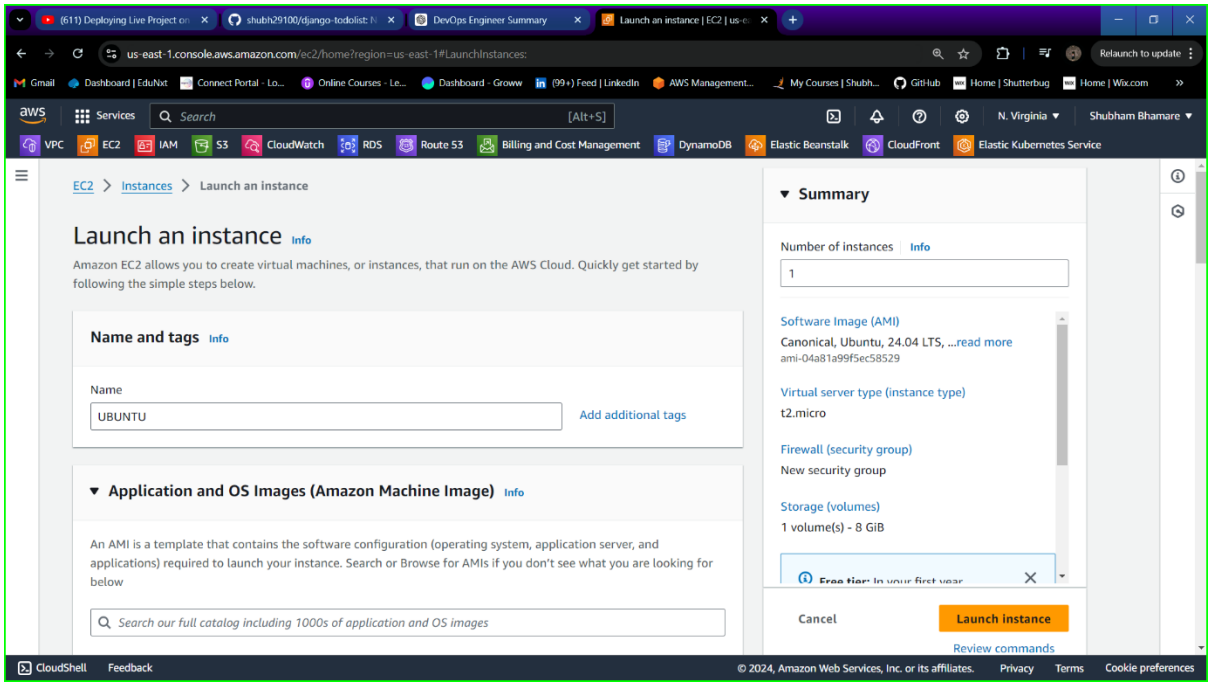
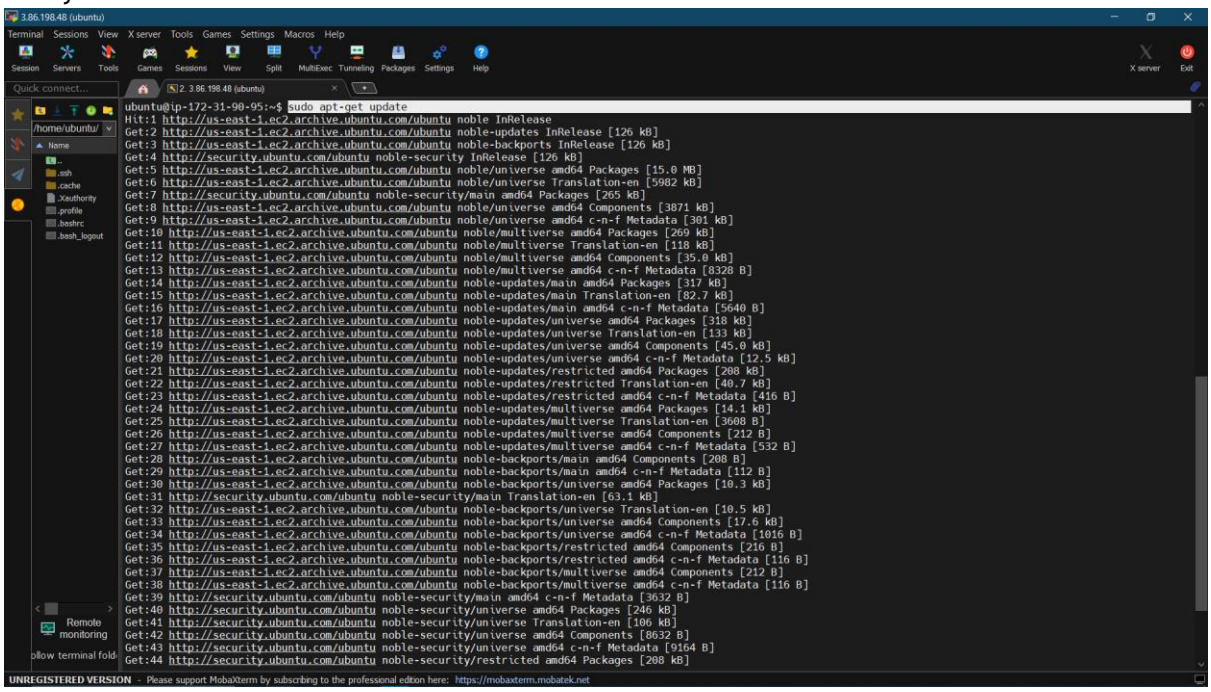
Create a CI/CD pipeline for web application using docker, Jenkins and AWS.

Step 1: Login into your AWS account. Go to EC2 dashboard and launch an instance (Allow SSH rule).

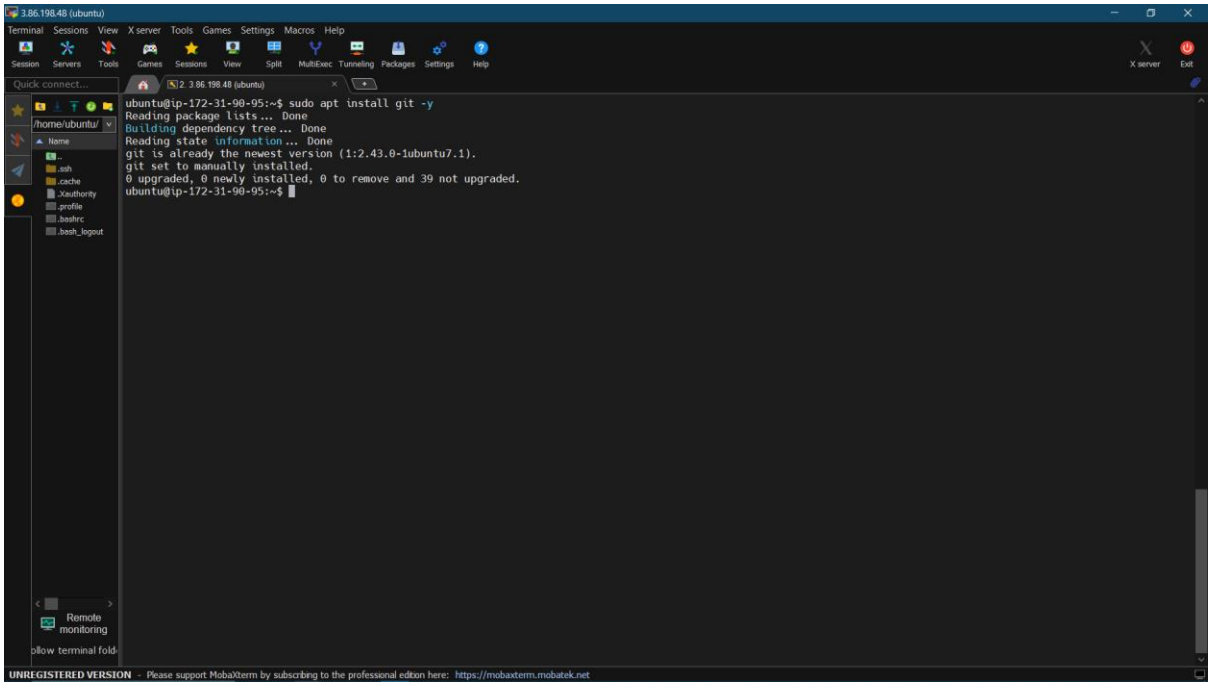


Step 2: Take remote access of your instance and run command “sudo apt-get update” to update the system.



Shubham Bhamare

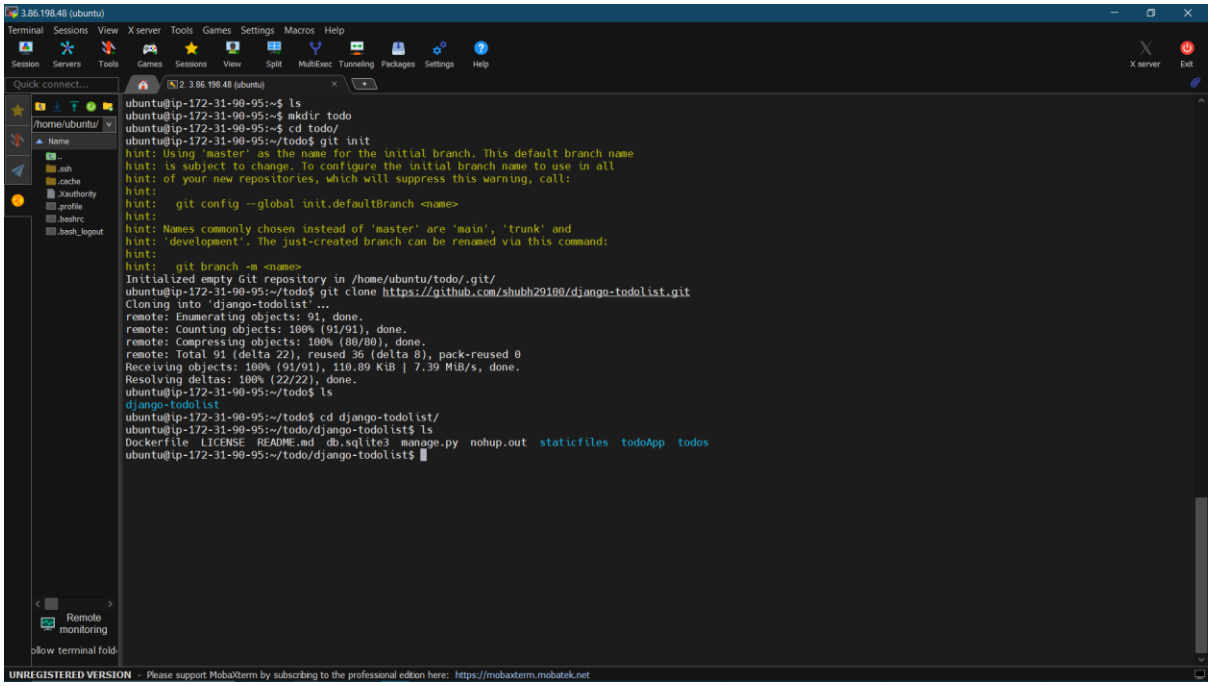
Step 3: Run “sudo apt install git -y” to install version control system (Github).



