InfoQ 极客传媒

基于容器构建的 AI 智能体基础设施落地实践

演讲人：黄涛

阿里云 / 弹性计算 资深技术专家

AiCon

全球人工智能开发与应用大会

目录

content

极客邦科技 2026 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：12月18-19日

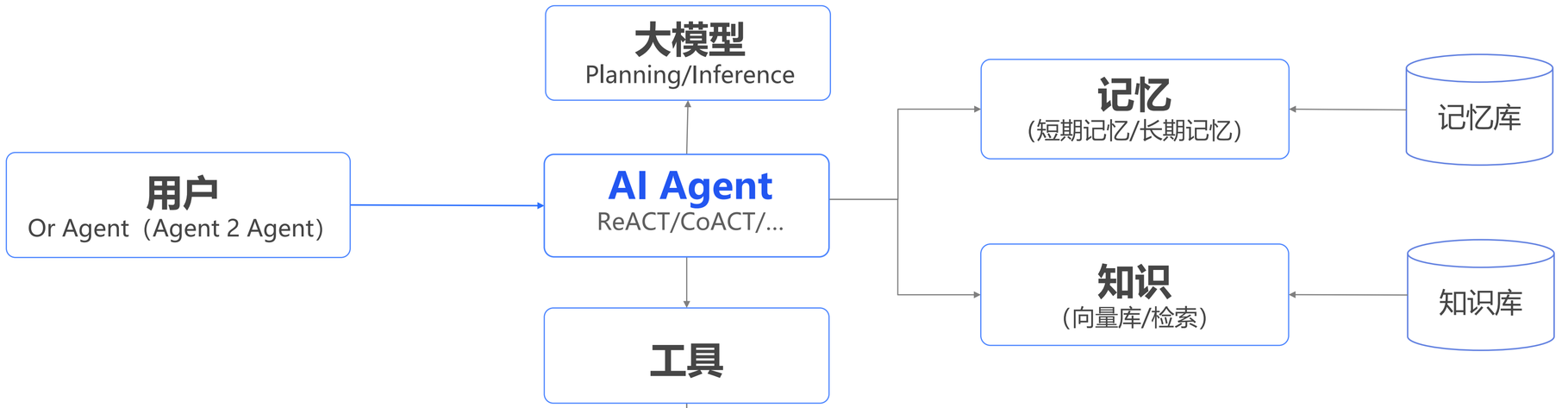
- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

01

基于容器构建的Agent Sandbox

阿里云容器服务智能体方案简图



MCP让工具使用更高效

MCP Client/MCP Protocol

MCP Servers

Agent Resource

ACS/ACK Pod

ACS Sandbox

Sandbox
（Code Interpreter/Browser Use/Compute Use）

Sandbox实现环境隔离、安全执行、极致弹性

ACK Agent Extension

Sandbox Resource

ACS Sandbox | ACS Sandbox | ... | ACS Sandbox | Pod | Pod | ... | Pod

AC R

Sandbox Templates
Code Interpreter
Browser
Compute Use

ACK：阿里云容器服务Kubernetes版
ACS：阿里云容器计算服务

ACR：阿里云容器镜像服务

智能体应用落地面临的业务挑战

Sandbox：高安全性、状态保持场景

数据安全

1. 攻击者提示词诱导恶意行为模型动态生成不可预期代码；
2. 多会话数据需要严格隔离

算力 需要运行在安全隔离的环境

大规模极速交付

1. 模型动态控制工具的执行，存在更大规模的秒级交付弹性并发
2. 会话数量、会话并行度，加剧资源需求动态波动

算力 要极致弹性能力

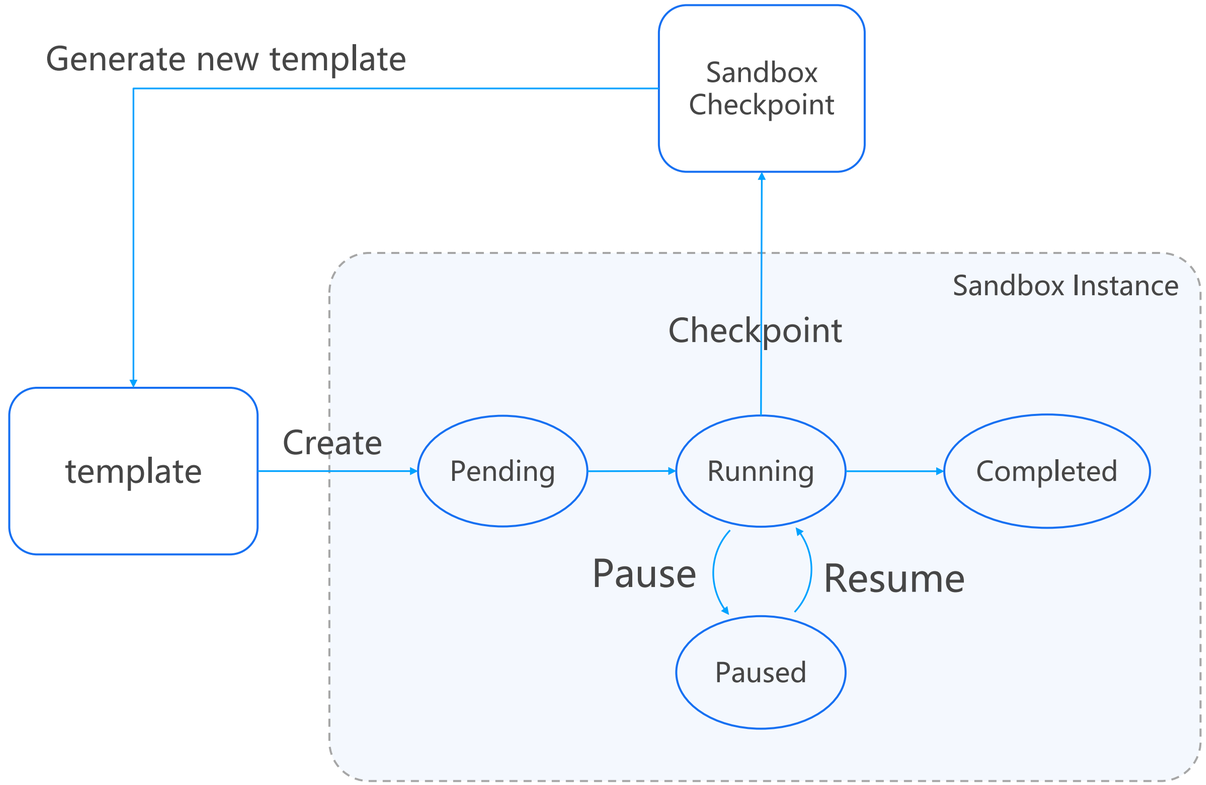
状态持久化

1. AI Agent 长周期/多步骤/强状态存在跨多轮交互与工具调用
2. 多会话并存加剧沙箱规模膨胀
3. 不是所有会话时刻存活

状态 要能有效保持
成本 要能合理控制

智能体应用落地面临的业务挑战

Sandbox：高安全性、状态保持场景



复杂的沙箱生命周期

1.模板： 包括镜像、编排和可选的checkpoint

2.Paused：资源占用最小化（无CPU等消耗）

3.Checkpoint: 内存、临时存储和显存状态数据

02

Agent Sandbox 关键技术实现

Sandbox安全隔离的典型技术实现

应对数据泄露、代码注入、网络攻击 等安全风险

计算隔离	网络隔离	存储隔离	鉴权隔离	可观测
• CPU/内存相互隔离，互不干扰	• 禁用东西向网络 • 南北向单向连通 • 独立公网访问	• 共享存储的挂载点隔离	• 单Agent的独立rbac鉴权	• 所有Agent行为可追踪、可审计

