

DeepSeek Harness

Ubuntu 24.04 npm Node

- **DeepSeek Harness** `dsh` DeepSeek agent “ ” Cordis [deepseek-ai/deepseek-harness](#)
- `npm latest = 0.1.5-rc.1` 2026-09-10 `next` = `0.1.5-rc.2`
- `Node.js ^22.19.0 || >=24.0.0` `package.json engines` Ubuntu 24.04 **Node 24 LTS**
- `http://127.0.0.1:3080`

- **npm** + **systemd** `dsh web` nginx + HTTPS
- **Docker** `pnpm` `node-pty` `musl/glibc` `profile patch` `npm`
- `127.0.0.1` `webserver` `TLS/` `SSH` `nginx+HTTPS`

1. Ubuntu 24.04

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl ca-certificates git ufw
# root sudo
```

2. Node.js 24 LTS NodeSource

```
curl -fsSL https://deb.nodesource.com/setup_24.x | sudo -E bash -
sudo apt install -y nodejs
node -v    # 22.19.1 v24.x
npm -v
```

3. 创建用户

```
sudo useradd --system --create-home --shell /usr/sbin/nologin dsh
sudo mkdir -p /srv/dsh/workspace
sudo chown -R dsh:dsh /srv/dsh
# 创建 workspace 目录，用于存放 Web UI 相关文件
```

4. 安装 DeepSeek Harness

```
sudo npm install -g @deepseek-ai/dsh@0.1.5-rc.1
dsh --version    # 0.1.5-rc.1
dsh --help
```

```
“ 安装 rc.2 版本 @deepseek-ai/dsh@0.1.5-rc.2 的 CLI 工具
web 模式 运行 web 模式 --host 127.0.0.1 / 0.0.0.0 --
port 8080 --trusted-host --no-open
```

5. 配置 DeepSeek API Key

配置环境变量 → \$DSH_HOME/.credentials.yaml → 配置 .env → \$DSH_HOME/.env

\$DSH_HOME 默认 ~/.dsh Web 模式 DEEPSEEK_API_KEY 配置 systemd

配置 6 个环境变量

配置环境变量 http://127.0.0.1:3080 → 配置 → DeepSeek API Key

配置 \$DSH_HOME/.credentials.yaml 环境变量

6. systemd 配置

```
# /etc/systemd/system/dsh.service
[Unit]
Description=DeepSeek Harness Web UI
After=network-online.target
Wants=network-online.target

[Service]
```

```
Type=simple
User=dsh
Group=dsh
WorkingDirectory=/srv/dsh/workspace
Environment=DSH_HOME=/home/dsh/.dsh
Environment=DEEPSEEK_API_KEY=sk-XXXX
ExecStart=/usr/bin/dsh web --no-open --host 127.0.0.1 --port 3080
Restart=on-failure
RestartSec=5
NoNewPrivileges=true
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
sudo systemctl enable --now dsh
journalctl -u dsh -f          # "" "dsh web:" URL XXXXXX tokenXXXXXXXX
curl -sI http://127.0.0.1:3080/ | head -1  # "" HTTP/1.1 200
```

```
"" "" dsh "" /usr/bin"" which dsh "" ExecStart ""
```

7.XXX nginx XX + HTTPSXXXXXXXX

```
# /etc/nginx/sites-available/dsh
server {
    listen 80;
    server_name dsh.example.com;
    location / {
        proxy_pass http://127.0.0.1:3080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;          # WebSocket XX
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

```
sudo ln -s /etc/nginx/sites-available/dsh /etc/nginx/sites-enabled/  
sudo nginx -t && sudo systemctl reload nginx  
sudo apt install -y certbot python3-certbot-nginx  
sudo certbot --nginx -d dsh.example.com
```

```
##### WebSocket ##### → ##### --  
trusted-host dsh.example.com ##### dsh-webui-auth  
##### ssh -L 3080:127.0.0.1:3080 user@server #####
```

8. #####

```
sudo ufw allow OpenSSH && sudo ufw allow 80/tcp && sudo ufw allow 443/tcp && sudo ufw enable
```

```
##### https://dsh.example.com → ## →### Key → ##### → #####  
“Summarize this repository” → #####
```



