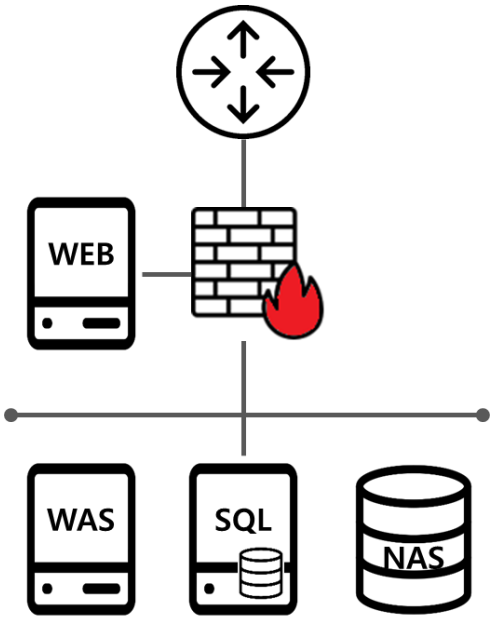
● 서비스 흐름 및 시스템 구성도



구분	WEB	WAS	DB	NAS
OS	Ubuntu 22.04	Ubuntu 22.04	Ubuntu 22.04	Ubuntu 22.04
SW	Apache	Flask, Python3, nfs-common	MariaDB	nfs-kernel-server
IP	192.168.35.100/24	101	102	103

서버 설치 및 구성

1. 서버 준비

- **Web 서버**: Apache 가 설치된 서버.
- **WAS 서버**: Python 백엔드를 실행할 서버.
- **DB 서버**: MariaDB 가 설치된 서버.
- **NAS 서버**

2. 소프트웨어 설치

- **Web 서버 (Apache 설치)**

```
sudo apt update
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
sudo systemctl status apache2
```

- **WAS 서버 (Python 및 필요 라이브러리 설치)**

```
sudo apt update
sudo apt install -y python3 python3-pip
pip3 install flask pymysql

sudo vi /etc/hosts
#{DB 서버 IP} {DB 서버 hostname}
192.168.0.30 dbnas
#테스트 방법: ping dbnas 해서 핑 가면 정상
```

- **DB 서버 (MariaDB 설치)**

```
sudo apt update
sudo apt install -y mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
sudo systemctl status mariadb
```

sudo vi /etc/hosts
#{WAS 서버 IP} {WAS 서버 hostname}
192.168.0.20 was

- NAS 서버

sudo apt update

sudo apt install -y nfs-kernel-server

sudo systemctl restart nfs-server

sudo systemctl enable nfs-server

sudo netstat -nap | grep 2049 #tcp 2049 포트 확인

sudo mkdir /mnt/{storage} #공유 폴더 만들기

예> sudo mkdir /mnt/test

sudo chmod -R 777 /mnt/{storage} #공유 폴더 권한 설정

예> sudo chmod -R 777 /mnt/test

sudo vi /etc/exports

예> /mnt/test 192.168.0.0/24(rw,sync,no_subtree_check,no_root_squash)

예: 모든 접근 허용>

/mnt/test 0.0.0.0/0(rw,sync,no_subtree_check,no_root_squash)

sudo systemctl restart nfs-server

sudo systemctl enable nfs-server

sudo systemctl status nfs-server

※ NAS 서버에 접속할 Client(예: WAS 서버) 설정

sudo apt-get install -y nfs-common

sudo mkdir /mnt/{storage}

예> sudo mkdir /mnt/test

1.마운트 위치를 설정파일로 지정 후 연결하기(서버 재부팅시 연결됨, 비추: Client 부팅시 NAS 가 반드시 동작 중이어야 함)

sudo vi /etc/fstab

예> 192.168.35.103:/mnt/test /mnt/test nfs rw 0 0

2.명령어로 직접 연결하기(추천)

sudo mount -t nfs -o rw {NAS IP}:{NAS 폴더} {Client 폴더}

예>sudo mount -t nfs -o rw 192.168.35.103:/mnt/test /mnt/test

예> sudo mount -t nfs -o rw 192.168.0.30:/mnt/test /mnt/test

sudo mount -t nfs -o rw ${nas_server_ip}:/mnt/test /mnt/test

Client 에서 /mnt/test 폴더가 보이는지 확인(예: ls /mnt/test) -> test.txt 만들고 NAS 서버 /mnt/test 폴더에 test.txt 생겼는지 확인

