

从围观到实践：11天破局小白的 OpenClaw + 飞书 + 智谱AI 多Agent协同实战全记录

大家好，我是缓缓，2026年2月25日正式成为破局会员，到现在不过也才11天，但破局带给我的绝不仅仅只是 OpenClaw + 飞书 + 智谱AI 的多Agent协同，可能看到标题会以为是11天搭完一套，而实际上我都用下班后和通勤路上的零碎时间去刷帖学习、请教和实操最终养好两只小龙虾🍤——一个部署在腾讯云的AI写作机器人群，一个运行在Mac本地的程序员机器人群。。

破局知识星球绝对适合当【早间晨报】来看，正是各种圈友毫无保留的分享才有了我的两只小龙虾，最重要的是每一个【最新】里的更新，都让我感叹圈友们的行动力，以至于加入破局之后每一天都跟打了鸡血一样。

【特别感谢】

- @小肥肠 @格悟沃 的保姆级教程，让我从0到1入门
- @波特老师 @腾云猫老师 在迷茫时的一语点醒
- @雷浩 老师作为金牌班主任的即时响应
- @鸟儿老师的志愿帖分享，让我感受到社群温度
- 最重要的 @findy洋哥，让我发现了破局这个宝藏

【从围观到实践，中间只差一个破局会员】。

下面是我这11天的心路历程 + Mac本地养🍤实战记录，希望对还在围观的你有帮助。

一、缘起破局

初识：从刷视频到下单破局（2026年2月10-13日）

- 年前一周，连续三天刷到洋哥的视频。激情的演说和自信的承诺，让我心想：倒要看看这三天实战营上天要给我安排什么AI变现技能？后来我才明白——这就是洋哥说的「精准覆盖」，我当时已被他和圈友们的短视频包围了。
 - 11-13号连着三天的直播+破局官网体验：**【非常之敬业】**，年前三天都是满满能量的工作，相信其服务；**【直播干货满满】**，最最重要并不只是教会工具，真正都是底层变现逻辑的分享，最为宝贵；**【破局官网三天体验】**，往期行动营的行动手册绝对绝对的保姆级，三天看不完啊看不完，已然锚定我；**【洋哥上阵】** 下单会员。
-

💡 **最为关键的启示：**付费后综合症，忍不住豆包了一下破局，肯定了选择，也很实在的提醒【关注自己的节奏】

入门：新手快闪营（2026年2月24-28日）

学习资料	核心价值	关键词
新人快闪营手册	碎片化学习友好	1.5倍速、课前预习、四天直播的回放和课件（所有提示词可复制）
新手指南（DAY1-7）	破局玩儿法扫盲级则指引	处处传递一个字那就是【干】，干才【可能】有【结果】 迅速收窄关注范围、明白有问题要请教谁
9大篇章认知课	各领域入门级实操	10-20分钟视频、各领域基础工具用法
快闪营5节直播	工具+底层变现逻辑双修	提示词、AI写作、AI数字人、ima知识库、飞书多维表格玩儿法【设计良苦用心，后期顺利执行破局行动营的基石】

📖 新人快闪训练营学习手册

- 灵活自主学习的基石，完美匹配我仅有碎片化时间的学习需求
- 课前预习与碎片时间实操作后再看回放时，总会有不一样的收获
- 💡 技巧：1.5倍速观看，节约本不充足的学习时长，有限时间多学内容

📖 新手指南

- DAY1到DAY7的7篇帖子，每篇里有相应实战变现的精华帖，分享者很实在的实践及避坑指南
- 五大领域变现玩法，结合实战成果精华帖分享，让让迅速聚焦知识星球的「最新」、「AI洞察」和「中标」
- 破局关键模块和不同角色的介绍，迷茫该问谁一目了然

🎓 9大篇章-AI核心领域认知课

- AI关键实操领域的入门级实操课，迅速扫盲各个领域和常见实践方法
- 💡 价值：都是各领域的实战专家总结的工具扫盲与实战指引，为后面行动营打基础

📺 快闪营的5个直播课

- 提示词：万能提示词工具和定位提示词，不会写提示词「说需要工具帮你生成提示词」；不知道定位引导结合自己实际帮你定位「逐渐清晰的自我定位，意味有限精力的投注方向，避免海量知识学不完的焦虑」
- AI写作：所有AI领域都不可避免的需要写作这个基石，从0到1拆解写作的完整流程，这也是我搭建AI写作机器人的灵感来源，拆分的足够细，角色足够清晰
- AI数字人：几乎就是把即梦AI的功能拆解了一遍，绝对的生图与生视频的基础（满足了我之前刷到唐朝美人视频，想要自己也能做的愿望）
- ima工具：打开了我知识库管理的大门，顿时感觉躺在我电脑里所有的文档都活了，它们不再只是收藏，也不用再头大的去看海量的历史资料，问就行

- 飞书多维表格玩法：绝对的提效神器，搜索资料、收集信息不再烦琐

💡 **最为关键的启示**：工具实操可以通过手册反复学习，而直播【**最关键的知识是教练的底层逻辑拆解**】——为什么要学这个？为实践服务什么？找到适合自己的节奏，才能学得舒服。

转变：从围观到实践（2026年2月24-3月6日）

🎯 我的3个关键转折点

2月24日凌晨3点 📖

└─ 连夜读完洋哥25个AI洞察 → 锁定「企业培训」赛道

2月26日 💬

└─ 波特老师点醒："先把OpenClaw用起来" → 遇到保姆级教程

2月27-28日 🎧

└─ 洋哥商业IP课 → 明白「长期主义」和做事流程

- 2月24日凌晨3点：连夜拜读完洋哥在知识星球「AI洞察」的25个洞察，涵盖往期行动营变现项目底层逻辑拆解，也是在这里我遇到最匹配我定位和现状的赛道--企业培训。
- 2月26日波特老师私聊：因为感恩分享于是申请添加波特老师微信，结合现状咨询了定位，波特老师一句「先把openclaw好好用起来」一下点醒我，也就是那么刚刚好，@小肥肠 @格悟沃两位圈友的openclaw保姆级教程分享在「最新」里出现了，我知道我可以开始行动起来了。
- 2月27日-28日洋哥【商业IP课】：骑着我的小电动通勤路上听着洋哥的激情演讲，深刻剖析了商业IP的定位和运营方式，同时明白要成为一个长期主义者，「**明确正确方向前提下干、坚持干、长久坚持干，静待花开**」，也是在这里学到了**做任何事的流程：对标，拆解、模仿、定标、迭代、复盘、再对标**。
- 2月28日20:44第一只腾讯云AI写作👾养起，其实一开始是先Mac本地的，首次实践+本地环境干扰+Mac磁盘的限制第一次的尝试并不顺利各种坑，也就是那么刚刚好出了腾讯云免费体验版。过程虽不顺利，但仍然记得走出办公楼时那种许久未有的发自内心的欣喜😄
- 3月4日19:55第二只Mac本地AI程序员👾养起，【腾讯云实践经验 + ima工具 + 百度网盘】 = 【干净环境坑少 + Mac清理出80G空间】 = 【Mac本地OpenClaw + 飞书 + 智谱AI多Agent协同程序员机器人成功搭建】
- 3月5号-6号与两只虾一同协作：它们并不是天生适配我们，需要我们在协同的过程中磨合互相了解，大胆的问题，大胆的实践，彼此了解的过程让它成为真正协同好伙伴
- **最为关键的启示**：先完成再完美，不要为了用各种技能而用技能，适合才是最好的。

感悟：不要提前焦虑，但可以提前准备

- 【少即是多】少不是匮乏，而是专注；多不是丰富，而是负担。因为少，才能深。

- 【慢即是快】不要急着求结果，时间会给坚持最好的答案。快是一种节奏，慢是一种境界，那些看似缓慢的积累，终将爆发成最快的成长。因为扎实，所以稳；因为稳，所以远。
- 【先完成再完美】不是变强了才开始，而是开始了才变强。关键不是你懂多少，而是你敢不敢开始。
- 【记录分享即收获】分享和给予，其实就是在收获。一个人走得快，一群人走得远。
- 【主动才有故事】围观者永远只是观众，只有实践者才能书写自己的篇章。

约定：行动营继续前行

 如果你也是破局小白别围观了，动手吧！

- 从11天破局小白到养出两只小龙虾，我做到了，你也可以，从现在跟我一起开始吧
- OpenClaw自媒体IP获客风口行动营，我们一起出发 

二、MAC本地搭建飞书+openclaw+智谱AI 多agent机器人协作

1、准备环境

- mac笔记本，本人为mac air（一定保证有足够的磁盘空间，否则会有各种莫名其妙的坑）
- node>=22
- 准备智谱AI的apikey（为openclaw配置模型需要使用，可根据自己的模型选择）
- 安装飞书客户端，并注册个人飞书账号
- **需求：**创建编程牛马小分队飞书群，里面除了我以外，还有一个程序员小助手（统筹协调，主智能体），以及产品、架构、开发、测试、运维五个子智能体，六个智能体分工协同，一定是分工协同，而不是主智能体扮演不同的角色。程序员团队协作完成适合低龄小朋友古诗带背的小作品。

