

# Homelab

☐ ☐ ☐

- [☐☐☐☐ ?](#)
- [☐☐☐☐](#)
  - [☐ ☐☐☐☐ ☐☐](#)
  - [OCI \(☐☐☐☐ ☐☐☐☐☐☐\)](#)
  - [UPS setting](#)
  - [rpi5 cluster](#)
  - [☐☐ ☐☐☐☐](#)
- [☐☐☐☐☐☐](#)
  - [pocket-id](#)
  - [traefik](#)
  - [pangolin](#)
  - [Obsidian](#)
  - [uptime kuma](#)



?

□□□□ □ □□□ □□ □□□□ , □□□□ "home laboratory = □□ □□ □□□□ " □□□□ . □□□□□□  
□□□□ □□ □□□□ (?)□□□□ □□ □□ □□ □□□□ □□□□ □□□□ □□ □□ □□□□□□ .



□□□□



□□ □□□□ □□□ □□□□□ □□ □□□ □□□ □□□□□ . □□□□□ □□ □□□□□ □□□ □□□□□  
□□ □□□□ □□□ □□□□□ , □□□□ □□ □□□□ □□□□□ .

1. 80(http), 443(https) port □ □□□□ □□ (□□□□□ inbound□ □□ )

1. □□□□ □□□□□ □□ □□□ □□□ □□□ □□□□□ □□□ □□□□□ □□□□□

2. □□□ □□

□ □□□ □□□□□ , □□□ □□□□ □ □□□□ □ □ □□ □□□ □□□□□□□□ .

OCI

# OCI (Oracle Cloud Infrastructure)

OCI is a cloud platform ...

OCI provides a wide range of services, including compute, storage, and networking. It is designed to be highly available and secure. OCI is compatible with many popular operating systems and applications. OCI is available in multiple regions, including AWS, Azure, and GCP (Google Cloud). OCI is also compatible with many popular databases, including Oracle Database, MySQL, and PostgreSQL. OCI is a highly scalable and flexible cloud platform that can be used for a wide range of applications and workloads.

UPS

# UPS setting

## UPS ?

UPS (Uninterruptible Power Supply) (不间断电源) 是用于保护设备免受电源中断影响的设备。它可以在主电源失效时提供备用电源，确保设备在一段时间内继续运行。UPS 通常用于服务器、网络设备和关键业务系统。 (DC 输出 或 sine AC 输出 )

24 小时不间断运行 或 在电源故障时提供备用电源 。

UPS 类型	工作原理	特点
后备式 UPS	市电 + 电池 供电	简单、经济
在线式 UPS	市电 经逆变器 供电	纯净、稳定
在线互动式 UPS	市电 经逆变器 (AC-DC-AC)	性价比高

## UPS 安装

UPS 安装 1. 确保 UPS 安装在通风良好的地方，ups 通过 usb cable 连接到设备。 2. 检查 UPS 的输入电压和频率是否与市电匹配。 3. 安装 ups agent 软件以监控 UPS 状态。

4. 配置 UPS 的报警和关机策略。 5. 定期检查 UPS 的电池状态和负载情况。 6. 确保 UPS 的接地良好。

7. 对于大型 UPS 系统，可能需要配置 NUT master/slave 架构。 8. 定期维护 UPS，包括清洁和更换电池。

[NUT](#) (Network UPS tools) 是一个开源的 UPS 管理软件。它支持多种 UPS 品牌，并提供 web 界面和 API。 配置 ups 和 PDU 可以通过 NUT 实现。 使用 dashboard docker app 或 [PeaNUT](#) 可以更方便地管理 UPS。

