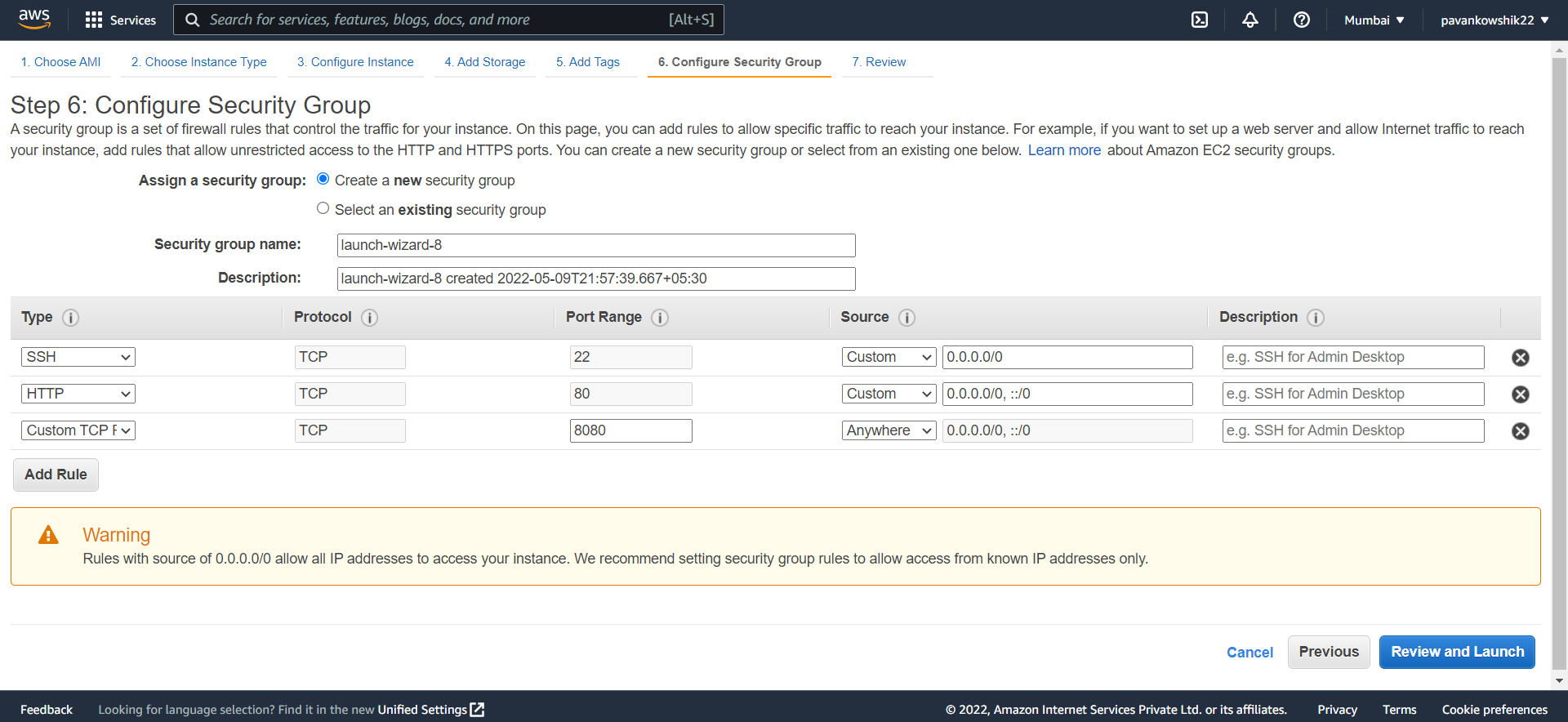
\* Jenikins:- CI/CD(Continious Intergration, Continious Deployement or Delivery)

\*For single change in software, we need to add the image to docker hub and deploye the code, and test the code. This all are done manual way. This is time consumption.

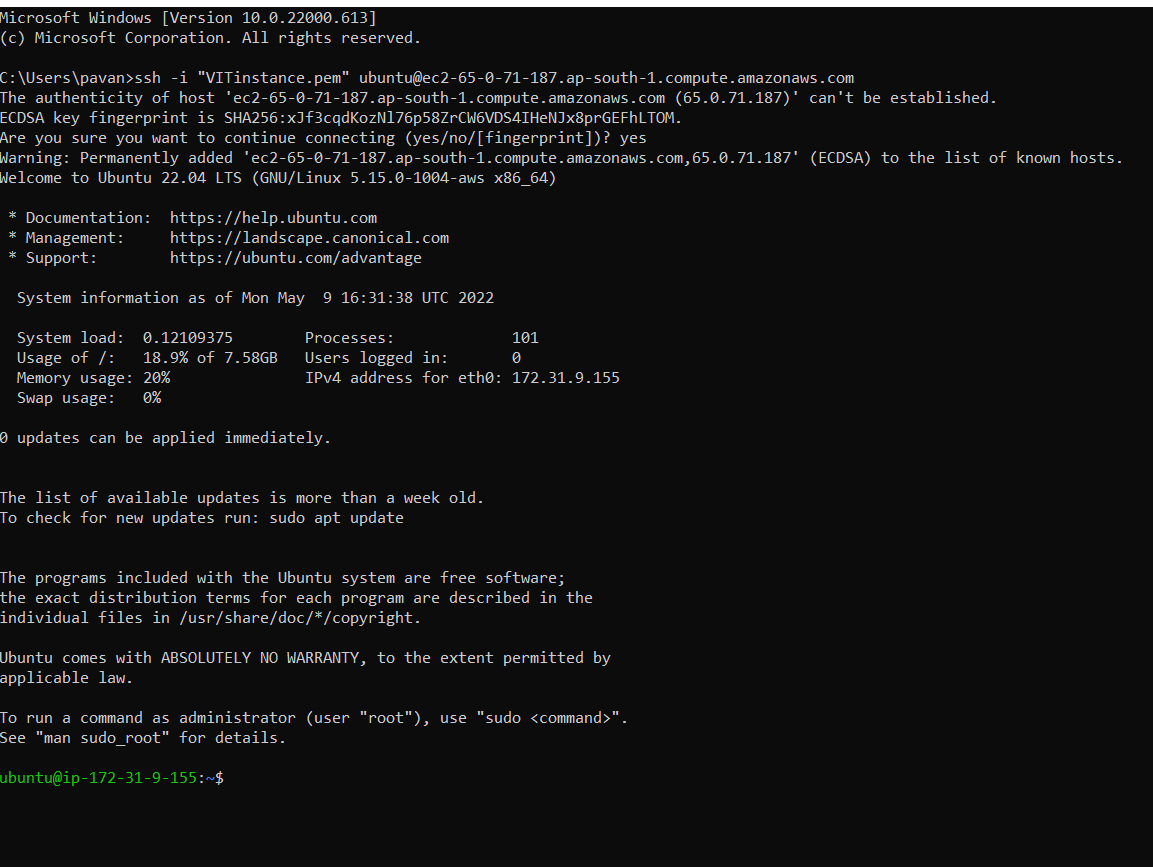
\*Jenkins is a automation Tool

\* Whenever we make change in the code, it has to call the pipeline(Which we build) and in this pipeline all the manual things are automated(Creating Docker image, uploading into docker hub , deploying the code on a server )

1. Login to AWS account and Launch an instance with Ubuntu os and 8080 port number



Connect To command prompt



1. Follow the following commands

$ sudo apt-get update

Now Follow the stepwise process of installing Jenkins in Ubuntu Server:

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-18-04>

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add –

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update

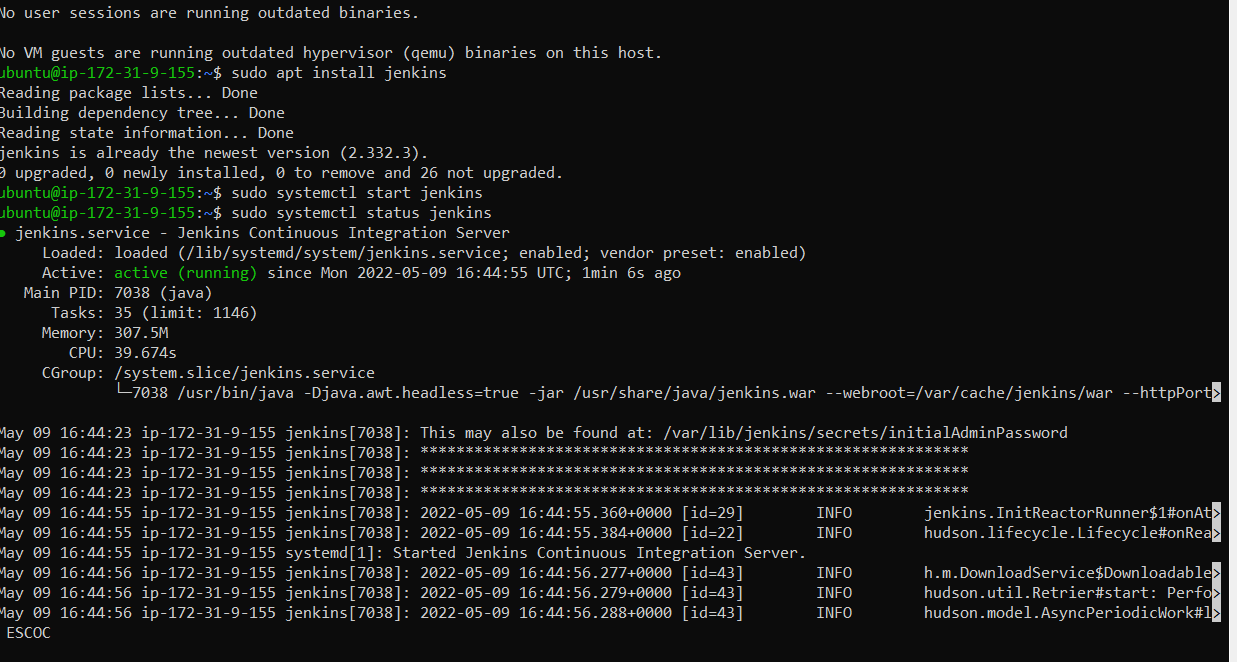
sudo apt-get install default-jdk

sudo apt-get install default-jre

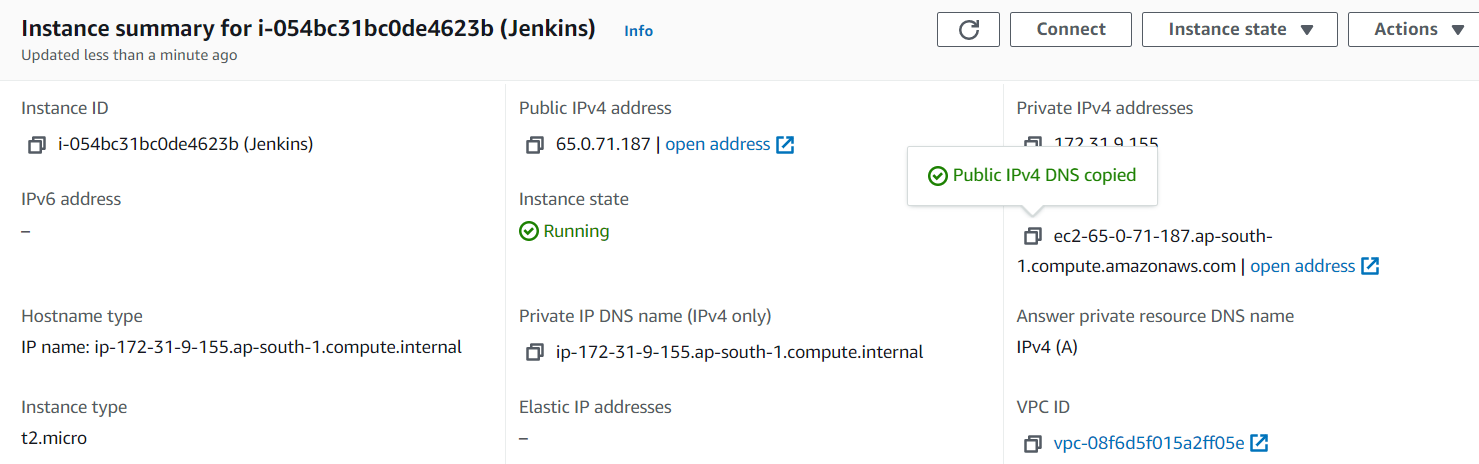
sudo apt install jenkins

sudo systemctl start jenkins

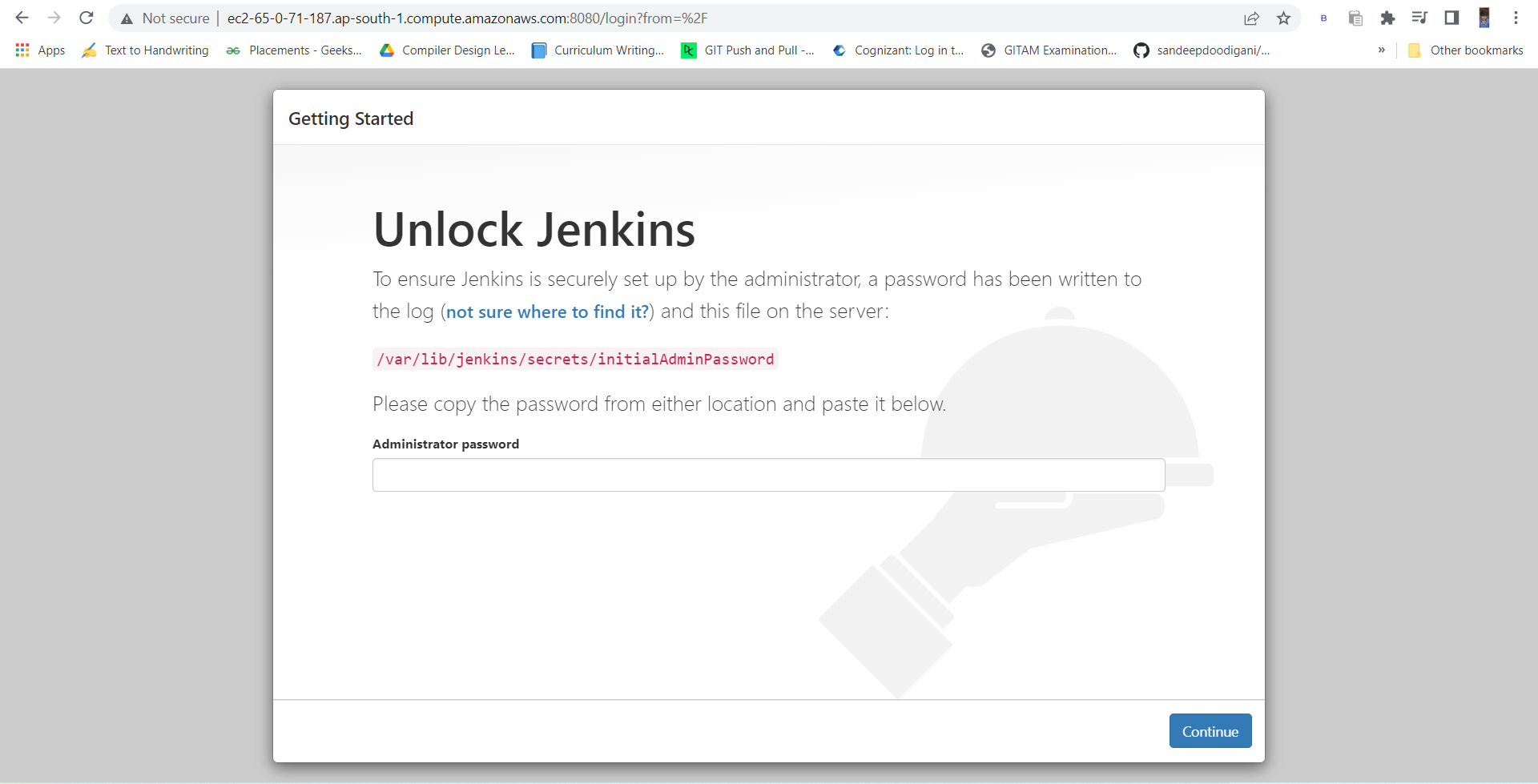
sudo systemctl status jenkins



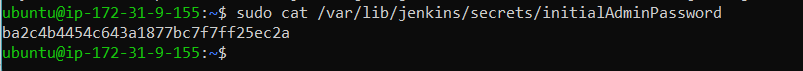
Now open the Public Dns and enter 8080 port number



<http://ec2-65-0-71-187.ap-south-1.compute.amazonaws.com:8080/login?from=%2F>



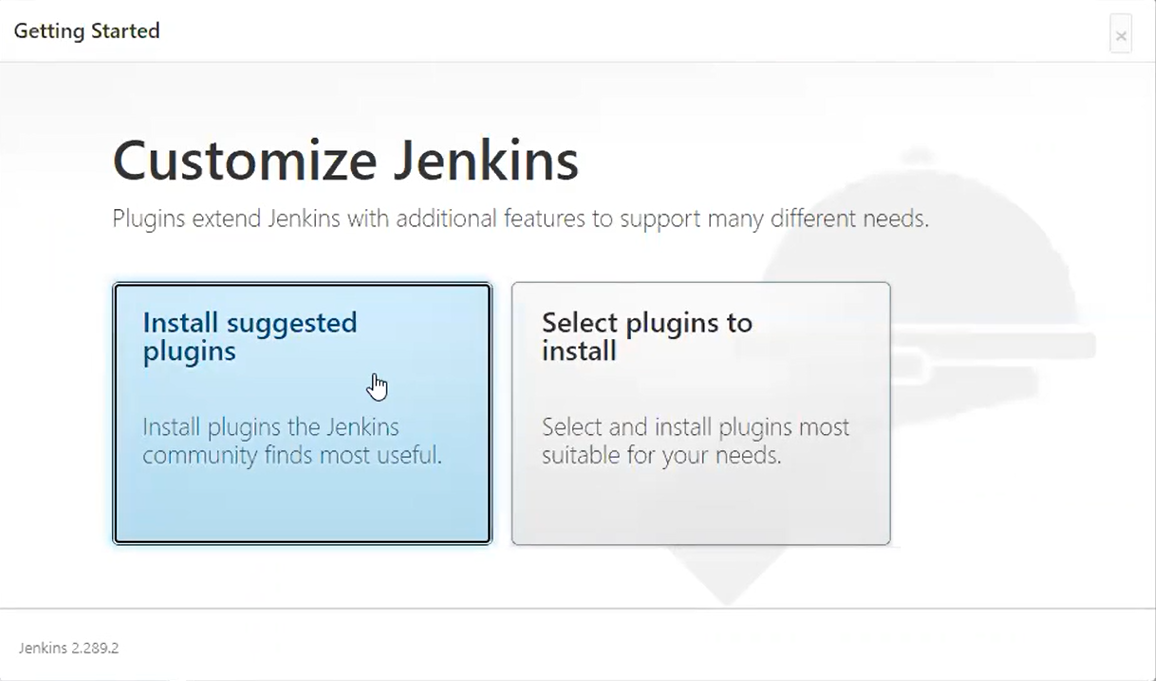
sudo cat /var/lib/jenkins/secrets/initialAdminPassword



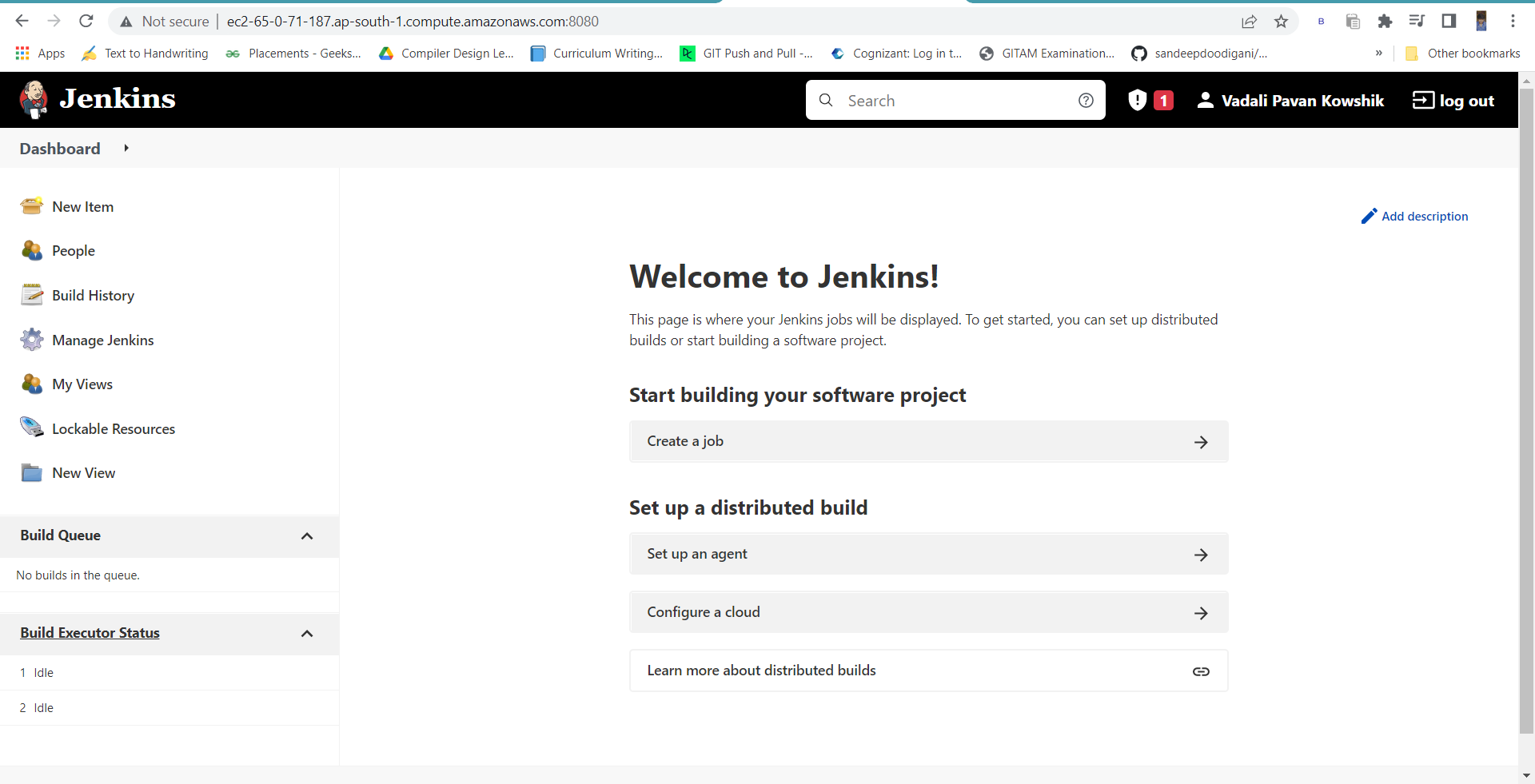
Copy the password and then paste in jenkins

ba2c4b4454c643a1877bc7f7ff25ec2a

1. Now click on install Suggested plugins



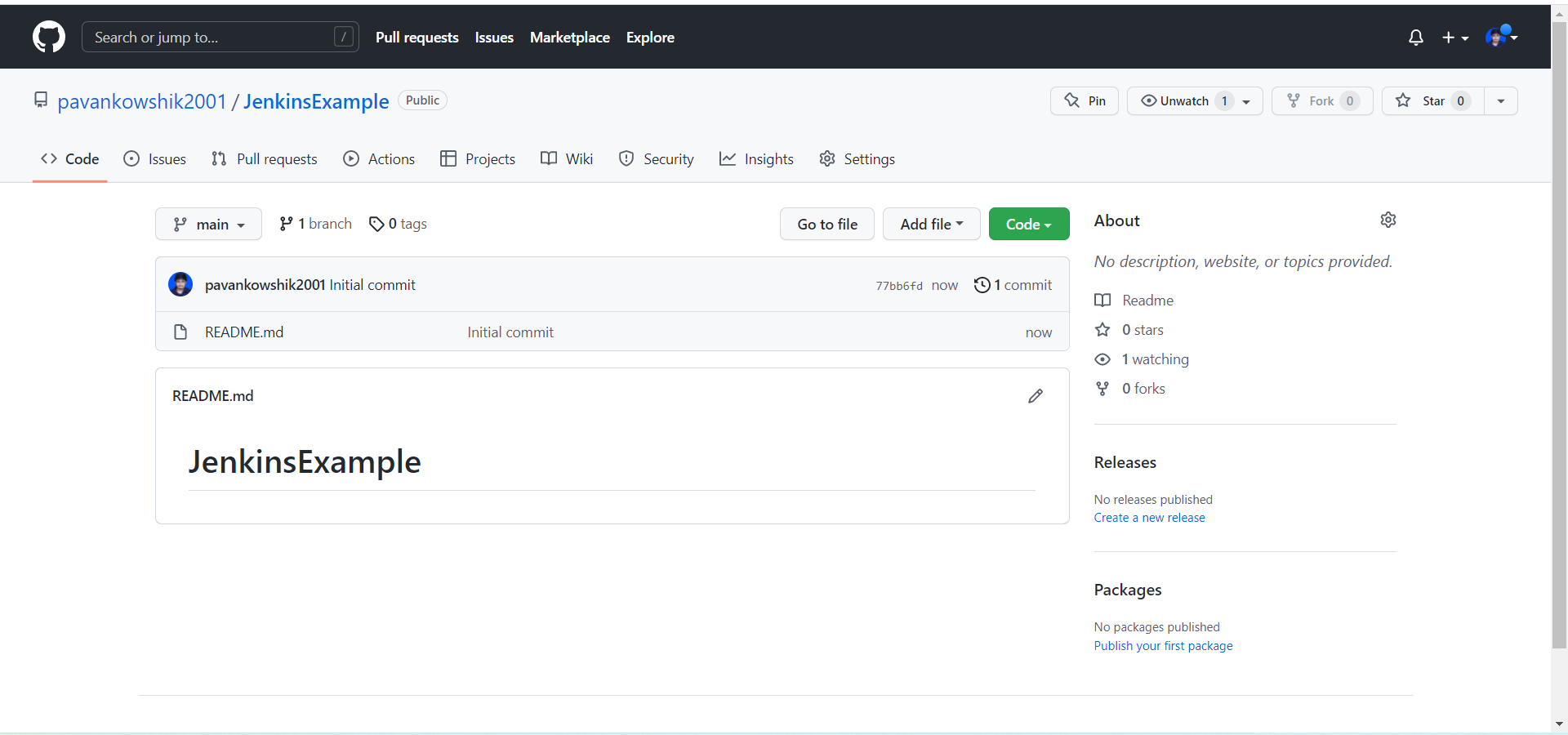
1. Create a Jenkins account username: pavankowshik22



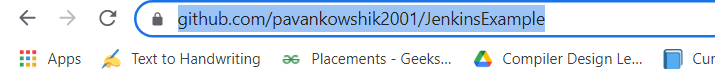
1. Now we need to Integrate the GITHUB to Jenkins

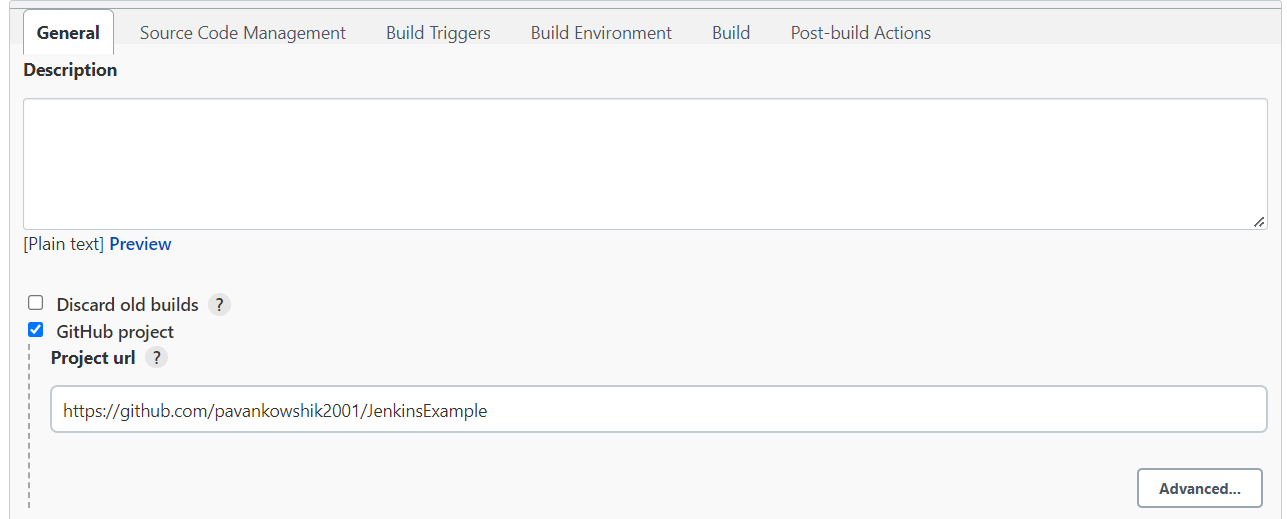
<https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project>

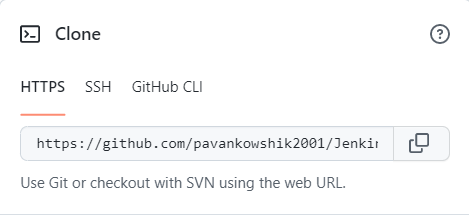
1. Create a GITHUB repository



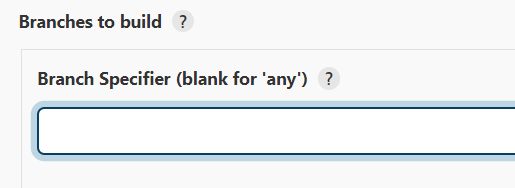
1. Open Jenkins and click on New Item, enter a name “My-pipeline” and select **Freestyle project and click on OK**
2. Now apply the following on Jenkins

