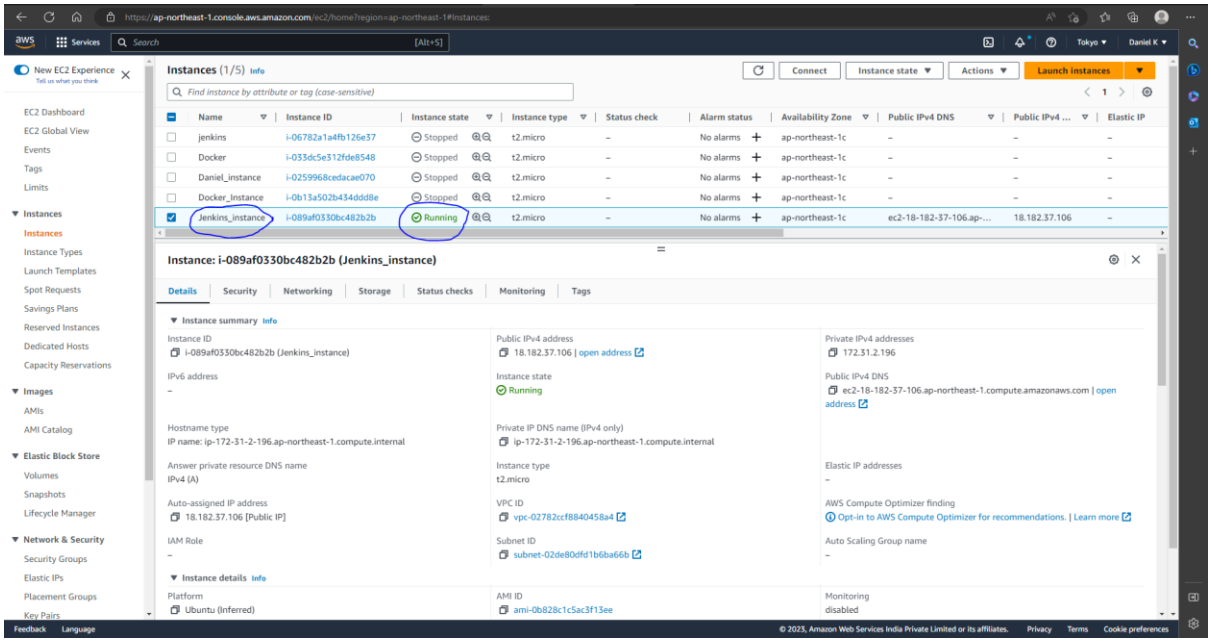
Devops Assignments - Module 4 - CICD using Jenkins

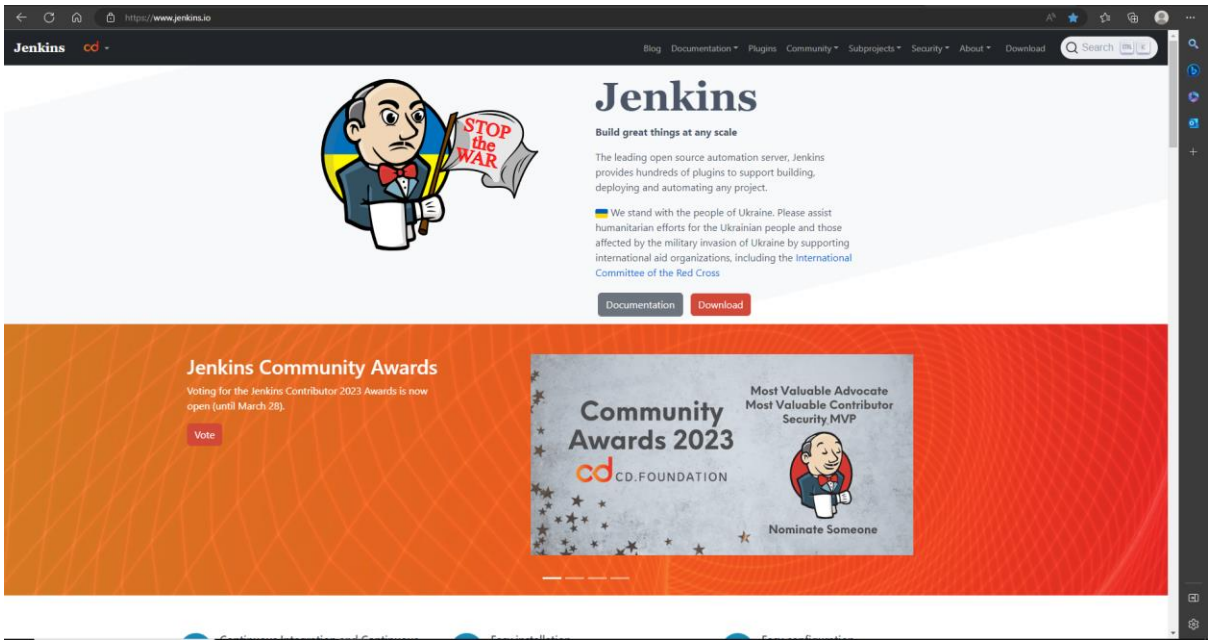
Assignment 1 -Installing Jenkins on EC2 host.docx.txt