大规模极致交付的容器资源管理技术

Sandbox资源管理的复杂度

- Sandbox运行时间短；对启动速度要求高
- Sandbox需等待人类或其它工具反馈，等待时间长，整体生命周期时长难预测
- Sandbox资源消耗难预测
- 极致弹性的技术实现复杂度高

Agent Sandbox 资源管理

Serverless节点池

业界典型方案：

- 阿里云ACS Pod
- AWS Fargate
- Azure AKS Pod
- GKE Agent Sandbox

绝大部分Agent用户倾向优先使用Serverless以简化Sandbox使用复杂度

K8S节点池

业界典型方案：

- 阿里云runD
- Kata/kata on PVM
- Firecracker
- gVisor

安全沙箱技术、二次虚拟化技术；计算、存储、网络隔离等技术

Sandbox状态保持技术的实现

- 文件系统的数据保持

容器rootfs数据，临时卷，持久化卷的数据检查点保留能力 。

- 内存/显存数据保持

CRIU、VM Memory Checkpoint等技术实现内存数据的快照点保留能力；NVIDIA/cuda-checkpoint等显存保持能力。

两种开源方案的思路：

firecracker

文件系统使用持久化存储

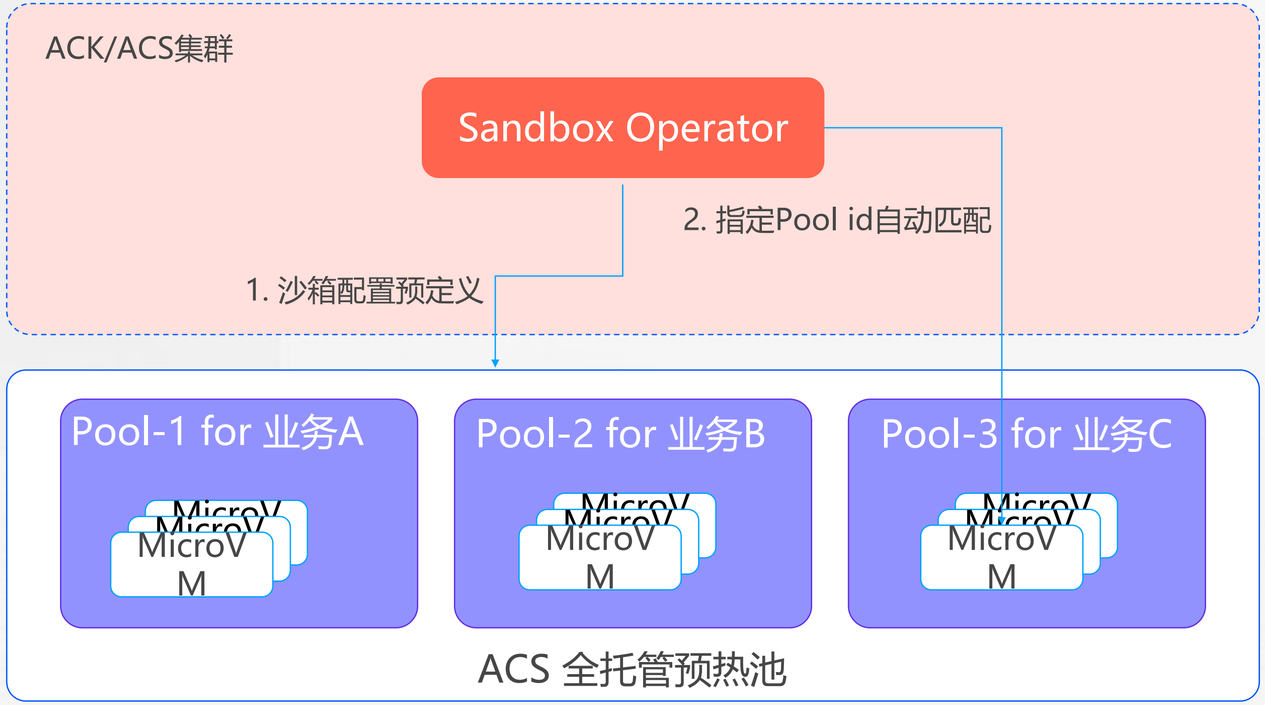
VM维度的内存Checkpoint

gvisor

文件系统使用数据持久化存储

gvisor的内存CRIU技术

阿里云ACS Sandbox极速唤醒的工程实践经验



MicroVM

- 计算规格、网卡、存储设备预热

RooFS

- ACS rootfs缓存预热缓存预

容器镜像缓存

- ACS镜像缓存自动创建挂载

1.池化技术：同构沙箱配置预定义，自动匹配实现极速启动与唤醒；预热池管理产品化

2.Serverless模式：具备大规模极致交付能力的Serverless算力，简化Sandbox资源管理

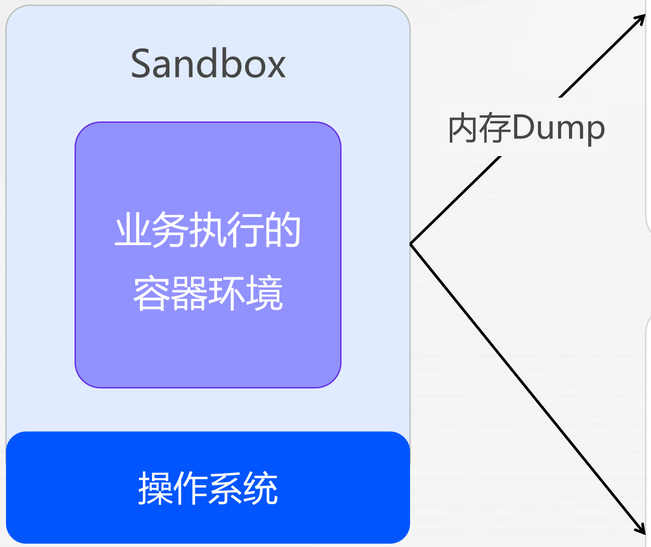
3. 基于已有的ACS安全沙箱隔离技术、网络、存储 隔离能力

4. 基于阿里云块存储的快照预热与复制能力，实现极速Rootfs恢复

5. Sandbox Operator简化客户使用方式，管理Sandbox生命周期；屏蔽预热池管理复杂度

- **Sandbox Operator生产实践经验进一步开源到OpenKruise Agents社区**

阿里云ACS Sandbox休眠唤醒运行时生产经验



内存Dump

Solution 1：利用CRIU技术实现容器环境的休眠和唤醒

进程级
内存数据

Lazy Load
秒级恢复

Sandbox

- 休眠轻量，数据量少，休眠快
- 内存状态管理数量少，仅需考虑业务程序的状态
- 硬件耦合度较低

Solution 2：利用虚拟机快照技术实现整机环境的休眠和唤醒

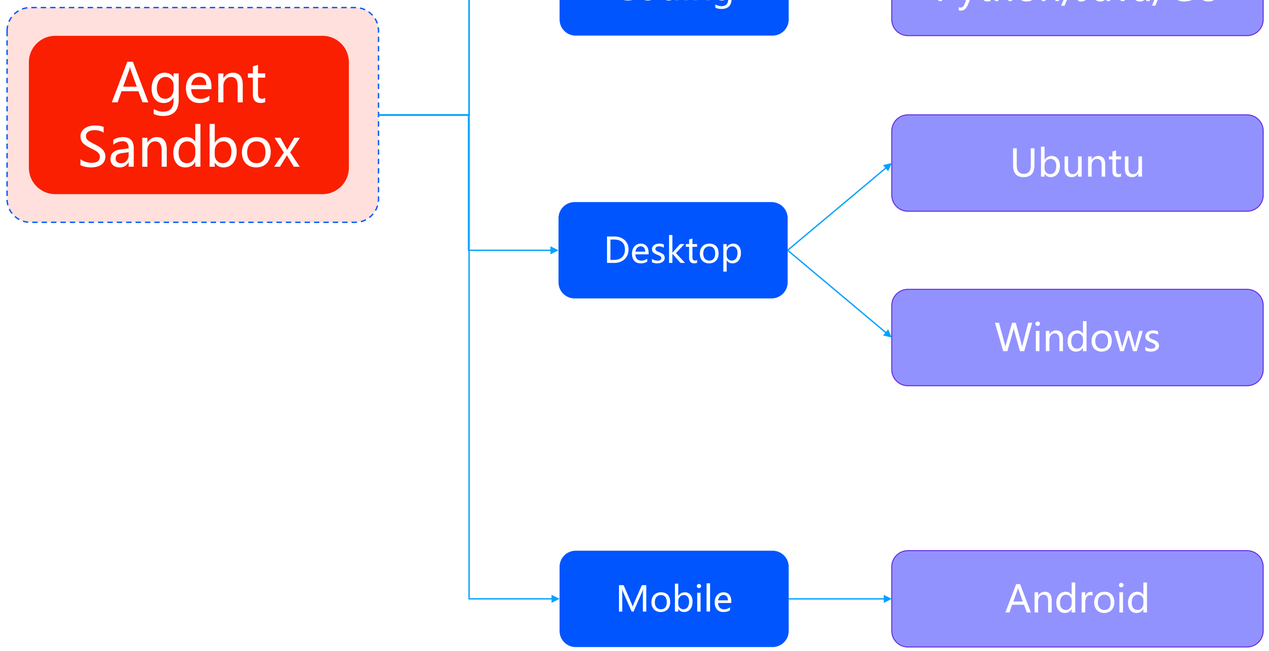
整机
内存数据

Lazy Load
秒级恢复

Sandbox