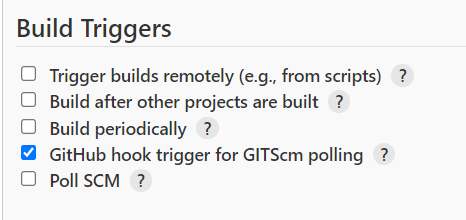




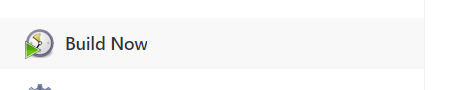






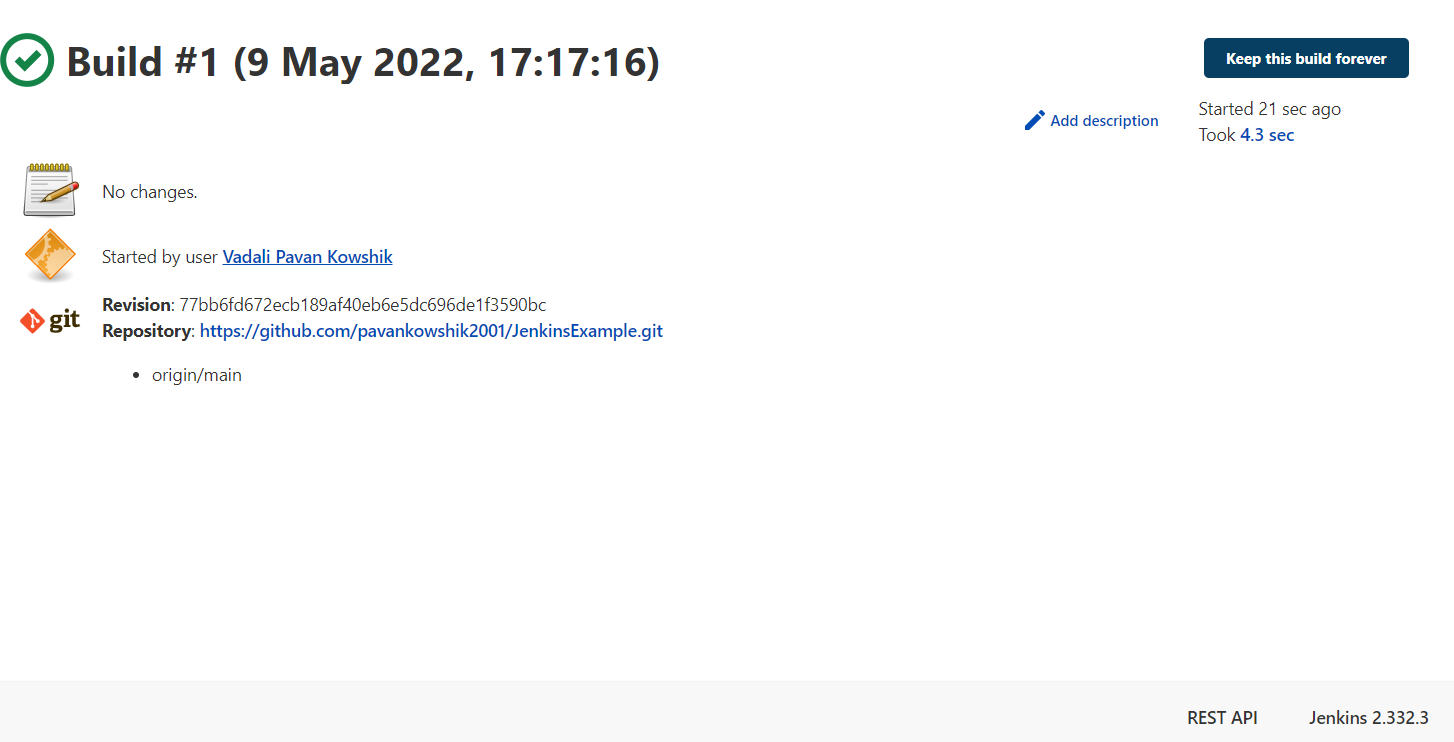


1. Click on Build Now

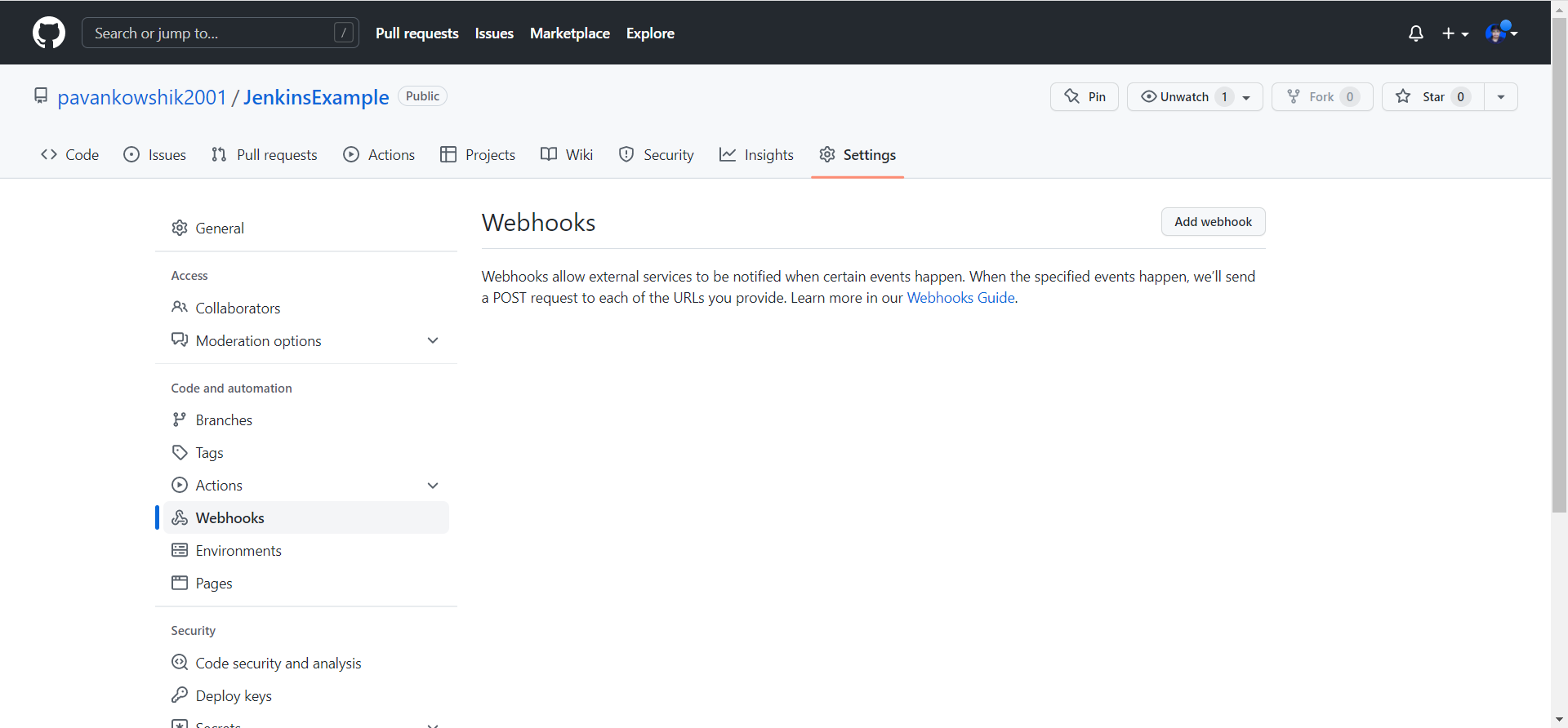


Pipelines are execution

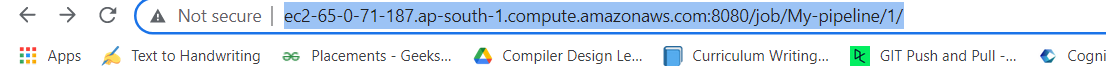


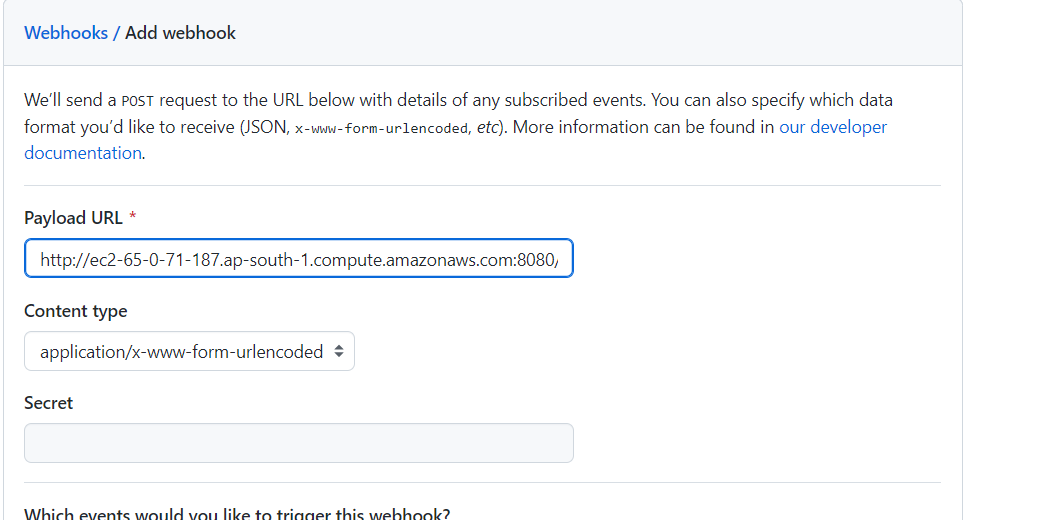


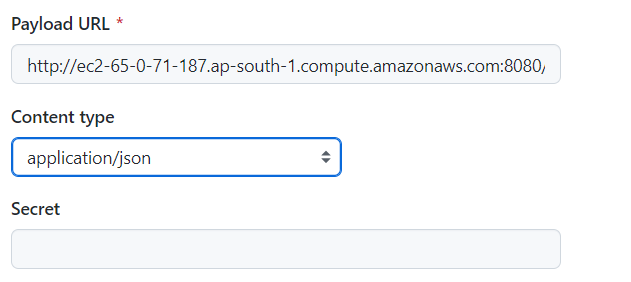
1. Now we are going to add WEBHOOKS(it tells the user any changes to the server)