Here is the official website of AWS Amazon web services: - https://aws.amazon.com/

1. Sign into to your Account
2. In search box search for EC2 and select EC2.
3. Click on Launch Instance
4. Provide the Required Information such as name, operating system, key pair, etc and click on Launch Instance
5. Amazon EC2 Instance is Created Successfully and Running.
6. To connect to Instance use Puttygen or SSH into cmd



Here is the official website Of Jenkins:- https://www.jenkins.io/



1. Jenkins is Developed using java, in order to use Jenkins, Java needs to be Installed.

   - Command to Install Java:
     → sudo apt install openjdk-11-jre

```
ubuntu@ip-172-31-2-196: ~
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-2-196:~$ java -version
openjdk version "11.0.18" 2023-01-17
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-2-196:~$
```

2. Installing Jenkins:

   - Commands to Add required dependencies for the Jenkins package

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

   - Command to Install Jenkins:
     → sudo apt-get install jenkins

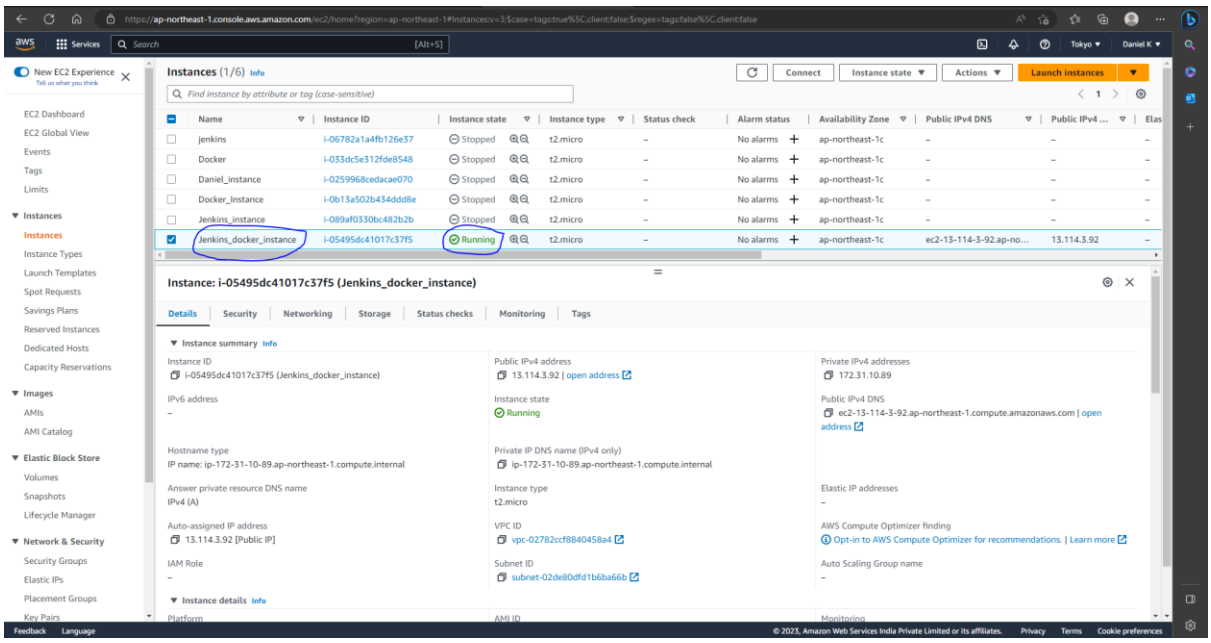
Jenkins has been Installed Successfully:

```
ubuntu@ip-172-31-2-196: ~
ubuntu@ip-172-31-2-196:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 49 not upgraded.
Need to get 96.3 MB of archives.
After this operation, 99.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.387.1 [96.1 MB]
Fetched 96.3 MB in 3s (28.2 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 65202 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.387.1_all.deb ...
Unpacking jenkins (2.387.1) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.387.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-2-196:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-03-14 11:51:22 UTC; 1min 9s ago
