

# 기술보고서

메가톤 미디어 서비스 인프라 고도화 사업

소속 : zeronine IT

작성자 : 김민경

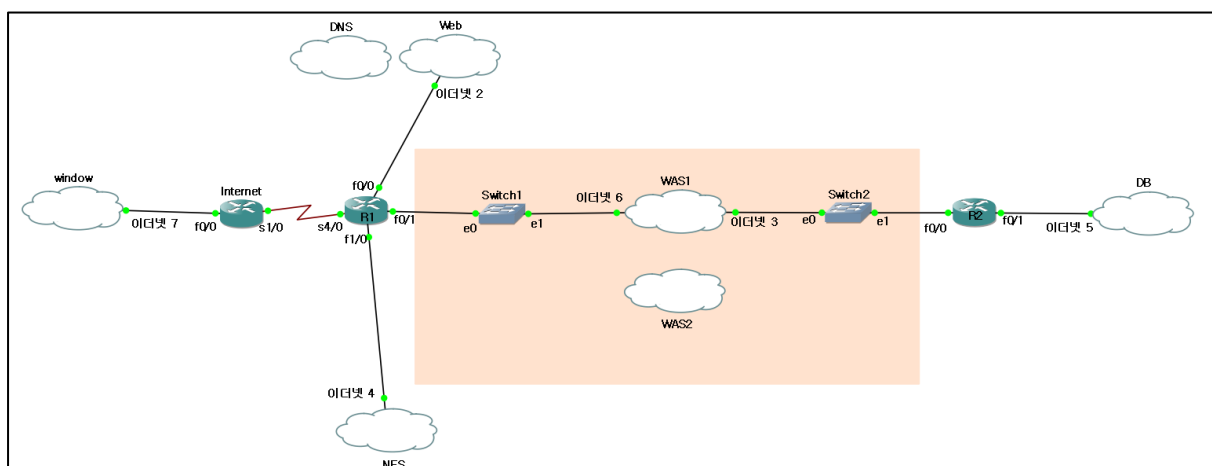
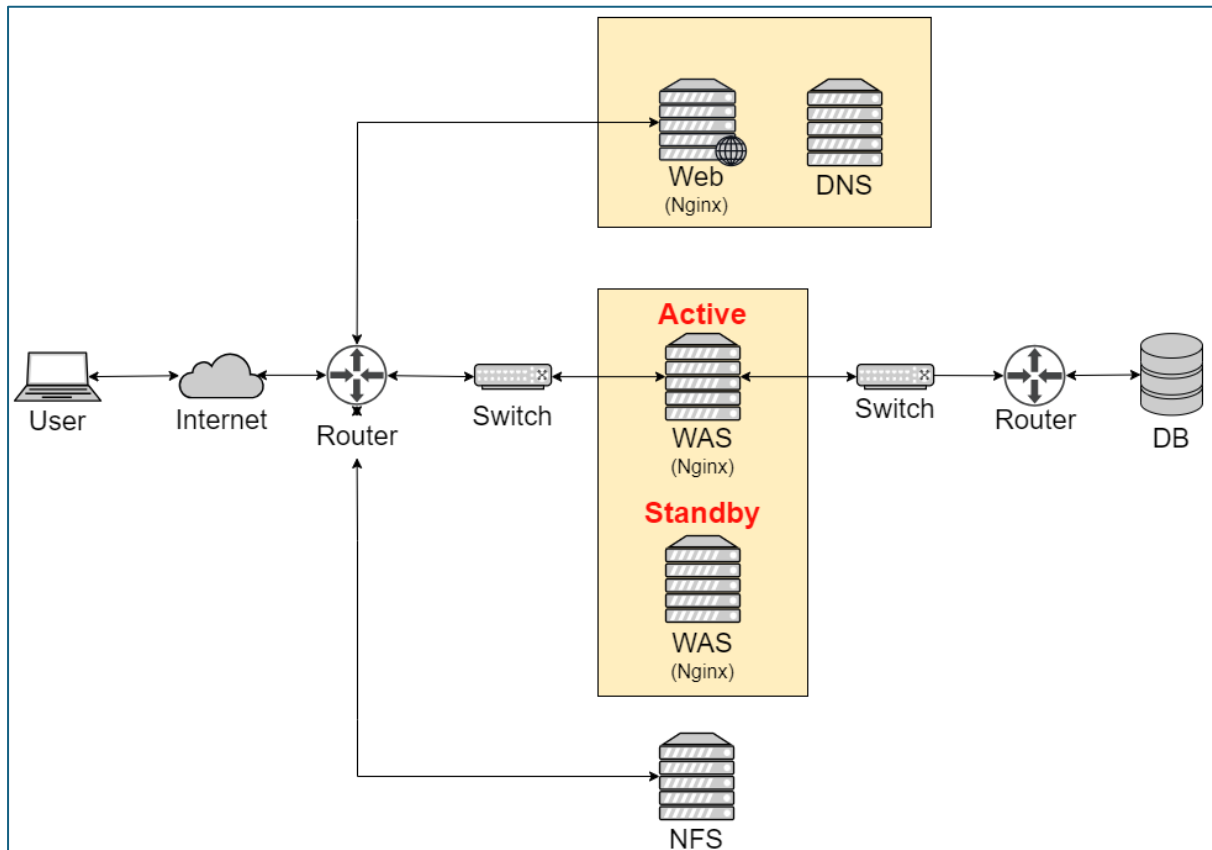
작성일 : 2024-03-18

## 내용

[개인 구성도] .....	3
[개인 시나리오] .....	4
[구현과정] .....	5
1. 3tier 구성하기 .....	5
2. Static route 설정하기 .....	6
3. Web 환경 구성하기 .....	8
4. WAS 환경 구성하기 .....	9
5. Web 프록시 설정하기 .....	11
6. DB 환경 구성하기 .....	12
7. NFS 서버 구성하기 .....	20
8. 연동된 NFS로부터 이미지 가져오기 .....	28
9. DNS 설정하기 .....	31
10. NAT 설정하기 .....	35

\*본 가이드는 ZERONINE IT가 메가톤의 서비스 인프라를 재구축하는데 있어서 필요한 기술과 구축 과정을 포함하고 있습니다.

## [개인 구성도]



# [개인 시나리오]

3Tier로 계층 분리를 통해 기존 사용자들에게 지속적인 서비스를 제공

<b>Web, WAS (Nginx)</b>	<b>Web과 WAS 서버는 Nginx로 구성함</b> <ul style="list-style-type: none"><li>• 더 적은 메모리를 사용하면서 높은 성능을 낼 수 있음</li><li>• 많은 트래픽이 있을 때 동시 접속을 처리하는데 적합한 웹 서버 프로그램임</li><li>• Nginx에 캐시를 적용할 수 있어 서버에 부하를 줄이고 응답속도를 개선하는데 사용될 수 있음</li></ul>
<b>DB (MariaDB)</b>	<b>DB 서버는 MariaDB로 구성함</b> <ul style="list-style-type: none"><li>• MariaDB는 쿼리 최적화, 테이블과 인덱스 관리 등 여러 면에서 최적화되어 있어 빠른 성능을 제공함<ul style="list-style-type: none"><li>→ 웹툰 이미지 경로와 같은 데이터를 빠르게 검색할 수 있음</li></ul></li><li>• 수직 및 수평 확장 모두를 지원하기 때문에 추후에 사용자가 증가하여 DB를 확장해야 할 때 더 편리하게 데이터베이스 확장 가능</li><li>• 오픈 소스이기 때문에 비용 효율적으로 사용 가능</li></ul>

## • WEB의 분리

- 하나의 Web을 Web, WAS 2개로 분리하여 사용자에게 지속적인 서비스를 제공할 수 있도록 구현
- 계층 분리를 통해 독립적인 문제 해결이 가능하고 개발 및 배포가 가능한 환경을 마련

## • NAT 적용

- 클라이언트가 인터넷에 접속하여 라우터 외부로 나갈 때 사설 IP가 공인 IP 주소로 바뀜. 따라서 공격자는 라우터 안쪽에 있는 사설 IP를 잘 모르기 때문에 최종 목적지로의 공격이 어려움. 이는 내부 네트워크를 보호할 수 있는 보안 솔루션 중 하나임.

