概览

资源概览

|  |  |
| --- | --- |
| 用例名称 | 资源概览 |
| 测试目的 | 验证用户可以正确获取到集群、命名空间、实例、应用和运行服务信息 |
| 预置条件 | 1.用户正常登录TSF控制台  2.用户已经预先创建好了集群、命名空间、应用  3.部分应用已经部署运行起来 |
| 测试过程 | 1.登录TSF控制台  2.单击“概览” |
| 预期结果 | 登录成功  在概览页面上半部分可以看到“资源概览”  资源概览中的集群、命名空间、应用、运行服务的个数与实际配置一致  “实例”项中会显示实例总数以及“已启动”和“未启动”的实例个数 |
| 测试结果 | [【TKE纳管TCNP需求】概览页配额概况展示丢失部分信息](https://www.tapd.cn/60031890/bugtrace/bugs/view?bug_id=1160031890001011299) |
| 备注 |  |

服务运行情况

|  |  |
| --- | --- |
| 用例名称 | 服务运行情况 |
| 测试目的 | 验证用户可以根据设置的时间段查询到“服务平均耗时（ms）Top10”、“接口平均耗时（ms）Top10”、“服务请求量Top10”、“接口请求量Top10”、“服务失败率Top10”、“接口失败率Top10”六项数据 |
| 预置条件 | 1.用户正常登录TSF控制台  2.用户已经预先创建好了集群、命名空间、应用  3.已经部署好测试应用并且运行起来  4.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台  2.在概览页面下半部分可以看到“服务运行情况”  3.选择命名空间和时间查询服务运行情况 |
| 预期结果 | 1.登录成功  2.成功查询到指定命名空间和时间段内的服务运行情况 |
| 测试结果 | |  | | --- | | [【 TKE纳管TCNP需求】需确认该版本中 概览页是否包含接口相关监控](https://www.tapd.cn/60031890/bugtrace/bugs/view?bug_id=1160031890001011335) | |
| 备注 | 查询到的统计数据与测试demo的实现有关  查询实时数据时（比如近24小时），由于采集数据的延时，会有微小误差 |

集群

创建虚机集群

|  |  |
| --- | --- |
| 用例名称 | 创建虚机集群 |
| 测试目的 | 验证用户可以创建虚机集群 |
| 预置条件 | 1.用户正常登录平台  2.用户具有创建集群的权限 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”>“新建集群”  3.选择集群类型为虚机集群  4.在弹出的对话框中设置“集群名”和“备注”  5.点击“提交”按钮 |
| 预期结果 | 1.登录成功  2.打开集群页面成功  3.提交之后在集群页面显示新建的集群  4.集群中有一个default命名空间 |
| 测试结果 | Pass |
| 备注 |  |

创建容器集群

|  |  |
| --- | --- |
| 用例名称 | 创建容器集群 |
| 测试目的 | 验证用户可以创建容器集群 |
| 预置条件 | 1.用户正常登录平台  2.用户具有创建集群的权限 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”>“新建集群”  3.选择集群类型为容器集群  4.在弹出的对话框中设置“集群名”和“备注”  5.点击“提交”按钮 |
| 预期结果 | 1.登录成功  2.打开集群页面成功  3.提交之后在集群页面显示新建的集群  4.集群中一个default命名空间 |
| 测试结果 | [【 TKE纳管TCNP需求】创建容器集群后容器内未新建default命名](https://www.tapd.cn/60031890/bugtrace/bugs/view?bug_id=1160031890001011336) |
| 备注 |  |

查询集群基本信息

|  |  |
| --- | --- |
| 用例名称 | 查询集群基本信息 |
| 测试目的 | 验证用户可以查询集群 |
| 预置条件 | 1.用户正常登录平台  2.用户具有创建集群的权限 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，选择需要查询的集群ID  IMG_256  3.切换至“基本信息”页面 |
| 预期结果 | 1.登录成功  2.打开集群页面成功  3.虚机集群基本信息页面显示集群的ID、名称、状态、云主机数、总资源、所在地域、所在可用区、集群网络、描述、创建时间、更新时间  4.容器集群基本信息页面显示集群ID、名称、状态、云主机数量、已分配资源、总资源、所在地域、所在可用区、集群网络、容器网络、描述、创建时间、更新时间 |
| 测试结果 | Pass |
| 备注 |  |

修改集群基本信息

|  |  |
| --- | --- |
| 用例名称 | 修改集群基本 |
| 测试目的 | 验证用户可以修改集群集群基本信息 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，选择需要修改的集群ID  3.切换的“基本信息”页面  4.点击“描述”右侧的按钮，修改描述  IMG_256 |
| 预期结果 | 1.登录成功  2.打开集群页面成功  3.成功修改描述 |
| 测试结果 | Pass |
| 备注 |  |

删除集群

|  |  |
| --- | --- |
| 用例名称 | 删除集群 |
| 测试目的 | 验证用户可以删除集群 |
| 预置条件 | 1.用户正常登录平台`  2.用户已经创建好集群  3.集群中没有非默认的命名空间 |
| 测试过程 | 1.登录TSF控制台  2.集群中没有非默认的命名空间和节点，点击需要删除的集群最右侧的“删除”按钮  3.在弹出的对话框中选择“确认” |
| 预期结果 | 1.登录成功  2.步骤3删除集群成功 |
| 测试结果 | Pass |
| 备注 |  |

创建命名空间

|  |  |
| --- | --- |
| 用例名称 | 创建命名空间 |
| 测试目的 | 验证用户可以创建命名空间 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，点击需要创建命名空间的集群ID  3.切换到“命名空间”页面，单击“关联命名空间”，单击“新建命名空间”  4.在弹出的对话框中设置“名称”和“备注”，点击提交  5.在另外一个集群中创建命名空间一个与步骤4同名的命名空间，点击提交 |
| 预期结果 | 1.当命名空间不存在时执行创建操作  2.当命命名空间已经存在时执行集群和命名空间绑定操作  3.命名空间以及页面可以查看到命名空间绑定的所有集群 |
| 测试结果 | Pass |
| 备注 |  |
| 备注 |  |

删除命名空间

|  |  |
| --- | --- |
| 用例名称 | 删除命名空间 |
| 测试目的 | 验证用户可以删除命名空间 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群和命名空间  3.命名空间没有被部署组引用  4.命名空间绑定多个集群 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，点击集群1的id  3.切换到“命名空间”页面，单击命名空间右侧的“解除绑定”按钮  4.在弹出的对话框中点击“确认”  5.进入到命名空间管理界面  IMG_256  6.单击命名空间右侧的“删除”按钮，在弹出的对话框中点击确认 |
| 预期结果 | 1.步骤4命名空间和集群1的绑定关系解除  2.步骤6命名空间被删除 |
| 测试结果 | Pass |
| 备注 |  |

容器集群导入云主机

|  |  |
| --- | --- |
| 用例名称 | 容器集群导入云主机 |
| 测试目的 | 用户可以给容器集群导入云主机 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群和命名空间 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，点击需要导入节点的集群ID  3.在“云主机列表”页面，单击“导入云主机”  4.在弹出的对话框中选择需要导入的服务器，选择镜像，设置密码  5.点击“提交”按钮 |
| 预期结果 | 1.登录成功  2.导入节点成功  3.在“节点列表”页面可以看到已经导入的云主机  4.导入的云主机会重装系统，重装完成之后，云主机状态为“运行中”，可用状态为“可用” |
| 测试结果 |  |
| 备注 |  |

