

# WEB 2Tier - 실습가이드

최초 작성일 : 2024/01/29

최종 제출일 : 2024/01/30

김민경

## 목차

### I. 개념정리-----

#### 1. Tier-----

- 1) 1 Tier-----
- 2) 2 Tier-----
- 3) 3 Tier-----

#### 2. 3Tier-----

### II. 실습 1-----

#### 1. 환경구성-----

#### 2. 1번 실습과정-----

#### 3. 2번 실습과정-----

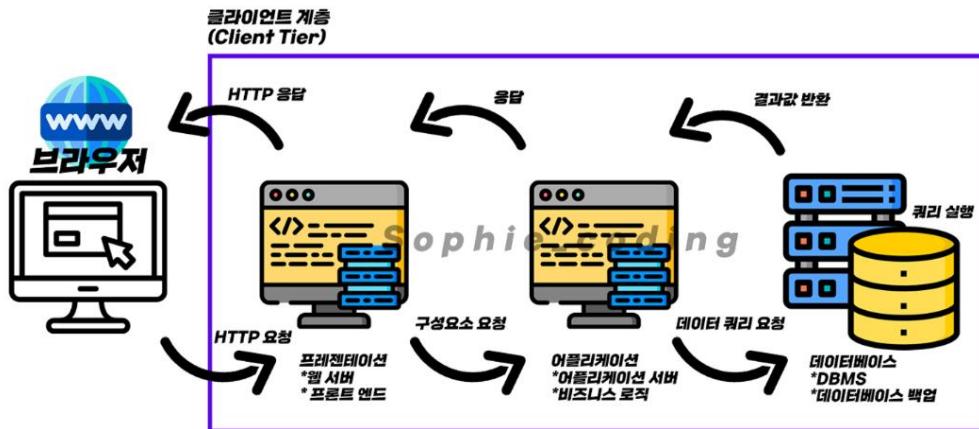
### III. 과제(게시판)-----

#### 1. 실습과정-----

# I. 개념정리

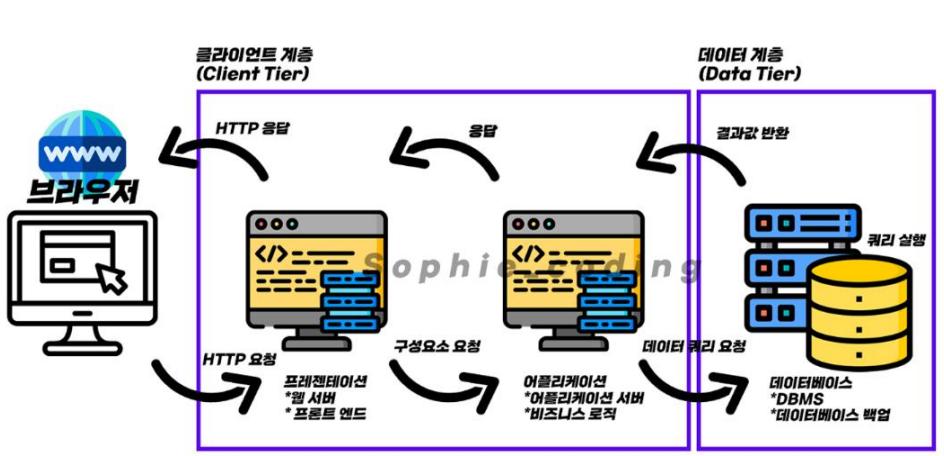
## 1. Tier

### 1 Tier



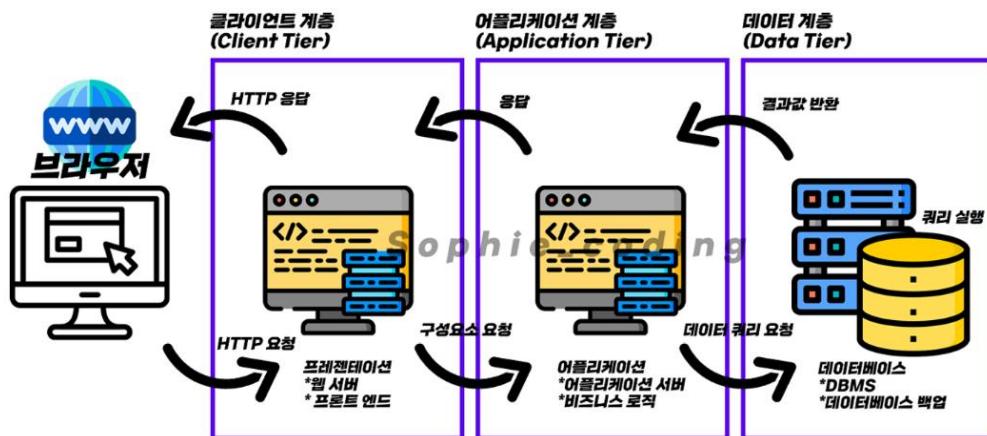
- 하나의 물리적인 컴퓨터 or 서버에 3 가지의 다른 기능으로 함께 구현한 방식
- 물리적인 장비를 새로운 장비로 변경하고자 하는 경우 ⇒ 모든 구성을 함께 변경해야 함

### 2 tier



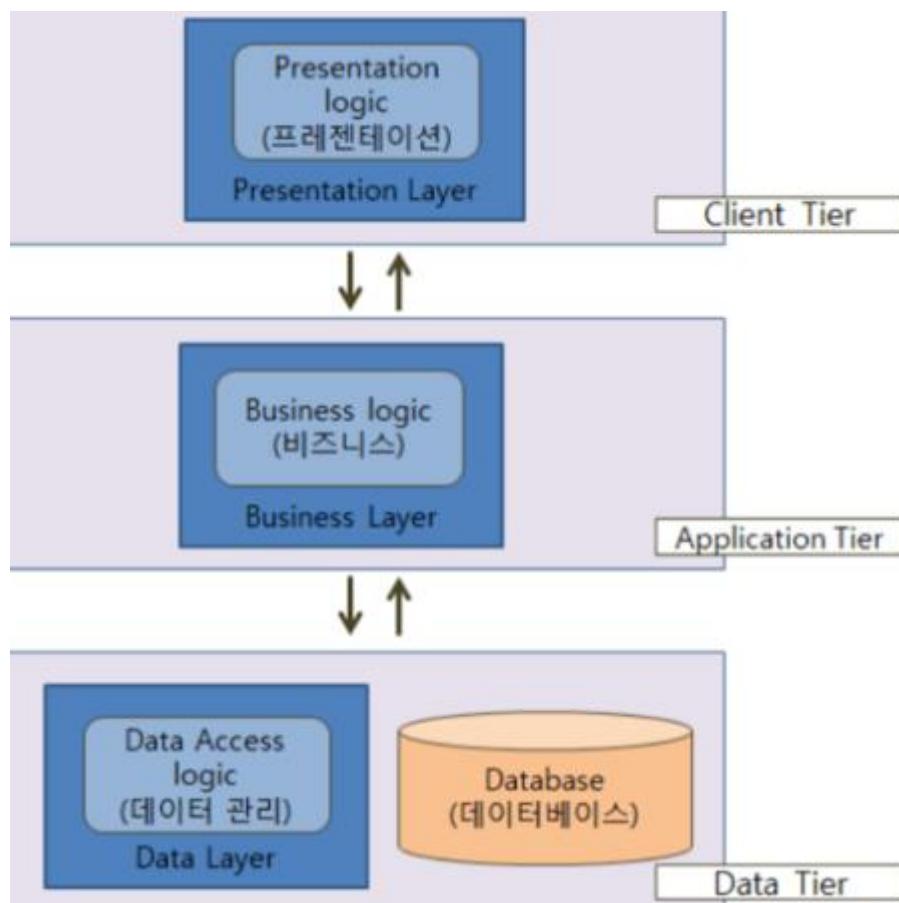
- 클라이언트 계층 & 데이터 계층의 물리적인 컴퓨터 or 서버로 구분
- so 클라이언트 계층에서의 변경 or DB 변경 시 서로 영향을 받지 X

## 3Tier



- 서버를 모두 물리적으로 나누어 구성하는 방식
  - 클라이언트 계층, 어플리케이션 계층, 데이터 계층
- 각각의 계층에서 변화가 일어나더라도 서로 영향받지 x 고, 독립적으로 운영됨

