|  |
| --- |
|  |



**腾讯微服务平台TSF**

**V1.29测试用例**

文档版本：01

发布日期：2021年02月01日

|  |
| --- |
| 腾讯云计算（北京）有限责任公司 |

**版权说明**

本文档著作权归腾讯云计算（北京）有限责任公司（以下简称“腾讯云”）单独所有，未经腾讯云事先书面许可，任何主体不得以任何方式或理由使用本文档，包括但不限于复制、修改、传播、公开、剽窃全部或部分本文档内容。

本文档及其所含内容均属腾讯云内部资料，并且仅供腾讯云指定的主体查看。如果您非经腾讯云授权而获得本文档的全部或部分内容，敬请予以删除，切勿以复制、披露、传播等任何方式使用本文档或其任何内容，亦请切勿依本文档或其任何内容而采取任何行动。

**商标说明**



“腾讯”、“腾讯云”及其它腾讯云服务相关的商标、标识等均为腾讯云及其关联公司各自所有。若本文档涉及第三方主体的商标，则应依法由其权利人所有。

**免责说明**

本文档旨在向客户介绍本文档撰写时，腾讯云相关产品、服务的当时的整体概况，部分产品或服务在后续可能因技术调整或项目设计等任何原因，导致其服务内容、标准等有所调整。因此，本文档仅供参考，腾讯云不对其准确性、适用性或完整性等做任何保证。您所购买、使用的腾讯云产品、服务的种类、内容、服务标准等，应以您和腾讯云之间签署的合同约定为准，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

修订记录

| 编号 | 发布日期 | 修订人 | 修订内容 |
| --- | --- | --- | --- |
| 01 | 2021-04-10 | P\_xuanqiao | 第一次正式发布。 |

