

Ubuntu로 배포 기초환경 구축하기

2023.02.03



목차

- ✓ 자주 등장하는 명령어
- ✓ UFW install & Setting
- ✓ Nginx install & Setting
- ✓ Nginx Proxy & Port Forwarding
- ✓ HTTPS 적용하기
- ✓ 참고 : Jenkins Install & Setting

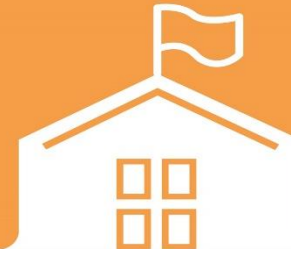


01

자주 등장하는 명령어



자주 등장하는 명령어



- ✓ `sudo apt-get update`
- ✓ `sudo apt-get upgrade`
- ✓ `sudo apt-get install <package name>`

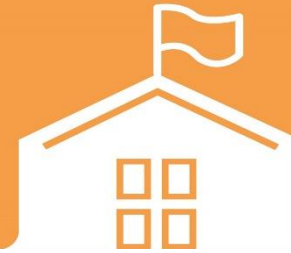
모든 설치 과정의 사전작업

02

UFW Install & Setting



UFW Install & Setting



✓ Ufw 설치 명령

✓ Sudo apt-get install ufw

```
wei@DESKTOP-DOGV PUB: ~$ sudo apt-get install ufw
[sudo] password for wei:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

명령어 실행을 통해 ufw가 설치 완료된 모습

UFW Install & Setting



- ✓ Ufw 상태 확인 명령
 - ✓ `sudo ufw status verbose`
 - ✓ `sudo ufw status`

```
wei@DESKTOP-DOGV PUB: ~$ sudo ufw status verbose  
Status: inactive
```

초기 status는 inactive인것을 볼 수 있다.



UFW Install & Setting

✓ UFW 활성화/ 비활성화

- ✓ `sudo ufw enable`
- ✓ `sudo ufw disable`

```
wei@DESKTOP-R2U96Q2: ~$ sudo ufw status
[sudo] password for wei:
Status: inactive
wei@DESKTOP-R2U96Q2: ~$ sudo ufw enable
ERROR: problem running ufw-init
```

어떤 방화벽 설정 없이 **enable** 명령 실행시 오류가 발생한다!



UFW Install & Setting

✓ UFW 기본 룰 확인

- ✓ deny : 들어오는 패킷에 대해서 전부 거부 (접속 불가능)
- ✓ allow : 나가는 패킷에 대해서 전부 허가 (접속 가능)

```
sudo ufw default deny  
sudo ufw default allow
```

```
sudo ufw allow 22 # 22번 ssh 포트를 모든 사용자가 접근할 수 있도록 허용
```

```
sudo ufw deny 22 # ssh 포트 아무도 접속 불가
```

```
sudo ufw allow from <ip> to any port 22 # 22번포트에 특정 사용자만 접근할 수 있도록 허용
```

```
sudo ufw deny from <ip> to any port 22
```

```
# 기본 정책이 Allow라면 다른 ip에서 22포트는 가능하지만, 특정 ip에서는 접속 불가능
```

기본 명령 예시



UFW Install & Setting

✓ 우리는 22, 80, 443번 포트를 기본으로 설정하고 진행하겠습니다.

```
sudo ufw allow 22
sudo ufw enable
sudo ufw allow 80
sudo ufw allow 443
sudo ufw state
```

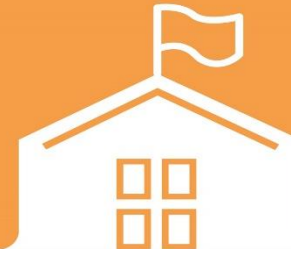
기본 명령 예시

03

Nginx install & Setting



Nginx install & Setting

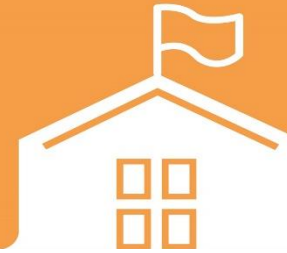


✓ Nginx 설치 명령

- ✓ apt-get update & apt-get upgrade 는 사전에 해 줄것
- ✓ Sudo apt-get install nginx

```
wei@DESKTOP-DOGVPLB:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 nginx nginx-common nginx-core
0 upgraded, 13 newly installed, 0 to remove and 186 not upgraded.
Need to get 1218 kB of archives.
After this operation, 4294 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Nginx install & Setting



- ✓ Ufw - 현재 사용가능한 서비스별 ufw 설정값들을 확인
 - ✓ `sudo ufw app list`

```
wei@DESKTOP-DOGV PUB:~$ sudo ufw app list
Available applications:
Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH
```



Wsl, EC2에는 기본적인 리스트로 들어와 있음

- Nginx HTTP: 포트 80번만 허용
- Nginx HTTPS: 포트 443번만 허용
- Nginx Full: 포트 80, 443번 허용



Nginx install & Setting

- ✓ Ufw 상태 확인
 - ✓ `sudo ufw state`

```
wei@DESKTOP-DOGVPLB: ~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
22/tcp ALLOW 59.20.195.133
80 ALLOW Anywhere
443 ALLOW Anywhere
Anywhere ALLOW 52.20.195.133/tcp
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
```

설정값에 따라 달라지게 된다



Nginx install & Setting

- ✓ Ufw 상태 확인
 - ✓ `sudo ufw state`

IPv4

```
wei@DESKTOP-DOGVPLB: ~$ sudo ufw status
Status: active

To Action From
22 ALLOW Anywhere
22/tcp ALLOW 59.20.195.133
80 ALLOW Anywhere
443 ALLOW Anywhere
Anywhere ALLOW 52.20.195.133/tcp
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
```

설정값에 따라 달라지게 된다



Nginx install & Setting

✓ Ufw 상태 확인

✓ `sudo ufw state`

```
wei@DESKTOP-DOGVPLUB:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
22/tcp ALLOW 59.20.195.133
80 ALLOW Anywhere
443 ALLOW Anywhere
Anywhere ALLOW 52.20.195.133/tcp
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
```