## 2. 3Tier



- 어떠한 플랫폼을 3 계층으로 나누어 별도의 논리적/물리적인 장치에 구축 및 운영하는 형태
- 웹 서버 운영을 예로 들면
  - 서버 한 대에 한꺼번에 모든 기능들을 구축하는 것이 x 닌
  - 데이터를 저장하고 읽는 **데이터 계층**
  - 데이터를 처리하는 **어플리케이션 계층**
  - 데이터를 표현해주는 **클라이언트(프레젠테이션) 계층**
  - 각각 3 계층으로 나누어 각각의 기능으로 별도의 논리적/물리적인 장치에서 운영하는 방식

### **프레젠테이션 계층**

- 사용자가 직접 마주하게 되는 계층
  - 사용자 (클라이언트)가 보려고 하는 GUI, 인터페이스, 웹 화면 등을 제공해주는 서버
- 주로 사용자 인터페이스(인터넷 브라우저 등)를 지원
- 웹 서버
- HTML, Javascript, CSS 등이 이 계층에 해당

### **어플리케이션 계층**

- 프레젠테이션 계층에서 요청되는 정보를 어떠한 규칙을 바탕으로 처리하고 가공하는 것들을 담당 (동적인 데이터 제공)
- 인터넷에서 단순히 인터페이스를 보는 것 이외에 상품 주문, 결제, 검색등의 다양한 기능을 사용함
  - AP 계층은 이러한 기능들을 실질적으로 처리하는 계층
  - 단, 기능 처리 시 회원정보 조회, 결제 정보 입력 등을 위해서는 데이터 조회 등이 필요하므로, 뒷단의 데이터 계층에 업무를 넘기거나 참조를 하는 기능도 같이 병행함
- WAS (Web Application Server) 서버
- 비즈니스 로직 계층 또는 트랜잭션 계층이라고도 함
- 첫 번째 계층에서 이 계층을 바라볼 때에는 서버처럼 동작하고(응답), 세 번째 계층의 프로그램에 대해서는 마치 클라이언트처럼 행동함(요청)
 

⇒ 따라서 미들웨어(Middleware) or 백엔드(back-end)라고도 불림
- PHP, Java 등

## 데이터계층

- 데이터베이스와 데이터베이스에 접근하여 데이터를 읽거나 쓰는 것을 관리하는 것을 포함함
- AP 계층에서 사용자의 요청을 처리할 때, 데이터 기입이나 변경 등이 필요할 시, 이에 대한 작업을 지원하고 사용자들의 데이터들을 보관하는 계층
- DBMS (Database Management System)이 해당
- 백엔드(back-end)라고도 부름
- MySQL, MongoDB 등이 해당됨

## 장점

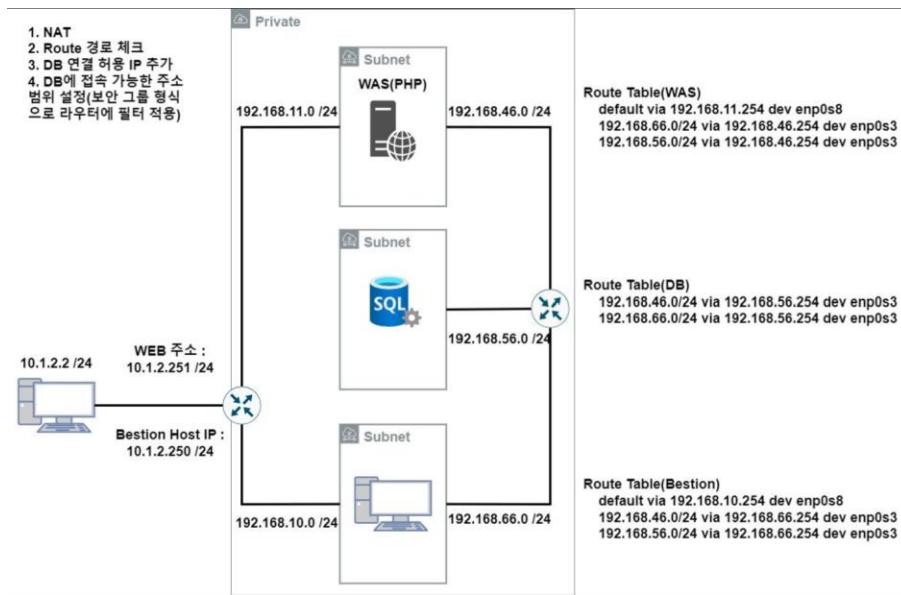
- 각 계층이 분리되어 있어 업무 분담 가능(업무 효율성이 증가)
- 여러 대의 서버로 나누어 각 계층이 동작하므로 서버의 부하를 줄여줄 수도 있음
- 경우에 따라 합리적인 스케일업(서버의 성능 업그레이드)이 가능

## 단점

- 1 계층으로만 사용하는 것 대비 관리가 더 필요함
- 장애가 발생하는 포인트가 더 늘어날 수 있음
- 비용도 그만큼 많이 발생

→ so 서비스 규모 및 사용자 증가에 따라 계층 구조를 설계 및 고려해야 함

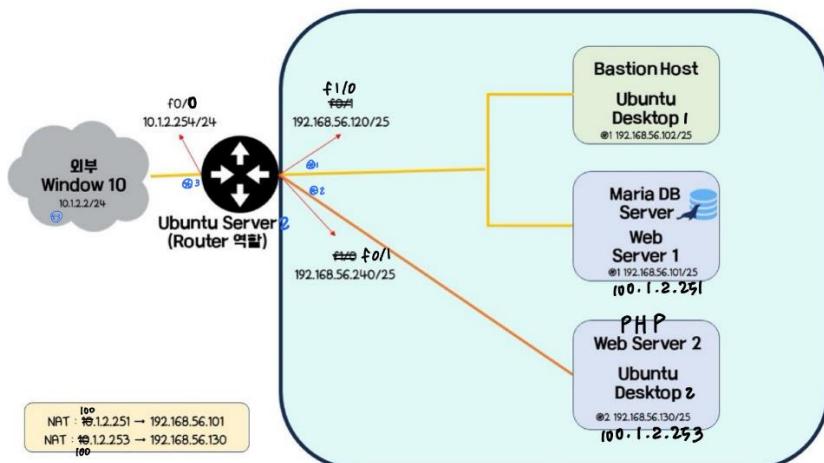
## II. 실습 1



- 위 구성도에서 DB Server에 접근하기 위한 다양한 방법을 실습 진행 합니다.
  - 동일한 사설 네트워크의 서브넷에서 접속하는 방법에 대해 실습 합니다.
  - 동일한 사설 네트워크의 다른 서브넷에서 접속하는 방법에 대해 실습 합니다.(DB에 접속 허용 주소가 계정과 맵핑되어 있습니다.)
  - 외부 네트워크에서 DB Server에 접속하는 방법에 대해 실습을 합니다.
    - NAT 구성으로 직접 DB에 연결
    - NAT로 DB에 직접 연결 하는 것이 아닌 Bastion Host를 사용하여 연결
    - SSH 터널링 기능을 활용하여 DB에 연결

- Web 서버를 추가하여 다음의 실습을 진행합니다.
  - 웹접근시 본인의 사진을 출력하는 화면을 띠워 주세요
  - PHP를 사용하여 DB에 연결 후 DB에 저장된 데이터를 조회 합니다.(필요한 패키지: apache2, php8.1, php8.1-mysql)

## 1. 환경 구성



- 전제조건
  - 이전의 Web 1 Tier 실습 결과물에 이어서 진행하기
  - 또 다른 Ubuntu Server가 라우터 기능을 하게끔 설정하기

## 2. 1번 실습과정

### 1) 환경 설정하기

- ip 설정하기
- 어댑터도 설정하기

호스트 전용 네트워크	NAT 네트워크	클라우드 네트워크
이름	IPv4 접두사	IPv6 접두사
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/25	DHCP 서버 사용 안 함
VirtualBox Host-Only Ethernet Adapter #2	192.168.56.129/25	사용 안 함
VirtualBox Host-Only Ethernet Adapter #3	10.1.2.1/24	사용 안 함

- ubuntu server의 gw도 확인하기

```
Ubuntu-Server1 (스냅샷 1/25) [실행 중] - Oracle VM VirtualBox
파일 마신 보기 입력 장치 도움말
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses: [192.168.56.101/25]
      gateway4: 192.168.56.120
      nameservers:
        addresses: [8.8.8.8]
        dhcp4: no
    enp0s8:
      dhcp4: true
version: 2
```