- 虚拟机休眠技术更成熟，稳定性好
- 硬件耦合度较高，不支持跨代际

阿里云ACS Agent Sandbox运行时技术



支持浏览器程序的执行环境

- 解决多pidns情况下的休眠失败的问题

提供多语言执行环境

Ubuntu操作系统的运行时，支持桌面服务的视频串流。

- 特权能力的支持
- 基于VNC服务的串流能力

执行Windows闭源系统运行，Windows应用安装，支持Windows桌面串流协议。

- 应用安装后，利用快照保存能力环境模板制作
- 视频串流能力的建设
- 内置接管的Windows mcp server

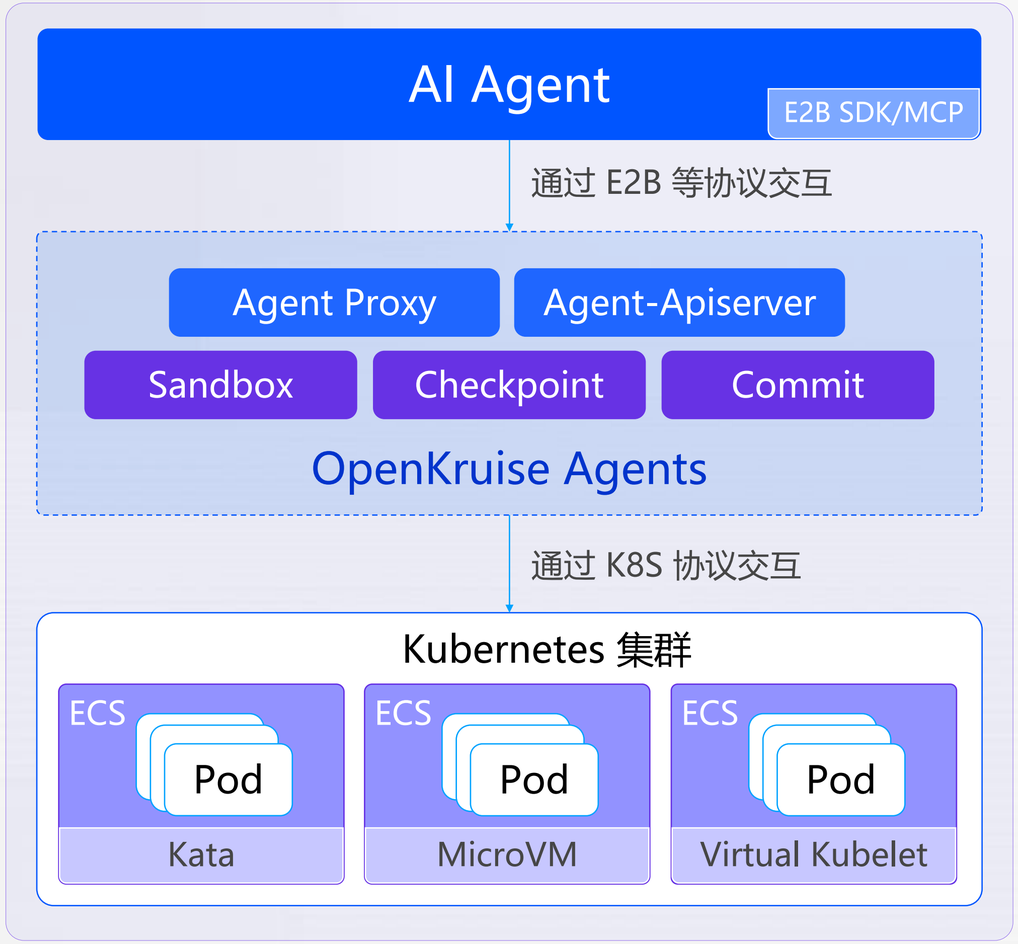
支持安卓虚拟机的运行，定制内核，支持App安装和视频串流。

- 定制OS支持安卓虚拟机运行
- 支持安卓环境的模块保存能力

03

阿里云开源的 OpenKruise Agents

应用生态层的衔接：阿里云开源的OpenKruise Agents



OpenKruise Agents提供了管理 Sandbox 生命周期的标准能力:

- Sandbox 创建、休眠、唤醒， 包括内存、读写层数据等
- 高效资源供给：通过资源池化和资源变配等技术， 加速沙箱的冷启动时间
- Checkpoint/Fork 能力，满足 RL 训练任务

同时， OpenKruise Agents 也是 AI Agent 与 Kubenetes 的中间纽带，对外提供 Agent 工具调用能力

- 向上通过E2B等协议，对接 AI Agent
- 向下通过K8S协议，使用 Pod 承载 Sandbox

OpenKruise Agents：AI Agent 集成方式

Python SDK（E2B兼容）

面向 AI 科学家 / 开发者：

- Agent-Apiserver 支持通过 E2B等协议 的 REST/gRPC 接口，无缝对接 E2B 原生 Python SDK
- 支持 create()、sleep()、wake()、destroy() 等高级生命周期操作
- 无需了解 Kubernetes 细节，以代码原生方式管理 Sandbox 实例，快速嵌入 AI Agent 工作流

```python
# Import the E2B SDK
from e2b_code_interpreter import Sandbox

# 指定Sandbox Template，创建 Sandbox
sbx = Sandbox.create(template="code-interpreter-custom", timeout=300)
print(f"sandbox id: {sbx.sandbox_id}")

# 通过 Sandbox RunCode
def execute_python_code(s: Sandbox, code: str):
    # Execute Python code inside the sandbox
# runcode 例子
execute_python_code(sbx, "print('hello world')")
```

