云效

使用指南

使用指南

企业管理

企业信息和成员管理

创建企业

创建入口

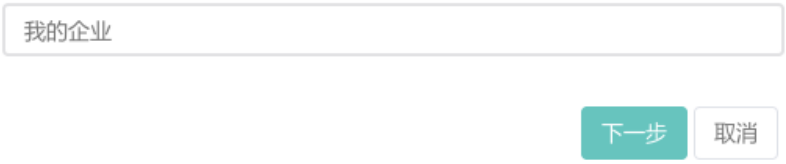
如果用户还没有创建或加入任何企业，第一次登录会提示创建新的企业，点击即可开始创建企业：

如果用户已经加入某个企业，可以通过右上角”+”入口新建企业：


搜索
新建任务
新建缺陷
新建需求
新建企业
新建项目
新建项目集

填写企业名称

输入所在企业的名称，然后点下一步：



邀请企业成员

填写企业成员邮箱邀请成员加入。如果在这一步还不能确定企业成员，可以先跳过这一步，后面在企业管理入

口再邀请成员：

进入企业

企业创建完毕后，点“进入企业”，然后开始进行项目、工作项、应用、代码和发布等管理工作：

管理企业

管理入口

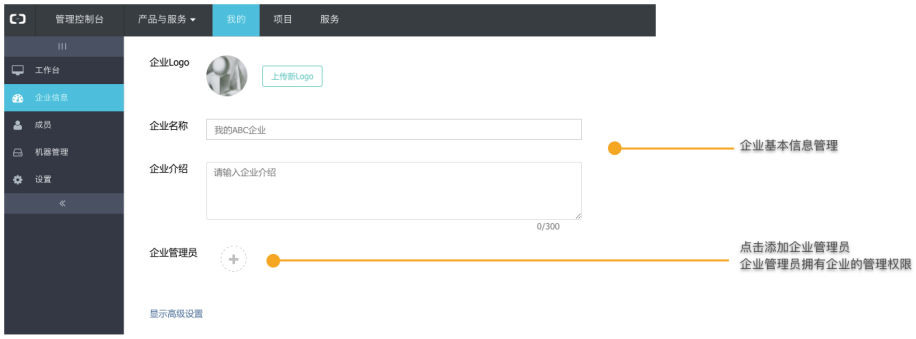
点击右上角设置图标，出现企业设置和企业切换入口：

- 点击企业设置可进入企业基本信息管理界面
- 如果同时加入多个企业，点击切换企业可进行企业切换

基本信息管理

在企业信息设置界面:

- 可以设置企业Logo、企业名称和企业介绍等信息
- 可以添加企业管理员，该管理员拥有企业的管理权限



企业成员管理

点击左侧的“成员”进入成员管理界面，点“添加成员”可邀请成员加入当前企业：

点这里立即体验RDC

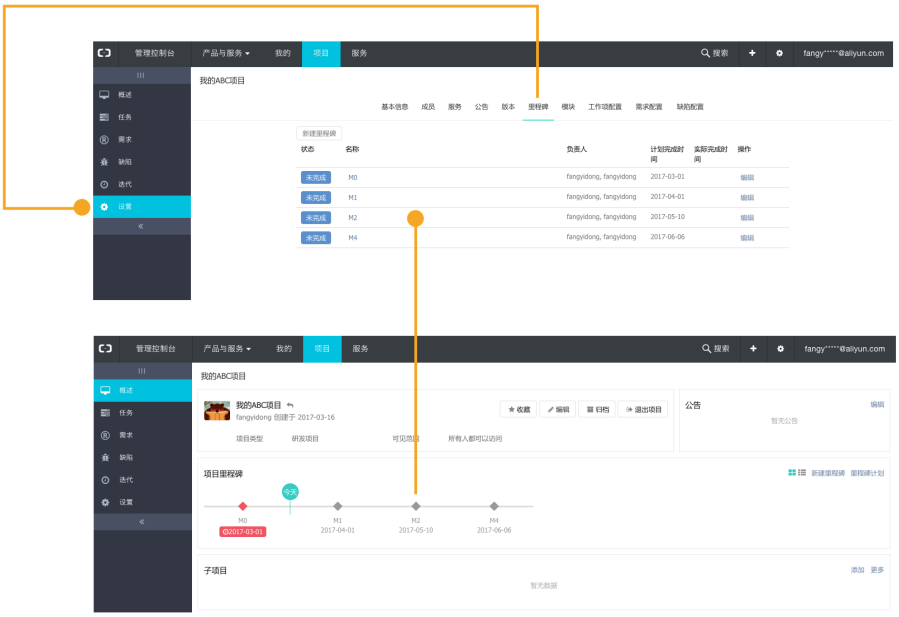
项目协作

里程碑计划

里程碑计划功能让项目管理者清晰定义项目目标和任务，并对项目里程碑计划进行实时监控。

启用里程碑计划

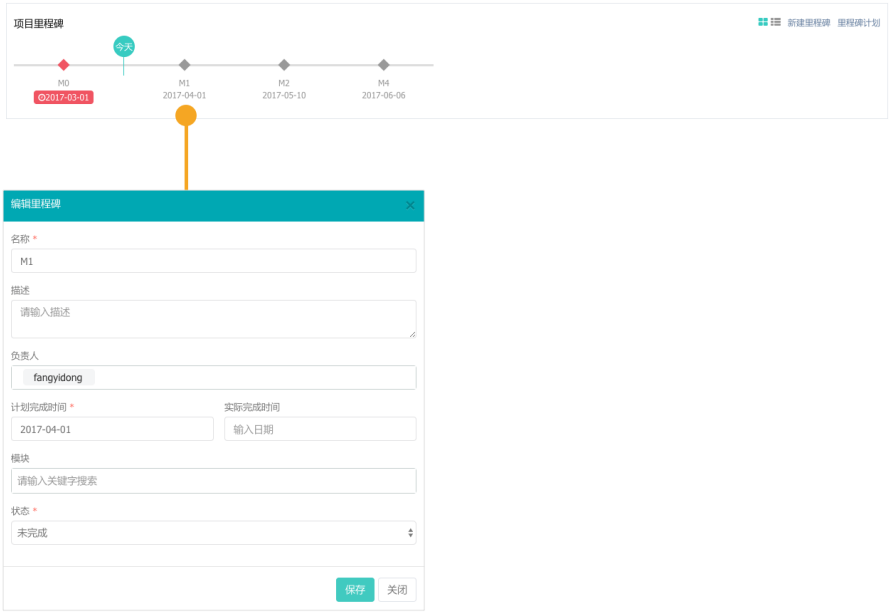
在项目里面,点“设置”，然后点“里程碑”TAB，增加第一个里程碑数据后，在项目概况里面会显示里程碑区



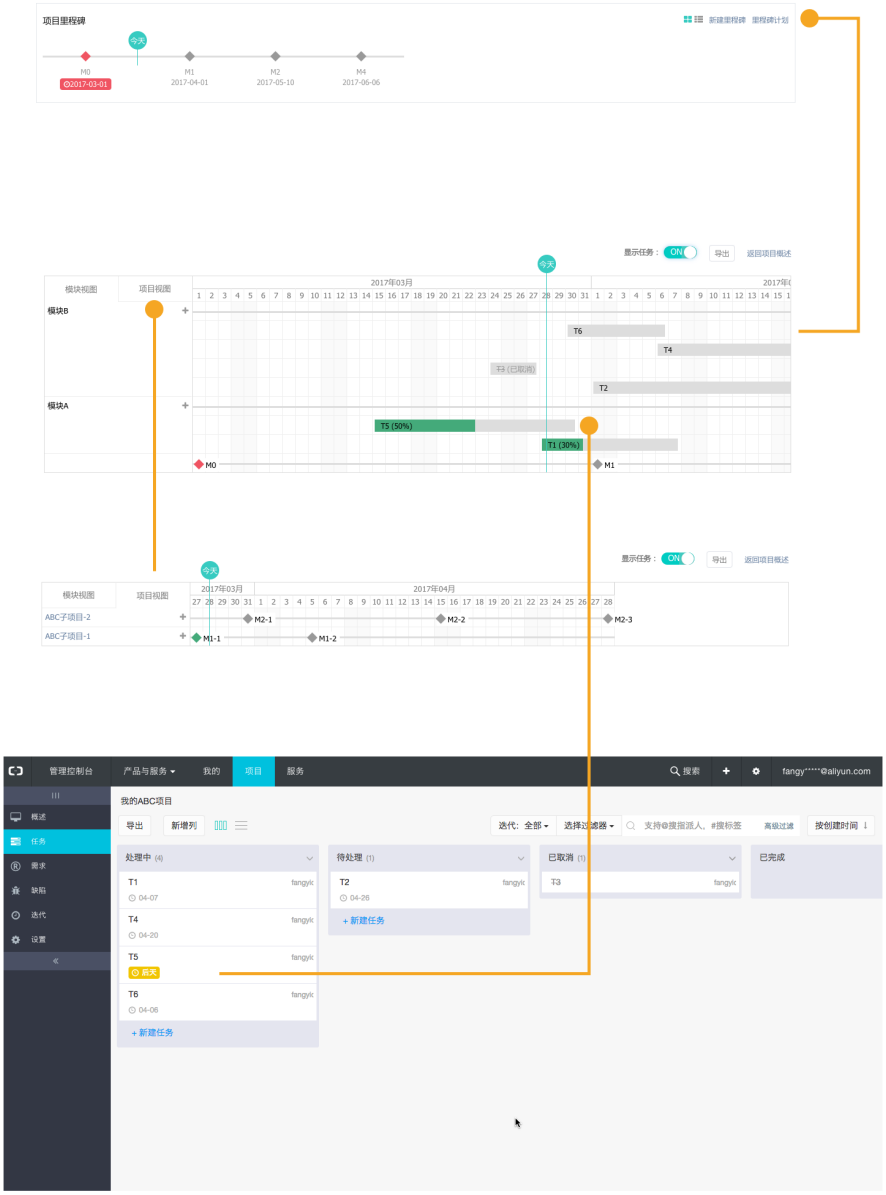
块。

管理里程碑和计划任务

延期的里程碑高亮显示，里程碑完成后，高亮消失，这样有助于项目负责人关注未完成的目标。注：如果要删除里程碑，把里程碑状态设置为“归档”即可。



在概况的里程碑区块，可以进入详细的里程碑计划页面，在这里可以对里程碑和计划任务进一步按照模块来划分不同的线。在这里，计划任务和在任务墙出现的任务是一致的（注：目前为了避免过多的日常任务进行干扰，只显示和某一个模块关联的任务）。此外，如果当前项目为父项目，可以从这个视图方便查看子项目的里程



碑和计划。

迭代管理

迭代是敏捷开发的概念，它是有开始和结束时间的轻量级计划，用来明确规划在开始和结束时间之间需要实现的需求、需要修复的缺陷和需要完成的任务。一个典型迭代的周期从1到6周不等，团队可根据自己的节奏或业务的需要来确定迭代周期。

以典型的Scrum为例，迭代规划的具体流程为：

1. 用户和业务方提出的需求和缺陷，由Product Owner（产品负责人）来统一管理，经分析、评估、

拆分和PK后，确定优先级，在计划会（排期会）上和ScrumMaster（迭代负责人）和研发团队进行排期，进入迭代

2. 研发同学在迭代周期里面，对自己负责的需求进行任务拆分、拉代码变更分支，并且每天更新进度和状态
3. 需求实现进行测试和验收后，进行发布，相关需求状态自动设为完成
4. 迭代完成后，如果有未完成的工作，移到下一个迭代

和团队一起进行迭代开发

创建迭代

迭代一般由ScrumMaster来创建和管理。ScrumMaster主要职责制定最佳工作模式，协调团队开发和跟进解决blocker，并保护团队避免受到外部干扰。

在项目里点左侧“迭代”TAB可创建迭代：

规划迭代内容

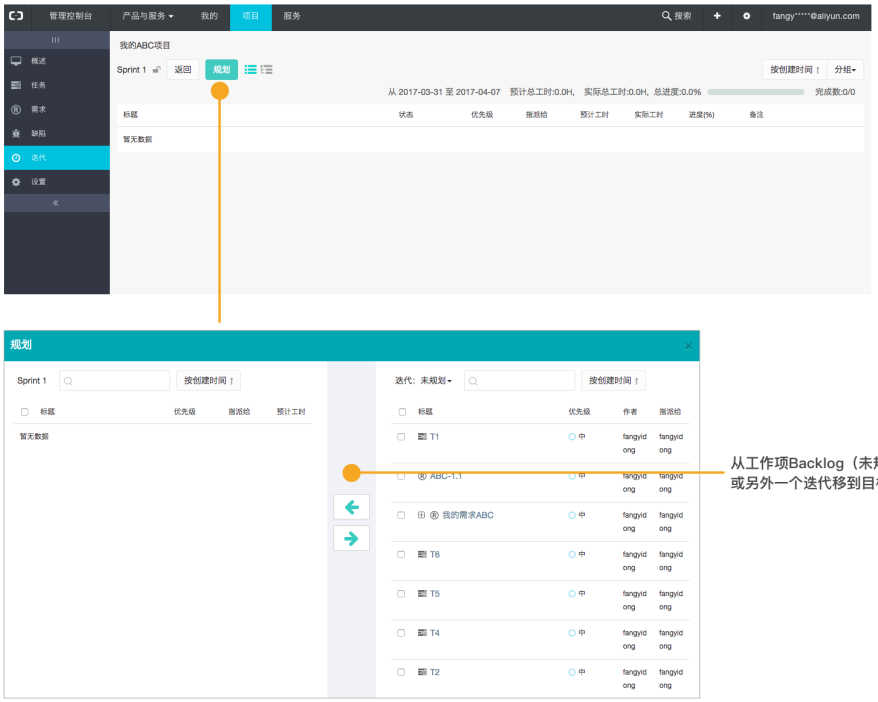
每个迭代具体要排期哪些内容，Product Owner定优先级，研发团队根据需求估算、团队速率和可承受并发度等确定能做多少内容。

在云效里面，把工作项（需求、任务、缺陷）规划进迭代有3种方式：

在工作项详情页，找到“迭代”字段，选择目标迭代

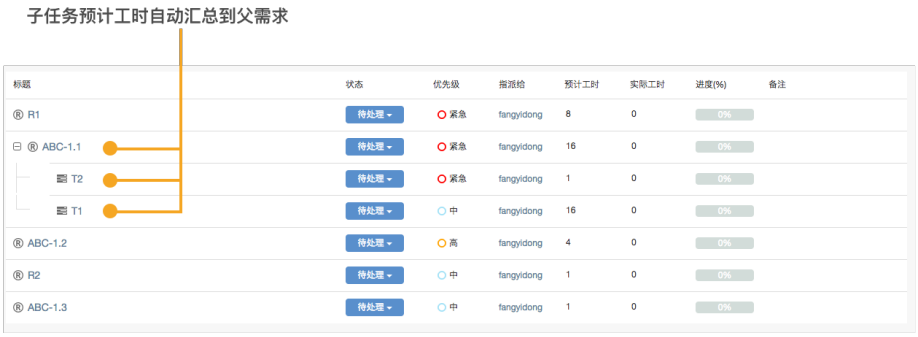
在工作项列表页，直接在迭代列点击选中目标迭代

在迭代里面，点“规划”按钮，可批量把工作项拉入迭代

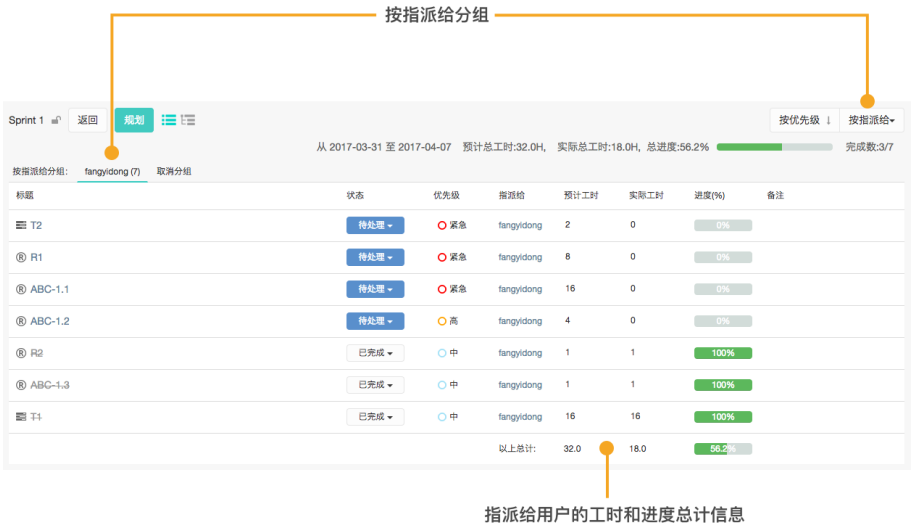


在规划迭代的时候，Product Owner按工作项优先级从高到低，进行需求讲解，然后研发一起进行预计工时评估（不需要很精确，而是快速进行评估），云效会自动对工时进行汇总，如果总工时到达团队一个迭代内可用工时，Product Owner停止讲解，并把剩余工作项移除迭代：

研发可以选择对需求进行任务分解，然后针对每个任务进行更细致的工时估算，这些工时会自动汇总到父需求:



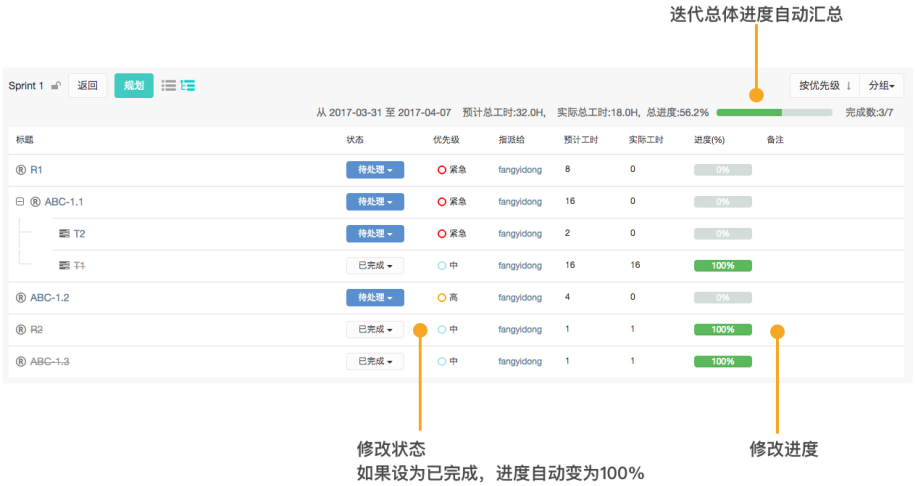
ScrumMaster可以对工作项按指派给分组，从而掌握团队成员的工作负荷分布，必要时，对任务分配进行平衡



：

迭代执行和跟进

研发负责的工作项完成后，把状态设为已完成，进度自动更新为100%，迭代总体进度会自动进行重新计算:



工作项管理

需求、缺陷和任务统称为工作项。需求、任务和缺陷的区别：

工作项	定义	使用场景
需求	代表所需要解决的问题	用户需要借助产品实现某个目标，但是产品尚未支持
缺陷	当系统没有按设计运行的时候，即产生了缺陷	产品出现故障和问题，运行方式和结果不符合设计期望
任务	代表一个小粒度的活动	研发实现某个需求和解决某个缺陷，对涉及到工作拆解成一系列的任务，如搭建环境、编写单元测试脚本等

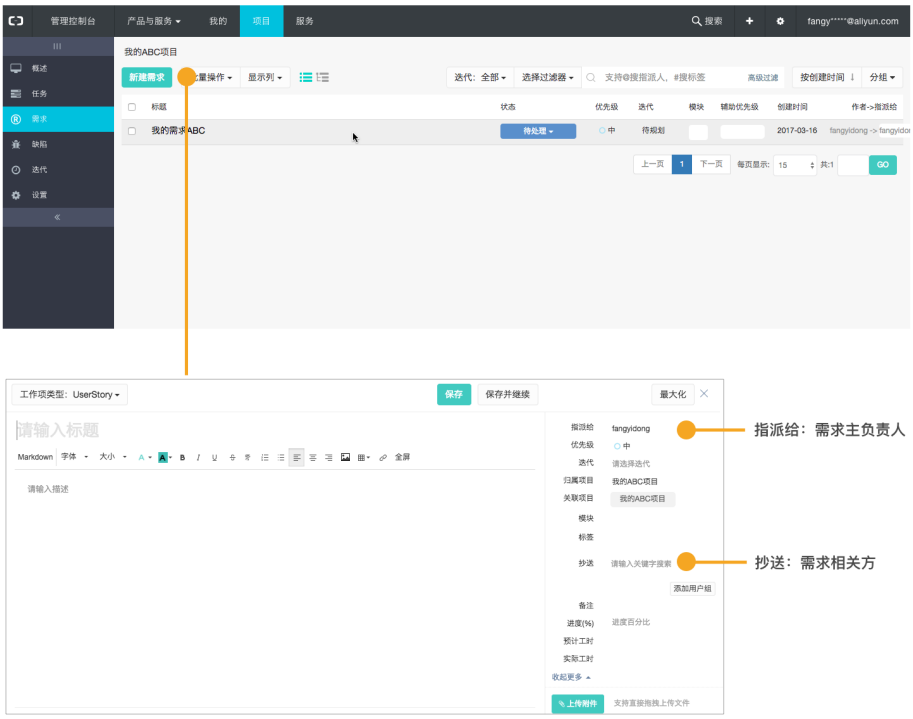
需求管理

录入需求

录入需求是比较简便的操作，只要点项目左边“需求”TAB，就可以发现新建需求的按钮，点一下按钮，然后填上需求的标题和正文，再点“保存”即可。

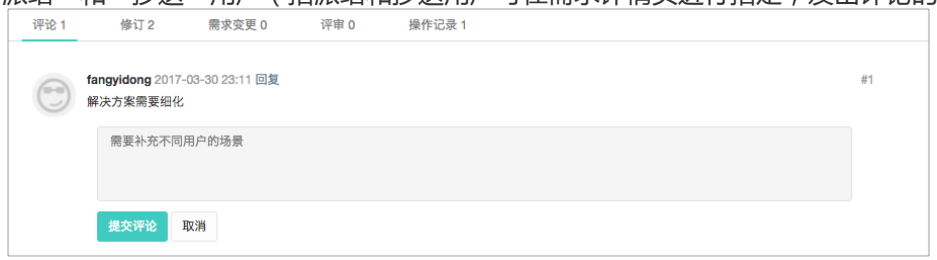
提示：

- 需求正文可直接按CTRL+V进行贴图
- 需求正文支持markdown方式编辑



团队在线上对需求进行讨论

在形成结论之前，可利用需求评论功能对需求进行讨论。所有讨论会完整记录下来，且实时发送邮件通知“指派给”和“抄送”用户（指派给和抄送用户可在需求详情页进行指定，发出评论的人不会收到邮件通知）。



需求进入迭代之前，对需求进行细化

一般来说，一个迭代一般是1-2周的周期，所以进入迭代的需求一般粒度不能太大，一般是从用户角度出发的端到端的小粒度功能，符合以下原则（INVEST）：

1. Independent，独立
2. Negotiable，可协商，不能定太死，开发过程可变通
3. Valuable，有价值
4. Estimable，可估算，不能估算意味着无法做相对准确的计划，一般是因为粒度还不够小
5. Small，足够小，一个迭代能够做完，不能跨迭代
6. Testable，可测试

如果需求太大，可以不断拆分直到符合上述标准。

和团队一起规划迭代

迭代一般由Scrum Master（迭代负责人）来创建和管理。每个迭代具体要排期哪些内容，Product Owner定优先级，研发团队根据需求估算、团队速率和可承受并发度等确定能做多少内容。

缺陷管理

用户在测试环境发现缺陷，提交缺陷

需求功能测试过程发现的缺陷在项目中直接录入，并且可规划到迭代中解决。

缺陷状态

缺陷状态如下：

- New: 新增加的、需要解决的BUG。
- Open: 正在定位问题，或正在解决中，或已经解决但未部署生效。
- Fixed: BUG已经解决，并且修改后程序已部署生效。

- Closed: 验证后，此BUG可以关闭。
- Reopen: 此BUG需要再解决。
- Later: 此BUG不在本项目的工作范围内，在后续版本中修复。
- Worksforme: 不能在当前环境中重现。
- Duplicate:和其它BUG描述现象重复。可以配置选择Duplicate状态时必填关联的缺陷ID。
- Invalid: 属于测试人员对测试需求的理解错误。
- External: 问题是由其他外部的原因引起，需要由外部处理。
- ByDesign：属于按照产品设计实现，不是问题。
- Wont’ fix：问题确实出现过，但是由于产品改动已经修复或者功能废弃，问题目前已不需要解决

上述的状态可以划分为待处理、已处理、已关闭等三大类：

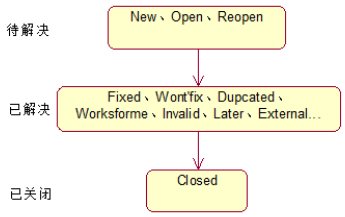
- 待处理：New、Open、Reopen
- 已处理：Fixed，Wont’ fix，Later，Worksforme，Duplicate，Invalid，External，ByDesign
- 已关闭：Closed

已处理中的6个状态都可以视为问题处理人对此问题的处理意见。我们称它为：解决方案。

经过以上调整，我们就有了以下规则：

1. 所有缺陷最终都应该由缺陷验证者或者与之平级权限的人变更到Closed状态。非Closed状态的都被认为是活动的缺陷。
2. 解决者可以将“待处理”状态的缺陷置为任意“已处理”状态。

状态流转图如下图：



再用文字来解释一下上图的一个经典流程：

1. 测试（验证者）发现了问题。提交一个问题。（状态为New)
2. 开发（解决者）去解决。先Open或者直接选择一个解决方案。
3. 测试通过就关闭它。如果验证后不正确，那就Reopen。再返回到步骤2。

任务管理

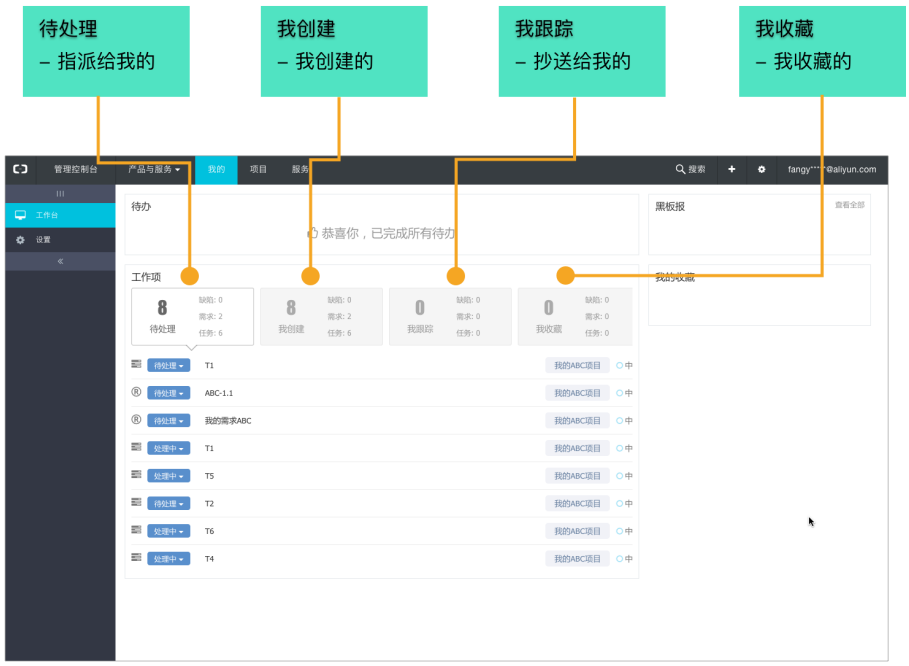
任务代表一项小粒度的活动，可以来自于对需求实现的任务拆分，也可以单独创建。

从需求拆分任务

直接创建任务

个人维度管理工作项

点顶部”我的”，进入工作台，里面显示和当前用户相关的任务和其它工作项（需求、缺陷等）：



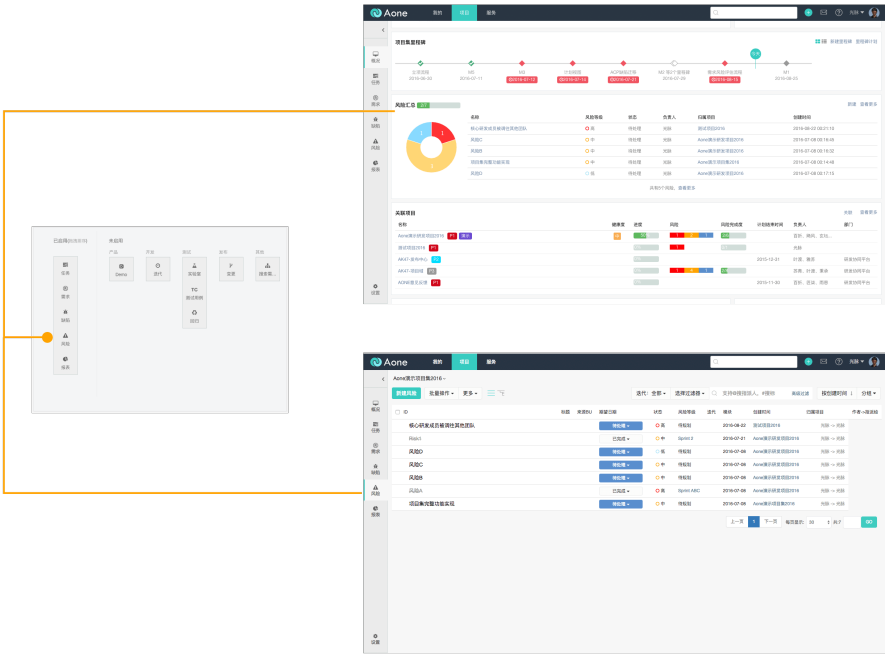
风险管理

风险管理功能可以帮助项目和项目集管理者对风险进行实时监控，让风险透明化，并对风险进行有效跟进和追踪。

风险分为高、中、低三个等级，是综合风险的影响和发生概率的一个指标。PMO（项目管理办公室）应该在组织全局范围内整体制定风险等级标准，根据风险对整体目标达成的影响来对风险进行定级。

启用风险服务

在项目里面，点“设置 > 服务”，启用风险服务。启用后，在项目概况会显示风险区块。项目和子项目创建风险对象后，会自动汇总到风险区块里面。



风险汇总

风险汇总区块自动汇总子项目的未完成风险，并和当前项目的风险一起显示。

子项目的风险信息显示

在子项目列表中，列出每个项目的未完成风险和风险完成占比信息。

风险状态更新自动发送通知

在每一个风险对象的详情页里，风险状态发生改变时，或者对风险相关方案进行讨论时，会自动发送邮件通知到作者（风险创建人）、指派给和抄送。

另外可以对每个风险拆分多个子任务，分别指派给不同的人进行协同处理和跟进。

项目集管理

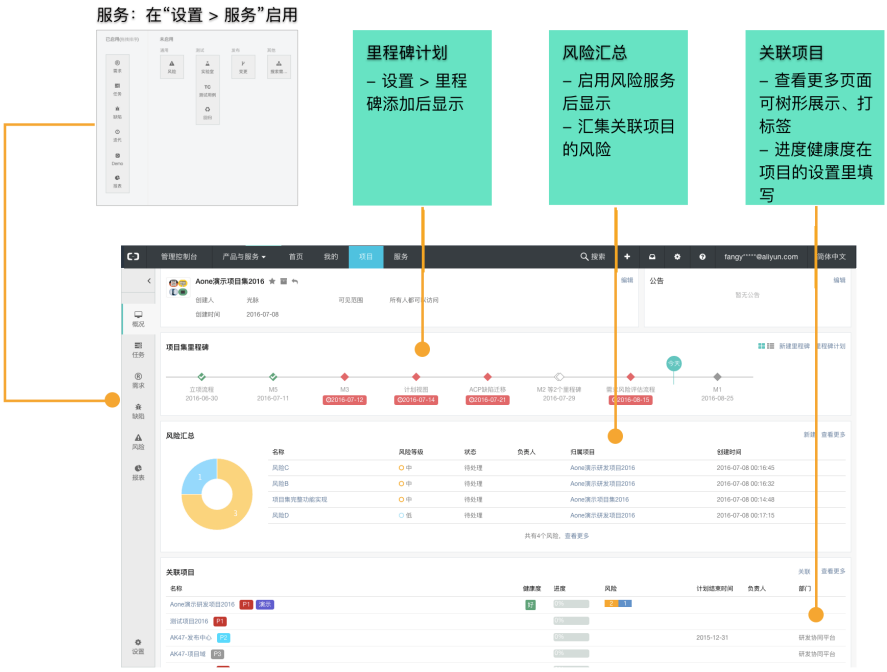
项目管理协会（PMI）把项目集定义为：经过协调管理以便获取单独管理这些项目时无法取得的收益和控制的一组相关联的的项目。这个定义有点绕，简单一点来说，如果有一组互相关联的项目，需要互相协作而获得一个共同的目标，那么可以放在一个项目集里面进行统一管理，例如双十一大促多项目管理和协同。

云效最新的项目集管理功能可以让您轻松管理和监控多项目的进展和风险，更好促进项目间协作。

创建项目集

顶部导航菜单，点“+”，选“新建项目集”，然后根据向导创建项目集。项目集创建后，列表和搜索和普通项目一样，统一管理。

主要功能入口



管理关联项目

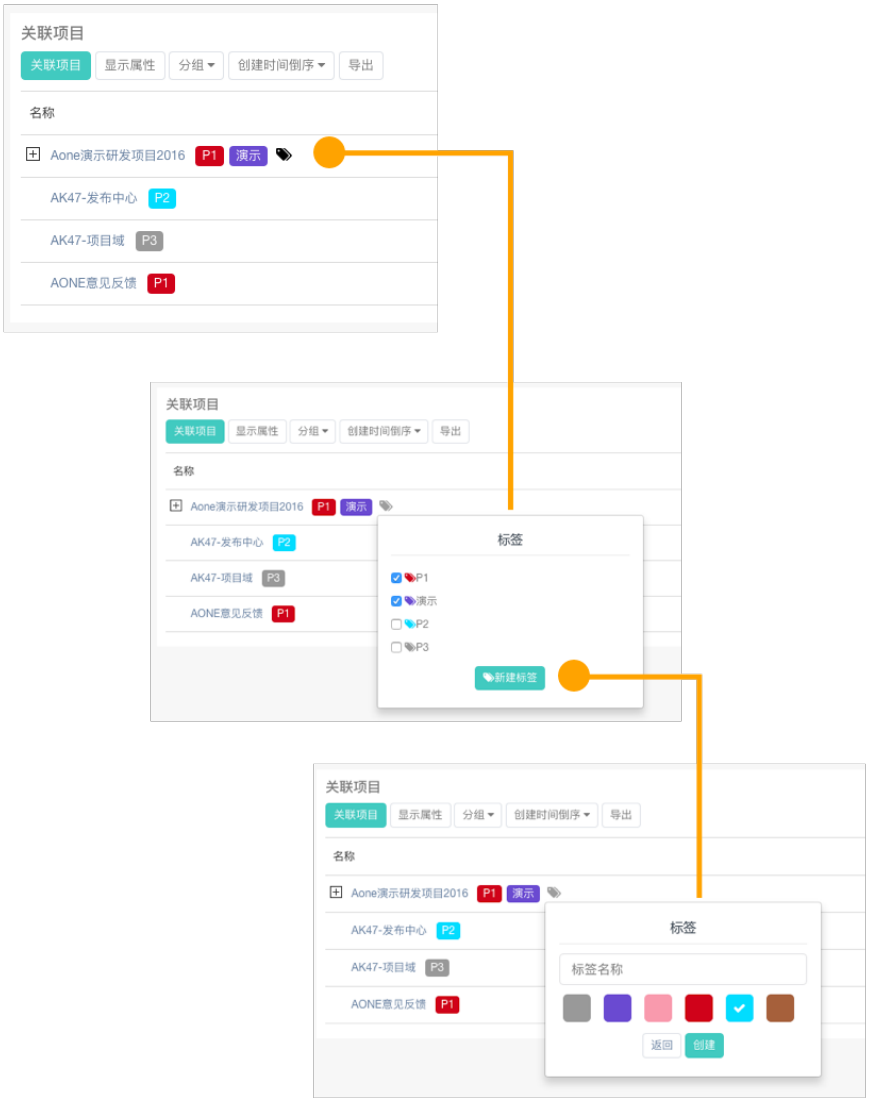
通过条目化项目列表管理，可以方便查看项目信息，了解项目进度、健康度和风险等信息。另外，通过标签功能，可以方便对项目进行任意维度的分组管理，这样可以快速管理重点关注项目。

过滤、分组、排序和树形展示

在概况里可直接关联项目，点击“显示更多”后进入项目列表页，可设置显示属性、分组和排序、过滤、切换树形展示，还可以给项目打标签：

给项目打标签

在关联项目的“显示更多”页面，鼠标移到项目上面的时候，出现标签图标，点击后可给项目打标签，然后可



以对标签进行过滤和分组：

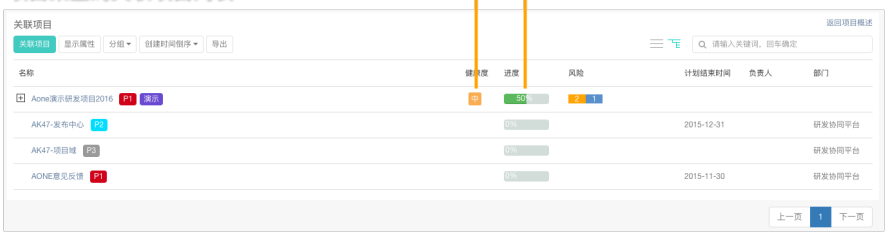
项目进度和健康度等信息填写

项目的进度和健康度等信息分权给关联项目的负责人进行填写，然后实时汇总到项目集。

在项目里，点“设置”，在基本信息界面，可填写所属项目集、项目计划开始和结束时间、项目实际开始和结束时间、健康度、进度和项目类别等信息：

项目里的“设置”界面

项目集里的关联项目列表

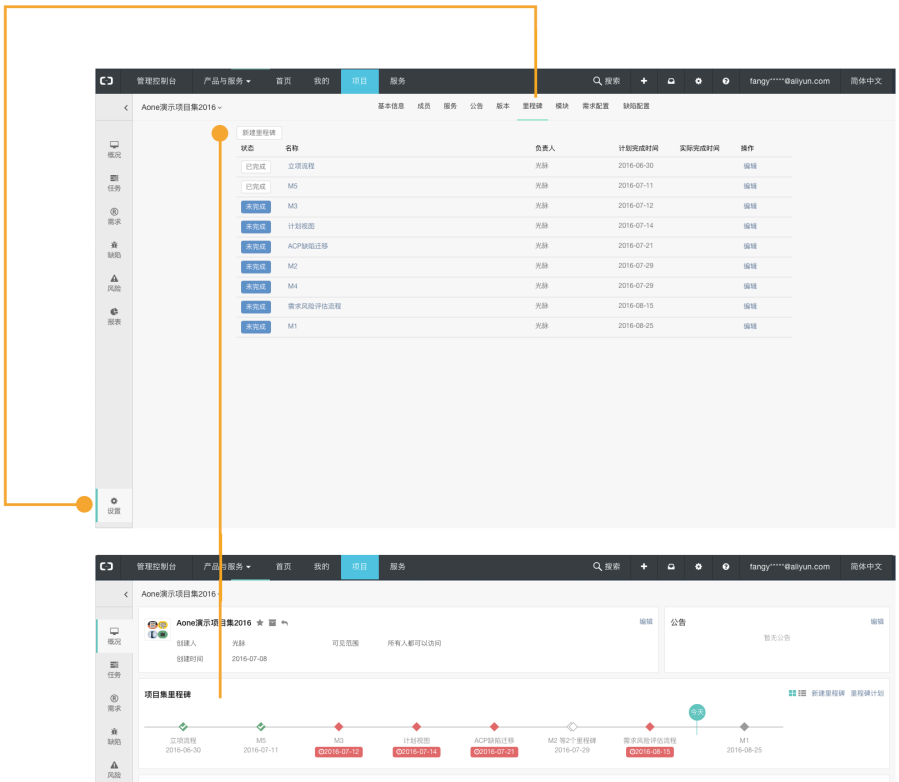


里程碑计划管理

里程碑计划功能让项目集管理者清晰定义项目集目标和任务，并对关联项目里程碑计划进行实时监控。

启用里程碑计划

在项目和项目集里面,点“设置”，然后点“里程碑”TAB，增加第一个里程碑数据后，在项目 / 项目集概况里



面会显示里程碑区块。

管理里程碑和计划任务

延期的里程碑高亮显示，里程碑完成后，高亮消失，这样有助于项目和项目集负责人关注未完成的目标。

在概况的里程碑区块，可以进入详细的里程碑计划页面，在这里可以对里程碑和计划任务进一步按照模块来划分不同的线。在这里，计划任务和在任务墙出现的任务是一致的（注：目前为了避免过多的日常任务进行干扰，只显示和某一个模块关联的任务）。此外，可以从这个视图方便查看关联项目的里程碑和计划。