- U-S2에 어댑터 3개 설정하기

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

```
Ubuntu-Server1 복제(router) (스냅샷 1/25) [실행 중] - Oracle VM VirtualBox
파일 마신 보기 입력 장치 도움말
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses: [192.168.56.120/25]
      dhcp4: no
    enp0s8:
      addresses: [192.168.56.240/25]
      dhcp4: no
    enp0s9:
      addresses: [10.1.2.254/24]
      dhcp4: no
      version: 2
```

## 리눅스에서 라우팅테이블 확인하기

```
netstat -nr // netstat 확인하기
```

```
root@ubuntu-server:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
10.1.2.0        0.0.0.0        255.255.255.0    U        0 0          0 enp0s10
192.168.56.0    0.0.0.0        255.255.255.128 U        0 0          0 enp0s3
192.168.56.128 0.0.0.0        255.255.255.128 U        0 0          0 enp0s8
```

참고) **netstat -nr**

- **netstat** : 네트워크 통계 및 연결 정보를 보여주는 명령어
- **-nr** : 라우팅 테이블을 숫자로 표시

## 2) 서버에서 IP 포워딩 활성화하기

- U-S1에서 /etc/sysctl.conf 파일을 열고 net.ipv4.ip\_forward를 1로 설정하기

```
sudo vim /etc/sysctl.conf
```

```
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```

참고) ip 포워딩 활성화

1. **IP 포워딩 활성화:** IP 포워딩은 패킷이 한 네트워크 인터페이스를 통해 수신되었을 때, 해당 패킷을 다른 네트워크 인터페이스를 통해 전달할 수 있도록 하는 기능입니다. 이 설정을 1로 설정하면 IP 포워딩이 활성화됩니다.

- 변경사항 저장하기

```
sudo sysctl -p
```

```
ubuntu@ubuntu:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
ubuntu@ubuntu:~$ _
```

### 3) FORWARD 체인에 -s(출발지 IP 주소) 패킷 허용하기

#### 참고) FORWARD

##### b) -P FORWARD(INPUT, OUTPUT)

- i) **FORWARD** : 라우터나 다른 네트워크 기기에서 패킷이 다른 인터페이스로 전달될 때 사용되는 체인
- ii) **INPUT** : 시스템으로 들어오는 패킷을 처리. 즉, 시스템에 도착한 패킷에 대한 정책을 결정
- iii) **OUTPUT** : 시스템에서 나가는 패킷을 처리. 시스템에서 생성된 패킷이나 응용 프로그램에서 나가는 요청 처리
- iv) **-P** : 어떤 체인을 적용할 것인지 명시하는 옵션

##### a) iptables를 사용하여 FORWARD 체인에 패킷을 허용하는 규칙을 추가하는 명령어

- i) A FORWARD: FORWARD 체인에 룰을 추가하는 옵션.
  - **-A**: "Append"를 나타내며, 이를 사용하면 규칙이 기존의 뒤에 추가됩니다
- ii) **-s [출발지 IP]**: 출발지 IP 주소를 나타내는 옵션
  - 출발지 주소가 이 범위에 속하는 패킷에 대해 해당 규칙이 적용됨을 의미합니다
- iii) **-j ACCEPT**: -j는 "Jump"를 나타내며, 패킷이 매칭되면 어떤 동작을 취할지를 지정하는 옵션
  - 여기서는 ACCEPT로 지정되어 있어 해당 패킷이 허용됩니다.

• sudo iptables -A FORWARD -s 192.168.56.0/25 -j ACCEPT

• sudo iptables -A FORWARD -s 192.168.56.128/25 -j ACCEPT

• sudo iptables -A FORWARD -s 10.1.2.1/24 -j ACCEPT

