东方证券DevOps测试报告

| **版本** | **作者** | **日期** | **备注** |
| --- | --- | --- | --- |
| v1.0 | 潘露 | 2019.8.9 |  |
|  |  |  |  |

**目录**

[1 概述 4](#_Toc16259253)

[1.1 测试目的 4](#_Toc16259254)

[1.2 测试计划 4](#_Toc16259255)

[2 软硬环境需求 4](#_Toc16259256)

[3 功能列表 5](#_Toc16259257)

[4 测试结果 6](#_Toc16259258)

[4.1 应用系统概览 6](#_Toc16259259)

[4.2 devops 7](#_Toc16259260)

[4.2.1 环境类型 7](#_Toc16259261)

[4.2.2 Pipeline任务模板 8](#_Toc16259262)

[4.2.3 工作流模板 9](#_Toc16259263)

[4.2.4 配置中心 10](#_Toc16259264)

[4.2.5 制品库 10](#_Toc16259265)

[4.2.6 应用列表与租户 11](#_Toc16259266)

[4.2.7 自定义图标Logo 12](#_Toc16259267)

[4.3 应用 12](#_Toc16259268)

[4.3.1 流程节点显示 12](#_Toc16259269)

[4.3.2 流程节点计时 14](#_Toc16259270)

[4.3.3 SIT提测-需求选择 14](#_Toc16259271)

[4.3.4 SIT提测-流程 15](#_Toc16259272)

[4.3.5 SIT提测-版本发布 16](#_Toc16259273)

[4.3.6 SIT测试-版本部署 17](#_Toc16259274)

[4.3.7 SIT测试 17](#_Toc16259275)

[4.3.8 UAT提测 18](#_Toc16259276)

[4.3.9 UAT测试-版本部署 19](#_Toc16259277)

[4.3.10 UAT测试 20](#_Toc16259278)

[4.3.11 上线申请 20](#_Toc16259279)

[4.3.12 服务管理 21](#_Toc16259280)

[4.3.13 pipeline 23](#_Toc16259281)

[4.3.14 版本管理 23](#_Toc16259282)

[4.4 权限中心 24](#_Toc16259283)

[4.4.1 权限中心-组织机构 24](#_Toc16259284)

[4.4.2 权限中心-角色管理 25](#_Toc16259285)

[5 测试总结 26](#_Toc16259286)

[5.1 结果分析 26](#_Toc16259287)

[5.2 功能和优势 26](#_Toc16259288)

# 概述

## 测试目的

编写本文档目的是描述使用devops研发运行一体化测试需求中提出的测试用例进行现场测试，测试内容包括devops基本配置、CI/CD流水线、工作流等。

## 测试计划

|  |  |  |
| --- | --- | --- |
| **项目** | **进行时间** | **人员** |
| 平台梳理 | 2019.5.28~2019.6.14 | 潘露 |
| 功能测试 | 2019.6.18~2019.7.12 | 潘露 |
| 功能测试 | 2019.7.16~2019.7.26 | 潘露 |
| 测试报告整理 | 2019.7.30~2019.8.2 | 潘露 |
| 确认测试结果 | 2019.8.6~2019.8.8 | 潘露 |
| 测试报告修订 | 2019.8.8~2019.8.9 | 潘露 |

# 软硬环境需求

本节详细介绍了本次测试的硬件和软件要求。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **主机名** | **系统** | **cpu** | **内存** | **磁盘** | **ip** | **组件** |
| portal1 | centos7.5-minimal | 8C | 16G | 100G | 10.45.4.229 | paas-business-config-portal paas-business-config-admin paas-business-config paas-web paas-application paas-deployer paas-runtime paas-appstore |
| portal2 | centos7.5-minimal | 8C | 16G | 100G | 10.45.4.228 | paas-basic-gateway paas-basic-task ，paas-basic-license，paas-basic-pipeline paas-basic-report ，paas-basic-monitor paas-generalpurpose XXL-JOB |
| devops1 | centos7.5-minimal | 8C | 16G | 100G | 10.45.4.228 | devops-flow-portal  devops-message devops-boclink devops-jira devops-workflow-engine devops-workflow pass-basic-upms |
| component | centos7.5-minimal | 8C | 16G | 300G | 10.45.4.226 | rabbitmq-server、consul、mariadb、registry、redis,NFS |
| ci | centos7.5-minimal | 8C | 16G | 200G | 10.45.4.225 | jenkins-master、sonar、clairwithpostgres、mysql，mysql |
| master | centos7.5-minimal | 8C | 16G | 100G | 10.45.4.227 | K8s Master、paas-agent、paas-kubewatch、paas-Fabric、LoGRotate |
| node01 | centos7.5-minimal | 8C | 16G | 200G | 10.45.4.231 | K8s node、Ingress、Prometheus(any）、Paas-Fabric、Beyondlet、LoGRotate |
| node02 | centos7.5-minimal | 8C | 16G | 200G | 10.45.4.230 | K8s node、Ingress、ELK(any）、Paas-Fabric、Beyondlet、LoGRotate |

