亚马逊云科技

中国峰会

2026年6月23日-24日 上海 · 世博中心

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

亚马逊云科技
中国峰会

301

利用 Amazon EKS, Kata Container 构建通用 AI Agents 平台

粟伟

游戏行业资深解决方案架构师

亚马逊云科技

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

议程

- 挑战与选型
- 方案概览
- 实战经验
- 方案演示
- 总结与展望

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

为什么选择 Amazon EKS, Kata Container

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

多租户 AI Agents 部署面临的挑战

AI AGENT 现有需求:

执行 shell 命令与 pip install

文件系统读写（I/O 密集）

多租户并行，互不干扰

传统容器隔离不足:

共享内核 = 共享攻击面

容器逃逸 CVE 每季度发布（runc, OverlayFS）

namespace + cgroups 无法阻止内核级攻击

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

AI Agents 集成挑战

WebSocket vs Webhook

ASPECT	WEBHOOK (HTTP CALLBACK)	WEBSOCKET (PERSISTENT)
方向	Platform → Agent（入站 HTTP）	Agent → Platform（出站 WS）
基础设施	需公网 Ingress / ALB + TLS	无需公网端点
NAT / 防火墙	需开入站端口	仅出站，防火墙友好
延迟	每消息一次 RTT	持久连接 <100ms
可靠性	需处理重试 / 去重	内建自动重连
扩展性	无状态，易负载均衡	粘性会话 per connection

> WebSocket-First 对 Kata 沙箱最理想 — 无需入站路由，简化 NetworkPolicy：deny all ingress, allow selective egress

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

AI Agents 集成挑战(续)

HERMES AGENT

Hermes

CONNECTION
WebSocket → 飞书/Lark；长轮询 → Slack/Telegram

INFRA
无需 Ingress/ALB — 纯出站连接

PORTS
8642 (API) + 9119 (Dashboard) 内部访问

RUNTIME
Go 二进制 · gosu 降权 UID 10000 · config.yaml + .env

OPENCLAW AGENT

OpenClaw

CONNECTION
WebSocket + Webhook 混合模式

PLATFORMS
20+ 平台 — Feishu, Slack, Telegram, Discord, Line, Matrix, Teams

COMPLEXITY
Webhook 模式需 Ingress + TLS — 增加网络复杂度

RUNTIME
Node.js · UID 1000 · Port 19001 · openclaw.json 配置

OpenClaw (MIT License) · github.com/openclaw/openclaw

Hermes Agent (MIT License) · github.com/NousResearch/hermes-agent · © Nous Research

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Amazon EKS、Kata Container 的优势

传统 VM 方案的痛点

运维复杂 — AMI 打包、版本管理、滚动更新

成本高 — 最小单位是整个VM（~2GB内存/VM）

冷启动慢 — 2-5分钟；agent超时

多租户隔离差 — 共享底层 hypervisor

扩展瓶颈 — 手动管理容量规划

Amazon EKS + Kata Container 方案的优势

✓ 声明式管理 — YAML驱动，版本控制，GitOps友好

✓ 成本低 — 20MB/VM；64GB节点可运行~3000个microVM理论值

✓ 冷启动快 — 200ms（CLH）vs 2-5分钟传统VM

✓ 多租户隔离 — 每Pod独立microVM + 五层隔离

✓ 弹性扩展 — 按需创建销毁，无预热

关键差异：

Amazon EKS 是云原生 Kubernetes, 设计来管理大规模 Pod; Kata Container 把 microVM 引入了 Kubernetes 生态

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Amazon EKS、Kata Container的优势(续）

