

DaoCloud 企业级 AI 智能体平台

ClawOS 用户手册

企业级 Agent 运行与治理平台

文档名称	ClawOS 操作手册
产品名称	ClawOS（AgentClaw）
适用平台	DCE（WS 模式）
文档版本	v0.5.3
内容范围	介绍、安装、快速入门、工作空间、管理员视角、参考文档、Release Notes、常见问题
整理来源	https://docs.daocloud.io/tf/clawos/
更新时间	2026 年 9 月

本手册基于 DaoCloud 官方文档中心 ClawOS 全站页面整理，供实施与运维人员参考。

目录

一、ClawOS 介绍	3
二、安装	8
三、快速入门	12
四、工作空间视角	16
五、管理员视角	37
六、参考文档	45
七、Release Notes 与升级	49
八、常见问题	51

一、ClawOS 介绍

（一）什么是 ClawOS

ClawOS 是面向企业的 Agent 运行与治理平台，用于统一创建、管理、集成、观测和运维企业内部的 AI Agent。当前版本中，ClawOS 管理的 Agent 类型主要为 OpenClaw 实例；后续平台可扩展支持更多类型的企业 Agent。

可以简单理解为：ClawOS 是企业内部运行 AI Agent 的**控制面**和**治理面**。

它不是一个简单的 Agent 列表管理平台，也不是只负责创建几个智能体实例的后台。它更接近企业 Agent 时代的基础设施：帮助企业把 Agent 安全、稳定、可控地运行起来，并纳入企业已有的权限、网络、协作、审计和运维体系。

如果说 OpenClaw 让 AI Agent 具备了「执行任务」的能力，那么 ClawOS 要解决的是：如何让这种执行能力在企业环境中**可管理、可审计、可集成、可规模化**地落地。

（二）OpenClaw 是什么

OpenClaw（小龙虾）是新一代开源 AI Agent 运行时，目标是将 AI 从「被动回答问题」升级为「主动规划并执行任务」。OpenClaw 不只是和用户聊天，还可以结合模型能力、工具调用、文件系统、终端命令、浏览器控制、API、Skill 和多 Agent 协作，完成更复杂的端到端任务。

能力维度	传统 AI 助手	OpenClaw 智能体
交互方式	被动问答，单轮响应	主动规划，多轮自主执行
能力边界	仅输出文字	调用工具、操作系统、执行代码
适用场景	信息查询、简单生成	端到端自动化工作流

OpenClaw 通常具备以下几类核心能力：

- **任务规划**：理解用户目标，并拆解执行步骤
- **工具调用**：调用文件系统、命令行、浏览器、API、第三方工具
- **Skill 扩展**：通过 Skill 封装可复用能力
- **上下文管理**：在多轮任务中维持任务状态
- **记忆与知识**：结合短期会话、长期记忆或知识库完成任务
- **多模型接入**：可对接不同大模型或企业私有模型

OpenClaw 更像是一个具体能干活的 Agent 运行实例；而 ClawOS 则是管理这些 OpenClaw 实例的平台。

1. ClawOS 与 OpenClaw 的关系

对象	定位
OpenClaw	运行对象，负责让 Agent 具备执行能力
ClawOS	运行和治理平台，负责让 Agent 在企业里可控地运行

在企业场景中，一个团队可能会创建多个 OpenClaw 实例：有的用于总结飞书消息，有的用于处理工单，有的用于知识库问答，有的用于研发辅助，有的用于文档审查。

如果每个 OpenClaw 实例都由个人自行配置、自行维护、自行接入模型和工具，企业很快就会遇到管理问题：权限不可控、成本不可见、日志不可查、网络边界不清、Skill 无法复用、异常无人排查。ClawOS 的价值，就是把这些分散的 Agent 运行起来、管起来、看起来、审起来。

(三) 主要能力

ClawOS 首先解决的是 Agent 企业化运行的问题。个人使用 Agent 时，重点通常是体验是否顺手、模型是否聪明、任务是否完成；但企业使用 Agent 时，重点会变成安全、治理、成本、集成和运维。

1. 实例生命周期管理

用户或管理员可以在 ClawOS 中创建、查看、编辑、删除 OpenClaw 实例。每个实例背后对应实际运行的服务资源，通常运行在 Kubernetes 或 DCE 等云原生底座之上。

当用户修改镜像、资源规格、模型配置、API Key、消息渠道等关键配置时，可能会触发实例重启或短暂不可用。因此，ClawOS 需要把这些运行影响清晰呈现给用户。详细操作请参见本手册「三、快速入门」。

2. 权限与多租户隔离

ClawOS 面向企业组织结构，不同角色拥有不同能力边界：

- 普通用户只能管理自己的实例
- 租户管理员可以查看工作空间内实例运行情况
- 平台管理员可以管理全局资源、策略和配置

这种权限模型保证 Agent 不会变成一个「谁都能乱用」的黑盒工具，而是被纳入企业身份、角色和空间隔离体系。详见本手册「六、参考文档」中的 ClawOS 权限说明。

3. 网络策略治理

Agent 一旦能够访问网络、调用工具、连接系统，就必须有明确的网络边界。ClawOS 可以帮助企业管理 Agent 的网络访问策略，例如：

- 哪些实例可以访问内网服务
- 哪些实例只能访问指定 API
- 哪些策略由平台默认强制启用
- 哪些策略允许用户选择

Agent 的能力越强，越需要清楚地限制它能访问哪里、不能访问哪里。

4. Skill 管理与分发

Skill 是 Agent 能力扩展的重要方式。在个人场景里，用户可以自己安装或编写 Skill；但在企业里，Skill 需要被审核、上架、分发、授权、追踪和下架。

因此，ClawOS 中的 ClawHub / SkillHub 更像是企业能力货架。企业可以把高价值 Skill 沉淀下来，统一管理，并按权限分发给不同团队和实例使用。

5. 消息渠道集成

Agent 不应该只存在于控制台里。真正有价值的 Agent，应该进入员工日常工作流。

ClawOS 可以集成飞书、Teams 等企业消息渠道，让 OpenClaw 实例在群聊、私信、频道等场景中收发消息、处理文件、响应任务。这样，Agent 从「平台里的一个实例」变成了「协作系统里的数字员工」。

6. 可观测、日志与运维

企业环境中，Agent 不能只是能回答，还必须能被排查、被追踪、被审计。ClawOS 需要提供实例状态、运行日志、Session transcript、Trajectory log、Token 用量、调用次数、错误率、告警等信息。

这些能力让管理员知道 Agent 是否正常运行、哪里出现异常、成本是否失控、某次任务到底执行了什么。

（四）核心价值

ClawOS 的核心价值，不是证明某个 Agent 比 ChatGPT、Claude、Codex 或 Cursor 更好用——这些个人 AI 工具解决的是个人效率问题。ClawOS 解决的是企业规模化使用 Agent 的问题。

它让企业能够回答这些问题：

- 我们有多少 Agent 在运行？
- 它们分别服务哪些团队？
- 哪些 Agent 正常，哪些异常？
- 哪些用户和实例消耗最多 Token？
- 哪些 Skill 被频繁使用？

- 哪些模型成本最高？
- 哪些任务失败率异常？
- 哪些操作需要审计和回放？
- 哪些网络访问存在风险？

换句话说，ClawOS 让 Agent 从「个人工具」变成「企业可运营资产」。

(五) 典型应用场景

1. HR 批量简历筛选

面对数十份格式各异的 PDF 简历，让小龙虾自动读取、提取关键技术栈、按岗位要求评分排序，输出结构化评估报告。原来需要半天的筛选工作，几分钟即可完成。



图 1 HR 批量简历筛选运行示例

两份简历对比分析：vLLM 推理优化工程师		
候选人概况		
维度	候选人 A	候选人 B
当前职位	AI 工程师 / 后端架构师	高级后端开发工程师
工作经验	5 年	8 年
教育背景	本科	硕士
技术栈	Python, Golang, Rust, Nix, Docker, StarRocks, Redis, Neovim	Go (精通), Python (熟悉), MySQL, Redis, Kafka
核心领域	AI 工程化、LLM 工具集成、分布式大数据、微服务治理	高并发增长系统、交易所核心业务、性能调优

图 2 HR 批量简历筛选输出结果

注意：简历包含敏感个人信息（PII）。DCE 私有化部署确保数据不出内网，禁止将简历发送至外部公有云 API。

2. 软件开发助手

对接 GitHub 后，配置多 Agent 架构（Main Agent + Research + Reviewer + Codex），让小龙虾分析源码、定位 Bug、创建 Pull Request、执行 CI 规范，打造完全个性化的自动化开发编排器。

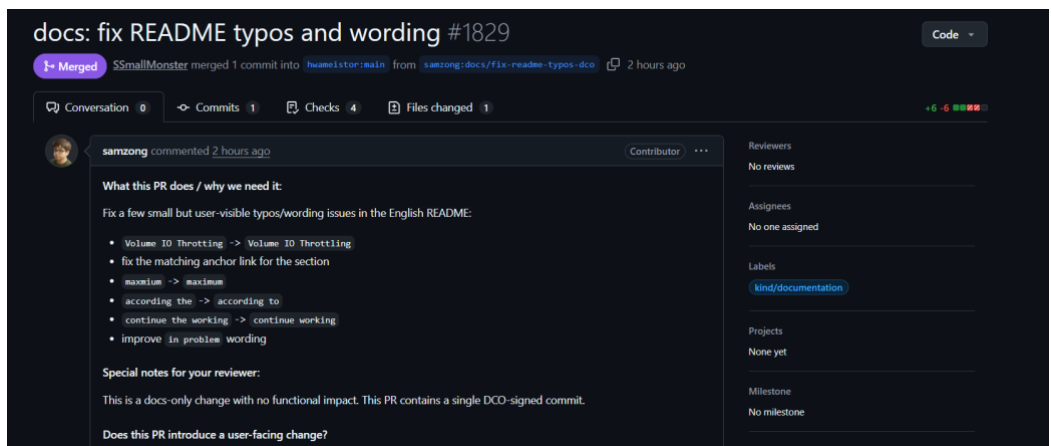


图 3 多 Agent 协作的软件开发助手

3. 批量文档审查

将「读取文档 → 提取关键数据 → 校验规则 → 生成审查意见」全流程封装为可复用的自定义 Skill（[createSkill](#)），一键处理数十份报告并输出汇总 CSV，告别重复劳动。

（六）产品优势

优势	说明
一键创建，开箱即用	无需手动配置环境、接入模型和权限，系统自动完成沙箱创建和 Token 注入，业务人员无需 IT 介入即可立即使用
安全沙箱隔离	每个 OpenClaw 实例运行在独立容器沙箱中，基于 DaoCloud zestU 内核级隔离技术，即使被攻破也严格限制在沙箱内，无法入侵内网或篡改宿主机文件
数据持久化，记忆不丢失	~/openclaw 目录完整持久化在 DCE 存储系统中，对话、配置和记忆在重启、暂停或释放实例后依然保留
安全可控的模型访问	所有模型调用通过 DCE AI 网关统一转发并执行安全策略，不直接暴露公网，数据和对话在平台内受保护
完整后台访问能力	DCE 提供 SSH 登录和 noVNC 网页访问两种方式，支持完整 CLI 操作，满足高级调试和自定义需求

（七）模块指引

- **安装：**ClawOS 的离线与在线安装方式，参见本手册「二、安装」
- **快速入门：**创建并访问第一个 OpenClaw 实例，参见本手册「三、快速入门」
- **工作空间：**概览、OpenClaw 实例、ClawHub、飞书集成、Microsoft Teams 配置，参见本手册「四、工作空间」
- **管理员视角：**平台概览、Skill 管理、网络策略管理，参见本手册「五、管理员视角」

- **参考文档：** Gateway API 实例访问配置、ClawOS 权限说明，参见本手册「六、参考文档」
- **Release Notes 与升级：** 版本演进与升级注意事项，参见本手册「七、Release Notes 与升级」
- **常见问题：** 参见本手册「八、常见问题」

二、安装

本部分说明离线和在线安装 ClawOS 的两种安装方式。

注意： ClawOS 只支持 WS 模式，不支持 CSP 模式。

（一）前置条件

安装前请确保目标 Kubernetes 集群已配置必要的依赖 Helm 应用：

依赖应用	版本要求
kpanda	0.45.0 及以上
ghippo	0.48.0 及以上
insight	0.41.0 及以上
hydra（WS 版本）	0.16.0 及以上

