

**01**

**AWS EC2 배포**

# 01. SpringBoot 배포 순서

## ■ AWS EC2 인스턴스 생성

- 키페어 저장

## ■ Spring Boot Build

- Gradle – tasks- build- bookjar

## ■ AWS EC2 인스턴스 접속

- `chmod 400 "20241211_ServerKey.pem"`
- `ssh -i "20241211_ServerKey.pem" ubuntu@ec2-43-202-46-116.ap-northeast-2.compute.amazonaws.com`

## ■ Sudo su

## ■ Apt update

## ■ Apt upgrade

## ■ Java -v

# 01. SpringBoot 배포 순서

- apt install openjdk-17-jre-headless
- Apt install mysql-server
- 다른 터미널을 열어 EC2에 파일 복사
  - `scp -i 20241211_ServerKey.pem "springboot build 파일명.jar" ubuntu@ec2-43-202-46-116.ap-northeast-2.compute.amazonaws.com:/home/ubuntu`
  - `scp -i 20241211_ServerKey.pem "export 한 데이터베이스 테이블명.sql" ubuntu@ec2-43-202-46-116.ap-northeast-2.compute.amazonaws.com:/home/ubuntu`
- EC2에서 Mysql `-u root -p` 접속
- Create database shop;
- Use shop;

## 01. SpringBoot 배포 순서

---

- Source “export한 데이터베이스 테이블.sql”;
- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '1234';
- Quit;
- Java -jar “springboot build 파일명.jar”

## 02. React 배포 순서

- EC2에서 탄력적 아이피 생성 인스턴스에 연결
- Frontend 요청 Ip 주소를 EC2의 아이피로 변경
- SpringBoot의 Frontend 아이피를 EC2의 아이피로 변경
- Frontend에서 npm run build
- SpringBoot 빌드 파일 .jar를 EC2 /home/ubuntu에 복사
- Frontend 빌드 폴더안의 모든 내용을 EC2  
/home/ubuntu/build안에 복사
- 서버의 Background실행을 위해 nohup설치
- apt update
- apt install coreutils

## 02. React 배포 순서

---

- `nohup java -jar Auth2JWT-0.0.1-SNAPSHOT.jar &`
- Serve 설치를 위한 `node.js` 설치
- `apt install -y nodejs`
- `apt install npm`
- `npm install -g serve`
- `serve -s build`