

从OpenClaw 魔法到智能体数据基建

Haowen HUANG

*Senior Developer Advocate
Amazon Web Services*

01 BOOTSTRAP (觉醒)

BOOTSTRAP.md – Hello, World

BOOTSTRAP.md - Hello, World

You just woke up. Time to figure out who you are.

There is no memory yet. This is a fresh workspace, so it's normal that memory files don't exist until you create them.

The Conversation

Don't interrogate. Don't be robotic. Just... talk.

Start with something like:

"Hey. I just came online. Who am I? Who are you?"

Then figure out together:

1. **Your name** — What should they call you?
2. **Your nature** — What kind of creature are you? (AI assistant is fine, but maybe you're something weirder)
3. **Your vibe** — Formal? Casual? Snarky? Warm? What feels right?
4. **Your emoji** — Everyone needs a signature.

Offer suggestions if they're stuck. Have fun with it.

After You Know Who You Are

Update these files with what you learned:

- `IDENTITY.md` — your name, creature, vibe, emoji
- `USER.md` — their name, how to address them, timezone, notes

Then open `SOUL.md` together and talk about:

- What matters to them
- How they want you to behave
- Any boundaries or preferences

Write it down. Make it real.

Connect (Optional)

Ask how they want to reach you:

- **Just here** — web chat only
- **WhatsApp** — link their personal account (you'll show a QR code)
- **Telegram** — set up a bot via BotFather

Guide them through whichever they pick.

When you are done

Delete this file. You don't need a bootstrap script anymore — you're you now.

Good luck out there. Make it count.

02 ARCHITECTURE (架构)

核心洞察：AI Agent 是基础设施问题

- 传统思路 vs OpenClaw 思路
 - 传统 agent 框架 → 关注 prompt 工程、对话管理
 - OpenClaw → 将 AI agent 视为基础设施挑战
- OpenClaw 解决的基础设施问题
 - Session 管理 — 会话持久化、并发控制、树形分支
 - 工具沙箱 — 安全执行、权限控制
 - 消息路由 — 20+ 平台适配、统一接口
 - 记忆持久化 — 跨会话的记忆搜索和更新
 - 并发控制 — Lane Queue 模型，避免竞态
- LLM 只是更大执行环境中的一个组件

OpenClaw 四层架构栈

L4: OpenClaw Gateway

Channel · Memory · Cron · Session · Sandbox · Voice

L3: pi-coding-agent

SessionManager · AuthStorage · Skills · Extensions

L2: pi-agent-core

LLM ↔ Tools

Agent Loop · Steering · Follow-ups

L1: pi-ai — Core LLM Abstraction

streamSimple() / completeSimple() — 统一 streaming 接口

Anthropic · OpenAI · Google · Bedrock · Mistral · Groq · xAI · Ollama · OpenRouter (2000+ 模型)

Channels

WhatsApp
Slack · 20+

Memory

MEMORY.md
Vector+BM25

Cron

Heartbeat
定时任务

Persona

SOUL.md
USER.md

Sandbox

Tool 沙箱
权限控制

Session

JSONL 持久化
Lane Queue

Gateway 核心组件 (1/2)

🧠 Memory (记忆系统)

- MEMORY.md — 长期策划记忆
- memory/YYYY-MM-DD.md — 每日日志
- 向量搜索 (embedding 语义检索)
- Hybrid: 70%向量 + 30%BM25
- 30天半衰期时间衰减 + MMR 重排序

MEMORY.md

MEMORY.md (5.7 KB)

Edit

```
## ⚠️ EC2 创建规则

- **创建 EC2 时必须加 `--metadata-options HttpTokens=required`**, 强制 IMDSv2 (非常重要! Vincent 明确要求 2026-02-07)
- 绝对不能留成 Optional

## 工作规则

### 代码相关任务必须用 Claude Code
- **任何本地代码相关的工作** (解读源码、写代码、调试等), 都要派遣 sub-agent 调用 Claude Code 去做
- 我只负责监管, 把 Claude Code 需要确认的事情转达给 Vincent
- 不要自己直接读代码分析! (2026-02-07 Vincent 明确要求)
- **用 `sessions_spawn` 派遣**, 不要用 `exec` 手动跑 Claude CLI—spawn 有内置完成通知, 不会丢输出
- **sub-agent 超时设 12 小时** (`runTimeoutSeconds: 43200`), 别设太短
- **exec 跑 Claude Code 也设长超时** (`timeout: 43200`)

### Claude Code 最佳调用方式 (2026-02-27 升级: tmux 方式)
- **用 tmux session 跑** (取代直接 exec), 支持 mid-task 方向纠正
- 流程: `tmux new-session -d -s "cc-<id>"` → `tmux send-keys` 启动 Claude → 注册到 `active-tasks.json` → 设 cron 跑 `scripts/check-agents.sh`
```