ClawOS v0.4.0 版本需要依赖 insight 0.42.2 版本。

此外还需要完成以下两项集群配置。

1. 更新 hydra AuthorizationPolicy

更新 hydra AuthorizationPolicy，允许 agent-claw-system 的流量。

```
kubectl -n hydra-system edit AuthorizationPolicy hydra-apiserver

apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: hydra-apiserver
  namespace: hydra-system
spec:
  action: ALLOW
  rules:
  - from:
    - source:
      namespaces:
      - dak-system
+   - agentclaw-system
```

2. 配置 *kube-state-metrics* 启动项

配置 insight-system 的 kube-state-metrics 的启动项（需要给部署 Agent 的集群添加此配置）。

```
kubectl -n insight-system edit deployment insight-agent-kube-state-metrics

apiVersion: apps/v1
kind: Deployment
metadata:
  name: insight-agent-kube-state-metrics
  namespace: insight-system
spec:
  template:
    spec:
      containers:
        - name: kube-state-metrics
          args:
            --metric-labels-allowlist=nodes=[feature.node.kubernetes.io/cpu-cpuid.HYPERVISOR]
            --port=8080
            --
      resources=configmaps,cronjobs,daemonsets,deployments,horizontalpodautoscalers,jobs,limitranges,namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes,pods,replicasets,replicationcontrollers,resourcequotas,secrets,services,statefulsets,storageclasses
      + --metric-annotations-allowlist=deployments=[agentclaw.io/instance-name,agentclaw.io/workspace-id]
```

（二）离线安装

1. 下载离线包

使用以下链接下载 ClawOS 离线包：

```
https://qiniu-download-public.daocloud.io/DaoCloud\_Enterprise/agentclaw\_v0.2.3\_amd64.tar
```

如果有更新版本，请替换以上 URL 中的 [v0.2.3](#)。

如果你需要 arm64 版本，请使用链接：https://qiniu-download-public.daocloud.io/DaoCloud_Enterprise/agentclaw_v0.2.3_arm64.tar

2. 从离线包中加载镜像和 *chart* 包

前置条件：将离线包上传至目标节点。

您可以根据下面两种方式之一加载镜像。当环境中存在镜像仓库时，建议选择 **chart-syncer** 同步镜像到镜像仓库，该方法更加高效便捷。

方式一：chart-syncer 同步镜像到镜像仓库

1. 创建 load-image.yaml。该 YAML 文件中的各项参数均为必填项。您需要一个私有的镜像仓库，并修改相关配置。

若当前环境已安装 chart repo，chart-syncer 也支持将 Chart 导出到指定仓库中：

```

source:
  intermediateBundlesPath: clawos-offline
target:
  containerRegistry: 10.16.10.111
  containerRepository: release.daocloud.io/clawos
repo:
  kind: HARBOR
  url: http://10.16.10.111/chartrepo/release.daocloud.io
  auth:
    username: "admin"
    password: "Harbor12345"
containers:
  auth:
    username: "admin"
    password: "Harbor12345"

```

序号	参数	说明
1	intermediateBundlesPath	到执行 <code>charts-syncer</code> 命令的相对路径，而不是此 YAML 文件和离线包之间的相对路径
2	containerRegistry	需更改为你的镜像仓库 URL
3	containerRepository	需更改为你的镜像仓库
4	repo.kind	也可以是任何其他支持的 Helm Chart 仓库类别
5	repo.url	需更改为 chart repo URL
6	repo.auth.username	你的镜像仓库用户名
7	repo.auth.password	你的镜像仓库密码
8	containers.auth.username	你的镜像仓库用户名
9	containers.auth.password	你的镜像仓库密码

若当前环境未安装 chart repo，chart-syncer 也支持将 Chart 导出为 `tgz` 文件，并存放在指定路径：

```

source:
  intermediateBundlesPath: clawos-offline
target:
  containerRegistry: 10.16.10.111
  containerRepository: release.daocloud.io/clawos
repo:
  kind: LOCAL
  path: ./local-repo
containers:
  auth:
    username: "admin"
    password: "Harbor12345"

```

2. 执行同步镜像命令。

```
charts-syncer sync --config load-image.yaml
```

方式二：Docker 或 containerd 直接加载

解压并加载镜像文件。

1. 解压 tar 压缩包。

```
tar xvf agentclaw_0.2.0.bundle.tar
```

解压成功后会得到几个文件：

- hints.yaml：镜像模版
- images.tar：镜像包
- original-chart：chart 包

2. 从本地加载镜像到 Docker 或 containerd。

Docker：

```
docker load -i images.tar
```

containerd：

```
ctr -n k8s.io image import images.tar
```

每个 node 都需要做 Docker 或 containerd 加载镜像操作，加载完成后需要 tag 镜像，保持 Registry、Repository 与安装时一致。

3. 通过 Kpanda UI 安装

在 chart-syncer 成功同步镜像和 chart 后，可以直接在 Kpanda Helm 应用中先更新离线仓库，然后找到 chart [agentclaw](#) 并开始安装，安装时不需要更改任何参数。

4. 通过 Helm 安装

1. 添加 Helm 仓库。

```
helm repo add agentclaw-release https://release.daocloud.io/chartrepo/clawos  
helm repo update
```

2. 安装 Chart。

```
helm upgrade agentclaw agentclaw-release -n agentclaw-system --install --create-namespace
```

若希望 OpenClaw 实例的 UI 与 MS Teams 回调通过 Gateway API 统一入口暴露（而非 NodePort 直连），请参见本手册「六、参考文档」中的 Gateway API 实例访问配置。

（三）在线安装

在线安装的前置条件与离线安装一样。

1. 通过 Kpanda UI 安装

先在 **Helm 应用** → **Helm 仓库** 中添加 [agentclaw-release](#) 仓库：

<https://release.daocloud.io/chartrepo/agentclaw>，然后到 **Helm 模板** 中找到 [agentclaw](#) chart 并点击 **安装**，安装时不需要更改任何参数。

2. 通过 Helm 安装

1. 添加仓库。

```
helm repo add agentclaw-release  
helm repo update
```

2. 安装 Chart。

```
helm upgrade agentclaw agentclaw-release -n agentclaw-system --install --create-namespace
```

三、快速入门

本部分将帮助您在 DCE 上快速创建和使用 OpenClaw 实例。

（一）前提条件

- 拥有 WS Editor、WS Admin 或 Admin 权限
- WS 的资源组中已经绑定 NS
- 已在大模型服务平台的同一个 WS 中创建了 API Key

（二）创建 OpenClaw 实例

1. 从左侧导航栏，选择 **ClawOS**，点击右侧的 **创建实例** 按钮。
2. 配置 OpenClaw 实例后点击右下角 **确定**。
 - 设置实例名称，选择对应的集群、命名空间、镜像版本，配置 CPU/内存资源、网络、模型（推荐使用 minimax 模型）和 API Key。
 - （可选）启用 **集成飞书** 并填入飞书的 App ID 和 App Secret。