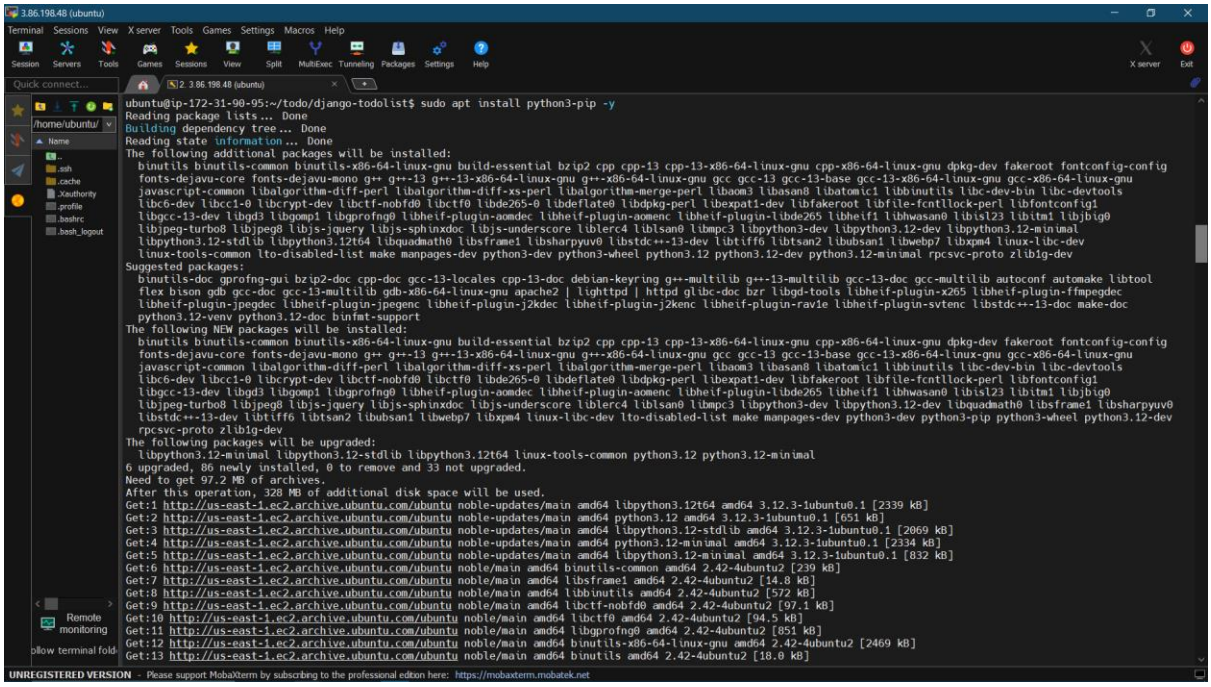
Step 4: Create a new directory using command “mkdir todo” , then go into that directory using command “cd todo” and then initialize git into that directory using command “git init”.

After initializing git clone the github repository in which the project is stored

“git clone https://github.com/shubh29100/todolist.git “. After this the project is stored in your local repository.

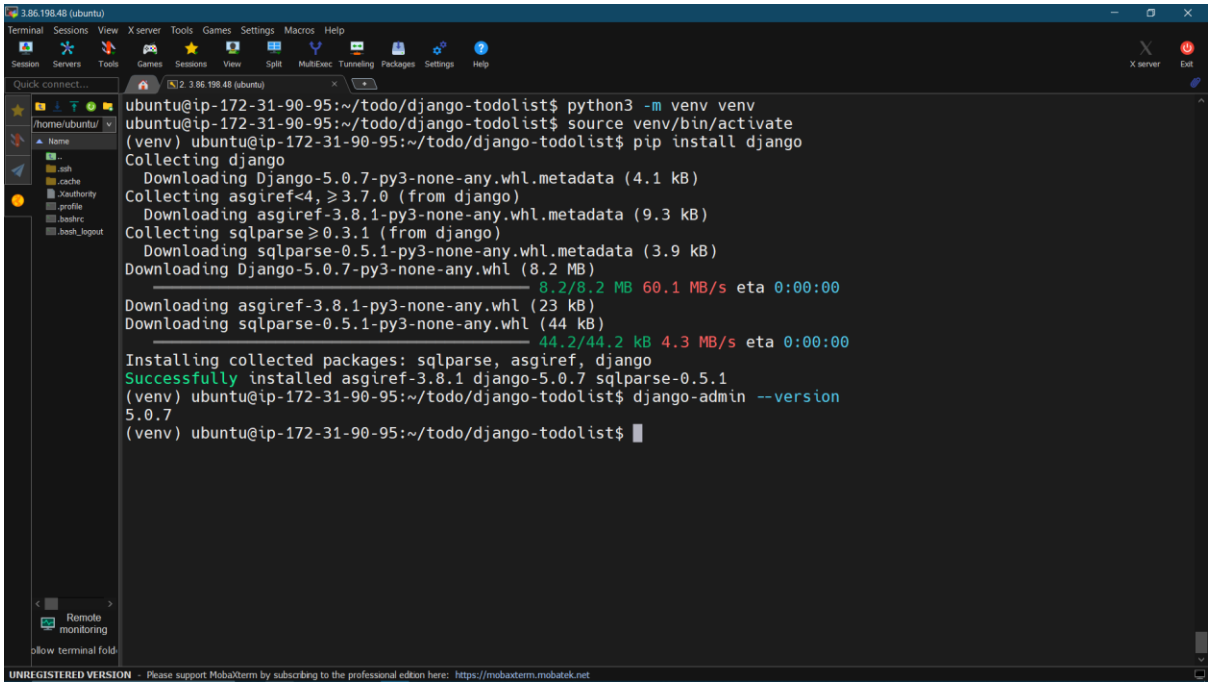


Step 5: Run command “ sudo apt install python3-pip -y” to install python 3 in your EC2 instance.

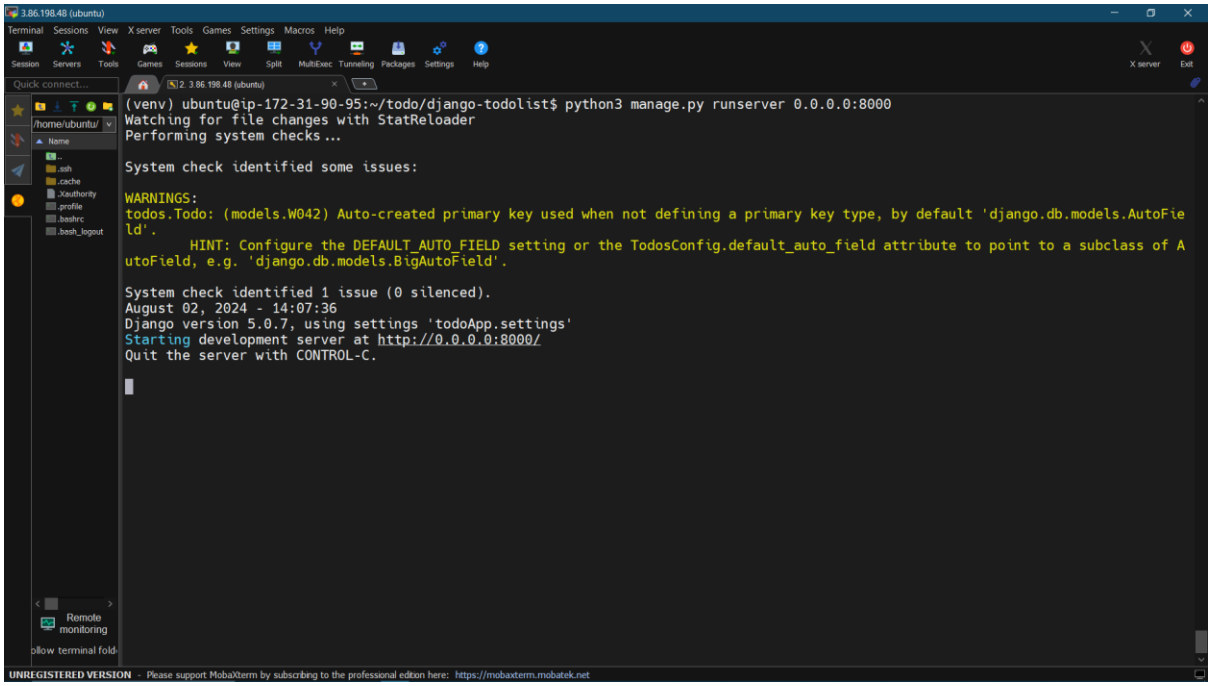


Step 6: now we have create virtual environment to run project “python3 -m venv venv”. After this run command “source venv/bin/activate” to activate the virtual environment.

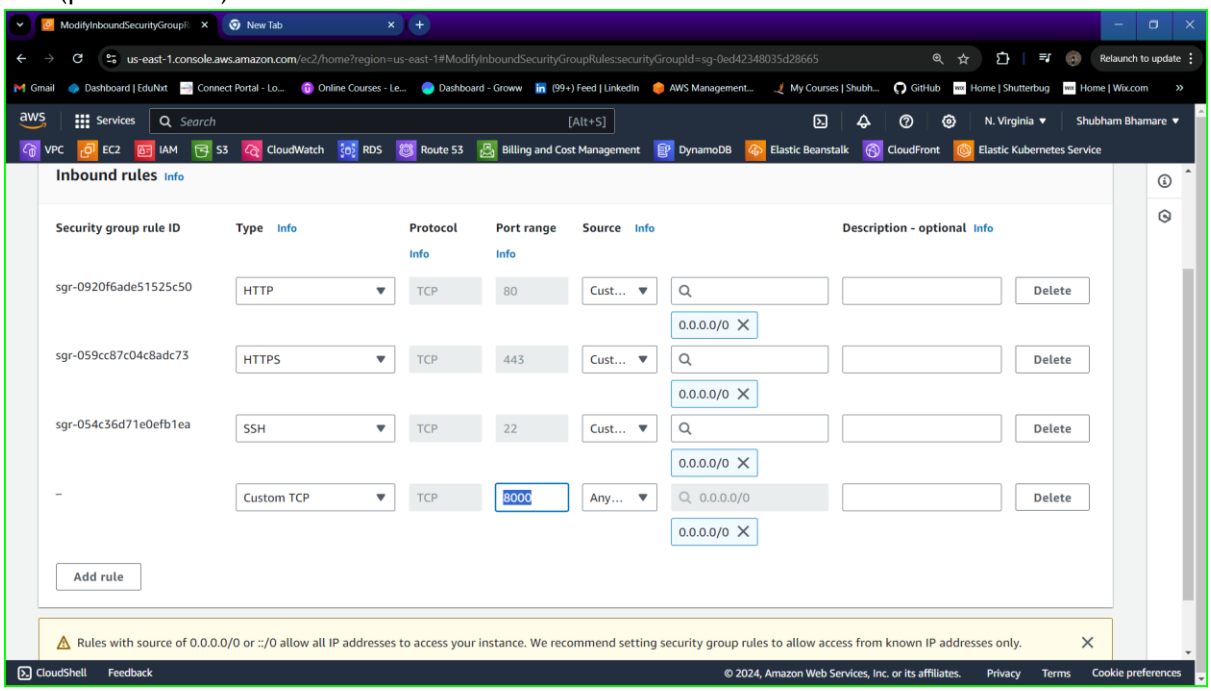
Now run command “pip install Django” to install Django into your virtual environment.



