

DEVOPS TASK- 2

1) Installation of Docker:

CODE :

```
sudo apt install docker.io
```

Docker –version

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

```
sudo systemctl status docker
```

SCREENSHOT:

```

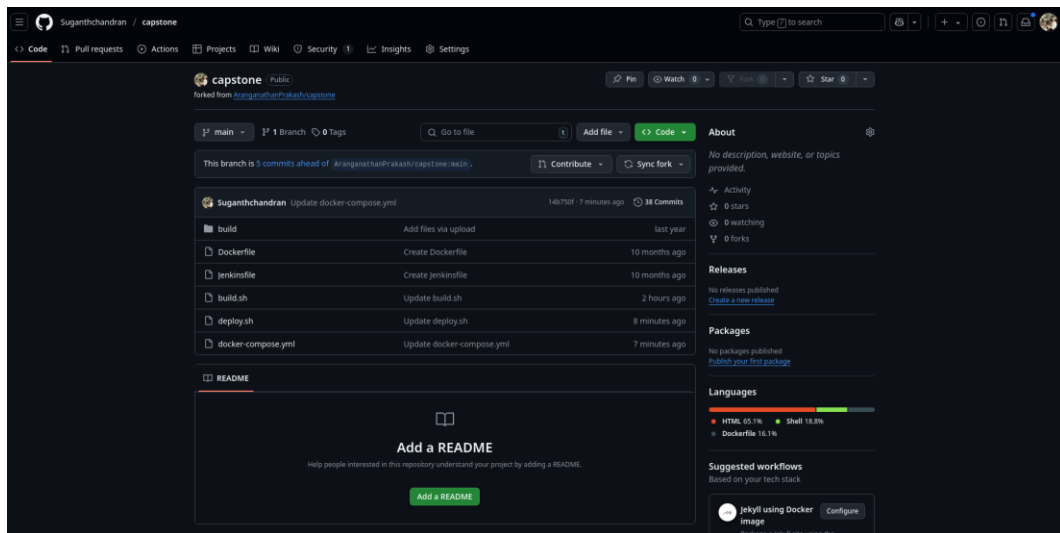
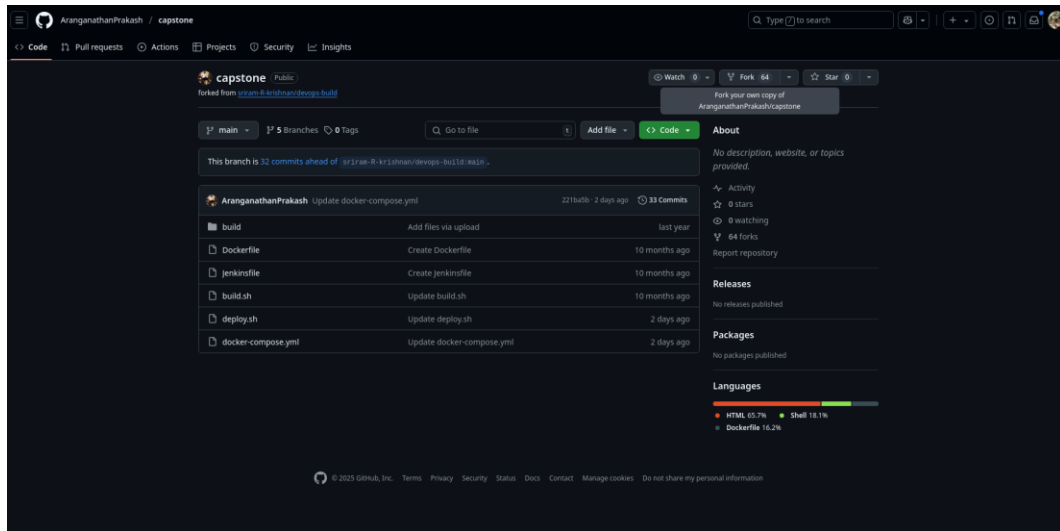
root@LAPTOP-6V70H2B0:~# apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1-24.04.1).
The following packages were automatically installed and are no longer required:
  libdrm-intel1 libpciaccess0 libensors-config libensors5
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
root@LAPTOP-6V70H2B0:~# docker --version
Docker version 26.1.3, build 1211113-0ubuntu1-24.04.1
root@LAPTOP-6V70H2B0:~# sudo systemctl start docker
root@LAPTOP-6V70H2B0:~# sudo systemctl enable docker
root@LAPTOP-6V70H2B0:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-20 06:44:32 UTC; 1h 32min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 9561 (dockerd)
      Tasks: 62.0M (O)
     Memory: 62.0M (O)
    CGroup: /system.slice/docker.service
            └─ 9561 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
               10253 /usr/bin/docker-proxy -proto tcp -host-ip ::0 -0.0.0 -host-port 70 -container-ip 172.17.0.2 -
               10261 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 70 -container-ip 172.17.0.2 -con>

Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185897791Z" level=warning msg="WARNIN>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185409232Z" level=warning msg="WARNIN>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185440810Z" level=warning msg="WARNIN>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185455418Z" level=warning msg="WARNIN>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185498240Z" level=info msg="Docker d>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185845402Z" level=info msg="Daemon o>
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.379205069Z" level=info msg="API list>
Mar 20 06:44:32 LAPTOP-6V70H2B0 systemd[1]: Started docker.service - Docker Application Container Engine.
Mar 20 06:45:16 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:45:16.408475978Z" level=info msg="Layer sh>
Mar 20 06:45:16 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:45:16.556116575Z" level=info msg="Layer sh>
lines 1-23/23 (END) ... skipping...
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)

