**Docker**

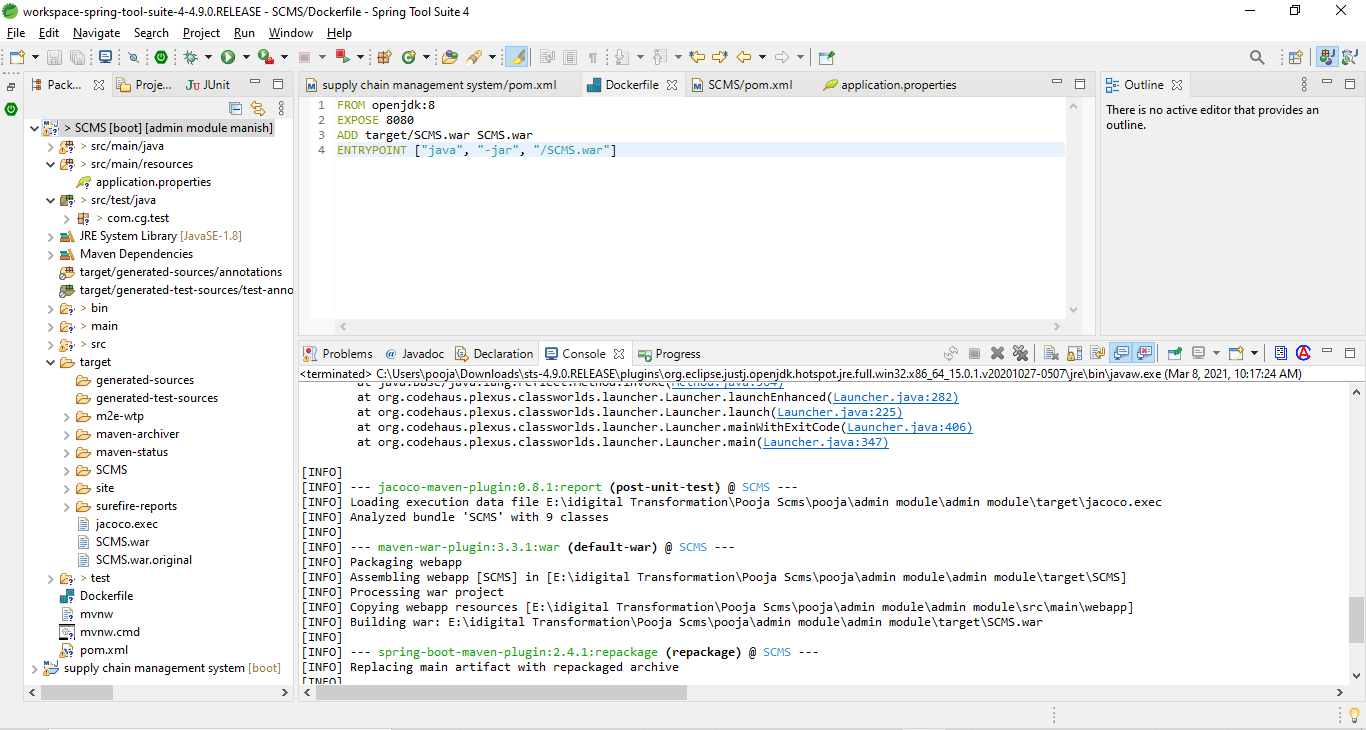
**Create Docker image for Spring boot application:**

**Steps:**

1. Install dockerhub desktop in the laptop

2. sign in to dockerhub

3. create Dockerfile in STS and write the below script



**Script:**

FROM openjdk:11

EXPOSE 8082

ADD target/ Ofos.war Ofos.war

ENTRYPOINT ["java", "-jar", "/ Ofos.war"]

4. open terminal from command prompt and check the docker version to cross verify whether docker installed successfully