项目集风险管理

风险汇总

风险汇总区块自动汇总关联项目（在项目集里）或子项目（在父项目里）的未完成风险，并和当前项目集或项目的风险一起显示。

项目集关联项目的风险信息显示

在项目集关联项目列表中，列出每个项目的未完成风险和风险完成占比信息。

代码管理

代码管理概述

code.aliyun.com

阿里云云效旗下的code.aliyun.com提供源代码托管服务。与国外的代码托管服务相比，它快速稳定。与国内的其他代码托管服务相比，它在底层与云效上从需求到开发的众多功能相集成。

在云效中各入口创建的Git库、Git库组，均实际存储在code.aliyun.com。一些设置也是在code.aliyun.com完成，比如个人全局设置。

与云效一样，code.aliyun.com也是使用阿里云账号登录。这意味着，也可以使用阿里云的RAM（主子账号）。

这里是code.aliyun.com的帮助文档的首页。

下面是个人初始设置相关的：

- 要想在本地命令行使用ssh协议（也就是使用类似git@code.aliyun.com:some-grp/some-git.git这样的Git库地址时）访问服务器端，需要在这里配置SSH Key。详见如何创建和添加SSH Keys和Windows下正确配置客户端以使用SSH协议。
- 要想在本地命令行使用http协议（也就是使用类似https://code.aliyun.com/some-grp/some-

git.git这样的Git库地址时）访问服务器端，需要在这里设置用户名和密码。详见相关帮助。**注意**，这里所说的用户名和密码，与访问Web页面所需的阿里云账户和登录密码是不同的概念。

其他常问问题：

- 使用web hooks(英文)

云效的代码服务

云效的代码服务对code.aliyun.com提供的底层服务进行了封装，并引入企业概念。具体来说：

- 管理云效上本企业的代码库和组，或把已有组纳入企业名下，请在RDC的代码服务上完成。详见代码库管理。此外，在使用新建一站式方案向导时，也可能自动创建Git组/库，或把已有Git组纳入本企业名下。
- 管理代码库和组的权限，云效的代码服务也提供了比code.aliyun.com更友好的方式。详见权限管理。它背后也是基于code.aliyun.com的权限模型(英文)。
- 此外，从“我的”->“分支”菜单进入，可以查看自己在各代码库中的分支。而在每个RDC项目里，亦可以从“分支”菜单项进入，配置一个到多个关联到该项目的代码库，于是将显示这个（些）代码库的分支列表等信息。以上，详见分支浏览与操作。

附：Git基础

学习Git，推荐阅读开源电子书Pro Git(中文版)。此外，可参阅下述快捷帮助文档：

- 开始在命令行中使用Git
- 基础的命令行命令
- Git基本命令

code.aliyun.com

code.aliyun.com概述

关于Code界面是英文的说明

首先Code作为研发协同代码托管的基础设施，我们在开源软件基础上进行了分布式改造。着重解决了稳定性、

性能及安全问题，并经过了阿里这样大体量实战检验。升级替换此前老版本的Code，界面也跟阿里内部版本一样只有英文版。对于喜欢英文原版的程序员这是一项福利，避免被不太准确的中文翻译误导的同时，也表达了对开源原版的敬意。对于喜欢中文界面的程序员，我们建议使用RDC-我的-代码。另外，RDC也提供了一站式研发协同的全部工具链，欢迎大家使用。

阿里云Code基础知识指南

一步步学习如何在命令行及阿里云Code上开启工作之旅。点击查看指南。

个人设置

要想在本地命令行使用ssh协议（也就是使用类似git@code.aliyun.com:some-grp/some-git.git这样的Git库地址时）访问服务器端，需要在这里配置SSH Key。详见如何创建和添加SSH Keys和Windows下正确配置客户端以使用SSH协议。要想在本地命令行使用http协议（也就是使用类似https://code.aliyun.com/some-grp/some-git.git这样的Git库地址时）访问服务器端，需要在这里设置用户名和密码。详见相关帮助。**注意**，这里所说的用户名和密码，与访问Web页面所需的阿里云账户和登录密码是不同的概念。

权限管理

code的权限管理已集成到RDC-我的-代码中，请看帮助文档

其他常问问题

使用web hooks(英文)

code容量限制

目前我们提供的单个project的存储上限是1G，一般来讲，Code提供存储1GB代码已经非常大了，只要不存在当云盘滥用的情况，足够个人或团队使用。万一遇到超限，有2种办法，清理误提交的大文件或者通过页面右下角Vone联系我们通过提高限额（目前是免费试用期，可以先提高限额，免费期过后进行收费）。清理Git库中历史上的大文件或含密码文件，参考 https://rtyley.github.io/bfg-repo-cleaner 或者参考 https://github.com/rtyley/bfg-repo-cleaner/issues/65在project页面TAG标签的右边有当前仓库大小值。

code数量限制

每个帐号创建的仓库数不能超过50个。

父子帐号的关系： 所有子帐号创建的库都叠加到父帐号上面； 父帐号创建的库是所有子帐号的集合。

与RDC企业的关系：个人创建的code库数据，与rdc企业没有关系；同一个帐号通过不同的企业创建的库叠加

合并计算。

Key has already been taken 怎么解决

“Key has already been taken”这个提示的意思是“这个key已经在Code平台中添加过了，Key是全局唯一的。”请确认你的机器是否是公用的机器，如果是那一般是其他同学已经加过了。如果你确认其它同学没添加过，那就直接重置SSHKey吧。点击查看添加SSHKey方法

代码服务

代码库管理

概念介绍

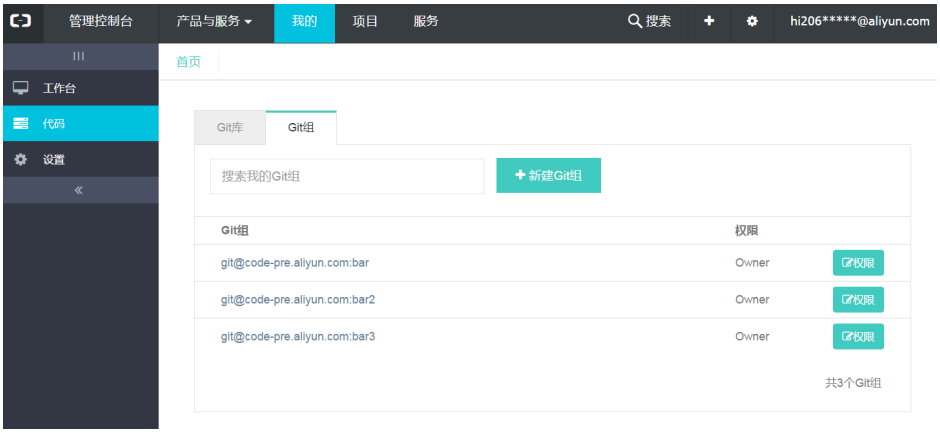
Git库是指托管在https://code.aliyun.com的Git库，即https://code.aliyun.com中的project。

Git组是若干个上述Git库的集合，即https://code.aliyun.com中的Group。

Git组归属到具体某个企业。于是Git组中的Git库也归属到这个企业。

Git库和组的列表

打开代码服务首页，是Git库和Git组两个标签页，分别是Git库和Git组列表。



在这里列出的，是当前用户有权限看到的，且属于当前企业的Git库和Git组。

搜索框用于在Git库或组的列表中搜索。

新建Git库或组

在Git组标签页中，点击“+新建Git组”，输入Git组名和描述后点击“确认”，即可创建一个新的Git组。该Git组属于当前企业。当前用户在该Git组是owner角色。

类似的，在Git库标签页中，点击“+新建Git库”，输入Git组名、Git库名和描述后点击“确认”，即可创建一个新的Git库。

其中，Git组必须是已存在的，属于当前企业的，且当前用户在该Git组中是master或owner角色。

已有Git库或组的权限管理

请点击该Git库或组条目中，“权限管理”按钮，前往权限管理页面。

已有Git库或组的其他操作

请点击该Git库或组条目中，库或组的名称，前往https://code.aliyun.com相应页面。

将已有代码纳入管理

代码已托管在https://code.aliyun.com

假定你的源代码所在Git库名为foo，托管在https://code.aliyun.com上的bar组。由于bar组不属于当前企业，因此在CRP代码服务中看不到该组和该库。在这种情况下，可以这样操作：

第一种方法：把整个代码组归到该企业名下。在代码组列表页面中，点击“关联已有组”，可以把用户自己是owner或master角色，且尚不属于其他企业的代码组，归属到当前企业。

第二种方法：把个别代码库归到该企业名下。第一步，若有必要，通过代码服务页面创建属于当前公司的Git组，比如baz。第二步，登录https://code.aliyun.com，在该Git库的Settings页面下方，进行Transfer project操作，将该库迁移到baz组。由于baz组属于当前公司，该库就属于当前公司，于是在CRP代码服务中就可以看到该库。执行第二步操作时，当前用户必须是bar组的owner角色，以及baz组的master或owner角色。

代码托管在其他Git托管站点

假定你的源代码托管在GitHub上bar组的foo库中。现在打算改为托管到https://code.aliyun.com。

第一步，若有必要，通过代码服务页面创建属于当前公司的Git组，比如baz。

第二步，通过代码服务页面在该Git组中创建一个新的Git库foo。

第三步，将原Git库克隆到本地。

```
$ git clone --mirror git@github.com:bar/foo.git
```

第四步，将本地Git库推送到https://code.aliyun.com。

```
$ cd foo.git
$ git push git@code.aliyun.com:baz/foo.git
```

代码存放在本地

如果已有代码在用户本地，请这样操作：

第一步，若有必要，通过代码服务页面创建属于当前公司的Git组，比如baz。

第二步，通过代码服务页面在该Git组汇总创建一个新的Git库，比如foo。

第三步，把已有代码加入版本控制，并推送到该新建Git库：

```
$ cd existing_folder
$ git init
$ git remote add origin git@code.aliyun.com:baz/foo.git
$ git add .
$ git commit -m "import"
$ git push -u origin master
```

代码库管理员及其权限

代码库管理员当前就是企业管理员对应的人。代码库管理员自动拥有企业所有代码组的owner权限。

权限管理

功能概述

通过权限管理功能，可以查看当前用户自己在特定Git库/组上的权限。当用户在特定Git库/组上有Master或Owner角色权限时，TA还可以查看和修改该Git库/组的其他成员的权限。

概念介绍

Git库即https://code.aliyun.com/中的Project。在Git库这级，可以把一个人设为以下四种角色之一：

- Guest：能看概况，可留言，但看不到源代码。
- Reporter：能看到源代码。
- Developer：可读写，但不能推送(push)改动到受保护分支(protected branch)。
- Master：可读写，甚至推送(push)改动到受保护分支(protected branch)。有一些管理权限，比如管理成员，但不能删除Git库、不能调整Visibility Level等。

Git组即https://code.aliyun.com/中的Group，可以包含若干个Git库。在Git组这级上设的权限，对组内的库都有效。如果一个人既在组上具有角色，又在其中某个库上具有角色，那么在该库的实际权限，取两者中权限高的。

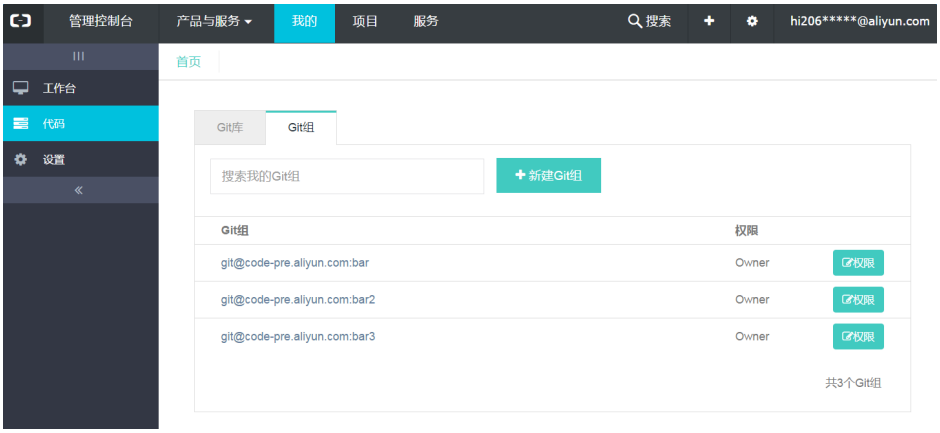
具体说来，GitLab组这级，可以把一个人设为以下五种角色之一：

- Guest：能看各库概况，可留言，但看不到源代码。
- Reporter：能看到各库源代码。
- Developer：各库可读写，但不能推送(push)改动到受保护分支(protected branch)。
- Master：各库可读写，甚至推送(push)改动到受保护分支(protected branch)。有一些管理权限，比如在组中创建新Git库、管理Git库的成员，但不能管理GitLab组的成员、不能删除组或库，不能调整库的Visibility Level等。
- Owner：拥有GitLab组及其所属Git库的所有读写和管理权限。

以上是大致介绍，详细介绍见https://code.aliyun.com的相关帮助文档。

前往特定Git库/组的权限管理页面

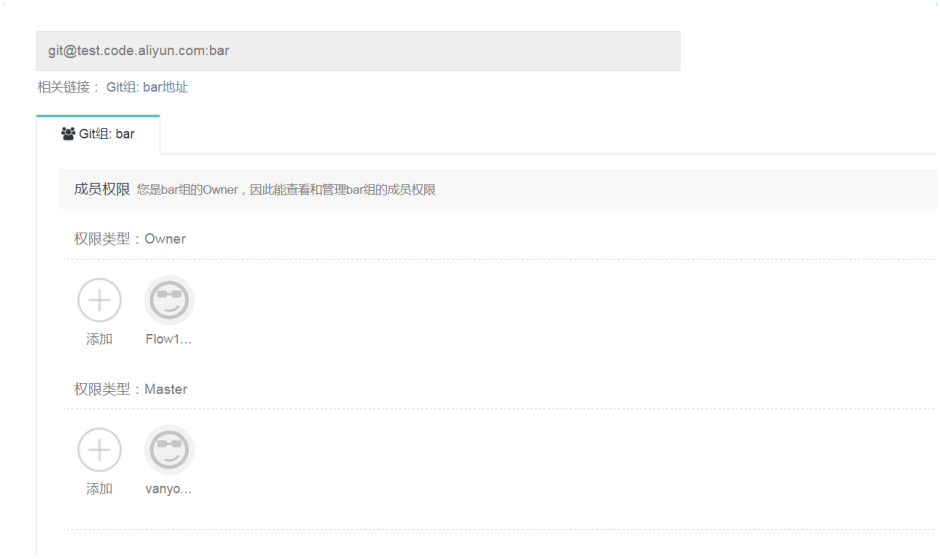
RDC代码服务的首页，显示当前用户所在的Git库/组的列表。每行显示当前用户在该Git库/组的角色，以及“权限”按钮。点击该按钮，进入该Git库/组的权限管理页面。



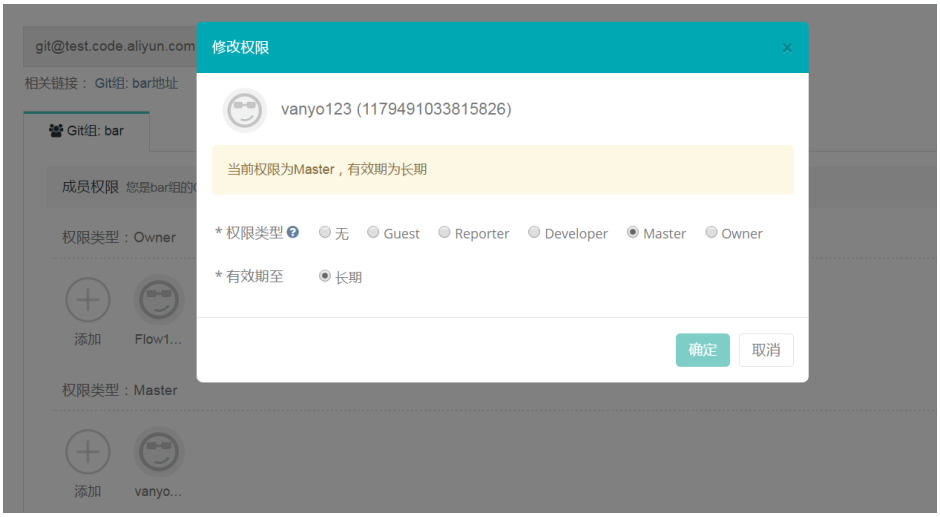
特定Git组的权限管理页面

若当前用户在该Git组上的角色为Guest、Reporter或Developer，将显示用户在该Git组的角色。用户可以操作将自己的权限降低：从Reporter到Guest，或者从Developer到Reporter或Guest。

若当前用户在该Git组上的角色为Master或Owner，那么他不仅能查看和操作自己的权限，还可以查看和操作该组其他成员的权限，或添加新成员。当前所有成员按照角色分组显示：

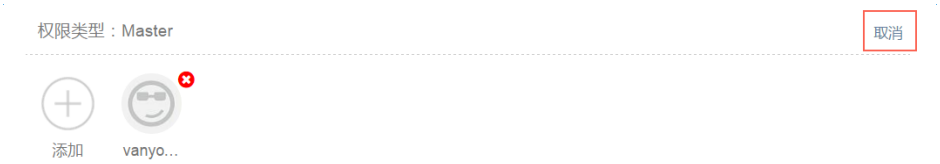


点击成员头像，可查看和修改已有成员权限：



点击左侧“+”号，可添加新成员：

点击右侧“删除成员”，进而点击特定成员头像右上方“-”号，可删除该成员：



特定Git库的权限管理页面

该页面包含两个标签页：该Git库的权限管理，和该Git库所在组的权限管理。后者页面功能类似于上一小节。下面重点介绍前者，即该Git库的权限管理部分。

页面仅显示本人的角色权限信息，还是显示该库所有成员的角色权限信息，取决于当前用户在该Git库的实际权限。即，TA在该Git库的角色（若有），与TA在该Git库所在Git组的角色（若有）中，权限高的。

如果两个角色中，最高权限是Guest或Reporter或Developer，则TA只能看到本人在该Git库的角色，同时可以将自己的权限降级。

如果两个角色中，最高权限是Master或Owner，则TA能看到该Git库上所有成员，并可添加、修改、删除成员权限。

企业的代码管理员

企业的代码管理员默认就是企业的管理员（修改功能即将上线）。本企业名下每当新增Git组时，代码管理员将自动获得它的Owner成员权限。

持续交付

概述

基本概念与原理

欢迎您了解和使用云效的持续交付相关功能！

为了帮助您快速理解和掌握使用方法，本文概要介绍一下相关基本概念和原理。理解了它们，就摸清了云效持续交付的脉络，学习具体内容就会容易很多。

项目

项目是一个“工作场所”。一伙人（或者一个人）为了一个特定的场景（比如开发一个应用/产品），在这个“工作场所”一起办公，方便地使用这个“工作场所”所关联的各种工具和各类数据。这不仅可以包括这个项目相关的需求、任务、缺陷这些工作项并进行迭代排期，也可以包括这个项目对应的代码库、应用、流水线等（详见下文）。典型的情况，一个项目对应一个代码库及其相应的一个应用，但也可能多个代码库和应用。

流水线

流水线的本质是研发-交付的流程，它把流程中的不同阶段和任务串接在一起，并且（可以设置为）自动化地一步一步地执行。简单的例子，手工触发，构建并部署到一个特定的环境，是一条基本的流水线。复杂的例子，源代码提交自动触发，通过各个环节和阶段的构建、部署、各种检测工作，直到上线，是一条完整的端到端的流水线。详情请从流水线概述开始阅读。

代码库

云效文档中提到代码库，是指存放源代码等内容的Git库。代码库托管在阿里云的code.aliyun.com。要想在本地查看、编写和提交源代码，请首先在您本地机器上安装Git，并设置ssh key以便连接到code.aliyun.com服务器端。更多内容请参考代码服务概述。

构建

构建指根据代码库中的源代码编译构建打包，它是流水线上的一个任务。云效根据Git库中源代码根目录下的<应用名称>.release文件构建打包，以便部署并运行。<应用名称>.release文件用键-值对儿的形式定义了如何把源代码构建打包。详情请从构建概述开始阅读。

应用

从源代码的角度看，源代码存放在一个个代码库里。而从部署运行的角度看，一个个Web应用被部署并运行起

来，相互配合，实现Web网站的功能。每个Web应用，通常对应了根据一个代码库里的源代码构建生成的包，以及它运行时相关的基础软件和环境的设置等等。它是部署运维的一个(最小)单元。这是应用这个概念的由来。

环境

每个Web应用，在集成测试的环境（在云效中通常称作日常环境）、预发的环境（称作预发环境）、对外提供服务的环境（称作正式环境）等不同的环境里运行。每个环境中，该应用运行在若干台机器（虚机/容器）上。部署时，可能分期分批。每台机器的部署，有特定的方法和步骤。这些都是定义在这个应用的特定环境上。

发布与部署

在云效中，发布是指一次部署以及相关的一些操作，它是流水线上的一个任务。如前文所述，发布和部署的配置，是在应用的环境上。应用、环境、发布与部署的详情，请从应用部署概述开始阅读。

上手步骤

在不同的使用场景里，配置和使用云效进行软件开发和发布的方法略有不同，但它们遵循基本相同的大致步骤：

新建和配置企业

新建和配置企业的一般方法见企业信息和成员管理。

一些场景下，还需要进一步做一些的企业级设置。比如，

- 如果使用脚本（而不是通过EDAS服务或容器服务）来部署Web应用，需要在“企业设置”->“机器管理”中配置机器资源以供各应用使用。详见部属配置：通过脚本部署中的相关描述。
- 如果使用容器服务来部署Web应用，需要在“企业设置”->“容器服务账号”中设置。详见部属配置：通过容器服务部署中的相关描述。

具体有哪些场景，需要做什么企业设置，请到下文这里查看。

使用向导创建一站式解决方案

向导可以帮助你创建(或关联)包括项目、代码库、应用、流水线在内的一站式解决方案。于是，从需求到发布上线的管理和操作，都可以在此一站式完成。

在使用向导的过程中，选择不同的选项，产生适合不同场景的配置。比如，是Web应用还是无线应用，是通过脚本部署还是通过EDAS或容器服务部署，等等。

一站式方案的大部分配置，可以在使用向导创建的过程中自动完成。

以上，详见新建向导概述。

更多配置

在不同场景下，还有一些配置信息，需要在使用一站式创建向导后，继续设置或调整。随着时间的推移，在试用过程中也可能想增加或调整一些配置。

一般而言，使用一站式创建向导后，依如下步骤检查/调整配置：

1. 先配置好构建部分（可能包括Docker image的生成），详见构建概述。运行默认流水线直到此步骤能成功。
2. 然后配置日常环境的部署，详见部署配置概述。运行默认流水线直到此步骤能成功。
3. 配置预发（如果没有可删除流水线上对应步骤）、正式环境的部署，运行默认流水线直到此步骤能成功。
4. 考虑配置流水线，增加、修改、删除流水线上的节点。比如增加自动检查和测试等任务。详见流水线概述。

日常使用

日常使用云效做集成和发布，主要是在流水线上完成。详见流水线概述。

另外，当“开发模式”选择使用“分支模式”（而不是“自由模式”或“Git Flow模式”）时，分支上的操作显得比较重要。因为每一条feature分支在开发完成后，需要执行“提交待发布”操作。这个操作，可以在该分支详情页点击完成，也可以在分支列表页面直接点击完成。详见分支模式和分支模式下的流水线。

不同场景下的上手方法

以下是一些典型场景的上手方法：

- 典型场景：通过脚本部署Web应用
- 典型场景：通过EDAS部署Web应用
- 典型场景：通过容器服务部署Web应用
- 典型场景：无线应用

典型场景：通过脚本部署Web应用

当在创建向导中选择了创建Web应用，并选择了通过自定义脚本部署时，有如下配置请检查修改或添加：

构建配置

源代码根目录下的<应用名称>.release文件是构建的配置。一般来说，它已经自动生成好，特定情况需要进一步补充。请参考Web应用构建配置。构建配置的更多内容请从构建配置概述开始了解。

部署配置

需要把机器资源添加到云效上本企业名下。机器资源通常是阿里云ECS，请先行购买。此外也支持使用其他途径的机器资源。

随后，在具体应用的具体环境中，配置本企业名下的机器。

除了配置机器资源，还需要配置/调整部署的批次，以及一台机器上部署的过程，比如使用的启动脚本和停止脚本。

根据创建向导中的选项，日常环境可能已为您关联了临时免费使用的机器。请替换。并继续配置预发环境和正式环境。

以上，详见部署配置：通过脚本部署。部署配置的更多内容请从部署配置概述开始了解。

其他内容

请参考上手步骤。

典型场景：通过EDAS部署Web应用

当在创建向导中选择了创建Web应用，并选择了通过EDAS部署时，有如下配置请检查修改或添加：

构建配置

源代码根目录下的<应用名称>.release文件是构建的配置。一般来说，它已经自动生成好，特定情况需要进一步补充。具体请参考使用EDAS部署时的构建配置。构建配置的更多内容请从构建配置概述开始了解。

部署配置

需要在EDAS系统中，为该应用的每个(打算使用EDAS的)环境，分别创建和配置EDAS应用。

随后，在云效中，把该应用的每个(打算使用EDAS的)环境，配置为关联到相应EDAS应用。

以上，详见部署配置：通过EDAS部署。部署配置的更多内容请从部署配置概述开始了解。

其他内容

请参考上手步骤。

典型场景：通过容器服务部署Web应用

当在创建向导中选择了创建Web应用，并选择了通过容器服务部署时，有如下配置请检查修改或添加：

构建配置

源代码根目录下的<应用名称>.release文件是构建的配置。一般来说，它已经自动生成好，特定情况需要进一步补充。注意其中不仅要构建产生包，还要进一步生成Docker image。请参考Docker镜像构建配置。构建配置的更多内容请从[构建配置概述]https://help.aliyun.com/document_detail/59292.html)开始了解。

部署配置

你需要在阿里云容器服务中已经有集群，应用和服务。随后在云效上，导入集群证书。

在具体应用的具体环境页面中，点击部署配置，选择该环境要发到容器中对应的集群、应用、服务。

以上，详见部署配置：通过容器服务部署。部署配置的更多内容请从部署配置概述开始了解。

其他内容

请参考上手步骤。

典型场景：无线应用

当在创建向导中选择了创建无线应用时，有如下配置请检查修改或添加：

构建配置

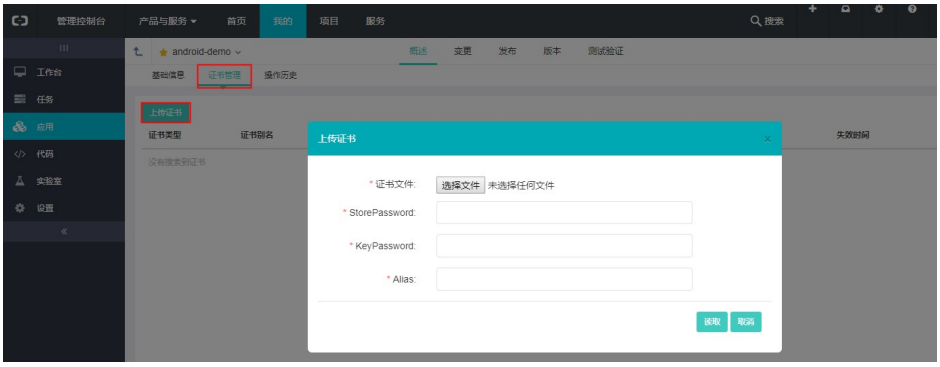
源代码根目录下的<应用名称>.release文件是构建的配置。一般来说，它已经自动生成好，特定情况需要进一步补充。请参考无线应用构建配置。构建配置的更多内容请从构建配置概述开始了解。

上传证书

Android需要证书才可以完成apk打包，可以在应用 概述-〉证书管理 中上传和管理证书。上传的证书会在构建打包时传递给打包脚本。

选择无线应用：

上传证书：



其他内容

请参考上手步骤。

新建向导

新建向导概述

在云效中，有多种途径来新建从需求到发布上线的一站式的研发协同解决方案。

其中，对于刚开始接触云效用户，首页的“快速开始”是开始了解和使用云效的最佳选择。详见新建：快速开始以及云效快速入门中的介绍。

首页的“更多选择”向导则提供了更多选择。这包括，可以选择使用一个已有的代码库而非总是新建一个；可以把Wweb应用打包为Docker镜像并使用阿里云容器服务部署；可以通过EDAS部署Web应用；可以构建发布无线应用而不仅仅是Web应用，等等。详见新建：更多选择

除了首页这两个主要入口，还有其他一些入口，起不同作用。详见新建：其他入口.

新建：快速开始

从吊顶中点击“首页”，进入首页后，点击“快速开始”，依步骤配置。

其中，第二步“配置代码库”中，开发模式请选择“自由模式”，初次接触云效，这是最容易上手的选择，因为流程最简单。“构建并部署到临时演示环境”也请确保勾选。

点击完成，将自动进行如下工作：

新建一个项目。

新建一个代码库并关联到这个项目。向该代码库中填充了样例代码，以及构建配置。

新建一个应用，与这个代码库对应。应用也关联到这个项目。完成这个应用及其环境的基本配置。其中，免费临时借给用户一台机器，并配置到了日常环境，用于演示。

新建一条端到端的流水线。运行这条流水线，构建并部署到日常环境。

此时，可以到界面给出的演示地址，查看应用运行的页面效果。也可以进入项目或流水线，查看配置和运行情况。

当前仅为日常环境配置了供演示用的临时机器。要想配置平时用的机器，以及为预发、正式环境配置机器，请参考完善配置：通过脚本部署Web应用。

更多内容，请从完善配置：概述读起。

新建：自定义配置

从吊顶中点击“首页”，进入首页后，“快速开始”按钮的右侧，有“自定义配置”按钮。它也是一个新建一站式解决方案的向导。与“快速开始”向导的区别是，有更多选项，对应更多场景，可以按需进行选择。这包括：

- 不仅可以新建项目，也可以在已有的项目中追加代码库和应用。
- 不仅可以新建代码库，也可以使用通过其他途径创建的代码库。
- 可以选择更多的开发模式。详细说明
- 不仅可以新建Web应用，还可以新建无线客户端应用（即将上线）。
- 可以选择更多编程语言、JDK版本、Maven版本、源代码初始化模板。
- 不仅支持自定义脚本部署到机器，也支持通过EDAS部署，以及构建打包为Docker image后通过容器服务部署。（即将上线）

其中的一些选项组合，继续提供了“构建并部署到临时演示环境”这个能力，可以勾选。

对于一些选项组合，在完成这个向导后，可能还需要进一步的配置。具体内容详见完善配置：概述。

新建：其他入口

点击吊顶上的“首页”，进入首页后，“快速开始”和“更多选择”这两个按钮，是新建一站式研发协同方案的主入口，根据向导，配置好项目-代码-应用-流水线。还有一些入口，本文一并介绍：

在应用列表页点击“注册应用”

点击吊顶的“我的”，随后点击左侧菜单项“应用”，进入应用列表页。其中有“注册应用”按钮。

当前，点击“注册应用”按钮，将进入新建应用向导，它将新建或关联代码库，新建流水线，但不会关联到项目。

我们即将改进它的功能，可以新建项目或关联已有项目。

在一个已有项目中新建或关联应用/代码库/流水线

在具体项目的概述页，可以看到名为“配置代码库”的一个小卡片。点击它，将进入类似于“更多选择”的向导。唯一的区别是，不必选择项目，必然是在本项目中新建或关联代码库、应用、流水线。

类似的，在具体项目的应用列表中点击“注册应用”，也是在本项目中新建或关联代码库、应用、流水线。

而点击具体项目的分支菜单项进入后，点击“配置代码库”（若当前该项目还没有关联代码库）或点击tab页标签旁的“+”号，，也是在本项目中新建或关联代码库、应用、流水线。

仅新建代码库

点击吊顶的“我的”，随后点击左侧菜单项“代码”，进入代码库列表页。在这里点击“+新建代码库”，当前仅会新建一个空的代码库。

仅新建项目/项目集

以下入口都只会新建项目/项目集，不会新建或关联代码库/应用/流水线：

点击吊顶上的“项目”，进入项目列表页，点击“新建项目”或“新建项目集”。

点击吊顶上的“+”号，进而选择“新建项目”或“新建项目及”。

点击吊顶上的“首页”，在下部“解决方案”区域，选择“启动Scrum”项目或“大型项目集管理”。

流水线

流水线概述

流水线的本质是研发-交付的流程，它把流程中的不同阶段和任务串接在一起，并且（可以设置为）自动化地一步一步地执行。

简单的例子，手工触发，构建并部署到一个特定的环境，是一条基本的流水线。

复杂的例子，源代码提交自动触发，通过各个环节和阶段的构建、部署、各种检测工作，直到上线，是一条完整的端到端的流水线。

学习使用流水线，请首先阅读流水线的运行，这里讲解了流水线的概念原理，以及如何使用。

通过一站式方案的新建向导创建的项目，通常已经自动生成了一条流水线。可以查看它的配置，修改补充，或者新建一条流水线。流水线的配置方法详见这里。

流水线上不同类型的流程任务，是通过不同的插件实现的。当前已有：

- 构建：构建打包，供部署使用。
- 部署：把构建成果部署到服务器运行。
- 人工卡点：需要人工判断是否OK的任务。可以用来作为流水线上人工测试、安全审核等流程卡点。

- 单元测试：自动运行一行命令，看是否能成功。

我们将研发更多插件，以支持更多类型的任务。同时，也将考虑开放插件接口，让用户可以自定义插件，放入流水线中。

以上，是对自定义流水线的介绍。目前当开发模式选用分支模式时，流水线还不是可以完全自定义的。相应情况请参考这里。

流水线的运行

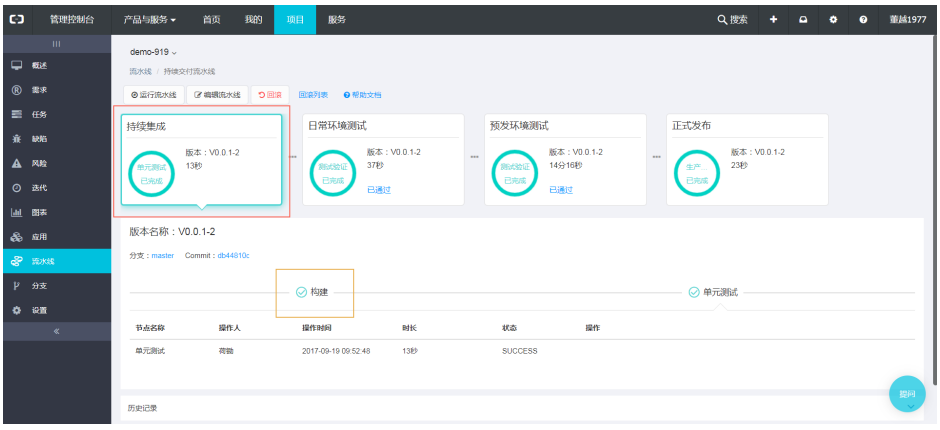
本文介绍流水线的使用方法。关于云效流水线服务的概要介绍，请参考这里。关于流水线的配置，请参考这里。

入口

当进入一个项目后，左侧菜单栏的“流水线”菜单项，是流水线的入口，点击打开流水线列表页，可选择进入具体流水线页面。

而如果左侧菜单栏没有出现“流水线”菜单项，请前往“设置”->“服务”为该项目配置流水线服务。

阶段与任务



图中，红色框“持续集成”是一个阶段，跟“日常环境测试”、“预发环境测试”、“正式发布”几个阶段串接在一起，形成整个流水线。

而每个阶段，可以包含若干任务。比如，“持续集成”阶段，包括黄色框“构建”这个任务，以及“单元测试”这个任务。两个任务串接在一起，是“持续集成”阶段的全部内容。

流水线的启动运行

可以把一条流水线设置为代码提交后自动触发、定时触发或手动触发。即使设置为前两种方式之一，也仍然可以手动触发。手动触发的方法是点击右上角的“运行流水线”按钮。

阶段的启动运行：设置为自动触发时

各阶段之间，可以设置为自动触发或手动触发。下面分别介绍。

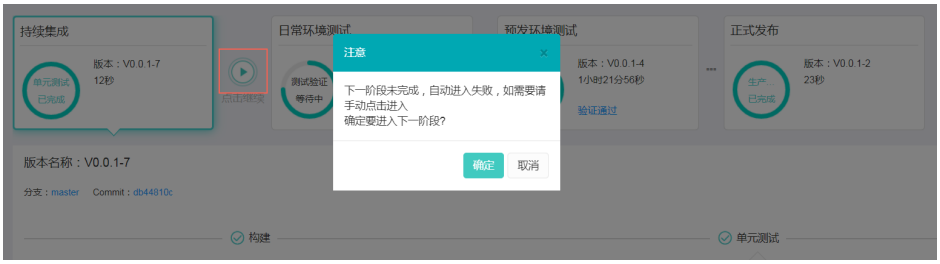
设置为自动触发时，当前一阶段完成且后一阶段空闲时，就会自动开始运行后一阶段。

当前一阶段完成时，若后一阶段正在运行，则后一阶段不受干扰，继续运行完成。待运行完成，出现空闲时，后一阶段自动启动，接续前一阶段版本运行。

若在后一阶段运行时，前一阶段完成了多次运行，则在后一阶段出现空闲时，接续前一阶段的最近一次运行版本运行。

举个例子：假定运行日常环境测试阶段需要一个小时，而运行持续集成阶段需要五分钟。在某个小时内，有三次代码提交，触发了持续集成阶段的三次运行。持续集成阶段第一次运行结束，触发了日常环境测试阶段的自动运行，将持续一小时。持续集成阶段第二次运行结束，由于日常环境测试阶段仍在运行，因此不会立即启动该阶段再次运行。持续集成阶段第三次运行结束，由于日常环境测试阶段仍在运行，因此也不会立即启动该阶段运行。待到日常环境测试阶段运行结束后，空闲下来，开始再次运行，此时是接续持续集成阶段第三次运行的版本继续运行，而不是接续持续集成阶段第二次运行的版本继续运行。

亦支持在前一阶段运行完成时，若后一阶段正在运行，强制中止后一阶段当前运行，以便后一阶段接续运行前一阶段最新版本。具体方法是，此时点击两个阶段之间的开始按钮（下图红框），然后在弹出框中点击确定：



