



腾讯云

腾讯TSF产品交流

目录

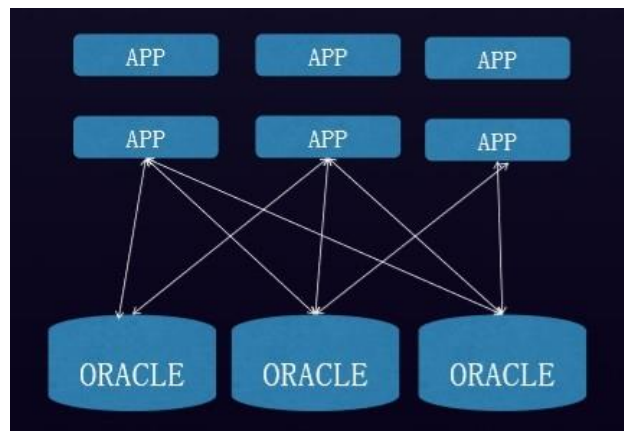
1. TSF 产品介绍
2. 客户案例

TSF 产品介绍



企业IT架构的变迁

传统IOE架构



- 每个应用彼此没有太大关系，按**烟囱式**排列，唯一的共通点在于都与底层的数据库相连；
- 每个应用都比较庞大，同时需要连接多个数据库；
- 架构中的应用数量较少，应用与应用之间的关系简单

传统“中心化”（ESB）系统架构



- 服务调用者与服务提供者通过企业服务总线相连接；
- **ESB成为瓶颈**：无论在性能上还是成本消耗上，ESB都会导致瓶颈出现；

去“中心化”的微服务架构



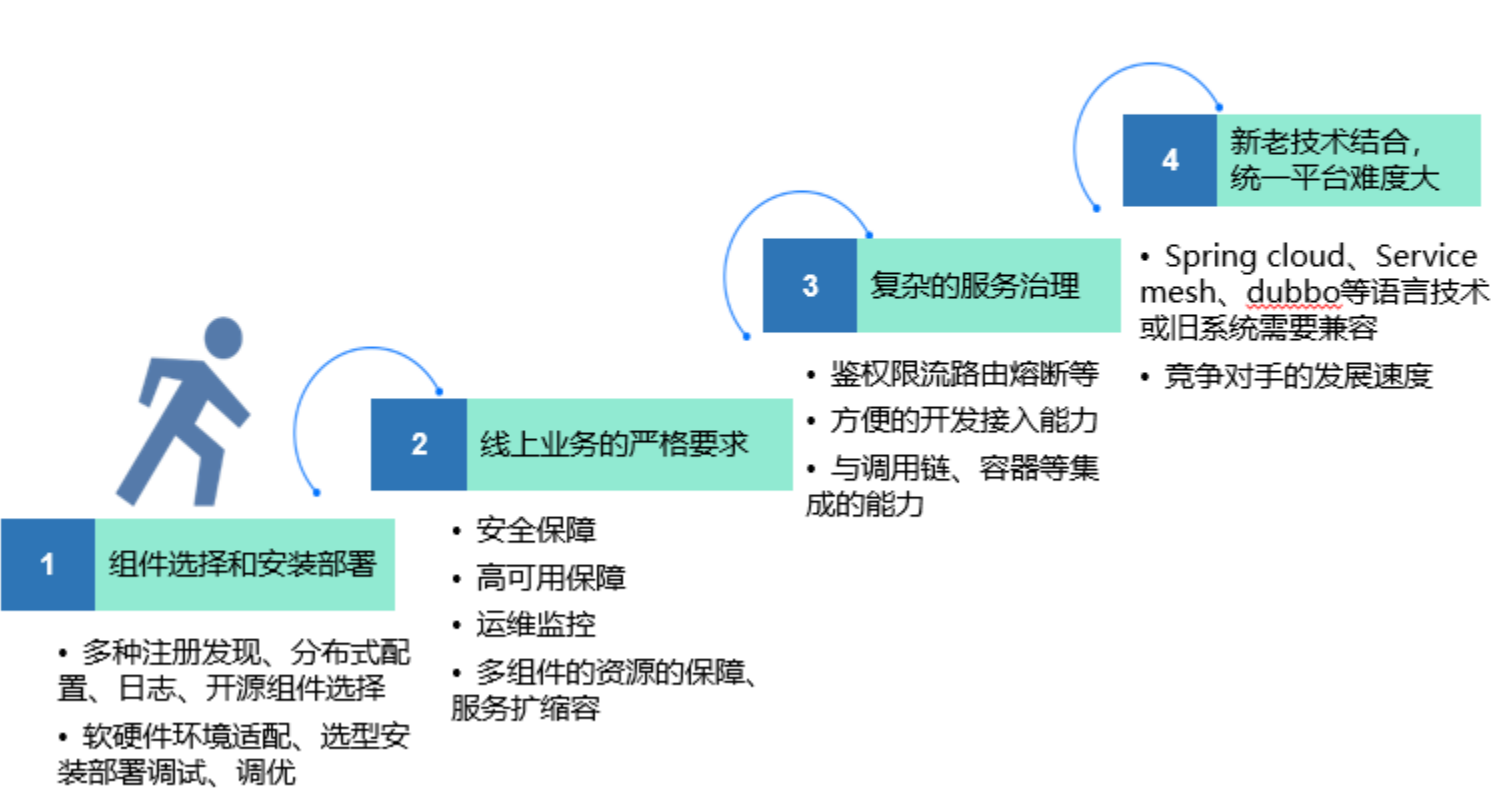
- 业务系统的扩展没有瓶颈，只需按照业务发展需要进行扩展；
- **敏捷**，迭代速度可以非常快；

单体架构 VS 微服务架构

	单体架构	微服务架构
迭代速度	较慢	快
部署频率	不经常部署	经常发布
系统性能	吞吐量小	吞吐量大
系统扩展性	扩展性差	扩展性好
技术栈多样性	单一、封闭	多样、开放
运维	简单	运维复杂
部署难度	容易部署	较难部署
架构复杂度	较小	复杂度高
查错	简单	定位问题困难
管理成本	主要在于开发成本	服务治理、运维

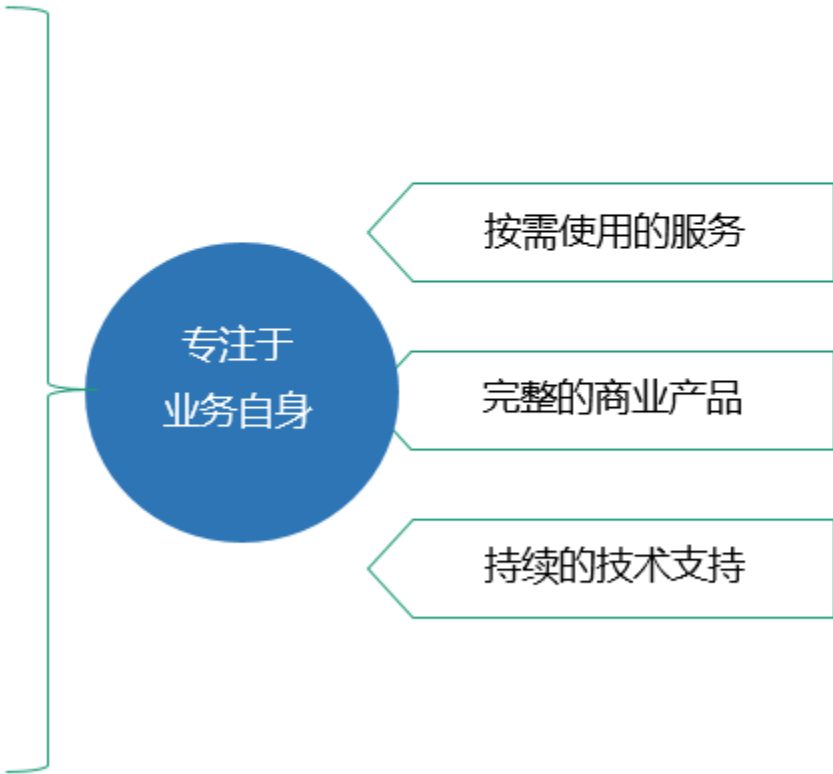
自建微服务的平台挑战

需要长期的大量研发投入、时间、资源很大，学习成本高，影响业务的发展时机。



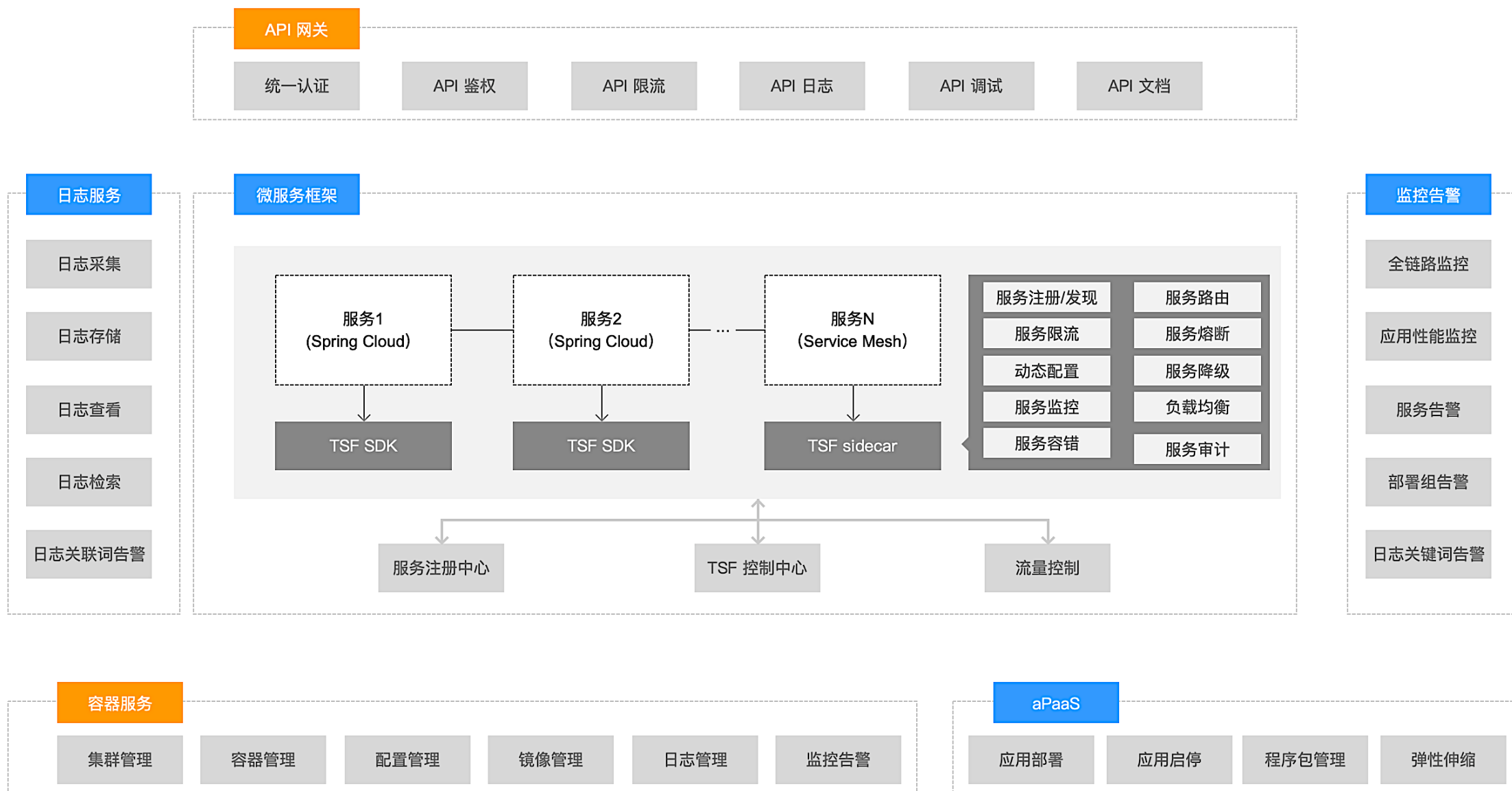
与云优势结合

将复杂的组件运维托管给云服务
享受微服务平台的便利



CONFIDENTIAL MATERIAL FROM TENCENT CLOUD

TSF能力全景



TSF 支持 Spring Cloud & Service Mesh 两种微服务，提供配套的开发使用指引。



Spring Cloud 商业化版

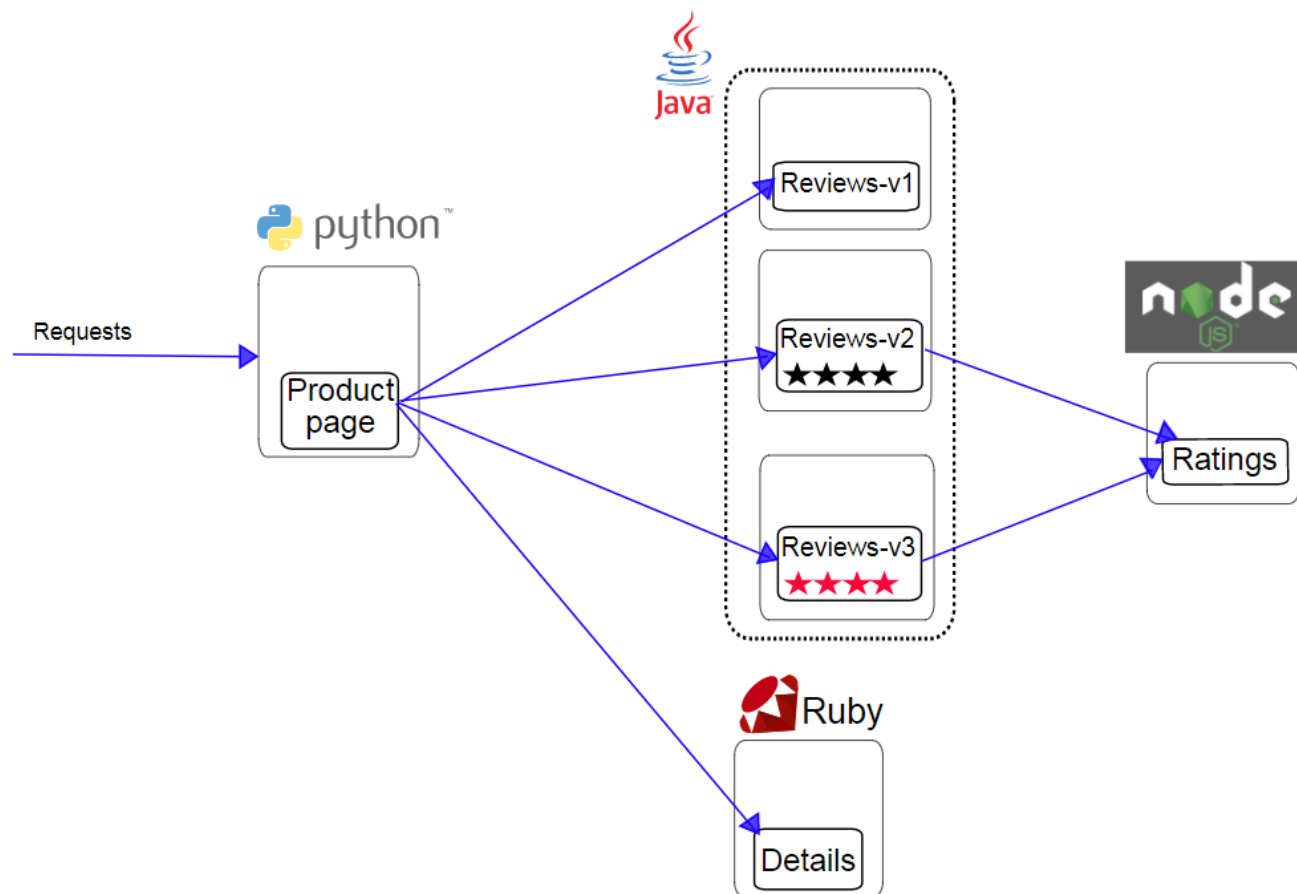
- 原生Spring Boot、Spring Cloud开发体验
- 迁移无改造量，自动接入TSF 注册中心
- 兼容 E / F / G 三个 Spring Cloud 版本
- 基于 **TencentJDK**，针对云应用场景定制新的功能及优化，具备更快的云应用启动速度，更好的性能以及更为便捷的分析、诊断工具
- Spring 框架层面整合腾讯消息队列、API网关、云 kafka 大数据套件等传统中间件



Tencent Service Mesh

- 原生 istio、envoy 进行深度构建
- 满足公司内部不同开发语言、框架的无缝互通
- Service Mesh 通过 sidecar 的方式组织服务间的流量，开发者更聚焦业务开发

Service Mesh典型场景

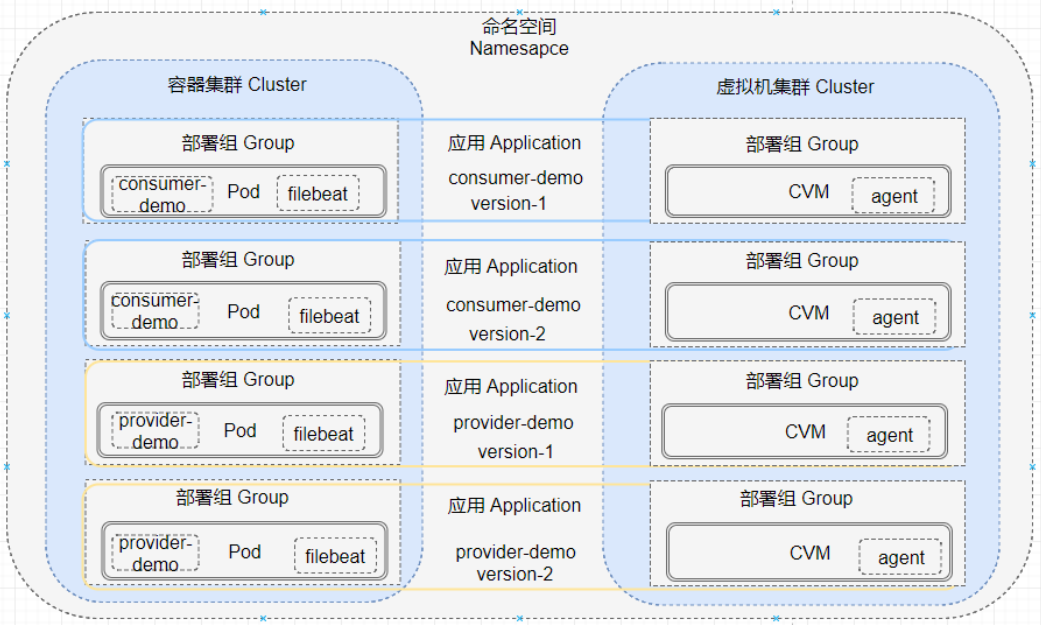


- 集群中存在**多语言**服务互通的场景
- 服务底层基于非Springboot框架进行开发
- **遗留系统**，不希望因为微服务化而做过多的侵入式变更
- 服务运维人员希望服务框架的升级无需业务联动升级

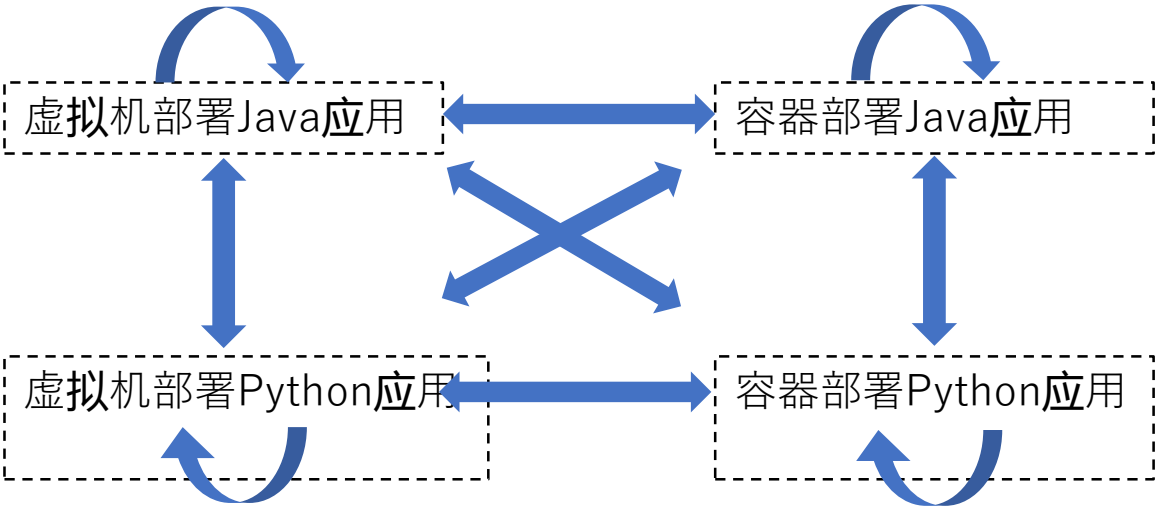
以上场景，我们都建议使用
Service Mesh

微服务TSF平台提供跨语言，跨物理资源的通道

打通容器和虚拟机的互访



打通spring cloud和service mesh 互访

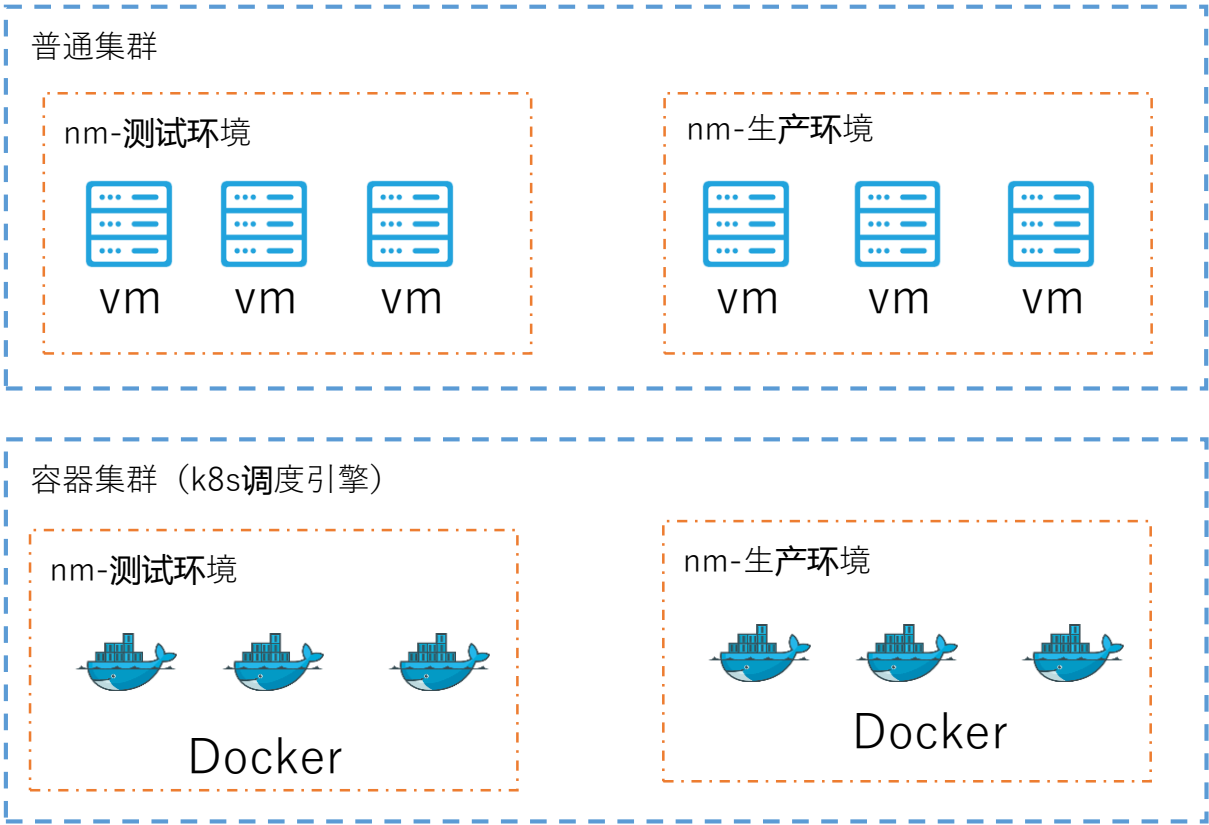


TSF 提供应用全生命周期管理能力

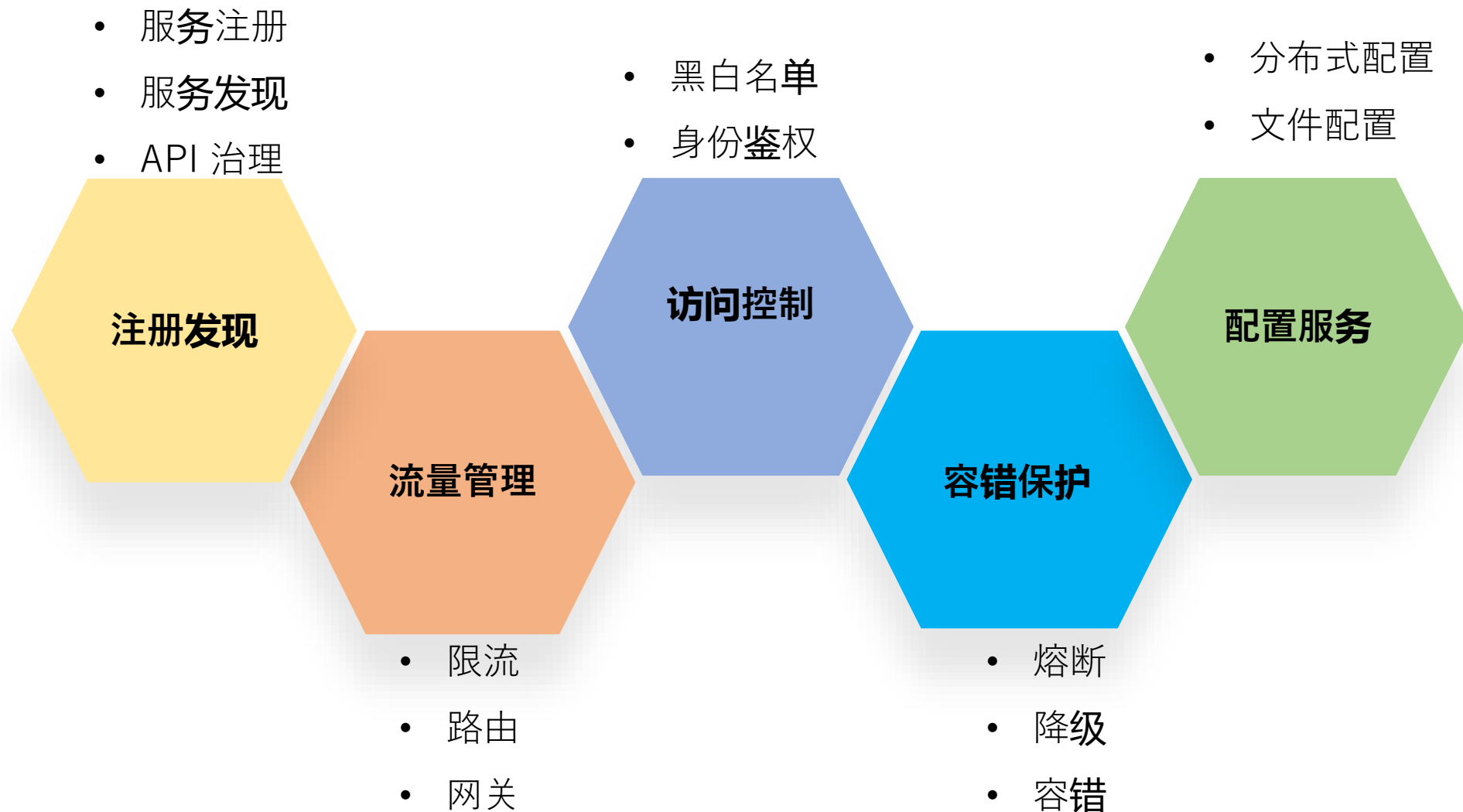
- 集群用于纳管云服务器，作为部署资源池
- 应用支持部署、启动、停止、手动自动扩缩容操作
- 支持配置启动参数
- 支持进程自动拉起
- Serveless / Docker / VM 三种部署方式
- 弹性伸缩规则支持 IaaS 指标和业务指标（响应时间、请求 QPS）



业务集群



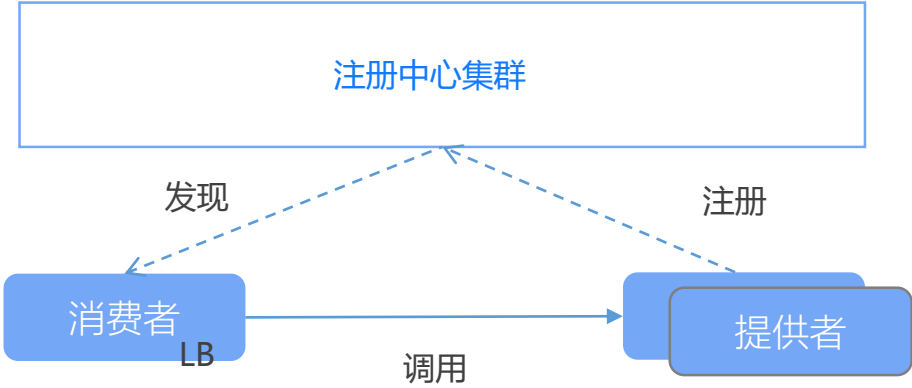
TSF 提供 5 大服务治理能力，为企业构建 高可靠、高容错、高效率 的微服务治理体系



服务注册与发现

核心优势：跨可用区容灾、本地缓存、动态上下线

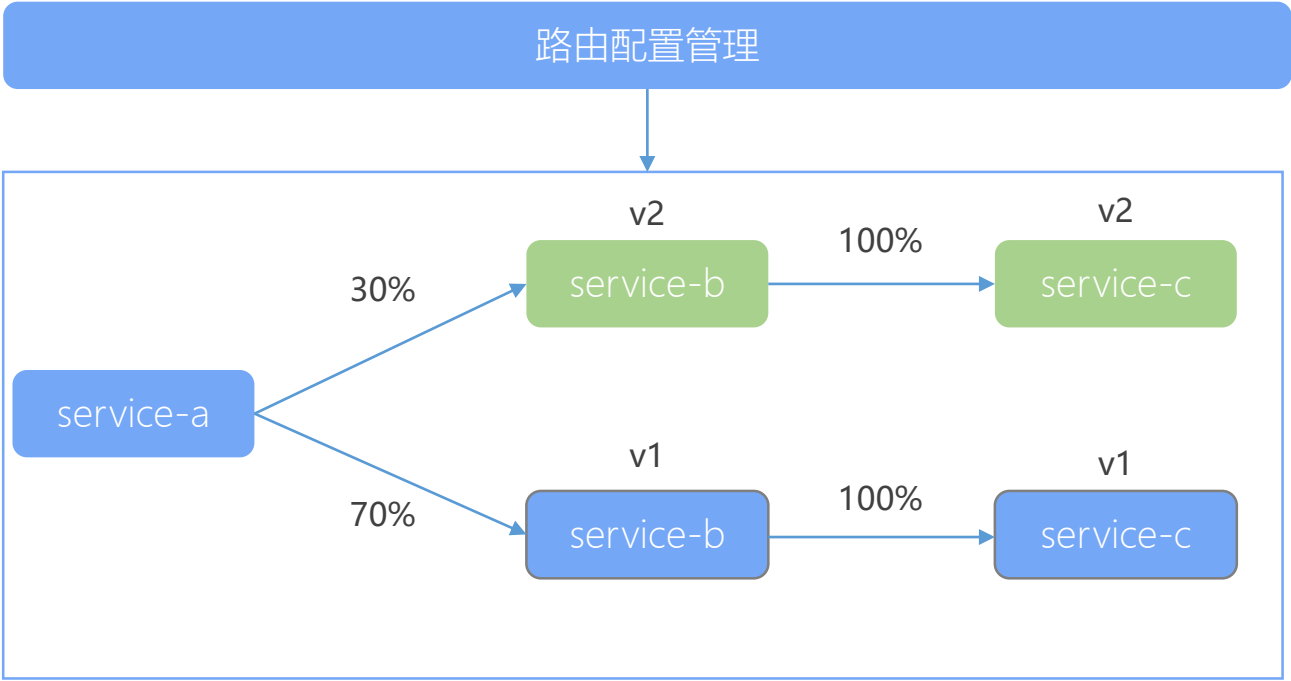
- 服务在启动时会**自动注册**到注册中心，同时从注册中心发现其他服务与**IP的映射表**
- 服务消费者通过**本地负载均衡**策略选择提供者的目标IP 进行调用
- 注册中心集群通过 Raft 算法保证高可用
- 服务会本地缓存服务与IP的映射表，当注册中心不可用时，消费者仍然能正常发起调用