## 配置

配置 ups 24 小时运行 或 在电源故障时提供备用电源。 24 小时运行 或 在电源故障时提供备用电源。 配置 24 小时运行 或 在电源故障时提供备用电源。

docker

ups usb rpi

```
lsusb | grep -i ups # ??? ???? ups? ???? ?? ??
```

```
root@amati-common:/home/preserde#  
root@amati-common:/home/preserde# lsusb | grep -i ups  
Bus 004 Device 002: ID 051d:0003 American Power Conversion UPS  
root@amati-common:/home/preserde#
```

vender id 051d product id 0003 . docker compose

```
services:  
  nut-upsd:  
    image: instantlinux/nut-upsd:latest  
    container_name: nut-upsd  
    restart: unless-stopped  
    ports:  
      - "3493:3493"  
    environment:  
      - TZ=Asia/Seoul  
      - API_USER=upsmon  
      - API_PASSWORD=upsmon # need to change if server expose  
      - DRIVER=usbhid-ups  
      - GROUP=nut  
      - NAME=ups  
      - POLLINTERVAL=15  
      - PORT=auto  
      - SERVER=master # master mode  
      - VENDORID=051d # lsusb  
      - DESCRIPTION=APC Smart-UPS SMT750RMI2UC  
    devices:  
      - /dev/bus/usb:/dev/bus/usb  
    privileged: true  
    volumes:  
      - nut-config:/etc/nut  
    healthcheck:  
      test: ["CMD", "upsc", "ups@localhost"]  
      interval: 30s  
      timeout: 10s  
      retries: 3  
  peanut:  
    image: brandawg93/peanut:latest  
    container_name: PeaNUT  
    restart: unless-stopped  
    depends_on:  
      - nut-upsd  
    environment:  
      WEB_PORT: 8080  
    ports:  
      - "8080:8080" # Access the dashboard at http://localhost:8080
```

```
volumes:
  - /path/to/config:/config
```

```
volumes:
  nut-config:
```

```
peanut .
. docker-compose docker
```

```
root@amati-common:/data/nut# nano docker-compose.yml
root@amati-common:/data/nut# docker compose up
[+] up 8/8
✓ Image instantlinux/nut-upsd:latest Pulled 5.2s
✓ Network nut_default Created 0.0s
✓ Volume nut_nut-config Created 0.0s
✓ Container nut-upsd Created 0.2s
Attaching to nut-upsd
nut-upsd | ** This container may not work without setting for SERIAL **
nut-upsd | Network UPS Tools upsdrvctl - UPS driver controller 2.8.3 release
nut-upsd | Network UPS Tools 2.8.3 release - Generic HID driver 0.62
nut-upsd | USB communication driver (libusb 1.0) 0.50
nut-upsd | interrupt pipe disabled (add 'pollonly' flag to 'ups.conf' to get rid of t
his message)
nut-upsd | Using subdriver: APC HID 0.100
nut-upsd | Listening on socket /var/run/nut/usbhid-ups-ups
nut-upsd | Network UPS Tools upsd 2.8.3 release
nut-upsd | Ignoring invalid pid number 0
nut-upsd | listening on 0.0.0.0 port 3493
nut-upsd | Connected to UPS [ups]: usbhid-ups-ups
nut-upsd | Found 1 UPS defined in ups.conf
nut-upsd | Network UPS Tools upsmon 2.8.3 release
nut-upsd | Ignoring invalid pid number 0
nut-upsd | Using power down flag file /etc/killpower
nut-upsd | UPS: ups@localhost (primary) (power value 1)
```

```
load
APC ups modbus upsd load
nut-upsd -> peanut -> homepage dashboard load NaN
```

<https://forums.unraid.net/topic/74208-apc-smartups-setup/#comment-736429>



PeaNUT

APC SMT750RMI2U 230V 750VA 500W

100%  
BATTERY CHARGE

NaN%  
UPS LOAD

Online  
UPS STATUS

[[[ APC UPS ]]] [[[[ modbus]]] ]]] .

1. [[[[ ]]] UPS]] ]]] .

2. ]]] ModBus]] ]]] .

3. ModBus]] ]]] . ]]] APC UPS ]]] ModBus]] ]]] ]]] .

📄

# rpi5 cluster

rpi5 cluster | microk8s | 📄 | 📄 | 📄 | 📄 | 📄 .

