**Week 9: Creation of images and container ..pushing them into dockerhub**

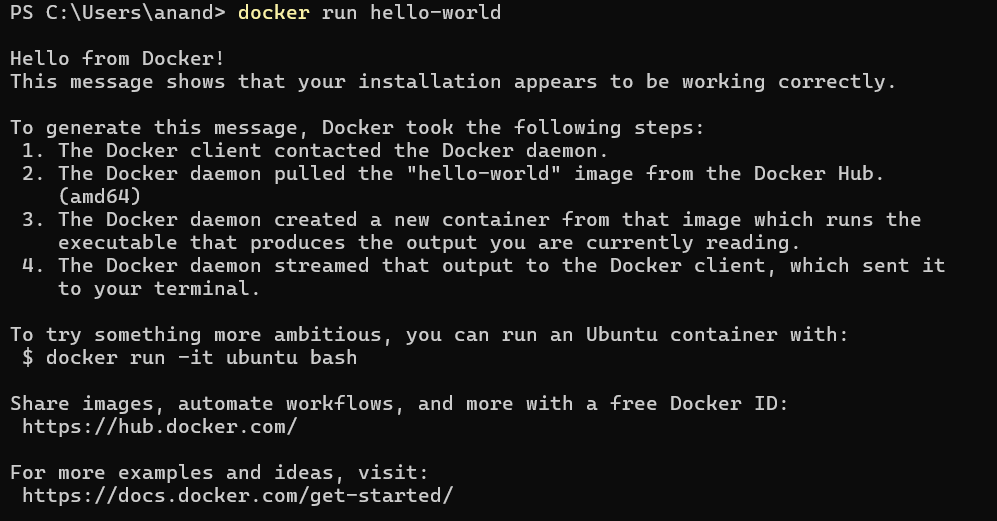
1. Evaluation of previous experiments.
2. Hands-on practice on docker for pulling image, creation of container ,running container, pushing it into docker hub, creation of image using the docker file.
3. Creating images using DockerFile and pushing the created image into DockerHub.
4. Upload the screenshots for above task and provide the link for docker hub

1. **Installation** of the docker( Follow the installation PPT and download the docker)
2. Sign In into Docker Hub
3. Click on personal-> continue with free
4. Go to gmail and verify your account
5. Once logged  into the DockerHub,there are no containers and Images in Docker Hub
6. Create a folder on desktop as”Docker”
7. Check Docker Version -- > docker –version