# [구현과정]

## 1. 3tier 구성하기

### 1-1. 서버 ip 설정하기

- web의 netplan 설정하기

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.56.101/24
      gateway4: 192.168.56.254
      dhcp4: no
  version: 2
```

- was의 netplan 설정하기

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses: [192.168.66.101/24]
      gateway4: 192.168.66.254
      dhcp4: no
    enp0s8:
      addresses: [192.168.96.101/24]
      gateway4: 192.168.96.254
      dhcp4: no
  version: 2
```

- db의 netplan 설정하기

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.86.101/24
      gateway4: 192.168.86.254
      dhcp4: no
  version: 2
```

### 1-2. 라우터 ip 할당하기

- R1에서 ip 할당하기

R1(config-if)#do show ip int b						
Interface	IP-Address	OK?	Method	Status	Protocol	
FastEthernet0/0	192.168.56.254	YES	manual	up	up	
FastEthernet0/1	192.168.96.254	YES	manual	up	up	
FastEthernet1/0	192.168.76.254	YES	manual	up	up	
FastEthernet2/0	100.1.1.102	YES	manual	up	up	

- R2에서 ip 할당하기

```
R2(config-if)#do show ip int b
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.66.254	YES	manual	up	up
FastEthernet0/1	192.168.86.254	YES	manual	up	up

- Internet 에서 ip 할당하기

```
Internet(config-if)#do show ip int b
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	10.1.2.254	YES	manual	up	up
FastEthernet0/1	unassigned	YES	manual	administratively down	down
Serial1/0	100.1.1.101	YES	manual	up	up

## 2. Static route 설정하기

### 2-1. Static route 설정하기

- 아직 Web, WAS, DB끼리의 통신이 불가능하기 때문에 라우팅 설정을 해줘야 함

- WEB - WAS ⇒ 실패

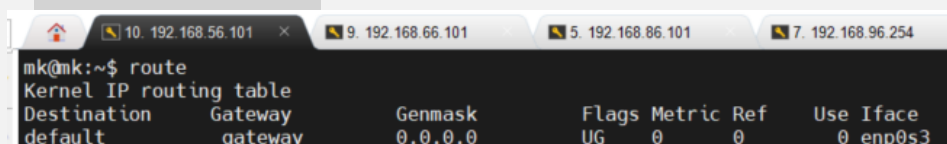
```
mk@mk:~$ ping 192.168.96.101
PING 192.168.96.101 (192.168.96.101) 56(84) bytes of data.
^Z
[2]+  Stopped                  ping 192.168.96.101
mk@mk:~$
```

- WEB -WAS ⇒실패

```
mk@mk:~$ ping 192.168.86.101
PING 192.168.86.101 (192.168.86.101) 56(84) bytes of data.
^C
--- 192.168.86.101 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4091ms
mk@mk:~$
```

### 참고) 리눅스에서 기본 GW 삭제 명령어

- `sudo route del default`



```
mk@mk:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 0 0 0 enp0s3
```

- Web에서 설정하기

```
sudo ip route add 192.168.96.0/24 via 192.168.56.254
```

```
sudo ip route add 100.1.1.0/24 via 192.168.56.254
```

```
sudo ip route add 10.1.2.0/24 via 192.168.56.254
```

- WAS 1에서 설정하기

- WAS 1 & WEB

```
sudo ip route add 192.168.56.0/24 via 192.168.96.254
```

- WAS 1 & DB

```
sudo ip route add 192.168.86.0/24 via 192.168.66.254
```

- WAS 2에서 설정하기

- WAS 2 & NFS

```
sudo ip route add 192.168.76.0/24 via 192.168.96.254
```

- R1에서 설정하기

```
ip route 10.1.2.0 255.255.255.0 100.1.1.101
```

- Internet에서 설정하기

```
ip route 192.168.56.0 255.255.255.0 100.1.1.102
```

## 2-2. ping 통신 확인하기

- WEB & WAS ⇒ 성공

```
mk@mk:~$ ping 192.168.96.101
PING 192.168.96.101 (192.168.96.101) 56(84) bytes of data.
64 bytes from 192.168.96.101: icmp_seq=1 ttl=63 time=21.4 ms
64 bytes from 192.168.96.101: icmp_seq=2 ttl=63 time=12.7 ms
^X64 bytes from 192.168.96.101: icmp_seq=3 ttl=63 time=21.5 ms
[18]+  Stopped                  ping 192.168.96.101
```

- WAS & DB ⇒ 성공

```
mk@mk:~$ ping 192.168.86.101
PING 192.168.86.101 (192.168.86.101) 56(84) bytes of data.
64 bytes from 192.168.86.101: icmp_seq=1 ttl=63 time=23.5 ms
64 bytes from 192.168.86.101: icmp_seq=2 ttl=63 time=15.6 ms
^Z
[21]+  Stopped                  ping 192.168.86.101
mk@mk:~$
```

- Web & DB ⇒ 실패

```
mk@mk:~$ ping 192.168.86.101
ping: connect: Network is unreachable
```

⇒ WEB과 WAS, WAS와 DB끼리의 통신은 가능하지만 WEB과 DB의 통신은 불가능함

### 3. Web 환경 구성하기

#### 3-1. Web에 Nginx 설치하기

```
sudo apt update
```

```
sudo apt-get install nginx
```

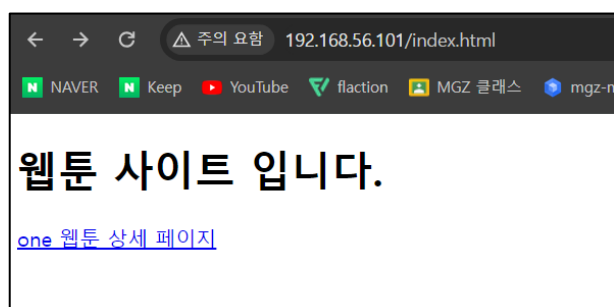
```
sudo service nginx start
```

```
sudo service nginx status
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
mk@mk:~$ sudo service nginx start
mk@mk:~$ sudo service nginx status
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-03-12 01:07:52 UTC; 30s ago
     Docs: man:nginx(8)
  Process: 1830 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;
  Process: 1831 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1925 (nginx)
    Tasks: 3 (limit: 3426)
   Memory: 5.4M
      CPU: 33ms
```

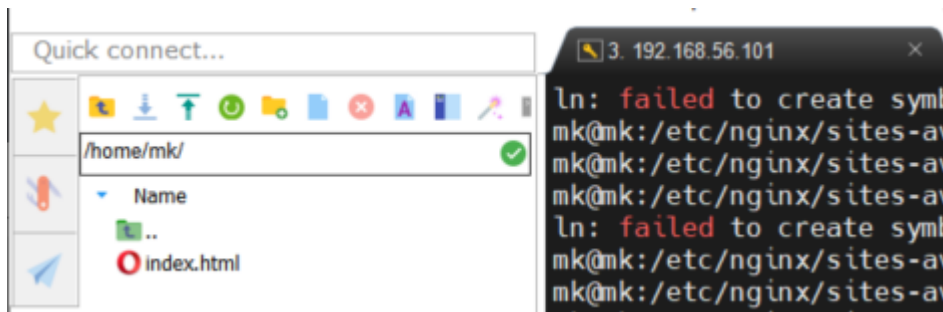
#### 3-2. 정적 페이지 접근

- Real pc에서 vm 내 nginx의 web으로의 접근





- Web으로 파일 옮기기



- 권한 설정하기

