Installing the docker

>> sudo amazon-linux-extras install docker

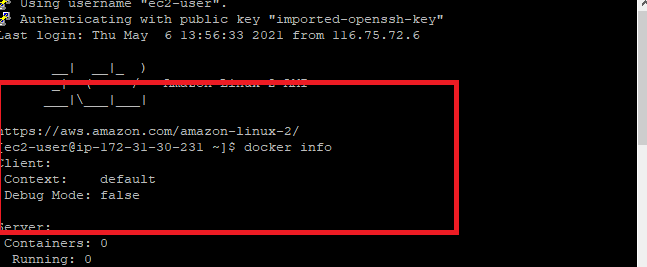
>> sudo yum install docker

>> sudo service docker start

>> sudo docker info

>> sudo usermod -a -G docker ec2-user

Above command enables you to enter the commands without sudo, to apply the settings you need re-login to the session from the putty.

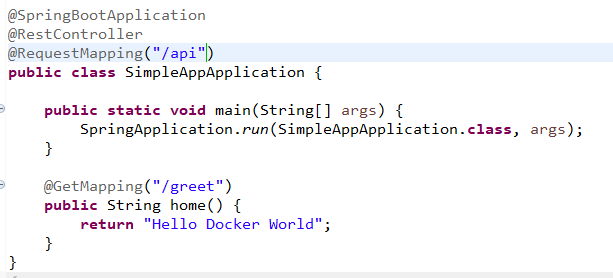


>> docker info

>> sudo docker images

Running the spring boot application in a docker

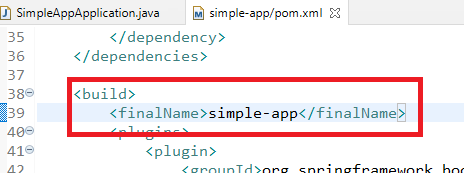
Create one simple hello world program using spring boot rest



You can first test whether the application is running on the localhost

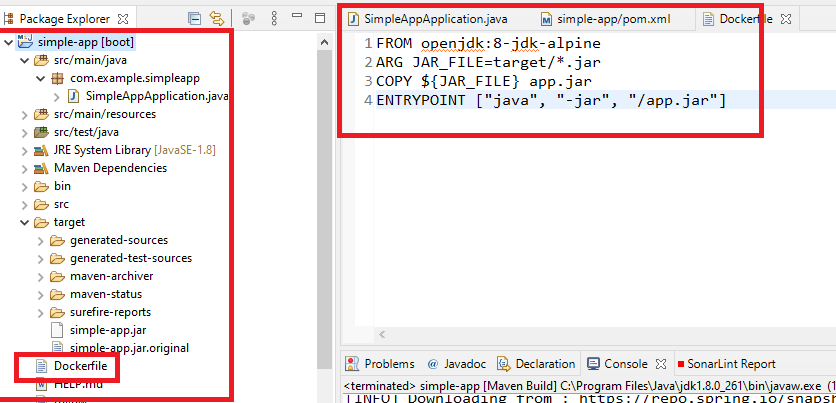
Mention the jar file you want to be created when you package

pom.xml



Create a docker file in the project folder without any extension

Dockerfile



Mention these entries in the Dockerfile

FROM openjdk:8-jdk-alpine

ARG JAR\_FILE=target/\*.jar

COPY ${JAR\_FILE} app.jar

ENTRYPOINT ["java", "-jar", "/app.jar"]

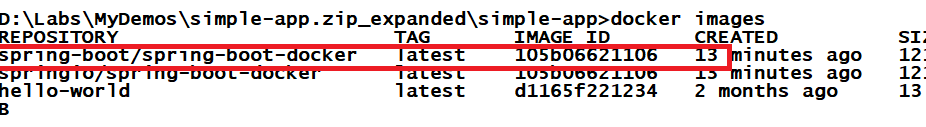
Enter the command from the project folder

docker build -t spring-boot/spring-boot-docker .

The above command creates the docker image which can be run using docker run command



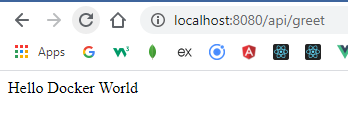
You can use docker images to find the docker images



To host the docker container

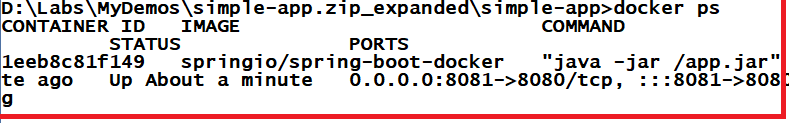
>>docker run -p 8080:8080 springio/spring-boot-docker

Output:



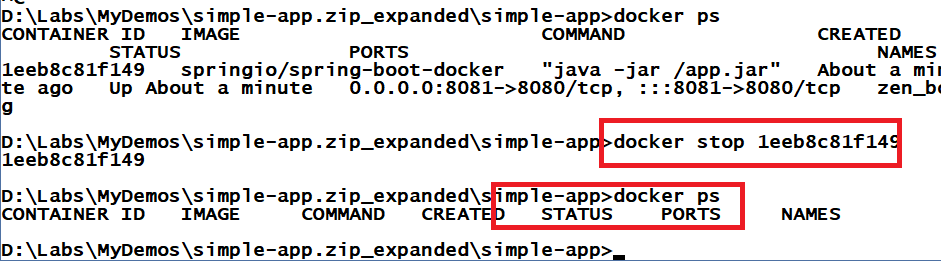
To check the running container

>> docker ps



To stop the container

>> docker stop container-id

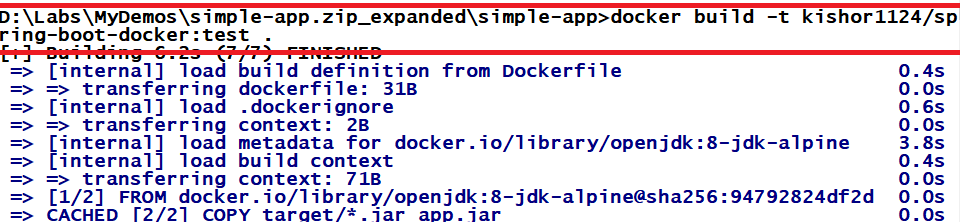


How to remove the images

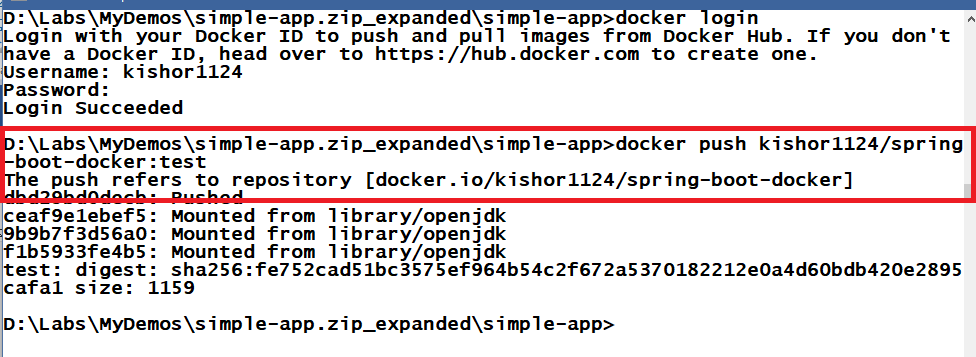
>> docker rmi imageId -f

Creating the docker image to push to the docker hub

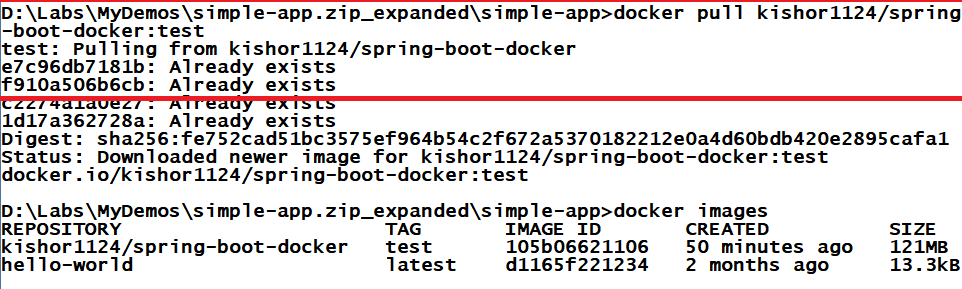
>> docker build -t kishor1124/spring-boot-docker:test .



Push to the docker hub



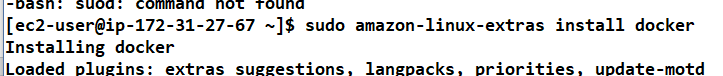
You can pull this repository from the docker hub



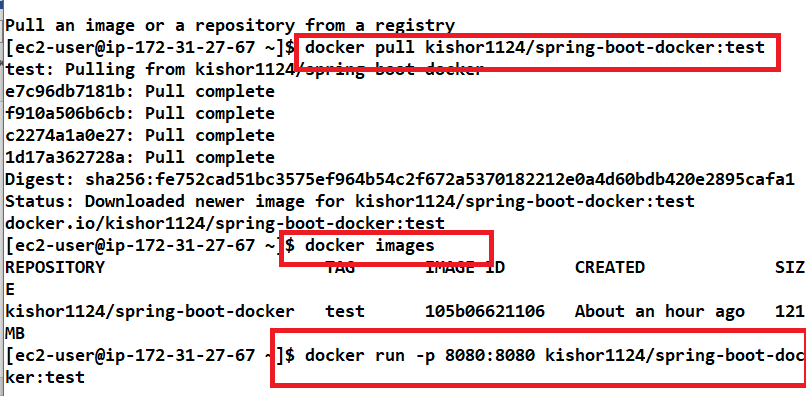
Pulling this image in the ES2 instance of AWS and running the images

Steps:

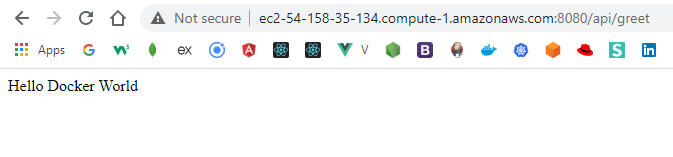
* Install the docker
* Login to the docker hub
* Pull the repository
* Run the docker image



Pulling the image



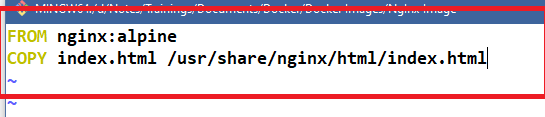
You can able to see the output in the browser



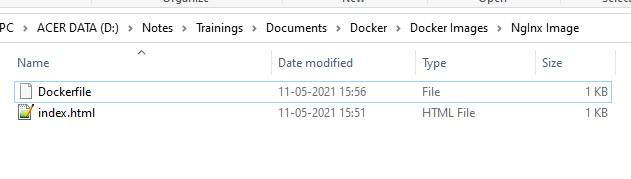
NgInx (Engine X):

It is an open-source reverse proxy server for HTTP, HTTPS, SMTP, POP3, IMAP, Loadbalancer, webserver focus on high concurrency, high performance and low memory usage.

Create a docker file using touch DockerFile



Create an index.html and write some content

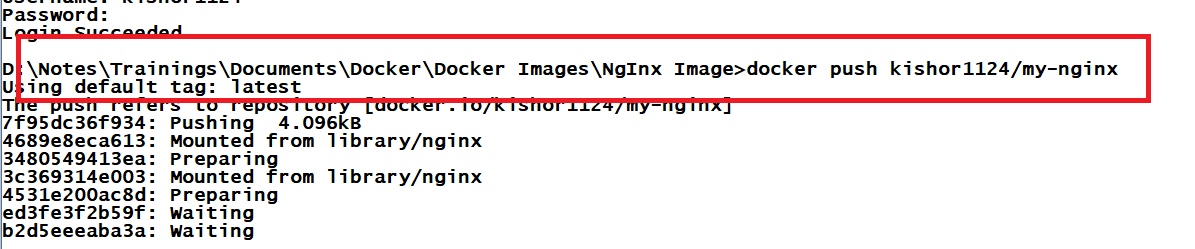


Creating the docker image

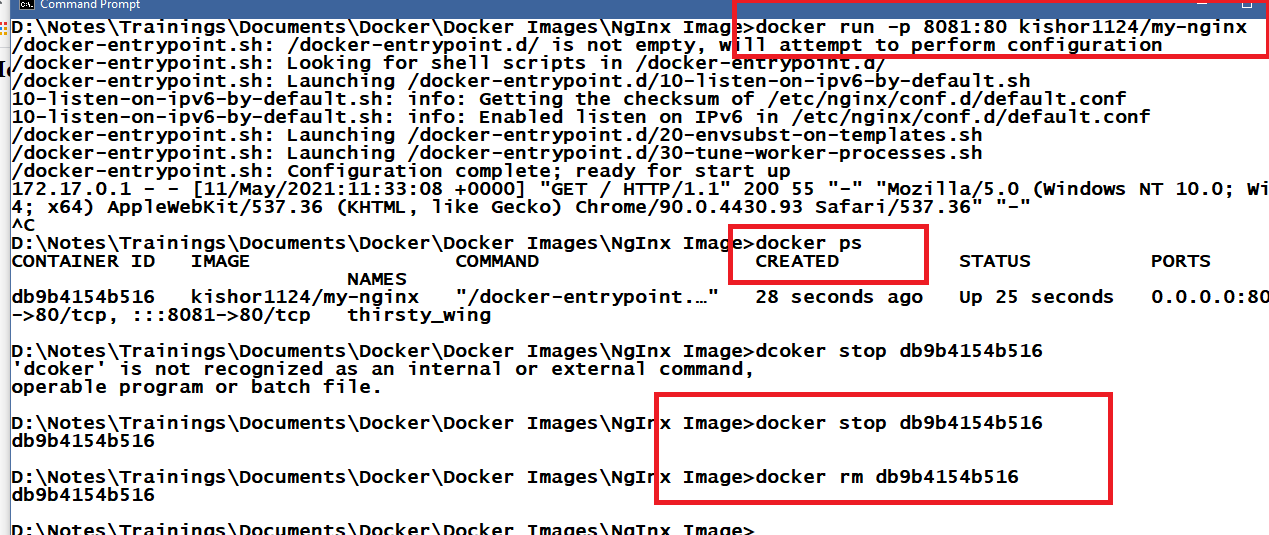
>> docker build -t kishor1124/my-nginx . docker build -t kishor1124/my-nginx .



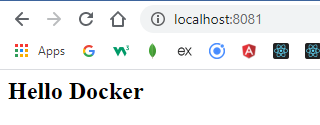
Pushing the image to the docker hub



Running the docker image



Output:



Pushing the image to the new/existing repository

