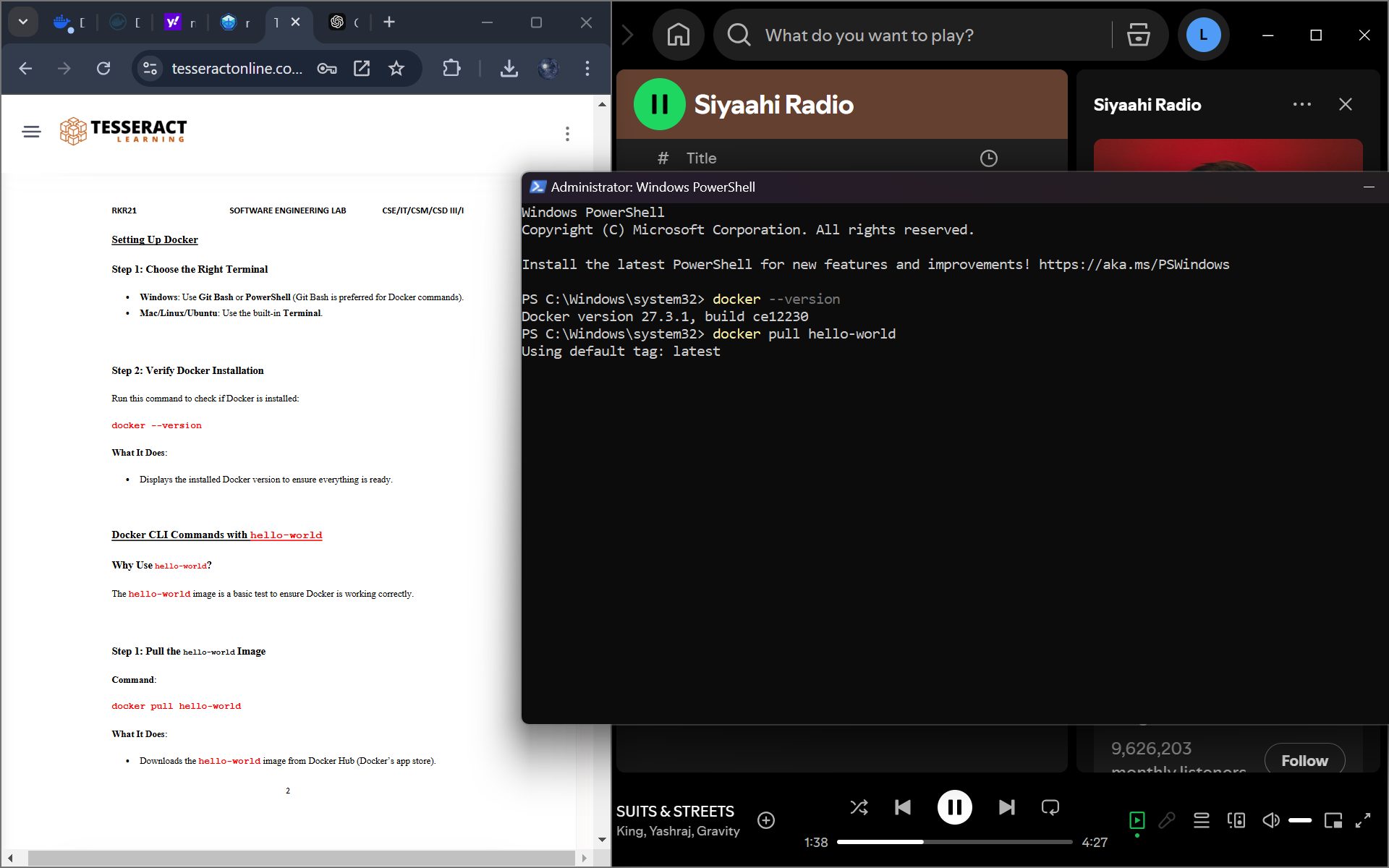
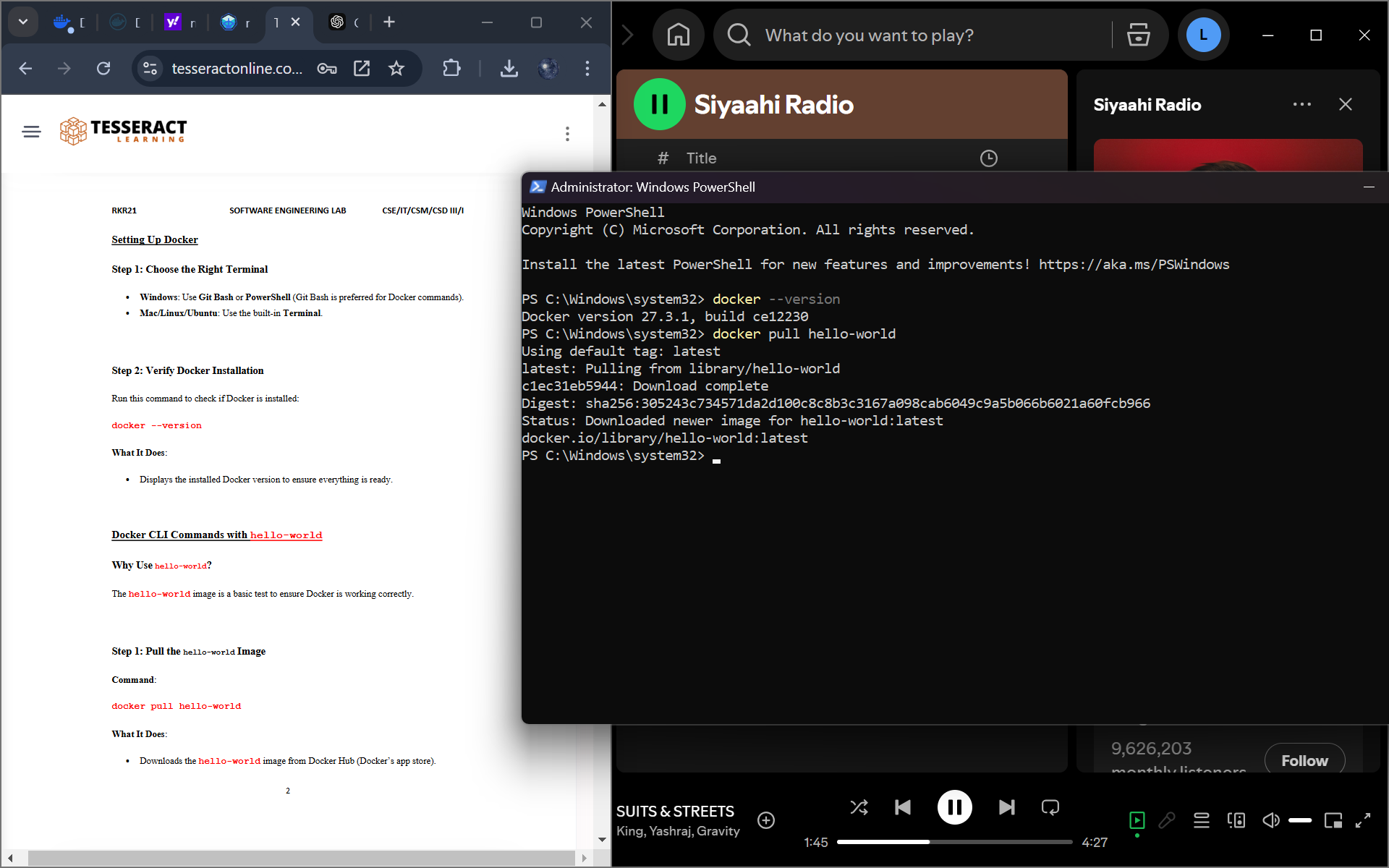
DOCKER COMMANDS