MariaDB 설정

1. **MariaDB 초기 설정**

sudo mysql_secure_installation

sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
bind-address = 0.0.0.0

2. **DB 테이블, 사용자 생성 및 권한 부여**

```
sudo mysql -u root -p

CREATE DATABASE board_db;

USE board_db;

CREATE TABLE posts (
    id INT(11) NOT NULL AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS post_files (
    id SERIAL PRIMARY KEY,
    post_id INTEGER NOT NULL,
    file_name VARCHAR(255) NOT NULL,
    original_file_name VARCHAR(255) NOT NULL,
    file_size BIGINT NOT NULL DEFAULT 0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (post_id) REFERENCES posts(id) ON DELETE CASCADE
);

CREATE USER 'board_user'@'%' IDENTIFIED BY 'new1234!';   -- new1234!는 패스워드
GRANT ALL PRIVILEGES ON board_db.* TO 'board_user'@'%';
FLUSH PRIVILEGES;
```

3. 방화벽 확인(필요시)

```
sudo ufw status
sudo ufw allow 3306/tcp                                  #MariaDB 는 3306 포트 사용
```

백엔드 (WAS 서버) 설정

Flask 애플리케이션 작성

1. **Flask 애플리케이션 구조**

#필요시, sudo apt install tree 설치

/flask_app
    /templates
        index.html
        post.html
        new_post.html
    app.py

2. **app.py 작성: 첨부 참조**
3. **템플릿 파일 작성: 첨부 참조**

- **index.html**
- **post.html**
- **new_post.html**

DB 서버 변경시(예: RDS 사용)

#한줄 명령

sudo sed -i 's/host="192.168.0.102"/host="10.0.10.102"/' /flask_app/app.py

#수작업 편집

sudo vi /flask_app/app.py

#아래 내용 중 DB 서버 IP(host="192.168.X.102") 변경

```
        db = pymysql.connect(
            host="192.168.0.102",   # MariaDB 서버의 IP 주소
            user="board_user",          # MariaDB 사용자 이름
            password="new1234!",      # MariaDB 사용자 비밀번호
            database="board_db",      # 데이터베이스 이름
            cursorclass=pymysql.cursors.DictCursor
        )
```

#app.py 재실행(app.py 가 boardapp.service 로 등록됨)

```
sudo systemctl daemon-reload
sudo systemctl restart boardapp.service
```

Apache 설정 (웹 서버)

1. **/etc/apache2/sites-available/000-default.conf 수정**

```
sudo vi /etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ProxyPass / http://{was_server_ip}:5000/
    ProxyPassReverse / http://{was_server_ip}:5000/
</VirtualHost>
```

2. **모듈 활성화 및 Apache 재시작**

sudo a2enmod proxy
sudo a2enmod proxy_http
sudo systemctl restart apache2
sudo systemctl enable apache2

테스트

1. **WAS 서버에서 Flask 애플리케이션 실행**

python3 /path/to/flask_app/app.py

2. **웹 브라우저에서 웹 서버 IP 접속**
    - http://web_server_ip 에 접속하여 게시판 확인
    - 글 작성, 조회, 상세보기 기능 테스트

요약

1. **Web 서버**: Apache 설치 및 Proxy 설정.
2. **WAS 서버**: Flask 기반 백엔드 구성.
3. **DB 서버**: MariaDB 설치 및 설정.
4. **Flask 애플리케이션**: 게시판 기능 구현.
5. **테스트**: 브라우저를 통해 서비스 정상 동작 확인.

이로써 게시판 서비스를 위한 서버 설정과 테스트가 완료됩니다.

WEB, WAS, DB 서버 설정

1.OS설치

-D:₩02.Utils₩01.OS

-ubuntu-22.04.4-live-server-amd64.iso

-ID/Password: cloud / new1234!

-대부분 설정 default 값 선택

-mirror 사이트 설정

.기존: /kr.archive.ubuntu.com/

.변경: /mirror.kakao.com/

ㄴ Mirror address: http://mirror.kakao.com/ubuntu/

-계정

.Your name: cloud

.Your servers name: msp

.Pick a username: cloud

.Choose a password: new1234!

-root password 설정

cloud@msp:~$ sudo passwd root

[sudo] password for cloud: {new1234!}

New password: {new1234!}

-openssh 설치, 서버 유형(aws-cli 등) 미선택

sudo apt update

sudo apt install openssh-server

sudo vi /etc/ssh/sshd_config

Port 22

ListenAddress 0.0.0.0

PasswordAuthentication yes

PermitRootLogin yes

PermitEmptyPasswords no

sudo systemctl restart ssh

-Ubuntu hostname 변경

sudo hostnamectl set-hostname {변경하는 호스트명}

-서버의 공인IP(MyIP) 확인

curl ip.me

-VM ip 설정 변경

https://travel-log.tistory.com/32

apt install net-tools

sudo vi /etc/netplan/00-installer-config.yaml

>>>>

network:

  ethernets:

```
    enp0s3:
      addresses:
        - 192.168.35.100/24
        #- 172.30.1.100/24
      gateway4: 192.168.35.1     #home
      #gateway4: 172.30.1.254     #office
      nameservers:
        addresses:
        - 8.8.8.8
        search: []
      #dhcp4: true
  version: 2
```

sudo netplan apply

-VM 여러대 IP 구성시 네트워크 구성

https://m.blog.naver.com/wideeyed/221087773726

.각VM의 설정에서 네트워크>어댑터1: 어댑터에 브리지(브리지 모드 설정)

.Advanced에서 각 VM의 MAC이 충돌하지 않게 확인(필요시, 변경)

.각VM의 어댑터1을 모두 '브리지 어댑터', '어댑터에 브리지' 등 선택

.PC가 연결된 인터넷 모뎀/라우터 등의 Gateway IP가 192.168.0.1이라고 하면

각VM의 netplan에서 192.168.0.100/101 등으로 설정에서

.ping 8.8.8.8 으로 확인

-각 VM에 MobaXterm으로 SSH 접속하기

-PC에서 VM에 SSH 접속해서 설정하기

.명령어 복사/붙여넣기 목적

●WEB 서버 설정

```
sudo apt update
sudo apt install apache2

sudo systemctl start apache2          #아파치 실행
sudo systemctl status apache2         #상태 체크
sudo systemctl enable apache2         #부팅시 자동실행
sudo systemctl stop apache2
sudo systemctl restart apache2
```

●WAS 서버 설정

```
sudo vi /etc/systemd/system/boardapp.service

[Unit]
Description=WAS for board service
After=network.target

[Service]
User=cloud
Group=www-data
WorkingDirectory=/home/cloud/flask_app
Environment="FLASK_APP=app.py"
```

ExecStart=/usr/bin/python3 -m flask run --host=0.0.0.0 --port=5000

[Install]

WantedBy=multi-user.target

sudo chown -R cloud:www-data /flask_app

sudo chmod -R 755 /flask_app

sudo systemctl daemon-reload

sudo systemctl start boardapp.service

sudo systemctl enable boardapp.service

sudo systemctl status boardapp.service

●DB 서버 설정

sudo systemctl start mariadb

sudo systemctl status mariadb

sudo systemctl enable mariadb