```
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.56.0/25 -j ACCEPT
[sudo] password for ubuntu:
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.56.128/25 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 10.1.2.1/24 -j ACCEPT
ubuntu@ubuntu:~$ _
```

- 이 주소들은 각 대역대의 대표 주소들임

#### 참고)

192.168.56.0 / 25



∴ 네트워크는 다음과 같이 나누짐

① 0 ~ 127 ⇒ 192.168.56.0 (대표주소)

192.168.56.128 (브로드캐스트주소)

② 128 ~ 255 ⇒ 192.168.56.128 (대표주소)

192.168.56.255 (브로드캐스트주소)

- 어댑터 주소도 각 ip 범위 내에서

가용 가능한 ip로 줘야됨!

이름	IPv4 접두사
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/25
VirtualBox Host-Only Ethernet Adapter #2	192.168.56.128/25
VirtualBox Host-Only Ethernet Adapter #3	10.1.2.1/24

## 4) NAT 설정하기

### 참고) NAT 설정

i) `iptables -A POSTROUTING -t nat -s 192.168.56.101 -j SNAT --to-source 100.1.2.251`

: 이 규칙은 192.168.56.101에서 나가는 패킷에 대해 출발지 주소를 100.1.2.251로 변경하여 NAT를 수행하는 것입니다. 주로 내부 네트워크에서 외부로 나가는 패킷에 대한 출발지 주소를 변환할 때 사용됩니다.

- **-A POSTROUTING** : POSTROUTING 체인에 룰을 추가하는 옵션, 패킷이 시스템을 떠나는 순간에 적용
- **-t nat** : 특정 테이블을 지정하는 옵션으로, 이 경우 NAT 테이블을 사용하도록 설정
- **-s {내부 IP}** : 출발지 IP 주소를 나타내는 옵션
- **-j SNAT** : 패킷 변환을 수행하는 동작을 지정하는 옵션으로, SNAT (Source NAT)을 수행, 출발지 주소를 변경
- **--to-source {외부 IP}** : SNAT 동작에서 변환될 출발지 IP 주소를 지정하는 옵션

ii) `iptables -A PREROUTING -t nat -d 100.1.2.251 -j DNAT --to-destination 192.168.56.101`

: 이 규칙은 100.1.2.251로 들어오는 패킷에 대해 목적지 주소를 192.168.56.101로 변경하여 NAT를 수행하는 것입니다. 주로 외부에서 내부로 들어오는 패킷에 대한 목적지 주소를 변환할 때 사용됩니다.

- **-A PREROUTING** : PREROUTING 체인에 룰을 추가하는 옵션, 패킷이 시스템에 도착하는 순간에 적용
- **-d {외부 IP}** : 목적지 IP 주소를 나타내는 옵션으로, 이 규칙은 설정한 목적지 IP 주소 패킷에 적용
- **-j DNAT** : 패킷 변환을 수행하는 동작을 지정하는 옵션, DNAT (Destination NAT)을 수행, 목적지 주소를 변경
- **--to-destination {내부 IP}** : DNAT 동작에서 변환될 목적지 IP 주소를 지정하는 옵션

- 192.168.56.101 -> 100.1.2.251 (내부에서 외부)

`iptables -A POSTROUTING -t nat -s 192.168.56.101 -j SNAT --to-source 100.1.2.251`

192.168.56.101 -> 100.1.2.251 (외부에서 내부)

`iptables -A PREROUTING -t nat -d 100.1.2.251 -j DNAT --to-destination 192.168.56.101`

- 192.168.56.130 -> 100.1.2.253 (내부에서 외부)

`iptables -A POSTROUTING -t nat -s 192.168.56.130 -j SNAT --to-source 100.1.2.253`

192.168.56.130 -> 100.1.2.253 (외부에서 내부)

`iptables -A PREROUTING -t nat -d 100.1.2.253 -j DNAT --to-destination 192.168.56.130`

### 참고) iptables

- 방화벽 설정 도구

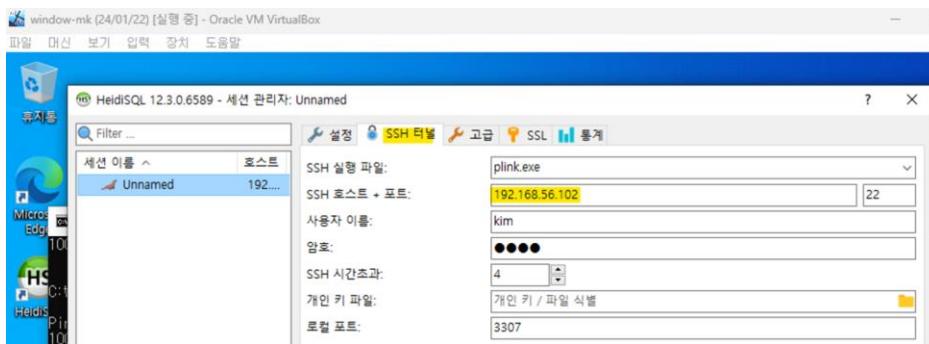
b) NAT

- i) PREROUTING은 DNAT만, POSTROUTING은 SNAT만 사용할 수 있다.
- ii) PREROUTING : 외부 네트워크에서 내부 네트워크로 도착했을 때, 포워딩 하는 역할인데, DNAT으로 외부 주소로 도착한 패킷을 내부 주소로 바꿔야 포워딩이 된다.
- iii) POSTROUTING : 내부 네트워크에서 외부 네트워크로 포워딩 하는 역할인데 출발 주소를 바꾸야 포워딩이 된다.

## 5) 결과 확인하기

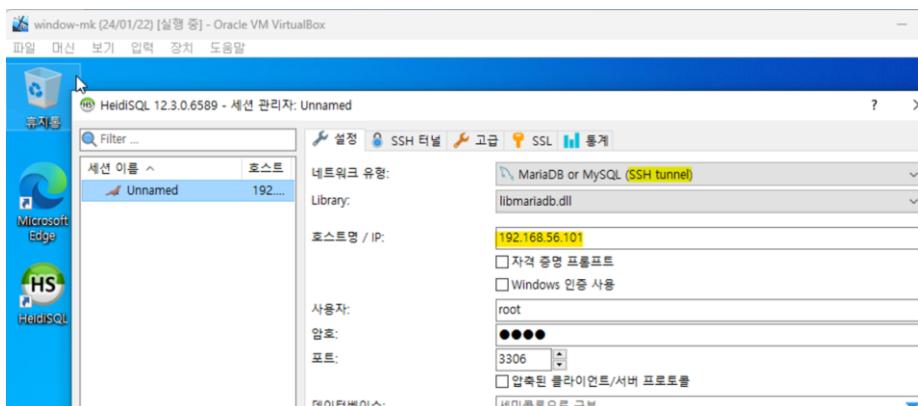
### (1) DB 접근 확인 (Window10 → U-D1(베스천) → U-S1)

- [SSH 터널]



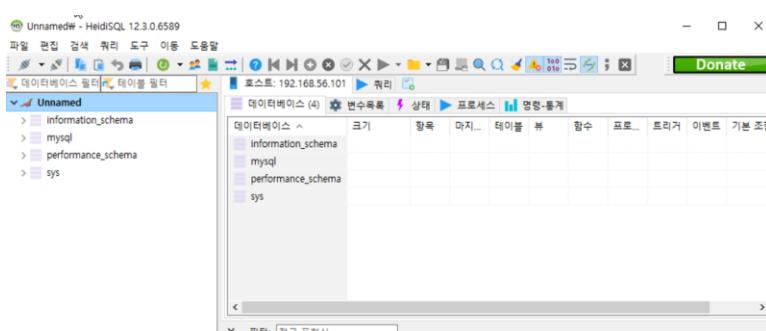
- SSH 호스트: Bastion Host ip 입력
- 외부에서 접속하는데, bastion hos 이기 때문에 내부 ip(192.178.56.102)를 노출시켰고, nat 설정 안해줬기 때문에 192.168.56.102 입력

- [설정]



- 호스트명: U-S1 의 ip 입력  
⇒ 192 점대로 입력해야 함 (U-D 와 연결하는 내부이기 때문에)

- 결과 확인



## (2) SSH 접근 확인하기

- window → U-D1 → U-S1

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ssh kim@192.168.56.102
kim@192.168.56.102's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Applications를 위한 확장된 보안 유지보수 비활성화됨.

9개의 업데이트가 즉시 적용 가능합니다.
해당 추가 업데이트를 확인하려면 apt list --upgradable 을 실행하세요.

ESM Apps(를) 활성화해서 미래의 추가적인 업데이트를 받을 수 있습니다.
https://ubuntu.com/esm 을 참조하거나 sudo pro status 를 실행하십시오.

Failed to connect to https://changelogs.ubuntu.com/meta-release-its. Check your Internet connection or proxy settings

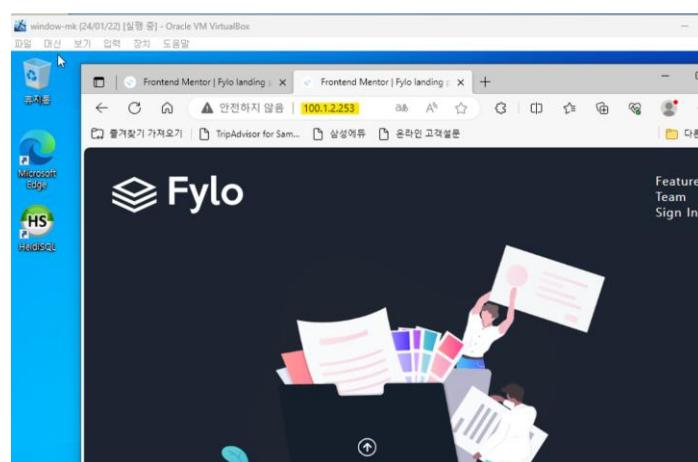
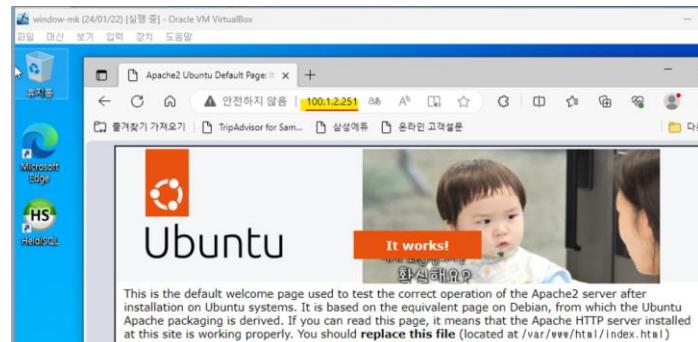
Last login: Sat Jan 27 16:42:26 2024 from 192.168.56.1
kim@kim-VirtualBox:~$ mysql -h 192.168.56.101 -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or #.
Your MariaDB connection id is 32
Server version: 10.6.12-MariaDB-Ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\w' to clear the current input statement.

MariaDB [(none)]>
```

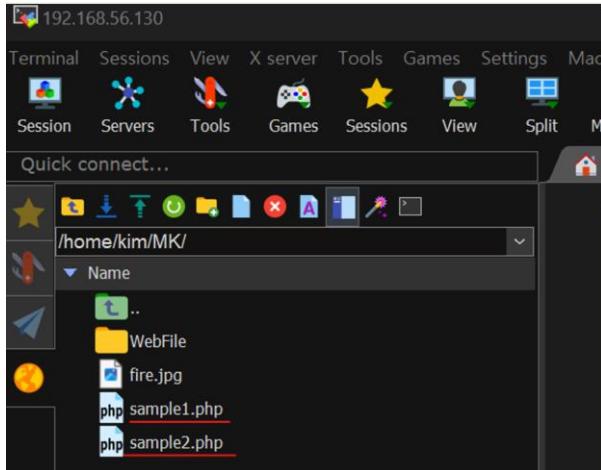
## (3) 웹서버 확인하기



### 3. 2번 실습과정

#### 1) php 설정하기

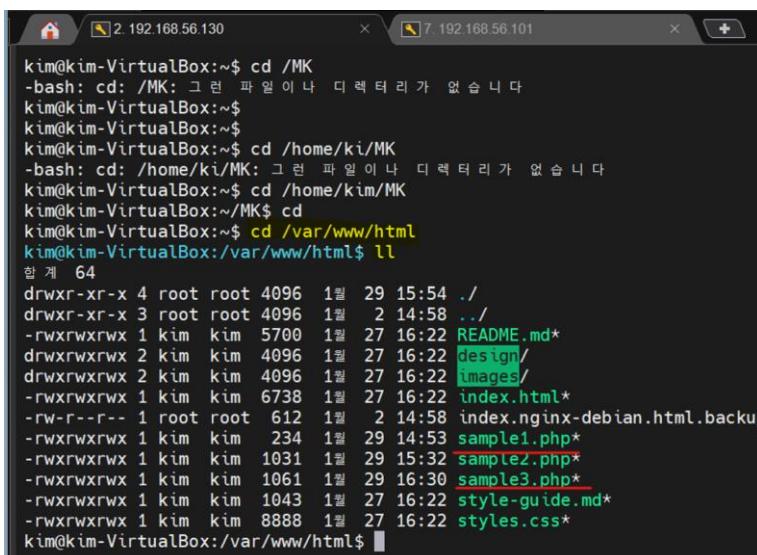
- php 파일을 모바일스팀을 통해 U-D2로 옮기기



- php 파일이 담겨있는 디렉토리로 들어가기 → 파일을 /var/www/html로 옮기기

```
kim@kim-VirtualBox:~$ cd /home/kim/MK
kim@kim-VirtualBox:~/MK$ sudo sample1.php /var/www/html
sudo: sample1.php: 명령이 없습니다
kim@kim-VirtualBox:~/MK$ sudo mv sample1.php /var/www/html
kim@kim-VirtualBox:~/MK$ sudo mv sample2.php /var/www/html
```

