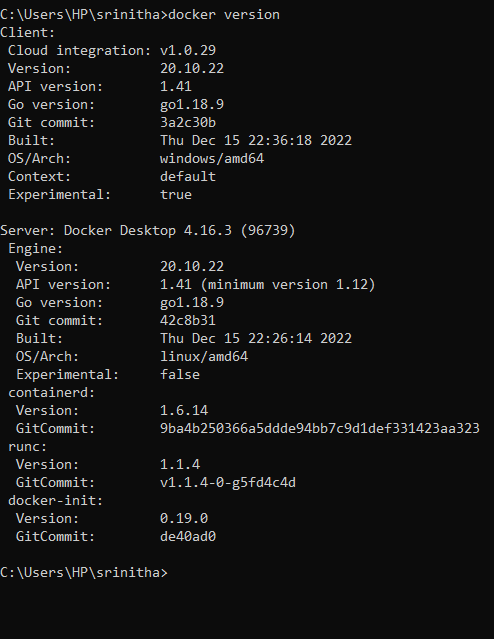
**Q1) Pull any image from the docker hub, create its container, and execute it showing the output.**

Docker is a set of platforms as a service products that use of operating system level visualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries, and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than a virtual machine.

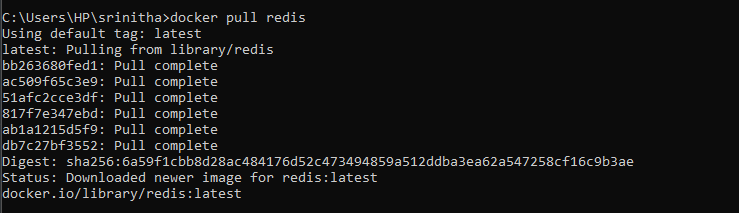
Docker version command:



Step 1:

We can pull the image from the docker hub using docker pull command

Lets us download a image named redis from the docker hub. Later we get this output.



Step 2:

We can create a new redis image from the downloaded image and expose it on port 90 using the following command.

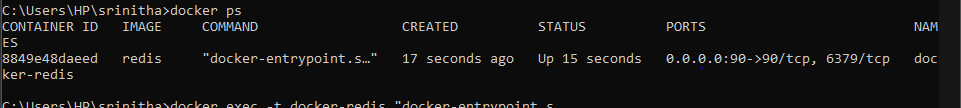
docker run --name docker-redis –p 90:90 –d redis



This is the executed output

Give docker ps command to get the recent downloaded image and its image id.

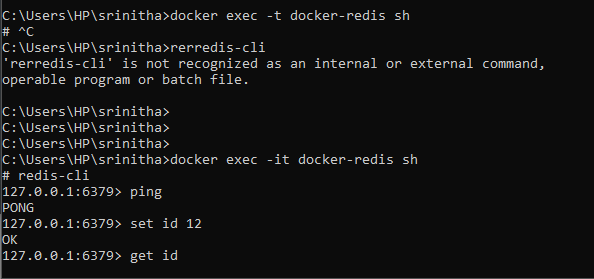
If we give docker ps –all then it show all the available docker images.



Now connect to the running container with command “docker run <image-name or image-id>”.

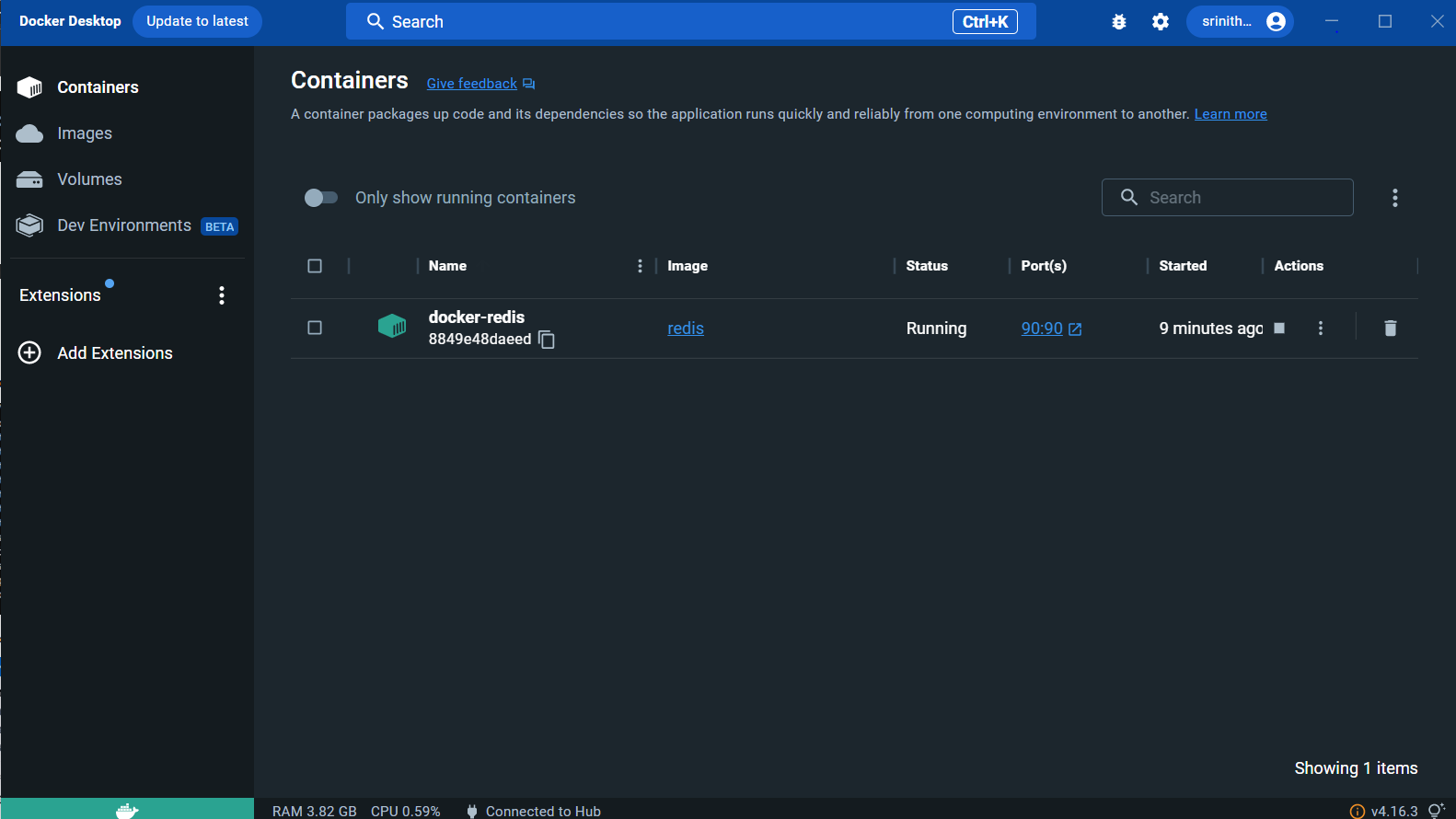
Or “docker exec –it docker-redis sh”

It results in the following output.



Now we are connected to the running container.

Open docker desktop to view the container is created or not.



**Q2) Create the basic java application, generate its image with necessary files, and execute it with docker.**

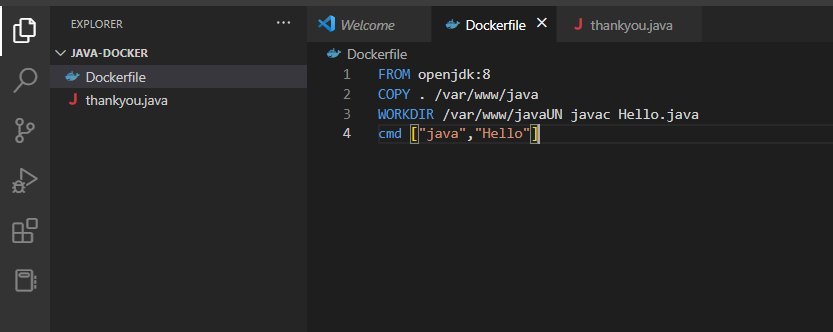
Creating the basic java application.

Step1:

Create a directory it is used to store the files.

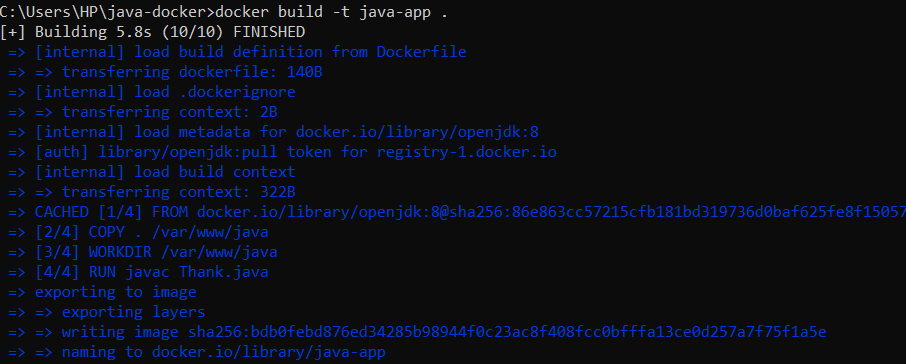


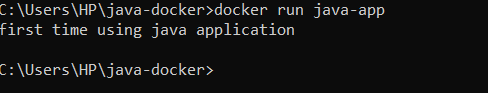


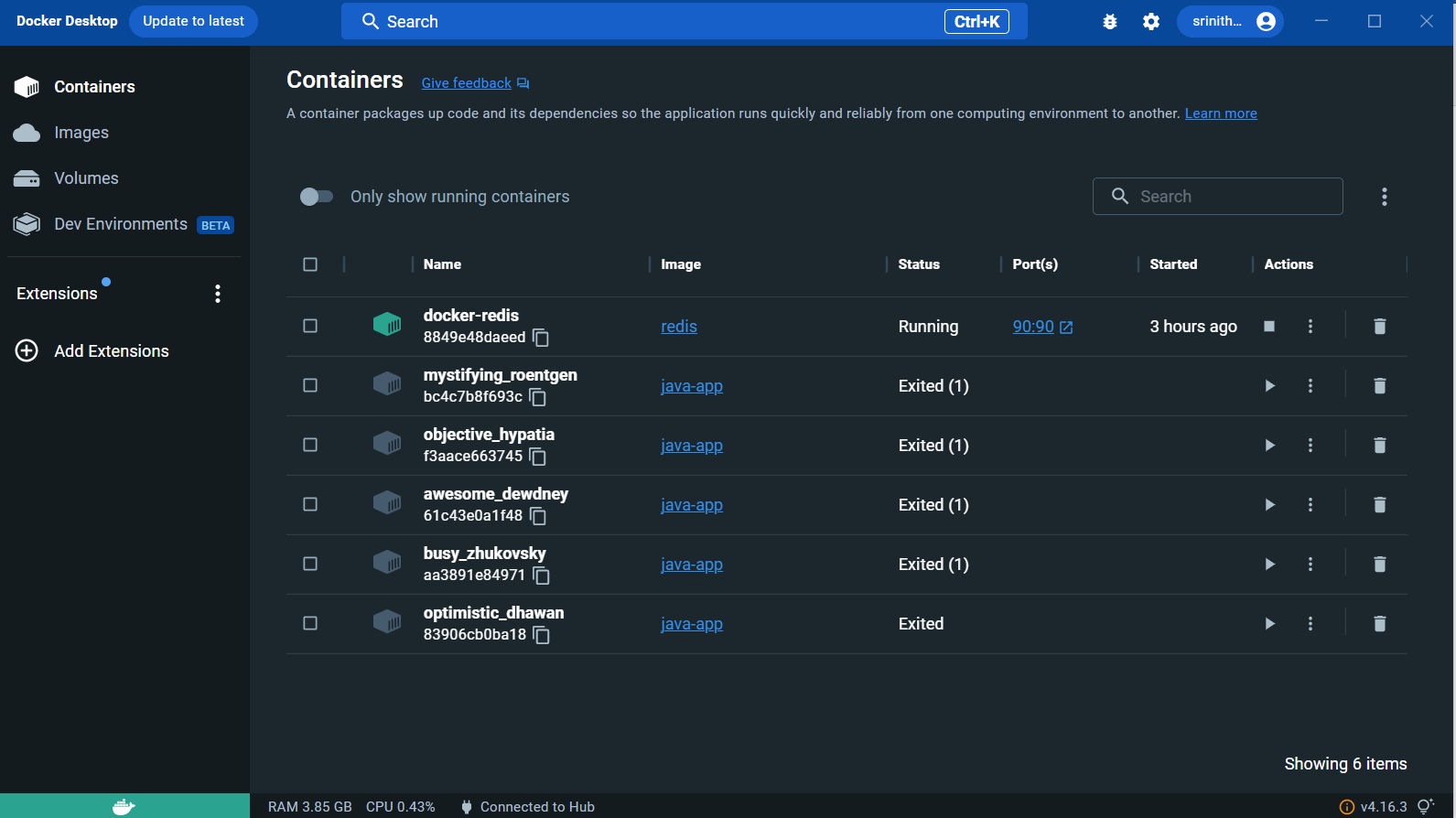


Create a docker file.

Create a java file, save it as Hello.java







https://github.com/srinitha062/DevOps\_aasignment.git