

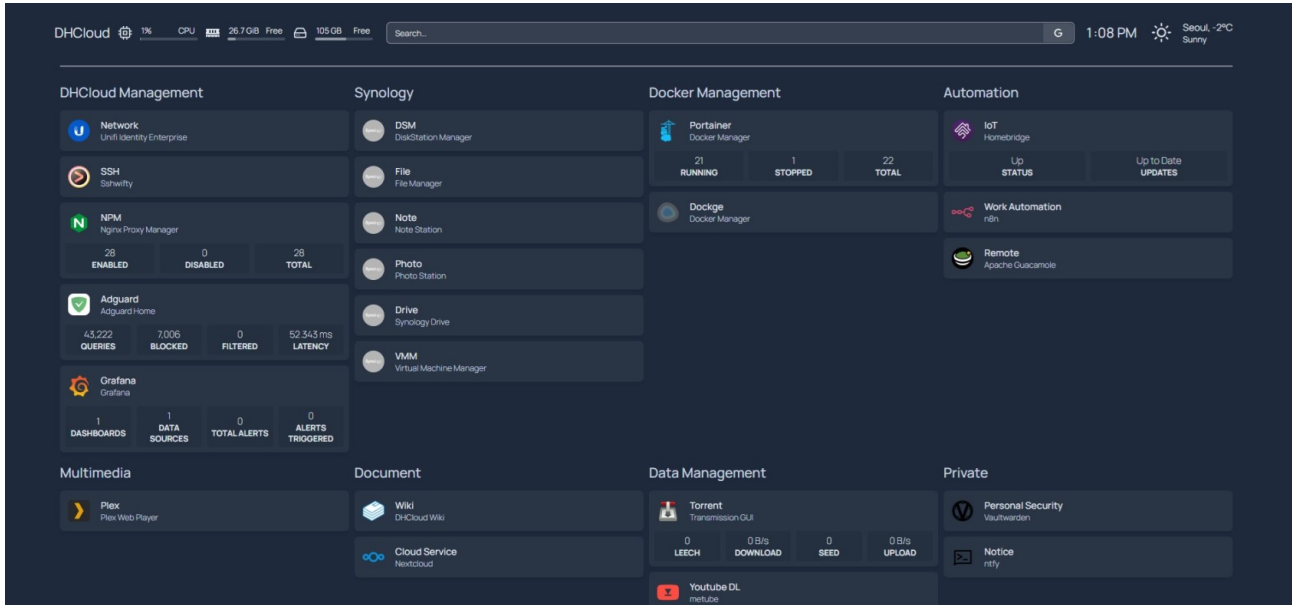
Best Dashboard : GetHomepage

- [GetHomepage 소개](#)
- [설치](#)
- [환경설정 \(Yaml 수정\)](#)

GetHomepage 소개

Yaml 파일로 쉽게 config를 설정할 수 있으며, 사용하시는 app과 쉽게 api 연결을 할 수 있습니다.

본 Wiki 없는 내용은 "<https://gethomepage.dev/main/>" 참고해 주세요.



설치

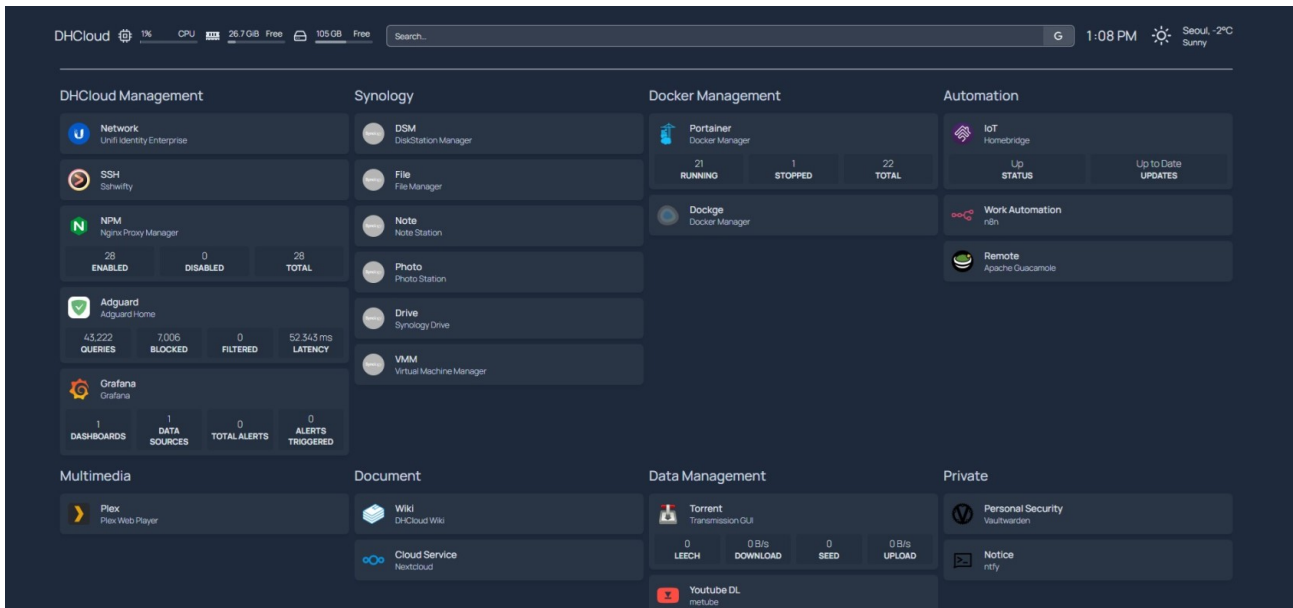
1. "volume1/docker/homepage" 디렉토리를 생성 합니다.
2. "volume1/docker/homepage/public" 디렉토리를 생성 합니다.
3. 아래의 "docker-compose.yml" 파일을 작성 합니다.

```
version: "3.3"
services:
  homepage:
    image: ghcr.io/gethomepage/homepage:latest
    container_name: homepage
    ports:
      - 3550:3000
    volumes:
      - /volume1/docker/homepage:/app/config
      - /var/run/docker.sock:/var/run/docker.sock
      - /volume1/docker/homepage/public:/app/public/images
    environment:
      PUID: 1026
      PGID: 100
    restart: always
```

