



链滴

使用群晖部署思源笔记的注意事项

作者: [kstephenlgz](#)

原文链接: <https://ld246.com/article/1637487161467>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

首先感谢 V大 及 D大 开发出如此好用的个人笔记管理系统

D大 | <https://github.com/88250>

V大 | <https://github.com/Vanessa219>

一.前言

- 群晖目前官方及社区还没有开发出思源笔记的应用套件，目前在群晖上部署思源笔记仅能使用docker部署
- 一般群晖的docker应用部署使用web界面进行，方便小白使用。具体可参考以下前人的教程，我也在以下<https://ld246.com/article/1637245472220>
- 但对shell脚本较为熟悉的老鸟，使用脚本部署不失为一种高效的方法，本文目的就在于此

二.用指令部署基于docker的思源笔记应用

使用指令部署意味着需要会使用基本的Linux命令，有一定的docker使用基础

□v1.5.0 及更老的版本

```
docker run --name siyuannotes -dit --restart=always -v /volume1/Notes/SiYuan:/root/Documents/SiYuan -p 6806:6806 b3log/siyuan:v1.5.0 -accessAuthCode YOUR_PASSWORD
```

- **YOUR_PASSWORD**为鉴权密钥，每次登陆思源笔记的web地址时需要输入
- **支持部署在任意文件夹** 意味着可以在群晖中新建一个共享文件夹专用与思源笔记。如上方宿主机录路径：/volume1/Notes/SiYuan

□sparkles v1.5.1及更新的版本

- **仅支持部署在群晖的/volume1/docker/目录或其子目录路径下**

step1: 新建带有1000用户权限的文件夹

从v1.5.1开始，使用uid及gid 1000:1000 代替root用户，所以在宿主机创建工作空间文件夹时 请注意设置该文件夹所属用户组：

```
root@MyNas:/volume1/docker**# mkdir SiYuan
root@MyNas:/volume1/docker**# chown -R 1000:1000 /volume1/docker/SiYuan
```

检查文件夹的用户权限是否1000：

```
root@MyNas:/volume1/docker**# ls -l
total 0
drwxrwxrwx+ 1 Handsome users 30 Oct 23 13:15 baidu-pcs
drwxrwxrwx+ 1 1000 1000 0 Nov 21 16:03 SiYuan
```

在终端中看到1000 1000 的文件夹SiYuan,表示需要映射到宿主机的文件夹配置成功。下一步, 配置器, 注意启动容器时需增加参数** -u 1000:1000。

step2: 指令部署docker 容器

```
docker run -u 1000:1000 -dit --name siyuannotes --restart=always -v /volume1/docker/SiYuan:/SiYuan/ -p 6806:6806 b3log/siyuan:v1.5.3 -resident -workspace /SiYuan -accessAuthCode YUR_PASSWORD
```

- **YOUR_PASSWORD**为鉴权密钥, 每次登陆思源笔记的web地址时需要输入
- **-v** 参数后注意 宿主机及容器两个目录路径最后"/" 的数量不相同
- 内核参数 **-resident** 表明容器需要常驻内存。思源笔记用户指南 - 0 请从这里开始 - 通用操作 - 内核参数 内是用 "--" 的, 与此处不相同。
- 宿主机6806端口不可变
- **--name** 参数是给容器起的名字, 最好别用大写字母
- **-dit** 表示关闭shell窗口后, 思源笔记依然能继续运行

以上经验均在v1.5.1版本后摸索出来。最开始是容器部署后马上报错, 后来是容器能部署但无法映射记文件到宿主机, 最后爬帖子, 跟着<https://ld246.com/article/1637245472220>用web部署并映射功, 最后整理出以上文章, 希望能帮到所有群晖用户。

另外有两点需要提醒:

- 升级思源笔记流程: 关闭容器 => 备份数据 => 升级 => 还原数据 => 启动容器
- 定期备份数据: 关闭容器 => 备份数据 => 启动容器