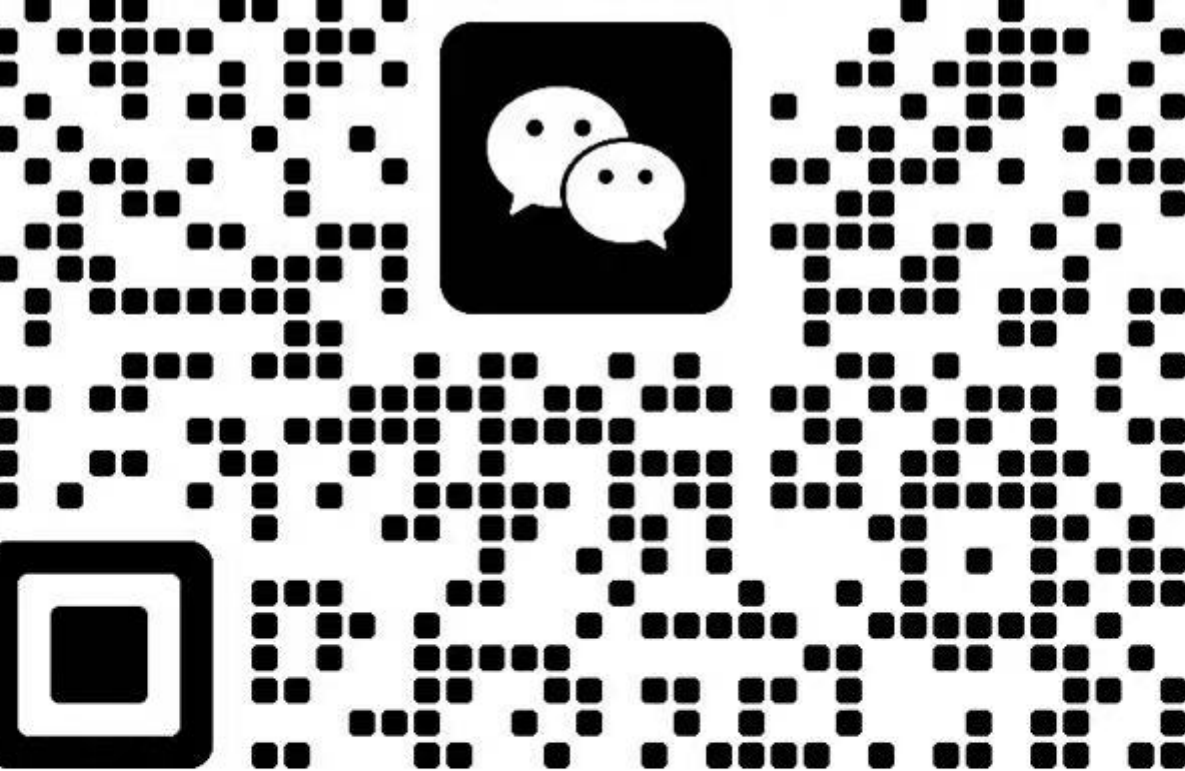




扫二维码，添加我为朋友。

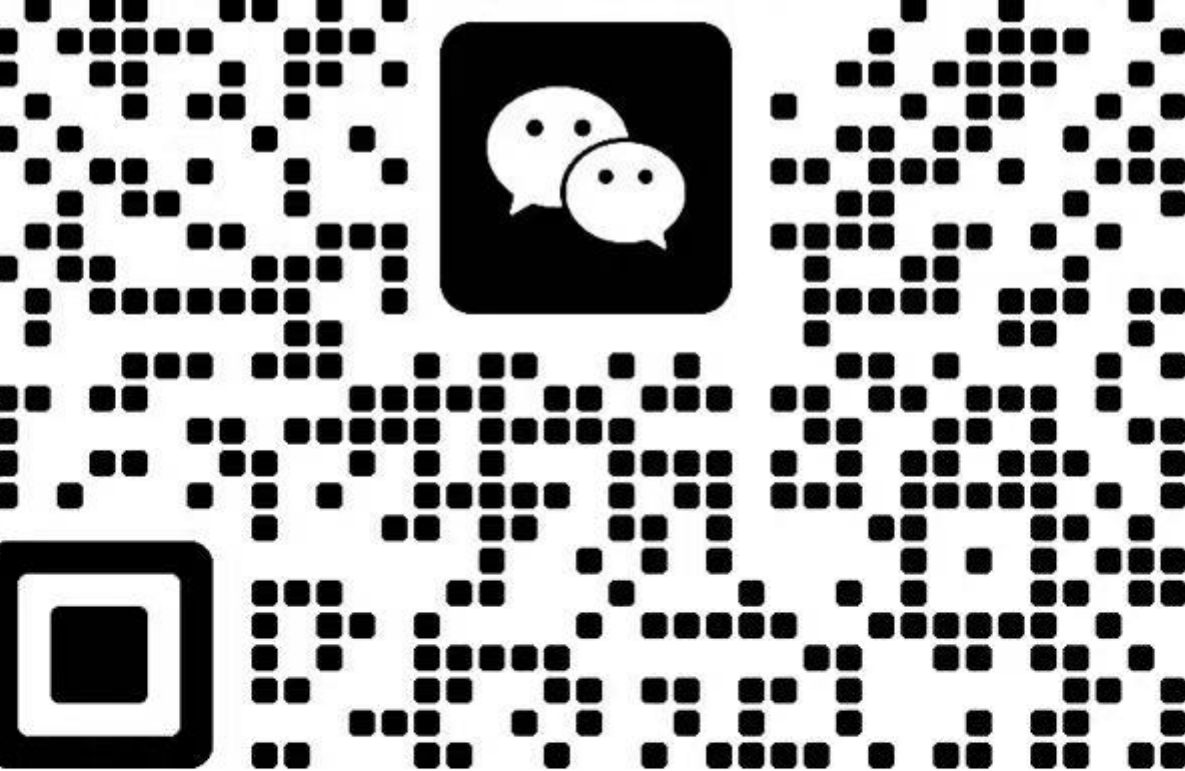
- 机器已有 [.NET 8 Desktop Runtime](#)：可用更小的框架依赖版