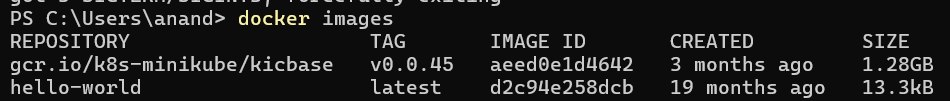


1. RUN image HELLO-WORLD IMAGE

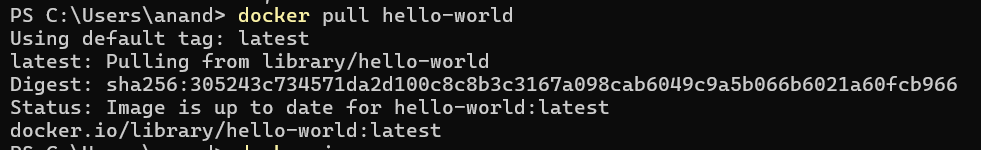
docker run hello-world



1. CHECK IMAGE 🡪 docker images

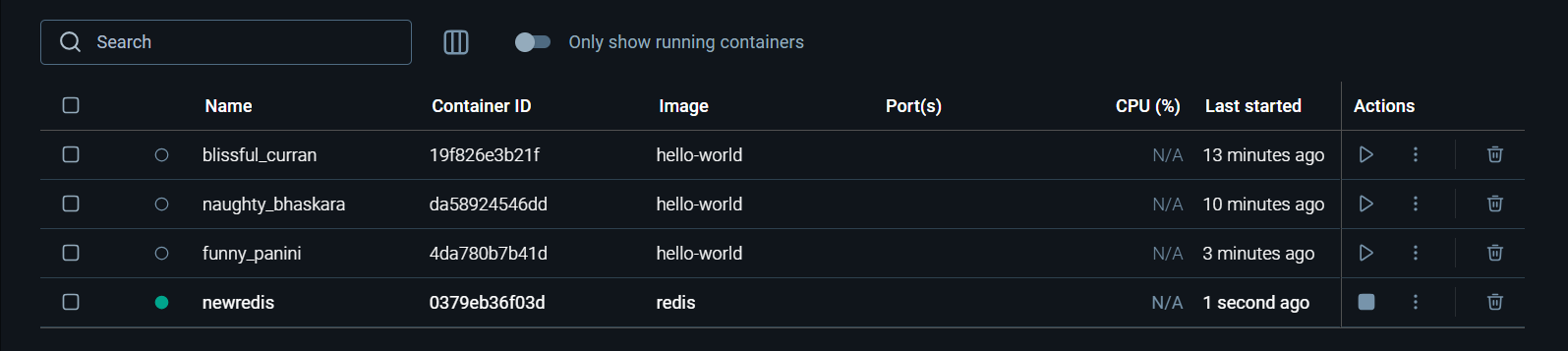


1. Pull Ubuntu Image from Docker Hub🡪docker pull ubuntu

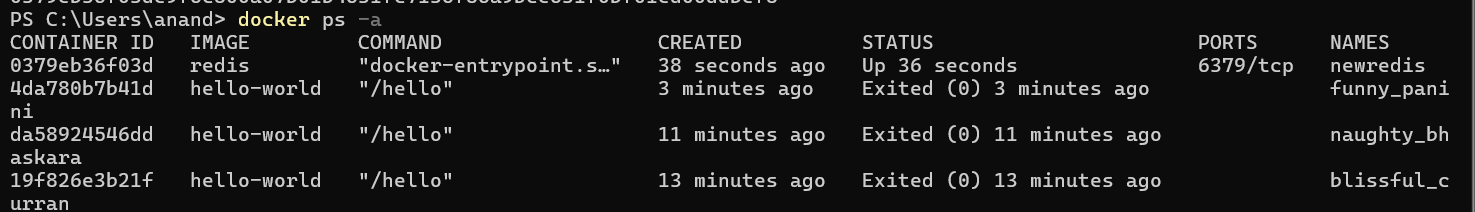


1. Run the image as docker container 🡪docker run -it -d ubuntu

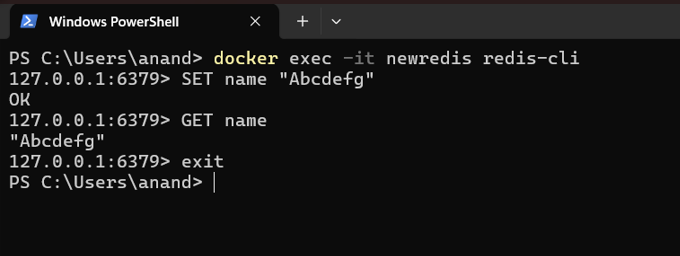


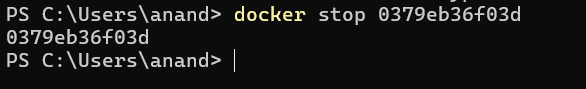


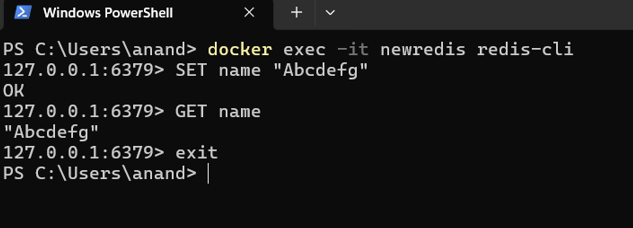
1. Check the  all running and stopped containers  🡪docker ps –a



1. During the execution of the container Ubuntu -> copy the container ID
2. To accessing/executing a running container (ubuntu) 🡪docker exec -it 1c6072756052 bash
3. Give echo message in the running container 🡪root@1c6072756052:/# echo Welcome to docker demo
4. Exit from the running container  🡪 root@1c6072756052:/# exit



1. To stop the running container 🡪 docker stop containerID 🡪PS C:\Users > docker stop 
2. run ubuntu with echo🡪docker run –it ubantu bash root@1c6072756052*:/# echo Anand*

  Pull the ubuntu image from the docker repository and run it as a container that will provide all the features of ubuntu

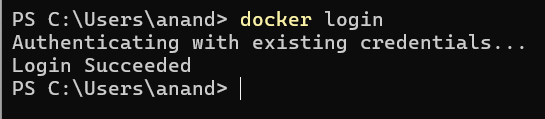
$ docker search ubuntu    displays **public images in ubuntu**



1. Pushing the image from local repository to docker hub

* Step1:create dockerhub account
* Step2:login using dochub
* *PS C:\Users > docker login*
* *Username:<< >>* ( here give user name)
* *Password:<< >>* ( here give user password)

*Login Succeeded*



**Creating Docker Image using Docker File**

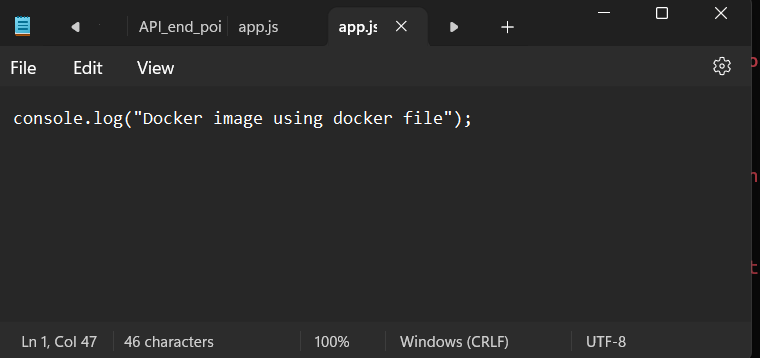
1. Create two files, one file Dockerfile without extension and the other app.js (.js file contains Javascript code for execution on webpages)