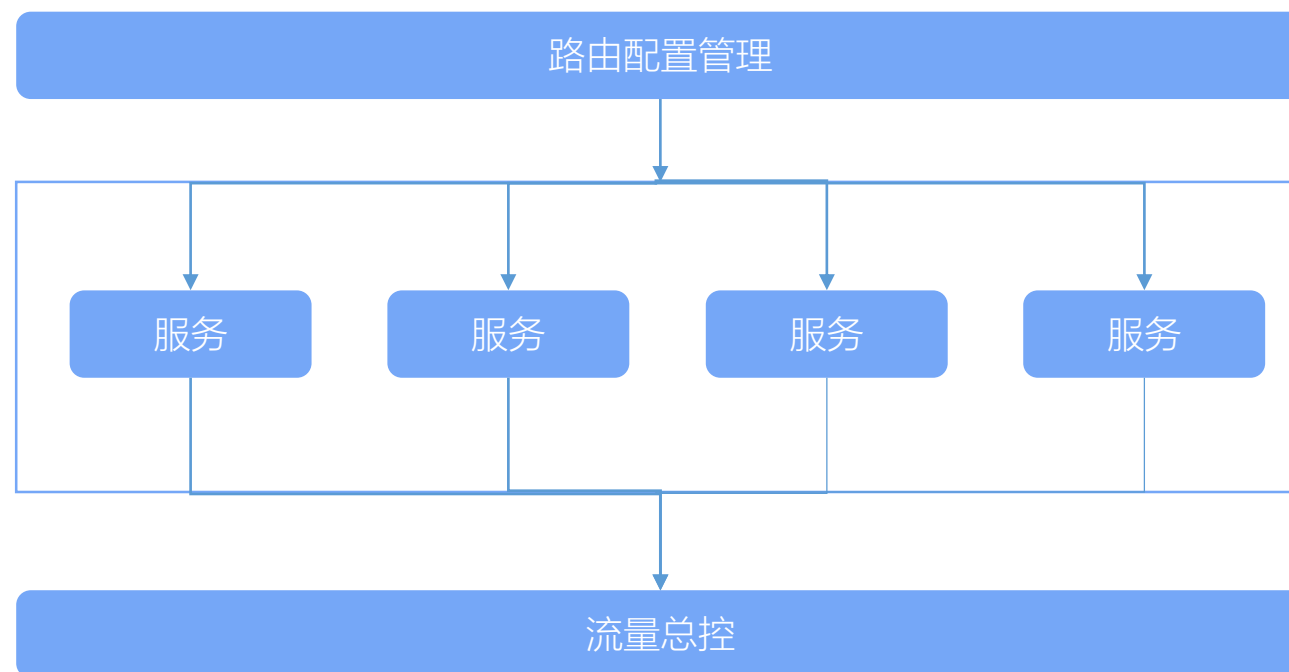
微服务名称	状态	运行服务实例数	请求量①	请求成功率①	请求平均耗时(ms)①
ms-m6yormyl consumer-demo	单点在线	1	2	0 %	0.58 ms
ms-gnair4aq provider-demo	在线	2	85,925	100 %	0.68 ms
共2项					

使用场景：灰度发布、就近路由、读写分离、功能内测

- 用户配置针对服务提供者服务路由规则
- 路由配置会下发到服务消费者，消费者根据**路由配置**进行流量分配
- 流量根据**系统标签**和业务**自定义标签**进行划分
- 用户可配置路由容错保护开关，开启时会采用兜底策略

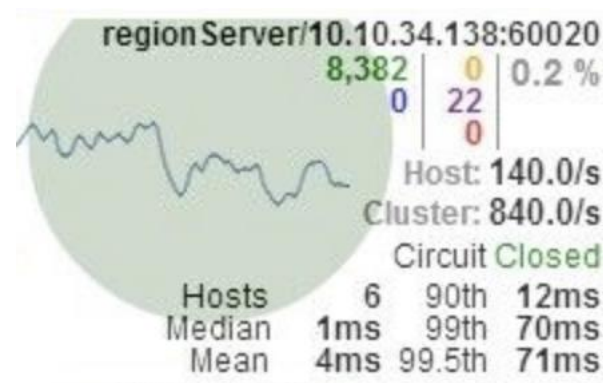
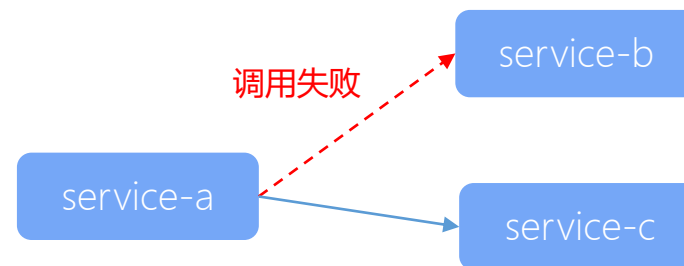


- 用户**配置**针对服务或者标签的**限流**下发到服务中的所有实例
- **流量总控**会实时获取每个节点上的流量情况
- 流量总控经过计算后再将是否拦截下发到服务的每个实例
- 整体实现限流**动态监测**和调整

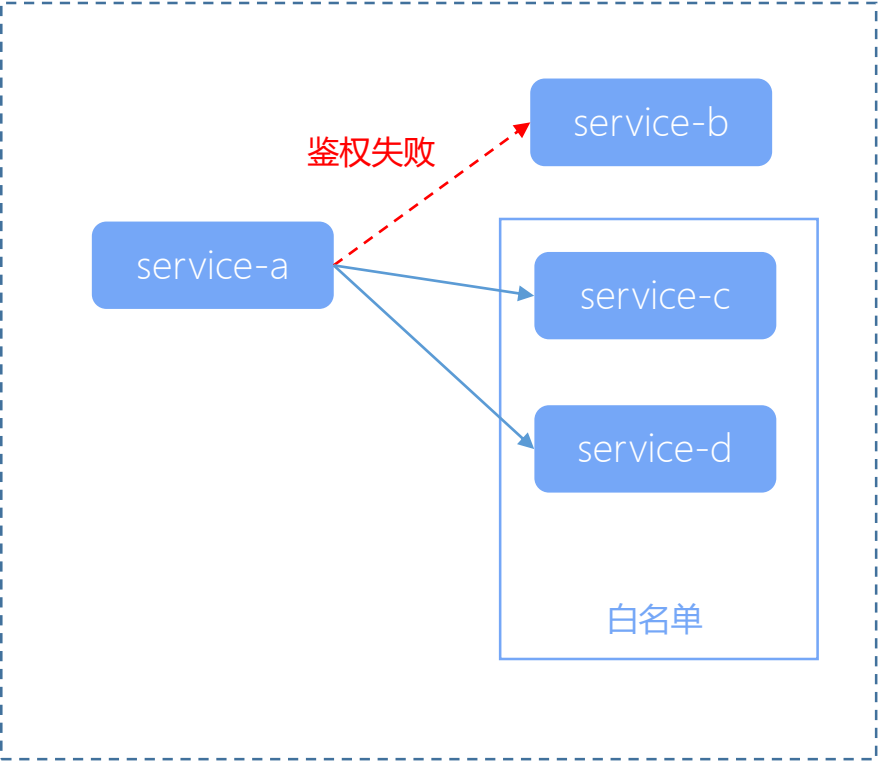


使用场景：异常检测、异常隔离、动态恢复

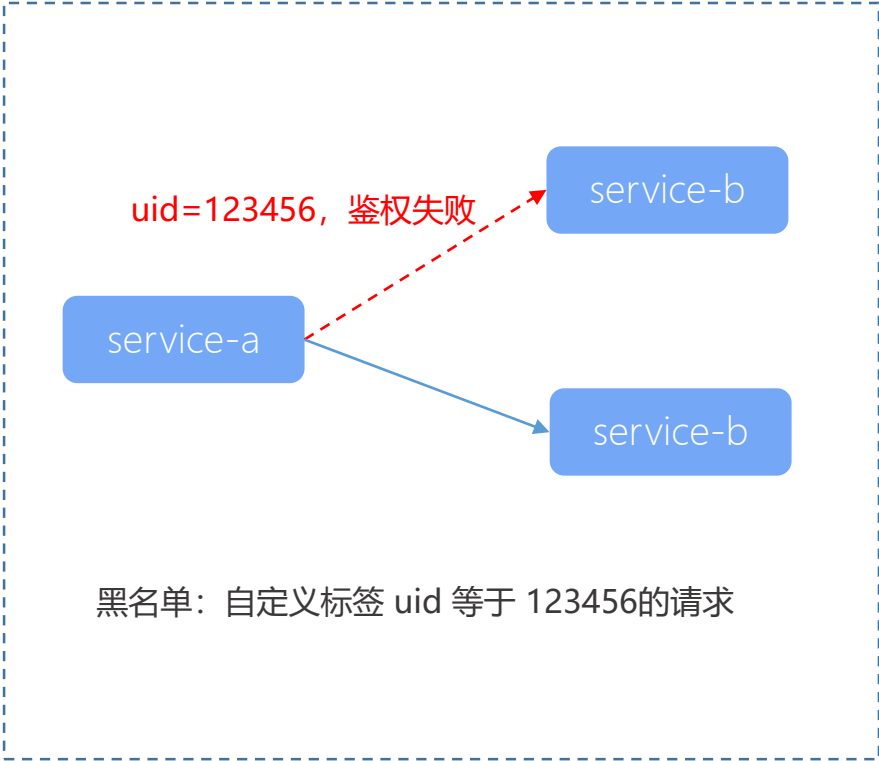
- 用户**配置**针对服务或者实例的**熔断**下发到服务中的所有实例
- 服务调用方统计滑动窗口内请求失败率，达到熔断条件进行熔断；sleep 时间后熔断器半开启状态
- 调用方可选择抛出异常或者执行 fallback 降级处理
- 对服务请求**响应耗时**、**成功失败次数**、**熔断器状态**进行监控



