

通过混沌工程构建高可用的分布式服务

01010101

肖长军（花名：穹谷）

阿里巴巴 技术专家

自我介绍

- 肖长军，花名 穹谷，阿里高可用架构团队
- 多年应用性能监控研发和分布式系统高可用架构经验
- 阿里云应用高可用服务（AHAS）产品技术负责人之一
- 阿里集团故障演练、突袭演练、攻防演练核心开发
- 开源项目 ChaosBlade 负责人
- 混沌工程布道师



目录

01

混沌工程
介绍

02

ChaosBlade
介绍

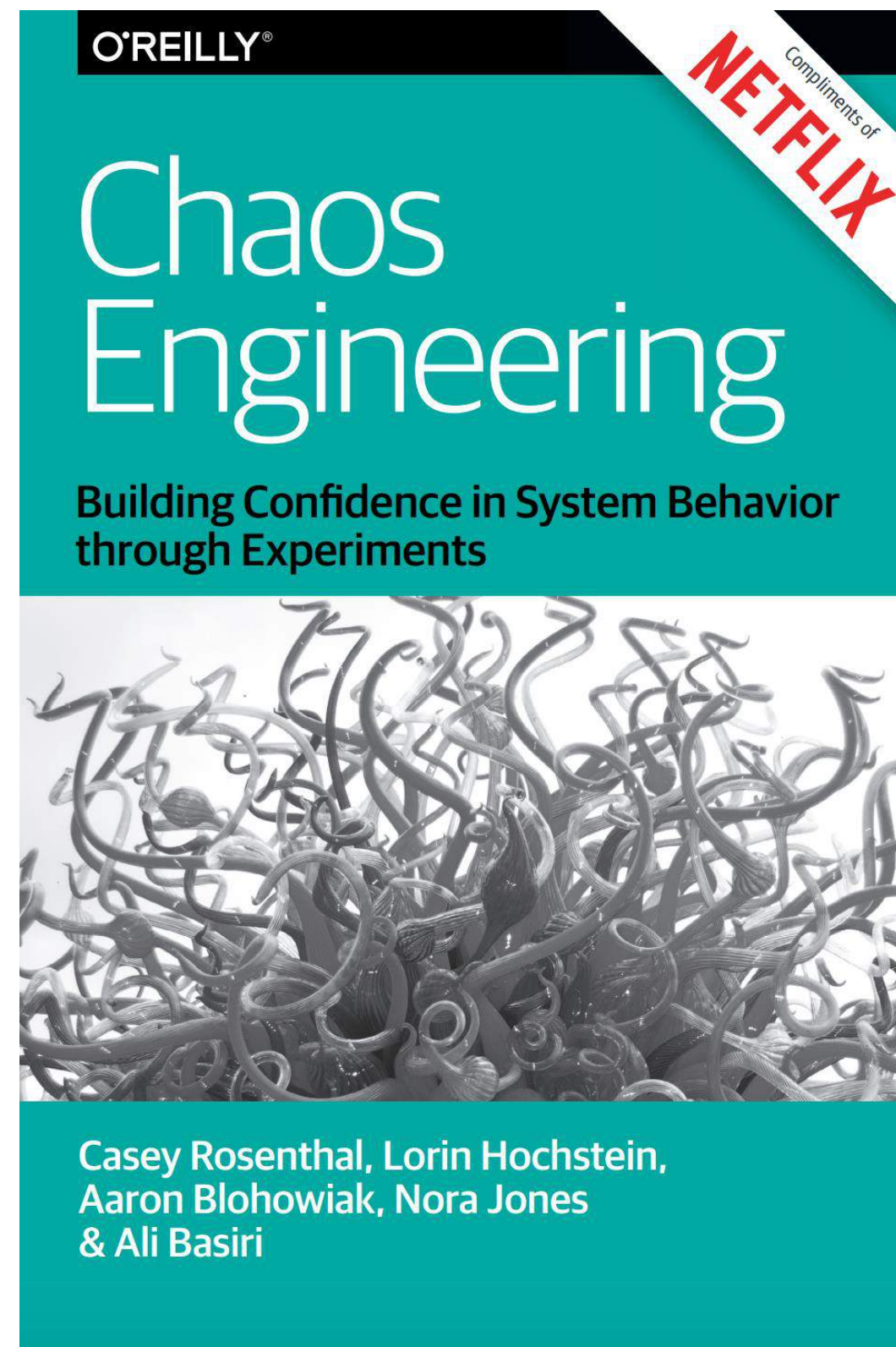
03

AHAS 平台
介绍

04

构建高可用的
分布式服务

混沌工程是什么



混沌工程是在分布式系统上进行实验的学科，旨在提升系统容错性，建立抵御生产环境中发生不可预知问题的信心

What does not kill me, makes me stronger.