Sandbox CR

面向 Agent 平台/运维工程师：

- 根据 Sandbox CR 自动完成 Pod 创建、状态同步、休眠调度与资源回收
- 类似 Deployment，通过 CRD SandboxSet 声明期望的 Sandbox 实例数量与配置

```yaml
apiVersion: agents.x-k8s.io/v1alpha1
kind: Sandbox
metadata:
  name: my-sandbox
spec:
  podTemplate:
    spec:
      containers:
      - name: my-container
        image: <IMAGE>
```

OpenKruise Agents 项目整体说明

- Agent-Sandbox 是 CNCF 孵化项目 OpenKruise 社区开源面向 AI Agent Sandbox 的端到端解决方案，旨在提供标准化的 Sandbox 生命周期管理能力。

Sandbox Apiserver

- 支持社区E2B等API
- Sandbox路由管理

Sandbox管理

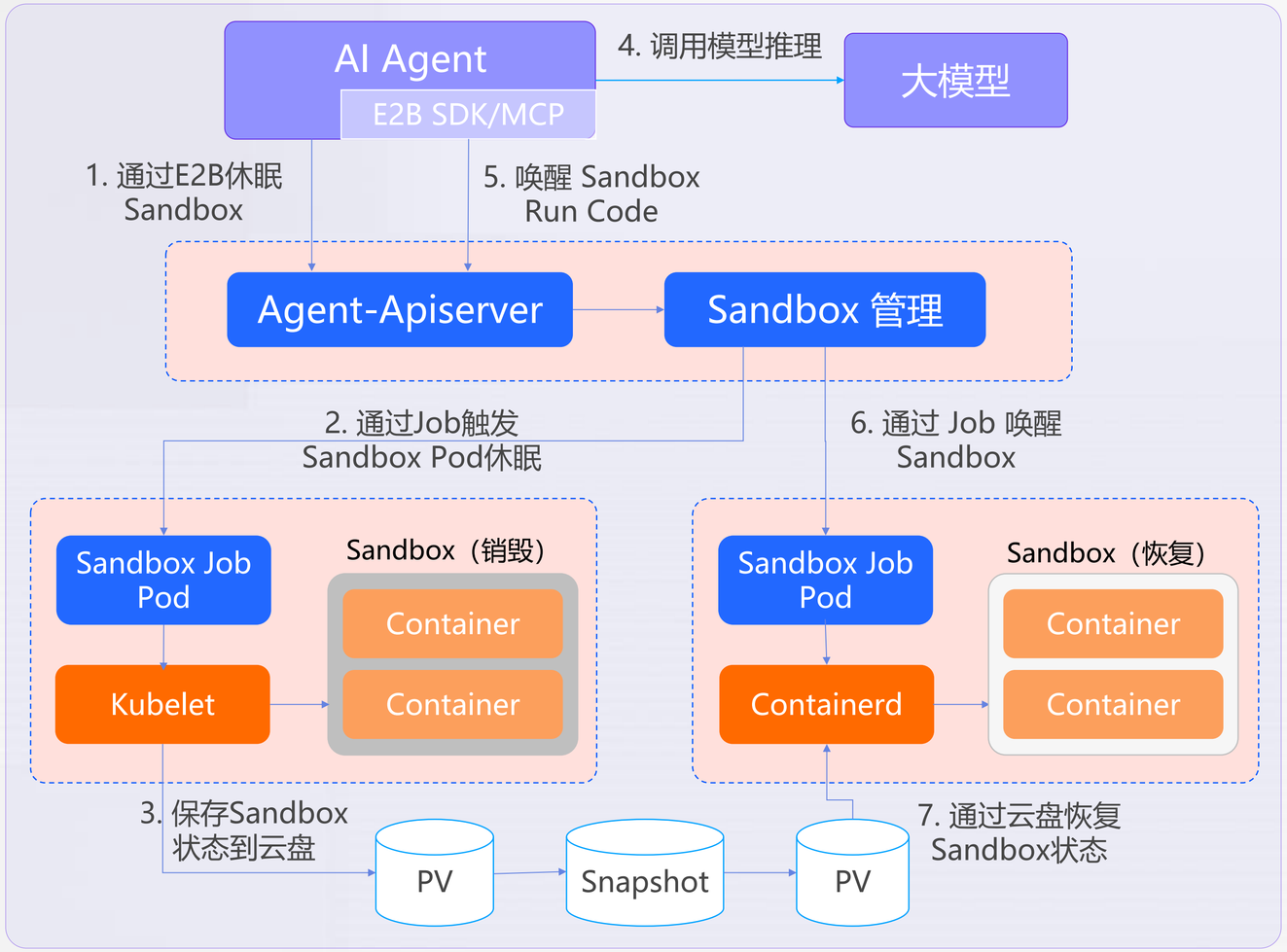
- Sandbox 休眠/唤醒
- Sandbox 池化

Checkpoint管理

- Pod的快照/克隆
- 容器的镜像Commit

项目地址：https://github.com/openkruise/agents

OpenKruise Agents : Sandbox 实例状态休眠、唤醒

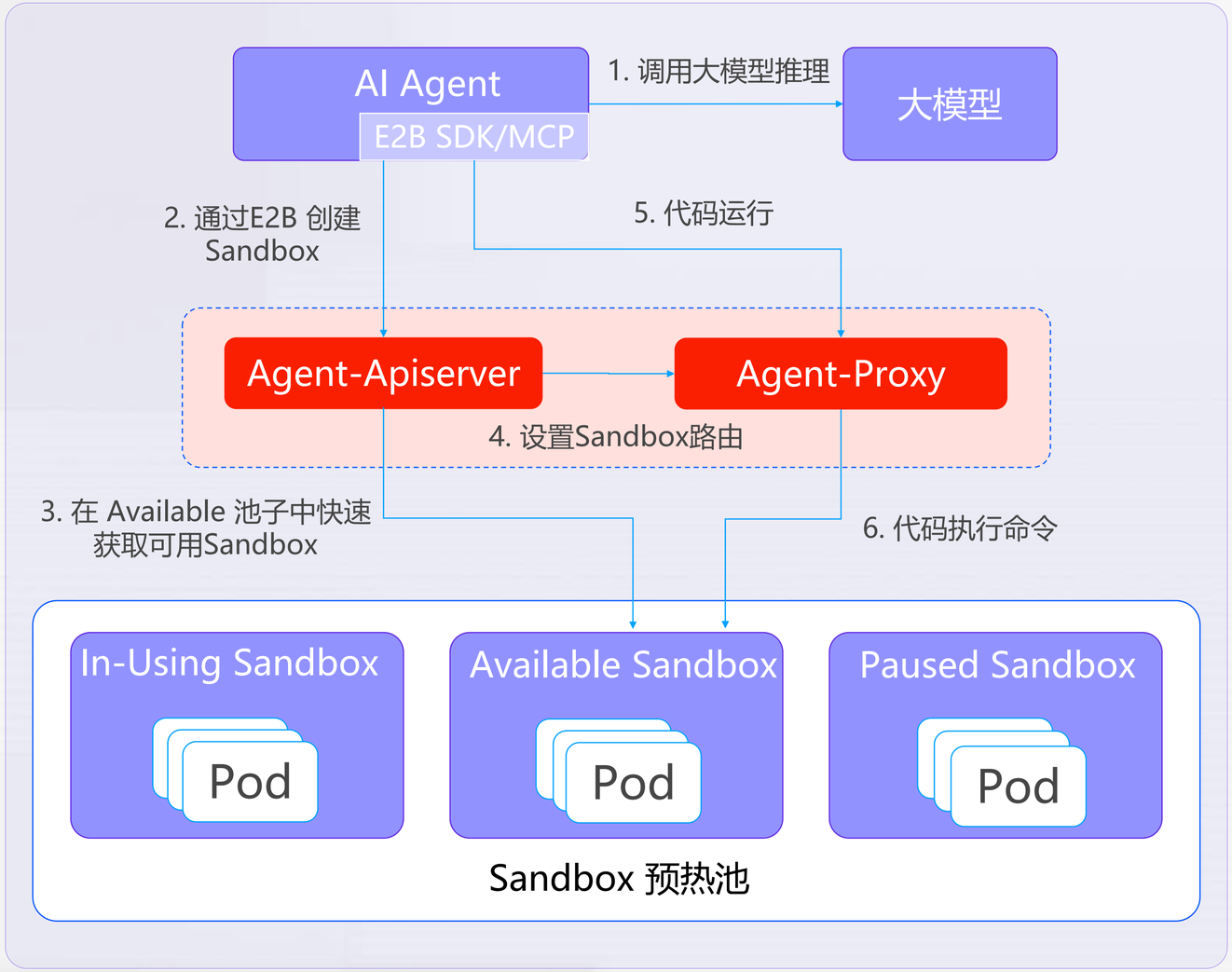


Sandbox 实例状态生命周期管理：

- **休眠（Sleep）**：空闲时，可将其转入休眠状态，降低成本
- **唤醒（Wake）**：当 AI Agent 再次需要执行任务时，能快速恢复 Sandbox 到休眠前的状态

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: Sandbox
metadata:
  name: sample
spec:
  pause: false
  persistentContents:
  - memory
  - filesystem
  - ip
  - gpuMemory
  template:
    metadata:
      labels:
        agent: sample
    spec:
      containers:
      - name: my-session
        image: session:v1
```

OpenKruise Agents : Sandbox 池化扩容



Sandbox 池化扩容：

- **快速获取**：预热池中秒级分配就绪实例，避免冷启动延迟
- **快速路由**：通过内置的Envoy边车对Sandbox进行路由

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: SandboxSet
metadata:
  name: demo
spec:
  replicas: 10
  template:
    metadata:
      labels:
        agent: testpause001
    spec:
      containers:
      - image: nginx:1.14.1-8.6
```