云主机移出容器集群

|  |  |
| --- | --- |
| 用例名称 | 云主机移出容器集群 |
| 测试目的 | 验证用户可以将节点移出容器集群 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群  3.集群中已经导入节点  4.集群中没有运行的部署组 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，点击目标集群ID  3.在“云主机列表”页面，单击节点右侧的“移出”按钮  图形用户界面, 应用程序, Teams  描述已自动生成  4.点击确认 |
| 预期结果 | 1.登录成功  2.移出云主机成功  3.在“云主机列表”页面可以看到对应云主机已删除 |
| 测试结果 |  |
| 备注 |  |

云主机移出虚机集群

|  |  |
| --- | --- |
| 用例名称 | 云主机移出虚机集群 |
| 测试目的 | 验证用户可以将节点移出虚机集群 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群  3.集群中已经导入节点  4.集群中没有运行的部署组 |
| 测试过程 | 1.登录TSF控制台  2.单击“集群”，点击目标集群ID  3.云主机上没有运行的应用，在“云主机列表”页面，单击右侧的“移出”按钮，点击确认  图形用户界面, 应用程序  描述已自动生成  4.云主机上有运行的应用，在“云主机列表”页面，单击右侧的“移出”按钮 |
| 预期结果 | 1.登录成功  2.步骤3移出成功  3.步骤4移出按钮不可操作 |
| 测试结果 |  |
| 备注 |  |

命名空间

设置命名空间code

|  |  |
| --- | --- |
| 用例名称 | 设置命名空间code |
| 测试目的 | 验证用户可以设置命名空间code用于API网关创建API |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 1.登录TSF控制台  2.单击“命名空间”，点击命名空间的“设置Code”操作  图形用户界面, 应用程序  描述已自动生成  在弹出的对话框中设置code，点击“提交”  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.创建命名空间成功 |
| 测试结果 | Pass |
| 备注 |  |

命名空间列表页操作

|  |  |
| --- | --- |
| 用例名称 | 命名空间列表页操作 |
| 测试目的 | 验证用户可以在命名空间一级页面搜索、翻页等操作 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 1.登录TSF控制台  2.单击集群id  3.右上角搜索框中输入命名空间名称或者id搜索  4.右下角设置每页显示行数  5.右下角进行翻页操作 |
| 预期结果 | 1.步骤2跳转到集群云主机列表页  2.步骤3按命名空间名称和id搜索成功  3.步骤4设置每页显示行数生效  3.步骤5翻页操作正常 |
| 测试结果 | Pass |
| 备注 |  |

部署组

部署组详情页

|  |  |
| --- | --- |
| 用例名称 | 部署组详情页 |
| 测试目的 | 验证用户可以点击部署组id进去详情页 |
| 预置条件 | 1.用户正常登录平台  2.用户已创建部署组 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择集群和命名空间  3.点击部署组id进入部署详情页 |
| 预期结果 | 1.部署组实例列表显示部署组下的实例列表  2.日志页面显示部署组的日志，包括业务日志和日志配置项日志  3.关联应用配置，查看关联的应用配置  4.基本信息页面部署组的基本信息和日志配置项篇日志、弹性伸缩配置、联调配置 |
| 测试结果 | Pass |
| 备注 |  |

创建容器部署组

|  |  |
| --- | --- |
| 用例名称 | 创建容器部署组 |
| 测试目的 | 验证用户可以在部署组一级页面创建容器部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用、集群、命名空间 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择容器集群和命名空间  3.单击新建部署组按钮  4.设置部署组名、选择关联应用  5.设置实例资源限制、实例个数  6.设置访问设置  7.设置更新方式和日志配置项  8.点击提交 |
| 预期结果 | 1.创建容器部署组成功 |
| 测试结果 | Pass |
| 备注 |  |

容器部署组部署应用

|  |  |
| --- | --- |
| 用例名称 | 容器部署组部署应用 |
| 测试目的 | 验证用户可以通过容器部署组部署容器应用 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器部署组  3.用户已经上传容器镜像 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择容器集群和命名空间  3.单击部署组的部署应用操作  4.在弹出的对话框中选择镜像，设置资源限制、实例数量、JVM启动参数，提交  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.部署应用成功  2.部署组的实例数等于设置的实例数  3.容器中的应用使用设置的JVM参数启动  kubectl get pod -n 命名空间名称，查询命名空间下的pod信息  Kubectl describe pod pod名称 -n 命名空间，查看pod中的容器信息  kubectl exec -it pod名称 -c 容器名 /bin/bash -n 命名空间名，进入容器  ps -ef|grep java，查看程序启动时使用的jvm参数 |
| 测试结果 | Pass |
| 备注 |  |

容器部署组启动应用

|  |  |
| --- | --- |
| 用例名称 | 容器部署组启动应用 |
| 测试目的 | 验证用户可以在部署组一级页面进行启动应用操作 |
| 预置条件 | 1.用户正常登录平台  2.容器部署组已停止 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择容器集群和命名空间  3.单击部署组的启动应用操作，弹框中点击确认 |
| 预期结果 | 1.容器部署组中所有实例启动成功 |
| 测试结果 | Pass |
| 备注 |  |

容器部署组停止应用

|  |  |
| --- | --- |
| 用例名称 | 容器部署组停止应用 |
| 测试目的 | 验证用户可以在部署组一级页面进行停止应用操作 |
| 预置条件 | 1.用户正常登录平台  2.容器部署组运行中 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择容器集群和命名空间  3.单击部署组的停止应用操作，弹框中点击确认 |
| 预期结果 | 1.容器部署组中所有实例停止 |
| 测试结果 | Pass |
| 备注 |  |

容器部署组实例操作

|  |  |
| --- | --- |
| 用例名称 | 容器部署组实例操作 |
| 测试目的 | 验证用户可以对容器部署组进行销毁操作 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署并启动应用 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”点击目标应用ID  3.在“部署组”页面，点击部署组id进入服务实例列表  4.点击实例的销毁操作  图片包含 屏幕截图  描述已自动生成 |
| 预期结果 | 1.实例销毁会把原来的实例删除，创建新的实例  kubectl get pod -n 命名空间名称，查看pod被重新创建 |
| 测试结果 | Pass |
| 备注 |  |

应用管理

创建容器部署方式应用

|  |  |
| --- | --- |
| 用例名称 | 创建容器部署方式应用 |
| 测试目的 | 验证用户可以创建容器部署方式的应用 |
| 预置条件 | 1.用户正常登录平台  2.用户拥有创建应用的权限 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，单击“新建应用”  3.在弹出的对话矿中设置“应用名”、“部署方式”选择虚容器部署  4.设置“应用类型”、“备注”，点击“提交”  IMG_256 |
| 预期结果 | 1.应用类型可以选择普通应用和Mesh应用  2.创建应用成功 |
| 测试结果 | Pass |
| 备注 |  |

查询应用基本信息

|  |  |
| --- | --- |
| 用例名称 | 查询应用基本信息 |
| 测试目的 | 验证用户可以查询应用基本信息 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，单击目标应用ID  3.切换到“基本信息”Tab页 |
| 预期结果 | 1.应用基本信息有如下  图形用户界面, 文本, 应用程序  描述已自动生成 |
| 测试结果 | Pass |
| 备注 |  |

修改应用基本信息

|  |  |
| --- | --- |
| 用例名称 | 修改应用基本信息 |
| 测试目的 | 验证用户可以修改应用基本信息 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，单击目标应用ID  3.切换到“基本信息”Tab页  4.点击备注后面的按钮进行修改 |
| 预期结果 | 1.应用的备注信息可以修改 |
| 测试结果 | 【TKE纳管TCNP需求】集群、应用管理 基本信息页面编辑描述，修改为空时实际修改未生效 |
| 备注 |  |