```
sudo chmod 777 -R ./index.html
```

- index.html을 /var/www/html로 옮기기

```
sudo mv index.html /var/www/html
```

```
sudo systemctl restart nginx
```

- proxy 설정하기

```
vim /etc/nginx/sites-available/default
```

#### 4. WAS 환경 구성하기

#### 4-1. WAS에 Nginx 설치하기

```
sudo apt update
```

```
sudo apt-get install nginx
```

```
sudo service nginx start
```

```
sudo service nginx status
```

```
Active: active (running) since Tue 2024-03-12 04:30:07 UTC; 27s ago
Docs: man:nginx(8)
Process: 2204 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
Process: 2205 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
Main PID: 2299 (nginx)
Tasks: 3 (limit: 3426)
Memory: 5.3M
CPU: 30ms
CGroup: /system.slice/nginx.service
├─2299 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
├─2302 "nginx: worker process"
└─2303 "nginx: worker process"
```

## 4-2. php 설치하기

```
sudo apt -y install software-properties-common
```

// software-properties-common 패키지 설치 (add-apt-repository와 같은 고급 패키지 관리 기능 제공)

```
sudo add-apt-repository ppa:ondrej/php
```

// ondrej/php : PHP와 관련된 패키지를 제공하는 개인 패키지 아카이브(PPA)

// 이 저장소는 최신 버전의 PHP를 Ubuntu에 제공함

```
sudo apt update
```

```
sudo apt install php7.4 php7.4-fpm php7.4-mysql -y
```

// php7.4, php7.4-fpm (PHP-FPM 서비스), php7.4-mysql (MySQL 지원)을 설치

```
sudo systemctl enable php7.4-fpm
```

// 시스템 부팅 시 PHP-FPM이 자동으로 시작되도록 설정

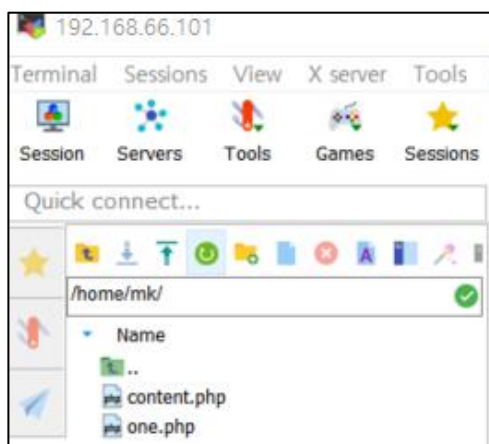
```
sudo systemctl start php7.4-fpm
```

// PHP-FPM 서비스 시작

```
sudo systemctl restart nginx
```

## 4-3. php 파일 위치 이동 및 설정하기

- WAS로 php 파일들 옮기기



- 현재 파일 위치 확인하기

```
mk@mk:~$ ls  
content.php  one.php
```

- php 파일을 현재 위치에서 /var/www/html로 옮기기

```
sudo mv one.php /var/www/html
```

```
sudo mv content.php /var/www/html
```

- 권한 설정하기

```
sudo chmod 777 -R ./one.php
```

```
sudo chmod 777 -R ./content.php
```

## 5. Web 프록시 설정하기

### 5-1. 프록시 설정하기

- Web에서 WAS로의 리버스 프록시 설정하기
  - 리버스 프록시는 로드 밸런서의 역할을 하기도 함
  - 서버를 감추는 효과를 줌

```
vim /etc/nginx/sites-available/default
```

```
server_name 192.168.56.101;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

location /one {
    proxy_pass http://192.168.96.101/one.php;
}

location ~ /contents/(\d+)$ {
    proxy_pass http://192.168.96.101/content.php?id=$1;
}

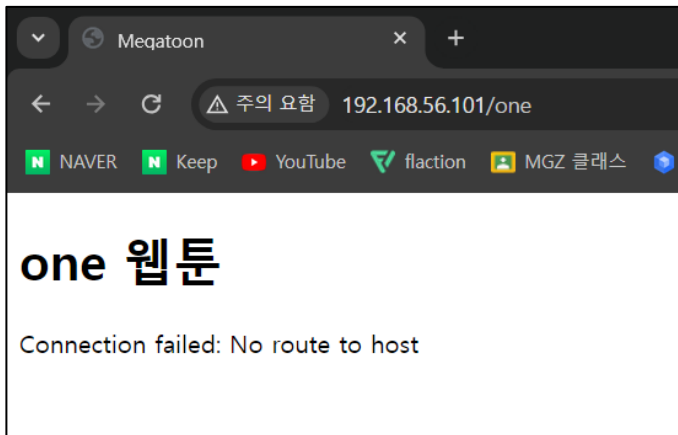
# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
    #
    # With php-cgi (or other tcp sockets):
    fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
```

```
sudo systemctl restart nginx
```

## 5-2. was 연결 확인하기

- was 연동은 성공하였으나, db 연동이 안됨을 확인할 수 있음



## 6. DB 환경 구성하기

### 6-1. DB 서버에 MariaDB 서버 설치하기

```
sudo apt-get upgrade
```

```
sudo apt-get update
```

```
sudo apt install mariadb-server
```

```
sudo apt install mariadb-client
```

```
sudo mysql_secure_installation
```

### 6-2. 접속 대상 설정하기

- 50-server.cnf 파일 수정하여 접속 대상 설정하기

```
sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
#
#user                    = mysql
pid-file                 = /run/mysqld/mysqld.pid
basedir                  = /usr
#datadir                 = /var/lib/mysql
#tmpdir                  = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address             = 0.0.0.0
```

- bind-address = 0.0.0.0은 root 계정으로 접근하는 모든 사용자의 접속을 허가하기

```
systemctl restart mariadb
```

### 6-3. mysql에 접속하기

```
sudo mysql -u root -p
```

```
mk@mk:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04
```

### 6-4. 설정하기

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON webtoon.* TO 'admin'@'%';
```

```
FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> CREATE USER 'admin'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.018 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON webtoon.* TO 'admin'@'%';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

### 6-5. database 생성하기

- 'webtoon'이라는 이름으로 데이터베이스 생성하기

```
create Database webtoon;
```

```
MariaDB [(none)]> create Database webtoon;
Query OK, 1 row affected (0.000 sec)
```

- 'webtoon' 테이블과 'content' 테이블 생성하기

```
Use webtoon;
```

```
MariaDB [(none)]> USE webtoon;
Database changed
```

```
CREATE TABLE webtoon (
    webtoon_id INT PRIMARY KEY,
    genre VARCHAR(50),
    introduce TEXT,
    serial_state VARCHAR(20),
    title VARCHAR(255)
);
```

```
MariaDB [webtoon]> CREATE TABLE webtoon (
->     webtoon_id INT PRIMARY KEY,
->     genre VARCHAR(50),
->     introduce TEXT,
->     serial_state VARCHAR(20),
->     title VARCHAR(255)
-> );
Query OK, 0 rows affected (0.019 sec)
```

```
CREATE TABLE content (
    content_id INT PRIMARY KEY,
    contents TEXT,
    title VARCHAR(255),
    webtoon_id INT,
    FOREIGN KEY (webtoon_id) REFERENCES webtoon(webtoon_id)
);
```

```
MariaDB [webtoon]> CREATE TABLE content (
->     content_id INT PRIMARY KEY,
->     contents TEXT,
->     title VARCHAR(255),
->     webtoon_id INT,
->     FOREIGN KEY (webtoon_id) REFERENCES webtoon(webtoon_id)
-> );
Query OK, 0 rows affected (0.038 sec)
```

- webtoon 테이블에 데이터(정보) 집어넣기

```
INSERT INTO webtoon (
    webtoon_id, genre, introduce, serial_state, title)
VALUES (1, 'modernFantasy', '테스트', 'Live', '테스트');
```