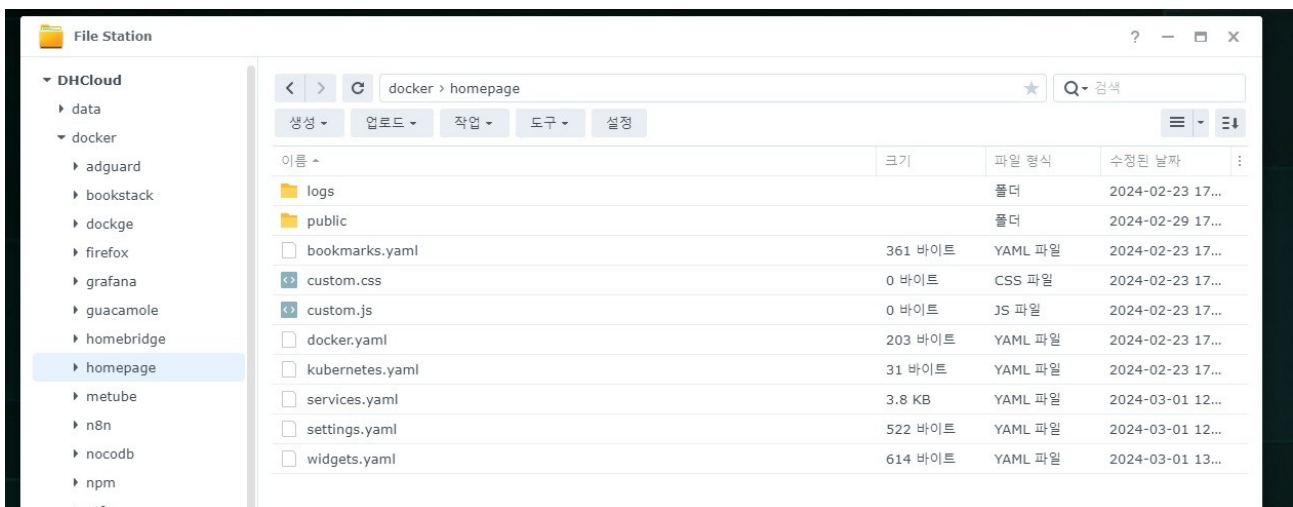
* PUID와 PGID는 사용하시는 값을 넣어 줍니다.

4. "docker-compose up -d"로 yaml을 실행해 도커를 설치 합니다.
5. 접속은 IP:3550입니다.

환경설정 (Yaml 수정)



"docker > homepage" 하위에 아래 와 같은 yaml 파일이 위치해 있습니다.