OpenKruise Agents : Sandbox 池化管理

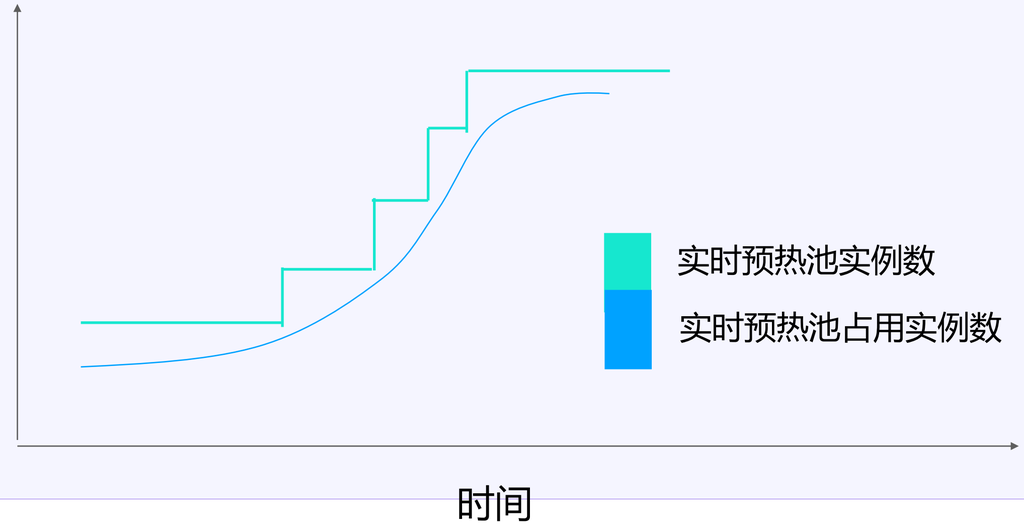
降低预热成本

AutoScaler

- 基于预热池水位的弹性
- 基于时间的弹性

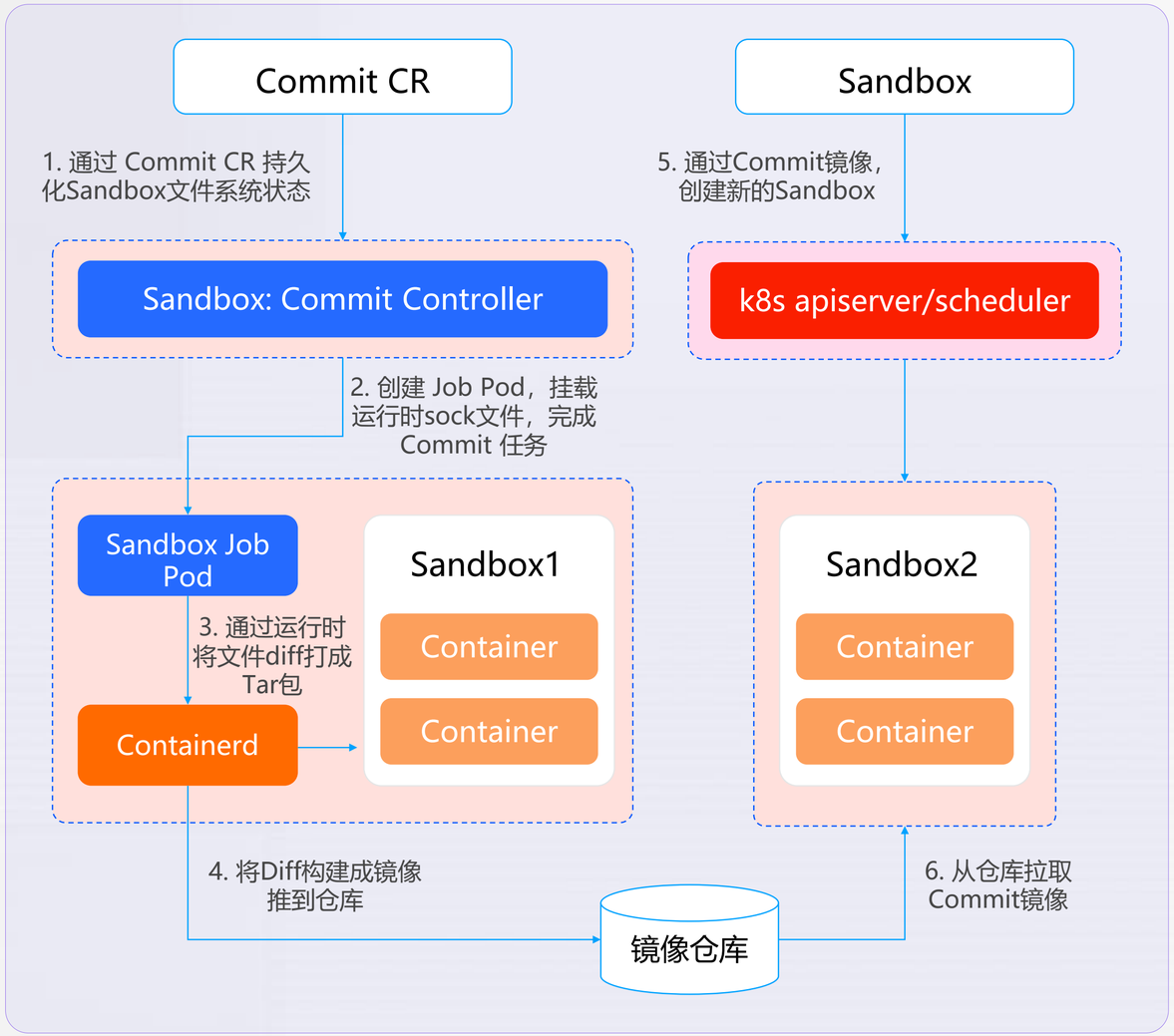
Refresher

- 已使用的Sandbox复用
- Sandbox镜像的动态更新




```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: PoolingAutoScaler
metadata:
  name: code-autoscaler
spec:
  scaleTargetRef:
    apiVersion: agents.kruise.io/v1alpha1
    kind: sandboxset
    name: code-interpreter
  observeWindowSeconds: 600
  # 预热池大小限制
  minReplicas: 3
  maxReplicas: 100
  # 基于预热池水位的弹性
  minAvailableRatio: 20%
  maxAvailableRatio: 90%
  # 基于时间调度的弹性
  schedule:
  - startTime: "2025-12-1 00:00:00"
    endTime: "2031-12-12 00:00:00"
    bounds:
    - cron: "* 0-8 ? * MON-FRI"
      minReplicas: 10
      maxReplicas: 150
```

OpenKruise Agents：状态保持 Commit

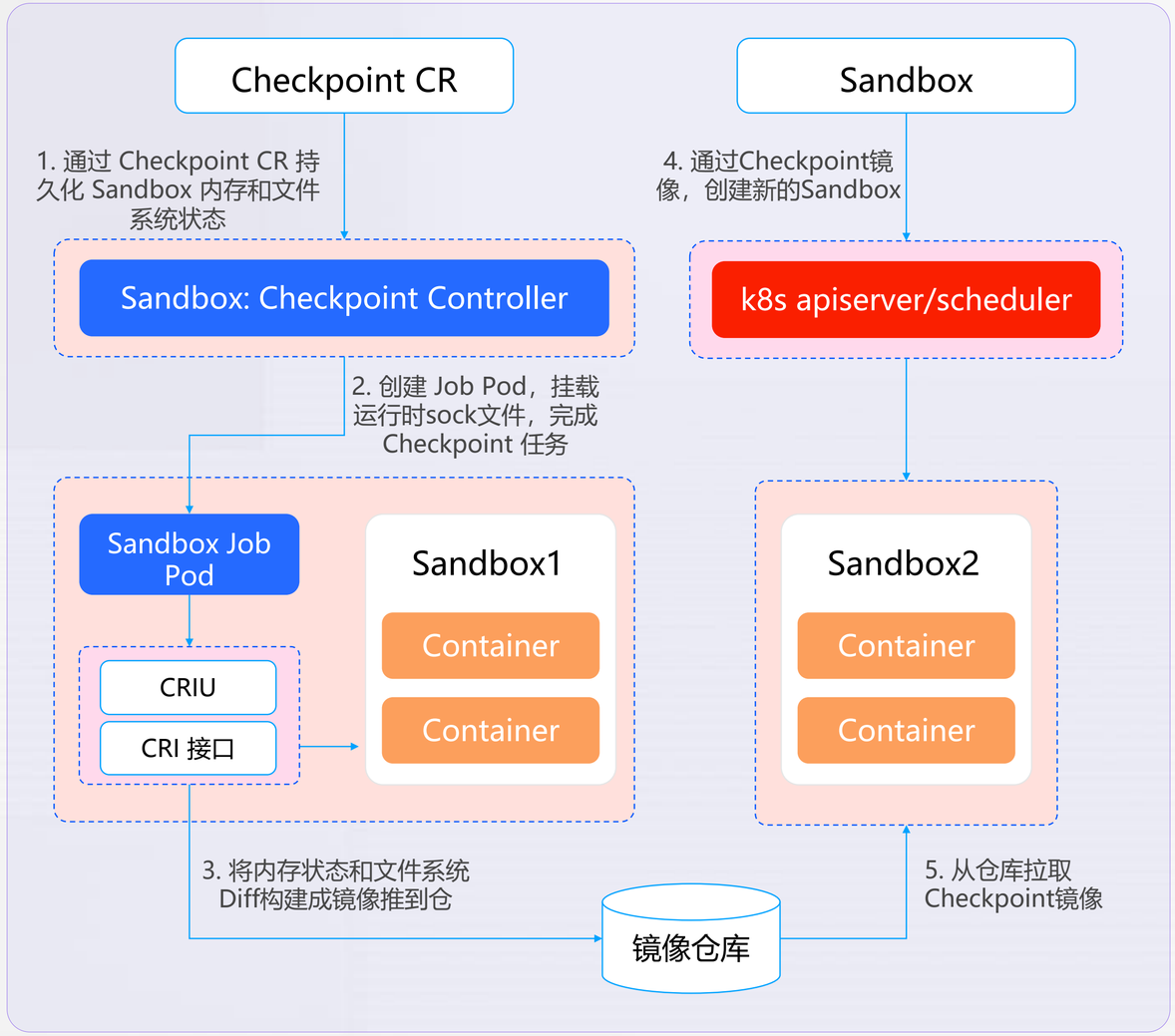


Commit CR：持久化文件系统读写层

- 通过 CRI 接口，捕获 Sandbox 的 overlayfs/rw-layer
- 将运行时文件系统差异层打包为标准 OCI 镜像
- 实现 Sandbox 状态的可回溯、可重建、可迁移