```
MariaDB [webtoon]> INSERT INTO webtoon (
->     webtoon_id, genre, introduce, serial_state, title)
->     VALUES (1, 'modernFantasy', '테스트', 'Live', '테스트');
Query OK, 1 row affected (0.013 sec)
```

- content 테이블에 데이터(정보) 집어넣기

```
INSERT INTO content (
content_id, contents, title, webtoon_id)
VALUES (3, '웹툰 내용', '웹툰 제목', 1);
```

```
MariaDB [webtoon]> INSERT INTO content (
-> content_id, contents, title, webtoon_id)
-> VALUES (3, '웹툰 내용', '웹툰 제목', 1);
Query OK, 1 row affected (0.017 sec)
```

- was에서 심볼릭 링크 생성하기

```
sudo ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/project
```

// '/etc/nginx/sites-available/default' 파일에 대한 심볼릭 링크를 /etc/nginx/sites-enabled/project에 생성하기

// sites-enabled 디렉토리 내에서 project라는 이름으로 심볼릭 링크 생성하기

```
mk@mk:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/project
```

```
root@mk:/etc/nginx/sites-enabled# ll
total 8
drwxr-xr-x 2 root root 4096 Mar 14 07:51 ./
drwxr-xr-x 8 root root 4096 Mar 14 05:11 ../
lrwxrwxrwx 1 root root 34 Mar 14 07:47 project -> /etc/nginx/sites-available/default
```

## 참고)

- 설정을 잘못 만져 default 파일도 심볼릭 링크를 생성하였음.

```
lrwxrwxrwx 1 root root 34 Mar 14 05:11 default -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 34 Mar 14 07:47 project -> /etc/nginx/sites-available/default
```

- 아래의 명령어로 연결 삭제하기

```
sudo rm /etc/nginx/sites-enabled/default
```

```
sudo nginx -t //nginx 구성 테스트하기
```

```
sudo systemctl restart nginx
```

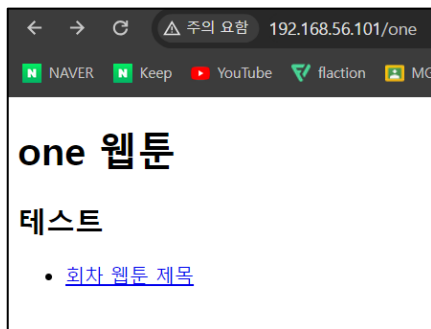
- 현재 권한이 제한적이기 때문에 권한 설정해주기

```
mk@mk:/etc/nginx/sites-available$ ll
total 12
drwxr-xr-x 2 root root 4096 Mar 12 01:07 ./
drwxr-xr-x 8 root root 4096 Mar 12 05:00 ../
-rw-r--r-- 1 root root 2412 May 30 2023 default
```

```
sudo chmod 777 -R ./default
```

```
mk@mk:/etc/nginx/sites-available$ ll
total 12
drwxr-xr-x 2 root root 4096 Mar 12 01:07 ./
drwxr-xr-x 8 root root 4096 Mar 12 05:00 ../
-rwxrwxrwx 1 root root 2412 May 30 2023 default*
```

- 결과 확인하기



⇒ 계속 504 Gateway Time-out 에러가 발생함. 이는 프록시 설정이 잘못 설정되어 있어 생기는 문제임. 프록시 설정을 올바르게 하면 해결됨



## 참고) 같은 네트워크 상에서의 3tier 구축

### 1) 개요 및 환경설정

- 기존의 인프라 환경은 web, was, db가 서로 다른 네트워크 상에 있었음
- 하지만 web, was, db가 연동이 되었다가 504 Gateway Time-out 에러가 발생하기도 하는 현상이 반복적으로 일어났음
- 따라서 같은 네트워크 상으로 설정하여 실습을 진행해 봄
- 조건) gns 사용하지 않고, 서버 3개 모두 동일한 어댑터를 사용해서 3tier 구축하기
  - 다음과 같이 ip 할당받음
    - **Web** : 192.168.56.101/24
    - **WAS** : 192.168.56.125/24 (enp0s3)  
192.168.56.120/24 (enp0s8)
    - **DB** : 192.168.56.121/24

### 2) 실습 과정

- Web-WAS, WAS-DB끼리의 ping 통신 확인하기
- 프록시 설정 수정하기

**vim /etc/nginx/sites-available/default**

```
index index.html index.htm index.nginx-debian.html;

server_name 192.168.56.101;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

location /one {
    proxy_pass http://192.168.56.120/one.php;
}

location ~ /contents/(\d+)$ {
    proxy_pass http://192.168.56.120/content.php?id=$1;
}
```

- WAS 서버에 있는 one.php, content.php 파일 설정 수정하기

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Megatoon</title>
</head>
<body>
  <h1>one 웹 톤 </h1>
  <?php
    // DB 연결 설정
    $servername = "192.168.56.121";
    $username = "admin";
    $password = "password";
    $dbname = "webtoon";
  </?php>

```

- was에서 심볼릭 링크 생성하기

```
sudo ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/project2
```

```

root@mk:/etc/nginx/sites-enabled# ll
total 8
drwxr-xr-x 2 root root 4096 Mar 14 20:31 ./
drwxr-xr-x 8 root root 4096 Mar 14 05:11 ../
lrwxrwxrwx 1 root root 34 Mar 14 07:47 project -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 34 Mar 14 20:31 project2 -> /etc/nginx/sites-available/default

```

- 이렇게 되면 심볼릭링크가 2개이기 때문에 nginx 구성 테스트를 하면 fail이 뜬

```

root@mk:/etc/nginx/sites-enabled# sudo nginx -t
nginx: [emerg] a duplicate default server for 0.0.0.0:80 in /etc/nginx/sites-enabled:22
nginx: configuration file /etc/nginx/nginx.conf test failed

```

- 따라서 잠시 심볼릭 링크를 삭제하지 않고 비활성화해주기. 다른 디렉토리로 위치 이동을 하여 NGINX가 더 이상 그 설정을 인식하지 못하게 되므로 서버 구성에서 무시하도록 설정

- 새로운 디렉토리 생성하기

```
mkdir /etc/nginx/sites-disabled
```

- 새로운 디렉토리에 project 심볼릭 링크 옮겨주기

```
mv /etc/nginx/sites-enabled/project /etc/nginx/sites-disabled/
```

