

# 开源混沌工程工具 ChaosBlade 介绍与实践

穹谷

# 目录

CONTENT

混沌工程介绍

ChaosBlade 项目介绍

ChaosBlade 工具介绍

ChaosBlade 场景实践

AHAS 平台介绍

# 混沌工程介绍

## 什么是混沌工程



奥运会全球指定云服务商

*Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production.*

*-- Principles of Chaos*

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，建立系统抵御生产环境中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。减少故障的最好方法就是让问题经常性的发生。在可控范围或环境下，通过不断重复失败过程，持续提升系统的稳定性和容错能力。



# 混沌工程介绍

## 混沌工程原则



奥运会全球指定云服务商

### 建立一个围绕稳定状态行为的假说

- ◆ 关注可测量输出，而不是系统内部属性。
- ◆ 短时间内的度量结果，代表了系统的稳定状态。
- ◆ 验证系统是否工作，而不是如何工作。

### 多样化真实世界的事件

- ◆ 混沌变量反映了现实世界中的事件。
- ◆ 通过潜在影响或预估频率排定事件的优先级。
- ◆ 任何能够破坏稳态的事件都是混沌实验中的一个潜在变量。

### 在生产环境中运行实验

- ◆ 系统的行为会根据环境和流量模式有所不同。
- ◆ 为了保证系统执行方式的真实性与当前部署系统的相关性，混沌工程强烈推荐直接采用生产环境流量进行实验。

### 持续自动化运行实验

- ◆ 手动运行实验是劳动密集型的，最终是不可持续的，所以我们要把实验自动化并持续运行。
- ◆ 混沌工程要在系统中构建自动化的编排和分析。

### 最小化爆炸半径

- ◆ 在生产中进行试验可能会造成不必要的客户投诉。但混沌工程师的责任和义务是确保这些后续影响最小化且被考虑到。



# 混沌工程介绍

## 混沌工程的应用场景

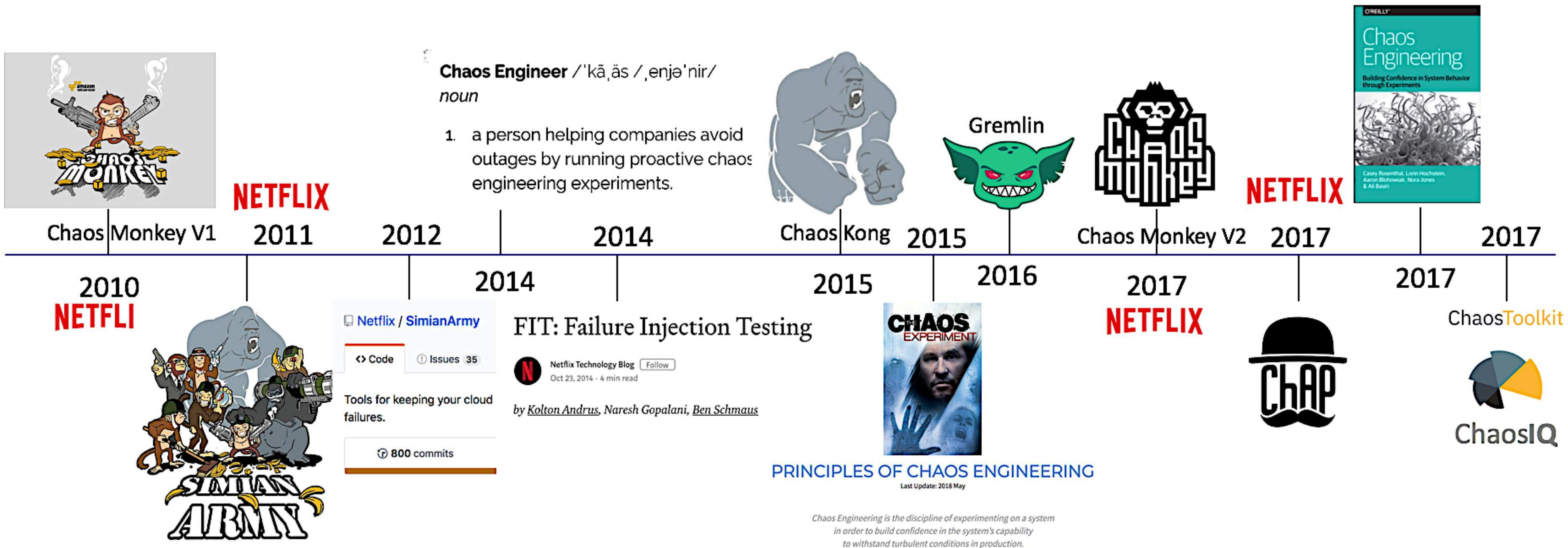
- 提升系统的容错能力，提高稳定性
- 评估系统容灾红线
- 验证云服务、云资源的动态扩容能力
- 验证监控、告警的有效性和问题处理流程是否完善
- 故障突袭，提升相关人员定位、恢复问题的能力
- 其他

