微服务框架（TSF）

产品文档

腾讯云TCE

文档目录

- 批量删除包
- 删除仓库
- 获取下载程序包信息
- 获取某个应用的程序包信息列表
- 查询仓库列表
- 查询仓库信息
- 获取上传程序包信息
- 更新上传程序包信息
- 更新仓库信息

部署组相关接口

- 创建容器部署组
- 创建虚拟机部署组
- 创建镜像凭证
- 创建Serverless部署组
- 删除容器部署组
- 删除虚拟机部署组
- 删除部署组
- 部署容器应用
- 部署虚拟机部署组应用
- 部署Serverless应用
- 查询容器部署组详情
- 容器部署组列表
- 查询虚拟机部署组详情
- 查询虚拟机部署组云主机列表
- 获取虚拟机部署组列表
- 获取凭证列表
- 获取凭证名称列表
- 查询Serverless部署组明细
- 查询Serverless部署组列表
- 查询简单部署组列表
- 虚拟机部署组添加实例
- 修改容器部署组
- 修改容器部署组实例数
- 缩容虚拟机部署组
- 虚拟机部署组下线实例
- 启动容器部署组
- 启动虚拟机部署组
- 停止容器部署组
- 停止虚拟机部署组
- 更新健康检查配置

配置管理相关接口

- 创建配置项
- 创建公共配置项
- 删除配置项
- 删除公共配置项

产品简介
产品概述

最近更新时间: 2025-02-18 16:02:00

什么是微服务平台 TSF？

微服务平台（Tencent Service Framework，TSF）是一个围绕着应用和微服务的 PaaS 平台，提供应用全生命周期管理、数据化运营、立体化监控和服务治理等功能。TSF 拥抱 Spring Cloud 、Service Mesh 微服务框架，帮助企业客户解决传统集中式架构转型的困难，打造大规模高可用的分布式系统架构，实现业务、产品的快速落地 。

TSF 以中间件团队多款成熟的分布式产品为核心基础组件，提供秒级推送的分布式配置服务、链路追踪等高可用稳定性组件。此外，TSF 与 API 网关和消息队列打通，帮助企业轻松构建大型分布式系统。

产品功能

最近更新时间: 2025-02-18 16:02:00

服务注册与发现

- **金融级高可用注册中心** 提供金融级高可用的服务注册中心，数据多副本，支持服务自动注册和发现，无须配置注册中心地址即可使用。
- **健康检查** 支持健康检查，如果出现宕机或服务不可用时，注册中心自动剔除不可用实例。
- **毫秒级推送** 客户端和服务注册中心建立长链接，任何服务注册信息变更，立即推送。
- **服务本地缓存** 客户端 SDK 拥有内存和文件级别缓存，当访问注册中心失败的时候会自动启用缓存数据，保证服务发现高可用。

细粒度的服务治理

提供服务和 API 级别的服务治理能力，提供高可用服务治理能力，保障服务高质量运行。

- **服务鉴权** 为服务提供安全的访问机制，支持黑白名单鉴权方式，支持系统和业务标签参数进行鉴权。
- **服务路由** 用户可以通过配置、权重标签的形式进行细粒度的流量控制，实现灰度发布、就近路由、部分账号内测、流量限制、访问权限控制等功能。
- **服务限流** 保障业务不被突发流量击垮，提高系统问题稳定性。支持服务和接口级限流配置和监控。
- **服务熔断** 当下游的服务因为某种原因导致服务不可用或响应过慢时，上游服务为了保证自己整体服务的可用性，不再继续调用目标服务，直接返回。当下游服务恢复后，上游服务会恢复调用。
- **服务容错&降级** 支持 failfast、failover 和 forking 容错策略和 fallback 降级方法。

全面的应用生命周期管理

- **多种应用托管方式** 支持虚拟机、容器、部署方式。使用虚拟机部署，应用可以独占资源；使用容器部署，可灵活分配资源实现资源共享。
- **应用全生命周期管理** 提供从创建应用到运行应用的全程管理，功能包括创建、删除、部署、回滚、扩容、下线、启动和停止应用。支持变更记录查询。
- **版本管理** 支持软件仓库和镜像仓库管理程序包版本和镜像版本，支持自定义软件仓库对接用户 COS。

高可靠的配置中心

TSF 提供分布式配置和文件配置两种配置功能，提供可视化的配置管理界面，支持在应用运行时动态修改配置。

- **可视化配置管理** 用户可以在控制台上管理配置，支持多版本管理，支持将配置发布到应用部署单元（部署组）或者命名空间范围。
- **配置动态推送，实时生效** 支持配置动态推送，服务从配置中心读取到更新后的配置进行逻辑处理，支持配置回调方法。支持查看部署组上已发布的配置，支持配置回滚操作。
- **推送记录查看** 支持按照部署组或者配置的维度查看推送的配置记录，支持配置文件的导入和导出。

可视化应用运维

提供全面的监控和分布式调用链分析工具，帮助用户把握应用上线后的运行状况。

- **服务监控** 支持服务和接口的成功率、调用量、耗时、异常次数等多维度监控和告警。
- **服务依赖拓扑** 支持查看服务之间的依赖关系，了解系统瓶颈服务和链路并进行针对性的服务优化。支持服务与 API 网关、消息队列、数据库等上下游组件的链路查看。
- **JVM 监控** 支持查看 JVM 内存分布、线程、堆栈、火焰图。

- **日志服务** 提供日志采集、日志存储、日志检索，日志关键词告警等功能。支持日志与调用链联动排查线上问题。

- **全链路灰度发布**

  在发布过程中，将具有一定特征或者比例的流量分配到需要被验证的版本中，用来观察新的验证版本的线上运行状态。当线上调用链路较为复杂时，全链路灰度发布可以将线上的各个服务隔离出一个单独的运行环境。

- **弹性伸缩**

  支持根据预先设定的弹性伸缩规则，动态增加或者减少部署组的实例数。

微服务网关

微服务网关作为后台架构的入口，提供路由转发、API 管理、访问过滤器等作用，是微服务架构中的重要组件。

- **请求转发**

  TSF 中的微服务网关可以通过页面配置灵活管理需要被转发请求的微服务 API。微服务网关会及时从注册中心感知后端服务节点健康状况，保证在后端服务节点变动情况下请求不中断。

- **API 管理**

  微服务网关集中管理了所有需要对外暴露的 API，帮助用户进行 API 的生命周期管理。

- **API 治理**

  支持 API 级别的限流、路由等能力，支持用户绑定系统插件或自定义插件。

产品优势

最近更新时间: 2025-02-18 16:02:00

TSF 服务治理平台对比基于开源SpringCloud自建平台，具有以下优势：

对比项	微服务平台TSF Paas平台	基于开源SpringCloud自建平台
服务注册中心	平台提供高可用的注册中心集群，金融级别容灾	自行搭建，维护困难，可靠性差。
调用链能力	符合国人习惯的交互方式，与日志服务联动	使用开源组件，英文界面
服务限流	支持多作用域限流规则的配置。	不支持
服务路由	支持灵活的路由规则配置。	不支持
服务鉴权	支持服务黑白名单和基于 tag 鉴权两种鉴权方式。	不支持
应用部署	成熟、灵活的部署方案，支持灰度发布等高级能力	自行开发部署模块
日志服务	提供采集、呈现、解析、检索的一站式能力，帮助开发者快速定位目标日志	自行搭建，功能简单，无法高效运维。
配置管理	分布式配置，支持从环境、版本、应用三个维度进行管理，配置动态推送和历史推送回溯能力	基于 Spring Cloud Config 进行额外的开发和封装，配置能力弱，维护困难。
微服务API 网关	API 网关提供微服务网关能力，支持配置负载均衡、熔断、限流等策略。	基于 Zuul 等组件开发
资源自动扩缩容	提供虚拟机集群和容器集群按照 CPU，内存使用率等指标实时自动扩缩容。	手动进行，无法做到实时，可扩展性差。
易用性	提供一站式的控制管理平台，提供可视化，简单操作的集群部署，应用管理，数字化运营，配置管理等能力， 降低部署和维护成本。	需要自行开发和维护，开发难度和维护难度大，功能简单。
售后服务	提供稳定及时的售后服务，强大的技术、运维团队支持。	企业运维团队支持。

应用场景

最近更新时间: 2025-02-18 16:02:00

构建分布式服务系统

单体应用转变为分布式系统后，实现系统间的可靠调用是关键问题之一，涉及到路由管理、序列化协议等技术细节。

TSF 提供了 RESTful 调用方式和自研的高性能 RPC 框架，能够构建高可用、高性能的分布式系统，TSF 系统地考虑了分布式服务发现、路由管理、安全、负载均衡等细节问题。同时，TSF 将在未来打通消息队列、API Gateway 等服务，满足用户多样化的需求。

应用发布和管理

相对于传统的应用发布需要运维人员登录到每一台服务器进行发布和部署，TSF 针对分布式系统的应用发布和管理，提供了简单易用的可视化控制台。用户通过控制台可以发布应用，包括创建、部署、启动应用，也支持查看应用的部署状态。除此之外，用户可以通过控制台管理应用，包括回滚应用、扩容、缩容和删除应用。

数据化运营

通过对日志埋点的收集和分析，可以得到一次请求在各个服务间的调用链关系，有助于梳理应用的请求入口与服务的调用来源、依赖关系。当遇到请求耗时较长的情况，可以通过调用链分析调用瓶颈，快速定位异常。下图表示 TSF 提供的服务依赖拓扑图，可以直观地了解服务与服务之间，服务与下游组件之间的调用关系。

服务治理

支持服务级别和 API 级别的服务治理能力，包括服务路由、服务限流、服务鉴权功能。服务路由功能支持将请求按权重路由到不同版本的服务上。

使用限制总览

最近更新时间: 2025-02-18 16:02:00

TSF 针对每个用户在使用上有如下限制：

资源与部署相关

限制类型	限制说明	备注
免费注册到注册中心的服务实例额度	20个	-
容器集群数量	最多5个	-
全局命名空间个数	1个	-
虚拟机集群导入云主机的操作系统	CentOS 7.5 、Ubuntu 18.04 LTS	-
程序包仓库存储容量	最多100GB	-
单个程序包最大上限	256MB	-

配置相关

限制类型	限制说明
应用/全局配置项的单个版本的大小	最大65535个字节
单个应用配置项 Key 数量	最多100个
文件配置单个配置最大上限	500KB

服务治理相关

限制类型	限制说明
方法调用设置 tag 数量	单个租户下最多16个
全链路灰度发布规则数量	最多100条
单个服务下服务路由规则数量	最多10条
微服务网关 API Path 最长长度	128字符

日志与调用链相关

限制类型	限制说明
日志存储上限	基础版每个节点最大2.5GB，专业版每个节点最大4GB，铂金版每个节点最大20GB
日志存储天数	最多30天
单次日志查询的日志条数	最多10000条

日志配置项数量	最多200条
单个日志配置项采集最长采集路径	1024字节

名词解释

最近更新时间: 2025-02-18 16:02:00

以下视频将为您介绍资源管理的重要概念及其关系说明：

名词解释

TSF 常用词汇解释如下，更多词汇请参考 词汇表。

名词	含义	使用说明
集群	计算资源的管理维度	TSF 中的集群分为虚拟机集群和容器集群。使用虚拟机或者容器计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。
命名空间	命名空间对微服务之间的调用起到隔离作用	同一命名空间下，微服务可以直接相互调用。 不同命名空间中运行的微服务不能直接相互调用，需要通过公网或者网关来相互连通。 通常情况下，命名空间可以起到环境隔离的作用（如隔离开发、测试环境）或服务分组的作用。
应用	用户的业务应用	通过应用可以对用户的程序包以及应用配置进行管理。
服务	用户线上运行的微服务	一个应用注册到注册中心后，会注册成为一个（Spring Cloud 或者 Mesh）或多个（Dubbo）微服务。TSF 通过用户应用程序包中声明的服务名来注册微服务。
部署组	状态相同的节点的最小集合	同一个部署组运行了相同的程序包、相同的配置、使用相同的启动参数。 用户在某一个应用下创建部署组，使用应用下某一个程序包，使用集群中的云主机或者容器资源，将应用部署在某一个环境（命名空间）中。
节点/实例/服务实例	一台虚拟机或者一个容器的 pod	-

概念之间关系

集群、命名空间、部署组三个概念之间的关系如下：

相关概念

最近更新时间: 2025-02-18 16:02:00

名词解释

TSF 常用词汇解释如下，更多词汇请参考 [词汇表]。

名词	含义	使用说明
集群	计算资源的管理维度	TSF 中的集群分为虚拟机集群和容器集群。使用虚拟机或者容器计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。
命名空间	命名空间对微服务之间的调用起到隔离作用	同一命名空间下，微服务可以直接相互调用。 不同命名空间中运行的微服务不能直接相互调用，需要通过公网或者网关来相互连通。 通常情况下，命名空间可以起到环境隔离的作用（如隔离开发、测试环境）或服务分组的作用。
应用	用户的业务应用	通过应用可以对用户的程序包以及应用配置进行管理。
服务	用户线上运行的微服务	一个应用注册到注册中心后，会注册成为一个（Spring Cloud 或者 Mesh）或多个（Dubbo）微服务。TSF 通过用户应用程序包中声明的服务名来注册微服务。
部署组	状态相同的节点的最小集合	同一个部署组运行了相同的程序包、相同的配置、使用相同的启动参数。 用户在某一个应用下创建部署组，使用应用下某一个程序包，使用集群中的云主机或者容器资源，将应用部署在某一个环境（命名空间）中。
节点/实例/服务实例	一台虚拟机或者一个容器的 pod	-

概念之间关系

集群、命名空间、部署组三个概念之间的关系如下：

快速入门
入门流程指引

最近更新时间: 2025-02-18 16:02:00

名词解释

TSF 常用词汇解释如下，更多词汇请参考 [词汇表]。

名词	含义	使用说明
集群	计算资源的管理维度	TSF 中的集群分为虚拟机集群和容器集群。使用虚拟机或者容器计算资源时，用户需要提前先将云主机导入集群中，才能进行应用的部署。
命名空间	命名空间对微服务之间的调用起到隔离作用	同一命名空间下，微服务可以直接相互调用。 不同命名空间中运行的微服务不能直接相互调用，需要通过公网或者网关来相互连通。 通常情况下，命名空间可以起到环境隔离的作用（如隔离开发、测试环境）或服务分组的作用。
应用	用户的业务应用	通过应用可以对用户的程序包以及应用配置进行管理。
服务	用户线上运行的微服务	一个应用注册到注册中心后，会注册成为一个（Spring Cloud 或者 Mesh）或多个（Dubbo）微服务。TSF 通过用户应用程序包中声明的服务名来注册微服务。
部署组	状态相同的节点的最小集合	同一个部署组运行了相同的程序包、相同的配置、使用相同的启动参数。 用户在某一个应用下创建部署组，使用应用下某一个程序包，使用集群中的云主机或者容器资源，将应用部署在某一个环境（命名空间）中。
节点/实例/服务实例	一台虚拟机或者一个容器的 pod	-

概念之间关系

集群、命名空间、部署组三个概念之间的关系如下：

获取访问授权
主账号获取访问授权

最近更新时间: 2025-02-18 16:02:00

操作背景

由于 TSF 需要访问其他云产品的 API（例如 TKE），所以需要授权 TSF 创建服务角色。

操作步骤

1. 应用**主账号**登录 TSF 控制台，进入概览页。由于没有授权微服务平台 TSF 服务角色权限无法访问其他云产品资源。

2. 单击【前往授权】，进入 CAM 控制台 授权，单击【同意授权】，则为微服务平台 TSF 授权服务角色访问您其他云产品资源。

> **注意：**
>
> 子账号使用TSF请参考【访问管理】进行访问授权。

子账号获取访问授权

最近更新时间: 2025-02-18 16:02:00

CAM 基本概念

主账号通过给子账号绑定策略实现授权，策略设置可精确到 **[API，资源，用户/用户组，允许/拒绝，条件]** 维度。

账户

- **主账号**：拥有腾讯云金融专区所有资源，可以任意访问其任何资源。
- **子账号：**包括子用户和协作者。
  - **子用户**： 由主账号创建，完全归属于创建该子用户的主账号。
  - **协作者**：本身拥有主账号身份，被添加作为当前主账号的协作者，则为当前主账号的子账号之一，可切换回主账号身份。
- **身份凭证**：包括登录凭证和访问证书两种，**登录凭证** 指用户登录名和密码，**访问证书** 指云 API 密钥（SecretId 和 SecretKey）。

资源与权限

- **资源**：资源是云服务中被操作的对象，如一个云服务器实例、COS 存储桶、VPC 实例等。
- **权限**：权限是指允许或拒绝某些用户执行某些操作。默认情况下，**主账号拥有其名下所有资源的访问权限**，而**子账号没有主账号下任何资源的访问权限。**
- **策略**：策略是定义和描述一条或多条权限的语法规范。**主账号**通过将**策略关联**到用户/用户组完成授权。

子账号使用 TSF

协作者与子账号使用TSF平台时，需要对三方面进行授权：

1. 要将角色（及其许可策略）传递至 TSF 服务，用户必须具有**传递角色**至服务的许可，即创建tsf_PassRole策略，详细操作参考[授予 tsf_PassRole 策略]。
2. 配置使用TSF平台的权限，平台支持为不同子账号灵活配置管理权限，包含对TSF中不同集群、命名空间、应用授予读权限或写权限，详细操作可以参考管理中心 。具体支持以下三种使用场景：

- 为子账号或协作者配置全部资源的全读写策略。您可以通过授予子账号 QcloudTSFFullAccess 策略给予子账号使用TSF平台的全量权限。
- 为子账号或协作者配置全部资源的部分操作权限，如可以为部分用户配置应用、微服务、配置的全读写策略，以及集群、命名空间的只读策略。
- 为子账号和协作者配置某些资源（一个或多个）的读或写权限，如可以为不同的子账号配置不同命名空间、不同应用的不可见、只读、全读写权限。

> **注意：**
>
> 该场景下不支持使用六段式资源自定义配置使用权限和灵活配置管理权限。

3. 使用TSF平台过程中，涉及到对其他产品的调用，如云服务器（CVM）、私有网络（VPC）、标签（TAG）、容器服务（TKE）、镜像仓库（TCR）等，需要主账号对子账号进行授权。详细说明参考[授予访问其他云产品权限]。

授予tsf_PassRole 策略

步骤1. 新建 tsf_PassRole 策略

1. 登录 访问管理控制台 。
2. 在左侧导航栏，单击【策略】，进入策略管理列表页。
3. 单击【新建自定义策略】。
4. 在选择创建策略方式的弹出框中，单击【按策略语法创建】，进入按策略语法创建页。
5. 在 按策略语法创建页 中，选择【空白模板】，并单击【下一步】。
6. 填写策略名和内容，并单击【创建策略】。

使用主账号或具有管理权限的子账号创建如下自定义策略，策略语法如下：

```
{
"version": "2.0",
"statement": {
"effect": "allow",
"action": [
"cam:PassRole"
],
"resource": "qcs::cam::uin/${OwnerUin}:roleName/TSF_QCSRole"
}
}
```

其中 ${OwnerUin} 为主账号ID，从控制台账号信息页面获取。

步骤2. 将 tsf_PassRole 策略绑定到用户

1. 在左侧导航栏，单击【用户】>【用户列表】，进入用户管理页面。
2. 选择要授予 TSF 使用权限的用户，单击操作列的【授权】。
3. 从策略列表中筛选出**步骤一**中的创建的策略（ 如 tsf_PassRole ）。
4. 单击【确定】，绑定策略。该策略会显示在用户的策略列表中。

授予访问其他云产品权限

TSF平台使用中涉及到以下云产品的调用。主账号需要对子账号进行单独授权才能保证对应TSF产品功能的使用。TSF中涉及到的对云产品的调用如下：

云产品	接口名	接口作用	影响到TSF平台的操作
云服务器（CVM）	DescribeKeyPairs	查询密钥对信息	创建集群后通过重装云主机方式导入云主机，选择密钥
云服务器（CVM）	DescribeInstances	查询实例列表	向集群中导入云主机时，查看可导入的云主机列表

云产品	接口名	接口作用	影响到TSF平台的操作
云服务器（CVM）	DescribeZones	查询可用区	创建集群时查看可用区列表
云服务器（CVM）	ResetInstance	重装云主机实例	创建集群后通过重装云主机方式导入云主机
云服务器（CVM）	ModifyInstancesAttribute	修改云主机名称与安全组	通过重装云主机方式导入云主机到集群中的过程中修改云主机安全组
云服务器（CVM）	DescribeSecurityGroups	查看安全组	导入云主机到集群中查询安全组信息
私有网络（VPC）	DescribeVpcs	查询VPC列表	创建集群过程中选择集群所属VPC
私有网络（VPC）	DescribeVpcsEx	查询VPC列表	创建集群过程中选择集群所属VPC
私有网络（VPC）	DescribeSubnets	查询子网列表	选择子网
私有网络（VPC）	DescribeSecurityGroups	查看安全组	通过重装云主机方式导入云主机到集群中的过程中修改云主机安全组
容器服务（TKE）	DescribeClusters	查看容器集群列表	查询容器集群列表
标签（TAG）	DescribeResourceTagsByResourceIds	按顺序查看资源关联的标签	查看按照标签查看对应资源
标签（TAG）	ModifyResourceTags	修改资源的标签信息	修改TSF中资源的标签信息
标签（TAG）	DescribeTagKeys	查看标签键	查看TSF中资源的标签键
标签（TAG）	DescribeResourcesByTagsUnion	通过标签查询资源列表并集	在TSF中通过标签查询资源列表并集
标签（TAG）	DescribeTagValues	查看标签值	查看TSF中资源的标签值
云监控（Monitor）	DescribeProductEventList	获取产品事件列表	在概览页中查看产品事件列表
云监控（Monitor）	GetMonitorData	拉取指标监控数据	查看TSF中监控数据
容器镜像服务（TCR）	全部接口	容器镜像云端托管服务	使用容器部署过程中镜像相关能力

云服务器（CVM）、标签（TAG）、容器服务（TKE）、云监控（Monitor）授权说明：

您可以通过云平台上访问管理能力对以上接口进行授权，详细可参考【创建自定义策略】。

在此给出授权示例： 如您希望子账号使用TSF中针对所有资源的创建集群、导入云主机、配置与查询标签、查看事件与监控能力，且该能力不对任何资源进行区分，可配置接口级别授权策略如下：

```
{
"version": "2.0",
"statement": [
{
"effect": "allow",
"action": [
"cvm:DescribeKeyPairs",
"cvm:DescribeInstances",
"cvm:ResetInstance",
"cvm:ModifyInstancesAttribute",
"cvm:DescribeSecurityGroups",
"tke:DescribeClusters",
"tag:DescribeResourceTagsByResourceIds",
"tag:ModifyResourceTags",
"tag:DescribeTagKeys",
"tag:DescribeTagValues",
"monitor:DescribeProductEventList",
"monitor:GetMonitorData"
],
"resource": [
"*"
]
}
]
}
```

> **注意：**
>
> - 这里进行授权后，子账号也直接获得了相关产品的直接使用权限。建议您按照需求严格管控，可以选择对相关产品接口进行资源级别授权。
> - 配置 "cvm:ResetInstance"和"cvm:ModifyInstancesAttribute"接口后子账号将获得重装实例和修改实例的属性的权限，请谨慎配置，避免出现权限过大的场景。

镜像服务（TCR）授权

如果主账号未开通过镜像仓库，会提示如下图所示信息，此时需要主账号登录 TSF 控制台，开通镜像仓库。主账号开通镜像仓库后协作者/子账号才能继续使用镜像仓库。

如子账号需要通过容器部署微服务，需要使用镜像仓库能力，建议为子账号授予 QcloudTCRFullAccess 策略。

私有网络（VPC）授权说明：

建议为子账号授予私有网络读权限 QcloudVPCReadOnlyAccess。注意子账号也将同时可在VPC及相关产品中对私有网络资源拥有读权限。

总结

最终子账号可以授予如下策略：

策略	是否必选	说明
tsf_PassRole	必选	手动创建。

策略	是否必选	说明
QcloudCamSubaccountsAuthorizeRoleFull	必选	访问管理（CAM）子账号授权服务角色相关权限，包含子账号在授权服务角色过程中涉及的全部权限。
QcloudTSFFullAccess	可选	微服务平台（TSF）全读写访问权限，也可以参考 [管理中心] 进行细粒度授权。
QcloudTCRFullAccess	可选	与镜像仓库相关，如果需要使用 TSF 中容器相关功能需要授权。
QcloudVPCReadOnlyAccess	可选	如果需要读取集群 VPC 等信息需要授权。
QcloudMonitorReadOnlyAccess	可选	如果需要读取监控数据需要授权，也可以按照上文说明进行接口级别授权。
QcloudTAGFullAccess	可选	如果需要读写TSF中资源的标签需要授权，也可以按照上文说明进行接口级别授权。
QcloudCVMFullAccess	可选	如果需要向TSF集群中导入云主机需要授权，也可以按照上文说明进行接口（DescribeKeyPairs，DescribeInstances，ResetInstance，ModifyInstancesAttribute，DescribeSecurityGroups）、资源级别授权。
QcloudTKEReadOnlyAccess	可选	如果需要创建容器集群需要授权，也可以按照上文说明进行接口（DescribeClusters）、资源级别授权。

补充说明

如何通过访问管理对子账号进行策略授权。

1. 登录 【访问管理控制台】 。
2. 在左侧导航栏，单击【用户】>【用户列表】 ，进入用户管理页面。
3. 选择要授予 TSF 使用权限的用户，单击操作列的【授权】。
4. 从策略列表中选择 `QcloudTCRFullAccess` 策略。

5. 单击【确定】，绑定策略。该策略会显示在用户的策略列表中。

使用容器部署微服务
步骤1：新建容器集群

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台上创建一个容器集群。

前提条件

- 已 获取访问授权
- 已 购买云服务器
- 已 购买节点

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏中，单击【集群】，进入集群列表页。
3. 在集群列表页的左上方，单击【新建】，设置集群的基本信息。

- **集群类型**：选择容器集群。
- **新建类型**：选择**直接创建**。
- **Kubernetes 版本**：选择最新版本。
- **集群名**：填写集群名称。
- **标签**：用于分类管理资源，可不选。
- **所在可用区**：选择默认可用区。
- **集群网络**：选择与已有云服务器相同的 VPC 网络，用来保证后续导入集群的云服务器属于同一 VPC。
- **容器网络**：为集群内容器分配在容器网络地址范围内的 IP 地址。
- **数据集**：选择"无"。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限。
- **备注**：填写备注，选填。

4. 单击【提交】，等待几分钟后集群状态变为运行中即可进行后续导入云服务器操作。

步骤2：导入云主机

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在新建集群后将准备好的云主机导入集群。

操作步骤

1. 在集群列表页面，单击刚刚创建好的集群操作栏的【导入云主机】。
2. 从集群所在 VPC 的云服务器列表中，选择需要添加到集群的主机，单击【下一步】。

3. 填写相关信息。

- **导入方式**：重装系统
- **操作系统**：根据实际选择。
- **登录方式**：提供三种对应登录方式。 a. 设置密码：请根据提示设置对应密码。 b. 立即关联密钥：密钥对是通过一种算法生成的一对参数，是一种比常规密码更安全的登录云服务器的方式。 c. 自动生成密码：自动生成的密码将通过站内信发送给您。
- **数据盘挂载**：将数据盘挂载至指定目录下。建议挂载至容器和镜像存储目录。
- **容器目录**：默认。
- **安全组**：选择 default。安全组具有防火墙的功能，用于设置云服务器的网络访问控制。

4. 单击【提交】，集群列表中云主机数量将变为1。

5. 单击集群的“ID/集群名”，进入云主机列表，等待几分钟后，云主机的可用状态将变为“可用”。

步骤3：创建应用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台创建一个应用。

操作步骤

1. 在左侧导航栏，单击【应用管理】，进入应用列表页。
2. 在应用列表上方单击【新建应用】。

3. 设置应用信息，单击【提交】。

- **应用名**：填写 provider。
- **部署方式**：选择 **容器部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **SpringCloud**。
- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。
- **数据集**：选择"无"。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限。
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击【确认】，前往上传镜像并部署应用。

步骤4：上传镜像或程序包
上传程序包（可选）

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在创建应用后开通镜像仓库并将镜像推送到镜像仓库中。

前提条件

下载 Demo（包含一个 provider 和一个 consumer 程序）。

操作步骤

1. 登控制台。
2. 在左侧导航栏 选择【应用管理】，单击目标应用的**ID/应用名**。
3. 在应用详情页，选择**镜像**标签页，单击【上传程序包/镜像】，选择【JAR包部署】。

- 单击【选择文件】，选择提前准备好的 Demo 中的 provider 的 jar 程序包。
- **程序包版本**：填写版本号，或单击【用时间戳作为版本号】。
- **备注**：填写备注。

4. 单击【上传程序包并制作镜像】，我们将自动为您制作镜像并上传到镜像仓库，右上角将出现任务进行的状态。

5. 任务完成后，在镜像标签页的镜像列表中将看到上传好的镜像。

上传镜像（推荐）

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在创建应用后开通镜像仓库并将镜像推送到镜像仓库中。

前提条件

- 安装 docker
- 使用 `sudo` 允许系统管理员让普通用户执行 `docker` 命令。
- 下载 Demo （包含一个 provider 和一个 consumer 程序）。

操作步骤

步骤一：初始化镜像仓库

首次使用镜像仓库时，需要进行初始化操作，设置登录仓库的密码。

TSF 会针对每个容器应用创建一个名为 `tsf_<账号ID>/<应用名>` 的镜像仓库。

步骤二：制作镜像

1. 解压下载的 Demo 程序包，在 dockerfile 所在目录下，执行如下命令。

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 `<主账号 ID>` 对应用户**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在控制台账号信息页面获取。），`<应用名>` 表示控制台上刚刚创建的应用名。`[tag]` 为镜像的 tag，用户可自定义。

参考示例如下：

2. 命令执行完成后，执行 `docker image ls` 命令查看创建的镜像。

可查看到该镜像 tag 和 ImageId，这两个参数将用于推送镜像到镜像仓库。

步骤三：推送镜像到镜像仓库

1. 在应用列表中，单击在 [步骤3：新建应用] 中创建的应用“ID/应用名”。

2. 单击【镜像】标签页，选择【上传程序包/镜像】，可以获得查看登录镜像仓库、拉取镜像和推送镜像到仓库的命令。
3. 复制【使用指引】中登录 docker registry 的命令并执行。

```
sudo docker login --username=<账号 ID> ccr.ccs.tencentyun.com
```

> **注意：**
>
> 用户需要输入两次密码，首次为 sudo 密码，第二次为镜像仓库登录密码。

命令行工具显示 `Login Succeeded` 即表示登录成功。

4. 登录成功后，复制【使用指引】中给镜像打tag的命令并执行。

```
sudo docker tag [ImageId] ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

其中 【ImageId】 和 【tag】是在制作镜像时获取。

> **注意：**
>
> 若此时需要输入密码，请输入sudo 密码。

5. 复制【使用指引】中推送镜像到仓库的命令并执行，其中 【tag】 和步骤4相同。

```
sudo docker push ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

运行结果如下：

6. 推送镜像成功后，在控制台刷新页面，可以看到上传镜像仓库中的镜像。

步骤5：部署应用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在创建应用并上传镜像后在 TSF 控制台部署应用。

操作步骤

1. 在应用列表中，单击在 【步骤3：新建应用】中创建的应用"ID/应用名"。
2. 选择【部署组】标签页，单击【新建部署组】。

3. 设置部署组相关信息。

- **组名**：部署组的名称。
- **集群**：选择 【步骤1：新建容器集群】中创建的集群。
- **命名空间**：选择集群关联的默认命名空间。
- **日志配置项**：用于采集应用的业务日志数据，此处可选择**无**。
- **日志投递**：用于日志转储，此处可选择**无**。

4. 单击【保存&下一步】，进入部署应用页面。

5. 设置部署相关信息。

- **选择镜像**：选择 【步骤4：上传镜像】 中推送到镜像仓库的镜像版本。
- **启动参数**（选填）：设置 Java 应用的启动参数。
- **资源配置**：应用容器的 CPU 和内存限制使用默认值即可，实例数设置为1。
- **访问设置**：
- 网络访问方式决定了部署组内应用的网络属性，不同访问方式的应用可以提供不同网络能力。此处设置**集群内访问**。
- 端口映射中选择 TCP 协议，容器端口和服务端口设置为8080。

6. 单击【提交】，完成应用部署。应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

7. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

步骤6：验证服务调用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 上部署一个 provider 服务和一个 consumer 服务并验证服务之间的调用功能。

操作步骤

1. 使用与 **[步骤1-步骤5]** 相同的流程在同一个集群和命名空间下部署一个 consumer 服务。

2. 在服务治理页面，选择集群和命名空间后，可以看到 provider 和 consumer 服务的运行状况。

3. 在集群列表页面，单击集群的"ID/集群名"，进入云主机列表。

4. 单击 consumer 服务所在云服务器操作栏的【登录】，输入登录密码，登录云服务器。

5. 在 consumer 服务容器中访问 provider 服务。 执行命令查看容器 ID：

```
sudo docker ps #查找容器 ID
```

进入容器内部：

```
sudo docker exec -it <容器id> /bin/bash #进入容器内部
```

执行 curl 命令调用 provider 服务：

```
curl localhost:18083/echo-rest/test
```

调用结果如下：

6. 在服务治理页面，单击 provider 服务的"微服务名称"，进入服务详情页面，可以看到两个服务的依赖关系

在选中时间范围内，consumer-demo 调用了 provider-demo 服务，调用成功比例为 100%（绿色部分）。其中平均每次调用耗时 1.91ms，请求频率为每分钟59.8次。

使用虚拟机部署微服务
步骤1：新建虚拟机集群

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在TSF控制台上创建一个虚拟机集群。

前提条件

- 已【获取访问授权】
- 已【购买云服务器】
- 已【购买节点】

操作步骤

1. 登录【TSF 控制台】

2. 在左侧导航栏中，单击【集群】，进入集群列表页。

3. 在集群列表页的左上方，单击【新建】。

   设置集群的基本信息。

- **集群类型**：选择 **虚拟机集群**。
- **集群名**：填写集群名称。
- **标签**：用于分类管理资源，可不选。
- **所在可用区**：选择默认可用区。
- **集群网络**：选择与已有云服务器相同的 VPC 网络，用来保证后续导入集群的云服务器属于同一 VPC。
- **数据集**：选择"无"。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限。
- **备注**：选填，可留空。

4. 单击【提交】，可以在集群列表看到创建好的集群，集群状态变为运行中即可进行后续导入云主机操作。

步骤2：导入云主机

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在新建集群后将准备好的云主机导入集群。

操作步骤

1. 在集群列表页面，单击刚刚创建好的集群操作栏的【导入云主机】。
2. 从集群所在 VPC 的云服务器列表中，选择需要添加到集群的主机，单击【下一步】。

3. 填写相关信息。

- **导入方式**：选择重装系统。
- **JDK 类型**：推荐选择 KONA JDK
- **操作系统**：根据实际选择。
- **登录方式**：提供三种对应登录方式 a.设置密码：请根据提示设置对应密码。 b.立即关联密钥：密钥对是通过一种算法生成的一对参数，是一种比常规密码更安全的登录云服务器的方式。 c.自动生成密码：自动生成的密码将通过站内信发送给您。

4. 单击【提交】，集群列表中云主机数量将变为1。

5. 单击集群的"ID/集群名"，进入云主机列表，等待几分钟后，云主机的可用状态将变为"可用"。

步骤3：创建应用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台创建一个应用。

操作步骤

1. 在左侧导航栏，单击【应用管理】，进入应用列表页。
2. 在应用列表上方单击【新建应用】。

3. 设置应用信息后，单击【提交】。

- **应用名**：填写 provider。
- **部署方式**：选择 **虚拟机部署**。
- **业务类型**：选择 **业务应用**。
- **开发语言**：选择 **JAVA**。
- **开发框架**：选择 **SpringCloud**。
- **应用类型**：选择 **普通应用**。
- **标签**：用于分类管理资源，可不选。
- **数据集**：选择"无"。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限
- **备注**：选填，可留空。

4. 在弹出的弹窗中单击【确认】，前往导入程序包并部署应用。

步骤4：上传程序包

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台创建应用后上传程序包。

前提条件

下载 Demo（包含一个 provider 和一个 consumer 程序）

操作步骤

1. 在程序包管理页面，单击【上传程序包】。

2. 在**上传程序包**对话框中填写相关参数，单击【提交】。

- **上传程序包**：单击【选择文件】，选择提前准备好的 Demo 中的 provider 程序包。
- **程序包版本**：填写版本号，或单击【用时间戳作为版本号】。
- **备注**：填写备注。

3. 上传完程序包后，在弹出的弹窗中，单击【前往部署】，前往部署应用。

步骤5：部署应用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在创建应用并上传程序包后在 TSF 控制台部署应用。

操作步骤

1. 在部署组页面，单击【新建部署组】。

2. 填写部署组相关信息。

- **组名**：部署组的名称，不超过60个字符。
- **集群**：选择 [步骤1：新建虚拟机集群] 中创建的集群。
- **命名空间**：选择集群关联的默认命名空间。
- **日志配置项**：应用的日志配置项用于指定 TSF 采集应用的日志路径。此处可选择**无**。
- **日志投递**：用于日志转储，此处可选择**无**。

3. 单击【保存&下一步】，选择 [步骤2：导入云主机] 中导入集群的云主机。

4. 单击【部署应用】，填写部署信息。

- 软件仓库：选择默认仓库。
- 程序包类型：选择 jar 包。
- 程序包/版本：勾选 [步骤4：上传程序包] 中上传的程序包。
- 启动参数：选填。
- 更新方式：选择立即更新。
- 健康检查：可选。

5. 单击【完成】，应用部署成功后，部署组中**已启动/总机器数**的数值发生变化。

6. 在服务治理页面，选择地域和应用关联的命名空间后，可以看到服务实例显示在线状态，表示服务注册成功。

步骤6：验证服务调用

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 上部署一个 provider 服务和一个 consumer 服务并验证服务之间的调用功能。

操作步骤

1. 使用与 **[步骤1-步骤5]** 相同的流程在同一个集群和命名空间下部署一个 consumer 服务。

2. 在服务治理页面，选择集群和命名空间后，可以看到 provider 和 consumer 服务的运行状况。
3. 在集群列表页面，单击集群的"ID/集群名"，进入云主机列表。

4. 单击 consumer 服务所在云服务器操作栏的【登录】，输入登录密码，登录云服务器。
5. 执行 curl 命令调用 provider 服务。

```
curl localhost:18083/echo-rest/test
```

调用结果如下： 6. 在服务治理页面，单击 provider 服务的"微服务名称"，进入服务详情页面，可以看到两个服务的依赖关系。

在选中时间范围内，consumer-demo 调用了 provider-demo 服务，调用成功比例为 100%（绿色部分）。其中平均每次调用耗时 1.91ms，请求频率为每分钟59.8次。

操作指南
资源管理
集群基本操作

最近更新时间: 2025-02-18 16:02:00

操作场景

集群是指云资源管理的集合，包含了运行应用的云主机等资源。集群包括虚拟机集群、容器集群和 Serverless 集群（内测中）三种类型。TSF 平台只支持 VPC 内的集群。

集群生命周期说明

状态	说明
创建中	集群正在创建，正在申请云资源。
运行中	集群正常运行。
删除中	集群在删除中。
异常	集群中存在异常，如节点网络不可达等。
闲置中	容器集群内没有云主机，处于完全不可用状态，通过导入云主机方式唤醒集群。

操作步骤

新建集群

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中，单击 【集群】。
3. 在集群列表页的左上方，单击【新建集群】。
4. 设置集群的基本信息。

- **集群类型**：选择集群类型。
- **虚拟机集群**：虚拟机集群中的主机用于部署虚拟机应用。
- **容器集群**：容器集群中的主机用于部署容器应用。
- **Serverless 集群**：Serverless 集群中的主机用于部署 Serverless 应用。
- **集群名称**：集群名称，不超过60个字符。
- **新建类型**：适用于容器集群类型。
- **直接创建**：通过 TSF 控制台创建容器集群，调用容器服务 TKE 新建集群 API 创建容器集群（Kubernetes 版本1.10.5，每个集群256个 Service，每个节点256个 Pod，每个集群255个节点）。
- **导入容器服务 TKE 集群**：用户已经在容器服务 TKE 上已经创建好了容器集群，导入到 TSF 集群列表。这种方式适用于更灵活的容器集群配置场景（如制定 Kubernetes 版本、独立部署模式集群）。
- **集群网络**：为集群内主机分配在主机网络地址范围内的 IP 地址。
- **容器网络**（仅适用于直接创建方式的容器集群）：为集群内容器分配在容器网络地址范围内的 IP 地址。

- **数据集**：将新建的集群添加到已有数据集中。
- **集群描述**：集群的描述，不超过200个字符。

5. 单击【提交】，完成创建。

集群创建成功后，需要导入云主机。 创建容器集群后，在集群列表的集群状态栏可以查看容器集群创建进度。

导入云主机

1. 在集群列表页中，单击目标集群的**ID/集群名**。
2. 在云主机列表标签页中，单击左上方的 【导入云主机】按钮。
3. 从集群所在 VPC 的云主机列表中，选择需要添加到集群的云主机。

4. 选择导入方式，并根据页面提示完成操作。

- 重装系统：该方式会使用 root 用户安装 agent。
- 安装 Agent（仅适用于虚拟机集群）： 该方式支持使用 root 用户和非 root 用户安装 agent。

您可以点击以下页签，查看对应导入方式的操作步骤。

> **注意：**
>
> CVM 竞价实例不支持重装系统。

5. 云主机配置

- 导入方式：支持重装系统部署或者安装 Agent 部署。使用安装 Agent 部署时，不会重装机器。
- JDK 类型：Kona JDK 是基于 openJDK 开发的 TencentJDK，修复开源老版本 JDK 高风险漏洞，针对云应用场景优化并支持更便捷的分析诊断工具。
- 操作系统：支持两种不同操作系统选择：Ubuntu 18.04版本与 CentOS 7.5版本。
- 登录方式：提供三种对应登录方式。 设置密码：请根据提示设置对应密码。 立即关联密钥：密钥对是通过一种算法生成的一对参数，是一种比常规密码更安全的登录云主机的方式。详请参阅 SSH 密钥。 自动生成密码：自动生成的密码将通过站内信发送给您。

- 安全组：安全组具有防火墙的功能，用于设置云主机 CVM 的网络访问控制。

6. 导入的云主机将出现在云主机名列表中。等待几分钟，刷新列表，正常情况下云主机的状态将变为运行中，可用状态变为"可用"。

安装：

> - 当前仅支持添加同一 VPC 下的云主机。
> - 导入方式为重装系统时，云主机的操作系统将被重装。

7. 在云主机列表中，单击操作列的【安装 Agent】，并选择 JDK 类型。

8. 复制脚本（下图脚本为图片，**不能复制**，用户需要在控制台上对应弹框中单击【复制】）。

9. 登录云主机。

- root 用户安装 agent：直接粘贴脚本并执行。
- 非 root 用户安装 agent：需要 root 用户新建好 `/data/tsf_apm/` 和 `/var/log/tsf/` 这两个目录，并赋予读写权限给tsf-agent的安装用户。tsf-agent 的安装用户就是部署应用的启动用户。粘贴脚本并执行。

0. 下图为脚本执行成功后终端最后几条打印日志的截图。

1. 脚本执行成功后，回到云主机列表，稍等片刻，云主机的可用状态会变为"可用"。

导入云主机后，您可以开始创建并部署应用。

移出云主机

> **注意：**
>
> 只有当节点上没有运行应用时，才可以将节点从集群中移出。停止应用需要到部署组的实例列表页面中操作。

1. 在集群列表页中，单击目标集群的**ID/名称**，进入集群的节点列表页面。
2. 在集群节点列表页面，选择需要移出的云主机，单击操作列的【删除】。
3. 在弹出的提示框中，单击【确定】将云主机移出节点。

删除集群

> **注意：**
>
> - 集群内存在云主机时，无法删除集群，您需要先移除云主机。
> - 集群在删除期间，无法对外提供服务，请提前做好准备，以免造成影响。

1. 在集群列表页中，单击目标集群操作列的【删除】。
2. 在弹出的提示框中，单击【确定】删除集群。

查看容器集群创建事件

登录 【TSF 控制台】 ，在集群页面新建容器集群后，在集群列表的集群状态栏可以查看容器集群状态。

- 闲置中：容器集群内没有云主机，处于完全不可用状态，可以通过导入云主机方式唤醒集群或者删除集群。

- 创建中：集群正在创建，正在申请云资源，点击集群状态栏的【查看进度】可查看创建进度。
- 运行中：集群正常运行。
- 异常：集群中存在异常，如节点网络不可达等，可点击查看异常原因。

集群及容器网络设置

最近更新时间: 2025-02-18 16:02:00

集群网络与容器网络是集群的基本属性。通过设置集群网络和容器网络可以规划集群的网络划分。

- **集群网络**：将为集群内主机分配在节点网络地址范围内的 IP 地址，您可以选择私有网络中的子网用于集群的节点网络。
- **容器网络**：将为集群内容器分配在容器网络地址范围内的 IP 地址，您可以自定义三大私有网段作为容器网络， 根据您选择的集群内服务数量的上限，自动分配适当大小的 CIDR 段用于 kubernetes service，同时容器网络自动为集群内每台云主机分配一个24位的网段用于该主机分配 Pod 的 IP 地址。

集群网络与容器网络的关系

- 集群网络和容器网络网段不能冲突；
- 同一 VPC 内，不同集群的容器网络网段不能冲突；
- 容器网络和 VPC 路由冲突时，优先在容器网络内转发。

集群网络与其他资源通信

- 集群内容器与容器之间互通；
- 集群内容器与节点直接互通；
- 集群内容器与【数据库 TencentDB】、【云数据库 Redis】、【云数据库 Memcached】等资源同一 VPC 下内网互通。

命名空间管理

最近更新时间: 2025-02-18 16:02:00

命名空间（Namespace）是对一组资源和对象的抽象集合，用于对服务相互访问的隔离，在网络连通性的前提下，同一命名空间内的服务可以相互发现和相互调用。

命名空间分类

命名空间有三种类型：

类型	个数限制	作用
系统命名空间	创建每个集群时会自动创建一个	命名规则是`<cluster-name>_default`，不支持绑定到其他集群。
非全局命名空间	每个用户可以创建多个	不同非全局命名空间内的服务之间不能相互调用。
全局命名空间	每个用户在每个地域下只能创建一个	非全局命名空间内的服务可以调用全局命名空间内服务 • 全局命名空间内服务支持相互调用，不支持调用非全局命名空间内的服务。

> **注意：**
>
> - 全局命名空间特性仅适用于1.18.0版本之后的 SDK。
> - 原生应用不支持全局命名空间特性。

使用场景

使用命名空间划分开发环境和测试环境

场景：用户希望部署一套开发环境和一套测试环境，两套环境支持部署同一个应用不同版本的程序包。两套环境可以共享一个集群作为底层部署资源。

解决方案：

1. 创建两个非全局命名空间分别为 *dev-env* 和 *test-env* 作为开发环境和测试环境。
2. 创建一个虚拟机集群，并绑定 *dev-env* 和 *test-env* 命名空间。
3. 创建应用（如 *promotion*），*并创建 2 个部署组（dev-promotion* 和 *test-promotion*），分别选择 *dev-env* 和 *test-env* 命名空间。
4. 部署组 *dev-promotion* 部署程序包版本 v1.1，部署组 *test-promotion* 部署程序包版本 v1.0。

> **注意：**
>
> 如果用户希望开发和测试环境的部署资源是隔离的，可以使用两个集群来划分开发和测试环境。

实现服务跨集群访问

在网络连通性的前提下，同一命名空间内的服务可以相互发现和相互调用。如果要实现**跨集群的服务访问**，有两个前提：

1. 集群内实例网络互通。
2. 集群关联相同的命名空间。**多个集群通过命名空间名称（而不是命名空间 ID）作为关联的 key。**用户可以在命名空间一级页面中看到关联的集群信息。

例如：有2个集群分别是 cluster-1 和 cluster-2，两个集群内实例网络互通（如在相同 VPC 内），并且都关联了命名空间 test-namespace。要实现跨集群访问，需要确保在创建部署组 provider-group 和 consumer-group 时，使用相同的命名空间 test-namespace。

使用全局命名空间来部署公共服务

场景：在电商场景中，订单业务和物流业务都希望访问用户信息服务。此时可以使用不同命名空间来划分业务领域，将相对独立的业务（如订单和物流业务）部署在非全局命名空间，将公共服务（如用户信息服务）部署在全局命名空间中。

解决方案：

1. 创建两个非全局命名空间 *order-biz* 和 *logistics-biz* 作为订单和物流的业务领域。
2. 创建全局命名空间 *common-biz* 作为公共服务的业务领域。
3. 将订单业务的服务部署在 *order-biz* 命名空间内，将物流的服务部署在 *logistics-biz* 内，将用户信息服务部署在 *common-biz* 内。
4. *order-biz* 和 *logistics-biz* 命名空间内的服务可以访问 *common-biz* 内的用户信息服务。

命名空间基本操作

创建命名空间并关联集群

创建命名空间

1. 登录 TSF 控制台 ，在左侧导航栏中，单击【命名空间】。
2. 在命名空间列表页的左上方，单击【新建命名空间】。
3. 在命名空间对话框内填写名称，并勾选是否为全局命名空间。
4. 单击【提交】，完成创建。

集群关联命名空间

5. 选择目标集群，进入集群详情页，单击【命名空间】标签页。
6. 在命名空间列表左上方，单击【关联命名空间】。
7. 选择要关联的命名空间，并单击【提交】。

> **注意：**
>
> 集群仅支持关联自定义命名空间。

删除命名空间

前提条件：删除命名空间前，需要先解除命名空间与集群的绑定关系（解除命名空间绑定时，需要先删除具有该命名空间属性的全部部署组）。

解除关联集群

1. 登录 TSF 控制台 ，在左侧导航栏中，单击【集群】。
2. 选择目标集群，进入集群详情页，单击【命名空间】标签页。
3. 在命名空间列表的操作列，单击【解除绑定】。

删除集群

4. 在控制台左侧导航栏中，单击【命名空间】。
5. 在命名空间列表的操作列，单击【删除】。
6. 单击【确认】，即可删除命名空间。

安全组设置

最近更新时间: 2025-02-18 16:02:00

用户将容器集群导入云主机时需要设置安全组。

此外，如果从 API 网关访问容器集群中的微服务，则需要在安全组中**增加服务的监听端口。**例如，provider - demo 的监听端口是18081，需要新增入站规则如下：

协议	端口号	网段	是否允许
TCP	18081	0.0.0.0/64295985640140800	允许

实例查询

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台中，通过搜索实例 ID、实例名称和实例 IP 查看节点所在集群、命名空间、部署组等信息。

操作步骤

使用方法如下：

1. 登录 TSF 控制台，在左侧导航栏单击【集群】。
2. 在集群页面，单击【实例查询】。
3. 在搜索框中输入实例信息，输入实例 ID、实例名称和实例 IP，即可查询所在集群、命名空间和部署组等信息。

搜索说明

- 支持同时搜索多个实例，实例之间使用 " | " 分隔。
- 当用户直接在搜索框中输入0 - 255之间数字或数字 + 点的组合时，会默认按照 IP 进行搜索。当用户输入 “ins - ” 开头的字符串将按照实例 ID 搜索，其他按照实例名称搜索。

- 用户点击搜索框选择过滤条件，选择实例 ID、IP 或名称作为过滤条件，添加过滤条件后请回车并搜索。

应用中心
虚拟机部署应用
部署组管理（虚拟机）

最近更新时间: 2025-02-18 16:02:00

部署组基本操作如下：

功能	说明
应用扩容	将 CVM 云服务器添加到部署组中，如果部署组此时已经关联了程序包，将执行部署命令。
部署应用	将应用部署到 CVM 云服务器上，并执行启动命令。
下线实例	停止 CVM 云服务器上的应用，然后将实例从部署组中移除。
停止应用	将部署组中所有的节点上运行的应用停止，停止后可以再启动。
启动应用	当应用处于停止状态时可以启动应用。

创建部署组

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中，单击【部署组】。
3. 在页面顶部选择集群。
4. 单击部署组列表上方的【新建部署组】。
5. 设置部署组相关信息。

- **部署组名称**：部署组的名称，不超过60个字符。
- **命名空间**：选择命名空间。
- **关联应用**：关联应用字段决定了后续程序包来源和应用配置来源。
- **日志配置项**：指定部署组内实例的业务日志采集规则。如果配置为"无"，将不采集业务日志。更多关于日志配置项的说明请参考 【日志配置项】。
- **日志投递**：指定日志的转储方式，将规则指定路径中的日志内容投递到指定的接收端。如果配置为"无"，将不投递业务日志。更多关于日志投递的功能说明请参考 【日志投递】 。

6. 单击【保存&下一步】，选择当前集群下可用的云主机实例。
7. 单击【部署应用】，按照 【部署应用】 进行操作。

部署应用

1. 单击部署组列表页右侧的【部署应用】。
2. 选择目标程序包版本。

程序包类型	说明
jar	启动命令固定为 java -jar，支持用户设置启动参数

程序包类型	说明
war	启动命令固定为 java -jar，支持用户设置启动参数
zip/tar.gz	程序包类型为zip/tar.gz时，支持【启停脚本配置】。使用本地start.sh和stop.sh：默认方式控制台配置：需要填写启动脚本和停止脚本（推荐）当部署组内的实例 agent 版本不支持【控制台配置】启停脚本时，不能选中该选项，请升级 agent 到最新版本。

3.（可选）设置启动参数。

> **注意：**
>
> 如果部署组关联的应用是 Mesh 应用，则无须设置启动参数。

4. 选择更新方式。

更新方式	说明
立即更新	会先停止所有运行实例，然后使用新的程序包版本部署，会造成发布期间短暂停服。
滚动更新	当部署组内有多个实例时，可以选择滚动更新方式进行分批发布。滚动更新支持设置发布策略：**beta 批次**：是否首次用一个实例来部署新版本，如果部署成功，才会部署后面的批次实例。**批次 N 实例占比**：除了 beta 批次，可以将部署组内剩余实例按百分比划分为多个批次部署，只有当批次 N 部署成功后才会继续部署批次 N+1。**分批执行方式**：批次之间可以选择自动或手动方式来开启下一批次的部署。**分批等待时间**：如果【分批执行方式】选择自动时，等待分批等待时间后会自动部署下一批次。

5.（可选）选择健康检查方式。

> **注意：**
>
> - 当部署组内的实例 agent 版本不支持存活检查和就绪检查时，不能开启检查功能，请升级 agent 到最新版本。
> - 当部署组内的多个实例 agent 版本不一致时，不能开启检查功能。只有当所有实例的 agent 的版本支持存活检查和就绪检查才开启该特性。

> -

健康检查方式	说明
存活检查	检查应用是否正常，不正常则重启实例。
就绪健康	检查应用是否就绪，不就绪会影响滚动更新。

检查方式	说明
HTTP 请求检查	任何大于200小于400的返回码都会认定是成功的返回码。其他返回码都会被认为是失败的返回码。HTTP 检查需要设置端口和请求路径。
TCP 端口检查	如果可以建立连接被认为是成功的。该检查方式需要设置检查端口。

检查方式	说明
执行命令检查	如果命令执行成功并且返回值为 0，认为是成功；其他返回值认为是失败。该检查方式需要填写执行命令。

高级参数：

参数	说明	参数范围
启动延时	延时启动健康检查的时间	最小值为0，默认值为10
超时时间	每次健康检查响应的最大超时时间	最小值为1，默认值为2
检查周期	进行健康检查的时间间隔	最小值为1， 默认值为10
健康阈值	表示从失败到成功的连续健康检查成功次数	存活检查不支持编辑，只能为1就绪检查支持编辑：最小值为1，默认值为1
不健康阈值	表示从成功到失败的连续健康检查成功次数	最小值为1，默认值为3

6. （可选）填写描述信息。
7. （可选）开启强制启动，开启强制启动则实例忽视consul服务注册报错信息正常启动。
8. 单击【完成】。应用部署成功后，部署组中的 **已启动/总机器数** 数值发生变化。

应用扩容

1. 单击部署组列表右侧的【更多】>【添加实例】。
2. 选择要添加进部署组的云服务器 CVM，单击【提交】。

> **注意：**
>
> 部署组如果已选择在控制台配置启停脚本，扩容时如果实例 agent 版本不支持该特性，请升级 agent 到最新版本。

3. 在部署组的实例列表页面中显示出刚才添加的 CVM。

下线实例

1. 单击部署组列表右侧的【更多】>【下线实例】。
2. 选择要下线的实例。
3. 在弹出的确认页面中，单击【提交】。

停止应用

1. 单击部署组列表页右侧的【更多】>【停止应用】。
2. 在弹出的确认页面中，单击【确认】。

启动应用

1. 单击部署组列表页右侧的【更多】>【启动应用】。

2. 在弹出的确认页面中，单击【提交】。

删除部署组

下线部署组内所有实例，才能执行部署组的删除操作。

1. 勾选您要删除的部署组，单击左上角的【删除】。
2. 在弹出的确认页面中，单击【提交】。

应用管理

最近更新时间: 2025-02-18 16:02:00

操作场景

应用管理包括创建应用和删除应用，您可以按照以下步骤进行操作。

操作步骤

创建应用

1. 登录 【TSF 控制台】。
2. 在左侧导航栏，单击【应用管理】。
3. 在应用列表左上方，单击【新建应用】。

4. 设置应用信息后，单击【提交】完成创建。

- **部署方式**：选择**虚拟机部署**。
- **业务类型**：选择业务应用或者微服务网关应用。
- **开发语言**：选择您的开发语言。
- **开发框架**：选择您的开发框架。
- **应用类型**：
- 普通应用：适用于 Spring Cloud 或者 Dubbo 应用。
- Mesh 应用：适用于 Service Mesh 方式接入。
- 微服务网关应用：适用于微服务网关（Zuul、Spring Cloud Gateway）。
- 原生应用：适用于 Spring Cloud 原生应用
- **标签**：用于分类管理资源，可不选。
- **数据集**：选择“无”。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限。
- **备注**：选填，可留空。

5. 单击【提交】完成应用创建。

创建应用后，需要添加实例并部署应用。

删除应用

> **注意：**
>
> - 删除应用会同时删除应用关联的程序包。
> - 当应用下有部署组时，无法执行删除操作，需要先删除所有部署组后才能删除应用。

1. 在应用列表的操作列，单击【删除】。
2. 在确认弹框中，单击【确认】完成删除。

程序包管理

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台上传程序包或删除程序包。

操作步骤

上传程序包

1. 登录 【TSF 控制台】。
2. 在左侧导航栏 选择【应用管理】，单击目标应用的**ID/应用名**。
3. 在应用详情页，选择**程序包管理**标签页，单击【上传程序包/镜像】。

- 程序包类型：可选择jar包，war包，tar.gz/zip压缩包

- 上传程序包：单击【选择文件】，选择提前准备好的压缩包。程序包格式说明参考[程序包格式说明]。
- 程序包版本：填写版本号。
- 备注：填写备注。

4. 单击【提交】，程序包上传成功后出现在程序包列表中。

上传程序包后，您需要创建部署组并部署应用，详情参考创建部署组。

删除程序包

1. 在控制台页面，单击左侧导航栏 【应用管理】。
2. 在应用管理列表页，单击目标应用的**ID/应用名**。
3. 在目标应用详情页，单击**程序包管理**标签页，单击程序包列表右侧的【删除】。
4. 单击【确认】完成删除。

服务治理
系统和业务自定义标签

最近更新时间: 2025-02-18 16:02:00

标签说明

TSF 引入**标签**概念以区分不同的请求来源，TSF 标签包括系统标签和业务自定义标签。

- **系统标签** 每一个 TSF 上运行的服务都已经被预先设置好了某些标签，如发起请求的服务消费方所在的部署组、IP、服务发起方的版本号等。
- **业务自定义标签** 在实际的使用中，如果系统自带标签不能保证用户使用的场景，用户可以自定义标签内容。对于 Spring Cloud 应用，TSF 提供了用户配置自定义标签的 SDK， 对于 Mesh 应用，用户需要在 header 中设置标签。

标签表达式

用户在控制台创建服务治理规则时，可以选择通过设置**标签表达式**区分请求来源。多个标签表达式之间是逻辑与（AND）的关系。例如两条标签表达式分别是：

- 系统标签主调服务名等于 consumer-demo
- 自定义标签 userid 等于123456

只有当一条请求是 consumer-demo 发出，**且**带有 userid 是123456的自定义标签时才满足上面2个标签表达式。

一条标签表达式中，逻辑关系与值的个数对应如下：

逻辑关系	值个数
包含（IN）	多个
不包含（ NOT IN）	多个
等于（==）	一个
不等于（!=）	一个
正则表达式（regex）	一个

服务统计

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 支持从主调和被调两个视角展示服务指标的统计信息。用户可以通过统计信息了解服务指标的变化情况。 服务指标以天为单位进行统计，在当天只能查看**昨天之前**的统计数据。

功能说明

- 一个服务可能既是主调服务，又是被调服务。TSF 会统计服务作为主调服务调用其他服务及接口的情况，以及服务作为被调，其接口被其他服务调用的情况。
- TSF 服务统计支持查看指标的日环比和周同比。
- 统计指标：

  a. 平均响应时间。

  b. 按照状态码统计：2xx 响应、3xx 响应、4xx 响应、5xx 响应、其他状态码响应。

  c. 按照异常请求统计：超时响应、不可用响应。其中超时响应表示服务端处理超时的请求响应，不可用响应表示服务端无可用实例时的异常请求响应。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【服务治理】，并单击某个服务的 ID 进入服务详情页。
3. 在服务详情页，单击顶部的【统计】，进入统计页面。
4. 选择主调或被调视角、统计日期。

- **主调视角**下会展示被调方服务的指标列表。单击某个被调服务前的箭头会展示该服务不同接口被调用的统计信息。
- **被调视角**下会展示该服务的接口指标列表。单击某个接口前的箭头会展示该接口被不同主调服务调用的统计信息。

5. 单击操作列的【查看日环比/周同比】，可以查看各指标日环比/周同比情况。

- 日环比表示查询日 T 和之前一天 T - 1 的指标数据对比。
- 周同比表示查询日 T 和查询日前7天 T - 7 的指标数据对比。

服务鉴权原理

最近更新时间: 2025-02-18 16:02:00

服务鉴权是处理微服务之间相互访问权限问题的解决方案。配置中心下发鉴权规则到服务，当请求到来时，服务根据鉴权规则判断鉴权结果，如果鉴权通过，则继续处理请求，否则返回鉴权失败的 HTTP 状态码403（Forbidden）。

鉴权原理

鉴权流程如下：

服务鉴权功能支持白名单和黑名单两种鉴权方式。

- **白名单**：当请求匹配**任意一条**鉴权规则时，允许调用；否则拒绝调用。
- **黑名单**：当请求匹配**任意一条**鉴权规则时，拒绝调用；否则允许调用。

以下视频将为您介绍 TSF 服务鉴权原理：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2038-24377?source=gw.doc.media&withPoster=1&notip=1

多个鉴权规则

一个服务可能有多个鉴权规则，多个鉴权规则之间是逻辑或（OR）的关系，只要请求满足任意一条鉴权规则，就相当于匹配成功。

示例说明

示例1

需求：服务 provider-demo 只允许来自 consumer-demo 服务**且**带有 user=foo 的自定义标签的请求调用。 **解决方案**：要满足上面的鉴权需求，用户可以在 provider-demo 的鉴权页面，设置鉴权方式为白名单，鉴权规则如下图（注意最后要将生效状态改为生效）： **结论**：要满足 `逻辑与 AND` （既满足条件 A ，又满足条件 B）时，需要使用标签表达式。

示例2

需求：服务 provider-demo 只允许来自 consumer-demo 服务**或**带有 user=foo 的自定义标签的请求调用。 **解决方案**：要满足上面的鉴权需求，用户可以在 provider-demo 的鉴权页面，设置鉴权方式白名单，创建2条鉴权规则，如下图： **结论**：要满足 `逻辑与 OR` （满足条件 A 或 条件 B）时，需要使用多条鉴权规则。

说明

白名单鉴权方式示例： 鉴权规则内容是 username 等于 foo，当请求中带有 username=foo 的 tag 时，因为匹配规则，服务允许调用；当请求中带有 username=bar 的 tag 时，因为不匹配规则，服务拒绝调用。

黑名单鉴权方式示例：鉴权规则内容是 username 等于 foo，当请求中带有 username=foo 的 tag 时，因为匹配规则，服务**拒绝**调用；当请求中带有 username=bar 的 tag 时，因为不匹配规则，服务允许调用。

服务鉴权使用说明

最近更新时间: 2025-02-18 16:02:00

使用鉴权功能时，用户需要先在客户端配置依赖项，然后在 TSF 控制台设置鉴权规则。

步骤1：配置依赖项

- 对于 Spring Cloud 应用，请参考开发者手册中的 【服务治理】。
- 对于 Mesh 应用，无须额外配置。

步骤2：设置鉴权规则

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【服务治理】。
3. 在服务列表，单击服务名，进入服务详情页，选择【服务鉴权】页签。
4. 在服务鉴权页面，鉴权方式选择【不启用】，单击【新建鉴权规则】，填写鉴权规则和生效状态。

- 不启用：关闭鉴权功能。
- 白名单：匹配任意一条规则的请求，允许调用。

- 黑名单：匹配任意一条规则的请求，拒绝调用。

5. 鉴权规则创建完成后，选择已生效规则对应的鉴权方式。

> **注意：**
>
> 至少有一条规则确认生效后，才能开启对应的鉴权方式。例如只有一条黑名单规则生效，则只能切换到黑名单鉴权方式，不能切换到白名单鉴权方式。

6. 在该鉴权方式下，可继续添加鉴权规则。

步骤3：切换鉴权方式（可选）

用户可以通过控制台，从一种鉴权模式切换到另外一种鉴权模式。

- 白名单切换到黑名单（或黑名单切换到白名单）：不能直接切换，需要先切换到不启用，生效一条黑名单（白名单）规则后，才能切换到黑名单（白名单）。
- 白名单（或黑名单）切换到不启用：关闭鉴权功能。

步骤4：检查鉴权效果

以官网 Demo 为例说明如何验证鉴权功能。consumer-demo 中已包含鉴权依赖 jar 包，因此这里只需要说明在控制台上创建鉴权规则用来限制特定 API 的调用。

consumer-demo 中提供了三个 API `/echo-rest/{str}` 、 `/echo-async-rest/{str}` 、 `/echo-feign/{str}` 。在控制台上新建鉴权规则，鉴权方式为**白名单**，鉴权规则的标签表达式： 创建好规则后，登录机器，使用 curl 命令来验证鉴权是否生效。

命令	预期
`curl IP:PORT/echo-rest/hello?user=test`	正常返回
`curl IP:PORT/echo-async-rest/hello?user=test`	返回鉴权失败
`curl IP:PORT/echo-feign/hello?user=test`	返回鉴权失败

限制说明

等于、不等于、包含、不包含属于严格匹配，正则表达式属于模糊匹配。因此当系统标签是被调方 API PATH 时，目前仅支持使用**正则表达式**的逻辑关系来匹配带参数的 API 请求。 例如标签的逻辑关系是正则表达式，值填写 `/echo/.*` ，可以匹配带参数的请求 /echo/test123（其中 test123 是参数）；当标签的逻辑关系是等于、不等于、包含、不包含关系，值是 `/echo/{param}` 时，不能匹配带参数的请求 /echo/test123（其中 test123 是参数）。

服务路由基本原理

最近更新时间: 2025-02-18 16:02:00

服务路由概述

用户在使用 TSF 运行自己的业务时，由于业务的复杂程度，经常需要部署数目庞大的服务运行在现网环境中。这些服务运行在属性不同的实例上、部署在不同的地域中，用户经常需要根据符合自己特定要求的属性选择服务的提供者，对服务间流量的分配起到掌控的作用。 同时，在微服务的场景下，用户研发新版本上线的迭代周期越来越快，稳定敏捷的上线新版本需要微服务框架能够支持灰度发布、金丝雀发布、滚动发布等发布方式。通过服务路由功能，用户可以配置流量分配权重，设置某些权重的流量被分配到某个版本号中，为灰度发布等上线模式提供了无需终止服务的底层能力支持。为了保证满足客户的定制化需求，TSF 支持用户定制自己的路由标签，并支持选择不同的逻辑形式配置标签值，定向分配流量。 总而言之，服务路由功能的主要作用是将调用流量按照自己的需求进行分配。

以下视频将为您介绍 TSF 的服务路由功能：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2038-24379?source=gw.doc.media&withPoster=1&notip=1

服务路由原理

要实现服务路由需要完成两部分操作：

- 在控制台上，给**服务端（服务提供者）**设置路由规则。
- **客户端（服务消费者）**获取路由规则， 根据规则来分发请求。

以 `user -&gt; shop -&gt; promotion` 为例说明服务路由的原理，三个服务特点如下：

- user：Spring Cloud 应用，使用路由 SDK。
- shop：Mesh 应用，有两个版本 v1 和 v2.0-beta。
- promotion：Spring Cloud 应用，有两个版本 v1 和 v2.0-beta。

服务调用和路由情况如下图所示。用户需要在控制台创建如下路由规则：

- shop 服务详情页中配置路由规则：90%的流量分配到 v1 版本，10%的流量分配到 v2 版本。
- promotion 服务详情页中配置路由规则：服务名等于 shop 且版本号为 v1 的流量 100% 分配到 v1 版本，服务名等于 shop 且版本号为 v2 的流量100%分配到 v2 版本。

服务路由文档最佳实践

最近更新时间: 2025-02-18 16:02:00

灰度发布

- 使用目的：当用户需要上线新的功能时，希望使用灰度发布的手段在小范围内进行新版本发布测试。
- 使用方法：用户可以将新的程序包上传到原有的应用中。用户选择按照权重的方式配置路由规则，填写权重大小，并选择目标版本版本号，便可以实现使用部分流量进行灰度发布的能力。生效中的权重可以被编辑，实时生效，间接实现了滚动发布的功能。

同地机房优先

- 使用目的：当企业规模较大时，单个机房的容量已经不能满足业务需求，业务经常出现跨机房部署的情况。然而由于异地跨机房调用出现的网络延迟问题，需要能够保证服务消费方能优先调用本地的服务消费方，这就需要采用服务路由的方式。
- 使用方法：用户选择系统自带标签路由选项，配置系统自带标签为发起方 IP，在正则表达式中填写服务消费方的 IP 字段规则。对于服务提供方，用户可以将 IP 地址相近的实例归属在同一个部署组上，作为目标部署组，实现优先调用同地机房。

部分帐号内测

- 使用目的：希望配置某些使用者使用的版本为新的内测版本。
- 使用方法：用户可以配置自定义标签为用户 ID，设置 ID 值的正则表达式计算方式，保证服务消费方发起的请求带有以上条件的流量分配到服务提供方的某个版本号上，实现帐号内测功能。

其他实践

在实际的使用中，用户也可以通过服务路由功能，实现优先保护重要服务的运行质量、前后端分离、读写分离等功能。

API 列表

最近更新时间: 2025-02-18 16:02:00

在服务的详情页中会显示服务提供的 API 列表。API 列表显示服务对外提供的 API。 单击 API 进入详情页，可以查看到 API 的详细信息。

API 详情按照【应用名/版本号】划分显示了 API 的详细信息，包括：路径、方法、描述、入参、出参。其中 Models 表示参数中的复杂类型。

API 落库

最近更新时间: 2025-02-18 16:02:00

操作场景

API 列表显示服务对外暴露的 API 列表。API 调试提供用户在线调试 API 的能力。 您可以在 TSF 控制台中，通过 API 列表查看 API 的详细信息，并进行 API 在线调试。

前提条件

要使用 API 列表和调试功能，需要先将服务的 API 注册到注册中心，具体请参考开发手册 [API 注册] 。

操作步骤

API 列表

1. 登录 【TSF 控制台】 ，在左侧导航栏中，选择【服务治理】。
2. 单击服务名称，进入服务详情页。
3. 在服务详情页，单击顶部的【API列表】，会显示服务对外提供的 API。
4. 单击 API 名称进入详情页，可以查看到 API 的详细信息。 API 详情按照【应用名/版本号】划分显示了 API 的详细信息，包括：路径、方法、描述、入参、出参。其中 Models 表示参数中的复杂类型。

手动录入API

TSF 当前支持手动录入API。手动录入 API 的场景主要有：当微服务尚未注册到注册中心，但是希望将 API 进行提前录入，方便配置一些服务治理规则时，可以手动录入。

录入方式：

1. 登录 【TSF 控制台】 ，在左侧导航栏中，单击【服务治理】。
2. 在服务治理列表页，单击微服务名称进入微服务详情页，单击【API列表】。
3. 单击【手动录入API】，输入 API 路径（其中 path 参数可以使用 {param} 来进行描述），并填写请求方法。
4. 同一个微服务下，通过请求路径和请求方法确认唯一一个API。不允许创建同请求方法和路径的API。

> **注意：**
>
> - 当注册中心获取到微服务注册的 API 与手动录入的 API 相同时，会认为是同一个 API。
> - 当注册中心判断某个 API 在当前任何一个部署组上都没有注册，会展示 API 状态为离线。
> - 仅当 API 状态为离线时，该 API 才可以被删除。

- 当注册中心获取到微服务注册的 API 与手动录入的 API 相同时，会认为是同一个 API。
- 当注册中心判断某个 API 在当前任何一个部署组上都没有注册，会展示 API 状态为离线。
- 仅当 API 状态为离线时，该 API 才可以被删除。

API 调试

1. 在 API列表页，单击 API 路径，进入 API 详情页。

2. 在 API 详情页，单击右上角的【调试】，进入 API 调试页面。

3. 填写调用 API 的默认参数，单击【发送请求】。

4. 右侧会展示调用 API 的返回结果。

接口列表

最近更新时间: 2025-02-18 16:02:00

操作场景

接口列表显示服务对外暴露的 API 列表。API 调试提供用户在线调试 API 的能力。 您可以在 TSF 控制台中，通过接口列表查看 API 的详细信息，并进行 API 在线调试。

前提条件

要使用 API 列表和调试功能，需要先将服务的 API 注册到注册中心，具体请参考开发手册 [API 注册] 。

操作步骤

查看API信息

1. 登录 【TSF 控制台】 ，在左侧导航栏中，选择【服务治理】。
2. 单击服务名称，进入服务详情页。
3. 在服务详情页，单击顶部的【接口列表】，会显示服务对外提供的 API。
4. 单击 API 名称进入详情页，可以查看到 API 的详细信息。 API 详情按照【应用名/版本号】划分显示了 API 的详细信息，包括：路径、方法、描述、入参、出参。其中 Models 表示参数中的复杂类型。

手动录入API

TSF 当前支持手动录入API。手动录入 API 的场景主要有：当微服务尚未注册到注册中心，但是希望将 API 进行提前录入，方便配置一些服务治理规则时，可以手动录入。

录入方式：

1. 登录 【TSF 控制台】 ，在左侧导航栏中，单击【服务治理】。
2. 在服务治理列表页，单击微服务名称进入微服务详情页，单击【API列表】。
3. 单击【手动录入API】，输入 API 路径（其中 path 参数可以使用 {param} 来进行描述），并填写请求方法。
4. 同一个微服务下，通过请求路径和请求方法确认唯一的一个 API。不允许创建同请求方法和路径的 API。

> **注意：**
>
> - 当注册中心获取到微服务注册的 API 与手动录入的 API 相同时，会认为是同一个 API。
> - 当注册中心判断某个 API 在当前任何一个部署组上都没有注册，会展示 API 状态为离线。
> - 仅当 API 状态为离线时，该 API 才可以被删除。

API 调试

1. 在 API 列表页，单击 API 路径，进入 API 详情页。
2. 在 API 详情页，单击右上角的【调试】，进入 API 调试页面。

3. 填写调用 API 的默认参数，单击【发送请求】。

4. 右侧会展示调用 API 的返回结果。

服务熔断

最近更新时间: 2025-02-18 16:02:00

TSF 服务治理支持可视化熔断规则管理，支持设置服务、实例、API 三种隔离级别的熔断规则。

熔断原理

定义

服务熔断定义：当下游的服务因为某种原因导致服务不可用或响应过慢时，上游服务为了保证自己整体服务的可用性，不再继续调用目标服务，直接返回。当下游服务恢复后，上游服务会恢复调用。

熔断器状态

服务熔断中涉及到关键概念**熔断器**，熔断器的状态转化如下：

1. 最开始处于 closed 状态，一旦检测到错误（或慢响应）达到一定阈值，便转为 open 状态，此时不再调用下游目标服务。
2. 等待一段时间后，会转化为 half open 状态，尝试放行一部分请求到下游服务。
3. 一旦检测到响应成功，回归到 closed 状态，也即恢复服务；否则回到 open 状态。

其中熔断器从 close 变为 open 状态要**同时**满足以下2个条件：

- 前提条件：在滑动时间窗口内至少有一定数量的请求（即**最少请求数**）
- 指标达到阈值：在滑动时间窗口内统计的错误请求率或慢请求率达到一定阈值

隔离级别及场景

TSF 支持服务、实例、API 三种隔离级别的熔断规则：

隔离级别	请求统计范围	熔断对象	适用场景
服务	下游目标服务的所有实例的所有 API	服务	当下游服务属于不主要的业务，可以熔断所有实例
实例	下游目标服务的单个实例的所有 API	达到熔断触发条件的实例	当下游服务属于比较重要的业务，只对异常的实例进行熔断，避免所有实例被熔断后导致服务雪崩
API	下游目标服务的所有实例的指定 API	达到熔断触发条件的单个 API	下游服务不同 API 的重要程度不同，需要根据不同 API 设置不同的熔断策略

使用方法

不同于服务限流、路由和鉴权规则在被调服务上设置，服务熔断规则是在**主调方服务**上设置。

开发指南

目前 TSF 支持 Spring Cloud 应用及 TSF Mesh 应用两种微服务框架的服务熔断。

Spring Cloud 熔断开发指南参考 [开发文档] ，注意 Spring Cloud 应用的服务熔断功能需要使用 1.19.0 版本及以上的 SDK，参考 【SDK 版本更新日志】 。

TSF Mesh 仅需完成控制台熔断规则配置并启用即可实现 TSF Mesh 服务熔断能力，无侵入操作简单。TSF Mesh 迁移开发指引，参考【TSF Mesh 指南】。

控制台基本操作

假设用户希望在 consumer-demo 上针对下游服务 provider-demo 设置一个熔断策略。

新建并启用熔断规则

> **注意：**
>
> 一个服务的不同熔断规则的下游目标服务不能重复。

1. 登录 [TSF 控制台]，在左侧导航栏单击【服务治理】，单击 consumer-demo 服务进入服务详情页。
2. 切换至【服务熔断】标签页，单击【新建熔断规则】，在创建熔断规则对话框中填写熔断规则：

- **下游服务**： 选择当前 provider-demo 所在命名空间和服务名。
- **隔离级别**：根据业务场景需求设置隔离级别
- 服务：选择服务隔离级别后，设置熔断策略各参数。
- 实例：选择实例隔离级别后，设置熔断策略各参数和最大熔断实例比率。
- API：选择 API 隔离级别后，可选择不同的 API 设置熔断策略，熔断策略中各参数适用于选中的每个 API。
- **滑动时间窗口**：用于统计熔断器关闭时的请求结果。
- **最少请求次数**：配置熔断器可以计算错误率之前的最小请求数。
- **触发条件**（满足以下**任一条件**触发熔断）：
- 失败请求率：在滑动时间窗口内统计的失败请求占所有请求比率（失败请求是指响应状态码为4XX和5XX，以及抛出异常的请求）。
- 慢请求率：在滑动时间窗口内统计的慢响应的请求占所有请求比率，其中「慢响应」的时长支持配置。
- **开启到半开间隔**：熔断器从 open 状态等待一段时间后变为 half-open 状态，尝试放行一部分请求到下游服务。
- **最大熔断实例比率**：该参数仅适用于**实例**隔离级别，用于控制最大熔断实例个数百分比，避免下游服务所有实例被熔断导致级联雪崩。例如当下游服务有20个实例且最大熔断实例比率为50%，熔断器最多熔断10个实例。

3. 单击【完成】，跳转至熔断规则列表。
4. 在熔断规则列表上，单击熔断规则的【启用】，启用该规则。

服务管理

最近更新时间: 2025-02-18 16:02:00

服务是微服务平台管理的基本单元，当微服务注册到注册中心时，服务会显示在服务列表中。您也可以提前手动创建服务，设置服务限流、路由等规则，当服务注册上来后规则会下发到匹配服务名的服务实例上。

创建服务

1. 登录 TSF 控制台。
2. 单击左侧导航栏的【服务治理】，选择集群和命名空间。
3. 单击服务列表页的【新建服务】。
4. 设置服务的基本信息后，单击【提交】。

- **服务名称**：要创建的服务的名称，不超过60个字符。服务名称由小写字母、数字和 - 组成，且由小写字母开头，小写字母或数字结尾。
- **服务描述**：服务的描述信息。

删除服务

> **注意：**
>
> 只有当服务的状态为【离线】时，即服务运行实例数为0时，才能删除服务。

1. 在 服务治理 页面，单击操作列的【删除】。
2. 在确认弹框中，单击【确认】即可删除服务。

服务监控

在 TSF 控制台服务治理页面可以看到线上服务的请求数、请求成功率、平均耗时等监控数据。数据统计周期都是24小时。

- **请求数**：对一个服务，统计其作为服务提供者，被所有消费他的服务消费者发起调用的24小时内总调用数。
- **请求成功率**：对于一个服务，统计24小时内其作为服务提供者，成功向消费他的所有服务消费者返回请求的总数比上服务请求总数。
- **平均耗时**：对于一个服务，统计24小时内其作为服务提供者，统计消费者从发起调用到调用返回到服务提供者的耗时平均值。

服务实例和手动下线

一个服务由多个服务实例构成，您可以在【服务详情页】>【服务实例列表】，查看服务下有多少实例。服务实例有「在线」和「离线」两种状态，离线的服务实例不会被其他服务发现，会在上次心跳时间24小时后自动清除。

当服务实例不可用且仍然注册到注册中心时，会导致请求发送到该问题实例上，此时可以开启【屏蔽实例】来手动下线该实例。服务被屏蔽后，该服务实例将不会被其他服务发现，流量不会分发到该实例上。

服务路由使用说明

最近更新时间: 2025-02-18 16:02:00

前提条件

要使用路由功能，用户需要在客户端配置依赖项，然后在 TSF 控制台设置路由规则。

使用服务路由

配置依赖项

Spring Cloud 应用请参考开发手册中的 【服务路由】 。对于 Mesh 应用，如果希望使用基于自定义标签的路由，需要在代码中设置标签，关于如何设置标签参考 【Mesh 开发使用指引】。

新建路由规则

1. 登录 【TSF 控制台】。
2. 在左侧菜单中，选择【服务治理】。
3. 在服务列表中，选择需要配置服务路由规则的服务，单击服务名称，进入服务详情页。
4. 选择服务路由选项，单击【新建路由规则】。
5. 新建路由规则，一个微服务下最多50条路由规则。
6. 路由规则名称
7. 填写规则

- 流量来源配置：设置系统标签和自定义标签表达式。
- 流量目的地：支持部署组和版本号两种目的地类型，确保权重加总为100。

> **注意：**
>
> - 当服务在线时，您可以通过服务当前关联的应用来过滤部署组，配置流量规则指向哪一个部署组。
> - 当服务尚未上线或已经离线时，系统无法判断该服务与哪一个部署组关联，您可以预先选择与将注册的服务相关联的应用，并选择对应的部署组或者版本号，提前配置路由规则。

3. 单击【提交】。

启用路由规则

1. 登录 【TSF 控制台】 ，单击左侧导航栏的【服务治理】，进入服务详情页面。

2. 选择服务路由标签。

3. 单击【生效状态】的切换按钮，当按钮为蓝色表明已经生效。

4. 配置生效后，可以在列表项的下面流量分配图中查看流量分配情况，用户可以选择时间段，查看部署组上流量分配情况。

- 24小时内的流量分配情况如下：

- 应用路由规则后10分钟之内的流量分配曲线如下：

5. 在流量分配图下方的流量分配表中，可以查看近五分钟内的平均每分钟请求数比例。

路由配置了流量的权重比例后，要使路由准确性达到预期，请求数至少要在1000以上；如果请求样本数不高的情况下偏差会比较大，样本数越高准确性就才越高。

容错保护

开启容错保护后，会实现兜底策略。例如服务设置了如下路由规则：

- 10%的流量分配到 v1 版本。
- 40%的流量分配到 v2 版本。
- 50%的流量请求分配到 v3 版本。

假设场景：v1 版本的实例全部不可用。

- 如果不开启容错保护，仍然会有10%的请求分发到 v1 版本的实例上，此时请求会失败。
- 如果开启容错保护，SDK 发现 v1 版本的实例不可用时，会采用 Round Robin 轮询算法将请求随机分发到所有可用实例上。

使用说明

- 填写路由规则需要在服务提供方进行配置，例如 A 服务调用 B 服务，需要在 B 服务上配置服务路由规则。
- 对于 Spring Cloud 服务，配置路由规则后，若配置的目标部署组无法运行，流量将按照原有默认的轮询方式分配到其他部署组上。
- 对于 Spring Cloud 服务，当服务提示未绑定应用时，需要在服务详情页单击编辑，绑定服务，才能开始配置路由规则。**服务绑定应用操作，一经绑定，不能修改。**
- Spring Cloud 服务调用其他服务的场景时，要使服务路由生效，需要确保 Spring Cloud 服务使用了 SDK 并添加开启路由注解，详情请参考开发手册中 【服务路由】。
- 对于 Mesh 应用，配置路由规则后，若配置的目标部署组无法运行，则路由规则配置失败，请求无法发送。

其他操作

编辑路由规则

> **注意：**
>
> 在生效状态的路由规则可以编辑，编辑之后立即生效。

1. 登录 【TSF 控制台】，单击左侧导航栏的【服务治理】，进入服务详情页面。
2. 选择服务路由标签。
3. 在服务路由页面已经提交的规则页面单击【编辑】。

在编辑页面，仅支持编辑规则的详情，不支持编辑规则类型。 4. 单击【提交】，完成路由编辑。

删除路由规则

> **注意：**
>
> 在生效状态的服务路由规则不能被删除，只能先停用，再删除。

1. 登录 【TSF 控制台】，单击左侧导航栏的【服务治理】，进入服务详情页面。
2. 选择服务路由标签。
3. 在服务路由页面已经提交的规则页面单击【删除】。

限制说明

等于、不等于、包含、不包含属于严格匹配，正则表达式属于模糊匹配。因此当系统标签是被调方 API PATH 时，目前仅支持使用**正则表达式**的逻辑关系来匹配带参数的 API 请求。

- 当标签的逻辑关系是正则表达式，值填写 `/echo/.*` 时，可以匹配带参数的请求 /echo/test123（其中 test123 是参数）。
- 当标签的逻辑关系是等于、不等于、包含、不包含关系，值是 `/echo/{param}` 时，不能匹配带参数的请求 /echo/test123（其中 test123 是参数）。

服务限流

最近更新时间: 2025-02-18 16:02:00

服务限流主要是保护服务节点或者数据节点，防止瞬时流量过大造成服务和数据崩溃，导致服务不可用。当资源成为瓶颈时，服务框架需要对请求做限流，启动流控保护机制。

限流原理

限流的原理是监控服务流量的 QPS 指标，当达到指定的阈值时进行流量控制，避免被瞬时高峰流量冲垮，从而确保服务的高可用。

TSF 限流方案采用了动态配额分配制，限流中控根据实例的历史流量记录，动态计算预测下一时刻该实例的流量，若所有实例的流量预测值都小于额定平均值（总配额/在线实例数），则以该平均值作为所有实例分配的配额；否则按预测流量的比例分配，且保证一个最小值。

TSF 目前支持在被调服务上设置限流规则，服务的限流对象（下文中称为"限流资源"）可以通过标签表达式灵活配置，常见的限流对象如当前服务、当前服务的特定 API 等，并且可以通过标签表达式区分不同的调用来源，针对不同的调用关系进行限流。一条限流规则主要包括以下几个元素：

- 限流粒度：通过标签表达式表示**被调方的限流资源**和**调用来源**。
- 限流阈值：单位时间和请求数，如果单位时间设置为1秒，则限流阈值为QPS。
- 生效状态：限流规则是否生效。

限流使用场景

场景1：根据调用方进行限流

调用关系中包括调用方和被调用方，一个被调服务可能同时被多个服务调用。在限流规则中，**限流粒度**字段可以用于根据调用来源进行流量控制，举例如下：

- **不区分调用者**：限流粒度选择**全局限流**时，来自任何调用者的请求都将进行限流统计。如果限流资源的调用总和超过了这条规则定义的阈值，则触发限流。
- **针对特定的调用者**：限流粒度选择**基于标签限流**，设置系统标签为**上游服务名**，逻辑关系为**等于**，值为特定的调用服务。
- **针对除特定调用者之外的调用方**：限流粒度选择**基于标签限流**，设置系统标签为**上游服务名**，逻辑关系为**不等于**，值为特定的调用服务。

区分调用方除了使用上游服务名等系统标签外，还可以使用自定义标签来区分带有不同业务信息的调用。例如针对特定用户 `foo` 的调用进行限流，可以在代码中设置**user**参数，然后在限流规则中配置业务标签为**user**，逻辑关系为**等于**，值为 `foo` 。

场景2：针对不同的资源进行限流

一个服务包含一个或多个 API，TSF 支持针对服务或者 API 进行限流。在限流规则中，**限流粒度**字段可以用于区分不同的限流资源，举例如下：

- **针对当前服务**：无须额外设置。
- **针对特定的API**： 限流粒度选择**基于标签限流**，设置系统标签为**当前服务的 API Path**，逻辑关系为**等于**，值为特定的 API Path。如果需要指定 HTTP Method，则需要再增加一条 `HTTP Method` 的系统标签来约束。

- **针对特定API之外的API**：限流粒度选择**基于标签限流**，设置系统标签为**当前服务的 API Path**，逻辑关系为**不等于**，值为特定的 API Path。如果需要指定 HTTP Method，则需要再增加一条 `HTTP Method` 的系统标签来约束。

使用限流功能

要使用限流功能，用户需要在客户端配置依赖项，然后在 TSF 控制台设置限流规则。

1. 配置依赖项

对于 Spring Cloud 应用，参考开发手册中的 [服务治理] 。对于 Mesh 应用，如果希望使用基于标签的限流，需要在代码中设置标签。

2. 新建限流规则

前提条件：服务列表上有"在线"状态的微服务。

1. 登录 [TSF 控制台] 。
2. 在左侧导航栏，单击【服务治理】。
3. 在服务列表页，单击服务名，进入服务详情页。
4. 选择**服务限流**标签页，单击【新建限流规则】。

5. 填写限流规则信息。

- **规则名**：填写规则名。
- **限流粒度**：
- 全局限流：不区分限流来源，统计所有请求。
- 基于标签限流：根据标签规则设置限流。
- **单位时间**：正整数，单位：秒。
- **请求数**：正整数，单位：次。
- **生效状态**：是否立即启用限流规则。
- **描述**：填写描述信息。

6. 单击【提交】完成新建。

3. 启动限流规则

在限流规则列表中，可以修改规则的【生效状态】。 多条限流规则都是生效状态时，只要服务接收到的请求满足**任意一条**限流规则，就会

触发限流逻辑。

4. 触发限流

假设服务提供了 `/echo` API，可以通过不断执行 `curl /echo` 来模拟限流场景。 **示例**：在 Demo 中 consumer-demo 服务提供了 `/echo-feign/{str}` API ，那么针对 consumer-demo 服务新建限流规则，限流粒度为全局限流，单位时间 2 秒，请求数 5 次。启用限流规则。 下载脚本 tsf_ratelimit.sh ，登录可以访问到 consumer-demo 的机器（consumer-demo 所在机器也可以），执行 `./tsf_ratelimit.sh &lt;IP&gt;:&lt;Port&gt;` ，其中 IP 是 consumer-demo 所在机器 IP， Port 为服务监听端口 18083。脚本的作用是**每 2 秒触发 10 次**调用。由于调用的频率大于限流规则，正常情况下，会收到 HTTP 429 (Too Many Requests) 的状态码。

5. 查看限流效果

如果请求数达到了限流阈值，任何到达的请求都会限流模块处理。如果该服务上的配额已经消耗完，会对请求返回 HTTP 429 （Too Many Requests ）；否则会正常放行。用户可以在限流规则列表下方的**请求数-时间**图中查看到被限制的请求数或者**被限制请求率-时间**图

中查看到被限制请求率（计算公式 `被限制请求率 = 被限制的请求数 / 请求数` ）随时间的变化。

限制说明

等于、不等于、包含、不包含属于严格匹配，正则表达式属于模糊匹配。因此当系统标签是被调方 API PATH 时，目前仅支持使用**正则表达式**的逻辑关系来匹配带参数的 API 请求。

- 当标签的逻辑关系是正则表达式，值填写 `/echo/.*` 时，可以匹配带参数的请求 /echo/test123 （其中test123是参数）。
- 当标签的逻辑关系是等于、不等于、包含、不包含关系，值是 `/echo/{param}` 时，不能匹配带参数的请求 /echo/test123 （其中test123 是参数）。

配置管理
配置管理概述

最近更新时间: 2025-02-18 16:02:00

配置类型

配置类型	功能说明	适用应用类型	关联对象	发布对象
【应用配置】	动态更新 Spring Cloud 或者 Dubbo 应用内的配置	Spring Cloud 或 Dubbo 应用	应用	应用关联的部署组
【全局配置】	动态更新 Spring Cloud 或者 Dubbo 应用内的配置	Spring Cloud 或 Dubbo 应用	命名空间	命名空间
【文件配置】	将文件配置发布到实例指定路径，发布成功后触发回调	任何应用类型	应用	应用关联的部署组

应用配置、全局配置、本地配置优先级

应用配置和全局配置属于 TSF 平台上的配置（下面称为 `远程配置` ），本地配置是应用程序在代码工程中创建的配置（如 `application.yml` 和 `bootstrap.yml` ）。应用配置和全局配置的根本区别在于**配置发布的范围**，应用配置发布的范围是部署组维度，全局配置发布的范围是命名空间维度。

> **注意：**
>
> 优先级：应用配置 > 全局配置 > 本地配置

当用户通过 TSF 控制台发布**远程配置**，微服务应用会按照配置的 key 来进行合并操作。例如，微服务应用本地 `application.yml` 配置文件的内容中包括：

```
# application.yml
username: test_user1
feature.status: false
feature.color: red
```

TSF 平台上 `远程配置` 的内容如下：

```
# TSF 应用配置或者全局配置
username: test_user2
feature.status: true
```

当 `远程配置` 的发布范围包含了上面的服务实例，微服务应用会将远程配置和本地配置按照 `key` 进行合并，最终生成的配置如下：

```
# 远程配置与本地配置合并结果
username: test_user2
feature.status: true
feature.color: red
```

多份应用配置发布到同一个部署组

TSF 支持多份应用配置发布到同一个部署组，多份配置会根据发布时间的先后顺序以 key 进行合并。例如，应用 A 有两个应用配置项： config-1 、 config-2 。

config-1 的配置内容：

```
# config-1
username: test_user1
feature.status: false
```

config-2 的配置内容：

```
# config-2
username: test_user2
feature.color: red
```

config-1 和 config-2 先后发布到部署组 group ，会按照 key 进行合并。

```
# config-1 与 config-2 合并结果
username: test_user2
feature.status: false
feature.color: red
```

配置模板

最近更新时间: 2025-02-18 16:02:00

操作场景

配置模板功能是为了方便用户保存常用的配置信息，也提供了 Ribbon、Hystrix、Zuul 等 Spring Cloud 组件的配置模板。用户可以基于已有的配置模板进行修改。

用户可以基于配置模板来创建 【应用配置】 或者 【全局配置】 。

前提条件

在使用配置模板功能之前，请确保已经按照 【分布式配置开发文档】 添加了代码注释。

操作步骤

新建配置模板

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【配置管理】>【配置模板】。

3. 单击【新建模板】。
4. 填写配置模板信息。

- 模板名：填写模板名。
- 类型：Ribbon、Hystrix、Zuul 或自定义。
- 配置内容：根据不同的类型，会自动生成对应的配置内容。用户可以进一步修改配置内容。
- 描述：填写描述信息。

5. 单击【提交】，完成新建。

使用配置模板

用户可以使用配置模板来创建应用配置或者全局配置，下面以全局配置举例。

1. 登录 【TSF 控制台】。
2. 在左侧导航栏，单击【配置管理】>【全局配置】。

3. 单击【导入配置模板】。

4. 选择要导入的配置模板。

5. 在新建配置页面中，补充全局配置的其他信息。

文件配置

最近更新时间: 2025-02-18 16:02:00

文件配置功能支持用户通过控制台将配置下发到服务器的指定目录。应用程序通过读取该目录下的配置文件实现特殊的业务逻辑。

文件配置支持如下功能：

- 创建文件配置项：一个文件配置项管理多个版本的配置。
- 生成新版本：基于历史版本生成新版本。
- 发布配置：支持发布配置到部署组。
- 发布情况：查看配置项的发布到哪些部署组。
- 回滚：回滚到上一个版本的配置。

使用场景

定时检查配置是否更新

- 应用程序中包含了读取指定目录配置文件的逻辑，例如定时去检查配置文件是否更新（通过文件 md5 是否变化等方式检查），如果更新了会执行特定逻辑。
- 在控制台上创建文件配置，下发到部署组。

动态替换 PHP 文件

通过控制台发布一个 PHP 文件到指定目录，来达到动态替换服务器上 PHP 文件的目的。

前提条件

能否使用文件配置功能，依赖于应用部署的环境是否满足以下条件：

- **对于使用虚拟机部署的应用**：只有2018年11月20号之后导入到集群的云主机上会具有满足应用配置功能的环境。
- **对于容器部署的应用**：该功能需要用户修改 Dockerfile。以下示例在 [制作镜像] 文档的基础上做修改：
- 需要将 `tsf-consul-template-docker.tar.gz`（下载地址）添加到 `/root/` 目录下：

```
ADD tsf-consul-template-docker.tar.gz /root/
```

- 启动脚本中，需要执行 `/root/tsf-consul-template-docker/script` 目录下的 `start.sh` 脚本：

```
CMD ["sh", "-ec", "sh /root/tsf-consul-template-docker/script/start.sh; exec java ${JAVA_OPTS} -jar ${jar} 2>&1"]
```

控制台基本操作

创建文件配置

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中，单击【配置管理】>【文件配置】。

3. 在文件配置页面，单击【新建配置】。
4. 填写文件配置信息：

- 配置名称
- 关联应用
- 文件保存编码
- 配置内容：支持上传文件或者控制台编辑
- 配置文件名称：下发到服务器的配置文件的文件名称
- 版本号

- 版本描述
- 配置下发路径：配置下发到服务器的路径

- 后置脚本（选填）：配置下发到服务器后执行的命令（**不需要** 包含 /bin/bash）

生成新版本

1. 在配置列表页，单击配置名称进入详情页。
2. 单击某个配置版本旁的【生成新版本】。
3. 填写变更的新版本的配置内容和版本号。
4. 单击【完成】，生成新版本。

发布配置

1. 在配置列表页，单击配置名称进入详情页。
2. 单击某个配置版本旁的【发布】。
3. 选择配置发布的目标部署组，填写发布描述。
4. 单击【提交】，完成发布。

查看文件所在部署组的配置发布历史

1. 在配置列表中，单击操作列的【查看发布信息】。
2. 展开所需查看的部署组，即可查看该部署组的配置发布历史。
3. 单击其中一条发布历史，可查看配置发布前后区别。

配置回滚

回滚配置会将部署组的配置回滚到**上一次**发布的版本。

1. 进入【发布情况】 界面， 查看文件所在部署组的配置发布历史 。
2. 在相应部署组的操作中，单击【回滚】。
3. 可查看回滚前后配置变化，单击【提交】。

加密配置

最近更新时间: 2025-02-18 16:02:00

配置加密功能提供了对配置值加密的存储全套解决方案。

通过增强源生 SDK 能力，同时兼容本地文件配置和分布式配置的配置值加密。

准备工作

1. 确保使用最新的 TSF SDK。
2. 按照 [分布式配置开发文档] 添加了代码注释。
3. 下载 SDK 加密工具。
4. 准备需要加密的相关信息（此处为举例，用户使用时请调整）

- 密码明文（plaintext）：TX_PwDemO_1hblsqT
- 密钥（encpty password）：encryptPassword

SDK 加密工具

1. 找到加密工具包（spring-cloud-tsf-encrypt-1.1.1-RELEASE.jar）。
2. 执行以下命令对配置明文密码进行加密（**需升级到 Java8 161或以上版本，或使用 补丁 解决问题**）：

```
D:\repo\com\tencent\tsf\spring-cloud-tsf-encrypt\1.1.1-RELEASE>java -jar spring-cloud-tsf-encrypt-1.1.1-RELEASE.jar encrypt TX_PwDemO_1hblsqT encryptPassword
```

输出结果：

```
[encrypt] result:
3M7wGw2XtFc5Y+rxOgNBLrm2spUtgodjIxa+7F3XcAo=
```

用例：

```
D:\repo\com\tencent\tsf\spring-cloud-tsf-encrypt\1.1.1-RELEASE>java -jar spring-cloud-tsf-encrypt-1.1.1-RELEASE.jar
At least 3 arguments required. Usage: [operation] [content] [password]
[operation]: Choose one from [encrypt | decrypt].
[content]: Plaintext when encrypt or ciphertext when decrypt.
[password]: Encrypt or decrypt password.
```

3. 执行以下命令对密文密码进行解密：

```
D:\repo\com\tencent\tsf\spring-cloud-tsf-encrypt\1.1.1-RELEASE>java -jar spring-cloud-tsf-encrypt-1.1.1-RELEASE.jar decrypt 3M7wGw2XtFc5Y+rxOgNBLrm2spUtgodjIxa+7F3XcAo= encryptPassword
```

输出结果：

```
[decrypt] result:
TX_PwDemO_1hblsqT
```

用例：

```
D:\repo\com\tencent\tsf\spring-cloud-tsf-encrypt\1.1.1-RELEASE>java -jar spring-cloud-tsf-encrypt-1.1.1-RELEASE.jar
At least 3 arguments required. Usage: [operation] [content] [password]
[operation]: Choose one from [encrypt | decrypt].
[content]: Plaintext when encrypt or ciphertext when decrypt.
[password]: Encrypt or decrypt password.
```

配置项填写方式

> **注意：**
>
> 本地配置和线上配置同时支持（需要符合 spring-config 源生规范）。

本地 YAML

配置在 application.yml或application.yml

```
tsf:
inventory:
password:
encrypt1: ENC(3M7wGw2XtFc5Y+rxOgNBLrm2spUtgodjIxa+7F3XcAo=)
```

配置中心 YAML

配置在全局配置/应用配置，并发布

```
tsf:
inventory:
password:
encrypt2: ENC(3M7wGw2XtFc5Y+rxOgNBLrm2spUtgodjIxa+7F3XcAo=)
```

本地 Properties

配置在application.properties或application.properties

```
tsf.inventory.password.encrypt3=ENC(3M7wGw2XtFc5Y+rxOgNBLrm2spUtgodjIxa+7F3XcAo=)
```

业务应用使用

环境变量（推荐）

在系统环境变量中配置密钥（password）：此时密钥泄露的风险最小。

```
tsf_config_encrypt_password=encryptPassword
```

JVM 参数（不推荐）

也可以在JVM参数中配置密钥（password）：

```
-Dtsf_config_encrypt_password=encryptPassword
```

启动参数（不推荐）

也可以在应用启动参数中配置密钥（password）：

```
--tsf_config_encrypt_password=encryptPassword
```

Java 测试代码

Java代 码按照常规配置使用。

配置类：

```
@ConfigurationProperties("tsf.inventory.password")
@Component
@RefreshScope
public class PasswordConfiguration {

private String encrypt1;
private String encrypt2;
private String encrypt3;

@Value("${tsf.inventory.password.encrypt1}")
private String encrypt4;
@Value("${tsf.inventory.password.encrypt2}")
private String encrypt5;
@Value("${tsf.inventory.password.encrypt3}")
private String encrypt6;

// getters and setters
}
```

测试类：

```


@RestController
public class TestController {

@Autowired
private PasswordConfiguration pwConfig;
/**
* 显示明文密码
*
* @return 明文密码
*/
@RequestMapping("/inventory/password")
public String showPassword() {
String content = "TX_PwDemO_1hblsqT";
StringBuffer sb = new StringBuffer("Test Config Encrypt/Decrypt:\n");
// 内存读取
sb.append(String.format("[%s]\t内存读取*.yml文件配置: %s\n",
content.equals(
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt1")),
```

```
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt1")));
sb.append(String.format("[%s]\t内存读取consul配置: %s\n",
content.equals(
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt2")),
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt2")));
sb.append(String.format("[%s]\t内存读取*.properties文件配置: %s\n",
content.equals(
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt3")),
SpringCloudTsfApplication.ctx.getEnvironment().getProperty("tsf.inventory.password.encrypt3")));
// Bean读取
sb.append(String.format("[%s]\tBean读取*.yml文件配置: %s\n", content.equals(pwConfig.getEncrypt1()),
pwConfig.getEncrypt1()));
sb.append(String.format("[%s]\tBean读取consul配置: %s\n", content.equals(pwConfig.getEncrypt2()),
pwConfig.getEncrypt2()));
sb.append(String.format("[%s]\tBean读取*.properties文件配置: %s\n", content.equals(pwConfig.getEncrypt3()),
pwConfig.getEncrypt3()));
// @Value读取
sb.append(String.format("[%s]\t@Value读取*.yml文件配置: %s\n", content.equals(pwConfig.getEncrypt4()),
pwConfig.getEncrypt4()));
sb.append(String.format("[%s]\t@Value读取consul配置: %s\n", content.equals(pwConfig.getEncrypt5()),
pwConfig.getEncrypt5()));
sb.append(String.format("[%s]\t@Value读取*.properties文件配置: %s\n", content.equals(pwConfig.getEncrypt5()),
pwConfig.getEncrypt5()));
return sb.toString();
}
}
```

输出结果如下：

```
Test Config Encrypt/Decrypt:
[true] 内存读取*.yml文件配置: TX_PwDemO_1hblsqT
[true] 内存读取consul配置: TX_PwDemO_1hblsqT
[true] 内存读取*.properties文件配置: TX_PwDemO_1hblsqT
[true] Bean读取*.yml文件配置: TX_PwDemO_1hblsqT
[true] Bean读取consul配置: TX_PwDemO_1hblsqT
[true] Bean读取*.properties文件配置: TX_PwDemO_1hblsqT
[true] @Value读取*.yml文件配置: TX_PwDemO_1hblsqT
[true] @Value读取consul配置: TX_PwDemO_1hblsqT
[true] @Value读取*.properties文件配置: TX_PwDemO_1hblsqT**
```

全局配置

最近更新时间: 2025-02-18 16:02:00

操作场景

全局配置功能用于动态更新应用代码中的配置。全局配置可以保证配置内容在某个集群或者命名空间中全局生效。 全局配置包括管理配置和发布配置两部分。管理配置包括创建配置、生成新版本配置和删除配置。配置可以发布到命名空间下的所有应用。

前提条件

在使用全局配置功能之前，请确保已经按照 【分布式配置开发文档】添加了代码注释。

操作步骤

创建配置

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【配置管理】>【全局配置】。
3. 在配置列表页，单击【新建配置】，进入配置界面。
4. 填写配置内容。配置可以按照 YAML 方式进行编辑。YAML 格式规范参考 [YAML 格式介绍] 。

> **注意：**
>
> 单个全局配置版本的大小不能超过65535个字节，如果实际使用的配置超过了该上限值，可以分成多个全局配置项发布到同一个命名空间，多个配置会合并成一份配置。

5. 单击【完成】，完成创建。

生成新版本配置

1. 单击配置项名称，进入配置详情页。

2. 单击配置列表右侧的【生成新版本】。
3. 填写变更的配置内容和版本号。

> **注意：**
>
> 新版本配置的版本号不能与原版本相同。

4. 单击【完成】，即可生成新版本。

删除配置

1. 单击配置项名称，进入配置详情页。

2. 删除每个配置版本，删除最后一个配置版本后，配置项将被删除。

> **注意：**
>
> 对于已经发布的配置，需要在【发布情况】页面中先删除配置，然后再删除配置版本，避免配置被误删除。

3. 在弹框中，单击【确认】。

发布配置

1. 在配置列表中，单击配置项名称，进入配置版本页面。
2. 单击版本号后面的【发布】，在弹框中选择命名空间，填写发布描述。
3. 单击【提交】。

配置合并逻辑说明

按照配置**下发时间**排序执行合并（merge）。不同名的配置项中如果存在相同 key 会进行合并。合并规则：按照配置下发时间排序，离当前时间近的优先级较高。举例如下：

1. 创建配置项 config-abc，配置内容是 custom-key: value-1，发布时间 15:00:00
2. 创建配置项 config-bcd，配置内容是 custom-key: value-2， 发布时间15:00:01

最终在实例上生效的配置：`custom-key: value-2`

查看命名空间对应配置发布历史

1. 在配置列表中，单击操作的【查看发布信息】。
2. 展开所需查看的命名空间，即可查看该命名空间的配置发布历史。
3. 单击其中一条发布历史，可查看配置发布前后区别。

配置回滚

1. 进入【发布情况】 界面，查看 命名空间对应配置发布历史。
2. 在相应命名空间的操作中，单击【回滚】。
3. 可查看回滚前后配置变化，单击【提交】。

应用配置

最近更新时间: 2025-02-18 16:02:00

操作场景

应用配置功能有两个入口，一个入口是在单个应用的应用详情页内，另一个入口是在**配置管理**模块的**应用配置**。应用配置功能**仅针对 Spring Cloud 应用和 Dubbo 应用生效**，应用配置支持如下功能：

- 创建配置项：一个配置项管理多个版本的配置。
- 生成新版本：基于历史版本生成新版本。
- 发布配置：支持发布配置到部署组。
- 发布情况：查看配置项的发布到哪些部署组。
- 回滚：回滚到上一个版本的配置。

前提条件

在使用控制台的应用配置功能前，请确保已经按照 【分布式配置】 配置了相关依赖项。

操作步骤

创建配置

1. 登录 【TSF 控制台】。
2. 在左侧导航栏，单击【配置管理】>【应用配置】。
3. 在应用配置页面顶部，选择目标应用。

4. 在配置列表标签页，单击【新建】。
5. 填写配置内容。

- 配置名称：填写配置名。
- 配置内容：按照 YAML 格式。
- 版本号：填写初始版本号。
- 版本描述：填写初始版本的描述。

6. 单击【完成】。

> **注意：**
>
> 单个应用配置版本的大小不能超过65535个字节，如果应用的配置超过了该上限值，可以分成多个应用配置项发布到同一个部署组，多个配置会合并成一份配置。

生成新版本配置

1. 在配置列表页，单击目标配置名称，进入详情页。

2. 在配置版本标签页，单击某个配置版本旁的【生成新版本】。
3. 填写变更的新版本的配置内容和版本号。

> **注意：**
>
> 新版本配置的版本号不能与原版本相同。

4. 单击【完成】。

发布配置

用户可以发布配置项的某个版本到部署组上。

1. 在配置列表页，单击目标配置名称，进入详情页。
2. 在配置版本标签页，单击某个配置版本旁的【发布】。
3. 选择配置发布的目标部署组，填写发布描述。
4. 单击【下一步】，单击【完成】。

配置合并逻辑说明

按照配置**下发时间**来排序执行合并（merge）。不同名的配置项中如果存在相同 key 会进行合并。合并规则：按照配置下发时间排序，离当前时间近的优先级较高。举例如下：

1. 创建配置项 config-abc，配置内容是 custom-key: value-1，发布时间 15:00:00
2. 创建配置项 config-bcd，配置内容是 custom-key: value-2， 发布时间15:00:01

最终在实例上生效的配置： `custom-key: value-2`

查看部署组的配置发布历史

用户可以单击操作列的【查看发布信息】查看该配置相关部署组的配置发布记录。

1. 在左侧导航栏，单击【配置管理】>【应用配置】>【查看发布信息】，进入发布情况页面。
2. 展开部署组，查看该部署组的配置发布记录。

3. 单击每条发布记录， 可查看配置发布前后区别。

回滚配置

回滚配置会将部署组的配置回滚到**上一次**发布的版本。

1. 进入【发布情况】 界面，查看部署组的发布历史。
2. 在相应部署组的操作中，单击【回滚】。

3. 可查看回滚前后配置变化，单击【提交】。

查看生效配置

最近更新时间: 2025-02-18 16:02:00

操作场景

用户可以通过 TSF 控制台查看发布到部署组上生效的配置，生效的配置是由发布到该部署组上的应用配置和全局配置合并之后的结果。

前提条件

已经创建一个或者多个应用配置或者全局配置，并发布到目标部署组。具体操作步骤参考 【应用配置】 和 【全局配置】。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏，单击【部署组】。
3. 单击某个部署组右侧操作【更多】>【查看配置】。

> **注意：**
>
> 由于生效配置是应用配置和全局配置合并之后的结果，因此不会保留原始配置的注释和结构。

弹性伸缩

最近更新时间: 2025-02-18 16:02:00

概述

弹性伸缩含义：根据预先设定的弹性伸缩规则，动态增加或者减少部署组的实例数。

弹性伸缩规则：由规则名、扩容活动、缩容活动、冷却时间等参数构成的规则，用来描述弹性扩缩容的触发条件、实例数量变化和限制。

弹性伸缩指标：

- CPU 利用率：在指定时间范围内，部署组内所有实例 CPU 利用率的平均值。
- 内存利用率：在指定时间范围内，部署组内所有实例内存利用率的平均值。
- 请求 QPS：在指定时间范围内，部署组内所有实例请求 QPS 的平均值。
- 响应时间：在指定时间范围内，部署组内所有实例响应时间的平均值。

冷却时间：设置冷却时间，可以确保在上一扩（缩）容活动生效前弹性伸缩不会启动或终止其他实例。弹性伸缩会等待冷却时间完成，然后再继续扩（缩）容活动。建议设置冷却时间大于持续时间。

以下视频将为您介绍 TSF 的弹性伸缩功能：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2038-24397?source=gw.doc.media&withPoster=1&notip=1

新建规则

1. 在 TSF 控制台左侧导航栏，单击【弹性伸缩】。
2. 在弹性伸缩页面左上方，单击【新建规则】。
3. 在新建弹性伸缩规则中，填写弹性伸缩规则内容。

- **规则名**
- **扩容活动** a. 触发条件：由指标、阈值、持续时间构成。多条触发条件为逻辑或（OR）的关系，满足任一条件。 b. 增加实例数：每次部署组的指标达到了触发条件后，增加的实例数量。 c. 最大实例数：部署组的实例数上限。
- **缩容活动** a. 触发条件：由指标、阈值、持续时间构成。多条触发条件为逻辑与 AND 的关系，必须同时满足。 b. 减少实例数：每次部署组的指标达到了触发条件后，减少的实例数量。 c. 最小实例数：部署组的实例数下限。
- **冷却时间**：建议设置冷却时间大于持续时间，如持续时间设置1分钟，冷却时间设置5分钟。

关联部署组

创建弹性伸缩规则后，将规则关联到部署组上。

1. 在弹性伸缩列表的操作栏，单击【关联部署组】。
2. 在关联部署组页面，选择应用，然后选择部署组。
3. 在关联部署组页面左下方，选择是否立刻开启规则。如果选择开启，则规则会在部署组上立刻生效，否则将不生效。用户可以后续在规则详情页的【关联部署组】 tab 页中修改启用状态。

解除规则和部署组的关联

1. 单击规则名称进入详情页，单击【关联部署组】标签页。
2. 在关联部署组列表右侧操作栏，单击【删除】，解除规则和部署组的关联。

删除规则

前提条件：已解除规则和部署组的关联。

1. 在弹性伸缩列表的操作栏，单击【删除】。
2. 在弹出的对话框中，单击【确认】，即可删除规则。

容器部署应用
上传镜像

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在容器部署场景下通过 TSF 控制台上传镜像。您可以选择自己制作镜像并推送到镜像仓库，也可以直接上传程序包，控制台将自动为您制作镜像并上传。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏 选择【应用管理】，单击目标应用的**ID/应用名。**
3. 在应用详情页，选择**程序包管理**标签页，单击【上传程序包/镜像】。

您可以选择自己制作镜像并推送到镜像仓库，也可以直接上传程序包，控制台将自动制作镜像并上传。

- 镜像：根据使用指引制作镜像并上传

详细操作参考【镜像仓库】。

- JAR包部署：直接上传程序包，无需制作镜像。

- JDK版本：KNOA JDK8或者OPEN JDK8
- 上传程序包：单击【选择文件】，选择编译为 fatjar 格式的程序包，程序包格式说明参考【程序包格式说明】。
- 程序包版本：填写版本号
- WAR包部署：直接上传程序包，无需制作镜像。

- JDK版本：KNOA JDK8或者OPEN JDK8
- 上传程序包：单击【选择文件】，选择编译为 war 格式的程序包，程序包格式说明参考【程序包格式说明】。
- 程序包版本：填写版本号

4. 单击【上传程序包并制作镜像】，右上角将出现任务进行的状态。
5. 单击【任务进行中】将看到任务进行的详情，若任务失败，可查看任务失败的原因。

上传镜像后后，您需要创建部署组并部署应用 。

应用管理

最近更新时间: 2025-02-18 16:02:00

操作场景

应用管理包括创建应用和删除应用，您可以按照以下步骤进行操作。

操作步骤

创建应用

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【[应用管理] 】。
3. 在应用列表左上方，单击【新建应用】。

4. 设置应用信息后，单击【提交】完成创建。

- **部署方式**：选择**容器部署**。
- **业务类型**：选择业务应用或者微服务网关应用。
- **开发语言**：选择您的开发语言。
- **开发框架**：选择您的开发框架。
- **应用类型**：
- 普通应用：适用于 Spring Cloud 或者 Dubbo 应用。
- Mesh 应用：适用于 Service Mesh 方式接入。
- 微服务网关应用：适用于微服务网关（Zuul、Spring Cloud Gateway）。
- 原生应用：适用于 Spring Cloud 原生应用
- **标签**：用于分类管理资源，可不选。
- **数据集**：选择"无"。用户可以通过数据集管理配置不同的子账号和协作者使用不同资源的权限 。
- **备注**：选填，可留空。

5. 单击【提交】完成应用创建。

创建应用后，需要添加实例并部署应用。详

删除应用

> **注意：**
>
> - 删除应用会同时删除应用关联的镜像。
> - 当应用下有部署组时，无法执行删除操作，需要先删除所有部署组后才能删除应用。

1. 在应用列表的操作列，单击【删除】。
2. 在确认弹框中，单击【确认】完成删除。

部署组管理（容器）

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台对容器应用部署组进行操作，包括：创建部署组、删除部署组、部署应用、启动应用、应用扩缩容等。

操作步骤

创建部署组

1. 登录 [TSF 控制台]。
2. 单击左侧导航栏中的 【部署组】。
3. 在页面顶部选择集群。
4. 单击部署组列表上方的【新建部署组】。
5. 设置部署组相关信息，单击【提交】。

- **部署组名称**：部署组的名称，不超过60个字符。
- **命名空间**：选择命名空间。
- **关联应用**：选择部署组关联的应用。关联应用字段决定了后续镜像来源和应用配置来源。
- **日志配置项**：选择日志配置项，用于采集应用的业务日志数据。
- **日志投递**：指定日志的转储方式，将规则指定路径中的日志内容投递到指定的接收端。如果配置为"无"，将不投递业务日志。更多关于日志投递的功能说明请参考 [日志投递]。

6. 单击【下一步】，按照 部署应用 进行操作。

部署应用

前提条件：已经将镜像推送到应用的镜像仓库中，参考 [制作镜像] 和 [镜像仓库]。

部署流程：

1. 在部署组列表页，单击目标部署组操作栏的【部署应用】。
2. 设置部署信息。

- **选择镜像**：选择要部署的镜像。
- **JVM 启动参数**（选填，仅适用普通应用）：设置 Java 应用的启动参数。参数会通过 JAVA_OPTS 环境变量带到容器运行环境中，参考 [制作镜像] 中的使用方式。

在java启动环境变量中应用容器参数应采用 $(var_name)的形式

- **环境变量**：设置应用容器中的变量。

- 自定义：您可以自定义环境变量
- Field：容器路径，包含metadata.name, metadata.namespace, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP

- ResourceField：容器资源，通常支持资源限制和请求，例如limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory 和 requests.ephemeral-storage。

3.（可选）设置高级参数。

参数	说明
更新方式	- 快速更新：直接关闭所有实例，启动相同数量的新实例。 - 滚动更新（推荐）：对实例进行逐个更新，这种方式可以让您不中断业务实现对服务的更新。
更新间隔	滚动更新的更新时间间隔。
更新策略	- 滚动更新方式可以设置更新策略。 - 启动新的Pod, 停止旧的Pod（默认）：需确认集群有足够的CPU和内存用于启动新的Pod，否则可能导致集群崩溃。 - 停止旧的Pod, 启动新的Pod ：适用于需要同时部署多个容器部署组，且集群内剩余资源不够时。 - 自定义：需要设置允许超出所需规模的最大Pod数量和允许最大不可用的Pod数量。
策略配置	可以设置批量启动或停止的 Pod 数量，可以设置百分比或正整数。详情请参考 【Kubernetes Deployment 滚动更新策略】和 【容器服务实例优雅下线】最佳实践进行配置。
健康检查	- 关于健康检查的作用和配置详情参考 【健康检查】。 - 存活检查：用于判断何时重启实例。 - 就绪检查：用于判断 Pod 何时变为 Ready 状态，会影响滚动更新。【默认检查】无须用户设置检查规则，会根据服务实例是否注册到注册中心来决定 Pod 变为 ready 状态。
Pod 调度策略	- 默认调度：优先保留yaml模版中设置的调度策略，如果未设置则按照集群资源调度，可能调度到某一可用区。 - 尽可能多可用区调度：尽可能调度到集群内不同可用区的主机上，可用性高。

4. 设置资源配置信息。

参数	说明
应用容器	运行用户指定的应用镜像，根据应用负载指定 CPU 和内存的资源限制。应用容器的资源配置参考[容器部署组资源限制]。 • request 用于预分配资源，当集群中的节点没有 request 所要求的资源数量时，会导致无法创建容器。 • limit 用于设置容器使用资源的最大上限，避免异常情况下节点资源消耗过多。
agent 容器	负责日志、JVM监控、调用链数据的采集，如果不部署 agent 容器将影响这些功能的使用。agent 容器的内存 limit 和日志量有关，通常使用默认值 400MiB 即可，如果出现 OOM 可适当增加 limit 值。
istio_proxy 容器	负责 Mesh 服务注册、流量转发等任务，通常使用默认的资源限制即可。
实例数量	实例数和实例资源限制的乘积不能超过集群剩余的可用资源。

5. 设置访问配置信息。

参数	说明

参数	说明
Service	负责 Mesh 服务注册、流量转发等任务，通常使用默认的资源限制即可。默认开启，关闭Service后，该Service下的负载均衡将一并销毁，销毁后不可恢复，请谨慎操作。
网络访问方式	参考 [**网络访问方式**] 了解不同访问方式的区别。
端口映射	容器端口与服务端口的映射关系。

通过直接编辑 YAML 更新部署组

容器部署组对应的 Kubernetes 里面的 `Deployment` 和 `Service` 对象，如果您希望直接修改 YAML 实现更能灵活地配置，可以按照下述步骤操作：

1. 单击部署组 ID 进入详情页，单击【基本信息】标签页。
2. 在基本信息卡片中找到 **YAML** 字段，单击【查看与编辑】，显示 `Deployment` 和 `Service` 的 YAML。
3. 单击【编辑】按钮，进入 YAML 编辑页面，编辑 YAML。
4. 单击【提交】按钮，容器部署组会以更新后的 YAML 进行重新部署。

通过 Webshell 登录容器

1. 单击"部署组 ID "进入服务实例列表。

2. 单击操作栏的【登录】通过 Webshell 登录容器。 登录后界面如下：

应用扩缩

1. 单击部署组右侧的【更多】>【应用扩缩】。
2. 选择扩缩的实例数量后，单击【提交】。

删除部署组

1. 单击部署组列表页右侧的【删除】。
2. 弹出提示页面，单击【确定】删除部署组。

网络访问方式

容器部署组的访问方式定义访问后端 Pod 的访问方式，并提供固定的虚拟访问 IP。您可以在 Service 中通过设置来访问后端的 Pod，不同访问方式的服务可提供不同网络能力。

TSF 目前提供以下四种服务访问方式：

- 提供公网访问

使用 Service 的 Loadbalance 模式，公网 IP 可直接访问到后端的 Pod，适用于 Web 前台类的服务。

创建完成后的服务在集群外可通过负载均衡域名或 IP + 服务端口访问服务，集群内可通过部署组名称 + 服务端口访问服务。

- 仅在集群内访问

Headless Service：不创建用于集群内访问的ClusterIP，访问Service名称时返回后端Pods IP地址，用于适配自有的服务发现机制。

使用 Service 的 ClusterIP 模式，自动分配 Service 网段中的 IP，用于集群内访问。数据库类等服务如 MySQL 可以选择集群内访问，以保证服务网络隔离。

创建完成后的服务，要在同一个命名空间的容器内，通过部署组名称 + 服务端口访问服务。

- VPC 内网访问

使用 Service 的 Loadbalance 模式，指定 annotations:service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: subnet-xxxxxxxx，即可通过内网 IP 直接访问到后端的 Pod。

创建完成后的服务在集群外可通过负载均衡域名或 IP + 服务端口访问服务，集群内可通过部署组名 + 服务端口访问服务。

- 主机端口(NodePort)访问

提供一个主机端口映射到容器的访问方式，支持 TCP、UDP、Ingress。可用于业务定制上层 LB 转发到 Node。

创建完成后的服务可以通过云服务器 IP + 主机端口或部署组名称 + 服务端口访问服务。

> **注意：**
>
> 一个账号在单地域创建的公网负载均衡 LB 实例有数量限制，参考负载均衡 【使用约束】 。如需要扩大负载均衡实例限额，请 【提交工单】处理。

TSF 和 TKE 容器部署的区别

TSF 简化了容器应用部署的参数配置，能做到自动处理，对比 TKE（容器服务）区别如下：

挂载点和数据卷

TSF 将应用配置日志项路径设置为挂载点，并自动创建数据卷并映射（本地硬盘）。

TSF 应用关联日志配置项路径：

TKE 界面如下：

容器部署组实例资源限制

最近更新时间: 2025-02-18 16:02:00

实例资源限制说明

新建容器部署组时可以设置实例数量和每个实例资源限制，实例的资源限制包括两个指标：CPU 和内存大小。

Request：容器使用的最小资源需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配资源量 >= 容器资源请求数时才允许将容器调度到该节点。但 Request 参数不限制容器的最大可使用资源值。 **Limit**： 容器能使用的资源最大值。

当 实例数量 * Request值 > 集群剩余的资源 时，会提示 "资源不足，请导入节点" 的提示语。此时用户需要去集群页面导入云服务器以扩充资源。例如实例数量为2，实例资源Request是 CPU=0.5核，内存=1GB ，而集群剩余资源为 CPU=0.8核，内存=2GB 时， 由于 CPU 核数资源不够，会提示资源不足的错误。

Java 应用的最大堆内存和容器内存大小关系

Java 应用通常需要设置 JVM 启动参数，包括最大堆内存（-Xmx）的设置。建议 JVM 最大堆内存和容器实例内存资源大小的关系符合以下条件，避免容器 OOM（Out of memory）：

```
容器内存 Request >= 1.25 * JVM 最大堆内存 ；

容器内存 Limit >= 2 * JVM 最大堆内存；
```

例如，启动参数中设置 -Xmx 为 1024m，应用容器实例内存的 Request 至少为 1280MiB，Limit 至少为 2560MiB。

关于如何使用启动参数，请参考 [制作镜像 > Spring Cloud 应用构建材料] 中 Dockerfile 启动命令中的 ${JAVA_OPTS} 。

查看容器部署组事件

最近更新时间: 2025-02-18 16:02:00

操作场景

当用户通过容器部署组发布应用遇到异常时，可以通过查看事件详情进行异常定位。

该任务指导您在TSF控制台查看事件详情。

操作步骤

查看容器部署组事件

登录 【TSF 控制台】，您可以通过两种方式查看容器部署组最近1小时的事件详情：

- 方式一：在部署组页面，单击目标部署组操作栏的【更多】>【查看事件】。
- 方式二：在部署组页面，单击目标部署组的"部署组名称/ID"，进入部署组详情页面，选择【事件】页签。

字段信息	说明
首次出现时间	该事件首次出现时间
最后出现时间	该事件最后出现时间
级别	normal、warning、error
资源类型	pod、container、Deployment、ReplicaSet等
资源名称	资源名称信息，可在容器服务进行查看
内容	描述关键信息
详细描述	事件详情
出现次数	出现次数统计

查看容器Pod事件

登录 【TSF 控制台】，在部署组页面，单击目标部署组的"部署组名称/ID"，进入服务实例列表页面

单击服务实例的"实例ID/名称"，进入实例概览页面，选择事件页签，查看最近1小时的事件详情。

镜像仓库

最近更新时间: 2025-02-18 16:02:00

操作场景

镜像仓库用于存放用户上传的镜像，该任务指导您熟悉镜像仓库的常用操作。

前提条件

- 安装 docker。
- 使用 sudo 允许系统管理员让普通用户执行 docker 命令。

操作步骤

初始化镜像仓库

首次使用镜像仓库时，需要进行初始化操作，设置登录仓库的密码。

TSF 会针对每个容器应用创建一个名为 tsf_<账号ID>/<应用名> 的镜像仓库。

获取镜像仓库使用指引

创建应用后，单击应用ID，在应用的详情页中找到【镜像】标签页。单击【上传程序包/镜像】，查看使用指引。

将镜像推送到镜像仓库

1. 制作容器镜像，参考【制作容器镜像】。
2. 复制【使用指引】中登录 docker registry 的命令并执行。

```
sudo docker login --username=<账号 ID> ccr.ccs.tencentyun.com
```

> **注意：**
>
> 用户需要输入两次密码，首次为 sudo 密码，第二次为镜像仓库登录密码。

命令行工具显示 Login Succeeded 即表示登录成功。

3. 登录成功后，复制【使用指引】中给镜像打tag的命令并执行。

```
sudo docker tag [ImageId] ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

其中 [ImageId] 和 [tag] 是在制作镜像时获取。

> **注意：**
>
> 若此时需要输入密码，请输入sudo 密码。

4. 复制【使用指引】中推送镜像到仓库的命令并执行，其中 [tag] 和步骤4相同。