IPv6

설정값에 따라 달라지게 된다



Nginx install & Setting

✓ Nginx ufw 설정 값 적용하기

✓ sudo ufw allow “설정값”

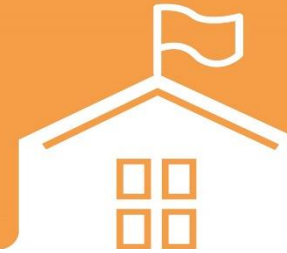
ex) sudo ufw allow “HTTP”

```
wei@DESKTOP-DOGVPIB: ~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
22/tcp ALLOW 59.20.195.133
80 ALLOW Anywhere
443 ALLOW Anywhere
Anywhere ALLOW 52.20.195.133/tcp
Nginx HTTP ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
Nginx HTTP (v6) ALLOW Anywhere (v6)
```

Ufw에 Nginx 설정이 추가가 된 모습

Nginx install & Setting



함께가요 미래로!
Enabling People

- ✓ Nginx 실행 상태 확인
 - ✓ `service nginx status`

```
wei@DESKTOP-DOGVPU: ~$ service nginx status
* nginx is running
```

Nginx 실행이 완료되었습니다

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.

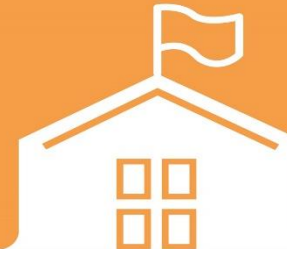
80포트 접속 시 보이는 화면

04

Nginx Proxy & Port Forwarding



Nginx Proxy & Port Forwarding



함께가요 미래로!
Enabling People

- ✓ EC2 서버 접속
- ✓ MobaXterm 사용
 - ✓ 설치 링크 : <https://mobaxterm.mobatek.net/download.html>

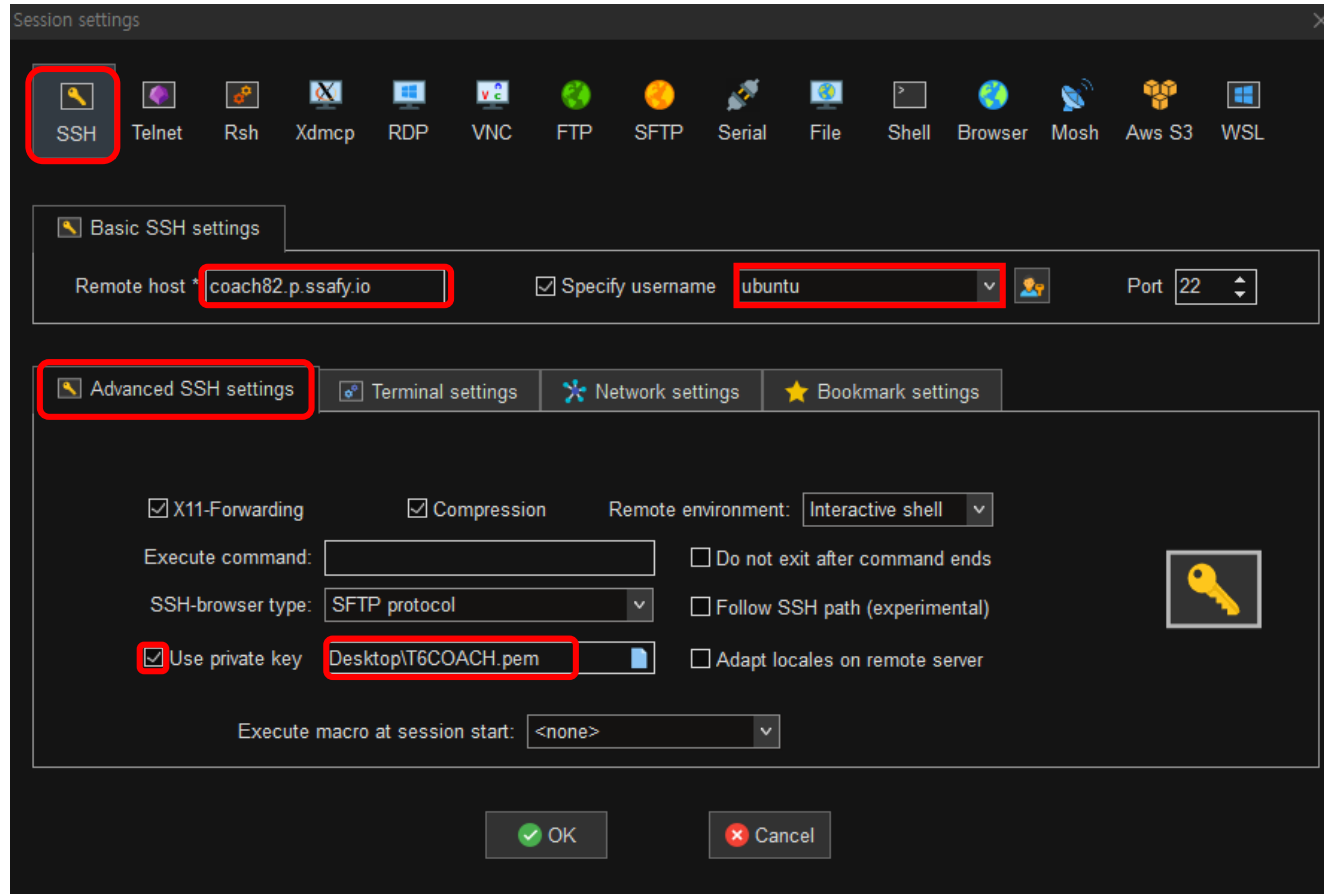


- ✓ Putty + WinSCP 기능을 통합으로 제공 (CMD 환경 + 파일 전송)
- ✓ 깔끔한 GUI

Nginx install & Setting

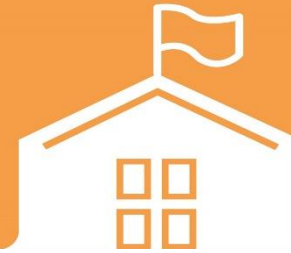


함께가요 미래로!
Enabling People



- ✓ 상단 왼쪽 메뉴의 SSH 클릭
- ✓ Remote host -> 부여받은 주소
- ✓ Specify username -> ubuntu
- ✓ Advanced SSH settings
 - ✓ Use private key 체크
 - ✓ 지금 받은 pem키 업로드

Nginx Proxy & Port Forwarding

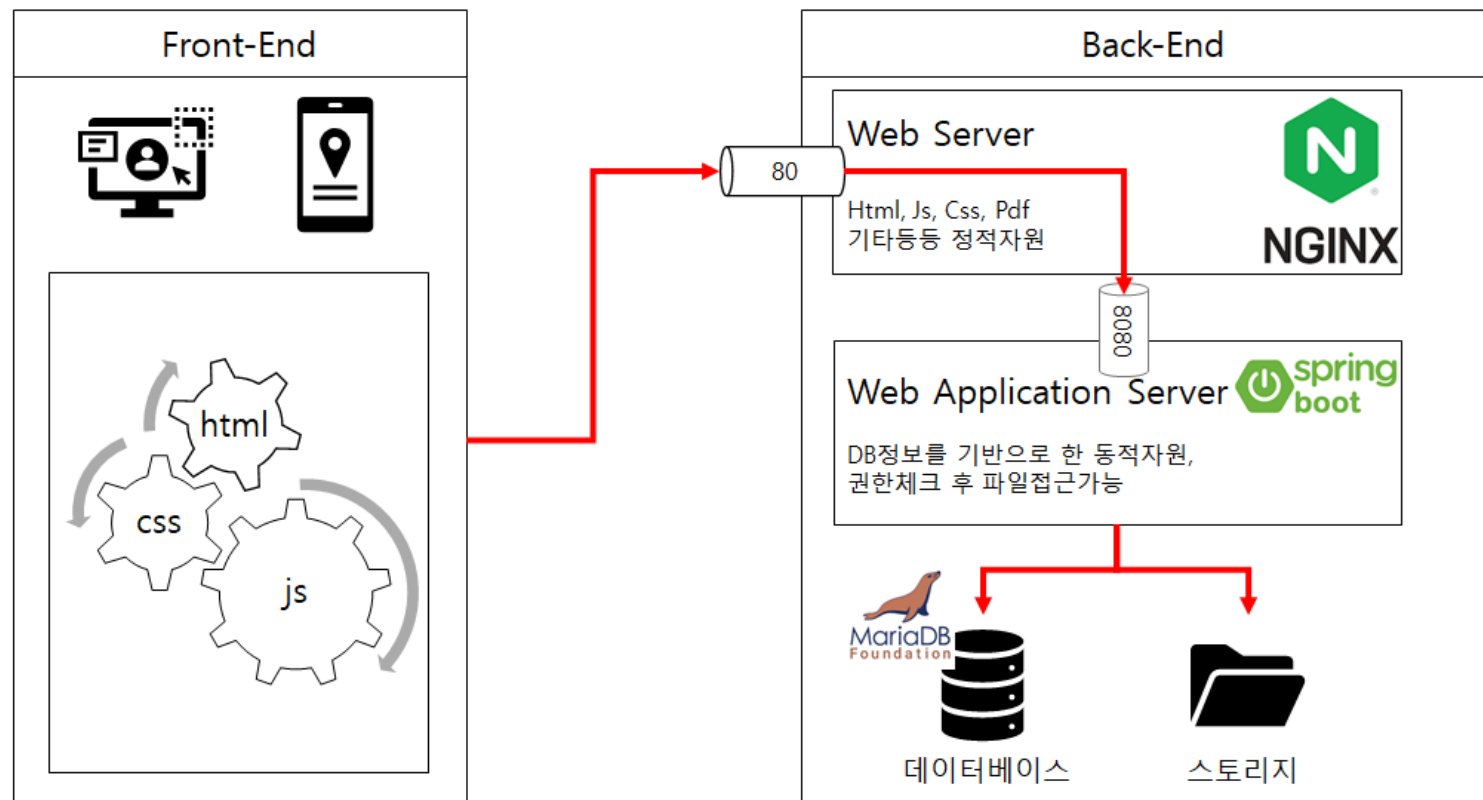


- ✓ 웹 서버
 - ✓ HTML, CSS, JS 이미지 등과 같은 정적인 정보를 사용자 에게 전송
- ✓ 리버스 프록시
 - ✓ 서버 내부망의 앞단에서 요청을 중계해주는 역할을 함

Nginx Proxy & Port Forwarding



함께가요 미래로!
Enabling People



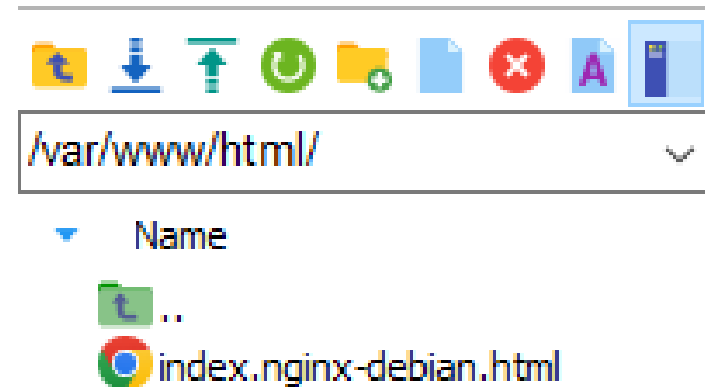
출처 - <https://papababo.tistory.com/entry/nginx-Spring-boot-를-위한-Reverse-Proxy-설정하기>

Nginx Proxy & Port Forwarding



✓ Nginx default setting

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```



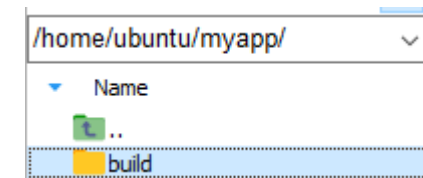
Nginx Proxy & Port Forwarding



함께가요 미래로!
Enabling People

✓ npm run build 명령어로 프로젝트 빌드

.git	2023-01-18 오후 4:27	파일 폴더	
build	2023-01-31 오전 11:56	파일 폴더	
node_modules	2023-01-31 오전 9:49	파일 폴더	
public	2023-01-18 오후 4:27	파일 폴더	
src	2023-01-20 오전 11:54	파일 폴더	
.gitignore	2023-01-18 오후 4:27	텍스트 문서	1KB
package	2023-01-31 오전 9:49	JSON 원본 파일	1KB
package-lock	2023-01-31 오전 9:49	JSON 원본 파일	1,867KB
README	2023-01-18 오후 4:27	Markdown 원본 ...	4KB

