

# 컨테이너 보안 및 운영

| [19반] 정민석\_7000

## 어플리케이션 소스코드

간단한 express 앱을 만들었습니다.

- package.json

```
{  
  "name": "express-app",  
  "version": "1.0.0",  
  "main": "index.js",  
  "dependencies": {  
    "express": "^4.18.2"  
  },  
  "scripts": {  
    "start": "node index.js"  
  }  
}
```

- index.js

```
const express = require("express");  
const app = express();  
const port = 3000;  
  
app.get("/", (req, res) => {  
  console.log("Request received");  
  res.send("Hello World!");  
});
```

```
app.listen(port, () => {  
  console.log(`App listening at http://localhost:${port}`);  
});
```

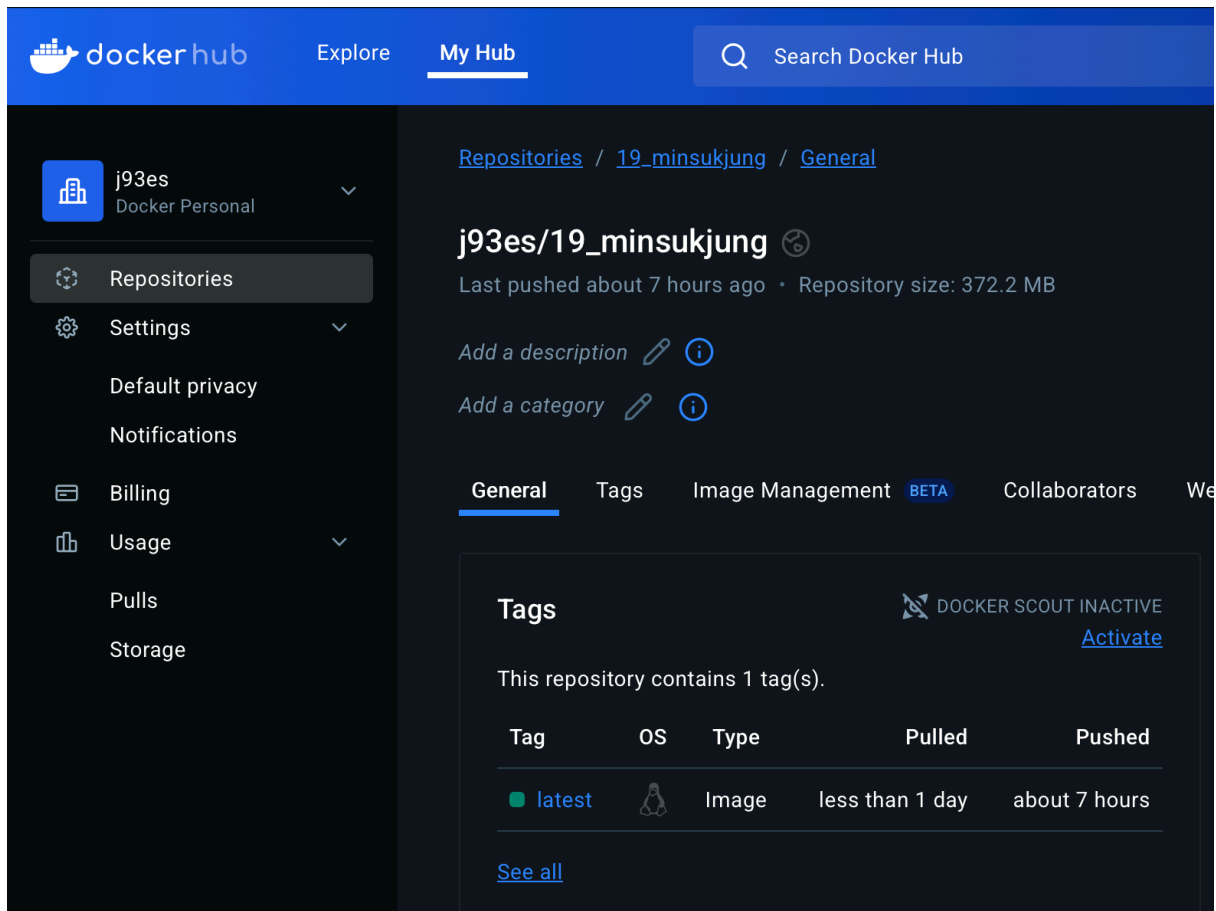
- Dockerfile

```
FROM node:20  
  
WORKDIR /app  
  
COPY package.json ./  
RUN npm install  
  
COPY index.js ./  
  
EXPOSE 3000  
  
CMD ["npm", "start"]
```

## Public Repository에 배포

아래의 명령어를 활용하여 도커 허브에 이미지를 public으로 push하였습니다.

```
docker login  
docker build -t j93es/19_minsukjung .  
docker push j93es/19_minsukjung
```



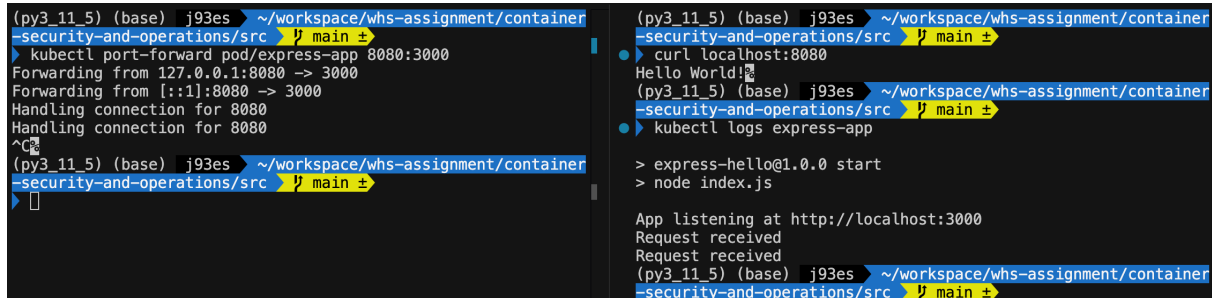
배포에 사용한 yamI은 다음과 같습니다.

- pod.yamI

```
apiVersion: v1
kind: Pod
metadata:
  name: express-app
spec:
  containers:
    - name: express-app
      image: j93es/19_minsukjung
      ports:
        - containerPort: 3000
```

이후 로컬에서 “배포/테스트/로그를 확인”하고자 다음의 명령어를 활용하였습니다.

```
kubectl apply -f pod.yaml
kubectl get pod
kubectl port-forward pod/express-app 8080:3000
kubectl logs express-app
```



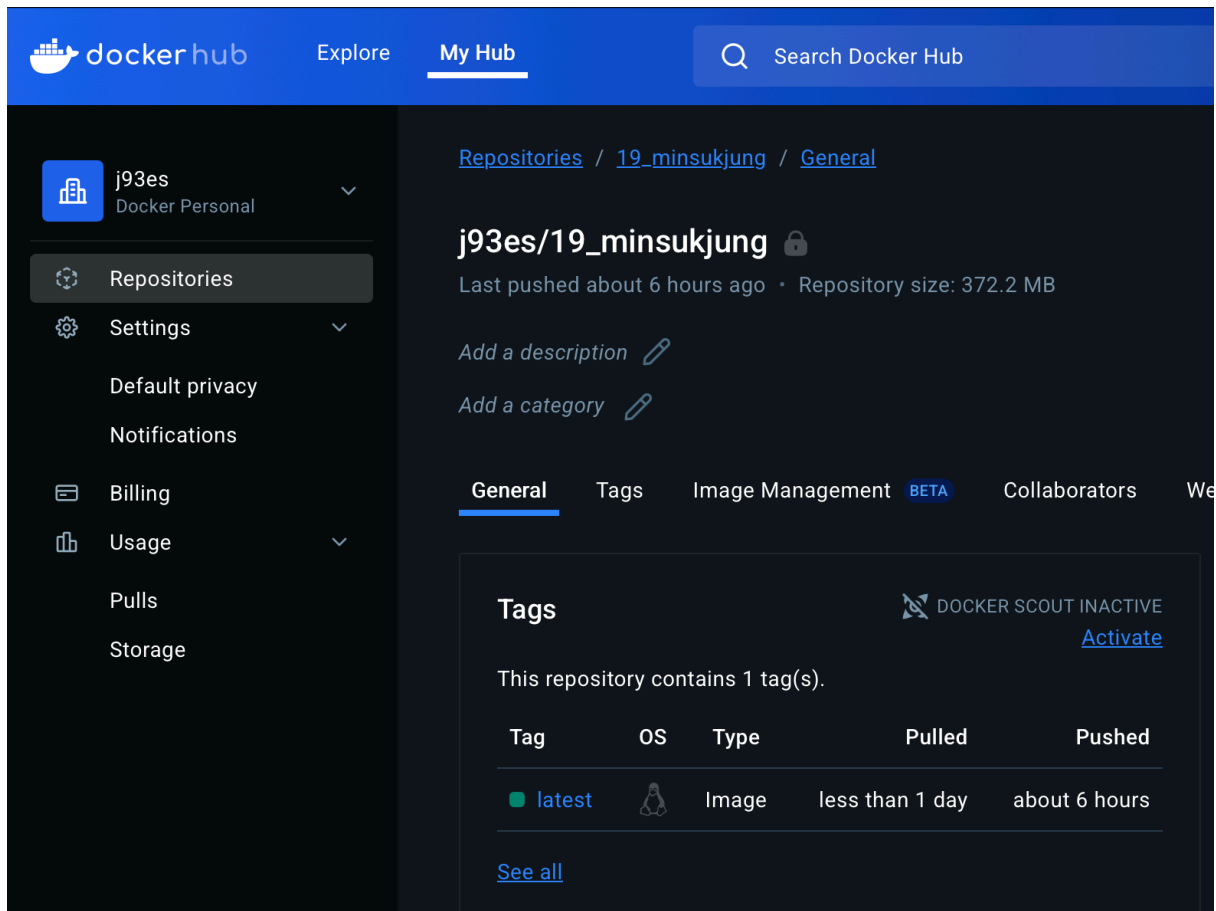
The image shows two terminal windows side-by-side. The left window shows the execution of `kubectl port-forward pod/express-app 8080:3000`, which successfully forwards port 8080 from the pod to the local machine. The right window shows the execution of `curl localhost:8080` returning "Hello World!" and `kubectl logs express-app` showing the application's startup logs, including "App listening at http://localhost:3000" and "Request received".

kind 내부와 외부로 포트포워딩하여, local에서 잘 실행되는 것을 확인할 수 있습니다.

## Private Repository에 배포

기존 public 레포지토리를 지우고, 빈 private 레포를 만들었습니다.(public에서 private로 바뀌도 되겠지만, private로 빈 레포를 생성하고 push가 잘 되는지도 확인해야한다고 생각 하였습니다. 회사를 다닌다고 가정했을때, public으로 올리고 private로 바꾸는것은 좋아보 이지 않습니다.) 그리고 다음의 명령어로 이미지를 push하였습니다.

```
docker login
docker build -t j93es/19_minsukjung .
docker push j93es/19_minsukjung
```



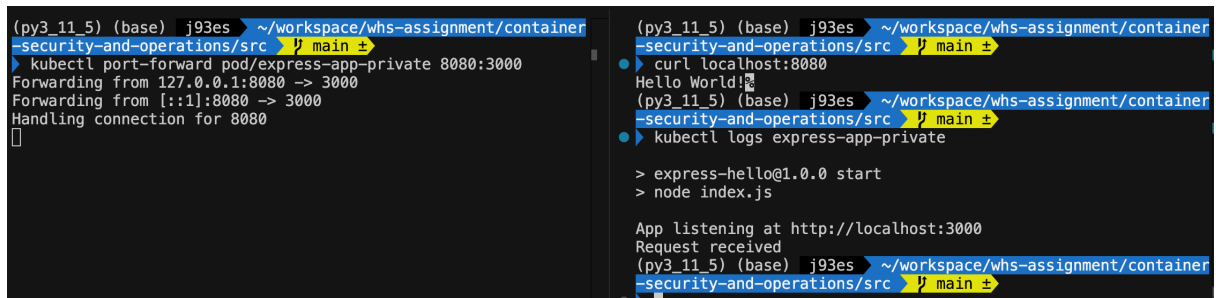
배포에 사용한 yamI은 다음과 같습니다.

- pod-private.yamI

```
apiVersion: v1
kind: Pod
metadata:
  name: express-app-private
spec:
  containers:
    - name: express-app-private
      image: j93es/19_minsukjung
      ports:
        - containerPort: 3000
  imagePullSecrets:
    - name: my-dockerhub-secret
```

이후 secret 설정을 해주고 배포하였습니다.

```
kubectl create secret docker-registry my-dockerhub-secret \
  --docker-server=https://index.docker.io/v1/ \
  --docker-username=USERNAME \
  --docker-password=PASSWORD \
  --docker-email=EMAIL
kubectl apply -f pod-private.yaml
kubectl get pod
kubectl port-forward pod/express-app-private 8080:3000
kubectl logs express-app-private
```



The image shows two terminal windows side-by-side. The left window shows the execution of `kubectl port-forward pod/express-app-private 8080:3000`, which successfully forwards port 8080 to port 3000. The right window shows the execution of `kubectl logs express-app-private`, which displays the output of the application: `express-hello@1.0.0 start`, `node index.js`, and `App listening at http://localhost:3000`. It also shows a `Request received` message and the response `Hello World!`.