```
sudo docker push ccr.ccs.tencentyun.com/tsf_<账号ID>/<应用名>:[tag]
```

拉取镜像

1. 复制【使用指引】中登录 docker registry 的命令并执行。

```
sudo docker login --username=<账号 ID> ccr.ccs.tencentyun.com
```

> **注意：**
>
> 用户需要输入两次密码，首次为 sudo 密码，第二次为镜像仓库登录密码。

命令行工具显示 `Login Succeeded` 即表示登录成功。

2. 复制【使用指引】中拉取镜像的命令并执行。

```
sudo docker pull [NAME]:[tag]
```

其中 `[NAME]` 表示镜像名称，`[tag]` 表示镜像标签，例如：

```
sudo docker pull ccr.ccs.tencentyun.com/tsf_123456/tsf_demo:v1.0
```

重置密码

用户可以通过 TSF 控制台重置镜像仓库密码。

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中，单击【镜像仓库】。

3. 单击【重置密码】。

4. 根据页面提示填写密码，单击【提交】完成重置。

程序包格式说明

最近更新时间: 2025-02-18 16:02:00

> **注意：**
>
> - 只有 Linux 虚拟机部署的应用可以进行程序包管理。
> - 当部署 Spring Cloud 类型的微服务时，当前只有 jar 包部署的微服务支持完整的服务注册发现、完整的服务监控、调用链和服务治理能力。

目前使用云服务器部署的应用支持的程序包格式包括 jar、war、tar.gz 和 zip。

- jar： FatJar 格式的程序包 。
- war： war 格式的程序包，在部署 war 包时，TSF 会自动安装 Tomcat 环境。示例 demo 下载 ，示例 demo 部署成功后，使用 `http://imgcache.finance.cloud.tencent.com:80<IP>:8080/sample` 访问，其中 IP 可以是实例的外网 IP。
- tar.gz 、zip ： 压缩包中**必须**包含三个文件，**确保文件名正确**：
- start.sh：启动脚本。
- stop.sh：停止脚本。
- cmdline：用于检查应用进程是否存在，**没有** .sh 后缀。

文件类型	启动方式
war	云服务器上的 agent 会使用`java -jar`命令启动程序。
jar	云服务器上的 agent 会使用`java -jar`命令启动程序。
tar.gz	云服务器上的 agent 会解压压缩包，使用解压目录下的`start.sh`脚本启动应用程序。
zip	云服务器上的 agent 会解压压缩包，使用解压目录下的`start.sh`脚本启动应用程序。

start.sh / stop.sh / cmdline 说明

以一个 Python 应用的压缩包为示例，解压后的文件目录如下：

- promotionService.py
- start.sh
- stop.sh
- cmdline

start.sh 启动脚本内容如下：

```
#! /bin/bash

already_run=`ps -ef|grep "python promotion"|grep -v grep|wc -l`
if [ ${already_run} -ne 0 ];then
echo "promotionService already Running!!!! Stop it first"
exit -1
fi

nohup python promotionService.py 8093 &
```

stop.sh 停止脚本内容如下：

```
#!/bin/bash

pid=`ps -ef|grep "python promotion"|grep -v grep|awk '{print $2}'`
kill -SIGTERM $pid
echo "process ${pid} killed"
```

cmdline 检测进程脚本，agent 通过 ps -ef | grep 'cmdline 内容' 来检测进程是否存在，示例如下：

```
python promotion
```

cmdline 更多说明

如果启动应用是 Java 应用，启动脚本中通过 java -jar xxx.jar 来启动应用。在 cmdline 文件中使用完整的 Java 启动命令。例如启动脚本中包含如下启动命令：

```
java -Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m -jar consumer-demo-0.0.1-SNAPSHOT.jar
```

那么在 cmdline 中内容为：

```
java -Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m -jar consumer-demo-0.0.1-SNAPSHOT.jar
```

当应用启动后，agent 会在服务器上执行 ps -ef | grep 'cmdline 内容' 来检查进程是否存在。

> **注意：**
>
> 如果没有 cmdline 文件或者 cmdline 文件内容不正确，在控制台上部署组的状态会显示为 "已停止"，即使此时服务器上的应用已经运行起来（但是 TSF agent 无法获取应用进程状态）。

MacOS 系统压缩软件说明

对于 MacOS 系统的用户，使用系统自带压缩软件时，会在压缩包里面生成 __MACOSX 的临时目录，从而导致 agent 无法找到启停脚本。用户可以 下载 Keka 压缩软件，选择 zip 压缩格式，勾选【排除 Mac 资源文件】选项。将文件拖拽到 keka 界面上进行压缩，这种方式生成的压缩包没有 __MACOSX 的临时目录。

应用部署概述

最近更新时间: 2025-02-18 16:02:00

名词解释

应用：是用户的业务应用，通过应用可以对用户的程序包以及应用配置进行管理。

部署组：是执行应用批量部署的逻辑概念。一个部署组内包括多个应用实例，每个应用实例上运行相同的应用程序。

应用场景

TSF 应用部署分为三种场景：

- **虚拟机应用部署**：通过程序包将应用部署在云服务器上。
- **容器应用部署**：通过镜像将应用部署在 Docker 容器中，Docker 应用部署时，将在云服务器上创建多个 Docker 容器实例
- **Serverless 应用部署**：通过程序包将应用部署在 Serverless 方式中。（内测中）

三种部署场景的对比如下：

部署场景	虚拟机部署	容器部署	Serverless 部署
应用托管方式	一台云服务器部署一个应用	使用 Docker 部署应用，一台云服务器可以部署多个应用	无需关心托管方式，开箱即用
使用场景	传统部署场景	对容器运行环境需要定制和希望提升资源利用率的场景	免运维，弹性扩容的高效部署场景
集群类型	虚拟机集群	容器集群	-
部署方式	JAR 包、zip 压缩包、tar.gz 压缩包	镜像	JAR 包、zip 包、.tar.gz
应用举例	Spring Boot、Dubbo	Spring Boot、Dubbo、MySQL、WordPress	Express、Koa、Egg

部署流程

1. 创建集群
2. 将导入云主机集群
3. 创建应用
4. 上传程序包或镜像
5. 创建部署组
6. 部署应用

健康检查

最近更新时间: 2025-02-18 16:02:00

TSF健康检查分为存活检查及就绪检查：

- 存活检查主要作用是确定进程存活状态，判断是否需要进行实例重启。例如存活检查可以捕捉到死锁（应用程序进程还存在，但是无法响应），重启进程或者容器可以让应用程序恢复可用。
- 就绪检查主要作用是确定服务实例能否支持对外服务，将健康检查结果与注册中心状态联动实现滚动更新及无损发布。当一个实例没有就绪，实例会在注册中心中被屏蔽（其他服务不会发送请求到微服务实例上）。

健康检查支持的产品能力：

- 存活检查
- 就绪检查
- 无损发布
- 检查顺序：http检查，tcp端口，执行命令检查
- 支持虚拟机部署组和容器部署组

健康检查与注册中心联动流程：

1. 就绪检查，检查实例状态是否ready。
2. 如果就绪检查ready则更新实例注册状态为passing，反之则检查状态为cirtical。
3. 监听注册中心服务提供方实例状态变更。
4. 存在状态变更更新缓存及本地文件。
5. 发起服务调用。

配置存活检查

1. 登录 【TSF 控制】，在左侧菜单栏中，单击【部署组】。
2. 找到目标部署组，单击操作列的【部署应用】。
3. 在部署应用页面，开启存活检查。
4. 检查方式：

- HTTP 请求检查：任何大于200小于400的返回码都会认定是成功的返回码。其他返回码都会被认为是失败的返回码。HTTP 检查需要设置端口和请求路径。
- TCP 端口检查：如果可以建立连接被认为是成功的。该检查方式需要设置检查端口。
- 执行命令检查：如果命令执行成功并且返回值为 0，认为是成功；其他返回值认为是失败。该检查方式需要填写执行命令。

5. 启动延时、超时时间、检测周期、健康阈值、不健康阈值使用默认值即可。

配置就绪检查

场景：服务A调用服务B。服务B使用 `/health` 接口是否返回 `200` 状态码判断是否健康，当就绪检查失败时，服务B在注册中心中被屏蔽，服务A通过注册中心监听到服务B状态变化后更新本地路由表；当就绪检查成功时，服务B在注册中心恢复健康状态。

1. 登录 【TSF 控制台】，在左侧菜单栏中，单击【部署组】。

2. 找到目标部署组，单击操作列的【部署应用】。
3. 在部署应用页面，开启就绪检查。

- 如果是容器部署组，可以选择**默认检查**或者**自定义**。【默认检查】无须用户设置检查规则，会根据服务实例是否注册到注册中心来决定 Pod 变为 ready 状态。
- 如果是虚拟机部署组，则只能自定义设置。

4. 检查方式：

- HTTP 请求检查：任何大于200小于400的返回码都会认定是成功的返回码。其他返回码都会被认为是失败的返回码。HTTP 检查需要设置端口和请求路径。
- TCP 端口检查：如果可以建立连接被认为是成功的。该检查方式需要设置检查端口。
- 执行命令检查：如果命令执行成功并且返回值为 0，认为是成功；其他返回值认为是失败。该检查方式需要填写执行命令。

5. 启动延时、超时时间、检测周期、健康阈值、不健康阈值使用默认值即可。

就绪检查和滚动更新

部署组支持立即更新和滚动更新两种更新方式：立即更新会先停止**所有**应用实例，然后使用新的程序包或者镜像版本部署；滚动更新会根据更新策略，分批更新部署组内的实例。

如果部署组使用滚动更新并且开启就绪检查，当就绪检查失败时滚动更新会被阻塞。

编辑健康检查

1. 登录【TSF 控制台】，在左侧菜单栏中，单击【部署组】。
2. 找到目标部署组，单击"ID/部署组名"。
3. 选择【基本信息】，单击【健康检查】模块右上角的【编辑】。
4. 编辑健康检查信息，单击【提交】。

TSF 健康检查和 Kubernetes 健康检查的关系

对于容器部署组，存活检查和就绪检查和 Kubernetes 的 Liveness 和 Readiness Probe 对应。除此之外，TSF 就绪检查还会和注册中心进行关联，当检查失败时会将实例从注册中心屏蔽，避免流量打到异常实例上。

> **注意：**
>
> 关于 Kubernetes 健康检查详情参考【Kubernetes 配置存活和就绪检查】。

运维中心
日志服务
日志服务概述

最近更新时间: 2025-02-18 16:02:00

日志服务简介

日志服务为用户提供一站式日志服务，从日志采集、日志存储到日志内容搜索，帮助用户轻松定位业务问题。用户通过指定部署组的日志配置项来指定日志采集规则，TSF Agent 根据日志配置项采集指定路径下的文件日志，并上传日志到日志存储模块。用户可以通过 TSF 控制台查看部署组实时日志，并根据关键词来检索日志。

下图是用户在 TSF 平台上使用日志服务的流程图：

日志服务原理

日志采集 通过 Agent 来采集日志，无须手动安装。

日志索引与查询

- 实时索引：采集的日志数据建立索引。
- 查询灵活：支持全文检索、关键词检索（短语或者分词）。

日志存储额度说明

- 存储时长：TSF 下默认存储日志时长：启动日志（Stdout）3天，业务日志（日志配置项）30天。
- 存储量：TSF 为用户提供了日志存储的免费额度。免费存储额度与客户使用的版本、节点数相关。

版本	基础版	专业版	铂金版
每个节点免费额度	3GB/节点/月	4GB/节点/月	20GB/节点/月

- 当日志达到免费额度上限时，平台将通过短信、邮件、微信等方式向您发送通知。超过上限时，我们将按天删除您系统中保存时间最久的日志直到日志总存储量低于免费额度。系统将为您保存最少2天日志。

> **注意：**
>
> 如果您希望定期删除日志，请 [提交工单] ，并说明日志存储时间要求。

- 如果您希望获得业务日志的存储时间，您可以使用日志服务（CLS）。虚拟机部署业务接入 CLS 请参考 【CLS 入门指南】。容器部署业务接入 CLS 请参考 【TKE 日志采集】。

> **注意：**
>
> 日志迁移到 CLS 中，您可以使用 CLS 中更丰富的日志分析能力，但 TSF 中提供的日志与调用链联动能力无法继续使用。

日志配置

最近更新时间: 2025-02-18 16:02:00

操作场景

日志配置项用于指定采集日志的规则，包括日志的采集路径和日志解析格式（功能待发布）。用户可以在 TSF 控制台上创建日志配置项，然后将配置项发布到部署组上。同一个部署组可以关联多个日志配置项。

操作步骤

创建日志配置项

1. 登录 【TSF 控制台】。
2. 在左侧导航栏 ，单击【日志服务】>【日志配置】。

3. 在日志配置页，单击【新建配置】，在"创建日志配置"页面中设置您的日志配置项信息。

- **名称**：填写日志配置名称，不超过 60 个字符。
- **日志类型**：根据应用程序的日志类型选择一种日志类型，目前支持 Spring boot、 Nginx Access 、自定义logback、自定义log4j、 自定义log4j2 、单行/多行文本、无解析规则七种类型。
- **采集路径**：设置您的日志采集路径，可配置一个或者多个日志采集路径。目前支持**绝对路径具体到日志文件**，如/data/log/2017.log，文件名称允许使用数字、字母、横杠 - 、下划线 _ 、通配符 * 和小数点 . 。日志文件必须具有文件后缀，文件名和目录支持通配符，如 /data/log/*.log 或 /var/log/*/*.log 。
- **日志格式**：当日志类型为"自定义 logback"、"自定义 log4j"、"自定义log4j2" 及 "单行/多行文本"时

> **注意：**
>
> - 若要实现调用链与业务日志联动，部署组关联的日志配置项必须遵守日志格式规范。
> - Spring Boot 格式默认可以支持日志调用链联动。
> - 自定义 Logback、自定义 Log4j、自定义 Log4j2 和单行/多行文本格式日志需要日志 pattern 中添加 %trace 才可以支持日志和调用链联动。
> - Nginx Access 和无解析规则格式日志无法支持日志和调用链联动。

设置日志格式后，您可以通过"格式解析"功能，检验当前的设置是否正确：

1. 单击【格式解析】，打开日志格式解析弹框。

2. 复制部分日志内容粘贴在"日志内容"区域，单击【解析】，查看"解析结果"区域的输出是否符合预期。 **备注**：选填，日志配置项的描述信息。
3. 单击【提交】完成创建。

默认日志配置项

TSF 提供默认日志配置项，格式是 Spring Boot。日志路径包括三部分，相对于应用启动路径：

- ./*.log

- ./log/*.log
- ./logs/*.log

用户可以在微服务应用的配置文件中配置 logging.file 为上述任意一个路径，然后在创建部署组时，选择默认日志配置项 default-log-config 。

> **注意：**
>
> 默认日志配置项仅适用于虚拟机部署的应用。

部署组关联日志配置项

部署组关联日志配置项有两个入口：

- 用户可以在创建应用时将日志配置项关联到应用。
- 在日志配置项列表，右侧操作栏中单击【发布规则】。

创建部署组时选择日志配置项

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【部署组】。
3. 单击【新建部署组】，在弹出框中选择关联的日志配置项。
4. 单击【提交】。

在日志配置项列表发布配置

5. 登录 【TSF 控制台】。
6. 在左侧导航栏 ，单击【日志服务】>【日志配置】。
7. 在日志配置页，单击操作列的【发布规则】。
8. 选择要绑定的部署组。
9. 单击【提交】完成发布。

删除日志配置项

当日志配置项没有被其他部署组关联时，才可以删除日志配置项。

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【日志服务】>【日志配置】。
3. 在日志配置页，单击操作列的【更多】>【删除】。
4. 在删除确认框中单击【确认】，即可删除日志配置项。

日志告警

最近更新时间: 2025-02-18 16:02:00

操作场景

日志告警功能允许您通过配置业务日志中的关键词，设置关键词出现频率的告警。您需要在 TSF 控制台上配置需要告警的关键词和监控对象，并在云监控界面上配置告警通知人。

前提条件

要触发日志告警，您需要先完成以下操作：

1. 已采集日志数据
   - 部署组已经关联了日志配置项
   - 程序打印的日志的路径和解析规则和日志配置项保持一致
2. 日志中有告警统计规则中的关键词

操作步骤

新建告警统计

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【日志服务】>【日志告警】。
3. 在日志告警页面，单击左上角【新建告警统计】，在弹框中填写相关信息。
   - 关键词：填写需要设置告警的关键词，关键词中可以包含空格、引号、逗号、句号、冒号等常见特殊符号。填入关键词输入框的内容都将被视为一个整体进行监控告警。整体输入内容不能超过60字符。当前暂不支持正则表达式。
   - 部署组：选择需要监控部署组。此时可以选择某集群和某命名空间下的多个部署组。统计时多个部署组分别统计。

4. 单击【提交】，完成告警统计。

配置告警策略

1. 配置好告警统计后，在【日志告警】列表页面，单击【前往配置告警】，跳转至 【云监控控制台】。
2. 在【云监控】>【告警配置】>【告警策略】页面，单击左上角的【新增】。
3. 在新增策略界面，填写新建策略内容。
   - 策略名称：20 字以内。
   - 策略类型：选择 TSF 日志告警。
   - 所属项目：通常为默认项目。

   - 告警对象：选择在已经在 TSF 控制台上配置的告警统计对象（关键词和需要对这个关键词进行统计的部署组）。此处支持多选。
   - 触发条件：针对日志告警，需要填写上面选择的关键词在一定长度的统计周期下出现次数超过一定范围触发告警。

- 告警渠道：设置接收组、有效时段和接收渠道。 接收组：单击输入框后面的【新增接收组】，跳转至 [访问管理控制台] 配置。 接收渠道：可以通过邮件、短信、微信等方式配置告警通知渠道。

- 接口回调：填写回调地址方便云监控将告警信息推送到该地址。

4. 单击【完成】。当被监控的对象发生告警时，告警接收组的用户即可在配置的邮件、短信或微信上收到监控信息。

查看告警策略详情

1. 登录【云监控控制台】。
2. 在【告警配置】>【告警策略】中，查看当前配置的告警策略列表。

3. 在策略列表中，单击目标策略的策略名称，即可查看策略详情。 在这个页面上，用户可以修改当前的告警策略。**当某一条策略被修改时，该策略导致的告警条目在告警列表中告警状态将展示为“数据不足”。**
4. 在管理告警策略页面底部，您可以填写告警回调信息，填写公网可访问到的 URL 作为回调接口地址（域名或 IP[:端口][path]），云监控将及时把告警信息推送到该地址。

查看告警列表

1. 登录 【云监控控制台】。

2. 在左侧导航栏中，选择【告警历史】，进入告警列表页。 在此页面上，可以看到近期的告警信息。其中告警状态表明的是曾经发生的告警在当前的状态是否依然能够触发告警。当显示为 “已恢复” 时，表明此时告警情况已经被修复。

编辑告警对象

告警对象，即需要配置告警的关键词和部署组，需要在 [TSF 控制台] 上进行编辑。 在【日志服务】>【日志告警】>【日志告警列表】

中，可以对告警对象进行编辑和删除。

日志检索

最近更新时间: 2025-02-18 16:02:00

操作场景

当部署组关联了日志配置项后，TSF 会对采集的日志数据实时建立索引。用户可以在 TSF 控制台中，通过日志检索功能使用关键词检索出关键日志信息。

操作步骤

1. 登录 TSF 控制台，单击左侧导航栏【日志服务】>【日志检索】。
2. 选择搜索的时间范围。
3. 选择日志数据源

- 日志配置项：日志配置项决定了日志格式、日志采集路径等规则；
- 部署组：从关联了日志配置项的部署组中进行选择；
- 实例：当选择单个部署组时，可以将数据源范围精细化到部署组内某个实例。

4. 选择排序方式，默认是【按关键词匹配度排序】。
5. 输入日志关键词，单击【查询】。 关键词搜索包含三种查询方式：基本查询、Lucene 语法查询 及 正则表达式。用户可根据需求自行进行选择。在选定其一查询方式后，输入关键字。

> **注意：**
>
> 中文日志场景下，请使用正则表达式方式进行检索。

6. 单击【查看日志上下文】，查看该条日志的上下文信息。

> **注意：**
>
> 时间范围选择组件的结束时间为进入日志检索页面的时间。

关键词查询

基本查询

输入待查询的关键词或短语，按下回车键，输入下一个待查询的关键词或短语，按下回车，直至全部输入完成。单击【查询】，日志列表将展示出所有符合条件的日志。 本查询方式默认将 `任意空白字符,.[]{}()?:=` 认定为分词符，含有上述符号的日志被按照分隔符所在位置拆分。例如 `org.apache.kafka.clients.NetworkClient` 将被分词为 `org apache kafka clients NetworkClient` 。

- 支持单词的搜索 例如：输入 `available` ，输出日志为所有含有单词 `available` 的日志。
- 支持短语的搜索 例如：输入 `Broker may not be available` ，输出日志为所有含有短语 `Broker may not be available` 的日志。

Lucene 语法查询

支持检索类型	规则	示例
全文检索-单词检索	输入单个词语进行搜索，精准输出所有包含该词语的日志	startup
全文检索-短语检索	输入双引号""囊括短语进行搜索，精准输出所有包含该短语的日志	startup
布尔表达式搜索	支持 A AND B 搜索，输出所有同时包含 A 和 B 的日志；支持 A OR B 搜索，输出所有含有 A 或所有含有 B 或同时含有 A 和 B 的日志；支持 A NOT B 搜索，输出所有含有 A 同时不含有 B 的日志；支持 +A，输出所有包含 Term 的日志；支持 -A，输出的日志不包含 A	-
通配符搜索	支持？搜索，？可匹配单个字符； 支持 * 搜索，* 可匹配0到多个字符	-
模糊搜索	支持 A~Beta（数值）启动模糊搜索，输出包含所有距离 A 相似度为 Beta 的单词的日志。该模糊检索查询出拼写错误的单词	-
优先级搜索	支持 A^beta（数值）B 搜索，使 A 的搜索优先级相较于 B 提高 Beta	-
转义特殊字符	以下字符当作值搜索时需要用 转义： +- = ( ) { } []	(1+1)\=2 用来查询 (1+1)=2

正则表达式查询

支持使用正则表达式对日志进行搜索。 例如：搜索形如以 p 开头 r 结尾的关键词，我们可以直接在搜索框内输入 `/p.*r/` 进行匹配。

日志投递

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 的日志服务，提供日志投递功能。您可通过配置日志投递规则，并将其发布至目标部署组，从而实现部署组日志的转存。

操作步骤

新建日志投递规则

1. 登录 【TSF 控制台】。
2. 在左侧菜单栏中，单击运维中心下的【日志服务】>【日志配置】，切换到【日志投递】标签页。
3. 单击【新建投递规则】，在弹出的【新建投递规则】弹框中输入您的规则设置。

- 规则名称：必填，文本，长度限制60个字符。
- 日志采集路径：必填，至少需输入一个路径，长度限制256个字符。
- 可通过【添加】按钮添加多个路径输入框，以设置多个路径。
- 接收端：日志转存至的目标位置。
- 仅支持设置一个接收端。
- 目前仅支持 Kafka 类型的接收端，您需设置 IP、端口号和 Topic。
- 所填写的 Kafka 集群，可为自建 kafka 集群或 CKafka。
- 投递配置：支持从不同路径采集到的日志投递到同一个 kafka 实例的不同 Topic 中。
- 采集路径：需要具体到某一个日志文件。
- 投递 Topic：填写该目录的文件投递到具体哪一个 Topic 中。
- 高级设置：配置投递到 Kafka 的鉴权规则。当打开鉴权配置后，需要填写 Kafka 访问的用户名与密码。

> **注意：**
>
> - 所配置的 Kafka 必须和应用部署的 VPC 相同。
> - 请在 CKafka 上开启白名单，以保证日志采集端和 Kafka 的网络联通性。
> - 日志采集路径请勿以 /data/tsf_apm 开头。
> - 同一个部署组上只支持添加一条投递规则。如果配置多条规则，最后配置的规则将覆盖原有规则。

4. 单击【保存&下一步】，进入绑定部署组页面。
5. 绑定部署组，并单击【完成】。 规则创建完成后，您可将规则绑定至多个部署组。绑定后，此投递规则即会在所选的部署组中生效，部署组中规则指定路径下的日志，会被自动转存至目标接收端。 

查看日志投递规则详情

日志投递规则创建后，会以列表项的形式呈现在【日志投递】页面的规则列表中。

1. 单击某列表项中的【规则 ID】，进入该规则的详情页。 

2. 在规则详情页中，查看其基本信息和发布情况（规则已绑定的部署组列表）。 

发布日志投递规则

投递规则创建完成后，您可通过【发布规则】将投递规则绑定至更多部署组。

1. 选择目标规则，进入发布规则页面。您可以通过以下任一方式操作：

- 方式一：在投递规则列表页中找到目标规则，单击操作列的【发布规则】。
- 方式二：进入目标规则的【规则详情】>【发布情况】页中，点击左上方的【发布规则】。

2. 选择所需绑定的部署组，单击【提交】即可。

> **注意：**
>
> 已绑定当前规则的部署组不可被选中；已绑定其它规则的部署组，被绑定新规则时，部署组的投递规则被替换为新的规则。

修改部署组的投递规则

您可通过以下方式，修改部署组所绑定的投递规则。

- 方式一：您可通过将新的规则绑定至部署组，替换其原有投递规则。
- 方式二：在部署组的详情页的【基本信息】中，编辑其投递规则；选择新的规则并保存后，部署组的投递规则即更新为新选的规则。

编辑日志投递规则

规则创建成功后，您可随时修改规则的配置信息

1. 在规则列表页中找到所需修改的规则项，单击规则 ID 进入其详情页。 

2. 单击详情页【基本信息】卡片右上方的“编辑”图标，在打开的【编辑投递规则】弹框中，编辑规则配置项后，单击【保存】即可。

3. 规则修改完成后，自动在规则已绑定的部署组中生效。

删除日志投递规则

投递规则创建后，您也可以随时将其删除。

> **注意：**
>
> 规则删除后，自动把该规则从所有已绑定该规则的部署组中清除，并停止此规则的日志投递。

1. 在投递规则列表页中找到目标规则，单击操作列中的【更多】>【删除】。

2. 二次确认后即可将规则删除。 

快速入门

最近更新时间: 2025-02-18 16:02:00

操作场景

您可以通过本文介绍的四种方式，在控制台上查看应用程序实时打印的日志，快速了解 TSF 日志服务的使用方式。

操作步骤

查看标准输出（stdout）日志

TSF 默认提供 stdout 标准输出日志的查看，无须额外设置。假设用户已经完成了通过部署组部署应用的操作，以下为查看日志的步骤：

1. 登录 【TSF 控制台】，在左侧导航栏单击【部署组】。
2. 在部署组操作栏中，单击【查看日志】。

使用 Spring Boot 默认日志格式

1. 在 TSF 控制台【[日志配置] 】界面，创建日志配置项 `consumer-demo-log` ，选择日志格式 **Spring Boot**，采集路径为 `/var/root.log` 。
2. 应用程序打印日志到指定目录，在配置文件（如 application.yml）中设置打印文件日志的路径，和步骤1中日志配置项的日志采集路径保持一致。

```
logging.file=/var/root.log
```

3. 程序打包后，在控制台上新建部署组，选择日志配置项 `consumer-demo-log` 。
4. 部署应用。
5. 在部署组操作栏中，单击【查看日志】。

使用自定义 logback 日志格式

1. 在 TSF 控制台【日志配置】界面创建日志配置项 `log-config` ，选择日志格式为**自定义logback**，设置解析规则和日志路径如下图：
2. 应用程序打印日志到指定目录，在配置文件 `logback.xml` 中配置日志的 Pattern。

```
<configuration scan="true" scanPeriod="60 seconds" debug="false">
<property name="log.path" value="/var/root.log" />
<appender name="stdout" class="ch.qos.logback.core.ConsoleAppender">
<encoder>
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %contextName [%thread] %-5level %logger{36} - %msg%n</pattern>
</encoder>
</appender>
</configuration>
```

3. 程序打包后，在控制台上新建部署组，选择日志配置项 `log-config` 。
4. 部署应用。

5. 单击部署组操作栏中【查看日志】，选择日志配置项 log-config 查看日志信息。

应用程序不配置日志路径

如果用户不想配置日志的打印路径，但是仍然希望将日志采集后做检索，可以使用 TSF 提供的默认日志打印和采集功能。该功能要求工程满足以下条件：

- 该工程为 Spring Cloud 应用。
- 工程的配置文件中**不要设置** logging.file 参数（如果用户设置了该参数，默认日志功能不会生效）。
- 在 pom.xml 中依赖 TSF 的 logger 依赖，该依赖会打印日志到默认日志路径。

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-logger</artifactId>
</dependency>
```

当工程满足上面三个条件后，在 TSF 上的操作步骤如下：

1. 在控制台上新建部署组时，选择日志配置项为 default-log-config ，该日志配置项会通知 TSF agent 去采集默认日志路径下的日志。
2. 部署应用。
3. 在部署组操作栏中，单击【查看日志】，选择日志配置项 default-log-config 查看日志信息。

日志格式说明

最近更新时间: 2025-02-18 16:02:00

TSF 目前支持 Spring boot、Nginx Access 、自定义 Logback、自定义 Log4j、自定义 Log4j2、单行/多行文本、无解析规则七种类型。

> **注意：**
>
> 请尽量保持**以年-月-日做日志开头**（如：2019-09-21、11:09:48.395，因为分割日志时以这种格式作为分割）。否则，可能导致日志显示分行异常，甚至不能显示日志。

1. Spring Boot

如果应用程序使用默认的 Spring Boot 日志，则选择 Spring Boot 日志类型。不需设置“日志格式”。

2. Nginx Access

如果应用程序使用默认的 Nginx Access 日志，则选择 Nginx Access 日志类型。不需设置“日志格式”。

3. 自定义 Logback

如果应用程序使用 Logback 日志配置，设置日志类型为 Logback，然后设置日志格式（即日志解析规则，对应 Logback 中的 pattern）。可参考 Logback 中关于 Pattern 的介绍。参考模板如下：

Pom

```
<project xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0" xmlns:xsi="http://imgcach
e.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0 http://imgcache.finance.
cloud.tencent.com:80maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version>1.16.0-Edgware-RELEASE</version>
</parent>

<artifactId>provider-demo</artifactId>
<packaging>jar</packaging>
<name>provider-demo</name>

<dependencies>
<!-- TSF启动器 包含完整依赖 -->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
</dependency>
</dependencies>
</project>
```

Logback.xml

使用时，应用应该依赖有 LogbackTraceConverter 类的 SDK。

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<springProperty scope="context" name="springAppName" source="spring.application.name"/>
<conversionRule conversionWord="trace" converterClass="com.tencent.tsf.logger.LogbackTraceConverter" />
<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
<!-- encoder 默认配置为PatternLayoutEncoder -->
<encoder>
<!-- <pattern>%d %L %c %t %msg%n</pattern>-->
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %trace %p %c [%t] %msg%n</pattern>
</encoder>
</appender>
<appender name="TSF" class="ch.qos.logback.core.rolling.RollingFileAppender">
<File>${LOG_FILE:-Startup}
</File>
<rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
<FileNamePattern>${LOG_FILE:-Startup}.%d{yyyy-MM-dd}.%i.log
</FileNamePattern>
<MaxHistory>30</MaxHistory>
<maxFileSize>100MB</maxFileSize>
<totalSizeCap>1GB</totalSizeCap>
</rollingPolicy>
<encoder>
<!-- <pattern>%d %L %c %t %msg%n</pattern>-->
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %trace %p %c [%t] %msg%n</pattern>
<charset>UTF-8</charset>
</encoder>
</appender>
<root level="info">
<appender-ref ref="STDOUT" />
<appender-ref ref="TSF" />
</root>
</configuration>
```

4. 自定义 Log4j

如果应用程序使用 Log4j 日志配置，设置日志类型为 Log4j，然后设置日志格式（即日志解析规则，对应 Log4j 中的 pattern）。可参考 Log4j 中关于 Pattern 的介绍。参考模板如下：

Pom

```
#排除日志中默认的logback依赖，引入log4j的依赖
<project xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0" xmlns:xsi="http://imgcach
e.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0 http://imgcache.finance.
cloud.tencent.com:80maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version>1.16.0-Edgware-RELEASE</version>
</parent>

<artifactId>consumer-demo</artifactId>
```

```
<packaging>jar</packaging>
<name>consumer-demo</name>

<dependencies>
<!-- TSF启动器 包含完整依赖 -->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<exclusions>
<exclusion>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-logging</artifactId>
</exclusion>
</exclusions>
</dependency>
<!--log4j-->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-log4j</artifactId>
<version>1.3.8.RELEASE</version>
</dependency>
</dependencies>
</project>
```

Log4j.xml

使用时，应用应该依赖有 Log4JPatternLayout 类的 SDK。请注意日志输出路径。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://imgcache.finance.cloud.tencent.com:80jakarta.apache.org/log4j/">
<appender name="console" class="org.apache.log4j.ConsoleAppender">
<param name="Target" value="System.out"/>
<layout class="com.tencent.tsf.logger.Log4JPatternLayout">
<param name="ConversionPattern" value="%d %p %l %c %m%n"/>
<!-- <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss,SSS} %T %p %F %r %m%n"/>-->
</layout>
</appender>
<appender name="TSF" class="org.apache.log4j.RollingFileAppender">
<param name="File" value="/tsf-demo-logs/provider-demo/root.log" /><!-- 设置日志输出文件名 -->
<!-- 设置是否在重新启动服务时，在原有日志的基础添加新日志 -->
<param name="Append" value="true" />
<param name="MaxBackupIndex" value="10" />
<layout class="com.tencent.tsf.logger.Log4JPatternLayout">
<param name="ConversionPattern" value="%d %p %l %c %m%n"/>
<!-- <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss,SSS} %T %p %F %r %m%n" />-->
</layout>
</appender>
<root>
<priority value ="info" />
<appender-ref ref="console" />
<appender-ref ref="TSF"/>
</root>
</log4j:configuration>
```

5. 自定义 Log4j2

如果应用程序使用 Log4j2 日志配置，设置日志类型为 Log4j2，然后设置日志格式。参考模板如下：

Pom

```
<project xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0" xmlns:xsi="http://imgcach
e.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0 http://imgcache.finance.
cloud.tencent.com:80maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version>1.16.0-Edgware-RELEASE</version>
</parent>

<artifactId>provider-demo</artifactId>
<packaging>jar</packaging>
<name>provider-demo</name>

<dependencies>
<!-- TSF启动器 包含完整依赖 -->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<exclusions>
<exclusion>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-logging</artifactId>
</exclusion>
</exclusions>
</dependency>

<!--log4j2-->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-log4j2</artifactId>
<version>2.1.6.RELEASE</version>
</dependency>
<!--log4j2-->
<dependency>
<groupId>commons-logging</groupId>
<artifactId>commons-logging</artifactId>
<version>1.2</version>
</dependency>

</dependencies>
</project>
```

Log4j2.xml

日志文件名应为 Log4j2.xml。请注意日志输出路径。

```
<PatternLayout pattern="%d %nano %p %M %uuid %m%n" />
<!-- <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss,SSS} %trace %level %F [%l] %message%n" />-->
</Console>
<RollingFile name="TSF" fileName="${basePath}/root.log" filePattern="${basePath}/root.%d{yyyy-MM-dd}.%i.log">
<PatternLayout>
<PatternLayout pattern="%d %nano %p %M %uuid %m%n" />
<!-- <pattern>%d{yyyy-MM-dd HH:mm:ss,SSS} %trace %level %F [%l] %message%n</pattern>-->
</PatternLayout>
<Policies>
<TimeBasedTriggeringPolicy interval="1" modulate="true" />
</Policies>
</RollingFile>
</appenders>

<!-- 然后定义logger，只有定义了logger并引入的appender，appender才会生效 -->
<loggers>
<root level="info">
<appender-ref ref="console"/>
<appender-ref ref="TSF" />
</root>

</loggers>
```

6. 单行/多行文本

当日志为自定义格式时，可选择日志类型"单行/多行文本"，日志日期时间部分提供四种可选的格式，然后设置剩余部分内容的格式。日期格式支持以下四种：

- yyyy-MM-dd HH:mm:ss.SSS
- yyyy-MM-dd HH:mm:ss,SSS
- dd/MMM/yyyy:HH:mm:ss.SSS
- dd/MMM/yyyy:HH:mm:ss,SSS

7. 无解析规则

指一行日志内容为一条完整的日志。日志服务在采集的时候，将使用换行符 `\n` 来作为一条日志的结束符。为了统一结构化管理，每条日志都会存在一个默认的键值 `__CONTENT__` ，但日志数据本身不再进行日志结构化处理，也不会提取日志字段，日志属性的时间项由日志采集的时间决定。

查看日志

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 会根据部署组的日志配置项来采集业务日志。例如日志配置项的采集路径是 `/tsf-demo-logs/tsf-inventory/*.log` ，TSF 会采集该路径下的文件日志。

TSF 默认提供 stdout 标准输出日志的查看，无须额外设置。

操作步骤

1. 登录 TSF 控制台，单击左侧导航栏【部署组】，进入部署组列表页。
2. 在目标部署组列表页，您可以通过两种方式查看日志：

- 方式一：单击操作列的【更多】>【查看日志】，即可查看实时日志。
- 方式二：单击部署组 ID 进入详情页中，单击顶部的【日志】标签页查看实时日志。

3. 日志信息如下图所示，切换日志类型查看日志配置项的业务日志或者 stdout 标准输出日志，切换右上角按钮查看实时日志和历史日志。

- 日志配置项：查看日志配置项的采集路径下的业务日志。
- stdout（标准输出）日志：查看实例标准输出日志。

全链路灰度发布
全链路灰度发布概述

最近更新时间: 2025-02-18 16:02:00

灰度发布是软件上线过程中常见的上线方式，是指在发布过程中，将具有一定特征或者比例的流量分配到需要被验证的版本中，用来观察新的验证版本的线上运行状态。 相比全量上线，灰度发布是更加谨慎的发布形式。当线上调用链路较为复杂时，全链路灰度发布可以将线上的各个服务隔离出一个单独的运行环境。

全链路灰度发布的常见场景如下： 如上图所示，我们划分出一个灰度环境，针对线上版本2的所有服务，我们希望都能配置一定规则。当满足一定规则后，所有请求都会流入灰度环境中，经过微服务 A 版本2的请求，一定会同样落入微服务 D 的版本2中。

名词解释

- **泳道**：泳道是一组部署组的集合，是灰度发布规则的目的地。泳道中的部署组属于不同的应用。可以认为用户通过划分泳道划分出了灰度环境。
- **灰度规则**：用户在灰度规则上配置请求需要满足的条件。当请求满足一定条件后，可以通过灰度规则将流量路由到某一个泳道中。

服务泳道

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，完成新建泳道、删除泳道、编辑泳道和查看泳道监控信息的操作。

前提条件

使用全链路灰度发布之前，需要先配置泳道。一条泳道相当于一个灰度环境，环境中包含应用中需要进行灰度测试的部署组。在 【全链路发布概述】中的示例图中，我们希望为不同微服务的版本2创建一条泳道，将微服务 A、B、D 部署版本2的部署组都放在一个泳道中。

假设当前访问链路为 gateway -> consumer -> provider。其中 consumer、provider 都有两个版本。希望通过请求参数 test = 1时配置流量通过网关访问到 consumer、provider 的版本1中。

操作步骤

新建泳道

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在全链路灰度发布页面单击【泳道配置】>【新建泳道】。
3. 填写泳道名称和备注，单击【下一步】。

- 泳道名称：必填。最长60个字符，只能包括小写字母、数字及分隔符。
- 备注：选填。最长200字符。

4. 在选择部署组页面，勾选需要导入泳道的部署组，单击【完成】。

此时，我们需要创建一个泳道，将 gateway、consumer 的版本1部署组、provider 的版本1部署组添加到这个泳道中。设置 gateway 为泳道的入口。

过滤条件

- 用户可以通过选择命名空间类型、命名空间名称、应用名称和部署组名称进行筛选过滤部署组。默认提供了命名空间类型和命名空间名称作为过滤条件。
- 当用户希望切换命名空间时，单击当前已经选中的命名空间的名称，在弹出的下拉列表中切换。
- 当用户希望通过应用名称筛选部署组时，单击命名空间右侧的空白位置，便可弹出应用和部署组的下拉选项，如图所示。

泳道入口

- 全链路灰度发布规则配置后，会在泳道的入口部署组上对请求规则进行校验，以此来判断请求是否应该进入某一个泳道中。通常情况下，泳道入口是一个网关。
- 同一个泳道中支持多个入口，在请求经过每一个入口部署组时，都会判断请求是否应该进入泳道中。

- 配置方法：在选择部署组页面勾选需要被导入到泳道中部署组后，在穿梭框的右侧通过泳道入口开关配置。

泳道与命名空间

- 当前 TSF 支持两种 [命名空间]：全局命名空间和普通命名空间。
- 泳道中可以包含全局命名空间中的部署组和普通命名空间中的部署组。

> **注意：**
>
> 当前 Mesh 应用不支持部署在全局命名空间中，不支持泳道中同时出现全局命名空间中的部署组和 Mesh 应用的部署组。

删除泳道

> **注意：**
>
> 当泳道上已经绑定了全链路灰度发布的规则，则不能删除。

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在全链路灰度发布页面单击【泳道配置】，进入泳道配置列表页。
3. 选择目标泳道，单击操作列的【删除】。删除后，全链路灰度规则在该泳道上将不可用。

编辑泳道

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在全链路灰度发布页面单击【泳道配置】，进入泳道配置列表页。
3. 单击目标泳道 ID进入泳道详情页。
4. 在基本信息中，单击右上角的【编辑】，可以修改泳道名称和备注。 在部署组管理中，可以向泳道添加或移除部署组，也可以设置某个部署组为泳道入口。

查看泳道监控信息

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在全链路灰度发布页面单击【泳道配置】，进入泳道配置列表页。
3. 单击目标泳道 ID进入泳道详情页，选择【泳道监控】页签。
4. 设置好时间范围后，即可查看该泳道的请求量、耗时和错误率信息。

灰度发布

最近更新时间: 2025-02-18 16:02:00

操作场景

创建好泳道后，我们需要配置灰度发布规则。延续泳道配置中的场景：假设当前访问链路为 gateway -> consumer -> provider。其中 consumer、provider 都有两个版本。希望通过请求参数 test = 1 时配置流量通过网关访问到 consumer、provider 的版本1中。

当前，gateway 以及 consumer、provider 的版本1 已经放在了网关中，我们需要配置一条规则，当请求参数 test = 1时，流量到这个泳道中。

前提条件

已 [创建服务泳道]。

操作步骤

新建灰度规则

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在灰度发布页面，单击【新建灰度规则】。
3. 填写规则名称和备注，单击【下一步】。

- 泳道名称：必填。支持中英文字符，不超过60字符。
- 备注：选填。不超过200字符。

4. 填写请求参数规则和规则生效关系，单击【下一步】。

- **标签名**：是 TSF 中的自定义标签，自定义标签是用户自定义的业务参数转换的 TSF 中使用的标签。 参数名最多32字节，参数值最多128字符。此处，我们配置标签名为 test，设置标签值为1。 自定义标签的来源有两种：
- 方式一：通过微服务网关将请求的参数转换为标签：如请求的 path、query、header 中包含的业务参数，通过微服务网关的 tag 插件转换为标签。详细使用说明可以参考 【微服务网关 Tag 插件】。
- 方式二：以在代码中配置标签，详细使用说明可以参考 【参数传递】。
- **逻辑关系**：支持等于、不等于、包含、不包含、正则表达式五种逻辑关系。
- 选择等于和不等于时，标签值只能填写为单个值。
- 选择包含和不包含时，标签值可以填写多个，使用英文半角逗号分隔。
- **规则生效关系**：上述条件的生效逻辑关系，或（满足任一规则）、与（满足全部规则）。每条全链路灰度规则最多包含10条子规则。

5. 配置流量目的地。选择已经创建好的泳道，单击【完成】。 针对上文中配置的 consumer、provider 的例子，需要配置规则 test = 1，指向已经创建好的泳道。

切换灰度规则生效状态

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。

2. 在灰度发布页面，通过生效状态按钮调整规则是否生效。同时可以生效多条规则。

调整灰度规则优先级

全链路灰度发布支持同时生效多条规则，并支持为规则配置优先级。当同一条请求同时满足多条规则时，会优先匹配高优先级的规则。

1. 登录 【TSF 控制台】，在左侧导航栏单击【全链路灰度发布】。
2. 在灰度发布页面，单击【调整规则优先级】。

3. 拖动规则名左侧的拖拽图标，进行优先级配置，并单击【保存】。 实际生效的优先级是生效状态的规则从页面上方向下的顺序进行排列的。

全链路灰度结果展示

在 TSF 控制台的【依赖分析】>【调用链查询】中，用户可以查询对应的请求是否按照预期走到了泳道中。

1. 登录 【TSF 控制台】，在左侧导航栏单击【依赖分析】>【调用链查询】。
2. 在调用链查询页面，选择时间、调用的命名空间和微服务，或通过其他条件筛选需要查询的调用链。 如果需要对标签和标签值进行过

   滤，可以单击【展开高级查询条件】，输入标签和标签值，进行过滤。如图所示：

3. 单击非泳道入口的微服务的调用链详情或者蓝色的耗时条，可以查看调用链详情。

全链路灰度流量流向规则与说明

- 当请求流量没有命中任何灰度规则，流量将走到没有被添加到泳道的部署组中。
- 当某一个微服务下的部署组没有被加入任何泳道中，请求将在该微服务下所有部署组的所有实例中轮询。经过该微服务后，请求将继续按照规则配置流入对应泳道。 举例：A -> B -> C 微服务调用，A 和 C 微服务下有版本1和版本2，B只有一个版本。配置灰度规则test = 1 的流量进入版本2对应的泳道后，流量将从 A 的版本2进入，经过 B 微服务，流入到 C 微服务的版本2中。
- 一个部署组可以属于多个泳道。
- 在泳道中，服务路由规则不生效。
- 泳道是一个隔离环境。当泳道上所有规则关闭后，流量不会进入泳道中。如希望恢复建立泳道前的流量分配方式，请将泳道删除。

服务拓扑依赖

最近更新时间: 2025-02-18 16:02:00

操作场景

服务依赖拓扑包含了查询服务之间相互依赖调用的拓扑关系，查询特定集群特定命名空间下服务之间调用的统计结果等功能。

以下视频将为您介绍服务依赖拓扑的使用方法：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2039-24428?source=gw.doc.media&withPoster=1&notip=1

操作步骤

查询拓扑关系

1. 登录 TSF 控制台。
2. 在左侧导航栏选择【运维中心】>【依赖分析】>【服务依赖拓扑】，进入服务依赖拓扑界面。
3. 在页面顶部数据中心位置，选择需要查看的服务所属命名空间。
4. 下方按钮选择需要依赖拓扑的时间，近30分钟、近10分钟、近5分钟以及选择特定时间段（**特定时间段的时间跨度最长为32天**）。
5. 选择之后将在下方空白处出现对应的服务依赖调用关系，单击右下角【查看图例】可查看相关说明。

- 灰色的圆圈表示主动调用的服务，箭头表示发出调用。
- 绿色的圆圈表示成功调用，黄色表示调用失败。
- 绿色和黄色组成的圆圈，绿色所占的比例是成功调用的比例，黄色为失败的比例。
- 圆圈中的数字表示平均请求耗时（单位：ms）和请求频率（单位：次/分钟）。

- 服务间带箭头曲线上的数字表示两个服务间调用的平均耗时。 在选中时间范围内，consumer-demo 调用了 provider-demo 服务，调用成功比例为 100%（绿色部分），其中平均每次调用耗时3.23ms。

> **注意：**
>
> - 当前服务依赖拓扑图中可以展示消息队列组件（CKafka）、网关组件（微服务网关、API 网关）、数据库（Redis、MySQL）。
> - 调用中出现消息队列组件时，暂时无法支持批量消费场景下的服务依赖拓扑图。

查询依赖详情

鼠标放置到图上特定位置可以显示调用依赖详情。

- 单击服务间的依赖线条，弹出面板显示该调用的主被调用方信息、调用数、调用成功率等信息，单击弹出框上的“查看调用链”可以进入到调用链查询界面。
- 单击服务圈内（白色底），可以展示该服务的调用数、调用成功率和平均调用延时，单击弹出框上的“查看调用链”可以进入到调用链查询界面。

查看监控

> **注意：**
>
> 要使用该功能，需要更新到1.12.0版本之后的 agent 和 SDK（Spring Cloud），否则无法看到调用概览。

单击依赖详情对话框中【查看图表】，侧边弹出半屏的监控数据。该页面中包括五部分监控信息：

- 请求概览：显示调用的请求量，错误率和平均响应耗时等信息。
- 实例：显示服务实例的请求量，错误率和平均响应耗时等监控信息。

- 部署组：显示部署组的请求量，错误率和平均响应耗时监控信息。
- 接口监控：显示接口的请求量，错误率和平均响应耗时监控信息。

- 事件与服务治理：显示最近发生的5条事件和正在生效的服务治理规则，单击【查看详情】可跳转到对应界面查看具体的事情详情和服务治理规则。

可视化参考

1. 对依赖拓扑图的数据说明： 图中，调用线上的时间，指从上游服务发出请求、到上游服务受到下游服务回包的时间。 调用线上会经历 Client service send、Server service receive、Server service send、Client service receive 的过程。 图中，服务圈内的数据，是站在

服务 server 端采集到的。 从平均耗时角度而言，会经历从 Server receive 到 Server send 的过程。 Client-Server 过程可参考下图：

1. 服务依赖拓扑使用应用性能指数（Apdex）对应用性能满意度进行量化，并使用不同颜色对不同区间 Apdex 的值进行标识，显示 span

   健康度，方便您快速发现应用性能问题。 应用性能指数（Apdex）的计算方式如下：

默认情况下，调用时延说明如下：

- 正常调用：指调用时延小于或等于200ms的调用。
- 慢调用：指调用时延大于200ms小于或等于800ms的调用（4 × 200ms）。
- 极慢调用：指调用时延大于800ms的调用。

Apdex 取值范围说明如下：

Apdex 取值	说明
Apdex > 0.75	表示调用延时正常，箭头为灰色。
0.75 ≥ Apdex > 0.25	表示调用延时较大，箭头为黄色。
Apdex ≤ 0.25	表示调用延时非常大，箭头为红色。

通用参考

Apdex（Application Performance Index）是由 Apdex 联盟开发的用于评估应用性能的工业标准。Apdex 标准从用户的角度出发，将对应用响应时间的表现，转为用户对于应用性能的可量化范围为 0-1 的满意度评价。

Apdex 的原理 Apdex 定义了应用响应时间的最优门槛为 T（即 Apdex 阈值，T 由性能评估人员根据预期性能要求确定），根据应用响应时间结合 T 定义了三种不同的性能表现：

性能表现	说明	示例
Satisfied（满意）	应用响应时间低于或等于 T。	例如，T 为1.5s，则一个耗时1s的响应结果则可以认为是 Satisfied 的。
Tolerating（可容忍）	应用响应时间大于 T，但同时小于或等于 4T。	例如，应用设定的 T 值为1s，则4 × 1=4s为应用响应时间的容忍上限。
Frustrated（烦躁期）	应用响应时间大于 4T。	-

调用链查询和详情

最近更新时间: 2025-02-18 16:02:00

调用链查询

调用链查询用来查询和定位具体某一次调用的情况。使用者可以通过具体的服务、接口定位、IP 等查询具体的调用过程，包括调用过程所需要的时间和运行情况。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏中选择【调用链查询】。
3. 在调用链查询中，设置查询条件，单击【查询】。

- **时间范围**：支持特定和自定义时间范围选择。特定时间范围包括：1分钟前、10分钟前和30分钟前。
- **调用服务/调用接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **被调服务/被调接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **仅查询出错的调用链**：勾选后，可以查询系统中的出错业务。
- **耗时大于**：设置耗时的阈值，可以查询系统中的慢业务。
- **客户端 IP**：客户端 IP 地址。
- **服务端 IP**：服务端 IP 地址。
- **标签**：用户在代码中设置的标签，参考 《参数传递》中设置调用链 Tag。

4. 根据查询结果，可以单击【TraceID】进入具体慢业务或出错业务，查看调用链详情。

调用链详情

您可以根据 TraceID 查询调用链的详细信息。调用链详情是为了定位在分布式链路调用过程中每个环节的耗时和异常（不包含本地方法调用情况，本地方法调用建议使用业务 log 的方式记录）。 通过调用链通常为了解决以下问题：

- 定位耗时较长的服务
- 不合理的调用逻辑（如一次请求多次调用某服务，建议改为批量调用接口）

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏中选择【调用链查询】。
3. 在 TraceID 查询中，在搜索框中输入目标 TraceID，单击【查询】。 如果搜索有结果，页面将显示调用链每个环节的耗时和状态。
4. 将鼠标移动到每个环节的时间条上并单击，会弹出 Span 的详细信息。Span 包含三部分信息：

- 基本信息：显示 Span 名、Span ID、状态和阶段耗时信息。
- 标签：显示系统和业务自定的标签
- 自定义 Metadata：显示用户在代码中设置的自定义元数据信息。参考开发手册中 《调用链》，了解如何设置 Metadata。

调用链与业务日志联动

目前要实现调用链与业务日志联动的前提条件是部署组关联的日志配置项的日志格式必须是 Spring Boot 日志格式。 在调用链详情页内，

单击日志的 icon，可以查看与这条调用链相关的业务日志。

监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导用户通过 TSF 控制台查看监控信息，可查看服务和应用两个维度的监控数据。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏，单击【监控】。
3. 设置查询条件。

- 时间范围：近6小时、近1天、近3天、或者自定义。
- 维度：选择服务或者应用。如果选择服务，需要继续选择命名空间、调用服务/调用接口、被调服务/被调接口；如果选择应用，需要继续选择应用、部署组。查看的是**被调视角**的监控数据。

4. 单击【查询】，查看监控视图。

- 健康概览：显示按分钟统计的正常请求和失败请求数量。
- 延时概览：显示请求耗时在不同时间段的分配比例。
- 状态码概览：显示不同请求状态码的分配比例。
- 并发概览：显示每分钟统计的连接并发数。

服务指标告警

最近更新时间: 2025-02-18 16:02:00

操作场景

服务指标告警功能允许您通过配置服务指标的阈值，设置接收请求量、请求平均耗时、请求失败率指标的告警。在 TSF 控制台上配置需要告警的指标策略，监控对象，和告警接收组。

操作步骤

新建告警策略

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【统计告警】-【告警】。
3. 选择【新增告警策略】。在新增策略界面，填写新建策略内容。

- 策略名称：20 字以内。
- 策略类型：选择指标告警。

- 告警对象：选择在已经在 TSF 控制台上配置的告警统计对象（需要进行统计指标的部署组）。此处支持多选。
- 选择告警策略。

- 服务指标告警策略中，可以选择**接收请求量**，**请求平均耗时**，**请求失败率**作为统计指标，自由设置触发告警的指标阈值

- 此处还可以勾选“事件告警”，在检测到服务不可用时进行告警
- 告警接收组：添加告警人员。
- 告警接收渠道：可以通过邮件或短信方式配置告警通知渠道。

4. 单击【完成】。当被监控的对象发生告警时，告警接受组的用户即可在配置的邮件或短信上收到监控信息。

查看或编辑告警策略详情

1. 登录 TSF 控制台。
2. 在【统计告警】-【告警】中，选择【告警策略列表】，查看当前配置的告警策略列表。
3. 在策略列表中，单击目标策略的策略名称，即可查看策略详情。 在这个页面上，用户可以修改当前的告警触发条件。**当某一条策略被修改时，该策略导致的告警条目在告警列表中告警状态将展示为“数据不足”。**
4. 在管理告警策略页面底部，您可以填写告警回调信息，填写公网可访问到的 URL 作为回调接口地址（域名或 IP[:端口][/path]），云监控将及时把告警信息推送到该地址。

查看告警列表

1. 登录 TSF 控制台。
2. 在左侧导航栏中，选择【统计告警】-【告警】，进入告警列表页。在此页面上，可以看到近期的告警信息。其中告警状态表明的是曾经发生的告警在当前的状态是否依然能够触发告警。当显示为“已恢复”时，表明此时告警情况已经被修复。

JVM 监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台查看 JVM 进程监控详情。

> **注意：**
>
> JVM 监控能力依赖实例上安装的探针。如您发现 JVM 监控不可用（2020年9月2日前导入集群的节点默认没有携带可以支持 JVM 监控的探针），针对虚拟机部署的业务，可以通过重新导入集群或重新安装 Agent 来使用 JVM 监控能力。针对容器部署的业务，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.2`（下载地址）。

操作步骤

进入实例监控详情页

1. 登录 【TSF 控制台】。
2. 在左侧菜单栏中，单击运维中心分类下的【JVM 监控】。
3. 在 JVM 监控的查询条件中，选择查询条件后，单击【查询】，符合条件的实例列表将会呈列在实例列表中。

- 命名空间：选择您所需要查询的服务所在的命名空间
- 服务：选择您所需要查询的服务
- 实例：选择需要查询的实例，不选则展示已选命名空间和服务下的所有实例

4. 从列表中，选择目标实例，单击【实例ID】，进入该实例的监控详情页。

查看实例 JVM 监控的基本信息

您可在【实例概览】标签页中，查看实例的基本信息、实时的指标信息以及近一小时的 CPU 使用率、堆内存使用量、活动线程数、已加载类数的时间变化曲线（曲线时间粒度为1分钟）。 基本信息：

JVM 监控概览：

查看实例 JVM 进程的内存信息

您可在【内存】标签页中，查看到进程 JVM 内存主要指标变化曲线，了解选定时段内，内存指标随时间的变化情况。 您可通过曲线图上方的【时间范围选择器】修改曲线显示的时间范围（时间粒度目前仅支持1分钟）。 支持的监控指标包括：

- 堆内存（MB）
- 非堆内存（MB）
- Eden Space（MB）
- Survivor Space（MB）
- Young GC（次数）
- Full GC（次数）

- Old Space（MB）
- Meta Space（MB）

堆内存的【max】展示的是堆内存真实可用的最大值（会扣除 to_space内存），而不是配置的 xmx 参数；JDK1.8版本部分 GC 算法（如 G1GC 的 Survivor space），其 max 值会设置为 max_int 值，此时值会显示为-1。

您还可以通过单击图片卡片右上方的放大图标 ，放大当前图片；或通过单击下载图标 ，将当前图片下载到本地（.png格式）。

查看实例 JVM 进程的线程信息

您可在【线程】标签页中，查看到当前进程的线程信息，并进行死锁检测。

线程详情

您可在【线程详情】卡片中查看到当前实例所有的 JVM 线程列表及其详细堆栈信息：

- 在【线程】tab中，以列表形式列出了当前实例的所有线程。
- 您可查看每个线程的名称、状态、CPU 利用率、堆内存使用量、阻塞计数、CPU 运行时长
- 您可根据"状态"对线程列表进行筛选，或根据"CPU利用率"或"堆内存使用量"对线程列表进行排序
- 您可通过列表上方的搜索框，快速查找感兴趣的线程。
- 单击某线程行所在区域，可在下方的展开区域中，查看到所选线程的详细信息。

所展示的线程列表为进入页面时刻获取的数据，您可通过单击卡片右上方的刷新按钮，拉取当前的最新数据。

死锁检测

在【线程详情】卡片中，单击【死锁检测】tab 卡片页中的【检测死锁】，实时检测当前进程中存在的死锁线程

> **注意：**
>
> 死锁检测目前的实现，是可以打印互相死锁的线程栈；多个线程等待同一个死锁的情况下，并不能检测出全部的死锁线程，只能找到死锁的根源；建议修复死锁后，重复检测以确认不出现嵌套死锁。

线程数

您可在【线程数】卡片中查看到当前实例的 JVM 线程总数、活动线程数、 daemon 线程数随时间的变化情况。 您可通过曲线图上方的【时间范围选择器】修改曲线显示的时间范围（时间粒度目前仅支持1分钟）。

查看实例 JVM 进程的火焰图

您可在【火焰图】标签页中，采集并查看指定时段内程序执行过程所形成的"火焰图"，用以进行性能分析。 操作步骤如下：

1. 选择您所需的数据采集时长。 目前支持的数据采集时长有：近5秒、近10秒、近30秒、近1分钟、近3分钟
2. 选择您需要采集的火焰图类型，目前支持CPU、Latency、Allocation三种类型。
3. 单击【开始采集】，等待所选时长后，可在页面中查看到所采集到的数据形成的火焰图。
4. 再次进入【火焰图】页面时，可查看到上一次采集的火焰图。您可通过步骤1和2重新采集当前实时的火焰图。

> **注意：**
>
> 数据量过大、无热点函数、或出现进程连接失败等异常情况时，火焰图可能采集失败。具体可查看 JVM 监控 【常见问题】 。

告警配置

最近更新时间: 2025-02-18 16:02:00

操作场景

云平台默认为所有用户提供云监控功能，无需用户手动开通。用户在使用了云平台某个产品后，云监控才可以开始收集监控数据。

TSF 支持监控您账户下创建的资源，包括服务、部署组、接口等等，帮助您实时掌握资源状态。您可以为监控指标配置告警规则，当监控指标达到设定的报警阈值时，云监控可以通过邮件、短信、微信、电话等方式通知您，帮助您及时应对异常情况。

告警类别

TSF 当前支持不同的告警形式配置：部署组告警，服务告警，接口告警，日志告警，应用仓库容量告警和实例告警。

TSF 当前支持的告警类别以及作用如下：

告警类别	作用对象	作用
部署组告警	部署组	指标告警：用户可以根据部署组的节点健康率配置告警。举例：当部署组中存在三个节点，希望当部署组中只保留一个节点存活的时候即发出告警，则需要配置部署组健康率在统计周期1分钟内健康率小于34%，并持续一个周期发出告警。 • 事件告警：用户可以对弹性伸缩触发扩缩容/部署组异常事件进行告警配置。如想实现当某个部署组弹性伸缩触发扩容触发告警则配置事件告警，事件内容为弹性伸缩规则触发扩容。
服务告警	微服务	指标告警：用户可以配置指标告警来针对微服务的接收请求平均耗时、接收请求失败率、接收请求量、http响应码4xx、http响应码5xx进行告警配置。 事件告警：用户可以对服务离线和服务熔断进行告警配置。如想实现当某个微服务离线则触发告警则配置事件告警，事件内容为服务离线。
接口告警	服务接口	监控服务接口请求量、请求错误率、请求平均耗时指标。当微服务接口请求数、请求错误率、请求平均耗时达到一定阈值则触发告警。
日志告警	部署组以及部署组运行日志中的关键词	统计日志中的关键词出现频率，当某些日志关键词出现频率超过一定限度则触发告警。例如，用户可以配置日志中 “error”等关键词出现的频率一分钟内出现10次即告警。
应用仓库容量告警	应用仓库	当用户应用仓库已用容量/总容量比例高于80%，触发告警。
实例告警	服务实例	统计实例的 FULL_GC 事件次数，当数量达到一定阈值后则触发告警。

操作步骤

> **注意：**
>
> 当前云产品的告警统一收归到云监控控制台进行配置。日志告警的配置请参见 [日志告警]。

配置告警规则

创建的告警会将一定周期内监控的指标与给定阈值的情况进行比对，从而判断是否需要触发相关通知。当 TSF 状态改变而导致告警触发后，您可以及时进行相应的预防或补救措施，合理地创建告警能帮助您提高应用程序的健壮性和可靠性。

1. 登录 【云监控控制台】。
2. 在告警策略页面，选择好策略类型和要设置告警的实例，设置好告警规则和告警通知模板。
   - **策略类型**：选择【TSF】。
   - **告警对象**：选择需要配置告警策略的 TSF 资源。
   - **触发条件**：支持【选择模板】和【手动配置】，默认选择手动配置，手动配置参见以下说明，新建模板参见 [新建触发条件模版] 。

> **注意：**
>
> - 指标：例如"部署组节点健康率"，选择统计粒度为1分钟，则在1分钟内，部署组节点健康率连续N个数据点超过阈值，就会出发告警。
> - 告警频次：例如"每30分钟警告一次"，指每30分钟内，连续多个统计周期指标都超过了阈值，如果有一次告警，30分钟内就不会再次进行告警，直到下一个30分钟，如果指标依然超过阈值，才会再次告警。

- **通知模板**：选择通知模版，也可以新建通知模版，设置告警接收对象和接收渠道。

iii. 单击【完成】，完成配置。

有关告警的更多信息，请参考 【云监控告警服务】。

新建触发条件模板

1. 登录 【云监控控制台】。
2. 在左侧导航栏中，单击【触发条件模板】，进入触发条件列表页面。
3. 在触发条件模板页单击【新建】。
4. 在新建模板页，配置策略类型。
   - **策略类型**：选择【TSF】。
   - **触发条件**：设置告警出发条件。

5. 确认无误后，单击【保存】。 ）

服务调用查询

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导用户通过 TSF 控制台查看监控信息，可查看服务和应用两个维度的监控数据。

以下视频将为您介绍 TSF 监控功能的使用方法：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2039-24430?source=gw.doc.media&withPoster=1&notip=1

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【依赖分析】>【服务调用查询】。
3. 设置查询条件。

- 时间范围：近6小时、近1天、近3天、或者自定义。
- 维度：选择服务或者应用。如果选择服务，需要继续选择命名空间、调用服务/调用接口、被调服务/被调接口；如果选择应用，需要继续选择应用、部署组。查看的是**被调视角**的监控数据。

4. 单击【查询】，查看监控视图。

- 健康概览：显示按分钟统计的正常请求和失败请求数量。
- 延时概览：显示请求耗时在不同时间段的分配比例。
- 状态码概览：显示不同请求状态码的分配比例。
- 并发概览：显示每分钟统计的连接并发数。

查看SideCar监控信息

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台查看 SideCar 监控信息，包含 SideCar 运行状态、SideCar 监控数据和 SideCar 日志。

> **注意：**
>
> 以下信息仅针对容器和虚拟机部署的 Mesh 应用生效。

操作步骤

1. 登录 TSF 控制台。

2. 在左侧导航栏选择【部署组】，单击"部署组 ID "进入部署组详情，默认在【服务实例列表】页，可查看 SideCar 状态。
3. 选择您感兴趣的实例，将鼠标放置在 SideCar 状态上，单击【查看日志】，跳转至 SideCar 日志 tab 页。

4. 在 SideCar 日志页面，选择日志类型、组件和时间，可查看日志信息。

5. 选择【SideCar监控】，查询 SideCar 的版本与组件状态和基本信息。

监控
实例监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，查看所选服务下某实例的详细监控信息。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧菜单栏中，单击【服务监控】，选择好命名空间、时间范围和微服务，可查看当前筛选条件下的服务的监控列表信息。
3. 单击目标服务操作列的【查看监控详情】，进入服务概览界面。
4. 选择页面上方【实例监控】页签，查看该服务下实例的监控信息。

- 请求概览 您可以点击具体指标右边的按钮对该服务下所有实例的指标进行排序，通过实例的请求错误率和耗时等指标判断实例是否有异常。
- JVM 监控：仅 Java 应用支持 JVM 监控能力。

JVM 监控能力依赖实例上安装的探针。如您发现 JVM 监控不可用（2020年9月2日前导入集群的节点默认没有携带可以支持 JVM 监控的探针），针对虚拟机部署的业务，可以通过重新导入集群或重新安装 Agent 来使用 JVM 监控能力。针对容器部署的业务，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.2`。

您可以单击右上角【前往查看实例诊断与实例日志】跳转至服务实例详情页面查看详细JVM监控信息。

5. 单击目标实例的"实例ID"，进入实例概览页面。 您可在实例详情中的【实例概览】中，查看到该实例的以下信息：

- 基本信息 您可以通过【实例所在节点IP】跳转到云服务器页面，查看云服务器信息。
- 经过该实例的请求监控信息
- 该实例的 JVM 监控概览（如果该实例为 Java 实例）

内存 如果实例为 Java 实例，您可在实例详情的【内存】tab 页中，查看到该实例的内存监控详情：

堆内存的【max】展示的是堆内存真实可用的最大值（会扣除 to_space内存），而不是配置的 xmx 参数；JDK1.8版本部分 GC 算法（如 G1GC 的 Survivor space），其 max 值会设置为 max_int 值，此时值会显示为-1。

您还可以通过单击图片卡片右上方的放大图标 ，放大当前图片；或通过单击下载图标 ，将当前图片下载到本地（.png格式）。

线程 如果实例为 Java 实例，您可在实例详情的【线程】tab 页中，查看到该实例的线程数变化情况、线程列表及线程详情，并进行死锁检

测，了解实例的死锁情况。 线程数： 死锁检测：

死锁检测目前的实现，是可以打印互相死锁的线程栈；多个线程等待同一个死锁的情况下，并不能检测出全部的死锁线程，只能找到死锁的根源；建议修复死锁后，重复检测以确认不出现嵌套死锁。 火焰图 如果实例为 Java 实例，您还可以在【火焰图】tab 页中，采集并查看指定时段内程序执行过程所形成的"火焰图"，用以进行性能分析。

数据量过大、无热点函数、或出现进程连接失败等异常情况时，火焰图可能采集失败。具体可查看 JVM 监控 【常见问题】 。 请求详情 您可在【请求详情】tab页中，查询经过该实例的请求的调用链详情。

列表页中的"耗时"和"状态"，为服务处理请求的本地耗时和状态。

日志 您可在【日志】tab页中，查看该实例上的日志，包括业务日志及 JVM 的 GC 日志。

您也可以开启右上角【查看实时日志】按钮查看实时日志信息。

若某个实例出现问题，会导致请求发送到该问题实例上，此时可以开启【屏蔽实例】来手动下线该实例。服务被屏蔽后，该服务实例将不会被其他服务发现，流量不会分发到该实例上。

在服务治理页面，点击目标服务的"ID/名称"，进入服务实例列表，在操作栏可以屏蔽实例。

接口监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，查看所选服务下某接口的详细监控信息。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧菜单栏中，单击【服务监控】，选择好命名空间、时间范围和微服务，可查看当前筛选条件下的服务的监控列表信息。
3. 单击目标服务操作列的【查看监控详情】，进入服务概览界面。
4. 选择页面上方【接口监控】页签。

- **请求概览** 单击右上角【查看调用链详情】查询经过该接口的请求的调用链详情。

> **注意：**
>
> 列表页中的”耗时”和”状态”，为服务处理请求的本地耗时和状态。

单击调用链的右三角，可查看调用链详情，点击日志按钮可以看到该调用链的日志详情。

单击操作栏的【详情】，在标签页面可以看到数据库的调用情况。

- 服务上游：展示服务上游的监控情况。

服务监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，查看某命名空间下所有微服务的运行状态，并且根据微服务的监控指标初步判断该是否出现异常。

操作步骤

1. 登录 【TSF 控制台】 。

2. 在左侧菜单栏中，单击【服务监控】，选择好命名空间、时间范围和微服务，可查看当前筛选条件下的服务的监控列表信息。 当某个服务的请求量，请求错误率，响应耗时等指标出现异常时，您可单击该服务操作列的【查看监控详情】查看详细监控信息：

- 该服务的相关的依赖拓扑图（服务及其上下游服务）：
- 请求概览：统计服务被请求的监控详情。
- 监控信息：选择【监控】页签，可查看服务的请求量、错误率、响应耗时、响应耗时分布和 HTTP 状态码等监控信息。 说明：响应耗时中 p95 代表线上95%的请求耗时都小于某个时间。 单击每条曲线右上角可添加指标同环比，后面部署组、实例和接口的各指标曲线同样可查看。
- 统计信息：选择【统计】页签，可查看该服务的请求量、错误率、响应耗时和 HTTP 状态码等监控指标的统计信息。

3. 在服务详情页面，您可以继续查看该服务下的部署组、实例和接口列表的监控信息。

部署组监控 单击【部署组监控】页签，可以查看该服务下部署组的监控信息。

您可以点击具体指标右边的按钮对该服务下所有部署组的指标进行排序，通过部署组的请求错误率和耗时等指标判断部署组是否有异常。若发现部署组异常，您可以单击"部署组ID/名称"继续向下排查问题，参考 【部署组监控】 。 实例监控 单击【实例监控】页签，查看该服务下实例的监控信息。

- 请求概览

您可以点击具体指标右边的按钮对该服务下所有实例的指标进行排序，通过实例的请求错误率和耗时等指标判断实例是否有异常。 若发现实例异常，您可以单击"实例ID/名称"继续向下排查问题 。

- JVM 监控：仅 Java 应用支持 JVM 监控能力。 JVM 监控能力依赖实例上安装的探针。如您发现 JVM 监控不可用（2020年9月2日前导入集群的节点默认没有携带可以支持 JVM 监控的探针），针对虚拟机部署的业务，可以通过重新导入集群或重新安装 Agent 来使用 JVM 监控能力。针对容器部署的业务，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.1`

您可以单击右上角【前往查看实例诊断与实例日志】跳转至服务实例详情页面查看详细 JVM 监控信息。 接口监控 单击【接口监控】页签，可以查看该服务下接口的监控信息。

- 请求概览 您可以点击具体指标右边的按钮对该服务下所有接口的指标进行排序，通过接口的请求错误率和耗时等指标判断接口是否出现异常。 若发现接口异常，您可以单击右上角【查看调用链详情】继续向下排查问题
- 服务上游：展示服务上游的监控情况。 :::

部署组监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，查看所选服务下某部署组的详细监控信息。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧菜单栏中，单击【服务监控】，选择好命名空间、时间范围和微服务，可查看当前筛选条件下的服务的监控列表信息。
3. 单击目标服务操作列的【查看监控详情】，进入服务概览界面。

4. 单击【部署组监控】页签，可以查看该服务下部署组的监控信息。 您可以点击具体指标右边的按钮对该服务下所有部署组的指标进行排序，通过部署组的请求错误率和耗时等指标判断部署组是否有异常。
5. 单击目标部署组的"部署组ID/名称"，进入部署组页面。

- 您可以在基本信息标签页查看部署组基本信息。

- 您可以在日志标签页查看部署组的日志，通过右上角按钮切换查看实时日志和历史日志。

调用链查询

最近更新时间: 2025-02-18 16:02:00

调用链查询

操作场景

调用链查询用来查询和定位具体某一次调用的情况。使用者可以通过具体的服务、接口定位、IP 等查询具体的调用过程，包括调用过程所需要的时间和运行情况。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中选择【调用链查询】。
3. 在调用链查询中，设置查询条件，单击【查询】。

- **时间范围**：支持特定和自定义时间范围选择。特定时间范围包括：5分钟前、10分钟前和30分钟前。
- **调用服务/调用接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **被调服务/被调接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **仅查询出错的调用链**：勾选后，可以查询系统中的出错业务。
- **状态码为**：状态码。
- **客户端 IP**：客户端 IP 地址。
- **服务端 IP**：服务端 IP 地址。
- **耗时范围**：设置耗时的阈值。
- **标签**：用户在代码中设置的标签 ，最多支持20个tag。

4. 根据查询结果，可以单击【TraceID】进入具体慢业务或出错业务，查看调用链详情。

调用链详情

操作场景

您可以根据 TraceID 查询调用链的详细信息。调用链详情是为了定位在分布式链路调用过程中每个环节的耗时和异常（不包含本地方法调用情况，本地方法调用建议使用业务 log 的方式记录）。 通过调用链通常为了解决以下问题：

- 定位耗时较长的服务
- 不合理的调用逻辑（如一次请求多次调用某服务，建议改为批量调用接口）

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中选择【调用链查询】。

3. 在 TraceID 查询中，在搜索框中输入目标 TraceID，单击【查询】。 如果搜索有结果，页面将显示调用链每个环节的耗时和状态。
4. 将鼠标移动到每个环节的时间条上并单击，会弹出 Span 的详细信息。Span 包含三部分信息：

- 基本信息：显示 Span 名、Span ID、状态和阶段耗时信息。
- 标签：显示系统和业务自定的标签。
- 自定义 Metadata：显示用户在代码中设置的自定义元数据信息。参考开发手册中 [调用链] ，了解如何设置 Metadata。

调用链与业务日志联动

目前要实现调用链与业务日志联动的前提条件是部署组关联的日志配置项的日志格式支持日志与调用链联动，并且必须遵守日志格式规范。

- Spring Boot 格式默认可以支持日志调用链联动。
- 自定义 Logback、自定义 Log4j、自定义 Log4j2 和单行/多行文本格式日志需要日志 pattern 中添加 %trace 才可以支持日志和调用链联动。
- Nginx Access 和无解析规则格式日志无法支持日志和调用链联动。

在调用链详情页内，单击日志的 icon，可以查看与这条调用链相关的业务日志。

运维排障指引

最近更新时间: 2025-02-18 16:02:00

本文主要介绍使用 TSF 进行排障的基础思路和操作步骤，方便运维人员在使用 TSF 过程中进行问题定位和处理。运维排障整体思路为：

1. 通过告警或者业务大屏发现问题。
2. 通过查看服务监控初步定位是实例、部署组还是接口出现问题。
3. 根据具体问题所在处继续向下进行详细排查。

步骤一：发现问题

发现问题通常有告警和业务大屏两种方式： 方式一：配置告警信息 配置告警信息，通常有以下三个层面告警：

- 微服务业务级别：微服务自身状态，微服务自身请求量、耗时和错误率，微服务接口请求量、耗时和错误率，部署组健康率等。
- 底层资源级别：CPU 内存、云主机状态、容器集群等。
- 组件级别：数据库实例、CKafka 实例等。

操作指引： 在云监控界面为 TSF 服务配置告警策略和 Dashboard，步骤如下：

1. 配置告警策略，通过告警发现问题。
2. 配置 Dashboard，通过展示的监控指标发现问题。

方式二：查看业务大屏 查看业务大屏，通过微服务列表和接口列表查看当前运行的服务的状态。

操作指引： 在 TSF 控制台通过查看服务依赖拓扑图和接口列表，步骤如下：

3. 查看服务依赖拓扑图，通过依赖拓扑图发现问题。
4. 查看服务列表中服务运行健康状态，通过业务请求量、耗时、错误率等发现问题。

步骤二：初步定位

当发现问题后，查看服务监控初步确定是服务实例还是部署组还是接口出现问题。

步骤三：详细排查

初步定位问题后，排障思路如下：

- 部署组问题：可以考虑程序包版本问题或者查看日志等方式排查
- 实例问题：可以通过查看日志和 JVM 进程等方式排查 。
- 接口问题：可以通过查看日志和调用链等方排查 。

配置Dashboard

最近更新时间: 2025-02-18 16:02:00

操作场景

Dashboard 是云监控针对云产品指标监控数据，提供的具备可视化和分析功能的智能仪表盘。

您可以对 TSF 的指标创建 Dashboard，Dashboard 会自动将监控数据以精美的图表形式呈现在监控面板中，使监控数据更加直观，协助您通过趋势和异常值分析指标。

TSF 当前支持展示的告警指标以及作用如下：

告警类别	作用对象	作用
日志告警	部署组以及部署组运行日志中的关键词	统计日志中的关键词出现频率。
服务告警	微服务	监控微服务的在线、离线状态以及服务被请求的请求量、错误率、耗时。
部署组告警	部署组	监控部署组中节点的运行健康程度。
接口告警	服务接口	监控服务接口的运行健康程度。
分布式调度任务告警	普通任务、工作流任务	监控普通任务和工作流任务的执行失败次数。

操作步骤

1. 登录 云监控控制台。
2. 在左侧导航栏中单击【Dashboard】>【Dashboard 列表】，进入 Dashboard 列表页。
3. 单击 Dashboard 列表左上角的【新建 Dashboard】，进入新建 Dashboard 管理页。

4. 单击面板区的，在弹出的窗口中填写面板名并选择 Dashboard 所属文件夹。
5. 单击【确定】即可快速创建 Dashboard。
6. 进入 Dashboard 管理页。单击【新建】>【新建图表】，进入编辑图表页配置指标信息

- **指标**：选择TSF告警指标
- **筛选**：选择需要展示监控指标的实例。
- **group by**（标签筛选条件无此功能）：类似 SQL 的 Group by 功能，根据指定标签对数据进行分组后再按照聚合算法聚合。当您不选择任何标签时，可自定义统计周期内指标统计方式，支持平均值（avg）、最大值（max）、最小值（min）和求和（sum）统计方式。
- **对比**：支持环比（昨天同时段）、同比（上周同时段）和自定义时间对比。当您都勾选后，图表会出现所选实例昨天同时段监控曲线和上周同时段监控曲线，方便您进行数据对比。
- **左 Y 轴、右 Y 轴**：支持调整 Y 轴左右放置。
- 更多配置：
- **别名**：支持一键命名所有实例别名。如需不同实例命名不同的别名，可新建多个指标，在各指标下输入别名。
- **开启排序功能**：图表所绑定的实例将按排序规则和展示数量进行排序，用于实现大批量监控机器高低负载功能。

- **排序规则**：支持多种方式对指标进行排序，可根据排序结果对实例进行筛选。
- **展示数量**：展示实例数量。 例如：设置排序规则为"最大值；降序"，展示数量为10。表示：图表中将按降序展示最大值 TOP10 的实例。

7. 单击右上角【保存】，完成监控指标设置。

组件中心
微服务网关
分组管理
分组与API管理

最近更新时间: 2025-02-18 16:02:00

操作场景

微服务网关通过分组管理微服务 API。 分组是微服务网关管理 API 的维度，同一个分组下的 API 使用相同的鉴权方法，使用相同的一个或者多个密钥进行访问鉴权。每一个分组有一个固定的 context 作为访问路径中的 path 参数。

操作步骤

新建分组

1. 登录 TSF 控制台。
2. 在左侧菜单栏选择【微服务网关】>【网关管理】。
3. 在微服务网关详情页，单击【分组管理】>【新建分组】，填写分组信息。

- 分组名称： 最长为60个字符，只能包含小写字母、数字及分隔符（ _ 、 - ），且不能以分隔符开头或结尾。
- 访问 context：context 是用户访问网关管理的某一个 API 的路径的路径参数。以"/"开头，不能为空。
- 托管 API 类型：当前微服务网关可以托管两种不同类型的 API。
- 微服务 API，即部署在 TSF 上的微服务的 API
- 外部 API，即后端服务未部署在 TSF 上或未注册到 TSF 的注册中心的服务的 API。托管外部 API 时，需要用户手动填写后端服务的 Host 地址。

- 鉴权类型：密钥鉴权或无鉴权。当选择密钥对鉴权时，请求参数中不带正确的 SecretId 和签名的访问会被拒绝。 img

4. 单击【保存&下一步】，将分组与微服务网关应用的部署组进行绑定。

> **注意：**
>
> - 当微服务网关作为微服务内部调用网关时，建议绑定微服务下全部部署组。当微服务网关作为外部与微服务间调用的外部网关时，可以选择将某个分组与某个或全部部署组绑定，通过访问部署组域名或IP来访问网关托管的API。
> - 只有将网关应用的部署组与分组绑定后，才能保证访问部署组 IP 后与对应 API path 访问畅通。
> - 此处分组绑定的部署组是微服务网关应用的部署组，也就是用作部署网关的节点所在的部署组。并非后端微服务的部署组。
> - 单个微服务网关部署组不能绑定多个 context 相同的分组。

5. 新建分组成功后，单击分组ID/名称，进入访问信息页，可查看分组的基本信息，并对绑定部署组和密钥进行操作管理。

删除分组

在【分组管理】页面，单击目标分组操作列的【删除】，即可删除分组。

> **注意：**
>
> 如果分组下有 API，则该分组不能被删除。

导入 API

1. 在左侧菜单栏选择【微服务网关】>【网关管理】。
2. 在微服务网关详情页，点击顶部【API管理】，选择【导入API】，将某个微服务下的 API 导入分组。

- 微服务 API： 支持两种导入方式：导入某个微服务下全部 API 和导入某个微服务下部分 API。 

> **注意：**
>
> - 当同时导入某个微服务下全部 API 时，后台导入完成前请勿重复导入。
> - 单部署组最多支持创建2000个API，超出后不保证发布成功。

- 外部 API： 填写请求路径、请求方法以及 host 地址。

> **注意：**
>
> 可以为外部 API 配置超时时间。

发布/下线分组

当且仅当分组被发布时，才能通过微服务网关访问微服务 API。

- 导入 API 后，单击操作列的【发布分组】，该分组即可被发布。
- 如果某一个分组不想被访问，可以单击操作列的【下线分组】，将分组下线。

访问微服务 API

分组下某个微服务 API 的访问路径为： `网关的域名或网关的 IP + port/分组 context/微服务 API 所在的微服务命名空间名称/微服务名称/API 路径`

举例：当分组 context 为 sell，所在命名空间为 test-env，微服务名称为 consumer，API 路径为 `/echo/{test}` 时，则访问路径为 `域名/sell/test-env/consumer/echo/{test}` 。 单击 API 路径后面的复制按钮，会自动复制从 context 以后的路径。 粘贴后的内容为 `/sell/test/provider-demo/v1/user/delete/user` 。

访问外部 API

访问外部 API 的路径为： `微服务网关域名或 VIP 或服务名（仅当客户端为注册到 TSF 中的微服务时可以通过网关服务名访问）/分组 context/API 路径` 。

微服务网关域名

在用户公有云场景下，需要用户将部署网关的节点的 IP 配置在 DNS 或 LB 上，以获取统一的访问网关的统一域名或统一 VIP。

- 对于虚拟机部署组，用户可以将虚拟机部署组的机器 IP 绑定在 CLB 后端。
- 对于容器部署组，用户可以设置容器部署组的访问方式为公网访问。

设置API超时时间

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台，为微服务网关托管的微服务 API 配置超时时间。

操作步骤

1. 登录 TSF 控制台。
2. 选择【微服务网关】>【网关管理】，单击目标网关操作栏的【配置分组与API】，进入【分组管理】页面。
3. 选择一个需要调整 API 超时时间的分组，单击分组 ID，进入详情页的【API列表】页面。
4. 设置超时时间（0 < t ≤ 600000ms，其中 t 为正整数），单击【提交】。

- 方式一：选择一个 API，单击操作列的【更多】>【设置超时时间】，为 API 配置超时时间。
- 方式二：通过页面上的【设置超时时间】按钮，批量为 API 配置超时时间。

> **注意：**
>
> - 若未对某个 API 配置超时时间，则使用微服务网关程序包中 application.yaml 文件中配置的超时时间。
> - 当前提供的微服务网关 Demo 包的默认超时时间为10s。

设置API限流规则

最近更新时间: 2025-02-18 16:02:00

操作背景

该任务指导您通过TSF控制台，为网关下的每一个 API 配置限流规则设置最大请求次数。

操作步骤

1. 在【TSF 控制台】，选择【微服务网关】>【网关管理】，单击目标网关的ID/名称，进入网关详情页。
2. 单击【API管理】，进入API 列表页面，选择目标API操作栏的【更多】>【编辑限流规则】。
3. 在设置限流规则对话框中，选择限流策略为"设置最大请求数"，并填写最大请求次数（QPS）。 配置了限流规则后，可以在配置 API 限流规则的微服务的服务限流列表中，找到该限流规则。

配置鉴权

最近更新时间: 2025-02-18 16:02:00

操作背景

该任务指导您通过TSF控制台，使用密钥对对访问进行鉴权。

操作步骤

1. 在【TSF 控制台 】，选择【微服务网关】>【网关管理】，单击目标网关操作栏的【配置分组与API】，进入【分组管理】页面。
2. 单击目标分组的【ID/名称】，进入【访问信息】页面。
3. 在基本信息模块，确认鉴权类型为"密钥对鉴权"。如不是，可单击模块右上角的【编辑】进行修改。
4. 单击【新建密钥】，填写密钥名。 密钥名最长50个字符，支持 a-z 、 A-Z 、 0-9 、 _ 。
5. 单击【确认】，会自动生成密钥对，并展示在控制台上。 您可以通过密钥末尾的复制图标进行复制。 当密钥需要禁用和更换时，可以单击操作列的【禁用】或【更换】完成操作。

单元化部署（操作指南）

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台上创建单元化规则并使用单元化功能。

操作步骤

步骤一：部署微服务网关

1. 登录 【TSF 控制台】，在左侧导航栏单击【应用管理】。

2. 在应用管理页面，单击【新建应用】，填写应用名、部署方式和应用类型。 应用类型选择**微服务网关应用**。
3. 单击【提交】，在弹出的提醒框中选择【确认】，跳转至【上传程序包】页面。
4. 在【上传程序包】页面，单击【上传程序包】。 微服务网关 Demo 程序包 GitHub 地址，相关操作请参考 【微服务网关开发】。

> **注意：**
>
> - 微服务网关兼具灵活性和便捷性。当用户仅需要使用请求转发、密钥对鉴权、服务治理等 TSF 已经提供的产品能力时，您仅需要使用 TSF-demo 中的 msgw - demo 程序包即可，可以选择其中的 msgw - zuul 以及 msgw - scg 来分别部署基于 Zuul 和基于 Spring Cloud Gateway 的微服务网关。
> - 当您希望定制化微服务网关的转发逻辑，您可以选择自行开发微服务网关，并依赖 TSF 的网关 SDK 来进行开发。
> - 直接使用开源的微服务网关（Demo 中的 opensource 版本）会导致 TSF 产品页面中的网关功能不可用。不建议您使用开源版本的微服务网关。

5. 单击页面上方的【部署组】，进入部署组页面，单击【新建部署组】。

> **注意：**
>
> 单元化部署仅支持部署在全局命名空间的网关。

6. 部署成功后，您可以在【微服务网关】>【网关管理】查看微服务网关。

步骤二：启用单元化部署并创建单元范围

新建好的网关默认状态是不支持单元化部署，需要手动开启单元化部署功能。

1. 在【网关管理】界面，单击目标网关的ID/名称，进入【基本信息】页面。
2. 在【单元化配置】模块，点击右上角的【编辑】，开启单元化部署功能。

> **注意：**
>
> 切换单元化部署状态，当前网关下分组、分组下API以及插件配置都将被清除，请谨慎操作。

3. 进入【单元范围】页面，点击【关联命名空间】，将命名空间添加到单元范围中。

> **注意：**
>
> - 同一个普通命名空间只能添加到一个单元范围中，不可重复添加。
> - 同一单元范围中最多可添加200个命名空间。

步骤三：创建单元化规则

单元化部署场景下通过微服务网关访问微服务路径为 `微服务网关域名或 IP/分组 context/微服务名称/api` 。请求将依据单元化流量划分规则选择目的地命名空间。

1. 在【单元化规则】页面，单击【新建规则】，填写规则信息，创建单元化规则。

> **注意：**
>
> 1条单元化规则最多可添加16个标识配置。

2. 在单元化规则列表，单击目标规则状态栏按钮，使其生效。

> **注意：**
>
> - 一个网关下可创建多条单元化规则，但同时只能有一条规则生效，当一条规则生效时，其他正在生效的规则将默认关闭。
> - 已生效的规则不能删除，关闭后才能删除。

步骤四：创建分组并导入 API

1. 在【分组管理】页面，单击【新建分组】，填写分组信息。

- 分组名称： 最长为60个字符，只能包含小写字母、数字及分隔符（ `_` 、 `-` ），且不能以分隔符开头或结尾。
- 访问 context：context 是用户访问网关管理的某一个 API 的路径的路径参数。以“/”开头，不能为空。
- 托管 API 类型：选择微服务API，单元化部署场景不支持托管外部API。
- 鉴权类型：密钥鉴权或无鉴权。当选择密钥对鉴权时，请求参数中不带正确的 SecretId 和签名的访问会被拒绝。 

2. 单击【保存&下一步】，将分组与微服务网关应用的部署组进行绑定。

> **注意：**
>
> - 当微服务网关作为微服务内部调用网关时，建议绑定微服务下全部部署组。当微服务网关作为外部与微服务间调用的外部网关时，可以选择将某个分组与某个或全部部署组绑定，通过访问部署组域名或IP来访问网关托管的API。
> - 只有将网关应用的部署组与分组绑定后，才能保证访问部署组 IP 后与对应 API path 访问畅通。
> - 此处分组绑定的部署组是微服务网关应用的部署组，也就是用作部署网关的节点所在的部署组。并非后端微服务的部署组。
> - 单个微服务网关部署组不能绑定多个 context 相同的分组。

3. 在【API管理】页面，选择目标分组，单击【导入API】，选择需要导入API的命名空间和微服务，将API导入分组。

> **注意：**
>
> 单元部署化场景下，只需要导入单元组中一个命名空间中的API，单元组中的其他命名空间的API将同时被导入。例如上图中导入命名空间shenzhen中的API，则单元组中其他命名空间如shanghai、chongqing、guangzhou等中的API将被自动导入分组。

4. 在【分组管理】页面，选择目标分组操作栏的【发布分组】。

- 当且仅当分组被发布时，才能通过微服务网关访问微服务 API。
- 如果某一个分组不想被访问，可以单击操作列的【下线分组】，将分组下线。

步骤五：调试 API

1. 在【API管理】页面，选择目标分组，单击需要调试的 API 路径，进入 API 详情页。
2. 在 API 详情页，单击上方【调试】页签后，选择右上角【调试】，进入API调试详情页。
3. 在 API 调试详情页，填写调用 API 的默认参数，单击【发送请求】，在页面右方可看到调用 API 返回结果，结果正常即成功。

4. 在控制台导航树选择【依赖分析】>【调用链查询】，输入刚刚调用API的查询条件，单击【查询】。
5. 根据查询结果，可以单击【TraceID】进入具体业务，查看调用链详情。

微服务网关跨命名空间访问

最近更新时间: 2025-02-18 16:02:00

使用场景

通常在一个微服务系统中，会有两个位置使用微服务网关：

- 微服务系统的边界，如前端与微服务后台之间的网关。

- 不同的微服务业务系统之间（如不同部门的业务后台都使用微服务架构）通过微服务网关进行调用；开发环境和预发布环境的业务都希望通过微服务网关访问某一个公共接口；在 set 化部署（分区部署）和单元化架构部署中，也常常通过微服务网关做不同业务系统之间访问的桥梁。在这类场景中，通常都希望不同系统之间的访问安全可控：不同系统之间进行物理或逻辑的隔离，通过网关实现"最小必须"的访问。

在微服务平台（TSF）中，我们通过命名空间来实现不同业务系统的访问逻辑隔离，本文主要介绍如何使用命名空间和微服务网关实现不同微服务业务系统之间的调用。

微服务网关与命名空间

在 [命名空间] 中，介绍了 TSF 中的两种命名空间：全局命名空间与非全局命名空间。其中：

- 全局命名空间中的微服务可以被其他命名空间中的微服务调用（典型场景：公共服务），但是不能调用其他命名空间中的微服务。
- 不同的非全局命名空间（包含系统命名空间）中部署的微服务不能直接相互调用。

微服务网关也可以部署在全局命名空间和非全局命名空间中。在实际使用中，用户创建微服务网关应用下的部署组时，可以选择是否将网关部署在全局命名空间中。不同命名空间与微服务网关的关系如下：

部署位置	微服务网关调用其他微服务	其他微服务调用微服务网关
全局命名空间	微服务网关可以调用所有命名空间中的微服务	微服务网关可以被所有微服务调用
非全局命名空间	微服务网关仅可以调用所在命名空间下的微服务	可以被所在命名空间中的微服务调用（常见）或被全局命名空间中的网关调用（少见）

示意图如下：

- 非全局命名空间中的微服务网关：

- 全局命名空间中的微服务网关：

用户可以通过全局命名空间中的微服务网关达到跨命名空间访问的目的。非全局命名空间中的微服务调用全局命名空间中的微服务网关调用另一个非全局命名空间中的微服务。此时，全局命名空间中的网关就是不同非全局命名空间中微服务调用的桥梁。在实际的业务场景中，您可以将不同部门的业务部署在不同的命名空间中，通过全局命名空间的微服务网关来进行跨部门的调用。

调用方法

网关调用微服务

微服务网关调用微服务时，用户可以通过网关托管 API 的路径进行调用。访问路径如下：`IP 或域名/分组 context/命名空间名/微服务名/API 路径`。访问路径规则与网关是否部署在全局命名空间中无关，但当微服务网关部署在非全局命名空间中，访问路径中却填写了错误命名空间名，则访问不通。

微服务调用网关

用户可以通过网关的服务名调用网关。当非全局命名空间 namespace1 中的微服务 consumer 希望通过全局命名空间中的网关 zuul 访问 namespace2 中的微服务 provider 时，可以在 consumer 中直接访问微服务 zuul，访问路径为 `/context/namespace2/provider/api-path`。

插件管理
配置JWT插件

最近更新时间: 2025-02-18 16:02:00

JWT 原理

JWT（JSON Web Token）本质是一个 Token，是一种紧凑的 URL 安全方法，用于在网络通信的双方之间传递声明。 JWT 的原理是，客

户端通过 JWT 认证服务器认证以后，会返回给客户端一个 JWT 令牌（Token），示例如下（真实长度会更长）： JWT 分为三部分：Header（头部）、Payload（负载）、Signature（签名），中间用点（.）分隔成三个部分。 此后，客户端与服务端通信的时候，都要携带这个 JWT 令牌（Token）。微服务网关 JWT 插件凭此令牌（Token）来校验客户端权限，服务端就不再保存任何 session 数据，此时服务端变成无状态了，比较容易实现横向扩展

Header

Header 部分是一个 JSON 对象，描述 JWT 的元数据，示例如下：

```
{
"alg": "HS256",
"typ": "JWT"
}
```

- `alg` 属性表示签名的算法（algorithm），默认是 HMAC SHA256（写成 HS256）。
- `typ` 属性表示这个令牌（token）的类型（type），JWT 令牌统一写为 `JWT`。

最后，将上面的 JSON 对象使用 Base64URL 算法转成字符串，即为 JWT 令牌中的 Header 部分。

Payload

Payload 部分也是一个 JSON 对象，用来存放实际需要传递的数据，包括官方定义的字段和用户自定义字段。

参数	英文全称	是否必选	说明	取值要求
iss	Issuer Identifier	是	提供认证信息者的唯一标识。	一般是一个 HTTPS 的 URL（不包含 querystring 和 fragment 部分）
sub	Subject Identifier	是	iss 提供的 EU 的标识，在 iss 范围内唯一。它会被 RP 用来标识唯一 的用户。	最长为255个 ASCII 字符
aud	Audience(s)	是	标识 ID Token 的受众。	必须包含 OAuth2 的 client_id
exp	Expiration time	是	过期时间，超过此时间的 ID Token 会作废不再被验证通过。	-
iat	Issued At Time	是	JWT 构建的时间。	-

参数	英文全称	是否必选	说明	取值要求
auth_time	AuthenticationTime	否	EU 完成认证的时间 。如果 RP 发送 AuthN 请求的时候携带 max_age 的参数，则此 Claim 是必须的。	-
nonce	-	否	RP 发送请求的时候提供的随机字符串，用来减缓重放攻击，也可以来关联 ID Token 和 RP 本身的 Session 信息。	-
acr	Authentication Context Class Reference	否	表示一个认证上下文引用值，可以用来标识认证上下文类。	-
amr	Authentication Methods References	否	表示一组认证方法。	-
azp	Authorized party	否	结合 aud 使用。只有在被认证的一方和受众（sud）不一致时才使用此值，一般情况下很少使用。	-

下面是一个典型数据格式的示例，供参考：

```
{
"iss": "http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/tsf/msgw/jwt",
"sub": "aaaaaaaa-bbbb-cccc-dddd-example",
"aud": "tsf",
"exp": 1500013000,
"iat": 1500009400,
"auth_time": 1500009400,
"nonce": "x-03_si1h4t",
"acr": "1",
"azp": "tsf",
"given_name": "Anaya",
"email": "anaya@example.com"
}
```

最后，将上面的 JSON 对象使用 Base64URL 算法转成字符串，即为 JWT 令牌中的 Payload 部分。

Signature

Signature 部分是对前两部分的签名，防止数据篡改。

生成 JWT 令牌（token）需要私钥，点此 生成与验证的公钥与私钥。使用 Header 里面指定的签名算法（默认是 HMAC SHA256），按照下面的公式产生签名。

```
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
secret)
```

算出签名以后，把 Header、Payload、Signature 三个部分拼成一个字符串，每个部分之间用"点"（`.`）分隔，即为JWT 令牌（token），最后返回客户端。

使用 Java 生成 JWT 令牌（Token）

1. 添加 JWT 的 pom 依赖。 微服务网关使用 jose4j 来实现。在 pom 文件中添加以下依赖：

```
<!-- jwt -->
<dependency>
<groupId>org.bitbucket.b_c</groupId>
<artifactId>jose4j</artifactId>
<version>0.6.5</version>
</dependency>
```

2. 编写生成令牌的 Java 代码。

```
// 下面省略了无关代码
import org.jose4j.json.JsonUtil;
import org.jose4j.jwa.AlgorithmConstraints;
import org.jose4j.jwa.AlgorithmConstraints.ConstraintType;
import org.jose4j.jwk.PublicJsonWebKey;
import org.jose4j.jwk.RsaJsonWebKey;
import org.jose4j.jws.JsonWebSignature;
import org.jose4j.jwt.JwtClaims;
import org.jose4j.jwt.MalformedClaimException;
import org.jose4j.jwt.consumer.InvalidJwtException;
import org.jose4j.jwt.consumer.JwtConsumer;
import org.jose4j.jwt.consumer.JwtConsumerBuilder;
import org.jose4j.lang.JoseException;
public static void main(String[] args) throws JoseException, MalformedClaimException {
// 通过私钥对生成jwk
RsaJsonWebKey jwk = new RsaJsonWebKey(JsonUtil.parseJson("{"
+ " \"kty\": \"RSA\","
+ " \"alg\": \"RS256\","
+ " \"n\": \"mrX5ROEw4kCYDXR94FQsm33gr5o5dQXvuOoe-eG_yvdNW83MMt9GgG_eBBq_1b7HgyP9lo15BfKX3GH1igCjEKoXJxcHC5xox4xC0tvbBNCDwG_987ZsqlgCb4f7X66DCcHh17AyAHYa8JhO2kXvtD1OIQxSajSgmk1C1sxI5kqJXfvJwRLcCEK87P6Bs9YNLnnJeouSkYce04AhspmyQKKax4GllbMjcrUUMRoqpCBMklh5Pgl9sOGRLo-6uzzowtI_SyF03YsE2ejh9m-YWqTYsx7PIg6SdWrNRIprKtjnhc9nk-QBzWbOTFH3bpMoXl0T9KPndaLpi1pXtaej9Q\","
+ " \"e\": \"AQAB\","
+ " \"d\": \"lL9vqdUl7fMS_qTJPf1QYjPV6qBKrAQIJ28aV0DA6YF6pFCrCyJ3I5frC2E4nmbuZl0dPTpKaPiFIAQjUwsnvSb8Wb4fLP-2El3-BcQSwX9FnalPrpnvwpwZw2gnvSgJn0EFRh6HBMCJSFf4QI7LWC01SDsTpj9xRsoQAHurf5YZ8YRpi_-XEWBaL-4R_RxpoeAnSgSbJkcGoJNcxqwWbCun37KYBS_71sd155VWycMd-uHtTRnW6SenVG49pexXq-tIQxOwmatpTj0XF7hhshKgdTF1xXPbMSX6XOvxiy929jPMercBL_-OUu0PPUZTTVp0gNziGdevzufHkEfHxQ\","
+ " \"p\": \"5dOs8Q0SHIuCq-gAOx2c2JaqXzPRsmmZ7nXx1P1jbddDIenVPA6q5zUVqXIRRQgMA7AdD9x7anJ2_kaKSFoE2D8peuObvmjrbmJeYE4F4138pNESOHlBmhUH93Oo4i5TvmNZ5hxu3CmonGKMafeDoMpopN15yeDGkTVFKoOfuys\","
+ " \"q\": \"rFRhmJIIj3o__CbjVOlUgiVzrk-ZgK9jGXHhyt6LELQae1nUiZNZuwZeHwgzTonsTMJ2JHnmCuDpwuhjf_kha5KHeuczE7gmnlaGd3s6kaKDyB9bXbMTs122SnNiB-IJcwm4wRNWI-irSh8PyHSQVnjvKkQCCEPi4i0Ky1KgDV8\","
+ " \"dp\": \"qBUJNDn09v9pD8Ra9uEPZq-55mqFgFAPDgEgXj76yshWBqV3F7c6cmG2d_g-fRgHgWL5vjHn6M_SCuEYHRYI2QZIleGEc9tT46T5IME7OS_xp7Bn_PlhawjajLT_3Hs5L9KFWu-MfGPTNpw0SQOGNsARjBGWEnjbgDNPZGpjFYU\","
+ " \"dq\": \"F_0rFNUHUgm_jHdRYAmXFQLnppU4FhzUG7-podb23t1jblZj6r7TV-CcC4_VrJIwjcLoNU2uw0bp45L7_t2MVHAyYd57Urxoy9PZphpGXe2UkLAkxNdf37Ek5hpHxDgqXFQ3HtF1RUxnQ8stJEdtrEvrZyPOcJ4aoEeK4CDhXNs\","
+ " \"qi\": \"QdwkKs9n6jswVsbYKprvpNr2Mbg-RBPp5xx1p-ypVJnFOd_lnCA6P3gRJ-pe6tSCIB6AYViEhZpzKMSu47f27_38VHkH9qOOL38ZVeVFo8Yt8lkBwMEKQdDOghfF74L5Fczo9bH7QX679dC847cDEa1oaV2Cdv6XcSKGywwvq3Y\""
+ "}"));

// 创建Claims，放在JWT payload位置
// Create the Claims, which will be the content of the JWT
```

```
JwtClaims claims = new JwtClaims();
claims.setIssuer("Issuer"); // who creates the token and signs it
claims.setAudience("Audience"); // to whom the token is intended to be sent
claims.setExpirationTimeMinutesInTheFuture(10); // time when the token will expire (10 minutes from now)
claims.setGeneratedJwtId(); // a unique identifier for the token
claims.setIssuedAtToNow(); // when the token was issued/created (now)
claims.setNotBeforeMinutesInThePast(2); // time before which the token is not yet valid (2 minutes ago)
claims.setSubject("subject"); // the subject/principal is whom the token is about
claims.setClaim("email","mail@example.com"); // additional claims/attributes about the subject can be added
List<String> groups = Arrays.asList("group-one", "other-group", "group-three");
claims.setStringListClaim("groups", groups); // multi-valued claims work too and will end up as a JSON array

JsonWebSignature jws = new JsonWebSignature();

// The payload of the JWS is JSON content of the JWT Claims
jws.setPayload(claims.toJson());

// The JWT is signed using the private key
jws.setKey(jwk.getPrivateKey());

// Set the Key ID (kid) header because it's just the polite thing to do.
// We only have one key in this example but a using a Key ID helps
// facilitate a smooth key rollover process
jws.setKeyIdHeaderValue("kid");

// Set the signature algorithm on the JWT/JWS that will integrity protect the claims
jws.setAlgorithmHeaderValue(jwk.getAlgorithm());

// 生成JWT 令牌
// Sign the JWS and produce the compact serialization or the complete JWT/JWS
// representation, which is a string consisting of three dot ('.') separated
// base64url-encoded parts in the form Header.Payload.Signature
// If you wanted to encrypt it, you can simply set this jwt as the payload
// of a JsonWebEncryption object and set the cty (Content Type) header to "jwt".
String jwt = jws.getCompactSerialization();

long expirationTime = claims.getExpirationTime().getValueInMillis();
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss z", Locale.SIMPLIFIED_CHINESE);
String msg = String.format("JWT expired at (%d -> %s)", expirationTime, dateFormat.format(expirationTime));

// 打印JWT 令牌过期时间
System.out.println(msg);
// 打印JWT 令牌
System.out.println(jwt);
}
```

JWT 插件配置步骤

1. 新建插件

微服务网关已经对外提供了 JWT 认证功能，用户可在【TSF 控制台】>【微服务网关】>【插件管理】页面创建 JWT 类型插件。

- 校验参数值：指用户存放 JWT 令牌（Token）参数名称，示例中参数名为 `token`。

- 校验参数携带位置：指用户存放 JWT 令牌（Token）的位置，目前支持 Query 和 Header 两种携带方式，示例中为 Query 方式。
- 公钥对 kid：密钥 ID（“kid”），用来标识此密钥。
- 公钥对 JSON 串：公钥对，示例中值为：

```
{
"kty": "RSA",
"alg": "RS256",
"n": "mrX5ROEw4kCYDXR94FQsm33gr5o5dQXvuOoe-eG_yvdNW83MMt9GgG_eBBq_1b7HgyP9lo15BfKX3GH1igCjEKoXJxc
HC5xox4xC0tvbBNCDwG_987ZsqlgCb4f7X66DCcHh17AyAHYa8JhO2kXvtD1OIQxSajSgmk1C1sxI5kqJXfvJwRLcCEK87P6Bs9
YNLnnJeouSkYce04AhspmyQKKax4GllbMjcrUUMRoqpCBMklh5Pgl9sOGRLo-6uzzowtI_SyF03YsE2ejh9m-YWqTYsx7PIg6Sd
WrNRIprKtjnhc9nk-QBzWbOTFH3bpMoXl0T9KPndaLpi1pXtaej9Q",
"e": "AQAB"
}
```

- claim 参数映射关系 JSON：指 JWT 令牌（Token）中 claim 携带的数据是否需要透传（可配置多组），示例中值为：

```
[{
"parameterName":"email",
"mappingParameterName":"new_email",
"location":"header"
}]
```

该示例表示，将 claim 中参数为 `email` 的值，通过放入 HTTP header 透传给下游，并重新命名为 `new_email`。

- `parameterName` 属性表示 claim 数据中待透传参数的名称。
- `mappingParameterName` 属性表示需要转换的参数名称。
- `location` 属性表示透传此值的位置。

2. 插件绑定对象

微服务网关插件通过绑定分组或 API 来生效。

配置OAuth插件

最近更新时间: 2025-02-18 16:02:00

操作场景

该文档指导您通过 TSF 控制台，创建微服务网关标签（OAuth）插件。

微服务网关 OAuth 插件提供了简单的第三方鉴权的能力。外部请求请求到网关，网关向第三方服务器发送请求对参数进行校验，校验结果返回给第三方。

OAuth 插件的时序图如下：

外部请求请求到业务 API，通过微服务网关时网关访问认证服务器，认证服务器将结果返回给网关并返回相应给外部请求。

操作步骤

新建插件

1. 登录 TSF 控制台，在左侧导航栏单击组件中心下的【微服务网关】>【插件管理】。
2. 在插件管理页面，单击【新建插件】，填写基本信息。

- 插件类型：选择 OAuth。
- 插件名称：必填。
- 插件描述：选填。

3. 设置请求参数。

- 请求参数名称：请求参数名称，即上图中的 A 字段。
- 请求参数携带位置：填写 A 参数的位置，网关将在这个位置将 A 参数传递给认证服务器。
- 验证请求参数地址：填写认证服务器地址。以 http://imgcache.finance.cloud.tencent.com:80 或者 http://imgcache.finance.cloud.tencent.com:80 开头。
- 请求方法：访问认证服务器的方法，GET 或 POST。
- 超时时间：网关访问认证服务器的超时时间，0秒 < 超时时间≤30秒。

4. 设置返回值。 返回值指的是认证服务器认证成功后，认证服务器可以将某些业务字段返回给网关，网关可以将这些字段通过 Payload 的形式返回给业务 API，方便后续业务依据这些字段进行业务逻辑处理。

- Payload 参数名称：选填。
- Payload 参数携带位置：填写 Header 或者 Query。

5. 单击【完成】，跳转至插件管理列表。

使用说明

- 认证服务器返回认证结果将在 response body 中返回 result 字段，返回值为 true / false。返回示例如下：

```
{
"result": true,
"payload": "this is payload"
}
```

- 认证失败后，网关将返回错误码401，错误信息为 “AuthCheckFailed”。
- 创建好插件后，需要将插件绑定在分组或者 API 上，才能生效。
- OAuth 插件与 JWT 插件的使用区别如下，用户可以酌情选择。

对比项	标鉴权认证实现方式	计算签名	性能
OAuth 插件	在认证服务器上实现	需要请求方计算	请求量较大时，由于 OAuth 插件需要等待远程认证服务器的响应，可能会对性能有一定影响。
JWT 插件	由网关完成	不需要	-

配置Tag插件

最近更新时间: 2025-02-18 16:02:00

操作场景

该文档指导您通过 TSF 控制台，创建和绑定微服务网关 Tag 插件。

标签是 TSF 中传递客户业务参数的形式，用户可以通过标签来实现灵活的基于业务参数的服务治理能力并依据标签过滤调用链。 典型的标签的使用场景是：通过业务参数（如 usertype 等字段）实现针对参数值的特殊路由、限流等。 TSF 中支持两种标签的配置方式：

- 通过微服务网关 Tag 插件将外部请求的请求参数转化为标签。
- 通过在代码或请求头中配置来实现标签。
- Spring Cloud 应用配置标签
- Service Mesh 应用配置标签

操作步骤

创建 Tag 插件

1. 登录 【TSF 控制台】，在左侧导航栏单击组件中心下的【微服务网关】>【插件管理】。
2. 在插件管理页面，单击【新建插件】，填写基本信息。

- 插件类型：选择 Tag。
- 插件名称：必填。
- 插件描述：选填。

3. 配置自定义标签。

- 参数位置：可选 Path、Query、Header 和 Cookie。
- 参数名：填写外部请求进来携带的参数名称。参数名请填写小写。 如果参数在 Path中，可以按照这样的规则填写：/a/{parameter}/b，其中{}中的是参数名称。此时的路径不需要包含微服务名称、命名空间等信息。
- 转换后标签名（选填）：如果不填写，会默认使用参数名作为转换后的标签名称。

> **注意：**
>
> - 通过微服务网关标签和代码中配置的标签总数是16个。
> - 使用1.21.0版本的微服务网关 SDK，可以保证配置后的标签在全链路中进行传递。
> - 标签名最长32字节。

- 设置为 TraceID：是一个非必须功能。当请求中某个参数值具有唯一性时，且希望在部署在 TSF 的后台业务与其他前台、后台系统联动时，可以通过设置 TraceID 进行追踪。

> **注意：**
>
> - 同一类请求只允许一个参数设置为 TraceID。

- 当将某个参数设置为 TraceID 时，TSF 依然将同时生成 TraceID，参数名为 "X-B2-TraceID"，并将两者关联，且同时返回给调用方。
- 用户可以通过调用链查询中的标签检索能力搜索过滤所需要的调用链。

4. 单击【完成】，本步骤创建的插件将显示在插件管理列表中。

绑定插件

创建好 Tag 类型的插件需要绑定到分组或 API 上才会生效。

1. 登录 【TSF 控制台】，在左侧导航栏单击组件中心下的【微服务网关】>【插件管理】。
2. 选择刚刚创建好的插件，单击操作列的【绑定对象】。
3. 在绑定对象中，将插件绑定到某个分组或者某个分组下的 API 上。
4. 单击【提交】。当用户访问对应的微服务网关分组时，插件就会生效。

配置小程序登录插件

最近更新时间: 2025-02-18 16:02:00

操作场景

该文档指导您通过 TSF 控制台，创建和绑定微信小程序登录插件。

微信小程序开发者在开发小程序的过程中，需要考虑用户登录认证相关的业务逻辑，基于此插件，开发者可以方便地实现业务的登录认证服务。

典型的微信小程序登录插件使用场景是：

- 实现微信小程序用户登录：遵守微信小程序官方登录规范，在微服务网关侧完成登录业务，生成用户登录态凭证，并返回给小程序前端处理，同时，将微信登录成功返回的原生登录态数据传递至业务后台处理。
- 实现用户登录态凭证校验：微服务网关对用户登录态凭证进行校验，若校验通过，则网关将请求放行，并携带微信原生登录态数据传递至业务后台，否则，返回失败原因至小程序前台，小程序客户端可以根据异常状态码，进行对应的操作（如因登录态过期失效进行重新登录操作）。

操作步骤

创建微信小程序登录插件

1. 登录 【TSF 控制台】，在左侧导航栏单击组件中心下的【微服务网关】>【插件管理】。
2. 在插件管理页面，单击【新建插件】，填写基本信息。

- 插件类型：选择 WeChatMpLogin。
- 插件名称：必填。
- 插件描述：选填。

3. 填写参数设置。

- APPID：小程序唯一凭证密钥，获取方式同 appid。
- APPSECRET：小程序唯一凭证密钥，获取方式同 appid。

4. 获取登录凭证 Code。 请求 Code 携带位置：微信后台通过 Code 换取用户的登录态信息，包括用户的唯一标识（openid）及本次登录的会话密钥（session_key）。
5. 向小程序返回登录态。

- 自定义登录态参数名：插件对微信登录态以及其他登录信息进行加密的参数名。 返回自定义登录态参数位置：插件对登录态进行加密后的参数携带位置。
- 登录态过期时间：登录态过期时间，以分为单位。

6. 选择登录态校验。

- 前台业务请求自定义登录态参数位置：小程序请求业务服务时，登录态携带位置。
- 向业务后台传输登录态参数位置：网关对登录态进行解密后，将提取的微信登录信息传递至用户后台服务的携带位置。

7. 单击【完成】，完成创建。

将请求参数转 Tag

TSF 提供了一系列将请求参数转 Tag 能力，方便进行细粒度的服务治理能力。

默认 Tag 操作说明

- 平台提供了一系列微信默认参数的灰度发布能力，包含用户名、性别等信息。用户可以勾选这些参数并提供参数的填写位置，方便平台将这些请求参数转为 TSF 中的标签。
- 当某个参数的传入具有请求的唯一性（该参数值不重复），则可以将改参数设置为 TraceID，平台将对该参数与平台生成的 TraceID 进行关联，方便用户进行唯一性定位与检索。
- 仅支持一个参数设置为 `TraceID/` 。

自定义 Tag 操作说明

自定义 Tag 使用请参考 [微服务网关 Tag 插件] 。

绑定插件

创建好 WeChatMpLogin 类型的插件需要绑定到分组或 API 上才会生效。

1. 登录 【TSF 控制台】，在左侧导航栏单击组件中心下的【微服务网关】>【插件管理】。
2. 选择刚刚创建好的插件，单击操作列的【绑定对象】。
3. 在绑定对象中，将插件绑定到某个分组或者某个分组下的 API 上。
4. 单击【提交】。当用户访问对应的微服务网关分组时，插件就会生效。

案例说明

1. 获取 APPID 与 APPSECRET。
2. 根据上述 操作步骤 ，完成微服务网关中微信小程序登录插件的创建与绑定。
3. 编写小程序登录业务。

```
// 登录
wx.login({
success: res => {
// console.log("code is : ", res.code);
wx.request({
url: ${LOGIN_BUSINESS_URL}, //登录业务访问路径
header: {
'content-type': 'application/json',
'js_code': res.code, //携带js_code
},
success: res => {
//本地缓存
wx.setStorageSync('user_session', res.header.user_session);
//let user_session = wx.getStorageSync('user_session') || ''
//console.log('idsession is :', idsession);
wx.showToast({
title: '登录成功'
})
}
}
```

```
)
}
})
```

4. 通过小程序访问业务服务。

```
BusinessRequestFunc: function () {
//获取登录时存储的user_session
let user_session = wx.getStorageSync('user_session') || '';
//console.log("user_session is  : " , user_session);
wx.request({
url: ${BUSINESS_URL}, //业务访问路径,
header: {
'content-type': 'application/json',
'user_session': user_session
},
success: res => {
//console.log("response : ", res.data)
// your business here
}
}
)
//具体接口使用文档参照微信小程序官方使用接口文档
}
```

概述

最近更新时间: 2025-02-18 16:02:00

微服务网关作为后台架构的入口，提供路由转发、API 管理、访问过滤器等作用，是微服务架构中的重要组件。TSF 中的微服务网关基于 Spring Cloud 中的 Zuul 和 Spring Cloud Gateway 实现，提供了符合微服务体系的灵活可自定义的网关功能。

微服务网关的主要功能

- **请求转发** 请求转发是微服务网关的基本功能。TSF 中的微服务网关可以通过页面配置灵活管理需要被转发请求的微服务 API。微服务网关会及时从注册中心感知后端服务节点健康状况，保证在后端服务节点变动情况下请求不中断。
- **API 管理** 微服务网关集中管理了所有需要对外暴露的 API，帮助用户进行 API 的生命周期管理。
- **API 治理** 支持 API 级别的限流、路由等能力，支持用户绑定系统插件或自定义插件。

典型使用场景

- 提供对外暴露 API 的统一入口
- 穿透命名空间隔离，实现 set 化部署
- 协议转换
- 灰度发布入口
- 自定义统一入口处理逻辑

功能优势

支持完整的服务治理能力

TSF 中的微服务网关形态等同一个普通的应用，也需要用户创建应用（选择"微服务网关类型"应用），进行部署、发布。部署成功且注册到注册中心后也可以在服务治理页面看到微服务网关服务。

微服务网关可以通过服务注册发现感知后端微服务状态，当后端微服务某节点下线时，网关流量不会打到该节点上。

微服务网关支持 TSF 提供的完整的服务治理能力，包含服务路由、服务鉴权、服务限流、监控、调用链、全链路灰度发布、应用配置等。

支持灵活定制

微服务网关作为业务层网关，通常需要承担部分相对通用的业务逻辑。由于这部分业务逻辑有很强的业务属性（如请求去重、特殊的鉴权逻辑等），我们提供了微服务网关 SDK，方便用户进行定制化开发。依赖微服务网关 SDK 后，网关的页面功能依然可用。

当用户只需要使用我们页面上提供的微服务网关能力时（如请求转发、监控、密钥对鉴权、第三方鉴权、请求参数转 tag 等），可以直接下载并部署微服务网关 Demo 程序包，不需要额外改动。

贴近开源

提供基于 Zuul 以及 Spring Cloud Gateway 两种选型的微服务网关。

支持灵活扩缩容

由于微服务网关本质也是一个微服务，用户可以通过手动或配置弹性伸缩规则的方式对网关进行横向扩展。

微服务网关开源对比优势

相比开源的微服务网关，TSF 中的微服务网关有以下使用优势：

对比项	TSF 微服务网关	开源微服务网关
API 管控	支持页面化 API 管理能力。自动获取微服务 API 无需手动录入。	API 需要手动录入到配置文件中，无法灵活调整。
可观测性	强，支持查看API各个维度监控指标，快速定位问题。	弱，需要配合其他监控组件。
服务与 API 治理	支持 API 限流、熔断、灰度等一系列服务与 API 治理能力。与 TSF 中提供的治理能力全面兼容，可实现全链路灰度发布等高级能力。	弱，需要用户通过插件自行开发。
插件	多。提供鉴权、小程序开发等一系列系统插件。同时支持自定义开发插件。	仅支持自定义开发。
使用场景	支持全链路灰度发布、单元化部署、set 化部署、高可用部署等多种复杂场景。	仅支持有限复杂场景。

网关监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在 TSF 控制台查看网关监控信息。

操作步骤

1. 登录 TSF 控制台，在左侧菜单栏选择【微服务网关】>【网关监控】。
2. 在网关监控页面，选择网关微服务和部署组之后，可以查看网关监控信息。

- 针对外部 API 调用的监控可以在 /external 分组中查询。

- 针对单元化场景下跨命名空间的调用监控可以在 crossunit 分组中查询。
- **近24小时使用量/总量**： 调用数：近24小时内服务调用的总次数。 错误数：近24小时内服务调用发生错误的次数。 成功率：近24小时内服务服务调用成功率。
- **分组及 API 使用统计**：在左侧 API 列表中选择 API 后，可以在右侧查看 API 的监控数据。
- **每日使用统计**：每天24点更新。 使用明细：选择时间范围后可查看该接口的调用量，平均错误率和平均响应耗时数据。

部署微服务网关

最近更新时间: 2025-02-18 16:02:00

操作场景

部署微服务网关相当于部署一个微服务。该任务指导您通过 TSF 控制台传程序包，并部署微服务网关。

操作步骤

新建微服务网关应用

1. 登录 【TSF 控制台】，在左侧导航栏单击【应用管理】。
2. 在应用管理页，单击【新建应用】。
3. 填写应用名、部署方式和应用类型 。 应用类型选择**微服务网关应用。**

4. 单击【提交】，完成创建。

部署微服务网关应用

1. 登录 【TSF 控制台】 ，在左侧导航栏单击【应用管理】。
2. 单击新建好的微服务网关应用 ID，进入应用详情页。
3. 在应用详情页顶部，单击【程序包管理】>【上传程序包】 微服务网关 Demo 程序包 GitHub 地址，相关操作请参考 【微服务网关开发指南】 。

> **注意：**
>
> - 微服务网关兼具灵活性和便捷性。当用户仅需要使用请求转发、密钥对鉴权、服务治理等 TSF 已经提供的产品能力时，您仅需要使用 TSF-demo 中的 msgw - demo 程序包即可，可以选择其中的 msgw - zuul 以及 msgw - scg 来分别部署基于 Zuul 和基于 Spring Cloud Gateway 的微服务网关。
> - 当您希望定制化微服务网关的转发逻辑，您可以选择自行开发微服务网关，并依赖 TSF 的网关 SDK 来进行开发。
> - 直接使用开源的微服务网关（Demo 中的 opensource 版本）会导致 TSF 产品页面中的网关功能不可用。不建议您使用开源版本的微服务网关。

4. 新建部署组。 其中针对微服务网关所部署的命名空间，当微服务网关部署在全局命名空间中时，该微服务网关可以将请求路由到全部命名空间的微服务上；当微服务网关部署在非全局命名空间时，该微服务网关只能转发当前命名空间下的请求。即：**非全局命名空间中的微服务网关，无法穿透命名空间的逻辑隔离。**
5. 部署应用。 部署成功后，可以在服务列表中找到微服务网关服务。

重定向配置请求路径

最近更新时间: 2025-02-18 16:02:00

操作场景

在很多场景下，为了不将内部接口信息过分暴露或为了保证上下游系统接口的一致性，需要对微服务对外暴露的接口配置新的重定向路径，如内部接口写为 `/group1/prepub/provider/echo` ，我们可以配置重定向接口为 `/echo` ，这样所有通过对应网关访问 `/echo` 这个接口的调用都会被指向到后端 `/group1/prepub/provider/echo` 这个接口上。

操作步骤

新建重定向配置

1. 登录 【TSF 控制台】。
2. 在左侧导航栏找到组件中心，单击【微服务网关】>【网关管理】>【重定向配置】。
3. 在重定向配置页面，在上方选择需要配置重定向路径的网关部署组。
4. 单击【新建重定向路径】，填写以下内容：

- 请求路径：请求路径指的是外部请求访问到网关的路径。填写时要以 / 开头。
- 原路径：指的是微服务实际对外暴露的路径，微服务开发者代码中定义的路径。
- 匹配顺序：当对一个微服务网关上配置多个重定向规则的时候，可能会出现多个规则最终落在同一个实际的微服务API上的情况，此时可以通过匹配顺序指定规则之间的前后匹配顺序。匹配顺序值为1-2000之间正整数，1为最高优先级。匹配顺序允许相同，匹配顺序相同时按照规则创建时间顺序匹配。
- 类别：在用户填写原路径配置时，可以通过类别筛选重定向规则匹配一个具体的API还是多个通配API。
- 屏蔽原路径：屏蔽原有路径的访问，屏蔽后，微服务API只能通过配置后的请求路径访问。

说明

1. 配置重定向规则前，请先确认原路径 API 所在的分组已经与页面上方选中的微服务网关部署组绑定。
2. 路径匹配规则支持正则表达式，详情如下：

正则表达式	说明
.	匹配除换行符以外的任意字符
?	重复0次或1次
+	重复1次或更多次
*	重复0次或更多次
\d	匹配数字
^	匹配字符串的开始
$	匹配字符串的介绍

正则表达式	说明
{n}	重复 n 次
{n,}	重复 n 次或更多次
[c]	匹配单个字符 c
[a-z]	匹配 a - z 小写字母的任意一个

> **注意：**
>
> 小括号()之间匹配的内容，可以在后面通过$1来引用，$2表示的是前面第二个()里的内容。

使用场景举例

1. 当前 TSF 微服务网关提供的默认对外访问路径为 `域名/分组名/命名空间/微服务名称/API路径` ，可以通过重定向配置支持将较长的路径映射为短路径：

- 请求路径填写为： `/provider/(.*)`
- 原路径填写为： `/groupone/defaultnamespace/provider/$1`

此时，通过网关直接请求/provider/echo等请求会直接被转发到 `/groupone/defaultnamespace/provider/echo`

2. 可以通过重定向配置为某一个 API 配置路径别名，如：

- 请求路径填写为： `/tsf/config`
- 原路径填写为： `/qcloudadmin/middleware/tsf/dispatch/config`

复制重定向规则

复制重定向规则支持将某一个网关的下的规则快速复制到另一个微服务网关部署组下面。

操作步骤：

1. 登录 【TSF 控制台】。
2. 在左侧导航栏找到组件中心，单击【微服务网关】>【网关管理】>【重定向配置】。
3. 勾选已经配置好的规则，单击列表上方的【复制到】，在弹窗中选择目标网关的部署组。
4. 单击【确定】，完成复制。

调用链查询（单元化）

最近更新时间: 2025-02-18 16:02:00

调用链查询

操作场景

调用链查询用来查询和定位具体某一次调用的情况。使用者可以通过具体的服务、接口定位、IP 等查询具体的调用过程，包括调用过程所需要的时间和运行情况。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中选择【[调用链查询]】。
3. 在调用链查询中，设置查询条件，单击【查询】。

- **时间范围**：支持特定和自定义时间范围选择。特定时间范围包括：5分钟前、10分钟前和30分钟前。
- **调用服务/调用接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **被调服务/被调接口**：单击下拉框，在下拉框中选择服务，可以输入关键字进行搜索。
- **仅查询出错的调用链**：勾选后，可以查询系统中的出错业务。
- **耗时大于**：设置耗时的阈值，可以查询系统中的慢业务。
- **客户端 IP**：客户端 IP 地址。
- **服务端 IP**：服务端 IP 地址。
- **标签**：用户在代码中设置的标签，参考 [参数传递] 中设置调用链 Tag。

4. 根据查询结果，可以单击【TraceID】进入具体慢业务或出错业务，查看调用链详情。

调用链详情

操作场景

您可以根据 TraceID 查询调用链的详细信息。调用链详情是为了定位在分布式链路调用过程中每个环节的耗时和异常（不包含本地方法调用情况，本地方法调用建议使用业务 log 的方式记录）。 通过调用链通常为了解决以下问题：

- 定位耗时较长的服务
- 不合理的调用逻辑（如一次请求多次调用某服务，建议改为批量调用接口）

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中选择【调用链查询】。
3. 在 TraceID 查询中，在搜索框中输入目标 TraceID，单击【查询】。 如果搜索有结果，页面将显示调用链每个环节的耗时和状态。