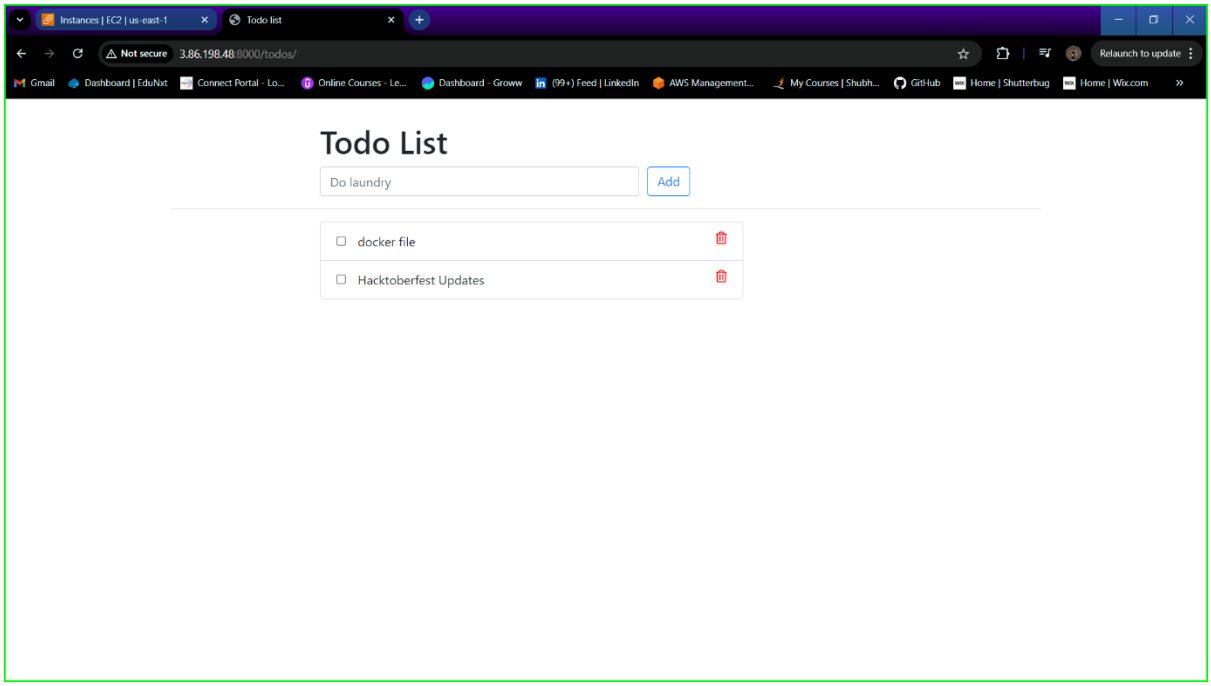
Step 7: Run command “python3 manage.py runserver 0.0.0.0:8000” to start your project on port no 8000 .



Step 8: Go to your security group which is attached to your EC2 Instance and add a rule for custom TCP (port no 8000) and save it.

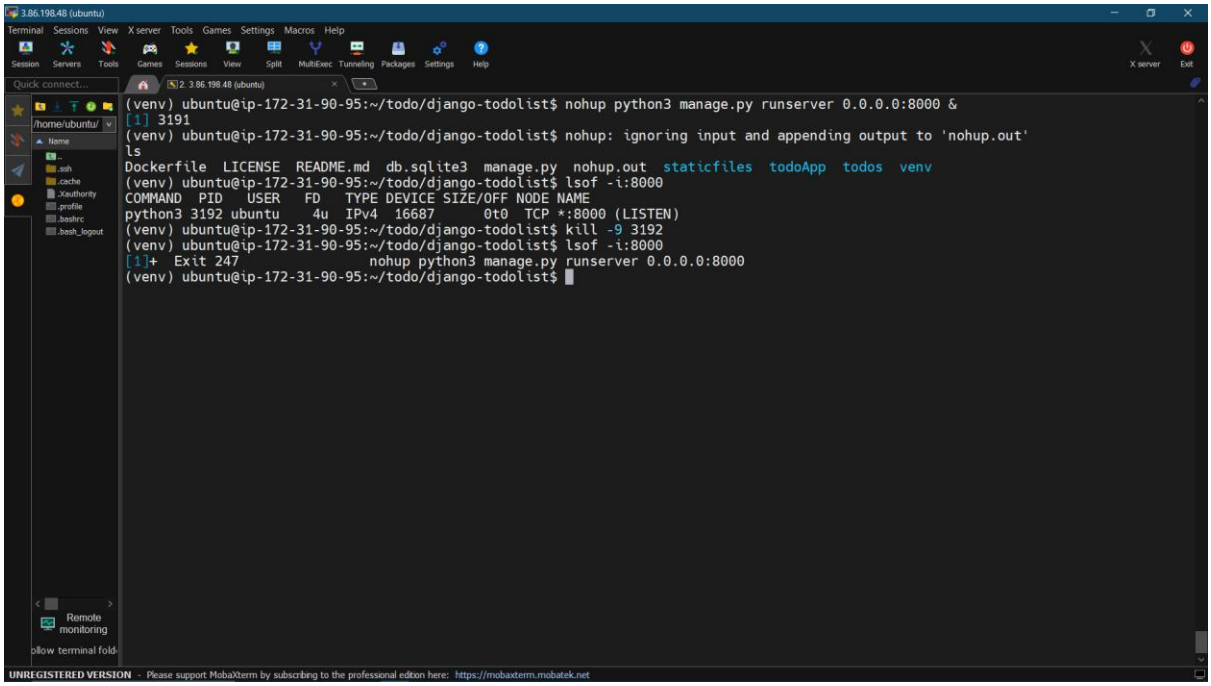


Step 9: Now run “public Ip:8080” and you can that the project is running successfully.



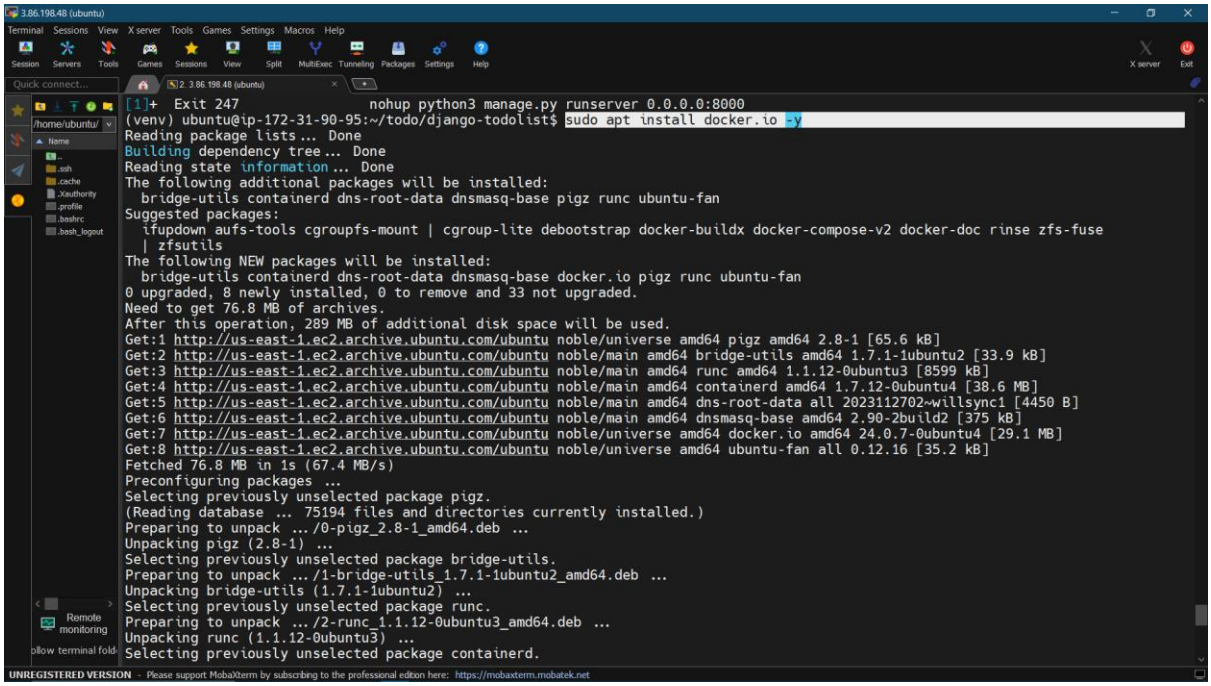
Step 10: Run command “nohup python3 manage.py runserver 0.0.0.0:8000 &” so that the project will run in background and you can use the terminal.

Run command “lsof -i:8000” to see which service is running on you 8000 port. Then run command “kill -9 PID” to kill that service.



Step 11: Now you have to install docker into your server to create a docker image which we later require to automate the process.

Run command “ sudo apt install docker.io -y” to install docker.



Step 12: Create a new Docker file ,Run command “vim Dockerfile” .

Press “I” to get into insert mode and then write

“ FROM python:3

RUN pip install django==3.2

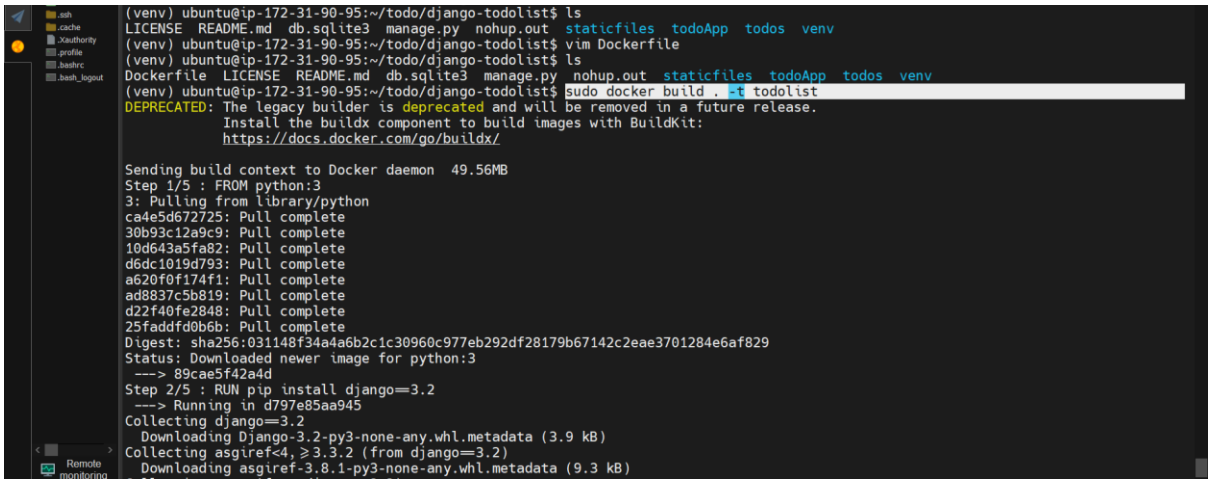
COPY . .

RUN python manage.py migrate

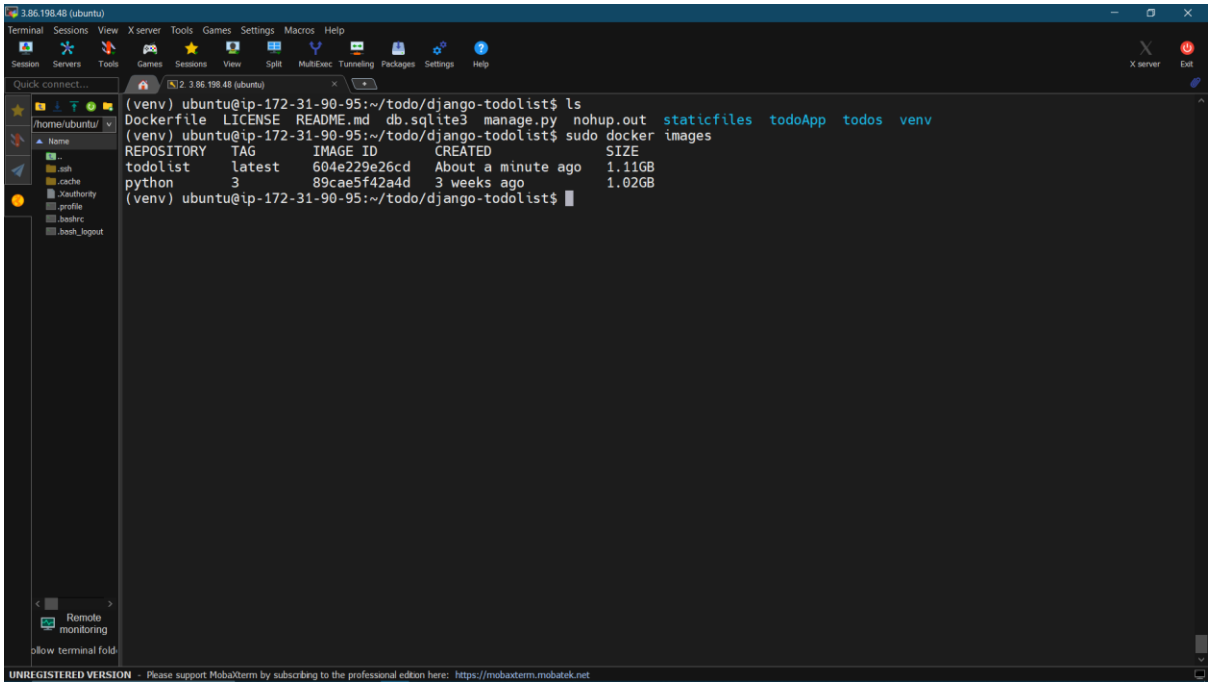
CMD [“python”,”manage.py”,runserver”,”0.0.0.0:8000”] “

Then press “esc” to come out of insert mode and the press “:wq!” to save the file.

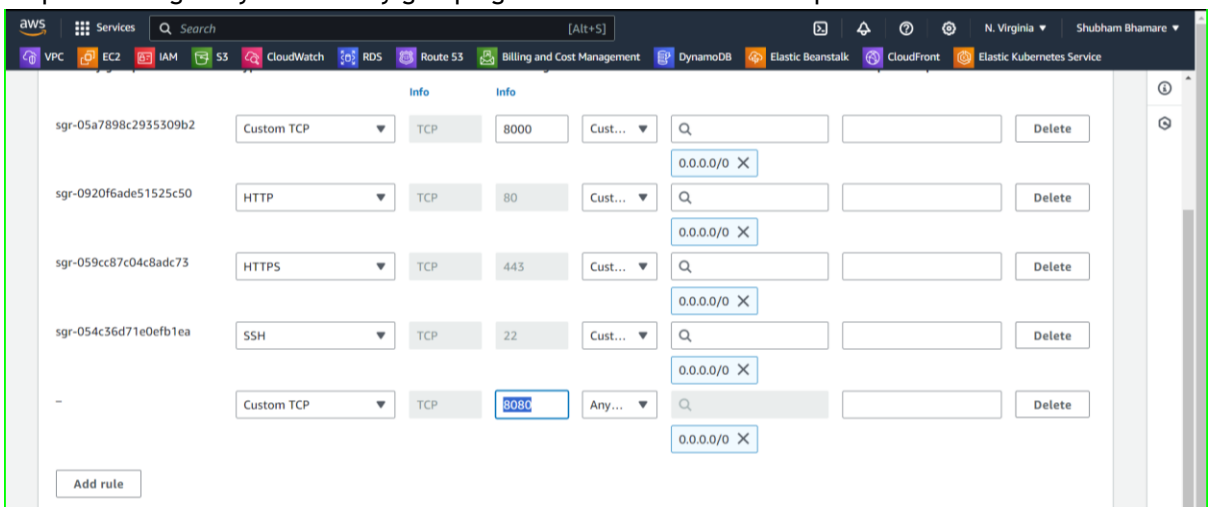
Afte saving Docker file run command “sudo docker build . -t todolist” to build a image from dockerfile.



Step 13: The image is built successfully. Run command “sudo docker images” to check if the image is present in your server.



Step 14: Now go to your security group again and add rule for 8080 port and save it.



Step 15: Now you have install java and Jenkins on your server.

Run command “sudo apt install openjdk-11-jdk -y” to install java.

“java -version” to check java version.

Now run “curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null” to import Jenkins GPG key.

Run “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” – adds Jenkins repository to your system.

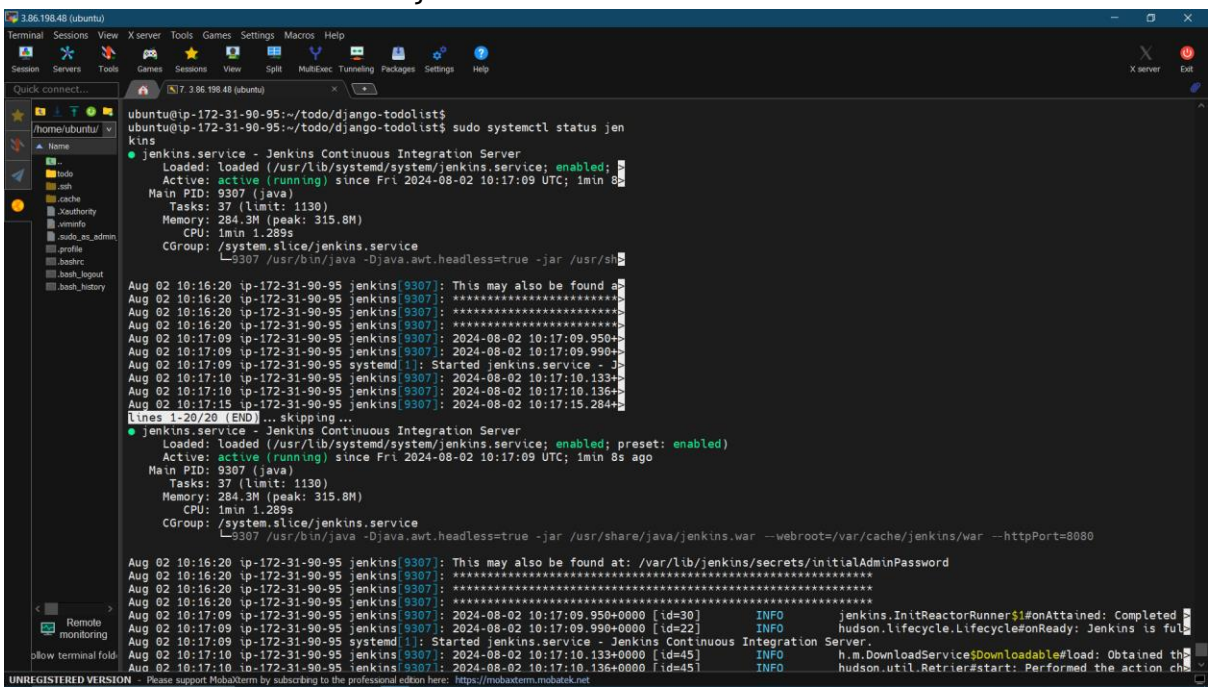
Run “ sudo apt update” - Update your package list again to include the new Jenkins repository.

Run “sudo apt install jenkins -y” to install Jenkins.

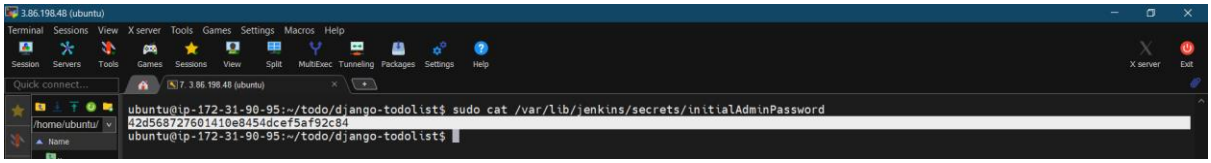
Run “sudo systemctl start Jenkins” to start Jenkins service.

Run “sudo systemctl enable Jenkins” – enable Jenkins to start on boot.

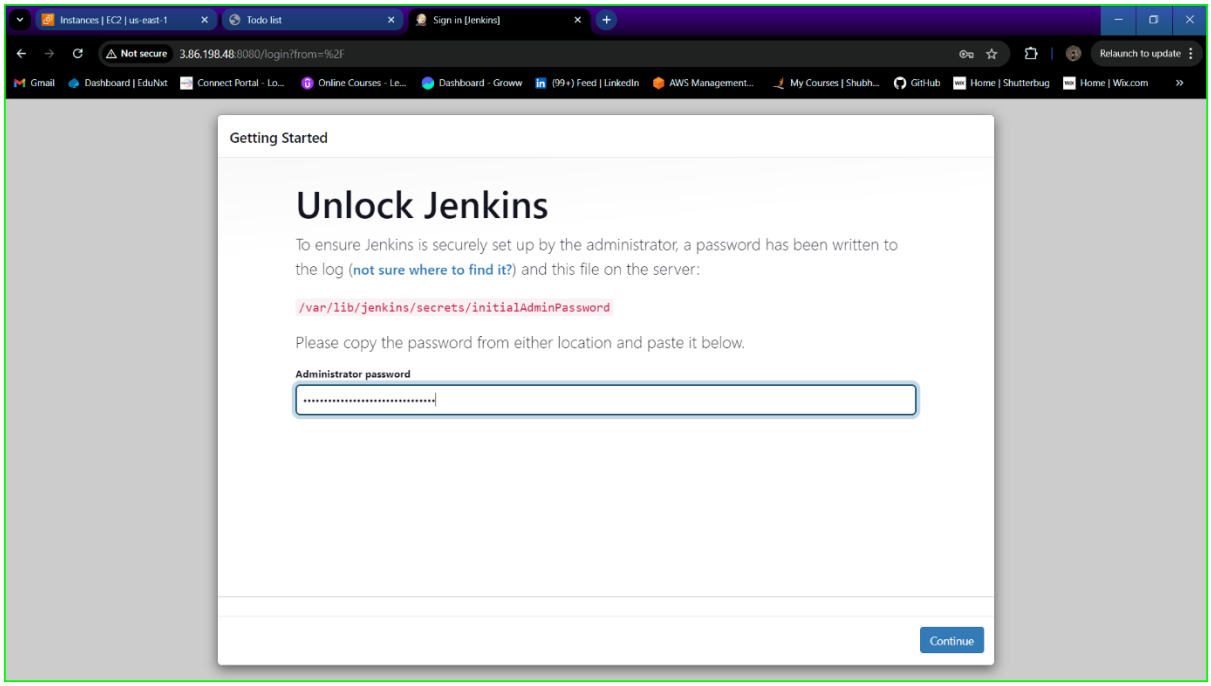
Run “sudo ufw allow 8080” – to adjust firewall.



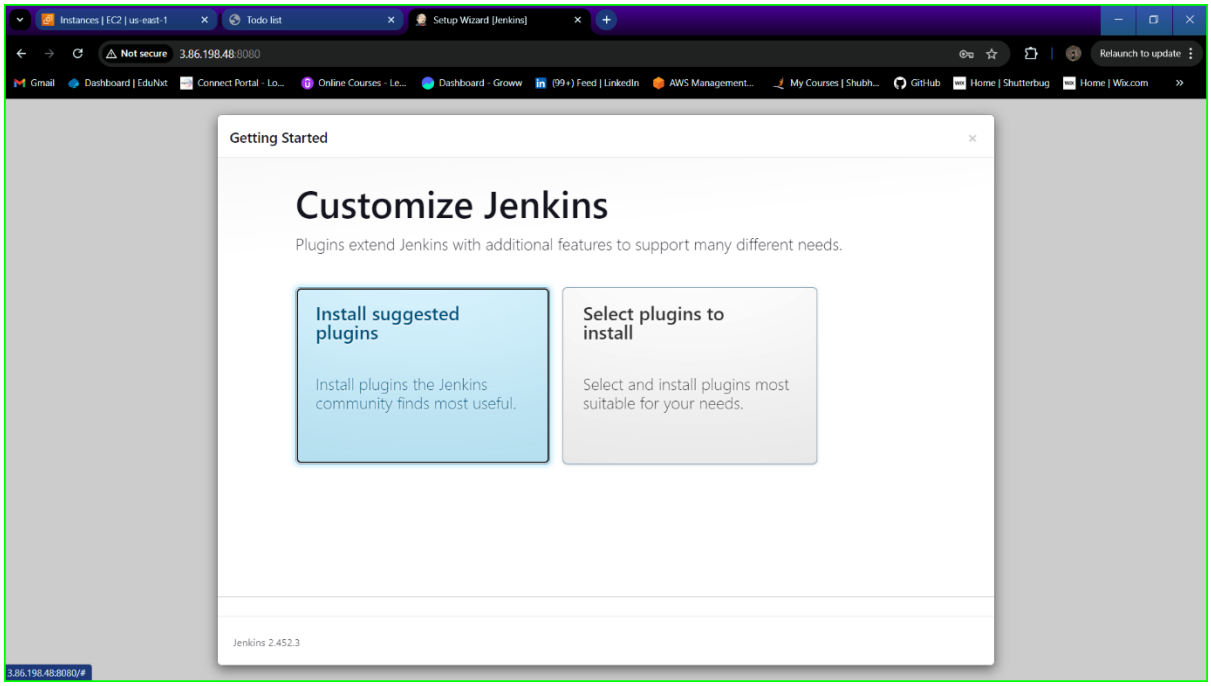
Step 16: Run command “sudo cat /var/lib/jenkins/secrets/initialAdminPassword” to view Jenkins password.



