



# 빌드 및 배포

## EC2로 배포하기

### ▼ EC2 인스턴스 생성하기

1. AWS EC2 대시보드에 접속한 뒤 우측 상단의 **인스턴스 시작**을 클릭
2. 인스턴스의 이름 설정
3. 인스턴스의 이미지 선택
  - **Ubuntu Server 20.04 LTS** 선택
4. 새 키페어 생성을 클릭해 새로운 **키 페어** 생성
5. 키 페어 이름을 지정하고 키 페어 유형과 프라이빗 키 파일 형식 설정 후 키페어 생성을 클릭해 키 페어 파일을 다운로드
6. **인스턴스 시작** 클릭

### ▼ putty로 접속하기

1. PuTTYgen 열기
2. Load 버튼을 눌러 갖고있던 pem키 불러오기
  - 확장자를 All Files로 바꾸기
3. Save private key 버튼을 눌러 ppk 생성하기
4. PuTTY 열기
5. Host Name (or IP address) 입력
6. 왼쪽 바에서 Connection > SSH > Auth > Credentials 에서 생성한 ppk 넣기
7. 왼쪽 바에서 Session을 눌러 session 이름 작성 후 Save 버튼 눌러 저장
8. 저장한 session 클릭 후 open버튼 클릭
9. 부여받은 아이디인 ubuntu로 로그인

## ▼ Git 설치

1. Git 설치하기

```
sudo apt-get install git
```

2. 설치 확인하기

```
git --versio
```

3. repo clone하기

```
git clone [주소]
```

4. git pull 하기

```
git pull origin [branch 이름]
```

## ▼ JAVA 설치

1. JAVA 설치하기

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

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

2. 설치 확인

```
java -version
```

```
javac -version
```

## ▼ MariaDB 설치

1. MariaDB 설치하기

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

 중간에 Y/N 나오면 Y 입력

2. MariaDB 접속하기

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

## ▼ Redis 설치

1. Redis 설치

```
sudo apt-get install redis-server
```

2. 설치 후 버전 확인

```
redis-server --version
```

3. redis.conf 파일 열어서 설정

```
sudo nano /etc/redis/redis.conf
```

bind 127.0.0.1 ::1 을 찾아 바꾸기

```
bind 0.0.0.0 ::1
```

requirepass를 찾아서 비밀번호 설정

```
requirepass [비밀번호]
```

maxmemory와 maxmemory-policy를 찾아서 바꾸기

```
maxmemory 1g
```

```
maxmemory-policy allkeys-lru
```

4. 변경내용을 적용하기 위해 Redis 다시 시작

```
sudo systemctl restart redis-server.service
```

5. Redis가 6379 포트를 쓰고 있는지 확인

```
netstat -nlpt | grep 6379
```

## ▼ Docker 설치

1. 오래된 버전 삭제하기

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

2. Docker 설치하기

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

3. 설치 확인

```
docker --version
```

4. sudo 없이 docker 실행

```
sudo usermod -aG docker [username]
```

5. docker hub에서 image 끌고오기

```
docker pull [image tag]
```

6. docker에 존재하는 image 확인하기

```
docker images
```

## ▼ React 배포

1. Dockerfile이 있는 위치로 이동해서 빌드

```
docker build -t <이미지 이름> . : 빌드할 파일 Dockerfile로 명시적 지정
```

2. docker image로 container run 시키기

```
docker run -d -p [host port 번호]:[container port 번호] --name [container 이름]  
[image 이름]
```

## ▼ Spring Boot 배포

1. Back-End 폴더로 이동

```
cd BE
```

2. 프로젝트 root 폴더에서

```
./gradlew clean build --exclude-task test
```

 : 빌드된 파일 지웠다가 다시 빌드 실행

3. jar 파일 실행하기

```
cd build/libs
```

```
nohup java -jar [jar파일.jar] &
```