目录

修订记录 ii

目录 iii

1 概览 3

1.1 资源概览 3

1.2 服务运行情况 3

2 集群 5

2.1 创建虚机集群 5

2.2 创建容器集群 5

2.3 查询集群基本信息 6

2.4 修改集群基本信息 7

2.5 删除集群 8

2.6 创建命名空间 9

2.7 删除命名空间 10

2.8 虚机集群导入云主机-脚本安装 11

2.9 容器集群导入云主机 12

2.10 云主机移出容器集群 12

2.11 云主机移出虚机集群 13

3 命名空间 15

3.1 设置命名空间code 15

3.2 命名空间列表页操作 15

4 部署组 17

4.1 创建虚机部署组 17

4.2 部署组详情页 21

4.3 虚机部署组部署应用 22

4.4 虚机部署组重启应用 23

4.5 虚机部署组应用扩容 23

4.6 虚机部署组下线实例 24

4.7 虚机部署组查看日志 24

4.8 虚机部署组删除 25

4.9 创建容器部署组 25

4.10 容器部署组部署应用 26

4.11 容器部署组启动应用 27

4.12 容器部署组停止应用 28

4.13 容器部署组实例操作 29

5 应用管理 30

5.1 创建虚机部署方式应用 30

5.2 创建容器部署方式应用 31

5.3 查询应用基本信息 32

5.4 修改应用基本信息 33

5.5 删除应用 33

5.6 虚机部署方式应用程序包上传 34

5.7 虚机部署方式应用程序包删除 35

5.8 虚机部署方式应用程序包下载 36

5.9 虚机应用创建部署组 36

5.10 虚机应用部署组操作 37

5.11 容器应用上传镜像 38

5.12 容器应用创建部署组 40

5.13 容器应用部署组操作 41

6 服务治理 43

6.1 普通虚机应用服务注册 43

6.2 普通容器应用服务注册 44

6.3 Mesh虚机应用服务注册 44

6.4 Mesh容器应用服务注册 45

6.5 服务删除 46

6.6 服务监控数据查看 46

6.7 普通虚机应用服务名鉴权规则 49

6.8 普通容器应用自定义标签鉴权规则 51

6.9 API列表可导出YAML格式 53

6.10 单个入参类型调试测试-Path调试测试 54

6.11 单个入参类型调试测试-Query调试测试 55

6.12 单个入参类型调试测试-Header调试测试 56

7 服务访问 57

7.1 普通虚机应用服务访问 57

7.2 普通容器应用服务访问 57

7.3 Mesh虚机应用服务访问 58

7.4 Mesh服务和springcloud服务之间访问 59

8 应用日志 61

8.1 创建日志配置 61

8.2 日志配置页面操作 61

8.3 日志配置删除 62

8.4 虚机应用日志检索 63

8.5 容器应用日志检索 64

9 调用链 66

9.1 普通虚机应用调用链 66

9.2 Mesh虚机应用调用链 67

9.3 Mesh容器应用调用链 68

9.4 调用链详情 69

9.5 服务依赖拓扑 71

10 监控 73

10.1 普通虚机应用监控数据 73

10.2 普通容器应用监控数据 74

10.3 普通虚机应用JVM监控数据 75

10.4 普通容器应用JVM监控数据 79

11 仓库中心 84

11.1 软件仓库 84

11.2 镜像仓库列表 84

12 服务编排 86

12.1 新建服务编排模板 86

13 分布式配置 88

13.1 新建应用配置 88

13.2 应用配置生成新版本 88

13.3 应用配置删除 89

13.4 应用配置发布 89

13.5 应用配置回滚 90

13.6 新建全局配置 92

13.7 全局配置删除 92

13.8 全局配置发布 93

13.9 全局配置回滚 94

13.10 Ribbon配置模板 94

13.11 自定义配置模板 96

14 告警 99

14.1 配置日志告警策略 99

14.2 日志告警上报和恢复 101

14.3 服务指标告警-接收请求量 102

14.4 服务指标告警-服务离线 103

15 服务限流 106

15.1 SpringCloud服务全局限流 106

15.2 SpringCloud服务基于自定义标签限流 107

15.3 Service Mesh服务基于系统标签限流 109

16 服务路由 112

16.1 SpringCloud服务系统标签路由 112

16.2 Service Mesh服务自定义标签路由 114

17 微服务网关 117

17.1 微服务网关-非单元化 117

17.1.1 新建微服务网关应用 117

17.1.2 上传微服务网关程序包 118

17.1.3 新建微服务网关部署组 119

17.1.4 微服务网关新建分组 119

17.1.5 网关调用测试-测试无鉴权的api 121

17.1.6 网关调用测试-测试鉴权的api 122

17.2 微服务网关-单元化 124

17.2.1 前端界面测试 124

17.2.1.1 网关列表 124

17.2.1.2 网关列表查询模块 125

17.2.1.3 网关列表刷新模块 125

17.2.1.4 基本信息内容验证 126

17.2.1.5 分组管理内容验证 127

17.2.1.6 API管理 128

17.2.1.7 单元范围 129

17.2.1.8 单元化规则 130

17.2.1.9 新建单元化规则 130

17.2.2 单元化配置 131

17.2.2.1 微服务网关开启单元化 131

17.2.3 单元范围 134

17.2.3.1 导入命名空间 134

17.2.3.2 移除命名空间 135

17.2.4 单元化规则 136

17.2.4.1 单个子规则单个标识 136

17.2.4.2 单个子规则多个标识标识逻辑关系为且 140

17.2.4.3 单个子规则多个标识标识逻辑关系为或 141

17.2.4.4 多个子规则单个标识 142

17.2.4.5 多个子规则多个标识逻辑关系为且 143

17.2.4.6 多个子规则多个标识逻辑关系为或 144

17.2.4.7 多条单元化规则只生效一条 146

17.2.4.8 单元化规则关闭 147

17.2.4.9 单元化规则删除 148

17.2.4.10 跨命名空间-多个子规则单个标识 149

17.2.4.11 跨命名空间-多个子规则多个标识逻辑关系为且 151

17.2.4.12 跨命名空间-多个子规则多个标识逻辑关系为或 153

17.2.4.13 多个子规则tag相同而目的命名空间不同 155

17.2.5 异常场景 157

17.2.5.1 开启单元后如果没有配置单元化规则，访问报错 157

17.2.5.2 开启单元后配置单元化规则，但是未携带key访问不通 158

17.2.5.3 开启单元化后原非单元化方式无法访问 159

17.2.6 分组管理 160

17.2.6.1 绑定单个部署组 160

17.2.6.2 绑定多个部署组 162

17.2.6.3 移除部署组 163

17.2.6.4 微服务网关密钥对鉴权 164

17.2.6.5 微服务网关绑定TAG插件 166

17.2.7 API管理 168

17.2.7.1 部分API导入 168

17.2.7.2 全部API导入 170

17.2.7.3 网关服务API级别超时时间 172

17.2.7.4 网关服务API级别限流时间 173

17.2.8 重定向配置 174

17.2.8.1 网关服务API重定向 174

17.2.9 服务治理 176

17.2.9.1 服务路由 176

17.2.9.2 服务限流 177

17.2.9.3 服务熔断 178

17.2.9.4 服务鉴权 179

18 弹性伸缩 181

18.1 虚机部署组根据响应时间弹性伸缩 181

18.2 虚机部署组根据请求QPS弹性伸缩 183

18.3 容器部署组根据CPU利用率弹性伸缩 186

18.4 容器部署组根据内存利用率弹性伸缩 189

19 role角色管理 193

19.1 创建角色 193

19.2 编辑角色 194

19.3 删除角色 195

19.4 数据集新建、删除、修改 196

20 原生应用 198

20.1 consul服务 198

20.1.1 服务注册 198

20.1.2 服务治理监控 199

20.1.3 接口列表管理 200

20.1.4 服务鉴权 202

20.1.5 服务路由 205

20.1.6 服务限流 209

20.1.7 服务熔断 212

20.2 Eureka服务 214

20.3 Gateway服务 214

20.3.1 服务注册 214

20.3.2 服务治理监控 215

20.3.3 服务鉴权 216

20.3.4 服务路由 219

20.3.5 服务限流 222

20.3.6 服务熔断 226

20.4 Zuul服务 227

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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

虚机集群导入云主机-脚本安装

|  |  |
| --- | --- |
| 用例名称 | 虚机集群导入云主机-重装系统 |
| 测试目的 | 验证用户可以给虚机集群导入云主机 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建好集群 |
| 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.在弹出的对话框中复制安装脚本，在需要导入的机器执行安装脚本IMG_256 |
| 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.安装脚本执行完成之后，云主机状态为“运行中”，可用状态为“可用1” |
| 测试结果 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

部署组

创建虚机部署组

|  |  |
| --- | --- |
| 用例名称 | 创建虚机部署组 |
| 测试目的 | 验证用户可以创建虚机部署组 |
| 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，进入应用管理页面  2.点击虚机应用id，在部署组列表页创建部署组  3.设置部署组名、命名空间、选择关联应用、日志配置项，点击“创建部署组并导入云主机”  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  给部署组添加云主机  图形用户界面, 文本, 应用程序  描述已自动生成  5.选择版本包，点击完成，部署应用  图形用户界面, 应用程序  描述已自动生成  6.部署组一级页面，点击新建部署组，设置部署组名、集群、命名空间、关联应用、日志配置项，点击“创建部署组并导入云主机”  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  7.添加云主机到部署组，点击部署应用  图形用户界面, 文本, 应用程序  描述已自动生成  8.选择程序包，点击完成，开始部署应用  图形用户界面, 应用程序, 电子邮件  描述已自动生成 |
| 预期结果 | 1.部署应用成功 |
| 测试结果 |  |
| 备注 |  |

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 |  |
| 备注 |  |

虚机部署组部署应用

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组部署应用 |
| 测试目的 | 验证用户可以通过虚机部署组部署应用 |
| 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.点击部署组的部署应用按钮，在弹出的对话框中选程序包，提交 |
| 预期结果 | 1.部署组部署应用成功 |
| 测试结果 |  |
| 备注 |  |

虚机部署组重启应用

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组启动应用 |
| 测试目的 | 验证用户可以通过虚机部署组重启应用 |
| 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“部署组”，选择集群和命名空间  3.点击部署组的“重启应用”操作 |
| 预期结果 | 1.部署组下的所以实例重启成功 |
| 测试结果 |  |
| 备注 |  |

虚机部署组应用扩容

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组应用扩容 |
| 测试目的 | 验证用户可以扩容虚机部署组 |
| 预置条件 | 1.用户正常登录平台  2.部署组运行中 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择集群和命名空间  3.点击部署组的“应用扩容”操作，在弹出的对话框中选择云主机 |
| 预期结果 | 1.部署组扩容成功 |
| 测试结果 |  |
| 备注 |  |

虚机部署组下线实例

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组下线实例 |
| 测试目的 | 验证用户可以给虚机部署组下线实例 |
| 预置条件 | 1.用户正常登录平台  2.部署组运行中 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择集群和命名空间  3.点击部署组的“下线实例”操作，弹框中点击提交 |
| 预期结果 | 1.部署组下所有实例下线 |
| 测试结果 |  |

虚机部署组查看日志

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组查看日志 |
| 测试目的 | 验证用户可以查看虚机部署组日志详情 |
| 预置条件 | 1.用户正常登录平台  2.部署组运行中 |
| 测试过程 | 1.登录TSF控制台  2.单击“部署组”，选择集群和命名空间  3.点击部署组的“查看日志”操作 |
| 预期结果 | 1.可以查看部署组“stdout日志”  2.可以查看日志配置项采集的日志  3.开启“自动刷新”按钮每隔两秒刷新一次；关闭“自动刷新”按钮会停止日志刷新，可以选择时间段查看日志； |
| 测试结果 |  |

虚机部署组删除

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组删除 |
| 测试目的 | 验证用户可以给虚机部署组删除 |
| 预置条件 | 1.用户正常登录平台  2.部署组中所有实例已下线 |
| 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.点击部署组的“删除”操作，弹框中点击确认 |
| 预期结果 | 1.部署组删除成功 |
| 测试结果 |  |

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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

应用管理

创建虚机部署方式应用

|  |  |
| --- | --- |
| 用例名称 | 创建虚机部署方式应用 |
| 测试目的 | 验证用户可以创建虚机部署方式的应用 |
| 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”选择虚拟机部署  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 |  |
| 备注 |  |

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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

虚机部署方式应用程序包上传

|  |  |
| --- | --- |
| 用例名称 | 程序包上传 |
| 测试目的 | 验证用户可以上传程序包 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机应用 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用管理”，点击需要上传程序包的应用ID  3.切换到“程序包管理”页面，点击“上传程序包”按钮  4.在弹出的对话框中选择要上传的程序包、设置版本号和备注，提交上传  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.只能上传.jar、tar.gz、zip格式的程序包  2.版本号可以手动填写，也可以使用时间戳  3.备注信息超度不能超过200字符  4.上传程序包成功 |
| 测试结果 |  |
| 备注 |  |

虚机部署方式应用程序包删除

|  |  |
| --- | --- |
| 用例名称 | 虚机部署方式应用程序包删除 |
| 测试目的 | 验证用户可以删除程序包 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用  3.用户已经上传程序包  4.程序包没有被部署 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用管理”，点击需要删除程序包的应用ID  3.切换到“程序包管理”页面，点击要删除程序包右侧的“删除”按钮，进行删除  IMG_256 |
| 预期结果 | 1.删除程序包成功 |
| 测试结果 |  |
| 备注 |  |

虚机部署方式应用程序包下载

|  |  |
| --- | --- |
| 用例名称 | 虚机部署方式应用程序包下载 |
| 测试目的 | 验证用户可以下载程序包 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用  3.用户已经上传程序包 |
| 测试过程 | 1.登录TSF控制台  2.单击“应用管理”>“程序包管理”，点击程序包的下载按钮  IMG_256 |
| 预期结果 | 1.下载程序包成功 |
| 测试结果 |  |
| 备注 |  |

虚机应用创建部署组

|  |  |
| --- | --- |
| 用例名称 | 虚机应用创建部署组 |
| 测试目的 | 验证用户可以创建虚机应用部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机应用 |
| 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“实例部署管理”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”  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.创建分组成功 |
| 测试结果 |  |
| 备注 |  |