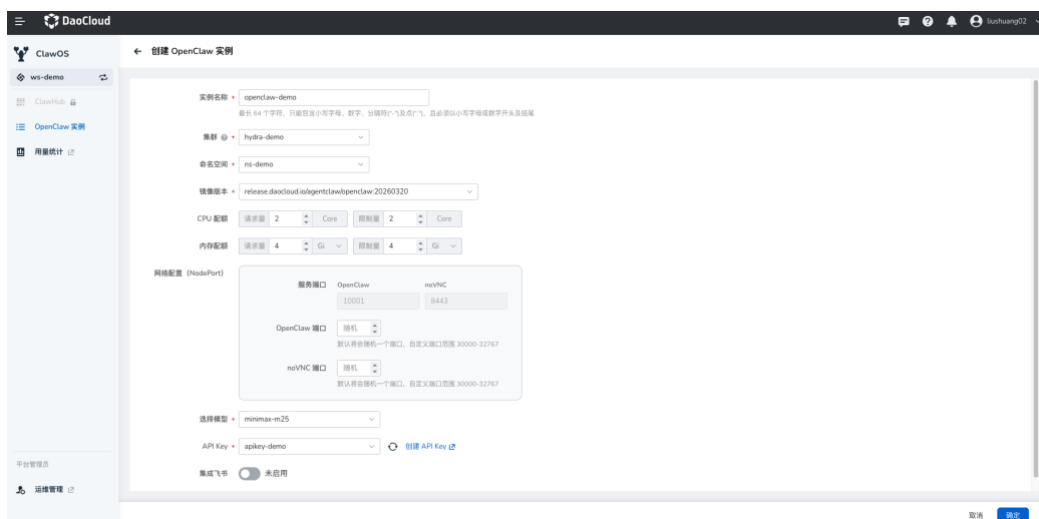


图 4 创建 OpenClaw 实例

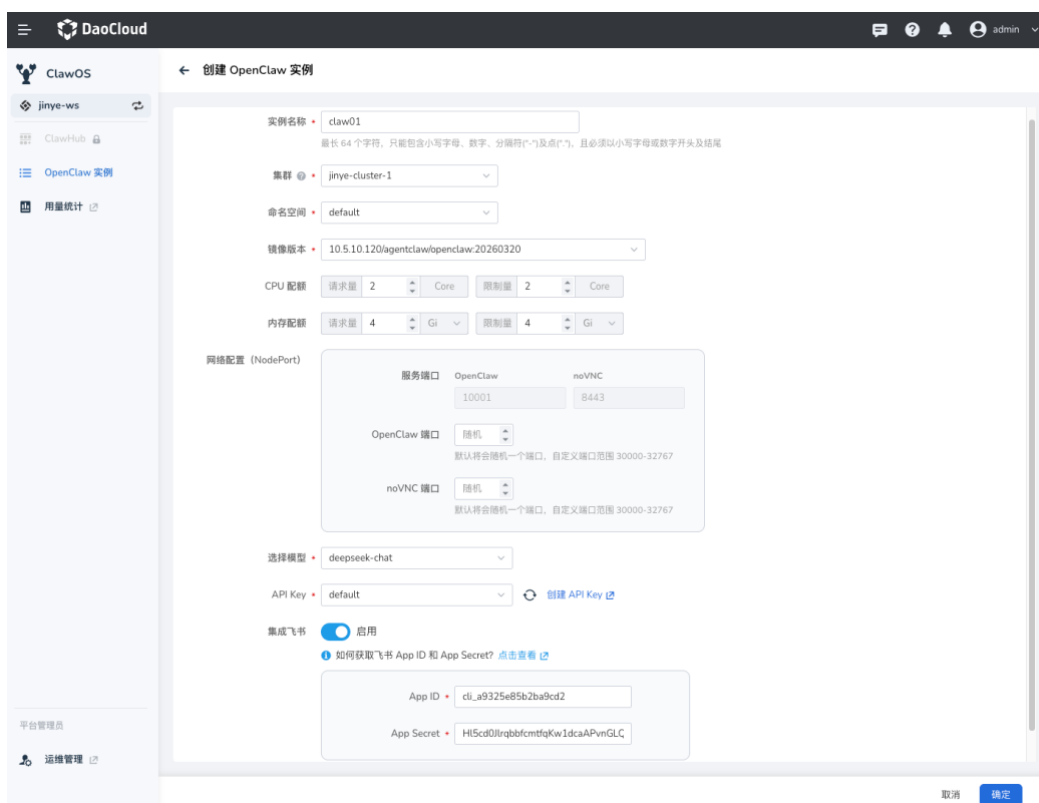


图 5 创建实例时集成飞书

有关如何获取飞书配置信息以及对接飞书的详细步骤，请参考本手册「四、工作空间」中的飞书集成部分。

3. 耐心等待实例创建完成。



图 6 等待实例创建完成

(三) 访问 OpenClaw

某个实例的状态显示为 **运行中** 后：

1. 点击右侧的 **访问工具** → **openclaw** 或 **noVNC**（浏览器里的远程桌面）。



图 7 访问工具入口

2. 打开 OpenClaw 管理页面。

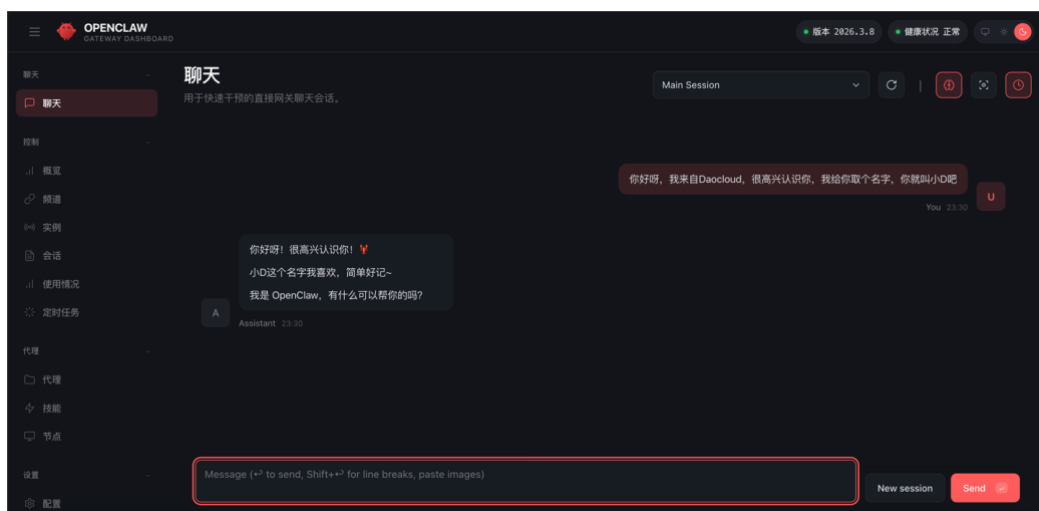


图 8 OpenClaw 管理页面

3. 打开 noVNC 管理页面。

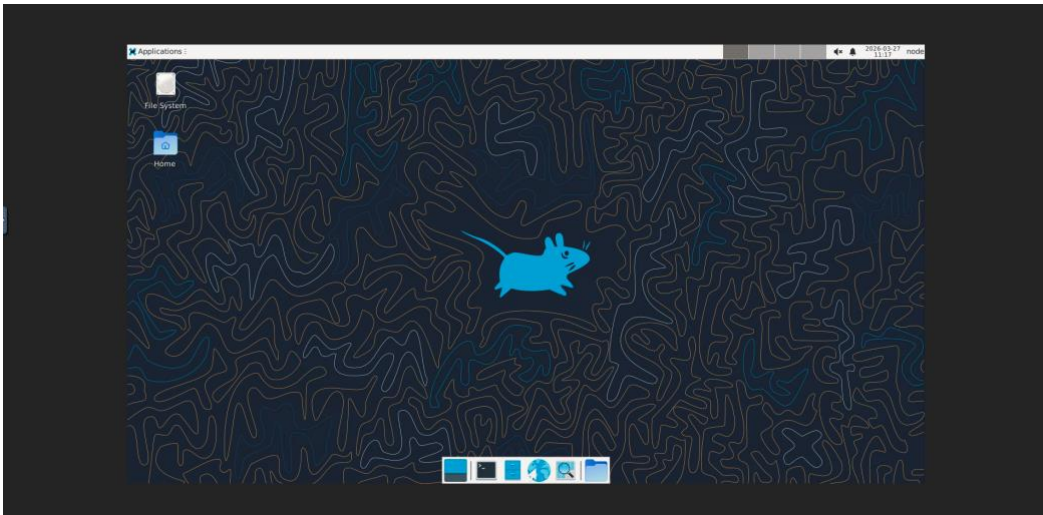


图 9 noVNC 管理页面

某些情况下，由于网络波动，在创建实例后，可能需要等待 1-2 分钟才能访问。
如果出现以下提示，可以点击 **继续访问网站**，就能在聊天窗口中使用小龙虾智能体。

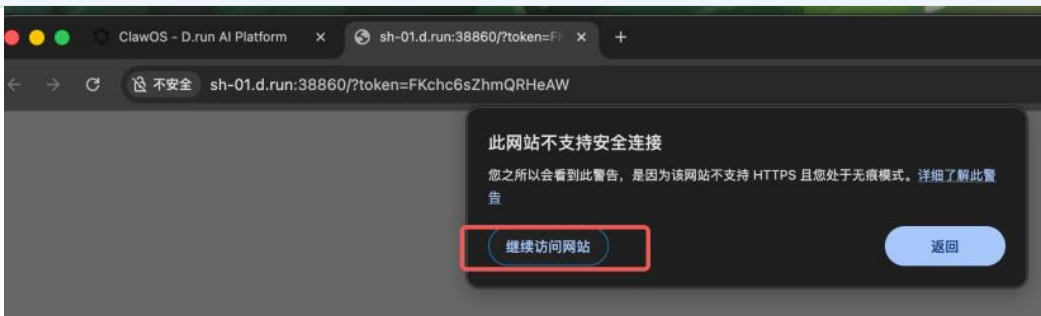
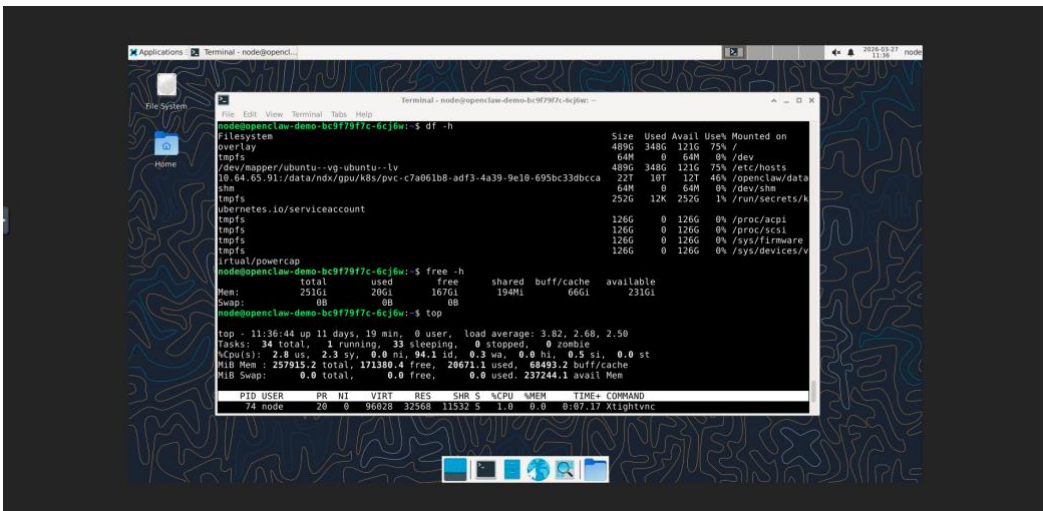


图 10 继续访问网站提示

（四）后台调试 OpenClaw

您可以通过实例列表中 noVNC 进行快捷操作。



1. 命令行操作

查看磁盘空间：

```
df -h # 查看磁盘空间
```

查看内存详细情况：

```
free -h
```

2. 常见问题

参阅本手册「八、常见问题」。

四、工作空间视角

（一）概览

工作空间视角下的「概览」页面面向工作空间管理员（租户管理员），提供当前工作空间内 **OpenClaw** 实例的整体运行、用量、成本、模型和风险情况。

通过该页面，您可以快速了解：

- 当前有多少 **Agent** 在运行
- **Token** 消耗是否异常
- 调用是否稳定
- 哪些用户或实例使用最多
- 主要使用了哪些模型

1. 模块作用

「概览」用于帮助工作空间管理员从整体层面掌握本工作空间内 **Agent** 的使用与运行情况。它不是单个实例的配置页，而是一个运营看板。主要用于：

- 查看工作空间内 **OpenClaw** 实例总体规模
- 查看活跃用户、实例数量、**API Key** 数量
- 查看 **Token** 配额、**Token** 消耗和调用趋势
- 查看错误率、调用次数等稳定性指标
- 查看 **Top** 用户或 **Top** 实例的资源消耗
- 查看不同模型的使用占比
- 辅助判断成本、风险和使用价值

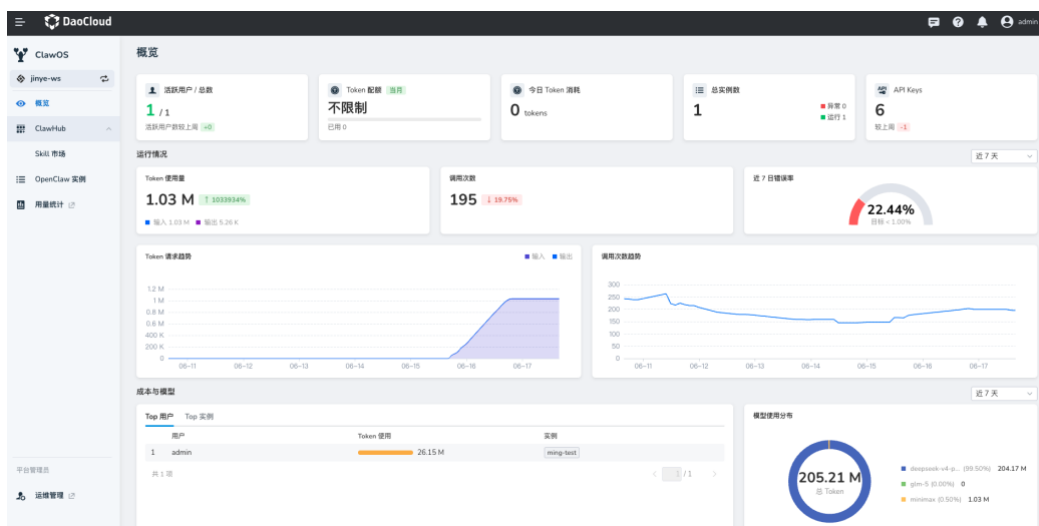


图 12 工作空间视角的概览页面

2. 核心指标说明

活跃用户/总数：表示当前工作空间中，在统计周期内实际使用过 ClawOS/OpenClaw 能力的用户数量，以及该工作空间内可使用平台的用户总数。页面中的「较上周期 +1」表示与上一统计周期相比，活跃用户数量增加了 1 人。

Token 配额：表示当前工作空间在本统计周期内可使用的 Token 总额度。例如 10,000,000 表示该工作空间当月可使用 1000 万 Token。下方进度条展示当前已使用比例。例如 已用 3,215,678 (32%) 表示已消耗 3,215,678 Token，占总配额的 32%。Token 配额用于帮助管理员控制模型调用成本，避免单个工作空间或团队无限制消耗模型资源。

当前原型中标注为「当月」，表示该指标按自然月或平台配置的月度周期统计。

总实例数：表示当前工作空间下已创建的 OpenClaw 实例总数。

- **运行中：**实例处于正常运行状态，可接收请求并执行任务
- **异常：**实例存在启动失败、运行错误、配置异常或健康检查失败等问题

该指标用于帮助管理员快速判断工作空间内 Agent 的整体部署规模和健康状态。

API Keys：表示当前工作空间内已配置或正在使用的 API Key 数量。API Key 通常用于 OpenClaw 调用大模型服务、平台网关或其他外部能力。该指标可帮助管理员了解凭证配置规模，并辅助排查调用失败、模型不可用或成本异常问题。

今日 Token 消耗：表示当前工作空间内所有 OpenClaw 实例在当天累计消耗的 Token 数。该指标适合用于快速判断当天是否存在异常高消耗。

Token 使用量：表示统计周期内的 Token 总使用量，并区分输入 Token 和输出 Token。

- **输入：**用户消息、上下文、工具返回内容、知识库检索内容等发送给模型的 Token
- **输出：**模型生成回复、计划、代码、总结等内容消耗的 Token

Token 使用量 = 输入 Token + 输出 Token。

调用次数：表示统计周期内 OpenClaw 实例或模型服务被调用的次数。调用次数可用于观察平台使用频率。

- 如果 Token 使用量增长但调用次数下降，可能说明单次任务变复杂
- 如果调用次数增长但 Token 没有明显增长，可能说明更多用户在进行轻量使用

近 7 日错误率：表示最近 7 天内调用失败次数占总调用次数的比例。错误率可反映平台运行稳定性。常见错误来源包括：

- 模型调用失败
- API Key 失效
- 网络访问失败
- Skill 执行异常
- 工具调用超时
- 实例状态异常

3. 趋势图说明

Token 请求趋势：展示统计周期内 Token 使用量的变化趋势。管理员可以通过该趋势判断是否存在突增、突降或周期性高峰。

调用次数趋势：展示统计周期内调用次数的变化情况，用于观察 Agent 使用活跃度。

- 如果调用次数持续上升，说明工作空间内 Agent 使用频率增加
- 如果调用次数突然下降，可能需要检查实例状态、消息渠道配置、API Key 或用户访问权限

4. 成本与模型

Top 用户/实例：该区域用于查看 Token 消耗最高的用户或实例。

- **用户维度：**展示不同用户的 Token 使用情况，帮助管理员了解哪些用户使用最活跃、哪些用户可能产生较高成本
- **实例维度：**展示不同 OpenClaw 实例的 Token 使用情况，帮助管理员识别高消耗实例，并进一步排查该实例是否承担了关键业务任务，或是否存在异常调用

