



# [Jenkins] Docker 기반으로 Jenkins 설정하기 (Dood 방식)



호스트의 Docker 소켓을 공유하는 Docker out of docker 방식으로 Jenkins를 설정하기

## 1. Jenkins-dood image를 빌드하기 위해 dockefile 작성

```
# base image
FROM jenkins/jenkins:jdk17

# permission 문제로 root 계정으로 아래 커맨드 실행
USER root

RUN apt-get update \
    && apt-get -y install lsb-release \

# 도커 설치
    && curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /usr/share/
keyrings/docker-archive-keyring.gpg \
    && echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] http
s://download.docker.com/linux/debian $(lsb_release -cs) stable" | tee /etc/apt/sources.lis
t.d/docker.list > /dev/null \
    && apt-get update \
# docker cli & docker compose 플러그인 설치
    && apt-get -y install docker-ce docker-ce-cli docker-compose-plugin containerd.io

# 사용자 추가
RUN usermod -u 1000 jenkins && \
    groupmod -g 998 docker && \
    usermod -aG docker jenkins

# 유저 설정
USER jenkins
```

## 2. 다음 명령어로 이미지 빌드

```
docker build -t chaekbang-jenkins .
```

## 3. Jenkins Container를 띄우기 위한 docker-compose.yml 파일을 작성한다.

```
version: "3.8"

services:
  jenkins:
    container_name: jenkins
    image: chaekbang-jenkins
    environment:
      - TZ=Asia/Seoul
    ports:
      - 8080:8080
      - 50000:50000
    volumes:
      - jenkins-data:/var/jenkins_home
      # host의 docker socket을 공유
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - chaekbang-jenkins-net
    user: jenkins

volumes:
  jenkins-data:

networks:
  chaekbang-jenkins-net:
    external: true
    name: "chaekbang-jenkins-net"
```

## 4. 실행 명령어는 다음과 같다.

```
docker compose -p jenkins up -d
```