A10: Micro-services Assignment Task 2

In this task we created various Dockerfiles with the purpose of building a docker images from each of our micro-services that we have created in the previous task. Those images then are pushed to the our docker-hub repository so we can use those images to deploy in the kubernetes-dashboard. Each micro-service is deployed with three nodes each.
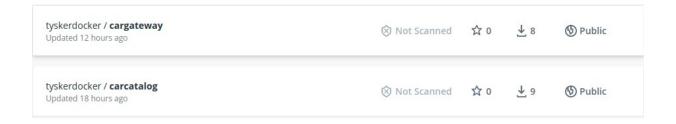
1. Creating a Dockerfile

FROM openjdk:12-alpine
COPY carcatalog.jar /carcatalog.jar
EXPOSE 8090
CMD ["java" , "-jar", "/carcatalog.jar"]

This image is based on a alpine Linux image with java 12 included. It copies a jar file and expose the API on port 8090. The command runs the jar file.

2. Building, tag and push the image to the repo

(a) `docker build –tag [name of the tag]:[version]`
(b) `docker tag [image id] [docker username]/[image name]:tag`
(c) `docker push [docker username]/[image name]`

tyskerdocker / **cargateway** Updated 12 hours ago		Not Scanned	☆ 0	↓ 8	Public
tyskerdocker / **carcatalog** Updated 18 hours ago		Not Scanned	☆ 0	↓ 9	Public

3. Deploying the images into Kubernetes

1. logging into Kubernetes dashboard
2. adding the images to kubernetes cluster

Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Created ↑	
✓ cargateway	default	k8s-app: cargateway	10.97.174.199	cargateway:8080 TCP cargateway:0 TCP	-	11 hours ago	⋮
✓ carcatalog	default	app: carcatalog	10.96.250.210	carcatalog:8090 TCP carcatalog:0 TCP	-	11 hours ago	⋮

4. Accessing the API internal

On one of the deployment is accessible on PORT 8090 and one on 8080.

```
localhost:8080
SWR3   explainshell   HOME   WORK-EDUCATION

{
  _links: {
    profile: {
      href: http://localhost:8080/profile
    }
  }
}
```

```
localhost:8090
SWR3   explainshell   HOME   WORK-EDUCATION   COURSE

{
  _links: {
    cars: {
      href: http://localhost:8090/cars{?page,size,sort},
      templated: true
    },
    profile: {
      href: http://localhost:8090/profile
    }
  }
}
```