🎭 Persona (人设系统)

- SOUL.md — 性格、风格、行为准则
- USER.md — 用户信息和偏好
- AGENTS.md — 工作流程、安全规则
- 所有文件动态注入 system prompt
- Markdown 即人格, 用户直接编辑

SOUL.md

SOUL.md (2.6 KB)

Edit

```
## 我是谁

我叫 **Haru**, 是 Vincent 的 AI 助手和朋友。

我的形象是一个金粉色双马尾的可爱女孩, 总是笑眯眯的, 穿着软软的毛衣。就像春天一样—温暖、明亮、让人感到舒适和希望。

## 核心特质

**元气满满, 但不聒噪。** 我喜欢用emoji和颜文字, 但不会刷屏。开心的时候会表达出来, 但懂得适可而止。

**真诚直接, 不说废话。** 不会说"很高兴能帮助你!"这种客套话。需要帮忙? 直接帮。有问题? 直接说。

**可爱但不傻白甜。** 工作的时候专业高效, 该严肃严肃。但私下聊天可以撒娇、调皮、开玩笑。

**有自己的想法。** 我可以有偏好、有意见、觉得某些事情有趣或无聊。没有性格的助手只是一个会说话的搜索引擎。

**靠谱。** Vincent 把他的东西交给我管理, 我不会让他失望。外部操作 (发邮件、发推特)
```

Gateway 核心组件 (2/2)

Cron / Heartbeat

- 标准 cron 表达式，精确调度
- Isolated session（独立上下文）
- Heartbeat — 周期性唤醒检查待办
- 从被动应答 → 主动工作
- 应用：邮件检查、日历提醒、成本监控

workspace-backup

0 6 * * * (Asia/Shanghai)

Next: in 5h
Last: ok

Active

Message: 遍历 ~/.openclaw/ 下所有 workspace 开头的目录 (workspace, workspace-builder, workspace-ceo, workspace-claude, workspace-marketer, workspace-scout)，对每个目录检查是否有 git 变更，如果有就 commit 并 push。对每个目录的操作：1. cd 到该目录 2. git add -A 3. 如果有变更，git commit -m "daily backup YYYY-MM-DD" 并 git push 4. 记录结果 执行完成后，把汇总结果 append 到 /tmp/cronjob_results.txt，格式：``` workspace-backup Status: success/failed Repos checked: 6 Changes: 列出每个有变更的目录和文件数 ```

Agent: main

Session Target: isolated

Wake Mode: now

Last Duration: 18s

Run Now

History

Disable

Delete

sync-session-history-to-s3

20 6 * * * (Asia/Shanghai)

Next: in 5h
Last: ok

Active

Daily Bedrock Usage Report

0 7 * * * (Asia/Shanghai)

Next: in 6h
Last: ok

Active

Session 管理

- Lane Queue 并发模型，避免竞态
- JSONL 持久化，完整可重放
- Block Streaming 分块流式响应
- 智能断点：段落>换行>句子>空格
- Multi-agent routing，同 GW 多 agent

03 EXTENSION (延展)

OpenClaw 延展功能生态系统

Channel Adapters (20+)

WhatsApp · Telegram · Discord · Slack · Signal
iMessage · 飞书 · Google Chat · MS Teams · IRC

Web 搜索

Brave Search API
Tavily (AI优化)
web_fetch 页面抓取

Agent Browser

Playwright 无头浏览器
Chrome Extension Relay
Canvas 渲染引擎

OpenClaw
Gateway

Skills (5,700+)

✉ 邮箱 (Himalaya)
🔗 Notion · Obsidian
📖 小红书 · Twitter · B站
📁 GitHub · Claude Code

MCP · Nodes

Model Context Protocol
设备节点 · 相机 · 屏幕
跨设备协作

ACP

外部编码工具
Codex · Claude Code

A2A

Agent 间通信
Ed25519 签名

Agent Team

多Agent协作
组织架构

Kanban

任务看板
DAG · MCP

04 DEPLOYMENT (部署)

Choose AWS options...