- 📄 rpi5 📄 4📄 📄 📄 , 📄 master node 3📄 , worker node 1📄 📄 📄
- 📄 5📄 📄 1📄 📄 📄 365📄 📄 📄 📄 📄 📄 .  
📄 📄 master 3📄 , worker 2📄 📄 📄

<https://ubuntu.com/tutorials/how-to-kubernetes-cluster-on-raspberry-pi> ubuntu | 📄 | 📄

microk8s | 📄 | 📄 | 📄 | 📄 | 📄 | 📄 | 📄 | 📄 | ubuntu +  
microk8s | 📄 | 📄 | 📄 | 📄 | 📄 .

📄 📄 📄 📄 📄 📄 📄 📄 (📄 📄 termius | 📄 📄 )📄 4📄  
📄 📄



```

sudo snap install microk8s --classic # ?? ??
sudo usermod -a -G microk8s $USER # sudo? ???? ?? microk8s ? ??? ? ??
sudo chown -f -R $USER ~/.kube
su - $USER

microk8s status --wait-ready # ?? ??? ??? ???? ??

```

alias bashrc .

```

{
  echo alias kubect1="'microk8s kubect1'"
  echo alias kcd="'microk8s kubect1 describe'"
} >> ~/.bashrc && \
source ~/.bashrc

```

alias addon

```

kubect1 get nodes
kubect1 get pods
kubect1 get services

microk8s enable ha-cluster # master node? 3? ??? ?? high availability cluster? ??? ? ????
addon ???

microk8s enable dashboard
microk8s enable dns
microk8s enable registry

```

microk8s add-node

```

$ microk8s add-node
From the node you wish to join to this cluster, run the following:
microk8s join 192.168.1.42:25000/25da722579420c2e724751d3c4be5595/dbf58b806c46

Use the '--worker' flag to join a node as a worker not running the control plane, eg:
microk8s join 192.168.1.42:25000/25da722579420c2e724751d3c4be5595/dbf58b806c46 --worker

If the node you are adding is not reachable through the default interface you can use one of
the following:
microk8s join 192.168.1.42:25000/25da722579420c2e724751d3c4be5595/dbf58b806c46
microk8s join
fd6c:2c0c:b217:3d0b:2ecf:67ff:fe93:8166:25000/25da722579420c2e724751d3c4be5595/dbf58b806c46

```

```

microk8s add-node 25000 vm 25000

```

2 3 rpi master node --worker join

```

$ microk8s join 192.168.1.42:25000/25da722579420c2e724751d3c4be5595/dbf58b806c46
WARNING: Hostpath storage is enabled and is not suitable for multi node clusters.

```

```
Contacting cluster at 192.168.1.42
Waiting for this node to finish joining the cluster. . . . .
Successfully joined the cluster.
```

```
4 rpi worker join --worker .
```

```
3 4 . microk8s status ha-cluster .
```

```
$ microk8s status
microk8s is running
high-availability: yes
  datastore master nodes: 192.168.1.42:19001 192.168.1.43:19001 192.168.1.44:19001
  datastore standby nodes: none
addons:
  enabled:
    dashboard      # (core) The Kubernetes dashboard
    dns             # (core) CoreDNS
...
```





□□ □□□ □□□□ , self-hosted □ □ □□





```

volumes:
  - "./data:/app/data"
# Optional healthcheck
labels:
  - "traefik.enable=true"
  - "traefik.http.routers.pocket-id.rule=Host(`YOUR_OIDC_DOMAIN`)"
  - "traefik.http.routers.pocket-id.entrypoints=websecure"
  - "traefik.http.routers.pocket-id.tls=true"
  - "traefik.http.routers.pocket-id.tls.certresolver=letsencrypt"
  - "traefik.http.services.pocket-id.loadbalancer.server.port=1411"
networks:
  - proxy
healthcheck:
  test: [ "CMD", "/app/pocket-id", "healthcheck" ]
  interval: 1m30s
  timeout: 5s
  retries: 2
  start_period: 10s

networks:
  proxy:
    external: true

