Docker & Docker Hub

[Thangarajerode@gmail.com](mailto:Thangarajerode@gmail.com)

Environment: **Windows 7**

Application: **Docker Toolbox**

Download link **:** https://github.com/docker-archive/toolbox/releases/tag/v19.03.1

Default IP: http://192.168.99.100/

Pre\_request:

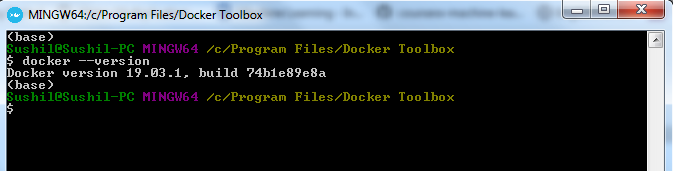
Docker hub username and password

**Assignment 1:**

**Demonstration of some of the important commands used in Docker.**

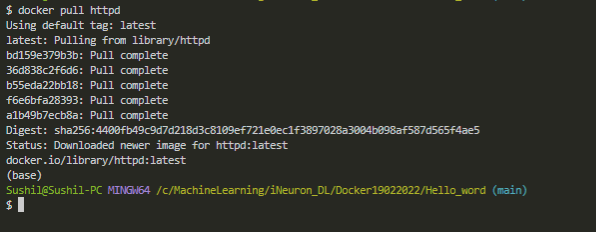
* **Finding the version**
* **docker –version**

This command is used to get the currently installed version of docker



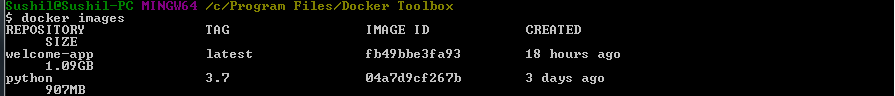
* **Downloading image**
* **docker pull httpd**

Let’s say you need to pull the docker image from [dockerhub](https://hub.docker.com/) (docker repository). The following example of pulling the Apache HTTP server image.



* **List images**
* **docker images**

List all the docker images pulled on the system with image details such as TAG/IMAGE ID/SIZE etc.



* **Run Docker**
* **docker run -it -d -p 5000:5000 welcome-app**

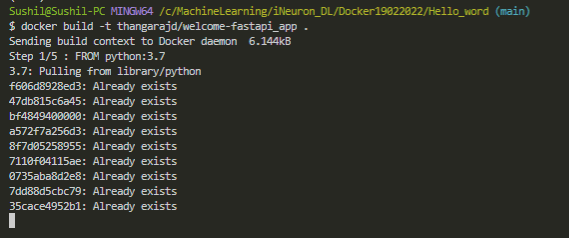
Docker run <Inteactive option> <detachable mode option> <Port option> <Host port number> : <docker port> <Image name>

Run the docker image mentioned in the command.



* **Build the Container**

The docker build command builds Docker images from a Dockerfile and a “context”



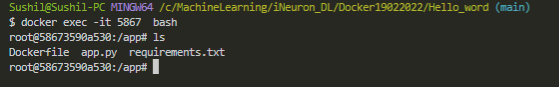
* **List all running/existed containers**
* **docker ps –a**

List all the docker containers running/exited/stopped with container details.

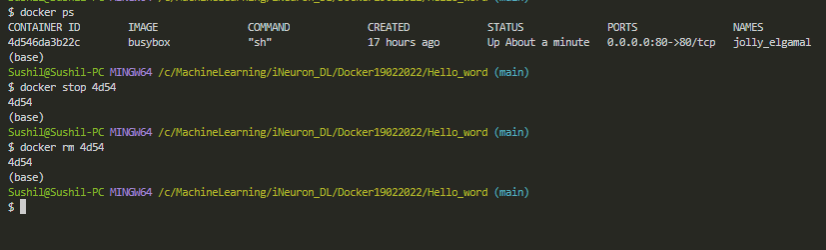


* **Access inside the container**
* **docker exec -it <Container id> bash**

Access the docker container and run commands inside the container.

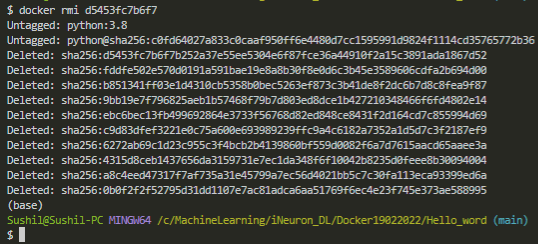


* **Remove the Container**
* **docker rm <<Container id>**



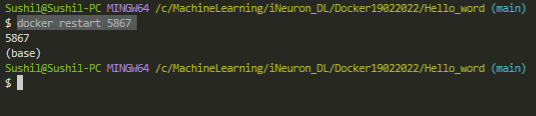
* **Remove the image**
* **docker rmi <image id>**

Remove the docker image with the docker image id mentioned in the command



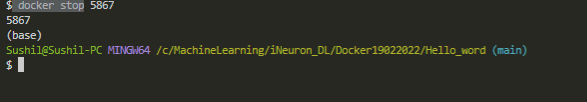
* **Restart the container**
* **docker restart <Container id>**

Restart the docker container with container id mentioned in the command.