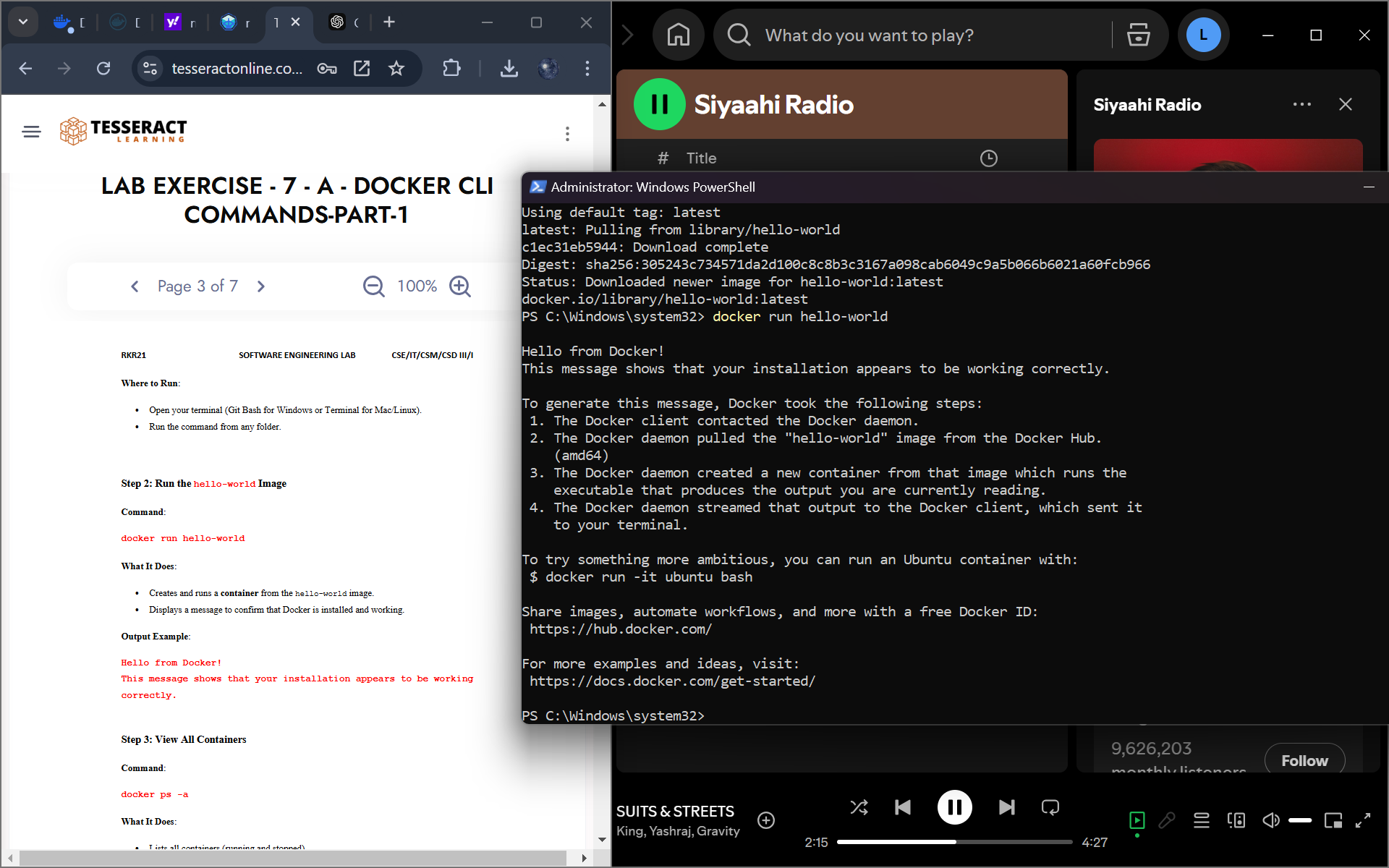
Check docker version using docker –version



Logging into docker

Using docker pull to pull and hello-world image from docker hub. 