**a)Creating a sample javascript file**

**Type the following commands**

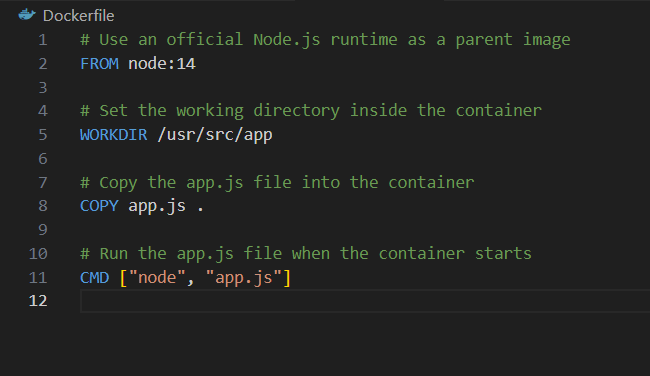
$ notepad app.js

 Console.log(“Docker Image using Docker File”)

**b)Creating Docker Image using Docker File**

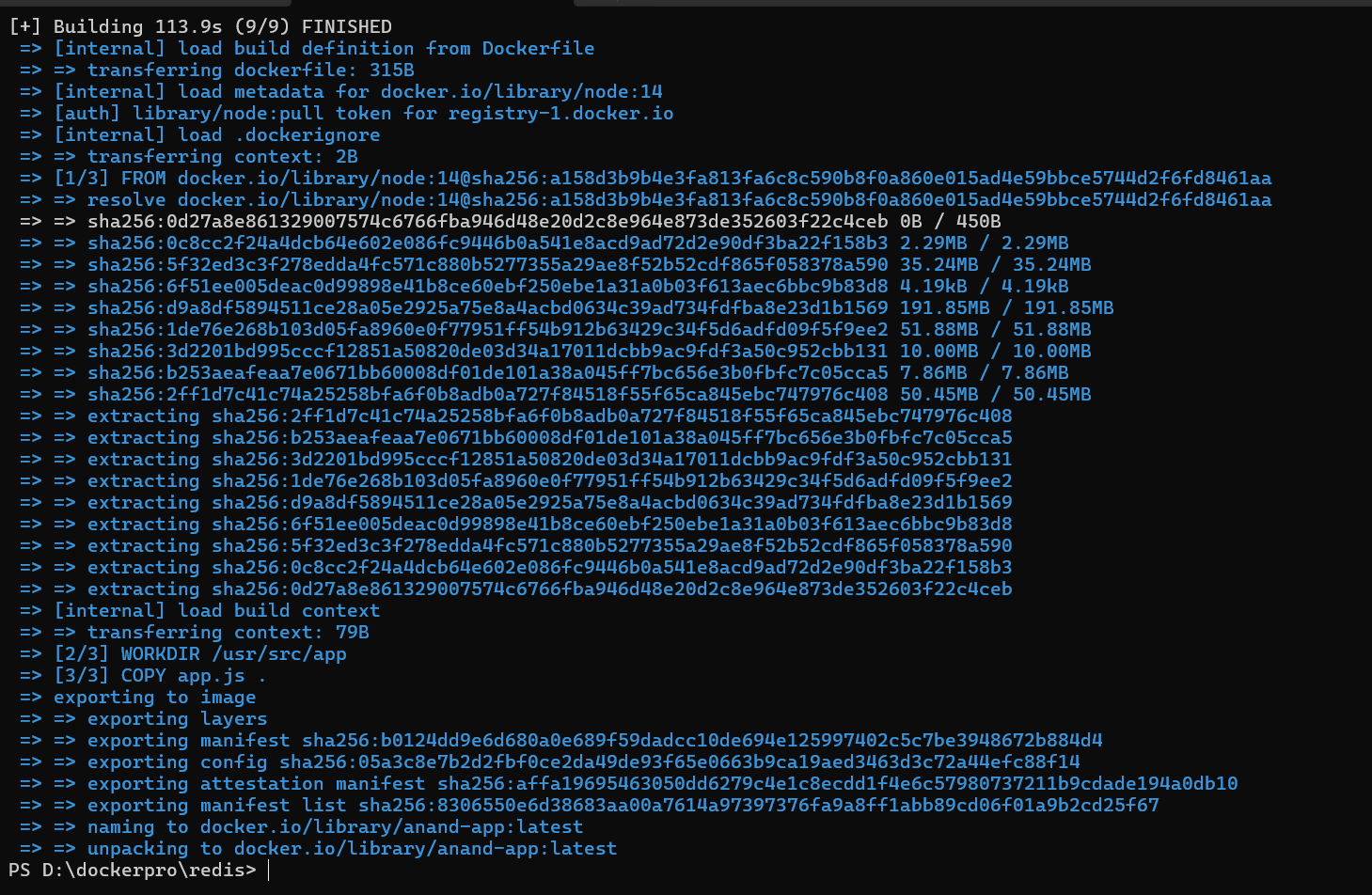
1. Type the following commands
2. Node app.js will execute app.js file in node

