

DEVOPS – Task 2

Name: Sudharsun M

Roll No : 22CSR209

1) Installation of Docker:

Code:

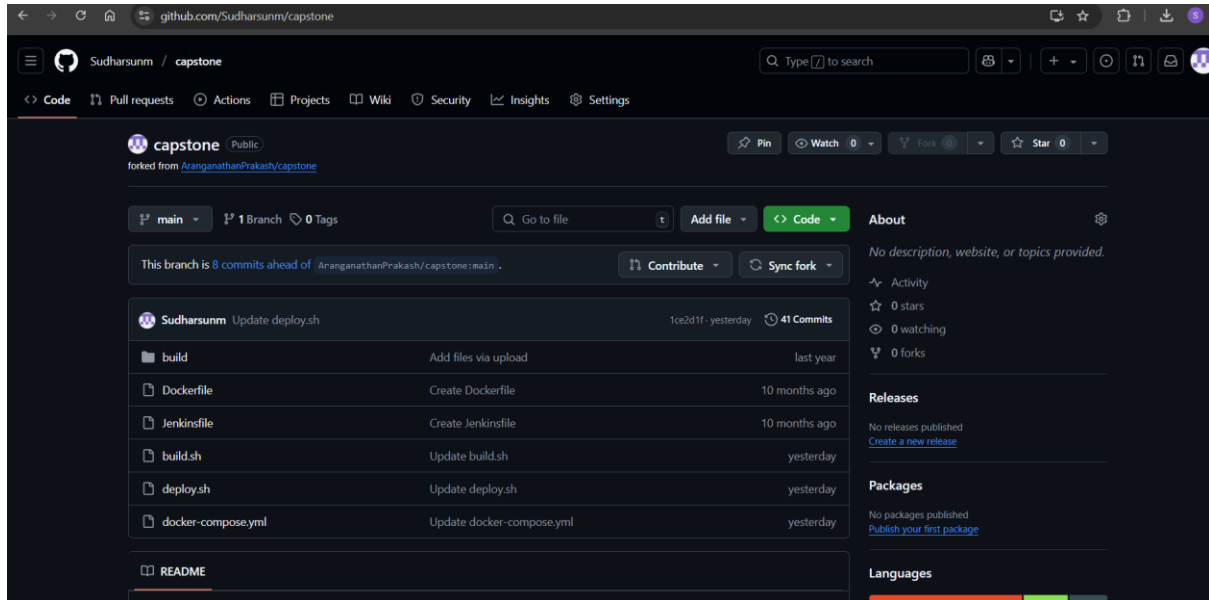
```
sudo apt install docker.io
docker --version
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

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: 30
     Memory: 62.0M (-)
    CGroup: /system.slice/docker.service
            └─ 9561 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
               18253 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 70 -container-ip 172.17.0.2 -com
               18261 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 70 -container-ip 172.17.0.2 -com
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)
```

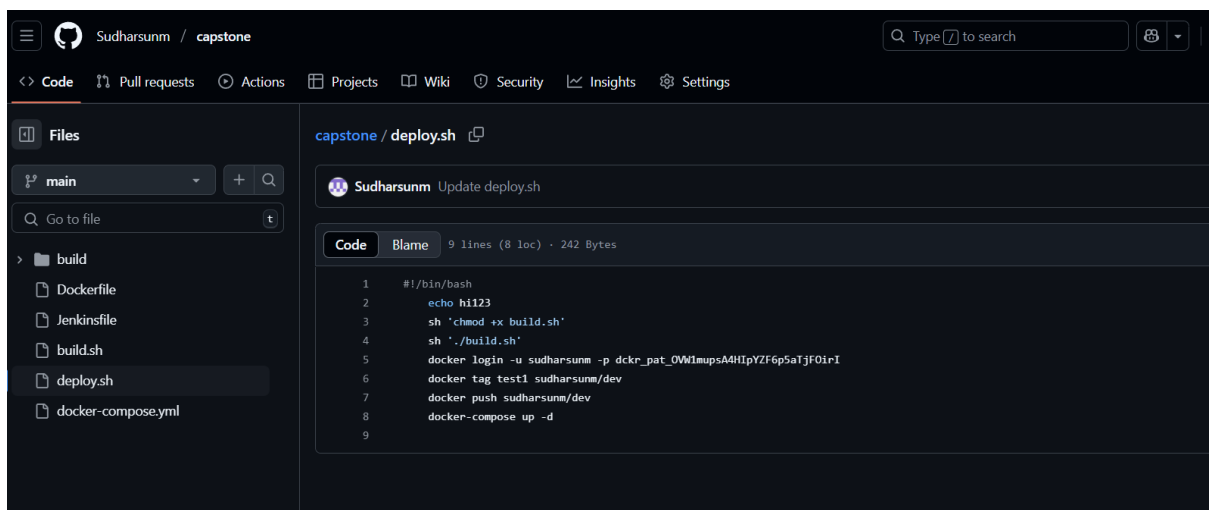
2) Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

Screenshot:



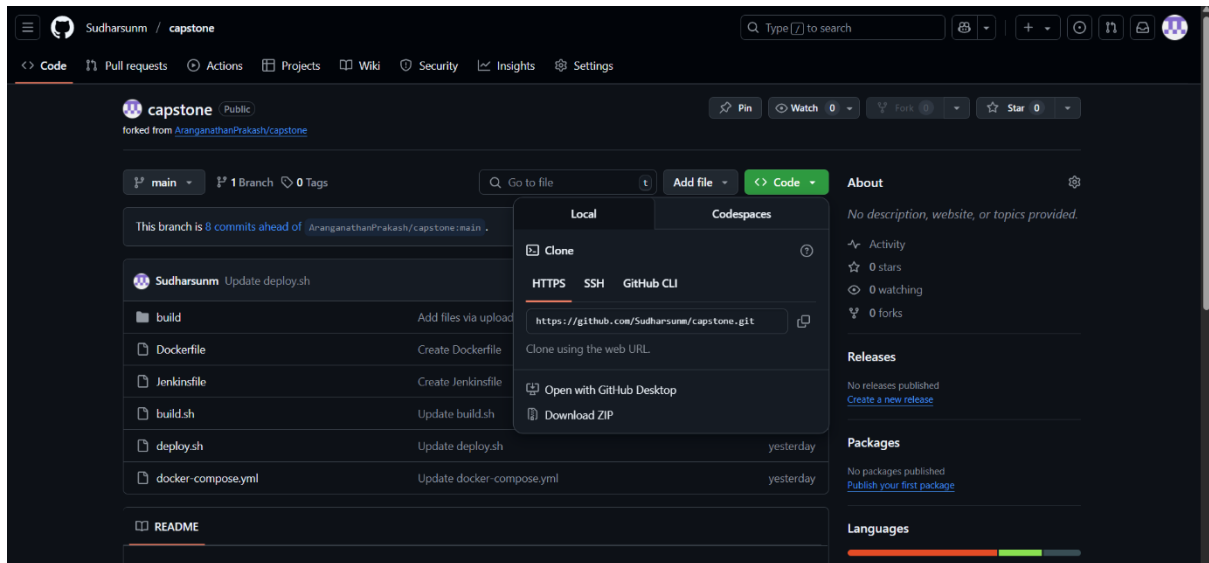
3) Then change the token and repo name of the docker Hub in the deploy.sh file which is in our repository.

Screenshot



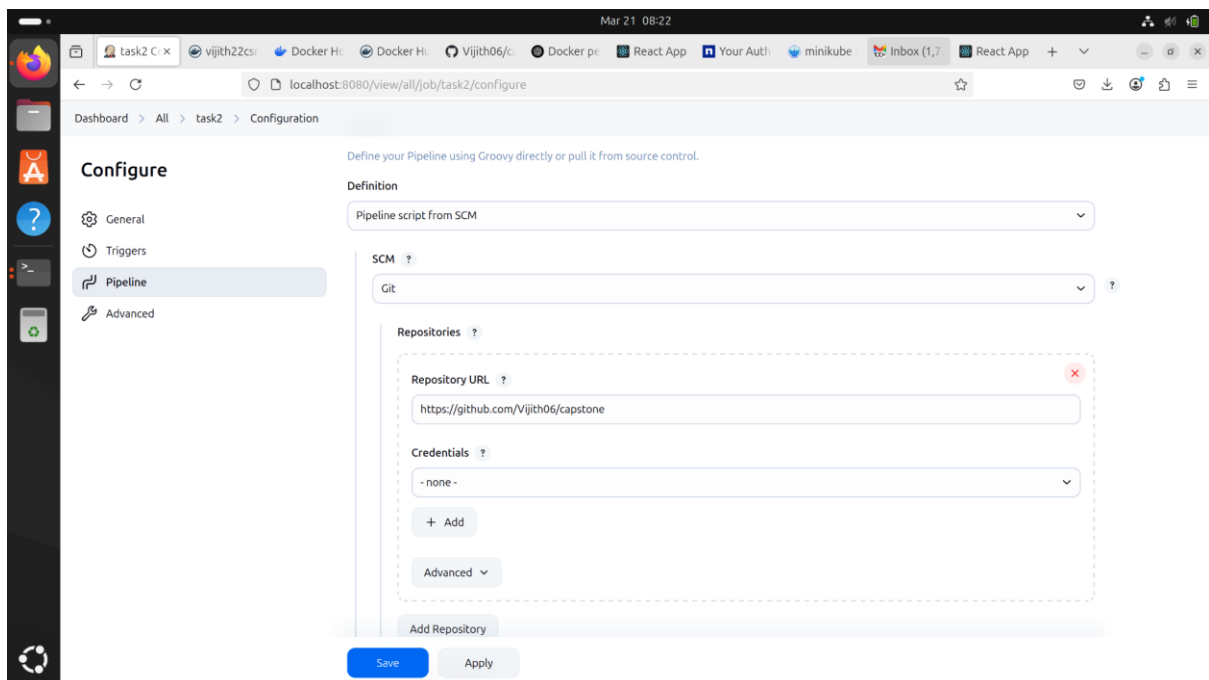
4) Then copy the GitHub link of the repository and go to Jenkins.

Screenshot:

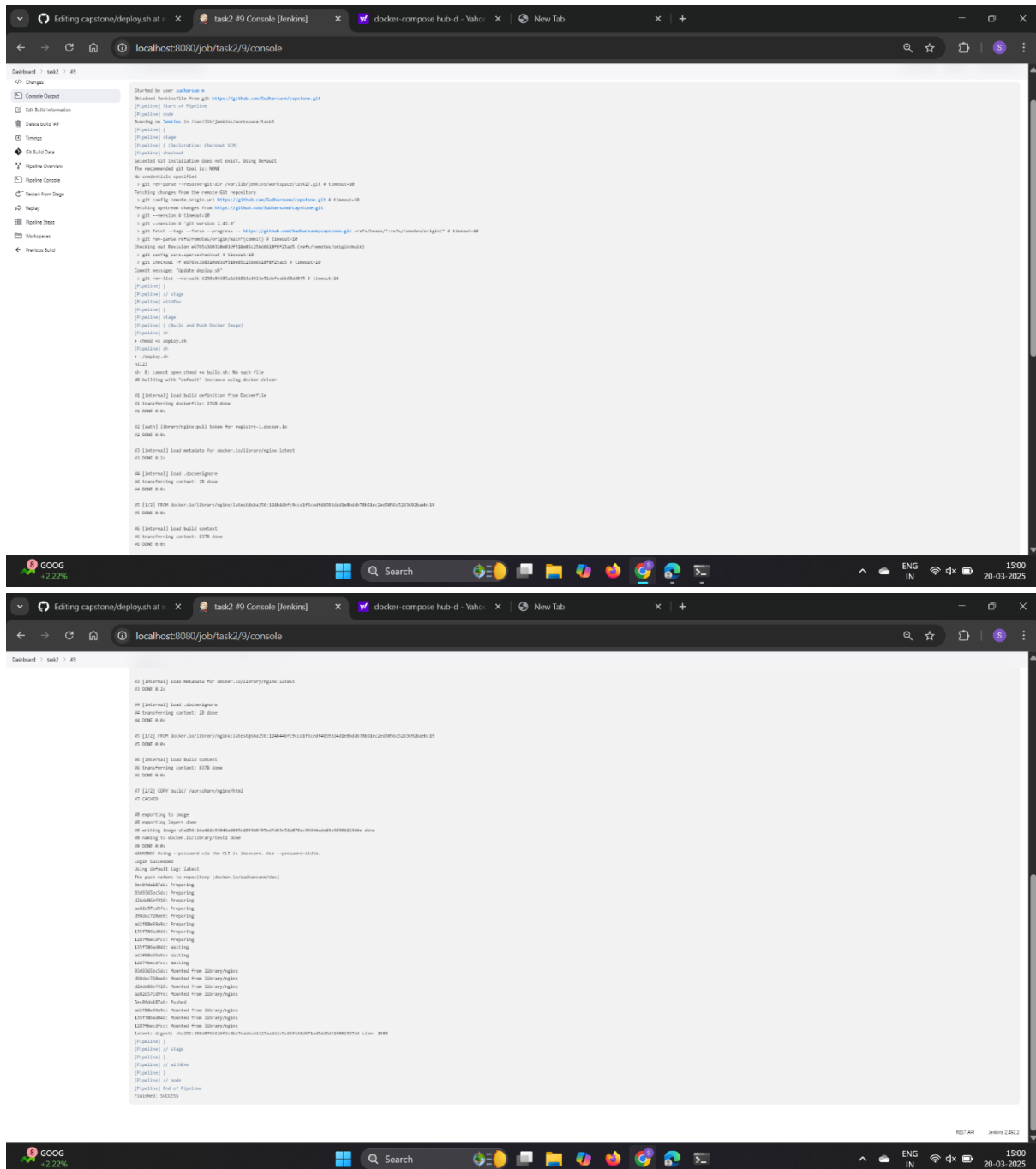


5) In Jenkins, create a new item (Job) with a type pipeline and add the copied GitHub url to it with the correct branch and Jenkinsfile.

ScreenShot:



Screenshot:



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

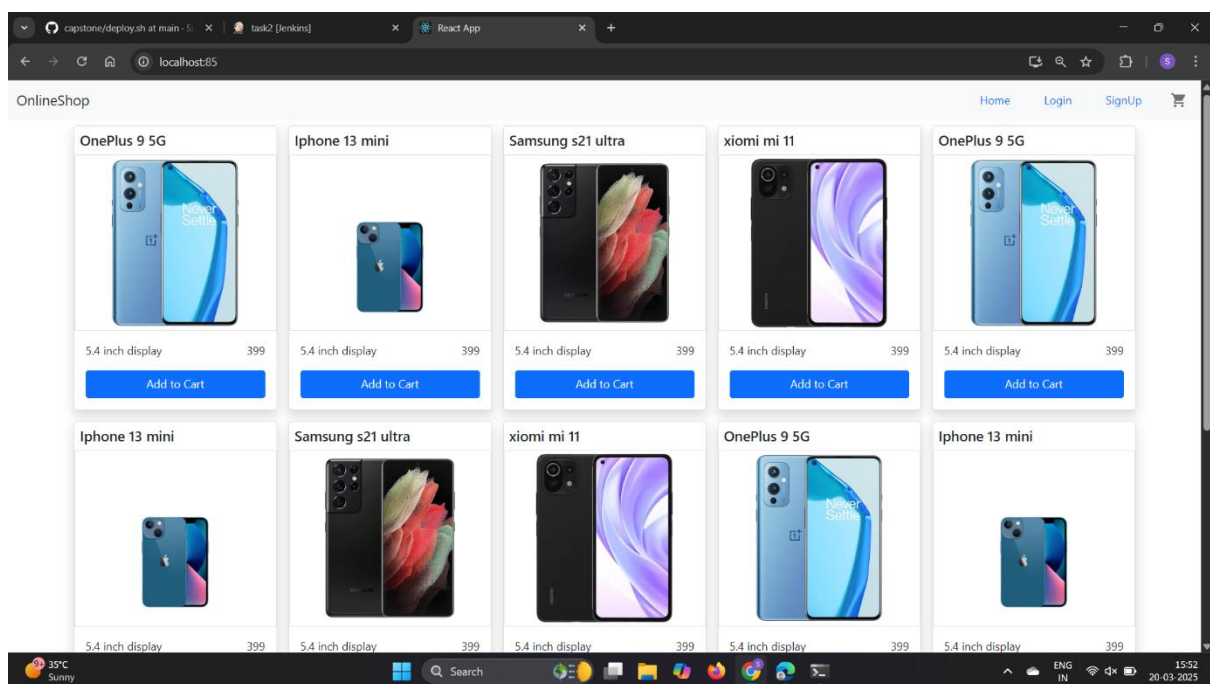
Code:

```
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:

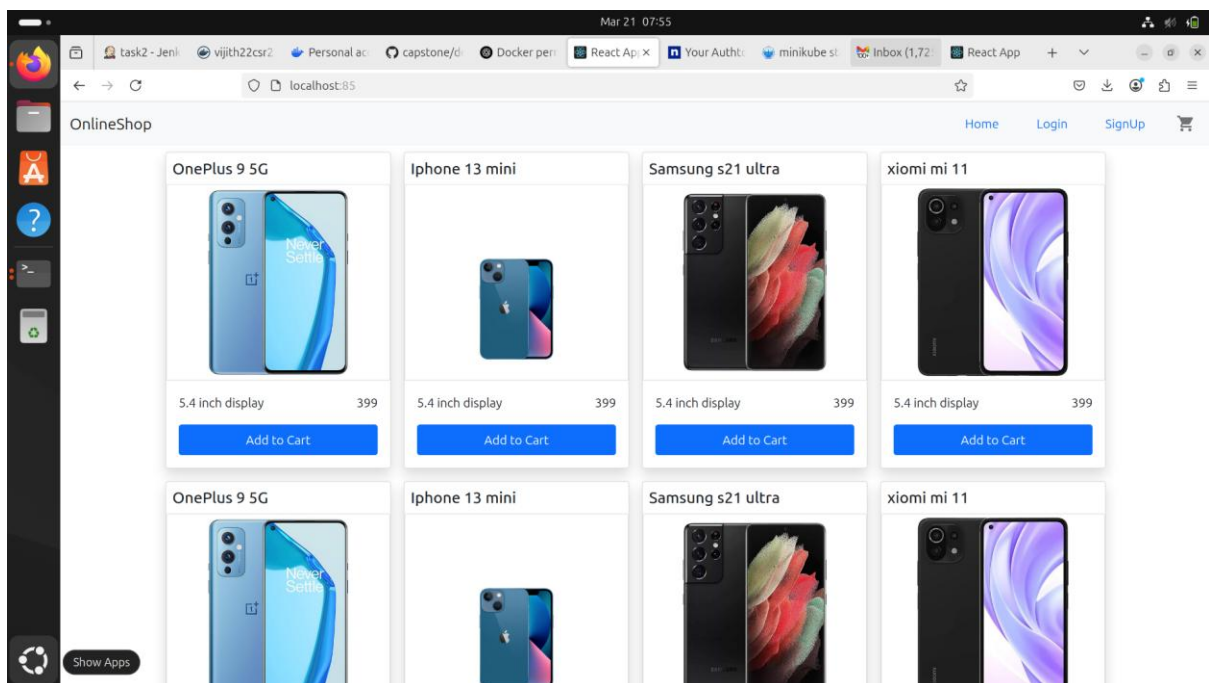


9) But, Instead of running the image by manually, we can also write the command for running in a file called `docker-compose.yml`

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:

```
sudharsun@unknown: ~$ ngrok
ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]

COMMANDS:
  config      update or migrate ngrok's configuration file
  http        start an HTTP tunnel
  tcp         start a TCP tunnel
  tunnel      start a tunnel for use with a tunnel-group backend

EXAMPLES:
  ngrok http 80                                # secure public URL for port 80 web server
  ngrok http --url baz.ngrok.dev 8080          # port 8080 available at baz.ngrok.dev
  ngrok tcp 22                                  # tunnel arbitrary TCP traffic to port 22
  ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com # secure your app with oauth

Paid Features:
  ngrok http 80 --url mydomain.com              # run ngrok with your own custom domain
  ngrok http 80 --cidr-allow 2600:8c00::a03c:91ee:fe69:9695/32 # run ngrok with IP policy restrictions
  Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
  -h, --help      help for ngrok

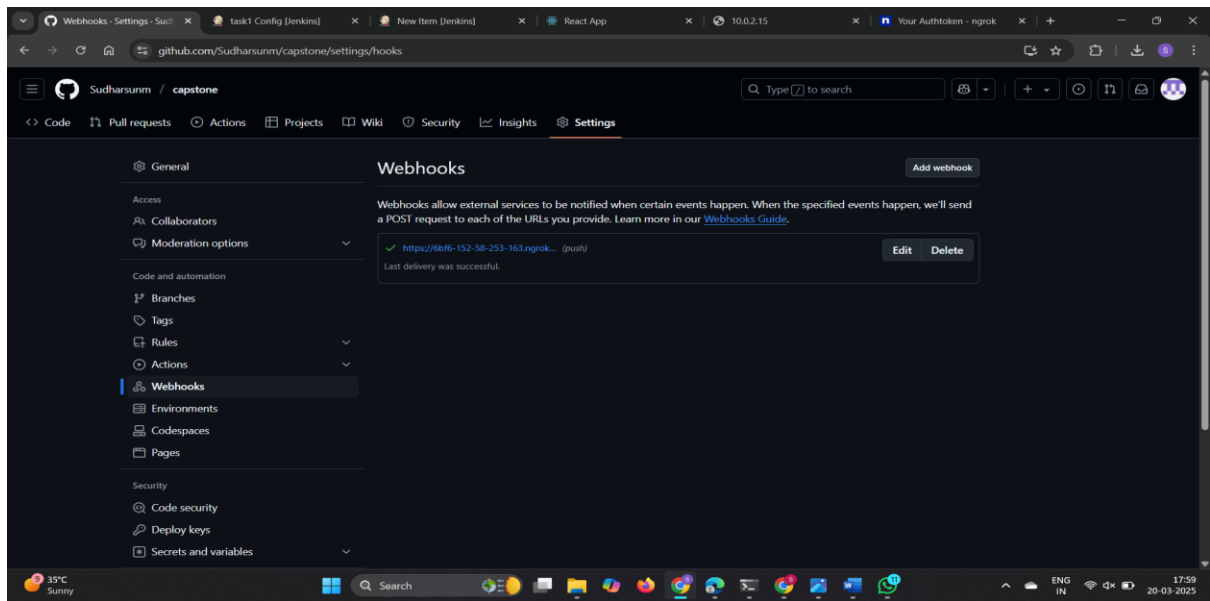
Use "ngrok [command] --help" for more information about a command.
sudharsun@unknown: ~$ ngrok http 8080
sudharsun@unknown: ~$ ngrok config add-authtoken 2ua05VVy5xXy0R71rjuWLWeX1D4_2dcZsbpa9S1dqe1157UzX
Authtoken saved to configuration file: /home/sudharsun/snap/ngrok/260/.config/ngrok/ngrok.yml
sudharsun@unknown: ~$ ngrok http 8080
sudharsun@unknown: ~$
```

```
ngrok
Route traffic by anything: https://ngrok.com/r/iep

Session Status
Account      Sudharsun (Plan: Free)
Version      3.22.0
Region       India (in)
Latency      156ms
Web Interface http://127.0.0.1:4040
Forwarding   https://ad26-103-196-28-178.ngrok-free.app -> http://localhost:8080

Connections
  ttl   opn   rt1   rt5   p50   p90
    9    2    0.09  0.03  0.33  29.31

HTTP Requests
-----
08:54:56.381 UTC POST /widget/BuildQueueWidget/ajax 200 OK
08:54:56.381 UTC POST /widget/ExecutorsWidget/ajax 200 OK
08:54:51.777 UTC POST /widget/BuildQueueWidget/ajax 200 OK
08:54:51.777 UTC POST /widget/ExecutorsWidget/ajax 200 OK
08:54:46.340 UTC POST /widget/BuildQueueWidget/ajax 200 OK
08:54:46.340 UTC POST /widget/ExecutorsWidget/ajax 200 OK
08:54:41.268 UTC GET /i18n/resourceBundle 200 OK
08:54:41.009 UTC GET /adjuncts/d1bc400e/jenkins/management/AdministrativeMonitorsDecorator/resources.css 200 OK
08:54:41.133 UTC GET /adjuncts/d1bc400e/jenkins/management/AdministrativeMonitorsDecorator/resources.js 200 OK
08:54:41.009 UTC GET /static/d1bc400e/jsbundles/pages/dashboard.js 200 OK
```



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

Screenshot:

