AGENT SANDBOX OPERATIONS

从「跑得起」到「管得住」

使用 OpenKruise Agents 破解 OpenClaw 日常运维难题

张

张振

OpenKruise Maintainer · 阿里云容器服务

阿里云

智能体与云沙箱的连接模式

智能体与云沙箱可通过两种模式连接，需根据部署位置、凭据暴露面、状态保持等综合评估。

沙箱内智能体模式 In-Sandbox Agent

Agent 运行时部署在沙箱内部，应用客户端通过网络接口与之通信

部署	镜像 / 容器 / VM 预置 Agent 框架与依赖
凭据	依赖模型 / 业务凭据，需流量转发注入
状态	沙箱实例承载完整运行时与会话状态
适用	托管应用 / 强环境依赖 / 长会话

沙箱工具化模式 Sandbox-as-Tool

Agent 运行在沙箱外，通过 SDK / MCP 远程调用沙箱能力

部署	Agent 在用户服务器，沙箱仅作执行工具
凭据	状态与凭据保留外部，仅传执行参数
状态	会话内保持变量 / 文件 / 依赖，可销毁
适用	快速迭代 / 短任务 / 多后端切换

云沙箱宜同时支持两种模式，按场景选择或组合使用。

阿里云

沙箱内智能体模式面临的挑战

01

数据安全隔离

- 多租户敏感数据隔离
- 防止跨租户逃逸与凭据泄露
- 网络 / 文件系统 / 进程多层边界

02

智能体版本管理

- 预热池升级:available 沙箱滚动换镜像
- 已分配实例:SandboxUpdateOps 批量在线升级
- 零会话中断、可灰度可回滚

03

状态持久化与成本

- 长会话保持运行时与文件系统状态
- 空闲实例占资源,需休眠 / 唤醒
- Checkpoint / Commit 平衡恢复性与开销

阿里云

OpenKruise Agents 项目概览

四大核心组件

- sandbox-manager
- sandbox-gateway
- sandbox-controller
- agent-runtime

核心 CRD 资源

Sandbox | SandboxSet | SandboxClaim | Checkpoint | SandboxUpdateOps | Commit

沙箱生命周期

Create › Claim › Run › Pause/Resume › Checkpoint › Upgrade › Delete

阿里云

为什么升级沙箱很难

已分配给用户的沙箱承载了实时业务，使其原地升级面临三重结构性挑战

01

携带用户状态

运行中的 Agent 会话与内存数据实时驻留，升级必须保留用户上下文与工作区

Stateful

02

升级兼容性无保证

跨版本升级常出现 API/依赖/运行时不兼容

OpenClaw 跨版本升级经常出现兼容性问题（例如：浏览器内核 / 沙箱镜像 / 工具链版本差异）

Compatibility

03

Pod 重建代价高

冷启动耗时叠加用户体验中断，IP 漂移与 rootfs 清空带来感知损失

High Cost

阿里云

预热池升级

> 预热池中 **available 沙箱** 可以像 **无状态副本** 一样滚动升级

控制并发

`maxUnavailable` 决定单批替换数量

只对 available 生效

已 claim 的 assigned 沙箱不受影响

与 AutoScaler 协同

升级期间由 SandboxAutoScaler 自动补齐池容量

阿里云

预热池分配时升级

预热池保留**稳定版本**镜像；用户在 Claim 时可声明**灰度新版本**，OpenKruise 就地替换镜像后拉起

场景与挑战 Why

- OpenClaw 等沙箱镜像存在多个版本（v1.2 stable / v1.3 beta）
- 预热池统一使用稳定版本，保证整体可用性
- 部分用户希望灰度试用新版本能力
- 整池升级代价高，且并非所有用户都愿升级