- 파일이 옮겨졌는지 확인하기



참고)

중간에 sample2.php 파일을 sample3.php 파일로 변경해서 진행함

- 권한 설정하기

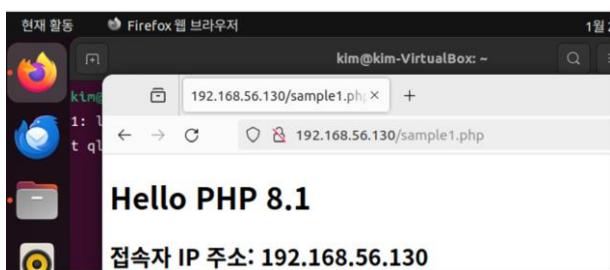
```
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 -R ./sample1.php
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 -R ./sample2.php
3
```

## 2) 결과 확인하기 ⇒ 실패

- php 설치를 하지 않았음

```
sudo apt install php //php 설치하기
```

- U-D2에서 sample1.php 확인하기 ⇒ 성공



## 3) sample3.php 파일 수정하기

- **cd /var/www/html**
- **vim sample3.php**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf8">
</head>
<body>
<?php
    echo "<h1> Hello everyone</h1>";
    $servername = "192.168.56.101"; // 데이터베이스 서버 주소
    $username = "kmk"; // 데이터베이스 사용자 이름
    $password = "0811"; // 사용자 비밀번호
    $dbname = "nation"; // 데이터베이스 이름
    $port = getenv("3306");

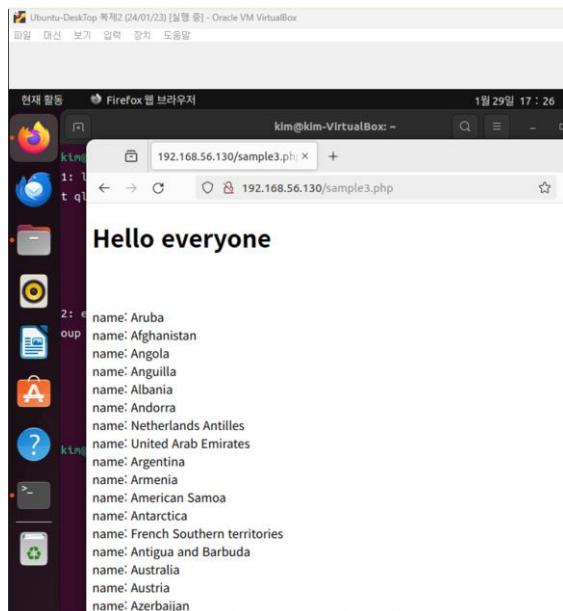
    echo getenv("db_addr") . "<br>";
    echo getenv("db_user") . "<br>";

    // 데이터베이스 연결
    $conn = new mysqli($servername, $username, $password, $dbname, $port);

    // 연결 확인
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // 간단한 SQL 쿼리
    $sql = "select name from countries";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        // 결과 행을 출력
        while($row = $result->fetch_assoc()) {
            echo "name: " . $row["name"] . "<br>";
        }
    }
}
```

#### 4) U-D2에서 sample3.php 확인하기



### III. 과제(게시판)

#### 1. 실습과정

##### 1) DB 구성하기

- mysql 접속하기

```
ubuntu@ubuntu:~$ mysql -u kmk -p -h 192.168.56.101
```

- DB 만들기

```
MariaDB [(none)]> CREATE DATABASE test;
Query OK, 1 row affected (0.003 sec)
```

- Table 만들기

```
MariaDB [(none)]> use test;
```

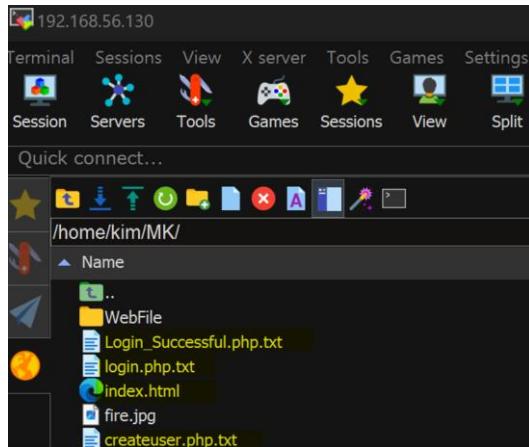
```
MariaDB [test]> CREATE TABLE user (ID VARCHAR(255) PRIMARY KEY, PW VARCHAR(255) NOT NULL);
Query OK, 0 rows affected (0.042 sec)

MariaDB [test]> INSERT INTO user (ID, PW) VALUES ('root', '0811');
Query OK, 1 row affected (0.016 sec)

MariaDB [test]> SELECT * FROM user;
+----+----+
| ID | PW |
+----+----+
| root | 0811 |
+----+----+
```

## 2) 강사님이 주신 파일 설정하기

- U-D2에 강사님이 주신 파일 옮기기



- 파일이 있는 디렉토리에 들어가서 /var/www/html 안으로 위치 옮기기

```
cd /home/kim/MK
```

```
kim@kim-VirtualBox:~/MK$ sudo mv Login_Successful.php.txt /var/www/html
[sudo] kim 암호 :
죄송합니다만, 다시 시도해주시오.
[sudo] kim 암호 :
kim@kim-VirtualBox:~/MK$ sudo mv Login_Successful.php.txt /var/www/html
mv: 'Login_Successful.php.txt' 상태 정보 확인 불가: 그런 파일이나 디렉터리가 없습니다
kim@kim-VirtualBox:~/MK$ sudo mv createuser.php.txt /var/www/html
kim@kim-VirtualBox:~/MK$ sudo mv index2.html /var/www/html
kim@kim-VirtualBox:~/MK$ sudo mv login.php.txt /var/www/html
```

- 파일이 잘 옮겨졌는지 확인하기

```
kim@kim-VirtualBox:~/MK$ cd /var/www/html
kim@kim-VirtualBox:/var/www/html$ ll
합계 80
drwxr-xr-x 4 root root 4096 1월 29 18:51 .
drwxr-xr-x 3 root root 4096 1월 2 14:58 ../
-rw-rw-r-- 1 kim kim 950 1월 29 18:43 Login_Successful.php.txt
-rwxrwxrwx 1 kim kim 5700 1월 27 16:22 README.md*
-rw-rw-r-- 1 kim kim 1354 1월 29 18:43 createuser.php.txt
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 design/
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 images/
-rwxrwxrwx 1 kim kim 6738 1월 27 16:22 index.html*
-rw-r----- 1 root root 612 1월 2 14:58 index.nginx-debian.html.backup
-rw-rw-r-- 1 kim kim 329 1월 29 18:42 index2.html
-rw-rw-r-- 1 kim kim 809 1월 29 18:43 login.php.txt
-rwxrwxrwx 1 kim kim 234 1월 29 14:53 sample1.php*
-rwxrwxrwx 1 kim kim 1031 1월 29 15:32 sample2.php*
-rwxrwxrwx 1 kim kim 1061 1월 29 17:23 sample3.php*
-rwxrwxrwx 1 kim kim 1043 1월 27 16:22 style-guide.md*
-rwxrwxrwx 1 kim kim 8888 1월 27 16:22 styles.css*
kim@kim-VirtualBox:/var/www/html$
```

- 파일에 권한주기

```
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 Login_Successful.php.txt
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 createuser.php.txt
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 index.nginx-debian.html
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 index2.html
```

```
kim@kim-VirtualBox:/var/www/html$ sudo chmod 777 login.php.txt
```

- 권한 준 것 확인