Running the hello-world image using docker run command



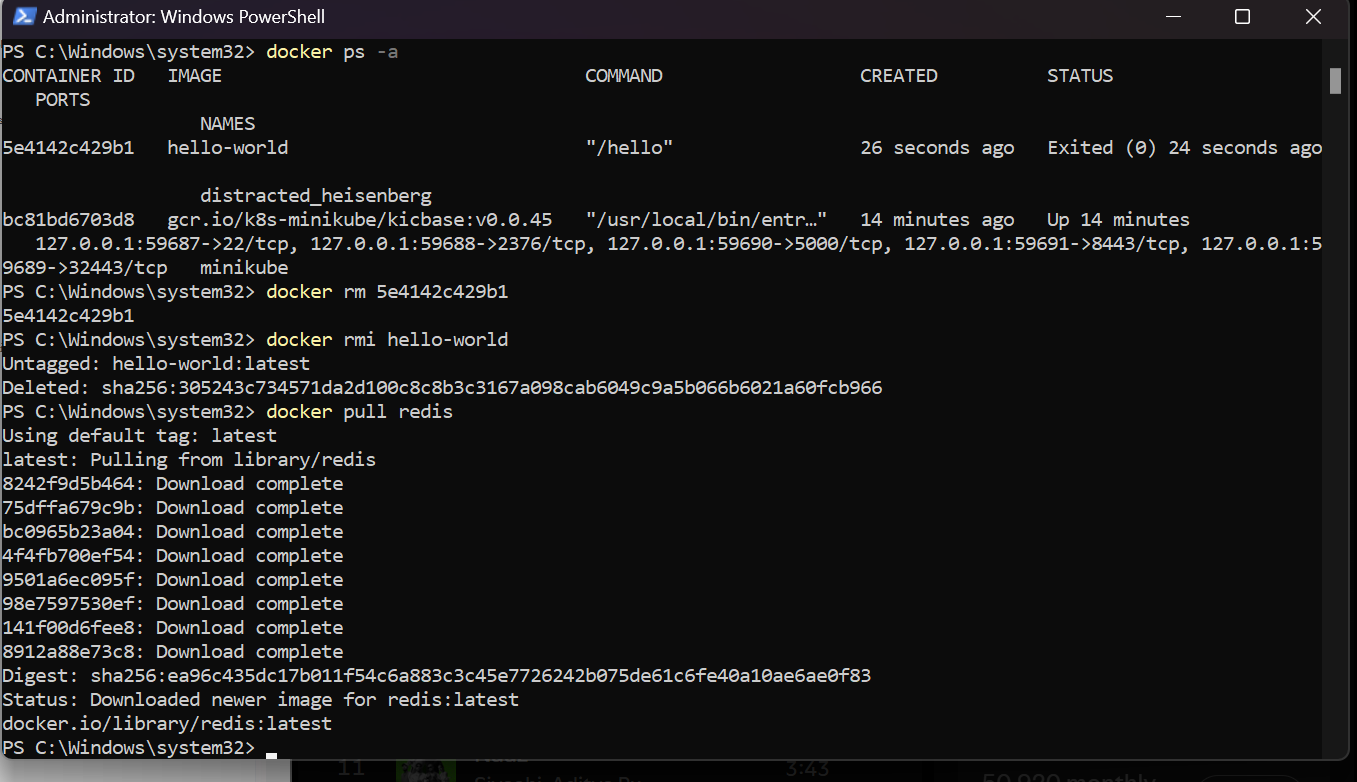
Lists all running Docker containers using docker ps

Lists all Docker containers, including running, stopped, and exited ones using docker ps -a.

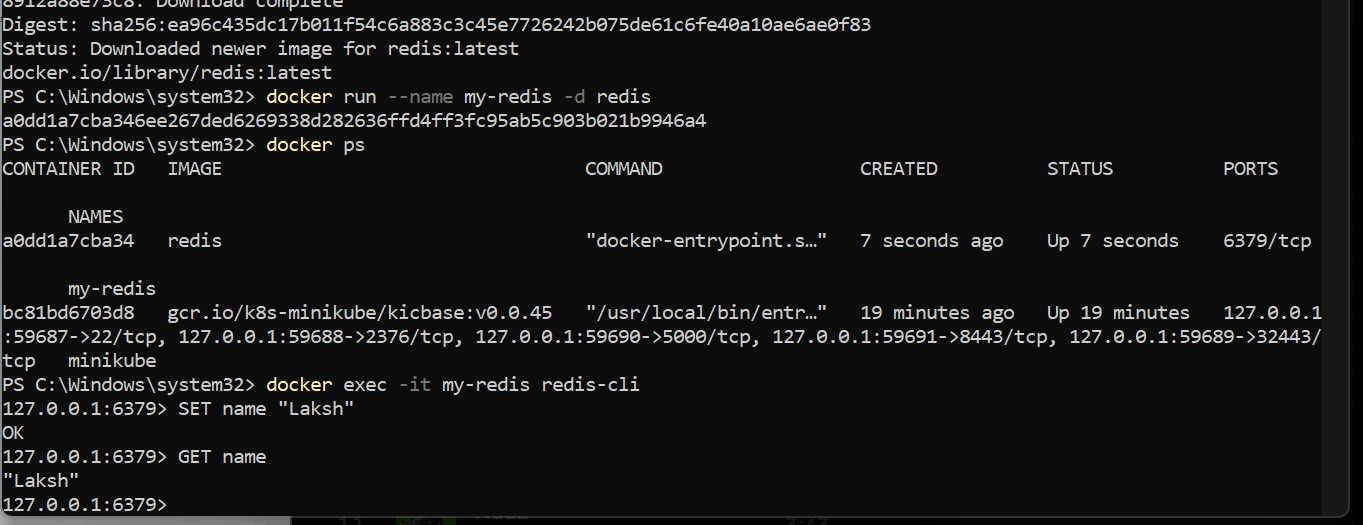
**R**emoving a stopped Docker container using docker rm.

Removing a Docker image from the local system using docker rmi.

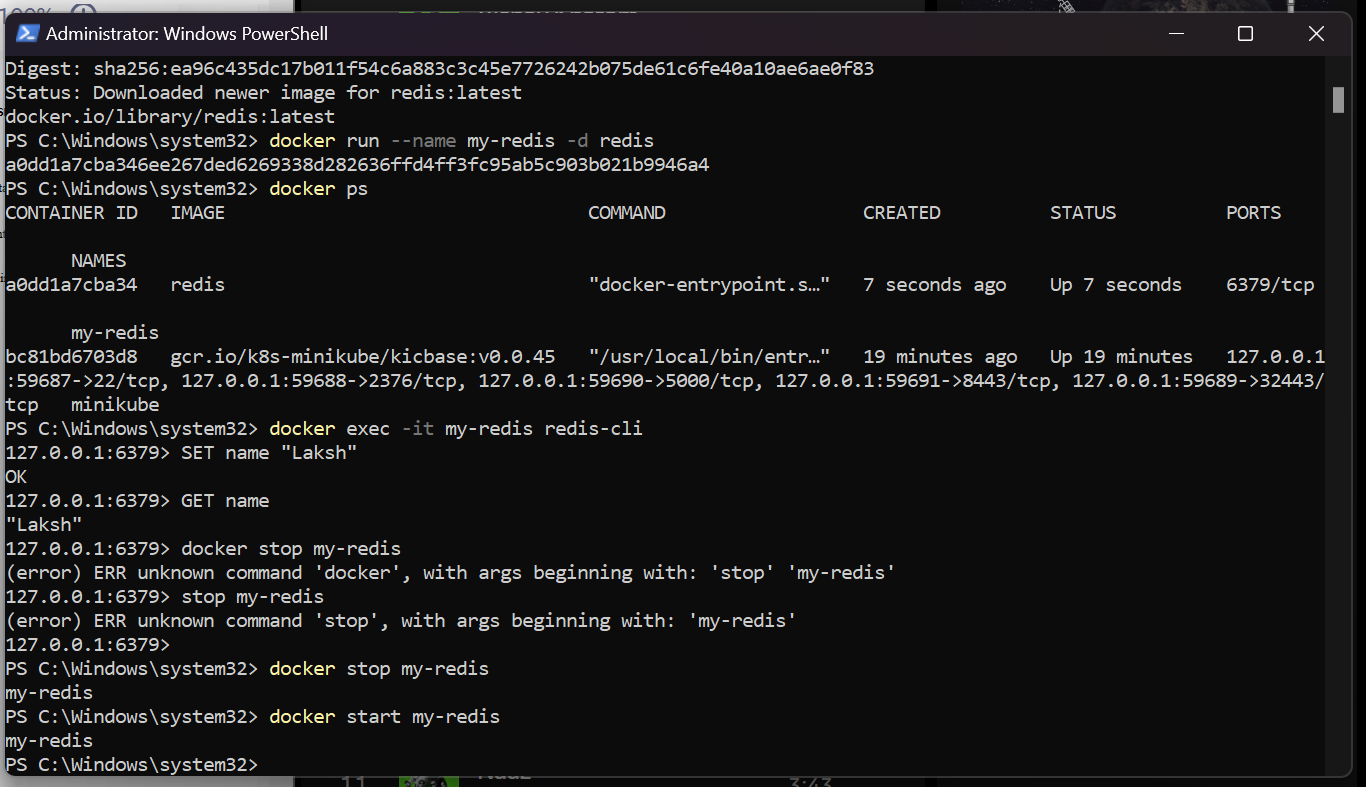
Downloading a Docker image from docker hub using docker pull.



Running a new container named newredis in detached mode using the redis image.

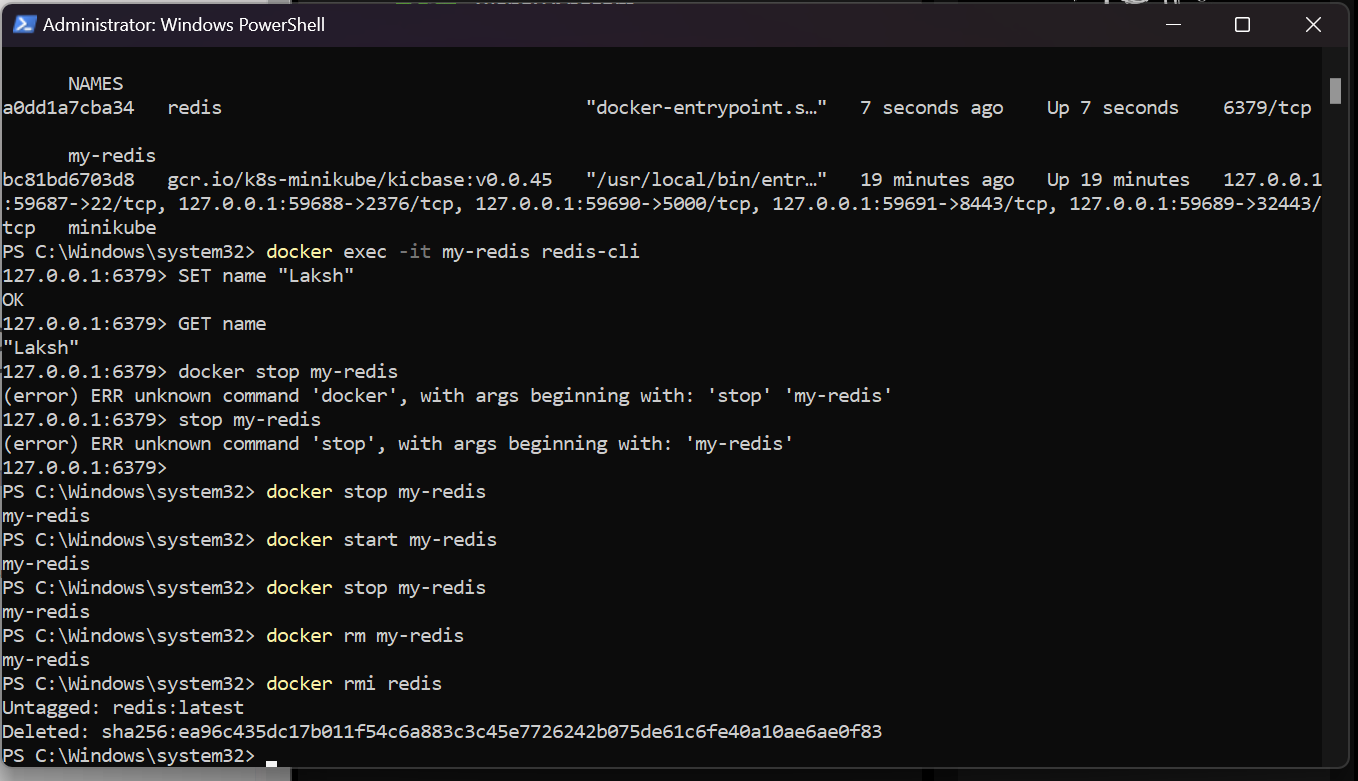
Starting an interactive session inside the newredis container, running the redis command like set and get. 

Stopping the newredis container that is running.



Starting the newredis container.

Deleting an continer using the command docker rm containerID and docker image using docker rmi.

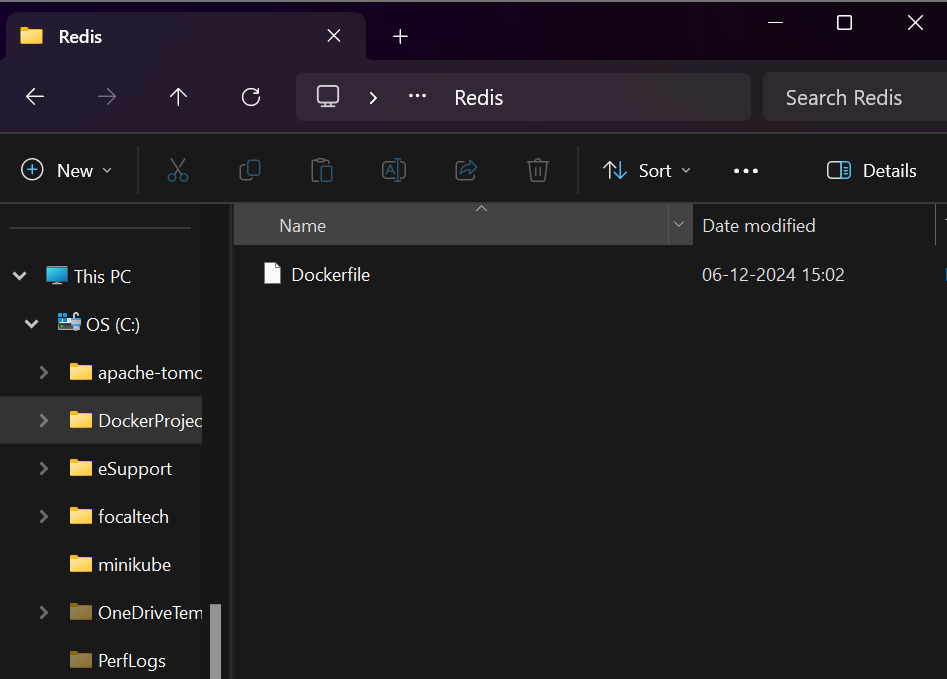


CREATING A CUSTOM DOCKER FILE

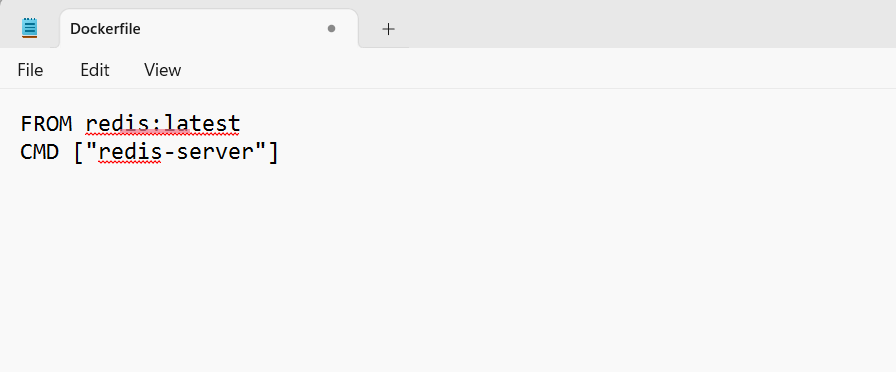
Creating a new Folder named redis1

Creating a one more new folder inside it with the name of redis to fetch the redis custom image.

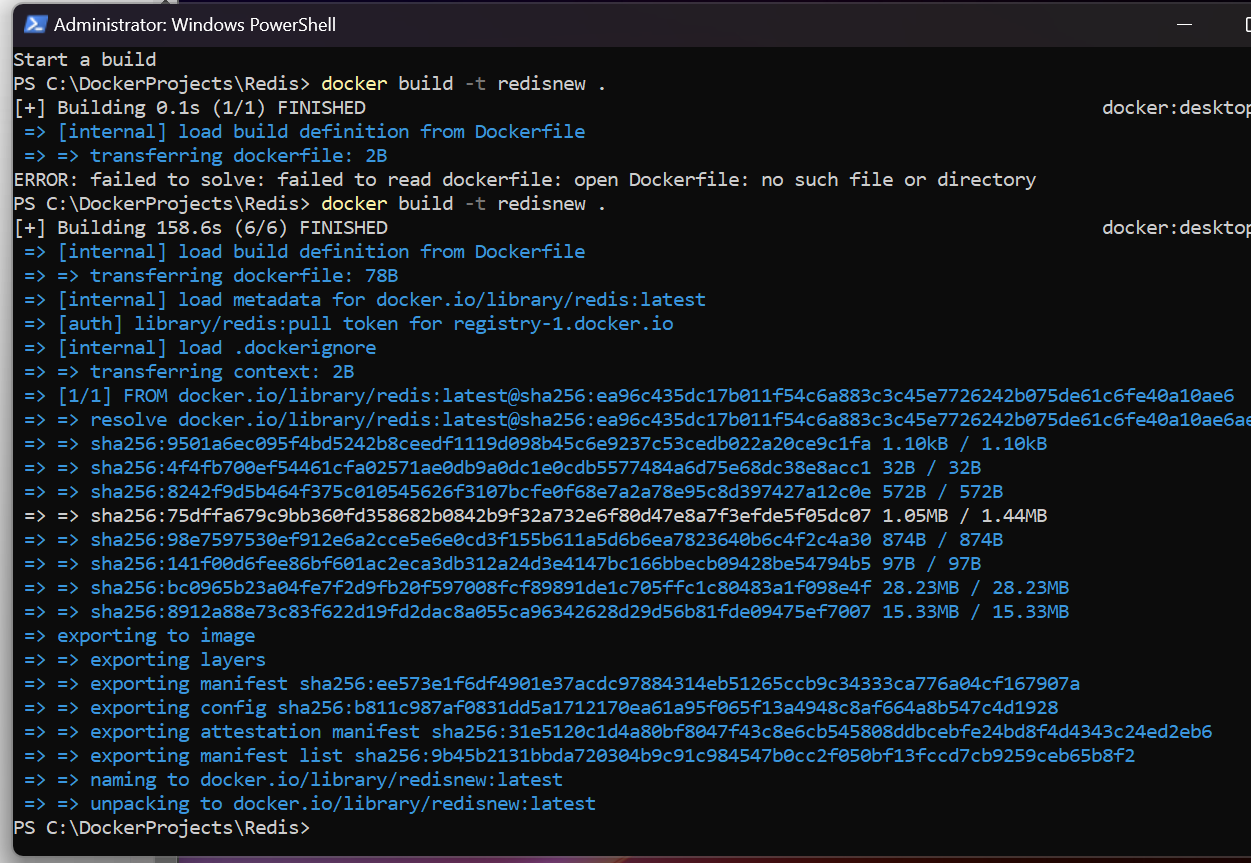
Create a new File name Dockerfile without any extensions.