- ## ##### \$DSH_HOME settings.yaml .credentials.yaml
 profiles/ ##### sudo npm install -g @deepseek-ai/dsh@<##> ## npm
 ##### [SAFETY.md](#)
- ## ##### URL ##### token API/WebSocket #####
 URL ##### /##### \$DSH_HOME/.credentials.yaml ##### chmod 600 0.0.0.0
 ## TLS #####
- ## ##### dsh --profile headless "#####" ##### dsh plugin --profile web add
 <#> ## pnpm sudo corepack enable && corepack prepare pnpm@latest --activate
 ##### [network-proxy](#)

```
##### pnpm##### corepack/npm ##### dsh plugin ##### --profile  
##### ---##### Codex ##### @deepseek-ai/dsh-subagent-codex  
##
```

pnpm dsh



```
sudo npm install -g pnpm@11.7.0      # 安装 pnpm 11.7.0
pnpm --version                        # 验证
```

■■■■■■■■■■

web 安装

```
sudo -u dsh /usr/bin/dsh plugin --profile web add @deepseek-ai/dsh-subagent-codex
```

■■■

- 通过 `sudo -u dsh` 配置 profile 为 `/home/dsh/.dsh/profiles/web` 并设置 `DSH_HOME`
- 使用 `--profile web` 安装 `headless`、`sdk`、`acp` 等 profile
- 通过 `pnpm add` 安装 `dsh.profile.bundles` 到 `package.json` 中，并配置 `"dsh": {"bundle": {"patch": "./cordis.patch.yml"}}`

■■■■■■■■■■

bundle 安装

```
sudo systemctl restart dsh
dsh-url      # 获取 token
```



```
sudo -u dsh /usr/bin/dsh plugin --profile web list      # 查看安装情况
sudo -u dsh env DSH_HOME=/home/dsh/.dsh /usr/bin/dsh --profile web --dump-config | grep -i subagent
```

Web UI 安装

Agent 安装

Codex 安装

Codex CLI 安装

■■

“■■■”■■■■■■■■■■

- 通过 `sudo -u dsh /usr/bin/dsh plugin --profile web remove @deepseek-ai/dsh-subagent-codex` 卸载

- `npm add /path/to/plugin` `... plugin --profile web add <npm>` `... add /path/to/plugin`
 GitHub `dsh-plugin` topic
- `npm config/` `github-review` `schedule` `mcp-memory` `cordis patch` `dsh --profile web --patch ./xx.yml` `cordis.patch.yml` `pnpm`

pnpm Codex

`$DSH_HOME/sessions/<>/` token
 Web

`DSH_HOME` + `token` + `alice`

1.

```
sudo useradd -m -s /usr/sbin/nologin alice
sudo mkdir -p /srv/dsh-alice/workspace
sudo chown -R alice:alice /srv/dsh-alice
```

2. `systemd` `DSH_HOME` `3081`

```
# /etc/systemd/system/dsh-alice.service
[Unit]
Description=DeepSeek Harness Web UI (alice)
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=alice
Group=alice
WorkingDirectory=/srv/dsh-alice/workspace
Environment=DSH_HOME=/home/alice/.dsh
Environment=DEEPSEEK_API_KEY=sk-alice
ExecStart=/usr/bin/dsh --profile web --no-open --host 127.0.0.1 --port 3081
Restart=on-failure
RestartSec=5
```

```
NoNewPrivileges=true
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload && sudo systemctl enable --now dsh-alice
```

3. nginx 配置 dsh 0.0.0.0

```
# /etc/nginx/sites-available/dsh-alice
server {
    listen 8081;                # alice 监听
    location / {
        proxy_pass http://127.0.0.1:3081;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
    }
}
```

```
sudo nginx -t && sudo systemctl reload nginx
```

4. 配置 alice token

```
sudo -u alice env DSH_HOME=/home/alice/.dsh /usr/bin/dsh-url # journalctl
```

Alice 通过 `http://192.168.4.17:8081/?token=<token>` 向 Bob 的 8082/3082 发起请求，其中 `--trusted-host 192.168.4.17` 表示信任该 IP。



- Alice 通过 `dsh-webui-auth` 获取 token/
- Alice 通过 `$DSH_HOME/sessions/...` 获取 token
- Alice 通过 `profile sessions` 获取 `$DSH_HOME`



□□□□□□ □□ /□□ /□□□□ □□□□□□□□□□□□□□

- □□□□□□□□ + □□ OS □□□ agent □□ OS □□□□□□□□□□□□□□
- □□□□ token □□□□□□□□□□ token □□□

□□□□□□□□□□

alice

 □□□□□□□□ unit□ nginx □□□ token
□□□□□□□□□□□□□□□□□□□□