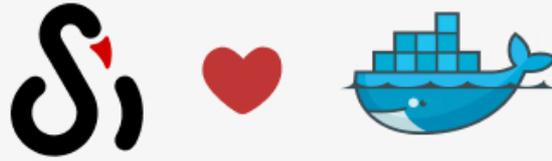


SwanLab Docker私有部署安装说明



SwanLab Docker

文档更新时间：2025年7月13日

目录

目录

SwanLab社区私有化版与团队协作版本对比说明

硬件配置与系统环境准备

安装步骤

SwanLab登录与激活

成员创建、管理与密码修改

使用私有化版本完成实验创建

升级版本

其他功能使用说明与技术支持

SwanLab社区私有化版与团队协作版本对比说明

SwanLab社区私有化版：

SwanLab团队协作/教育版：

- 仅个人学习使用
- 无项目数量限制
- 无实验规模数限制
- 无存储空间限制/请求限制
- 支持docker部署
- 支持在线/离线安装验证
- 社区技术支持
- 小规模团队多人团队协作训练
- 包含社区版在内的全部功能
- 支持http/https部署
- 首次安装部署技术支持
- 支持数据备份、预约版本升级期间技术保障
- 工作日内邮件/线上会议技术支持
- 新特性优先发布

硬件配置与系统环境准备

在安装 SwanLab 之前，请确保您的机器满足以下最低系统要求：

- CPU \geq 2核 (x86架构)
- 内存 \geq 4GB
- 存储空间 \geq 20GB
- 网络 \geq 100Mbps/s

不同机器学习任务对于服务的配置要求不同，一般情况下，大量的指标意味着较高的服务器性能要求。以大语言模型训练场景为例，我们推荐部署的服务器配置如下：

- CPU \geq 64核 (x86架构)
- 内存 \geq 128G
- 存储空间 \geq 1TB，需为固态硬盘或等效高性能网络存储，如包含媒体数据建议4TB及以上
- 网络 \geq 10Gbps/s，或等效网络配置。

此配置能保证单次实验1w+的指标记录，根据您的实验规模，您可能需要增加服务器规格来增加负载，您需要额外使用服务器负载监控工具观测服务运行状态。

除了数据库存储外，SwanLab服务的主要存储数据分为标量、日志、媒体（分别对应数字、字符串和二进制），其中媒体数据对磁盘存储空间较为敏感，在大模型训练的任务场景下，我们建议您准备4TB及以上的SSD磁盘空间供SwanLab服务使用。

此配置能保证单次实验1w+的指标记录，根据您的实验规模，您可能需要增加服务器规格来增加负载，您需要额外使用服务器负载监控工具观测服务运行状态。

如后续存在扩容和迁移问题，可联系contact@swanlab.cn

SwanLab私有化部署版，需要使用 Docker Compose 进行安装与部署，请根据你的操作系统，对下面的表格选择正确的Docker及compose版本。

操作系统	软件	解释
macOS 10.14 或更高版本	Docker Desktop	将 Docker 虚拟机 (VM) 设置为至少使用 2 个虚拟 CPU (vCPU) 和 8 GB 初始内存。否则，安装可能会失败。有关更多信息，请参阅 Mac 版 Docker Desktop 安装指南 。
Windows (启用了 WSL 2)	Docker Desktop	我们建议将源代码和其他与 Linux 容器绑定的数据存储在 Linux 文件系统中，而不是 Windows 文件系统中。有关更多信息，请参阅 Windows 上使用 WSL 安装 Linux 与 在 Windows 上使用 WSL 2 后端的 Docker Desktop 安装指南 。
Linux	Docker 19.03 或更高版本 Docker Compose 1.28 或更高版本	有关如何安装 Docker 和 Docker Compose 的更多信息，请参阅 Docker 安装指南 和 Docker Compose 安装指南 。

端口说明

如果你将 SwanLab 部署在服务器上，并希望能够远程访问与实验记录，那么请确保服务器开放以下两个端口：

端口号	是否可配置	用途说明
8000	是	网关服务端口，可用于接收外部请求，建议在公网环境中设置为 <code>80</code>
9000	否	MinIO 签名端口，用于对象存储访问，端口固定不可修改

⚠️ 如果您有 HTTPS 等需求，我们强烈建议您为 SwanLab 服务绑定一个初始流量负载均衡器，用于 SSL/TLS 终止（并且保证 IP 地址等响应头的一致性）——SwanLab 并不内置类似的服务功能。