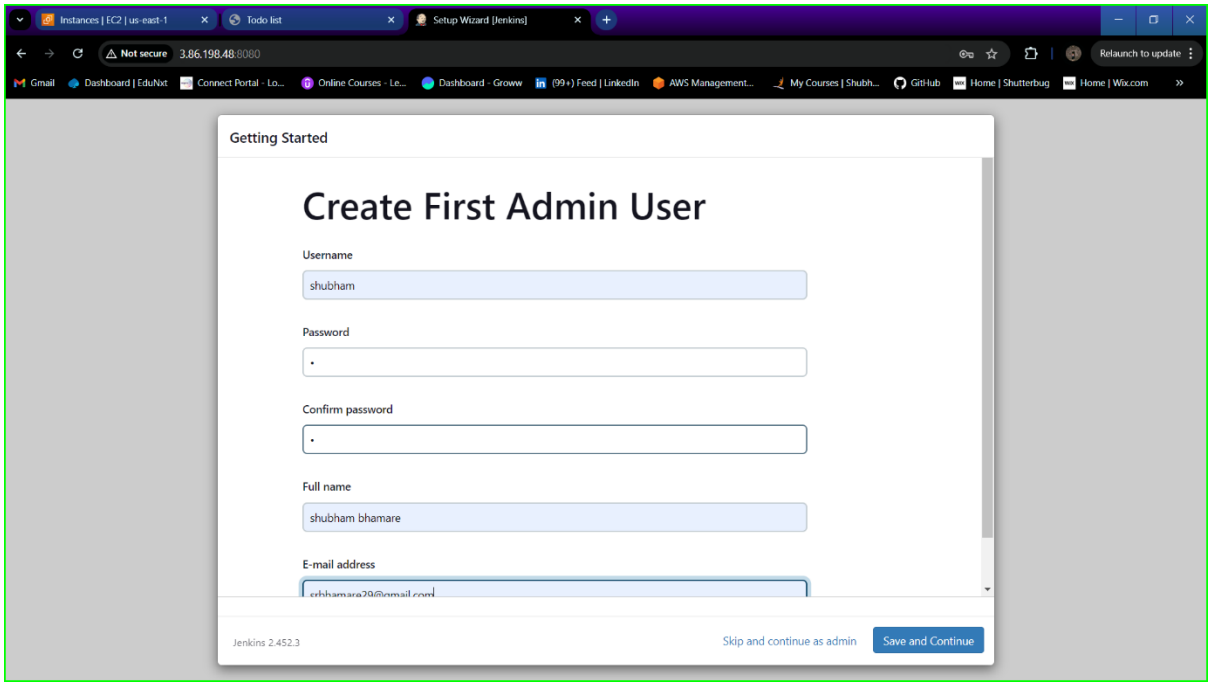
Step 17: To access Jenkins go to browser and run “<PublicIp>:8080” and pate the Jenkins initial password.



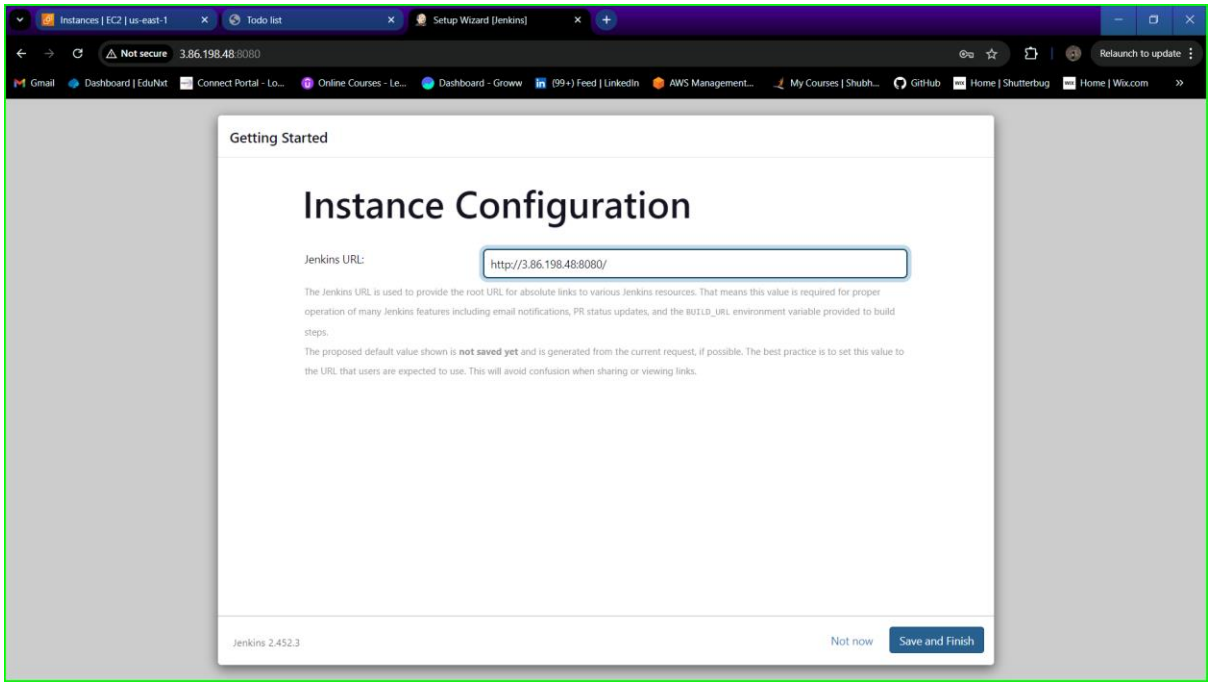
Step 18: Click on Install suggested plugins and let the plugins install.



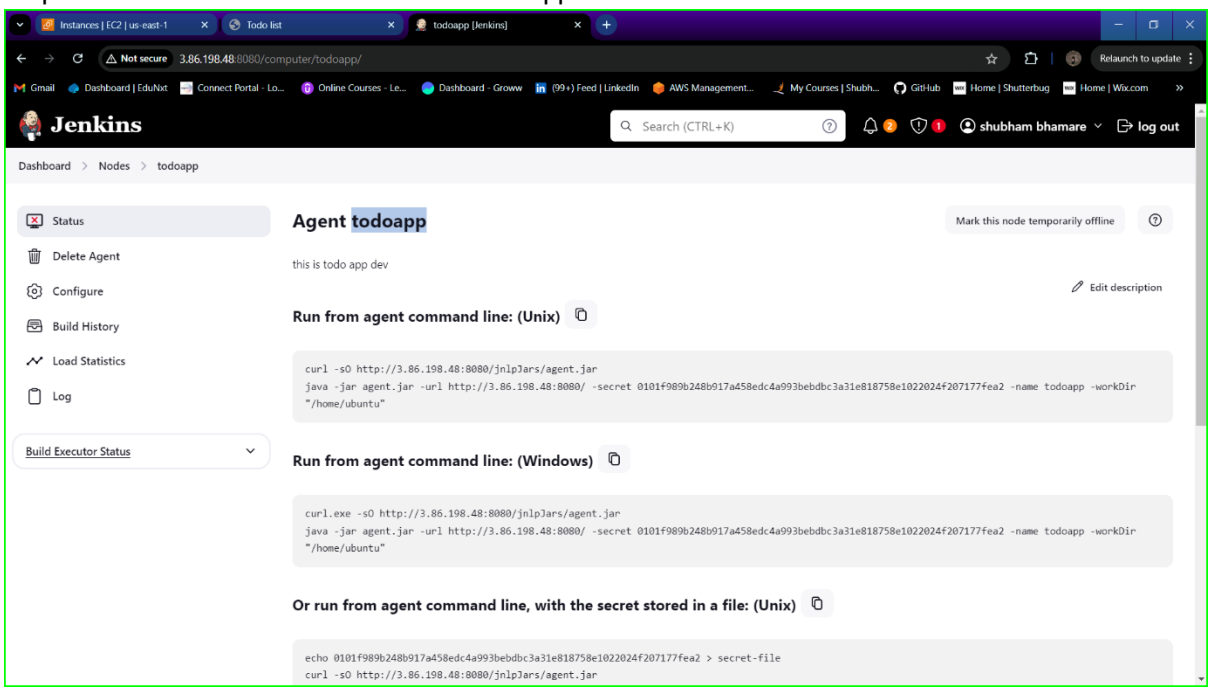
Step 19: Now we have to create User to access Jenkins fill in all details and click on save and continue.



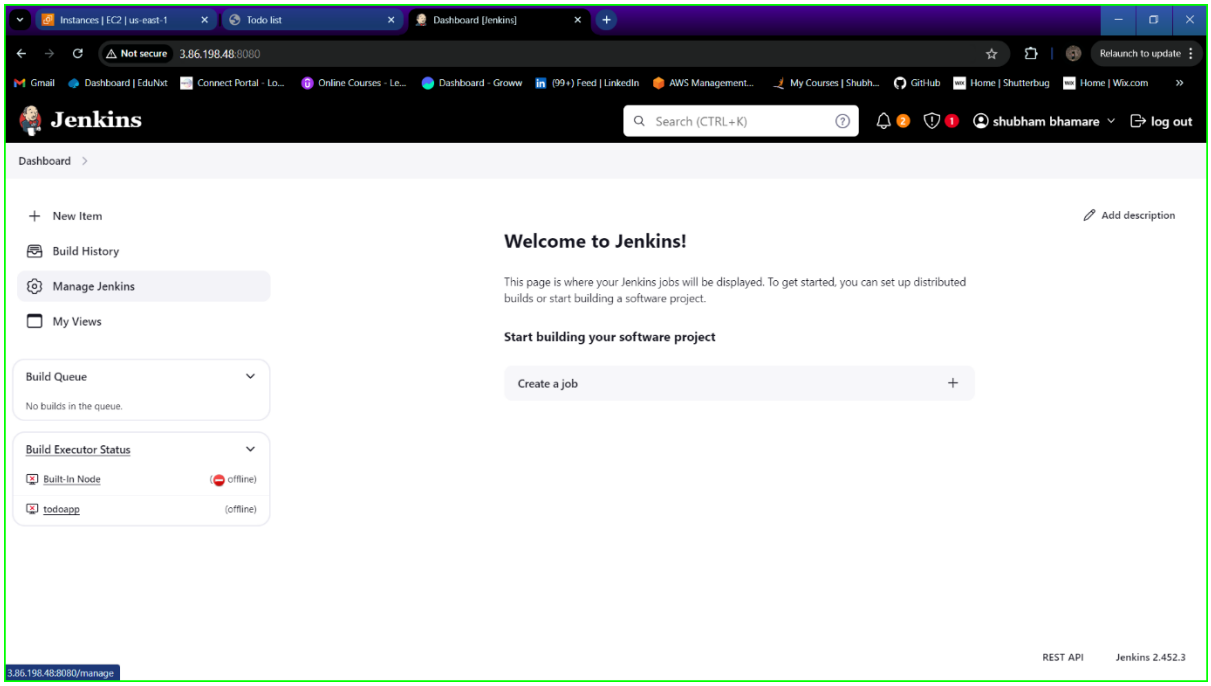
Step 20: Click on save and continue.



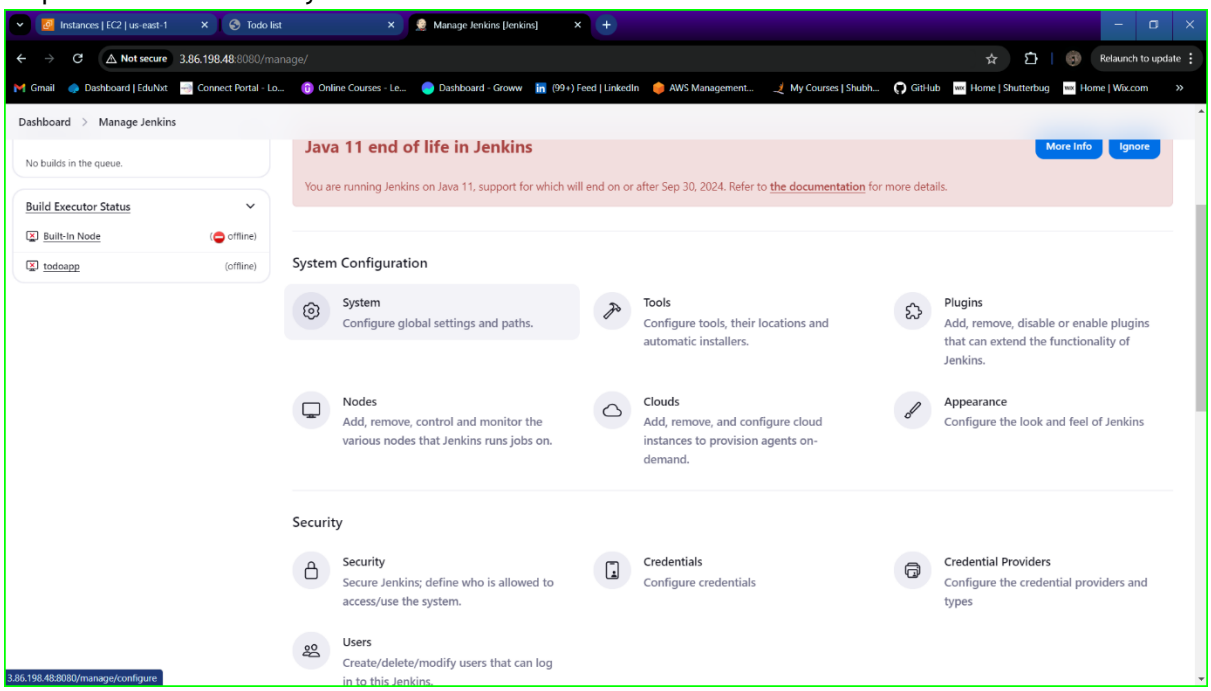
Step 21: Now create one node named “todoapp”.



Step 22: Now you have to add your github credentials to Jenkins. For that you have to go to “manage Jenkins” tab.

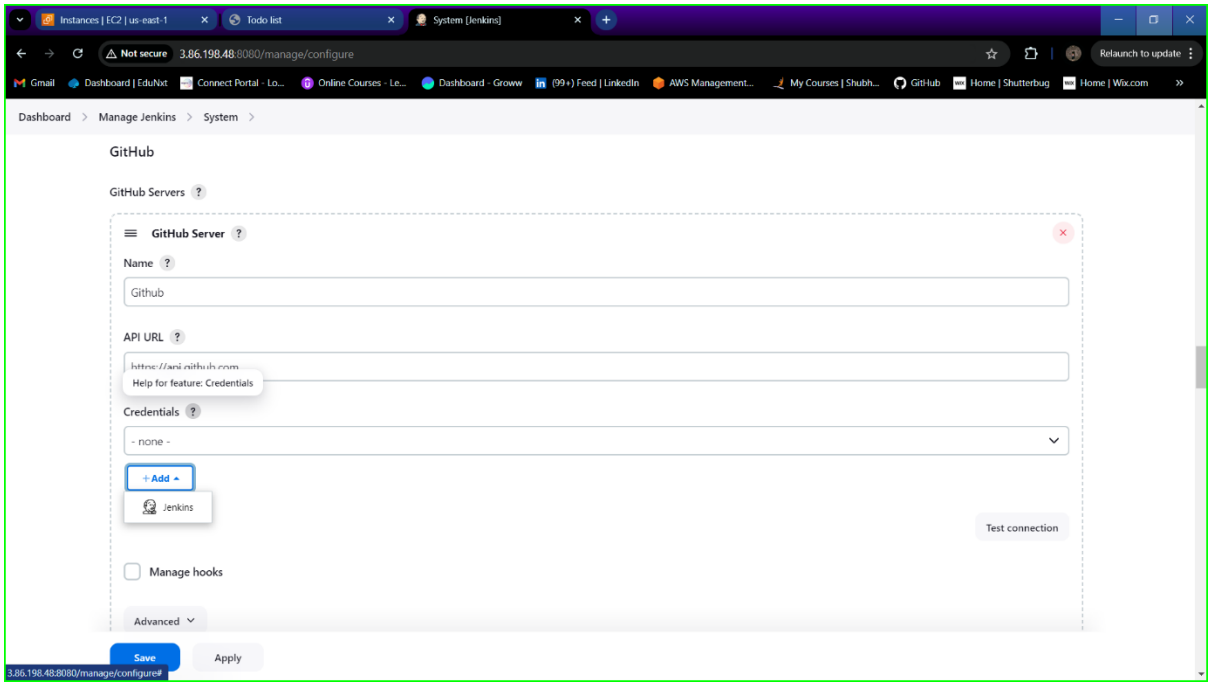


Step 23: Then click on “System”.



Step 24: Scroll down until you find Github section. Then add a github sever,

In credentials click on “Add” and then click on Jenkins.

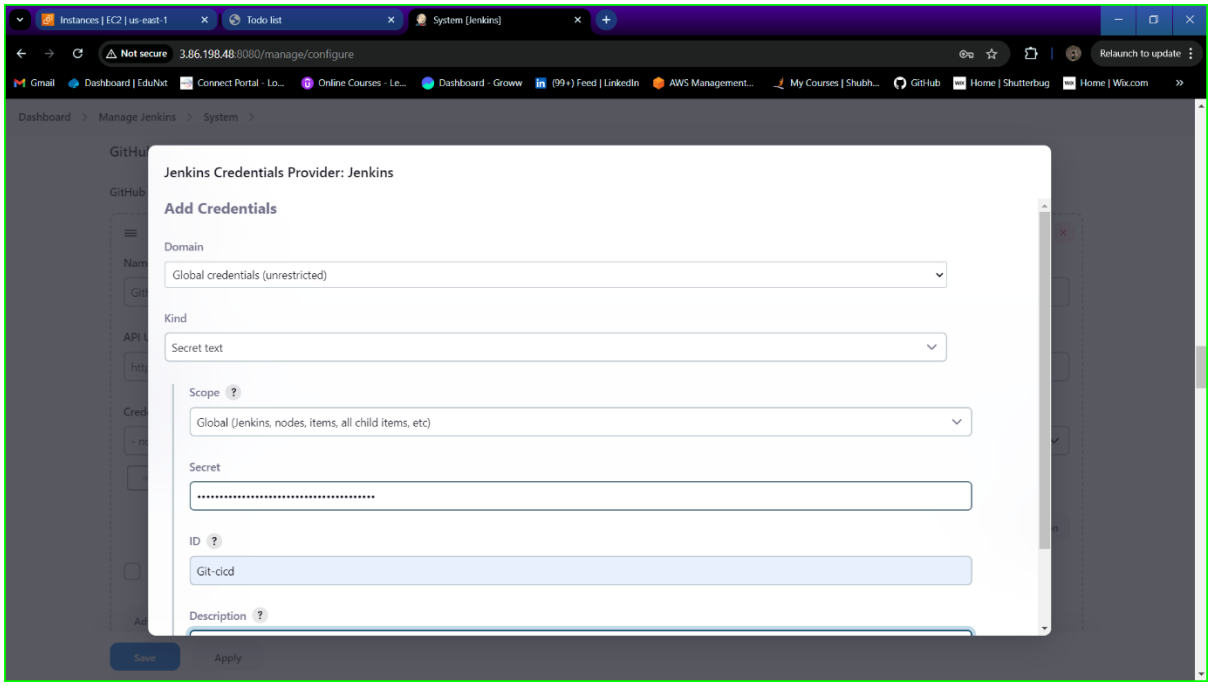


Step 25: Select Kind- Secret Text

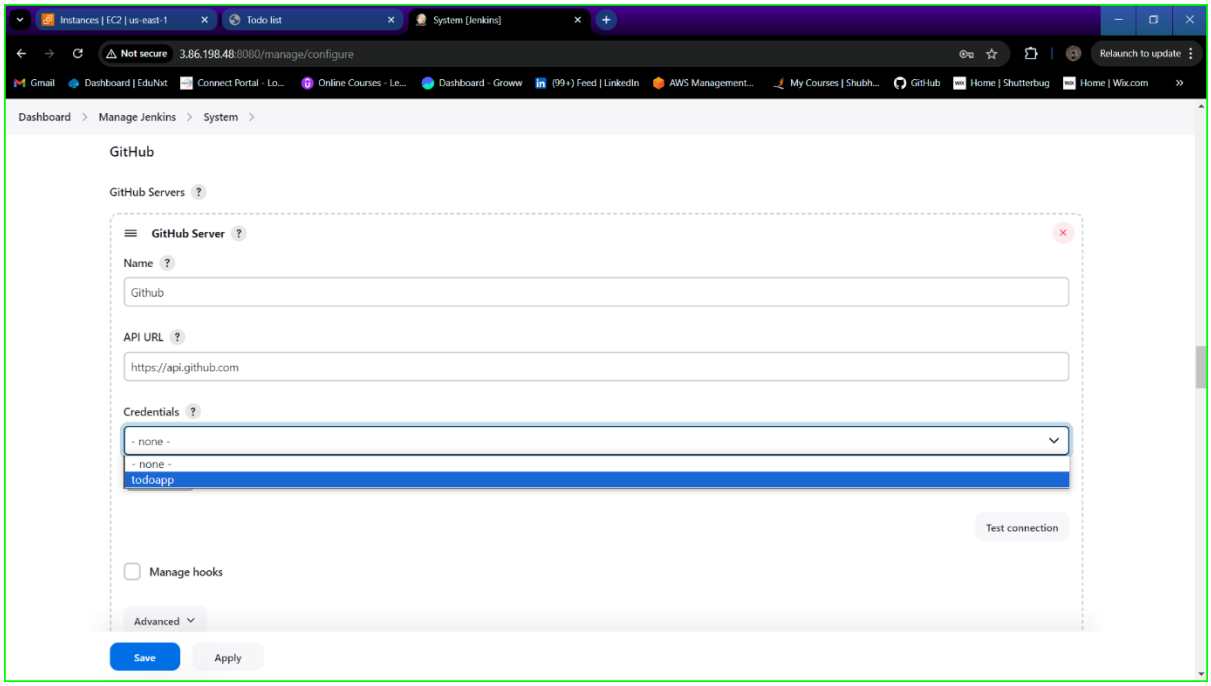
Scope- Global

Secret: Enter your github secret token here.