   Main PID: 4774 (java)
      Tasks: 36 (limit: 1143)
     Memory: 306.8M
        CPU: 42.311s
     CGroup: /system.slice/jenkins.service
             └─4774 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 14 11:50:50 ip-172-31-2-196 jenkins[4774]: d8d6efd608594e4ca28e3181fabc62f9
Mar 14 11:50:50 ip-172-31-2-196 jenkins[4774]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 14 11:50:50 ip-172-31-2-196 jenkins[4774]: ************************************************************
Mar 14 11:50:50 ip-172-31-2-196 jenkins[4774]: ************************************************************
Mar 14 11:50:50 ip-172-31-2-196 jenkins[4774]: ************************************************************
Mar 14 11:51:22 ip-172-31-2-196 jenkins[4774]: 2023-03-14 11:51:22.501+0000 [id=28]       INFO       jenkins.InitReactorRunner$1#onAttained: Completed initialization
Mar 14 11:51:22 ip-172-31-2-196 jenkins[4774]: 2023-03-14 11:51:22.539+0000 [id=22]       INFO       hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Mar 14 11:51:22 ip-172-31-2-196 systemd[1]: Started Jenkins Continuous Integration Server.
Mar 14 11:51:26 ip-172-31-2-196 jenkins[4774]: 2023-03-14 11:51:26.676+0000 [id=44]       INFO       h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Mar 14 11:51:26 ip-172-31-2-196 jenkins[4774]: 2023-03-14 11:51:26.677+0000 [id=44]       INFO       hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
ubuntu@ip-172-31-2-196:~$
```

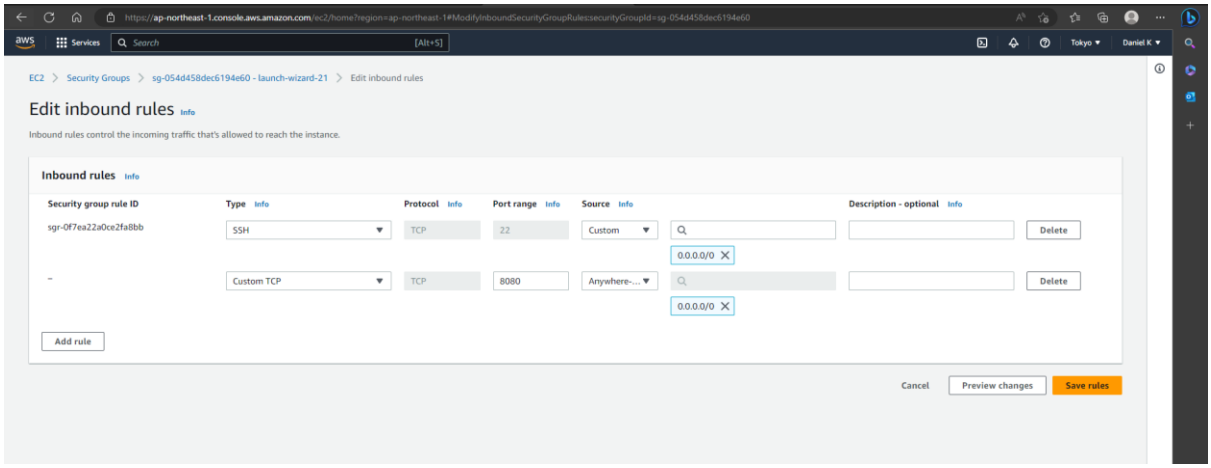
Assignment 2 - Installing & Configuring Jenkins using Containers.docx.txt