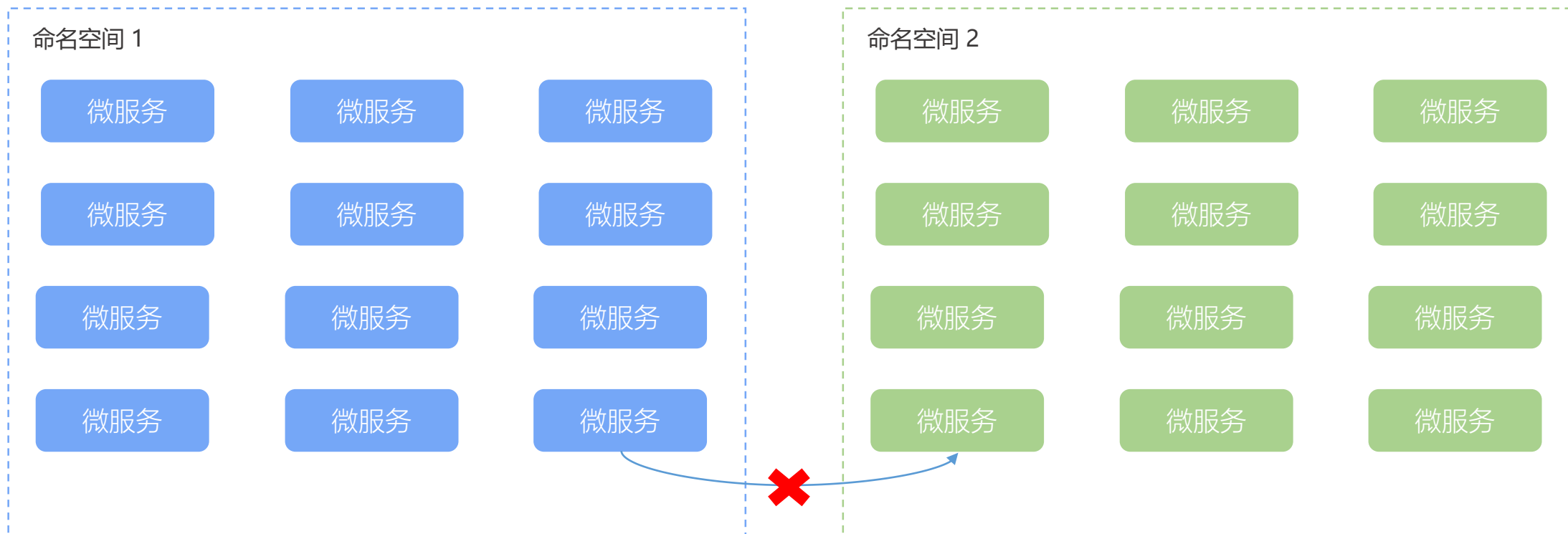
白名单：仅匹配鉴权规则的请求允许调用
系统标签 & 自定义标签



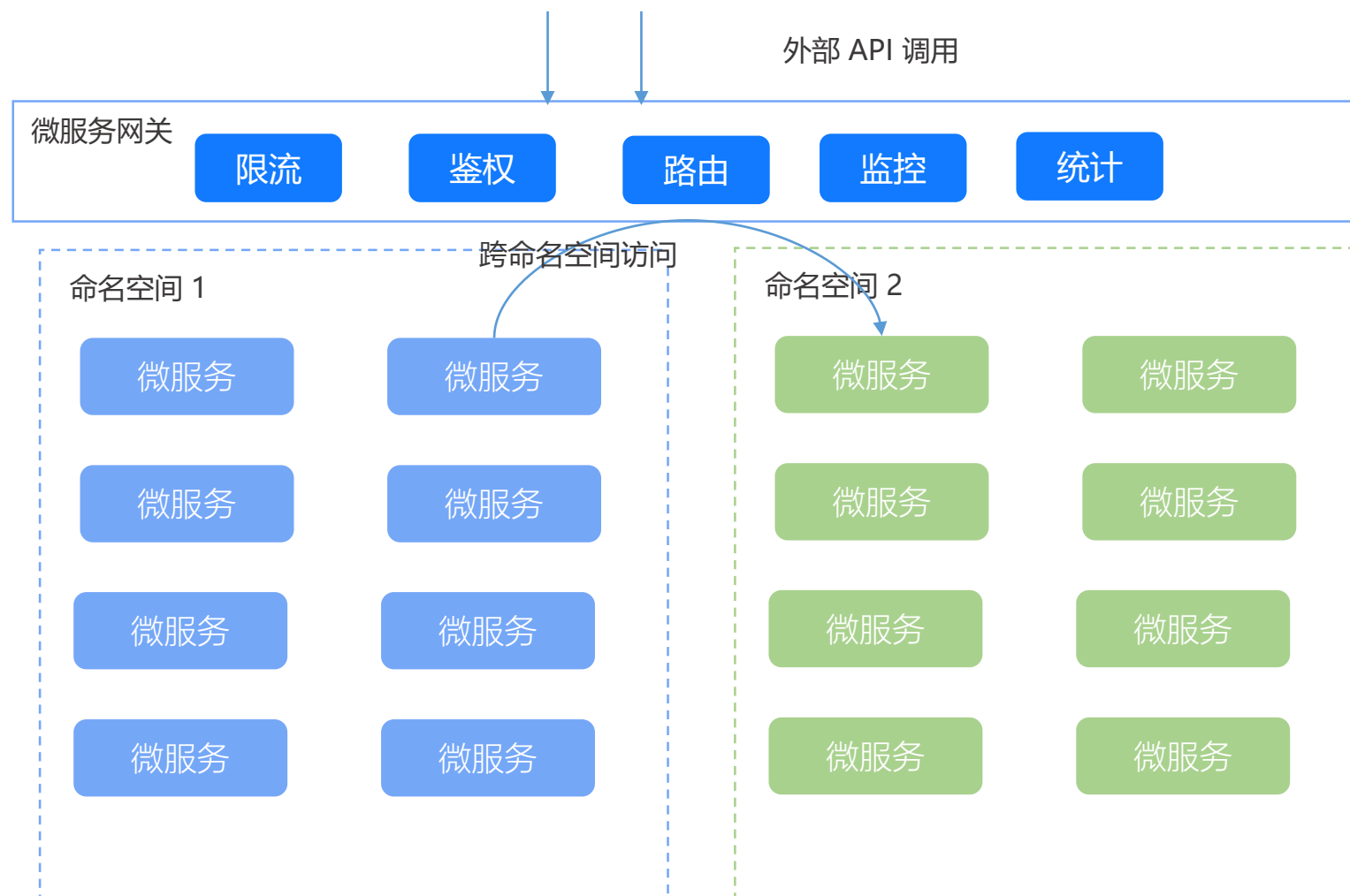
黑名单：匹配鉴权规则的请求拒绝调用
系统标签 & 自定义标签



服务隔离

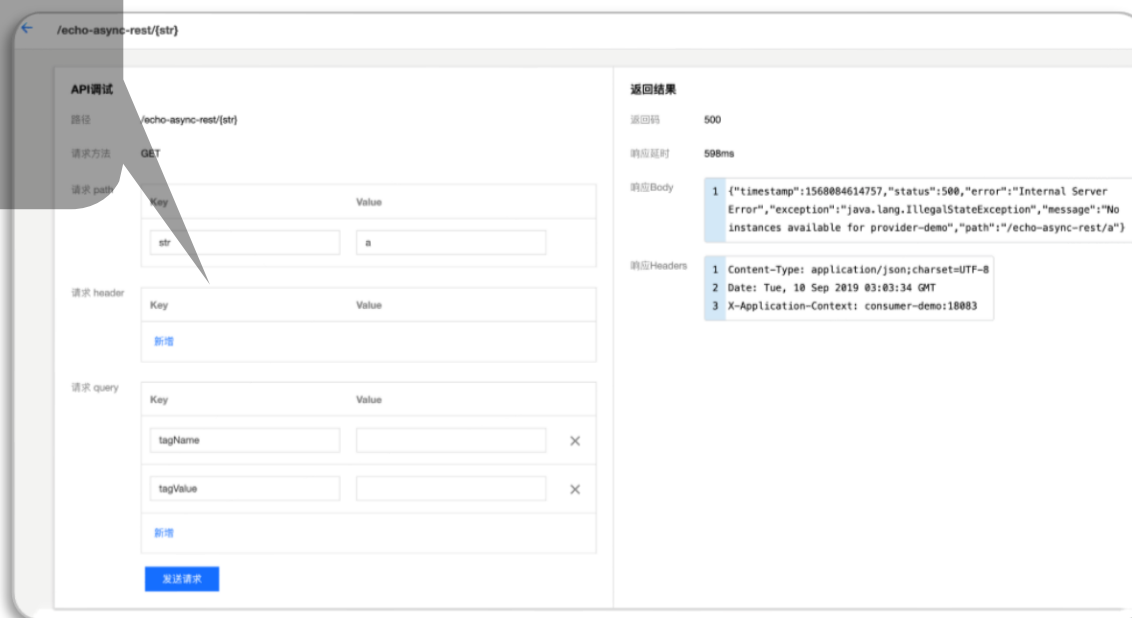


- 请求鉴权
- 请求限流
- 分组管理
- 密钥管理
- 域名配置
- API导入
- 网关管理
- 自动伸缩
- 监控/统计
- 跨命名空间访问
- 请求转换



- 提供自动上报业务 API 能力，在线查看 API 详情
- 生成并导出 API 文档

- API 在线调试，提高服务调用的联调效率
- Mock 测试



- 批量部署
- 版本管理
- 灰度发布
- 优雅启停
- 弹性伸缩
- 自定义部署脚本
- 包管理&镜像管理
- 健康检查
- 监控统计

腾讯微服务平台

应用中心

应用管理

application-qv35ngy7 (demo_provider)