阶段的启动运行：设置为手动触发时

设置为手动触发时，当前一阶段完成后，在两个阶段之间出现人工触发按钮，点击开始运行后一阶段。

此时，如果此时下一阶段正在运行，将弹出对话框确认中止正在运行的内容。

如果在下一阶段前，上一阶段已运行多次，则人工启动的下一阶段运行时，使用上一阶段最近一次运行的版本。

任务的启动运行

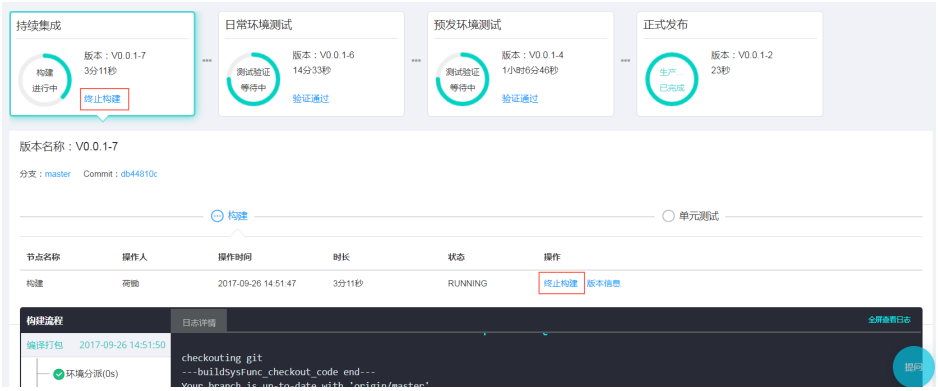
一个阶段内的串联的各任务，依次运行。前一个任务的完成，自动触发下一个任务的运行。

一个阶段内串联的各任务依次运行完毕后，该阶段才会再次运行。因此不会出现串联的各任务同时运行的情况。

另外，任务间并联，以便同时运行的功能已在开发中，敬请期待。

中止任务运行

有些类型的任务，提供了中止任务运行的方法。相应操作出现在流水线上的阶段中，和/或页面下方的任务中。以构建组件为例，见下图两个红框：



需要人参与交互的任务

任务既可以是完全自动运行的，也可以是需要人工参与的。比如，人工测试、人工发布审核等等。

此时，可以在流水线的相应阶段上，或下方相应任务中，进行人工操作。如图两个红框：

任务运行失败时的处理

当有一个任务运行失败时，该阶段的该次运行即停止。不会再触发下一阶段的运行。也就是说，本次流水线运

行至此停止。

此时，如果分析是配置或环境的问题，可以考虑在修复问题后，再次运行。具体操作为，点击流水线上的“重试”按钮。如下图红框：

一个小技巧：如果希望任务运行失败时流水线能够继续运行，可以考虑改变任务的配置，让它总是向流水线返回成功。某些类型的任务能否实现这个方法，具体配置随任务类型的不同而不同。举例来说，对于单元测试组件，可以让自定义的命令行总是返回0。

附原理说明：任务间如何传递“版本”

前面提到，后一阶段接续前一阶段的版本继续运行。那么，“版本”具体是什么，如何承载呢？

流水线的第一个阶段的每次启动，都将会产生一个版本记录。典型的，包括源代码版本信息、相应包版本信息等等。该记录会逐任务逐阶段传递下去。

此外，每种任务类型，可以定义完成时向流水线上下文中，输出哪些参数。同时可以自定义开始时从流水线上下文中，获取哪些参数的值。据此，可以实现灵活的参数传递。

流水线的配置

本文介绍流水线的配置。若您尚不熟悉云效流水线服务，推荐从流水线概述开始阅读。

流水线的添加、修改与删除

当进入一个项目后，左侧菜单栏的“流水线”菜单项，是流水线的入口。（如果左侧菜单栏没有出现“流水线”菜单项，请前往“设置”->“服务”为该项目配置流水线服务。）

点击“流水线”菜单项，进入流水线列表页。此时左上角，有“新建流水线”按钮，可添加新的流水线。

流水线列表的每一行，末尾有“修改”按钮，点击进入该流水线的编辑页面，可修改该流水线的配置。而如果点击流水线列表每一行左侧的流水线标题，进入该流水线的主页面后，也可以点击主页面左上角的“编辑流水线”按钮，进入该流水线的编辑页面。如下图：

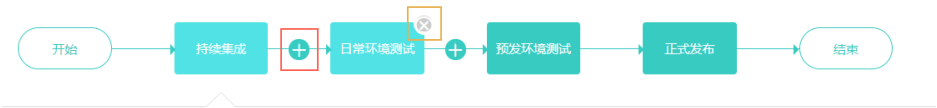
在该流水线的编辑页面，可配置如下内容：

- 流水线名称。
- 流水线的管理员。仅管理员可编辑流水线。新建流水线时，当前用户被设为管理员。
- 监听设置。其中，自动触发是指，在源代码的修改被推送到服务器端代码库指定分支时，触发流水线运行。另外，配置为自动触发或定时触发后，在流水线主页也仍然可以手工触发流水线执行。
- 流水线各阶段的添加、修改与删除。详见下节。

流水线删除功能待上线。

阶段的添加、修改与删除

在流水线编辑页面的中间部分，显示流水线各阶段。当把鼠标移动到两阶段之间的连线时，页面显示两阶段间的加号图标（下图中红色框），和一个阶段的删除图标（下图中黄色框）。点击可分别新建阶段或删除一个已有阶段。



要想修改一个阶段，请用鼠标选中该阶段。于是下方将显示该阶段详情，可进行该阶段任务的添加、修改和删除，详见下一节。另外，阶段的名称也是可以修改的，只需再显示阶段名称的地方编辑即可。

任务的添加、修改与删除

选中一个阶段后，在下方按顺序显示该阶段的各已有任务，同时可以添加任务：

点击“+添加任务”，可以添加一个新任务。其中第一步是选择新任务的类型：

选中一个已有任务，在任务条目右侧，将展开该任务的配置，进而可以填写和修改。不同任务类型，其配置内容是不一样的。但通常有如下两项：

- 任务类型。这一项是在新建任务时确定的，不可修改。
- 任务名称。在流水线编辑时和运行时，都将显示该字段，标识这个任务节点。

点击每个已有任务上的删除图标，将删除该任务。

当前可选任务类型

构建

构建打包，供部署使用。这部分相关知识较多，请从构建概述读起。其中，流水线上构建任务的配置和运行，详见这里。

自定义脚本

如果您有比较定制化的需求（比如向使用自定义脚本发布静态资源 ，或者执行一些定时任务），那么您可以使用自定义脚本。详见这里。

部署

把构建成果部署到服务器运行。这部分相关知识较多，请从应用部署概述读起。其中，流水线上部署任务的配置和运行，详见这里。

人工卡点

需要人工判断是否OK的任务。在流水线运行时，需要处理人到页面点击是否OK。这类任务可以用来作为流水线上人工测试、安全审核等流程卡点。

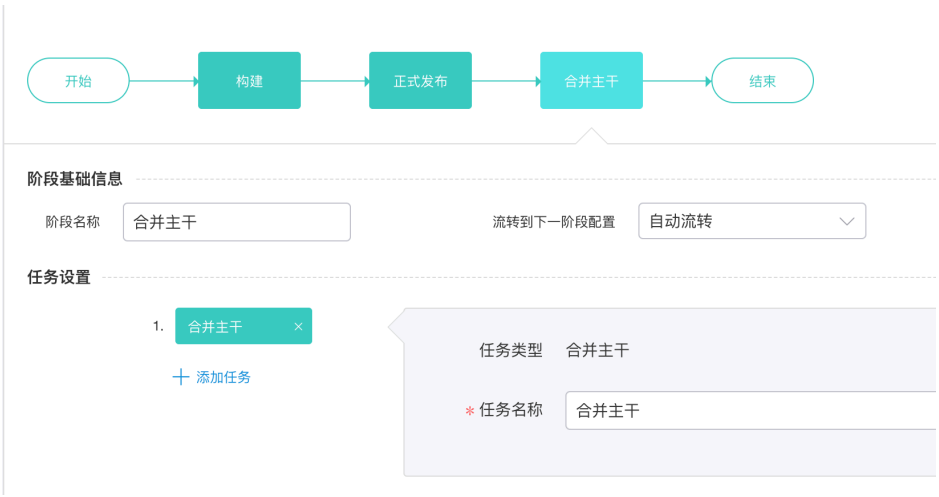
其中，处理人可以配置为具体人员（可多个），也可以配置为流水线所关联应用上的角色（仅可选一个）。

单元测试

自动运行一行命令，看是否能成功。

合并主干

您可以使用该组件将某个分支合并到主干（master），如下图所示：



系统账户

默认情况下，流水线上的与代码有关的操作（如clone，pull，push等）使用的都是当前操作人的权限。但对于合并主干的场景，不适合使用当前操作人权限。比如很多开发团队会把主干设置为保护分支，大部分开发人员没有权限进行代码push。对于这种情况，云效提供了系统账户作为解决方案。您可以在企业设置->代码托管中配置该账户。

设置系统账号之后，当企业内新建代码组时，本账号将自动成为该代码组的owner。对于已存在的代码组，请自行保证其有拥有相关权限（比如，更改系统账户时，需要手动对已存在代码组进行owner组权限赋权）。

合并主干异常排查方法

如果合并主干出现了异常，请按照如下方式进行排查。

1. 确认当前企业是否配置了系统账户。
2. 配置了系统账户时，请查看当前系统账户，是否拥有当前操作代码组的owner权限。如果没有请赋权。赋权之后，进行重试操作。
3. 未配置系统账户时，如果clone失败，请确认当前操作人拥有当前代码库权限；如果push失败，请确认当前操作人拥有当前代码库master权限或当前代码组owner权限。如果没有，请赋权之后进行重试，或者请拥有相应权限的用户进行重试操作。

分支模式下的流水线

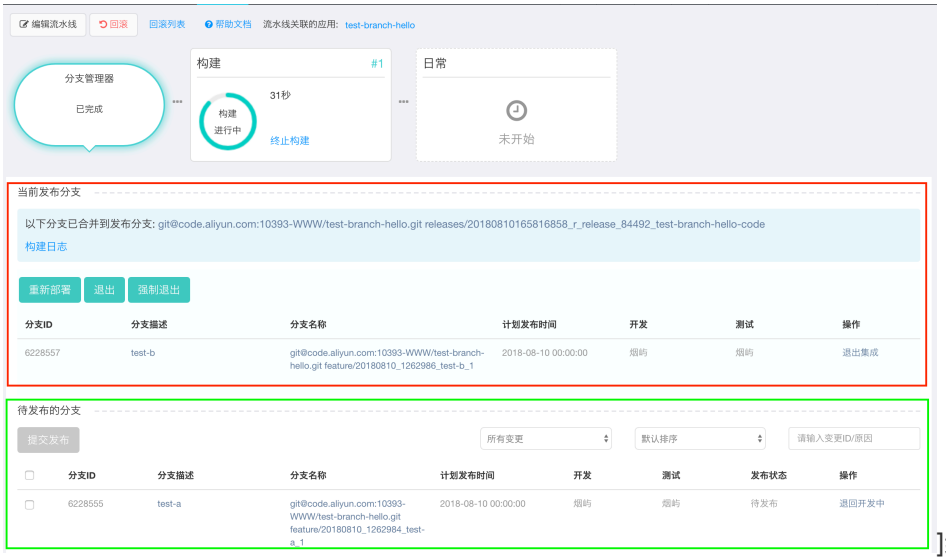
当使用向导新建一站式研发解决方案时，若研发模式选择了分支模式，则向导将自动创建分支模式所特有的流水线。

阅读本文之前，需要首先学习理解开发模式中的分支模式。详细介绍见这里。

分支模式流水线的特殊之处

分支管理

在分支模式下，feature分支（或特性分支）承载了单个缺陷或功能的开发，而release分支是用于集成和发布的。分支模式的流水线使用分支管理器来管理这两种类型的分支。在构建前，分支管理器会创建一条release分支，并将发布列表中的分支都合并到release分支上，然后用release分支的内容进行构建部署。【分支管理器】页面，有两个区域：当前发布分支和待发布的分支。



]绿色方框内展示的是待发布的分支列表，是所有已经开发完毕并做了适当检测，可以进行集成和发布的feature分支列表。做集成和发布时，就从这个列表中挑选，哪些合并到当前流程对应的release分支，以便部署和进一步集成测试。红色的方框展示的是当前发布的分支列表，就是是从待集成区中挑出来的，(打算)合并到release分支，并随后部署到当前流程相应的运行环境的feature分支列表。这个列表，反映的是当前环境中，包含了哪些代码改动。这些改动将被放在一起测试，进而发布上线用户仅需在页面上维护这两个feature分支列表，系统将自动完成release分支的创建和管理，feature分支到release分支的合并等一系列工作。

合并到主干

默认创建的分支模式流水线里面都有构建和部署（日常/预发/正式）两个阶段。正式的阶段包含两个任务【部署】和【合并主干】。【合并主干】任务执行的动作是将release分支的代码合并入master分支。

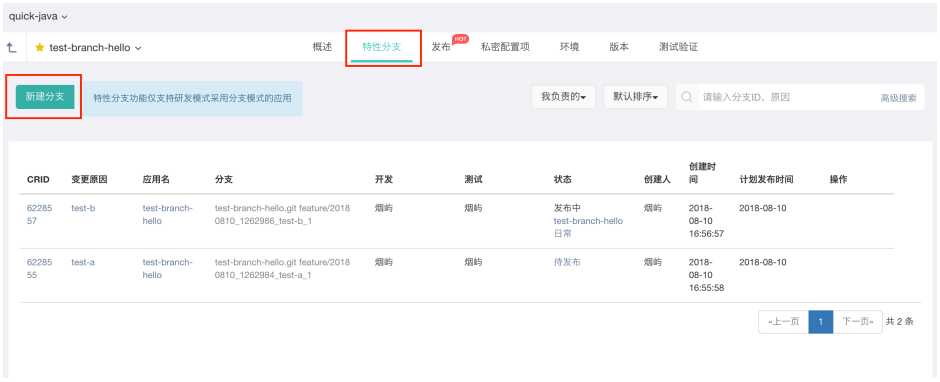
配置流水线的入口和方法

新建入口与其他流水线相同，当新建流水线时选择了分支模式的应用对应的代码库，系统会自动创建包含【分支管理器】的分支模式流水线。流水线的其他配置详见文档

日常操作

创建特性分支

创建特性分支是开始开发工作的第一步，打开应用的详情页面后，进入【特性分支】tab，点击【新建分支】按

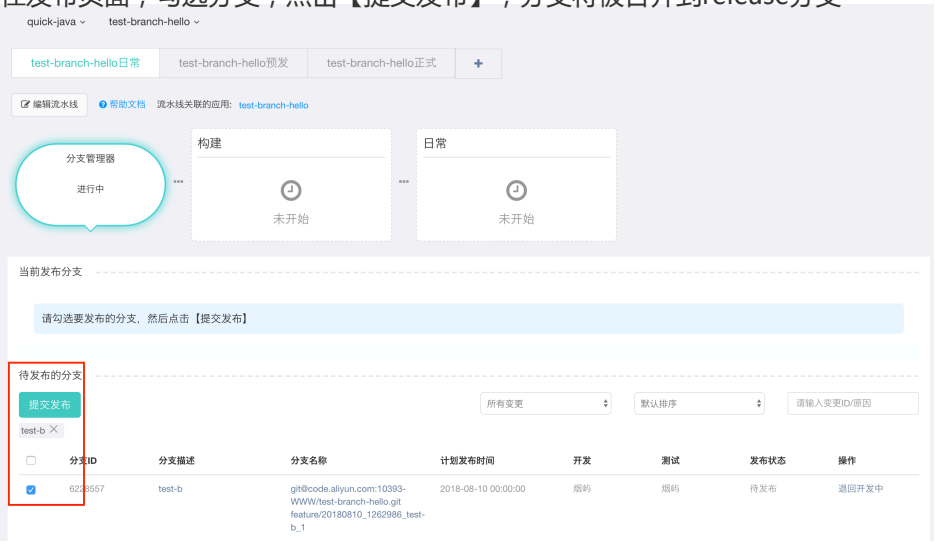


钮即可创建特性分支。

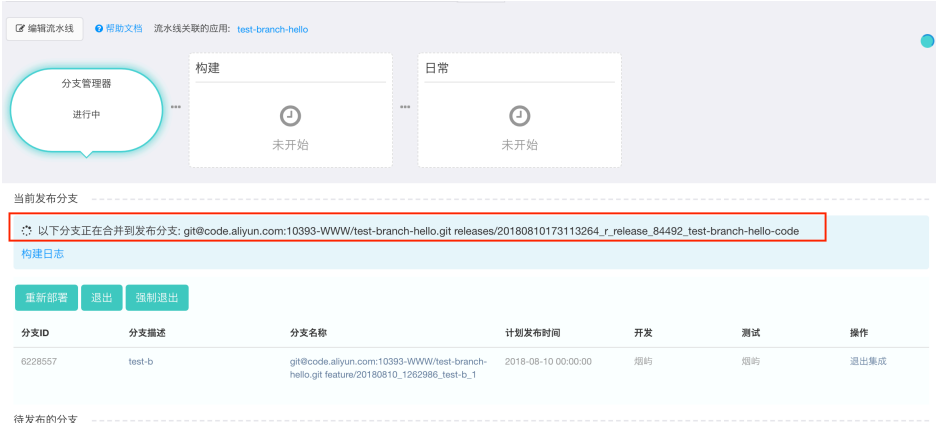
挑选分支合入release分支

特性分支创建后，默认状态是”开发中”。每个分支左侧，有“提交待发布”按钮。点击可将该分支状态置为“待发布”，也就是说，标记该特性分支已开发完毕并做了适当检测，可以进行集成和发布了。于是，该特性分支就进入了待集成区在特性分支页面，点击【提交集成】按钮，分支将进入待发布区，页面跳转至发布页面

在发布页面，勾选分支，点击【提交发布】，分支将被合并到release分支



提交后，页面将展示新创建的release分支信息



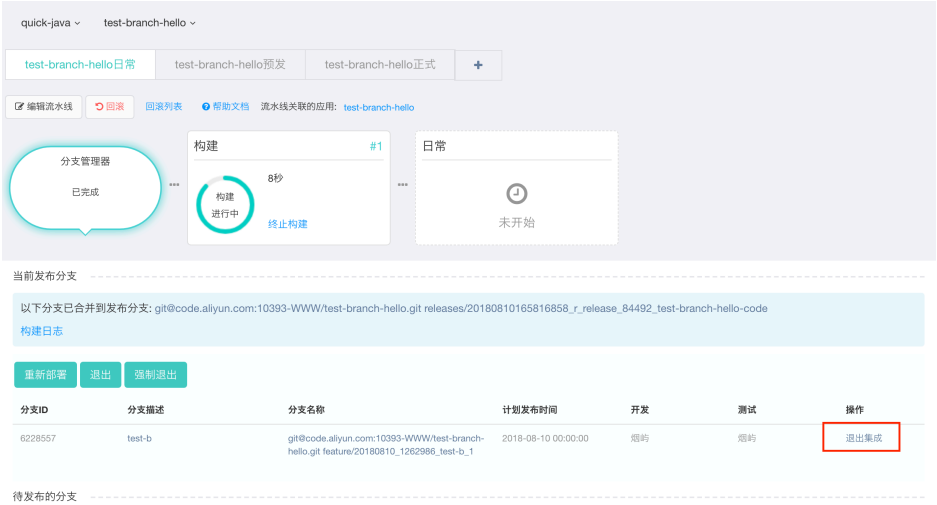
feature分支更新后，更新release分支并部署

feature分支有新提交后，【分支管理器】页面有红色提示信息，点击【重新部署】按钮将更新release分支内容并触发部署动作：

将某个特性分支的内容从集成中删除

当某个分支确定不需要部署到环境，在【分支管理器】页面，在当前发布列表右侧操作一栏，点击【退出集成】，分支将回到待发布列表，系统将：

1. 基于master分支，创建新的release分支
2. 除了该特性分支外的其他在发布区的分支合并到release分支
3. 把release分支的最新内容部署到环境



使用流水线进行无线应用构建

本文讲解如何在流水线页面进行无线应用构建打包。

关于流水线的详细说明，请参见流水线概述

关于如何配置无线应用构建，请参见无线应用构建配置

发起构建

首次运行时，在无线客户端应用的发布页面，点击构建配置按钮，选择构建构建的分支，打包的版本号，以及打包类型。现在支持测试包、正式包两种类型，并支持同时打包。点击确定按钮后开始构建打包。
打包过程中可以在屏幕下方查看打包脚本执行情况。

查看下载打包结果

打包结束后，可以页面中直接查看构建产物，通过链接下载生成的apk包，同时提供手机扫码下载的功能。

也可以在这里点击下载详细的打包日志。

用实验室测试生成的包(可选)

打包完成后，会自动流转到第二阶段无线实验室，对打出来的包进行测试。测试结果会显示在当前页面上。

如果需要进行测试需要首先配置测试实验室。如果没有配置，这一阶段会自动跳过，并标记为成功。默认情况下是没有无线实验室相关的配置的。

关于如何无线测试实验室，请参见持续交付和质量红线 以及自动化测试与集成

构建

构建概述

简单说来，云效流水线上的构建任务以及特性分支和分支集上的构建任务，根据指定Git库源代码根目录下的<应用名称>.release文件，进行构建打包工作，以便随后流水线上的部署任务进行部署。

<应用名称>.release文件，是用键-值对儿的形式定义了如何把源代码构建打包，在什么样的构建环境中打包，等等。。它的完整语法见可配置键的完整列表

有时我们需要构建产生不同内容的包，用于不同的运行环境（比如集成测试环境和生产环境）。甚至，为某个环境构建产生压缩包而为另一个环境产生Docker镜像。还有的时候，我们希望在构建时使用一些当时构建上下文的参数，比如构建时间、源代码分支名称等。云效支持这样的场景：

- 在流水线上的构建组件，支持一些相关的高级配置。详见流水线上的构建任务。
- 特性分支和分支集上的构建任务可以选择包标签。

- 构建过程可以受输入参数的影响。详见使用参数影响构建行为。

还有的时候，我们有一些私密配置信息，不适合与源代码存放在一起。云效提供了存放私密配置项的功能，详见这里。

以下文档给出了一些典型场景下配置构建的方法：

- Web应用构建配置
- 使用EDAS部署时的构建配置
- Docker镜像构建配置
- 无线应用构建配置

关于Maven仓库，目前云效使用全局的Nexus仓库maven.aliyun.com，供下载。若需要上传，企业可考虑搭建并使用企业私有的Maven仓库，详见在云效中使用私有Maven仓库。

流水线上的构建任务

构建任务，是云效流水线上的一类任务，它负责构建打包，供后续的部署任务使用。

构建任务的运行

构建任务一般不需要在运行时输入信息，就会自动运行。运行期间和运行结束后，可以在页面下方查看构建日志：

构建任务的配置

基本配置

在流水线编辑页面，添加任务时，请选择“构建”，并填写其基本配置：

高级配置：使用包标签

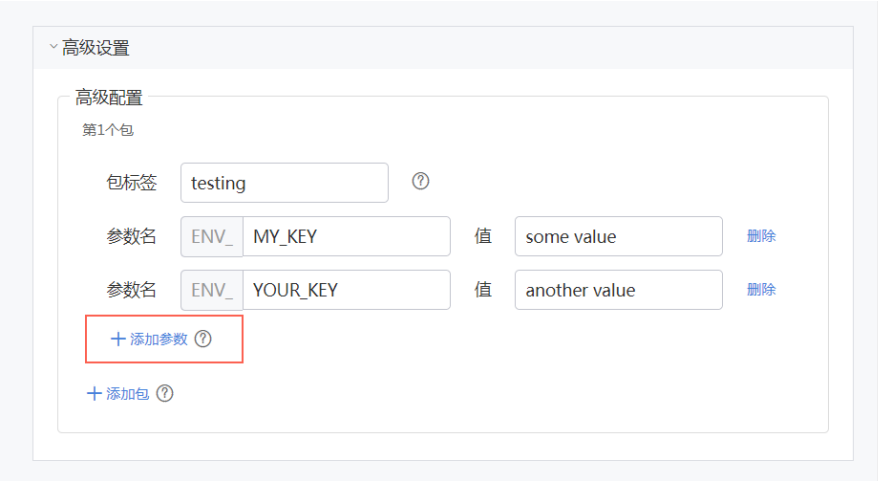
在流水线上配置构建任务时，点击打开高级配置，会看到“包标签”这个配置：

包标签的默认值是default，但可以调整这个配置。构建打包得到的包，用于不同用途时，可以在上面打上相应的标签。比如部署到日常测试环境的包和部署到预发环境的包、生产环境的包，需要有不同的内容，那么可以分别用testing、staging、production来区分这三类包。流水线上的部署组件，就可以根据需要，配置取得特定标签对应的包。比如，取得testing对应的包，用来部署日常测试环境。

那么，构建组件是如何根据包标签名的不同，打出不同的包呢？在构建时，系统会把包标签的值通过环境变量的方式，传到构建的上下文中。具体来说，该环境变量的key是PACKAGE_LABEL，值就是包标签的名字。于是，构建过程就可以据此进行调整，以产生适合这个构建目的的包。详见使用传入参数改变构建行为。

高级配置：增加输入参数

在包标签设置的下方，有“+添加参数”按钮，点击可添加key-value对：



这些key-value，将作为环境变量，传入构建过程。比如上图中，构建过程将获得ENV_MY_KEY这个环境变量，值为some value，以及ENV_YOUR_KEY这个环境变量，值为another value。构建过程就可以据此进行调整，以产生适合这个构建目的的包。详见使用传入参数改变构建行为。

高级配置：同时构建不同用途的多个包

可以在一个构建任务中，构建多个包。不同的包，至少包标签不同，还可以有不同的其他输入参数：

这些包，将在流水线上执行到该构建任务时，在多台机器上同时构建，以尽可能提高效率。其中任何一个包构建失败，将标记为该构建任务失败，流水线本次运行中止。

流水线上的自定义脚本任务

自定义脚本任务，是云效流水线上的一个任务，您可以使用该组件执行一些比较定制化的任务，比如使用自定义脚本发布静态资源 。

构建任务的配置

基本配置

在流水线编辑页面，添加任务时，请选择“自定义脚本” ，并填写其基本配置：

目前支持的脚本类型：shell。运行环境为云效提供的基础环境：registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0。

环境变量

云效内置了以下环境变量供您使用：

变量名	描述	示例	备注
PIPELINE_ID	流水线ID	18	
BUILD_NUMBER	流水线运行编号	22	
EMPLOYEE_ID	操作人	aliyun_1234	
CODE_INFO	流水线里包含的代码信息	[{ “appId” :45869,” appType” :” APP” ,” branch” :” git@code.aliyun.com:sample-group/sample-app.git master” ,” revision” :” 2f238c881fbda09c403a9183513b45bd8b481b71” }]	流水线里包含代码类型的参数才会有，否则此参数为 null
VERSION	流水线里代码构建的版	[{ “appId” :45869,”	在此任务前有构建任务

	本	appType”:”APP”,”branch”:”master”,”buildNumber”:7,”major”:0,”minor”:2,”revision”:35,”versionNumber”:”0.2.35.7”}]	，才会输出此参数，否则为 null
PACKAGES	流水线里代码构建打包结果	[{“appId”:45869,”fileMd5”:”ca0239f746476698d98fc1f5a4e55c3e”,”fileName”:”sample-app.tgz”,”fileUrl”:”https://rdc-build.aliyuncs.com/aone2/anonymous/build/package/download?path=0/15_0647187d-01e3-4a41-8252-72775dc96890-sample-app-2018-04-12-14-40-05-747-sample-app.tgz&md5Sign=91c20112201751f83470de3edf52249d","packageLabel":"default","packageName":"sample-app","packageType":"APP"}]	在此任务前有构建任务，才会输出此参数，否则为null。并且：fileUrl的下载有效时间为5分钟

高级配置：运行时变量

您可以在触发任务时，指定不同的环境变量。如下图所示。

运行时您会看到：

点击设置参数之后，进入填写参数页面：

填写完毕之后，点击确认，流水线会继续运行。在本例子中，会继续打印出hello world。

特性分支和分支集上的构建任务

概述

如果您是在使用云效专有云版，且您的企业配置了在特性分支/集上构建的功能，那么您可以在特性分支/集上直接进行构建打包。

在特性分支上构建

在特性分支的详情页，有构建卡片，可点击构建。

您也可以在“我的”-> “特性分支”列表中，看到构建卡片，并点击构建。

在一个分支集的详情页面中，有特性分支列表。列表中可以看到各特性分支的构建卡片，并点击构建。

选择包标签

在构建启动前，系统将让您选择包标签。

包标签的默认值是default，但可以调整这个配置。构建打包得到的包，用于不同用途时，可以在上面打上相应的标签。比如部署到日常测试环境的包和部署到预发环境的包、生产环境的包，需要有不同的内容，那么可以分别用testing、staging、production来区分这三类包。流水线上的部署组件，就可以根据需要，配置取得特定标签对应的包。比如，取得testing对应的包，用来部署日常测试环境。

那么，构建组件是如何根据包标签名的不同，打出不同的包呢？在构建时，系统会把包标签的值通过环境变量的方式，传到构建的上下文中。具体来说，该环境变量的key是PACKAGE_LABEL，值就是包标签的名字。于是，构建过程就可以据此进行调整，以产生适合这个构建目的的包。详见使用传入参数改变构建行为。

在特性分支集的多个特性分支上构建

分支集的详情页面中，可勾选打算构建的特性分支，然后点击“构建”按钮，一并构建。

系统除了让您选择包标签外，您还将有机会调整构建的顺序。

可配置键的完整列表

<应用名>.release文件存放在源代码所在Git库的根目录下。流水线的构建任务，根据这个文件构建打包，供后续的部署任务使用。

<应用名>.release是键-值形式的。例如：

```
code.language=oracle-jdk1.9
build.output=target/abc.war
```

这些键，可能带有前缀。比如docker.file带testing前缀，写为testing.docker.file。这些键的值，可能不是常数，而是带变量，比如docker.tag=${PACKAGE_LABEL}_${TIMESTAMP}。相关内容，详见使用传入参数改变构建行为。

下面给出这些可配置的键的完整列表：

键	默认值	可填写值	说明	是否必填
code.language	无	php5.6 php7.0 node6.x node7.x node8.x oracle-jdk1.7 oracle-jdk1.8 oracle-jdk1.9 scripts	用来确定构建使用的环境(详情)和默认构建命令(见说明1)	必填
build.command	见说明1	任意命令行	构建时执行的命令	选填
build.output	如果编程语言是	相对路径形式	需要最终打成	选填

	node，php，scripts，则默认值为./。其它情况下，需要显式填写。	，从代码库根目录算起。可以是文件（比如target/xxx.war）、目录下全部文件（比如target/*，此时解压后无该目录名）或目录（比如target）。	tgz压缩包的内容。	
build.output.nottgz	False	True False	不要对build.output指定的输出物打压缩包	选填
deploy.appctl.path	无	该文件的相对路径形式，从根目录算起，比如appctl.sh	需要添加到压缩包的部署脚本文件 详情	选填
docker.repo	无	比如registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app	推送到Docker Registry上的镜像名称	制作Docker镜像则必填
docker.repo.pull	无	内容格式与docker.repo相同	当设置该值时，云效依然会使用docker.repo中的地址进行构建和push，但在传递给部署系统（比如阿里云容器服务）时，会使用docker.repo.pull指定的url为基准的镜像地址。一个典型的使用场景是阿里云容器服务集群在VPC中，希望使用registry的vpc地址进行镜像下载，则可以指定docker.repo.pull为registry-vpc.cn-hangzhou.aliyuncs.com/mynamespace/container-app	选填
docker.file	Dockerfile	该文件的相对路径形式，从根目录算起，比如Dockerfile	制作Docker镜像所用Dockerfile的路径	选填

docker.tag	${PACKAGE_LABEL}_${TIMESTAMP}	比如${TIMESTAMP}	推送到Docker Registry上的镜像标签名称	选填

说明1：build.command的默认值：

- 编程语言是Java的Web应用：mvn -U clean package -Dappname=$APP_NAME -P$PACKAGE_LABEL（关于$APP_NAME和$PACKAGE_LABEL，请参看使用参数影响构建行为）。
- 编程语言是Java的安卓无线应用：./gradlew clean assembleDebug(assembleRelease) --info -s。
- 编程语言是Node时的Web应用：npm --python=/usr/alibaba/install/python-3.5.0/bin/python3 --registry=https://registry.npm.taobao.org install --production。其中的--python部分是为了进行包含本地扩展的Node模块的编译，详见：https://github.com/nodejs/node-gyp/https://github.com/nodejs/node-gyp。
- 其他情况，默认值为空，于是不进行构建。（可能进行生成Docker镜像、打压缩包等工作）

构建环境

本文描述云效使用的构建环境。下面会介绍可用的环境，如果您在构建中遇到了问题，可以查看构建环境调试。

基础环境

所有构建环境基于Ubuntu系统。

执行构建命令的用户是admin，拥有sudo权限。所以您可以使用sudo apt-get update && sudo apt-get install -y xxx 来安装需要的软件。

已经预装的软件：

1. g++ 4.9.2
2. gcc 4.9.2
3. make 4.0
4. curl
5. wget
6. unzip
7. python 3.5（不在PATH中，需要使用/usr/alibaba/install/python-3.5.0/bin/python3来引用）
8. git 1.9

各个语言的环境

通过设置release文件中的code.language的值，您可以使用相应语言、版本的构建环境。

Java环境

基于基础环境，并安装了：

1. maven 3.5
2. gradle 4.1

提供三个JDK版本：

1. oracle-jdk1.7（code.language=oracle-jdk1.7）
2. oracle-jdk1.8（code.language=oracle-jdk1.8）
3. oracle-jdk1.9（code.language=oracle-jdk1.9）

NodeJS环境

基于基础环境，并安装了：

1. yarn 0.27.5
2. 为了进行node-gyp的编译，该镜像中包含了两个python的版本：
    i. python 2.7，PATH中的python即为该版本。
    ii. python 3.5，继承基础环境中的python3.5，使用/usr/alibaba/install/python-3.5.0/bin/python3来引用该版本。

提供三个nodejs版本：

1. node6.11.3 npm3.10.10（code.language=node6.x）
2. node7.10.0 npm4.20（code.language=node7.x）
3. node8.4.0 npm5.3.0（code.language=node8.x）

注意：上述三个node及npm版本会随着相应的node大版本的更新而更新，但code.language的取值不变。比如您配置了code.language=node6.x，目前实际使用的是node6.11.3，如果node6的版本升级到了6.12.0，则您实际用到的可能就是6.12.0。

PHP环境

基于基础环境，并安装了：

1. composer 1.0

提供两个php版本：

1. php5.6 phpunit 5.7（code.language=php5.6）
2. php7.0 phpunit 6.3（code.language=php7.0）

Golang环境

基于基础环境，并安装了：

1. go-wrapper

提供两个golang版本：

1. go1.8.5（code.language=golang1.8）
2. go1.9.2（code.language=golang1.9）

Python环境

基于基础环境，提供两个python版本：

1. python2.7.13（code.language=python2.7）
2. python3.5.0（code.language=python3.5）

其它

如果您的构建对环境没有特殊需求，可以使用code.language=scripts。此时会使用基础环境。

构建环境调试

云效使用docker镜像的方式提供构建环境，镜像可以公开下载到，具体的镜像地址可以在构建日志中看到。所以如果您在构建中遇到了问题，可以按照构建日志的提示，将相应的镜像拉取到本地，假设镜像地址为registry.cn-beijing.aliyuncs.com/rdc-builds/php:7.0，则使用下面的命令进行本地调试：

```
//在宿主机上执行，并进入容器
$ docker exec -ti `docker run -d registry.cn-beijing.aliyuncs.com/rdc-builds/php:7.0` bash
//在容器中切换到admin，因为云效会使用admin账户进行构建
root@eed5d6e8a6bc:/#su admin
// 开始您的调试
。。。
```

如果我们的镜像不能满足您的需求，建议使用自定义构建镜像。

使用传入参数改变构建行为

尽管一个应用仅对应一个<应用名称>.release文件，但根据构建时上下文中传入的参数的不同，可以改变构建的行为，输出不同的构建结果。典型的，为不同的运行环境（比如测试环境和线上环境），打出不同内容，甚至不同类型的包。下面详细讲解基于同一份构建配置文件，根据上下文参数改变构建行为的方法。

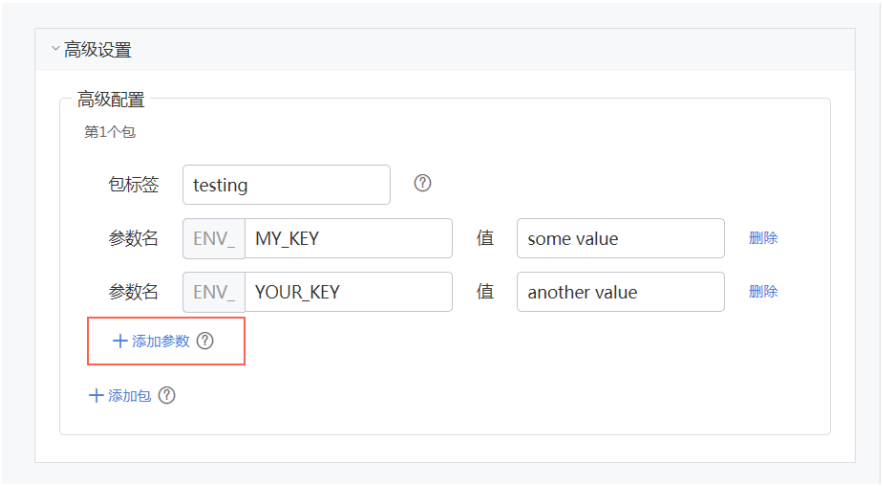
原理：包标签及其他环境变量

在流水线上配置构建任务时，点击打开高级配置，会看到“包标签”这个配置：

包标签的默认值是default，但可以调整这个配置。构建打包得到的包，用于不同用途时，可以在上面打上相应的标签。比如部署到日常测试环境的包和部署到预发环境的包、生产环境的包，分别用testing、staging、production来区分。流水线上的部署组件，就可以根据需要，配置取得特定标签对应的包。比如，取得testing对应的包，用来部署日常测试环境。

那么，构建组件是如何根据包标签名的不同，打出不同的包呢？在构建时，系统会把包标签的值通过环境变量的方式，传到构建的上下文中。具体来说，该环境变量的key是PACKAGE_LABEL，值就是包标签的名字。于是，构建过程就可以据此进行调整，以产生适合这个构建目的的包。调整方法详见下文描述。

除了包标签PACKAGE_LABEL这个环境变量，构建时系统还将把其他一些环境变量传入构建上下文。其中的一些，是系统自带的，详见**构建传入环境变量完整列表**。此外，用户还可以自定义若干key-value对：流水线上配置构建组件时的高级配置部分，在包标签设置的下方，有“+添加参数”按钮，点击可添加key-value对儿。



构建过程可以根据以上这些环境变量进行调整。详见下文。

方法1：以PACKAGE_LABEL的值作为配置键的前缀

<应用名称>.release中的键（比如docker.file），可以用PACKAGE_LABEL的值作为前缀（比如testing.docker.file）。构建时，如果找到以当时PACKAGE_LABEL的值（比如testing）作为前缀的键（比如

testing.docker.file），就将以它的值（比如Dockerfile），作为键（比如docker.file）的值。

当PACKAGE_LABEL为testing时候，云效会寻找所有不带前缀的键值，并与带testing前缀的键值，进行合并。带前缀的键值拥有更高的优先级。也就是说下面的例子中，docker.file最终的值是Dockerfile_test

```
docker.file=Dockerfile
testing.docker.file=Dockerfile_test
```