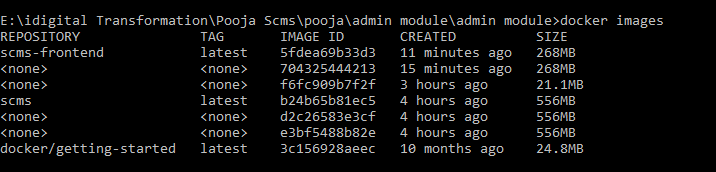
**docker -v**

5. Build the Docker file using the below command which creates the docker image as scms (image name should always be smaller case letters)

**docker build -f Dockerfile -t scms .**

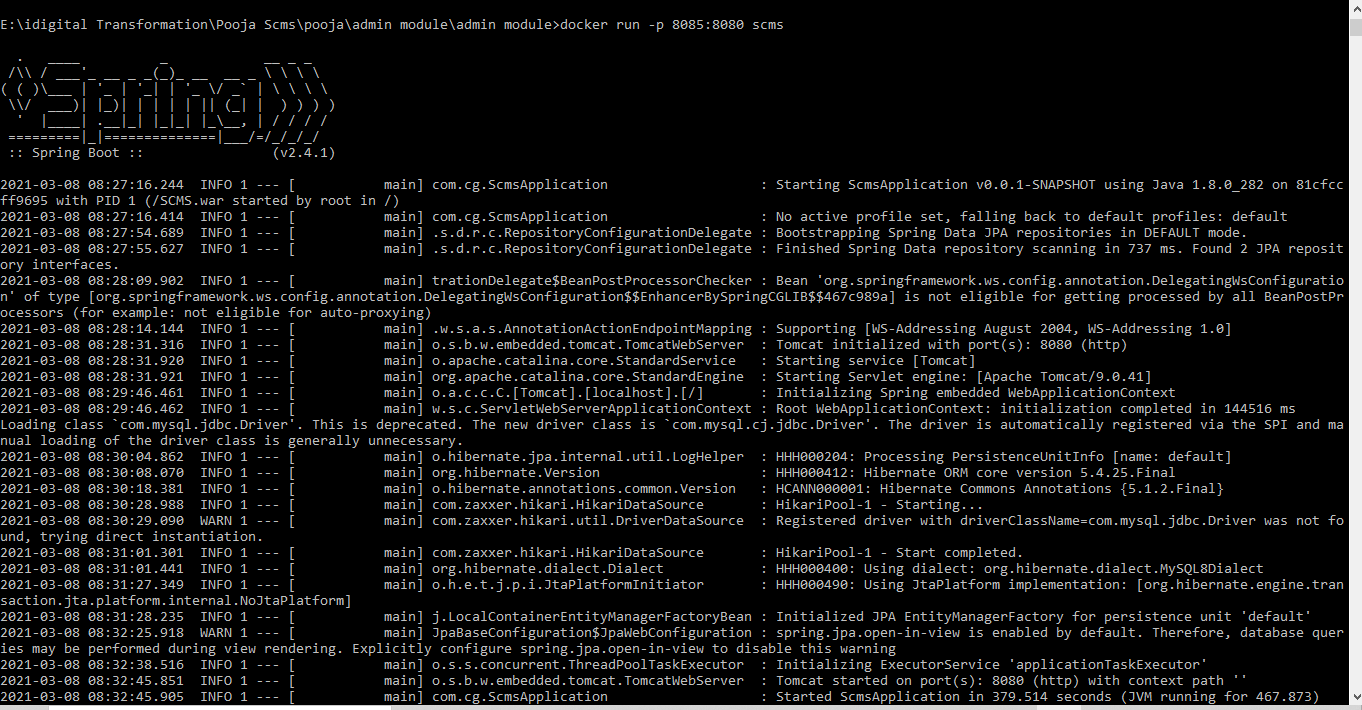
6. check whether the image created using the command

