

## Step For build images from DockerFile and push/pull steps:

1. Create a maven project using any framework

2. For cucumber -junit use before cucumber api to launch browser using localhost:4444/wd/hub

3. Create a Dockerfile under project structure with your requirements Ex:

```
FROM openjdk:8u191-jre-alpine3.8
```

```
RUN apk add curl jq
```

```
# Workspace
```

```
WORKDIR /usr/share/udemy
```

```
# ADD .jar under target from host
```

```
# into this image
```

```
ADD target/selenium-docker.jar selenium-docker.jar
```

```
ADD target/selenium-docker-tests.jar selenium-docker-tests.jar
```

```
ADD target/libs libs
```

```
# in case of any other dependency like .csv / .json / .xls
```

```
# please ADD that as well
```

```
# ADD suite files
```

```
# ADD book-flight-module.xml
```

```
book-flight-
```

```
module.xml
```

```
# ADD search-module.xml
```

```
search-module.xml
```

```
# ADD health check script
```

```
ADD healthcheck.sh
```

```
healthcheck.sh
```

```
# BROWSER
```

```
# HUB_HOST
```

```
# MODULE
```

```
ENTRYPOINT sh healthcheck.sh
```

4. Create a healthcheck.sh file for:

```
#!/usr/bin/env bash
```

```
# Environment Variables
```

```
# HUB_HOST
```

```
# BROWSER
```

```
# MODULE
```

```
echo "Checking if hub is ready - $HUB_HOST"
```

```
while [ "$( curl -s http://$HUB_HOST:4444/wd/hub/status | jq -r .value.ready )" != "true" ]  
do  
    sleep 1  
done
```

```
# start the java command  
java -cp selenium-docker.jar:selenium-docker-tests.jar:libs/*  
\   
-DHUB_HOST=$HUB_HOST \  
-DBROWSER=$BROWSER \  
-Dcucumber.options="$CUCUMBER_OPTIONS" \  
org.junit.runner.JUnitCore com.runner.TestRunner
```

5. To Create selenium jars under target file run the following command from the project location

```
mvn clean package -DskipTests
```

6. Check whether selenium jars are generated using below command

ls from target folder

7. Docker image commands

```
docker pull imageName  
docker images : To see list of images  
docker ps -a : To see the list of containers  
docker system prune -af : To remove unused containers  
docker rmi imageName : To remove the image
```

8. To create a docker image and push it to Docker Hub

i. Go to the Dockerfile path and run below command

```
docker build -t="dockerUsername/imageName" .
```

Note: "." Refers to the current path in the above command

ii. To push it to DockerHub run below commands

```
docker login  
enter username, password observe login succeeded  
docker push imageName which created above
```

9. Create a docker compose yams file with details like Hub, Chrome, Firefox and Image name services as below

```
version: "3"  
services:  
  hub:
```

```

    image: selenium/hub:3.14
    ports:
      - "4444:4444"
  chrome:
    image: selenium/node-chrome:3.14
    depends_on:
      - hub
    environment:
      - HUB_HOST=hub
  bdd:
    image: sai11051992/imageName
    depends_on:
      - chrome
    environment:
      - HUB_HOST=hub
      - BROWSER=chrome
      - CUCUMBER_OPTIONS=classpath:features
    volumes:
      - ./output:/usr/share/sai/target/cucumber-reports

```

10 To package as Jar run below command

```
java -cp selenium-dcoker.jars:selenium-docker-tests.jar:libs/* -Dcucumber.options=classpath:features org.junit.runner.JUnitCore runner.TestRunner
```

Or you may run with docker-compose.yaml file

11.To run docker compose yams file, run below command

```
docker-compse up : Start
docker-compose down : Stop
```

12.After any update in feature files or any files, repeat step 5,8,9 & 11