4. 将鼠标移动到每个环节的时间条上并单击，会弹出 Span 的详细信息。Span 包含三部分信息：

- 基本信息：显示 Span 名、Span ID、状态和阶段耗时信息。

- 标签：显示系统和业务自定的标签。

- 自定义 Metadata：显示用户在代码中设置的自定义元数据信息。参考开发手册中 [调用链] ，了解如何设置 Metadata。

调用链与业务日志联动

目前要实现调用链与业务日志联动的前提条件是部署组关联的日志配置项的日志格式支持日志与调用链联动，并且必须遵守日志格式规范。

- Spring Boot 格式默认可以支持日志调用链联动。
- 自定义 Logback、自定义 Log4j、自定义 Log4j2 和单行/多行文本格式日志需要日志 pattern 中添加 %trace 才可以支持日志和调用链联动。
- Nginx Access 和无解析规则格式日志无法支持日志和调用链联动。

在调用链详情页内，单击日志的 icon，可以查看与这条调用链相关的业务日志。

网关监控（单元化）

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在TSF控制台查看网关监控信息。

操作步骤

1. 登录 【TSF 控制台】，在左侧菜单栏选择【微服务网关】>【网关监控】。
2. 在网关监控页面，选择网关微服务和部署组之后，可以查看网关监控信息。

- 近24小时使用量/总量

调用数：近24小时内服务调用的总次数

错误数：近24小时内服务调用发生错误的次数

成功率：近24小时内服务服务调用成功率

- 分组及API统计：在左侧API列表中选择API后，可以在右侧查看API的监控数据

每日使用统计：每天24点更新

单元化使用统计：选择时间范围后，可以查看该API在不同命名空间下的调用情况，包含调用量、平均错误率和平均响应耗时等。

平台管理
概述

最近更新时间: 2025-02-18 16:02:00

为满足企业客户不同成员使用 TSF 平台的接口操作权限、资源操作权限的需求，当前微服务平台支持主账号为子账号灵活分配不同角色、不同数据集的权限。

操作场景

当前 TSF 可以支持以下几种场景的使用：

- 为子账号或协作者配置全部资源的全读写策略。
- 为子账号或协作者配置全部资源的部分操作权限，如可以为部分用户配置应用、微服务、配置的全读写策略，以及集群、命名空间的只读策略。
- 为子账号和协作者配置某些资源（一个或多个）的读或写权限，如可以为不同的子账号配置不同命名空间、不同应用的不可见、只读、全读写权限。

操作步骤

在子账号和协作者使用 TSF 平台之前，需要进行以下几个步骤：

1. 按照 【访问管理】 中，"子账号和协作者使用 TSF"部分文档进行操作，这一步保证了用户使用 TSF 时，TSF 访问 VPC、 TKE 等云资源的服务角色权限。
2. 按照 【角色管理】 、【数据集管理】 、【授权管理】 三个文档进行操作，配置子账号和协作者使用 TSF 平台时操作和数据集权限。

预设策略

TSF 提供了 QcloudTSFFullAccess 和 QcloudTSFReadOnlyAccess 两条预设策略。

策略名称	作用
QcloudTSFReadOnlyAccess	只读策略，包含 TSF 全部集群、命名空间、应用、服务等资源的全部只读操作
QcloudTSFFullAccess	全读写策略，包含 TSF 全部集群、命名空间、应用、服务等所有资源的全部操作

预设策略只限制用户操作，没有对数据进行过滤。当用户对数据过滤没有需求时，仅需绑定预设策略，无需进行上述使用步骤中的步骤2（步骤1还需要进行）。

> **注意：**
>
> - 截止至2019年10月10日，TSF 现网所有子账号和协作者都默认绑定了 QcloudTSFFullAccess 权限，当用户需要精细化的配置操作和数据集权限时，需要解绑全读写策略 QcloudTSFFullAccess 并按照上述步骤2进行操作。
> - 当为用户绑定了多条策略时，多条策略以白名单的形式取并集生效。
> - QcloudTSFReadOnlyAccess 相当于配置一条只读角色操作权限和全数据集权限。当主账号想要设置某子账号对某些资源不可见时，不要绑定 QcloudTSFReadOnlyAccess 策略。

角色管理

最近更新时间: 2025-02-18 16:02:00

操作场景

角色管理是配置子账号与协作者使用 TSF 的操作权限。当前 TSF 平台采取白名单机制，如子账号和协作者没有被授权任何角色，则无法使用 TSF 平台。

操作步骤

新建角色

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【角色管理】。
3. 在角色管理页面，单击页面左上角的【新建角色】。
4. 在新建角色页面，填写角色名称并选择角色权限。
5. 单击【提交】，完成角色创建。

> **注意：**
>
> - 角色权限中的通用权限包含的 TSF 访问概览页面、监控等页面的基本操作，建议对所有用户授权。
> - 日志告警、服务告警、部署组告警等告警策略的配置在云监控页面配置，不受 TSF 权限管控。
> - 镜像仓库列表不受 TSF 权限管控。

编辑角色

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【角色管理】。
3. 单击选择目标角色。
4. 单击右上角的【编辑】即可对角色进行编辑。

删除角色

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【角色管理】。
3. 将鼠标悬停在目标角色上，单击角色名称右侧的【删除】即可删除该角色。

> **注意：**
>
> 当已经给某用户绑定角色和策略组后，删除角色会导致策略不可用，请谨慎删除。

数据集管理

最近更新时间: 2025-02-18 16:02:00

操作场景

用户可以通过数据集管理配置不同的子账号和协作者使用不同集群、命名空间、应用的数据集权限。数据集是白名单机制，未配置数据集会导致无法看到 TSF 上任何资源。

操作步骤

新建数据集

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【数据集管理】。
3. 在数据集管理页面，单击左上角的【新建数据集】。
4. 填写数据集名称和备注，勾选资源范围。

5. 单击【完成】，完成数据集创建。

- 当前仅支持集群、命名空间、应用三种资源的数据集权限。
- 全部集群、命名空间、应用代表的是当前全部数据以及未来可能新增的集群、命名空间、应用。如果仅将指定集群权限授予某个用户，则新增集群对用户不可见。
- 当用户对集群、命名空间、应用中某一类不勾选时，该类别资源对被授权的用户不可用。

编辑数据集

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【数据集管理】。
3. 单击目标数据集进入编辑页面。
4. 单击基本信息卡片右侧的【编辑】修改数据集名称，单击资源范围卡片右侧的【编辑】修改数据集权限。

删除数据集

> **注意：**
>
> 当某个数据集已经对用户授权后，删除数据集将导致授权策略对用户不生效。

1. 登录 TSF 控制台。
2. 在左侧导航栏，单击【管理中心】下的【数据集管理】。
3. 选择目标数据集，单击操作列的【删除】删除该数据集。

授权管理

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台配置并生成策略，并通过访问管理（CAM）将该策略管理用户/用户组。

操作步骤

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【管理中心】下的【角色管理】。

3. 单击【策略配置】，进入策略生成器页面。
4. 配置策略名称和权限。

- 策略名称：支持英文字母、数字、以及符号"+ = , . @ _ -"，限制128个字符以内。
- 权限配置：选择对应角色与资源组。示例：当用户希望对集群1配置全读写权限，对集群2配置只读策略时，可以将包含全读写集群的角色与包含集群1的数据集关联作为权限配置的第一条数据，将包含集群只读的角色与包含集群2的数据集关联作为第二条数据。

5. 单击【前往 CAM 进行授权】，并在访问管理产品页面单击【创建策略】。

> **注意：**
>
> 为避免策略不生效，建议不要在此处修改策略脚本。

6. 将生成的策略 【关联用户/用户组】 。
7. 单击【确定】，完成授权管理。

标签管理

最近更新时间: 2025-02-18 16:02:00

使用场景

标签是云平台提供的用于标识云上资源的标记，是一个键-值对（Key-Value）。

您可以根据各种维度（例如业务、用途、负责人等）使用标签对TSF中的集群、命名空间和应用资源进行分类管理，通过标签非常方便地筛选过滤出对应的资源。标签键值对云平台没有任何语义意义，会严格按字符串进行解析匹配。

使用限制

标签数量限制如下：

- 资源维度：一个资源最多 50 个不同的 key。
- 标签维度：单个用户最多 1000 个不同的 key。
- 一个 key 最多有 1000 个 value。

其他标签命名相关限制请参考 【使用限制】。

使用案例

以下案例以集群为例进行说明，命名空间和应用的标签使用方法类似，可以选择在创建资源时设置标签或者在编辑标签中设置标签。

案例描述

某公司在云平台上拥有5个集群，这5个集群的使用部门、业务范围以及负责人的信息如下：

集群/ID	使用部门	业务范围	负责人
Cluster-abcdefg1	电商	营销活动	张三
Cluster-abcdefg2	电商	营销活动	王五
Cluster-abcdefg3	游戏	游戏 A	李四
Cluster-abcdefg4	游戏	游戏 B	张三
Cluster-abcdefg5	娱乐	前端制作	王五

以 Cluster-abcdefg1 为例，我们可以给该集群添加以下三组标签 ：

标签键	标签值
dept	ecommerce
business	mkt

标签键	标签值
owner	zhangsan

类似的，其他实例也可以根据其使用部门、业务范围和负责人的不同设置其对应的标签。

在 TSF 控制台上设置标签

以上文场景为例，当您完成标签键和标签值的设计后，可以登录TSF控制台进行标签的设置。

1. 登录 【TSF 控制台】。
2. 在左侧导航栏选择【集群】，单击【新建】，在新建集群页面添加标签。

> **注意：**
>
> - 若现有标签不符合要求，可以单击【标签管理】创建标签。
> - 在集群列表，选择目标实例，单击操作栏的【更多】>【编辑标签】。

3. 在弹出的窗口中，可以添加、删除或者修改标签。

通过标签键筛选实例

1. 在搜索框中，选择【标签】。
2. 在【标签】后输入标签或者选择标签，单击进行搜索。 您可以同时根据多个标签键进行筛选。例如，输入 `标签键：key1|key2` 可筛选出绑定了标签键 `key1` 或 `key2` 的实例。

最佳实践
灰度发布实践

最近更新时间: 2025-02-18 16:02:00

当用户需要上线新的功能时，希望使用灰度发布的手段在小范围内进行新版本发布测试。 TSF 支持通过部署组和服务路由来实现灰度发布。

场景说明

consumer 调用 provider 时，provider 有两个版本 v1 和 v2-beta，其中 v2-beta 是测试版本。consumer 首先将90%的请求分配给 provider v1 版本，剩下的10%分配给 v2-beta 版本。如果发现 v2-beta 版本运行正常，则增加该版本的流量比例。

前提条件

- 已经下载基于 TSF Spring Cloud SDK 或者 Mesh 编写的代码程序包。
- 已经创建了集群和命名空间，集群中导入2个云服务器。
- 已经创建了 consumer 和 provider 应用（虚拟机部署方式），同时已经创建并部署了 consumer 部署组。

操作步骤

一、provider 创建2个部署组

1. 登录 TSF 控制台。
2. 在左侧导航栏中，单击【应用管理】，进入应用列表页，选择 provider 应用，进入应用详情页。
3. 单击顶部【程序包管理】，上传 v1 和 v2-beta 的程序包。
4. 单击顶部【部署组】，在部署组列表页面创建两个部署组：

- 部署组 provider-group-1，添加1个实例，实例规格为1核1GB。
- 部署组 provider-group-2，添加1个实例，实例规格为1核1GB。

5. provider-group-1 部署程序包 v1。
6. provider-group-2 部署程序包 v2-beta。

二、配置服务路由规则及初始路由权重 90:10

1. 在左侧导航栏中，单击【服务治理】，进入服务列表页，单击 provider 服务，进入服务详情页。
2. 单击顶部【服务路由】， 新建路由规则，流量来源为 主调服务名等于consumer-demo ，流量目的地如下图所示，设置 provider-group-1的权重为90，provider-group-2 的权重为10。
3. 在服务路由规则列表中，单击生效状态列的图标启动该规则，稍等几分钟，刷新页面观察列表下方流量比例变化情况，如果发现 provider-group-1 和 provider-group-2 的流量比例接近 90:10，说明路由规则生效。

三、修改路由权重为 50:50

当 v2-beta 版本服务已经正常运行一段时间后，逐步增加 v2-beta 版本的流量比例，并减少 v1 版本的流量比例。

1. 在服务路由规则列表中，单击【编辑】。
2. 修改流量目的地中 provider-group-1 和 provider-group-2 的流量比例分别为50和50。
3. 稍等几分钟，刷新页面观察列表下方流量比例变化情况，如果 provider-group-1 和 provider-group-2 的流量比例接近 50:50，说明路由规则生效。

四、修改路由权重为 10:90

逐步增加 v2-beta 版本的流量比例，并减少 v1 版本的流量比例。操作方法参考 步骤三。

五、修改路由权重为 0:100

1. 修改路由规则，将 provider-group-1 的权重设置为100。
2. 稍等几分钟，刷新页面观察列表下方流量比例变化情况，如果只有 provider-group-1 有流量，说明路由规则生效。

3. 此时可以停止部署组 provider-group-2 。

使用说明

在真实使用场景中，有以下几点需要考虑：

- 评估单个 provider 服务实例可以处理多少请求，并根据流量分配来规划部署组的实例数量。
- 如果流量比例变化后，可能需要调整部署组的实例数量来满足新的流量比例。
- 可以通过设置**弹性伸缩规则**来支持动态调整实例数量。

就近路由和跨可用区容灾

最近更新时间: 2025-02-18 16:02:00

TSF 支持业务同城可用区就近路由和跨可用区容灾。该功能在1.12.0版本 发布之后生效，使用前提条件：

- 如果是 Spring Cloud 应用，必须使用1.12.0之后版本的 SDK。
- 如果是虚拟机应用，部署实例必须是1.12.0发布之后导入集群的云主机。
- 如果是容器应用，部署实例必须是1.12.0发布之后创建或者重新部署。

功能概述

命名空间、集群有如下特点：不同集群可关联同一命名空间，服务支持跨集群访问 。

以 consumer 服务调用 provider 服务为例说明服务跨可用区访问的原理。在下方示意图中有两个集群，分别位于广州一区和广州二区。集群1分别部署了 consumer 和 provider 微服务，集群2部署了 provider 微服务。两个可用区的服务都会注册到同一个注册中心集群。

- 当**开启就近路由**时，广州一区的 consumer 会优先调用同一可用区的 provider。
- 当**开启就近路由**时，当广州一区的 provider 不可用时，consumer 会跨可用区调用广州二区的 provider。
- 当**关闭就近路由**时，广州一区的 consumer 会从两个可用区 provider 中随机选择实例进行调用。

就近路由原理

微服务的路由模块会将可用区作为一种系统标签，当**开启就近路由**时，服务消费者会将请求流量**全部**路由到相同可用区标签的服务提供者，只有当同一可用区服务提供者不可用时，才会进行跨可用区的服务调用；当关闭就近路由时，服务消费者会随机选择不同可用区的服务提供者实例进行调用。

最佳实践

跨可用区容灾场景下的就近路由

场景：当**开启就近路由**时，广州一区的 consumer 会优先调用同一可用区的 provider ，只有当广州一区的 provider 不可用时，consumer 才会跨可用区调用广州二区的 provider。

步骤一：准备资源

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏中，单击【集群】，进入集群列表页，单击【新建集群】。
3. 新建虚拟机集群 cluster1，【所在可用区】选择广州一区，设置其他属性。
4. 新建虚拟机集群 cluster2，【所在可用区】选择广州二区，设置其他属性。
5. 集群 cluster1 和集群 cluster2 分别导入所在可用区的云服务器。
6. 在集群 cluster1 的集群详情页 >【命名空间】标签页，单击【新建】，新建命名空间 dev-ns。
7. 在集群 cluster2 的集群详情页 >【命名空间】标签页，单击【新建】，新建命名空间 dev-ns。
8. 确认命名空间 dev-ns 的就近路由开关默认是开启的。

步骤二：部署应用

1. 在左侧导航栏中，单击【应用管理】，进入应用管理页，单击【新建应用】。
2. 新建应用 consumer-app 和 provider-app。
3. 在应用详情页 >【程序包管理】页面上传 Demo 程序包 。
4. 进入 conumser-app 应用详情页 >【部署组】标签页。 新建部署组 consumer-group， 属于集群 cluster1，命名空间 dev-ns。添加实例，部署 consumer-demo 程序包。
5. 在 provider-app 应用详情页 >【部署组】标签页

- 新建部署组 provider-group-03，属于集群 cluster1，命名空间 dev-ns，添加实例，部署 provider-demo 程序包。
- 新建部署组 provider-group-04，属于集群 cluster2，命名空间 dev-ns，添加实例，部署 provider-demo 程序包。

步骤三：验证就近路由

consumer 和 provider 部署成功后，观察服务依赖拓扑图是否出现，如果出现说明服务间调用正常。

1. 在控制台左侧导航栏，单击【服务治理 】。
2. 单击 provider-demo 进入微服务详情页，单击【服务路由】标签页的流量详情中可以观察来自 consumer 的请求是否全部路由到部署组 provider-group-03 中，如果请求全部路由到部署组 provider-group-03 证明就近路由功能生效。

步骤四：验证容灾

1. 停止部署组 provider-group-03。
2. 观察 provider-demo 服务路由页面的流量详情中可以观察到流量分配到部署组 provider-group-04，证明容灾能力生效。

步骤五：关闭就近路由

1. 启动部署组 provider-group-03，观察到流量分配回 provider-group-03。
2. 在命名空间页面关闭就近路由开关，观察到流量均匀分配到2个部署组。

就近路由与路由规则

开启就近路由后，服务消费者会按照路由规则优先调用同一可用区的服务提供者，只有当同一可用区的服务提供者不可用时，会按照路由规则进行跨可用区调用。

场景：集群1和集群2都部署了 provider 的两个版本 v1 和 v2；路由规则是90%的流量分配到 v1，10%的流量分配到 v2；命名空间开启了就近路由。

- 当集群1中 provider 的 v1 和 v2 版本实例都正常时，consumer 会按照路由规则调用同一可用区的 provider。
- 当集群1中 provider 的 v1 实例不正常， v2 版本实例正常时，consumer 会按照路由规则将90%请求发送给可用区二的 v1 ，10%请求发送给可用区一的 v2。
- 当集群1中 provider 的 v1 和 v2 实例都不正常，consumer 会按照路由规则将所有90%请求发送给可用区二的 v1，10%请求发送给可用区二的 v2。

基于业务参数的服务治理最佳实践

最近更新时间: 2025-02-18 16:02:00

操作场景

本文档指导您通过微服务网关和服务路由能力，即可实现基于业务参数的服务治理。您无需修改代码，即可实现基于业务参数的路由、鉴权功能，并依据业务参数过滤请求调用链。

在实际的业务场景中，经常有依据某些请求参数值决定请求路由到某个服务的某个版本中的场景，或依据请求参数值对请求进行限流、鉴权等场景。 如下图中，当外网请求通过网关访问后端微服务时：

- 当请求参数 region = guangzhou 时，路由到微服务的版本1中。
- 当请求参数 region = shanghai 时，路由到微服务的版本2中。

前提条件

在开始本文的实践前，您需要先了解 TSF 的以下功能：

- 【微服务网关的部署 、【通过微服务网关管理 API】
- 微服务网关插件（tag 类型插件）
- 【服务路由】
- 【标签】

操作步骤

步骤1：新建微服务网关分组

1. 准备 consumer -demo 和 provider - demo 两个微服务。要求如下：

- provider - demo 存在两个版本的包：provider-demo-guangzhou.jar 和 provider-demo-shanghai.jar。
- provider - demo 两个版本在访问请求中携带 region 参数，且访问 /test-region 接口时，会分别返回 shanghai 和 guangzhou；访问 provider - demo 时，可以携带两个 Header 参数、region 和 usertype。
- consumer - demo 部署在一个部署组上，provider - demo 部署在两个部署组上，每个部署组部署一个版本的 jar 包。

2. 部署一个微服务网关 ，部署后服务名为 msgw - demo，部署组名称为 test，默认端口为8080（给对应机器开放8080端口）。在本 demo 中，将直接通过网关访问微服务 provider。

3. 新建微服务网关分组，并将微服务网关分组绑定在创建好的网关应用部署组上。 新建微服务网关： 绑定网关部署组：

4. 将微服务 API 导入到分组中，并将分组进行发布。

步骤2：配置微服务网关插件

在这一步中，我们在网关配置插件，将请求参数转化为 TSF 中的标签信息。

1. 在 【TSF 控制台】 左侧导航栏中，单击【微服务网关】>【插件管理】。
2. 在插件管理页左上角，单击【新建插件】，填写以下信息：

- 插件类型：选择 Tag。
- 插件名称：最长60个字符。

- 配置自定义标签：将请求参数中 Header 参数中的 region、usertype 设置为标签。

3. 单击【完成】，跳转至插件管理列表页。

4. 选择目标插件，单击操作列的【绑定对象】，将创建好的插件与步骤1中创建的分组进行绑定。
5. 单击【提交】，完成配置。

步骤3：配置服务治理规则

在这一步中，我们配置依据步骤2已经转化的标签，配置服务治理规则。当前 TSF 标签可以与服务路由、服务鉴权、服务限流、调用链进行联动。此处将为您介绍路由、鉴权、以及调用链联动功能。

服务路由

1. 在 【TSF 控制台】 左侧导航栏中，单击【服务治理】，进入服务列表页。
2. 找到创建好的 provider - demo 微服务，单击微服务名称，进入服务详情页。
3. 在服务详情页顶部，单击【服务路由】，进入服务路由页面。
4. 在服务路由页左上角，单击【新建路由规则】，创建两条规则：

- 当自定义标签 region 值为 guangzhou 时，100%的流量指向部署了 provider-demo-guangzhou.jar 的部署组。

- 当自定义标签 region 值为 shanghai 时，100%的流量指向部署了provider-demo-shanghai.jar 的部署组。规则配置可参考下图：

> **注意：**
>
> 路由规则始终是在被调用方进行配置的。

5. 单击【完成】，该规则即可生效。

结果验证： 公网发送请求 ： `http://imgcache.finance.cloud.tencent.com:80129.XX.XX.XX:8080/test-region/tj-test-1_default/provider-demo/test-region/12` 并携带 Header 参数 region = shanghai 若配置的路由生效，则请求会路由到返回值为

shanghai 的部署组中。

同理，当请求 Header 参数 = guangzhou 时，则请求会路由到返回值为 guangzhou 的部署组中。

服务鉴权

1. 在 [TSF 控制台] 左侧导航栏中，单击【服务治理】，进入服务列表页。
2. 找到创建好的 provider - demo 微服务，单击微服务名称，进入服务详情页。
3. 在服务详情页顶部，单击【服务鉴权】，进入服务鉴权页面。
4. 选择服务鉴权方式为黑名单鉴权，并单击页面左下角的【新建鉴权规则】。
5. 配置鉴权规则。当请求参数 Header 中携带了参数 usertype = user 时，请求不通过，且永久生效。配置方式如下：

- 标签类型：自定义标签
- 标签名：usertype
- 逻辑关系：等于
- 值：user
- 生效状态：永久生效

6. 单击【完成】，完成服务鉴权，自动跳转至服务鉴权页面。

结果验证： 同理发送请求，携带 Header 参数 usertype = user，region = shanghai。发现返回请求失败。

调用链查询

在 TSF 中，我们提供了基于请求标签过滤调用链的能力，您可以依据业务数据过滤对应请求的调用链。最为常见的场景是查询某个用户 ID 的请求调用成功失败情况以及层级耗时。

使用方法：

1. 在 【TSF 控制台】左侧导航栏中，单击【依赖分析】>【调用链查询】，进入调用链查询页面。
2. 选择对应的命名空间和微服务，单击 “展开高级查询条件”。
3. 输入查询标签。 我们输入 userid：1000001，过滤 userid 为1000001的请求数据。

4. 单击【查询】，即可查看携带对应标签的请求 Trace 数据列表。

Spring Cloud 原生应用"0"改造迁移上TSF

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 支持原生 Spring Cloud 应用无侵入接入，无需改造即可直接接入 TSF，享受服务注册与发现、服务治理、应用监控和调用链跟踪等功能。

本文档以一个 [开源商城系统] 为示例，为您介绍将原生 Spring Cloud 应用迁移到 TSF 的方法。

该系统由以下几部分组成：

模块	说明
mall-admin	后台管理系统
mall-auth	角色认证模块
mall-gateway	网关模块，请求入口
mall-portal	前台商城系统
mall-search	商品检索模块
mall-demo	用来测试 API 的样例工程
mall-monitor	Spring 自带的监控模块，TSF 自带监控能力，因此可以略过

环境准备

环境配置建议

> **注意：**
>
> 以下配置仅做建议，具体以您的实际业务需求为主。

- 开发环境：指含有 mall demo 程序源码的计算环境。
- 部署环境：指购买的云主机，并且运用 TSF 服务部署商城系统的环境。

环境	环境分类	配置
开发环境	-	CPU: 4核内存：8GB 网络：50Mbps
部署环境	中间件部署服务器	配置：1台云主机CPU：4核内存：8GB网络：50Mbps
部署环境	微服务部署服务器	配置：每个服务1台云主机CPU：1核内存：2GB网络：20Mbps，按量计费磁盘：50GB

中间件部署服务器准备

1. 参考环境配置建议 【购买云服务器】 。

2. 安装 【Docker】 和 【Docker Compose】 。

3. 下载 【mall-demo程序包】 ，并将其上传到云服务器中。

4. 进入 `tsf-demo-public/document/docker` 目录，执行如下命令，等待下载和容器拉起完成。

```
docker-compose -f docker-compose-env.yml up -d
```

下载时间根据实际网络带宽可能需等待几分钟到几十分钟不等。

5. 执行下面的命令创建 RabbitMQ 的 virtual_host、用户和权限，需要等 RabbitMQ 启动完成，如果下面的命令报错，再次执行。

```
docker exec -it rabbitmq /init.sh
正常情况下屏幕会显示如下：
Adding user "mall" ...
Setting tags for user "mall" to [administrator] ...
Adding vhost "/mall" ...
Setting permissions for user "mall" in vhost "/mall" ...
```

迁移上云

步骤1. 准备应用程序包

前提条件

安装 Maven

操作步骤

1. 下载 mall-demo程序包 到本地。

2. 在 `tsf-demo-public` 根目录下执行如下命令，进行依赖初始化，耗时根据网速可能不同。

```
mvn clean
```

3. 进入每个项目的 `src/main/resource` 目录，根据已经部署的容器所有的云服务器地址，修改 application.yml 文件中的连接信息。

若在本地安装调试可以忽略本步骤，即在本地安装docker和所有基础组件，在本地启动Spring Cloud调试。

```
yml
# mysql中替换localhost为内网IP
url: jdbc:mysql://localhost:3306/mall?useUnicode=true&characterEncoding=utf-8&serverTimezone=Asia/Shanghai
# redis中替换localhost为内网IP
host: localhost
# rabbitmq中替换localhost为内网IP
host: localhost
# mongo中替换localhost为内网IP
host: localhost
```

```
# ES中替换127.0.0.1为内网IP
uris: 127.0.0.1:9200
```

4. 进入 mall-mbg 项目的 `src/main/resource` 目录，修改 generator.properties 文件中 MySQL 的连接信息，修改localhost为指定主机名/IP。

5. 在 `tsf-demo-public` 根目录下，执行如下命令将项目进行打包。

```
mvn clean package -DskipTests
```

6. 在 target 目录下，可看到生成的 jar 程序包。

```
需上传jar包本地路径
mall-admin/target/mall-admin-1.0-SNAPSHOT # 后台管理系统
mall-auth/target/mall-auth-1.0-SNAPSHOT # 角色认证模块
mall-gateway/target/mall-gateway-1.0-SNAPSHOT # 网关模块，请求入口
mall-portal/target/mall-portal-1.0-SNAPSHOT # 前台商城系统
mall-search/target/mall-search-1.0-SNAPSHOT # 商品检索模块
两个可选部署的服务
mall-demo/target/mall-demo-1.0-SNAPSHOT # 用来测试API的样例工程
mall-monitor/target/mall-monitor-1.0-SNAPSHOT # Spring自带的监控模块，TSF自带监控能力，故可以略过
```

步骤2. 部署应用到 TSF

以部署 mall-search 服务为例，介绍在 TSF 上部署一个应用的流程。

2.1 新建集群

1. 登录 【TSF 控制台】，左侧导航栏选择【集群】，单击新建，创建一个名为 mall-demo 的集群。
2. 单击集群操作栏的【导入云主机】，将购买的云服务器全部导入到集群中。

2.2 新建日志配置项

在左侧导航栏选择【日志服务】>【日志配置】，单击【新建日志配置项】，创建日志采集规则。

2.3 创建并部署应用

1. 在左侧导航栏选择【应用管理】，单击【新建应用】，创建一个名为 mall-search 的应用。

2. 单击【提交】后，在提醒弹窗"是否前往倒入程序包，并部署应用"中单击【确认】，前往上传程序包。
3. 在程序包管理页面，单击【上传程序包】，将 mall-search-1.0-SNAPSHOT.jar 程序包上传到TSF平台。
4. 单击【提交】后，在弹窗"已上传完程序包，是否部署应用"中选择【前往部署】，前往创建部署组。
5. 在部署组页面，单击【新建部署组】，填写部署组信息。

- 集群：选择**2.1步骤**中创建的集群

- 日志配置项：选择**步骤2.2**中创建的日志配置项

6. 单击【保存&下一步】，选择要部署的云主机，单击【部署应用】。
7. 在部署应用页面，选择刚刚上传的程序包版本，健康检查建议勾选"存活检查"和"就绪检查"，因为项目已经集成 actuator，如图填写请求路径即可，端口号根据 application.yml 中定义填写。

健康检查：

8. 单击【完成】，完成应用部署。

2.4 查看部署结果

重复本章节**2.3步骤**依次将所有服务部署到TSF上，服务部署顺序建议：**服务网关 mall-gateway -> mall-auth -> mall-admin -> mall-portal -> mall-search -> mall-demon。**

当完成所有的服务部署，部署结果如下。

部署结果验证

步骤1. 验证服务依赖功能

通过部署前端页面，验证服务依赖功能

1. 登录中间件部署服务器，在服务器上安装 【node.js】 。
2. 下载前端代码，地址 [mall-admin-web] 。
3. 在项目根目录下执行如下命令，安装前端项目所需的第三方依赖。

```
npm install
```

4. 修改 dev.env.js 文件中的 BASE_API 配置为网关服务的端口，示例如下：IP 为gateway 服务机器内网 IP，port 为服务的端口号。

```
http://imgcache.finance.cloud.tencent.com:80IP:PORT/mall-admin
```

5. 执行如下命令运行前端项目。

```
npm run dev
```

6. 访问前端页面，地址： `http://imgcache.finance.cloud.tencent.com:80中间件服务器的外网 IP: 8090` ，体验服务。
7. 登录 【TSF 控制台】，在【依赖分析】>【服务依赖拓扑】页面，选择命名空间和时间后，可看到如下图的依赖关系。

步骤2. 验证服务治理功能

2.1 验证服务限流功能

服务限流详细介绍请参考【服务限流】。

典型业务问题：后端业务被高频恶意访问，导致核心业务链路阻塞，系统瘫痪。

场景：用户频繁访问拉取商品列表接口。

需求：保证核心服务 mall - admin 被每秒中最多被请求20次。

规则配置：在TSF控制台服务治理页面找到 mall-admin 服务，进入服务详情页面，配置服务限流规则。

效果验证：

2.2 验证服务鉴权功能

服务鉴权详细介绍请参考【服务鉴权】。

电商典型场景：后端敏感业务需要对访问权限进行控制。

场景：对于后台商品管理模块，仅支持有权限的服务对它进行访问。例如，在这个场景中，我们限制gateway微服务可以不访问mall admin微服务，所有从gateway发起的请求都会被拒绝。

配置方式：在 【TSF 控制台】 服务治理页面找到 mall-admin 服务，进入服务详情页面，配置服务鉴权规则。

效果验证：

自动化部署

当应用非常多，不希望使用控制台逐个部署怎么办呢？ 或者已经使用了 jenkins、travis 等工具，如何对接到 TSF 平台上呢？我们可以参考下面的操作来进行实践。

【mall-demo 程序包】 中的 deploy.py 脚本支持自动上传和部署一个新的应用到现有的集群中，默认选择集群中可用实例中的第一个实例机器部署应用。

1. 在 deploy 目录下的 deploy.py 文件中配置 secret_id、secret_key，clusterId 和 namespace 等参数。

参数	是否必选	说明
path	必选	程序包路径
applicationName	必选	应用名称
appId	必选	账号 APPID

参数	是否必选	说明
groupName	可选	默认采用和应用名称同名，不可重复
microserviceType	可选	默认“NATIVE" 云原生应用。否，填写 “N”
applicationType	可选	默认“V”表示虚拟机部署
pkgVerstion	可选	上传的程序包版本号，默认当前时间戳，时间戳格式：”YYYYmmddHHMMSS”

2. 在 travis.yml 中添加脚本任务和任务所需的执行参数。依次是：程序包路径、应用名和 APPID。

```
- ./scripts/deploy.py mall-demo/target/mall-demo-1.0-SNAPSHOT.jar "test" "1234567890"
```

3. 提交 commit，并且推送到远程分支，自动触发 Travis CI 流程。Travis 流程执行成功。

4. 登录 【TSF 控制台】，可看到新的应用、部署组和运行实例。 应用： 部署组： 运行实例：

Spring Cloud Alibaba迁移TSF

最近更新时间: 2025-02-18 16:02:00

操作场景

应用迁移到TSF平台后即可享受平台一站式微服务解决方案：管理资源和应用更加便利，丰富的服务治理能力，多维度的监控和跨可用区高可用方案等。

本文介绍从Spring Cloud Alibaba迁移到TSF平台所需的改造工作，经过改造后，服务可以在TSF平台上成功注册和互相调用。现在就目前Spring Cloud Alibaba常见的Restful和Dubbo两类服务框架，分别说明改造方法。

操作步骤

Restful服务改造

这部分改造主要包括删除Nacos依赖和配置，添加TSF的依赖和启动类的注解。

步骤1. 依赖调整

1. 以下Nacos配置中心、注册中心、Spring Boot、Spring Cloud的依赖，在迁移的时候，需要删除。

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>com.alibaba.cloud</groupId>
<artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
</dependency>
<dependency>
<groupId>com.alibaba.cloud</groupId>
<artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<!-- Spring Cloud Open Feign -->
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>
```

2. 添加TSF依赖，TSF SDK已经包含Spring Boot、Spring Cloud的依赖。

```
<!-- TSF 启动器 -->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
```

```
<version><!-- 调整为 SDK 版本号 --></version>
</dependency>
```

步骤2. 配置调整

以下是Nacos服务注册中心和配置中心的配置项示例，在迁移时，要删除配置文件中这部分内容。

Nacos服务注册中心

```
spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
#spring.cloud.nacos.discovery.instance-enabled=true

spring.cloud.nacos.username=nacos
spring.cloud.nacos.password=nacos

management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
```

Nacos配置中心

```
spring.cloud.nacos.config.server-addr=127.0.0.1:8848

#nacos certification information
spring.cloud.nacos.username=nacos
spring.cloud.nacos.password=nacos

#spring.cloud.nacos.config.refreshable-dataids=common.properties
#spring.cloud.nacos.config.shared-data-ids=common.properties,base-common.properties
spring.cloud.nacos.config.shared-configs[0]= common333.properties
spring.cloud.nacos.config.shared-configs[1].data-id= common111.properties
spring.cloud.nacos.config.shared-configs[1].group= GROUP_APP1
spring.cloud.nacos.config.shared-configs[1].refresh= true
spring.cloud.nacos.config.shared-configs[2]= common222.properties

#spring.cloud.nacos.config.ext-config[0]=ext.properties
spring.cloud.nacos.config.extension-configs[0].data-id= extension1.properties
spring.cloud.nacos.config.extension-configs[0].refresh= true
spring.cloud.nacos.config.extension-configs[1]= extension2.properties
spring.cloud.nacos.config.extension-configs[2].data-id= extension3.json
```

步骤3. 代码调整

向应用启动类中添加 `@EnableTsf` 注解。

步骤4. 应用部署

实现以上三步操作并且成功编译打包后，即完成改造，进入应用部署环节。TSF平台支持多种应用类型，Restful应用可以采用普通应用类型进行部署，更多信息请参考【应用部署】，按照文档指引操作即可。

Dubbo服务改造

TSF平台纳管Spring Cloud Alibaba Dubbo应用采用的是TSF Mesh技术，TSF Mesh支持Dubbo、Http等协议，通过Sidecar注册到服务注册中心、处理服务间通信。

改造主要涉及两方面内容，一个是Mesh化，引入Sidecar，另一个是移除原有注册中心的依赖和配置，TSF兼容双注册，在不移除原有注册中心的情况下，经过少量代码调整，也可以成功部署应用，但是为了降低程序包的大小，加快应用部署和启动速度，建议要移除原有注

册中心的依赖和配置。

步骤1. Mesh改造

服务定义和注册（必选）

创建spec.yaml文件，定义服务名、暴露的端口和协议，服务名和接口名相同。由于Dubbo 2采用的是接口级服务发现，所以需要每个接口都要进行定义，在最新版的Dubbo 3中，采用了应用级服务发现，操作会简化很多。

```
apiVersion: v1
kind: Application
spec:
services:
- name: com.dubbo.service.UserService
ports:
- targetPort: 20881
protocol: dubbo
healthCheck:
path:
```

API 定义和上报（可选）

创建apis目录，该目录放置服务的 API 定义。一个服务对应一个 yaml 文件，文件名即服务名，API 遵循 OPENAPI 3.0 规范。

```
openapi: 3.0.0
info:
version: "1.0.0"
title: user service
paths:
/api/v6/user/create:
get:
responses:
'200':
description: OK
'401':
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
/api/v6/user/account/query:
get:
responses:
'200':
description: OK
'401':
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
/health:
get:
responses:
'200':
description: OK
'401':
```

```
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
```

调用链 Header 传递（可选）

要实现 Mesh 应用调用链和服务依赖拓扑功能，需要在请求中带上9个相关 header。

```
// 9个调用链相关的头，具体说明(http://imgcache.finance.cloud.tencent.com:80www.envoyproxy.io/docs/envoy/v1.8.0/configuration/http_conn_man/headers.html?highlight=tracing)
traceHeaders = ['x-request-id',
'x-trace-service',
'x-ot-span-context',
'x-client-trace-id',
'x-b3-traceid',
'x-b3-spanid',
'x-b3-parentspanid',
'x-b3-sampled',
'x-b3-flags']
```

步骤2. 依赖调整

删除nacos服务注册和配置的依赖，如下例：

```
<dependency>
<groupId>com.alibaba.cloud</groupId>
<artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
</dependency>
<dependency>
<groupId>com.alibaba.cloud</groupId>
<artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
</dependency>
```

步骤3. 配置调整

关闭Dubbo服务注册中心配置，如下例：

```
spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
#spring.cloud.nacos.discovery.instance-enabled=true

spring.cloud.nacos.username=nacos
spring.cloud.nacos.password=nacos

management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
```

删除Nacos配置中心的配置，如下例：

```
spring.cloud.nacos.config.server-addr=127.0.0.1:8848

#nacos certification information
spring.cloud.nacos.username=nacos
spring.cloud.nacos.password=nacos
```

```
#spring.cloud.nacos.config.refreshable-dataids=common.properties
#spring.cloud.nacos.config.shared-data-ids=common.properties,base-common.properties
spring.cloud.nacos.config.shared-configs[0]= common333.properties
spring.cloud.nacos.config.shared-configs[1].data-id= common111.properties
spring.cloud.nacos.config.shared-configs[1].group= GROUP_APP1
spring.cloud.nacos.config.shared-configs[1].refresh= true
spring.cloud.nacos.config.shared-configs[2]= common222.properties

#spring.cloud.nacos.config.ext-config[0]=ext.properties
spring.cloud.nacos.config.extension-configs[0].data-id= extension1.properties
spring.cloud.nacos.config.extension-configs[0].refresh= true
spring.cloud.nacos.config.extension-configs[1]= extension2.properties
spring.cloud.nacos.config.extension-configs[2].data-id= extension3.json
```

删除Hystrix配置

```
feign:
hystrix:
enabled: true
```

步骤4. 代码调整

1. 删除服务发现的注解 `@EnableDiscoveryClient` 。
2. 服务消费端的 `@Reference` 注解添加url属性，属性值为 `dubbo：//服务名:端口` ，添加 `check=fales` 属性。在兼容双注册的场景下，这个代码需要额外添加。

步骤5. 应用部署

代码改造完并且成功编译打包后，进入应用部署环节。在TSF平台创建Mesh应用类型进行部署，更多信息请参考【应用部署】，按照文档指引操作即可。

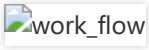
使用工具部署应用
使用 Coding 创建持续集成

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 应用可以使用 CODING 构建持续集成方案。CODING 持续集成全面兼容 Jenkins 的持续集成服务

CODING DevOps 工作流

下图展示了使用 CODING 持续集成，TSF 作为应用部署平台的工作流。 

准备工作

在开始持续集成之前，需要完成下述的准备工作：

1. 参考 【使用 Python 脚本部署应用】 获取部署脚本及使用说明。
2. 在 TSF 平台创建了 【虚拟机应用部署组】 或 【容器应用部署组】 。
3. 获取 TSF 私服地址，详情请参见 【SDK 下载】。
4. 主账号已开通 【CODING DevOps】 ，并创建 CODING 项目、代码仓库等基础环境。

CODING 平台操作

步骤一：创建代码仓库并提交代码

1. 登录 【CODING DevOps 控制台】 ，进入示例项目。

2. 单击左侧导航栏【代码仓库】> 【代码浏览】，单击顶部导航栏，创建代码仓库（如 `provider-demo` ）。

3. 下载 演示工程源码，上传到 Coding 的 `provider-demo` 代码仓库中。

> **注意：**
>
> 用户需要修改示例代码中的 Python 部署脚本，替换其中访问密钥、地域等信息 。

步骤二：创建并配置构建任务

创建构建任务

1. 单击 Coding 控制台左侧导航栏【构建与部署】>【构建】。
2. 单击【新建】。
3. 填写新建信息，选择代码仓库 `provider-demo` ，使用静态配置的 Jenkinsfile ，选择简易模板。

设置环境变量

单击【变量与缓存】选项卡，输入环境变量：

- PKG_VERSION：程序包/镜像版本号
- TSF_GROUP_ID: TSF 平台的部署组 ID
- TSF_APPLICATION_ID：TSF 平台的应用 ID （仅适用于虚拟机应用部署）
- TSF_APPID：用户在云上的 APPID （仅适用于虚拟机应用部署）
- TSF_STARTUP_PARAM：Java 应用启动参数，注意**不要**带上引号（仅适用于虚拟机应用部署）

流程配置-构建

1. 在【流程配置】选项卡【图形化编辑器】中，选中【构建】阶段方块。

2. 单击【执行 Shell 脚本】。

3. 编辑【Shell 脚本】，填写 maven package 命令 ，并使用工程中包含 TSF 私服地址的settings.xml。单击【保存】。

```
mvn clean package -U --settings settings.xml -Dmaven.test.skip=true
```

流程配置-部署

1. 在【流程配置】选项卡【图形化编辑器】中，选中【部署】阶段方块。
2. 单击【增加步骤】。
3. 单击【执行 Shell 脚本】，输入 Shell 脚本，填写 Python 部署脚本执行命令。

a. 虚拟机应用部署脚本命令：

```
python upload_virtual_machine_deploy.py provider-demo/target/provider-demo-0.0.1-SNAPSHOT.jar ${TSF_APPLICATION_ID} ${PKG_VERSION} ${TSF_APPID} ${TSF_GROUP_ID} ${TSF_STARTUP_PARAM}
```

b. 容器应用部署脚本命令：

```
python upload_container_deploy.py ${TSF_GROUP_ID} ${PKG_VERSION}
```

步骤三：执行构建任务

构建任务可以通过代码提交等方式自动触发，也可以手动执行。

1. 选中目标构建任务，单击【立即构建】。

2. 填写环境变量，将【值】替换为用户自己的参数，单击【立即构建】。

步骤四：验证构建结果

1. 构建任务通常会执行几分钟，查看构建任务的结果。

2. 在 TSF 控制台，查看对应部署组的状态。

使用 Jenkins 创建持续集成

最近更新时间: 2025-02-18 16:02:00

TSF持续集成之 Jenkins

TSF 应用可以使用 Jenkins 构建持续集成方案。

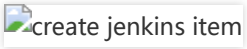
准备工作

在开始持续集成之前，需要完成下述的准备工作。

1. 参考 【使用Python脚本部署应用】 获取部署脚本及使用方法的说明。
2. 安装 Jenkins ，确保 Jenkins 安装机器上的 Python 不低于 2.7.14 版本，并已安装 PIP 等 Python 包管理工具。
3. 在 TSF 平台创建了【虚拟机应用部署组】 或【容器应用部署组】。

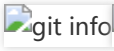
安装和配置 Jenkins

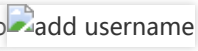
1. 进入 Jenkins 官网下载安装 Jenkins。
2. 安装 Maven 并配置TSF私服地址 。
3. 在 Jenkins 控制台的菜单栏中选择系统管理 > 插件管理，安装 Maven Integration plugin 插件,已安装请忽略。
4. 创建Jenkins项目 4.1 在 Jenkins 首页左侧导航栏中单击新建，创建 Jenkins 任务，并选择构建一个自由风格的软件项目。



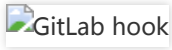
4.2 在 源码管理 页面中选择 Git，并设置相关参数。 Repository URL：您的项目的 Git 协议地址。

Credentials：安全凭证，选择无（前提是运行 Jenkins 软件的用户的 SSH RSA 公匙已添加到该 Git 项目所在的 GitLab或GitHub 中，否则这里会报错）或者添加用户名密码。

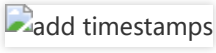




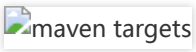
4.3 单击构建触发器页签，勾选Build when a change is pushed to GitLab。 GitLab webhook URL 中的IP要确保GitLab能访问。



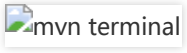
4.4 单击构建环境页签，勾选 Add timestamps to the Console Output（为控制台输出的信息添加时间戳）。



4.5 单击构建页签，然后单击增加构建步骤，选择 Invoke top-level Maven targets。



目标：填入 clean package （如有其它参数，请根据实际情况填入）。



4.6 单击构建页签，然后单击增加构建步骤，选择 Execute shell。

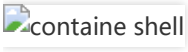
命令中填入 应用部署准备 步骤中准备好的Shell命令。 i.虚拟机如下：

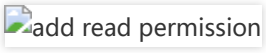
```
python2.7.14 upload_virtual_machine_deploy.py ./consumer-demo/target/consumer-demo-1.10.0-RELEASE.jar application-qab76pxv v001 1300555551 group-gvk5pbdv '-Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=
```

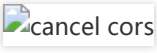
```
512m'
```

ii.容器如下

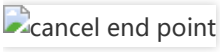
```
python2.7.14 upload_container_deploy.py group-zvw397wa v1
```

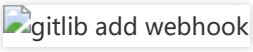


4.7 配置GitLab所需权限。 i. 管理Jenkins > Configure Global Security 勾选 匿名用户具有可读权限。 

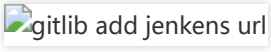
取消 防止跨站点请求伪造。 

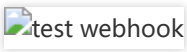
ii. 管理Jenkins > 系统配置 取消 Enable authentication for '/project' end-point

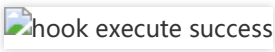


5. 配置 GitLab 的 Web Hook，实现自动构建。 5.1 项目 > Settings > Integrations 进入添加webhook界面 

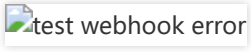
5.2

将 4.3 中Jenkins产生的GitLab webhook URL填入URL，其他选项使用默认设置，点击"Add webhook"。 

5.3 测试 WebHook 

成功如图 

如出现 403 异常，请检查 4.7 配置。



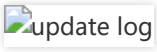
配置正确后，提交变更到 GitLab

如果上述步骤配置正确，这次提交会触发一次 GitLab Hook。 Jenkins 在接受到这个 Hook 后会构建您的 Maven 项目，并在构建过程中触发Python脚本自动部署TSF应用。

1. 虚拟机提交部署成功输出的日志信息（Build Number > 控制台输出）。

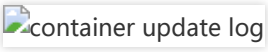
```
15:12:40 + python2.7.14 upload_virtual_machine_deploy.py ./consumer-demo/target/consumer-demo-1.10.0-RELEASE.jar application-qab76pxv v001 1300555551 group-gvk5pbdv '-Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m'
15:12:41 POST
15:12:41 /
15:12:41
15:12:41 content-type:application/json
15:12:41 host:tsf.tencentcloudapi.com
15:12:41
```

```
15:12:41 content-type;host
15:12:41 0c2f4a50b1dc4df621a84d1731ec07cbd756ac04b1f4853fa792c5dbcf104f28
15:12:41 TC3-HMAC-SHA256
15:12:41 1576739561
15:12:41 2019-12-19/tsf/tc3_request
15:12:41 46cdc18f96456edef21e4b01e787030d62ce54bb7b300994fed4d5d473709701
15:12:41 1d501d695675696fef1c2b2576bb21f42f6f31f55f07c7b27c5ca051d20f242e
15:12:41 TC3-HMAC-SHA256 Credential=AKIDxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/2019-12-19/tsf/tc3_request, SignedH
eaders=content-type;host, Signature=1d501d695675696fef1c2b2576bb21f42f6f31f55f07c7b27c5ca051d20f242e
15:12:41 {"Response":{"RequestId":"0b6aa4ee-38fe-4782-97e4-be1061e7cdb1", "Result":{"Content":[{"PkgId":"pkg-f51038e
0","PkgName":"consumer-demo-1.10.0-RELEASE.jar","PkgType":"fatjar","PkgVersion":"v001","PkgDesc":"","UploadTime":"20
19-12-18 11:03:26","Md5":"4ad6f32f20c1d2dcd60575fa50b16e1d","PkgPubStatus":1}],"TotalCount":1}}}
15:12:41 [INFO] application-qab76pxv has uploaded version v001, no need upload
15:12:41 POST
15:12:41 /
15:12:41
15:12:41 content-type:application/json
15:12:41 host:tsf.tencentcloudapi.com
15:12:41
15:12:41 content-type;host
15:12:41 99f3e6ddeaf00f7b64315835fdd6f29d7a0a5520743184c21a43d8a609d6891a
15:12:41 TC3-HMAC-SHA256
15:12:41 1576739561
15:12:41 2019-12-19/tsf/tc3_request
15:12:41 e5ebc373c1c432910ca0110e48f1fe4cf29ffcdedc21dc79dd59a99bf802ea23
15:12:41 73e21ed01f4ad54668e94a2839d6de17c07e6188e1ff9e176452b80fb776be15
15:12:41 TC3-HMAC-SHA256 Credential=AKIDxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/2019-12-19/tsf/tc3_request, SignedH
eaders=content-type;host, Signature=73e21ed01f4ad54668e94a2839d6de17c07e6188e1ff9e176452b80fb776be15
15:12:41 200
15:12:41 {"Response":{"RequestId":"86c93d7e-ee44-4c8d-8d34-537a0dbe6871","Result":{"TaskId":"task-76l2v6m7"}}}
15:12:41 Finished: SUCCESS
```

TSF平台可以查看虚拟机部署变更记录。 

2. 容器部署成功日志信息

```
16:32:28 + python2.7.14 upload_container_deploy.py group-zvw397wa v1
16:32:31 {"Result": {"InstanceNum": 1, "NamespaceName": "zsf-test-jenkins-docker_default", "ApplicationType": "C", "Curre
ntNum": 1, "MicroserviceType": "N", "Status": "Running", "LbIp": "", "MemRequest": "128", "ClusterId": "cls-52tydz2q", "Up
dateIvl": 0, "Envs": [{"Name": "tsf_consul_ip", "Value": "169.254.0.77"}, {"Name": "tsf_consul_port", "Value": "8000"}, {"Name":
"PATH", "Value": "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"}, {"Name": "workdir", "Value": "/app/"}, {"Nam
e": "jar", "Value": "consumer-demo-1.10.0-RELEASE.jar"}, {"Name": "tsf_app_id", "Value": "1300555551"}, {"Name": "tsf_toke
n", "Value": "VjCCe9gixVmdeBfTpM74PikHFfU8ZmZVL7Z4FMGvXjI="}, {"Name": "tsf_application_id", "Value": "application-b
yx3zwdy"}, {"Name": "tsf_cluster_id", "Value": "cls-52tydz2q"}, {"Name": "tsf_namespace_id", "Value": "namespace-py5e9zj
y"}, {"Name": "tsf_group_id", "Value": "group-zvw397wa"}, {"Name": "tsf_prog_version", "Value": "v1"}, {"Name": "tsf_regio
n", "Value": "ap-guangzhou"}, {"Name": "tsf_ratelimit_master_ip", "Value": "169.254.0.77"}, {"Name": "tsf_ratelimit_master_p
ort", "Value": "7000"}], "TagName": "v1", "ClusterIp": "172.18.255.122", "MemLimit": "256.00", "CpuLimit": "0.50", "Applicati
onId": "application-byx3zwdy", "ApplicationName": "zsf-tsf-docker", "ProtocolPorts": [{"Protocol": "TCP", "TargetPort": 80,
"Port": 80}], "UpdateType": 0, "Server": "ccr.ccs.tencentyun.com", "GroupName": "zsf-test-docker-consumer", "NodePort": 3
0215, "AccessType": 2, "ClusterName": "zsf-test-jenkins-docker", "NamespaceId": "namespace-py5e9zjy", "Reponame": "tsf
_100011913960/zsf-tsf-docker", "CpuRequest": "0.25", "Message": "", "CreateTime": "2019-12-17 15:09:08", "GroupId": "gro
up-zvw397wa"}, "RequestId": "aff9a2b6-24f5-44b1-8173-112a6a24b581"}
16:32:31 {"Result": true, "RequestId": "263c8a64-1bb1-427e-80e2-870aa0bc7d68"}
16:32:31 Finished: SUCCESS
```

TSF平台可以查看容器部署变更记录。 

3. 容器部署成功日志信息 如果部署失败，需要分析Jenkins console日志。

使用Python脚本部署应用

最近更新时间: 2025-02-18 16:02:00

TSF 应用可以使用 Python 脚本来部署。

前提条件

在开始持续集成之前，需要完成下述的准备工作：

1. 保证机器上安装的 Python 版本不低于 `2.7.14` 版本，并已安装 PIP 等 Python 包管理工具。
2. 获取云平台的 【访问密钥】（SecretId 和 SecretKey）。
3. 在 TSF 平台创建了【虚拟机应用部署组】 或 【容器应用部署组】。
4. 了解 Python 脚本使用。

虚拟机应用部署准备

1. 机器上保证安装的 Python 版本不低于 `2.7.14` 版本，并已安装 PIP 等 Python 包管理工具。
2. 从 GitHub仓库 下载虚拟机部署 Python 脚本。
3. 修改脚本中的secret_id 、secret_key 为访问密钥，region 为 TSF 服务所在地域。

```
secret_id = "改为您的 SecretId"
secret_key = "改为您的 SecretKey。"
region = "改为您的服务所在地域，如 ap-guangzhou"
```

4. 安装脚本依赖包。

```
pip install requests cos-python-sdk-v5
```

5. 准备脚本参数，要严格保证参数顺序。

- `path` ：本地文件路径，可以是针对脚本的相对路径，支持 `.tar.gz` 、 `.jar` 、 `.war` 、 `.zip` 结尾的文件。
- `applicationId` ：应用 ID。在 【TSF 控制台】>【应用管理】中，选择目标应用第一列的应用 ID（如 `application-qab76pxv` ）。
- `pkg_version` ：程序包版本，最长32个字符，支持 a-z、A-Z、0-9、横杠（-）、下划线（_）。
- `appId` ：用户 APPID。在控制台【[账号中心] 】>【账号信息】中获取 APPID（如 `1300555551` ）。
- `group_id` ：部署组 ID。在【TSF 控制台】>【应用管理】中，单击目标应用 ID，进入详情页，在【部署组】标签页中获取部署组的 ID（如 `group-gvk5pbdv` ）。
- `startup_params` ：启动参数。用户视情况可以自定义。

```
-Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m
```

> **注意：**
>
> 执行的脚本参数中如果版本号不变时，脚本会选择 TSF 平台已有的 jar 包进行再次部署。

6. 将以上参数按照顺序整理待用，格式如下：

```
python2.7.14 upload_virtual_machine_deploy.py ./consumer-demo/target/consumer-demo-1.10.0-RELEASE.jar application-qab76pxv v001 1300555551 group-gvk5pbdv '-Xms128m -Xmx512m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=512m'
```

容器应用部署准备

1. 保证机器上能够构建、上传镜像 ， 保证 Python 版本不低于2.7.14版本，并已安装 PIP 等 Python 包管理工具。
2. 从 GitHub仓库 下载容器部署 Python 脚本。
3. 修改脚本中的 secret_id 、secret_key 为访问密钥，region 为 TSF 服务所在地域。修改脚本中的 docker_build_command、docker_push_command 为实际的 docker build 和 push 命令。

```
secret_id = "改为Access Key ID"
secret_key = "改为Access Key Secret。"
docker_build_command = "改为 docker build 命令"
docker_push_command = "改为 docker push 命令"
region = "改为您的服务所在地域，如 ap-guangzhou"
```

4. 安装脚本依赖包。

```
pip install tencentcloud-sdk-python
```

5. 准备脚本参数，要严格保证参数顺序。

- `group_id` ：部署组 ID。在【TSF 控制台】>【应用管理】中，单击目标应用 ID，进入详情页，在【部署组】标签页中获取部署组的 ID（如 `group-zvw397wa` ）。
- `tag_name` ：镜像版本名称，如v1。

> **注意：**
>
> 目前容器应用部署脚本只将 group_id 和 tag_name 作为参数，但实际上用户可以修改脚本，将其他字段（如 JVM 启动参数 JvmOpts 等）作为脚本参数 。

6. 将以上参数按照顺序整理，并且加入第一步中用户自己的 docker build 、push 命令待用，格式如下：

```
python2.7.14 upload_container_deploy.py group-zvw397wa v1
```

> **注意：**
>
> docker 相关命令必须按照 [制作容器镜像] 和 [镜像仓库] 调整为用户自己的账号和应用名。

命名空间高可用模式

最近更新时间: 2025-02-18 16:02:00

操作场景

在注册中心异常或组件升级更新场景中，当提供方服务实例无法与注册中心维持正常心跳，导致注册中心服务实例变更离线状态时，调用方服务订阅注册中心由于获取提供方异常服务实例列表并更新本地缓存信息，从而引发服务调用异常。 针对以上异常场景，TSF 面向命名空间维度提供高可用模式配置，业务在命名空间开启高可用模式后，当注册中心发生异常时优先从本地缓存获取提供服务实例列表，并启动定时任务探测注册中心状态，在注册中心恢复后2个心跳周期（保障提供方服务完成服务注册）后恢复注册中心订阅机制，并更新本地缓存信息从而保障业务服务高可用。

> **注意：**
>
> 需配套1.25以上版本 SDK 方可生效。

实现原理

命名空间高可用模式实现原理如下：

正常状态服务调用流程：

1. 提供方注册服务信息；
2. 调用方订阅服务信息；
3. 调用方更新本地缓存；
4. 调用方发起服务调用。

注册中心异常状态服务调用流程：

1. 如果服务进程重启则从本地缓存获取服务列表发起调用，如果服务未重新启动则内存中缓存服务列表；
2. 调用方发起服务调用；
3. 调用方启动定时任务定时探测中心中心状态。

注册中心恢复后服务调用流程：

1. 提供方服务正常进行服务注册；
2. 调用方服务通过缓存正常发起调用；
3. 调用方定时探测注册中心状态恢复并进行计；
4. 待注册中心两个心跳周期后订阅服务（保障提供方服务完成服务注册）；
5. 更新本地缓存服务列表。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏中，单击【命名空间】，进入命名空间列表页。

3. 在命名空间列表页，选择指定的命名空间，单击打开高可用模式开关。

基于TSF Mesh的前端静态资源托管

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导用户依托 TSF Mesh 技术以服务形式托管 Node.js 和 Nginx，最终验证 Node.js 与 Nginx 可以以服务形式注册、发现及成功调用。概要架构图如下：

- Nginx：前端资源托管。
- Node.js：后端服务。
- Sidecar：和服务运行在同一个 Pod 中，与 Pod 共享网络，服务感受不到 Sidecar 的存在。
- Sidecar 代理服务向注册中心注册服务相关信息，以便其他服务发现自身。
- Sidecar 作为 Pod 内服务的 HTTP 代理，可以自动发现其他服务。
- Consul Cluster：管理和配置 Sidecar 来执行策略并收集遥测。

操作步骤

步骤1：部署 Node.js 服务

1. 准备 Node.js 应用代码。 可参考 Demo 中配置文件进行构建，其它语言的 TSF Mesh 的 Demo 示例可参考 【TSF Mesh Demo】 。
2. 制作 Node.js + express 应用镜像
3. 上传到镜像仓库 。
4. 部署 Node.js 应用：
5. 【创建应用】 应用类型选择 Mesh 应用。
6. 在左侧导航栏选择【部署组】>【新建部署组】，创建 Node.js 的部署组。
7. 在部署组页面，选择部署组右侧操作列的【部署应用】，选择刚刚创建的镜像，设置部署相关信息。

步骤2：托管前端静态资源

1. 手动在云服务器上安装 Nginx，详细操作请参考 Nginx 官网的 安装说明。

2. 准备程序包。程序包的目录为：

   - `www` 目录：该目录名可任意，用于存放 Web 静态资源文件。
   - `start.sh` 脚本：将 www 目录下文件移动到 Nginx 站点目录下，同时启动 Nginx 服务。
   - `stop.sh` 脚本：停止 Nginx 服务。
   - `cmdline` 文件：检查 Nginx 进程是否存在的 `grep` 命令关键字。

   用户可参考 nginx_demo 了解各文件的具体作用和写法。

3. 根据 Dockerfile 生成本地镜像并上传到镜像仓库。

4. 部署 Nginx 应用：

5. 【创建应用】 应用类型选择 Mesh 应用。

6. 在左侧导航栏选择【部署组】>【新建部署组】，创建 Node.js 的部署组。

7. 在部署组页面，选择部署组右侧操作列的【部署应用】，选择刚刚创建的镜像，设置部署相关信息。

步骤3：测试 Node.js 服务和 Nginx 服务之间的调用

1. 通过外网 IP 访问测试 Node.js 应用。

2. 通过外网 IP 访问测试前端静态资源。

3. 在 Node.js 服务容器内 curl 访问 Nginx 服务。

```
sudo docker ps #查找容器id
```

```
sudo docker exec -it cfa4343f4a22 /bin/bash #进入容器内部
```

在 Node.js 服务容器内 curl 访问 Nginx 服务成功。

容器服务实例优雅下线

最近更新时间: 2025-02-18 16:02:00

操作场景

当应用在滚动发布时，应用实例下线后向注册中心注销实例信息，服务调用方在监听到实例下线更新本地服务实例列表期间仍会将流量分发到已经下线的服务实例，造成业务异常。因此在应用滚动发布过程中，可以通过调整部署组滚动发布更新配置达到业务无中断优雅下线的目的。

> **注意：**
>
> 目前仅支持集群类型为容器集群对应的部署组应用的服务优雅下线操作。

实现原理

当容器应用进行滚动更新时，provider 实例接受到实例下线通知，向注册中心注销实例信息后实例下线此时实例下线的变更并未推送至 consumer，consumer 会向已经下线的实例正常发起业务调用，造成业务异常。

当启动容器服务滚动发布优雅下线配置后，TSF 会先启动一个 provider 实例向服务注册中心注册实例，而后向下线实例发送下线通知注销实例，待 consumer 将流量路由到新上线实例后在进行原有实例的服务下线，保障滚动发布升级时服务无中断。

操作步骤

1. 登录 TSF 控制台。
2. 在左侧导航栏中，找到【应用中心】分类，单击【应用管理】，进入应用列表页。
3. 在应用列表页，单击应用 ID（或筛选应用），进入应用详情页 。
4. 单击顶部【部署组】，再选择指定的部署组，单击操作列的【部署应用】。
5. 在【资源配置】字段勾选【部署 agent 容器】并设置 agent 容器的 CPU 和内存 request 和 limit。
6. 单击【展开高级设置】，配置滚动发布策略按如下推荐配置启动服务优雅下线能力。

- **配置更新方式**：滚动更新。滚动更新会在应用发布时按照更新策略启动新实例销毁旧实例，在整个发布过程中保证实例数一致。
- **更新间隔**：60秒。等待pod就绪的最小时间设置为60s，保证 pod 实例正常启动并注册到 consul 后，被确认为就绪状态并继续滚动更新。
- **更新策略**：启动新的 pod，停止旧的 pod。选择该策略则先启动新实例完成流量导入后再删除旧实例
- **策略配置**：pods 数为1，选择 pods 数为1则每次更新则先启动一个新实例，再下线一个旧实例。

7. 单击【提交】，即可完成对容器服务实例服务的优雅下线。

> **注意：**
>
> 已经运行的部署组，需要通过配置上述滚动更新发布策略并进行重新部署后 pod 实例才具备优雅下线能力。

微服务网关作为请求入口

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 提供了微服务网关 SDK，并封装了限流、密钥对鉴权、第三方鉴权、设置访问标签等功能，方便您在界面上管理微服务 API。您需要自行准备计算资源（虚拟机或容器），将微服务网关部署、并导入微服务 API。

操作步骤

步骤1：部署微服务网关

1. 创建微服务网关应用 登录 【TSF 控制台】，在左侧导航栏单击【应用管理】，并新建应用，应用类型选择微服务网关应用。

2. 部署微服务网关应用。
3. 在微服务网关应用详情页，单击顶部【程序包管理】>【上传程序包】。 我们使用不包含任何自定义过滤器的 【微服务网关 Demo】，将文件中 msgw - demo 进行打包和上传。
4. 单击顶部【部署组】>【新建部署组】，将网关部署在希望对外暴露微服务 API 的微服务所在的命名空间中

> **注意：**
>
> 提供的 Demo 端口默认为8080，可通过启动参数进行调整。

步骤2：通过微服务网关托管微服务 API

1. 在 【TSF 控制台】 的左侧导航栏中，单击组建中心下的【微服务网关】>【分组管理】。
2. 在分组管理页，单击【新建分组】，并填写分组信息。

- 分组名称：最长为60个字符，只能包含小写字母、数字及分隔符（"_"、"-"），且不能以分隔符开头或结尾。
  - 访问 context：context 是访问网关管理的某一个 API 的路径的路径参数。以"/"开头，不能为空。在这个例子中，添加为 `/myfirsttest`。
  - 鉴权类型：密钥对鉴权或免鉴权。当选择密钥对鉴权时，请求参数中不带正确的 SecretId 和签名的访问会被拒绝。

3. 单击【保存&下一步】，进行网关部署组绑定。
4. 将分组与步骤1中创建的微服务网关应用的部署组进行绑定。

> **注意：**
>
> 此处绑定的是创建的微服务网关应用的部署组，而不是需要对外暴露接口的微服务的部署组。

5. 在【API 列表】中，单击【导入 API】，选择对外暴露的微服务下的 API，完成后单击【提交】。

6. 返回分布管理列表，单击操作列的【发布分组】。发布之后，即可通过访问微服务网关的部署组 + 分组下的 API 访问路径进行访问。

步骤3：设置 API 限流策略

1. 在 【TSF 控制台】 的左侧导航栏中，单击组建中心下的【微服务网关】>【分组管理】，查看分组详情中 API 列表页面。

2. 为已经导入的 API 设置限流。

说明与注意

访问路径相关说明：

微服务网关访问路径为：`微服务网关域名或 IP + port / 分组 contxt / 微服务所在命名空间 / 微服务名称/API path`

示例：创建一个应用类型为微服务网关的应用，部署组中节点 IP 为 10.10.10.10，端口为8080，需要访问命名空间为 default 的微服务 provider，API 路径为 `/echo`，为此创建了一个访问路径为 test 的分组。此时应用的访问路径为 `10.10.10.10:8080/test/default/provider/echo`。

其中，微服务网关的默认端口为8080，您可以通过修改启动参数对网关默认端口进行修改。

为微服务网关所在的部署组配置对外 VIP 的说明：

- 当微服务网关部署在虚拟机上时，您可以将微服务网关所在的部署组的机器挂载在负载均衡实例后面。
- 当微服务网关部署在容器上时，且需要将微服务 API 暴露在公网访问，则在新建容器部署组时，选择公网访问方式。

词汇表

最近更新时间: 2025-02-18 16:02:00

资源管理相关

专有名词	对应英文	解释
集群	Cluster	集群是指云资源管理的集合，包含了运行应用的云主机等资源。集群中的云资源可以是容器、可以是虚拟机等。在同一个集群中，资源之间可以互相通信。
命名空间	Namespace	命名空间是对一组资源和对象的抽象集合。命名空间起到的作用是将同一个集群下的资源进行隔离，使用相同的账号创建不同的环境。在同一个集群中可以有多个命名空间。集群中的资源只能属于一个命名空间。在使用中，用户可以将一个集群隔离为开发环境、测试环境等等。
部署组	Deployment Group	部署组是执行应用批量部署的逻辑概念。一个部署组内包括多个实例，每个实例上运行相同的应用程序。同一个部署组上可以运行一个或者多个应用，但其中的每个实例应用相同。

应用生命周期管理相关

专有名词	对应英文	解释
应用	Application	应用是可部署的软件实体，包含一个或一组容器或进程。
TSF Agent	TSF Agent	TSF Agent 是 TSF 中部署在应用机器上的 Daemon 程序，在运行中承担接收和执行扩容等任务、上报机器运行状态、上报应用监控数据等功能；同时也是 TSF 控制台与应用程序之间通信的桥梁。
TSF 应用生命周期管理	TSF Application Lifecycle Management	应用是 TSF 管理的基本单位，TSF 提供了完整的应用生命周期管理能力，可以完成从部署到运行过程的全程管理，包括应用创建、部署、启动、回滚，扩容缩容和停止下线等操作。

服务框架相关

专有名词	对应英文	解释
微服务	Microservice	微服务是一些协同工作的小而自治的服务，具有弹性扩展、简化部署、可组合等优点。一个微服务既可以是服务提供者，也可以是服务消费者。
Spring Boot	Spring Boot	Spring Boot 是一种用于简化 Spring 应用的初始搭建以及开发过程的框架，该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板式的配置。
Spring Cloud	Spring Cloud	Spring Cloud 是一系列框架的集合，利用 Spring Boot 开发简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等，都可以用 Spring Boot 的开发风格做到一键启动和部署。

数据化运营与监控相关

专有名词	对应英文	解释
IaaS 基础监控	IaaS Basic Monitoring	TSF 监控功能之一，能够针对应用的运行状态，对机器的 CPU、内存、负载、网络和磁盘等基础指标进行详细的监控。
应用监控	Application Monitoring	TSF 的监控功能之一，监控应用的QPS, 请求时间和出错情况等指标。
TSF 分布式跟踪系统	TSF Distributed Tracing System	TSF 调用链跟踪能够分析每一次请求的调用路径，帮助用户了解调用链各阶段耗时情况和状态，从而精准发现系统的瓶颈和隐患。

分布式工具相关

专有名词	对应英文	解释
TSF 分布式配置管理	TSF Distributed Configuration Management	TSF 分布式配置管理实现了将分布式系统的配置信息在 TSF 控制台上进行集中管理，可以实时新增、修改、删除配置，并将更新的配置推送到全局或者应用内部。

常见问题
资源管理相关

最近更新时间: 2025-02-18 16:02:00

云服务器加入容器集群后，项目属性为何变为"默认项目"？

由于加入到 TSF 容器集群中的云服务器 CVM 实例可能属于不同项目，TSF 将加入到容器集群中的云服务器 CVM 的项目属性统一修改为**默认项目**。如果需要集群内云服务器分布在不同的项目，请自行前往云服务器控制台迁移项目。

虚拟机集群内云主机的可用状态为何显示不可用？

虚拟机集群中的云服务器显示"不可用"状态是因为 agent 无法连接 TSF 后台服务器导致的，因此需要检查 agent 的可用状态。

1. 首先检查云服务器的状态是否为"运行中"。如果云服务器的状态不是运行中，请确保云服务器为开机运行状态。
2. 登录云服务器，执行 `ps aux | grep &#39;agent&#39;` 命令查看是否有包含 `tsf-agent` 命名的进程。如果没有，请查看步骤3。
3. 切换到 `/root` 目录下，查看是否有 `tsf_agent` 目录。

- 如果没有，需要将主机从集群中删除后，再重新导入。
- 如果有，进入 `/root/tsf_agent/ops` 目录下，先执行 `uninstall.sh` ，再执行 `install.sh` 命令。然后通过 ps 命令检查进程。

如何手动恢复 tsf-agent 进程？

tsf-agent 有 crontab 做进程检查和恢复，如果 crontab 被停用、卸载、修改会导致 tsf-agent 宕机无法自恢复。手动恢复 tsf-agent 的方法如下：

1. 检查 tsf-agent 的安装目录是否存在，通常在 /root/tsf-agent。
2. 检查 tsf-agent 的安装用户， ls -l tsf-agent/，通常是 root，那以 root 身份到安装目录下执行 ./ops/start.sh。

TSF 容器集群数达到上限如何处理？

默认情况下每个账号在一个地域最多能创建5个容器集群，如需扩大额度，请 [提交工单] 给 **容器服务 TKE** 产品。

TSF 不同集群内的服务是否支持相互调用？

在网络连通性的前提下，同一命名空间内的服务可以相互发现和相互调用。如果要实现跨集群的服务访问，有两个前提：

1. 集群内实例网络互通
2. 集群关联相同的命名空间。多个集群通过命名空间名称（而不是命名空间 ID）作为关联的 key。您可以在命名空间一级页面中看到关联的集群信息。

举例：有2个集群分别是 cls-a 和 cls-b，两个集群内实例网络互通（如在相同 VPC 内），并且都关联了命名空间 test-ns。要实现跨集群访问，需要确保在创建部署组 provider（cls-a集群内）和 consumer（cls-b集群内）时，使用相同的命名空间 test-ns。

TSF 如何设置开发环境和测试环境？

在同一个账号下，您可以通过集群或者命名空间来划分开发环境和测试环境：

- **使用集群划分**：创建两个集群并使用不同的 VPC 作为集群网络，通过网络实现服务之间隔离的作用。使用集群来划分的特点是不同集群不会共享底层的云服务器等资源。
- **使用命名空间划分**：创建两个非全局命名空间，不同命名空间内的服务默认不支持相互调用。

集群网络和容器网络如何设置？

集群网络与容器网络是集群的基本属性。通过设置集群网络和容器网络可以规划集群的网络划分。

- **集群网络**：为集群内主机分配在节点网络地址范围内的 IP 地址，您可以选择私有网络中的子网用于集群的节点网络 。
- **容器网络**： 为集群内容器分配在容器网络地址范围内的 IP 地址，您可以自定义三大私有网段作为容器网络， 根据您选择的集群内服务数量的上限，自动分配适当大小的 CIDR 段用于 kubernetes service，同时容器网络自动为集群内每台云主机分配一个24位的网段用于该主机分配 Pod 的 IP 地址。

CVM 集群应用的日志清理机制是什么？

- 若不设置日志转储，会在30天后自动删除。
- 若选择日志转储至 Ckafka ，则可以存储自定义的时长。具体时长由用户购买的 Ckafka 实例规格决定。

应用管理相关

最近更新时间: 2025-02-18 16:02:00

TSF 是否支持在同一台服务器上安装多个应用？

在 TSF 中，应用的部署有两种类型：

- CVM 云服务器独占实例：在一台 CVM 云服务器上，仅部署单独一个应用。通常根据应用需要的资源配置来购买 CVM 云服务器。
- 容器实例：TSF 使用 Docker 容器在一台独立的 CVM 云服务器上创建多个 Docker 实例，允许在每一个 Docker 实例上部署一个应用。

如何查看实例的 Agent 版本？

1. 在【部署组】页面，单击目标部署组"ID/部署组名"，进入服务实例详情列表。

2. 在目标实例操作栏单击【查看agent版本】即可查看实例的 Agent 版本。

如何将实例 Agent 升级至最新版本？

- 虚拟机部署场景：您需要把实例移出集群后，再重新移入集群，重新部署服务实例。
- 容器部署场景：您需重新编写 dockerfile，生成新的镜像后重新部署服务实例 。

TSF 应用实例状态为什么显示 Agent 异常？

TSF Agent 会定期上报心跳数据给 TSF 管理模块，如果 Agent 停止上报状态，则某段时间后该机器将会被判定为未知状态。通常而言，该问题是由于 Agent 停止导致。

您可以尝试在云服务器界面，重启该云服务器。

重启服务器后 TSF Agent 是否能自动重启？

是的，重启服务器后 TSF Agent、应用都会自动重启。

容器部署组执行部署操作时提示内存（或 CPU）不足？

请检查该部署组所在集群和命名空间中的节点的内存（或 CPU）的使用情况，确保在执行部署操作时填写的内存（或 CPU）数值小于剩余内存（或 CPU）资源 。

您可以在集群的节点列表页面中找到已分配 CPU 和已分配内存信息。

如何排查应用是否部署失败？

1. 在应用详情页，单击部署组操作列的【查看日志】查看 stdout 日志，通过日志初步定位是否是业务程序本身问题。如果没有日志信息，进行步骤2。

2. 单击【变更记录】，查看本次部署任务的 taskid。
3. 登录虚拟机或容器，查看本次任务的日志信息 /root/tsf-agent/agent/task/<taskid> ，其中 taskid 是步骤2中的任务 ID。
4. 您可以通过日志信息初步定位部署失败原因，如果无法排查，可 【提交工单】 反馈任务日志信息。

程序包无法上传如何解决？

当发现程序包无法上传时，请检查浏览器是否设置了代理。您可以尝试换一个浏览器或者切换网络重新进行上传。

创建应用时，提示命名空间数达到配额如何解决？

您需要在容器服务控制台中，选择【镜像仓库】>【我的镜像】，删除超过限额的命名空间。

删除应用后是否可以恢复？

不可以，删除应用操作不可逆，所有的数据都会被清除。

程序包容量达到上限时如何解决？

默认每个租户程序包的存储容量为100GB，如果超过上限，上传程序包时会提示错误信息。此时您需要删除仓库中历史版本的程序包后才能上传新的程序包。

如何选择容器部署组的访问方式？

请参考 【容器应用部署组】 。

将一台已在 CVM 创建的实例添加到 TSF 的时候，是否需要重装 Agent?

需要。由于系统是克隆的（虽然克隆前已安装 Agent），但实例 ID、Local IP 等信息还是之前系统的，导致 Agent 上报实例状态异常，所以克隆后的系统仍需要再次重装 Agent。

Spring Cloud 应用接入相关

最近更新时间: 2025-02-18 16:02:00

TSF 支持哪些 Spring Cloud 版本？

TSF目前支持 Spring Cloud Edgware、Spring Cloud Finchley、Spring Cloud Greenwich 三个版本。

- Spring Cloud Edgwar 对应 Spring Boot 1.5.x
- Spring Cloud Finchley 对应 Spring Boot 2.0.x
- Spring Cloud Greenwich 对应 Spring Boot 2.1.x

TSF 在开源 Spring Cloud 的基础上做了加强，用户只需要替换掉部分依赖并开启注解即可将改造后的应用部署在 TSF 平台上。

在工程的配置文件中，是否需要填写服务注册中心地址？

- 对于本地开发调试的应用，在启动 Java 应用的启动参数中需要填写轻量级服务注册中心 consul 的 IP 和 Port。配置文件（如 application.yml）中则无需填写。
- 对于通过 TSF 平台部署的应用，既不需要在启动参数中设置注册中心的地址，也不需要在配置文件中设置注册中心地址。SDK 会通过环境变量获取注册中心地址。

Mesh 应用相关

最近更新时间: 2025-02-18 16:02:00

应用部署成功后，服务列表中为何没有出现服务？

请检查部署压缩包中是否包含了 `spec.yaml` ，且 `spec.yaml` 的格式是否正确。 如果不存在 `spec.yaml` 或者格式不对，TSF 无法将服务注册到服务注册中心。

服务实例显示离线状态如何解决？

在 Mesh 环境下，TSF Sidecar 会定期通过调用服务的健康检查接口获取服务健康状态，并将健康状态上报到服务注册中心。由于某些原因，例如用户健康检查接口信息配置错误、端口配置错误、或者服务实例出现访问失败，则会导致服务不健康。 您可以通过以下步骤进行排查：

1. 查看服务配置信息

- 服务配置信息错误 查看应用的软件包，获取服务配置信息（spec.yaml），检查服务名是否为期望暴露的服务名、端口号是否为服务真实监听的端口号、健康检查接口是否存在、检查健康接口格式是否正确（不含 ip:port，类似 `/health` 是符合的）。
- 服务配置文件格式错误 将 spec.yaml 内容，复制到 yamllint 中，校验 yaml 格式是否正确。如果格式正确，则继续检查字段名称，是否与下面示例的格式一致。

```
apiVersion: v1
kind: Application
metadata:
name: service1
namespace: nsTester
spec:
services:
- name: user
healthCheck:
path: /health
ports:
- targetPort: 8089
protocol: http
```

- 服务配置没有挂载到正确的位置
- 在容器环境下，排查业务是否在容器的启动脚本中，把 spec.yaml 和 apis 目录（可选），复制到挂载路径 `/opt/tsf/app_config` 下面。
- 在虚拟机环境下，排查业务程序包根目录下面，是否存在 spec.yaml 以及 apis 目录（可选），如果没有，则需要修改。
- 通用排查方法：

1. 检查 pilot-agent 加载配置文件的绝对目录。 输入： `curl 127.0.0.1:15020/config/agent|grep appConfigDir` ， 输出： `appConfigDir: /data/demo/shopService` 代表 pilot-agent 会从 `/data/demo/shopService` 中加载 spec.yaml。
2. 确认 appConfigDir 指向的目录存在 spec.yaml。

2. 查看健康检查返回

登录应用所在的容器或者虚拟机，通过调用本地的健康检查路径，查看返回码是否为200，如果不是200，证明服务存在健康问题。

```
curl -i -H 'Host: local-service' {ip}:{Port}/<healthCheck_path>
```

3. 通过运维接口查看 sidecar 相关信息

登录应用所在的容器或者虚拟机，调用 pilot-agent 的运维接口，查看 sidecar 容器的相关状态信息。

- 查看接口列表 调用 `GET 127.0.0.1:15020/help` 接口查看接口列表：

```
[root@TENCENT64 ~/pilot-agent/op]# curl 127.0.0.1:15020/help
admin commands are:
GET /health: print out the health info for data-plane
GET /config_dump/{component}: print out the configuration of the component, component can be pilot-agent/envoy/mesh-dns
GET /help: print out list of admin commands
GET /config/agent: print out the pilot-agent configuration
GET /config/services: print out the services info
GET /config/global: print out the global mesh configuration
GET /config/envoy: print out the envoy startup configuration
GET /version: print out the pilot-agent version
GET /version/{component}: print out the version of the component, component can be pilot-agent/envoy/mesh-dns
GET /latest_error: print out the latest error info
GET /status: print out the status, result could be <INIT>, <CONFIG_READY>, <FLOW_TAKEOVER>, <SERVICE_REGISTERED>
GET /epoch/{component}: print out the component restart epoch, component can be envoy/mesh-dns
POST /hot_restart/{component}: hot restart the component process, component can be envoy/mesh-dns
PUT /update: Update the component
PUT /logging/{scope}/{level}: dynamic change logging level, scope can be healthz/admin/ads/default/model, level can be info/warn/error/none/debug
```

- 查看版本号 调用 `GET 127.0.0.1:15020/version` 接口查看版本号：

```
[root@TENCENT64 ~/pilot-agent/op]# curl 127.0.0.1:15020/version
1.0.13.release-20190225_103608
```

- 查看 sidecar 健康状态 调用 `GET 127.0.0.1:15020/health` 接口查看 sidecar 健康状态：

```
[root@TENCENT64 ~/pilot-agent/op]# curl 127.0.0.1:15020/health
{"envoy":{"status":"UP"},"mesh-dns":{"status":"UP"},"status":"UP"}
```

如果出现部件不健康的组件（status 不为 UP），或者接口调用失败，则需要通过 [提交工单] 联系后台运维人员处理。

4. 调用 clusters 接口

登录应用所在的容器或者虚拟机，调用 envoy 的 clusters 接口，查看本地 cluster（格式为 in#port#serviceName）是否存在或者健康状态是否 healthy。 输入：

```
curl http://imgcache.finance.cloud.tencent.com:80127.0.0.1:15000/clusters
```

输出：

```
in#8080#reporttimeb::default_priority::max_connections::1024
in#8080#reporttimeb::default_priority::max_pending_requests::1024
in#8080#reporttimeb::default_priority::max_requests::1024
in#8080#reporttimeb::default_priority::max_retries::3
in#8080#reporttimeb::high_priority::max_connections::1024
in#8080#reporttimeb::high_priority::max_pending_requests::1024
```

```
in#8080#reporttimeb::high_priority::max_requests::1024
in#8080#reporttimeb::high_priority::max_retries::3
in#8080#reporttimeb::added_via_api::true
in#8080#reporttimeb::9.77.7.132:8080::cx_active::0
in#8080#reporttimeb::9.77.7.132:8080::cx_connect_fail::0
in#8080#reporttimeb::9.77.7.132:8080::cx_total::4
in#8080#reporttimeb::9.77.7.132:8080::rq_active::0
in#8080#reporttimeb::9.77.7.132:8080::rq_error::0
in#8080#reporttimeb::9.77.7.132:8080::rq_success::4
in#8080#reporttimeb::9.77.7.132:8080::rq_timeout::0
in#8080#reporttimeb::9.77.7.132:8080::rq_total::4
in#8080#reporttimeb::9.77.7.132:8080::health_flags::healthy
in#8080#reporttimeb::9.77.7.132:8080::weight::1
in#8080#reporttimeb::9.77.7.132:8080::region::
in#8080#reporttimeb::9.77.7.132:8080::zone::
in#8080#reporttimeb::9.77.7.132:8080::sub_zone::
in#8080#reporttimeb::9.77.7.132:8080::canary::false
in#8080#reporttimeb::9.77.7.132:8080::success_rate::-1
```

- 如果 cluster 不存在，则需要通过 [提交工单] 联系后台运维人员处理。
- 如果状态不为 healthy，则执行后续步骤。

5. 调用 config_dump 接口

登录应用所在的容器或者虚拟机，调用 envoy 的 clusters 接口，查看本地的路由配置信息：

```
curl http://imgcache.finance.cloud.tencent.com:80127.0.0.1:15000/config_dump -o config.json
```

通过 vi 打开 config.json 文件，并查找 in#8080#reporttimeb（本地 cluster，格式为 in#port#serviceName），查看配置中的服务地址（address）、端口（port_value）是否正确。 健康检查信息 health_checks、熔断配置 circuit_breakers 是否正确。如果不正确，且确认服务没有被熔断，则需要通过 【提交工单】 联系后台运维人员处理。

```
"dynamic_active_clusters": [
{
"version_info": "2018-10-17T11:31:50+08:00",
"cluster": {
"name": "BlackHoleCluster",
"connect_timeout": "1s"
},
"last_updated": "2018-10-16T19:31:41.792Z"
},
{
"version_info": "2018-10-17T11:31:50+08:00",
"cluster": {
"name": "in#8080#reporttimeb",
"connect_timeout": "5s",
"hosts": [
{
"socket_address": {
"address": "9.77.7.132",
"port_value": 8080
}
}
],
"health_checks": [
{
```

```
"timeout": "5s",
"interval": "10s",
"unhealthy_threshold": 2,
"healthy_threshold": 2,
"http_health_check": {
"path": "/health"
}
}
]
```

如何联系后台运维人员？

您可以通过 [提交工单] 的方式联系后台运维人员。 您需要提供以下信息，方便运维人员快速定位问题。

- 问题：B 服务注册失败，已初步定位，原因在于下发的配置与服务配置不相符。
- 服务节点信息：A 调用 B 服务，B 服务192.168.3.1，A 服务192.168.2.1。
- 服务类型信息：A 服务为 springcloud，B 服务为 Mesh。

日志服务相关

最近更新时间: 2025-02-18 16:02:00

为什么无法显示 stdout 日志？

对于使用容器部署的应用，在下述两种情况下 TSF 会采集 stdout 日志：

- **业务容器启动服务进程时不重定向 stdout** 此时可以登录上容器所在的机器，通过 kubectl logs 或者 docker logs 观察程序自身有无输出 stdout
- **业务容器启动服务进程时将 stdout 重定向到** /data/tsf_std/stdout/logs/sys_log.log 此时可以登录上容器所在的机器，通过 kubectl exec 或者 docker exec 进入到容器中，观察上述 sys_log.log 文件有无内容。

为什么无法显示自定义的文件日志？

1. 检查日志配置项中的日志采集规则

- 确认日志配置项中的**日志路径**和应用程序在配置文件中的日志路径相同。
- 确认日志配置项中的**日志解析格式**和应用程序在配置文件中的日志解析格式相同。例如，如果日志配置项为默认日志配置项（Spring Boot 日志格式），而应用程序实际使用 logback 和自定义的日志 pattern，那么 TSF 无法采集日志。用户需要创建一个 logback 格式的日志配置项，然后关联到部署组。

2. 在日志配置项的详情页中，查看日志配置项是否已经发布到对应的部署组中。
3. 日志配置项修改后，虚拟机部署的应用会自动使用修改后的日志配置项来采集日志，容器部署的应用需要重启部署组（先停止部署组，再启动部署组）后才会使用修改后的日志配置项来采集日志。
4. 如果容器镜像的时区不对，则无法检索到日志，需要在 Dockerfile 中调整时区。

```
# GMT+8 for CentOS
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
```

为什么实时日志一直加载不出来？

1. 请确认日志配置项已正确配置。
2. 实时日志会有15s时延，即显示从15秒前开始打印的日志。如果一直在加载中，可以确认下最近15s有没有日志打印，如果没有则会一直到有新的日志打印才会显示。

能否关掉 tsf-task-schedule-xxx 线程或者日志？

这个名称的线程都是 TSF SDK 的 schedule 线程，关闭后会导致部分数据同步功能不正常（如服务治理、配置等）。 如果不需要看到日志，可以将对 package 或 class 的日志级别调高即可。

镜像相关

最近更新时间: 2025-02-18 16:02:00

协作者为何无法推送镜像到主账号的镜像仓库？

主账号需要将协作者与 `QcloudCCRFullAccess` 策略关联 。

默认容器是 UTC 时区，和宿主机的时区不一致如何解决？

保证容器的时区和宿主机一致，避免调用链日志时间收集等有偏差。如果基础镜像时区是没调整过的，那么在编写 `Dockerfile` 时再调整时区，例如基础镜像是 centos，那么微服务 `Dockerfile` 增加以下配置，并重新 build 即可：

```
#GMT+8
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
```

协作者使用镜像仓库时，为何显示未开通仓库？

如果主账号未开通过镜像仓库，会提示如下图所示信息，此时需要主账号登录 TSF 控制台，开通镜像仓库。主账号开通镜像仓库后，协作者/子账号才能继续使用镜像仓库。

JVM 监控相关

最近更新时间: 2025-02-18 16:02:00

JVM 监控功能为何无法使用？

JVM 监控依赖实例上安装的探针。探针版本较低时 JVM 能力不可用。您可以通过 如下操作 升级探针版本。

JVM 如何升级 tsf-agent？

升级操作：

- VM 场景：您需要把实例移出集群后，再重新移入集群，重新部署服务实例。
- 容器场景：您需重新编写 dockerfile，生成新的镜像后重新部署服务实例 。

另外， JVM 监控功能依赖 javaagent，jdk1.5 以后才引入了 javaagent 技术，所以您需使用 jdk1.6 及以上版本，建议您使用 jdk1.8。

- TencentCloudJvmMonitor-1.1.0监听11339端口，请注意避免冲突。
- 请在新建容器部署组时需要额外加上端口映射。

火焰图采集失败如何处理？

火焰图适用于在 CPU 利用率持续较高的情况下进行热点函数分析。 在如下场景下，火焰图采集可能失败：

- 数据量过大：目前火焰图的数据量上限为2MB，超出时会采集失败；此时，请缩短所选择的采集时间后，重新进行采集。
- 当前无热点函数：进程处于低 CPU 消耗状态，即当前无热点函数时，火焰图采集可能失败（无法采集到热点函数）；此时，可尝试延长采集时间后重新采集，或待服务请求量较大时再重新采集。
- 无法和进程建立连接：发生无法和实例连接等异常情况时，采集任务会执行超时；此时，请您检查实例的连接状态，确认状态正常后再重新采集。

为何无法查看 JVM 日志？

JVM 日志的采集依赖于 TSF 的 GC 日志配置。如果您自定义了 GC 日志配置，将会覆盖 TSF 的 GC 日志配置，这将导致无法在界面中查看 JVM 日志。

其他问题

最近更新时间: 2025-02-18 16:02:00

如何开通 TSF？

如果需 TSF，请先做以下准备工作：

1. 主账号在 【TSF 控制台概览页】 或者 【CAM 控制台】创建服务角色。
2. 如果用户是子账号或者协作者，还需要主账号创建 PassRole 策略，并将策略绑定到子账号或协作者。

start.sh 的作用是什么？

Service Mesh 应用部署使用 tar.gz 、zip 压缩格式的程序包，在压缩包中除了应用程序外，还需要包含了 start.sh、stop.sh、cmdline 三个文件：

- start.sh：用于启动应用程序
- stop.sh：用于停止应用程序
- cmdline：用于检查应用进程是否存在，**没有** `.sh` **后缀**

TSF 能单独输出部署到公司内网吗？

可以，请 [提交工单] ，会有专门的技术支持人员联系您。

JVM 内存不足如何处理？

JVM 内存不足（java.lang.OutOfMemoryError）时，需要您手动设置内存大小，具体设置方法请在互联网中搜索"JVM 内存设置"。

TSF 支持哪些编程语言？

- 若使用 Spring 接入，则需使用 Java 语言。
- 若通过 Mesh 接入，则不限制编程语言。TSF Mesh 支持不同语言实现的微服务之间相互调用，并支持不同框架如 Mesh 和 Spring Cloud 运行微服务相互调用。

功能和概念相关

最近更新时间: 2025-02-18 16:02:00

TSF 有哪些功能？

TSF 提供应用全生命周期管理、数据化运营、立体化监控和服务治理等功能。TSF 拥抱 Spring Cloud 、Service Mesh 微服务框架，帮助企业客户解决传统集中式架构转型的困难，打造大规模高可用的分布式系统架构，实现业务、产品的快速落地。

TSF 有哪些版本？

TSF 有基础版、专业版和铂金版，不同版本支持的功能和服务会有差异，对应的价格也会不同。

- 基础版：包括基本的应用生命周期管理、应用监控、调用链与日志等功能。
- 专业版：包含了基础版的所有功能，同时支持服务限流、路由、熔断等服务治理功能。
- 铂金版：包含了基础版和专业版的所有功能，同时支持微服务网关、全链路灰度等高级能力。

TSF 与 TKE 的关系是什么？

TKE（Tencent Kubernetes Engine）是基于原生 Kubernetes 提供以容器为核心的、高度可扩展的高性能容器管理服务。TSF 是以微服务为核心的服务治理平台，用户可以使用云服务器或者容器来部署微服务，其中容器集群管理和容器应用部署使用了 TKE 提供的服务。

应用和服务之间是什么关系？

在 TSF 中应用是管理一组程序包、镜像、配置的抽象概念，服务表示已经注册到注册中心的程序。可以从以下维度来区分应用和服务：

概念	程序类型	主要功能	与部署组的关系
应用	无特定类型	程序包版本管理、配置版本管理	部署组使用应用中的程序包和配置
服务	注册到注册中心的微服务	服务注册、服务限流、服务路由等服务治理功能	在相同命名空间下具有相同服务名的部署组属于一个服务

TSF 支持 Sentinel 流控相关的组件应用吗?

暂不支持

协作者子用户使用相关

最近更新时间: 2025-02-18 16:02:00

主账号发生无权限访问如何解决？

主账号在使用 TSF 功能时提示无权限错误，此时需要主账号授权 TSF 创建服务角色 TSF_QCSRole 。

子账号身份使用 TSF 发生无权限报错如何解决？

子账号在使用 TSF 前，需要主账号将子账号关联 `tsf_PassRole` 策略 。

使用控制台时，提示“当前账号尚未获得操作权限[action]，请联系主账号授权”如何解决？

您可根据提示的 action 判断：

- 是否是 TSF 产品内的权限。若出现明显的 CVM、TKE、VPC 字样，则需要主账号在 【CAM 策略】 中，配置相关读写权限。
- 若根据 action 语义判断，属于 TSF 内的功能权限。您可参考 [角色管理] ，申请主账号为您配置相应权限点。

子账号使用 TSF 时，弹窗提示 “您没有权限执行此操作”如何解决？

问题描述： 子账号使用 TSF 平台时，可能报错 “您没有权限执行此操作”，显示如下：

问题分析： 在使用 TSF 平台时，需要调用其他云产品的接口获取一些信息，包含 VPC、CVM、API 网关、镜像仓库、云监控、TKE 等云产品。

解决方法： 建议对子账号授权 tsf_PassRole 策略，如果没有，需要主账号或者具有 QcloudCamRoleFullAccess 策略的用户创建角色。

子账号使用 TSF 时，需要主账号授予 tsf_PassRole 策略。

要将角色（及其许可策略）传递至 TSF 服务，用户必须具有**传递角色**至服务的许可。这有助于管理员确保仅批准的用户可配置具有能够授予许可的角色的服务。

最终子账号需被授予如下策略：

策略	是否必选	说明
tsf_PassRole	必选	手动创建。
QcloudCamSubaccountsAuthorizeRoleFull	可选	访问管理（CAM）子账号授权服务角色相关权限，包含子账号在授权服务角色过程中涉及的全部权限。
QcloudTSFFullAccess	可选	微服务平台（TSF）全读写访问权限。
QcloudCCRFullAccess	可选	与镜像仓库相关，如果需要使用 TSF 中容器相关功能需要授权。
QcloudVPCReadOnlyAccess	可选	如果需要读取集群 VPC 等信息需要授权。

如担心策略范围过大，可参考访问管理的 【排除故障】 文档对具体接口进行授权。

开发手册
TSF Mesh 指南
Mesh Demo工程概述

最近更新时间: 2025-02-18 16:02:00

下载 Demo

Demo 语言	下载地址	说明
Python Demo（vm）	tsf_python_vm_demo	-
Python Demo（docker）	tsf_python_docker_demo	-
.NET Demo（vm & docker）	tsf_mesh_demo_dotnet	其中 REAME.md 介绍了程序包和镜像两种构建方式
Java Demo（vm）	tsf_mesh_demo_java	-
PHP Demo（vm）	tsf_php_vm_demo	-
Dubbo Demo（vm）	tsf_mesh_dubbo_vm_demo	-

> **注意：**
>
> Mesh 的部署和使用与语言无关，具体可参考 Python 使用方式进行改造。

调用说明

下文以 Python Demo 为例进行介绍。Python Demo 提供了3个应用，对应的服务名和应用监听端口为：

- user （8089）
- shop （8090）
- promotion （8091）

3个应用之间的调用关系是：`user -&gt; shop -&gt; promotion`，相互访问时可以用默认的80或者业务的真实端口（对应 Demo 中的 sidecarPort），如 shop 监听8090，user 访问 shop 可以用 `shop:80/api/v6/shop/items` 或者 `shop:8090/api/v6/shop/items`。

> **注意：**
>
> Mesh 的调用链通过头传递实现。如果用户想要串联整个服务调用关系，需要在访问其他服务时，带上父调用的9个相关调用链头，具体示例如下：

```
// 9个调用链相关的头，具体说明见(http://imgcache.finance.cloud.tencent.com:80www.envoyproxy.io/docs/envoy/v1.8.0/configuration/http_conn_man/headers.html?highlight=tracing)
traceHeaders = ['x-request-id',
'x-trace-service',
'x-ot-span-context',
'x-client-trace-id',
```

```
'x-b3-traceid',
'x-b3-spanid',
'x-b3-parentspanid',
'x-b3-sampled',
'x-b3-flags']

// 填充调用链相关的头
def build_trace_headers(handler):
retHeaders = {}
for header in traceHeaders:
if handler.headers.has_key(header):
header_value = handler.headers.get(header)
retHeaders[header] = header_value
return retHeaders

// 访问 shop 服务的端口，使用默认的80，或者 shop 的真实端口8090
sidecarPort = 80
def do_GET(self):
// 调用 shop 服务时填充父调用的调用链相关头
if self.path == '/api/v6/user/create':
print "headers are %s" % self.headers.keys()
logger.info("headers are %s" % self.headers.keys())
headers = common.build_trace_headers(self)
if common.sendAndVerify("shop", sidecarPort, "/api/v6/shop/items", headers):
self.send_response(200)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"result":{"userId":"1234", "userName":"vincent"}}
self.wfile.write(json.dumps(msg))
else:
self.send_response(500)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"exception":"Error invoke %s" % "/api/v6/shop/items"}
self.wfile.write(json.dumps(msg))
```

Spring Cloud 应用和 Mesh 应用调用打通 tracing

Spring Cloud 应用和 Mesh 应用相互调用时，如果要打通 tracing，需要在请求中传递调用链相关的 header（参考 上文）。

工程目录

虚拟机工程目录

以 tsf_python_vm_demo 中的 userService 为例说明虚拟机应用工程目录。

- **userService.py 和 common.py**：Python 应用程序
- **start.sh**：启动脚本
- **stop.sh**：停止脚本
- **cmdline**：检查进程是否存活的文件
- **spec.yaml**：服务描述文件
- **apis 目录**：存放 API 定义的目录

其中 star.sh、stop.sh、cmdline 的编写方法请参考 【上传程序包要求】。

容器应用工程目录

以 tsf_python_docker_demo 中的 demo-mesh-user 为例说明容器应用工程目录。

> **注意：**
>
> 您需要在容器启动后通过用户程序的启动脚本拷贝目录，不可以在 Dockerfile 中提前拷贝。

- **Dockerfile**：使用 userService 目录中 start.sh 脚本来启动 Python 应用。
- **userService**目录：基本结构类似 tsf_python_vm_demo 中 userService 目录，但是没有 stop.sh 和 cmdline 文件。
- **start.sh**：应用的启动脚本，user demo 的启动脚本如下：

```
#! /bin/bash
cp /root/app/userService/spec.yaml /opt/tsf/app_config/
cp -r /root/app/userService/apis /opt/tsf/app_config/
cd /root/app/userService/
python ./userService.py 80 1>./logs/user.log 2>&1
```

脚本说明：

- 应用工作目录为 `/root/app/userService/` ，应用日志目录为 `/root/app/userService/logs/user.log` 。
- 第2行：创建 `/opt/tsf/app_config/apis` 目录，并将 `spec.yaml` 文件拷贝到 `/opt/tsf/app_config/` 中。
- 第3行：将 `apis` 目录拷贝到 `/opt/tsf/app_config/` 中。
- 第5行：启动 user 应用。

TSF Mesh 概述

最近更新时间: 2025-02-18 16:02:00

Mesh 微服务平台（Tencent Service Mesh Framework，以下简称 TSF Mesh）是一个基础设施层，用于处理服务间的通信。TSF Mesh 是由一系列轻量级的网络代理组成，这些代理（又称 Sidecar）与应用程序部署在一起，而应用程序不感知 Sidecar 的存在。 TSF Mesh 是处于 TCP/IP 之上的一个抽象层。TCP 解决了网络端点间字节传输问题，TSF Mesh 解决服务节点间请求的路由问题。

以下视频将为您介绍 TSF Mesh 的基本架构和优势：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2037-24358?source=gw.doc.media&withPoster=1&notip=1

TSF Mesh 优势

TSF Mesh 具有如下优势：

- 多编程语言应用兼容。
- 业务代码零侵入，代码无须改造。

实现原理

TSF Mesh 可以代理使用云服务器或者容器部署的应用。下面以容器为例说明 TSF Mesh 的实现原理。Sidecar 是 L7 层代理，和服务运行在同一个 Pod 中，与 Pod 共享网络，其中 Sidecar 与服务的关系如下：

- Sidecar 代理服务向注册中心注册服务相关信息，以便其他服务发现自身。
- Sidecar 作为 Pod 内服务的 HTTP 代理，可以自动发现其他服务。

使用场景

TSF Mesh 主要有三种使用场景：

- 仅服务消费者作为 Mesh 应用部署。
- 仅服务提供者作为 Mesh 应用部署。
- 服务消费者和服务提供者均作为 Mesh 应用部署。

场景1：仅服务消费者作为 Mesh 应用部署

- 服务提供者使用 TSF-Spring Cloud 框架实现，注册到服务注册中心；
- 服务消费者作为 Mesh 应用部署，由 Sidecar 注册到服务注册中心。

场景2：仅服务提供者作为 Mesh 应用部署

- 服务提供者作为 Mesh 应用部署，由 Sidecar 注册到服务注册中心；
- 服务消费者使用 TSF-Spring Cloud 框架实现，注册到服务注册中心。

场景3：服务消费者和服务提供者均作为 Mesh 应用部署

- 服务提供者作为 Mesh 应用部署，由 Sidecar 注册到服务注册中心；
- 服务消费者作为 Mesh 应用部署，由 Sidecar 注册到服务注册中心。

应用部署（容器场景）

最近更新时间: 2025-02-18 16:02:00

操作场景

本文以容器应用为例，指导您通过 TSF 控制台，创建并部署 Mesh 应用、查看服务是否注册成功、验证服务调用。

准备工作

1. 下载 TSF 提供的 Python Mesh Demo（docker）（该步骤预计耗时1min）。
2. 解压 Demo 压缩包，分别进入二级目录（如 `demo-mesh-promotion` 目录），执行 `docker build` 命令制作容器镜像。
3. 在 TSF 控制台上已创建容器集群并添加节点，详情参考 [集群]（对于未创建容器集群和添加节点的用户，该步骤预计耗时10min）。
4. 用户的开发机上已安装 `docker` 环境（用于推送镜像到镜像仓库。对于本地无 `docker` 环境的用户，该步骤预计耗时20min）。

操作步骤

步骤1：创建并部署 Mesh 应用

1. 创建应用

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏，单击【应用管理】，进入应用列表页。
3. 在应用列表页，单击【新建应用】，并填写以下信息：
   - 应用名：填写应用名称
   - 部署方式：选择**容器部署**
   - 应用类型：选择 **Mesh 应用**
   - 服务配置：可以选择【使用本地Spec.yaml】（默认方式）或者【控制台配置】。 若选择【控制台配置】，需要填写以下信息：

```
-
- 服务名：最长60个字符，只能包含字母、数字和分隔符（“-”），且不能以分隔符开头或结尾。
- 监听端口：只有1个（接口使用数组类型）。协议：HTTP/HTTP2/gRPC/Dubbo；端口范围 1~65535。
- 心跳检查接口：校验逻辑，以 / 开头。不超过200个字符。
- 标签：选择已有标签或者前往标签管理创建。
```

4. 单击【提交】，完成应用创建。

2. 将镜像推送到仓库

1. 在左侧导航栏，单击【镜像仓库】，进入镜像列表页。首次使用时，您需要设置镜像仓库密码。
2. 在镜像列表页，单击【应用管理】>【ID/应用名】>【镜像】，单击【使用指引】，根据指引中的命令将 Python demo 应用的镜像（参考 准备工作 中的第2步）推送到镜像仓库中 。

3. 创建部署组

1. 在应用详情页内，单击【[部署组] 】，进入部署组列表页。
2. 在部署组列表页，单击【新建部署组】，并填写以下信息：
   - 部署组名：填写部署组名称
   - 集群：选择应用将部署的集群
   - 命名空间：选择命名空间属性
   - 实例资源限制：0.5核，512MiB
   - 实例数：1
3. 设置访问设置、更新方式、日志配置项。
   - 网络访问方式：NodePort
   - 端口协议：协议选择 TCP，容器端口和服务端口填写相同的数值（user：8089，shop：8090，promotion：8091）
   - 更新方式：快速更新
   - 日志配置项：选择无
4. 单击【提交】，完成部署组创建。

4. 部署应用

1. 在部署组列表页，单击部署组右侧的【部署应用】。
2. 部署相关信息，使用 【步骤2】 中仓库中的镜像 。
3. 单击【提交】，部署组状态变为运行中，则表示应用部署成功。

步骤2：查看服务是否注册成功

1. 在左侧导航栏，单击【服务治理】，进入服务列表页，查看服务是否注册成功。如果注册成功，服务显示在线状态。
2. 在服务列表页，单击服务 ID，进入服务详情页。单击【API 列表】标签页，可以查看上报的 API 定义。

步骤3：验证服务调用

使用同样的步骤部署 user、shop 和 promotion 三个应用。user、shop、promotion 三个服务的接口间调用关系如下：

1. 触发 user 服务调用 shop 和 promotion 服务

为了验证 user 服务能通过服务名来调用 shop 服务，需要用户通过以下三种方式来触发 user 服务的接口调用：

- **负载均衡 IP + 服务端口**：如果部署组在部署时，选择了公网访问方式，可以通过**负载均衡 IP + 服务端口**来访问 `user` 服务的 `/api/v6/user/account/query` 接口。
- **云主机 IP + NodePort**： 如果部署组在部署时，选择了 NodePort 访问方式，可以通过**云主机 IP + NodePort**来访问 `user` 服务的 `/api/v6/user/account/query` 接口。其中 `云主机 IP` 为集群中任一云主机的内网 IP， `NodePort` 可以在部署组的基本信息页面被查看。用户首先登录到集群所在 VPC 的机器，然后执行如下命令：

```
curl -XGET <云主机 IP>:<NodePort>/api/v6/user/account/query
```

- **API 网关**：用户可以通过在 API 网关配置微服务 API 来调用 user 服务的接口。关于如何配置微服务 API 网关

2. 在控制台验证服务之间是否调用

可以通过以下两种方式验证服务是否成功被 Sidecar 代理注册到注册中心，同时验证服务之间是否成功地进行了调用。

- **服务治理**界面：选择集群和命名空间后，如果服务列表中的服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，表示服务提供者被服务消费者请求成功。

- **依赖拓扑**界面：选择集群和命名空间后，调整时间范围，使其覆盖服务运行期间的时间范围，正常情况下，将出现服务之间相互依赖的界面。

应用部署（虚拟机场景）

最近更新时间: 2025-02-18 16:02:00

操作场景

本文以虚拟机应用为例，指导您通过 TSF 控制台，创建并部署 Mesh 应用、查看服务是否注册成功、验证服务调用。

准备工作

1. 下载 TSF 提供的 Python Mesh Demo（vm）（该步骤预计耗时1min）。
2. 解压 Demo 压缩包，解压出三个压缩包分别为 `promotionService.tar.gz`、`shopService.tar.gz`、`userService.tar.gz`（不需要再解压）。
3. 在 TSF 控制台上已创虚拟机集群并添加节点，参考 【集群】 。对于未创建虚拟机集群和添加节点的用户（该步骤预计耗时10min）。
4. 主机上已安装应用运行的环境（如 Python 应用的相关依赖等，TSF 对相关依赖的版本没有限制，该步骤预计耗时根据运行环境的复杂度有所不同）。

操作步骤

步骤1：创建并部署 Mesh 应用

1. 创建应用

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏单击【应用管理】，进入应用管理列表页。
3. 在应用列表页，单击【新建应用】，并填写以下信息：

- 应用名：填写应用名称
- 部署方式：选择**虚拟机部署**
- 应用类型：选择 **Mesh 应用**

- 服务配置：可以选择【使用本地Spec.yaml】（默认方式）或者【控制台配置】。 若选择【控制台配置】，需要填写以下信息：
- 服务名：最长60个字符，只能包含字母、数字和分隔符（“-”），且不能以分隔符开头或结尾。
- 监听端口：只有1个（接口使用数组类型）。协议：HTTP/HTTP2/gRPC/Dubbo；端口范围 1~65535。
- 心跳检查接口：校验逻辑，以 / 开头。不超过200个字符。
- 标签：选择已有标签或者前往标签管理创建。

4. 单击【提交】，完成应用创建。

2. 上传程序包

1. 在左侧导航栏单击【应用管理】，选择某一应用的【ID/应用名】，进入应用服务详情页。
2. 在应用服务详情页单击【程序包管理】标签页。
3. 在标签页单击【上传程序包】，选择程序包（如 `promotionService.tar.gz` ），填写程序包相关信息。
4. 单击【提交】，完成上传。

3. 创建部署组

详细操作 【虚拟机应用部署组】 中关于**创建部署组**的内容。

4. 部署应用

详细操作 【虚拟机应用部署组】 中关于**部署应用**的内容。

步骤2：查看服务是否注册成功

1. 在左侧导航栏单击 【服务治理】，进入服务列表页，查看服务是否注册成功。如果成功，服务显示在线状态。
2. 在服务列表页单击服务 ID，进入服务详情页。单击【API 列表】标签页，可以查看上报的 API 定义。

步骤3：验证服务调用

使用同样的步骤部署 user、shop 和 promotion 三个应用。user、shop、promotion 三个服务的接口间调用关系如下：

用户可以登虚拟机集群 VPC 下的任一机器，然后通过 `curl` 命令验证 user 服务是否健康，以及触发 user 服务调用 shop 和 promotion 服务。

1. 登录服务器验证服务间调用

为了验证 user 服务能通过服务名来调用 shop 服务，需要用户通过以下几种方式来触发 user 服务的接口调用：

- 登录 user 所在云服务器，执行如下 `curl` 命令调用 user 服务接口。

```
curl localhost:<user端口>/api/v6/user/account/query
```

- 登录 user 所在云服务器，执行如下 `curl` 命令调用 shop 服务接口（注意使用服务名来调用）。

```
curl shop:<shop端口>/api/v6/shop/order
```

- **API 网关**：用户可以通过在 API 网关配置微服务 API 来调用 user 服务的接口 。

2. 在控制台验证服务之间是否调用

可以通过以下两种方式验证服务是否成功被 Sidecar 代理注册到注册中心，同时服务之间是否成功地进行了调用。

- **服务治理**界面：选择集群和命名空间后，如果服务列表中的服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，表示服务提供者被服务消费者请求成功。
- **依赖拓扑**界面：选择集群和命名空间后，调整时间范围，使其覆盖服务运行期间的时间范围，正常情况下，将出现服务之间相互依赖的界面。

开发使用指引

最近更新时间: 2025-02-18 16:02:00

开发说明

本文将以 Python 应用为例说明如何改造代码来接入 TSF。您不需要修改 Python 服务代码，只需要修改服务间调用的 host。

- 将原来的 IP:Port 替换为服务名，如果不使用服务名调用，流量不会经过 sidecar 直接传递到被调服务，被调服务无法识别主调服务的服务名。
- 端口使用80或者业务真实的监听端口。
- 其他代码不做修改。

> **注意：**
>
> 以下代码片段可参考 Demo 工程内 userService.py。

改造前：

```
sidecarPort = 80
if common.sendAndVerify("127.0.0.1", sidecarPort, "/api/v6/shop/items", headers):
self.send_response(200)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"result":{"userId":"1234", "userName":"vincent"}}
self.wfile.write(json.dumps(msg))
else:
self.send_response(500)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"exception":"Error invoke %s" % "/api/v6/shop/items"}
self.wfile.write(json.dumps(msg))
```

改造后：

```
sidecarPort = 80
if common.sendAndVerify("shop", sidecarPort, "/api/v6/shop/items", headers):
self.send_response(200)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"result":{"userId":"1234", "userName":"vincent"}}
self.wfile.write(json.dumps(msg))
else:
self.send_response(500)
self.send_header('Content-type', 'application/json')
self.end_headers()
msg = {"exception":"Error invoke %s" % "/api/v6/shop/items"}
self.wfile.write(json.dumps(msg))
```

可以看到，代码行中除了访问方式发生变化（ `127.0.0.1` 变为 `shop` ）外，其他都不需要改动。

服务定义和注册（必选）

如果是虚拟机部署，需要在应用程序所在目录中设置创建 spec.yaml 文件；**如果是容器部署**，需要在应用启动时，在 /opt/tsf/app_config 下写入 spec.yaml 文件，该文件用于描述服务信息。Sidecar 会通过服务描述文件将服务注册到服务注册中心。

> **注意：**
>
> 当前支持在控制台上选择【使用本地spec.yaml】和【控制台配置】两种方式描述服务信息，推荐在控制台上直接设置。

若选择使用本地 spec.yaml 文件，spec.yaml 格式如下：

```
apiVersion: v1
kind: Application
spec:
services:
- name: user # 服务名
ports:
- targetPort: 8091 # 服务监听端口
protocol: http # 目前支持 HTTP、HTTP2 和 gRPC
healthCheck:
path: /health # 健康检查 URL
```

> **注意：**
>
> - healthCheck 是健康检查的接口，请确认本地调用 curl -i -H 'Host: local-service' {ip}:{Port}/health 能返回200，否则，健康检查失败会导致此服务实例变为离线状态，其它服务将无法调用该服务实例；如果不提供此健康检查接口，sidecar 会通过 TCP 的方式探测 targetPort 是否连通来判断此服务实例是否健康。
> - Host: local-service 是代理加的 header，业务如果对 Host 有检查（如 Nginx 配置的 server_name），则需将 local-service 加到白名单。

服务间调用方式

使用服务名调用

在 user 服务所在实例上执行 curl 命令，通过使用 shop 服务名进行访问。

```
curl shop:<shop端口>/api/v6/shop/order
```

API 定义和上报（可选）

TSF 支持 Mesh 应用 API 上报功能，用于 API 级别的服务治理，如路由、鉴权和限流等，不需要可以跳过。

- **如果是虚拟机部署**，需要在应用程序所在目录中创建 apis 目录。
- **如果是容器部署**，需要在 /opt/tsf/app_config 下创建 apis 目录，该目录放置服务的 API 定义。

一个服务对应一个 yaml 文件，文件名即服务名，如 petstore 服务对应 petstore.yaml 配置。API 遵循 【OPENAPI 3.0 规范】。配置文件遵循 【样例参考】。user.yml 的 API 定义如下：

```
openapi: 3.0.0
info:
version: "1.0.0"
title: user service
paths:
/api/v6/user/create:
get:
responses:
'200':
description: OK
'401':
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
/api/v6/user/account/query:
get:
responses:
'200':
description: OK
'401':
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
/health:
get:
responses:
'200':
description: OK
'401':
description: Unauthorized
'402':
description: Payment Required
'403':
description: Forbidden
```

设置自定义标签（可选）

Mesh 支持通过 HTTP Header 设置自定义标签（标签可用于服务治理 ）。 以 Python 应用为例说明如何设置自定义标签。

```
>>> import requests
>>> url = 'http://imgcache.finance.cloud.tencent.com:80api.github.com/some/endpoint'
>>> headers = headers = {'tsf-mesh-tag': 'custom-key=custom-value'}
>>> r = requests.get(url, headers=headers)
```

> **注意：**
>
> 以上示例已经在依赖机器上安装了 requests 库。

调用链 Header 传递（可选）

要实现 Mesh 应用调用链和服务依赖拓扑功能，需要在请求中带上9个相关 header。

```
// 9个调用链相关的头，具体说明见(http://imgcache.finance.cloud.tencent.com:80www.envoyproxy.io/docs/envoy/v1.8.0/configuration/http_conn_man/headers.html?highlight=tracing)
traceHeaders = ['x-request-id',
'x-trace-service',
'x-ot-span-context',
'x-client-trace-id',
'x-b3-traceid',
'x-b3-spanid',
'x-b3-parentspanid',
'x-b3-sampled',
'x-b3-flags']
```

更新服务配置

最近更新时间: 2025-02-18 16:02:00

操作场景

当 Mesh 应用的服务配置选择【控制台配置】方式时，可以在应用基本信息页中更新服务配置；如果是使用 spec.yaml，则不显示服务配置信息。

该任务指导您在 TSF 控制台更新 Mesh 应用服务配置。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏选择【应用管理】。
3. 选择目标应用，单击应用的"ID/应用名"，进入应用【基本信息】页。

4. 在【服务配置】模块，选择【编辑】，修改服务配置信息。

5. 单击【提交】后，在弹出框中选择【确认】去重启部署组。

> **注意：**
>
> 更新服务配置后，需要重启部署组才会生效。

6. 在部署组列表页，在部署组状态栏单击【重启】弹出重启操作对话框。

> **注意：**
>
> - 更新服务配置后，对于**运行中**状态的部署组，如果部署组的服务配置和应用的服务配置信息不一致可重启部署组进行更新。
> - 当 Mesh 应用（且使用虚拟机部署）的部署组内存在实例的 agent 版本较低时，不支持【服务配置】特性，请升级 agent 版本。

7. 在重启应用对话框中，单击【提交】后，可看到部署组的运行状态变为正常。

运维接口说明

最近更新时间: 2025-02-18 16:02:00

Service Mesh 微服务架构的核心是在用户的服务侧同机（虚拟机）或同 Pod（容器）部署一个网络代理来让用户实现无侵入的接入微服务。因为代理是以本地额外的服务实现的，本地应用无感知，为了快速定位出现的问题，代理侧提供了大量的 HTTP 运维接口。

代理的组件

本地代理分为以下三个组件：

- **pilot-agent** 管理整个代理侧环境，如 iptables 规则的更新、本地应用和服务的配置管理，同时负责 mesh-dns 和 envoy 组件的生命周期管理。
- **Mesh-DNS** 托管本地的 DNS 请求，将 Mesh 结构内的流量导入本地代理。
- **Envoy** 负责服务发现和路由，HTTP 请求的负载均衡等，负责处理流入本地服务和从本地服务流出的所有流量。

iptables 规则

为了将流入和流出的流量导入到本地代理中，TSF Mesh 使用了 iptables 作流量转发，一般规则如下（虚拟机环境在本地，容器环境在 sidecar 容器中）：

iptables -t nat -L -n

```
Chain PREROUTING (policy ACCEPT)
target prot opt source destination
ISTIO_INBOUND tcp -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
ISTIO_OUTPUT all -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800
DNS_OUTPUT all -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800

Chain POSTROUTING (policy ACCEPT)
target prot opt source destination

Chain DNS_OUTPUT (1 references)
target prot opt source destination
RETURN all -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800 owner UID match 1000
DNAT udp -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800 udp dpt:53 to:127.0.0.1:55354
DNAT tcp -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800 tcp dpt:53 to:127.0.0.1:55354

Chain ISTIO_INBOUND (1 references)
target prot opt source destination
ISTIO_REDIRECT tcp -- 0.0.0.0/64295985640140800 9.77.7.28 tcp dpt:8089

Chain ISTIO_OUTPUT (1 references)
target prot opt source destination
RETURN all -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800 owner UID match 1000
```

```
DNAT tcp -- 0.0.0.0/64295985640140800 {特定 IP} to:9.77.7.28:15001

Chain ISTIO_REDIRECT (1 references)
target prot opt source destination
REDIRECT tcp -- 0.0.0.0/64295985640140800 0.0.0.0/64295985640140800 redir ports 15001
```

DNS_OUTPUT

- 托管本地的 DNS 请求到 mesh-dns 进程。
- 第一条 RETURN 规则非常重要，不再托管从 mesh-dns 转发出来的 DNS 请求。
- 后面两条是托管本地的53端口（即 DNS 原生请求端口）的流量。

ISTIO_INBOUND

- 托管访问本地服务的流量。
- 可以看到托管的流量非常细粒度，只托管注册到 Mesh 框架的 `9.77.7.28:8089` 的流量。

ISTIO_OUTPUT

- 托管从本地流出的流量。
- 第一条 RETURN 规则非常重要，不再托管从代理转发出来的请求。
- 第二条是将访问{特定 IP} - 从 DNS 中获得的流量导入到本地的代理中（即引用的 ISTIO_REDIRECT 规则）。

本地接口

如果在本地调用 `netstat -lntp | grep 127.0.0.1` ，会出现以下三个 listen 服务，分别是各个本地组件提供的运维接口的 IP 和 port。

```
tcp 0 0 127.0.0.1:15020 0.0.0.0:* LISTEN 5168/pilot-agent
tcp 0 0 127.0.0.1:15021 0.0.0.0:* LISTEN 5261/mesh-dns
tcp 0 0 127.0.0.1:15000 0.0.0.0:* LISTEN 5266/envoy
```

pilot-agent 运维接口

curl 127.0.0.1:15020/help

```
admin commands are:
GET /health: print out the health info for data-plane
GET /config_dump/{component}: print out the configuration of the component, component can be pilot-agent/envoy/mesh-dns
GET /help: print out list of admin commands
GET /config/agent: print out the pilot-agent configuration
GET /config/services: print out the services info
GET /config/global: print out the global mesh configuration
GET /config/envoy: print out the envoy startup configuration
GET /version: print out the pilot-agent version
GET /version/{component}: print out the version of the component, component can be pilot-agent/envoy/mesh-dns
GET /latest_error: print out the latest error info
GET /status: print out the status, result could be <INIT>, <CONFIG_READY>, <FLOW_TAKEOVER>, <SERVICE_REGISTERED>
GET /epoch/{component}: print out the component restart epoch, component can be envoy/mesh-dns
POST /hot_restart/{component}: hot restart the component process, component can be envoy/mesh-dns
PUT /update: update the component
```

```
PUT /logging/{scope}/{level}: dynamic change logging level, scope can be healthz/admin/ads/default/model, level can be debug/info/warn/error/none
```

其中主要的接口如下：

- status：查看本地各个组件的状态。
- health：查看本地各个组件的健康信息。
- version：查看本地代理服务的版本。

Mesh-DNS 运维接口

curl 127.0.0.1:15021/help

```
admin commands are:
GET /health: print out the health info for mesh-dns
GET /config_dump: print out all of the mesh-dns runtime configs
GET /latest_error: print out the latest error info
GET /help: print out list of admin commands
GET /version: print out version info
GET /status: print out status info
PUT /logging/{scope}/{level}: dynamic change logging level, scope can be admin/default/healthz/model, level can be error/none/debug/info/warn
```

其中主要的接口如下： config_dump：查看本地托管的 DNS 信息。

Envoy 运维接口

curl 127.0.0.1:15000/help

```
admin commands are:
/: Admin home page
/certs: print certs on machine
/clusters: upstream cluster status
/config_dump: dump current Envoy configs (experimental)
/cpuprofiler: enable/disable the CPU profiler
/healthcheck/fail: cause the server to fail health checks
/healthcheck/ok: cause the server to pass health checks
/help: print out list of admin commands
/hot_restart_version: print the hot restart compatibility version
/listeners: print listener addresses
/logging: query/change logging levels
/quitquitquit: exit the server
/reset_counters: reset all counters to zero
/runtime: print runtime values
/runtime_modify: modify runtime values
/server_info: print server version/status information
/stats: print server stats
/stats/prometheus: print server stats in prometheus format
```

其中主要的接口如下：

- clusters：查看各个部署组下的实例信息和健康信息，例如查看本地服务节点的健康信息 `curl -s 127.0.0.1:15000/clusters | grep &quot;in#&quot;` 。
- config_dump：将服务路由信息打印出来，一般调用 `curl -s 127.0.0.1:15000/config_dump -o 1.json` ，打开1.json即可查看路由信息。

查看SideCar监控信息

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过 TSF 控制台查看 SideCar 监控信息，包含 SideCar 运行状态、SideCar 监控数据和 SideCar 日志。

> **注意：**
>
> 以下信息仅针对容器和虚拟机部署的 Mesh 应用生效。

操作步骤

1. 登录 【TSF 控制台】。
2. 在左侧导航栏选择【部署组】，单击"部署组 ID "进入部署组详情，默认在【服务实例列表】页，可查看 SideCar 状态。

3. 选择您感兴趣的实例，将鼠标放置在 SideCar 状态上，单击【查看日志】，跳转至 SideCar 日志 tab 页。
4. 在 SideCar 日志页面，选择日志类型、组件和时间，可查看日志信息。

5. 选择【SideCar监控】，查询 SideCar 的版本与组件状态和基本信息。

配置 Sidecar 过滤器

最近更新时间: 2025-02-18 16:02:00

操作场景

Service Mesh 应用支持在 Sidecar 上运行用户自定义的 Lua 脚本的能力。用户可以自定义一些业务逻辑放在 Sidecar 上执行，来实现一些自定义的业务逻辑（例如在请求头中添加参数、设置服务路由规则等）。配置 Sidecar 过滤器可以大大加强 Mesh 应用的灵活性。

操作步骤

新建过滤器

1. 登录 【TSF 控制台】，在左侧导航栏单击【应用管理】。
2. 在应用管理列表页，单击目标应用的 ID，进入应用详情页。
3. 访问 Sidecar过滤器 页面，新建过滤器。

- 填写过滤器名称以及备注。
- 填写过滤器作用位置。
- 作为服务端是指：该脚本将在部署组运行启动的微服务作为服务端时生效。
- 作为客户端是指：该脚本将在部署组运行启动的微服务作为客户端的时候生效。当选择过滤器位置为客户端时，需要填写被调用的微服务的名称。仅当调用被调服务的时候，过滤器会生效。
- 填写脚本内容。脚本需要严格按照 Lua 脚本的格式进行书写。
- Lua 脚本最大为65535字节。
- 已经下线的过滤器才能删除。
- 编辑后的 Lua 脚本，需要再次发布之后才能生效。

以下为微服务作为客户端时，向请求头中添加两个字段并返回请求体大小的示例：

```
function envoy_on_request(request_handle)
request_handle:headers():add("foo", "bar")
end
function envoy_on_response(response_handle)
body_size = response_handle:body():length()
response_handle:headers():add("response-body-size", tostring(body_size))
end
```

有关 Lua 脚本更多的书写原则，可以参考 Envoy 使用文档。

发布过滤器

创建好的过滤器需要发布才能生效。当前 TSF 支持将同一个过滤器发布到多个部署组上，或将某一个过滤器下线。过滤器也可以发布在离线状态的部署组上，当部署组启动时会将过滤器加载。

TSF 原生应用开发指南
Demo 工程概述

最近更新时间: 2025-02-18 16:02:00

获取 Demo

开发前，请确保：

- 已安装 Java 和 Maven
- （如果需要部署容器）已安装 Docker

下载 Demo 地址： spring cloud demo。README 中介绍了打包 jar 包和 Docker 镜像的方法。

> **注意：**
>
> 原生应用不支持全局命名空间特性。

Demo 目录结构

```
.
├── pom.xml # parent pom.xml
├── consul-consumer # 使用 consul 的 consumer
├── consul-provider # 使用 consul 的 provider
├── eureka-consumer # 使用 eureka 的 consumer
├── eureka-provider # 使用 eureka 的 provider
├── eureka-server # eureka server
├── gateway # 使用 Spring Cloud Gateway 的网关应用
└── zuul1 # 使用 Zuul1 的网关应用
```

从 pom.xml 可以看出，只使用了开源的原生组件。

调用说明

Demo 提供的服务和监听端口为：

- consul-consumer:8001
- consul-provider:8002
- eureka-consumer:8003
- eureka-provider:8004

可以访问 consumer 来调用 provider，例如：

