



# HTTPS 적용 및 nginx 리버스 프록시 적용법 (with Docker)



본 문서는 Jenkins Blue Ocean 배포 문서와 이어집니다.

## SSL 적용

- letsencrypt 설치하기

```
$ sudo apt-get update  
$ sudo apt-get install letsencrypt
```

- Nginx 중지

```
$ sudo docker stop <front container name>
```



80 포트를 사용중이면 에러가 발생할 수 있습니다

- 인증서 발급

```
$ sudo letsencrypt certonly --standalone -d [도메인 네임]  
ex ) sudo letsencrypt certonly --standalone -d i4d101.p.ssafy
```



정상적으로 발급 시 /etc/letsencrypt/live/[도메인 네임] 아래에 키 값을 얻을 수 있다.

## Docker Network 생성

- Reverse Proxy를 위한 Docker network 생성

```
$ sudo docker create network <network-name>
```



<https://www.daleseo.com/docker-networks/> 에서 docker network를 자세히 알아보세요!

## SSL 옮기기

```
$ sudo cp /etc/letsencrypt/live/<도메인 네임>/fullchain.pem <볼륨 1>  
$ sudo cp /etc/letsencrypt/live/<도메인 네임>/privkey.pem <볼륨 1>
```

- Nginx 설정 파일 수정
- Jenkinsfile 수정

```
pipeline {  
    agent none  
    options { skipDefaultCheckout(true) }  
    stages {  
        stage('Build and Test') {  
            agent {  
                docker {  
                    image 'maven:3-alpine'  
                    args '-v /root/.m2:/root/.m2'  
                }  
            }  
            options { skipDefaultCheckout(false) }  
            steps {  
                sh 'mvn -B -DskipTests -f <your-pom.xml-direct>  
            }  
        }  
        stage('Docker build') {  
            agent any
```

```

        steps {
            sh 'docker build -t <front-image-name>:latest'
            sh 'docker build -t <back-image-name>:latest'
        }
    }
    stage('Docker run') {
        agent any
        steps {
            sh 'docker ps -f name=<front-image-name> -q \
                | xargs --no-run-if-empty docker containe
            sh 'docker ps -f name=<back-image-name> -q \
                | xargs --no-run-if-empty docker containe
            sh 'docker container ls -a -f name=<front-ima
                | xargs -r docker container rm'
            sh 'docker container ls -a -f name=<back-imag
                | xargs -r docker container rm'
            sh 'docker images -f dangling=true && \
                docker rmi $(docker images -f "da
            sh 'docker run -d --name <front-image-name> \
                -p 80:80 \
                -p 443:443 \
                -v <ssl 키값을 옮긴 폴더 위치>:<컨테이
                --network <만들어둔 docker network
                <front-image-name>:latest'
            sh 'docker run -d --name <back-image-name> \
                --network <만들어둔 docker network
                <back-image-name>:latest'
        }
    }
}

```