删除应用

|  |  |
| --- | --- |
| 用例名称 | 删除应用 |
| 测试目的 | 验证用户可以删除应用 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用  3.应用中没有创建分组和版本包 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”  3.单击需要删除的应用右侧的“删除”按钮 |
| 预期结果 | 1.应用中没有部署组、软件包、镜像时可以删除 |
| 测试结果 | Pass |
| 备注 |  |

容器应用上传镜像

|  |  |
| --- | --- |
| 用例名称 | 容器应用上传镜像 |
| 测试目的 | 验证用户可以在容器应用中上传镜像 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器应用 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，点击目标应用ID  3.在“镜像”页面，点击“使用指引”按钮  图形用户界面, 应用程序  描述已自动生成  4.按弹出的指引信息上传镜像 |
| 预期结果 | 1.镜像上传成功  2.在镜像页面可以查看到新上传的镜像 |
| 测试结果 | Pass |
| 备注 |  |

容器应用创建部署组

|  |  |
| --- | --- |
| 用例名称 | 容器应用创建部署组 |
| 测试目的 | 验证用户可以在容器应用中创建部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器应用  3.用户已经创建容器集群和命名空间 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，点击目标应用ID  3.在“部署组”页面，点击“新建部署组”按钮  4.在弹出的对话框中设置组名，选择集群、命名空间、日志配置项，提交  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.创建容器部署组成功 |
| 测试结果 | Pass |
| 备注 |  |

容器应用部署组操作

|  |  |
| --- | --- |
| 用例名称 | 容器应用部署操作 |
| 测试目的 | 验证用户可以对容器部署组进行部署、启动、停止、扩缩、删除操作 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器部署组  3.用户已经上传docker镜像  4.制作镜像请使用 “验收物料包->通用测试包”路径下包 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用服务”>“应用管理”，点击目标应用ID，切换到部署组页面  图形用户界面, 应用程序  描述已自动生成  点击部署应用操作，在弹框中选择镜像，设置资源限制、实例数、JVM启动参数，查看变更记录  点击停止应用操作，查看变更记录  点击启动应用操作，查看变更记录  点击应用扩缩操作，在弹框中设置实例数，查看变更记录  点击删除操作，查看变更记录 |
| 预期结果 | 1.部署应用成功，部署组下的实例数与设置一致，变更记录中有部署记录  2.停止应用成功，变更记录中有停止记录  3.启动应用成功，变更记录中有启动记录  4.应用扩缩成功，变更记录中有扩容或缩容记录  5.部署组删除成功，变更记录中有销毁记录  命令说明：  kubectl get pod -n 命名空间名称，查询命名空间下的pod  Kubectl get service -n 命名空间名称，查询命名空间下的service |
| 测试结果 | Pass |
| 备注 |  |

服务治理

普通容器应用服务注册

|  |  |
| --- | --- |
| 用例名称 | 普通容器应用服务注册 |
| 测试目的 | 验证普通容器应用服务可以注册到注册中心 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器部署组  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.单击“部署组”，进入部署组列表选择容器集群和命名空间，部署SpringCloud容器应用  IMG_256  3.切换到服务治理页面，选择部署组对应的集群和命名空间，查看微服务状态 |
| 预期结果 | 1.应用运行起来之后，微服务状态为单点在线（单实例）或在线（多实例） |
| 测试结果 | Pass |
| 备注 |  |

服务删除

|  |  |
| --- | --- |
| 用例名称 | 服务删除 |
| 测试目的 | 验证用户可以删除服务 |
| 预置条件 | 用户正常登录平台  用户已经创建和部署服务 |
| 测试过程 | 1.登录TSF控制台  2.进入服务的应用管理页面，停止服务所对应的部署组  3.进入服务治理页面，查看服务状态为离线，点击删除操作  IMG_256 |
| 预期结果 | 1.服务删除成功 |
| 测试结果 | Pass |
| 备注 |  |

服务监控数据查看

|  |  |
| --- | --- |
| 用例名称 | 服务基本信息 |
| 测试目的 | 验证用户可以查看服务基本信息 |
| 预置条件 | 用户正常登录平台  用户已经创建和部署服务 |
| 测试过程 | 1.登录TSF控制台  2.进入服务的服务管理页面，点击微服务名称  图形用户界面, 应用程序, Word  描述已自动生成  3.进入服务，查看服务概览信息如下：  图形用户界面, 应用程序  描述已自动生成  图形用户界面, 应用程序  描述已自动生成  4.服务实例列表tab页，点击实例id查看监控信息如下：  图形用户界面, 应用程序  描述已自动生成  图形用户界面  中度可信度描述已自动生成电脑屏幕截图  描述已自动生成  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  5、查看接口列表信息  图形用户界面, 应用程序, 表格, Excel  描述已自动生成 |
| 预期结果 | 1.可以查看微服务的监控新信息 |
| 测试结果 | Pass |
| 备注 |  |

普通容器应用自定义标签鉴权规则

|  |  |
| --- | --- |
| 用例名称 | 普通容器应用自定义标签鉴权规则 |
| 测试目的 | 验证容器应用支持自定义标签鉴权 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署容器应用服务  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像  4.使用consumer-demo和provider-demo测试，调用关系如下  形状  描述已自动生成 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.单击“服务治理”，进入服务列表页面  3.点击服务列表中的服务id进入服务详情页面，切换到服务鉴权tab页  IMG_256  4.点击编辑按钮，设置服务鉴权，打开鉴权开关，点击“新增自定义标签鉴权”  图形用户界面, 应用程序  描述已自动生成  匹配方式设置为相等，值设置为test，分别使用以下命令访问触发consumer访问provider  命令1 curl ip:18083/echo-rest/str?test  命令2 curl ip:18083/echo-rest/str?ssss  匹配方式设置为不相等，值设置为test，分别使用以下命令访问触发consumer访问provider  命令1 curl ip:18083/echo-rest/str?test  命令2 curl ip:18083/echo-rest/str?ssss  匹配方式设置为包含，值设置为test,分别使用以下命令访问触发consumer访问provider  命令1 curl ip:18083/echo-rest/str?test  命令2 curl ip:18083/echo-rest/str?ssss  8.匹配方式设置为不包含，值设置为test，分别使用以下命令访问触发consumer访问provider  命令1 curl ip:18083/echo-rest/str?test  命令2 curl ip:18083/echo-rest/str?ssss  9.匹配方式设置为正则表达式，值设置为t.\*t,分别使用以下命令访问触发consumer访问provider  命令1 curl ip:18083/echo-rest/str?test  命令2 curl ip:18083/echo-rest/str?Ssss  注：命令中ip为consumer服务的ip，18083位consumer服务端口 |
| 预期结果 | 1.步骤5，命令1成功，命令2失败  2.步骤6，命令1失败，命令2成功  3.步骤7，命令1成功，命令2失败  4.步骤8，命令1失败，命令2失败  5.步骤9，命令1成功，命令2失败 |
| 测试结果 | Pass |
| 备注 |  |

API列表可导出YAML格式

|  |  |
| --- | --- |
| 用例名称 | API列表可导出YAML格式 |
| 测试目的 | 验证API列表可导出YAML格式 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组或容器部署组、集群和命名空间 |
| 测试过程 | 登录TSF控制台，进入TSF控制台  进入服务治理中的app页面，查看页面也是增加了导出API按钮  点击到处API按钮，选择格式为YAML  点击提交，导出API |
| 预期结果 | 1.增加了导出按钮 3.可导出API，格式为YAML |
| 测试结果 | Pass |
| 备注 |  |

