Docker Lab: Installations and Configurations

In this lab we will install docker on top of ubuntu machine and perform different container operations on that.

Task Breakdown

- Install docker on ubuntu
- Create account on docker hub
- Create account on github
- Running docker containers
- Performing containers operations
- Explore other container commands

Task 1: Install docker on ubuntu

Open Ubuntu terminal and run below commands to install docker

sudo su - (Provide sudo password)

Comment: This command gets you the root(admin) access to install docker packages.

apt-get update

Comment: This command will update/connect the satellite repository with the latest package details.

apt-get remove docker docker-engine docker.io

Comment: This command will remove any residues of the old docker package if present. This is not a mandatory command to run.

apt install docker.io

Comment: This command will download the latest package from docker's website and install docker on the machine.

systemctl start docker && systemctl enable docker

Comment: This command will start and enable the docker service which automatically starts docker engine if the OS restarts.

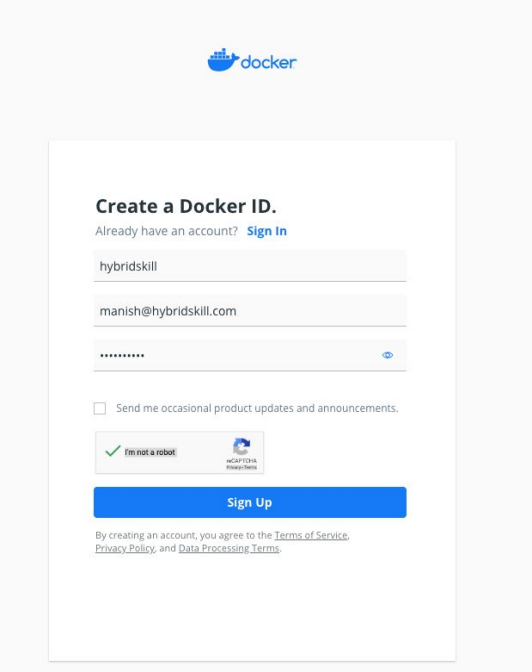
docker --version

Comment: To check the docker version installed currently in the machine

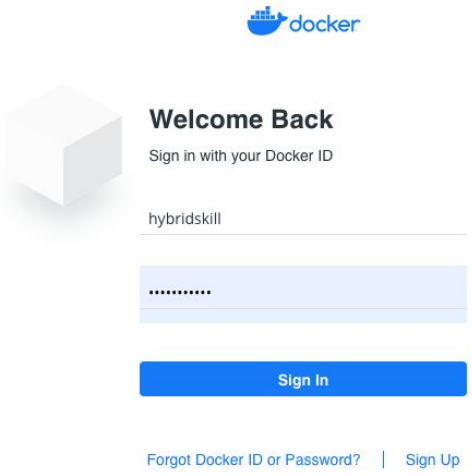
Docker Lab: Installations and Configurations

Task 2: Create account in docker hub

1. Go to https://hub.docker.com/
2. Click on the Sign up button.
3. Provide the account information and click on sign up



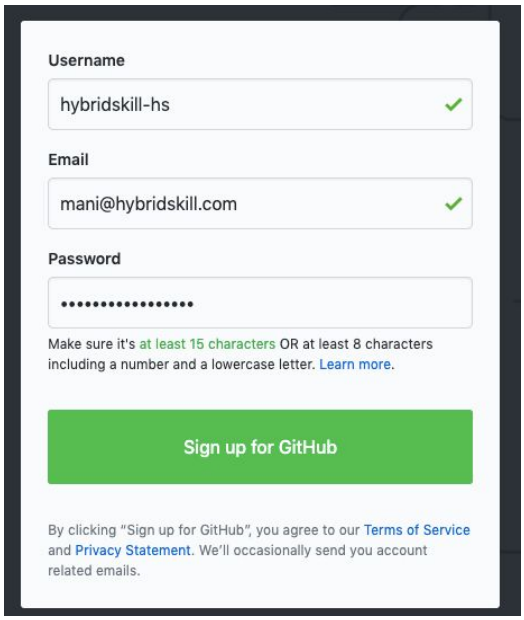
4. Confirm your email by logging to your email account.
5. Once confirmed, go to https://hub.docker.com/ and click on sign in.
6. Provide your credentials and click on sign in.