Claim 时镜像覆盖 image override on claim

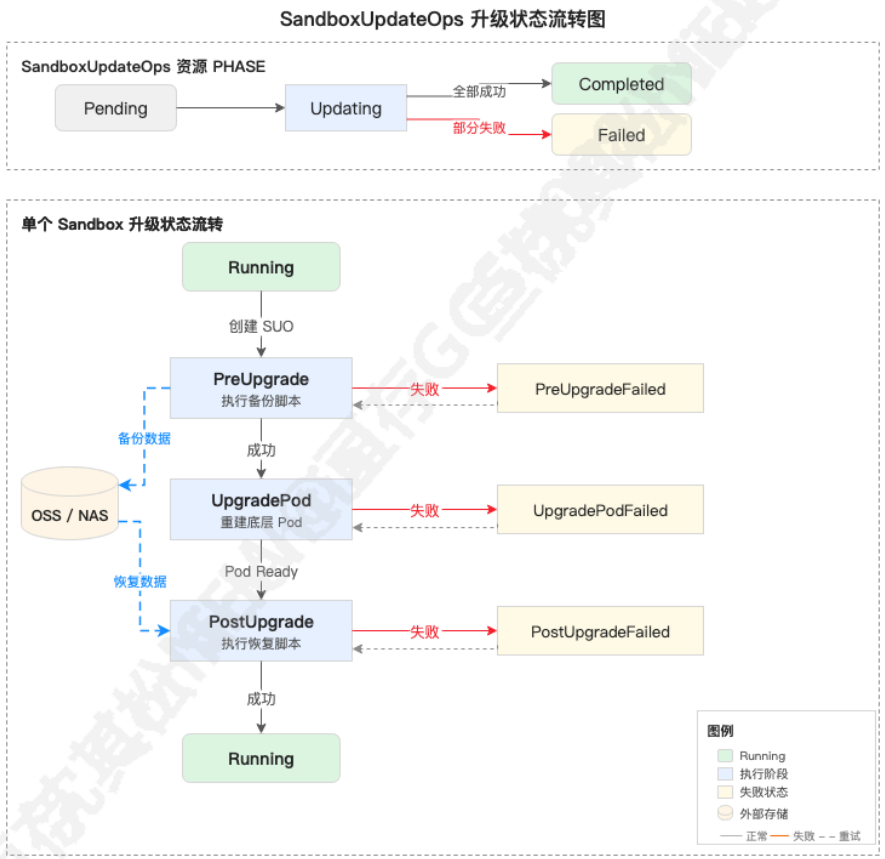
在 SandboxClaim.spec.inplaceUpdate.image 声明目标镜像；从预热池申领时通过原地升级替换镜像（非 sub-second 交付）

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: SandboxClaim
metadata:
  name: demo-sandbox-claim
  namespace: default
spec:
  templateName: demo
  inplaceUpdate:
    image: "new-image:latest"
```

- 预热池仍保持 v1.2 稳定版本，原地升级仅作用于此 Claim
- 直接 SandboxSet 新建路径会用新镜像创建沙箱

阿里云

已分配沙箱升级：SandboxUpdateOps



SandboxUpdateOps CR 示例 v0.3.0

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: SandboxUpdateOps
spec:
  selector:
    matchLabels:
      sandbox-template: demo
  updateStrategy:
    maxUnavailable: 2
  patch:
    spec:
      template:
        containers:
          - image: sandbox:v2
```

- ☑ 声明式批量升级 – 一次提交，控制器驱动滚动
- ☑ selector + maxUnavailable + patch – 三段式约束，安全可控

阿里云

Recreate 策略与 Lifecycle Hooks

PreUpgrade → UpgradePod → PostUpgrade 三阶段状态机及 sidecar 依赖

阶段一

PreUpgrade

通知 sidecar 优雅停止会话，workspace 状态落盘

备份 workspace 到共享存储

```
tar czf /mnt/shared/backup.tar.gz
-C /home/user/workspace .
```