部署组

程序包管理

变更记录

应用配置

监控

基本信息

新建部署组

请输入分组名

ID/部署组名	集群	命名空间	状态	已部署程序	应用	已启动/总机器数	操作
group-jy9j98vg grey-provider	cluster-byx63dyf Demo Cluster	Demo Cluster_default	运行中	1.12.2 provider-demo-1.12.2-Edgware-...	application-qv35ngy7 demo_provider	已启动1台/共1台	部署应用 更多
group-nygwemv2 normal-provider	cluster-byx63dyf Demo Cluster	Demo Cluster_default	运行中	20190401190311 provider-demo-1.12.1-Edgware-...	application-qv35ngy7 demo_provider	已启动1台/共1台	部署应用 更多

新建规则

新建弹性伸缩规则

名称

触发以下 所有 指标:

CPU利用率

1分钟内

≥

%

连续1次

内存利用率

1分钟内

≥

%

连续1次

响应时间

1分钟内

≥

ms

连续1次

触发指标后的伸缩活动:

减少

台

云主机

冷却时间 60 秒

最小实例数

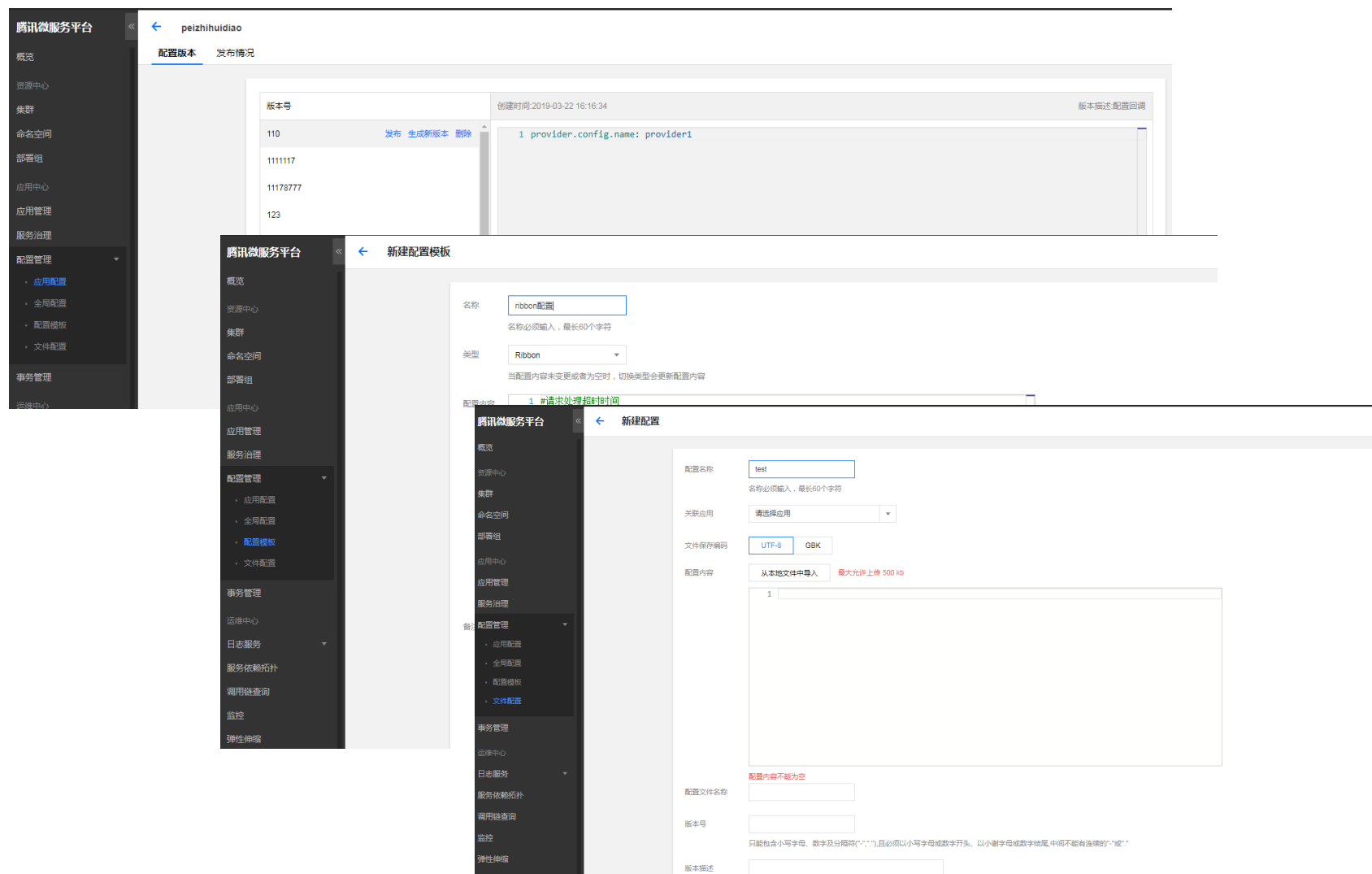
确定 取消

dongyuan_tsfas_01 详情

基本信息 关联应用 伸缩活动

活动ID	描述	影响主机	状态	开始时间	结束时间
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj ins-9h3tnhbj ins-9h3tnhbj	成功	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj ins-9h3tnhbj ins-9h3tnhbj 等共5台云主机	失败	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj	成功	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-h7xyj089 ins-kdht2xkn ins-14v1c40r ins-1276y8ir ins-9h3tnhbj ins-ho7zj9jl ins-213cvq0p ins-7moyn1fp	成功	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj	成功	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj	成功	2018-02-27 08:03:50	2018-02-28 15:42:18
tsfasa-213cvq0p	因CPU利用率大于60%，触发app_1, group_1 增加5%的云主机	ins-9h3tnhbj	成功	2018-02-27 08:03:50	2018-02-28 15:42:18

- 配置版本管理
- 实时更新
- 灰度发布
- 多环境发布
- 动态配置
- 配置回调
- 文件配置
- 配置模板
- 全局配置
- 配置本地缓存
- 配置加密



分布式事务：对多个系统进行操作，如何保证同时成功或者同时失败。典型场景：A向B转账10元。

TCC 事务

TCC 模式是一种补偿性分布式事务，包含 Try、Confirm、Cancel 三种操作

FMT 事务

FMT 模式是一种补偿性分布式事务，无须开发者编写 Cancel 操作，框架自动生成数据库逆操作

消息队列事务

不同系统间通过消息队列传递数据，系统维护业务表和消息状态发送（或消费）表

事务管理

时间

近1小时

近6小时

近12小时

近1天

近3天

2018-01-28 11:53:33 至 2019-01-29 11:53:33

🔍

状态

事务已确认

事务待确认

事务已取消

事务已超时

全部

关键字

查询

ID	上游节点	服务名	方法名	子事务数	创建时间更新时间	状态	超时时长
9c3c9466-0651-31bb-9d1a-4ba2039ac57b	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:58 2018-01-28 20:43:58	事务已确认	-
a6d5c03f-1c32-3c1d-9489-5564	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:58 2018-01-28 20:43:58	事务已确认	-
c11af020-0a19-38a3-8287-0862	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:57 2018-01-28 20:43:59	事务已确认	-
f10833ae-73a5-39c9-882a-4ee	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:57 2018-01-28 20:43:57	事务已确认	-
0ac94609-31a4-35d7-8809-60	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:56 2018-01-28 20:43:57	事务已确认	-
0f0a6050-7a78-38a9-a8a4-6844	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:56 2018-01-28 20:43:57	事务已确认	-
22b5f009-8d29-38a2-8f85-69e	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:56 2018-01-28 20:43:57	事务已确认	-
a9021300-6a62-3af6-aee0-dca	10.0.1.118	serviceBuy	serviceBuy	2	2018-01-28 20:43:56 2018-01-28 20:43:58	事务已确认	-

58.0

← 9c3c9466-0651-31bb-9d1a-4ba2039ac57b

基本信息

ID

9c3c9466-0651-31bb-9d1a-4ba2039ac57b

服务名

serviceBuy

方法名

serviceBuy

上游节点

10.0.1.118

子事务数

2个

状态

事务已确认

超时时间

-

创建时间

2018-01-28 20:43:58

修改时间

2018-01-28 20:43:58

子事务

ID	服务名	方法名	创建时间更新时间	状态
9c3c9466-0651-31bb-9d1a-4ba2039ac57b	service1	service1Try	2018-01-28 20:43:58 2018-01-28 20:43:58	事务已确认
9c3c9466-0651-31bb-9d1a-4ba2039ac57b	service2	service2Try	2018-01-28 20:43:58 2018-01-28 20:43:58	事务已确认

基于业务的弹性伸缩

- 场景：在一些突发情况下，应用访问量增加，导致机器高负载，需要进行扩容；当访问量降低时自动缩容
- 指标：CPU利用率、内存利用率、访问量和处理时间等维度自定义策略

新建弹性伸缩规则

名称

最长60个字符

扩容活动

☒ 启用

满足以下任一指标，将触发扩容操作

CPU利用率

>

90

%

持续1分钟

删除

内存利用率

>

80

%

持续1分钟

删除

响应时间

>

500

ms

持续1分钟

删除

请求QPS

>

10000

个/秒

持续1分钟

删除

增加

10

个

实例，最大实例数

50

个

缩容活动

☐ 启用

冷却时间

秒

建议冷却时间大于扩容活动中的持续时间，如持续时间设置1分钟，冷却时间设置5分钟

提交

关闭



监控告警

- 对象
服务、应用、日志
- 指标
响应时间、失败率、QPS、服务健康度、日志关键词

依赖拓扑&调用链

- 优势
定位系统瓶颈、服务健康度全局视角
- 增强
上下游组件打通、联动日志和监控

日志服务

- 优势
日志聚合，关键词检索，日志告警
- 增强
与调用链联动、实时日志

警告 监控盘

- 监控：**多维度**监控（应用、服务、接口）。请求耗时分布、状态码分布、并发数、请求失败率
- 告警：服务指标、服务健康度、日志关键词

告警对象

全部对象

选择部分对象(已选0个)

选择实例组

新建实例组

地域: 广州

Q

<input type="checkbox"/>	微服务ID/名称	命名空间ID/命名空间名
<input type="checkbox"/>	ms-5ygr2vj echodubbo-service	namespace-qv3zkm9y pre-monitor
<input type="checkbox"/>	ms-gvkb8lvo citydubbo-service	namespace-qv3zkm9y pre-monitor
<input type="checkbox"/>	ms-nyg23pv2 timeoutchodubbo-service	namespace-qv3zkm9y pre-monitor
<input type="checkbox"/>	ms-qv3mpea7 exceptionchodubbo-ser...	namespace-qv3zkm9y pre-monitor