**docker images**

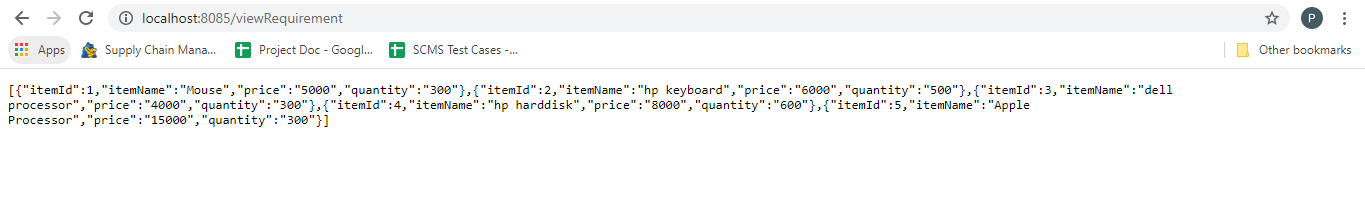


7. Run the created image

**docker run -p 8085:8080 scms**



8. Check the browser localhost://8085 port whether the app is running



**Create Docker image for Angular application:**

(<https://www.youtube.com/watch?v=MbA71IuYUhg>)

1. Create Dockerfile in Angular app root folder

2. Create script inside Dockerfile

**Script:**

FROM node:alpine

WORKDIR /usr/src/app

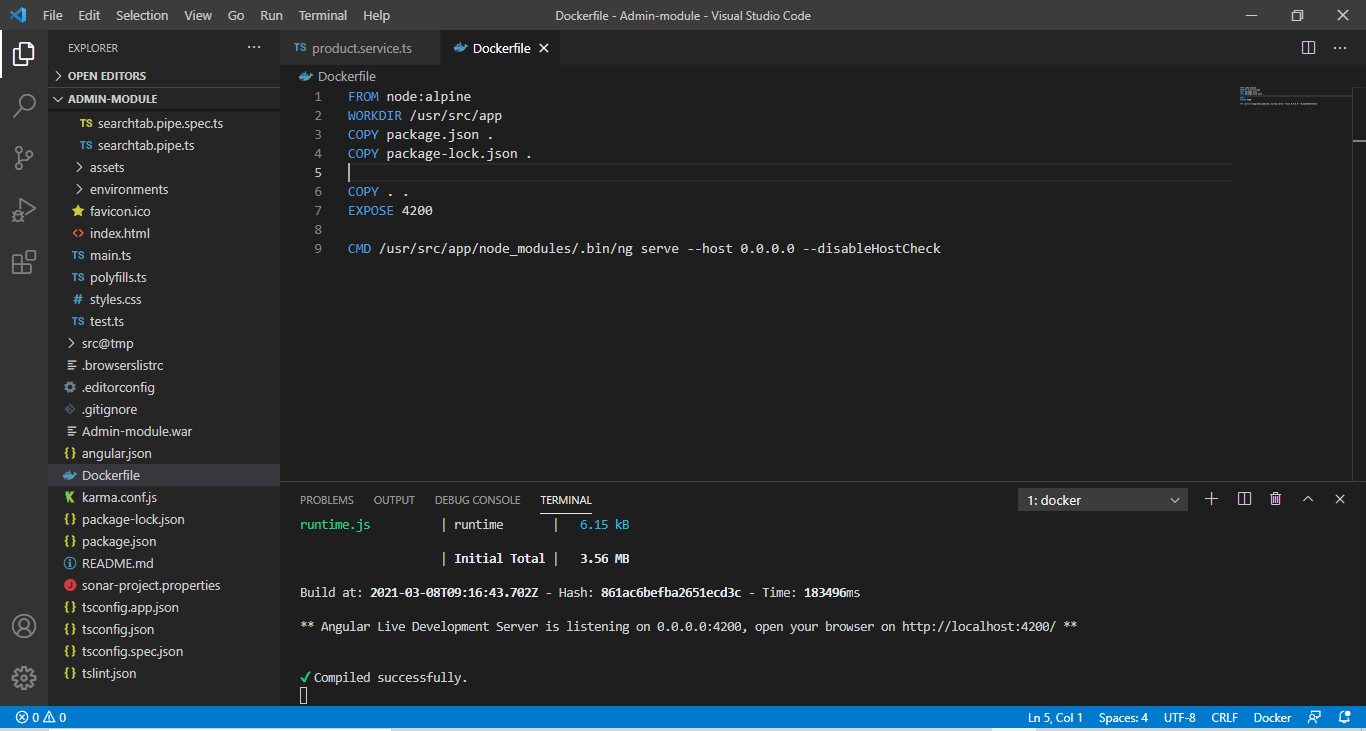
COPY package.json .

