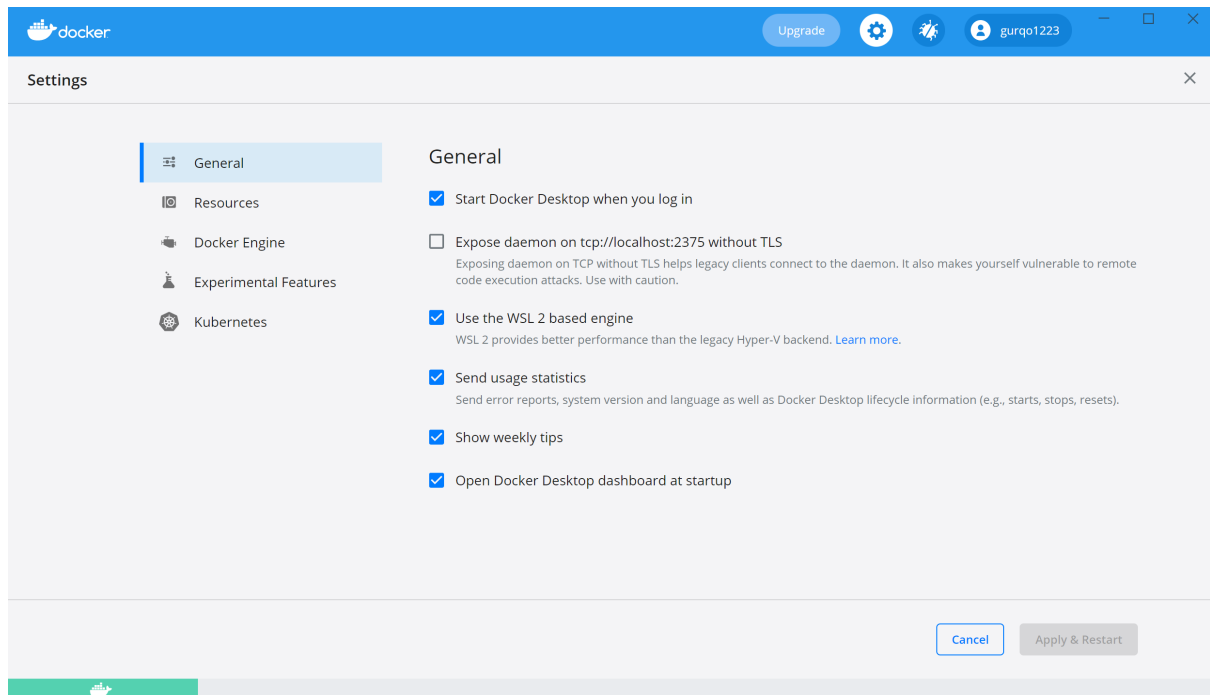


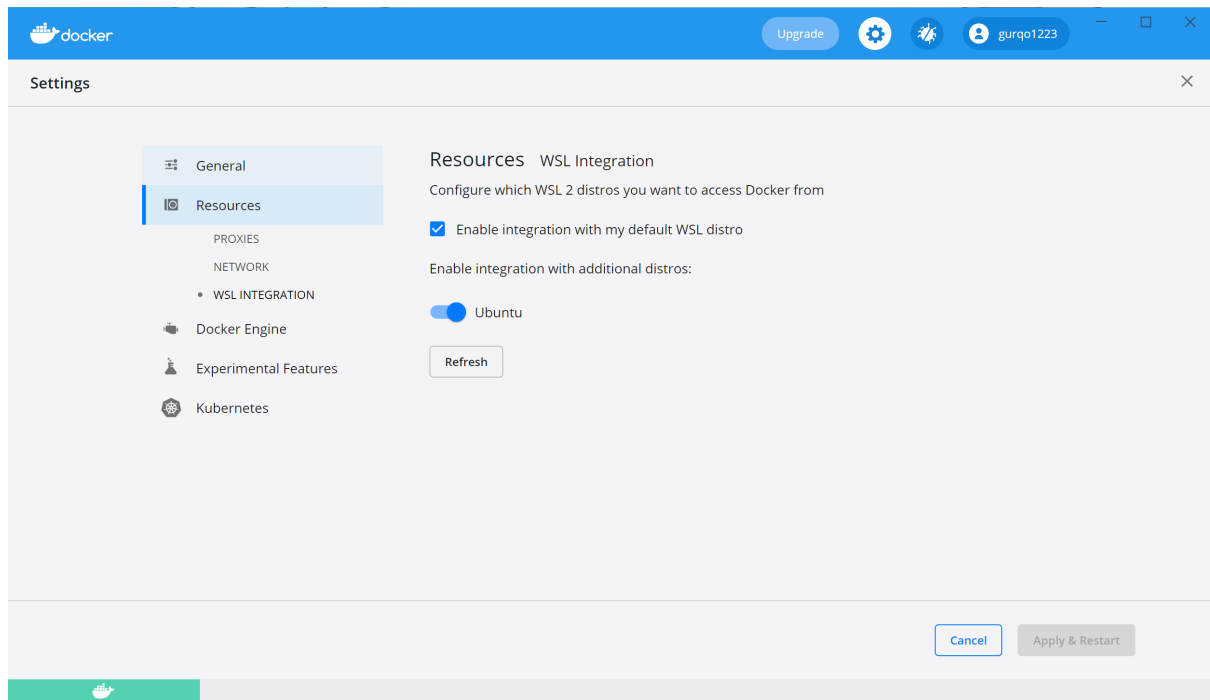
02_ft_server_install

Docker Download

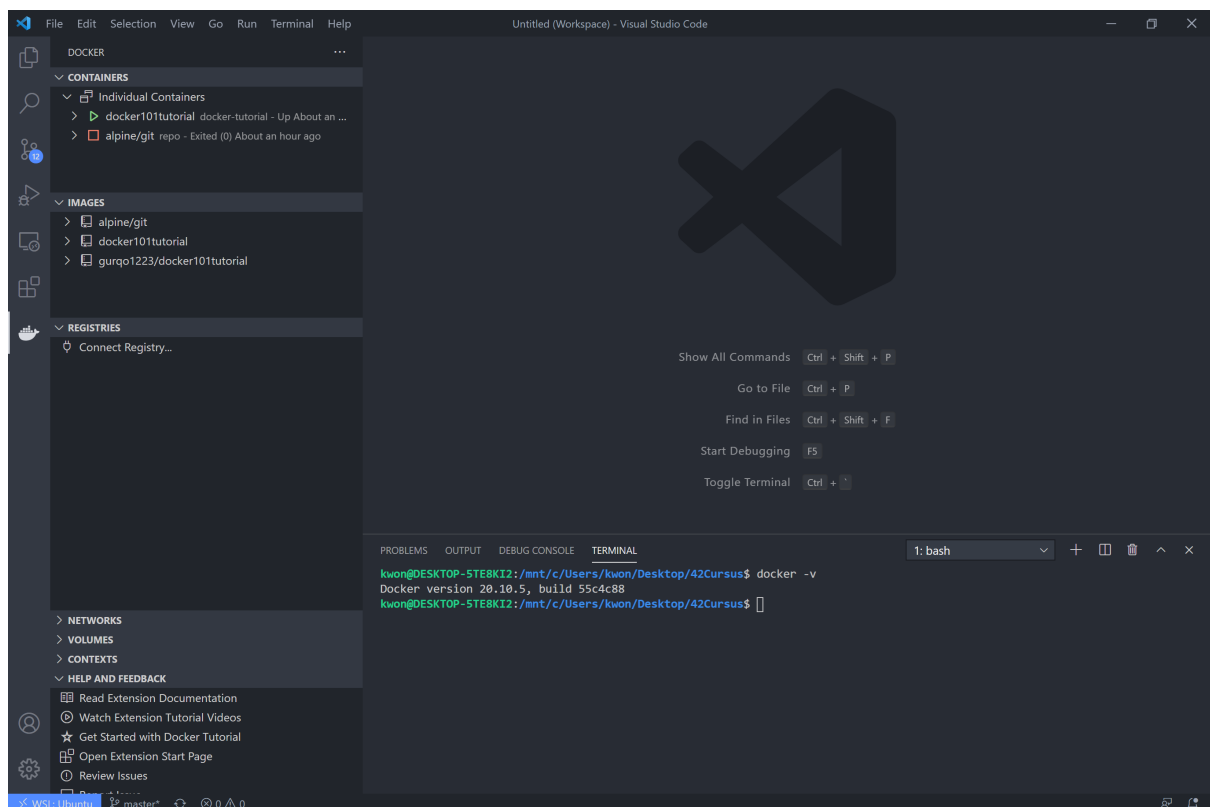
- [Docker Download Window](#) 에서 Window용 Docker를 다운받는다.
- 설치가 완료되면 튜토리얼을 진행한 후 WSL에서 사용하기 위해 설정을 해주어야 한다.
 - WSL(Window Subsystem for Linux) : Windows10에서 Linux Syetme을 사용할 수 있도록 지원하는 기능.
 - [WSL2 설치 가이드](#)
- Settings - General
 - Use the WSL 2 based engine 항목이 체크가 되어있는지 확인하자.



