

# 백엔드 배포 및 실행

## JDK 설치

도커 설치 전에 jdk 맞는 버전을 설치해야 함  
설치 후,

## 도커

<https://seongwon.dev/DevOps/20220715-CICD구축기1/>

여기서

## 4.2. Docker 설치

도커 설치 내용은 [Docker docs](#)에 더 상세히 나와있습니다.

1. `apt` package index 최신으로 업데이트 및 `apt` 가 HTTPS를 통해 저장소를 사용할 수 있도록 패키지를 설치

여기 부터 보면서 도커만 설치

## 젠킨스

<https://velog.io/@ssstark0529/EC2-서버-하나에-도커와-젠킨스를-활용해-자동배포를-구현해보자>

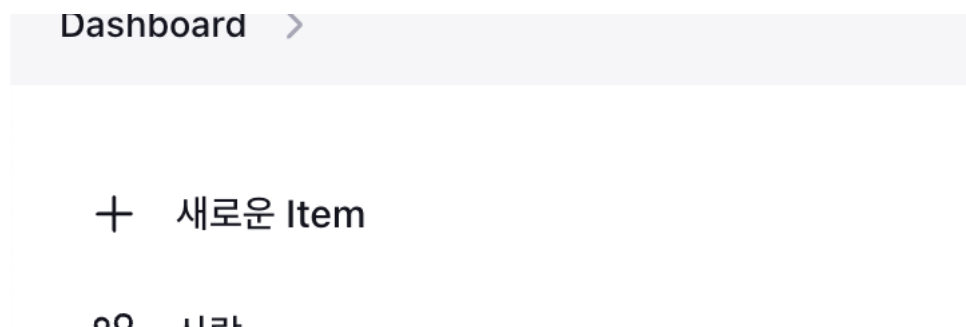
→ 여기서 젠킨슨 설치하는 것만 가져와서 설치

curl <http://169.254.169.254/latest/meta-data/public-ipv4>

→ 이 명령어를 통해 ip주소 “get” 한다

젠킨슨 접속해서 추천 플러그인 설치 → 정보 등록

## 인스턴스 구성





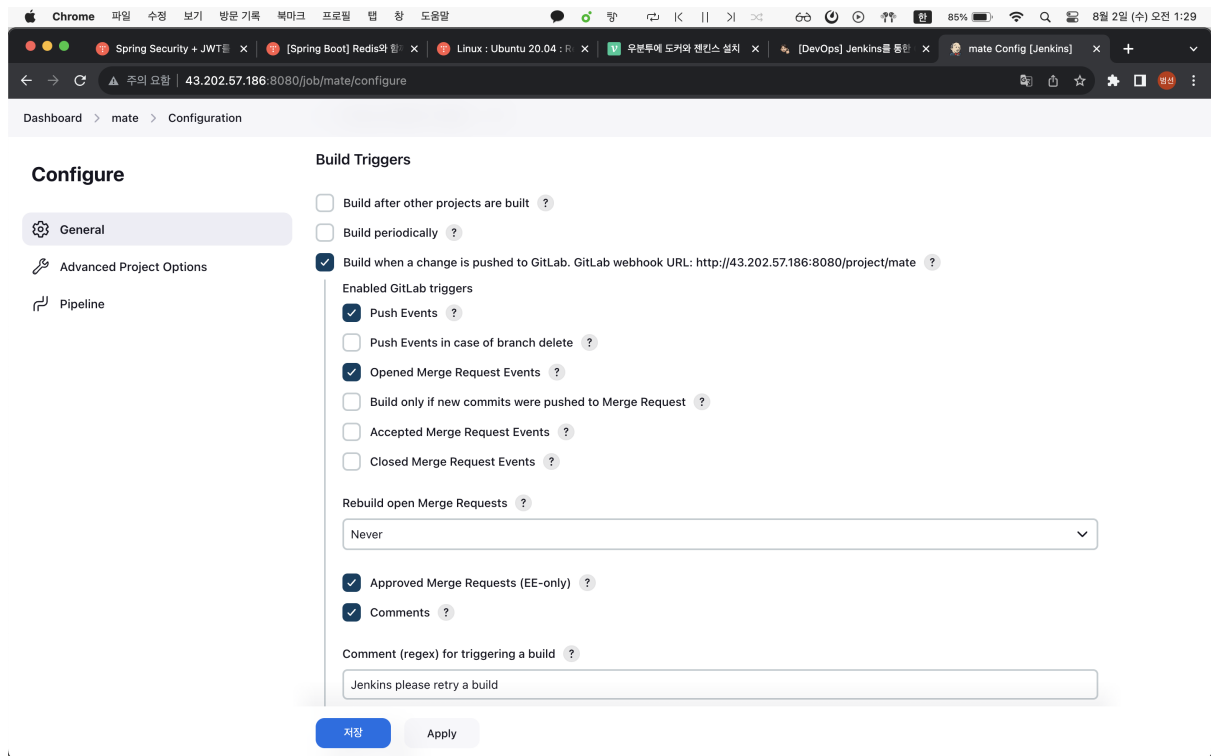
## Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

→ 파이프라인 선택

```
pipeline {
  agent any
  tools {
    gradle 'gradle'
  }
  stages {
    stage('Git Clone') {
      steps {
        git branch: 'develop-be', credentialsId: 'access', url: 'https://lab.ssafy.com/s09-webmobile1-sub2/S09P12B206.git'
      }
    }
    stage('BE-Build') {
      steps {
        dir("./Backend/mate") {
          sh "chmod +x ./gradlew"
          sh "./gradlew clean build"
        }
      }
    }
    stage('Deploy') {
      steps {
        sshagent(credentials: ['ssh_key']) {
          sh '''
            ssh -o StrictHostKeyChecking=no ubuntu@43.202.57.186
            scp /var/jenkins_home/workspace/mate/Backend/mate/build/libs/mate-0.0.1-SNAPSHOT.jar ubuntu@43.202.57.186:/home/ubuntu/
            ssh -t ubuntu@43.202.57.186 ./deploy_be.sh
          '''
        }
        timeout(time: 30, unit: 'SECONDS') {
          // ...
        }
      }
    }
  }
}
```

파이프라인 코드



## 트리거 설정

☒ Build when a change is pushed to GitLab. GitLab webhook URL: <http://43.202.57.186:8080/project/mate> ?

→ 깃랩에 등록할 때 필요

Secret token ?

5db4a8c440d4a233e6ae1d5503ddedd6

Generate

→ 깃랩에 등록할 때 필요

## credentials

깃에 접근할 수 있는 Access Token 을 가지고 있는 credentials 1개





SSH 인증 정보 1개

→ 총 2개만 있으면 돼

## Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

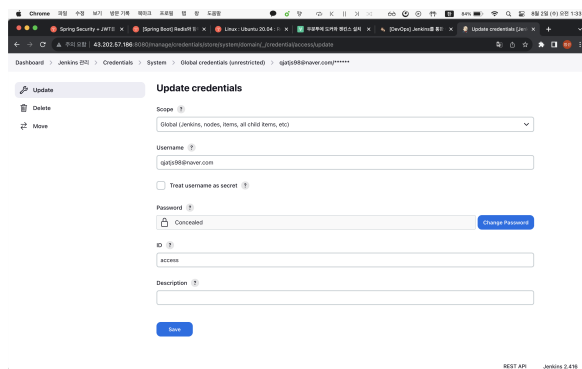
ID	Name	Kind	Description
 access	qjats98@naver.com/*****	Username with password	
 ssh_key	ubuntu	SSH Username with private key	

아이콘: S M L

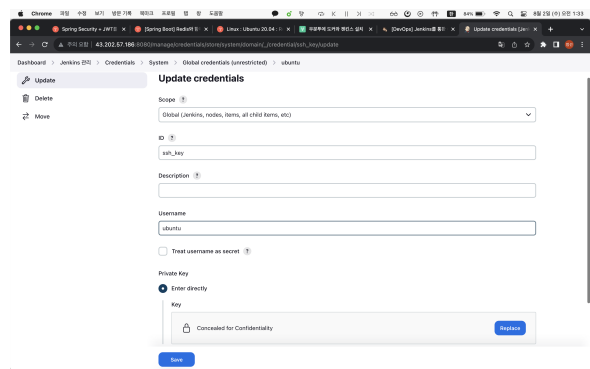
Kind 를 잘 선택해 줘야 함

왼쪽에는 깃 Access Token 을 넣어주면 돼

오른쪽에는 pem KEY를 통해 나오는 RSA KEY를 넣어주면 돼



The screenshot shows the 'Update credentials' form for the 'access' credential. The 'Scope' is set to 'Global ( Jenkins, nodes, items, all child items, etc )'. The 'Username' is 'qjats98@naver.com'. The 'Password' field is masked with 'Concealed'. The 'ID' is 'access' and the 'Description' is empty. There are 'Update', 'Delete', and 'Move' buttons on the left, and a 'Save' button at the bottom.



The screenshot shows the 'Update credentials' form for the 'ssh\_key' credential. The 'Scope' is set to 'Global ( Jenkins, nodes, items, all child items, etc )'. The 'ID' is 'ssh\_key' and the 'Description' is empty. The 'Username' is 'ubuntu'. The 'Private Key' section is active, showing 'Enter directly' and 'Key' fields. The 'Key' field is masked with 'Concealed for Confidentiality'. There are 'Update', 'Delete', and 'Move' buttons on the left, and a 'Save' button at the bottom.

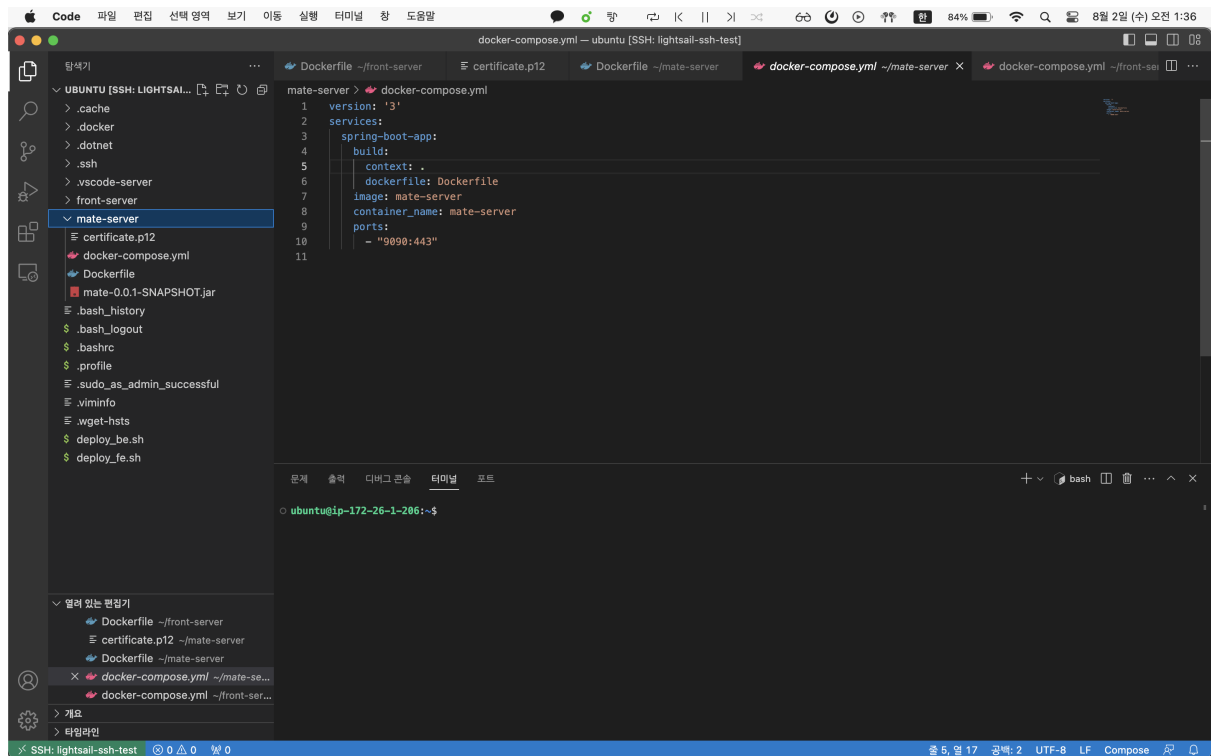
credential 설정은 아래와 같이 kind는 SSH Username with private key를 선택 후 아래와 같이 인증 정보를 입력해주면 된다.

key에는 pem 키 정보를 입력해주면 된다. 아래의 명령어를 통해 나오는 키 값을 `-----BEGIN RSA PRIVATE KEY-----` 부터 `-----END RSA PRIVATE KEY-----` 까지 모두 복사해 입력하면 된다.

```
cat {key 이름}.pem

// 결과
-----BEGIN RSA PRIVATE KEY-----
{키 값들}
-----END RSA PRIVATE KEY-----
```

BEGIN 부터 END 까지 잘 등록 해줘야 함



## 우분투 서버 구조

```

stage('Deploy') {
  steps {
    sshagent(credentials: ['ssh_key']) {
      sh '''
        ssh -o StrictHostKeyChecking=no ubuntu@43.202.57.186
        scp /var/jenkins_home/workspace/mate/Backend/mate/build/libs/mate-0.0.1-SNAPSHOT.jar ubuntu@43.202.57.186:/home/ubuntu
        ssh -t ubuntu@43.202.57.186 ./deploy_be.sh
      '''
    }
    timeout(time: 30, unit: 'SECONDS') {
    }
  }
}

```

여기 부분

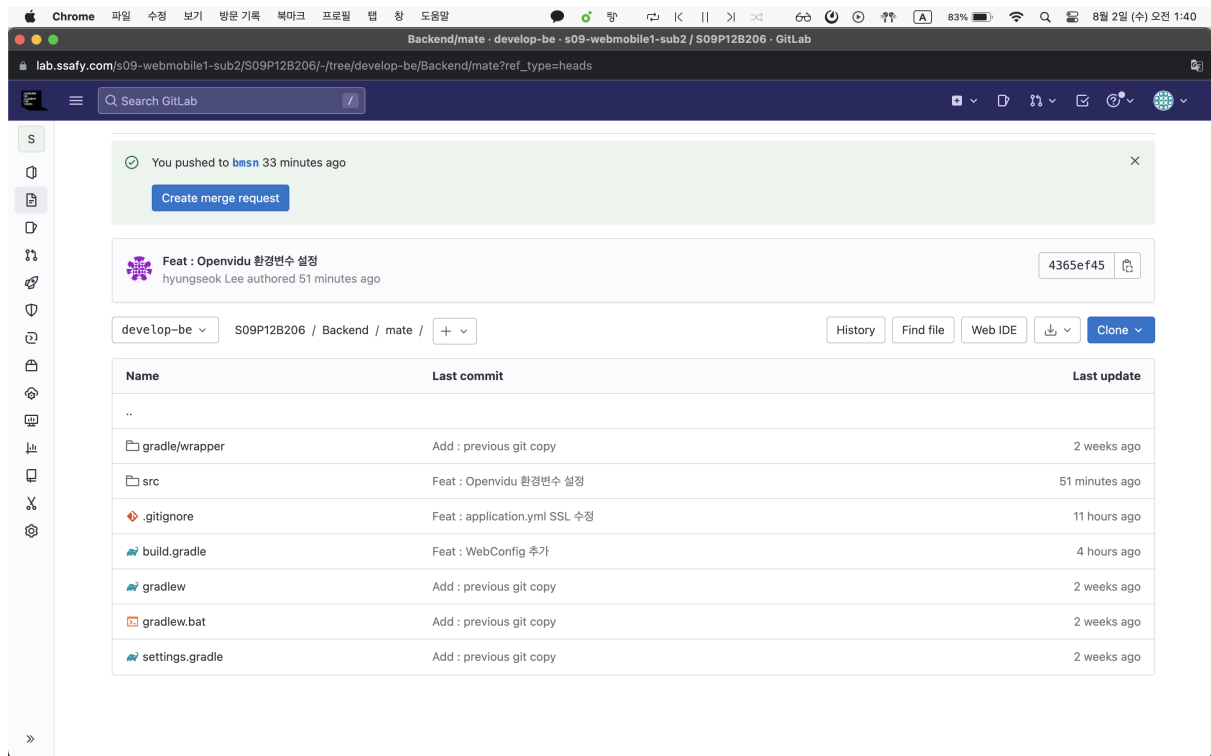
scp 는 내가 알기로 secure copy 이런 느낌임

/var/jenkins\_home/workspace/mate/Backend/mate/build/libs/mate-0.0.1-SNAPSHOT.jar

이 경로는 젠킨스 상에서 jar파일을 빌드하고 생성해 주는 경로임(정확하지 않음 ㅈㅈ;;)

/var/jenkins\_home/workspace/ → 까지는 공통

mate/Backend/mate/ → 첫번째 mate 는 인스턴스 이름인거 같음



Backend/mate/ → build.gradle 있는 것 경로를 설정

build/libs/ → 이것도 젠킨스에서 만들어주는 경로로 알고 있음(정확 X)

mate-0.0.1-SNAPSHOT.jar

mate는 자바 프로젝트이름 + "-" + 0.0.1-SNAPSHOT는 스프링부트에 있는 버전 가져다 쓰면 돼 + ".jar"

이렇게 하면 경로 설정은 끝 ㅋㅋ

## 복사

위에서 설정한 경로에 있는 jar 파일을 → jenkins안에 있는 파일

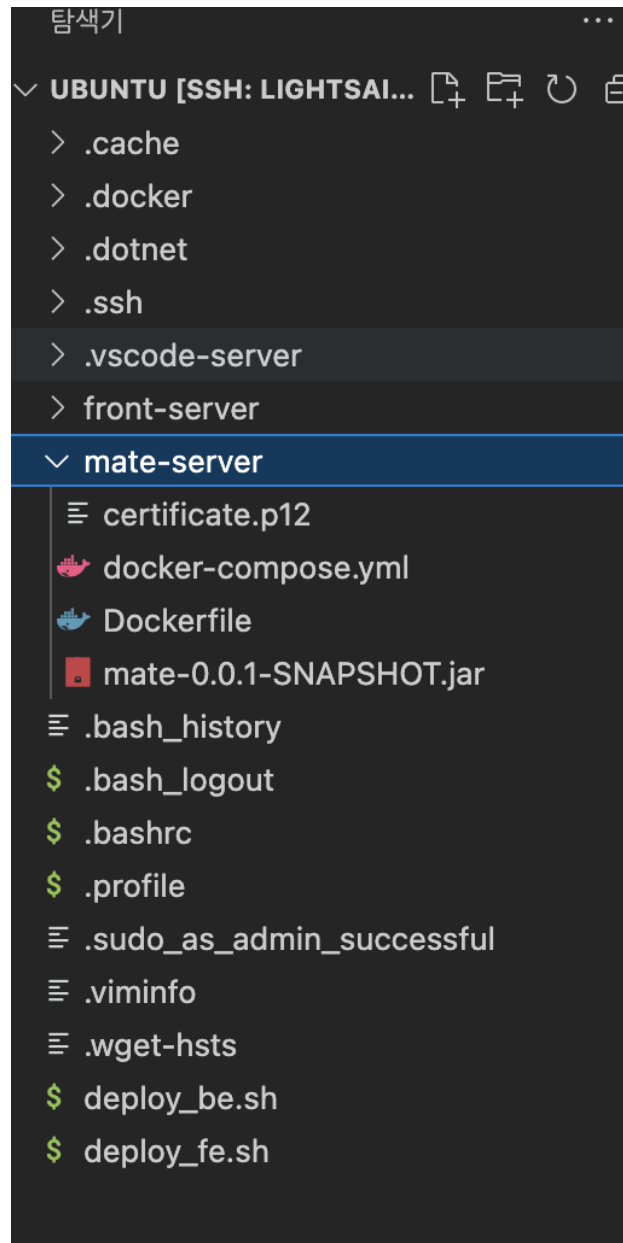
ubuntu@43.202.57.186:/home/ubuntu/mate-server