--Nietzsche

打不倒我的必使我强大



为什么要实施混沌工程



架构师：验证系统架构的容错能力



开发&运维：提高故障的应急效率



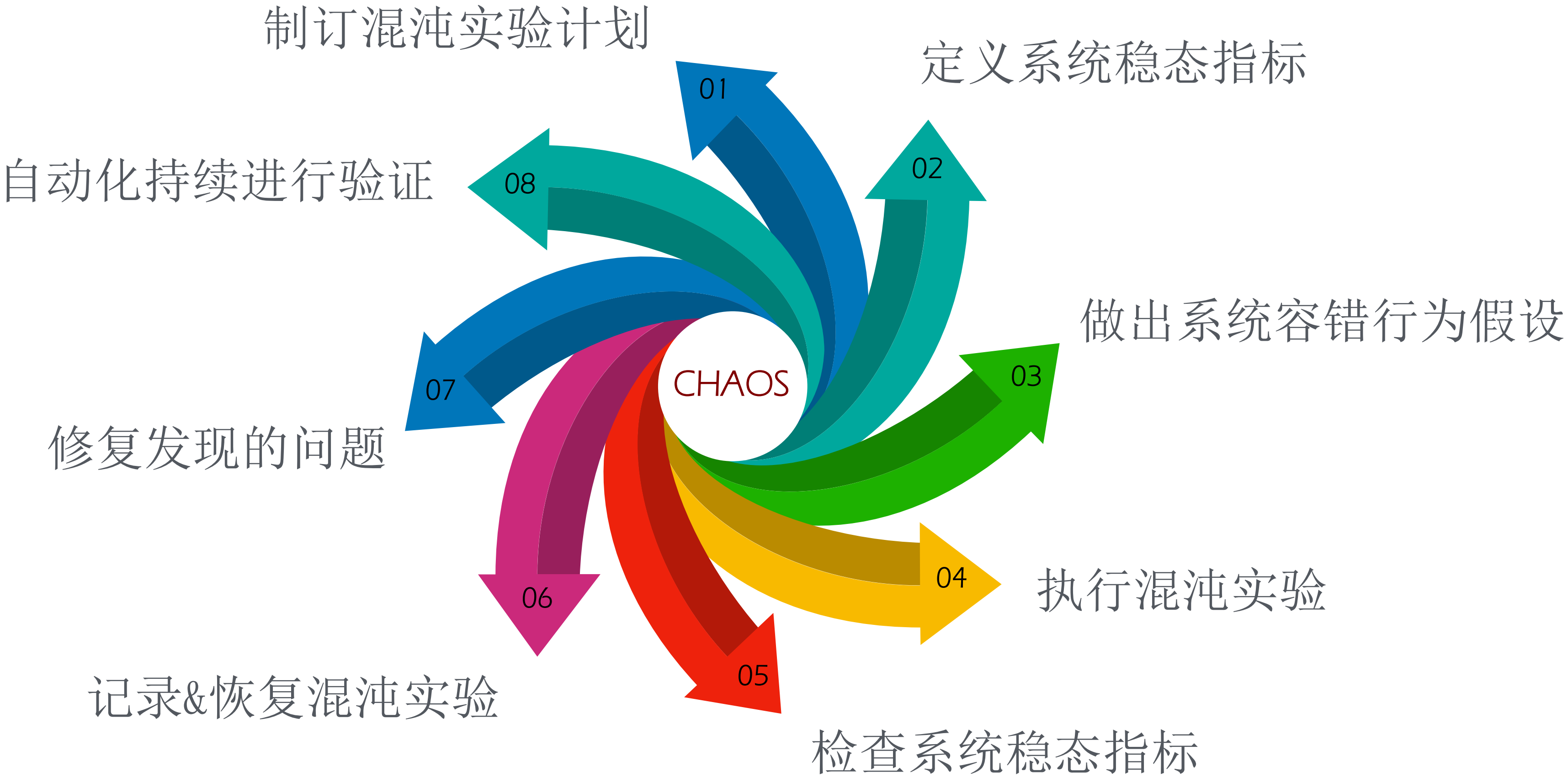
测试：提早暴露线上问题，降低故障复发率



产品&设计：提升客户使用体验



实施混沌工程的步骤



阿里混沌工程技术演进



目录

01



混沌工程
介绍

02



ChaosBlade
介绍

03



AHAS 平台
介绍

04



构建高可用的
分布式服务

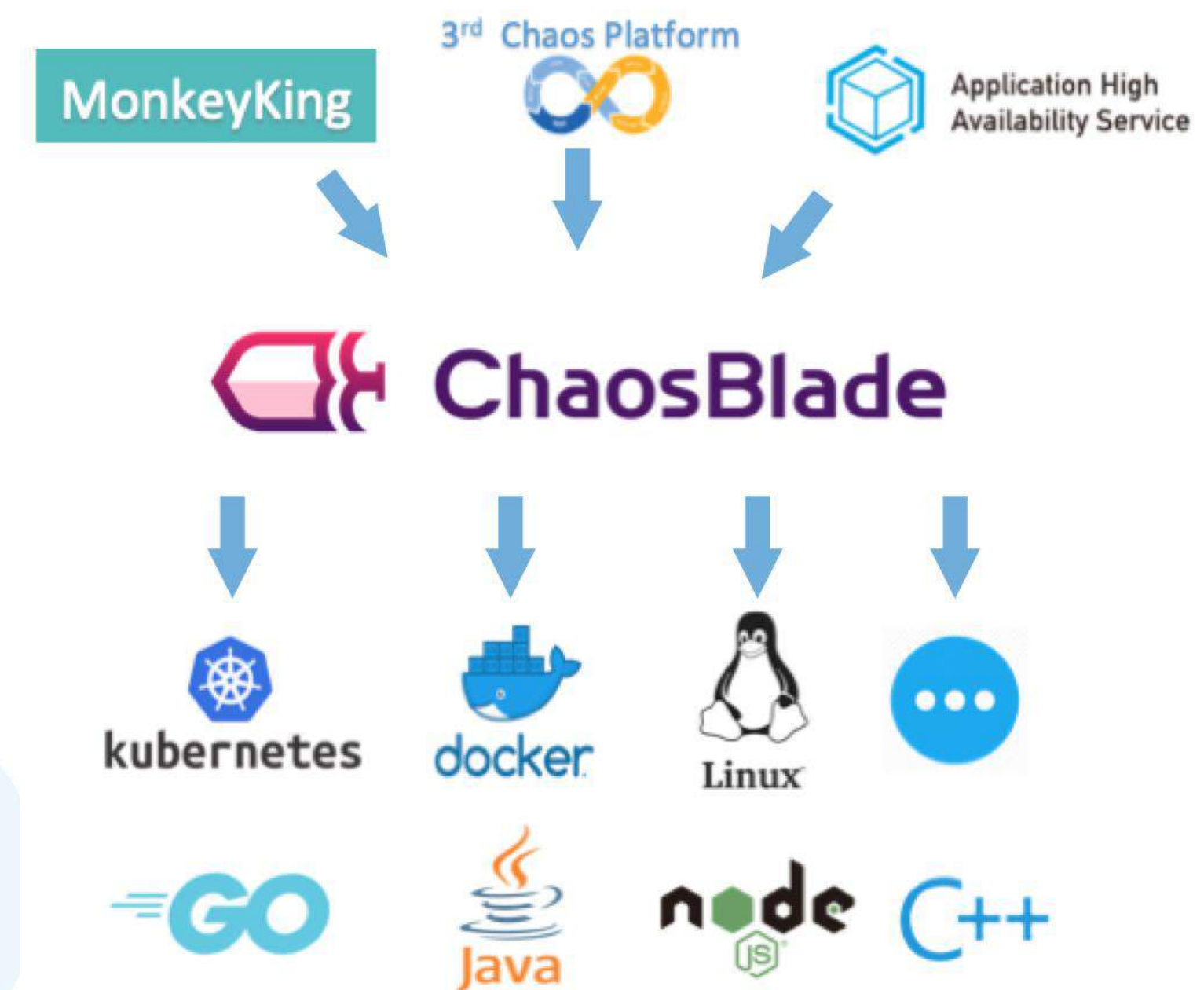
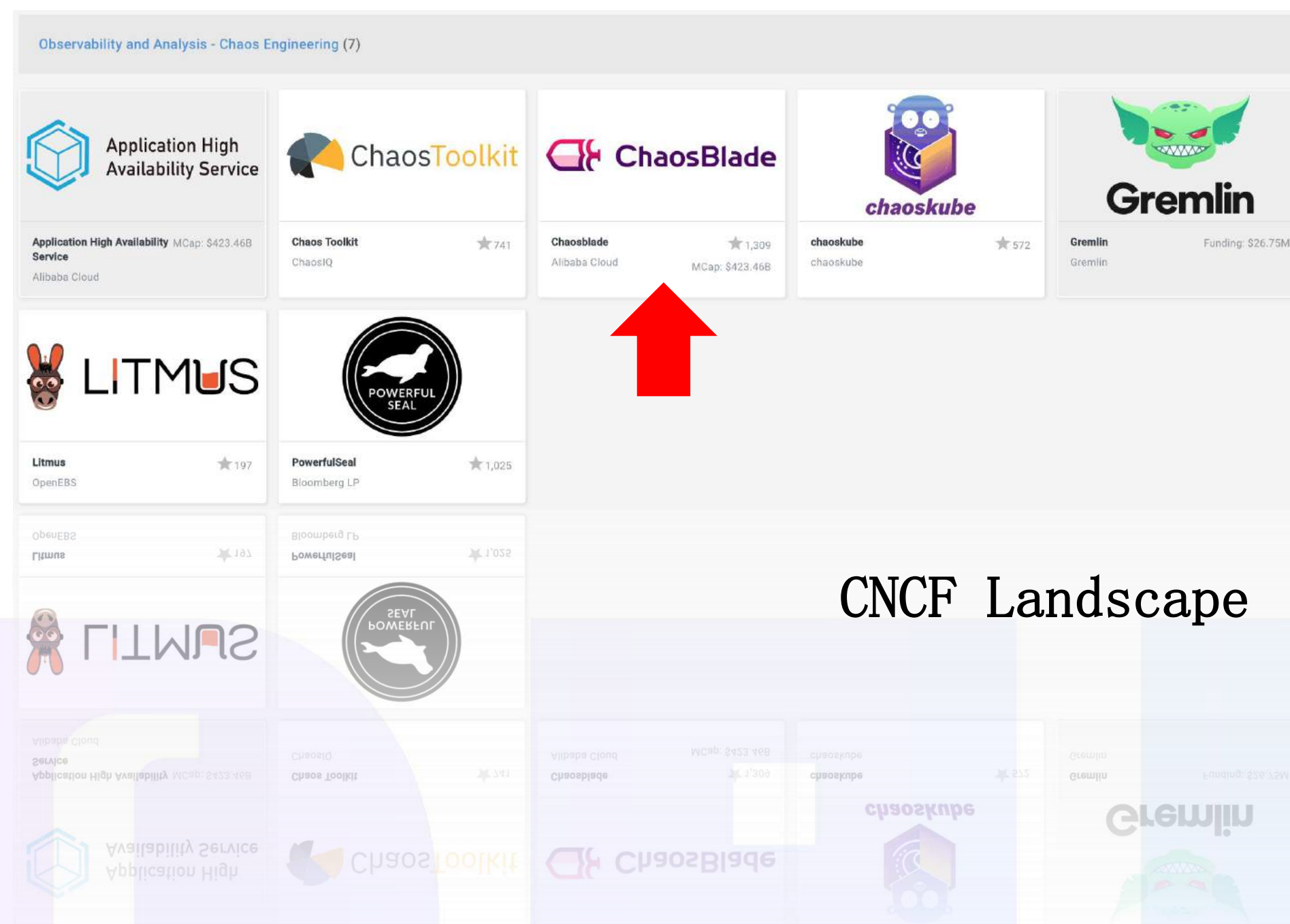


阿里开源项目 ChaosBlade 介绍

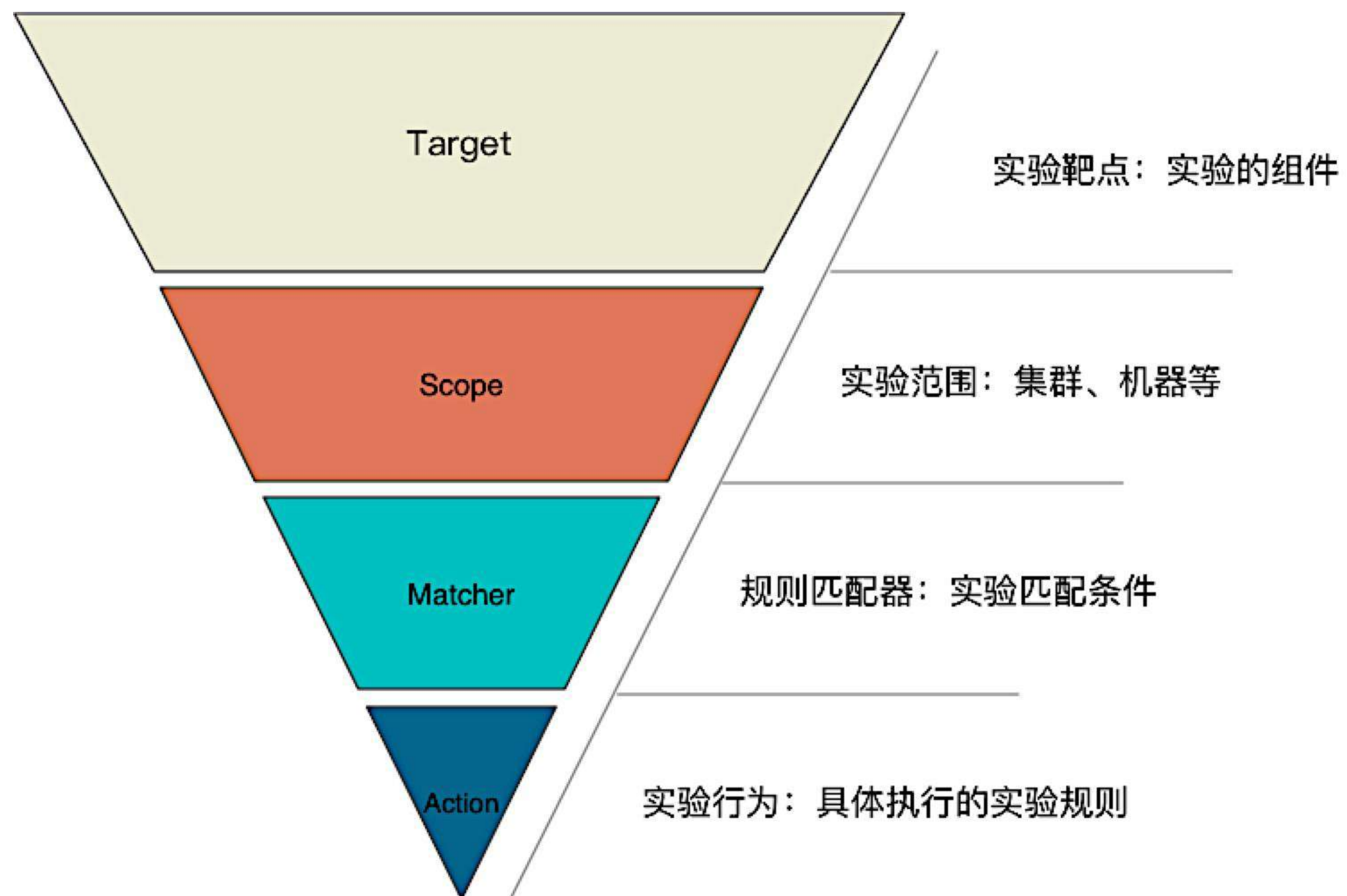
ChaosBlade（混沌之刃）是一款遵循混沌实验模型，简单易用，功能强大的混沌实验工具