模型使用分布：展示当前统计周期内不同模型的 Token 使用占比。该指标可帮助管理员了解模型使用结构，并辅助成本治理。例如，高成本模型占比过高时，可以考虑配置模型路由、预算限制，或推荐部分场景使用更低成本模型。

时间范围：页面中部分卡片支持按时间范围筛选，例如近 7 天、近 30 天或当月。切换时间范围后，相关指标和趋势图会同步更新。

（二）OpenClaw 实例

OpenClaw 实例是 ClawOS 当前主要管理的 Agent 类型。每个实例可以理解为一个独立运行的 AI Agent，能够调用模型、使用工具、接入消息渠道，并在企业环境中完成任务。

ClawOS 通过 **OpenClaw 实例** 模块，帮助用户创建、访问、配置和运维这些 Agent。当前主要支持 OpenClaw 实例，后续可扩展支持更多类型的企业 Agent。

1. 创建实例

在实例列表页右上角点击 **创建实例**，进入创建 OpenClaw 实例页面。

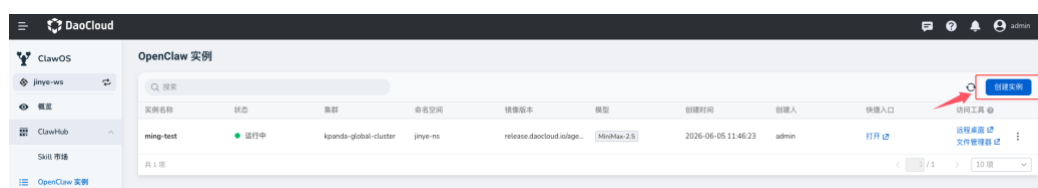


图 13 创建实例入口

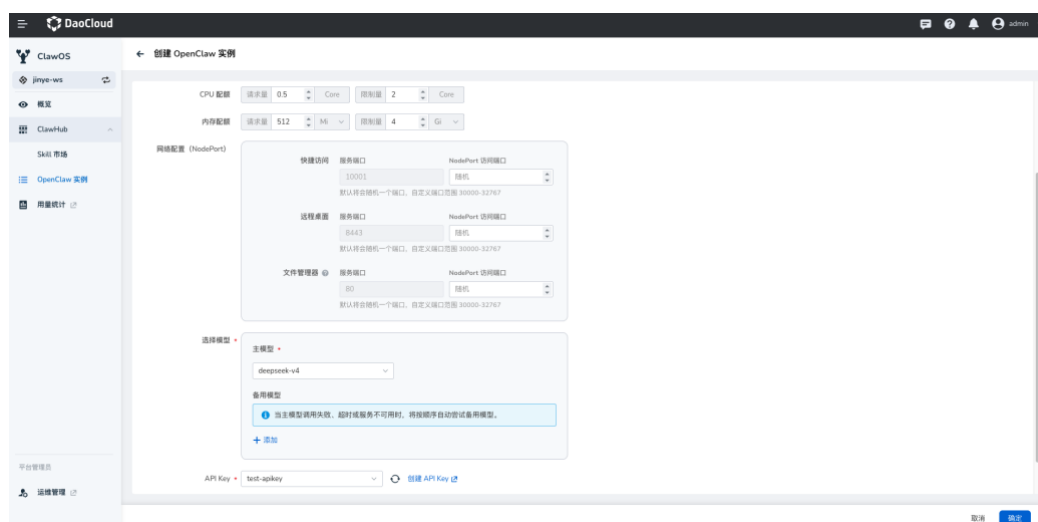


图 14 创建实例配置项

网络配置：网络配置用于生成实例的访问入口，包括快捷访问、远程桌面和文件管理器。如果不确定端口如何配置，可以使用默认端口和随机 **NodePort**，平台会自动分配可用访问端口。

模型配置：选择实例使用的主模型。主模型是 OpenClaw 执行任务时默认调用的模型。模型可以是平台统一提供的公共模型，也可以是工作空间（租户）接入的私有模型。

如果配置了备用模型，当主模型调用失败、超时或不可用时，系统会按顺序自动尝试备用模型。备用模型主要用于提高实例稳定性，适合生产任务或对可用性要求较高的场景。

API Key：选择实例调用模型时使用的 API Key。如果没有可用 API Key，可以点击 **创建 API Key** 跳转创建。

API Key 配置错误或失效时，实例可能无法正常调用模型。

配置消息渠道：OpenClaw 实例可以接入企业消息工具，让用户在飞书或 Microsoft Teams 中直接与 Agent 对话。在创建实例时，可以启用消息渠道，并选择对应渠道类型。

- **飞书：**支持扫码完成快捷配置，也支持手动创建企业自建应用，详见本手册飞书集成部分
- **Microsoft Teams：**选择 Microsoft Teams 后，需要填写 Client ID、Client Secret、Tenant ID，详情参照本手册 Microsoft Teams 配置部分。实例创建完成后，系统会在实例详情页生成 **Messaging endpoint**。请将该地址复制到 Microsoft Azure Bot 的 **Configuration > Messaging endpoint** 中。配置完成后，Teams 中的消息即可发送到对应 OpenClaw 实例，由 Agent 进行处理和回复。

2. 实例详情

点击实例名称可进入实例详情页，查看单个 OpenClaw 实例的运行情况，包括 **概览** 和 **调用记录**。

概览主要包含：

- **基础信息：**实例名称、实例 ID、状态、模型、创建时间
- **消息渠道：**当前是否启用飞书或 Teams，以及对应的 Messaging endpoint
- **Token 消耗：**今日、本月和累计 Token 使用情况
- **使用表现：**调用次数、成功率、平均响应时间
- **行为分析：**Skill 调用分布、模型使用情况等

通过概览可判断实例是否正常运行、消息渠道是否配置成功，以及该实例的使用量和稳定性是否正常。

调用记录用于了解实例在什么时间被调用、用户输入了什么内容、消耗了多少 Token、耗时多久，以及调用是否成功。

调用记录详情会展示一次对话或任务的完整执行链路，包括对话、响应、工具调用、模型输出等信息。右侧会展示对应节点的元数据，例如 conversation ID、span ID、开始时间、结束时间、Token 使用量等。该页面主要用于排查以下问题：

- 用户说了什么
- Agent 如何理解和响应
- 是否调用了工具
- 哪个环节耗时较长
- 本次调用消耗了多少 Token
- 失败发生在哪一步

调用记录适合用于运维排查、成本分析和审计追踪。对于生产环境中的 OpenClaw 实例，建议在出现异常、响应变慢、Token 消耗升高或用户反馈结果不符合预期时，优先查看调用记录。

3. 常用操作

在实例列表页，可对 OpenClaw 实例执行以下操作：

操作	说明
打开	进入 OpenClaw 实例的使用入口
远程桌面	进入实例桌面环境，适合排查运行环境或执行需要桌面的操作
文件管理器	查看和管理实例中的文件，例如上传资料、下载结果、检查任务产物
编辑	修改实例配置，例如模型、备用模型、API Key、消息渠道或资源规格。部分配置修改后可能会触发实例重启，重启期间实例会短暂不可用
重启	当实例异常、配置未生效或需要重新加载环境时使用
关机	关机后实例停止运行，无法继续访问，也不会继续占用运行资源
删除	移除对应的运行资源。删除前请确认实例中没有需要保留的文件、配置或任务结果

（三）ClawHub：Skill 市场

Skill 市场是 ClawHub 中面向工作空间成员的 Skill 浏览与安装入口。用户可以在这里查看当前可用的 Skill，并根据安装说明前往 OpenClaw 实例中安装使用。

Skill 是 OpenClaw 实例可调用的能力扩展。安装 Skill 后，OpenClaw 可以获得对应能力，例如文件处理、数据分析、系统查询、业务工具调用等。Skill 市场分为两类：

- **公开 Skill：**由超级管理员发布，所有工作空间成员都可以查看和使用
- **私有 Skill：**由当前工作空间发布并审核通过，仅当前工作空间成员可见和使用

1. 查看 Skill

进入 **ClawHub > Skill 市场** 后，可以在页面上方切换 **公开 Skill** 和 **私有 Skill**。点击 Skill 卡片中的 **详情**，可进入 Skill 详情页。详情页一般包含三个 Tab：

Tab	说明
概览	展示 Skill 的 README 或使用说明，帮助用户理解该 Skill 的用途和使用方法
文件	展示 Skill 包中的文件结构，可查看 README、配置文件、依赖文件、脚本等内容
版本	展示该 Skill 的历史版本、当前版本和安全审核结果

2. 安装 Skill

在 Skill 详情页右侧会展示安装说明。用户可以按页面提示安装 Skill，通常有两种方式：

1. 复制安装命令，进入 OpenClaw 实例后执行安装
2. 前往 OpenClaw 实例，在 **技能** 或相关入口中选择该 Skill 进行安装

安装完成后，该 OpenClaw 实例即可使用该 Skill 提供的能力。

3. 使用建议

普通用户通常只需要在 Skill 市场中完成三件事：

1. 找到需要的 Skill
2. 查看说明，确认它能解决自己的问题
3. 按安装说明前往 OpenClaw 实例安装

如果找不到需要的 Skill，可以联系工作空间管理员发布私有 Skill（参见本手册 Skill 管理部分），或由平台管理员发布公开 Skill。

（四）ClawHub：Skill 管理

Skill 管理是 ClawHub 中面向工作空间管理员的管理入口。工作空间管理员可以在这里发布、编辑、下架或删除本工作空间的 Skill；审核通过后，Skill 会在 **Skill 市场** 的 **私有 Skill** 分类中展示。

Skill 发布后需要经过审核。审核通过后，该 Skill 会出现在当前工作空间的 **私有 Skill** 市场中，仅本工作空间成员可见和使用。

1. 查看 Skill 列表

进入 **ClawHub > Skill 管理** 后，可以查看当前工作空间内已创建或已发布的 Skill。常见状态包括：

状态	说明
----	----

已发布	Skill 已审核通过，工作空间成员可以在私有 Skill 市场中看到并使用
审核中	Skill 已提交发布申请，正在等待审核
审核失败	Skill 未通过审核，需要修改后重新提交
已下架	Skill 暂停对成员可见，不能继续安装使用

2. 发布 Skill

1. 点击右上角 **发布 Skill**，进入发布页面。

2. 填写以下信息：

- **Skill 名称**：用于在 Skill 市场中展示，建议使用简洁、能表达用途的名称
- **版本号**：本次发布的版本标识
- **Skill 文件**：上传 Skill 压缩包

1. 上传完成后，点击 **申请并发布**，系统会提交审核流程。

审核通过后，该 Skill 会出现在当前工作空间的 **私有 Skill** 市场中。

Skill 文件要求：上传的 Skill 文件通常为压缩包格式。文件中应包含 Skill 运行所需的说明文件、配置文件、脚本和依赖声明。建议至少包含：

- Skill 说明文件
- manifest 配置文件
- 入口脚本或工具定义
- 依赖说明
- 示例或 README

文件内容应避免包含敏感信息，例如明文密钥、账号密码、内部 Token 等。

3. 编辑 Skill

对于已创建的 Skill，管理员可以点击 **编辑** 修改 Skill 信息。编辑时不可修改原文件名称，只能修改内容后重新提交。如果修改了 Skill 文件或版本内容，通常需要重新提交审核；审核通过后，新版本才会对工作空间成员可用。

4. 下架 Skill

如果某个 Skill 暂时不希望继续被成员安装，可以执行 **下架**。下架后，该 Skill 不再出现在私有 Skill 市场中，新用户无法继续安装。已经安装到 OpenClaw 实例中的 Skill 是否继续可用，取决于平台策略和实例侧处理方式。

5. 删除 Skill

删除用于移除 Skill 记录。删除前请确认该 Skill 不再需要维护，也没有成员依赖它进行日常工作。如果 Skill 仍在被多个 OpenClaw 实例使用，建议先下架并通知相关用户，再执行删除。