```
curl consul-consumer:8001/ping-provider # 会调用 consul-provider:8002/ping

curl eureka-consumer:8003/ping-provider # 会调用 eureka-provider:8004/ping
```

TSF 原生应用概述

最近更新时间: 2025-02-18 16:02:00

TSF 原生应用基于开源的 Spring Cloud 应用，无需改造可直接接入TSF，使用 TSF 服务治理和调用链等功能。

优势说明

- 无需修改应用代码
- 无需引入额外 SDK（除调用链外）
- 无需额外配置

实现原理

应用到注册中心的所有请求如注册、发现，会被代理到我们自己的注册中心从而完成服务的注册和发现。当前的服务治理功能是基于 [TSF Mesh] 来实现的，服务调用会经过 TSF Mesh 的 sidecar，在 sidecar 中实现负载均衡、服务路由等治理功能。

> **注意：**
>
> 服务中的熔断、限流功能可能会 TSF 自身的功能冲突，请确保只开启了一种，否则可能会产生非预期的结果。如果确定要启用 TSF 的治理功能来代替服务自身的，可以参考 【关闭服务熔断和限流规则】 。

功能说明

目前仅支持 Spring Cloud，且支持 Eureka/Consul 两种注册中心。后续可能支持更多语言和框架，以及更多注册中心。具体如下：

spring cloud功能	开源实现	原生应用	说明
注册发现	Netflix Eureka Consul Discovery	兼容	-
负载均衡	Netflix Ribbon Spring Cloud loadbalancer	兼容	-
服务调用	Feign RestTemplate	兼容	自定义标签，需在 HTTP 请求头添加 tsf-mesh-tag: KEY=VALUE
服务限流	-	支持	自定义标签，需在 HTTP 请求头添加 tsf-mesh-tag: KEY=VALUE
服务熔断	hystrix	支持	-
服务鉴权	-	支持	自定义标签，需在 HTTP 请求头添加 tsf-mesh-tag: KEY=VALUE
配置管理	Config Server Consul Config	支持	引入 tsf-sdk
链路追踪	Spring Cloud Sleuth zipkin	兼容	-
微服务网关	Spring Cloud Gateway Netflix Zuul	兼容	-

关闭服务熔断和限流规则

最近更新时间: 2025-02-18 16:02:00

操作场景

用户在服务限流和服务熔断 tab 下新建服务限流/熔断规则时，如果已通过 Hystrix/Sentinel 等组件配置了限流熔断规则，可能会与此处新建的规则冲突，需要关闭已自建的规则。该任务指导您关闭自建的服务限流/熔断规则。

> **注意：**
>
> 某些配置可能会关闭其他功能，请注意使用，做好验证。

操作步骤

限流

组件	关闭方法
Resilience	不支持通过 property 配置，需自行修改代码来关闭
Sentinel	不支持通过 property 配置，需自行修改代码来关闭

熔断

Hystrix/Spring Cloud Hystrix

可以通过 property hystrix.command.default.circuitBreaker.enabled 关闭，修改 application.yml：

```
hystrix:
command:
default:
circuitBreaker:
enabled: false
```

如果用了 Feign，此方式也可以关闭其中的熔断功能。

Resilience

不支持配置，只能通过 transitionToDisabledState() 或自行修改代码来关闭。

`transitionToDisabledState` 示例如下：

```
public class ProviderServiceResilience {
private final static String cbName = "default";

private final CircuitBreakerRegistry cbRegistry;

public ProviderServiceResilience() {
cbRegistry = CircuitBreakerRegistry.ofDefaults();
}
```

```
public String run() {
try {
cbRegistry.circuitBreaker(cbName).executeCallable(...);
} catch (Exception e) {
}
return "resilience fallback\n";
}

public void disable() {
cbRegistry.circuitBreaker(cbName).transitionToDisabledState();
}
}
```

Spring Cloud Circuit Breaker - Resilience

> **注意：**
>
> 此方法会关闭其他 Resilience 功能，如 TimeLimiter，请谨慎使用。

可以通过 property `spring.cloud.circuitbreaker.resilience4j.enabled` 关闭，修改 application.yml：

```
spring:
cloud:
circuitbreaker:
resilience4j:
enabled: false
```

Sentinel

需要自行修改代码来关闭。

Spring Cloud Circuit Breaker - Sentinel

可以通过 property `spring.cloud.circuitbreaker.sentinel.enabled` 关闭，修改 application.yml：

```
spring:
cloud:
circuitbreaker:
sentinel:
enabled: false
```

应用部署（容器场景）

最近更新时间: 2025-02-18 16:02:00

准备工作

1. 下载 TSF 提供的 Demo 。
2. 解压 Demo 压缩包，按 README 提示执行命令 `make docker-build` 。
3. 在 TSF 控制台上已创建容器集群并添加节点，详情参考 [集群] 。

操作步骤

步骤1：创建并部署原生应用

1. 创建应用

1. 登录 【TSF 控制台】。
2. 在左侧导航，单击【应用管理】，进入应用列表页。
3. 在应用列表页，单击【新建应用】，并填写以下信息：
   - 应用名：填写应用名称
   - 部署方式：选择**容器部署**
   - 应用类型：选择 **原生应用**
   - 标签（可选）：可选择已有标签或者点击标签管理去新建标签。
4. 单击【提交】，完成应用创建。

2. 将镜像推送到仓库

1. 在左侧导航栏，单击【镜像仓库】，进入镜像列表页。首次使用时，您需要设置镜像仓库密码。
2. 在镜像列表页，单击【应用管理】>【ID/应用名】>【镜像】，单击【使用指引】，根据指引中的命令将 Demo 应用的镜像（参考 准备工作 中的第2步）推送到镜像仓库中 。

3. 创建部署组

详细操作请参考 【容器应用部署组】 中关于**创建部署组**的内容。

4. 部署应用

详细操作请参考 【容器应用部署组】 中关于**部署应用**的内容。部署应用后部署组状态变为运行中。

步骤2：查看服务是否注册成功

1. 在左侧导航栏，单击【服务治理】进入服务列表页，选择正确的地域和命名空间，查看服务是否注册成功。如果注册成功，服务显示在线状态。

2. 在服务列表页，单击服务 ID，进入服务详情页，查看具体信息。

步骤3：验证服务调用

使用与之前相同的流程部署一组 consumer 和 provider（如 consul-consumer 和 consul-provider）。

1. 请求 consumer 来调用 provider

有3种方式可以从公网请求 consumer：

- **云主机 IP + NodePort**： 如果部署组在部署时，选择了 NodePort 访问方式，可以通过**云主机 IP + NodePort**来访问 `consumer` 服务的 `/ping-provider` 接口。其中 `云主机 IP` 为集群中任一云主机的 IP， `NodePort` 可以在部署组的基本信息页面被查看。

```
curl <云主机 IP>:<NodePort>/ping-provider
```

- **负载均衡 IP + 服务端口**：如果部署组在部署时，选择了公网访问方式，可以通过**负载均衡 IP + 服务端口**来访问 `consumer` 服务的 `/ping-provider` 接口。
- **API 网关**：用户可以通过在 API 网关配置微服务 API 来调用 user 服务的接口。

也可以在容器内通过服务名和服务端口请求 consumer。先通过 kubectl 等方式进入 容器，然后调用：

```
wget -O- consul-consumer:8001/ping-provider
wget -O- eureka-consumer:8003/ping-provider
```

2. 在控制台验证服务之间是否调用

可以验证服务是否成功被注册，同时验证服务之间是否成功地进行了调用。

在**服务治理**界面：选择集群和命名空间后，如果服务列表中的服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，表示 provider 被 consumer 请求成功。

在**依赖拓扑**界面：调整时间范围，使其覆盖服务运行期间的时间范围，正常情况下，将出现服务之间相互依赖的界面。

应用部署（虚拟机场景）

最近更新时间: 2025-02-18 16:02:00

准备工作

1. 本地安装 java 和 maven 环境。
2. 下载 TSF 提供的 Demo
3. 解压 Demo 压缩包，按 README.md 提示执行命令 `make build` 。
4. 在 TSF 控制台上已创虚拟机集群并添加节点，参考 【集群】 。

操作步骤

步骤1：创建并部署原生应用

1. 创建应用

1. 登录 【TSF 控制台】 。
2. 在左侧导航栏单击【应用管理】，进入应用管理列表页。
3. 单击【新建应用】，并填写以下信息：

- 应用名：填写应用名称
- 部署方式：选择**虚拟机部署**
- 应用类型：选择 **原生应用**
- 标签（可选）：可选择已有标签或者点击标签管理去新建标签。

4. 单击【提交】，完成应用创建，在弹出的窗口中选择【确认】，前往上传程序包并部署应用。

2. 上传程序包

1. 上传程序包页面，单击【上传程序包】，选择程序包（如 `consul-provider/target/consul-provider-*.jar` ），填写程序包相关信息。
2. 单击【提交】，完成上传。

3. 创建部署组

详细操作请参考 【虚拟机应用部署组】 中关于**创建部署组**的内容。

4. 部署应用

详细操作请参考 【虚拟机应用部署组】 中关于**部署应用**的内容。部署应用后部署组状态变为运行中。

步骤2：查看服务是否注册成功

1. 在左侧导航栏单击 【服务治理】，进入服务列表页，查看服务是否注册成功。如果成功，服务显示在线状态。

2. 在服务列表页单击服务 ID，进入服务详情页，查看具体信息。

步骤3：验证服务调用

使用与之前相同的流程部署一组 consumer 和 provider（如 consul-consumer 和 consul-provider）。

1. 登录服务器验证服务间调用

为了验证 consumer 服务能通过服务名来调用 provider 服务，需要用户通过以下方式来请求 provider 服务的调用（以 consul 为例）：

- 登录 consul-consumer 所在云服务器，执行如下 `curl` 命令调用 provider 服务接口。

```
curl localhost:8001/ping-provider
```

- 登录 consul-consumer 所在云服务器，执行如下 `curl` 命令调用 provider 服务接口。

```
curl consul-provider:8002/ping
```

- **API 网关**：用户可以通过在 API 网关配置微服务 API 来调用 consumer 服务的接口。 。

2. 在控制台验证服务之间是否调用

可以验证服务是否成功被注册，同时验证服务之间是否成功地进行了调用。

在**服务治理**界面：选择集群和命名空间后，如果服务列表中的服务状态为**在线**或**单点在线**，表示服务被代理注册成功。如果服务提供者的请求量大于0，表示 provider 被 consumer 请求成功。

在**依赖拓扑**界面：调整时间范围，使其覆盖服务运行期间的时间范围，正常情况下，将出现服务之间相互依赖的界面。

调用链使用

最近更新时间: 2025-02-18 16:02:00

操作场景

使用原生应用时，如果服务已经在使用调用链后端（如 Skywalking、Jaeger），用户可以选择继续使用自己搭建的调用链后端，也可以改用 TSF 提供的调用链功能。

操作步骤

使用自己搭建的调用链后端

只需要保证 TSF 部署的应用到调用链后端的网络是通的，然后正常部署原生应用即可。目前已测试通过 Skywalking、Zipkin、Jaeger、Pinpoint。

要使用 Skywalking/Pinpoint 这类需要 Java Agent 的后端，如果在虚拟机部署，需要先自行把相关文件安装在虚拟机上，然后做好路径配置。如果部署容器，也是类似。

使用 TSF 的调用链功能

需要在应用中传递调用链信息，目前支持 Zipkin 的 B3 Propagation，所以只要是和 Zipkin B3 兼容的 SDK 均可，例如 Spring Cloud Sleuth。

使用 Spring Cloud Sleuth 来传递调用链信息

在 pom.xml 加入 `spring-cloud-starter-sleuth` 依赖即可：

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
```

通过 Feign/RestTemplate 等调用服务时，会传递如下 HTTP header：

```
X-B3-SpanId: 381cea277131b627
X-B3-ParentSpanId: 7fba7505d61e4db2
X-B3-Sampled: 0
X-B3-TraceId: 7fba7505d61e4db2
```

制作容器镜像-KonaJDK

最近更新时间: 2025-02-18 16:02:00

该任务指导您通过 Spring Cloud 和 Mesh 两种方式制作容器镜像。

准备构建材料

Spring Cloud 应用构建材料

1. 简化版本

简化版本的 Dockerfile 不包含文件配置和 JVM 监控功能，仅需要用户替换掉 Dockerfile 中 Spring Cloud 应用 jar 包名称，您也可以先试用 TSF 提供的 Spring Cloud 应用 Demo JAR 包（下载地址）。

关于JDK版本，推荐使用 Tencent KonaJDK，请下载 KonaJDK 安装文件（下载地址）。

> **注意：**
>
> 在 Spring Cloud 应用 JAR 包同级目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

2. 使用 JVM 监控功能

如果您希望使用 [JVM 监控] 功能，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.2`（下载地址），然后在 CMD 命令中启动该组件。

> **注意：**
>
> 将 Spring Cloud 应用 JAR 包和 JVM 监控组件放在同级目录下，并在该目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
```

```
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar 包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JVM 监控组件要和您的 Dockerfile 位于同一级目录，并创建 JVM 监控数据采集目录
ENV agentjar TencentCloudJvmMonitor-1.1.2-RELEASE.jar
# 若容器的基础版本为 非 gnu-libc 版本，如 Alpine，请添加如下语句
# RUN ln -sf /lib/libc.musl-x86_64.so.1 /lib/ld-linux-x86-64.so.2
COPY ${agentjar} ${workdir}

RUN mkdir -p /data/tsf_apm/monitor/jvm-metrics/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 使用 JVM监控功能需要加上 gclog 和 javaagent 的配置, 否则将无法提供 jvm 监控能力
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "exec java -Xloggc:/data/tsf_apm/monitor/jvm-metrics/gclog.log -XX:+PrintGCDateStamps -XX:+PrintGC
Details -verbose:gc -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=8 -XX:GCLogFileSize=50M -javaagent:${workdi
r}/${agentjar}=hascontroller=true ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

3. 使用文件配置

如果您希望使用 TSF [文件配置] 功能，则需要在 Dockerfile 中增加文件配置组件 `tsf-consul-template-docker.tar.gz`（下载地址），然后在 CMD 启动命令中启动该组件。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# tsf-consul-template-docker 用于文件配置功能，如不需要可注释掉该行
ADD tsf-consul-template-docker.tar.gz /root/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "sh /root/tsf-consul-template-docker/script/start.sh; exec java ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

私有化版本使用建议：

私有化的 TSF 要支持 stdout 日志，需要在启动命令中将 stdout 及 stderr 重定向到一个文件中。将上文的 CMD 一行替换成：

```
RUN mkdir -p /data/tsf_std/stdout/logs
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -Xshare:off -jar ${jar} 2>&1 > /data/tsf_std/stdout/logs/sys_log.log"]
```

Mesh 应用构建材料

1. 下载 Mesh 应用 Demo 包：userService.tar.gz。
2. 在该 tar.gz 包同级目录下，编写 Dockerfile 文件：

```
FROM centos:7
RUN mkdir /root/app/
# 其中 userService.tar.gz 是 Mesh 应用压缩包
ADD userService.tar.gz /root/app/
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENTRYPOINT ["bash","/root/app/userService/start.sh"]
```

Mesh 应用压缩包解压后的文件目录结构及文件规范参考 【Mesh Demo 介绍】 。

使用文件配置功能

如果容器应用需要使用 TSF 文件配置功能，需要修改 Dockerfile，具体使用指引参考 【文件配置>前提条件】。

构建镜像

1. 在 Dockerfile 所在目录执行 build 命令：

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 <主账号 ID> 对应用户的**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在个人信息页面获取），<应用名> 表示

控制台上的应用名。 tag 为镜像的 tag，用户可自定义。 2. 命令执行完成后，通过 docker image ls 查看创建的镜像。

制作容器镜像

最近更新时间: 2025-02-18 16:02:00

该任务指导您通过 Spring Cloud 和 Mesh 两种方式制作容器镜像。

如需使用 Tencent KonaJDK 替换 openJDK，请查看 【制作容器镜像-KonaJDK】 说明。

准备构建材料

Spring Cloud 应用构建材料

1. 简化版本

简化版本的 Dockerfile 不包含文件配置和 JVM 监控功能，仅需要用户替换掉 Dockerfile 中 Spring Cloud 应用 jar 包名称，您也可以先试用 TSF 提供的 Spring Cloud 应用 Demo JAR 包（下载地址）。

> **注意：**
>
> 在 Spring Cloud 应用 JAR 包同级目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -jar ${jar}"]
```

2. 使用 JVM 监控功能

如果您希望使用 【JVM 监控】功能，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.2`（下载地址），然后在 CMD 命令中启动该组件。

> **注意：**
>
> 将 Spring Cloud 应用 JAR 包和 JVM 监控组件放在同级目录下，并在该目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
```

```
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar 包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JVM 监控组件要和您的 Dockerfile 位于同一级目录，并创建 JVM 监控数据采集目录
ENV agentjar TencentCloudJvmMonitor-1.1.2-RELEASE.jar
# 若容器的基础版本为 非 gnu-libc 版本，如 Alpine，请添加如下语句
# RUN ln -sf /lib/libc.musl-x86_64.so.1 /lib/ld-linux-x86-64.so.2
COPY ${agentjar} ${workdir}

RUN mkdir -p /data/tsf_apm/monitor/jvm-metrics/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 使用 JVM监控功能需要加上 gclog 和 javaagent 的配置, 否则将无法提供 jvm 监控能力
CMD ["sh", "-ec", "exec java -Xloggc:/data/tsf_apm/monitor/jvm-metrics/gclog.log -XX:+PrintGCDateStamps -XX:+PrintGC
Details -verbose:gc -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=8 -XX:GCLogFileSize=50M -javaagent:${workdi
r}/${agentjar}=hascontroller=true ${JAVA_OPTS} -jar ${jar}"]
```

3. 使用文件配置

如果您希望使用 TSF [文件配置] 功能，则需要在 Dockerfile 中增加文件配置组件 `tsf-consul-template-docker.tar.gz`（下载地址），然后在 CMD 启动命令中启动该组件。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# tsf-consul-template-docker 用于文件配置功能，如不需要可注释掉该行
ADD tsf-consul-template-docker.tar.gz /root/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
CMD ["sh", "-ec", "sh /root/tsf-consul-template-docker/script/start.sh; exec java ${JAVA_OPTS} -jar ${jar}"]
```

私有化版本使用建议：

私有化的 **TSF 1.12 及之前版本**要支持 stdout 日志，需要在启动命令中将 stdout 及 stderr 重定向到一个文件中。将上文的 `CMD` 一行替换成：

```
RUN mkdir -p /data/tsf_std/stdout/logs
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -jar ${jar} 2>&1 > /data/tsf_std/stdout/logs/sys_log.log"]
```

Mesh 应用构建材料

1. 下载 Mesh 应用 Demo 包：userService.tar.gz。

2. 在该 tar.gz 包同级目录下，编写 Dockerfile 文件：

```
FROM centos:7
RUN mkdir /root/app/
# 其中 userService.tar.gz 是 Mesh 应用压缩包
ADD userService.tar.gz /root/app/
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENTRYPOINT ["bash","/root/app/userService/start.sh"]
```

Mesh 应用压缩包解压后的文件目录结构及文件规范参考 【Mesh Demo 介绍】 。

使用文件配置功能

如果容器应用需要使用 TSF 文件配置功能，需要修改 Dockerfile，具体使用指引参考 【文件配置>前提条件】 。

构建镜像

1. 在 Dockerfile 所在目录执行 build 命令：

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 <主账号 ID> 对应用户云平台的**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在云平台个人信息页面获取。），<应用名> 表示控制台上的应用名。 tag 为镜像的 tag，用户可自定义。 2. 命令执行完成后，通过 docker image ls 查看创建的镜像。

应用开发
Go 应用开发
Go 应用接入TSF

最近更新时间: 2025-02-18 16:02:00

TSF-Go 基于轻量级 Go 微服务框架 Kratos 为用户现存的 Go 应用提供了接入TSF 治理平台的能力。本文档介绍 Go 应用从接入TSF 到部署应用的操作方法及相关注意事项。

> **注意：**
>
> 您也可以通过 SDK 文档 查看 TSF Go 最新使用说明。

功能特性

- 自动集成 TSF 平台治理能力：分布式远程配置、远程日志、分布式调用链追踪、监控、服务鉴权、服务路由、全链路灰度发布、API 自动上报。
- 同时支持 gRPC 和 HTTP 协议，并可以和 Java Spring Cloud 服务互相调用。
- 开放性高，丰富的 Middlewares、Options 可以自定义组件。
- 一切围绕 Protobuf 定义 Service、Interface、Error、Validating、Swagger json 等。
- 拥抱开源规范、Trace、Validate、API Definition 等都直接使用开源规范和协议。
- 支持私有化部署。

安装依赖

1. 安装 protoc v3.15.0+

请根据自己使用的操作系统，优先选择对应的包管理工具安装，如：

- Linux 下用 yum或 apt 等安装
- macOS 通过 brew 安装
- Windows 通过下载可执行程序或者其他安装程序来安装

2. 安装 protoc-gen-xxx

```
go get -u github.com/golang/protobuf/protoc-gen-go
go get -u google.golang.org/grpc/cmd/protoc-gen-go-grpc
go get -u github.com/go-kratos/kratos/cmd/protoc-gen-go-http
```

服务端开发

1. 通过 protobuf 定义服务接口

```
syntax = "proto3";
```

```
// 定义protobuf包名pacakage_name
package helloworld;

// 如果不使用restful http协议，可以不引入此proto
import "google/api/annotations.proto";

// 这里go_package指定的是protofbu生成文件xxx.pb.go在git上的地址
option go_package = "github.com/tencentyun/tsf-go/examples/helloworld/proto";

// 定义服务名service_name
service Greeter {
// 方法接口名,rpc_method
rpc SayHello (HelloRequest) returns (HelloReply) {
}
}

// 请求参数
message HelloRequest {
string name = 1;
}

// 响应参数
message HelloReply {
string message = 1;
}
```

如上，这里我们定义了一个Greeter服务，这个服务里面有个 SayHello 方法，接收一个包含 msg 字符串的 HelloRequest 参数，返回 HelloReply 数据。 这里需要注意以下几点：

- syntax 必须是 proto3，tsf go 都是基于 proto3 通信的。
- package 后面必须有 option go_package="github.com/tencentyun/tsf-go/examples/helloworld/proto";指明您的pb.go生成文件的git存放地址，协议与服务分离，方便其他人直接引用
- 编写 protobuf 时必须遵循 谷歌官方规范。

2. 生成服务端桩代码 xxx.pb.go 代码

通过protoc命令生成服务代码(grpc协议) `protoc --proto_path=. --proto_path=./third_party --go_out=paths=source_relative:. --go-grpc_out=paths=source_relative:. *.proto`

3. 编写 service 实现层代码

```
import pb "github.com/tencentyun/tsf-go/examples/helloworld/proto"

// server is used to implement helloworld.GreeterServer.
type server struct {
pb.UnimplementedGreeterServer
}

// SayHello implements helloworld.GreeterServer
func (s *server) SayHello(ctx context.Context, in *pb.HelloRequest) (*pb.HelloReply, error) {
return &pb.HelloReply{Message: fmt.Sprintf("Welcome %+v!", in.Name)}, nil
}
```

4. 编写 server(grpc 协议) 启动入口 main.go

```
import pb "github.com/tencentyun/tsf-go/examples/helloworld/proto"
import tsf "github.com/tencentyun/tsf-go"
import "github.com/go-kratos/kratos/v2"
import "github.com/go-kratos/kratos/v2/transport/grpc"

func main() {
flag.Parse()
// grpc协议
grpcSrv := grpc.NewServer(
// 定义grpc协议监听地址
grpc.Address(":9000"),
grpc.Middleware(
// 使用tsf默认的middleware
tsf.ServerMiddleware(),
),
)
// 将第三步骤中的service实现结构体注入进grpc server中
s := &server{}
pb.RegisterGreeterServer(grpcSrv, s)

// 应用配置
opts := []kratos.Option{
// 定义注册到服务发现中的服务名
kratos.Name("provider-grpc"),
// 添加grpc server至应用运行时
kratos.Server(grpcSrv),
}
// 添加tsf应用默认启动配置
// 如果不想启用服务注册，可以加入tsf.EnableReigstry(false)该Option
opts = append(opts, tsf.AppOptions()...)
app := kratos.New(opts...)
// 应用阻塞式启动
if err := app.Run(); err != nil {
panic(err)
}
}
```

5. 服务启动

- 参考 [轻量级服务注册中心] 搭建并启动一个本地轻量级 consul 注册中心；如果暂时不想启动服务注册直接调试，则可以在第4步骤中传入 `tsf.EnableReigstry(false)` 即： `opts = append(opts, tsf.AppOptions(tsf.EnableReigstry(false))...)`
- 执行 `go run main.go` 即可启动 server。

客户端开发（gRPC 协议）

1. 编写客户端代码

```
import pb "github.com/tencentyun/tsf-go/examples/helloworld/proto"
import tsf "github.com/tencentyun/tsf-go"
import "github.com/go-kratos/kratos/v2/transport/grpc"
import "import "github.com/go-kratos/kratos/v2"

func main() {
```

```
flag.Parse()
// 指定被调方服务连接地址:<scheme>://<authority>/<service_name>
// 如果使用服务发现，此处scheme固定为discovery，authority留空，service_name为定义注册到服务发现中的服务名
// 如果不使用服务发现，直接填写"<ip>:<port>"即可
clientOpts := []grpc.ClientOption{grpc.WithEndpoint("discovery:///provider-grpc")}
// 如果不使用服务发现，此行可以删除
clientOpts = append(clientOpts, tsf.ClientGrpcOptions()...)
conn, err := grpc.DialInsecure(context.Background(), clientOpts...)
if err != nil {
panic(err)
}
client := pb.NewGreeterClient(conn)
reply, err := client.SayHello(context.Background(), &pb.HelloRequest{Name: "tsf_grpc"})
if err != nil {
panic(err)
}
}
```

2.启动客户端

执行 `go run main.go` 即可启动客户端。

部署至 TSF 治理平台

1. 在 TSF 上创建应用和镜像仓库

参考文档 【应用管理】 创建应用并开通镜像仓库。

2. 编写 Dockerfile

```
FROM centos:7

RUN echo "ip_resolve=4" >> /etc/yum.conf
#RUN yum update -y && yum install -y ca-certificates

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

COPY provider ${workdir}
WORKDIR ${workdir}

# tsf-consul-template-docker 用于文件配置功能，如不需要可注释掉该行
#ADD tsf-consul-template-docker.tar.gz /root/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。如果加了${JAVA_OPTS},需要在TSF的容器部署组启动参数中删除默认的"-Xms128m xxx"参数,否则会启动失败
#使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
CMD ["sh", "-ec", "exec ${workdir}provider ${JAVA_OPTS}"]
```

您需要将上述的 provider 替换为实际的可执行二进制文件名。

3. 打包镜像

将 `GOOS=linux go build` 编译出的二进制文件放在 Dockfile 同一目录下，执行如下命令。

```
docker build . -t ccr.ccs.tencentyun.com/tsf_xxx/provider:1.0
```

镜像地址改成应用上传镜像页面中实际的地址

4. 推送镜像

执行如下命令推送镜像到镜像仓库。

```
docker push ccr.ccs.tencentyun.com/tsf_xxx/provider:1.0
镜像地址改成应用上传镜像页面中实际的地址
```

Examples

- gRPC
- HTTP
- gin-go
- log
- error
- tracing
- breaker

HTTP2 应用开发

最近更新时间: 2025-02-18 16:02:00

操作场景

本文以 provider-demo 和 consumer-demo 两个工程为例为你介绍 TSF 应用配置 http2 的操作方法。

前提条件

- 【安装1.8或以上版本 JDK】
- 【下载 TSF Demo 工程】（springboot 版本2.0+，tomcat 版本8.5+）

> **注意：**
>
> - 推荐使用 1.29.0-Finchley-RELEASE。
> - 本文 springboot-2.0.9.RELEASE 和 tomcat-8.5.56 为例。

> -

操作步骤

步骤1. 制作 SSL 证书

通过 JDK 自带的 keytool 执行以下命令：

```
keytool -genkey -alias tomcat -keyalg RSA -keystore ./keystore.jks -storepass 123456
```

执行成功后当前目录下会生成 keystore.jks 证书文件。

步骤2. 改造 provider-demo 工程

1. 复制 jks 证书文件到 provider-demo 工程的 resources 目录下。

2. 修改 spring 配置文件，在bootstrap.yaml 文件中增加如下配置。

```
server.http2.enabled=true
server.ssl.key-store=classpath:keystore.jks
server.ssl.key-store-password: 123456
```

3. 启动 provider-demo，浏览器访问 `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18081/echo/1` 。

Chrome 可能会提示“您的连接不是私密连接”。

只需单击任意空白处，键入“thisisunsafe”即可。

打开 Console，Protocol 的值为 h2 表示配置成功。

4. 开启 http 访问端口（可选）。

由于此时只能通过 https 方式访问，可加入 Tomcat 配置开放新端口来支持 http 访问。

5. 在启动类中增加 Bean。

java package com.tsf.demo.provider;

```
import org.apache.catalina.connector.Connector;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.web.embedded.tomcat.TomcatServletWebServerFactory;
import org.springframework.boot.web.servlet.server.ServletWebServerFactory;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.context.annotation.Bean;
import org.springframework.tsf.annotation.EnableTsf;

@SpringBootApplication
@EnableFeignClients // 使用Feign微服务调用时请启用
@EnableTsf
public class ProviderApplication {

@Value("${http.port}")
private Integer port;

/**
* Tomcat增加支持http访问
**/
@Bean
public ServletWebServerFactory servletContainer() {
TomcatServletWebServerFactory tomcat = new TomcatServletWebServerFactory();
Connector connector = new Connector(TomcatServletWebServerFactory.DEFAULT_PROTOCOL);
// connector.setSecure(false);
// connector.setScheme("http");
connector.setPort(port);
tomcat.addAdditionalTomcatConnectors(connector);
return tomcat;
}
```

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

6. 在 Bootstrap.yml 配置文件中增加自定义配置。

```
http.port=18082
```

7. 重启 provider-demo，浏览器访问 `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18082/echo/1` 。

此时 provider-demo 完成支持 https 和 http（通过不同端口访问），其中 https 访问时使用 http2 协议。

步骤3. 改造 consumer-demo 工程

1. proxy 中的 @FeignClient 注解需指定 https 方式访问。

```
@FeignClient(name = "http://imgcache.finance.cloud.tencent.com:80provider-demo")
```

2. RestTemplate 同理。

```
restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/" + str, String.class);
```

3. 通过注入不同的 bean 选择是否使用SSL证书认证（以下步骤二选一）：

- 使用SSL证书认证访问方式

4. 复制 jks 证书文件到 consumer-demo 工程的 resources 目录。
5. spring 配置文件中增加自定义配置。

```
test-ssl-config.key-store=classpath:keystore.jks
test-ssl-config.key-store-password: 123456
```

6. 修改 bean（restTemplate、feignClient）

```
package com.tsf.demo.consumer;

import feign.Client;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.TrustSelfSignedStrategy;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.ssl.SSLContexts;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.cloud.netflix.ribbon.SpringClientFactory;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.cloud.openfeign.ribbon.CachingSpringLoadBalancerFactory;
import org.springframework.cloud.openfeign.ribbon.LoadBalancerFeignClient;
import org.springframework.context.annotation.Bean;
import org.springframework.http.client.HttpComponentsClientHttpRequestFactory;
import org.springframework.tsf.annotation.EnableTsf;
import org.springframework.util.ResourceUtils;
import org.springframework.web.client.AsyncRestTemplate;
import org.springframework.web.client.RestTemplate;

import javax.net.ssl.*;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.*;
import java.security.cert.CertificateException;

@SpringBootApplication
@EnableFeignClients // 使用Feign微服务调用时请启用
@EnableTsf
public class ConsumerApplication {

@Value("${test-ssl-config.key-store}")
private String file;
@Value("${test-ssl-config.key-store-password}")
private String password;

@LoadBalanced
@Bean
public RestTemplate restTemplate() {
SSLConnectionSocketFactory csf = new SSLConnectionSocketFactory(getSSLSocket(file, password),
new String[]{"TLSv1"},
null,
NoopHostnameVerifier.INSTANCE);
CloseableHttpClient httpClient = HttpClients.custom()
.setSSLSocketFactory(csf)
.build();
HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
requestFactory.setHttpClient(httpClient);
```

```
return new RestTemplate(requestFactory);
}

@Bean
public Client feignClient(CachingSpringLoadBalancerFactory cachingFactory, SpringClientFactory clientFactory) {
return new LoadBalancerFeignClient(
new Client.Default(getSSLSocket(file, password).getSocketFactory(), (hostname, session) -> true), cachingFactory, clientFac
tory
);
}

public static SSLContext getSSLSocket(String file, String password) {
SSLContext sslContext = null;
try {
KeyStore keyStore = KeyStore.getInstance(KeyStore.getDefaultType());
InputStream keyStoreInput = new FileInputStream(ResourceUtils.getFile(file));
keyStore.load(keyStoreInput, password.toCharArray());

KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
InputStream trustStoreInput = new FileInputStream(ResourceUtils.getFile(file));
trustStore.load(trustStoreInput, null);
sslContext = SSLContexts.custom().loadKeyMaterial(keyStore, password.toCharArray())
.loadTrustMaterial(trustStore, new TrustSelfSignedStrategy()).build();
} catch (NoSuchAlgorithmException | KeyManagementException | KeyStoreException | CertificateException | IOException |
UnrecoverableKeyException e) {
e.printStackTrace();
}
return sslContext;
}

@LoadBalanced
@Bean
public AsyncRestTemplate asyncRestTemplate() {
return new AsyncRestTemplate();
}

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

- 忽略SSL认证方式 只需要修改 bean（restTemplate、feignClient）

与第一种方式的差异在 getSSLSocket() 方法。

java

```
package com.tsf.demo.consumer;</dx-codeblock>

import feign.Client;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
```

```
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.cloud.netflix.ribbon.SpringClientFactory;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.cloud.openfeign.ribbon.CachingSpringLoadBalancerFactory;
import org.springframework.cloud.openfeign.ribbon.LoadBalancerFeignClient;
import org.springframework.context.annotation.Bean;
import org.springframework.http.client.HttpComponentsClientHttpRequestFactory;
import org.springframework.tsf.annotation.EnableTsf;
import org.springframework.web.client.AsyncRestTemplate;
import org.springframework.web.client.RestTemplate;

import javax.net.ssl.*;
import java.security.*;
import java.security.cert.X509Certificate;

@SpringBootApplication
@EnableFeignClients // 使用Feign微服务调用时请启用
@EnableTsf
public class ConsumerApplication {

@LoadBalanced
@Bean
public RestTemplate restTemplate() {
SSLConnectionSocketFactory csf = new SSLConnectionSocketFactory(getSSLSocket(),
new String[]{"TLSv1"},
null,
NoopHostnameVerifier.INSTANCE);
CloseableHttpClient httpClient = HttpClients.custom()
.setSSLSocketFactory(csf)
.build();
HttpComponentsClientHttpRequestFactory requestFactory = new HttpComponentsClientHttpRequestFactory();
requestFactory.setHttpClient(httpClient);
return new RestTemplate(requestFactory);
}

@Bean
public Client feignClient(CachingSpringLoadBalancerFactory cachingFactory, SpringClientFactory clientFactory) {
return new LoadBalancerFeignClient(
new Client.Default(getSSLSocket().getSocketFactory(), (hostname, session) -> true), cachingFactory, clientFactory
);
}

public static SSLContext getSSLSocket() {
SSLContext sslContext = null;
try {
sslContext = SSLContext.getInstance("TLS");
X509TrustManager tm = new X509TrustManager() {
@Override
public void checkClientTrusted(X509Certificate[] chain, String authType) {}

@Override
public void checkServerTrusted(X509Certificate[] chain, String authType) {}
```

```
@Override
public X509Certificate[] getAcceptedIssuers() {
return null;
}
};
sslContext.init(null, new TrustManager[]{tm}, null);
} catch (NoSuchAlgorithmException | KeyManagementException e) {
e.printStackTrace();
}
return sslContext;
}

@LoadBalanced
@Bean
public AsyncRestTemplate asyncRestTemplate() {
return new AsyncRestTemplate();
}

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

}
```

7. 验证结果。

启动 consumer-demo，浏览器访问 `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18083/echo-feign/123` 。

浏览器访问 `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18083/echo-rest/123` 。

应用开发概述

最近更新时间: 2025-02-18 16:02:00

TSF 支持 Spring Cloud 原生应用、普通应用和多协议多语言 Mesh 应用，您可以根据业务场景需要开发应用，并部署到 TSF 上。

功能		原生应用	普通应用	Mesh 应用
适用场景		存量业务应用开源 Spring Cloud 零代码改造	新业务全新技术框架选型	适配不同协议（Dubbo、HTTP、gRPC）不同语言接入（PHP、Java、Python）
注册发现		✔	✔	✔
服务治理	服务鉴权	✔	tsf-sdk	Mesh 流量劫持
	服务限流	✔	tsf-sdk	Mesh 流量劫持
	服务熔断	✔	tsf-sdk	Mesh 流量劫持
	服务路由	✔	tsf-sdk	Mesh 流量劫持
调用链		业务应用 Spring Cloud Sleuth、Zipkin 组件能够接入 TSF 调用链支持服务间调用链不支持方法级调用链 业务应用 SkyWalking 能够对接用户自建的 SkyWalking 服务端	tsf-sdk	支持服务间调用链串联
日志服务		✔	✔	✔
配置管理	实时配置（分布式配置	不支持	tsf-sdk	支持
	文件配置	支持	支持	支持
增强能力	服务优雅下线	结合微服务网关 + Mesh 标签	结合微服务网关 + SDK	结合微服务网关 + Mesh 标签
	全链路灰度	✔	✔	✔
	单元化	不支持	结合微服务网关 + SDK	不支持

应用迁移
Spring Cloud Alibaba迁移TSF

最近更新时间: 2025-02-18 16:02:00

操作场景

本文档指导您将现有 Dubbo 应用迁移至 TSF 平台，并采用虚拟机 Mesh 的部署方式，通过 Service Mesh 的方式实现 Dubbo 应用透明的服务注册发现和无侵入的服务治理，享受 TSF 平台一站式的微服务框架解决方案。

下文以 Dubbo Mesh Demo 为例介绍具体的迁移方式和操作步骤。Dubbo Mesh Demo 提供了 client、greet 和 hello 3个应用，3个应用之间的调用关系是：client -> greet -> hello，client 应用每隔5秒会自动发起请求。

迁移价值

- TSF 为您提供**一站式应用生命周期管理服务。**提供从应用部署到应用运行的全流程管理，包括创建、删除、部署、回滚、扩容、下线、启动和停止应用并支持版本回溯能力。
- TSF 为您提供**高效的服务注册发现能力。**支持秒级的服务注册发现并提供本地注册信息缓存、服务实例注册发现异常告警、注册中心跨 AZ 区容灾等完善的高可用保障机制。
- TSF 为您提供**细粒度服务治理能力。**支持服务和 API 多级服务治理能力，通过配置标签形式进行细粒度的流量控制，实现灰度发布、就近路由、熔断限流、服务容错、访问鉴权等功能。
- TSF 为您提供**立体化应用数据运营。**提供完善应用性能指标监控和分布式调用链分析、服务依赖拓扑、日志服务工具，帮助您高效分析应用性能瓶颈及故障问题排查。

前提条件

- 已经下载 Dubbo Mesh Demo 的代码程序包。
- 已经创建了集群和命名空间，集群中导入3个云服务器。
- 已经创建了 client、greet 和 hello 应用（虚拟机部署方式）。

操作步骤

步骤1：修改配置通过直连服务名方式访问服务提供者

配置直连服务提供者的方式，详情请参考 Dubbo 官方文档 直连提供者。

在 Dubbo Mesh Demo 中，greet 服务需要访问 hello 服务，采用 XML 配置方式，可以配置为：

```
<dubbo:reference id="helloService" check="false" interface="org.apache.dubbo.samples.api.HelloService" url="dubbo://org.apache.dubbo.samples.api.HelloService:20880">
```

此处的直连地址为 `dubbo://org.apache.dubbo.samples.api.HelloService:20880`，`org.apache.dubbo.samples.api.HelloService` 为步骤2中 hello 服务注册的服务名。

步骤2：按照 TSF Mesh 方式构建程序包

要将 Dubbo 应用以 Mesh 的方式部署到 TSF 平台上，需要遵循 Mesh 的方式构建程序包，主要包含以下几个文件：

1. 编译好的 Dubbo 应用 jar 包

建议使用 FATJAR 包，可直接运行，如 Demo 中 greet 服务的 jar 包为 `dubbo-samples-greetservice-1.0.jar`。

2. spec.yaml 文件

此 yaml 文件用于描述 Dubbo 应用的服务注册信息，至少需要配置服务名、服务监听端口并指定 Dubbo 协议，如下所示：

```
apiVersion: v1
kind: Application
spec:
services:
- name: org.apache.dubbo.samples.api.GreetingService
ports:
- targetPort: 20881
protocol: dubbo
healthCheck:
path:
```

> **注意：**
>
> 此处的服务名如上面的 `org.apache.dubbo.samples.api.GreetingService` 将被注册到注册中心，其它 Dubbo 应用将通过该服务名进行调用，同时，服务名需要跟该 Dubbo 应用提供的 interface 名保持一致。

3. start.sh 启动脚本

此 shell 脚本用于启动部署的 Dubbo 应用，编写时保证下面两点即可：

- 幂等执行，已经启动的 Dubbo 应用不会因再次执行该脚本而被停止
- 后台执行，将日志输出到本地文件

编写完后可手工执行验证是否启动成功。

以下是 Demo 中 greet 服务的启动脚本：

```
#!/bin/bash

already_run=`ps -ef|grep "dubbo-samples-greetservice-1.0.jar"|grep -v grep|wc -l`
if [ ${already_run} -ne 0 ];then
echo "dubbo-greet already Running!!!! Stop it first"
exit -1
fi
nohup java -jar dubbo-samples-greetservice-1.0.jar 1>stdout.log 2>&1 &
```

4. stop.sh 停止脚本

此 shell 脚本用于停止部署的 Dubbo 应用，编写完后可手工执行验证是否停止成功，以下是Demo中 greet 服务的停止脚本：

```
#!/bin/bash

pid=`ps -ef|grep "dubbo-samples-greetservice-1.0.jar"|grep -v grep|awk '{print $2}'`
kill -SIGTERM $pid
echo "process ${pid} killed"
```

5. cmdline 进程执行命令文件

TSF 平台部署在主机上的 tsf-agent 会通过检查系统运行进程中是否有 cmdline 文件中的执行命令，来验证 Dubbo 应用进程是否存活，一般设置为进程的启动命令即可，如 Demo 中 greet 服务的 cmdline：

```
java -jar dubbo-samples-greetservice-1.0.jar
```

> **注意：**
>
> 这里不需要包括上面启动脚本中的 `nohup` 和 `1&gt;stdout.log 2&gt;&amp;1 &amp;` 。

步骤3：部署 Dubbo 应用到 TSF 平台

1. 登录 【TSF 控制台】。
2. 在左侧导航栏中，单击【应用管理】，进入应用列表页，选择已创建的 Dubbo 应用，进入应用详情页。
3. 单击顶部【程序包管理】，上传步骤2中构建好的程序包。
4. 单击顶部【部署组】，在部署组列表页面创建部署组，选择集群和命名空间，添加实例，选择相应版本的程序包，完成部署，具体操作方式参考 [虚拟机应用部署组]。
5. 部署完成后，在应用的【部署组】页查看对应部署组的状态，如果是 `运行中` 表示部署成功。
6. 在左侧导航栏中，单击【服务治理】，进入服务列表页，如果可以看到步骤2中 spec.yaml 文件指定的服务名且是 `在线` 状态, 表示该 Dubbo 服务注册成功，其它 Dubbo 服务可通过此服务名进行调用。

步骤4：配置服务治理规则

Dubbo 服务路由

1. 在左侧导航栏中，单击【服务治理】，进入服务列表页，单击需要设置路由的目的 Dubbo 服务名，进入服务详情页。
2. 单击顶部【服务路由】， 新建路由规则，具体操作方式参考 【服务路由使用方法】，如下图所示：

Dubbo 服务监控

1. 在左侧导航栏中，单击【服务治理】，进入服务列表页，单击需要监控的Dubbo服务名，进入服务详情页。
2. 单击顶部【服务概览】， 可查看该服务依赖拓扑、请求概览等信息，如下图所示：

Dubbo 调用链追踪

1. 在左侧导航栏中，单击【服务依赖拓扑】，进入服务依赖拓扑页面。
2. 在顶部选择对应 Dubbo 服务所在的命名空间，再选择对应时间段，页面将自动显示该命名空间下所有服务的调用关系图，如下图所示：

云上Spring Cloud应用平滑迁移TSF

最近更新时间: 2025-02-18 16:02:00

操作场景

该文档帮助您将自有集群的应用服务平滑迁移至 TSF，并在迁移过程中确保服务正常的注册发现。

当您的 Spring Cloud 应用集群已经部署在云上的生产环境并承接业务流量，同时您想将 Spring Cloud 应用迁移至 TSF 享受一站式的微服务框架解决方案，在迁移过程中保障业务流量不中断，完全实现对业务用户的透明访问即为平滑迁移。

前提条件

您已经将 Spring Cloud 应用部署在腾讯云金融专区上。

迁移价值

- TSF 为您提供**一站式应用生命周期管理服务。**提供从应用部署到应用运行的全流程管理，包括创建、删除、部署、回滚、扩容、下线、启动和停止应用并支持版本回溯能力。
- TSF 为您提供**高效的服务注册发现能力。**支持秒级的服务注册发现并提供本地注册信息缓存、服务实例注册发现异常告警、注册中心跨AZ区容灾等完善的高可用保障机制。
- TSF 为您提供**细粒度服务治理能力。**支持服务和 API 多级服务治理能力，通过配置标签形式进行细粒度的流量控制，实现灰度发布、就近路由、熔断限流、服务容错、访问鉴权等功能。
- TSF 为您提供**立体化应用数据运营。**提供完善应用性能指标监控和分布式调用链分析、服务依赖拓扑、日志服务工具，帮助您高效分析应用性能瓶颈及故障问题排查。
- TSF 为您提供**灵活的分布式配置管理能力。**支持配置版本管理、动态推送、热生效能力。

迁移方案

为保障应用迁移过程中流量不中断，您可以通过**切流迁移**和迁移实例的**双注册发现迁移**两种方式实现平滑迁移。具体方案如下：

切流迁移方案

- **实现方法**：业务应用需要改造后在 TSF 集群全量部署，应用全部改造上线后通过切换域名配置将域名指向新集群的 CLB 服务均衡地址实现流量切换。
- **方案优点**：操作简单，原有应用集群和 TSF 集群不存在相互影响。
- **方案缺点**：需要保持两套集群同时运行，存在资源浪费；此外，由于是全量迁移，迁移过程中存在业务异常的概率较大。

双注册发现方案

- **方案说明**：您仅需要将迁移的应用引入 TSF 组件依赖并进行双注册中心服务注册配置，将改造后的应用通过 TSF 进行部署，部署后的 Spring Cloud 应用即可被原应用集群和 TSF 集群中的服务发现，正常进行服务调用。

- **方案优点**：
- 改造量小，仅需引入 TSF 组件依赖并进行完成配置操作，无需改动代码即可实现双注册发现能力。
- 在迁移过程中实现已迁移应用与未迁移应用的互相发现，实现服务调用，有效保障业务连续性。
- 整体服务迁移完成后，通过TSF提供配置管理能力可动态变更服务注册发现策略，无需重启应用即可完成迁移。
- 迁移后的应用优先调用TSF内的服务实例，避免跨集群调用产生性能损耗。

> **注意**：
>
> 目前 TSF 支持原有注册中心引擎为 Eureka、Zookeeper、Consul 与 TSF 注册中心引擎的双注册发现能力。

操作步骤

1. 在迁移的服务应用的 pom.xml 文件中引入 TSF 依赖：

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-dependencies</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</parent>
```

> **注意**：
>
> 应用该功能 SDK 版本必须在1.24.0-Finchley-RELEASE、1.24.0-Greenwich-RELEASE 以上。

2. 在 `application.properties` 中添加双注册发现配置。 exclude 字段代表不将服务注册到哪些服务注册中心引擎；subscribes 字段代表从哪些服务注册引擎发现服务信息。

```
tsf:
migration:
registry:
// 不注册到ZOOKEEPER
excludes: ZOOKEEPER
// 从 EUREKA、TSF 进行服务发现
subscribes:
- EUREKA
- TSF
```

3. 将应用打包。 登录 【TSF 控制台】 ，参照 【应用管理】 完成应用发布。

4. 验证结果。 通过原有集群服务调用可发现服务是否正常，此外在【应用中心】>【服务治理】查看应用发布地域及命名空间查看是否存在服务信息，如果服务注册正常则具有该微服务名称、状态、运行实例、请求量等信息。

5. 在以上应用验证成功后，可逐步其他应用。

6. 清理迁移配置。 迁移完成后，登录 【TSF 控制台】 ，参照 【配置管理】 动态变更服务注册发现方式，将原有集群的注册中心配置移除。

微服务网关密钥对鉴权使用说明

最近更新时间: 2025-02-18 16:02:00

HTTP 请求头

HTTP 请求头中各参数说明如下：

名称	位置	是否必选	说明
x-mg-traceid	请求/响应	是	请求响应 ID，用于跟踪异常请求调用。
x-mg-secretid	请求	是	授权的 SecretID，用于加签。开启密钥对鉴权时需要。
x-mg-alg	请求/响应	否	加密算法。开启密钥对鉴权时需要。 0：hmac_md5 1：hmac_sha_1 2：hmac_sha_256 3：hmac_sha_512
x-mg-sign	请求/响应	否	签名值。开启密钥对鉴权时需要。
x-mg-nonce	请求/响应	否	随机数。开启密钥对鉴权时需要。
x-mg-code	响应	是	响应码。

签名算法

算法列表

- Hmac_MD5
- HMAC_SHA_1
- HMAC_SHA_256
- HMAC_SHA_512

生成规则

- digetValue = (x-mg-noce)+secretId+secretKey
- signValue = Base64String(签名算法(secretKey, digetValue),"utf-8")

案例说明

签名算法： hmac_md5 secretId: hKhATL/DHVXXXXgeROMrrQ== secretKey: +t9tTMTXXXXUcE+RKOleg== x-mg-noce: D7pAR5fqXXXXx1yacuVzdO digetValue: D7pAR5fqXXXXx1yacuVzdOhKhATL/DHVXXXXgeROMrrQ==+t9tTMTXXXXXUcE+RKOleg== signValue: FE6nLWwYBDXXXXU2CVtFndUBoyg=

校验规则

将请求头中的 x-mg-sign 与服务端根据签名生成规则计算的 signValue 进行比对，判断是否通过鉴权。

代码实现

Java 代码：

```
/**
* 生成签名
*
* @param nonce 随机字符串
* @param secretId 密钥 ID
* @param secretKey 密钥值
* @param algType 签名算法 {@link AlgType}
*/
public static String generate(String nonce, String secretId, String secretKey, AlgType algType) {
String digestValue = nonce + secretId + secretKey;
byte[] serverSignBytes;
switch (algType) {
case HMAC_MD5:
serverSignBytes = HmacUtils.hmacMd5(secretKey, digestValue);
break;
case HMAC_SHA_1:
serverSignBytes = HmacUtils.hmacSha1(secretKey, digestValue);
break;
case HMAC_SHA_256:
serverSignBytes = HmacUtils.hmacSha256(secretKey, digestValue);
break;
case HMAC_SHA_512:
serverSignBytes = HmacUtils.hmacSha512(secretKey, digestValue);
break;
default:
throw new UnsupportedException("不支持的鉴权算法: " + algType);
}
String signValue = Base64.encodeBase64String(serverSignBytes);
if (logger.isDebugEnabled()) {
logger.debug("签名明文：{}，签名密文：{}", digestValue, signValue);
}
return signValue;
}

public static void main(String[] args) {
String secretId = "hKhATL/DHVXXXXgeROMrrQ==";
String secretKey = "+t9tTMTXXXXUcE+RKOleg==";
String nonce = "D7pAR5fqXXXXx1yacuVzdO";
AlgType algType = AlgType.HMAC_SHA_1;
System.out.println(SignUtil.generate(nonce, secretId, secretKey, algType));
}
```

Python 代码：

```
# -*- coding: utf-8 -*-
"""
使用 SHA1 算法生成签名，入参为 SecretId 和 SecetKey
"""
import sys
from hashlib import sha1
import string
import random
import hmac
import base64
```

```
import uuid
seed=
['1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','

nonce=string.join(random.sample(seed,22)).replace(" ","")
signstr=nonce+sys.argv[1]+sys.argv[2]
local_sign_seed = hmac.new(sys.argv[2],signstr , sha1).digest()
sign = base64.b64encode(local_sign_seed)
print ""
print "generate local sign: " +sign
print ""
print "=== http request headers as followed === "
print "x-mg-nonce: "+nonce
print "x-mg-secretid: "+sys.argv[1]
print "x-mg-traceid: "+str(uuid.uuid1())
print "x-mg-alg: 1"
print "x-mg-sign: "+sign
```

使用说明

生成签名

使用上面 Python 代码 生成签名，文件命名为：gen_sign.py。

```
python gen_sign.py {secretId} {secretKey}
```

{secretId} 替换成密钥 ID，{secretKey} 替换成密钥 KEY，示例如下：

```
python gen_sign.py hKhATL/DHVXXXXgeROMrrQ== +t9tTMTXXXXUcE+RKOleg==
```

输出内容：

```
generate local sign: FE6nLWwYBDXXXXU2CVtFndUBoyg=
=== http request headers as followed ===
x-mg-nonce: D7pAR5fqXXXXx1yacuVzdO
x-mg-secretid: hKhATL/DHVXXXXgeROMrrQ==
x-mg-traceid: 8a237eba-71b2-11e9-acee-5254001d2da0
x-mg-alg: 1
x-mg-sign: FE6nLWwYBDXXXXU2CVtFndUBoyg=
```

curl 方式测试

- 测试无鉴权的 API：

```
curl -X {GET} -H 'x-mg-traceid:71b2-11e9-acee-5254001-8a237eba' http://imgcache.finance.cloud.tencent.com:80{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}
```

- 测试鉴权的 API：

```
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://imgcache.
```

```
finance.cloud.tencent.com:80{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}
```

> **注意：**
>
> { } 中的参数请根据实际内容替换。

微服务网关开发指南

最近更新时间: 2025-02-18 16:02:00

准备工作

开始实践微服务网关功能前，请确保您已完成 [SDK 下载]

> **注意：**
>
> - 从1.22.0版本开始，TSF 微服务网关 SDK（TSF-MSGW）提供基于 Zuul 和 SCG (Spring Cloud Gateway) 两种类型的实现。
> - 从1.22.0以前版本进行升级的用户，需要注意 POM 依赖发生了调整。

> -

Zuul

快速上手

使用微服务网关功能前，您需要在 pom.xml 中添加网关依赖项，同时在代码中使用网关开关注解。

1.22.0- Series -RELEASE 以及之后版本

1. 向工程中添加依赖。在 pom.xml 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-msgw-zuul</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 @EnableZuulProxy 和 SpringBootApplication ：

```
// 下面省略了无关的代码
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.zuul.EnableZuulProxy;
@EnableZuulProxy
@SpringBootApplication
public class Application {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```

3. 创建分组，导入 API。

1.22.0-Edgware-RELEASE/1.22.0-Finchley-RELEASE 之前的版本

1. 向工程中添加依赖。在 pom.xml 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-gateway</artifactId>
<version><!-- 1.22.0 之前版本 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsfGateway` ：

```
// 下面省略了无关的代码
import com.tencent.tsf.gateway.core.annotation.EnableTsfGateway;
@EnableTsfGateway
public class Application {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```

3. 创建分组，导入 API。

集成 TSF 其它功能

您在使用微服务网关功能的同时，还可以集成 TSF 其它功能，包括分布式配置、监控、服务治理等，您需要在 `pom.xml` 中添加其对应依赖项，同时在代码中启用注解，具体参考其功能接入文档。以下示例为集成所有功能：

1. 向工程中增加依赖。在 `pom.xml` 中**按顺序**添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-msgw-zuul</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
<!--TSF 其它 SDK 依赖，添加到 msgw-zuul 依赖的后面-->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中增加注解 `@EnableTsf` ：

```
// 下面省略了无关的代码
import com.tencent.tsf.gateway.core.annotation.EnableTsfGateway;
import org.springframework.tsf.annotation.EnableTsf;
@EnableZuulProxy
@SpringBootApplication
@EnableTsf
public class Application {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```

自定义网关过滤器

msgw-zuul SDK 提供通过继承 `TsfGatewayZuulFilter`（或原生 `ZuulFilter`）的方式自定义 Filter，同时需要在类上添加 `@TsfGatewayFilter` 注解（或其它生成Bean方式）。目前 msgw-zuul 基于 zuul1 实现，其 Filter 功能和原生 `ZuulFilter` 保持一致。

```
import static org.springframework.cloud.netflix.zuul.filters.support.FilterConstants.PRE_TYPE;

import com.netflix.zuul.exception.ZuulException;
import com.tencent.tsf.gateway.core.annotation.TsfGatewayFilter;
import com.tencent.tsf.gateway.zuul1.filter.TsfGatewayZuulFilter;

@TsfGatewayFilter
public class TestFilter extends TsfGatewayZuulFilter {

@Override
public String filterType() {
return PRE_TYPE;
}

@Override
public int filterOrder() {
return 100;
}

@Override
public boolean shouldFilter() {
return true;
}

@Override
public Object run() throws ZuulException {
System.out.println("hello world");
return null;
}
}
```

Dubbo 接口协议转换

从1.29.0版本开始，TSF 微服务网关 SDK 支持 Alibaba Dubbo 的泛化调用。 在微服务网关-API管理页面将其导入网关。对应网关分组发布后，即可通过网关的 HTTP 接口进行访问。

请求示例：

```
curl -H 'Content-Type: application/json' -X POST --data "{\"name\":\"xiaoming\"}" <ip>:<port>/<gateway-context>/<namespace-name>/personservice/findNameByPersion
```

其中 为网关分组的访问 Context， 是后端服务的命名空间名称， personservice 为 dubbo 服务名示例，/findNameByPersion 是 dubbo 方法的 API 路径示例。

SCG（Spring Cloud Gateway）

快速上手

使用微服务网关功能前，您需要在 pom.xml 中添加网关依赖项，同时在代码中使用网关开关注解。或参考官方 TSF Demo 中 msgw-demo 模块来编写。

1. 向工程中添加依赖。在 pom.xml 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-msgw-scg</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 @SpringBootApplication ：

```
// 下面省略了无关的代码
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class Application {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```

3. 创建分组，导入 API。

集成 TSF 其它功能

您在使用微服务网关功能的同时，还可以集成 TSF 其它功能，包括分布式配置、监控、服务治理等，您需要在 pom.xml 中添加其对应依赖项，同时在代码中启用注解，具体参考其功能接入文档。以下示例为集成所有功能：

1. 向工程中增加依赖。在 pom.xml 中**按顺序**添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-msgw-scg</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
<!--TSF 其它 SDK 依赖，添加到 msgw-scg 依赖的后面-->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<exclusions>
<exclusion>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
</exclusion>
<exclusion>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-swagger</artifactId>
</exclusion>
</exclusions>
</dependency>
```

2. 向 Application 类中增加注解 `@EnableTsf` ：

```
// 下面省略了无关的代码
import org.springframework.tsf.annotation.EnableTsf;

@SpringBootApplication
@EnableTsf
public class Application {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```

自定义网关过滤器

msgw-scg SDK 提供通过继承 `AbstractTsfGlobalFilter` 的方式，或通过实现原生 `GlobalFilter` 接口的方式自定义 Filter。同时需要在类上添加 `@TsfGatewayFilter` 注解（或其它生成 Bean 方式）。 `AbstractTsfGlobalFilter` 提供了和编写 ZuulFilter 类似的体验：

```
// 下面省略了无关的代码
@TsfGatewayFilter
public class TestFilter extends AbstractTsfGlobalFilter {
@Override
public int getOrder() {
return 100;
}

@Override
public boolean shouldFilter(ServerWebExchange exchange, GatewayFilterChain chain) {
return true;
}

@Override
public Mono<Void> doFilter(ServerWebExchange exchange, GatewayFilterChain chain) {
System.out.println("hello world");
return null;
}
}
```

Dubbo 接口协议转换

从1.29.0版本开始，TSF 微服务网关 SDK 支持 Alibaba Dubbo 的泛化调用。 将 Dubbo API 上报后。在微服务网关-API管理页面将其导入网关。对应网关分组发布后，即可通过网关的 HTTP 接口进行访问。

请求示例：

```
curl -H 'Content-Type: application/json' -X POST --data "{\"name\":\"xiaoming\"}" <ip>:<port>/<gateway-context>/<namespace-name>/personservice/findNameByPersion
```

其中 为网关分组的访问 Context， 是后端服务的命名空间名称， personservice 为 dubbo 服务名示例，/findNameByPersion 是 dubbo 方法的 API 路径示例。

微服务网关配置HTTPS

最近更新时间: 2025-02-18 16:02:00

操作场景

本文档指导您通过 TSF 的微服务网关配置 HTTPS 访问。

操作步骤

步骤1：准备 SSL 证书

您可以通过以下任一种方式准备一个 SSL 证书：

- 方式一：购买（通过证书授权机构购买）
- 方式二：自己生成（通过 keytool 或 openssl 工具生成）

步骤2：配置证书信息

1. 将 SSL 证书放到 resources下 的 https 文件夹（可自定义名称）中：

2. 在配置文件 application.yml 中设置证书的基本信息。如下：

```
# ssl证书相关配置
server：
ssl:
#启用ssl
enabled: true
#证书地址
key-store: classpath:https/keystore.p12
#证书密码
key-store-password: 123456
#证书类型
key-store-type: JKS
```

SpringCloudZuul 与 SpringCloudGateway 配置方式一致，是因为 SpringBoot 服务使用 SSL-HTTPS 的通用方式。

步骤3：访问验证

zuul application.yml 配置如下：

```
server:
port: 18000
ssl:
enabled: true
key-store: classpath:https/keystore.p12
key-store-password: 123456
key-alias: tomcathttps
key-store-type: JKS
```

```
spring:
application:
name: zuul-demo
cloud:
consul:
host: 127.0.0.1
port: 8500
discovery:
heartbeat:
enabled: true
ttl-value: 5
ttl-unit: s
preferIpAddress: true
ribbon:
ReadTimeout: 10000
ConnectTimeout: 5000
zuul:
max:
host:
connections: 500
host:
socket-timeout-millis: 10000
connect-timeout-millis: 5000
routes:
api-v1:
path: /v1/**
serviceId: provider-demo
stripPrefix: true
api-v2:
path: /v2/**
url: http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18081
stripPrefix: true
api-v3:
path: /*/v3/**
serviceId: provider-demo
stripPrefix: true
ignored-patterns: /**/ignore/**
hystrix:
command:
default:
execution:
timeout:
enabled: true
isolation:
thread:
timeoutInMilliseconds: 30000
```

HTTPS 访问网关成功的显示如下：

scg application.yml 配置如下：

```
server:
port: 8080
ssl:
enabled: true
key-store: classpath:https/keystore.p12
```

```
key-store-password: 123456
key-alias: tomcathttps
key-store-type: JKS
error:
include-exception: true
spring:
application:
name: sc-gateway
cloud:
gateway:
discovery:
locator:
enabled: true
lower-case-service-id: false
httpclient:
connectTimeout: 10
responseTimeout: 20s
routes:
- id: api-v1
uri: lb://PROVIDER-DEMO
order: 0
predicates:
- Path=/provider/**
filters:
- StripPrefix=1
# consul:
# host: 127.0.0.1
# port: 8500
# enabled: true
# scheme: HTTP
logging:
file: /var/logs/${spring.application.name}/${spring.application.name}.log
level:
root: INFO
com.tencent.tsf: INFO
```

HTTPS 访问网关成功的显示如下：

常见问题

问题描述：当使用自签名证书时，Chrome 会拦截请求，并展示 ERR_CERT_INVALID，旧版本可以选择跳过，继续访问，但是新版本 Chrome 不允许继续，且提示：您的连接不是私密连接攻击者可能会试图从 XX.XX.XX.XX 窃取您的信息（例如：密码、通讯内容或信用卡信息）。

解决方案：在 Chrome 该页面上，输入 thisisunsafe，即可进行访问，用来验证 HTTPS 配置是否成功。线上环境不推荐使用自签名证书。

组件开发
微服务网关开发
配置兼容开源网关功能

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 微服务网关在开源原生网关 Spring Cloud Gateway 和 Zuul 的基础上增加了命名空间、网关分组、支持 TSF 服务治理、TSF 网关插件以及 TSF 配置 API 限流和超时等 TSF 特有的功能，但增加了这些 TSF 特有的功能以后，一些开源网关的功能使用反而受到了限制，因此需要在使用 TSF 网关的同时兼容使用开源网关的功能，使用户更加灵活方便地扩展一些定制化的功能。

功能对比说明

从1.32.0 版本SDK （2021年6月发布）开始，TSF 微服务网关 SDK 支持兼容使用开源网关功能（包括但不限于支持网关自定义 Routes，支持转发 WebSocket 协议，支持跨域，支持开源网关 API 使用 TSF 服务治理等功能）。

- TSF API 请求：受 TSF 托管，走 TSF 请求的逻辑，会校验网关分组、请求 URL 请求路径等是否合法，支持 TSF 服务治理、支持使用网关 API 限流和超时以及 API 插件等 TSF 特有的功能，但是不支持使用开源网关自定义 Routes 以及支持转发 WebSocket 协议。
- 开源 API 请求：不受 TSF 托管，走开源请求的逻辑，不会校验网关分组和 URL 请求路径等是否合法，可兼容开源原生 API 的逻辑，支持开源网关自定义 Routes、支持转发 WebSocket 协议等，同时可兼容使用 TSF 服务治理功能， 但由于不受 TSF 托管，所以这类请求不支持使用网关 API 限流和超时以及 API 插件等 TSF 特有的功能。

功能	TSF API 请求	开源 API 请求
TSF 服务治理	支持	支持
跨域	支持	支持
TSF 网关 API 限流和超时以及 API 插件等 TSF 特有的功能	支持	不支持
开源网关自定义 Routes	不支持	支持
转发 WebSocket 协议	不支持	支持

前提条件

开始实践配置微服务 SDK 支持兼容使用开源网关功能之前，请确保您已完成 [微服务网关 SDK 使用指南]。

操作步骤

您可以通过以下两种方式配置兼容开源网关功能。

方式一：在微服务网关的请求Header头设置 在微服务网关的请求 Header 头里设置标识某个请求是开源网关 API 的请求。

- 请求的 Header 头里设置 **TSF-Opensource-Mode=true**：表示本次网关请求是开源 API 的请求，走开源请求的逻辑，否则依然走 TSF API 请求（默认都是 TSF API 请求）。
- 请求的Header 头里设置 **TSF-NamespaceId=namespace-xxxxx** ：表示本次网关请求路由到命名空间 ID 为 namespace-xxxxx 的下游服务，如果不设置则默认请求到与网关相同命名空间的下游服务。

> **注意：**
>
> TSF-NamespaceId 填写的是命名空间 ID，而非命名空间名称。

方式二：在微服务网关应用的配置文件设置 在微服务网关应用的配置文件设置 **tsf.gateway.opensource-mode: true。**

> **注意：**
>
> 只有全局命名空间的网关才可以路由到任意一个命名空间的服务，普通命名空间的网关只能路由到与网关相同普通命名空间的服务。

```
tsf:
gateway:
opensource-mode: true
```

> **注意：**
>
> tsf.gateway.opensource-mode 默认为 false，表示默认是 TSF 网关模式，如果设置为 true 表示开启 TSF 兼容开源网关模式。
>
> 开启 TSF 兼容开源网关模式以后，所有非 TSF 网关分组前缀的请求都会被当作是开源 API 的请求，例如某个 TSF 网关分组名是 /test，那么只有 http://imgcache.finance.cloud.tencent.com:80ip:port/test/xxx 的请求才会走原先 TSF API 网关的逻辑，其他请求（如 http://imgcache.finance.cloud.tencent.com:80ip:port/abc/xxx）会走开源 API 请求的逻辑。）会走开源 API 请求的逻辑。

单元化部署（开发指南）

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 支持使用单元化功能以达到让不同的业务流量根据一定的单元化规则分发到指定的单元里，不同单元之间通过微服务网关实现跨单元调用，当某个单元内的服务器实例出现问题时也不会影响到其他单元业务的使用，使得业务受影响粒度达到最小，同时单元化也为业务容灾高可用提供了强有力的保障。

> **注意：**
>
> 单元化网关支持跨命名空间调用，单元化配置仅支持部署在全局命名空间下的网关。
>
> 从1.28.0版本开始，TSF 微服务网关 SDK 支持提供基于 Zuul 的单元化功能。
>
> 微服务网关要么是单元化网关，要么是非单元化网关，开启了单元化的功能，该网关就是单元化网关，否则就是非单元化网关。

前提条件

开始实践单元化功能前，请确保已完成了【SDK 下载】，同时请确保 SDK 版本高于**1.28**。

操作步骤

先暴露 Provider 服务接口，然后将单元化规则配置在调用方 Consumer 工程或微服务网关工程里面。

一、配置Consumer 工程或者微服务网关工程

Consumer 工程

1. 向 Consumer 工程中添加依赖

在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsf` 和 `@ComponentScan`

```
// 下面省略了无关的代码
@SpringBootApplication
@EnableTsf
@EnableFeignClients // 使用Feign微服务调用时请启用
@ComponentScan(basePackages="com.tsf.demo.consumer.*") // 需要扫描单元化@TsfUnitRule注解所在的包
public class ConsumerApplication {
public static void main(String[] args) throws InterruptedException {
SpringApplication.run(ConsumerApplication.class, args);
```

```
}
}
```

3. 配置单元化规则

Feign调用

1. 使用 `@TsfUnitRule` 配置单元化规则。

```
@FeignClient(name = "provider-demo")
public interface ProviderDemoService {

@TsfUnitRule(ruleGenerator = "com.tsf.demo.consumer.util.UserIdGenerator")
@RequestMapping(value = "/echo/unit/{str}", method = RequestMethod.GET)
String echoUnit(@PathVariable("str") String str);
}
```

2. 在代码中编写单元化规则。 需要编写一个类实现接口TsfUnitRuleGenerator的方法generateRule，计算出单元化规则的tag标签值，该类就是步骤3中 `@TsfUnitRule` 的ruleGenerator属性指定的类。

```
// 下面省略了无关的代码
public class UserIdGenerator implements TsfUnitRuleGenerator {

@Override
public Map<String, String> generateRule(Method method, Object[] args) {
String userId = (String)args[0];
Map<String, String> ruleMap = new HashMap<>();
// 添加单元化规则的tag key
ruleMap.put("userId", userId);
return ruleMap;
}

}
```

可以看到，TSF 的单元化功能针对Feign调用只需在需要单元化的类或方法上增加一个注解并添加对应的单元化规则即可（需要该 Bean 类被 Spring 所管理）。

> **注意：**
>
> 注解
>
> `@TsfUnitRule`
>
> 支持配置在类上，也支持配置在方法上。如果是配置在类上，则单元化规则对该类所有的方法都生效。

RestTemplate调用

在 Consumer 中设置 tag ，使用com.tencent.tsf.unit包中的TsfUnitContext类，设置单元化 Tag 的方法

```
@RequestMapping(value = "/echo-rest-unit/{str}", method = RequestMethod.GET)
public String restUnit(@PathVariable String str,
@RequestParam(required = false) String tagName,
@RequestParam(required = false) String tagValue) {
if (!StringUtils.isEmpty(tagName)) {
// 添加单元化规则tag key和value
```

```
TsfUnitContext.putTag(tagName, tagValue);
}
return restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/unit/" + str, String.cl
ass);
}
```

> **注意：**
>
> 设置单元化规则的tag类是com.tencent.tsf.unit.TsfUnitContext，不是org.springframework.tsf.core.TsfContext。

微服务网关工程

开始实践微服务网关单元化功能之前，请确保您已完成 [微服务网关 SDK 使用指南]。

向自定义网关过滤器 Filter 类中添加 tag 标签：

```
// 下面省略了无关的代码
import com.tencent.tsf.gateway.core.annotation.TsfGatewayFilter;
import com.tencent.tsf.unit.TsfUnitContext;

@TsfGatewayFilter
public class TestFilter extends TsfGatewayZuulFilter {

private Logger logger = LoggerFactory.getLogger(TestFilter.class);

@Override
public String filterType() {
return PRE_TYPE;
}

@Override
public int filterOrder() {
// 注：设置 putTag 的filter的order必须在ZUUL_UNIT_ORDER之前
return TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER - 1;
}

@Override
public boolean shouldFilter() {
return true;
}

@Override
public Object run() throws ZuulException {
// 单元化场景的putTag操作
RequestContext ctx = RequestContext.getCurrentContext();
HttpServletRequest request = ctx.getRequest();
String userId = request.getHeader("userId");

if(StringUtils.isNotBlank(userId)) {
logger.info("put unit tag userId:{}", userId);
TsfUnitContext.putTag("userId", userId);
}

System.out.println("hello world");
logger.info("hello world");
return null;
```

```
}
}
```

- 使用包 com.tencent.tsf.unit 下的类 TsfUnitContext 的方法 putTag 来设置单元化标签
- 自定义设置 putTag 的 Filter 的顺序必须在 TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER 之前

开启单元化功能并创建单元化规则

在TSF控制台开启单元化功能并创建单元化规则，详细操作参考【单元化部署】。

配置单元化

最近更新时间: 2025-02-18 16:02:00

准备工作

开始实践微服务网关单元化功能之前，请确保您已完成 [微服务网关 SDK 使用指南]。

> **注意：**
>
> - 单元化网关支持跨命名空间调用，单元化配置仅支持部署在全局命名空间下的网关。
> - 从1.28.0版本开始，TSF 微服务网关 SDK 支持提供基于 Zuul 的单元化功能。
> - 微服务网关要么是单元化网关，要么是非单元化网关，开启了单元化的功能，该网关就是单元化网关，否则就是非单元化网关。

Zuul

微服务网关 SDK 使用单元化功能

1. 向自定义网关过滤器 Filter 类中添加 tag 标签：

```
// 下面省略了无关的代码
import com.tencent.tsf.gateway.core.annotation.TsfGatewayFilter;
import com.tencent.tsf.unit.TsfUnitContext;
```

@TsfGatewayFilter public class TestFilter extends TsfGatewayZuulFilter {

```
private Logger logger = LoggerFactory.getLogger(TestFilter.class);

@Override
public String filterType() {
return PRE_TYPE;
}

@Override
public int filterOrder() {
// 注：设置 putTag 的filter的order必须在ZUUL_UNIT_ORDER之前
return TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER - 1;
}

@Override
public boolean shouldFilter() {
return true;
}

@Override
public Object run() throws ZuulException {
// 单元化场景的putTag操作
RequestContext ctx = RequestContext.getCurrentContext();
HttpServletRequest request = ctx.getRequest();
String userId = request.getHeader("userId");
```

```
if(StringUtils.isNotBlank(userId)) {
logger.info("put unit tag userId:{}", userId);
TsfUnitContext.putTag("userId", userId);
}

System.out.println("hello world");
logger.info("hello world");
return null;
}
```

}

- 使用包 com.tencent.tsf.unit 下的类 TsfUnitContext 的方法 putTag 来设置单元化标签
- 自定义设置 putTag 的 Filter 的顺序必须在 TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER 之前

2. 开启单元化功能并创建单元化规则。

开启单元化功能并创建单元化规则

操作步骤

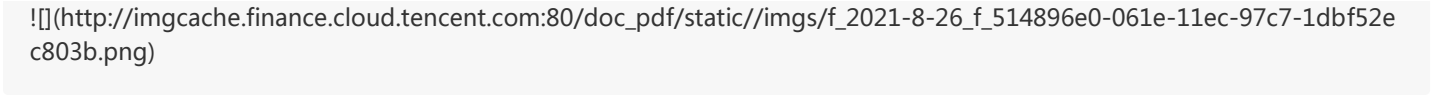
1. 登录 【TSF 控制台】，在左侧菜单栏选择【微服务网关】>【网关管理】。
2. 在网关管理页面，点击需要配置单元化规则的微服务网关，进入该微服务网关相关的配置页面，在上方菜单栏中的【基本信息】的"单元化配置"中，点击右侧的"编辑"来开启微服务网关的单元化功能。

3. 开启单元化功能以后，刷新该微服务网关相关的配置页面，此时在页面上方的菜单栏中会出现菜单【单元范围】和【单元化规则】。
4. 点击菜单【单元范围】的按钮"关联命名空间"，选择需要关联到该单元下的命名空间（可通过单元化规则路由到该命名空间），点击"提交"按钮。

5. 点击菜单【单元化规则】的按钮"新建规则"，填写对应的单元化规则，最后选择目的地命名空间（该命名空间是单元范围中的其中一个命名空间），点击"提交"按钮。

- 单元化规则之间的按顺序匹配，即如果未命中规则1，则会继续往下去匹配规则2，如果未命中规则2，则再继续往下去匹配规则3，直到命中规则为止。
- 假如没有命中任何一条规则且在开启了该单元化规则的情况下，那么以单元化形式调用的请求会直接报错（单元化形式调用的请求指的是客户端 putTag 了但该 tag 未命中规则或者直接以HTTP方式调用了单元化网关但没有 putTag 的请求）。
- 单元规则中的标签配置之间有"且"和"或"的逻辑关系可供选择。

6. 最后，在菜单【单元化规则】列表中选择一个需要生效的单元化规则在其"状态"列点击开启即可。



微服务网关开发指南

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 微服务网关在开源原生网关 Spring Cloud Gateway 和 Zuul 的基础上增加了命名空间、网关分组、支持 TSF 服务治理、TSF 网关插件以及 TSF 配置 API 限流和超时等 TSF 特有的功能，但增加了这些 TSF 特有的功能以后，一些开源网关的功能使用反而受到了限制，因此需要在使用 TSF 网关的同时兼容使用开源网关的功能，使用户更加灵活方便地扩展一些定制化的功能。

功能对比说明

从1.32.0 版本SDK （2021年6月发布）开始，TSF 微服务网关 SDK 支持兼容使用开源网关功能（包括但不限于支持网关自定义 Routes，支持转发 WebSocket 协议，支持跨域，支持开源网关 API 使用 TSF 服务治理等功能）。

- TSF API 请求：受 TSF 托管，走 TSF 请求的逻辑，会校验网关分组、请求 URL 请求路径等是否合法，支持 TSF 服务治理、支持使用网关 API 限流和超时以及 API 插件等 TSF 特有的功能，但是不支持使用开源网关自定义 Routes 以及支持转发 WebSocket 协议。
- 开源 API 请求：不受 TSF 托管，走开源请求的逻辑，不会校验网关分组和 URL 请求路径等是否合法，可兼容开源原生 API 的逻辑，支持开源网关自定义 Routes、支持转发 WebSocket 协议等等，同时可兼容使用 TSF 服务治理功能， 但由于不受 TSF 托管，所以这类请求不支持使用网关 API 限流和超时以及 API 插件等 TSF 特有的功能。

功能	TSF API 请求	开源 API 请求
TSF 服务治理	支持	支持
跨域	支持	支持
TSF 网关 API 限流和超时以及 API 插件等 TSF 特有的功能	支持	不支持
开源网关自定义 Routes	不支持	支持
转发 WebSocket 协议	不支持	支持

前提条件

开始实践配置微服务 SDK 支持兼容使用开源网关功能之前，请确保您已完成 [微服务网关 SDK 使用指南]。

操作步骤

方法一、在微服务网关的请求header头里设置标识某个请求是开源网关 API 的请求

（1）请求的Header 头里设置 **TSF-Opensource-Mode=true**：表示本次网关请求是开源 API 的请求，走开源请求的逻辑，否则依然走 TSF API 请求（默认都是 TSF API 请求）。

（2）请求的Header 头里设置 **TSF-NamespaceId=namespace-xxxxx** ：表示本次网关请求路由到命名空间 ID 为 namespace-xxxxx 的下游服务，如果不设置则默认请求到与网关相同命名空间的下游服务。

> **注意：**

- 只有全局命名空间的网关才可以路由到任意一个命名空间的服务，普通命名空间的网关只能路由到与网关相同普通命名空间的服务。
- TSF-NamespaceId 填写的是命名空间 ID，而非命名空间名称。

方法二、在微服务网关应用的配置文件设置 **tsf.gateway.opensource-mode: true**

```
tsf:
gateway:
opensource-mode: true
```

注意：

- tsf.gateway.opensource-mode 默认为 false，表示默认是 TSF 网关模式，如果设置为 true 表示开启 TSF 兼容开源网关模式。
- 开启 TSF 兼容开源网关模式以后，所有非 TSF 网关分组前缀的请求都会被当作是开源 API 的请求，比如某个 TSF 网关分组名是 /test，那么只有 http://imgcache.finance.cloud.tencent.com:80ip:port/test/xxx 的请求才会走原先 TSF API网关的逻辑，其他请求，比如 http://imgcache.finance.cloud.tencent.com:80ip:port/abc/xxx 的请求会走开源 API 请求的逻辑。

微服务网关密钥对鉴权使用说明

最近更新时间: 2025-02-18 16:02:00

HTTP 请求头

HTTP 请求头中各参数说明如下：

名称	位置	是否必选	说明
x-mg-traceid	请求/响应	是	请求响应 ID，用于跟踪异常请求调用。
x-mg-secretid	请求	是	授权的 SecretID，用于加签。开启密钥对鉴权时需要。
x-mg-alg	请求/响应	否	加密算法。开启密钥对鉴权时需要。 0：hmac_md5 1：hmac_sha_1 2：hmac_sha_256 3：hmac_sha_512
x-mg-sign	请求/响应	否	签名值。开启密钥对鉴权时需要。
x-mg-nonce	请求/响应	否	随机数。开启密钥对鉴权时需要。
x-mg-code	响应	是	响应码。

签名算法

算法列表

- Hmac_MD5
- HMAC_SHA_1
- HMAC_SHA_256
- HMAC_SHA_512

生成规则

- digetValue = (x-mg-noce)+secretId+secretKey
- signValue = Base64String(签名算法(secretKey, digetValue),"utf-8")

案例说明

签名算法： hmac_md5 secretId: hKhATL/DHVXXXXgeROMrrQ== secretKey: +t9tTMTXXXXUcE+RKOleg== x-mg-noce: D7pAR5fqXXXXx1yacuVzdO digetValue: D7pAR5fqXXXXx1yacuVzdOhKhATL/DHVXXXXgeROMrrQ==+t9tTMTXXXXXUcE+RKOleg== signValue: FE6nLWwYBDXXXXU2CVtFndUBoyg=

校验规则

将请求头中的 x-mg-sign 与服务端根据签名生成规则计算的 signValue 进行比对，判断是否通过鉴权。

代码实现

Java 代码：

```
/**
* 生成签名
*
* @param nonce 随机字符串
* @param secretId 密钥 ID
* @param secretKey 密钥值
* @param algType 签名算法 {@link AlgType}
*/
public static String generate(String nonce, String secretId, String secretKey, AlgType algType) {
String digestValue = nonce + secretId + secretKey;
byte[] serverSignBytes;
switch (algType) {
case HMAC_MD5:
serverSignBytes = HmacUtils.hmacMd5(secretKey, digestValue);
break;
case HMAC_SHA_1:
serverSignBytes = HmacUtils.hmacSha1(secretKey, digestValue);
break;
case HMAC_SHA_256:
serverSignBytes = HmacUtils.hmacSha256(secretKey, digestValue);
break;
case HMAC_SHA_512:
serverSignBytes = HmacUtils.hmacSha512(secretKey, digestValue);
break;
default:
throw new UnsupportedException("不支持的鉴权算法: " + algType);
}
String signValue = Base64.encodeBase64String(serverSignBytes);
if (logger.isDebugEnabled()) {
logger.debug("签名明文：{}，签名密文：{}", digestValue, signValue);
}
return signValue;
}

public static void main(String[] args) {
String secretId = "hKhATL/DHVXXXXgeROMrrQ==";
String secretKey = "+t9tTMTXXXXUcE+RKOleg==";
String nonce = "D7pAR5fqXXXXx1yacuVzdO";
AlgType algType = AlgType.HMAC_SHA_1;
System.out.println(SignUtil.generate(nonce, secretId, secretKey, algType));
}
```

Python 代码：

```
# -*- coding: utf-8 -*-
"""
使用 SHA1 算法生成签名，入参为 SecretId 和 SecetKey
"""
import sys
from hashlib import sha1
import string
import random
import hmac
import base64
```

```
import uuid
seed=
['1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','

nonce=string.join(random.sample(seed,22)).replace(" ","")
signstr=nonce+sys.argv[1]+sys.argv[2]
local_sign_seed = hmac.new(sys.argv[2],signstr , sha1).digest()
sign = base64.b64encode(local_sign_seed)
print ""
print "generate local sign: " +sign
print ""
print "=== http request headers as followed === "
print "x-mg-nonce: "+nonce
print "x-mg-secretid: "+sys.argv[1]
print "x-mg-traceid: "+str(uuid.uuid1())
print "x-mg-alg: 1"
print "x-mg-sign: "+sign
```

使用说明

生成签名

使用上面 Python 代码 生成签名，文件命名为：gen_sign.py。

```
python gen_sign.py {secretId} {secretKey}
```

{secretId} 替换成密钥 ID，{secretKey} 替换成密钥 KEY，示例如下：

```
python gen_sign.py hKhATL/DHVXXXXgeROMrrQ== +t9tTMTXXXXUcE+RKOleg==
```

输出内容：

```
generate local sign: FE6nLWwYBDXXXXU2CVtFndUBoyg=
=== http request headers as followed ===
x-mg-nonce: D7pAR5fqXXXXx1yacuVzdO
x-mg-secretid: hKhATL/DHVXXXXgeROMrrQ==
x-mg-traceid: 8a237eba-71b2-11e9-acee-5254001d2da0
x-mg-alg: 1
x-mg-sign: FE6nLWwYBDXXXXU2CVtFndUBoyg=
```

curl 方式测试

- 测试无鉴权的 API：

```
curl -X {GET} -H 'x-mg-traceid:71b2-11e9-acee-5254001-8a237eba' http://imgcache.finance.cloud.tencent.com:80{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}
```

- 测试鉴权的 API：

```
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://imgca
```

```
che.finance.cloud.tencent.com:80{gatewayIp}:{gatewayPort}/{groupContext}/{namespaceName}/{serviceName}/{apiPath}
```

> **注意：**
>
> { } 中的参数请根据实际内容替换。

微服务网关配置HTTPS

最近更新时间: 2025-02-18 16:02:00

操作场景

本文档指导您通过 TSF 的微服务网关配置 HTTPS 访问。

操作步骤

步骤1：准备 SSL 证书

您可以通过以下任一种方式准备一个 SSL 证书：

- 方式一：购买（通过证书授权机构购买）
- 方式二：自己生成（通过 keytool 或 openssl 工具生成）

步骤2：配置证书信息

1. 将 SSL 证书放到 resources下 的 https 文件夹（可自定义名称）中：

2. 在配置文件 application.yml 中设置证书的基本信息。如下：

```
# ssl证书相关配置
server：
ssl:
#启用ssl
enabled: true
#证书地址
key-store: classpath:https/keystore.p12
#证书密码
key-store-password: 123456
#证书类型
key-store-type: JKS
```

> **注意：**
>
> SpringCloudZuul 与 SpringCloudGateway 配置方式一致，是因为 SpringBoot 服务使用 SSL-HTTPS 的通用方式。

步骤3：访问验证

zuul application.yml 配置如下：

```
server:
port: 18000
ssl:
enabled: true
```

```
key-store: classpath:https/keystore.p12
key-store-password: 123456
key-alias: tomcathttps
key-store-type: JKS
spring:
application:
name: zuul-demo
cloud:
consul:
host: 127.0.0.1
port: 8500
discovery:
heartbeat:
enabled: true
ttl-value: 5
ttl-unit: s
preferIpAddress: true
ribbon:
ReadTimeout: 10000
ConnectTimeout: 5000
zuul:
max:
host:
connections: 500
host:
socket-timeout-millis: 10000
connect-timeout-millis: 5000
routes:
api-v1:
path: /v1/**
serviceId: provider-demo
stripPrefix: true
api-v2:
path: /v2/**
url: http://imgcache.finance.cloud.tencent.com:80127.0.0.1:18081
stripPrefix: true
api-v3:
path: /*/v3/**
serviceId: provider-demo
stripPrefix: true
ignored-patterns: /**/ignore/**
hystrix:
command:
default:
execution:
timeout:
enabled: true
isolation:
thread:
timeoutInMilliseconds: 30000
```

HTTPS 访问网关成功的显示如下：

scg application.yml 配置如下：

```
server:
port: 8080
ssl:
enabled: true
key-store: classpath:https/keystore.p12
key-store-password: 123456
key-alias: tomcathttps
key-store-type: JKS
error:
include-exception: true
spring:
application:
name: sc-gateway
cloud:
gateway:
discovery:
locator:
enabled: true
lower-case-service-id: false
httpclient:
connectTimeout: 10
responseTimeout: 20s
routes:
- id: api-v1
uri: lb://PROVIDER-DEMO
order: 0
predicates:
- Path=/provider/**
filters:
- StripPrefix=1
# consul:
# host: 127.0.0.1
# port: 8500
# enabled: true
# scheme: HTTP
logging:
file: /var/logs/${spring.application.name}/${spring.application.name}.log
level:
root: INFO
com.tencent.tsf: INFO
```

HTTPS 访问网关成功的显示如下：

常见问题

问题描述：当使用自签名证书时，Chrome 会拦截请求，并展示 ERR_CERT_INVALID，旧版本可以选择跳过，继续访问，但是新版本 Chrome 不允许继续，且提示：您的连接不是私密连接攻击者可能会试图从 XX.XX.XX.XX 窃取您的信息（例如：密码、通讯内容或信用卡信息）。

解决方案：在 Chrome 该页面上，输入 thisisunsafe，即可进行访问，用来验证 HTTPS 配置是否成功。线上环境不推荐使用自签名证书。

通用开发指引
SDK下载

最近更新时间: 2025-02-18 16:02:00

以下视频将为您介绍 TSF 应用开发环境中，SDK 安装的基本流程和步骤：

http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/edu/learning/quick-play/2039-24415?source=gw.doc.media&withPoster=1&notip=1

开发前准备

在执行安装脚本之前，请确保您的机器上已经安装了 Java 和 Maven。

1. 安装 Java

1.1 检查 Java 安装

打开终端，执行如下命令：

```
java -version
```

如果输出 Java 版本号，说明 Java 安装成功；如果没有安装 Java，请 下载安装 Java 软件开发套件（JDK）。

1.2 设置 Java 环境

设置 JAVA_HOME 环境变量，并指向您机器上的 Java 安装目录。 以 Java 1.6.0_21 版本为例，操作系统的输出如下：

操作系统	输出
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdk1.6.0_21
Linux	export JAVA_HOME=/usr/local/java-current
Mac OSX	export JAVA_HOME=/Library/Java/Home

将 Java 编译器地址添加到系统路径中。

操作系统	输出
Windows	将字符串";C:\Program Files\Java\jdk1.6.0_21\bin"添加到系统变量"Path"的末尾
Linux	export PATH=$PATH:$JAVA_HOME/bin/
Mac OSX	not required

使用上面提到的 **java -version** 命令验证 Java 安装。

2. 安装 Maven

2.1 下载 安装 Maven

参考 Maven 下载。

2.2 设置 MAVEN_HOME 和 PATH 环境变量

- Windows 系统下

```
新建系统变量 MAVEN_HOME 变量值：E:\Maven\apache-maven-3.3.9
编辑系统变量 Path 添加变量值： ;%MAVEN_HOME%\bin
```

- Linux、macOS 系统下

```
export MAVEN_HOME=/usr/local/maven/apache-maven-3.3.9
export PATH=$MAVEN_HOME/bin:$PATH
```

2.3 验证 Maven 安装

当 Maven 安装完成后， 通过执行如下命令验证 Maven 是否安装成功。

```
mvn --version
```

若出现正常的版本号信息后，说明 Maven 安装成功。

3. Maven 配置 TSF 私服地址

3.1 添加私服配置

找到 Maven 所使用的配置文件，一般在 `~/.m2/settings.xml` 中，在 settings.xml 中加入如下配置 ：

您也可以下载 setting.xml 样例文件>> （鼠标右键另存为链接）。

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/SETTINGS/1.0.0"
xmlns:xsi="http://imgcache.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/SETTINGS/1.0.0 http://imgcache.fin
ance.cloud.tencent.com:80maven.apache.org/xsd/settings-1.0.0.xsd">
<!-- localRepository
| The path to the local repository maven will use to store artifacts.
|
| Default: ${user.home}/.m2/repository
<localRepository>/path/to/local/repo</localRepository>
-->

<pluginGroups></pluginGroups>
<proxies></proxies>
<servers></servers>
<mirrors></mirrors>

<profiles>
<profile>
<id>nexus</id>
<repositories>
<repository>
<id>central</id>
<url>http://imgcache.finance.cloud.tencent.com:80repo1.maven.org/maven2</url>
<releases>
```

```
<enabled>true</enabled>
</releases>
<snapshots>
<enabled>true</enabled>
</snapshots>
</repository>
</repositories>
<pluginRepositories>
<pluginRepository>
<id>central</id>
<url>http://imgcache.finance.cloud.tencent.com:80repo1.maven.org/maven2</url>
<releases>
<enabled>true</enabled>
</releases>
<snapshots>
<enabled>true</enabled>
</snapshots>
</pluginRepository>
</pluginRepositories>
</profile>
<profile>
<id>qcloud-repo</id>
<repositories>
<repository>
<id>qcloud-central</id>
<name>qcloud mirror central</name>
<url>http://imgcache.finance.cloud.tencent.com:80mirrors.cloud.tencent.com/nexus/repository/maven-public/</url>
<snapshots>
<enabled>true</enabled>
</snapshots>
<releases>
<enabled>true</enabled>
</releases>
</repository>
</repositories>
<pluginRepositories>
<pluginRepository>
<id>qcloud-plugin-central</id>
<url>http://imgcache.finance.cloud.tencent.com:80mirrors.cloud.tencent.com/nexus/repository/maven-public/</url>
<snapshots>
<enabled>true</enabled>
</snapshots>
<releases>
<enabled>true</enabled>
</releases>
</pluginRepository>
</pluginRepositories>
</profile>
</profiles>

<activeProfiles>
<activeProfile>nexus</activeProfile>
<activeProfile>qcloud-repo</activeProfile>
</activeProfiles>

</settings>
```

3.2 验证配置是否成功

在命令行执行如下命令：

`mvn help:effective-settings` 。

- 查看执行结果，没有错误表明 setting.xml 格式正确。
- profiles 中包含 qcloud-repo ，则表明 qcloud-repo 私服已经加入到 profiles 中；activeProfiles 中包含 qcloud-repo，则表明 qcloud-repo 私服已经激活成功。可以通过 `mvn help:effective-settings | grep &#39;qcloud-repo&#39;` 命令检查。

> **注意：**
>
> 执行正确的 Maven 命令后，如果无法下载 qcloud 相关依赖包，请重启 IDE，或者检查 IDE Maven 相关配置。

安装 SDK

在您的Java工程的 `pom.xml` 所在目录执行 `mvn clean package` 即可下载 TSF SDK。

> **注意：**
>
> 如果无法下载相关依赖，请检查网络是否有防火墙限制。

YAML 格式介绍

最近更新时间: 2025-02-18 16:02:00

YAML 专门用来写配置文件的语言。

语法规则

YAML 的基本语法规则如下：

- 大小写敏感。
- 使用缩进表示层级关系。
- 缩进时**不允许**使用 Tab 键，只允许使用空格。
- 缩进的空格数目不重要，只要相同层级的元素左侧对齐即可。

数据结构

YAML 支持三种数据结构：对象、数组和纯量。

- 对象：键值对的集合，又称为映射（mapping）/ 哈希（hashes） / 字典（dictionary）
- 数组：一组按次序排列的值，又称为序列（sequence） / 列表（list）
- 纯量（scalars）：单个的、不可再分的值

对象

简单对象

```
foo: whatever
bar: stuff
```

复杂对象:

fruit: apple name: steve sport: baseball

- more
- python: rocks perl: papers ruby: scissorses

转换为 JavaScript 代码后： javascript { foo: 'whatever', bar: [{ fruit: 'apple', name: 'steve', sport: 'baseball' }, 'more', { python: 'rocks', perl: 'papers', ruby: 'scissorses' }] }

数组

```
- Cat
- Dog
- Goldfish
```

纯量

纯量是最基本的、不可再分的值。

```
- 字符串
- 布尔值
- 整数
- 浮点数
- Null
- 时间
- 日期
```

字符串

字符串是比较复杂的类型，举例说明：

```
str: 这是一行字符串
```

如果字符串之中包含空格或特殊字符，需要放在引号之中。

```
str: '内容: 字符串'
```

单引号和双引号都可以使用，双引号不会对特殊字符转义。

```
s1: '内容\n字符串' # 会对 \n 字符转义
s2: "内容\n字符串" # 不会对 \n 字符转义
```

多行字符串可以使用 | 保留换行符，也可以使用 > 折叠换行。 yaml this: | Foo Bar that: > Foo Bar

转换为 JavaScript 代码：

```
{ this: 'Foo\nBar\n', that: 'Foo Bar\n' }
```

工具

- 提供了一个 YAML 的格式校验工具，供参考
- YAML 和 Properties 格式互转工具

参考

- YAML 语言教程 - 阮一峰
- Yaml Cookbook：提供了很多典型的 YAML 用例
- YAML Syntax - Ansible：写了一些 YAML 的常见陷阱

如何打 FatJar 包

最近更新时间: 2025-02-18 16:02:00

FatJar 是一种可执行的 Jar 包（Executable Jar）。FatJar 和普通的 Jar 不同在于它包含了依赖的 Jar 包。

添加 FatJar 打包方式

在工程的 pom.xml 文件中添加插件：

```
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
```

打包 FatJar 文件

添加完插件后，在工程的主目录下，使用 maven 命令 `mvn clean package` 进行打包，即可在 target 目录下找到打包好的 FatJar 文件。

服务编排基本操作

最近更新时间: 2025-02-18 16:02:00

操作场景

为提高用户开发微服务代码的效率， TSF 提供服务编排功能。服务编排可以实现快速生成基于 TSF 框架的微服务工程源码，帮助业务设计人员快速构建可运行的微服务程序。

TSF 引入了服务编排模板的概念用于保存工程信息，方便后续可以基于模板修改工程参数和下载工程源码。

操作步骤

新建服务编排模板

1. 登录 【TSF 控制台】。
2. 单击左侧导航**服务编排**。
3. 单击 【新建模板】。
4. 填写工程配置和 POM 配置信息。

- **工程名**：Spring Boot 工程名，字母开头，支持大小写字母、数字组成，不超过24个字符长度。
- **包路径**：package 路径，小写字母开头，支持小写字母、数字或小数点组成，不超过60个字符长度。
- **GroupID**： pom.xml 文件中的 groupId。
- **Artifact ID**： pom.xml 文件中的 artifactId。
- **Name**：pom.xml 文件中的 name。
- **Version**：pom.xml 文件中的 version。
- **Description**：选填，pom.xml 文件中的 description。

5. 填写服务基本信息：可填写多个

- **服务名**：对应工程文件中的 `spring.application.name` 。
- **端口**：服务监听端口，仅支持0 - 65535。
- **Controller 类名前缀**：可填写多个，将会生成多个带有前缀的 Controller 类。

6. 调用方式：选择 Feign、RestTemplate 或 AsyncRestTemplate。
7. Controller 调用关系（仅服务个数多余1时可配置）：可填写多个，将在生成的工程中显示服务之间以 `/echo` 接口的调用逻辑代码。最多支持配置5个。

- 主调服务名：选择主调服务的名称
- 主调 Controller 类：选择主调服务的 Controller 类。
- 被调服务名：选择被调服务的名称，不能与主调服务名称重复。
- 被调 Controller 类：选择被调服务的 Controller 类。

8. 单击【保存并下载】，保存模板，并下载工程的 zip 文件。或者单击【保存】，仅保存模板，不执行下载操作。

> **注意：**
>
> 界面将会根据 Controller 调用关系实时生成桑基图，以便用户更直观地看到服务相互间的依赖关系。

修改服务编排模板

1. 单击服务编排模板列表上目标模板的名称。

2. 可修改模板的参数信息，各参数字段含义参考 新建操作。

删除服务编排模板

1. 单击服务编排模板列表上目标模板右侧的【删除】。

2. 在弹框中单击【确认】，删除服务编排模板。

使用模板工程

最近更新时间: 2025-02-18 16:02:00

项目整体结构

命名	描述
tsf-projectName	TSF 总工程
projectName-parent	工程父依赖
projectName-common	TSF 公共基础组件（Jar 包插件，非微服务）
projectName-xxx	微服务模块

包命名规范

基础包路径(packageBase)：{orgnization}.{工程名}.{模块名}.{子模块名}

模块内结构：

包	存放内容
packageBase	模型
packageBase.constant	常量、枚举等
packageBase.controller	控制器
packageBase.service	服务接口
packageBase.service.impl	服务实现
packageBase.proxy	远程接口或代理
packageBase.dao	数据层操作
packageBase.exception	异常

项目部署准备

1. 参考 【SDK 下载】下载 TSF 相关依赖（SDK），并安装到本地 Maven 仓库中。
2. 修改工程的 pom.xml ，检查并修改 `projectName-parent` 工程的 pom.xml 的**依赖项**和**版本号**与步骤1中下载的依赖项相同。

3. 参考官网文档 【轻量级服务注册中心】进行本地联调。

项目访问路径

被调服务开放地址：ip:port/被调服务controller前缀/{str} 主调服务访问地址：ip:port/主调服务controller前缀/{str}

生成工程的 SDK 版本说明

目前服务编排功能使用的 TSF SDK 可能和最新的 SDK 不一致，如果您希望使用最新的 SDK，可以参考开发手册 【SDK 下载】将最新依赖包安装到本地，并更新工程的 `pom.xml` 文件。

制作容器镜像-KonaJDK

最近更新时间: 2025-02-18 16:02:00

该任务指导您通过 Spring Cloud 和 Mesh 两种方式制作容器镜像。

准备构建材料

Spring Cloud 应用构建材料

1. 简化版本

简化版本的 Dockerfile 不包含文件配置和 JVM 监控功能，仅需要用户替换掉 Dockerfile 中 Spring Cloud 应用 jar 包名称，您也可以先试用 TSF 提供的 Spring Cloud 应用 Demo JAR 包（下载地址）。

关于JDK版本，推荐使用 Tencent KonaJDK，请下载 KonaJDK 安装文件（下载地址）。

> **注意：**
>
> 在 Spring Cloud 应用 JAR 包同级目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

2. 使用 JVM 监控功能

如果您希望使用 [JVM 监控] 功能，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.0`（下载地址），然后在 CMD 命令中启动该组件。

> **注意：**
>
> 将 Spring Cloud 应用 JAR 包和 JVM 监控组件放在同级目录下，并在该目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
```

```
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar 包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JVM 监控组件要和您的 Dockerfile 位于同一级目录，并创建 JVM 监控数据采集目录
ENV agentjar TencentCloudJvmMonitor-1.1.0-RELEASE.jar
# 若容器的基础版本为 非 gnu-libc 版本，如 Alpine，请添加如下语句
# RUN ln -sf /lib/libc.musl-x86_64.so.1 /lib/ld-linux-x86-64.so.2
COPY ${agentjar} ${workdir}

RUN mkdir -p /data/tsf_apm/monitor/jvm-metrics/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 使用 JVM监控功能需要加上 gclog 和 javaagent 的配置, 否则将无法提供 jvm 监控能力
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "exec java -Xloggc:/data/tsf_apm/monitor/jvm-metrics/gclog.log -XX:+PrintGCDateStamps -XX:+PrintGC
Details -verbose:gc -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=8 -XX:GCLogFileSize=50M -javaagent:${workdi
r}/${agentjar}=hascontroller=true ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

3. 使用文件配置

如果您希望使用 TSF [文件配置] 功能，则需要在 Dockerfile 中增加文件配置组件 `tsf-consul-template-docker.tar.gz`（下载地址），然后在 CMD 启动命令中启动该组件。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
#安装 KonaJDK
ADD ./java-8-konajdk.rpm /java-8-konajdk.rpm
RUN yum update -y && yum install -y java-8-konajdk.rpm

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# tsf-consul-template-docker 用于文件配置功能，如不需要可注释掉该行
ADD tsf-consul-template-docker.tar.gz /root/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 考虑到容器场景对于内存的要求，建议添加-Xshare:off选项关闭CDS功能
CMD ["sh", "-ec", "sh /root/tsf-consul-template-docker/script/start.sh; exec java ${JAVA_OPTS} -Xshare:off -jar ${jar}"]
```

私有化版本使用建议：

私有化的 TSF 要支持 stdout 日志，需要在启动命令中将 stdout 及 stderr 重定向到一个文件中。将上文的 CMD 一行替换成：

```
RUN mkdir -p /data/tsf_std/stdout/logs
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -Xshare:off -jar ${jar} 2>&1 > /data/tsf_std/stdout/logs/sys_log.log"]
```

Mesh 应用构建材料

1. 下载 Mesh 应用 Demo 包：userService.tar.gz。

2. 在该 tar.gz 包同级目录下，编写 Dockerfile 文件：

```
FROM centos:7
RUN mkdir /root/app/
# 其中 userService.tar.gz 是 Mesh 应用压缩包
ADD userService.tar.gz /root/app/
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENTRYPOINT ["bash","/root/app/userService/start.sh"]
```

Mesh 应用压缩包解压后的文件目录结构及文件规范参考 【Mesh Demo 介绍】。

使用文件配置功能

如果容器应用需要使用 TSF 文件配置功能，需要修改 Dockerfile，具体使用指引参考 【文件配置>前提条件】。

构建镜像

1. 在 Dockerfile 所在目录执行 build 命令：

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 <主账号 ID> 对应用户云平台的**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在云平台个人信息页面获取），<应用名> 表示控制台上的应用名。 tag 为镜像的 tag，用户可自定义。

2. 命令执行完成后，通过 docker image ls 查看创建的镜像。

制作容器镜像

最近更新时间: 2025-02-18 16:02:00

该任务指导您通过 Spring Cloud 和 Mesh 两种方式制作容器镜像。

如需使用 Tencent KonaJDK 替换 openJDK，请查看 [制作容器镜像-KonaJDK] 说明。

准备构建材料

Spring Cloud 应用构建材料

1. 简化版本

简化版本的 Dockerfile 不包含文件配置和 JVM 监控功能，仅需要用户替换掉 Dockerfile 中 Spring Cloud 应用 jar 包名称，您也可以先试用 TSF 提供的 Spring Cloud 应用 Demo JAR 包（下载地址）。

> **注意：**
>
> 在 Spring Cloud 应用 JAR 包同级目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk

# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -jar ${jar}"]
```

2. 使用 JVM 监控功能

如果您希望使用 [JVM 监控] 功能，则需要在 Dockerfile 中增加 JVM 监控组件 `TencentCloudJvmMonitor-1.1.0`（下载地址），然后在 CMD 命令中启动该组件。

> **注意：**
>
> 将 Spring Cloud 应用 JAR 包和 JVM 监控组件放在同级目录下，并在该目录下编写 Dockerfile。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
```

```
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar 包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# JVM 监控组件要和您的 Dockerfile 位于同一级目录，并创建 JVM 监控数据采集目录
ENV agentjar TencentCloudJvmMonitor-1.1.0-RELEASE.jar
# 若容器的基础版本为 非 gnu-libc 版本，如 Alpine，请添加如下语句
# RUN ln -sf /lib/libc.musl-x86_64.so.1 /lib/ld-linux-x86-64.so.2
COPY ${agentjar} ${workdir}

RUN mkdir -p /data/tsf_apm/monitor/jvm-metrics/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
# 使用 JVM监控功能需要加上 gclog 和 javaagent 的配置, 否则将无法提供 jvm 监控能力
CMD ["sh", "-ec", "exec java -Xloggc:/data/tsf_apm/monitor/jvm-metrics/gclog.log -XX:+PrintGCDateStamps -XX:+PrintGC
Details -verbose:gc -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=8 -XX:GCLogFileSize=50M -javaagent:${workdi
r}/${agentjar}=hascontroller=true ${JAVA_OPTS} -jar ${jar}"]
```

3. 使用文件配置

如果您希望使用 TSF [文件配置] 功能，则需要在 Dockerfile 中增加文件配置组件 `tsf-consul-template-docker.tar.gz`（下载地址），然后在 CMD 启动命令中启动该组件。

```
FROM centos:7
RUN echo "ip_resolve=4" >> /etc/yum.conf
RUN yum update -y && yum install -y java-1.8.0-openjdk
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENV workdir /app/

# 下面的 jar 包可替换为您的 Spring Cloud 应用 jar包，注意这个 jar 包要和您的 dockerfile 位于同一级目录
ENV jar provider-demo-0.0.1-SNAPSHOT.jar
COPY ${jar} ${workdir}
WORKDIR ${workdir}

# tsf-consul-template-docker 用于文件配置功能，如不需要可注释掉该行
ADD tsf-consul-template-docker.tar.gz /root/

# JAVA_OPTS 环境变量的值为部署组的 JVM 启动参数，在运行时 bash 替换。使用 exec 以使 Java 程序可以接收 SIGTERM 信号。
CMD ["sh", "-ec", "sh /root/tsf-consul-template-docker/script/start.sh; exec java ${JAVA_OPTS} -jar ${jar}"]
```

私有化版本使用建议：

私有化的 **TSF 1.12 及之前版本**要支持 stdout 日志，需要在启动命令中将 stdout 及 stderr 重定向到一个文件中。将上文的 `CMD` 一行替换成：

```
RUN mkdir -p /data/tsf_std/stdout/logs
CMD ["sh", "-ec", "exec java ${JAVA_OPTS} -jar ${jar} 2>&1 > /data/tsf_std/stdout/logs/sys_log.log"]
```

Mesh 应用构建材料

1. 下载 Mesh 应用 Demo 包：userService.tar.gz。

2. 在该 tar.gz 包同级目录下，编写 Dockerfile 文件：

```
FROM centos:7
RUN mkdir /root/app/
# 其中 userService.tar.gz 是 Mesh 应用压缩包
ADD userService.tar.gz /root/app/
# 设置时区。这对于日志、调用链等功能能否在 TSF 控制台被检索到非常重要。
RUN /bin/cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
RUN echo "Asia/Shanghai" > /etc/timezone
ENTRYPOINT ["bash","/root/app/userService/start.sh"]
```

Mesh 应用压缩包解压后的文件目录结构及文件规范参考 【Mesh Demo 介绍】。

使用文件配置功能

如果容器应用需要使用 TSF 文件配置功能，需要修改 Dockerfile，具体使用指引参考 【文件配置>前提条件】。

构建镜像

1. 在 Dockerfile 所在目录执行 build 命令：

```
docker build . -t ccr.ccs.tencentyun.com/tsf_<主账号 ID>/<应用名>:[tag]
```

其中 <主账号 ID> 对应用户云平台的**主账号 ID**（注意不是当前登录账号 ID，主账号 ID 可以在云平台个人信息页面获取。），<应用名> 表示控制台上的应用名。 tag 为镜像的 tag，用户可自定义。

2. 命令执行完成后，通过 docker image ls 查看创建的镜像。

配置单元化功能

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 支持使用单元化功能以达到让不同的业务流量根据一定的单元化规则分发到指定的单元里，不同单元之间通过微服务网关实现跨单元调用，当某个单元内的服务器实例出现问题时也不会影响到其他单元业务的使用，使得业务受影响粒度达到最小，同时单元化也为业务容灾高可用提供了强有力的保障。

> **注意：**
>
> - 单元化网关支持跨命名空间调用，单元化配置仅支持部署在全局命名空间下的网关。
> - 从1.28.0版本开始，TSF 微服务网关 SDK 支持提供基于 Zuul 的单元化功能。
> - 微服务网关要么是单元化网关，要么是非单元化网关，开启了单元化的功能，该网关就是单元化网关，否则就是非单元化网关。

前提条件

开始实践单元化功能前，请确保已完成了 【SDK 下载】，同时请确保 SDK 版本高于**1.28**。

操作步骤

先暴露 Provider 服务接口，然后将单元化规则配置在调用方 Consumer 工程或微服务网关工程里面。

一、配置工程

::: 微服务网关工程 开始实践微服务网关单元化功能之前，请确保您已完成 【微服务网关 SDK 使用指南】 。

向自定义网关过滤器 Filter 类中添加 tag 标签： java // 下面省略了无关的代码 import com.tencent.tsf.gateway.core.annotation.TsfGatewayFilter; import com.tencent.tsf.unit.TsfUnitContext;

@TsfGatewayFilter public class TestFilter extends TsfGatewayZuulFilter {

```
private Logger logger = LoggerFactory.getLogger(TestFilter.class);

@Override
public String filterType() {
return PRE_TYPE;
}

@Override
public int filterOrder() {
// 注：设置 putTag 的filter的order必须在ZUUL_UNIT_ORDER之前
return TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER - 1;
}

@Override
public boolean shouldFilter() {
return true;
}
```

```
@Override
public Object run() throws ZuulException {
// 单元化场景的putTag操作
RequestContext ctx = RequestContext.getCurrentContext();
HttpServletRequest request = ctx.getRequest();
String userId = request.getHeader("userId");

if(StringUtils.isNotBlank(userId)) {
logger.info("put unit tag userId:{}", userId);
TsfUnitContext.putTag("userId", userId);
}

System.out.println("hello world");
logger.info("hello world");
return null;
}
```

}

- 使用包 com.tencent.tsf.unit 下的类 TsfUnitContext 的方法 putTag 来设置单元化标签。
- 自定义设置 putTag 的 Filter 的顺序必须在 TsfGatewayFilterOrderConstants.ZUUL_UNIT_ORDER 之前。

Consumer工程

1. 向 Consumer 工程中添加依赖

在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsf` 和 `@ComponentScan`

java // 下面省略了无关的代码 @SpringBootApplication @EnableTsf @EnableFeignClients // 使用Feign微服务调用时请启用 @ComponentScan(basePackages="com.tsf.demo.consumer.*") // 需要扫描单元化@TsfUnitRule注解所在的包 public class ConsumerApplication { public static void main(String[] args) throws InterruptedException { SpringApplication.run(ConsumerApplication.class, args); } }

3. 配置单元化规则

- **Feign 调用：**

1. 使用 `@TsfUnitRule` 配置单元化规则。

java @FeignClient(name = "provider-demo") public interface ProviderDemoService {

```
@TsfUnitRule(ruleGenerator = "com.tsf.demo.consumer.util.UserIdGenerator")
@RequestMapping(value = "/echo/unit/{str}", method = RequestMethod.GET)
String echoUnit(@PathVariable("str") String str);
```

}

2. 在代码中编写单元化规则。 需要编写一个类实现接口 TsfUnitRuleGenerator 的方法 generateRule，计算出单元化规则的 tag 标签值，该类就是步骤3中 `@TsfUnitRule` 的 ruleGenerator 属性指定的类。 java // 下面省略了无关的代码 public class UserIdGenerator implements TsfUnitRuleGenerator {

```
@Override
public Map<String, String> generateRule(Method method, Object[] args) {
String userId = (String)args[0];
Map<String, String> ruleMap = new HashMap<>();
// 添加单元化规则的tag key
ruleMap.put("userId", userId);
return ruleMap;
}
```

}

可以看到，TSF 的单元化功能针对Feign调用只需在需要单元化的类或方法上增加一个注解并添加对应的单元化规则即可（需要该 Bean 类被 Spring 所管理）。

> **注意：**
>
> 注解 `@TsfUnitRule` 支持配置在类上，也支持配置在方法上。如果是配置在类上，则单元化规则对该类所有的方法都生效。

- **RestTemplate 调用：** 在 Consumer 中设置 tag ，使用 com.tencent.tsf.unit 包中的 TsfUnitContext 类，设置单元化 Tag 的方法：

java @RequestMapping(value = "/echo-rest-unit/{str}", method = RequestMethod.GET) public String restUnit(@PathVariable String str, @RequestParam(required = false) String tagName, @RequestParam(required = false) String tagValue) { if (!StringUtils.isEmpty(tagName)) { // 添加单元化规则tag key和value TsfUnitContext.putTag(tagName, tagValue); } return restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/unit/" + str, String.class); } 设置单元化规则的 tag 类是 com.tencent.tsf.unit.TsfUnitContext，不是 org.springframework.tsf.core.TsfContext。

二、开启单元化功能并创建单元化规则

在 TSF 控制台开启单元化功能并创建单元化规则，详细操作参考 【单元化部署】 。

端云联调

最近更新时间: 2025-02-18 16:02:00

操作场景

在进行应用开发时，您可以使用 TSF 提供的插件满足本地应用和部署在云端的 TSF 应用测试联调的需求，无需搭建 VPN，帮助您快速提升开发效率。

前提条件

创建流量代理

1. 您需购买一台与云端联调服务同地域的 CVM 资源，将其加入到云端联调服务的 VPC 中，并其绑定一个公网 IP。
2. 如果您需要云端调用本地测试服务，需要在上述已购买的 CVM 资源的 SSH 服务进行如下配置：
3. 编辑 sshd_config 配置

```
sudo vim /etc/ssh/sshd_config
```

4. 添加如下配置信息允许远程主机连接本地的转发端口，并保存：

```
GatewayPorts clientspecified
```

5. 重启 SSH 服务 Debian/Ubuntu linux：

```
sudo systemctl restart ssh
```

CentOS / RHEL / Fedora / Redhat Linux

```
sudo systemctl restart sshd
```

使用限制

- 作为跳板机的 CVM 必须和联调服务集群处于同一个私有网络 VPC
- 暂时只支持 TSF Spring Cloud 应用、TSF Service Mesh 应用
- 不支持分布式日志、监控、分布式链路追踪的云端联调能力
- TSF Servie Mesh 应用的云端服务在调用本地服务的时候不能显示指定被调服务的端口号（或者将端口号设置为80）

操作步骤

步骤1：创建应用

如果本地测试服务未在云端部署则需为其创建应用并创建部署组获取联调配置，如果本地测试服务已在云端部署仅为本地开发测试实例则仅需在已有的云端应用下创建部署组获取联调配置，您可以使用 TSF 官网提供的 Demo（点击下载）进行验证。

步骤2：创建部署组获取联调配置

1. 创建部署组 创建部署组关联云端测试服务的集群及命名空间，确保其能相互访问，无需关联实例及部署应用点击保存。
2. 获取联调配置 单击部署组 ID，查看基本信息获取联调配置，具体参数如下所示：

参数	描述
tsf_token	TSF 服务注册 token 认证信息
tsf_namespace_id	TSF 命名空间 ID
tsf_applicaton_id	TSF 应用 ID
tsf_appid_id	用户账号 ID 信息
tsf_group_id	TSF 部署组 ID

步骤3：本地环境配置

1. 下载本地联调 agent 程序包（点击下载）。

2. 打开本地 IDE 环境，配置 JVM 启动参数。

JVM 启动参数配置如下：

```
-javaagent:"agent绝对路径=ssh_host=cvm公网ip&&user=用户名&&pass=密码&&local_port=本地服务启动端口&&remote_ip=cvm内网ip"
-Dtsf_token=TSF服务注册token认证信息
-Dtsf_namespace_id=TSF命名空间ID
-Dtsf_application_id=TSF应用ID
-Dtsf_app_id=账户ID信息
-Dtsf_group_id=TSF部署组ID
```

示例：

```
-javaagent:"/Users/root/agent.jar=ssh_host=139.136.79.195&&user=root&&pass=123456tt&&local_port=8080&&remote_ip=172.30.0.93"
-Dtsf_token=rakxSMEnGjXtAvSe9I2BlvqOootl_WAw9YzIiaJ-RD4=
-Dtsf_namespace_id=namespace-py5lr6v4
-Dtsf_application_id=application-nygxjma2
-Dtsf_app_id=1300555551
-Dtsf_group_id=group-jy9zr8ag
```

配置详情描述如下：

参数	描述	必填

参数	描述	必填
javaagent	JavaAgent 动态代理 jar 包路径	必填
ssh_host	SSH 服务器地址	必填
ssh_port	SSH 服务监听端口	必填
user	SSH 服务登录用户名	必填
pass	SSH 服务登录密码	选填，pass 和 key 必须要有一个不为空
key	SSH 服务登录私钥路径	选填，pass 和 key 必须要有一个不为空
local_port	本地启动服务端口	选填，如果本地调试服务只是正向调用云端服务则可以不填写，如果需要被云端服务访问则需要填写
remote_ip	CVM 内网 IP	选填，如果本地调试服务只是正向调用云端服务则可以不填写，如果需要被云端服务访问则需要填写
tsf_token	TSF 服务注册 token 认证信息	必填
tsf_namespace_id	TSF 命名空间 ID	必填
tsf_applicaton_id	TSF 应用 ID	必填
tsf_appid_id	账号 ID 信息	必填
tsf_group_id	TSF 部署组 ID	必填
tsf_instance_id	注册实例 ID	选填 ，如果有多个同类型的本地服务需要注册，为了避免实例 ID 相同可以特殊指定

3. 启动本地服务。

步骤4：联调测试验证

本地服务启动后会在 TSF 服务治理对应的服务详情中显示服务实例列表信息，包含实例信息、服务端口、部署组信息。

在本地通过联调测试远程服务，显示 response 信息则验证成功。

Dubbo 应用接入
Dubbo Demo 工程概述

最近更新时间: 2025-02-18 16:02:00

获取 Demo

基于 Alibaba Dubbo 版本 SDK 的 Demo 下载 本工程只是示例，现存Dubbo 应用可以直接看【Dubbo 应用接入TSF】文档。

> **注意：**
>
> 如果从零开始新写的微服务系统推荐采用 Spring Cloud

工程目录

`tsf-dubbo-xxx-demo` 的工程目录如下：

```
|- consumer-demo
|- provider-demo
|- pom.xml
```

其中 consumer-demo 表示服务消费者， provider-demo 表示服务提供者，pom.xml 中定义了工程需要的依赖包：

```
<project xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0" xmlns:xsi="http://imgcach
e.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0 http://imgcache.finance.
cloud.tencent.com:80maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.1.3.RELEASE</version>
</parent>

<artifactId>provider-demo</artifactId>
<version>1.1.7-alibaba-RELEASE</version>
<packaging>jar</packaging>
<name>provider-demo</name>

<properties>
<start-class>com.tsf.demo.provider.ProviderApplication</start-class>
</properties>

<dependencies>
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<version>1.18.6</version>
<scope>provided</scope>
</dependency>
```

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-dubbo-registry</artifactId>
<!-- 调整为 SDK 最新版本号 -->
<version>1.1.7-alibaba-RELEASE</version>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>

</project>
```

关于 Maven 环境安装，请参考 Spring Cloud 【SDK 下载】 时的 Maven 配置。

Dubbo 应用接入Mesh

最近更新时间: 2025-02-18 16:02:00

操作场景

Dubbo 作为一款传统基于 SDK 的 Java RPC 框架，通过引入 jar 包的方式实现服务间远程方法调用、服务注册发现及服务治理。Dubbo 开发的应用在不修改任何业务代码的前提下，通过简单的 Mesh 封包部署到 TSF 平台，可透明实现服务注册发现和无侵入的服务治理能力。

本文档以 Dubbo Mesh Demo 为例介绍 Dubbo 应用在 TSF 平台接入 Mesh 的操作方案及相关注意事项。

前提条件

已下载 Dubbo Mesh Demo

操作步骤

1. Maven 环境安装

详细操作请参考 【Maven 安装】。

2. 配置 spec.yaml 服务注册文件

Dubbo Mesh Demo 中提供了三个 Dubbo 应用，以 greet 应用为例，其 spec.yaml 配置为：

```
apiVersion: v1
kind: Application
spec:
services:
- name: org.apache.dubbo.samples.api.GreetingService # 需要注册的 Dubbo 服务名，必须跟提供的 interface 名保持一致
ports:
- targetPort: 20881 # 该 Dubbo 服务监听的端口
protocol: dubbo # 协议指定为 Dubbo
healthCheck:
path:
```

> **注意：**
>
> 部署到 TSF 平台后，Mesh 的 sidecar 解析 Dubbo 应用的 spec.yaml，根据服务名、监听端口、协议等信息自动注册服务到注册中心。

屏蔽 greet 应用原注册中心（可选）：

```
<dubbo:registry address="N/A" subscribe="false" register="false"/>
```

> **注意：**
>
> 此处也可保留原注册中心，实现 Dubbo 服务双注册。

3. 编译打包

Dubbo Mesh Demo 中每个应用都提供了 build.sh 脚本用于构建部署在 TSF 上的 Mesh 程序包，执行 ./build.sh 将在当前目录下生成对应的 tar.gz 包，该包可直接部署在 TSF 上，以 greet 应用为例，生成的程序包为 dubbo-greet.tar.gz ，文件目录包括：

- dubbo-samples-greetservice-1.0.jar: Maven 构建的 FATJAR 包
- start.sh
- stop.sh
- cmdline
- spec.yaml

build.sh 脚本及程序包中文件详细内容请 下载 Dubbo Mesh Demo 查看。

Dubbo 兼容说明

- TSF Mesh 为 Dubbo 应用提供服务间通信、服务注册发现、负载均衡、服务路由等治理能力。
- Dubbo 应用可以保留原有的注册方式。
- Dubbo 应用可以继续使用原有的 filter 机制。

Dubbo 应用接入TSF

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 为用户现存的 Dubbo 应用提供了两种插件，分别适合不同场景下的使用：

- 当前只是轻度使用 Dubbo，希望使用 TSF 的治理能力，可以使用**完整版插件**。完整版插件提供完整的治理能力（可能会和用户已有的治理能力冲突）。
- 当前已经使用了 Dubbo 的治理能力，只是希望注册到 TSF，可以使用**轻量级插件**。轻量级插件只提供注册和发现的能力。

> **注意：**
>
> - 两款插件目前均只支持 Alibaba Dubbo，版本为2.6.8（其余低版本只要和2.6.8兼容即可）。
> - 两款插件不会实时同步更新全部的 TSF 能力，具体能力支持以文档为准。

> -

完整版插件接入

完整版插件通过 Dubbo filter 的机制，将 TSF 的全部能力适配至 Dubbo 上，允许用户只修改几行依赖和配置即可体验完整的治理和监控体验。

> **注意：**
>
> TSF 对治理能力的定义和 Dubbo 原生的略有区别，包括但不限于路由，负载均衡等，如果用户已经使用并依赖了 Dubbo 自身的治理能力，并且不希望行为变化，可以采用轻量级的框架接入 TSF。

操作步骤

1. Maven 环境安装

详细操作请参考 【Maven 安装】。

2. 注册中心配置

Dubbo 官网 Demo：

```
<dubbo:registry address="multicast://224.5.6.7:1234"/>
```

TSF Demo（**注册中心地址使用注册中心 IP 和端口替换**）：

```
<dubbo:registry address="tsfconsul://127.0.0.1:8500">
<dubbo:parameter key="router" value="tsfrouter"/>
</dubbo:registry>
```

- 协议名为 tsfconsul。
- "注册中心地址:端口" 可以填写 127.0.0.1:8500。在 TSF 控制台部署时，SDK 会替换为正确的地址。

3. 添加依赖

根据业务使用的对应的 TSF Dubbo 版本 SDK 如下：

3.1 在 pom 中增加插件依赖：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>atom-extension-dubbo</artifactId>
<!-- 修改为对应的版本号 -->
<version>1.0.0-RELEASE</version>
</dependency>
```

3.2 在 pom 中剔除其他 brave 依赖，否则可能会造成类加载冲突。

> **注意：**
>
> 由于完整版插件支持了全套的 TSF 能力、包括监控，依赖了一些开源包，这里如果在之前使用 Dubbo 时也依赖了一些开源包的话，建议删除之前的依赖。

4. 打包 FatJar

和 Spring Boot 结合的时候，您可以通过 **spring-boot-maven-plugin** 构建一个包含所有依赖的 jar 包（FatJar），执行命令 `mvn clean package` 。

如果是单纯的 Dubbo 应用，可以使用 Maven 的 FatJar 插件。

```
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-jar-plugin</artifactId>
<configuration>
<archive>
<manifestEntries>
<Implementation-Version>${project.version}</Implementation-Version>
</manifestEntries>
</archive>
</configuration>
</plugin>
<plugin>
<artifactId>maven-assembly-plugin</artifactId>
<configuration>
<archive>
<manifest>
<!--这里指定要运行的main类-->
<mainClass>com.tencent.tsf.atom.example.alibaba.dubbo.client.ConsumerApplication</mainClass>
</manifest>
</archive>
<descriptorRefs>
<descriptorRef>jar-with-dependencies</descriptorRef>
</descriptorRefs>
</configuration>
<executions>
<execution>
<id>make-assembly</id>
<phase>package</phase>
<goals>
```

```
<goal>single</goal>
</goals>
</execution>
</executions>
</plugin>
</plugins>
```

5. 使用自定义 Tag

完整版插件支持 TSF 平台的自定义 Tag 能力。

```
// 引入 TsfContext
import com.tencent.tsf.atom.adaptor.tsf.common.TsfContext;

/**
* 放入将自定义的 Tag 放入 Context 中
* Context 会将该 Tag 往下传递
* 用户可以对某个 Tag 进行查询和删除操作
*/
TsfContext.putTag("local", "yes");
```

> **注意：**
>
> Tag 会自动往下传递，如果不希望某个 Tag 继续传递可以在代码中手动删除。

6. 使用自定义 Filter

Dubbo 框架中如果要实现自定义 Filter 需要 resources 目录下建立特定名称的文件夹。 官方推荐目录为 META-INF/dubbo，用户如果在此文件夹下声明自定义 Filter 则不需要任何改动。

TSF 将自定义插件目录设置为 META-INF/services，如果用户希望在 services 目录下实现自定义 filter， 则需要在 filter 的文件中显示加入 TSF 的三个插件。

> **注意：**
>
> Maven 打包时会把相同目录的文件保留一份，所以推荐用户不要使用 services，而是采用 Dubbo 目录。