虚机应用部署组操作

|  |  |
| --- | --- |
| 用例名称 | 应用部署操作 |
| 测试目的 | 验证用户可以对虚机部署进行部署应用、重启应用、停止应用、应用扩容、下线实例、添加实例、删除操作、 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建应用  3.用户已经创建应用分组 |
| 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.对部署组进行应用部署操作，查看变更记录  4.对部署组进行添加实例操作，查看变更记录  5.对部署组进行停止应用操作，查看变更记录  6.对部署组进行启动应用操作，查看变更记录  7.对部署组进行应用扩容操作，查看变更记录  8.对部署组进行下线实例操作，查看变更记录  9.对部署组进行删除操作 |
| 预期结果 | 1部署成功，变更记录中有部署记录  2.添加实例成功，变更记录中有扩容记录  3.停止应用成功，变更记录中有停止记录  4.启动应用成功，变更记录中有启动记录  5.应用扩容成功，变更记录中有扩容记录  6.下线实例成功，变更记录中有下线记录  7.删除成功 |
| 测试结果 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

服务治理

普通虚机应用服务注册

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用服务注册 |
| 测试目的 | 验证普通虚机应用服务可以注册到注册中心 |
| 预置条件 | 用户正常登录平台  2.用户已经创建普通虚机部署组  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 |
| 测试过程 | 1.登录TSF控制台，进入TSF控制台  IMG_2562.单击“部署组”，进入部署组列表，选择虚机集群和部署组部署SpringCloud应用  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）  IMG_256 |
| 测试结果 |  |
| 备注 |  |

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 |  |
| 备注 |  |

Mesh虚机应用服务注册

|  |  |
| --- | --- |
| 用例名称 | Mesh虚机应用服务注册 |
| 测试目的 | 验证Mesh虚机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Mesh虚机应用  3.部署应用请使用“验收物料包->mesh测试包”路径下的测试包 |
| 测试过程 | 登录TSF控制台，进入TSF控制台  2.进入应用管理界面，选择mesh虚机应用，创建部署组，部署Mesh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 |  |
| 备注 |  |

Mesh容器应用服务注册

|  |  |
| --- | --- |
| 用例名称 | Mesh容器应用服务注册 |
| 测试目的 | 验证Mesh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Mesh容器应用  3.部署应用请使用“验收物料包->mesh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.进入应用管理界面，选择mesh容器应用，创建部署组，部署mesh应用  图形用户界面, 应用程序  描述已自动生成  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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

普通虚机应用服务名鉴权规则

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用服务名鉴权规则 |
| 测试目的 | 验证虚机应用支持服务名鉴权 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署虚机应用服务  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  4.使用consumer-demo和provider-demo测试，调用关系如下，consumer会自动调用provider  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“新增服务鉴权”  图形用户界面, 应用程序, Teams  描述已自动生成  5.匹配方式设置为包含，值设置为consumer-demo，查看consumer和provider的日志  6.匹配方式设置为不包含，值设置为consumer-demo，查看consumer和provider的日志 |
| 预期结果 | 1.步骤5，consumer访问provider成功，日志中有接口调用成功打印  2.步骤6，consumer访问provider失败，日志中有接口调用失败打印 |
| 测试结果 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

服务访问

普通虚机应用服务访问

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用服务访问 |
| 测试目的 | 验证普通虚机应用之间的服务发现 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组、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 |
| 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  3.单击“应用管理”，进入虚机应用的部署组页面，分别部署consumer、provider虚机应用  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 |  |
| 备注 |  |

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 |  |
| 备注 |  |

Mesh虚机应用服务访问

|  |  |
| --- | --- |
| 用例名称 | Mesh虚机应用服务访问 |
| 测试目的 | 验证Mesh虚机应用之间的服务发现 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建虚机部署组、集群和命名空间  Mesh服务的Python demo应用部署时请使用“验收物料包->mesh测试包”路径下的测试包：  三者调用关系如下： |
| 测试过程 | 登录TSF控制台，进入TSF控制台  单击“服务治理”，选择集群和命名空间，新增mesh应用微服务user、shop、promotion，关联应用  3.单击“应用管理”，进入关联虚机应用的部署组页面，分别部署user、shop、promotion虚机应用  4.切换到服务治理页面查看部署组对应的服务状态在线  5.使用以下命令触发user、shop、promotion之间的调用  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/api/v6/user/account/query |
| 预期结果 | 1.命令执行成功，user、shop、promotion之间访问正常 |
| 测试结果 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

虚机应用日志检索

|  |  |
| --- | --- |
| 用例名称 | 虚机应用日志检索 |
| 测试目的 | 验证虚机应用日组织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 |
| 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  3.创建虚机应用和部署组，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 |  |
| 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 |  |
| 备注 |  |

调用链

普通虚机应用调用链

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用调用链 |
| 测试目的 | 验证普通虚机应用调用链采集查询正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署consumer、provider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.consumer、provider自动调用触发调用链  2.登录TSF控制台，进入调用链查询页面    选择时间段、起点服务、起点接口，点击查询  勾选仅查询出错的调用链，点击查询  设置耗时，点击查询 |
| 预期结果 | 1.步骤3，查询满足条件的调用链  2.步骤4，仅查询出错日志  3.步骤5，查询出耗时大于设置值得调用链 |
| 测试结果 |  |
| 备注 |  |

Mesh虚机应用调用链

|  |  |
| --- | --- |
| 用例名称 | Mesh虚机应用调用链 |
| 测试目的 | 验证Mesh虚机应用调用链采集查询正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署Mesh应用demo：user、shop、promotion  3.部署应用请使用“验收物料包->mesh测试包”路径下的测试包 |
| 测试过程 | 1. 使用以下命令触发user、shop、promotion之间的调用  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/api/v6/user/account/query  2.登录TSF控制台，进入调用链查询页面    3.选择时间段、命名空间、起点服务，点击查询  4.勾选仅查询出错的调用链，点击查询  5.设置耗时，点击查询  6.设置客户端和服务端ip，点击查询 |
| 预期结果 | 1.步骤3，查询满足条件的调用链  2.步骤4，仅查询出错日志  3.步骤5，查询出耗时大于设置值得调用链  4.步骤6，查询出设置ip的调用链 |
| 测试结果 |  |
| 备注 |  |

Mesh容器应用调用链

|  |  |
| --- | --- |
| 用例名称 | Mesh容器应用调用链 |
| 测试目的 | 验证Mesh容器应用调用链采集查询正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署Mesh容器应用demo：user、shop、promotion  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. 使用以下命令触发user、shop、promotion之间的调用  curl -H "Content-type:application/json" http:// <user实例ip>: <user端口>/api/v6/user/account/query  2.登录TSF控制台，进入调用链查询页面    3.选择时间段、命名空间、调用服务/调用接口、被调服务/被调接口，点击查询  4.勾选仅查询出错的调用链，点击查询  5.设置耗时，点击查询  6.设置客户端和服务端ip，点击查询 |
| 预期结果 | 1.步骤3，查询满足条件的调用链  2.步骤4，仅查询出错日志  3.步骤5，查询出耗时大于设置值得调用链  4.步骤6，查询出设置ip的调用链 |
| 测试结果 |  |
| 备注 |  |

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 |  |
| 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 |  |
| 备注 |  |

监控

普通虚机应用监控数据

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用监控数据 |
| 测试目的 | 验证普通虚机应用监控数据上报正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经部署consumer、provider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. consumer、provider自动调用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  IMG_256  2.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  IMG_256  3.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 |
|  | 1. consumer、provider都有监控数据 |
| 测试结果 |  |
| 备注 |  |

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 |  |
| 备注 |  |

普通虚机应用JVM监控数据

|  |  |
| --- | --- |
| 用例名称 | 普通虚机应用JVM监控数据 |
| 测试目的 | 验证普通虚机应用JVM监控数据上报正常 |
| 预置条件 | 1.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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 备注 |  |

微服务网关-单元化



前端界面测试

网关列表

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 网关管理tab页信息查看 |
| 测试目的 | 验证用户可以查看网关管理tab页列表信息 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单 |
| 预期结果 | 1.网关列表包括名称.运行状态.所属命名空间.在线实例/总实例数.单元化部署.操作等字段  2.可以拉取到所有创建好的网关应用  3.名称网关ID可点击跳转到基本信息页面  4.运行状态与预期一致  5.所属命名空间可点击跳转到命名空间基本信息页面  6.在线实例数/总实例数与预期一致  7.单元化部署状态与预期一致，默认存量的数据都应该为未启用状态  8.分页功能正常，可以按照10/20/50分页查询展示数据  9.运行状态可选择过滤，过滤后结果与预期一致  10.命名空间可选择过滤，过滤后结果与预期一致  11.单元化部署可选择过滤，过滤后结果与预期一致 |
| 测试结果 |  |
| 备注 |  |