# 功能列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号ID** | **功能模块** | **功能说明** | **优先级** | **备注** |
| 1、 | 应用系统概览 | 分应用系统展示开发流程，工作流配置节点按照相应功能跳转与推动 | 高 |  |
| 2、 | 环境类型 | 设置环境类型，启用与停止，分配资源，实现租户资源分配 | 高 |  |
| 3、 | 工作流模板 | 自定义设置开发工作流程，配置工作流节点 | 高 |  |
| 4、 | 制品库 | 添加制品库节点，为各个应用创建相应制品仓库 | 高 |  |
| 5、 | 应用列表 | 关联jira项目与应用，将jira项目需求开发纳管至devops平台 | 高 |  |
| 6、 | 工作流-节点展示与推动 | 展示开发工作流至何种状态，当前租户下需求开发测试状态 | 高 |  |
| 7、 | SIT测试 | 流程中SIT环节需求的选择、版本提测即服务发布，版本部署，SIT测试 | 高 |  |
| 8、 | UAT测试 | 流程中UAT环节，提测、版本部署，SIT测试 | 高 |  |
| 9、 | 上线申请 | UAT测试通过，选择通过版本提交上线申请，可查看完成需求状态 | 高 |  |
| 10、 | 服务管理 | 支持应用服务的管理、执行pipeline与部署，单服务版本发布、部署 | 高 |  |
| 11、 | Pipeline | 自定义CI/CD；创建符合应用服务的satge，容器和非容器的CI/CD | 高 |  |
| 12、 | 版本管理 | 支持版本控制，便于开发测试流程 | 高 |  |
| 13、 | 权限中心 | 支持LDAP导入用户，自定义角色权限如提测、测试和配置权限 | 高 |  |

# 测试结果

## 应用系统概览

|  |  |
| --- | --- |
| **测试项目** | 分应用系统展示资源 |
| **测试编号** |  |
| **测试目的** | 实时展示工作流的进展，迭代流程、动态 |
| **预置条件** | 平台基本配置完成，应用创建完成配置且关联工作流，节点设置 |
| **操作步骤** | 1. 租户管理员登入平台首页，点击应用列表，应用列表页展示了该管理员管理的多个应用 2. 指定某应用，点击应用名称，进入概览 3. 指定应用版本，可展示当前版本下应用的工作流，以及工作流运行情况 4. 点击节点上的资源可跳转至指定的菜单栏 |
| **预期结果** | 各个页面跳转准确稳定，应用可按照版本展示工作流运行和pipeline构建情况，并且支持点击节点跳转到对应资源 |
| **实测结果** | 测试通过 |
| **备注** |  |