timeoutSeconds: 120

阶段二

UpgradePod

Recreate 策略：销毁旧 Pod，按新镜像版本重建容器与 sidecar，挂载持久化卷

策略：Recreate · 短暂不可用

阶段三

PostUpgrade

恢复会话，探针就绪后 SLB 重新挂载流量

恢复 workspace

```
tar xzf /mnt/shared/backup.tar.gz
-C /home/user/workspace
```

timeoutSeconds: 120, envd 执行

Pod 生命周期节点　Sidecar / 探针依赖

阿里云

升级保障与方案对比

升级前自动 Checkpoint 备份，失败可一键回滚，全程不丢失会话状态。

方案 A

Recreate + Lifecycle Hooks

适用
镜像变更 / 大版本切换

优势
完全替换，干净彻底

劣势
依赖备份数据到外部

方案 B

In-Place Upgrade（原地升级）

适用
仅容器镜像更新

优势
不依赖外部存储做备份

劣势
非镜像更新不支持

方案 C

Patch 文件（增量补丁）

适用
少量配置/程序补丁

优势
业务中断少，无需额外备份

劣势
沙箱镜像/文件不一致

阿里云

沙箱休眠

两种休眠策略

定时器到期休眠

触发条件 超时计时器到期（如 30 分钟无活动）

典型场景 用户离开后自动节省资源

基于本地探测空闲状态

触发条件 探测脚本返回空闲

典型场景 智能体无任务运行时智能挂起

OpenClaw 空闲探测示例

```
# 执行命令
openclaw sessions --active 10 --json
# 空的预期返回结果
{
  "path": "/root/.openclaw/agents/main/sessions/sessions.json",
  "count": 0,
  "activeMinutes": 10,
  "sessions": []
}
```

count=0 表示无活跃会话，触发休眠

沙箱唤醒

两种唤醒策略

定时计划唤醒

触发条件： Cron 表达式定时唤醒

典型场景： 工作日早9点自动恢复开发环境

请求到达即唤醒

触发条件： 用户请求到达时即时唤醒

典型场景： 用户打开IDE或发起API调用时自动恢复

> 唤醒延迟取决于休眠深度：Pause态秒级恢复，Checkpoint态需加载快照（约10-30s）

Cron 状态查询示例

OpenClaw Cron 状态查询

```
# 执行命令
openclaw cron status --json

# 预期输出
{
  "enabled": true,
  "storePath": "/root/.openclaw/cron/jobs.json",
  "jobs": 3,
  "nextWakeAtMs": 1776970800000
}
```

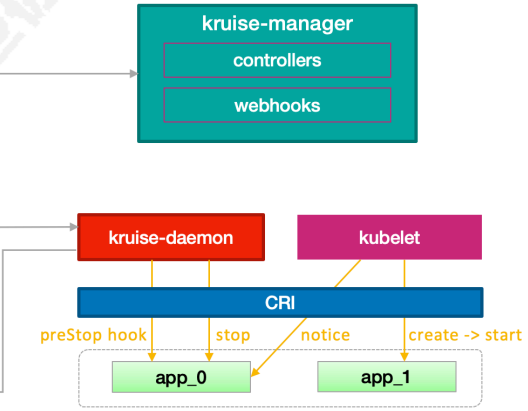
nextWakeAtMs 标记下次定时唤醒时间戳

阿里云

沙箱重启

CRR 实现原理 · InPlace Restart

```
apiVersion: apps.kruise.io/v1alpha1
kind: ContainerRecreateRequest
metadata:
  namespace: pod-namespace
  name: xxx
spec:
  podName: pod-name
  containers:
  - name: app
  strategy:
    failurePolicy: Fail
    orderedRecreate: false
    terminationGracePeriodSeconds: 30
    unreadyGracePeriodSeconds: 3
  activeDeadlineSeconds: 300
  ttlSecondsAfterFinished: 1800
status:
  containerRecreateStates:
  - name: app
    phase: Succeeded
  phase: Completed
  # ...
```