网关列表查询模块

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P2 |
| 用例名称 | 网关列表查询 |
| 测试目的 | 验证用户可以在网关管理搜索框查询网关信息 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.输入服务名称搜索查询 |
| 预期结果 | 如果匹配到应该只查到相关网关，否则提示无数据 |
| 测试结果 |  |
| 备注 |  |

网关列表刷新模块

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P2 |
| 用例名称 | 网关列表刷新模块 |
| 测试目的 | 验证网关列表刷新模块可刷新 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，对其所在部署组进行扩容1台  3.刷新网关管理页面刷新按钮 |
| 预期结果 | 1.在线实例/总实例数与预期扩容后一致 |
| 测试结果 |  |
| 备注 |  |

基本信息内容验证

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 网关基本信息内容验证 |
| 测试目的 | 验证用户可以查看网关基本信息 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.默认切换到基本信息Tab |
| 预期结果 | 1.网关详情Tab栏位包括基本信息.分组管理.API管理.路径重定向.单元范围.单元化规则  2.基本信息栏存在服务名称.所属命名空间.在线实例.总实例数与预期结果一致  3.基本信息栏存在【查看实例】可跳转到网关所在部署组实例  4.基本信息栏存在【配置服务治理】可跳转到网关服务的服务治理页面  5.单元化配置栏位存在单元化部署字段.结果为启用/未启用状态  6.单元化配置栏位存在【编辑】按钮  7.部署组栏位列表存在名称.运行状态.所在应用.所在集群.在线实例/总实例数.负载均衡IP/服务IP.操作等字段部署组栏位列表存在名称.运行状态.所在应用.所在集群.在线实例/总实例数.负载均衡IP/服务IP.操作等字段  8.部署组ID可跳转后部署组详情页面  9.应用ID可跳转后应用详情页面  10.集群ID可跳转后集群详情页面 |
| 测试结果 |  |
| 备注 |  |

分组管理内容验证

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 分组管理内容验证 |
| 测试目的 | 验证分组管理内容可以正常使用 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.切换到分组管理Tab |
| 预期结果 | 1.显示【新建分组】按钮，点击后可弹出新建分组弹出框（与原功能一致）  2.分组列表包括分组ID/名称.访问Context.鉴权类型.发布状态.创建时间.绑定网关部署组.操作，相关字段与创建的分组信息一致  3.分组ID可跳转到分组详细信息，与预期结果一致  4.网关部署组ID可跳转到相关部署组页面  5.点击发布可执行发布操作，返回失败或者成功  6.点击绑定部署组可选择部署组进行部署  7.搜索功能可以按照分组ID搜索正常  8.分页功能按照10/20/50翻页正常 |
| 测试结果 |  |
| 备注 |  |

API管理

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | API管理 |
| 测试目的 | 验证用户可以正常使用API管理页面内容 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.切换到API管理Tab |
| 预期结果 | 1.验证原API管理功能一致  2.API可以通过选择不同分组ID过滤 |
| 测试结果 |  |
| 备注 |  |

单元范围

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 单元范围 |
| 测试目的 | 验证用户可以正常使用单元范围内容 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.切换到单元组Tab |
| 预期结果 | 1.新增关联命名空间按钮  2.列表包括命名空间ID/名称.操作字段  3.分页功能按照10/20/50翻页正常  4.可以按照命名空间/ID搜索  5.点击【移除】可以移除命名空间  6.点击【关联命名空间】弹出命名空间.点击【保存】，可增加命名空间 |
| 测试结果 |  |
| 备注 |  |

单元化规则

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 单元化规则 |
| 测试目的 | 验证用户可以正常使用单元化规则页面内容 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.切换到单元规则Tab |
| 预期结果 | 1.新增【新建】按钮，点击可进入新增单元化规则页面  2.列表包括名称.备注.生效状态.操作字段  3.点击名称可进入规则详情  4.生效状态可以通过按钮操作  5.可点击【编辑】进入编辑页面  6.可点击删除删除规则 |
| 测试结果 |  |
| 备注 |  |

新建单元化规则

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P |
| 用例名称 | 新建单元化规则 |
| 测试目的 | 验证用户可以新建单元化规则 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关 |
| 测试过程 | 1.点击【微服务网关】进入【网关管理】菜单  2.选择一个微服务网关，点击服务网关ID  3.切换到单元规则Tab  4.点击【新建】 |
| 预期结果 | 1.基本信息包括名称.备注可输入  2.标识配置包括Key位置.Key名称.逻辑关系.值  3.最多可增加15个配置  4.配置可删除  5.规则可删除  6.目的命名空间只在单元组范围内  7.可以增加多条规则  8.点击【单元组】可跳转到单元组页面 |
| 测试结果 |  |
| 备注 |  |

单元化配置

微服务网关开启单元化

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P |
| 用例名称 | 微服务网关开启单元化 |
| 测试目的 | 验证用户可以开启网关单元化 |
| 预置条件 | 无 |
| 测试过程 | 1. 用户正常登录租户端平台  2. 集群页面，新建集群并导入机器    3. 给该集群关联全局命名空间（单元化网关部署需要部署在全局命名空间下）    应用管理页面，新建虚拟机网关应用并上传程序包    应用信息界面，选择新建部署组  填写部署组信息（命名空间选择全局名空间），完成部署    进入网关管理列表页，点击网关id    点击编辑单元化配置，开启单元化 |
| 预期结果 | 1.网关单元化开启成功 |
| 测试结果 |  |
| 备注 |  |

单元范围

导入命名空间

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 导入非全局命名空间 |
| 测试目的 | 验证用户单元范围可以导入非全局命名空间 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关  3.网关已开启单元化 |
| 测试过程 | 1.网关管理，进入单元范围tab页  2.点击关联命名空间  3.选择单个命名空间导入  4.选择多个命名空间导入  5.导入全局命名空间 |
| 预期结果 | 1.可以导入单个非全局命名空间  2.可以导入多个非全局命名空间  3.全局命名空间置灰不可导入 |
| 测试结果 |  |
| 备注 |  |

移除命名空间

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 移除命名空间 |
| 测试目的 | 验证用户可以在单元范围移除命名空间 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关，单元范围已导入命名空间 |
| 测试过程 | 1.网关管理，进入单元范围tab页  2.点击命名空间右侧的【移除】    3.再次导入命名空间，并配置单元化规则  4.再次点击【移除】 |
| 预期结果 | 1.步骤2，未配置单元化规则的命名空间可移除  2.步骤4，配置了单元化规则的命名空间不可移除 |
| 测试结果 |  |
| 备注 |  |

单元化规则

单个子规则单个标识

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 单个子规则单个标识 |
| 测试目的 | 验证单元化规则下只有一个子规则一个标识可以生效 |
| 预置条件 | 1.用户正常登录租户端平台  2.部署网关服务（部署组在全局ns）.部署spring cloud服务consumer(部署在ns1).provider（部署在ns1)    创建分组并导入api（consumer服务下的/echo-rest-unit/{str}）    发布分组 |
| 测试过程 | 单元范围导入命名空间ns1    tag名称和值设置为userId:123    3.逻辑关系选择等于  4.目的命名空间设置为ns1  5.进入后台网关机器发起请求  curl -H "userId:123" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123> #121.4.168.46:8080为网关IP和端口，/test1/consumer-demo/echo-rest-unit/{str}为api管理处复制出来的路径  6.查看请求结果  7.返回单元化规则切换逻辑关系为不等于  查看请求结果  9.重复步骤7和步骤8，分别测试逻辑关系包含（值填写123,abc,test3).不包含（值填写123,abc,test3).正则表达式（值填写【0-9】） |
| 预期结果 | 1.子规则的逻辑关系命中等于请求成功，命中不等于请求失败，命中包含请求成功，命中不包含请求失败，正则命中请求成功  请求成功：    请求失败： |
| 测试结果 |  |
| 备注 |  |

单个子规则多个标识标识逻辑关系为且

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 单个子规则多个标识标识逻辑关系为且 |
| 测试目的 | 验证单元化规则下一个子规则多个标识逻辑关系为且可以生效 |
| 预置条件 | 1.用户正常登录租户端平台  2.部署网关（全局ns）.consumer(ns1).provider（ns1)—部署方式参考用例2.2.4.1  3.创建分组并导入api发布 |
| 测试过程 | 单元范围导入命名空间ns1  设置单元化规则    后台请求网关：  curl -H "userId:123" -H "age:aaa" http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123 |
| 预期结果 | 1.标识逻辑关系为且，请求需要同时命中多个标识才会请求成功 |
| 测试结果 |  |
| 备注 |  |

单个子规则多个标识标识逻辑关系为或

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 |  |
| 测试目的 | 验证单元化规则下一个子规则多个标识逻辑关系为或可以生效 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.创建分组并导入api发布 |
| 测试过程 | 1.单元范围导入命名空间ns1  2.配置单元化规则    3.后台请求网关：  curl -H "age:aaa" http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123 |
| 预期结果 | 标识逻辑关系为或，请求只需要命中一个标识就可以请求成功 |
| 测试结果 |  |
| 备注 |  |

多个子规则单个标识

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 多个子规则单个标识 |
| 测试目的 | 验证单元化规则下多个子规则单个标识可以生效 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.创建分组并导入api发布 |
| 测试过程 | 1.单元范围导入命名空间ns1  2.填写单元化规则，设置多个子规则    后台请求网关  curl -H "age:ccc" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123>  curl -H "userId:abc" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123>  curl -H "userId:abcd" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123>（未命中） |
| 预期结果 | 请求按多个子规则顺序匹配，命中任意子规则则请求成功，未命中子规则则请求失败 |
| 测试结果 |  |
| 备注 |  |

多个子规则多个标识逻辑关系为且

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 多个子规则多个标识逻辑关系为且 |
| 测试目的 | 验证单元化规则下多个多个子规则多个标识逻辑关系为且可以生效 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.创建分组并导入api发布 |
| 测试过程 | 1.单元范围导入命名空间ns1  填写单元化规则，设置多个子规则.多个标识    3.后台请求网关  curl -H "age:ccc" -H "city:shanghai" http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123 |
| 预期结果 | 1.请求按多个子规则顺序匹配，命中任意子规则则请求成功，未命中子规则则请求失败 |
| 测试结果 |  |
| 备注 |  |

多个子规则多个标识逻辑关系为或

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 多个子规则多个标识逻辑关系为或 |
| 测试目的 | 验证单元化规则下多个多个子规则多个标识逻辑关系为或可以生效 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.创建分组并导入api发布 |
| 测试过程 | 1.单元范围导入命名空间ns1  2.填写单元化规则，设置多个子规则.多个标识    后台请求网关  curl -H "age:ccc" -H "city:shanghai" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123>  curl "city:shanghai" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/123> |
| 预期结果 | 1.请求按多个子规则顺序匹配，命中任意子规则则请求成功，未命中子规则则请求失败 |
| 测试结果 |  |
| 备注 |  |

多条单元化规则只生效一条

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P2 |
| 用例名称 | 多条单元化规则只生效一条 |
| 测试目的 | 验证网关单元化下多条单元化规则只生效一条 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.创建分组并导入api发布 |
| 测试过程 | 1.单元范围导入命名空间ns1  2.设置多条单元化规则    3.请求命中状态已开启的单元化规则  请求命中状态未开启的单元化规则 |
| 预期结果 | 请求命中状态以开启的单元化规则，请求成功  请求命中状态未开启的单元化规则，请求失败 |
| 测试结果 |  |
| 备注 |  |

单元化规则修改

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 单元化规则修改 |
| 测试目的 | 验证用户可以修改单元化规则 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.已创建分组并导入api发布  3.单元化规则已生效 |
| 测试过程 | 进入单元化规则tab页  点击规则【编辑】，修改规则    后台请求网关 |
| 预期结果 | 1.请求命中更改后的规则，请求成功 |
| 测试结果 |  |
| 备注 |  |

单元化规则关闭

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 单元化规则关闭 |
| 测试目的 | 验证用户可以关闭单元化规则 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.已创建分组并导入api发布  3.单元化规则已生效 |
| 测试过程 | 进入单元化规则tab页  把单元化规则状态改为关闭    3.后台请求网关 |
| 预期结果 | 1.请求命中规则，但规则已关闭所以请求失败 |
| 测试结果 |  |
| 备注 |  |

单元化规则删除

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 单元化规则删除 |
| 测试目的 | 验证用户可以删除单元化规则 |
| 预置条件 | 1.部署网关（全局ns）.consumer(ns1).provider（ns1)  2.已创建分组并导入api发布 |
| 测试过程 | 进入单元化规则tab页  点击删除开启状态的单元化规则    点击删除关闭状态的单元化规则 |
| 预期结果 | 删除开启状态下的单元化规则，删除失败  删除关闭状态下的单元化规则，删除成功 |
| 测试结果 |  |
| 备注 |  |

跨命名空间-多个子规则单个标识

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 跨命名空间-多个子规则单个标识 |
| 测试目的 | 验证跨命名空间服务下单元化规则可以生效 |
| 预置条件 | 部署网关（全局ns）.consumer1(ns1).provider2（ns1).consumer2(ns2).provider2(ns2)    2.已创建分组并导入api发布 |
| 测试过程 | 进入单元化规则tab页  设置单元化规则，使请求落到consumer1再落到provider2    后台请求网关  curl -H "city:test1" -H "userId:test2" http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2  4.后台查看provider1.provider2实时日志 |
| 预期结果 | 请求成功后，provider1日志未收到请求信息，provider2日志可以收到请求信息  网关请求成功打印：    Provider1日志无相关信息打印：    Provider2日志： |
| 测试结果 |  |
| 备注 |  |

跨命名空间-多个子规则多个标识逻辑关系为且

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 跨命名空间-多个子规则多个标识逻辑关系为且 |
| 测试目的 | 验证跨命名空间服务下单元化规则可以生效 |
| 预置条件 | 部署网关（全局ns）.consumer1(ns1).provider2（ns1).consumer2(ns2).provider2(ns2)    2.已创建分组并导入api发布 |
| 测试过程 | 进入单元化规则tab页  设置单元化规则，使请求落到consumer1再落到provider2    后台请求网关  curl -H "city:test1" -H "userId:test2" -H "age:test3" http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2  4.后台查看provider1.provider2实时日志 |
| 预期结果 | 请求命中子规则1和子规则2多个标识请求成功，请求成功后provider1日志未收到请求信息，provider2日志可以收到请求信息  网关请求成功打印：    Provider1日志无相关信息打印：    Provider2日志： |
| 测试结果 |  |
| 备注 |  |

跨命名空间-多个子规则多个标识逻辑关系为或

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 跨命名空间-多个子规则多个标识逻辑关系为或 |
| 测试目的 | 验证跨命名空间服务下单元化规则可以生效 |
| 预置条件 | 部署网关（全局ns）.consumer1(ns1).provider2（ns1).consumer2(ns2).provider2(ns2)    2.已创建分组并导入api发布 |
| 测试过程 | 进入单元化规则tab页  设置单元化规则，使请求落到consumer1再落到provider2    后台请求网关  命令一：curl -H "city:test1" -H "userId:test2" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  命令二：curl -H "city:test1" -H "age:test3" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  命令三：curl -H "city:test1" -H "userId:test2" -H "age:test3" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  4.后台查看provider1.provider2实时日志 |
| 预期结果 | 1.请求命中子规则1和子规则2任意一个标识则请求成功，请求成功后provider1日志未收到请求信息，provider2日志可以收到请求信息  网关请求成功打印：        Provider1日志无相关信息打印：    Provider2日志： |
| 测试结果 |  |
| 备注 |  |

多个子规则tag相同而目的命名空间不同

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 用例名称 | 多个子规则同时生效，按顺序匹配 |
| 测试目的 | 验证单元化规则多个子规则tag相同而目的命名空间不通情况下，请求按顺序落到相应命名空间服务中 |
| 预置条件 | 1.部署网关（全局ns）.consumer1(ns1).provider2（ns1).consumer2(ns2).provider2(ns2)    2.已创建分组并导入api发布 |
| 测试过程 | 进入单元化规则tab页  设置单元化规则，设置多个相同的tag，目的命名空间落到不同的命名空间中    后台请求网关  curl -H "age:23" <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  4.后台查看provider1.provider2实时日志 |
| 预期结果 | 如果单元化下子规则tag相同而目的命名空间不同，请求会落到最前面的子规则下的命名空间的服务  网关请求成功打印：    Provider1日志无相关信息打印：    Provider2日志： |
| 测试结果 |  |
| 备注 |  |