这个方法，不仅可以让特定键在为不同环境构建时，获得不同的值，甚至可以仅在特定环境获得值。比如，如果仅日常环境使用阿里云容器服务，需要进行镜像构建，但在生产环境希望使用ECS部署，不需要进行镜像构建。那么对于这个场景，可以使用下面的写法。

```
code.language=java
baseline.jdk=jdk-1.7.0_51
build.tools.maven=maven2.2.1

# 使用`PACKAGE_LABEL`的前缀（testing, staging, production）的键值，只会在相应环境的构建中生效
testing.docker.file=Dockerfile
testing.docker.repo=registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app
testing.docker.tag=${TIMESTAMP}
```

方法2：键值因为环境变量的值而改变

可以在键值中使用环境变量，例如：

```
docker.tag=${PACKAGE_LABEL}
```

还可以是组合，例如：

```
docker.tag=${PACKAGE_LABEL}_${TIMESTAMP}
```

再举个例子，使用PACKAGE_LABEL作为maven构建的profile。在release文件中配置build.command=mvn clean install -P$PACKAGE_LABEL。并且在您的pom文件中设置对应的profile，在其中定义不同环境的不同行为。

方法3：构建过程中引用环境变量

比如，设置build.command=sh build.sh，然后在build.sh中使用环境变量的值来进行判断，并执行不同的操作。

一个具体的例子：我想在不同包的构建中，使用不同的数据库配置。假设程序运行时从application.properties中读取数据库配置，则可以在代码库中放置三个配置文件：application.properties.testing，application.properties.staging，application.properties.production，在自定义流水线中也定义了的三个的PACKAGE_LABEL的值是testing，staging，production。然后在build.sh中包含这么一行，将属于相应环境的配置文件覆盖到程序读取的文件，也就是

application.properteis。

```
cp application.properties.$PACKAGE_LABEL application.properties
```

注意：上面只是一个例子，具体的配置文件的路径以你的项目的实际情况为准。

方法4：dockerfile中如何使用变量？

云效的应用中都有个<应用名称>.release文件,需要到release 文件中声明要传入的变量名称和值。

```
build.tools.docker.args=--build-arg APP_NAME=${APP_NAME} --build-arg ENV_TYPE=${ENV_TYPE} --build-arg
ENV_MY_KEY=${ENV_MY_KEY}
```

这里面的变量有两种，一种是系统内置的变量，如APP_NAME和ENV_TYPE，另一种就是用户在构建组件界面中配置的自定义变量，如MY_KEY,注意使用的时候需要加上ENV_前缀。Dockerfile 加入如下信息:

```
ARG APP_NAME
ARG ENV_TYPE
ARG ENV_MY_KEY
#将传入的应用名设置成环境变量
ENV APP_NAME ${APP_NAME}
ENV ENV_MY_KEY ${ENV_MY_KEY}
#类似用法可以发挥想象,注意shell 命令必须转换成行形式。可以续行
RUN if [[ "${ENV_TYPE}" = "testing" ]]; then echo ${APP_NAME}; fi
```

非自定义流水线时的情况

前面讲的在流水线上配置构建组件时配置PACKAGE_LABEL的值，指的是用户可自定义的流水线。目前有些情况下，系统仍然在使用不能这样自定义的流水线。典型的，当使用分支模式时，每个环境（典型的：日常测试环境、预发环境、正式环境），都对应一条不能用户灵活配置的流水线。此时，PACKAGE_LABEL的值被分别设置为testing，staging，production。（也就是旧版本云效中的ENV_TYPE的值）

此外，当不是自定义流水线时，用户也无法传入更多的自定义环境变量。

构建传入系统环境变量的完整列表

流水线上的构建任务，接受流水线框架传入的环境变量（包括系统自带的和用户在构建任务里自定义的），并可据此改变构建行为。详细介绍见使用传入参数改变构建行为。本文档列出其中所有的系统自带的环境变量。

环境变量名	说明
PACKAGE_LABEL	包标签，比如testing、staging、production或默认值default。 详情

APP_NAME	应用名。
CODE_BRANCH	代码库分支名。
TIMESTAMP	当前时间戳，比如20170622232633。

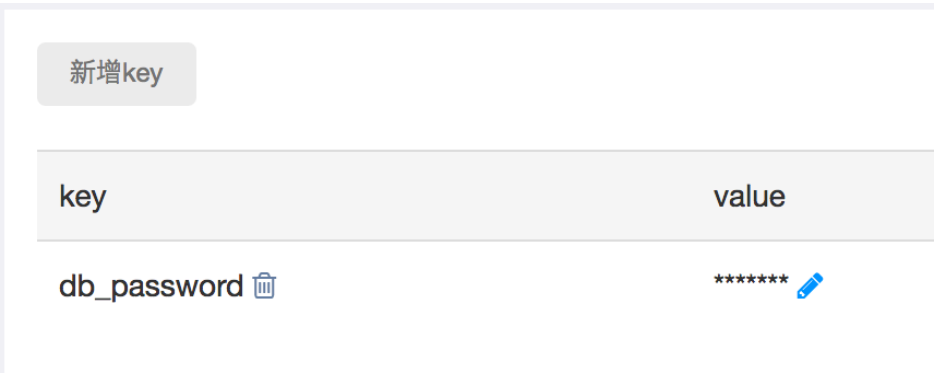
私密配置项

在应用构建中，通常会需要一些配置项，如：

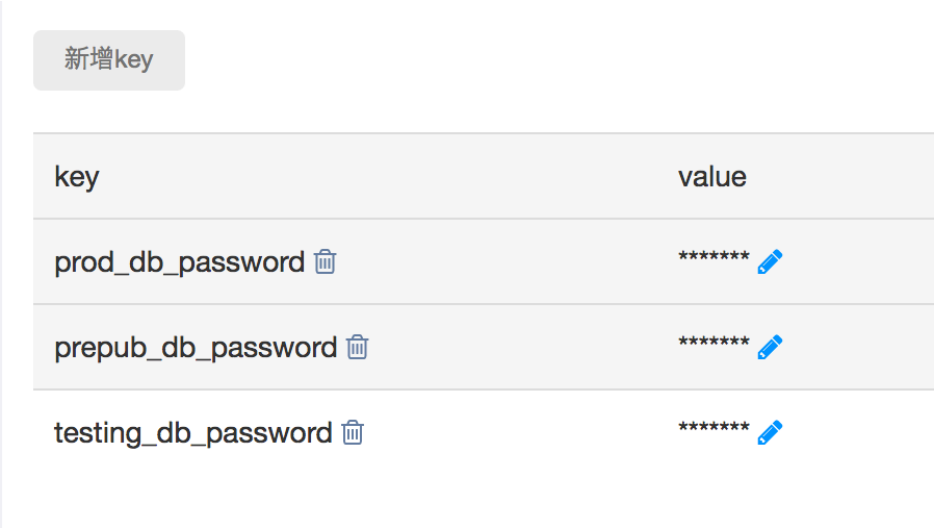
1. 功能开关
2. 依赖系统的URL
3. 数据库链接用户名密码

对于前两项，云效没有提供额外的支持，推荐您直接在代码中保存不同的配置文件，然后在构建时根据PACKAGE_LABEL的环境变量，选取正确的配置文件。详见**使用传入参数改变构建行为**。

第三项中的配置项会涉及一些私密信息，不适合放在代码库中。云效提供了私密配置项的保存功能。您可以在具体应用的私密配置项页面（从具体应用的顶部菜单栏中“私密配置项”菜单项进入）添加和配置**应用**级别的私密配置项，比如：



如果您需要在多个环境都使用私密配置项，则可以考虑把PACKAGE_LABEL的环境变量的值作为配置项的一部分：



配置好这些私密配置项之后，在进行构建时，云效会把这些配置项转换成为一个明文的文件，并将其放置在根目录下的rdc_security_config.properties中，比如：

rdc_security_config.properties:

```
prod_db_password=someprodpasswd
prepub_db_password=someprepasswd
tesing_db_password=sometestingpasswd
```

您可以在自己的构建脚本中读取该文件，并按照您自己的方式进行使用。其中，由于在构建上下文中可以获得环境变量PACKAGE_LABEL的值，因此可以据此知道相应私密配置项的名称，进而取得值。详见使用传入参数改变构建行为。

Web应用构建配置

概述

本文讲解Web应用构建相应的配置。关于构建的更多内容，比如<应用名>.release是什么，请从构建配置概述读起。

完成Web应用的构建配置后，请继续部署配置，参见部署配置：通过脚本部署。

Java构建

此时<应用名>.release文件基本内容：

```
code.language=oracle-jdk1.9
build.output=target/<应用名>.war
```

这意味着，将在Java构建环境（详见构建环境）中，使用Java默认构建命令mvn -U clean package -Dappname=$APP_NAME -P$PACKAGE_LABEL进行构建，随后把构建输出target/<应用名>.war打为tgz包并保存，供后续部署使用。

如果想改变构建命令，需要设置build.command。详见可配置键的完整列表中的build.command。

默认的maven settings会把所有的repository都镜像到maven.aliyun.com下载依赖，如果您需要不同的配置，只需要在代码根目录放置您的settings.xml，云效会使用该文作为构建的settings.xml。

如果需要使用私有maven仓库下载依赖或上传二方库，具体做法详见在云效中使用私有maven仓库。

Node构建

此时<应用名>.release文件基本内容：

```
code.language=node8.x
```

这意味着，将在Node构建环境（详见构建环境）中，使用Node默认构建命令npm --python=/usr/alibaba/install/python-3.5.0/bin/python3 --registry=https://registry.npm.taobao.org install --productionL进行构建，随后把构建输出./(源代码根目录)打为tgz包并保存，供后续部署使用。

如果想改变构建命令，需要设置build.command。详见可配置键的完整列表中的build.command。类似的，如果想改变打包范围，需要设置build.output。

Node构建通过engines的方式来获得特定的版本，具体方式是在package.json中添加如下片段：

```
...
"engines": {
"node": ">=5.1.0"
},
...
```

则云效会根据使用您指定的版本。该机制背后使用的是nvm，所以只要是nvm支持的版本，都可以填写。

PHP构建

此时<应用名>.release文件基本内容：

```
code.language=php7.0
```

系统将简单的把./(源代码根目录)打为tgz包并保存，供后续部署使用。如果希望构建，请设置

build.command，于是将在PHP构建环境（详见构建环境）中，据此构建后再打包。

其他情况

此时<应用名>.release文件基本内容：

```
code.language=scripts
```

系统将简单的把./(源代码根目录)打为tgz包并保存，供后续部署使用。如果希望构建，请设置build.command，于是将在基础环境（详见构建环境）中，据此构建后再打包。

补充说明

灵活配置构建环境

在build.command中，可以指定任意构建命令，比如build.command=sh build.sh，所以如果需要安装软件，或者执行复杂的命令，都可以通过这种方式实现。

环境变量对于构建过程的影响

关于环境变量对于构建过程的影响，请参看使用参数影响构建行为

不同环境使用不同的构建配置

云效支持为不同的运行环境打不同的包。为此，在不同的环境中使用不同的构建配置。详见使用参数影响构建行为。

修改一个环境的构建配置后，考虑相应的修改该环境的部署配置。详见应用部署概述。

关于包的管理

目前不提供压缩包的下载，该压缩包会在进行部署时候，直接传到指定机器上。详见部署配置：通过脚本部署。

使用EDAS部署时的构建配置

使用EDAS部署Web应用时，Web应用构建的配置有少许特殊性，详见下文。

关于Web应用构建的一般方法，请参考Web应用构建配置。关于构建的更多内容，请从构建配置概述读起。

完成构建配置后，请继续部署配置。关于Web应用通过EDAS部署，请参见部署配置：通过EDAS部署。关于部署的更多内容，请从部署配置概述读起。

配置云效不对构建物进行压缩

云效默认会将build.output所指示的war包或者jar包再打成tgz包，而EDAS接受的是war包或者jar包。所以需要在<应用名称>.release文件中进行如下配置，使得云效不再打包。配置示例如下：

```
...
# 打包的产物为target/xxx.war
build.output=target/xxx.war
# 不要再对 build.output 指定的输出物再进行打包
build.output.nottgz=True
...
```

一个完整的release文件的例子

假设应用名为edas-app。

edas-app.release:

```
code.language=oracle-jdk1.9
build.output=target/edas-app.war
build.output.nottgz=True
```

Docker镜像构建配置

在完成Web应用构建的基础上，继续构建生成Docker镜像，以便通过阿里云容器服务部署。本文讲解如何构建生成Docker镜像。

关于Web应用构建的一般方法，请参考Web应用构建配置。关于构建的更多内容，请从构建配置概述读起。

完成构建配置后，请继续部署配置。关于通过容器服务部署，请参见部署配置：通过容器服务部署。关于部署的更多内容，请从部署配置概述读起。

企业全局配置

镜像构建的用户名密码：一个企业内部可以共享一个docker login的用户名密码。可以在企业设置-> 容器服务

账号中添加。

一个构建配置示例

下面给出一个容器构建配置的完整示例（假设应用名为container-app）。

代码库目录结构：

```
$ tree .
.
├── Dockerfile
├── pom.xml
├── src
└── container-app.release
```

构建配置文件container-app.release：

```
code.language=oracle-jdk1.9
build.output=target/container-app.war

# docker构建所用的Dockerfile的路径
docker.file=Dockerfile

# docker构建完成之后，要push到的docker repo
docker.repo=registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app

# 使用时间戳做docker tag，这样打出来的docker镜像就形如：registry.cn-
hangzhou.aliyuncs.com/mynamespace/container-app:20170622232633
docker.tag=${TIMESTAMP}
```

Dockerfile：

```
# 为自己的应用程序打一个基础镜像，把基础的软件安装好，并且包括启动的entrypoint
FROM registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app:base

# 上面提到了，云效会把container-app.tgz放到Dockerfile的同级目录，所以可以这么写，把云效打出来的软件包拷贝到镜像
中
COPY container-app.tgz /home/admin/container-app.tgz
```

pom.xml，src 略。

按照上述的配置，云效会：

1. 先按照默认的Java语言的构建方式打出war包在target/container-app.war。
2. 把target/container-app.war打成container-app.tgz，并放置在代码库根目录。
3. 运行docker login命令：docker login -u "xxx" -p "xxx" registry.cn-hangzhou.aliyuncs.com。
4. 运行docker构建命令：docker build --pull -f Dockerfile -t registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app:20170622232633

/home/admin/xxxxx/container-app/
5. 再次运行第3步中的命令。
6. 运行docker push命令：docker push registry.cn-hangzhou.aliyuncs.com/mynamespace/container-app:20170622232633

构建配置详解

容器构建配置项

1. docker.repo：必填，镜像仓库的地址。
2. docker.file：选填，dockerfile相对代码根目录的相对路径。
3. docker.tag：选填，镜像tag的规则。

更多见可配置键的完整列表。

可用的环境变量

系统提供的所有环境变量见这里。下面列举几个常用的环境变量。

1. PACKAGE_LABEL：标识当前构建的参数。对于自定义流水线来说，如果您没有进行特殊配置，则PACKAGE_LABEL的值为default。您可以在自定义流水线的配置页面，对不同的包配置不同的PACKAGE_LABEL的值。如果您使用的是非自定义流水线，则PACKAGE_LABEL的值与ENV_TYPE的值相同。其值可以为testing, staging, production。
2. CODE_BRANCH：标识当前构建的分支名称。如果使用分支模式，则每次构建的分支是一个集成分支，形如：releases/20170623154859032_r_release_35191_app-code。这时最好不要使用CODE_BRANCH。
3. TIMESTAMP：当前时间戳，形如：20170622232633。

配置项默认值规则

1. docker.repo：无默认值。
2. docker.file：默认值为Dockerfile。
3. docker.tag：默认值为${PACKAGE_LABEL}_${TIMESTAMP}。

镜像构建的Context

关于镜像构建的Context的基础知识见链接。

云效会使用Dockerfile所在的路径进行镜像构建，也就是说镜像构建的Context就是Dockerfile所在的目录。

举个例子，如果Dockerfile的路径是docker/files/Dockerfile_testing，那么云效会把docker/files作为镜像构建的Context。

云效还会把打包产物自动拷贝到镜像构建的Context中。在上面的例子中，会把打出来的tgz包拷贝到

docker/files目录下。tgz包的默认名称为<应用名>.tgz。这就意味着你可以在Dockerfile中直接这么写：

```
COPY `<应用名>.tgz` /home/admin/app-path/
```

典型场景

可以通过上述环境变量的不同组合来满足不同的使用场景：

不同环境使用不同的Dockerfile

release文件：

```
...
docker.file=Dockerfile_${PACKAGE_LABEL}
...
```

这样，在日常环境就会使用Dockerfile_testing，预发环境使用Dockerfile_staging，正式环境使用Dockerfile_production

使用时间戳作为镜像tag

release文件：

```
...
docker.tag=${TIMESTAMP}
...
```

这样，生成的tag就是形如20170622232633这样的字符串。

在tag中区分环境

使用同一个Dockerfile的用户，如果希望从镜像的tag中，看出来镜像是哪个环境的，则可以这样配置release文件：

```
...
# 这也是tag的默认值
docker.tag=${PACKAGE_LABEL}_${TIMESTAMP}
# 或者
docker.tag=${TIMESTAMP}_${PACKAGE_LABEL}
...
```

在tag中使用分支

有些团队会使用分支区分开发状态，比如develop分支上打出来的包都是测试包，master上打出来的是正式包。这时候可以在tag中包含CODE_BRANCH这个环境变量。比如docker.tag=${TIMESTAMP}_${PACKAGE_LABEL}_${CODE_BRANCH}。

无线应用构建配置

无线应用的构建打包，云效目前支持Android（iOS将在后续支持），下面会讲解构建环境和构建行为，及如何进行定制化配置。

关于构建的更多内容，请从构建配置概述读起。

基础环境

操作系统

基于centos 5

自定义软件

您可以在构建环境中运行任意命令安装您需要的软件。但目前不支持安装rpm包。

自定义构建脚本

您需要在代码库中放置一个文件：<应用名>.release（如果在创建应用时，选择新建代码库，则云效会帮您生成这个文件，并提交到代码库中）。该文件以键值对的形式描述构建行为。

您可以在release文件中通过build.command指定任意构建命令，比如build.command=sh build.sh，所以如果需要安装软件，或者执行复杂的命令，都可以通过这种方式实现。

不同无线客户端构建行为

Android构建

使用Android构建，release文件需要按如下形式编写：

```
# 必填，表示是Java构建
code.language=java

# 选填，取值可以是jdk-1.6，jdk-1.7，jdk-1.8，默认值为jdk-1.7
baseline.jdk=jdk-1.7.0_51
```

现在云效支持 gradle-2.1.0，gradle-4.0 两个版本，你可以在代码中build.gradle文件中指定需要的版本。

Android的默认构建命令为：./gradlew clean assembleDebug(assembleRelease) --info -s，您可以通过build.command进行覆盖。

iOS构建

iOS构建现在暂时还不支持。

自定义构建镜像

当构建环境中预置的的编译环境不能满足您的要求时。您可以使用自定义构建镜像的功能来定制所需的构建环境。

制作构建镜像

您需要按照如下的约束来编写构建环境使用的Dockerfile

- 使用指定的基础镜像：registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0
- 根据需要，安装软件和设置环境变量（admin为构建使用账号，不要删除或修改UID）。
- 整个镜像大小需控制在1G之内。

一个完整的Dockerfile示例如下：

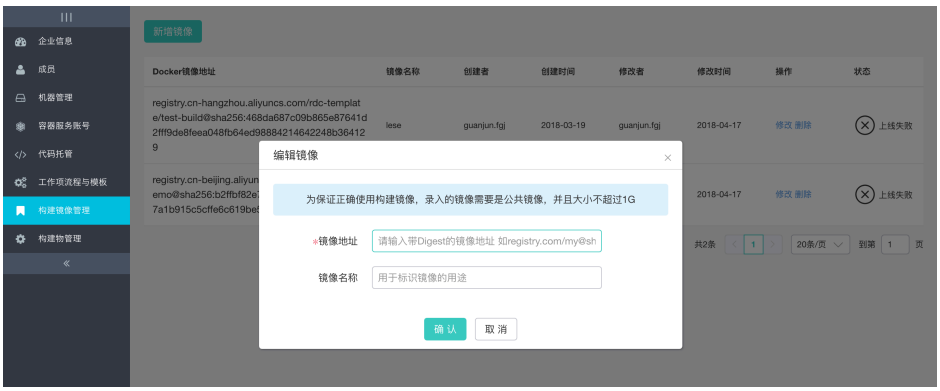
```
FROM registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0

RUN cd /tmp && \
wget http://rdc-public-software.oss-cn-hangzhou.aliyuncs.com/jdk-7u80-linux-x64.tar.gz && \
tar xf jdk-7u80-linux-x64.tar.gz -C /srv/java && \
ln -s /srv/java/jdk* /srv/java/jdk
ENV JAVA_HOME=/srv/java/jdk \
PATH=${PATH}:/srv/java/jdk/bin:/srv/java
```

在本地调试通过后，将镜像上传到阿里云或其他公网可访问的registry，且为公开权限。

录入镜像信息

镜像上传成功后，在“企业管理”->“构建镜像管理”页面，点击【新增镜像】按钮进行录入。



注意，为了保证镜像地址的准确性，云效要求精确到digest级别，所以要求镜像地址格式为：REPOSITORY@DIGEST，例如：registry.cn-hangzhou.aliyuncs.com/rdc-template/test-build@sha256:468da687c09b865e87641d2fff9de8feea048fb64ed98884214642248b364129。

如果您更新了镜像，需要到云效更新镜像digest，才能保证构建使用的是更新后的镜像。

使用构建镜像

在代码库的根目录下的<appName>.release文件，添加镜像配置：build.image=<your image repo url>。<your image repo url>需要与在镜像录入页面填写的镜像地址保持一致。修改后触发构建，新的构建会使用配置的镜像作为构建环境。

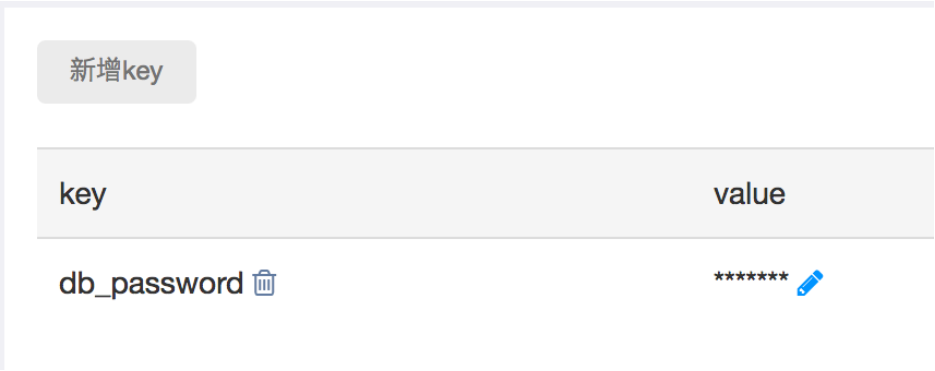
私密配置项

在应用构建中，通常会需要一些配置项，如：

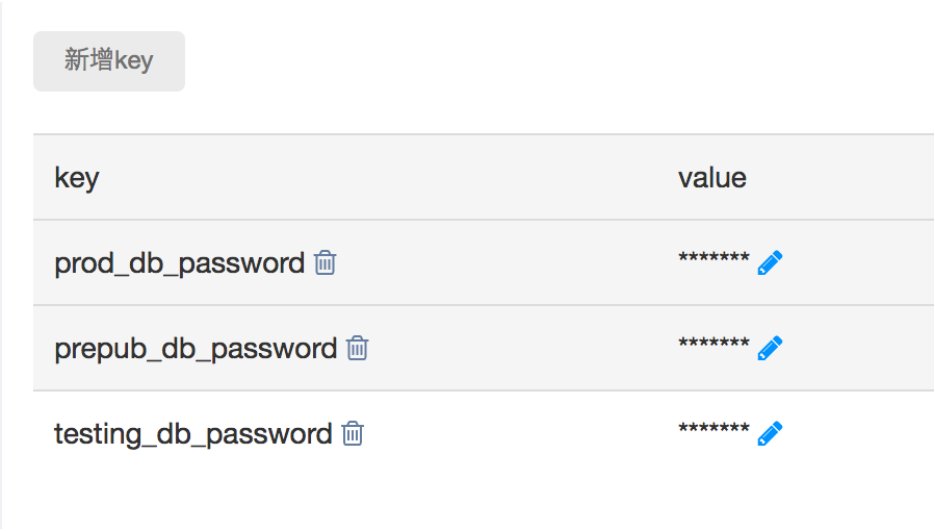
1. 功能开关
2. 依赖系统的URL
3. 数据库链接用户名密码

对于前两项，云效没有提供额外的支持，推荐您直接在代码中保存不同的配置文件，然后在构建时根据ENV_TYPE的环境变量，选取正确的配置文件。

第三项中的配置项会涉及一些私密信息，不适合放在代码库中。云效提供了私密配置项的保存功能。您可以在应用的私密配置项页面（https://rdc.aliyun.com/ec/app/xxx/securityConfig）添加和配置**应用**级别的私密配置项。比如：



如果您需要在多个环境都使用私密配置项，则可以使用如下的方式：



配置好这些私密配置项之后，在进行构建时，云效会把这些配置项转换成为一个明文的文件，并将其放置在根目录下的rdc_security_config.properties中，比如：

rdc_security_config.properties:

```
db_password=somepasswd
```

您可以在自己的构建脚本中读取该文件，并按照您自己的方式进行使用。

部署

应用部署概述

每个Web应用，在集成测试的环境（在云效中通常称作日常环境）、预发的环境（称作预发环境）、对外提供服务的环境（称作正式环境）等不同的环境里运行。每个环境中，该应用运行在若干台机器（虚机/容器）上。部署时，可能分期分批。每台机器的部署，有特定的方法和步骤。这些都是定义在这个应用的特定环境上。

云效支持您创建自定义环境，详见环境与环境级别。

这意味着，可以为同一应用的不同环境，配置不同的部署参数，甚至不同的部署方法。比如日常环境通过容器服务部署，而正式环境通过脚本部署。

下面详述当前云效支持的以下几种部署方式的配置：

- 部署配置：通过脚本部署
- 部署配置：通过EDAS部署
- 部署配置：部署容器服务的Swarm集群
- 部署配置：部署容器服务的Kubernetes集群
- 不需要传包，直接通过git pull的方式进行部署

环境与环境级别

本文介绍云效中的环境、环境级别的概念，及如何同时部署一个环境级别中的多个环境。目前这一部分内容，是针对分支模式这种开发模式的。

若要了解更加通用的，如何在环境上配置资源进行发布，请查看应用部署概述。

基本概念

首先要理解云效中环境级别的概念。

云效中预置了日常环境，预发环境和正式环境三个**环境级别**。每个环境级别可以包含一个或者多个具体的环境。云效在新建的应用中为您预置了日常环境、预发环境、正式环境三个**环境**，分别对应上述的三个环境级别。

使用环境级别的场景

一个环境级别代表了一组环境，比如您的应用使用阿里云容器服务进行部署，希望把同一个镜像部署到杭州和北京两个容器集群。则可以创建两个，环境级别为**正式**的环境：杭州正式，北京正式，然后在两个环境中分别关联相应的容器集群中的相应服务。关于容器部署的更多内容，请参看部署配置：通过容器服务部署。

在分支模式下，预置的三个流水线，日常部署，预发部署，正式部署，会部署相应的环境级别中的所有环境。比如对于上述的杭州正式，北京正式两个环境，在正式部署的流程中进行部署时，会并行部署这两个环境。

新建环境

您可以在应用的概述->环境配置中添加新的环境，如图：

添加完成之后，可以看到：

此时，您的正式环境这个环境级别中包含两个**环境**：正式环境，正式环境1。

添加完成之后，您可以继续在最上层的环境tab中，编辑相应环境的部署配置。

同时部署多个环境

新环境创建完成之后，您就可以在原有的正式发布流程中，同时发布正式环境，正式环境1这两个环境了，而流程本身没有任何变化。如下图所示：

点击查看正式部署日志，可以看到多个环境的各自的部署信息。

流水线上的部署任务

部署任务，是云效流水线上的一类任务，它负责把构建得到的包，部署到运行环境并启动。

部署任务的运行

部署任务一般不需要在运行时输入信息，就会自动运行。运行期间和运行结束后，可以在页面下方点击“查看发布单”，查看更多细节：

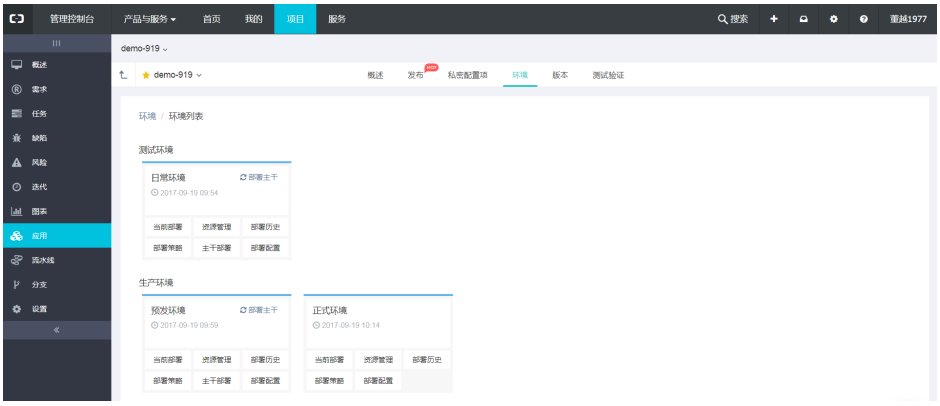
部署任务的配置

在流水线编辑页面，添加任务时，请选择“部署”，并填写其基本配置：

配置的核心思路是，选择合适的包，部署到合适的地方。其中，选择合适的包，是“应用”和“包标签”这两项决定的。部署到合适的地方，是“应用”和“环境”这两项决定的。

“包标签”是构建时用来区分同一个应用的不同用途（比如为不同运行环境）的包，而打上的标签。详见**流水线上的构建任务**中，对包标签的介绍。这里是选择本次部署所需要的包对应的标签。

“应用”和“环境”的概念见**这里**。环境的配置，包括环境关联到哪些机器、部署时使用的脚本等等。详见部属配置系列文档，比如**部署配置：通过脚本部署**。环境的配置是从应用入口进入的，进入后选择“环境”菜单项：



部署配置：通过脚本部署

本文讲解如何配置通过自定义脚本把Web应用部署到指定服务器。

关于如何配置构建产生部署用的包，请参见Web应用构建配置。

关于如果进行企业机器资源的管理，如果已有主机，请参见把已有机器关联到云效本企业，如果暂时没有主机，请参见配置授权云效、配置ECS模板、购买ECS机器。

应用环境关联到机器

在“应用”-“环境”页面，点击“资源管理”，可以增加关联的机器。

提示

- 应用可以关联的机器，来自于企业管理员在企业设置中导入的机器；
- 如选择不到机器，请联系管理员导入机器；

应用部署信息配置

新建好应用之后，在环境页面，您可以看到，云效会为您预置日常、预发、正式三个环境。并且可以对每个环

境做部署的配置：

注意：当前环境正在部署时，部署配置无法修改。（若在部署过程中修改部署配置，会生成中间版本，当部署完成写入正式基线后，该中间版本会被清除掉）

不同的应用可以有自己的定制化的部署脚本，这里给出了一个基于SLB的滚动发布脚本示例，供您参考。

此外，可以把部署脚本放在代码库中，当它的内容更新时，将在部署时自动同步到各机器。详述见在代码库中存储部署脚本。

部署配置：通过EDAS部署

一、概述

EDAS是阿里云上的一个服务，提供了中间件，部署，及运维等能力，详情见EDAS文档。云效对EDAS进行了集成，可以把在云效上打出来的war包或者jar包部署到EDAS中。

为了在云效上集成EDAS，需要保证您的应用可以在云效上打出war包或者jar包。详见使用EDAS部署时的构建配置。

云效支持多种研发模式，及部署回滚等功能。EDAS提供了中间件、部署、运维、日志、监控等服务。云效与EDAS结合，可以很好的提供一站式持续交付体验。

EDAS提供了多种部署能力，云效目前只支持基于war包和jar包的部署，不支持EDAS容器部署。

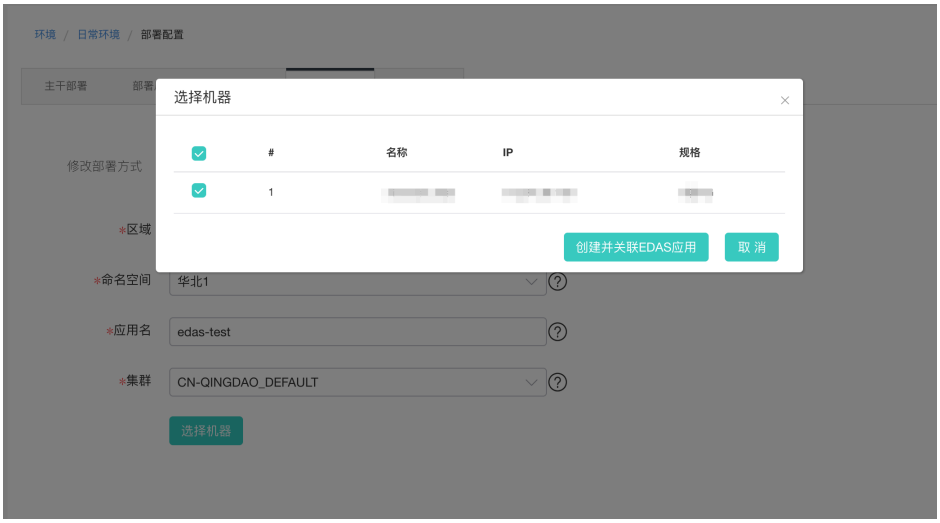
你可以在EDAS上创建应用，也可以在云效上创建EDAS应用，然后使用云效进行集成发布。

二、云效上创建EDAS应用

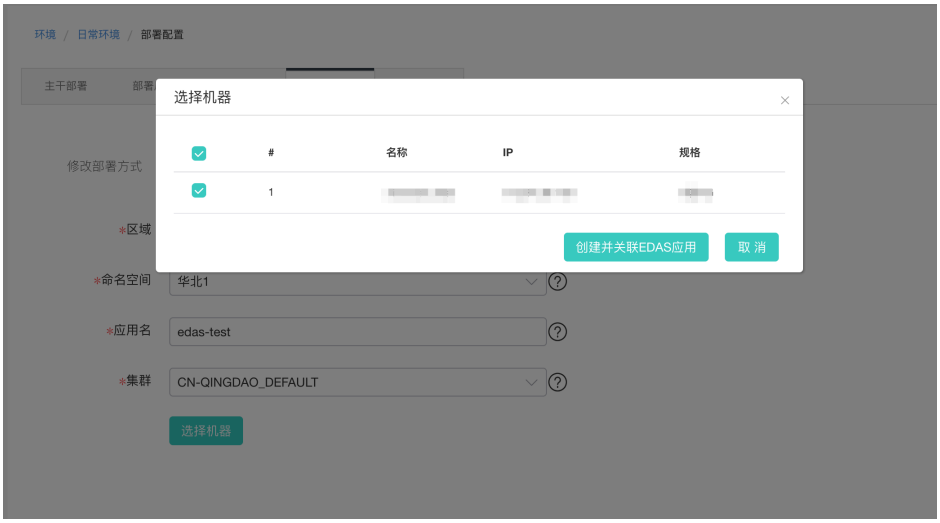
目前在云效上只支持创建普通EDAS应用和DOCKER EDAS应用，暂不支持创建Kubernetes应用。

在应用 -> 环境 -> 部署配置 的页面 可以创建EDAS应用。创建应用前需要在EDAS控制台相应的区域下，添加相应集群，并关联好机器。

2.1 创建普通 EDAS应用



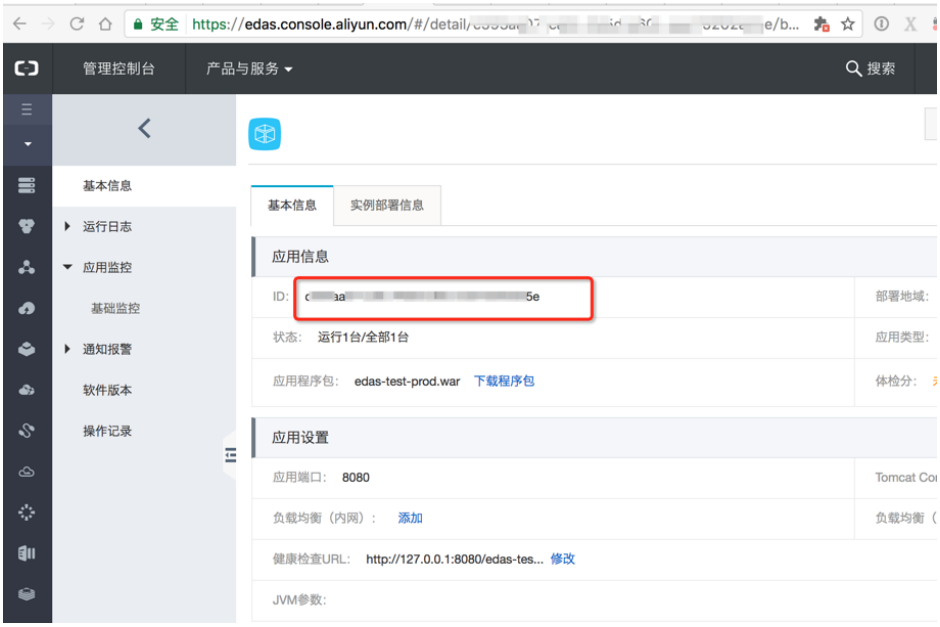
2.2 创建Docker EDAS应用



三、云效上关联EDAS应用

在应用 -> 环境 -> 部署配置 的页面配置对应的EDAS的应用ID，如图所示：

EDAS应用ID可以从EDAS应用详情页面获取：
https://edas.console.aliyun.com/#/detail/xxxxxxx/basicInfo.info。 如图：

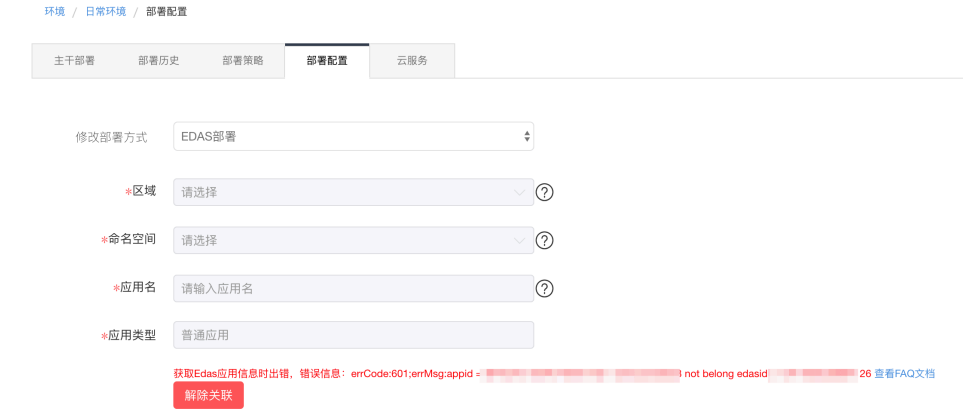


四、EDAS操作常见问题

4.1 查看EDAS应用信息出错

4.1.1 无权限查看应用信息

在部署配置页面，查看EDAS应用信息时，出现以下错误，是由于当前用户无相应EDAS应用权限，如遇这种情况，请联系相应的EDAS应用管理员，为您的阿里云账号添加权限。关于如何添加权限，请参考EDAS账号体系



。

4.1.2 查看的EDAS应用不存在

在部署配置页面，查看EDAS应用信息时，出现以下错误，是由于当前环境关联的EDAS应用不存在，请到EDAS控制台确认当前应用是否存在，如果应用已删除，可解除该EDAS应用与云效的关联关系。

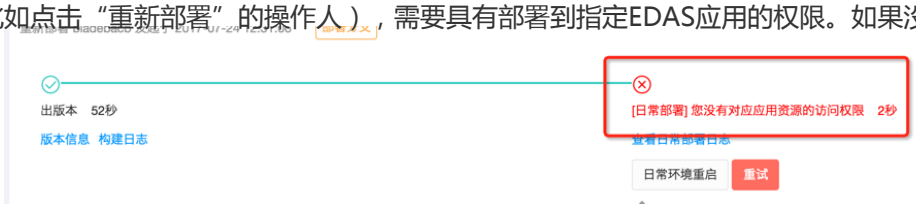
4.2 部署应用时出错

4.2.1 未配置云效不对构建物进行压缩

使用云效进行EDAS部署时，出现以下错误，是由于未配置云效不对构建物进行压缩，可以按照下面方式解决。详见使用EDAS部署时的构建配置

4.2.2 无权限部署EDAS应用

当前操作人（比如点击“重新部署”的操作人），需要具有部署到指定EDAS应用的权限。如果没有权限，则会报如下的错误：



如遇这种情况，请联系相应的EDAS应用管理员，为您的阿里云账号添加权限。关于如何添加权限，请参考EDAS账号体系。

4.2.3 EDAS应用不存在

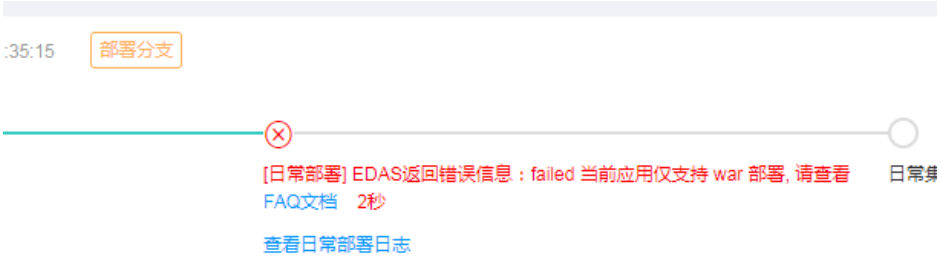
使用云效进行EDAS部署时，出现以下错误，是由于当前环境关联的EDAS应用已删除，请到EDAS控制台确认当前应用是否存在，如不存在，可以解除当前环境与该EDAS应用的关联，并重新关联可用EDAS应用。

4.2.4 EDAS应用无可部署机器

使用云效进行EDAS部署时，出现以下错误，是由于当前环境关联的EDAS应用无可部署机器，请到EDAS控制台关联机器后，重新进行部署。

4.2.5 部署包格式不正确

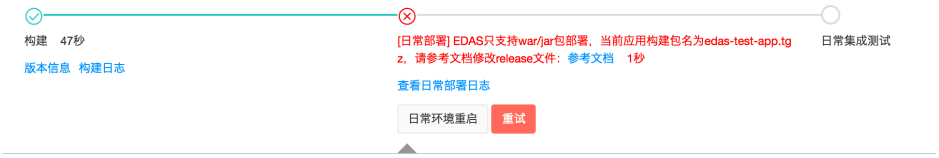
EDAS的ECS部署支持jar包和war包两种格式。如果你看到了下面的错误：



那么有两种可能：

1. 该EDAS应用所选择的容器不支持jar包的部署方式。需要您在创建EDAS应用时选择支持fatjar部署的容器版本。
2. 该EDAS应用之前使用过war部署，而本次尝试部署尝试使用jar包进行部署。

如果遇到下面的错误：



说明您的构建配置不正确，请参看构建配置进行修复。

4.2.6 EDAS应用已有部署单执行

使用云效进行EDAS部署时，出现以下错误是由于当前环境关联的EDAS应用已经存在部署单，可到EDAS控制台查看该应用部署信息，待上一个部署单部署结束后，在云效相应页面点击重试即可。

部署配置：部署容器服务的Swarm集群

本文讲解如何配置通过容器服务把Docker镜像形式的Web应用部署到指定服务器。

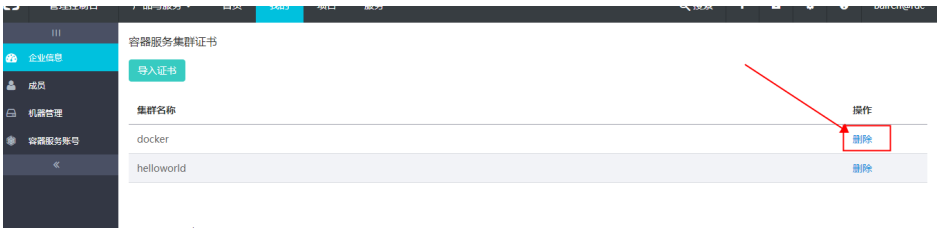
关于如何配置构建产生部署用的包，请参见Docker镜像构建配置。

你需要在阿里云容器服务中已经有集群，应用和服务。

导入集群证书：在企业设置-容器服务账号中，点击导入证书。

这里的集群就是容器服务中当前用户下的所有集群。

对于不需要展示的集群也可以在rdc中删除，再次点导入就会重新全部导入。



在环境页面中，点击部署配置，选择该环境要发到容器中对应的集群、应用、服务。

要使用蓝绿部署，请确保首先阅读并理解阿里云容器服务的相关文档。

选择了蓝绿部署。在进行部署的过程中，云效会新创建一个服务，服务名为原有服务名加上一个_rdc_blue_green<时间戳>的后缀，比如现在服务名称是web，则第一次蓝绿部署就会新增一个web_rdc_blue_green20171214103512的服务，供蓝绿切换之用。后续每次蓝绿部署都会使用相同规则的后缀来生成新的服务名。

如果从蓝绿部署切换回到标准发布，则会保留最后一次的服务名（比如web_rdc_blue_green20171214103512）。

如果再次从标准发布切换到蓝绿发布，则云效会识别出已有的后缀，并继续使用蓝绿的规则对服务名进行修改。

举个实际的例子，一开始服务名为web。

1. 切换至蓝绿部署，进行一次部署，生成新的服务名web_rdc_blue_green20171214103512，您可以在容器服务的控制台，在web和web_rdc_blue_green20171214103512这两个服务之间进行切流，确认发布完成之后，容器服务会销毁web，保留web_rdc_blue_green20171214103512。切流和确认操作，请参看**本篇容器服务文档** ，在其中搜索单击 确定，发布变更小节的内容。
2. 一分钟之后，再进行一次部署，生成新的服务名web_rdc_blue_green20171214103612，您可以在容器服务的控制台，在web_rdc_blue_green20171214103612和web_rdc_blue_green20171214103512这两个服务之间进行切流，确认发布完成之后，容器服务会销毁web_rdc_blue_green20171214103512，保留web_rdc_blue_green20171214103612。
3. 一分钟之后，切换至标准发布，进行一次部署，云效会保持使用最后一次生成的服务名，也就是web_rdc_blue_green20171214103612，进行部署。
4. 一分钟之后，再次切换回到蓝绿发布，进行一次部署，云效会生成一个新的服务名web_rdc_blue_green20171214103812，您可以在容器服务的控制台，在web_rdc_blue_green20171214103612和web_rdc_blue_green20171214103812这两个服务之间进行切流，确认发布完成之后，容器服务会销毁web_rdc_blue_green20171214103612，保留web_rdc_blue_green20171214103812。

部署中，可以点击查看部署日志查看部署进度。

如果您当前登录云效的阿里云账号，没有该容器服务应用的权限，则无法看到该日志。

部署成功或者失败后，流程就会结束。

部署配置：部署到容器服务的Kubernetes集群

概述

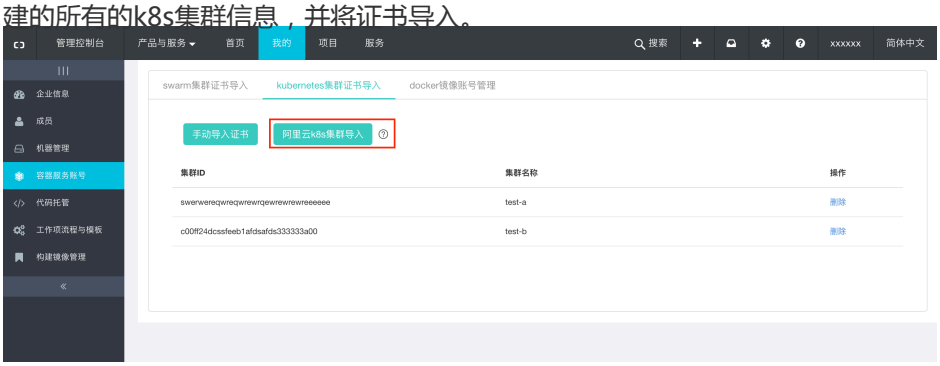
Kubernetes 是当前主流的开源容器编排技术，为容器应用的管理带来很大的便利。但是，将镜像部署到集群的过程需要开发者者有k8s知识基础，而且如果通过手工操作容易出错，影响发布效率。云效提供构建到部署全链路自动化的流程，支持部署到阿里云容器服务的创建的k8s集群。让用户专注于业务开发，降低发布成本。

集群导入

集群属于企业信息，需要企业管理员操作，通过【企业设置】进入【容器服务账号】页面。选择【kubernetes】tab页面后,可以看到导入的操作入口。

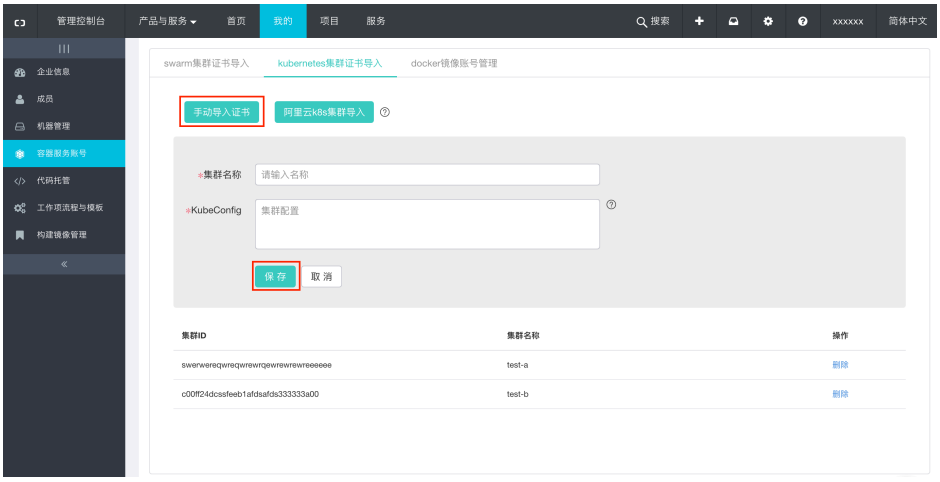
容器服务集群导入

点击【阿里云K8S集群导入】按钮， 系统会根据当前操作人（***需要是阿里云主账号***）的账号，查询容器服务创建的所有的k8s集群信息，并将证书导入。



其他托管或自建集群的导入

点击手动导入证书，将显示集群导入的输入操作框，填写信息后，点击【保存】按钮。系统会校验集群的证书，成功后保存集群信息，并刷新页面。

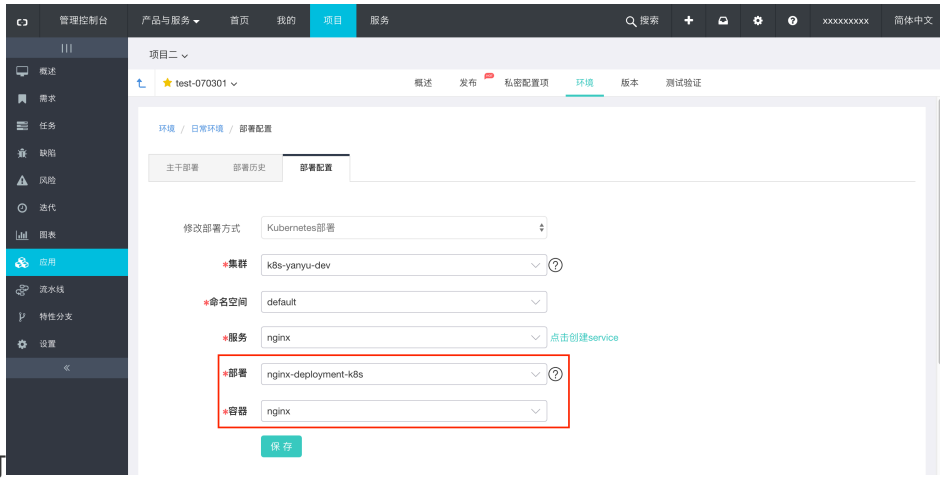


对于不需要展示的集群，可以进行删除操作。

环境配置

到【环境】页面，进入某个环境的【部署配置】。选择发布的集群、Namespace和service，点击【保存】。

如果当前service对应的deployment不止一个，页面会展现deployment和container的选项，选择后点击【保



存】即可

如果你需要新建service，点击service选项旁边的【新建service】链接，填写信息后，点击【保存】，将创建service并保存部署配置

部署操作

通过流水线进行应用的部署，在部署过程中，可以点击【查看部署日志】查看部署进度

点击后，跳转到容器服务，查看对应的deployment（如果报没有权限，请联系创建集群的主账号给您添加访问权限）

使用“git pull”的方式更新应用

如果您的应用不需要打包，在生产服务器上直接通过git pull的方式进行更新，那么可以按照如下的方式进行操作。

release文件

在您的代码库根目录中添加<应用名>.release文件（如果不存在的话）。内容如下：

```
code.language=scripts

# 将当前的git版本号写入元信息文件
build.command=git rev-parse HEAD > rdc_build_meta

# 告诉云效把元信息文件打包成package.tgz
build.output=rdc_build_meta
```

部署配置

按照如下方式进行部署配置（您可以在应用->环境->部署配置中找到如下的配置表单）。

下载路径：/home/admin/package.tgz (需要您保证/home/admin目录存在，或者替换成实际存在的某个目录)

解压目录：/home/admin/package-explode （可以按照您的需求，替换成别的目录）

Stop：echo noops (如果不需要stop，随便填即可；如果需要，按实际情况填写。)

Start：cd /home/admin/app && git fetch && git checkout `cat /home/admin/package-explode/rdc_build_meta`（这条命令把构建时打包的rdc_build_meta文件解压出来，然后checkout到文件中指定的版本）

执行用户：admin （这个例子中使用的是admin用户进行部署，您可以替换成实际的用户）

添加Agent失败FAQ

我在机器上安装了Agent，但在企业的机器列表中看不到

请按照下列步骤依次排查。执行完每一步之后，请确认问题是否解决，若未解决，请继续尝试后续步骤。

目前agent只支持64位的Linux操作系统。

确认您是否曾经使用另一个企业的agent安装命令在该机器上执行过，如果是，请删除/usr/sbin/staragent_sn（正常情况下该文件内容为机器SN，SN为机器唯一标识，并与特定企业绑定），并重装agent。

执行命令cat /usr/sbin/staragent_sn查看内容。若文件内容为空，删除此文件，并重装agent（PS：请勿手动修改该文件）。

执行命令/home/staragent/bin/staragentctl status查看agent状态。若输出异常（比如ServerAddr为空，或者报错），请执行命令cat /home/staragent/conf/staragent.conf查看文件内

容，如果文件存在，且其中的URL为rdc-xxx.aliyuncs.com或者staragent-configservice.aliyuncs.com，则为正常。如果不是，有可能是您的机器之前安装过其它产品的agent，请重装云效agent。

查看cat /home/staragent/conf/channels.conf是否存在，如果不存在，请执行命令：curl 'http://<从staragent.conf中获取的服务URL>/api/configservice?action=findChannelListForAgent&agentIpList=101.37.119.155%2C10.80.237.52&needAllChannels=true&serviceTag=ea263ff8-2d60-48f0-86c4-33a04214cad9&version=2'，如果结果类似{"appCode":"_successful_","msg":"","restCode":200,"result":{"allChannels":[],"channelIPPort":[{"ip":"100.100.18.88","port":8000},{"ip":"100.100.18.89","port":8000},{"ip":"100.100.45.99","port":8000},{"ip":"182.92.29.36","port":8000},{"ip":"182.92.29.39","port":8000},{"ip":"100.100.45.100","port":8000}]，请尝试重启agent（参看下面的agent基础操作）。如重启后/home/staragent/conf/channels.conf仍不存在，请点击云效页面右下角“提问”联系我们。如果不能返回类似结果，则表示您的机器到<从staragent.conf中获取的服务URL>的连接有问题。有可能是在安装agent时候，选错了区域。请在添加机器页面选择正确的区域，生成agent安装命令，重装agent。

执行命令cat /home/staragent/conf/channels.conf查看该文件，内容会是一个ip+port的列表，尝试运行telnet <ip> <port>，只要任意一个连通，则服务正常；如果全部不通，请检查您的网络。

EDAS的agent与云效的agent不能共存。如果您的机器上安装了EDAS的agent，请彻底卸载，或重置操作系统，再尝试安装云效agent。

附agent基础操作：

```
启动：/home/staragent/bin/staragentctl restart;
重启：/home/staragent/bin/staragentctl restart;
查看状态：/home/staragent/bin/staragentctl status;
卸载：
1. /home/staragent/bin/staragentctl stop;
2. rm -rf /home/staragen;
3. rm /usr/sbin/staragent_sn
```

若根据以上排查手段依然未能找到问题，请点击右下角“提问”联系我们。

开发模式

开发模式概述

开发模式意味着：

- 关于各类分支的命名、用法的约定
- 云效工具相应的行为

下面详细介绍各开发模式：

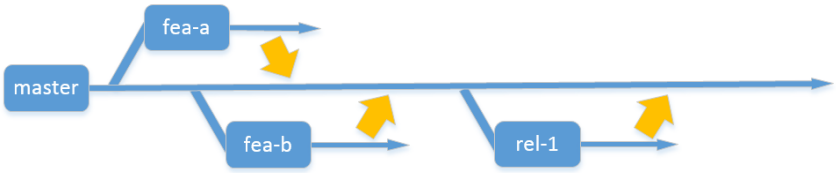
- 自由模式
- 分支模式
- Git Flow模式(待上线)

自由模式

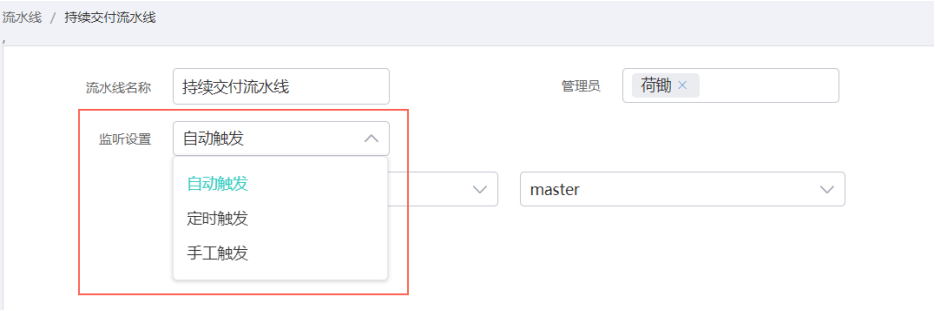
自由模式，顾名思义，用户可以使用任何分支（包括master）进行打包、发布等操作。

在自由模式下，常见用master分支这一条分支来承载开发、集成和发布，这被称作**主干开发方式**。使用这种方式，只有在特定情况下，才会使用其他的分支。包括：

- 确有必要时，拉出feature分支开发特定feature，开发完成并验证后合并回master。
- 确有必要时，拉出release分支，发布特定版本。随后合并回master。



自由模式时，在流水线上，通常默认配置为，master分支变化时自动触发流水线运行，取master分支做构建，并随后部署和发布。如果用其他分支构建，请修改触发条件：



以及构建任务的配置：

详见流水线的配置以及流水线上的构建任务。

分支模式

分支模式是云效支持的三种研发模式的一种。每种研发模式，不仅意味着其中各(类)分支的使用方式，也意味着云效能够向用户提供的相应支持，分支模式也不例外。事实上，云效对分支模式提供了强有力的支持：用户可以只需要关心集成和发布哪些feature分支，而对release分支创建和管理、分支间合并等一系列工作，可以托付给云效系统完成。

本文详细介绍分支模式下，各(类)分支的使用方式。关于云效提供的相应支持，详情请阅读分支模式下的流水线。

master代表最新发布版本

master分支代表最新发布版本。当需要最新发布版本的内容时，直接取分支末端即可。

不论其他哪类分支，都建议一般从master分支创建，并且经常从master分支合并，以便跟上“潮流”，减少将来集成时的各种问题，比如代码合并冲突。

每当软件正式发布前，系统会确保它基于master最新。

每当软件正式发布后，系统会把相应内容合并回master，以便让master分支始终代表最新发布版本。

一般来说，**使用者不要直接“写”东西到master分支。**把“写”的工作交给系统适时自动完成。

在各feature分支上开发

一条feature分支（又称变更分支、开发分支），通常用来承载一个缺陷的修复，或者一个需求（如果不是很大

的话）的开发，或者任务分解后一个任务的开发。

一般来讲，基于master分支最新版本创建feature分支。然后在feature分支上开发、测试，直到这个feature功能完成，质量OK，准备好去集成和发布。

release分支上的集成

release分支用于集成和发布。基于master分支最新版本创建一条release分支，然后把想要集成的各条feature分支合并到这条release分支，进行部署和测试工作。

如果有新的feature分支要加入本次集成，那就把它也合并进这条release分支，然后再次部署并测试。

如果测试发现问题，就到feature分支上修复，然后把它再次合并到release分支，把修复带到release分支。

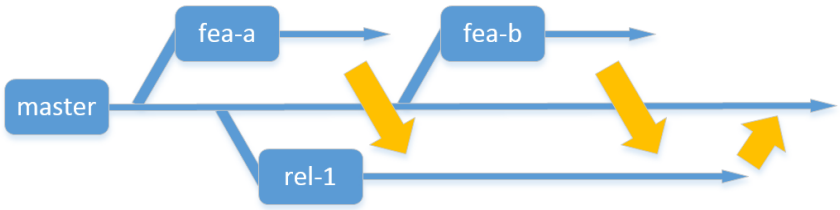
当然如果一个feature的问题太多太大，那干脆就放弃它。也就是说，新建一条release分支，把其他feature分支都合并过去，唯独不再合并这条feature分支。

就像master分支一样，release分支也是由系统自动管理的。**使用者不要直接在上面改代码**，代码修改请总是在feature分支完成。

release分支上的发布上线

当release分支上的质量足够好，想本次想上线的功能也都具备之后，就要考虑发布上线的问题啦。如前面讲的，发布上线前，会确保它基于master最新。而发布后会把release分支合并回master，让master代表最新发布版本。

以上几节介绍的内容，见下图：



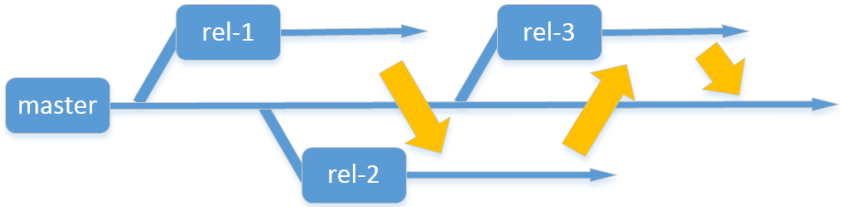
多个环境/流程时

假定要想集成发布上线，要经过日常测试环境上的测试这个流程，还要经过预发环境上的测试这个流程，那么两个流程用一条release分支就有些不合适。因为两个流程可能同时在测不同的feature分支集合。

分支模式用这个办法避免这个问题：每一个测试环境，也就是每个流程，关联它自己的release分支。日常测试、预发测试这两个环境（也就是两个流程），分别关联两条release分支。这样就不会相互影响。推而广之，为正式运行环境，也对应一条release分支。也就是说，每个环境都有对应的release分支。

当把集成成果从一个环境传递到下一个环境时，就是把一个环境下已合并到一起的feature分支，再往另一个环境对应的release分支上合并一遍……这么做有点儿笨。系统实际的做法是，基于master分支创建另一个环境对应的release分支，然后把前一个环境对应的release分支合并到新的release分支上。

本节介绍的内容，对应下图：



小结

以上就是关于分支模式这种研发模式的原理性介绍。更多细节在分支模式下的流水线中讲解。

特性分支管理

特性分支

概述

特性分支是指为一个特定的需求/任务/缺陷创建的分支，在其上完成相应开发后，一般会把它合并到集成/发布分支，与其他改动（若有）一起集成并最终发布。

当研发模式是分支模式时，云效平台为特性分支提供了特别的支持：可以查看特性分支列表，管理每次集成进入哪些特性分支，查看每次发布包括哪些特性分支，自动完成从特性分支到集成-发布分支的合并，等等。

当研发模式不是分支模式时，云效平台暂不提供特别的支持，仅提供基本的代码托管服务。您可以自行完成特性分支（若有）的创建、合并、删除等操作。

特性分支列表

点击吊顶“我的”菜单项，然后从左侧菜单中选择“特性分支”，即进入我的特性分支列表。这里列出的是我是开发者的所有的特性分支。

在进入具体项目后，从左侧菜单中选择“特性分支”，即进入该项目的各应用的特性分支的总列表。

在进入具体项目的具体应用后，从上方菜单中选择“特性分支”，即进入该应用的特性分支列表。

上述各列表中仅显示分支模式的应用的特性分支。另一方面，当具体应用不是分支模式时，它的菜单中不会出现“特性分支”菜单项。

上述各列表中的每条特性分支，有若干属性显示，有可能有相关操作可点击。

新建特性分支

在具体应用的特性分支列表页，点击左上方“新建”按钮，进入新建特性分支页面，填写各项内容，即可新建特性分支。

在新建特性分支页面，也可以选择关联代码库中已有的分支，将其注册为特性分支。于是，该分支就会出现在特性分支列表中，可对其进行特性分支相关的各种操作和查看。

特性分支详情页

在特性分支列表中，点击某条特性分支，即进入该特性分支的详情页。

在特性分支详情页，除展示信息外，通常还有一些操作按钮。比如提交待合并等操作。具体操作方法，详见分支模式中的介绍。

分支集

适用范围

本文档描述的功能仅供部分专有云用户使用。云效公有云尚未开通此功能。

概述

为了完成一个特性（比如一个需求或者研发任务），有时仅修改一个应用的源代码是不够的。这时，就要在不止一个代码库中，拉出相应的特性分支，修改源代码，并随后集成发布。这些为完成该特性的特性分支的集合，我们简称分支集。

如果本企业在云效上配置了分支集这个功能，那么当研发模式是分支模式时，云效平台为分支集提供了特别的支持：可以查看项目的和我的分支集列表，查看分支集包含的特性分支，方便地查看分支集及其各特性分支的状态并进行提交待发布等操作。

如果本企业在云效上没有配置分支集这个功能，或者当研发模式不是分支模式时，云效平台暂不提供特别的支持。您可以自行完成特性分支（若有）的创建、合并、删除等操作。

分支集列表

点击吊顶“我的”菜单项，然后从左侧菜单中选择“特性分支”，进而选择“特性分支”旁的“分支集”标签页，即进入我的分支集列表。这里列出的是我所在的所有的分支集。

在进入具体项目后，从左侧菜单中选择“特性分支”，，进而选择“特性分支”旁的“分支集”标签页，即进入该项目的各应用的分支集的总列表。

上述各列表中的每个分支集，有若干属性显示，有可能有相关操作可点击。

新建分支集

在分支集列表页，点击左上方“新建”按钮，进入新建分支集页面，填写各项内容，即可新建分支集。

分支集详情页

在分支集列表中，点击某个分支集，即进入该分支集的详情页。

分支集详情页的内容主要包括：

- 一些基本信息，如分支集名称、说明、各角色人员、分支集状态等。
- 该分支集包含的特性分支列表。可以在列表中查看这些特性分支的关键信息，进行高频操作，或前往特性分支详情页。详见特性分支中的介绍。

特性分支应用级集成视图

如果您的企业在使用分支模式下自定义流水线功能（目前仅对部分企业开通该功能），当使用向导新建一站式研发解决方案时，若研发模式选择了分支模式(而不是自由模式或Git Flow模式)，则向导将自动创建带有特性分支集成视图的流水线。

阅读本文之前，需要首先学习理解开发模式中的分支模式。详细介绍见这里。

如果您的企业在使用云效专有云版，且开启了全局集成功能，请前往这里阅读相应使用说明。

概述

分支模式下的应用的流水线，通常会带有一个“分支管理器”，也就是这里说的应用级特性分支集成视图。分支管理器负责把该应用上用户指定的各特性分支合并到一条发布分支，然后把发布分支上最新的源代码版本，交给流水线作为输入。流水线据此运行。

典型的，一个应用有“日常”、“预发”、“正式”三条流水线，对应日常环境的部署和测试、预发环境的部署和测试、正式发布。每条流水线有其自己的分支管理器。“日常”流水线成功运行完毕，用户可以在分支管理器中把相应的特性分支一起带到“预发”流水线，进而“正式”流水线，已发布到线上。

待集成区与集成区

每个分支管理器，有待集成区（下图红框）和集成区（下图黄框）两个区域：

两个区域，都是特性分支的列表。

待集成区里，是所有已经开发完毕并做了适当检测，可以进行集成和发布的特性分支列表。做集成和发布时，就从这个列表中挑选，哪些合并到当前流程对应的发布分支，以便部署和进一步集成测试。

集成区里，就是是从待集成区中挑出来的，(打算)合并到发布分支，并随后部署到当前流程相应的运行环境的特性分支列表。这个列表，反映的是当前环境中，包含了哪些代码改动。这些改动将被放在一起测试，进而发布上线。

用户仅需在页面上维护这两个特性分支列表，系统将自动完成发布分支的创建和管理，特性分支到release分支的合并等一系列工作。

当前流程对应的发布分支，显示在集成区中。

日常操作

把特性分支标记为可供集成

在本项目的特性分支列表页（从本项目的左侧菜单中，“特性分支”菜单项进入），或者我的特性分支列表页（点击吊顶“我的”，再点击左侧菜单中“特性分支”菜单项进入），每个特性分支左侧，有“提交待发布”按钮。点击可将该分支状态置为“待发布”，也就是说，标记该特性分支已开发完毕并做了适当检测，可以进行集成和发布了。于是，该特性分支就进入了该应用的各流水线上的分支管理器中的待集成区。

挑选特性分支合入发布分支

在待集成区（图中文案“待发布的变更”）勾选打算合并到发布分支的特性分支。然后点击“提交发布”按钮，即把这些特性分支加入到集成区，并开始自动合并工作。合并完成后，自动继续进行构建和部署等。

如果当时集成区里还没有发布分支，那么此时还没有一条发布分支和当前流水线相关联，于是系统会基于master分支创建一条新的发布分支，然后开始合并工作。

如果当时集成区里已有特性分支，那么此时已有一条发布分支和当前流水线相关联，于是系统就会继续使用这条发布分支，开始合并工作。

在合并过程中，除了合入本次新勾选的特性分支，还将逐个检查已在集成区中，也就是曾经合并到发布分支的特性分支，看是否又有更新还不在发布分支上。如果有，将合入发布分支。类似的，master分支也将被检查。以确保，合并完成后，该发布分支上，包含了新合入的各特性分支，曾合入的各特性分支，以及master分支上的最新内容。

如果合并过程中出现了需要人工解决的合并冲突，页面将提示如何人工解决冲突并继续流程。

特性分支和/或master分支有更新后，更新发布分支并部署

若某(几)条特性分支在合入发布分支后，其内容又有了更新（即，又有人向该分支做了git push操作），或者master分支上因为发布上线而有了更新，那么可以点击“重新部署”按钮一键完成：

- 相应更新发布分支，让它包含master分支和各特性分支最新的内容。
- 把发布分支的最新内容，部署到运行环境。

把某个特性分支的内容，从集成中摘除

在集成区该特性分支条目中，点击右侧的“退出”钮，即可实现该目的。

系统将：

- 基于master分支，自动创建一条新的发布分支，并关联到当前流水线。
- 把去除了该特性分支后，集成区中的所有特慈宁宫分支再次合并到这条新的发布分支。
- 把发布分支的最新内容，部署到运行环境。

因此在效果上，就把该发布分支从集成中摘除了。

把当前集成的全部内容都摘除

点击集成区上方的“退出”钮。于是，集成区被清空。同时，当前流程，不再对应任何一条发布分支。

把集成内容带入另一条流水线

比如，当在日常测试环境的测试结束后，把集成内容带入预发环境进行测试。

每个流水线运行完毕，不会自动触发另一条流水线的运行。如果想把当前流水线的内容带到另一个流流水线，也就是把当前集成区中的所有特性分支带入另一个流水线的集成区，请点击页面上的“进入……”按钮（如果有）。比如“进入预发部署”。

点击“进入……”（如果有）后，系统完成如下操作：

- 将基于master创建一条新的发布分支，并关联到当前进入到的流水线。
- 把前一个流水线对应的发布分支合并到当前流水线对应的这条发布分支。
- 把集成区中所有特性分支合并到这条发布分支。这是为了保证特性分支上的任何更新也都反映到新的发布分支上。

以上，都是针对进入到的流水线（比如“预发”）。原流水线（比如“日常”），没有任何变化：集成区的特性分支列表、对应的发布分支及其上的内容，都不会发生变化。

正式发布后合并回master

这个工作不是在分支管理器中完成的，而是在流水线上完成的。正式发布流水线中，有一个任务是“合并主干”。运行至此时，将把发布分支合并到master分支，以使得master分支总是代表最新发布版本。

特性分支全局集成视图

前言

本文档适用于云效专有云，分支模式，且您的企业启用了特性分支全局集成视图。目前在云效的阿里云公有云版尚不提供此功能。

阅读本文之前，需要首先学习理解开发模式中的分支模式。详细介绍见这里。

概述

使用特性分支全局集成视图，可以在一个网页中，看到本企业所有已开发测试完毕的特性分支/集（待提交区）、排队等待集成发布的特性分支/集（队列区），以及已合并到发布分支，正在走集成发布流水线的特性分支/集（集成区）。

一个特性分支/集，在开发测试完毕后，提交待集成，于是出现在待提交区。进而提交集成，于是出现在队列区

。当条件合适时，将自动进入集成区，在此（与该应用上其他特性分支一起）合并到该应用的一条发布分支，然后发布分支的最新版本开始走该应用的流水线，经过集成测试等一系列步骤后，最终发布上线。

待提交区、队列区与集成区

鼠标点击“我的”->“全局集成”，进入全局集成视图：

视图包括三个区域，都是特性分支的列表。具体来说：

待提交区里，是所有已经开发完毕并做了适当检测，可以进行集成和发布的特性分支列表。做集成和发布时，就从这个列表中挑选，哪些适合去走集成发布流程。挑选好后，点击“提交发布”钮，于是进入队列区。当然，也可以在特性分支/集上，直接“提交发布”，跳过待提交区，直接进入队列区。

在队列区，所有分支按照先后顺序排列。从上到下依次判断，如果某个分支集，它所包含的各应用，目前在集成区都没有正在集成的特性分支，那么它就会自动进入集成区。

在集成区，每个应用，各个特性分支被合并到一条发布分支。随后，发布分支的内容去跑流水线，经过构建、部署、测试等环节，若一切顺利，最终发布到正式生产环境，发布分支被合并回master分支。

用户仅需在页面上向待提交区和队列区增减特性分支/集，系统将自动完成排队集成，特性分支到发布分支的合并等一系列工作。

日常操作

把特性分支标记为可供集成

在本项目的特性分支列表页（从本项目的左侧菜单中，“特性分支”菜单项进入），或者我的特性分支列表页（点击吊顶“我的”，再点击左侧菜单中“特性分支”菜单项进入），每个特性分支左侧，有“提交待发布”按钮。点击可将该分支状态置为“待发布”，也就是说，标记该特性分支已开发完毕并做了适当检测，可以进行集成和发布了。于是，该特性分支就进入了全局集成视图中的待提交区。

也可以在分支集页面中，选中一个或多个特性分支，点击“提交待发布”，共同进入待提交区。

挑选特性分支去排队集成

在全局集成页面的待提交区，勾选打算去排队集成的特性分支/集，点击“提交发布”，于是进入队列区。

此外，还可以一步完成上述“把特性分支标记为可供集成”“挑选特性分支去排队集成”两个步骤：在特性分支或分支集上，点击“提交发布”，于是直接进入队列区。

自动从队列区进入集成发布流水线

这一步无需人工干预，系统会按照算法（见上文介绍）自动将满足条件的特性分支/集带入集成区。于是，每个应用，各特性分支被自动合并到一条从master分支拉出来的新的发布分支，并随后走集成发布流水线，直到发布上线。

如果合并过程中出现了需要人工解决的合并冲突，页面将提示如何人工解决冲突并继续流程。

正式发布后合并回master

这个工作不是在全局集成视图上完成的，而是在流水线上完成的。流水线中，有一个任务是“合并主干”。运行至此时，将把发布分支合并到master分支，以使得master分支总是代表最新发布版本。

机器资源管理

机器资源管理概述

适用场景

云效支持不同类型的运行环境及相应的部署方法。比如，通过阿里云EDAS管理环境资源，并通过EDAS部署，详见部署配置：通过EDAS部署。比如通过阿里云容器服务管理环境资源，并通过容器服务部署，详见部署配置：通过容器服务部署。本文档与这样的场景无关。

云效也支持用户通过可以自定义的部署脚本，将应用程序直接部署到机器上运行。在这种情况下，云效就需要直接管理企业的机器资源，以便可以按部署配置：通过脚本部署中的描述，配置各应用各环境分别使用企业的哪些机器资源，进而把应用部署到那里。本文档描述云效如何管理企业的机器资源。

把已有机器关联到云效本企业

你可以在阿里云上自行购买ECS机器，随后把机器关联到云效本企业。也可以把企业自有机房等其他途径的机器，关联到云效本企业，只要这些机器可以从公网访问。详见把已有机器关联到云效本企业。

通过云效直接购买机器

你也可以通过云效直接购买阿里云ECS机器。这需要：

- 第一步，在企业管理页面配置授权云效，让云效以指定阿里云账户的身份完成机器购买等操作。
- 第二步，在企业管理页面配置ECS模板，于是云效就知道要购买什么样的机器，做怎样的初始化工作。
- 第三步，在企业管理页面购买ECS机器，告诉云效用哪个ECS模板，为本企业购买多少台。

把已有机器关联到云效本企业

企业机器资源的管理

企业管理员，点击右上角设置，进入“企业设置”页面，点击左侧“主机管理”，然后点击添加主机，通过安装agent的方式，将机器加入云效的管理。

按照上述方式，您可以生成一条命令，该命令可以在指定区域的多台机器上反复运行。为了安全起见，您可以设定该命令的有效时间。

用这种方法，您不仅可以把您的阿里云ECS机器纳入管理，还可以把您其他途径的机器纳入管理，只要他的机器可以访问公网。

agent安装依赖说明：

```
#该agent依赖Python2.7，当您的机器上的Python版本非2.7，或是您的机器上缺失了zlib-dev openssl-devel  bzip2-devel包，则请按照如下步骤，首先安装Python2.7：
wget "http://agent-install.oss-cn-hangzhou.aliyuncs.com/Python-2.7.13.tgz"
#下载及解压压缩包：
tar -zxvf Python-2.7.13.tgz
#安装必要的工具包：
#centos/redhat系统使用命令
yum install -y zlib-dev openssl-devel bzip2-devel
#debian/ubuntu系统使用命令
apt-get install -y zlib1g libssl-dev libbz2-dev
#在解压后的路径下执行:
./configure --with-zlib
make
make install
```

安装agent完成后，如果在“机器管理”里没有发现新加入的机器，请参照添加Agent失败FAQ进行排查

添加Agent失败FAQ

我在机器上安装了Agent，但在企业的机器列表中看不到

请按照下列步骤依次排查。执行完每一步之后，请确认问题是否解决，若未解决，请继续尝试后续步骤。

目前agent只支持64位的Linux操作系统。

确认您是否曾经使用另一个企业的agent安装命令在该机器上执行过，如果是，请删除/usr/sbin/staragent_sn（正常情况下该文件内容为机器SN，SN为机器唯一标识，并与特定企业绑定），并重装agent。

执行命令cat /usr/sbin/staragent_sn查看内容。若文件内容为空，删除此文件，并重装agent（PS：请勿手动修改该文件）。

执行命令/home/staragent/bin/staragentctl status查看agent状态。若输出异常（比如ServerAddr为空，或者报错），请执行命令cat /home/staragent/conf/staragent.conf查看文件内容，如果文件存在，且其中的URL为rdc-xxx.aliyuncs.com或者staragent-configservice.aliyuncs.com，则为正常。如果不是，有可能是您的机器之前安装过其它产品的agent，请重装云效agent。

查看cat /home/staragent/conf/channels.conf是否存在，如果不存在，请执行命令：curl 'http://<从staragent.conf中获取的服务URL>/api/configservice?action=findChannelListForAgent&agentIpList=101.37.119.155%2C10.80.237.52&needAllChannels=true&serviceTag=ea263ff8-2d60-48f0-86c4-33a04214cad9&version=2'，如果结果类似{"appCode":"_successful_","msg":"","restCode":200,"result":{"allChannels":[],"channelIPPort":[{"ip":"100.100.18.88","port":8000},{"ip":"100.100.18.89","port":8000},{"ip":"100.100.45.99","port":8000},{"ip":"182.92.29.36","port":8000},{"ip":"182.92.29.39","port":8000},{"ip":"100.100.45.100","port":8000}]，请尝试重启agent（参看下面的agent基础操作）。如重启后/home/staragent/conf/channels.conf仍不存在，请点击云效页面右下角“提问”联系我们。如果不能返回类似结果，则表示您的机器到<从staragent.conf中获取的服务URL>的连接有问题。有可能是在安装agent时候，选错了区域。请在添加机器页面选择正确的区域，生成agent安装命令，重装agent。

执行命令cat /home/staragent/conf/channels.conf查看该文件，内容会是一个ip+port的列表，尝试运行telnet <ip> <port>，只要任意一个连通，则服务正常；如果全部不通，请检查您的网络。

EDAS的agent与云效的agent不能共存。如果您的机器上安装了EDAS的agent，请彻底卸载，或重置操作系统，再尝试安装云效agent。

附agent基础操作：

```
启动：/home/staragent/bin/staragentctl restart;
```

```
重启：/home/staragent/bin/staragentctl restart;
查看状态：/home/staragent/bin/staragentctl status;
卸载：
1. /home/staragent/bin/staragentctl stop;
2. rm -rf /home/staragen;
3. rm /usr/sbin/staragent_sn
```

若根据以上排查手段依然未能找到问题，请点击右下角“提问”联系我们。

配置授权云效

概述

作为一站式研发协同平台，云效在很多场景下需要和阿里云的其它云服务进行交互，比如ECS购买、运维相关的云服务集成等。云效以指定的阿里云账户身份，通过Open API调用，完成上述操作。为此，在云效本企业中，企业管理员需要先做适当的授权配置。本文详细讲解如何进行这样的授权配置。

添加授权并绑定

确定付费账户

请确定一个阿里云主账户。云效将以这个账户的身份购买机器。

该阿里云主账户满足下述两个选项之一：

选项一，该阿里云主账户本身是云效用户，且是云效中本企业的企业管理员。

选项二，该阿里云主账户的某个RAM子账户是云效用户，且是云效中本企业的企业管理员。该RAM子账户还需要满足一个条件：它具有为该阿里云主账户创建RAM角色的权限。一般来说，这通过在RAM控制台中，向该RAM子账户（或其所属组）添加系统授权策略“AliyunRAMFullAccess”来完成。相关知识请参阅RAM授权策略管理帮助文档。

授权

这一步的目的是，授权云效，可以用该阿里云主账户的身份，完成购买机器等操作。这一步完成后，云效就有此权限了。至于在云效的本企业中，是否用此权力，将在下一步配置绑定。

请以上述阿里云主账户或其RAM子账户登录云效本企业，从页面右上角齿轮图标处进入“企业设置”，选择“主机管理”，进而选择“授权”，即进入授权配置页面。

在授权配置页面的“我授权并绑定”区域，点击“授权”，并在随后页面中点击确认，即可完成。

原理：授权意味着，在阿里云RAM服务中，为该主账户添加了“AliyunRDCDefaultRole”这个角色并赋予其授权策略“AliyunRDCRolePolicy”，允许云效通过该角色完成操作。可前往RAM控制台中，该账户的RAM角色管理页面查看。相关知识请参阅RAM角色帮助文档中，“服务角色”相关内容。

错误提示与解决办法：如遇弹窗提示“权限不足”，说明该RAM子账户没有为其阿里云主账户创建RAM角色的权限。请参考上文“确定付费账户”->“选项二”解决。

绑定

这一步的目的是，告诉云效，当云效中本企业要进行购买机器等操作时，以该阿里云主账户的身份完成。

在授权配置页面（进入方法见上文）的“我授权并绑定”区域，点击“绑定”，即可完成。

修改绑定

当本企业已绑定某个阿里云主账户后，可以修改为绑定另一个阿里云主账户，同时解除原绑定关系。过程与初次配置授权时相同：

- 确定要更换到的阿里云主账户。
- 若尚未授权，以该主账户登录云效本企业，在授权配置页面的“我授权并绑定”区域点击“授权”。
- 在授权配置页面继续点击“绑定”。

解除绑定

可以解除当前本企业与某个阿里云主账户的绑定关系。方法为，企业管理员在授权配置页面的“解除绑定”区域，点击“解除”。

账户与企业解除绑定关系后，若希望亦去掉授权（即云效以该账户名义操作的权力），可前往RAM控制台中，该账户的RAM角色管理页面中，删除“AliyunRDCDefaultRole”角色。相关知识请参阅RAM角色帮助文档。

配置ECS模板

概述

云效为了方便企业管理员添加相同云效环境的机器，提供了ECS镜像模板功能，企业管理员可以按照语言类别维护ECS镜像模板，通过ECS镜像模板购买相同运行环境的ECS。

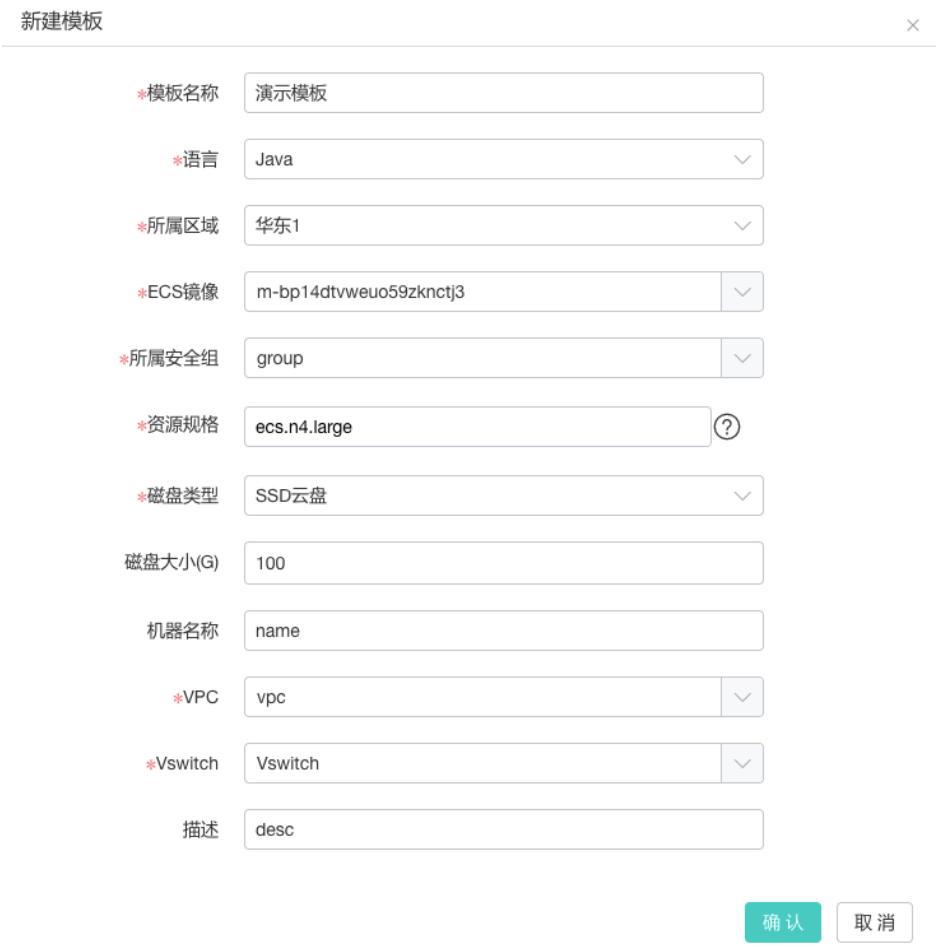
使用限制

- 只支持阿里云云服务器ECS
- 只支持VPC网络
- 已经创建ECS自定义镜像
- 已经创建VPC和交换机（Vswitch）
- 已经创建安全组

镜像维护

企业管理员，点击右上角设置，进入“企业设置”页面，点击左侧“主机管理”，然后点击“ECS镜像模板”。

确保已经完成配置授权云效，点击“新建模板”来添加ECS镜像模板



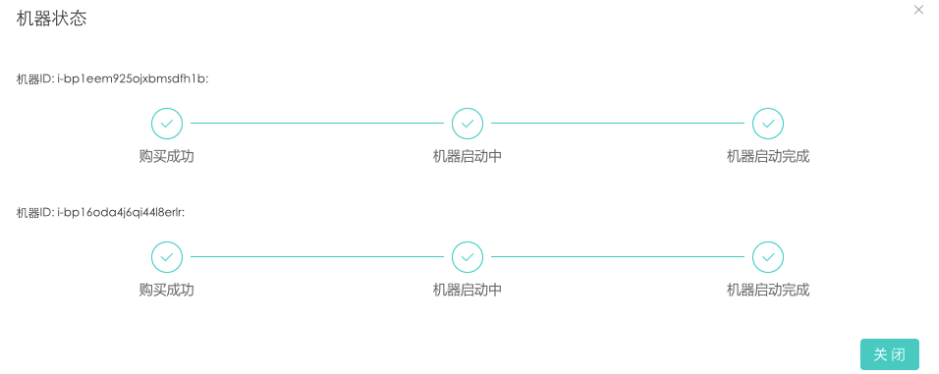
购买ECS机器

概述

云效为用户提供购买阿里云云服务器ECS功能，在使用之前，请确保先完成配置授权云效和配置ECS模板。在购买ECS完成后，云效自动启动ECS，并安装agent，直接加入到企业的机器管理列表中。

企业管理员，点击右上角设置，进入“企业设置”页面，点击左侧“主机管理”，然后点击“购买机器”购买ECS。

购买成功后，会弹出云服务器ECS启动流程。



私有云部署

概述

云效新增私有云部署功能，支持企业私有云或者自建机房等不能访问公网机器部署。该方案只需要一台能访问公网机器作为部署代理机，其他部署应用的机器都可以不直接访问公网。

快速开始

第一步：维护代理机

企业管理员，点击右上角设置，进入“企业设置”页面，点击左侧“机器管理” ，然后点击“代理机”,维护代理机信息。

第二步：设置代理机

代理机安装docker，启动docker，拉取代理镜像（docker pull registry.cn-hangzhou.aliyuncs.com/rdctest/private-idc-proxy:1.0），确保80，8000，443端口没有被占用，启动镜像（docker run —restart=always -p 8000:8000 -p 80:80 -p 443:443 -d registry.cn-hangzhou.aliyuncs.com/rdctest/private-idc-proxy:1.0）。

第三步：添加主机，

选择私有云部署，生成安装命令，在不能访问公网的部署机器上安装agent，安装成功后，就可以进行部署了。立即使用

FQA

添加机器安装agent后，在机器列表里不能展示

检查文件/home/staragent/conf/channels.conf是否存在，内容是否类似格式如下（注意，proxy机器的ip和8000之间需要是一个tab，而不能是空格）

```
<代理机器的ip>  8000
```

部署失败

检测/etc/hosts,确保已经添加了如下内容

```
192.168.0.152 oss-cn-beijing.aliyuncs.com
```

测试环境管理

测试环境管理功能介绍

适用场景

云效的测试环境管理功能用来支持类似这样的场景：

某个产品，有几套测试环境，比如第一套、第二套测试环境。当前同时有多个特性（功能）在其相应的特性分支开发。

某个特性，比如特性A，现在希望在特性分支上，在代码改动还没合并到集成/发布分支的时候，就对该特性进行比较充分的调试和测试，以减轻集成时的负担。于是现在要把一套测试环境分配给它。当前第一套测试环境空闲，于是分配给特性A。类似的，第二套测试环境分配给特性B。

当特性A开发测试完毕，合并到集成-发布分支，于是不再需要第一套测试环境时，就把第一套测试环境置为空闲状态，还回“池子”。于是等将来特性C需要测试环境时，再把第一套测试环境分配给特性C。

一些基本概念

环境实例：上文中所说的第一套、第二套测试环境，在云效中，我们把它称作是两个环境实例。环境实例，对应着可以实际用来测试的真实环境资源。

环境模板：一个环境模板包含若干环境实例，这些环境实例彼此相像，可以任选一个使用。比如上文中的两个环境实例，就属于同一个环境模板。环境模板上，定义了它包括哪些应用，每个应用如何部署等信息，这些信息是各环境实例（通常）都相同的。

测试环境管理有哪些功能

记录和展现各环境实例是否空闲，正在被谁使用，使用的目的，部署了各应用的什么版本等信息，以方便各环境实例的分配、查看和释放。

把各应用合适的版本构建并部署到特定环境实例。比如，为开发特性A，在应用1和应用2的代码库里分别拉出了特性分支1-A和2-A，那么测试特性A时，可以在特定环境实例中，一键完成部署应用1的分支1-A上最新版本、应用2的分支2-A上最新版本、以及其他应用的最新发布版本。

如何开始使用

第一步，搭建若干套真实的测试环境。这一过程可能包括购买ECS机器并初始化；购买RDS数据库服务并初始化；配置应用间基于HTTP的调用的域名绑定，以保证同一套测试环境中的应用可以互相访问而不干扰其他测试环境；等等。这些内容，云效测试环境管理功能将逐渐提供自动化的更便捷的支持。当前还请自行完成。

第二步，在云效中，创建环境模板条目，包含测试环境相关各应用。每个应用，其部署路径、部署脚本等配置，从日常环境(详见环境与环境级别介绍）中被自动复制过来。

第三步，在云效中，该环境模板下，创建若干个环境实例条目。每个条目对应一套真实的测试环境。该条目中，指定每个应用要部署到具体哪台机器。

第四步，通过云效，向一个空闲的特性环境实例部署各应用的特定版本。

使用入口

云效测试环境管理功能，有几个相关的入口。

第一个是，在（包含若干应用的）具体项目中，在完成相关配置后，可以从左侧菜单栏“测试环境”菜单项进入，浏览该项目名下的各环境模板和各环境实例。方便起见，此处还能浏览本企业全部机器资源及使用情况。

第二个是，在具体应用的具体特性分支的详情页面，其“概述”标签页，有测试环境相关信息块。并且有名为“测试环境”的标签页，是测试环境相关详细内容。

（即将上线）第三个是，在具体应用的“环境”页面，展示在该应用上定义的各个环境，这些是与本文讲述的

“测试环境管理”功能无关的。而如果该应用参与到了本文讲述的环境模板和环境实例，那么在具体应用的“环境”页面，将追加展示包含本应用的环境模板和环境实例中，本应用相关内容，并可以通过链接前往该环境模板和环境实例的详情页。

进入实际操作

请前往测试环境管理操作指南继续阅读。

测试环境管理操作指南

使用须知

1. 请首先阅读测试环境管理功能介绍。
2. 该功能必须要有特性分支，因此无法创建特性分支的应用不能支持，比如自由模式的应用不支持。
3. 应用必须有日常环境(详见环境与环境级别介绍），而且部署成功过。
4. 多个应用部署成功后，服务间基于HTTP的调用的域名绑定需要您自行完成。

在项目中开启“测试环境”服务

登录云效，选择项目——具体项目——设置——服务，刷新后就在左侧就会出现“测试环境”菜单。

从测试环境入口使用

1. 新建测试环境模板。模板就是定义哪些应用在一起组成了一个测试环境。

2. 点击新建模板后，输入模板名称，以及管理的应用和应用部署顺序，点击保存。

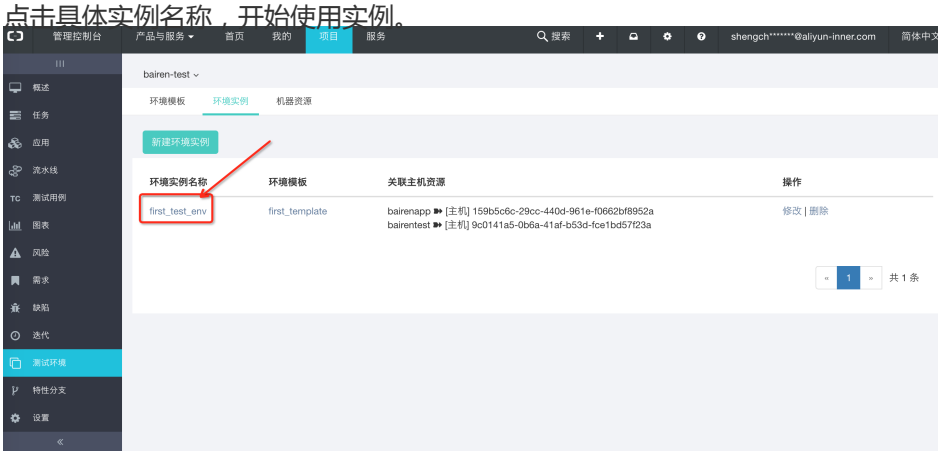
3. 展示环境模板列表。

4. 创建环境实例。环境实例将绑定运行资源(机器)，可以供应用实际运行。一个环境模板可以对应多个环境实例。

5. 填写环境实例名称，选择模板，配置关联机器。

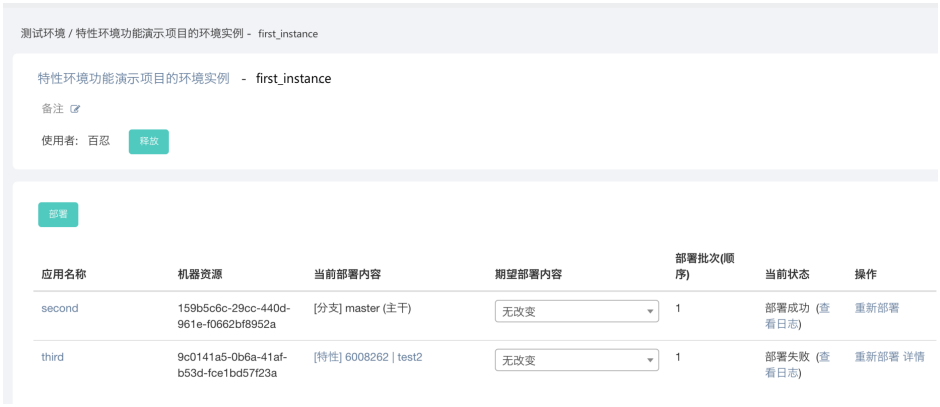
6. 保存后可以看到所有环境实例。

7. 点击具体实例名称，开始使用实例。



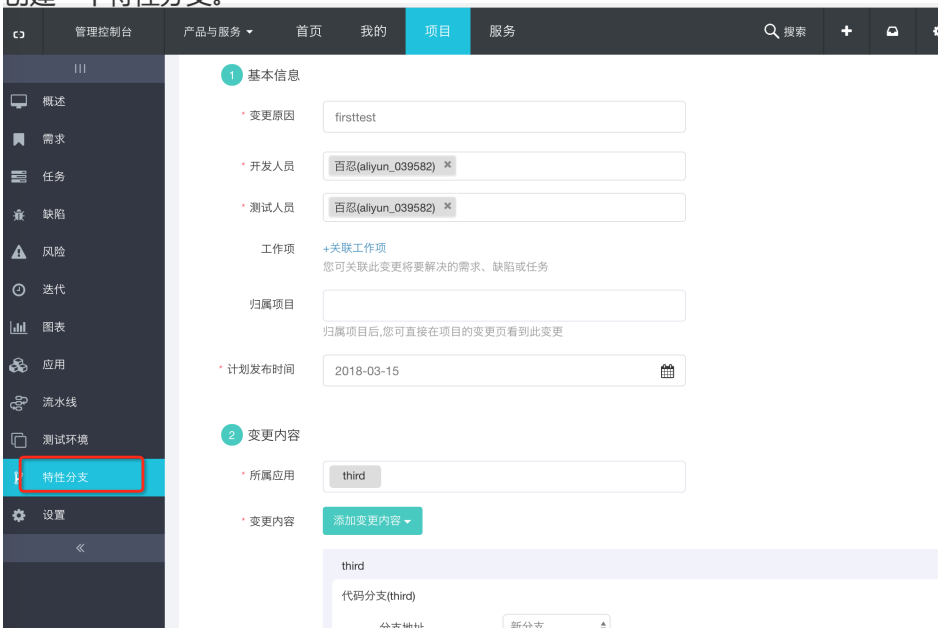
8. 在实例详情里选择要部署的特性分支，然后点击部署。注：“无改变”就是不部署当前应用，“主干”不属于特性分支，必须选择一个特性分支（一个环境实例的一次部署中，至少要有一个应用选择特性分支，否则不能发起部署）。如果没有请到特性分支菜单里创建一个。

9. 部署后在详情页面可以点击看部署状态和重新部署。



从特性分支入口使用

1. 创建一个特性分支。



2. 在特性分支详情中，选择使用特性环境或者增加特性环境。

3. 选择模板和对应的实例，如果没有就会跳到之前的创建页面去创建，参考上面步骤。

4. 确定后，就会展示当前特性分支使用的所有环境实例。

5. 在特性环境里可以选择查看具体对应的流程详情以及可以再使用其他特性环境。

机器资源

在机器资源里可以看到**当前企业下所有机器**的使用情况，比如在哪个模板、实例中定义关联了该资源，以及当前资源是否在实例中部署了应用。

云服务集成

概述

为什么要在云效中集成云产品

阿里云提供了大量的优秀的云产品，比如ECS，SLB，云监控，日志服务，帮助用户进行线上服务的部署，运维，监控，告警。

但实际用起来之后，会发现一个问题。那就是有些概念，比如机器分组，会在多个产品中重复实现。假设有一个线上的Web应用，包含了5台机器。那么需要在日志服务中将这5台机器配置到一个分组，然后再在云监控中把同样的5台机器分到云监控的分组，再把这5台机器挂在某个SLB下。当应用扩容一台机器时，各个云服务的机器组也需要手工同步。造成这种不便的原因就是缺乏了一个基础的公共概念：应用。

云效作为研发协同平台，以应用为核心的。应用下面有不同的环境，每个环境对应一个机器组，使用环境的概念，就可以将各个云产品的机器组的概念统一起来。通过Open API的方式，云效可以在环境机器有变化时，把上述的这些相关服务配置好。同时应用也会成为一个访问其他各个云产品的聚合入口，从而更好的将研发过程和运维过程有机的结合起来，助力企业的DevOps转型。

使用的前提

授权云效

云效通过阿里云Open API和其它产品进行集成。通常企业会使用统一的账号来进行进行各种云产品的购买和管

理，因此在使用云产品集成功能之前，需要先使用该云账号对云效进行授权。后续对这些云产品的操作都会使用以该账号的身份进行。

已集成的云产品

目前开放的云产品集成：云监控。

后续会陆续集成更多的云产品，也欢迎您在页面底部的“以上内容是否对您有帮助”环节进行评分，进而将您的需求反馈给我们。

云监控

概述

阿里云的云监控可以对一组ECS机器进行基础监控。对于Web服务来讲，每个ECS分组本质上对应的是一个应用的一个环境下的机器。在关联云监控后，云效会自动同步环境下ECS到相应的云监控分组，并自动安装云监控agent。环境关联或删除机器，都会同步维护对应的云监控分组。

在使用之前，请确保先完成配置授权云效。

使用限制

- 只支持阿里云云服务器ECS
- 只支持VPC网络

开启方式

开发人员，点击相关应用，进入“环境”页面，点击“云服务”。

点击开启radio button。如果授权账户下尚不存在云监控分组，则云效会直接弹出新建云监控分组的对话框。

填入期望的云监控分组名称和告警联系人分组。点击确定之后，云效就会使用授权的云账号创建相应的云监控分组，并把当前环境下的机器添加到该分组中。如果您的机器尚未安装云监控的agent，云效也会自动进行安装。

如果授权账户下已经存在云监控分组，则点击开启radio button之后，您可以在下拉框中选择已有的分组，或者创建新分组：

如果您选择关联已有的分组，云效只会根据该环境下机器的增删操作，对该云监控分组进行同步，不会影响该分组已经存在的机器。

去除关联

如果您不再希望云效对云监控分组进行同步，请点击关闭radio button，然后点击右上角的全部保存。则云效会停止环境内机器到云监控分组的同步操作。

功能限制

- 云效新建云监控分组后，不会关联告警模板到该分组，需要您到云监控控制台手动进行关联。

Maven私库

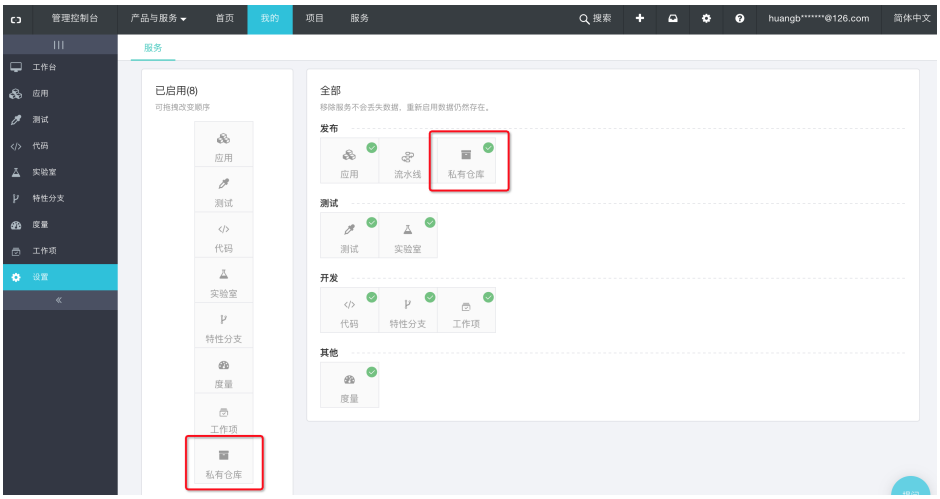
概述

背景

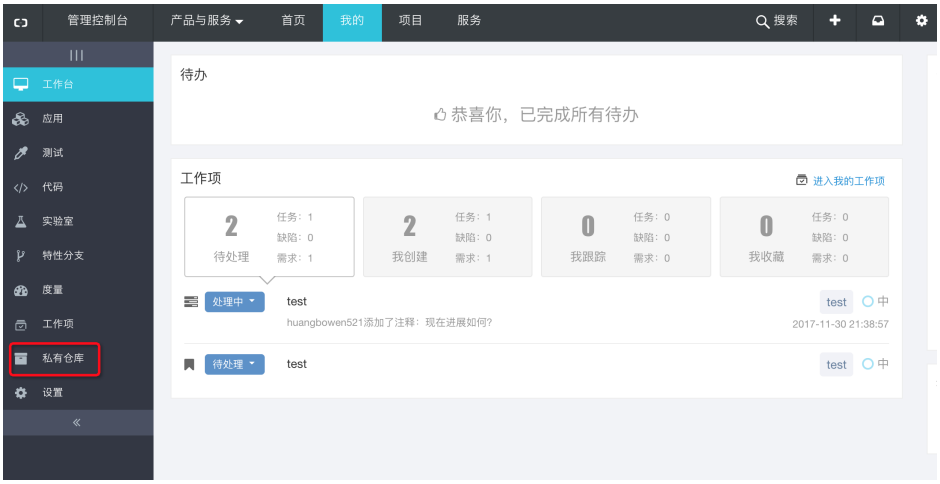
在云效中如果需要上传、下载私有的二方库，可以使用云效的企业级Maven私有仓库服务。

将私有仓库服务加入侧边栏

点击‘我的’链接，选择左侧菜单栏中的‘设置’按钮，将‘私有仓库’服务加入到左侧菜单栏中。

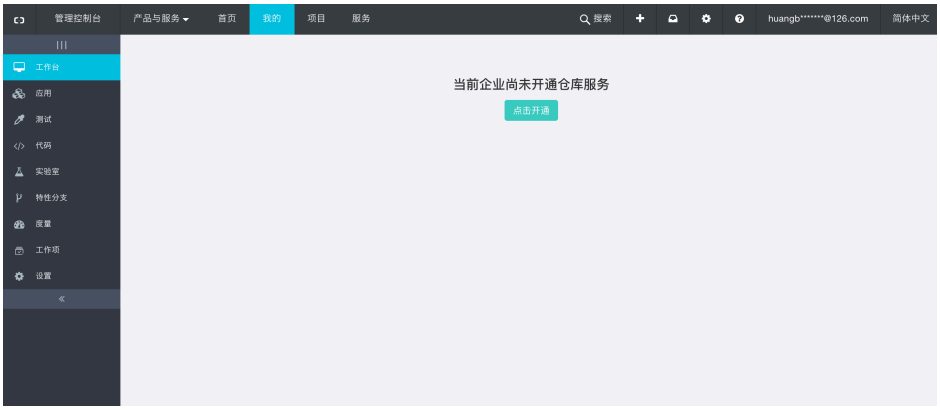


这样在左侧菜单栏会看到‘私有仓库’链接。



开通仓库

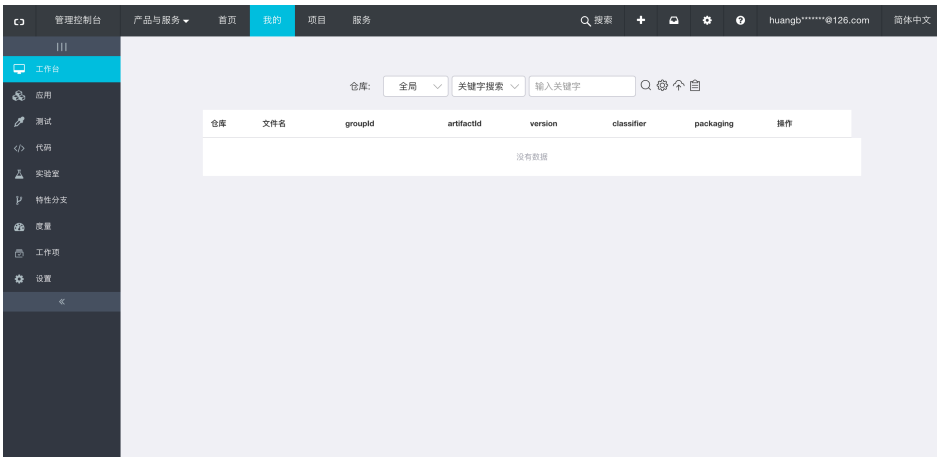
虽然启用了私有仓库服务，但云效并没有真正的为您创建企业级Maven私有仓库。点击左侧菜单栏‘私有仓库’链接后，如果您是企业管理员，会出现以下界面：



‘点击开通’即可开通仓库服务。

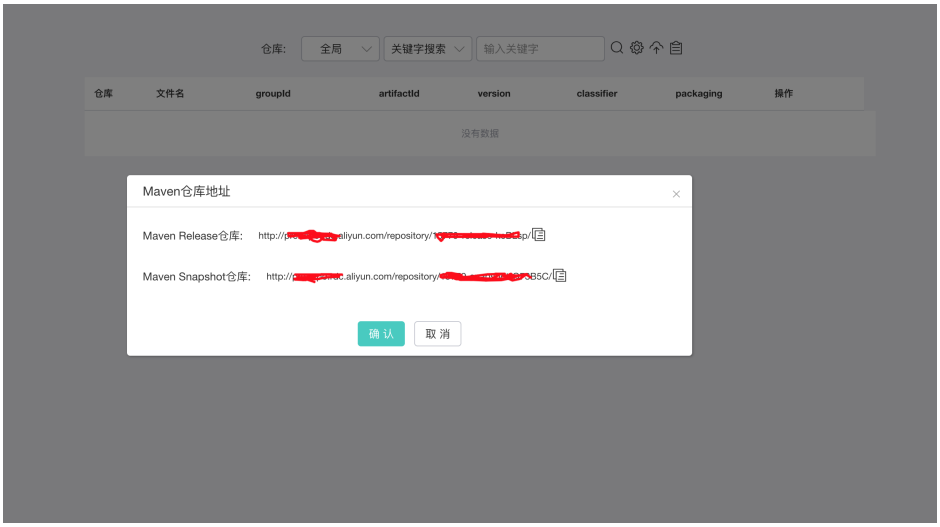
企业的普通用户并没有开通仓库的权限，则需要联系您的企业管理员进行开通操作。

开通成功以后显示界面如下：



仓库地址

云效会自动为该企业生成两个Maven私库，一个是Release仓库，用于存储正式版本的二方库；另一个是Snapshot仓库，用于存放Snapshot版本的二方库。



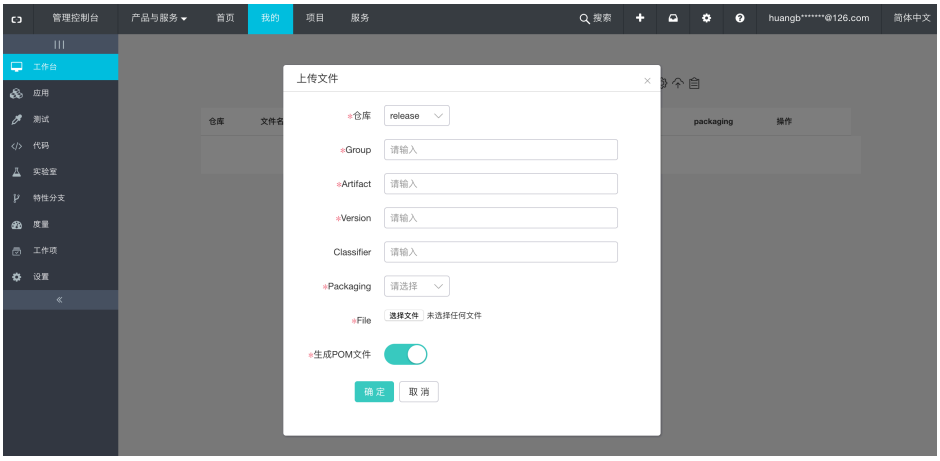
settings.xml配置

私有仓库不允许匿名上传和下载二方库，云效为每个私有仓库生成了相应的用户名和密码。请注意不要泄露该用户名和密码。

用户可以通过该页面下载完整的settings.xml文件,也可以根据自己的需求在settings.xml文件中添加公共仓库的镜像地址。

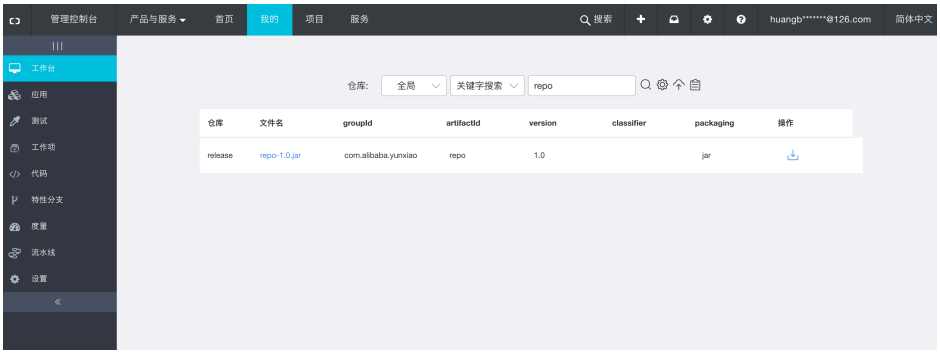
上传二方库

用户可以通过UI上传二方库。目前支持通过GAV的模式进行上传，单个二方库的大小限制为300M。



检索

对二方库的检索支持关键字搜索和GAV搜索两种模式。



用户可以查看检索出来的二方库的基本信息，也可以下载二方库。

用户可以查看在云效构建中使用Maven私有仓库服务了解更多内容。

在云效构建中使用Maven私有仓库服务

当用户开通了Maven私有仓库服务后，云效会为用户生成两个私有仓库，一个用于存放release版本的二方库，一个用于存储SNAPSHOT版本的二方库。

release仓库地址示例:

```
https://repo.rdc.aliyun.com/repository/24409-release-87w1FL/
```

SNAPSHOT仓库地址示例：

```
https://repo.rdc.aliyun.com/repository/24409-snapshot-AA0Hx0/
```

云效构建时从私有仓库下载二方库

如果用户项目代码库的根目录没有Maven的settings.xml文件，那么云效构建时会为用户自动生成一个

settings.xml文件。该文件不仅包括了maven.aliyun.com等公共仓库地址，也自动引入了该企业的两个私有仓库地址。所以用户通过云效构建时，无需任何额外配置就可以实现下载私有仓库中的二方库。

如果用户项目代码库的根目录定制了Maven的settings.xml文件，那么用户需要自行将私有仓库的配置信息添加到该文件中。具体可以参考Maven私有仓库服务。

通过云效上传二方库到私有仓库

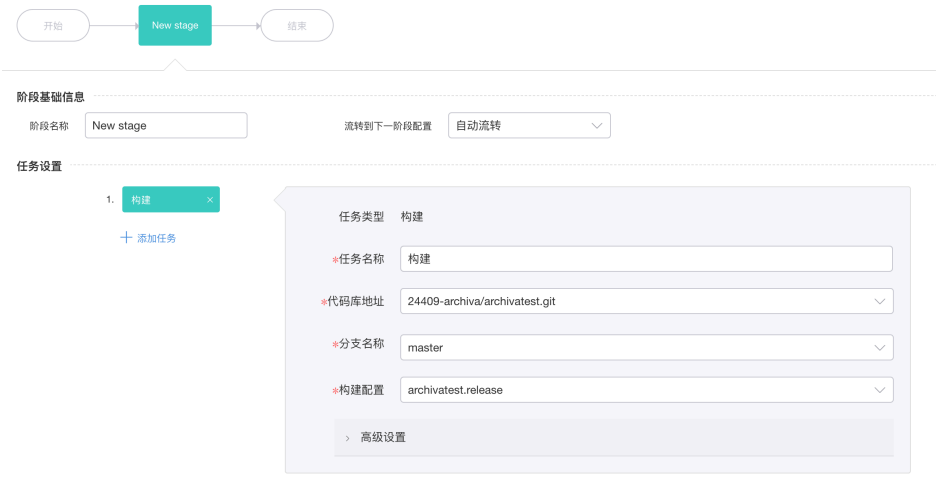
如果想通过流水线发布二方库到私有仓库，可以先在项目代码库根目录的pom.xml中指定分发的仓库地址，示例如下：

```
<distributionManagement>
<repository>
<id>rdc-releases</id>
<url>http://repo.rdc.aliyun.com/repository/24409-release-87w1FL/</url>
</repository>
</distributionManagement>
```