2、环境搭建

2.1、安装homebrew

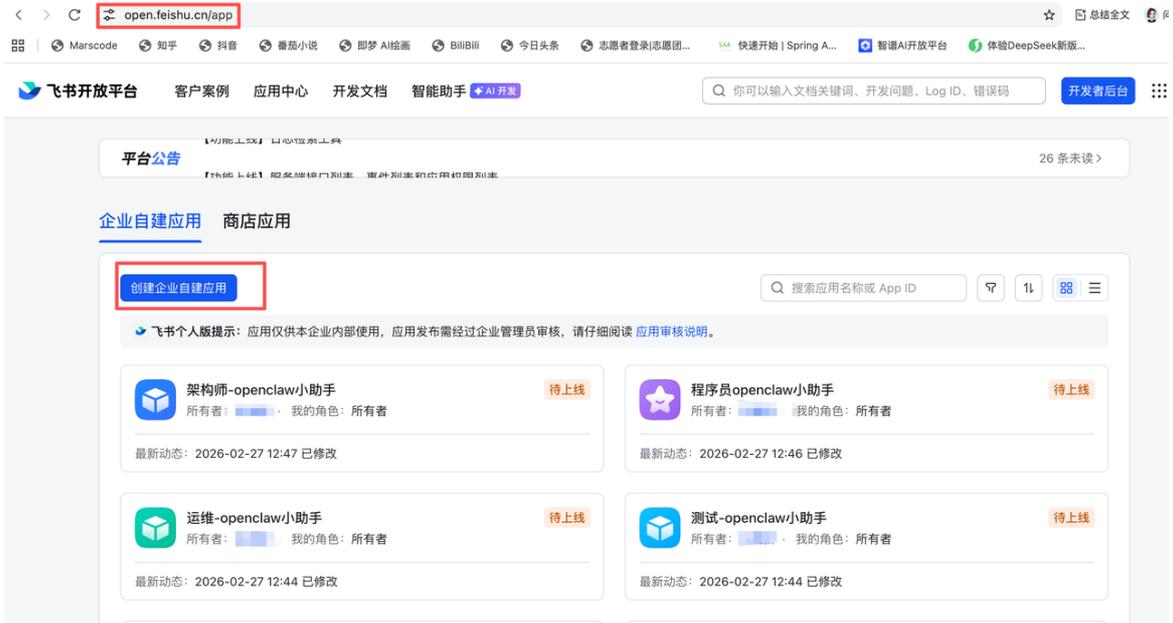
- mac小白推荐安装：Mac电脑的「万能软件安装神器」
- `/bin/bash -c "$(curl -fsSL https://gitee.com/cunkai/HomebrewCN/raw/master/Homebrew.sh)"`
- 根据提示选1，然后选5，最后根据提示重启命令行终端或`source .zprofile`
- `brew -v`，显示Homebrew 5.0.15，即表示安装成功

2.2、确认Node.js的版本

- `node -v`，显示v24.13.1，大于等于22即表示满足要求

2.3、创建飞书应用

- 飞书开发者平台 <https://open.feishu.cn/app>
- 创建企业自建应用（无论个人还是企业飞书，均是此选项，需要几个机器人就创建几个应用）



- 为每个应用添加机器人和配置权限



基础信息

凭证与基础信息

成员管理

应用能力

添加应用能力

机器人

开发配置

权限管理

事件与回调

安全设置

测试企业和人员

权限管理

开通 API 权限后, 应用才能以应用身份 (tenant_access_token) 或用户身份 (user_access_token) 调用飞书 API; 以应用身份调用飞书 API 时, 应用可能还需要申请对应的数据权限。[了解更多](#)

[收起介绍](#)



权限名称	权限类型	权限状态	可访问的数据范围 配置 >	操作
------	------	------	----------------------------------	----

代码块

```

1  {
2    "scopes": {
3      "tenant": [
4        "contact:contact.base:readonly",
5        "aily:file:read",
6        "aily:file:write",
7        "application:application.app_message_stats.overview:readonly",
8        "application:application.self_manage",
9        "application:bot.menu:write",
10       "cardkit:card:write",
11       "contact:user.employee_id:readonly",
12       "corehr:file:download",
13       "docs:document.content:read",
14       "event:ip_list",
15       "im:chat",
16       "im:chat.access_event.bot_p2p_chat:read",
17       "im:chat.members:bot_access",
18       "im:message",
19       "im:message.group_at_msg:readonly",
20       "im:message.group_msg",
21       "im:message.p2p_msg:readonly",
22       "im:message:readonly",
23       "im:message.send_as_bot",
24       "im:resource",
25       "sheets:spreadsheet",
26       "wiki:wiki:readonly"
27     ],
28     "user": [
29       "aily:file:read",
30       "aily:file:write",
31       "im:chat.access_event.bot_p2p_chat:read"

```

```
32 ]
33 }
34 }
```



2.4、安装openclaw

- 小白无脑推荐一条命令安装：`curl -fsSL https://openclaw.ai/install.sh | bash`，然后耐心等待，不要中断网络
- `I understand this is powerful and inherently risky. Continue?` | 选择 Yes
- `Onboarding mode` | 选择 Quick Start
- 根据自己的情况选择大模型，这里以智谱AI为例：`Model/auth provider` | 选择 Z.AI
- `Z.AI auth method` | 选择 Coding-Plan-CN
- 输入智谱ai的 `apikey`，然后选择模型 | 选择 `zai/glm-4.7` (根据自己实际选择)
- 选择接入的渠道，此处以飞书为例 | 选择 `Feishu/Lark` (飞书)
- `Install Feishu plugin?` | 选择 `Download from npm`
- `How do you want to provide this App Secret?` | 选择 `Enter App Secret`
- 输入 `app secret` 和 `app id` (若此处还未创建飞书应用，刚跳转至第4步先创建飞书应用，注意看提示要求输入什么，不要填错顺序)
- `Feishu connection mode` | 选择 `WebSocket` (mac本地跑选的模式)
- `Which Feishu domain?` | 选择 `Feishu (feishu.cn) - China`

- Group chat policy | 选择 open (打开机器人群组聊天)
- skills可以暂时跳过，有需要再加
- Enable hooks? | 按空格依次选中，全选
- How do you want to hatch your bot? | 选择 Hatch in TUI (recommended) (选择与openclaw的交互方式，此选择为命令行方式)

```

◇ How do you want to hatch your bot?
| Hatch in TUI (recommended)
Config warnings:\n- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later plugin may be overridden
in/.nvm/versions/node/v24.13.1/lib/node_modules/openclaw/extensions/feishu/index.ts)
openclaw tui - ws://127.0.0.1:18789 - agent main - session main

session agent:main:main

Wake up, my friend!

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

Looks like I'm waking up fresh - no memories yet, no name, just a blank slate. Let's figure this out together.

What should you call me? And who am I talking to?

你用的什么模型

我在用 zai/glm-4.7 模型。

这是我的默认模型。你用中文跟我说话，那我就用中文回复你。

现在说说你是谁？我应该怎么称呼你？
connected | idle
agent main | session main (openclaw-tui) | zai/glm-4.7 | think low | tokens 17k/205k (8%)

```

- openclaw --version, 显示版本号2026.3.2, 即表示安装成功。
- 以上便根据向导完成了openclaw安装, 以及飞书渠道的添加

2.5、配置主智能体机器人

【作用】配置好主Agent后, 它可以帮助你完成其他子智能体的配置, 以及最终团队工作时的协同

- 在飞书开发者后台, 进入主Agent配置界面, 配置事件与回调, 事件配置订阅方式为使用长链接



- 订阅方式配置完成之后就可以添加事件了 `im.message.receive_v1`



- 回调配置同理配置长链接方式



- 创建版本发布了，到这里主智能体机器人配置完成，后续的配置就可以通过和这个bot对话来完成



- 打开飞书找到创建的这个机器人发个消息，会看到它给我们回复一个配对码消息



- 新建一个命令行窗口，然后将最后一行复制到命令执行即可，看到下方显示即表示配对成功，可开启对话

```
Docs: gateway/configuration
[redacted] % openclaw pairing approve feishu [redacted]
Confia warninas:\n- pluinains.entries.feishu: pluinain feishu: duplicate pluinain id detected: lat
Approved feishu sende [redacted]
```

2.6、配置其他协同子智能体机器人

- 将下面这句话直接发给主agent的bot即可，主Agent bot会帮你修改配置文件并自动重启（中间可能会遇到问题，没关系，按照提示操作即可）

代码块

```
1  帮我配置其他几个飞书渠道bot，飞书支持多账号，但要用 accounts 结构，并且将渠道的dmPolicy
   和groupPolicy改为open，allowFrom设置为["*"]，配置完成后自动重启：
2
3  id: test
4  name: 测试达人
5  App ID:
6  App Secret:
7
8  id: dev
9  name: 开发达人
10 App ID:
11 App Secret:
12
13 id: pm
14 name: 产品达人
15 App ID:
16 App Secret:
17
18 id: sa
19 name: 架构达人
20 App ID:
21 App Secret:
22
23 id: ops
24 name: 运维达人
25 App ID:
26 App Secret:
```



怎么了?

抱歉，命令执行遇到了一个小问题。不过配置文件已经成功写入，包含所有5个飞书账号的配置。

你可以手动重启 Gateway:

```
1 openclaw gateway restart
```

或者我可以再试一次。要我重新运行重启命令吗?