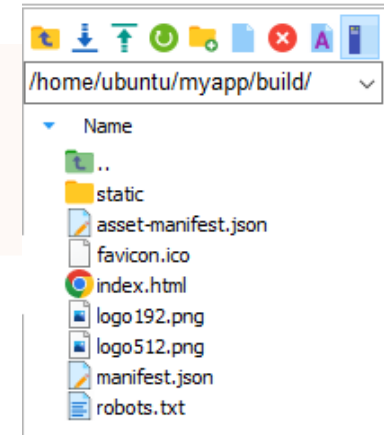


Nginx Proxy & Port Forwarding



함께가요 미래로!
Enabling People

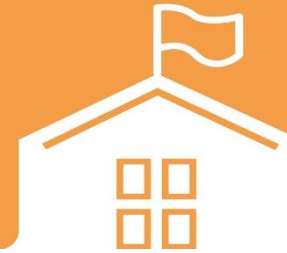
✓ Nginx config 수정 (/etc/nginx/sites-available/default)



```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

```
#  
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /home/ubuntu/myapp/build/;  
    index index.html index.htm;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ /location;  
    }  
}
```

Nginx Proxy & Port Forwarding



- ✓ `sudo service nginx restart` 후 배포 된 프론트 화면

안녕하세요 여러분 인간집사 박동준입니다.

api 요청

요청 상황 : 요청 전
결과값 : 없써염

Nginx Proxy & Port Forwarding



함께가요 미래로!
Enabling People

✓ Spring boot 프로젝트 배포하기

```
ubuntu@ip-172-26-4-177:~/myapp$ java -jar crudtest-1-0.0.1-SNAPSHOT.jar
```

```
ubuntu@ip-172-26-4-177:~/myapp$ java -jar crudtest-1-0.0.1-SNAPSHOT.jar
```

```

  ____ _
 / ___ \
/ /   _ \
/_/_/\___/
:: Spring Boot ::
                (v2.7.0)

```

```
2023-01-31 17:00:06.707 INFO 900375 --- [main] com.ssafy.crudtest.Crudtest1Application
ion using Java 11.0.17 on ip-172-26-4-177 with PID 900375 (/home/ubuntu/myapp/crudtest-1-0.0.1-SNAPS
/home/ubuntu/myapp)
2023-01-31 17:00:06.716 INFO 900375 --- [main] com.ssafy.crudtest.Crudtest1Application
ling back to 1 default profile: "default"
2023-01-31 17:00:08.049 INFO 900375 --- [main] .s.d.r.c.RepositoryConfigurationDelegate
JPA repositories in DEFAULT mode.
2023-01-31 17:00:08.109 INFO 900375 --- [main] .s.d.r.c.RepositoryConfigurationDelegate
itory scanning in 47 ms. Found 1 JPA repository interfaces.
2023-01-31 17:00:08.974 INFO 900375 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer
t(s)= 8080 (http)
```

▼ Name



..



build

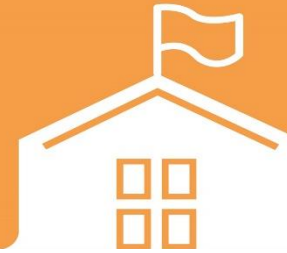


src



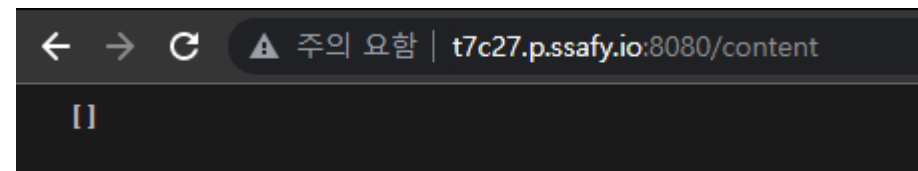
crudtest-1-0.0.1-SNAPSHOT.jar

Nginx Proxy & Port Forwarding



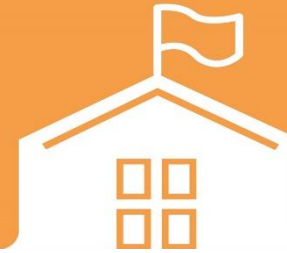
✓ 요청 정보 확인

8080 포트로 실행된 spring



예시 api 요청에 따른 response data

Nginx Proxy & Port Forwarding



✓ 프록시 설정하기

포트는 어디
... ?

```
function App() {
  const [response, setResponse] = useState(null);
  const [status, setStatus] = useState(false)
  const getAxios = async () => await axios.get("/content").then(
    res => {
      setResponse(res.data)
      setStatus(true)
      console.log(res.data, '데이터예요')
    }
  ).catch((err) => console.log(err, '에러'))

  return (
    <div className="App">
      <h1>안녕하세요 여러분 인간집사 박동준입니다.</h1>
      <Button onClick={() => getAxios()}>api 요청</Button>
      <div>요청 상황 : {status === false? <>요청 전</> : <>요청 후</></div>
      <div>결과값 : {response ? <>
        {response.map((v,i) => {
          return (
            <div id={v.title + i}>
              <div>path :{v.path}</div>
              <div>title :{v.title}</div>
              <div>uid :{v.uid}</div>
            </div>
          )
        })}
      </> : <>없쨌염</></div>
    </div>
  );
}
```

Nginx Proxy & Port Forwarding



- ✓ Nginx config 수정 (/etc/nginx/sites-available/default)

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /home/ubuntu/myapp/build/  
    index index.html index.htm;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
  
        try_files $uri $uri/ /index.html;  
    }  
  
    location /content {  
        proxy_pass http://t7c27.p.ssafy.io:8080;  
    }  
}
```