- 위치 이동 확인하기

```

root@mk:/etc/nginx/sites-disabled# ll
total 8
drwxr-xr-x 2 root root 4096 Mar 14 20:33 ./
drwxr-xr-x 9 root root 4096 Mar 14 20:33 ../
lrwxrwxrwx 1 root root 34 Mar 14 07:47 project -> /etc/nginx/sites-available/default

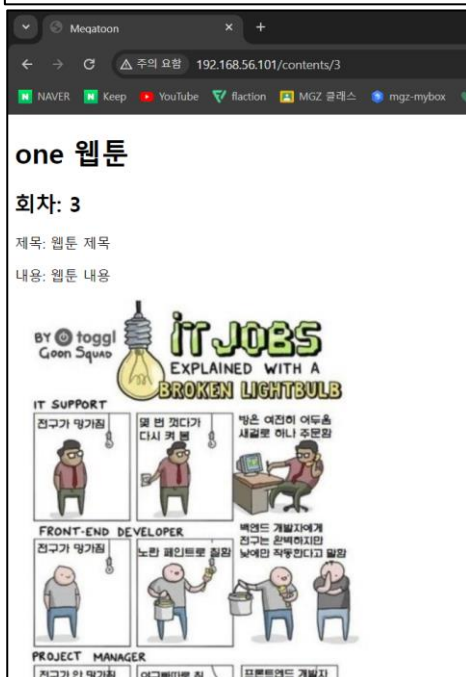
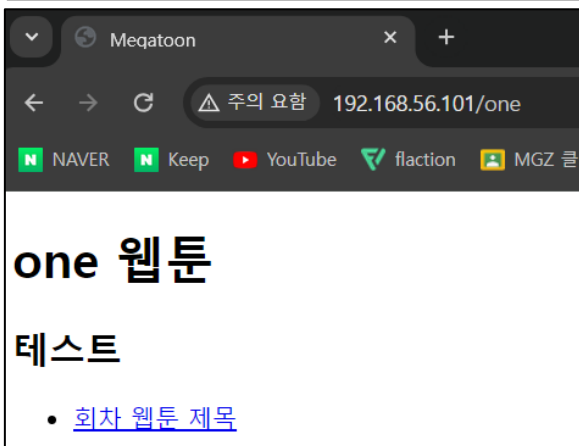
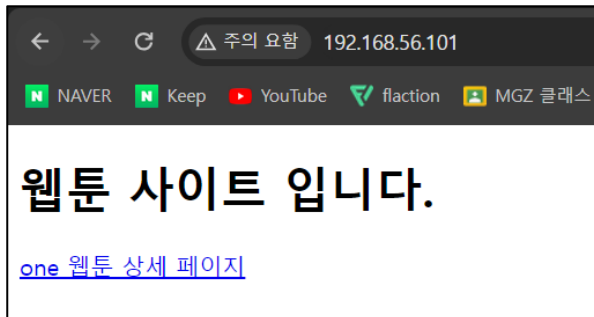
```

- 위치 이동 후 nginx 구성 다시 검사하고 서비스 재시작하기

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

- 새로 고침해서 계속 접속해보아도 504 gateway time-out 에러는 뜨지 않음을 확인할 수 있음



## 7. NFS 서버 구성하기

### 참고) NFS

- 네트워크 상에 연결된 다른 컴퓨터의 하드 디스크를 내 컴퓨터의 하드처럼 사용하는 것
  - 네트워크에 연결된 리눅스끼리 NFS를 통해 파일 공유 가능
  - 공통으로 사용되는 파일이 있으면 각 컴퓨터에 파일을 저장하는 것이 아니라 서버 1개에만 저장하여 효율적으로 사용 가능
- ⇒ 공간을 줄이고 효율적인 사용 가능

### 7-1. NFS 서버 환경설정하기

- ip 할당하기

```
root@mk:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_c
    link/ether 08:00:27:4c:0c:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.76.101/24 brd 192.168.76.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::200:27ff:fe4c:9898/64 scope link
```

- WAS & NFS의 연동 위해 WAS에서 static route 설정하기

```
sudo ip add 192.168.76.0/24 via 192.168.96.254
```

### 7-2. NFS 설치 및 부팅 시 자동 실행 설정하기

- NFS 설치하기

```
sudo apt-get update
```

```
sudo apt-get install nfs-kernel-server
```

```
cat /proc/fs/nfsd/versions
```

```
root@mk:~# cat /proc/fs/nfsd/versions
-2 +3 +4 +4.1 +4.2
```

- 부팅 시 자동 실행 설정하기

```
sudo systemctl is-enabled nfs-server
```

```
mk@mk:~$ sudo systemctl is-enabled nfs-server
enabled
```

**sudo systemctl status nfs-server** //활성화 확인하기

```
mk@mk:~$ sudo systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2024-03-15 04:18:19 UTC; 1h 28min ago
     Main PID: 2449 (code=exited, status=0/SUCCESS)
        CPU: 5ms

Mar 15 04:18:18 mk systemd[1]: Starting NFS server and services...
Mar 15 04:18:18 mk exportfs[2448]: exportfs: can't open /etc/exports for reading
Mar 15 04:18:19 mk systemd[1]: Finished NFS server and services.
```

### 7-3. NFS 서버 글로벌 루트 디렉토리 설정하기

#### 참고) 글로벌 루트 디렉토리

- exported directory
  - 네트워크를 통해 공유하려는 파일 시스템의 루트 디렉토리
- 
- NFS 서버에 공유 디렉토리 생성하기

**sudo mkdir -p /srv/backups**

// '/srv/backups' 디렉토리 생성하기

// -p : 필요한 경우 상위 디렉토리 함께 생성하기

- 권한 설정하기

**sudo chown -R nobody:nogroup /srv/backups**

// '/srv/backups'와 '/mnt/sared' 디렉토리 소유권을 사용자 nobody와 그룹 nogroup으로 변경

//-R : 해당 디렉토리 및 그 안에 포함된 모든 파일과 하위 디렉토리에 대해 변경 적용하기

// 보통 파일 공유나 특정 서비스에서 익명 접근 허용 시 사용됨

```
mk@mk:~$ cd /srv
mk@mk:/srv$ ll
total 12
drwxr-xr-x  3 root  root   4096 Mar 15 05:50 ./
drwxr-xr-x 20 root  root   4096 Mar 11 04:16 ../
drwxr-xr-x  2 nobody nogroup 4096 Mar 15 05:50 backups/
```

## **sudo chmod 777 /srv/backups**

// '/srv/backups'와 '/mnt/shared'의 읽기, 쓰기, 실행 권한 모두 제공하기

```
mk@mk:/srv$ ll
total 12
drwxr-xr-x 3 root root 4096 Mar 15 05:50 ./
drwxr-xr-x 20 root root 4096 Mar 11 04:16 ../
drwxrwxrwx 2 nobody nogroup 4096 Mar 15 05:50 backups/
```

## **sudo vim /etc/exports**

// '/srv/backups' 디렉토리는 IP 주소가 "192.168.96.101"인 NFS 클라이언트에서만 사용 가능

// rw : 공유 디렉토리에 대한 읽기 및 쓰기 활성화하기

// syn : 모든 데이터가 디스크에 즉시 기록됨. 클라이언트의 요청이 처리되기 전에 변경 사항을 기록하고 클라이언트에게 항상 o 제공되도록 함

// no\_subtree\_check : 하위 트리 검사(파일이 이동되었거나 삭제된 후에도 여전히 올바른 파일을 가리키고 있는지 확인)를 비활성화하고, 파일 이름이 변경되더라도 클라이언트가 예전 위치에서 파일에 접근하려고 할 때 발생할 수 있는 문제를 막아주기

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4       gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
/srv/backups     192.168.96.101(rw,sync,no_subtree_check)
```

- 적용하기

## **sudo exportfs -a**

- NFS 다시 시작 및 활성화 확인하기

## **sudo systemctl status nfs-server**

- 공유 가능한 공유 디렉토리 확인하기

## **sudo exportfs -v**

```
mk@mk:~$ sudo exportfs -v
/srv/backups 192.168.96.101(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
mk@mk:~$
```

#### 7-4. 방화벽으로 NFS 서버 보호하기

- 각 공유 디렉터리는 특정 클라이언트 IP 주소와 네트워크를 통해 액세스 가능함

따라서 NFS 서비스 포트에 액세스할 수 있는 UFW 방화벽 규칙에서 클라이언트 IP 주소 또는 네트워크를 지정해줄 필요가 있음

```
sudo ufw allow from 192.168.96.101 to any port nfs
```

```
mk@mk:~$ sudo ufw allow from 192.168.96.101 to any port nfs
Rules updated
```

- 방화벽 규칙 다시 로드하기

```
sudo ufw reload
```

```
sudo ufw status
```

```
root@mk:~# sudo ufw status
Status: inactive
```

- 하지만 inactive라고 뜸

⇒ 이는 UFW가 활성화되지 않아 생기는 문제임°

아래와 같이 활성화시키고 다시 해보기