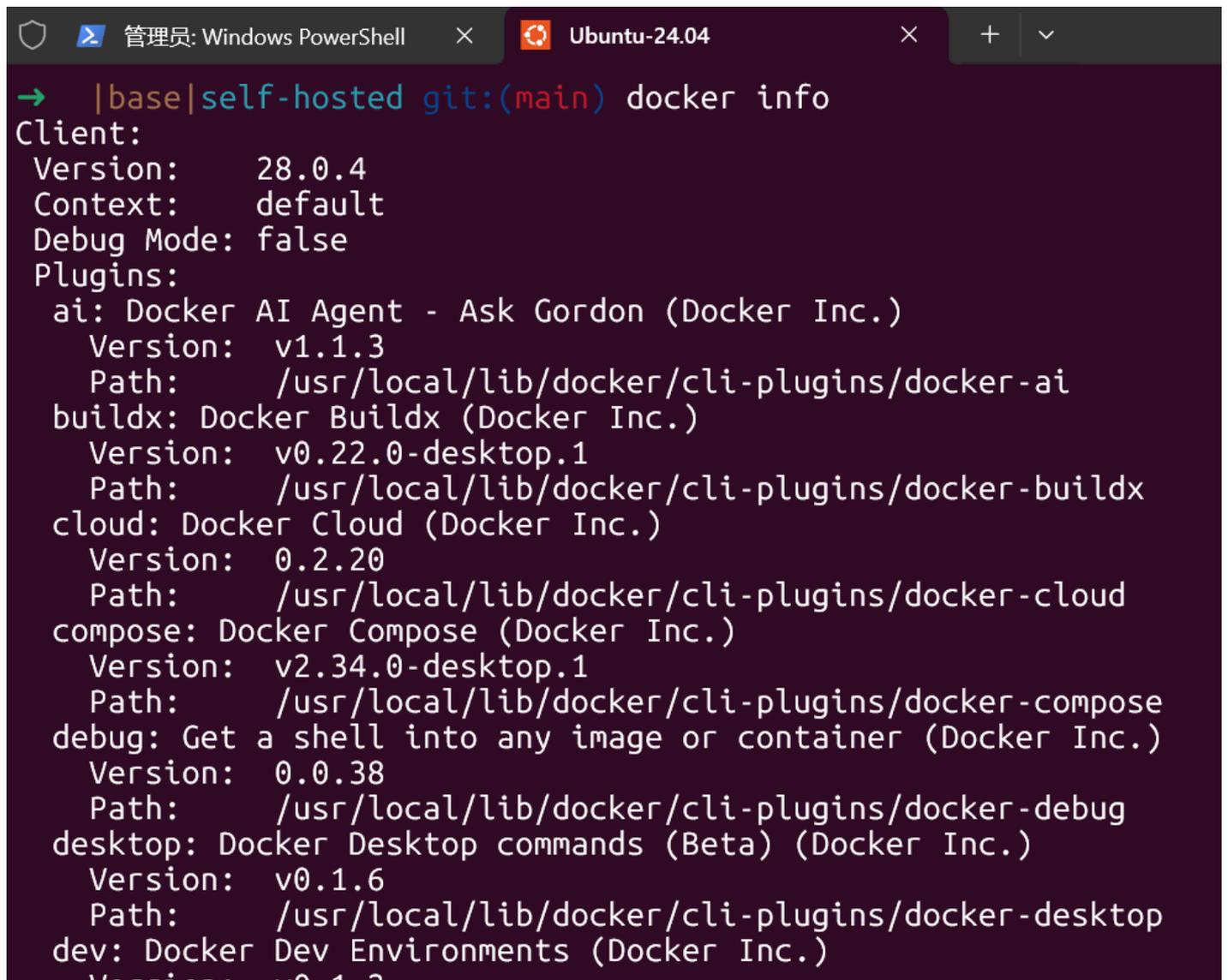
安装步骤

使用 Git 克隆 `self-hosted` 仓库：

代码块

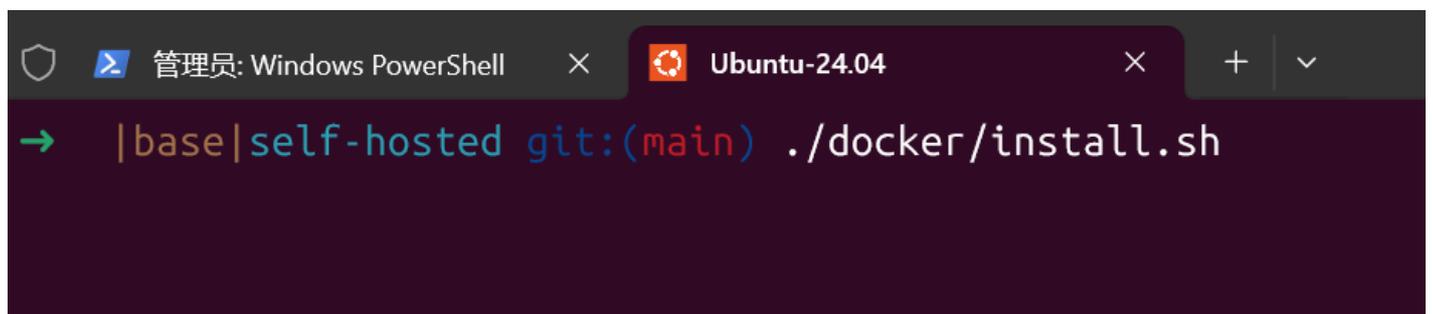
```
1 git clone https://github.com/SwanHubX/self-hosted.git cd self-hosted
```

如果你使用的是Windows系统，请确保已安装并开启 WSL2 和 Docker Desktop



```
→ |base|self-hosted git:(main) docker info
Client:
Version:      28.0.4
Context:      default
Debug Mode:   false
Plugins:
ai: Docker AI Agent - Ask Gordon (Docker Inc.)
  Version:    v1.1.3
  Path:       /usr/local/lib/docker/cli-plugins/docker-ai
buildx: Docker Buildx (Docker Inc.)
  Version:    v0.22.0-desktop.1
  Path:       /usr/local/lib/docker/cli-plugins/docker-buildx
cloud: Docker Cloud (Docker Inc.)
  Version:    0.2.20
  Path:       /usr/local/lib/docker/cli-plugins/docker-cloud
compose: Docker Compose (Docker Inc.)
  Version:    v2.34.0-desktop.1
  Path:       /usr/local/lib/docker/cli-plugins/docker-compose
debug: Get a shell into any image or container (Docker Inc.)
  Version:    0.0.38
  Path:       /usr/local/lib/docker/cli-plugins/docker-debug
desktop: Docker Desktop commands (Beta) (Docker Inc.)
  Version:    v0.1.6
  Path:       /usr/local/lib/docker/cli-plugins/docker-desktop
dev: Docker Dev Environments (Docker Inc.)
  Version:    v0.1.2
```

在WSL2的文件系统中执行 `.sh` 安装脚本



```
→ |base|self-hosted git:(main) ./docker/install.sh
```