<http://t7c27.p.ssafy.io/content>



Nginx ^_^



<http://t7c27.p.ssafy.io:8080/content>

Nginx Proxy & Port Forwarding



- ✓ Nginx config 수정 (/etc/nginx/sites-available/default)

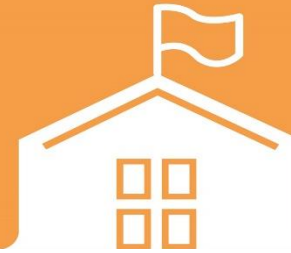
```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /home/ubuntu/myapp/build/  
    index index.html index.htm;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ /index.html;  
    }  
  
    location /content {  
        proxy_pass http://t7c27.p.ssafy.io:8080;  
    }  
}
```

<http://t7c27.p.ssafy.io/content>

Nginx ^_^

<http://t7c27.p.ssafy.io:8080/content>

Nginx Proxy & Port Forwarding



✓ 결과 확인하기

안녕하세요 여러분 인간집사 박동준입니다.

api 요청

요청 상황 : 요청 후

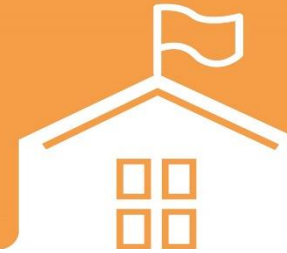
결과값 :

path :KakaoTalk_20230110_134943259_01.jpg

title :1234

uid :1

Nginx Proxy & Port Forwarding



- ✓ CI(Continuous Integration)
 - ✓ 지속적인 통합
 - ✓ 빌드/ 테스트 자동화 과정
- ✓ CD(Continuous Delivery/Deployment)
 - ✓ 지속적인 서비스 제공/배포
 - ✓ 배포 자동화 과정



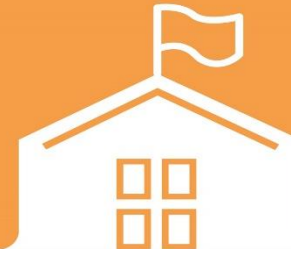
Jenkins

05

HTTPS 적용하기



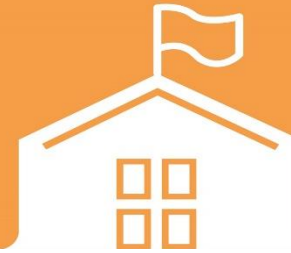
HTTPS 적용하기



- ✓ 적용 전 필수사항!!!
 - ✓ Ufw 80 Port allow
 - ✓ Server
 - ✓ Nginx

필수입니다!!! 필수!!!!

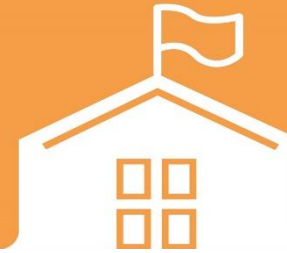
HTTPS 적용하기



- ✓ **Certbot 설치 및 Let's Encrypt에서 SSL인증서 발급**
 - ✓ `sudo snap install certbot -classic`
 - ✓ `sudo certbot -nginx` (80번 포트는 유틸 상태로 개방 되어 있어야 한다.)

- ✓ **이메일 입력**
- ✓ **약관 동의**
- ✓ **도메인 주소 입력 ex) t7c... (http:// -> 생략)**

HTTPS 적용하기



✓ 변경사항 확인 (/etc/nginx/sites-available/default)

```
server {
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/t7c27.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/t7c27.p.ssafy.io/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    root /home/ubuntu/myapp/build/;
    index index.html index.htm;
    server_name t7c27.p.ssafy.io; # managed by Certbot

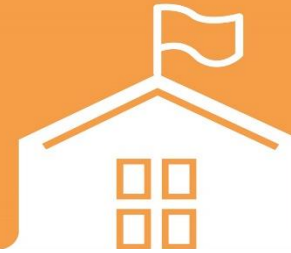
    #
    location / {
        # as directory, then fall back to displaying a 404.
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }

    location /content {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_pass http://localhost:8080;
    }
}
```

```
}
server {
    if ($host = t7c27.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

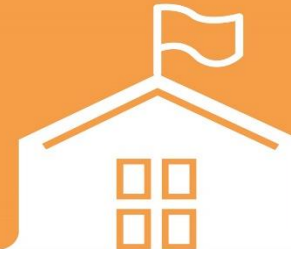
    listen 80 ;
    listen [::]:80 ;
    server_name t7c27.p.ssafy.io;
    return 404; # managed by Certbot
}
```

HTTPS 적용하기



뭔가 많이 바뀐 것 같죠??
작동하는 순서대로 확인 해 보겠습니다.

HTTPS 적용하기

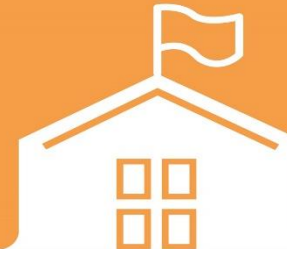


✓ Step 1

- ✓ 80 포트로 들어온 요청 중, Host header가 domain 이름이라면,
- ✓ 443 port 즉 Https로 redirect 시켜줍니다.

```
}  
server {  
    if ($host = t7c27.p.ssafy.io) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    listen 80 ;  
    listen [::]:80 ;  
    server_name t7c27.p.ssafy.io;  
    return 404; # managed by Certbot  
}
```


HTTPS 적용하기



✓ Step 2

✓ 443 port 로 redirect 된 요청을 수신합니다.

```
server {  
    listen [::]:443 ssl ipv6only=on; # managed by Certbot  
    listen 443 ssl; # managed by Certbot  
    ssl_certificate /etc/letsencrypt/live/t7c27.p.ssafy.io/fullchain.pem; # managed by Certbot  
    ssl_certificate_key /etc/letsencrypt/live/t7c27.p.ssafy.io/privkey.pem; # managed by Certbot  
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  
  
    root /home/ubuntu/myapp/build/;  
    index index.html index.htm;  
    server_name t7c27.p.ssafy.io; # managed by Certbot  
}
```