异常场景

开启单元后如果没有配置单元化规则，访问报错

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 开启单元后如果没有配置单元化规则，访问报错 |
| 测试目的 | 验证开启单元化后没有配置单元化规则，网关访问失败 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关  3.已经创建分组并导入api发布  4.单元范围下关联命名空间 |
| 测试过程 | 进入网关管理页面  进入api管理页面复制api，拼接请求命令  curl <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  3.进入后台网关发起请求 |
| 预期结果 | 网关开启单元化后，未配置单元化规则，网关请求失败 |
| 测试结果 |  |
| 备注 |  |

开启单元后配置单元化规则，但是未携带key访问不通

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 开启单元后配置单元化规则，但是未携带key访问不通 |
| 测试目的 | 验证网关开启单元化后，请求未携带key，或key携带错误，访问不通 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关  3.已经创建分组并导入api发布 |
| 测试过程 | 进入网关管理页面  单元范围下导入命名空间    配置单元化规则    进入api管理页面复制api，拼接请求命令  未携带key：curl <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  携带key错误：curl –H “userId:abc” <http://121.4.168.46:8080/test1/consumer-demo/echo-rest-unit/test2>  3.进入后台网关发起请求 |
| 预期结果 | 网关开启单元化后，配置单元化规则，请求未携带key或携带的key错误，则请求失败 |
| 测试结果 |  |
| 备注 |  |

开启单元化后原非单元化方式无法访问

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P 0 |
| 用例名称 | 开启单元化后原先非单元化访问方式无法访问 |
| 测试目的 | 验证开启单元化后原先非单元化访问方式无法访问 |
| 预置条件 | 1.用户正常登录租户端平台  2.已经创建好微服务网关  3.创建分组 |
| 测试过程 | 进入网关管理页面  进入api管理界面导入，provider服务下的/echo/{param}    发布分组  网关后台发起请求  curl http://121.4.168.46:8080/test1/ns1/provider-demo/echo/123 |
| 预期结果 | 请求失败 |
| 测试结果 |  |
| 备注 |  |

分组管理

绑定单个部署组

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 绑定单个部署组 |
| 测试目的 | 验证微服务网关分组可绑定到部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【分组管理】tab页  3.点击【分组管理】，输入如下信息后，点击确定  图形用户界面, 应用程序  描述已自动生成  4.选择单个部署组，点击【确定】  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | 1.创建网关分组成功 |
| 测试结果 |  |
| 备注 |  |

绑定多个部署组

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 绑定多个部署组 |
| 测试目的 | 验证微服务网关分组可绑定到部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署多个网关在不同部署组中 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【分组管理】tab页  3.点击【分组管理】，输入如下信息后，点击确定  图形用户界面, 应用程序  描述已自动生成  4.选择多个部署组，点击【确定】 |
| 预期结果 | 1.创建网关分组成功 |
| 测试结果 |  |
| 备注 |  |

移除部署组

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 移除部署组 |
| 测试目的 | 验证微服务网关分组可移除部署组 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署多个网关在不同部署组中 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【分组管理】tab页  3.点击【分组管理】，输入如下信息后，点击确定  图形用户界面, 应用程序  描述已自动生成  4.选择单个部署组，点击【确定】  5.点击分组ID进入分组详情  6.选择部署组后点击【解除绑定】  图形用户界面, 文本, 应用程序  描述已自动生成 |
| 预期结果 | 1.解除绑定部署组成功 |
| 测试结果 |  |
| 备注 |  |

微服务网关密钥对鉴权

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 微服务网关发布密钥 |
| 测试目的 | 验证微服务网关分组可绑定密钥（关于密钥可参考<https://cloud.tencent.com/document/product/649/41238>） |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署多个网关在不同部署组中 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【分组管理】tab页  3.点击【分组管理】，输入如下信息后，点击确定  图形用户界面, 应用程序  描述已自动生成  4.选择单个部署组，点击【确定】  5.点击分组ID进入分组详情  6.点击【新建密钥】.输入python，然后点击提交  图形用户界面, 应用程序  描述已自动生成  7.通过CURL测试  携带密钥信息  curl -H 'userId:123' -H 'x-mg-secretid:WcQnZbnlRMiHhh4TRVM0Ng==' -H 'x-mg-sign:HvDLymcpXJRPZnj8YQQvsKrJs0w=' -H 'x-mg-nonce:trgkqU3ZF9v4Ti86xaPlHK' -H 'x-mg-alg:1' -H 'x-mg-traceid:71b2-11e9-acee-5254001-8a237eba' <http://172.17.32.102:8080/new/consumer-demo/echo-rest/abc>  不携带密钥信息  curl <http://172.17.32.102:8080/new/consumer-demo/echo-rest/abc>  8.点击【禁用】密钥 |
| 预期结果 | 携带密钥信息可以访问，不携带密钥信息不可以访问  请求无法访问 |
| 测试结果 |  |
| 备注 |  |

微服务网关绑定TAG插件

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 微服务网关绑定插件 |
| 测试目的 | 验证微服务网关分组可绑定插件 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【分组管理】tab页  3.点击【分组管理】，输入如下信息后，点击确定  图形用户界面, 应用程序  描述已自动生成  4.选择单个部署组，点击【确定】  图形用户界面, 应用程序  描述已自动生成  6.参考单元化规则部分用例配置网关—consumer—provider的规则可调用  7. 进入【插件管理】页面，点击【新建插件】管理，创建如下插件  图形用户界面, 应用程序, Word  描述已自动生成  8.点击绑定对象到刚才创建的网关分组，通过curl请求curl -H "userId:123" http://网关ip:8080/test/provider-demo/echo/abc  9.点击解除绑定对象到刚才创建的网关分组，通过curl请求curl -H "userId:123" http://网关ip:8080/test/provider-demo/echo/abc |
| 预期结果 | 绑定插件到网关分组成功，通过调用链可以查看到数据已经打上tag标签  解除绑定插件到网关分组成功，通过调用链可以查看到数据已经没有tag标签 |
| 测试结果 |  |
| 备注 |  |

API管理

部分API导入

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 部分API导入 |
| 测试目的 | 验证微服务网关分组可部分导入API |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【API管理】tab页  3.点击【导入API】  4.跳转到导入页面，选择部分api导入（命名空间不显示非单元组的命名空间）  图形用户界面, 文本, 应用程序  描述已自动生成  5.选择需要导入的API  图形用户界面, 应用程序  描述已自动生成 |
| 预期结果 | API导入成功  图形用户界面, 表格  描述已自动生成 |
| 测试结果 |  |
| 备注 |  |

全部API导入

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 全部API导入 |
| 测试目的 | 验证微服务网关分组可全部导入API |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【API管理】tab页  3.点击【导入API】  4.跳转到导入页面，选择全部api导入（命名空间不显示非单元组的命名空间）  图形用户界面, 文本, 应用程序  描述已自动生成 |
| 预期结果 | 1.API导入成功  图形用户界面, 表格  描述已自动生成 |
| 测试结果 |  |
| 备注 |  |

网关服务API级别超时时间

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | API 级别超时规则 |
| 测试目的 | 验证微服务网关可设置API 级别超时规则 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【API管理】tab页  3.点击导入provider下的/echo/slow/{str}接口，设置超时时间  图形用户界面, 应用程序, Teams  描述已自动生成  4.curl请求发送curl -H "userId:123" http://网关IP:8080/new/provider-demo/echo/slow/abc |
| 预期结果 | 1.超时时间设置成功，并且curl请求时返回超时错误 |
| 测试结果 |  |
| 备注 |  |

网关服务API级别限流时间

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | API 级别限流规则 |
| 测试目的 | 验证微服务网关可设置API 级别限流规则 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【API管理】tab页  3.点击/echo/rest/{str}接口设置限流规则  图形用户界面, 应用程序  描述已自动生成  4.curl请求发送curl -H "userId:123" http://网关IP:8080/new/consumer-demo/echo-rest/abc |
| 预期结果 | 1.限流设置成功，并且curl请求时返回超时错误 |
| 测试结果 |  |
| 备注 |  |