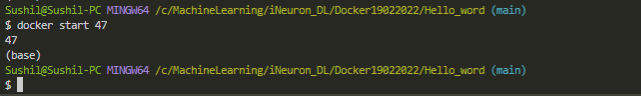
* **Stopping the docker**
* **docker stop <Container id>**

Stop a container with container id mentioned in the command



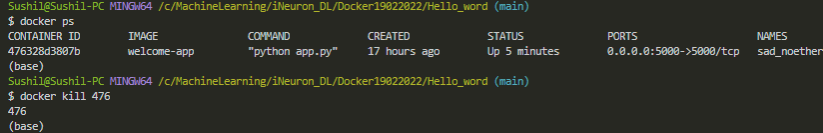
* **Start the docker**
* **docker start <Container id>**

This command in docker starts the docker container with container id



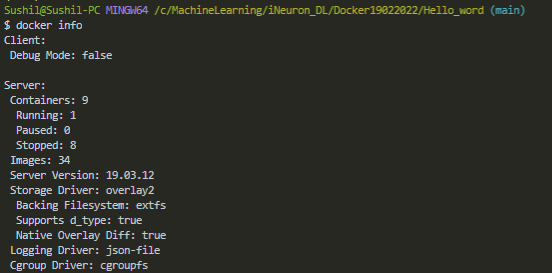
* Docker kill
* **docker kill <Container id>**

Stop the docker container immediately. Docker stop command stops the container gracefully, that’s the difference between a kill and stop commands.



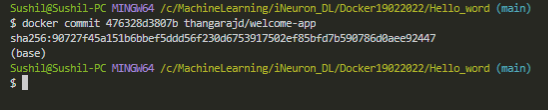
* **Docker information**
* **docker info**

Get detailed information about docker installed on the system including the kernel version, number of containers and images, etc.



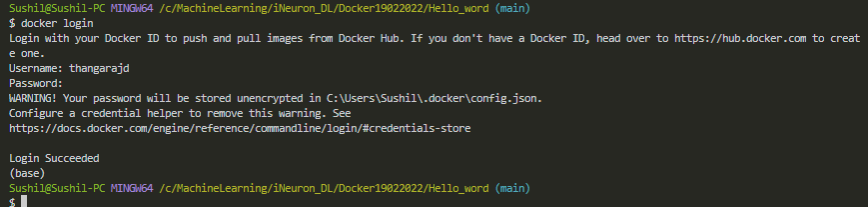
* **Commit**
* **docker commit <container id > <username>/<imagename>**

Save a new docker image with container id mentioned in the command on the local system. In the example below, thangarjd is the username, and welcome-app is the image name.



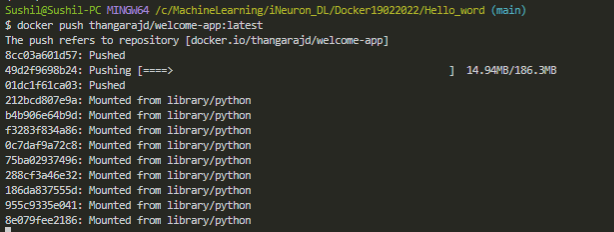
* **Login**
* **docker login**

Login into docker hub. You will be asked your docker hub credentials to log in.

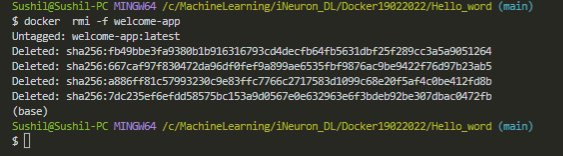


* **Push**
* **docker push <username>/<imagename>**

Upload a docker image with the image name mentioned in the command on the dockerhub.