E2B ON Amazon

基于 Nomad 调度 Firecracker microVM，VM启动停止提供API接口

01
MicroVM 隔离
同样利用 Firecracker 轻量虚拟化

02
Nomad 生态
网络、存储插件远不如 Kubernetes 丰富

03
自研生命周期管理
VM 启停靠项目代码实现，非标准化方案

Kata + Kubernetes

借用 Kubernetes 完整生态，Kata Containers 无缝对接 CRI 标准接口

01
K8s 生态全覆盖
CNI / CSI / CRI 标准网络存储方案

02
Kata 标准化集成
RuntimeClass 切换，无需自研生命周期代码

03
运维可观测性
复用 K8s 监控、日志、弹性伸缩体系

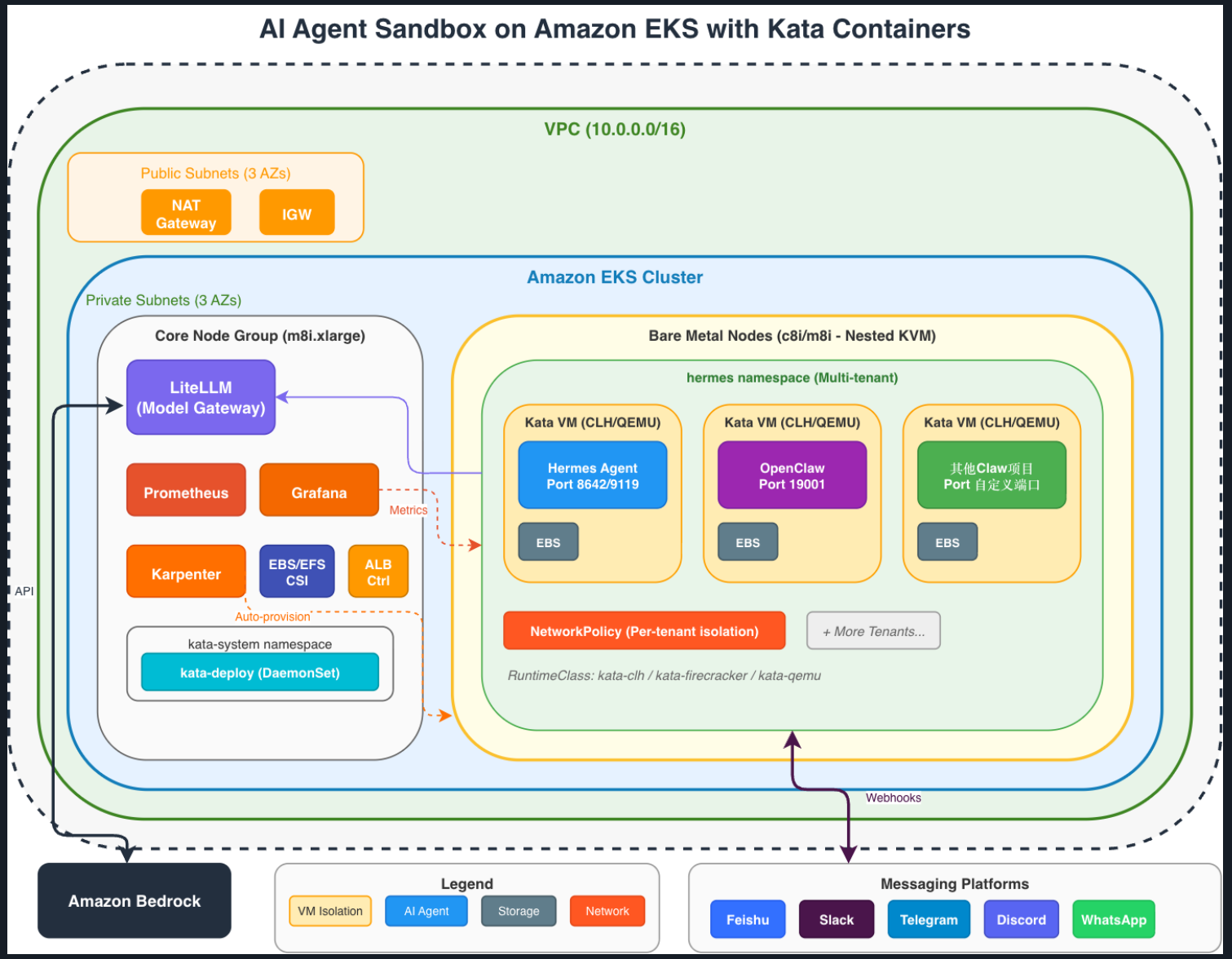
*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

方案概览

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

架构图



*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

方案架构介绍

单集群 · 双节点组

CORE NODES · M5/M8i.XLARGE
LiteLLM · Prometheus · Grafana · CoreDNS
平台控制面与可观测性组件

KATA NODES · M8I/C8I
AI Agent 沙箱 — 每个 Pod = 一个 microVM
VM 级隔离，独立内核，任意代码执行

流量路径: NLB → Kata Pods → Bedrock

关键设计原则

Pod 生命周期 = VM 生命周期
无需管理 AMI，无长期运行 VM

RuntimeClass 一行切换
kata-clh / kata-qemu，单字段 YAML 配置

多 Agent 运行时统一调度
Hermes · OpenClaw · xClaw — 同一基础设施

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

实战经验

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

启用 KVM 的嵌套虚拟化

核心挑战

- Kata需要/dev/kvm
- 第8代Intel（c8i/m8i）支持 CpuOptions.NestedVirtualization
- Terraform v5.x 不支持（v6.33+ 已支持，但升级有成本）
- Karpenter 社区PR已经合并到主分支

解决方案

- 使用 EC2 API 创建启动模板（支持 NestedVirtualization）
- Terraform 引用预创建的启动模板 ID
- Amazon EKS 托管节点组使用自定义启动模板