重定向配置

网关服务API重定向

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | API 重定向配置 |
| 测试目的 | 验证微服务网关可设置API重定向配置 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 测试过程 | 1.登录TSF控制台，进入【微服务网关】页面  2.点击服务id进入服务详情页，切换到【重定向配置】tab页  3.新建重定向配置  /test/(.\*)  -->  /new/consumer-demo/echo-rest/$1  图形用户界面, 应用程序  描述已自动生成  4.通过curl请求curl -H 'userId:123' http://172.17.32.61:8080/test/123  5.删除重定向配置后通过curl请求curl -H 'userId:123' http://172.17.32.61:8080/test/123 |
| 预期结果 | 1.重定向配置设置成功，并且curl请求时返回成功  2.删除重定向配置后，curl请求返回错误 |
| 测试结果 |  |
| 备注 |  |

服务治理

服务路由

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 服务路由验证 |
| 测试目的 | 验证微服务网关单元化场景下服务路由功能正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1，另外还需创建一个provider与前一个provider不在一个部署组  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 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【服务路由】tab页  3.配置流量来源为上游服务为consumer到两个不同的部署组的provider流量比例2:8的路由规则，并开启路由规则  4.通过curl持续发送流量curl -H 'userId:123' http://172.17.32.61:8080/new/consumer-demo/echo-rest/unitabc |
| 预期结果 | 1.服务路由生效，流量比例正常  图形用户界面, 应用程序  描述已自动生成 |
| 测试结果 |  |
| 备注 |  |

服务限流

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 服务限流验证 |
| 测试目的 | 验证微服务网关单元化场景下服务限流功能正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 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.配置全局限流规则1s/1次请求，并开启限流规则  4.通过curl持续发送流量curl -H 'userId:123' http://172.17.32.61:8080/new/consumer-demo/echo-rest/unitabc |
| 预期结果 | 1.服务限流生效，每秒超过一个请求时，请求返回失败 |
| 测试结果 |  |
| 备注 |  |

服务熔断

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 服务熔断验证 |
| 测试目的 | 验证微服务网关单元化场景下服务熔断功能正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 测试过程 | 1.登录TSF控制台  2.在【微服务网关】被测网关的【API管理】点击导入provider下的/echo/error/{str}接口  3.点击zuul服务id进入服务详情页，切换到【服务熔断】tab页  4.配置如下熔断规则  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  5.通过curl持续发送流量curl -H 'userId:123' <http://172.17.32.61:8080/new/provider-demo/echo/error/unitabc> |
| 预期结果 | 1.发送请求到一定数量时，触发熔断时间，请求开始报错（熔断错误提示） |
| 测试结果 |  |
| 备注 |  |

服务鉴权

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P1 |
| 用例名称 | 服务鉴权验证 |
| 测试目的 | 验证微服务网关单元化场景下服务鉴权功能正常 |
| 预置条件 | 1.用户正常登录平台  2.用户已经创建或部署网关（全局命名空间），consuemr以及provider部署在创建的非全局命名空间unit1  3.已经创建好网关分组，并且配置好服务调用链参考单元化规则部分用例配置网关—consumer—provider的规则可调用 |
| 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【服务鉴权】tab页  3.配置黑名单鉴权，系统标签上游服务为consumer的服务不可调用，并开启  4.通过curl持续发送流量curl -H 'userId:123' <http://172.17.32.61:8080/new/consumer-demo/echo-rest/unitabc>  5.切换为规则为白名单，继续发送流量 |
| 预期结果 | 1.黑名单时服务调用返回失败  2.白名单时服务调用返回成功 |
| 测试结果 |  |
| 备注 |  |

弹性伸缩

虚机部署组根据响应时间弹性伸缩

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组根据响应时间弹性伸缩 |
| 测试目的 | 验证虚机部署组可以根据响应时间弹性伸缩 |
| 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->mesh测试包”路径下的测试包  3.用户已经部署user和shop服务 |
| 测试过程 | 1.部署user、shop服务，实例数为1，机器配置为1C1G  2.弹性伸缩页面创建基于响应时间的伸缩规则  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shop-1  图形用户界面, 应用程序  描述已自动生成  使用curl -H "user=test" http:// <user实例ip>: <user端口>/api/v6/sleep触发user调用shop，每次调用shop会延迟5s才返回  6.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  7.使用curl -H "user=test" http:// http:// <user实例ip>: <user端口>/api/user/create触发user调用shop，调用没有延迟，满足缩容条件  8.等待几分钟之后shop-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, 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 |  |

虚机部署组根据请求QPS弹性伸缩

|  |  |
| --- | --- |
| 用例名称 | 虚机部署组根据请求QPS弹性伸缩 |
| 测试目的 | 验证虚机部署组可以根据请求QPS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->mesh测试包”路径下的测试包  3.用户已经部署user和shop服务务 |
| 测试过程 | 1.部署user、shop服务，实例数为1，机器配置为1C1G  2.弹性伸缩页面创建基于请求QPS的伸缩规则  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shop-1  图形用户界面, 应用程序  描述已自动生成  使用以下脚本触发user调用shop，使QPS大于每秒10次  #!/bin/bash  function caller() {  for ((i=0;i<20;i++))  do  curl -H "user=test" http:// http:// <user实例ip>: <user端口>/api/v6/user/create  echo -e "\r"  done  }  while true; do  caller  echo -e "\r"  sleep 1  done  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  7.修改脚本降低每秒请求次数，等待几分钟之后shop-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, 文本, 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 |  |

原生应用

consul服务



服务注册

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务注册成功 |
| 测试耗时 | 0.2H |
| 预置条件 | 1.用户正常登录微服务平台 |
| 测试过程 | 1.应用管理界面创建原生应用  2.上传consul-consumer和consul-provider应用服务包  3.创建部署组部署consul-consumer和consul-provider服务  4.查看服务治理信息页  5.后台调用，查看调用数据监控信息 |
| 预期结果 | 1.步骤4结果    2.调用成功 |
| 测试结果 |  |
| 备注 |  |

服务治理监控

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务监控 |
| 测试耗时 | 0.2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 1.后台发起请求：  curl consul-consumer:8001/ping-provider  2.查看服务治理监控页面查看请求监控数据 |
| 预期结果 |  |
| 测试结果 |  |

接口列表管理

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务接口可使用 |
| 测试耗时 | 1H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 进入服务治理consul-provider服务的接口列表管理页面  选择一个api进行调试  后台调用该接口  查看接口概览  查看请求详情 |
| 预期结果 | 1.接口列表    2.调试结果    3.后台调用    4.接口概览    5.调用详情 |
| 测试结果 |  |
| 备注 |  |

服务鉴权

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务服务鉴权规则生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 1.进入服务鉴权页面    2.开启黑名单鉴权  3.新建鉴权规则，标签选择上游服务名，逻辑关系为等于    4.修改逻辑关系为不等于、包含、不包含、正则表达式，查看返回结果  5.重复步骤3、步骤4，标签名测试上游应用    6.重复步骤3、步骤4，标签名测试上游应用版本号    7.重复步骤3、步骤4，标签名测试上游部署组    8.重复步骤3、步骤4，标签名测试上游ip    9. 重复步骤3、步骤4，标签名测试当前应用版本号    10. 重复步骤3、步骤4，标签名测试当前部署组    11. 重复步骤3、步骤4，标签名测试当前服务API PATH    12. 重复步骤3、步骤4，标签名测试HTTP METHOD    13.重复步骤3、步骤4，标签名测试自定义标签    请求命令：curl -H “userId:123” consul-consumer:8001/ping-provider  14.打开白名单，设置逻辑关系为不等于    15.关闭白名单 |
| 预期结果 | 1.黑名单鉴权标签规则生效后返回结果  2. 黑名单鉴权标签规则关闭后返回结果    3.白名单鉴权标签规则生效后返回结果    4.白名单鉴权标签规则关闭后返回结果 |
| 测试结果 |  |