COPY package-lock.json .

COPY . .

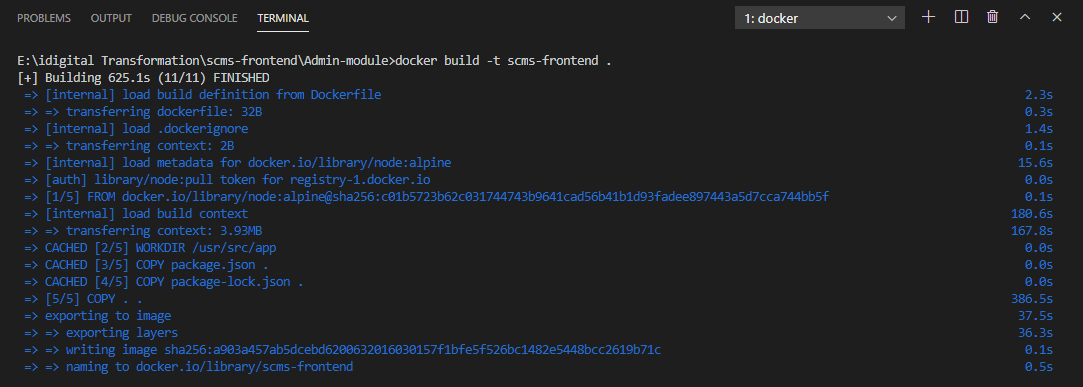
EXPOSE 4200

CMD /usr/src/app/node\_modules/.bin/ng serve --host 0.0.0.0 --disableHostCheck



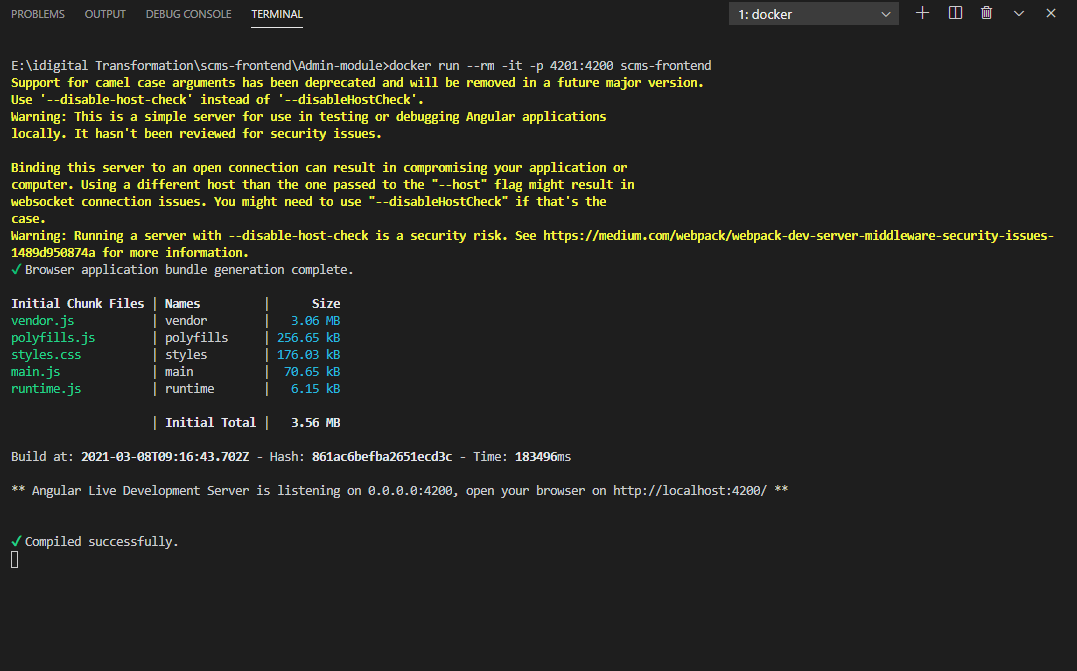
3. build the docker image using the below command