```
// 自定义 filter
logFilter=com.alibaba.dubbo.demo.consumer.LogFilter

// Tsf 系统 filter，需要显示加入
localAddressFilter=com.tencent.tsf.atom.extensions.dubbo.filter.LocalAddressFilter
atomConsumerFilter=com.tencent.tsf.atom.extensions.dubbo.filter.AtomConsumerFilter
atomProviderFilter=com.tencent.tsf.atom.extensions.dubbo.filter.AtomProviderFilter
```

7. API 上报

TSF 支持上报 Alibaba Dubbo 程序的 API，在服务治理-接口列表中展示。该接口能够导入 TSF 微服务网关，并通过协议转换的功能转为 HTTP 接口对外暴露。请确保完整版插件依赖（atom-extension-dubbo）版本高于 **1.1.0**。

使用方法如下：

为需要上报的 Dubbo 方法增加 @TsfDubboApiMethod 注解，该注解支持支持两个参数，path 表示协议转换后用于调用 HTTP API，desc 表示该 API 的描述信息。

> **注意：**
>
> - 一个服务（Dubbo Interface）下，path 需要唯一，用于 HTTP 访问时的准确请求。
> - @TsfDubboApiMethod 注解需要作用于基类上，而不是实现类。

上报成功后，可以在服务治理-接口列表中展示： 单击接口路径，即可查看详细信息。

兼容性说明

- 提供对 TSF 平台的兼容支持。
- 暂不支持全局命名空间。
- 部分系统tag与部分监控能力暂时不支持，具体已实际展示为准。
- 暂不支持对非 Restful 协议的 API 进行调试。
- Dubbo 应用的其他能力（如 filter 机制），可以继续使用。

常见错误

- 问题1：启动时报 ClassNotFound 异常。 解决方案：一般是类加载冲突，可以将 pom 中 Dubbo 相关的依赖删除。
- 问题2：路由治理不生效。 解决方案：查看配置中心中是否指定了 tsfrouter 作为路由器。

轻量级插件接入

轻量级框架为 Dubbo 应用提供服务注册中心，Dubbo 应用可通过依赖 jar 包的方式接入该项服务。下文将介绍 Dubbo 应用从接入TSF到部署应用的操作方法及相关注意事项。

操作步骤

1. Maven 环境安装

详细操作请参考 【Maven 安装】。

2. 注册中心配置

Dubbo 官网 Demo：

```
<dubbo:registry address="multicast://224.5.6.7:1234"/>
```

TSF Demo（**注册中心地址使用注册中心 IP 和端口替换**）：

```
<dubbo:registry address="tsfconsul://注册中心地址:端口"/>
```

- 协议名为 tsfconsul。
- "注册中心地址:端口" 可以填写 127.0.0.1:8500。在 TSF 控制台部署时，SDK 会替换为正确的地址。

3. 添加依赖

根据业务使用的对应的 TSF Dubbo 版本 SDK 如下：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-dubbo-registry</artifactId>
<!-- 修改为对应的版本号 -->
<version>1.1.7-alibaba-RELEASE</version>
</dependency>
```

> **注意：**
>
> 目前只支持 Alibaba Dubbo。

4. 打包 FATJAR

和 Spring Boot 结合的时候，您可以通过 **spring-boot-maven-plugin** 构建一个包含所有依赖的 jar 包（FatJar），执行命令 `mvn clean package`。

兼容性说明

- TSF 提供服务注册中心，Dubbo 应用通过依赖 jar 包的方式使用。
- TSF 支持 Dubbo 应用的 Dubbo 服务注册、Dubbo 服务调用。
- Dubbo 应用的其他能力（如 filter 机制），可以继续使用。

常见错误

部分包对版本有要求，如果发生**包冲突**，请尝试主动依赖以下版本：

```
<dependency>
<groupId>com.ecwid.consul</groupId>
<artifactId>consul-api</artifactId>
<version>1.4.2</version>
</dependency>
```

SDK 文档
Edgware

最近更新时间: 2025-02-18 16:02:00

基于 Spring Cloud Edgware 版本 SDK，支持 Spring Boot 1.5.x。

> **说明：**
>
> 2020年5月19日起，TSF 主要支持 Greenwich 和 Finchley 版本的功能更新，Edgware 版本主要进行缺陷修复，建议您优先使用 Finchley和 Greenwich 版本（社区 Edgware 版本 于2019年8月停止更新）。

1.21.5-Edgware-RELEASE（2021-02-07）

Bug 修复

- 处理 Spring 组件开源漏洞风险，升级 Spring Framework 到4.3.29版本。
- spring-cloud-tsf-core 修复与 spring-boot-devtools 的冲突。
- 修复多个限流规则时，全局限流无法关闭的问题。
- 修复路由关闭问题。
- 修复网关多个命名空间时 consul index 混用导致第一次跨命名空间调用加载慢的问题。
- 修复分布式配置下发 spring.application.name 时，无法上报 swagger 问题。
- 修复服务治理时 API PATH 标签匹配 PATH 参数失败问题。
- 修复泳道规则内存可见性 Bug。

优化

- 心跳请求增加重试。
- 解析 input stream 失败只是打 warn 日志，不抛异常

1.21.4-Edgware-RELEASE（2020-08-20）

Bug 修复

- 修复 MySQL 调用链数据对多数据源支持。
- spring-cloud-tsf-msgw： 修复 application/x-www-form-urlencoded 类型请求，当绑定插件通过 zuul 网关代理访问时出错的问题。

1.21.3-Edgware-RELEASE（2020-07-16）

Bug 修复

修复网关 MSGW SDK 和服务发现 SDK 不兼容，造成拉取服务列表过快的问题。

优化

spring-cloud-tsf-gateway 网关兼容新插件类型。

1.21.2-Edgware-RELEASE（2020-07-06）

Bug 修复

处理 tomcat 组件开源漏洞风险。

- 升级 org.apache.tomcat.embed.tomcat-embed-core 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.56版本。

优化

- 调整泳道标签的传递属性。
- 调整泳道入口行为。

1.22.1-Edgware-RELEASE（2020-08-19）

优化

- 优化 TSF MSGW 未知插件类型提示。
- 优化 TSF 第三方组件依赖。

Bug 修复

- 处理 tomcat 组件开源漏洞风险。
- 升级 org.apache.tomcat.embed.tomcat-embed-core 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.56版本。
- spring-cloud-tsf-msgw:
- 修复 application/x-www-form-urlencoded 类型请求，当绑定插件通过 zuul 网关代理访问时出错的问题。
- 修复无法使用 Feign 发起微服务调用的问题。

版本建议

支持向后兼容，建议全量升级。

1.22.0-Edgware-RELEASE（2020-04-29）

优化

- 优化默认日志配置支持容器部署场景。
- 优化 TSF MSGW zuul 依赖。

版本建议

支持向后兼容，建议全量升级。

1.21.1-Edgware-RELEASE（2020-04-29）

Bug 修复

修复泳道 ID 在非泳道起始应用中传递丢失的问题。

优化

- 优化调用链生成文件名称的生成规则。

版本建议

- 支持向后兼容，建议全量升级。

1.21.0-Edgware-RELEASE（2020-04-17）

新特性

- 全链路灰度发布。
- 增加熔断状态变更事件上报。

Bug 修复

- spring-cloud-tsf-swagger 修复 @ApiParam 注解 Example 属性解析异常问题。
- 修复 Tag 在 AsyncRestTemplate 下不传递的问题。
- 修复 Feign 无法使用绝对 URL 请求的问题。
- spring-cloud-tsf-gateway：
- 修复 Tag 标签插件未在调用中透传的问题。
- 修复当绑定网关插件后造成 Query 参数未透传的问题。

优化

支持 swagger 自动扫描包多路径特性。

1.20.0-Edgware-RELEASE（2020-03-02）

Bug 修复

- spring-cloud-tsf-gateway 修复 tag plugin 中 header 类型取值大小写敏感的问题。
- 处理 tomcat 组件开源漏洞风险。
- 升级 org.apache.tomcat.embed.tomcat-embed-core 到8.5.51版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.51版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.51版本。
- 升级 org.apache.tomcat.tomcat-annotations-api 到8.5.51版本。
- 修复 application/x-www-form-urlencoded 类型请求，通过 zuul 网关代理访问 provider 时，服务端不到请求参数的问题。

优化

spring-cloud-tsf-gateway 新增 tag plugin 中 path 类型取值。

版本建议

支持向后兼容，建议全量升级。

1.19.0-Edgware-RELEASE（2020-01-16）

新特性

新增 服务熔断 功能。

版本建议

支持向后兼容，建议全量升级。

1.18.1-Edgware-RELEASE（2020-01-14）

Bug 修复

- spring-cloud-tsf-route 修复路由权重不准的问题。
- spring-cloud-tsf-consul-discovery 修复服务发现线程池上限的问题。
- spring-cloud-tsf-sleuth 修复 druid 连接池事务兼容问题。
- spring-cloud-tsf-sleuth 修复同时依赖多个数据库连接池问题。
- spring-cloud-tsf-core 修复 Custom Metadata 设置接口不兼容。

优化

支持通过 `tsf.discovery.watch.enabled` 关闭服务发现时的 watch 监听。

版本建议

支持向后兼容，建议全量升级。

1.18.0-Edgware-RELEASE（2019-12-25）

Bug 修复

- spring-cloud-tsf-sleuth 修复 JDBC 代理过程 NPE bug 问题。
- spring-cloud-tsf-route 修复路由系统标签匹配的问题。

新特性

- 服务治理支持全局命名空间。
- 新增 spring-cloud-tsf-gateway 微服务网关（zuul1 版）SDK，基于此 SDK 二次研发，无缝集成 TSF 平台服务治理能力。
- 新增自定义日志配置需要的 Converter 和 Layout 类，支持用户使用自定义 logback\log4j\log4j2 日志配置。

优化

- spring-cloud-tsf-sleuth 优化 TraceStatementProxyHandler JDBC 代理过程 SDK 内部异常处理逻辑：非代理异常、非 SDK 产生的异常，直接抛出；代理异常或 SDK 产生的异常，直接调用服务不经过调用链逻辑。

版本建议

支持向后兼容，建议全量升级。

1.16.3-Edgware-RELEASE (2020-04-29)

优化

优化调用链生成文件名称的生成规则。

1.16.2-Edgware-RELEASE (2020-03-02)

Bug 修复

spring-cloud-tsf-sleuth bug fixed：

- 修复 application/x-www-form-urlencoded 类型请求，通过 zuul 网关代理访问 provider 时，服务端不到请求参数的问题。
- 处理 Custom Metadata 设置接口不兼容。
- 调用链输出用户自定义 Tag 和 Metadata。
- 修复 druid 连接池事务兼容问题。
- 修复同时依赖多个数据库连接池问题。

1.16.1-Edgware-RELEASE（2019-12-3）

Bug 修复

API 注册兼容从环境变量和启动参数中读取 TSF 参数信息。

1.16.0-Edgware-RELEASE（2019-10-11）

新特性

- Kafka 的链路追踪能力。
- 增加 swagger-ui 依赖包。

优化

- 集成 spring-cloud-tsf-swagger 包后，本地启动无需设置 tsf.swagger.enabled=false。
- 集成 spring-cloud-tsf-swagger 包后，支持本地使用 swagger-ui 进行调试。

Bug 修复

- 修复在自定义 RedisTemplate 中指定序列化方式时的错误。
- 修复对 StringRedisTemplate 的支持。
- 修复引入 swagger 包后，低版本 guava 包引起冲突。
- 配置回调功能空指针异常。

版本建议

支持向后兼容，建议全量升级。

1.14.2-Edgware-RELEASE（2019-09-10）

Bug 修复

- 限流 Bug fix。
- TsfContext.putTag 覆盖 bug fix。

版本建议

- 支持向后兼容，建议全量升级。

1.14.1-Edgware-RELEASE（2019-07-24）

Bug 修复

- 修复 tsf sdk 依赖的 scheduler 和业务自身的 scheduler 相互影响的问题。
- 修复 spring-cloud-tsf-ratelimit 包限流不准确问题。
- 修复 spring-cloud-tsf-sleuth 包数据源和 Mybatis 兼容性问题。

版本建议

支持向后兼容，建议全量升级。

1.14.0-Edgware-RELEASE（2019-06-21）

新特性

支持 MySQL JDBC、Redis、MongoDB、CMQ 组件调用链。

版本建议

支持向后兼容，建议全量升级。

1.12.5-Edgware-RELEASE（2020-07-17）

Bug 修复

修复 spring-cloud-tsf-route 包路由不准确问题。

优化

调整心跳请求的超时时间，当出现丢包时能够快速重试。

版本建议

支持向后兼容，建议全量升级。

1.12.4-Edgware-RELEASE（2019-08-15）

Bug 修复

- 修复 tsf sdk 依赖的 scheduler 和业务自身的 scheduler 相互影响的问题。
- 修复 spring-cloud-tsf-ratelimit 包限流不准确问题。

版本建议

支持向后兼容，建议全量升级。

1.12.2-Edgware-RELEASE（2019-04-22）

Bug 修复

修复 Edgware 版本自定义 tag 问题。

版本建议

支持向后兼容，建议全量升级。

1.12.1-Edgware-RELEASE（2019-03-25）

Bug 修复

修复配置回调功能未生效问题。

版本建议

支持向后兼容，建议全量升级。

1.12.0-Edgware-RELEASE（2019-03-13）

新特性

- 支持自动重注册，服务鉴权/路由/限流策略本地缓存。
- 服务路由支持基于可用区和地域就近访问策略。

优化

- 升级分布式配置监听，精确并减小监听范围，处理更新为空的场景，避免大范围 key 刷新事件。
- 优化分部署配置回调触发逻辑。

Bug 修复

- spring-cloud-commons 升级到1.3.1解决 RetryTemplate 会导致 LoadBalanceInterceptor thread unsafe 问题。
- 修复启用 hystrix 时配置会导致 tsf-route 与 feignbuilder 冲突的问题。

版本建议

支持向后兼容，建议全量升级。

1.10.0-RELEASE（2018-11-12）

新特性

- 服务路由（srping-cloud-tsf-route）：支持服务下单个 API 请求级别的路由。

- 服务限流（spring-cloud-tsf-ratelimite）：支持服务下单个 API 请求级别的限流。
- 日志输出（spring-cloud-tsf-logger）：支持默认日志输出。
- API 注册（spring-cloud-tsf-swagger）：支持服务下 API 信息自动注册，查看 API 出入参请求结构。

Bug 修复

- 解决 RestTemplate Bean 冲突问题。

升级建议

- 支持向后兼容。
- 新功能建议全量升级。

1.1.1-RELEASE（2018-08-26）

新特性

- 服务限流（spring-cloud-tsf-rate-limit）：支持针对所有请求、单个服务的请求进行流量控制。
- 服务路由（spring-cloud-tsf-route）：支持基于部署组、系统标签、自定义标签的路由设置。
- 服务鉴权（spring-cloud-tsf-auth）：支持基于服务名和标签的鉴权设置。
- 配置加密（spring-cloud-tsf-encrypt）：支持配置加密功能。
- 应用状况监控（spring-cloud-tsf-metrics）：支持查看 Spring Boot 应用的 JVM 内存分布、线程、HTTP Traces、环境变量。
- 调用链（spring-cloud-tsf-sleuth）：支持在调用链上设置标签和自定义 Metada。

Bug 修复

调用链 SDK 问题修复。

升级建议

全部建议升级。

Finchley

最近更新时间: 2025-02-18 16:02:00

基于 Spring Cloud Finchley 版本 SDK，支持 spring boot 2.0.x。

1.32.0-Finchley-RELEASE（2020-06-21）

新特性

- 支持微服务网关可扩展性。支持使用 TSF 网关 SDK 的同时，自定义网关路由策略、支持 websocket、支持跨域等原生网关能力。
- Oauth 插件支持第三方鉴权地址为微服务 API 的能力。
- 支持原生网关使用熔断治理的能力。
- 支持服务监听触发回调。

优化

- consul 异常时，避免一直刷日志。
- 增加 tsf launcher。

Bug 修复

- 修复 Feign 在指定 URL 的模式下无法请求的问题。
- 修改scg metrics duration异常问题

版本建议

支持向后兼容，建议全量升级。

1.29.0-Finchley-RELEASE（2020-05-07）

新特性

- 微服务网关增加单元化功能。
- 微服务网关增加 Dubbo 协议转换功能。
- spring-cloud-tsf-sleuth: 新增 cmq-tcp-client 和 cmq-http-client 调用支持。

优化

- 优化和开源 spring cloud consul 依赖的冲突。
- 支持通过配置 -Dspring.cloud.consul.enabled=false 关闭连接 consul，适配单元测试场景时的启动。
- actuator 依赖改为 optional。
- spring-cloud-tsf-sleuth：优化 getProperties 性能。
- spring-cloud-tsf-ratelimit：优化限流的 httpclient。

Bug 修复

- spring-cloud-tsf-logger：修复自定义日志格式没有服务名的问题。
- spring-cloud-tsf-sleuth：修复调用链获取 IP 偶现获取不到问题。

- spring-cloud-tsf-swagger：修复 IgnoreGatewayApi 注解导致的潜在空指针异常。
- spring-cloud-tsf-consul-discovery：修复被调方实例不存在时不断打印异常日志的问题。

版本建议

支持向后兼容，建议全量升级。

1.26.1-Finchley-RELEASE（2020-12-31）

优化

spring-cloud-tsf-sleuth 新增 CMQ 调用支持。

版本建议

支持向后兼容，建议全量升级。

1.26.0-Finchley-RELEASE（2020-12-07）

新特性

- spring-cloud-tsf-msgw-scg：
- 补齐 Spring Cloud Gateway 网关的服务治理能力，支持用户按照需求灵活选择 Zuul 或 Spring Cloud Gateway。
- 支持托管外部 API。
- spring-cloud-tsf-msgw-zuul：支持托管外部 API。
- spring-cloud-tsf-swagger：支持添加注解 @IgnoreGatewayApi 来忽略某个网关 API 不被发现（忽略该网关的 API，但服务治理 API 不受影响）。

版本建议

支持向后兼容，建议全量升级。

1.25.0-Finchley-RELEASE（2020-12-04）

新特性

spring-cloud-tsf-msgw-zuul 支持服务熔断能力。

Bug 修复

- spring-cloud-tsf-ratelimit：修复多个限流规则时，全局限流无法关闭的问题。
- spring-cloud-tsf-route：修复当只有一个路由规则时，路由规则关闭不生效的问题。
- spring-cloud-tsf-lane：修复泳道规则内存可见性 Bug。

版本建议

支持向后兼容，建议全量升级。

1.24.0-Finchley-RELEASE（2020-09-25）

新特性

- 支持云上 Spring Cloud 应用平滑迁移 TSF。
- 支持 PostgreSQL 组件调用链。

Bug 修复

- spring-cloud-tsf-consul-config：
- 修复本地加密配置不能正确解密的问题。
- 修复 MySQL 调用链对多数据源支持。
- spring-cloud-tsf-core： 增加线程上下文接口，在父亲线程中塞入线程局部变量后，子线程不论是线程池反复使用还是一次性使用都能正确继承父线程局部变量。

版本建议

支持向后兼容，建议全量升级。

1.23.9-Finchley-RELEASE（2021-06-23）

Bug 修复

修改 scg metrics duration 异常问题。

优化

- 服务发现增加零实例保护。
- consul 异常时，避免一直刷日志。

版本建议

支持向后兼容，建议全量升级。

1.23.8-Finchley-RELEASE（2021-04-13）

优化

网关支持适配特殊 url。例如：用户请求 url 是 `/echo`、`/echo/`、`/echo\` 时，网关统一会当 `/echo` 处理。

版本建议

支持向后兼容，建议全量升级。

1.23.7-Finchley-RELEASE（2021-02-02）

Bug 修复

- 修复服务治理时 API PATH 标签匹配 PATH 参数失败问题。
- 修复本地启动时监听原生 consul 路径的问题。

优化

统一第三方组件的版本号。

版本建议

支持向后兼容，建议全量升级。

1.23.6-Finchley-RELEASE（2020-12-21）

Bug 修复

- 处理 Spring 组件开源漏洞风险，升级 Spring Framework 到5.0.19版本。
- spring-cloud-tsf-core 修复与 spring-boot-devtools 的冲突。
- spring-cloud-tsf-ratelimit：修复当只有一个限流规则时，限流规则关闭不生效的问题。
- spring-cloud-tsf-route：修复当只有一个路由规则时，路由规则关闭不生效的问题。
- spring-cloud-tsf-swagger 修复通过分布式配置下发 spring.application.name 时，API 上报失败的问题。
- 修复网关多个命名空间时 consul index 混用问题。

优化

- spring-cloud-tsf-consul-discovery 心跳请求增加重试。
- spring-cloud-tsf-consul-config 支持本地加密配置解析。
- spring-cloud-tsf-lane：优化泳道规则生效逻辑。

版本建议

支持向后兼容，建议全量升级。

1.23.5-Finchley-RELEASE（2020-11-11）

优化

- spring-cloud-tsf-msgw-zuul 支持服务熔断能力。
- spring-cloud-tsf-sleuth 修改调用 SQL 存储的最长长度到64000字符。
- 调整泳道入口行为。

版本建议

支持向后兼容，建议全量升级。

1.23.4-Finchley-RELEASE（2020-09-16）

Bug 修复

- 修复 MySQL 调用链中对多数据源支持。
- 修复 feign 请求调用链中只展示 HTTP 方法。
- 修复定时任务的线程数问题。
- 修复网关使用就近命名空间的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.3-Finchley-RELEASE（2020-09-14）

Bug 修复

- spring-cloud-tsf-msgw： 修复网关 MSGW SDK 和服务发现 SDK 不兼容，造成拉取服务列表过快的问题，从而导致注册中心负载压力过大的问题。
- spring-cloud-tsf-consul-discovery： 修复服务发现线程数不准确（少于需要请求的服务数），导致服务发现线程调度不及时，节点状态更新可能会延迟30s的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.2-Finchley-RELEASE（2020-08-19）

Bug 修复

spring-cloud-tsf-msgw： 修复 application/x-www-form-urlencoded 类型请求，当绑定插件通过 zuul 网关代理访问时出错的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.1-Finchley-RELEASE（2020-08-12）

Bug 修复

spring-cloud-tsf-msgw： 修复 scg 版本网关不支持 HTTP 请求中文编码的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.0-Finchley-RELEASE（2020-07-06）

新特性

- spring-cloud-tsf-msgw：
- 新增网关路径重写配置功能。
- 新增网关微信小程序登录插件功能。
- spring-cloud-tsf-sleuth：
- 新增调用链支持 RocketMQ。
- 修复 Kafka 中的类型转发错误。
- spring-cloud-tsf-core：
- 监控数据结构中增加 HTTP 请求方法、以及请求模版路径。
- 调用链数据结构中增加 HTTP 请求方法。

Bug 修复

- spring-cloud-tsf-msgw：
- 修复数据同步时，可能会短暂获取到错误数据的问题。
- 修复 SCG Tag 中数据未正确清除的问题。
- 处理 tomcat 组件开源漏洞风险：
- 升级 org.apache.tomcat.embed.tomcat-embed-core 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.56版本。

1.22.1-Finchley-RELEASE（2020-05-06）

优化

优化 TSF MSGW scg 使用，用户无需显示配置全局路由。

版本建议

支持向后兼容，建议全量升级。

1.22.0-Finchley-RELEASE（2020-04-29）

新特性

- 支持 SpringCloud Gateway 链路追踪和调用监控。
- TSF MSGW scg 版本发布。

Bug 修复

- 修复在使用 redis，自定义多个 LettuceConnectionFactory 时，不能链路追踪所有请求的问题。
- 修复调用监控禁用场景内存泄露问题。

优化

- 优化默认日志配置支持容器部署场景。
- 优化 TSF MSGW zuul 依赖。

1.21.9-Finchley-RELEASE（2021-02-02）

Bug 修复

- 修复服务治理时 API PATH 标签匹配 PATH 参数失败问题。
- 修复本地启动时监听原生 consul 路径的问题。

优化

统一第三方组件的版本号。

版本建议

支持向后兼容，建议全量升级。

1.21.8-Finchley-RELEASE（2020-12-31）

Bug 修复

spring-cloud-tsf-sleuth：修复特殊场景调用链 IP 获取失败问题。

版本建议

支持向后兼容，建议全量升级。

1.21.7-Finchley-RELEASE（2020-12-21）

Bug 修复

- spring-cloud-tsf-ratelimit：修复当只有一个限流规则时，限流规则关闭不生效的问题。
- spring-cloud-tsf-route：修复当只有一个路由规则时，路由规则关闭不生效的问题。
- spring-cloud-tsf-swagger 修复通过分布式配置下发 spring.application.name 时，API 上报失败的问题。
- 修复网关多个命名空间时 consul index 混用问题。

优化

- spring-cloud-tsf-sleuth 新增 CMQ 调用支持。
- spring-cloud-tsf-consul-discovery 心跳请求增加重试。
- spring-cloud-tsf-consul-config 支持本地加密配置解析。
- spring-cloud-tsf-lane：优化泳道规则生效逻辑。

版本建议

支持向后兼容，建议全量升级。

1.21.6-Finchley-RELEASE（2020-10-19）

Bug 修复

- 处理 Spring 组件开源漏洞风险，升级 Spring Framework 到 5.0.19 版本。
- spring-cloud-tsf-core 修复与 spring-boot-devtools 的冲突。

优化

- spring-cloud-tsf-gateway 支持服务熔断能力。
- spring-cloud-tsf-sleuth 修改调用 SQL 存储的最长长度到64000字符。

版本建议

支持向后兼容，建议全量升级。

1.21.5-Finchley-RELEASE（2020-09-09）

优化

spring-cloud-tsf-gateway 优化因配置被误删除可能导致的问题。

版本建议

支持向后兼容，建议全量升级。

1.21.4-Finchley-RELEASE（2020-08-20）

Bug 修复

- 修复 MySQL 调用链支持多数据源问题。
- 修复 feign 请求调用链只展示 HTTP 方法。
- spring-cloud-tsf-msgw： 修复 application/x-www-form-urlencoded 类型请求，当绑定插件通过 zuul 网关代理访问时出错的问题。

版本建议

支持向后兼容，建议全量升级。

1.21.3-Finchley-RELEASE（2020-07-16）

Bug 修复

- 修复网关 MSGW SDK 和服务发现 SDK 不兼容，造成拉取服务列表过快的问题。
- 修复 MySQL 调用链中 SQL 截断问题。

优化

spring-cloud-tsf-gateway 网关兼容新插件类型。

1.21.2-Finchley-RELEASE（2020-07-06）

Bug 修复

处理 tomcat 组件开源漏洞风险：

- 升级 org.apache.tomcat.embed.tomcat-embed-core到 8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.56版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.56版本。

优化

- 调整泳道标签的传递属性
- 调整泳道入口行为

1.21.1-Finchley-RELEASE（2020-04-29）

Bug 修复

- 修复泳道 ID 在非泳道起始应用中传递丢失的问题。
- 修复调用链生成文件名称问题。

1.21.0-Finchley-RELEASE (2020-04-17)

新特性

- 全链路灰度发布。
- 增加熔断状态变更事件上报。

Bug 修复

- 修复 Feign 无法使用绝对 URL 请求的问题。
- spring-cloud-tsf-swagger 修复 @ApiParam 注解 Example 属性解析异常问题。
- spring-cloud-tsf-gateway：
- 修复 Tag 标签插件未在调用中透传问题。
- 修复当绑定网关插件后造成 Query 参数未透传问题。

优化

支持 swagger 自动扫描包多路径特性。

1.20.0-Finchley-RELEASE（2020-03-02）

Bug 修复

- spring-cloud-tsf-gateway 修复 tag plugin中header 类型取值大小写敏感的问题。
- 处理 tomcat 组件开源漏洞风险。
- 升级 org.apache.tomcat.embed.tomcat-embed-core 到8.5.51版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到8.5.51版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到8.5.51版本。

优化

spring-cloud-tsf-gateway 新增 tag plugin 中 path 类型取值。

版本建议

支持向后兼容，建议全量升级。

1.19.0-Finchley-RELEASE（2020-01-16）

新特性

新增 `服务熔断` 功能。

版本建议

支持向后兼容，建议全量升级。

1.18.5-Finchley-RELEASE（2020-10-27）

Bug 修复

- 修复 druid 连接池事务兼容问题。
- 修复同时依赖多个数据库连接池问题。
- 修复调用链生成文件名称问题。
- 修复服务发现线程数不准确问题。
- 修复 Feign 无法使用绝对 URL 请求的问题。

版本建议

支持向后兼容，建议全量升级。

1.18.4-Finchley-RELEASE（2020-10-20）

优化

spring-cloud-tsf-sleuth 修改调用 SQL 存储的最长长度到64000字符。

版本建议

支持向后兼容，建议全量升级。

1.18.3-Finchley-RELEASE（2020-10-13）

Bug 修复

- 处理 Spring 组件开源漏洞风险，升级 Spring Framework 到5.0.19版本。
- spring-cloud-tsf-core 修复与 spring-boot-devtools 的冲突。

版本建议

支持向后兼容，建议全量升级。

1.18.2-Finchley-RELEASE（2020-08-21）

Bug 修复

- spring-cloud-tsf-route 修复网关使用就近命名空间的问题。
- spring-cloud-tsf-consul-discovery 修复服务发现线程池上限的问题。
- spring-cloud-tsf-sleuth 修复 MySQL 调用链支持多数据源问题。
- spring-cloud-tsf-gateway 修复网关 MSGW SDK 和服务发现 SDK 不兼容，造成拉取服务列表过快的问题。
- spring-cloud-tsf-gateway 兼容低版本 MSGW SDK。

版本建议

支持向后兼容，建议全量升级。

1.18.1-Finchley-RELEASE（2020-01-14）

Bug 修复

- spring-cloud-tsf-route 修复路由权重不准的问题。
- spring-cloud-tsf-consul-discovery 修复服务发现线程池上限的问题。
- spring-cloud-tsf-sleuth 修复 druid 连接池事务兼容问题。
- spring-cloud-tsf-sleuth 修复同时依赖多个数据库连接池问题。
- spring-cloud-tsf-core 修复 Custom Metadata 设置接口不兼容。

优化

支持通过 `tsf.discovery.watch.enabled` 关闭服务发现时的 watch 监听。

版本建议

支持向后兼容，建议全量升级。

1.18.0-Finchley-RELEASE（2019-12-25）

Bug 修复

- spring-cloud-tsf-sleuth 修复 JDBC 代理过程 NPE bug 问题。
- spring-cloud-tsf-consul-discovery 修复 ConsulProperties 中同时使用 @Value 和 @ConfigurationProperties 方式进行属性注入，先后顺序导致的 bug 问题。
- spring-cloud-tsf-sleuth 修复监控日志可能出现的 NPE bug 问题。
- spring-cloud-tsf-core 修复 ContextConfiguration Bean 初始化依赖顺序问题。

新特性

- 服务治理支持全局命名空间。
- 新增 spring-cloud-tsf-gateway 微服务网关（zuul1 版）SDK，基于此 SDK 二次研发，无缝集成 TSF 平台服务治理能力。
- 新增自定义日志配置需要的 Converter 和 Layout 类，支持用户使用自定义 logback\log4j\log4j2 日志配置。

优化

spring-cloud-tsf-sleuth 优化 TraceStatementProxyHandler JDBC 代理过程 SDK 内部异常处理逻辑：非代理异常、非 SDK 产生的异常，直接抛出；代理异常或 SDK 产生的异常，直接调用服务不经过调用链逻辑。

版本建议

支持向后兼容，建议全量升级。

1.16.2-Finchley-RELEASE (2020-03-02)

Bug 修复

spring-cloud-tsf-sleuth bug fixed：

- 处理 Custom Metadata 设置接口不兼容。
- 调用链输出用户自定义 Tag和Metadata。
- 修复 druid 连接池事务兼容问题。
- 修复同时依赖多个数据库连接池问题。

1.16.1-Finchley-RELEASE（2019-12-3）

Bug 修复

- 增加使用 jedis 作为 redis 客户端的调用链追踪功能。
- 修复因为 kafka 生产者、消费者使用 sdk 版本不匹配导致的错误。
- API 注册兼容从环境变量和启动参数中读取 TSF 参数信息。

1.16.0-Finchley-RELEASE（2019-10-11）

新特性

- kafka 的链路追踪能力。
- 增加 swagger-ui 依赖包。

优化

- 集成 spring-cloud-tsf-swagger包后，本地启动无需设置 tsf.swagger.enabled=false。
- 集成 spring-cloud-tsf-swagger包后，支持本地使用 swagger-ui 进行调试。

Bug 修复

- 修复 DEBUG 日志级别启动时，spring-cloud-tsf-sleuth 包空指针异常。
- 修复引入 swagger 包后，低版本 guava 包引起冲突。
- 配置回调功能空指针异常。

版本建议

支持向后兼容，建议全量升级。

1.14.3-Finchley-RELEASE（2019-09-10）

Bug 修复

- 限流 Bug fix。
- TsfContext.putTag 覆盖 bug fix。

版本建议

支持向后兼容，建议全量升级。

1.14.2-Finchley-RELEASE（2019-08-14）

Bug 修复

- 修复使用 RedisConnectionFactory 获取 Redis 连接，这种方式使用时的一个类型转化错误。
- 修复在给 span.tag(key,value) 传入 value 时没判空抛出异常的问题。

版本建议

支持向后兼容，建议全量升级。

1.14.1-Finchley-RELEASE（2019-07-24）

Bug 修复

- 修复 tsf sdk 依赖的 scheduler 和业务自身的 scheduler 相互影响的问题
- 修复 spring-cloud-tsf-route 包路由不生效的问题
- 修复 spring-cloud-tsf-ratelimit 包限流不准确问题
- 修复 spring-cloud-tsf-sleuth 包数据源和 Mybatis 兼容性问题

版本建议

支持向后兼容，建议全量升级。

1.14.0-Finchley-RELEASE（2019-06-21）

新特性

支持 MySQL JDBC、Redis、MongoDB、CMQ 组件调用链。

版本建议

支持向后兼容，建议全量升级。

1.12.6-Finchley-RELEASE（2021-03-25）

Bug 修复

- 处理 Spring 组件开源漏洞风险，升级 Spring Framework 到5.0.19版本。
- spring-cloud-tsf-core 修复与 spring-boot-devtools 的冲突。
- spring-cloud-tsf-ratelimit：修复多个限流规则时，全局限流无法关闭的问题。

优化

- spring-cloud-tsf-consul-discovery 心跳请求增加重试。
- spring-cloud-tsf-consul-config 支持本地加密配置解析。
- spring-cloud-tsf-swagger 支持多路径扫码。

版本建议

支持向后兼容，建议全量升级。

1.12.5-Finchley-RELEASE（2020-07-17）

Bug 修复

修复 spring-cloud-tsf-route 包路由不准确问题。

优化

调整心跳请求的超时时间，当出现丢包时能够快速重试。

版本建议

支持向后兼容，建议全量升级。

1.12.4-Finchley-RELEASE（2019-08-15）

Bug 修复

- 修复 tsf sdk 依赖的 scheduler 和业务自身的 scheduler 相互影响的问题。
- 修复 spring-cloud-tsf-route 包路由不生效的问题。
- 修复 spring-cloud-tsf-ratelimit 包限流不准确问题。

版本建议

支持向后兼容，建议全量升级。

1.12.3-Finchley-RELEASE（2019-05-17）

Bug 修复

修复 Finchley 版本服务调用监控问题。

版本建议

支持向后兼容，建议全量升级。

1.12.2-Finchley-RELEASE（2019-04-22）

Bug 修复

- 修复 Finchley 版本 TSF Route 启动问题。
- 修复 Finchley 版本 Feign HttpClient 调用链问题。

版本建议

支持向后兼容，建议全量升级。

1.12.1-Finchley-RELEASE（2019-03-25）

Bug 修复

修复配置回调功能未生效问题。

版本建议

支持向后兼容，建议全量升级。

1.12.0-Finchley-RELEASE（2019-03-13）

新特性

- 支持自动重注册，服务鉴权/路由/限流策略本地缓存。
- 服务路由支持基于可用区和地域就近访问策略。

优化

- 升级分布式配置监听，精确并减小监听范围，处理更新为空的场景，避免大范围 key 刷新事件。
- 优化分部署配置回调触发逻辑。

Bug 修复

- spring-cloud-commons 升级到1.3.1解决 RetryTemplate 会导致 LoadBalanceInterceptor thread unsafe 问题。
- 修复启用 hystrix 时配置会导致 tsf-route 与 feignbuilder 冲突的问题。

版本建议

支持向后兼容，建议全量升级。

Greenwich

最近更新时间: 2025-02-18 16:02:00

基于 Spring Cloud Greenwich 版本 SDK，支持 spring boot 2.1.6。

1.29.0-Greenwich-RELEASE（2021-06-23）

新特性

- 微服务网关增加单元化功能。
- spring-cloud-tsf-sleuth: 新增 cmq-tcp-client 和 cmq-http-client 调用支持。

优化

- 优化和开源 spring cloud consul 依赖的冲突。
- actuator 依赖改为 optional。
- spring-cloud-tsf-sleuth：优化 getProperties 性能。
- spring-cloud-tsf-sleuth：监控数据添加 http method 和 path template。
- spring-cloud-tsf-ratelimit：优化限流的 httpclient。

Bug 修复

- spring-cloud-tsf-logger：修复自定义日志格式没有服务名的问题。
- spring-cloud-tsf-sleuth：修复调用链获取 IP 偶现获取不到问题。
- spring-cloud-tsf-sleuth：修改 scg metrics duration 异常问题。
- spring-cloud-tsf-sleuth：修复未发布分组时，网关没法作为组件显示成蓝色 logo 的 Bug。
- spring-cloud-tsf-swagger：修复 IgnoreGatewayApi 注解导致的潜在空指针异常。
- spring-cloud-tsf-consul-discovery：修复被调方实例不存在时不断打印异常日志的问题。
- 修复 Feign 在指定 URL 的模式下无法请求的问题。

版本建议

支持向后兼容，建议全量升级。

1.26.2-Greenwich-RELEASE（2021-04-25）

优化

支持通过配置 `-Dspring.cloud.consul.enabled=false` 关闭连接 consul ，适配单元测试场景时的启动。

版本建议

支持向后兼容，建议全量升级。

1.26.1-Greenwich-RELEASE（2020-12-31）

优化

spring-cloud-tsf-sleuth 新增 CMQ 调用支持。

版本建议

支持向后兼容，建议全量升级。

1.26.0-Greenwich-RELEASE（2020-12-07）

新特性

- spring-cloud-tsf-msgw-scg：
- 补齐 Spring Cloud Gateway 网关的服务治理能力，支持用户按照需求灵活选择 Zuul 或 Spring Cloud Gateway。
- 支持托管外部 API。
- spring-cloud-tsf-msgw-zuul：支持托管外部 API。
- spring-cloud-tsf-swagger：支持添加注解 @IgnoreGatewayApi 来忽略某个网关 API 不被发现（忽略该网关的 API，但服务治理 API 不受影响）。

版本建议

支持向后兼容，建议全量升级。

1.25.0-Greenwich-RELEASE（2020-12-04）

新特性

spring-cloud-tsf-msgw-zuul 支持服务熔断能力。

Bug 修复

- spring-cloud-tsf-ratelimit：修复多个限流规则时，全局限流无法关闭的问题。
- spring-cloud-tsf-route：修复当只有一个路由规则时，路由规则关闭不生效的问题。
- spring-cloud-tsf-lane：修复泳道规则内存可见性 Bug。

版本建议

支持向后兼容，建议全量升级。

1.24.0-Greenwich-RELEASE（2020-09-25）

新特性

- 支持云上 Spring Cloud 应用平滑迁移 TSF。
- 支持 PostgreSQL 组件调用链。

Bug 修复

- spring-cloud-tsf-consul-config：
- 修复本地加密配置不能正确解密的问题。
- 修复 MySQL 调用链对多数据源支持。

- spring-cloud-tsf-core： 增加线程上下文接口，在父亲线程中塞入线程局部变量后，子线程不论是线程池反复使用还是一次性使用都能正确继承父线程局部变量。

版本建议

支持向后兼容，建议全量升级。

1.23.9-Greenwich-RELEASE（2021-06-11）

新特性

- 支持服务监听触发回调。

优化

- 服务发现增加零实例保护。
- consul 异常时，避免一直刷日志。

版本建议

支持向后兼容，建议全量升级。

1.23.8-Greenwich-RELEASE（2021-02-07）

Bug 修复

修复 msgw-scg 依赖 actuator 缺失导致启动失败的问题。

优化

- spring-cloud-tsf-fault-tolerance 和 spring-cloud-tsf-circuitbreaker 对 zuul 的依赖改为 optional。
- spring-cloud-tsf-route 对 actuator 依赖改为 optional。

版本建议

支持向后兼容，建议全量升级。

1.23.7-Greenwich-RELEASE（2021-01-25）

Bug 修复

- 修复服务治理时 API PATH 标签匹配 PATH 参数失败问题。
- 修复当存在多个限流规则的时候，全局限流规则开启后，无法删除的问题。
- 修复泳道规则内存可见性 Bug。
- 修复路由关闭问题。
- 修复分布式配置下发 spring.application.name 时，无法上报 swagger 问题。
- 修复本地加密配置不能被正确解密的问题。
- 修复网关多个命名空间时 consul index 混用导致第一次跨命名空间调用加载慢的问题。
- 修复 Spring Framework 反射型文件下载漏洞。

- 解决和 spring-boot-devtools 的冲突。

优化

- actuator 依赖改为 optional。
- TTL 单独超时时间，并增加重试。
- 优化 spring-cloud-tsf-sleuth 的 getProperties 性能。

版本建议

支持向后兼容，建议全量升级。

1.23.6-Greenwich-RELEASE（2020-11-11）

优化

- spring-cloud-tsf-msgw-zuul 支持服务熔断能力。
- spring-cloud-tsf-sleuth 修改调用 SQL 存储的最长长度到64000字符。

版本建议

支持向后兼容，建议全量升级。

1.23.5-Greenwich-RELEASE（2020-09-21）

优化

调整泳道入口行为。

版本建议

支持向后兼容，建议全量升级。

1.23.4-Greenwich-RELEASE（2020-09-16）

Bug 修复

- 修复 MySQL 调用链中对多数据源支持。
- 修复 feign 请求调用链中只展示 HTTP 方法。
- 修复定时任务的线程数问题。
- 修复网关使用就近命名空间的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.3-Greenwich-RELEASE（2020-09-14）

Bug 修复

- spring-cloud-tsf-msgw： 修复网关 MSGW SDK 和服务发现 SDK 不兼容，造成拉取服务列表过快的问题，从而导致注册中心负载压力过大的问题。
- spring-cloud-tsf-consul-discovery： 修复服务发现线程数不准确（少于需要请求的服务数），导致服务发现线程调度不及时，节点状态更新可能会延迟30s的问题。

1.23.2-Greenwich-RELEASE（2020-08-19）

Bug 修复

spring-cloud-tsf-msgw-zuul：

- 修复无法在 filter 中使用 Feign 发起微服务调用的问题。
- 修复 application/x-www-form-urlencoded 类型请求，当绑定插件通过 zuul 网关代理访问时错误问题。

版本建议

支持向后兼容，建议全量升级。

1.23.1-Greenwich-RELEASE（2020-08-12）

Bug 修复

spring-cloud-tsf-msgw： 修复 scg 版本网关不支持 HTTP 请求中文编码的问题。

版本建议

支持向后兼容，建议全量升级。

1.23.0-Greenwich-RELEASE（2020-07-06）

新特性

- spring-cloud-tsf-msgw：
- 新增网关路径重写配置功能。
- 新增网关微信小程序登录插件功能。
- spring-cloud-tsf-sleuth：新增调用链支持 RocketMQ。
- spring-cloud-tsf-core：
- 监控数据结构中增加 HTTP 请求方法、以及请求模版路径。
- 调用链数据结构中增加 HTTP 请求方法。

Bug 修复

- spring-cloud-tsf-msgw：
- 修复数据同步时，可能会短暂获取到错误数据的问题。
- 修复 SCG Tag 中数据未正确清除的问题。
- 处理 tomcat 组件开源漏洞风险：
- 升级 org.apache.tomcat.embed.tomcat-embed-core 到9.0.36版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到9.0.36版本。

- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到9.0.36版本。
- spring-cloud-tsf-sleuth：修复 Kafka 中的类型转发错误。

1.22.1-Greenwich-RELEASE（2020-05-06）

优化

优化 TSF MSGW scg 使用，用户无需显示配置全局路由。

版本建议

支持向后兼容，建议全量升级。

1.22.0-Greenwich-RELEASE（2020-04-29）

新特性

- 支持 SpringCloud Gateway 链路追踪和调用监控。
- TSF MSGW scg 版本发布。
- TSF MSGW zuul1 版本发布。

Bug 修复

- 修复在使用 redis，自定义多个 LettuceConnectionFactory 时，不能链路追踪所有请求的问题。
- 修复调用监控禁用场景内存泄露问题。
- 修复 Spring Cloud Gateway 无法使用 TSF 服务注册和发现的问题。

优化

优化默认日志配置支持容器部署场景。

1.21.4-Greenwich-RELEASE (2020-08-20)

bug 修复

- 处理 MySQL 中 SQL 获取截断的问题。
- 修复 MySQL 调用链中对多数据源支持。

1.21.3-Greenwich-RELEASE（2020-07-16）

Bug 修复

修复 MySQL 中 SQL 获取截断的问题。

1.21.2-Greenwich-RELEASE（2020-07-06）

Bug 修复

处理 tomcat 组件开源漏洞风险：

- 升级 org.apache.tomcat.embed.tomcat-embed-core 到9.0.36版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到9.0.36版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到9.0.36版本。

优化

- 调整泳道标签的传递属性
- 调整泳道入口行为

1.21.1-Greenwich-RELEASE（2020-04-29）

Bug 修复

修复泳道 ID 在非泳道起始应用中传递丢失的问题。

优化

- 修复调用链生成文件名称问题。

1.21.0-Greenwich-RELEASE（2020-04-17）

新特性

- 全链路灰度发布。
- 增加熔断状态变更事件上报。

Bug 修复

- 修复 Feign 无法使用绝对 URL 请求的问题
- 修复 spring-cloud-tsf-swagger 包中 @ApiParam 注解 Example 属性解析异常问题。

优化

- 支持 swagger 自动扫描包多路径特性。

1.20.0-Greenwich-RELEASE（2020-03-02）

新特性

- spring-cloud-tsf-sleuth 支持 `kafka` 和 `rocketmq` 链路追踪功能。

Bug 修复

处理 tomcat 组件开源漏洞风险。

- 升级 org.apache.tomcat.embed.tomcat-embed-core 到9.0.31版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-el 到9.0.31版本。
- 升级 org.apache.tomcat.embed.tomcat-embed-websocket 到9.0.31版本。

1.19.0-Greenwich-RELEASE（2020-01-16）

新特性

新增 `服务熔断` 功能。

版本建议

支持向后兼容，建议全量升级。

1.18.1-Greenwich-RELEASE（2020-01-14）

Bug 修复

- spring-cloud-tsf-route 修复路由权重不准的问题。
- spring-cloud-tsf-consul-discovery 修复服务发现线程池上限的问题。
- spring-cloud-tsf-sleuth 修复 druid 连接池事务兼容问题。
- spring-cloud-tsf-sleuth 修复同时依赖多个数据库连接池问题。
- spring-cloud-tsf-core 修复 Custom Metadata 设置接口不兼容。

优化

支持通过 `tsf.discovery.watch.enabled` 关闭服务发现时的 watch 监听。

版本建议

支持向后兼容，建议全量升级。

1.18.0-Greenwich-RELEASE（2019-12-25）

Bug 修复

- spring-cloud-tsf-sleuth 修复 JdbcDataSourceBeanPostProcessor NPE bug 问题。
- spring-cloud-tsf-consul-discovery 修复 ConsulProperties 中同时使用 @Value 和 @ConfigurationProperties 方式进行属性注入，先后顺序导致的 bug 问题。
- spring-cloud-tsf-sleuth 修复监控日志可能出现的 NPE bug 问题。

新特性

- 服务治理支持全局命名空间。
- 新增自定义日志配置需要的 Converter 和 Layout 类，支持用户使用自定义 logback\log4j\log4j2 日志配置。

优化

spring-cloud-tsf-sleuth 优化 TraceStatementProxyHandler JDBC 代理过程内部异常处理逻辑：非代理异常、非 SDK 产生的异常，直接抛出；代理异常或 SDK 产生的异常，直接调用服务不经过调用链逻辑。

版本建议

支持向后兼容，建议全量升级。

1.16.2-Greenwich-RELEASE（2020-03-02）

Bug 修复

- spring-cloud-tsf-sleuth bug fixed：
- 处理 Custom Metadata 设置接口不兼容。
- 调用链输出用户自定义 Tag 和 Metadata。
- 修复druid连接池事务兼容问题。
- 修复同时依赖多个数据库连接池问题。

1.16.1-Greenwich-RELEASE（2019-12-3）

Bug 修复

API 注册兼容从环境变量和启动参数中读取 TSF 参数信息。

1.16.0-Greenwich-RELEASE（2019-11-5）

新特性

服务限流（spring-cloud-tsf-rate-limit）

- 支持针对所有请求、单个服务的请求进行流量控制。
- 支持服务下单个 API 请求级别的限流。

服务路由（spring-cloud-tsf-route）

- 支持基于部署组、系统标签、自定义标签的路由设置。
- 支持服务下单个 API 请求级别的路由。
- 支持自动重注册，服务鉴权/路由/限流策略本地缓存。
- 服务路由支持基于可用区和地域就近访问策略。

服务鉴权

支持基于服务名和标签的鉴权设置。

链路跟踪（spring-cloud-tsf-sleuth）

- 支持微服务调用全链路跟踪。
- 支持 MySQL JDBC、Redis、MongoDB、CMQ 组件调用链。
- 支持在调用链上设置标签和自定义 Metada。

分布式配置（spring-cloud-tsf-config）

- 支持分布式配置功能。
- 配置回调。
- 配置加密 spring-cloud-tsf-encrypt。

API注册（spring-cloud-tsf-swagger）

- API 注册：支持服务下 API 信息自动注册，查看 API 出入参请求结构。

- 集成 spring-cloud-tsf-swagger 包，支持本地使用 swagger-ui 进行调试。

Hoxton

最近更新时间: 2025-02-18 16:02:00

基于 Spring Cloud Hoxton 版本 SDK，支持 spring boot 2.2.1。

1.27.0-Hoxton-RELEASE（2021-01-25）

新特性

服务限流（spring-cloud-tsf-rate-limit）

- 支持针对所有请求、单个服务的请求进行流量控制。
- 支持服务下单个 API 请求级别的限流。

服务路由（spring-cloud-tsf-route）

- 支持基于部署组、系统标签、自定义标签的路由设置。
- 支持服务下单个 API 请求级别的路由。
- 支持自动重注册，服务鉴权/路由/限流策略本地缓存。
- 服务路由支持基于可用区和地域就近访问策略。

服务鉴权

支持基于服务名和标签的鉴权设置。

链路跟踪（spring-cloud-tsf-sleuth）

- 支持微服务调用全链路跟踪。
- 支持 MySQL JDBC、Redis、MongoDB、CMQ 组件调用链。
- 支持在调用链上设置标签和自定义 Metada。

分布式配置（spring-cloud-tsf-config）

- 支持分布式配置功能。
- 配置回调。
- 配置加密 spring-cloud-tsf-encrypt。

API注册（spring-cloud-tsf-swagger）

- API 注册：支持服务下 API 信息自动注册，查看 API 出入参请求结构。
- 集成 spring-cloud-tsf-swagger 包，支持本地使用 swagger-ui 进行调试。

SDK版本概述

最近更新时间: 2025-02-18 16:02:00

版本配套关系说明

TSF 目前支持 Spring Cloud Edgware、Spring Cloud Finchley、Spring Cloud Greenwich、Spring Cloud Hoxton 四个版本。Spring Cloud 、Spring Boot 及 TSF SDK 版本之间的关系如下表所示。

Spring Cloud	Spring Boot
Hoxton	2.2.x
Greenwich	2.1.x
Finchley	2.0.x
Edgware	1.5.x

> **说明：**
>
> 2020年5月19日起，TSF 主要支持 Greenwich 和 Finchley 版本的功能更新，Edgware 版本主要进行缺陷修复（[社区 Edgware 版本] 于2019年8月停止更新）。

长期维护 SDK 版本

TSF 长期维护 LTS (Long Term Support) 版本，SDK 的第三位版本号会根据缺陷修复递增。

SDK 版本号	新增特性
1.29.x	微服务网关增加单元化功能；微服务网关增加 Dubbo 协议转换功能；补齐 Spring Cloud Gateway 网关的服务治理能力；微服务网关支持托管外部 API；支持云上 Spring Cloud 应用平滑迁移 TSF；调用链支持 CMQ、PostgreSQL。
1.23.x	新增 spring cloud gateway 微服务网关，链路追踪和调用监控；微服务网关路径重写配置和微信小程序登录插件 ；调用链支持 RocketMQ。
1.21.x	支持服务熔断；支持服务容错；支持全链路灰度发布。
1.18.x	支持微服务网关（zuul1 版）SDK，基于此 SDK 二次研发，无缝集成 TSF 平台服务治理能力 ；增加 swagger-ui 依赖包 ；调用链支持 MySQL JDBC、Redis、MongoDB、CMQ、Kafka；支持全局命名空间；新增自定义日志配置需要的 Converter 和 Layout 类，支持用户使用自定义 logback\log4j\log4j2 日志配置
1.12.x	支持服务限流；支持服务路由；支持服务鉴权；支持分布式配置；支持调用链。

SDK 版本使用说明

私有化 TSF

对于 TSF 私有云的用户，SDK 版本号需要和 TSF 平台版本**保持一致**，SDK 的缺陷会在第三位版本号上体现，例如用户使用 TSF 1.12.4 版本，推荐使用的 SDK 版本为 1.12.x。

TSF 私有化平台版本	Edgware	Finchley	Greenwich
1.29.x	-	1.29.0-Finchley-RELEASE	1.29.0-Greenwich-RELEASE
1.23.x	-	1.23.7-Finchley-RELEASE	1.23.8-Greenwich-RELEASE
1.21.x	1.21.5-Edgware-RELEASE	1.21.9-Finchley-RELEASE	1.21.4-Greenwich-RELEASE
1.18.x	1.18.1-Edgware-RELEASE	1.18.5-Finchley-RELEASE	1.18.1-Greenwich-RELEASE
1.12.x	1.12.5-Edgware-RELEASE	1.12.5-Finchley-RELEASE	-

Spring Cloud SDK 更新日志

最近更新时间: 2025-02-18 16:02:00

TSF SDK Release Note

- Spring-Cloud-TSF-Edgware
- Spring-Cloud-TSF-Finchley
- Spring-Cloud-TSF-Greenwich
- Spring-Cloud-TSF-Hoxton

Spring Cloud 应用接入服务治理

最近更新时间: 2025-02-18 16:02:00

操作场景

服务治理功能包含服务鉴权、服务限流、服务路由和服务熔断，该任务指导您在应用开发过程中配置服务治理功能。

前提条件

- 完成 【SDK 下载】 。

> **注意：**
>
> 若要使用服务熔断功能，请确保 SDK 版本高于**1.19**。

- 熟悉服务治理功能 。

添加依赖

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】 。

1. **向工程中添加依赖。** 在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. **向 Application 类中添加注解 `@EnableTsf`** ：

```
// 下面省略了无关的代码
@SpringBootApplication
@EnableTsf
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

服务鉴权

您需要在服务主调方和被调方都添加依赖项和注解的开启。此时您已经对服务开启了鉴权功能，任何到达服务的请求都会被鉴权，鉴权不通过时会返回 HTTP 403 Forbidden。

如果请求双方想使用基本 tag 的鉴权规则，那么：

- 对于 provider 而言，需要在控制台上设置 tag 鉴权规则。
- 对于 consumer 而言，需要在业务代码中设置 tag 的内容。

1. 控制台上配置鉴权规则 。
2. 在 consumer 中设置 tag ，使用 `org.springframework.tsf.core` 包中的 `TsfContext` 类。设置 Tag 的方法签名如下：

```
/**
* 设置多个 tag。如果有某个 tag 之前已经被设置过，那么它的值会被覆盖。
*/
public static void putTags(Map<string, string=""> tagMap, Tag.ControlFlag... flags) {}</string,></dx-codeblock>

/\*\*
\* 设置单个 tag。如果该 key 之前已经被设置过，那么它的值会被覆盖。
\*/
public static void putTag(String key, String value, Tag.ControlFlag... flags) {}
```

其中 `flags` 决定 tag 的使用场景，如果您没有特殊需要，不传即可：

```
public enum ControlFlag {
TRANSITIVE, // 表示标签要传递下去，默认不启用。
NOT_IN_AUTH, // 表示标签不被使用在服务鉴权，默认是被使用的。
NOT_IN_ROUTE, // 表示标签不被使用在服务路由，默认是被使用的。
NOT_IN_SLEUTH // 表示标签不被使用在调用链，默认是被使用的。
}
```

TSF 提供的 Demo `consumer-demo/src/main/java/com/tsf/demo/consumer/Controller.java` 中提供了一个设置 tag 的例子：

```
@RequestMapping(value = "/echo-rest/{str}", method = RequestMethod.GET)
public String rest(@PathVariable String str, @RequestParam String user) {
TsfContext.putTag("user", user);
return restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/" + str, String.class);
}
```

服务限流

在应用工程中添加依赖和注解后，您可以直接在控制台上配置限流规则 。

当您开启服务限流功能后，任何到达的请求都会被限流模块处理。如果该服务上的配额已经消耗完，会对请求返回 HTTP 429 Too Many Requests；否则会正常放行。

服务路由

在应用工程中添加依赖和注解后，您可以直接在控制台上配置限流规则 。

服务熔断

TSF 摒弃了已经不再继续维护的 Hystrix 断路器，采用官方推荐的 Resilience4J 作为底层实现。相比较原有单一的接口级别熔断，我们在此之上扩展成为：实例、API、服务级别熔断。同时，除了错误比率，TSF 也支持超时比率熔断，允许用户根据业务自行选择熔断配置。请配合 TSF 其他功能一起使用。

1. **关闭 Hystrix。** 使用 TSF 熔断功能需要将 Hystrix 关闭（默认是关闭的，如果之前有打开还请关闭） 。

> **注意：**
>
> 如果您已经使用了 feign 的 fallback 或者 fallbackFactory 功能，此处将会不再生效。如果您需要继续使用该功能或需要使用更丰富的容错功能 。

```
feign:
hystrix:
enabled: false
```

2. 进行熔断配置

目前支持两种方式的熔断配置：

- 方式一：在线动态配置下发 。
- 方式二：本地静态配置（适合本地联调使用，如果线上使用会被在线配置覆盖，请谨慎使用），配置在 yaml 配置文件中。

```
#此处为namespaceId，若为本地联调，则保持一致即可。
#线上环境不需要此字段
tsf_namespace_id: default_namespace

tsf:
circuit-breaker:
# 可以配置多条规则
rules:
# 需要熔断的目标微服务名
- targetServiceName: provider-demo
# 熔断级别 API/SERVICE/INSTANCE
isolationLevel: API
# 目标熔断服务的namespaceId，如果本地联调，需与tsf_namespace_id保持一致
targetNamespaceId: "default_namespace"
# SERVICE和INSTANCE级别，只允许配置一个策略
# API级别可以针对不同的API配置多个策略，也可以多个API配置一个策略
strategyList:
# 滑动窗口大小
- slidingWindowSize: 10
# 最小熔断请求数
minimumNumberOfCalls: 10
# 熔断错误比例
failureRateThreshold: 60
```

```
# 打开到半开状态的时间
waitDurationInOpenState: 5
# 最大熔断实例个数百分比
# 只在INSTANCE级别生效
maxEjectionPercent: 51
# 慢请求阈值，单位为ms
slowCallDurationThreshold: 60000
# 慢请求熔断比例
slowCallRateThreshold: 50
# 该策略作用的API，可以同时作用于多个API
apiList:
- method: GET
path: "/echo/{param}"
- method: GET
path: "/echo2/{str}"
```

服务注册与发现

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您在本地开发Spring Cloud微服务示例应用并注册到TSF 服务注册发现中心，或者将已经接入 Eureka 服务注册与发现的应用迁移到TSF 服务注册发现中心。

前提条件

开始实践服务注册发现功能前，请确保已完成 【SDK 下载】。

本地开发应用

创建 tsf-demo 工程，文件结构如下：

```
|- consumer-demo
|- provider-demo
|- pom.xml
```

其中 `pom.xml` 文件参考 [Demo 工程概述] 中的 `pom.xml` 内容。

1. 创建服务提供者

在本地创建服务提供者应用工程，此服务提供者提供一个简单的 echo 服务，并将自身注册到服务注册中心。

1.1 创建 provider 工程

创建一个 Spring Cloud 工程，命名为 `provider-demo` 。

1.2 修改 pom 依赖

在 `pom.xml` 中引入需要的依赖内容：

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version><!-- 关联 parent version 属性--></version>
</parent>

<artifactId>provider-demo</artifactId>
<packaging>jar</packaging>
<name>provider-demo</name>

<dependencies>
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
</dependency>
</dependencies>
```

> **注意：**
>
> Finchley 版本 SDK 无须添加 monitor 依赖包 。

1.3 开启服务注册发现

添加服务提供端的代码。

```
// 省略部分 import
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.tsf.annotation.EnableTsf;

@SpringBootApplication
@EnableFeignClients // 使用Feign微服务调用时请启用
@EnableTsf
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

1.4 提供 echo 服务

创建一个 `EchoController` ，提供简单的 `echo` 服务。

```
@RestController
public class EchoController {
@RequestMapping(value = "/echo/{string}", method = RequestMethod.GET)
public String echo(@PathVariable String string) {
return string;
}
}
```

1.5 修改配置

在 `resource` 目录下的 `application.yml` 文件中配置应用名与监听端口号。

```
server:
port: 18081
spring:
application:
name: provider-demo
```

> **注意：**
>
> 运行在 TSF 平台上的应用无须配置服务注册中心地址，SDK 会通过环境变量自动获取注册中心地址。

2. 创建服务消费者

本小节中，我们将创建一个服务消费者，消费者通过 `RestTemplate` 、 `AsyncRestTemplate` 、 `FeignClient` 这三个客户端去调用服务提供者。

2.1 创建 consumer 工程

创建一个 Spring Cloud 工程，命名为 `consumer-demo` 。

2.2 修改 pom 依赖

在 pom.xml 中引入需要的依赖内容：

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version><!-- 关联 parent version 属性--></version>
</parent>

<artifactId>consumer-demo</artifactId>
<packaging>jar</packaging>
<name>consumer-demo</name>

<dependencies>
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
</dependency>
</dependencies>
```

2.3 开启服务注册发现

与服务提供者 provider-demo 相比，除了开启服务与注册外，还需要添加两项配置才能使用 RestTemplate 、 AsyncRestTemplate 、 FeignClient 这三个客户端：

- 添加 @LoadBalanced 注解将 RestTemplate 与 AsyncRestTemplate 与服务发现结合。
- 使用 @EnableFeignClients 注解激活 FeignClients 。

```
// 省略部分 import
import org.springframework.cloud.netflix.feign.EnableFeignClients;
import org.springframework.tsf.annotation.EnableTsf;
import org.springframework.web.client.AsyncRestTemplate;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
@EnableFeignClients // 使用 Feign 微服务调用时请启用
@EnableTsf
public class ConsumerApplication {
@LoadBalanced
@Bean
public RestTemplate restTemplate() {
return new RestTemplate();
}

@LoadBalanced
@Bean
public AsyncRestTemplate asyncRestTemplate() {
return new AsyncRestTemplate();
}

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

2.4 设置调用信息

在使用 EchoService 的 FeignClient 之前，还需要完善它的配置。配置服务名以及方法对应的 HTTP 请求，服务名为 provider-demo 工程中配置的服务名 provider-demo ，代码如下：

```
@FeignClient(name = "provider-demo")
public interface EchoService {
@RequestMapping(value = "/echo/{str}", method = RequestMethod.GET)
String echo(@PathVariable("str") String str);
}
```

2.5 创建 Controller

创建一个 Controller 供调用测试。

- /echo-rest/* 验证通过 RestTemplate 去调用服务提供者。
- /echo-async-rest/* 验证通过 AsyncRestTemplate 去调用服务提供者。
- /echo-feign/* 验证通过 FeignClient 去调用服务提供者。

```
@RestController
public class Controller {
@Autowired
private RestTemplate restTemplate;
@Autowired
private AsyncRestTemplate asyncRestTemplate;
@Autowired
private EchoService echoService;
@RequestMapping(value = "/echo-rest/{str}", method = RequestMethod.GET)
public String rest(@PathVariable String str) {
return restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/" + str, String.class);
}
@RequestMapping(value = "/echo-async-rest/{str}", method = RequestMethod.GET)
public String asyncRest(@PathVariable String str) throws Exception{
ListenableFuture<ResponseEntity<String>> future = asyncRestTemplate.
getForEntity("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/"+str, String.class);
return future.get().getBody();
}
@RequestMapping(value = "/echo-feign/{str}", method = RequestMethod.GET)
public String feign(@PathVariable String str) {
return echoService.echo(str);
}
}
```

2.6 修改配置

```
server:
port: 18083
spring:
application:
name: consumer-demo
```

> **注意：**
>
> 运行在 TSF 平台上的应用无须配置服务注册中心地址，SDK 会通过环境变量自动获取注册中心地址。

3. 部署应用

将打包好的 FatJar 程序包上传到 TSF 控制台，进行部署操作，无需关心额外配置。

从 Eureka 迁移应用

已经接入 Eureka 服务注册与发现的应用，只需要修改 pom.xml 依赖，就可以将服务接入 TSF 服务注册发现中心。

1. 在工程根目录的 pom.xml 中增加 spring-cloud-tsf-dependencies 的 parent。参考上文中的 Demo 工程。
2. 在单个 Spring Cloud 应用的 pom.xml 中，将 spring-cloud-starter-eureka 替换成 spring-cloud-tsf-consul-discovery 。

- 替换前：

```
<dependency>
<groupid>org.springframework.cloud</groupid>
<artifactid>spring-cloud-starter-eureka</artifactid>
</dependency>
```

- 替换后：

```
<dependency>
<groupid>com.tencent.tsf</groupid>
<artifactid>spring-cloud-tsf-consul-discovery</artifactid>
</dependency>
<!-- consul SDK 依赖的版本控制参考 Demo 或之前 pom 说明-->
```

3. 修改代码中的 Eureka 的相关注解。

```
@EnableEurekaClient =&gt; @EnableDiscoveryClient
```

服务熔断

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 摒弃了已经不再继续维护的 Hystrix 断路器，采用官方推荐的 Resilience4J 作为底层实现。相比较原有单一的接口级别熔断，我们在此之上扩展成为：实例、API、服务级别熔断。同时，除了错误比率，TSF 也支持超时比率熔断，允许用户根据业务自行选择熔断配置。请配合 TSF 其他功能一起使用。

前提条件

开始实践服务熔断功能前，请确保已完成了 [SDK 下载] ，同时请确保 SDK 版本高于**1.19**。

操作步骤

> **注意：**
>
> 步骤1和步骤2与其他模块相同，已经使用过其他模块的可直接跳至步骤3。

1. **向工程中添加依赖。** 在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. **向 Application 类中添加注解** `@EnableTsf` ：

```
// 下面省略了无关的代码
@SpringBootApplication
@EnableTsf
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

3. **关闭 Hystrix。** 使用 TSF 熔断功能需要将 Hystrix 关闭（默认是关闭的，如果之前有打开还请关闭）。

> **注意：**
>
> 如果您已经使用了 feign 的 fallback 或者 fallbackFactory 功能，此处将会不再生效。如果您需要继续使用该功能或需要使用更丰富的容错功能，请参考 【服务容错】 步骤3进行配置。

```
feign:
hystrix:
enabled: false
```

4. **进行熔断配置。** 目前支持两种方式的熔断配置：

- 方式一：在线动态配置下发。
- 方式二：本地静态配置（适合本地联调使用，如果线上使用会被在线配置覆盖，请谨慎使用），配置在 yaml 配置文件中。

```
#此处为namespaceId，若为本地联调，则保持一致即可。
#线上环境不需要此字段
tsf_namespace_id: default_namespace</dx-codeblock>

tsf:
circuit-breaker:
\# 可以配置多条规则
rules:
\# 需要熔断的目标微服务名
\- targetServiceName: provider-demo
\# 熔断级别 API/SERVICE/INSTANCE
isolationLevel: API
\# 目标熔断服务的namespaceId，如果本地联调，需与tsf_namespace_id保持一致
targetNamespaceId: "default_namespace"
\# SERVICE和INSTANCE级别，只允许配置一个策略
\# API级别可以针对不同的API配置多个策略，也可以多个API配置一个策略
strategyList:
\# 滑动窗口大小
\- slidingWindowSize: 10
\# 最小熔断请求数
minimumNumberOfCalls: 10
\# 熔断错误比例
failureRateThreshold: 60
\# 打开到半开状态的时间
waitDurationInOpenState: 5
\# 最大熔断实例个数百分比
\# 只在INSTANCE级别生效
maxEjectionPercent: 51
\# 慢请求阈值，单位为ms
slowCallDurationThreshold: 60000
\# 慢请求熔断比例
slowCallRateThreshold: 50
\# 该策略作用的API，可以同时作用于多个API
apiList:
\- method: GET
path: "/echo/{param}"
\- method: GET
path: "/echo2/{str}"
```

服务监听触发回调

最近更新时间: 2025-02-18 16:02:00

操作场景

服务监听允许程序监听特定服务的上下线情况，从而触发对应的业务逻辑。

前提条件

完成 【SDK 下载】。

> **注意：**
>
> 若要使用服务监听触发回调功能，请确保 SDK 版本高于**1.32**。

添加依赖

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

使用说明

监听 bean 需要使用 `@ConsulServiceChangeListener` 或 `@LocalServiceChangeListener` 注解，并且实现 `ConsulServiceChangeCallback` 接口。callback 接口有三个入参：

- currentServices：表示当前在线的服务实例列表。
- addServices：表示和上一次变更相比，新增的实例列表。
- deleteServices：表示和上一次变更相比，删除的实例列表。

实现原理

初始化 spring bean 时，当检测到该 bean 有使用 `@ConsulServiceChangeListener` 或 `@LocalServiceChangeListener` 注解，并且有实现 `ConsulServiceChangeCallback` 接口时，开启一个定时任务，与注册中心 consul 保持长轮序，当监听服务有实例上线或下线时，会立刻触发相关事件并回调对应 bean 的 callback 方法。

> **注意：**
>
> 每个 bean 会启动一个额外线程执行，线程池的默认大小为10，如果监听服务的数量较多或 callback 里的业务逻辑较为复杂时可能会导致线程不够用，此时可以通过 spring.cloud.consul.discovery.callbackPoolSize 参数进行调整。

相关注解说明及使用示例

使用 `@ConsulServiceChangeListener` 注解：

```
/**
* @ConsulServiceChangeListener 监听当前命名空间或全局命名空间的任意服务
* serviceName: 监听服务的服务名
* global: 监听服务是否全局命名空间，默认 false。注：如果当前服务位于全局命名空间，global 属性无论 true 还是 false，都是监听全局命名空间的服务
*/
@Component
@ConsulServiceChangeListener(serviceName = "provider-demo", global = false)
public class ProviderDemoChangeCallback implements ConsulServiceChangeCallback {

private static final Logger log = LoggerFactory.getLogger(ProviderDemoChangeCallback.class);

@Override
public void callback(List<HealthService> currentServices, List<HealthService> addServices, List<HealthService> deleteServices) {
log.info("current size:{}, add size:{}, delete list:{}", currentServices.size(), addServices.size(), deleteServices.size());
log.info("current list:{}, add list:{}, delete list:{}", currentServices, addServices, deleteServices);
for (HealthService healthService: currentServices) {
// 可以从 meta 信息中获取节点对应的 TSF 信息，如部署组id、应用id、包版本
Map<String, String> meta = healthService.getService().getMeta();
String groupId = meta.get("TSF_GROUP_ID");
String applicationId = meta.get("TSF_APPLICATION_ID");
String progVersion = meta.get("TSF_PROG_VERSION");
}
}
}
```

使用 `@LocalServiceChangeListener` 注解：

```
/**
* @LocalServiceChangeListener 监听当前服务(spring.application.name)
* excludeLocalInstance: 回调的 currentServices 是否包含当前实例，默认为 false
*/
@Component
@LocalServiceChangeListener(excludeLocalInstance = false)
public class LocalServiceChangeCallback implements ConsulServiceChangeCallback {

private static final Logger log = LoggerFactory.getLogger(LocalServiceChangeCallback.class);

@Override
public void callback(List<HealthService> currentServices, List<HealthService> addServices, List<HealthService> deleteServices) {
log.info("current size:{}, add size:{}, delete list:{}", currentServices.size(), addServices.size(), deleteServices.size());
log.info("current list:{}, add list:{}, delete list:{}", currentServices, addServices, deleteServices);
}
}
```

服务监控

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您使用服务监控功能。

前提条件

开始实践服务监控功能前，请确保已完成了 【SDK 下载】 。

操作步骤

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】 。

Finchley 版本 SDK 相关设置

1. 向工程中添加依赖。在 `pom.xml` 中添加以下代码，**依赖的是** `spring-cloud-tsf-sleuth` 而不是 `spring-cloud-tsf-monitor` 。

```
<dependency>
<groupid>com.tencent.tsf</groupid>
<artifactid>spring-cloud-tsf-sleuth</artifactid>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsfMonitor` ：

```
// 下面省略了无关的代码
import com.tencent.tsf.monitor.annotation.EnableTsfMonitor;
@SpringBootApplication
@EnableTsfMonitor
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

服务路由

最近更新时间: 2025-02-18 16:02:00

准备工作

开始实践服务路由功能前，请确保已完成了 【SDK 下载】 。

快速上手

使用服务路由功能前，您需要在 `pom.xml` 中添加路由依赖项，同时在代码中使用路由开关注解。

1. 向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。
2. 路由规则配置 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】 。

服务限流

最近更新时间: 2025-02-18 16:02:00

准备工作

开始实践服务限流功能前，请确保已完成了 【SDK 下载】 。

快速上手

1. 向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。
2. 限流规则配置 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】。

验证结果

此时您已经对服务开启了限流功能，任何到达的请求都会被限流模块处理。如果该服务上的配额已经消耗完，会对请求返回 HTTP 429 Too Many Requests；否则会正常放行。

本地开发联调

最近更新时间: 2025-02-18 16:02:00

本文介绍了两种本地开发联调的使用场景：

- 本地服务之间调用
- 本地服务调用云端服务

本地服务之间调用

操作场景

开发者通过搭建本地轻量级注册中心，将本地服务注册到轻量级注册中心上，服务之间通过服务名来进行调用。

操作步骤

1. 启动轻量级注册中心

本地开发调试时，需要使用轻量级注册中心，轻量级注册中心包含了 TSF 服务注册发现服务端的轻量版 。

2. 启动应用

本地启动应用可以通过 IDE 和 FatJar 两种方式。

IDE 中启动

在 IDE 中启动，通过 VM options 配置启动参数 `-Dtsf_consul_ip=127.0.0.1 -Dtsf_consul_port=8500 -Dtsf_application_id=a -Dtsf_group_id=b -Dtsf.swagger.enabled=false` ，通过 main 方法直接启动。其中 IP 和 port 取值为轻量级服务注册中心的地址，使用了分布式配置功能的模块，需要设置 `-Dtsf_application_id=a -Dtsf_group_id=b` ， 取值可为任意值。

FatJar 启动

（1）添加 FatJar 打包方式

```
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
```

（2）打包 FatJar 文件

添加完插件后，在工程的主目录下，使用 Maven 命令 `mvn clean package` 进行打包，即可在 target 目录下找到打包好的 FatJar 文件。

（3）通过 Java 命令启动

```
java -Dtsf_instance_id=1111 -Dtsf_consul_ip=127.0.0.1 -Dtsf_consul_port=8500 -Dtsf.swagger.enabled=false -jar provider-demo-0.0.1-SNAPSHOT.jar
```

其中 `127.0.0.1` 和8500为轻量级服务注册中心地址，在本地调试时 tsf_application_id 和 tsf_group_id 可以填任意值。

> **注意：**

- Instance_id 是服务实例唯一标识，需要保证唯一性。。
- 由于轻量级服务注册中心（原生的 consul）的 metadata 只能支持512个字节，因此需要关闭 TSF 的 API 上报能力 `-Dtsf.swagger.enabled=false` ，如果没有这个启动参数，在本地运行 Demo 时将会报错，错误信息中包含 `Value is too long (limit 512 characters)` 。

3. 验证

启动服务，分别进行调用，观察调用结果。

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/echo-rest/test?user=test-tsf`

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/echo-async-rest/test?user=test-tsf`

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/test?user=test-tsf`

本地服务调用云端服务

操作场景

如果开发者需要使本地正在开发的微服务和远端的微服务做联调，不需在本地启动服务注册中心，可直接使用本文提供的轻量级联调操作方法。其中，远端的微服务指通过 TSF 部署的 Spring Cloud 应用。 假设本地开发环境的 consumer-demo 服务，希望调用 TSF 上的

provider-demo 服务提供的接口，provider-demo 服务有一个实例提供**外网 IP**，服务监听端口为 18081。

前提条件

- 依赖 1.11.0-RELEASE 及以上版本的 SDK。
- 确保本地开发机和 provider-demo 的实例的网络连通性，确保从外网可访问 provider-demo 的端口号。
- 本地未启动 consul 或者 consumer-demo 未连接本地 consul ，否则 consumer-demo 会通过本地 consul 进行服务注册与发现。

操作步骤

1. 在本地开发机器上，创建 `${System.getProperty(&quot;user.home&quot;)}/.tsf/discovery/` 目录，其中 ${System.getProperty("user.home")} 是本地开发机器的 home 目录。
2. 在该目录下创建服务缓存文件，缓存文件名称为 `${service-name}.cache` ，其中 `${service-name}` 是被调服务名，在本例中是 provider-demo.cache。 在 provider-demo.cache 中填写远端服务的描述信息：

```
{
"statusCode": 200, # statusCode 字段必填，取值为200，表示正常状态码；请勿更改；
"content": [{
"service": {
"id": "provider-demo-18081", # 随机字符，确保唯一性
"service": "provider-demo", # 被调服务的服务名
"tags": [],
"address": "<被调服务的实例IP>", # 被调服务实例的 IP
"port": 18081, # 被调服务监听端口
"meta": {
}
},
"checks": []
}]
}
```

实际使用过程中，开发者仅需关注 content.service 结构体中的内容（id、service、address、port），其他字段内容可以复用上述示例。

3. 启动 consumer-demo，启动命令中包含必要的 启动参数：

```
java -Dtsf_instance_id=1111 -Dspring.cloud.consul.config.fail-fast=false -Dspring.cloud.consul.discovery.register=false -Dspring.cloud.consul.discovery.fail-fast=false -Dspring.cloud.consul.discovery.catalogServicesWatch.enabled=false -Dspring.cloud.consul.config.watch.enabled=false -jar consumer-demo-0.0.1-SNAPSHOT.jar
```

> **注意：**
>
> Instanceid是服务实例唯一标识，需要保证唯一性。

4. consumer-demo 会轮询调用 provider-demo 的接口，如果控制台的日志显示正常，说明调用成功。

启动参数说明

参数	含义	默认值
spring.cloud.consul.discovery.register	是否开启 consul 服务注册能力。	true
spring.cloud.consul.discovery.fail-fast	是否开启服务注册发现快速失败能力。关闭后，如果发起服务注册失败，会继续进行服务启动。	true
spring.cloud.consul.discovery.catalogServicesWatch.enabled	是否开启服务注册中心的异常请求。关闭后，会减少到服务注册中心的异常请求。	true
spring.cloud.consul.config.watch.enabled	分布式配置长轮询监听定时任务。关闭后，会减少连接服务注册中心的异常日志输出。	true
spring.cloud.consul.config.fail-fast	是否开启分布式配置快速失败功能。关闭后，如果请求 consul 失败，则输出异常日志，继续进行服务启动。	true

调用链
CMQ 链路追踪

最近更新时间: 2025-02-18 16:02:00

CMQ 调用链组件目前支持使用 cmq-http-client 和 cmq-tcp-client 两种方式。

cmq-http-client 方式

1. 引入 CMQ 的依赖1.0.7.4以上版本（低版本不支持）：

```
<dependency>
<groupId>com.qcloud</groupId>
<artifactId>cmq-http-client</artifactId>
<version>1.0.7.4</version>
</dependency>
```

2. 直接使用已经构建好的 bean： @Autowired private Account account;
3. CMQ 接入配置：

```
cmq:
server:
endpoint: http://imgcache.finance.cloud.tencent.com:80ocloud-cmq-queue-nameserver # cmq的端点地址
secret-id: ******* # 获取账号secret-id
secret-key: ****** # 获取账号secret-key
```

cmq-tcp-client 方式

1. 在 pom 中引入 tcp 的依赖，版本要求1.1.2以上（低版本不支持）：

```
<dependency>
<groupId>com.qcloud</groupId>
<artifactId>cmq-tcp-client</artifactId>
<version>1.1.2</version>
</dependency>
```

2. 直接使用已经构建好的 bean： @Autowired private Consumer consumer;

@Autowired private Producer producer;

3. CMQ 接入配置：

```
cmq:
server:
endpoint: http://imgcache.finance.cloud.tencent.com:80ocloud-cmq-queue-nameserver # CMQ 的端点地址
```

```
secret-id: ******* # 获取账号 SecretId
secret-key: ****** # 获取账号 SecretKey
```

注意事项

- TSF SDK 1.21.10-Finchley 以及 1.21.x后续版本支持。
- TSF SDK 1.28.1-Finchley 以及上版本提供支持。
- 单条接收的调用链信息在同个线程中自动传递，批量接收的场景在接收之后的流程里面需要继续传递调用链则需要使用如下方法将调用链信息注入：

```
plaintext
@Autowired
private JoinSapn joinSapn;</dx-codeblock>

joinSapn.joinBatchReceiveSpan(message);
```

MongoDB 链路追踪

最近更新时间: 2025-02-18 16:02:00

- 考虑到 spring-data-mongodb 库的易用性，TSF 目前只对 spring-boot-starter-data-mongodb 进行支持，在引用 spring-boot-starter-data-mongodb 时不要指定版本，只需要整个工程依赖 parent pom 即可，示例如下：

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-dependencies</artifactId>
<version>tsf的版本号（1.14以后开始支持 MongoDB）</version>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>
</parent>
```

spring-boot-starter-data-mongodb 的版本即是 parent pom 文件管理的 mongoldb starter 的版本。在代码中具体使用时，引入 MongoTemplate，然后使用其方法即可。

  - 如果需要制定 MongoDB 连接的 URI，需要满足以下格式：

```
mongodb://host[:port1][/[database][?options]] #暂时只支持单节点 MongonDB
```

也可以不用填写，即默认 host 是本机，默认端口27017。

  - 如果通过其他方式引入 MongoDB 客户端，例如直接 new MongoClient(host,port) ，则在 TSF 的链路中将无法查看到相应的信息。

MySQL 链路追踪

最近更新时间: 2025-02-18 16:02:00

TSF 支持实现 JDBC 规范的 MySQL 驱动器和各类连接池（如 Tomcat-JDBC、DBCP、Hikari、Druid），在使用时需引入 SpringBoot 相关依赖和 MySQL 驱动依赖：

```
<!-- spring-boot-starter-jdbc -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jdbc</artifactId>
<version>RELEASE</version>
</dependency>
<!-- mysql -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>RELEASE</version>
</dependency>
```

引入依赖后，根据需要添加相关数据库连接池依赖或直接使用 SpringBoot 默认选项。

Redis 链路追踪

最近更新时间: 2025-02-18 16:02:00

添加依赖

考虑到 Redis 库的多样性，以及 spring-data-redis 库的易用性，目前只对 `spring-boot-starter-data-redis` 进行支持，在引用 spring-boot-starter-data-redis 时不要指定版本，只需要整个工程依赖 parent pom 即可：

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-dependencies</artifactId>
<version>tsf 的版本号（1.14以后开始支持 Redis）</version>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
</parent>
```

`spring-boot-starter-data-redis` 的版本为 parent pom 文件管理的 Redis Starter 的版本。在代码中具体使用时，引入 RedisTemplate，然后使用其方法即可。不建议直接引用 Jedis 和 Lettuce 相关的依赖，spring-boot-starter-data-redis 会自动引用相关的依赖，并做适配。 如果通过其他方式引入 Redis 客户端（例如直接 new Jedis），则将无法在 TSF 的链路中查看到相应的信息。

16.1-Finchley-RELEASE 版本 Redis 调用链使用方式

lettuce 方式

使用 lettuce 方式，pom 依赖如下：

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

不需要额外依赖，自动依赖了 lettuce 方面的 jar 包，可以直接使用。通过该配置，也具有链路追踪的能力。

jedis 方式

使用 jedis 方式，pom 依赖如下：

```
<dependency>
<groupId>org.springframework.data</groupId>
<artifactId>spring-data-redis</artifactId>
</dependency>
<!-- redis -->
<dependency>
<groupId>redis.clients</groupId>
<artifactId>jedis</artifactId>
</dependency>
```

通过以上配置，也能使用 SDK 的链路追踪能力。

RocketMQ 链路追踪

最近更新时间: 2025-02-18 16:02:00

TSF 从1.23.0版本开始支持 RocketMQ 使用联路追踪能力。

链路追踪原理

利用 springboot 提供自动配置原理，加入一个能插手 bean DefaultMQProducer、和 bean DefaultMQPushConsumer 创建过程的自动配置类。使用代理来增强 DefaultMQProducer/DefaultMQPushConsumer，在调用相应方法是加入 tracing 的逻辑，在方法结束时将 tracing 数据搜集起来，统一存储。最后分析展示给前端页面。

引入依赖及配置

```
<!-- TSF 启动器 -->
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
</dependency>
<!--RocketMQ-->
<dependency>
<groupId>org.apache.rocketmq</groupId>
<artifactId>rocketmq-client</artifactId>
<version>4.4.0</version>
</dependency>
apache:
rocketmq:
consumer:
pushConsumer: myConsumer
producer:
producerGroup: myProducer
name-server: xxx.xxx.xx.xx #rocketMq ip
topic: long-topic
```

生产者示例

向 Spring 容器中加入 bean DefaultMQProducer：

```
@Configuration
public class ProducerConfiguration {
private static final Logger logger = LoggerFactory.getLogger(ProducerConfiguration.class);
@Value("${apache.rocketmq.name-server}")
private String namesrvAddr;
@Value("${apache.rocketmq.producer.producerGroup}")
private String producerGroup;
@Bean
public DefaultMQProducer defaultMQProducer() throws MQClientException {
DefaultMQProducer producer = new DefaultMQProducer(producerGroup);
```

```
producer.setNamesrvAddr(namesrvAddr);
producer.setVipChannelEnabled(false);
producer.setRetryTimesWhenSendAsyncFailed(0);
producer.start();
logger.info("rocketmq producer is starting");
return producer;
}
}
```

引用 DefaultMQProducer，发送消息：

```
@Service
public class RocketmqService {
private static final Logger logger = LoggerFactory.getLogger(RocketmqService.class);
@Autowired
private DefaultMQProducer defaultMQProducer;
// 使用 application.properties 里定义的 topic 属性
@Value("${apache.rocketmq.topic}")
private String springTopic;
private AtomicLong id =new AtomicLong(0);
@Scheduled(fixedDelayString = "${consumer.auto.test.interval:5000}")
public String prepareSend() throws InterruptedException, RemotingException, MQClientException, MQBrokerException {
return send();
}
public String send() throws InterruptedException, RemotingException, MQClientException, MQBrokerException {
String sentText = "rocketmq message: msg id="+id.addAndGet(1);
Message message = new Message(springTopic,"push", sentText.getBytes());
message.putUserProperty("traceID","1234567");
SendResult sendResult = defaultMQProducer.send(message);
logger.info("消息id:"+id.get()+","+"发送状态:"+sendResult.getSendStatus());
return sendResult.getMsgId();
}
}
```

消费者示例

```
@Configuration
public class ConsumerConfiguration {
private static final Logger logger = LoggerFactory.getLogger(ConsumerConfiguration.class);
@Value("${apache.rocketmq.name-server}")
private String namesrvAddr;
@Value("${apache.rocketmq.topic}")
private String springTopic;
@Value("${apache.rocketmq.consumer.pushConsumer}")
private String consumerGroup;
@Bean
public DefaultMQPushConsumer pushConsumer() {
DefaultMQPushConsumer consumer = new DefaultMQPushConsumer(consumerGroup);
consumer.setNamesrvAddr(namesrvAddr);
try {
// 订阅PushTopic下Tag为push的消息,都订阅消息
consumer.subscribe(springTopic, "push");
// 程序第一次启动从消息队列头获取数据
```

```
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
//可以修改每次消费消息的数量，默认设置是每次消费一条
consumer.setConsumeMessageBatchMaxSize(1);
//在此监听中消费信息，并返回消费的状态信息
consumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
// 会把不同的消息分别放置到不同的队列中
for (Message msg : msgs) {
System.out.println("接收到了消息：" + new String(msg.getBody()));
logger.info("接收到了消息：" + new String(msg.getBody()));
try {
Thread.sleep(50);
} catch (InterruptedException e) {
e.printStackTrace();
}
}
return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
consumer.start();
} catch (Exception e) {
e.printStackTrace();
}
return consumer;
}
}
```

调用链快速入门

最近更新时间: 2025-02-18 16:02:00

为了节约用户的开发成本和提升使用效率，TSF 提供了 Spring Cloud 全链路跟踪的组件。用户只需要在代码中配置组件依赖，即可直接使用 TSF 的全链路跟踪功能，无需关心日志采集、分析、存储等过程。 用户仅需要安装了对应的依赖包及添加依赖项即可，无须其他配置。

准备工作

开始实践调用链功能前，请确保已完成 【SDK 下载】 。

依赖项

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】 。

微服务对下游组件访问的链路支持

微服务对 Redis、MySQL、MongoDB、消息队列 CMQ 的访问操作会产生跟踪日志， TSF 会对该日志进行采集、分析、统计，这些组件的调用会展现在 TSF 平台的链路追踪中。具体使用说明如下：

- Redis 链路使用说明
- MySQL 链路使用说明
- MongoDB 链路使用说明
- 消息队列 CMQ 链路使用说明
- Kafka 链路追踪使用说明
- RocketMQ 链路追踪使用说明

> **注意：**
>
> 非微服务组件的链路支持功能依赖于1.14.0版本及以上版本的 SDK，详情请参考 【SDK 更新日志】 。

标签与自定义元数据

调用链支持用户在代码中设置标签（Tag） 和自定义元数据（CustomMetada），分别用于调用链的筛选和附带业务信息。

接口定义：

```
public class TsfContext {
/**
* 设置多个 Tag。如果有某个 Tag 之前已经被设置过，那么它的值会被覆盖。
```

```
*/
public static void putTags(Map<String, String> tagMap, TagControlFlag... flags) {}

/**
* 设置 Tag。如果该 key 之前已经被设置过，那么它的值会被覆盖。
*/
public static void putTag(String key, String value, TagControlFlag... flags) {}

/**
* 设置 CustomMetadata。
*/
public static void putCustomMetadata(Object customMetadata)
}
```

注入和获取 trace 信息

TSF SDK1.23 及之后版本，支持注入和获取 traceId、spanId、tag 信息。

> **注意：**
>
> - traceId、spanId 格式不建议自定义，可能有未知错误。
> - TSF 不保证注入后调用链的正确性。

E 版本 SDK 使用方式

```
private static final Logger LOG = LoggerFactory.getLogger(MethodHandles.lookup().lookupClass());
@Autowired
private Tracer tracer;

public void addSpan() throw Exception{
//获取 traceId、spanId、tag
Span oldSpan = tracer.getCurrentSpan();
String traceId = oldSpan.getTraceId();
String spanId = oldSpan.getSpanId();
Map<String, String> tags = oldSpan.tags();
LOG.info("traceId: {}, spanId: {}", traceId, spanId);
for (String key : tags.keySet()) {
LOG.info("key: {}, value: {}", key, tags.get(key));
}

//设置 tag
tracer.addTag("http.method","get");
Map<String, String> tags1 = tracer.getCurrentSpan().tags();
for (String key : tags1.keySet()) {
LOG.info("key: {}, value: {}", key, tags1.get(key));
}

//设置 traceId、spanId（必须先新建 span）
Span.SpanBuilder spanBuilder = Span.builder();
Span span = spanBuilder.name("mockName").traceId(UUID.randomUUID().toString()).spanId(UUID.randomUUID().toString
()).tag("key", "value").build();

tracer.continueSpan(span);
```

```
Span newSpan = tracer.getCurrentSpan();
String newTraceId = newSpan.getTraceId();
String newSpanId = newSpan.getSpanId();

LOG.info("traceId: {}, spanId: {}", newTraceId, newSpanId);
}
```

F、G 版本 SDK 使用方式

```
private static final Logger LOG = LoggerFactory.getLogger(MethodHandles.lookup().lookupClass());
@Autowired
private Tracing tracing;

public void addSpan() throw Exception{
TraceContext traceContext = tracing.tracer().currentSpan().context();

TraceContext context = traceContext.toBuilder().traceId(2035873338086630653L).spanId(-2035873338086630653L).build();

CurrentTraceContext currentTraceContext = tracing.currentTraceContext();
currentTraceContext.newScope(context);

Span spanCur = tracing.tracer().currentSpan();


long traceId1 = spanCur.context().traceId();
long spanId1 = spanCur.context().spanId();
LOG.info("traceId1:{},spanId1:{}",traceId1, spanId1);


//添加tag
spanCur.tag("qx-test1","value");
spanCur.tag("qx-test2","value");
spanCur.tag("qx-test1","value");

//获取tag
Field state1 = spanCur.getClass().getDeclaredField("state");
state1.setAccessible(true);
MutableSpan mutableSpan1 = (MutableSpan)state1.get(spanCur);
Field tags1 = mutableSpan1.getClass().getDeclaredField("tags");
tags1.setAccessible(true);
ArrayList<String> list1 = (ArrayList<String>) tags1.get(mutableSpan1);
String key1 =null, value1 =null;
for (int i = 0, length = list1.size(); i < length; i += 2) {
key1 = list1.get(i);
value1 = list1.get(i + 1);

LOG.info("tags key1: {}, value1: {}", key1, value1);
}

//获取 traceid、spanid
TraceContext traceContext1 = spanCur.context();
long traceId2 = traceContext1.traceId();
long spanId2 = traceContext1.spanId();
LOG.info("traceId1:{},spanId1:{}",traceId2, spanId2);
}
```

Kafka 链路追踪

最近更新时间: 2025-02-18 16:02:00

Kafka 的链路追踪能力通过 Spring Boot 的自动配置方式实现。

1.16.0-Edgware 版本说明

实现方式是向 bean 容器中各加入了一个 producerFactory 和 consumerFactory 类的 bean。他们各自分别重写了 createKafkaProducer 方法和 createKafkaConsumer 方法。使用该 producerFactory/consumerFactory 便具有了链路追踪的能力。建议使用 Spring 默认的 KafkaTemplate，无须自定义。

1.16.0-Edgware 版使用的 spring-kafka 版本必须在**1.3.0**以上，因为从这个版本开始 kafka-client 中增加了请求头，这才能方便的实现链路追踪。

```
<!--支持kafka使用调用链-->
<dependency>
<groupId>org.springframework.kafka</groupId>
<artifactId>spring-kafka</artifactId>
<version>1.3.0.RELEASE</version>
</dependency>
```

参考配置：

```
#=============== kafka ====================
# 指定 kafka 代理地址，可以多个
spring.kafka.bootstrap-servers=

#=============== provider =======================
spring.kafka.producer.retries=0
# 每次批量发送消息的数量
spring.kafka.producer.batch-size=16384
spring.kafka.producer.buffer-memory=33554432

# 指定消息 key 和消息体的编解码方式
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.apache.kafka.common.serialization.StringSerializer

#=============== consumer =======================
# 指定默认消费者 Group ID
spring.kafka.consumer.group-id=test-consumer-group

spring.kafka.consumer.auto-offset-reset=earliest
spring.kafka.consumer.enable-auto-commit=true
spring.kafka.consumer.auto-commit-interval=100

# 指定消息 key 和消息体的编解码方式
spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
spring.kafka.consumer.value-deserializer=org.apache.kafka.common.serialization.StringDeserializer
```

其他更详细配置、使用方法详见 Demo 示例。

1.16.0-Finchley 版本说明

实现方式是向 Spring 容器中各增加了一个对切面，分别作用于 ProducerFactory.createProducer 方法和 ConsumerFactory.createConsumer 方法的，使得通过他们返回的 KafkaProducer 和 KafkaConsumer 具有链路追踪的能力。

使用 1.16.0-Finchley 版本**无须**考虑 spring-kafka 版本问题，无须指定它的版本，因为底层依赖的 Finchley 版本的 spring cloud 会自动为它配置高于 1.3.0.RELEASE 版本的spring-kafka。

```
<!--支持kafka使用调用链-->
<dependency>
<groupId>org.springframework.kafka</groupId>
<artifactId>spring-kafka</artifactId>
</dependency>
```

参考配置：

```
server:
servlet:
context-path: /
port: xxxxx
spring:
application:
name: xxxxxx
kafka:
bootstrap-servers: xxxxxx
#生产者的配置，大部分我们可以使用默认的，这里列出几个比较重要的属性
producer:
#每批次发送消息的数量
batch-size: 16
#设置大于0的值将使客户端重新发送任何数据，一旦这些数据发送失败。注意，这些重试与客户端接收到发送错误时的重试没有什么不同。允许重试将潜在的改变数据的顺序，如果这两个消息记录都是发送到同一个partition，则第一个消息失败第二个发送成功，则第二条消息会比第一条消息出现要早。
retries: 0
#producer可以用来缓存数据的内存大小。如果数据产生速度大于向broker发送的速度，producer会阻塞或者抛出异常，以"block.on.buffer.full"来表明。这项设置将和producer能够使用的总内存相关，但并不是一个硬性的限制，因为不是producer使用的所有内存都是用于缓存。一些额外的内存会用于压缩（如果引入压缩机制），同样还有一些用于维护请求。
buffer-memory: 33554432
#key序列化方式
key-serializer: org.apache.kafka.common.serialization.StringSerializer
value-serializer: org.apache.kafka.common.serialization.StringSerializer
#消费者的配置
consumer:
#Kafka中没有初始偏移或如果当前偏移在服务器上不再存在时,默认区最新 ，有三个选项 【latest, earliest, none】
auto-offset-reset: latest
#是否开启自动提交
enable-auto-commit: true
#自动提交的时间间隔
auto-commit-interval: 100
#key的解码方式
key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
#value的解码方式
value-deserializer: org.apache.kafka.common.serialization.StringDeserializer
#在/usr/local/etc/kafka/consumer.properties中有配置
group-id: test-consumer-group
tsf:
```

```
swagger:
enabled: false
logging:
file: /tsf-demo-logs/${spring.application.name}/root.log
```

其他更详细配置、使用方法详见 Demo 示例。

轻量级服务注册中心

最近更新时间: 2025-02-18 16:02:00

轻量级服务注册中心给开发者提供在开发、调试、测试的过程中的服务发现、注册和查询功能。

在一个公司内部，通常只需要在一台机器上安装轻量级服务注册中心。具体安装和使用的步骤请参见下文。

下载轻量级服务注册中心

推荐您找一台专门的机器启动轻量服务注册中心，例如某台测试机器。根据是否涉及到多个微服务联调测试，分为单机调试和多微服务联调两种场景进行说明。

本地使用轻量服务注册中心

如果**不涉及**到多个微服务联调场景，可以通过本地机器启动一个 Consul 作为轻量服务注册中心。单机调试支持 Windows 和 Linux / macOS 操作系统，多操作系统版本 Consul 下载地址。

确保机器以下的端口是空闲的：`8300`，`8301`，`8302`，`8500`，`8600`。

> **注意：**
>
> 用户可通过执行 `netstat -apn|grep LISTEN` 命令查看端口占用信息。

启动轻量级服务注册中心

本地使用轻量服务注册中心场景下支持 Windows 和 Linux / macOS 操作系统。 进入解压目录，启动服务注册中心。

- Windows 操作系统：

```
.\consul.exe agent -dev
```

- Linux / macOS 操作系统：

```
./consul agent -dev
```

验证服务注册中心启动

- 查看 8301 和 8500 的端口是否被监听；
- 通过浏览器查看服务注册中心页面（ `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500` ）。

多微服务联调环境的轻量服务注册中心

在多个微服务联调场景下，可以找一台可以被微服务访问的机器来部署轻量服务注册中心。目前本场景下仅支持 Linux 系统的 Consul，Linux 系统 Consul 下载地址。

启动轻量级服务注册中心

1. 将 consul 二进制文件放到任意一个目录，例如 /data/。
2. 将 start.sh 也放到同一个目录。下载脚本 start.sh。
3. 执行如下命令：

```
chmod +x start.sh; sudo ./start.sh local_ip
```

其中 `local_ip` 填写本机 IP。例如，Linux 机器上的 IP 为 192.168.1.10，那么执行的命令是： `sudo ./start.sh 192.168.1.10` 。

验证服务注册中心启动

```
./consul members -http-addr=127.0.0.1:8500
curl http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500/v1/catalog/services
```

如果有正常输出，代表 Consul 已经正常启动。

配置管理

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过轻量级服务注册中心或者 TSF 控制台下发动态配置。

前提条件

开始实践分布式配置功能前，请确保已完成了 [SDK 下载] 。

添加依赖

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本 。

修改配置

用户可通过两种方式更新代码中的配置信息：使用配置类 `@ConfigurationProperties` 和 `@Value` 注解。

- `@Value` 比较适用于配置比较少的场景
- `@ConfigurationProperties` 则更适用于有较多配置的情况

用户也可以动态更新应用配置文件（如 application.yml）中的配置，如动态更改 redis 的地址或者鉴权功能开关等。

使用配置类 @ConfigurationProperties

在 `provider-demo` 的 ProviderNameConfig 类中，有一个字符串类型的变量 `name` 。其中：

- 使用 `@ConfigurationProperties` 注解来标明这个类是一个配置类。
- 使用 `@RefreshScope注解` 开启 refresh 机制。

```
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.cloud.context.config.annotation.RefreshScope;
import org.springframework.stereotype.Component;</dx-codeblock>

@Component
@RefreshScope
@ConfigurationProperties(prefix="provider.config")
public class ProviderNameConfig {
private String name = "echo-provider-default-name";
```

```
public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}
```

使用 @Value 注解

在启动类 `ProviderApplication` 中，使用 `@Value` 注解来标识一个配置变量。下面的示例中 `${test.demo.prop:default}` 中 `test.demo.prop` 是在动态配置下发中使用的 key，value 默认是 `default` 。

- 使用 `@RefreshScope注解` 开启 refresh 机制。

```
// 其他 import
import org.springframework.web.bind.annotation.RestController;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;</dx-codeblock>

@RefreshScope
public class ProviderApplication {
public static void main(String\[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
```

```
@Value("${test.demo.prop:default}")
private String propFromValue;

@RequestMapping("/hello/") public String index() {
String result = "propFromValue: " + propFromValue + "\n";
return result;
}
```

配置更新触发回调

配置更新触发回调功能允许程序在不重启的情况下动态修改业务逻辑。当配置更新时，触发配置回调方法的调用。配置更新触发回调功能的使用场景包括：

- 程序使用一个防刷开关配置，当开关为启用状态时，启动防刷逻辑，当开关为关闭状态时，停用防刷逻辑。
- 程序使用一个 `ReidsConfig` 的动态配置类，包含 redis 的 host 和 port，当更新这个配置时，更新 Redis 实例。
- 配置更新后发出通知消息，通知本地或者远程的其他模块执行变更逻辑。

TSF 分布式配置支持使用 `@ConfigChangeListener` 注解且实现 `ConfigChangeCallback` 接口。在 `SimpleConfigurationListener` 类中，设置 `callback` 回调方法。支持场景包括：

- 支持按照配置项前缀模糊匹配方式。
- 支持配置项精确匹配方式。
- 支持回调方法同步 sync 或者异步 async 方式执行。

```
// prefix： 配置信息前缀，前缀匹配
// async: 回调事件是否异步执行，默认 false
// value： 精确匹配配置项
// 监听前缀为 io 的配置项变更
@ConfigChangeListener(prefix = "io",async = true)
public class SimpleConfigurationListener implements ConfigChangeCallback {
@Autowired
private SimpleService simpleService;
@Override
public void callback(ConfigProperty lastConfigItem, ConfigProperty currentConfigItem) {
simpleService.saySomething("receive Consul Config Change Event >>>> ");
simpleService.saySomething("Last config: [" + currentConfigItem.getKey() + ":" +lastConfigItem.getValue() + "]");
simpleService.saySomething("Current config: [" + currentConfigItem.getKey() + ":" +currentConfigItem.getValue() + "]");
}
}
@Component
public class SimpleService {
public void saySomething(String words){
System.out.println(words);
}
}
```

轻量级服务注册中心下发动态配置

用户可以使用轻量级服务注册中心下发动态配置，下面以 provider-demo 为例说明更新的具体步骤：

前提条件

1. 已完成轻量级服务注册中心的安装和启动 。
2. 服务已连接到轻量级服务注册中心。

操作步骤

1. 浏览器访问 **consul 控制台**（http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500）。

> **注意：**
>
> 只有成功启动了 consul 后，才能访问 consul 控制台。

2. 单击菜单栏【Key/Value】。
3. 单击【Create】，在配置新建界面填写 key 和 value。

- key 是 /config/application/data 或 /config/application/{spring.profiles.active}/data（适用于原生 Spring Cloud ）
- value 是配置的内容，截图中使用 provider-demo 中的自定义配置 provider.config.name=testname123。

4. 单击【Save】，观察 stdout 中是否出现 [TSF SDK] Configuration Change Listener: key: {}, old value: null, new value: testname123 ，如果出现说明配置生效。

TSF 控制台下发动态配置

用户可以通过 TSF 控制台来下发动态配置。

前提条件

- 已经在 TSF 平台上部署了 provider-demo 和 consumer-demo 应用。
- 部署 provider-demo 的部署组的日志配置项的日志路径中包含了 /tsf-demo-logs/provider-demo/root.log ，以确保打印的日志被采集后，可以通过控制台查看应用的日志。

操作步骤

关于如何通过控制台创建及下发更新的配置 。

如果希望修改 ProviderNameConfig 类中的 providerName 的值，创建配置时，配置内容填写：

```
provider:
config:
name: testname123
```

将配置发布到已部署 provider-demo 的部署组上，检查打印的日志中是否 name 的值已更新。如果已更新，说明更新的配置生效。

```
provider-demo -- provider config name: testname123
```

API 注册

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 框架在微服务注册时，会自动收集并注册微服务提供的 API 接口，用户可通过 TSF 控制台实时掌握当前微服务提供的 API 情况。API 注册功能基于 [OpenApi Specification 3.0] 规范注册 API 元数据信息。 用户在查看 API 接口的同时，可查看到 API 出入参数据结构信息。

前提条件

开始实践 API 注册功能前，请确保已完成了 【SDK 下载】。

添加依赖

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解 。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】 。

配置选项

API 注册功能基于 Swagger 原生规范实现，提供多个配置以适配 Swagger 不同配置应用场景，可用配置如下表所示：

配置项	类型	必填	默认值	说明
tsf.swagger.enabled	Boolean	否	true	是否开启 TSF API 注册功能
tsf.swagger.basePackage	String	否	ApplicationMainClass 所在包路径	注册 API 的扫描包路径。 推荐将 ApplicationMainClass 写在外层 Package
tsf.swagger.excludePath	String	否	（空）	排除扫描的包路径
tsf.swagger.group	String	否	default	swagger docket 分组

代码和示例

- SDK 会自动扫描 API 的 path 和 出入参。
- 如果需要上报 API 的描述，需要 import io.swagger.annotations.ApiOperation; ，同时在 API 上加上注解 `@ApiOperation(value = &quot;url路径值&quot;,notes = &quot;对api资源的描述&quot;)` 。如果不关注 API 描述，可以不设置 @ApiOperation。

```
package com.tsf.demo.provider.controller;
// 省略掉部分 import
import io.swagger.annotations.ApiOperation;
import com.tsf.demo.provider.config.ProviderNameConfig;

@RestController
public class ProviderController {
private static final Logger LOG = LoggerFactory.getLogger(ProviderController.class);

@Autowired
private ProviderNameConfig providerNameConfig;
@ApiOperation(value= "/echo/{param}", notes = "notes") // notes 对应 API 描述
@RequestMapping(value = "/echo/{param}", method = RequestMethod.GET)
public String echo(@PathVariable String param) {
LOG.info("provider-demo -- request param: [" + param + "]");
String result = "request param: " + param + ", response from " + providerNameConfig.getName();
LOG.info("provider-demo -- provider config name: [" + providerNameConfig.getName() + ']');
LOG.info("provider-demo -- response info: [" + result + "]");
return result;
}
}
```

Demo工程概述

最近更新时间: 2025-02-18 16:02:00

开发前准备

开发前，请确保：

1. 已下载安装了 Java 和 Maven
2. 已配置了 TSF 私服地址

获取 Demo

Demo 下载地址 >>

- release/<版本号>：对应 Spring Cloud Edgware 系列的 Demo
- release/<版本号>-finchley：对应 Spring Cloud Finchley 系列的 Demo
- release/<版本号>-greenwich：对应 Spring Cloud Greenwich 系列的 Demo

Demo 工程目录

`tsf-simple-demo` 的工程目录如下：

工程名称	工程说明
consumer-demo	TSF 微服务治理服务消费者
provider-demo	TSF 微服务治理服务提供者
msgw-demo	基于 TSF Spring Cloud MS Gateway 网关示例
opensource-zuul-demo	基于开源 Zuul 的微服务网关示例
opensource-scg-demo	基于开源 Spring Cloud Gateway 的微服务网关示例
rocketmq-producer	支持 RocketMQ 消息队列调用链的消息生产者示例
rocketmq-consumer	支持 RocketMQ 消息队列调用链的消息消费者示例
kafka-demo	支持 Kafka 调用链的示例，包含了消息消费者和生产者
mongodb-demo	支持 MongoDB 调用链的微服务示例
mysql-demo	支持 MySQL 调用链的微服务示例
redis-demo	支持 Redis 调用链的微服务示例

pom.xml 中定义了工程需要的依赖包（以下以基于 Spring Cloud Finchley 版本 SDK 举例说明）：

```
<project xmlns="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0" xmlns:xsi="http://imgcach
e.finance.cloud.tencent.com:80www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://imgcache.finance.cloud.tencent.com:80maven.apache.org/POM/4.0.0 http://imgcache.finance.
cloud.tencent.com:80maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>



<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-dependencies</artifactId>
<version><!-- 调整为 SDK 长期维护（LTS）版本号 --></version>
</parent>



<groupId>com.tencent.tsf</groupId>
<artifactId>tsf-demo</artifactId>
<version><!-- 调整为 SDK 长期维护（LTS）版本号 --></version>
<packaging>pom</packaging>



<modules>
<module>provider-demo</module>
<module>consumer-demo</module>
<module>opensource-zuul-demo</module>
<module>rocketmq-demo</module>
<module>mysql-demo</module>
<module>redis-demo</module>
<module>mongodb-demo</module>
<module>kafka-demo</module>
<module>msgw-demo</module>
<module>opensource-scg-demo</module>
</modules>



<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>



<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
```

```
</project>
```

其中 parent 描述了不同微服务 demo 共同的 TSF 依赖。

```
<parent>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-dependencies</artifactId>
<version><!-- 调整为 SDK 长期维护（LTS）版本号 --></version>
</parent>
```

依赖项及注解使用

1. 向工程中添加依赖。在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 长期维护（LTS）版本号 --></version>
</dependency>
```

`spring-cloud-tsf-starter` 中包含了服务注册发现、服务路由、服务鉴权、服务限流、服务熔断、服务容错、服务监控、分布式配置、调用链功能。

2. 向 Application 类中添加注解 `@EnableTsf` ：

```
// 下面省略了无关的代码
import org.springframework.tsf.annotation.EnableTsf;
@SpringBootApplication
@EnableTsf
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】。

Spring Cloud TSF SDK 历史版本使用方法

最近更新时间: 2025-02-18 16:02:00

本文档仅适用于使用 `1.15.0-Edgware-RELEASE` / `1.15.0-Finchley-RELEASE` 和之前版本的 SDK。参考使用 `1.14.2-Finchley-RELEASE` 的 Demo 了解全部依赖项及注解使用方法。

服务鉴权

1. 向 provider 和 consumer 工程都添加依赖。在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-auth</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsfAuth` ：

```
// 下面省略了无关的代码
import org.springframework.tsf.auth.annotation.EnableTsfAuth;
@SpringBootApplication
@EnableTsfAuth
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

服务限流

如果用户要对某个服务开启限流的能力，即对调用它的请求做限流，可以按下面的步骤打开限流开关。

1. 向工程中添加依赖。在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-ratelimit</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsfRateLimit` ：

```
// 下面省略了无关的代码
import org.springframework.tsf.ratelimit.annotation.EnableTsfRateLimit;
@SpringBootApplication
@EnableTsfRateLimit
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

服务路由

1. 向工程中添加依赖。在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-route</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsfRoute` ：

```
// 下面省略了无关的代码
import org.springframework.cloud.tsf.route.annotation.EnableTsfRoute;
@SpringBootApplication
@EnableTsfRoute
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

分布式配置

添加 pom.xml 依赖：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-consul-config</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
<!-- 1.12.0之前（不包含1.12.0）版本 SDK，使用分布式配置自动刷新功能时，要添加 actuator 的依赖包
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
-->
```

调用链

添加 pom.xml 依赖：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-sleuth</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

服务监控

Edgware 版本 SDK 相关设置

向工程中添加依赖。在 pom.xml 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-monitor</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

Finchley 版本 SDK 相关设置

1. 向工程中添加依赖。在 pom.xml 中添加以下代码，**依赖的是 spring-cloud-tsf-sleuth **而不是 spring-cloud-tsf-monitor 。

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-sleuth</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 @EnableTsfMonitor ：

```
// 下面省略了无关的代码
import com.tencent.tsf.monitor.annotation.EnableTsfMonitor;
@SpringBootApplication
@EnableTsfMonitor
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

API 注册

在 pom.xml 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-swagger</artifactId>
<version><!-- 调整为历史版本 SDK 版本号 --></version>
<scope>compile</scope>
</dependency>
```

添加依赖包后，TSF API 注册功能即生效。

Spring Cloud 应用概述

最近更新时间: 2025-02-18 16:02:00

TSF 支持原生 Spring Cloud 微服务框架，开发者只需要添加依赖和修改配置即可使用服务注册、调用链、分布式配置等能力。

兼容性说明

TSF 兼容主流 SDK 版本（Edgware、Finchley、Greenwich）。Spring Cloud 功能、开源实现及 TSF 兼容性如下表所示：

Spring Cloud 功能	开源实现	TSF 兼容性	说明
服务注册与发现	Netflix Eureka Consul	基于开源增强	提供金融级高可用注册中心，无须用户自行搭建
负载均衡	Netflix Ribbon	兼容	-
服务调用	RestTemplate/AsyncRestTemplate Feign	兼容	-
调用链	Spring Cloud Sleuth	基于开源增强	提供服务依赖拓扑、调用链查询基础功能，同时支持调用链与业务日志联动、调用链支持下游组件等高级特性
分布式配置	Spring Cloud Config Consul Config	基于开源增强	支持通过控制台管理配置，发布配置和查看配置发布历史
消息驱动	Kafka	兼容	提供调用链传递到消息队列 CMQ、Ckafka、开源 Kafka
安全	Spring Cloud Security	兼容	-
微服务网关	Spring Cloud Gateway NetflixZuul	兼容 Zuul、Spring Cloud Gateway	-
熔断降级	Spring Cloud Hystrix	自研	TSF 采用官方推荐的 Resilience4J 作为底层实现，扩展支持实例、API 和服务级别的熔断

准备工作

最近更新时间: 2025-02-18 16:02:00

为了更好地将中间件的功能以云服务的方式提供给大家，TSF 对 Spring Cloud 的一些组件进行了替换，增强了安全性和易用性。

获取 Demo

Demo 源码下载 >>

获取 SDK

1. 环境准备

在执行安装脚本之前，需要确保机器上已经安装了 Maven 和设置了环境变量。

1.1 安装 Maven 。

1.2 当 Maven 安装完成后， 通过执行如下命令验证 Maven 是否安装成功。

- Windows 系统下

```
mvn --version
```

- Linux、macOS 系统下

```
mvn -v
```

若出现正常的版本号信息后，说明 Maven 安装成功。

1.3 设置环境变量

- Windows 系统下

```
新建系统变量 MAVEN_HOME 变量值：E:\Maven\apache-maven-3.3.9
编辑系统变量 Path 添加变量值： ;%MAVEN_HOME%\bin
```

- Linux、macOS 系统下

```
export MAVEN_HOME=/usr/local/maven/apache-maven-3.3.9
export PATH=MAVEN_HOME/bin:$PATH
```

1.4 设置 JAVA_HOME 环境变量

- Windows 系统下

```
JAVA_HOME=C:/Program Files/Java/jdk1.5.0_05
```

- Linux、macOS 系统下

```
export JAVA_HOME=$(/usr/libexec/java_home)
```

2. 安装 SDK

TSF 提供两种安装 SDK 到本地 Maven 仓库的方式：从远程仓库下载和使用本地安装脚本。

2.1 从远程仓库下载 SDK 到本地仓库

参考 [《Maven 配置 TSF 仓库地址》]，在 Demo 工程 `pom.xml` 所在目录执行 `mvn clean package` 即可下载 TSF SDK。

> **注意：**
>
> 注意：仓库中提供的 TSF SDK 的依赖版本号从 1.1.1.TSF-RELEASE 开始。

2.2 脚本安装 SDK 到本地仓库

SDK（1.1.1.TSF-RELEASE）下载 >>

解压后的 SDK 下载包中提供了 Windows 和 Linux 系统下的安装脚本。 执行脚本，脚本将依赖库安装到本地 Maven 仓库中。

- Windows 系统下，鼠标双击 `tsf_install_mvn.bat` 文件。
- Linux 、 macOS 系统下，在命令行下执行 `.\tsf_install_mvn.sh` 文件。

若脚本执行成功，出现如下界面。

参数传递

最近更新时间: 2025-02-18 16:02:00

元数据类别

TSF 提供两种类型的元数据供开发者在代码中进行设置：

类型	特点
标签信息（Tags）	可设置传递性，仅支持 key-value 数据结构，key 和 value 均为字符串类型。
辅助信息（CustomMetadata）	仅供展示，不支持筛选，不具备传递性。

场景说明：

- **标签信息**：用于信息分类，使用场景包括：
- 服务鉴权：被调方通过标签来决定是否提供服务。
- 服务路由：通过标签来判断应该访问什么服务，可用于实现金丝雀发布等。
- 调用链：可用于调用链的筛选和附带业务信息。
- **辅助信息**（CustomMetadata）：仅供展示，不支持筛选，不具备传递性。

元数据的传递性

以调用关系 A ≥ B ≥ C，说明传递性的概念：

- 可传递（Transitive）：需要传递的标签，在整条链路都传递，即用户在 A 设置的标签，会传递到 B 再传递到 C。
- 不可传递：不需要传递的标签，即用户在 A 设置的标签，会传递到 B，但是不会传递到 C。

标签信息允许用户设置是否支持传递，**辅助信息**不支持传递。 不同的标签可以设置不同传递性，例如一些业务场景：

- `userid` 标签是需要传递的：
- 可以作为整条调用链上的服务的 上下文信息。
- 可以实现按 uin 区分的服务路由，例如 A、B、C 三个服务同时做滚动发布，那么可以让一批 uin 都走新版本的 A、B、C 服务，其他用户走老版本。
- `level=高级会员` 这种标签，很可能就不需要在调用间传递下去。

使用元数据

依赖项

在 `pom.xml` 中添加依赖项：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-core</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

接口

```
public enum ControlFlag {
TRANSITIVE // 表示标签要传递下去，默认不启用
NOT_IN_AUTH // 表示标签不被使用在服务鉴权，默认是被使用的
NOT_IN_ROUTE // 表示标签不被使用在服务路由，默认是被使用的
NOT_IN_SLEUTH // 表示标签不被使用在调用链，默认是被使用的
}

public class TsfContext {
/**
* 设置多个 Tag。如果有某个 Tag 之前已经被设置过，那么它的值会被覆盖。
*/
public static void putTags(Map<String, String> tagMap, TagControlFlag... flags) {}

/**
* 设置 Tag。如果该 key 之前已经被设置过，那么它的值会被覆盖。
*/
public static void putTag(String key, String value, TagControlFlag... flags) {}
}
```

场景1：设置鉴权 Tag

TSF 提供的 Demo `consumer-demo/src/main/java/com/tsf/demo/consumer/Controller.java` 中设置了键为 user，请求参数作为值的 Tag。

```
@RequestMapping(value = "/echo-rest/{str}", method = RequestMethod.GET)
public String rest(@PathVariable String str, @RequestParam String user) {
TsfContext.putTag("user", user);
return restTemplate.getForObject("http://imgcache.finance.cloud.tencent.com:80provider-demo/echo/" + str, String.class);
}
```

场景2：设置调用链 Tag

设置 `user=12345678` 的标签，用户可以在控制台 **调用链查询界面** 通过标签 user=12345678 来检索调用链。

```
TsfContext.putTag("user", "12345678", TRANSITIVE);
```

Tag 的长度限制

用户传递到下流的 Tag（包含从上流带过来的有传递性的 Tag），数量上限为16个。Key 的长度上限为 UTF-8 编码后32字节，value 的长度上限为 UTF-8 编码后128字节。

服务容错

最近更新时间: 2025-02-18 16:02:00

操作场景

TSF 摒弃了已经不再继续维护的 Hystrix 作为容错模块，在原有单一的 fallback 上新增了 failfast、failover 和 forking。请配合 TSF 其他功能一起使用。

前提条件

开始实践服务容错功能前，请确保已完成了 【SDK 下载】 ，同时请确保 SDK 版本高于**1.19**。

操作步骤

> **注意：**
>
> 步骤1和步骤2与其他模块一样，已经使用过其他模块的可直接跳至步骤3。

1. 向工程中添加依赖。 在 `pom.xml` 中添加以下代码：

```
<dependency>
<groupId>com.tencent.tsf</groupId>
<artifactId>spring-cloud-tsf-starter</artifactId>
<version><!-- 调整为 SDK 最新版本号 --></version>
</dependency>
```

2. 向 Application 类中添加注解 `@EnableTsf` **：**

```
// 下面省略了无关的代码
@SpringBootApplication
@EnableTsf
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}
}
```

3. 使用 Feign 上的 fallback 以及 fallbackFactory 功能。 TSF 容错兼容 feign 的容错功能，如果需要使用 feign 的如下降级功能，则需要关闭 Hystrix 开关。

```
@FeignClient(name = "circuit-breaker-mock-service", fallbackFactory = HystrixClientFallbackFactory.class)
@FeignClient(name = "circuit-breaker-mock-service", fallback = FeignClientFallback.class)
```

关闭 Hystrix 开关（默认是关闭的，如果之前使用了该功能，可以删除该配置或者关闭）：

```
feign:
hystrix:
```

```
enabled: false
```

打开 TSF 开关：

```
feign:
tsf:
enabled: true
```

4. 在代码中使用容错功能。

```
// 下面省略了无关的代码
@TsfFaultTolerance(strategy = TsfFaultToleranceStragety.FAIL_OVER, parallelism = 2, fallbackMethod = "doWorkFallback")
public void doWork() throws InterruptedException {
String response = providerDemoService.echo("1234");
LOG.info("consumer-demo auto test, response: [" + response + "]");
}
public void doWorkFallback() {
System.out.println("fallback");
}
// fallbackMethod可以加也可以不加，用户可以自行选择
@TsfFaultTolerance(strategy = TsfFaultToleranceStragety.FAIL_FAST)
public void doWork2() throws InterruptedException {
String response = providerDemoService.echo2("1234");
LOG.info("consumer-demo auto test, response: [" + response + "]");
}
```

可以看到，TSF 的容错使用起来非常方便，您只需在需要保护的方法上增加一个注解即可（需要该 Bean 被 Spring 所管理）。

容错策略说明

failfast、failover 和 forking 容错策略是可选择的配置：

容错策略	含义	支持配置
failfast	直接失败，对于没有幂等性的下游服务推荐 failfast	无
failover	请求错了之后会重试	重试次数 maxAttempts
forking	同时发送多个请求，需要用户配置并行度，例如同时发出两个请求，哪个先返回，就把这个结果返回回去。如果第一个请求是异常，则会等另一个请求，如果全部都异常，则返回异常。	并行度 parallelism

三种容错策略均支持 fallback 方法，要求入参类型和返回值类型与原方法相同。您可以选择是否添加某一策略。

除了如上所述的基本功能外，TSF 还提供了选择容错异常的能力：

```
// 下面省略了无关的代码
@TsfFaultTolerance(strategy = TsfFaultToleranceStragety.FAIL_OVER, parallelism = 2, ignoreExceptions = {FeignException.class}, raisedExceptions = {RuntimeException.class, InterruptedException.class}, fallbackMethod = "doWorkFallback")
public void doWork() throws InterruptedException {
String response = providerDemoService.echo("1234");
LOG.info("consumer-demo auto test, response: [" + response + "]");
}

public void doWorkFallback() {
```

```
System.out.println("fallback");
}
```

- 用户如果设置了 ignoreExceptions，且当前异常是其中一个的子类的话，则不执行容错逻辑。
- 如果用户没有设置 ignoreExceptions，或当前异常不是 ignoreExceptions 的子类且满足如下条件则执行容错逻辑：
- 用户未设置 raisedExceptions，则执行容错逻辑。
- 用户设定了 raisedExceptions，且当前异常为用户设置的 raisedExceptions 其中一个的子类，则执行容错逻辑。

分布式配置

最近更新时间: 2025-02-18 16:02:00

操作场景

该任务指导您通过轻量级服务注册中心或者TSF控制台下发动态配置。

前提条件

开始实践分布式配置功能前，请确保已完成了 【SDK 下载】。

添加依赖

向工程中添加 `spring-cloud-tsf-starter` 依赖并开启 `@EnableTsf` 注解，详情请参考 [Demo工程概述] 文档。

> **注意：**
>
> 如果您使用的是 1.15.0-Edgware-RELEASE/1.15.0-Finchley-RELEASE 及之前的版本，使用方法参考 【Spring Cloud SDK 历史版本使用方法】。

修改配置

用户可通过两种方式更新代码中的配置信息：使用配置类 `@ConfigurationProperties` 和 `@Value` 注解。

- `@Value` 比较适用于配置比较少的场景
- `@ConfigurationProperties` 则更适用于有较多配置的情况

用户也可以动态更新应用配置文件（如 application.yml）中的配置，如动态更改 redis 的地址或者鉴权功能开关等。

使用配置类 @ConfigurationProperties

在 provider-demo 的 ProviderNameConfig 类中，有一个字符串类型的变量 `name`。其中：

- 使用 `@ConfigurationProperties` 注解来标明这个类是一个配置类。
- 使用 `@RefreshScope注解` 开启 refresh 机制。

```
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.cloud.context.config.annotation.RefreshScope;
import org.springframework.stereotype.Component;

@Component
@RefreshScope
@ConfigurationProperties(prefix="provider.config")
public class ProviderNameConfig {
private String name = "echo-provider-default-name";
```

```
public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}
}
```

使用 @Value 注解

在启动类 ProviderApplication 中，使用 @Value 注解来标识一个配置变量。下面的示例中 ${test.demo.prop:default} 中 test.demo.prop 是在动态配置下发中使用的 key，value 默认是 default 。

- 使用 @RefreshScope注解 开启 refresh 机制。

```
// 其他 import
import org.springframework.web.bind.annotation.RestController;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;

@RefreshScope
public class ProviderApplication {
public static void main(String[] args) {
SpringApplication.run(ProviderApplication.class, args);
}

@Value("${test.demo.prop:default}")
private String propFromValue;

@RequestMapping("/hello/") public String index() {
String result = "propFromValue: " + propFromValue + "\n";
return result;
}
}
```

配置更新触发回调

配置更新触发回调功能允许程序在不重启的情况下动态修改业务逻辑。当配置更新时，触发配置回调方法的调用。配置更新触发回调功能的使用场景包括：

- 程序使用一个防刷开关配置，当开关为启用状态时，启动防刷逻辑，当开关为关闭状态时，停用防刷逻辑。
- 程序使用一个 ReidsConfig 的动态配置类，包含 redis 的 host 和 port，当更新这个配置时，更新 Redis 实例。
- 配置更新后发出通知消息，通知本地或者远程的其他模块执行变更逻辑

TSF 分布式配置支持使用 @ConfigChangeListener 注解且实现 ConfigChangeCallback 接口。在 SimpleConfigurationListener 类中，设置 callback 回调方法。支持场景包括：

- 支持按照配置项前缀模糊匹配方式。
- 支持配置项精确匹配方式。
- 支持回调方法同步 sync 或者异步 async 方式执行。

```
// prefix： 配置信息前缀，前缀匹配
// async: 回调事件是否异步执行，默认 false
// value： 精确匹配配置项
// 监听前缀为 io 的配置项变更
@ConfigChangeListener(prefix = "io",async = true)
public class SimpleConfigurationListener implements ConfigChangeCallback {
@Autowired
private SimpleService simpleService;
@Override
public void callback(ConfigProperty lastConfigItem, ConfigProperty currentConfigItem) {
simpleService.saySomething("receive Consul Config Change Event >>>> ");
simpleService.saySomething("Last config: [" + currentConfigItem.getKey() + ":" +lastConfigItem.getValue() + "]");
simpleService.saySomething("Current config: [" + currentConfigItem.getKey() + ":" +currentConfigItem.getValue() + "]");
}
}
@Component
public class SimpleService {
public void saySomething(String words){
System.out.println(words);
}
}
```

轻量级服务注册中心下发动态配置

用户可以使用轻量级服务注册中心下发动态配置，下面以 provider-demo 为例说明更新的具体步骤：

前提条件：

1. 已完成轻量级服务注册中心的安装和启动（参考 【轻量级服务注册中心】）。
2. 服务已连接到轻量级服务注册中心。

操作步骤

3. 浏览器访问 **consul 控制台**（http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500）。

> **注意：**
>
> 只有成功启动了 consul 后，才能访问 consul 控制台。

4. 单击菜单栏【Key/Value】。
5. 单击【Create】，在配置新建界面填写 key 和 value。

- key 是 /config/application/data 或 /config/application/{spring.profiles.active}/data（适用于原生 Spring Cloud）
- value 是配置的内容，截图中使用 provider-demo 中的自定义配置 provider.config.name=testname123。

6. 单击【Save】，观察 stdout 中是否出现 `[TSF SDK] Configuration Change Listener: key: {}, old value: null, new value: testname123`，如果出现说明配置生效。

TSF控制台下发动态配置

用户可以通过 TSF 控制台来下发动态配置。

前提条件

- 已经在 TSF 平台上部署了 provider-demo 和 consumer-demo 应用。
- 部署 provider-demo 的部署组的日志配置项的日志路径中包含了 /tsf-demo-logs/provider-demo/root.log ，以确保打印的日志被采集后，可以通过控制台查看应用的日志。参考[日志配置项]。