## devops

### 环境类型

|  |  |
| --- | --- |
| **测试项目** | 环境类型 |
| **测试编号** |  |
| **测试目的** | 支持环境类型基本操作，与集群分区相关联，具体操作如启用与停止 |
| **预置条件** | 平台部署完成，且k8s集群已添加进来，并有可用的节点 |
| **操作步骤** | 1. 平台管理员登录平台，添加环境类型，输入环境类型的基本信息，环境类型名称、环境类型描述，选择启用该环境类型，点击确定 2. 列表中的环境类型即时刷新，状态为启用状态 3. 进入基础设施-分区管理下，可看到刚刚创建好的分区即环境类型 4. 用户登录平台，申请环境时，可勾选刚刚创建的分区-环境类型。默认的分区节点数为1 5. 指定新建的分区，点击操作栏下的迁入迁出节点，即可新增或减少该分区下的节点 6. 回到devops-环境类型下，指定某环境，点击其操作栏下的停止按钮 7. 该环境类型的状态转变为停止，用户登录平台，申请环境时，停止状态的环境类型不显示在下拉框中 8. 回到devops环境类型下，将停止的环境类型启用，用户登录平台就可申请该环境类型的环境，进行使用 9. 用户申请该环境类型的环境并投入使用后，回到devops环境类型下，将环境类型停止，返回已被引用，不可停止，编辑时被引用的环境类型也不可编辑 10. 指定已被部署实例的环境类型，点击删除按钮，返回操作失败提示该环境类型下有未删除的应用 |
| **预期结果** | 环境类型与卡口管理功能实现稳定，符合用户使用习惯，环境类型下有应用信息不可被删除 |
| **实测结果** | 测试通过 |
| **备注** |  |

### Pipeline任务模板

|  |  |
| --- | --- |
| **测试项目** | pipeline任务模板 |
| **测试编号** |  |
| **测试目的** | 支持自定义Pipeline的任务模板基本参数配置 |
| **预置条件** | 平台部署完毕 |
| **操作步骤** | 1. 创建任务模板，输入模板的名称、选择任务模板类型、logo、模板内容、参数等，参数包含系统参数和普通参数，点击保存 2. 列表中的任务模板即时更新，点击模板内容列“查看”，弹框显示模板内容详细信息，点击参数变量列“查看”，弹框显示参数详细信息。 3. 编辑pipeline模板，新增系统参数如制品仓库，点击保存 4. 查看参数变量，可看到刚刚编辑新增的系统参数制品仓库 5. 再次编辑该任务模板，新增系统参数选择制品仓库，点击确定，返回提示已有该系统参数 |
| **预期结果** | 任务模板编辑通过，支持自定义创建jenkins的stage |
| **实测结果** | 测试通过 |
| **备注** |  |

### 制品库

|  |  |
| --- | --- |
| **测试项目** | 制品库 |
| **测试编号** |  |
| **测试目的** | 保证制品库的正常使用，被引用的制品库不可删除 |
| **预置条件** |  |
| **操作步骤** | 1. 平台管理员登录平台，进入devops-制品库，点击添加制品库按钮 2. 在弹出的新增对话框，输入制品库的基本信息： 制品库节点、地址、类型、用户名和密码，点击确定 3. 制品库列表即时刷新 4. 新增与已有的制品库相同地址和节点的制品库，新增失败，提示已有重名的制品库 5. 用户使用制品库，并已上传制品至相应的服务制品库下，此时制品库不可被删除 |
| **预期结果** | 支持制品库的基本管理操作，被引用的制品库不可删除 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 应用列表与租户(应用分类)

|  |  |
| --- | --- |
| **测试项目** | 应用列表 |
| **测试编号** |  |
| **测试目的** | 支持将jira上项目应用关联，并接入平台管理 |
| **预置条件** |  |
| **操作步骤** | 1. 管理员登录平台，进入DevOps-应用分类，点击新增应用按钮 2. 输入应用的基本信息如名称、关键字、关联jira、项目类别、负责人和管理员，点击确定 3. 在对应的类别下列表即时刷新，展示刚刚创建的应用，在租户管理下，可以看到刚刚新增的租户，关键字即租户名称 4. 应用卡片右侧操作-编辑，可编辑应用标签、类别、名称与描述 5. 管理员到租户管理下，找到刚刚创建的租户，点击操作下的资源配额，在弹出的资源配额，选择对应devops创建的分区和资源配额，点击确定 6. 租户的配额设置成功，可在租户管理员登录创建环境时，看到对应的配额 7. 指定租户，点击操作下的租户管理员，编辑租户的所属租户管理员 |
| **预期结果** | Jira上的应用和项目关联入平台，应用与租户建立关联 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 自定义图标Logo