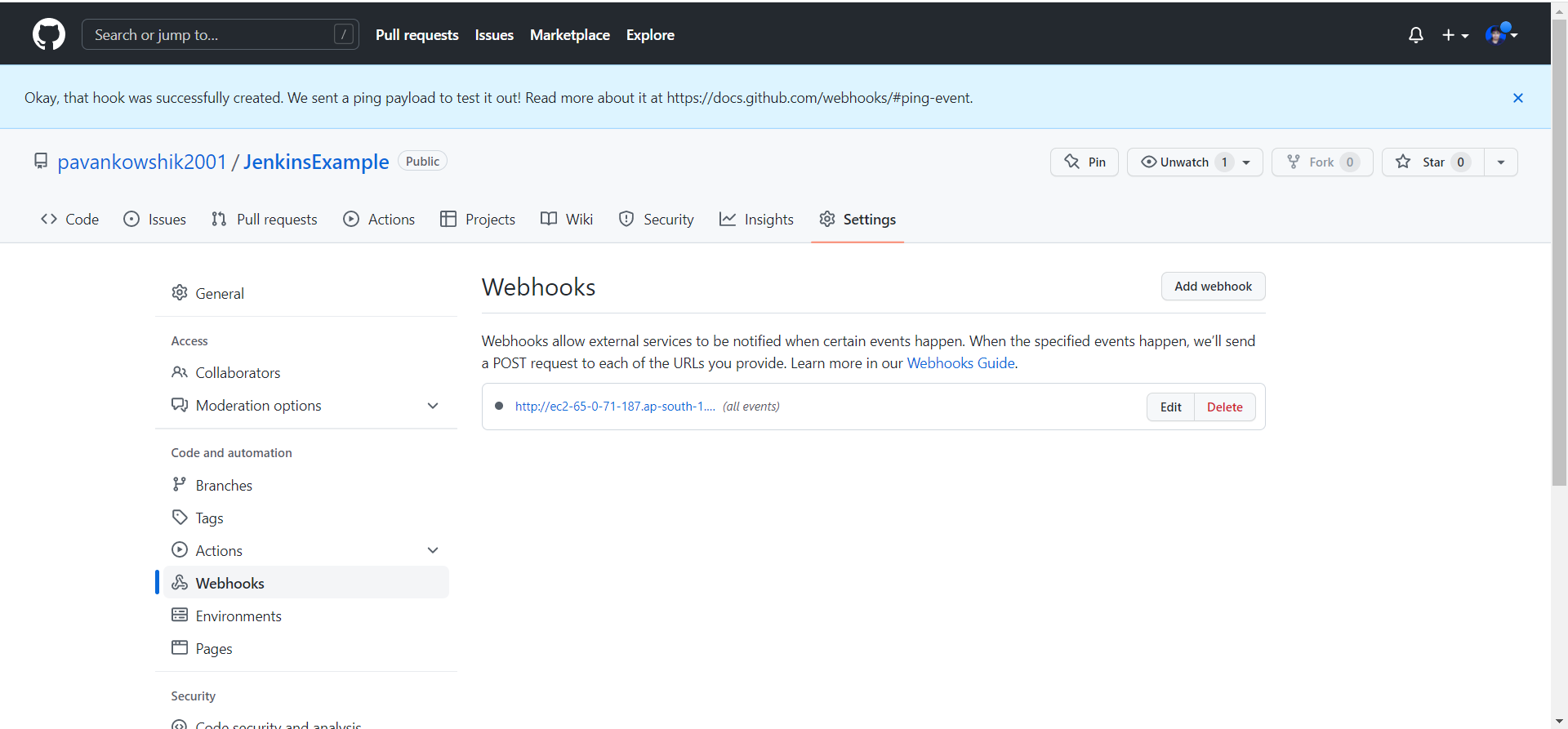


1. Add the Jenkins URL http://ec2-65-0-71-187.ap-south-1.compute.amazonaws.com:8080/github-webhook/

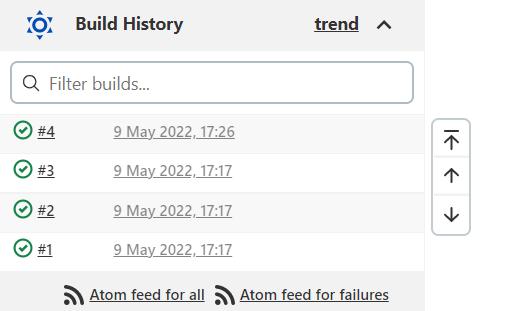




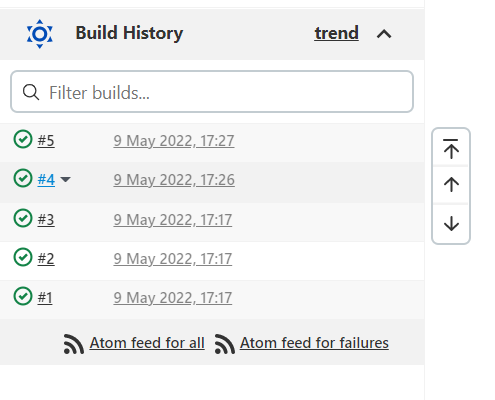




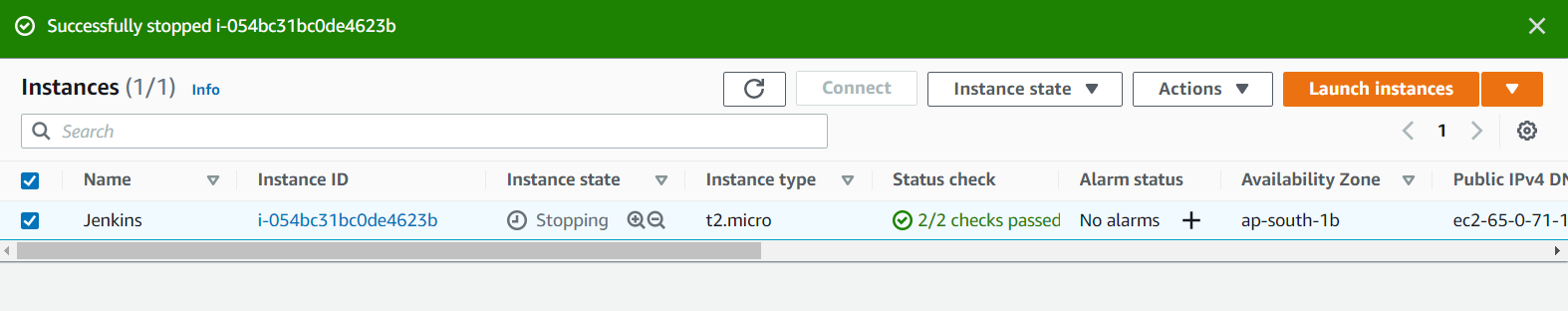
Changes are a



Change in GIT REPOSITORY changes in the Jenkins

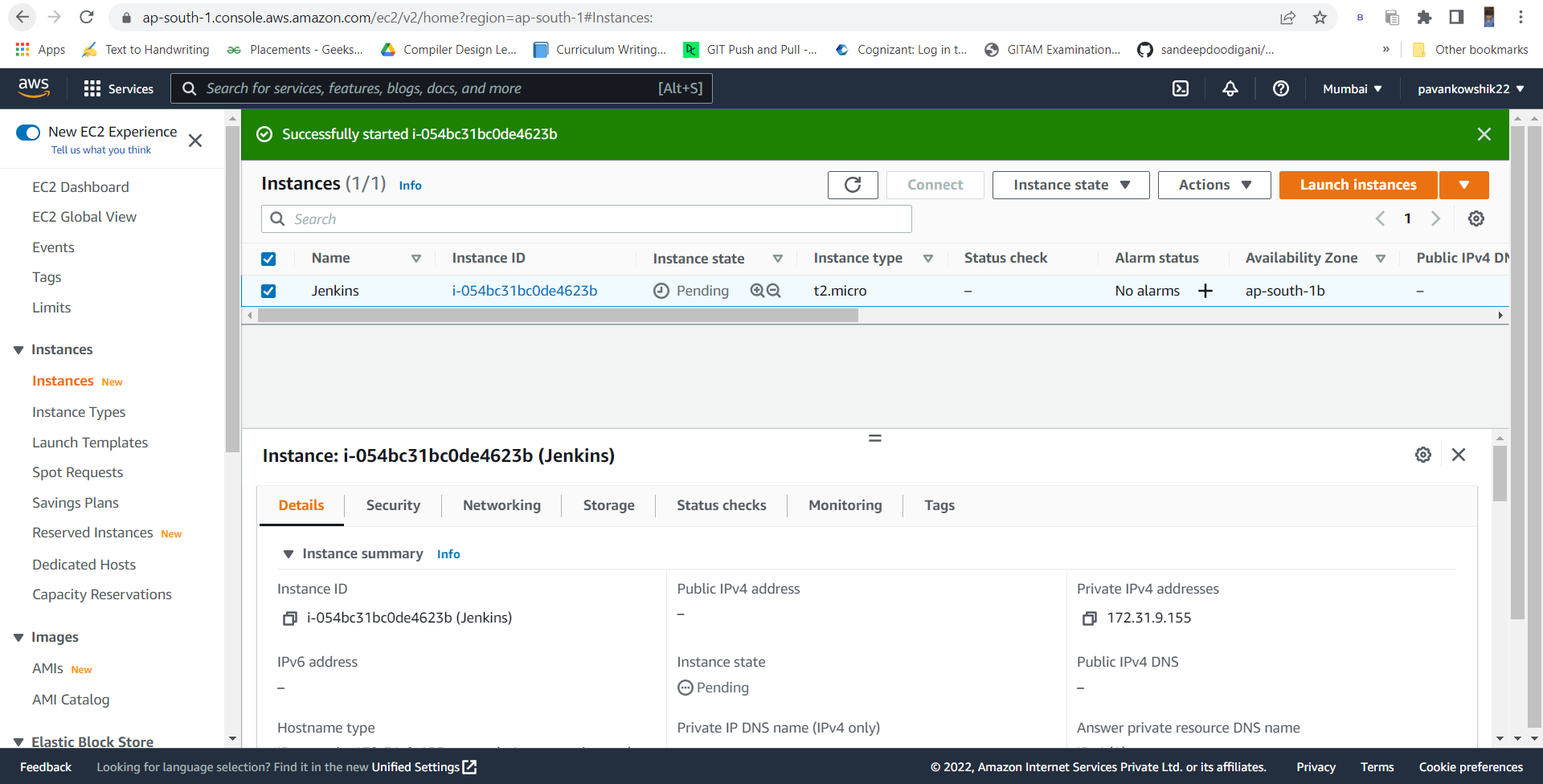


