合并规则 以控制合并准入。
- 展示合并冲突列表。
- 压至合并，让提交历史记录更清晰。
- 查看合并请求版本，合并请求版本基于 push 产生，您可以选择并比较这些合并请求差异的版本。

应用场景

合并请求的准入设置可以有多种情况，例如：团队中的开发人员需要提交代码。从创建合并请求到通过评审、合并代码分支的整体流程如下：

1. 创建一个新的分支并修改、提交代码。
2. 创建合并请求，提交代码变更。
3. 团队其他成员进行评审，在合并请求的评审记录中进行反馈、讨论。
4. 开发人员根据评审意见进行修改，解决问题。
5. 通过评审后，分支予以合并。

合并请求设置

在合并请求设置中，您可以设置合并规则，要求代码分支必须通过评审才能合并。操作步骤如下：

1. 从左侧导航栏进入 **代码服务 > 我的仓库**，选择目标代码仓库，点击进入代码仓库详情页面。
2. 在页面上方，点击 **设置 > 合并请求设置**。
3. 开启 **是否必须评审才能合并** 开关。一旦开启，分支必须通过评审才能合并；如果关闭，可以由单个合并请求确定是否需要评审。
4. 点击 **更新** 使设置生效。

评审设置

在评审设置中，您可以设置评审规则。操作步骤如下：

1. 从左侧导航栏进入 **代码服务 > 仓库**，选择目标代码仓库，点击进入代码仓库详情页面。
2. 在页面上方，点击 **设置 > 评审设置**。
3. 在 **评审通过票数** 中，设置通过评审所需的票数。
4. 开启 **评审人可以设置为提交人** 开关，允许评审人和提交人为同一个人。该设置项主要针对一次合并请求包含多人修改的场景。
5. 开启 **当源分支更新时清空投票** 开关。一旦开启，若合并请求还有未完成的评审，当源分支代码变更时，原投票数清 0。
6. 点击 **更新** 使设置生效。

② 说明

负责人有权限修改评审设置，如果在创建合并请求后修改设置，则合并请求会保留之前的设置，但可以通过编辑合并请求来修改。

新建合并请求

可通过以下任一入口创建合并请求：

- 在代码仓库详情页面，点击 **合并请求** > **新建合并申请**，开始创建合并请求。
- 在代码仓库详情页面，点击 **分支** > **合并请求**，开始创建合并请求。

提交评论

进入合并请求详情页面，对合并请求进行评审。

评审模式

支持提交单个问题和开启评审两种模式：

- 提交单个问题/评论：提交评审后就会发出通知。
- 开始评审：点击 **开始评审** 后，会触发评审模式。评审完成后，整体提交评审并发出通知。

评论类型

评论区分为问题和评论。问题是需要本次解决的；评论可能是一个优化建议不需要本次解决。

评论范围

评论的范围有两种：针对 `commit diff` 的评论和整体评论。

查看 commit diff 的评论

在合并请求详情页中，点击 **提交记录** 标签可查看提交记录列表。

点击 **文件变更** 标签可查看修改的内容，可选择对应的行数添加评论。创建的评论都会显示评论区域。

整体评论

指在讨论区域针对整个合并请求进行整体评论。

给问题增加标记

在代码仓库详情页面，点击 **设置** > **标记**，可创建并管理标记。可通过标记对评审问题进行分类，比如标记为代码风格、代码逻辑等。

查找差异文件

在 **文件变更** 标签页中，可通过搜索框查找差异文件。当有大量文件变更时，可以通过下拉导航轻松地跳转到任何更改的文件。

更改问题状态

可将问题标记为以下状态：已解决、未解决、无效问题。

评审通过

在合并请求详情页，评审人点击 **通过** 后，系统会根据通过票数确定合并请求是否通过评审。

评审区域展示评审情况：

如果已到达评审票数（即评审通过的票数等于设置的评审通过票数），则分支可以合并。但是，除了评审票数外，如合并冲突、目标分支写权限不满足等也同样无法进行合并操作。

合并

在代码库中使用保护分支控制合并权限。开发人员将功能分支推送到代码库中，创建合并请求通过评审后，拥有保护分支写权限的用户（Master 用户）完成合并。

合并冲突

当出现冲突时，合并请求中会展示冲突文件列表，可通过点击 **解决冲突** 旁的问号图标查看如何在本地解决冲突。

说明

不检测路径变更的重命名冲突。例如，在分支 a，执行 `git mv file1 file2`；在分支 b，执行 `git mv file1 file3`，这不会视为冲突，这两个文件合并后都将出现在分支中。