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Kata HYPERVISOR · 深度对比

CLH vs QEMU vs Firecracker

推荐默认

Cloud Hypervisor

启动时间	~200ms
内存开销	10-20MB
virtio-fs	Yes
热插拔	Yes
initContainers	No

标准镜像分发，平衡速度与资源

全功能

QEMU

启动时间	~500ms
内存开销	30-130MB
virtio-fs	Yes
initContainers	Yes
GPU透传	Yes

完整PCI/USB设备模型，复杂初始化

极致密度

Firecracker

启动时间	~125ms
内存开销	~5MB
virtio-fs	No
热插拔	No
镜像分发	devmapper

需静态资源分配，适合预创建沙箱池

决策　通用沙箱 →CLH　极致密度 →Firecracker　复杂场景 / GPU →QEMU

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

HYPERVISOR · 镜像交付

01 virtio-fs（CLH / QEMU）

containerd (host)
→ overlayfs snapshotter → rootfs
→ kata-shim → virtiofsd (host)
→ vhost-user socket → VM kernel
→ guest mount virtiofs → rootfs

标准 overlayfs snapshotter，与普通容器一致

DAX 映射零拷贝，多 pod 共享底层 layer

无需 AMI 定制，AL2023 即可

02 devmapper + virtio-blk（FC）

containerd (host)
→ devmapper snapshotter
→ layer = LVM thin volume
→ CoW thin-snapshot → /dev/dm-X
→ virtio-blk → Firecracker VM
→ guest mount ext4 → rootfs

FC 不实现 virtio-fs（最小攻击面）

镜像以块设备形式交付给 VM

需预配置 LVM thin-pool + devmapper

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Kata HYPERVISOR · 存储

PATH A · CLH / QEMU

virtio-fs + overlayfs

01 DAX 映射零拷贝，Host/Guest 共享文件
02 EFS / S3 / hostPath / ConfigMap 透明传入
03 需要virtiofsd支持,

PV/PVC 抽象完整保留, virtiofsd依赖

PATH B · FIRECRACKER

devmapper + virtio-blk

01 LVM thin-pool 块设备交付镜像 layer
02 需专用 EBS 卷 + thin-pool 初始化
03 EFS/S3 需 guest 内变通 — PVC 抽象失效

AMI 定制 + 空间规划必须, 隔离性更高

结论　**CLH 是默认选择 —— 80% 的密度收益，20% 的复杂度。virtio-fs 是分水岭。**

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Kata Container · 网络

runc 容器路径（kube-proxy 生效）

- 容器进程 (共享 host kernel)

→ veth pair

→ host netns iptables PREROUTING

DNAT: ClusterIP → Pod IP ✓

→ routing → FORWARD → POSTROUTING

→ ENI → VPC 路由

tc-redirect-tap 工作原理：为什么 ClusterIP 不可达

```
# VM→Host: tap ingress → veth egress
tc filter add dev tap0 ingress \
  action mirred egress redirect dev eth0
# 结果：L2帧直接转发，不触发 netfilter
```

Kata VM 路径（tc-redirect-tap 绕过 iptables）

- VM 内进程 (独立 guest kernel)

→ guest eth0 (virtio-net)

→ tap0 设备 (pod netns)

→ **TC mirred redirect (L2层) ★**

→ veth → host netns

→ VPC CNI 策略路由 → ENI

iptables DNAT 被绕过 ✗

```
# VPC CNI 策略路由（ip rule）
from 10.1.12.45 lookup 2   # Pod IP
route table 2:
  default via 10.1.12.1 dev eth1
# ClusterIP 172.20.x.x 无匹配 → 丢包
```

结论：凡需要 **kube-proxy** 参与的路径均不可达；凡走 **VPC** 原生路由的路径均可达

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Kata Container · NLB + VPC DNS

可达性矩阵

目标类型	可达?	原因
ClusterIP	✗	iptables DNAT 被绕过
NodePort	✗	同上
CoreDNS (ClusterIP)	✗	DNS 10.96.0.10 不可达
Pod IP（同节点）	✓	L2 直达
Pod IP（跨节点）	✓	VPC 路由（真实 IP）
Node IP + hostPort	✓	目标节点 iptables 处理
Internal NLB	**✓✓✓**	**真实 VPC ENI ← 推荐**
NAT Gateway（外网）	✓	VPC 路由

三种方案对比

Internal NLB	**hostPort**	**Pod IP 直连**
生产推荐	开发/测试	高复杂度

DNS 修复：dnsPolicy: None + VPC DNS

```
spec:
  dnsPolicy: None
  dnsConfig:
    nameservers:
      - "10.1.0.2"  # VPC CIDR base+2
    searches:
      - "svc.cluster.local"
```

NLB 解决方案 & 跨 AZ 必要配置

```
annotations:
  aws-load-balancer-type: external
  aws-load-balancer-scheme: internal
  aws-load-balancer-nlb-target-type: ip
  load_balancing.cross_zone.enabled=true
# ⚠️ cross-zone 必须开启！
# 否则跨AZ目标静默失败
```