特点：

- 场景丰富度高
- 使用简洁，易于理解
- 动态加载，无侵入
- 场景扩展方便



混沌实验模型



blade create cpu fullload

blade destroy 7c1f7afc281482c8

blade create dubbo delay

–time 3000

–service com.alibaba.demo.HelloService

–consumer

目录

01



混沌工程
介绍

02



ChaosBlade
介绍

03



AHAS 平台
介绍

04



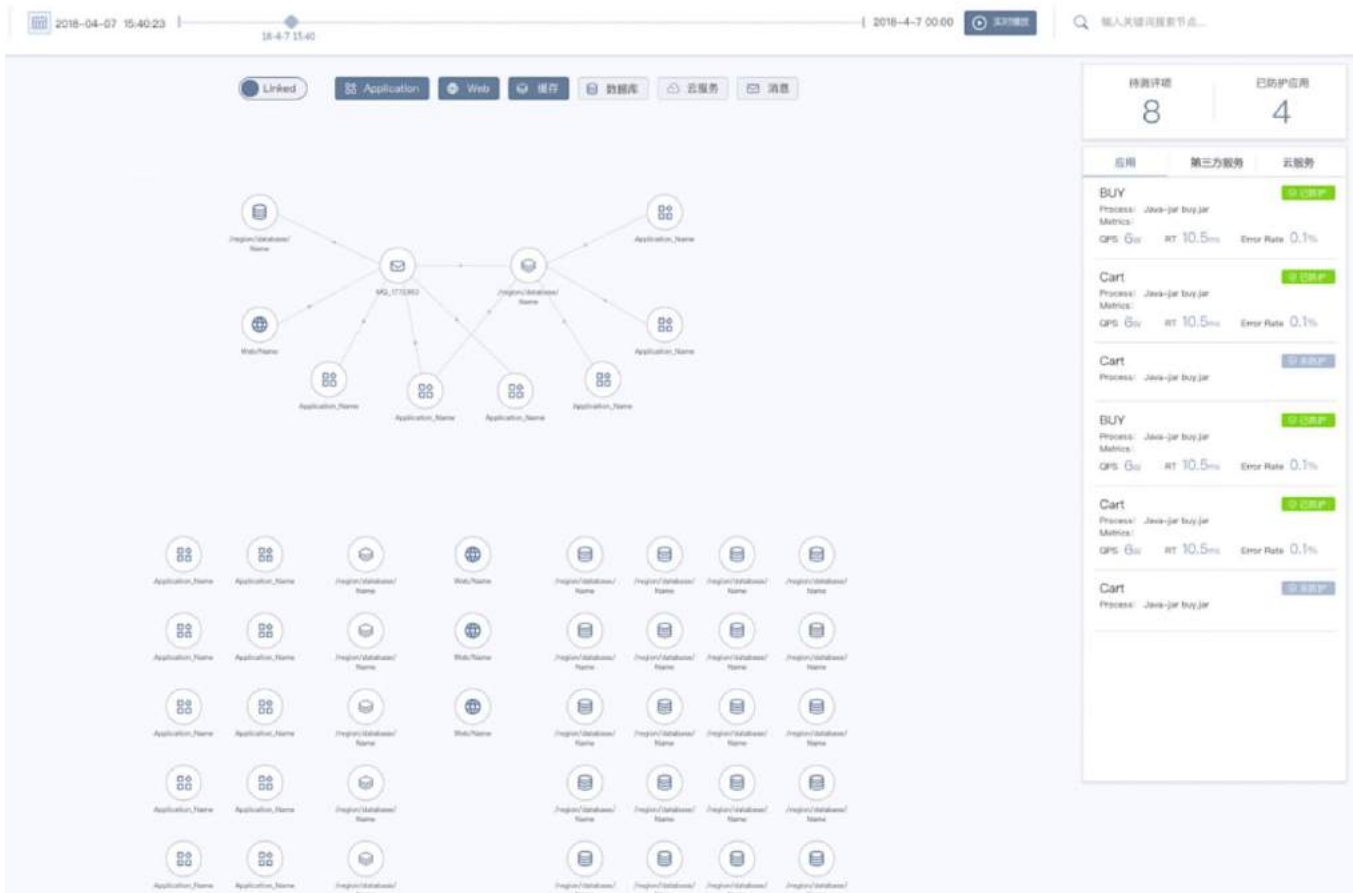
构建高可用的
分布式服务



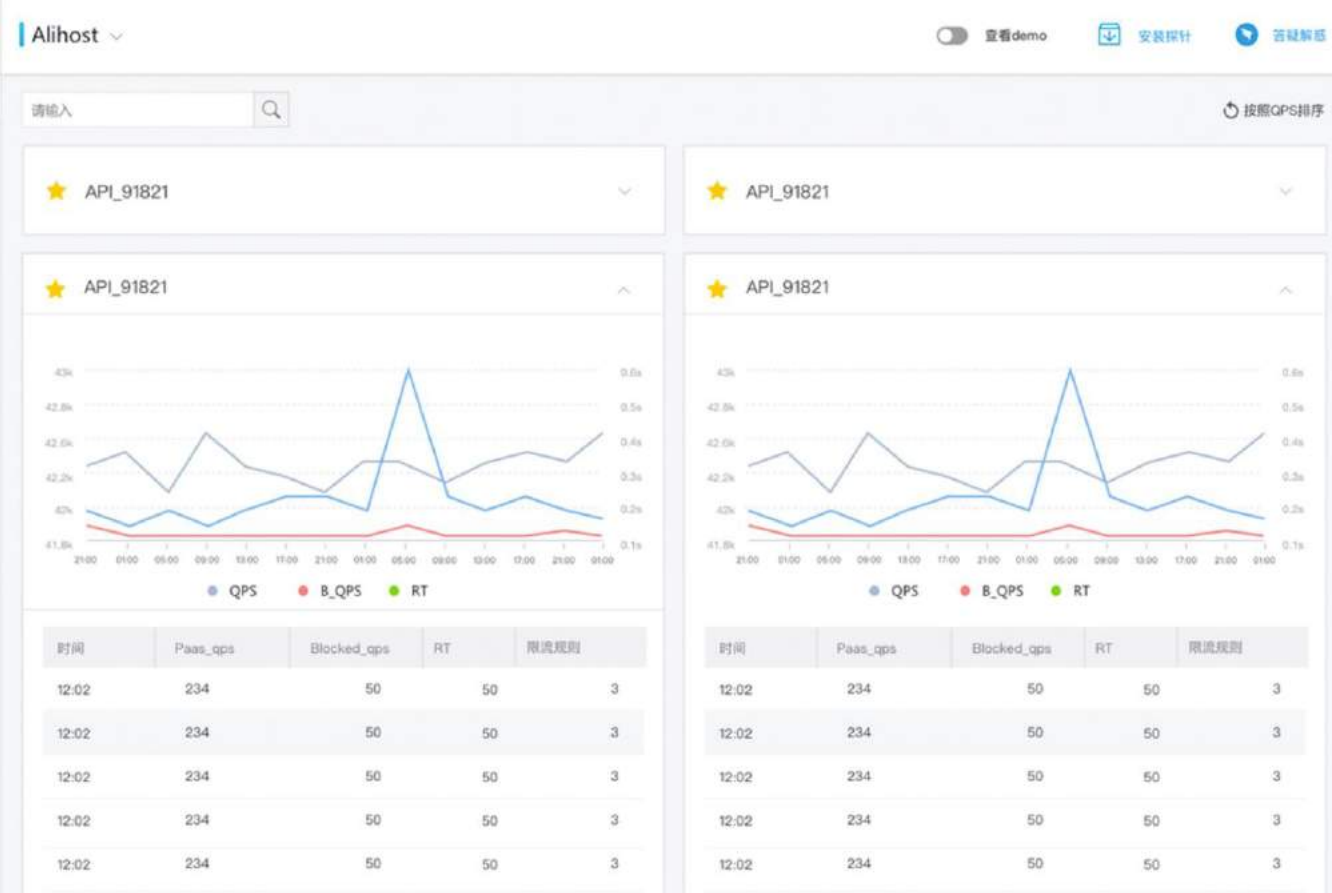
AHAS 平台介绍