单个入参类型调试测试-Path调试测试

|  |  |
| --- | --- |
| 用例名称 | Path调试测试 |
| 测试目的 | API可调试界面 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组或容器部署组、集群和命名空间  3.部署应用请使用“验收物料包->通用测试包”路径下的consumer-demo-1.18.0-Greenwich-SNAPSHOT.jar测试包  4.集群中有应用正在运行 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.进入服务治理中的api页面，查看页面也是增加了导出API按钮  3.选择[/echo-feign-url/{str}](http://106.52.31.105/tsf/service-api-detail?rid=1&id=%2Fecho-feign-url%2F%7Bstr%7D&namespaceId=namespace-jnyg93a2&serviceId=ms-9maery38&method=GET)API，进入调试界面  图形用户界面, 应用程序  描述已自动生成  4.修改Path请求参数中的值，Content-Type选择application/x-www-form-urlencoded,点击调试，然后检查右边的返回结果  5.将步骤3中的Content-Type修改为application/json,再次点击调试，然后检查右边的返回结果 |
| 预期结果 | 1.每个api后面都增加了调试按钮 2.返回结构中包含返回码，响应耗时，响应body，响应headers，并且可以看到path的调试结果 3. 可以看到path的调试 |
| 测试结果 | Pass |
| 备注 |  |

单个入参类型调试测试-Query调试测试

|  |  |
| --- | --- |
| 用例名称 | Query调试测试 |
| 测试目的 | Query可调试界面 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组或容器部署组、集群和命名空间  3.部署应用请使用“验收物料包->通用测试包”路径下的consumer-demo-1.18.0-Greenwich-SNAPSHOT.jar测试包  4.集群中有应用正在运行 |
| 测试过程 | 1.进入服务治理页面中的API列表界面，查看该页面是否增加了调试出API按钮； 2. 选择[/echo-feign-url/{str}](http://106.52.31.105/tsf/service-api-detail?rid=1&id=%2Fecho-feign-url%2F%7Bstr%7D&namespaceId=namespace-jnyg93a2&serviceId=ms-9maery38&method=GET)API，进入调试界面； 3.修改Query请求参数中的值，Content-Type选择application/x-www-form-urlencoded，点击调试，然后检查右边的返回结果； 4.将步骤3中的Content-Type修改为application/json，再次点击调试，然后检查右边的返回结果 |
| 预期结果 | 1.每个api后面都增加了调试按钮 2.返回结构中包含返回码，响应耗时，响应body，响应headers，并且可以看到Query的调试结果 3. 可以看到Query的调试 |
| 测试结果 | Pass |
| 备注 |  |

单个入参类型调试测试-Header调试测试

|  |  |
| --- | --- |
| 用例名称 | Header调试测试 |
| 测试目的 | Header可调试界面 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组或容器部署组、集群和命名空间  3.部署应用请使用“验收物料包->通用测试包”路径下的consumer-demo-1.18.0-Greenwich-SNAPSHOT.jar测试包  4.集群中有应用正在运行 |
| 测试过程 | 1.进入服务治理页面中的API列表界面，查看该页面是否增加了调试出API按钮； 2. 选择[/echo-feign-url/{str}](http://106.52.31.105/tsf/service-api-detail?rid=1&id=%2Fecho-feign-url%2F%7Bstr%7D&namespaceId=namespace-jnyg93a2&serviceId=ms-9maery38&method=GET)API，进入调试界面； 3.修改Header请求参数中的值，Content-Type选择application/x-www-form-urlencoded，点击调试，然后检查右边的返回结果； 4.将步骤3中的Content-Type修改为application/json，再次点击调试，然后检查右边的返回结果 |
| 预期结果 | 1.每个api后面都增加了调试按钮 2.返回结构中包含返回码，响应耗时，响应body，响应headers，并且可以看到Header的调试结果 |
| 测试结果 | Pass |
| 备注 |  |

服务访问

普通容器应用服务访问

|  |  |
| --- | --- |
| 用例名称 | 普通容器应用服务访问 |
| 测试目的 | 验证普通容器应用之间的服务发现 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器部署组、集群和命名空间  3.制作镜像请使用 “验收物料包->通用测试包”路径下包 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像  3.单击“应用管理”，进入容器应用的部署组页面，分别部署consumer、provider容器应用  4.切换到服务治理页面查看部署组对应的服务状态在线 |
| 预期结果 | 1.consumer、provider之间访问正常 |
| 测试结果 | Pass |
| 备注 |  |

Mesh服务和springcloud服务之间访问

|  |  |
| --- | --- |
| 用例名称 | Mesh服务和springcloud服务之间访问 |
| 测试目的 | 验证Mesh服务和springcloud之间的服务发现正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建容器部署组、集群和命名空间  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包的springcloud服务使用consumer-demo、provider-demo  Mesh服务请使用“验收物料包->mesh测试包”路径下的userService.tar.gz测试包  三者调用关系如下：  user——>consumer  user——>provider |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.单击“服务治理”，选择集群和命名空间，新增mesh应用微服务user  3.单击“应用管理”，进入关联应用的部署组页面，部署user应用  4.进入普通应用中部署consumer和provider应用  5.切换到服务治理页面查看部署组对应的服务状态在线  6、使用以下命令触发user和consumer、provider之间的调用  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>[/call/provider](http://106.52.31.105/tsf/service-api-detail?rid=1&id=%2Fcall%2Fprovider&namespaceId=namespace-jnyg93a2&serviceId=ms-opy5pa4q&method=GET)    curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/ [/call/consumer](http://106.52.31.105/tsf/service-api-detail?rid=1&id=%2Fcall%2Fconsumer&namespaceId=namespace-jnyg93a2&serviceId=ms-opy5pa4q&method=GET) |
| 预期结果 | 1.查看consumer、provider日志，调用成功 |
| 测试结果 |  |
| 备注 |  |

应用日志

创建日志配置

|  |  |
| --- | --- |
| 用例名称 | 创建日志配置 |
| 测试目的 | 验证用户可以创建日志配置 |
| 预置条件 | 1.用户正常登录平台  2.用户有创建日志配置的权限 |
| 测试过程 | 1.登录TSF控制台  2.单击“日志配置”，单击“新建配置”按钮  3.在弹出的对话框中，设置“名称”、“日志路径”、“描述”  4.单击日志路径右面的“添加”按钮可以添加多个日志路径  5.在日志路径中使用通配符\*  6.单击“提交” |
| 预期结果 | 1.名称不能超过60字符的长度  2.可以添加多个日志路径  3.描述信息不能超过200字符长度  4.单击提交之后创建日志配置成功，在日志配置页面可以看到新创建的日志配置 |
| 测试结果 | Pass |
| 备注 |  |

日志配置页面操作

|  |  |
| --- | --- |
| 用例名称 | 日志配置页面操作 |
| 测试目的 | 验证用户可以在日志配置页面进行搜索、下载、单击关联应用的操作 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建日志配置  3.日志配置已经被应用关联 |
| 测试过程 | 1.登录TSF控制台  2.单击“日志配置”  3.在右上角的搜索框中输入配置名称，然后回车或者点击搜索按钮  4.单击右侧的下载按钮下载日志配置列表  5.单击日志配置的关联应用 |
| 预期结果 | 1.可以按照日志配置的名称搜索  2.点击下载按钮之后，日志配置列表数据下载保存为csv文件  3.单击日志配置的关联应用名称，页面跳转到应用的“服务实例列表”界面 |
| 测试结果 | Pass |
| 备注 |  |

日志配置删除

|  |  |
| --- | --- |
| 用例名称 | 日志配置删除 |
| 测试目的 | 验证用户可以删除未使用的志配置 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建日志配置  3.日志配置没有被应用关联 |
| 测试过程 | 1.登录TSF控制台  2.单击“日志配置”  3.单击要删除的日志配置右侧的“删除”按钮  4.在弹出的对话框中单击“确认”按钮 |
| 预期结果 | 1.日志配置被成功删除 |
| 测试结果 | Pass |
| 备注 |  |