6. 查看发布后的 Skill

Skill 审核通过后，管理员和工作空间成员可以在 **Skill 市场 > 私有 Skill** 中看到该 Skill。进入详情页后，可以查看概览、文件、版本和安装说明。成员可按安装说明前往 OpenClaw 实例安装使用。

（五）飞书集成

本部分介绍如何将 OpenClaw 与飞书进行集成，实现通过飞书与 OpenClaw 对话。

1. 快捷配置（推荐）

点击 **快速生成**，打开飞书 APP 扫码，按照提示步骤完成授权即可。

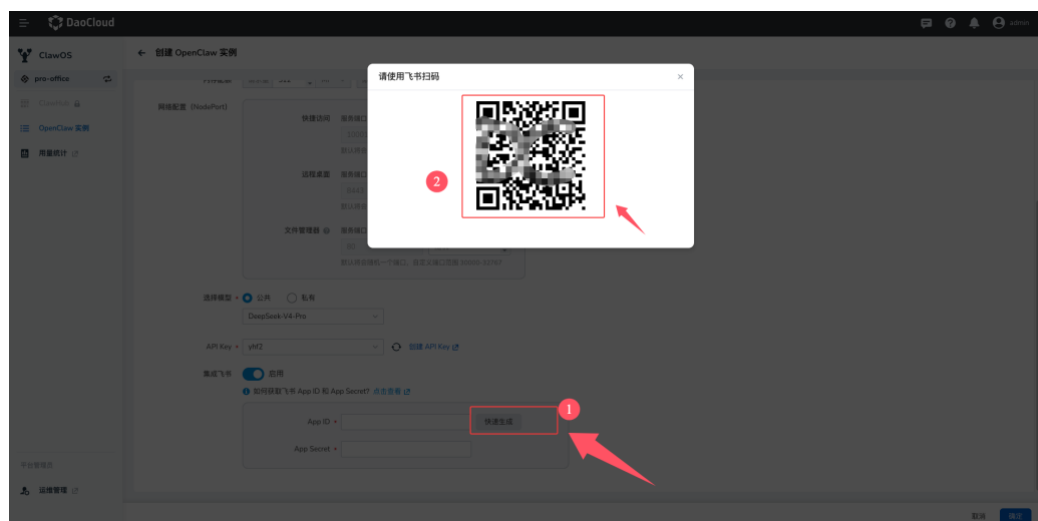


图 15 飞书快捷配置

2. 手动配置

创建企业自建应用

1. 登录飞书开放平台（<https://open.feishu.cn/app>）。
2. 点击 **创建企业自建应用** 按钮。



图 16 创建企业自建应用

配置应用信息：配置应用名称、描述及图标，点击 **创建** 按钮完成创建。

图 17 配置应用信息

添加机器人能力

1. 在左侧目录树选择 **应用能力** → **添加应用能力**
2. 选择 **按能力添加** 页签
3. 点击 **机器人** 能力卡片的 **添加** 按钮

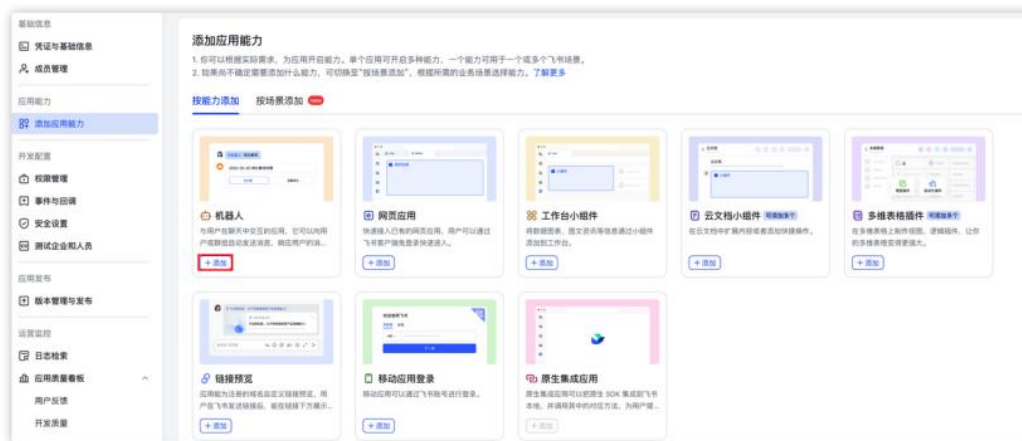


图 18 添加机器人能力

3. 配置权限

导入权限配置

1. 在左侧目录树选择 **开发配置** → **权限管理**
2. 点击 **批量导入/导出权限** 按钮。
3. 在 **导入** 页签中，粘贴下方权限配置代码。
4. 点击 **下一步**，**确认新增权限** 按钮。
5. 在弹窗中确认权限无误后，点击 **申请开通** 按钮。



图 19 批量导入/导出权限

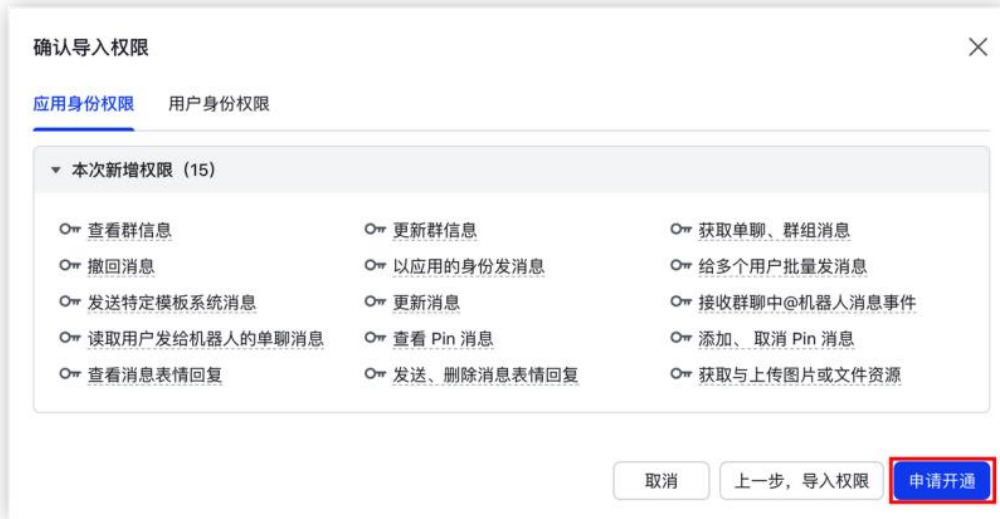


图 20 权限配置确认

权限配置代码

```
{
  "scopes": {
    "tenant": [
      "contact:contact.base:readonly",
      "docx:document:readonly",
      "im:chat:read",
      "im:chat:update",
      "im:message.group_at_msg:readonly",
      "im:message.p2p_msg:readonly",
      "im:message.pins:read",
      "im:message.pins:write_only",
      "im:message.reactions:read",
      "im:message.reactions:write_only",
      "im:message:readonly",
      "im:message:recall",
      "im:message:send_as_bot",
      "im:message:send_multi_users",
      "im:message:send_sys_msg",
      "im:message:update",
      "im:resource",
      "application:application:self_manage",
      "cardkit:card:write",
      "cardkit:card:read"
    ],
    "user": [
      "contact:user.employee_id:readonly",
      "offline_access",
      "base:app:copy",
      "base:field:create",
      "base:field:delete",
      "base:field:read",
      "base:field:update",
      "base:record:create",
      "base:record:delete",
      "base:record:retrieve",
      "base:record:update",
      "base:table:create",
      "base:table:delete",
      "base:table:read",
      "base:table:update",
      "base:view:read",
      "base:view:write_only",
    ]
  }
}
```

```

"base:app:create",
"base:app:update",
"base:app:read",
"sheets:spreadsheet.meta:read",
"sheets:spreadsheet:read",
"sheets:spreadsheet:create",
"sheets:spreadsheet:write_only",
"docs:document:export",
"docs:document.media:upload",
"board:whiteboard:node:create",
"board:whiteboard:node:read",
"calendar:calendar:read",
"calendar:calendar.event:create",
"calendar:calendar.event:delete",
"calendar:calendar.event:read",
"calendar:calendar.event:reply",
"calendar:calendar.event:update",
"calendar:calendar.free_busy:read",
"contact:contact.base:readonly",
"contact:user.base:readonly",
"contact:user:search",
"docs:document.comment:create",
"docs:document.comment:read",
"docs:document.comment:update",
"docs:document.media:download",
"docs:document:copy",
"docx:document:create",
"docx:document:readonly",
"docx:document:write_only",
"drive:drive.metadata:readonly",
"drive:file:download",
"drive:file:upload",
"im:chat.members:read",
"im:chat:read",
"im:message",
"im:message.group_msg:get_as_user",
"im:message.p2p_msg:get_as_user",
"im:message:readonly",
"search:docs:read",
"search:message",
"space:document:delete",
"space:document:move",
"space:document:retrieve",
"task:comment:read",
"task:comment:write",
"task:task:read",
"task:task:write",
"task:task:writeonly",
"task:tasklist:read",
"task:tasklist:write",
"wiki:node:copy",
"wiki:node:create",
"wiki:node:move",
"wiki:node:read",
"wiki:node:retrieve",
"wiki:space:read",
"wiki:space:retrieve",
"wiki:space:write_only"
]
}
}

```

相关权限的具体含义可查看飞书 API 权限列表：

<https://open.larkoffice.com/document/server-docs/application-scope/scope-list>

4. 发布应用

创建版本并发布

1. 点击顶部的 **创建版本** 按钮。

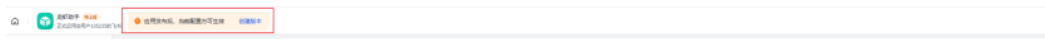


图 21 创建版本

2. 按需配置应用版本号、默认能力及更新说明等信息。



图 22 配置版本

3. 点击页面底部的 **保存** 按钮创建版本。



图 23 保存版本

4. 点击页面右上角的 **确认发布** 按钮完成应用发布。

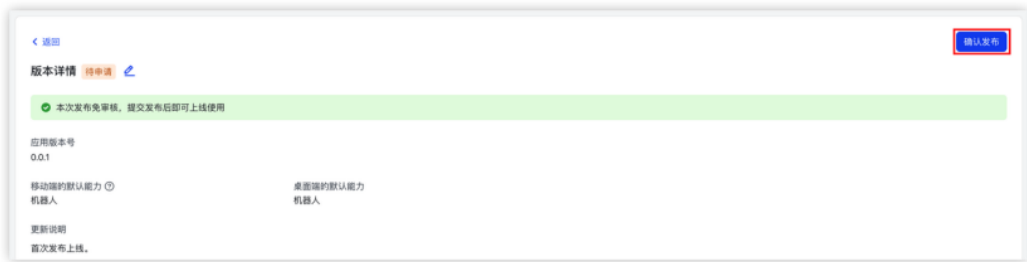


图 24 发布应用

获取配置信息

1. 在左侧目录树选择 **基础信息** → **凭证与基础信息**
2. 在 **应用凭证** 模块中，获取并记录 **App ID** 与 **App Secret** 信息。



图 25 获取应用凭证

5. 配置事件订阅与回调长连接

对于未配置长连接的机器人应用，需要在获取 **App ID** 与 **App Secret** 后，启动 OpenClaw 实例，才能保存事件与回调长连接配置。请参考本手册「三、快速入门」启动 OpenClaw 实例。

