**### MariaDB (Ubuntu 20.04) ###**

01. 설치 및 실행 확인

**# apt update**

**# apt install -y mariadb-server mariadb-client**

**# systemctl status mariadb**

>>> **q**

**# mariadb**

MariaDB [(none)]> **exit**

--> 설치 및 실행 확인

02. mariadb 계정 설정

**# mysql\_secure\_installation**

Enter current password for root (enter for none):

--> **No MariaDB root: enter 입력**

Setting the root password ensures that nobody can log into the MariaDB

root user without the proper authorisation.

Set root password? [Y/n] **Y**

New password: **mariadb**

Re-enter new password: **mariadb**

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone

to log into MariaDB without having to have a user account created for

them. This is intended only for testing, and to make the installation

go a bit smoother. You should remove them before moving into a

production environment.

Remove anonymous users? [Y/n] **Y**

... Success!

Normally, root should only be allowed to connect from 'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] **Y**

... Success!

By default, MariaDB comes with a database named 'test' that anyone can

access. This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n] **Y**

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n] **Y**

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB

installation should now be secure.

Thanks for using MariaDB!

03. root 사용자와 패스워드로 접근 가능한지 확인

**# mariadb -u root -p**

Enter password: **mariadb**

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 45

Server version: 10.3.38-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

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

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

MariaDB [(none)]> **exit**

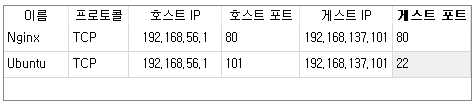
**### Nginx (Ubuntu 20.04) ###**

01. nginx 설치

**# apt install -y nginx**

02. 웹 브라우저에서 연결 상태 확인

>>> **http://localhost:80**



**### PHP (Ubuntu 20.04) ###**

01. 설치 가능한 php-fpm 버전을 확인

**# apt-cache madison php-fpm**

php-fpm | **2:7.4+75** | http://kr.archive.ubuntu.com/ubuntu focal/universe amd64 Packages

php-fpm | **2:7.4+75** | http://kr.archive.ubuntu.com/ubuntu focal/universe i386 Packages

02. php 설치

**# apt install -y php-fpm=2:7.4+75**

**# php --version**

PHP 7.4.3-4ubuntu2.18 (cli) (built: Feb 23 2023 12:43:23) ( NTS )

Copyright (c) The PHP Group

Zend Engine v3.4.0, Copyright (c) Zend Technologies

with Zend OPcache v7.4.3-4ubuntu2.18, Copyright (c), by Zend Technologies

03. php Configuration File

**# vi /etc/nginx/sites-available/default**

====================================================================

**>>> index.php index.php?$args** **두 개의 항목 추가** (**44G**)

# Add index.php to the list if you are using PHP

index **index.php index.php?$args** index.html index.htm index.nginx-debian.html;

**>>> 주석 제거** (**56G**)

**location ~ \.php$ {**

**>>> 주석 제거**

**include snippets/fastcgi-php.conf;**

#

# # With php-fpm (or other unix sockets):

**>>> 주석 제거 & 설치된 버전으로 수정**

**fastcgi\_pass unix:/run/php/php7.4-fpm.sock;**

# # With php-cgi (or other tcp sockets):

# fastcgi\_pass 127.0.0.1:9000;

**>>> 주석 제거**

**}**

====================================================================

**# systemctl daemon-reload**

**# systemctl restart nginx**

04. 웹 페이지 연결 정보 확인

**# cd /var/www/html**

**# vi index.php**

**<?php**

**phpinfo();**

**?>**

05. 웹 브라우저에서 연결 상태 확인

>>> **http://localhost:80**

**### WordPress (Ubuntu 20.04) ###**

01. WordPress 프로그램 다운

**# cd /var/www**

**# wget https://ko.wordpress.org/latest-ko\_KR.tar.gz**

**# tar -xvf latest-ko\_KR.tar.gz**

**# rm -rf html**

**# mv wordpress html**

**# chown -Rf www-data:www-data html**

02. 워드프레스에서 mariadb에 접속가능하도록 php7.4-mysql을 설치

**# apt-get install php7.4-mysql**

>>> **http://localhost:80**

03. 데이터베이스 생성

**# mariadb -u root -p**

Enter password: **mariadb**

MariaDB [(none)]> **show databases;**

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

+--------------------+

3 rows in set (0.015 sec)

MariaDB [(none)]> **use mysql**

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [mysql]> **create database wordpress\_db;**

Query OK, 1 row affected (0.000 sec)

MariaDB [mysql]> **GRANT ALL ON wordpress\_db.\* TO 'wpuser'@'localhost' IDENTIFIED BY 'mariadb' WITH GRANT OPTION;**

Query OK, 0 rows affected (0.014 sec)

MariaDB [mysql]> **FLUSH PRIVILEGES;**

Query OK, 0 rows affected (0.001 sec)

MariaDB [mysql]> **exit**

Bye

04. wordpress를 사용한 블로그 생성하기

>>> **http://localhost:80** 접속 >>> 시작합니다! (클릭)

데이터베이스 이름 : **wordpress\_db**

사용자명 : **wpuser**

비밀번호 : **mariadb**

데이터베이스 호스트: **localhost**

테이블 접두어 : **wp\_**

>>> 제출 (클릭)

>>> 설치 실행 (클릭)

05. 블로그 생성

사이트 제목: **Test Blog**

사용자명: **tester**

비밀번호: **test**

>>> 비밀번호 확인: **약한 비밀번호 사용 확인** (선택)

이메일 주소: **test@test.com**

>>> **워드프레스 설치** (클릭)

06. Login

>>> **http://localhost/wp-login.php**

사용자명 또는 이메일 주소

>>> **tester**

비밀번호

>>> **test**

>>> **로그인** (클릭)

* **수고하셨습니다 -**