Docker

Make sure the virtualization should enable while installing the docker desktop

Once the installation was done. Then open the command prompt and type docker to make sure whether the docker is installed or not in our local system





Docker Version



Open docker hub







Run the container



Check whether container is working or not



Accessing the IP



Stop the docker container



Digits in the command are container id



Docker Desktop



Remove the docker image



Docker Desktop



Remove the container manually if you want.

Hello-world container



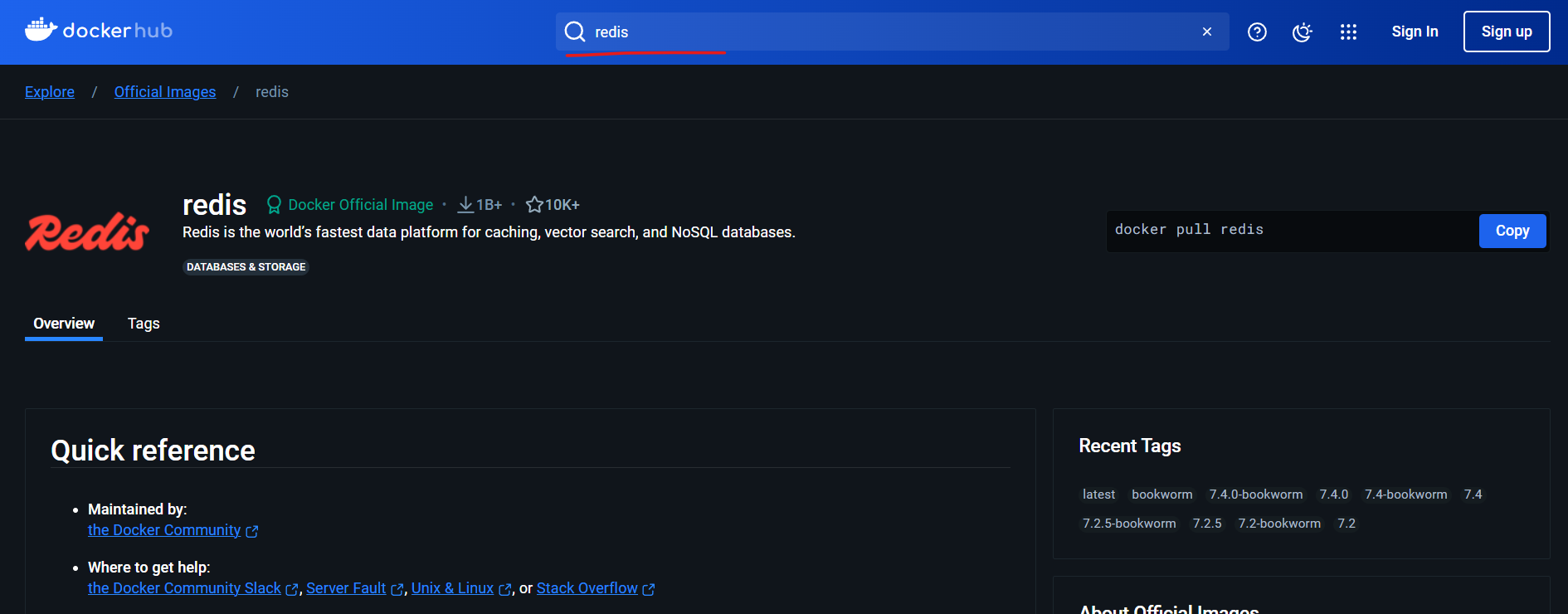
Pull the image

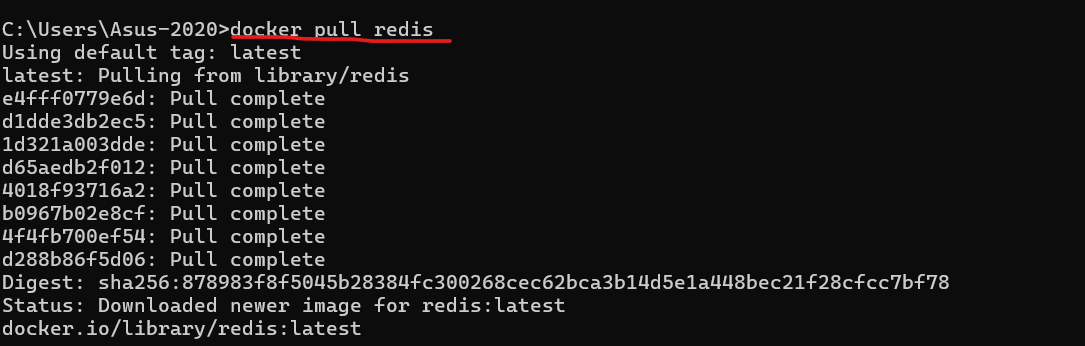


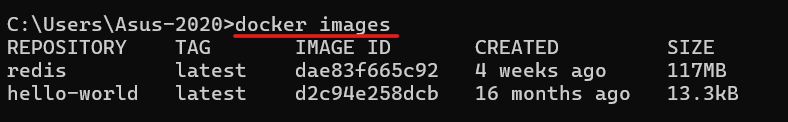
Run the docker

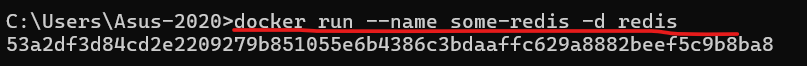


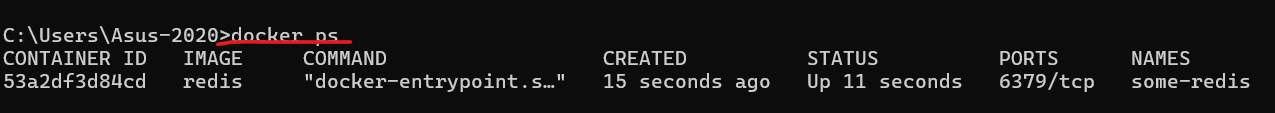
Redis image

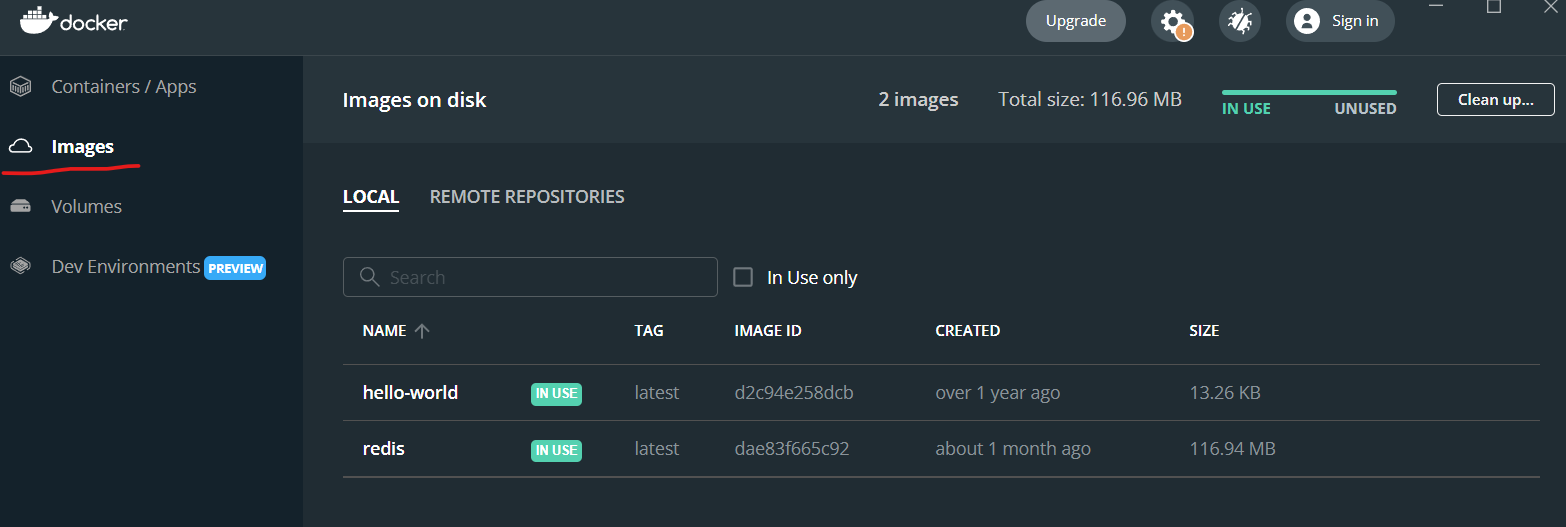


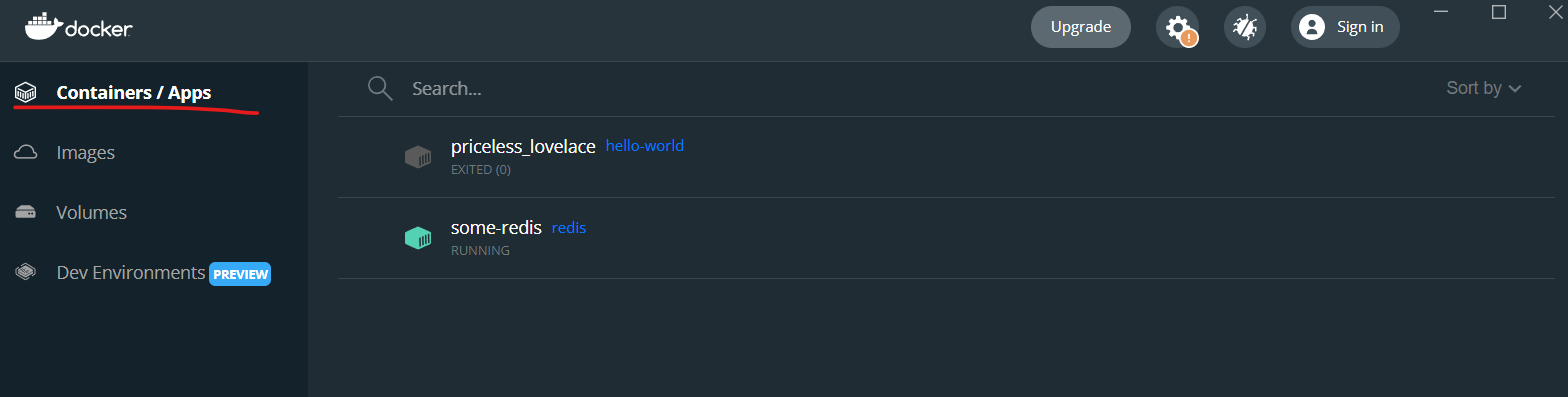






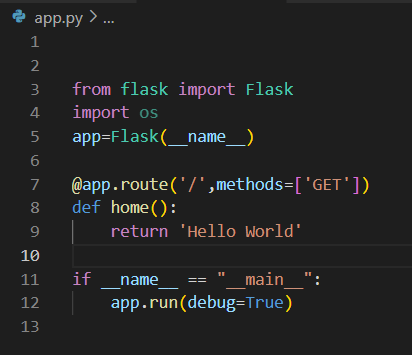






Creating Docker image

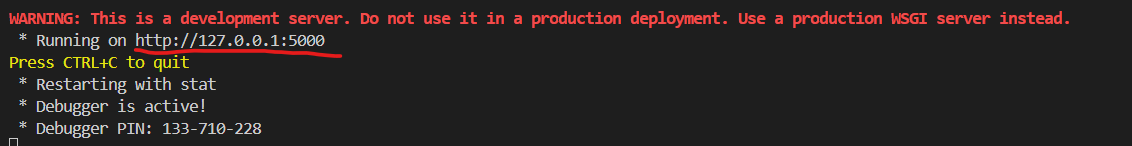
Create a simple hello world flask application

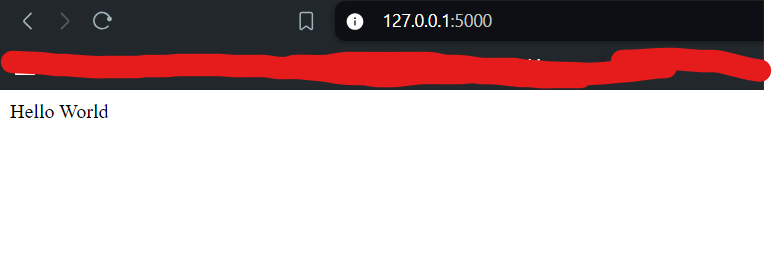




Run the app.py file

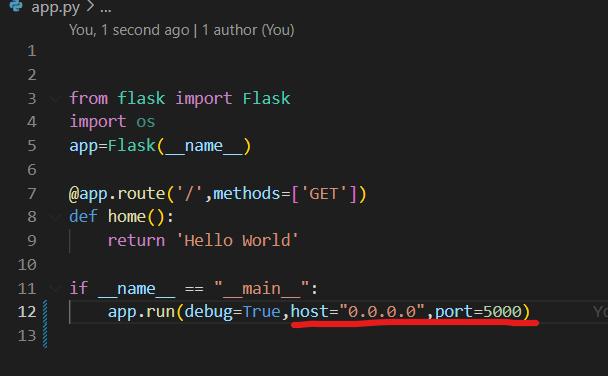






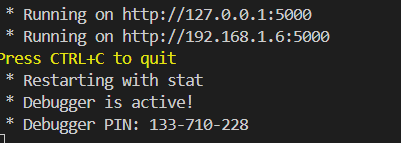
Dockerize the application

One change we need to do



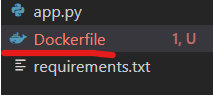
Host address = with the help of host ip and local address also we can access the application

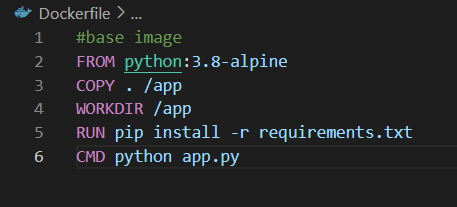
Again run the app.py file



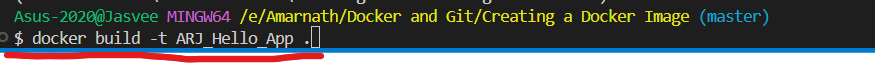
Then it will work as expected

We need to create a new file for the docker

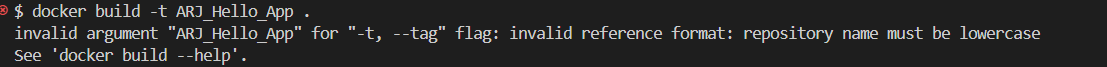




Now build the docker file

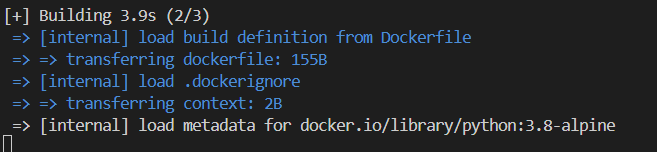


Error





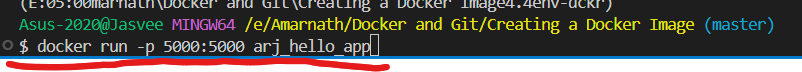
Build has started

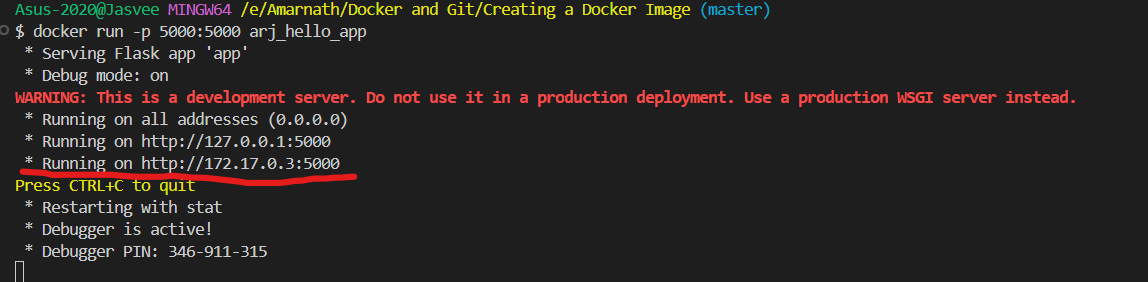


Check whether we have the image or not



Now run the specific docker image as a container

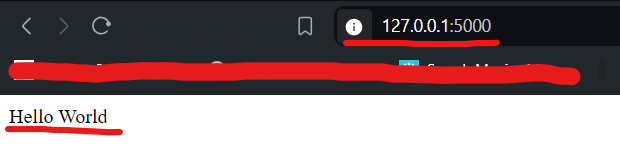




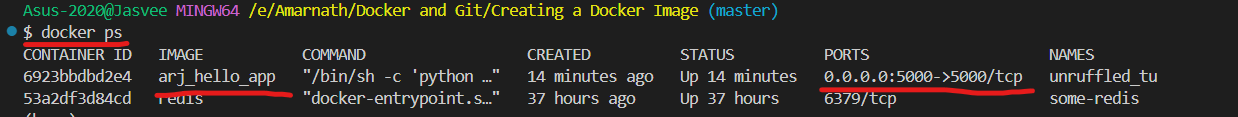
Here we can see two ip’s

If I click the 172.170.3 the app will not run, because this ip present inside my container and it runs in the 500 port, so obviously we cannot access it but we access it from our host container through a specific port.

But if I execute the ip 127.0.0.1:5000 then it will work and works well also with local host 0:0:0:0 with port number 5000

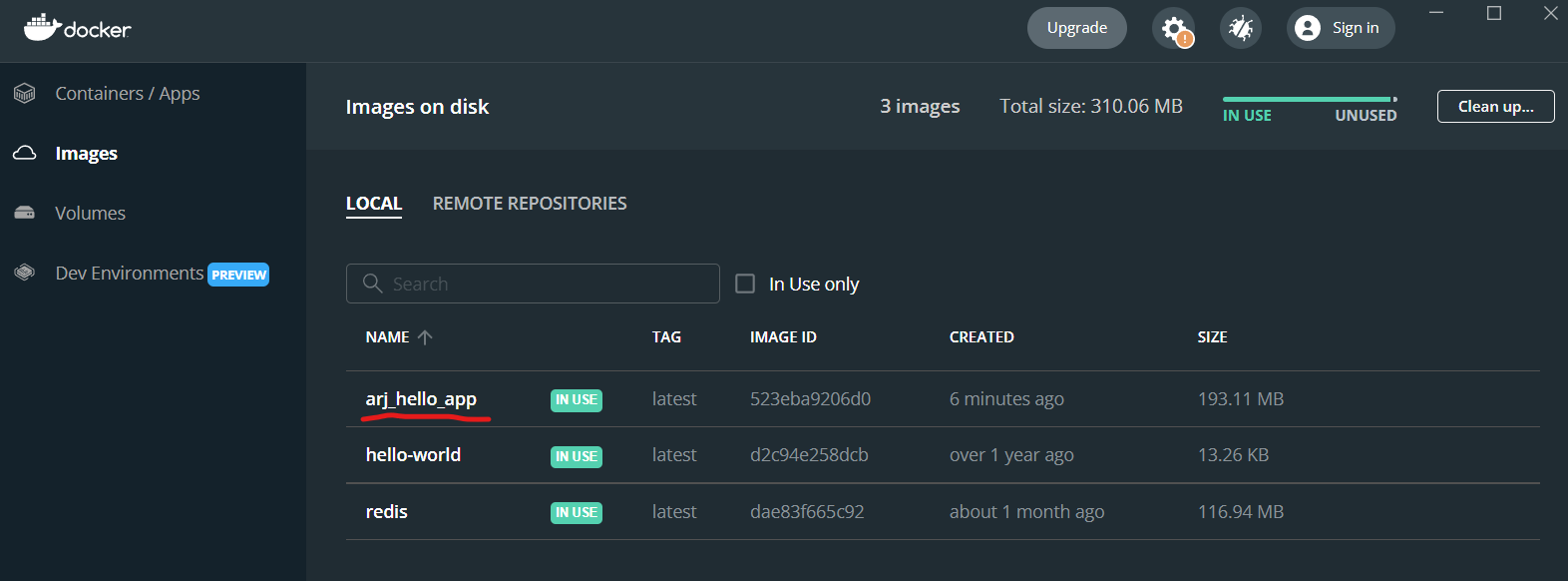


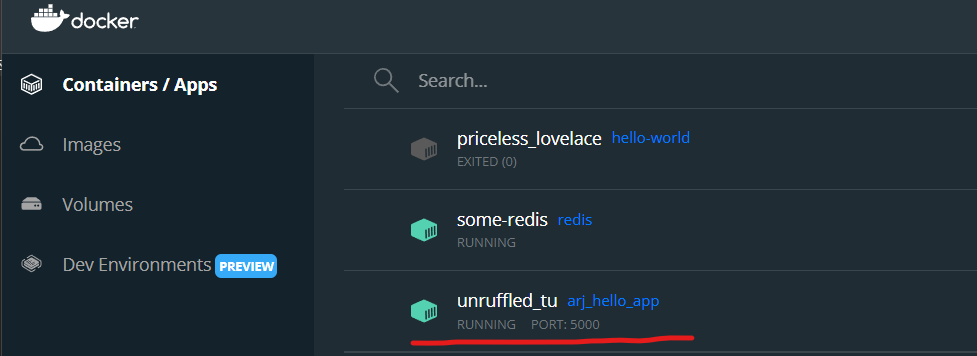
Docker ps



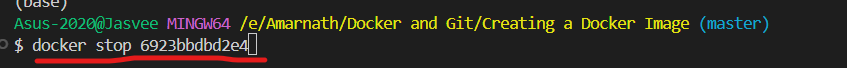
The specific docker container is running, so we can access the container through or local host by using the specific port

Docker Desktop





Stop the container

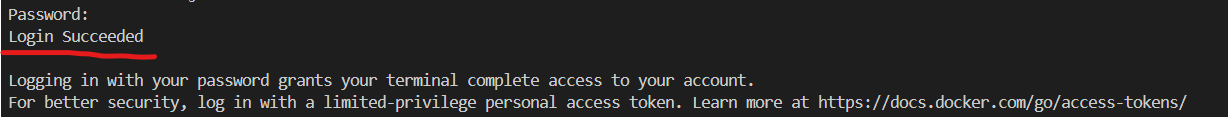


Push the docker image that we have created into the docker hub repository

Make sure you have to login into your docker hub account

Docker login:





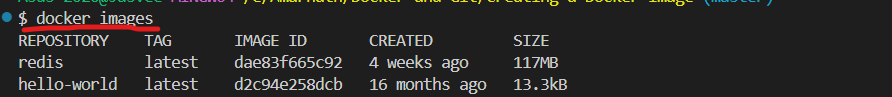
Rename the docker image

Two ways

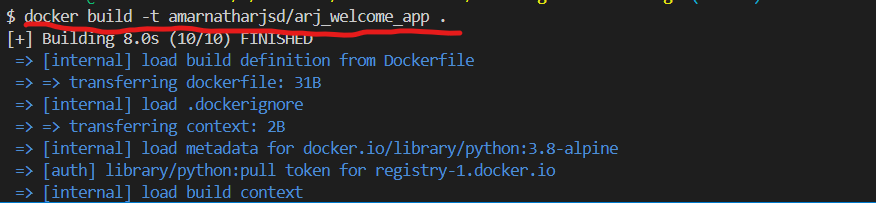
1. Remove and build a new one
2. Docker Tag

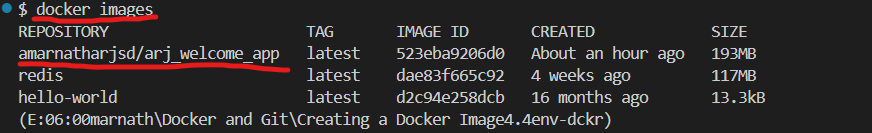
Method 1:





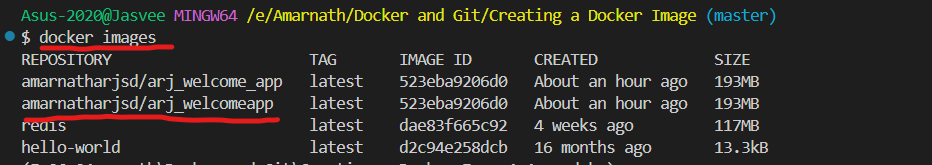
Build the docker image



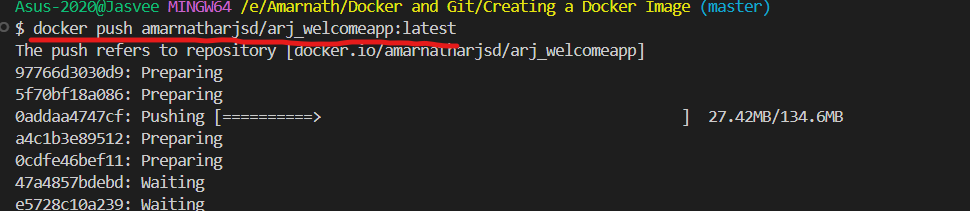


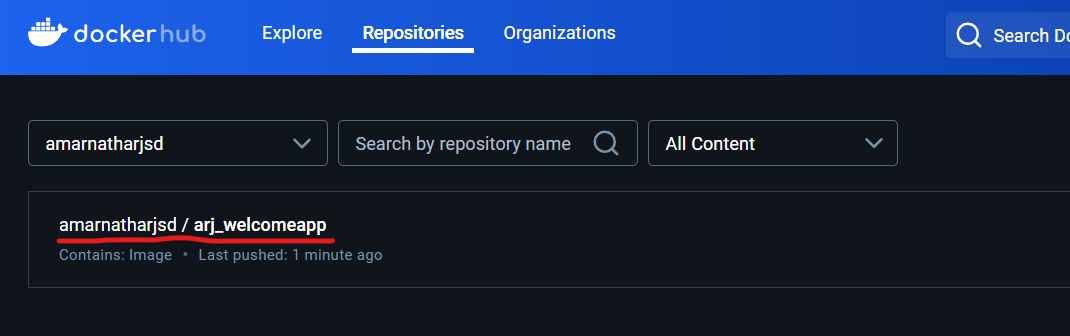
Method 2:

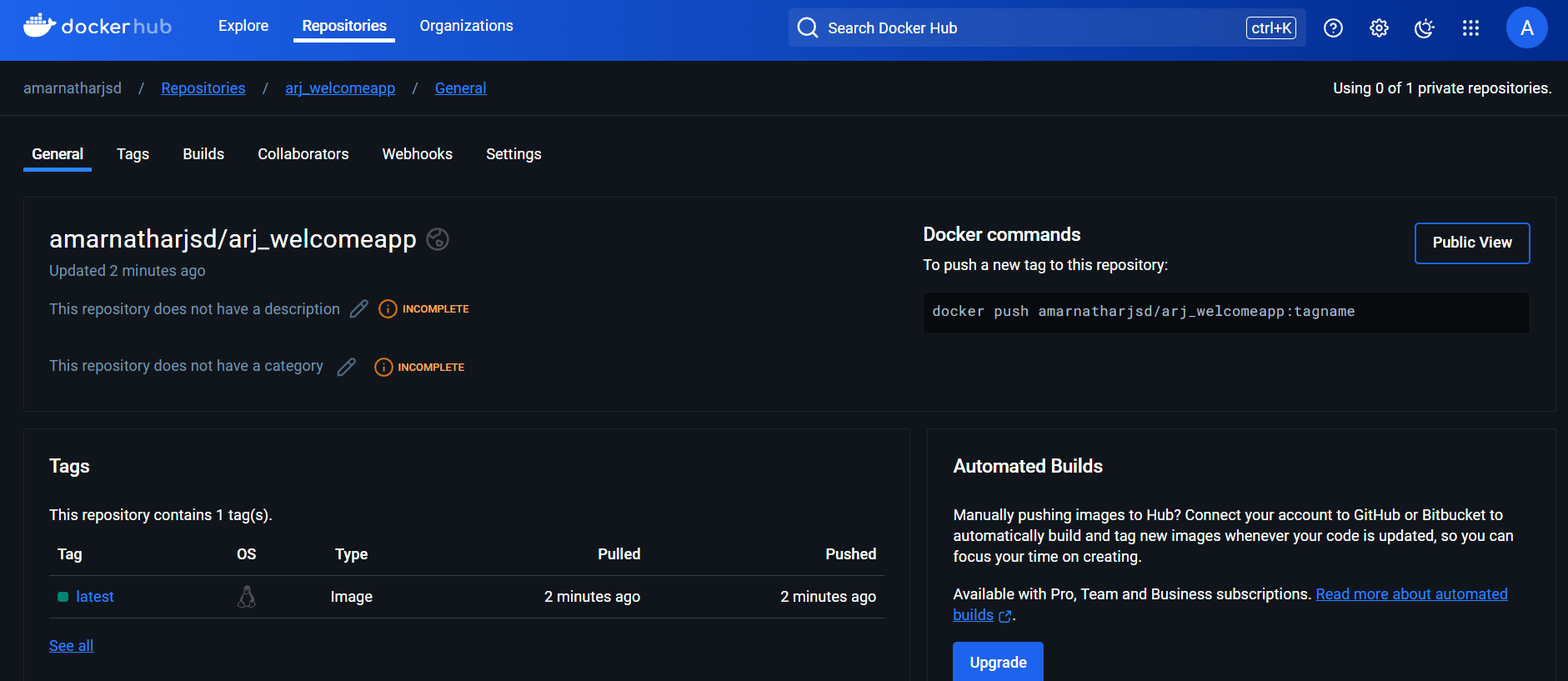




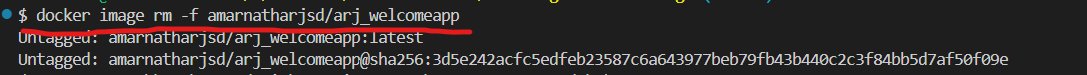
Push the docker image







Lets remove the image and again pull from the docker hub repository

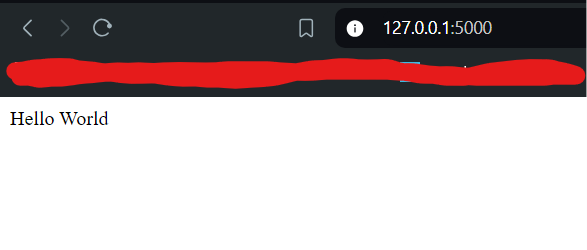


Build



Run



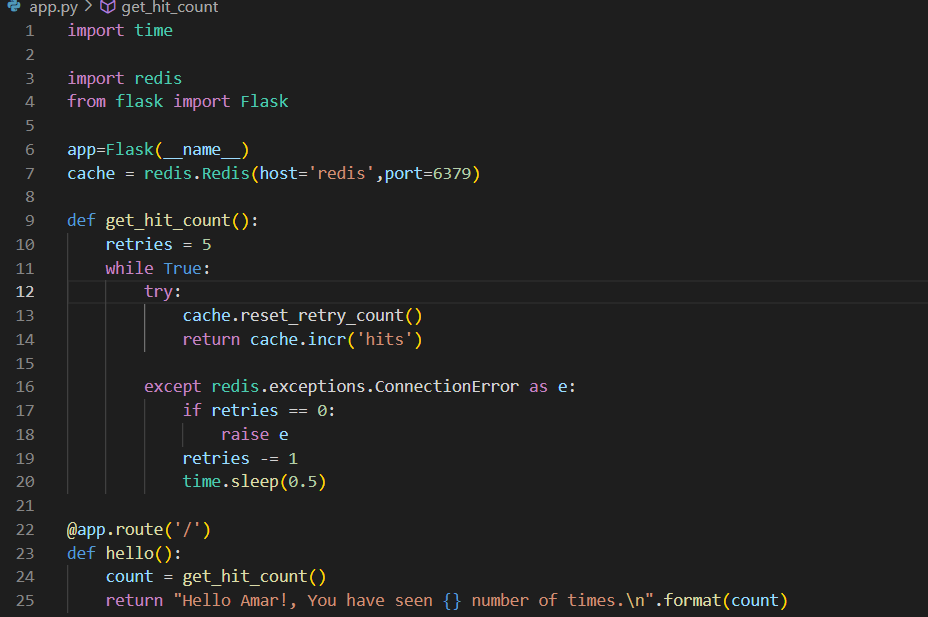


Docker compose

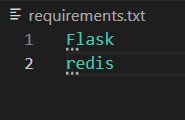
It is tool for defining and running multi containers in a docker applications

# redis present as an image in the docker hub

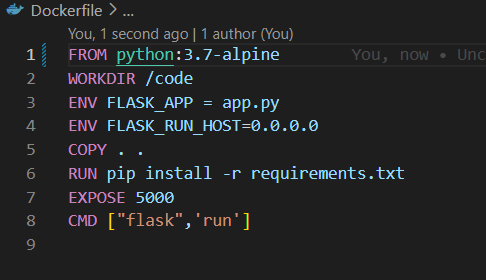
App.py



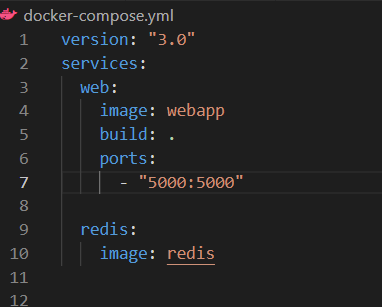
Requirements.txt



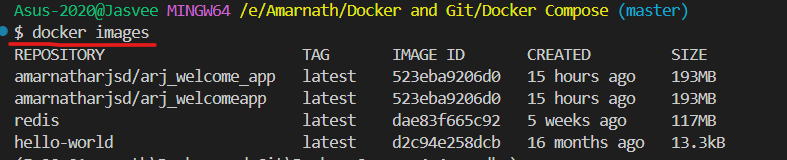
Dockerfile



Docker-compose.yml



Run the docker compose



Run docker compose