配置事件订阅

1. 点击左侧事件与回调，配置 **事件订阅** 方式为 **长连接**。



图 26 配置事件订阅

2. 添加接收消息事件（可自定义其他事件）。



图 27 添加接收消息事件

配置回调

1. 点击 **回调配置**
2. 配置方式选择 **长连接**。



图 28 回调配置

3. 添加回调。



图 29 添加回调

发布版本：发布版本并等待审核通过。



图 30 发布版本并等待审核

6. 开始对话

在飞书中找到您的机器人，开始愉快地对话吧！

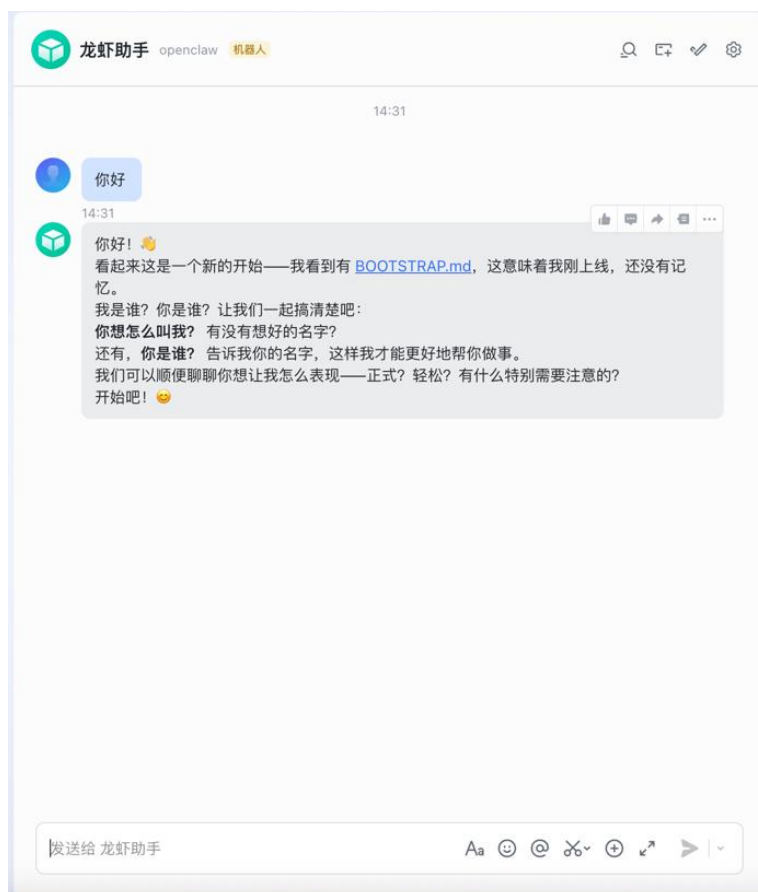


图 31 在飞书中与小龙虾对话

（六）Microsoft Teams 配置

本部分介绍如何注册 Microsoft Entra 单租户应用，创建 Azure Bot，并将 OpenClaw 接入 Microsoft Teams。

1. 前置条件

开始前请确认：

- 已拥有客户 Microsoft Entra 租户和 Azure 订阅
- 具备创建应用注册和 Azure Bot 的权限
- Messaging endpoint 是公网可访问的 HTTPS 地址

2. 字段对应关系

OpenClaw 字段	Microsoft 中的名称	填写内容	注意事项
Client ID	Application (client) ID / Microsoft App ID	Entra 应用注册概览页 中的 GUID	不要填写 Object ID

Client Secret	Client secret → Value	创建客户端密码后显示的 Value	不要填写 Secret ID; Value 只显示一次
Tenant ID	Directory (tenant) ID / App Tenant ID	Entra 租户 GUID	必须与应用注册所在租户一致
Messaging endpoint	Azure Bot → Configuration → Messaging endpoint	OpenClaw 提供的完整 URL	必须使用完整 HTTPS 地址

3. 获取 Client ID、Tenant ID 及 Client Secret

注册应用

- 访问 Microsoft Entra 管理中心 (<https://entra.microsoft.com/>)，选择 **Entra ID** → **App registrations** → **New registration**，填写：
 - Name:** 例如 [OpenClaw-Teams-Prod](#)
 - Supported account types:** 选择 [Accounts in this organizational directory only](#)
 - Redirect URI:** 本文流程通常不需要填写
- 注册完成后，在 **Overview** 页面复制：
 - Application (client) ID:** 作为 OpenClaw 的 Client ID
 - Directory (tenant) ID:** 作为 OpenClaw 的 Tenant ID

创建 Client Secret

- 在应用注册中选择 **Certificates & secrets** → **Client secrets** → **New client secret**
- 填写描述和有效期，选择 **Add**。
- 创建后立即复制 **Value**，该值就是 OpenClaw 的 Client Secret。

Microsoft 不会再次显示 Secret Value。如果忘记保存，只能重新创建一个 Client Secret。

4. 创建并配置 Azure Bot

打开 Azure 门户 (<https://portal.azure.com/>)，选择 **Create a resource** → 搜索 **bot** → **Azure Bot** → **Create**。

配置应用身份：在 **Microsoft App ID** 或身份配置区域：

- 选择 **Single Tenant**。
- 选择 **Use existing app registration**。
- 输入前面创建的 **Application (client) ID**。
- 如果页面要求输入 Tenant ID，填写同一个 **Directory (tenant) ID**。

完成配置后选择 **Review + create** → **Create**。

配置 Messaging endpoint

- 1. 进入 Azure Bot 资源，点选 **Settings** → **Configuration**
- 2. 将 OpenClaw 提供的完整 endpoint 填入 **Messaging endpoint**，然后选择 **Apply**。

endpoint 地址必须：

- 使用 HTTPS
- 可以从公网访问
- 保留 OpenClaw 提供的完整路径

启用 Microsoft Teams channel

- 1. 进入 **Channels** → **Microsoft Teams**
- 2. 同意相关条款。如果页面显示 **Cloud environment**，选择与客户 Teams 环境匹配的选项，然后选择 **Apply**。

5. 在 OpenClaw 中填写信息

OpenClaw 输入框	填写内容
Client ID	Entra 应用的 Application (client) ID
Client Secret	Entra Client secret 的 Value
Tenant ID	Entra 应用的 Directory (tenant) ID

保存后，确认各字段没有多余空格或换行。

6. 将 Bot 添加到 Teams

启用 Teams channel 后，还需要将 Bot 添加到 Teams。

测试环境

- 1. 在 Azure Bot，选择 **Channels** → **Microsoft Teams**
- 2. 页面获取 Teams 测试链接，打开后选择 Teams 客户端或 Teams Web，将 Bot 添加到 Teams。

生产环境：对于生产环境，建议创建 Teams App，将 Bot ID 设置为 Entra 应用的 **Application (client) ID**，然后上传或发布到客户 Teams 租户。仅通过 Bot GUID 添加适合测试，不建议用于生产环境。

7. 验证清单

检查项	通过标准
应用注册	应用类型为 Single Tenant
Client ID	OpenClaw、Entra 和 Azure Bot 中的应用 ID 一致
Client Secret	使用 Secret Value，且未过期
Tenant ID	OpenClaw 与 Entra 的 Tenant ID 一致
Messaging endpoint	配置为 OpenClaw 提供的完整 HTTPS 地址
Teams channel	Microsoft Teams channel 已启用
Teams 安装	Bot 已添加到 Teams
消息测试	Teams 消息能够触发 OpenClaw 回复

8. 密钥轮换

Client Secret 到期前：

1. 创建新的 Client Secret。
2. 将新的 Value 更新到 OpenClaw。
3. 发送 Teams 测试消息确认生效。
4. 再删除旧 Secret。

9. 常见问题

Teams 中找不到 Bot： 检查 Teams channel 是否启用，以及 Bot 是否已经通过测试链接或 Teams App 添加到 Teams。

发送消息后没有回复： 检查以下内容：

- Messaging endpoint 是否完整
- endpoint 是否可以公网访问
- Client ID、Client Secret 和 Tenant ID 是否填写正确
- Cloud environment 是否选择正确

出现 401 或 403： 重点检查：

- 是否误填了 Secret ID
- Client Secret 是否已过期
- Tenant ID 是否属于当前应用注册所在的租户

10. 参考文档

- Register a bot with Azure: <https://learn.microsoft.com/en-us/azure/bot-service/bot-service-quickstart-registration>
- Connect a Bot Framework bot to Microsoft Teams: <https://learn.microsoft.com/en-us/azure/bot-service/channel-connect-teams>

五、管理员视角

（一）概览

平台管理员概览用于从 **全局维度** 监控 ClawOS 平台的整体调用情况、资源消耗、实例运行状态和用户/API Key 使用分布，帮助管理员快速判断平台是否处于健康、稳定、可控的运行状态。

与工作空间视角下的概览不同，本页面面向 **平台管理员**，展示跨工作空间的汇总数据，用于平台级运营与治理。

1. 模块作用

「概览」是 ClawOS 管理员视角下的运营看板，不是单个实例或单个工作空间的配置页。主要用于：

- 查看平台整体调用量与 Token 消耗
- 监控实例运行规模与健康状态
- 分析资源消耗在工作空间、用户、模型之间的分布
- 定位高消耗用户、API Key 和重点调用来源
- 识别流量高峰、异常突增和成本风险
- 辅助配额管理、成本核算、容量规划和问题排查

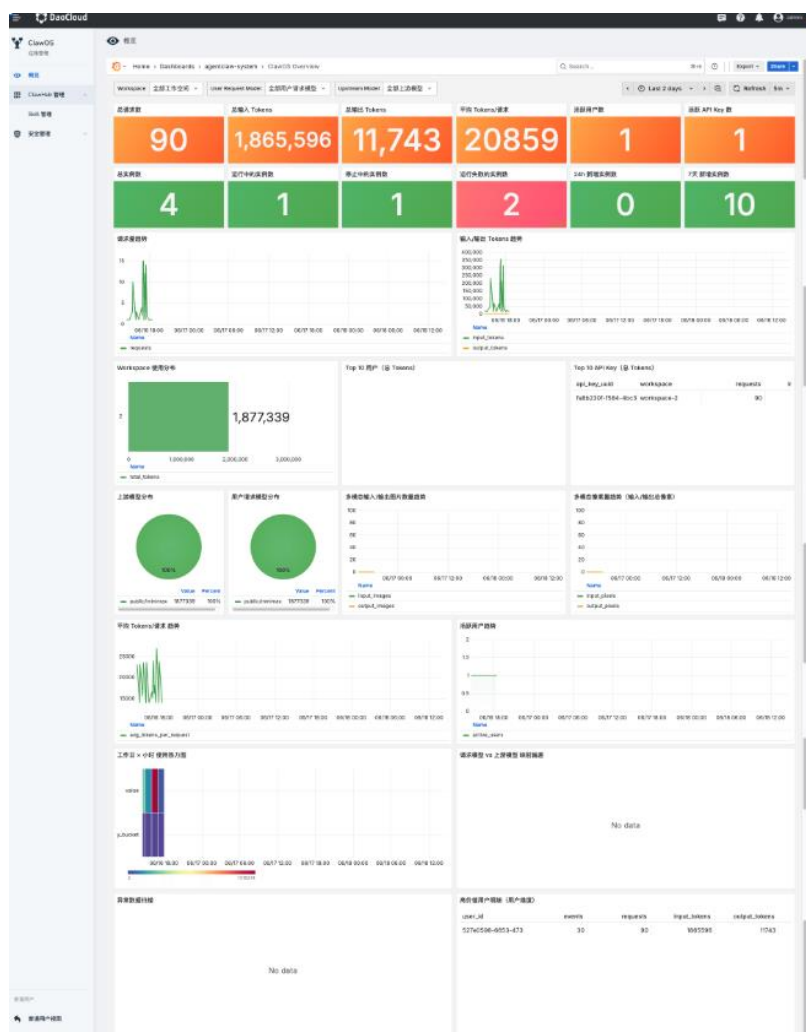


图 32 平台管理员视角的概览页面

2. 筛选与时间范围

管理员可以按以下维度筛选，并查看指定时间范围内的平台运行数据：

- 工作空间
- 用户请求模型：用户或实例在请求时选择的模型
- 上游模型：平台实际转发调用的底层模型

页面中部分卡片支持按时间范围筛选，例如近 7 天、近 30 天或当月。切换时间范围后，相关指标和趋势图会同步更新。

3. 核心指标

页面顶部集中展示以下关键指标：

类别	指标
调用与 Token	平台总请求数、输入 Token、输出 Token、平均每次请求 Token 数

用户与凭证	活跃用户数、活跃 API Key 数
实例状态	实例总数、运行中实例数、停止中实例数、运行失败实例数

通过这些指标，管理员可以快速判断平台当前负载、资源消耗规模和实例整体健康情况。

4. 趋势图说明

趋势图用于观察核心指标随时间的变化，便于识别流量高峰、异常突增或资源消耗波动。

常见趋势包括：

- 平台请求量趋势
- Token 消耗趋势
- 平均 Token / 请求趋势
- 活跃用户趋势

如果某段时间请求量或 Token 消耗突然升高，可能需要进一步查看分布分析和排名明细，排查是否存在异常调用或成本风险。

5. 分布分析

分布分析区域展示不同维度的资源使用占比，帮助管理员了解资源消耗结构。

工作空间分布：展示各工作空间的总 Token 消耗占比，用于分析哪些工作空间占用了主要平台资源。

上游模型分布：展示平台实际调用的底层模型使用占比，用于了解当前平台主要的模型消耗结构。

用户请求模型分布：展示用户或实例在请求时选择的模型使用占比，用于了解业务侧的主要模型偏好。

6. 排名与明细

排名与明细区域提供 Top 排行和高价值用户明细，帮助管理员定位主要调用来源。

区域	说明
Top 用户	按请求数或 Token 消耗排序，定位最活跃或高消耗用户
Top API Key	定位高消耗或高频使用的 API Key
高价值用户明细	展示重点用户的请求数、输入 Token、输出 Token 等明细