```

.env `TRUST_PROXY` `MAXMIND_LICENSE_KEY` `PGID` `PUID` `ENCRYPTION_KEY` `APP_URL` .

```

# See the documentation for more information: https://pocket-
id.org/docs/configuration/environment-variables

# These variables must be configured for your deployment:
APP_URL=https://auth.your.domain

# Encryption key (choose one method):
# Method 1: Direct key (simple but less secure)
# Generate with: openssl rand -base64 32
ENCRYPTION_KEY=...
# Method 2: File-based key (recommended)
# Put the base64 key in a file and point to it here.
# ENCRYPTION_KEY_FILE=/path/to/encryption_key

# These variables are optional but recommended to review:
TRUST_PROXY=true
MAXMIND_LICENSE_KEY=...
PUID=1000
PGID=1000

```

`TRUST_PROXY` `MAXMIND_LICENSE_KEY` `PGID` `PUID` `ENCRYPTION_KEY` `APP_URL` `docker` `audit log` `IP` `ip` .

`audit log` `IP` `ip` , pocket-id docs <https://pocket-id.org/docs/client-examples>







📄

# pangolin

📄 📄 📄 pangolin 📄 . 📄 📄 newt 📄 wireguard 📄 VPN 📄 cloudflare 📄 📄 📄 pangolin UI + traefik + newt 📄 📄 .

- 📄 📄 VPN 📄 📄
- 📄 SSO 📄 📄 , 📄 OIDC provider 📄 📄 📄 .
- CrowdSec 📄

pangolin 📄 traefik 📄 📄 📄 pangolin 📄 traefik 📄 wrapper 📄 📄 , 📄 📄 traefik 📄 📄 📄 . traefik + authentic 📄 📄 📄 UI 📄 📄 📄 .

📄 , Traefik 📄 📄 📄 .

- traefik 📄 📄 k8s, docker 📄 label 📄 📄 📄 📄 📄 📄 dynamic config 📄 📄 (file provider) 📄 📄 .
- pangolin 📄 📄 / 📄 config 📄 📄 , (traefik 📄 dynamic config 📄 ) 📄 📄 UI 📄 📄 .

📄 📄 VM 📄 📄 📄 📄 📄 📄 📄 📄 dynamic config 📄 📄 📄 UI 📄 📄 📄 📄 .

📄 newt 📄 vpn 📄 tailscale 📄 📄 📄 . 📄

📄 📄 📄 📄 📄 📄 . 📄 📄 📄 📄 📄 📄 .

- 📄 📄 📄
- 📄 IdP 📄 📄
- 📄 📄 📄 API

📄 📄 📄 📄 homelab 📄 📄 📄 📄 📄 📄 📄 supporter key 📄 📄 📄 📄 .

📄 📄 📄 📄 , 📄 📄 📄 📄 📄 , 📄 ... pangolin 📄 📄 📄 📄 , 📄 📄 📄 📄 📄 📄 📄 📄 ... 📄 ...



Obsidian

# Obsidian

Obsidian은 markdown 파일을 사용하여 개인 지식을 정리하고, 동기화 및 게시 기능을 제공하는 개인 지식 관리 도구입니다.

- 오픈 소스, 크로스 플랫폼 - markdown, cross platform
- 동기화 (sync, publish) 기능
- [Obsidian Sync](#) - 클라우드 동기화 서비스
- 모바일 앱
- 유료 버전 (월 4\$)

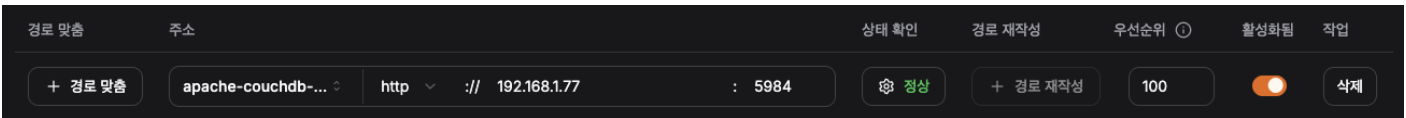
Obsidian은 개인용이지만, 팀 협업이나 공개 블로그를 위한 플러그인도 있습니다.