支持按住shift键进行多选

微服务ID/名称

命名空间ID/命名空间名

触发条件

触发条件模板

新增触发条件模板

配置触发条件

指标告警

接收请求平均耗时

统计周期1分钟

>

0

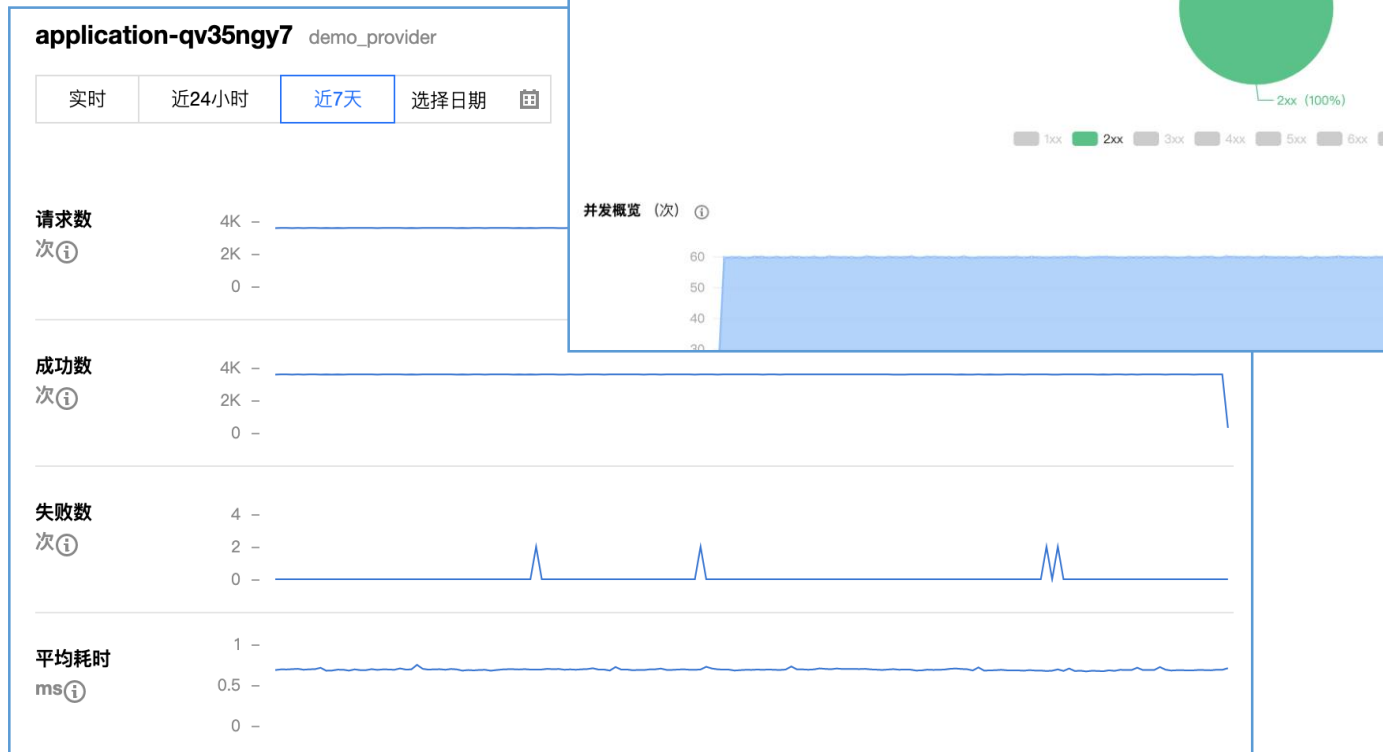
毫秒

持续1个周期

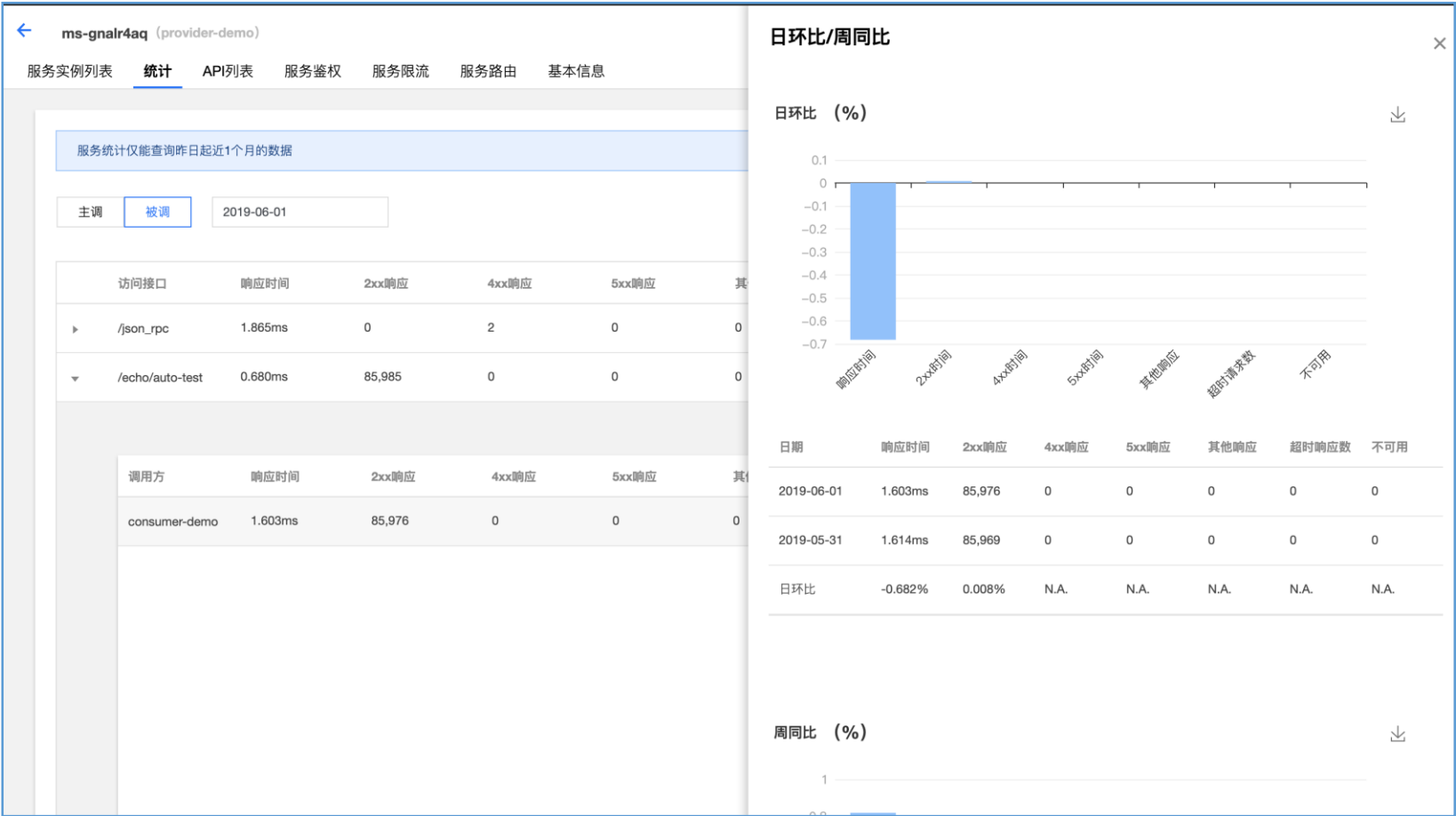
每1天警告一次

①

添加

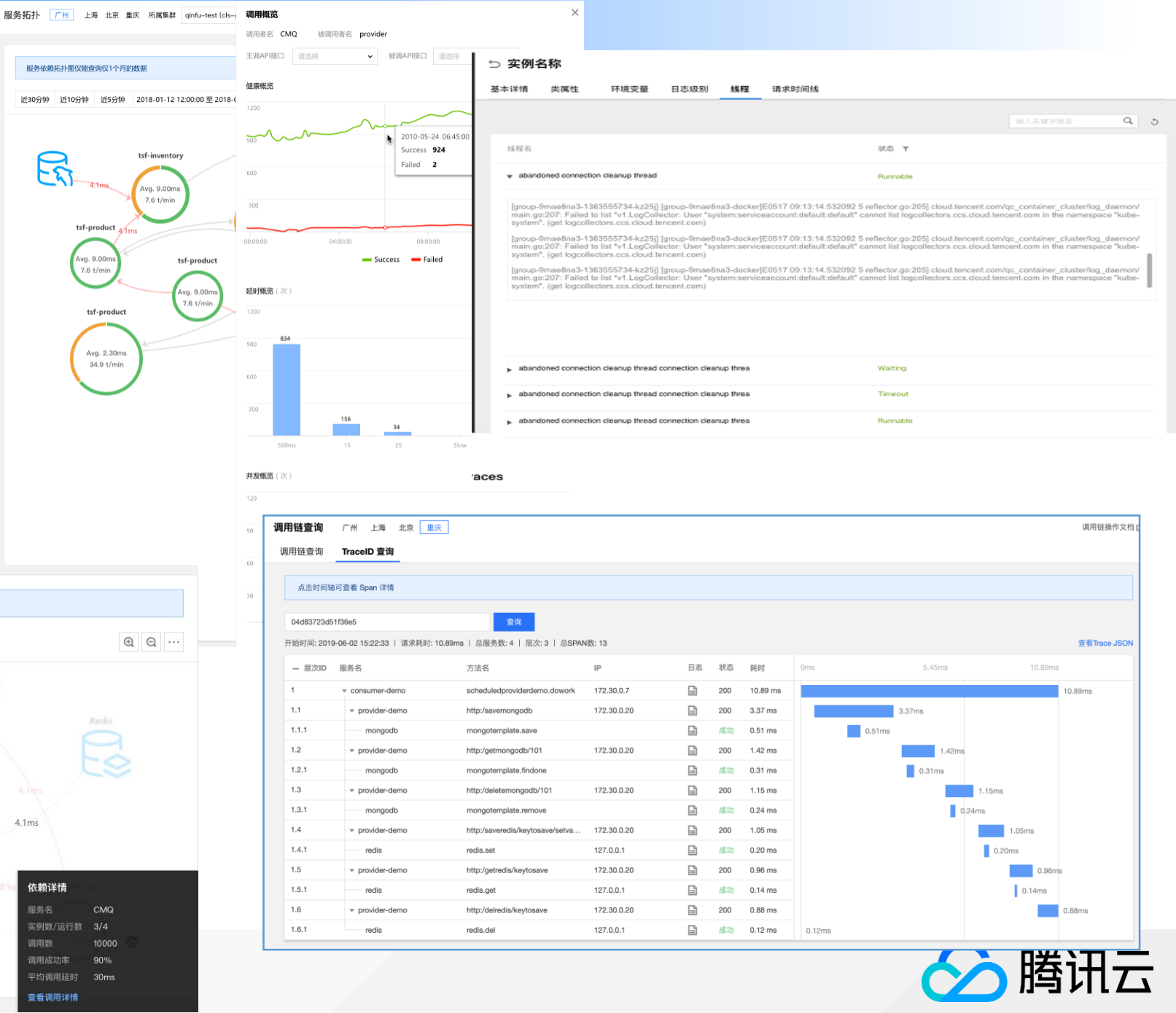


- 主调/被调两种视角
- 服务 / 接口两种维度
- 响应类型：2xx、 4xx、 5xx、不可用、超时
- 日环比、周同比数据对比



应用性能分析

- 通过依赖拓扑图查看服务之间、服务与上下游组件（MySQL、Redis、MongoDB、CMQ）之间的依赖关系和请求情况
- 通过调用链分析瓶颈、出错服务
- 调用链与日志联动，通过日志查看程序运行上下文