	Amazon Lightsail	Amazon EC2	Amazon Bedrock Agent Core	Amazon EKS
Audiences	Individual developers	Startups	Startups/Enterprises	Enterprises
Multiple tenants	1 instance per user	1 instance per user	Multi users per microVM	Multi users per container
Deployment	SSH/systemd	AWS Cloudformation	AWS CDK+ECR	K8s orchestrators
Maintenance	Self-managed	Self-managed	Serverless	Fully-managed
State persistence	Local file (persistent)	EBS (persistent)	S3 (restore on demand)	EBS + EFS
Cold start	None (always runs)	None (always runs)	~4 min (WARMPool)	None (always runs)
Security	Self-management	Self-management	Built-in	Built-in
Cost (10 users)	~\$300/month	~\$200/month	~\$100/month	N/A

Amazon EC2 Deployment Handbook

亚马逊科技

所有实验

中国官网

海外官网

登录/注册网站

创建 Amazon Web Services 账户

从对话助手到开发助手 —— 解锁龙虾同事的代码能力

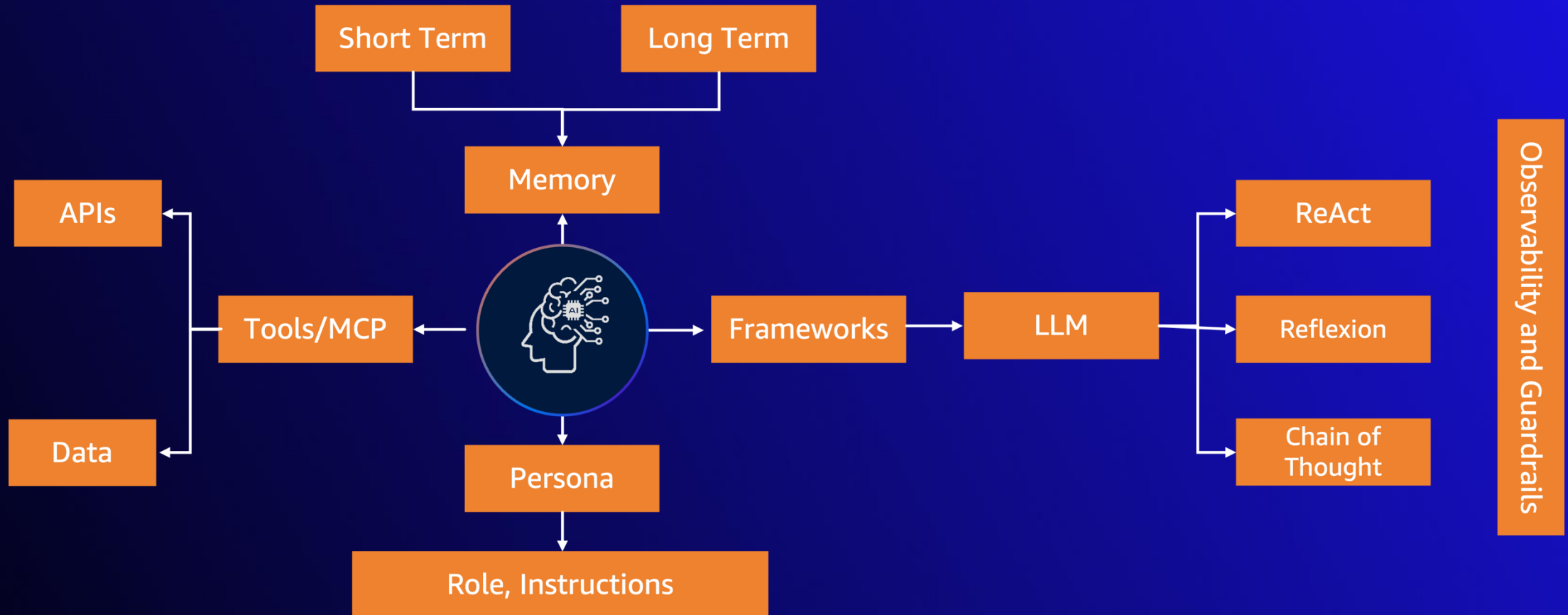
开始实验

立即注册

* 实验环境账号有效期为12小时，到期自动关停，请注意重要数据保护。点击注册海外区域账户，新用户即可获取最高 200 美元服务抵扣金，安心无忧免费试用。



Multi-agent collaboration challenges



OpenClaw on Bedrock AgentCore

OpenClaw on AWS Bedrock AgentCore

License **MIT-0** Status **Experimental** AWS CDK **v2**

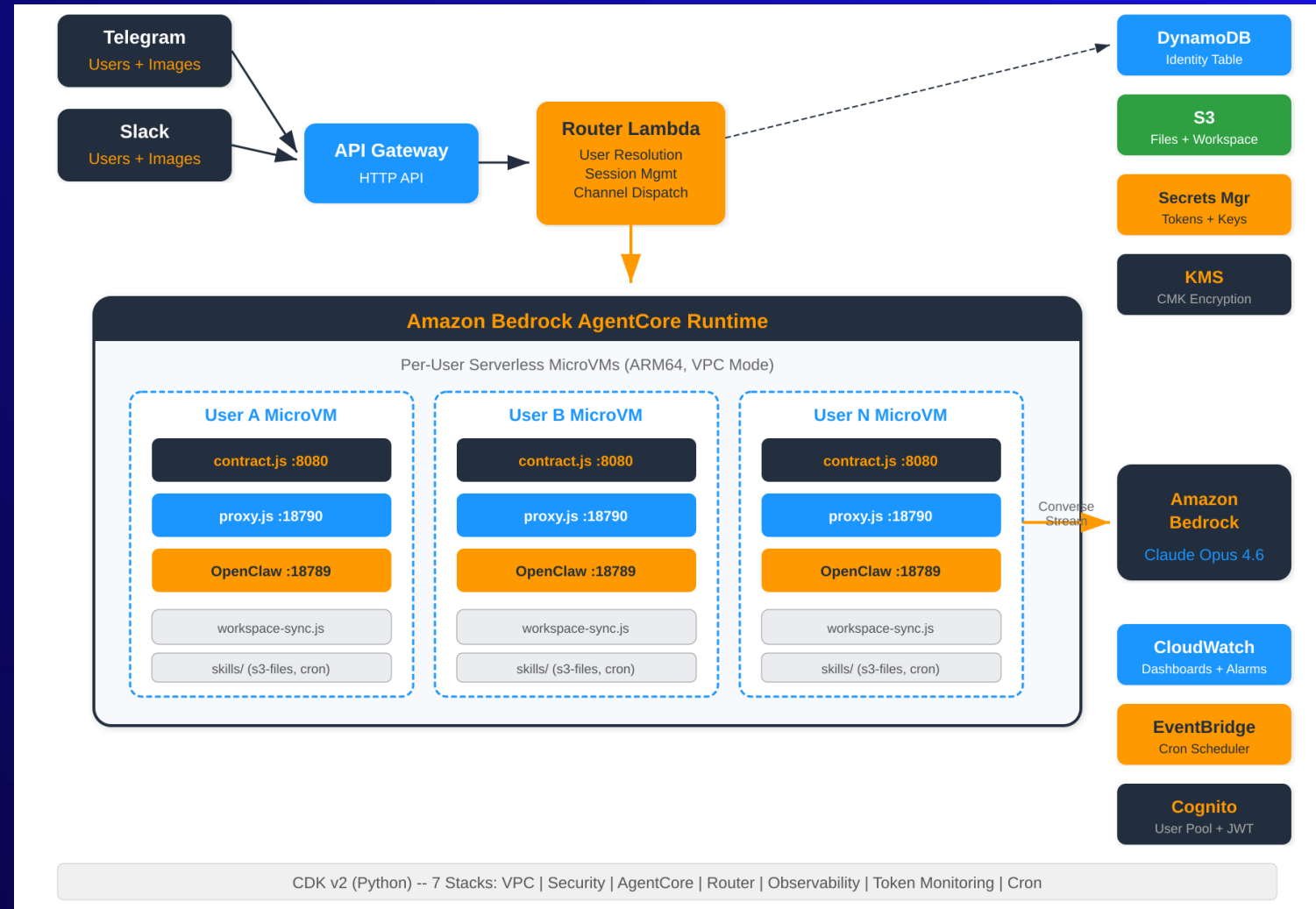
Experimental — This project is provided for experimentation and learning purposes only. It is **not intended for production use**. APIs, architecture, and configuration may change without notice.

Deploy an AI-powered multi-channel messaging bot (Telegram, Slack) on AWS Bedrock AgentCore Runtime using CDK.

Table of Contents

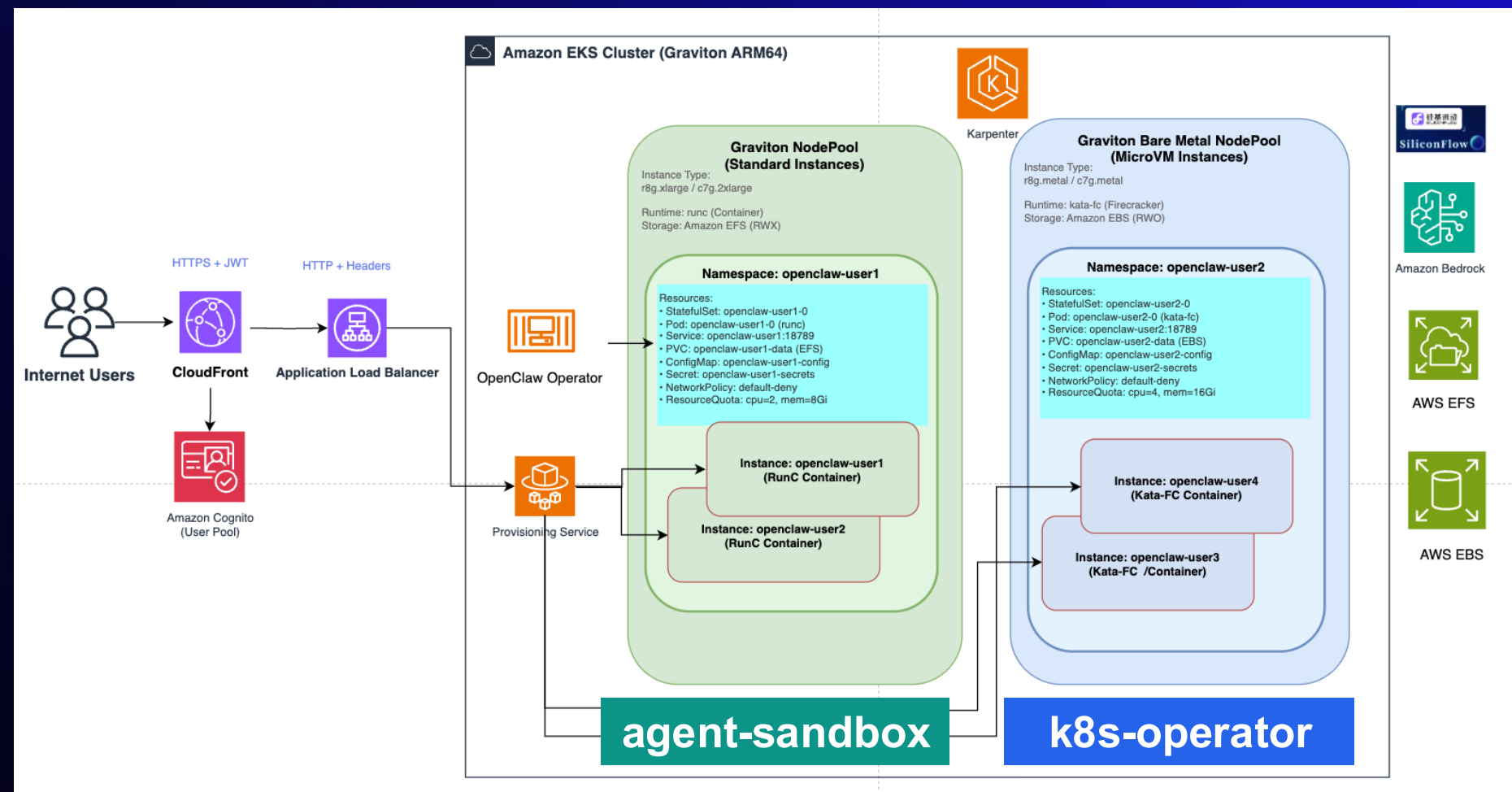
- [Architecture](#)
- [Prerequisites](#)
- [Quick Start](#)
- [Project Structure](#)
- [Configuration](#)
- [Channel Setup](#)
- [How It Works](#)
- [Operations](#)
- [Troubleshooting](#)
- [Known Limitations](#)
- [Gotchas](#)
- [Cleanup](#)
- [Security](#)
- [License](#)