2. 运行

1. 解压到任意目录（建议路径不要带中文空格）
2. 双击 `ProxyNodeHub.exe`
3. 按 **F5**，或点「搜索并分析」

3. 常用快捷键

键	作用
F5	开始 / 取消搜索
Ctrl+C	复制最佳订阅链接
Ctrl+L	详情面板
Ctrl+D	收藏夹
Ctrl+F	筛选框
Esc	关弹窗



扫二维码，添加我为朋友。

amd64。

1. 准备目录与管理员密码（只做一次）

把下面路径换成你自己的（示例沿用 README）：

```
mkdir -p /mnt/usb1-1/proxynodehub/data /mnt/usb1-1/proxynodehub/secrets
test -e /mnt/usb1-1/proxynodehub/secrets/admin-password || \
    (umask 077; openssl rand -base64 -out /mnt/usb1-1/proxynodehub/secrets/admin-password 32)
chown -R 1654:1654 /mnt/usb1-1/proxynodehub/data /mnt/usb1-1/proxynodehub/secrets
chmod 750 /mnt/usb1-1/proxynodehub/data /mnt/usb1-1/proxynodehub/secrets
chmod 600 /mnt/usb1-1/proxynodehub/secrets/admin-password
```

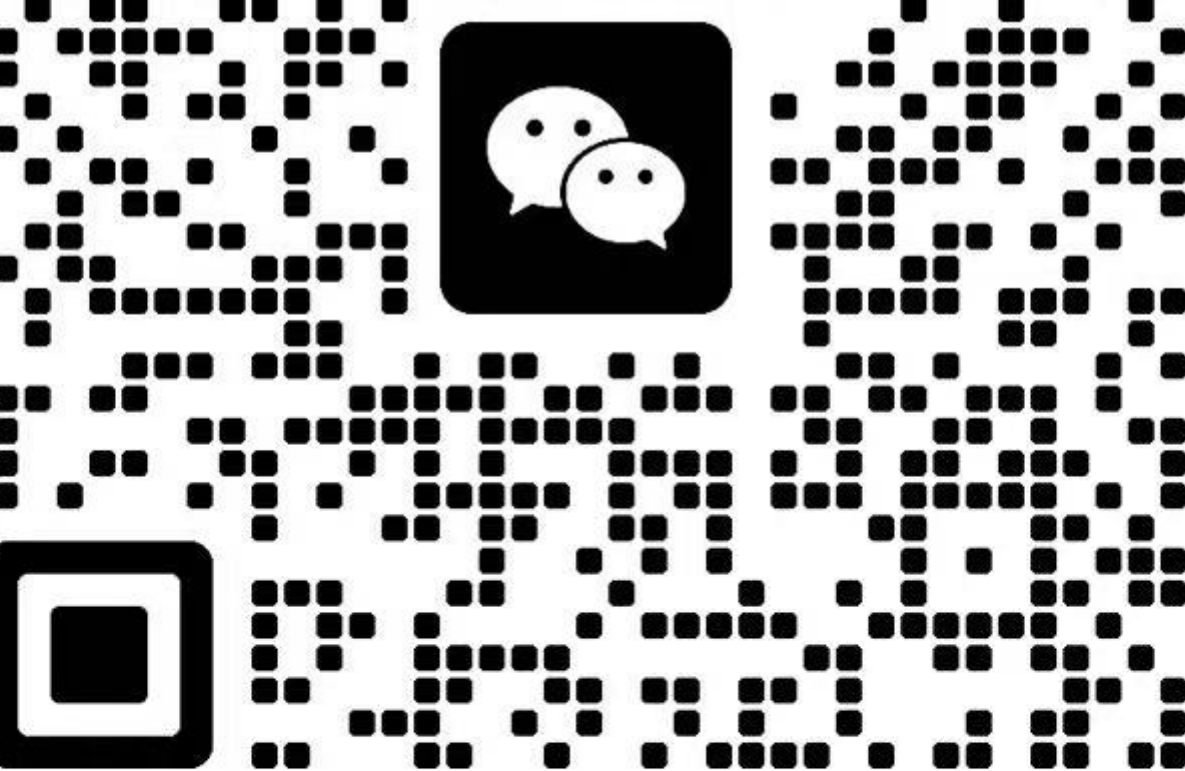
密码只在服务器上看，不要发到聊天里。

2. 启动容器

先拉镜像：

```
docker pull helloworldz1024/proxynodehub:latest
```

生产建议用 `sha-<完整提交>` 钉死版本，不要只信 `latest`。



扫二维码，添加我为朋友。

1. **发现仓库**：点搜索/分析，看评分与候选；可筛选、多选、导出、批量收藏
2. **我的收藏**：长期跟的仓库
3. **订阅检测**：接上外部 **subs-check** 后才有速度/媒体结果（可选）
4. **运行记录**：每轮任务日志、清理搜索记忆
5. **学习记录**：特征库学到的路径
6. **服务设置**：Token、定时（1-24 小时）、候选数、检测器连接等

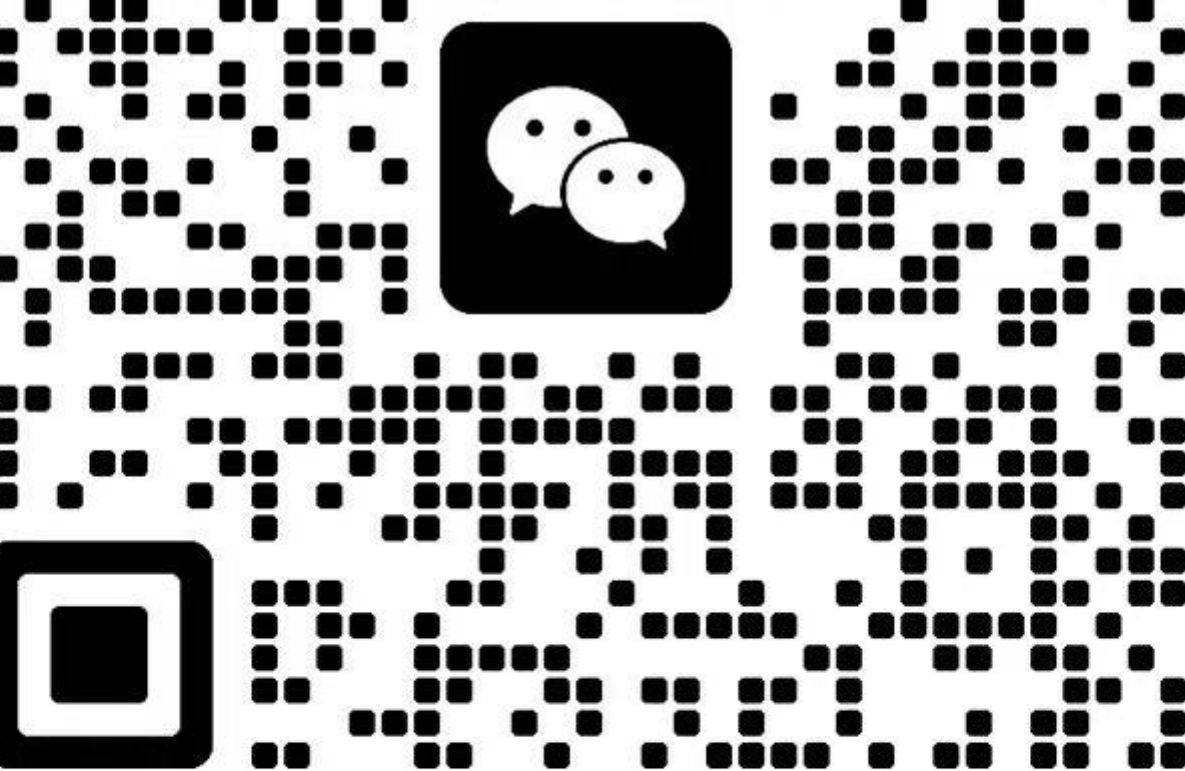
4. 重要接口含义（别搞混）

地址	含义
<code>/live</code>	进程活着（Docker healthcheck）
<code>/health</code>	有约 25 小时内有效结果，否则 503
<code>/subscriptions.txt</code>	来源订阅 URL 清单，不是测速后的节点包