操作步骤

关于如何通过控制台创建及下发更新的配置，请参考 【应用配置】。

如果希望修改 ProviderNameConfig 类中的 providerName 的值，创建配置时，配置内容填写：

```
provider:
config:
name: testname123
```

将配置发布到已部署 provider-demo 的部署组上，检查打印的日志中是否 name 的值已更新。如果已更新，说明更新的配置生效。

```
provider-demo -- provider config name: testname123
```

本地开发联调

最近更新时间: 2025-02-18 16:02:00

本文介绍了两种本地开发联调的使用场景：

- 本地服务之间调用
- 本地服务调用云端服务

本地服务之间调用

操作场景

开发者通过搭建本地轻量级注册中心，将本地服务注册到轻量级注册中心上，服务之间通过服务名来进行调用。

操作步骤

1. 启动轻量级注册中心

本地开发调试时，需要使用轻量级注册中心，轻量级注册中心包含了 TSF 服务注册发现服务端的轻量版，详请参见 [轻量级服务注册中心]。

2. 启动应用

本地启动应用可以通过 IDE 和 FatJar 两种方式。

IDE 中启动

在 IDE 中启动，通过 VM options 配置启动参数 `-Dtsf_consul_ip=127.0.0.1 -Dtsf_consul_port=8500 -Dtsf_application_id=a -Dtsf_group_id=b -Dtsf.swagger.enabled=false` ，通过 main 方法直接启动。其中 IP 和 port 取值为轻量级服务注册中心的地址，使用了分布式配置功能的模块，需要设置 `-Dtsf_application_id=a -Dtsf_group_id=b` ， 取值可为任意值。

FatJar 启动

（1）添加 FatJar 打包方式

```
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
```

（2）打包 FatJar 文件

添加完插件后，在工程的主目录下，使用 Maven 命令 `mvn clean package` 进行打包，即可在 target 目录下找到打包好的 FatJar 文件。

（3）通过 Java 命令启动

```
java -Dtsf_instance_id=1111 -Dtsf_consul_ip=127.0.0.1 -Dtsf_consul_port=8500 -Dtsf.swagger.enabled=false -jar provider-demo-0.0.1-SNAPSHOT.jar
```

其中 `127.0.0.1` 和8500为轻量级服务注册中心地址，在本地调试时 tsf_application_id 和 tsf_group_id 可以填任意值。

> **注意：**
>
> - Instance_id 是服务实例唯一标识，需要保证唯一性。。
> - 由于轻量级服务注册中心（原生的 consul）的 metadata 只能支持512个字节，因此需要关闭 TSF 的 API 上报能力 `-Dtsf.swagger.enabled=false` ，如果没有这个启动参数，在本地运行 Demo 时将会报错，错误信息中包含 `Value is too long (limit 512 characters)` 。

3. 验证

启动服务，分别进行调用，观察调用结果。

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/echo-rest/test?user=test-tsf`

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/echo-async-rest/test?user=test-tsf`

`http://imgcache.finance.cloud.tencent.com:80{consumer-demo-ip}:{consumer-demo-port}/test?user=test-tsf`

本地服务调用云端服务

操作场景

如果开发者需要使本地正在开发的微服务和远端的微服务做联调，不需在本地启动服务注册中心，可直接使用本文提供的轻量级联调操作方法。其中，远端的微服务指通过 TSF 部署的 Spring Cloud 应用。 假设本地开发环境的 consumer-demo 服务，希望调用 TSF 上的 provider-demo 服务提供的接口，provider-demo 服务有一个实例提供**外网 IP**，服务监听端口为 18081。

前提条件

- 依赖 1.11.0-RELEASE 及以上版本的 SDK。
- 确保本地开发机和 provider-demo 的实例的网络连通性，确保从外网可访问 provider-demo 的端口号。
- 本地未启动 consul 或者 consumer-demo 未连接本地 consul ，否则 consumer-demo 会通过本地 consul 进行服务注册与发现。

操作步骤

1. 在本地开发机器上，创建 `${System.getProperty(&quot;user.home&quot;)}/.tsf/discovery/` 目录，其中 ${System.getProperty("user.home")} 是本地开发机器的 home 目录。
2. 在该目录下创建服务缓存文件，缓存文件名称为 `${service-name}.cache` ，其中 `${service-name}` 是被调服务名，在本例中是 provider-demo.cache。 在 provider-demo.cache 中填写远端服务的描述信息：

```
{
"statusCode": 200, # statusCode 字段必填，取值为200，表示正常状态码；请勿更改；
"content": [{
"service": {
"id": "provider-demo-18081", # 随机字符，确保唯一性
"service": "provider-demo", # 被调服务的服务名
"tags": [],
"address": "<被调服务的实例IP>", # 被调服务实例的 IP
"port": 18081, # 被调服务监听端口
"meta": {
}
},
```

```
"checks": []
}]
}
```

实际使用过程中，开发者仅需关注 content.service 结构体中的内容（id、service、address、port），其他字段内容可以复用上述示例。

3. 启动 consumer-demo，启动命令中包含必要的 启动参数：

```
java -Dtsf_instance_id=1111 -Dspring.cloud.consul.config.fail-fast=false -Dspring.cloud.consul.discovery.register=false -Dspring.cloud.consul.discovery.fail-fast=false -Dspring.cloud.consul.discovery.catalogServicesWatch.enabled=false -Dspring.cloud.consul.config.watch.enabled=false -jar consumer-demo-0.0.1-SNAPSHOT.jar
```

> **注意：**
>
> Instanceid是服务实例唯一标识，需要保证唯一性。

4. consumer-demo 会轮询调用 provider-demo 的接口，如果控制台的日志显示正常，说明调用成功。

启动参数说明

参数	含义	默认值
spring.cloud.consul.discovery.register	是否开启 consul 服务注册能力。	true
spring.cloud.consul.discovery.fail-fast	是否开启服务注册发现快速失败能力。关闭后，如果发起服务注册失败，会继续进行服务启动。	true
spring.cloud.consul.discovery.catalogServicesWatch.enabled	是否开启服务注册中心的异常请求。关闭后，会减少到服务注册中心的异常请求。	true
spring.cloud.consul.config.watch.enabled	分布式配置长轮询监听定时任务。关闭后，会减少连接服务注册中心的异常日志输出。	true
spring.cloud.consul.config.fail-fast	是否开启分布式配置快速失败功能。关闭后，如果请求 consul 失败，则输出异常日志，继续进行服务启动。	true

轻量级服务注册中心

最近更新时间: 2025-02-18 16:02:00

轻量级服务注册中心给开发者提供在开发、调试、测试的过程中的服务发现、注册和查询功能。

在一个公司内部，通常只需要在一台机器上安装轻量级服务注册中心。具体安装和使用的步骤请参见下文。

下载轻量级服务注册中心

推荐您找一台专门的机器启动轻量服务注册中心，例如某台测试机器。根据是否涉及到多个微服务联调测试，分为单机调试和多微服务联调两种场景进行说明。

本地使用轻量服务注册中心

如果**不涉及**到多个微服务联调场景，可以通过本地机器启动一个 Consul 作为轻量服务注册中心。单机调试支持 Windows 和 Linux / macOS 操作系统，多操作系统版本 Consul 下载地址。

确保机器以下的端口是空闲的：`8300`，`8301`，`8302`，`8500`，`8600`。

> **注意：**
>
> 用户可通过执行 `netstat -apn|grep LISTEN` 命令查看端口占用信息。

启动轻量级服务注册中心

本地使用轻量服务注册中心场景下支持 Windows 和 Linux / macOS 操作系统。 进入解压目录，启动服务注册中心。

- Windows 操作系统：

```
.\consul.exe agent -dev
```

- Linux / macOS 操作系统：

```
./consul agent -dev
```

验证服务注册中心启动

- 查看 8301 和 8500 的端口是否被监听；
- 通过浏览器查看服务注册中心页面（ `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500` ）。

多微服务联调环境的轻量服务注册中心

在多个微服务联调场景下，可以找一台可以被微服务访问的机器来部署轻量服务注册中心。目前本场景下仅支持 Linux 系统的 Consul，Linux 系统 Consul 下载地址。

启动轻量级服务注册中心

1. 将 consul 二进制文件放到任意一个目录，例如 /data/。
2. 将 start.sh 也放到同一个目录。下载脚本 start.sh。
3. 执行如下命令：

```
chmod +x start.sh; sudo ./start.sh local_ip
```

其中 `local_ip` 填写本机 IP。例如，Linux 机器上的 IP 为 192.168.1.10，那么执行的命令是： `sudo ./start.sh 192.168.1.10` 。

验证服务注册中心启动

```
./consul members -http-addr=127.0.0.1:8500
curl http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8500/v1/catalog/services
```

如果有正常输出，代表 Consul 已经正常启动。

API文档
腾讯微服务平台 TSF（tsf）
版本（2018-03-26）
API概览

最近更新时间: 2025-02-18 17:50:35

API版本

V3

其他接口

接口名称	接口功能
CreateTaskFlow	创建工作流
DescribeBasicResourceUsage	TSF基本资源信息概览
DescribeContainerEvents	获取容器事件列表
DescribePodInstances	获取部署组实例列表
DownloadMultipartPkg	分片下载程序包
SearchContainerStdoutLog	查询容器标准输出日志
TerminateTaskFlowBatch	停止一个工作流批次

分布式任务调度相关接口

接口名称	接口功能
ContinueRunFailedTaskBatch	续跑任务批次
CreateTask	创建任务
DeleteTask	删除任务
DescribeFlowLastBatchState	查询工作流最近一个批次的执行状态
DescribeTaskDetail	查询任务详情
DescribeTaskLastStatus	查看任务最近执行批次状态
DisableTask	停用任务

接口名称	接口功能
DisableTaskFlow	停用工作流
EnableTask	启用任务
EnableTaskFlow	启用工作流
ExecuteTask	手动执行一次任务。
ExecuteTaskFlow	执行一次工作流
ModifyTask	修改任务
RedoTask	重新执行任务
RedoTaskBatch	重新执行任务批次
RedoTaskExecute	重新执行任务的一次执行
RedoTaskFlowBatch	重新执行工作流批次
StopTaskBatch	停止执行中的任务批次
StopTaskExecute	停止正在执行的任务

命名空间相关接口

接口名称	接口功能
CreateNamespace	创建命名空间
DeleteNamespace	删除命名空间
DescribeSimpleNamespaces	查询简单命名空间列表

应用相关接口

接口名称	接口功能
CreateApplication	创建应用
DeleteApplication	删除应用
DescribeApplication	获取应用详情
DescribeApplicationAttribute	获取应用列表其它字段
DescribeApplications	获取应用列表
DescribeSimpleApplications	查询简单应用列表

微服务网关相关接口

接口名称	接口功能
BindApiGroup	网关与API分组批量绑定
BindPlugin	批量绑定插件
ChangeApiUsableStatus	启用或禁用API
CreateAllGatewayApiAsync	一键导入API分组
CreateApiGroup	创建API分组
CreateApiRateLimitRule	创建API限流规则
CreateGatewayApi	批量导入API至api分组
CreatePathRewrites	创建路径重写
CreateUnitRule	创建单元化规则
CreateWildCardGatewayApi	创建通配符API
DeleteApiGroup	删除Api分组
DeletePathRewrites	删除路径重写
DeleteUnitNamespaces	删除单元化命名空间
DeleteUnitRule	删除单元化规则
DescribeApiGroup	查询API分组
DescribeApiGroups	查询API 分组信息列表
DescribeApiRateLimitRules	查询API限流规则
DescribeApiUseDetail	查询网关API监控明细数据
DescribeEnabledUnitRule	查询生效的单元化规则
DescribeGatewayAllGroupApis	查询网关所有分组下Api列表
DescribeGatewayMonitorOverview	查询网关监控概览
DescribeGroupBindedGateways	查询某个API分组已绑定的网关部署组信息列表
DescribeGroupGateways	查询某个网关绑定的API 分组信息列表
DescribeGroupUseDetail	查询网关分组监控明细数据
DescribeGroupsWithPlugin	查询某个插件下绑定或未绑定的API分组
DescribePathRewrite	查询路径重写
DescribePathRewrites	查询路径重写列表

接口名称	接口功能
DescribePluginInstances	查询网关分组或API绑定（或未绑定）的插件列表
DescribeUnitApiUseDetail	网关调用监控统计查询类
DescribeUnitNamespaces	查询单元化命名空间列表
DescribeUnitRule	查询单元化规则详情
DescribeUnitRules	查询单元化规则列表
DescribeUsableUnitNamespaces	查询可用于被导入的命名空间列表
DisableUnitRoute	禁用单元化路由
DisableUnitRule	禁用单元化规则
DraftApiGroup	下线Api分组
EnableUnitRoute	启用单元化路由
EnableUnitRule	启用单元化规则
ModifyPathRewrite	修改路径重写
ReleaseApiGroup	发布Api分组
UnbindApiGroup	API分组批量与网关解绑
UpdateApiGroup	更新Api分组
UpdateApiRateLimitRule	更新API限流规则
UpdateApiRateLimitRules	批量更新API限流规则
UpdateApiTimeouts	更新API超时
UpdateGatewayApi	更新API
UpdateUnitRule	更新单元化规则

服务治理相关接口

接口名称	接口功能
CreateLane	创建泳道
CreateLaneRule	创建泳道规则
DeleteLane	删除泳道
DescribeLaneRules	查询泳道规则列表
DescribeLanes	查询泳道列表

接口名称	接口功能
ModifyLane	更新泳道信息
ModifyLaneRule	更新泳道规则

服务相关接口

接口名称	接口功能
CreateMicroservice	新增微服务
DeleteMicroservice	删除微服务
DescribeApiDetail	查询API详情
DescribeApiVersions	查询API版本
DescribeCreateGatewayApiStatus	查询一键导入API分组任务的状态
DescribeMicroservice	查询微服务详情
DescribeMicroservices	获取微服务列表
DescribeMsApiList	查询服务API列表
ModifyMicroservice	修改微服务详情

程序包相关接口

接口名称	接口功能
CreateRepository	创建仓库
DeletePkgs	批量删除包
DeleteRepository	删除仓库
DescribeDownloadInfo	获取下载程序包信息
DescribePkgs	获取某个应用的程序包信息列表
DescribeRepositories	查询仓库列表
DescribeRepository	查询仓库信息
DescribeUploadInfo	获取上传程序包信息
ModifyUploadInfo	更新上传程序包信息
UpdateRepository	更新仓库信息

部署组相关接口

接口名称	接口功能
CreateContainGroup	创建容器部署组
CreateGroup	创建虚拟机部署组
CreateSecret	创建镜像凭证
CreateServerlessGroup	创建Serverless部署组
DeleteContainerGroup	删除容器部署组
DeleteGroup	删除虚拟机部署组
DeleteServerlessGroup	删除部署组
DeployContainerGroup	部署容器应用
DeployGroup	部署虚拟机部署组应用
DeployServerlessGroup	部署Serverless应用
DescribeContainerGroupDetail	查询容器部署组详情
DescribeContainerGroups	容器部署组列表
DescribeGroup	查询虚拟机部署组详情
DescribeGroupInstances	查询虚拟机部署组云主机列表
DescribeGroups	获取虚拟机部署组列表
DescribeSecretList	获取凭证列表
DescribeSecretNames	获取凭证名称列表
DescribeServerlessGroup	查询Serverless部署组明细
DescribeServerlessGroups	查询Serverless部署组列表
DescribeSimpleGroups	查询简单部署组列表
ExpandGroup	虚拟机部署组添加实例
ModifyContainerGroup	修改容器部署组
ModifyContainerReplicas	修改容器部署组实例数
ShrinkGroup	缩容虚拟机部署组
ShrinkInstances	虚拟机部署组下线实例
StartContainerGroup	启动容器部署组
StartGroup	启动虚拟机部署组

接口名称	接口功能
StopContainerGroup	停止容器部署组
StopGroup	停止虚拟机部署组
UpdateHealthCheckSettings	更新健康检查配置

配置管理相关接口

接口名称	接口功能
CreateConfig	创建配置项
CreatePublicConfig	创建公共配置项
DeleteConfig	删除配置项
DeletePublicConfig	删除公共配置项
DescribeConfig	查询配置
DescribeConfigReleaseLogs	查询配置发布历史
DescribeConfigReleases	查询配置发布信息
DescribeConfigSummary	查询配置汇总列表
DescribeConfigs	查询配置项列表
DescribePublicConfig	查询公共配置（单条）
DescribePublicConfigReleaseLogs	查询公共配置发布历史
DescribePublicConfigReleases	查询公共配置发布信息
DescribePublicConfigSummary	查询公共配置汇总列表
DescribePublicConfigs	查询公共配置项列表
DescribeReleasedConfig	查询group发布的配置
ReleaseConfig	发布配置
ReleasePublicConfig	发布公共配置
RevocationConfig	撤回已发布的配置
RevocationPublicConfig	撤回已发布的公共配置
RollbackConfig	回滚配置

镜像相关接口

接口名称	接口功能
DeleteImageTags	批量删除镜像版本
DescribeImageRepository	镜像仓库列表
DescribeImageTags	镜像版本列表

集群相关接口

接口名称	接口功能
AddClusterInstances	集群添加云主机
AddInstances	集群导入云主机
CreateCluster	创建集群
DescribeClusterInstances	查询集群实例
DescribeSimpleClusters	查询简单集群列表
RemoveInstances	移除云主机

调用方式
接口签名v1

最近更新时间: 2025-02-18 17:50:35

tcecloud API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。 签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。 安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录tcecloud管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou

参数名称	中文	参数值
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
'Timestamp' : 1465185768,
'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。 将把上一步排序好的请求参数格式化成"参数名称"="参数值"的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 Action=DescribeInstances 。 **注意：“参数值”为原始值而非url编码后的值。**

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。 签名原文字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.finance.cloud.tencent.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 / 。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。 首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中"X"和"Y"为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误

错误代码	错误描述
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的tcecloud SDK 3.0 ，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java
- PHP
- Go
- JavaScript
- .NET

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK 。

最终输出的 url 可能为： http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url ，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp 。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TceCloudAPIDemo {
private final static String CHARSET = "UTF-8";

public static String sign(String s, String key, String method) throws Exception {
Mac mac = Mac.getInstance(method);
```

```
SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
mac.init(secretKeySpec);
byte[] hash = mac.doFinal(s.getBytes(CHARSET));
return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
StringBuilder s2s = new StringBuilder("GETcvm.finance.cloud.tencent.com/?");
// 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
for (String k : params.keySet()) {
s2s.append(k).append("=").append(params.get(k).toString()).append("&");
}
return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
StringBuilder url = new StringBuilder("http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?");
// 实际请求的url中对参数顺序没有要求
for (String k : params.keySet()) {
// 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
params.put("Action", "DescribeInstances"); // 公共参数
params.put("Version", "2017-03-12"); // 公共参数
params.put("Region", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE", "HmacSHA1")); // 公共
参数
System.out.println(getUrl(params));
}
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： `pip install requests` 。

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import time

import requests
```

```
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
s = method + endpoint + "/?"
query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
return s + query_str

def sign_str(key, s, method):
hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
return base64.b64encode(hmac_str)

if __name__ == '__main__':
endpoint = "cvm.finance.cloud.tencent.com"
data = {
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2025-02-18 17:50:35

tcecloud API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。 签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。 安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录tcecloud管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；

- CanonicalURI：URI 参数，API 3.0 固定为正斜杠（/）；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为：`content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n`
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1）头部 key 统一转成小写；2）多个头部 key（小写）按照字典排序进行拼接，并且以分号（;）分隔。此例中为：`content-type;host`
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。**Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致**；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

> 1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。

> 2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
Algorithm + ' ' +
'Credential=' + SecretId + '/' + CredentialScope + ', ' +
'SignedHeaders=' + SignedHeaders + ', '
'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

3. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class TceCloudAPITC3Demo {
private final static String CHARSET = "UTF-8";
```

```
private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
private final static String PATH = "/";
private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencoded";
private final static String CT_JSON = "application/json";
private final static String CT_FORM_DATA = "multipart/form-data";

public static byte[] sign256(byte[] key, String msg) throws Exception {
Mac mac = Mac.getInstance("HmacSHA256");
SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
mac.init(secretKeySpec);
return mac.doFinal(msg.getBytes(CHARSET));
}

public static void main(String[] args) throws Exception {
String service = "cvm";
String host = "cvm.finance.cloud.tencent.com";
String region = "ap-guangzhou";
String action = "DescribeInstances";
String version = "2017-03-12";
String algorithm = "TC3-HMAC-SHA256";
String timestamp = "1539084154";
//String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
// 注意时区，否则容易出错
sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

// ************ 步骤 1：拼接规范请求串 ************
String httpRequestMethod = "GET";
String canonicalUri = "/";
String canonicalQueryString = "Limit=10&Offset=0";
String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host + "\n";
String signedHeaders = "content-type;host";
String hashedRequestPayload = DigestUtils.sha256Hex("");
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ************ 步骤 2：拼接待签名字符串 ************
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ************ 步骤 3：计算签名 ************
byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
byte[] secretService = sign256(secretDate, service);
byte[] secretSigning = sign256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ************ 步骤 4：拼接 Authorization ************
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);
```

```
TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Host", host);
headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
payload = ""
if http_request_method == "POST":
canonical_querystring = ""
ct = "json"
payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + "/" + service + "/" + "tc3_request"
```

```
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)