|  |  |
| --- | --- |
| **测试项目** | 自定义图标Logo |
| **测试编号** |  |
| **测试目的** | 支持自定义图标Logo |
| **预置条件** |  |
| **操作步骤** | 1. 准备需要替换的logo文件  2. 点击平台设置，系统配置，按照要求上传顶部logo文件，菜单logo文件  3. 退出后清除缓存查看是否生效 |
| **预期结果** | 支持用户自定义logo上传 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 驾驶舱配置

|  |  |
| --- | --- |
| **测试项目** | 驾驶舱配置 |
| **测试编号** |  |
| **测试目的** | 支持自定义驾驶舱的应用系统基本参数配置 |
| **预置条件** |  |
| **操作步骤** | 1、编辑应用系统配置总数，输入应用总数和自主研发数量并保存  2、创建应用系统概述，输入上线数量、新项目立项数量，并选择年份和月份，在此不能修改应用总数和自主研发数量，然后点击保存。可以看到新增信息显示无误，并且可以执行编辑、删除等操作。  3、登录用户平台，切换到概览下的驾驶舱界面，可以选择当前应用系统并能够看到管理员再1、2、3步骤创建的应用系统概述、上线数量、立项数量和它们的统计图等信息。 |
| **预期结果** | 应用系统配置模块测试通过，用户平台的驾驶舱卡片中能够正常显示其数据和统计信息 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 工作流模板

|  |  |
| --- | --- |
| **测试项目** | 工作流模板的基本配置 |
| **测试编号** |  |
| **测试目的** | 工作流模板支持编辑 |
| **预置条件** | 平台部署完成 |
| **操作步骤** | 1. 进入devops-工作流，查看工作流模板信息 2. 支持编辑节点，在对应的工作节点关联资源和设置jira节点，资源来自资源管理，jira节点。 3. jira中issue的工作流走向是：待办->开发中->开发集成->待测试->SIT测试中->阻塞->SIT完成->UAT测试中->UAT完成->待发布->已发布 |
| **预期结果** | 支持查看工作流模板，关联资源和jira节点 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 资源管理

|  |  |
| --- | --- |
| **测试项目** | 支持资源的基本配置 |
| **测试编号** |  |
| **测试目的** | 资源支持自定义设置 |
| **预置条件** | 平台部署完成 |
| **操作步骤** | 1. 进入资源配置，新增资源，输入资源名称、url、image、状态和类型信息，确定保存 2. 可以看到创建的资源信息完整展示 3. 进入工作流，选择一条已经创建的工作流模板并点击编辑节点 4. 编辑其关联资源和关联jira节点，关联资源即为上面步骤1和2中已经配置的资源；关联的jira节点即是jira中issue工作流的各种状态，将两者与系统中的工作流节点绑定 |
| **预期结果** | 支持自定义资源，并关联资源和jira节点 |
| **实测结果** | 测试通过 |
| **备注** |  |

## 应用

### 流程节点显示

|  |  |
| --- | --- |
| **测试项目** | 流程节点显示 |
| **测试编号** |  |
| **测试目的** | 支持节点随研发状态即时显示 |
| **预置条件** |  |
| **操作步骤** | 1. 版本规划后，在jira上有关联版本的需求，研发流水线形成 2. 需求池节点默认显示进行中-即橙色，下方数量展示了当前应用下未关联版本的需求 3. 版本需求默认显示关联此版本的所有需求，显示为完成状态-蓝色 4. 开发配置节点下包含各项配置，默认显示进行中-即橙色 5. 当有一个需求进入研发中状态，分母显示当前版本需求数量，分子显示jira进入开发中及之后状态的当前版本需求数量 6. Sit提测后，sit节点变为进行中，当sit测试失败，那么在下一次sit提测成功之前，sit节点为失败 7. Uat节点当提测时状态展示为进行中，失败展示为红色，成功-蓝色 8. 上线审批节点在用户申请时，由未开始到进行中，审批通过-完成 9. 当上线成功后，上线节点状态点亮 |
| **预期结果** | 节点上的状态按照实际运行展示 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 流程节点计时

