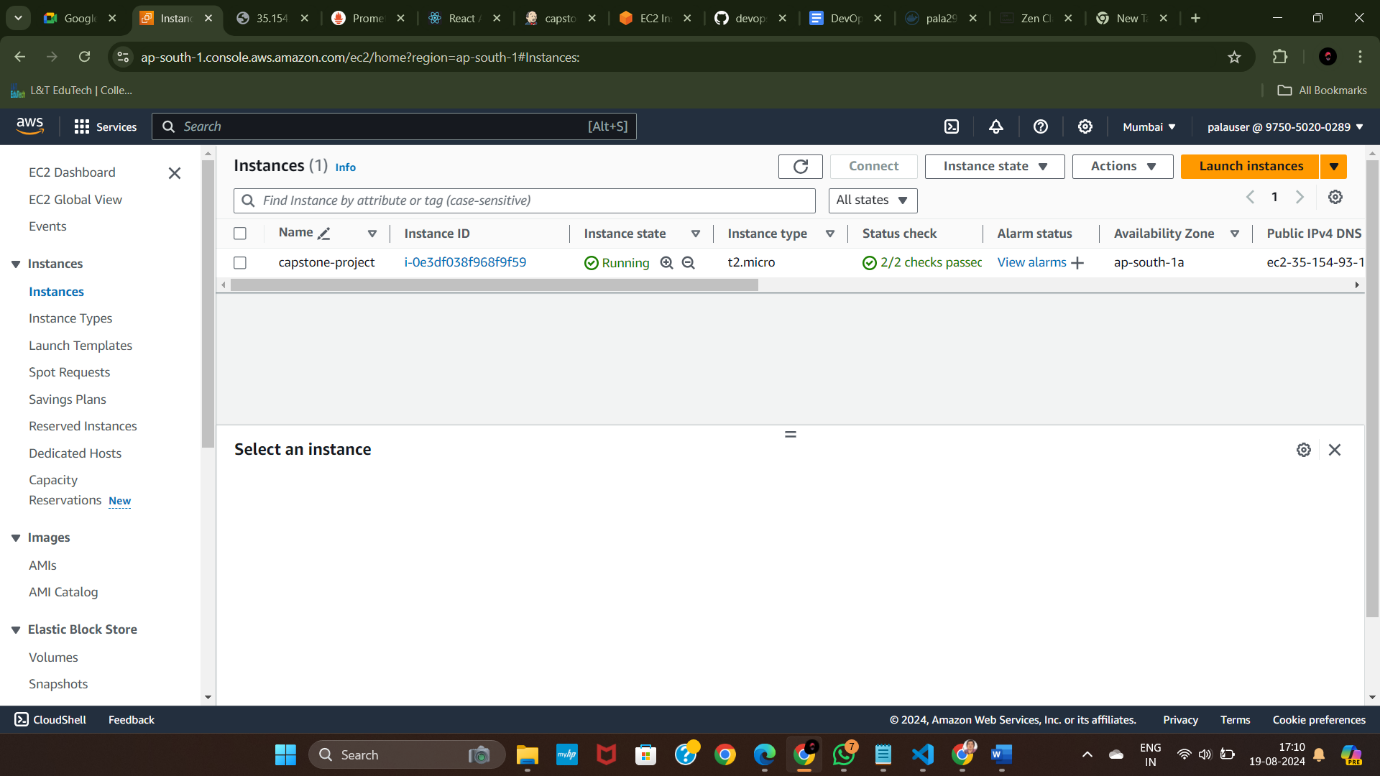
CAPSTONE PROJECT

APPLICATION DEPLOYMENT

1. first create a AWS ec2 instance of t2.micro type.



2. install nginx on your instance,

sudo apt update

sudo apt install nginx -y

sudo systemctl start nginx

sudo systemctl enable nginx

sudo systemctl status nginx

3. next do git clone of your repository,

Git clone <https://github.com/pala26/devops-build.git> [ this is the repository we’re going to deploy.

4. next open that directory using cd command

5. we should create 4 files using vi editor:

(a)Dockerfile

(b)docker-compose.yml

(c)build.sh – to build docker images

(d)deploy.sh – to deploy docker images to your server

6. Version Control Push Code to GitHub:

sudo apt-get update

sudo apt-get install git

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

Go to GitHub and create a new repository.

Copy the repository URL (you'll need it later). cd /path/to/your/project git int git remote add origin https://github.com/your-username/your-repo-name.git

git add .

git commit -m "Your commit message"

git push -u origin master

If you are pushing to a branch other than master, replace master with your branch name.

7. Install docker in your instance

sudo apt update

Install required packages for Docker:

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y

Add Docker’s official GPG key:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Set up the Docker stable repository:

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt install docker-ce -y

sudo systemctl start docker

sudo systemctl enable docker

sudo docker –version

sudo usermod -aG docker $USER

docker login

also install docker compose;

Download the Docker Compose binary:

Replace v2.22.0 with the latest version if needed.

sudo curl -L "https://github.com/docker/compose/releases/download/v2.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Apply executable permissions to the Docker Compose binary:

sudo chmod +x /usr/local/bin/docker-compose

**Verify the installation:**

docker-compose --version

8. Create two repositories: dev (public) and prod (private).

9. install Jenkins:

sudo apt update

sudo apt install openjdk-17-jdk -y

Add the Jenkins repository key to your system:

curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

Add the Jenkins repository to your system:

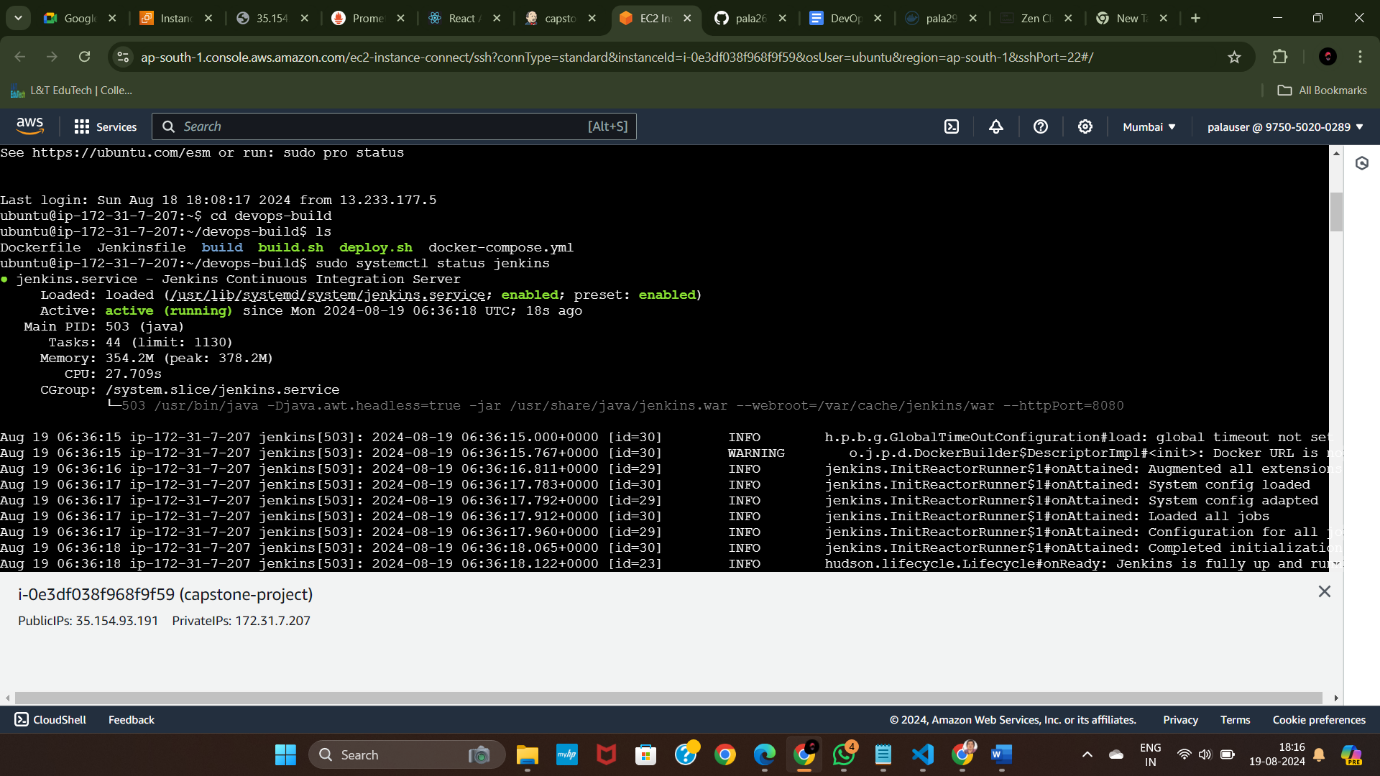
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt update

sudo apt install jenkins -y

sudo systemctl start jenkins

sudo systemctl enable Jenkins



Open your web browser and navigate to http://<your-ec2-public-ip>:8080.