1. "settings.yaml" : 레이아웃, 페이지 색상등 기본 설정 (Sample File : [settings.yaml](#))

title: 대쉬보드 이름

theme: dark # dark 또는 light로 선택

color: slate # 나열된 색상 중 선택: slate, gray, zinc, neutral, stone, amber, yellow, lime, green, emerald, teal, cyan, sky, blue, indigo, violet, purple, fuchsia, pink, rose, red, white

layout: # 레이아웃 스타일 및 컬럼수 설정

Media:

```
style: row
columns: 4
```

showStats: true # 도커 상태를 표기할 경우 true로 설정

headerStyle: underlined # 헤더 스타일로 4개중 선택 : underlined, boxed, clean, boxedWidgets

2. "widgets.yaml" : 아래 사진과 같이 Header 부분에 들어간 widget 설정 (Sample File : [widgets.yaml](#))



```
- greeting: # 상단좌측 인사말
  text_size: xl
  text: DHCloud

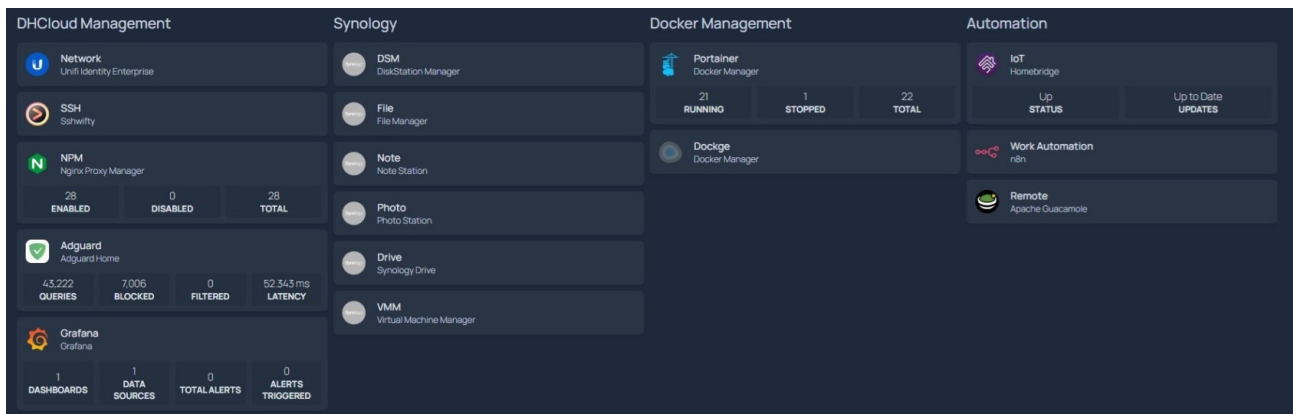
- resources:
  cpu: true
  memory: true
  disk: /

- search: # 검색엔진 선택
  provider: google
  target: _blank

- datetime: # 시간표시 위젯
  text_size: xl
  format:
    timeStyle: short

- weatherapi: # 날씨 위젯
  label: Seoul
  latitude: 37.5519 # 서울지역의 경도
  longitude: 126.9918 # 서울지역의 위도
  units: metric # or imperial
  apiKey: your_api_key # https://www.weatherapi.com/signup.aspx에서 발급 받은 api key 값 입력
  cache: 5 # Time in minutes to cache API responses, to stay within limits
```

3. "services.yaml" : 아래 사진과 같이 Body 부분에 들어간 widget 설정 (Sample File : [services.yaml](#))



1) Nginx Proxy Manager Widget

- NPM:
icon: nginx.png
href: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
description: Nginx Proxy Manager
widget:
type: npm
url: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
username: NPM_아이디
password: NPM_비밀번호

2) Adguard-home Widget

- Adguard:
icon: adguard-home.png
href: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
description: Adguard Home
widget:
type: adguard
url: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
username: Adguard_아이디
password: Adguard_비밀번호

3) Grafana Widget

- Grafana:
icon: grafana.png
href: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
description: Grafana
widget:
type: grafana
url: [https://도메인\(또는 IP\)](https://도메인(또는 IP))
username: Grafana_아이디
password: Grafana_비밀번호

4) Portainer Widget (Portainer Business Edition만 지원)

```
- Portainer:
  icon: portainer.png
  href: https://도메인(또는 IP)
  description: Docker Manager
  widget:
    type: portainer
    url: https://도메인(또는 IP)
    env: 2
    key: Portainer_API_Key
```

5) Homebridge Widget

```
- IoT:
  icon: homebridge.png
  href: https://도메인(또는 IP)
  description: Homebridge
  widget:
    type: homebridge
    url: https://도메인(또는 IP)
    username: Homebridge_아이디
    password: Homebridge_비밀번호
```

6) Transmission Widget

```
- Torrent:
  icon: transmission.png
  href: https://도메인(또는 IP)
  description: Transmission GUI
  widget:
    type: transmission
    url: https://도메인(또는 IP)
    username: Transmission_아이디
    password: Transmission_비밀번호
```

7) Tautulli Widget

```
- Tautulli:
  icon: tautulli.png
  href: https://도메인(또는 IP)
  description: Plex Server Monitoring
  widget:
    type: tautulli
    url: https://도메인(또는 IP)
    key: ****
```

4. "bookmarks.yaml" : 아래 사진과 같이 Bottom 부분에 들어간 바로가기 설정

Developer

GH

Github

Social

github.com

RE

Reddit

Entertainment

reddit.com

YT

YouTube

youtube.com