|  |  |
| --- | --- |
| **测试项目** | 流程节点计时 |
| **测试编号** |  |
| **测试目的** | 各个节点计时统计无误，展示研发进度 |
| **预置条件** |  |
| **操作步骤** | 1. 研发节点的计时规则：当版本的第一个需求在jira中进入开发时开始计时，当版本最后一个需求在jira中进入待测试结束计时；再次插入需求，则再按这种方法计时，最终将所有计时求和显示  2. sit节点计时：当点击sit测试开始时开始计时，点击测试完成时结束计时，如果有多轮sit测试，最终将所有计时求和显示  3.uat节点：当测试人员点击uat测试开始时开始计时，点击测试完成时结束计时，如果有多轮uat测试，最终将所有计时求和显示 |
| **预期结果** | 支持节点计时统计，展示研发进度 |
| **实测结果** | 测试通过 |
| **备注** |  |

### SIT提测-需求选择

|  |  |
| --- | --- |
| **测试项目** | Sit提测-需求选择 |
| **测试编号** |  |
| **测试目的** | 支持租户关联分区，租户可以分配一个或者多个分区 |
| **预置条件** | 平台部署完成，创建数据中心， 创建2个分区 |
| **操作步骤** | 正常情况下sit提测时，是在所选需求完成开发处于待测试情况下，可进行提测   1. 异常情况一，在提测之前有提测单A（或者多个提测单）的状态处在测试未开始和进行中，那之前提测单A（或者多个提测单）中提测过的需求不可以选择   操作步骤：  ①对beta1版本进行提测，已完成的需求有1，2和3，其中1，2所属服务为A，3所属服务B  ②提测需求选择1和3，此时下方服务列表已筛选出提测需求所属服务，进行sit版本部署，部署完后未点击开始，状态处于测试未开始  ③对beta2版本提测，其中需求1和3置灰不可选择  ④将beta1版本点击开始测试，其状态变为正在测试中，再次选择beta2版本提测在选择提测需求时，需求1和3不可选择   1. 异常情况二，在此之前有提测单B（或者多个提测单）状态处在测试结束，那与提测单B（或者多个提测单）中未通过测试的需求相关联的服务的所有需求都不可以选择。   操作步骤：  ①上面beta1开始测试，发现需求1测试阻塞有问题，需求3测试通过，测试结束，测试结果为不通过，需求1未通过测试  ②当对beta2版本提测时，提测需求框中需求1和需求3均不可选择 |
| **预期结果** | Sit提测需求按照设计筛选提测 |
| **实测结果** | 测试通过 |
| **备注** |  |

### SIT提测-流程

|  |  |
| --- | --- |
| **测试项目** | SIT提测-流程 |
| **测试编号** |  |
| **测试目的** | 支持sit提测，服务发布至制品库 |
| **预置条件** | 应用测试版本新增成功，版本需求进入待测试 |
| **操作步骤** | 1. 选择提测版本，勾选完成的需求，列表中筛选出需求关联的服务，选择服务测试版本 2. 选择版本执行pipeline、分支名，这里有单服务发布和一键发布，需要注意的是单服务发布不关联需求，一般用一键发布   3、当前服务提测版本有变动请求停止发布，点击服务列表中的停止按钮  4、更改服务分支，点击重新发布，发布成功 |
| **预期结果** | 支持待测试需求提测，服务发布，多版本 |
| **实测结果** | 测试通过 |
| **备注** |  |

### SIT提测-版本发布

|  |  |
| --- | --- |
| **测试项目** | Sit提测-版本发布 |
| **测试编号** |  |
| **测试目的** | 版本发布失败情况下sit提测 |
| **预置条件** |  |
| **操作步骤** | 1. 选择提测版本，勾选完成的需求，列表中筛选出需求关联的服务，选择服务测试版本 2. 点击确定即一键发布，此时版本管理中相应的beta版本发布作业中会记录正在执行的版本发布操作 3. 根据记录，可以查看版本发布状态，此时有服务ci失败，则该版本发布失败 4. 回到sit提测，选择刚刚发布的版本，版本后缀为未提测 |
| **预期结果** | 版本发布过程中如果某服务发布失败，则该版本就发布失败 |
| **实测结果** | 测试通过 |
| **备注** |  |