典型用户场景

- 智能体重启生效新配置 / 提示词 / 工具集
- 智能体进程假死、循环或 OOM 后的快速自愈
- 镜像热修补后无需重建 Pod 即可恢复服务

- 保留 Pod 网络与 Volume，**身份与状态完全不变**
- 只重建指定容器，**秒级恢复**，无需调度

文件系统保持

社区版 OpenKruise

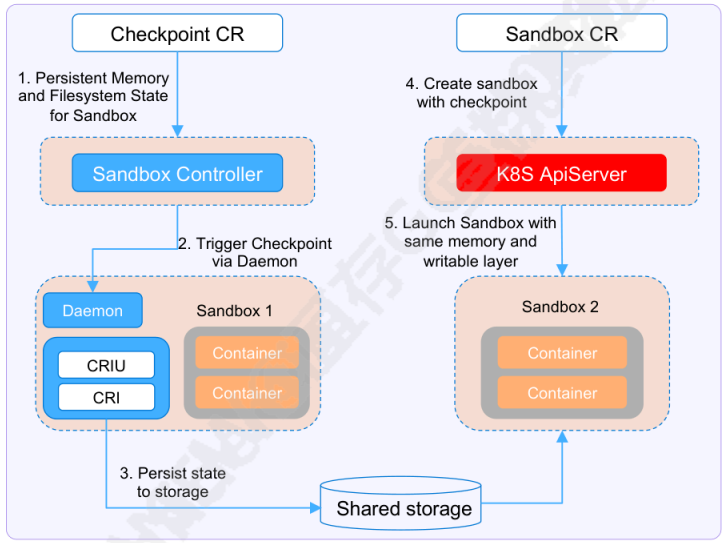
- emptyDir 数据保持
- PVC 数据保持

ACS 增强版

emptyDir + PVC 数据保持

额外支持容器读写层数据保持

Checkpoint：状态快照与克隆



核心能力

1 内存 + 文件系统快照

- 通过 CRIU等技术 保存进程状态（寄存器、内存页、网络连接）
- 通过 CRI 接口导出容器 rw-layer

2 快照输出

- 保存为包（内存 dump + fs diff）
- 推送至共享存储，支持跨节点恢复

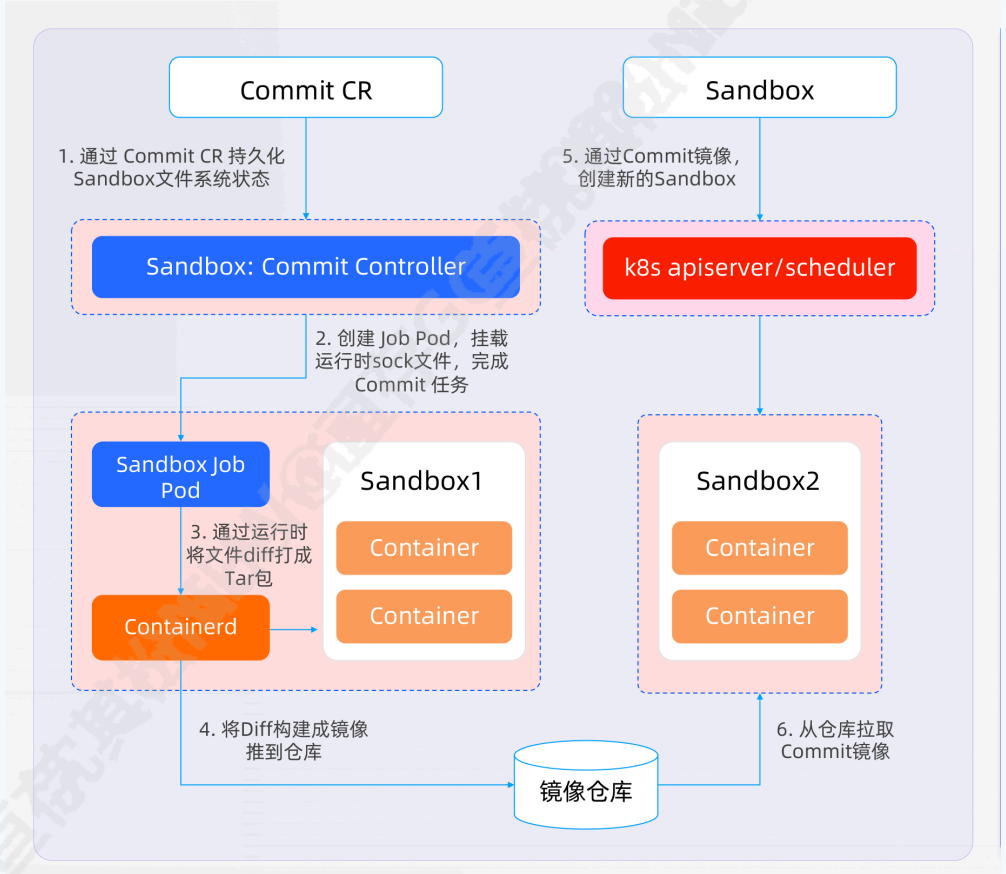
3 克隆多实例

- 从同一 Checkpoint 创建多个新 Sandbox
- 适用于 RL 训练任务并行分叉

阿里云

Commit：文件系统层持久化

Commit 6步流程原理



Commit CRD YAML

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: Commit
metadata:
  name: my-sandbox-commit
spec:
  podName: sandbox-pod-0
  containerName: openclaw