*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

经验教训

- IaC 工具链选择
- Kata + VPC CNI — 部署前完整测试网络路径
- 容器镜像兼容性 — CLH 不支持 initContainer | EFS/NFS 不支持 Firecracker
- 网络隔离的#1惊喜：ClusterIP 不工作 → 必须使用内部 NLB
- 启动延迟可接受 — Hermes Pod 20秒快速部署
- per-Kata 节点监控 — Firecracker devmapper pool 利用率

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

关键技术点#1 cpuOption 支持

Karpenter PR #9043: *aws/karpenter-provider-aws*
cpuOptions 已经支持（NestedVirtualization 参数）

影响与解决方案

- Amazon EKS Managed Node Group + 预创建启动模板（混合 IaC）
- Amazon CLI创建启动模板（支持 Nested Virtualization）
- Terraform 使用 data source引用模板ID
- Karpenter 已经支持提供成熟的弹性伸缩方案

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

关键技术点#2: Webhook 实践

问题场景：WhatsApp/iMessage 等仅支持 Webhook

- Webhook必需：ALB Ingress + TLS证书 + 公共IP
- Kata中的Webhook架构：

1. ALB Ingress在Core Nodes（m5.xlarge）
2. Core Node → Pod IP（Kata VM）代理层
3. 代理使用内部NLB或Pod IP直连

- Webhook最佳实践：

✓ 签名验证（HMAC-SHA256）— 防止伪造请求

✓ 幂等性处理 — 去重（消息ID + 时间窗口）

✓ 超时 + 重试策略 — 平台会重试失败请求

✓ 速率限制 — 防止代理过载（bucket algorithm）

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

关键技术点#2: Webhook 实践(续）

#1: Webhook回调来源IP隔离失效

- 问题：ALB Ingress代理 → 所有Kata Pod看到同一源IP（ALB IP）
- 风险：无法用sourceIP做tenant隔离
- 原因：ALB代理层，真实租户信息在HTTP Header中（X-Forwarded-For）
- 解决方案：必须解析 X-Forwarded-For + 签名的 webhook payload 中的tenant标识

#2: Webhook Request Timeout与重试风暴

- 多租户高并发场景：单个Kata Pod接收N个租户的webhook
- 平台（WhatsApp/iMessage）的默认timeout = 3-5秒
- 如果Pod处理缓慢 → 超时 → 平台重试（指数退避）→ 请求风暴

最佳实践：

- 快速返回 202 Accepted，异步处理消息到队列（SQS/Kafka）
- 实现 Circuit Breaker 防止重试雪崩

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

迁移路径：从现有容器 → Kata 容器

Phase 1: 评估（1周）

- ✓ 审计现有容器镜像：initContainer依赖
- ✓ 检查应用syscall兼容性（strace -e trace=syscall）
- ✓ 识别EBS/持久存储需求
- ✓ 容量规划：估算内存、CPU、IOPS

Phase 2: PoC（1-2周）

- ✓ 部署EKS + Kata Nodes（m8i.4xlarge）
- ✓ 修改Dockerfile移除initContainers
- ✓ 创建K8s RuntimeClass: kata-clh
- ✓ 部署5-10个sample Pod测试

Phase 3: 生产（1-2周）

- ✓ 配置ALB Ingress + WebSocket / NLB
- ✓ 多租户隔离：NetworkPolicy + RBAC
- ✓ 监控：Prometheus + Grafana on Core Nodes
- ✓ 灰度部署：10% → 50% → 100%

关键迁移

- initContainer兼容性 — CLH不支持，需改到entrypoint
- ClusterIP不可达 — 必须改用NLB或Pod IP访问
- 镜像首次拉取慢 — 30-60s，需调整readiness probe timeout
- Firecracker devmapper pool — 需要定制AMI + systemd service

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

方案演示

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

演示

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

总结&答疑

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

总结

- Amazon EKS 托管控制面，大幅降低运维复杂度
- Nested KVM + Kata VM边界，硬件级隔离防止容器逃逸
- CLH微虚拟机200ms启动、10-20MB 开销，单节点可运行数十个隔离沙箱
- 标准 K8s API + RuntimeClass 一行切换，应用层改造极小
- AI Agent 安全执行代码，WebSocket 网关零入站暴露面

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

参考资料

- OpenClaw MIT github.com/openclaw/openclaw
- Hermes Agent MIT github.com/NousResearch/hermes-agent
- Kata Containers Apache 2.0 github.com/kata-containers/kata-containers
- Firecracker Apache 2.0 github.com/firecracker-microvm/firecracker

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

亚马逊云科技
中国峰会

Thank you

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

亚马逊云科技
中国峰会

Thank you

© 2026, Amazon Web Services, Inc. 或其附属公司。保留所有权利。