服务路由

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务服务路由生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 1.进入服务路由页面  2.进入consul-provider服务路由，创建路由规则  3.流量来源设置（设置一条逆向路由链路）      4.再consul-consumer机器后台发起请求，查看返回结果  请求命令：curl consul-provider:8002/ping  5.修改逻辑关系，测试等于、包含、不包含、正则表达式  6.重复步骤3、步骤4、步骤5测试标签名为命名空间+上游服务名    7.重复步骤3、步骤4、步骤5测试标签名为上游应用    8.重复步骤3、步骤4、步骤5测试标签名为上游应用版本号    9.重复步骤3、步骤4、步骤5测试标签名为上游部署组    10.重复步骤3、步骤4、步骤5测试标签名为上游ip    11.重复步骤3、步骤4、步骤5测试标签名为当前服务API PATH    12.重复步骤3、步骤4、步骤5测试标签名为请求HTTP METHOD    13.重复步骤3、步骤4、步骤5测试标签名为自定义标签    自定义标签测试请求命令：curl -H "userId:123" consul-provider:8002/ping  14.查看服务路由页面流量数据曲线 |
| 预期结果 | 1.路由标签生效返回结果    2.服务路由页面流量数据曲线可以显示 |
| 测试结果 |  |

服务限流

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务限流功能生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 1.进入服务治理页面  2.进入consul-provider服务，创建限流规则  3.设置全局限流规则  持续请求命令：for i in {1..60000};do sleep 0.5; echo $(date +%T); curl consul-consumer:8001/ping-provider; done    4.设置限流规则为系统标签，标签名测试上游服务名    5. 设置限流规则为系统标签，标签名测试上游应用    6. 设置限流规则为系统标签，标签名测试上游应用版本号    7. 设置限流规则为系统标签，标签名测试上游部署组    8. 设置限流规则为系统标签，标签名测试上游IP    9. 设置限流规则为系统标签，标签名测试当前应用版本号    10. 设置限流规则为系统标签，标签名测试当前部署组    11. 设置限流规则为系统标签，标签名测试当前服务API PATH    12. 设置限流规则为系统标签，标签名测试请求HTTP METHOD    13.查看限流曲线 |
| 预期结果 | 1.全局限流规则返回结果  文本  描述已自动生成  2.系统标签限流规则返回结果  文本  描述已自动生成 |
| 测试结果 |  |

服务熔断

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证consul-consumer和consul-provider服务服务熔断生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署consul-consumer、consul-provider服务 |
| 测试过程 | 1.进入服务治理页面  2.进入consul-consumer服务熔断页面，创建熔断规则    3.进入consul-provider服务鉴权页面，创建鉴权规则制造请求返回错误来触发熔断  4. 后台请求触发熔断    5.查看熔断事件  6.如果熔断事件触发，关闭熔断、鉴权规则请求一段时间，让熔断事件恢复 |
| 预期结果 | 1.熔断事件生效    2.熔断事件恢复 |
| 测试结果 |  |

Eureka服务

备注：eureka-consumer和eureka-provider服务测试用例参考用例20.1.1~20.1.7consul-consumer和consul-provider服务的测试用例

Gateway服务

服务注册

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务注册成功 |
| 测试耗时 | 0.2H |
| 预置条件 | 1.用户正常登录微服务平台 |
| 测试过程 | 1.应用管理界面创建原生应用  2.上传gateway、consul-provider应用服务包  3.创建部署组部署gateway和consul-provider服务  4.查看服务治理信息页  5.后台调用，查看调用数据监控信息 |
| 预期结果 | 1.步骤4结果    2.调用成功 |
| 测试结果 |  |
| 备注 |  |

服务治理监控

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务监控 |
| 测试耗时 | 0.2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署gateway、consul-provider服务 |
| 测试过程 | 1.后台发起请求：  curl gateway:8005/consul-provider/ping  2.查看服务治理监控页面查看请求监控数据 |
| 预期结果 |  |
| 测试结果 |  |

服务鉴权

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务服务鉴权规则生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署gateway、consul-provider服务 |
| 测试过程 | 1.进入consul-provider服务鉴权页面    2.开启黑名单鉴权  3.新建鉴权规则，标签选择上游服务名，逻辑关系为等于  后台请求命令：curl gateway:8005/consul-provider/ping    4.修改逻辑关系为不等于、包含、不包含、正则表达式，查看返回结果  5.重复步骤3、步骤4，标签名测试上游应用    6.重复步骤3、步骤4，标签名测试上游应用版本号    7.重复步骤3、步骤4，标签名测试上游部署组    8.重复步骤3、步骤4，标签名测试上游ip    9. 重复步骤3、步骤4，标签名测试当前应用版本号    10. 重复步骤3、步骤4，标签名测试当前部署组    11. 重复步骤3、步骤4，标签名测试当前服务API PATH    12. 重复步骤3、步骤4，标签名测试HTTP METHOD    13.重复步骤3、步骤4，标签名测试自定义标签    请求命令：curl -H "userId:123" gateway:8005/consul-provider/ping  14.打开白名单，设置逻辑关系为不等于    15.关闭白名单 |
| 预期结果 | 1.黑名单鉴权标签规则生效后返回结果    2. 黑名单鉴权标签规则关闭后返回结果    3.白名单鉴权标签规则生效后返回结果    4.白名单鉴权标签规则关闭后返回结果 |
| 测试结果 |  |

服务路由

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务服务路由生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署gateway、consul-provider服务 |
| 测试过程 | 1.进入服务路由页面  2.进入consul-provider服务路由，创建路由规则  3.流量来源设置（设置一条逆向路由链路）    4.再gateway机器后台发起请求，查看返回结果  请求命令：curl gateway:8005/consul-provider/ping    5.修改逻辑关系，测试等于、包含、不包含、正则表达式  6.重复步骤3、步骤4、步骤5测试标签名为命名空间+上游服务名    7.重复步骤3、步骤4、步骤5测试标签名为上游应用    8.重复步骤3、步骤4、步骤5测试标签名为上游应用版本号    9.重复步骤3、步骤4、步骤5测试标签名为上游部署组    10.重复步骤3、步骤4、步骤5测试标签名为上游ip    11.重复步骤3、步骤4、步骤5测试标签名为当前服务API PATH    12.重复步骤3、步骤4、步骤5测试标签名为请求HTTP METHOD    13.重复步骤3、步骤4、步骤5测试标签名为自定义标签    自定义标签测试请求命令：curl -H "userId:123" gateway:8005/consul-provider/ping  14.查看服务路由页面流量数据曲线 |
| 预期结果 | 1.路由标签生效请求返回结果    2.路由标签规则关闭后请求返回结果    2.服务路由页面流量数据曲线可以显示 |
| 测试结果 |  |

服务限流

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务限流功能生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署gateway、consul-provider服务 |
| 测试过程 | 1.进入服务治理页面  2.进入consul-provider服务，创建全局限流规则  3.后台发起请求  持续请求命令：for i in {1..60000};do sleep 0.5; echo $(date +%T); curl gateway:8005/consul-provider/ping; done    4.进入consul-provider服务设置限流规则为系统标签，标签名测试上游服务名    5. 设置限流规则为系统标签，标签名测试上游应用    6. 设置限流规则为系统标签，标签名测试上游应用版本号    7. 设置限流规则为系统标签，标签名测试上游部署组    8. 设置限流规则为系统标签，标签名测试上游IP    9. 设置限流规则为系统标签，标签名测试当前应用版本号    10. 设置限流规则为系统标签，标签名测试当前部署组    11. 设置限流规则为系统标签，标签名测试当前服务API PATH    12. 设置限流规则为系统标签，标签名测试请求HTTP METHOD    13.查看限流曲线 |
| 预期结果 | 1.全局限流规则返回结果  文本  描述已自动生成  2.系统标签限流规则返回结果  文本  描述已自动生成 |
| 测试结果 |  |

服务熔断

|  |  |
| --- | --- |
| 用例编号 |  |
| 优先级 | P0 |
| 测试目的 | 验证gateway服务服务熔断生效 |
| 测试耗时 | 2H |
| 预置条件 | 1.用户正常登录微服务平台  2.已部署gateway、consul-provider服务 |
| 测试过程 | 1.进入服务治理页面  2.进入gateway服务熔断页面，创建熔断规则      3.进入consul-provider服务鉴权页面，创建鉴权规则制造请求返回错误来触发熔断  4. 后台请求触发熔断    5.查看熔断事件  6.如果熔断事件触发，关闭熔断、鉴权规则请求一段时间，让熔断事件恢复 |
| 预期结果 | 1.熔断事件生效    2.熔断事件恢复 |
| 测试结果 |  |

Zuul服务

备注：zuul服务测试用例参考用例20.3.1~20.3.6gateway服务的测试用例  
测试请求curl包：curl gateway:8005/consul-provider/ping