首次成功发现前，`/subscriptions.txt` 可能 503；失败时尽量保留上一份有效快照。

5. 可选：接 subs-check

在已有检测器配置里追加（别整文件覆盖）：

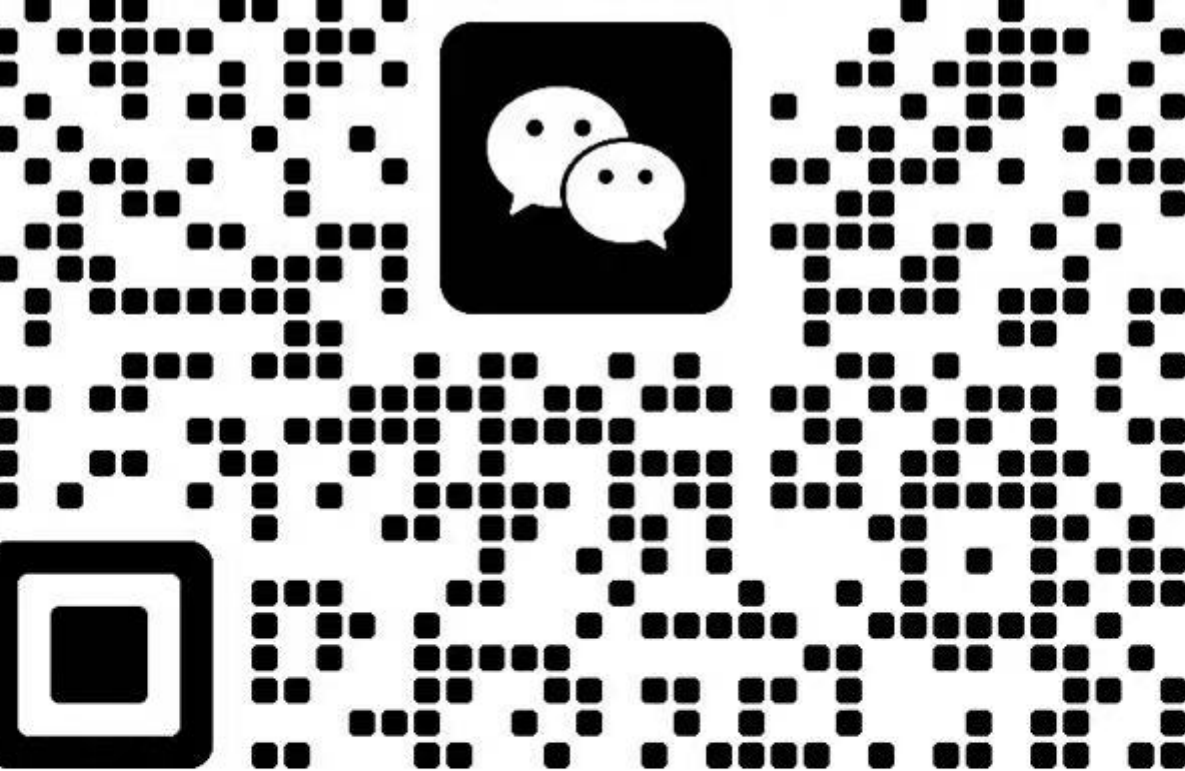


扫二维码，添加我为朋友。

四、配置速查（Web 环境变量）

变量	说明
ADMIN_PASSWORD / ADMIN_PASSWORD_FILE	必填，二选一
GITHUB_TOKEN	可选；也可在页面覆盖
REFRESH_HOURS	首次默认 6；之后以页面 <code>state.json</code> 为准
REPO_COUNT	首次默认分析 10 个候选
INACTIVE_DAYS	首次默认只看近 7 天有更新的
AUTO_REFRESH	是否后台自动发现
SUBSCHECK_*	可选检测器连接

调度类变量只初始化新数据卷；改完页面设置后以 `/data/state.json` 为准。备份请整卷备份 `/data`（含密钥环）；丢密钥环会解不开在线连接配置。



扫二维码，添加我为朋友。

```
self-contained
```

```
# 框架依赖
```

```
dotnet publish src/ProxyNodeHub.csproj -c Release -r win-x64 -p:SelfContained=false -o dist/  
framework-dependent
```

许可：MIT。作者页面见仓库说明。

说明根据仓库 README 整理；镜像名、端口、路径以你实际部署为准。