压制合并

选择压制合并（squash merge）可以在合并时将合并请求的所有提交合并为一个，并保留一个清除历史记录。它将合并请求中的所有更改作为单个提交。

假设某用户提交了 3 个合并请求，分别是 “1st”、“2nc”、“3rd”：

- 如果采用一般的合并方式，会有 “1st”、“2nc”、“3rd” 及合并产生的 “Merge branch...” 共 4 个提交；

- 如果采用压制合并，就会把“1st”、“2nc”、“3rd”这3个提交压制成“Squash merge...”这1个提交及合并产生的“Merge branch...”提交。

压制合并的提交信息是除“Squash merge...”说明外，还会把被压制的3个提交信息汇总展示。

启用压制合并

在合并请求详情页中，选择 **压制提交**：

使用场景

在 feature 分支上工作时，有时想要提交当前的进度，但不关心提交消息。那些提交不一定包含重要信息，因此不需要也不应该将其包含在目标分支中。

通过压制合并，可以将合并请求中的提交包含到单个提交中。这样，基本分支的历史记录就会保持干净，并提供有意义的提交消息，并且在必要时更容易回滚。

合并请求版本

每次推送到与合并请求关联的分支时，都会创建新版本的合并请求差异。当您访问包含多个推送的合并请求时，可以选择并比较这些合并请求差异的版本。

合并请求版本基于 push 产生。因此，如果您在一次推送中 push 了 5 次，那么下拉菜单中会计入 5 个选项。默认情况下，会显示最新版本的更改。但是，您可以从版本下拉列表中选择较旧的版本。

您还可以将合并请求版本与旧版本进行比较，以查看中间发生了哪些变化。

4.8.2. 自定义构建发布

自定义构建功能，指通过读取代码库中的配置文件，根据配置文件指定的 Docker 镜像，拉起容器进行构建。构建的步骤、执行命令、以及构件产物的定义都是在配置文件中配置。

构建过程

- 触发构建之后，系统根据接口提供的代码仓库地址，将代码下载下来。
- 读取构建配置文件内容并解析，根据配置对应的 Docker 镜像拉起容器，并在容器中执行构建命令。
- 构建成功之后，根据构建产物配置，将对应产物上传到对象存储（OSS）中。

配置文件

配置文件路径

默认读取代码目录下的 `.linkb-build.yml` 文件作为构建配置文件。

配置文件示例

配置文件以 YAML 进行构建的编排，以下为配置文件示例（SpringBoot 工程示例）。

```
build:
  image: reg-docker.linke.alipay.com/maven-build/builder-maven3.6.0-jdk1.8:1.0.0
  script:
  - mvn --version
  - mvn clean install -s ./settings.xml -Dmaven.test.skip=true-Dmaven.repo.local=$HOME/.m2
  - tar zcvf target/output.tgz xxx_1/target/xxx.jar xxx_2/target/xxx.jar
  artifacts:
  -{
    name:"demo-springboot-0.0.1-SNAPSHOT.jar",
    desc:"Springboot executable fat jar",
    paths:["target/demo-springboot-0.0.1-SNAPSHOT.jar"]
  }
  -{
    name:"output.tgz",
    paths:["target/output.tgz"]
  }
```

- **image 标签**：指定构建所使用的容器。示例中的

`reg-docker.linke.alipay.com/maven-build/builder-maven3.6.0-jdk1.8:1.0.0` 是目前唯一支持的 Docker 镜像。其中，镜像提供了一个 CentOS 7 的容器环境，容器中配置了 Maven 3.6.0 和 JDK 1.8。

- **script 标签**：指定构建所需要执行的 Shell 命令，从上到下依次执行。
- **artifacts 标签**：指定构建的产物，支持依次构建多个产物的配置。其中：
 - **name**：表示依次构建的产物名称。在 artifacts 中，每个产物的 **name** 必须唯一，不能为空。
 - **desc**：构建产物的描述，可以为空。
 - **paths**：在执行完构建命令之后，产物在代码目录的路径，不能为空。