$ Vi dockerfile



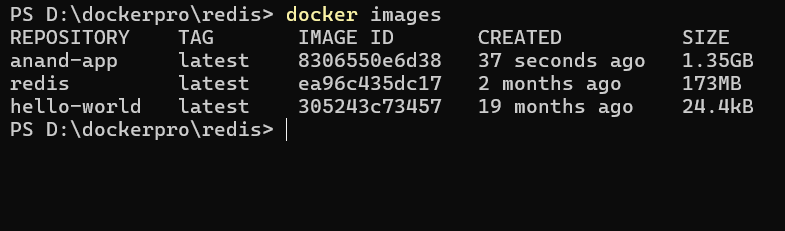
1. Press Esc-> :wq-> to save and exit

$ docker  build –t anand-app .

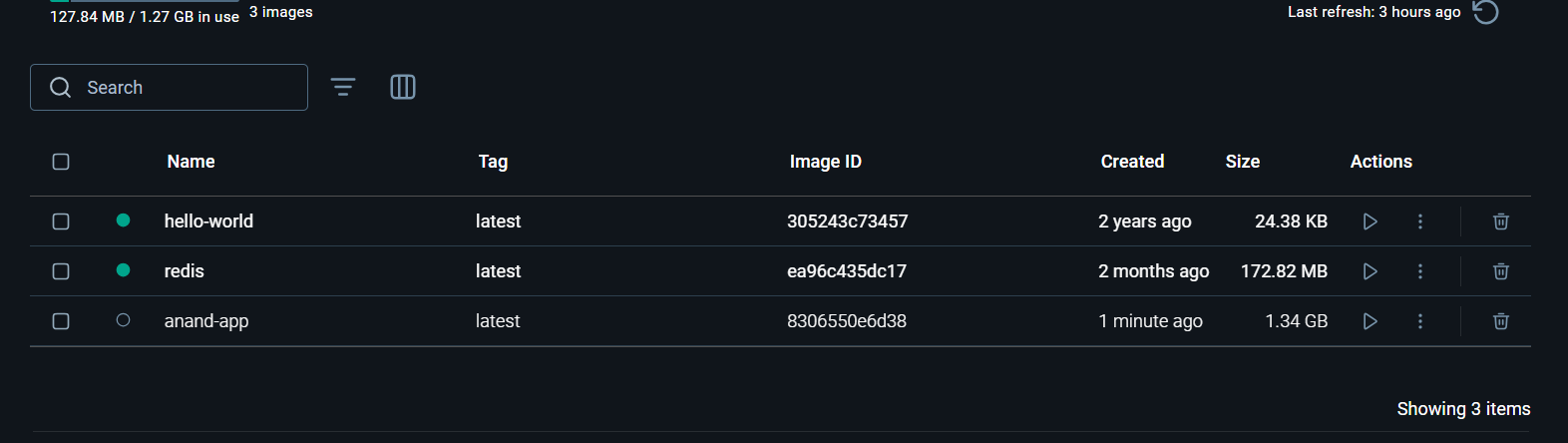


$ docker images

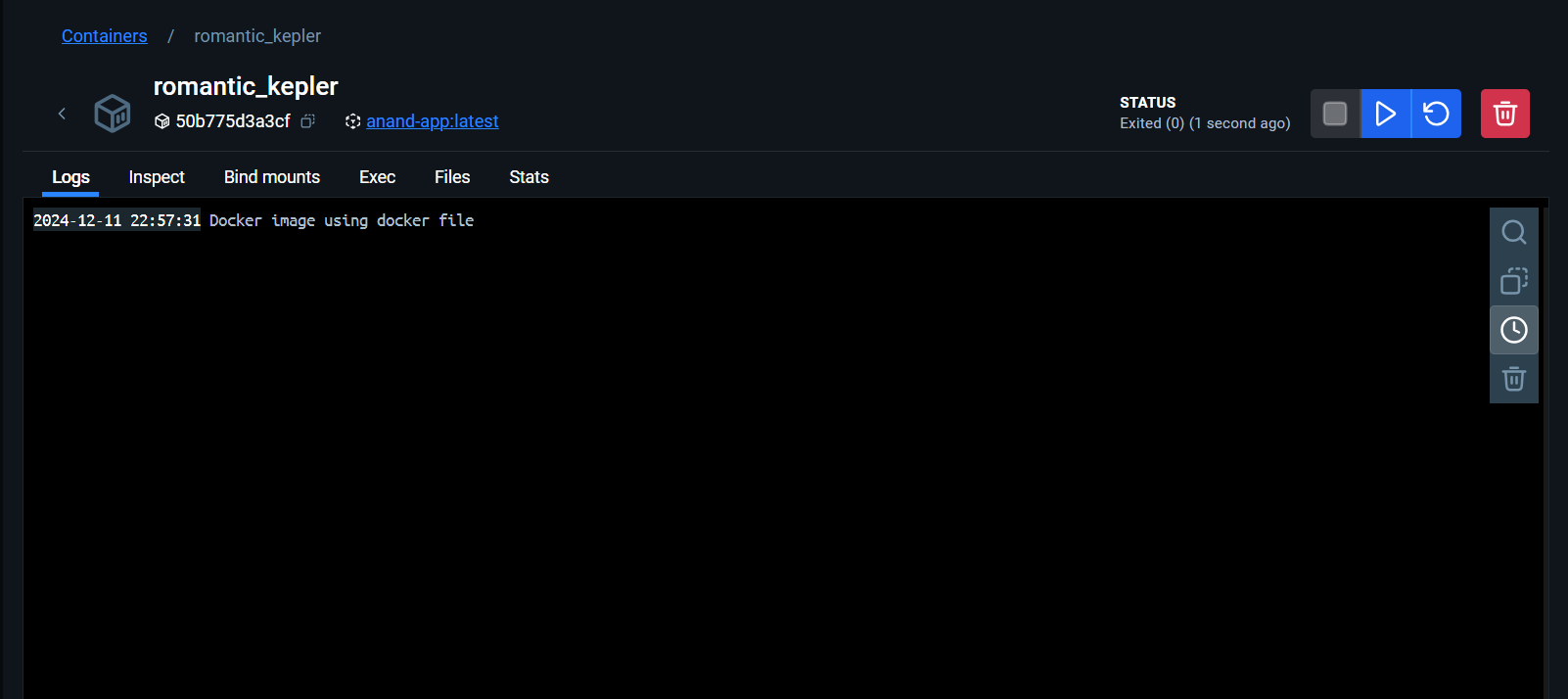