```
kim@kim-VirtualBox:/var/www/html$ ll
합계 80
drwxr-xr-x 4 root root 4096 1월 29 18:51 .
drwxr-xr-x 3 root root 4096 1월 2 14:58 ..
-rwxrwxrwx 1 kim kim 950 1월 29 18:43 Login_Successful.php.txt*
-rwxrwxrwx 1 kim kim 5700 1월 27 16:22 README.md*
-rwxrwxrwx 1 kim kim 1354 1월 29 18:43 createuser.php.txt*
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 design/
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 images/
-rwxrwxrwx 1 kim kim 6738 1월 27 16:22 index.html*
-rwxrwxrwx 1 root root 612 1월 2 14:58 index.nginx-debian.html.backup*
-rwxrwxrwx 1 kim kim 329 1월 29 18:42 index2.html*
-rwxrwxrwx 1 kim kim 809 1월 29 18:43 login.php.txt*
-rwxrwxrwx 1 kim kim 234 1월 29 14:53 sample1.php*
-rwxrwxrwx 1 kim kim 1031 1월 29 15:32 sample2.php*
-rwxrwxrwx 1 kim kim 1061 1월 29 17:23 sample3.php*
-rwxrwxrwx 1 kim kim 1043 1월 27 16:22 style-guide.md*
-rwxrwxrwx 1 kim kim 8888 1월 27 16:22 styles.css*
```

- ~php.txt 로 되어있는 것을 ~php 로 변경해주기

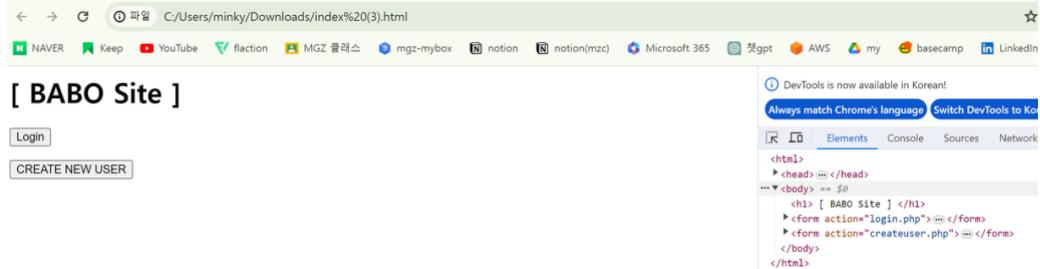
- 다음과 같이 파일 3개 모두 이름 변경

```
kim@kim-VirtualBox:/var/www/html$ sudo mv createuser.php.txt createuser.php
```

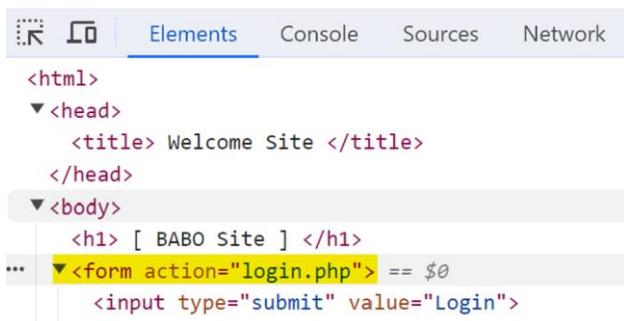
### 3) php 파일 수정하기

- 강사님이 주신 html 파일에 접속하기 > F12

- html 구조를 좀 더 쉽게 볼 수 있음



- login.php 파일 설정하기



- U-D2에서 login.php 파일 수정하기

```
vim login.php
```

```

[ login.php ]
<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $db = new mysqli('192.168.56.101', 'kmk', '0811', 'test');

    if($db->connect_error){
        die('Not Connected : ' . $db->connect_error);
    }

    $query = "select ID, PW from user where ID='$_POST[ID]' && PW='$_POST[PW]'";
    $result = $db->query($query);
    if($result->num_rows != 0){
        header("location: Login_Successful.php");
    } else{
        echo "Information that does not exist";
    }
}
?>

<html>
<h1> LOGIN PAGE </h1>
<form action="" method="post">
    <label for="ID">ID : </label>
    <input type="text" name="ID">

    <label for="PW">PW : </label>
    <input type="text" name="PW">

    <input type="submit" value="submit">
</form>
</html>

```

참고)

You  
html에서 <form action="login.php"> == \$0 이 뭐야

ChatGPT  
'<form action="login.php">'은 HTML에서 사용자의 입력을 서버로 전송하는 HTML 폼(form)을 정의하는 태그입니다. 이 폼은 'login.php'로 데이터를 보내도록 설정되어 있습니다.

'action' 속성은 폼이 제출될 때 데이터가 전송되는 대상 페이지를 지정합니다. 위의 코드에서는 'login.php'로 데이터가 전송됩니다.

- createuser.php 수정하기

```

<html>
    <head>
        <title> Welcome Site </title>
    </head>
    <body>
        <h1> [ BABO Site ] </h1>
        <form action="login.php">
            <input type="submit" value="Login">
        </form>
        ...
        <form action="createuser.php"> == $0
            <input type="submit" value="CREATE NEW USER">
    </body>
</html>

```

**vim createuser.php**

```

[ createuser.php ]
<?php

if ( $_SERVER["REQUEST_METHOD"] == "POST" ){
    $db = new mysqli('192.168.56.101', 'kmk', '0811', 'test');
    if($db->connect_error){
        die('Not Connected : ' . $db->connect_error);
    }

    $sql2 = "select ID from user where ID='$_POST[ID]'";
    $result2 = $db->query($sql2);
    if($result2->num_rows !== 0){
        die("User Exists!");
    }
    $sql = "insert into user (ID,PW) values ('$_POST[ID]', '$_POST[PW]')";
    $result = $db->query($sql);
    if($result){
        echo '<p>' . "List of users" . '</p>';
        echo "-----";
        $conn=mysqli_connect('192.168.56.101', 'kmk', '0811', 'test');
        $sql="select * from user";
        $result=mysqli_query($conn, $sql);

        while( $row = mysqli_fetch_array( $result ) ) {
            echo '<p>' . "ID : " . $row[ 'ID' ] . "&nbsp;&nbsp;" . "PW : " . $row[ 'PW' ] . '</p>';
        }
    }
    else{
        echo "ERROR ";
    }
}
?>

<html>
<h1> CREATE A NEW USER! </h1>
<form action="" method="post"> ← → − 100% + ⊞ ↻
    <label for="ID">NEW ID : </label>

```

- Login\_Successful.php 파일 수정하기

```

[ Login_Successful.php ]
<html>
<div style="text-align:center"> <b> <font size="40"> Login Successful </font> </b> </div>
</html>

<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){

    $db = new mysqli('192.168.56.101', 'kmk', '0811', 'test');

    if($db->connect_error){
        die('Not Connected : ' . $db->connect_error);
    }

    $query = "select ID, PW from user where ID='$_POST[ID]' && PW='$_POST[PW]';

    $result = $db->query($query);
    if($result->num_rows !== 0){
        header("location: Login_Successful.php");
    }
    else{
        echo "Information that does not exist";
    }
}
?>

<html>
<h1> LOGIN PAGE </h1>
<form action="" method="post">
    <label for="ID">ID : </label>
    <input type="text" name="ID">

