**DEVOPS TASK 2**

**1)**Installation of Docker:

**Code**:

sudo apt install docker.io

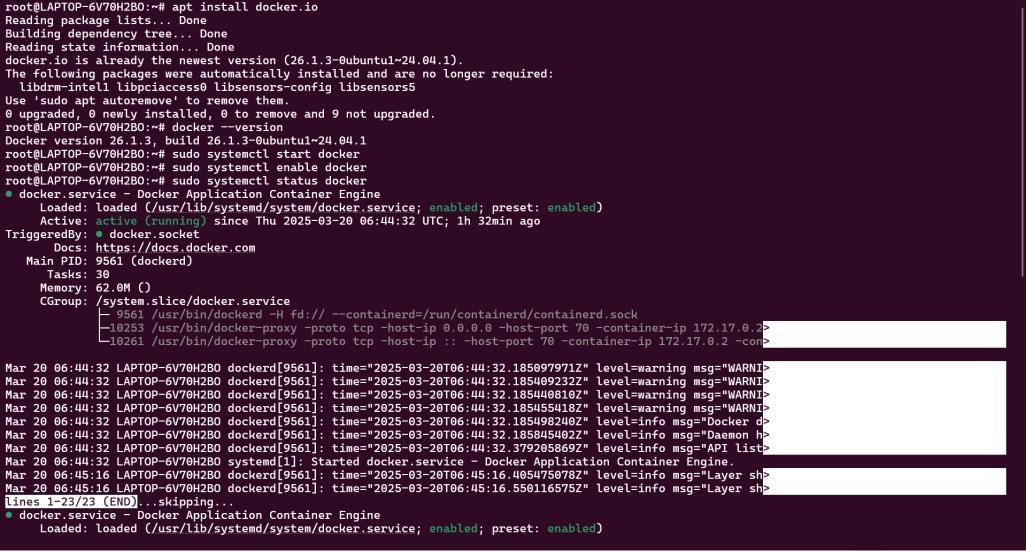
docker –version

sudo systemctl start docker

sudo systemctl enable docker

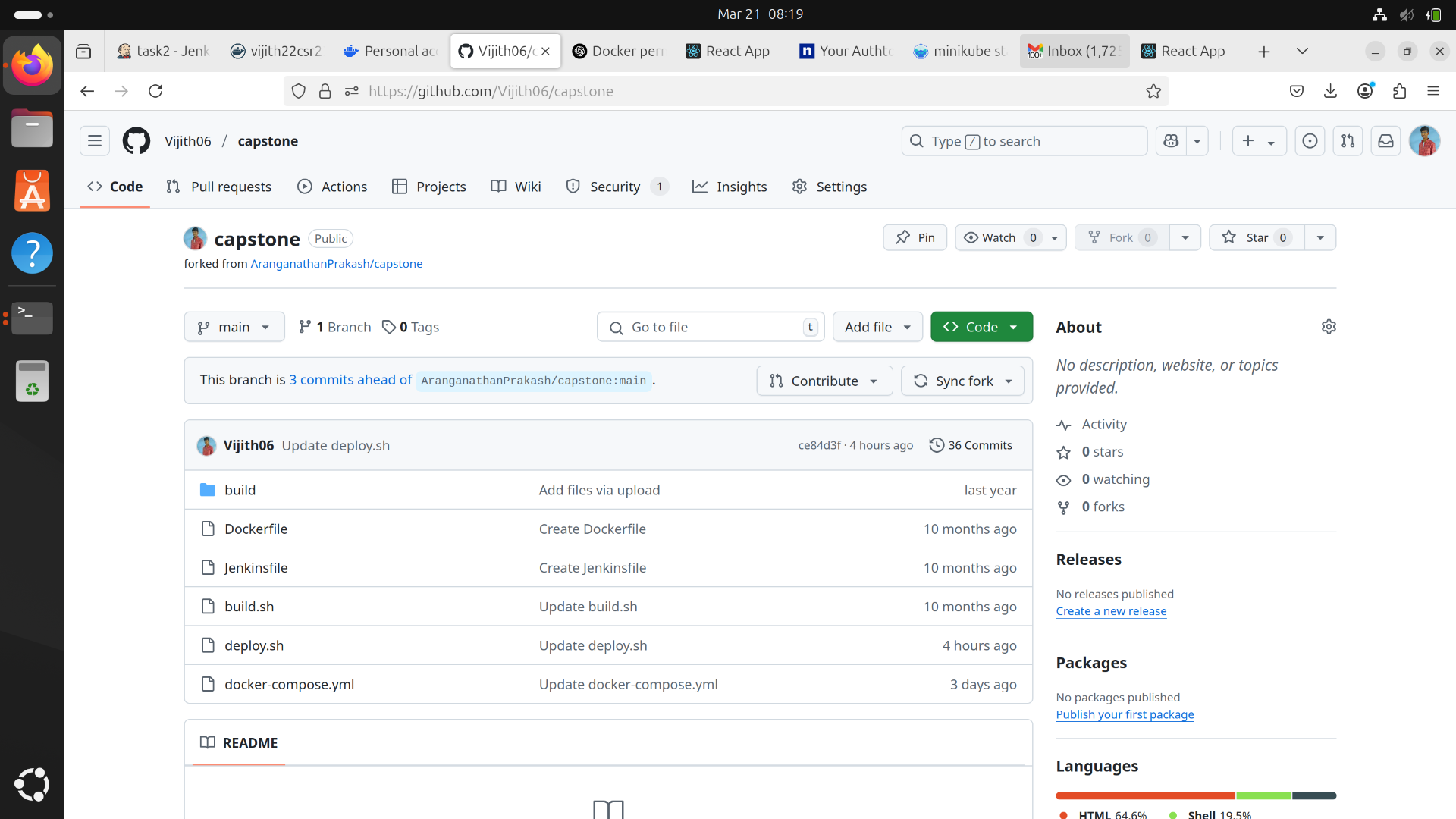
sudo systemctl status docker

**Screenshot**:



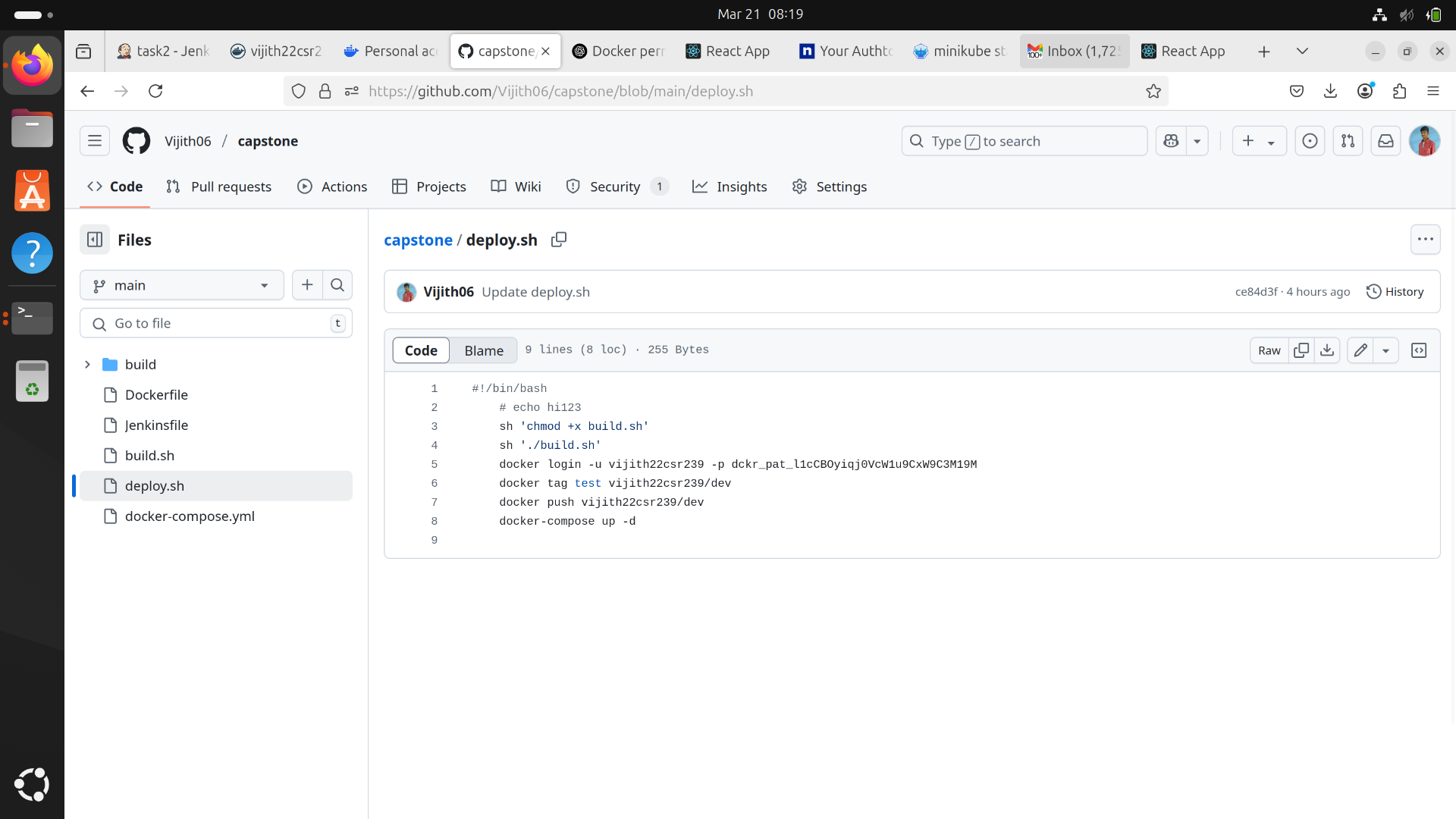
**2)**Fork a copy of a GitHub repo which contains the necessary files which will result in the clone of that repo in our own repository

**Screenshot:**

****

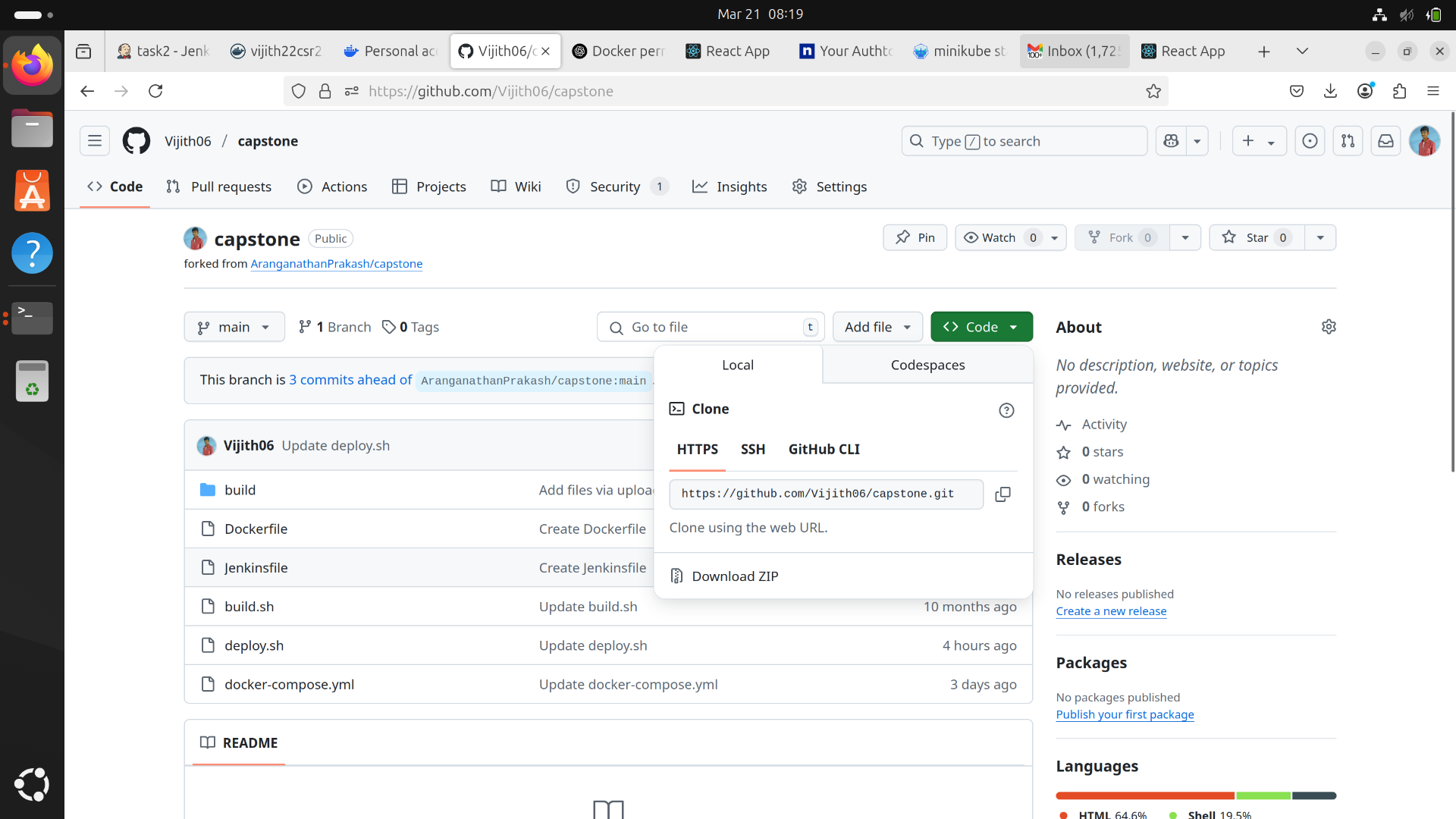
**3)**Then change the token and repo name of the docker Hub in the deploy.sh file which is in our repository.

**Screenshot**

****

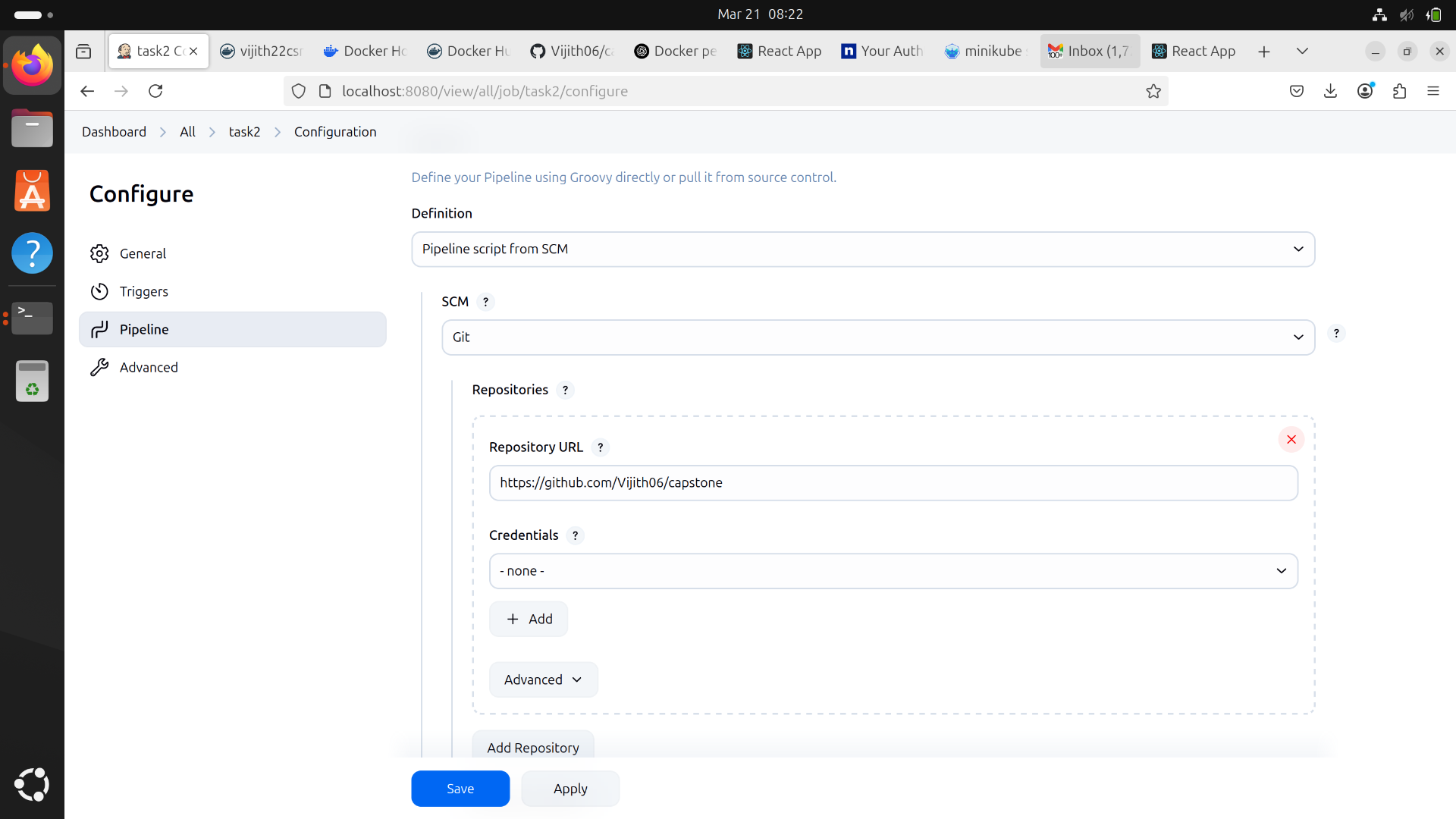
**4)**Then copy the GitHub link of the repository and go to Jenkins.

**Screenshot:**

****

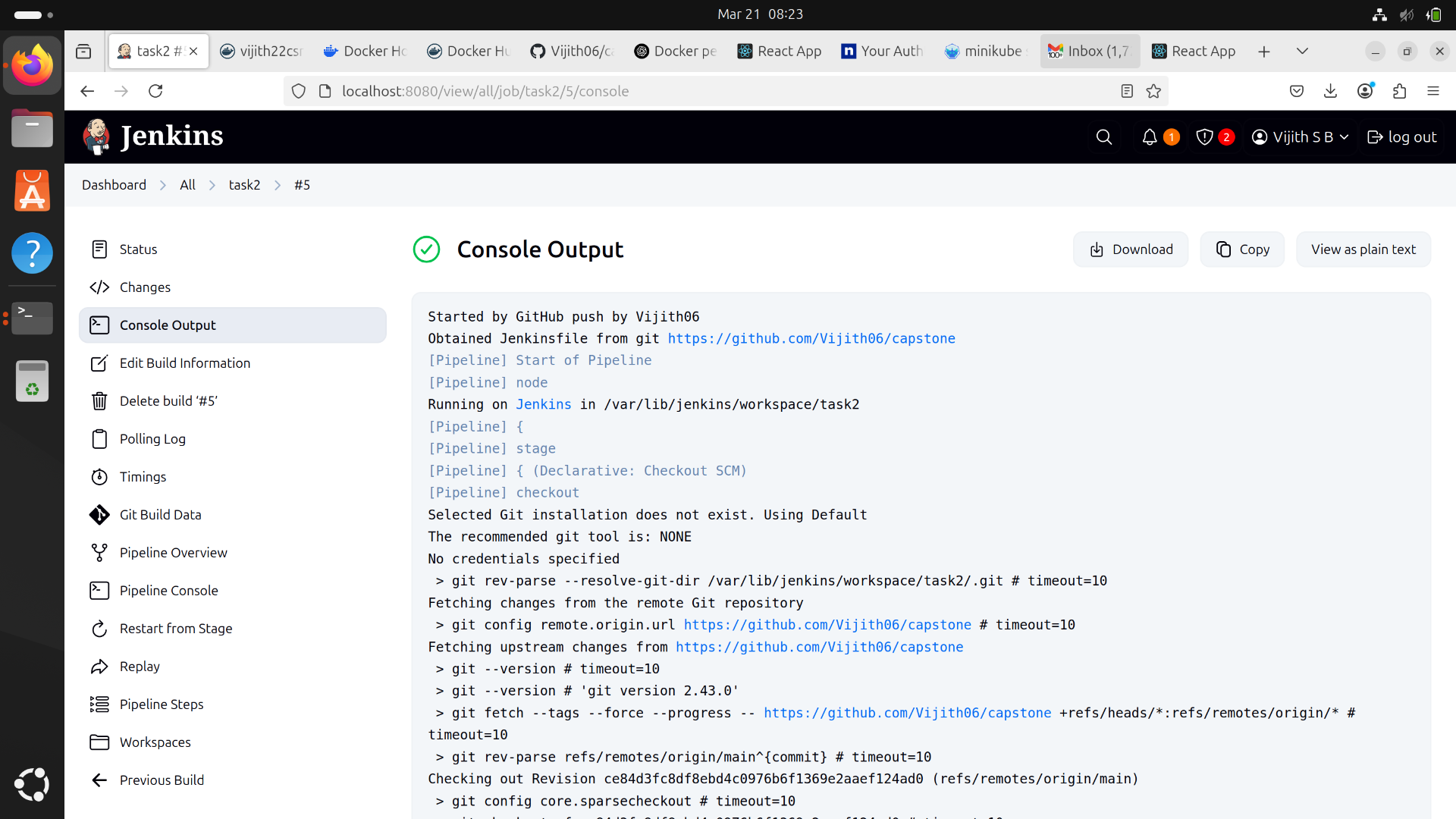
**5)**In Jenkins, create a new item (Job) with a type pipeline and add the copied GitHub url to it with the correct branch and Jenkinsfile.

**ScreenShot:**

****

**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.

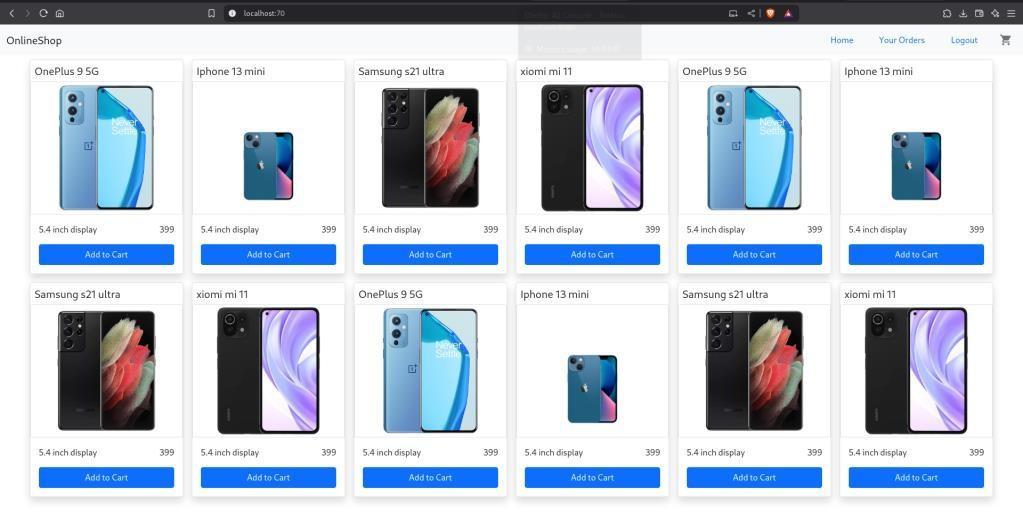
**Code:**

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:**

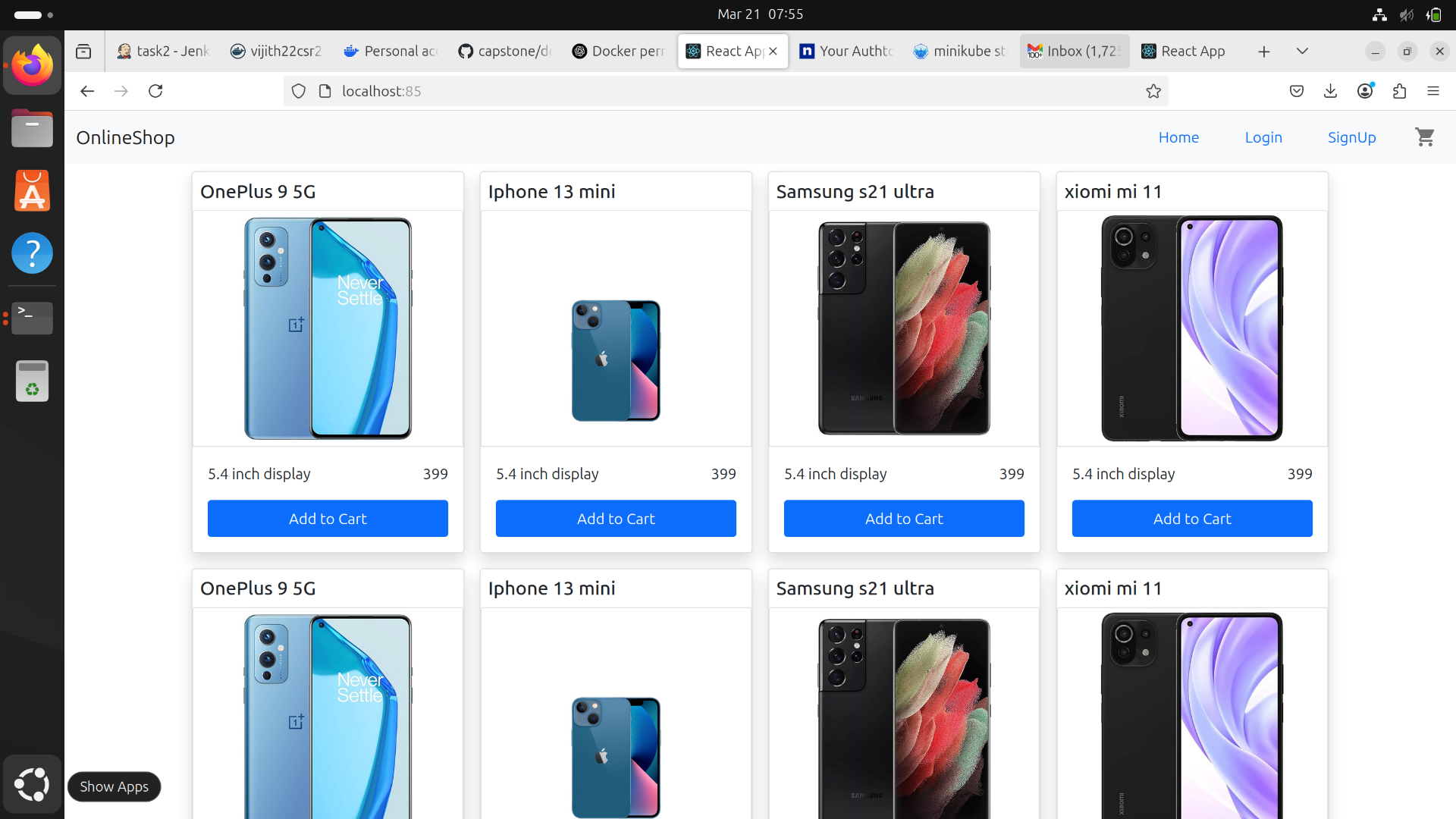
version: '3' services:

react-capstoneimage:"test1"

ports:

- "85:80"

**Screenshot:**

****

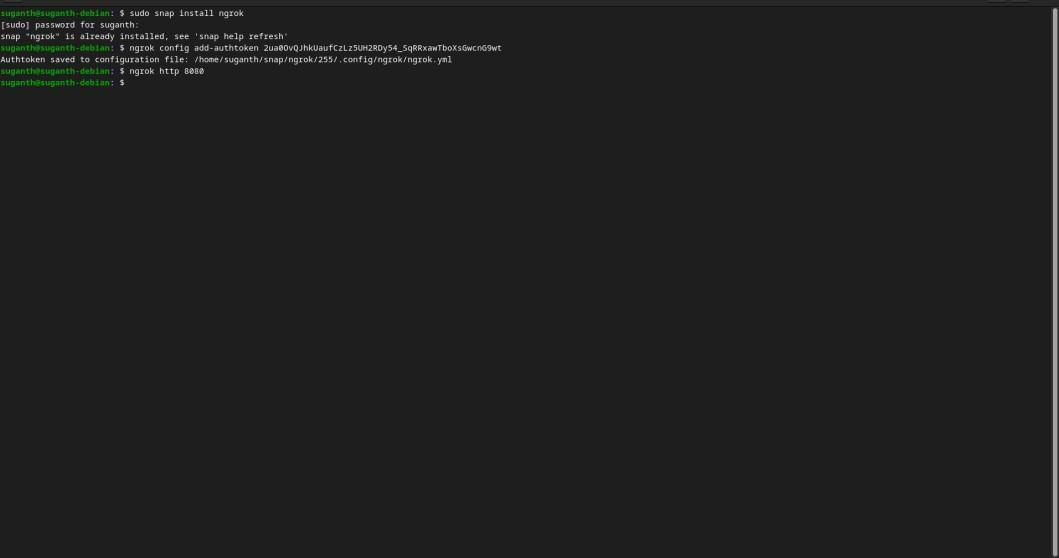
By Creating this, we no need to run the image by manually. (It will automatically run)

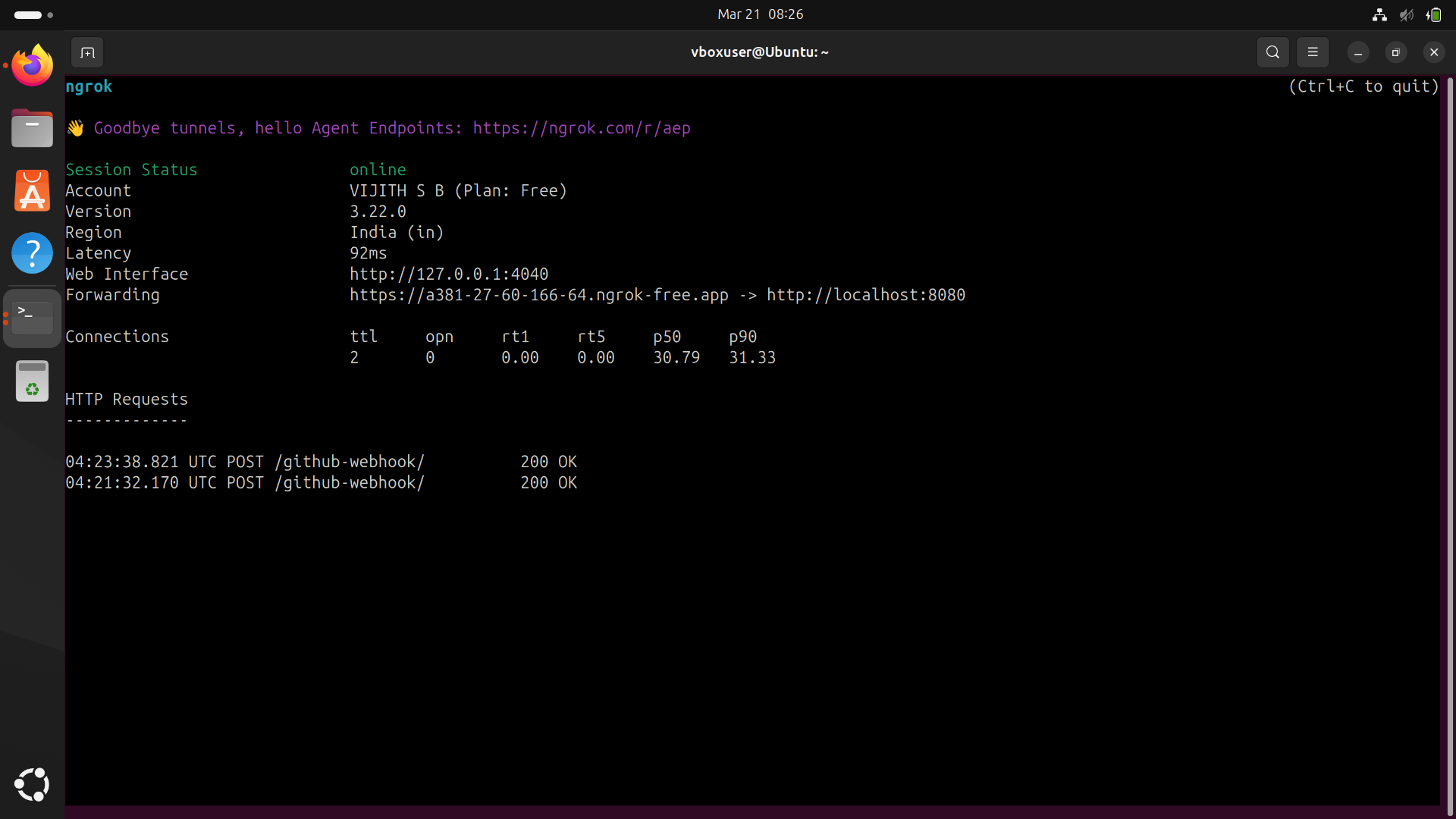
**10)**Adding Webhook to it which is available in GitHub for automatic build of the

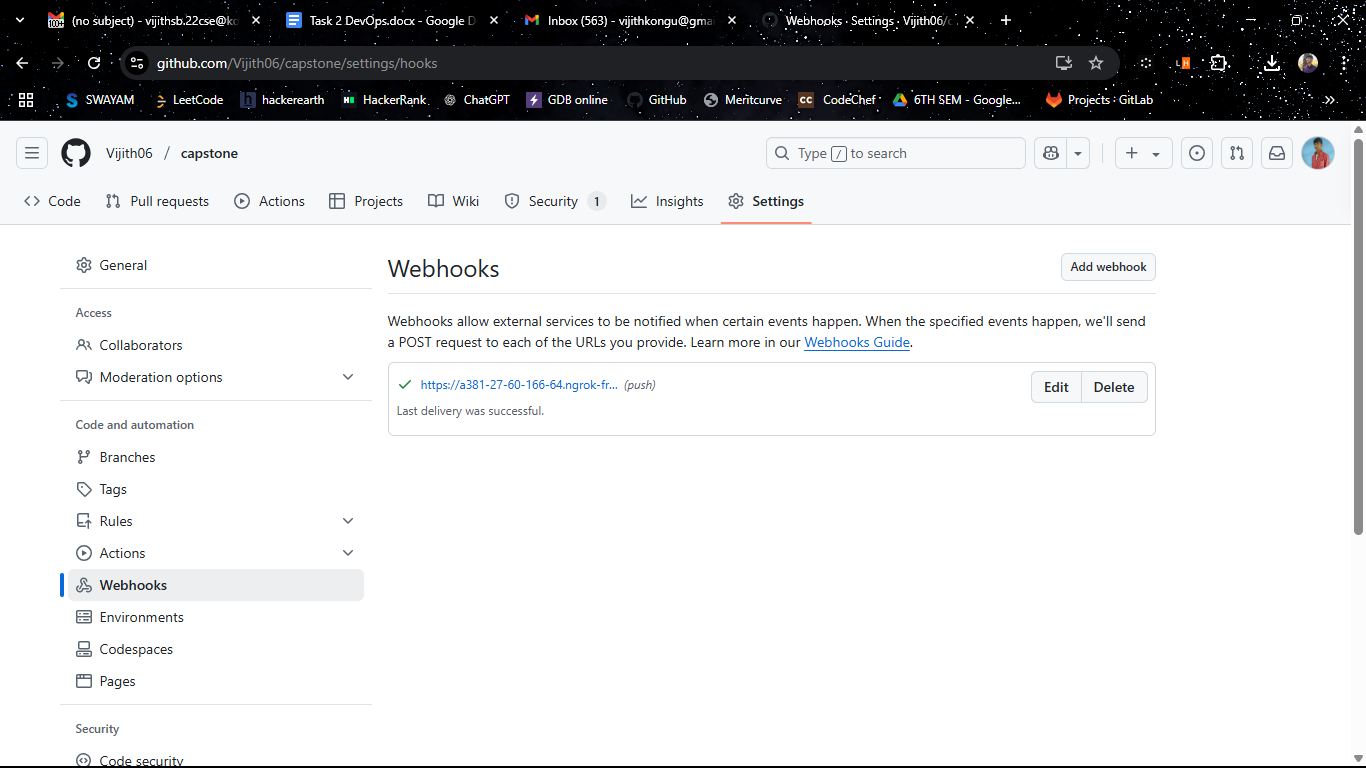
project.

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

**Screenshot:**



****

****

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

**Screenshot:**