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: Commit
metadata:
  name: commit-01
spec:
  # commit Pod对象
  podName: commit-01
  # commit 容器名
  containerName: main
  # commit 镜像地址
  image: commit/nginx:commit-01
  # 回收 Commit CR 策略
  ttl: 72h
  # push 镜像鉴权
  registryAuth:
    secrets:
    - push-secrets
```

OpenKruise Agents：状态保持 Checkpoint



Checkpoint CR：持久化内存和文件系统读写层

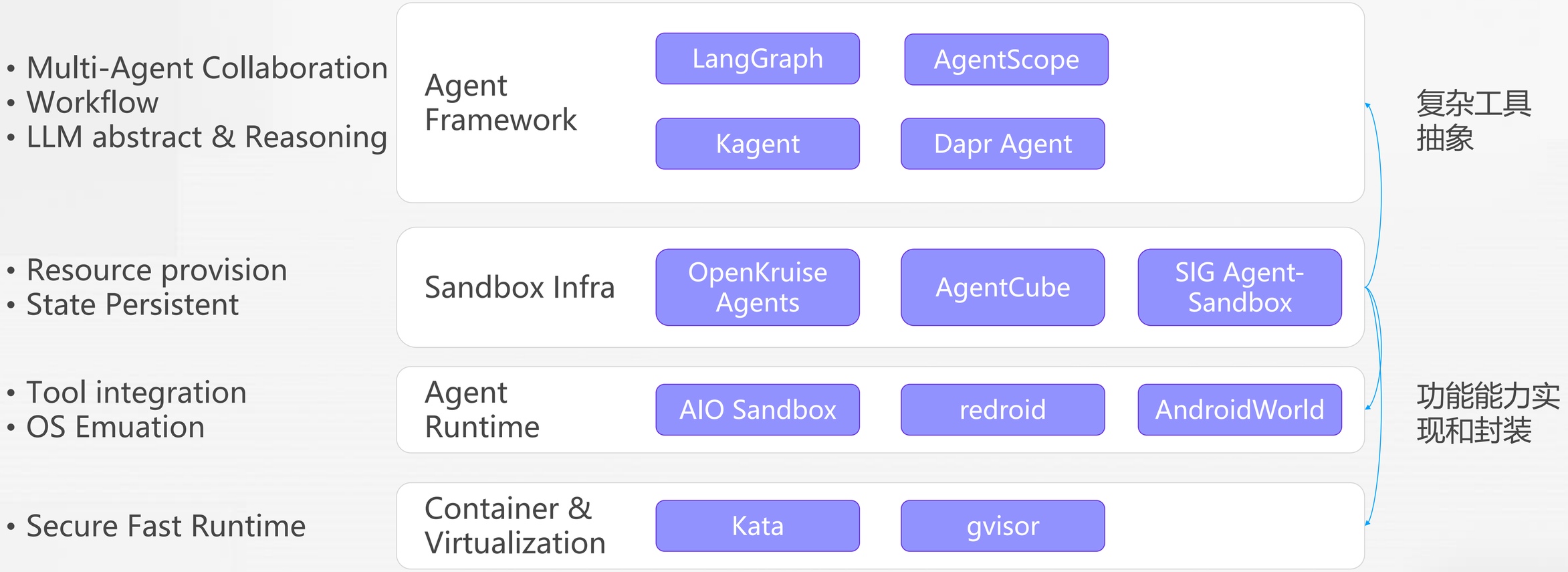
- 文件系统层：调用 CRI 接口，导出容器 rw-layer 为标准 OCI 镜像
- 内存与进程状态：通过 CRIU 执行 checkpoint，保存寄存器、内存页、网络连接等上下文
- 输出：一个可重建的完整 Sandbox 快照（镜像 + 内存 dump）
- 当前通过镜像commit实现，未来通过云盘实现

