

DEVOPS TASK- 2

1) Installation of Docker:

CODE :

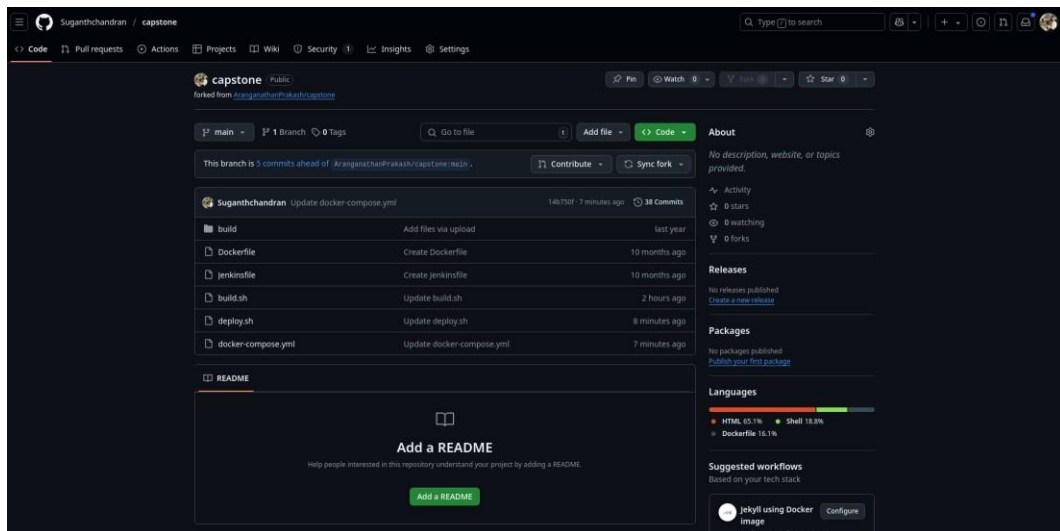
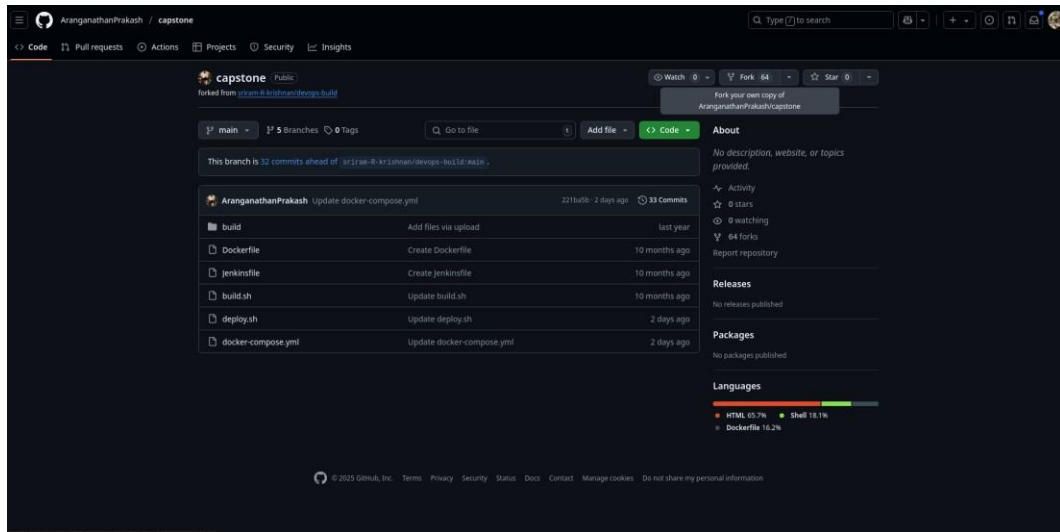
```
sudo apt install docker.io
Docker --version sudo systemctl
start docker sudo systemctl
enable docker
sudo systemctl status docker
```