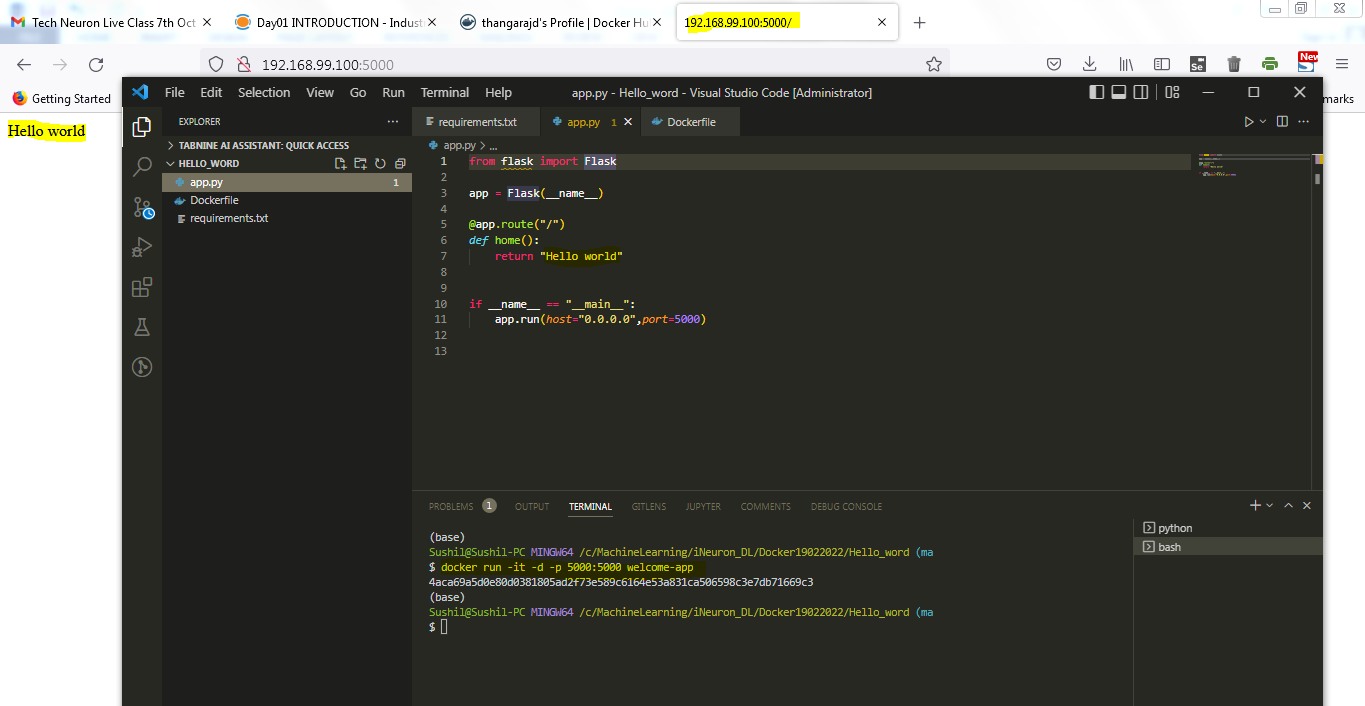
* **Remove image Forcefully**



**Assignment 2**

**Run Hello World Docker Image Locally.**

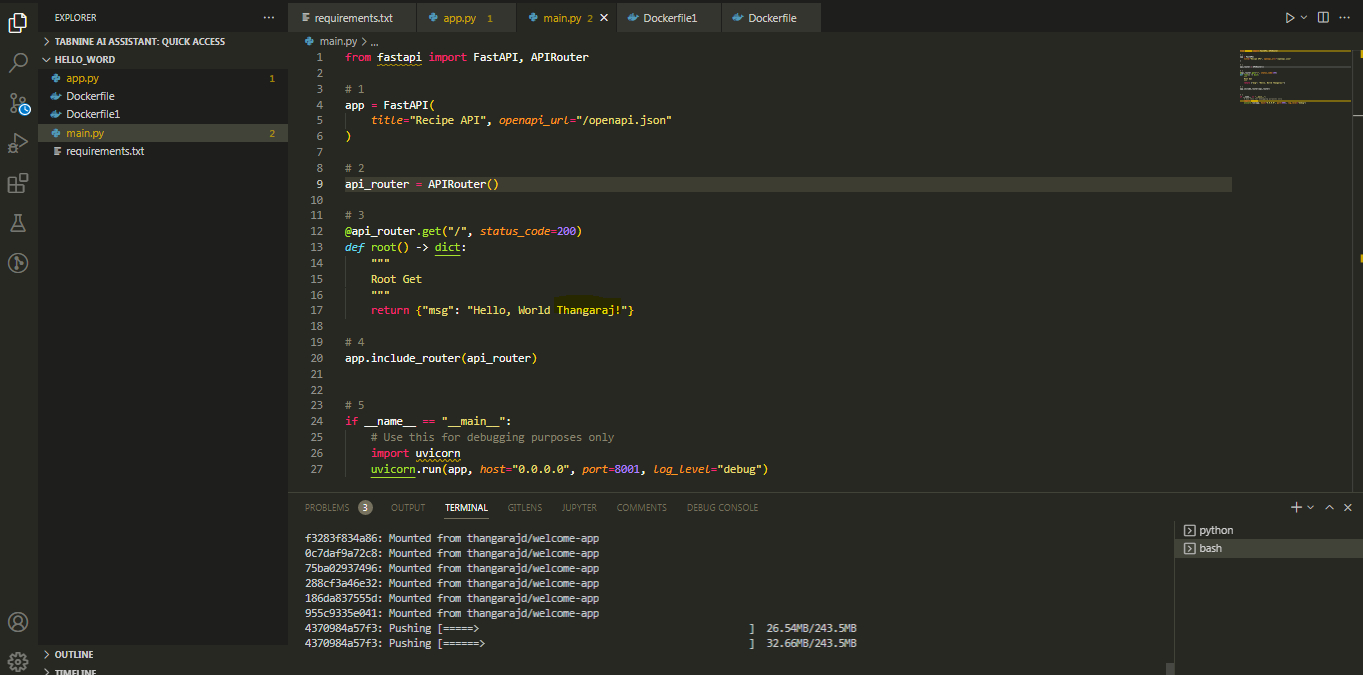
**FLASK API:**



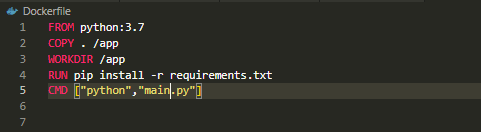
**Assignment 3**

Goal:

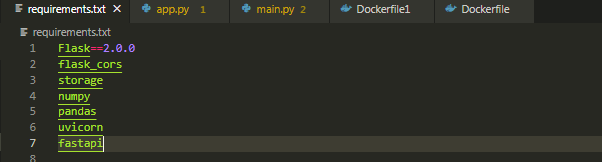
Create a hello world fastapi application.  
Create a Dockerfile for your fastapi hello world application.  
Build Docker image using Docker file.  
Run docker image build in previous step.  
Push your Docker image to Docker Hub.

1.main.py(Fastapi)

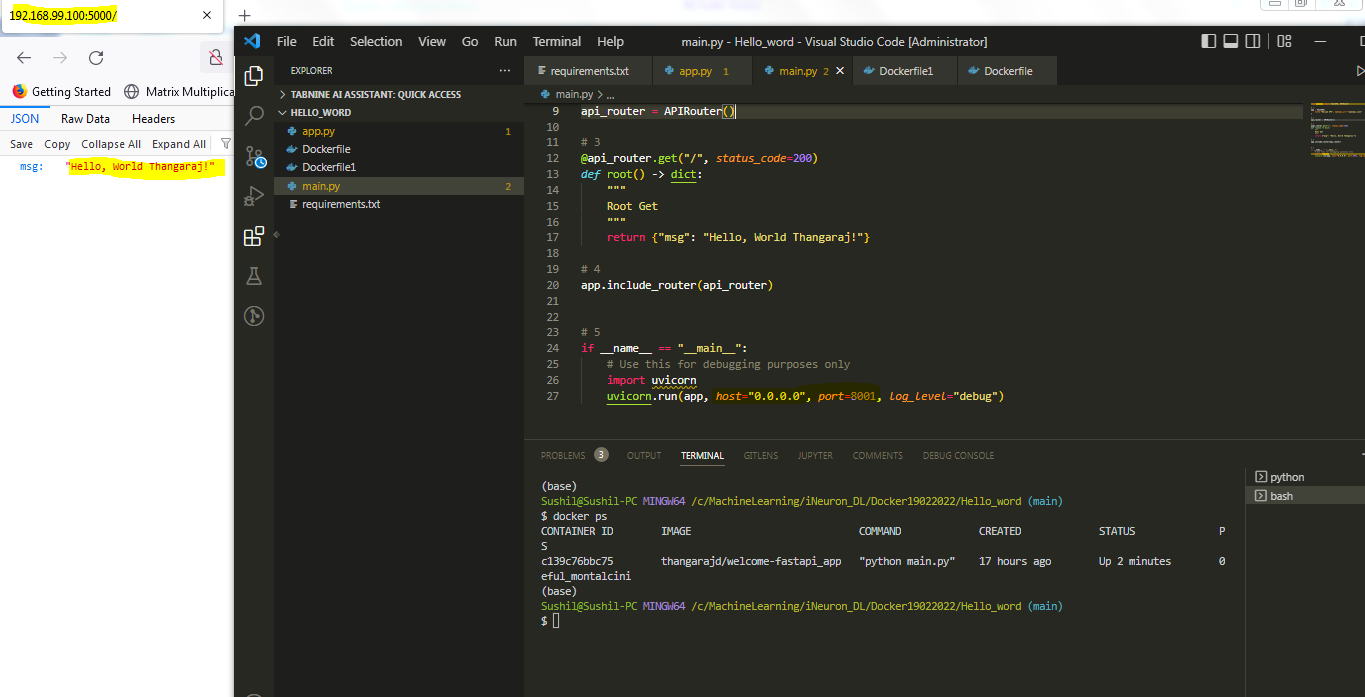
2.Docker file.

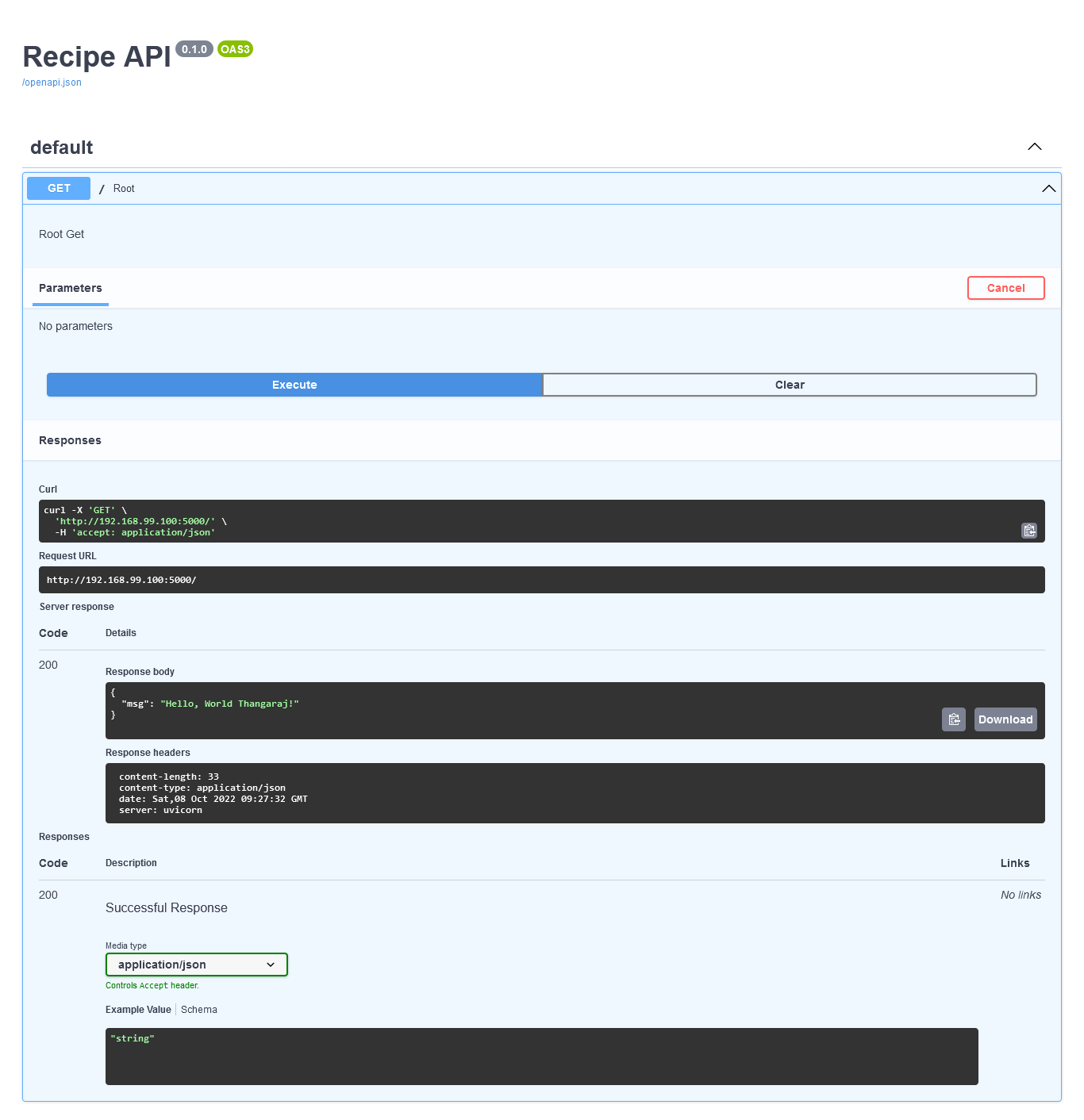


3.Requirments.text

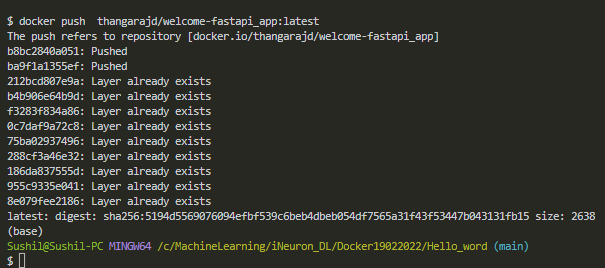


4. Local execution

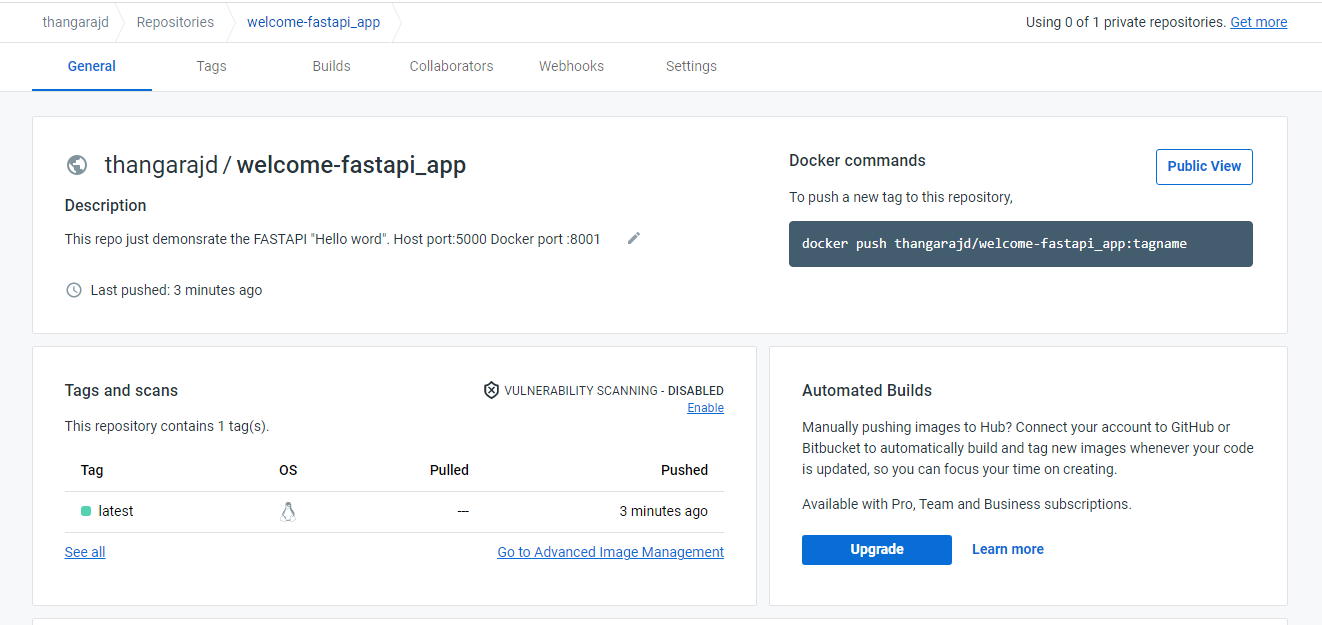




5.pushing image to docker hub



Docker hub:

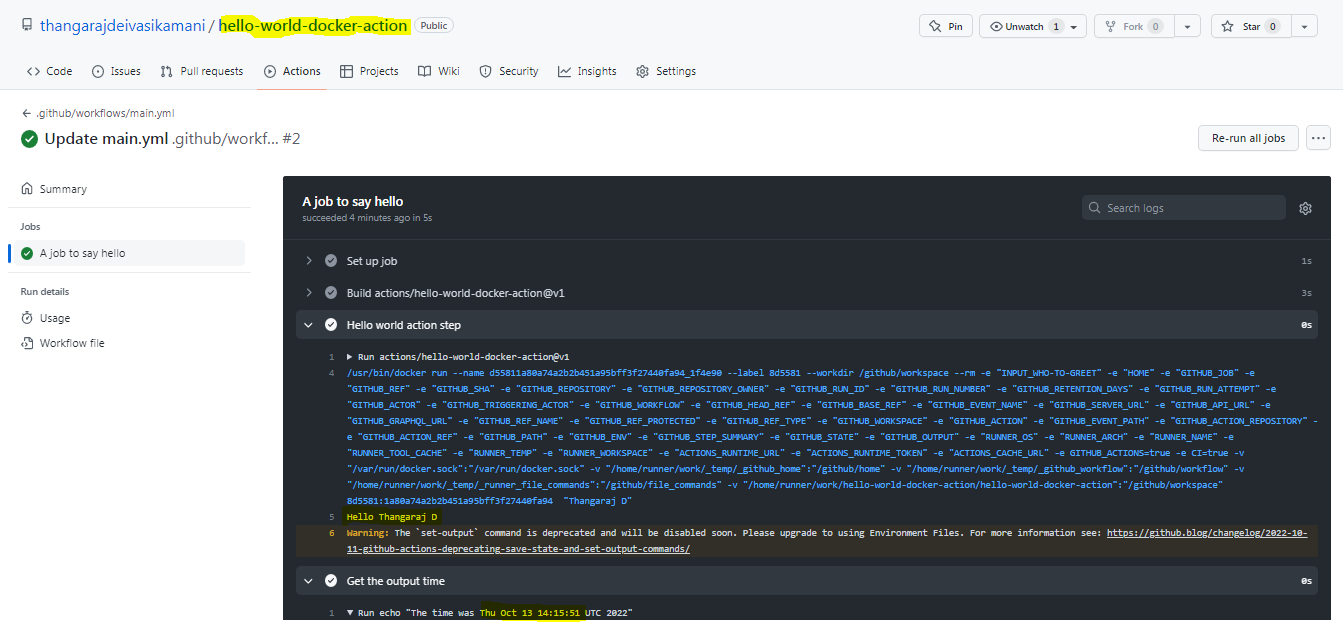


**Assignment 4:**

Goal:

Automate Assignment below task using github action. 1. Build Docker Image 2. Push Docker Image to Docker hub

Hello word:

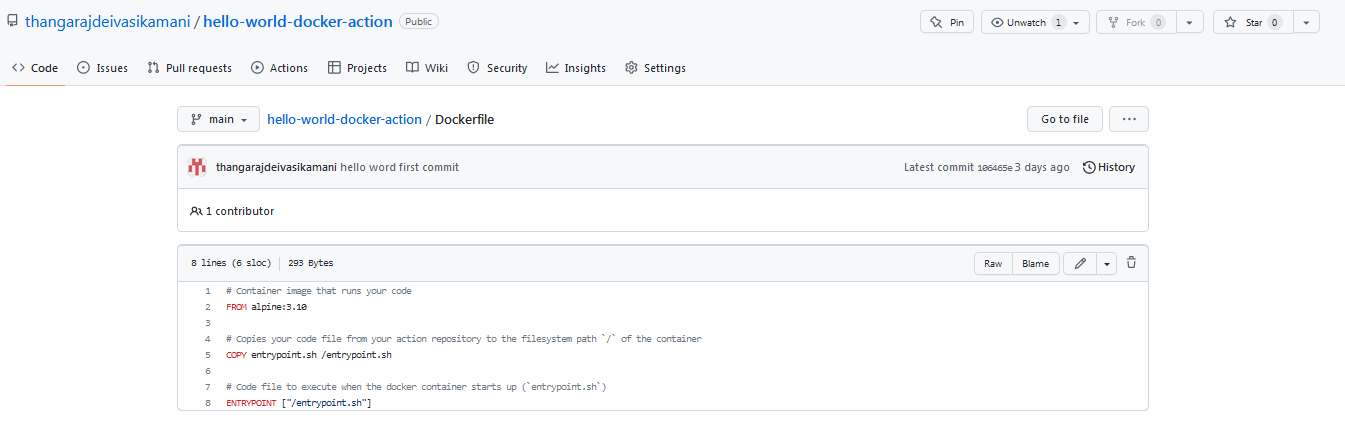


Git hub action:

Document reference:

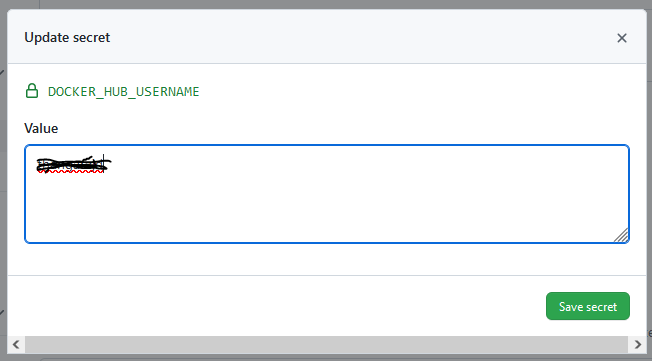
<https://docs.docker.com/ci-cd/github-actions/>

Repo:

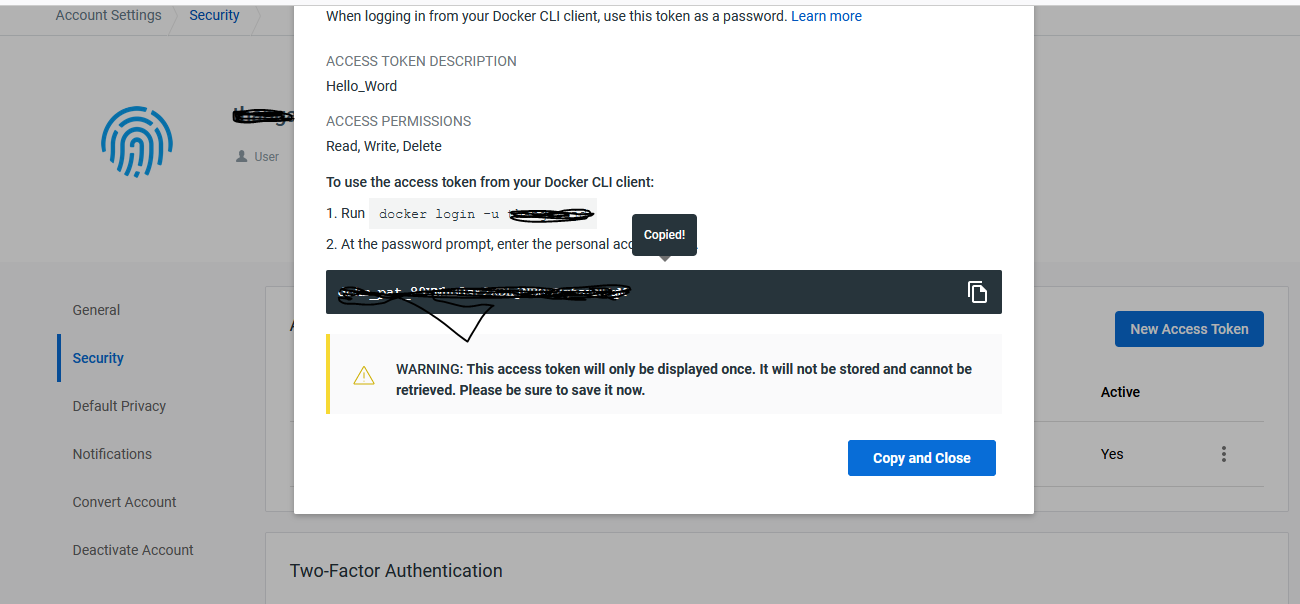


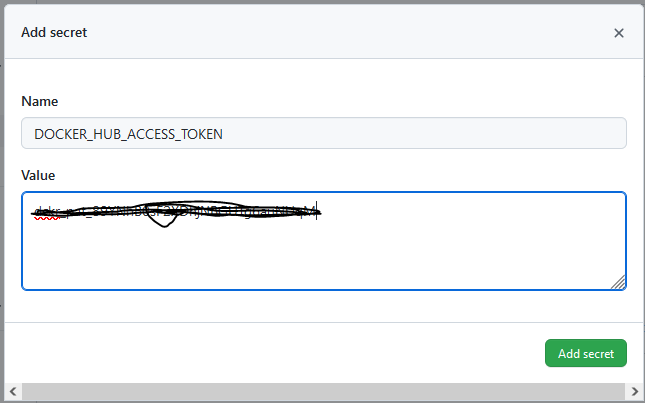


1.Create the Envoriment user name scerate in github – setting – secret – action – New repostitiry secret.

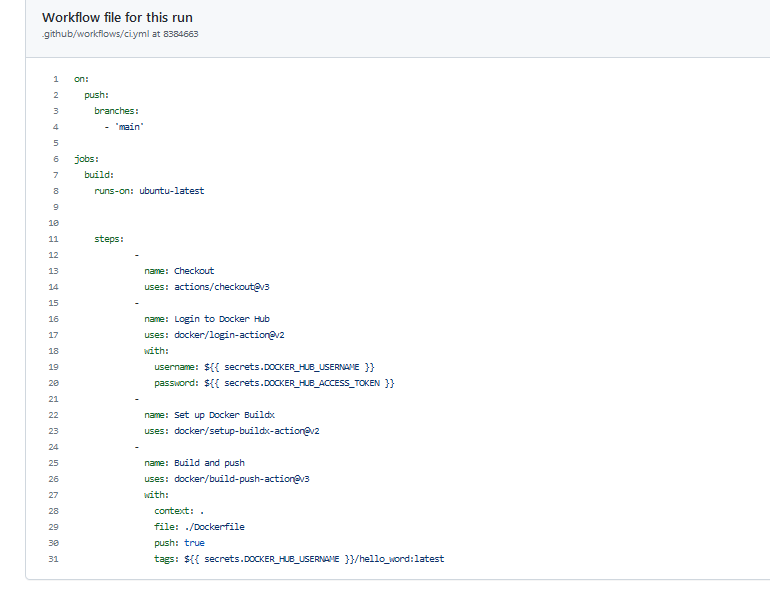


2. Create the token in docker

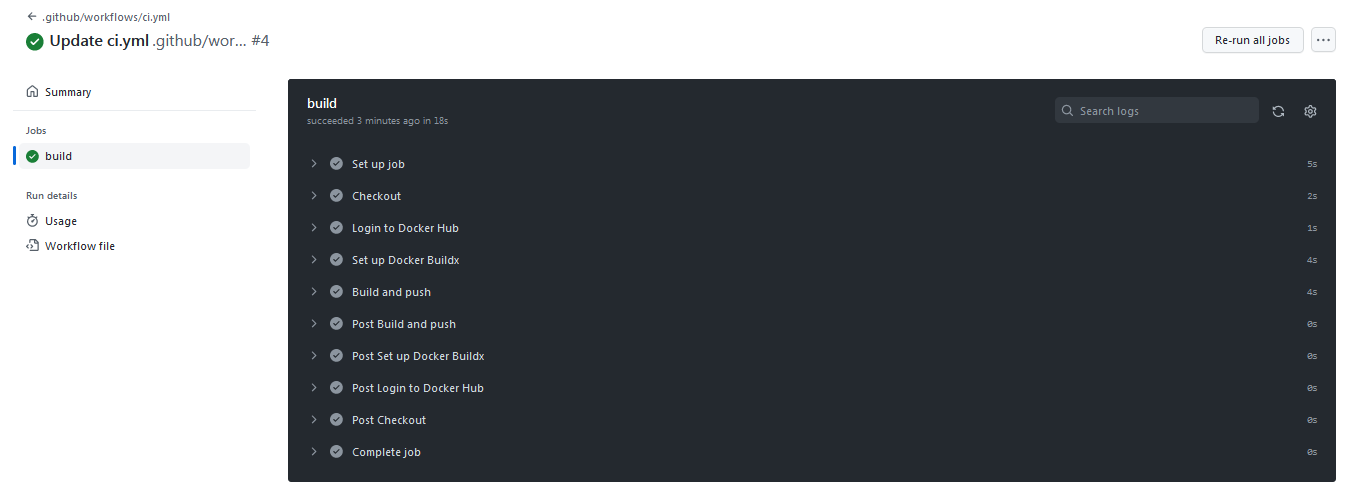




4. Add the ci.yaml file



5. run the workflow if you want manually.



6. Check in Docker hub repo whether the hello word auto pushed.