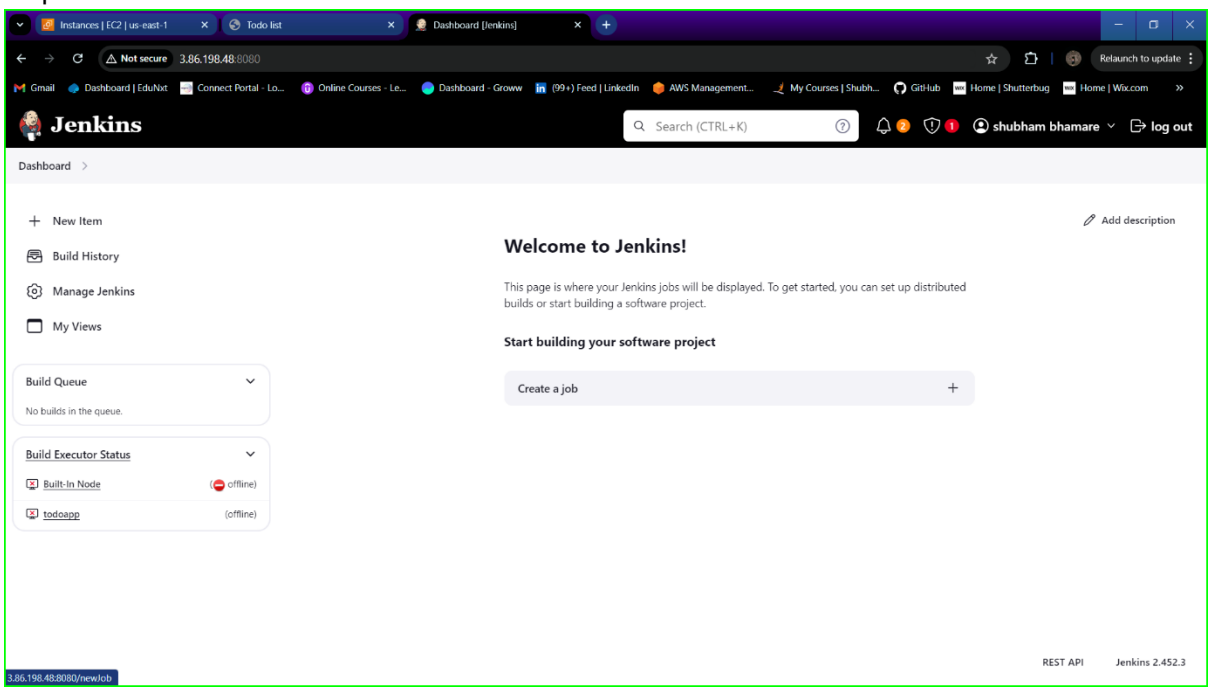
Id- Enter the name to your credentials and then save it.



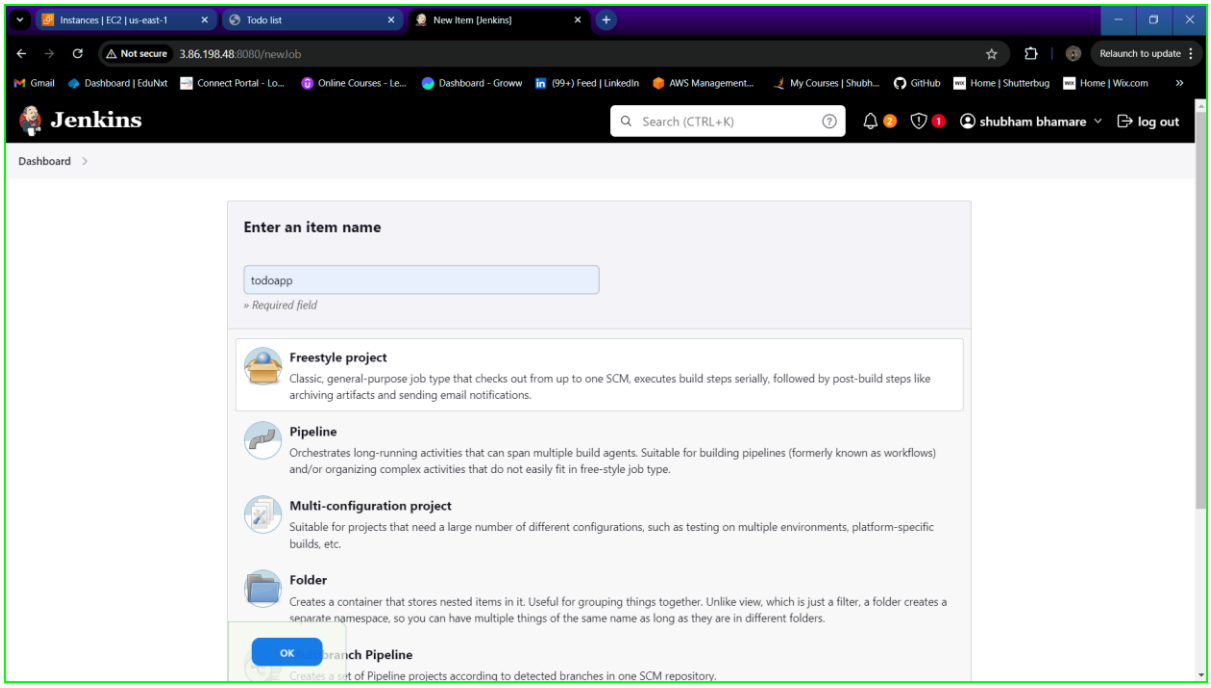
Step 26: Then again in credentials select your credentials and save.



Step 27: Then come to dashboard cand click on “New Item”.

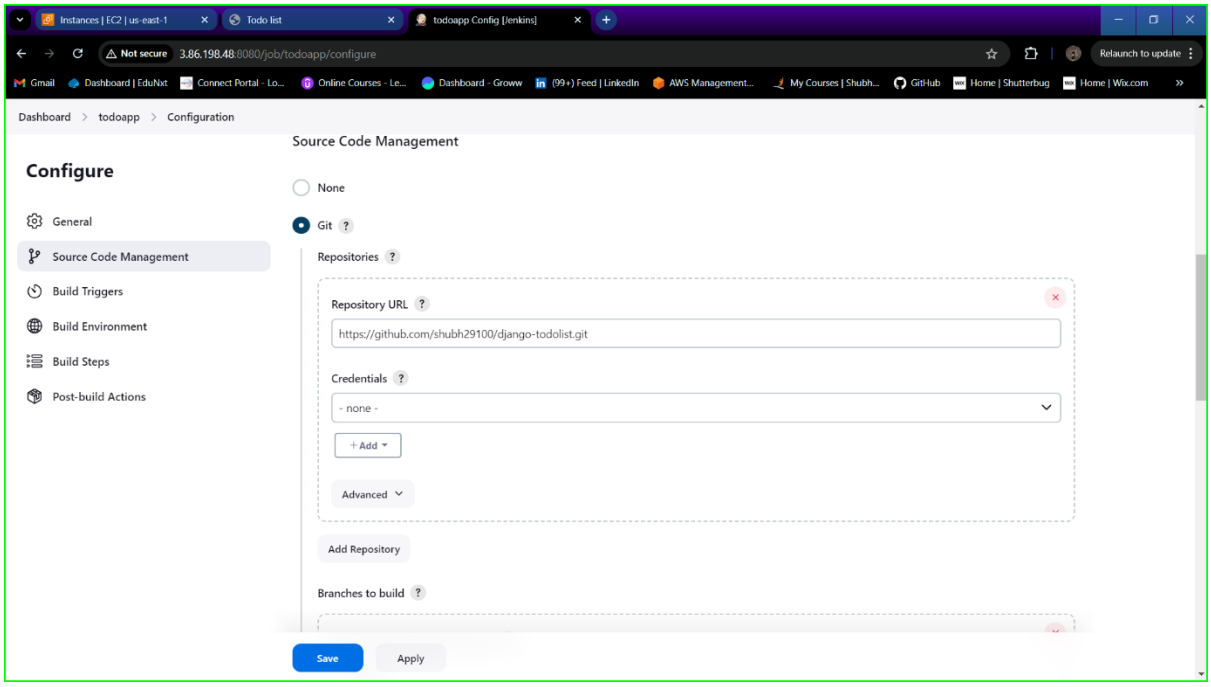


Step 28: Select Freestyle project. Specify the name and click on “ok”.

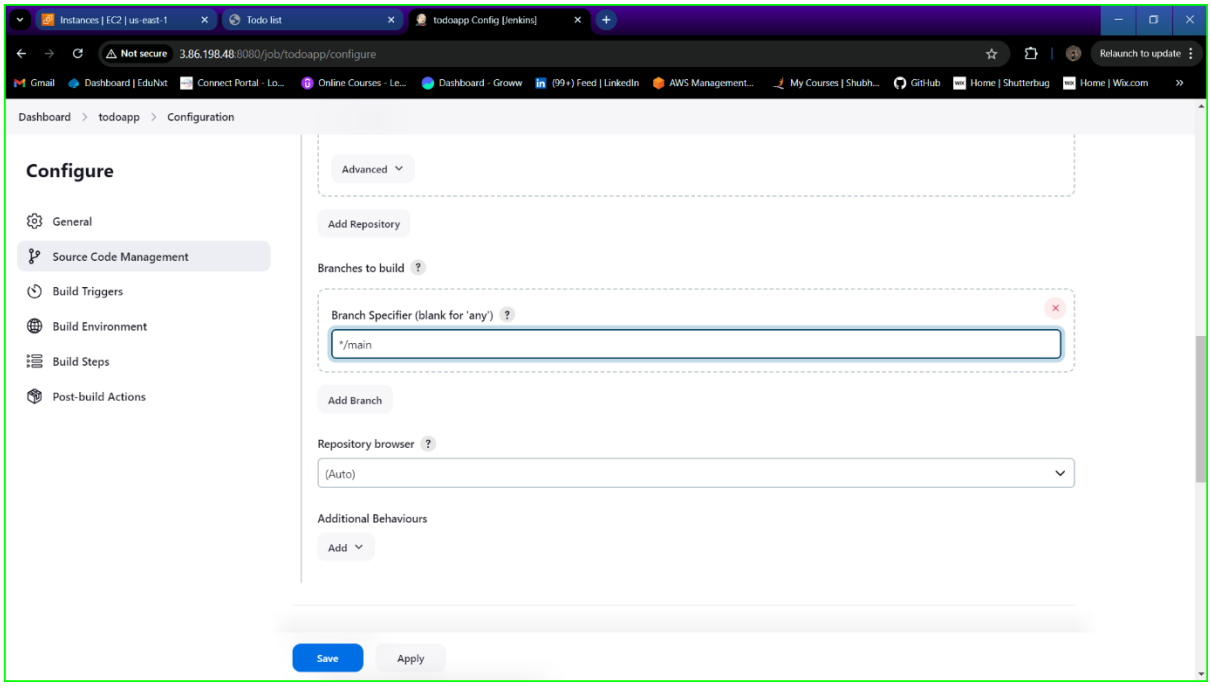


Step 29: Scroll down and paste your github repository link in “Repository URL”.

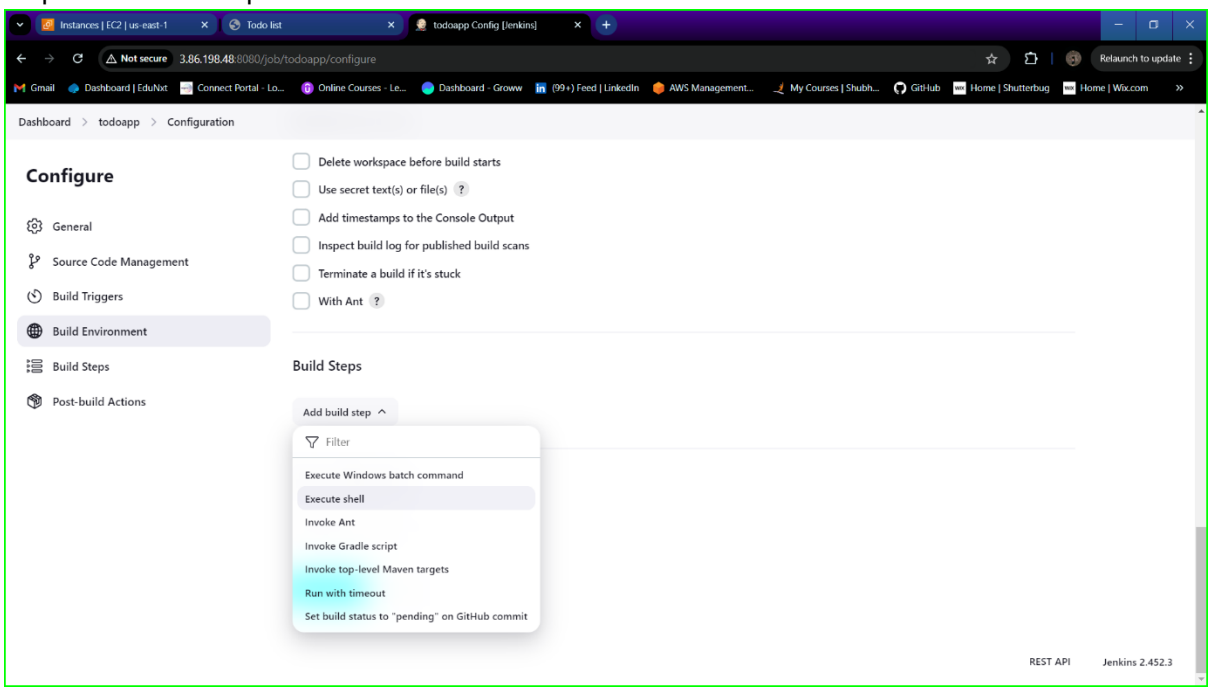
In credentials- leave it blank.