### SIT测试-版本部署

|  |  |
| --- | --- |
| **测试项目** | Sit版本部署 |
| **测试编号** |  |
| **测试目的** | Sit版本部署支持物理和容器部署 |
| **预置条件** |  |
| **操作步骤** | 1. 租户管理员登录平台，进入sit节点-sit版本部署 2. 弹出的sit版本部署对话框，指定最新的提测版本，查看需求详细 3. 点击部署，弹出sit版本部署对话框，选择环境，并指定该版本下所有服务的最新版本，选择执行的pipeline，一键部署 |
| **预期结果** | 支持sit版本容器和物理部署 |
| **实测结果** | 测试通过 |
| **备注** |  |

### SIT测试

|  |  |
| --- | --- |
| **测试项目** | Sit测试 |
| **测试编号** |  |
| **测试目的** | 支持Sit测试 |
| **预置条件** | Sit版本部署完成 |
| **操作步骤** | 1. 点击概览中的sit节点-sit版本部署，对话框展示当前已提测过的多个版本 2. 选择部署完成的sit版本，点击开始测试，此时当前提测单中所有的需求由待测试-sit测试中 3. 根据测试返回，需求状态有sit完成、阻塞   第一种情况：当前提测单中所有需求测试没有问题，sit完成，点击测试完成，输入通过测试  第二种情况：当前提测单中有一个需求测试未通过，阻塞，并提交了一二级bug，点击测试完成，则提示不可设置通过测试，展示出issue关联的bug  第三种情况：当前提测单中所有需求处于sit完成，测试完成选择通过，会返回当前需求存在一二级bug，是否确定提交通过，确认提交通过，则将此次存在一二级BUG的测试通过操作进行记录审计。 |
| **预期结果** | Sit测试将所有issue的状态由待测试转至sit测试中，阻塞和sit完成，测试通过验证一二级bug提示准确 |
| **实测结果** | 测试通过 |
| **备注** |  |

### UAT提测

|  |  |
| --- | --- |
| **测试项目** | UAT提测 |
| **测试编号** |  |
| **测试目的** | Uat提测 |
| **预置条件** | 经过几轮sit测试后所有需求开发完成，指定某版本进入uat测试 |
| **操作步骤** | 1. 点击sit节点上uat提测，弹出uat提测对话框 2. 对话框展示了当前所有进行sit提测的版本信息，选择一可作为规划版本需求完成度的版本，点击操作中的需求详细 3. 需求详细中展示了指定的sit版本关联的需求信息以及sit测试结果 4. 关闭需求详细，点击uat提测，弹出uat提测确认对话框 5. 点击确定，流程节点推送到uat节点。   Uat提测的唯一性：同一时间段只能进行一次uat提测，再上一次uat提测结束之前，不可再次提交uat提测  1、点击另一sit版本的uat提测，返回提示当前已有uat提测 |
| **预期结果** | Uat提测符合设计，需求详细展示准确，再上一次uat未完成之前不可再次提交uat提测 |
| **实测结果** | 测试通过 |
| **备注** |  |

### UAT测试-版本部署

|  |  |
| --- | --- |
| **测试项目** | UAT测试-版本部署 |
| **测试编号** |  |
| **测试目的** | UAT版本部署一键部署验证通过 |
| **预置条件** |  |
| **操作步骤** | 1、点击UAT节点的uat版本部署，弹出uat版本对话框  2、选择指定一sit版本，验证提测版本信息的准确性  3、点击部署，弹出版本部署对话框，框内展示当前版本下的服务的最新版本及其状态  4、选择环境，点击一键部署 |
| **预期结果** | 支持uat版本容器和物理部署 |
| **实测结果** | 测试通过 |
| **备注** |  |