**docker build -t ofos.**



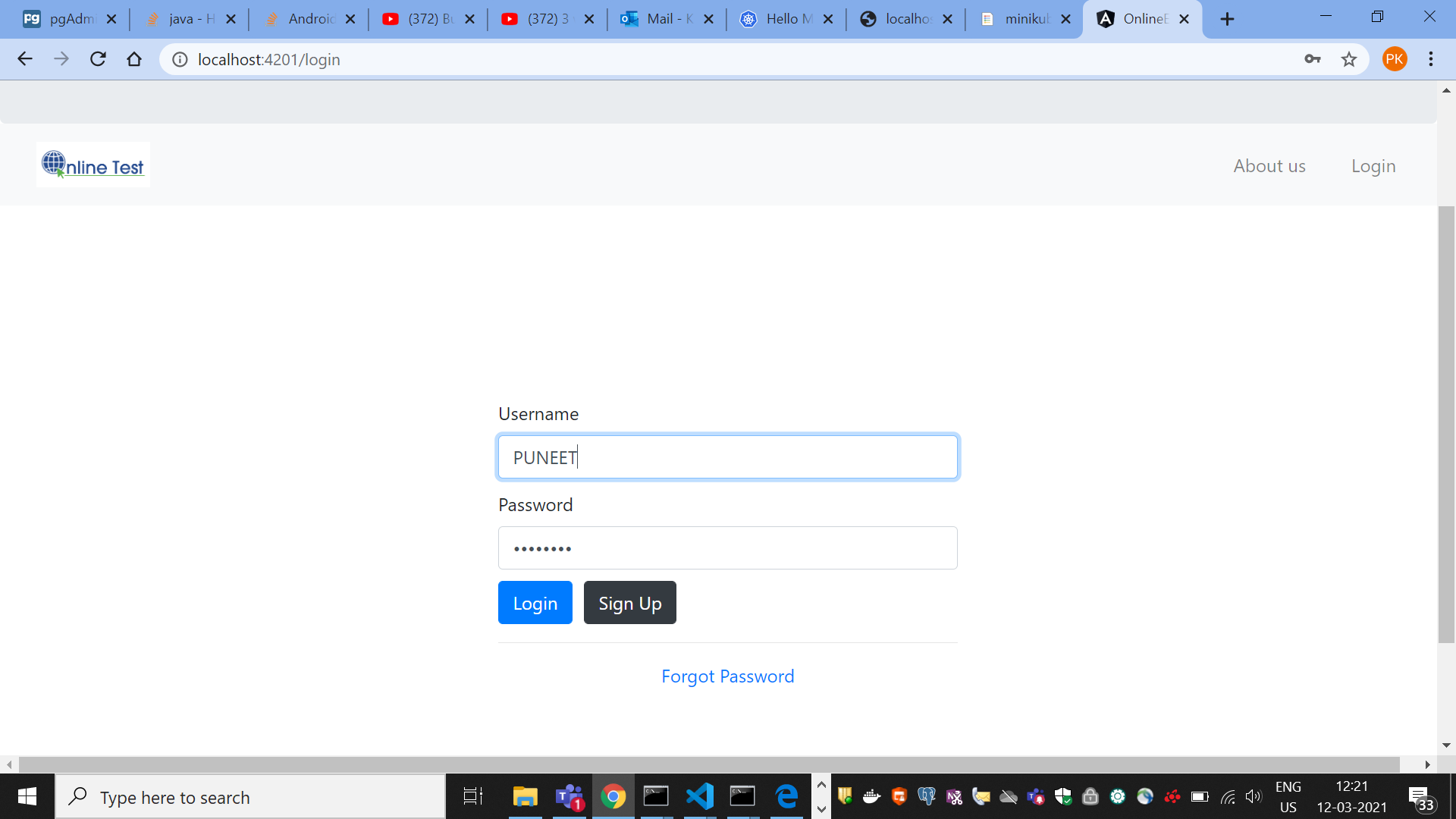
4. Run the docker image using below command

