

## Task 2 For DevOps:

### 1) Installation of Docker:

#### Code:

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

```
docker --version
```

```
sudo systemctl start docker
```

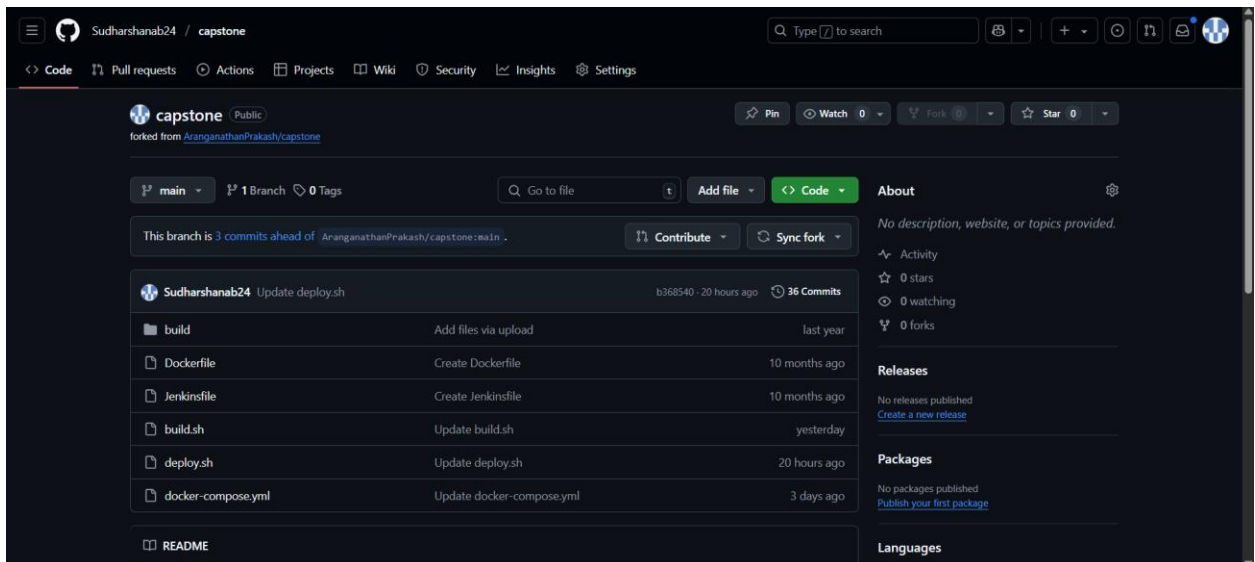