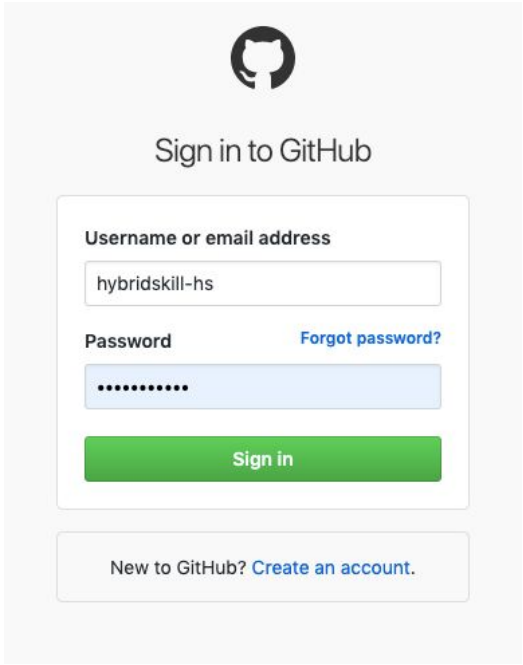
Docker Lab: Installations and Configurations

Task 3: Create account on Github

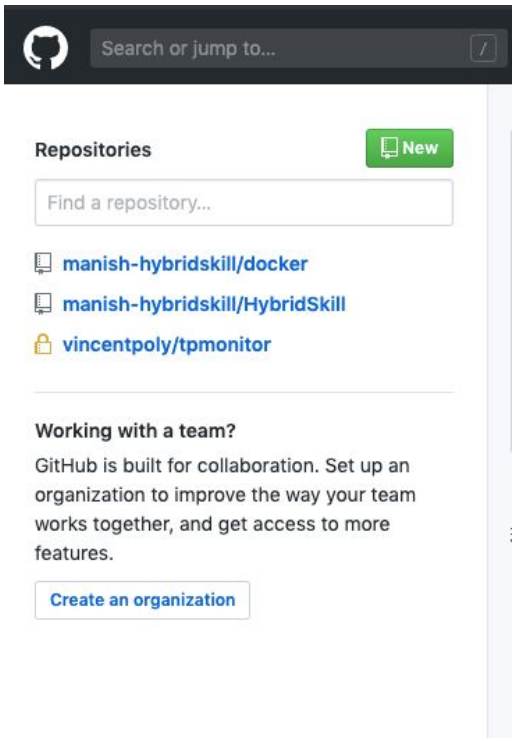
1. Go to https://github.com/
2. Click on the Sign up button.
3. Provide the account information and click on sign up



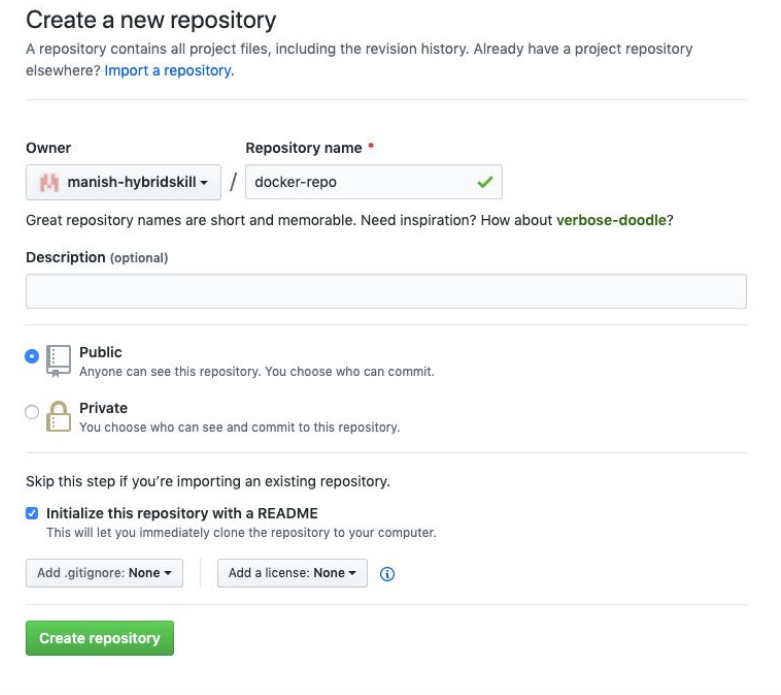
4. Confirm your email by logging to your email account.
5. Once confirmed, go to https://github.com/ and click on sign in.
6. Provide your credentials and click on sign in.



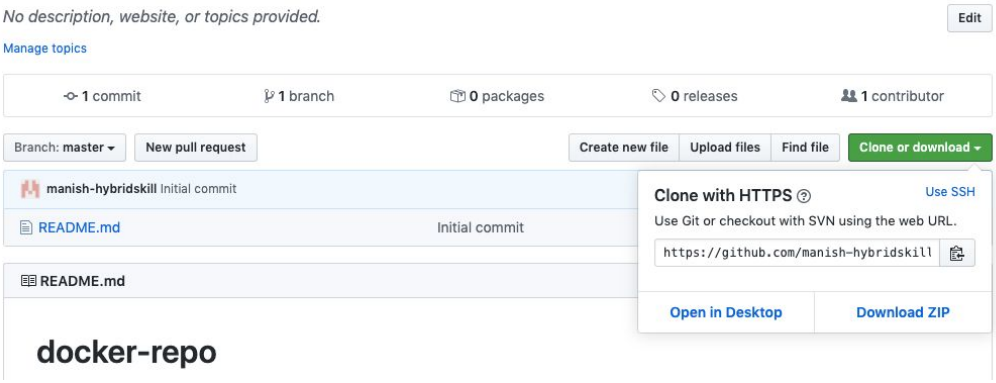
7. Click on create new and give a name to your repository.



8. Click on create. Your repository will be created.



9. Click on clone and copy the URL



10. Open the terminal of your virtual machine and run the below commands to install and configure github.

apt-get update

apt-get install git

git --version

git clone https://github.com/manish-hybridskill/docker-repo.git (Provide your repo URL here)

git config --global user.name "Manish Sharma" (Provide your username)

git config --global user.email "manish@hybridskill.com" (Provide your email id)

Task 4: Running docker containers

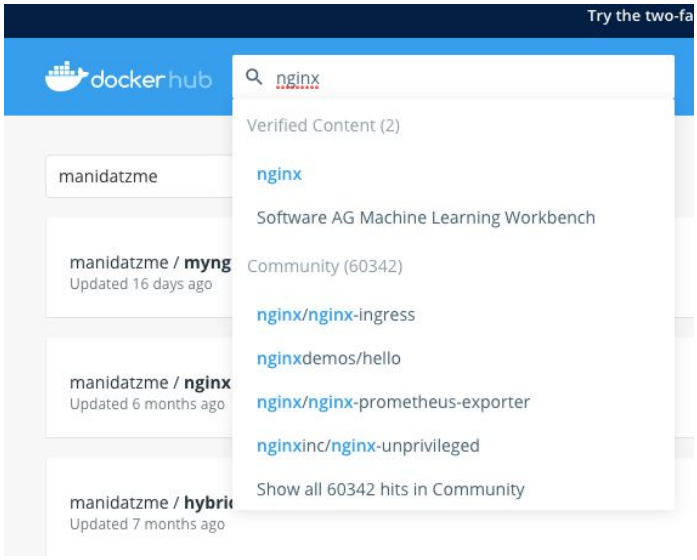
docker login

Id: Same id of docker hub as created in task 2
Password: Password of docker hub as created in task 2

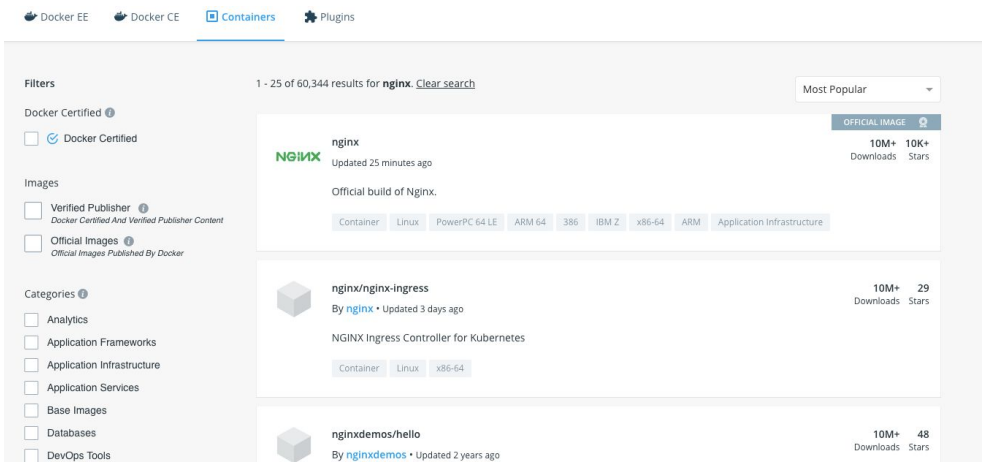
Comment: This command will enable the access from your local machine to push/pull/manage your images on your private docker hub registry.

Docker Lab: Installations and Configurations

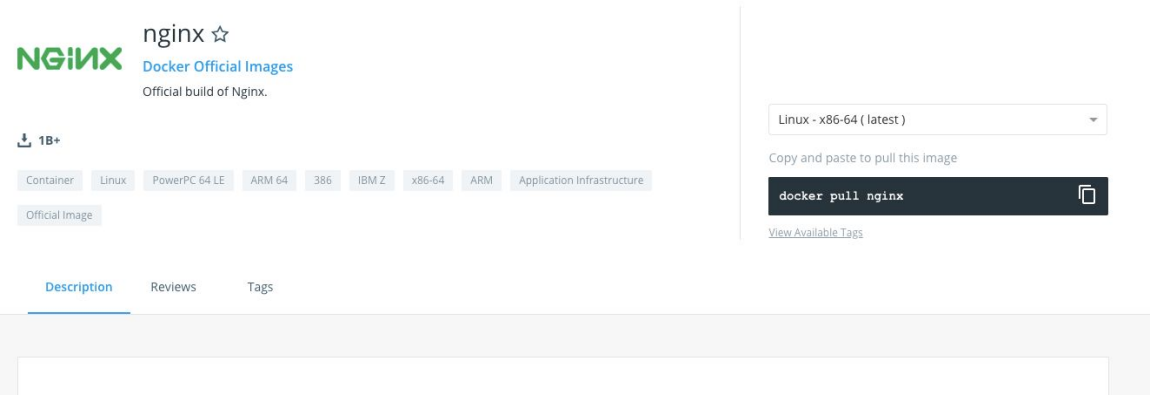
1. Navigate to https://hub.docker.com/ and search for the nginx image in the search bar.



2. It will list out all the images which have nginx as a package in it.



3. Click on the official image of Nginx



4. Pull the image to the virtual machine.

docker pull nginx

Comment: This command will pull the nginx image down to your virtual machine.

docker images

Comment: This command will list out all the images present in the virtual machine.

5. Run a nginx container and access out of the nginx image

docker container run -dt --name mywebcontainer -p 8080:80 nginx

Comment: This command will launch one container from nginx image

6. Verify that the container is running.

docker ps

Comment: This command shows the running instances of the container

7. Visit the Web page of the container to verify the process/port mapping by browsing the localhost ip with the port in your local browser **127.0.0.1:8080**

8. You should see web page in the browser “Welcome to Nginx”

Docker Lab: Installations and Configurations

Task 5: Performing containers operations

1. Login to the container to change the index file

docker container exec -it mywebcontainer bash

Comment: This command will allow you login to the container instance

2. Go to the nginx doc directory.

cd /usr/share/nginx/html/

3. Change the index.html file to display your name

cat > index.html
"Welcome to Nginx, This is Manish Sharma"
ctl + d

Comment: If you are good at VIM or any other editor, use the same to make the changes.

To Install vim editor, run the below commands, as vim is not included in the base package of nginx

apt-get update && apt-get install vim -y

4. Exit the container and visit the browser again.
5. You will see the web page shows "Welcome to Nginx, This is Manish Sharma"

Task 6: Explore other container commands

docker container stop mycontainer

Comment: To stop the running container

docker container start mycontainer

Comment: To start the stopped container

docker container rm mycontainer

Comment: To remove the container completely

docker container kill mycontainer

Comment: To kill the unresponsive container

docker container stop $(docker ps -qa)

Comment: To stop all the containers

docker container run -dt --name mycontainer busybox ping -c10 google.com

Comment: To change the default process of the container

docker container logs mycontainer

Comment: To check the logs of the container

docker system df -v

Comment: To check the space/disk usage of the docker objects

docker container inspect mycontainer

Comment: To get the details of the container like network, volume, Ip address etc.

Note: There are more than 20 commands/options are available to run/manage docker containers , please visit https://docs.docker.com/ to explore more commands.