HTTPS 적용하기



✓ Step 3

✓ Proxy pass가 변동되었습니다.

```
root /home/ubuntu/myapp/build/;
index index.html index.htm;
server_name t7c27.p.ssafy.io; # managed by Certbot

#
# location / {
#     # as directory, then fall back to displaying a 404.
#     proxy_set_header X-Real-IP $remote_addr;
#     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
#     proxy_set_header Host $http_host;
# }

location /content {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_pass http://localhost:8080;
```

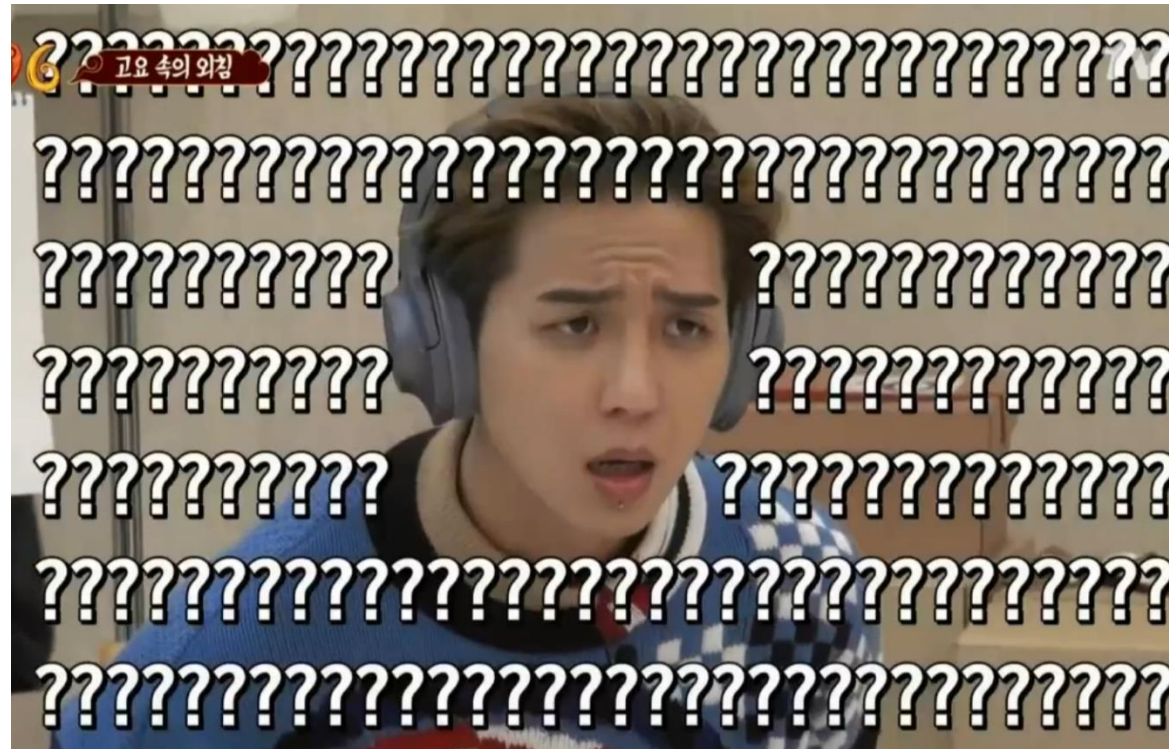
/content 라는 요청이 감지된다면,

Localhost:8080 번으로 pass가 넘어가게
됩니다.

기존 “도메인이름” 에서
localhost로 변경이 되었네요



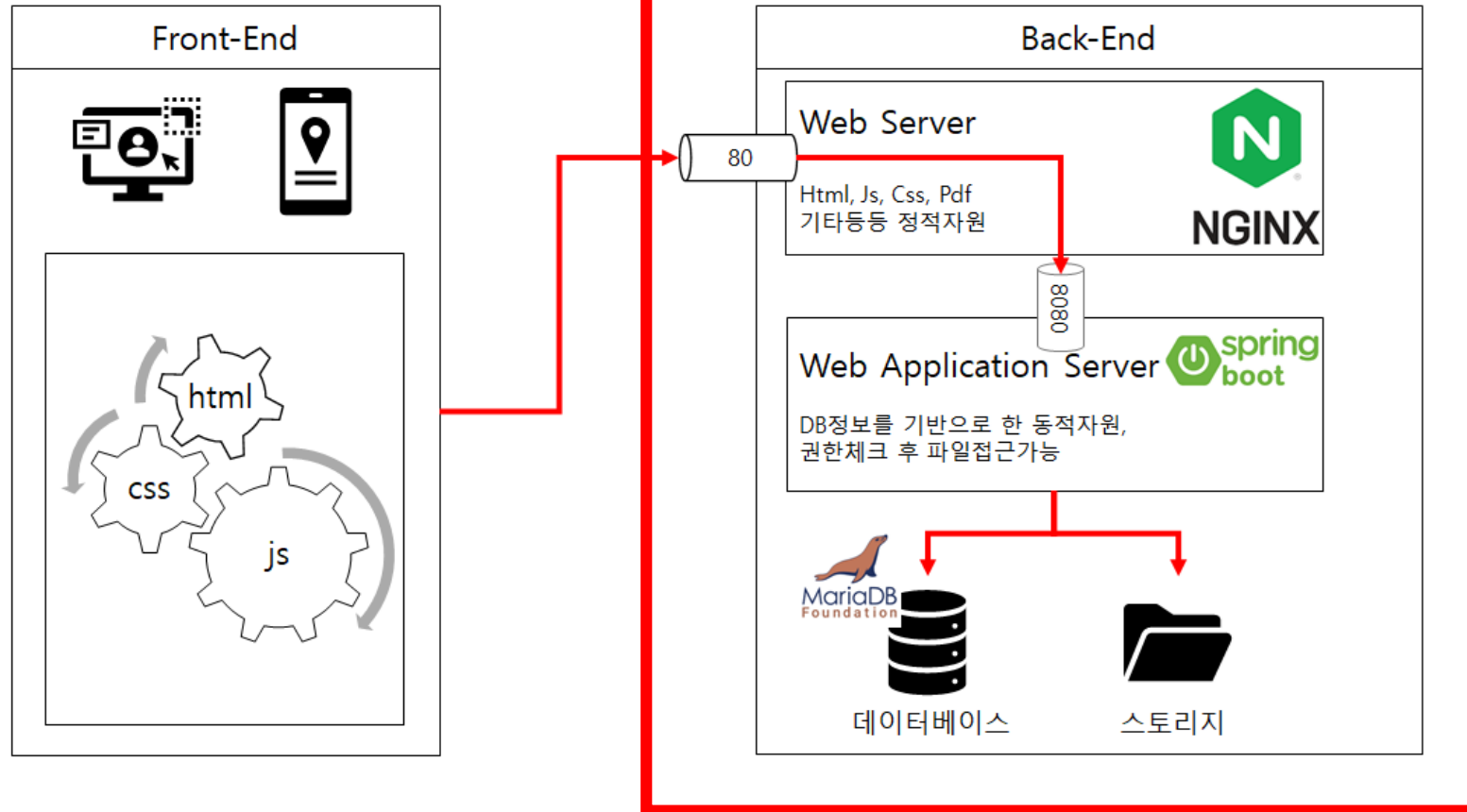
잉?! 갑자기 Localhost??????