默认的安装脚本在 `docker/install.sh`，直接执行即可一键安装所有需要的容器以及执行初始化配置。

代码块

```
1 cd ./docker./install.sh
```

默认脚本链接的镜像源在中国

如果你需要使用 [DockerHub](#) 作为镜像源，则可以使用下面的脚本进行安装：

代码块

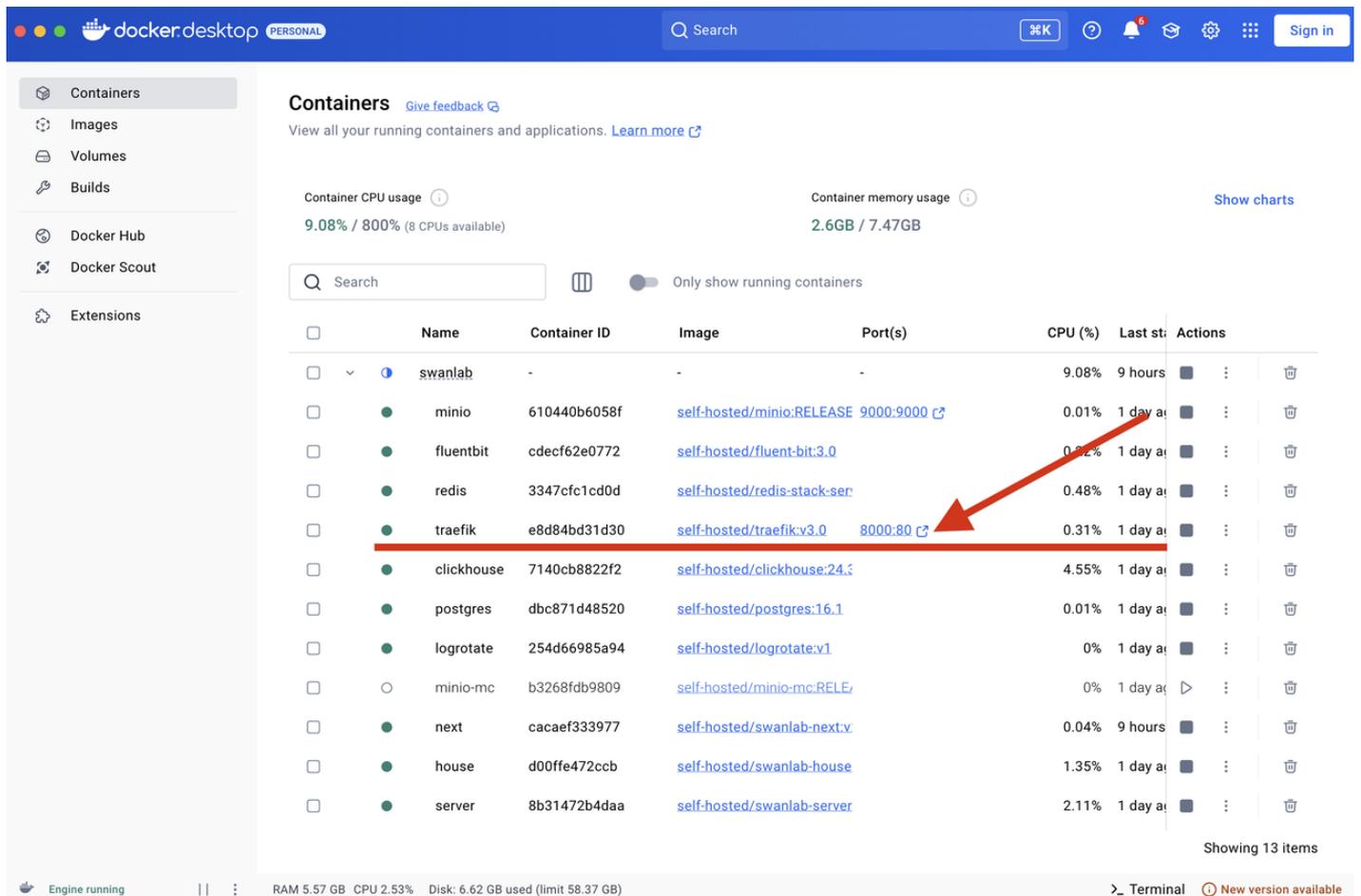
```
1 ./install-dockerhub.sh
```

SwanLab默认使用 `8000` 端口，如果你使用的是默认配置，那么可以直接访问：

<http://localhost:8000>，看到如下界面即。

在端口冲突的情况下SwanLab有可能默认部署到其他非占用端口，可在container下找到名称为swanlab的容器组，并寻找到名称为traefik的容器所映射的端口，即为主网站端口。比如