# 混沌工程介绍

## 国外混沌工程演进线



奥运会全球指定云服务商





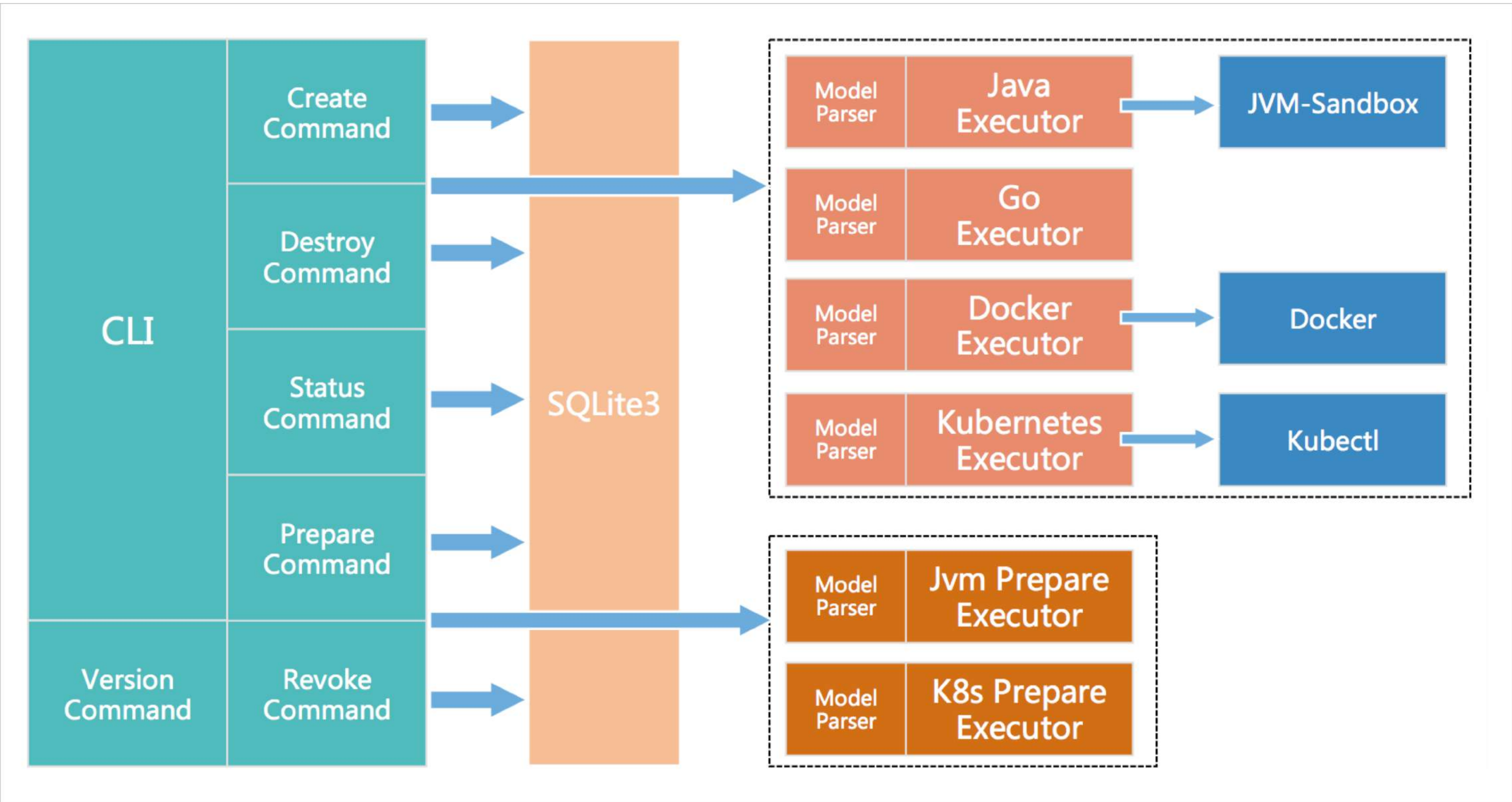
# 混沌工程介绍

阿里巴巴混沌工程演进线



# ChaosBlade 项目介绍

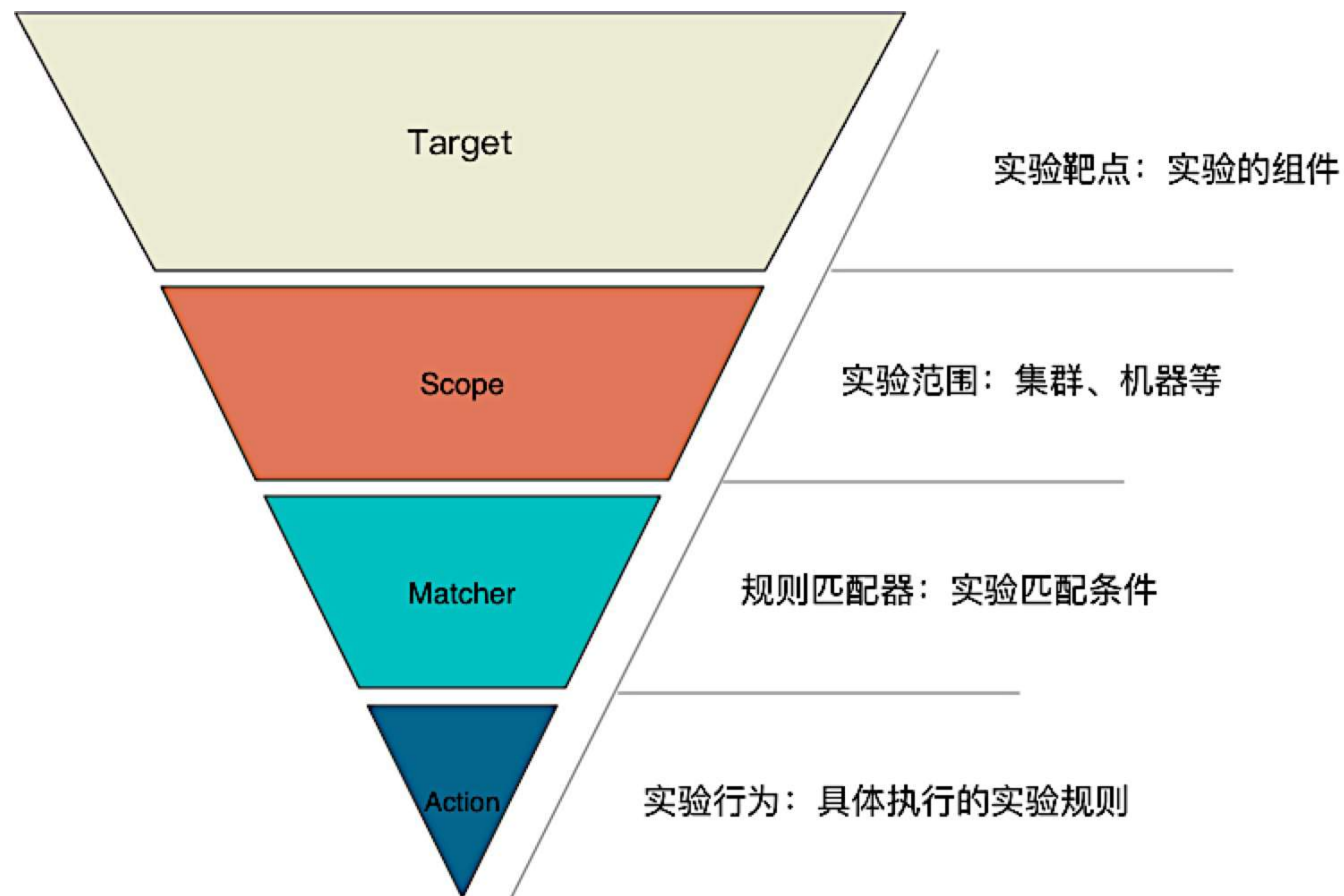
是一款遵循混沌工程实验原理，提供丰富故障场景实现，帮助分布式系统提升容错性和可恢复性的混沌工程工具，它的特点是操作简洁、无侵入、扩展性强。Github 地址：<https://github.com/chaosblade-io>