```
sudo ufw enable
```

```
mk@mk:~$ sudo ufw status
[sudo] password for mk:
Status: active

To Action From
--
2049 ALLOW 192.168.96.101
22 ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
```

° NFS 포트번호: 2049

#### 7-5. NFS 클라이언트 설치 및 구성하기

- nfs-common 패키지 설치하기

참고) nfs-common 패키지

- NFS 클라이언트 시스템에 필요한 클라이언트 사이드 유틸리티들을 포함하고 있음
- NFS 공유를 마운트하고 사용할 수 있게 해주는 필수 파일과 데몬이 포함되어 있음

```
sudo apt update
```

```
sudo apt install nfs-common
```

## 7-6. 마운트 디렉토리 설정하기

- NFS 서버에 생성한 '/srv/backups' 디렉토리와 공유할 디렉토리를 NFS 클라이언트(WAS 서버)에 생성하기

```
sudo mkdir -p /mnt/data
```

```
sudo mount 192.168.76.101:/srv/backups /mnt/data
```

```
sudo df -h
```

- 마운트 된 디스크 목록을 확인할 수 있음 (/srv/backups와 /mnt/data 마운트 확인)

```
mk@mk:~$ sudo df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	297M	1.2M	296M	1%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	12G	5.3G	5.4G	50%	/
tmpfs	1.5G	0	1.5G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	2.0G	129M	1.7G	8%	/boot
tmpfs	297M	4.0K	297M	1%	/run/user/1000
192.168.76.101:/srv/backups	12G	5.2G	5.5G	49%	/mnt/data

- NFS 공유 디렉토리에 대한 쓰기 액세스 확인하기

- WAS에서 쓰기 액세스 확인하기

```
echo "Hello. I am an NFS client." > /mnt/data/test.txt
```

```
cat /mnt/data/test.txt
```

```
mk@mk:~$ echo "Hello. I am an NFS client." > /mnt/data/test.txt
mk@mk:~$
mk@mk:~$ cat /mnt/data/test.txt
Hello. I am an NFS client.
```

- NFS 서버에서 확인하기

```
ls /srv/backups/
```

```
cat /srv/backups/test.txt
```

```
root@mk:~# ls /srv/backups/
test.txt
root@mk:~# cat /srv/backups/test.txt
Hello. I am an NFS client.
```



## 7-7. NFS 서버 자동 마운트 설정하기

- 기본적으로 NFS 마운트는 시스템을 재부팅 하면 사라짐

⇒ 재부팅 후에도 마운트를 유지하기 위해 마운트 정보를 /etc/fstab 파일에 추가하기

- /etc/fstab 파일 편집하기 전 /mnt/data에서 마운트 해제하기

```
sudo umount /mnt/data
```

- 하지만 mount 해제 실패

```
mk@mk:/mnt/data$ sudo umount /mnt/data
umount.nfs4: /mnt/data: device is busy
```

⇒ 이는 해당 마운트 포인트(/mnt/data)가 현재 사용 중이기 때문

현재 실행 중인 프로세스를 확인하고 지운 후 다시 시도하기

- 실행 중인 프로세스 확인 및 삭제하기

```
sudo lsof /mnt/data
```

```
mk@mk:/mnt/data$ sudo lsof /mnt/data
COMMAND  PID USER  FD   TYPE DEVICE SIZE/OFF  NODE NAME
bash      4407 mk    cwd   DIR   0,50    4096 526272 /mnt/data (192.168.76.101:/srv/backups)
sudo     4736 root   cwd   DIR   0,50    4096 526272 /mnt/data (192.168.76.101:/srv/backups)
sudo     4737 root   cwd   DIR   0,50    4096 526272 /mnt/data (192.168.76.101:/srv/backups)
lsof      4738 root   cwd   DIR   0,50    4096 526272 /mnt/data (192.168.76.101:/srv/backups)
lsof      4739 root   cwd   DIR   0,50    4096 526272 /mnt/data (192.168.76.101:/srv/backups)
```

```
sudo kill -9 {PID}
```

```
sudo lsof /mnt/data
```

```
mk@mk:~$ sudo lsof /mnt/data
mk@mk:~$
```

- 실행 중인 프로세스가 없다는 것을 확인함

- 다시 /mnt/data에서 마운트 해제하기

```
sudo umount /mnt/data
```

- 시스템 부팅 시 자동 마운트 설정하기

**sudo nano /etc/fstab**

```
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
192.168.76.101:/srv/backups /mnt/data nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

// nfs : 파일 시스템 유형으로 nfs를 사용함

// auto : 시스템이 부팅 시 자동 마운트 설정

// nofail : 마운트 실패 시에도 부팅 계속 진행

// noatime : 파일 시스템에 접근할 때마다 접근 시간(atime)을 업데이트하지 않음으로써 성능 향상시키기

// nolock : NFS 클라이언트의 잠금 기능을 비활성화

#### 참고) 잠금 기능

- 여러 클라이언트 또는 프로세스가 동시에 동일한 파일에 접근할 때, 파일 잠금을 통해 한 번에 하나의 프로세스만 파일을 수정할 수 있도록 제어

// intr : 마운트된 파일 시스템에 대한 요청을 인터럽트할 수 있게 해 줌.

#### 참고) 마운트된 파일 시스템에 대한 요청을 인터럽트

- 사용자가 파일 시스템에 대한 요청(예를 들어, 파일 읽기 또는 쓰기 작업)을 중단하거나 취소할 수 있음을 의미함

- 예를 들어, NFS를 통해 대용량 파일을 복사하고 있는데 네트워크 지연이 발생하여 복사 프로세스가 멈춘 상태라고 가정.

이 때, 인터럽트 기능이 활성화되어 있으면, 사용자는 해당 복사 명령을 중단하고 시스템에 다른 명령을 내릴 수 있음

// tcp : NFS 연결에 TCP 프로토콜 사용

// actimeo=1800 : 속성 캐시 타임아웃을 1800초로 설정함

#### 참고) 속성 캐시 타임 아웃

- 파일이나 디렉토리의 속성 정보(예: 권한, 소유자, 크기 등)를 클라이언트 측에서 캐싱하는 시간을 설정

- 여기서 1800초는 클라이언트 시스템이 해당 정보를 1800초(30분) 동안 캐시하고 유지하겠다는 것을 의미

// 0 : 덤프(dump) 유틸리티가 이 파일 시스템을 덤프하지 않아야 함을 나타냅니다.

// 0 : 파일 시스템의 fsck(파일 시스템 체크) 우선 순위를 0으로 설정함

#### 참고) fsck

- 파일 시스템의 체크 우선 순위를 나타냄
- 0 : 부팅 시 그 파일 시스템을 체크하지 x
- 1 : 파일 시스템을 가장 먼저 체크함
- 2 or 3 ... : 해당 파일 시스템의 체크 순서를 나타냄

- 아래의 마운트 명령을 실행해 /etc/fstab 파일 확인하기 (구성이 올바르면 모든 /etc/fstab 파일 시스템이 마운트됨)

#### sudo mount -a

```
mk@mk:~$ sudo mount -a
mk@mk:~$
```

- 마운트 된 디스크 목록 확인하기

#### sudo df -h

```
mk@mk:~$ sudo df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     297M        1.2M  296M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 12G       5.3G   5.4G  50% /
tmpfs                     1.5G         0   1.5G   0% /dev/shm
tmpfs                     5.0M         0   5.0M   0% /run/lock
/dev/sda2                  2.0G      129M   1.7G   8% /boot
tmpfs                     297M        4.0K  297M   1% /run/user/1000
192.168.76.101:/srv/backups 12G       5.3G   5.5G  49% /mnt/data
```

- 다시 시작하거나 재부팅하면 NFS 공유 디렉토리가 자동으로 /mnt/data 디렉토리에 마운트 됨

#### 참고) dump 유틸리티

- 리눅스 시스템에서 전통적으로 사용되는 백업 도구
- 0 : 파일 시스템에 대한 자동 백업을 수행하지 않겠다는 것을 dump 프로그램에 알리는 것
- 1 : dump 유틸리티가 이 파일 시스템을 백업해야 함을 나타냄

## 7-8. test.txt 변동 후 연동 확인하기

- NFS 서버에서 test.txt 내용을 아래와 같이 'client 1'으로 변경함