# ************* 步骤 3：计算签名 *************
# 计算签名摘要函数
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* 步骤 4：拼接 Authorization *************
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
"Authorization": authorization,
"Host": host,
"Content-Type": "application/%s" % ct,
"X-TC-Action": action,
"X-TC-Timestamp": str(timestamp),
"X-TC-Version": version,
"X-TC-Region": region,
}
```

请求结构

最近更新时间: 2025-02-18 17:50:35

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 **查询地域列表** 查看完成的地域列表。

2. 通信协议

tcecloud API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data（仅部分接口支持），必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

4. 字符编码

均使用UTF-8编码。

返回结果

最近更新时间: 2025-02-18 17:50:35

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
"Response": {
"TotalCount": 0,
"InstanceStatusSet": [],
"RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
}
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0， InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
"Response": {
"Error": {
"Code": "AuthFailure.SignatureFailure",
"Message": "The provided credentials could not be validated. Please check your signature is correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

公共参数

最近更新时间: 2025-02-18 17:50:35

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。

参数名称	类型	必选	描述
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。tcecloud交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 **查询地域列表** 查看完成的地域列表。

其他接口
创建工作流

最近更新时间: 2025-02-18 17:50:36

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建工作流

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 19:38:50。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateTaskFlow
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowName	是	否	String	工作流名称
TriggerRule	是	否	TaskRule	触发方式
FlowEdges	是	否	Array of TaskFlowEdge	工作流任务节点列表
TimeOut	是	否	Uint64	工作流执行超时时间

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 工作流 ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
InternalError.TaskInternalError	
FailedOperation.TaskCreateError	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	

TSF基本资源信息概览

最近更新时间: 2025-02-18 17:50:37

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

TSF基本资源信息概览接口

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-24 11:57:20。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeBasicResourceUsage
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
All	否	否	Bool	是否全量数据

3. 输出参数

参数名称	类型	描述
Result	OverviewBasicResourceUsage	**此参数对外不可见。** TSF基本资源信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnsupportedOperation.UnsupportAction	

错误码	描述
UnauthorizedOperation.NoPrivilege	
FailedOperation.NamespaceQueryFailed	
InternalError.UnhandledException	
ResourceNotFound.MicroserviceOffline	

获取容器事件列表

最近更新时间: 2025-02-18 17:50:37

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取容器事件列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-05-12 11:35:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeContainerEvents
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ResourceType	是	否	String	event 的资源类型, group 或者 instance
ResourceId	是	否	String	event 的资源 id
Offset	否	否	Int64	偏移量，取值从0开始
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~50
GroupId	否	否	String	当类型是 instance 时需要

3. 输出参数

参数名称	类型	描述
Result	TsfPageContainerEvent	**此参数对外不可见。** events 分页列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取部署组实例列表

最近更新时间: 2025-02-18 17:50:38

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取部署组实例列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-02-22 16:24:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePodInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	实例所属groupId
Offset	否	否	Int64	偏移量，取值从0开始
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~50
PodNameList	否	否	Array of String	过滤字段

3. 输出参数

参数名称	类型	描述
Result	GroupPodResult	**此参数对外不可见。** 查询的权限数据对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ContainergroupGroupNamespaceClusterNotFound	
InternalError.ContainergroupKuberneteConnectError	
InvalidParameter.ParamError	
ResourceNotFound.ClusterNotExist	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
InvalidParameterValue.ContainergroupGroupidNull	
ResourceNotFound.MicroserviceOffline	

分片下载程序包

最近更新时间: 2025-02-18 17:50:40

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

分片下载程序包

默认接口请求频率限制：20次/秒。

接口更新时间：2022-05-13 15:59:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DownloadMultipartPkg
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PkgId	是	否	String	程序包ID
FileSeg	是	否	Uint64	当前分片 (从1开始)

3. 输出参数

参数名称	类型	描述
Result	MultipartPkg	**此参数对外不可见。** 分片下载响应详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询容器标准输出日志

最近更新时间: 2025-02-18 17:50:42

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询容器标准输出日志

默认接口请求频率限制：20次/秒。

接口更新时间：2022-04-07 16:06:38。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SearchContainerStdoutLog
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	容器部署组ID
InstanceId	否	否	String	容器部署组实例 ID，可选，若为空，则返回所有实例的日志
TailLines	否	否	Uint64	最大显示行数[1~1000]，默认200
IsRealtime	否	否	Bool	是否是实时日志
Type	否	否	String	类型

3. 输出参数

参数名称	类型	描述
Result	TsfBusinessLog	**此参数对外不可见。** 容器日志输出结果
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

停止一个工作流批次

最近更新时间: 2025-02-18 17:50:42

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停止一个工作流批次

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 20:02:59。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：TerminateTaskFlowBatch
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowBatchId	是	否	String	工作流批次 ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否停止成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

分布式任务调度相关接口
续跑任务批次

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

对执行失败的任务批次执行续跑

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 19:30:08。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ContinueRunFailedTaskBatch
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
BatchId	是	否	String	批次ID。

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 成功或失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	

错误码	描述
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
FailedOperation.TaskCreateError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
FailedOperation.TaskOperationFailed	
UnauthorizedOperation.NoPrivilege	

创建任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建任务

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-01 15:05:46。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskName	是	否	String	任务名称，任务长度64字符
TaskContent	是	否	String	任务内容，长度限制65536个字节
ExecuteType	是	否	String	执行类型，UNICAST/BROADCAST
TaskType	是	否	String	任务类型
TimeOut	是	否	Uint64	任务超时时间， 时间单位 ms
TaskRule	否	否	TaskRule	触发规则
RetryCount	否	否	Uint64	重试次数，0 <= RetryCount<= 10
RetryInterval	否	否	Uint64	重试间隔， 0 <= RetryInterval <= 600000， 时间单位 ms
GroupId	是	否	String	部署组ID
ShardCount	否	否	Int64	分片数量
ShardArguments	否	否	Array of ShardArgument	分片参数
SuccessOperator	否	否	String	判断任务成功的操作符

参数名称	必选	允许NULL	类型	描述
SuccessRatio	否	否	String	判断任务成功率的阈值，如100
AdvanceSettings	否	否	AdvanceSettings	高级设置
TaskArgument	否	否	String	任务参数，长度限制10000个字符

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
InternalError.TaskInternalError	
FailedOperation.TaskCreateError	
FailedOperation.TaskOperationForbidden	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	
InternalError.UnhandledException	

删除任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除任务

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 14:55:58。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除成功or失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskDeleteError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
UnauthorizedOperation.NoPrivilege	

查询工作流最近一个批次的执行状态

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询工作流最新一个批次的状态信息

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 19:32:53。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeFlowLastBatchState
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowId	是	否	String	工作流 ID

3. 输出参数

参数名称	类型	描述
Result	TaskFlowLastBatchState	**此参数对外不可见。** 工作流批次最新状态
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

查询任务详情

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询任务详情

默认接口请求频率限制：20次/秒。

接口更新时间：2022-04-22 17:37:12。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeTaskDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID
TaskLogId	否	否	String	任务历史ID

3. 输出参数

参数名称	类型	描述
Result	TaskRecord	**此参数对外不可见。** 任务详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	

错误码	描述
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
ResourceNotFound.TaskNotFound	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	

查看任务最近执行批次状态

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询任务最近一次执行状态

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 15:09:06。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeTaskLastStatus
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	TaskLastExecuteStatus	**此参数对外不可见。** 任务上一次执行状态
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
UnauthorizedOperation.NoPrivilege	

停用任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停用任务

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 14:56:40。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DisableTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作成功 or 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskUpdateError	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
UnauthorizedOperation.NoPrivilege	

停用工作流

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停用工作流

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 15:04:36。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DisableTaskFlow
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowId	是	否	String	工作流 ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true成功，false: 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskUpdateError	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

启用任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启用任务

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 14:55:10。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：EnableTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	启用任务

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作成功or失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskUpdateError	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
UnauthorizedOperation.NoPrivilege	

启用工作流

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启用工作流

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 15:15:12。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：EnableTaskFlow
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowId	是	否	String	工作流 ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true成功，false: 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskUpdateError	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

手动执行一次任务。

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

手动执行一次任务。

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 11:18:33。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ExecuteTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务 ID。

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 成功/失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnsupportedOperation.UnsupportAction	
FailedOperation.TaskOperationFailed	
UnauthorizedOperation.NoPrivilege	

执行一次工作流

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

执行一次工作流

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 15:05:29。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ExecuteTaskFlow
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowId	否	否	String	工作流 ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 工作流批次ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
FailedOperation.TaskPushError	
UnauthorizedOperation.NoPrivilege	

修改任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

修改任务

默认接口请求频率限制：20次/秒。

接口更新时间：2022-04-22 17:37:18。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID
TaskName	否	否	String	任务名称
TaskType	否	否	String	任务类型
TaskContent	否	否	String	任务内容
ExecuteType	否	否	String	任务执行类型
TaskRule	否	否	TaskRule	触发规则
TimeOut	否	否	Uint64	超时时间，单位 ms
GroupId	否	否	String	分组ID
ShardCount	否	否	Int64	分片数量
ShardArguments	否	否	Array of ShardArgument	分片参数
AdvanceSettings	否	否	AdvanceSettings	高级设置
SuccessOperator	否	否	String	判断任务成功的操作符 GT/GTE
SuccessRatio	否	否	Int64	判断任务成功率的阈值
RetryCount	否	否	Uint64	重试次数

参数名称	必选	允许NULL	类型	描述
RetryInterval	否	否	Uint64	重试间隔
TaskArgument	否	否	String	任务参数，长度限制10000个字符

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 更新是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskUpdateError	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
InvalidParameterValue.InvalidParameterFormat	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	

重新执行任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

重新执行任务

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 14:53:04。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RedoTask
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 操作成功or失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
FailedOperation.TaskTerminateFailed	
UnsupportedOperation.TaskNotSupported	
UnauthorizedOperation.NoPrivilege	

重新执行任务批次

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

重新执行任务批次

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 19:40:10。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RedoTaskBatch
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
TaskId	是	否	String	任务ID
BatchId	是	否	String	批次ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 批次ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	

错误码	描述
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

重新执行任务的一次执行

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

重新执行在某个节点上执行任务。

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 11:17:46。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RedoTaskExecute
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
BatchId	是	否	String	任务批次ID
ExecuteId	是	否	String	任务执行ID
TaskId	是	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

重新执行工作流批次

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

重新执行工作流批次

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-17 15:10:59。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RedoTaskFlowBatch
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
FlowBatchId	是	否	String	工作流批次 ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 工作流批次历史 ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	

错误码	描述
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
FailedOperation.TaskOperationFailed	
UnauthorizedOperation.NoPrivilege	

停止执行中的任务批次

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停止执行中的任务批次， 非运行中的任务不可调用。

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 19:58:39。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StopTaskBatch
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
BatchId	是	否	String	批次ID
TaskId	否	否	String	参数ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作成功 or 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	

错误码	描述
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
InvalidParameterValue.TaskParameterInvalid	
UnauthorizedOperation.NoPrivilege	

停止正在执行的任务

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停止正在某个节点上执行的任务

默认接口请求频率限制：20次/秒。

接口更新时间：2020-09-25 20:01:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StopTaskExecute
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
BatchId	否	否	String	任务批次ID
ExecuteId	是	否	String	任务执行ID
TaskId	否	否	String	任务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作成功 or 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
UnauthorizedOperation.LicenseInactive	
UnauthorizedOperation.CamTsfRoleNoPermission	
UnauthorizedOperation.CamTsfRoleNotExist	
FailedOperation.TaskQueryError	
InternalError.TaskInternalError	
FailedOperation.TaskOperationForbidden	
ResourceNotFound.TaskNotFound	
MissingParameter.TaskParameterMissed	
FailedOperation.TaskTerminateFailed	
UnauthorizedOperation.NoPrivilege	
InternalError.UnhandledException	

命名空间相关接口
创建命名空间

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2020-12-24 19:27:56。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateNamespace
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterId	否	否	String	集群ID
NamespaceName	是	否	String	命名空间名称
NamespaceDesc	否	否	String	命名空间描述
NamespaceResourceType	否	否	String	命名空间资源类型(默认值为DEF)
NamespaceType	否	否	String	是否是全局命名空间(默认是DEF，表示普通命名空间；GLOBAL表示全局命名空间)
NamespaceId	否	否	String	命名空间ID
IsHaEnable	否	否	String	是否开启高可用
ProgramId	否	否	String	需要绑定的数据集ID

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	String	**此参数对外不可见。** 成功时为命名空间ID，失败为null
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteConnectError	
ResourceNotFound.ClusterNotExist	
UnauthorizedOperation.NoPrivilege	
InternalError.CpClusterUnavailable	
FailedOperation.NamespaceCreateFailed	
InvalidParameterValue.NamespaceAlreadyBindCluster	
InvalidParameterValue.NamespaceNameExist	
InvalidParameterValue.NamespaceNameInvalid	
InternalError.KubernetesApiCreateSecretError	

删除命名空间

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2019-11-05 17:46:13。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteNamespace
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
NamespaceId	是	否	String	命名空间ID
ClusterId	否	否	String	集群ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除命名空间是否成功。 true：删除成功。 false：删除失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ClusterNotExist	

错误码	描述
UnauthorizedOperation.NoPrivilege	
FailedOperation.GroupExists	
ResourceInUse.DefaultNamepsaceCannotBeDeleted	
ResourceInUse.NamespaceCannotDelete	
ResourceNotFound.NamespaceNotExist	
InternalError.KubernetesCallError	

查询简单命名空间列表

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询简单命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-12-18 14:41:00。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSimpleNamespaces
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
NamespaceIdList	否	否	Array of String	命名空间ID列表，不传入时查询全量
ClusterId	否	否	String	集群ID，不传入时查询全量
Limit	否	否	Int64	每页条数
Offset	否	否	Int64	起始偏移量
NamespaceId	否	否	String	命名空间ID，不传入时查询全量
NamespaceResourceTypeList	否	否	Array of String	查询资源类型列表
SearchWord	否	否	String	通过id和name进行过滤
NamespaceTypeList	否	否	Array of String	查询的命名空间类型列表
NamespaceName	否	否	String	通过命名空间名精确过滤
IsDefault	否	否	String	通过是否是默认命名空间过滤，不传表示拉取全部命名空间。0：默认，命名空间。1：非默认命名空间

3. 输出参数

参数名称	类型	描述
Result	TsfPageNamespace	**此参数对外不可见。** 命名空间分页列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.ClusterNotExist	
UnauthorizedOperation.NoPrivilege	
FailedOperation.NamespaceQueryFailed	
InternalError.UnhandledException	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	

应用相关接口
创建应用

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建应用

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-27 10:25:44。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateApplication
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationName	是	否	String	应用名称
ApplicationType	是	否	String	应用类型，V：虚拟机应用；C：容器应用；S：serverless应用
ApplicationDesc	否	否	String	应用描述
ApplicationLogConfig	否	否	String	应用日志配置项，废弃参数
MicroserviceType	是	否	String	应用微服务类型，M：service mesh应用；N：普通应用；G：网关应用
ApplicationResourceType	否	否	String	应用资源类型，废弃参数
ApplicationRuntimeType	否	否	String	应用runtime类型
ProgramId	否	否	String	需要绑定的数据集ID
ServiceConfigList	否	否	Array of ServiceConfig	服务配置信息列表

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 应用ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ApplicationNameExist	
UnauthorizedOperation.NoPrivilege	
InternalError.ApplicationMasterNuknownError	
InternalError.ApplicationScalableInitError	
InternalError.CloudApiProxyError	
InvalidParameterValue.ApplicationMicroTypeInvalid	
InvalidParameterValue.ApplicationNameRegxInvalid	
InvalidParameterValue.ApplicationTypeInvalid	
LimitExceeded.ErrNamespaceMaxLimit	
LimitExceeded.ErrRepoMaxLimit	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.CamGeneralError	

删除应用

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除应用

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:33:22。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteApplication
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除应用操作是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse.ApplicationCannotDelete	
UnauthorizedOperation.NoPrivilege	

错误码	描述
ResourceInUse.CvmcaeMasterCannotDelete	
InternalError.ApplicationRepoDeletePkg	
InternalError.CloudApiProxyError	
InternalError.RemoteServiceCallError	
ResourceNotFound.ApplicationNotExist	

获取应用详情

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取应用详情

默认接口请求频率限制：20次/秒。

接口更新时间：2022-09-26 16:01:08。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApplication
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID

3. 输出参数

参数名称	类型	描述
Result	ApplicationForPage	**此参数对外不可见。** 应用信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	

错误码	描述
FailedOperation.ApplicationQueryFailed	
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
MissingParameter.ApplicationIdNull	
ResourceNotFound.MicroserviceOffline	

获取应用列表其它字段

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取应用列表其它字段，如实例数量信息等

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:34:56。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApplicationAttribute
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID

3. 输出参数

参数名称	类型	描述
Result	ApplicationAttribute	**此参数对外不可见。** 应用列表其它字段返回参数
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ContainergroupGroupNamespaceClusterNotFound	
InvalidParameter.ParamError	

错误码	描述
UnauthorizedOperation.NoPrivilege	
FailedOperation.ApplicationQueryFailed	
FailedOperation.TsfPrivilegeError	
ResourceNotFound.MicroserviceOffline	

获取应用列表

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取应用列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-12-22 13:29:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApplications
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段
OrderBy	否	否	String	排序字段
OrderType	否	否	Int64	排序类型
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数
ApplicationType	否	否	String	应用类型
MicroserviceType	否	否	String	应用的微服务类型
ApplicationResourceTypeList	否	否	Array of String	应用资源类型数组
ApplicationIdList	否	否	Array of String	IdList

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageApplication	**此参数对外不可见。** 应用分页列表信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.ApplicationQueryFailed	
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
InvalidParameterValue.ApplicationPageLimitInvalid	
ResourceNotFound.MicroserviceOffline	

查询简单应用列表

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询简单应用列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-10 22:24:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSimpleApplications
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationIdList	否	否	Array of String	应用ID列表
ApplicationType	否	否	String	应用类型
Limit	否	否	Int64	每页条数
Offset	否	否	Int64	起始偏移量
MicroserviceType	否	否	String	微服务类型
ApplicationResourceTypeList	否	否	Array of String	资源类型数组
SearchWord	否	否	String	通过id和name进行关键词过滤

3. 输出参数

参数名称	类型	描述
Result	TsfPageSimpleApplication	**此参数对外不可见。** 简单应用分页对象

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
ResourceNotFound.MicroserviceOffline	

微服务网关相关接口
网关与API分组批量绑定

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

网关与API分组批量绑定

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 11:37:59。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：BindApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupGatewayList	是	否	Array of GatewayGroupIds	分组绑定网关列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

错误码	描述
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

批量绑定插件

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

插件与网关分组/API批量绑定

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-21 12:14:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：BindPlugin
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PluginInstanceList	是	否	Array of GatewayPluginBoundParam	分组/API绑定插件列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

错误码	描述
InternalError.GatewayCommonError	
InternalError.GatewayConsulError	

启用或禁用API

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启用或禁用API

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:57:51。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ChangeApiUsableStatus
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiId	是	否	String	API ID
UsableStatus	是	否	String	切换状态，enabled/disabled

3. 输出参数

参数名称	类型	描述
Result	ApiDetailInfo	**此参数对外不可见。** API 信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

一键导入API分组

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

一键导入API分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:47:50。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateAllGatewayApiAsync
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	API分组ID
MicroserviceId	是	否	String	微服务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

创建API分组

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建API分组

默认接口请求频率限制：20次/秒。

接口更新时间：2021-02-25 16:05:36。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupName	是	否	String	分组名称, 不能包含中文
GroupContext	是	否	String	分组上下文
AuthType	否	否	String	鉴权类型。secret： 密钥鉴权； none:无鉴权
Description	否	否	String	备注
GroupType	否	否	String	分组类型,默认ms。 ms： 微服务分组； external:外部Api分组
GatewayInstanceId	否	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** API分组ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
InvalidParameterValue.GatewayParameterInvalid	
UnauthorizedOperation.NoPrivilege	
InternalError.GatewayDbError	

创建API限流规则

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建API限流规则

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:40:41。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateApiRateLimitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiId	是	否	String	Api Id
UsableStatus	否	否	String	开启/禁用，enabled/disabled, 不传默认开启
MaxQps	是	否	Uint64	qps值

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	

批量导入API至api分组

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

批量导入API至api分组(也支持新建API到分组)

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 11:59:53。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateGatewayApi
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	API 分组ID
ApiList	是	否	Array of ApiInfo	Api信息

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

错误码	描述
MissingParameter.GatewayParameterRequired	
InternalError.GatewayDbError	
UnsupportedOperation.GatewayTooManyRequestParameter	

创建路径重写

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建路径重写

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-19 11:31:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreatePathRewrites
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PathRewrites	是	否	PathRewriteCreateObject	路径重写列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true/false
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

创建单元化规则

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 17:49:53。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayInstanceId	是	否	String	网关实体ID
Name	是	否	String	规则名称
Description	否	否	String	规则描述
UnitRuleItemList	否	否	Array of UnitRuleItem	规则项列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

创建通配符API

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建通配符API

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-03 21:12:49。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateWildCardGatewayApi
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Path	是	否	String	通配路径
Method	是	否	String	通配方法
GroupId	是	否	String	API分组ID
NamespaceId	是	否	String	命名空间ID
MicroserviceId	是	否	String	微服务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

删除Api分组

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除Api分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:45:28。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	API 分组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

删除路径重写

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除路径重写

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-19 11:28:26。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeletePathRewrites
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PathRewriteIds	是	否	Array of String	路径重写规则IDs

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true/false
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

删除单元化命名空间

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除单元化命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:54:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteUnitNamespaces
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayInstanceId	是	否	String	网关实体ID
UnitNamespaceList	是	否	Array of String	单元化命名空间ID数组

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

错误码	描述
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

删除单元化规则

最近更新时间: 2025-02-18 17:50:43

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:02:38。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	规则ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

查询API分组

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询API分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:28:05。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	API 分组ID

3. 输出参数

参数名称	类型	描述
Result	ApiGroupInfo	**此参数对外不可见。** API分组信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayDbError	

查询API 分组信息列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询API 分组信息列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-02-25 15:56:04。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiGroups
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索关键字
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20
GroupType	否	否	String	分组类型。 ms： 微服务分组； external:外部Api分组
AuthType	否	否	String	鉴权类型。 secret： 秘钥鉴权； none:无鉴权
Status	否	否	String	发布状态, drafted: 未发布。 released: 发布
OrderBy	否	否	String	排序字段："created_time"或"group_context"
OrderType	否	否	Int64	排序类型：0(ASC)或1(DESC)
GatewayInstanceId	否	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageApiGroupInfo	**此参数对外不可见。** 翻页结构体
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.NoPrivilege	

查询API限流规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询API限流规则

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:33:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiRateLimitRules
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiId	是	否	String	Api ID

3. 输出参数

参数名称	类型	描述
Result	ApiRateLimitRule	**此参数对外不可见。** 限流结果
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询网关API监控明细数据

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询网关API监控明细数据

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:10:11。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiUseDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayDeployGroupId	是	否	String	网关部署组ID
ApiId	是	否	String	网关分组Api ID
StartTime	是	否	Datetime	查询的日期,格式：yyyy-MM-dd HH:mm:ss
EndTime	是	否	Datetime	查询的日期,格式：yyyy-MM-dd HH:mm:ss

3. 输出参数

参数名称	类型	描述
Result	GroupApiUseStatistics	**此参数对外不可见。** 日使用统计对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
FailedOperation.GatewayRemoteCallError	

查询生效的单元化规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询生效的单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:17:57。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeEnabledUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayInstanceId	是	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述
Result	UnitRule	**此参数对外不可见。** 单元化规则对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

查询网关所有分组下Api列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询网关所有分组下Api列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:52:06。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGatewayAllGroupApis
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayDeployGroupId	是	否	String	网关部署组ID
SearchWord	否	否	String	搜索关键字，支持分组名称或API Path

3. 输出参数

参数名称	类型	描述
Result	GatewayVo	**此参数对外不可见。** 网关分组和API列表信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

错误码	描述
InvalidParameterValue.DeployGroupNotExists	

查询网关监控概览

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询网关监控概览

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:29:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGatewayMonitorOverview
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayDeployGroupId	是	否	String	网关部署组ID

3. 输出参数

参数名称	类型	描述
Result	MonitorOverview	**此参数对外不可见。** 监控概览对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
UnauthorizedOperation.NoPrivilege	
FailedOperation.GatewayRemoteCallError	

查询某个API分组已绑定的网关部署组信息列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询某个API分组已绑定的网关部署组信息列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-02-20 10:29:08。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroupBindedGateways
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	API 分组ID
SearchWord	否	否	String	搜索关键字
Offset	是	否	Int64	翻页查询偏移量
Limit	是	否	Int64	翻页查询每页记录数

3. 输出参数

参数名称	类型	描述
Result	TsfPageGatewayDeployGroup	**此参数对外不可见。** 翻页结构体
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
FailedOperation.GatewayRemoteCallError	

查询某个网关绑定的API 分组信息列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询某个网关绑定的API 分组信息列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:42:05。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroupGateways
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayDeployGroupId	是	否	String	网关部署组ID
SearchWord	否	否	String	搜索关键字
Offset	是	否	Int64	翻页查询偏移量
Limit	是	否	Int64	翻页查询每页记录数

3. 输出参数

参数名称	类型	描述
Result	TsfPageApiGroupInfo	**此参数对外不可见。** API分组信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询网关分组监控明细数据

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询网关分组监控明细数据

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:19:49。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroupUseDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayDeployGroupId	是	否	String	网关部署组ID
GroupId	是	否	String	网关分组ID
StartTime	是	否	Datetime	查询的日期,格式：yyyy-MM-dd HH:mm:ss
EndTime	是	否	Datetime	查询的日期,格式：yyyy-MM-dd HH:mm:ss
Count	否	否	Int64	指定top的条数,默认为10

3. 输出参数

参数名称	类型	描述
Result	GroupDailyUseStatistics	**此参数对外不可见。** 日使用统计对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
FailedOperation.GatewayRemoteCallError	

查询某个插件下绑定或未绑定的API分组

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询某个插件下绑定或未绑定的API分组

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-21 12:09:28。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroupsWithPlugin
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PluginId	是	否	String	插件ID
Bound	是	否	Bool	绑定/未绑定: true / false
SearchWord	否	否	String	搜索关键字
Offset	是	否	Int64	翻页偏移量
Limit	是	否	Int64	每页记录数量
GatewayInstanceId	否	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述
Result	TsfPageApiGroupInfo	**此参数对外不可见。** API分组信息列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询路径重写

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询路径重写

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-19 11:27:36。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePathRewrite
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PathRewriteId	是	否	String	路径重写规则ID

3. 输出参数

参数名称	类型	描述
Result	PathRewrite	**此参数对外不可见。** 路径重写规则对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

查询路径重写列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询路径重写列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-19 11:25:51。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePathRewrites
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayGroupId	是	否	String	网关部署组ID
SearchWord	否	否	String	根据正则表达式或替换的内容模糊查询
Limit	否	否	Uint64	每页数量
Offset	否	否	Uint64	起始偏移量

3. 输出参数

参数名称	类型	描述
Result	PathRewritePage	**此参数对外不可见。** 路径重写翻页对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
UnauthorizedOperation.NoPrivilege	

查询网关分组或API绑定（或未绑定）的插件列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

分页查询网关分组/API绑定（或未绑定）的插件列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-21 12:17:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePluginInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ScopeValue	是	否	String	分组或者API的ID
Bound	是	否	Bool	绑定: true; 未绑定: false
Offset	是	否	Int64	翻页偏移量
Limit	是	否	Int64	每页展示的条数
Type	否	否	String	插件类型
SearchWord	否	否	String	搜索关键字

3. 输出参数

参数名称	类型	描述
Result	TsfPageGatewayPlugin	**此参数对外不可见。** 插件信息列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

网关调用监控统计查询类

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

无

默认接口请求频率限制：20次/秒。

接口更新时间：2021-05-25 11:00:30。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUnitApiUseDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiId	否	否	String	网关实体ID（单元化使用详情的请求参数，非单元化忽略本参数字段）
EndTime	是	否	String	查询结束时间 格式：yyyy-MM-dd HH:mm:ss
GatewayDeployGroupId	是	否	String	网关部署组ID
GatewayInstanceId	否	否	String	网关实体ID（单元化使用详情的请求参数，非单元化忽略本参数字段）
GroupId	是	否	String	部署组ID
Limit	是	否	Int64	最大输出条数
Offset	是	否	Int64	偏移量
Period	是	否	Int64	查询监控统计粒度
StartTime	是	否	String	查询开始时间 格式：yyyy-MM-dd HH:mm:ss

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	GroupUnitApiUseStatistics	**此参数对外不可见。** 单元化使用统计对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
FailedOperation.GatewayRemoteCallError	
MissingParameter.GatewayParameterRequired	

查询单元化命名空间列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询单元化命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:34:49。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUnitNamespaces
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayInstanceId	是	否	String	网关实体ID
SearchWord	否	否	String	根据命名空间名或ID模糊查询
Offset	否	否	Int64	翻页查询偏移量
Limit	否	否	Int64	翻页查询每页记录数

3. 输出参数

参数名称	类型	描述
Result	TsfPageUnitNamespace	**此参数对外不可见。** 单元化命名空间对象列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

查询单元化规则详情

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询单元化规则详情

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-24 13:42:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	单元化规则ID

3. 输出参数

参数名称	类型	描述
Result	UnitRule	**此参数对外不可见。** 单元化规则对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

查询单元化规则列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询单元化规则列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 17:28:31。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUnitRules
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GatewayInstanceId	是	否	String	网关实体ID
SearchWord	否	否	String	根据规则名或备注内容模糊查询
Status	否	否	String	启用状态, disabled: 未发布， enabled: 发布
Offset	否	否	Int64	翻页查询偏移量
Limit	否	否	Int64	翻页查询每页记录数

3. 输出参数

参数名称	类型	描述
Result	TsfPageUnitRule	**此参数对外不可见。** 分页列表信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

查询可用于被导入的命名空间列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询可用于被导入的命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:41:49。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUsableUnitNamespaces
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	根据命名空间名或ID模糊查询
Offset	否	否	Int64	翻页查询偏移量
Limit	否	否	Int64	翻页查询每页记录数

3. 输出参数

参数名称	类型	描述
Result	TsfPageUnitNamespace	**此参数对外不可见。** 单元化命名空间对象列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

禁用单元化路由

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

禁用单元化路由

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 15:38:05。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DisableUnitRoute
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

禁用单元化规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

禁用单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:12:49。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DisableUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	规则ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

下线Api分组

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

下线Api分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-11 16:53:22。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DraftApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	Api 分组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true: 成功, false: 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

启用单元化路由

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启用单元化路由

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 15:29:36。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：EnableUnitRoute
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	网关实体ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

启用单元化规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启用单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 20:08:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：EnableUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	规则ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

错误码	描述
InternalError.GatewayConsulError	

修改路径重写

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

修改路径重写

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-19 11:28:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyPathRewrite
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
PathRewriteId	是	否	String	路径重写规则ID
Regex	否	否	String	正则表达式
Replacement	否	否	String	替换的内容
Blocked	否	否	String	是否屏蔽映射后路径，Y: 是 N: 否
Order	否	否	Int64	规则顺序，越小优先级越高

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true/false
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

发布Api分组

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

发布Api分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-11 16:56:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ReleaseApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	Api 分组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 成功/失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
UnauthorizedOperation.NoPrivilege	

API分组批量与网关解绑

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

API分组批量与网关解绑

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 11:37:10。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UnbindApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupGatewayList	是	否	Array of GatewayGroupIds	分组网关id列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.GatewayConsulError	

更新Api分组

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新Api分组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 19:30:17。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateApiGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	Api 分组ID
GroupName	否	否	String	Api 分组名称
Description	否	否	String	Api 分组描述
AuthType	否	否	String	鉴权类型
GroupContext	否	否	String	分组上下文

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，true: 成功, false: 失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	

更新API限流规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新API限流规则

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:36:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateApiRateLimitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RuleId	是	否	String	限流规则ID
UsableStatus	是	否	String	开启/禁用，enabled/disabled
MaxQps	否	否	Int64	qps值，开启限流规则时，必填

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

批量更新API限流规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

批量更新API限流规则

默认接口请求频率限制：20次/秒。

接口更新时间：2020-12-02 10:24:04。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateApiRateLimitRules
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiIds	是	否	Array of String	API ID 列表
UsableStatus	是	否	String	开启/禁用，enabled/disabled
MaxQps	否	否	Int64	QPS值。开启限流规则时，必填

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
UnauthorizedOperation.LicenseUnauthorized	
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
FailedOperation.RatelimitConsulError	
InternalError.GatewayConsistencyError	

更新API超时

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

批量更新API超时

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-17 19:56:33。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateApiTimeouts
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiIds	是	否	Array of String	API ID 列表
UsableStatus	是	否	String	开启/禁用，enabled/disabled
Timeout	否	否	Int64	超时时间，单位毫秒，开启API超时时，必填

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

更新API

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新API

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-12 20:44:09。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateGatewayApi
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApiId	是	否	String	API ID
Path	否	否	String	API 路径
Method	否	否	String	Api 请求方法
PathMapping	否	否	String	请求映射
Host	否	否	String	api所在服务host
Description	否	否	String	api描述信息

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 返回结果，成功失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	

更新单元化规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新单元化规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-11 19:53:47。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateUnitRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Id	是	否	String	规则ID
Name	是	否	String	规则名称
Description	否	否	String	规则描述
UnitRuleItemList	否	否	Array of UnitRuleItem	规则项列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GatewayParameterInvalid	
MissingParameter.GatewayParameterRequired	
InternalError.GatewayConsulError	

服务治理相关接口
创建泳道

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建泳道

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-13 15:34:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateLane
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
LaneName	是	否	String	泳道名称
Remark	是	否	String	泳道备注
LaneGroupList	是	否	Array of LaneGroup	泳道部署组信息

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 泳道ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	

错误码	描述
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	
FailedOperation.ServiceInsertFailed	

创建泳道规则

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建泳道规则

默认接口请求频率限制：20次/秒。

接口更新时间：2021-05-21 15:44:51。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateLaneRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RuleName	是	否	String	泳道规则名称
Remark	否	否	String	泳道规则备注
RuleTagList	是	否	Array of LaneRuleTag	泳道规则标签列表
RuleTagRelationship	是	否	String	泳道规则标签关系
LaneId	是	否	String	泳道Id

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 泳道规则Id
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameter.LaneRuleTagValueTotalTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExist	
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneInfoNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameterValue.LaneRuleTagValueTooLong	

错误码	描述
FailedOperation.LaneRuleMaxLimit	
InvalidParameter.LaneRuleTagValueTooLong	
FailedOperation.LaneInfoReleaseConsulFailed	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneRuleTagNameNotEmpty	
FailedOperation.LaneInfoDeleteConsulFailed	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
FailedOperation.LaneRuleEnableConsulFailed	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameter.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneInfoNameNotEmpty	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneInfoNameTooLong	

删除泳道

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除泳道

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-13 16:31:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteLane
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
LaneId	是	否	String	泳道Idl

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true / false
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	

错误码	描述
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	

错误码	描述
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	

查询泳道规则列表

最近更新时间: 2025-02-18 17:50:44

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询泳道规则列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-13 15:30:16。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeLaneRules
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Limit	是	否	Int64	每页展示的条数
Offset	是	否	Int64	翻页偏移量
SearchWord	否	否	String	搜索关键词
RuleId	否	否	String	泳道规则ID（用于精确搜索）

3. 输出参数

参数名称	类型	描述
Result	LaneRules	**此参数对外不可见。** 泳道规则列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	

错误码	描述
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	
UnauthorizedOperation.NoPrivilege	
ResourceNotFound.LicenseServerNotFound	

查询泳道列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询泳道列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-09-26 14:46:12。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeLanes
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Limit	否	否	Int64	每页展示的条数
Offset	否	否	Int64	翻页偏移量
SearchWord	否	否	String	搜索关键字

3. 输出参数

参数名称	类型	描述
Result	LaneInfos	**此参数对外不可见。** 泳道列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	

错误码	描述
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	
UnauthorizedOperation.NoPrivilege	

更新泳道信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新泳道信息

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-13 15:14:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyLane
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
LaneId	是	否	String	泳道ID
LaneName	是	否	String	泳道名称
Remark	是	否	String	备注

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作状态
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	

错误码	描述
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	

更新泳道规则

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新泳道规则

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-13 15:33:20。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyLaneRule
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RuleId	是	否	String	泳道规则ID
RuleName	是	否	String	泳道规则名称
Remark	是	否	String	泳道规则备注
RuleTagList	是	否	Array of LaneRuleTag	泳道规则标签列表
RuleTagRelationship	是	否	String	泳道规则标签关系
LaneId	是	否	String	泳道ID
Enable	是	否	Bool	开启状态

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 操作状态
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.LaneInfoNameNotEmpty	
InvalidParameter.LaneInfoNameTooLong	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameter.LaneInfoAlreadyUsed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNotExist	
InvalidParameter.LaneRuleNotExist	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameter.LaneRuleNameAlreadyUsed	
InvalidParameter.LaneRuleNameInvalid	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.LaneRuleTagNotEmpty	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameter.LaneRuleTagNameTooLong	
InvalidParameter.LaneRuleTagValueTooLong	
InvalidParameter.LaneRuleTagValueTotalTooLong	
FailedOperation.LaneRuleEnableConsulFailed	
FailedOperation.LaneInfoReleaseConsulFailed	
FailedOperation.LaneInfoDeleteConsulFailed	
FailedOperation.LaneInfoGroupNotEmpty	
InvalidParameter.LaneRuleNameNotEmpty	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.LaneInfoNameNotEmpty	

错误码	描述
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.LaneInfoNameInvalid	
InvalidParameterValue.LaneInfoNameAlreadyUsed	
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameterValue.LaneInfoNotExist	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.LaneRuleNameNotEmpty	
InvalidParameterValue.LaneRuleNameTooLong	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.LaneRuleRemarkTooLong	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.LaneRuleTagNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	

服务相关接口
新增微服务

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

新增微服务

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:38:15。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateMicroservice
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
NamespaceId	是	否	String	命名空间ID
MicroserviceName	是	否	String	微服务名称
MicroserviceDesc	否	否	String	微服务描述信息

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 新增微服务是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ServiceNameRepeated	
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ServiceDescLength	

删除微服务

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除微服务

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:38:34。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteMicroservice
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除微服务是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ServiceNotExist	
UnauthorizedOperation.NoPrivilege	

查询API详情

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询API详情

默认接口请求频率限制：20次/秒。

接口更新时间：2020-06-12 14:42:09。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务id
Path	是	否	String	请求路径
Method	是	否	String	请求方法
PkgVersion	是	否	String	包版本
ApplicationId	是	否	String	应用ID

3. 输出参数

参数名称	类型	描述
Result	ApiDetailResponse	**此参数对外不可见。** API 详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.ApiMetaParseFailed	
FailedOperation.ServiceQueryFailed	
ResourceNotFound.ServiceNotExist	
MissingParameter.RequiredParameterMissing	

查询API版本

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询API 版本

默认接口请求频率限制：200次/秒。

接口更新时间：2020-06-12 14:41:20。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApiVersions
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务ID
Path	否	否	String	API 请求路径
Method	否	否	String	请求方法

3. 输出参数

参数名称	类型	描述
Result	ApiVersionArray	**此参数对外不可见。** API版本列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
ResourceNotFound.ServiceNotExist	
MissingParameter.RequiredParameterMissing	

查询一键导入API分组任务的状态

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询一键导入API分组任务的状态

默认接口请求频率限制：20次/秒。

接口更新时间：2020-11-30 15:15:09。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCreateGatewayApiStatus
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	否	否	String	请求方法
MicroserviceId	否	否	String	微服务ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否已完成导入任务
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询微服务详情

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询微服务详情

默认接口请求频率限制：20次/秒。

接口更新时间：2021-03-18 16:53:17。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeMicroservice
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务ID
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数
GroupIds	否	否	Array of String	可选，根据部署组ID进行过滤

3. 输出参数

参数名称	类型	描述
Result	TsfPageMsInstance	**此参数对外不可见。** 微服务详情实例列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ServiceNotExist	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
MissingParameter.ServiceIdRequired	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	

获取微服务列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取微服务列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-04-06 15:40:13。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeMicroservices
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段
NamespaceId	是	否	String	命名空间ID
OrderBy	否	否	String	排序字段
OrderType	否	否	Int64	排序类型
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数
Status	否	否	Array of String	状态过滤，online、offline、single_online
MicroserviceIdList	否	否	Array of String	IdList
MicroserviceNameList	否	否	Array of String	搜索的服务名列表

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageMicroservice	**此参数对外不可见。** 微服务分页列表信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
MissingParameter.NamespaceIdRequired	
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	

查询服务API列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询服务API列表

默认接口请求频率限制：200次/秒。

接口更新时间：2020-06-12 14:40:41。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeMsApiList
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务ID
SearchWord	否	否	String	搜索关键字
Limit	否	否	Int64	每页的数量
Offset	否	否	Int64	翻页偏移量

3. 输出参数

参数名称	类型	描述
Result	TsfApiListResponse	**此参数对外不可见。** 相应结果
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.ServiceQueryFailed	
ResourceNotFound.ServiceNotExist	
MissingParameter.RequiredParameterMissing	
FailedOperation.ServiceInsertFailed	
InternalError.UnhandledException	
ResourceNotFound.MicroserviceOffline	

修改微服务详情

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

修改微服务详情

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:39:22。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyMicroservice
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
MicroserviceId	是	否	String	微服务 ID
MicroserviceDesc	是	否	String	微服务备注信息

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 修改微服务详情是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ServiceNotExist	

程序包相关接口
创建仓库

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建仓库

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-29 19:53:11。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRepository
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RepositoryName	是	否	String	仓库名称
RepositoryType	是	否	String	仓库类型（默认仓库：default，私有仓库：private）
BucketName	是	否	String	仓库所在桶名称
BucketRegion	是	否	String	仓库所在桶地域
Directory	否	否	String	目录
RepositoryDesc	否	否	String	仓库描述

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 创建仓库是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	
ResourceInUse.ObjectExist	

批量删除包

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

从软件仓库批量删除程序包。 一次最多支持删除1000个包，数量超过1000，返回UpperDeleteLimit错误。

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-29 21:30:29。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeletePkgs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID
PkgIds	是	否	Array of String	需要删除的程序包ID列表
RepositoryType	否	否	String	程序包仓库类型
RepositoryId	否	否	String	程序包仓库id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	

错误码	描述
InternalError.RuntimeError	
InvalidParameter.PackageInUse	
InvalidParameter.UpperDeleteLimit	

删除仓库

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除仓库

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-23 11:17:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRepository
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RepositoryId	是	否	String	仓库ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除仓库是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	

错误码	描述
InvalidParameter.RepositoryNotEmpty	
ResourceNotFound.ObjectNoExist	

获取下载程序包信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

TSF上传的程序包存放在腾讯云对象存储（COS）中，通过该API可以获取从COS下载程序包需要的信息，包括包所在的桶、存储路径、鉴权信息等，之后使用COS API（或SDK）进行下载。 COS相关文档请查阅：
http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/436

默认接口请求频率限制：20次/秒。

接口更新时间：2020-06-22 11:36:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeDownloadInfo
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID
PkgId	是	否	String	程序包ID
RepositoryId	否	否	String	程序包仓库ID
RepositoryType	否	否	String	程序包仓库类型

3. 输出参数

参数名称	类型	描述
Result	CosDownloadInfo	**此参数对外不可见。** COS鉴权信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ObjectNoExist	

获取某个应用的程序包信息列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

无

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-27 14:11:47。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePkgs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID（只传入应用ID，返回该应用下所有软件包信息）
SearchWord	否	否	String	查询关键字（支持根据包ID，包名，包版本号搜索）
OrderBy	否	否	String	排序关键字（默认为"UploadTime"：上传时间）
OrderType	否	否	Uint64	升序：0/降序：1（默认降序）
Offset	否	否	Uint64	查询起始偏移
Limit	否	否	Uint64	返回数量限制
RepositoryType	否	否	String	程序包仓库类型
RepositoryId	否	否	String	程序包仓库id
PackageTypeList	否	否	Array of String	程序包类型数组支持（fatjar jar war tar.gz zip）

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	PkgList	**此参数对外不可见。** 符合查询程序包信息列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	

查询仓库列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询仓库列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-23 11:19:53。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRepositories
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	查询关键字（按照仓库名称搜索）
Offset	否	否	Uint64	查询起始偏移
Limit	否	否	Uint64	返回数量限制
RepositoryType	否	否	String	仓库类型（默认仓库：default，私有仓库：private）

3. 输出参数

参数名称	类型	描述
Result	RepositoryList	**此参数对外不可见。** 符合查询仓库信息列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	

查询仓库信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询仓库信息

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-29 19:56:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRepository
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RepositoryId	是	否	String	仓库ID

3. 输出参数

参数名称	类型	描述
Result	RepositoryInfo	**此参数对外不可见。** 查询的仓库信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	

错误码	描述
ResourceNotFound.ObjectNoExist	

获取上传程序包信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

TSF会将软件包上传到腾讯云对象存储（COS）。调用此接口获取上传信息，如目标地域，桶，包Id，存储路径，鉴权信息等，之后请使用COS API（或SDK）进行上传。 COS相关文档请查阅：
http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/436

默认接口请求频率限制：20次/秒。

接口更新时间：2020-06-19 11:03:19。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeUploadInfo
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID
PkgName	是	否	String	程序包名
PkgVersion	是	否	String	程序包版本
PkgType	是	否	String	程序包类型
PkgDesc	否	否	String	程序包介绍
RepositoryType	否	否	String	程序包仓库类型
RepositoryId	否	否	String	程序包仓库id

3. 输出参数

参数名称	类型	描述
Result	CosUploadInfo	**此参数对外不可见。** COS上传信息

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	
ResourceInUse.ObjectExist	
ResourceInsufficient.PackageSpaceFull	

更新上传程序包信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

调用该接口和COS的上传接口后，需要调用此接口更新TSF中保存的程序包状态。 调用此接口完成后，才标志上传包流程结束。

默认接口请求频率限制：20次/秒。

接口更新时间：2020-10-09 11:10:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyUploadInfo
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用ID
PkgId	是	否	String	调用DescribeUploadInfo接口时返回的软件包ID
Result	是	否	Int64	COS返回上传结果（默认为0：成功，其他值表示失败）
Md5	是	否	String	程序包MD5
Size	否	否	Uint64	程序包大小（单位字节）
RepositoryType	否	否	String	程序包仓库类型
RepositoryId	否	否	String	程序包仓库id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InternalError.RuntimeError	
ResourceNotFound.ObjectNoExist	

更新仓库信息

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新仓库信息

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-29 19:46:04。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateRepository
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
RepositoryId	是	否	String	仓库ID
RepositoryDesc	否	否	String	仓库描述

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 更新仓库是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	

错误码	描述
InternalError.RuntimeError	

部署组相关接口
创建容器部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-04-07 10:59:52。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateContainGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	分组所属应用ID
NamespaceId	是	否	String	分组所属命名空间ID
GroupName	是	否	String	分组名称字段，长度1~60，字母或下划线开头，可包含字母数字下划线
CpuLimit	否	否	String	最大分配 CPU 核数，对应 K8S limit
MemLimit	否	否	String	最大分配内存 MiB 数，对应 K8S limit
InstanceNum	是	否	Int64	实例数量
AccessType	是	否	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	是	否	Array of ProtocolPort	数组对象，见下方定义
GroupComment	否	否	String	分组备注字段，长度应不大于200字符
UpdateType	否	否	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	否	否	Int64	滚动更新必填，更新间隔

参数名称	必选	允许NULL	类型	描述
ClusterId	是	否	String	集群ID
CpuRequest	否	否	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	否	否	String	初始分配的内存 MiB 数，对应 K8S request
GroupResourceType	否	否	String	部署组资源类型
SubnetId	否	否	String	子网ID
AgentCpuRequest	否	否	String	agent 容器分配的 CPU 核数，对应 K8S 的 request
AgentCpuLimit	否	否	String	agent 容器最大的 CPU 核数，对应 K8S 的 limit
AgentMemRequest	否	否	String	agent 容器分配的内存 MiB 数，对应 K8S 的 request
AgentMemLimit	否	否	String	agent 容器最大的内存 MiB 数，对应 K8S 的 limit
IstioCpuRequest	否	否	String	istioproxy 容器分配的 CPU 核数，对应 K8S 的 request
IstioCpuLimit	否	否	String	istioproxy 容器最大的 CPU 核数，对应 K8S 的 limit
IstioMemRequest	否	否	String	istioproxy 容器分配的内存 MiB 数，对应 K8S 的 request
IstioMemLimit	否	否	String	istioproxy 容器最大的内存 MiB 数，对应 K8S 的 limit

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 返回创建成功的部署组ID，返回null表示失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ContainergroupProtocolInvalid	
InvalidParameterValue.ContainergroupPortInvalid	
InvalidParameter.ParamError	
InvalidParameterValue.GroupNameExist	
InternalError.GroupMasterNuknownError	

错误码	描述
InvalidParameterValue.ContainergroupInvalidCpuInfo	
InvalidParameterValue.ContainergroupInvalidMemInfo	
InvalidParameterValue.ContainergroupNodePortInvalid	
ResourceNotFound.GroupApplicationNotExist	
ResourceNotFound.GroupNamespaceNotExist	
InvalidParameterValue.ContainergroupGroupnameRegexMatchFalse	
InvalidParameterValue.ContainergroupGroupnameLegnth	

创建虚拟机部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建虚拟机部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-04-14 11:27:33。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	部署组所属的应用ID
NamespaceId	是	否	String	部署组所属命名空间ID
GroupName	是	否	String	部署组名称
GroupDesc	否	否	String	部署组描述
ClusterId	是	否	String	集群ID
GroupResourceType	否	否	String	部署组资源类型

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** groupId， null表示创建失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GroupNameExist	
InvalidParameterValue.GroupNameLength	
InvalidParameterValue.GroupNameRegxMismatch	
UnauthorizedOperation.NoPrivilege	
ResourceNotFound.CvmcaeMasterResourceNotFound	
InternalError.GroupMasterNuknownError	
ResourceNotFound.ContainergroupClusterNotfound	
ResourceNotFound.GroupApplicationNotExist	
ResourceNotFound.GroupNamespaceNotExist	
ResourceNotFound.MicroserviceOffline	

创建镜像凭证

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建镜像凭证

默认接口请求频率限制：20次/秒。

接口更新时间：2022-05-12 15:40:17。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateSecret
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	凭证名称
RepositoryDomain	是	否	String	镜像仓库域名
Username	否	否	String	镜像仓库登录用户名
Password	否	否	String	镜像仓库登录密码
AuthType	否	否	String	认证类型，参考 K8S 认证类型，目前仅支持 'kubernetes.io/dockercfg'，可不传。默认为 'kubernetes.io/dockercfg'

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 是否新建成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

创建Serverless部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建Serverless部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-12-26 10:38:36。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateServerlessGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	分组所属应用ID
GroupName	是	否	String	分组名称字段，长度1~60，字母或下划线开头，可包含字母数字下划线
NamespaceId	是	否	String	分组所属名字空间ID
ClusterId	是	否	String	分组所属集群ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 创建成功的部署组ID，返回null表示失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue	
InvalidParameterValue.GroupNameExist	
InvalidParameterValue.GroupNameNull	
InvalidParameterValue.GroupNameLength	
InvalidParameterValue.GroupNameRegxMismatch	
InvalidParameterValue.GroupPkgNull	

删除容器部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:02:00。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteContainerGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID，分组唯一标识

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除操作是否成功： true：成功 false：失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteConnectError	
InternalError.ContainergroupKuberneteApiInvokeError	

错误码	描述
UnauthorizedOperation.NoPrivilege	
ResourceNotFound.ContainergroupGroupNotFound	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.CamGeneralError	

删除虚拟机部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:06:13。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 删除部署组操作是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.GroupNotExist	
InternalError.UnhandledException	

删除部署组

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除Serverless部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-11-21 19:38:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteServerlessGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	groupId，分组唯一标识

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 结果true：成功；false：失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.GroupNotExist	
InvalidParameterValue.GroupIdNull	

错误码	描述
InternalError.CloudApiProxyError	

部署容器应用

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

部署容器应用

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-02 21:01:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeployContainerGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID，分组唯一标识
Server	否	否	String	镜像server
Reponame	否	否	String	旧版镜像名，如/tsf/nginx
TagName	是	否	String	镜像版本名称,如v1
CpuLimit	否	否	String	业务容器最大的 CPU 核数，对应 K8S 的 limit；不填时默认为 request 的 2 倍
MemLimit	否	否	String	业务容器最大的内存 MiB 数，对应 K8S 的 limit；不填时默认为 request 的 2 倍
InstanceNum	是	否	Int64	实例数量
JvmOpts	否	否	String	jvm参数
CpuRequest	否	否	String	业务容器分配的 CPU 核数，对应 K8S 的 request
MemRequest	否	否	String	业务容器分配的内存 MiB 数，对应 K8S 的 request
DoNotStart	否	否	Bool	是否不立即启动
RepoName	否	否	String	（优先使用）新版镜像名，如/tsf/nginx

参数名称	必选	允许NULL	类型	描述
UpdateType	否	否	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	否	否	Int64	滚动更新必填，更新间隔
AgentCpuRequest	否	否	String	agent 容器分配的 CPU 核数，对应 K8S 的 request
AgentCpuLimit	否	否	String	agent 容器最大的 CPU 核数，对应 K8S 的 limit
AgentMemRequest	否	否	String	agent 容器分配的内存 MiB 数，对应 K8S 的 request
AgentMemLimit	否	否	String	agent 容器最大的内存 MiB 数，对应 K8S 的 limit
IstioCpuRequest	否	否	String	istioproxy 容器分配的 CPU 核数，对应 K8S 的 request
IstioCpuLimit	否	否	String	istioproxy 容器最大的 CPU 核数，对应 K8S 的 limit
IstioMemRequest	否	否	String	istioproxy 容器分配的内存 MiB 数，对应 K8S 的 request
IstioMemLimit	否	否	String	istioproxy 容器最大的内存 MiB 数，对应 K8S 的 limit
MaxSurge	否	否	String	kubernetes滚动更新策略的MaxSurge参数
MaxUnavailable	否	否	String	kubernetes滚动更新策略的MaxUnavailable参数
HealthCheckSettings	否	否	HealthCheckSettings	健康检查配置信息，若不指定该参数，则默认不设置健康检查。
Envs	否	否	Array of Env	部署组应用运行的环境变量。若不指定该参数，则默认不设置额外的环境变量。
ServiceSetting	否	否	ServiceSetting	容器部署组的网络设置。
DeployAgent	否	否	Bool	是否部署 agent 容器。若不指定该参数，则默认不部署 agent 容器。
SchedulingStrategy	否	否	SchedulingStrategy	节点调度策略。若不指定改参数，则默认不使用节点调度策略。
RepoType	否	否	String	仓库类型
SecretName	否	否	String	secret 名称
VolumeInfoList	否	否	Array of VolumeInfo	数据卷信息，list
VolumeMountInfoList	否	否	Array of VolumeMountInfo	数据卷挂载点信息
VolumeClean	否	否	Bool	是否清除数据卷信息
VolumeInfos	否	否	Array of VolumeInfo	数据卷信息，list
VolumeMountInfos	否	否	Array of VolumeMountInfo	数据卷挂载点信息

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 部署容器应用是否成功。 true：成功。 false：失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ContainergroupUpdateivlInvalid	
InvalidParameterValue.ContainergroupProtocolInvalid	
InternalError.ContainergroupKuberneteApiInvokeError	
InvalidParameterValue.ImagerepoReponameInvalid	
InvalidParameter.ParamError	
ResourceNotFound.ClusterNotExist	
InvalidParameter.KubernetesParamError	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	
FailedOperation.ClusterQueryFailed	
InternalError.UnhandledException	
InvalidParameterValue.ContainergroupAccesstypeNull	
InvalidParameterValue.ContainergroupCpulimitOver	
InvalidParameterValue.ContainergroupGroupidNull	
InvalidParameterValue.ContainergroupInvalidCpuInfo	
InvalidParameterValue.ContainergroupInvalidMemInfo	
InvalidParameterValue.ContainergroupMemlimitOver	
InvalidParameterValue.ContainergroupNodePortInvalid	
InvalidParameterValue.ContainergroupPortsRepeat	
InvalidParameterValue.ContainergroupProtocolPortsNull	

错误码	描述
InvalidParameterValue.ContainergroupReponameInvalid	
InvalidParameterValue.ContainergroupResourceAgentValueInvalid	
InvalidParameterValue.ContainergroupTargetPortsRepeat	
InvalidParameterValue.ImagerepoReponameNull	
InvalidParameterValue.ImagerepoTagnameNull	
ResourceNotFound.ContainergroupGroupNotFound	
InvalidParameterValue.ContainergroupYamlUserContainerNotFound	

部署虚拟机部署组应用

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

部署虚拟机部署组应用

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-27 10:42:28。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeployGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID
PkgId	是	否	String	程序包ID
StartupParameters	否	否	String	部署组启动参数
DeployDesc	否	否	String	部署应用描述信息
ForceStart	否	否	Bool	是否允许强制启动
EnableHealthCheck	否	否	Bool	是否开启健康检查
HealthCheckSettings	否	否	HealthCheckSettings	开启健康检查时，配置健康检查
UpdateType	否	否	Uint64	部署方式，0表示快速更新，1表示滚动更新
DeployBetaEnable	否	否	Bool	是否启用beta批次
DeployBatch	否	否	Array of Float	滚动发布每个批次参与的实例比率
DeployExeMode	否	否	String	滚动发布的执行方式
DeployWaitTime	否	否	Uint64	滚动发布每个批次的时间间隔
StartScript	否	否	String	启动脚本 base64编码
StopScript	否	否	String	停止脚本 base64编码

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.CvmCaeMasterAgentNotFound	
InvalidParameterValue.CvmCaeMasterAgentBusy	
InvalidParameter.RepoPackageParamError	

部署Serverless应用

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

部署Serverless应用

默认接口请求频率限制：20次/秒。

接口更新时间：2020-01-15 18:03:14。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeployServerlessGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID
PkgId	是	否	String	程序包ID
Memory	否	否	String	所需实例内存大小，取值为 1Gi 2Gi 4Gi 8Gi 16Gi，缺省为 1Gi，不传表示维持原态
InstanceRequest	否	否	Uint64	要求最小实例数，取值范围 [1, 4]，缺省为 1，不传表示维持原态
StartupParameters	否	否	String	部署组启动参数，不传表示维持原态

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 结果true：成功；false：失败；
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.GroupNotExist	
InvalidParameterValue.GroupIdNull	
ResourceInsufficient.InstanceExcessLimit	
InvalidParameterValue.CvmCaeMasterAgentNotFound	
InvalidParameterValue.CvmCaeMasterAgentBusy	
InternalError.UnhandledException	
ResourceInUse.GroupInOperation	

查询容器部署组详情

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

容器部署组详情

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:02:17。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeContainerGroupDetail
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	分组ID

3. 输出参数

参数名称	类型	描述
Result	ContainerGroupDetail	**此参数对外不可见。** 容器部署组详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ContainergroupGroupNamespaceClusterNotFound	
InternalError.ContainergroupKuberneteConnectError	

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	
UnauthorizedOperation.NoPrivilege	
InternalError.UnhandledException	
InvalidParameterValue.ContainergroupGroupidNull	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	

容器部署组列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

容器部署组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-09-26 16:29:34。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeContainerGroups
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段，模糊搜索groupName字段
ApplicationId	是	否	String	分组所属应用ID
OrderBy	否	否	String	排序字段，默认为 createTime字段，支持id， name， createTime
OrderType	否	否	Int64	排序方式，默认为1：倒序排序，0：正序，1：倒序
Offset	否	否	Int64	偏移量，取值从0开始
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~50
ClusterId	否	否	String	集群ID
NamespaceId	否	否	String	命名空间 ID

3. 输出参数

参数名称	类型	描述
Result	ContainGroupResult	**此参数对外不可见。** 查询的权限数据对象

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ContainergroupApplicationIdNull	
ResourceNotFound.MicroserviceOffline	

查询虚拟机部署组详情

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询虚拟机部署组详情

默认接口请求频率限制：20次/秒。

接口更新时间：2020-10-21 23:13:20。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	VmGroup	**此参数对外不可见。** 虚拟机部署组详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	

错误码	描述
UnauthorizedOperation.NoPrivilege	
InternalError.GroupCommonError	
InternalError.UnhandledException	
MissingParameter.GroupIdNull	
ResourceNotFound.MicroserviceOffline	
InvalidParameter.RepoPackageParamError	

查询虚拟机部署组云主机列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询虚拟机部署组云主机列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:06:58。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroupInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段
GroupId	是	否	String	部署组ID
OrderBy	否	否	String	排序字段
OrderType	否	否	Int64	排序类型
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数

3. 输出参数

参数名称	类型	描述
Result	TsfPageInstance	**此参数对外不可见。** 部署组机器信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
InternalError.CloudApiProxyError	
InternalError.InstanceCommonError	
MissingParameter.GroupIdNull	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.CamGeneralError	

获取虚拟机部署组列表

最近更新时间: 2025-02-18 17:50:45

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取虚拟机部署组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-12-18 19:38:26。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeGroups
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段
ApplicationId	否	否	String	应用ID
OrderBy	否	否	String	排序字段
OrderType	否	否	Int64	排序方式
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数
NamespaceId	否	否	String	命名空间ID
ClusterId	否	否	String	集群ID
GroupResourceTypeList	否	否	Array of String	部署组资源类型列表
Status	否	否	String	部署组状态过滤字段

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageVmGroup	**此参数对外不可见。** 虚拟机部署组分页信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.ApplicationQueryFailed	
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
InvalidParameterValue.GroupPageLimitInvalid	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	
InvalidParameterValue.GroupStatusInvalid	

获取凭证列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取凭证列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-05-12 15:40:45。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSecretList
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组 ID
SecretName	否	否	String	凭证名称，如有，则为指定名称查询

3. 输出参数

参数名称	类型	描述
Result	ImageRepositorySecret	**此参数对外不可见。** 凭证数据列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取凭证名称列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

获取凭证名称列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-05-12 15:41:01。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSecretNames
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组 ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 凭证名称数据列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询Serverless部署组明细

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询Serverless部署组明细

默认接口请求频率限制：20次/秒。

接口更新时间：2019-11-18 10:46:00。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeServerlessGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	ServerlessGroup	**此参数对外不可见。** 结果
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InvalidParameterValue	

错误码	描述
ResourceNotFound.GroupNotExist	
InvalidParameterValue.GroupIdNull	
InternalError.UnhandledException	
FailedOperation.TsfAsResourceServerError	

查询Serverless部署组列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询Serverless部署组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-12-28 17:41:17。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeServerlessGroups
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段，模糊搜索groupName字段
ApplicationId	否	否	String	分组所属应用ID
OrderBy	否	否	String	排序字段，默认为 createTime字段，支持id， name， createTime
OrderType	否	否	String	排序方式，默认为1：倒序排序，0：正序，1：倒序
Offset	否	否	Uint64	偏移量，取值从0开始
Limit	否	否	Uint64	分页个数，默认为20， 取值应为1~50
NamespaceId	否	否	String	分组所属名字空间ID
ClusterId	否	否	String	分组所属集群ID

3. 输出参数

参数名称	类型	描述
Result	ServerlessGroupPage	**此参数对外不可见。** 数据列表对象

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue	
InvalidParameterValue.ApplicationIdNull	

查询简单部署组列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询简单部署组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-12-17 11:57:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSimpleGroups
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupIdList	否	否	Array of String	部署组ID列表，不填写时查询全量
ApplicationId	否	否	String	应用ID，不填写时查询全量
ClusterId	否	否	String	集群ID，不填写时查询全量
NamespaceId	否	否	String	命名空间ID，不填写时查询全量
Limit	否	否	Int64	每页条数
Offset	否	否	Int64	起始偏移量
GroupId	否	否	String	部署组ID，不填写时查询全量
SearchWord	否	否	String	模糊查询，部署组名称，不填写时查询全量
AppMicroServiceType	否	否	String	部署组类型，精确过滤字段，M：service mesh, P：原生应用，G：网关应用

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageSimpleGroup	**此参数对外不可见。** 简单部署组列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
InvalidParameter.BadRequest	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.NoLicense	

虚拟机部署组添加实例

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

虚拟机部署组添加实例

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:07:53。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ExpandGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID
InstanceIdList	是	否	Array of String	扩容的机器实例ID列表

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.GroupNotExist	

错误码	描述
InvalidParameterValue.CvmCaeMasterAgentBusy	
InvalidParameter.CvmCaeMasterUnknownInstanceStatus	

修改容器部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

修改容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-06-18 16:44:33。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyContainerGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	否	否	String	部署组ID
AccessType	否	否	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	否	否	Array of ProtocolPort	ProtocolPorts数组
UpdateType	否	否	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	否	否	Int64	更新间隔,单位秒
SubnetId	否	否	String	子网ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 更新部署组是否成功。 true：成功。 false：失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ContainergroupAccesstypeNull	
InvalidParameterValue.ContainergroupGroupidNull	

修改容器部署组实例数

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

修改容器部署组实例数

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:04:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyContainerReplicas
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID，部署组唯一标识
InstanceNum	是	否	Int64	实例数量

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 结果true：成功；false：失败；
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteApiInvokeError	

错误码	描述
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ContainergroupCpulimitOver	
InvalidParameterValue.ContainergroupMemlimitOver	
ResourceNotFound.ContainergroupGroupNotFound	

缩容虚拟机部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

下线部署组所有机器实例

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:08:11。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ShrinkGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

虚拟机部署组下线实例

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

虚拟机部署组下线实例

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:08:25。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ShrinkInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID
InstanceIdList	是	否	Array of String	下线机器实例ID列表

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.GroupNotExist	

错误码	描述
InvalidParameterValue.CvmCaeMasterAgentBusy	
InternalError.CvmCaeMasterDispatchError	

启动容器部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启动容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:04:10。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StartContainerGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 启动操作是否成功。 true：启动成功 false：启动失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteApiInvokeError	
InvalidParameter.KubernetesParamError	

错误码	描述
UnauthorizedOperation.NoPrivilege	
FailedOperation.ContainergroupGroupHasrun	

启动虚拟机部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

启动分组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:07:27。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StartGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	

错误码	描述
InvalidParameterValue.CvmCaeMasterAgentBusy	
InternalError.CvmCaeMasterDispatchError	

停止容器部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停止容器部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2020-07-27 10:54:34。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StopContainerGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 停止操作是否成功。 true：停止成功 false：停止失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteApiInvokeError	
UnauthorizedOperation.NoPrivilege	

错误码	描述
FailedOperation.ContainergroupGroupHasstop	
ResourceNotFound.ContainergroupGroupNotFound	

停止虚拟机部署组

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

停止虚拟机部署组

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 22:07:38。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：StopGroup
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	TaskId	**此参数对外不可见。** 任务ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.GroupNotExist	

错误码	描述
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.CvmCaeMasterAgentBusy	
InvalidParameterValue.CvmCaeMasterGroupNoAgent	

更新健康检查配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

更新健康检查配置

默认接口请求频率限制：20次/秒。

接口更新时间：2020-10-20 15:35:33。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateHealthCheckSettings
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID
EnableHealthCheck	否	否	Bool	是否能使健康检查
HealthCheckSettings	否	否	HealthCheckSettings	健康检查配置

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 更新健康检查配置操作是否成功。 true：操作成功。 false：操作失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
FailedOperation.CvmCaeMasterHealthCheckConfigError	

配置管理相关接口
创建配置项

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建配置项

默认接口请求频率限制：20次/秒。

接口更新时间：2022-09-27 15:06:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigName	是	否	String	配置项名称
ConfigVersion	是	否	String	配置项版本
ConfigVersionDesc	否	否	String	配置项版本描述
ConfigValue	是	否	String	配置项值
ApplicationId	是	否	String	应用ID
ConfigType	否	否	String	配置项值类型
EncodeWithBase64	否	否	Bool	Base64编码的配置项(使用base64编码时该项为必填项)

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：创建成功；false：创建失败

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
MissingParameter.ConfigNameRequired	
InvalidParameterValue.ConfigNameInvalid	
MissingParameter.ConfigTypeRequired	
MissingParameter.ApplicationIdRequired	
MissingParameter.ConfigVersionRequired	
InvalidParameterValue.ConfigVersionInvalid	
InvalidParameterValue.ConfigVersionDescInvalid	
MissingParameter.ConfigValueRequired	
InvalidParameterValue.ConfigValueTooLong	
InvalidParameterValue.ConfigValueFormatInvalid	
InvalidParameterValue.ConfigExists	
FailedOperation.ConfigCreateFailed	
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ApplicationNotExists	
InvalidParameterValue.ResourcePermissionDenied	

创建公共配置项

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建公共配置项

默认接口请求频率限制：20次/秒。

接口更新时间：2022-09-27 15:07:06。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreatePublicConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigName	是	否	String	配置项名称
ConfigVersion	是	否	String	配置项版本
ConfigVersionDesc	否	否	String	配置项版本描述
ConfigValue	是	否	String	配置项值，总是接收yaml格式的内容
ConfigType	否	否	String	配置项类型
EncodeWithBase64	否	否	Bool	Base64编码的配置项(使用base64编码时该项为必填项)

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：创建成功；false：创建失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ConfigNameInvalid	
InvalidParameterValue.ConfigVersionInvalid	
InvalidParameterValue.ConfigValueFormatInvalid	
InvalidParameterValue.ConfigExists	
UnauthorizedOperation.NoPrivilege	

删除配置项

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除配置项

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:50:19。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	是	否	String	配置项ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：删除成功；false：删除失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ConfigNotExistsOrPermissionDenied	

错误码	描述
InvalidParameterValue.ReleasedConfigCanNotBeDeleted	

删除公共配置项

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

删除公共配置项

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:44:31。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeletePublicConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	是	否	String	配置项ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：删除成功；false：删除失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ConfigNotExistsOrPermissionDenied	
InvalidParameterValue.ReleasedConfigCanNotBeDeleted	

错误码	描述
MissingParameter.ConfigIdRequired	

查询配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询配置

默认接口请求频率限制：50次/秒。

接口更新时间：2022-09-26 14:59:21。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	否	否	String	配置项ID

3. 输出参数

参数名称	类型	描述
Result	Config	**此参数对外不可见。** 配置项
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.ConfigApplicationQueryFailed	

查询配置发布历史

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询配置发布历史

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:48:45。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeConfigReleaseLogs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	否	否	String	部署组ID，不传入时查询全量
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20
NamespaceId	否	否	String	命名空间ID，不传入时查询全量
ClusterId	否	否	String	集群ID，不传入时查询全量
ApplicationId	否	否	String	应用ID，不传入时查询全量

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfigReleaseLog	**此参数对外不可见。** 分页的配置项发布历史列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询配置发布信息

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询配置发布信息

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:48:31。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeConfigReleases
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigName	否	否	String	配置项名称，不传入时查询全量
GroupId	否	否	String	部署组ID，不传入时查询全量
NamespaceId	否	否	String	命名空间ID，不传入时查询全量
ClusterId	否	否	String	集群ID，不传入时查询全量
Limit	否	否	Int64	每页条数
Offset	否	否	Int64	偏移量
ConfigId	否	否	String	配置ID，不传入时查询全量
ApplicationId	否	否	String	应用ID，不传入时查询全量

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfigRelease	**此参数对外不可见。** 分页的配置发布信息

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
UnauthorizedOperation.NoLicense	
FailedOperation.ConfigGroupQueryFailed	

查询配置汇总列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询配置汇总列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:47:45。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeConfigSummary
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	否	否	String	应用ID，不传入时查询全量
SearchWord	否	否	String	查询关键字，模糊查询：应用名称，配置项名称，不传入时查询全量
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfig	**此参数对外不可见。** 配置项分页对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
FailedOperation.ConfigApplicationQueryFailed	

查询配置项列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询配置项列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-03 19:21:57。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeConfigs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	否	否	String	应用ID，不传入时查询全量
ConfigId	否	否	String	配置项ID，不传入时查询全量，高优先级
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	每页条数
ConfigIdList	否	否	Array of String	配置项ID列表，不传入时查询全量，低优先级
ConfigName	否	否	String	配置项名称，精确查询，不传入时查询全量
ConfigVersion	否	否	String	配置项版本，精确查询，不传入时查询全量
OrderBy	否	否	String	OrderBy值
OrderType	否	否	Int64	OrderType值

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	TsfPageConfig	**此参数对外不可见。** 分页后的配置项列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.ConfigApplicationQueryFailed	
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	

查询公共配置（单条）

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询公共配置（单条）

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:42:41。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePublicConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	是	否	String	需要查询的配置项ID

3. 输出参数

参数名称	类型	描述
Result	Config	**此参数对外不可见。** 全局配置
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.TsfPrivilegeError	
UnauthorizedOperation.NoLicense	

查询公共配置发布历史

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询公共配置发布历史

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:43:38。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePublicConfigReleaseLogs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
NamespaceId	否	否	String	命名空间ID，不传入时查询全量
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfigReleaseLog	**此参数对外不可见。** 分页后的公共配置项发布历史列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询公共配置发布信息

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询公共配置发布信息

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:43:24。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePublicConfigReleases
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigName	否	否	String	配置项名称，不传入时查询全量
NamespaceId	否	否	String	命名空间ID，不传入时查询全量
Limit	否	否	Int64	每页条数
Offset	否	否	Int64	偏移量
ConfigId	否	否	String	配置项ID，不传入时查询全量

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfigRelease	**此参数对外不可见。** 公共配置发布信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.TsfPrivilegeError	
InternalError.UnhandledException	
UnauthorizedOperation.NoLicense	
FailedOperation.ConfigNamespaceQueryFailed	

查询公共配置汇总列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询公共配置汇总列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:43:11。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePublicConfigSummary
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	查询关键字，模糊查询：配置项名称，不传入时查询全量
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfig	**此参数对外不可见。** 分页的全局配置统计信息列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	

查询公共配置项列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询公共配置项列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-05-17 16:09:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePublicConfigs
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	否	否	String	配置项ID，不传入时查询全量，高优先级
Offset	否	否	Int64	偏移量，默认为0
Limit	否	否	Int64	每页条数，默认为20
ConfigIdList	否	否	Array of String	配置项ID列表，不传入时查询全量，低优先级
ConfigName	否	否	String	配置项名称，精确查询，不传入时查询全量
ConfigVersion	否	否	String	配置项版本，精确查询，不传入时查询全量
OrderBy	否	否	String	排序字段
OrderType	否	否	String	排序类型

3. 输出参数

参数名称	类型	描述
Result	TsfPageConfig	**此参数对外不可见。** 分页后的全局配置项列表

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	

查询group发布的配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询group发布的配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:46:42。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeReleasedConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
GroupId	是	否	String	部署组ID

3. 输出参数

参数名称	类型	描述
Result	String	**此参数对外不可见。** 已发布的配置内容
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
UnauthorizedOperation.NoPrivilege	
InternalError.UnhandledException	

错误码	描述
InvalidParameterValue.GroupNotExists	
InvalidParameterValue.ResourcePermissionDenied	

发布配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

发布配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:46:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ReleaseConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	是	否	String	配置ID
GroupId	是	否	String	部署组ID
ReleaseDesc	否	否	String	发布描述

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：发布成功；false：发布失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.ConfigNotExistsOrPermissionDenied	
InternalError.UnhandledException	
InvalidParameterValue.ConfigAlreadyReleased	
InvalidParameterValue.ConfigGroupApplicationIdNotMatch	
InvalidParameterValue.GroupNotExists	
InvalidParameterValue.ResourcePermissionDenied	
InternalError.ConsulServerError	

发布公共配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

发布公共配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:42:03。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ReleasePublicConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigId	是	否	String	配置ID
NamespaceId	是	否	String	命名空间ID
ReleaseDesc	否	否	String	发布描述

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：发布成功；false：发布失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameterValue.ConfigValueTooLong	
InvalidParameter.ParamError	
InvalidParameterValue.ConfigNotExistsOrPermissionDenied	
InvalidParameterValue.ConfigAlreadyReleased	
InvalidParameterValue.ResourcePermissionDenied	

撤回已发布的配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

撤回已发布的配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:45:41。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RevocationConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigReleaseId	是	否	String	配置项发布ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：回滚成功；false：回滚失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ConfigReleaseNotExists	
InvalidParameterValue.ResourcePermissionDenied	

错误码	描述
InternalError.ConsulServerError	
MissingParameter.ConfigReleaseIdRequired	

撤回已发布的公共配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

撤回已发布的公共配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-10 17:41:34。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RevocationPublicConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigReleaseId	是	否	String	配置项发布ID

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：撤销成功；false：撤销失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.ConfigReleaseNotExists	

回滚配置

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

回滚配置

默认接口请求频率限制：20次/秒。

接口更新时间：2019-10-12 17:54:32。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RollbackConfig
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ConfigReleaseLogId	是	否	String	配置项发布历史ID
ReleaseDesc	否	否	String	回滚描述

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** true：回滚成功；false：回滚失败
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.GroupNotExists	

错误码	描述
InternalError.ConsulServerError	

镜像相关接口
批量删除镜像版本

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

批量删除镜像版本

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:47:20。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteImageTags
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ImageTags	是	否	Array of DeleteImageTag	镜像版本数组

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 批量删除操作是否成功。 true：成功。 false：失败。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

镜像仓库列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

镜像仓库列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-10-29 10:29:19。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeImageRepository
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	仓库名，搜索关键字,不带命名空间的
Offset	否	否	Int64	偏移量，取值从0开始
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~100

3. 输出参数

参数名称	类型	描述
Result	ImageRepositoryResult	**此参数对外不可见。** 查询的权限数据对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
UnauthorizedOperation.NoPrivilege	
InternalError.CloudApiProxyError	
InternalError.UnhandledException	
ResourceNotFound.ErrNoUser	
UnauthorizedOperation.CamGeneralError	

镜像版本列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

镜像版本列表

默认接口请求频率限制：20次/秒。

接口更新时间：2020-03-12 15:26:26。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeImageTags
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ApplicationId	是	否	String	应用Id
Offset	否	否	Int64	偏移量，取值从0开始
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~100
QueryImageIdFlag	否	否	Int64	不填和0:查询 1:不查询
SearchWord	否	否	String	可用于搜索的 tag 名字

3. 输出参数

参数名称	类型	描述
Result	ImageTagsResult	**此参数对外不可见。** 查询的权限数据对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ErrNoRepo	
UnauthorizedOperation.NoPrivilege	
FailedOperation.ApplicationQueryFailed	
InternalError.CamRoleRequestError	
InternalError.CloudApiProxyError	
InternalError.UnhandledException	
ResourceNotFound.GroupApplicationNotExist	
ResourceNotFound.MicroserviceOffline	
UnauthorizedOperation.CamGeneralError	

集群相关接口
集群添加云主机

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

添加云主机节点至TSF集群

默认接口请求频率限制：20次/秒。

接口更新时间：2022-01-06 19:36:38。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddClusterInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterId	是	否	String	集群ID
InstanceIdList	是	否	Array of String	云主机ID列表
OsName	否	否	String	操作系统名称
ImageId	否	否	String	操作系统镜像ID
Password	否	否	String	重装系统密码设置
KeyId	否	否	String	重装系统，关联密钥设置
SgId	否	否	String	安全组设置
InstanceImportMode	否	否	String	云主机导入方式，虚拟机集群必填，容器集群不填写此字段，R：重装TSF系统镜像，M：手动安装agent
OsCustomizeType	否	否	String	镜像定制类型
FeatureIdList	否	否	Array of String	镜像特征ID列表

参数名称	必选	允许NULL	类型	描述
InstanceAdvancedSettings	否	否	InstanceAdvancedSettings	实例额外需要设置参数信息
SecurityGroupIds	否	否	Array of String	部署组ID
InstanceCpuType	否	否	String	实例cpu架构

3. 输出参数

参数名称	类型	描述
Result	AddInstanceResult	**此参数对外不可见。** 添加云主机的返回列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ClusterNotExist	
FailedOperation.InstanceResetError	
FailedOperation.InstanceUpdateFailed	
UnauthorizedOperation.NoPrivilege	
InternalError.CpClusterUnavailable	
FailedOperation.InstanceResetTimeout	
FailedOperation.TkeClusterQueryFailed	
InternalError.CloudApiProxyError	
InternalError.TkeApiFailedOperation	
ResourceInUse.InstanceHasBeenUsed	
ResourceNotFound.InstanceNotExist	
ResourceNotFound.TkeClusterNotExists	
UnauthorizedOperation.CamGeneralError	

集群导入云主机

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

添加云主机节点至TSF集群

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-09 17:47:40。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterId	是	否	String	集群ID
InstanceIdList	是	否	Array of String	云主机ID列表
OsName	否	否	String	操作系统名称
ImageId	否	否	String	操作系统镜像ID
Password	否	否	String	重装系统密码设置
KeyId	否	否	String	重装系统，关联密钥设置
SgId	否	否	String	安全组设置
InstanceImportMode	否	否	String	云主机导入方式，虚拟机集群必填，容器集群不填写此字段，R：重装TSF系统镜像，M：手动安装agent

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 添加云主机是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue	
ResourceNotFound.ClusterNotExist	

创建集群

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

创建集群

默认接口请求频率限制：20次/秒。

接口更新时间：2023-06-02 20:32:08。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateCluster
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterName	是	否	String	集群名称
ClusterType	是	否	String	集群类型
VpcId	是	否	String	私有网络ID
ClusterCIDR	否	否	String	分配给集群容器和服务IP的CIDR
ClusterDesc	否	否	String	集群备注
TsfRegionId	否	否	String	集群所属TSF地域
TsfZoneId	否	否	String	集群所属TSF可用区
SubnetId	否	否	String	私有网络子网ID
ClusterVersion	否	否	String	集群版本
MaxNodePodNum	否	否	Uint64	集群中每个Node上最大的Pod数量。取值范围4～256。不为2的幂值时会向上取最接近的2的幂值。
MaxClusterServiceNum	否	否	Uint64	集群最大的service数量。取值范围32～32768，不为2的幂值时会向上取最接近的2的幂值。
ProgramId	否	否	String	需要绑定的数据集ID

参数名称	必选	允许NULL	类型	描述
ClusterCpuType	否	否	String	集群cpu架构
SecurityGroupId	否	是	String	安全组

3. 输出参数

参数名称	类型	描述
Result	String	集群ID
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
ResourceNotFound.ClusterNotExist	
FailedOperation.TkeClusterCreateFailed	
InvalidParameterValue.ClusterNameExist	
UnauthorizedOperation.NoPrivilege	
FailedOperation.ClusterQueryFailed	
InternalError.CloudApiProxyError	
InternalError.ClusterCommonError	
InternalError.UnhandledException	
InvalidParameterValue.ClusterCidrConflict	
InvalidParameterValue.ClusterRegionInvalid	
InvalidParameterValue.ClusterTypeInvalid	
InvalidParameterValue.ClusterZoneInvalid	
LimitExceeded.TkeClusterNumberExceedLimit	
MissingParameter.ClusterSubnetRequired	
ResourceNotFound.ClusterVpcNotExist	
UnauthorizedOperation.CamGeneralError	

查询集群实例

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询集群实例

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-02 21:21:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeClusterInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SearchWord	否	否	String	搜索字段
ClusterId	是	否	String	集群ID
OrderBy	否	否	String	排序字段
OrderType	否	否	Int64	排序类型
Offset	否	否	Int64	偏移量
Limit	否	否	Int64	分页个数

3. 输出参数

参数名称	类型	描述
Result	TsfPageInstance	**此参数对外不可见。** 集群机器实例分页信息
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.ContainergroupKuberneteConnectError	
InvalidParameter.ParamError	
ResourceNotFound.ClusterNotExist	
FailedOperation.InstanceUpdateFailed	
UnauthorizedOperation.NoPrivilege	
FailedOperation.TsfPrivilegeError	
InternalError.CloudApiProxyError	
InternalError.UnhandledException	
ResourceNotFound.MicroserviceOffline	

查询简单集群列表

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

查询简单集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-10 22:21:07。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSimpleClusters
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterIdList	否	否	Array of String	需要查询的集群ID列表，不填或不传入时查询所有内容
ClusterType	否	否	String	需要查询的集群类型，不填或不传入时查询所有内容
Offset	否	否	Int64	查询偏移量，默认为0
Limit	否	否	Int64	分页个数，默认为20， 取值应为1~50
SearchWord	否	否	String	对id和name进行关键词过滤

3. 输出参数

参数名称	类型	描述
Result	TsfPageCluster	**此参数对外不可见。** TSF集群分页对象
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	
UnauthorizedOperation.NoPrivilege	
ResourceNotFound.MicroserviceOffline	

移除云主机

最近更新时间: 2025-02-18 17:50:46

1. 接口描述

接口请求域名： tsf.api3.finance.cloud.tencent.com。

从 TSF 集群中批量移除云主机节点

默认接口请求频率限制：20次/秒。

接口更新时间：2019-07-04 21:04:02。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RemoveInstances
Version	是	否	String	公共参数，本接口取值：2018-03-26
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
ClusterId	是	否	String	集群 ID
InstanceIdList	是	否	Array of String	云主机 ID 列表

3. 输出参数

参数名称	类型	描述
Result	Bool	**此参数对外不可见。** 集群移除机器是否成功
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.ClusterNotExist	

错误码	描述
UnauthorizedOperation.NoPrivilege	
InvalidParameterValue.CvmCaeMasterAgentBusy	
InternalError.TkeApiFailedOperation	
ResourceNotFound.InstanceNotExist	

数据结构

最近更新时间: 2025-02-18 17:50:46

TagRouteItemV2

标签路由规则

被如下接口引用：

名称	必选	允许NULL	类型	描述
TagRouteId	否	否	String	标签路由规则项ID
TargetField	是	否	String	标签路由匹配目标字段
TargetValue	是	否	String	标签路由匹配目标取值
RouteRuleId	否	否	String	标签路由所属路由ID
SourceList	是	否	Array of TagRouteItemSourceV2	标签路由源标签列表

TaskIdV2

任务id

被如下接口引用：

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	任务Id

PagedDtsTransaction

DTS主事务分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数，特定在该接口中总是会返回null
Content	是	是	Array of DtsTransaction	主事务分组列表

MonitorStatisticsPolicyResultV2

MonitorStatisticsPolicyResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of MonitorStatisticsPolicy	Content

ApiDailyUseStatisticsEntity

ApiDailyUseStatisticsEntity

被如下接口引用：DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
Name	是	是	String	name
Count	是	是	String	count
Ratio	是	是	String	ratio

PagedPermCat

tsf-privilege模块，分页权限组列表

被如下接口引用：DescribePermissionCategories

名称	必选	允许NULL	类型	描述
TotalCount	否	否	Int64	总条数
Content	是	否	Array of PermCat	产品列表

TsfPageVmGroup

列表中部署组分页信息

被如下接口引用：DescribeGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	虚拟机部署组总数目
Content	是	是	Array of VmGroupSimple	虚拟机部署组列表信息

MicroserviceWrong

微服务

被如下接口引用：

名称	必选	允许NULL	类型	描述
MicroserviceId	否	是	String	微服务Id
MicroserviceName	否	是	String	名称
MicroserviceDesc	否	是	String	描述
MicroserviceType	否	是	String	类型
AssociateApplicationId	否	是	String	关联的应用 ID
MicroserviceProtocol	否	是	String	协议
MicroserviceListeningPort	否	是	Int64	监听端口
HealthCheckUrl	否	是	String	健康检查参数
ServiceClustertype	否	是	String	集群类型
CreateTime	否	是	Int64	创建时间
UpdateTime	否	是	Int64	更新时间
NamespaceId	否	是	String	命名空间Id
RunInstanceCount	否	是	Int64	微服务的运行实例数目
AssociateApplication	否	是	MsApplication	关联的应用详情

MsApiArray

微服务API数组

被如下接口引用：DescribeInterfaceList、DescribeMsApiList

名称	必选	允许NULL	类型	描述
Path	是	否	String	API 请求路径
Method	是	否	String	请求方法
Description	是	是	String	方法描述
Status	是	是	Int64	API状态 0:离线 1:在线

BusinessLogConfigSchemaV2

业务日志配置解析规则

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
SchemaType	是	否	Int64	解析规则类型
SchemaContent	否	是	String	解析规则内容
SchemaDateFormat	否	是	String	解析规则时间格式
SchemaMultilinePattern	否	是	String	解析规则对应的多行匹配规则
SchemaCreateTime	否	是	String	解析规则创建时间
SchemaPatternLayout	否	是	String	解析规则内容，包含解析规则时间格式和解析规则内容
SchemaLogMessage	否	是	String	模拟解析的日志内容

GraphEdge

依赖拓扑图边

被如下接口引用：

名称	必选	允许NULL	类型	描述
SourceName	是	是	String	来源服务名
TargetName	是	是	String	目标服务名
SourceType	是	是	String	来源服务类型
TargetType	是	是	String	目标服务类型
ReqTotalQty	是	是	Uint64	服务间总请求量
ReqSuccessQty	是	是	Uint64	服务间成功请求量
ReqFailedQty	是	是	Uint64	服务间失败请求量
ReqPerMin	是	是	Float	服务间每分钟平均请求量
AvgDurationMs	是	是	Float	服务间平均耗时（毫秒）

GroupsByScalableRuleId

GroupsByScalableRuleId

被如下接口引用：DescribeScalableRuleAttributeByGroup、ListGroupsByScalableRuleId

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	ApplicationId
ApplicationName	是	是	String	ApplicationName

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	GroupId
GroupName	是	是	String	GroupName
Status	是	是	Int64	Status
UpdateTime	是	是	String	UpdateTime
RuleId	是	是	String	弹性伸缩规则id
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
RuleName	是	是	String	弹性伸缩规则名

MsInstance

微服务实例信息

被如下接口引用：DescribeMicroservice

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器实例ID信息
InstanceName	是	是	String	机器实例名称信息
Port	是	是	String	服务运行的端口号
LanIp	是	是	String	机器实例内网IP
WanIp	是	是	String	机器实例外网IP
InstanceAvailableStatus	是	是	String	机器可用状态
ServiceInstanceStatus	是	是	String	服务运行状态
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间ID

名称	必选	允许NULL	类型	描述
NamespaceName	是	是	String	命名空间名称
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
InstanceStatus	是	是	String	机器TSF可用状态
HealthCheckUrl	是	是	String	健康检查URL
ClusterType	是	是	String	集群类型
ApplicationPackageVersion	是	是	String	应用程序包版本
ApplicationType	是	是	String	应用类型
ServiceStatus	是	是	String	服务状态，passing 在线，critical 离线
RegistrationTime	是	是	Int64	注册时间
LastHeartbeatTime	是	是	Int64	上次心跳时间
RegistrationId	是	是	String	实例注册id
HiddenStatus	是	是	String	屏蔽状态，hidden 为屏蔽，unhidden 为未屏蔽

FindContainerGroupResult

1.16 容器部署组详情(FindContainerGroup) Result

被如下接口引用：FindContainerGroup

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	groupId
GroupName	是	是	String	分组名称
InstanceCount	是	是	Int64	实例总数
RunInstanceCount	是	是	Int64	已启动实例总数
CreateTime	是	是	String	创建时间
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
LbIp	是	是	String	负载均衡ip
ClusterIp	是	是	String	Service ip
NodePort	是	是	String	NodePort端口，只有公网和NodePort访问方式才有值
CpuLimit	是	是	String	最大分配cpu 核数，如0.6
MemLimit	是	是	String	最大分配内存M数
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
Envs	是	是	Array of Env	环境变量数组
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id

ApiDetailResponse

ApiDetailResponse描述

被如下接口引用：DescribeApiDetail

名称	必选	允许NULL	类型	描述
Request	是	否	Array of ApiRequestDescr	API 请求参数
Response	是	是	Array of ApiResponseDescr	API 响应参数
Definitions	是	否	Array of ApiDefinitionDescr	API 复杂结构定义
RequestContentType	是	是	String	API 的 content type
CanRun	是	是	Bool	API 能否调试

名称	必选	允许NULL	类型	描述
Status	是	是	Int64	API 状态 0:离线 1:在线，默认0
Description	是	是	String	API 描述

ServerlessGroupPage

ServerlessGroup 翻页对象

被如下接口引用：DescribeServerlessGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总记录数
Content	是	是	Array of ServerlessGroup	列表信息

ConfigTemplateV2

配置模板对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigTemplateId	否	是	String	配置模板Id
ConfigTemplateName	否	是	String	配置模板名称
ConfigTemplateDesc	否	是	String	配置模板描述
ConfigTemplateType	否	是	String	配置模板对应的微服务框架
ConfigTemplateValue	否	是	String	配置模板数据
CreateTime	否	是	String	创建时间
UpdateTime	否	是	String	更新时间

DescribeTransactionsResp

查询主事务列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	返回的事务数量
Content	是	是	Array of TxMainTransaction	主事务信息

ResourceGetBatchIndexResult

获取执行批次索引的描述

被如下接口引用：GetResourceBatchIndex

名称	必选	允许NULL	类型	描述
BatchNum	是	否	Uint64	执行批次索引

DescribeResourceConfigLicenseResource

DescribeResourceConfig

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Name	是	是	String	Name
Quota	是	是	Uint64	Quota

RouteRule

路由规则信息

被如下接口引用：DescribeRouteReleaseHistory、DescribeRouteRule、DescribeRouteRules

名称	必选	允许NULL	类型	描述
RouteRuleId	是	是	String	路由规则Id
RouteRuleName	是	是	String	路由规则名称
RouteRuleType	是	是	String	路由规则类型，包括标签路由和权重标签，标签路由：T，权重路由： W
RouteRuleDesc	是	是	String	路由规则描述信息
MicroserviceId	是	是	String	微服务Id
EnableStatus	是	是	Bool	true/false，true：路由规则为启用状态， false：路由规则为停用状态
CreateTime	是	是	String	路由规则创建时间
UpdateTime	是	是	String	路由规则更新时间
TagRouteItemList	是	是	Array of TagRouteItemList	标签路由规则项信息
WeightRouteItemList	是	是	Array of WeightRouteItemList	权重路由规则项信息

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	微服务所属命名空间Id
MicroserviceName	是	是	String	微服务名称

OverviewGroupResourceUsage

部署组统计信息

被如下接口引用：DescribeClusterGroupCount、DescribeGroupResourceUsage

名称	必选	允许NULL	类型	描述
GroupCount	是	否	Uint64	部署组总数
RunGroupCount	是	否	Uint64	运行部署组总数
StopGroupCount	是	否	Uint64	停止中部署组总数
AbnormalGroupCount	是	否	Uint64	异常部署组总数

StatisticsEntryV2

概览页统计列表条目

被如下接口引用：

名称	必选	允许NULL	类型	描述
PrimeKey	是	是	String	主字段
SubKey	是	是	String	附字段
Value	是	是	String	值

TsfPageConfigRelease

TSF配置项发布信息分页对象

被如下接口引用：DescribeConfigReleases、DescribePublicConfigReleases

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of ConfigRelease	配置项发布信息数组

BillingLicense

计费许可

被如下接口引用：DescribeAllBillingLicenses

名称	必选	允许NULL	类型	描述
Id	是	否	String	许可ID
AppId	是	否	String	租户ID
Uin	是	是	String	账号ID
SpecType	是	是	Int64	规格类型
Spec	是	是	String	规格
NodeSize	是	是	Int64	节点数
Resource	是	是	String	资源
Function	是	是	String	功能
Status	是	是	Int64	状态
SourceType	是	是	Int64	来源类型
AutoRenewFlag	是	是	Int64	自动续费标记
CreateTime	是	是	String	创建时间
ModifyTime	是	是	String	修改时间
IsolateTime	是	是	String	隔离时间
WhitelistType	是	是	Int64	白名单类型，0 正常，1 不收费白名单内
Nickname	是	是	String	昵称
ExpireTime	是	是	String	过期时间
BillingRecordCount	是	是	Int64	订单数

MetricDimension

指标维度

被如下接口引用：DescribeInvocationMetricDataCurve

名称	必选	允许NULL	类型	描述
Name	否	是	String	指标维度名称
Value	否	是	String	指标维度取值

SimpleApplicationV2

简单应用

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
MicroserviceType	是	是	String	应用微服务类型

LaneInfos

泳道分页查询

被如下接口引用：DescribeLanes

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数
Content	是	是	Array of LaneInfo	泳道信息列表

DescribeTransactionResp

查询主事务详情

被如下接口引用：

名称	必选	允许NULL	类型	描述
Transaction	是	是	TxMainTransaction	主事务详情
Error	是	是	TxError	查询主事务异常信息

TsfRegionResult

地域信息

被如下接口引用：DescribeRegions

名称	必选	允许NULL	类型	描述
TRegionId	否	是	String	tRegionId
TRegionName	否	是	String	tRegionName
TRemark	否	是	String	tRemark

FindContainerGroupResultV2

1.16 容器部署组详情(FindContainerGroup) Result

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	groupId
GroupName	是	是	String	分组名称
InstanceCount	是	是	Int64	实例总数
RunInstanceCount	是	是	Int64	已启动实例总数
CreateTime	是	是	String	创建时间
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
LbIp	是	是	String	负载均衡ip
ClusterIp	是	是	String	Service ip
NodePort	是	是	String	NodePort端口，只有公网和NodePort访问方式才有值
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
Envs	是	是	Array of Env	环境变量数组

名称	必选	允许NULL	类型	描述
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id

CloudMonitorMicroserviceResult

CloudMonitorMicroserviceResult

被如下接口引用：DescribeMicroServiceList、ListCloudMicroServiceFindPagedList

名称	必选	允许NULL	类型	描述
TotalCount	否	否	Int64	TotalCount
Content	否	否	Array of CloudMonitorMicroservice	Content

BusinessLogPatternAnalysis

日志Pattern合法性校验/模拟解析出参

被如下接口引用：AnalyzeLogSchema、ValidateLogSchema

名称	必选	允许NULL	类型	描述
Success	是	否	Bool	校验/解析是否成功
Hint	是	是	String	校验/解析用户提示，包含校验不通过的原因，是否成功联动调用链等
AnalyzeResult	是	是	Array of EnvsV2	解析结果，复用Envs对象，本质是一个Map<String, String>

ConfigV2

配置项

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
ConfigVersionDesc	是	是	String	配置项版本描述

名称	必选	允许NULL	类型	描述
ConfigValue	是	是	String	配置项值
ConfigType	是	是	String	配置项类型
CreationTime	是	是	String	创建时间
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
DeleteFlag	是	是	Bool	删除标识，true：可以删除；false：不可删除
LastUpdateTime	是	是	String	最后更新时间
ConfigVersionCount	是	是	Int64	配置项版本数量

ThreadDetailInfos

线程详情信息数组

被如下接口引用：DescribeThreadDetailList

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ThreadDetailInfo	线程详情数组
TotalCount	是	是	Int64	接口执行状态success/error

SchedulingStrategy

tsf 容器集群节点调度策略

被如下接口引用：DeployContainerGroup

名称	必选	允许NULL	类型	描述
Type	是	是	String	NONE：不使用调度策略；CROSS_AZ：跨可用区部署

VolumeItem

容器卷挂载项信息

被如下接口引用：DescribeContainerVolumeOptions

名称	必选	允许NULL	类型	描述
Name	否	否	String	数据卷名称

VmInstanceResourceConfig

虚拟机实例相关的参数配置

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
ImportMode	是	是	Array of String	实例导入方式，可多个，公有云为 ["R", "M"]，独立版的取值仅有 "M" 脚本模式

DescribeSingleContainerGroupsResult

DescribeSingleContainerGroupsResult

被如下接口引用：DescribeSingleContainerGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	列表数量
Content	是	是	Array of DescribeSingleContainerGroups	列表内容

GroupsByScalableRuleIdList

GroupsByScalableRuleIdList

被如下接口引用：ListGroupsByScalableRuleId

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of GroupsByScalableRuleId	Content

JavaFlameGraph

DescribeFlameGraph接口返回火焰图数据封装

被如下接口引用：DescribeFlameGraph

名称	必选	允许NULL	类型	描述
FlameGraphData	是	是	String	火焰图Json,历史无采集成功的火焰图返回""
CreationTime	是	是	String	火焰图创建时间,格式为(yyyy-MM-dd HH:mm:ss),历史无采集成功的火焰图返回""
Duration	是	是	Int64	火焰图采集时长,历史无采集成功的火焰图返回0

名称	必选	允许NULL	类型	描述
Status	是	是	String	调用成功success/调用失败error
StatusInfo	是	是	String	调用成功为""/调用失败为对应失败信息
StatusCode	是	是	Int64	调用成功为0/调用失败为对应的错误码
Event	是	是	String	采集类型

InstanceEnrichedInfo

包含虚拟机所在TSF中的位置信息

被如下接口引用：DescribeInstances

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器ID
InstanceName	是	是	String	机器名称
LanIp	是	是	String	机器内网IP
WanIp	是	是	String	机器外网IP
VpcId	是	是	String	机器所在VPC
InstanceStatus	是	是	String	机器运行状态 Pending Running Stopped Rebooting Starting Stopping Abnormal Unknown
InstanceAvailableStatus	是	是	String	机器可用状态（表示机器上的Agent在线）
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
GroupId	是	是	String	机器所在部署组ID
GroupName	是	是	String	部署组名称

HealthCheckSettings

健康检查参数

被如下接口引用：DeployContainerGroup、DeployGroup、DescribeContainerGroupDeployInfo、DescribeContainerGroupDetail、DescribeGroup、UpdateHealthCheckSettings

名称	必选	允许NULL	类型	描述
LivenessProbe	否	是	HealthCheckSetting	存活健康检查
ReadinessProbe	否	是	HealthCheckSetting	就绪健康检查

ScalableTaskResultV2

ScalableTaskResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ScalableTask	Content

TsfPageVmSubTask

虚拟机子任务分页信息

被如下接口引用：DescribeSubTasks

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	子任务总数目
SuccessCount	是	是	Int64	成功子任务数目
RunCount	是	是	Int64	运行中子任务数目
FailCount	是	是	Int64	失败子任务数目
Content	是	是	Array of VmSubTask	子任务列表信息

ContainerGroupV2

部署组

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	否	否	String	groupId

名称	必选	允许NULL	类型	描述
GroupName	否	否	String	分组名称
CreateTime	否	否	String	创建时间
Server	是	否	String	镜像server
Reponame	是	否	String	镜像名，如/tsf/nginx
TagName	是	否	String	镜像版本名称
ClusterId	是	否	String	集群id
ClusterName	是	否	String	集群名称
NamespaceId	是	否	String	命名空间id
NamespaceName	是	否	String	命名空间名称
ApplicationId	是	否	String	应用名称
ApplicationName	是	否	String	应用类型
ApplicationType	是	否	String	分组创建时间
UpdateTime	是	否	String	分组更新时间
MicroserviceType	是	是	String	微服务类型

SidecarStatus

sidecar 状态

被如下接口引用：DescribeMeshSidecarStatus

名称	必选	允许NULL	类型	描述
Envoy	是	是	String	envoy状态
MeshDns	是	否	String	mesh-dns 状态
PilotAgent	是	否	String	pilot-agent状态

TsfPageZipkinSpanInfo

调用链Span列表

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条目数

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ZipkinSpanInfo	调用链Span列表

Permission

权限点

被如下接口引用：

名称	必选	允许NULL	类型	描述
PermId	否	是	String	权限点ID
PermCode	否	是	String	权限点编码
PermDesc	否	是	String	权限点描述
ServiceCode	否	是	String	产品编码
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间
DeleteFlag	否	是	Bool	删除标识，true：可以删除；false：不可删除

UploadInfo

上传包ID

被如下接口引用：UploadPkg

名称	必选	允许NULL	类型	描述
UploadId	是	是	String	上传包ID

TemplateProjectV2

ProjectList

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProjectId	是	是	String	工程id
ProjectName	是	是	String	工程名
BasePackage	是	是	String	包路径
LastTime	是	是	Int64	修改时间

名称	必选	允许NULL	类型	描述
Data	否	是	String	Data
AppId	否	是	String	AppId
Uin	否	是	String	Uin
SubAccountUin	否	是	String	SubAccountUin

LicenseGrantee

LicenseGrantee

被如下接口引用：DescribeLicenseApplications、IssueLicense、UploadLicenseApplication

名称	必选	允许NULL	类型	描述
Name	否	是	String	被授予者名称

TaskRecordSimpleInfoPage

翻页查询基本任务信息列表

被如下接口引用：DescribeFlowAvailableTasks

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Uint64	总数量
Content	是	否	Array of TaskRecordSimpleInfo	基本任务信息数组

TsfPageBusinessLogConfig

业务日志配置项列表

被如下接口引用：DescribeBusinessLogConfigs、DescribeGroupBusinessLogConfigs

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of BusinessLogConfig	业务日志配置项列表

ApplicationServerLogContent

日志内容

被如下接口引用：DescribeApplicatoinServerLog

名称	必选	允许NULL	类型	描述
LogContent	是	否	String	日志内容
Timestamp	是	否	String	时间戳
LogType	是	否	String	日志类型

InstancesResult

InstancesResult

被如下接口引用：ListTsfModulesDetail

名称	必选	允许NULL	类型	描述
Ip	是	是	String	Ip值
Status	是	是	String	Status值
Ports	是	是	Array of PortsResult	Ports值
InstanceId	是	是	String	InstanceId值
ModuleId	是	是	String	ModuleId值
ModuleName	是	是	String	ModuleName值
ModuleType	是	是	String	ModuleType值
UserName	是	是	String	UserName值
Password	是	是	String	Password值
ModuleCommon	是	是	Int64	ModuleCommon值
UpdateTime	是	是	String	UpdateTime值
DeployTime	是	是	String	DeployTime值
MachineId	是	是	String	MachineId值
ZoneId	是	是	String	ZoneId值
ZoneName	是	是	String	ZoneName值
ModuleRole	是	是	String	ModuleRole值
SshPort	是	是	Int64	SshPort值

InstanceV2

机器实例

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器ID
InstanceName	是	是	String	机器名称
LanIp	是	是	String	机器内网地址ip
WanIp	是	是	String	机器外网地址ip
InstanceDesc	是	是	String	机器描述信息
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
InstanceStatus	是	是	String	VM的状态 虚机：虚机的状态 容器：Pod所在虚机的状态
InstanceAvailableStatus	是	是	String	VM的可使用状态 虚机：虚机是否能够作为资源使用 容器：虚机是否能够作为资源部署POD
ServiceInstanceStatus	是	是	String	服务下的服务实例的状态 虚机：应用是否可用 + Agent状态 容器：Pod状态
CountInTsf	是	是	Int64	标识此instance是否已添加在tsf中
GroupId	是	是	String	机器所属分组Id
ApplicationId	是	是	String	机器所属应用Id
ApplicationName	是	是	String	机器所属应用名称
InstanceCreatedTime	是	是	String	机器实例在cvm的创建时间
InstanceExpiredTime	是	是	String	机器实例在CVM的过期时间
InstanceChargeType	是	是	String	机器实例在CVM的计费模式
InstanceTotalCpu	是	是	Float	机器实例总CPU信息
InstanceTotalMem	是	是	Float	机器实例总内存信息
InstanceUsedCpu	是	是	Float	机器实例使用的CPU信息
InstanceUsedMem	是	是	Float	机器实例使用的内存信息
InstanceLimitCpu	是	是	Float	机器实例limitCPU信息
InstanceLimitMem	是	是	Float	机器实例limit内存信息
InstancePkgVersion	是	是	String	包版本
ClusterType	是	是	String	集群类型
RestrictState	是	是	String	机器实例业务状态
UpdateTime	是	是	String	最后更新时间

名称	必选	允许NULL	类型	描述
OperationState	是	是	Int64	实例执行状态

ConfigRelease

配置项发布信息

被如下接口引用：DescribeConfigReleases、DescribePublicConfigReleases

名称	必选	允许NULL	类型	描述
ConfigReleaseId	是	是	String	配置项发布ID
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
ReleaseTime	是	是	String	发布时间
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ReleaseDesc	是	是	String	发布描述
ApplicationId	是	是	String	应用ID

ResultStatus

APM模块任务下发成功/失败 状态描述, 由于直接返回true或者false不足以前端做逻辑处理顾封装成Status对象,前端以状态码可做逻辑处理

被如下接口引用：CreateFlameGraph

名称	必选	允许NULL	类型	描述
Status	是	否	String	success任务下发成功/error任务下发失败
StatusInfo	是	否	String	success时为"",失败时为对应的错误信息
StatusCode	是	否	Int64	success时为0,失败时为对应错误码

ServerlessGroup

Serverless部署组信息

被如下接口引用：DescribeServerlessGroup、DescribeServerlessGroups

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	分组名称
CreateTime	是	是	String	创建时间
Status	是	是	String	服务状态
PkgId	是	是	String	程序包ID
PkgName	是	是	String	程序包名
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
VpcId	是	是	String	vpc ID
SubnetId	是	是	String	vpc 子网ID
PkgVersion	是	是	String	程序包版本
Memory	是	是	String	所需实例内存大小
InstanceRequest	是	是	Uint64	要求最小实例数
StartupParameters	是	是	String	部署组启动参数
ApplicationId	是	是	String	应用ID
InstanceCount	是	是	Uint64	部署组实例数
ApplicationName	是	是	Array of String	应用名称

CommonGroupAttribute

通用部署组额外属性

被如下接口引用：DescribeCommonGroupAttribute

名称	必选	允许NULL	类型	描述
Status	是	是	String	部署组状态

名称	必选	允许NULL	类型	描述
InstanceCount	是	是	Int64	部署组实例数

ContainGroup

部署组列表（应用下钻界面的）

被如下接口引用：DescribeContainerGroups

名称	必选	允许NULL	类型	描述
GroupId	否	是	String	部署组ID
GroupName	否	是	String	分组名称
CreateTime	否	是	String	创建时间
Server	是	是	String	镜像server
RepoName	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit

ListTsfRegionResult

ListTsfRegionResult

被如下接口引用：DescribeRegions

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of TsfRegionResult	Content

PkgListV2

包列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	包总量
Content	是	是	Array of PkgInfo	报信息列表

TsfPageApiDetailInfo

ApiDetailInfo 翻页对象

被如下接口引用：DescribeApisWithPlugin、DescribeGatewayApis、DescribeUsableApis

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of ApiDetailInfo	API 信息列表

TaskFlowPage

工作流分页对象

被如下接口引用：DescribeTaskFlows

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	工作流总数
Content	是	否	Array of TaskFlow	工作流分页对象

KafkaDeliveryConfig

投递kafka配置项

被如下接口引用：DescribeDeliveryConfig

名称	必选	允许NULL	类型	描述
ConfigId	是	是	String	配置项id
ConfigName	是	是	String	配置名称
CollectPath	是	是	Array of String	采集路径

名称	必选	允许NULL	类型	描述
KafkaVIp	是	是	String	kafka vip
KafkaVPort	是	是	String	kafka vport
Topic	是	是	String	kafka topic
LineRule	是	是	String	换行规则

ListMonitorStatisticsPolicyResult

ListMonitorStatisticsPolicy结果返回类型

被如下接口引用：

名称	必选	允许NULL	类型	描述
PolicyId	是	否	String	监控统计策略id
KeyWords	是	否	String	关键词
GroupList	是	否	GroupList	部署组列表信息
NamespaceId	是	否	String	命名空间id
CreateTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间

GetContainGroupOtherResult

部署组其他字段获取

被如下接口引用：GetContainGroupOther

名称	必选	允许NULL	类型	描述
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动实例总数
LbIp	是	否	String	负载均衡ip
ClusterIp	是	否	String	Service ip
Status	是	否	String	服务状态，请参考后面的的状态定义
Message	是	否	String	异常信息,服务状态为Abnormal时才有

StatisticsEntry

概览页统计列表条目

被如下接口引用：GetTopAvgTimeCostInterfaces、GetTopAvgTimeCostServices、GetTopFailureRateInterfaces、GetTopFailureRateServices、GetTopReqAmountInterfaces、GetTopReqAmountServices

名称	必选	允许NULL	类型	描述
PrimeKey	是	是	String	主字段
SubKey	是	是	String	附字段
Value	是	是	String	值

ManagerHttp

运营端透传http请求的入参类型

被如下接口引用：

名称	必选	允许NULL	类型	描述
Url	否	否	String	url
Body	否	否	String	body
Method	否	否	String	method

StatisticsCoord

指标坐标

被如下接口引用：

名称	必选	允许NULL	类型	描述
CoordX	是	是	String	指标横坐标值
CoordY	是	是	String	指标纵坐标值
CoordTag	是	是	String	指标标签，用于标识附加信息

GroupsByScalableRuleIdListV2

GroupsByScalableRuleIdList

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount

名称	必选	允许NULL	类型	描述
Content	是	是	Array of GroupsByScalableRuleId	Content

TemplateResultV2

TemplateResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TotalCount
Content	是	是	Array of TemplateProject	Content

AlarmReceiverResult

AlarmReceiverResult

被如下接口引用：DescribeAlarmPolicy、ListAlarmPolicies、ListAlarmReceivers

名称	必选	允许NULL	类型	描述
PolicyId	是	是	String	PolicyId
ReceiverId	是	是	String	ReceiverId
Name	是	是	String	Name
CellPhoneNumber	是	是	String	CellPhoneNumber
Email	是	是	String	Email

DeployGroupV2

部署组字段信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组Id
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群Id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间Id
NamespaceName	是	是	String	命名空间名称
GroupDesc	是	是	String	部署组描述
CreateTime	是	是	String	部署组创建时间
UpdateTime	是	是	String	部署组更新时间
InstanceCount	是	是	Int64	部署组实例数目
RunInstanceCount	是	是	Int64	部署组运行实例数目
OffInstanceCount	是	是	Int64	部署组停止实例数目
GroupStatus	是	是	String	部署组状态
StartupParameters	是	是	String	部署组启动参数
PackageId	是	是	String	部署组程序包Id
PackageName	是	是	String	部署组程序包名称
PackageVersion	是	是	String	部署组程序包版本
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
CpuLimit	是	是	String	最大分配cpu 核数，如0.6
MemLimit	是	是	String	最大分配内存M数
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数
ClusterIp	是	是	String	Service ip
LbIp	是	是	String	负载均衡ip
Envs	是	是	Array of Env	环境变量数组对象

名称	必选	允许NULL	类型	描述
Message	是	是	String	pod错误信息描述
NodePort	是	是	Int64	NodePort端口，只有公网和NodePort访问方式才有值
Status	是	是	String	部署组状态
MicroserviceType	是	是	String	应用微服务类型

GraphNode

依赖拓扑图节点

被如下接口引用：

名称	必选	允许NULL	类型	描述
Name	是	是	String	节点服务名
Type	是	是	String	节点服务类型
ReqTotalQty	是	是	Uint64	节点服务总请求量
ReqSuccessQty	是	是	Uint64	节点服务成功请求量
ReqFailedQty	是	是	Uint64	节点服务失败请求量
ReqPerMin	是	是	Float	节点服务每分钟平均请求量
AvgDurationMs	是	是	Float	节点服务平均耗时（毫秒）
EdgeList	是	是	Array of GraphEdge	节点服务为目的的边列表（入度）

OverviewBasicResourceUsage

TSF基本资源信息概览

被如下接口引用：DescribeBasicResourceUsage

名称	必选	允许NULL	类型	描述
ApplicationCount	是	是	Int64	应用总数
NamespaceCount	是	是	Int64	命名空间总数
GroupCount	是	是	Int64	部署组个数
PackageSpaceUsed	是	是	Int64	程序包存储空间用量，单位字节
ConsulInstanceCount	是	是	Int64	已注册实例数

PkgList

包列表

被如下接口引用：DescribePkgs、ListPkg

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	程序包总量
Content	是	是	Array of PkgInfo	程序包信息列表
RepositoryId	是	是	String	程序包仓库id
RepositoryType	是	是	String	程序包仓库类型
RepositoryName	是	是	String	程序包仓库名称

SidecarFilters

分页展示过滤器

被如下接口引用：DescribeSidecarFilters

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数
Content	是	否	Array of SidecarFilter	过滤器列表

PagedProductNews

产品动态分组分页对象

被如下接口引用：DescribeProductNews

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of ProductNews	产品动态列表

BusinessLogConfig

业务日志配置

被如下接口引用：DescribeBusinessLogConfig、DescribeBusinessLogConfigs、DescribeGroupBusinessLogConfigs

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ConfigId	否	否	String	配置项ID
ConfigName	否	否	String	配置项名称
ConfigPath	否	是	String	配置项日志路径
ConfigDesc	否	是	String	配置项描述
ConfigTags	否	是	String	配置项标签
ConfigPipeline	否	是	String	配置项对应的ES管道
ConfigCreateTime	否	是	String	配置项创建时间
ConfigUpdateTime	否	是	String	配置项更新时间
ConfigSchema	否	是	BusinessLogConfigSchema	配置项解析规则
ConfigAssociatedGroups	否	是	Array of BusinesLogConfigAssociatedGroup	配置项关联部署组

GroupInfo

日志投递kafka用，描述部署组信息

被如下接口引用：AssociateConfigWithGroup、DescribeAssociateRelation、DescribeDeliveryConfigs

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	部署组id
GroupName	是	否	String	部署组名称
ClusterId	否	是	String	集群id
ClusterName	否	是	String	集群名称
ClusterType	是	否	String	集群类型
NamespaceName	否	是	String	命名空间名称
AssociateTime	否	是	String	绑定时间

TaskFlowBatchEdge

工作流执行批次的图信息

被如下接口引用：DescribeFlowBatchGraph

名称	必选	允许NULL	类型	描述
FlowId	是	否	String	工作流 ID

名称	必选	允许NULL	类型	描述
FlowBatchId	是	否	String	工作流批次 ID
FlowBatchLogId	是	否	String	工作流批次历史 ID
BatchId	是	是	String	批次 ID
BatchLogId	是	是	String	批次历史 ID
GraphId	是	否	String	图 ID
NodeId	是	否	String	节点 ID
ChildNodeId	是	是	String	子节点 ID
CoreNode	是	是	String	是否核心节点
NodeName	是	是	String	节点名称
EdgeType	是	是	String	边节点类型
NodeType	是	是	String	节点类型
SuccessRatio	是	是	Int64	节点执行成功率
PositionX	是	是	String	X轴坐标
PositionY	是	是	String	Y轴坐标
State	是	是	String	节点状态
TaskId	是	是	String	任务节点ID
TaskLogId	是	是	String	任务节点历史ID

ContainerGroupResult

镜像版本列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ContainerGroup	镜像版本列表
TotalCount	是	是	Int64	总记录数

GroupPod

部署组实例列表

被如下接口引用：DescribePodInstances

名称	必选	允许NULL	类型	描述
PodName	是	是	String	实例名称(对应到kubernetes的pod名称)
PodId	是	是	String	实例ID(对应到kubernetes的pod id)
Status	是	是	String	实例状态，请参考后面的实例以及容器的状态定义
Reason	是	是	String	实例处于当前状态的原因，例如容器下载镜像失败
NodeIp	是	是	String	主机IP
Ip	是	是	String	实例IP
RestartCount	是	是	Int64	实例中容器的重启次数
ReadyCount	是	是	Int64	实例中已就绪容器的个数
Runtime	是	是	String	运行时长
CreatedAt	是	是	String	实例启动时间
ServiceInstanceStatus	是	是	String	服务实例状态
InstanceAvailableStatus	是	是	String	机器实例可使用状态
InstanceStatus	是	是	String	机器实例状态
NodeInstanceId	是	是	String	节点实例id

ThreadDetails

DescribeThreadDetail接口返回的数据列表

被如下接口引用：DescribeThreadDetail

名称	必选	允许NULL	类型	描述
Data	是	是	Array of ThreadDetail	线程详情列表
Status	是	是	String	接口执行状态success/error
StatusInfo	是	是	String	success时为"", error时为错误信息
StatusCode	是	是	Int64	success时为, error时为错误码

TsfPageSimpleClusterV2

TSF分页简单集群

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleCluster	简单集群列表

TsfPageTraceInterface

调用链接口列表

被如下接口引用：GetOssTraceInterfaces、GetTraceInterfaces

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	符合条件的总调用链接口数量
Content	是	是	Array of String	调用链接口列表

Tag

标签

被如下接口引用：DescribeLicenses

名称	必选	允许NULL	类型	描述
TagKey	否	是	String	标签键
TagValue	否	是	String	标签值

PolicyItem

tsf-privilege模块，策略项

被如下接口引用：CreatePolicyDocument

名称	必选	允许NULL	类型	描述
RoleId	是	是	String	角色ID
ProgramId	是	是	String	数据集ID

ProgramItem

tsf-privilege模块，数据项

被如下接口引用：CreateProgram、DescribeProgram、DescribePrograms、ModifyProgram

名称	必选	允许NULL	类型	描述
ProgramItemId	否	是	String	数据项ID
Resource	否	是	Resource	资源
ValueList	否	是	Array of String	数据值列表
IsAll	否	是	Bool	全选标识，true: 全选；false: 非全选
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间
DeleteFlag	否	是	Bool	删除标识，true: 可删除；false: 不可删除
ProgramId	否	是	String	数据集ID

DescribeResourceConfigCluster

返回给前端的控制信息

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Container	是	是	DescribeResourceConfigClusterContainer	返回给前端的控制信息

ApiUseStatisticsEntity

API 日统计数据点

被如下接口引用：DescribeApiUseDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	名称
Count	是	否	String	次数
Ratio	是	否	String	比率

ModuleParameterResult

ModuleParameterResult

被如下接口引用：SaveDeployModuleParams

名称	必选	允许NULL	类型	描述
ParameterConfigType	否	否	String	ParameterConfigType

名称	必选	允许NULL	类型	描述
ParameterSubType	否	否	String	ParameterSubType
ParameterName	否	否	String	ParameterName
ParameterValue	否	否	String	ParameterValue

Metric

指标

被如下接口引用：DescribeInvocationMetricDataCurve

名称	必选	允许NULL	类型	描述
Name	否	是	String	指标名称
Function	否	是	String	指标计算方式

UnitNamespace

微服务网关单元化命名空间

被如下接口引用：CreateUnitNamespaces、DescribeUnitNamespaces、DescribeUsableUnitNamespaces

名称	必选	允许NULL	类型	描述
Id	否	是	String	单元化命名空间ID
NamespaceId	是	否	String	命名空间ID
NamespaceName	是	否	String	命名空间Name

TsfPageUnitNamespace

单元化命名空间翻页对象

被如下接口引用：DescribeUnitNamespaces、DescribeUsableUnitNamespaces

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	记录总数
Content	是	否	Array of UnitNamespace	记录实体列表

ResourceBatchOperationDescription

资源批次操作描述

被如下接口引用：DescribeResourceBatchOperation

名称	必选	允许NULL	类型	描述
EnableForcedStop	是	否	Bool	是否强制终止变更
EnableBeta	是	否	Bool	是否存在beta批次
BatchNum	是	否	Uint64	批次个数
BatchExeMode	是	否	String	批次执行方式
TaskType	是	否	Uint64	批次任务类型
GroupId	是	否	String	部署组ID
GroupName	是	否	String	部署组名称
CreateTime	是	否	String	创建时间

ContainerAdditionalResourceRequirement

应用使用容器部署时需要的额外资源

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Cpu	是	是	String	CPU 核数
Mem	是	是	String	内存 MiB 数

InstanceAdvancedSettings

容器导入实例高级设置

被如下接口引用：AddClusterInstances

名称	必选	允许NULL	类型	描述
MountTarget	是	否	String	数据盘挂载点, 默认不挂载数据盘. 已格式化的 ext3，ext4，xfs 文件系统的数据盘将直接挂载，其他文件系统或未格式化的数据盘将自动格式化为ext4并挂载，请注意备份数据! 无数据盘或有多块数据盘的云主机此设置不生效。注意，注意，多盘场景请使用下方的DataDisks数据结构，设置对应的云盘类型、云盘大小、挂载路径、是否格式化等信息。注意：此字段可能返回 null，表示取不到有效值。
DockerGraphPath	是	否	String	dockerd --graph 指定值, 默认为 /var/lib/docker注意：此字段可能返回 null，表示取不到有效值。

InterfaceStatistic

接口统计

被如下接口引用：DescribeInterfaceStatistic

名称	必选	允许NULL	类型	描述
Path	是	否	String	请求路径模版
Method	是	否	String	请求方法
RequestCount	是	是	Int64	接口请求量
ErrorRate	是	是	Float	错误率0-1之间小数
AvgTimeConsuming	是	是	Int64	平均耗时,单位ms
P50	是	是	Int64	p50,单位ms
P99	是	是	Int64	p99,单位ms

DescribeResourceConfigLicenseFunction

DescribeResourceConfig

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Name	是	是	String	name
Enable	是	是	Bool	enable

DescribeSubTransactionResp

查询子事务列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
Content	是	是	Array of SubTransaction	子事务列表
TotalCount	是	是	Int64	子事务个数
Error	是	是	TxError	错误信息

TsfPageCommonPkg

公共包列表

被如下接口引用：DescribeCommonPkg

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数
Content	是	否	Array of CommonPkg	公共包信息

TsfPageSimpleGroup

TSF简单部署组分页列表

被如下接口引用：DescribeSimpleGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleGroup	简单部署组列表

GatewayNamespace

微服务网关可用的命名空间信息

被如下接口引用：DescribeAuthNamespaces

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	命名空间主键
NamespaceCode	是	是	String	命名空间编码
NamespaceName	是	是	String	命名空间名称
NamespaceDesc	是	是	String	命名空间备注
ClusterId	是	是	String	集群ID
NamespaceResourceType	是	是	String	命名空间资源类型
NamespaceType	是	是	String	命名空间类型, DEF GLOBAL

MsApplicationV2

应用

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	否	是	String	应用 ID，如 application-qv3dkda7
ApplicationName	否	是	String	应用名

名称	必选	允许NULL	类型	描述
ApplicationDesc	否	是	String	应用描述
ApplicationType	否	是	String	应用 *部署* 类型，V 表示 CVM 应用，C 表示容器应用
ApplicationJvmArg	否	是	String	应用启动时的 JVM 参数
MicroserviceType	否	是	String	应用类型，M 表示 Mesh 应用，N 表示 Spring Cloud 应用
ProgLang	否	是	String	应用编程语言，J 表示 Java，P 表示 Python
NotExist	否	是	Bool	应用是否存在

TsfPageRegion

TSF地域列表对象

被如下接口引用：DescribeTsfRegions

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TSF地域总数目
Content	是	是	Array of TsfRegion	TSF地域列表

RouteTagV2

服务路由规则项匹配条件

被如下接口引用：CreateRoute、DescribeRoute、DescribeRoutes、ModifyRoute

名称	必选	允许NULL	类型	描述
TagId	否	是	String	路由规则项TAG匹配项ID
TagType	是	否	String	标签类型，S：系统标签， U：自定义标签
TagField	是	否	String	标签字段名称
TagOperator	是	否	String	标签匹配规则.EQUAL：等于、NOT_EQUAL：不等于、IN：包含、NOT_IN：不包含、REGEX：正则
TagValue	是	否	String	标签取值
RouteRuleId	否	是	String	TAG匹配项所述路由规则项ID

TaskFlowEdge

工作流图中的边

被如下接口引用：CheckFlowGraphValidity、CreateTaskFlow、DescribeTaskFlowGraph、ModifyTaskFlow

名称	必选	允许NULL	类型	描述
NodeId	是	否	String	节点 ID
ChildNodeId	是	是	String	子节点 ID
CoreNode	是	是	String	是否核心任务,Y/N
EdgeType	是	是	String	边类型
NodeType	是	否	String	任务节点类型
PositionX	是	是	String	X轴坐标位置
PositionY	是	是	String	Y轴坐标位置
GraphId	是	是	String	图 ID
FlowId	是	是	String	工作流 ID
NodeName	是	是	String	节点名称
TaskId	是	是	String	任务ID
TaskLogId	是	是	String	任务历史ID

ContainerGroupOtherV2

部署组列表-其它字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动实例总数
LbIp	是	否	String	负载均衡ip
ClusterIp	是	否	String	Service ip
Status	是	否	String	服务状态，请参考后面的的状态定义
Message	是	否	String	服务状态，请参考后面的的状态定义
Envs	是	否	Array of Env	环境变量

ZipkinAnnotation

调用链Span注解

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
Value	是	是	String	注解值
Timestamp	是	是	Uint64	注解时间戳
Endpoint	是	是	ZipkinEndPoint	注解端点信息

TsfPageZone

TSF可用区列表对象

被如下接口引用：DescribeTsfZones

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TSF可用区总数目
Content	是	是	Array of TsfZone	TSF可用区列表

ContainerGroupResourceConfig

容器部署组相关的参数配置

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
AdditionalResourceRequirement	是	是	ContainerAdditionalResourceRequirementMap	不同类型的应用的容器部署组，部署时的额外资源要求

Person

tsf告警策略的接收者

被如下接口引用：CreateAlarmPolicy

名称	必选	允许NULL	类型	描述
CellPhoneNumber	是	否	String	接收成员电话
Email	是	否	String	接收成员email
Name	是	否	String	接收成员的名字

TsfPageApplication

应用分页信息

被如下接口引用：DescribeApplications

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	应用总数目
Content	是	是	Array of ApplicationForPage	应用信息列表

TaskId

任务id

被如下接口引用：DeployGroup、DeployInstance、ExpandGroup、ShirkGroup、ShirkInstance、ShrinkGroup、ShrinkInstance、ShrinkInstances、StartGroup、StartInstance、StopGroup、StopInstance

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	任务ID

ServiceStatisticsV2

服务运行统计指标

被如下接口引用：DescribeServiceStatistics

名称	必选	允许NULL	类型	描述
ServiceId	是	是	String	服务ID
ReqTotalQty	是	是	Uint64	总请求量
ReqSuccessRate	是	是	Float	请求成功率
ReqAvgDuration	是	是	Float	请求平均耗时

AlarmOverviewContent

告警记录概览页信息

被如下接口引用：DescribeAlarmOverviewList

名称	必选	允许NULL	类型	描述
MetricName	是	否	String	展示给前端的告警记录的指标名
RestoredNums	是	否	Int64	该告警指标24小时内，已恢复的告警条数

名称	必选	允许NULL	类型	描述
UnRestoredNums	是	否	Int64	该告警指标24小时内，未恢复的告警条数

ContainGroupResultV2

部署组列表（应用下钻）

被如下接口引用：

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ContainGroup	部署组列表
TotalCount	是	否	Int64	总记录数

OverviewMsResultV2

概览页微服务信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
RunMicroserviceCount	是	是	Int64	概览页微服务数目

RecordV2

操作记录对象

被如下接口引用：DescribeRecords

名称	必选	允许NULL	类型	描述
AppId	是	是	String	账号AppId
Uin	是	是	String	账号Uin
SubAccountUin	是	是	String	账号SubAccountUin
RecordId	是	是	String	操作记录Id
ModuleType	是	是	String	模块类型
OperationType	是	是	String	操作类型
OperationResource	是	是	String	操作资源信息
OperationMsg	是	是	String	操作记录描述
OperationTime	是	是	String	操作记录时间

名称	必选	允许NULL	类型	描述
OperationStatus	是	是	String	操作状态
Operator	是	是	String	操作人

ContainerAdditionalResourceRequirementMap

不同类型的应用的容器部署组，部署时的额外资源要求

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
M	是	是	ContainerAdditionalResourceRequirement	Mesh 应用部署时需要的额外资源
N	是	是	ContainerAdditionalResourceRequirement	普通应用部署时需要的额外资源

TsfPageApplicationV2

应用分页信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	应用总数目
Content	是	是	Array of ApplicationForPage	应用信息列表

InstanceEnrichedInfoPage

InstanceEnrichedInfo列表结构

被如下接口引用：DescribeInstances

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	总数量
Content	是	是	Array of InstanceEnrichedInfo	列表

TsfPageZipkinTraceInfo

调用链列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条目数
Content	是	是	Array of ZipkinTraceInfo	调用链列表

PagedService

tsf-privilege模块，分页产品列表

被如下接口引用：DescribeServices

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of Service	产品列表

TsfPageRouteAffinity

TSF就近访问列表对象

被如下接口引用：DescribeNamespaceAffinities

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TSF就近策略总数目
Content	是	是	Array of RouteAffinity	TSF就近策略列表

MetricDataCurve

指标监控数据曲线

被如下接口引用：DescribeInvocationMetricDataCurve、DescribeStatistics

名称	必选	允许NULL	类型	描述
MetricName	是	是	String	指标名称
MetricFunction	是	是	String	指标计算方式
MetricDataPoints	是	是	Array of MetricDataPoint	指标数据点集合

LaneInfo

泳道

被如下接口引用：DescribeGroupLane、DescribeLane、DescribeLanes、DisableLaneGroupEntrance

名称	必选	允许NULL	类型	描述
LaneId	是	是	String	泳道ID
LaneName	是	是	String	泳道名称
Remark	是	是	String	泳道备注
CreateTime	是	是	Int64	创建时间
UpdateTime	是	是	Int64	更新时间
LaneGroupList	是	是	Array of LaneGroup	泳道部署组
Entrance	是	是	Bool	是否入口应用
NamespaceIdList	是	是	Array of String	泳道已经关联部署组的命名空间列表

ContainerEvent

返回容器的事件，比如 k8s deployment 或者 pod 的 events

被如下接口引用：DescribeContainerEvents

名称	必选	允许NULL	类型	描述
FirstTimestamp	否	是	Int64	第一次出现的时间，以 ms 为单位的时间戳
LastTimestamp	否	是	Int64	最后一次出现的时间，以 ms 为单位的时间戳
Type	否	是	String	级别
Kind	否	是	String	资源类型
Name	否	是	String	资源名称
Reason	否	是	String	内容
Message	否	是	String	详细描述
Count	否	是	Int64	出现次数

CloudMonitorPolicyResultV2

云监控对象策略对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of CloudMonitorPolicies	Content

MsRunningApplicationV2

微服务应用信息

被如下接口引用：DescribeMsRunningApplications

名称	必选	允许NULL	类型	描述
ApplicationId	是	否	String	应用ID
ApplicationName	是	是	String	应用名称

TxListV2

查询事务列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	返回的事务数量
Content	是	是	TxMainTransaction	主事务信息
Error	是	是	TxError	事务异常信息

GatewayTagPlugin

微服务网关TAG插件明细

被如下接口引用：DescribeGatewayTagPlugin

名称	必选	允许NULL	类型	描述
Id	是	否	String	网关插件id
Name	是	是	String	插件名称
Description	是	是	String	插件描述
Type	是	是	String	插件类型
CreatedTime	是	是	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	是	String	插件更新时间 如:2019-06-20 15:51:28
Status	是	是	String	发布状态: drafted/released
TagPluginInfoList	是	是	String	参数配置json串

ResourceNodeStatus

模块节点的资源使用情况

被如下接口引用：DescribeResourceUsageRate

名称	必选	允许NULL	类型	描述
Used	是	否	Uint64	已使用的资源量
Total	是	否	Uint64	总的资源量

TsfPageMsInstanceWrong

微服务实例的分页内容

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of MsInstanceWrong	微服务实例列表内容
MicroserviceType	是	是	String	微服务类型

ConfigReleaseLog

配置项发布日志

被如下接口引用：DescribeConfigReleaseLogs、DescribePublicConfigReleaseLogs

名称	必选	允许NULL	类型	描述
ConfigReleaseLogId	是	是	String	配置项发布日志ID
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ReleaseTime	是	是	String	发布时间

名称	必选	允许NULL	类型	描述
ReleaseDesc	是	是	String	发布描述
ReleaseStatus	是	是	String	发布状态
LastConfigId	是	是	String	上次发布的配置项ID
LastConfigName	是	是	String	上次发布的配置项名称
LastConfigVersion	是	是	String	上次发布的配置项版本
RollbackFlag	是	是	Bool	回滚标识

ListMonitorStatisticsPolicyResultV2

ListMonitorStatisticsPolicy结果返回类型

被如下接口引用：

名称	必选	允许NULL	类型	描述
PolicyId	是	否	String	监控统计策略id
KeyWords	是	否	String	关键词
GroupList	是	否	GroupList	部署组列表信息
NamespaceId	是	否	String	命名空间id
CreateTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间

TxRetry

事务重试接口出参

被如下接口引用：AutoRetryTransaction、RetryTransaction

名称	必选	允许NULL	类型	描述
Success	是	是	Array of String	重试成功的事务ID列表
Missing	是	是	Array of String	重试失败的事务ID列表
Error	是	是	TxError	重试异常信息

RepositoryInfo

仓库信息

被如下接口引用：DescribeRepositories、DescribeRepository

名称	必选	允许NULL	类型	描述
RepositoryId	是	否	String	仓库ID
RepositoryName	是	是	String	仓库名称
RepositoryType	是	是	String	仓库类型（默认仓库：default，私有仓库：private）
RepositoryDesc	是	是	String	仓库描述
IsUsed	是	是	Bool	仓库是否正在被使用
CreateTime	是	是	String	仓库创建时间
BucketName	是	是	String	仓库桶名称
BucketRegion	是	是	String	仓库桶所在地域
Directory	是	是	String	仓库目录

ClusterV2

集群详情

被如下接口引用：DescribeCluster、DescribeClusters

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterDesc	是	是	String	集群描述
ClusterType	是	是	String	集群类型
VpcId	是	是	String	集群所属私有网络ID
ClusterStatus	是	是	String	集群状态
ClusterCIDR	是	是	String	集群CIDR
ClusterTotalCpu	是	是	Float	集群总CPU，单位: 核
ClusterTotalMem	是	是	Float	集群总内存，单位: G
ClusterUsedCpu	是	是	Float	集群已使用CPU，单位: 核
ClusterUsedMem	是	是	Float	集群已使用内存，单位: G
InstanceCount	是	是	Int64	集群机器实例数量
RunInstanceCount	是	是	Int64	集群运行中的机器实例数量

名称	必选	允许NULL	类型	描述
NormalInstanceCount	是	是	Int64	集群正常状态的机器实例数量
DeleteFlag	是	是	Bool	删除标记：true：可以删除；false：不可删除
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
TsfRegionId	是	是	String	集群所属TSF地域ID
TsfRegionName	是	是	String	集群所属TSF地域名称
TsfZoneId	是	是	String	集群所属TSF可用区ID
TsfZoneName	是	是	String	集群所属TSF可用区名称
DeleteFlagReason	是	是	String	集群不可删除的原因
SubnetId	是	是	String	集群所属私有网络子网ID
ClusterLimitCpu	是	是	String	集群剩余 cpu limit
ClusterLimitMem	是	是	String	集群剩余 memory limit
RunServiceInstanceCount	是	是	Int64	运行服务实例数
OperationInfo	是	是	OperationInfo	给前端的按钮控制信息
ClusterVersion	是	是	String	容器集群版本
GroupCount	是	是	Uint64	部署组总数
RunGroupCount	是	是	Uint64	运行中部署组数
StopGroupCount	是	是	Uint64	停止中部署组数
AbnormalGroupCount	是	是	Uint64	异常部署组数
ClusterRemarkName	是	是	String	集群备注名
ClusterCpuType	否	是	String	集群cpu架构
DisableProgramAuthCheck	否	是	Bool	关闭数据集校验
KuberneteApiServer	否	是	String	kuberneteApiServer
ProgramId	否	是	String	数据集id
Unbind	否	是	Bool	unbind
KuberneteNativeType	否	是	String	类型
MaxClusterServiceNum	否	是	Int64	最大服务数
KuberneteNativeSecret	否	是	String	secret
MaxNodePodNum	否	是	Int64	最大pod数

TsfPageZipkinTraceInfoV2

调用链列表

被如下接口引用：SearchOssTrace、SearchTrace

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条目数
Content	是	是	Array of ZipkinTraceInfoV2	调用链列表

TsfPageVmTaskV2

虚拟机任务分页信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	任务总数目
Content	是	是	Array of VmTask	任务详情列表

MultipartPkg

分片包详情

被如下接口引用：DownloadMultipartPkg

名称	必选	允许NULL	类型	描述
PkgId	否	否	String	程序包ID
FileSeg	否	否	Uint64	当前分片 (从1开始)
FileSegments	否	是	Uint64	总分片数
SegmentSize	否	是	Uint64	每个分片的大小 (字节数)
Md5	否	是	String	文件的MD5
Size	否	是	Uint64	文件大小（总字节数）
File	否	是	String	文件编码数据 (@为首，表示base64编码)

PropertyField

属性字段

被如下接口引用：DescribeApiDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	属性名称
Type	是	否	String	属性类型
Description	是	是	String	属性描述

BusinessLogV2

业务日志

被如下接口引用：SearchBusinessLog、SearchOssBusinessLog、SearchOssRealtimeBusinessLog、SearchOssSpanBusinessLog、SearchOssStaticBusinessLog、SearchOssSurroundBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog、SearchSpanBusinessLog、SearchSurroundBusinessLog

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID
Content	是	是	String	日志内容
Timestamp	是	是	Uint64	日志时间戳
InstanceIp	是	是	String	实例IP
LogId	是	是	String	日志ID
GroupId	是	是	String	部署组ID

MonitorGroup

部署组信息，用于云监控

被如下接口引用：DescribeCloudMonitorGroups

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称

名称	必选	允许NULL	类型	描述
GroupDesc	是	是	String	部署组描述信息

TsfPageSimpleCluster

TSF分页简单集群

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleCluster	简单集群列表

TsfPageImageFeature

镜像特征列表

被如下接口引用：DescribeImageFeatures

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	镜像特征总数目
Content	是	是	Array of ImageFeature	镜像特征列表

TagRouteItemSourceV2

标签路由规则项源标签

被如下接口引用：

名称	必选	允许NULL	类型	描述
TagRouteSourceId	否	否	String	标签路由规则项源标签ID
SourceType	是	否	String	标签路由，标签类型，表示系统标签或自定义标签, 系统标签： S, 自定义标签： U
SourceField	是	否	String	标签路由匹配源字段
SourceMatchRule	是	否	String	标签路由匹配源规则, 等于: EQUAL， 不等于： NOT_EQUAL， 包含： IN， 不包含： NOT_IN， 正则表达式： REGEX
SourceValue	是	否	String	标签路由匹配源取值
TagRouteItemId	否	否	String	标签路由规则项ID

ImageTag

列表信息

被如下接口引用：DescribeImageTags

名称	必选	允许NULL	类型	描述
RepoName	是	否	String	仓库名
TagName	是	否	String	版本名称
TagId	是	否	String	版本ID
ImageId	是	否	String	镜像ID
Size	是	否	String	大小
CreationTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间
Author	是	否	String	镜像制作者
Architecture	是	否	String	CPU架构
DockerVersion	是	否	String	Docker客户端版本
Os	是	是	String	操作系统
PushTime	是	否	String	push时间
SizeByte	是	否	Int64	单位为字节

AdvanceSettings

高级选项设置

被如下接口引用：CreateTask、DescribeTaskDetail、DescribeTaskRecords、ModifyTask

名称	必选	允许NULL	类型	描述
SubTaskConcurrency	否	否	Int64	子任务单机并发数限制，默认值为2

ImageUserIsExistsResultV2

ImageUserIsExistsResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
Data	是	是	ImageUserIsExists	用户是否存在

LicenseApplicationRecord

LicenseApplicationRecord

被如下接口引用：DescribeLicenseApplications

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Uint64	创建时间
Grantee	是	否	LicenseGrantee	授予对象
Product	否	否	Array of LicenseProduct	所申请的产品信息列表
Phone	否	是	Array of String	联系电话
Email	否	是	Array of String	联系邮箱
Description	否	是	String	备注
IssueTime	否	是	Uint64	授予许可时间
Duration	否	是	LicenseDuration	有效期信息

RatelimitRule

限流规则

被如下接口引用：

名称	必选	允许NULL	类型	描述
Id	否	否	String	规则ID
Name	是	否	String	规则名字，在一个微服务下唯一
Status	否	否	Uint64	状态 0表示启用 1表示停用
SourceService	否	是	String	限流规则区分的来源微服务名
DurationSecond	是	否	Uint64	限流周期，单位秒
DurationQuota	是	否	Uint64	每周期内的限流配额
ModifyTime	否	否	Uint64	最近一次修改规则时间，UTC秒数
Description	否	是	String	描述
Dimensions	是	是	Array of RatelimitDimension	限制条件列表

GroupPodV2

部署组实例列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
PodName	是	是	String	实例名称(对应到kubernetes的pod名称)
PodId	是	是	String	实例ID(对应到kubernetes的pod id)
Status	是	是	String	实例状态，请参考后面的实例以及容器的状态定义
Reason	是	是	String	实例处于当前状态的原因，例如容器下载镜像失败
NodeIp	是	是	String	主机IP
Ip	是	是	String	实例IP
RestartCount	是	是	Int64	实例中容器的重启次数
ReadyCount	是	是	Int64	实例中已就绪容器的个数
Runtime	是	是	String	运行时长
CreatedAt	是	是	String	实例启动时间
ServiceInstanceStatus	是	是	String	服务实例状态
InstanceAvailableStatus	是	是	String	机器实例可使用状态
InstanceStatus	是	是	String	机器实例状态

TsfPageRouteV2

路由分页列表

被如下接口引用：DescribeRoutes

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	服务路由总数目
Content	是	是	Array of RouteV2	服务路由详情列表

PagedResource

分页资源列表

被如下接口引用：DescribeResources

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of Resource	资源列表

ProjectListV2

ProjectList

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	工程id
ProjectName	是	否	String	工程名
BasePackage	是	否	String	包路径
LastTime	是	否	Int64	修改时间
AppId	否	否	String	AppId
Uin	否	否	String	Uin
SubAccountUin	否	否	String	SubAccountUin
Data	否	否	String	Data

TsfPageFileConfigRelease

文件配置项发布信息列表

被如下接口引用：DescribeFileConfigReleases

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	数量
Content	是	是	Array of FileConfigRelease	列表

ShardArgument

分片参数

被如下接口引用：CreateTask、DescribeTaskDetail、DescribeTaskExecuteShardArgument、DescribeTaskRecords、ModifyTask

名称	必选	允许NULL	类型	描述
ShardKey	是	否	Uint64	分片参数 KEY，整形, 范围 [1,1000]
ShardValue	是	是	String	分片参数 VALUE

TemplatePageResultV2

TemplatePageResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TotalCount
Content	是	否	Array of ProjectList	Content

BusinessLogConfigSchema

业务日志配置解析规则

被如下接口引用：CreateBusinessLogConfig、DescribeBusinessLogConfig、DescribeBusinessLogConfigs、DescribeGroupBusinessLogConfigs、UpdateBusinessLogConfig

名称	必选	允许NULL	类型	描述
SchemaType	是	否	Int64	解析规则类型
SchemaContent	否	是	String	解析规则内容
SchemaDateFormat	否	是	String	解析规则时间格式
SchemaMultilinePattern	否	是	String	解析规则对应的多行匹配规则
SchemaCreateTime	否	是	String	解析规则创建时间
SchemaPatternLayout	否	是	String	用户填写的解析规则

PagedPermission

分页权限点

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of Permission	权限点列表

Namespace

命名空间

被如下接口引用：DescribeNamespaces、DescribeSimpleNamespaces

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	命名空间ID

名称	必选	允许NULL	类型	描述
NamespaceCode	是	是	String	命名空间编码
NamespaceName	是	是	String	命名空间名称
NamespaceDesc	是	是	String	命名空间描述
IsDefault	是	是	String	默认命名空间
NamespaceStatus	是	是	String	命名空间状态
DeleteFlag	是	是	Bool	删除标识
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
ClusterList	是	是	Array of Cluster	集群数组，仅携带集群ID，集群名称，集群类型等基础信息。
ClusterId	是	是	String	集群ID
NamespaceResourceType	是	是	String	集群资源类型
NamespaceType	是	是	String	命名空间类型
IsHaEnable	是	是	String	是否开启高可用
ProgramId	否	是	String	数据集id

TsfPageRouteRule

Tsf 路由规则分页路由

被如下接口引用：DescribeRouteRules

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of RouteRule	路由规则数组

DtsRetryError

DTS重试事务异常信息

被如下接口引用：RetryTransactions

名称	必选	允许NULL	类型	描述
TxId	是	否	Int64	主事务ID

名称	必选	允许NULL	类型	描述
ErrorMessage	是	是	String	异常信息

AppPkgInfoV2

应用的包信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	无
PkgCount	是	是	Int64	无

ListApplicationServerResult

ListApplicationServerResult

被如下接口引用：ListApplicationServers

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ApplicationServerResult	Content

TsfPageMicroservice

微服务列表信息

被如下接口引用：DescribeMicroservices、DescribeMicroservicesByGroupIds、DescribeServiceNames

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	微服务总数目
Content	是	是	Array of Microservice	微服务列表信息

EventPolicyResult

EventPolicyResult

被如下接口引用：CreateAlarmPolicy、DescribeAlarmPolicy、ListAlarmPolicies、ModifyAlarmPolicy

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TriggerCondition	否	是	Int64	TriggerCondition
Frequency	否	是	Int64	Frequency
EventPolicyId	否	是	String	EventPolicyId
PolicyId	否	是	String	PolicyId

InvocationStatisticsV2

服务调用统计

被如下接口引用：DescribeInvocationStatistics、DescribeInvocationStatisticsRatio

名称	必选	允许NULL	类型	描述
InvocationStatisticsName	是	是	String	调用统计名称
InvocationSumQuantity	是	是	Int64	请求总数
InvocationAvgDuration	是	是	Float	请求平均耗时，单位毫秒
Invocation2xxStatusQuantity	是	是	Int64	2xx状态码响应请求数
Invocation4xxStatusQuantity	是	是	Int64	4xx状态码响应请求数
Invocation5xxStatusQuantity	是	是	Int64	5xx状态码响应请求数
InvocationOtherStatusQuantity	是	是	Int64	其它状态码响应请求数
InvocationConnectFailedQuantity	是	是	Int64	连接失败请求数
InvocationTimeoutQuantity	是	是	Int64	超时请求数
InvocationUnavailableQuantity	是	是	Int64	服务不可用请求数
InvocationQuantityProportion	是	是	Float	请求数占比

ZipkinSpanInfo

调用链Span信息

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
Id	是	是	String	Span ID
TraceId	是	是	String	调用链ID
ParentId	是	是	String	父Span ID

名称	必选	允许NULL	类型	描述
Name	是	是	String	Span名称
Timestamp	是	是	Uint64	Span时间戳
Duration	是	是	Uint64	Span耗时
ResultStatus	是	是	String	Span结果
ServiceName	是	是	String	Span服务名（前端展示）
AnnotationList	是	是	Array of ZipkinAnnotation	Span注解列表
BinaryAnnotationList	是	是	Array of ZipkinBinaryAnnotation	Span二进制注解列表
MetadataList	是	是	Array of ZipkinMetadata	Span元数据列表

GatewayDeployGroup

api分组已绑定的网关部署组

被如下接口引用：DescribeApiGroup、DescribeApiGroups、DescribeGroupBindedGateways、DescribeGroupGateways、DescribeGroupsWithPlugin、DescribeUsableGatewayGroups

名称	必选	允许NULL	类型	描述
DeployGroupId	是	是	String	网关部署组ID
DeployGroupName	是	是	String	网关部署组名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用分类：V：虚拟机应用，C：容器应用
GroupStatus	是	是	String	部署组应用状态,取值: Running、Waiting、Paused、Updating、RollingBack、Abnormal、Unknown
ClusterType	是	是	String	集群类型，C ：容器，V：虚拟机

LicenseFunction

LicenseFunction

被如下接口引用：DescribeLicenseApplications

名称	必选	允许NULL	类型	描述
Name	是	否	String	功能名
Enable	是	否	Bool	是否启用

VolumeMountInfo

容器卷挂载点信息

被如下接口引用：DeployContainerGroup

名称	必选	允许NULL	类型	描述
VolumeMountName	是	否	String	挂载数据卷名称
VolumeMountPath	是	否	String	挂载路径
VolumeMountSubPath	否	否	String	挂载子路径
ReadOrWrite	否	否	String	读写，1：读 2：读写

ConfigTemplateResult

配置模板列表

被如下接口引用：DescribeConfigTemplates

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of ConfigTemplate	配置模板列表

ImageTagsResultV2

镜像版本列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Reponame	是	否	String	仓库名,含命名空间,如tsf/ngin
Server	是	否	String	镜像服务器地址
Content	是	否	Array of ImageTag	列表信息

Provider

provider

被如下接口引用：CreateAndDownloadTemplate、CreateTemplate

名称	必选	允许NULL	类型	描述
ProviderControllerName	是	是	String	ProviderControllerName

Result

FindMonitorObject返回的结果

被如下接口引用：FindMonitorObject、FindServiceMonitorObject

名称	必选	允许NULL	类型	描述
ObjName	是	否	String	KeyWordsName 和 GroupName

MicroserviceGroup

微服务下的部署组（对服务实例按照部署组进行归类）

被如下接口引用：DescribeMicroserviceGroups

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ConsulInstanceCount	是	是	Int64	注册中心在线节点数
ClusterType	是	是	String	集群类型
AppMicroServiceType	是	是	String	应用微服务类型

TaskInstance

TaskInstance

被如下接口引用：ListScalableTasks

名称	必选	允许NULL	类型	描述
Id	是	是	String	id
Status	是	是	Int64	status

名称	必选	允许NULL	类型	描述
Mtime	是	是	String	mtime

RouteDestV2

服务路由目标规则

被如下接口引用：CreateRoute、DescribeRoute、DescribeRoutes、ModifyRoute

名称	必选	允许NULL	类型	描述
DestId	否	是	String	路由规则项路由目标ID
DestWeight	是	否	Int64	路由目标权重
DestItemList	是	是	Array of RouteDestItemV2	路由目标匹配条件列表
RouteRuleId	否	是	String	服务路由目标所属路由规则项ID

SidecarFilter

lua过滤器

被如下接口引用：DescribeSidecarFilter、DescribeSidecarFilters

名称	必选	允许NULL	类型	描述
FilterId	否	是	String	过滤器Id
FilterName	是	是	String	过滤器名称
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
Description	是	是	String	备注
WorkPosition	是	是	String	作用位置
TargetServiceNames	是	是	String	被调服务名
LuaValue	是	是	String	lua脚本值
Status	否	是	String	发布状态
CreateTime	否	是	String	创建时间
UpdateTime	否	是	String	更新时间
ReleasedTime	否	是	String	末次发布时间
AppId	否	是	String	APPID

名称	必选	允许NULL	类型	描述
Uin	否	是	String	Uin
SubAccountUin	否	是	String	subAccountUin
FilterGroupList	否	是	Array of SidecarFilterGroup	部署组列表

StdoutLogV2

标准输出日志

被如下接口引用：SearchOssRealtimeBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog、SearchStdoutLog

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID
Content	是	是	String	日志内容
Timestamp	是	是	Uint64	日志时间戳
InstanceIp	是	是	String	实例IP

GroupPodResult

部署组实例列表

被如下接口引用：DescribePodInstances

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总记录数
Content	是	是	Array of GroupPod	列表信息

AuthRuleGroup

微服务权限规则组

被如下接口引用：DescribeAuthorizationType

名称	必选	允许NULL	类型	描述
Rules	是	是	Array of AuthRule	规则列表
RuleProgram	是	是	String	规则列表计算逻辑
Type	是	是	String	权限类型，D：未启用；B：黑名单模式；W：白名单模式

名称	必选	允许NULL	类型	描述
MicroserviceId	是	是	String	微服务ID

NamespaceWrong

命名空间

被如下接口引用：

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	命名空间ID
NamespaceCode	是	是	String	命名空间编码
NamespaceName	是	是	String	命名空间名称
NamespaceDesc	是	是	String	命名空间描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型
IsDefault	是	是	String	默认命名空间