HTTPS 적용하기



함께가요 미래로!
Enabling People



동일한 도메인 서버 구축 환경
“내부”에서 작동하기 때문에

도메인 주소 대신 **localhost** 가
사용이 가능함

굳이 외부망으로 돌릴 필요가 없음

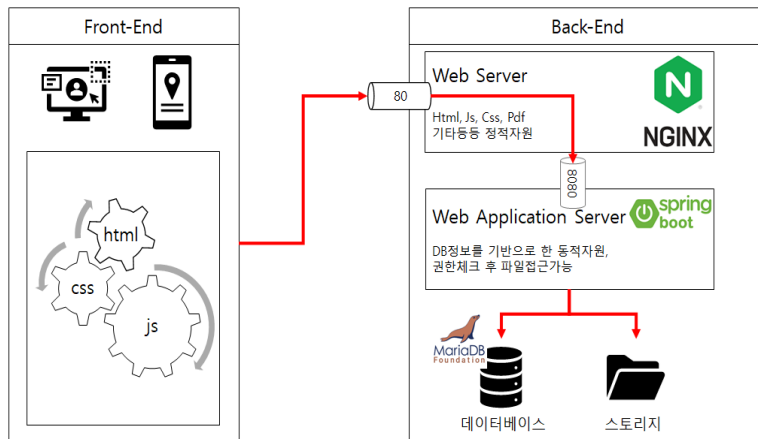
HTTPS 적용하기



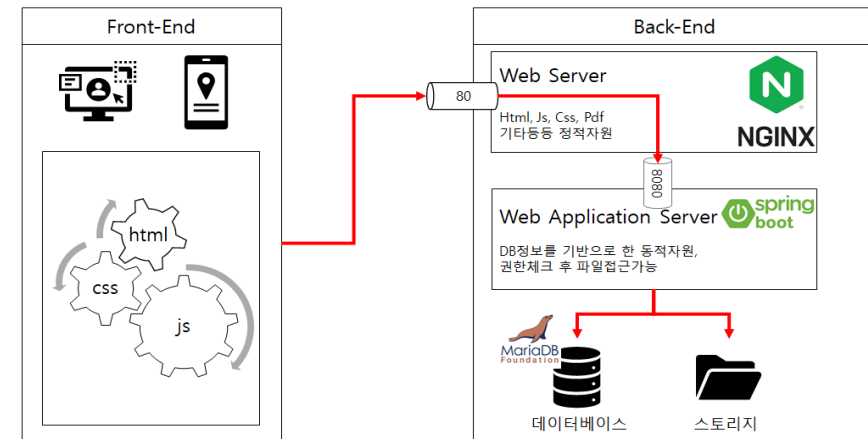
함께가요 미래로!
Enabling People

✓ 각 도메인의 SSL여부에 따라 달라지게 됩니다.

1번 서버: https



2번 서버: https



https (O)

http (X)

HTTPS 적용하기



HTTPS 적용하기



✓ Step 4

✓ Location / 주식 처리 및 root 경로 끌어올리기

```
root /home/ubuntu/myapp/build/;
index index.html index.htm;
server_name t7c27.p.ssafy.io; # managed by Certbot

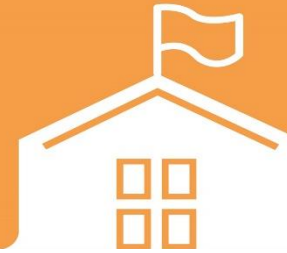
#
#   location / {
#       # as directory, then fall back to displaying a 404.
#       proxy_set_header X-Real-IP $remote_addr;
#       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
#       proxy_set_header Host $http_host;
#
#   }

location /content {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_pass http://localhost:8080;
```

아까 언급 드렸던 80포트에 대한 리스는
없어야 합니다.

대신 root 경로에 대한 설정을 제일 위로
끌어올렸습니다.

HTTPS 적용하기



✓ 만약.. Location이 아닌 domain 주소가 있었다면?

```
root /home/ubuntu/myapp/build/;
index index.html index.htm;
server_name t7c27.p.ssafy.io; # managed by Certbot

#
# location / {
#     # as directory, then fall back to displaying a 404.
#     proxy_set_header X-Real-IP $remote_addr;
#     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
#     proxy_set_header Host $http_host;
# }

#
location /content {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_pass http://localhost:8080;
```

윙? 도메인 요청이 왔네?

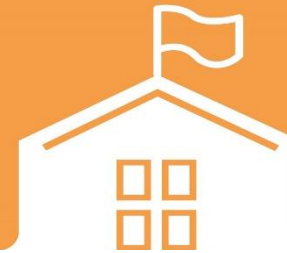
443으로 redirect!

```
}
server {
    if ($host = t7c27.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 ;
    listen [::]:80 ;
    server_name t7c27.p.ssafy.io;
    return 404; # managed by Certbot
}
```

Domain 주소였다면????