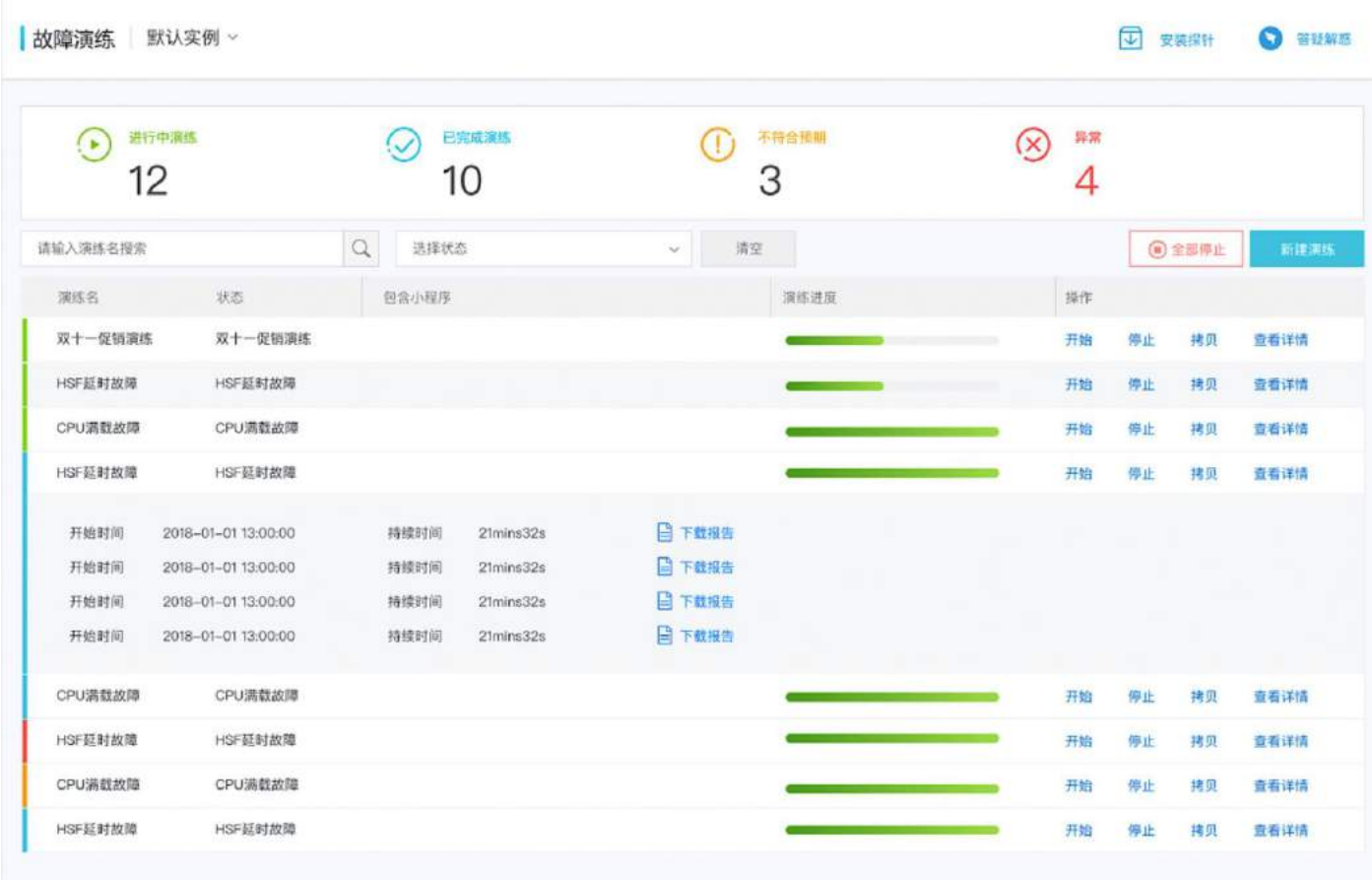
是一款专注于提高系统高可用能力的云产品



架构感知

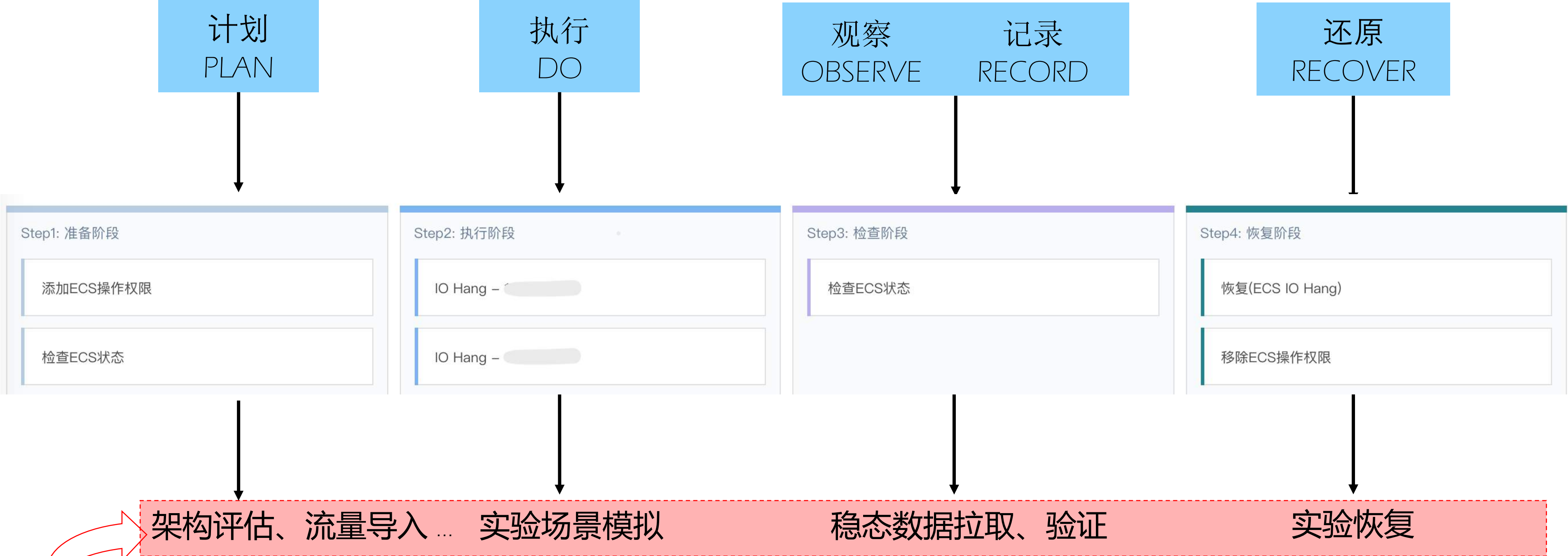


限流降级



故障演练

故障演练平台介绍



混沌工程插件

目录

01



混沌工程
介绍

02



ChaosBlade
介绍

03



AHAS 平台
介绍

04

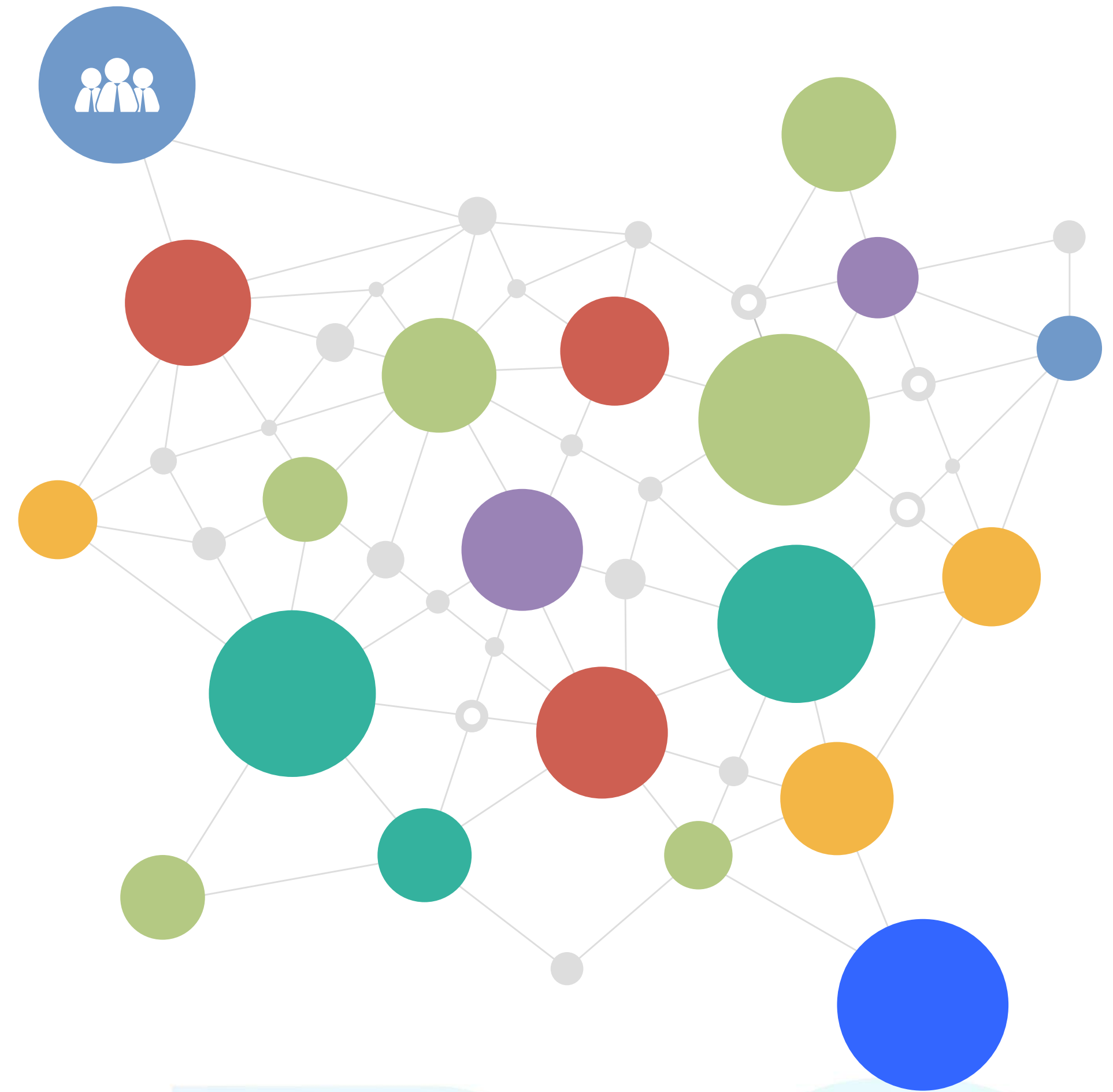


**构建高可用的
分布式服务**



分布式服务系统面临的问题

- 分布式系统日益庞大
很难评估单个故障对整个系统的影响
- 服务间的依赖错综复杂，配置不合理
单个服务不可用可能拖垮整个服务
- 请求链路长，监控告警、日志记录不完善
定位问题难
- 业务、技术迭代速度快
系统稳定性受到更大的挑战