Open the file using any editor example notepad and Copy the Following Code into the file which fetches the redis of latest version and runs the redis server.



Open the Redis folder in a cmd prompt using administrator mode.

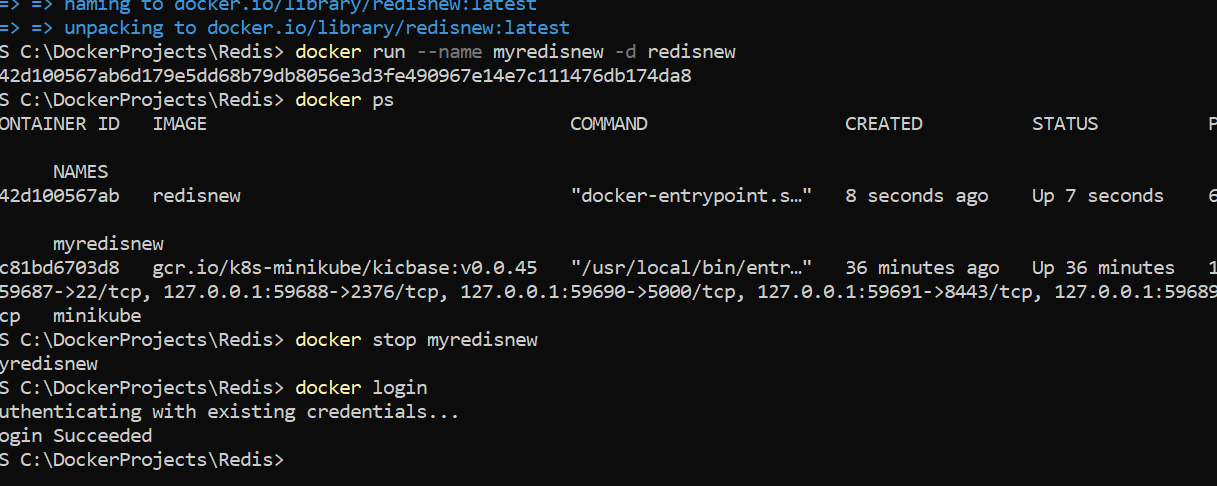
Building the image with the current image using the cmd docker build -t redisnew . 

Run the command docker images to view all the images available

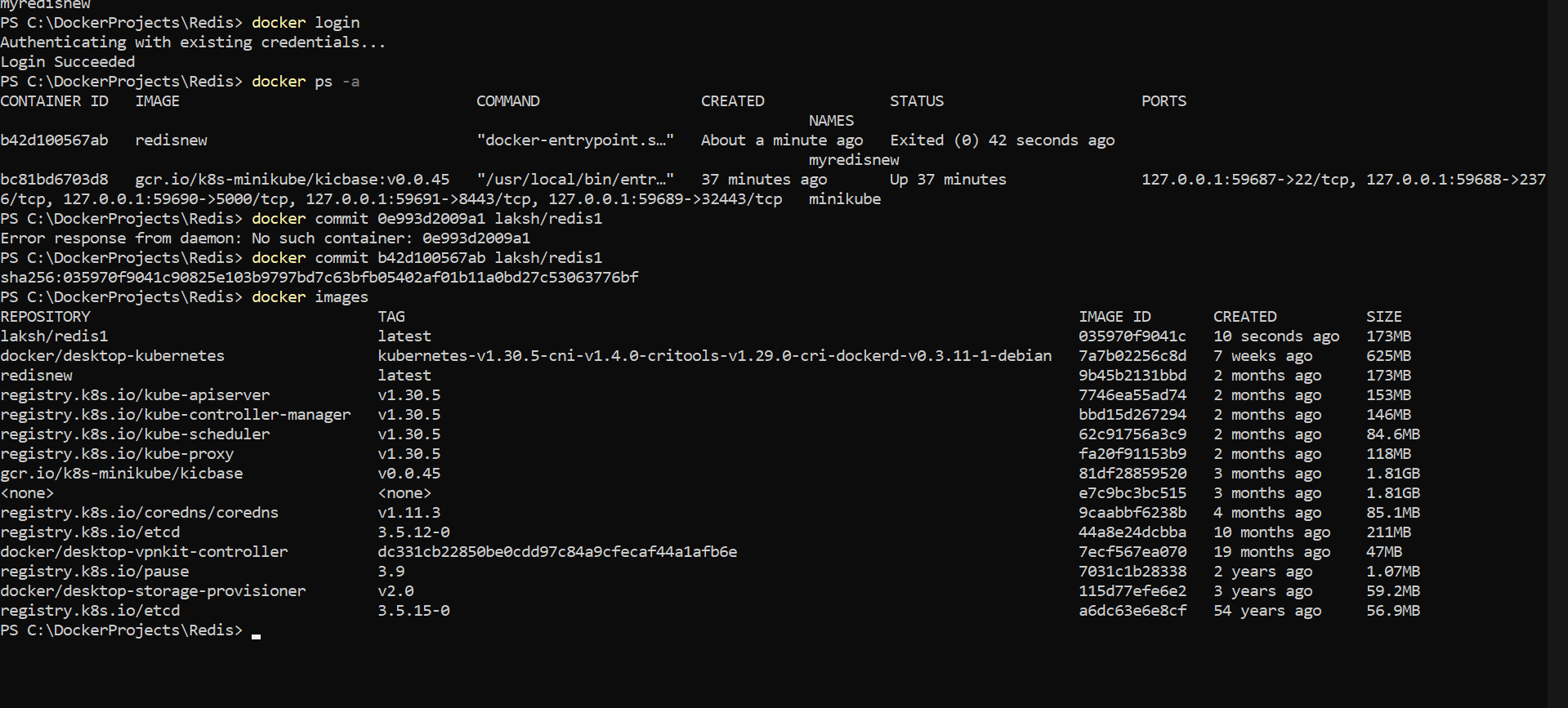
Running the new latest redis image that we created.

Viewing the docker container that are running.

Stopping the docker container od redislatest.



Commit the changes into the system using the command docker commit contianerId newname. Now using this new name i.e your username folloew by /name of the container is created in to docker hub



We can see that the images is saved locally by the name iamabdurrasheed/redisnew.

Using the docker push nameOfimage we can push our hub into the docke hub cloud.

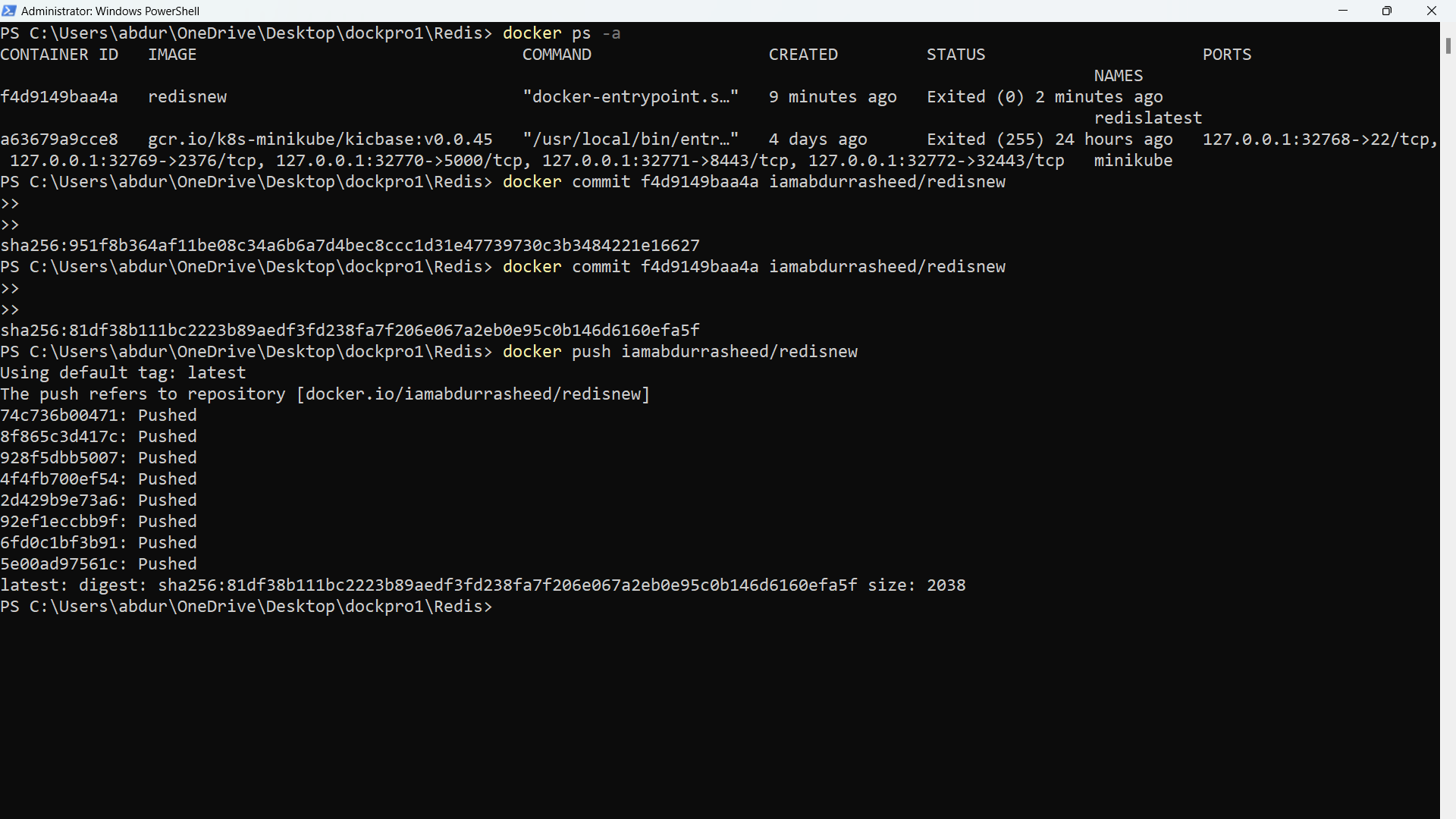
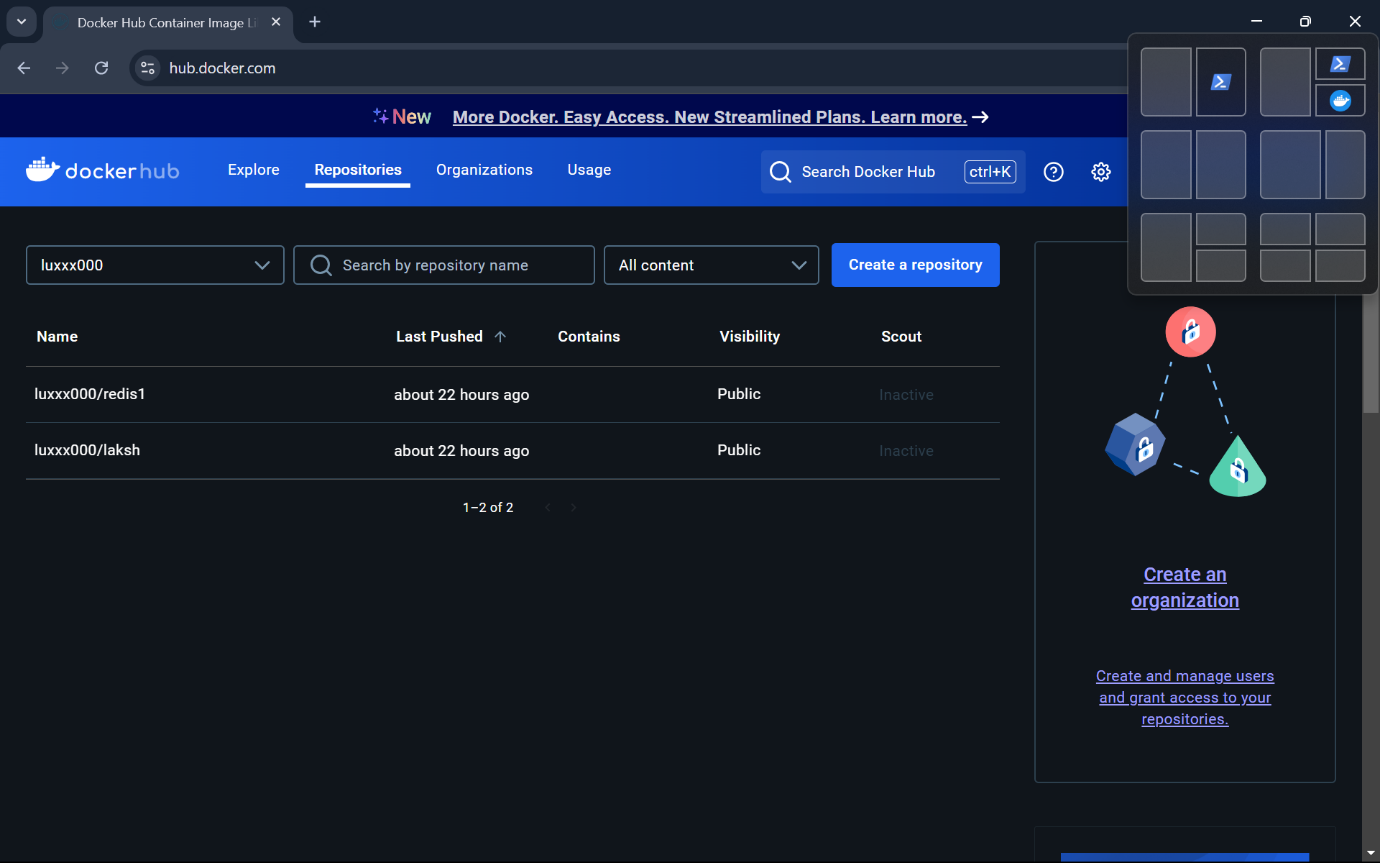
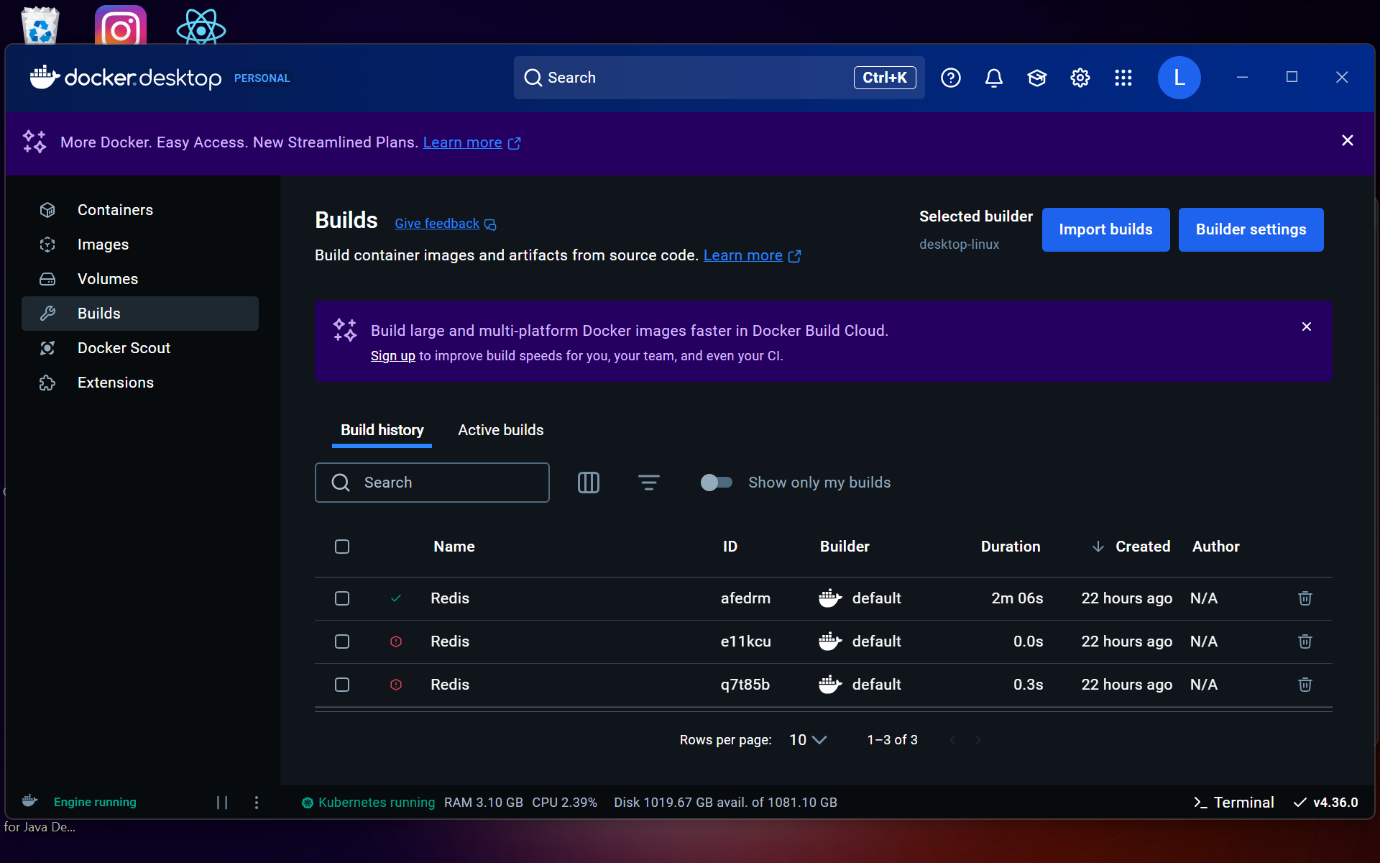


Image after it is pushed into the docker hub. 

Now removing the images and container locally.

Pulling the image from the hub that we just pushed into the docker hub.

Running the new redis file just we just pulled.

Opening the interactive redis cli and setting the name and getting the name that we stored temporarily.

Stopping the container that was running.

Deleting the container and images after it is stopped.