    <label for="PW">PW : </label>
    <input type="text" name="PW">

    <input type="submit" value="submit">
</form>
</html>

```

#### 4) 결과 확인하기

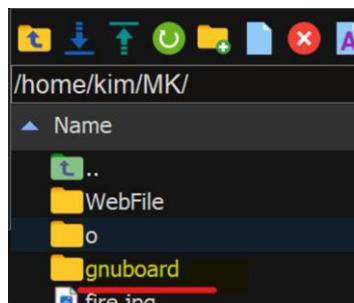
The screenshots illustrate the following steps:

- Login Page:** The first screenshot shows the login page at `192.168.56.130/login.php?`. It displays a table with one row and fields for ID (babo) and PW (1234). The "submit" button is visible.
- Login Successful:** The second screenshot shows the success page at `192.168.56.130/Login_Successful.php`. It displays the message "Login Successful" in a yellow box.
- Create User Page:** The third screenshot shows the user creation page at `192.168.56.130/createuser.php?`. It displays the message "CREATE A NEW USER!" and fields for NEW ID (sia) and NEW PW (0001).
- User Creation Confirmation:** The fourth screenshot shows the user creation page again, but now with three rows of users listed in the table. The new user "sia" has been added with ID "sia" and PW "0001". The MySQL command prompt on the right shows the query `select * from user` and its results, confirming the new user was created successfully.

#### 4) 파일 설정하기

- 그누보드에서 zip 파일을 RPC에 다운받기 > 압축풀기

[https://sir.kr/g5\\_pds/3220](https://sir.kr/g5_pds/3220)

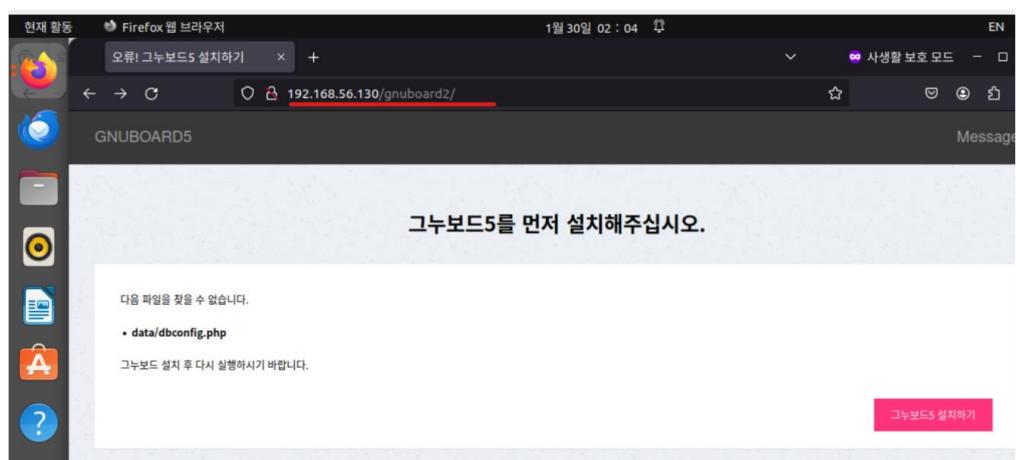


- 파일 옮기기

```
kim@kim-VirtualBox:/var/www/html$ cd /home/kim/MK
kim@kim-VirtualBox:~/MK$ sudo mv gnuboard2 /var/www/html
kim@kim-VirtualBox:~/MK$ 
kim@kim-VirtualBox:~/MK$ cd /var/www/html
kim@kim-VirtualBox:/var/www/html$ ll
합계 92
drwxr-xr-x 7 root root 4096 1월 30 02:03 .
drwxr-xr-x 3 root root 4096 1월 2 14:58 ../
-rwxrwxrwx 1 kim kim 953 1월 29 19:53 Login_Successful.php*
-rwxrwxrwx 1 kim kim 5700 1월 27 16:22 README.md*
-rwxrwxrwx 1 kim kim 1360 1월 29 20:25 createuser.php*
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 design/
drwxrwxrwx 13 kim kim 4096 1월 30 00:39 gnuboard/
drwxrwxr-x 15 kim kim 4096 1월 29 20:41 gnuboard-gnuboard5-c67118f/
drwxrwxr-x 15 kim kim 4096 1월 30 02:01 gnuboard2/
drwxrwxrwx 2 kim kim 4096 1월 27 16:22 images/
```

- U-D2에서 해당 파일로 접속하고 설치하기

- 접속하기 > '그누보드 5 설치하기' 클릭



- 해당 파일 내에 config.php 가 있는 곳에서 data 폴더 생성 & data 에 권한 부여하기

```
kim@kim-VirtualBox:/var/www/html$ cd gnuboard2/
kim@kim-VirtualBox:/var/www/html/gnuboard2$ ll
합계 292
drwxrwxr-x 15 kim  kim  4096  1월 30 02:01 .
drwxr-xr-x  7 root root  4096  1월 30 02:03 ..
-rw-rw-r--  1 kim  kim  63147  1월 30 02:00 LICENSE.txt
-rw-rw-r--  1 kim  kim     0  1월 30 02:00 README.md
-rw-rw-r--  1 kim  kim   277  1월 30 02:00 _common.php
-rw-rw-r--  1 kim  kim   109  1월 30 02:00 _head.php
-rw-rw-r--  1 kim  kim   109  1월 30 02:00 _tail.php
drwxrwxr-x  6 kim  kim  4096  1월 30 02:01 adm/
drwxrwxr-x  2 kim  kim  4096  1월 30 02:01 bbs/
-rw-rw-r--  1 kim  kim  35832  1월 30 02:01 common.php
-rw-rw-r--  1 kim  kim   9549  1월 30 02:01 config.php
drwxrwxr-x  3 kim  kim  4096  1월 30 02:01 css/
drwxrwxr-x  2 kim  kim  4096  1월 30 02:01 extend/
-rw-rw-r--  1 kim  kim   2538  1월 30 02:01 g4_import.php
-rw-rw-r--  1 kim  kim  11542  1월 30 02:01 g4_import_run.php
-rw-rw-r--  1 kim  kim   9643  1월 30 02:01 head.php
-rw-rw-r--  1 kim  kim   4974  1월 30 02:01 head.sub.php
drwxrwxr-x  3 kim  kim  4096  1월 30 02:01 img/
-rw-rw-r--  1 kim  kim   2568  1월 30 02:01 index.php
drwxrwxr-x  3 kim  kim  4096  1월 30 02:01 install/
drwxrwxr-x  7 kim  kim  4096  1월 30 02:01 js/
drwxrwxr-x  7 kim  kim  4096  1월 30 02:01 lib/
drwxrwxr-x  4 kim  kim  4096  1월 30 02:00 mobile/
-rw-rw-r--  1 kim  kim   7344  1월 30 02:00 orderupgrade.php
-rw-rw-r--  1 kim  kim    398  1월 30 02:00 perms.sh
drwxrwxr-x 20 kim  kim  4096  1월 30 02:00 plugin/
drwxrwxr-x 10 kim  kim  4096  1월 30 02:00 shop/
-rw-rw-r--  1 kim  kim   6746  1월 30 02:00 shop.config.php
drwxrwxr-x 17 kim  kim  4096  1월 30 02:00 skin/
-rw-rw-r--  1 kim  kim   3000  1월 30 02:00 tail.php
-rw-rw-r--  1 kim  kim   554  1월 30 02:00 tail.sub.php
drwxrwxr-x  3 kim  kim  4096  1월 30 02:00 theme/
-rw-rw-r--  1 kim  kim    562  1월 30 02:00 version.php
-rw-rw-r--  1 kim  kim   7239  1월 30 02:00 yc4_import.php
-rw-rw-r--  1 kim  kim  25221  1월 30 02:00 yc4_import_run.php
kim@kim-VirtualBox:/var/www/html/gnuboard2$ mkdir data
kim@kim-VirtualBox:/var/www/html/gnuboard2$ sudo chmod 777 data
kim@kim-VirtualBox:/var/www/html/gnuboard2$
```

- '동의합니다' 체크 > 다음

**라이센스(License) 내용을 반드시 확인하십시오.**  
라이센스에 동의하시는 경우에만 설치가 진행됩니다.

소프트웨어 제품은 저작권법 및 국제저작권 협약을 비롯하여, 기타 지적재산권법 및 협약의 보호를 받습니다.

프로그램 명칭 : 그누보드5 ( GNU Board 5 )

저작자 : (주)에스아이알소프트 <http://sir.kr>

라이센스 (License)

번역문 아래에 원문이 있습니다.

주의 )

- 번역문과 원문의 내용상 차이가 있는 경우 원문의 내용을 우선으로 따릅니다.
- 법적인 분쟁이 발생한 경우 저작자의 회사 소재지를 관할하는 관할법원에서 분쟁을 해결합니다.
- 미 라이센스 파일 및 내용은 저작자를 제외한 어느 누구도 추가, 수정, 삭제할 수 없습니다.

----- LGPL 번역문 -----

동의합니다.

다음

## 5) DB 만들고 해당 설정하기

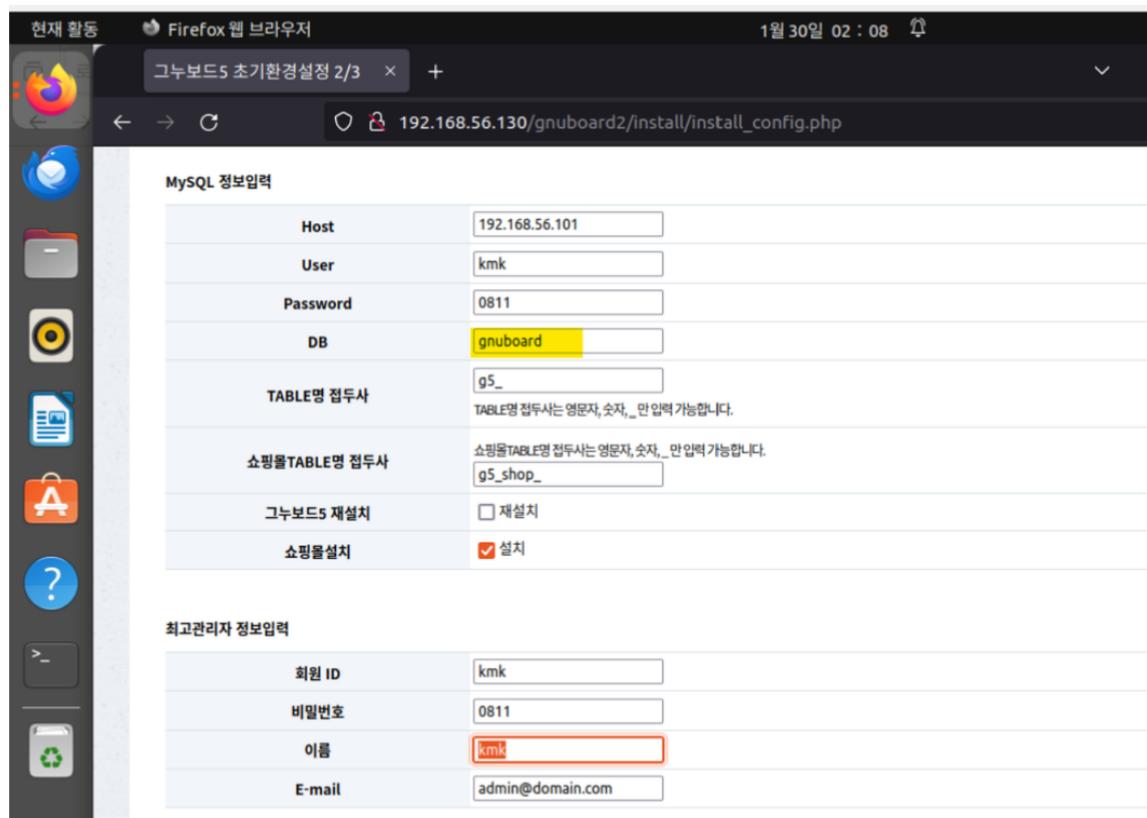
- U-S1에서 DB 만들기

