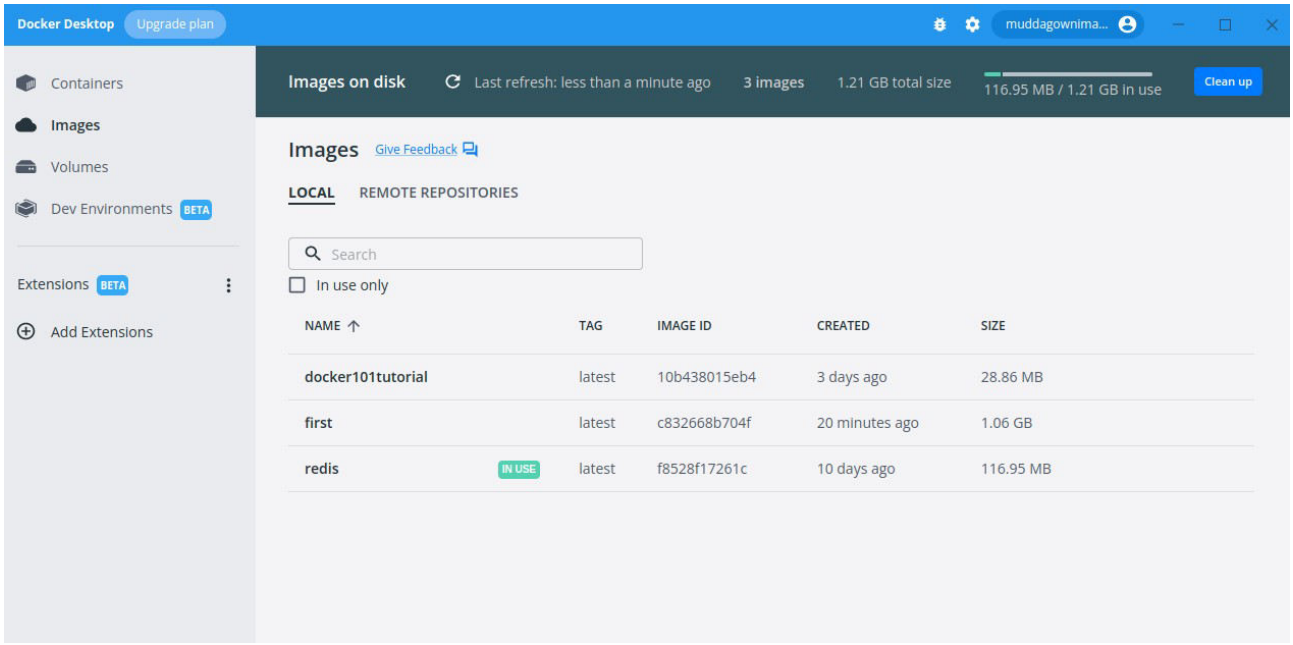
Docker Assignment:

1. docker build . -t helloworld: To build an image we need to run the above command, without creating images we cannot create container.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker build . -t first
[+] Building 11.8s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 37B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3.7
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 146B
 => CACHED [1/4] FROM docker.io/library/python:3.7@sha256:51dcbb98ba807f3631366d5a0678bc6815ca71901ddab6ed2ed5319da79f1c4d
 => [2/4] COPY . /app
 => [3/4] WORKDIR /app
 => [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:c832668b704f7bd4de0a7e2cfb9a46a6c2693f3b5ddf62ee9c5179c5cef60461
 => => naming to docker.io/library/first

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
```



2. Docker run first : To start the container

3. Docker images: we can display the images that are present in our docker desktop. So here we can see the first image that I have created. It consists of all the files to run the docarised application. where you will have **base image** to start.