容器应用日志检索

|  |  |
| --- | --- |
| 用例名称 | 容器应用日志检索 |
| 测试目的 | 验证容器应用日志可以根据关键字检索 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建日志配置  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像 |
| 测试过程 | 1.登录TSF控制台  2.创建日志配置项，日志路径为 /tsf-demo-logs/consumer-demo/root.log  或者/tsf-demo-logs/provider-demo/root.log  3.创建容器应用和部署组，部署consumer、provider应用  4.应用运行一段时间之后，切换到日志检索页，选择集群命名空间，选择不同时间段，选择部署组、实例，输入要搜索的关键字，点击查询  IMG_256 |
| 预期结果 | 1.日志检索正常 |
| 测试结果 | Pass |
| 备注 |  |

调用链

调用链详情

|  |  |
| --- | --- |
| 用例名称 | 调用链详情 |
| 测试目的 | 验证用户可以根据Trace ID查询调用链详情 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署和运行应用  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台  2.单击“调用链查询”，切换到span查询页面  3.在输入框中输入调用链的Trace ID，点击“查询按钮”  4. 步骤3中查询出来的trace，点击查看详情 |
| 预期结果 | 3.点击“查询”按钮，得到调用链的详细信息    4.结果如下 |
| 测试结果 | Pass |
| 备注 |  |

服务依赖拓扑

|  |  |
| --- | --- |
| 用例名称 | 服务依赖拓扑 |
| 测试目的 | 验证可以根据调用链生成服务拓扑 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署和运行应  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  2.单击“服务依赖拓扑”，进入服务拓扑界面  3.选择集群和命名空间  4.设置时间段，查询出服务拓扑  5.鼠标悬浮于服务图标上显示调用链统计信息 |
| 预期结果 | 1.查询出命名空间下该时间段的所有服务拓扑  IMG_256 |
| 测试结果 | Pass |
| 备注 |  |

监控

普通容器应用监控数据

|  |  |
| --- | --- |
| 用例名称 | 普通容器应用监控数据 |
| 测试目的 | 验证普通容器应用监控数据上报正常 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像  3.用户已经部署consumer、provider应用 |
| 测试过程 | 1. consumer、provider自动调用链产生监控数据  2.登录TSF控制台，进入应用管理界面，点击应用列表中的监控图标查看监控数据  3.点击应用id进入应用详情页，切换到监控tab页查看监控数据  4.点击应用id进入应用详情页，在部署组tab页点击监控图标产看监控数据  5.在应用的部署组列表页，点击部署组id进入部署组实例列表，点击监控图标查看监控数据  6.切换到部署组一级页面，选择容器集群和命名空间，点击部署组列表的监控图标，查看监控数据 |
|  | 1. consumer、provider都有监控数据 |
| 测试结果 | Pass |
| 备注 |  |

普通容器应用JVM监控数据

|  |  |
| --- | --- |
| 用例名称 | 普通容器应用JVM监控数据 |
| 测试目的 | 验证普通容器应用JVM监控数据上报正常 |
| 预置条件 | 用户正常登录平台  2.用户已经部署consumer、provider应用 |
| 测试过程 | 1. consumer、provider应用启动产生监控数据  2.登录TSF控制台，点击监控，进入JVM监控，选择命名空间和对应服务，点击查询    点击JVM日志，查看JVM日志    3.点击监控详情，进入基本信息tab页，查看基本信息      切换内存tab页，查看内存信息    切换线程tab页，查看内存信息      点击检测死锁，查看死锁线程信息    切换火焰提tab页，若之前生成过火焰图则可以查出来，若未生成火焰图，则选择数据采集时长，点击开始采集 |
|  | 1. consumer、provider都有JVM监控数据 |
| 测试结果 | Pass |
| 备注 |  |

仓库中心

软件仓库

|  |  |
| --- | --- |
| 用例名称 | 事务查询 |
| 测试目的 | 验证用户可以通过软件仓库查看所有应用的程序包版本 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用和上传版本包 |
| 测试过程 | 登录TSF控制台，切换到软件仓库页面  IMG_256  2.点击应用id跳转到应用的程序包管理界面 |
| 预期结果 | 1.软件仓库显示所有应用的软件包版本个数  2.点击应用id跳转到应用的程序包管理界面 |
| 测试结果 |  |
| 备注 |  |

镜像仓库列表

|  |  |
| --- | --- |
| 用例名称 | 镜像仓库列表 |
| 测试目的 | 用户可以通过镜像仓库查看所有应用镜像仓库 |
| 预置条件 | 1.用户正常登录平台  2.用户已经初始化镜像仓库 |
| 测试过程 | 1.登录TSF控制台，切换到镜像仓库页面  2.点击仓库名跳转到容器应用的镜像页面  IMG_256 |
| 预期结果 | 1.跳转正常 |
| 测试结果 | Pass |
| 备注 |  |

服务编排

新建服务编排模板

|  |  |
| --- | --- |
| 用例名称 | 新建服务编排模板 |
| 测试目的 | 验证用户可以新建服务编排模板来生成工程 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 登录TSF控制台，切换到服务编排界面  点击新建模板按钮，设置工程配置和POM配置  1584584881(1)  添加服务基本信息  IMG_256  设置调用方式和依赖关系  1584585061(1)  5.点击保存并生成工程按钮 |
| 预期结果 | 1.设置依赖关系之后生成依赖关系图  2.点击保存并生成工程之后会将工程下载，用户可以基于这个工程开发自己的业务代码 |
| 测试结果 | Pass |
| 备注 |  |

分布式配置

新建应用配置

|  |  |
| --- | --- |
| 用例名称 | 新建应用配置 |
| 测试目的 | 验证用户可以创建应用配置 |
| 预置条件 | 用户正常登录平台  用户已经创建应用 |
| 测试过程 | 登录TSF控制台，进入应用管理页面  点击应用id进入应用详情页，切换到应用配置页面，点击新建创建应用配置  在新建配置对话框中设置名称、编辑方式、配置内容、版本号、版本描述，点击完成  配置列表页点击预览图标查看配置信息 |
| 预期结果 | 1.创建应用配置成功  2.预览应用配置成功 |
| 测试结果 | Pass |
| 备注 |  |

应用配置生成新版本

|  |  |
| --- | --- |
| 用例名称 | 应用配置生成新版本 |
| 测试目的 | 验证用户可以在老的配置基础上修改内容生成新版本 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用配置 |
| 测试过程 | 1.登录TSF控制台，进入应用管理页面  2.点击应用id进入应用详情页，切换到应用配置页面  3.点击应用的“生成新版本”操作，在弹出的对话框中修改配置内容，设置新的版本号，点击完成 |
| 预期结果 | 1.生成新的版本成功 |
| 测试结果 | Pass |
| 备注 |  |

应用配置删除

|  |  |
| --- | --- |
| 用例名称 | 应用配置删除 |
| 测试目的 | 验证用户可以删除不用的应用配置 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用配置 |
| 测试过程 | 1.登录TSF控制台，进入应用管理页面  2.点击应用id进入应用详情页，切换到应用配置页面  3.应用配置中没有已发布的部署组，点击应用配置的删除操作 |
| 预期结果 | 1.应用配置删除成功 |
| 测试结果 | Pass |
| 备注 |  |

应用配置发布

|  |  |
| --- | --- |
| 用例名称 | 应用配置发布 |
| 测试目的 | 验证用户可以将配置发布到部署组，部署组实时更新配置 |
| 预置条件 | 1.正常登录平台  2.已经创建应用配置  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包  4.已经部署consumer、provider应用（consumer每秒调用一次provider） |
| 测试过程 | 1.正常登录平台  2.已经创建应用配置  3.已经部署consumer、provider应用（consumer每秒调用一次provider）, 创建应用配置内容（v1）如下，将配置发布到provider部署组IMG_256 |
| 预期结果 | 1.步骤3中看到检查部署provider-demo的部署组上，打印的日志中是否 name 的值已更新  provider-demo -- provider config name: testname123 |
| 测试结果 | Pass |
| 备注 |  |

应用配置回滚

|  |  |
| --- | --- |
| 用例名称 | 应用配置回滚 |
| 测试目的 | 验证用户可以在配置发布历史中将配置回滚 |
| 预置条件 | 1.正常登录平台  2.已经创建应用配置  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包  4.已经部署consumer、provider应用（consumer每秒调用一次provider） |
| 测试过程 | 1.登录TSF控制台，进入应用管理页面  2.点击应用id进入应用详情页，切换到应用配置页面  3.创建应用配置内容（v1）如下，将配置发布到provider部署组  IMG_256  4.在步骤3的基础上，生成新的版本（v2），将name设置为test123，将配置发布到provider部署组  5.在配置发布历史中点击v2版本后面的回滚按钮，配置回滚到v1再次执行上述命令 |
| 预期结果 | 1.步骤3中看到检查部署provider-demo的部署组上，打印的日志中是否 name 的值已更新  provider-demo -- provider config name: testname123  2.步骤4看到检查部署provider-demo的部署组上，打印的日志中是否 name 的值已更新  provider-demo -- provider config name: test123 |
| 测试结果 | Pass |
| 备注 |  |

新建全局配置

|  |  |
| --- | --- |
| 用例名称 | 新建全局配置 |
| 测试目的 | 验证用户可以创建全局配置 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 1.登录TSF控制台，进入全局配置页面  2.在配置列表页面点击新建按钮  3.在弹出的对话框中设置名称、编辑方式、配置内容、版本号、版本描述，点击完成  4.配置列表页面点击预览按钮查看配置信息 |
| 预期结果 | 1.创建全局配置成功  2.预览全局配置成功 |
| 测试结果 | Pass |
| 备注 |  |

全局配置删除

|  |  |
| --- | --- |
| 用例名称 | 全局配置删除 |
| 测试目的 | 验证用户可以删除全局配置 |
| 预置条件 | 1.用户正常登录平台  2.已经创建全局配置 |
| 测试过程 | 1.登录TSF控制台，进入全局配置页面  2.点击全局配置的已发布命名空间为空，点击删除操作 |
| 预期结果 | 1.删除成功 |
| 测试结果 | Pass |
| 备注 |  |

全局配置发布

|  |  |
| --- | --- |
| 用例名称 | 全局配置发布 |
| 测试目的 | 验证用户可以将全局配置发布到命名空间，对命名空间下所有部署组生效 |
| 预置条件 | 1.用户正常登录平台  2.已经创建全局配置  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入全局配置页面  2.创建全局配置如下，开启应用的鉴权功能  IMG_256  3.点击全局配置的发布操作，选择要发布的集群和命名空间，设置发布信息，点击提交  4.在命名空间分别创建consumer、provider |
| 预期结果 | 检查部署provider-demo的部署组上，打印的日志中是否 name 的值已更新  provider-demo -- provider config name: testname123 |
| 测试结果 | Pass |
| 备注 |  |

全局配置回滚

|  |  |
| --- | --- |
| 用例名称 | 全局配置回滚 |
| 测试目的 | 验证用户可以将已发布的全局配置回滚到历史版本 |
| 预置条件 | 1.用户正常登录平台  2.已经创建全局配置  3.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入全局配置页面  2.创建全局配置如下：  auth v1  IMG_256  auth v2  IMG_256  3.在命名空间分别创建consumer、provider |
| 预期结果 | 回滚成功 |
| 测试结果 | Pass |
| 备注 |  |

Ribbon配置模板

|  |  |
| --- | --- |
| 用例名称 | Ribbon配置模板 |
| 测试目的 | 验证用户在tsf上创建Ribbon配置模板并且根据模板生成配置 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.进入配置管理的配置模板页面，点击新建模板，设置模板名称，选择类型为Ribbon  IMG_256  2.在配置内容中修改相关参数或新增参数配置  3.填写备注，点击完成  4.在应用配置或者全局配置中点击导入配置模板，选择需要的Ribbon模板，点击提交  IMG_256  5.设置名称、选择关联应用、编辑配置内容、设置版本号和版本描述，点击完成  IMG_256 |
| 预期结果 | 1.步骤1会显示Ribbon的一些常见参数，用户可以修改、删除和添加配置  2.步骤5中会显示模板中的配置内容，用户可以修改、删除和添加配置 |
| 测试结果 | Pass |
| 备注 |  |

自定义配置模板

|  |  |
| --- | --- |
| 用例名称 | 自定义配置模板 |
| 测试目的 | 验证用户在tsf上创建自定义配置模板并且根据模板生成配置 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.进入配置管理的配置模板页面，点击新建模板，设置模板名称，选择类型为自定义  2.在配置内容中添加自定义的参数  1584604744(1)  3.填写备注，点击完成  4.在应用配置或者全局配置中点击导入配置模板，选择需要的自定义模板，点击提交  1584604778(1)  5.设置名称、选择关联应用、编辑配置内容、设置版本号和版本描述，点击完成  1584604808(1) |
| 预期结果 | 1.步骤1中模板的配置内容为空，用户可以添加配置内容  2.步骤5中会显示模板中的自定义配置内容，用户可以修改、删除和添加配置 |
| 测试结果 | Pass |
| 备注 |  |

告警

配置日志告警策略

|  |  |
| --- | --- |
| 用例名称 | 配置日志告警策略 |
| 测试目的 | 验证用户可以创建日志告警策略 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入监控告警页面，切换到告警策略页面，点击新建告警策略    2.设置策略名，选择策略类型为“TSF日志告警”，选择告警对象，设置告警关键词、告警策略、告警接收组和告警渠道，点击提交 |
| 预期结果 | 1.创建告警策略成功 |
| 测试结果 | Pass |
| 备注 |  |

日志告警上报和恢复

|  |  |
| --- | --- |
| 用例名称 | 日志告警上报和恢复 |
| 测试目的 | 验证日志告警上报和恢复正常 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 登录TSF控制台，进入监控告警页面  部署consumer-demo、provider-demo  创建告警策略如下    4.使用如下命令循环调用consumer触发provider产生包含hello的日志，使hello的出现频率达到告警门限  curl -H "Content-type:iapplication/json" http://<consumerdemo实例ip>:18083/echo-rest/hello?tagName=user\&tagValue=test  5.停止调用consumer命令 |
| 预期结果 | 1.步骤4，触发告警，告警列表中可以看到告警记录  2.步骤5，告警恢复 |
| 测试结果 | Pass |
| 备注 |  |

服务指标告警-接收请求量

|  |  |
| --- | --- |
| 用例名称 | 服务指标告警-接收请求量 |
| 测试目的 | 验证可以根据服务接受请求量来产生告警 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入监控告警页面  2.创建部署组consumer-demo、provider-demo  3.创建服务指标告警策略      4.使用如下命令触发consumer调用provider，使请求量大于每分钟100次  curl -H "Content-type:iapplication/json" http://<consumerdemo实例ip>:18083/echo-rest/hello?tagName=user\&tagValue=test  5.调整命令发型频率是consumer调用provider的请求量小于每分钟100次 |
| 预期结果 | 1.步骤4，触发告警，告警列表中可以看到告警记录  2.步骤5，告警恢复 |
| 测试结果 | Pass |
| 备注 |  |

