

Title: kubernetes: image can't be pulled

Post Body:

Before onboarding to the enterprise k8s / AWS EKS, I am using docker-desktop for local k8s testing on Mac with the following Dockerfile:

```
FROM openjdk:11-jre-slim ARG JAR_FILE=target/*.jar COPY ${JAR_FILE} app.jar ENTRYPOINT [ "java", "-jar", "/app.jar" ]
```

with the following deployment steps to docker-desktop:

```
docker build . -t cpchung/rema kubectl create deployment rema --image=cpchung/rema --dry-run -o=yaml > deployment.yaml echo -
```

This generates the deployment.yaml:

```
apiVersion: apps/v1 kind: Deployment metadata:   creationTimestamp: null   labels:     app: rema   name: rema spec:   replicas
```

and test the deployment with the following health check:

```
kubectl port-forward svc/rema 8080:8080
```

```
curl localhost:8080/actuator/health
```

But I am getting this error:

```
robinhood $ kubectl get all NAME                                READY   STATUS             RESTARTS   AGE pod/rema-57df6cf5fc-px8pc
```

But I really have the image from dockers:

```
robinhood $ docker images REPOSITORY                                TAG                                IMAGE ID                                CREATED
```

What could be wrong here? I am using mac with docker-desktop.

Accepted Answer:

<https://stackoverflow.com/a/52763242/3514300> should solve the issue for you

In your deployment add imagePullPolicy: Never

By default for images tagged with no tag are given the latest tag in docker. Kubernetes by default tries to pull the images for latest tag and it wasn't able to find the image in dockerhub. <https://kubernetes.io/docs/concepts/containers/images/#updating-images>

Other way is to push your image to dockerhub too.

Highest Rated Answer: None