- Settings - Resources - WSL Integration
 - 자신이 사용하고 있는 wsl2 배포판이 맞는지 확인하고 선택을 해주자.



- Docker 설치가 완료되면 좌측 바에 docker icon이 나타난다.
- wsl에서 사용을 하기위해 설정을 했으므로 현재 container와 image 목록을 확인할 수 있다.



Docker Container에 올릴 image 생성

- Debian:buster는 리눅스와 같이 OS 중의 하나.
- image란 vmware와 같은 가상환경에서 unbuntu를 설치하기 위해 사용했던 unbuntu.iso와 같은 것.

```
//image 생성
docker pull debian:buster
//image 확인
docker images
```

```
> docker pull debian:buster
buster: Pulling from library/debian
e22122b926a1: Pull complete
Digest: sha256:9d4ab94af82b2567c272c7f47fa1204cd9b40914704213f1c257c44042f82aac
Status: Downloaded newer image for debian:buster
docker.io/library/debian:buster
```

```
> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
debian	buster	dc2eddc15825	8 days ago	114MB

Docker로 debian:buster 환경 실행 및 접속

- -i : interactive(입출력)
- -t : tty(터미널) 활성화
- --name [container name] : container 이름을 설정한다. 설정하지 않으면 랜덤으로 지정.
- -p host port:container port : container의 port를 개방한 뒤 host port와 연결.
 - 80번과 443번 port를 사용하겠다는 의미.
 - 왜 80번과 443번 port를 사용하는가?
- debian:buster : 명시하지 않아도 자동으로 최신버전을 불러온다.

```
docker run -it --name con_debian -p 80:80 -p 443:443 debian:buster
```

- 성공하면 다음과 같이 debian bash에 접속이 된 것을 확인할 수 있다.
- exit를 입력하여 종료하여도 container가 중지되는 것은 아니다. container는 실행중 인 상태에서 접속을 끊은 것.
- attach를 입력하여 다시 접속할 수 있다.

```
> docker run -it --name con_debian -p 80:80 -p 443:443 debian:buster
root@e90c9f4dfb5e:/#
```

Debian:buster에 Nginx, cURL 설치

- Debian에서는 패키지 관리자로 apt-get을 사용한다.
- 패키지 목록을 최신으로 설치한 후 업그레이드까지 진행하자.
- -y : 설치에 동의하는지 묻는 질문에 동의한다는 의미.
- curl(Client URL) : 서버와 통신할 수 있는 커맨드 명령어 툴.
 - 클라이언트에서 커맨드 라인이나 소스코드로 손쉽게 웹 브라우저처럼 활동할 수 있도록 해주는 기술.(참고)

```
apt-get update
apt-get upgrade
apt-get -y install nginx curl vim
```

Nginx 서버 연결 확인

- nginx를 시작하고 작동하는지 확인을 한다.
- [ok] nginx is running 문구가 나오면 잘 작동한다는 의미.

```
service nginx start  
service nginx status
```

```
root@e90c9f4dfb5e:/# service nginx start  
[ ok ] Starting nginx: nginx.  
root@e90c9f4dfb5e:/# service nginx status  
[ ok ] nginx is running.
```

- localhost:80 웹 사이트에서도 확인을 할 수 있다.
- 다음과 같이 Welcome to nginx 문구가 나오면 서버와 성공적으로 소통했다는 것을 알 수 있다.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- 터미널을 통해서 확인할 수도 있다.

```
curl localhost
```

```

root@e90c9f4dfb5e:/# curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed
and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```

- 설치가 제대로 되었고 연결이 잘 되는 것을 확인했으니 종료를 해준다.

```
service nginx stop
```

```

root@e90c9f4dfb5e:/# service nginx stop
[ ok ] Stopping nginx: nginx.

```

openssl로 self-signed SSL 인증서 생성

- 사이트를 안전하고 서버와 암호화된 통신을 할 수 있게 해주는 인증서를 만들자.
- 인증서를 만드는 방법은 2가지가 있다.
 - CSR을 만들어 CA에 요청하여 발급 받는 방법.
 - Self-signed 인증서를 만드는 방법.
 - CSR 명시적 생성 → 인증서에 self-sign → 인증서 완성.
 - CSR을 명시적으로 생성하지 않고 key와 부가 정보를 입력하여 직접 self-sign하여 인증서 완성.
- self-signed SSL 인증서란 자체적으로 발급받은 인증서로 로그인 및 기타 개인 계정 인증 정보를 암호화한다.

- 무료 오픈소스 openssl을 이용하여 인증서를 만들 것이다.

- openssl 설치→개인키 생성→인증서생성→권한제한

- openssl 설치

```
apt-get -y install openssl
```

- 개인키 및 인증서 생성

```
openssl req -newkey rsa:4096 -days 365 -nodes -x509 -subj "/C=KR/ST=Seoul/L=Seoul/O=42Seoul/OU=hkwon/CN=localhost" -keyout localhost.d
```

- localhost.dev.key와 localhost.dev.crt가 생성된다.

- 옵션

- req : 인증서 요청 및 인증서 생성 유틸.
- -newkey : 개인키를 생성하기 위한 옵션.
- -keyout <키 파일 이름> : 키 파일 이름을 지정해 키 파일 생성.
- -out <인증서 이름> : 인증서 이름을 지정해 인증서 생성.
- days 365 : 인증서의 유효기간.
- CN(Command Name) : 일반 이름(인증서 고유 이름), 대부분의 인증기관 CA에서는 SSL인증서 신청 시 도메인명을 CN을 지정.
- O(Organization) : 기관명.
- OU(Organization Unit) : 회사/기관 내의 '사업부, 부분, 부서, 본부, 과, 팀' 정도.
- L(City/Locality) : 시/도.
- S(State/County/Region) : 구/군.
- ST(Street) : 나머지 상세 주소. (OV, EV 인증시 필요)
- C(Country) : 국가를 나타내는 ISO 코드를 지정. 한국은 KR, 미국은 US 등 2자리 코드.

- 권한제한

- 생성된 키 파일과 인증서 파일의 경로를 이동시킨다.
- 각 인증서 파일에 대한 권한을 644로 변경하여 root 소유주만 수정이 가능하게 한다.

```
mv localhost.dev.crt etc/ssl/certs/  
mv localhost.dev.key etc/ssl/private/  
chmod 644 etc/ssl/certs/localhost.dev.crt etc/ssl/private/localhost.dev.key
```

Nginx에 SSL 설정 및 url redirection 추가

- default 파일에 https 연결을 위한 작업.
- 기존 파일은 서버 블록이 하나이며 80 port만 수신대기 상태.
- https 연결(ssh 설정)을 위해 443 port를 수신대기하고 있는 서버 블록을 추가 작성한다.

```
# Default server configuration  
#  
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    return 301 https://$host$request_uri;  
}  
  
# SSL configuration  
#
```

```
server {
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    # ssl setting
    ssl on;
    ssl_certificate /etc/ssl/certs/localhost.dev.crt;
    ssl_certificate_key /etc/ssl/private/localhost.dev.key;

    # set root dir
    root /var/www/html;

    # Add index.php to the list if you are using PHP
    # autoindex
    index index.html index.htm index.nginx-debian.html;

    server_name ft_server;

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

- 기존 80 port로 들어온 요청을 https(443 port) 요청으로 redirect 하도록 하였다.
- 443 port로 들어온 요청은 설정한 ssl_certificate, ssl_certificate_key 파일을 이용하여 ssl 통신을 하게 된다.
- 아래의 명령어 중 하나를 입력하여 수정사항을 적용시킨다.

```
service nginx reload
service nginx restart
```

- localhost를 열어 다음과 같이 경고문구가 뜨면 성공.



연결이 비공개로 설정되어 있지 않습니다.

공격자가 **localhost**에서 정보(예: 비밀번호, 메시지, 신용카드 등)를 도용하려고 시도 중일 수 있습니다. [자세히 알아보기](#)

NET::ERR_CERT_AUTHORITY_INVALID



Chrome에서 가장 강력한 보안 기능을 사용하려면 [향상된 보호 모드를 사용 설정](#) 하세요.

고급

안전한 페이지로 돌아가기

php-fpm 설치 및 nginx 설정

- php
 - php란 동적 페이지를 위해 필요한 기술. 대표적인 서버 사이드 스크립트 언어.
 - wordpress 웹 페이지 또한 php기반으로 php가 설치되어야 정상적으로 동작이 가능하다.
- php-fpm(FastCGI Process Manger)
 - 일반 CGI보다 빠른 처리가 가능한 FastCGI.

- php-fpm을 통해 nginx와 php를 연동시켜 웹 서버가 정적 content 뿐만 아니라 동적 content를 다룰 수 있게 만드는 것.
- CGI: Common Gateway Interface
 - 두 컴퓨터(서버-클라이언트) 사이의 HTML 등의 언어를 양방향으로 번역해주는 것.
 - nginx는 웹 서버이기 때문에 정적 content밖에 다루지 못한다.
 - 동적 페이지를 구현하기 위해서는 웹 서버 대신 동적 content를 읽은 후 html로 변환시켜 웹 서버에 다시 전달 해주는 외부 프로그램이 필요하다. → php module
 - 이러한 연결 과정의 방법 또는 규약을 정의한 것.
- php-fpm을 설치하고 nginx default 파일에 php 처리를 위한 설정을 추가한다.

```
apt-get install php-fpm
```

- 기본 default 파일에 php 설정이 주석처리가 되어 있으므로 주석해제를 해주자.
- autoindex에 index.php를 추가해주자.(php를 쓰려면 추가해주어야 한다)

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    return 301 https://$host$request_uri;
}

# SSL configuration
#
server {

    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    # ssl setting
    ssl on;
    ssl_certificate /etc/ssl/certs/localhost.dev.crt;
    ssl_certificate_key /etc/ssl/private/localhost.dev.key;

    # set root dir
    root /var/www/html;

    # Add index.php to the list if you are using PHP
    # autoindex
    index index.html index.htm index.nginx-debian.html index.php;

    server_name ft_server;

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

    # php
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.3-fpm.sock;
    }
}
```

- php-fpm 작동 확인

```
service php7.3-fpm start
service php7.3-fpm status
```

```
root@2c3377131b9f:/# service php7.3-fpm start
root@2c3377131b9f:/# service php7.3-fpm status
[ ok ] php-fpm7.3 is running.
```


Nginx autoindex 설정

- autoindex 기능을 켜줘야 파일이 아닌 디렉토리를 가리키는 url에 대한 요청을 받을 때 파일의 content를 반환해준다.
 - 요청한 url에 대응되는 디렉토리 안에서 index.html 혹은 index.htm 이름의 파일을 반환.
- autoindex 기능이 켜져있을때 해당 디렉토리에 index 목록에 해당하는 파일이 없으면, 웹 서버는 자동으로 그 디렉토리 파일들을 크기, 변경일, 해당 파일에 대한 링크에 대한 정보담긴 HTML 파일을 반환한다.
- location / 부분에 autoindex 추가.

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

- default 파일에서 index.nginx-debian.html을 주석처리하면 다음과 같이 읽을 파일이 없다고 생각하여 전체 파일 목록을 반환한다.

Index of /

../	22-Mar-2021 16:53	612
index.nginx-debian.html		

MariaDB(mysql) 설치

- 과제에서는 mySQL을 이용하라고 하지만, debian buster에서는 mariaDB만을 지원한다.
 - mariaDB도 mySQL과 동일한 소스코드를 기반으로 하여 대부분의 작업이 호환된다.
- php-mysql : php에서 mysql에 접근할 수 있게 해주는 module.
 - php에서 mysql 명령어를 실행시킬 수 있게 해준다.
- php-mbstring : 2byte 확장 문자를 읽을 수 있게 해주는 module.
 - 한국어, 중국어, 일본어와 같은 multibyte를 처리할 수 있게 해준다.
- 에러 발생 시 mariaDB를 삭제하고 다시 설치할 경우 [참고](#)

```
apt-get -y install mariadb-server php-mysql php-mbstring
```

Wordpress 설치 및 설정

- 설치
 - 이전에 설치한 curl을 이용하여 <https://wordpress.org/latest.tar.gz> 를 다운받아 설치한다.
 - [curl](#), [tar](#) 명령어 참고
 - nginx.conf에 user가 www-data로 작성되어있다. wordpress의 유저그룹을 그에 맞게 권한 설정을 해준다.
 - chown : 리눅스에서 소유자를 변경하는 command.
 - -R -recursive : 에러메세지가 있어도 출력하지 않게 한다.

```
#curl -O https://wordpress.org/latest.tar.gz  
#tar -xzf latest.tar.gz -C var/www/html/  
wget https://wordpress.org/latest.tar.gz; \  
tar -xvf latest.tar.gz; \  
mv wordpress/ var/www/html/; \  
chown -R www-data:www-data /var/www/html/wordpress
```

- wp-config.php 파일 수정
 - 압축을 푼 폴더 안에 보면 `wp-config-sample.php` 파일이 존재한다. 이는 sample 파일로 사용할 파일은 `wp-config.php` 이다.
 - 실제 사용할 `wp-config.php` 파일 안에 DB_NAME, DB_USER, DB_PASSWORD 3가지 항목을 수정해준다.
 - ARG 명령어를 통해 관리할 수도 있다.

```
cp -rp var/www/html/wordpress/wp-config-sample.php var/www/html/wordpress/wp-config.php
```

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'hkwon' );

/** MySQL database password */
define( 'DB_PASSWORD', 'hkwon' );
```

- wordpress database 생성
 - mysql을 실행시킨 후 접속하여 database를 생성한다.
 - ';' 를 반드시 입력하자.
 - [mysql 문법](#) 참고
 - user를 생성한다.
 - '@localhost' 는 localhost에서만 접속만 허용하겠다는 뜻
 - '@%'로 작성하면 외부 접속을 허용하겠다는 뜻.
 - password를 'hkwon'으로 만들겠다는 뜻.
 - 설정 업데이트
 - `exit` 로 mysql을 빠져나온 후 php7.3-fpm 재시작.
 - php-mysql로 인하여 php이 설정이 변경되었다.
 - 아래와 같이 나오면 접속에 성공한 것이다.