### UAT测试

|  |  |
| --- | --- |
| **测试项目** | UAT测试 |
| **测试编号** |  |
| **测试目的** | 支持UAT测试 |
| **预置条件** | Uat版本部署成功 |
| **操作步骤** | 1. 点击概览中的uat节点-uat版本部署，对话框展示当前提交的uat版本 2. 选择部署完成的uat版本，点击开始测试，此时当前提测单中所有的需求由sit完成-uat测试中   3、根据测试返回，需求状态有uat完成、阻塞  第一种情况：当前提测单中所有需求测试没有问题，sit完成，点击测试完成，输入通过测试  第二种情况：当前提测单中有一个需求测试未通过，阻塞，并提交了一二级bug，点击测试完成，则提示不可设置通过测试，展示出issue关联的bug  第三种情况：当前提测单中所有需求处于sit完成，测试完成选择通过，会返回当前需求存在一二级bug，是否确定提交通过，确认提交通过，则将此次存在一二级BUG的测试通过操作进行记录审计。 |
| **预期结果** | UAT测试验证通过 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 上线申请

|  |  |
| --- | --- |
| **测试项目** | 上线申请 |
| **测试编号** |  |
| **测试目的** | 支持上线申请 |
| **预置条件** | 已有uat测试完成 |
| **操作步骤** | 1. 租户管理员登录平台进入概览-UAT节点，点击上线申请   2、选择某一uat版本，查看需求完成情况，查看详细，验证需求信息准确性，点击上线申请提交按钮   1. 上线申请提交成功 |
| **预期结果** | 上线申请提交成功 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 服务管理

|  |  |
| --- | --- |
| **测试项目** | 服务管理 |
| **测试编号** |  |
| **测试目的** | 支持服务的增删改查 |
| **预置条件** |  |
| **操作步骤** | 1、租户管理员登录平台，进入服务管理，点击创建服务  2、输入服务名称、服务中文名，这里服务名输入字母，建议与工程名一致；负责人从当前租户下导入的用户选择  服务创建成功后，创建配置服务的pipeline   1. 点击服务名称，进入服务详情-pieline 2. 点击创建pipeline，输入pipeline的名称、所属服务和触发方式，点击确定 3. 在下方可自定义编辑pipeline各个stage，这里ci，cd分开，以java任务为例 4. 点击+按钮，新增阶段，输入阶段名，点击确定 5. 新增并行任务，选择任务模板，可根据任务模板类型选择对应的模板 6. 点击模板编辑按钮，可输入服务具体参数信息，点击确定 7. 组织完成pipeline后，点击确定   服务版本创建   1. 点击服务构建tab，点击创建服务版本 2. 输入版本的基本信息、pipeline，点击确定 3. 为服务release版本添加beta版本，点击操作下添加版本，输入基本信息 4. 点击更多下的构建记录，查看构建记录，可进入查看服务pipeline执行   服务发布 |
| **预期结果** | 服务管理验证成功，devops上服务与jira上相应项目-应用下服务一一对应 |
| **实测结果** | 测试通过 |
| **备注** |  |

### pipeline

|  |  |
| --- | --- |
| **测试项目** | Pipeline |
| **测试编号** |  |
| **测试目的** | 支持服务pipeline的管理执行 |
| **预置条件** | 平台中已添加应用服务，服务代码仓库配置成功 |
| **操作步骤** | 1、点击创建pipeline，输入pipeline名称及所属服务，点击确定  2、下方展示出pipeline定义的tab，新增阶段和选择任务模板，组成一条pipeline  3、点击操作下的执行按钮，可调动jenkins执行任务，点击记录可看到刚刚执行的作业  4、点击序号可查看每个stage的日志信息  5、在执行过程中，停止pipeline构建，到相应记录下，点击操作下停止按钮  5、pipeline列表点击操作下的克隆按钮，可快速复制一条pipeline |
| **预期结果** | Pipeline验证通过 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 版本管理

|  |  |
| --- | --- |
| **测试项目** | 版本管理 |
| **测试编号** |  |
| **测试目的** | 支持应用版本管理 |
| **预置条件** | 已有应用，服务信息 |
| **操作步骤** | 1、点击创建release版本，弹出release版本对话框  2、输入版本的基本信息如版本名称、选择工作流，版本描述、开发日期与关联的服务版本信息及pipeline，点击确定  3、指定一release版本，点击更多下添加依赖beta版本  4、输入beta版本的基本信息，勾选关联服务beta版本，点击确定  5、指定一应用beta版本，点击操作下版本发布，弹出版本发布对话框  6、选择其关联服务版本的pipeline与分支，点击确定  7、点击操作下发布作业可看到刚刚发布的作业，查看记录  发布版本  1、指定某发布版本，点击操作下的版本详情，可查看版本的基本信息  2、点击操作下的版本部署，选择部署环境与beta版本关联服务版本的pipeline，点击确定 |
| **预期结果** | 应用版本管理、版本发布与部署功能正常 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 代码仓库