Stoping the instance

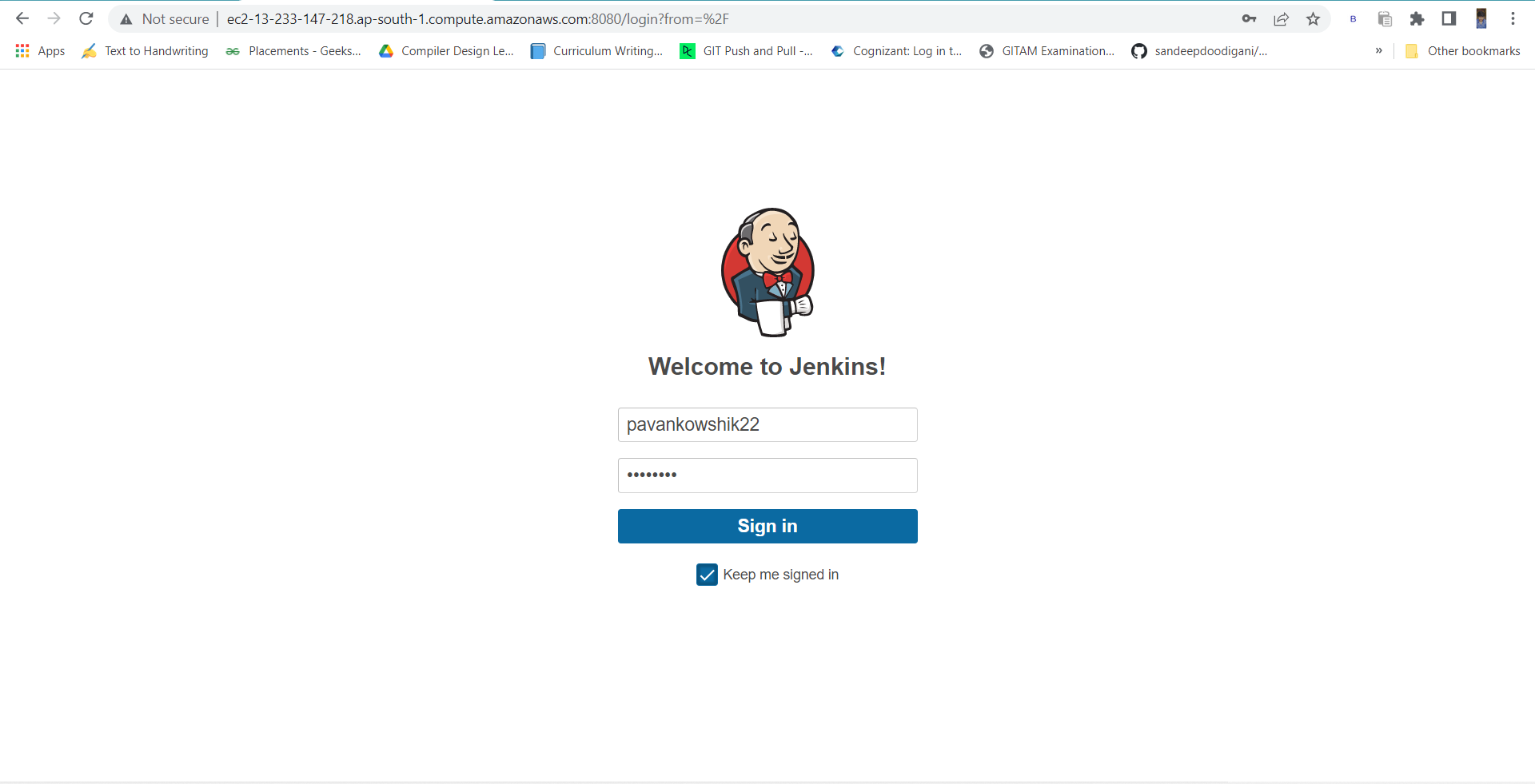


Day 2 Creating Docker Pipeline

1. Run the Jenkins instance on Aws EC2 instance



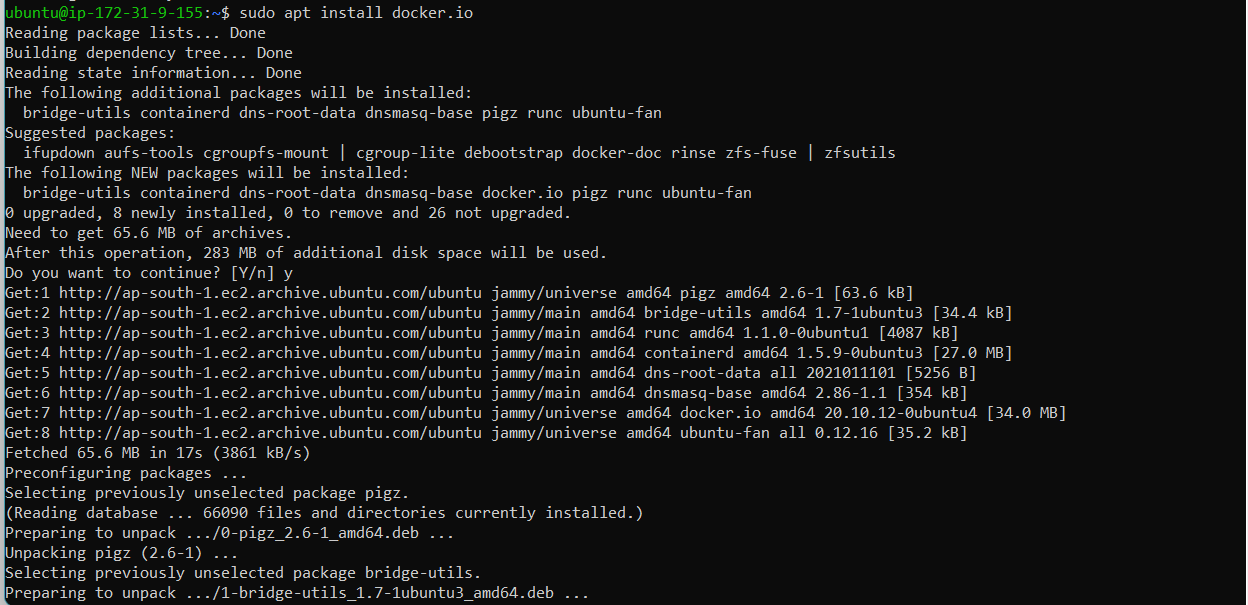
1. Now Use Ipv4 address and login to Jenkins with username and password



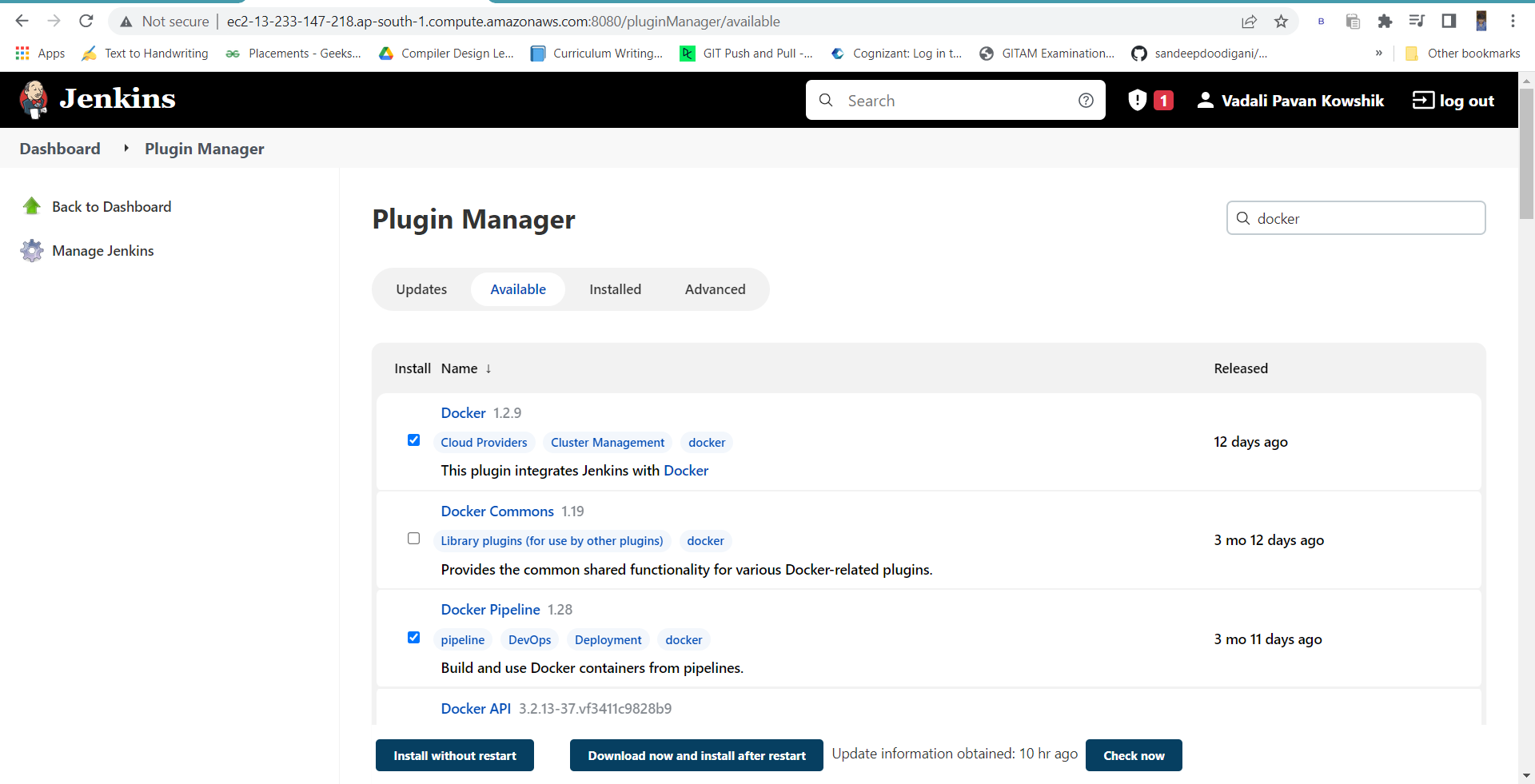
1. Now we are going to install docker in the Ubuntu EC2 instance , so as to create a pipeline