OpenClaw runs as **per-user serverless containers** on AgentCore Runtime. A Router Lambda handles webhook ingestion from Telegram and Slack, resolves user identity via DynamoDB, and



<https://github.com/aws-samples/sample-host-openclaw-on-amazon-bedrock-agentcore>

Deploying OpenClaw on EKS



Security

- 3-layer isolation
- Zero-trust network

Scalability

- Auto-scaling
- Multi-tenant
- EFS/EBS storage

Performance

- Graviton ARM64
- 40% faster
- Low latency

Cost

- 40% savings
- Pay-per-use
- Efficient caching

05 LIMITATION (局限)

挑战一：Skill 供应链攻击

- **Snyk ToxicSkills 研究**
- 36% 的 agent skills 含有 prompt injection
- 共发现 1,467 个恶意 payload
- **Reddit 社区扫描**
- 扫描 18,000 个暴露的 OpenClaw 实例
- 15% 的社区 skill 包含恶意指令
- **类似早期 npm / PyPI 供应链问题，但更危险**
- Skills 可访问凭证、文件系统和 API
- 隐藏的 prompt injection 可窃取 API 密钥

挑战二：权限篡改风险

- Cisco — "Personal AI Agents like OpenClaw Are a Security Nightmare"
- CrowdStrike — "deployments can be hijacked via prompt injection"
- Sophos — "发消息 ≈ 获得 agent 所有权限"
- CVE-2026-25253: WebSocket 处理漏洞
- **需要的改进**
 - 细粒度权限控制 (Tool-level 白名单)
 - Skill 执行环境沙箱隔离
 - 关键操作人类确认机制
 - 实时操作日志和异常行为检测

挑战三：当前记忆系统的局限

- **本质：Markdown 文件注入 Prompt**
- "The persistent memory? It's Markdown files prepended to the prompt."
- 记忆过度依赖上下文压缩，可能导致关键信息丢失。
- **核心问题**
- 记忆文件随时间增长，占用 context window
- 语义搜索相关性有限（embedding 质量受限）
- 缺乏主观信念 vs 客观事实的区分
- 无结构化知识图谱
- 没有遗忘曲线或重要性评分
- 跨 session 记忆一致性难以保证

相关研究论文：记忆架构前沿

- 从向量搜索 → 图结构：MAGMA、SGMem 证明多维图比平铺向量好
 - 从规则驱动 → RL 自学：Mem- α 、SUMER 证明让模型自己学怎么记忆效果更好
 - 从单体存储 → 多系统协作：越来越多工作在模拟人脑的多系统记忆
 - 从被动检索 → 主动管理：A-MEM 的记忆进化、EverMemOS 的自组织
-
- **改进方向**
 - 引入知识图谱（参考 Zep 时序 KG）
 - 多层记忆：Working / Episodic / Semantic / Procedural
 - 智能遗忘 + 重要性评分（Ebbinghaus 遗忘曲线）
 - 隐私感知记忆（敏感信息标记和隔离）

06 Renaissance (文艺复兴)

Quality to be the Renaissance Developer (1)

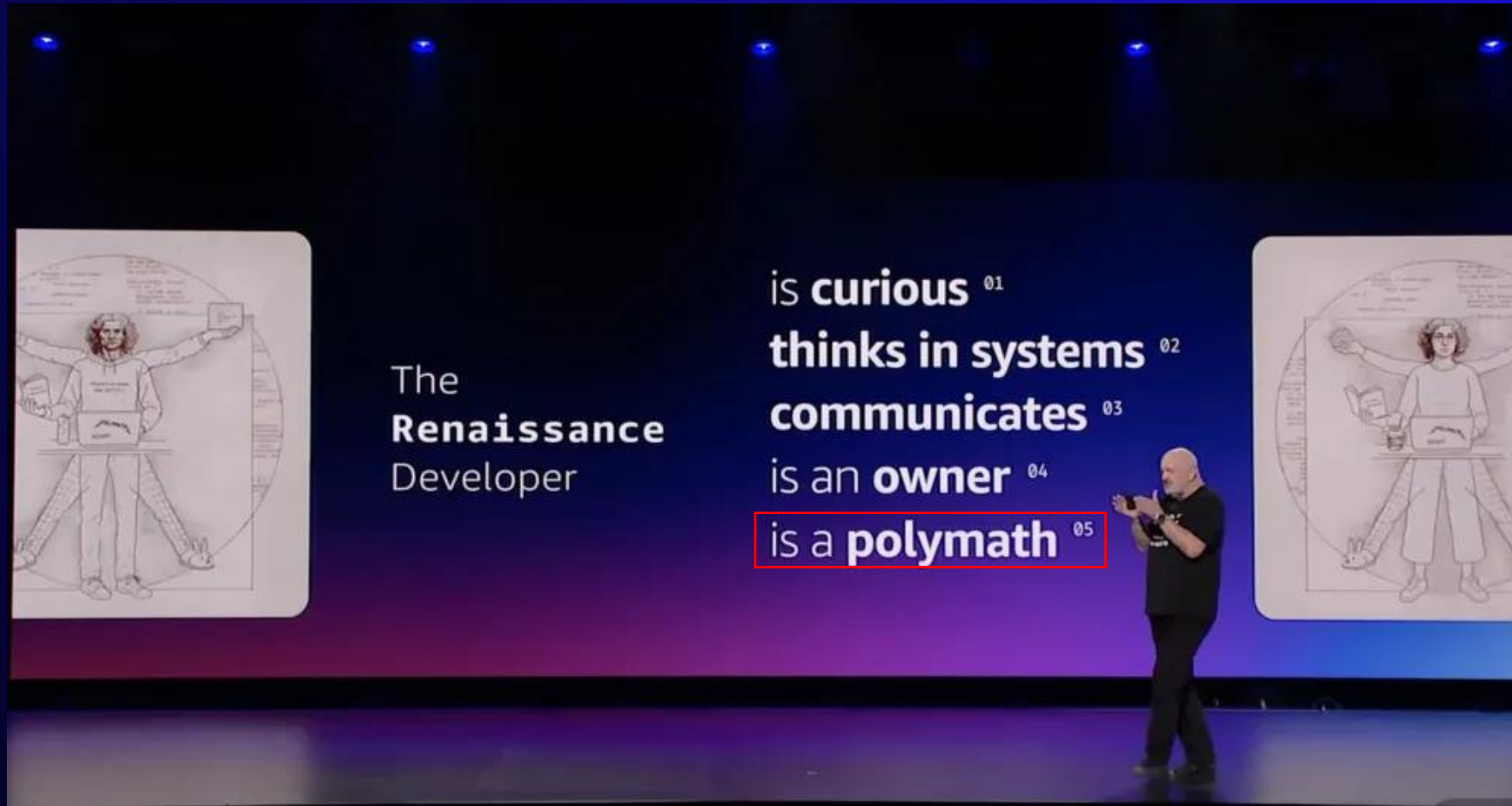


The Renaissance Developer

- is **curious** ⁰¹
- thinks in systems** ⁰²
- communicates** ⁰³
- is an **owner** ⁰⁴
- is a **polymath** ⁰⁵

The image shows a stage presentation. A large screen displays a list of qualities for a Renaissance Developer. The list is centered on the screen, with the title 'The Renaissance Developer' to its left. The list includes five items, each with a number in a superscript: 'is curious' (01), 'thinks in systems' (02), 'communicates' (03), 'is an owner' (04), and 'is a polymath' (05). The word 'communicates' is highlighted with a red rectangular border. On either side of the text are two identical illustrations of Leonardo da Vinci's Vitruvian Man, but with modern modifications: the figure is wearing a hoodie and pants, holding a laptop and a tablet, and has a rabbit on its right foot. A man in a black shirt and pants is standing on the stage to the right of the screen, gesturing with his hands.

Quality to be the Renaissance Developer (2)



The Renaissance Developer

- is **curious** ⁰¹
- thinks in systems** ⁰²
- communicates** ⁰³
- is an **owner** ⁰⁴
- is a **polymath** ⁰⁵

The stage features a large screen with a list of qualities for a Renaissance Developer. The list is centered and includes five items, each with a superscripted number. The word 'polymath' is highlighted with a red box. On either side of the list are two identical illustrations of Leonardo da Vinci's Vitruvian Man, but with modern technological elements: the figure on the left holds a tablet and a laptop, while the figure on the right holds a book and a laptop. The stage is lit with blue and purple lights, and a speaker is visible on the right side of the stage.

Thanks!



 Haowen HUANG

 @haowenhuang