  image: registry.cn-hangzhou.aliyuncs.com/
    my-repo/openclaw:v1.2-snapshot
  ttl: 168h
```

字段说明

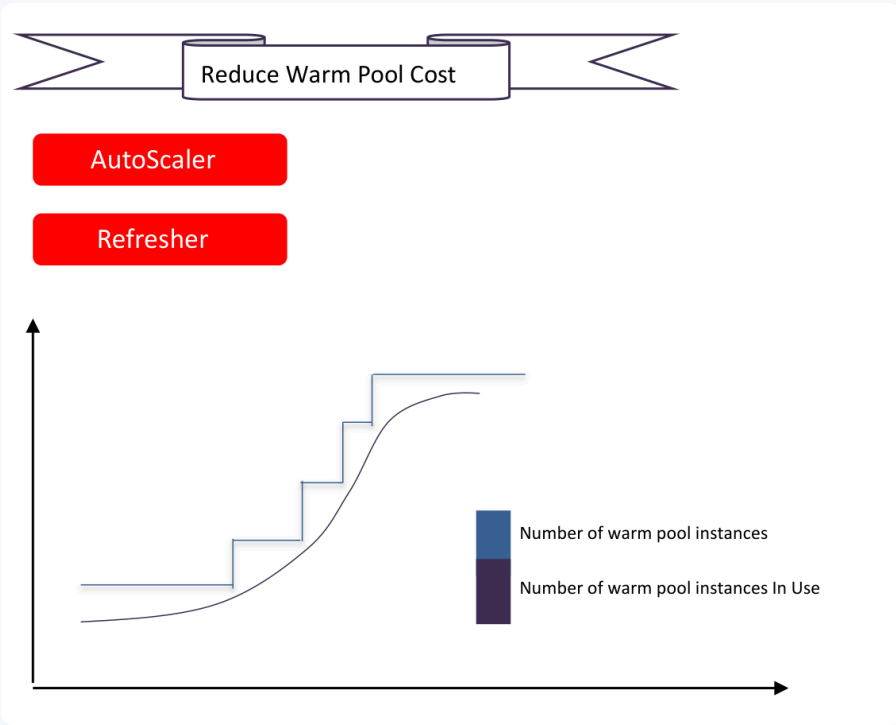
- **podName** – 目标 Pod
- **containerName** – 目标容器
- **image** – 输出镜像地址
- **ttl** – 自动回收（默认 7 天）

阿里云

预热池水平弹性

PoolingAutoScaler

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: PoolingAutoScaler
metadata:
  name: code-autoscaler
spec:
  scaleTargetRef:
    apiVersion: agents.kruise.io/v1alpha1
    kind: SandboxSet
    name: code-warm-pool
  observeWindowSeconds: 600
  minReplicas: 3
  maxReplicas: 100
  minAvailableRatio: 20%
  maxAvailableRatio: 90%
  schedule:
  - startTime: "2026-03-15 00:00:00"
    endTime: "2026-04-15 00:00:00"
    bounds:
    - cron: " 0-8 ? MON-FRI"
      minReplicas: 10
      maxReplicas: 150
```