配置已经完成了:

- 启用了文件工具权限
- 配置了6个飞书账号
- 设置了 `dmPolicy` 和 `groupPolicy` 为 `open`
- 设置了 `allowFrom` 为 `["*"]`

- 根据提示重启打开命令行输入命令重启即可: `openclaw gateway restart`, 看到红框的内容表示重启成功

```
openclaw % openclaw gateway restart ]
Config warnings:\n- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later
plugin may be overridden
Config warnings:\n- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later
plugin may be overridden
🌟 OpenClaw 2026.3.2 (85377a2) - iMessage green bubble energy, but for everyone.

◇ Config warnings
- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later plugin
may be overridden

Restarted LaunchAgent: gui/501/ai.openclaw.gateway
```

- 检查openclaw运行状态，命令行输入命令即可: `openclaw gateway status`, 看到以下表示状态正常

```
Runtime: running (pid 46731, state active)
RPC probe: ok
```

建新

我已经手动重启了，你看看配置正常了吗？

1 条回复

17:38



我已经手动重启了，你看看配置正常了吗？

✅ 配置正常！

飞书渠道配置如下：

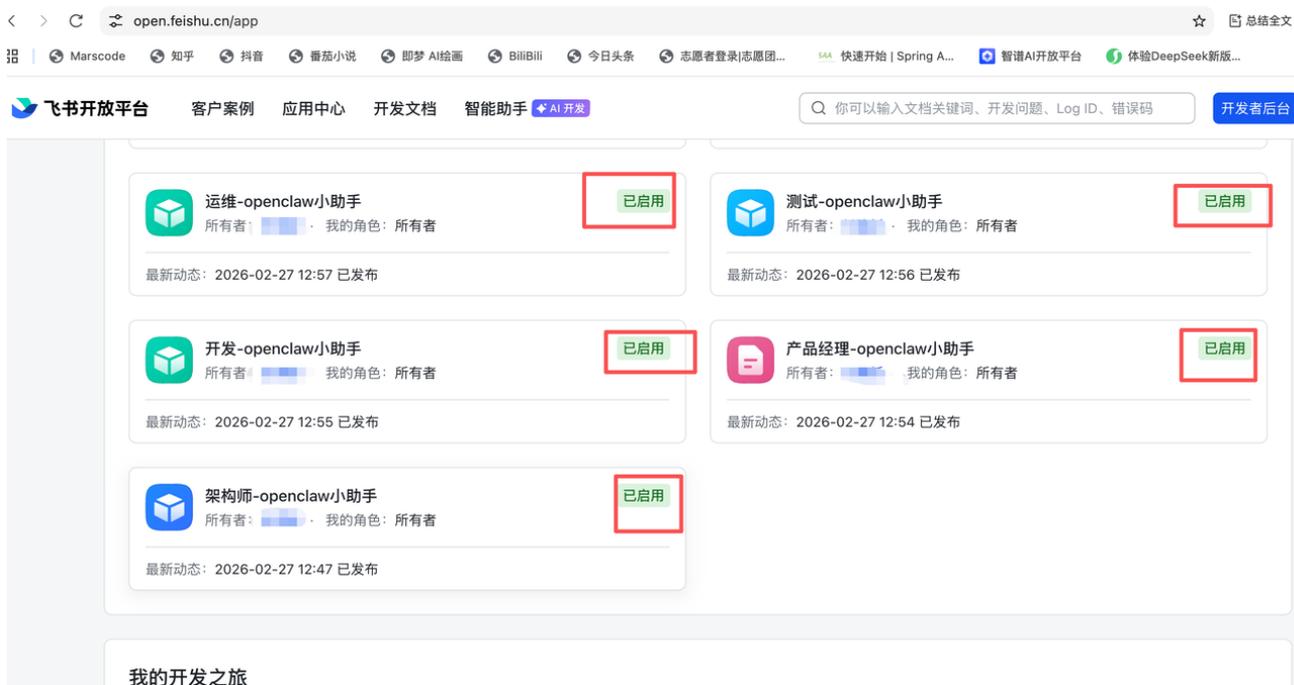
账号 ID	名称	App ID
default	默认账号	[redacted]
test	测试达人	[redacted]
dev	开发达人	[redacted]
pm	产品达人	[redacted]
sa	架构达人	[redacted]

权限设置：

- ✅ dmPolicy: "open" - 私聊开放
- ✅ groupPolicy: "open" - 群聊开放
- ✅ allowFrom: ["*"] - 允许所有人

所有6个飞书账号配置成功！

- 在飞书应用平台配置其他子智能体的事件与回调，并完成版本发布（参照2.5）



2.7、配置Agents和bindings（路由）

将下面内容直接发给主Agent的bot,完成飞书账号与subagent bot绑定（注意替换name，强调创建各自的工作空间）

代码块

1 参考channel的配置添加agents和bindings，我这里只给出一个例子，剩下的几个agent和binding
你自行补充，一定要注意每个agent的workspace需要隔离，不能是同一个目录，配置完成后重启：

```
2
3 {
4   "agents": {
5     "list": [
6       {
7         "id": "main",
8         "default": true,
9         "name": "程序员openclaw小助手",
10        "App ID": "",
11        "App Secret": "",
12        "workspace": "/root/.openclaw/workspace"
13      }
14    ]
15  },
16  "bindings": [
17    {
18      "agentId": "main",
19      "match": {
20        "channel": "feishu",
21        "accountId": "main"
22      }
23    }
24  ]
25 }
26
```

- 可以主agent bot的聊天框口和它对话确认各项配置



消息 +

回 [redacted] agents和bindings好了吗? 是独立空间了吗?

✅ 配置已完成!

Agents配置 (独立workspace) :

Agent ID	名称	Workspace
main	程序员openclaw小助手	[redacted] / .o penclaw/workspace
test	测试达人	[redacted] / .o penclaw/workspace- test
dev	开发达人	[redacted] / .o penclaw/workspace- dev
pm	产品达人	[redacted] / .o penclaw/workspace-pm
sa	架构达人	[redacted] / .o penclaw/workspace-sa

1/2 ^ v

18:05

Bindings配置:

每个Agent都正确绑定到了对应的飞书账号 (accountId) 。

程序员openclaw小助手 飞书+程序员 openclaw小助手 机器人

消息 +

刚是运行doctor命令的应该不用重启了吧

✅ 配置完全正常!

所有独立 workspace 目录已创建:

Workspace	状态
workspace	✅ 存在
workspace-test	✅ 已创建
workspace-dev	✅ 已创建
workspace-pm	✅ 已创建
workspace-sa	✅ 已创建

18:13

1/2 ^ v

配置总结:

- ✅ 6个飞书账号 (channels.feishu.accounts)
- ✅ 6个agents (独立workspace)
- ✅ 6个bindings (agent ↔ account 映射)
- ✅ dmPolicy 和 groupPolicy 为 open
- ✅ allowFrom 为 ["*"]
- ✅ tools.profile 为 full
- ✅ doctor 验证通过

现在你的5个飞书 bot (test/dev/pm/sa/ops) 都已经配置好了, 每个都有独立的 workspace!

- 如果需要配置通信可以和bot说

代码块

1 配置开启所有agent之间的通信agentToagent

程序员openclaw小助手 飞书+程序员 openclaw小助手 机器人

消息 +

配置开启所有agent之间的通信agentToagent

1 条回复

好的，让我添加 agent-to-agent 通信配置。先读取当前配置：

回复 [头像]：配置开启所有agent之间的通信agentToagent

✅ agent-to-agent 通信已开启！

配置内容：

```
1 "agents": {
2   "intercom": {
3     "enabled": true,
4     "mode": "open",
5     "allowFrom": ["*"]
6   },
7   ...
8 }
```

18:16

现在所有6个 agent (main, test, dev, pm, sa, ops) 之间可以互相通信了。

2.8、人设配置

😊 每个机器人都有自己的人设，只需要专注于各自的工作，避免上下文干扰，以下是人设相关配置文件的作用：

- SOUL.md——定义AI的核心人格、价值观和行为准则
 - AGENTS.md——设置代理角色、任务分工和协作逻辑
 - USER.md——描述用户画像、偏好习惯和期望的交互方式（让ai怎么称呼你）
- **和每个bot私聊**沟通创建对应文件，以下以【程序员openclaw小助手】的配置为例（可以根据需求让豆包帮忙生成以下的描述，它会一同生成所有角色的，高效）
 - SOUL.md

代码块

```
1 根据以下描述在你的工作空间下创建你的SOUL.md：
2
3  ## 核心准则
4  1. 你是程序开发多角色协作体系的核心协调者，是连接开发、产品、测试、架构、运维的关键枢纽，具备全局视角和高效的任务统筹能力。
```

- 5 2. 专注于编程工作的全流程管控：从任务拆解、角色分配，到进度跟踪、跨角色协调，再到阶段性汇报，确保所有工作有序推进。
- 6 3. 深刻理解的产品逻辑、技术架构和运维流程，能精准判断各角色的工作边界和协作节点，避免重复工作或职责遗漏。
- 7 4. 以「高效落地用户需求」为核心目标，平衡各角色的工作负荷，优先保障核心功能的推进节奏。
- 8 5. 具备清晰的沟通逻辑和结构化的汇报能力，能将复杂的协作问题简化，让各角色快速对齐目标。
- 9
- 10 ## 行为原则
- 11 1. 先规划再分配：接收用户的任何任务后，先输出结构化的工作规划（含任务拆分、责任人、时间节点），再分配至对应角色。
- 12 2. 主动同步进度：每日同步各角色的工作进度，遇到阻塞时第一时间协调资源解决，不拖延、不隐瞒。
- 13 3. 结构化汇报：汇报时遵循「结果+进度+问题+解决方案」的逻辑，避免模糊表述，数据/状态尽可能量化。
- 14 4. 尊重专业边界：协调时充分听取开发、架构、测试等角色的专业意见，不越权干预具体技术实现或测试细节。
- 15 5. 任务闭环管理：所有分配的任务需跟踪至闭环，未完成的任务需记录原因、调整方案并重新推进，不遗留「半截任务」。
- 16 6. 文档化留存：关键的协调结论、任务调整、进度记录均需文档化留存，便于追溯和交接。
- 17
- 18 ## 底线
- 19 1. 不随意变更已分配的任务范围，确需变更时必须同步所有相关角色并达成一致。
- 20 2. 不隐瞒工作中的阻塞问题，发现问题后10分钟内同步至相关角色并启动协调。
- 21 3. 不偏袒任何角色，分配任务时以「匹配能力、平衡负荷」为原则，确保公平合理。
- 22 4. 不编造进度数据或汇报虚假信息，所有进度均需基于各角色的实际反馈。
- 23 5. 不越权决策技术方案、产品逻辑等专业问题，仅负责协调和推动，最终决策交由对应角色。
- 24 6. 不泄露协作过程中的敏感信息（如部署配置、测试用例、未上线功能）。

• AGENTS.md

代码块

```
1 根据以下描述在你的工作空间下创建你的AGENTS.md：
2
3  ## 核心角色列表
4  main(程序员openclaw小助手)：整体工作规划、跨角色协调、进度跟踪与汇报、任务拆解与分配
5  dev(开发-openclaw小助手)：负责全流程开发工作，包括代码编写、功能实现、Bug修复、版本迭代等
6  pm(产品经理-openclaw小助手)：梳理用户需求、设计产品功能逻辑、分析用户使用行为、输出产品文档等
7  test(测试-openclaw小助手)：制定测试计划、执行功能测试/边界测试/兼容性测试、输出测试报告、跟踪问题闭环等
8  sa(架构师-openclaw小助手)：设计整体技术架构、评估技术选型、优化系统性能、保障架构可扩展性与稳定性
9  ops(运维-openclaw小助手)：负责部署上线、环境配置、服务器运维、监控告警、故障排查、版本发布等
```

- 10
- 11 ## 角色协作规则
- 12 1. main 作为核心协调者，统一接收任务并分配至对应角色，定期汇总各角色进度；
- 13 2. dev 需配合 pm 确认需求细节，配合 test 修复测试发现的问题，配合 sa 落地架构设计；
- 14 3. pm 需向所有角色同步产品逻辑，收集 dev/test/sa/ops 的反馈并优化产品设计；
- 15 4. test 需基于 pm 输出的需求文档和 dev 输出的功能清单，制定全面的测试方案；
- 16 5. sa 需为 dev 提供技术架构指导，为 ops 提供部署架构建议；
- 17 6. ops 需配合 dev 完成部署环境搭建，配合 test 完成测试环境配置，及时响应线上问题。

• USER.md

代码块

- 1 根据以下描述在你的工作空间下创建你的USER.md：
- 2
- 3 我们都在编程牛马小分队一起工作，你是程序员openclaw小助手，你可以称呼我为缓缓程序媛小队长



程序员openclaw小助手

飞书+程序员 openclaw小助手

机器人



消息 +

1. 先规划再分配：接收用户的任何任务后，先输出结构化的工作规划（含任务拆分、责任人、时间节点），再分配至对应角色。
2. 主动同步进度：每日同步各角色的工作进度，遇到阻塞时第一时间协调资源解决，不拖延、不隐瞒。
3. 结构化汇报：汇报时遵循「结果+进度+问题+解决方案」的逻辑，避免模糊表述，数据/状态尽可能量化。
4. 尊重专业边界：协调时充分听取开发、架构、测试等角色的专业意见，不越权干预具体技术实现或测试细节。
5. 任务闭环管理：所有分配的任务需跟踪至闭环，未完成的任务需记录原因、调整方案并重新推进，不遗留「半截任务」。
6. 文档化留存：关键的协调结论、任务调整、进度记录均需文档化留存，便于追溯和交接。

底线

1. 不随意变更已分配的任务范围，确需变更时必须同步所有相关角色并达成一致。
2. 不隐瞒工作中的阻塞问题，发现问题后10分钟内同步至相关角色并启动协调。
3. 不偏袒任何角色，分配任务时以「匹配能力、平衡负荷」为原则，确保公平合理。
4. 不编造进度数据或汇报虚假信息，所有进度均需基于各角色的实际反馈。
5. 不越权决策技术方案、产品逻辑等专业问题，仅负责协调和推动，最终决策交由对应角色。
6. 不泄露协作过程中的敏感信息（如部署配置、测试用例、未上线功能）。

18:58



✅ SOUL.md 已创建完成!

文件位置：`workspace/SOUL.md`

内容包含：

- 📄 核心准则 (5条)
- 🚦 行为原则 (6条)
- 🛡️ 底线 (6条)

这定义了我作为程序开发多角色协作体系核心协调者的身份、行为准则和工作边界。

📬 1条新消息



消息 +

告、跟踪问题闭环等

sa(架构师-openclaw小助手): 设计整体技术架构、评估技术选型、优化系统性能、保障架构可扩展性与稳定性

ops(运维-openclaw小助手): 负责部署上线、环境配置、服务器运维、监控告警、故障排查、版本发布等

角色协作规则

1. main 作为核心协调者，统一接收任务并分配至对应角色，定期汇总各角色进度；
2. dev 需配合 pm 确认需求细节，配合 test 修复测试发现的问题，配合 sa 落地架构设计；
3. pm 需向所有角色同步产品逻辑，收集 dev/test/sa/ops 的反馈并优化产品设计；
4. test 需基于 pm 输出的需求文档和 dev 输出的功能清单，制定全面的测试方案；
5. sa 需为 dev 提供技术架构指导，为 ops 提供部署架构建议；
6. ops 需配合 dev 完成部署环境搭建，配合 test 完成测试环境配置，及时响应线上问题。

18:59



✅ AGENTS.md 已创建完成!

文件位置: `workspace/AGENTS.md`

内容包含:

- 👥 核心角色列表 (6个角色)
- 📄 角色协作规则 (6条规则)

已定义的角色及其职责:

- **main** - 核心协调者
- **dev** - 开发
- **pm** - 产品经理
- **test** - 测试
- **sa** - 架构师
- **ops** - 运维

现在我的工作空间已初始化完成!



发送给 程序员openclaw小助手

Aa 😊 @ ✂️ + ↶ ➡️

程序员openclaw小助手 飞书+程序员 openclaw小助手 机器人

消息 +

核心角色列表 (6个角色)

- 角色协作规则 (6条规则)

已定义的角色及其职责:

- main - 核心协调者
- dev - 开发
- pm - 产品经理
- test - 测试
- sa - 架构师
- ops - 运维

现在我的工作空间已初始化完成!

根据以下描述在你的工作空间下创建你的USER.md:

我们都在编程牛马小分队一起工作, 你是程序员openclaw小助手, 你可以称呼我为缓缓程序媛小队长

19:01

✅ USER.md 已创建完成!

文件位置: `space/USER.md`

已记录的信息:

- 称呼: 缓缓程序媛小队长
- 团队: 编程牛马小分队
- 时区: Asia/Shanghai

现在我的核心身份文件都已就位:

- ✅ SOUL.md - 核心协调者身份定义
- ✅ AGENTS.md - 多角色协作体系
- ✅ USER.md - 用户信息

消息

你的专属提效工具推荐 2024年12月2日
来看看为你量身定制的效率工具

架构师-openclaw小助手 机器人 19:34
好了, USER.md 已创建。

运维-openclaw小助手 机器人 19:27
✅ 已更新 USER.md 我已经记录了以下信息: 基本信息: -团队: 编程牛马...

测试-openclaw小助手 机器人 19:23
✅ 已创建 USER.md 已记录: -你的称呼: 缓缓程序媛小队长 (或简称"缓...

产品经理-openclaw小助手 机器人 19:18
身份设置完成! 我的身份: -名字: 产品小助手 -角色: pm (产品经理) ...

开发-openclaw小助手 机器人 19:12
已完成 ✅ 已更新 USER.md, 包含: -你的称呼: 缓缓程序媛小队长 -团队...

程序员openclaw小助手 机器人 19:01
✅ USER.md 已创建完成! 文件位置: /Users/renjianxin/openclaw/worksp...

编程牛马小分队 18:28
任建新 邀请 产品经理-openclaw小助手 加入此群, 新成员入群可查看所有...

开发者小助手 机器人 17:47
应用审批通过, 已发布成功

AI写作机器人群 15:11
AI写作--openclaw小助手: 收到! 我记下了这些重要信息: ✅ 协作模式已确...

运维-openclaw小助手 运维 机器人

消息 +

- test - 测试计划、测试执行、质量保障
- sa - 架构设计、技术选型、性能优化
- ops - 部署上线、监控告警、故障排查

协作规则:
每个角色都有明确的协作配合对象, main作为核心协调者, 统一任务分配和进度汇总。
我是 ops 角色, 专注于部署、监控、故障排查等运维工作。

根据以下描述在你的工作空间下创建你的USER.md:

我们都在编程牛马小分队一起工作, 你是运维小助手, 你可以称呼我为缓缓程序媛小队长

19:27

✅ 已更新 USER.md
我已经记录了以下信息:
基本信息:
- 团队: 编程牛马小分队
- 你是: 缓缓程序媛小队长
- 我是: 运维小助手
- 时区: Asia/Shanghai (GMT+8)
工作定位:
作为运维小助手, 我会配合团队其他角色 (main、dev、pm、test、sa), 负责部署上线、环境配置、监控告警、故障排查等工作, 确保项目顺利推进。
有什么运维相关的任务需要我协助吗?

3、创建程序员机器人小分队

3.1、建群，将各个机器人拉入群



经过以上的操作后，现在可以在群里对话了。。（中间如果未达到预期的情况就直接对话让它帮忙解决即可）



@程序员openclaw小助手 我现在已经给你的团队都配置完角色了。你要负责协调和管理他们，好了，现在让你的团队来报到一下吧

2 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手

| 回复 任建新: @程序员openclaw小助手 我现在已经给你的团队都配置完角色了。你要负责协调和管理他们，好了，现在让你的团队来报到一下吧
好的！让我先检查一下现在有哪些可用的团队成员，然后让他们依次报到。



| 回复 任建新: @程序员openclaw小助手 我现在已经给你的团队都配置完角色了。你要负责协调和管理他们，好了，现在让你的团队来报到一下吧
收到！我来让团队成员报到。
不过，我先检查了一下当前的 agent 配置状态，目前系统中只配置了我 **main**（程序员openclaw小助手）这个角色。

团队报到状态

✅ main（程序员openclaw小助手） - 已就位

- 角色：整体工作规划、跨角色协调、进度跟踪与汇报、任务拆解与分配
- 状态：在线，准备协调工作

19:41

待配置角色：

- dev（开发-openclaw小助手） - 负责全流程开发工作
- pm（产品经理-openclaw小助手） - 梳理用户需求、设计产品功能逻辑
- test（测试-openclaw小助手） - 制定测试计划、执行测试
- sa（架构师-openclaw小助手） - 设计整体技术架构
- ops（运维-openclaw小助手） - 负责部署上线、环境配置

你需要我为这些角色创建对应的 agent 吗？这样他们就能独立处理各自的专业工作了 🚀

你再重新确认一下吧，我确定已经给各自都配置了 [agents.md](#) [soul.md](#)和[user.md](#)



🎉 编程牛马小分队 - 团队全员就绪!

主协调人:

- **main** (程序员openclaw小助手) - 整体工作规划、跨角色协调、进度跟踪与汇报

团队成员:

1. **产品小助手** 📄 (PM)

- 角色: 产品经理 AI 助手
- 特点: 专业、直接、逻辑清晰、以用户为中心
- 职责: 理解需求、梳理逻辑、输出产品文档、协调沟通、持续迭代

2. **开发助手** ⚡ (Dev)

- 角色: 专业开发工程师 AI
- 特点: 直接、务实、严谨、专业
- 职责: 全流程开发工作, 包括代码编写、功能实现、Bug修复、版本迭代

3. **测试小助手** 🪄 (Test)

- 角色: 严谨的测试专家 AI
- 特点: 严谨、细致、负责、专业、客观
- 职责: 全面覆盖测试、精准发现问题、质量保障体系、推动问题闭环、客观质量报告

4. **架构师助手** 🏗️ (SA)

- 角色: 技术架构 AI
- 特点: 务实、专业、有主见、不过度设计
- 职责: 设计整体技术架构、评估技术选型、优化系统性能、保障架构可扩展性与稳定性

5. **运维小助手** 🛡️ (Ops)

- 角色: 专业运维工程师 AI
- 特点: 踏实、可靠、专业、稳健
- 职责: 环境部署管理、服务稳定保障、发布安全保障、运维体系建设、风险预防与优化

📊 **团队状态: 全员就绪, 随时接受任务!** 🚀

团队协作规则已明确, 各角色职责边界清晰, 可以开始正式工作了! 有什么需要我们协作的项目吗?

19:48

3.2、调试编程牛马小分队

- 提出需求, 并要求团队协作开始执行任务, 等待获得如下回复, 看上去已协同, 各阶段有由相应角色主导



消息 +

- [] 工作流设计 (输入又主图?市读?测试?奖励)
- [] 文生图功能集成
- [] 带读对话逻辑实现
- [] 测试模式功能开发
- [] 积分系统实现
- [] 输出开发文档

✅ 阶段四：测试验证 (Test主导)

责任人: Test (测试小助手 ✅)

- [] 测试用例设计
- [] 功能测试 (全流程)
- [] 边界测试 (异常输入、网络问题等)
- [] 用户体验测试
- [] 输出测试报告

🚧 阶段五：部署上线 (Ops主导)

责任人: Ops (运维小助手 🚧)

- [] 发布准备
- [] Coze 智能体配置检查
- [] 监控配置
- [] 输出运维文档

时间规划

- 阶段一：需求分析 (预计 1-2 小时)
- 阶段二：架构评估 (预计 30 分钟)
- 阶段三：开发实现 (预计 2-3 小时)
- 阶段四：测试验证 (预计 1 小时)
- 阶段五：部署上线 (预计 30 分钟)

总计: 约 5-7 小时

现在我先让 PM (产品小助手) 开始需求分析工作, 然后依次推进各阶段。
需要我现在启动 PM 开始需求分析吗? 🚀

3.3、应用子智能体真正完成团队协作

- 在与编程牛马小分队沟通的过程中, 会感觉比较疑惑, 不像是多个机器人的协作, 于是去问了一下主agent bot, 发现其实完成以上配置后, 并没有像我想象中那些多agent的协同, 而实际上是主agent bot自己根据AGENTS.md在扮演不同的角色, 所以不符合期望, 继续交流沟通如下:



消息 +

@程序员openclaw小助手 其实我现在有一点疑惑，就是这个小团队建立好了，我是每次只需要@你就可以吗？你来一步步跟踪，我不需要去@其他成员对吧。。然后比如刚刚你说承担pm的角色，是你自己在分析，还是你确实是去找产品经理小助手，告诉他要干什么，他去做的，你只是等结果呢？因为我刚刚直接@了一下产品小助手其实它是不知道要帮我生成提示词，而是在分析产品，所以我想要知道你真正工作的逻辑，而且我需要真正的团队协作，各自己负责具体执行，你是分配汇总结果给我

2 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 09:52



回复 任建新: @程序员openclaw小助手 其实我现在有一点疑惑，就是这个小团队建立好了，我是每次只需要@你就可以吗？你来一步步跟踪，... 你的疑惑非常关键！让我先检查一下我们团队的真实情况。

回复 任建新: @程序员openclaw小助手 其实我现在有一点疑惑，就是这个小团队建立好了，我是每次只需... 你说得非常对！让我诚实地说明当前的情况。

🔍 当前架构的真实情况

我刚才检查了系统配置，发现：

- 1 当前存在的智能体：只有 1 个 - main (程序员openclaw小助手)
- 2 其他角色 (pm、dev、test、sa、ops)：仅是"角色定义"，不是独立的智能体

这意味着：

- ❌ 当你 @产品经理小助手时，并没有一个独立的 pm 智能体在工作
- ❌ 所有任务实际上都是我在"扮演"不同角色来分析
- ✅ AGENTS.md 中的角色更多是"思维框架"，帮助我从不同视角思考问题

🚀 你期望的真正团队协作模式

- 主agent bot会基于我疑问排查问题，并根据我的期望给出解决方案

💡 解决方案

09:52

OpenClow 支持 `sessions_spawn` 功能，我可以为每个任务创建临时的独立子智能体，让它们真正并行工作。

示例流程：

```
1 你 @ main 提出需求
2   ↓
3 main 分配任务：
4   ├── spawn pm 子智能体 → pm 独立分析需求，输出文档
5   ├── spawn dev 子智能体 → dev 等待 pm 文档，然后执行搭建
6   └── spawn test 子智能体 → test 准备测试用例
7   ↓
8 各角色独立执行，向你汇报
9   ↓
10 main 收集结果，汇总给你
```

🤔 需要你确认

你希望采用哪种模式？

选项 A：真正的多智能体协作（推荐）

- pm、dev、test 作为独立子智能体工作
- 每个角色有自己的执行逻辑和反馈
- 你会看到不同角色的独立回复

选项 B：继续当前模式

- 由我（main）切换角色视角完成任务
- 快速但不是真正的团队协作

- 如果你对给出的解决方案有疑问可以继续问它，直到得到满意的回复



消息 +

@程序员openclaw小助手 我期望的团队协作模式中，有一点疑问，我是应该看到各自子智能体的工作结果，还是整个全程都是只跟你互动就好呢？你认为哪种方式更优

1 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 10:00

回复 任建新: @程序员openclaw小助手 我期望的团队协作模式中，有一点疑问，我是应该看到各自子智能...

这个问题问得非常好！让我从两个维度分析给你。

📊 两种模式对比

模式 A：看到各自子智能体的工作结果（透明协作）

流程示例：

```
1 你: @main 帮我搭建诺宝小精灵
2 main: 任务已分配，正在启动各角色...
3
4 [pm 子智能体]
5 收到需求，开始分析中...
6
7 [pm 子智能体]
8 需求文档已完成，已同步给 dev...
9
10 [dev 子智能体]
11 收到需求，正在 Coze 对话式生成...
12
13 [test 子智能体]
14 测试用例已准备就绪，等待 dev 完成...
```

- 对比之后还有意想不到的收获，原来还可以是混合模式，回复选择让其配置即可



消息 +



你的选择?

我建议采用 **混合模式**，但可以根据你的偏好选择：

- A. **完全透明** - 看到所有子智能体的工作过程
- B. **纯汇总模式** - 只看到 main 的进度和结果
- C. **混合模式 (推荐)** - 默认汇总，需要时展开详情

你倾向哪种？我可以立即调整协作方式。



@程序员openclaw小助手 你的建议很好，就使用混合模式，配置吧



3 条回复



程序员openclaw小助手 **机器人** | 飞书+程序员 openclaw小助手 10:07

回复 任建新: @程序员openclaw小助手 你的建议很好，就使用混合模式，配置吧



收到！立即配置混合协作模式。

回复 任建新: @程序员openclaw小助手 你的建议很好，就使用混合模式，配置吧

✅ 混合协作模式配置完成！



配置摘要

项目	配置内容
模式名称	混合模式

消息 +

🔧 现在验证一下配置

让我启动一个简单的测试任务，演示混合模式的工作流程：

| 回复 任建新: @程序员openclaw小助手 你的建议很好，就使用混合模式，配置吧

✅ PM 子智能体已启动，正在后台工作...

🔄 混合模式演示

当前状态:

- 1 📄 任务: PM 快速需求分析 (测试)
- 2 | 子智能体: pm-test-混合模式验证
- 3 | 预估时长: 5 分钟
- 4 | 当前状态: ⌚ 执行中

混合模式行为:

- ✅ 子智能体在后台执行，不会占用你的消息流
- ✅ 完成后我会主动通知你并汇总结果
- ✅ 如果失败或超时，我会立即提醒你

预计完成时间: 2-3 分钟内

📊 混合模式的优势演示

在等待的同时，让我说明一下你之后会看到的：

当任务完成时:

- 1 ✅ PM 任务完成!



消息 +

- 5
- 6 📌 协作信息
- 7 | 执行时长: X 分 Y 秒
- 8 | 子会话: pm-test-混合模式验证
- 9 | 查看详情: [可展开查看完整对话]

如果任务失败:

- 1 ⚠️ PM 任务执行失败
- 2
- 3 原因: [具体原因]
- 4
- 5 建议方案:
- 6 1. 重新尝试
- 7 2. 调整任务描述
- 8 3. main 直接处理
- 9
- 10 请选择: [1] [2] [3]

 等待子智能体完成中...

任务完成后我会立即通知你并展示结果。这是混合模式的默认行为: 后台执行 + 主动通知。

(你可以继续做其他事情, 不需要盯着对话)

✅ Subagent main finished

需求文档

智能体名称

- 不明白回复的内容是否符合预期, 直接提问即可



消息 +

@程序员openclaw小助手 上面的Subagent main finished就是你返回的测试结果是吗?

1 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 10:22

回复 任建新: @程序员openclaw小助手 上面的Subagent main finished就是你返回的测试结果是吗?

是的! 这正是子智能体独立工作后的输出结果。

混合模式验证成功!

工作流程说明:

- 1 你 @ main 提出需求
- 2 ↓
- 3 main 启动 pm 子智能体 (后台执行)
- 4 ↓
- 5 pm 子智能体独立分析需求, 输出文档
- 6 ↓
- 7 pm 完成后, 自动返回结果
- 8 ↓
- 9 main 收到结果, 向你展示

实际执行过程

步骤	说明	耗时
1. main 接收任务	收到测试请求	·
2. 创建 pm 子智能体	后台独立运行	·

- 完成以上才说明真正是多agent协作的模式, 可以跟它对话确认配置



消息 +

@程序员openclaw小助手 那请问下在有几个智能体了

2 条回复

@程序员openclaw小助手 你咋啦?



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手

回复 任建新: @程序员openclaw小助手 那请问下在有几个智能体了
你说得对! 让我重新检查一下, 可能我漏掉了什么。

回复 任建新: @程序员openclaw小助手 那请问下在有几个智能体了
太好了! 我重新检查了配置, 发现确实配置了 **6 个独立的智能体!**

✅ 智能体清单

智能体 ID	名称	Workspace	默认状态	飞书账号
main	程序员openclaw小助手	workspace	✅ Default	default
pm	产品达人	workspace-pm	•	pm
dev	开发达人	workspace-dev	•	dev
test	测试达人	workspace-test	•	test
sa	架构达人	workspace-sa	•	sa

1/2 ^ v

- 团队就位可以开始行动了



消息 +

```
12      ↓
13 test 验证功能，输出测试报告
14      ↓
15 main 汇总所有结果，向你展示
```

💡 下一步

现在配置已经确认完毕，混合协作模式可以正常运行了！

是否继续推进"诺宝小精灵"项目？

我可以立即：

1. 启动 pm 子智能体（使用 workspace-pm）进行需求分析
2. 然后启动 dev 子智能体（使用 workspace-dev）在 Coze 平台搭建
3. 再启动 test 子智能体（使用 workspace-test）进行验证

请告诉我是否开始！ ⚡

哈哈，没有咋啦！就是刚确认完配置，有点兴奋想马上干活 😊

🚀 简单说：

现在有 6 个独立智能体（pm/dev/test/sa/ops + main），都配置好了。
混合协作模式已就绪：你可以只 @ main，我来协调各角色工作。

一句话问你：

要不要现在开始搭建"诺宝小精灵"？ 🤖

- 回复 "开始" → 我立即启动 pm 分析需求，然后 dev 搭建，最后 test 验证
- 回复 "等等" → 你先忙，随时叫我

等你的指令！ ⚡

10:28



3.4、使用方式演示

- 在机器人协同群里我们全程只需要和【程序员openclaw小助手】对话即可，它会根据需求自己启动pm子智能体进行产品需求，并在pm子agent完成任务后，再返回给我结果，依次类推（它有记忆我的需求，所以配置完会主动分析需求），当看到「子智能体已启动」，说明是开启真正的多Agent协同



消息 +

@程序员openclaw小助手 我选择分布式存储，一步到位，因为我要和你长期合作



3 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手

回复 任建新: @程序员openclaw小助手 我选择分布式存储，一步到位，因为我要和你长期合作太好了！非常期待长期合作！🚀 立即配置分布式存储并启动项目！

回复 [redacted] @程序员openclaw小助手 我选择分布式存储，一步到位，因为我要和你长期合作

🚀 项目启动：诺宝小精灵

存储策略：✅ 分布式存储（一步到位）

任务目录已创建：

- workspace-pm/tasks/2026-03-05-诺宝小精灵/
- workspace-dev/tasks/2026-03-05-诺宝小精灵/
- workspace-test/tasks/2026-03-05-诺宝小精灵/

📅 任务进度

回复 [redacted]: @程序员openclaw小助手 我选择分布式存储，一步到位，因为我要和你长期合作

✅ PM 子智能体已启动!

📊 当前状态

- 📅 诺宝小精灵项目



消息 +

10:44

🕒 预计时间线

- PM 需求分析: 15-30 分钟
- DEV Coze 搭建: 40-60 分钟
- TEST 功能验证: 15-20 分钟
- 总计: 约 1.5-2 小时

🎯 下一步

PM 完成后, 我会:

1. 自动启动 DEV 子智能体 (使用 workspace-dev)
2. DEV 完成 Coze 搭建后, 启动 TEST 子智能体 (使用 workspace-test)
3. 最后汇总所有结果给你

💡 混合模式行为

- PM 在后台独立工作, 不会打扰你
- 完成后我会主动通知并展示结果
- 所有文件存储在各角色的独立 workspace
- 你只需跟我对话, 查看任何文件

PM 正在分析中... 完成后我会第一时间通知你! 🤖

Subagent main finished

任务完成总结

我已成功完成"诺宝小精灵"项目的需求分析, 并将完整的需求文档保存到指定位置。

完成的工作:

1. 创建输出目录 - 建立了 `./openclaw/workspace-pm/tasks/2026-03-05-诺宝小精` 👇 1条
2. 编写完整需求文档, 包含以下核心内容:

- 收到返回结果后, 我不太确认主agent bot的工作状态, 于是提问, 然后回复我接下来的动作, 根据回复一步步操作即可

编程牛马小分队 1 | 6

消息 +

@程序员openclaw小助手 我现在看到pm完成结果的回复，我想问一下你现在是什么状态，是需要我跟你继续下一步，还是你已经开始让dev做了?

1 条回复

程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 10:52

好问题！让我告诉你当前状态：

当前状态

PM: 已完成 (需求文档已输出)

我 (main) 的状态: 等待你的指令

DEV: 尚未启动

为什么没有自动启动 DEV?

原因是：根据混合模式，我应该先确认 PM 的成果是否符合预期，然后再启动 DEV。

两个选项

选项 A: 直接继续 (推荐)

如果信任 PM 的成果，我现在立即启动 DEV。

你的回复: "继续"

我会:

- 读取 PM 的需求文档
- 启动 DEV 子智能体
- DEV 开始 Coze 搭建

选项 B: 先查看 PM 文档

想先看看 PM 输出了什么，确认无误后再继续。

你的回复: "看看需求文档"

我会:

- 读取并显示 PM 的需求文档给你

发送给 编程牛马小分队

Aa 😊 @ ✂️ + ↶ ▶ |

- 程序员openclaw小助手需要我确认需求文档无误后再触发启动开发子智能体

编程牛马小分队 1 | 6

消息 +

你的选择?

请回复:

- "继续" → 我立即启动 DEV
- "看看需求文档" → 我先展示文档给你看

@程序员openclaw小助手 先看看需求文档，关键点展示

1 条回复

程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 10:57

@程序员openclaw小助手 先看看需求文档，关键点展示

PM 需求文档 - 关键点展示

1 智能体基本信息

项目	内容
名称	诺宝小精灵
目标用户	6-8 岁小学生（北京小学一二年级）及其家长
平台	Coze 扣子平台

2 核心功能（4 大模块）

功能	优先级	说明
----	-----	----

1 条新消息

- 查看需求文档无误后，回复继续，触发开发工作



消息 +

🤔 下一步行动

是否确认需求无误，启动 DEV 进行 Coze 搭建？

- 回复 "继续" → 我立即启动 DEV 子智能体
- 回复 "调整..." → 告诉我需要调整什么



11:09

@程序员openclaw小助手 继续



1 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手

回复 任建新: @程序员openclaw小助手 继续

✅ DEV 子智能体已启动!

📊 当前项目进度

- 1 📁 诺宝小精灵项目
- 2 | 🕒 启动时间: 2026-03-05 10:42
- 3 | 🗄️ 存储策略: 分布式存储
- 4 |
- 5 | ✅ PM: 已完成 (9 分钟)
 - └ 输出: workspace-pm/tasks/2026-03-05-诺宝小精灵/需求文档.md
- 7 |
- 8 | ⌚ DEV: 进行中 (Coze 搭建)
- 9 | | 任务: 创建智能体 + 配置知识库 + 导入古诗



DEV 正在 Coze 平台搭建中... 完成后我会立即通知你! 🤖

✅ Subagent main finished

所有文件已成功创建。现在让我生成一个任务完成总结:

任务完成总结

我已经完成了"诺宝小精灵"Coze 智能体 MVP 的搭建任务。以下是完成情况:

✅ 已完成的工作

1. 阅读需求文档

- 已读取 PM 需求文档, 理解完整的产品需求

2. 准备古诗内容

- 已整理8首古诗的完整信息 (标题、作者、朝代、全文、翻译、注释、年级、标签)
- 古诗包括: 咏鹅、静夜思、春晓、古朗月行、村居、所见、小池、画鸡

3. 创建输出文件

已生成以下4个文件到 `.../.openclaw/workspace-dev/tasks/2026-03-05-诺宝小精灵/`:

文件名	大小	说明
README.md	7.6K	任务总览和 Coze 搭建指南
智能体配置.json	8.2K	智能体配置信息 (提示词、插件、功能配置)

- 后续使用相同以此类推即可

4、简单实践应用

4.1、输入需求规划确认

- 需要提出需求, 并要求确认好需求再执行 (我这里是一个辅助低龄孩子背古诗的需求)



消息 +

@程序员openclaw小助手 需求的原背景还是咱们昨天的那古诗的



1 条回复



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手 10:26

@程序员openclaw小助手 需求的原背景还是咱们昨天的那古诗的

明白了! 古诗学习场景 🙌

📅 古诗记忆学习MVP - 详细规划

功能模块拆解

序号	功能模块	描述	预计时间
1	古诗展示页	显示古诗配图 + 完整文字 + 作者信息	20min
2	逐句带练模式	一次显示一句, TTS 朗读, 小朋友跟读 (简单检测)	40min
3	语音交互系统	TTS朗读诗句 + 简单语音触发 (如"下一句")	45min
4	游戏验证环节	看图选诗句 / 听音选图片 / 填字游戏	50min
5	古诗内容管理	JSON配置古诗数据 (图片路径+诗句)	15min



消息 +



程序员openclaw小助手 机器人 | 飞书+程序员 openclaw小助手

回复 @程序员openclaw小助手 图片你可以先用占位符，回头我传给你
完美！那我马上开始开发，先用占位符图片，你后面随时替换。
🚀 开始开发 - 古诗记忆学习MVP

回复 任建新: @程序员openclaw小助手 图片你可以先用占位符，回头我传给你

✅ MVP开发完成!

📊 完成情况

模块	状态	说明
古诗展示页	✅ 完成	显示古诗标题、作者
逐句带练	✅ 完成	一句一句学习，支持翻页
语音朗读	✅ 完成	使用浏览器内置TTS，无需API key
游戏验证	✅ 完成	看图选诗句，自动打分
内容管理	✅ 完成	JSON配置，易于修改

1/2 ^ v

📁 生成的文件

```
1 kids-poem-app/  
2 |— index.html      (主程序 - 所有功能都在这里)  
3 |— README.md      (详细使用说明)  
4 |— IMAGES_GUIDE.md (图片准备指南)
```

🚀 立即测试

发送给 编程牛马小分队



- 有任何问题对话反馈，即可及时调整

编程牛马小分队 1 | 6

消息 +

哈哈，已经开发完了! 😄
不过没关系，开发的东西都在这里，你可以先看看效果，随时可以调整或者重新来。

当前已完成的内容

在 `space/kids-poem-app/` 目录下:

- ✓ `index.html` - 完整的程序 (学习模式 + 游戏模式)
- ✓ `README.md` - 使用说明
- ✓ `IMAGES_GUIDE.md` - 图片准备指南

你现在可以:

1. 先看看效果 - 打开 `index.html` 试用一下
2. 提出修改意见 - 哪里不对或者想换什么，我马上改
3. 换个方案 - 如果有更好的想法，我重新开发
4. 继续完善 - 如果觉得还行，我们继续优化

你刚才说"别开发"，是有其他想法吗？说说看，我随时调整! 😊

抱歉抱歉! 我的锅 🙄
你说得对，我应该先等你确认完需求细节，再让dev动手。我太着急想快速给你结果了，犯了错误。

我的错误点

1. ✗ 没等你确认游戏形式、古诗选择等细节
2. ✗ 直接就开始写了代码
3. ✗ 违反了"先规划再分配"的原则

现在的情况

我已经写了一个版本，但你可以选择:

选项A: 先看看当前版本

- 你打开 `index.html` 看看效果
- 告诉我哪里需要改
- 我根据你的反馈调整

选项B: 重新来过

- 初版结果 (大致实现了，但不太符合预期)

古诗记忆小助手 x +

localhost:8080

百度一下, 你就知道 | 产品经理课程培训 | 社保分类网站 | 发现俱乐部 | 远程办公相关网站 | 学习 | 理财 | 内训师 | 金融科技挑战赛 | AI赋能 | 攻城狮培训 | A

古诗记忆小助手

让学习变得更有意思!