```
# mysql 실행 후 접속
service mysql start
mysql
# Database 생성
CREATE DATABASE wordpress;
# wordpress databaste 사용
USE wordpress;
# User 생성
CREATE USER 'hkwon'@'localhost' IDENTIFIED BY 'hkwon';
GRANT ALL PRIVILEGES ON wordpress.* TO 'hkwon'@'localhost' WITH GRANT OPTION;
# update
FLUSH PRIVILEGES;
# exit
service php7.3-fpm restart
```



- Italiano
- 日本語
- Basa Jawa
- ქართული
- Taqbaylit
- Қазақ тілі
- ភាសាខ្មែរ
- ಕನ್ನಡ
- 한국어**
- کوردی
- ლიტუვų kalba
- Lietuvių kalba
- Latviešu valoda
- Македонски јазик
- മലയാളം
- Монгол
- मराठी
- Bahasa Melayu
- සමෘත
- Norsk bokmål

Continue



환영합니다

인기있는 5분 워드프레스 설치 절차에 오신 것을 환영합니다! 아래에 있는 정보를 채우기만 하면 세계 최고의 확장성과 강력한 개인 발행 플랫폼을 사용할 수 있습니다.

정보가 필요합니다

다음 정보를 제공하기 바랍니다. 걱정하지 마세요. 이 설정을 나중에 언제든지 바꿀 수 있습니다.

사이트 제목

사용자 이름

사용자 이름은 알파벳, 숫자, 스페이스, 밑줄, 하이픈, 마침표, @ 기호만 가능합니다.

비밀번호

6sEP#GZrZM4vOicLPU

숨기기

강함

중요: 로그인할 비밀번호가 필요할 것입니다. 안전한 위치에서 저장하기 바랍니다.

이메일 주소

계속하기 전에 이메일 주소를 다시 확인하세요.

검색 엔진 가시성

☐ 검색 엔진이 이 사이트를 검색하는 것을 차단하기

이 요청의 수행은 검색 엔진이 판단합니다.

워드프레스 설치하기

phpMyAdmin 설치

- Database 관리를 위한 phpmyadmin을 설치
 - 설치 후 압축을 해제하고 폴더이름을 phpmyadmin으로 수정하여 서버의 루트 디렉토리(/var/www/html/)에 위치시킨다.

```
# curl -O https://files.phpmyadmin.net/phpMyAdmin/5.0.4/phpMyAdmin-5.0.4-all-languages.tar.gz
# tar -xvf phpMyAdmin-5.0.4-all-languages.tar.gz
# build 할 때 error가 발생해 wget으로 바꿔 주었다.
# error가 발생하는 부분을 보니 phpMyAdmin/5.0.4를 5.0.2로 압축을 해제하고 있었다.
wget https://files.phpmyadmin.net/phpMyAdmin/5.0.4/phpMyAdmin-5.0.4-all-languages.tar.gz; \
tar -xvf phpMyAdmin-5.0.2-all-languages.tar.gz; \
mv phpMyAdmin-5.0.2-all-languages phpmyadmin
mv phpmyadmin /var/www/html/
```

- 쿠키 권한을 위해 blowfish 암호 설정
 - phpmyadmin 폴더에 config.sample.inc.php를 복사해 config.inc.php 파일을 만든다.
 - blowfish 암호 생성 사이트에서 키를 생성/복사하여 config.inc.php에 추가한다.

```
cp -rp var/www/html/phpmyadmin/config.sample.inc.php var/www/html/phpmyadmin/config.inc.php
vim var/www/html/phpmyadmin/config.inc.php
```

```
# blowfish 암호
# ' ' 에 입력
$cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
```

- phpmyadmin을 위한 DB table 생성

```
# mysql 접속
service mysql start
# create_table.sql 파일을 mysql로 redirect
mysql < var/www/html/phpmyadmin/sql/create_tables.sql;
```

- phpmyadmin local 접속 확인
- <https://127.0.0.1/phpmyadmin> 에 접속한 후 wordpress/wp-config.php 에서 설정한 ID와 PW를 입력하면 데이터베이스를 GUI로 편리하게 관리할 수 있다.