AutoScaler / Refresher 架构示意

阿里云

预热池垂直弹性（VPA）

Claim 时原地 CPU Resize

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: SandboxClaim
metadata:
 name: demo-sandbox-claim
 namespace: default
spec:
 templateName: demo
 inplaceUpdate:
  resources:
   requests:
    cpu: "1000m"
   limits:
    cpu: "2"
```

inplaceUpdate.resources 声明目标 CPU 规格

小规格预热 · 按需放大

1. 预热池以最小规格（如 0.5C）预占资源，降低闲置成本
2. 用户 Claim 时原地放大 CPU（如 0.5C → 2C），无需重建 Pod
3. 无暖池可用时直接以目标规格创建新 Pod

约束条件

- 仅支持主容器 CPU resize（memory 暂不支持）
- 不能改变 Pod QoS class
- 依赖 K8s InPlacePodVerticalScaling（1.33+ 默认开启）

阿里云

实战案例：OpenClaw 8000 实例

800 ms
P99 Claim 延迟

-60 %
空闲休眠节省成本

8,000
实例 · 首日交付

5 天上线
从集群到生产

零中断
升级无会话感知

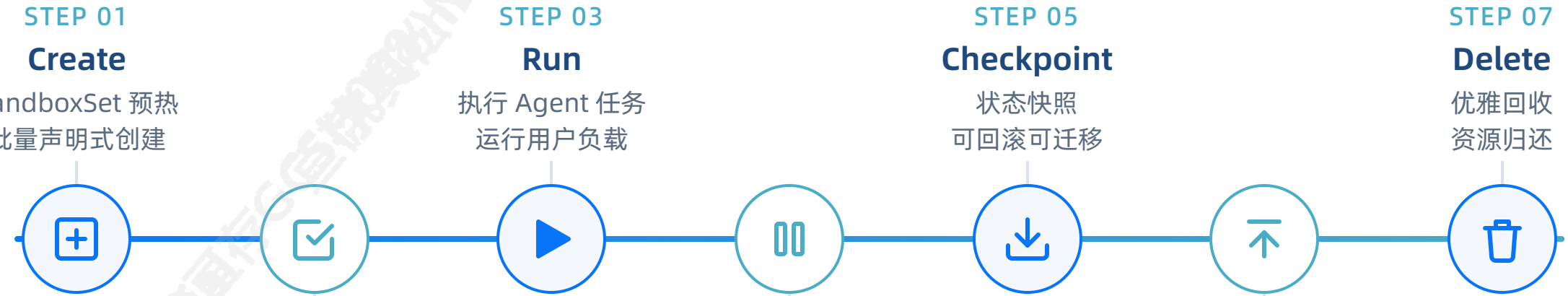
<3 s
Checkpoint 备份恢复

OpenKruise Agents 把 Sandbox **当 Pod 一样管理**，平台运维只需关注业务

阿里云

运维操作全景图

沙箱全生命周期 **7 个核心动作** · 声明式 API + 单一控制平面统一收敛



统一控制平面： 所有 7 个动作均通过 Sandbox CRD 声明式下发，复用 K8s Reconcile 闭环，无独立运维通道

阿里云

未来路线

FUTURE ROADMAP · 开源共建 / Runtime 解耦 / 成本与安全演进

扫码关注 OpenKruise Agents

01

开源 Checkpoint

跟进PodCheckpoint API 演进， 多运行时支持

PodCheckpoint | CRI | 社区贡献

02

通用 Sidecar Runtime

解耦 Agent 与 Runtime，统一命令执行 / 文件传输能力

Sidecar | 可插拔 | 协议标准化

03

成本优化

Spot 实例混部 + 长休眠状态下沉至磁盘，进一步压缩单价

Spot 实例 | 磁盘下沉 | 单价↓

04

流量管理与安全策略

Gateway 限流熔断、网络隔离与多租户安全 CRD 策略管控

Gateway 限流 | 网络隔离 | SecurityProfile