```
2. 192.168.56. 3. 192.168.66. 4. 192.168.86. 5. 192.168.76.  
GNU nano 6.2 test.txt  
Hello. I am an NFS client1.
```

- NFS 클라이언트(WAS)에서도 test.txt 내용이 변경되었는지 확인하기 ⇒ 변경됨

```
2. 192.168.56. 3. 192.168.66. 4. 192.168.86. 5. 192.168.76.  
GNU nano 6.2 test.txt  
Hello. I am an NFS client1.
```

## 8. 연동된 NFS로부터 이미지 가져오기

### 8-1. NFS 서버에 이미지 넣기

- NFS 클라이언트와 마운트 된 파일에 웹툰 넣기

```
root@mk:/srv/backups# ls  
sea.jpg test.txt
```

### 8-2. NFS 클라이언트에서도 이미지 여부 확인하기

- NFS 서버와 연동된 NFS 클라이언트의 디렉토리에 이미지 생성되었는지 확인하기

```
mk@mk:/mnt/data$ ls  
sea.jpg test.txt
```

⇒ 연동되어 자동으로 이미지가 첨가됨을 확인할 수 있음

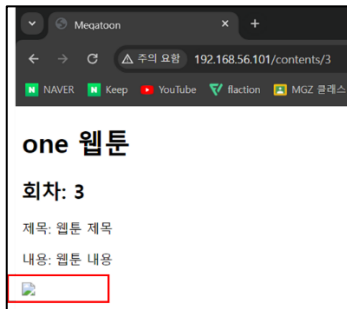
### 8-3. WAS와 연동된 NFS 서버로부터 이미지 가져오기

- WAS 서버의 content.php에서 nfs와 마운트된 was 디렉토리의 이미지 불러오기

- 아래와 같이 경로를 수정함

```
// 데이터 베 이스 연결 종료  
$conn->close();  
?>  
<img src='http://192.168.96.101/mnt/data/sea.jpg' alt='웹툰 이미지'>
```

- 하지만 이미지가 불러오지 않는 문제가 발생함



⇒ nginx는 /var/www/html 디렉토리만 관련하여 참조하기 때문임. 비록 /mnt/data라는 디렉토리가 nginx 안에 있는 디렉토리이지만, nginx는 기본적으로 /var/www/html 디렉토리를 참조하도록 설정되어 있음. 따라서 기존의 /mnt/data 디렉토리와 마운트 설정한 것을 삭제한 후에 /var/www/html 디렉토리와 NFS 서버의 /srv/backups 파일을 다시 마운트하기

#### 8-4. 마운트 재설정하기

- WAS에서 WAS의 /mnt/data 디렉토리 마운트 해제하기

```
sudo umount /mnt/data
```

```
sudo df -h
```

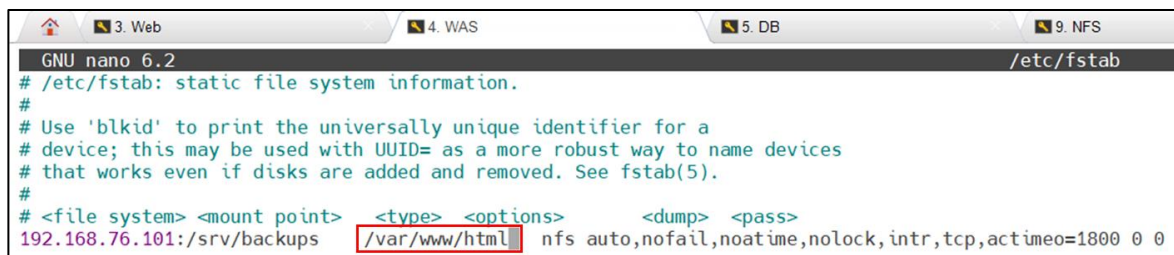
```
mk@mk:/var/www/html$ sudo df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	297M	1.2M	296M	1%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	12G	5.4G	5.4G	51%	/
tmpfs	1.5G	0	1.5G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sda2	2.0G	129M	1.7G	8%	/boot
tmpfs	297M	4.0K	297M	1%	/run/user/1000

- 마운트 한 것이 사라짐을 알 수 있음

- 시스템 부팅 시 자동 마운트 설정하기

```
sudo nano /etc/fstab
```



- /etc/fstab 파일 구성 확인하기

```
sudo mount -a
```

- 마운트 된 디스크 목록 확인하기

```
sudo df -h
```

```
mk@mk:/var/www/html$ sudo df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                      297M        1.2M  296M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 12G       5.4G    5.4G  51% /
tmpfs                      1.5G         0    1.5G   0% /dev/shm
tmpfs                      5.0M         0    5.0M   0% /run/lock
/dev/sda2                  2.0G       129M    1.7G   8% /boot
tmpfs                      297M        4.0K    297M   1% /run/user/1000
192.168.76.101:/srv/backups 12G       5.3G    5.5G  50% /var/www/html
```

- content.php에서 이미지 경로를 다음과 같이 수정하기

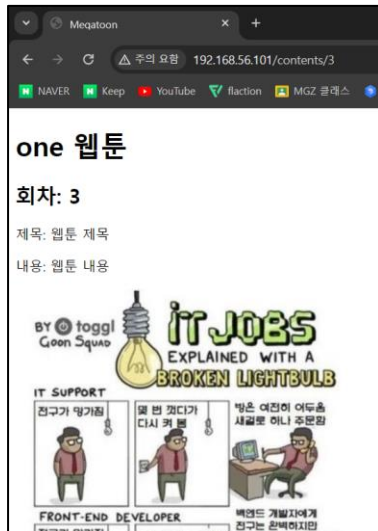
```
// 데이터 베이스 연결 종료
$conn->close();
?>
<img src='http://192.168.96.101/developer.jpg' alt='웹툰 이미지'>
```

- 다시 재시작 후 확인하기

```
sudo systemctl restart nginx
```

- 결과 확인하기

- NFS와 마운트된 NFS 클라이언트의 /var/www/html에서 웹툰 이미지를 성공적으로 불러올 수 있음



## 9. DNS 설정하기

### 9-1. bind9 설치하기

```
sudo apt update
```

```
sudo apt install bind9 -y
```

```
sudo systemctl status bind9
```

```
mk@mk:~$ sudo systemctl status bind9
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-03-18 01:33:36 UTC; 8s ago
     Docs: man:named(8)
  Process: 1791 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 1792 (named)
    Tasks: 6 (limit: 3426)
   Memory: 7.7M
      CPU: 22ms
  CGroup: /system.slice/named.service
          └─1792 /usr/sbin/named -u bind
```

```
sudo vim /etc/bind/named.conf.options
```

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====

    dnssec-validation auto;

    listen-on-v6 { any; };

    listen-on-port 53 { any; };
    allow-query { any; };
    recursion yes;
};
```

- listen-on port 53 { any; }; // 53번 포트에서 모든 인터페이스를 통해 들어오는 요청을 수신하도록 설정하기
  - allow-query { any; } // 누구나 해당 DNS 서버를 쿼리할 수 있음
  - recursion yes; // 재귀적인 쿼리 수락하기
- 
- 정방향, 역방향 도메인 설정하기

```
sudo vim /etc/bind/named.conf.local
```

```
zone "megatoon.com" {
    type master;
    file "/etc/bind/zones/db.megatoon.com";
};

zone "56.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.56.168.192";
};
```



- DNS 서버 관련 설정하기

```
sudo mkdir -p /etc/bind/zones/ // '/etc/bind/zones/' 디렉토리 생성하기
```

```
sudo cp /etc/bind/db.local /etc/bind/zones/db.megatoon.com
```

// 'megatoon.com' 도메인에 대한 DNS 정보를 설정할 수 있는 기본 템플릿 만들기