项目代码库根目录的<应用名>.release中指定构建命令为上传二方库，例如：

```
build.command=mvn clean deploy -Dmaven.test.skip
```

然后在云效中创建一条流水线，创建一个构建任务。示例配置如下：



也可以在构建命令中指定分发的仓库地址。方式是在Maven命令中指定-DaltDeploymentRepository参数。

```
build.command=mvn -DaltDeploymentRepository=rdc-
releases::default::https://repo.rdc.aliyun.com/repository/24409-release-87w1FL/ deploy -Dmaven.test.skip
```

altDeploymentRepository指定了id::layout::url。在云效的Maven私有仓库服务中release仓库的id为rdc-releases。SNAPSHOT仓库的id为rdc-snapshots。layout一般使用默认值default，而url则为release仓库或SNAPSHOT仓库的url。上传到SNAPSHOT仓库的示例命令如下：

```
build.command=mvn -DaltDeploymentRepository=rdc-snapshots::default::https://repo.rdc.aliyun.com/repository/24409-snapshot-AA0Hx0/ deploy -Dmaven.test.skip
```

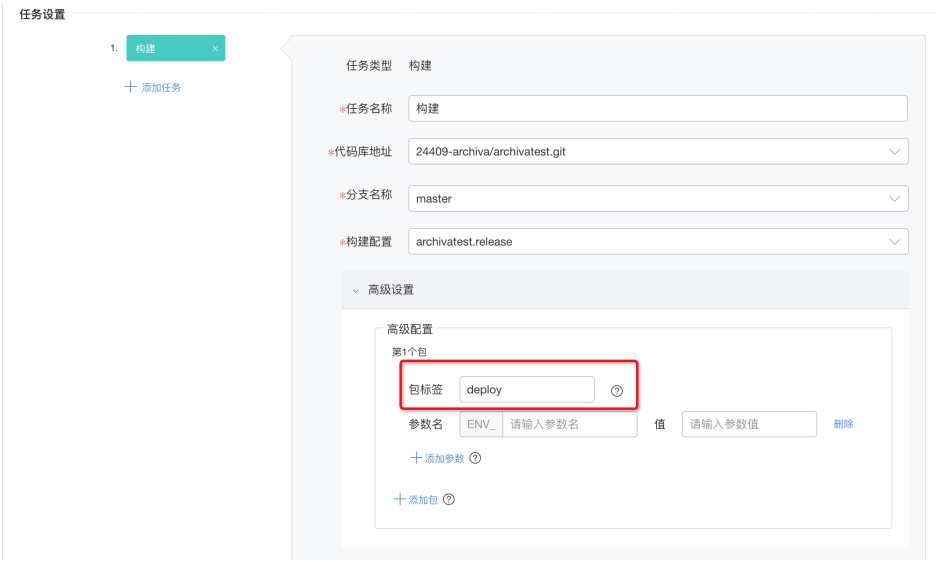
单应用同时支持应用构建和二方库发布

如果一个项目既要实现打包和部署，又要为其他项目提供SDK二方库，那么单个build.command配置就无法满足这种场景。您可以使用传入参数改变构建行为中的方式，使用PACKAGE_LABEL区分不同的构建命令。一个完整的例子如下。

在代码库根目录的<应用名>.release文件中指定如下配置项:

```
deploy.build.command=mvn -DaltDeploymentRepository=rdc-releases::default::https://repo.rdc.aliyun.com/repository/24409-release-87w1FL/ deploy -Dmaven.test.skip
```

这个配置项使用前缀deploy作为包标签。然后创建一条流水线，在构建任务中打开高级配置项，进行如下配置：



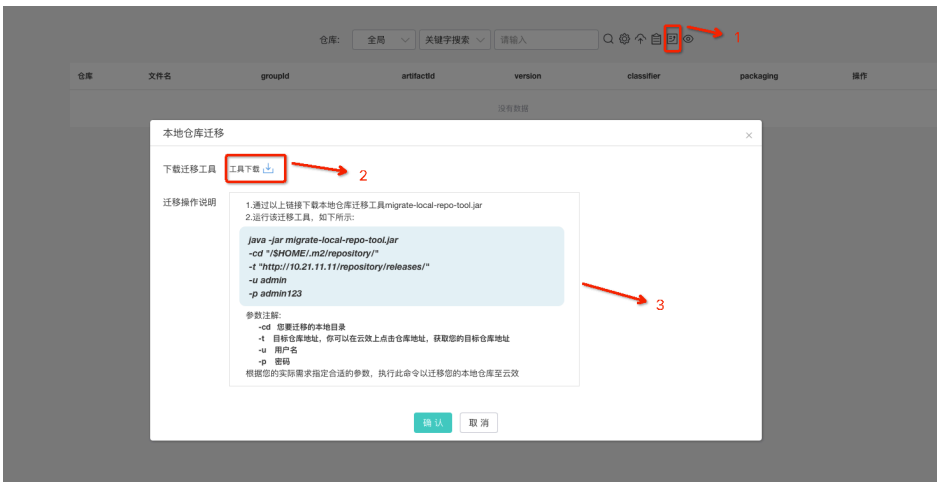
与上一个构建任务唯一不同的是这里将高级配置中的包标签的值改为deploy，这样触发构建时执行的就是deploy.build.command中指定的命令。

将已有私库迁移至云效

本文档帮助您将已有Maven私库中的artifacts批量迁移到云效的Maven私库中。

本地artifacts迁移

您可以在云效私有仓库管理界面，获取到详细的本地仓库迁移操作说明，如下图所示：



操作步骤：

1.下载迁移工具migrate-local-repo-tool.jar

2.在您本地运行该迁移工具，（请首先确保您的JDK版本为1.8及以上）。运行命令如下：

```
java
-jar migrate-local-repo-tool.jar
-cd "/$HOME/.m2/repository/"
-t "http://10.21.11.11/repository/releases/"
-u admin
-p admin123
```

参数注解：

-cd 您要迁移的本地目录，例如：/$HOME/.m2/repository/

-t 目标仓库地址（您可以在【私有仓库】界面点击仓库地址，获取您的目标仓库地址）

-u 用户名

-p 密码

注：用户名和密码为您要上传的目标仓库用户名及密码，您可在setting.xml中获取对应仓库的username和password

根据您的实际需求指定合适的参数，然后执行该命令，稍等片刻，您的本地仓库中的artifacts将会被批量迁移到云效中您所指定的Maven私库中。

将现有的私库关联到云效的私库

云效提供了关联远程仓库的功能。你可以将云效仓库关联到你现有的私服仓库。当使用云效私库下载包时，它也会尝试从远程仓库拉取包，并且缓存在云效私库。

远程仓库地址为您的私库地址，这个地址必须是公网可以访问的。如果该私库可以匿名访问，那么无需配置访问账号和密码。点击关联仓库可以保存配置。您随后可以修改配置或者解除关联。注意只有已经缓存在云效私库的制品包才能被搜索到。

测试管理

测试用例与测试计划

RDC提供测试用例和测试计划的功能，用于帮助开发者管理和执行手工用例，针对现在测试更加轻量快捷的特点，提供了以下功能：

测试用例用于管理和组织手工用例，支持方便快捷编辑和查看用例。

测试计划于规划和执行手工用例。测试计划支持任务流概念，可以方便进行测试的评审。

- 发现方便创建和关联缺陷，并提供全面的测试报告分析

一. 测试用例管理

用例集

用例集是组织用例的方式，用于对用例进行分组。用例集支持嵌套用例集。

用例

名称：用例的一个简短描述。限定在100字以内。如果名称描述不清楚用例，请在描述中继续写。

创建人：创建测试用例的人。克隆别人的用例不会更改作者。当然，作者是可以更改的，使用批量修用例信息，功能可以实现修改作者。

步骤：用例的具体操作步骤。

注释：对用例的补充说明。

优先级：用于标识用例执行的优先程度，可选值：P0，P1，P2，P3，默认P3。

用例的导入和导出

Excel导出

1. 选择某个测试集。选中后，将会导出该测试集下（包含子测试集）所有用例。
2. 点击Excel导出。

脑图（mm、xmind)导出

1. 选择某个测试集。选中后，将会导出该测试集下（包含子测试集）所有用例。
2. 点击MM导出。

Excel导入

导入模板如下：

编号	名称 必填	产品名	测试集	作者	优先级	标识符	步骤	注释	时间
516	桌面反复拖动分组内图标	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
518	连续点击应用图标启动应用后置于后台	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
520	连续左右滑动屏幕	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
522	连续安装并卸载应用	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
524	连续点击应用图标启动应用并退出	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
526	反复调用Widget列表	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26
528	桌面反复拖动图标操作	OS事业群/BugFree	桌面 for CB 2.9	果薇	P2				2015-11-26

脑图（mm，xmind导入

导入全面兼容mm，xmind格式。格式说明见：

默认将叶子节点作为用例节点，其他节点作为测试集

> 如果要将指定的节点作为用例节点则在用例名称之前添加关键字：TC或tc。此时该节点会被识别成用例节点，该节点的直接子节点被识别成步骤。再往后的子孙节点会被忽略。

> 如果需要添加优先级，则用例名称为tc:pn_casename 其中“n”为优先级（0，1，2，3）

复制、移动，引用已有用例

复制用例集

用例支持在用例集列表上进行用例集的复制，右键点击测试集，支持测试集复制。

从其他项目引用用例

右键点击测试集，选择引用。

移动测试集

支持通过拖拽测试集实现测试集的移动。

回收站

用例支持回收站，删除的用例会放在回收站中，可以在回收站对删除的用例进行拖拽恢复。

用例标签

【待补充】

二. 测试计划

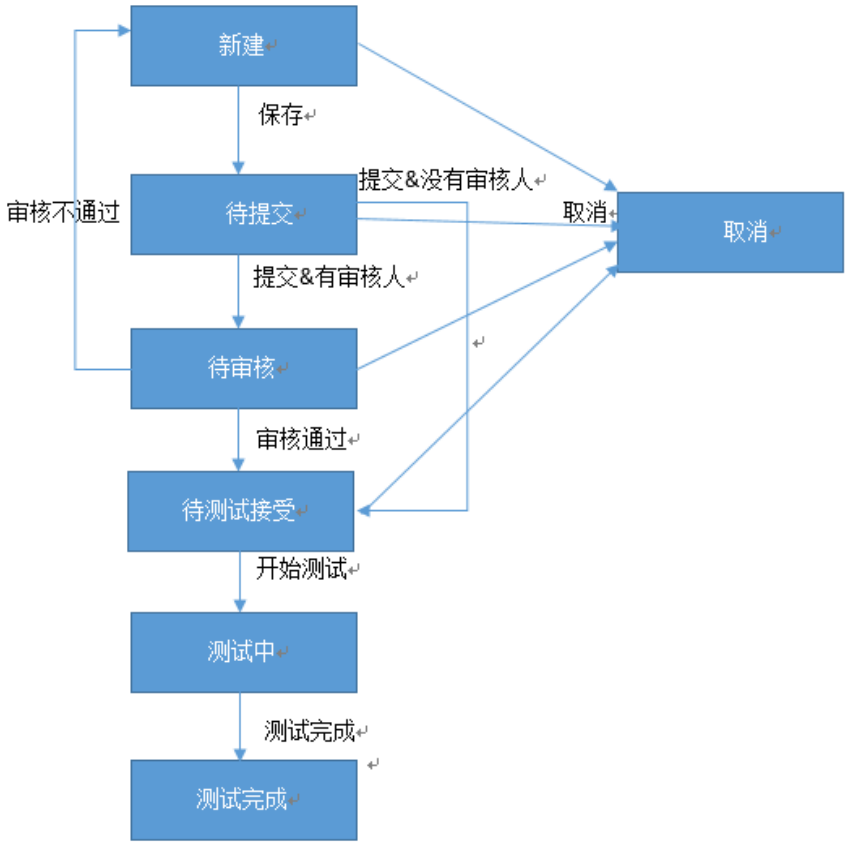
用例加入测试计划才能执行，测试计划是用来规划一次测试过程的载体。

新建测试计划

新建测试计划，可以将这个测试计划需要执行的用例加入计划，用于标识这个测试计划需要执行的内容。

测试计划流程

测试计划流程如下图：



一些规则

1. 计划所属的项目成员和指派人可执行和修改计划
2. 计划在审核中，不能进行修改

执行用例关联缺陷

用例执行时，如果发现缺陷，可以直接关联或者新建缺陷。

测试结果查看

如图，测试结果会展现测试进度，测试通过率和测试状态。

测试报告

三. 使用技巧Q&A

误删除用例如何恢复

所有删除的用例都是逻辑删除，放在了回收站内，可在回收站中恢复。

用例如何过滤？

如图 

测试计划中编辑用例会在全局生效吗？是否还要同步？

是的，全局生效，不需要同步。

自动化测试与持续集成

实验室

实验室是RDC提供的自动化测试和持续集成托管服务。在公测期间，实验室首先提供自动化测试服务，支持 阿里云Code 托管的代码项目(如果是Private的代码项目，需要有相应的访问权限)。

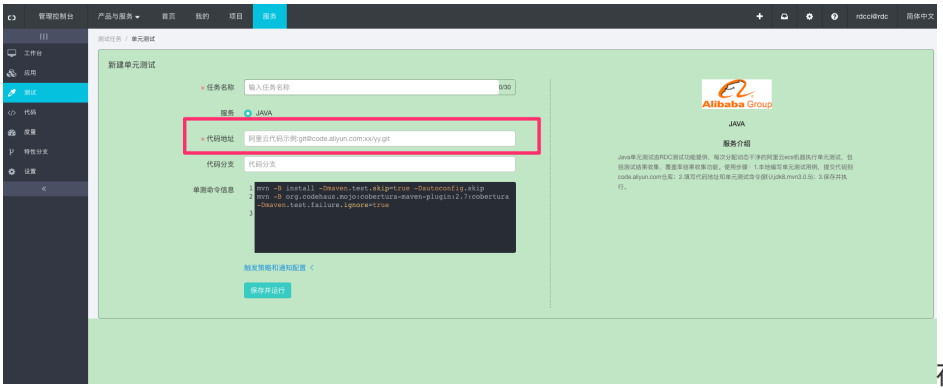
启用测试服务

在 项目 目录的 设置 页面里，切换到 服务 模块，在 测试 图标上点击 添加 。

测试 会出现在左侧的菜单栏里，这样就完成了 启用测试服务 。点击就可以进入到测试服务页面。

新建测试服务

在 测试 页面，在 快速开始测试 里选择要创建的测试服务 （我们以单元测试为例说明）



在表单里一共有四项内容：代码地址，分支，单测命令，名称。填入对应的值或者使用默认指令，保存并运行， 页面会跳转到运行页面。

新建实验室

在 测试 页面，在 快速开始测试 里选择实验室

其中需要将 构建配置文件 放置在代码根目录下，点击 新建配置文件 进入到 构建配置编辑 页面，点击右上角

可以 切换到源码模式

一个典型的构建配置如下：

```
pipeline:
- 单元测试
stage:
单元测试:
env:
cluster: rdc
plugin:
-
param:
url: '${repo}'
branch: '${branch}'
path: '${source_root}'
name: checkout
pos: front
-
param:
path: '${source_root}'
buildId: '${build_id}'
stageName: '${stage_name}'
pipelineId: '${pipeline_id}'
system: '${_system_okt} ${_system_mqut}'
startAt: '${start_at}'
name: case_result_parser
pos: back
-
param:
source_path: '${source_root}'
source_old_path: '${source_root}/../sourceold'
type: java
diffcoverage: open
coverage: open
pipelineId: '${pipeline_id}'
system: '${_system_okt} ${_system_mqut}'
startAt: '${start_at}'
name: java_coverage_collector
pos: back
exec:
- 'mvn -B install -Dmaven.test.skip=true -Dautoconfig.skip'
- 'mvn -B org.codehaus.mojo:cobertura-maven-plugin:2.7:cobertura -Dmaven.test.failure.ignore=true'
```

stage 执行阶段，一个构建可以包含多个阶段。
env 执行环境
cluster 集群环境，目前提供公共集群 rdc, 一次构建只能使用一个环境
load 复用环境
exec 执行脚本
plugin 插件
pipeline 执行顺序，数组类型。数组里的每一项对应阶段键名。按数组顺序依次串行执行。

编辑完成后，点击下载。将下载下来的 .rdcci.yml 文件提交到代码项目里。填写完表单和上述操作后，点击新建按钮就可以看到构建的进行情况了。

指定执行机器

默认使用的执行机器是RDC提供的动态机器，对应的YAML中通过如下参数：

```
env:
cluster: rdc
```

如果需要使用自有机器执行持续集成，需要1）安装Staragent

获取安装脚本wget http://cise-rdc-staragent.oss-cn-zhangjiakou.aliyuncs.com/staragent/install.sh

安装staragent

sh install.sh

- 启动

sudo /etc/init.d/staragentctl start

- 停止

sudo /etc/init.d/staragentctl stop

- 重启

sudo /etc/init.d/staragentctl restart

- 状态

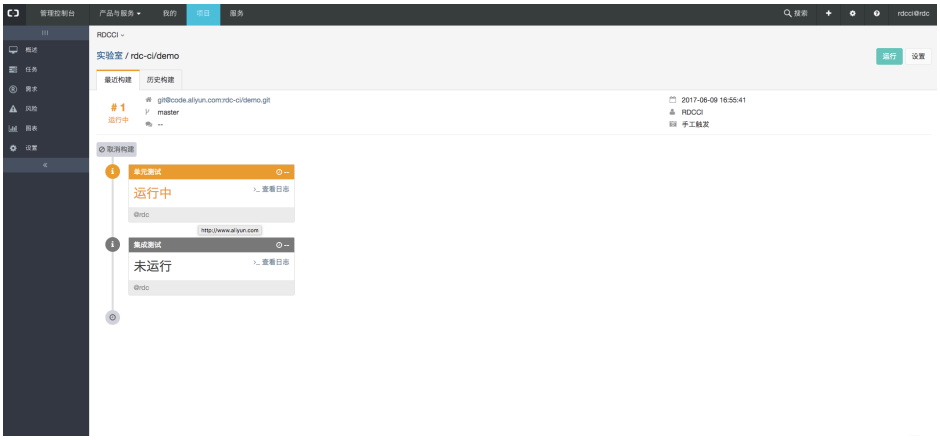
sudo /home/staragent/bin/staragent2 -e GetConfig

2）在YAML中指定有机器IP进行执行

```
env:
host: xx.xx.xx.xx
user: root
```

运行实验室

新建完实验室就会直接运行了。点击右上角的 运行 按钮可以再次构建了。



阿里巴巴代码规约检测

《阿里巴巴 Java 开发手册》是阿里巴巴集团技术团队的集体智慧结晶和经验总结，经历了多次大规模一线实战的检验及不断的完善，系统化地整理成册，反馈给广大开发者. 阿里巴巴 Java 开发手册检测的能力也被集成在RDC的自动化测试服务中，可以直接对代码进行扫描以检测室是否符合阿里巴巴代码规约。

代码扫描支持 - 全量扫描和增量扫描

1. 阿里巴巴代码规约检测全量扫描通过RDC对Java代码工程进行编码规约全量检测优点： 支持跨文件引用， 代码扫描全面缺点： 但扫描速度较慢，问题量会比较多
2. 阿里巴巴代码规约检测增量扫描是基于代码的一次push，自动获取diff内容，对diff文件用编码规约规则进行扫描，并过滤出此次提交产生diff规约问题功能。（目前增量扫描功能即将上线，尽情期待）优点： 只扫描diff文件，扫描速度很快，增量问题直接关联到人，能有效防止代码提交引进新问题数。缺点： 因只扫描diff文件，不能发现跨文件引用出现的规约问题。

怎么使用RDC进行阿里巴巴代码规约检测

1. 通过”测试”服务中“阿里巴巴代码规约检测”创建扫描任务

新建代码规约扫描，填入要扫描的地址，开启代码扫描

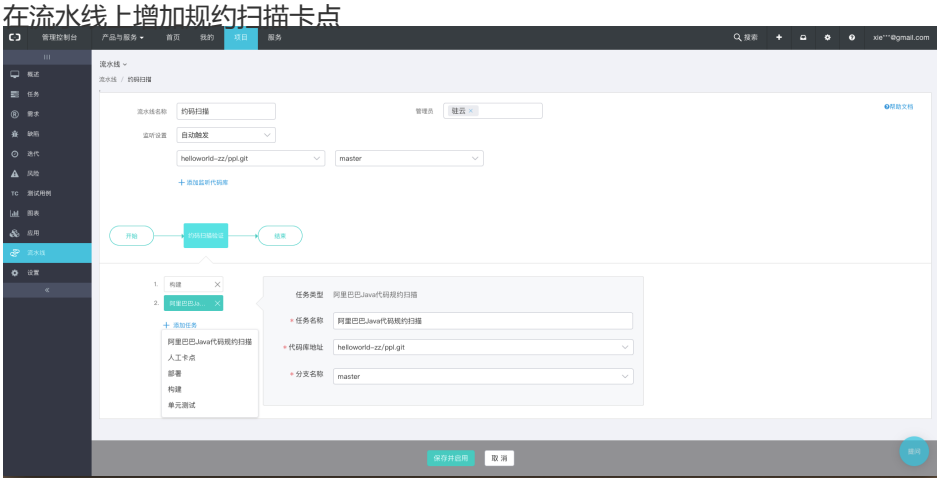
2. 配置代码提交或者定时触发“阿里巴巴代码规约检测”

功能待上线

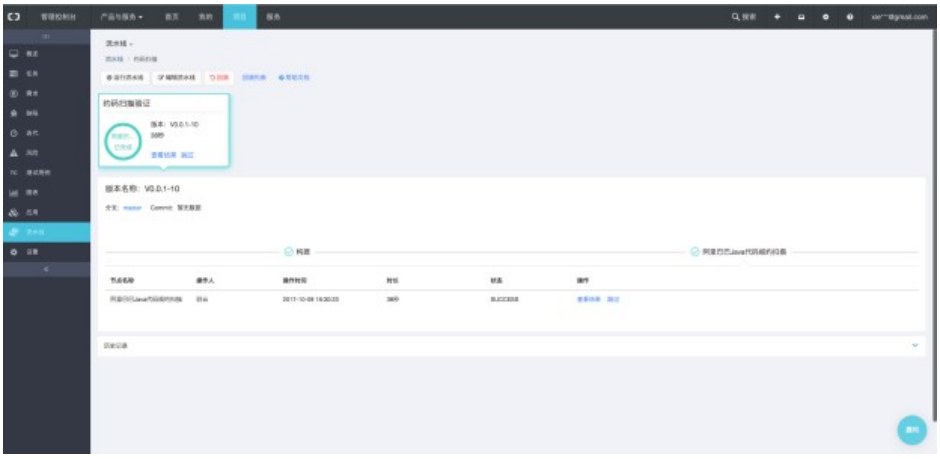
3. 发布时进行代码规约扫描并进行发布卡点

RDC支持在发布流水线上配置阿里巴巴代码规约扫描（支持全量和增量扫描），可以将阿里巴巴代码规约扫描组件加入流水线， 并设定相应的通过条件，就可以支持代码规约扫描卡点。

在流水线上增加规约扫描卡点



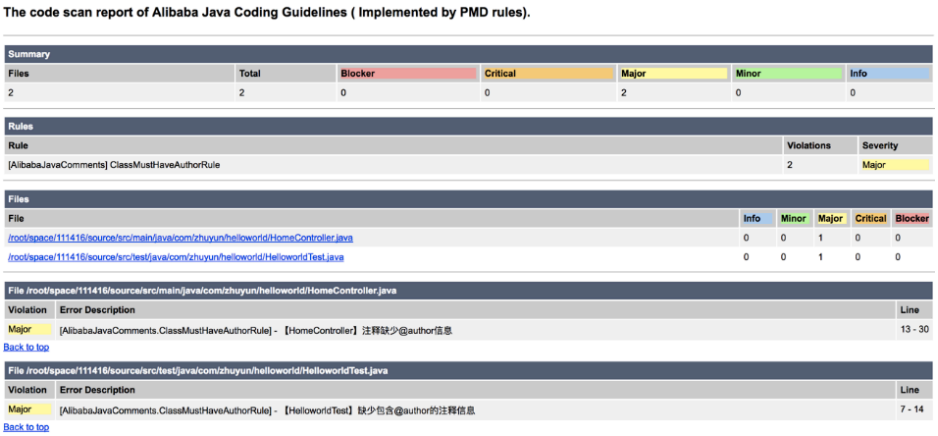
规约扫描结果在流水线上展现



查看扫描结果

执行完代码扫描任务后，可以看到相应的执行结果。

点击问题数可以查看具体问题以及相应解决方法。



The code scan report of Alibaba Java Coding Guidelines (Implemented by PMD rules).

Summary

Files	Total	Blocker	Critical	Major	Minor	Info
2	2	0	0	2	0	0

Rules

Rule	Violations	Severity
[AlibabaJavaComments] ClassMustHaveAuthorRule	2	Major

Files

File	Info	Minor	Major	Critical	Blocker
/root/space/111416/source/src/main/java/com/zhuyun/helloworld/HomeController.java	0	0	1	0	0
/root/space/111416/source/src/test/java/com/zhuyun/helloworld/HelloworldTest.java	0	0	1	0	0

File /root/space/111416/source/src/main/java/com/zhuyun/helloworld/HomeController.java

Violation	Error Description	Line
Major	[AlibabaJavaComments.ClassMustHaveAuthorRule] - 【HomeController】注释缺少@author信息	13 - 30

Back to top

File /root/space/111416/source/src/test/java/com/zhuyun/helloworld/HelloworldTest.java

Violation	Error Description	Line
Major	[AlibabaJavaComments.ClassMustHaveAuthorRule] - 【HelloworldTest】缺少包含@author的注释信息	7 - 14

Back to top

无线测试接入RDC发布流程

MQC接入RDC发布流程步骤概述

通过实验室可以在RDC发布流程中，把无线构建打包后自动的触发MQC，完成移动测试的自动验证。配置如下：

1. 在实验室创建无线测试任务
2. 在发布流程增加实验室流程验证点

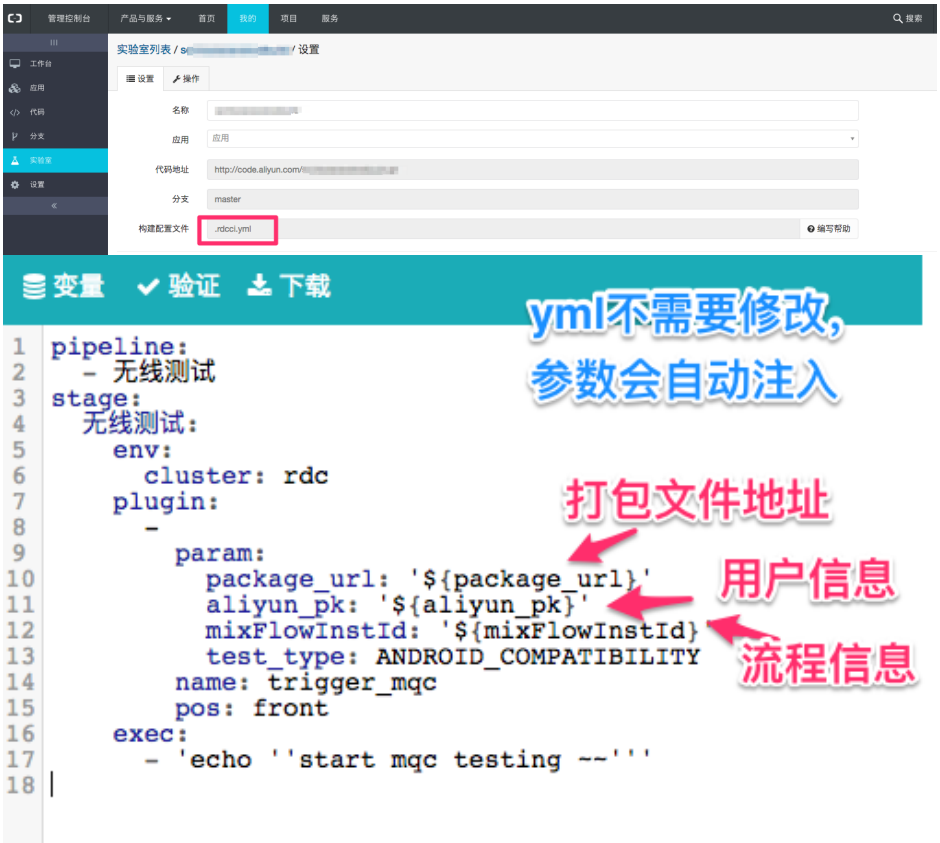
在实验室创建无线测试任务

1. 将一下内容保存到代码地址根目录下.rdcci.yml

```
pipeline:
- 无线测试
stage:
无线测试:
```

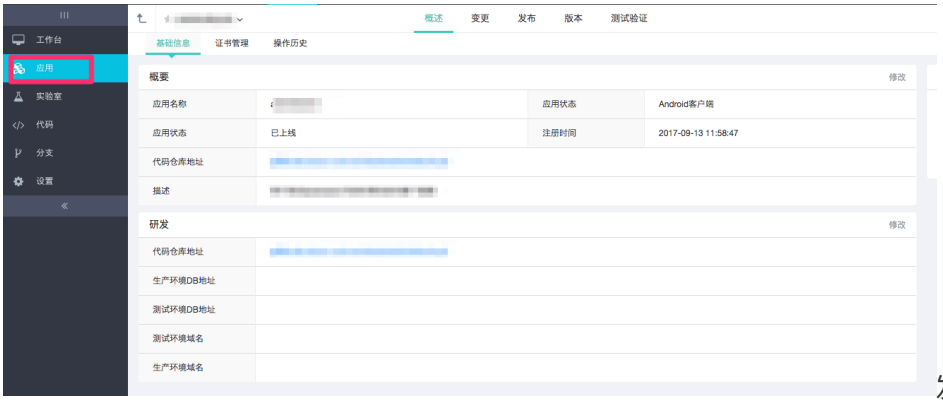
```
env:
cluster: rdc
plugin:
-
param:
package_url: '${package_url}'
aliyun_pk: '${aliyun_pk}'
mixFlowInstId: '${mixFlowInstId}'
test_type: ANDROID_COMPATIBILITY
name: trigger_mqc
pos: front
exec:
- 'echo ''start mqc testing ~~'''
```

2. 在实验室创建对应代码地址任务



在发布流程增加实验室流程验证点

1. 准备好你的应用和发布流程



发布流程

1. 增加发布流程的测试验证

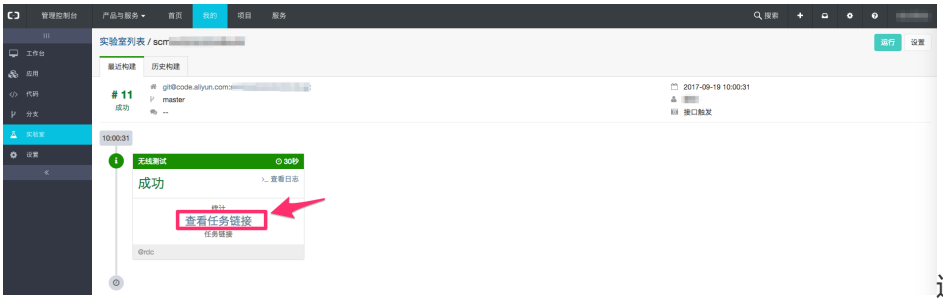
1. 配置对应实验室任务

到这来我们的配置就完成了，下面可以看一下实际运行的效果

测试验证效果一览

1. 发布流程的无线构建完成， 点击进入无线测试

1. 这里可以看到测试任务运行结果， 通过连接可以看到实验室的任务详情



这里可以连接到mqc执行详情页查看移动测试结果详情

附：MQC 无线测试

移动测试（MQC）是为广大企业客户和移动开发者提供真机测试服务的云平台

http://mqc.console.aliyun.com/

持续交付和质量红线

持续交付和质量红线

自动化测试保障持续交付质量

RDC提供了完备的Pipeline, 在整个研发过程开发代码提交后自动触发单元测试, 静态代码扫描。应用发布打包,部署, 自动触发集成测试, 构成了开发和测试共同参与的一套流水线. 在持续交付的实践中, 这样的做法可以有效的加快开发测试效率, 以最小的成本, 找到代码中的错误, 保持代码的质量平稳, 发布周期可预。

RDC持续交付提供质量验证卡点

验证卡点是用于保障交互质量的重要手段，为了达到持续交付的目标，我们建议通过分层测试和测试卡点通过才继续流转的方式来保障整个持续交付的顺利进行

1）代码提交后自动运行相应的实验室：系统自动监控代码提交事件，分析代码的变更情况（变化的Java文件，类，方法），自动执行测试任务。为保障开发分支的质量，这个阶段我们推荐用户配置单元测试和静态扫描。

2）在应用部署后，会根据不同的发布阶段分别流转日常阶段、预发阶段、线上阶段的测试任务执行，用于验证相应阶段的质量。

- 日常测试阶段：将分支合并主干后部署到日常环境，关联的实验室任务可以按照自己的需求进行配置；
- 预发测试阶段：将日常部署分支部署到预发环境，关联的实验室任务可以按照自己的需求进行配置；
- SIT集成测试验证：将预发部署分支部署到线上环境，关联的实验室任务可以按照自己的需求进行配置；
- 线上集成验证：将预发部署分支部署到线上环境，关联的实验室任务可以按照自己的需求进行配置；

通过实验室创建自动化测试任务

RDC是通过实验室来实现自动化测试任务，如何使用请查看“自动测试和集成”

在Pipeline上配置验证卡点

应用的测试验证配置页面配置Pipeline的测试任务和通过条件

1. 点击“添加验证点”，添加每一验证阶段关联的实验室任务。
2. 通过条件就代表对应阶段运行的标准, 默认通过条件是任务成功，但也可以添加单测成功率, 覆盖率, 静态扫描缺陷等条件来控制是否通过。

双引擎自动回归服务

双引擎回归测试平台介绍

什么是双引擎平台

概述

双引擎自动回归平台（简称双引擎或者doom）是一个将线上真实流量复制并用于自动回归测试的平台。 通过创新的自动mock机制不仅支持读接口的回归验证，同时支持了写接口（例如用户下单接口、付款接口）的回归验证。 基于容器隔离机制以及完善的异常处理和监控机制，确保对应用本身的侵入非常小。 通过它，不仅能够实现低成本的日常自动化回归，同时通过它提供扩展能力可以支持系统重构升级的自动回归。 天猫、淘宝核心交易链路系统通过它实现低成本高覆盖率的日常自动化测试回归，同时内部的一些重大重构项目通过它保证系统的无故障升级。 它是复杂的难以通过传统方式做测试回归的业务系统以及金融类对故障十分敏感的系统的稳定性利器。

大致原理图如下

它与tcpcopy或者diffy的区别：tcpcopy、diffy是在应用外的网络层实现流量录制和回放的，它们只能实现一些只读页面的验证，而且无法实现跨环境的流量回放。双引擎是在应用内部通过aop切面编程方式实现的流量录制和回放功能，因此可以做到应用内部接口级别的回归验证，当然也支持服务级别或者http级别的回归验证。而通过独创的中间件级mock以及内部自定义的mock，可实现写流量的回归验证以及跨环境的回归验证（线上引流到测试环境）。

双引擎是一个封闭的系统吗？答案是否定的，我们希望把它打造成一个开放的平台，系统聚焦录制、回放、比对的核心能力。而数据的存储、数据的传输、中间件的扩展以及回放流程的都可以自定义扩展，使用者可以基于这些基础能力搭建适合自己业务的自动化回归平台。

目前双引擎在阿里巴巴集团内部被广泛使用，成为交易核心重构升级必不可少的利器。同时基于它扩展的‘天启’平台成为交易日常自动化回归的主要工具。

接入使用文档

应用场景

- 系统重构时，复制真实线上环境流量到被测试环境进行回归，相当于在不影响业务的情况下提前上线检测系统潜在的问题。
- 可以将录制的流量作为用例管理起来进行自动化归回。

优势

- 低成本：无需编写测试用例，通过流量录制形成丰富的测试用例。
- 高覆盖：一方面线上大量真实流量确保覆盖率，另一方面支持中间过程的验证，例如发送消息的内容、中间计算过程等等的全对象的对比验证，传统手工编写验证点很难实现。
- 支持写流量验证：（注：写流量是指可能导致有数据变更的流量）不用担心写流量回放污染应用数据，支持线上引流到测试环境以及写流量的自动化mock。
- 低应用侵入：通过隔离容器技术、字节码级别的AOP技术、中间件级MOCK避免接入类冲突以及降低接入成本。

录制回放的原理是什么

概念

主调用：待验证的流量入口，可以是http请求、也可以是一个java调用（公测版目前暂时只支持java调用）。
子调用：主调用执行流程中的一些方法调用，这些方法调用入参返回值会被记录下来用于回放时再次调用到该方法是时进行mock。

读接入模式：主调用是读接口，所有对外请求可以不进行mock，也就是不存在子调用的概念，这样的好处是可

以验证到整个请求链路。

写接入模式：如果对外有写请求，那么需要通过配置中间件隔离将对外请求作为子调用进行mock。

原理解析

要实现引入线上流量来做回归验证牵涉到几个问题：流量如何复制？如何保证录制不影响业务？如何回放？如何解决写流量回放对业务的影响？如判断别被测系统bug？

流量如何复制?

java方法调用可以通过方法aop实现主调用以及子调用的入参和返回值的录制；http流量可以通过设置httpfilter或者对特定的servlet容器进行aop实现对http请求参数录制。当请求结束会启用异步线程池将录制数据序列化后发送给回放机器。

如何保证不影响业务？

针对用于生产环境的录制机器，任何流量录制异常不影响业务流程正常执行。同时在系统负载高时具备主动停止流量录制的能力。 针对用于生产环境的回放机器（beta机器），客户端可以支持指定rpc中间件不会真正对外提供服务，支持指定的消息中间件不去发送和订阅消息，同时支持指定的数据库框架不去真正执行写数据到数据库。如果是web服务器需要接入方摘除回放机器的vip。

如何解决写流量回放对业务的影响？

当回放后，可能会执行一些写逻辑，比如写分布式缓存、写db等等。而通过子调用mock的机制可以确保写不会真正发生，因此不会影响业务数据。

如判断别被测系统bug？

对于读接口，我们主要关注在相同请求下正常系统和待测系统的返回结果的差异，读接口也提倡对所有对外请求进行mock，这样回放时能保持当时的一个现场环境，保证验证的准确性。

对于写接口, 只验证接口返回结果是不够的，需要验证它具体写入的数据是否正确。例如创建订单会调用tp的写订单接口，那么我们需要验证回放时调用tp的参数和录制时调用tp的参数是否一致。

对比默认是全对象字段逐一对比，如果存在必然不一致字段，例如时间，系统ip等等可以配置忽略该字段的对比。默认开启子调用对比后所有子调都会对比，也可以配置排除一些不关心的自调用的对比。

线上流量可以到线下回放吗？

答案是可以的，可能大家会疑问，线下库和线上都不一样，怎么跑的通？实际上如果所有对外请求都做了mock，那么线下回访并不会真正发起对外调用，都用线上数据mock，所以可以跑通。利用这点我们可以在项目测试环境回放线上执行的过程，帮助问题复现排查。

哪些调用会成为子调用？

平台有中间件隔离配置，例如配置了hsf隔离，那么所有hsf对外调用都会作为子调用。类似的还包括tair、

notify等等。另外也支持自定义一个类方法作为子调用。当然我们也提供白名单配置指定某些调用不进行mock，这样可以验证到下游的读逻辑或者解决无法mock的问题。

