하드웨어

홈랩을 구성하는 하드웨어 부분

- 홈 네트워크 구성
- OCI (오라클 클라우드)
- UPS setting
- rpi5 cluster
- 홈랩 구축기

홈 네트워크 구성

여러 서비스를 스스로 운영하려면 결국 서버를 돌려야 합니다. 한국에서는 일반 가정집에서 서버를 돌리는 것은 여러가지 제약이 있는데, 다음과 같은 문제가 있습니다.

1. 80(http), 443(https) port 가 막혀있는 경우 (정확히는 inbound가 막힘)
    1. 외부에서 네트워크를 통해 내부의 정보를 요청할 수 없으므로 서버가 있어도 서비스가 불가능
2. 속도가 느림

위 문제가 대표적인데, 이것을 우회하여 웹 서비스를 할 수 있는 방법이 몇가지 존재합니다.

OCI (오라클 클라우드)

오라클 클라우드에 대해 ...

여러 클라우드 서비스가 있고 인터넷에 연결된 VM을 운영할 수 있지만, 완전히 무료로 쓸 수 있는 신뢰할 수 있는 서비스가 몇 없습니다. 가장 크고 기업용으로도 쓰이는 서비스가 aws, azure, gcp (google cloud) 이 세가지인데, 아무리 light하게 쓰더라도 비용을 피할 수는 없습니다.

UPS setting

UPS란 ?

UPS는 `Uninterruptible Power Supply`(무정전 전원 장치)의 약자로, 갑작스러운 순간 정전이나 긴 정전, 전력이 불안정한 상황에 연결된 장치에 일정시간 동안 안정된 전력을 계속 공급할 수 있는 장치이고, 고급형의 경우 외부 전력을 양질의 전력으로 바꿔주는 기능도 합니다. (DC로 변환한 후 다시 순수 sine파 AC로 재변환)

24시간 동작하는 서버 등 중요한 장비의 오작동을 방지하고 안전한 종료를 보장하는 기기입니다.

유형	변환 방식	양질 전력 제공 여부
오프라인	직통 + 정전 시 변환	제한적
라인 인터랙티브	부분 조정 + 변환	중간
온라인	항상 이중 변환 (AC-DC-AC)	우수

UPS 운영 전략

만약 가지고 있는 기기가 UPS와 서버 1대로만 이루어져 있다면, ups에서 제공하는 usb cable과 전원 케이블을 서버와 직통으로 연결해서 관리해도 됩니다. 보통 시놀로지, unraid 등의 유명 운영체제는 기본적으로 ups agent가 미리 내장되어 설정할 수 있습니다.

다만, ups에 서버가 여러대 연결될 경우엔, usb cable에 연결된 장치는 보호받을 수 있지만 정전 정보를 받을 수 없는 나머지 기기들은 배터리가 다 소진될때까지 동작하다가 갑자기 꺼져버리게 되어 의미가 없습니다.

이런 경우 대안은 NUT를 운영하는 것입니다. ups의 usb 케이블은 NUT master에서 받아서 전원을 관리하고, 유사 시 정전 정보를 브로드캐스팅하며 각 서버의 NUT slave에서 이 정보를 받아 안전하게 서버를 종료합니다.

NUT는 `Network UPS tools`의 약자이며 ups 뿐만 아니라 PDU, 태양광 장치 등의 전력 기기의 모니터링을 위한 소프트웨어 모음입니다. 서드파티에서 관리하는 만큼 최신형 기기나 해당 ups가 네트워크 관련 특화로 별도 드라이버를 제공하지 않는 경우 NUT에서 모니터링이 어려운 경우도 있습니다. 여담으로 ups 상태를 모니터링하는 dashboard docker app으로 PeaNUT이 있습니다.

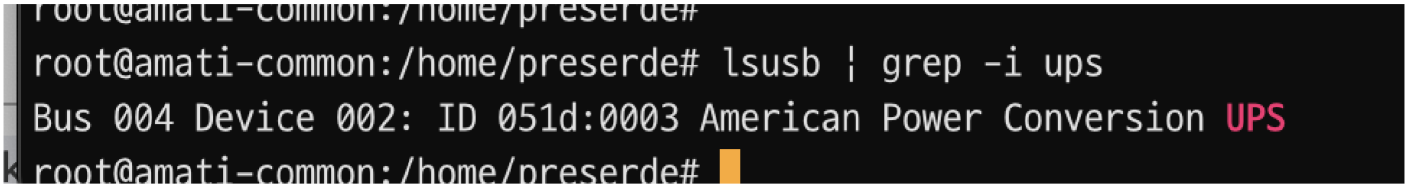
라즈베리파이 서버

우선 ups 정보를 24시간 감시할 저전력 서버를 준비합니다. 24시간 감시를 해야하므로 항상 켜져 있어야 하지만, 시스템 자원은 아주 적게 소모하기 때문에 라즈베리파이 서버가 가장 좋은 대안이긴 합니다. 하지만 필요에 따라서 docker를 운영할 수 있는 어떤 서버도 문제 없습니다.

우선 ups의 usb 케이블을 연결한 후 rpi의 터미널에서 다음 명령으로 출력을 확인합니다 .

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

잘 연결되어 있는 상태라면 다음과 비슷한 응답이 돌아옵니다 .



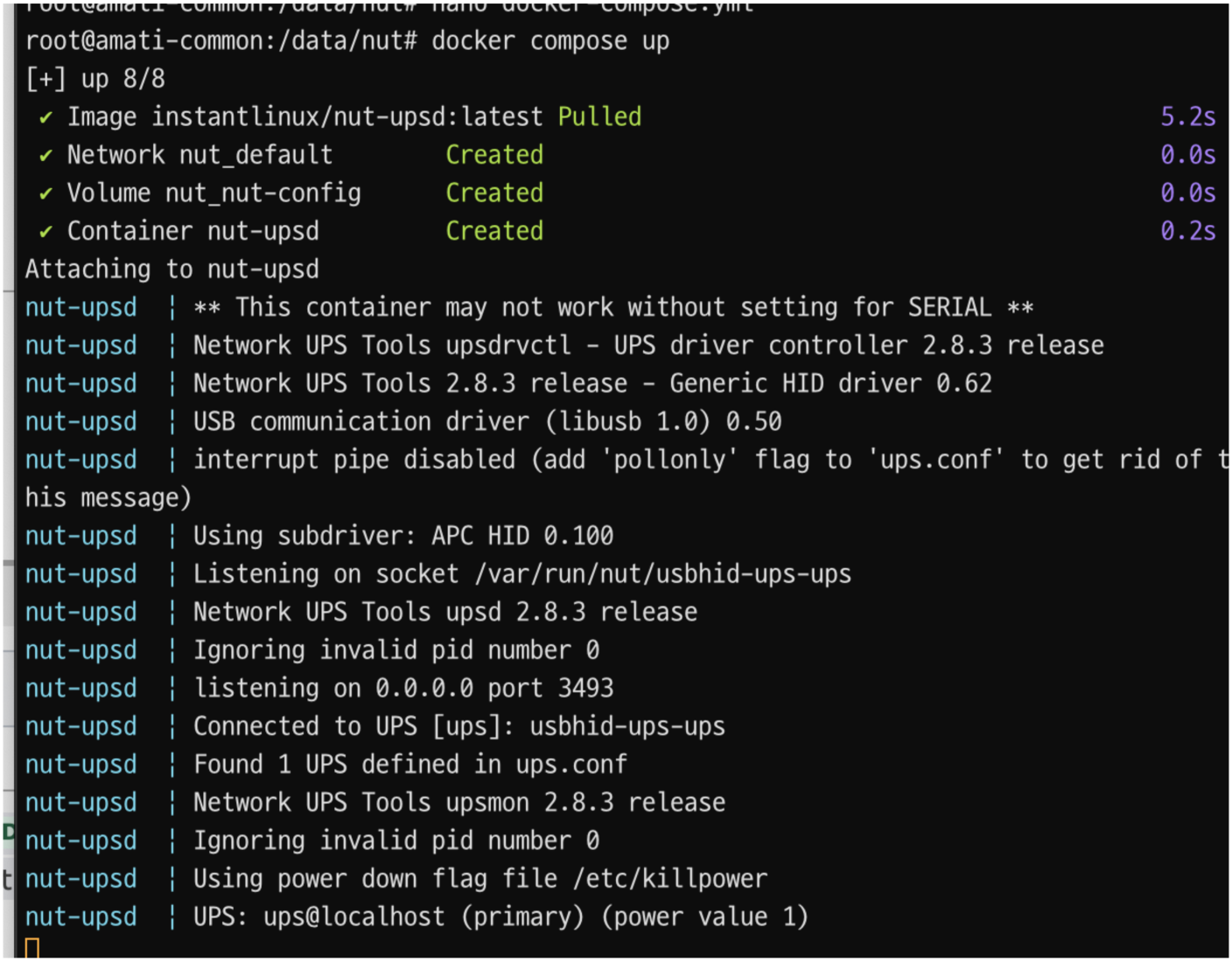
여기서는 vender id `051d` 와 product id `0003` 이 중요합니다 . 다음으로 docker compose 파일을 작성합니다 .

```
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:
```

위 설정에서는 간단한 배터리 대시보드인 peanut 서비스까지 통합했습니다 .

보통 포트는 일반적으로 3493번을 이용합니다 . docker-compose 파일을 작성한 후 docker 를 동작시키면 다음과 같은 응답을 받을 수 있습니다 .



load가 표시되지 않는 경우

APC ups의 경우 기본적으로 modbus가 비활성화되어 있고 이것 때문에 upsd의 출력 중에 load가 노출되지 않는 경우가 있습니다 . 그래서 nut-upsd -> peanut -> homepage dashboard에서 load가 NaN으로 표시되는 경우가 있습니다 .

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

이때엔 APC UPS 자체의 설정에서 modbus를 활성화하면 된다고 합니다 .

1. 화면에 표시되는 메뉴를 통해 UPS의 고급 설정 옵션을 활성화합니다 .
2. 설정 옵션을 열고 ModBus를 찾을 때까지 아래로 스크롤합니다 .
3. ModBus를 활성화로 변경하십시오 . 모든 APC UPS 장치에서 ModBus는 기본적으로 비활성화되어 있습니다 .

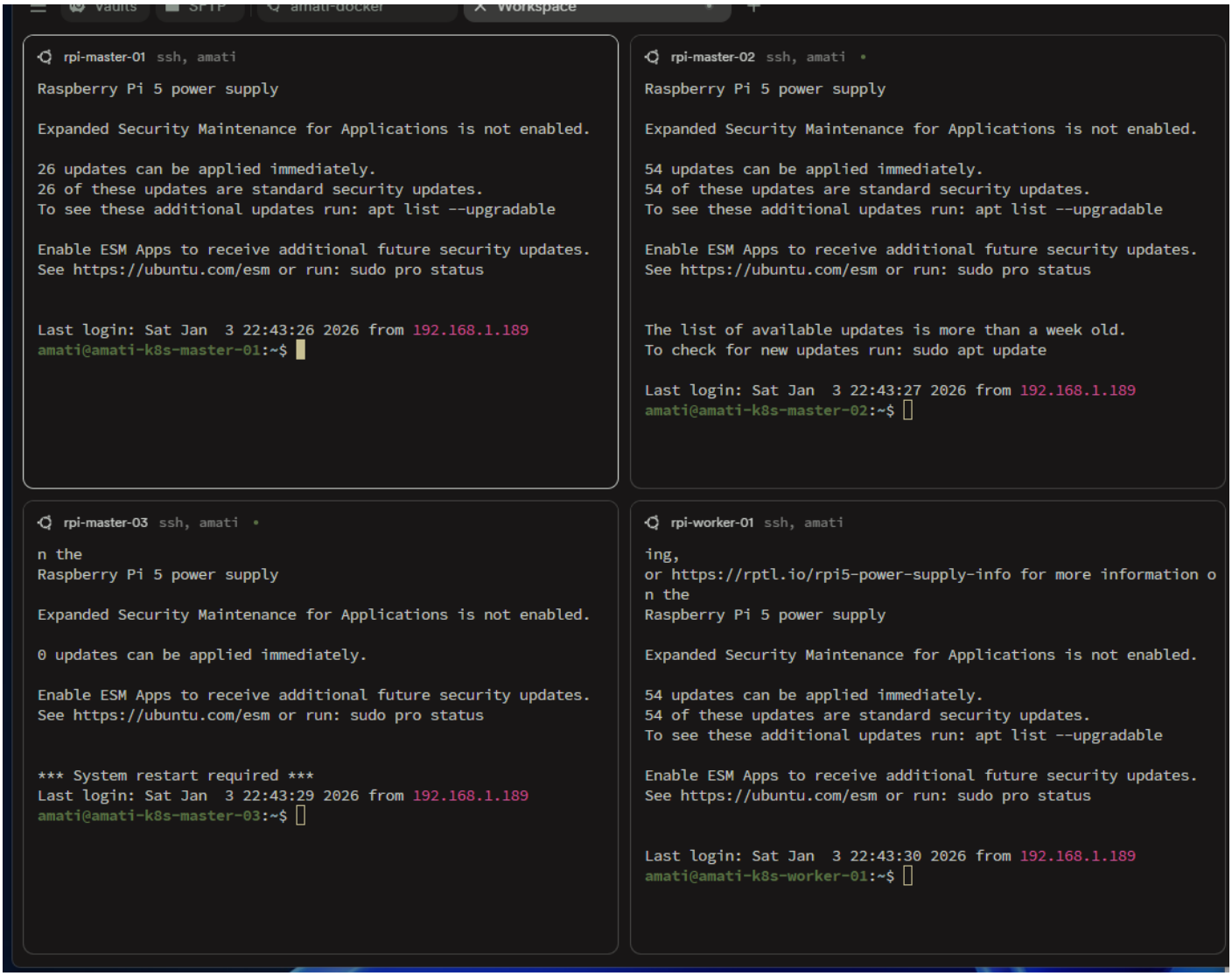
rpi5 cluster

rpi5 cluster에 microk8s 패키지를 설치해서 클러스터를 구성하는 방법을 알아보자.

- 현재 rpi5 기기는 4대가 준비되어 있고, 구성은 master node 3대, worker node 1대로 구성하기로 함
- 원래 5대였으나 1대는 집안에서 완벽하게 365일 전원이 들어오는 서버로 활용하려고 따로 빼놓음. 만약 이경우엔 master 3대, worker 2대로 구성했을 듯

https://ubuntu.com/tutorials/how-to-kubernetes-cluster-on-raspberry-pi ubuntu를 만든 캐노니컬에서 microk8s도 만들고 있기 때문에 아주 권장되는 조합이고 공식 매뉴얼에서도 라즈베리 몇대를 묶어 ubuntu + microk8s 를 설치하는 가이드를 찾을 수 있다.

우선 여러개의 터미널을 동시에 입력시킬 수 있는 앱 (여기서는 윈도에서 termius를 사용합니다)으로 4개의 터미널을 띄움



```
Raspberry Pi 5 power supply

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
26 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sat Jan  3 22:43:26 2026 from 192.168.1.189
amati@amati-k8s-master-01:~$
```

```
Raspberry Pi 5 power supply

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
54 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Jan  3 22:43:27 2026 from 192.168.1.189
amati@amati-k8s-master-02:~$
```

```
n the
Raspberry Pi 5 power supply

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Sat Jan  3 22:43:29 2026 from 192.168.1.189
amati@amati-k8s-master-03:~$
```

```
ing,
or https://rptl.io/rpi5-power-supply-info for more information o
n the
Raspberry Pi 5 power supply

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
54 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sat Jan  3 22:43:30 2026 from 192.168.1.189
amati@amati-k8s-worker-01:~$
```

이런 터미널 앱들은 기본적으로 모든 터미널에 같은 키보드 입력을 할 수 있는 기능이 있으므로 기본적인 업데이트나 k8s 설치를 동시에 진행하는 것이 편리합니다 .

참고로 microk8s 를 설치할 것이기 때문에 docker 를 설치할 필요는 없어서 기본적인 업데이트 후 바로 snap 으로 microk8s를 설치합니다 .

```
apt update && apt upgrade
```

ubuntu와 rpi5의 조합에서는 microk8s를 설치하기 전에 몇가지 작업이 더 필요하다 . `/boot/firmware/cmdline.txt` 에 cgroups를 활성화시켜 줘야 하고 , 24.04 이전 버전의 경우 추가 패키지도 설치해야 한다 . 자세한 내용은 여기__ 참조

```
sudo nano /boot/firmware/cmdline.txt
```

마지막 줄에 다음 내용 추가

```
cgroup_enable=memory cgroup_memory=1
```

이제 microk8s 설치

```
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 # ?? ??? ??? ???? ???
```

이제 편리한 alias 설정을 위해 bashrc 파일을 편집하자 .

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

위의 alias가 잘 동작하는지 확인 및 필요한 addon 활성화

```
kubectl get nodes
kubectl get pods
kubectl get services

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

microk8s enable dashboard
microk8s enable dns
microk8s enable registry
```

이제부터는 클러스터 구성을 위해 전체 입력 가능한 기능을 끄고 개별적으로 입력하면 됩니다 .

```
$ 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` 를 입력하면 각 노드를 클러스터로 묶기 위한 코드와 키를 확인할 수 있습니다 . 이 키는 한번만 사용할 수 있기 때문에 노드를 추가할 때마다 실행해서 새로운 키를 받아야 합니다 . 또한 포트는 25000을 사용하기 때문에 내부 네트웍이 아닌 외부에서 vm끼리 묶을 경우 25000번 포트를 방화벽에서 허용할 필요도 있습니다 .

2번 및 3번 rpi는 master node로 추가할 것이기 때문에 `--worker` 옵션을 넣지 않고 join하면 됩니다 .

```
$ 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.
```

4번 rpi는 worker 노드로 join할 것이기 때문에 `--worker` 옵션을 추가합니다 .

이 과정을 3번 반복하면 총 4대가 클러스터링이 됩니다 . 이후 `microk8s status` 명령으로 상태를 확인하면 자동으로 ha-cluster 기능이 활성화된 것을 확인할 수 있습니다 .

```
$ 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
...
```

홈랩 구축기

대략의 구성은 다음과 같다 .

1. OCI VM : 외부에 도메인과 연결해서 홈랩 내부 서비스를 proxy해주는 서버 , 모니터링이나 OIDC SSO 서비스도 여기서 docker로 추가
2. 라우터 : n305 미니피시 , wifi7 ap, 및 opnsense 구성
3. 스위치허브 : 알리익스프레스 산 10Gbe, POE 지원 스위치 허브 . 10Gb로 구성됨
4. 홈서버 #1 : proxmox (PVE) 하이퍼바이저 운영 . VM, LXC, 그리고 docker로 여러 서비스 운영
5. 홈서버 #2 : 베어메탈로 proxmox backup server (PBS) 운영하면서 vm 등의 컨테이너와 unraid hdd 백업
6. 홈서버 #3 : unraid storage server 운영 . rtx 4000 ada GPU도 포함되어 있어 AI 모델 테스트도 병행
7. 라즈베리파이 5 클러스터 : 쿠버네티스 테스트용 . 개인적으로는 k8s를 본격적으로 공부하긴 벅차고 microk8s 기반 워드프레스로 블로그 설치하는 정도로 운영
8. 메인 데스크탑 : 델 웍스 (스레드리퍼 7955wx) - 2cpu에서 미개봉 제품에서 옵션을 엄청나게 제거해서 ... 스레드리퍼 치고 생각보다 매우 싸게 구매