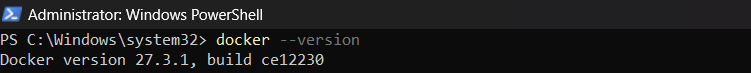
**DOCKER CLI COMMANDS**

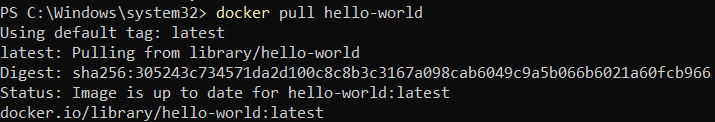
docker –version:

This displays the docker version



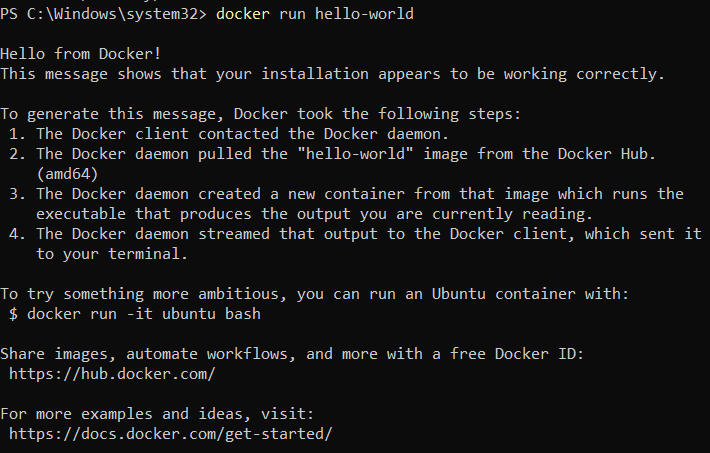
docker pull hello-world:

Downloads the hello-world image from Docker Hub (Docker’s app store).



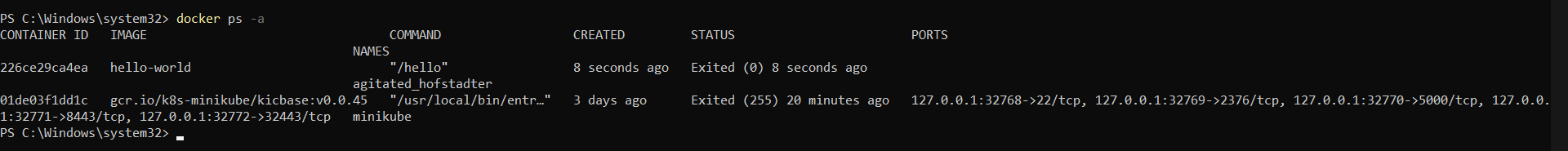
docker run hello-world:

Creates and runs a container from the hello-world image and displays a message to confirm that Docker is installed and working.



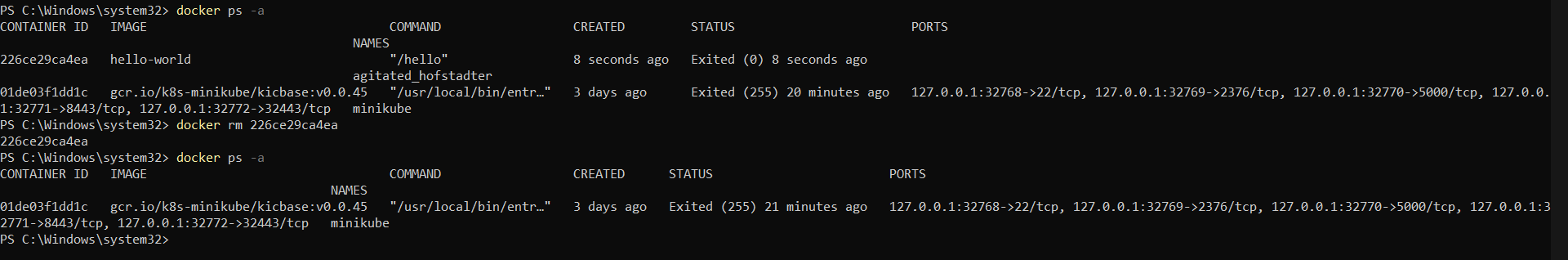
docker ps -a:

Lists all containers (running and stopped)



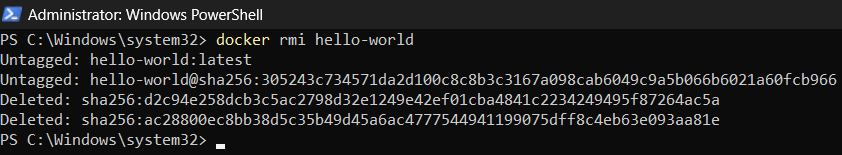
docker rm [container-id]:

Deletes the container to free up space



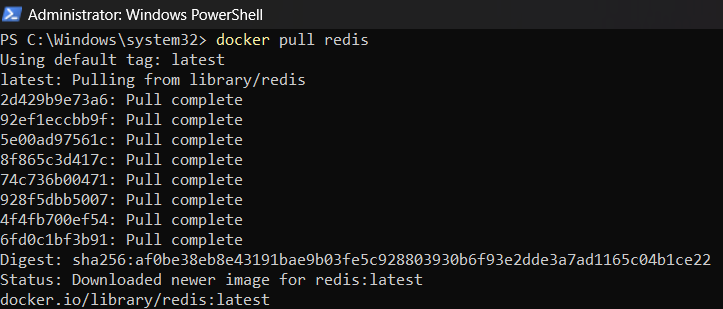
docker rmi hello-world:

Deletes the hello-world image if you no longer need it



docker pull redis:

Downloads the official redis image from Docker Hub to your system



docker run --name my-redis -d redis:

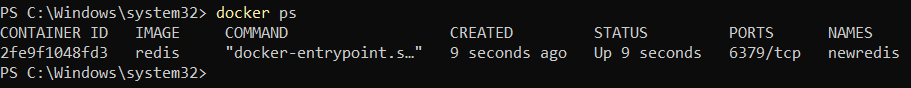
Creates and starts a container named my-redis from the redis image.

The -d flag runs the container in the background.



docker ps:

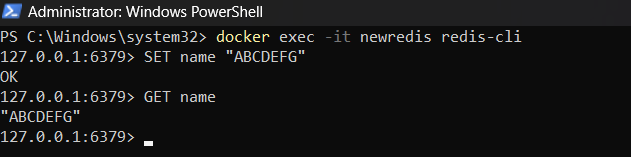
Lists all running containers



docker exec -it my-redis redis-cli:

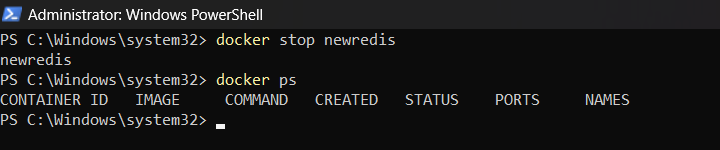
Opens the Redis command-line tool (redis-cli) inside the container.

You can now send commands directly to the Redis server.



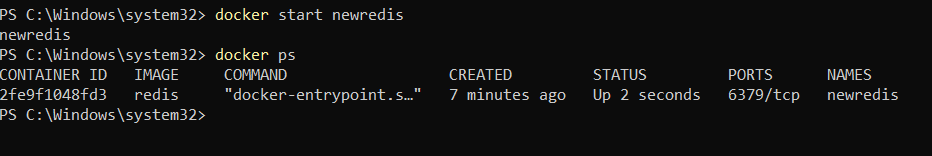
docker stop my-redis:

Stops the Redis container but doesn’t delete it



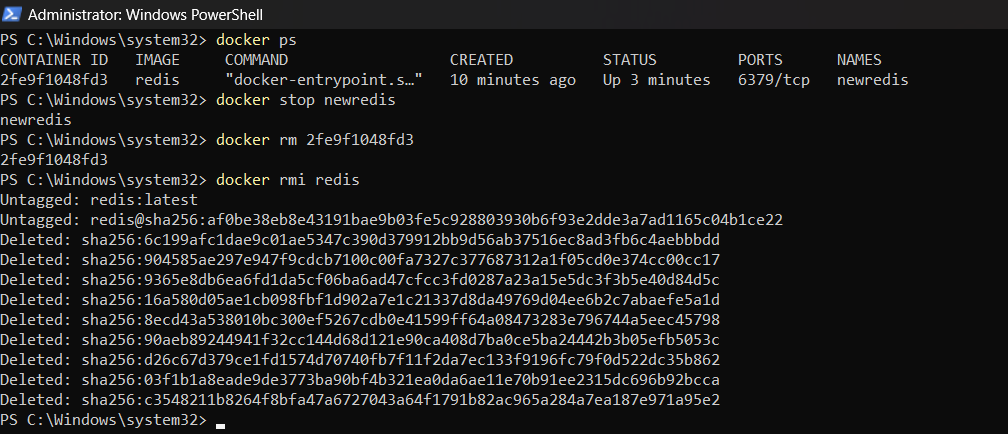
docker start my-redis:

Restarts the stopped container



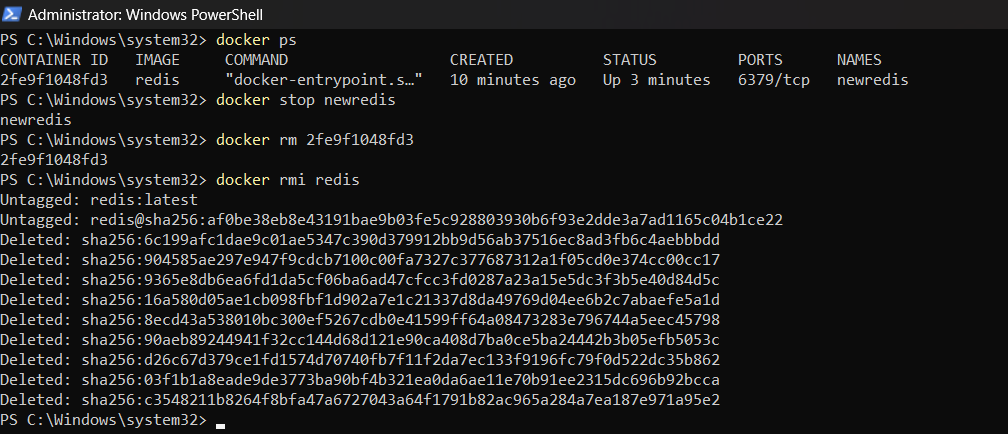
docker rm my-redis:

Deletes the container permanently

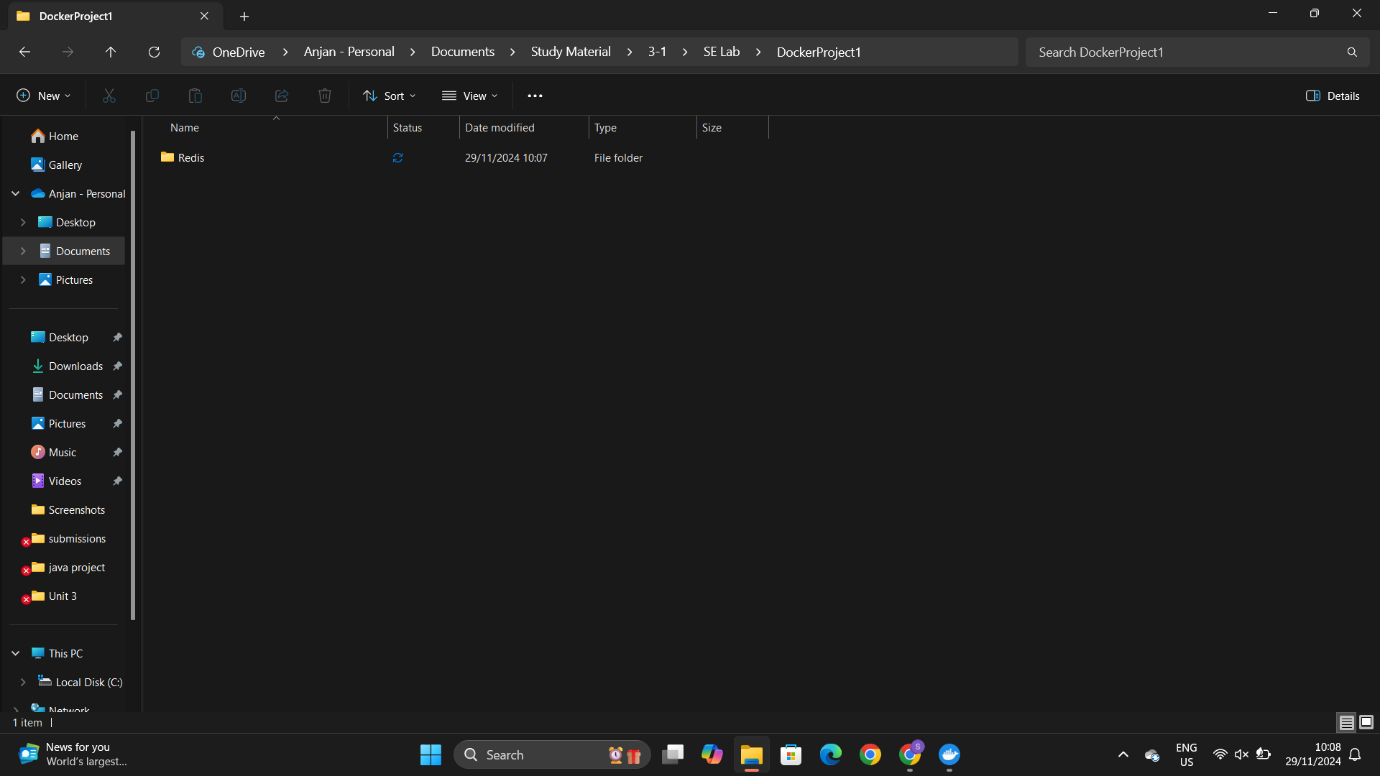


docker rmi redis:

Deletes the Redis image from your local system

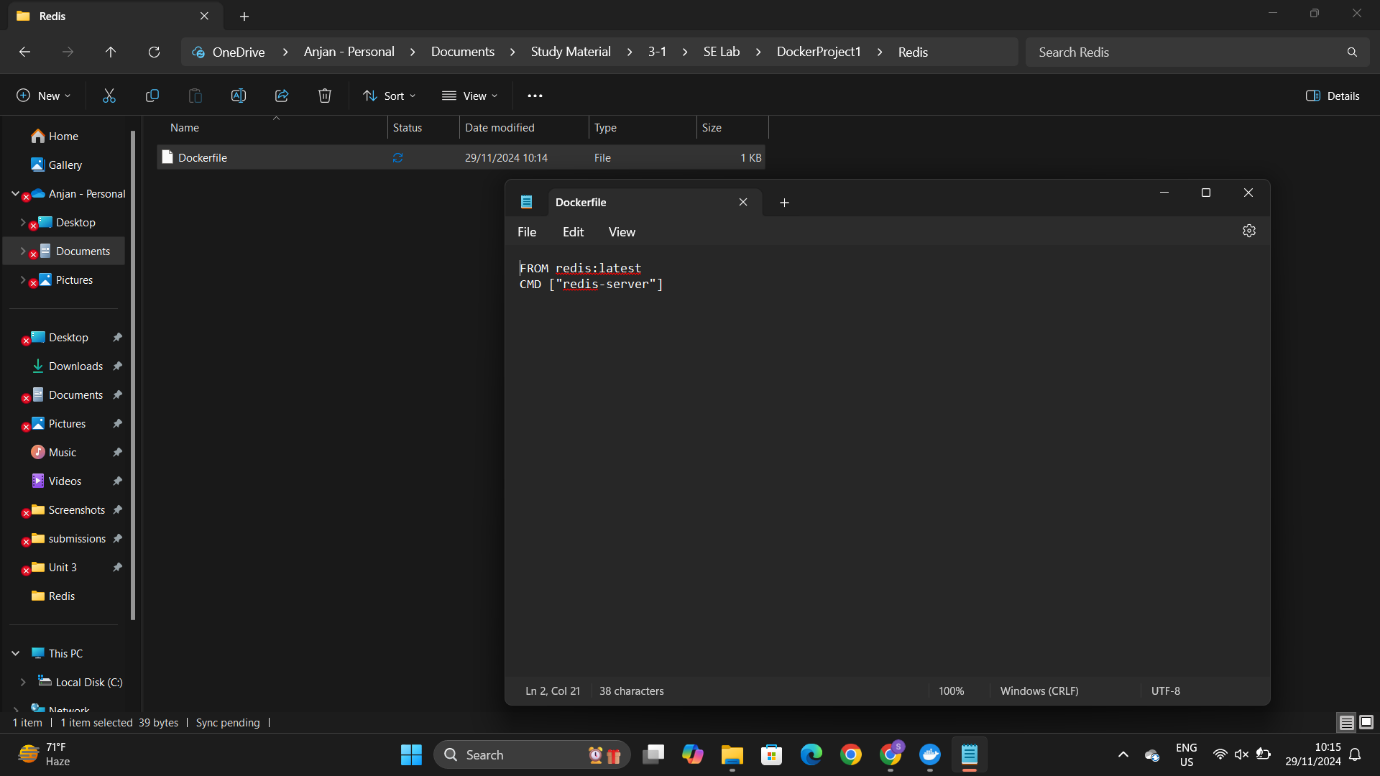


Create a folder for docker project. In that create a folder named redis

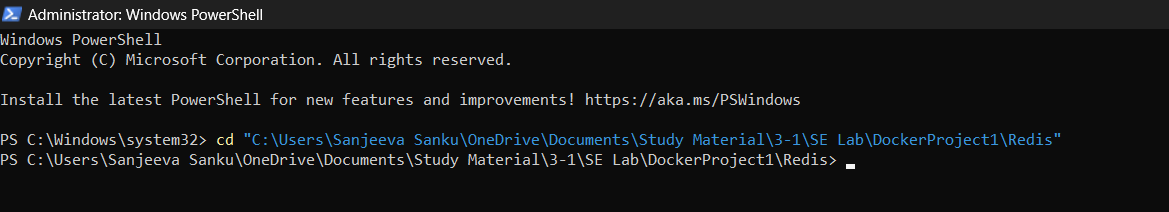


1. Inside the folder, create a file named Dockerfile (no extension).

2. Add the following content: FROM redis:latest CMD ["redis-server"]



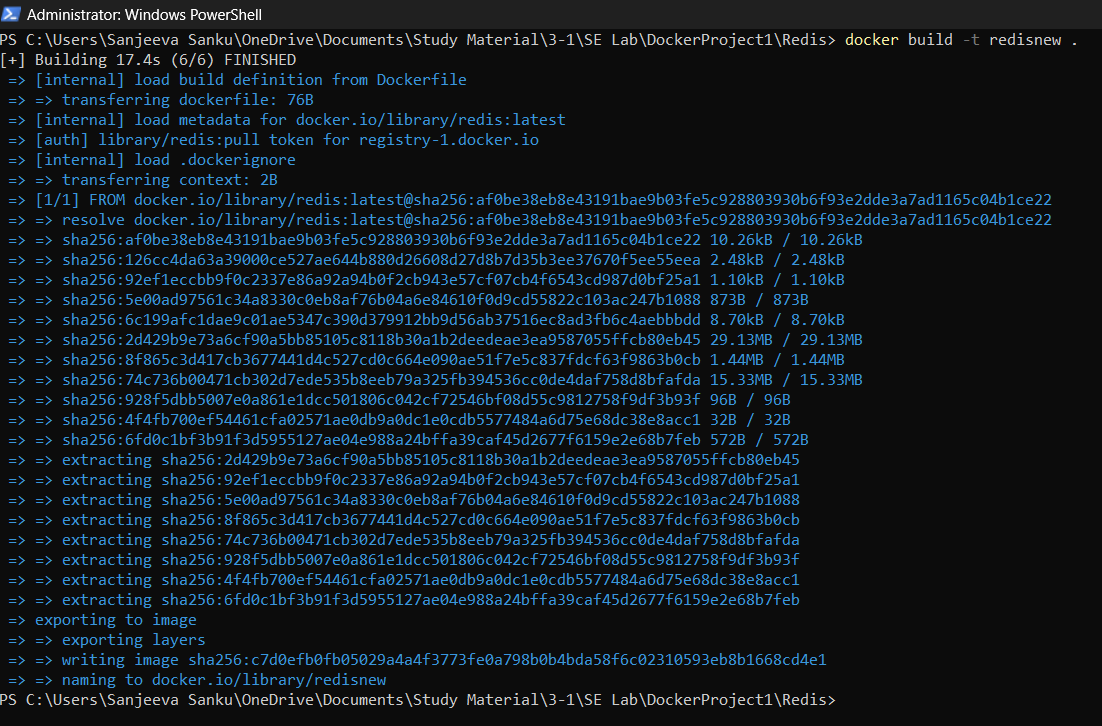
Change the directory



docker build -t redisnew:

This creates (builds) a Docker image using the recipe (Dockerfile) in the current folder (.).

-t redisnew: Gives the image a name/tag ("redisnew"), so you can find it easily.

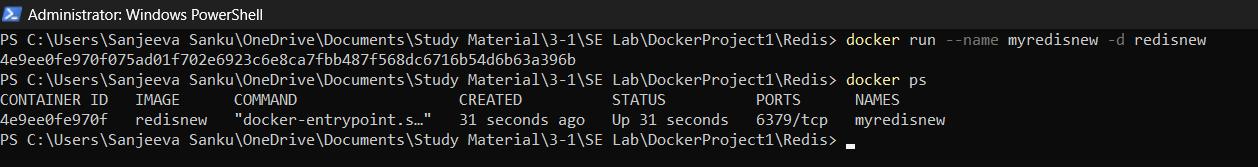


docker run --name myredisnew -d redisnew:

Starts a new container (mini computer) from the redisnew image.

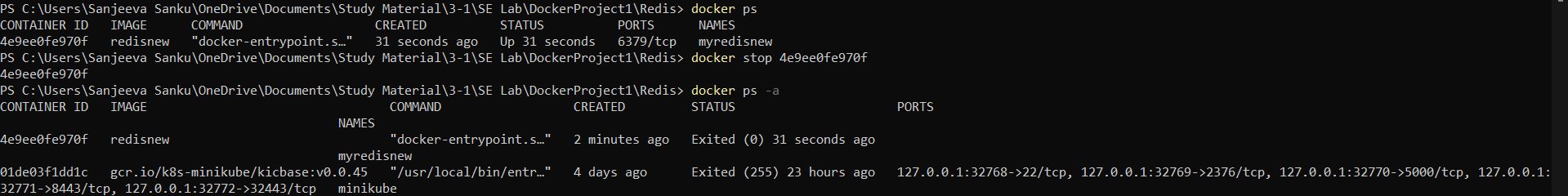
--name myredisnew: Names the container "myredisnew" so it’s easy to identify.

-d: Runs the container in the background.



docker stop myredisnew:

Stops the container named "myredisnew" (like turning off a computer)

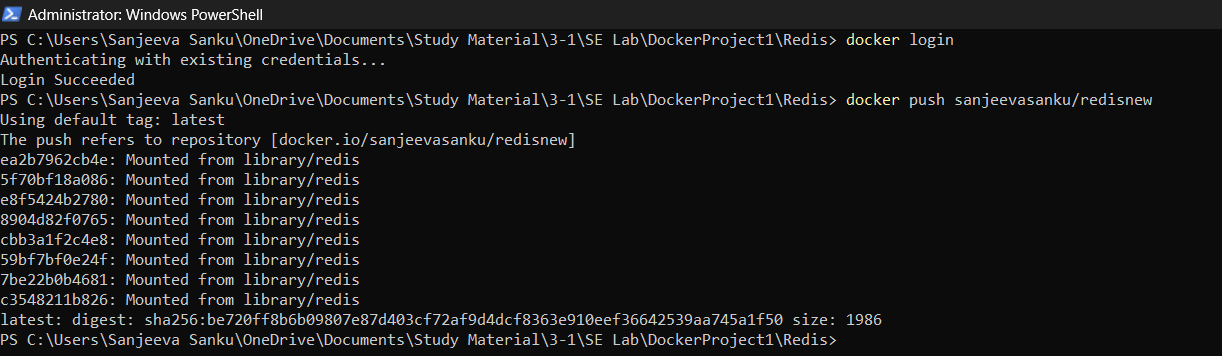


docker login:

Logs you into your Docker Hub account, so you can upload images.

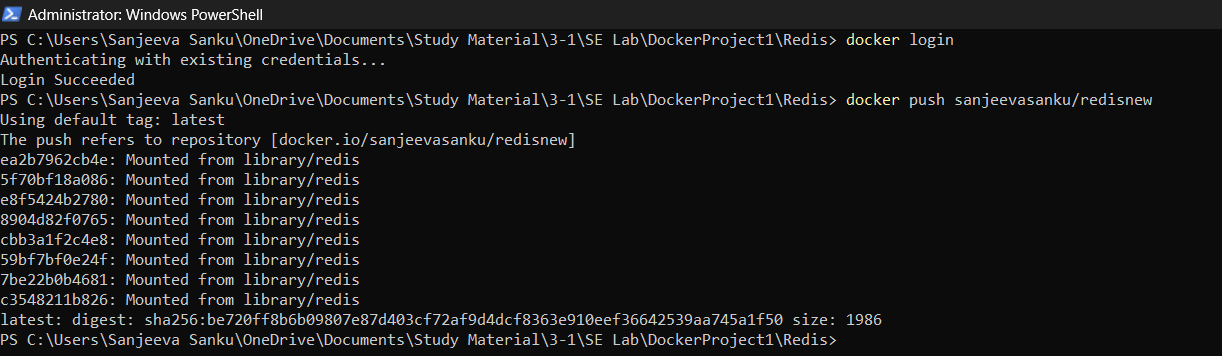
docker commit [id]:

Takes a snapshot (saves changes) of the container with ID and creates a new image called sanjeevasanku/redisnew.



docker push sanjeevasanku/redisnew:

Uploads the image sanjeevasanku/redisnew to Docker Hub, so others can download it.

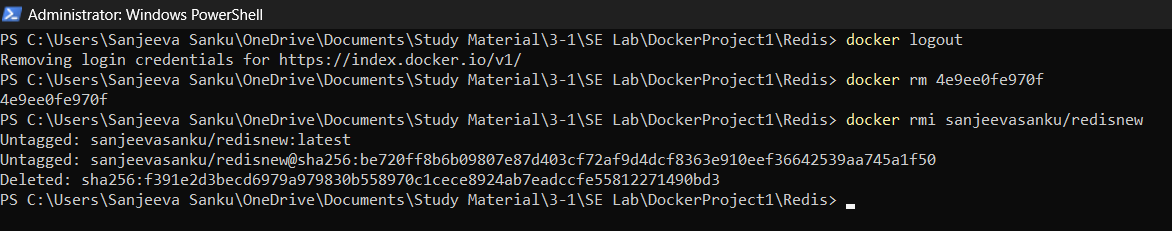


docker rm [id]:

Deletes the container with ID

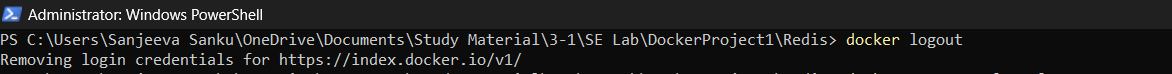
docker rmi sanjeevasanku/redisnew:

Deletes the image sanjeevasanku/redisnew from your system.



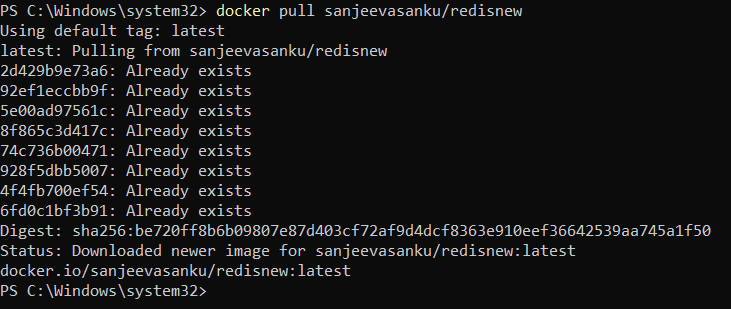
docker logout:

Logs you out of Docker Hub



docker pull sanjeevasanku/redisnew:

Downloads the image sanjeevasanku/redisnew from Docker Hub.



docker run --name myredis -d sanjeevasanku/redisnew:

Starts a new container using the image sanjeevasanku/redisnew



docker exec -it myredis redis-cli:

Opens the Redis command-line interface (like a terminal) inside the running container myredis

SET name "Abcdef":

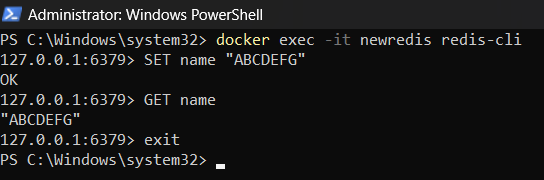
aves a key-value pair in Redis (key = name, value = Abcdef).

GET name:

Retrieves the value of the key name from Redis (it will return "Abcdef").

Exit:

Exits the Redis CLI.



docker images:

Lists all images again to confirm which ones remain.