# ChaosBlade 项目介绍

## 混沌实验模型



混沌实验前要明确的问题：

- 对什么做混沌实验
- 混沌实验实施的范围是是什么
- 实验生效的匹配条件有哪些
- 具体实施什么实验

举例：一个运行在 10.0.0.1 机器上的应用，调用

com.example.HelloService@1.0.0  
dubbo 服务延迟 3s

```
blade create dubbo delay --time 3000 --consumer --service com.example.HelloService --version 1.0.0
```

# ChaosBlade 工具介绍

## 工具特点&支持的实验

- 场景丰富度高
- 使用简洁，易于理解
- 动态加载，无入侵
- 场景扩展方便
- 支持运行时长设置
- CPU 满载
- 网络延迟、丢包、屏蔽
- 域名屏蔽
- 磁盘填充，磁盘 IO 读写负载
- 杀进程
- 删容器、Pod
- Java 应用或组件内延迟、抛异常、指定返回值



奥运会全球指定云服务商



# ChaosBlade 工具介绍

## 工具使用

~/chaosblade » blade

An easy to use and powerful chaos engineering experiment toolkit

Usage:

blade [command]

Available Commands:

create	Create a chaos engineering experiment
destroy	Destroy a chaos experiment
help	Help about any command
prepare	Prepare to experiment
query	Query the parameter values required for chaos experiments
revoke	Undo chaos engineering experiment preparation
status	Query preparation stage or experiment status
version	Print version info

Flags:

-d, --debug	Set client to DEBUG mode
-h, --help	help for blade

Use "blade [command] --help" for more information about a command.



奥运会全球指定云服务商

- blade create cpu fullload
- blade destroy 7c1f7afc281482c8
- blade status 7c1f7afc281482c8
- blade prepare jvm --process dubbo.consumer
- blade status --type prepare
- blade create dubbo delay --time 3000 --service com.alibaba.demo.HelloService --methodname hello --consumer --process dubbo.consumer
- blade create jvm throwCustomException --exception java.lang.Exception --classname com.example.controller.DubboController --methodname hello
- blade revoke e669d57f079a00cc
- blade query disk mount-point

# ChaosBlade 工具介绍

## 操作演示



体验 demo 详见: <https://github.com/chaosblade-io/chaosblade#experience-demo>

Demo 必要条件: 安装 docker

## 操作步骤:

- 下载 demo 镜像: `docker pull registry.cn-hangzhou.aliyuncs.com/chaosblade/chaosblade-demo:latest`
- 运行 demo: `docker run -it registry.cn-hangzhou.aliyuncs.com/chaosblade/chaosblade-demo:latest`
- 阅读 demo 中的 README.txt 进行操作

5 分钟快速上手指南: <https://github.com/chaosblade-io/chaosblade/wiki/%E6%96%B0%E6%89%8B%E6%8C%87%E5%8D%97>



# ChaosBlade 场景实践

## 场景一：验证服务隔离能力

### 演练场景：

对 Dubbo 服务提供方 Provider A 注入延迟，验证问题实例隔离能力

### 演练目标：

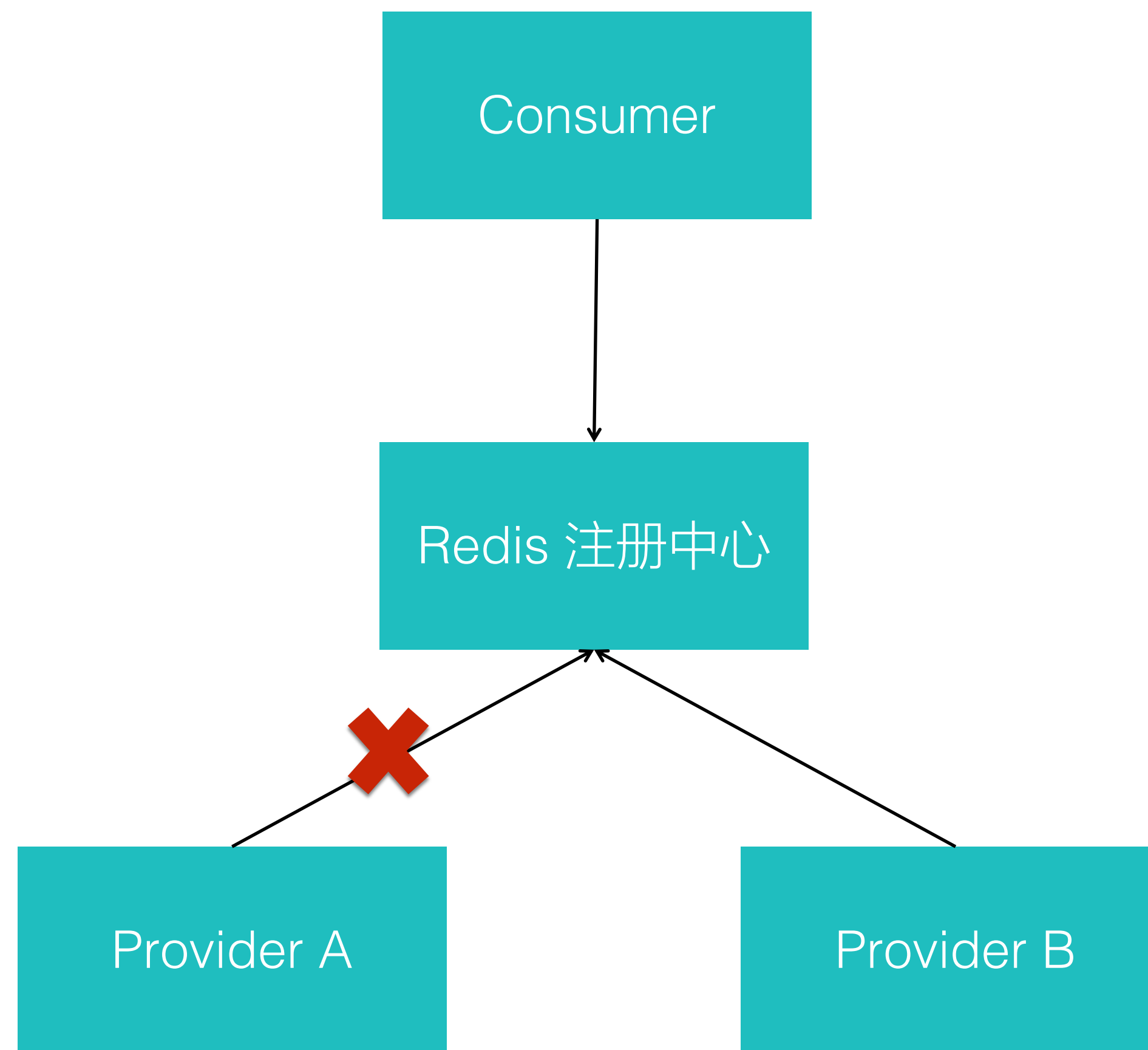
下线 Provider A，所有请求都访问 Provider B，减少对上层服务的影响

### 容灾方案：

检测 Provider A 服务质量，自动下线 Provider A 实例，Dubbo 将所有请求发送给 Provider B

### blade 相关命令：

```
blade prepare jvm --process dubbo-provider  
blade c dubbo delay --provider --service  
com.alibaba.demo.HelloService --methodname  
hello --time 5000 --process dubbo-provider
```



# ChaosBlade 场景实践

## 场景二：验证熔断降级

### 演练场景：

对 Dubbo 服务提供方 Provider A 和 Provider B 注入延迟，模拟服务提供方访问慢的场景，验证上层服务熔断降级的能力

### 演练目标：

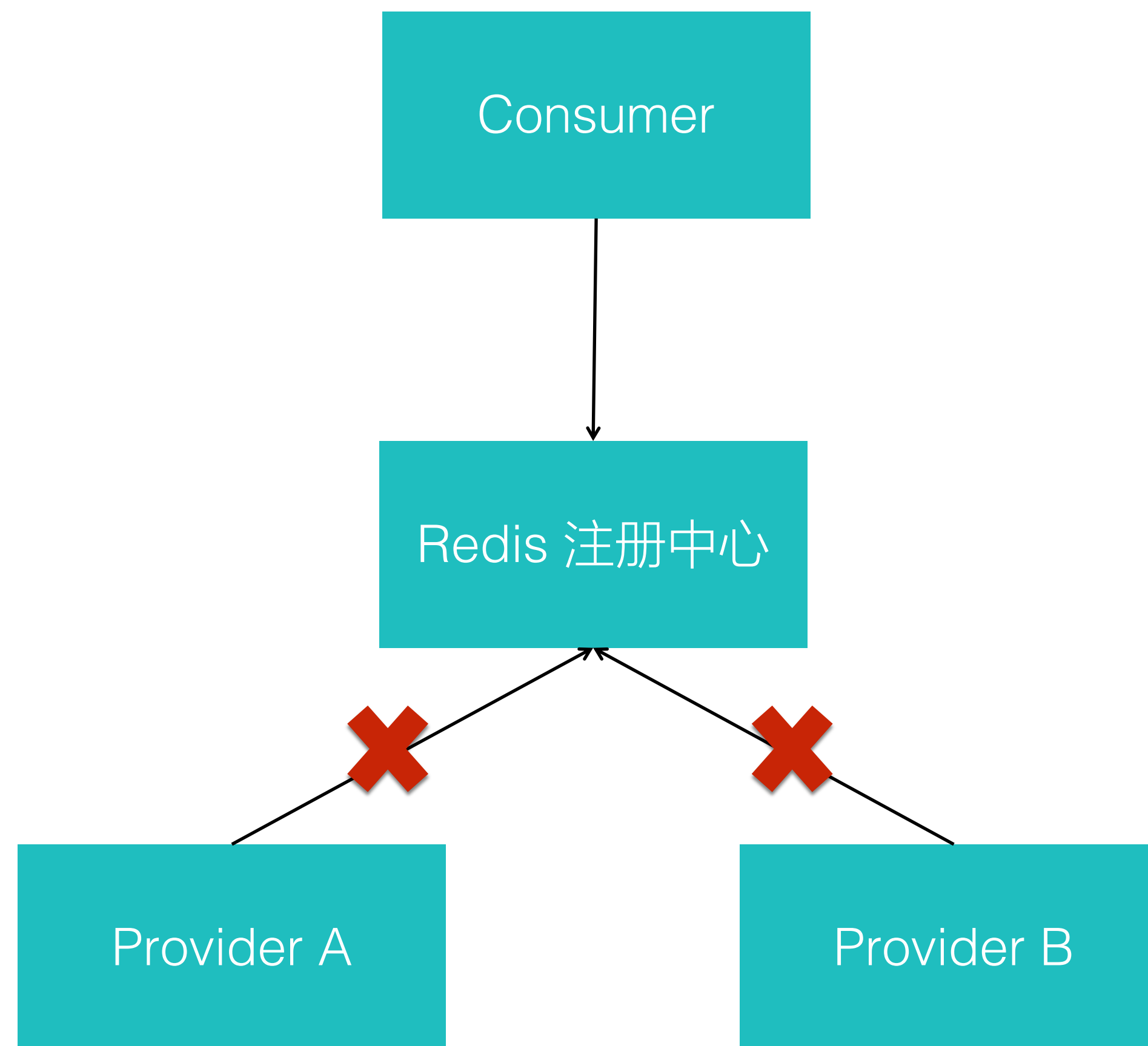
Consumer 访问服务超时，触发 Consumer 系统熔断，下游服务故障不会导致 Consumer 服务不可用

### 容灾方案：

给 Consumer 配置降级规则，当调用下游服务超过 2s 时，触发熔断，不会继续等待下游的响应。

### blade 相关命令：

```
blade prepare jvm --process dubbo-provider  
blade c dubbo delay --provider --service  
com.alibaba.demo.HelloService --methodname  
hello --time 5000 --process dubbo-provider
```





# AHAS 平台介绍



AHAS 控制台地址: <https://ahas.console.aliyun.com>

# ChaosBlade 开源讨论群（钉群）

ChaosBlade开源讨论群

687人







扫一扫群二维码，立刻加入该群。





奥运会全球指定云服务商