```
MariaDB [(none)]> create database gnuboard;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| gnuboard |
| information_schema |
| mysql |
| nation |
| performance_schema |
| sys |
| test |
+-----+
7 rows in set (0.002 sec)

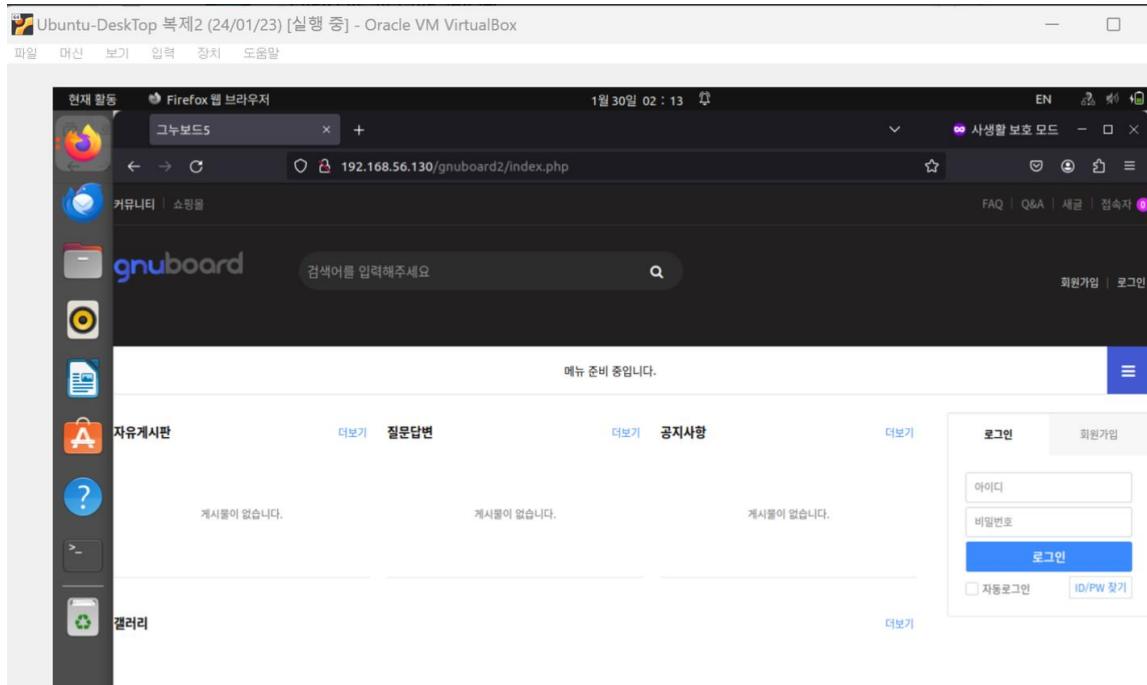
MariaDB [(none)]>
```

- 다시 U-D2에서 다음과 같이 설정해주기



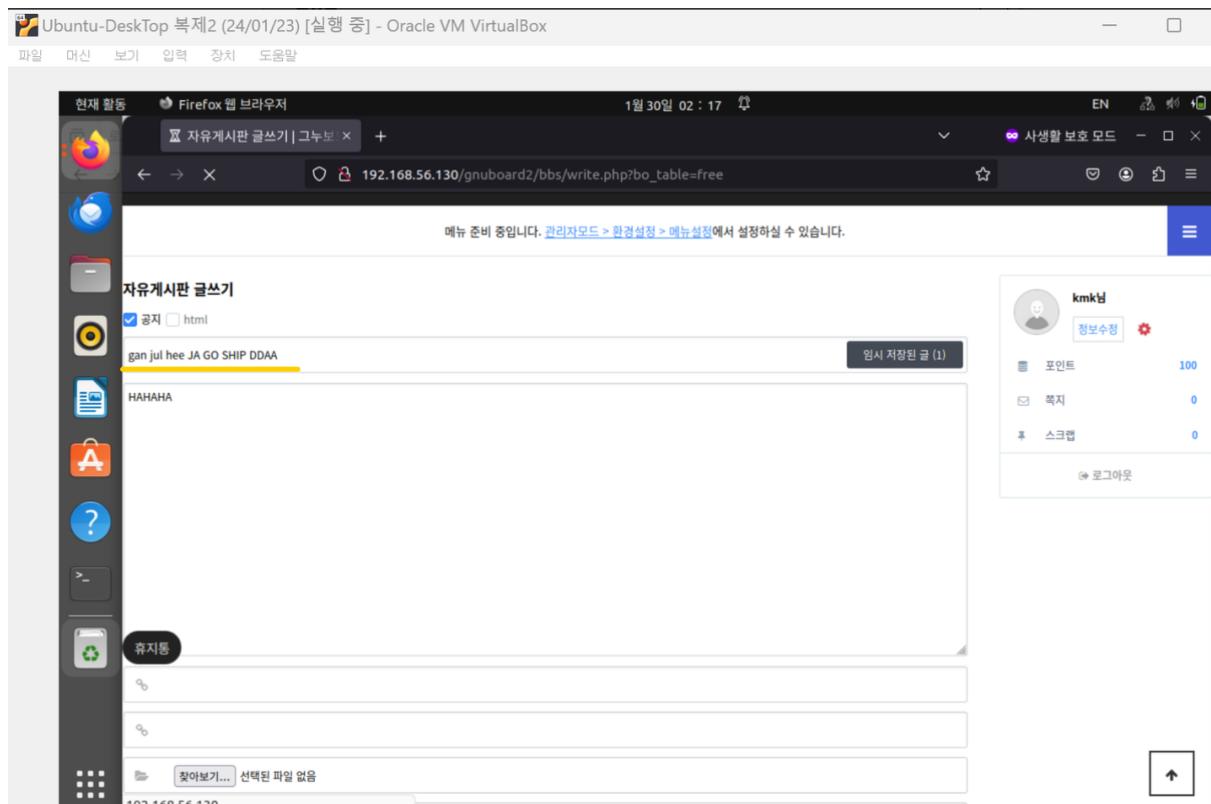
- Host : db 서버의 IP
- User : DB user
- Password : DB password

## 6) 결과물 확인하기



- 관리자 모드로 로그인

- 자유게시판 글쓰기



서나경 실습가이드

<https://jaws-coding.tistory.com/m/9>

<https://dbknowledge.tistory.com/m/78>