```

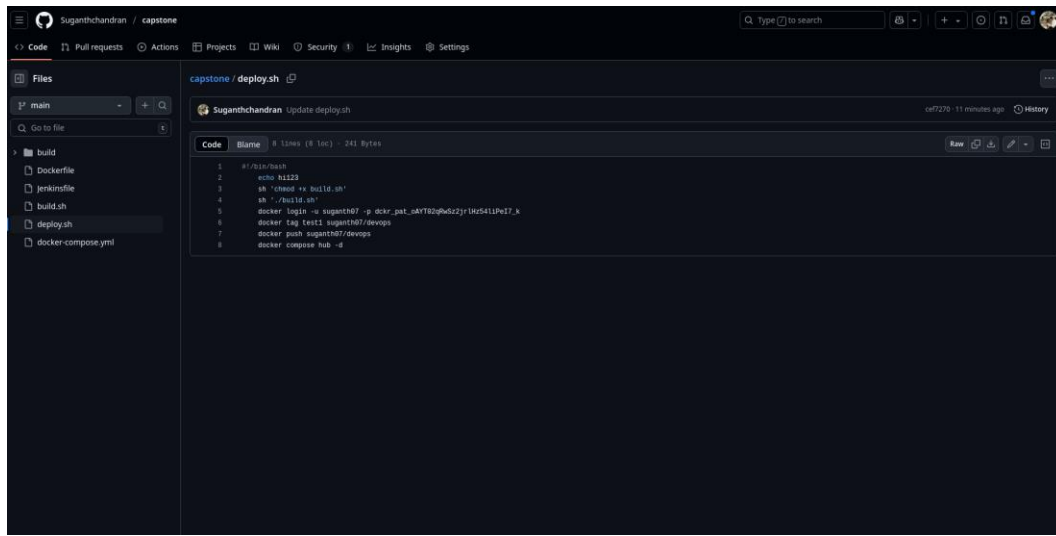
- 2) Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

SCREENSHOT :



- 3) Then change the token and repo name of the docker Hub in the deploy.sh file which is in our repository.