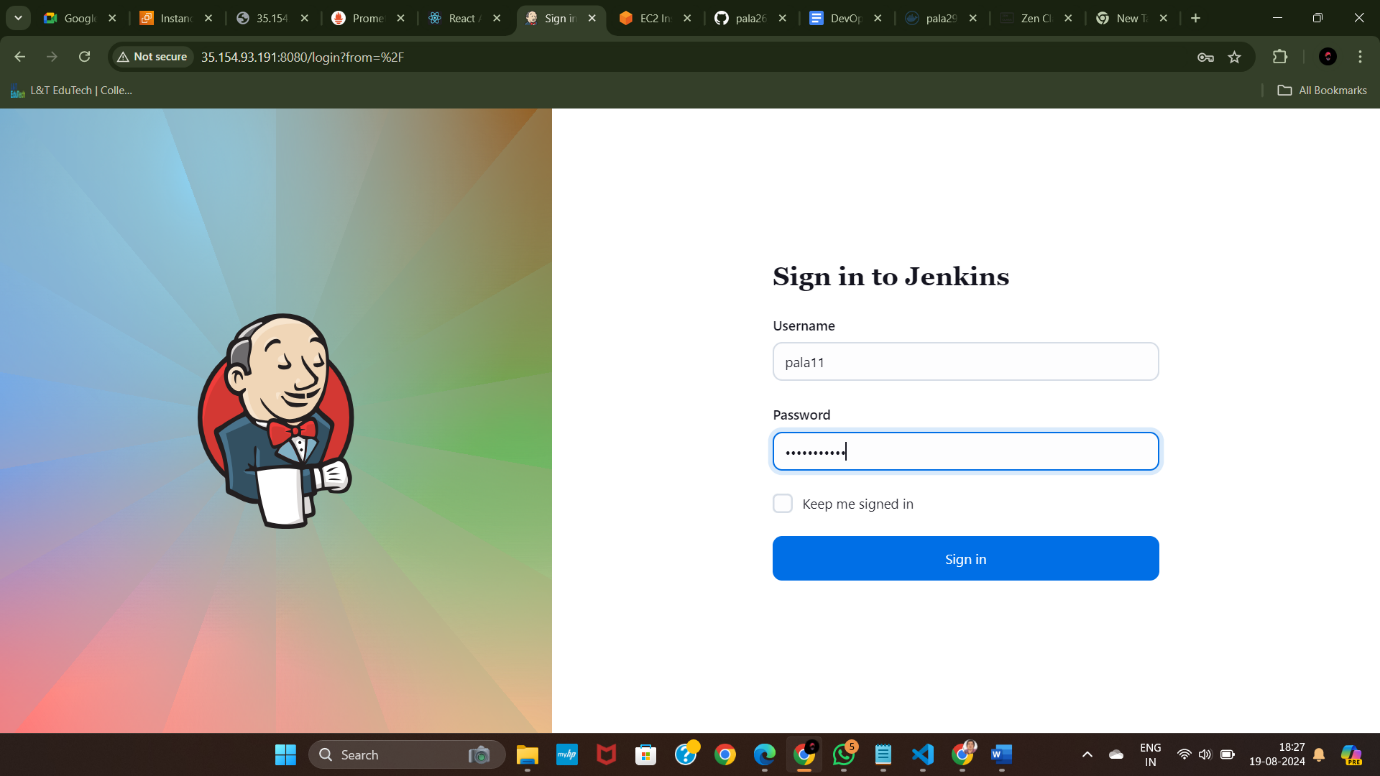
<http://35.154.93.191:8080/> [ my Jenkins url ] for this you should enable port 8080 in your instance security group .

Retrieve the initial Jenkins admin password:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

install the suggested plugins and signup inside Jenkins.

Next install required plugins [ docker,github ]



10. next open manage Jenkins->system configuration->system->environment variables->add

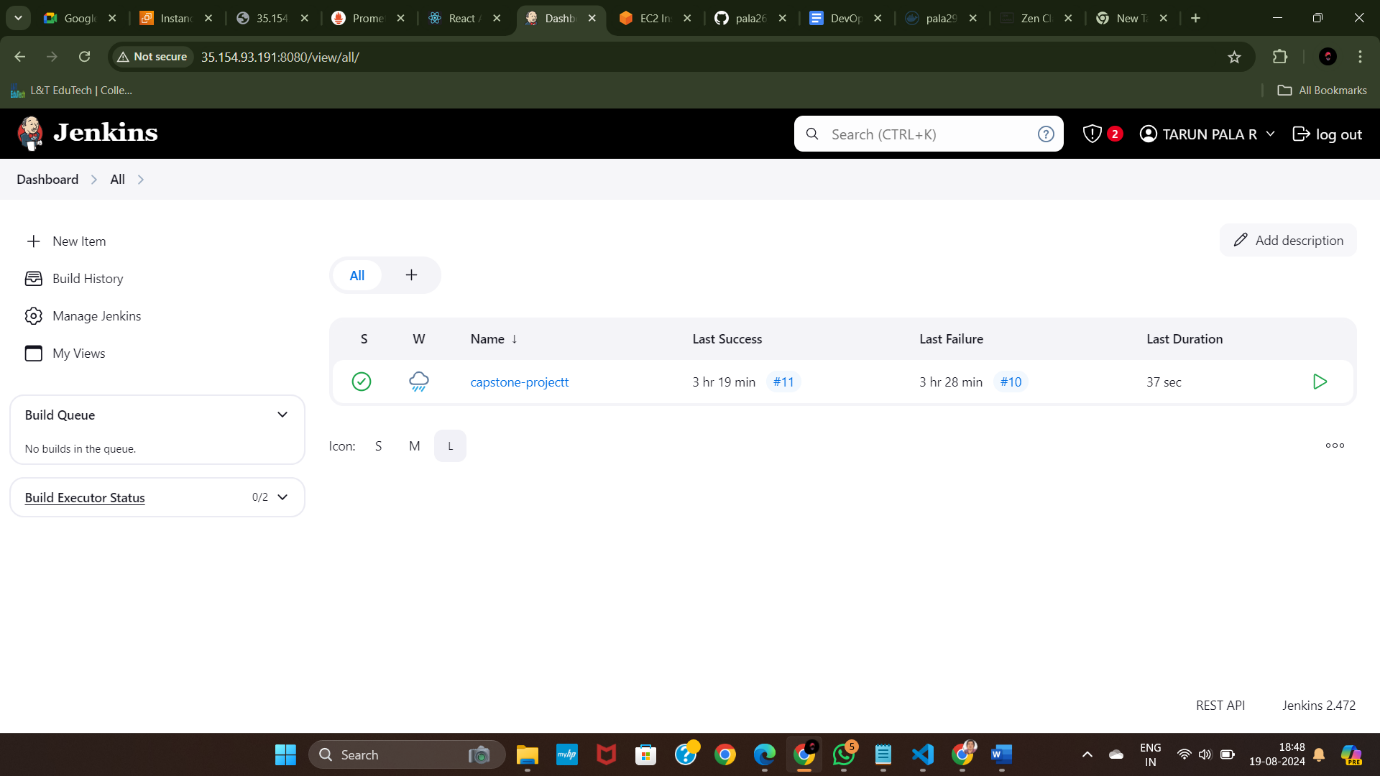
Here add your environment variables for docker i.e docker username and password

11. Next in same manage Jenkins->security->credentials->global

Add docker login credentials then only docker image can be built and pushed .

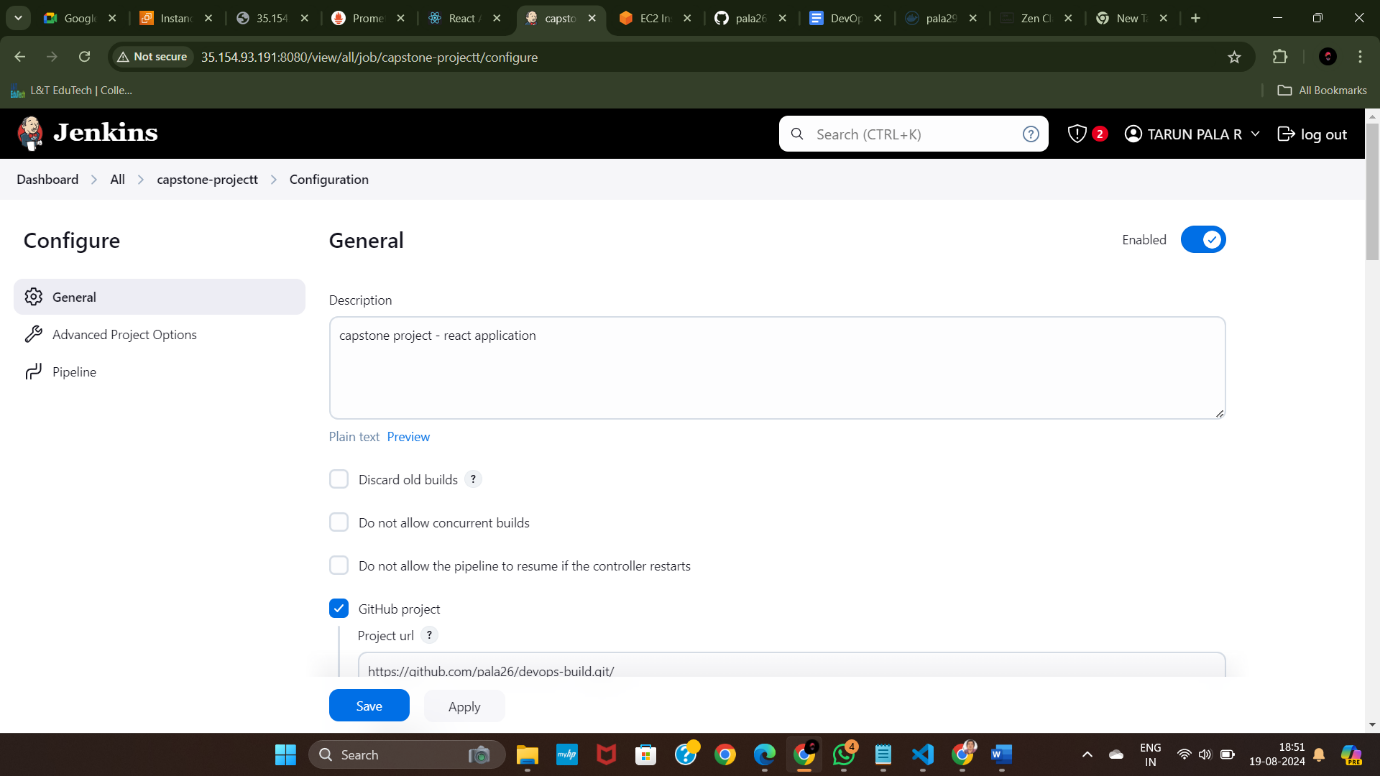
12. now create new item-> give a title for it->click pipeline

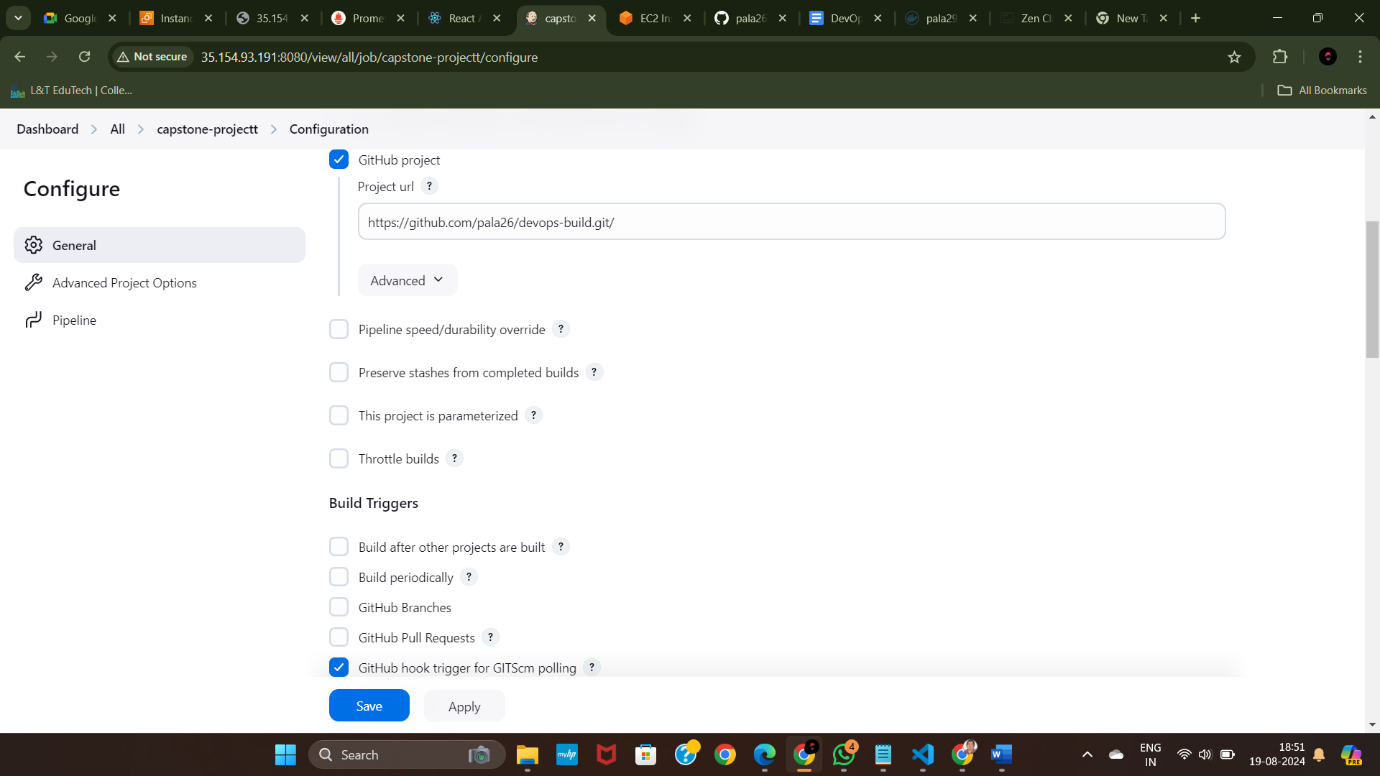
Then your item is created.

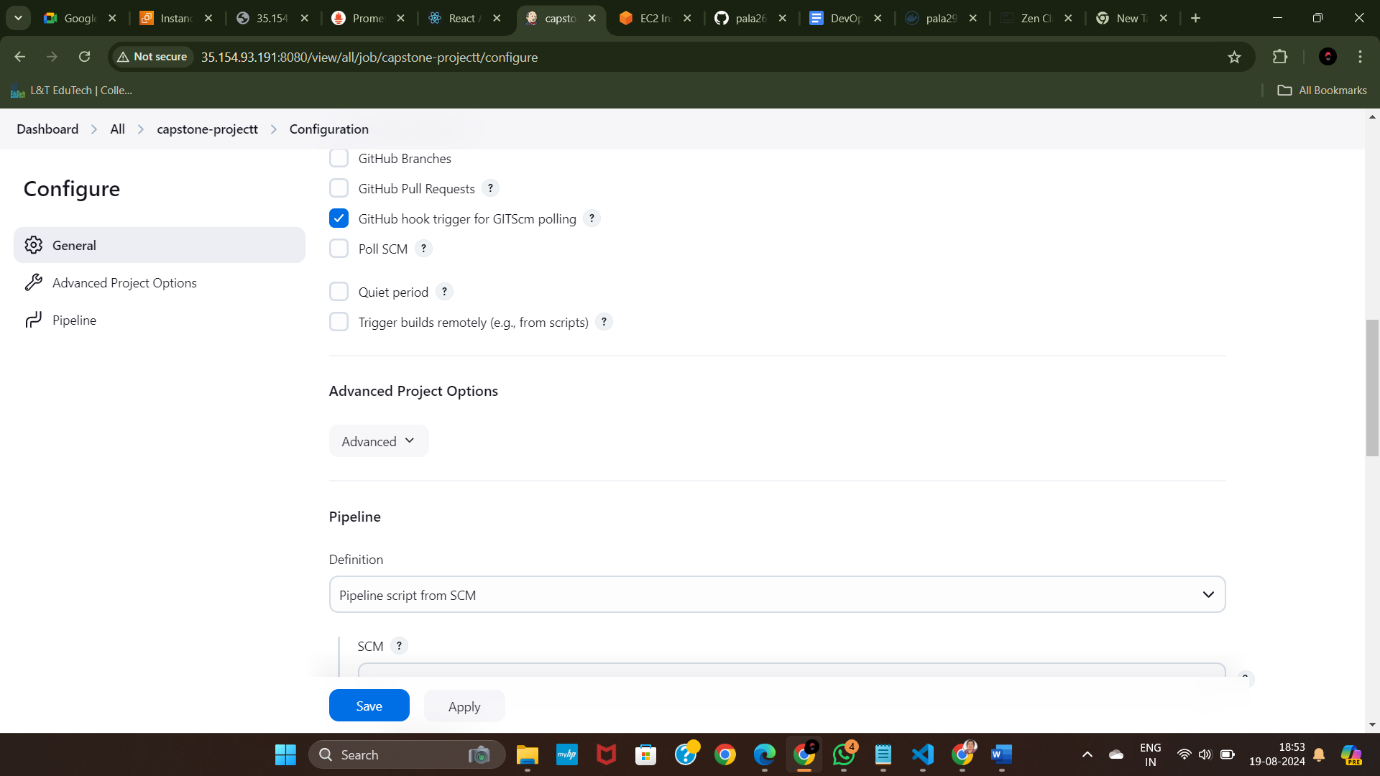


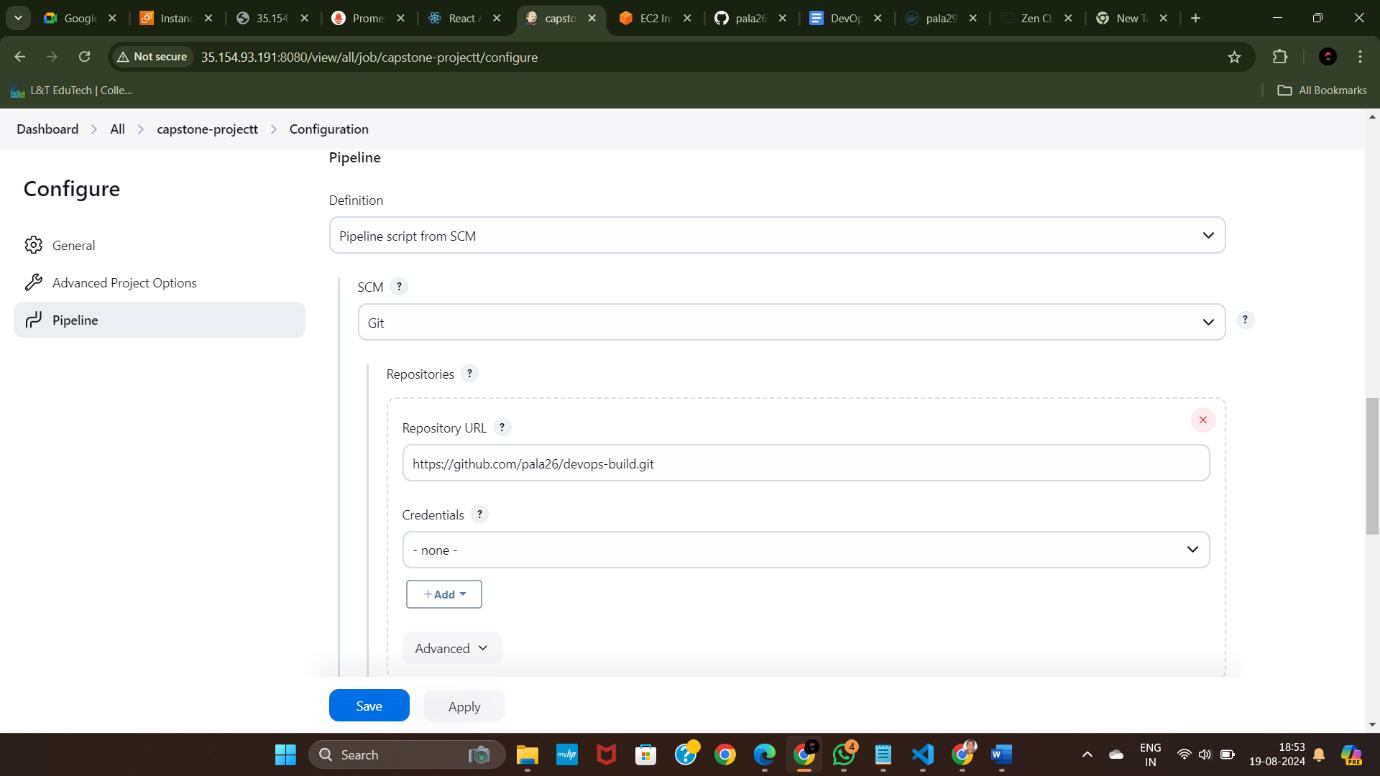
This is already build but just for reference I’ve showed you here.

13.configure it.

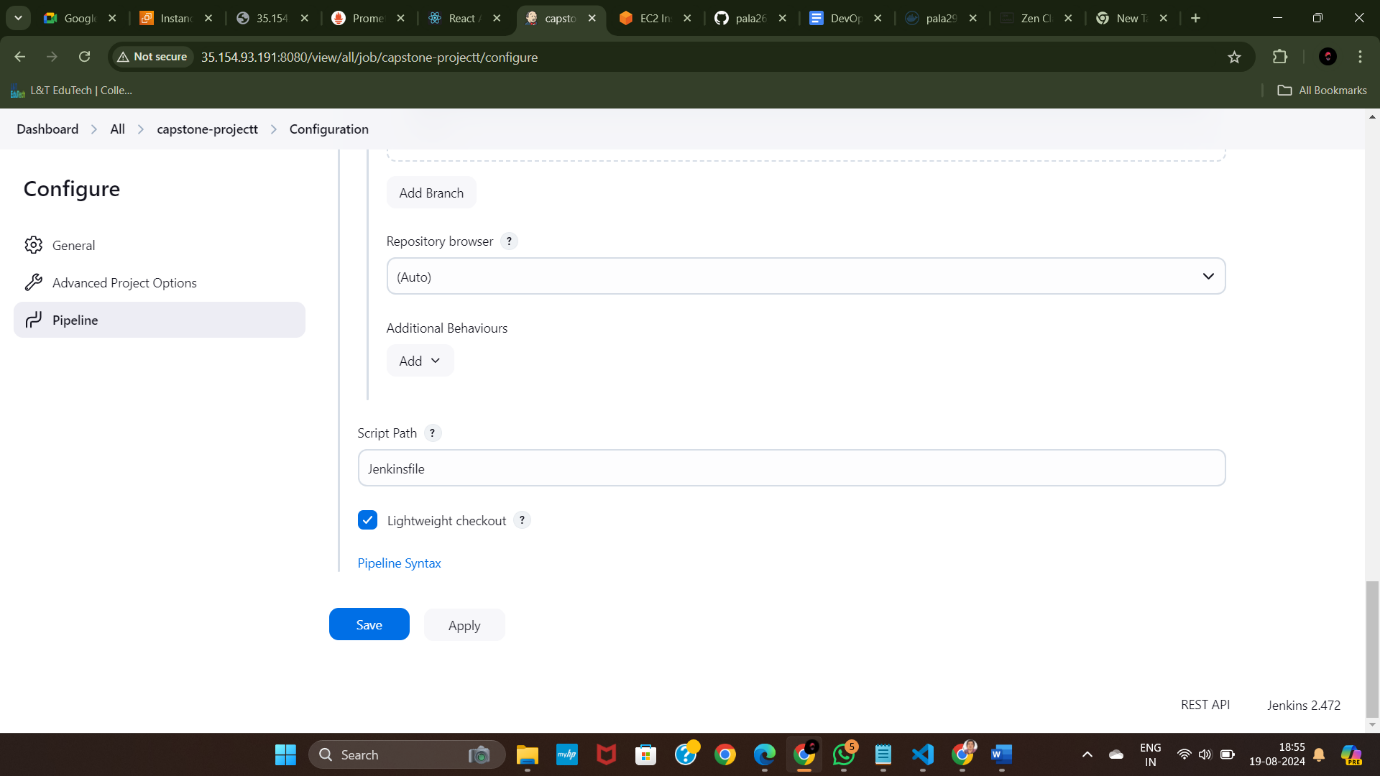






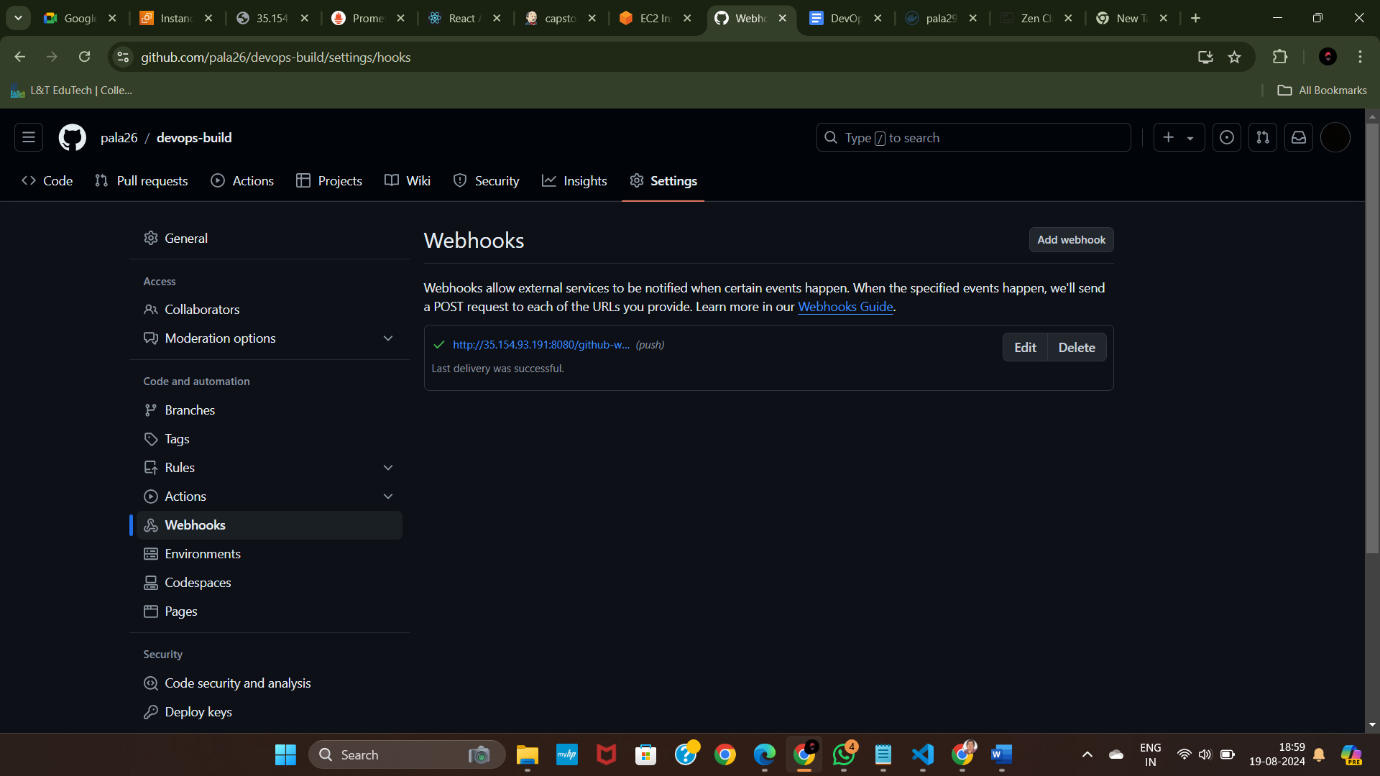






14. **Connect Jenkins to GitHub:**

* Configure GitHub webhook to trigger builds on push events.
* Go to GitHub repository settings -> Webhooks -> Add webhook -> Jenkins URL .



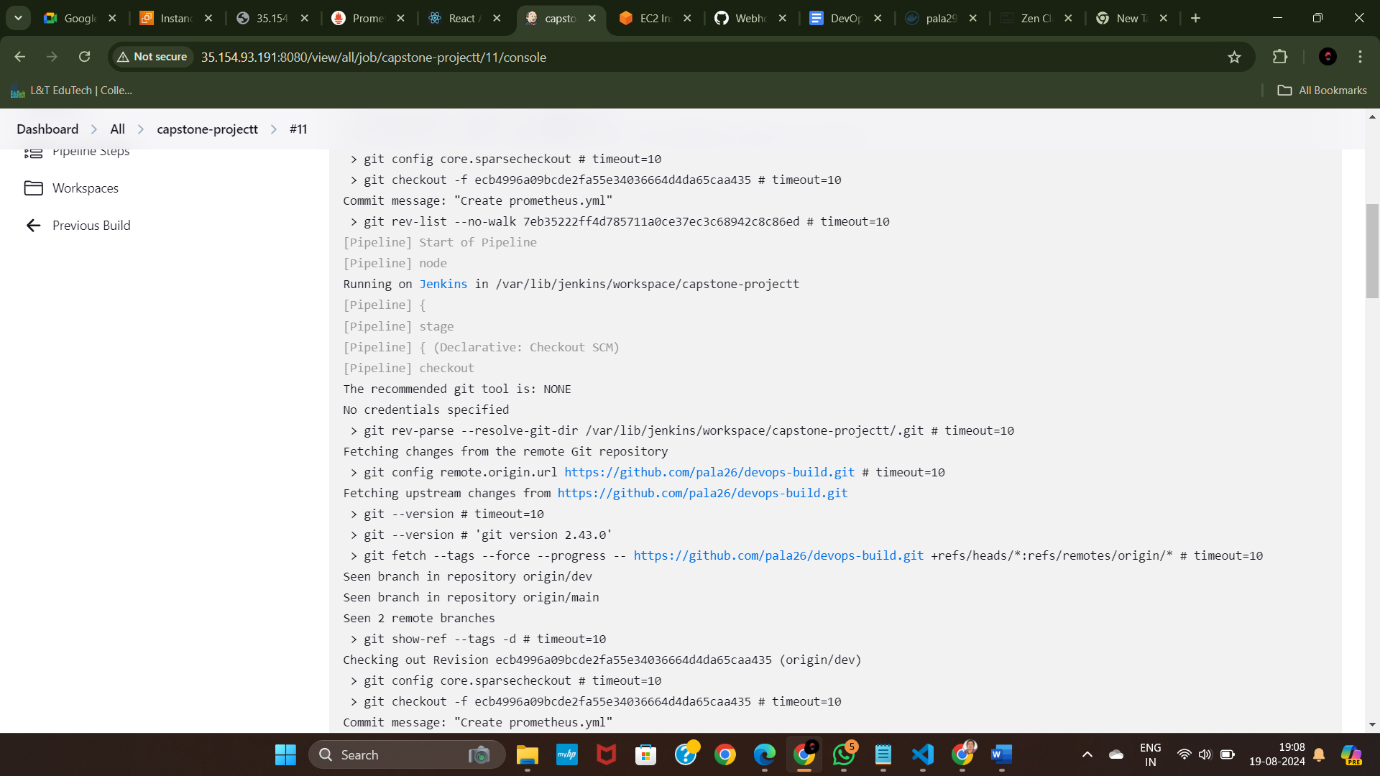
15. go to your Jenkins item [ capstone-project ] and click build now.

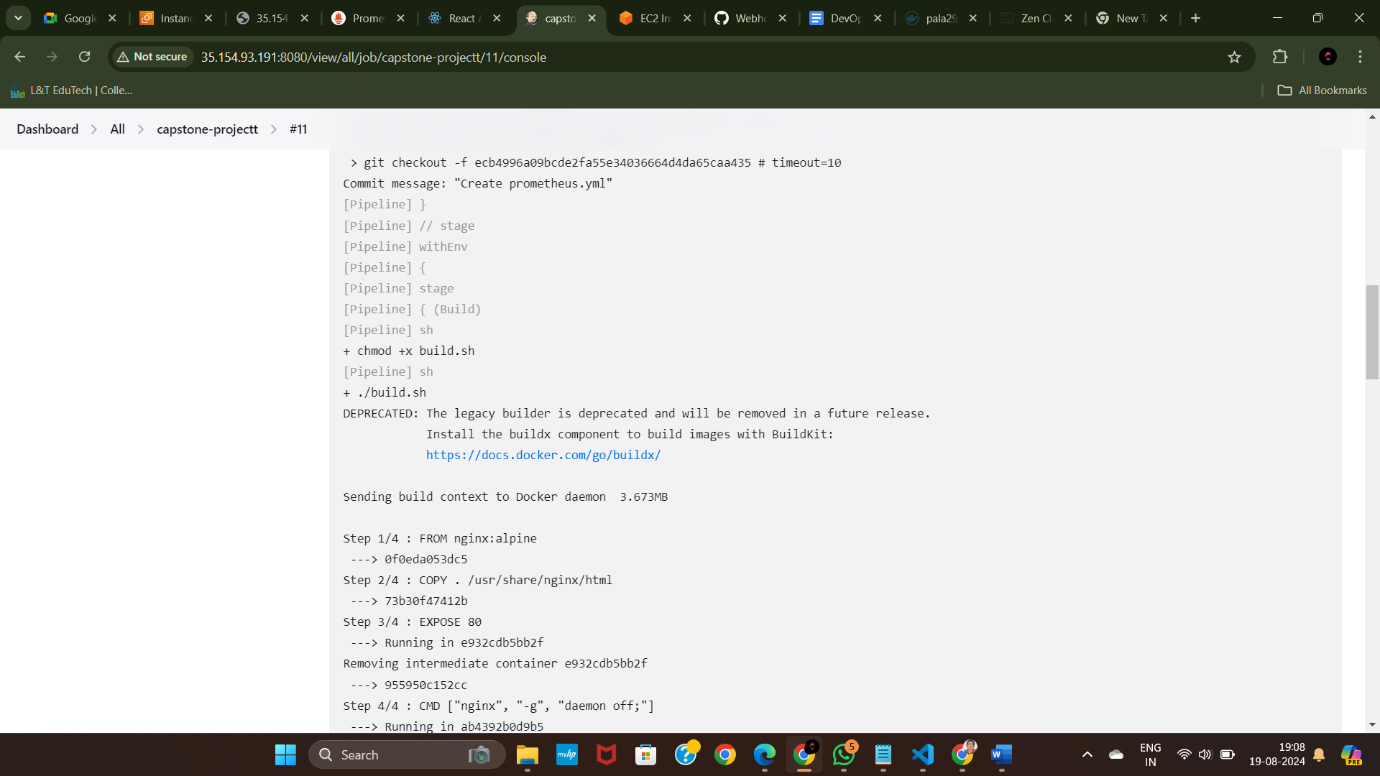
16. Check the console output whether docker image gets created and pushed

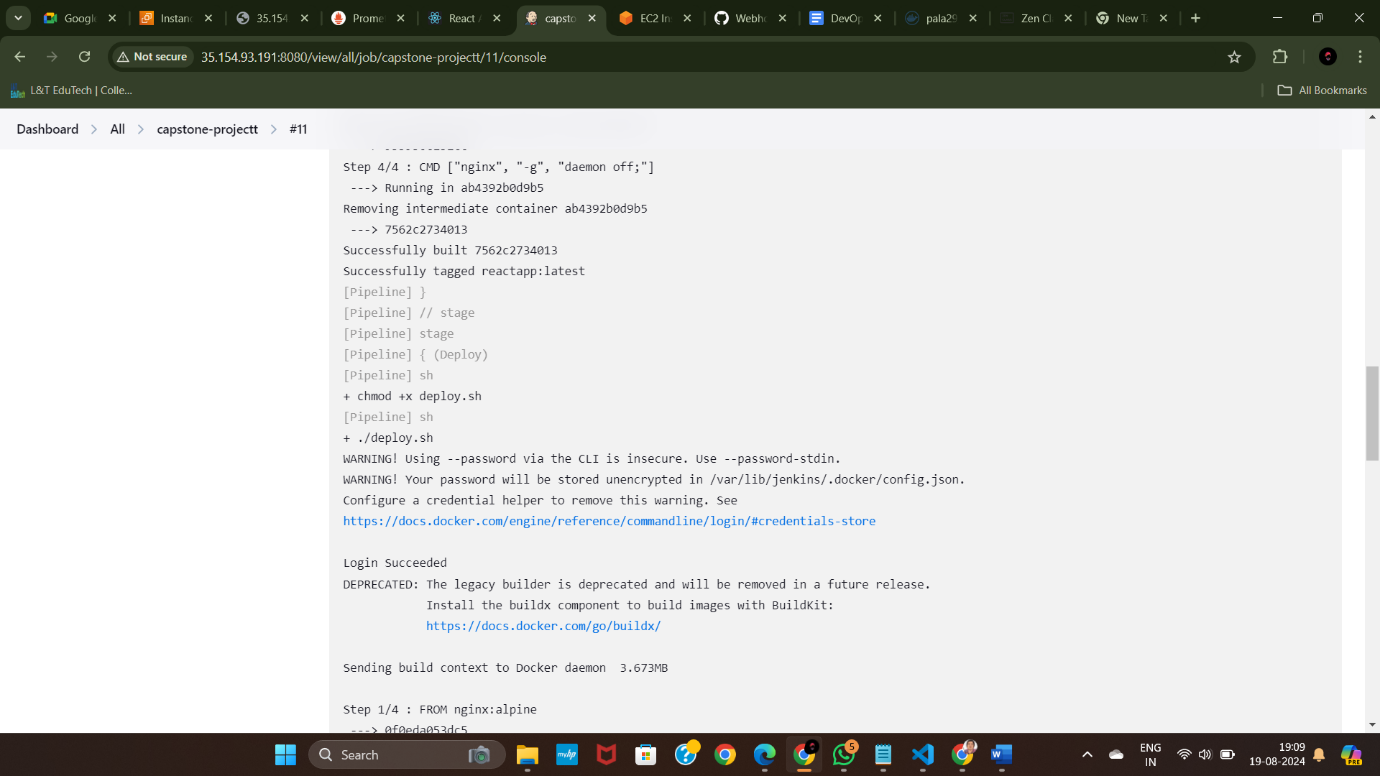
17. Also I got some error so I did update commit .

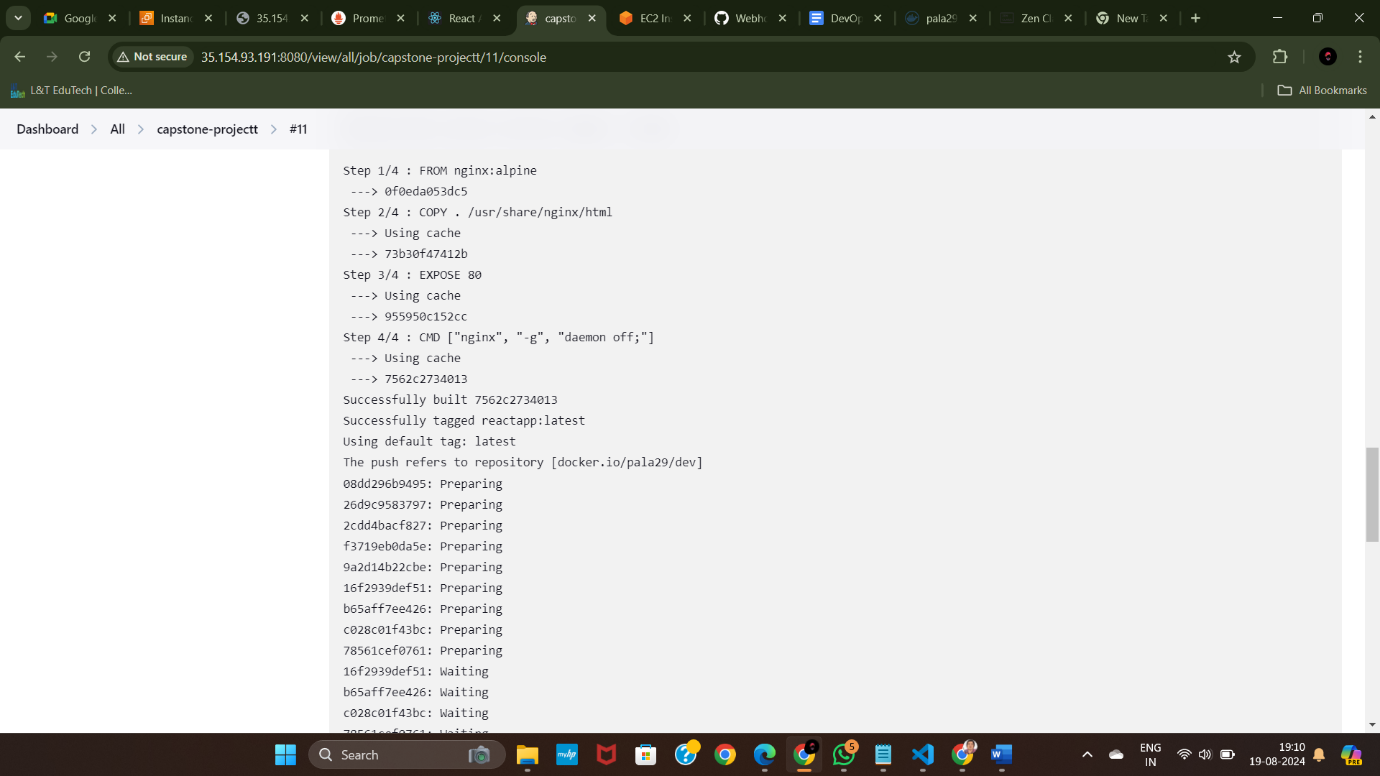
18. Then you can check in Jenkins build gets triggered automatically .

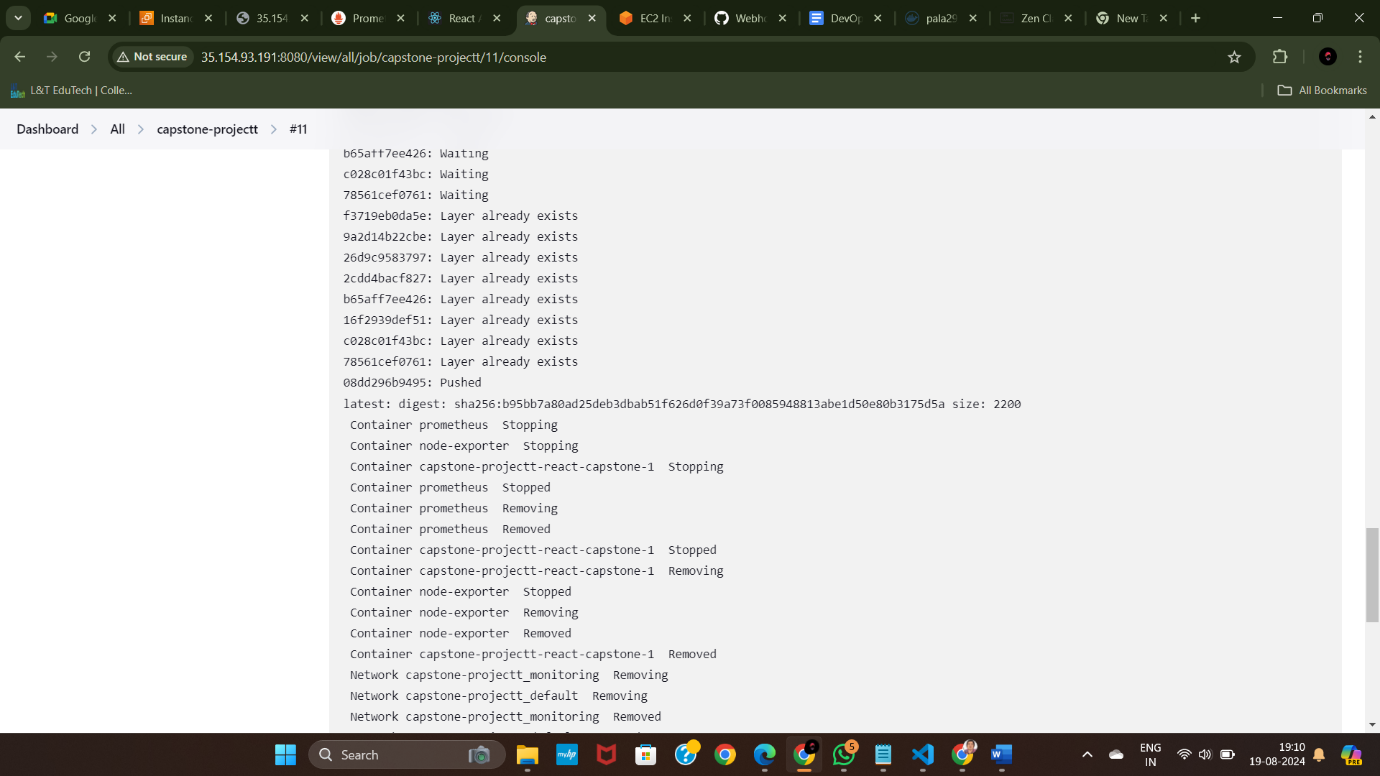






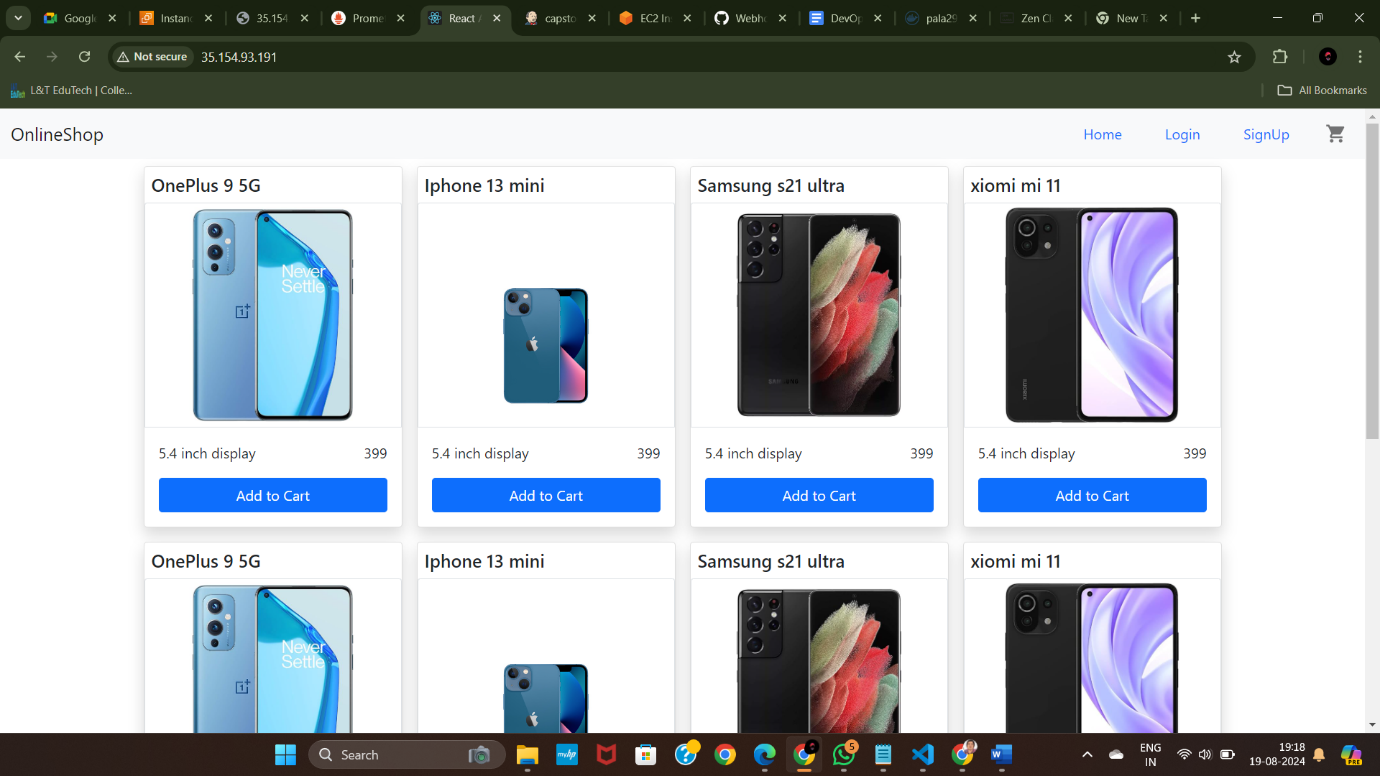


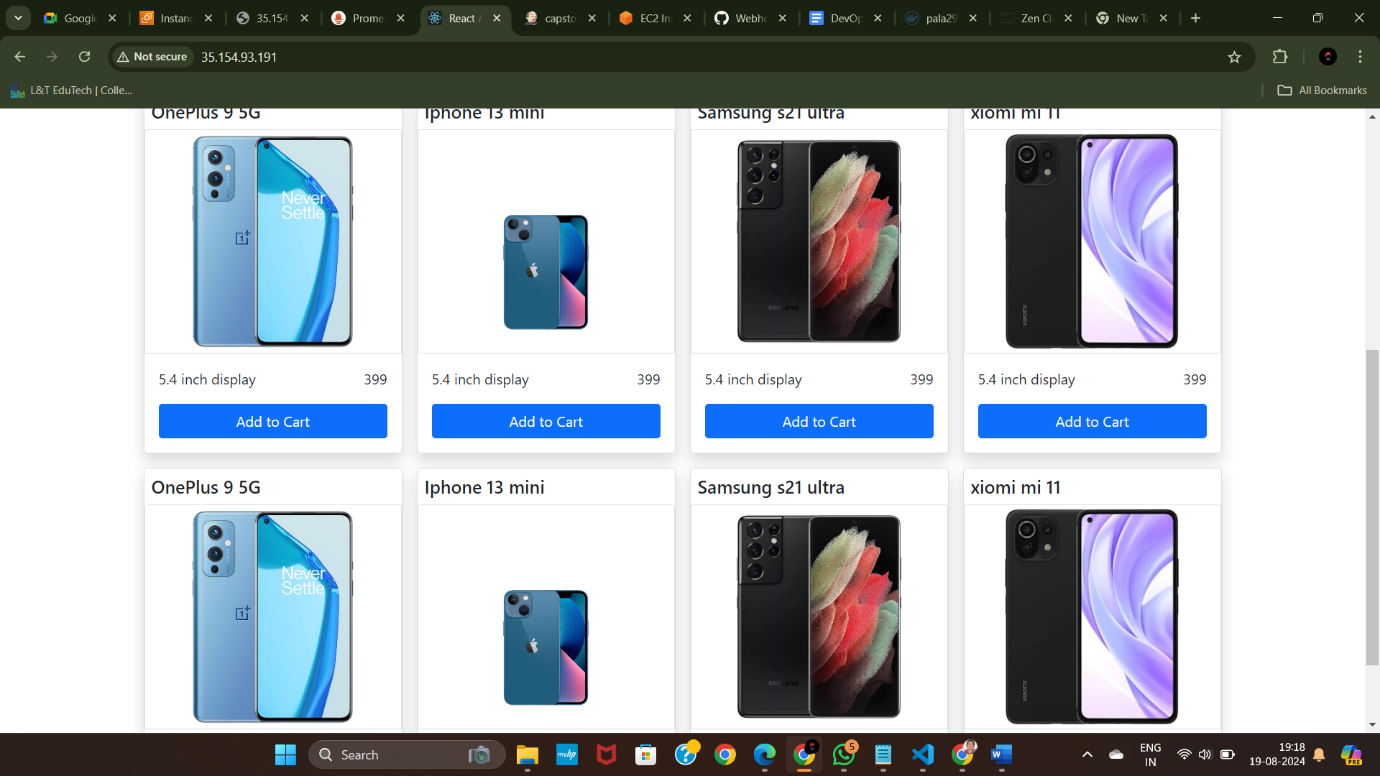




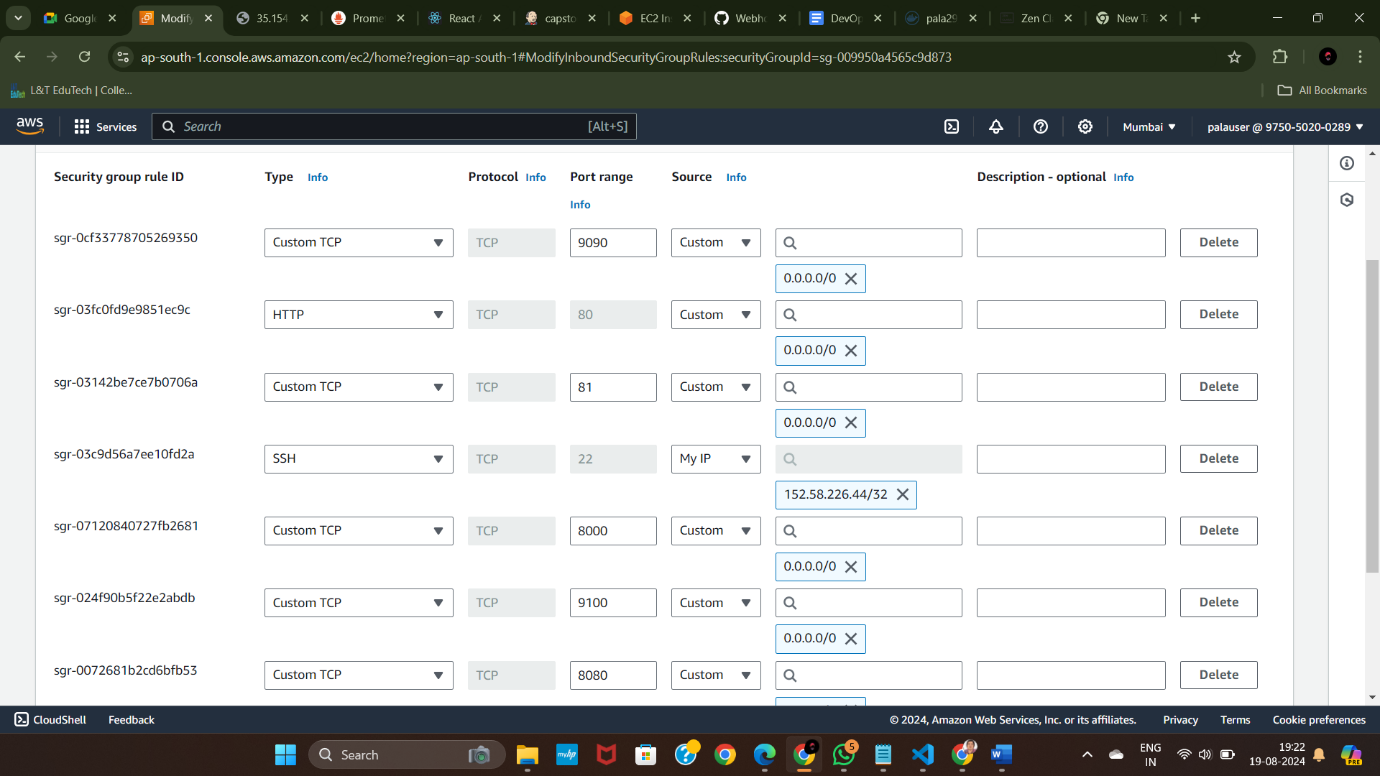


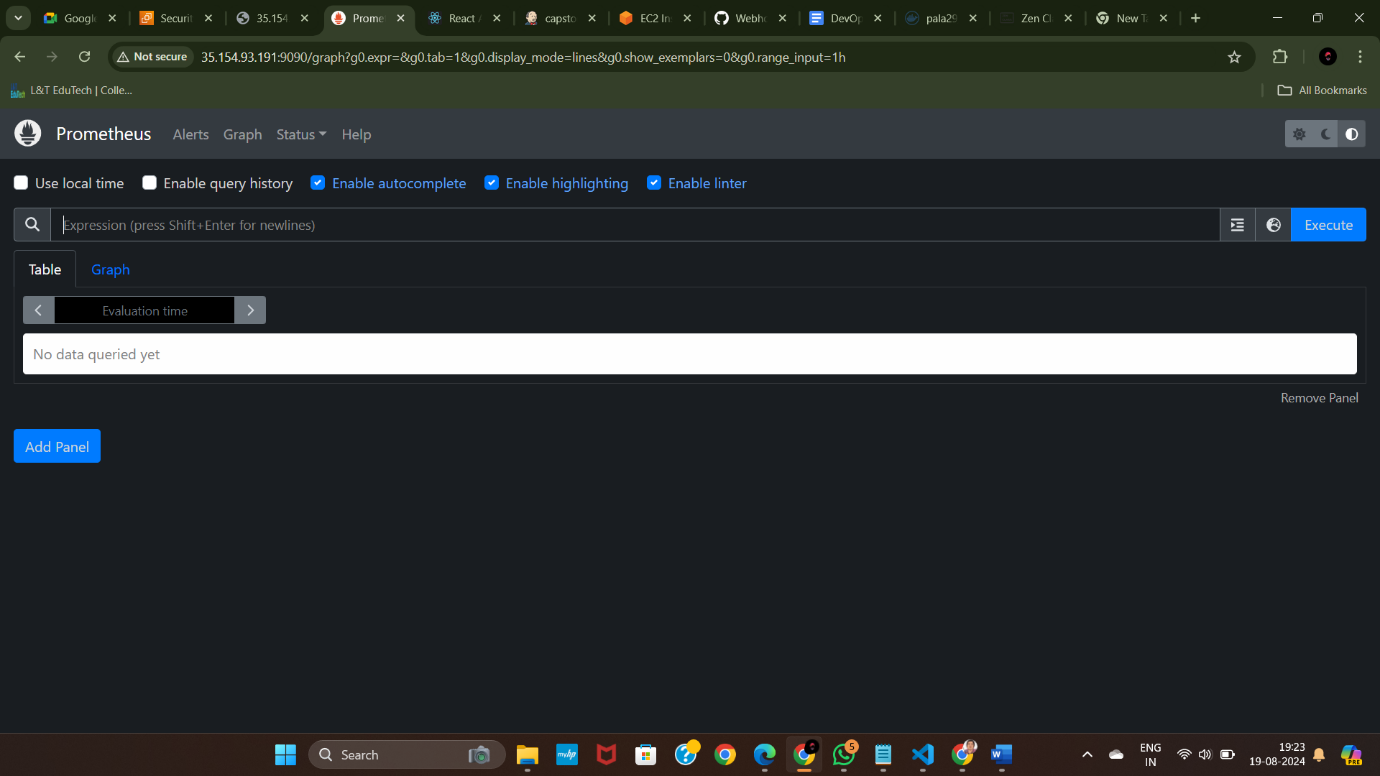
19.