**docker run --rm -it -p 4201:4200 ofos**



5. Check the application running in the port 4201

**Push and Pull Docker image to docker hub**



1. Login to the docker hub via terminal using the below command

**docker login**

2. create the image with tag name as repo name

**docker tag ofoshackerp/ofos**

3. Push the image to the dockerhub

**docker push hackerp/ofos**

4.pull the image using the cli mentioned in docker hub

**docker pull hackerp/ofos**

5. run the application in local port

**docker run -p 8082:8082 hackerp/ofos**

**Deploy docker container on AWS**

1. Create EC2 instance and chage the security groups as follows

SSH : Anywhere

HTTP: Anywhere

HTTPS: Anywhere

2. Run the puTTy virtual machine with the following commands

**sudo yum install docker –y(**This command install docker on virtual machine**)**

**sudo service docker start(**This command used to start the docker**)**

**sudo docker run -p 80:8080 hackerp/ofos(**used to run the sts app**)**

**sudo docker run -p 80:4200 hackerp/ofos(**used to run the angular app**)**

3. After following all the above cli application will be accessible at port number 80

**Kubernetes**

**Setup minikube using below commands:**

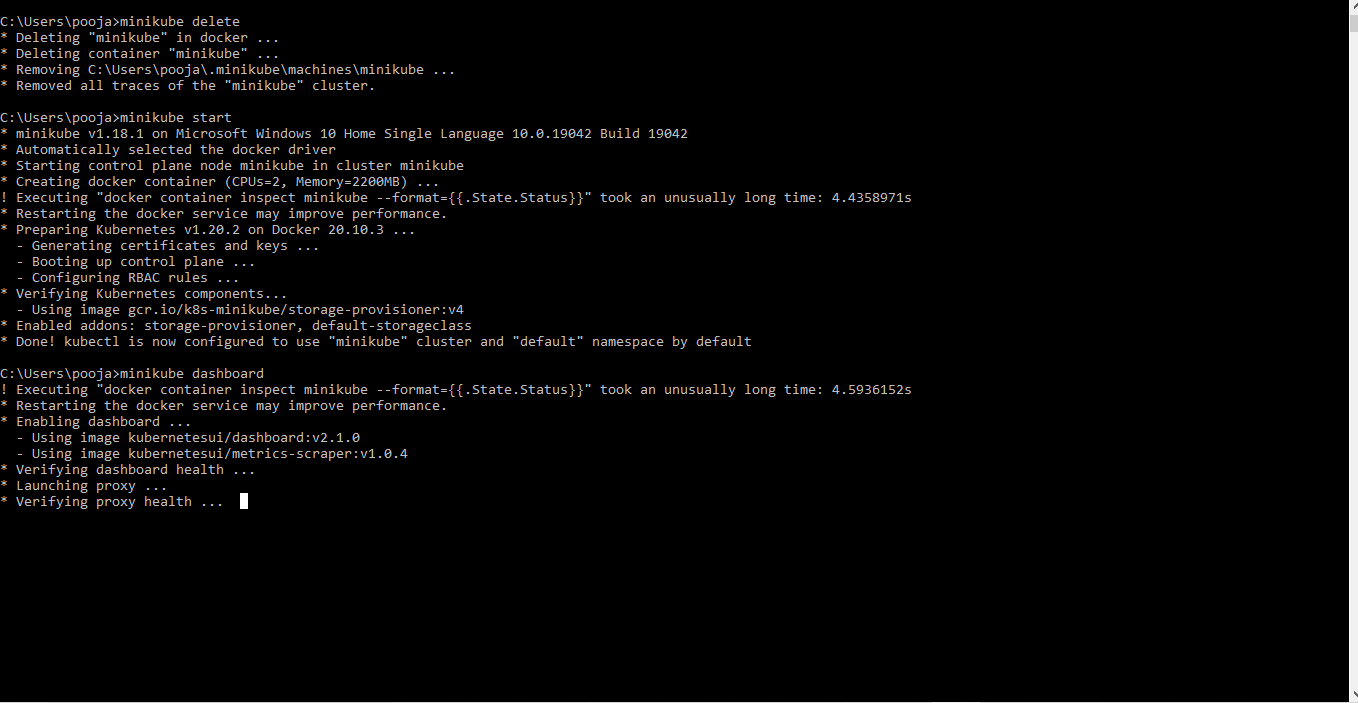
1. Install minikube from stand-alone windows installer

2. open command prompt as administrator and run the below command

“minikube start”

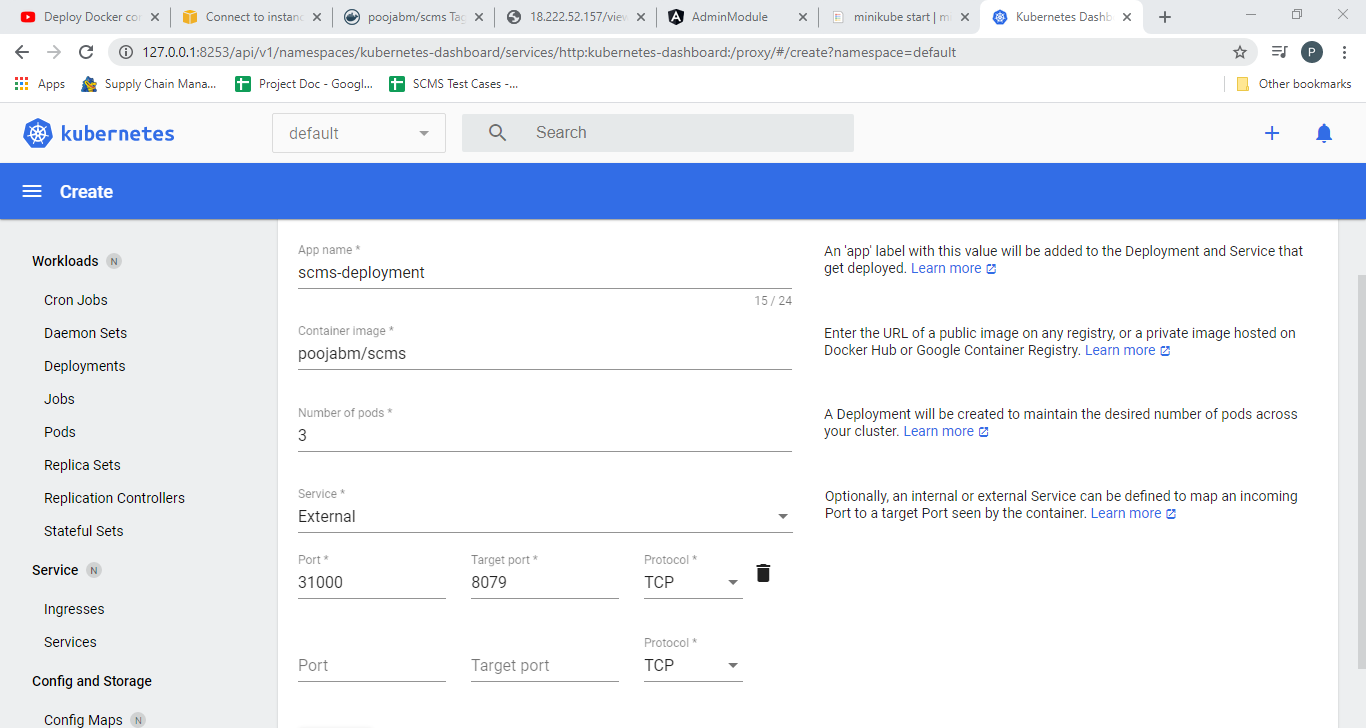
This command creates container in docker hub to deploy the application.

3.Run the “minikube dashboard” command it creates kubernetes dashboard for application deployment and services.



4. open the kubernates dashboard and deploy the application running in docker hub. (http://127.0.0.1:8253/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/deployment?namespace=default)

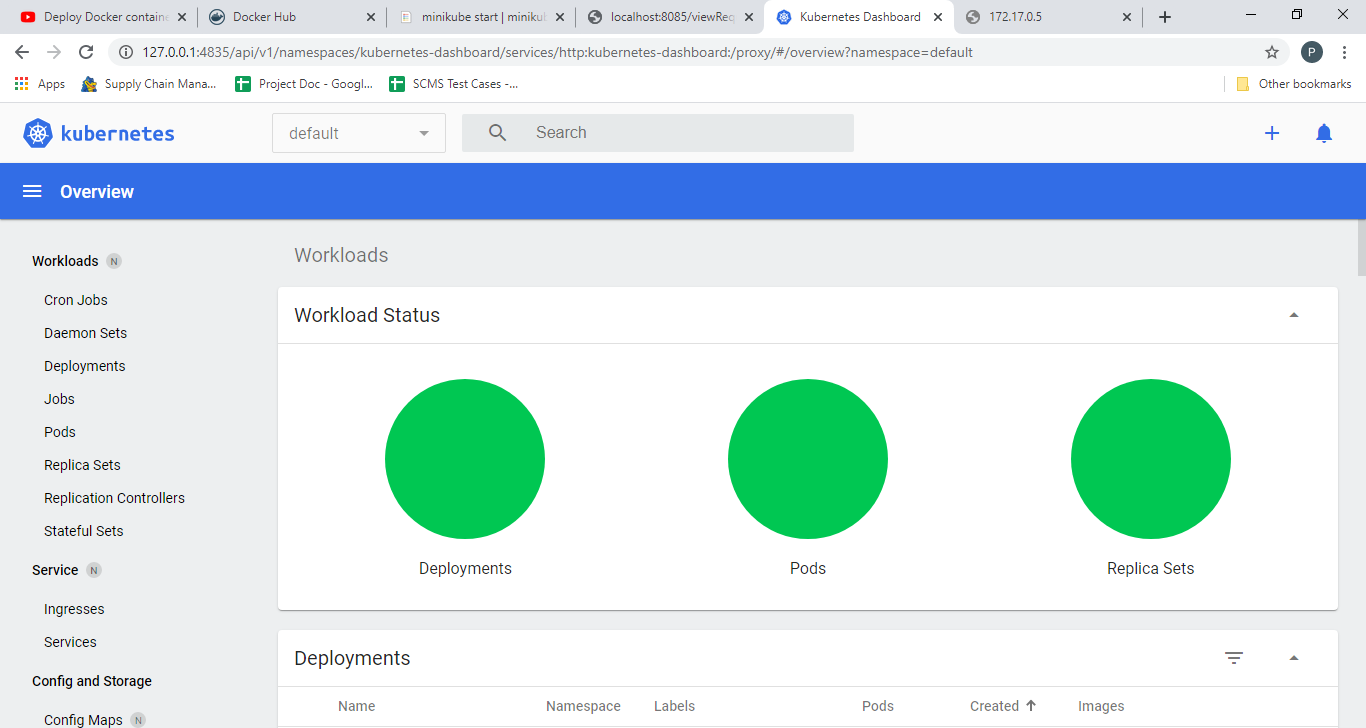
5. create an app inside kubernetes dashboard for deployment and mention the docker image to be deploy.



6.Use target port 8085(port should be same as the docker hub image running port).

7. change loadbalancer of JSON file in service tab to NodePort

8.Deploy the application successfully and you can see the below screen.



9. The docker image is running at the port 127.0.0.1:8085