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:
  libdw-intel libpciaccess0 libsensors-config libsensors5
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 26.1.3-0ubuntu1~24.04.1
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: 30
     Memory: 62.0M (-)
    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 -con
               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="2025-03-20T06:44:32.185997971Z" 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="2025-03-20T06:44:32.1854058119Z" level=warning msg="WARNIN
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185408240Z" level=info msg="Docker o
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185845402Z" level=info msg="Daemon h
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.379205869Z" 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="2025-03-20T06:45:16.405475078Z" level=info msg="Layer sh
Mar 20 06:45:16 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:45:16.550116975Z" 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)
```

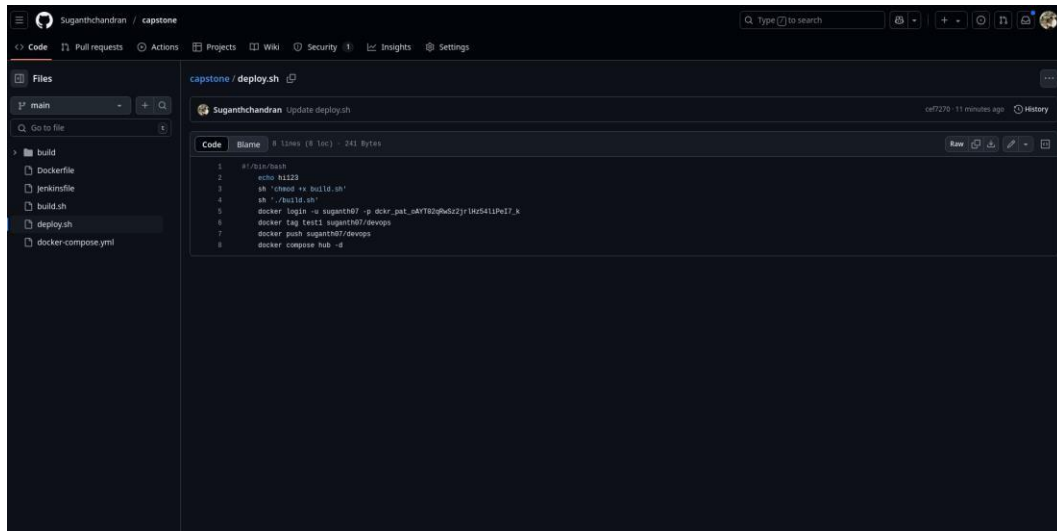
2) Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

SCREENSHOT :



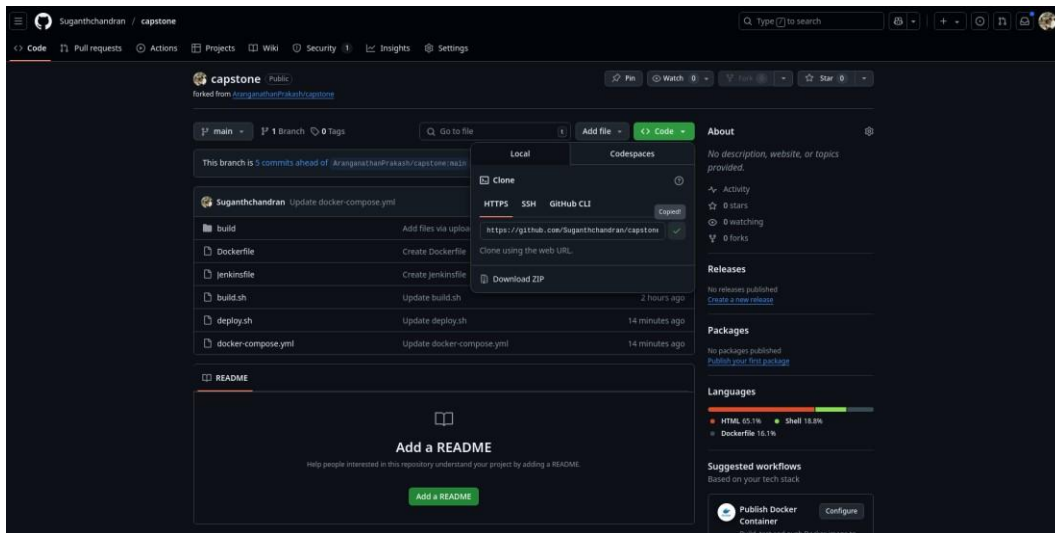
- Then change the token and repo name of the docker Hub in the deploy.sh file which is in our repository.

SCREENSHOT :



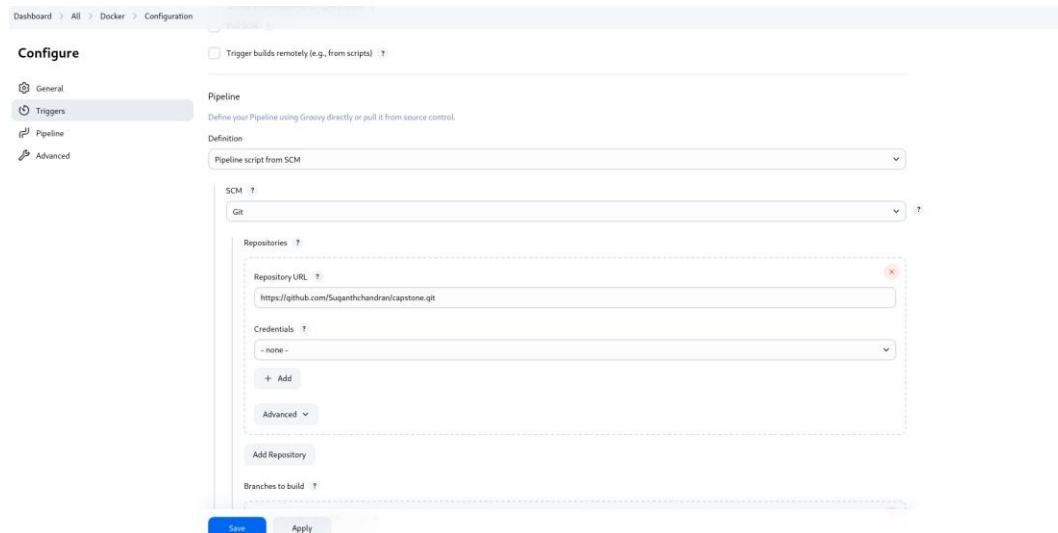
- 4) Then copy the GitHub link of the repository and go to Jenkins.

SCREENSHOT:



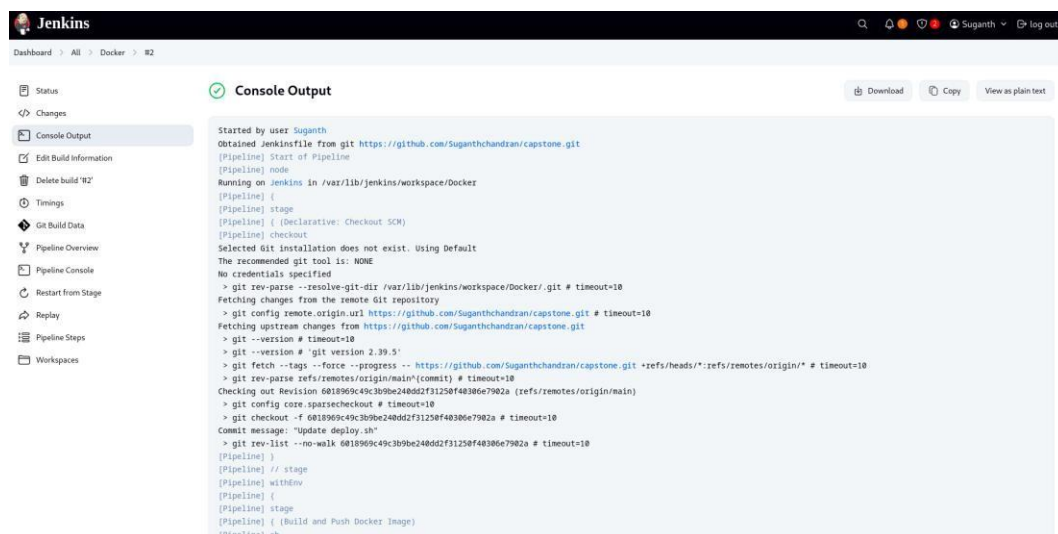
- 5) In Jenkins, create a new item (Job) with a type pipeline and add the copied GitHub url to it with the correct branch and Jenkinsfile.

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:

```

Password:
Error saving credentials: error storing credentials - err: exit status 1, out: 'error storing credentials - err: exit status 1, out: 'exit status 1: gpg: suganth0
gpg: [stdin]: encryption failed: No public key
Password encryption aborted.
suganth@suganth-debian: $ rm -f .docker/config.json
suganth@suganth-debian: $ docker login -u suganth07

Info - A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

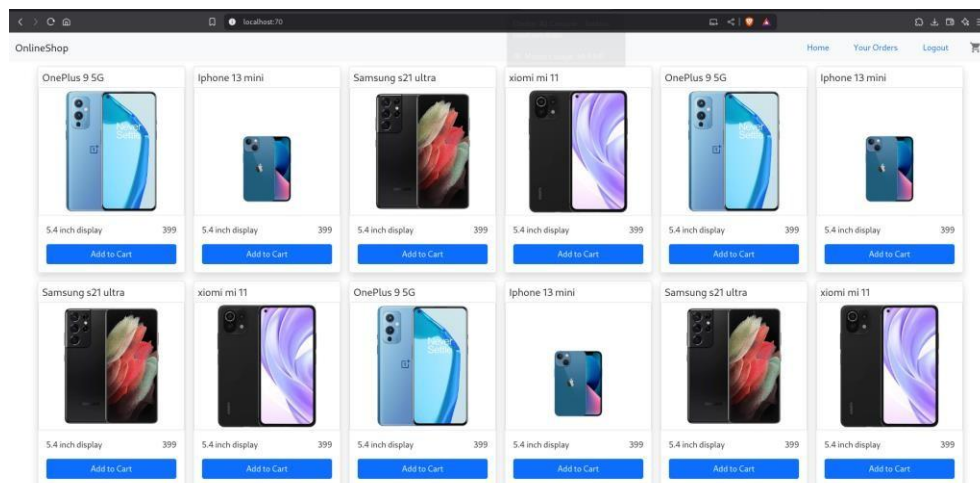
Password:

WARNING! Your credentials are stored unencrypted in '/home/suganth/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
suganth@suganth-debian: $ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
suganth07/devops    latest     d96625e7ec0b  2 hours ago   195MB
test1               latest     d96625e7ec0b  2 hours ago   195MB
hello-world         latest     74cc54e27dc4  8 weeks ago   10.1kB
suganth@suganth-debian: $ docker run -itd -p 70:80 test1
09183a957145c5486005700a48291d3153d2148d50390f0fbc9a31afc27eb68
suganth@suganth-debian: $ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
09183a957145   test1     "/docker-entrypoint..." About a minute ago Up About a minute   0.0.0.0:70->80/tcp, [::]:70->80/tcp   flamboyant_chatterjee
suganth@suganth-debian: $ color a
bash: color: command not found
suganth@suganth-debian: $ history
1008 git add .
1009 git commit -m "vercel hosting updated"
```

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

SCREENSHOT:

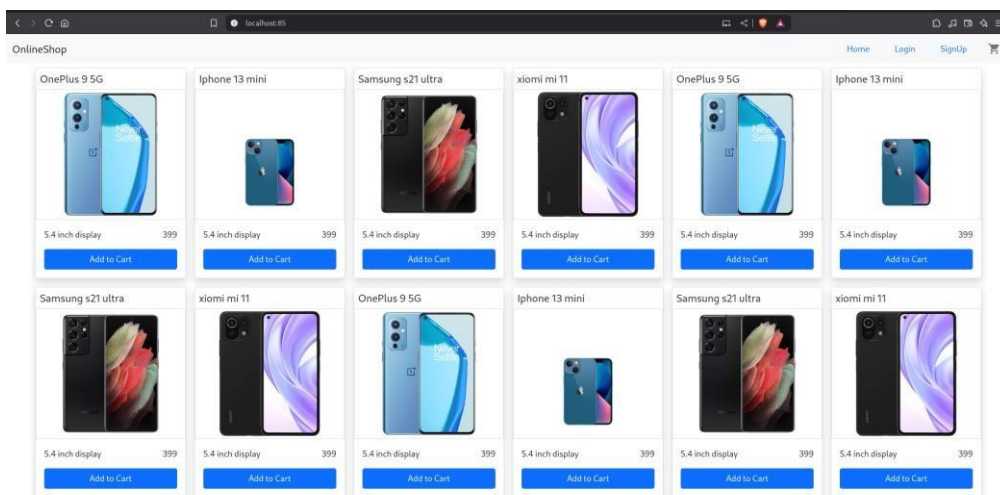


- 9) But, Instead of running the image by manually , we can also write the command for running in a file called docker-compose.yml

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)

- 10) Adding Webhook to it which is available in GitHub for automatic build of the project.

Installing ngrok and with these command to get the Webhook Link.

SCREENSHOT:

```
suganth@suganth-debian:~$ sudo snap install ngrok
[sudo] password for suganth:
snap "ngrok" is already installed, see 'snap help refresh'
suganth@suganth-debian:~$ ngrok config add-authtoken 2aa80vQJhkuuufCzLz3UHz28Dy54_Sq8xawTb0ts0wcn0wt
Auth token saved to configuration file: /home/suganth/snap/ngrok/255/.config/ngrok/ngrok.yml
suganth@suganth-debian:~$ ngrok http 8080
suganth@suganth-debian:~$
```

```
suganth@suganth-debian:~$ sudo snap install ngrok
[sudo] password for suganth:
snap "ngrok" is already installed, see 'snap help refresh'
suganth@suganth-debian:~$ ngrok config add-authtoken 2aa80vQJhkuuufCzLz3UHz28Dy54_Sq8xawTb0ts0wcn0wt
Auth token saved to configuration file: /home/suganth/snap/ngrok/255/.config/ngrok/ngrok.yml
suganth@suganth-debian:~$ ngrok http 8080
suganth@suganth-debian:~$
```

The screenshot shows the GitHub repository settings for 'suganthchandan/capstone'. The 'Webhooks' tab is selected in the left sidebar. The main content area shows a list of webhooks. One webhook is listed with the URL 'https://8071-2609-4094-3043-9807...@github.com'. The status is 'Last delivery was successful'. There are 'Edit' and 'Delete' buttons for this webhook. The top navigation bar includes links for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The bottom of the page shows the GitHub footer with copyright information and links for 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact', and 'Manage cookies'.

11) Tick the checkbox of GitHub hook trigger for GITScm polling in Jenkins.

SCREENSHOT :

