

DEVOPS TASK 2

NAME : SNEHITH S K

ROLL NO : 22CSL264

1) Installation of Docker:

Code:

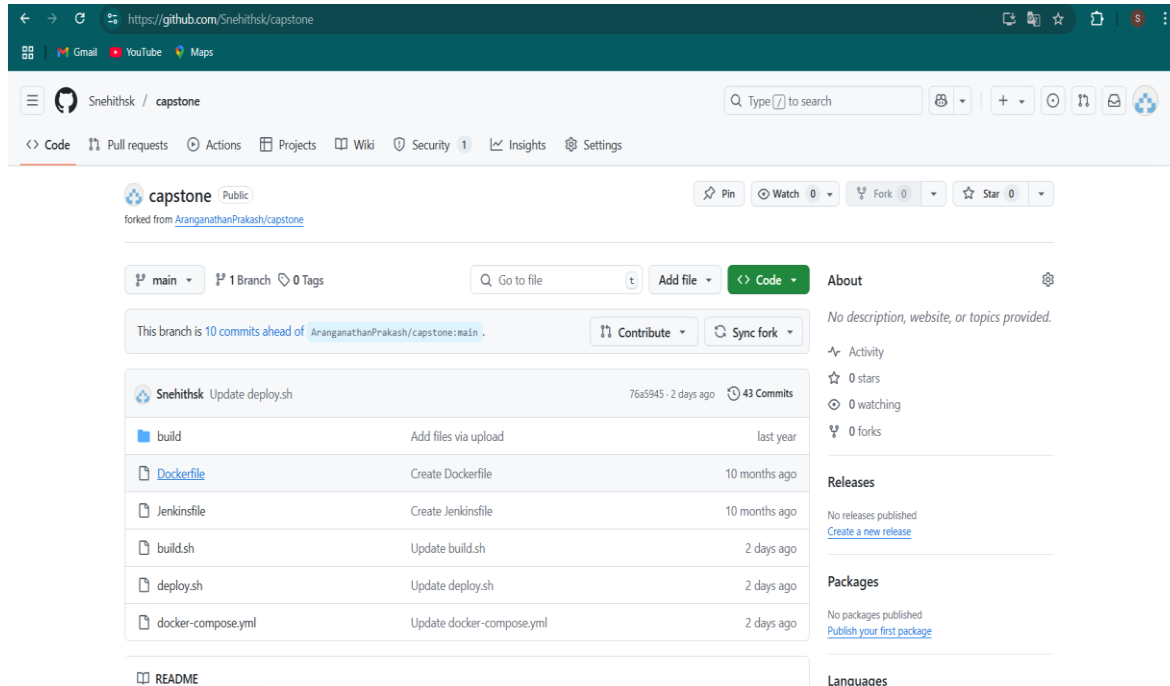
```
sudo apt install docker.io
docker --version
sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker
```

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 libpccaccess0 libpsensors-config libpsensors5
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 -com
              10261 /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="2025-03-20T06:44:32.185697971Z" 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="2025-03-20T06:44:32.185455018Z" 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="2025-03-20T06:44:32.379285869Z" 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)
```

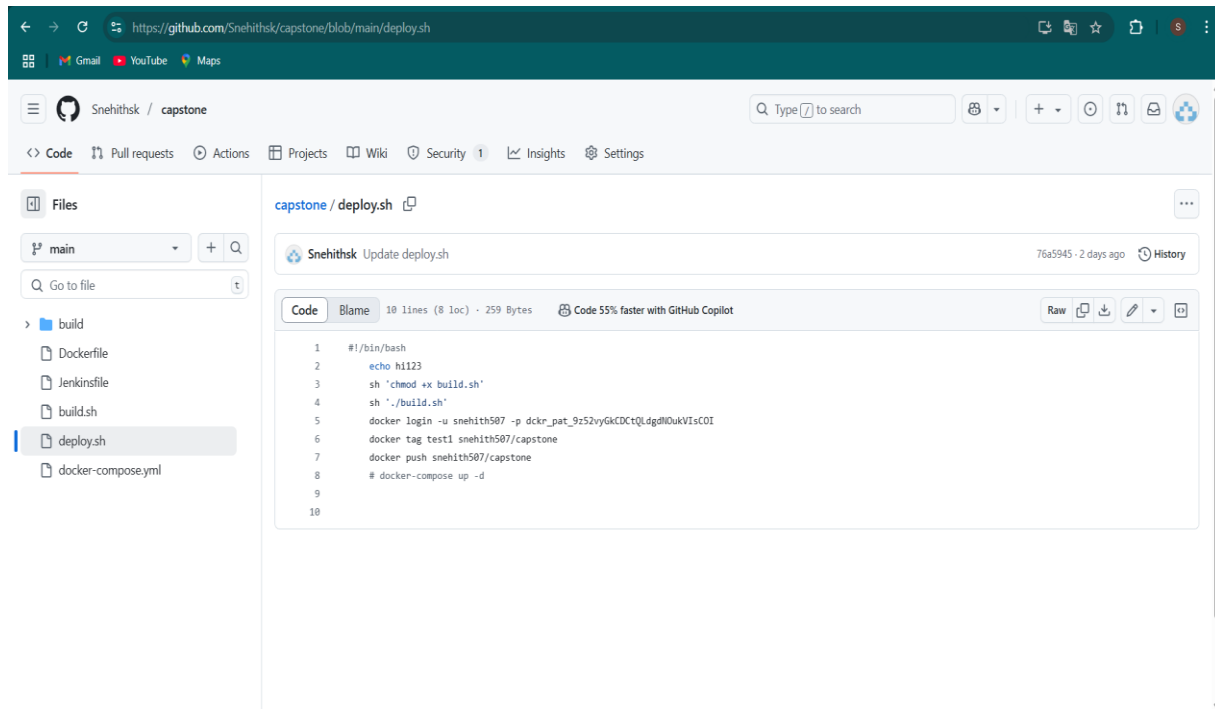
2) Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

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 'Snehithsk / capstone'. The URL in the address bar is 'https://github.com/Snehithsk/capstone/blob/main/deploy.sh'. The repository's main page is visible, including the file list on the left and the code viewer for 'deploy.sh'.

Files:

- main
- build
 - Dockerfile
 - Jenkinsfile
 - build.sh
 - deploy.sh
 - docker-compose.yml

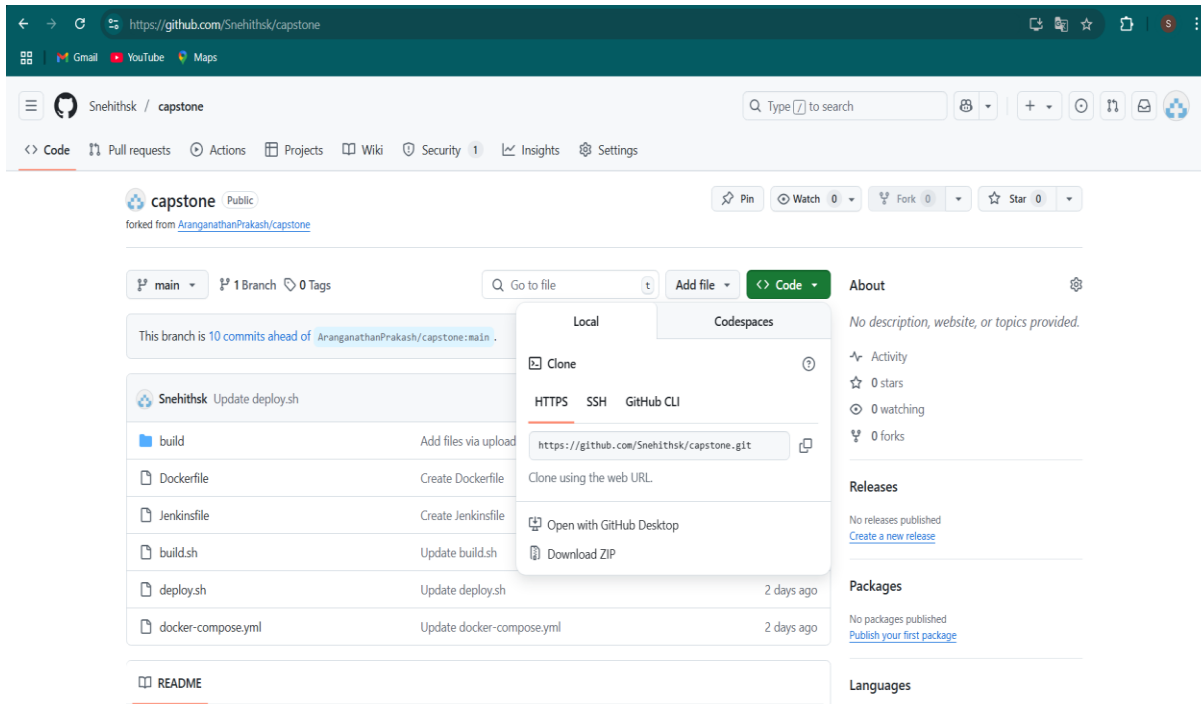
capstone / deploy.sh

Snehithsk Update deploy.sh 76a5945 · 2 days ago History

Code Blame 10 lines (8 loc) · 259 Bytes Code 55% faster with GitHub Copilot Raw Copy Edit

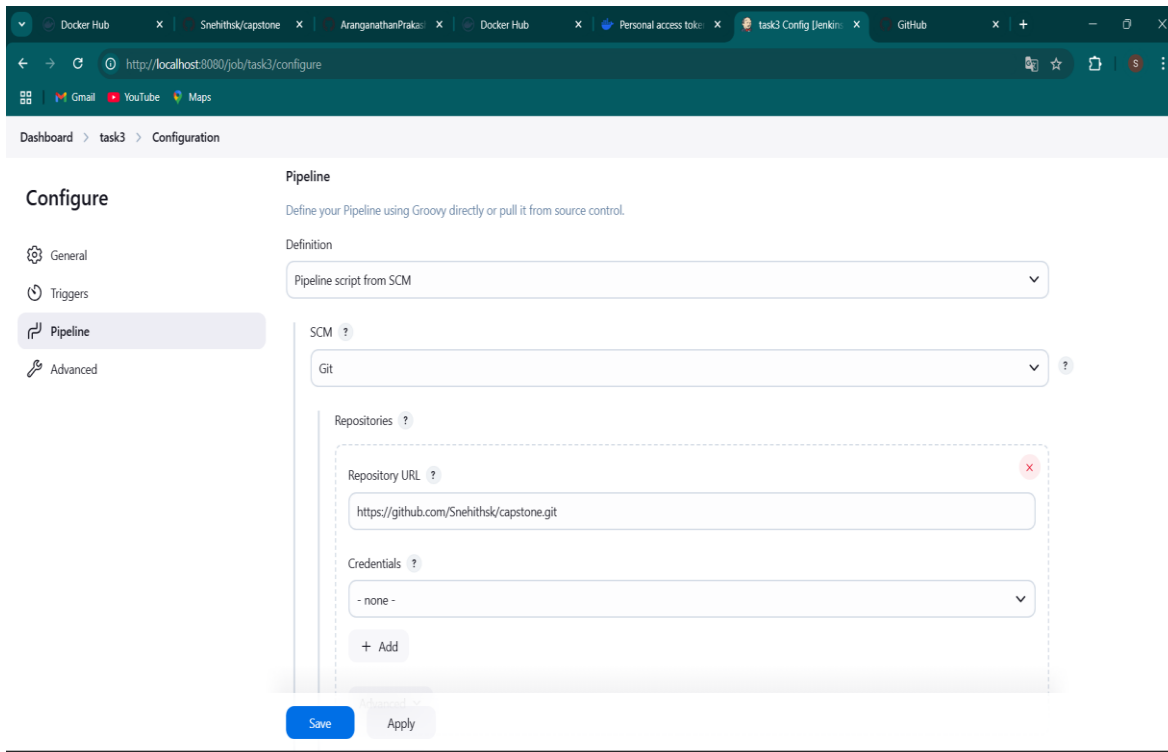
```
1 #!/bin/bash
2 echo hi123
3 sh 'chmod +x build.sh'
4 sh './build.sh'
5 docker login -u snehith507 -p dckr_pat_9z52vyGkCDctQLdgdllDuKVIscOI
6 docker tag test1 snehith507/capstone
7 docker push snehith507/capstone
8 # docker-compose up -d
9
10
```

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.

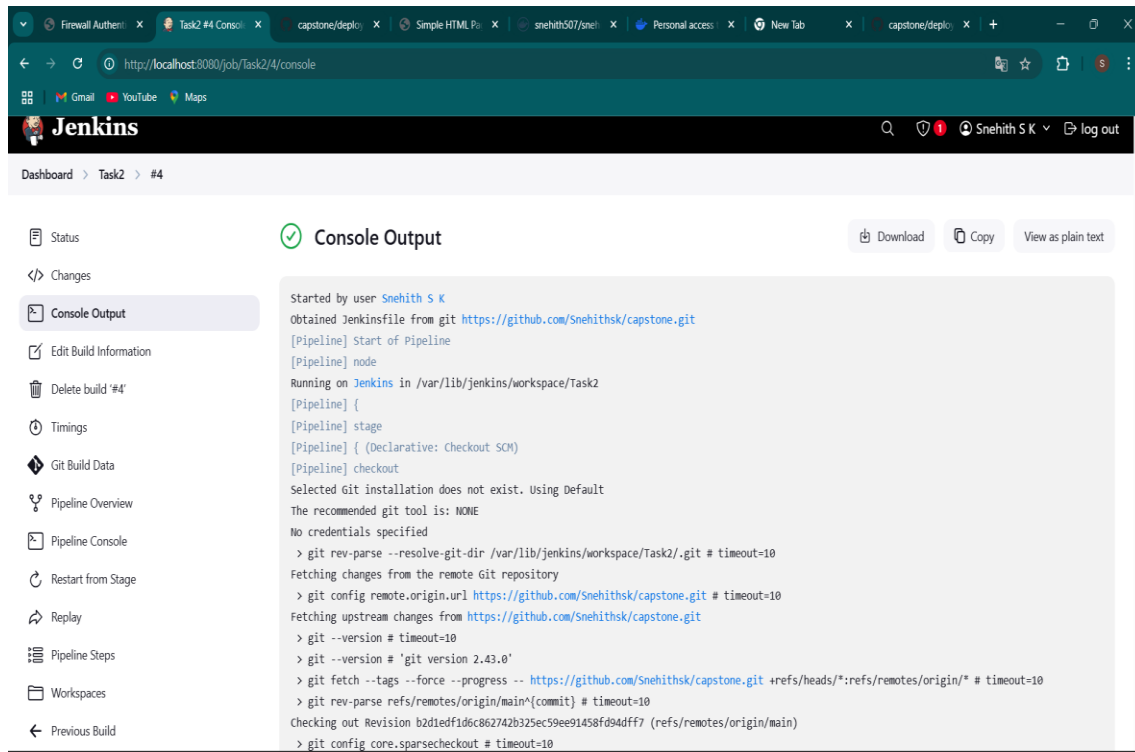
ScreenShot:



The screenshot shows the Jenkins web interface in a browser. The address bar displays `http://localhost:8080/job/task3/configure`. The breadcrumb navigation at the top reads `Dashboard > task3 > Configuration`. On the left sidebar, the `Configure` section is active, with sub-items: `General`, `Triggers`, `Pipeline` (highlighted), and `Advanced`. The main content area is titled `Pipeline` and includes the instruction: `Define your Pipeline using Groovy directly or pull it from source control.` Under the `Definition` section, the `Pipeline script from SCM` option is selected. Below this, the `SCM` dropdown is set to `Git`. The `Repositories` section contains one entry with the `Repository URL` `https://github.com/Snehithsk/capstone.git`. The `Credentials` dropdown is set to `- none -`. At the bottom of the configuration area, there are `Save` and `Apply` buttons.

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.

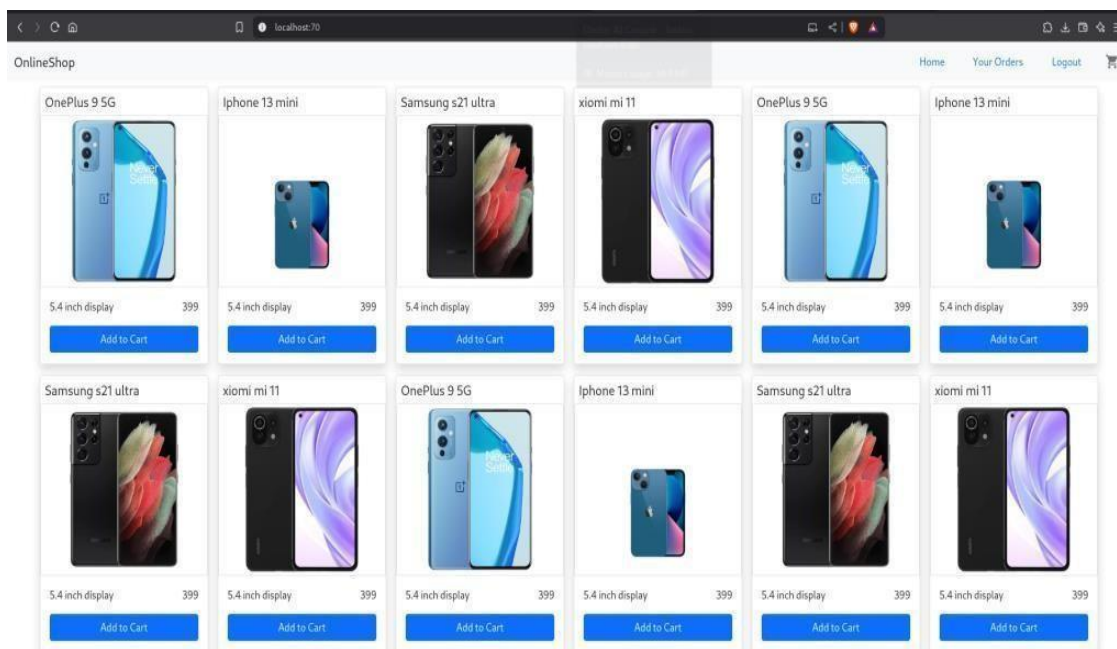
Code:

```
docker images
```

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

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

Screenshot:

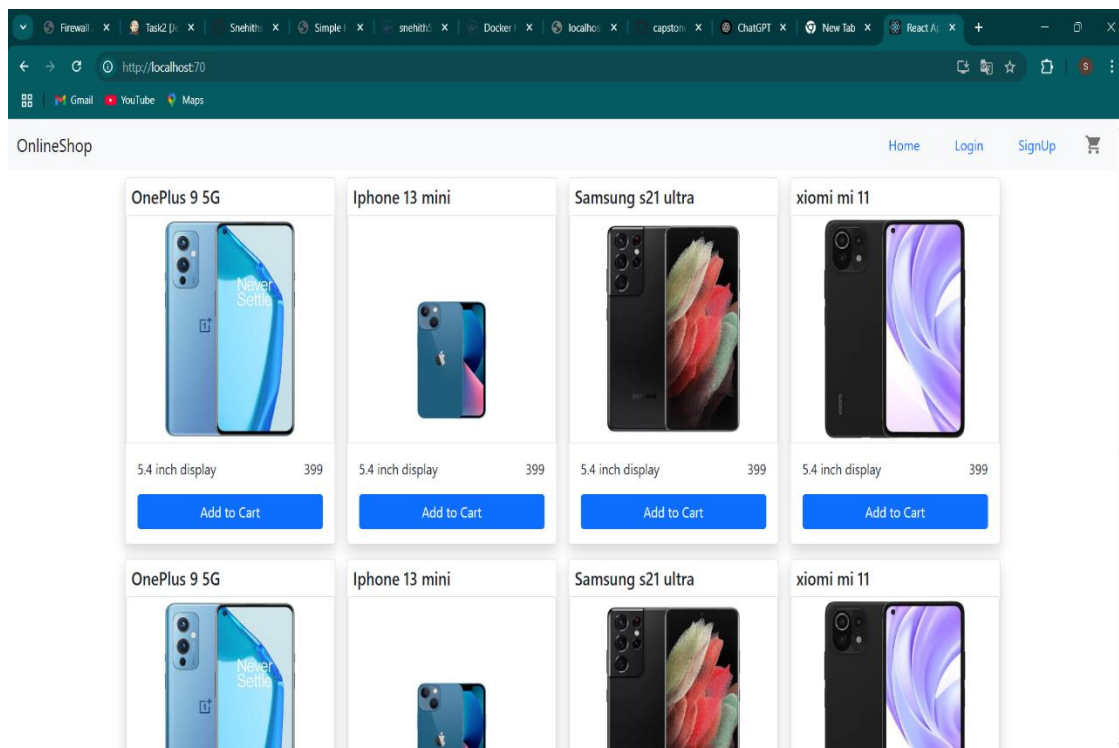


9) But, Instead of running the image by manually , we can also write the command for running in a file called docker-compose.yml

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:


```
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 2ua80vQJhkuuFczLz5uH28DyS4_SgRxeW7boXsGwcd9wt
Auth token saved to configuration file: /home/suganth/snap/ngrok/255/.config/ngrok/ngrok.yml
suganth@suganth-debian:~$ ngrok http 8080
suganth@suganth-debian:~$
```

```
snehith@Snehith: ~  
ngrok (Ctrl+C to quit)  
🔥 Goodbye tunnels, hello Agent Endpoints: https://ngrok.com/r/aep  
  
Session Status      online  
Account             SNEHITH S K (Plan: Free)  
Version             3.21.0  
Region              India (in)  
Latency              52ms  
Web Interface        http://127.0.0.1:4040  
Forwarding           https://f707-103-196-28-174.ngrok-free.app -> http://localhost:8080  
  
Connections          ttl    opn    rt1    rt5    p50    p90  
1                0      0.00   0.00   30.54  30.54  
  
HTTP Requests  
-----  
12:14:56.694 UTC POST /github-webhook/      200 OK
```

localhostTasks1 Config (J)Webhooks - SetiSent Mail - snehFirewall AuthentiDashboard (JenDocker Hubhow to install (k) + -

https://github.com/Snehithsk/capstone/settings/hooks

GmailYouTubeMaps

Snehithsk / capstoneType (Z) to search

<> CodePull requestsActionsProjectsWikiSecurity1InsightsSettings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓https://f707-103-196-28-174.ngrok-... (push)

EditDelete

Last delivery was successful.

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

Screenshot:

