

DEVOPS TASK 2

NAME : SANDEEP PS

ROLL NUM : 22CSL261

1) Installation of Docker:

Code:

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

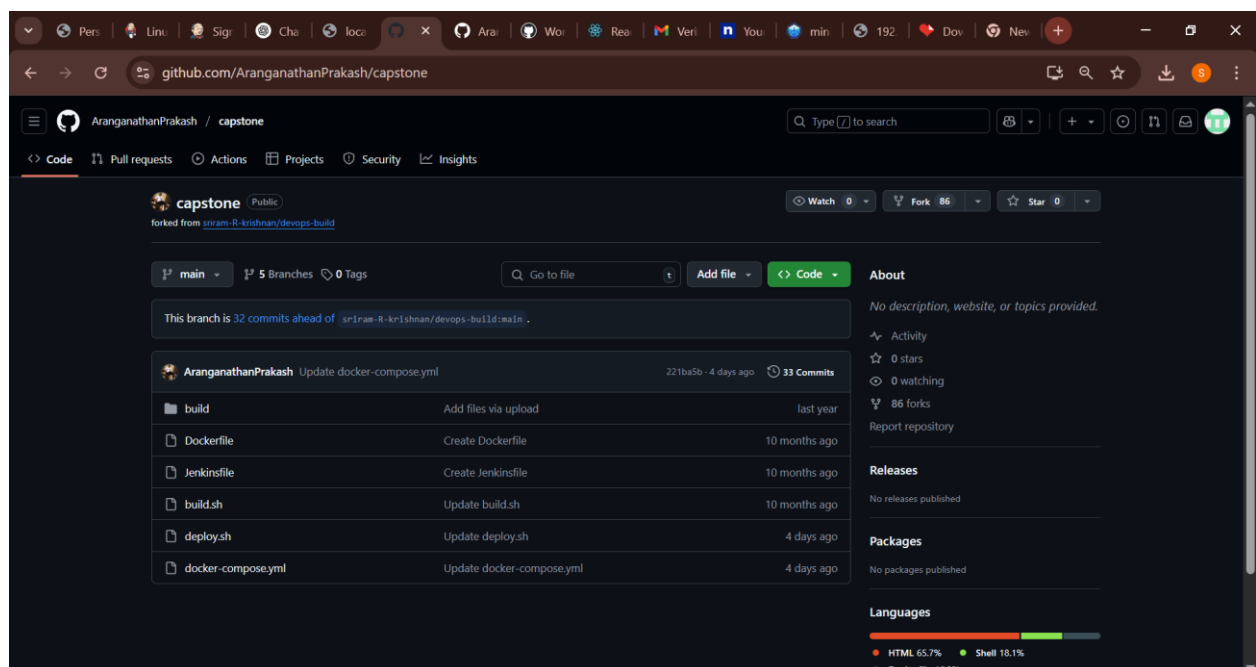
Screenshot:

```
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 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: 38
     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="2025-03-20T06:44:32.185897971Z" level=warning msg="WARNIN
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185899232Z" level=warning msg="WARNIN
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.185440816Z" 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="2025-03-20T06:44:32.185845002Z" level=info msg="Damon h
Mar 20 06:44:32 LAPTOP-6V70H2B0 dockerd[9561]: time="2025-03-20T06:44:32.379208869Z" 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)
```

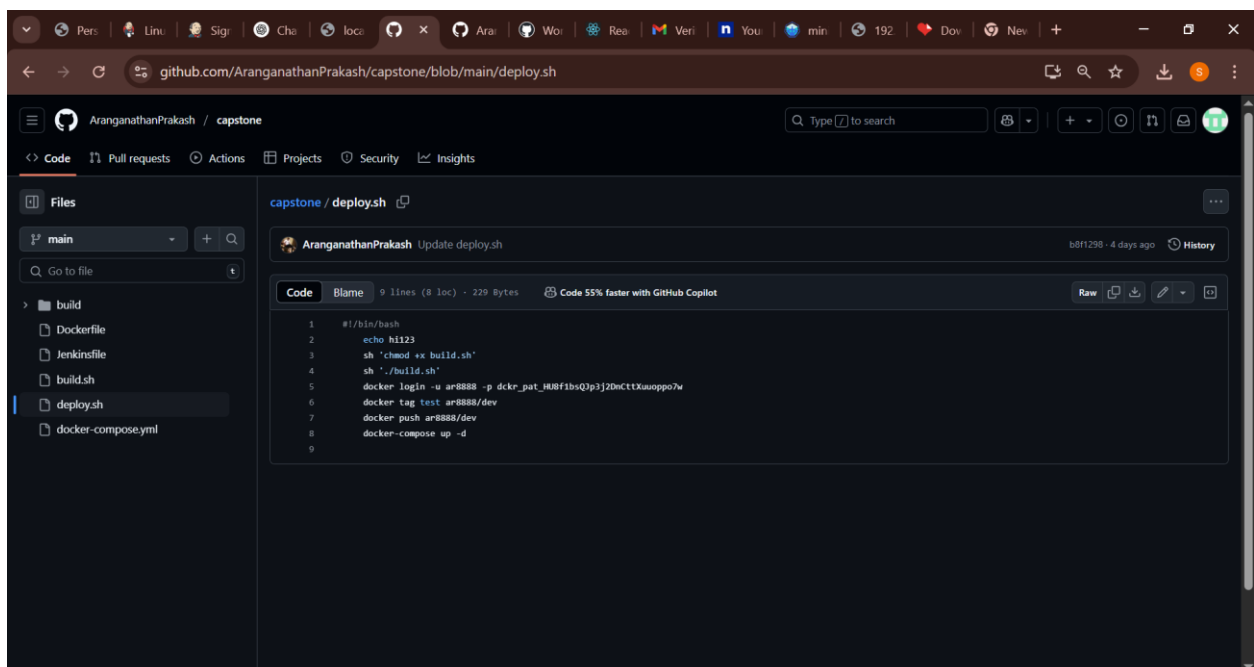
2) Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

Screenshot:



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

Screenshot

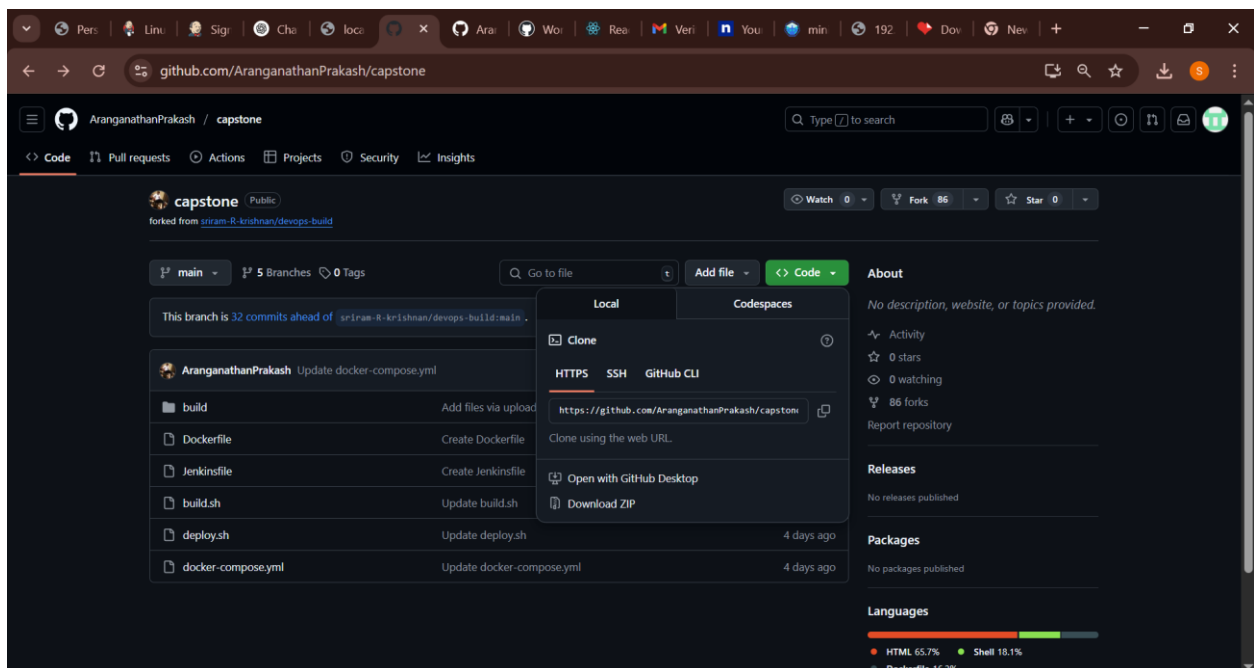


The screenshot shows a web browser displaying the GitHub repository page for 'AranganathanPrakash / capstone'. The file 'deploy.sh' is selected in the left sidebar. The main content area shows the code for 'capstone / deploy.sh', which is a shell script for Docker deployment. The script includes comments and commands for logging in to Docker Hub, tagging the image, pushing it to the 'dev' branch, and running the container.

```
1 #!/bin/bash
2 echo hi123
3 sh 'chmod +x build.sh'
4 sh './build.sh'
5 docker login -u ar8888 -p dckr_pat_H8Mf1bsQp3j2DncttXauoppo7w
6 docker tag test ar8888/dev
7 docker push ar8888/dev
8 docker-compose up -d
9
```

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

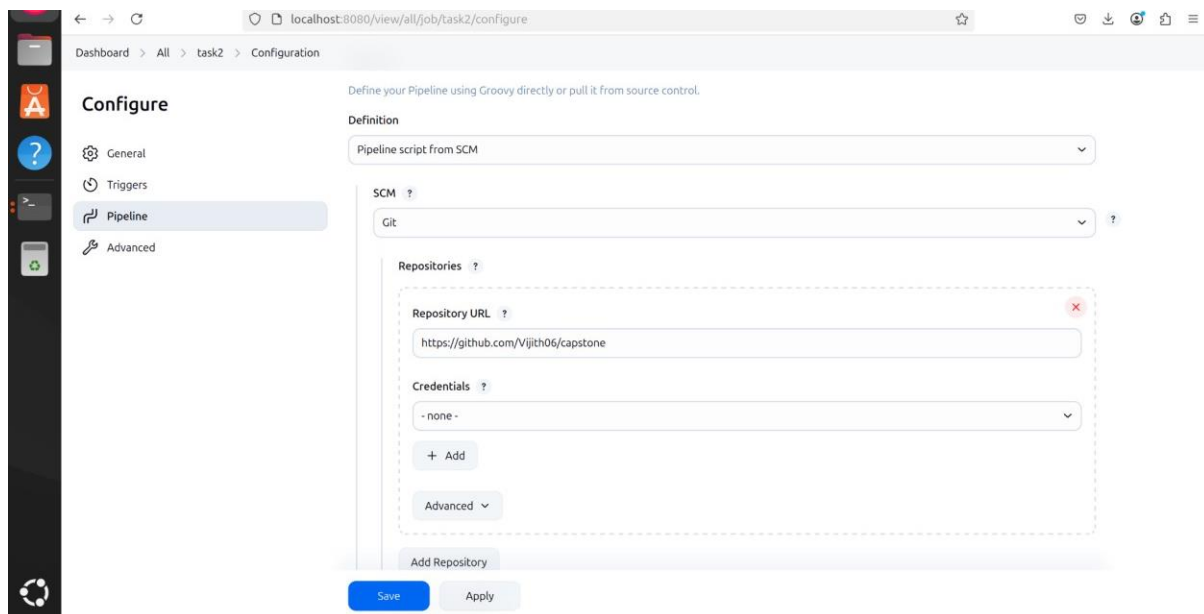
Screenshot:



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.

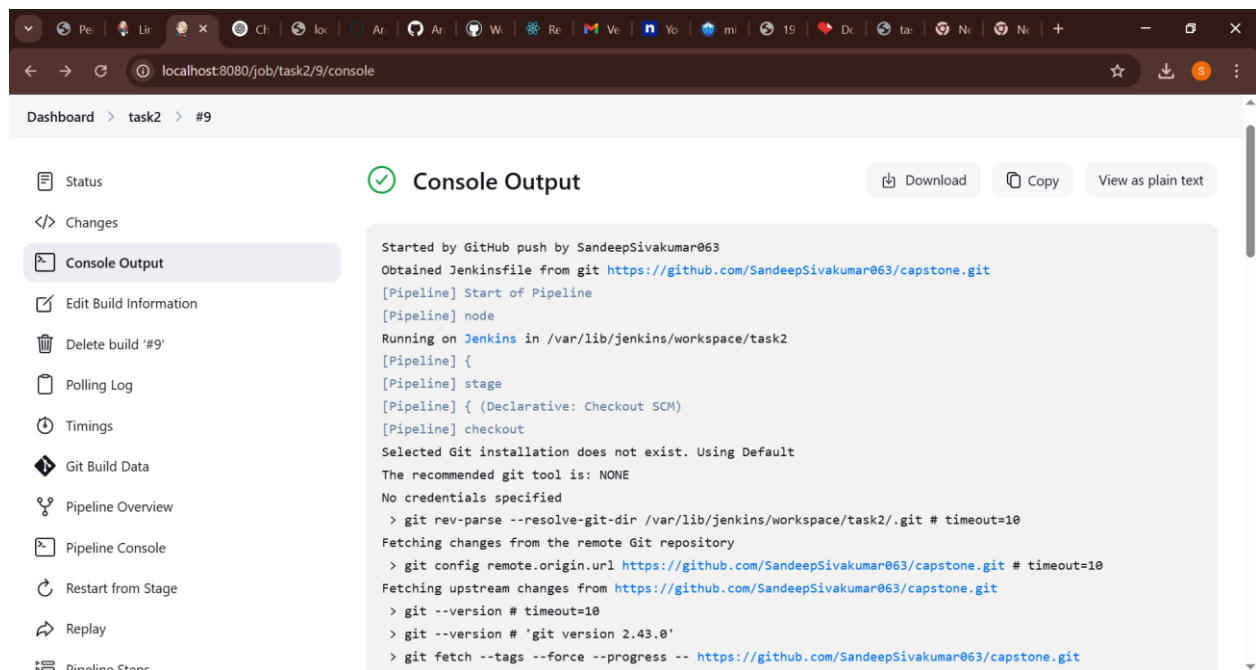
ScreenShot

:



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

Screenshot:



The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `localhost:8080/job/task2/9/console`. The breadcrumb navigation shows `Dashboard > task2 > #9`. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for the build, which is a Jenkins Pipeline. The output starts with 'Started by GitHub push by SandeepSivakumar063' and 'Obtained Jenkinsfile from git https://github.com/SandeepSivakumar063/capstone.git'. It then shows the pipeline stages: 'Start of Pipeline', 'node', and 'stage'. The 'stage' section is expanded, showing a 'checkout' step. The output for the checkout step includes: 'Selected Git installation does not exist. Using Default', 'The recommended git tool is: NONE', 'No credentials specified', and a series of git commands: `> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/task2/.git # timeout=10`, `> git config remote.origin.url https://github.com/SandeepSivakumar063/capstone.git # timeout=10`, `> git --version # timeout=10`, `> git --version # 'git version 2.43.0'`, and `> git fetch --tags --force --progress -- https://github.com/SandeepSivakumar063/capstone.git`. At the top of the console output area, there is a green checkmark icon and the text 'Console Output'. To the right of this, there are three buttons: 'Download', 'Copy', and 'View as plain text'.

```
Started by GitHub push by SandeepSivakumar063
Obtained Jenkinsfile from git https://github.com/SandeepSivakumar063/capstone.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/task2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/task2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/SandeepSivakumar063/capstone.git # timeout=10
Fetching upstream changes from https://github.com/SandeepSivakumar063/capstone.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/SandeepSivakumar063/capstone.git
```

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

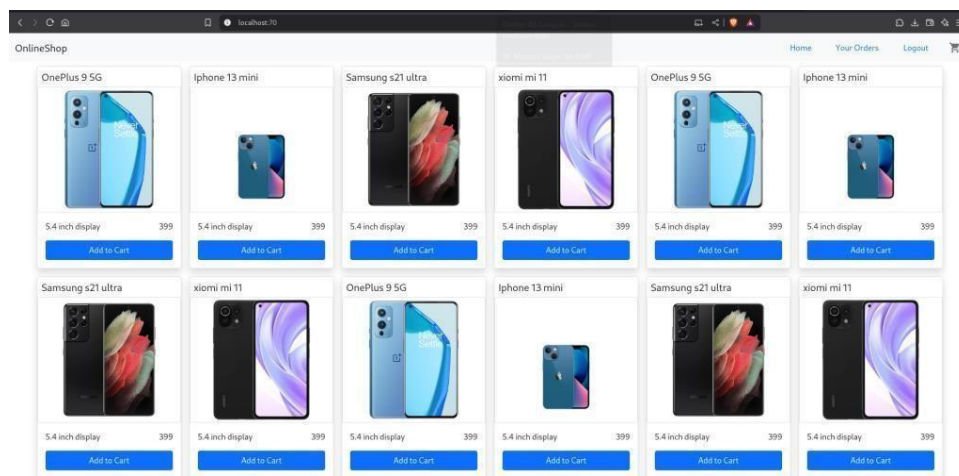
Code:

```
docker images
```

```
docker run -itd -p 70:80 test1
```

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

Screenshot:

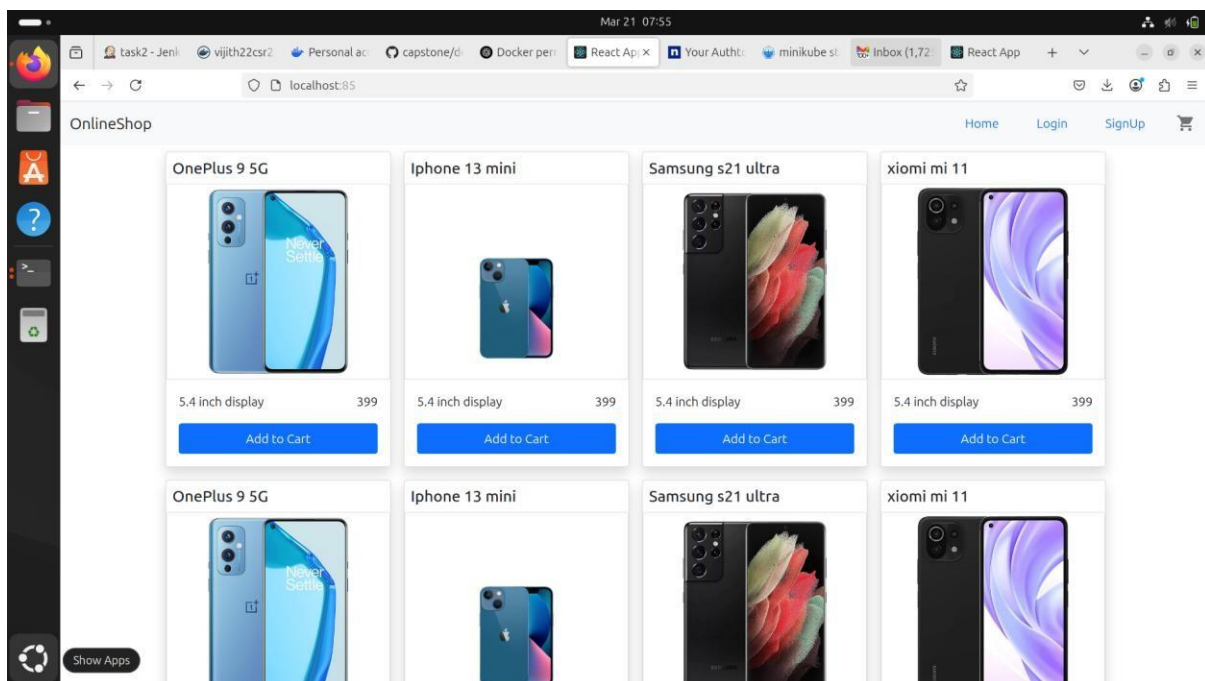


9) But, Instead of running the image by manually, we can also write the command for running in a file called docker-compose.yml

Code:

```
version: '3'
services:
  react-
  capstoneimage
  : "test1"
  ports:
    - "85:80"
```

Screenshot:



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

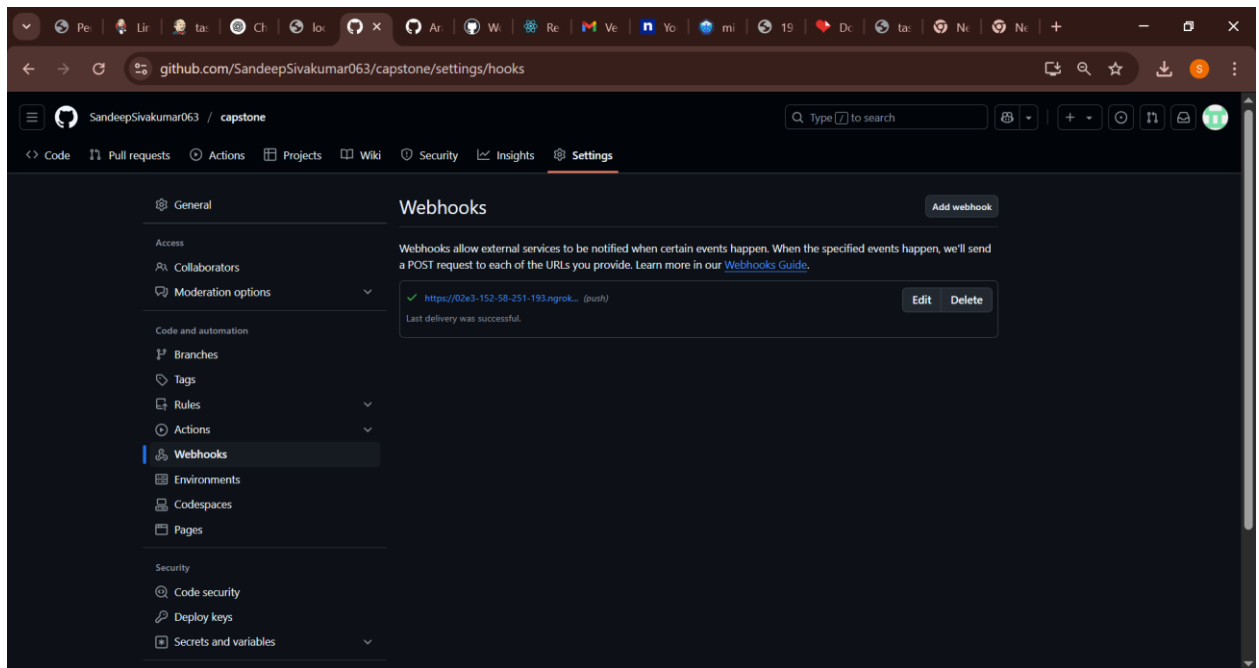
10) Adding Webhook to it which is available in GitHub for automatic build of the project.

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

Screenshot:

```
sandeep@DESKTOP-DNL16NN: ~  
Use "ngrok [command] --help" for more information about a command.  
sandeep@DESKTOP-DNL16NN:~$ sudo snap install ngrok  
snap "ngrok" is already installed, see 'snap help refresh'  
sandeep@DESKTOP-DNL16NN:~$ ngrok config add-authtoken 2ua0EI0PybzKGwG1um4wwqfhx9_3E4FayVamoTtTarRoD61h  
Authtoken saved to configuration file: /home/sandeep/snap/ngrok/268/.config/ngrok/ngrok.yml  
sandeep@DESKTOP-DNL16NN:~$ ngrok http 8080  
sandeep@DESKTOP-DNL16NN:~$
```

```
sandeep@DESKTOP-DNL16NN: ~  
ngrok  
Route traffic by anything: https://ngrok.com/r/iep  
Session Status      online  
Account             sandeep p s (Plan: Free)  
Version             3.22.0  
Region             India (in)  
Latency             95ms  
Web Interface       http://127.0.0.1:4040  
Forwarding           https://61ab-152-58-251-59.ngrok-free.app -> http://localhost:8080  
Connections  
    ttl    opn    rt1    rt5    p50    p90  
    0      0      0.00  0.00  0.00  0.00
```



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

Screenshot:

