

서버 CI/CD

CI/CD 구축

기본 설정

▼ aws ec2

▼ 도커

1. Docker 설치

```
$ sudo apt-get update
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg
```

2. Docker 공식 GPG 키 추가

```
$ sudo mkdir -m 0755 -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

3. Docker Repository 설치

```
$ echo \
    "deb [arch=$(dpkg --print-architecture)] signed-by=/etc/apt/keyrings/docker.gpg https://download.docker.com/linux/ubuntu \
    " $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Docker 설치

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

5. Docker 실행 테스트

```
$ sudo docker run hello-world

# 실행된 도커 컨테이너 확인
$ sudo docker ps

# 이미지 확인
$ sudo docker images
```

▼ 젠킨스

1. 젠킨스 이미지 다운로드

```
$ docker pull jenkins/jenkins:ls
```

2. 젠킨스 컨테이너 띄우기

```
$ sudo docker run -d -p 8080:8080 -v /jenkins:/var/jenkins_home --name jenkins -u root jenkins/jenkins:ls
```

- **d**: 컨테이너를 데몬으로 띄웁니다.

- **p 8080:8080** : 컨테이너 외부와 내부 포트를 **포워딩**합니다. 좌측이 호스트 포트, 우측이 컨테이너 포트입니다.
- **v /jenkins:/var/jenkins_home** : 도커 컨테이너의 데이터는 **컨테이너가 종료되면 휘발**됩니다. 도커 컨테이너의 데이터를 보존하기 위한 여러 방법이 존재하는데, 그 중 한 방법이 **볼륨 마운트**입니다. 이 옵션을 사용하여 젠킨스 컨테이너의 `/var/jenkins_home` 이라는 디렉토리를 호스트의 `/jenkins` 와 마운트하고 데이터를 보존할 수 있습니다.
- **-name jenkins** : 도커 컨테이너의 이름을 설정합니다.
- **u root** : 컨테이너가 실행될 리눅스의 사용자 계정을 root 로 명시합니다.

3. docker-compose 사용하기

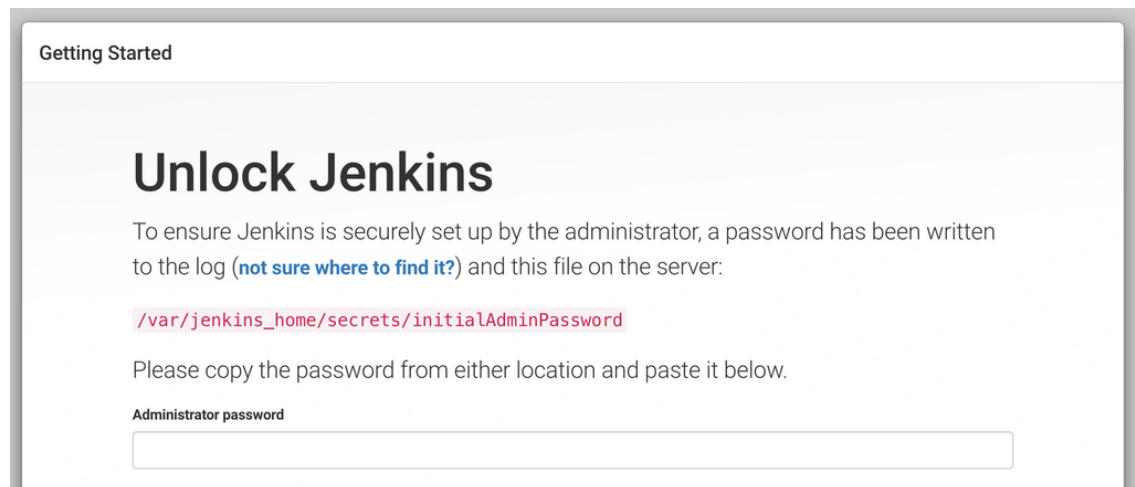
```
$ sudo apt install docker-compose
```

docker-compose.yml

```
version: "3"
services:
  jenkins:
    image: jenkins/jenkins:lts
    user: root
    volumes:
      - ./jenkins:/var/jenkins_home
    ports:
      - 8080:8080
```

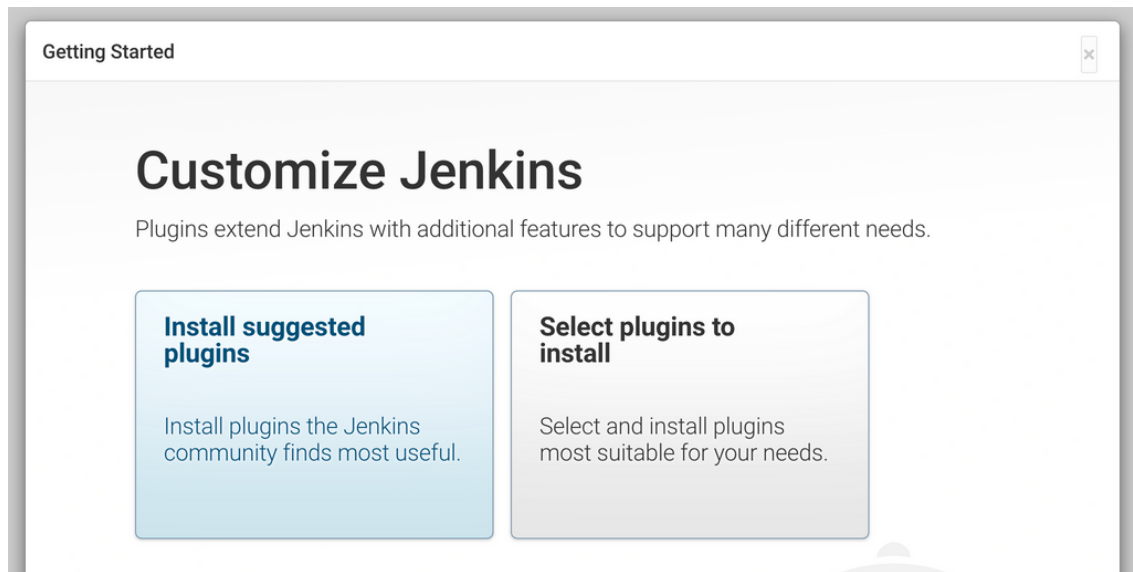
4. 젠킨스 컨테이너 시작

```
$ sudo docker-compose up -d
```



```
$ sudo docker logs jenkins
```

위 명령을 사용하면, `jenkins` 컨테이너에 출력된 로그를 확인할 수 있습니다. 젠킨스를 최초로 설치하고 실행하면 사진에서 요구하는 initial admin password를 출력해줍니다. 이를 입력하시면 됩니다.



그후 install suggested plugins를 선택해주시면 기본 설정은 끝이 납니다.

MongoDB

▼ 상세 내용

1. MongoDB 공개 GPG 키를 가져오기

```
wget -q0 - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

2. MongoDB를 위한 list 파일을 생성 - 버전 별로 다른 Ubuntu 20.04 기준

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

```
sudo apt update
```

3. MongoDB 설치

```
sudo apt install -y mongodb-org
```

4. MongoDB 실행

```
sudo systemctl start mongod
```

5. user 생성하기

```
$ mongosh
> use admin
switched to db admin
> db.createUser({user: 'ssafy', pwd: 'ssafy1111', roles: ['root']})
Successfully added user: { "user" : "ssafy", "roles" : [ "root" ] }
> exit
bye
```

6. MongoDB GUI 툴 사용시 (MongoDBCompass 기준)

New Connection

Connect to a MongoDB deployment



FAVORITE

URI ⓘ

Edit Connection String ☐

mongodb://ssafy:*****@3.36.125.122:27017/?authMechanism=DEFAULT&authSource=admin

▼ Advanced Connection Options

General

Authentication

TLS/SSL

Proxy/SSH

In-Use Encryption

Advanced

Connection String Scheme

mongodb

mongodb+srv

Standard Connection String Format. The standard format of the MongoDB connection URI is used to connect to a MongoDB deployment: standalone, replica set, or a sharded cluster.

Host

3.36.125.122:27017



TLS/SSL is disabled. If possible, enable TLS/SSL to avoid security vulnerabilities.

Save

Save & Connect

Connect

URI 란에 위와 같이 입력

-----참고-----
mongodb://{몽고아이디}/{몽고비밀번호}@{ec2 ipv4 주소}:{몽고db포트번호}/?authMechanism=DEFAULT&authSource={사용할 db이름}

몽고db 터지면 conf 파일 다시 설정하고 아래 입력

```
chown -R mongodb:mongodb /var/lib/mongodb  
chown mongodb:mongodb /tmp/mongodb-27017.sock
```

Jenkins 설정 정보

▼ 상세 내용

▼ BE

▼ Jenkins 설정

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/93ce2c87-306b-42a0-a96c-7a15819ce210/b_e_manual.pdf

▼ Dockerfile

```
FROM openjdk:11-jdkARG
JAR_FILE = build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

▼ FE

▼ Jenkins 설정

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1e02b304-ec91-4967-8382-3a560fe0a02d/f_e_manual.pdf