```yaml
apiVersion: agents.kruise.io/v1alpha1
kind: Checkpoint
metadata:
  name: checkpoint-demo-1
  namespace: default
spec:
  # 目标Pod Name
  podName: pod-demo-1
  # checkpoint之后是否要求Pod处于Running状态。
  # 如果是 false ，Pod状态变为 Succeeded
  keepPodRunning: false
  # Checkpoint 回收时间，ttlAfterFinished 后会主动回收 checkpoint 资源
  ttlAfterFinished: 30m, 30h, 30d
  persistentContents: # 目前仅支持 memory,filesystem; filesystem 两种组合。默认启用 memory,filesystem
  - memory
  - filesystem
```

04

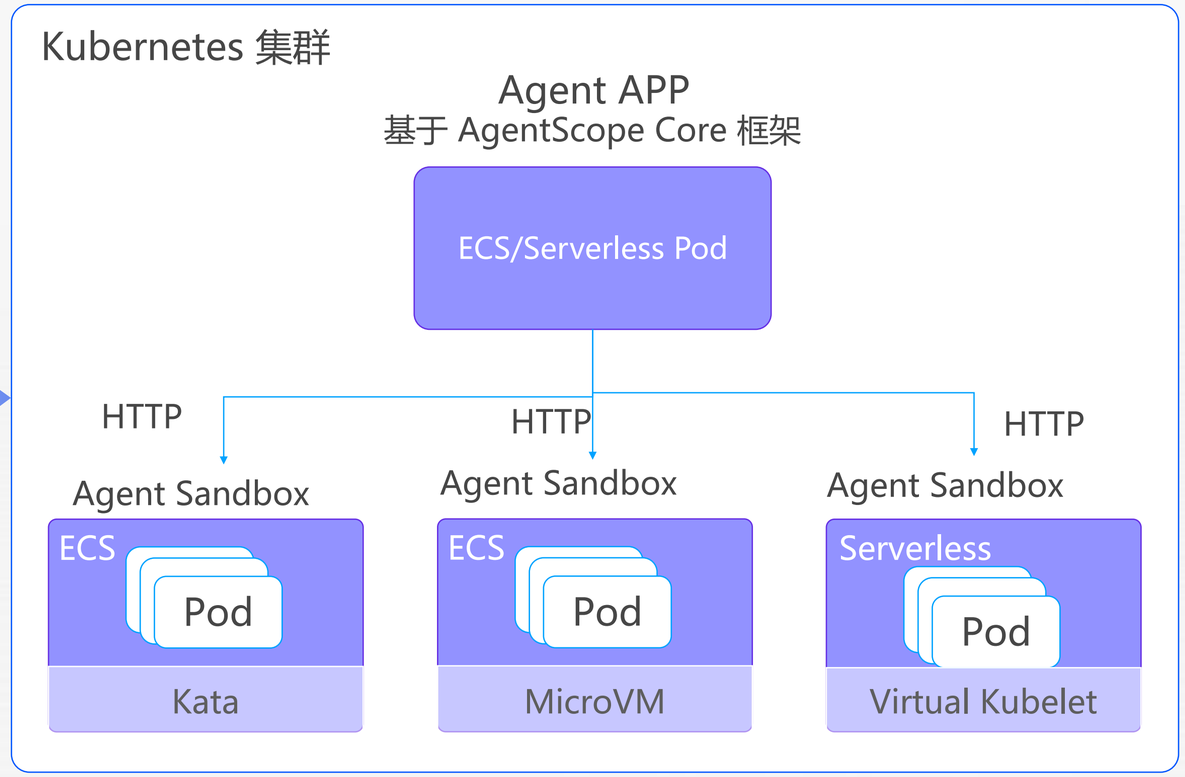
Agent Sandbox生态

Openkruise Agents: 与其它Agent生态的集成



AgentScope on K8s

一键部署



项目地址：https://github.com/agentscope-ai/agentscope

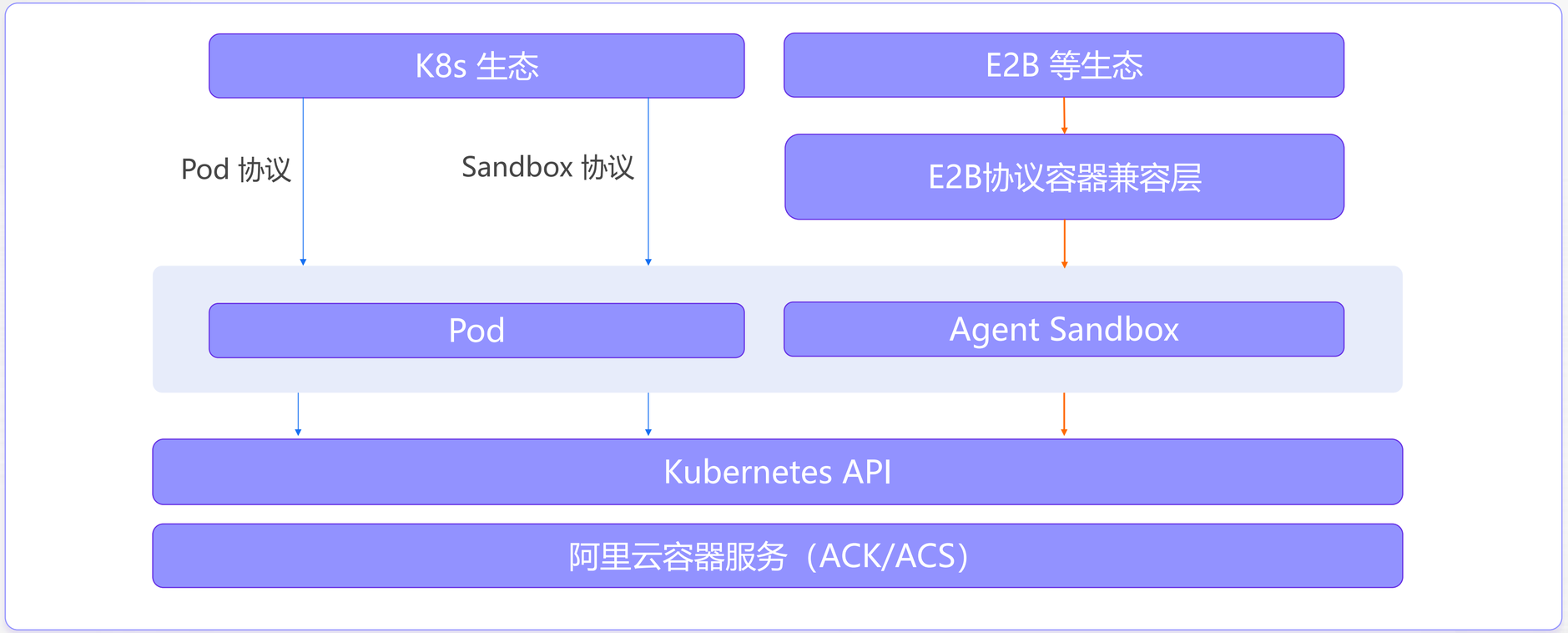
智能体应用on K8s整体概览图

AgentScope

百炼生态
AgentScope/百炼插件

Langchain

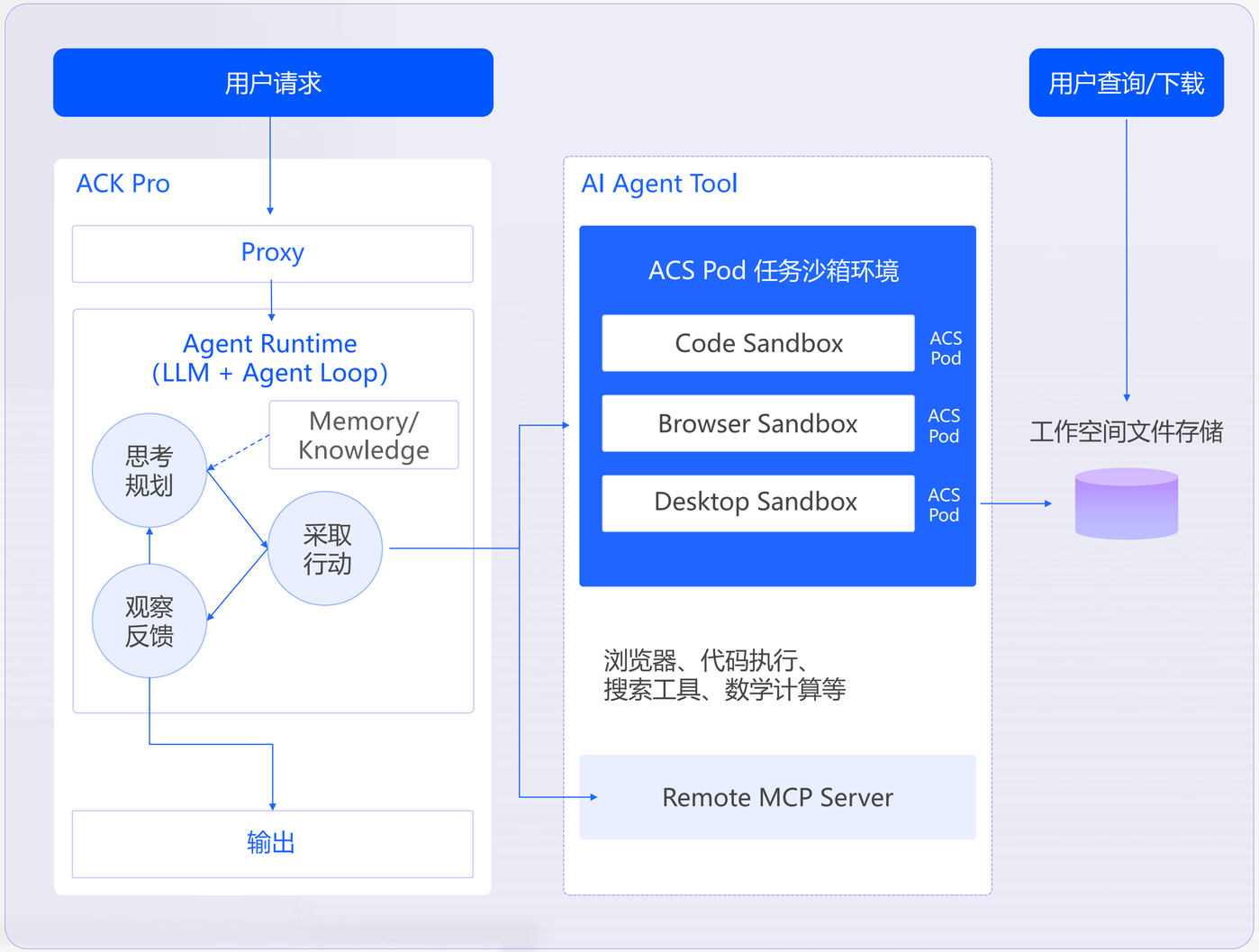