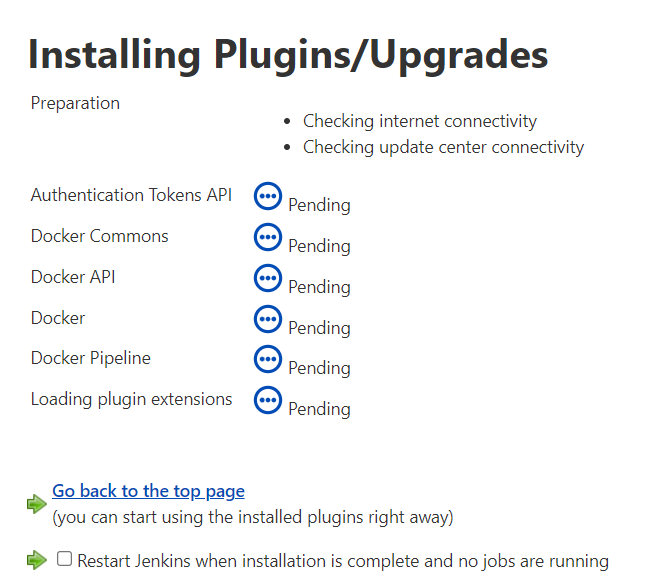


1. $ sudo apt install docker.io



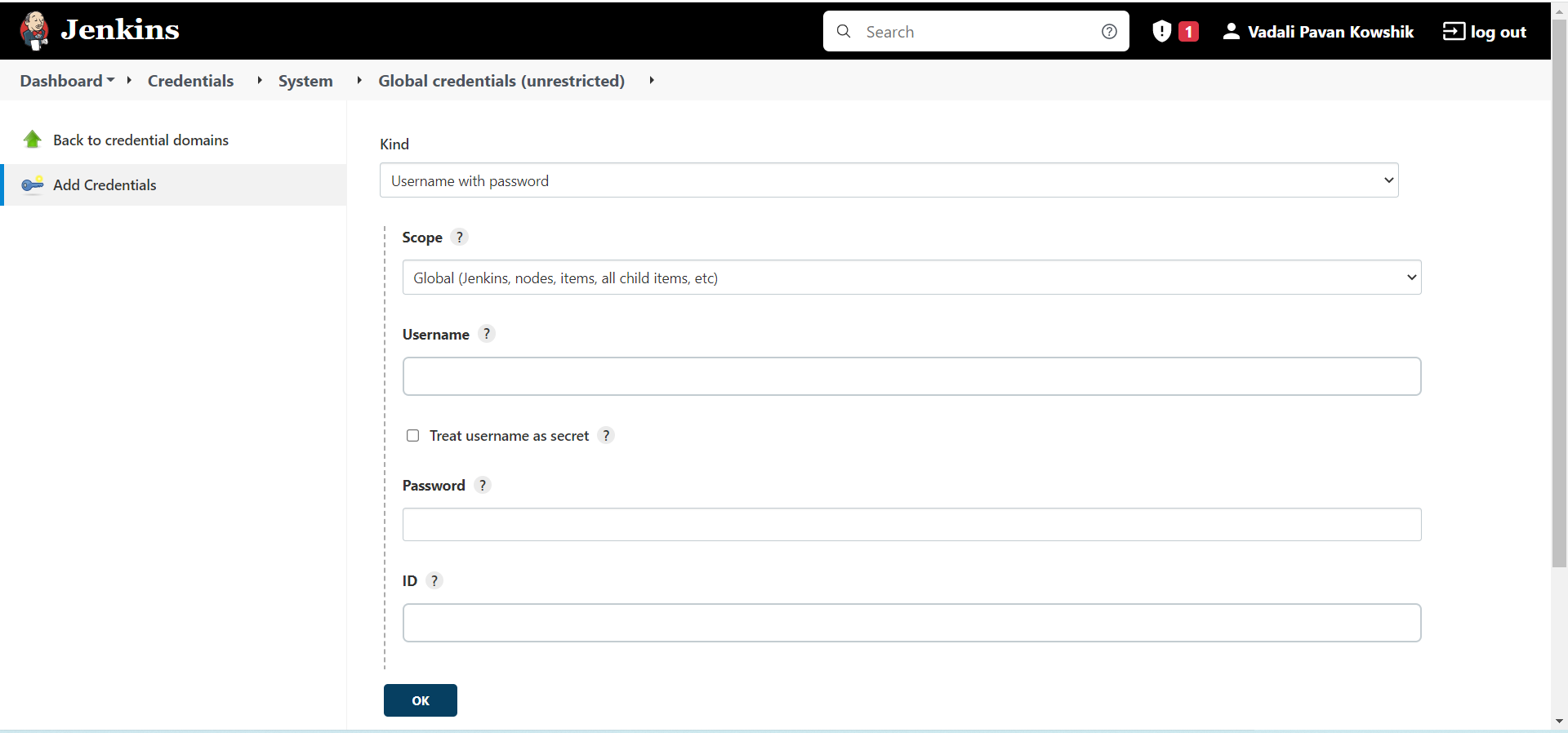
1. $ sudo usermod -aG docker jenkins
2. Sudo chmod 777 /var/run/docker.sock
3. Now we are going to add docker plugin in Jenkins (Which is similar like adding GITHUB piugins) and click on install without restart

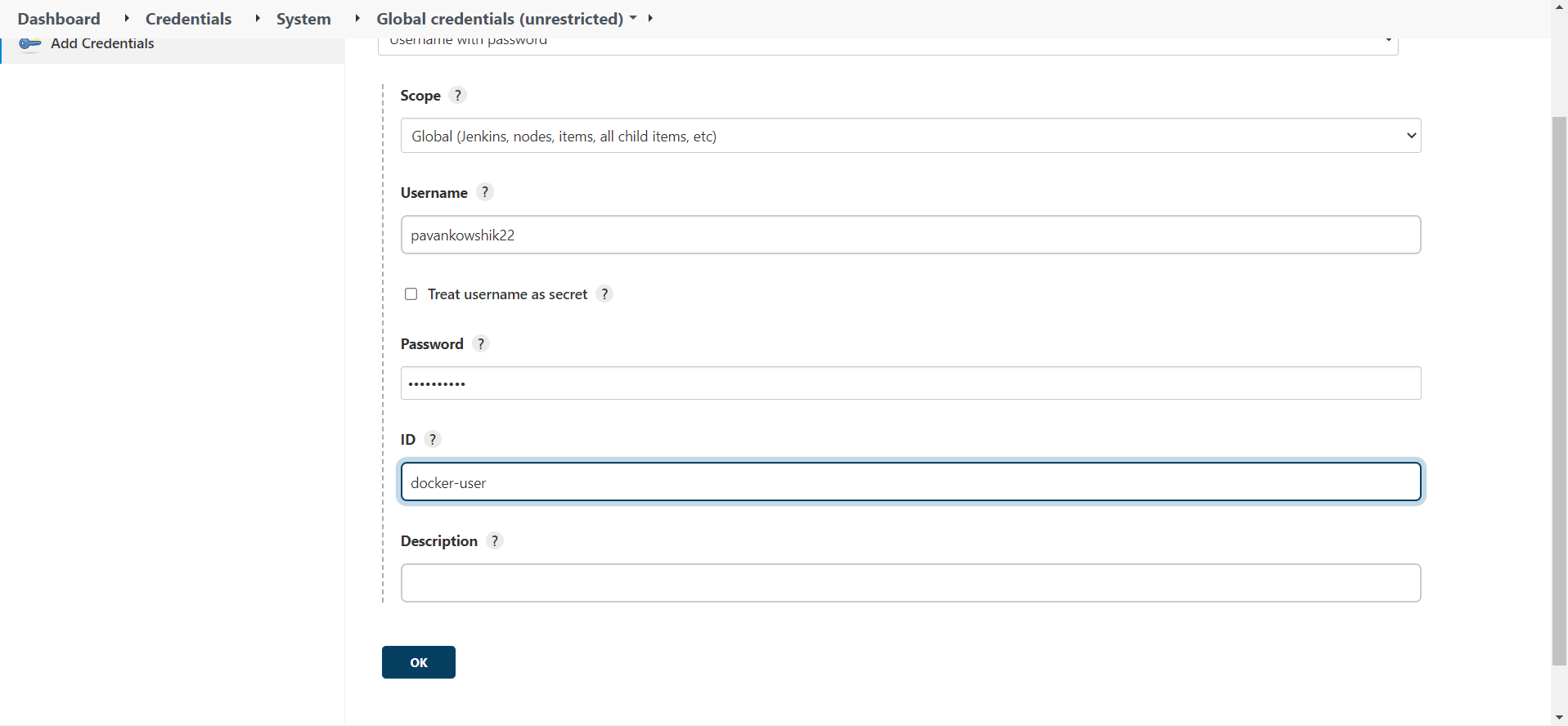




1. Now we are going to pull the image to Docker hub. For that we need to login to dockerHub.

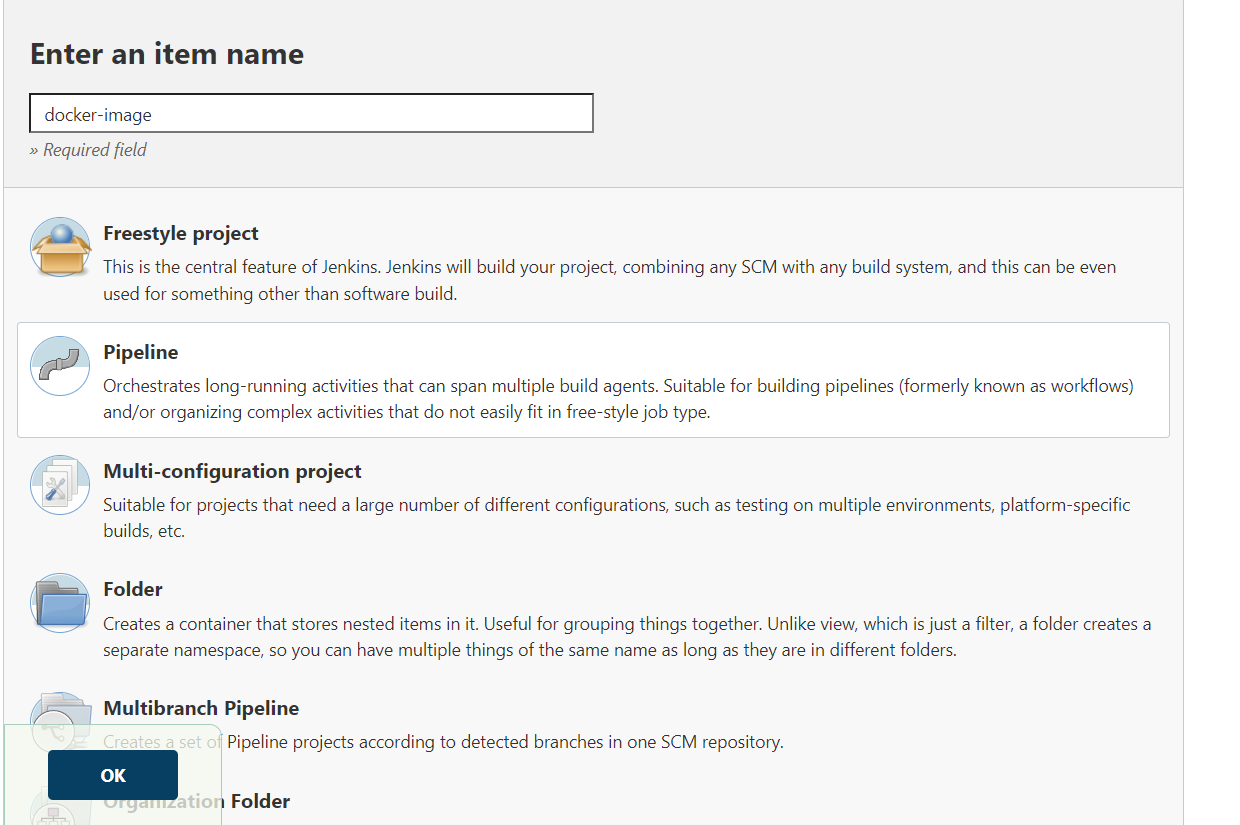
Go to Manage Jenkins and click on Manage Credentials, click on Jenkins, click on [Global credentials (unrestricted)](http://ec2-13-233-147-218.ap-south-1.compute.amazonaws.com:8080/credentials/store/system/domain/_), and click on Add Crendentials