우분투 경로에 복사하는 거임

ubuntu@43.202.57.186:/home/ubuntu/

→ 여기까지가 우분투에 접속했을 때의 위치

나는 mate-server라는 경로를 만들어 줬어



취향인거 같은데 docker-compose.yml , Dockerfile, jar 파일을 한 위치에

심지어 ssl 저것도 같은 곳에

그게 나중에 경로설정하는데 편한거 같음

```
ssh -t ubuntu@43.202.57.186 ./deploy_be.sh
```

jar 파일을 우분투 경로에 복사해 주고,

스크립트에서 그 다음 실행되는 명령어임

우분투 안에 있는 deploy\_be.sh 이 녀석을 실행시킨다 이런 느낌이라고 이해 하고 있음

```
deploy_be.sh  
  
cd mate-server  
sudo docker-compose down
```

```
sudo docker-compose up -d --build
yes | sudo docker system prune
```

아까 생성한 mate-server 로 이동

docker-compose 를 내리고

sudo docker-compose up -d --build → 백그라운드 환경으로 도커 컴포즈 실행 시킨다 이런느낌

yes | sudo docker system prune → 실행하고 서버가 꺼지면서 원래 있던 애들이 리소스가 없어지나 할튼 그 전에것들을 지원하는 명령어임

```
docker-compose.yml

version: '3'
services:
  spring-boot-app:
    build:
      context: .
      dockerfile: Dockerfile
    image: mate-server
    container_name: mate-server
    ports:
      - "9090:443"
```

느낌 오자나?

```
Dockerfile

FROM adoptopenjdk/openjdk11
ENV PUBLIC_IP=43.202.57.186:3500/mate
ENV DB_USER_NAME=B206
ENV DB_USER_PASSWORD=PWB206
ENV SSL_PASSWORD=ssafy
ENV jwt_secretKey=VGhpcyBpcyBhIGRlbW8gcHJvamVjdCBmb3IgaW1wbGVtZW50aw5nIGp3dC4=VGhpcyBpcyBhIGRlbW8gcHJvamVjdCBmb3IgaW1wbGVtZW50aw5nIGp3
ENV jwt_access_expiration=3600000
ENV jwt_access_header=Authorization
ENV jwt_refresh_expiration=1209600000
ENV jwt_refresh_header=Authorization-refresh
ENV SSL_KEY_PATH=
ENV OPENVIDU_URL=https://i9b206.p.ssafy.io:4443
ENV OPENVIDU_PASSWORD=PWB206206
COPY ./certificate.p12 /certificate.p12
COPY ./mate-0.0.1-SNAPSHOT.jar /mate-0.0.1-SNAPSHOT.jar
CMD ["java", "-jar", "mate-0.0.1-SNAPSHOT.jar"]
```

환경변수 설정 good

믿는다