自建应用层



05

客户案例

某客户案例：基于阿里云容器服务构建AI Agent业务



客户痛点

不可信 Tool 工具调用执行的安全风险、漫长的冷启动导致的响应延迟、闲置的高资源成本。

解决方案

强安全隔离：基于 MicroVM 沙箱技术，ACS Pod 为每个 Agent 任务提供独立的、硬件级别的计算安全隔离环境。同时，结合 Network Policy、Fluid 等能力增强，提供 Pod 级别网络、存储的端到端安全运行环境。

极致弹性速度：通过用户负载特征的预调度优化、容器镜像缓存加速等，ACS Pod 支持秒级快速启动，15000 Pod/分钟极致弹性，提升大规模并发的 Agent 任务响应体验。

普惠易用：支持 0.5 vCPU 1 GiB 精细化步长递进，同时支持秒级按需热变配，可根据 AI Agent 真实资源需求按需使用 ACS CPU/GPU Pod，降低综合资源成本。整体是 Serverless 模式，无需运维节点，普惠易用。

客户价值

无需预先采购服务器，当 Agent 任务触发时，按需拉起海量 ACS Pod，整个任务执行是计算、网络、存储的安全强隔离环境；任务结束，Pod 自动释放。整体方案降低资源及运维管理成本。

极客邦科技 2026 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：12月18-19日

- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

THANKS

探索 AI 应用边界

Explore the limits of AI applications

AiCon

全球人工智能开发与应用大会