服务指标告警-服务离线

|  |  |
| --- | --- |
| 用例名称 | 服务指标告警-服务离线 |
| 测试目的 | 验证可以根据服务状态产生告警 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包 |
| 测试过程 | 1.登录TSF控制台，进入监控告警页面  2.创建部署组consumer-demo、provider-demo  3.创建服务指标告警策略      4.停止provider部署组，使服务离线  5.启动provider部署组，使服务上线 |
| 预期结果 | 1.步骤4，触发告警，告警列表中可以看到告警记录  2.步骤5，告警恢复 |
| 测试结果 |  |
| 备注 |  |

服务限流

SpringCloud服务全局限流

|  |  |
| --- | --- |
| 用例名称 | SpringCloud服务全局限流 |
| 测试目的 | 验证SpringCloud服务全局限流功能 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包  3.用户已经部署consumer和provider服务 |
| 测试过程 | 1.登录TSF控制台，进入服务治理页面  2.点击provider服务id进入服务详情页，切换到服务限流tab页  3.点击新建限流规则，设置限流规则的主调服务、单位时间、请求数、生效状态，点击提交  图形用户界面, 应用程序  描述已自动生成  4.使用以下命令触发consumer调用provider，频率大于1秒10次  curl -H "Content-type:iapplication/json" http://<consumerdemo实例ip>:18083/echo-rest/hello?tagName=user\&tagValue=test  5.在服务限流页面可以查看被限制请求数  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成 |
| 预期结果 | 1.全局限流成功，每秒只允许10次调用 |
| 测试结果 | Pass |
| 备注 |  |

SpringCloud服务基于自定义标签限流

|  |  |
| --- | --- |
| 用例名称 | SpringCloud服务基于自定义标签限流 |
| 测试目的 | 验证SpringCloud基于自定义标签限流功能 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包  3.用户已经部署consumer和provider服务  4.用户已经创建服务限流规则 |
| 测试过程 | 1.登录TSF控制台，进入服务治理页面  2.点击服务id进入服务详情页，切换到服务限流tab页，创建自定义标签规则  图形用户界面, 应用程序  描述已自动生成  3.使用以下命令触发consumer调用provider，频率大于1秒10次  curl -H "Content-type:iapplication/json" http://<consumerdemo实例ip>:18083/echo-rest/hello?tagName=user\&tagValue=test  4.在服务限流页面可以查看被限制请求数  图形用户界面, 图表, 应用程序, 折线图  描述已自动生成 |
| 预期结果 | 1.满足标签规则的请求达到限流门限，触发限流，否则不限流 |
| 测试结果 | Pass |
| 备注 |  |

Service Mesh服务基于系统标签限流

|  |  |
| --- | --- |
| 用例名称 | Service Mesh服务基于系统标签限流 |
| 测试目的 | 验证Service Mesh服务基于系统标签限流功能 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->mesh测试包”路径下的测试包  3.用户已经部署user和shop服务  4.用户已经创建服务限流规则 |
| 测试过程 | 1.登录TSF控制台，进入服务治理页面  2.点击服务id进入服务详情页，切换到服务限流tab页，创建基于系统标签的限流规则  社交网络的手机截图  描述已自动生成  3.使用以下命令触发user调用shop，频率大于1秒10次  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/api/v6/user/create  4.在服务限流页面可以查看被限制请求数  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.满足标签规则的请求达到限流门限，触发限流，否则不限流 |
| 测试结果 |  |
| 备注 |  |

服务路由

SpringCloud服务系统标签路由

|  |  |
| --- | --- |
| 用例名称 | SpringCloud服务系统标签路由 |
| 测试目的 | 验证SpringCloud服务可以更具系统标签进行路由 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包  3.用户已经部署consumer-demo和provider-demo |
| 测试过程 | 1.登录TSF控制台，进入服务治理页面  2.点击服务id进入服务详情页，切换到服务路由tab页，点击新建路由规则，填写名称  3.流量来源配置中，标签选择系统标签，选择标签名、逻辑关系，设置标签值  图形用户界面  描述已自动生成  标签名可以选择  图形用户界面, 文本, 应用程序  描述已自动生成  逻辑关系可以选择  图形用户界面, 应用程序  描述已自动生成  流量目的地选择所在应用，目的地类型可以选择部署组和版本号，选择对应的部署组和版本号，设置权重，点击完成  开启路由规则生效开关  图形用户界面  中度可信度描述已自动生成  6.使用以下命令触发consumer调用provider  curl -H "Content-type:iapplication/json" http://<consumerdemo实例ip>:18083/echo-rest/hello?tagName=user\&tagValue=test  7.查看流量详情  图表, 折线图  描述已自动生成 |
| 预期结果 | 1.满足标签规则的流量按比例分布到不同的部署组上 |
| 测试结果 | Pass |
| 备注 |  |

Service Mesh服务自定义标签路由

|  |  |
| --- | --- |
| 用例名称 | Service Mesh服务自定义标签路由 |
| 测试目的 | 验证Service Mesh服务按自定义标签路由 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->mesh测试包”路径下的测试包  3.用户已经部署user和shop |
| 测试过程 | 1.登录TSF控制台，进入服务治理页面  2.点击服务id进入服务详情页，切换到服务路由tab页，点击新建路由规则，填写名称  3.流量来源配置中，标签选择自定义标签，设置标签名、选择逻辑关系，设置标签值  图形用户界面, 应用程序  描述已自动生成  逻辑关系可以选择  图形用户界面, 应用程序  描述已自动生成  4.流量目的地选择所在应用，目的地类型可以选择部署组和版本号，选择对应的部署组和版本号，设置权重，点击完成  5.开启路由规则生效开关  图形用户界面, 应用程序  描述已自动生成  6. 使用以下命令触发user调用shop  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/api/v6/user/create  7.查看流量详情  图表, 折线图  描述已自动生成 |
| 预期结果 | 1.满足标签规则的流量按配置的比例分布到不同的部署组上 |
| 测试结果 | Pass |
| 备注 |  |

微服务网关

微服务网关-非单元化



新建微服务网关应用

|  |  |
| --- | --- |
| 用例名称 | 新建微服务网关应用 |
| 测试目的 | 验证用户可新建微服务网关应用 |
| 预置条件 | 1、用户正常登录平台 |
| 测试过程 | 进入应用管理页面  点击新建应用  填写应用名，选择微服务网关应用  图形用户界面, 应用程序  描述已自动生成  点击提交 |
| 预期结果 | 创建成功 |
| 测试结果 | Pass |
| 备注 |  |

上传微服务网关程序包

|  |  |
| --- | --- |
| 用例名称 | 上传微服务网关程序包 |
| 测试目的 | 验证用户可传微服务网关程序包 |
| 预置条件 | 用户正常登录平台  已新建集群  已新建命名空间 |
| 测试过程 | 点击应用id  图形用户界面, 应用程序  描述已自动生成  进入程序包管理tab页，点击上传程序包，路径为“验收物料包->网关测试包”路径下的msgw-zuul1-1.18.0-Finchley-SNAPSHOT.jar测试包  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 程序包上传成功 |
| 测试结果 | Pass |
| 备注 |  |

新建微服务网关部署组

|  |  |
| --- | --- |
| 用例名称 | 新建微服务网关部署组 |
| 测试目的 | 验证用户可新建微服务网关部署组 |
| 预置条件 | 1、用户正常登录平台  2.网关部署应用请使用“验收物料包->网关测试包”路径下的测试包 |
| 测试过程 | 进入部署组页面  点击新建部署组  填写名称，选择命名空间，选择对应的微服务网关应用  点击提交 |
| 预期结果 | 新建成功 |
| 测试结果 | Pass |
| 备注 |  |

微服务网关新建分组

|  |  |
| --- | --- |
| 用例名称 | 新建微服务网关分组 |
| 测试目的 | 验证用户可新建微服务网关分组 |
| 预置条件 | 1、用户正常登录平台 |
| 测试过程 | 进入微服务网关分组页面  点击配置分组与API  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  图形用户界面, 应用程序  描述已自动生成  点击新建分组  图形用户界面, 应用程序  描述已自动生成  填写名称和分组  图形用户界面, 应用程序  描述已自动生成  点击保存 |
| 预期结果 |  |
| 测试结果 | Pass |
| 备注 |  |

网关调用测试-测试无鉴权的api

|  |  |
| --- | --- |
| 用例名称 | 测试API鉴权 |
| 测试目的 | 测试API鉴权 |
| 预置条件 | 1.用户正常登录TSF控制台  2.用户已导入API  3.分组已发布 |
| 测试过程 | 1.登录TSF控制台  2.单击“微服务网关”>分组管理  3.. 在同命名空间找一台能够拼通的微服务网关，进入终端，使用如下脚本  curl -X {GET} -H 'x-mg-traceid:71b2-11e9-acee-5254001-8a237eba' http://{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}  备注：{}中的参数请按需替换 |
| 预期结果 | 步骤3执行成功并返回正确信息 |
| 测试结果 | Pass |
| 备注 |  |

网关调用测试-测试鉴权的api

|  |  |
| --- | --- |
| 用例名称 | 测试鉴权API |
| 测试目的 | 测试鉴权API |
| 预置条件 | 1.用户正常登录TSF控制台  2.用户已导入API  3.分组已发布 |
| 测试过程 | 1.登录TSF控制台  2.单击“微服务网关”>分组管理  3.点击分组id  4.在访问信息列表下新建密钥  5. 在同命名空间找一台能够拼通的微服务网关，进入终端，使用脚本和密钥生成参数  文本  低可信度描述已自动生成  图形用户界面, 文本, 应用程序  描述已自动生成  6.运行如下脚本  curl -X {POST} -H 'content-type: application/json;charset=utf-8' -H 'x-mg-secretid:{secretid}' -H 'x-mg-sign: {sign}' -H 'x-mg-nonce: {nonce}' -H 'x-mg-alg:1' -H 'x-mg-traceid:71b2-11e9-acee-5254001-8a237eba' -d '{具体的报文}' http://{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}  备注：{}中的参数请按需替换 |
| 预期结果 | 步骤6执行成功并返回正确信息 |
| 测试结果 | Pass |
| 备注 |  |

弹性伸缩

容器部署组根据CPU利用率弹性伸缩

|  |  |
| --- | --- |
| 用例名称 | 容器部署组根据CPU利用率弹性伸缩 |
| 测试目的 | 验证容器部署组可以根据CPU利用率弹性伸缩 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像  3.用户已经部署consumer-demo和provider-demo |
| 测试过程 | 1.部署provider-demo服务，实例数为1  2.弹性伸缩页面创建基于cpu利用率的伸缩规则  图形用户界面, 文本, 电子邮件  描述已自动生成  3.点击弹性伸缩规则id在“关键部署组”tab页将弹性伸缩规则关联到部署组provider-1  图形用户界面, 应用程序  描述已自动生成  4.登录到provider-1部署组的实例所在机器上，使用“docker exec -it 容器id /bin/bash”，在容器内执行dd if=/dev/zero of=/dev/null命令，该命令可以将一个cpu核冲到100%，使业务容器的CPU利用率达到扩容门限  文本  描述已自动生成  文本  描述已自动生成  5.等待几分钟查看部署组实例变成2个，变更记录中有扩容成功的记录  图形用户界面, 应用程序  描述已自动生成  图形用户界面, 应用程序  描述已自动生成  6.停止dd命令使CPU利用率低于缩容门限，等待几分钟，部署组provider-1的实例变成1，变更记录中有下线成功的记录  图形用户界面, 应用程序  描述已自动生成  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1、自动扩容正常  2、自动缩容正常  3、扩容达到最大实例数之后，不再扩容  4、缩容达到最小实例数之后，不再缩容 |
| 测试结果 |  |
| 备注 |  |

容器部署组根据内存利用率弹性伸缩

|  |  |
| --- | --- |
| 用例名称 | 容器部署组根据内存利用率弹性伸缩 |
| 测试目的 | 验证容器部署组可以根据内存利用率弹性伸缩 |
| 预置条件 | 1.用户正常登录平台  2.部署应用请使用“验收物料包->通用测试包”路径下的测试包制作的镜像  3.用户已经部署consumer-demo和provider-demo |
| 测试过程 | 部署provider-demo服务，实例数为1，进入provider容器部署的的tsf-agent容器中，在/var/log/tsf/metric目录下查看容器当前的内存利用率  文本  描述已自动生成  文本  描述已自动生成  2.弹性伸缩页面创建基于内存利用率的伸缩规则，设置扩容门限为35%  图形用户界面, 文本  描述已自动生成  3.点击弹性伸缩规则id在“关键部署组”tab页将弹性伸缩规则关联到部署组provider-1  图形用户界面, 应用程序  描述已自动生成  4.等待几分钟查看部署组实例变成3个，变更记录中有两次扩容成功的记录，两次扩容之间的时间间隔大于冷却时间  图形用户界面, 应用程序  描述已自动生成  图形用户界面, 应用程序  描述已自动生成  5.修改弹性伸缩规则，使内存利用率满足缩容条件  图形用户界面, 文本  描述已自动生成  6.等待几分钟之后provider-1部署组缩容成功，实例数变成1，变更记录中有下线记录  图形用户界面, 应用程序  描述已自动生成  图形用户界面  中度可信度描述已自动生成 |
| 预期结果 | 1、自动扩容正常  2、自动缩容正常  3、扩容达到最大实例数之后，不再扩容  4、缩容达到最小实例数之后，不再缩容 |
| 测试结果 |  |
| 备注 |  |

role角色管理

创建角色

|  |  |
| --- | --- |
| 用例名称 | 创建角色 |
| 测试目的 | 用户可创建角色 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 进入角色管理页  点击新建角色  1584609906(1)  3、填写名称，选择权限  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  4、点击提交 |
| 预期结果 | 用户创建角色成功，列表有新建的角色 |
| 测试结果 |  |
| 备注 |  |

编辑角色

|  |  |
| --- | --- |
| 用例名称 | 编辑角色 |
| 测试目的 | 用户可编辑角色 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 1、进入角色管理页  2、点击要修改的角色  1584609980(1)  点击编辑  点击提交 |
| 预期结果 | 编辑角色成功 |
| 测试结果 |  |
| 备注 |  |

删除角色

|  |  |
| --- | --- |
| 用例名称 | 删除role角色 |
| 测试目的 | 用户可删除角色 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 1584610025(1) |
| 预期结果 | 角色已删除 |
| 测试结果 |  |
| 备注 |  |

数据集新建、删除、修改

|  |  |
| --- | --- |
| 用例名称 | 数据集新建、删除、修改 |
| 测试目的 | 验证用户可创建、删除、修改数据集 |
| 预置条件 | 1.用户正常登录平台 |
| 测试过程 | 登录TSF控制台，进入数据集页面  点击新建数据集  1584610058(1)  点击删除  1584610094(1)  点击id、点击编辑  1584610151(1) |
| 预期结果 | 数据集可新建、删除、修改 |
| 测试结果 |  |
| 备注 |  |