  myimage will display here



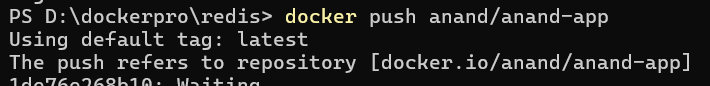
1. Go to docker desktop-> local-> check myimage will exist-> click on run



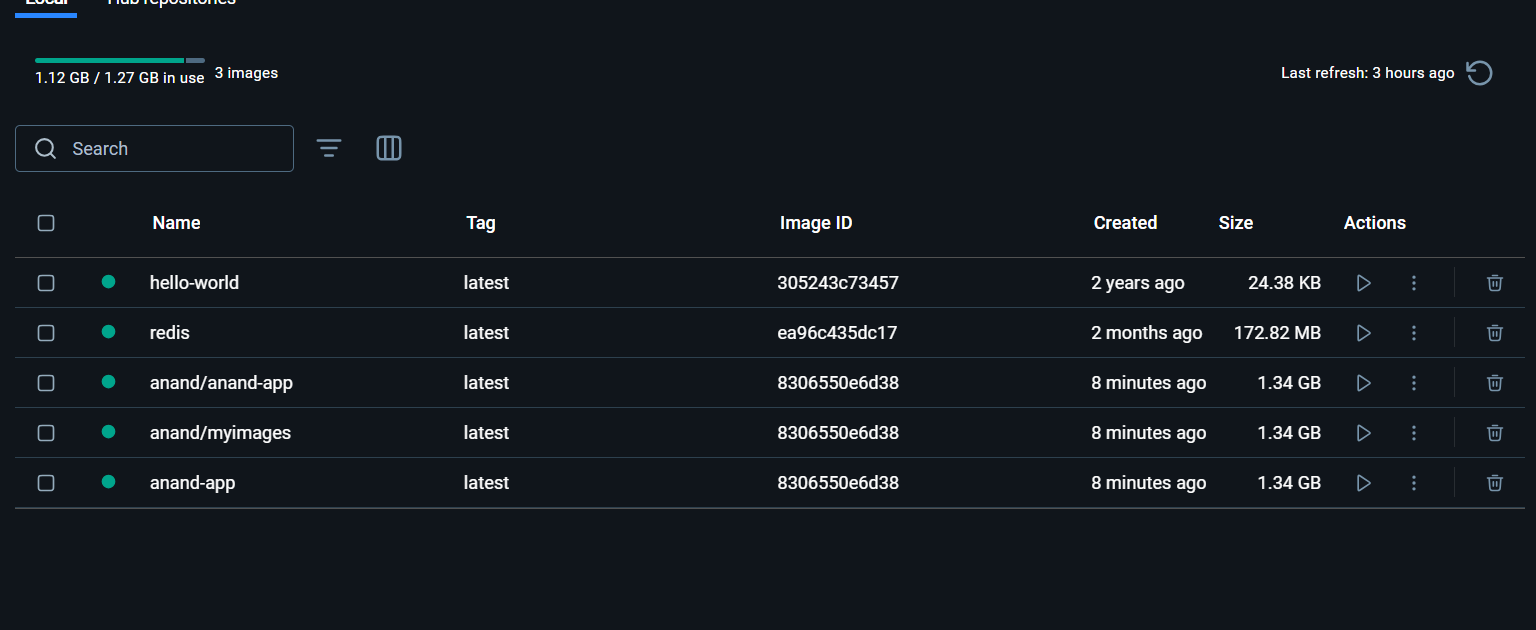
1. Diplays the console output as “Docker Image using Docker File”



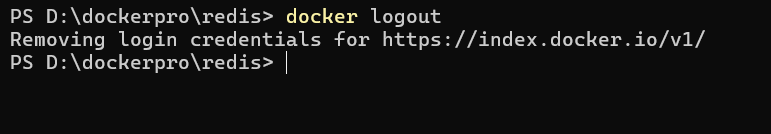
1. $ docker  tag myimage anand/myimage
2. $ docker  push anand-app/myimage



1. Go to desktop docker and check under the specified directory myimage will display



1. $ docker  logout

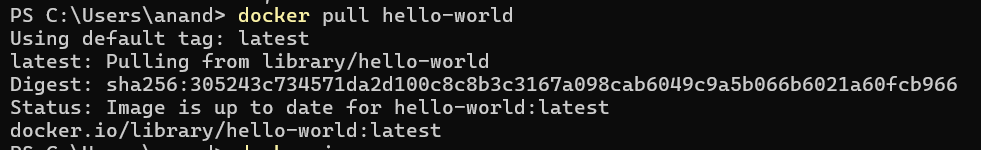


Docker Commands

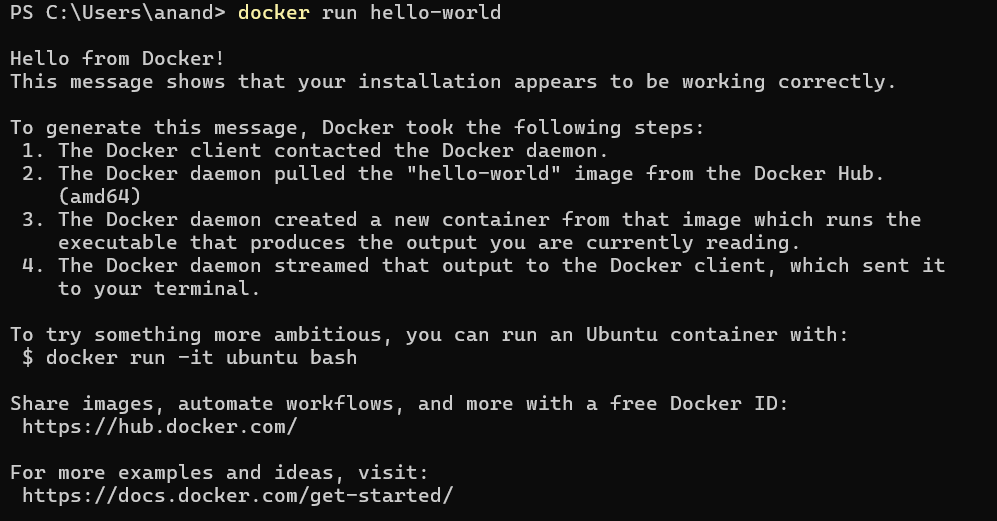
1. git version



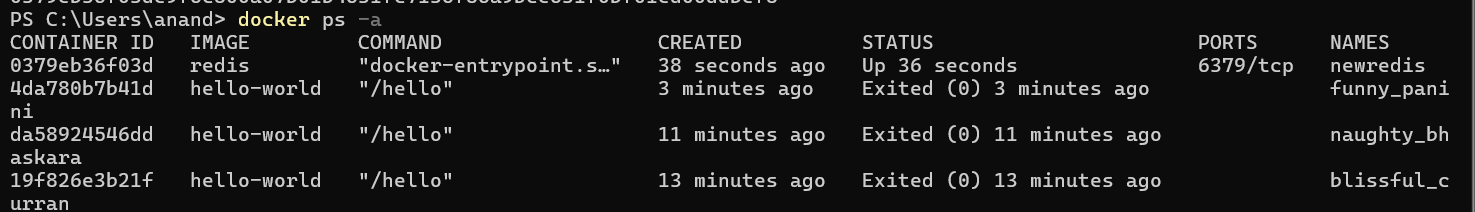
2. Docker pull hello-world



3. Docker run hello-world



4. Docker ps -a

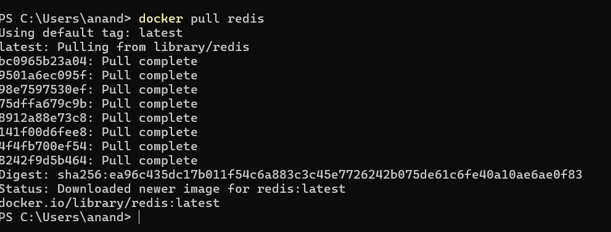


5. Docker rm id

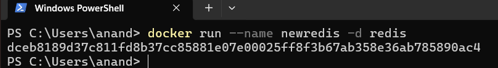


Docker CLI commands with redis

1. docker pull redis



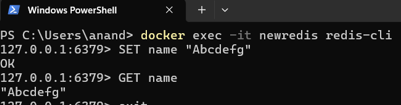