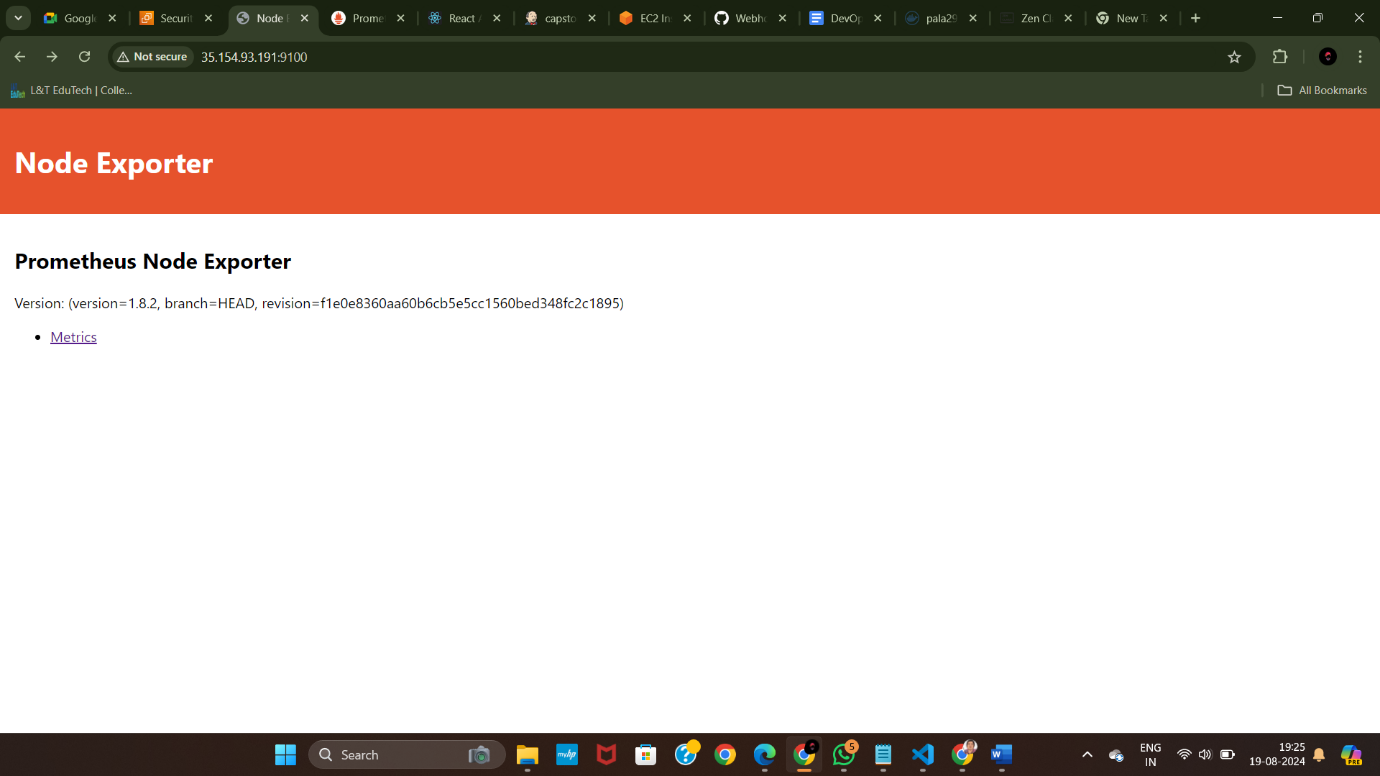


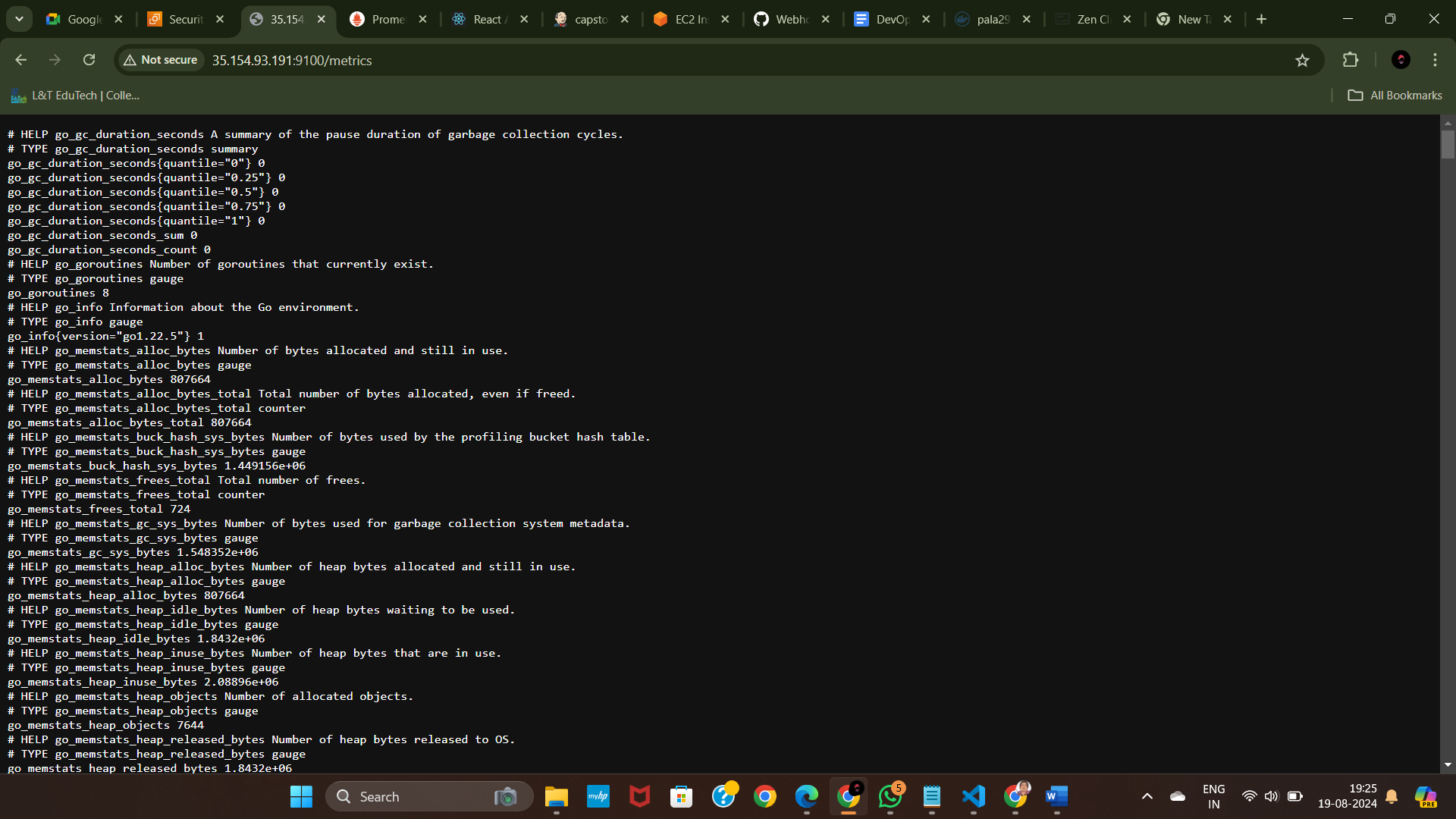


This is the application we were given to deploy. This is actually a react based application.

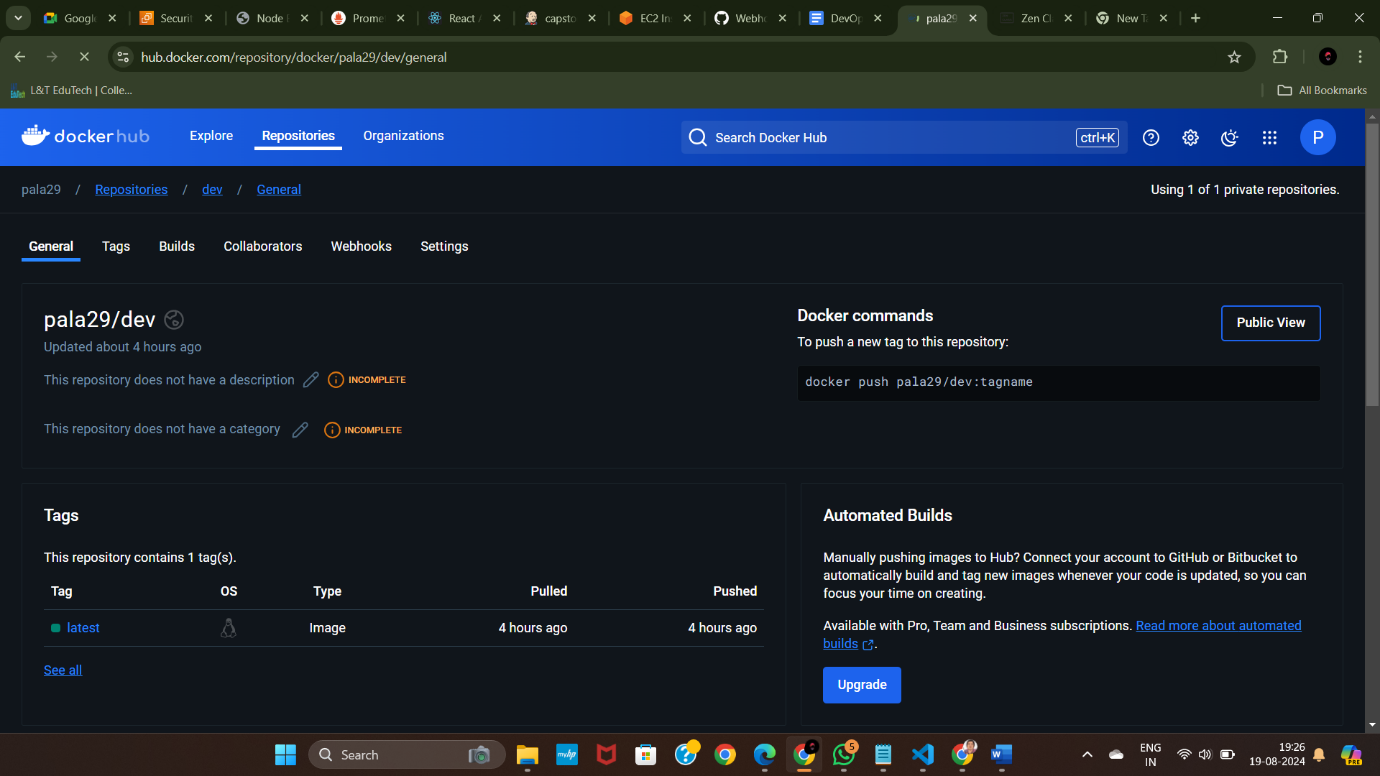
20. these are the ports we have enabled on security group.

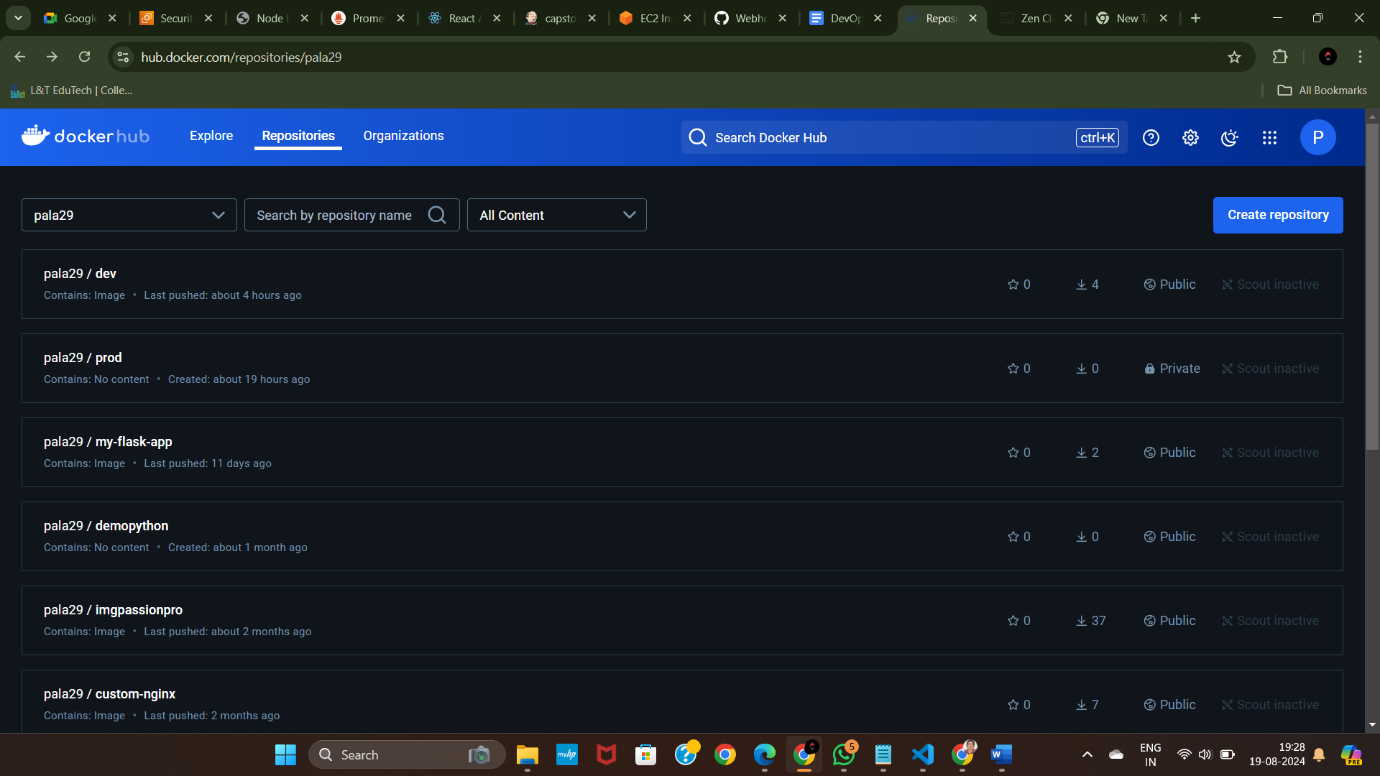
21.prometheus after installed it

22. node exporter



23. docker hub





[ for code I’ve given my github repository in the beginning itself you can check that ]

----------X----------