Step 30: Check your branch name.



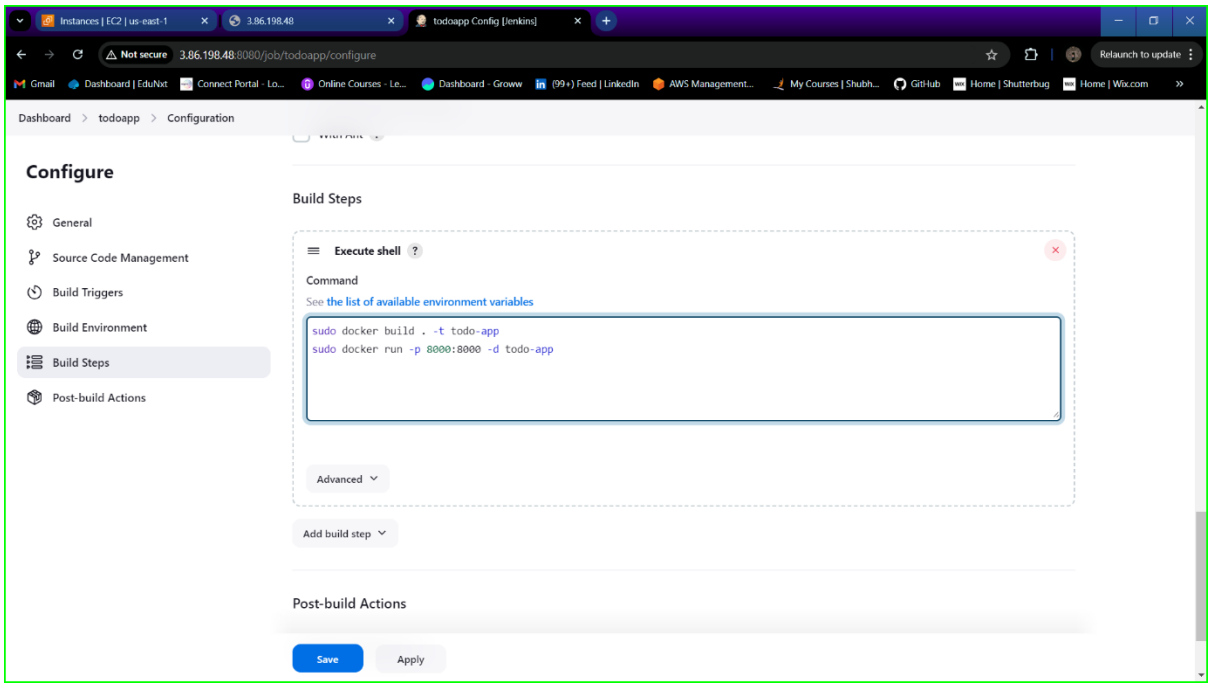
Step 31: In Build Step: Select “Execute Shell”.



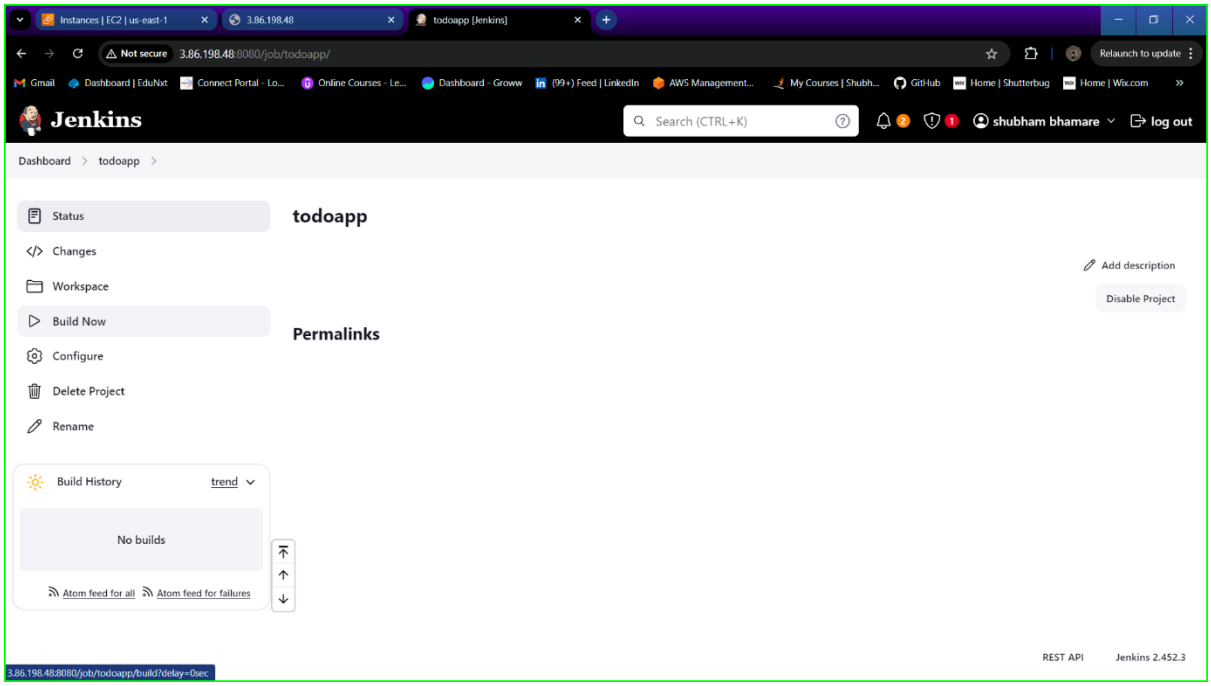
Step 32: Enter the commands

“sudo docker build -t todo-app” and “sudo docker run -p 8000:8000 -d todo-app”.

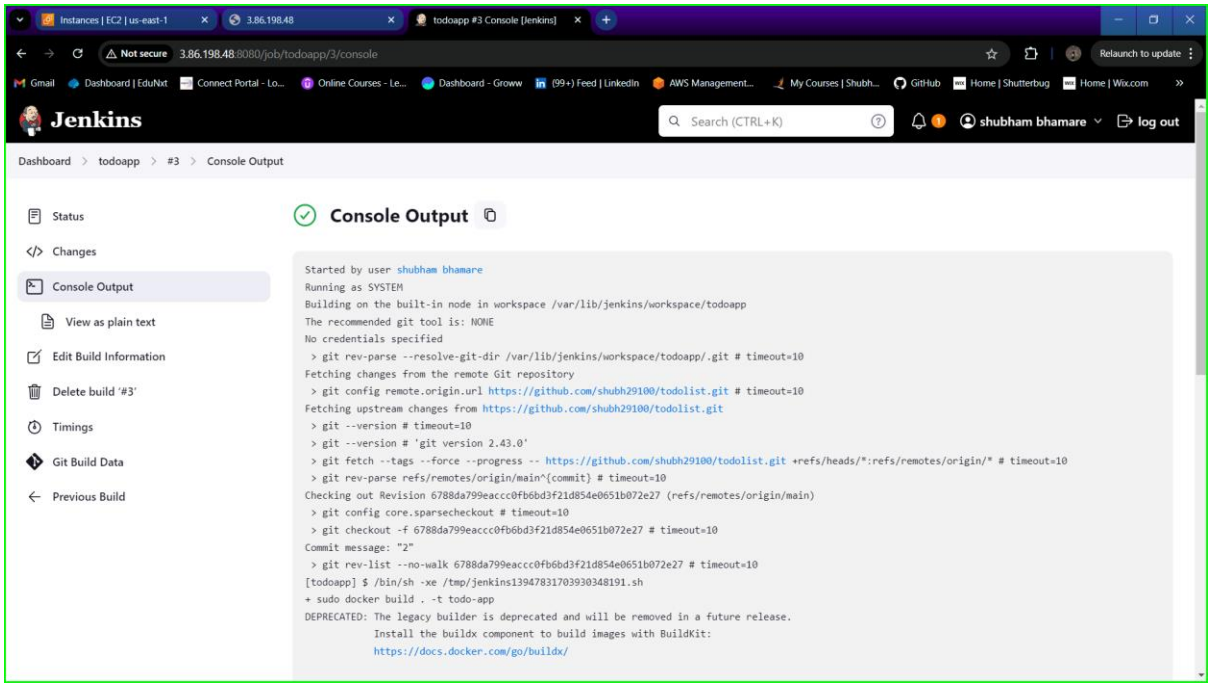
Then click on “save”.



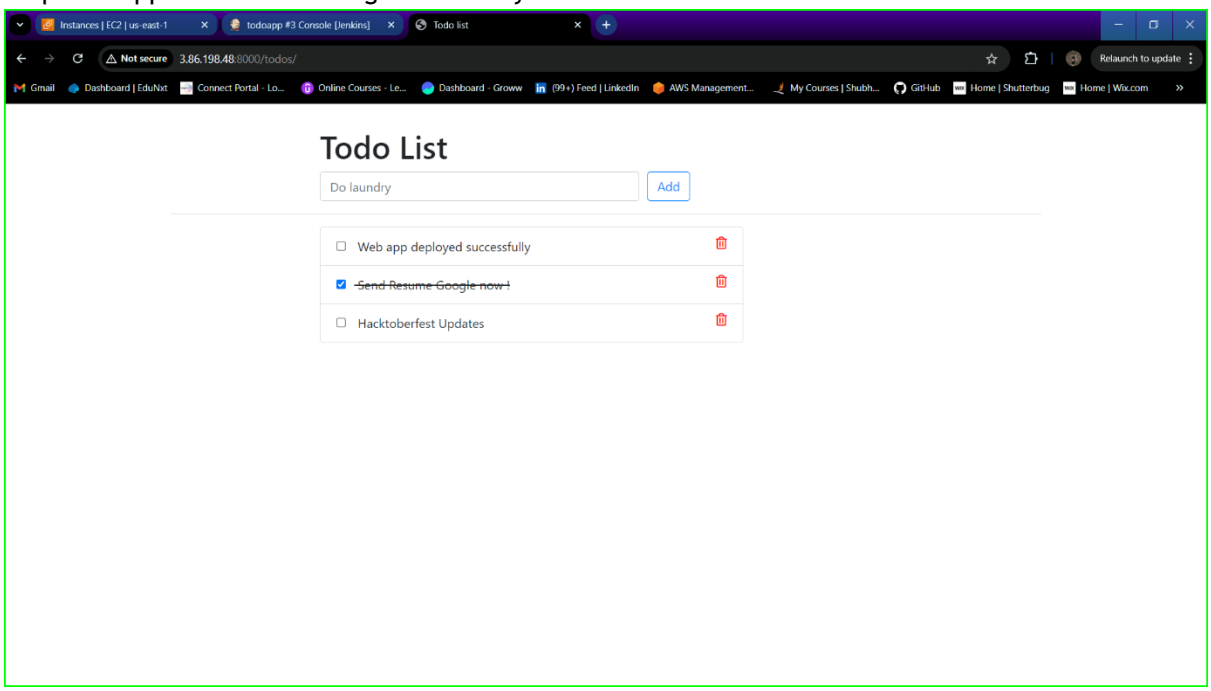
Step 33: Job is created successfully. Now click on build now to build your job.



Step 34: Job is completed successfully.



Step 35: Application is running successfully.



Step 36: You can also check that one docker container is created by Jenkins for your application.