2. docker run –name my-redis -d redis



3. docker ps

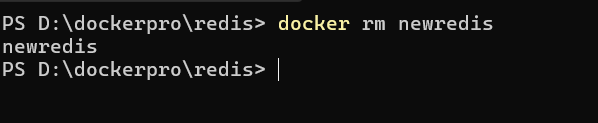


4. docker exec -it my-redis redis-cli



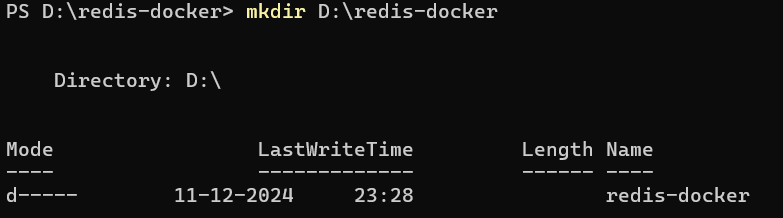
5. docker stop my-redis

6. docker rm my-redis

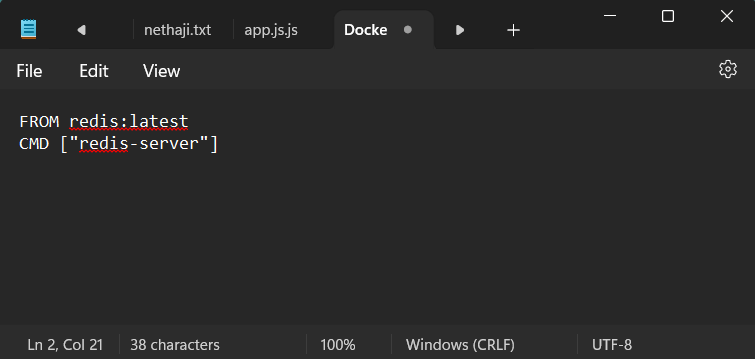


Docker commands using Docker file

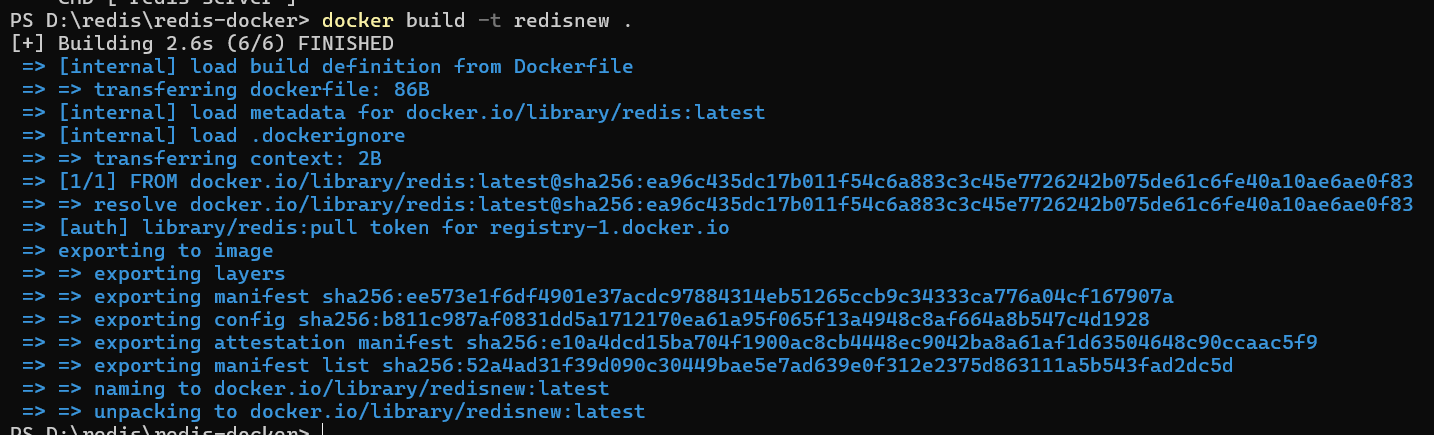
i. set up folder



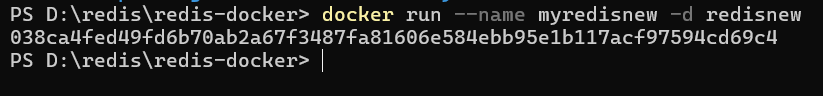
ii.write the dockerfile



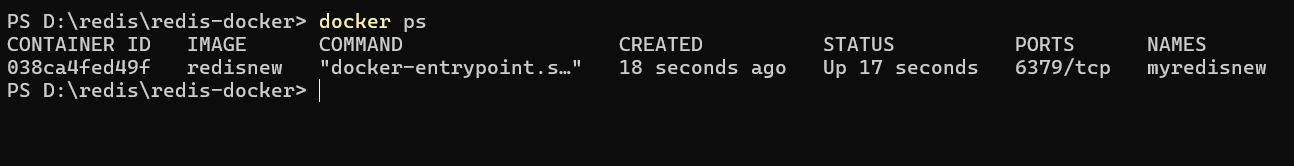
1.docker build -t redisnew



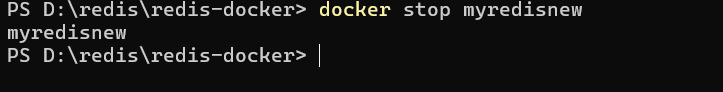
2.docker run –name myredisnew -d redisnew



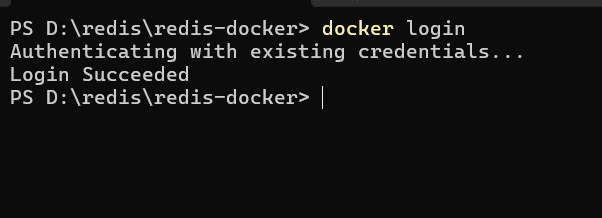