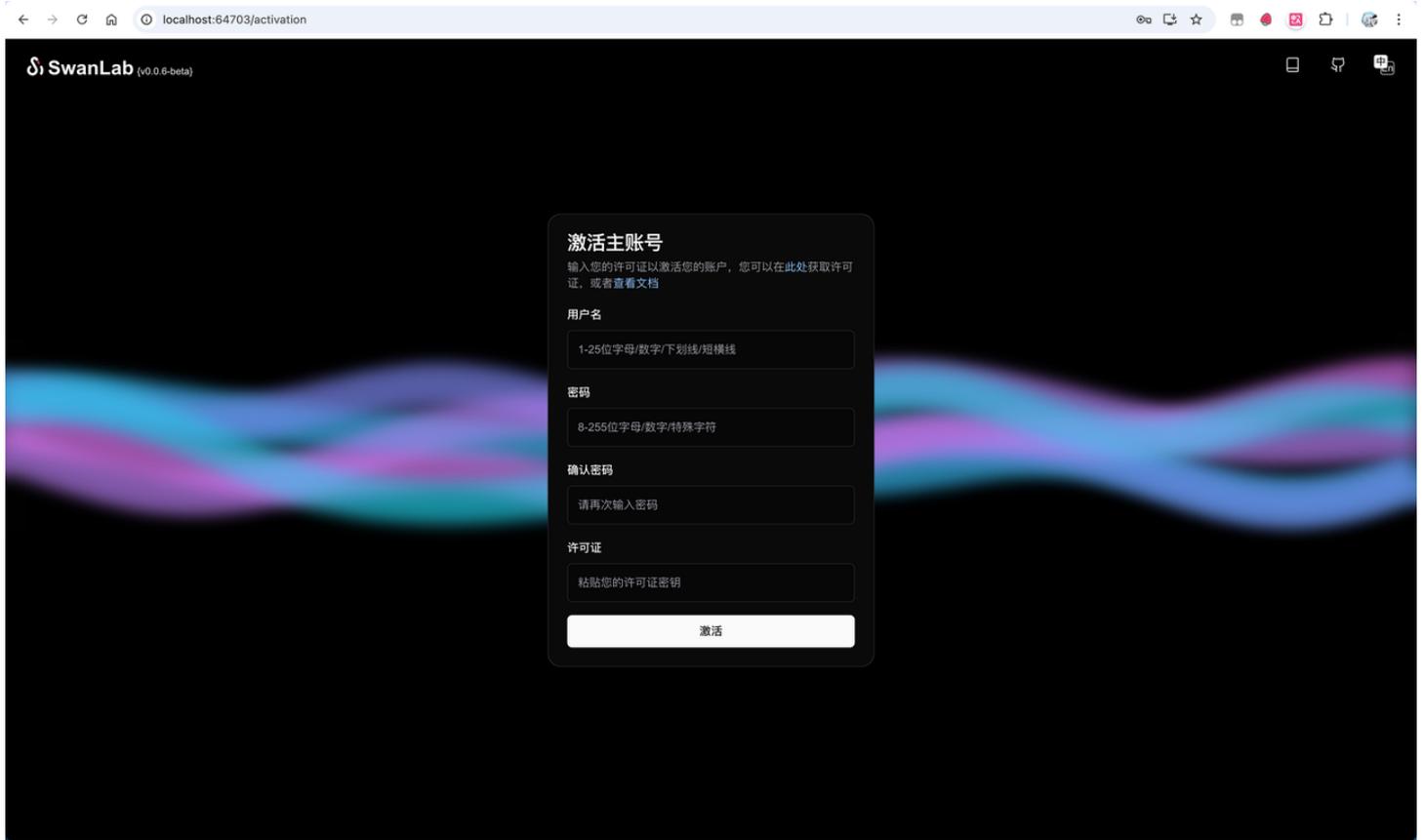
`64703:80`，那么你应该访问 <http://localhost:64703>。



The screenshot shows the Docker Desktop interface with a list of running containers. The 'traefik' container is highlighted with a red arrow pointing to its '8000:80' port mapping. The interface includes a sidebar with navigation options like Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area displays a table of containers with columns for Name, Container ID, Image, Port(s), CPU (%), Last state, and Actions. The 'traefik' container is running and has a CPU usage of 0.31% and a last state of '1 day ago'.

Name	Container ID	Image	Port(s)	CPU (%)	Last state	Actions
swanlab	-	-	-	9.08%	9 hours	Stop, Restart, Refresh, Delete
minio	610440b6058f	self-hosted/minio:RELEASE	9000:9000	0.01%	1 day ago	Stop, Restart, Refresh, Delete
fluentbit	cdecf62e0772	self-hosted/fluent-bit:3.0		0.02%	1 day ago	Stop, Restart, Refresh, Delete
redis	3347cfc1cd0d	self-hosted/redis-stack-ser		0.48%	1 day ago	Stop, Restart, Refresh, Delete
traefik	e8d84bd31d30	self-hosted/traefikv3.0	8000:80	0.31%	1 day ago	Stop, Restart, Refresh, Delete
clickhouse	7140cb8822f2	self-hosted/clickhouse:24.5		4.55%	1 day ago	Stop, Restart, Refresh, Delete
postgres	dbc871d48520	self-hosted/postgres:16.1		0.01%	1 day ago	Stop, Restart, Refresh, Delete
logrotate	254d66985a94	self-hosted/logrotate:v1		0%	1 day ago	Stop, Restart, Refresh, Delete
minio-mc	b3268fdb9809	self-hosted/minio-mc:RELE		0%	1 day ago	Stop, Restart, Refresh, Delete
next	cacaef333977	self-hosted/swanlab-next:v		0.04%	9 hours	Stop, Restart, Refresh, Delete
house	d00ffe472ccb	self-hosted/swanlab-house		1.35%	1 day ago	Stop, Restart, Refresh, Delete
server	8b31472b4daa	self-hosted/swanlab-server		2.11%	1 day ago	Stop, Restart, Refresh, Delete