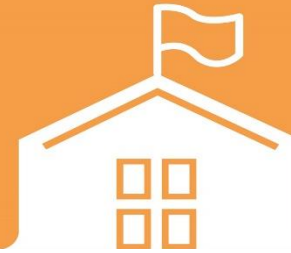
HTTPS 적용하기



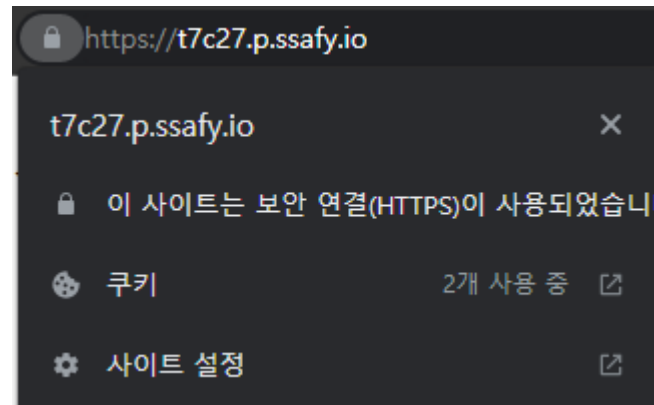
함께가요 미래로!
Enabling People



HTTPS 적용하기



✓ 결론적으로 https 연결이 잘 이루어진 모습



감사합니다



참고

Jenkins Install & Setting





Jenkins install & Setting

✓ JDK 설치

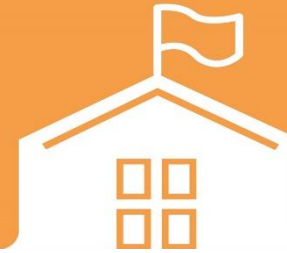
✓ 동일하게 update & upgrade 진행

✓ `sudo apt-get install openjdk-11-jre openjdk-11-jdk`

```
wei@DESKTOP-DOGVPLB:~$ java -version
openjdk version "11.0.17" 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu220.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu220.04, mixed mode, sharing)
```

JDK가 설치완료 된 모습,

Jenkins install & Setting



✓ Jenkins keyvalue update

✓ `wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -`

✓ `sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >`

`W /etc/apt/sources.list.d/jenkins.list'`

```
wei@DESKTOP-R2U96Q2:/etc/nginx$ sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
OK
wei@DESKTOP-R2U96Q2:/etc/nginx$ echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.
list.d/jenkins.list'echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenki
ns.list'
tee: '/etc/apt/sources.list.d/jenkins.list'echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/ap
t/sources.list.d/jenkins.list': No such file or directory
deb http://pkg.jenkins.io/debian-stable binary/
```



Jenkins install & Setting

✓ Jenkins 설치하기

- ✓ `sudo apt-get update`
- ✓ `sudo apt-get install jenkins`

```
wei@DESKTOP-DOGVPU:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 94.7 MB of archives.
After this operation, 96.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Ufw에 Nginx 설정이 추가가 된 모습



Jenkins install & Setting

- ✓ Jenkins 서버 포트 번호 변경
 - ✓ 필요에 따라 변경 해 줍니다.
 - ✓ Default port : 8080
- ✓ 명령어
 - ✓ `sudo vi /etc/default/jenkins`

```
# port for HTTP connector (default 8080; disable with -1)  
HTTP_PORT=8080
```

변경이 필요할 때만 바꿔준다



Jenkins install & Setting

- ✓ 재시작 하기
 - ✓ 설정 값들이 변경 되었다면 재시작이 필요함
- ✓ 명령어
 - ✓ `sudo service Jenkins (restart or start)`

```
wei@DESKTOP-DOGVPLB:~$ sudo service jenkins status
Correct java version found
* jenkins is not running
```

설치 후 아직 실행되어지지 않은 상태

```
wei@DESKTOP-R2U96Q2:/etc/nginx$ sudo service jenkins restart
* Restarting Jenkins Automation Server jenkins
Correct java version found
Correct java version found
Setting up max open files limit to 8192
```

```
wei@DESKTOP-DOGVPLB:~$ sudo service jenkins start
* Starting Jenkins Automation Server jenkins
Correct java version found
Setting up max open files limit to 8192
```

Restart or start 를 통해 실행시켜주자



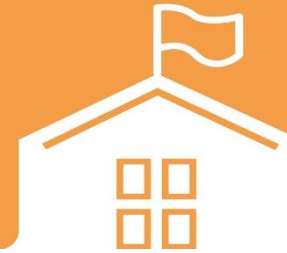
Jenkins install & Setting

- ✓ Jenkins 서비스 상태 확인 및 초기 비밀번호 확인
 - ✓ `sudo service jenkins status`
 - ✓ `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

```
wei@DESKTOP-R2U96Q2:/etc/nginx$ sudo service jenkins status
Correct java version found
* jenkins is running
```

비밀번호 확인 후 따로 기록해두는 것 추천!

Jenkins install & Setting



함께가요 미래로!
Enabling People

✓ Localhost 이동 후 비밀번호 입력 및 설치

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Create First Admin User

계정명

암호

암호 확인

이름

이메일 주소

Jenkins 2.375.2

Skip and continue as adminSave and Continue

비밀번호 확인 후 따로 기록해두는 것 추천!

Jenkins install & Setting



함께가요 미래로!
Enabling People

✓ 설치 완료~!

The screenshot shows the Jenkins dashboard. At the top is a black header with the Jenkins logo, a search bar, and user information. Below the header is a sidebar with navigation links: '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리', and 'My Views'. The main content area has a heading 'Jenkins에 오신 것을 환영합니다.' followed by a paragraph explaining the page's purpose. Below this are two sections: 'Start building your software project' with a 'Create a job' button, and 'Set up a distributed build' with buttons for 'Set up an agent', 'Configure a cloud', and a link to 'Learn more about distributed builds'. The footer contains 'REST API' and 'Jenkins 2.375.2'.

Jenkins 검색 (CTRL+K) ? 1 박동준 로그아웃

Dashboard >

+ 새로운 Item 상세 내용 입력

사람

빌드 기록

Jenkins 관리

My Views

빌드 대기 목록 ▼

빌드 대기 항목이 없습니다.

빌드 실행 상태 ▼

1 대기 중

2 대기 중

Jenkins에 오신 것을 환영합니다.

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

REST API Jenkins 2.375.2