分布式服务系统高可用原则

入口服务

- 负载均衡
- 流量调度
- 请求限流

下游服务

- 超时重试
- 服务降级
- 调用熔断
- 强弱依赖
- 幂等处理
- 最优调用

应用进程

- 资源隔离
- 异步调用
- 热点防护

消息服务

- 异步传递
- 消息分级
- 削峰填谷
- 消息存储

数据缓存

- 热点隔离
- 热点散列
- 主从备份

数据存储

- 读写分离
- 分库分表
- 主从备份
- 一致性保障

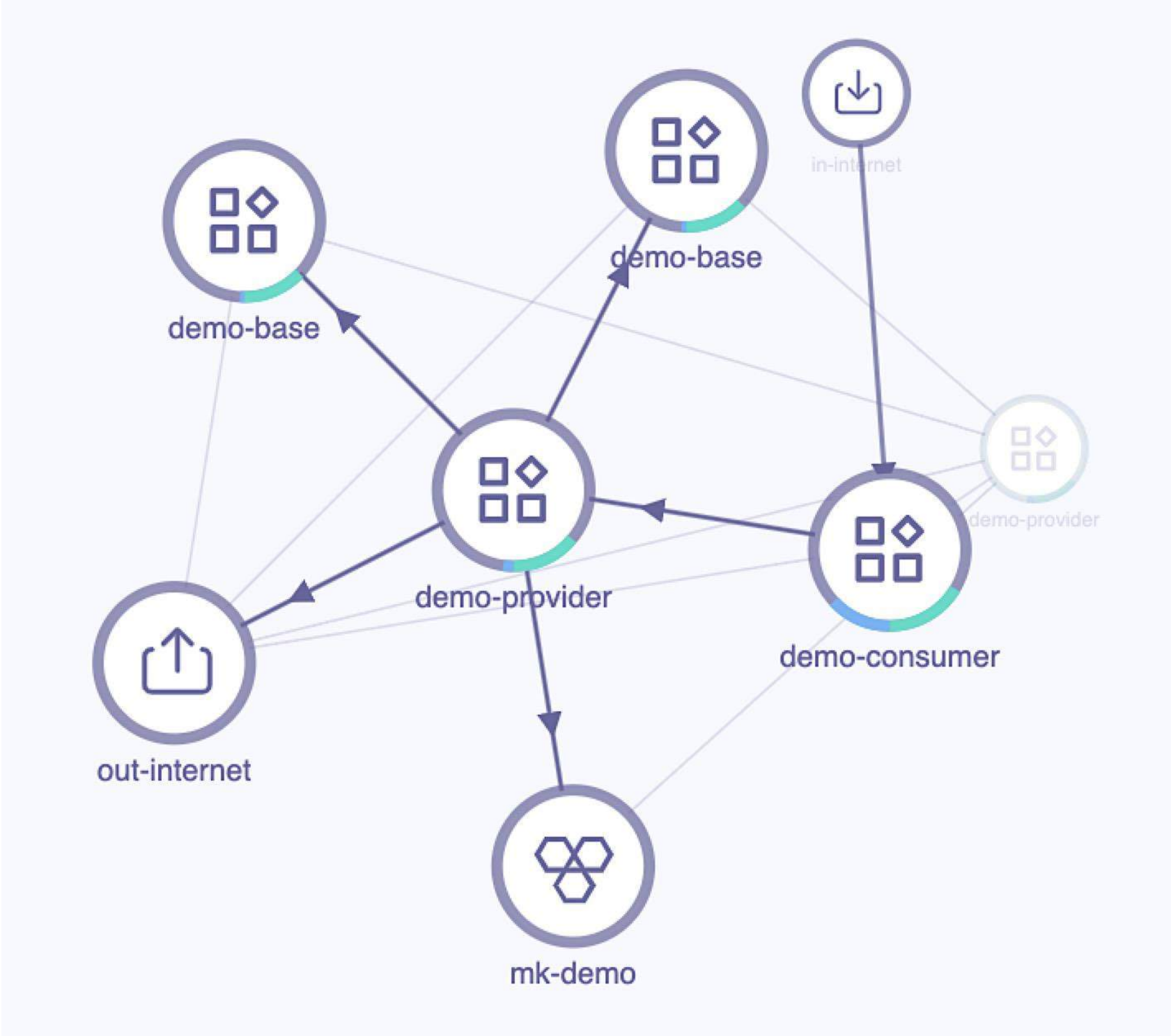
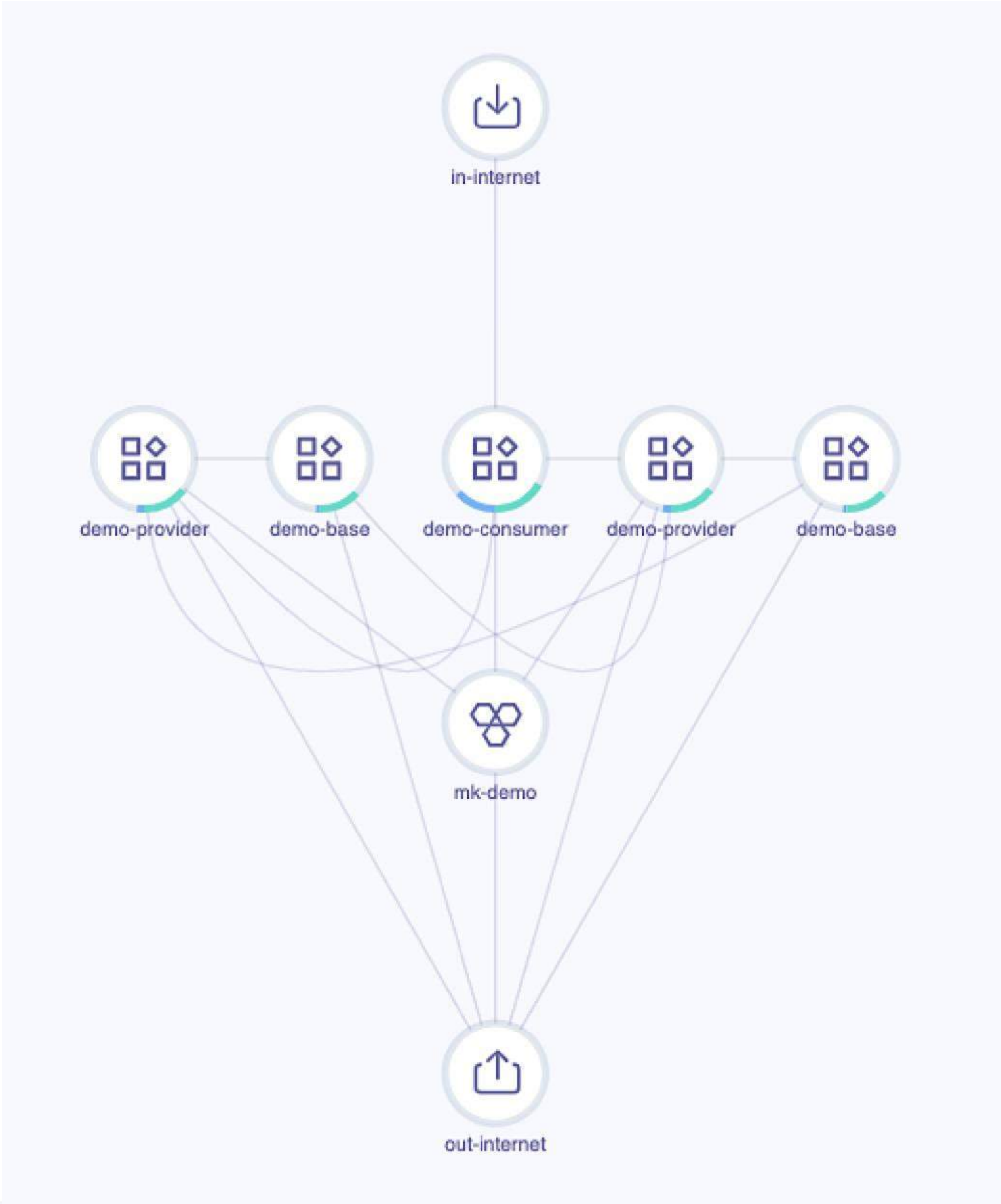
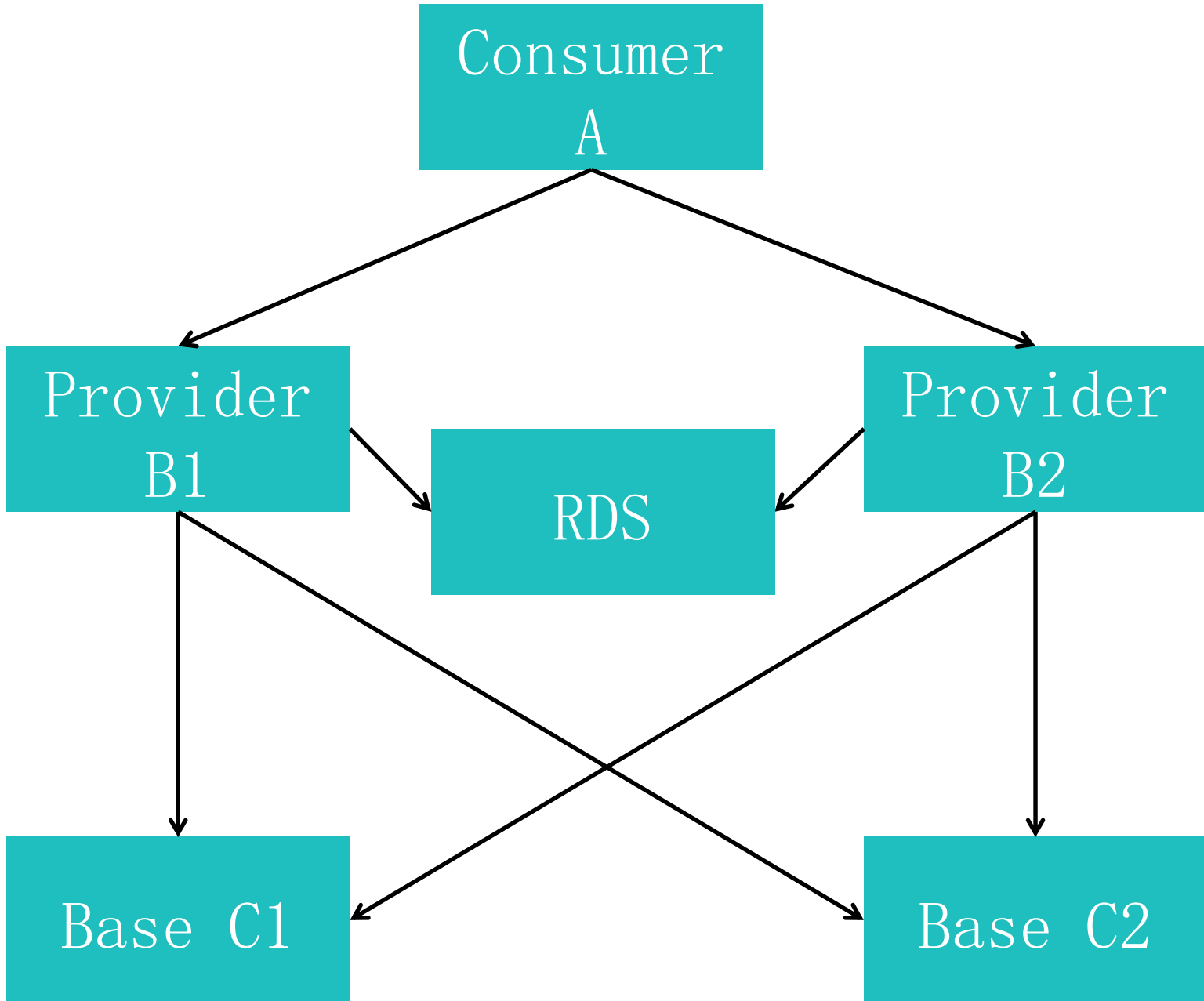
系统运维