默认情况下使用浏览器访问 <http://localhost:8000>，看到如下网页即代表swanlab私有化部署版成功部署！🎉

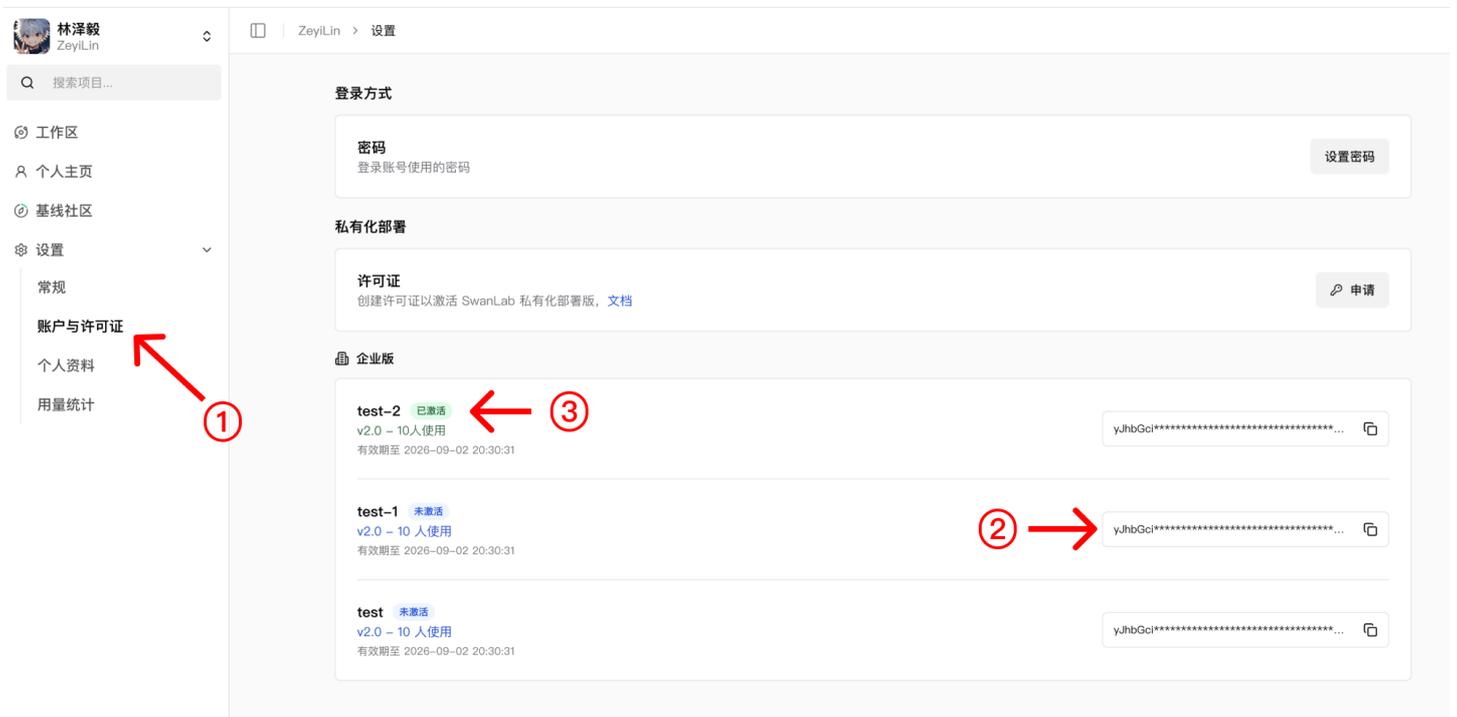


SwanLab登录与激活

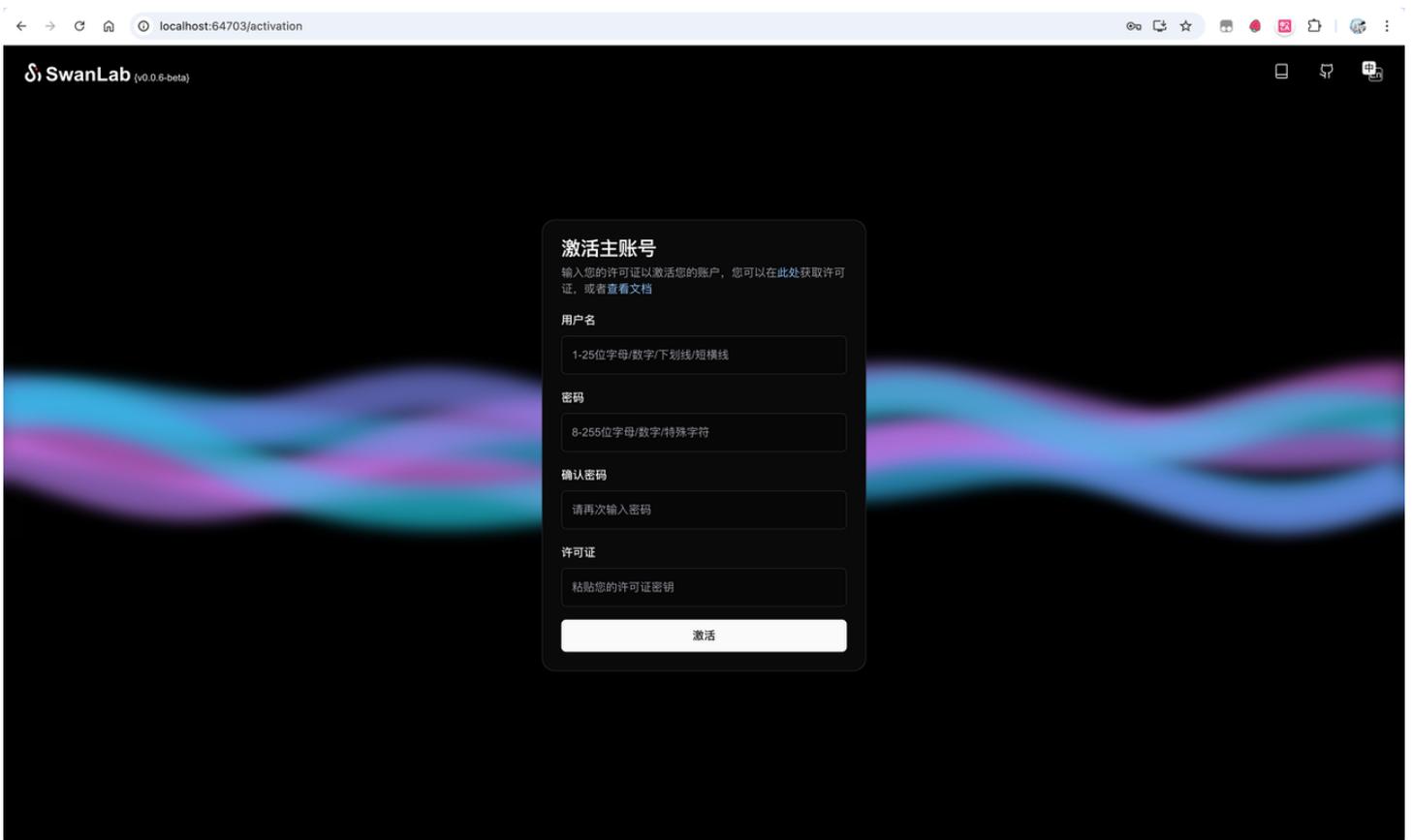
购买SwanLab协作版/教育版的客户可在[购买邮件或者指定SwanLab官网账号](#)下找到SwanLab私有化部署license。如未找到swanlab私有部署license，可通过邮箱contact@swanlab.cn联系官方重新获取。

下面列举SwanLab官方账号下获取私有部署license的办法

1. 在联网环境下访问<https://swanlab.cn>并登录，在左边栏“设置”中点击账号与许可证（箭头1）
2. 寻找到未使用的swanlab私有化部署许可，并复制对应许可证激活码（箭头2）
3. 注：完成使用的许可样式为（箭头3），并可在下方查询到许可证过期时间。



打开私有部署的SwanLab协作版主页（默认在<http://swanlab.cn>），根据引导填入管理员用户名、密码、以及对应的许可证激活码。



完成激活后将进入到工作区。

chenshaohon
chenshaohon

chenshaohon > 工作区

搜索项目...

工作区

个人主页

控制面板

设置

按框架

- Python / PyTorch
- PyTorch Lightning
- HuggingFace
- Ultralytics
- Swift
- Stable Baseline3
- 更多框架

按任务

- 入门
- 图像分类
- 目标检测
- 医学影像分割
- 大模型微调
- 文本分类
- 量化交易
- 强化学习

用SwanLab跟踪你的第一次训练

选择任务或训练框架，按照说明集成SwanLab到你的代码中，或者下载右侧的现成案例。如果你想迁移来自Tensorboard、Wandb或MLFlow上的项目，可查看[迁移指南](#)。

Python / PyTorch

- 在命令行，使用pip安装swanlab

```
pip install swanlab
```

- 登录到swanlab，并在提示时粘贴你的API Key

```
swanlab login --host http://localhost:8000
```

你的API Key: DpAUOSJ* *****

- 运行下面的Python脚本，开启第一次训练

```
import swanlab
import random

# 创建一个SwanLab项目
swanlab.init(
    # 设置项目名称
    project="my-awesome-project",

    # 设置超参数
    config={
        "learning_rate": 0.02,
        "architecture": "CNN",
        "dataset": "CIFAR-100",
        "epochs": 10
    }
)

# 模拟一次训练
epochs = 10
offset = random.random() / 5
for epoch in range(2, epochs):
    acc = 1 - 2 ** -epoch - random.random() / epoch - offset
    loss = 2 ** -epoch + random.random() / epoch + offset

    # 记录训练指标
    swanlab.log({"acc": acc, "loss": loss})

# [可选] 完成训练，这在notebook环境中是必要的
swanlab.finish()
```

创建新项目 +

下载案例

下载示例文件并运行脚本。查看SwanLab的运行情况。

first_example.py

或尝试Google Colab版：

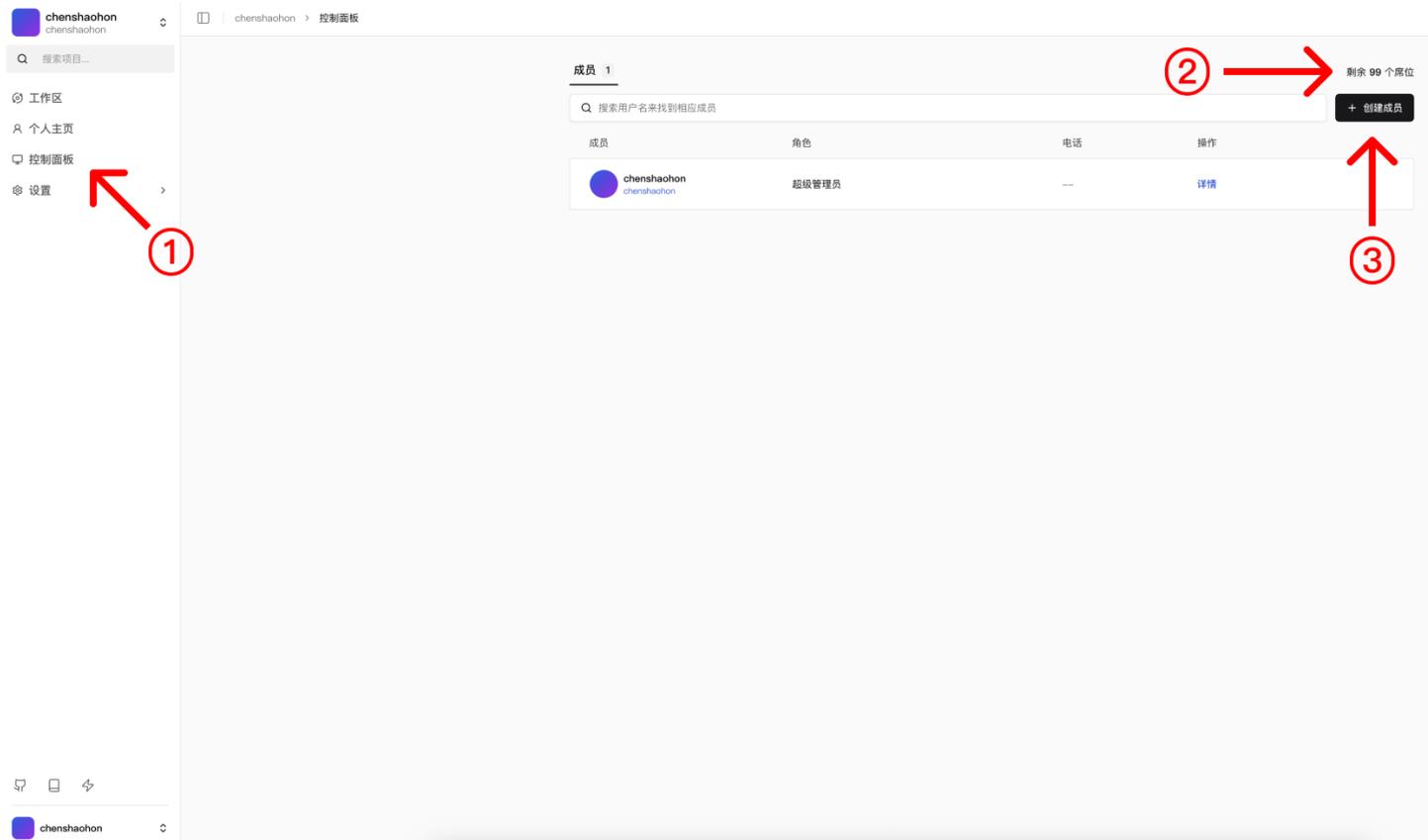
在Google Colab打开

你的API Key

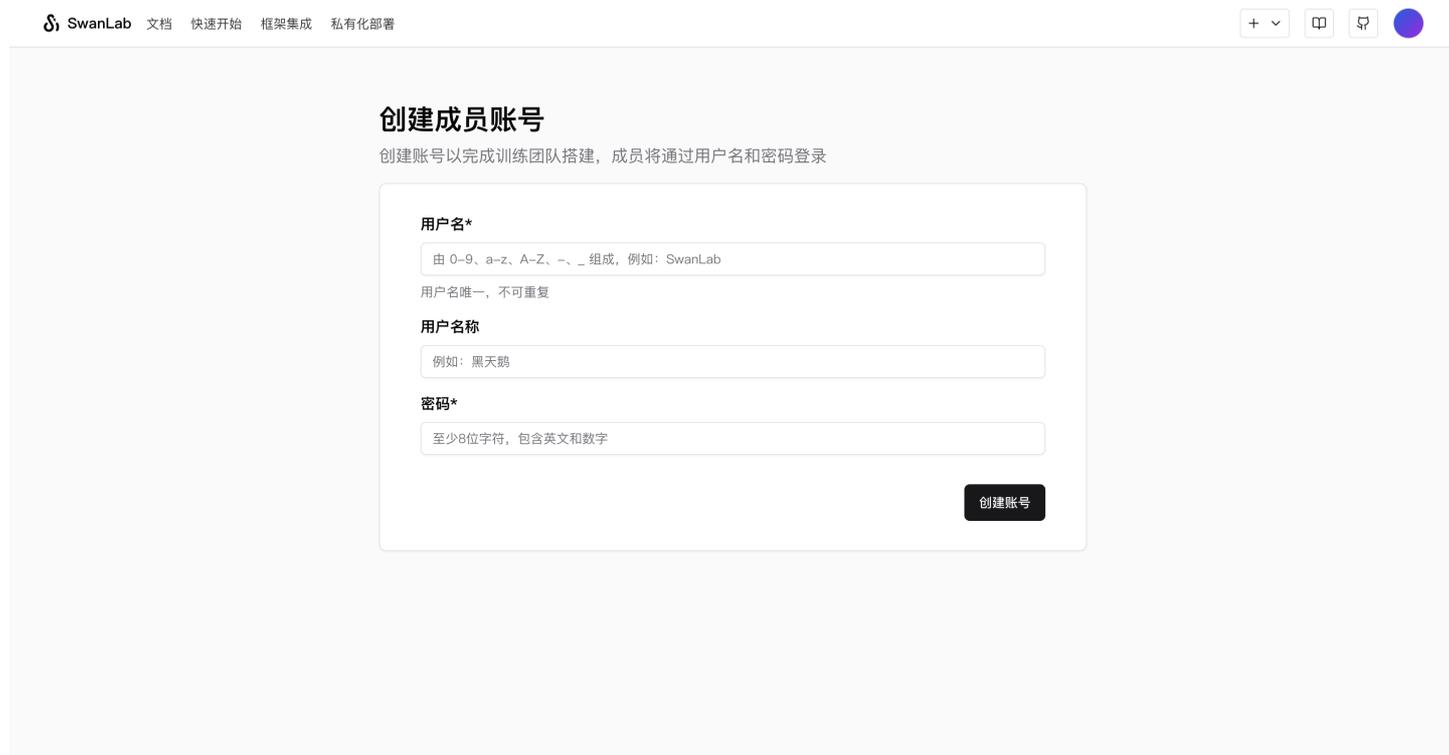
DpAUOSJ* *****

成员创建、管理与密码修改

1. 点击左边栏控制面板（箭头1）
2. 可在右上角处查看到当前剩余协作席位（箭头2）
3. 点击右上角添加新成员账号（箭头3）

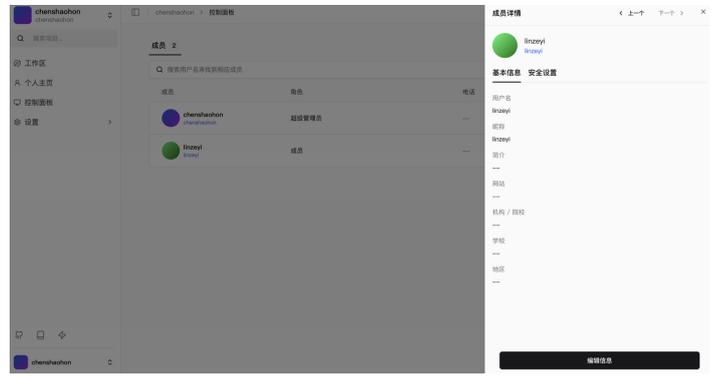
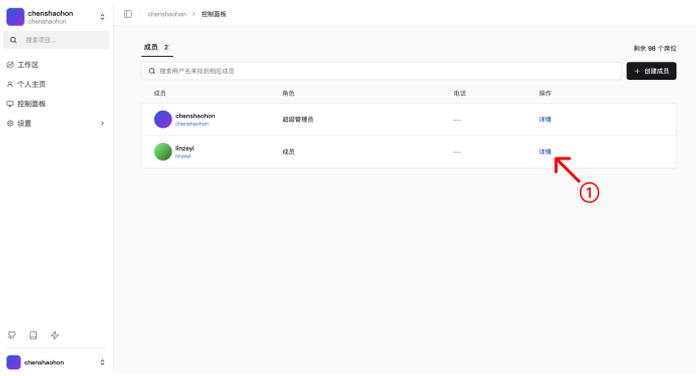


输入成员账号名称、用户名称、密码等信息。



完成创建后可在控制面板查看新创建的协作成员

点击详情即可进一步对该账号身份信息进行修改、以及完成修改登录密码等操作。



使用私有化版本完成实验创建

在Python中使用pip安装SwanLab SDK:

代码块

```
1 # 国内源安装
2 pip install -i https://mirrors.cernet.edu.cn/pypi/web/simple swanlab
```

⚠ 建议使用最新版swanlab安装包，如遇技术问题请联系contact@swanlab.cn

在Python SDK完成登录，注意此处需要使用`--host`命令指向部署服务器IP地址

代码块

```
1 swanlab login --host <IP地址>
```

如果你之前登录过swanlab，想要重新登录，请使用：`swanlab login --host <IP地址> --relogin`。

按回车，填写API Key，完成登录。之后你的SwanLab实验将会默认传到私有化部署的SwanLab上。

测试脚本:

代码块

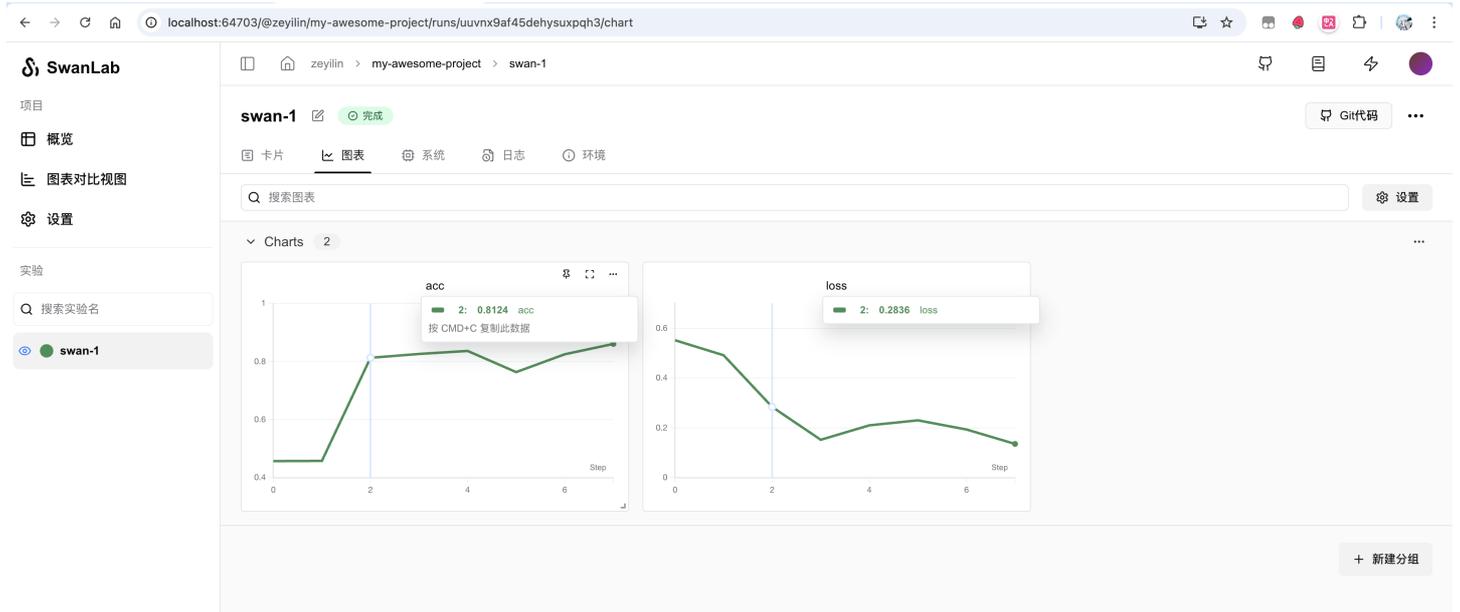
```
1 import swanlab
2 import random
3
4 # 创建一个SwanLab项目
5 swanlab.init(
6     # 设置项目名
7     project="my-awesome-project",
8
```

```

9     # 设置超参数
10    config={
11        "learning_rate": 0.02,
12        "architecture": "CNN",
13        "dataset": "CIFAR-100",
14        "epochs": 10
15    }
16 )
17
18 # 模拟一次训练
19 epochs = 10
20 offset = random.random() / 5
21 for epoch in range(2, epochs):
22     acc = 1 - 2 ** -epoch - random.random() / epoch - offset
23     loss = 2 ** -epoch + random.random() / epoch + offset
24
25     # 记录训练指标
26     swanlab.log({"acc": acc, "loss": loss})
27
28 # [可选] 完成训练, 这在notebook环境中是必要的
29 swanlab.finish()

```

运行后在网页查看实验：



升级版本

如果你想要将你本地的私有化部署版本升级到最新版，请使用下面的命令：

代码块 在你之前本地部署的 self-hosted 项目目录下cd ./docker./upgrade.sh

升级完成的命令行样式:

```
(base) → docker git:(main) ./upgrade.sh
Updating the container version will restart docker compose. Do you agree? [y/N] y
begin update
[+] Pulling 42/42
✓ fluent-bit Pulled
✓ clickhouse Pulled
✓ redis Pulled
✓ swanlab-cloud Pulled
✓ swanlab-house Pulled
✓ swanlab-server Pulled
✓ logrotate Pulled
✓ swanlab-next Pulled
✓ minio Pulled
✓ create-buckets Pulled
✓ postgres Pulled
✓ traefik Pulled

[+] Running 12/12
✓ Container swanlab-postgres Healthy
✓ Container swanlab-redis Healthy
✓ Container swanlab-logrotate Running
✓ Container swanlab-clickhouse Healthy
✓ Container swanlab-traefik Running
✓ Container swanlab-minio Healthy
✓ Container swanlab-fluentbit Running
✓ Container swanlab-minio-mc Started
✓ Container swanlab-house Started
✓ Container swanlab-cloud Started
✓ Container swanlab-next Started
✓ Container swanlab-server Healthy
finish update
```

其他功能使用说明与技术支持

有关SwanLab使用说明请参考:

- <https://docs.swanlab.cn/>

如需技术支持, 请联系:

- contact@swanlab.cn