结合这些数据，管理员可以进一步追踪资源使用主体，为配额管理、成本核算、异常排查和用户运营提供依据。

7. 高级分析

除常规调用与 Token 指标外，平台管理员还可以查看以下高级分析信息：

- **多模态统计：**输入 / 输出图片数量、像素量趋势
- **使用热力图：**工作日小时使用热力图，观察调用高峰时段
- **模型映射偏差：**请求模型与上游模型映射偏差，发现模型路由异常
- **异常数据扫描：**识别非预期调用行为或潜在数据问题

8. 核心价值

平台管理员概览重点帮助回答以下问题：

- 平台整体用量是多少？
- 资源消耗主要集中在哪些工作空间、用户或模型？
- 哪些用户或 API Key 是主要调用来源？
- 实例运行是否健康？
- 是否存在异常请求、成本突增或模型使用偏差？

如需查看单个工作空间内的运行情况，请参见本手册「四、工作空间」中的概览部分。

（二）Skill 管理

平台管理员视角下的 Skill 管理是 ClawHub 的平台级供给与治理后台，用于发布、审核、上架、下架和维护 公开 Skill。

平台管理员不消费 Skill，也不需要进入 Skill 市场安装 Skill。因此管理员后台不提供 Skill 市场入口，只提供 Skill 管理能力。由平台管理员发布并上架的 Skill，会进入所有工作空间 ClawHub Skill 市场的 公开 Skill 列表，所有工作空间成员都可以在各自工作空间中安装并使用。

工作空间管理员发布私有 Skill 的操作，请参见本手册「四、工作空间」中的 Skill 管理部分。

1. 角色与边界

平台管理员：平台管理员负责维护平台公共 Skill 供给，包括：

- 发布新的 Skill
- 上传 Skill 文件包
- 查看平台已发布 Skill 列表
- 编辑 Skill 基本信息和版本
- 发起或处理 Skill 审核
- 查看自动安全扫描结果
- 将 Skill 上架到公开市场

- 将 Skill 从公开市场下架
- 删除未上架或不再需要的 Skill

工作空间成员：工作空间成员不在管理员后台出现。成员只能在工作空间视角的 ClawHub Skill 市场 中发现公开 Skill，并在工作空间中安装和使用。

工作空间管理员：工作空间管理员可以在工作空间内安装、启用、停用、配置或卸载 Skill，但不能发布平台级公开 Skill，也不能绕过平台审核流程。

2. 发布 Skill

1. 平台管理员点击 **发布 Skill**，进入发布页面。

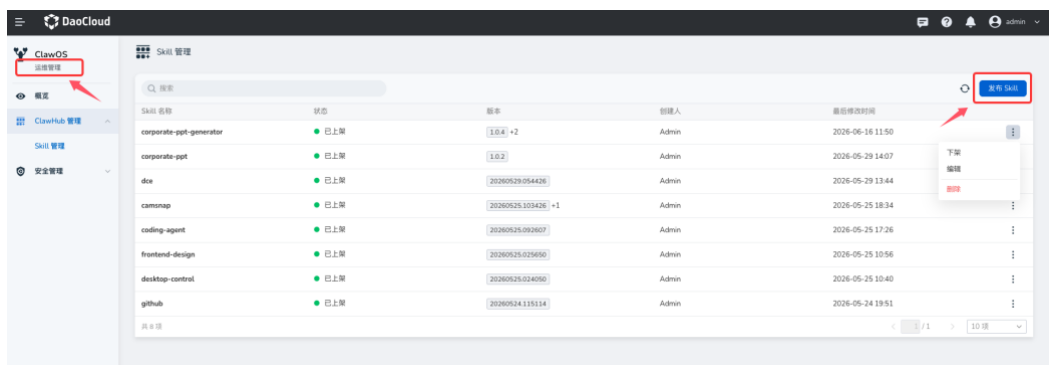


图 33 发布 Skill 入口

2. 填写 Skill 信息并上传 Skill 文件包，提交发布。

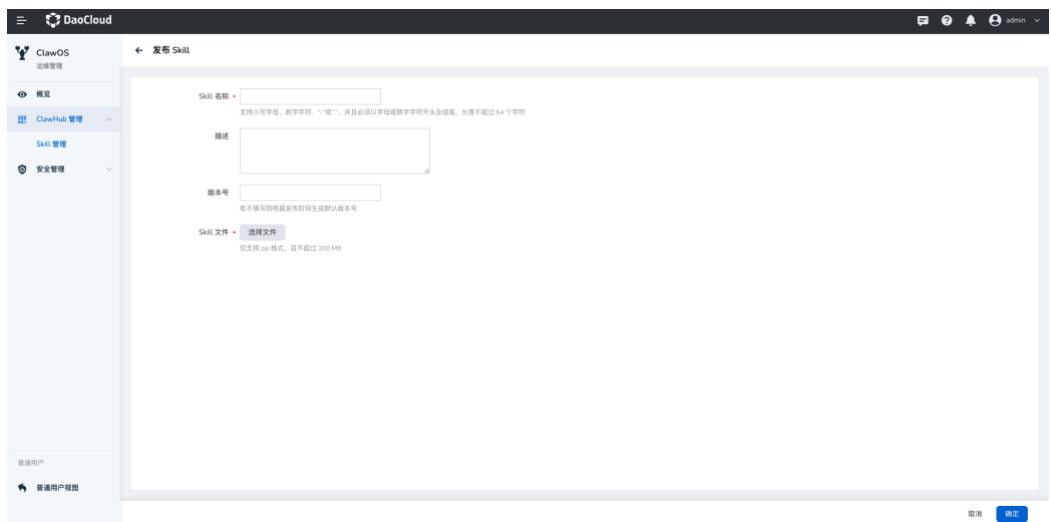


图 34 发布 Skill 表单

提交发布后，系统创建一条 Skill 记录，并进入审核流程。Skill 不会立即 出现在工作空间 Skill 市场中。

发布后状态流转

1. 平台管理员填写信息并上传 Skill 文件

2. 系统创建 Skill 草稿
3. 系统自动发起安全扫描
4. 安全扫描完成后进入待审核状态
5. 平台管理员或审核员执行审核
6. 审核通过后，Skill 进入未上架状态
7. 平台管理员执行上架
8. Skill 出现在所有工作空间的 ClawHub Skill 市场 公开 Skill 中

3. 上架与下架

上架：平台管理员可以将审核通过的 Skill 上架。上架后：

- Skill 状态变为 已上架
- Skill 进入所有工作空间的 ClawHub Skill 市场
- Skill 在市场中归类为 公开 Skill
- 所有工作空间成员都可以查看、安装并使用该 Skill
- 已安装该 Skill 的工作空间可以继续在工作空间内管理启用、停用和配置

下架：平台管理员可以将已上架 Skill 下架。下架后：

- Skill 从所有工作空间的 ClawHub Skill 市场 公开 Skill 列表中移除
- 新的工作空间不能再安装该 Skill
- 已安装该 Skill 的工作空间是否继续可用，由平台策略决定
- 下架不会立即卸载已安装 Skill；已安装工作空间可以继续使用当前版本
- 工作空间不可再安装新实例，也不可升级到下架版本

（三）网络策略管理

ClawOS 网络策略管理用于将 Kubernetes 的网络隔离能力产品化为可复用的策略模板。

平台管理员可以预先定义网络访问边界，用户在创建 OpenClaw 实例时选择或默认应用对应策略；ClawOS 会在实例创建后自动下发真实的网络策略，限制实例的入站或出站访问范围。

该能力适用于企业环境中对 Agent 运行边界有安全要求的场景，例如：

- 限制 OpenClaw 实例访问敏感内网网段
- 控制公网访问范围
- 允许访问指定数据库/API 服务
- 隔离不同用户实例

1. 适用场景

- 限制 OpenClaw 实例访问企业内网网段

- 为不同安全等级的 OpenClaw 实例提供不同网络隔离策略
- 将平台安全基线固化为默认策略，在实例创建时自动生效
- 为有特殊访问诉求的实例提供可选网络策略模板

2. 角色与权限说明

平台管理员可以进行以下操作：

- 创建/编辑/启用/停用网络策略模板
- 设置模板是否为平台默认策略

工作空间成员可以进行以下操作：

- 创建 OpenClaw 实例时查看可用的网络策略模板
- 选择平台允许使用的可选网络策略
- 查看当前实例已启用的网络策略
- 根据权限跳转到容器管理中创建自定义 NetworkPolicy

普通用户不能直接编辑平台管理员创建的网络策略模板。

3. 网络策略类型说明

平台默认策略：平台默认策略由平台管理员配置，用于定义 OpenClaw 实例运行时必须遵守的安全基线。

- 默认策略会在创建 OpenClaw 实例时 **自动启用**，用户不能取消
- 通常用于保障平台整体安全

用户可选策略：用户可选策略由平台管理员创建，用户可以在创建 OpenClaw 实例时按需选择。

- 用户只能选择是否启用该策略，**不能修改** 策略内容

自定义网络策略：当平台预置策略无法满足特殊场景时，用户可以根据权限前往容器管理中的原生 **NetworkPolicy** 页面创建自定义网络策略（拥有 NS 编辑权限的用户即可创建）。

4. 创建网络策略模板

1. 点击页面右上角的 **创建网络策略模板**，进入创建页面
2. 按页面提示填写模板信息并配置访问规则
3. 配置完成后点击 **保存**

策略类型选择——平台默认策略

- 在创建 OpenClaw 实例时自动生效，用户不能取消
- 适合放置平台必须执行的安全规则，例如：

- 禁止访问敏感内网网段
- 禁止访问集群核心服务
- 保留必要的 DNS 解析能力

策略类型选择——用户可选策略

- 展示在 OpenClaw 实例创建页面，用户可按场景选择是否启用
- 适合放置按场景启用的规则，例如：
 - 允许访问模型服务
 - 允许访问企业知识库
 - 允许访问 Teams/飞书消息渠道
 - 允许访问指定业务系统

配置访问规则：策略配置分为 **入流量策略** 和 **出流量策略**。如果源 Pod 想要成功连接到目标 Pod，源 Pod 的出流量策略和目标 Pod 的入流量策略都需要允许连接；任何一方不允许，都会导致连接失败。

点击 **加号** 开始配置策略，支持配置多条策略。多条网络策略的效果相互叠加，只有同时满足所有网络策略，才能成功建立连接。

配置项	说明	用途
podSelector	选择指定 Pod	允许或限制访问带有指定标签的服务
namespaceSelector	选择指定命名空间	允许或限制访问某个命名空间下的服务

保存后，该模板会出现在网络策略模板列表中。若创建时勾选 **启用** 且模板为 **平台默认策略**，已创建及未来创建的 OpenClaw 实例均会立即自动启用，用户不能取消。若不确定配置效果，可在创建时取消勾选「启用」。

5. 创建 OpenClaw 实例时使用网络策略

进入路径：**OpenClaw 实例** → **创建实例** → **网络策略**。相关操作请参见本手册「四、工作空间」中的 OpenClaw 实例部分。

平台默认策略：默认启用，不可取消。这是平台管理员设置的安全基线，所有实例必须遵守。

可选策略：用户可以根据实例用途按需选择。

自定义策略：若平台默认策略和可选策略均不能满足 OpenClaw 实例需求，可通过快捷链接前往创建原生 **NetworkPolicy**。

创建后的原生 NetworkPolicy 不会 出现在 OpenClaw 实例的网络策略下拉框中。

6. 注意事项

- 入流量策略 控制「谁能访问当前实例」
- 出流量策略 控制「当前实例能访问谁」
- podSelector 用于选择带有指定标签的 Pod
- namespaceSelector 用于选择带有指定标签的命名空间
- 平台默认策略用户不能取消
- 可选策略用户可以按需选择
- 修改网络策略可能影响模型调用、知识库访问或消息渠道收发，请谨慎操作

六、参考文档

(一) Gateway API 实例访问配置

ClawOS 实例的 UI（OpenClaw web / noVNC / filebrowser）与 MS Teams 回调可以通过 Gateway API 统一入口暴露。管理员只需在集群里准备一个打了 label 的 Gateway，ClawOS 会在创建实例时自动挑中它并生成 HTTPRoute；挑不到可用 Gateway 时，实例自动回退 NodePort 直连，不影响创建。

最小配置就两步：给 Gateway 打上 agentclaw.io/gateway: "true"，再用 agentclaw.io/base-url 注解告诉 ClawOS 用户从外面访问的地址是什么。下面逐项说明。

1. 前置条件

- 集群已安装 Gateway API CRD（gateway.networking.k8s.io/v1）及对应的 GatewayClass 控制器（如 kgateway、Envoy Gateway、Istio）。
- 控制器必须支持 WebSocket 透传：OpenClaw 与 noVNC 均走 WS。ClawOS 生成的实例 Service 已在对应端口标注 appProtocol: kubernetes.io/ws，部分实现（kgateway）依赖它来放行 WebSocket。

2. Gateway 需要满足的条件

下列条件缺一不可，不满足的 Gateway 会被跳过（不会报错，实例直接回退 NodePort）：

条件	说明
label agentclaw.io/gateway: "true"	值必须是字符串 "true"
至少一个 listener 的 protocol 是 HTTP 或 HTTPS	TCP / TLS / UDP listener 承载不了 HTTPRoute
该 listener 的 allowedRoutes.namespaces.from 为 All	必须显式写

该 listener 的 <code>allowedRoutes.kinds</code> 未指定，或包含 <code>HTTPRoute</code>	只声明了 <code>GRPCRoute</code> 的 listener 不可用
能确定对外访问地址	即"配了合法的 <code>agentclaw.io/base-url</code> 注解"或" <code>status.addresses</code> 非空"

ClawOS 不检查 Gateway 的 [Accepted / Programmed condition](#)。控制器还没把 Gateway 编程完成时，它依然可能被选中——此时 `HTTPRoute` 能创建成功，但要等控制器就绪后才真正可访问。

示例（HTTPS listener + 显式对外域名）：

```
kind: Gateway
apiVersion: gateway.networking.k8s.io/v1
metadata:
  name: clawos-gateway
  namespace: public
  labels:
    agentclaw.io/gateway: "true"
  annotations:
    agentclaw.io/base-url: https://agentclaw.example.com
spec:
  gatewayClassName: kgateway
  listeners:
    - name: https
      port: 443
      protocol: HTTPS
      tls:
        mode: Terminate
        certificateRefs:
          - name: agentclaw-tls
      allowedRoutes:
        namespaces:
          from: All
```

3. 对外访问地址（`agentclaw.io/base-url`）

这个地址会被直接拼上 `/agentclaw/<instanceID>/...` 前缀，作为下发给用户的访问链接，所以它必须是用户浏览器实际访问的地址，而不是集群内部地址。

方式一：注解显式指定（推荐）：注解一旦存在就以它为准，不再看 `status.addresses`。格式必须是 `scheme://host[:port]`，写得不合法会导致该 Gateway 整体不可用（回退 NodePort）。

方式二：不写注解，自动推导：取 `status.addresses` 中第一个非空 `value`，拼成 `<listener 协议小写>://<value>`；listener 端口不是该协议的默认端口（HTTP 80 / HTTPS 443）时才追加 `:<port>`，IPv6 地址自动加方括号。

4. 多个 Gateway 时怎么选

- 列出集群内所有打了 label 的 Gateway，按 `namespace/name` 字典序取第一个满足上述条件的。这个顺序只是为了让结果可复现，不代表“更合适”——建议全集群只保留一个打 label 的 Gateway。

- 同一个 Gateway 内有多个 listener 时，HTTPS 优先于 HTTP；同协议时按 listener name 字典序。该顺序只影响自动推导出的 scheme 与端口；配了注解时对最终地址没有影响。

5. 生成的路由

实例创建时，ClawOS 在 **实例所在 namespace** 下创建名为 `agent-route-<instanceID>` 的 HTTPRoute，`parentRefs` 指向选中的 Gateway，并带 `ownerReferences`（实例删除时自动回收）。路径前缀：

路径前缀	后端
/agentclaw/<instanceID>/openclaw	OpenClaw UI
/agentclaw/<instanceID>/vnc	noVNC
/agentclaw/<instanceID>/files	filebrowser
/agentclaw/<callbackID>/msteams	MS Teams 回调（端口 3978）

各前缀通过 `ReplacePrefixMatch` 重写为 / 再转发到实例 Service。若 Gateway 前面还有一层反向代理，务必把这些路径原样透传，不要再做前缀剥离。

6. 生效时机与排查

- Gateway 只在 **实例创建 / 更新** 时评估一次。事后新增 Gateway 或修改注解，不会自动改写已有实例的 HTTPRoute 和访问地址，需要更新实例才会重新选择。
- 任何一环失败（没有候选 Gateway、没有可用 listener、注解非法、拿不到地址）都不会阻断实例创建，只是回退成 NodePort 直连。
- 排查时把 apiserver 日志级别调到 `-v=2`，搜 `No usable access gateway`，`reason` 字段里是逐个 Gateway 的具体原因，例如 `invalid agentclaw.io/base-url annotation: gateway base URL must not contain a path`。

（二）ClawOS 权限说明

ClawOS 支持四种用户角色：

- **Admin**：拥有 [工作空间视角](#) 和 [运维管理](#) 全部功能的增删改查的权限。
- **Workspace Admin**：拥有授权工作空间视角全部功能的增删改查的权限。
- **Workspace Editor**：拥有授权工作空间视角全部功能的更新、查询的权限。
- **Workspace Viewer**：拥有授权工作空间视角全部功能的查询的权限。

每种角色具有不同的权限，具体说明如下。

菜单对象	操作	Admin	Workspace Admin	Workspace Editor	Workspace Viewer
工作空间视角	—	—	—	—	—
概览	查看概览	有	有	有	有
ClawHub 管理	查看 Skill 市场	有	有	有	有
OpenClaw 实例	查看（仅可查看自己的 OpenClaw 实例）	有	有	有	无
OpenClaw 实例	创建	有	有	有	无
OpenClaw 实例	编辑（包含编辑/重启/开机/关机）	有	有	有	无
OpenClaw 实例	删除	有	有	有	无
OpenClaw 实例	查看所有实例（可以查看 workspace 下的全部 OpenClaw 实例）	有	有	无	无
OpenClaw 实例	管理所有实例（可以管理 workspace 下的全部 OpenClaw 实例）	有	无	无	无
管理员视角	—	—	—	—	—
概览	查看概览	有	有	无	无
ClawHub Skill 管理	查看	有	有	无	无
ClawHub Skill 管理	创建	有	有	无	无
ClawHub Skill 管理	编辑（包含上下架操作）	有	有	无	无

ClawHub Skill 管理	删除	有	有	无	无
---------------------	----	---	---	---	---

七、Release Notes 与升级

（一）版本演进

ClawOS 智能体面向企业级 AI 智能体使用场景，提供基于 DCE 的 OpenClaw 托管运行能力，帮助用户快速创建、访问和管理自己的 AI 数字员工。OpenClaw 不再只是被动回答问题的 AI 助手，而是能够主动规划任务、调用工具、执行代码、操作文件和完成端到端工作流的智能体运行时。

1. v0.3.3 (2026-05-31)

- **新增** 支持自定义 ClawOS 角色
- **新增** 支持 Skill 市场
- **优化** 创建 OpenClaw 实例时，自动对 API key 可见性进行隔离

2. v0.3.2 (2026-04-30)

- **新增** 一键创建 OpenClaw 实例
- **新增** 一键集成飞书
- **新增** 支持手动配置飞书应用信息
- **新增** 安全容器沙箱隔离
- **新增** 支持通过快捷入口一键访问 OpenClaw 实例
- **新增** 支持通过远程桌面访问 OpenClaw 实例
- **新增** 支持通过文件管理器上传文件
- **新增** 支持选择公有/私有模型创建 OpenClaw 实例
- **新增** 支持在管理员模式下查看工作空间下的 overview

从 v0.2.0 升级到 v0.3.2 有 breaking change，需要更新已有的 OpenClaw 实例，下载 agentclaw_upgrade_v0.3_patch_tool.tar.gz 工具，通过它自动完成更新。具体使用手册在压缩包中。下载地址：https://qiniu-download-public.daocloud.io/DaoCloud_Enterprise/agentclaw_upgrade_v0.3_patch_tool.tar.gz

3. v0.2.0 (2026-03-23)

AgentClaw Server

- **新增** ListWorkspaceAgentImage API，支持列举可用的 Agent 容器镜像
- **新增** Agent 实例（AgentInstance）完整 CRUD 接口：创建、查询、更新、删除
- **新增** Grafana 总览（Overview）仪表盘

- 修复 列举 Agent 实例时 Token 为空（nil）导致的异常
- 修复 Agent 实例列表分页（Pagination）错误
- 优化 API Base URL 统一增加 /v1 前缀
- 升级 前端 UI 至 v0.1.0

（二）升级注意事项

本部分说明将 ClawOS 从 v0.3.3 升级到 v0.5.3 时需要注意的事项。

1. 存量实例镜像升级

v0.3.3 和 v0.5.3 的默认 OpenClaw 实例镜像不同：

- v0.3.3: release.daocloud.io/agentclaw/openclaw:2026.4.7
- v0.5.3: release.daocloud.io/agentclaw/openclaw:2026.7.1

[openclawPrevious](#) 在 v0.5.3 中为 [2026.6.10](#)，不属于本次存量实例统一升级目标。

升级完成后，需要对系统中的存量实例逐个执行更新，使 OpenClaw 容器镜像切换到 [2026.7.1](#)。

2. SkillHub 配置

v0.5.3 默认关闭 ClawOS 的 SkillHub 能力，如需开启，请设置以下 values：

```
skillhub:
  enabled: true
  # 集群外执行 npx clawhub install 时可访问的节点 IP 或域名
  publicHost: <集群可访问的节点 IP 或域名>

skillhubChart:
  enabled: true
  secrets:
    scannerLlmApiKey: <LLM API key>
    scannerLlmBaseUrl: http://<内网模型网关>:<端口>/v1
    scannerLlmModel: openai/public/minimax-m25
  scanner:
    analyzers:
      llmProvider: openai
      useLlm: true
      useBehavioral: true
      enableMeta: true
```

其中：

- [skillhub.publicHost](#) 必须填写集群中可被 OpenClaw 实例访问的节点 IP。NodePort 会监听集群中的每个节点，不要求填写 master 节点。
- [skillhubChart.secrets.scannerLlmApiKey](#)、[scannerLlmBaseUrl](#)、[scannerLlmModel](#) 和 [scanner.analyzers.llmProvider](#) 用于配置 Skill Scanner 的 LLM 分析。
- [scannerLlmBaseUrl](#) 应指向 OpenAI 兼容的 /v1 端点；[scannerLlmModel](#) 使用 LiteLLM 路由格式，例如 [openai/public/minimax-m25](#)。

- `useLlm`、`useBehavioral` 和 `enableMeta` 同时开启时，Scanner 会组合 LLM 分析、行为分析和 meta 分析结果。

八、常见问题

（一）小龙虾的记忆会因为实例问题而丢失吗？

不会。无论是重启 OpenClaw、暂停实例（功能即将上线）还是释放实例，记忆都不会丢失。因为 `~/openclaw` 的数据都持久化存储在 DCE 的存储系统中。

（二）我的小龙虾安全吗？

是的。DaoCloud 的 ClawOS 是面向企业级的「养虾平台」：

- 使用容器技术作为龙虾的安全沙箱
- 基于 Kubernetes 技术实现计算/网络/存储各层级的数据安全保障
- 数据和对话都使用 DCE 大模型服务平台的大模型支撑，敏感数据得到保障

（三）可以把文件交给小龙虾分析吗？

当然可以。您可以通过以下方式：

1. 通过飞书聊天将文件发送给小龙虾处理
2. 通过 DCE 的文件管理界面上传和管理文件

（四）OpenClaw 实例创建失败

请确认：

1. 是否完成实名认证
2. 账号余额大于所选资源的单价

（五）飞书集成后不生效

一个飞书应用同时只能被一个 OpenClaw 连接使用，如果有多个 OpenClaw 实例，每个 OpenClaw 需使用独立的 App ID 或 Secret。