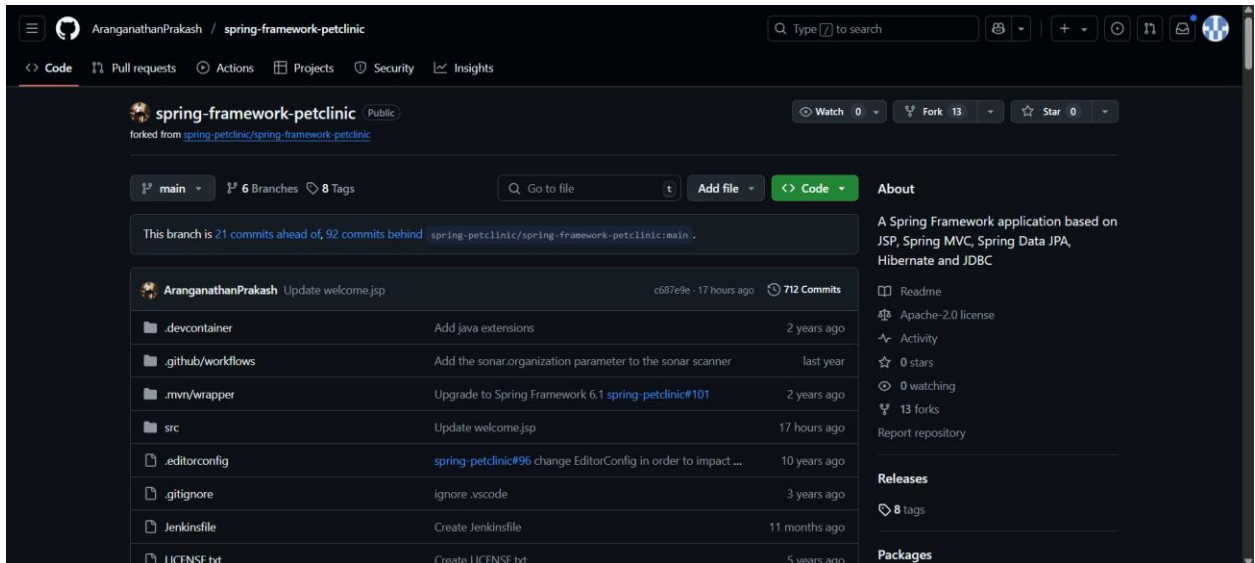
```
sudo systemctl status docker
```

```
sudo systemctl enable docker
```

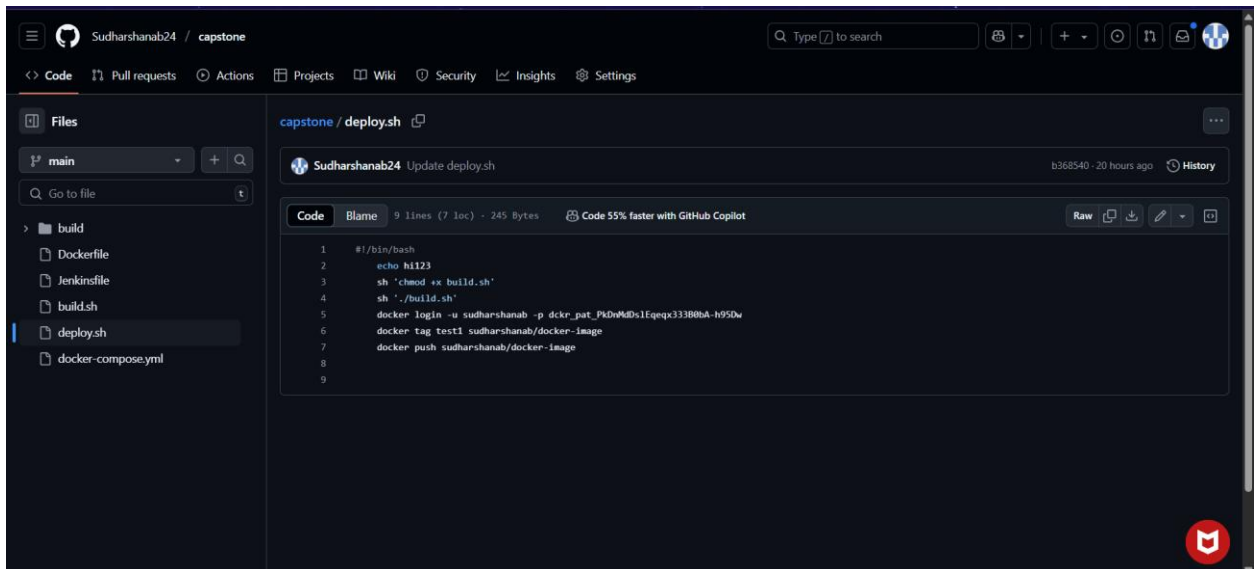
```
root@LAPTOP-6V70H2B0:~# apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1~24.04.1).
The following packages were automatically installed and are no longer required:
  libdrm-intel1 libpciaccess0 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>
                  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="2025-03-20T06:44:32.185097971Z" 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 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="2025-03-20T06:45:16.550116575Z" 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

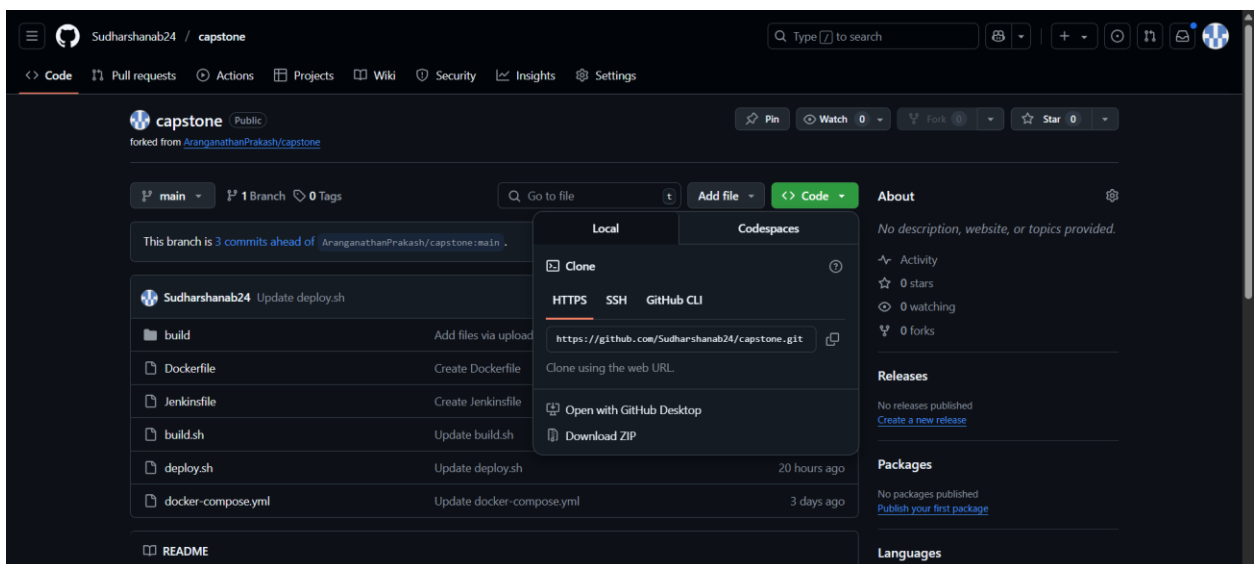


3) Then edit the deploy.sh file and enter the corresponding username and tokens



```
1 #!/bin/bash
2 echo hi123
3 sh 'chmod +x build.sh'
4 sh './build.sh'
5 docker login -u sudharshanab -p dckr_pat_PkDnM4DsIEeqqx333B0ba-h950w
6 docker tag test1 sudharshanab/docker-image
7 docker push sudharshanab/docker-image
8
9
```

4) Copy the GitHub link of the repository and go to Jenkins to create a pipeline project



5) In Jenkins create a new item(Job) with a pipeline type and add the Git URL to the respective branch and Jenkinsfile

Dashboard > task2 > Configuration

### Configure

- General
- Triggers
- Pipeline**
- Advanced

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Sudharshanab24/capstone.git

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Save Apply