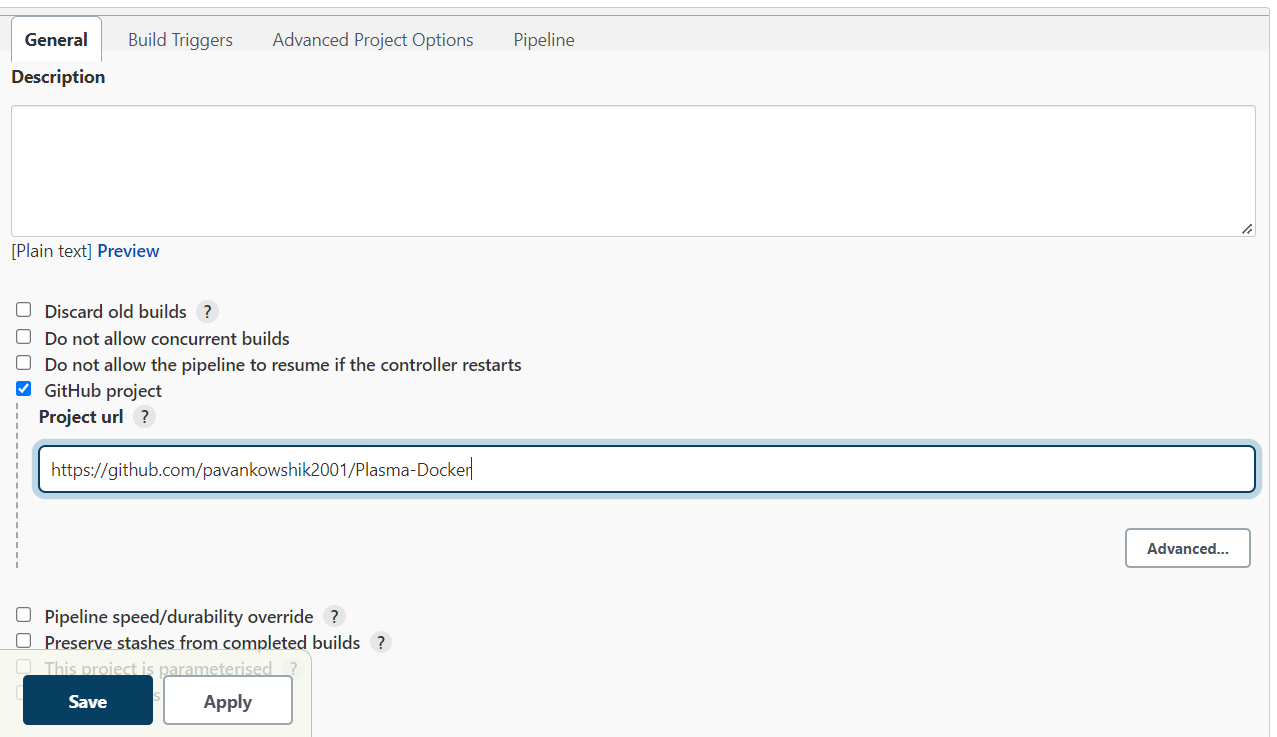


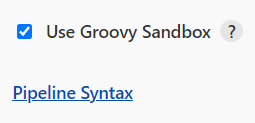
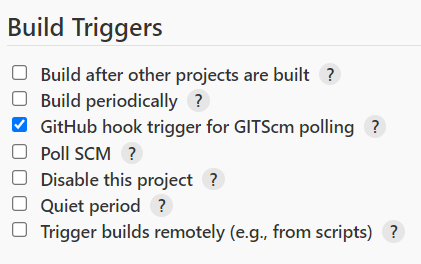


1. Now we are creating a new pipeline(Select Pipeline)

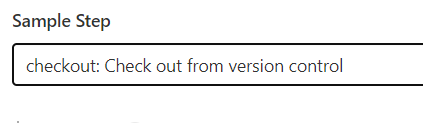


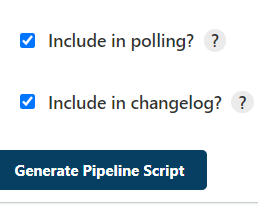
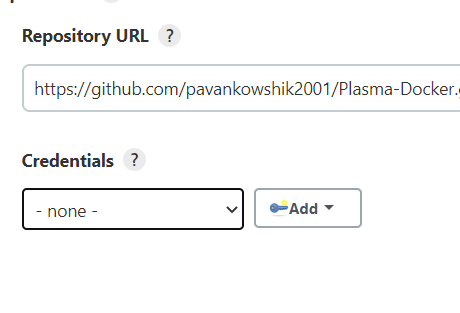
1. Now add the GITHUB Project



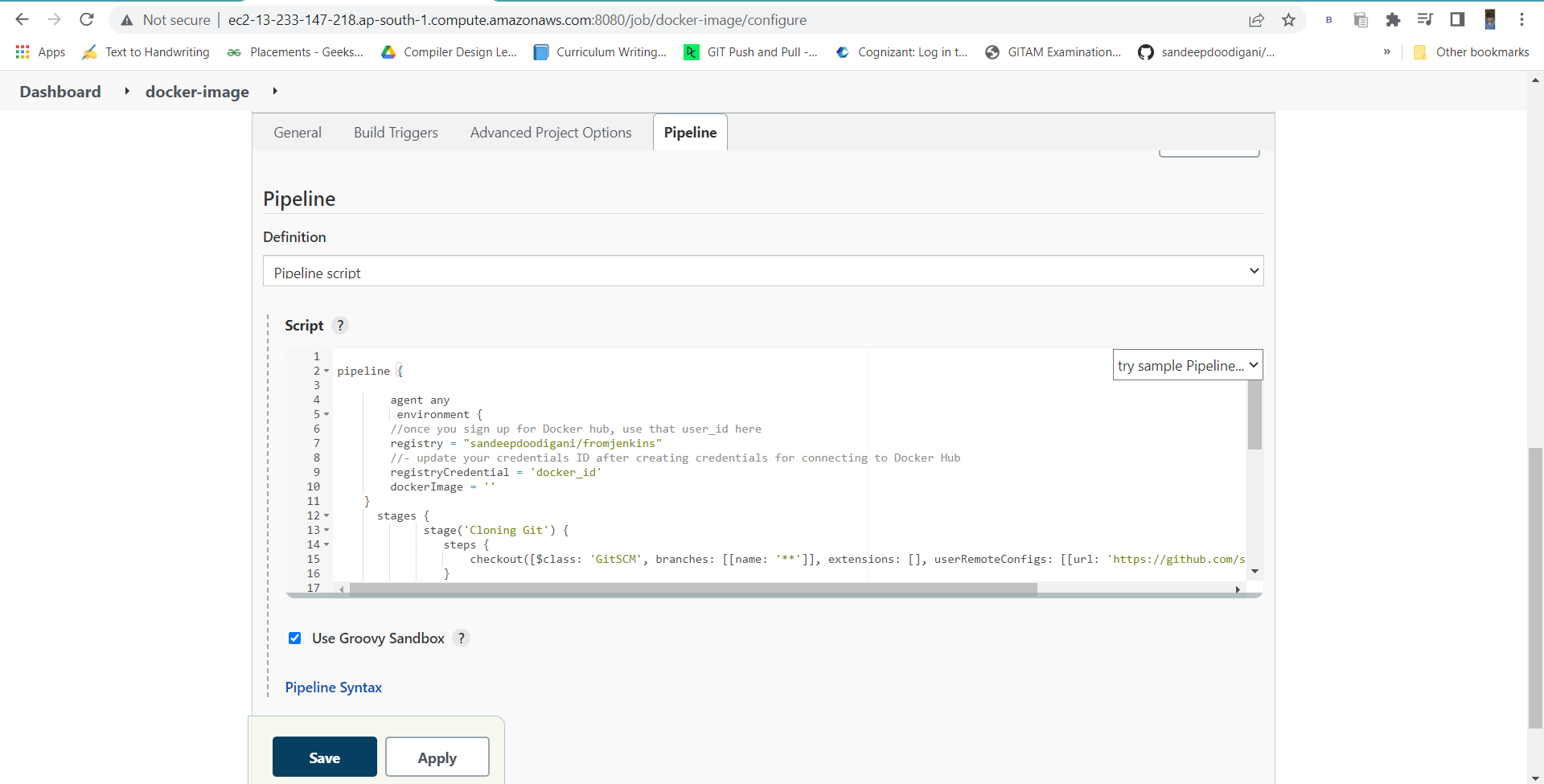


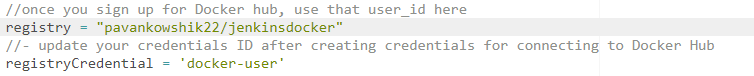
Click on Pipeline Syntax(Process of Generating Scripts)





Paste the code here





Registry link=Docker repository link

registryCredentials= Credentials name which we had created

1. Click on Apply