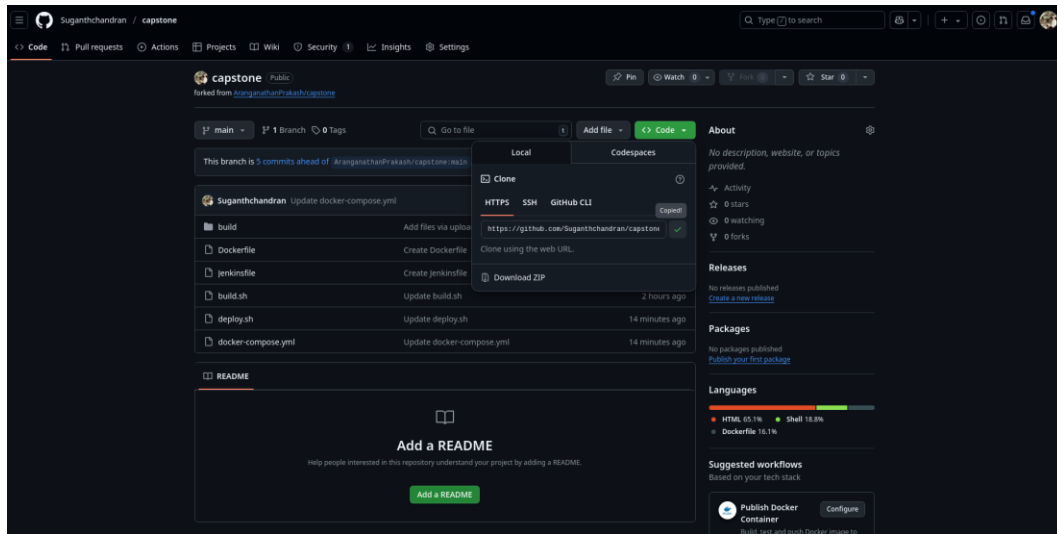
SCREENSHOT :



```
1 #!/bin/bash
2 echo hi123
3 sh 'chmod +x build.sh'
4 sh '-x build.sh'
5 docker login -u suganth07 -p dckr_pat_sAYT8ZqhwGZJr1Hc541PeS7_3
6 docker tag test1 suganth07/devops
7 docker push suganth07/devops
8 docker compose up -d
```

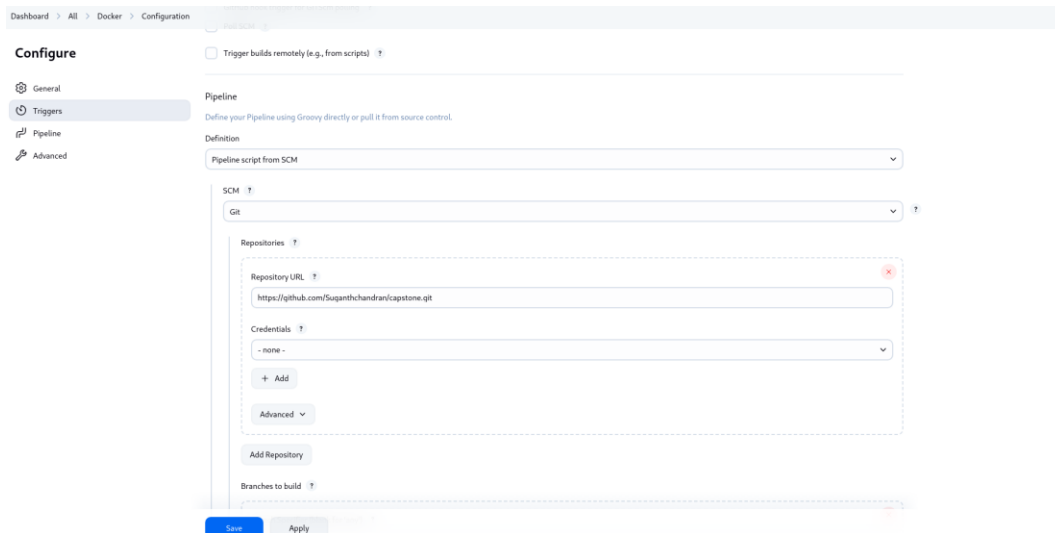
4) Then copy the GitHub link of the repository and go to Jenkins.

SCREENSHOT:



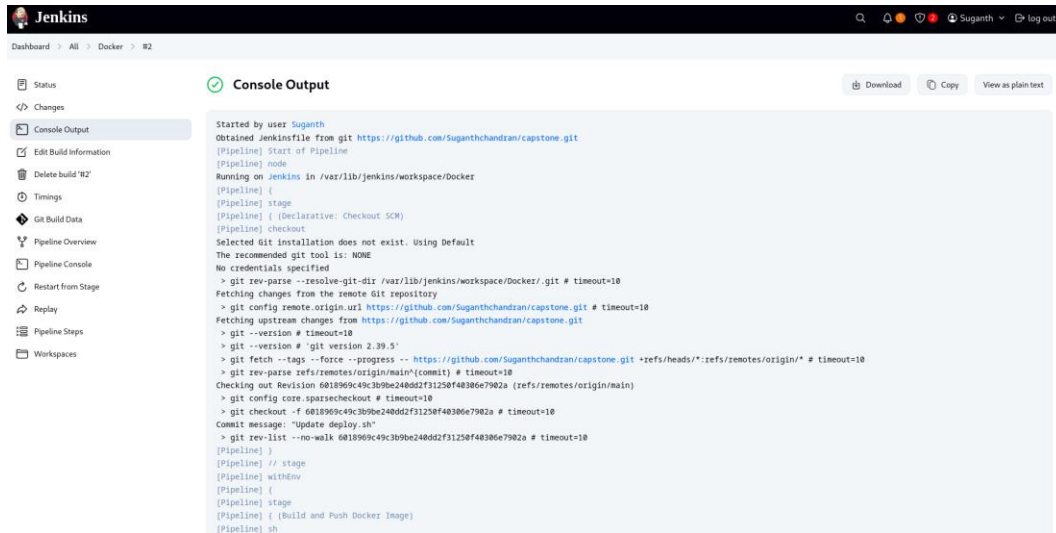
- 5) In Jenkins, create a new item (Job) with a type pipeline and add the copied GitHub url to it with the correct branch and Jenkinsfile.

SCREENSHOT:



- 6) After Creating the job, build it and it will give the console output and the docker image will be created.

SCREENSHOT:



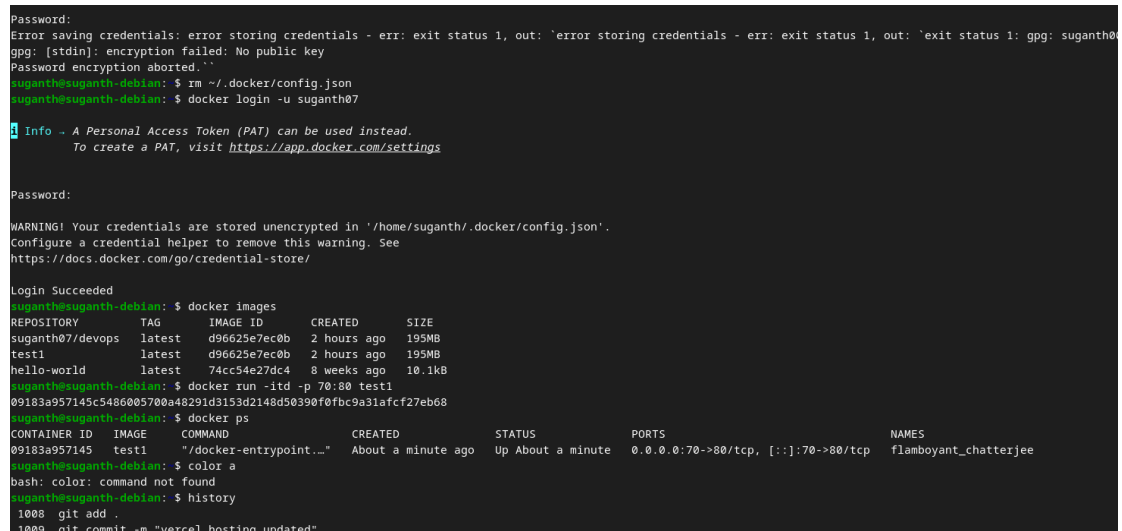
7) Now Built this docker image in the terminal with desired port number to it.

CODE:

docker images

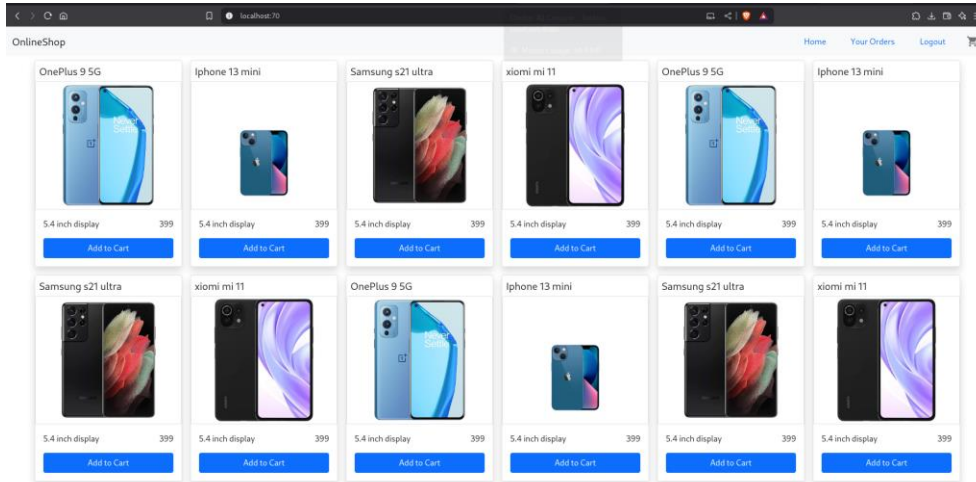
docker run -itd -p 70:80 test1

SCREENSHOT:



- 8) Go to the Browser and search for localhost:<PORT_NUMBER> and the respective application will be hosted.

SCREENSHOT:



- 9) But, Instead of running the image by manually , we can also write the command for running in a file called docker-compose.yml

CODE:

version: '3'

services:

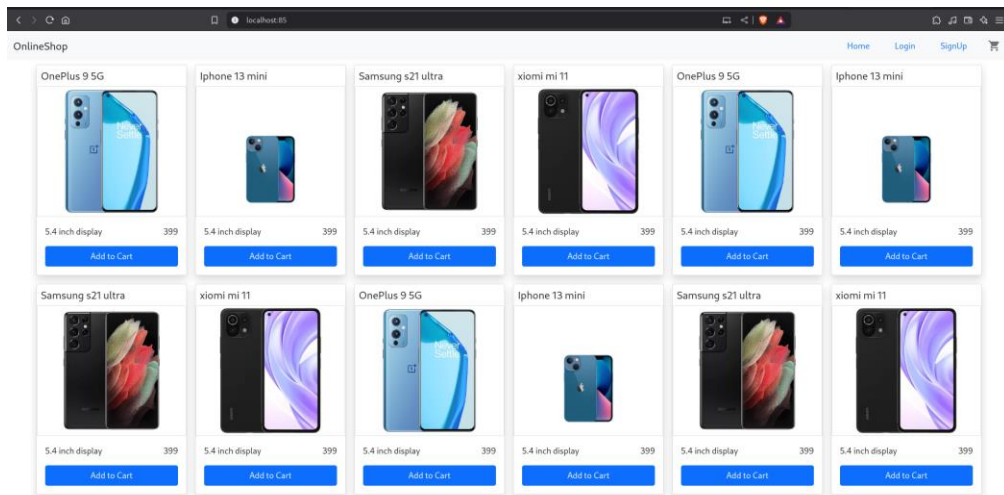
 react-capstone:

 image: "test1"

 ports:

 - "85:80"

SCREENSHOT:



By Creating this, we no need to run the image by manually. (It will automatically run)