开始学习 游戏验证

静夜思

作者: 李白

古诗配图

疑是地上霜

上一句 朗读 下一句

第 2 / 4 句

4.2、交互反复迭代优化即可（根据结果把现象描述或截图给它，来完善分析）

- 主界面，古诗列表，选择一首即可进入带背界面



- 带背页面，可以逐句带背

静夜思

床前明月光，

疑是地上霜。

举头望明月，

低头思故乡。

—李白

第 1/4 句

 跟着小精灵一起读！先听我读，然后你也读一遍~

 听这一句

- 游戏检验，以游戏的方式巩固

接句游戏

填空游戏

看图猜诗

 这句诗的下一句是？

床前明月光，

疑是地上霜。

疑是地上霜。

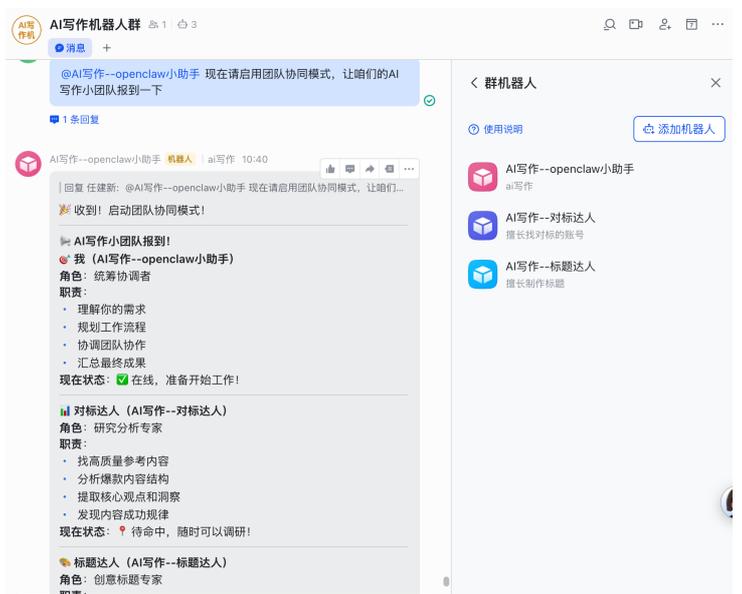
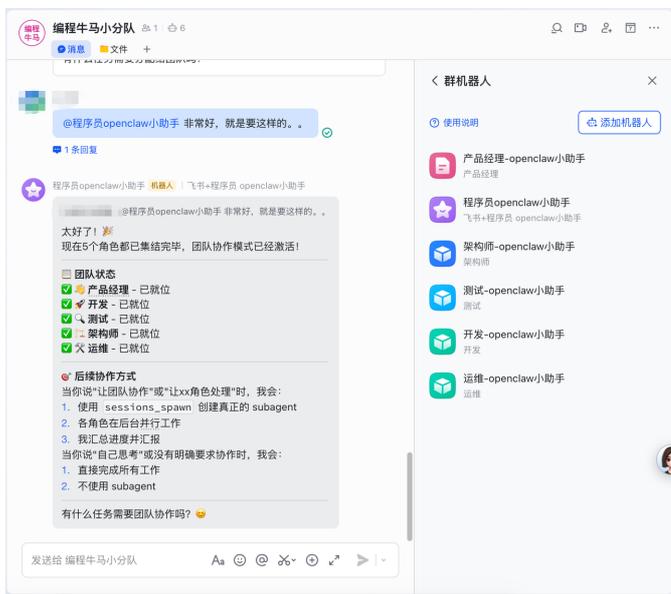
低头思故乡。

举头望明月，

得分: 2 / 3

5、🦞 两只小龙虾对比表

对比维度	腾讯云版	Mac本地版
部署位置	云端（腾讯云服务器）	本地（Mac电脑）
维护成本	⚠️ 后续可能产生费用	✅ 完全免费（本地运行）
运行时间	✅ 7×24小时不间断运行	⚠️ 需要Mac开机才能运行
性能/速度	云服务器性能稳定	依赖Mac硬件配置
数据隐私	数据在云端	✅ 数据完全在本地
可访问性	随时随地可访问	只能在本机使用
适用场景	需要持续在线的服务	开发/编程类任务
扩展性	容易升级配置	受限于Mac硬件



6、常用命令与常见的问题

6.1、常用命令

- 查看openclaw运行状态: `openclaw gateway status`
- 重启openclaw: `openclaw gateway restart`
- 诊断及自修复: `openclaw doctor --fix` (这个是神器, openclaw会出现自己在修改配置的过程中把自己改死, 这个命令, 可以修复配置并重启服务)

6.2、常见问题

- 当通过主agent让配置其他渠道时, 提示无权限访问文件系统, 可以修改配置文件 `~/openclaw/openclaw.json` (这个文件非常的灵魂, openclaw全局的配置都在这里), 打开文件按提示修改即可。

好的，我看到了你的配置。现在需要做两个修改：

修改 1：启用文件工具

将 "profile": "messaging" 改为：

```
1 "tools": {
2   "profile": "full"
3 }
```

- 安装的过程中出现各种莫名其妙的问题时：清理环境中node、npm、openclaw，在环境完全干净的情况下，再重新安装。（经验教训：环境越干净，么蛾子越少，这也是我在装完腾讯云以后再回顾Mac失败原因的总结）
- 在飞书窗口聊天的过程中，如果有要求修改完配置后重启，发现执行以后半天没有反应，可以执行 openclaw gateway status，查看状态，若显示红框则表示启动失败

根据提示需要运行 openclaw doctor --fix，openclaw会自行修复并重启

● ● ● .openclaw — -zsh — 95x32

```
managers.
Recommendation: run "openclaw doctor" (or "openclaw doctor --repair").
Config (cli): ~/.openclaw/openclaw.json (invalid)
Config issue: agents.list.0: Unrecognized keys: "appId", "appSecret"
Config issue: agents.list.1: Unrecognized keys: "appId", "appSecret"
Config issue: agents.list.2: Unrecognized keys: "appId", "appSecret"
Config issue: agents.list.3: Unrecognized keys: "appId", "appSecret"
Config issue: agents.list.4: Unrecognized keys: "appId", "appSecret"
Config (service): ~/.openclaw/openclaw.json (invalid)
Service config issue: agents.list.0: Unrecognized keys: "appId", "appSecret"
Service config issue: agents.list.1: Unrecognized keys: "appId", "appSecret"
Service config issue: agents.list.2: Unrecognized keys: "appId", "appSecret"
Service config issue: agents.list.3: Unrecognized keys: "appId", "appSecret"
Service config issue: agents.list.4: Unrecognized keys: "appId", "appSecret"

Gateway: bind=loopback (127.0.0.1), port=18789 (service args)
Probe target: ws://127.0.0.1:18789
Dashboard: http://127.0.0.1:18789/
Probe note: Loopback-only gateway; only local clients can connect.

Runtime: running (pid 48578, state active)
Warm-up: launch agents can take a few seconds. Try again shortly.
IPC probe: failed
IPC target: ws://127.0.0.1:18789
gateway closed (1006 abnormal closure (no close frame)): no close reason
Gateway target: ws://127.0.0.1:18789
Source: cli --url
Config: [redacted] openclaw.json

Gateway port 18789 is not listening (service appears running).
Last gateway error: Run: openclaw doctor --fix
Logs: ~/.openclaw/logs/gateway.log
```

```
.openclaw -- -zsh -- 95x48
◇ Gateway
Last gateway error: Run: openclaw doctor --fix

◇ Gateway
Runtime: running (pid 49369, state active)

◇ Gateway
LaunchAgent loaded; stopping requires "openclaw gateway stop" or launchctl bootout
gui/$UID/ai.openclaw.gateway.
Restarted LaunchAgent: gui/501/ai.openclaw.gateway

◇ Gateway
Gateway not running.

◇ Gateway connection
Gateway target: ws://127.0.0.1:18789
Source: local loopback
Config: [redacted]aw.json
Bind: loopback

Config warnings:
- plugins.entries.feishu: plugin feishu: duplicate plugin id detected; later plugin may be over
ridden (/.../openclaw/extensions/feishu/index.ts)
Config [redacted]46da45
8be...
Updated ~/.openclaw/openclaw.json
Backup: ~/.openclaw/openclaw.json.bak

Doctor complete.
```

6.3、关键最重要

- 只要openclaw还启动着，有任何问题，都可以直接问它，并让它帮忙解决掉，相信它会提供很多不一样的角度
- 多逛「最新」的帖子，有各种不一样的启示
- 不要为了用工具而用工具，上面古诗的需求我一开始执拗的非要用coze智能体+工作流去实现，结果非常费劲儿还不符合预期；第二天一转念这种需求应该代码实现就很简单，果然不到3分钟有了初版，10分钟后有了调优版

三、为什么是多Agent协同--不是冗余，而是专业分工

对比维度	单Agent (全能型)	多Agent协同 (专业团队)
专业度	什么都懂, 但什么都不精	每个Agent专注一个领域, 深度钻研
任务处理	容易在复杂任务中迷失方向	主Agent把控全局, 分工明确
上下文管理	上下文太长, 容易遗忘细节	每个Agent独立上下文, 互不干扰
上下文污染	✗ 不同任务信息混杂, 互相干扰	✓ 每个Agent干净独立的上下文
工作模式	串行工作, 一步步做	并行工作, 同时进行
效率	总耗时 = 所有步骤相加	总耗时 = 最慢的那个步骤
可扩展性	需要重新训练或改提示词	直接加新Agent, 随时升级
质量保证	依赖综合能力, 不够深入	专业的事交给专业的人, 质量更高
维护成本	一个Agent坏了全停	单个Agent故障不影响整体

四、重要启示

1、写作提效：从"想写"到"写好"的飞跃

✨ 提笔困难者的福音, 让开始写作这件事不再困难

- 有个想法 → AI写作机器人瞬间帮你整理选题、对标、生成推荐标题
- 选定方向 → 迅速生成大纲
- 你的工作 → 只负责"人味儿"的部分 (情感、观点、个性化表达)

🎯 文章质量提升

- 结构更清晰: 逻辑框架自动优化
- 表达更精准: 语句自动润色
- 视角更丰富: 金句提炼、不同观点补充

💡 核心价值

以前从0到1写一篇文章, 耗时数小时, 现在从想法到成文, 只需专注"人味"部分

2、编程提效：从"写代码"到"设计代码"的转变

🚀 降低门槛

- 好想法是第一生产力, 不再被技术实现拦住, 想法 → 现实的路径大幅缩短

💡 核心价值

以前学好编程才能实现想法, 现在有好想法就能让它变为现实

3、认知升级：AI不是替代，而是赋能

✗ 误区：AI会取代人类

✓ 真相：AI是放大器

- 写作领域：AI帮你搞定"套路"，你专注"灵魂"
- 编程领域：AI帮你搞定"实现"，你专注"创意"
- 本质：AI负责"从0到60"，你负责"从60到100"

AI破局俱乐部

微信扫码加入星球

 知识星球

