Docker Important Command

FOR LOGIN:

docker login

FOR CREATE IMAGE:

SYNTEX:-

docker build -t repostioryName/Image_name :<Version or Tag its optional> .

EXAMPLE:-

docker build -t ramnageena/docker-hello-world :v1 .

FOR RUN IMAGE:

SYNTEX:-

docker run -d -p 8081:8080 <repostioryName/Image_name>

EXAMPLE:-

docker run -d -p 8081:8080 ramnageena/docker-hello-world :v1

CHECK RUNNING IMAGE:

docker images

CHECK RUNNING CONTAINER:

docker container ls

STOP CONTAINER:

docker container stop <conatiner_id>

COMMAND REMOVES A STOPPED CONTAINER :

docker rm <conatiner_id>

COMMAND REMOVES A DOCKER IMAGE :

docker rmi <image_name>

COMMAND PULLS AN IMAGE FROM A DOCKER REGISTRY:

```
docker pull ubuntu:latest
```

COMMAND PUSHES AN IMAGE TO A DOCKER REGISTRY:

```
docker push repostioryName/Image_name
```

COMMAND FETCHES THE LOGS OF A CONTAINER:

```
docker logs <image_id>
```

COMMAND FETCHES THE RUNNING LOGS OF A CONTAINER:

```
docker logs -f <image_id>
```

Steps to Dockerize the Spring boot Application with Mysql database

Step 1:

pull the mysql image from docker hub

docker pull mysql:5.7 -- if this will not work use 2nd one

docker pull --platform linux/x86_64 mysql:5.7

Step 2:

Create a docker network to communicate Spring boot app and Mysql database

docker network create springboot-mysql-net

Step 3:

Run the mysql container in the network

docker run --name mysql --network springboot-mysql-net -e MYSQL_ROOT_PASS=1234 -e MYSQL_DATABASE=<database_name> -e MYSQL_USER=sa -e MYSQL_PASSWORD=1234 -d mysql:5.7

Step 4:

Check the create database

docker exce -it<conatiner_id> bash

mysql -u<username> -p<password>

show database

Step 5:

Update the application.properties file

spring.datasource.url=mysql://mysqldb:3306/<database_name>

spring.datasource.username=sa

spring.datasource.password=1234

Step 6:

build the spring boot docker image

docker build -t springbootmysql .

Step 7:

Start the spring boot container on the same network

docker run --network springboot-mysql-net --name springboot-container -p 8080:8080 -d springbootmysql