NamespaceStatus	是	是	String	命名空间状态
DeleteFlag	是	是	Bool	删除标识
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

ServiceStatisticsResult

服务统计结果

被如下接口引用：DescribeStatistics

名称	必选	允许NULL	类型	描述
Path	是	是	String	请求模版路径:type为接口时返回，服务时不返回
Method	是	是	String	请求方法:type为接口时返回，服务时不返回
MicroserviceId	是	否	String	微服务Id
MicroserviceName	是	否	String	微服务名称
RequestCount	是	否	Uint64	请求数

名称	必选	允许NULL	类型	描述
ErrorRate	是	否	Float	请求错误率，不带百分号
AvgTimeConsuming	是	否	Float	平均响应耗时ms
MetricDataCurves	是	否	Array of MetricDataCurve	响应耗时曲线
InstanceId	是	是	String	实例id
InstanceName	是	是	String	实例name
GroupId	是	是	String	部署组id
GroupName	是	是	String	部署组name
ClusterType	是	是	String	部署组类型
GroupExist	是	是	Int64	部署组是否存在
InstanceExist	是	是	Int64	实例是否存在，仅限cvm

ConfigTemplate

配置模板对象

被如下接口引用：DescribeConfigTemplate、DescribeConfigTemplates

名称	必选	允许NULL	类型	描述
ConfigTemplateId	否	是	String	配置模板Id
ConfigTemplateName	否	是	String	配置模板名称
ConfigTemplateDesc	否	是	String	配置模板描述
ConfigTemplateType	否	是	String	配置模板对应的微服务框架
ConfigTemplateValue	否	是	String	配置模板数据
CreateTime	否	是	String	创建时间
UpdateTime	否	是	String	更新时间

BillingDeal

计费订单

被如下接口引用：DescribeBillingDealRecords

名称	必选	允许NULL	类型	描述
DealName	是	是	String	订单号

名称	必选	允许NULL	类型	描述
ResourceId	是	是	String	资源ID
CategoryId	是	是	Int64	种类ID
CreateTime	是	是	String	创建时间
AppId	是	是	String	租户ID
Uin	是	是	String	账号ID
SubAccountUin	是	是	String	子账号ID
Currency	是	是	String	币种
TotalCost	是	是	Int64	原价（单位分）
RealTotalCost	是	是	Int64	折后价（单位分）
TotalPolicy	是	是	Int64	总折扣
CommonPolicy	是	是	Int64	官网折扣
UserPolicy	是	是	Int64	用户折扣
DealDetail	是	是	String	订单详情

TaskFlowBatchHistory

查询工作流执行批次的历史

被如下接口引用：

名称	必选	允许NULL	类型	描述
FlowId	是	否	String	工作流 ID
FlowLogId	是	是	String	工作流历史 ID
FlowBatchLogId	是	是	String	工作流批次历史 ID
FlowBatchId	是	是	String	工作流批次 ID
StartTime	是	是	Uint64	开始时间
EndTime	是	是	Uint64	结束时间
SpanTime	是	是	Int64	耗时
HistoryCount	是	否	Int64	历史数
State	是	是	String	状态
TriggerType	是	是	String	触发类型

名称	必选	允许NULL	类型	描述
ScheduleFireTime	是	是	Int64	下次开始时间
FlowName	是	是	String	工作流名称

ApplicationServerResult

ApplicationServerResult

被如下接口引用：ListApplicationServers

名称	必选	允许NULL	类型	描述
ServerId	是	是	String	ServerId
ClusterId	是	是	String	ClusterId
GroupId	是	是	String	GroupId
Ip	是	是	String	Ip
Type	是	是	Int64	Type
TaskStatus	是	是	Int64	TaskStatus
AgentStatus	是	是	Int64	AgentStatus
MasterId	是	是	Int64	MasterId
GroupDesc	是	是	String	GroupDesc
BasicEnvDesc	是	是	String	BasicEnvDesc
ApplicationType	是	是	String	ApplicationType
AgentVersion	是	是	String	AgentVersion

ContainerTasksV2

变更记录任务

被如下接口引用：

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	taskId
TaskTime	是	是	String	任务开始时间
Type	是	是	Int64	任务类型字段，0：没有任务（在此接口中，不用出现0），1：发布程序包；2.部署操作；3.扩容操作；4.启动操作；5.停止操作；6.缩容操作；7.发布日志配置,8.删除销毁操作

名称	必选	允许NULL	类型	描述
Status	是	是	Int64	变更状态，0：成功 1:失败 2：执行中
GroupId	是	是	String	分组id
GroupName	是	是	String	分组名称
ImageName	是	是	String	镜像名称
ImageVersion	是	是	String	镜像版本
TaskDesc	是	是	String	任务详情描述
TotalCount	是	是	Int64	任务总个数
SuccessCount	是	是	Int64	成功任务个数
FailCount	是	是	Int64	失败任务个数

AuthMicroserviceV2

在 TSF Auth 中使用的微服务结构体

被如下接口引用：

名称	必选	允许NULL	类型	描述
MicroserviceId	是	否	String	服务 ID
MicroserviceName	是	否	String	服务名
NamespaceId	是	否	String	命名空间 ID
IsNotExist	是	否	Bool	服务是否存在
AppId	是	否	String	帐号 appid
Uin	是	否	String	账号 uin
SubAccountUin	是	否	String	子账号 uin

TaskFlowLogPage

工作流列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	数据总数
Content	是	否	Array of TaskFlowLog	工作流数据列表

ContainerInfo

容器详细信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
Name	是	是	String	容器名
ContainerId	是	是	String	容器ID
Status	是	是	String	容器状态
Reason	是	是	String	容器的Reason
Image	是	是	String	镜像地址

RouteReleaseHistory

路由规则启停记录

被如下接口引用：DescribeRouteReleaseHistory

名称	必选	允许NULL	类型	描述
RouteReleaseLogId	是	是	String	路由规则发布记录id
MicroserviceId	是	是	String	微服务id
RouteRuleId	是	是	String	路由规则Id
EnableTime	是	是	String	路由规则部署开始时间
DisableTime	是	是	String	路由规则停止时间
ReleaseDesc	是	是	String	路由规则发布
RouteRule	是	是	RouteRule	路由规则详情
AppId	是	是	String	账号APPID
Uin	是	是	String	账号Owner用户唯一ID
SubAccountUin	是	是	String	账号用户唯一ID

PkgInfoV2

包信息

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
PkgId	是	是	String	无
PkgName	是	是	String	无
PkgType	是	是	String	无
PkgVersion	是	是	String	无
PkgDesc	是	是	String	无
UploadTime	是	是	String	无
Md5	是	是	String	无
PkgPubStatus	是	是	Int64	无

WeightRouteItemList

权重路由规则项

被如下接口引用：CreateRouteRule、DescribeRouteReleaseHistory、DescribeRouteRule、DescribeRouteRules、ModifyRouteRule

名称	必选	允许NULL	类型	描述
WeightRouteId	否	否	String	权重路由规则项Id
SourcePercent	是	否	Int64	权重路由，百分比字段
TargetField	是	否	String	权重路由规则匹配目标字段
TargetValue	是	否	String	权重路由规则匹配目标取值
RouteRuleId	否	否	String	权重路由规则所属路由Id

BillingConfig

计费配置

被如下接口引用：DescribeBillingConfig

名称	必选	允许NULL	类型	描述
Enable	是	否	Bool	是否开启计费

DailyUseStatistics

日使用统计对象

被如下接口引用：DescribeGatewayDailyUseStatistics

名称	必选	允许NULL	类型	描述
TotalRequestCount	是	否	DailyUseStatisticsEntity	总调用数
AvgErrorRate	是	否	DailyUseStatisticsEntity	平均错误率
AvgResponseCost	是	否	DailyUseStatisticsEntity	平均响应耗时

InvocationMetricDataPoint

InvocationMetricDataPoint

被如下接口引用：DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
Key	是	是	String	Key
Value	是	是	String	Value
Tag	是	是	String	Tag

CloudMonitorMicroservice

CloudMonitorMicroservice

被如下接口引用：DescribeMicroServiceList、ListCloudMicroServiceFindPagedList

名称	必选	允许NULL	类型	描述
NamespaceId	否	否	String	namespaceId
NamespaceName	否	否	String	namespaceName
MicroserviceId	否	否	String	microserviceId
MicroserviceName	否	否	String	microserviceName

TsfApiListResponse

TsfApiListResponse

被如下接口引用：DescribeInterfaceList、DescribeMsApiList

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	数量
Content	是	是	Array of MsApiArray	API 列表

OverviewConfig

概览页配置

被如下接口引用：DescribeResourceUsageConfig

名称	必选	允许NULL	类型	描述
ClusterThreshold	是	否	Uint64	集群显示阈值

ImageUserIsExists

镜像仓库用户是否已经开通

被如下接口引用：DescribeImageUserIsExists

名称	必选	允许NULL	类型	描述
IsExist	是	否	Bool	子账号是否存在,true：存在；false：不存在
MainIsExist	是	否	Bool	主账号是否存在，true：存在；false：不存在

SimpleCluster

简单集群

被如下接口引用：

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型

OperationInfoDetail

提供给前端控制按钮显示逻辑的字段

被如下接口引用：DescribeCluster、DescribeClusterInstanceCount、DescribeClusters、DescribeNamespaces、DescribeSimpleClusters、DescribeSimpleNamespaces

名称	必选	允许NULL	类型	描述
DisabledReason	是	是	String	不显示的原因
Enabled	是	是	Bool	该按钮是否可点击
Supported	是	是	Bool	是否显示该按钮

TsfPageRouteReleaseHistory

路由规则启停记录分页对象

被如下接口引用：DescribeRouteReleaseHistory

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of RouteReleaseHistory	路由规则部署记录列表信息

VmTaskV2

虚拟机操作记录

被如下接口引用：

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	任务Id
CreateTime	是	是	String	创建时间
TaskType	是	是	Int64	任务类型
GroupId	是	是	String	分组Id
GroupName	是	是	String	分组名称
PkgId	是	是	String	程序包Id
PkgName	是	是	String	程序包名称
PkgVersion	是	是	String	程序包版本
TaskDesc	是	是	String	任务描述
TotalCount	是	是	Int64	子任务数目
SuccessCount	是	是	Int64	成功的子任务数目
RunCount	是	是	Int64	运行中的子任务数目
FailCount	是	是	Int64	失败的子任务数目

OperationInfo

提供给前端，控制按钮是否显示

被如下接口引用：DescribeCluster、DescribeClusterInstanceCount、DescribeClusters、DescribeNamespaces、DescribeSimpleClusters、DescribeSimpleNamespaces

名称	必选	允许NULL	类型	描述
Init	是	是	OperationInfoDetail	初始化按钮的控制信息
AddInstance	是	是	OperationInfoDetail	添加实例按钮的控制信息
Destroy	是	是	OperationInfoDetail	销毁机器的控制信息

PagedProductHelp

产品帮助分组分页对象

被如下接口引用：DescribeProductHelp

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of ProductHelp	产品帮助列表

LaneRules

泳道规则分页查询

被如下接口引用：DescribeLaneRules

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数
Content	是	否	Array of LaneRule	泳道规则列表

MsApplication

应用

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	否	是	String	应用 ID，如 application-qv3dkda7
ApplicationName	否	是	String	应用名
ApplicationDesc	否	是	String	应用描述
ApplicationType	否	是	String	应用 *部署* 类型，V 表示 CVM 应用，C 表示容器应用
ApplicationJvmArg	否	是	String	应用启动时的 JVM 参数
MicroserviceType	否	是	String	应用类型，M 表示 Mesh 应用，N 表示 Spring Cloud 应用

名称	必选	允许NULL	类型	描述
ProgLang	否	是	String	应用编程语言，J 表示 Java，P 表示 Python
NotExist	否	是	Bool	应用是否存在

ServiceSetting

容器网络设置。

被如下接口引用：DeployContainerGroup

名称	必选	允许NULL	类型	描述
AccessType	是	是	Int64	0:公网, 1:集群内访问, 2：NodePort, 3: VPC 内网访问
ProtocolPorts	是	是	Array of ProtocolPort	容器端口映射
SubnetId	是	是	String	子网ID

ApplicationAttribute

应用列表其它字段

被如下接口引用：DescribeApplicationAttribute、DescribeApplicationsAttribute、DescribeApplicationsOther

名称	必选	允许NULL	类型	描述
InstanceCount	是	是	Int64	总实例个数
RunInstanceCount	是	是	Int64	运行实例个数
GroupCount	是	是	Int64	应用下部署组个数

DumpResult

DumpResult

被如下接口引用：GetDump

名称	必选	允许NULL	类型	描述
Name	否	是	String	Name
State	否	是	String	State
Detail	否	是	String	Detail

RatelimitRuleForUpdateV2

用于修改删除操作的限流规则

被如下接口引用：UpdateRatelimit

名称	必选	允许NULL	类型	描述
Id	是	否	String	规则ID
Name	否	否	String	规则名字，在一个微服务下唯一
Status	否	否	Uint64	状态 0表示启用 1表示停用
DurationSecond	否	否	Uint64	限流周期，单位秒
DurationQuota	否	否	Uint64	每周期内的限流配额
Description	否	否	String	描述
Dimensions	否	否	Array of RatelimitDimension	标签规则

SimpleGroupV2

部署组

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称

VmGroupForPage

虚拟机部署组列表中使用到的虚拟机信息字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群Id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间Id
NamespaceName	是	是	String	命名空间名称
GroupId	是	是	String	分组id
GroupName	是	是	String	分组名称
GroupDesc	是	是	String	分组描述
CreateTime	是	是	String	分组创建时间
UpdateTime	是	是	String	分组更新时间
StartupParameters	是	是	String	部署组启动参数
PkgId	是	是	String	程序包Id

TsfPageInvocationStatisticsV2

服务调用统计查询分页结果

被如下接口引用：DescribeInvocationStatistics

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数量
Content	是	是	Array of InvocationStatisticsV2	内容列表

ImageRepositoryV2

镜像仓库

被如下接口引用：

名称	必选	允许NULL	类型	描述
Reponame	是	是	String	仓库名,含命名空间,如tsf/nginx
Repotype	是	是	String	仓库类型

名称	必选	允许NULL	类型	描述
TagCount	是	是	Int64	镜像版本数
IsPublic	是	是	Int64	是否公共,1:公有,0:私有
IsUserFavor	是	是	Bool	是否被用户收藏。true：是，false：否
IsQcloudOfficial	是	是	Bool	是否是腾讯云官方仓库。 是否是腾讯云官方仓库。true：是，false：否
FavorCount	是	是	Int64	被所有用户收藏次数
PullCount	是	是	Int64	拉取次数
Description	是	是	String	描述内容
CreationTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

PagedProgram

tsf-privilege模块，分页数据集列表

被如下接口引用：DescribePrograms

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of Program	数据集列表

Controller

tsf模板

被如下接口引用：CreateAndDownloadTemplate、CreateTemplate

名称	必选	允许NULL	类型	描述
ControllerName	是	是	String	ControllerName

Cluster

集群

被如下接口引用：DescribeClusterInstanceCount、DescribeNamespaces、DescribeSimpleClusters、DescribeSimpleNamespaces

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID

名称	必选	允许NULL	类型	描述
ClusterName	是	是	String	集群名称
ClusterDesc	是	是	String	集群描述
ClusterType	是	是	String	集群类型
VpcId	是	是	String	集群所属私有网络ID
ClusterStatus	是	是	String	集群状态
ClusterCIDR	是	是	String	集群CIDR
ClusterTotalCpu	是	是	Float	集群总CPU，单位: 核
ClusterTotalMem	是	是	Float	集群总内存，单位: G
ClusterUsedCpu	是	是	Float	集群已使用CPU，单位: 核
ClusterUsedMem	是	是	Float	集群已使用内存，单位: G
InstanceCount	是	是	Int64	集群机器实例数量
RunInstanceCount	是	是	Int64	集群可用的机器实例数量
NormalInstanceCount	是	是	Int64	集群正常状态的机器实例数量
DeleteFlag	是	是	Bool	删除标记：true：可以删除；false：不可删除
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
TsfRegionId	是	是	String	集群所属TSF地域ID
TsfRegionName	是	是	String	集群所属TSF地域名称
TsfZoneId	是	是	String	集群所属TSF可用区ID
TsfZoneName	是	是	String	集群所属TSF可用区名称
DeleteFlagReason	是	是	String	集群不可删除的原因
ClusterLimitCpu	是	是	Float	集群最大CPU限制，单位：核
ClusterLimitMem	是	是	Float	集群最大内存限制，单位：G
RunServiceInstanceCount	是	是	Int64	集群可用的服务实例数量
SubnetId	是	是	String	集群所属子网ID
OperationInfo	是	是	OperationInfo	返回给前端的控制信息
ClusterVersion	是	是	String	集群版本

FunctionStatusResult

功能状态返回结果

被如下接口引用：DescribeFunctionStatus

名称	必选	允许NULL	类型	描述
Enable	是	是	Bool	是否开启
Used	是	是	Bool	是否使用过
ForbiddenActions	是	是	Array of String	关闭时需要禁用的Action列表
FunctionName	是	是	String	功能名称

ContainerGroupDetail

容器部署组详情

被如下接口引用：DescribeContainerGroupDetail

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	分组名称
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数
CreateTime	是	是	String	创建时间
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用ID
LbIp	是	是	String	负载均衡ip
ApplicationType	是	是	String	应用类型
ClusterIp	是	是	String	Service ip
NodePort	是	是	Int64	NodePort端口，只有公网和NodePort访问方式才有值

名称	必选	允许NULL	类型	描述
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
AccessType	是	是	Uint64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
Envs	是	是	Array of Env	环境变量数组对象
ApplicationName	是	是	String	应用名称
Message	是	是	String	pod错误信息描述
Status	是	是	String	部署组状态
MicroserviceType	是	是	String	服务类型
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id
GroupResourceType	是	是	String	部署组资源类型
InstanceCount	是	是	Uint64	部署组实例个数
UpdatedTime	是	是	Int64	部署组更新时间戳
MaxSurge	是	是	String	kubernetes滚动更新策略的MaxSurge参数
MaxUnavailable	是	是	String	kubernetes滚动更新策略的MaxUnavailable参数
HealthCheckSettings	是	是	HealthCheckSettings	部署组健康检查设置

TsfPageConfigV2

TsfPage<Config>

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TsfPageConfig
Content	是	否	Array of Config	配置项列表

ProductNews

产品动态

被如下接口引用：DescribeProductNews、ReleaseProductNews

名称	必选	允许NULL	类型	描述
NewsId	否	是	String	id
Content	否	是	String	内容
Link	否	是	String	链接
Order	否	是	Int64	排序
StyleFlag	否	是	String	样式Flag
Valid	否	是	Int64	是否有效
ValidTime	否	是	String	生效时间（当valid从0 -> 1，刷新生效时间）
Title	否	是	String	标题
ServiceTag	否	是	String	产品标识(默认为tsf)

PathRewriteCreateObject

路径重写创建对象

被如下接口引用：CreatePathRewrites

名称	必选	允许NULL	类型	描述
GatewayGroupId	是	否	String	网关部署组ID
Regex	是	否	String	正则表达式
Replacement	是	否	String	替换的内容
Blocked	是	否	String	是否屏蔽映射后路径，Y: 是 N: 否
Order	是	否	Int64	规则顺序，越小优先级越高

TaskExecuteHistoryRecordPage

任务执行记录的流水列表

被如下接口引用：DescribeTaskExecuteHistoryRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	分页总数

名称	必选	允许NULL	类型	描述
Content	是	否	Array of TaskExecuteHistoryRecord	分页对象

VmGroupForPageV2

虚拟机部署组列表中使用到的虚拟机信息字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群Id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间Id
NamespaceName	是	是	String	命名空间名称
GroupId	是	是	String	分组id
GroupName	是	是	String	分组名称
GroupDesc	是	是	String	分组描述
CreateTime	是	是	String	分组创建时间
UpdateTime	是	是	String	分组更新时间
StartupParameters	是	是	String	部署组启动参数
PkgId	是	是	String	程序包Id

TsfPageVmSubTaskV2

虚拟机子任务分页信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	子任务总数目
SuccessCount	是	是	Int64	成功子任务数目
RunCount	是	是	Int64	运行中子任务数目

名称	必选	允许NULL	类型	描述
FailCount	是	是	Int64	失败子任务数目
Content	是	是	Array of VmSubTask	子任务列表信息

ScalableSubRuleV2

弹性扩缩容监控指标类型

被如下接口引用：

名称	必选	允许NULL	类型	描述
Indicators	是	否	Uint64	监控指标，如监控CPU和MEM，则为百分比0-100；监控RT，则为响应时间(ms), 0-99999999
IndicatorType	是	否	Uint64	规则指标类型: 1:CPU, 2:MEM, 3:RT

ListScalableRuleResult

ListScalableRuleResult

被如下接口引用：ListScalableRule

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ScalableRule	Content

CircuitBreakerRule

熔断规则

被如下接口引用：DescribeCircuitBreakerRule、DescribeCircuitBreakerRules

名称	必选	允许NULL	类型	描述
RuleId	否	是	String	熔断规则主键
MicroserviceId	是	是	String	熔断规则微服务ID
MicroserviceName	是	是	String	微服务服务名称
NamespaceId	是	是	String	微服务所属命名空间id
TargetServiceName	是	是	String	目标服务名
TargetNamespaceId	是	是	String	目标服务所属命名空间ID

名称	必选	允许NULL	类型	描述
StrategyList	是	是	Array of CircuitBreakerStrategy	熔断策略
IsolationLevel	是	是	String	熔断级别
Enable	否	是	Bool	是否开启此规则
CreateTime	否	是	Int64	创建时间
UpdateTime	是	是	Int64	更新时间
AppId	否	是	String	APPID
Uin	否	是	String	Uin
SubAccountUin	否	是	String	subAccountUin
TargetNamespaceName	是	是	String	目标服务所属命名空间

TsfPageVmTask

虚拟机任务分页信息

被如下接口引用：DscribeTasks

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	任务总数目
Content	是	是	Array of VmTask	任务详情列表

ExpandSubrule

ExpandSubrule

被如下接口引用：DescribeScalableRuleAttribute

名称	必选	允许NULL	类型	描述
Indicators	是	是	Int64	Indicators
IndicatorType	是	是	Int64	IndicatorType
RuleType	是	是	Int64	RuleType

GatewayWeChatMiniProgramLoginPlugin

微服务网关微信登录插件

被如下接口引用：DescribeGatewayWeChatMiniProgramLoginPlugin

名称	必选	允许NULL	类型	描述
Id	是	否	String	网关插件ID
Name	是	否	String	插件名称
Description	是	是	String	插件描述
Type	是	否	String	插件类型
CreatedTime	是	否	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	否	String	插件更新时间 如:2019-06-20 15:51:28
Status	是	否	String	发布状态: drafted/released
WeChatAppId	是	否	String	微信小程序AppId
RequestCodeBaggagePosition	是	否	String	微信小程序请求code携带位置：header/cookie
SessionKeyName	是	否	String	自定义登录态参数名
SessionExpireTime	是	否	Int64	自定义登录态过期时间，单位：秒
RequestSessionBaggagePosition	是	否	String	前台业务请求自定义登录态参数位置：header/cookie
BusinessSessionBaggagePosition	是	否	String	向业务后台传输登录态参数位置：header/query/cookie
ResponseSessionBaggagePosition	是	否	String	返回自定义登录态参数位置：header
MetaDataTagInfoList	是	是	String	元数据转标签配置的Json串
CustomizeTagInfoList	是	是	String	自定义标签配置的Json串

GroupsByScalableRuleIdV2

GroupsByScalableRuleId

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	ApplicationId
ApplicationName	是	是	String	ApplicationName
GroupId	是	是	String	GroupId
GroupName	是	是	String	GroupName
Status	是	是	Int64	Status
UpdateTime	是	是	String	UpdateTime
RuleId	是	是	String	弹性伸缩规则id

ImageTagsResult

镜像版本列表

被如下接口引用：DescribeImageTags

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
RepoName	是	否	String	仓库名,含命名空间,如tsf/ngin
Server	是	否	String	镜像服务器地址
Content	是	否	Array of ImageTag	列表信息

ThreadDetail

DescribeThreadDetail 接口出参封装

被如下接口引用：DescribeThreadDetail

名称	必选	允许NULL	类型	描述
ThreadName	是	否	String	线程名称
ThreadState	是	否	String	线程状态
ThreadInfos	是	否	String	线程堆栈信息

TsfBusinessLog

容器日志输出

被如下接口引用：SearchContainerStdoutLog

名称	必选	允许NULL	类型	描述
RealtimeTs	否	是	Uint64	时间戳，入参传入，若不为实时日志，可能为空
ScrollId	否	是	Uint64	游标ID
BusinessLogSet	否	是	Array of BusinessLog	容器日志输出列表

TsfPageInstance

TSF机器实例分页对象

被如下接口引用：DescribeAddibleInstances、DescribeClusterInstances、DescribeGroupAddibleInstances、DescribeGroupInstances、DescribeSimpleInstances

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	机器实例总数目
Content	是	是	Array of Instance	机器实例列表

Service

tsf-privilege模块，产品

被如下接口引用：DescribeServices

名称	必选	允许NULL	类型	描述
ServiceId	否	是	String	产品ID
ServiceCode	否	是	String	产品编码
ServiceName	否	是	String	产品名称
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	更新时间
DeleteFlag	否	是	Bool	删除标识，true: 可以删除；false: 不可删除

UnitRuleTag

微服务网关单元化规则标签

被如下接口引用：CreateUnitRule、DescribeEnabledUnitRule、DescribeUnitRule、DescribeUnitRules、UpdateUnitRule

名称	必选	允许NULL	类型	描述
UnitRuleItemId	否	是	String	单元化规则项ID
Id	否	是	String	规则ID
TagType	是	否	String	标签类型：U(用户标签)
TagField	是	否	String	标签名
TagOperator	是	否	String	操作符:IN/NOT_IN/EQUAL/NOT_EQUAL/REGEX
TagValue	是	否	String	标签值

TaskFlowLastBatchState

工作流最近批次的状态

被如下接口引用：DescribeFlowLastBatchState

名称	必选	允许NULL	类型	描述
FlowBatchId	是	是	String	批次ID
FlowBatchLogId	是	是	String	批次历史ID
State	是	是	String	状态,WAITING/SUCCESS/FAILED/RUNNING/TERMINATING

InterfaceRequests

多条接口请求详情

被如下接口引用：DescribeInterfaceRequest

名称	必选	允许NULL	类型	描述
Content	是	是	Array of InterfaceRequest	interfaceRequest数组
TotalCount	是	是	Int64	总条数

ContainerGroupDeployV2

获取部署组

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	groupId
GroupName	是	是	String	分组名称
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	是	是	Array of ProtocolPort	ProtocolPorts

名称	必选	允许NULL	类型	描述
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
JvmOpts	是	是	String	jvm参数
SubnetId	是	是	String	子网id

DescribeResourceConfigResultV2

DescribeResourceConfig

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Sts	是	是	DescribeResourceConfigSts	STS参数配置
License	是	是	DescribeResourceConfigLicense	许可信息
Group	是	是	GroupResourceConfig	部署组相关的参数配置
Instance	是	是	InstanceResourceConfig	实例相关的参数配置
Cluster	是	是	DescribeResourceConfigCluster	Cluster相关配置信息
Package	是	是	PackageConfig	程序包相关配置信息

MetricDataPointMap

监控统计数据点Map集合（单元化网关使用）

被如下接口引用：DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
SumReqAmount	是	否	Array of InvocationMetricDataPoint	总调用次数监控数据点集合
AvgFailureRate	是	否	Array of InvocationMetricDataPoint	平均错误率监控数据点集合
AvgTimeCost	是	否	Array of InvocationMetricDataPoint	平均响应时间监控数据点集合

ConfigReleaseV2

配置项发布信息

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ConfigReleaseId	是	是	String	配置项发布ID
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
ReleaseTime	是	是	String	发布时间
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ReleaseDesc	是	是	String	发布描述

DtsAuth

分布式事务鉴权

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	事务分组ID
AppId	是	否	String	租户ID
Uin	是	否	String	主账号ID
SubAccountUin	是	否	String	子账号ID

TsfPageMicroserviceGroup

微服务下部署组列表

被如下接口引用：DescribeMicroserviceGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	数据总数
Content	是	是	Array of MicroserviceGroup	翻页数据

Consumer

Consumer

被如下接口引用：CreateAndDownloadTemplate、CreateTemplate

名称	必选	允许NULL	类型	描述
ConsumerControllerName	是	是	String	ConsumerControllerName
ToProviderName	是	是	String	ToProviderName
ToProviderServiceName	是	是	String	ToProviderServiceName

ProtocolPortsV2

端口对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
Protocol	是	否	String	TCP UDP
Port	是	否	Int64	服务端口
TargetPort	是	否	Int64	容器端口
NodePort	否	是	Int64	主机端口

GraphEdgeV2

依赖拓扑图边

被如下接口引用：GetOssTopologyGraph、GetTopologyGraph

名称	必选	允许NULL	类型	描述
SourceName	是	是	String	来源服务名
TargetName	是	是	String	目标服务名
SourceType	是	是	String	来源服务类型
TargetType	是	是	String	目标服务类型
ReqTotalQty	是	是	Uint64	服务间总请求量
ReqAvgQty	是	是	Float	请求平均数，次每分钟，两位小数
ReqSuccessRate	是	是	Float	请求成功率，两位小数

名称	必选	允许NULL	类型	描述
ReqAvgDuration	是	是	Float	请求平均耗时，单位毫秒，两位小数
Apdex	是	是	Float	健康度指标
SourceId	是	是	String	来源节点标识
TargetId	是	是	String	目标节点标识
SourceNamespaceId	是	是	String	来源命名空间ID
TargetNamespaceId	是	是	String	目标命名空间ID

UnitRule

微服务网关单元化规则

被如下接口引用：DescribeEnabledUnitRule、DescribeUnitRule、DescribeUnitRules

名称	必选	允许NULL	类型	描述
Id	否	是	String	规则ID
GatewayInstanceId	否	是	String	网关实体ID
Name	是	否	String	规则名称
Description	否	是	String	规则描述
Status	否	是	String	使用状态：enabled/disabled
UnitRuleItemList	否	是	Array of UnitRuleItem	规则项列表

Instance

机器实例

被如下接口引用：DescribeAddibleInstances、DescribeClusterInstances、DescribeGroupAddibleInstances、DescribeGroupInstances、DescribeSimpleInstances

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器实例ID
InstanceName	是	是	String	机器名称
LanIp	是	是	String	机器内网地址IP
WanIp	是	是	String	机器外网地址IP
InstanceDesc	是	是	String	机器描述信息

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
InstanceStatus	是	是	String	VM的状态 虚机：虚机的状态 容器：Pod所在虚机的状态
InstanceAvailableStatus	是	是	String	VM的可使用状态 虚机：虚机是否能够作为资源使用 容器：虚机是否能够作为资源部署POD
ServiceInstanceStatus	是	是	String	服务下的服务实例的状态 虚机：应用是否可用 + Agent状态 容器：Pod状态
CountInTsf	是	是	Int64	标识此instance是否已添加在tsf中
GroupId	是	是	String	机器所属部署组ID
ApplicationId	是	是	String	机器所属应用ID
ApplicationName	是	是	String	机器所属应用名称
InstanceCreatedTime	是	是	String	机器实例在CVM的创建时间
InstanceExpiredTime	是	是	String	机器实例在CVM的过期时间
InstanceChargeType	是	是	String	机器实例在CVM的计费模式
InstanceTotalCpu	是	是	Float	机器实例总CPU信息
InstanceTotalMem	是	是	Float	机器实例总内存信息
InstanceUsedCpu	是	是	Float	机器实例使用的CPU信息
InstanceUsedMem	是	是	Float	机器实例使用的内存信息
InstanceLimitCpu	是	是	Float	机器实例Limit CPU信息
InstanceLimitMem	是	是	Float	机器实例Limit 内存信息
InstancePkgVersion	是	是	String	包版本
ClusterType	是	是	String	集群类型
RestrictState	是	是	String	机器实例业务状态
UpdateTime	是	是	String	更新时间
OperationState	是	是	Int64	实例执行状态
NamespaceId	是	是	String	NamespaceId Ns ID
InstanceZoneId	是	是	String	InstanceZoneId 可用区ID
InstanceImportMode	是	是	String	InstanceImportMode 导入模式
ApplicationType	是	是	String	ApplicationType应用类型
ApplicationResourceType	是	是	String	ApplicationResourceType 资源类型

名称	必选	允许NULL	类型	描述
ServiceSidecarStatus	是	是	String	sidecar状态
GroupName	是	是	String	部署组名
NamespaceName	是	是	String	NS名
Reason	是	是	String	健康检查原因
AgentVersion	是	是	String	agent版本
InstanceCpuType	否	是	String	实例cpu架构

TsfPageGatewayDeployGroup

GatewayDeployGroup 翻页对象

被如下接口引用：DescribeGroupBindedGateways、DescribeUsableGatewayGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	记录总数
Content	是	否	Array of GatewayDeployGroup	记录实体列表

TsfPageFileConfig

文件配置项列表

被如下接口引用：DescribeFileConfigSummary、DescribeFileConfigs

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of FileConfig	文件配置数组

ProjectList

ProjectList

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	工程id
ProjectName	是	否	String	工程名
BasePackage	是	否	String	包路径

名称	必选	允许NULL	类型	描述
LastTime	是	否	Int64	修改时间
AppId	否	否	String	AppId
Uin	否	否	String	Uin
SubAccountUin	否	否	String	SubAccountUin
Data	否	否	String	Data

ResultV2

FindMonitorObject返回的结果

被如下接口引用：

名称	必选	允许NULL	类型	描述
ObjName	是	否	String	KeyWordsName 和 GroupName

TsfPageInstanceV2

TSF机器实例分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of Instance	机器实例列表

IndicatorCoord

监控指标坐标

被如下接口引用：DescribeInovcationIndicators

名称	必选	允许NULL	类型	描述
CoordX	是	是	String	指标横坐标值
CoordY	是	是	String	指标纵坐标值
CoordTag	是	是	String	指标标签，用于标识附加信息

DescribeTemplateResult

DescribeTemplateResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	ProjectId
ProjectName	是	否	String	ProjectName
BasePackage	是	否	String	BasePackage
PomGroupId	是	否	String	PomGroupId
PomArtifactId	是	否	String	PomArtifactId
PomVersion	是	否	String	PomVersion
PomName	是	否	String	PomName
PomDesc	是	否	String	PomDesc
GetMethod	是	否	String	GetMethod
Ms	是	否	Array of Ms	Ms
Action	否	否	String	action
AppId	否	否	String	appId
Uin	否	否	String	Uin
SubAccountUin	否	否	String	SubAccountUin

BusinessLogConfigV2

业务日志配置

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigId	否	否	String	配置项ID
ConfigName	否	否	String	配置项名称
ConfigPath	否	是	String	配置项日志路径
ConfigDesc	否	是	String	配置项描述
ConfigTags	否	是	String	配置项标签
ConfigPipeline	否	是	String	配置项对应的ES管道
ConfigCreateTime	否	是	String	配置项创建时间
ConfigUpdateTime	否	是	String	配置项更新时间

名称	必选	允许NULL	类型	描述
ConfigSchema	否	是	BusinessLogConfigSchema	配置项解析规则
ConfigAssociatedGroups	否	是	Array of BusinesLogConfigAssociatedGroup	配置项关联部署组

ConsumerV2

Consumer

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConsumerControllerName	是	是	String	ConsumerControllerName
ToProviderName	是	是	String	ToProviderName
ToProviderServiceName	是	是	String	ToProviderServiceName

Program

tsf-privilege模块 Program数据集

被如下接口引用：DescribeProgram、DescribePrograms

名称	必选	允许NULL	类型	描述
ProgramId	否	是	String	数据集ID
ProgramName	否	是	String	数据集名称
ProgramDesc	否	是	String	数据集描述
DeleteFlag	否	是	Bool	删除标识，true: 可以删除; false: 不可删除
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间
ProgramItemList	否	是	Array of ProgramItem	数据项列表，无值时返回空数组

ZipkinBinaryAnnotationV2

调用链Span二进制注解

被如下接口引用：GetOssTraceSpans

名称	必选	允许NULL	类型	描述
Key	是	是	String	注解键

名称	必选	允许NULL	类型	描述
Value	是	是	String	注解值
Endpoint	是	是	ZipkinEndpointV2	注解端点信息

TsfZoneResult

可用区信息

被如下接口引用：DescribeTsfmanagerZones、DescribeZones

名称	必选	允许NULL	类型	描述
TZoneId	否	是	String	tZoneId
TZoneName	否	是	String	tZoneName
TRemark	否	是	String	tRemark
TRegionId	否	是	String	tRegionId

TsfPageAuthorization

虽然挂着 TsfPage，但是它没有继承 TsfPage 类（后面可能会）。这是个全量接口。

被如下接口引用：DescribeAuthorizationInfo

名称	必选	允许NULL	类型	描述
IsEnabled	是	否	Bool	鉴权规则是否被启用
Conditions	是	是	Array of AuthCondition	鉴权规则列表

AuthMicroservice

在 TSF Auth 中使用的微服务结构体

被如下接口引用：DescribeAuthorizationInfo、ModifyAuthorizationInfo

名称	必选	允许NULL	类型	描述
MicroserviceId	是	否	String	服务 ID
MicroserviceName	是	否	String	服务名
NamespaceId	是	否	String	命名空间 ID
IsNotExist	是	否	Bool	服务是否存在
AppId	是	否	String	帐号 appid

名称	必选	允许NULL	类型	描述
Uin	是	否	String	账号 uin
SubAccountUin	是	否	String	子账号 uin

CosUploadInfo

cos上传所需信息

被如下接口引用：CheckUploadInfo、DescribeUploadInfo、GetUploadInfo

名称	必选	允许NULL	类型	描述
PkgId	是	是	String	程序包ID
Bucket	是	是	String	桶
Region	是	是	String	目标地域
Path	是	是	String	存储路径
Credentials	是	否	CosCredentials	鉴权信息

SecretKeyInfo

微服务网关密钥信息

被如下接口引用：ChangeSecretStatus、DescribeGroupSecrets、UpdateGroupSecret

名称	必选	允许NULL	类型	描述
SecretName	是	是	String	secret 名称
SecretId	是	是	String	secret Id
SecretKey	是	是	String	secret key
CreatedTime	是	是	String	创建时间，格式 yyyy-MM-dd HH:mm:ss
Status	是	是	String	enabled: 启用状态。disabled: 禁用状态
Id	是	是	String	密钥ID
GwSecretId	是	是	String	secret Id
ExpiredTime	是	是	String	过期时间，格式 yyyy-MM-dd HH:mm:ss
UpdatedTime	是	是	String	更新时间，格式 yyyy-MM-dd HH:mm:ss
GroupId	是	是	String	API分组ID

TraceSpanDetail

span查询的出参

被如下接口引用：DescribeSpanDetail

名称	必选	允许NULL	类型	描述
TraceId	否	否	String	调用链traceid
SpanId	否	否	String	调用链spanid
ParentId	否	否	String	调用链parentid
Kind	否	否	String	调用链类型
Name	否	否	String	调用链name
Timestamp	否	否	Uint64	时间戳
Duration	否	否	Float	耗时
ResultStatus	否	否	String	状态
LocalComponent	否	否	String	组件
LocalEndpoint	否	否	ZipkinEndpointV2	local端点信息
RemoteEndpoint	否	否	ZipkinEndpointV2	remote端点信息
TagList	否	否	Array of SpanLabel	tags列表
GroupId	否	否	String	部署组id
GroupName	否	否	String	部署组名称
ClusterType	否	否	String	集群类型
NamespaceId	否	否	String	命名空间id
NamespaceName	否	否	String	命名空间名称
ResultCode	否	否	String	状态码
LaneId	否	否	String	泳道id
LaneName	否	否	String	泳道名称
CustomTagList	否	否	Array of SpanLabel	用户自定义metadata列表
ServiceName	否	否	String	服务名
ServiceId	否	否	String	服务Id

InterfaceRequest

单条接口请求详情

被如下接口引用：DescribeInterfaceRequest

名称	必选	允许NULL	类型	描述
TraceId	是	是	String	trace id
Timestamp	是	是	Int64	起始时间
Duration	是	是	Int64	耗时
ResultStatus	是	是	String	结果状态
ResultError	是	是	String	错误
InstanceInfo	是	是	InstanceInfo	实例信息

CosCredentials

cos临时帐号信息

被如下接口引用：CheckUploadInfo、DescribeDownloadInfo、DescribeUploadInfo、GetDownloadInfo、GetUploadInfo

名称	必选	允许NULL	类型	描述
SessionToken	是	是	String	会话Token
TmpAppId	是	是	String	临时应用ID
TmpSecretId	是	是	String	临时调用者身份ID
TmpSecretKey	是	是	String	临时密钥
ExpiredTime	是	是	Int64	过期时间
Domain	是	是	String	所在域

InstanceResourceConfig

实例相关的参数配置

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Container	是	是	ContainerInstanceResourceConfig	容器实例相关的参数配置
Vm	是	是	VmInstanceResourceConfig	虚拟机实例相关的参数配置

HealthCheckConfig

健康检查配置

被如下接口引用：CreateApplication、DescribeApplication、DescribeApplications、ModifyApplication

名称	必选	允许NULL	类型	描述
Path	否	是	String	健康检查路径

MsV2

Ms

被如下接口引用：

名称	必选	允许NULL	类型	描述
MsName	是	是	String	MsName
MsPort	是	是	String	MsPort
Controller	是	是	Array of Controller	Controller
Provider	是	是	Array of Provider	Provider
Consumer	是	是	Array of Consumer	Consumer

TsfPageMicroserviceWrong

微服务列表信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of MicroserviceWrong	微服务列表信息

TsfPageMeshLog

mesh 日志列表

被如下接口引用：SearchMeshLog、SearchOssRealtimeBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of MeshLog	mesh日志 列表

名称	必选	允许NULL	类型	描述
ScrollId	是	是	String	游标Id
Status	是	是	String	查询状态

GatewayPluginDefinition

微服务网关插件定义数据

被如下接口引用：DescribeGatewayPluginTypes

名称	必选	允许NULL	类型	描述
Id	是	否	String	网关插件定义id
Name	是	是	String	插件名称
Description	是	是	String	插件描述
Type	是	是	String	插件类型
Tag	是	是	String	插件标签
CreatedTime	是	是	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	是	String	插件更新时间 如:2019-06-20 15:51:28
Order	是	是	Int64	插件排序

RecordActionTypeNameV2

操作记录操作类型名称

被如下接口引用：

名称	必选	允许NULL	类型	描述
ActionType	是	是	String	操作记录操作类型
ActionTypeName	是	是	String	操作记录操作类型名称

Resource

tsf-privilege 模块，资源

被如下接口引用：CreateProgram、DescribeProgram、DescribePrograms、DescribeResources、ModifyProgram

名称	必选	允许NULL	类型	描述
ResourceId	否	是	String	资源ID

名称	必选	允许NULL	类型	描述
ResourceCode	否	是	String	资源编码
ResourceName	否	是	String	资源名称
ServiceCode	否	是	String	资源所属产品编码
ResourceAction	否	是	String	选取资源使用的Action
IdField	否	是	String	资源数据查询的ID字段名
NameField	否	是	String	资源数据查询的名称字段名
SelectIdsField	否	是	String	资源数据查询的ID过滤字段名
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间
DeleteFlag	否	是	Bool	删除标识
ResourceDesc	否	是	String	资源描述
CanSelectAll	否	是	Bool	是否可以选择全部
SearchWordField	否	是	String	资源数据查询的模糊查询字段名
Index	否	是	Int64	排序

SubTransaction

子事务详情

被如下接口引用：

名称	必选	允许NULL	类型	描述
TransactionId	是	否	String	事务ID
ServiceName	是	否	String	事务服务名
MethodName	是	否	String	事务调用方法
StartTime	是	否	String	事务开始时间
EndTime	是	否	String	事务结束时间
Status	是	否	Int64	事务状态
InvokeType	是	否	Int64	事务触发类型
Args	是	否	String	事务参数

ListAlarmReceiverResult

ListAlarmReceiverResult

被如下接口引用：ListAlarmReceivers

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of AlarmReceiverResult	Content

TsfPageRecordV2

操作记录列表对象

被如下接口引用：DescribeRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of RecordV2	操作记录列表

Microservice

微服务

被如下接口引用：DescribeMicroservices、DescribeMicroservicesByGroupIds、DescribeServiceNames

名称	必选	允许NULL	类型	描述
MicroserviceId	否	是	String	微服务ID
MicroserviceName	否	是	String	微服务名称
MicroserviceDesc	否	是	String	微服务描述
CreateTime	否	是	Int64	创建时间
UpdateTime	否	是	Int64	更新时间
NamespaceId	否	是	String	命名空间ID
RunInstanceCount	否	是	Int64	微服务的运行实例数目
CriticalInstanceCount	否	是	Int64	微服务的离线实例数目

ZipkinTraceInfoV2

调用链信息

被如下接口引用：SearchOssTrace、SearchTrace

名称	必选	允许NULL	类型	描述
TraceId	是	是	String	调用链ID
Timestamp	是	是	Uint64	调用链时间戳
Duration	是	是	Uint64	调用链时耗
ResultStatus	是	是	String	调用链结果
ResultError	是	是	Bool	调用链错误标识

TsfRegion

TSF地域

被如下接口引用：DescribeTsfRegions

名称	必选	允许NULL	类型	描述
Id	是	是	String	TSF地域ID
Name	是	是	String	TSF地域名称

DeployGroup

部署组字段信息

被如下接口引用：DescribeDeployGroup

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组Id
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群Id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间Id
NamespaceName	是	是	String	命名空间名称
GroupDesc	是	是	String	部署组描述

名称	必选	允许NULL	类型	描述
CreateTime	是	是	String	部署组创建时间
UpdateTime	是	是	String	部署组更新时间
InstanceCount	是	是	Int64	部署组实例数目
RunInstanceCount	是	是	Int64	部署组运行实例数目
OffInstanceCount	是	是	Int64	部署组停止实例数目
GroupStatus	是	是	String	部署组状态
StartupParameters	是	是	String	部署组启动参数
PackageId	是	是	String	部署组程序包Id
PackageName	是	是	String	部署组程序包名称
PackageVersion	是	是	String	部署组程序包版本
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
CpuLimit	是	是	String	最大分配cpu 核数，如0.6
MemLimit	是	是	String	最大分配内存M数
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数
ClusterIp	是	是	String	Service ip
LbIp	是	是	String	负载均衡ip
Envs	是	是	Array of Env	环境变量数组对象
Message	是	是	String	pod错误信息描述
NodePort	是	是	Int64	NodePort端口，只有公网和NodePort访问方式才有值
Status	是	是	String	部署组状态
MicroserviceType	是	是	String	应用微服务类型

VmTask

虚拟机操作记录

被如下接口引用：DscribeTasks

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	任务Id
CreateTime	是	是	String	创建时间
TaskType	是	是	Int64	任务类型
GroupId	是	是	String	分组Id
GroupName	是	是	String	分组名称
PkgId	是	是	String	程序包Id
PkgName	是	是	String	程序包名称
PkgVersion	是	是	String	程序包版本
TaskDesc	是	是	String	任务描述
TotalCount	是	是	Int64	子任务数目
SuccessCount	是	是	Int64	成功的子任务数目
RunCount	是	是	Int64	运行中的子任务数目
FailCount	是	是	Int64	失败的子任务数目
TaskStatus	是	是	Uint64	任务的变更状态

TaskLastExecuteStatus

任务最近一次执行状态

被如下接口引用：DescribeTaskLastStatus

名称	必选	允许NULL	类型	描述
BatchId	是	是	String	批次ID
State	是	是	String	运行状态，RUNNING/SUCCESS/FAIL/HALF/TERMINATED
BatchLogId	是	是	String	批次历史ID

TsfRecordCodeResult

操作审计日志术语码返回结果

被如下接口引用：DescribeRecordCodes

名称	必选	允许NULL	类型	描述
Operations	是	否	Array of TypeCode	操作类型术语
Modules	是	否	Array of TypeCode	模块类型术语

ZipkinMetadataV2

调用链Span元数据

被如下接口引用：GetOssTraceSpans

名称	必选	允许NULL	类型	描述
Value	是	是	String	元数据值
Timestamp	是	是	Uint64	元数据时间戳
Endpoint	是	是	ZipkinEndpointV2	元数据端点信息

VolumeInfo

容器卷挂载信息

被如下接口引用：DeployContainerGroup

名称	必选	允许NULL	类型	描述
VolumeType	是	否	String	数据卷类型
VolumeName	是	否	String	数据卷名称
VolumeConfig	否	否	String	数据卷配置

ZipkinMetadata

调用链Span元数据

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
Value	是	是	String	元数据值
Timestamp	是	是	Uint64	元数据时间戳
Endpoint	是	是	ZipkinEndPoint	元数据端点信息

PathRewritePage

路径重写翻页对象

被如下接口引用：DescribePathRewrites

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of PathRewrite	路径重写规则列表

VmSubTask

虚拟机子任务信息

被如下接口引用：DescribeSubTasks

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器Id
LanIp	是	是	String	内网IP
WanIp	是	是	String	外网IP
InstanceName	是	是	String	机器名称
AgentState	是	是	Int64	机器Agent状态
InstanceState	是	是	Int64	机器状态
TaskStatus	是	是	Int64	任务状态
TaskRecord	是	是	String	任务记录

TaskExecuteRecordPage

翻页查询任务执行返回结果

被如下接口引用：DescribeTaskExecuteRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Uint64	总数量
Content	是	否	Array of TaskExecuteRecord	任务执行列表

ApiDefinitionDescr

API 对象类型描述

被如下接口引用：DescribeApiDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	对象名称
Properties	是	否	Array of PropertyField	对象属性列表

ApiVersionArray

API版本数组

被如下接口引用：DescribeApiVersions

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	App ID
ApplicationName	是	是	String	App 名称
PkgVersion	是	是	String	App 包版本

TsfPageBusinessLog

业务日志列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of BusinessLog	业务日志列表

GroupListV2

部署组列表信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	部署组id
GroupName	是	否	String	部署组名

VmGroupOtherV2

虚拟机部署组其他字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	分组Id
PackageId	是	是	String	程序包Id
PackageName	是	是	String	程序包名称
PackageVersion	是	是	String	程序包版本
InstanceCount	是	是	Int64	分组实例数
RunInstanceCount	是	是	Int64	分组运行中实例数
OffInstanceCount	是	是	Int64	分组中停止实例数
GroupStatus	是	是	String	分组状态

TxMainTransactionV2

主事务描述

被如下接口引用：

名称	必选	允许NULL	类型	描述
TransactionId	是	是	String	事务Id
ServerIp	是	是	String	服务器Ip
ServiceName	是	是	String	服务名称
MethodName	是	是	String	方法名
SubCount	是	是	Int64	子事务数量
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
Status	是	是	Int64	事务状态
TimeoutMs	是	是	Int64	超时时间
RetryTimes	是	是	Int64	重试次数
AutoRetry	是	是	Bool	是否开启自动重试

InvocationStatisticsRatio

服务调用监控指标数据比例

被如下接口引用：DescribeInvocationStatisticsRatio

名称	必选	允许NULL	类型	描述
InvocationCurrentStatistics	是	是	InvocationStatisticsV2	当前统计数据
InvocationTargetStatistics	是	是	InvocationStatisticsV2	目标统计数据
InvocationSumQuantityRatio	是	是	Float	请求总数对比
InvocationAvgDurationRatio	是	是	Float	请求平均耗时对比
Invocation2xxStatusQuantityRatio	是	是	Float	2xx状态码响应请求数对比
Invocation4xxStatusQuantityRatio	是	是	Float	4xx状态码响应请求数对比
Invocation5xxStatusQuantityRatio	是	是	Float	5xx状态码响应请求数对比
InvocationOtherStatusQuantityRatio	是	是	Float	其它状态码响应请求数对比
InvocationConnectFailedQuantityRatio	是	是	Float	连接失败请求数对比
InvocationTimeoutQuantityRatio	是	是	Float	超时请求数对比
InvocationUnavailableQuantityRatio	是	是	Float	服务不可用请求数对比

DescribeSingleContainerGroupsResultV2

DescribeSingleContainerGroupsResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	列表数量
Content	是	是	Array of DescribeSingleContainerGroups	列表内容

GroupDailyUseStatistics

分组日使用统计对象

被如下接口引用：DescribeGroupUseDetail

名称	必选	允许NULL	类型	描述
TopReqAmount	是	否	Array of GroupUseStatisticsEntity	总调用数
TopFailureRate	是	否	Array of GroupUseStatisticsEntity	平均错误率
TopAvgTimeCost	是	否	Array of GroupUseStatisticsEntity	平均响应耗时

ScalableTaskResult

ScalableTaskResult

被如下接口引用：ListScalableTasks

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ScalableTask	Content

ImageTagsV2

列表信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
RepoName	是	否	String	仓库名
TagName	是	否	String	版本名称
TagId	是	否	String	版本ID
ImageId	是	否	String	镜像ID
Size	是	否	String	大小
CreationTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间
Author	是	否	String	镜像制作者
Architecture	是	否	String	CPU架构
DockerVersion	是	否	String	Docker客户端版本
Os	是	否	String	操作系统
PushTime	是	否	String	push时间
SizeByte	是	否	Int64	单位为字节

DescribeResourceConfigSts

DescribeResourceConfig

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
Uin	是	是	String	uin

Filter

用于请求参数中的条件过滤字段

被如下接口引用：DescribeInstances

名称	必选	允许NULL	类型	描述
Name	是	否	String	过滤条件名
Values	是	否	Array of String	过滤条件匹配值，几个条件间是或关系

TsfPageCluster

Tsf分页集群对象

被如下接口引用：DescribeSimpleClusters

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of Cluster	集群列表

ServiceStatisticsResults

服务统计结果集

被如下接口引用：DescribeStatistics

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ServiceStatisticsResult	返回结果
TotalCount	是	是	Uint64	条数

GatewayPluginId

微服务网关插件Id

被如下接口引用：CreateGatewayJwtPlugin、CreateGatewayOAuthPlugin、CreateGatewayQQMiniProgramLoginPlugin、CreateGatewayTagPlugin、CreateGatewayWeChatMiniProgramLoginPlugin

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
PluginId	是	否	String	网关插件ID

ModulesDetailResult

ModulesDetailResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ModulesResult	Content

TsfPageRecordResourceTypeNameV2

操作记录资源类型分页列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of RecordResourceTypeNameV2	资源类型列表

ModuleParamResult

ModuleParamResult

被如下接口引用：FindDeployModuleParams

名称	必选	允许NULL	类型	描述
ParameterName	是	是	String	ParameterName
ParameterValue	是	是	String	ParameterValue
ParameterType	是	是	String	ParameterType
ParameterEnums	是	是	String	ParameterEnums
Desc	是	是	String	Desc
Demo	是	是	String	Demo
ParameterId	是	是	String	ParameterId
Visible	是	是	String	Visible

名称	必选	允许NULL	类型	描述
ParameterConfigType	是	是	String	ParameterConfigType
ParameterSubType	是	是	String	ParameterSubType

TsfPrice

Tsf 询价返回

被如下接口引用：DescribePrice

名称	必选	允许NULL	类型	描述
TotalCost	是	否	Int64	总费用，不包含优惠，单位：分
RealTotalCost	是	否	Int64	优惠后总价, 单位: 分
Label	是	否	String	计费项标签 key
TimeSpan	是	否	Int64	计费项标签值，也是询价月份信息

RepositoryError

仓库相关错误

被如下接口引用：

名称	必选	允许NULL	类型	描述
Code	是	是	String	无
Message	是	是	String	无

GatewayPluginInstance

微服务插件实例明细

被如下接口引用：

名称	必选	允许NULL	类型	描述
PluginInstanceId	是	否	String	插件实例ID
Id	是	否	String	插件ID
Name	是	否	String	插件名称
Description	是	是	String	插件描述
PluginId	是	否	String	插件ID

名称	必选	允许NULL	类型	描述
Type	是	否	String	插件类型
CreatedTime	是	否	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	否	String	插件更新时间 如:2019-06-20 15:51:28

FileConfigReleaseLog

文件配置项发布日志

被如下接口引用：DescribeFileConfigReleaseLogs

名称	必选	允许NULL	类型	描述
ConfigReleaseLogId	是	是	String	配置项发布日志ID
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
LastConfigId	是	是	String	上次配置项ID
LastConfigName	是	是	String	上次配置项名称
LastConfigVersion	是	是	String	上次配置项版本
ReleaseDesc	是	是	String	发布描述
ReleaseStatus	是	是	String	发布状态
ReleaseTime	是	是	String	发布时间
RollbackFlag	是	是	Bool	回滚标识
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称

ThreadDetailInfo

线程数据详情

被如下接口引用：DescribeThreadDetailList

名称	必选	允许NULL	类型	描述
ThreadName	是	否	String	线程名
ThreadState	是	否	String	线程状态
ThreadInfos	是	否	String	线程堆栈信息
ThreadCpuUtil	是	是	Float	CPU利用率
ThreadAllocatedBytes	是	是	Int64	堆内存大小
ThreadCpuTime	是	是	Int64	CPU 运行时间
ThreadBlockCount	是	是	Int64	线程阻塞个数

RatelimitListResult

none

被如下接口引用：DescribeRatelimit

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Uint64	总数
Content	是	是	Array of RatelimitRuleV2	列表内容

TsfPageTraceService

调用链服务名列表

被如下接口引用：GetOssTraceServices、GetTraceServices

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	符合条件的总调用链服务名数量
Content	是	是	Array of String	调用链服务名列表

QuantileEntity

分位数据模型

被如下接口引用：DescribeApiUseDetail、DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
MaxValue	是	是	String	最大值

名称	必选	允许NULL	类型	描述
MinValue	是	是	String	最小值
FifthPositionValue	是	是	String	五分位值
NinthPositionValue	是	是	String	九分位值

ContainGroupV2

部署组列表（应用下钻界面的）

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	否	是	String	groupId
GroupName	否	是	String	分组名称
CreateTime	否	是	String	创建时间
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit

TaskFlowLog

工作流修改历史流水

被如下接口引用：DescribeTaskFlow

名称	必选	允许NULL	类型	描述
FlowLogId	是	否	String	批次历史 ID

名称	必选	允许NULL	类型	描述
TaskCount	是	是	Int64	工作流中任务数量
FlowId	是	否	String	工作流历史 ID
State	是	是	String	工作流可用状态
FlowName	是	是	String	工作流名称
TriggerRule	是	是	TaskRule	触发规则
GraphId	是	是	String	工作流对应的图 ID
TimeOut	是	是	Int64	超时时间

CircuitBreakerRules

分页展示路由规则

被如下接口引用：DescribeCircuitBreakerRules

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数
Content	是	否	Array of CircuitBreakerRule	路由规则

TaskGroupPage

任务分组翻页查询结果

被如下接口引用：DescribeTaskGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of String	分组ID数组

TaskRule

任务规则

被如下接口引用：CreateTask、CreateTaskFlow、DescribeTaskDetail、DescribeTaskFlow、DescribeTaskFlows、DescribeTaskRecords、ModifyTask、ModifyTaskFlow

名称	必选	允许NULL	类型	描述
RuleType	是	否	String	触发规则类型, Cron/Repeat

名称	必选	允许NULL	类型	描述
Expression	否	是	String	Cron类型规则，cron表达式。
RepeatInterval	否	是	Uint64	时间间隔， 单位毫秒

RatelimitRuleV2

限流规则

被如下接口引用：CreateRatelimit、DescribeRatelimit

名称	必选	允许NULL	类型	描述
Id	否	否	String	规则ID
Name	是	否	String	规则名字，在一个微服务下唯一
Status	否	否	Uint64	状态 0表示启用 1表示停用
SourceService	否	否	String	限流规则区分的来源微服务名
DurationSecond	是	否	Uint64	限流周期，单位秒
DurationQuota	是	否	Uint64	每周期内的限流配额
ModifyTime	否	否	Uint64	最近一次修改规则时间，UTC秒数
Description	否	否	String	描述
Dimensions	否	是	Array of RatelimitDimension	限制条件列表

RouteReleaseHistoryV2

路由规则启停记录

被如下接口引用：

名称	必选	允许NULL	类型	描述
RouteReleaseLogId	是	是	String	路由规则发布记录ID
MicroserviceId	是	是	String	微服务ID
RouteRuleId	是	是	String	路由规则ID
EnableTime	是	是	String	路由规则部署开始时间
DisableTime	是	是	String	路由规则停止时间
ReleaseDesc	是	是	String	路由规则发布
RouteRule	是	是	RouteV2	路由规则详情

名称	必选	允许NULL	类型	描述
AppId	是	是	String	账号APPID
Uin	是	是	String	账号Owner用户唯一ID
SubAccountUin	是	是	String	账号用户唯一ID

DtsGroup

事务分组

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	事务分组ID
GroupName	是	是	String	事务分组名称
GroupDesc	是	是	String	事务分组描述
CreationTime	是	是	Int64	创建时间戳，UTC，精确到毫秒
LastUpdateTime	是	是	Int64	更新时间戳，UTC，精确到毫秒
DeleteFlag	是	是	Bool	删除标识，true：可以删除；false：不可删除

GatewayPlugin

微服务网关插件实例对象

被如下接口引用：DescribeGatewayPlugins、DescribePluginInstances

名称	必选	允许NULL	类型	描述
Id	是	是	String	网关插件id
Name	是	是	String	插件名称
Type	是	是	String	插件类型
Description	是	是	String	插件描述
CreatedTime	是	是	String	创建时间
UpdatedTime	是	是	String	更新时间
Status	是	是	String	发布状态

TsfPageAuthRule

TSF标准分页的AuthRule列表

被如下接口引用：DescribeAuthorizations

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数量
Content	是	否	Array of AuthRule	微服务权限规则列表

ImageUserIsExistsV2

镜像仓库用户是否已经开通

被如下接口引用：

名称	必选	允许NULL	类型	描述
IsExist	是	否	Bool	子账号是否存在,true：存在；false：不存在
MainIsExist	是	否	Bool	主账号是否存在，true：存在；false：不存在

TsfPageConfigSummary

TsfPage<Config>

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of ConfigSummary	配置项列表

DescribeApplicationsOtherResultV2

应用列表其它字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceCount	是	是	Int64	总实例个数
RunInstanceCount	是	是	Int64	运行实例个数

CloudMonitorPoliciesV2

云监控对象策略对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
KeywordsId	是	是	String	keywordsId
KeyWords	是	是	String	keyWords
GroupId	是	是	String	groupId
GroupName	是	是	String	groupName
NamespaceId	是	是	String	namespaceId
NamespaceName	是	是	String	namespaceName
ApplicationId	是	是	String	applicationId
ApplicationName	是	是	String	applicationName
ClusterId	是	是	String	clusterId
ClusterName	是	是	String	clusterName

GatewayVo

网关部署组、分组、API列表数据

被如下接口引用：DescribeGatewayAllGroupApis

名称	必选	允许NULL	类型	描述
GatewayDeployGroupId	是	是	String	网关部署组ID
GatewayDeployGroupName	是	是	String	网关部署组名称
GroupNum	是	是	Uint64	API 分组个数
Groups	是	是	Array of GatewayApiGroupVo	API 分组列表

ApiDetailInfo

API 明细

被如下接口引用：ChangeApiUsableStatus、DescribeApisWithPlugin、DescribeGatewayApis、DescribeUsableApis

名称	必选	允许NULL	类型	描述
ApiId	是	是	String	API ID
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称

名称	必选	允许NULL	类型	描述
MicroserviceId	是	是	String	服务ID
MicroserviceName	是	是	String	服务名称
Path	是	是	String	API 请求路径
PathMapping	是	是	String	Api 映射路径
Method	是	是	String	请求方法
GroupId	是	是	String	所属分组ID
UsableStatus	是	是	String	是否禁用
ReleaseStatus	是	是	String	发布状态
RateLimitStatus	是	是	String	开启限流
MockStatus	是	是	String	是否开启mock
CreatedTime	是	是	String	创建时间
UpdatedTime	是	是	String	更新时间
ReleasedTime	是	是	String	发布时间
GroupName	是	是	String	所属分组名称
Timeout	是	是	Int64	API 超时，单位毫秒
Host	是	是	String	Api所在服务host
ApiType	是	是	String	API类型。 ms ： 微服务API； external :外部服务Api
Description	是	是	String	Api描述信息
GatewayDeployGroupId	是	是	String	部署组ID

GroupApiUseStatistics

API监控明细数据

被如下接口引用：DescribeApiUseDetail

名称	必选	允许NULL	类型	描述
TopStatusCode	是	是	Array of ApiUseStatisticsEntity	总调用数
TopTimeCost	是	是	Array of ApiUseStatisticsEntity	平均错误率
Quantile	是	是	QuantileEntity	分位值对象

ContainerInstanceResourceConfig

容器实例相关的参数配置

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
ImportMode	是	是	Array of String	实例导入方式，可多个，公有云为 ["R"]，独立版的取值有 "M" 脚本模式、"S" SSH 模式
MasterNumLimit	是	是	Int64	SSH 模式时，前端应该限制用户填这个数量的 master 主机信息
NodeNumLimitPerSetup	是	是	Int64	SSH 模式时，前端应该限制用户填的最高数量的 node 主机信息

TxErrorV2

事务查询异常信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
Code	是	是	String	错误码
Message	是	是	String	错误信息

PagedRole

tsf-privilege 模块 分页的Role列表

被如下接口引用：DescribeRoles

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of Role	角色列表

ProviderV2

provider

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProviderControllerName	是	是	String	ProviderControllerName

DeliveryKafkaInfo

kafka投递的topic和path的信息

被如下接口引用：CreateDeliveryConfig、UpdateDeliveryConfig

名称	必选	允许NULL	类型	描述
Topic	否	是	String	投递kafka的topic
Path	否	是	Array of String	采集日志的path

TsfPageMsRunningApplicationV2

微服务运行态应用分页列表对象

被如下接口引用：DescribeMsRunningApplications

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	运行微服务的应用个数
Content	是	是	Array of MsRunningApplicationV2	运行微服务的应用列表

Role

tsf-privilege模块 Role 角色

被如下接口引用：DescribeRole、DescribeRoles

名称	必选	允许NULL	类型	描述
RoleId	否	是	String	角色ID
RoleName	否	是	String	角色名称
RoleDesc	否	是	String	角色描述
DeleteFlag	否	是	Bool	删除标识
PermCatList	否	是	Array of PermCat	角色权限集列表
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间

TaskRecord

任务定义

被如下接口引用：DescribeTaskDetail、DescribeTaskRecords

名称	必选	允许NULL	类型	描述
TaskName	是	否	String	任务名称
TaskType	是	否	String	任务类型
ExecuteType	是	否	String	执行类型
TaskContent	否	否	String	任务内容，长度限制65535字节
GroupId	否	否	String	分组ID
TimeOut	是	否	Int64	超时时间
RetryCount	否	是	Int64	重试次数
RetryInterval	否	是	Int64	重试间隔
TaskRule	是	否	TaskRule	触发规则
TaskState	是	否	String	是否启用任务,ENABLED/DISABLED
TaskId	是	否	String	任务ID
SuccessOperator	是	是	String	判断任务成功的操作符
SuccessRatio	是	是	Int64	判断任务成功的阈值
ShardCount	是	是	Int64	分片数量
AdvanceSettings	是	是	AdvanceSettings	高级设置
ShardArguments	是	是	Array of ShardArgument	分片参数
BelongFlowIds	是	是	Array of String	所属工作流ID
TaskLogId	是	是	String	任务历史ID
TriggerType	是	是	String	触发类型
TaskArgument	是	是	String	任务参数，长度限制10000个字符

OverviewMsResult

概览页微服务信息

被如下接口引用：DescribeOverviweService

名称	必选	允许NULL	类型	描述
RunMicroserviceCount	是	是	Int64	概览页微服务数目

MsInstanceWrong

微服务实例信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器实例ID信息
InstanceName	是	是	String	机器名称信息
InstancePort	是	是	Int64	机器运行的端口号
LanIp	是	是	String	机器实例内网IP
WanIp	是	是	String	机器实例外网IP
InstanceAvailableStatus	是	是	String	机器实例可用状态
ServiceInstanceStatus	是	是	String	服务运行状态
PkgVersion	是	是	String	程序包版本
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
InstanceStatus	是	是	String	机器TSF可用状态
PodName	是	是	String	容器pod名
PodId	是	是	String	容器pod ID
Status	是	是	String	容器状态
NodeIp	是	是	String	容器node IP
Ip	是	是	String	容器IP
Reponame	是	是	String	容器仓库名
TagName	是	是	String	容器tag

ZipkinEndpointV2

调用链Span端点信息

被如下接口引用：DescribeSpanDetail、GetOssTraceSpans、SearchSpan

名称	必选	允许NULL	类型	描述
ServiceName	是	是	String	端点服务名
Ipv4	是	是	String	端点IP地址（v4）
Ipv6	是	是	String	端点IP地址（v6）
Port	是	是	Int64	端点端口号

LaneGroups

泳道部署组分页查询结果集

被如下接口引用：DescribeLaneGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总个数
Content	是	是	Array of LaneGroup	泳道部署组列表

TraceSpanInfo

span查询的出参

被如下接口引用：SearchSpan

名称	必选	允许NULL	类型	描述
TraceId	否	否	String	调用链traceid
SpanId	否	否	String	调用链spanid
ParentId	否	否	String	调用链parentid
Kind	否	否	String	调用链类型
Name	否	否	String	调用链name
Timestamp	否	否	Uint64	时间戳
Duration	否	否	Float	耗时
ResultStatus	否	否	String	状态
LocalComponent	否	否	String	组件
LocalEndpoint	否	否	ZipkinEndpointV2	local端点信息
RemoteEndpoint	否	否	ZipkinEndpointV2	remote端点信息

名称	必选	允许NULL	类型	描述
TagList	否	否	Array of SpanLabel	tags列表
GroupId	否	否	String	部署组id
GroupName	否	否	String	部署组名称
ClusterType	否	否	String	集群类型
NamespaceId	否	否	String	命名空间id
NamespaceName	否	否	String	命名空间名称
ResultCode	否	否	String	状态码
ServiceName	否	否	String	服务名
ServiceId	否	否	String	服务id

ContainerGroupDetailV2

容器部署组详情

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	groupId
GroupName	是	是	String	分组名称
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数
CreateTime	是	是	String	创建时间
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用id
LbIp	是	是	String	负载均衡ip

名称	必选	允许NULL	类型	描述
ApplicationType	是	是	String	应用类型
ClusterIp	是	是	String	Service ip
NodePort	是	是	Int64	NodePort端口，只有公网和NodePort访问方式才有值
CpuLimit	是	是	String	最大分配的 CPU 核数，对应 K8S limit
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
AccessType	是	是	Uint64	0:公网 1:集群内访问 2：NodePort
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
ProtocolPorts	是	是	Array of ProtocolPort	端口数组对象
Envs	是	是	Array of Env	环境变量数组对象
ApplicationName	是	是	String	应用名称
Message	是	是	String	pod错误信息描述
Status	是	是	String	部署组状态
MicroserviceType	是	是	String	服务类型
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id

DeleteImageTagV2

镜像版本

被如下接口引用：

名称	必选	允许NULL	类型	描述
Reponame	是	否	String	仓库名，如/tsf/nginx
TagName	是	否	Array of String	版本号:如V1

DescribeSingleContainerGroups

部署组-独立菜单

被如下接口引用：DescribeSingleContainerGroups

名称	必选	允许NULL	类型	描述
GroupId	否	是	String	部署组id
GroupName	否	是	String	分组名称
CpuLimit	否	是	String	最大分配的 CPU 核数，对应 K8S limit
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用名称
ApplicationName	是	是	String	应用类型
ApplicationType	是	是	String	分组创建时间
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
MicroserviceType	是	是	String	微服务类型
ProtocolPorts	是	是	Array of ProtocolPort	端口列表
UpdateType	是	是	Int64	更新类型
UpdateIvl	是	是	Int64	更新时间间隔
AccessType	是	是	Int64	网络访问类型
CreateTime	是	是	String	创建时间
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id
InstanceCount	是	是	Int64	实例个数
GroupResourceType	是	是	String	部署组资源类型
Envs	是	是	String	环境变量
UpdatedTime	是	是	Int64	部署组更新时间戳

Application

应用

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	否	是	String	应用 ID，如 application-qv3dkda7
ApplicationName	否	是	String	应用名
ApplicationDesc	否	是	String	应用描述
ApplicationType	否	是	String	应用 *部署* 类型，V 表示 CVM 应用，C 表示容器应用
ApplicationJvmArg	否	是	String	应用启动时的 JVM 参数
MicroserviceType	否	是	String	应用类型，M 表示 Mesh 应用，N 表示 Spring Cloud 应用
ProgLang	否	是	String	应用编程语言，J 表示 Java，P 表示 Python
CreateTime	否	是	String	应用创建时间
UpdateTime	否	是	String	应用修改时间
InstanceCount	否	是	Int64	应用实例数
RunInstanceCount	否	是	Int64	应用当前运行实例数

GatewayJwtPlugin

微服务网关JWT插件

被如下接口引用：DescribeGatewayJwtPlugin

名称	必选	允许NULL	类型	描述
Id	是	是	String	网关插件id
Name	是	是	String	插件名称
Description	是	是	String	插件描述
Type	是	是	String	插件类型
CreatedTime	是	是	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	是	String	插件更新时间 如:2019-06-20 15:51:28
Kid	是	是	String	公钥标识
PublicKeyJson	是	是	String	公钥对
RedirectUrl	是	是	String	重定向地址
TokenBaggagePosition	是	是	String	token携带位置，网关取token位置与发送认证请求时token位置一致,值:query/header

名称	必选	允许NULL	类型	描述
TokenKeyName	是	是	String	token的key值
ClaimMappingJson	是	是	String	claim参数映射关系json
Status	是	是	String	发布状态: drafted/released

VmGroupSimpleV2

虚拟机部署组列表简要字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组Id
GroupName	是	是	String	部署组名称
ApplicationType	是	是	String	应用类型
GroupDesc	是	是	String	部署组描述
UpdateTime	是	是	String	部署组更新时间
ClusterId	是	是	String	集群Id
StartupParameters	是	是	String	部署组启动参数
NamespaceId	是	是	String	命名空间Id
CreateTime	是	是	String	部署组创建时间
ClusterName	是	是	String	集群名称
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
NamespaceName	是	是	String	命名空间名称
MicroserviceType	是	是	String	应用微服务类型

TaskInstanceV2

TaskInstance

被如下接口引用：

名称	必选	允许NULL	类型	描述
Id	是	是	String	id

名称	必选	允许NULL	类型	描述
Status	是	是	Int64	status
Mtime	是	是	String	mtime

SidecarFilterGroup

过滤器关联的部署组

被如下接口引用：DescribeSidecarFilter、DescribeSidecarFilters、ReleaseSidecarFilter

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	部署组Id
FilterId	是	否	String	过滤器Id
GroupName	是	是	String	部署组名称
NamespaceId	是	否	String	部署组所在命名空间Id
NamespaceName	是	是	String	部署组所在名称空间名称

DescribeTemplateResultV2

DescribeTemplateResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	ProjectId
ProjectName	是	否	String	ProjectName
BasePackage	是	否	String	BasePackage
PomGroupId	是	否	String	PomGroupId
PomArtifactId	是	否	String	PomArtifactId
PomVersion	是	否	String	PomVersion
PomName	是	否	String	PomName
PomDesc	是	否	String	PomDesc
GetMethod	是	否	String	GetMethod
Ms	是	否	Array of Ms	Ms
Action	否	否	String	action

名称	必选	允许NULL	类型	描述
AppId	否	否	String	appId
Uin	否	否	String	Uin
SubAccountUin	否	否	String	SubAccountUin

MonitorOverview

监控概览对象

被如下接口引用：DescribeGatewayMonitorOverview

名称	必选	允许NULL	类型	描述
InvocationCountOfDay	是	是	String	近24小时调用数量
InvocationCount	是	是	String	总调用数量
ErrorCountOfDay	是	是	String	近24小时调用错误数量
ErrorCount	是	是	String	总调用错误数量
SuccessRatioOfDay	是	是	String	近24小时调用成功率
SuccessRatio	是	是	String	总调用成功率

TaskBatchRecord

任务批次

被如下接口引用：DescribeTaskBatchHistoryRecords、DescribeTaskBatchRecord、DescribeTaskBatchRecords

名称	必选	允许NULL	类型	描述
BatchId	是	否	String	任务批次ID
TaskId	是	否	String	任务ID
State	是	否	String	任务状态
StartTime	是	是	Uint64	开始时间
EndTime	是	是	Uint64	结束时间
TotalCount	是	否	Uint64	总共数据
SuccessCount	是	否	Uint64	成功的任务数量
FailedCount	是	否	Uint64	失败的任务数量
TimeoutCount	是	否	Uint64	超时的任务数量

名称	必选	允许NULL	类型	描述
TerminateCount	是	否	Uint64	停止的任务数量
GroupId	是	是	String	任务分组ID
TaskName	是	是	String	任务名称
TaskLogId	是	否	String	任务历史 ID
BatchLogId	是	是	String	批次历史 ID
ScheduleFireTime	是	是	Int64	任务计划触发的时间点
FlowBatchId	是	是	String	工作流批次 ID
BatchType	是	是	String	批次类型
TriggerType	是	是	String	任务触发类型
FlowBatchLogId	是	是	String	工作流批次历史 ID
SuccessRatio	是	是	Int64	执行成功率
HistoryCount	是	是	Int64	执行批次历史数量
SpanTime	是	是	Int64	耗时
FlowId	是	是	String	所属工作流ID
FlowName	是	是	String	所属工作流名称
FlowLogId	是	是	String	所属工作流历史ID

ApplicationApiAccess

Serverless应用外网权限信息

被如下接口引用：DescribeApiAccess

名称	必选	允许NULL	类型	描述
Status	是	是	String	发布状态: SUCCEEDED/FAILED
SubDomain	是	是	String	系统给该服务自动分配的域名
ServiceId	是	是	String	ApiGateway 服务ID
Protocol	是	是	String	协议类型

ContainerGroup

部署组

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	否	否	String	部署组ID
GroupName	否	否	String	分组名称
CreateTime	否	否	String	创建时间
Server	是	否	String	镜像server
Reponame	是	否	String	镜像名，如/tsf/nginx
TagName	是	否	String	镜像版本名称
ClusterId	是	否	String	集群ID
ClusterName	是	否	String	集群名称
NamespaceId	是	否	String	命名空间ID
NamespaceName	是	否	String	命名空间名称
ApplicationId	是	否	String	应用名称
ApplicationName	是	否	String	应用类型
ApplicationType	是	否	String	分组创建时间
UpdateTime	是	否	String	分组更新时间
MicroserviceType	是	是	String	微服务类型

MonitorStatisticsPolicyGroupV2

MonitorStatisticsPolicyGroup

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	GroupId
GroupName	是	是	String	GroupName
GroupDesc	是	是	String	GroupDesc
NamespaceId	是	是	String	NamespaceId
NamespaceName	是	是	String	NamespaceName
ApplicationId	是	是	String	ApplicationId
ApplicationName	是	是	String	ApplicationName

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	ClusterId
ClusterName	是	是	String	ClusterName

InstanceInfo

实例信息

被如下接口引用：DescribeInterfaceRequest

名称	必选	允许NULL	类型	描述
InstanceId	否	是	String	实例id
GroupId	否	是	String	部署id
ClusterType	否	是	String	集群类型
InstanceIp	否	是	String	实例ip

ContainerTasks

变更记录任务

被如下接口引用：DescribeContainerTasks

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	taskId
TaskTime	是	是	String	任务开始时间
Type	是	是	Int64	任务类型字段，0：没有任务（在此接口中，不用出现0），1：发布程序包；2.部署操作；3.扩容操作；4.启动操作；5.停止操作；6.缩容操作；7.发布日志配置,8.删除销毁操作
Status	是	是	Int64	变更状态，0：成功 1:失败 2：执行中
GroupId	是	是	String	分组id
GroupName	是	是	String	分组名称
ImageName	是	是	String	镜像名称
ImageVersion	是	是	String	镜像版本
TaskDesc	是	是	String	任务详情描述
TotalCount	是	是	Int64	任务总个数
SuccessCount	是	是	Int64	成功任务个数

名称	必选	允许NULL	类型	描述
FailCount	是	是	Int64	失败任务个数

DescribeContainerTasksResultV2

DescribeContainerTasksResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	列表总数
Content	是	是	Array of ContainerTasks	列表内容

Ms

Ms

被如下接口引用：CreateAndDownloadTemplate、CreateTemplate

名称	必选	允许NULL	类型	描述
MsName	是	是	String	MsName
MsPort	是	是	String	MsPort
Controller	是	是	Array of Controller	Controller
Provider	是	是	Array of Provider	Provider
Consumer	是	是	Array of Consumer	Consumer

TaskFlowBatchLastState

工作流最近批次状态

被如下接口引用：

名称	必选	允许NULL	类型	描述
FlowBatchId	是	是	String	批次ID
FlowBatchLogId	是	是	String	批次历史ID
State	是	是	String	状态

InstanceRequest

单条实例请求详情

被如下接口引用：DescribeInstanceRequest

名称	必选	允许NULL	类型	描述
TraceId	是	是	String	trace id
Timestamp	是	是	Int64	产生时间
Duration	是	是	Int64	耗时
ResultStatus	是	是	String	状态
ResultError	是	是	String	错误
InterfaceInfo	是	是	Interface	接口信息

DailyUseStatisticsEntity

日统计数据节点

被如下接口引用：DescribeGatewayDailyUseStatistics

名称	必选	允许NULL	类型	描述
Value	是	是	String	日统计值
DailyRate	是	是	String	日环比值
WeekRate	是	是	String	周同比值

TsfPageFileConfigReleaseLog

文件配置项发布日志列表

被如下接口引用：DescribeFileConfigReleaseLogs

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	数量
Content	是	是	Array of FileConfigReleaseLog	列表

TsfPageNamespaceWrong

Tsf命名空间分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of NamespaceWrong	命名空间列表

VmGroupOther

虚拟机部署组其他字段

被如下接口引用：DescribeGroupAttribute、DescribeGroupOther

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
PackageId	是	是	String	程序包ID
PackageName	是	是	String	程序包名称
PackageVersion	是	是	String	程序包版本
InstanceCount	是	是	Int64	部署组实例数
RunInstanceCount	是	是	Int64	部署组运行中实例数
OffInstanceCount	是	是	Int64	部署组中停止实例数
GroupStatus	是	是	String	部署组状态
IsNotEqualServiceConfig	是	是	Bool	服务配置信息是否匹配

DescribeContainerTasksResult

DescribeContainerTasksResult

被如下接口引用：DescribeContainerTasks

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	列表总数
Content	是	是	Array of ContainerTasks	列表内容

ApplicationV2

应用

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ApplicationId	否	是	String	应用 ID，如 application-qv3dkda7
ApplicationName	否	是	String	应用名
ApplicationDesc	否	是	String	应用描述
ApplicationType	否	是	String	应用 *部署* 类型，V 表示 CVM 应用，C 表示容器应用
ApplicationJvmArg	否	是	String	应用启动时的 JVM 参数
MicroserviceType	否	是	String	应用类型，M 表示 Mesh 应用，N 表示 Spring Cloud 应用
ProgLang	否	是	String	应用编程语言，J 表示 Java，P 表示 Python
CreateTime	否	是	String	应用创建时间
UpdateTime	否	是	String	应用修改时间
InstanceCount	否	是	Int64	应用实例数
RunInstanceCount	否	是	Int64	应用当前运行实例数

RatelimitDimension

限流规则作用的标签维度

被如下接口引用：CreateRatelimit、DescribeRatelimit、UpdateRatelimit

名称	必选	允许NULL	类型	描述
TagType	是	否	String	`S`：系统标签/ `U`：自定义标签
TagField	是	否	String	标签名
TagOperator	是	否	String	EQUAL/NOT_EQUAL/IN/NOT_IN/REGEX
TagValue	是	否	String	标签匹配值

DescribeResourceConfigClusterContainer

返回给前端的控制信息

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
NeedSubnetWhenCreatingCluster	是	是	Bool	是否需要子网

TsfPageSimpleGroupV2

TSF简单部署组分页列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleGroup	简单部署组列表

AuthCondition

用于 TSF 服务鉴权，表示鉴权条件

被如下接口引用：DescribeAuthorizationInfo、ModifyAuthorizationInfo

名称	必选	允许NULL	类型	描述
Type	是	否	String	类型，meta 或者 tag；条件为服务名时用 meta，其他用 tag
Key	是	否	String	条件为服务名时，key 为 sourceServiceId；条件是 tag 时，key 为用户自定义
Operator	是	否	String	操作符，从中选其一，equal, notEqual, in, notIn, regex
ValueList	是	是	Array of String	如果操作符不是 in / notIn， 值只有一个
SourceServiceList	否	是	Array of AuthMicroservice	如果条件是服务名，在回包中这个数组中包含主调服务信息

MonitorStatisticsPolicy

MonitorStatisticsPolicy

被如下接口引用：DescribeMonitorStatisticsPolicy、ListMonitorStatisticsPolicy

名称	必选	允许NULL	类型	描述
PolicyId	是	是	String	监控统计策略id
KeyWords	是	是	String	关键词
GroupList	是	是	Array of MonitorStatisticsPolicyGroup	部署组列表信息
NamespaceId	是	是	String	命名空间id
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

DeliveryConfigBindGroups

描述配置项绑定的部署组

被如下接口引用：DescribeDeliveryConfigs

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	公共条数
Content	是	是	Array of DeliveryConfigBindGroup	内容

ClusterLimitResourceV2

集群limit资源

被如下接口引用：DescribeClusterLimitResource

名称	必选	允许NULL	类型	描述
CpuLimit	是	是	String	最大分配 CPU 核数
MemLimit	是	是	String	最大分配内存 MiB 数
CpuRequest	是	是	String	初始分配 CPU 核数
MemRequest	是	是	String	初始分配内存 MiB 数

DeleteImageTag

需要删除的镜像版本

被如下接口引用：DeleteImageTags

名称	必选	允许NULL	类型	描述
RepoName	是	否	String	仓库名，如/tsf/nginx
TagName	是	否	String	版本号:如V1

ProductHelp

产品帮助

被如下接口引用：DescribeProductHelp、ReleaseProductHelp

名称	必选	允许NULL	类型	描述
HelpId	否	是	String	id
Content	否	是	String	内容
Link	否	是	String	链接

名称	必选	允许NULL	类型	描述
Order	否	是	Int64	排序
StyleFlag	否	是	String	样式Flag
Valid	否	是	Int64	是否有效
ValidTime	否	是	String	生效时间（当valid从0 -> 1，刷新生效时间）
Title	否	是	String	标题
ServiceTag	否	是	String	产品标识(默认为tsf)

GetContainGroupDeployInfoV2

获取容器部署信息(GetContainGroupDeployInfo) 【部署更新的时候需要获取部署组信息】 输出参数

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	GroupId
GroupName	是	否	String	分组名称
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动的实例数
Server	是	否	String	镜像server
Reponame	是	否	String	镜像名，如/tsf/nginx
TagName	是	否	String	镜像版本名称
CpuRequest	是	否	String	预分配cpu 核数，如0.2，
CpuLimit	是	否	String	最大分配cpu 核数，如0.6
MemRequest	是	否	String	预分配内存M数
MemLimit	是	否	String	最大分配内存M数
AccessType	是	否	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	是	否	Array of ProtocolPort	数组
UpdateType	是	否	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	否	Int64	更新间隔,单位秒
JvmOpts	是	否	String	jvm参数
ClusterId	是	否	String	无用

名称	必选	允许NULL	类型	描述
ClusterName	是	否	String	无用
ApplicationId	是	否	String	无用
ApplicationName	是	否	String	无用
ApplicationType	是	否	String	无用
KuberneteApiServer	是	否	String	无用
KuberneteUser	是	否	String	无用
KubernetePassword	是	否	String	无用
NamespaceId	是	否	String	无用
NamespaceName	是	否	String	无用
GroupComment	是	否	String	无用
PodId	是	否	String	无用
PodName	是	否	String	无用
ClusterIp	是	否	String	无用
LbIp	是	否	String	无用
NodePort	是	否	String	无用
IsStop	是	否	String	无用
Status	是	否	String	无用
Message	是	否	String	无用
ChangType	是	否	Int64	无用
ChangNum	是	否	Int64	无用
IsFirst	是	否	Int64	无用
CreateTime	是	否	String	无用
UpdateTime	是	否	String	无用
AppId	是	否	String	无用
SubAccountUin	是	否	String	无用
Uin	是	否	String	无用
Limit	是	否	Int64	无用
OrderType	是	否	String	无用

ZipkinAnnotationV2

调用链Span注解

被如下接口引用：GetOssTraceSpans

名称	必选	允许NULL	类型	描述
Value	是	是	String	注解值
Timestamp	是	是	Uint64	注解时间戳
Endpoint	是	是	ZipkinEndpointV2	注解端点信息

TsfPageBillingDeal

BillingDeal 翻页对象

被如下接口引用：DescribeBillingDealRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录总数
Content	是	是	Array of BillingDeal	记录实体列表

StdoutLog

标准输出日志

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID
Content	是	是	String	日志内容
Timestamp	是	是	Uint64	日志时间戳

TsfPageConfigReleaseLog

TSF配置项发布日志分页对象

被如下接口引用：DescribeConfigReleaseLogs、DescribePublicConfigReleaseLogs

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ConfigReleaseLog	配置项发布日志数组

DtsGroupStatistics

事务分组统计信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	事务分组ID
TxQty	是	是	Int64	主事务总数
BranchQty	是	是	Int64	分支事务总数
ErrorTxQty	是	是	Int64	异常主事务数

SimpleGroup

部署组

被如下接口引用：DescribeSimpleGroups

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
StartupParameters	是	是	String	启动参数
GroupResourceType	是	是	String	部署组资源类型
AppMicroServiceType	是	是	String	应用微服务类型

TsfPageApplicatoinServerLog

应用服务器的日志信息

被如下接口引用：DescribeApplicatoinServerLog

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of ApplicationServerLogContent	日志内容

RepositoryList

仓库列表

被如下接口引用：DescribeRepositories

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	仓库总量
Content	是	是	Array of RepositoryInfo	仓库信息列表

ConfigTemplateResultV2

配置模板列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of ConfigTemplate	配置模板列表

JvmMonitorData

DescribeJvmMonitor查询jvm监控数据接口返回数据封装

被如下接口引用：DescribeJvmMonitor

名称	必选	允许NULL	类型	描述
HeapMemory	是	是	MemoryPicture	堆内存监控图,三条线
NonHeapMemory	是	是	MemoryPicture	非堆内存监控图,三条线
EdenSpace	是	是	MemoryPicture	伊甸园区监控图,三条线

名称	必选	允许NULL	类型	描述
SurvivorSpace	是	是	MemoryPicture	幸存者区监控图,三条线
OldSpace	是	是	MemoryPicture	老年代监控图,三条线
MetaSpace	是	是	MemoryPicture	元空间监控图,三条线
ThreadPicture	是	是	ThreadPicture	线程监控图,三条线
YoungGC	是	是	Array of CurvePoint	youngGC增量监控图,一条线
FullGC	是	是	Array of CurvePoint	fullGC增量监控图,一条线
CpuUsage	是	是	Array of CurvePoint	cpu使用率,一条线
ClassCount	是	是	Array of CurvePoint	加载类数,一条线

CurvePoint

构成监控数据图的曲线坐标点

被如下接口引用：DescribeJvmMonitor

名称	必选	允许NULL	类型	描述
Label	是	否	String	当前坐标 X轴的值 当前是日期格式:"yyyy-MM-dd HH:mm:ss"
Value	是	否	String	当前坐标 Y轴的值
Timestamp	是	否	String	该坐标点时间戳

TsfJvmLogPage

Jvm日志查询返回数据封装

被如下接口引用：DescribeJvmLogs

名称	必选	允许NULL	类型	描述
Content	是	是	Array of JvmLog	jvm日志详情列表
TotalCount	是	是	Int64	总数目
ScrollId	是	是	String	游标ID, 根据游标Id查询后续结果

PagedDtsBranch

DTS分支事务分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	是	Array of DtsBranch	分支事务分组列表

TsfPageSimpleApplication

TSF分页简单应用对象

被如下接口引用：DescribeSimpleApplications

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleApplication	简单应用列表

DescribeResourceConfigResult

DescribeResourceConfig

被如下接口引用：

名称	必选	允许NULL	类型	描述
Sts	是	是	DescribeResourceConfigSts	Sts
License	是	是	DescribeResourceConfigLicense	License

AlarmPolicyResult

AlarmPolicyResult

被如下接口引用：DescribeAlarmPolicy、ListAlarmPolicies

名称	必选	允许NULL	类型	描述
PolicyId	否	是	String	PolicyId
PolicyName	否	是	String	PolicyName
EventPolicies	否	是	Array of EventPolicyResult	EventPolicies
EnabledEmail	否	是	Int64	EnabledEmail
EnabledSMS	否	是	Int64	EnabledSMS
Enabled	否	是	Int64	Enabled
Receivers	否	是	Array of AlarmReceiverResult	Receivers

名称	必选	允许NULL	类型	描述
AlarmSubId	否	是	Int64	AlarmSubId
UpdateTime	否	是	String	UpdateTime
CreateTime	否	是	String	CreateTime
EnabledWeChat	否	是	String	EnabledWeChat
EnabledRtx	否	是	String	EnabledRtx

GatewayGroupIds

网关部署组ID和网关API分组ID元组

被如下接口引用：BindApiGroup、UnbindApiGroup

名称	必选	允许NULL	类型	描述
GatewayDeployGroupId	是	否	String	网关部署组ID
GroupId	是	否	String	分组id

ImageRepositoryResultV2

镜像仓库列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总记录数
Server	是	是	String	镜像服务器地址
Content	是	是	Array of ImageRepository	列表信息

MetricDataPoint

监控统计数据点

被如下接口引用：DescribeGWOverviewInvocation、DescribeInvocationMetricDataCurve、DescribeOverviewInvocation、DescribeStatistics

名称	必选	允许NULL	类型	描述
Key	是	是	String	数据点键
Value	是	是	String	数据点值

名称	必选	允许NULL	类型	描述
Tag	是	是	String	数据点标签

PkgBind

描述程序包关联信息

被如下接口引用：DescribePkgs、ListPkg

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用id
GroupId	是	是	String	部署组id

TsfPageVmGroupV2

列表中部署组分页信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of VmGroupSimple	部署组列表信息

Ports

服务端口

被如下接口引用：CreateApplication、DescribeApplication、DescribeApplications、ModifyApplication

名称	必选	允许NULL	类型	描述
TargetPort	是	否	Uint64	服务端口
Protocol	是	否	String	端口协议

TsfPageZipkinSpanInfoV2

调用链Span列表

被如下接口引用：GetOssTraceSpans

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条目数
Content	是	是	Array of ZipkinSpanInfoV2	调用链Span列表

BusinessLog

业务日志

被如下接口引用：SearchContainerStdoutLog

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID
Content	是	是	String	日志内容
Timestamp	是	是	Uint64	日志时间戳

ZipkinTraceInfo

调用链信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
TraceId	是	是	String	调用链ID
Timestamp	是	是	Uint64	调用链时间戳
Duration	是	是	Uint64	调用链时耗
ResultStatus	是	是	String	调用链结果

TsfPageRecordActionTypeNameV2

操作记录操作类型名称列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总数目
Content	是	是	Array of RecordActionTypeNameV2	操作类型名称列表

OverviewInstanceResourceUsage

实例相关信息概览

被如下接口引用：DescribeInstanceResourceUsage

名称	必选	允许NULL	类型	描述
ClusterCount	是	是	Int64	集群数目
InstanceCount	是	是	Int64	实例总数目
UnNormalClusterCount	是	是	Int64	异常集群数目
UnNormalInstanceCount	是	是	Int64	异常实例数目
RunInstanceCount	是	是	Int64	运行中实例数目
OffInstanceCount	是	是	Int64	停止实例数目

JavaInstance

DescribeJavaInstance接口返回的一系列指标封装

被如下接口引用：DescribeJavaInstance

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例Id
ElapsedTime	是	是	Int64	服务运行时间, 单位秒
CupConsumingTime	是	是	Int64	cpu消耗总时长, 单位秒
JitCompileTime	是	是	Int64	服务JIT编译时间, 单位秒
HeapUsed	是	是	Int64	已用堆大小,单位KB
HeapFree	是	是	Int64	可用堆大小,单位KB
HeapSize	是	是	Int64	堆大小总量,单位KB
HeapCommitted	是	是	Int64	commit堆内存, 单位KB
WaitBuildObjectCount	是	是	Int64	等待构析的对象数量
YoungGcCount	是	是	Int64	产生Young GC总次数
FullGcCount	是	是	Int64	产生Full GC总次数
YoungGcConsumingTime	是	是	Int64	Young GC总耗时长, 单位秒
FullGcCountConsumingTime	是	是	Int64	Full GC总耗时长, 单位秒
ActiveThreadCount	是	是	Int64	活跃线程数
CpuPercent	是	是	Int64	cpu使用率
StatusInfo	是	是	String	异步执行错误时,各种错误信息

名称	必选	允许NULL	类型	描述
StatusCode	是	是	Int64	异步执行错误时,对应的错误码

ControllerV2

tsf模板

被如下接口引用：

名称	必选	允许NULL	类型	描述
ControllerName	是	是	String	ControllerName

InvocationStatistics

调用统计数据

被如下接口引用：

名称	必选	允许NULL	类型	描述
ReqSuccessStats	是	是	Array of StatisticsCoord	请求健康概览统计成功请求数
ReqFailedStats	是	是	Array of StatisticsCoord	请求健康概览统计失败请求数
ReqStatusStats	是	是	Array of StatisticsCoord	请求健康概览统计状态码分布
ReqDurationStats	是	是	Array of StatisticsCoord	请求时延概览统计时延分布
ReqQuantityStats	是	是	Array of StatisticsCoord	请求并发概览统计并发请求次数

TsfPageLicenseTag

LicenseTag 翻页对象

被如下接口引用：DescribeLicenses

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录总数
Content	是	是	Array of LicenseTag	记录实体列表

ScalableRuleV2

ScalableRule

被如下接口引用：

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	RuleId
Name	是	是	String	Name
ExpandVmCountLimit	是	是	Int64	ExpandVmCountLimit
ShrinkVmCountLimit	是	是	Int64	ShrinkVmCountLimit
GroupCount	是	是	Int64	GroupCount

GroupPodResultV2

部署组实例列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总记录数
Content	是	是	Array of GroupPod	列表信息

Config

配置项

被如下接口引用：DescribeConfig、DescribeConfigSummary、DescribeConfigs、DescribePublicConfig、DescribePublicConfigSummary、DescribePublicConfigs

名称	必选	允许NULL	类型	描述
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
ConfigVersionDesc	是	是	String	配置项版本描述
ConfigValue	是	是	String	配置项值
ConfigType	是	是	String	配置项类型
CreationTime	是	是	String	创建时间
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
DeleteFlag	是	是	Bool	删除标识，true：可以删除；false：不可删除

名称	必选	允许NULL	类型	描述
LastUpdateTime	是	是	String	最后更新时间
ConfigVersionCount	是	是	Int64	配置项版本数量

GroupResourceConfig

部署组相关的参数配置

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Container	是	是	ContainerGroupResourceConfig	容器部署组相关的参数配置

OverviewLicenseResourceUsage

license相关信息概览

被如下接口引用：DescribeLicenseInfo

名称	必选	允许NULL	类型	描述
ConsulInstanceCount	是	是	Int64	已注册实例数目

TsfPageContainerEvent

分页的 ContainerEvent

被如下接口引用：DescribeContainerEvents

名称	必选	允许NULL	类型	描述
TotalCount	否	否	Int64	返回个数
Content	否	否	Array of ContainerEvent	events 数组

TemplateResult

TemplateResult

被如下接口引用：DescribeTemplates

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TotalCount

名称	必选	允许NULL	类型	描述
Content	是	是	Array of TemplateProject	Content

TaskFlowBatchPage

工作流翻页数据

被如下接口引用：DescribeFlowBatchHistoryRecords、DescribeFlowBatchRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	数据总数
Content	是	否	Array of TaskFlowBatch	翻页数据

TaskFlowBatch

工作流对象

被如下接口引用：DescribeFlowBatchHistoryRecords、DescribeFlowBatchRecord、DescribeFlowBatchRecords

名称	必选	允许NULL	类型	描述
FlowId	是	否	String	工作流 ID
FlowLogId	是	是	String	工作流历史 ID
FlowBatchLogId	是	否	String	工作流批次历史 ID
FlowBatchId	是	否	String	工作流批次 ID
State	是	是	String	状态
TriggerType	是	是	String	触发类型
ScheduleFireTime	是	是	Int64	预计触发时间
StartTime	是	是	Int64	开始执行时间
EndTime	是	是	Int64	结束执行时间
HistoryCount	是	是	Int64	批次历史数量
SpanTime	是	是	Int64	执行耗时
FlowName	是	是	String	工作流名称
TimeOut	是	是	Int64	超时时间

ServiceStatistics

服务运行统计指标

被如下接口引用：GetServiceStatistics

名称	必选	允许NULL	类型	描述
ServiceId	是	是	String	服务ID
RequestAmount	是	是	String	总请求量
SuccessAmount	是	是	String	请求成功量
SuccessRate	是	是	String	请求成功率
MaxDuration	是	是	String	请求最大耗时
MinDuration	是	是	String	请求最小耗时
AvgDuration	是	是	String	请求平均耗时

TemplateProject

ProjectList

被如下接口引用：DescribeTemplates

名称	必选	允许NULL	类型	描述
ProjectId	是	是	String	工程id
ProjectName	是	是	String	工程名
BasePackage	是	是	String	包路径
LastTime	是	是	Int64	修改时间
Data	否	是	String	Data
AppId	否	是	String	AppId
Uin	否	是	String	Uin
SubAccountUin	否	是	String	SubAccountUin

ResourceStatus

模块资源的使用情况

被如下接口引用：DescribeResourceUsageRate

名称	必选	允许NULL	类型	描述
Used	是	否	Uint64	该模块多个节点的总的资源使用量

名称	必选	允许NULL	类型	描述
Total	是	否	Uint64	该模块的多个节点的总的资源量
NodesStatus	是	否	Array of ResourceNodeStatus	各个节点的资源的情况

GroupUnitApiUseStatistics

查询网关API监控明细数据（单元化网关使用详情）

被如下接口引用：DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of UnitApiDailyUseStatistics	查询网关API监控明细对象集合

ImageRepositorySecret

镜像仓库密钥

被如下接口引用：DescribeSecretList

名称	必选	允许NULL	类型	描述
Id	否	是	String	凭证ID
SecretName	否	是	String	凭证名称
RepositoryDomain	否	是	String	镜像仓库域名
GroupId	否	是	String	部署组ID
Username	否	是	String	登录用户名
Password	否	是	String	登录密码
AuthType	否	是	String	认证类型，默认-"kubernetes.io/dockercfg"
AuthData	否	是	Array of AuthDataMap	认证数据

RouteAffinity

路由规则就近访问策略

被如下接口引用：DescribeNamespaceAffinities

名称	必选	允许NULL	类型	描述
NamespaceId	是	否	String	命名空间ID

名称	必选	允许NULL	类型	描述
Affinity	是	是	Bool	就近访问策略是否开启
AffinityId	是	是	String	就近访问策略ID
UpdateTimestamp	是	是	String	就近访问策略更新时间

TsfPageGatewayInstance

网关实体分页信息

被如下接口引用：DescribeGatewayInstances

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	数据总数
Content	是	是	Array of GatewayInstance	翻页数据

ImageUserIsExistsResult

ImageUserIsExistsResult

被如下接口引用：DescribeImageUserIsExists

名称	必选	允许NULL	类型	描述
Data	是	是	ImageUserIsExists	用户是否存在

TsfPageStdoutLogV2

标准输出日志列表

被如下接口引用：SearchOssRealtimeBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog、SearchStdoutLog

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of StdoutLogV2	标准输出日志列表
ScrollId	是	是	String	游标ID
Status	是	是	String	查询状态

AgentTask

DescribeApmTaskStatus 接口返回参数封装

被如下接口引用：DescribeApmTaskStatus

名称	必选	允许NULL	类型	描述
TaskId	是	是	String	任务Id
InstanceId	是	是	String	查询的实例Id
Type	是	是	String	任务类型
Status	是	是	String	任务执行状态 AVAILABLE(可执行), BUSY(执行中), ERROR(执行出错), COMPLETED(执行成功)
StatusInfo	是	是	String	status 为ERROR时该字段才有值. EXCEED_SIZE表示采集数据过大, MAL_FUNC表示内部错误 NO_CONNECTION 表示无法和用户进程建立连接, EXECUTE_TIMEOUT 表示任务执行超时
CreationTime	是	是	String	任务创建时间,格式为(yyyy-MM-dd HH:mm:ss)
MetaInfo	是	是	String	元数据,该任务的部分请求参数,如火焰图采集时长{"duration":30}
ExecutedTime	是	是	Int64	任务已执行时长, 单位:秒

ScalableTaskV2

ScalableTask

被如下接口引用：

名称	必选	允许NULL	类型	描述
Taskid	是	是	String	任务id
Desc	是	是	String	任务描述
CreateTime	是	是	String	CreateTime
EndTime	是	是	String	EndTime
TaskInstanceList	是	是	Array of TaskInstance	TaskInstanceList

ContainGroupResult

部署组列表（应用下钻）

被如下接口引用：DescribeContainerGroups

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ContainGroup	部署组列表
TotalCount	是	否	Int64	总记录数

OverviewResource

资源概览数据信息

被如下接口引用：DescribeOverviewResource

名称	必选	允许NULL	类型	描述
ClusterCount	是	是	Int64	集群数目
NamespaceCount	是	是	Int64	命名空间数目
InstanceCount	是	是	Int64	实例总数目
RunInstanceCount	是	是	Int64	运行中实例数目
OffInstanceCount	是	是	Int64	停止实例数目
UnNormalInstanceCount	是	是	Int64	异常实例数目
ApplicationCount	是	是	Int64	应用数目
GroupCount	是	是	Int64	部署组数目
UnNormalClusterCount	是	是	Int64	异常集群数
RunHostCount	是	是	Int64	运行云主机数
HostCount	是	是	Int64	云主机总数
ConsulInstanceCount	是	是	Int64	已注册实例数
PackageSpaceUsed	是	是	Uint64	程序包存储空间用量，单位字节

TsfPageClusterV3

Tsf分页集群对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	集群总数目
Content	是	是	ClusterV3	集群列表

SpanLabel

Span标签

被如下接口引用：DescribeSpanDetail、GetOssTraceSpans、SearchSpan

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
Key	否	是	String	标签键
Value	否	是	String	标签值

TypeCode

术语描述对象，码表

被如下接口引用：DescribeRecordCodes

名称	必选	允许NULL	类型	描述
Code	是	否	String	术语英文描述
Name	是	否	String	术语中文描述

TsfPageMsInstance

微服务实例的分页内容

被如下接口引用：DescribeMicroservice

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	微服务实例总数目
Content	是	是	Array of MsInstance	微服务实例列表内容

ApiRateLimitRule

微服务网关API限流规则

被如下接口引用：DescribeApiRateLimitRules

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	rule Id
ApiId	是	是	String	API ID
RuleName	是	是	String	限流名称
MaxQps	是	是	Uint64	最大限流qps
UsableStatus	是	是	String	生效/禁用, enabled/disabled
RuleContent	是	是	String	规则内容
TsfRuleId	是	是	String	Tsf Rule ID

名称	必选	允许NULL	类型	描述
Description	是	是	String	描述
CreatedTime	是	是	String	创建时间
UpdatedTime	是	是	String	更新时间

MachineList

MachineList

被如下接口引用：CreateMachines、DeleteMachines、ListMachines、ModifyMachines

名称	必选	允许NULL	类型	描述
MachineId	否	是	String	机器id
Ip	否	是	String	ip值
UserName	否	是	String	UserName值
Password	否	是	String	Password值
ModuleNames	否	是	Array of String	ModuleNames值
ZoneId	否	是	String	ZoneId值
ZoneName	否	是	String	ZoneName值
SshPort	否	是	Int64	SshPort值

PathRewrite

路径重写

被如下接口引用：DescribePathRewrite、DescribePathRewrites

名称	必选	允许NULL	类型	描述
PathRewriteId	是	否	String	路径重写规则ID
GatewayGroupId	是	否	String	网关部署组ID
Regex	是	否	String	正则表达式
Replacement	是	否	String	替换的内容
Blocked	是	否	String	是否屏蔽映射后路径，Y: 是 N: 否
Order	是	否	Int64	规则顺序，越小优先级越高

ImageRepository

镜像仓库

被如下接口引用：DescribeImageRepository

名称	必选	允许NULL	类型	描述
Reponame	是	是	String	仓库名,含命名空间,如tsf/nginx
Repotype	是	是	String	仓库类型
TagCount	是	是	Int64	镜像版本数
IsPublic	是	是	Int64	是否公共,1:公有,0:私有
IsUserFavor	是	是	Bool	是否被用户收藏。true：是，false：否
IsQcloudOfficial	是	是	Bool	是否是腾讯云官方仓库。 是否是腾讯云官方仓库。true：是，false：否
FavorCount	是	是	Int64	被所有用户收藏次数
PullCount	是	是	Int64	拉取次数
Description	是	是	String	描述内容
CreationTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

CosCredentialsV2

cos临时帐号信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
SessionToken	是	是	String	无
TmpAppId	是	是	String	无
TmpSecretId	是	是	String	无
TmpSecretKey	是	是	String	无
ExpiredTime	是	是	Int64	无
Domain	是	是	String	无

ImageRepositoryResult

镜像仓库列表

被如下接口引用：DescribeImageRepository

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总记录数
Server	是	是	String	镜像服务器地址
Content	是	是	Array of ImageRepository	列表信息

TsfZone

TSF可用区信息

被如下接口引用：DescribeTsfZones

名称	必选	允许NULL	类型	描述
Id	是	是	String	TSF可用区ID
Name	是	是	String	TSF可用区名称
TsfRegionId	是	是	String	TSF可用区所属TSF地域ID

InstanceRequests

多条实例请求详情

被如下接口引用：DescribeInstanceRequest

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	内容
Content	是	是	Array of InstanceRequest	总条数

ApplicationForPageV2

分页的应用描述信息字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
ApplicationDesc	是	是	String	应用描述
ApplicationType	是	是	String	应用类型

名称	必选	允许NULL	类型	描述
MicroserviceType	是	是	String	微服务类型
ProgLang	是	是	String	编程语言
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

ThreadPicture

jvm监控数据线程数据封装

被如下接口引用：DescribeJvmMonitor

名称	必选	允许NULL	类型	描述
ThreadCount	是	否	Array of CurvePoint	总线程数
ThreadActive	是	否	Array of CurvePoint	活跃线程数
DeamonThreadCount	是	否	Array of CurvePoint	守护线程数

ListScalableRuleResultV2

ListScalableRuleResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of ScalableRule	Content

DtsTransaction

DTS主事务

被如下接口引用：

名称	必选	允许NULL	类型	描述
TxId	是	是	Int64	主事务ID
TxBegin	是	是	Int64	主事务开始时间戳，UTC，精确到毫秒
TxEnd	是	是	Int64	主事务结束时间戳，UTC，精确到毫秒
TxCommit	是	是	Int64	主事务提交时间戳，UTC，精确到毫秒

名称	必选	允许NULL	类型	描述
TxRollback	是	是	Int64	主事务回滚时间戳，UTC，精确到毫秒
TxError	是	是	Int64	主事务异常停止时间戳，UTC，精确到毫秒
Timeout	是	是	Int64	主事务超时时长，单位毫秒
Status	是	是	Int64	主事务状态
EndFlag	是	是	Int64	主事务结束标识
TimeoutFlag	是	是	Int64	主事务超时标识
Comment	是	是	String	异常信息
GroupId	是	是	String	事务分组ID
Server	是	是	String	主事务来源服务标识
BranchQty	是	是	Int64	分支事务数量
RetryFlag	是	是	Bool	重试标识：true：可以重试；false：不可重试

CircuitBreakerStrategy

熔断策略

被如下接口引用：CreateCircuitBreakerRule、DescribeCircuitBreakerRule、DescribeCircuitBreakerRules、UpdateCircuitBreakerRule

名称	必选	允许NULL	类型	描述
StrategyId	否	否	String	熔断策略ID
RuleId	否	否	String	熔断策略所属熔断规则ID
MinimumNumberOfCalls	是	否	Int64	最少请求数
SlidingWindowSize	是	否	Int64	滚动窗口统计时间
WaitDurationInOpenState	是	否	Int64	熔断开启到半开间隔,单位s
FailureRateThreshold	否	是	Int64	失败请求比例
MaxEjectionPercent	否	是	Int64	最大熔断实例的比例
SlowCallDurationThreshold	否	是	Int64	慢请求阈值
SlowCallRateThreshold	否	是	Int64	慢请求比例
ApiList	否	是	Array of CircuitBreakerApi	熔断策略作用API

FileConfigRelease

文件配置项发布信息

被如下接口引用：DescribeFileConfigReleases

名称	必选	允许NULL	类型	描述
ConfigReleaseId	否	是	String	配置项发布ID
ConfigId	否	是	String	配置项ID
ConfigName	否	是	String	配置项名称
ConfigVersion	否	是	String	配置项版本
ReleaseDesc	否	是	String	发布描述
ReleaseTime	否	是	String	发布时间
GroupId	否	是	String	部署组ID
GroupName	否	是	String	部署组名称
NamespaceId	否	是	String	命名空间ID
NamespaceName	否	是	String	命名空间名称
ClusterId	否	是	String	集群ID
ClusterName	否	是	String	集群名称

MonitorStatisticsPolicyGroup

MonitorStatisticsPolicyGroup

被如下接口引用：DescribeMonitorStatisticsPolicy、ListMonitorStatisticsPolicy

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	GroupId
GroupName	是	是	String	GroupName
GroupDesc	是	是	String	GroupDesc
NamespaceId	是	是	String	NamespaceId
NamespaceName	是	是	String	NamespaceName
ApplicationId	是	是	String	ApplicationId
ApplicationName	是	是	String	ApplicationName
ClusterId	是	是	String	ClusterId
ClusterName	是	是	String	ClusterName

GatewayGroupApiVo

网关API简单信息

被如下接口引用：DescribeGatewayAllGroupApis

名称	必选	允许NULL	类型	描述
ApiId	是	否	String	API ID
Path	是	否	String	API 请求路径
MicroserviceName	是	否	String	API 微服务名称
Method	是	是	String	API 请求方法
NamespaceName	是	是	String	命名空间名称

OverviewClusterResourceUsage

集群概览页信息

被如下接口引用：DescribeClusterUsage

名称	必选	允许NULL	类型	描述
ClusterId	是	否	String	集群ID
ClusterName	是	否	String	集群名
ClusterTotalCpu	是	否	Float	集群总 CPU
ClusterUsedCpu	是	否	Float	集群已使用 CPU
ClusterTotalMem	是	否	Float	集群总内存
ClusterUsedMem	是	否	Float	集群已使用内存

MicroserviceExtra

微服务（信息更丰富版）

被如下接口引用：DescribeMicroservicesByAssociateApplication

名称	必选	允许NULL	类型	描述
MicroserviceId	否	是	String	微服务 ID
MicroserviceName	否	是	String	名称
MicroserviceDesc	否	是	String	描述
MicroserviceType	否	是	String	类型

名称	必选	允许NULL	类型	描述
AssociateApplicationId	否	是	String	关联的应用 ID
MicroserviceProtocol	否	是	String	协议
MicroserviceListeningPort	否	是	Int64	监听端口
HealthCheckUrl	否	是	String	健康检查参数
ServiceClustertype	否	是	String	集群类型
CreateTime	否	是	Int64	创建时间
UpdateTime	否	是	Int64	更新时间
NamespaceId	否	是	String	命名空间 ID
NamespaceName	否	是	String	命名空间名称
ClusterId	否	是	String	集群 ID
ClusterName	否	是	String	集群名称

IpListResult

IpListResult

被如下接口引用：CreateServiceInstance

名称	必选	允许NULL	类型	描述
InstanceId	否	是	String	InstanceId
Ip	否	是	String	Ip
UserName	否	是	String	UserName
Password	否	是	String	Password
MachineId	否	是	String	machineId

ServiceInfo

微服务明细

被如下接口引用：

名称	必选	允许NULL	类型	描述
MicroserviceId	是	是	String	微服务ID
MicroserviceName	是	是	String	微服务名称

TsfPageStdoutLog

标准输出日志列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of StdoutLog	标准输出日志列表

TsfPageBusinessLogV2

业务日志列表

被如下接口引用：SearchBusinessLog、SearchOssBusinessLog、SearchOssRealtimeBusinessLog、SearchOssSpanBusinessLog、SearchOssStaticBusinessLog、SearchOssSurroundBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog、SearchSpanBusinessLog、SearchSurroundBusinessLog

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of BusinessLogV2	业务日志列表
ScrollId	是	是	String	游标ID
Status	是	是	String	查询状态

JvmLog

Jvm日志查询时每行日志详情封装

被如下接口引用：DescribeJvmLogs

名称	必选	允许NULL	类型	描述
Content	是	是	String	日志内容
InstanceIp	是	是	String	实例ip
InstanceId	是	是	String	实例id
Timestamp	是	是	String	日志产生的时间戳
GroupId	是	是	String	部署组Id

ZipkinSpanInfoV2

调用链Span信息

被如下接口引用：GetOssTraceSpans

名称	必选	允许NULL	类型	描述
Id	是	是	String	Span ID
TraceId	是	是	String	调用链ID
ParentId	是	是	String	父Span ID
Name	是	是	String	Span名称
Timestamp	是	是	Uint64	Span时间戳
Duration	是	是	Uint64	Span耗时
ResultStatus	是	是	String	Span结果
ServiceName	是	是	String	Span服务名（前端展示）
AnnotationList	是	是	Array of ZipkinAnnotationV2	Span注解列表
BinaryAnnotationList	是	是	Array of ZipkinBinaryAnnotationV2	Span二进制注解列表
MetadataList	是	是	Array of ZipkinMetadataV2	Span元数据列表
InterfaceName	是	是	String	Span接口名（前端展示）
LocalEndpoint	是	是	ZipkinEndpointV2	Span本地端点
RemoteEndpoint	是	是	ZipkinEndpointV2	Span远程端点
LabelList	是	是	Array of SpanLabel	Span标签列表

UnitRuleItem

微服务网关单元化规则项

被如下接口引用：CreateUnitRule、DescribeEnabledUnitRule、DescribeUnitRule、DescribeUnitRules、UpdateUnitRule

名称	必选	允许NULL	类型	描述
Id	否	是	String	规则项ID
UnitRuleId	否	是	String	单元化规则ID
Relationship	是	否	String	逻辑关系：AND/OR
DestNamespaceId	是	否	String	目的地命名空间ID
DestNamespaceName	是	否	String	目的地命名空间名称
Name	是	否	String	规则项名称
Priority	否	是	Int64	规则顺序，越小优先级越高：默认为0

名称	必选	允许NULL	类型	描述
Description	否	是	String	规则描述
UnitRuleTagList	否	是	Array of UnitRuleTag	规则标签列表

RepositoryErrorV2

仓库相关错误

被如下接口引用：

名称	必选	允许NULL	类型	描述
Code	是	是	String	无
Message	是	是	String	无

DtsRetryResp

DTS重试主事务响应

被如下接口引用：RetryTransactions

名称	必选	允许NULL	类型	描述
ErrorList	是	是	Array of DtsRetryError	异常列表

TsfPageTraceSpanInfo

分页spaninfo的结构

被如下接口引用：SearchSpan

名称	必选	允许NULL	类型	描述
TotalCount	否	是	Uint64	总数
Content	否	是	Array of TraceSpanInfo	span列表

TaskExecuteHistoryRecord

任务执行记录的流水列表

被如下接口引用：DescribeTaskExecuteHistoryRecords

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ExecuteLogId	是	否	String	执行流水ID
ExecuteId	是	否	String	执行ID
GroupId	是	是	String	任务执行分组
InstanceId	是	否	String	实例ID
State	是	否	String	实例执行状态
TriggerType	是	否	String	触发方式
StartTime	是	否	Int64	开始时间
EndTime	是	是	Int64	结束时间
TimeOut	是	是	Int64	超时时间
ExecuteLog	是	是	String	执行日志
HistoryCount	是	是	Int64	历史执行次数
ShardKey	是	是	Uint64	分片Key

VpcConfig

vpc 配置信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
VpcId	是	否	String	VpcId
SubnetId	是	否	String	SubnetId 子网ID

AddInstanceResult

添加实例到集群的结果

被如下接口引用：AddClusterInstances

名称	必选	允许NULL	类型	描述
FailedInstanceIds	是	是	Array of String	添加集群失败的节点列表
SuccInstanceIds	是	是	Array of String	添加集群成功的节点列表
TimeoutInstanceIds	是	是	Array of String	添加集群超时的节点列表
FailedReasons	是	是	Array of String	失败的节点的失败原因

ContainerGroupResultV2

镜像版本列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
Content	是	是	Array of ContainerGroup	镜像版本列表
TotalCount	是	是	Int64	总记录数

VmSubTaskV2

虚拟机子任务信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	机器Id
LanIp	是	是	String	内网IP
WanIp	是	是	String	外网IP
InstanceName	是	是	String	机器名称
AgentState	是	是	Int64	机器Agent状态
InstanceState	是	是	Int64	机器状态
TaskStatus	是	是	Int64	任务状态
TaskRecord	是	是	String	任务记录

SimpleKafkaDeliveryConfig

日志投递kafka配置描述的缩简版

被如下接口引用：DescribeDeliveryConfigByGroupId

名称	必选	允许NULL	类型	描述
ConfigId	是	是	String	配置项id
ConfigName	是	是	String	配置项名称

CloudMonitorPolicies

云监控对象策略对象

被如下接口引用：ListColudMonitorStatisticsPolicy

名称	必选	允许NULL	类型	描述
KeywordsId	是	是	String	keywordsId
KeyWords	是	是	String	keyWords
GroupId	是	是	String	groupId
GroupName	是	是	String	groupName
NamespaceId	是	是	String	namespaceId
NamespaceName	是	是	String	namespaceName
ApplicationId	是	是	String	applicationId
ApplicationName	是	是	String	applicationName
ClusterId	是	是	String	clusterId
ClusterName	是	是	String	clusterName

TaskFlow

工作流对象

被如下接口引用：DescribeTaskFlows

名称	必选	允许NULL	类型	描述
FlowId	是	否	String	工作流Id
FlowName	是	否	String	工作流名称
State	是	是	String	状态
TaskCount	是	是	Int64	任务数量
TriggerRule	是	否	TaskRule	触发规则
TimeOut	是	是	Int64	工作流超时时间
FlowLogId	是	否	String	工作流历史 ID
GraphId	是	是	String	工作流图 ID

GatewayApiGroupVo

网关分组简单信息

被如下接口引用：DescribeGatewayAllGroupApis

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	分组ID
GroupName	是	是	String	分组名称
GroupApiCount	是	是	Uint64	分组下API个数
GroupApis	是	是	Array of GatewayGroupApiVo	分组API列表

FileConfig

文件配置项

被如下接口引用：DescribeFileConfig、DescribeFileConfigSummary、DescribeFileConfigs

名称	必选	允许NULL	类型	描述
ConfigId	否	是	String	配置项ID
ConfigName	否	是	String	配置项名称
ConfigVersion	否	是	String	配置项版本
ConfigVersionDesc	否	是	String	配置项版本描述
ConfigFileName	否	是	String	配置项文件名
ConfigFileValue	否	是	String	配置项文件内容
ConfigFileCode	否	是	String	配置项文件编码
CreationTime	否	是	String	创建时间
ApplicationId	否	是	String	配置项归属应用ID
ApplicationName	否	是	String	应用名称
DeleteFlag	否	是	Bool	删除标识
ConfigVersionCount	否	是	Int64	配置项版本数量
LastUpdateTime	否	是	String	配置项最后更新时间
ConfigFilePath	否	是	String	发布路径
ConfigPostCmd	否	是	String	后置命令
ConfigFileValueLength	否	是	Uint64	配置项文件长度

GetClusterLimitResourceResult

获取集群limit资源(GetClusterLimitResource)返回对象

被如下接口引用：GetClusterLimitResource

名称	必选	允许NULL	类型	描述
CpuLimit	是	否	String	最大分配cpu 核数
MemLimit	是	否	String	最大分配mem内存数，单位M

HealthCheckSetting

健康检查配置信息，若不指定该参数，则默认不设置健康检查。

被如下接口引用：DeployContainerGroup、DeployGroup、DescribeContainerGroupAttribute、DescribeContainerGroupDeployInfo、DescribeContainerGroupDetail、DescribeGroup、UpdateHealthCheckSettings

名称	必选	允许NULL	类型	描述
ActionType	是	是	String	健康检查方法。HTTP：通过 HTTP 接口检查；CMD：通过执行命令检查；TCP：通过建立 TCP 连接检查。
InitialDelaySeconds	否	是	Uint64	容器延时启动健康检查的时间。
TimeoutSeconds	否	是	Uint64	每次健康检查响应的最大超时时间。
PeriodSeconds	否	是	Uint64	进行健康检查的时间间隔。
SuccessThreshold	否	是	Uint64	表示后端容器从失败到成功的连续健康检查成功次数。
FailureThreshold	否	是	Uint64	表示后端容器从成功到失败的连续健康检查成功次数。
Scheme	否	是	String	HTTP 健康检查方法使用的检查协议。支持HTTP、HTTPS。
Port	否	是	Uint64	健康检查端口，范围 1~65535 。
Path	否	是	String	HTTP 健康检查接口的请求路径。
Command	否	是	Array of String	执行命令检查方式，执行的命令。
Type	否	是	String	TSF_DEFAULT：tsf 默认就绪探针。K8S_NATIVE：k8s 原生探针。不填默认为 k8s 原生探针。

DescribeSingleContainerGroupsV2

部署组-独立菜单

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	否	是	String	groupId
GroupName	否	是	String	分组名称

名称	必选	允许NULL	类型	描述
CpuLimit	否	是	String	最大分配的 CPU 核数，对应 K8S limit
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
ClusterId	是	是	String	集群id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用名称
ApplicationName	是	是	String	应用类型
ApplicationType	是	是	String	分组创建时间
MemLimit	是	是	String	最大分配的内存 MiB 数，对应 K8S limit
MicroserviceType	是	是	String	微服务类型
ProtocolPorts	是	是	Array of ProtocolPort	端口列表
UpdateType	是	是	Int64	更新类型
UpdateIvl	是	是	Int64	更新时间间隔
AccessType	是	是	Int64	网络访问类型
CreateTime	是	是	String	创建时间
CpuRequest	是	是	String	初始分配的 CPU 核数，对应 K8S request
MemRequest	是	是	String	初始分配的内存 MiB 数，对应 K8S request
SubnetId	是	是	String	子网id

TaskBatchRecordPage

任务批次翻页对象

被如下接口引用：DescribeTaskBatchHistoryRecords、DescribeTaskBatchRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数量
Content	是	否	Array of TaskBatchRecord	任务批次翻页查询结果

ApiInfo

微服务网关API信息

被如下接口引用：CreateGatewayApi

名称	必选	允许NULL	类型	描述
NamespaceId	是	否	String	命名空间Id，若为外部API,为固定值："namespace-external"
MicroserviceId	是	否	String	服务Id，若为外部API,为固定值："ms-external"
Path	是	否	String	API path
Method	是	否	String	Api 请求
PathMapping	是	否	String	请求映射
Host	否	否	String	api所在服务host,限定外部Api填写。格式: `http://imgcache.finance.cloud.tencent.com:80127.0.0.1:8080`
Description	否	否	String	api描述信息

BusinesLogConfigAssociatedGroup

业务日志配置关联部署组信息

被如下接口引用：DescribeBusinessLogConfig、DescribeBusinessLogConfigs、DescribeGroupBusinessLogConfigs

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	部署组所属应用ID
ApplicationName	是	是	String	部署组所属应用名称
ApplicationType	是	是	String	部署组所属应用类型
NamespaceId	是	是	String	部署组所属命名空间ID
NamespaceName	是	是	String	部署组所属命名空间名称
ClusterId	是	是	String	部署组所属集群ID
ClusterName	是	是	String	部署组所属集群名称
ClusterType	是	是	String	部署组所属集群类型
AssociatedTime	是	是	String	部署组关联日志配置时间

PackageConfig

程序包相关配置信息

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
SpaceSize	是	是	Uint64	程序包存储空间大小，单位字节

PortsResult

PortsResult

被如下接口引用：CreateServiceInstance、ListTsfModulesDetail

名称	必选	允许NULL	类型	描述
PortId	否	是	String	PortId值
PortType	否	是	String	PortType值
Port	否	是	String	Port值
DefaultPort	否	是	String	DefaultPort值
ModuleId	否	是	String	ModuleId值
EnableRegInConsul	否	是	Int64	EnableRegInConsul值

BillingOperationRecord

计费操作记录

被如下接口引用：DescribeBillingOperationRecords

名称	必选	允许NULL	类型	描述
ResourceId	是	是	String	资源ID
CategoryId	是	是	Int64	类别ID
CreateTime	是	是	String	创建时间
AppId	是	是	String	租户ID
Uin	是	是	String	账号ID
Nickname	是	是	String	用户昵称
Operator	是	是	String	操作人
Description	是	是	String	描述
OperationDetail	是	是	String	操作详情

MemoryPicture

Jvm监控内存数据封装

被如下接口引用：DescribeJvmMonitor

名称	必选	允许NULL	类型	描述
Max	是	否	Array of CurvePoint	内存最大值
Used	是	否	Array of CurvePoint	已用内存大小
Committed	是	否	Array of CurvePoint	系统分配内存大小

AppPkgInfo

应用的包信息

被如下接口引用：ListAppPkg

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用ID
PkgCount	是	是	Int64	应用下的程序包数量

JvmLockDetail

Jvm检测死锁返回死锁详情封装

被如下接口引用：RunJvmDeadLockCheck

名称	必选	允许NULL	类型	描述
ThreadCount	是	是	Int64	死锁线程数, 没有死锁时为0
LockInfos	是	是	String	死锁检测响应信息,没有死锁时为空串""
Status	是	是	String	调用成功success/调用失败error
StatusInfo	是	是	String	调用成功为""/调用失败为对应失败信息
StatusCode	是	是	Int64	调用成功为0/调用失败为对应的错误码

TsfPageConfig

TsfPage<Config>

被如下接口引用：DescribeConfigSummary、DescribeConfigs、DescribePublicConfigSummary、DescribePublicConfigs

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TsfPageConfig
Content	是	否	Array of Config	配置项列表

LicenseProduct

LicenseProduct

被如下接口引用：DescribeLicenseApplications

名称	必选	允许NULL	类型	描述
Name	是	否	String	产品名
Spec	否	是	String	产品规格
Function	否	是	Array of LicenseFunction	选配功能列表
Resource	否	是	Array of LicenseResource	受限资源列表

ZipkinEndPoint

调用链Span端点信息

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
ServiceName	是	是	String	端点服务名
Ipv4	是	是	String	端点IP地址（v4）
Ipv6	是	是	String	端点IP地址（v6）
Port	是	是	Int64	端点端口号

TsfPageClusterV2

Tsf分页集群对象

被如下接口引用：DescribeClusters

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	集群总数目
Content	是	是	Array of ClusterV2	集群列表

TsfPageBusinessLogConfigV2

业务日志配置项列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of BusinessLogConfig	业务日志配置项列表

DockerForUseV2

Docker使用指引

被如下接口引用：

名称	必选	允许NULL	类型	描述
Server	是	否	String	仓库中心地址
Username	是	否	String	用户名

TsfPageConfigReleaseV2

TSF配置项发布信息分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of ConfigRelease	配置项发布信息数组

TxList

查询事务列表

被如下接口引用：DescribeTransactions

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	返回的事务数量
Content	是	是	Array of TxMainTransaction	主事务信息
Error	是	是	TxError	请求异常信息

CosUploadInfoV2

cos上传所需信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
PkgId	是	是	String	无
Bucket	是	是	String	无
Region	是	是	String	无
Path	是	是	String	无
Credentials	是	否	CosCredentials	无

MonitorStatisticsPolicyResult

MonitorStatisticsPolicyResult

被如下接口引用：ListMonitorStatisticsPolicy

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of MonitorStatisticsPolicy	Content

AuthRule

微服务权限规则

被如下接口引用：DescribeAuthorization、DescribeAuthorizationType、DescribeAuthorizations

名称	必选	允许NULL	类型	描述
RuleId	是	否	String	规则ID
RuleName	是	否	String	规则名称
IsEnabled	是	否	String	是否启用
CreateTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间
MicroserviceId	是	否	String	微服务ID
Tags	是	否	Array of AuthConditionV2	标签列表

OverviewResourceV2

资源概览数据信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
ClusterCount	是	是	Int64	集群数目
NamespaceCount	是	是	Int64	命名空间数目
InstanceCount	是	是	Int64	实例总数目
RunInstanceCount	是	是	Int64	运行中实例数目
OffInstanceCount	是	是	Int64	停止实例数目
UnNormalInstanceCount	是	是	Int64	异常实例数目
ApplicationCount	是	是	Int64	应用数目
GroupCount	是	是	Int64	部署组数目

TsfPageUnitRule

单元化规则翻页对象

被如下接口引用：DescribeUnitRules

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	记录总数
Content	是	否	Array of UnitRule	记录实体列表

RouteDestItemV2

服务路由目标匹配项

被如下接口引用：CreateRoute、DescribeRoute、DescribeRoutes、ModifyRoute

名称	必选	允许NULL	类型	描述
RouteDestItemId	否	是	String	路由规则路由目标匹配项ID
DestItemField	是	否	String	路由规则目标字段名称
DestItemValue	是	否	String	路由规则目标字段取值
RouteDestId	否	是	String	所属路由规则路由目标ID

GatewayOAuthPlugin

GatewayOAuthPlugin

被如下接口引用：DescribeGatewayOAuthPlugin

名称	必选	允许NULL	类型	描述
Id	是	否	String	网关插件id
Name	是	是	String	插件名称
Description	是	是	String	插件描述
Type	是	是	String	插件类型
CreatedTime	是	是	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	是	String	插件更新时间 如:2019-06-20 15:51:28
TokenAuthUrl	是	是	String	验证token路径
TokenAuthMethod	是	是	String	验证token请求方法:get/post
ExpireTime	是	是	Int64	认证请求超时时间,单位:秒 范围:0~30
RedirectUrl	是	是	String	重定向地址
TokenBaggagePosition	是	是	String	token携带位置，网关取token位置与发送认证请求时token位置一致,值:query/header
TokenKeyName	是	是	String	token的key值
PayloadMappingName	是	是	String	payload的映射参数名称
PayloadMappingPosition	是	是	String	payload映射到后端服务的携带位置,值:query/header
Status	是	是	String	发布状态: drafted/released

ResourceBatchOperationStatusResult

描述批次操作的执行结果

被如下接口引用：ContinueResourceBatchOperation

名称	必选	允许NULL	类型	描述
OperationResult	是	否	Bool	批次操作的执行结果
BatchNum	是	否	Uint64	下一个需要执行操作的批次索引

GetClusterLimitResourceResultV2

获取集群limit资源(GetClusterLimitResource)返回对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
CpuLimit	是	否	String	最大分配cpu 核数
MemLimit	是	否	String	最大分配mem内存数，单位M

GroupList

部署组列表信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	部署组id
GroupName	是	否	String	部署组名

Interface

描述接口信息

被如下接口引用：DescribeInstanceRequest

名称	必选	允许NULL	类型	描述
MicroserviceId	是	是	String	微服务id
Path	是	是	String	路径模版
Method	是	是	String	http请求方法

ContainerGroupDeploy

获取部署组

被如下接口引用：DescribeContainerGroupDeployInfo

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组id
GroupName	是	是	String	分组名称
InstanceNum	是	是	Int64	实例总数
CurrentNum	是	是	Int64	已启动实例总数

名称	必选	允许NULL	类型	描述
Server	是	是	String	镜像server
Reponame	是	是	String	镜像名，如/tsf/nginx
TagName	是	是	String	镜像版本名称
CpuRequest	是	是	String	业务容器初始分配的 CPU 核数，对应 K8S request
CpuLimit	是	是	String	业务容器最大分配的 CPU 核数，对应 K8S limit
MemRequest	是	是	String	业务容器初始分配的内存 MiB 数，对应 K8S request
MemLimit	是	是	String	业务容器最大分配的内存 MiB 数，对应 K8S limit
AccessType	是	是	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	是	是	Array of ProtocolPort	端口映射
UpdateType	是	是	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	是	Int64	更新间隔,单位秒
JvmOpts	是	是	String	jvm参数
SubnetId	是	是	String	子网id
AgentCpuRequest	是	是	String	agent容器初始分配的 CPU 核数，对应 K8S request
AgentCpuLimit	是	是	String	agent容器最大分配的 CPU 核数，对应 K8S limit
AgentMemRequest	是	是	String	agent容器初始分配的内存 MiB 数，对应 K8S request
AgentMemLimit	是	是	String	agent容器最大分配的内存 MiB 数，对应 K8S limit
IstioCpuRequest	是	是	String	istioproxy容器初始分配的 CPU 核数，对应 K8S request
IstioCpuLimit	是	是	String	istioproxy容器最大分配的 CPU 核数，对应 K8S limit
IstioMemRequest	是	是	String	istioproxy容器初始分配的内存 MiB 数，对应 K8S request
IstioMemLimit	是	是	String	istioproxy容器最大分配的内存 MiB 数，对应 K8S limit
Envs	是	是	Array of Env	部署组的环境变量数组，这里没有展示 tsf 使用的环境变量，只展示了用户设置的环境变量。
HealthCheckSettings	是	是	HealthCheckSettings	健康检查配置信息，若不指定该参数，则默认不设置健康检查。
DeployAgent	是	是	Bool	是否部署Agent容器

LicenseTag

许可标签

被如下接口引用：DescribeLicenses

名称	必选	允许NULL	类型	描述
LicenseId	是	是	String	许可ID
Tags	是	是	Array of Tag	标签列表

CircuitBreakerApi

熔断API规则

被如下接口引用：CreateCircuitBreakerRule、DescribeCircuitBreakerRule、DescribeCircuitBreakerRules、UpdateCircuitBreakerRule

名称	必选	允许NULL	类型	描述
ApiId	否	否	String	API ID
Path	是	否	String	API PATH
Method	是	否	String	API Method
StrategyId	否	否	String	熔断策略ID

TsfPageConfigSummaryV2

TsfPage<Config>

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总条数
Content	是	否	Array of ConfigSummary	配置项列表

AuthDataMap

认证数据map

被如下接口引用：DescribeSecretList

名称	必选	允许NULL	类型	描述
Key	否	否	String	键

名称	必选	允许NULL	类型	描述
Value	否	否	String	值

ApiResponseDescr

API 响应的参数结构描述

被如下接口引用：DescribeApiDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	参数描述
Type	是	否	String	参数类型
Description	是	否	String	参数描述

EnvsV2

环境变量

被如下接口引用：AnalyzeLogSchema、ValidateLogSchema

名称	必选	允许NULL	类型	描述
Name	是	否	String	环境变量名称
Value	是	否	String	服务端口

WeightRouteItemV2

权重路由规则项

被如下接口引用：

名称	必选	允许NULL	类型	描述
WeightRouteId	否	否	String	权重路由规则项ID
SourcePercent	是	否	Int64	权重路由，百分比数值
TargetField	是	否	String	权重路由规则匹配目标字段
TargetValue	是	否	String	权重路由规则匹配目标取值
RouteRuleId	否	否	String	权重路由规则所属路由ID

TsfPageAuthorizationV2

虽然挂着 TsfPage，但是它没有继承 TsfPage 类（后面可能会）。这是个全量接口。

被如下接口引用：

名称	必选	允许NULL	类型	描述
IsEnabled	是	否	Bool	鉴权规则是否被启用
Conditions	是	是	Array of AuthCondition	鉴权规则列表

SimpleClusterV2

简单集群

被如下接口引用：

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterType	是	是	String	集群类型

TaskExecuteRecord

任务执行信息

被如下接口引用：DescribeTaskExecuteRecords

名称	必选	允许NULL	类型	描述
BatchId	是	否	String	批次ID
ExecuteId	是	否	String	执行ID
State	是	否	String	状态
StartTime	是	是	Uint64	开始时间
EndTime	是	是	Uint64	结束时间
InstanceId	是	否	String	实例ID
RetryCount	是	是	Uint64	重试次数
GroupId	是	是	String	任务执行分组
TimeOut	是	是	Uint64	超时时间，单位ms
HistoryCount	是	是	Int64	执行历史数量
TriggerType	是	是	String	触发类型

名称	必选	允许NULL	类型	描述
ShardKey	是	是	Uint64	分片索引参数
ExecuteLogId	是	是	String	执行历史 ID
BatchLogId	是	是	String	批次历史 ID
TaskId	是	否	String	任务 ID
BatchType	是	是	String	批次类型
FailRetryTime	是	是	Int64	失败重试次数
ExecuteLog	是	是	String	执行日志
ExtraData	是	是	String	附加数据

ApiGroupInfo

API分组信息

被如下接口引用：DescribeApiGroup、DescribeApiGroups、DescribeGroupGateways、DescribeGroupsWithPlugin

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	Api Group Id
GroupName	是	是	String	Api Group 名称
GroupContext	是	是	String	分组上下文
AuthType	是	是	String	鉴权类型。 secret： 密钥鉴权； none:无鉴权
Status	是	是	String	发布状态, drafted: 未发布。 released: 发布
CreatedTime	是	是	String	分组创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	是	String	分组更新时间 如:2019-06-20 15:51:28
BindedGatewayDeployGroups	是	是	Array of GatewayDeployGroup	api分组已绑定的网关部署组
ApiCount	是	是	Int64	api 个数
AclMode	是	是	String	访问group的ACL类型
Description	是	是	String	描述
GroupType	是	是	String	分组类型。 ms： 微服务分组； external: 外部Api分组

RatelimitRuleForUpdate

用于修改删除操作的限流规则

被如下接口引用：DeleteRatelimit、DisableRatelimit、EnableRatelimit

名称	必选	允许NULL	类型	描述
Id	是	否	String	规则ID
Name	否	否	String	规则名字，在一个微服务下唯一
Status	否	否	Uint64	状态 0表示启用 1表示停用
DurationSecond	否	否	Uint64	限流周期，单位秒
DurationQuota	否	否	Uint64	每周期内的限流配额
Description	否	否	String	描述

TsfPageRouteReleaseHistoryV2

路由规则启停记录分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	服务路由部署记录列表总数目
Content	是	是	Array of RouteReleaseHistoryV2	服务路由规则部署记录列表信息

UnitApiDailyUseStatistics

UnitApiDailyUseStatistics

被如下接口引用：DescribeUnitApiUseDetail

名称	必选	允许NULL	类型	描述
NamespaceId	是	是	String	NamespaceId
NamespaceName	是	是	String	NamespaceName
SumReqAmount	是	是	String	SumReqAmount
AvgFailureRate	是	是	String	AvgFailureRate
AvgTimeCost	是	是	String	AvgTimeCost
MetricDataPointMap	是	是	Array of MetricDataPointMap	MetricDataPointMap
TopStatusCode	是	是	Array of ApiDailyUseStatisticsEntity	TopStatusCode

名称	必选	允许NULL	类型	描述
TopTimeCost	是	是	Array of ApiDailyUseStatisticsEntity	TopTimeCost
Quantile	是	是	QuantileEntity	Quantile

TaskFlowBatchHistoryPage

批次历史分页

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数
Content	是	否	Array of TaskFlowBatch	分页数据

ScalableRuleAttribute

ScalableRuleAttribute

被如下接口引用：DescribeScalableRuleAttribute

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	RuleId
Name	是	是	String	Name
EnableShrink	是	是	Int64	EnableShrink
EnableExpand	是	是	Int64	EnableExpand
ExpandPeriod	是	是	Int64	ExpandPeriod
ShrinkPeriod	是	是	Int64	ShrinkPeriod
ExpandVmCount	是	是	Int64	ExpandVmCount
ShrinkVmCount	是	是	Int64	ShrinkVmCount
CoolTime	是	是	Int64	CoolTime
ExpandVmCountLimit	是	是	Int64	ExpandVmCountLimit
ShrinkVmCountLimit	是	是	Int64	ShrinkVmCountLimit
ExpandSubruleList	是	是	Array of ExpandSubrule	ExpandSubruleList
ShrinkSubruleList	是	是	Array of ExpandSubrule	ShrinkSubruleList
CreateTime	是	是	String	CreateTime

名称	必选	允许NULL	类型	描述
UpdateTime	是	是	String	UpdateTime

TaskRecordPage

翻页查询的任务记录返回

被如下接口引用：DescribeTaskRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总数量
Content	是	否	Array of TaskRecord	任务记录列表

ConfigSummary

配置项统计信息，其实就是Config的部分字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigName	是	否	String	配置项名称
ConfigVersionCount	是	否	Int64	配置项值
LastUpdateTime	是	否	String	配置项类型
ApplicationId	是	否	String	应用ID
ApplicationName	是	否	String	应用名称

ClusterLimitResource

集群limit资源

被如下接口引用：

名称	必选	允许NULL	类型	描述
CpuLimit	是	否	String	最大分配cpu 核数
MemLimit	是	否	String	最大分配内存M数

DumpResultV2

DumpResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
Name	否	是	String	Name
State	否	是	String	State
Detail	否	是	String	Detail

TxError

事务查询异常信息

被如下接口引用：AutoRetryTransaction、DescribeTransactions、RetryTransaction

名称	必选	允许NULL	类型	描述
Code	是	是	String	错误码
Message	是	是	String	错误信息

ScalableRule

ScalableRule

被如下接口引用：ListScalableRule

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	RuleId
Name	是	是	String	Name
ExpandVmCountLimit	是	是	Int64	ExpandVmCountLimit
ShrinkVmCountLimit	是	是	Int64	ShrinkVmCountLimit
GroupCount	是	是	Int64	GroupCount

ContainerVolumeOption

容器卷挂载信息

被如下接口引用：DescribeContainerVolumeOptions

名称	必选	允许NULL	类型	描述
ConfigMap	否	否	VolumeItem	ConfigMap配置
Secret	否	否	VolumeItem	Secret配置

名称	必选	允许NULL	类型	描述
PersistentVolumeClaim	否	否	VolumeItem	Pvc配置

ApiRequestDescr

ApiRequestDescr

被如下接口引用：DescribeApiDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	参数名称
Type	是	否	String	参数类型
In	是	否	String	参数位置
Description	是	否	String	参数描述
Required	是	否	Bool	参数是否必须
DefaultValue	是	是	String	参数的默认值

ApiStatusResponse

API在线状态

被如下接口引用：DescribeApiStatus

名称	必选	允许NULL	类型	描述
Status	是	是	Int64	API 状态：0:离线，1:在线

TsfPageConfigReleaseLogV2

TSF配置项发布日志分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of ConfigReleaseLog	配置项发布日志数组

DtsBranch

DTS分支事务

被如下接口引用：

名称	必选	允许NULL	类型	描述
BranchId	是	是	Int64	分支事务ID
ParentId	是	是	Int64	父级分支事务ID，顶级时返回null
BranchBegin	是	是	Int64	分支事务开始时间戳，UTC，精确到毫秒
BranchEnd	是	是	Int64	分支事务结束时间戳，UTC，精确到毫秒
BranchType	是	是	Int64	分支事务类型
Name	是	是	String	分支事务名称
Status	是	是	Int64	分支事务状态
GroupId	是	是	String	事务分组ID
Server	是	是	String	分支事务来源服务标识
Params	是	是	String	分支运行参数，Json格式字符串
TxId	是	是	Int64	主事务ID

GatewayQQMiniProgramLoginPlugin

微服务网关qq登录插件

被如下接口引用：DescribeGatewayQQMiniProgramLoginPlugin

名称	必选	允许NULL	类型	描述
Id	是	否	String	网关插件ID
Name	是	否	String	插件名称
Description	是	是	String	插件描述
Type	是	否	String	插件类型
CreatedTime	是	否	String	插件创建时间 如:2019-06-20 15:51:28
UpdatedTime	是	否	String	插件更新时间 如:2019-06-20 15:51:28
Status	是	否	String	发布状态: drafted/released
QqAppId	是	否	String	QQ小程序AppId
RequestCodeBaggagePosition	是	否	String	QQ小程序请求code携带位置：header/cookie
SessionKeyName	是	否	String	自定义登录态参数名
SessionExpireTime	是	否	Int64	自定义登录态过期时间，单位：秒

名称	必选	允许NULL	类型	描述
RequestSessionBaggagePosition	是	否	String	前台业务请求自定义登录态参数位置：header/cookie
BusinessSessionBaggagePosition	是	否	String	向业务后台传输登录态参数位置：header/query/cookie
ResponseSessionBaggagePosition	是	否	String	返回自定义登录态参数位置：header
MetaDataTagInfoList	是	是	String	元数据转标签配置的Json串
CustomizeTagInfoList	是	是	String	自定义标签配置的Json串

DescribeResourceConfigLicense

DescribeResourceConfig

被如下接口引用：DescribeResourceConfig

名称	必选	允许NULL	类型	描述
Function	是	是	Array of DescribeResourceConfigLicenseFunction	功能
Resource	是	是	Array of DescribeResourceConfigLicenseResource	资源
ExpireTime	是	是	Uint64	utc时间 单位秒
Countdown	是	是	Uint64	utc时间 单位秒
Spec	是	是	String	规格

RouteRuleV2

路由规则项

被如下接口引用：CreateRoute、DescribeRoute、DescribeRoutes、ModifyRoute

名称	必选	允许NULL	类型	描述
RouteRuleId	否	是	String	路由规则项ID
TagList	否	是	Array of RouteTagV2	路由规则项包含TAG列表
DestList	是	否	Array of RouteDestV2	路由规则项包含目标列表
RouteId	否	是	String	路由规则项所属路由规则ID

VportTypeResult

VportTypeResult

被如下接口引用：ListTsfModulesDetail、ModifyModuleConf

名称	必选	允许NULL	类型	描述
VportId	是	是	String	VportId
ModuleId	是	是	String	ModuleId
VportType	是	是	String	VportType
DefaultVport	是	是	String	DefaultVport
Vport	是	是	String	Vport

ScalableSubRule

弹性扩缩容监控指标类型

被如下接口引用：CreateScalableRule、ModifyScalableRule

名称	必选	允许NULL	类型	描述
Indicators	是	否	Uint64	监控指标的阈值。如指标为 CPU 利用率或内存利用率，则为百分比 0-100；如指标为响应时间或请求个数，则为 0-99999999
IndicatorType	是	否	Uint64	规则指标类型: 1: CPU 利用率, 2: 内存利用率, 3: 请求响应时间, 4: 请求个数

TagRouteItemList

标签路由规则

被如下接口引用：CreateRouteRule、DescribeRouteReleaseHistory、DescribeRouteRule、DescribeRouteRules、ModifyRouteRule

名称	必选	允许NULL	类型	描述
TagRouteId	否	否	String	标签路由规则项Id
SourceType	是	否	String	标签路由，标签类型，表示系统标签或自定义标签, 系统标签： S, 自定义标签： U
SourceField	是	否	String	标签路由匹配源字段
SourceMatchRule	是	否	String	标签路由匹配源规则, 等于: EQUAL， 不等于： NOT_EQUAL， 包含： IN， 不包含： NOT_IN， 正则表达式： REGEX
SourceValue	是	否	String	标签路由匹配源取值
TargetField	是	否	String	标签路由匹配目标字段
TargetValue	是	否	String	标签路由匹配目标取值
RouteRuleId	否	否	String	标签路由所属路由Id

TemplatePageResult

TemplatePageResult

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	TotalCount
Content	是	否	Array of ProjectList	Content

AppPkgListV2

用用的包列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	无
Content	是	是	Array of AppPkgInfo	无

ProtocolPort

端口对象

被如下接口引用：CreateContainGroup、DeployContainerGroup、DescribeContainerGroupDeployInfo、DescribeContainerGroupDetail、DescribeDeployGroup、DescribeSingleContainerGroups、FindContainerGroup、GetContainGroupDeployInfo、ModifyContainerGroup、UpdateContainerGroup

名称	必选	允许NULL	类型	描述
Protocol	是	否	String	TCP UDP
Port	是	否	Int64	服务端口
TargetPort	是	否	Int64	容器端口
NodePort	否	是	Int64	主机端口

LaneGroup

泳道部署组

被如下接口引用：CreateLane、CreateLaneGroup、DeleteLaneGroup、DescribeGroupLane、DescribeLane、DescribeLaneGroupExist、DescribeLaneGroups、DescribeLanes、DisableLaneGroupEntrance、EnableLaneGroupEntrance、ModifyGroupLane

名称	必选	允许NULL	类型	描述
LaneGroupId	否	是	String	泳道部署组ID
LaneId	否	是	String	泳道ID
GroupId	是	是	String	部署组ID
GroupName	否	是	String	部署组名
ApplicationId	否	是	String	应用ID
ApplicationName	否	是	String	应用名
NamespaceId	否	是	String	命名空间ID
NamespaceName	否	是	String	命名空间名称
Entrance	是	是	Bool	是否入口应用
CreateTime	否	是	Int64	创建时间
UpdateTime	否	是	Int64	更新时间
ClusterType	否	是	String	集群类型

ApplicationForPage

分页的应用描述信息字段

被如下接口引用：DescribeApplication、DescribeApplications

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationDesc	是	是	String	应用描述
ApplicationType	是	是	String	应用类型
MicroserviceType	是	是	String	微服务类型
ProgLang	是	是	String	编程语言
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
ApplicationResourceType	是	是	String	应用资源类型
ApplicationRuntimeType	是	是	String	应用runtime类型
ApigatewayServiceId	是	是	String	Apigateway的serviceId

名称	必选	允许NULL	类型	描述
ApplicationRemarkName	是	是	String	应用备注名
ServiceConfigList	否	是	Array of ServiceConfig	服务配置信息列表

LicenseDuration

LicenseDuration

被如下接口引用：DescribeLicenseApplications

名称	必选	允许NULL	类型	描述
ActiveTime	否	是	Uint64	开始时间，utc秒数
Period	否	是	Uint64	有效时长，单位秒

RecordResourceTypeNameV2

操作记录资源类型名称

被如下接口引用：

名称	必选	允许NULL	类型	描述
ResourceType	是	是	String	操作记录资源类型
ResourceTypeName	是	是	String	操作记录资源类型名称

LicenseResource

LicenseResource

被如下接口引用：DescribeLicenseApplications

名称	必选	允许NULL	类型	描述
Name	是	否	String	资源名
Quota	是	否	Uint64	配额

MasterApiResult

透传masterApi的response回前端

被如下接口引用：

名称	必选	允许NULL	类型	描述
Data	否	是	String	Data
ErrMsg	否	是	String	ErrMsg

InvocationIndicator

服务调用监控指标

被如下接口引用：DescribeInovcationIndicators

名称	必选	允许NULL	类型	描述
InvocationQuantity	是	是	Int64	总请求数
InvocationSuccessRate	是	是	Float	请求成功率，百分比
InvocationAvgDuration	是	是	Float	请求平均耗时，单位毫秒
InvocationSuccessDistribution	是	是	Array of IndicatorCoord	成功请求数时间分布
InvocationFailedDistribution	是	是	Array of IndicatorCoord	失败请求数时间分布
InvocationStatusDistribution	是	是	Array of IndicatorCoord	状态码分布
InvocationDurationDistribution	是	是	Array of IndicatorCoord	时延分布
InvocationQuantityDistribution	是	是	Array of IndicatorCoord	并发请求次数时间分布

TaskRuleCheckResult

任务触发规则检查结果

被如下接口引用：CheckTaskRule

名称	必选	允许NULL	类型	描述
Success	是	否	Bool	是否校验成功
Msg	否	是	String	校验结果信息
NextFireTimes	否	是	Array of Uint64	预计下三次触发时间点

BusinesLogConfigAssociatedGroupV2

业务日志配置关联部署组信息

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationId	是	是	String	部署组所属应用ID
ApplicationName	是	是	String	部署组所属应用名称
ApplicationType	是	是	String	部署组所属应用类型
NamespaceId	是	是	String	部署组所属命名空间ID
NamespaceName	是	是	String	部署组所属命名空间名称
ClusterId	是	是	String	部署组所属集群ID
ClusterName	是	是	String	部署组所属集群名称
ClusterType	是	是	String	部署组所属集群类型
AssociatedTime	是	是	String	部署组关联日志配置时间

TsfPageApiGroupInfo

ApiGroupInfo翻页结构体

被如下接口引用：DescribeApiGroups、DescribeGroupGateways、DescribeGroupsWithPlugin

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of ApiGroupInfo	API分组信息

LaneRule

泳道规则

被如下接口引用：DescribeLaneRule、DescribeLaneRules

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	泳道规则ID
RuleName	是	是	String	泳道规则名称
Priority	是	是	Int64	优先级
Remark	是	是	String	备注
RuleTagList	是	是	Array of LaneRuleTag	泳道规则标签列表

名称	必选	允许NULL	类型	描述
RuleTagRelationship	是	是	String	泳道规则标签关系
LaneId	是	是	String	泳道ID
Enable	是	是	Bool	开启状态
CreateTime	是	是	Int64	创建时间
UpdateTime	是	是	Int64	更新时间

CommonPkg

公共包信息

被如下接口引用：DescribeCommonPkg

名称	必选	允许NULL	类型	描述
ApplicationName	是	否	String	公共包名
ApplicationId	是	否	String	包所属的应用ID，在微服务网关相关应用中使用
ApplicationDesc	是	否	String	包的描述信息
CallbackAction	是	否	String	回调接口
ServiceType	是	否	String	服务类型

RealtimeLogSet

实时日志查询结果集合

被如下接口引用：SearchOssRealtimeBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog

名称	必选	允许NULL	类型	描述
RealtimeTs	是	否	Uint64	请求基准时间戳（用于下次调用）
BusinessLogSet	是	是	TsfPageBusinessLogV2	业务日志集合
StdoutLogSet	是	是	TsfPageStdoutLogV2	标准输出日志集合
MeshLogSet	是	是	TsfPageMeshLog	Meh日志集合

GraphNodeV2

依赖拓扑图节点

被如下接口引用：GetOssTopologyGraph、GetTopologyGraph

名称	必选	允许NULL	类型	描述
Name	是	是	String	节点服务名
Type	是	是	String	节点服务类型
ReqTotalQty	是	是	Uint64	节点服务总请求量
ReqAvgQty	是	是	Float	节点请求平均数，次每分钟，两位小数
ReqSuccessRate	是	是	Float	节点请求成功率，两位小数
ReqAvgDuration	是	是	Float	节点请求平均耗时，单位毫秒，两位小数
EdgeList	是	是	Array of GraphEdgeV2	节点服务为目的的边列表（入度）
Id	是	是	String	节点标识符
NamespaceId	是	是	String	节点命名空间ID

VmGroupV2

虚拟机部署组信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	分组Id
GroupName	是	是	String	分组名称
GroupStatus	是	是	String	分组状态
PackageId	是	是	String	程序包Id
PackageName	是	是	String	程序包名称
PackageVersion	是	是	String	程序包版本号
ClusterId	是	是	String	集群Id
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间Id
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用Id
ApplicationName	是	是	String	应用名称
InstanceCount	是	是	Int64	分组机器数目

名称	必选	允许NULL	类型	描述
RunInstanceCount	是	是	Int64	分组运行中机器数目
StartupParameters	是	是	String	部署组启动参数信息
CreateTime	是	是	String	分组创建时间
UpdateTime	是	是	String	分组更新时间
OffInstanceCount	是	是	String	分组停止机器数目
GroupDesc	是	是	String	分组描述信息
MicroserviceType	是	是	String	微服务类型

AuthConditionV2

用于 TSF 服务鉴权，表示鉴权条件

被如下接口引用：CreateAuthorization、DescribeAuthorization、DescribeAuthorizationType、DescribeAuthorizations、ModifyAuthorization

名称	必选	允许NULL	类型	描述
TagType	是	是	String	标签类型
TagField	是	是	String	标签名
TagOperator	是	是	String	标签运算符
TagValue	是	是	String	标签值
TagId	否	是	Int64	标签ID

ResourceOperationStatusResult

描述操作执行结果

被如下接口引用：InterruptResourceBatchOperation

名称	必选	允许NULL	类型	描述
OperationResult	是	否	Bool	操作的执行状态

Env

环境变量

被如下接口引用：DeployContainerGroup、DescribeContainerGroupAttribute、DescribeContainerGroupDeployInfo、DescribeContainerGroupDetail、DescribeDeployGroup、FindContainerGroup

名称	必选	允许NULL	类型	描述
Name	是	否	String	环境变量名称
Value	是	否	String	环境变量值

TaskRecordSimpleInfo

任务基本信息

被如下接口引用：DescribeFlowAvailableTasks

名称	必选	允许NULL	类型	描述
TaskId	是	否	String	任务ID
TaskName	是	否	String	任务名称

TsfPageImageInfo

TsfPageImageInfo

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	TotalCount
Content	是	是	Array of ImageInfo	TsfPageImageInfo详情

ListAlarmPoliciesResult

ListAlarmPoliciesResult

被如下接口引用：ListAlarmPolicies

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of AlarmPolicyResult	Content

ListTsfZoneResult

ListTsfZoneResult

被如下接口引用：DescribeTsfmanagerZones、DescribeZones

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of TsfZoneResult	Content

SimpleApplication

简单应用

被如下接口引用：DescribeSimpleApplications

名称	必选	允许NULL	类型	描述
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
ApplicationType	是	是	String	应用类型
MicroserviceType	是	是	String	应用微服务类型
ApplicationDesc	是	是	String	ApplicationDesc
ProgLang	是	是	String	ProgLang
ApplicationResourceType	是	是	String	ApplicationResourceType
CreateTime	是	是	String	CreateTime
UpdateTime	是	是	String	UpdateTime
ApigatewayServiceId	是	是	String	ApigatewayServiceId
ApplicationRuntimeType	是	是	String	ApplicationRuntimeType
ApplicationRemarkName	否	否	String	应用备注

GroupUseStatisticsEntity

API分组日使用统计对象数据点

被如下接口引用：DescribeGroupUseDetail

名称	必选	允许NULL	类型	描述
ApiPath	是	是	String	API 路径
ServiceName	是	是	String	服务名
Value	是	是	String	统计值
ApiId	是	是	String	API ID

TsfPageNamespace

Tsf命名空间分页对象

被如下接口引用：DescribeNamespaces、DescribeSimpleNamespaces

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	命名空间总条数
Content	是	是	Array of Namespace	命名空间列表

DockerForUse

Docker使用指引

被如下接口引用：DescribeDockerForUse

名称	必选	允许NULL	类型	描述
Server	是	否	String	仓库中心地址
Username	是	否	String	用户名

PagedDtsGroup

事务分组分页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of DtsGroup	事务分组列表

PkgInfo

包信息

被如下接口引用：DescribePkgs、ListPkg

名称	必选	允许NULL	类型	描述
PkgId	是	是	String	程序包ID
PkgName	是	是	String	程序包名
PkgType	是	是	String	程序包类型

名称	必选	允许NULL	类型	描述
PkgVersion	是	是	String	程序包版本
PkgDesc	是	是	String	程序包描述
UploadTime	是	是	String	上传时间
Md5	是	是	String	程序包MD5
PkgPubStatus	是	是	Int64	程序包状态
PkgBindInfo	是	是	Array of PkgBind	程序包关联关系

ContainerTasksResult

变更记录任务列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of ContainerTasks	列表信息

RouteV2

路由规则信息

被如下接口引用：DescribeRoute、DescribeRoutes

名称	必选	允许NULL	类型	描述
RouteId	是	是	String	路由规则ID
RouteName	是	是	String	路由规则名称
RouteDesc	是	是	String	路由规则描述信息
MicroserviceId	是	是	String	微服务ID
EnableStatus	是	是	Bool	路由启用状态。true：路由规则为启用状态，。false：路由规则为停用状态
CreateTime	是	是	String	路由规则创建时间
UpdateTime	是	是	String	路由规则更新时间
NamespaceId	是	是	String	微服务所属命名空间ID
MicroserviceName	是	是	String	微服务名称

名称	必选	允许NULL	类型	描述
RuleList	是	是	Array of RouteRuleV2	路由规则包含路由规则项列表

GatewayInstance

网关实体（单元化后使用）

被如下接口引用：DescribeGatewayInstance、DescribeGatewayInstances

名称	必选	允许NULL	类型	描述
Id	是	是	String	网关实体ID
ServiceId	是	是	String	微服务ID（ms-xxx）
ServiceName	是	是	String	微服务名
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名
Type	是	是	String	网关实例类型：单元化(unit)/非单元化网关(none)
ConsulInstanceCount	是	是	Int64	注册中心在线节点数

ExpandSubruleV2

ExpandSubrule

被如下接口引用：

名称	必选	允许NULL	类型	描述
Indicators	是	是	Int64	Indicators
IndicatorType	是	是	Int64	IndicatorType
RuleType	是	是	Int64	RuleType

ScalableTask

ScalableTask

被如下接口引用：ListScalableTasks

名称	必选	允许NULL	类型	描述
Taskid	是	是	String	任务id

名称	必选	允许NULL	类型	描述
Desc	是	是	String	任务描述
CreateTime	是	是	String	CreateTime
EndTime	是	是	String	EndTime
TaskInstanceList	是	是	Array of TaskInstance	TaskInstanceList

ConfigSummaryV2

配置项统计信息，其实就是Config的部分字段

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigName	是	否	String	配置项名称
ConfigVersionCount	是	否	Int64	配置项值
LastUpdateTime	是	否	String	配置项类型
ApplicationId	是	否	String	应用ID
ApplicationName	是	否	String	应用名称

ScalableRuleAttributeV2

ScalableRuleAttribute

被如下接口引用：

名称	必选	允许NULL	类型	描述
RuleId	是	是	String	RuleId
Name	是	是	String	Name
EnableShrink	是	是	Int64	EnableShrink
EnableExpand	是	是	Int64	EnableExpand
ExpandPeriod	是	是	Int64	ExpandPeriod
ShrinkPeriod	是	是	Int64	ShrinkPeriod
ExpandVmCount	是	是	Int64	ExpandVmCount
ShrinkVmCount	是	是	Int64	ShrinkVmCount
CoolTime	是	是	Int64	CoolTime

名称	必选	允许NULL	类型	描述
ExpandVmCountLimit	是	是	Int64	ExpandVmCountLimit
ShrinkVmCountLimit	是	是	Int64	ShrinkVmCountLimit
ExpandSubruleList	是	是	Array of ExpandSubrule	ExpandSubruleList
ShrinkSubruleList	是	是	Array of ExpandSubrule	ShrinkSubruleList
CreateTime	是	是	String	CreateTime
UpdateTime	是	是	String	UpdateTime

VmGroupSimple

虚拟机部署组列表简要字段

被如下接口引用：DescribeGroups

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
ApplicationType	是	是	String	应用类型
GroupDesc	是	是	String	部署组描述
UpdateTime	是	是	String	部署组更新时间
ClusterId	是	是	String	集群ID
StartupParameters	是	是	String	部署组启动参数
NamespaceId	是	是	String	命名空间ID
CreateTime	是	是	String	部署组创建时间
ClusterName	是	是	String	集群名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称
NamespaceName	是	是	String	命名空间名称
MicroserviceType	是	是	String	应用微服务类型
GroupResourceType	是	是	String	部署组资源类型
UpdatedTime	是	是	Int64	部署组更新时间戳
DeployDesc	是	是	String	部署应用描述信息

DeliveryConfigBindGroup

描述投递配置项绑定的部署组

被如下接口引用：DescribeAssociateRelation、DescribeDeliveryConfigs

名称	必选	允许NULL	类型	描述
ConfigId	是	否	String	配置id
ConfigName	是	否	String	配置名
CollectPath	是	是	Array of String	采集路径
Groups	是	是	Array of GroupInfo	关联部署组信息
CreateTime	是	是	String	创建时间

ResourceTaskStatusResult

资源任务转态结果

被如下接口引用：DescribeResourceTaskStatus

名称	必选	允许NULL	类型	描述
TaskStatus	是	是	Uint64	任务的执行状态

CloudMonitorPolicyResult

云监控对象策略对象

被如下接口引用：ListColudMonitorStatisticsPolicy

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	TotalCount
Content	是	是	Array of CloudMonitorPolicies	Content

MeshLog

mesh 组件日志

被如下接口引用：SearchMeshLog、SearchOssRealtimeBusinessLog、SearchRealtimeBusinessLog、SearchRealtimeMeshLog、SearchRealtimeStdoutLog

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID

名称	必选	允许NULL	类型	描述
Content	是	是	String	日志内容
Timestamp	是	是	Uint64	日志时间戳
InstanceIp	是	是	String	实例IP
Type	是	是	Int64	mesh 组件类型（0：pilotagent，1：envoy，2：MeshDns）
GroupId	是	是	String	部署组ID

ServiceConfig

服务配置

被如下接口引用：CreateApplication、DescribeApplication、DescribeApplications、ModifyApplication

名称	必选	允许NULL	类型	描述
Name	是	否	String	服务名
Ports	是	否	Array of Ports	端口信息列表
HealthCheck	否	是	HealthCheckConfig	健康检查配置

ContainerTasksResultV2

变更记录任务列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	总记录数
Content	是	否	Array of ContainerTasks	列表信息

TxMainTransaction

主事务描述

被如下接口引用：DescribeTransactions

名称	必选	允许NULL	类型	描述
TransactionId	是	是	String	事务Id
ServerIp	是	是	String	服务器Ip
ServiceName	是	是	String	服务名称

名称	必选	允许NULL	类型	描述
MethodName	是	是	String	方法名
SubCount	是	是	Int64	子事务数量
StartTime	是	是	String	创建时间
EndTime	是	是	String	更新时间
Status	是	是	Int64	事务状态
TimeoutMs	是	是	Int64	超时时间
RetryTimes	是	是	Int64	重试次数
AutoRetry	是	是	Bool	是否开启自动重试

ConfigReleaseLogV2

配置项发布日志

被如下接口引用：

名称	必选	允许NULL	类型	描述
ConfigReleaseLogId	是	是	String	配置项发布日志ID
ConfigId	是	是	String	配置项ID
ConfigName	是	是	String	配置项名称
ConfigVersion	是	是	String	配置项版本
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ReleaseTime	是	是	String	发布时间
ReleaseDesc	是	是	String	发布描述
ReleaseStatus	是	是	String	发布状态
LastConfigId	是	是	String	上次发布的配置项ID
LastConfigName	是	是	String	上次发布的配置项名称

名称	必选	允许NULL	类型	描述
LastConfigVersion	是	是	String	上次发布的配置项版本
RollbackFlag	是	是	Bool	回滚标识

TsfPageBillingOperationRecord

BillingOperationRecord 翻页对象

被如下接口引用：DescribeBillingOperationRecords

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录总数
Content	是	是	Array of BillingOperationRecord	记录实体列表

GetContainGroupOtherResultV2

部署组其他字段获取

被如下接口引用：

名称	必选	允许NULL	类型	描述
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动实例总数
LbIp	是	否	String	负载均衡ip
ClusterIp	是	否	String	Service ip
Status	是	否	String	服务状态，请参考后面的的状态定义
Message	是	否	String	异常信息,服务状态为Abnormal时才有

PermCat

tsf-privilege 模块 PermissionCategory 权限组

被如下接口引用：DescribePermissionCategories、DescribeRole、DescribeRoles

名称	必选	允许NULL	类型	描述
PermCatId	否	是	String	权限组ID
PermCatName	否	是	String	权限组名称
ResourceId	否	是	String	资源ID

名称	必选	允许NULL	类型	描述
ResourceName	否	是	String	资源名称
CreationTime	否	是	Int64	创建时间
LastUpdateTime	否	是	Int64	最后更新时间
DeleteFlag	否	是	Bool	删除标识
ServiceCode	否	是	String	产品编码
ServiceName	否	是	String	产品名称
PermCatDesc	否	是	String	权限组描述

MonitorStatisticsPolicyV2

MonitorStatisticsPolicy

被如下接口引用：

名称	必选	允许NULL	类型	描述
PolicyId	是	是	String	监控统计策略id
KeyWords	是	是	String	关键词
GroupList	是	是	Array of MonitorStatisticsPolicyGroup	部署组列表信息
NamespaceId	是	是	String	命名空间id
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间

TsfPageMonitorGroup

列表中部署组分页信息，用于云监控

被如下接口引用：DescribeCloudMonitorGroups

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	部署组总数目
Content	是	是	Array of MonitorGroup	部署组列表信息

TsfPageGatewayPlugin

GatewayPlugin 翻页对象

被如下接口引用：DescribeGatewayPlugins、DescribePluginInstances

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录总数
Content	是	是	Array of GatewayPlugin	记录实体列表

GetContainGroupDeployInfo

获取容器部署信息(GetContainGroupDeployInfo) 【部署更新的时候需要获取部署组信息】 输出参数

被如下接口引用：GetContainGroupDeployInfo

名称	必选	允许NULL	类型	描述
GroupId	是	否	String	GroupId
GroupName	是	否	String	分组名称
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动的实例数
Server	是	否	String	镜像server
Reponame	是	否	String	镜像名，如/tsf/nginx
TagName	是	否	String	镜像版本名称
CpuRequest	是	否	String	预分配cpu 核数，如0.2，
CpuLimit	是	否	String	最大分配cpu 核数，如0.6
MemRequest	是	否	String	预分配内存M数
MemLimit	是	否	String	最大分配内存M数
AccessType	是	否	Int64	0:公网 1:集群内访问 2：NodePort
ProtocolPorts	是	否	Array of ProtocolPort	数组
UpdateType	是	否	Int64	更新方式：0:快速更新 1:滚动更新
UpdateIvl	是	否	Int64	更新间隔,单位秒
JvmOpts	是	否	String	jvm参数
ClusterId	是	否	String	无用
ClusterName	是	否	String	无用
ApplicationId	是	否	String	无用
ApplicationName	是	否	String	无用

名称	必选	允许NULL	类型	描述
ApplicationType	是	否	String	无用
KuberneteApiServer	是	否	String	无用
KuberneteUser	是	否	String	无用
KubernetePassword	是	否	String	无用
NamespaceId	是	否	String	无用
NamespaceName	是	否	String	无用
GroupComment	是	否	String	无用
PodId	是	否	String	无用
PodName	是	否	String	无用
ClusterIp	是	否	String	无用
LbIp	是	否	String	无用
NodePort	是	否	String	无用
IsStop	是	否	String	无用
Status	是	否	String	无用
Message	是	否	String	无用
ChangType	是	否	Int64	无用
ChangNum	是	否	Int64	无用
IsFirst	是	否	Int64	无用
CreateTime	是	否	String	无用
UpdateTime	是	否	String	无用
AppId	是	否	String	无用
SubAccountUin	是	否	String	无用
Uin	是	否	String	无用
Limit	是	否	Int64	无用
OrderType	是	否	String	无用

ImageFeature

镜像特征

被如下接口引用：DescribeImageFeatures

名称	必选	允许NULL	类型	描述
FeatureId	是	是	String	特征ID
FeatureType	是	是	String	特征类型
Name	是	是	String	名称
Version	是	是	String	版本

ImageInfo

镜像信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
ImageId	是	是	String	ImageId
OsName	是	是	String	OsName
ImageSize	是	是	Int64	ImageSize
ImageType	是	是	String	ImageType
ImageName	是	是	String	ImageName
ImageDesc	是	是	String	ImageDesc
CreateTime	是	是	String	CreateTime
LoginName	是	是	String	LoginName

GatewayPluginBoundParam

微服务网关插件绑定对象

被如下接口引用：BindPlugin、UnbindPlugin

名称	必选	允许NULL	类型	描述
PluginId	是	否	String	插件id
ScopeType	是	否	String	插件绑定到的对象类型:group/api
ScopeValue	是	否	String	插件绑定到的对象主键值，例如分组的ID/API的ID

TsfPageSimpleApplicationV2

TSF分页简单应用对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Int64	总条数
Content	是	是	Array of SimpleApplication	简单应用列表

ModulesResult

ModulesResult

被如下接口引用：ListTsfModulesDetail

名称	必选	允许NULL	类型	描述
ModuleId	是	是	String	ModuleId值
ModuleName	是	是	String	ModuleName值
ModuleType	是	是	String	ModuleType值
ExternalVip	是	是	String	ExternalVip值
InternalVip	是	是	String	InternalVip值
Vports	是	是	Array of VportTypeResult	Vports值
Instances	是	是	Array of InstancesResult	Instances值
AlarmType	是	是	String	AlarmType值
ModuleCommon	是	是	Int64	ModuleCommon值
DeployOrder	是	是	Int64	DeployOrder值
Visible	是	是	String	Visible值
ViportVisible	是	是	Bool	ViportVisible值
HasRole	是	是	Bool	HasRole值
Password	是	是	String	Password值
HasRoleStr	是	是	String	HasRoleStr值

TsfPageBillingLicense

BillingLicense 翻页对象

被如下接口引用：DescribeAllBillingLicenses

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录总数
Content	是	是	Array of BillingLicense	记录实体列表

AppPkgList

应用的包列表

被如下接口引用：ListAppPkg

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Int64	程序包总量
Content	是	是	Array of AppPkgInfo	应用的包列表

ClusterV3

集群详情

被如下接口引用：

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
ClusterDesc	是	是	String	集群描述
ClusterType	是	是	String	集群类型
VpcId	是	是	String	集群所属私有网络ID
ClusterStatus	是	是	String	集群状态
ClusterCIDR	是	是	String	集群CIDR
ClusterTotalCpu	是	是	Float	集群总CPU，单位: 核
ClusterTotalMem	是	是	Float	集群总内存，单位: G
ClusterUsedCpu	是	是	Float	集群已使用CPU，单位: 核
ClusterUsedMem	是	是	Float	集群已使用内存，单位: G
InstanceCount	是	是	Int64	集群机器实例数量
RunInstanceCount	是	是	Int64	集群运行中的机器实例数量
NormalInstanceCount	是	是	Int64	集群正常状态的机器实例数量

名称	必选	允许NULL	类型	描述
DeleteFlag	是	是	Bool	删除标记：true：可以删除；false：不可删除
CreateTime	是	是	String	创建时间
UpdateTime	是	是	String	更新时间
TsfRegionId	是	是	String	集群所属TSF地域ID
TsfRegionName	是	是	String	集群所属TSF地域名称
TsfZoneId	是	是	String	集群所属TSF可用区ID
TsfZoneName	是	是	String	集群所属TSF可用区名称

LaneRuleTag

泳道规则标签

被如下接口引用：CreateLaneRule、DescribeLaneRule、DescribeLaneRules、ModifyLaneRule

名称	必选	允许NULL	类型	描述
TagId	是	是	String	标签ID
TagName	是	是	String	标签名
TagOperator	是	是	String	标签操作符
TagValue	是	是	String	标签值
LaneRuleId	是	是	String	泳道规则ID
CreateTime	是	是	Int64	创建时间
UpdateTime	是	是	Int64	更新时间

PurchaseInfo

购买信息

被如下接口引用：DescribeUserPurchaseInfo

名称	必选	允许NULL	类型	描述
CanPurchase	是	否	Bool	true 表示可以新购，false 时能够续费
Spec	是	是	String	规格
NodeSize	是	是	Int64	节点数
ExpireTime	是	是	Int64	过期时间 utc时间 单位秒

名称	必选	允许NULL	类型	描述
ResourceId	是	是	String	资源ID
Label	是	是	String	询价参数 key
RenewFlag	是	是	Int64	自动续费标记，0 默认，1 自动续费，2 不自动续费
CanModify	是	是	Bool	CanPurchase 为 false 时，用于判断能否变配
LabelValue	是	是	Int64	询价参数 value

TsfMiscroserviceApi

TsfMiscroserviceApi

被如下接口引用：

名称	必选	允许NULL	类型	描述
Id	是	是	String	微服务ID
Path	是	是	String	路径
Method	是	否	String	方法
PkgVersion	是	否	String	包版本
ApplicationId	是	否	String	应用ID
MicroserviceId	是	否	String	微服务ID
NamespaceId	是	否	String	命名空间ID
MicroserviceName	是	否	String	微服务名称
Description	是	否	String	描述

MicroserviceExtraV2

微服务（信息更丰富版）

被如下接口引用：

名称	必选	允许NULL	类型	描述
MicroserviceId	否	是	String	微服务 ID
MicroserviceName	否	是	String	名称
MicroserviceDesc	否	是	String	描述
MicroserviceType	否	是	String	类型

名称	必选	允许NULL	类型	描述
AssociateApplicationId	否	是	String	关联的应用 ID
MicroserviceProtocol	否	是	String	协议
MicroserviceListeningPort	否	是	Int64	监听端口
HealthCheckUrl	否	是	String	健康检查参数
ServiceClustertype	否	是	String	集群类型
CreateTime	否	是	Int64	创建时间
UpdateTime	否	是	Int64	更新时间
NamespaceId	否	是	String	命名空间 ID
NamespaceName	否	是	String	命名空间名称
ClusterId	否	是	String	集群 ID
ClusterName	否	是	String	集群名称

VmGroup

虚拟机部署组信息

被如下接口引用：DescribeGroup

名称	必选	允许NULL	类型	描述
GroupId	是	是	String	部署组ID
GroupName	是	是	String	部署组名称
GroupStatus	是	是	String	部署组状态
PackageId	是	是	String	程序包ID
PackageName	是	是	String	程序包名称
PackageVersion	是	是	String	程序包版本号
ClusterId	是	是	String	集群ID
ClusterName	是	是	String	集群名称
NamespaceId	是	是	String	命名空间ID
NamespaceName	是	是	String	命名空间名称
ApplicationId	是	是	String	应用ID
ApplicationName	是	是	String	应用名称

名称	必选	允许NULL	类型	描述
InstanceCount	是	是	Int64	部署组机器数目
RunInstanceCount	是	是	Int64	部署组运行中机器数目
StartupParameters	是	是	String	部署组启动参数信息
CreateTime	是	是	String	部署组创建时间
UpdateTime	是	是	String	部署组更新时间
OffInstanceCount	是	是	Int64	部署组停止机器数目
GroupDesc	是	是	String	部署组描述信息
MicroserviceType	是	是	String	微服务类型
ApplicationType	是	是	String	应用类型
GroupResourceType	是	是	String	部署组资源类型
UpdatedTime	是	是	Int64	部署组更新时间戳
DeployDesc	是	是	String	部署应用描述信息
UpdateType	是	是	Uint64	滚动发布的更新方式
DeployBetaEnable	是	是	Bool	发布是否启用beta批次
DeployBatch	是	是	Array of Float	滚动发布的批次比例列表
DeployExeMode	是	是	String	滚动发布的批次执行方式
DeployWaitTime	是	是	Uint64	滚动发布的每个批次的等待时间
EnableHealthCheck	是	是	Bool	是否开启了健康检查
HealthCheckSettings	是	是	HealthCheckSettings	健康检查配置
PackageType	是	是	String	程序包类型
StartScript	是	是	String	启动脚本 base64编码
StopScript	是	是	String	停止脚本 base64编码

CosDownloadInfo

Cos下载所需信息

被如下接口引用：DescribeDownloadInfo、GetDownloadInfo

名称	必选	允许NULL	类型	描述
Bucket	是	是	String	桶名称

名称	必选	允许NULL	类型	描述
Region	是	是	String	地域
Path	是	是	String	路径
Credentials	是	是	CosCredentials	鉴权信息

ContainerGroupOther

部署组列表-其它字段

被如下接口引用：DescribeContainerGroupAttribute

名称	必选	允许NULL	类型	描述
InstanceNum	是	否	Int64	实例总数
CurrentNum	是	否	Int64	已启动实例总数
LbIp	是	否	String	负载均衡ip
ClusterIp	是	否	String	Service ip
Status	是	否	String	服务状态，请参考后面的的状态定义
Message	是	否	String	服务状态，请参考后面的的状态定义
Envs	是	否	Array of Env	环境变量
NodePort	是	是	Uint64	Service NodePort
SubnetId	是	是	String	子网ID
HealthCheckSettings	是	是	Array of HealthCheckSetting	健康检查相关字段
IsNotEqualServiceConfig	是	是	Bool	服务配置信息是否匹配

TsfPageGatewayPluginInstance

GatewayPluginInstance 翻页对象

被如下接口引用：

名称	必选	允许NULL	类型	描述
TotalCount	是	否	Uint64	记录总数
Content	是	否	Array of GatewayPluginInstance	记录实体列表

ZipkinBinaryAnnotation

调用链Span二进制注解

被如下接口引用：GetTraceSpans

名称	必选	允许NULL	类型	描述
Key	是	是	String	注解键
Value	是	是	String	注解值
Endpoint	是	是	ZipkinEndPoint	注解端点信息

OverviewHostResourceUsage

TSF实例相关基本信息概览

被如下接口引用：DescribeHostResourceUsage

名称	必选	允许NULL	类型	描述
RunHostCount	是	是	Int64	运行云主机数
HostCount	是	是	Int64	云主机总数

错误码

最近更新时间: 2025-02-18 17:50:46

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
"Response": {
"Error": {
"Code": "AuthFailure.SignatureFailure",
"Message": "The provided credentials could not be validated. Please check your signature is correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。 请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。 签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

错误码	说明
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
MissingParameter.SourceServiceListRequired	
ResourceNotFound.NamespaceNotExist	
ResourceNotFound.ErrNoUser	
InvalidParameterValue.LaneRuleNameNotEmpty	
MissingParameter.TaskParameterMissed	
InvalidParameter.TsfApmBusiLogCfgIdParamError	
ResourceInUse.GatewayActionForbid	
InvalidParameterValue.ApplicationTypeInvalid	
ResourceNotFound.CvmcaeMasterResourceNotFound	
InvalidParameterValue.CircuitBreakerInvalidSlowCallDurationThreshold	
FailedOperation.TsfApmBusiLogCfgWriteError	
ResourceNotFound.GroupNotExist	
InvalidParameterValue.AuthTypeInvalid	

错误码	说明
InvalidParameterValue.ContainergroupInvalidMemInfo	
FailedOperation.TsfApmEsClientRequestError	
InvalidParameterValue.LaneInfoAlreadyUsed	
InvalidParameter.TsfApmBusiLogSearchRequestParamError	
MissingParameter.ConfigIdRequired	
InvalidParameterValue.SidecarFilterNameRepeated	
ResourceNotFound.InstanceNotExist	
InvalidParameterValue.ConfigExists	
ResourceInUse.GroupCannotDelete	
MissingParameter.TsfApmJvmMonitorRequestParamMissing	
ResourceNotFound.ApplicationNotExist	
InvalidParameterValue.CircuitBreakerExist	
InvalidParameter.ClusterAlreadySynchronized	
ResourceInUse.RouteCannotDelete	
InvalidParameterValue.RunApiParameterTooLong	
InvalidParameterValue.ServiceNotExists	
InvalidParameterValue.GroupNotExists	
InvalidParameterValue.ConfigGroupApplicationIdNotMatch	
ResourceNotFound.ContainergroupClusterNotfound	
ResourceInsufficient.InstanceExcessLimit	
InvalidParameterValue.ContainergroupMemlimitOver	
InvalidParameter.TsfApmTraceSearchRequestParamError	
InvalidParameterValue.ClusterTypeInvalid	
InvalidParameterValue.ConfigValueTooLong	
InvalidParameterValue.ApplicationPageLimitInvalid	
InternalError.TsfApmAgentUnknownError	
InternalError.ContainergroupKuberneteNodePortRepeat	
InvalidParameterValue.DuplicateRoleName	
InvalidParameterValue.ContainergroupLimitValueInvalid	

错误码	说明
InvalidParameterValue.InstanceRequestValueInvalid	
FailedOperation.QueryServiceNodesFailed	
InvalidParameter.LaneRuleInfoNotExist	
InvalidParameterValue.CvmCaeMasterTaskInEnd	
MissingParameter.TsfAsApplicationidNull	
InternalError.ApplicationMasterNuknownError	
FailedOperation.GroupExists	
InvalidParameterValue.ContainergroupYamlNull	
FailedOperation.TsfApmAgentVersionNotMatch	
InvalidParameterValue.GroupNameRegxMismatch	
InternalError.ContainergroupCloudapiInvokeError	
FailedOperation.TsfResourceStatusEsParseResponseFail	
InvalidParameterValue.ContainergroupApplicationIdNull	
InvalidParameterValue.CvmCaeMasterTaskBatchConfigNotExist	
MissingParameter.RequiredParameterMissing	
MissingParameter.ConfigReleaseIdRequired	
InvalidParameterValue.NamespaceNameInvalid	
FailedOperation.TsfApmCtsdbClientRequestError	
FailedOperation.NamespaceCreateFailed	
InvalidParameter.LaneRuleTagValueTotalTooLong	
InvalidParameterValue.LaneInfoNameAlreadyUsed	
InvalidParameterValue.FileConfigNameInvalid	
InternalError.TsfApmTooManyRequests	
InternalError.GatewayDbError	
InvalidParameterValue.ContainergroupReponameInvalid	
InternalError.ApplicationMasterFeignError	
InvalidParameterValue.GatewayParameterInvalid	
InvalidParameterValue.CircuitBreakerInvalidSlowCallRateThreshold	
InternalError.ConsulServerError	

错误码	说明
ResourceNotFound.LicenseServerNotFound	
InternalError.TsfAsDbDeleteFail	
FailedOperation.CircuitBreakerNotSupportMaxEjectionPercent	
InvalidParameter.TsfApmConfigPathNotAllowed	
InvalidParameterValue.GroupPkgNull	
MissingParameter.FileConfigFileValueRequired	
ResourceNotFound.MicroserviceNotExists	
InvalidParameterValue.ClusterCidrConflict	
FailedOperation.TaskCreateError	
InvalidParameterValue.ContainergroupYamlUserContainerNotFound	
InternalError.TsfAsMasterModifyFail	
InternalError.ContainergroupKuberneteApiInvokeError	
UnauthorizedOperation.NamespaceCodeCannotModify	
InternalError.GatewayConsulError	
InvalidParameterValue.RouteValueInvalid	
ResourceNotFound.TsfAsGroupNotExist	
InvalidParameter.ParamError	
InvalidParameterValue.InvalidParameterFormat	
InternalError.TsfApmTraceSearchParseJsonRspError	
InternalError.CvmCaeMasterInternalError	
InvalidParameter.CvmCaeMasterJsonDecodeFail	
InternalError.CvmCaeMasterQueryError	
InvalidParameterValue.GroupNameExist	
InvalidParameter.UpperDeleteLimit	
InvalidParameterValue.RouteDisableInvalid	
InternalError.KubernetesCallError	
FailedOperation.TsfApmAgentJsonParseFailed	
FailedOperation.TaskOperationFailed	
InvalidParameterValue.DuplicateGroupName	

错误码	说明
InternalError.GatewayCommonError	
InvalidParameterValue.CircuitBreakerInvalidMsName	
FailedOperation.CvmCaeMasterHealthCheckConfigError	
InvalidParameterValue.CircuitBreakerInvalidMaxEjectionPercent	
InternalError.CamRoleResponseError	
InvalidParameterValue.LaneRuleNameInvalid	
InvalidParameterValue.SidecarFilterInvalidLuaValueLength	
ResourceNotFound.ServiceApiNotExists	
InvalidParameterValue.ClusterNameLength	
UnsupportedOperation.TaskNotSupported	
FailedOperation.ApplicationQueryFailed	
FailedOperation.ContainergroupGroupHasstop	
InvalidParameterValue.CvmCaeMasterAgentNotFound	
UnauthorizedOperation.CamTsfRoleNotExist	
InvalidParameterValue.GroupNotExistsOrPermissionDenied	
InvalidParameter.LaneRuleNameAlreadyUsed	
InternalError.CamRoleRequestError	
InvalidParameterValue.CircuitBreakerInvalidWaitDurationInOpenState	
InternalError.RouteMeshEnableFailed	
InternalError.ClusterCommonError	
InvalidParameterValue.AuthTypeNonuniformity	
InvalidParameterValue.ProgramNotExists	
InvalidParameterValue.AuthorizationNotExist	
ResourceInUse.GroupInOperation	
ResourceNotFound.ContainergroupGroupNotFound	
InvalidParameterValue.TsfAsGroupNotExist	
InvalidParameterValue.ImagerepoServerNull	
InvalidParameterValue.WrongDontStartValue	
FailedOperation.InstanceUpdateFailed	

错误码	说明
InvalidParameterValue.NamespaceCodeExists	
MissingParameter.ProjectIdIsNull	
InvalidParameterValue.ContainergroupInvalidCpuInfo	
FailedOperation.ServiceQueryFailed	
InvalidParameter.TsfOperationInvalidParam	
InvalidParameterValue.RouteFieldInvalid	
FailedOperation.RouteNamespaceRequestError	
InvalidParameterValue.LaneInfoNotExistEntrance	
InvalidParameterValue.CircuitBreakerInvalidTargetNamespaceId	
InternalError.RemoteServiceCallError	
InvalidParameterValue.SidecarFilterNameLength	
FailedOperation.LaneRuleEnableConsulFailed	
ResourceNotFound.RouteNotExist	
ResourceNotFound.InterfaceNotFound	
InternalError.TsfAsMasterDelFail	
InvalidParameterValue.MsParameterInvalid	
InvalidParameterValue	
InvalidParameterValue.SidecarFilterDescLength	
MissingParameter.BasePackageRequied	
ResourceInUse.NonDefaultNamespaceExists	
InternalError.TaskCommonError	
FailedOperation.ApiMetaParseFailed	
FailedOperation.TsfTxServerFailed	
InvalidParameterValue.ContainergroupNodePortInvalid	
InvalidParameterValue.CvmCaeMasterTaskNotExist	
FailedOperation.ServiceInsertFailed	
MissingParameter.ConfigTemplateNameRequired	
FailedOperation.CircuitBreakerDisableConsulFailed	
FailedOperation.InstanceResetError	

错误码	说明
ResourceNotFound.ClusterVpcNotExist	
InternalError.EndpointIsNullError	
InvalidParameterValue.InvalidNamespaceCode	
InternalError.CloudServiceCallError	
InvalidParameterValue.CvmCaeMasterGroupNoAgent	
InvalidParameterValue.LaneInfoNameInvalid	
InternalError.TsfAsDbQueryFail	
InvalidParameterValue.LaneInfoRemarkTooLong	
InvalidParameterValue.CvmCaeMasterTaskBatchInHealthCheck	
InvalidParameterValue.ConfigTemplateNameInvalid	
ResourceNotFound.TsfApmAgentNoFile	
ResourceNotFound.RatelimitRuleNotExistError	
InvalidParameterValue.InvalidRoleName	
InternalError.TsfApmEsResponseStatusException	
MissingParameter.ConfigVersionRequired	
InternalError.ResourceServerFail	
InvalidParameterValue.ContainergroupPortsRepeat	
InvalidParameter.BadRequest	
InternalError.ContainergroupKuberneteConnectError	
InvalidParameterValue.CircuitBreakerInvalidTargetMsName	
InvalidParameterValue.ContainergroupPortInvalid	
UnauthorizedOperation.LicenseInactive	
InternalError.TsfApmBusiLogCfgAppRelationLogicError	
ResourceNotFound.TsfAsRuleNotExist	
MissingParameter.TsfAsNameNull	
InvalidParameter.LaneRuleNameTooLong	
InvalidParameterValue.ImagerepoTagnameNull	
InvalidParameter.TsfApmBusiLogCfgPathParamError	
InvalidParameterValue.InvalidParameter	

错误码	说明
InvalidParameterValue.GroupNameLength	
InvalidParameterValue.TsfAsIndicatorsError	
InvalidParameterValue.SidecarFilterInvalidLuaValueFormat	
InvalidParameterValue.TsfAsVmcountError	
ResourceInUse.DefaultNamepsaceCannotBeDeleted	
InvalidParameterValue.ContainergroupNamespaceIdNull	
InvalidParameterValue.FileConfigFileNameInvalid	
InvalidParameter.LaneRuleTagNameNotEmpty	
InvalidParameterValue.TemplateErrors	
ResourceInUse.TsfAsIndicatortypeRepeat	
InvalidParameterValue.ProgramNotExist	
InvalidParameter.LaneInfoNameNotEmpty	
ResourceInUse.GroupExists	
InvalidParameterValue.NamespaceNameExist	
ResourceNotFound.ServiceNotExist	
LimitExceeded.TkeClusterNumberExceedLimit	
InternalError.TsfCmonitorBaradApiInvokeError	
InvalidParameterValue.GroupClusterTypeMismatch	
InvalidParameter.CvmCaeMasterAgentStatusError	
InvalidParameterValue.TsfAsApplicationNotExist	
InvalidParameterValue.ConfigNotExists	
ResourceNotFound.ContainergroupPodNotfound	
UnsupportedOperation.VmClusterNotSupport	
FailedOperation.NamespaceQueryFailed	
UnauthorizedOperation.CamTsfRoleNoPermission	
InvalidParameterValue.InvalidLogStatusForRollback	
FailedOperation.ClusterQueryFailed	
FailedOperation.TsfAsExpandIndicatorsLessShrink	
InternalError.KubernetesApiCreateSecretError	

错误码	说明
MissingParameter.ReuqsetIsNull	
InvalidParameterValue.RouteNameExist	
InvalidParameter.ErrRepoExist	
MissingParameter.SourceServiceRequired	
FailedOperation.GatewayRemoteCallError	
UnsupportedOperation.UnsupportAction	
MissingParameter.ApplicationIdRequired	
InvalidParameterValue.GroupClusterNamespaceNotBound	
InvalidParameter.LaneRuleTagNameTooLong	
FailedOperation.ConfigTemplateSearchListFailed	
InvalidParameter.TsfApmBusiLogCfgAppRelationParamError	
InvalidParameterValue.ClusterZoneInvalid	
InvalidParameterValue.DeployGroupNotExists	
InvalidParameterValue.RouteItemTagTypeInvalid	
FailedOperation.ConfigApplicationQueryFailed	
FailedOperation.TsfAsExpandLimitLessShrinkLimit	
InvalidParameterValue.ContainergroupOrderbyInvalid	
InvalidParameterValue.ContainergroupProtocolPortsNull	
InvalidParameterValue.InvalidProgramName	
InvalidParameterValue.LastConfigNotExists	
InvalidParameterValue.ConfigTemplateNameTooLong	
InvalidParameterValue.LaneRuleTagValueTooLong	
FailedOperation.RouteEnableConsulFailed	
InvalidParameterValue.TsfAsPeriodError	
FailedOperation.CircuitBreakerContainRepeatApi	
InternalError.TsfApmBusiLogCfgUpdateTargetNotExistsError	
FailedOperation.TsfAsResourceServerError	
FailedOperation.LaneInfoGroupNotEmpty	
ResourceNotFound.MicroserviceOffline	

错误码	说明
InvalidParameterValue.CvmCaeMasterTaskBatchAlreadyRun	
InvalidParameterValue.RecordPageLimitInvalid	
InvalidParameterValue.AuthRuleNameExistError	
InvalidParameterValue.LaneRuleTagNotEmpty	
InvalidParameter.LaneInfoAlreadyUsed	
FailedOperation.OperationDeleteFailed	
ResourceNotFound.MicroserviceNotExist	
InvalidParameterValue.RouteDestWeightInvalid	
InvalidParameterValue.CvmCaeMasterGroupNoPkg	
InvalidParameterValue.TsfAsIndicatortypeError	
InvalidParameterValue.CvmCaeMasterTaskBatchInHealthChcek	
InvalidParameterValue.ApplicationDescLength	
InvalidParameterValue.GroupValidInstanceNull	
InvalidParameterValue.TsfAsRuleNotExist	
InvalidParameterValue.ClusterNameRequired	
MissingParameter.GatewayParameterRequired	
UnauthorizedOperation.GatewayTooManyRequestParameter	
InternalError.TsfBillingCommonError	
InvalidParameterValue.GroupPageLimitInvalid	
ResourceNotFound.GroupApplicationNotExist	
InvalidParameterValue.SourceServiceAlreadyExist	
MissingParameter.TsfMonitorRequestParamMissing	
InvalidParameterValue.ContainergroupGroupnameRegexMatchFalse	
ResourceNotFound.TsfAsApplicaionNotExist	
FailedOperation.TkeClusterQueryFailed	
InvalidParameter.TsfApmStatsSearchRequestParamError	
MissingParameter.ClusterCidrRequired	
MissingParameter.TsfAsGroupidNull	
MissingParameter.ServiceIdRequired	

错误码	说明
InvalidParameterValue.ServiceDescLength	
ResourceNotFound.ClusterNotExist	
InternalError.TaskInternalError	
InternalError.TsfApmJsonDeserializationFailed	
InternalError.TsfAsMasterAddFail	
ResourceUnavailable.CcsResponseResolveFaild	
InvalidParameterValue.FileConfigExists	
InvalidParameterValue.ConfigAlreadyReleased	
InvalidParameterValue.ConfigNotExistsOrPermissionDenied	
InvalidParameterValue.GroupApplicationTypeMismatch	
InvalidParameterValue.FileConfigReleaseNotExists	
InvalidParameterValue.ServiceNameRepeated	
InvalidParameterValue.ContainergroupTargetportNull	
InvalidParameter.LaneRuleNameInvalid	
UnauthorizedOperation.NoPrivilege	
UnsupportedOperation.TsfApmUnknownInstanceStatus	
InvalidParameter.CvmCaeMasterTaskOperationNotSupport	
FailedOperation.TsfMonitorWaitedTimeout	
MissingParameter.ClusterSubnetRequired	
InvalidParameter.CvmCaeMasterUnknownInstanceStatus	
InvalidParameterValue.PathInvalid	
UnsupportedOperation.NamespaceCodeCannotModify	
InvalidParameterValue.ContainergroupTargetPortsRepeat	
InvalidParameterValue.ContainergroupResourceIstioValueInvalid	
InvalidParameterValue.TaskApplicationTypeMismatch	
MissingParameter.ApplicationIdNull	
FailedOperation.AuthorizationInsertFailed	
InvalidParameter.RepoPackageParamError	
InvalidParameter.LaneRuleTagNotEmpty	

错误码	说明
InvalidParameter.LaneInfoNameTooLong	
InvalidParameterValue.TsfAsStatusError	
MissingParameter.TargetServiceRequired	
FailedOperation.ConfigNamespaceQueryFailed	
FailedOperation.TsfAsGroupRelateRule	
InvalidParameterValue.ImagerepoReponameNull	
FailedOperation.ConfigTemplateDeleteFailed	
ResourceInsufficient.PackageSpaceFull	
FailedOperation.CircuitBreakerEnableConsulFailed	
InternalError.TsfApmAgentUnknownStatus	
FailedOperation.RatelimitConsulError	
InternalError.GroupMasterFeignError	
MissingParameter.ProjectNameRequied	
InvalidParameterValue.ClusterDescLength	
UnauthorizedOperation.NoLicense	
InvalidParameterValue.ContainergroupUpdateivlInvalid	
InvalidParameter.TsfApmJvmMonitorRequestParamIllegal	
InvalidParameterValue.CircuitBreakerInvalidFailRateThreshold	
InvalidParameterValue.ConfigTemplateTypeInvalid	
FailedOperation.TaskDeleteError	
InternalError.GroupCommonError	
UnauthorizedOperation.LicenseUnauthorized	
FailedOperation.ConfigTemplateImportFailed	
InvalidParameterValue.CircuitBreakerInvalidMinNumberOfCalls	
MissingParameter.TsfAsRuleIdNull	
ResourceInUse.InstanceExists	
LimitExceeded.ErrNamespaceMaxLimit	
InvalidParameterValue.ApplicationNameRegxInvalid	
InternalError.ApplicationScalableInitError	

错误码	说明
FailedOperation.AuthorizationQueryFailed	
InvalidParameterValue.SidecarFilterInvalidFilterName	
InvalidParameterValue.FileConfigNotExistsOrPermissionDenied	
InternalError.NamespaceMasterFeignError	
InvalidParameterValue.ProgramItemNotExists	
InternalError.TsfMonitorEsData	
InvalidParameterValue.ApplicationMicroTypeInvalid	
InternalError.TsfApmApaasExecuteError	
InvalidParameter.ApplicationDeleteFailed	
InvalidParameter.LaneRuleNotExist	
InvalidParameterValue.GroupStatusInvalid	
InvalidParameterValue.TsfApmTopologyQueryParseReqDateFormatError	
InvalidParameterValue.ContainergroupCpulimitOver	
MissingParameter.ConfigTemplateTypeRequired	
InvalidParameterValue.FileConfigVersionDescInvalid	
ResourceNotFound.TkeClusterNotExists	
FailedOperation.TkeClusterCreateFailed	
MissingParameter.AuthTypeRequired	
FailedOperation.TsfApmBusiLogCfgSchemaQueryError	
InvalidParameterValue.ConfigVersionDescInvalid	
InvalidParameter.OperationError	
InvalidParameterValue.TsfApmBusiLogCfgDuplicateNameError	
InvalidParameterValue.CircuitBreakerStrategyNotEmpty	
InvalidParameterValue.LaneInfoNotExist	
FailedOperation.TaskPushError	
FailedOperation.RunApiError	
FailedOperation.ConfigTemplateCreateFailed	
ResourceInUse.ApplicationCannotDelete	
FailedOperation.LaneInfoReleaseConsulFailed	

错误码	说明
LimitExceeded.TsfApmBusiLogCfgExceedTwelveError	
InternalError.RouteMeshApiServiceError	
InternalError.EsSearchError	
ResourceNotFound.GroupNamespaceNotExist	
InvalidParameter.TsfMonitorRequestParamIllegal	
InvalidParameterValue.TaskParameterInvalid	
InternalError.ClusterMasterFeignError	
InternalError.TsfApmBusiLogCfgDeleteBindedCfgError	
UnsupportedOperation.TsfApmAgentVersionNotMatch	
InternalError.GroupMasterNuknownError	
ResourceInUse.NamespaceCannotDelete	
InternalError.TsfOperationAppServerError	
InvalidParameterValue.ConfigVersionInvalid	
MissingParameter.ConfigTypeRequired	
InvalidParameter.TsfAsJsonFormatError	
ResourceInsufficient.ResourceLeakIp	
InvalidParameterValue.ReleasedConfigCanNotBeDeleted	
InvalidParameterValue.TsfAsIndicatortypeRepeat	
FailedOperation.TsfApmWaitedTimeout	
InvalidParameterValue.LaneInfoNameTooLong	
InvalidParameterValue.CircuitBreakerInvalidNamespaceId	
ResourceNotFound.TaskNotFound	
InvalidParameterValue.SidecarFilterIsNotExist	
InvalidParameterValue.RecordDateformatInvalid	
InternalError.GatewayConsistencyError	
FailedOperation.TsfApmBusiLogCfgSchemaWriteError	
MissingParameter.TsfAsEnableAllNull	
InvalidParameter.TsfApmBusiLogCfgSchemaParamError	
InvalidParameter.ErrNSMisMatch	

错误码	说明
InternalError.ContainergroupSqlFailed	
InvalidParameterValue.CircuitBreakerInvalidSlidingWindowSize	
InternalError.TemplateErrors	
UnknownParameter.TsfApmAgentUnknownTaskId	
FailedOperation.ContainergroupGroupHasrun	
InternalError.TsfApmDateParseFailed	
FailedOperation.InstanceResetTimeout	
FailedOperation.AuthReleaseFailed	
InvalidParameterValue.ContainergroupUpdatetypeInvalid	
InvalidParameterValue.ImagerepoReponameInvalid	
InvalidParameterValue.SidecarFilterInvalidLuaValue	
InvalidParameterValue.TsfApmBusiLogCfgPathVerifyError	
MissingParameter.NamespaceIdRequired	
InvalidParameterValue.ClusterRegionInvalid	
InternalError.NamespaceMasterUnknownError	
InvalidParameter.ModifyCheckError	
InternalError.TsfApmTsfAgentCallApmAgentFailed	
InvalidParameterValue.GroupNameNull	
InternalError.TsfApmBusiLogCfgAppRelationMasterError	
InternalError.TsfApmInternalError	
InternalError.TsfAsDbUpdateFail	
InvalidParameterValue.ApplicationIdNull	
MissingParameter.ClusterCidrConflict	
MissingParameter.ClusterIdRequired	
ResourceInUse.TsfAsNameRepeat	
InvalidParameter.TsfAnalystParamError	
UnsupportedOperation.GatewayTooManyRequestParameter	
InvalidParameter.LaneInfoNameInvalid	
InvalidParameterValue.ApplicationNameExist	

错误码	说明
ResourceNotFound.TsfApmNamespaceUnavailable	
ResourceNotFound.ContainergroupGroupNamespaceClusterNotFound	
InternalError.ApplicationRepoDeletePkg	
InternalError.NotifyConsulError	
InternalError.TsfAsDbInsertFail	
InvalidParameterValue.DuplicatePolicyItemRequired	
MissingParameter.LastConfigIdRequired	
InvalidParameterValue.NamespaceNotExists	
ResourceNotFound.GroupDeployPkgNotExist	
InvalidParameter.TsfAsCooltimeError	
InternalError.CloudApiProxyError	
InvalidParameter.LaneRuleRemarkTooLong	
InvalidParameter.KubernetesParamError	
ResourceNotFound.AuthRuleNotExists	
InvalidParameterValue.LaneRuleRemarkTooLong	
FailedOperation.LaneRuleMaxLimit	
InvalidParameterValue.RatelimitParamError	
InternalError.TsfApmCallApaasFailed	
InternalError.TkeApiFailedOperation	
MissingParameter.ProgramIdRequired	
InvalidParameterValue.LaneInfoNameNotEmpty	
InvalidParameterValue.FileConfigFilePathInvalid	
InternalError.TsfApmCallMasterCheckVersionFailed	
InvalidParameter.TsfApmStdoutSearchRequestParamError	
InvalidParameterValue.NamespaceDescInvalid	
InvalidParameter.RepositoryNotEmpty	
UnauthorizedOperation.CamGeneralError	
InvalidParameterValue.NamespaceAlreadyBindCluster	
FailedOperation.TkeClusterDeleteFailed	

错误码	说明
InvalidParameter.TsfApmBusiLogCfgCloudParamError	
FailedOperation.TsfApmStatsSearchServiceQueryError	
ResourceInUse.ObjectExist	
ResourceNotFound.ErrNoRepo	
MissingParameter.RequestIsNull	
FailedOperation.ConfigTemplateUpdateFailed	
ResourceNotFound.StartShNotFound	
InvalidParameter.TsfAsPeriodError	
InvalidParameterValue.ClusterNameExist	
MissingParameter.TaskApplicationIdNull	
InvalidParameter.TsfApmBusiLogCfgAppParamError	
InternalError.SidecarFilterLuaDecodingFailed	
InvalidParameterValue.TsfAsNameRepeat	
InvalidParameterValue.ServiceApiExists	
FailedOperation.ConfigCreateFailed	
FailedOperation.UpdateHiddenStatusFailed	
FailedOperation.TaskTerminateFailed	
ResourceInUse.InstanceHasBeenUsed	
FailedOperation.TsfPrivilegeError	
InvalidParameter.TsfApmNoPermissionError	
InvalidParameterValue.ResourcePermissionDenied	
MissingParameter.ConfigValueRequired	
FailedOperation.TaskQueryError	
InvalidParameter.TsfApmUnknownInstance	
InvalidParameterValue.LaneRuleInfoNotExist	
InvalidParameter.LaneInfoNameAlreadyUsed	
InvalidParameter.LaneInfoNotExist	
InternalError.UnhandledException	
InvalidParameterValue.TsfAsVmcountlimitError	

错误码	说明
ResourceInUse.RatelimitRuleExistError	
MissingParameter.TsfAsRulelistNull	
InvalidParameterValue.ConfigReleaseNotExists	
InvalidParameterValue.CvmCaeMasterTaskExeBatchNotExist	
InvalidParameterValue.FileConfigAlreadyReleased	
FailedOperation.LaneInfoDeleteConsulFailed	
MissingParameter.VpcIdRequired	
InvalidParameterValue.LaneRuleTagValueTotalTooLong	
InvalidParameterValue.RoleNotExist	
InvalidParameterValue.CircuitBreakerInvalidMsId	
InvalidParameter.PackageInUse	
FailedOperation.TaskOperationForbidden	
InternalError.InstanceCommonError	
InvalidParameter.ConfigTemplateNameInvalid	
InvalidParameterValue.ApplicationNotExists	
ResourceInUse.CvmcaeMasterCannotDelete	
InvalidParameter.LaneInfoRemarkTooLong	
InvalidParameterValue.ReleasedFileConfigCanNotBeDeleted	
InvalidParameterValue.DuplicateProgramName	
InvalidParameterValue.GroupDeleteClusterTypeMismatch	
InvalidParameterValue.ContainergroupGroupnameLegnth	
FailedOperation.TsfCmonitorCtsdbClientRequestFail	
InternalError.RouteMsServiceError	
LimitExceeded.ErrRepoMaxLimit	
InternalError.CloudapiInvokeError	
ResourceNotFound.ObjectNoExist	
InternalError.CvmCaeMasterDispatchError	
InvalidParameterValue.GatewayParameterError	
InvalidParameterValue.ContainergroupCreateUinNull	

错误码	说明
InvalidParameterValue.LaneRuleNameTooLong	
InternalError.RuntimeError	
InvalidParameter.LaneRuleTagValueTooLong	
InternalError.CpClusterUnavailable	
FailedOperation.TsfMsServerError	
InvalidParameter.LaneInfoNotExistEntrance	
InvalidParameterValue.LaneRuleNameAlreadyUsed	
FailedOperation.TsfAsDelRuleFail	
InvalidParameterValue.BuketInvalid	
FailedOperation.TaskUpdateError	
MissingParameter.GroupIdNull	
InternalError.TsfApmBusiLogSearchNoTargetConfigError	
InvalidParameterValue.ConfigTemplateDescTooLong	
InvalidParameterValue.LaneRuleTagNameTooLong	
MissingParameter.ApplicationMicroTypeNull	
InvalidParameterValue.TsfAsCooltimeError	
InvalidParameterValue.TargetServiceInSource	
FailedOperation.ConfigGroupQueryFailed	
MissingParameter.ConfigNameRequired	
InvalidParameterValue.InstanceInvalidImage	
InvalidParameterValue.ContainergroupResourceAgentValueInvalid	
InternalError.TsfAsResourceServerError	
ResourceInUse.RoleNotExist	
InvalidParameterValue.ConfigNameInvalid	
InvalidParameterValue.LaneRuleTagNameNotEmpty	
InvalidParameterValue.CvmCaeMasterAgentBusy	
InvalidParameterValue.ContainergroupAccesstypeNull	
InternalError.TsfApmAgentInternalError	
InvalidParameterValue.ConfigValueFormatInvalid	

错误码	说明
MissingParameter.ConfigTemplateIdRequired	
InternalError.ContainergroupKuberneteDeploymentNotfound	
UnauthorizedOperation.TsfApmApaasUnsupportedVersion	
InvalidParameterValue.SidecarFilterDraftNotReleasedFilterError	
InvalidParameterValue.CvmCaeMasterTaskBatchIndexNotMatch	
InternalError.CanNotConnConsulServer	
InvalidParameterValue.TsfAsNameExceedLen	
InternalError.TsfOperationApplicationError	
InternalError.TsfMonitorInternalError	
InvalidParameterValue.LaneRuleNotExist	
InvalidParameterValue.ContainergroupProtocolInvalid	
FailedOperation.CircuitBreakerParamInvalid	
InvalidParameter.LaneRuleNameNotEmpty	
InvalidParameterValue.GroupIdNull	
InternalError.TsfMonitorDateParseFailed	
InvalidParameterValue.ContainergroupGroupidNull	