如何解决回归对比噪音

问题

将部署了稳定代码的服务器作为流量采集源，对于待正式发布的代码，可以用录制的流量来进行回放和差异比对，以便于发现待测系统的问题。那么录制环境和回放环境所处环境不同，有一些必然不一致的信息，例如服务器ip、时间、以及一些随机数等等。怎么去处理这些差异呢？

方案

- 排除法：平台支持指定字段排除对比，将不需要的字段排除即可。
- 指定对比法：将关心的业务数据进对比。

使用限制是什么

- 平台目前仅支持基于spring框架的java应用，非spring框架java应用需要做额外定制扩展。
- 目前支持java调用的录制回放、http流量的在集团内部版本支持，云版本后续会支持。
- 支持的隔离中间件以配置平台为准，如不支持请联系相关技术支持同学。

使用双引擎对应用有哪些影响

- 数据录制有一定的系统资源消耗，具体视应用而定，一般都在可接受范围以内。
- 运行客户端大概有100~300M的额外内存消耗，请规划好应用的内存。

双引擎接入使用文档

原理介绍

平台介绍

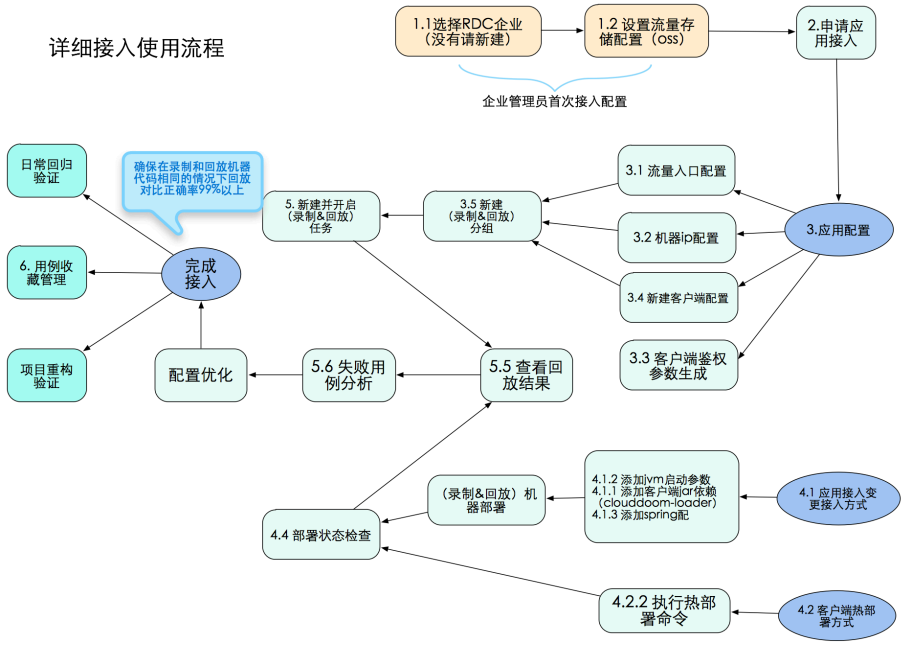
适用范围

推荐云效上的java应用直接使用。也支持非云效上的其他任何java应用使用，前提是您的应用服务器可以正常

访问域名：doom.rdc.aliyun.com 。如果因网络问题无法使用可联系我们进一步沟通解决方案。

联系方式（email）： mufeng.qcg@alibaba-inc.com

详细接入流程图



平台入口

入口链接

1. 企业设置

1.1 新建或加入企业

访问 云效并根据提示新建企业，如果已经存在企业请忽略。您也可以接收企业管理员要邀请加入企业。

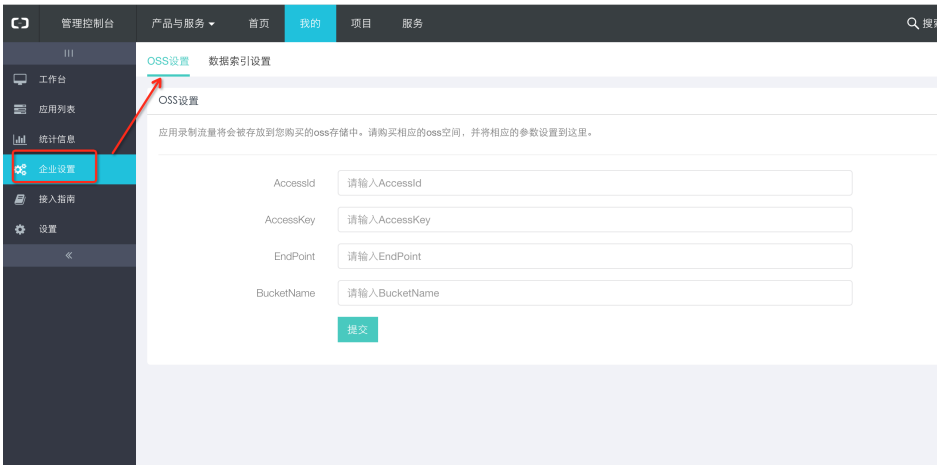
1.2 数据存储设置

OSS配置

录制流量将保存到oss中，因此企业使用前需要先进行购买和设置。请预先评估好需要容量，oss 申请地址

警告：若流量不够请申请扩容，如果更换oss将导致原录制的数据无法使用。

设置方法：



2. 应用接入申请

rdc中的应用需要申请后才能正常接入使用，目前公测需要联系相关技术支持同学开通使用，一个企业限制接入5个应用。

2.1 申请方法

2.2 审批结果

目前公测阶段由产品后台方审批，审批完成后会通过用户在rdc绑定的邮箱发送审核结果。如果在3个工作日后还未收到通知结果，请发邮箱咨询我们。 联系人： 穆风 mufeng@aibaba-inc.com

公测结束后，会把权限下放到企业管理员。

3. 管理平台接入配置

3.1 第一步：流量入口（主调用）配置

在申请审批通过后进入该应用，并选择 【应用配置】-> 【流量入口】 点击 【新增】

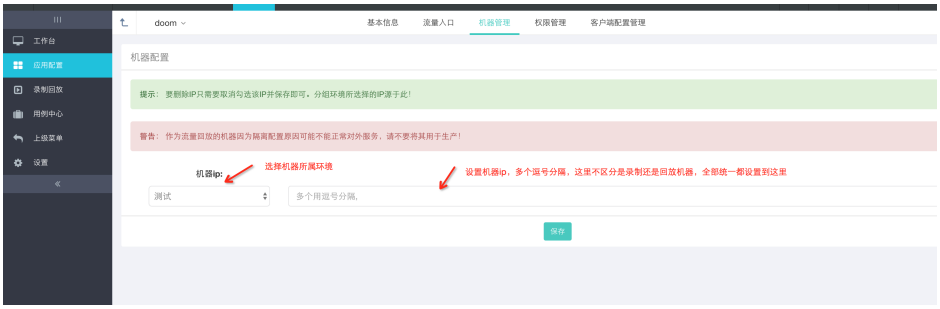
注意事项

1. 流量入口其实就是所说的主调用，请确保设置的实现全类名为具体实现类，非接口。
2. 流量入口如果存在在流量A中，调用入口流程B的代码链路，可能会存在采集数据的问题引起回放的不准确，请设置流量入口时应尽量规避。

然后设置相关流量入口并保存。（目前版本暂时只支持java入口，http流量后续开放）

3.2 第二步：设置待接入机器IP

一般生产机器找一两台机器来作为流量录制机器，另外找一两台测试机器作为回放机器。环境类型包括 测试、预发、线上。



测试流程通常是从日常->预发（灰度）->线上可以支持线上采集的流量跨环境回放，取决于你回放任务设置的数据采集源的环境。

3.3 第三步：生成客户端访问参数

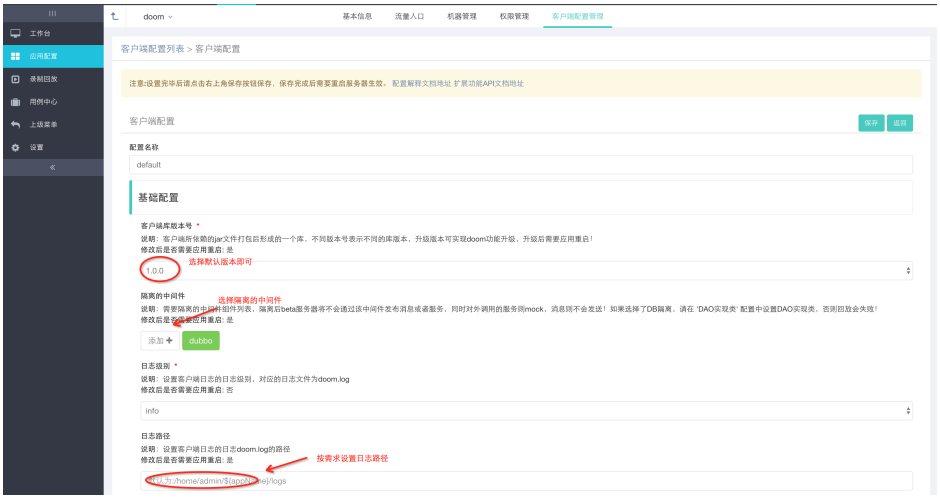
该参数是后续应用变更时需要配置的参数，生成即可

3.4 第四步：生成客户端配置

客户端配置是埋点客户端运行需要的一些参数配置，一般生成一个默认的即可。

1）新建客户端配置

2）给客户端配置命名&选择隔离的中间件一般应用使用到了哪些中间件就把这些中间件配置上即可，然后保存。如果应用使用的中间件平台选择不了请联系技术支持同学。



- 客户端隔离中间目前支持 dubbo,redis,mybatis，如果你有其他中间件，请先联系我们。我们根据评估为你们应用做适配。
- 作为隔离环境，服务端提供者将不再提供服务，例如dubbo隔离后，dubbo provider不会注册服务到注册中心。原因是回放因为隔离了环境，提供服务明显程序运行不完整，会产生程序错误等。
- 还需要了解高级配置请到用户手册客户端配置查阅。

3.5 第四步：流量分组配置（也叫分组环境）

3.5.1 什么是流量分组

流量分组（分组环境）是针对流量录制和回放两种场景设置的相关管理机制，分组会管理相应的服务器ip、流量入口配置以及相关客户端配置信息。利用流量分组可以实现流量的分配管理以及多项目协作管理。

3.5.2 流量分组类型

流量分组有3种类型，录制分组、回放分组、录制&回放 分组。

录制分组

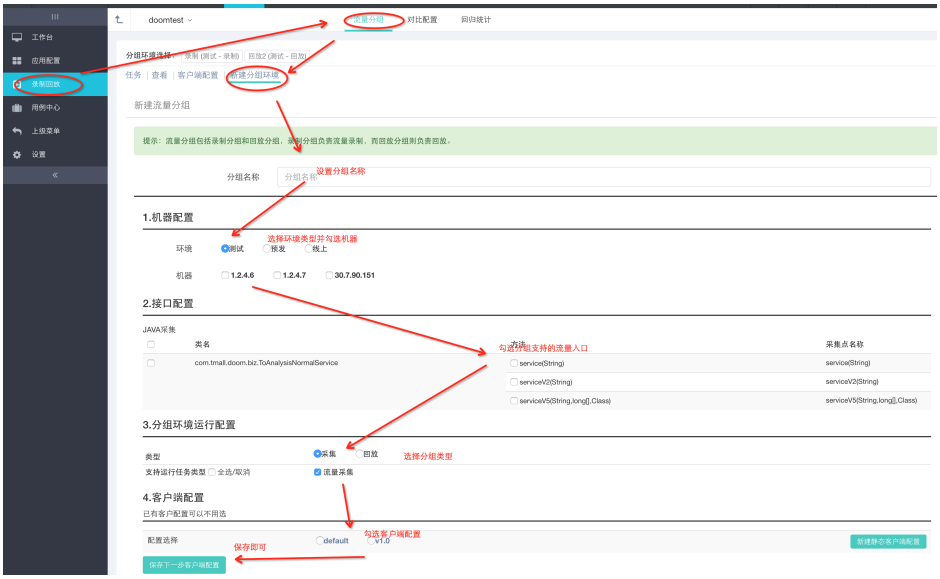
仅负责录制数据,对应用的影响和侵入最小。

回放分组

仅负责数据回放,会对应用的对外请求进行隔离，也就是它不能直接对外提供服务，线上或者beta回放请选择此分组。

‘录制&回放’ 分组

既支持录制也支持回放，也支持一台机器同时录制和回放(便于接入调试)。不会对应用的对外请求隔离，请谨慎用于线上或者预发回放。



4. 应用接入变更

4.1 常规接入模式

4.1.1 sdk下载

sdk.zip

将sdk下载并解压后得到 doom-common-client-xxx.jar,clouddoom-loader.jar 两个文件。

doom-common-client-xxx.jar 添加到应用依赖中。（可以将doom-common-client-xxx.jar 文件作为三方库上传到自己的私用maven库，这样就可以直接依赖了）

4.1.2 JVM配置项添加

1）将 clouddoom-loader.jar 放到服务器中某个目录例如 /home/admin/doom 中

2）新增jvm启动项：

// ‘appId:1#clientKey:KsvU...#endPoint:http://doom.rdc.aliyun.com/’ 此参数可以从管理后台的【应用配置】-> 【权限管理】的ClientParam参数中获取。

-javaagent:/home/admin/doom/clouddoom-loader/target/clouddoom-loader.jar=appId:1#clientKey:KsvU...#endPoint:http://doom.rdc.aliyun.com/

4.1.3 系统初始化BEAN配置

在应用spring配置文件中新增一个配置

```
<bean class="com.alibaba.doom.client.ClientInitialBean" lazy-init="false"/>
```

针对spring boot应用没有xml配置可以如下配置

```
import com.alibaba.doom.client.ClientInitialBean;


@SpringBootApplication
public class DemoApplication {

@Bean
public ClientInitialBean getDoomClient() {
return new ClientInitialBean();
}

public static void main(String[] args) {
SpringApplication.run(DemoApplication.class, args);
}
}
```

4.1.4 配置检查

当应用变更完成后执行部署，可以在平台的应用基本信息观察到客户端启动状态，判断变更是否成功。

4.2 免应用变更接入模式

4.2.1 目的

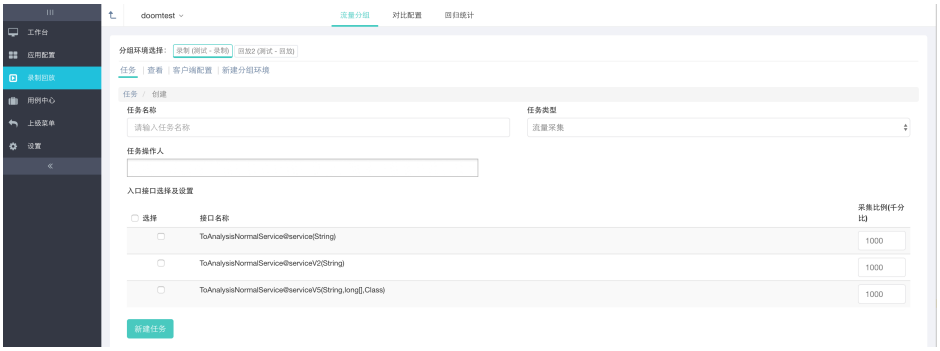
为进一步降低doom接入使用成本，提特定场景下快速接入使用doom的能力，即应用不需要做任何变更以及重启便能热启动doom客户端，完成流量录制和回放目的。（线上环境建议通过常规接入模式接入。）

4.2.2 方法

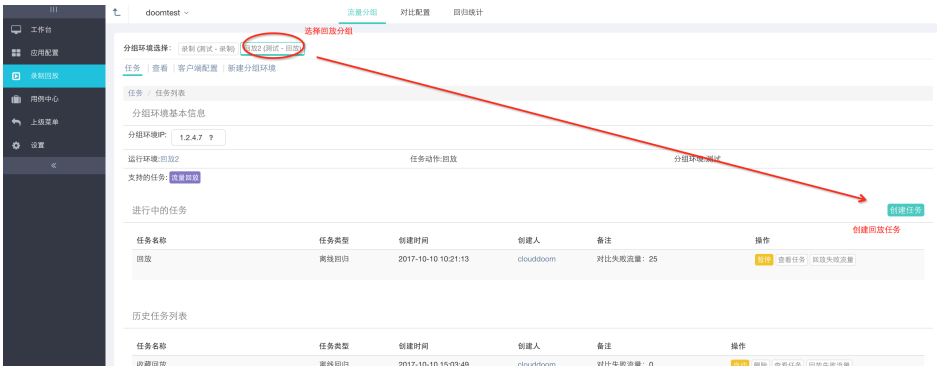
```
#执行客户端热启动命令（注意使用该命令需要联系本平台负责的同学做相关配置才能支持）
#这里的id和key对应:应用配置->权限管理->客户端参数配置 中的 appId和clientKey。
curl -s "https://doom.rdc.aliyun.com/client/gate/install_bash.do?id=xxx&key=xxxx" | bash
```

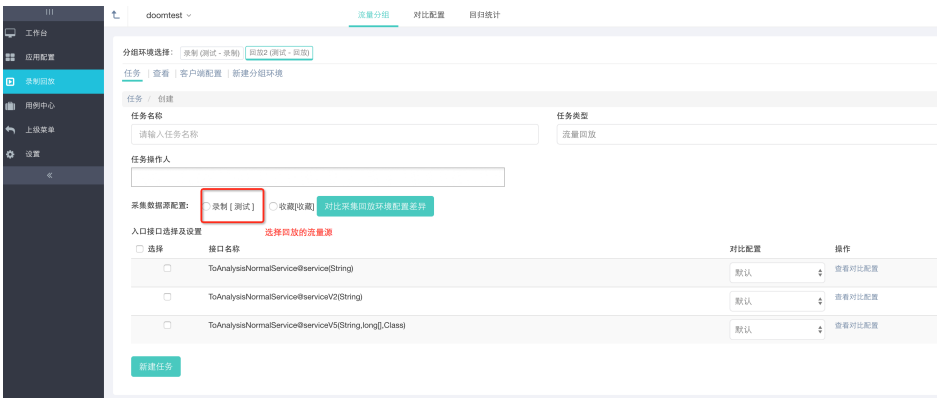
5. 回归任务的创建与回归分析

5.1 创建录制任务



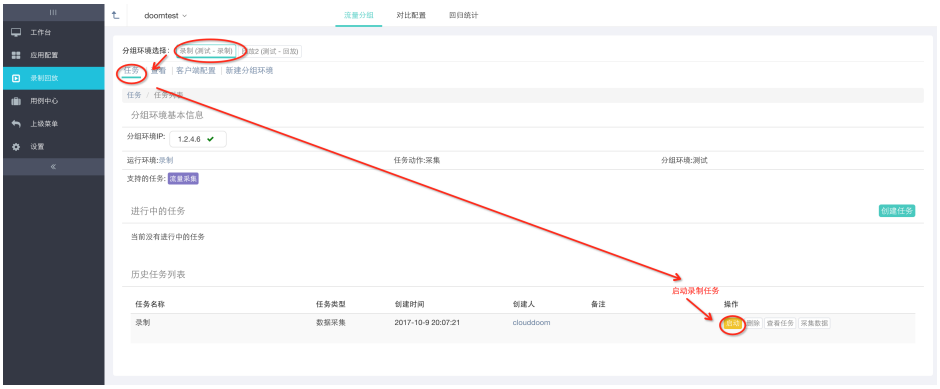
5.2 创建回放任务





5.3 开启任务

开启录制任务



开启回放任务

5.4 触发流量

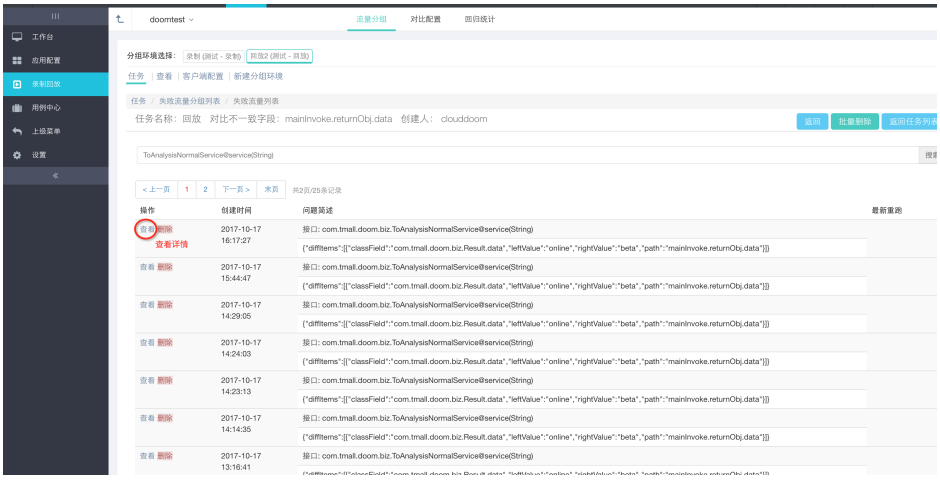
如果是用测试环境，那么请制造些流量来实现录制

5.5 查看流量回放统计

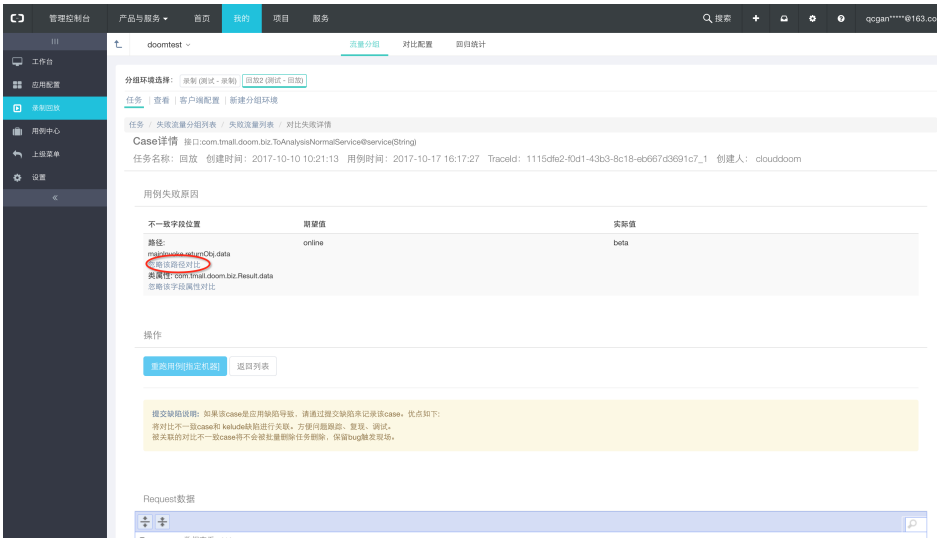
5.6 回放失败分析

进入查看失败原因

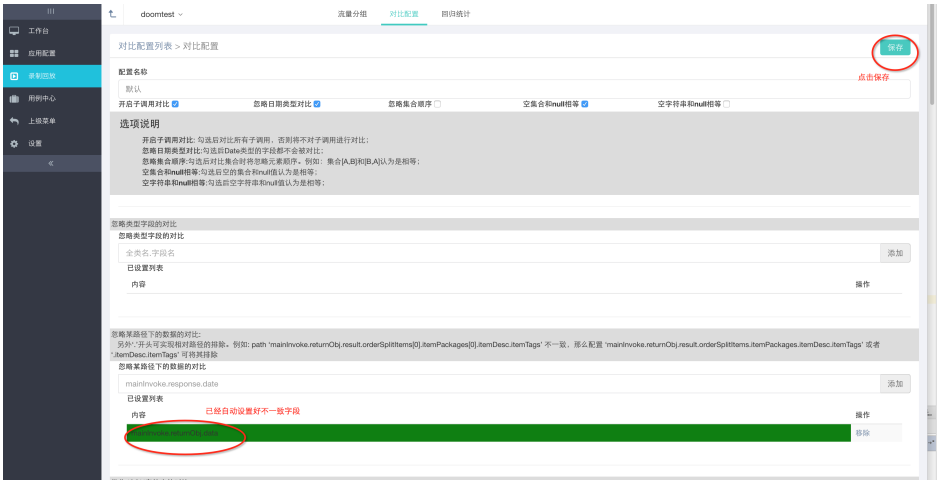
进入查看失败具体列表



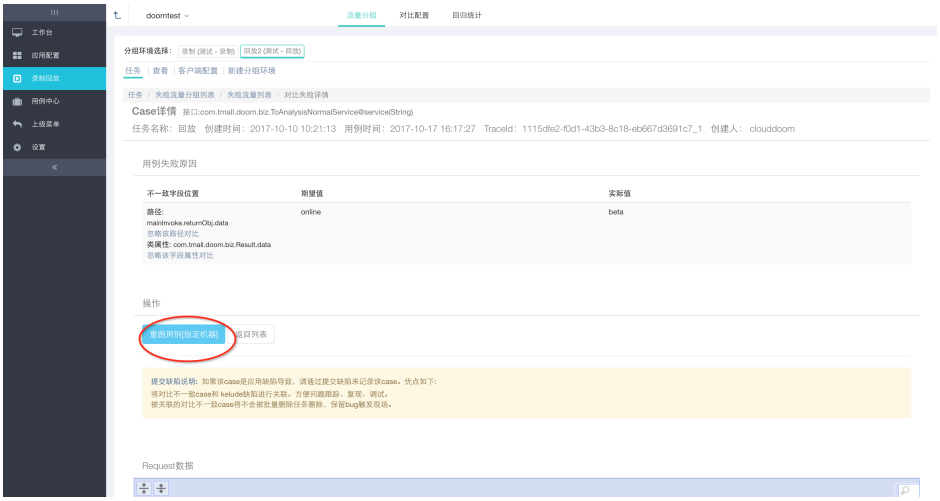
查看具体对比失败原因，查看是否回放失败，如果没有回放失败则查看不一致字段，判断是否为噪音，如果是的话可以点击排除



排除噪音字段后保存



重跑验证（重跑的前提是回放机器客户端要正常运行）



6. 录制流量的检索与收藏（用例管理）

6.1 原理

支持录制流量通过opensearch进行搜索并收藏感兴趣的流量来回放。当opensearch相关配置设置完成后，当客户端录制到流量的入参以及返回结果会通过opensearch进行索引，通过控制台可供搜索和查看。并支持添加到用例集，通过新建回放任务并关联相关用例集实现用例回归。

6.2 步骤

- 开通opensearch&新建应用&设置数据表
- doom平台配置

6.3 开通opensearch&设置数据表

- 进入https://opensearch-cn-beijing.console.aliyun.com/#/apps并点击创建应用
- 选择‘标准版’并立即创建
- 输入应用名例如‘doom_data’进入下一步
- 选择‘手动创建应用结构’进入下一步
- 新建数据表，字段如下

字段名称	类型	主键
id	LITERAL	是
invoke_identity	LITERAL	否
timestamp	INT	否
source_id	LITERAL	否
container_id	INT	否
trace_id	LITERAL	否

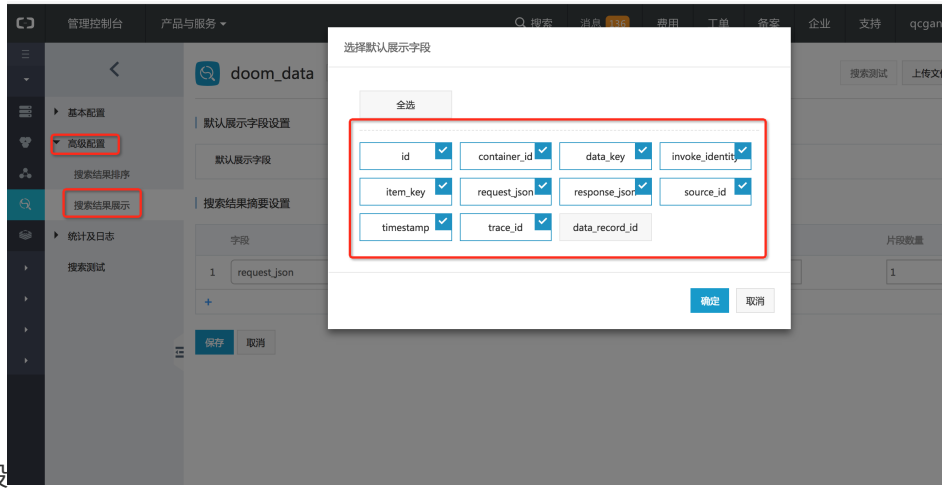
item_key	LITERAL	否
data_key	LITERAL	否
request_json	TEXT	否
response_json	TEXT	否
data_record_id	INT	否

- 字段索引设置

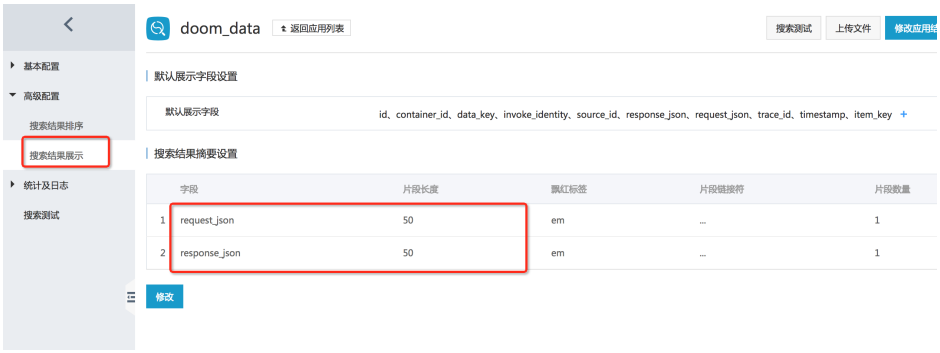
索引名称	包含字段	分词方式
id	id	不分词
default	request_json、response_json	中文 - 基础分词
invoke_identity	invoke_identity	不分词
trace_id	trace_id	不分词
item_key	item_key	不分词
data_key	data_key	不分词
time_stamp	timestamp	不分词
source_id	source_id	不分词
data_record_id	data_record_id	不分词

- 在数据源设置也直接点击完成

设置默认展示字段

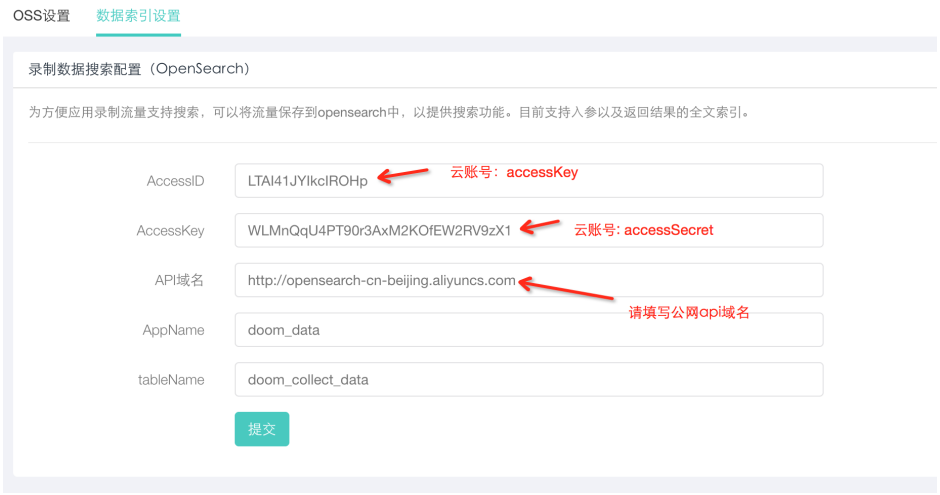


结果摘要设置



- 回到应用列表，激活应用（根据需求合理选择容量大小，也可以申请免费容量试用）

6.4 平台opensearch配置（仅企业管理员可设置）

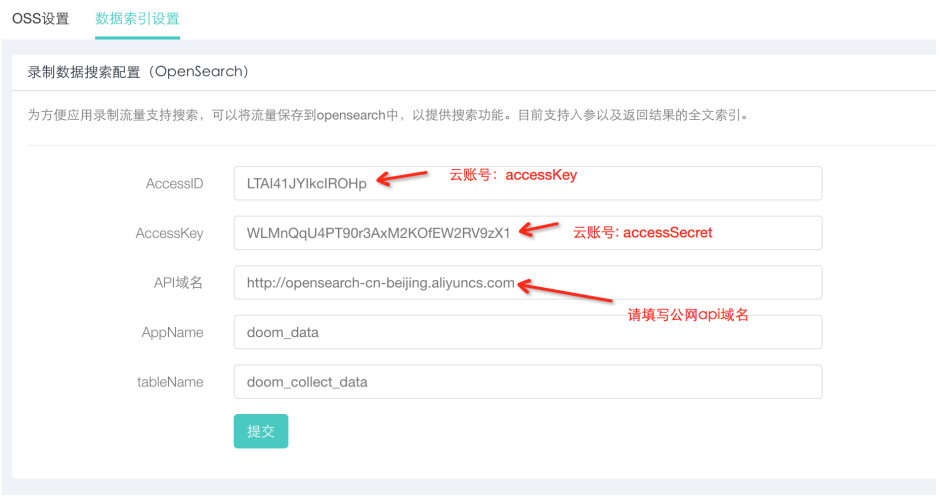


6.5 流量收藏

方法一：进入【用例中心】->【录制流量】选择其中一个流量点击‘查看’，在流量列表页选择‘收藏’方法二：进入【用例中心】->【流量搜索】选择其中一个流量点击‘收藏’。

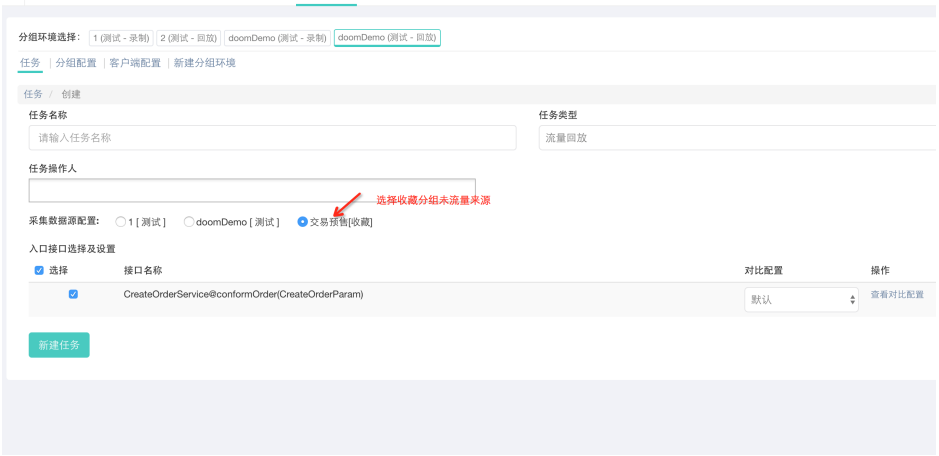
6.6 收藏分组的新建

在点击收藏后显示收藏对话框。点击‘加号进行分组添加’



6.6 新建收藏分组任务并执行回归

在回放分组新建一个回放任务，并关联当前新建的收藏分组即可



测试服务

测试服务

云效测试服务是包含多种自动化测试能力，将测试进行场景分类，除了通用的单元测试，代码扫描，接口测试，也包含了阿里巴巴特有的测试能力如持续集成实验室，流量回归测试。对于测试服务使用者，通过测试服务页面能够快捷的找到所需要的测试服务入口。测试服务传送门

测试服务类型

当前运行提供的测试服务包含：

启用测试服务

测试服务在新建企业中默认启用，启用后，在“我的”->我的导航中显示“测试”菜单，点击后，默认进入测试服务新建页。如果相应的企业中，未显示测试服务，请点击设置

新建测试服务

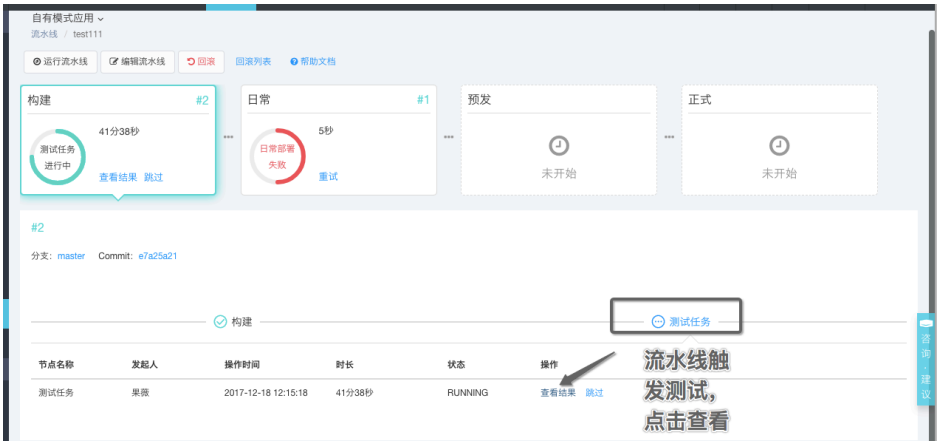
测试服务和流水线

测试服务通过将测试能力进行抽象，提供方便创建表单，提供了通用的持续集成，持续验证，结果通知，触发调度的能力。用户在测试服务中积累的任务,可以和流水线关联，作为持续发布验证的质量关卡。

在流水线中添加测试节点

在云效的自定义流水线中可以新增测试节点。

在发布中查看测试结果



测试集合 - 查看测试日志，重跑，创建缺陷

点击详情链接，可以看到更清楚的测试集合页面。在页面上可以支持查看测试的日志，支持将失败用例重跑，并支持创建缺陷。

测试报告

云效提供默认的测试报告，支持按照时间，项目，应用，团队等各个维度来组织对测试任务进行分析。

研发度量

研发总览

研发总览

功能概述

研发度量是RDC为企业用提供的研发效能度量的功能，通过对在RDC中企业的项目管理，代码活动， 应用发布 ， 测试等研发协同功能中进行数据的同步抽取清洗分析建模，为企业提供研发效能分析和解读。

度量指标定义

- 需求完成数： 统计时间段内，该团队完成的需求数； 这个指标通常用于度量整个团队的交付能力 ，越多代表团队的交付越多。
- 需求平均完成时长： 统计时间段内，该团队完成的需求从创建到终态（正常结束）的平均时长（自然日按天计算）；这个指标是用于度量团队的交付效率，越短代表团队的交付越快越敏捷。
- 新增需求数：统计时间段内，该团队被指派的需求数量；这个指标用于反映整个团队的负载。
- 新增缺陷数：统计时间段内，该团队被指派的缺陷数量；这个指标用于度量团队产生的缺陷数，间接反映质量。
- 缺陷Reopen率： 统计时间段内，该团队缺陷Reopen次数除以解决缺陷数；这个指标用于度量团队缺陷的修复质量，Reopen率越高代表返工越多。
- 缺陷平均修复时长：统计时间段内，该团队关闭的缺陷从创建到Fixed状态的平均时长（自然日按天计算）；这个指标用于度量开发团队缺陷的修复效率，时长越短，代表缺陷修复越快。
- 缺陷平均关闭时长：统计时间段内，该团队关闭的缺陷从Fixed到Closed状态的平均时长（自然日按天计算）；这个指标用于度量测试团队缺陷的验证效率，时长越短，代表缺陷被验证的越快。
- 人均提交代码量——统计时间段内，该团队提交的代码数量/人数；
- 代码规约扫描 ——统计时间段内，该团队相关应用通过规约扫描的Critical Issue， Major Issue数量 ，越低代表代码质量越高。

功能介绍

研发总览

研发总览提供企业研发关键指标和其趋势的分析展现能力

应用质量

应用质量对企业的应用构建发布的质量进行分析

项目进度

项目进度对企业的活跃项目进行进度追踪和效率分析