调用链与日志联动

- 调用链能**分析瓶颈、出错**服务，但是无法了解**错误**的具体原因
- 通过调用链与日志**联动**，可以**查看调用发生时刻服务打印的标准输出日志**



通过调用链可了解，consumer 调用 provider 返回 403 错误，通过 consumer 日志了解到是鉴权失败

调用链查询

广州 上海 上海金融 北京 重庆

调用链查询 TraceID 查询

点击时间轴可查看 Span 详情

80eb85f31a724f14 查询

开始时间: 2019-09-10 10:49:50 | 请求耗时: 21.61s | 总服务数: 3 | 层次: 3 | 总SPAN数: 3

层次ID	服务名	方法名	IP	日志	状态
1	producer-demo	do-work	172.21.64.2		200
1.1	kafka	send	172.21.0.98		成功
1.1.1	consumer-demo	poll	172.21.64.12		成功

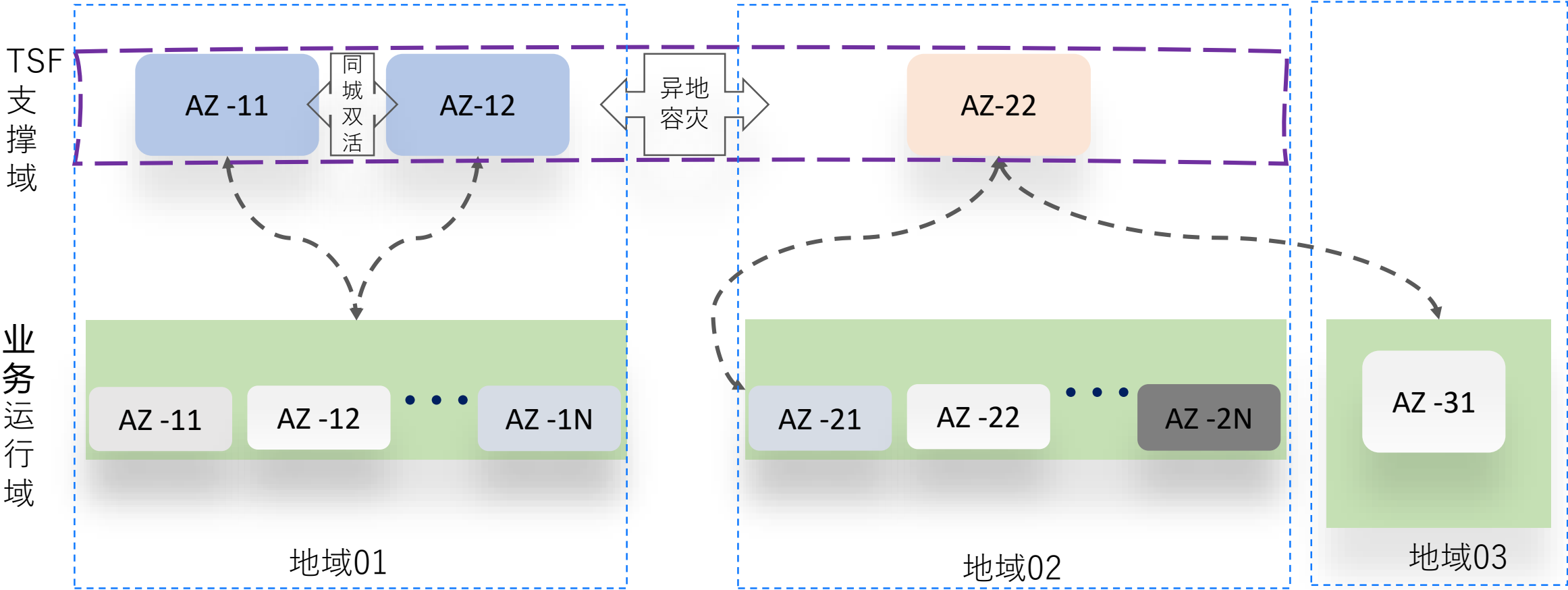
poll

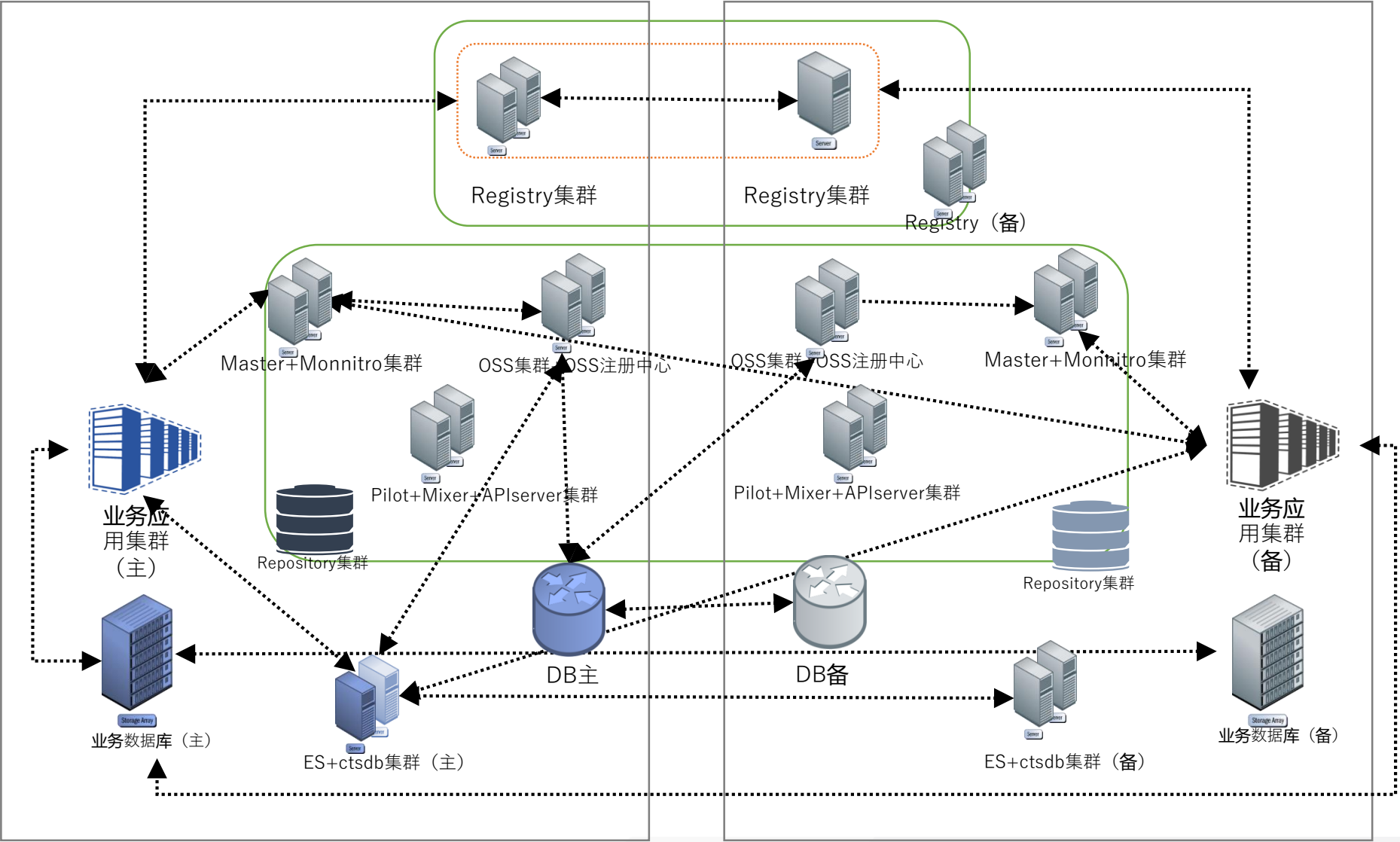
spanid: 70443ba975b41d7c

1 [] [2019-09-10 10:49:50.499 INFO [consumer-demo,80eb85f31a724f14,70443ba975b41d7c,true] 18298 --- [org.springframework.kafka.KafkaListenerEndpointContainer#0-0-C-1] c.t.k.consumer.cumsumer.KafkaReceiver : ----- record =ConsumerRecord(topic = zhizhang3, partition = 0, offset = 12131, CreateTime = 1568083790484, serialized key size = -1, serialized value size = 92, headers = RecordHeaders(headers = [], isReadOnly = false), key = null, value = {"id":55,"msg":"8bc45ddb-f027-4f9d-94f6-25d80e8fd2ad","sendTime":"Sep 10, 2019 10:49:50 AM"})

2 [] [2019-09-10 10:49:50.500 INFO [consumer-demo,80eb85f31a724f14,70443ba975b41d7c,true] 18298 --- [org.springframework.kafka.KafkaListenerEndpointContainer#0-0-C-1] c.t.k.consumer.cumsumer.KafkaReceiver : ----- message = {"id":55,"msg":"8bc45ddb-f027-4f9d-94f6-25d80e8fd2ad","sendTime":"Sep 10, 2019 10:49:50 AM"}]

TSF容灾架构，实现跨可用区实现同城双活、异地备份





客户案例



南方电网

南方电网于2019年启动南网云建设项目，目标将南网现有业务系统云化并整体迁移上云。安畅作为PaaS部分供应商提供TSF微服务平台及相关实施服务。

安畅提供服务：

- TSF微服务平台部署
- TSF微服务平台定制化功能开发
- TSF微服务平台运维
- TSF培训

客户收益：

- 实现多个分支机构的资源、应用的统一管理。
- 减少应用云化后运维成本的增幅，客户IT团队可以专注于业务升级。
- 大幅提高系统在业务高峰期的扩展能力和吞吐能力。
- 大幅增强了南网对企业及合作伙伴的服务能力和共享能力。



图：南网云PaaS总体架构

谢谢