Dashboard > devops > Configuration

### Configure

- General
- Triggers
- Pipeline**
- Advanced

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

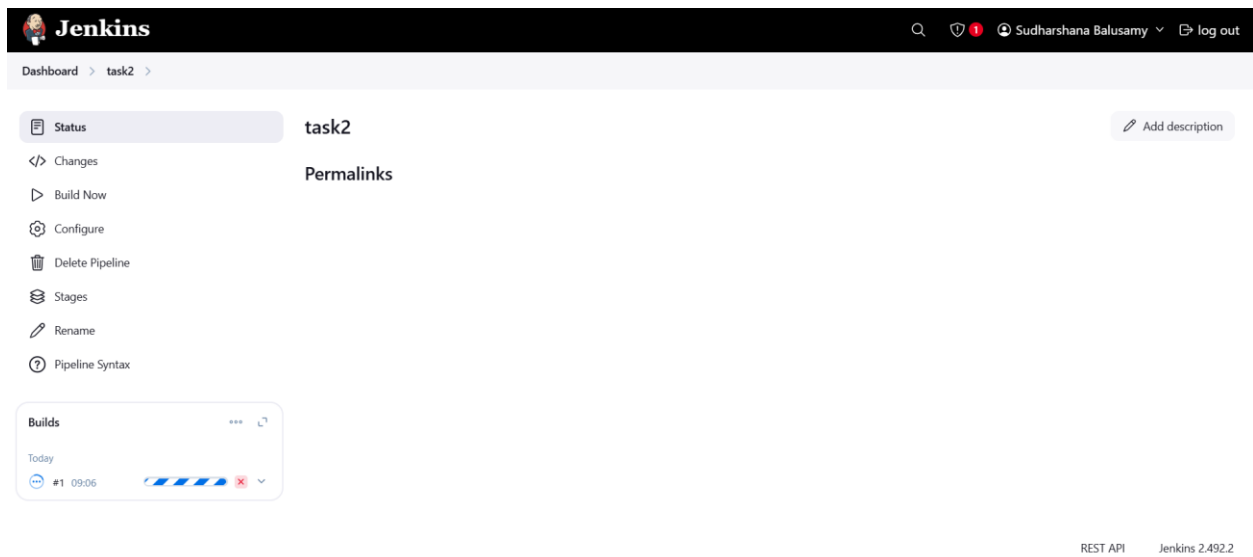
Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Save Apply

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



7) Now attach the desired port number to the build image

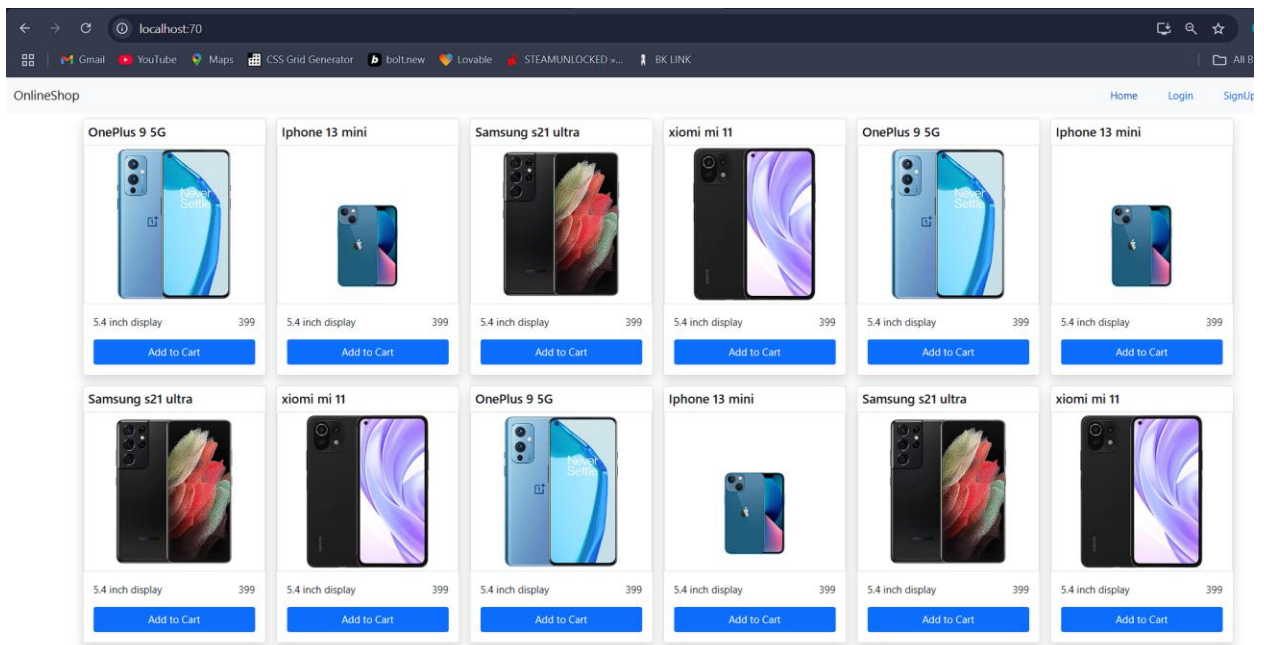
**CODE:**

docker images

docker build -itd -p 70:80 test1

**ScreenShot:**

8) Go to the Browser and search for localhost:<PORT\_NUMBER> and the respective website will be hosted



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

**Code:**

version: '3'

services:

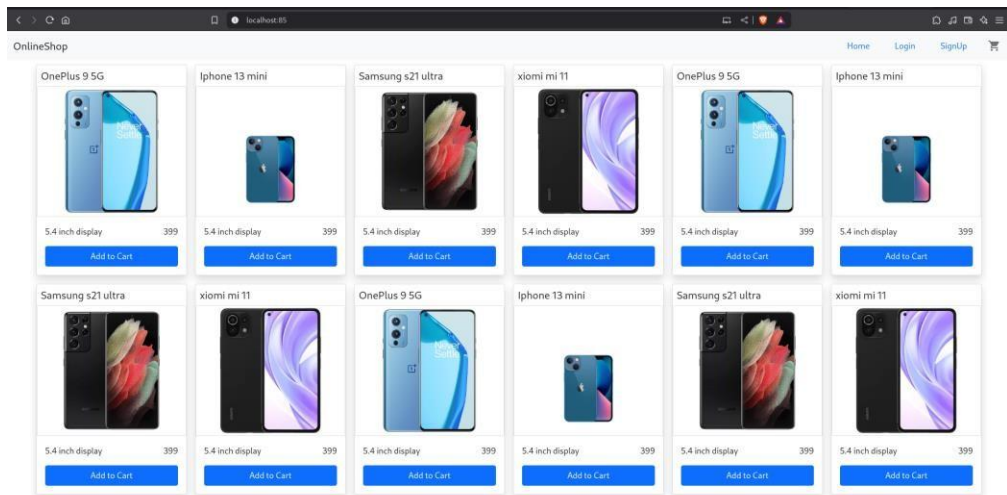
react-capstone:

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:

```
root@sudharshana:~# snap install ngrok
2025-03-22T03:39:35Z INFO Waiting for automatic snapd restart...
ngrok (v3/stable) 3.22.0 from Ngrok (ngrok-publisher) installed
root@sudharshana:~# curl -sSL https://ngrok-agent.s3.amazonaws.com/ngrok.asc \
| sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null \
&& echo "deb https://ngrok-agent.s3.amazonaws.com buster main" \
| sudo tee /etc/apt/sources.list.d/ngrok.list \
&& sudo apt update \
&& sudo apt install ngrok
deb https://ngrok-agent.s3.amazonaws.com buster main
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Get:8 https://ngrok-agent.s3.amazonaws.com buster InRelease [20.3 kB]
Get:9 https://ngrok-agent.s3.amazonaws.com buster/main amd64 Packages [7615 B]
Fetched 27.9 kB in 2s (16.4 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
128 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ngrok
0 upgraded, 1 newly installed, 0 to remove and 128 not upgraded.
Need to get 6756 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://ngrok-agent.s3.amazonaws.com buster/main amd64 ngrok amd64 3.22.0 [6756 kB]
Fetched 6756 kB in 4s (1646 kB/s)
Selecting previously unselected package ngrok.
(Reading database ... 43985 files and directories currently installed.)
Preparing to unpack .../ngrok_3.22.0_amd64.deb ...
Unpacking ngrok (3.22.0) ...
Setting up ngrok (3.22.0) ...
root@sudharshana:~#
```

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

### SCREENSHOT:



Configure

- General
- Triggers
- Pipeline
- Advanced

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ Github hook trigger for GITScm polling
- ☐ Poll SCM
- ☐ Trigger builds remotely (e.g., from scripts)

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/shafeer2511/capstone.git

Credentials

- none -

+ Add

Save Apply