```
sudo cp /etc/bind/db.0 /etc/bind/zones/db.56.168.192
```

// 역방향 DNS에 대한 정보를 설정할 수 있는 기본 템플릿 만들기

```
sudo vim /etc/bind/zones/db.megatoon.com
```

```
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     megatoon.com. root.megatoon.com. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS      megatoon.com.
@         IN      A       192.168.56.101
```

```
sudo vim /etc/bind/zones/db.56.168.192
```

```
;
; BIND reverse data file for broadcast zone
;
$TTL      604800
@         IN      SOA     megatoon.com. root.megatoon.com. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS      localhost.
101       IN      PTR     megatoon.com.
```

- 수정한 파일들의 문법 오류 확인하기

```
sudo named-checkzone megatoon.com /etc/bind/zones/db.megatoon.com
```

```
sudo named-checkzone megatoon.com /etc/bind/zones/db.56.168.192
```

```
mk@mk:/etc/bind/zones$ sudo named-checkzone megatoon.com /etc/bind/zones/db.megatoon.com
zone megatoon.com/IN: loaded serial 2
OK
mk@mk:/etc/bind/zones$ sudo named-checkzone megatoon.com /etc/bind/zones/db.56.168.192
zone megatoon.com/IN: loaded serial 1
OK
```

- 변경한 설정 적용하기

### sudo systemctl status bind9

```
root@mk:~# sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-03-18 03:49:30 UTC; 10s ago
     Docs: man:named(8)
  Process: 3636 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 3637 (named)
      Tasks: 6 (limit: 3426)
     Memory: 5.7M
        CPU: 20ms
    CGroup: /system.slice/named.service
            └─3637 /usr/sbin/named -u bind
```

- 정방향, 역방향 조회 확인하기

```
root@mk:~# nslookup
> server 192.168.56.121
Default server: 192.168.56.121
Address: 192.168.56.121#53
> megatoon.com
Server:          192.168.56.121
Address:         192.168.56.121#53

Name:   megatoon.com
Address: 192.168.56.101
Name:   megatoon.com
Address: 127.0.0.1
> 192.168.56.101
101.56.168.192.in-addr.arpa    name = megatoon.com.
>
```

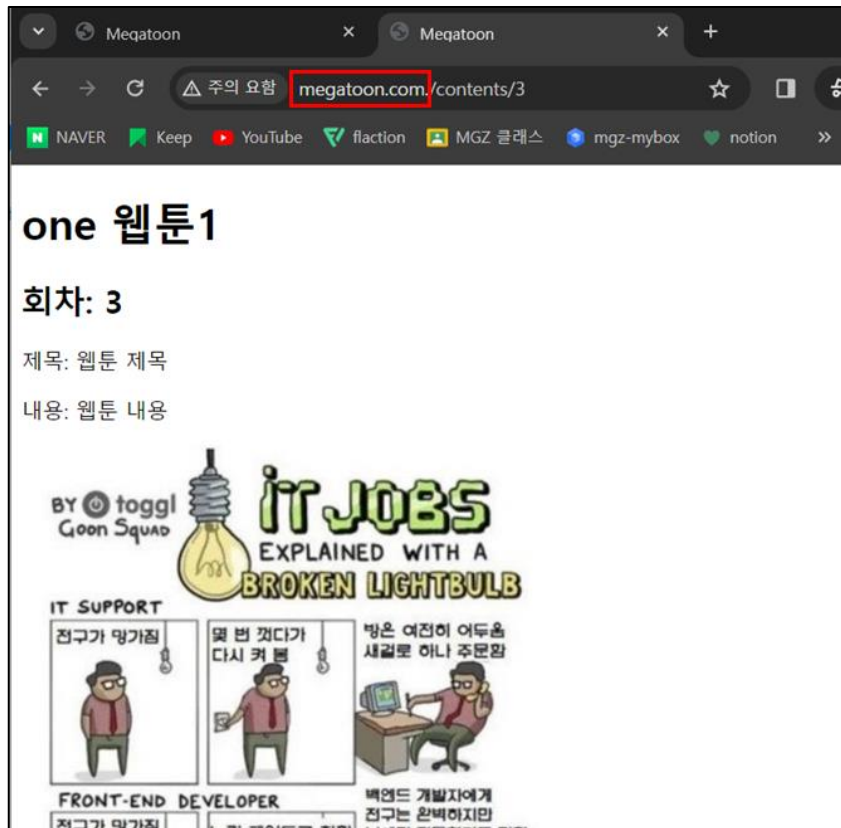
- 도메인으로 접속하기

- 로컬 pc의 DNS 서버를 DHCP 서버 ip 주소로 변경하기

☐ 자동으로 DNS 서버 주소 받기(B)
 ☒ 다음 DNS 서버 주소 사용(E):
 

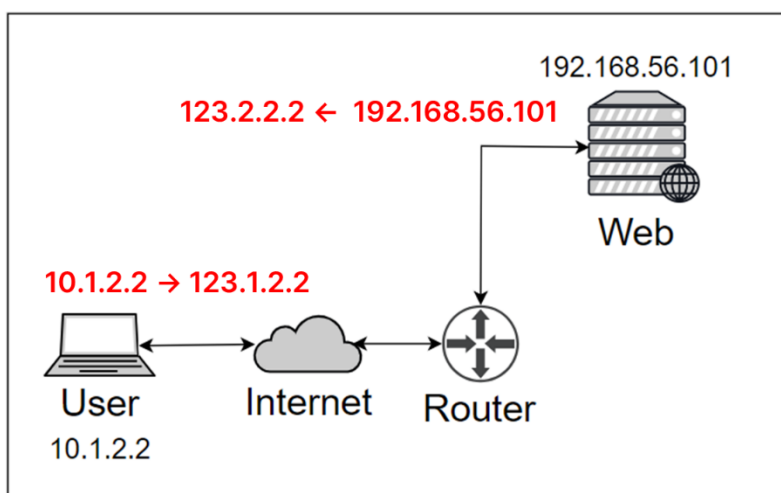
기본 설정 DNS 서버(P): 192 . 168 . 56 . 121

- 결과 확인하기



## 10. NAT 설정하기

### 10-1. [ 사용자 → Internet ] NAT 설정



- ip 변환에 사용할 주소 풀 설정하기

`ip nat inside source static 10.1.2.2 123.1.2.2`

- NAT를 인터페이스에 적용하기

```
int f0/0
```

```
ip nat inside
```

```
int s1/0
```

```
ip nat outside
```

```
Internet(config-if)#do sh ip nat tr
Pro Inside global      Inside local      Outside local      Outside global
--- 123.1.2.2          10.1.1.2.2        ---                ---
```

- 결과 확인하기

```
C:\Windows\system32>ping 100.1.1.102

Ping 100.1.1.102 32바이트 데이터 사용:
100.1.1.102의 응답: 바이트=32 시간=26ms TTL=254
100.1.1.102의 응답: 바이트=32 시간=16ms TTL=254
100.1.1.102의 응답: 바이트=32 시간=22ms TTL=254
100.1.1.102의 응답: 바이트=32 시간=28ms TTL=254

100.1.1.102에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 16ms, 최대 = 28ms, 평균 = 23ms
```

## 10-2. [ Web → R1 ] NAT 설정

- ip 변환에 사용할 주소 풀 설정하기

```
ip nat inside source static 192.168.56.101 123.2.2.2
```

- NAT를 인터페이스에 적용하기

```
int f0/0
```

```
ip nat inside
```

```
int s4/0
```

```
ip nat outside
```

```
R1(config-if)#do sh ip nat tr
Pro Inside global      Inside local      Outside local      Outside global
--- 123.2.2.2          192.168.56.101    ---                ---
```