3.docker ps



4.docker stop myredisnew



5.docker login



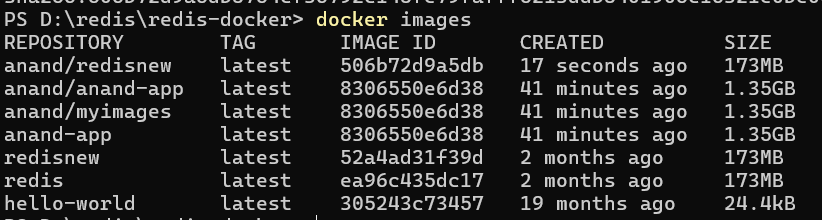
6. docker ps -a



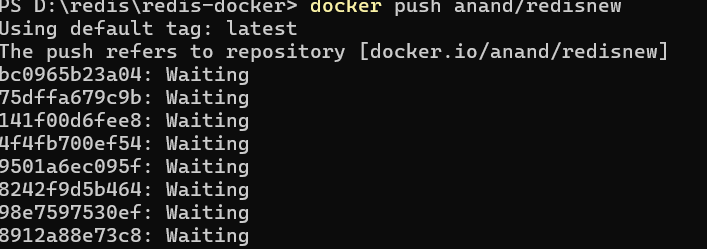
7.docker commit



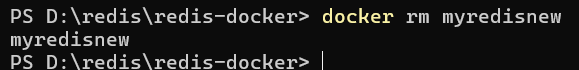
8.docker images



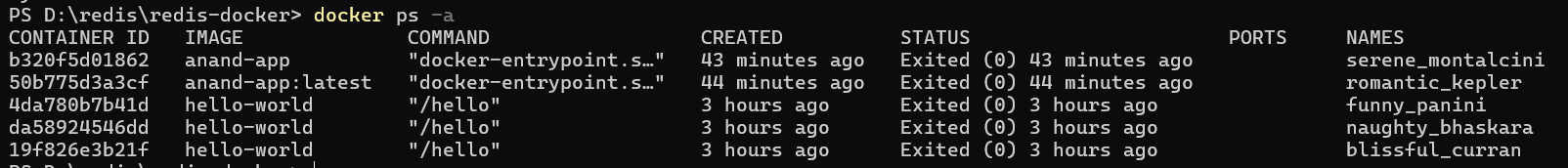
9. docker push anand/redis



10.docker rm



11.Docker ps –a



12.docker logout



13.docker pull anand.redis

14.docker run –name ,myredis -d anand/redis1

15. docker exec -it myredis redis -cli

16. SET name “222bd1a6615”

17. GET name

18.exit

20.docker pa -a

21.docker stop myredis

22.docker rm 283910

23.docker images

24. docker rmi anand/redis