Now time for practicals guys, Let us do Dockerizing Spring boot application

---

-> Spring Boot: framework which is used to develop enterprise based applications.
-> Spring Boot applications will be packaged as a jar file for deployment.
-> To run the jar file we will use command : java -jar <file-name.jar>
-> To run springboot application jar file we will use tomcat server as "embedded server".
-> By default spring boot application will run on port number 8232

Step 1: Create Docker File

```
FROM openjdk:17

COPY target/demo-app.jar  /usr/app/

WORKDIR /usr/app/

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "demo-app.jar"]
```

in pom .xml
under build

```
<finalName>demo-app</finalName>
```

create ec2 instance t3.medium ,open security 8232

```
sudo yum update -y
sudo yum install docker -y
sudo service docker start
sudo usermod -aG docker ec2-user
exit
```

sudo yum install git -y

sudo yum install maven -y

git clone https://github.com/sasank2323/docker-test.git

cd docker-test

mvn clean package

docker build -t sasank2323/travelguideapp:testing .

docker images

copy it and paste in run

docker run -p 8080:8080 sasank2323/travelguideapp:testing

6) docker logs <container-id>

5) Access application URL in browser: http://public-ip:8080/

docker login
Enter username:sasank2323
Enter password

-> push docker image
docker push sasank2323/travelguideapp:testing

Great question! Let's break it down:

When you use `docker build`, you're creating a Docker image locally on your EC2 machine. This has nothing to do with Docker Hub *yet*. Docker Hub only comes into play when you want to push that image to your Docker Hub account.

So why do you use your Docker Hub repo name and login?

You're doing this in order to tag and push your image to Docker Hub after building it locally.

🔄 Step-by-step connection between EC2 and Docker Hub

Build image locally on EC2

```
docker build -t my-image .
```

1.

Tag image with Docker Hub repo name
Suppose your Docker Hub username is `shashank123` and repo name is `myapp`.

```
docker tag my-image shashank123/myapp
```

2.

Login to Docker Hub from EC2
This links your EC2 Docker client to your Docker Hub account:

```
docker login
```

3. **You'll enter your Docker Hub username and password.**

Push the image to Docker Hub

```
docker push shashank123/myapp
```

4.

🔒 So how is it connected to Docker Hub?

When you run `docker login`, your EC2 machine (Docker client) gets credentials (usually saved at `~/.docker/config.json`) to authenticate and push images to your Docker Hub account.

Without login, Docker Hub will reject the `docker push`.

Summary:

- **`docker build`: Local image creation.**
- **`docker tag`: Prepares image for Docker Hub with proper naming.**
- **`docker login`: Authenticates your EC2 machine with Docker Hub.**
- **`docker push`: Actually uploads image to Docker Hub under your account.**

Would you like a practical example with full commands from build to push?