```
 => => transferring dockerfile: 37B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3.7
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 146B
 => CACHED [1/4] FROM docker.io/library/python:3.7@sha256:51dcbb98ba807f3631366d5a0678bc6815ca71901ddab6ed2ed5319da79f1c4d
 => [2/4] COPY . /app
 => [3/4] WORKDIR /app
 => [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:c832668b704f7bd4de0a7e2cfb9a46a6c2693f3b5ddf62ee9c5179c5cef60461
 => => naming to docker.io/library/first

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
first               latest    c832668b704f   56 seconds ago   1.06GB
docker101tutorial   latest    10b438015eb4   3 days ago       28.9MB
redis               latest    f8528f17261c   10 days ago      117MB
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
```

4. **Docker ps :** It show all the live containers that are running

5. docker ps -a : It shows all the containers that are stopped and running

6. **docker run pensive_blackburn :** To run the container

- docker ps command to see the live containers that are running

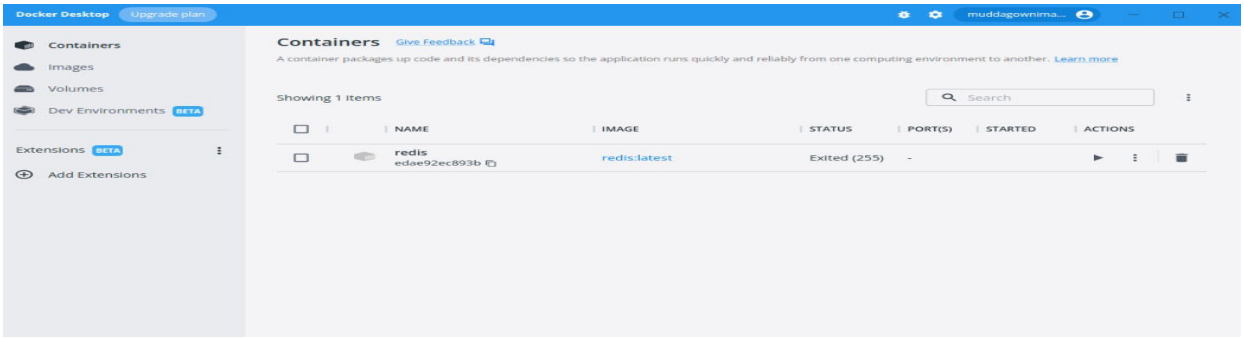
```
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker run pensive_blackburn
Unable to find image 'pensive_blackburn:latest' locally
docker: Error response from daemon: pull access denied for pensive_blackburn, repository does not exist or may require 'docker login': deni
See 'docker run --help'.
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps
CONTAINER ID   IMAGE     COMMAND           CREATED         STATUS         PORTS     NAMES
e430e8536431   first     "python app.py"   6 minutes ago   Up 9 seconds             pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
```

7. **docker stop pensive_blackburn:** To stop the live container

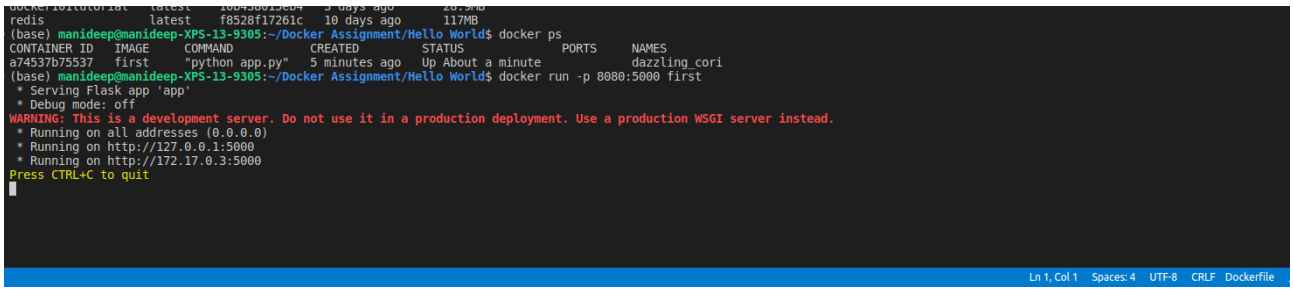
```
e430e8536431   first     "python app.py"   6 minutes ago   Up 9 seconds             pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ ^C
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker stop pensive_blackburn
pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker run pensive_blackburn
Unable to find image 'pensive_blackburn:latest' locally
docker: Error response from daemon: pull access denied for pensive_blackburn, repository does not exist or may require 'docker login': denie
See 'docker run --help'.
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps
CONTAINER ID   IMAGE     COMMAND           CREATED          STATUS         PORTS     NAMES
e430e8536431   first     "python app.py"   11 minutes ago   Up 8 seconds             pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
  edae92ec893b   redis     "docker-entrypoint.s…"   2 days ago       Exited (255) 35 minutes ago   6379/tcp   redis
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
```

8. docker rm pensive_blackburn: To remove the container. Before removing the container we need to stop it first and then remove

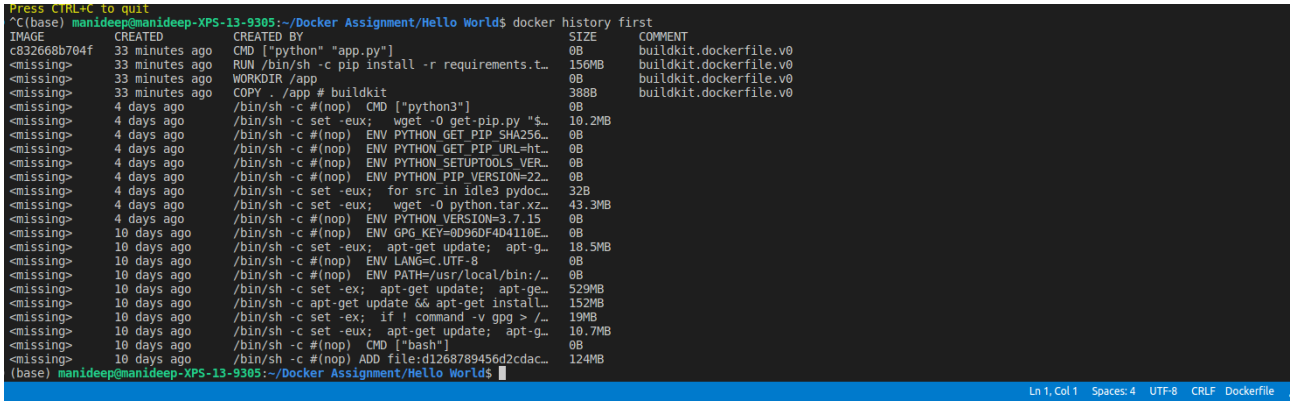
```
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rm pensive_blackburn
Error response from daemon: You cannot remove a running container e430e85364317e16f30f64c84992f20f9d8ba51bcbece38978e4ab55a65cfa41. Stop the container before attempting removal or for
emove
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker stop pensive_blackburn
pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rm pensive_blackburn
pensive_blackburn
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
first               latest    c832668b704f   20 minutes ago   1.06GB
docker101tutorial   latest    10b438015eb4   3 days ago       28.9MB
redis               latest    f8528f17261c   10 days ago      117MB
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS     NAMES
edae92ec893b   redis     "docker-entrypoint.s…"   2 days ago       Exited (255) 53 minutes ago   6379/tcp   redis
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
Ln 1, Col 1   Spaces: 4   UTF-8   CRLF   Dockerfile
```



9. **docker run p 8000:5000 first** : Changing port from 8000 to 5000. here first is my image name.

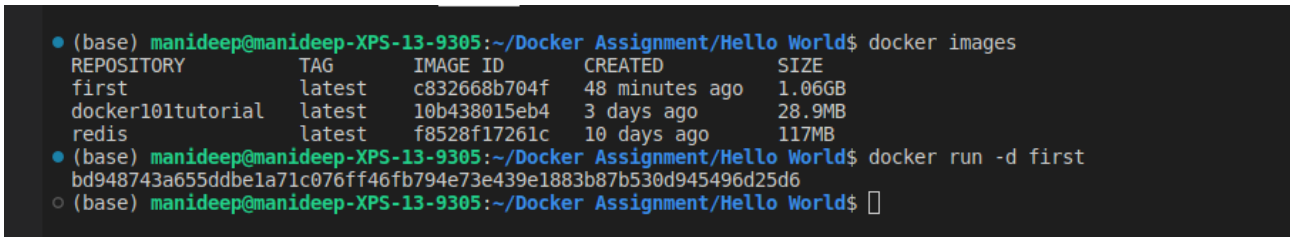


10. **docker history first**: It will show detailed information about the image



11. **docker run -d first:** our sample application is a web server and we should not have to have our terminal connected to the container.

Detached mode will be very helpful when we are using Linux. because in windows we can open number of terminals. so that you can run more commands which are needed.



12. **docker pull hello-world :** After deleting the images we can get back in two ways by pulling and the other is running

running : docker run hello-world so that it will be downloaded from the docker hub (so it is used when we want to run a container)

Pull an Image : **docker pull hello-world** (so it is used when we want only image)

```
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

13. **docker rmi hello-world:** To remove the image firstly we need to stop the container or delete the container.

```
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rmi hello-world
Error response from daemon: conflict: unable to remove repository reference "hello-world" (must force) - container 60094154334e is using
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rm hello-world
Error: No such container: hello-world
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS                     PORTS      NAMES
60094154334e   hello-world   "/hello"                 5 minutes ago   Exited (0) 5 minutes ago              exciting_zhukovsky
edae92ec893b   redis         "docker-entrypoint.s…"   2 days ago      Exited (255) 2 hours ago   6379/tcp   redis
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rm exciting_zhukovsky
exciting_zhukovsky
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddbc346
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
```

14. docker run python:3.8.0: To run a particular version of the image we use **tag**

```
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddbc346
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker run python.3.8.0
Unable to find image 'python.3.8.0:latest' locally
docker: Error response from daemon: pull access denied for python.3.8.0, repository does not exist or may require 'docker login': denied
See 'docker run --help'.
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$ docker run python:3.8
Unable to find image 'python:3.8' locally
3.8: Pulling from library/python
f606d8928ed3: Already exists
47db815c6a45: Already exists
bf4849400000: Already exists
a572f7a256d3: Already exists
8f7d05258955: Already exists
7110f04115ae: Already exists
a8dd20751451: Pull complete
dc4fe6dbfdaf: Pull complete
0cd7d3e2da99: Pull complete
Digest: sha256:11e6a1575d61c5534e2cb711e58f1d6616847370b1337a0605bba751c9fbca37
Status: Downloaded newer image for python:3.8
(base) manideep@manideep-XPS-13-9305:~/Docker Assignment/Hello World$
```

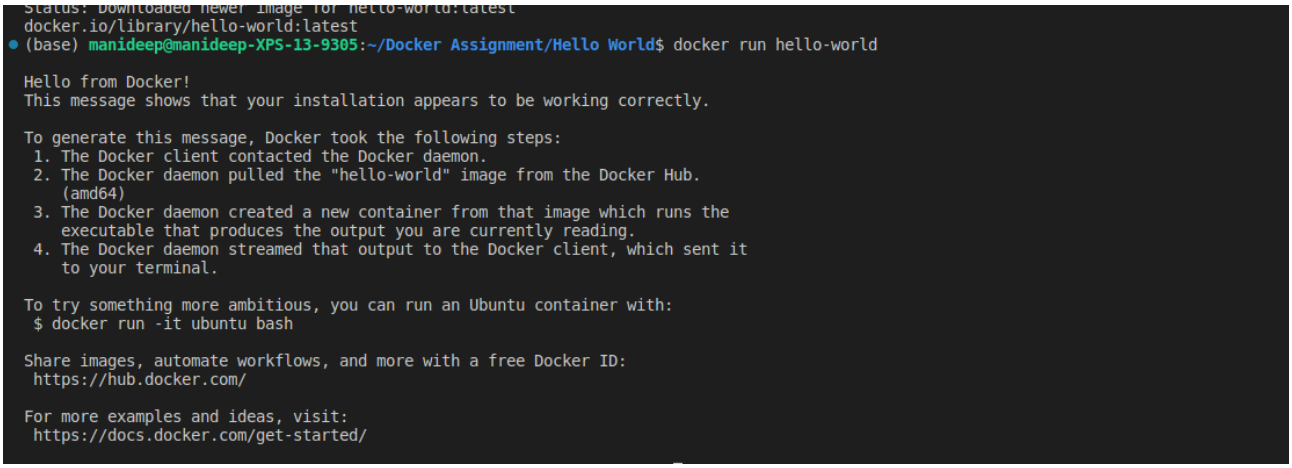
15. docker run -d helloworld: To run a container in detached mode

ASSIGNMENT 2:

docker pull hello-world : After deleting the images we can get back in two ways by pulling and the other is running

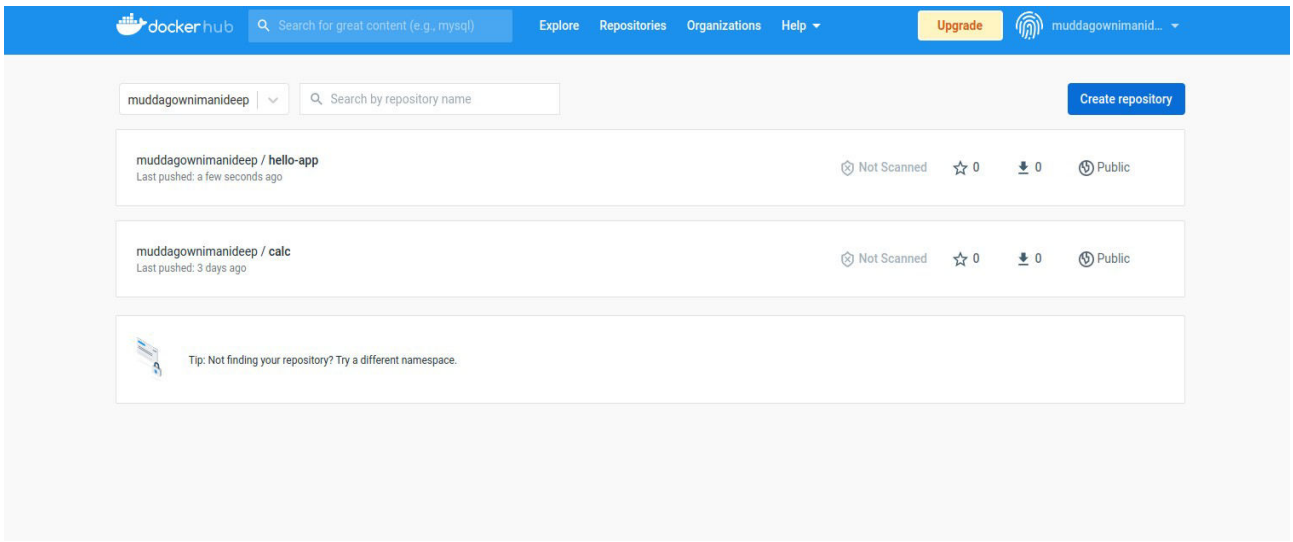
running : docker run hello-world so that it will be downloaded from the docker hub (so it is used when we want to run a container)

Pull an Image : **docker pull hello-world** (so it is used when we want only image)



ASSIGNMENT 3:

- docker login
- enter password
- docker build . t muddagownimanideep/hello-app
- docker images
- docker run muddagownimanideep/hello-app
- docker push muddagownimanideep/hello-app


docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Help
Upgrade
muddagownimanid...
muddagownimanideep
Search by repository name
Create repository
muddagownimanideep / hello-app
Last pushed: a few seconds ago
Not Scanned
0
0
Public
muddagownimanideep / calc
Last pushed: 3 days ago
Not Scanned
0
0
Public
Tip: Not finding your repository? Try a different namespace.