▼ Dockerfile

```
FROM nginx:latest
COPY . /usr/share/nginx/html
```

▼ Nginx

▼ 설치

```
// docker 로 nginx 설치
sudo apt update
sudo apt install nginx
// 인증서 다운
sudo snap install certbot --classic
sudo certbot --nginx// /etc/nginx/ 경로에 파일등 생성됨.
// sites-available 에 default 파일 있음. ( 해당 파일로 바꿔도 됨)
// conf.d 파일에 설정 파일 생성 후 코드 작성
vim nginx.conf
```

```
server {

    location / {
        proxy_pass <http://localhost:8081>;
    }

    location /be {
        proxy_pass <http://localhost:8082/be>;
    }

    location /api {
        proxy_pass <http://localhost:8083/api>;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot

    client_max_body_size 0;

    ssl_certificate /etc/letsencrypt/live/k8d202.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/k8d202.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
}

server {
    if ($host = k8d202.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    server_name k8d202.p.ssafy.io;
```

```
    return 404; # managed by Certbot
}
```

```
// nginx reload
nginx -s reload
```

- [참고한 링크](#)

▼ nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: P00DLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml+
    rss text/javascript;

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;

    proxy_request_buffering off;
}
```

```
#mail {

# # See sample authentication script at:

# # [http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript](http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript)

#

# # auth_http localhost/auth.php;

# # pop3_capabilities "TOP" "USER";

# # imap_capabilities "IMAP4rev1" "UIDPLUS";

#

# server {

# listen localhost:110;

# protocol pop3;

# proxy on;

# }

#

# server {

# listen localhost:143;

# protocol imap;

# proxy on;

# }

#}
```

▼ conf.d

```
server {
    location / {
        proxy_pass <http://localhost:3000>;
        proxy_request_buffering off;
        proxy_buffering off;
    }

    location /be {
        proxy_pass <http://localhost:8082/be>;
        proxy_request_buffering off;
        proxy_buffering off;
    }

    location /Admin {
        try_files $uri $uri/ /index.html;
        root /usr/share/nginx/html;
        index index.html;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot

    ssl_certificate /etc/letsencrypt/live/k8s101.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/k8s101.p.ssafy.io/privkey.pem; # managed by Certbot
}

server {
    if ($host = [k8s101.p.ssafy.io](http://k8s101.p.ssafy.io/)) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    ...

    server_name k8s101.p.ssafy.io;
    return 404; # managed by Certbot

    ...
}
```

```
}
```

▼ 참고자료

- [링크](#)

▼ 원격 서버

▼ Dockerfile

```
# python 이미지 다운
FROM python:3.8.12
# 컨테이너 속, 명령어 사용할 경로 지정 ( 작업 디렉토리 )
WORKDIR /app
# 현재 폴더에 있는 내용 복사, 작업 디렉토리에 붙여넣기
COPY . .
# requirements.txt 파일로, 추후에 라이브러리 추가 하기

RUN apt-get update && apt-get install -y ssh

RUN pip install --no-cache-dir -r requirements.txt
RUN echo "root:root" | chpasswd
# 작업 디렉토리에서, 명령어 사용하기, 시작 프로그램 실행
# CMD [ "python", "./setup.py" ]
CMD [ "/usr/sbin/sshd", "-D" ]
# 시작 프로그램 마친후에, 메인 프로그램 실행
ENTRYPOINT [ "python3.8", "CsvFileMaker.py" ]
```

▼ jenkins 설정

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/33a0d7e6-13ec-4e8b-8c34-e1fc59d2497a/%EC%9B%90%EA%B2%A9%EC%84%9C%EB%B2%84_manual.pdf

▼ 감지 서버

▼ Dockerfile

파이썬 코드 내 ip주소가 있어서 변경해야 함

```
FROM python:3.8.12

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD [ "python3.8", "detect.py" ]
```

▼ jenkins 설정

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7f6c179f-3237-4d21-aaa0-f14874e8318b/detect_manual.pdf

원격 서버 접속 후 설정

▼ 상세 내용

```
docker run -p 8282:22 --name test -d test
```


컨테이너 시작 후 openssh-server 설치 → ssh 연결을 하기 위한 설정

```
sudo apt-get update
sudo apt-get install openssh-server
vim /etc/ssh/sshd_config
```

ListenAddress 열기

PermitRootLogin yes 로 바꾸기

컨테이너 재시작

```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

```
service ssh start
apt install inotify-tools
```

inotify-tools 는 파일 변경 생성 감지 툴

감지서버로 이동 후

ssh 원격서버계정@ip -p 포트

```
ssh root@3.36.125.122 -p 8282
```

(이게 되어 원격서버에 접속해서 감지 가능)

입력하면 패스워드가 나오는 창이 나오는데 거기는 원격 서버 계정 비밀번호 입력 - 계정 없으면 생성

passwd root

```
파이썬 백그라운드 실행
-----자동화 완료 안해도 됨-----
nohup python filename.py &
tail -f nohup.out (로그확인)
ps -ef | grep filename.py(백그라운드 작업 확인 가능)
kill {your ProcessId}(백그라운드 작업 중지)
```