- 监控告警
- 日志跟踪
- 健康检查
- 灰度发布
- 发布回滚
- 弹性伸缩
- 容量规划
- 服务治理
- 异地多活

混沌工程

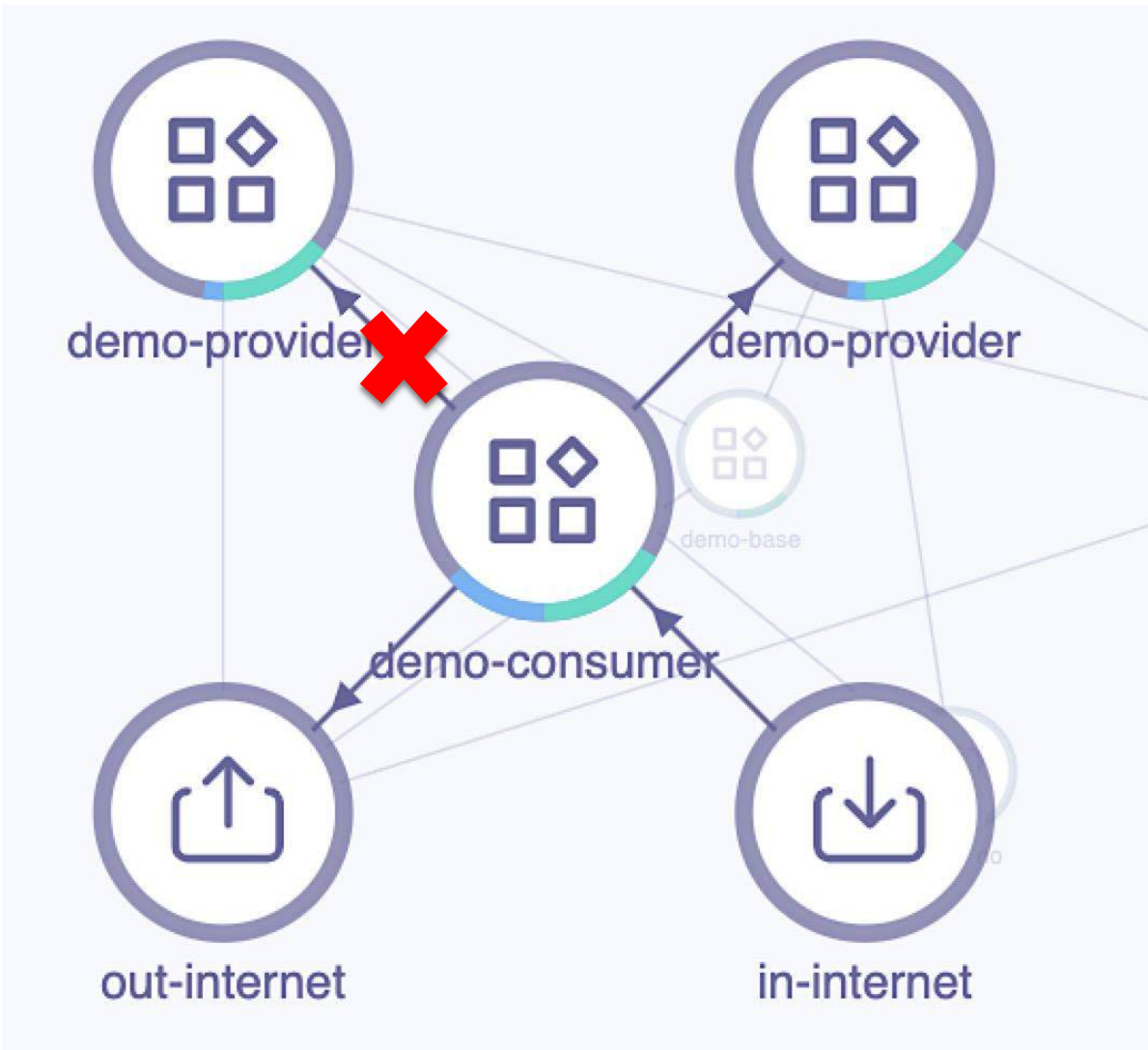
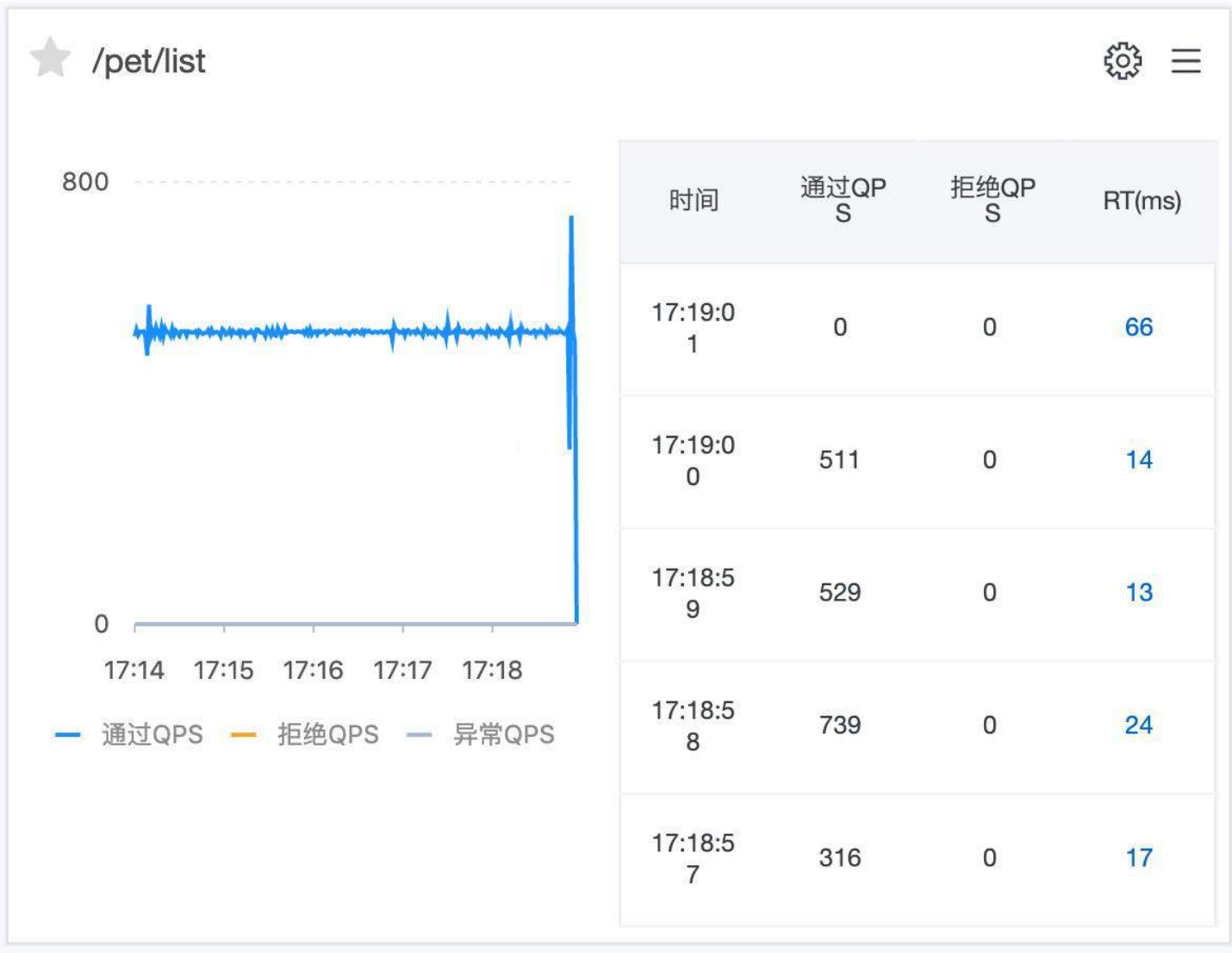


案例 Demo 拓扑图



验证失败重试

场景：调用下游一个服务实例异常
监控指标：QPS，稳态在 510 左右
容错假设：QPS 几乎无变化，请求会路由到另一实例重试
混沌实验：对 demo-provider-1 注入异常故障
监控指标：QPS 几乎不变，符合预期



Provider 异常

演练进度 60%

演练时长 2mins 15s

运行成功 3

不符合预期 0

异常 0

Step1: 准备阶段

安装javaAgent

检查探针版本

Step2: 执行阶段

Dubbo抛自定义异常

Step3: 检查阶段

Dubbo抛自定义异常

appname

当前应用作为服务提供方

service

version

当前应用作为服务消费方

方法名, 无需填写参数, 如...

抛出的异常, java.lang.Exce...

应用进程关键词,通过ps -ef...

机器信息

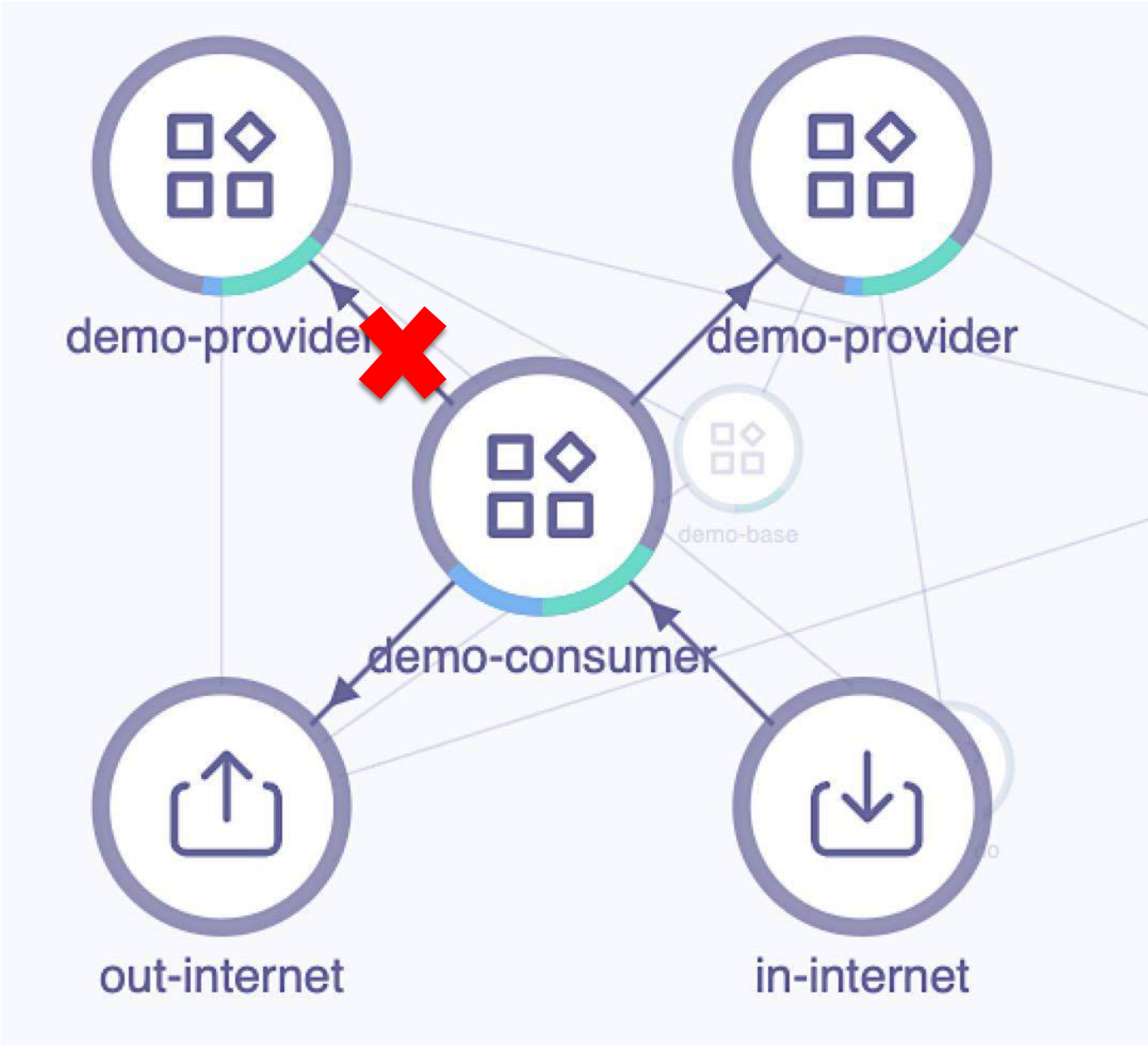
172.16.93.146

成功

查看详情

验证异常实例隔离

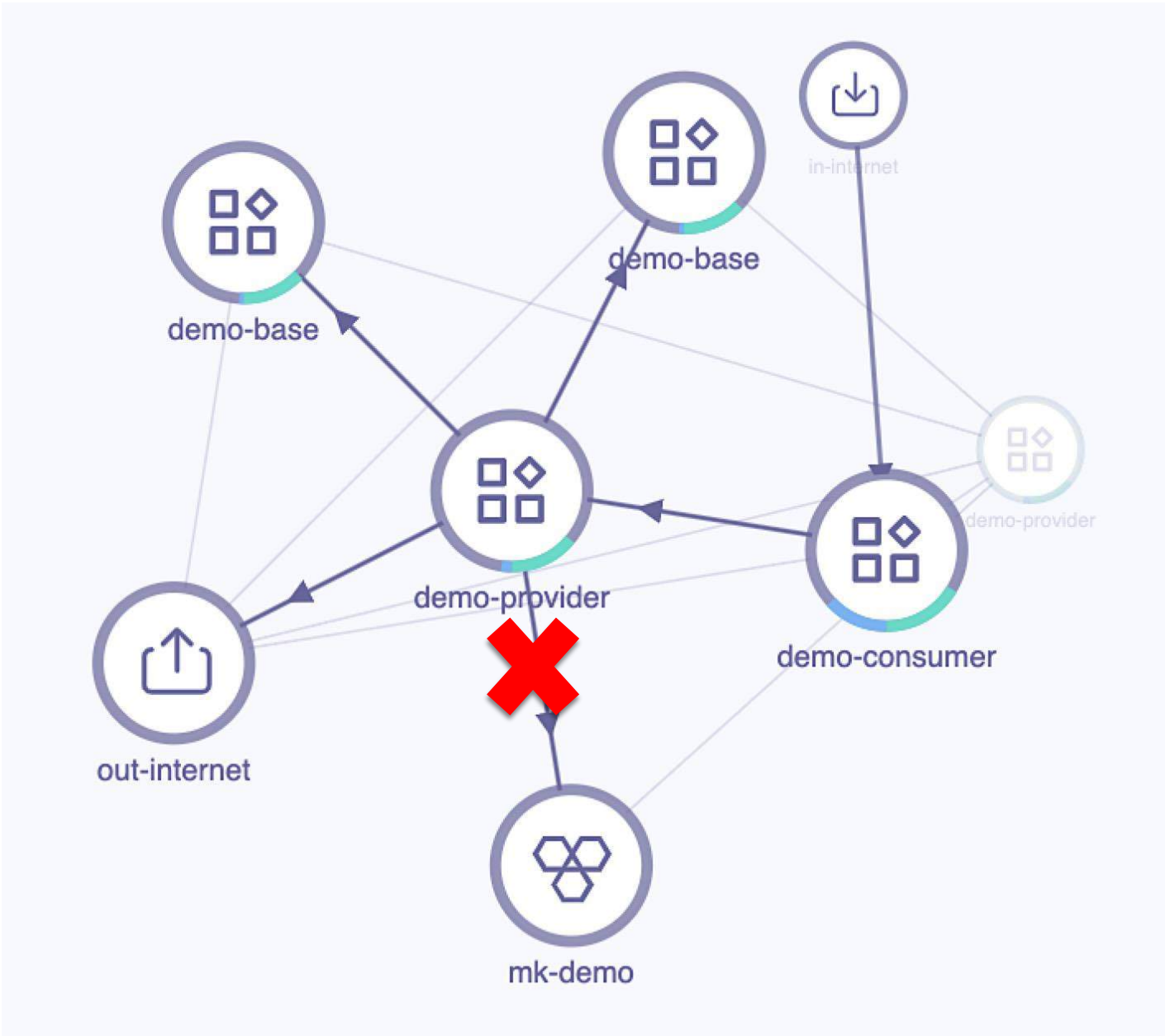
- 场景：下游一个服务实例出现延迟
- 监控指标：QPS，稳态在 510 左右
- 容错假设：QPS 会出现几秒的下跌，但很快恢复；系统会自动隔离或下线出问题服务实例，防止请求路由到此实例
- 混沌实验：对 demo-provider-1 注入延迟故障
- 监控指标：QPS 下跌到 40，不会自动恢复，不符合预期
- 业务方应急处理：下线出问题的实例，QPS 恢复
- 问题记录：系统缺失服务质量检查，不能对异常服务实例做隔离



验证监控告警

- 场景：数据库调用延迟
- 监控指标：慢 SQL 数，告警信息
- 期望假设：慢 SQL 数增加，钉钉群收到慢 SQL 告警
- 混沌实验：对 demo-provider 注入调用 mk-demo 数据库延迟故障
- 监控指标：慢 SQL 数增加，钉钉群收到告警，符合预期
- 问题排查：通过 ARMS 慢调用链路排查

备注：以上告警和链路跟踪来自于阿里云 ARMS 产品



```
blade create mysql delay
--time 600
--database demo
--table d_discount
--sqltype select
--effect-percent 50
```

故障故障 robot

报警名称:monkeyking-demo-provider-慢 SQL 告警

筛选条件:

报警时间: 01:18:02

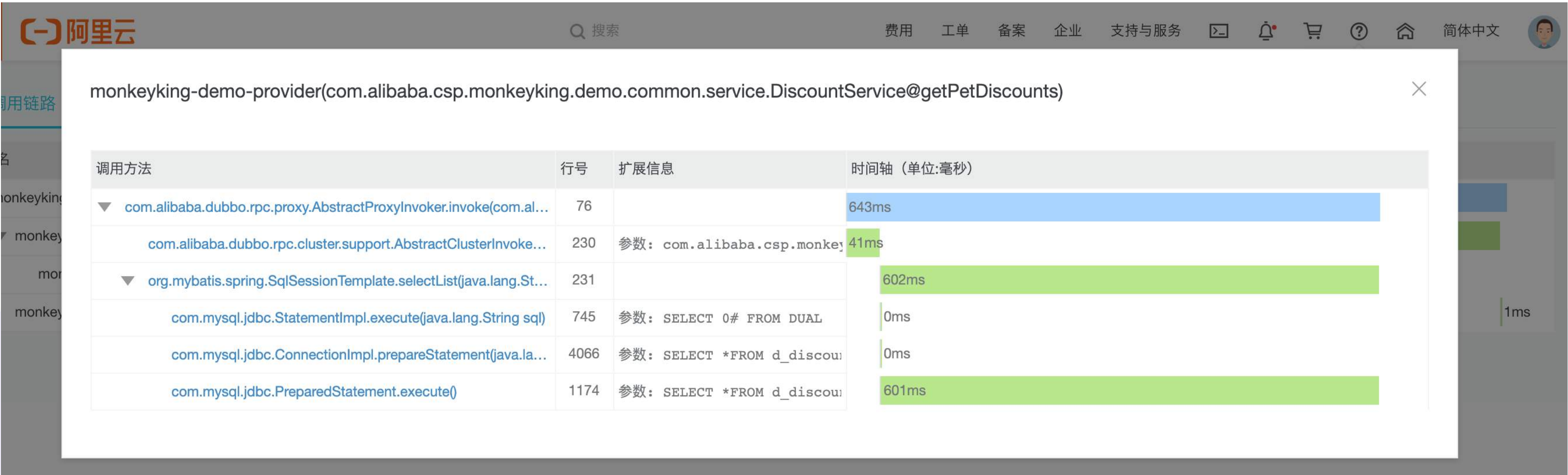
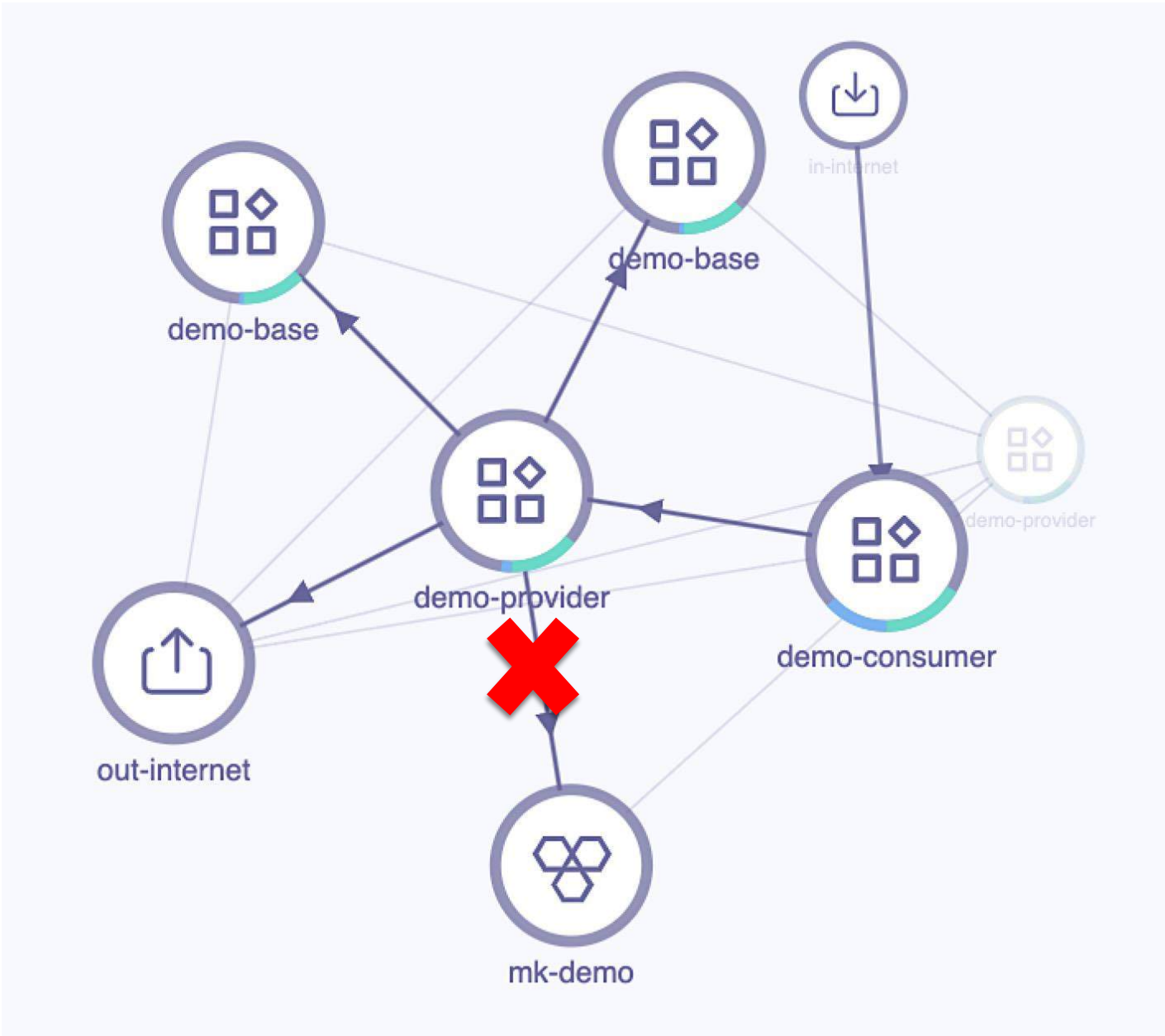
报警内容: 最近1分钟数据库调用响应时间_ms最大值 300.89 大于等于300

注意: 该报警未收到恢复邮件之前, 正在持续报警中, 24小时后会再次提醒您!

验证监控告警

- 场景：数据库调用延迟
- 监控指标：慢 SQL 数，告警信息
- 期望假设：慢 SQL 数增加，钉钉群收到慢 SQL 告警
- 混沌实验：对 demo-provider 注入调用 mk-demo 数据库延迟故障
- 监控指标：慢 SQL 数增加，钉钉群收到告警，符合预期
- 问题排查：通过 ARMS 慢调用链路排查

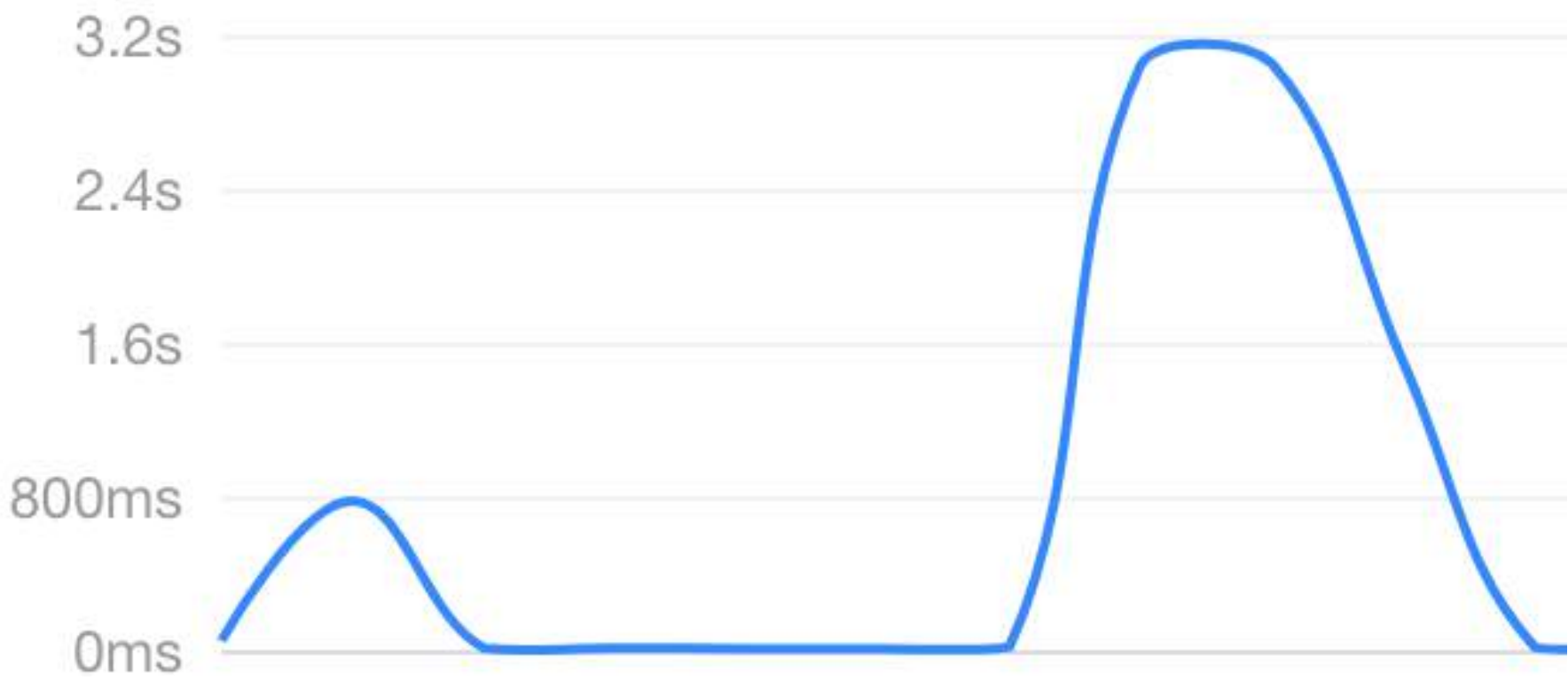
备注：以上告警和链路跟踪来自于阿里云 ARMS 产品



验证降级熔断

- 场景：上游调用下游服务超时
- 监控指标：RT 平均 19ms
- 容错假设：RT 短暂增加，但很快恢复
- 混沌实验：对 demo-provider 两个服务实例注入延迟故障
- 监控指标：RT 增加到 3s，不会自动恢复，不符合预期
- 业务方应急处理：配置熔断策略，调用下游服务快速失败

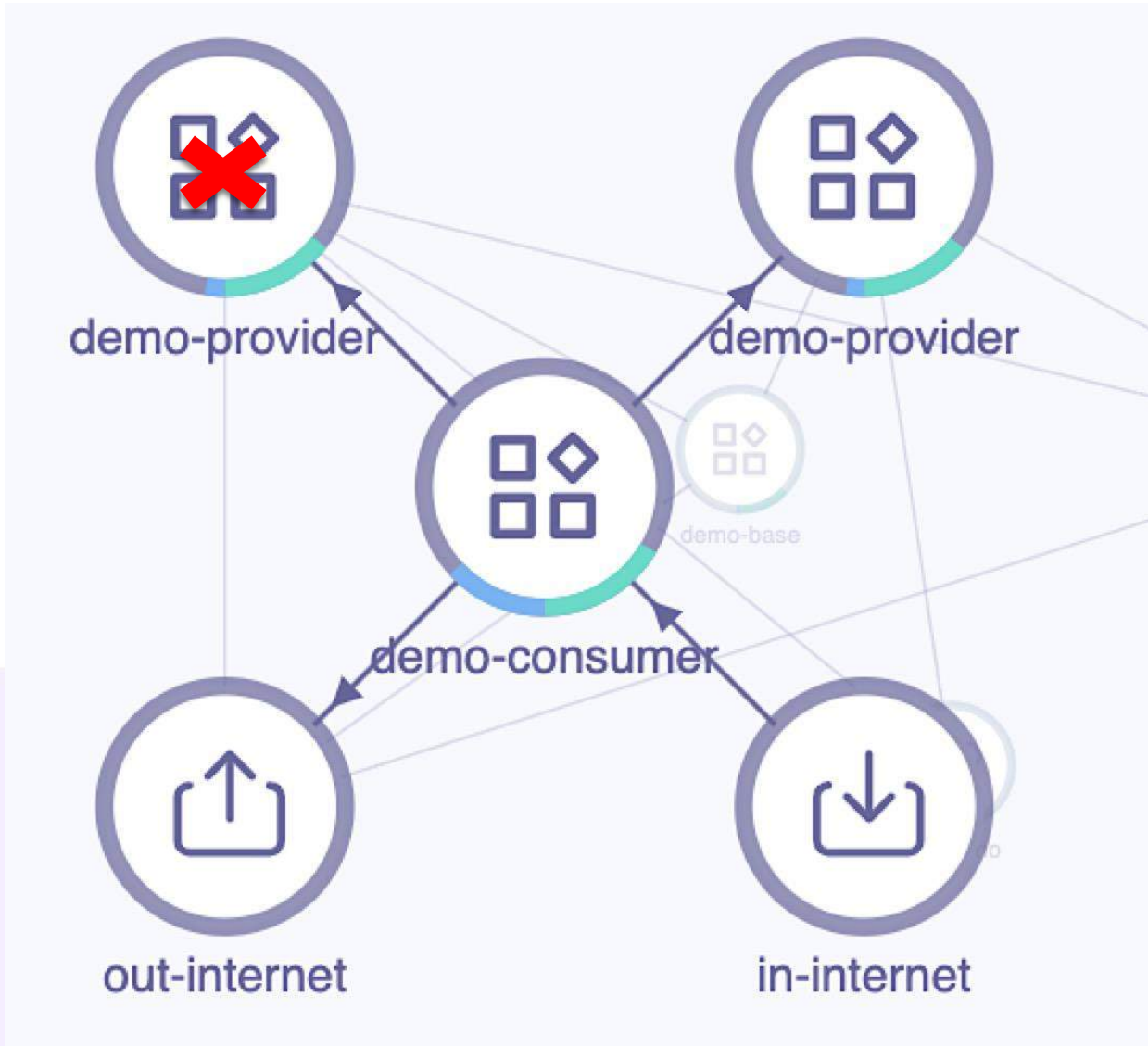
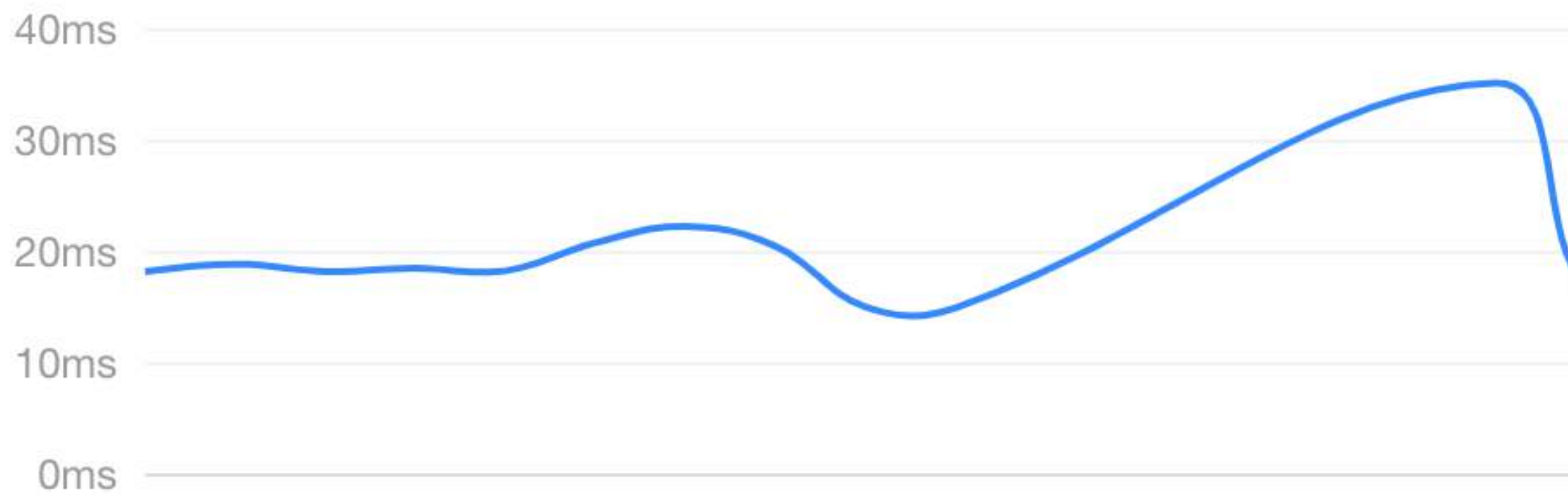
应用提供服务平均响应时间 / 每分钟



验证流量调度

- 场景：下游服务一个实例 CPU 满载
- 监控指标：RT 平均 19ms
- 容错假设：RT 会升高，但很快恢复；系统会将流量调度到其他服务实例
- 混沌实验：对 demo-provider-1 服务实例注入 CPU 满载故障
- 监控指标：RT 升高到 35ms，不会自动恢复，不符合预期
- 业务方应急处理：下线出问题的实例，RT 恢复
- 问题记录：系统缺失服务质量检查，不能对异常服务实例做隔离

应用提供服务平均响应时间 / 每分钟



总结回顾

- 混沌工程是一种主动防御的稳定性手段，体现了反脆弱的思想
- 选择合适的工具和平台，控制演练风险，实现持续的自动化演练
- 实施混沌工程不能只是把故障制造出来，需要有明确的驱动目标
- 通过混沌工程发现系统存在的问题，持续提升分布式系统韧性



口令：CHAOS





1

Thank you !

01010101