|  |  |
| --- | --- |
| **测试项目** | 代码仓库 |
| **测试编号** |  |
| **测试目的** | 支持代码仓库增删改 |
| **预置条件** | 已有服务，代码仓库的地址、用户名和密码 |
| **操作步骤** | 1、点击创建代码库，弹出申请代码库对话框  2、输入代码库所属服务、仓库地址、账号和密码信息，点击确定  3、代码库正常添加  4、编辑代码仓库，修改其信息(一般一个代码仓库和一个服务关联，不建议修改)，修改正常  5、删除代码仓库正常 |
| **预期结果** | 代码仓库增删改操作数据展示无误 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 环境管理

|  |  |
| --- | --- |
| **测试项目** | 环境管理 |
| **测试编号** |  |
| **测试目的** | 支持环境管理 |
| **预置条件** | 管理员平台已经创建好环境类型并且为其分配资源，环境处于启用状态；用户可以使用创建好的环境类型 |
| **操作步骤** | 1、点击创建环境，输入环境名称，然后选择环境类型，并选择资源类型。  2、若资源类型为docker，则用户需要为其分配cpu、内存、实例数和存储等信息；若资源类型为物理/虚拟机，则需要填写此物理机/虚拟机的ip、用户名、密码、cpu、存储和内存等信息，然后保存  3、刷新环境管理界面，用户创建的环境名称，分配的资源类型、环境类型和各种配额等信息的显示和步骤2中用户创建/分配的资源数一致  4、点击编辑，可以修改环境的信息  5、点击环境名称超链接，进入环境中，可以看到此环境的各种如环境名称、环境类型、实例总数等信息，并且能够看到在此环境中运行的服务和其运行状态，可以查询服务的日志信息、或者停止、运行此服务  6、删除环境之前，首先将环境中运行的服务或应用停止并删除，再删除环境 |
| **预期结果** | 环境管理的新增、分配、删除等操作无误 |
| **实测结果** | 测试通过 |
| **备注** |  |

## 权限中心

### 权限中心-组织机构

|  |  |
| --- | --- |
| **测试项目** | **权限中心-组织机构** |
| **测试编号** |  |
| **测试目的** | 支持自定义组织 |
| **预置条件** | 平台部署完成 |
| **操作步骤** | 1、租户管理员登录平台，进入权限中心-组织管理  2、指定用户点击操作下分配角色按钮，选择角色，点击确定  3、点击用户名称，查看用户的基本信息和角色信息 |
| **预期结果** | 支持创建自定义组织，支持单个用户分配自定义角色 |
| **实测结果** | 测试通过 |
| **备注** |  |

### 权限中心-角色管理

|  |  |
| --- | --- |
| **测试项目** | 权限中心-自定义角色 |
| **测试编号** |  |
| **测试目的** | 支持自定义角色以及角色的功能授权 |
| **预置条件** | 平台部署完成 |
| **操作步骤** | 1、点击新增按钮，填写角色名称信息  2、勾选角色菜单权限以及devops权限  3、点击确定  4、用户以创建的角色登录平台，验证权限设置 |
| **预期结果** | 自定义角色新增完成，角色权限设置通过 |
| **实测结果** | 测试通过 |
| **备注** |  |

# 测试总结

## 结果分析

本次功能测试，主要测试了devops工作流配置、与jira 的关联、应用版本管理、pipeline自定义stage创建、服务发布与部署、多租户多环境支持、容器与非容器CI/CD，Java的持续集成与持续部署等功能，测试结果与客户的需求基本吻合。

Devops平台以容器、Kubernetes为底层技术和调度框架，集成jira项目管理，通过自定义stage、涵盖持续开发持续交付全过程，支持开发、测试一体化，从需求的创建、到开发、集成、部署、测试，工作流的配置，使得开发流程运行态一目了然，能够更好的跟踪定位当前项目的开发测试情况，规范开发流程从而提高工作效率。