Obsidian을 self-hosted 방식으로 배포하는 방법은 [self-hosted livesync](#) 플러그인이나 `db`를 `apache couchDB`로 대체할 수 있습니다.

설치

couchDB LXC (on proxmox). proxmox script [couchDB LXC](#) (IP 주소 127.0.0.1:5984)

pangolin reverse proxy ( <https://couchdb-sample.your.domain> -> 127.0.0.1:5984)



Healthcheck URL 설정: `url: http://127.0.0.1:5984/_up`

### 상태 확인 설정

192.168.1.77:5984에 대한 상태 모니터링 설정

**상태 확인 활성화**  
 이 대상을 모니터링하여 건강 상태를 확인하세요. 필요에 따라 대상과 다른 엔드포인트를 모니터링할 수 있습니다.

방법	IP / 호스트	포트	경로
HTTP	192.168.1.77	5984	/_up

HTTP 메소드

livesync plugin DB require\_valid\_user: true healthcheck url require\_valid\_user\_except\_for\_up: true . health check url

chttpd	bind_address	0.0.0.0	
	enable_cors	true	
	max_http_request_size	4294967296	
	port	5984	
	require_valid_user	true	
	require_valid_user_except_for_up	true	

json http login dialog

[https://www.reddit.com/r/selfhosted/comments/1eo7knj/guide\\_obsidian\\_with\\_free\\_selfhosted\\_instance\\_sync/?tl=ko](https://www.reddit.com/r/selfhosted/comments/1eo7knj/guide_obsidian_with_free_selfhosted_instance_sync/?tl=ko) db obsidian app

📄

# uptime kuma

📄 📄 📄 📄 . 📄 📄 📄 📄 📄 📄 prometheus + grafana 📄 📄 📄 📄 📄 , 📄 📄 📄 📄 📄 📄 uptime kuma 📄 📄 📄 📄 .

<https://github.com/louislam/uptime-kuma>

- 📄 📄 📄 📄 : 📄 📄 📄 📄 📄 📄 📄 📄 📄 📄 .
- 📄 📄 📄 📄 : HTTP/HTTPS, TCP, Ping, DNS, Keyword 📄 📄 📄 📄 📄 .
- 📄 📄 : 📄 📄 📄 📄 (Slack), 📄 📄 (Discord), 📄 📄 (Telegram), 📄 📄 📄 📄 .
- SSL 📄 📄 : SSL 📄 📄 📄 📄 📄 📄 📄 📄 .
- 📄 📄 (Self-Hosted): 📄 📄 📄 📄 📄 📄 📄 📄 📄 📄 .
- 📄 📄 📄 : 📄 📄 📄 📄 UI 📄 📄 📄 📄 📄 📄 📄 .
- 📄 📄 📄 : Uptime Kuma Manager 📄 📄 📄 📄 📄 📄 📄 .

📄 📄 docker compose 📄 📄 📄 📄 📄 . (📄 📄 📄 traefik 📄 📄 📄 📄 )

```
services:
  uptime-kuma:
    container_name: uptime-kuma
    image: louislam/uptime-kuma:2
    restart: unless-stopped
    volumes:
      - ./data:/app/data
    ports:
      - "3001:3001"
    environment:
      - PUID=1000
      - PGID=1000
    #labels:
    # - "traefik.enable=true"
    # - "traefik.http.routers.uptimekuma.rule=Host(`monitor.neue.nz`)"
    # - "traefik.http.routers.uptimekuma.entrypoints=websecure"
    # - "traefik.http.routers.uptimekuma.tls=true"
    # - "traefik.http.routers.uptimekuma.tls.certresolver=letsencrypt"
    # - "traefik.http.services.uptimekuma.loadbalancer.server.port=3001"
    #networks:
    # - proxy

#networks:
```



### some service is down

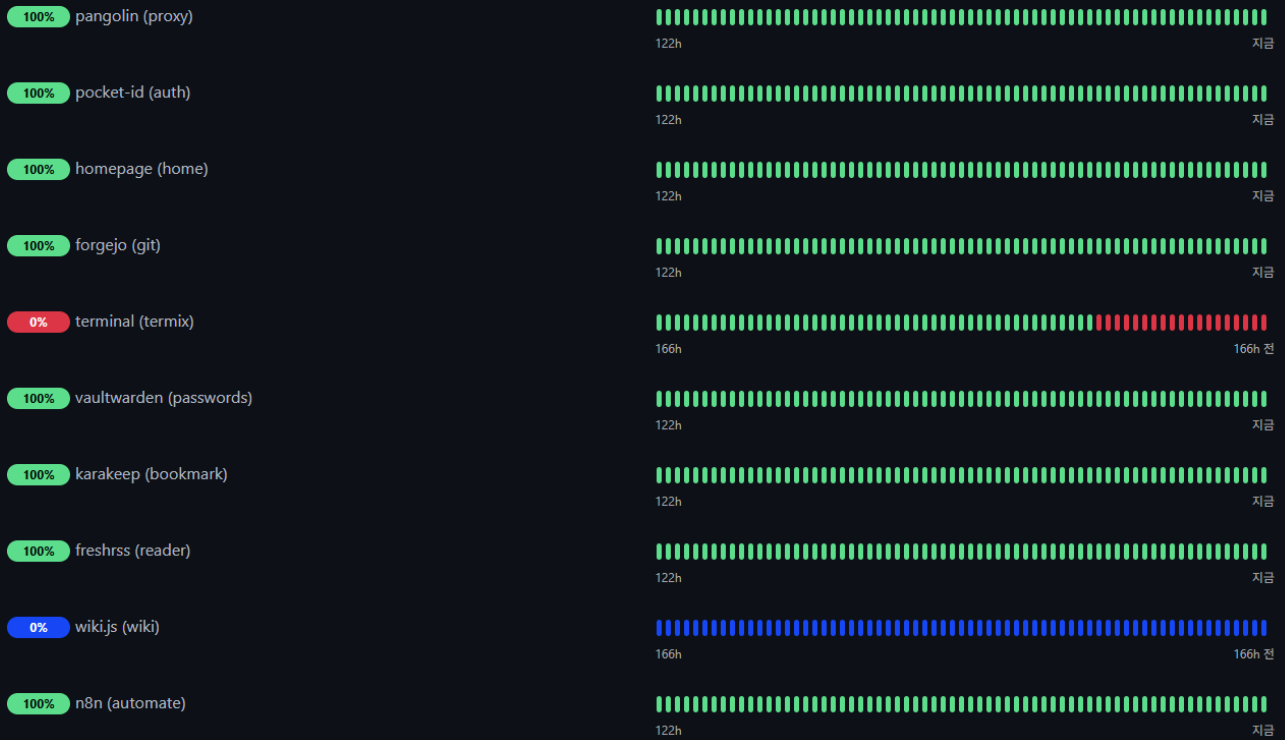
wiki, termix is down

생성된 날짜: 2026-01-09 16:36:20 (7 days ago)

점검 중

I'm not a kiwi :)

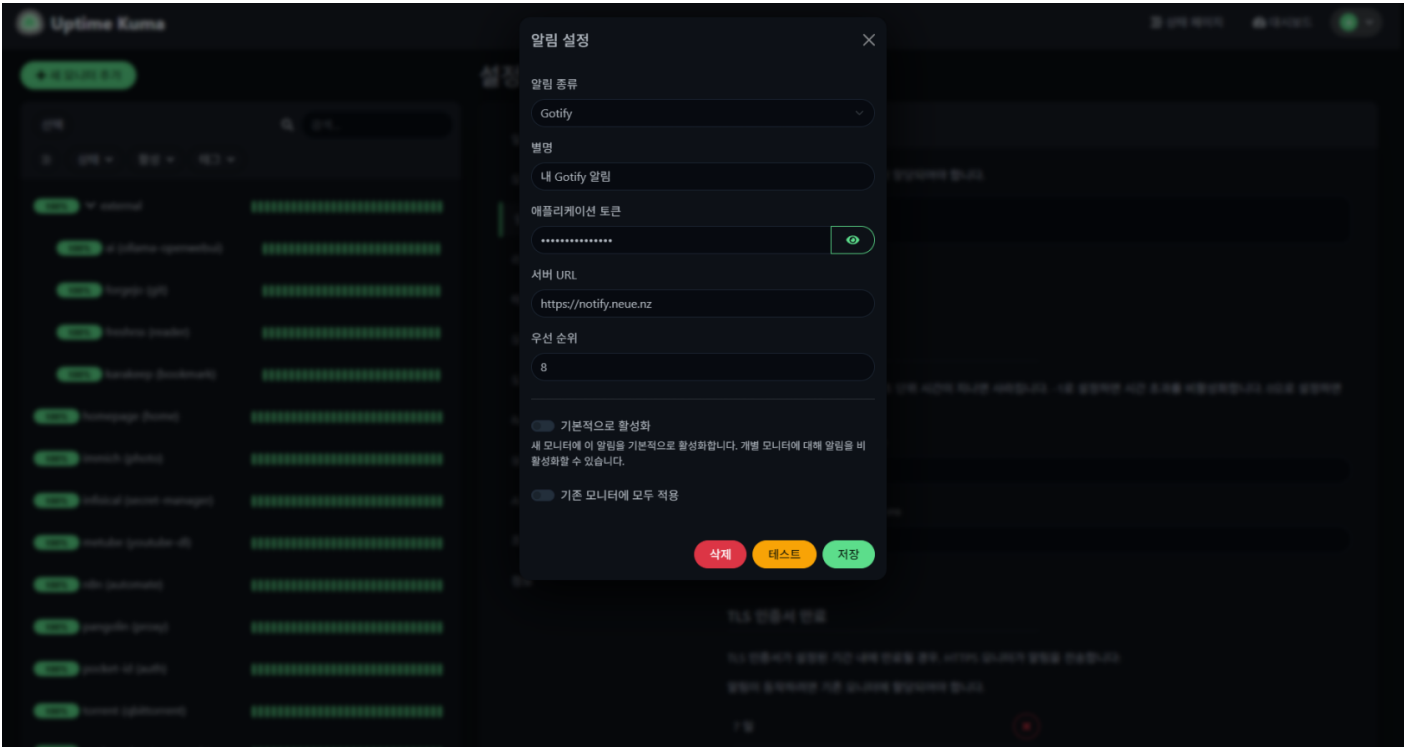
### 서비스



□□ □□□□ □□ □□ □□ (□□ □□ □□ □□ □□□□ traefik□□ OIDC middleware□  
□□□□ □□□□ □□□□ □□ ) □□□□ □□ □□□□ status □□□□ □ □ □□ □□ □□ □□

.





이 글에서는 Uptime Kuma의 알림 설정 방법을 소개합니다. 알림 설정은 Uptime Kuma의 중요한 기능 중 하나입니다.

- Uptime Kuma는 다양한 알림 채널을 지원합니다.
- Uptime Kuma는 docker compose, config 파일을 사용하여 설정을 관리할 수 있습니다.
- Uptime Kuma는 prometheus exporter를 사용하여 모니터링 데이터를 수집하고, grafana를 사용하여 시각화할 수 있습니다.

이 글에서는 Uptime Kuma의 알림 설정 방법을 소개합니다. 알림 설정은 Uptime Kuma의 중요한 기능 중 하나입니다.