aws

빌드/ 배포 가이드

버튼을 누르면 목차를 보실 수 있습니다.

▼ 목차

1. Docker 및 Docker Compose 설치

> 프로젝트는 Docker와 Docker-Compose를 기반으로 돌아갑니다.
> 해당 툴의 설치는 필수적입니다!
> * Ubuntu 20.04, Docker 20.10.12, Docker-Compose 1.29.2 를 기준으로 작성했습니다.

1.1 관리자 권한 실행

`sudo -i`

1.2 저장소 설정

```
apt-get update
apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
```

1.3 Docker 설치

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

1.4 Docker Compose 1.29.2 버전 설치

```
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/d
```

1.5 바이너리에 실행 권한 추가

`chmod +x /usr/local/bin/docker-compose`

1.6 Docker Compose 버전 확인

`docker-compose -v`

2. Jenkins 설치

> Docker Compose 파일의 volumes 경로를 Jenkins 폴더를 기준으로 설정했습니다!
> * Jenkins를 설치하고 싶지 않다면 , S06P12A201/docker-compose.yml 파일 안의 volumes 경로를

> `/home/jenkins/workspace/INVIEW/` → `./` 바꾸고, nginx의 volumes 경로도 `/etc` → `./`로 바꿔주세요.

```
docker run -d \

--name jenkins \

-p 9090:8080 \

-v /home/jenkins:/var/jenkins_home \

-v /var/run/docker.sock:/var/run/docker.sock \

-v $(which docker):/usr/bin/docker \

-v /usr/local/bin/docker-compose:/usr/local/bin/docker-compose \

-u root jenkins/jenkins
```

3. SSL 인증서 갱신

> 비디오, 오디오 기능은 https에서만 동작하기 때문에 서비스의 주요 기능을 사용하기 위해 인증서 설정은 필수적입니다!
> 현재 인증서는 Let's Encrypt 90일간 유효한 인증서로 2022/05/17 09:25:30까지 사용 가능 합니다. 그 이후에는 갱신이 필요합니다.

3.1 인증서 만료 날짜 확인

`openssl s_client -connect i6a201.p.ssafy.io:443 2>/dev/null | openssl x509 -noout -dates`

3.2 갱신

`certbot renew`

4. 배포

> INVIEW 프로젝트는 Jenkins와 Git을 이용해 CI/CD를 진행하기 때문에 4. 배포 과정이 필요하지 않습니다!
> 4. 배포 과정은 로컬에서 실행할 경우만 해당됩니다.

*사전 작업

node 16.13.0 , gradle 6.7 설치

4.1 frontend 빌드(backend 폴더에서 진행)

```
npm install
npm run build
```

4.2 backend 빌드(frontend 폴더에서 진행)

```
rm -r src/main/generated
gradle clean
gradle build
```

4.3 배포(프로젝트 폴더에서 진행)

`docker-compose up --build -d`