Here is the official website of AWS Amazon web services: - https://aws.amazon.com/

1. Sign into to your Account
2. In search box search for EC2 and select EC2.
3. Click on Launch Instance
4. Provide the Required Information such as name, operating system, key pair, etc and click on Launch Instance
5. Amazon EC2 Instance is Created Successfully and Running.
6. To connect to Instance use Puttygen or SSH into cmd

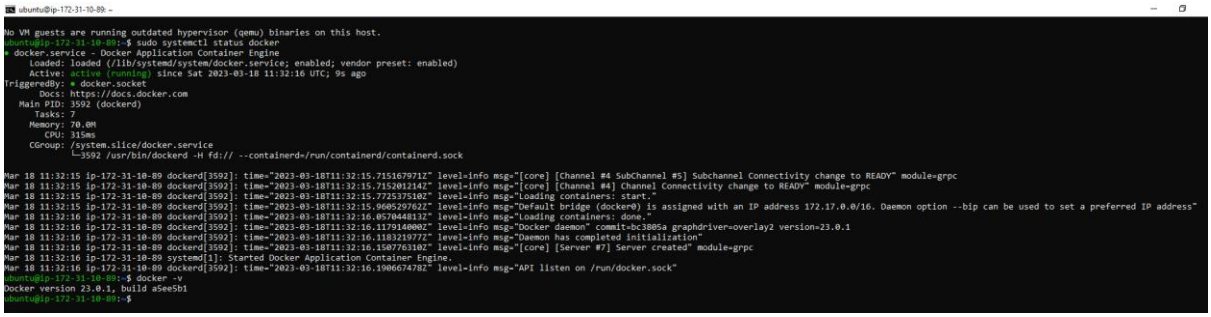


1. Jenkins Work on 8080 Port, so Enabling 8080 port on Security Inbound Rules:

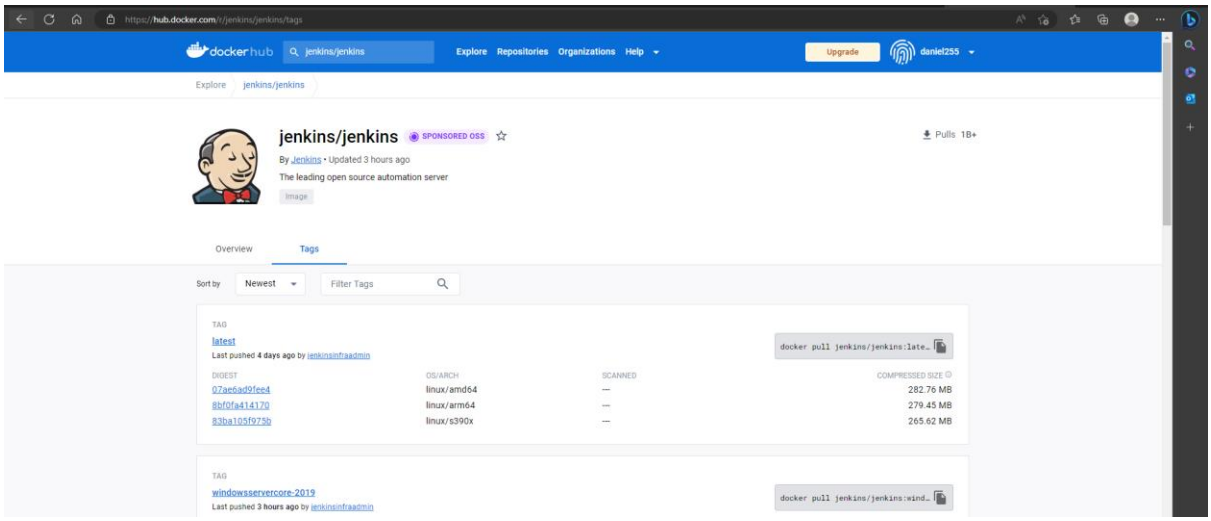


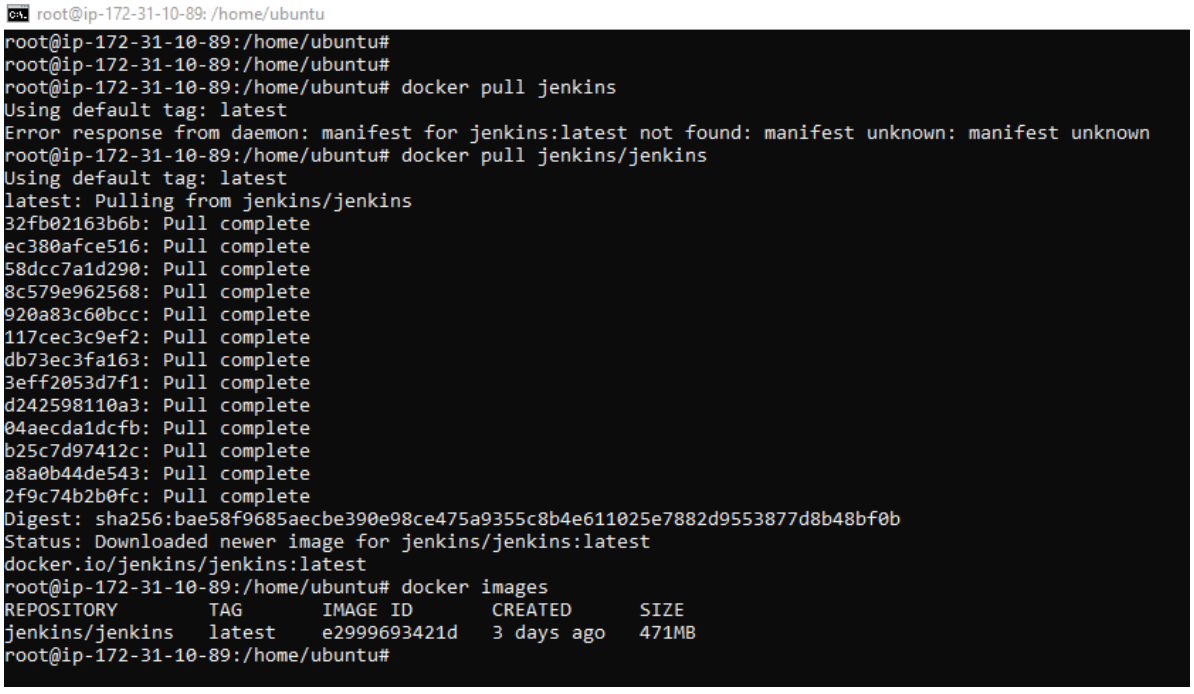
2. After launching, SSH to that EC2 instance. And Install Docker by running the following commands:
   - **sudo apt update** → To Update all the packages
   - **sudo apt install apt-transport-https ca-certificates curl software-properties-common** → install a few prerequisite packages

- **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"** → Adding the Docker repository
- **sudo apt install docker-ce** → To Install Docker
- **sudo systemctl status docker** → To Check Docker Status



3. Now we will pull the Jenkins image using docker from the docker hub:

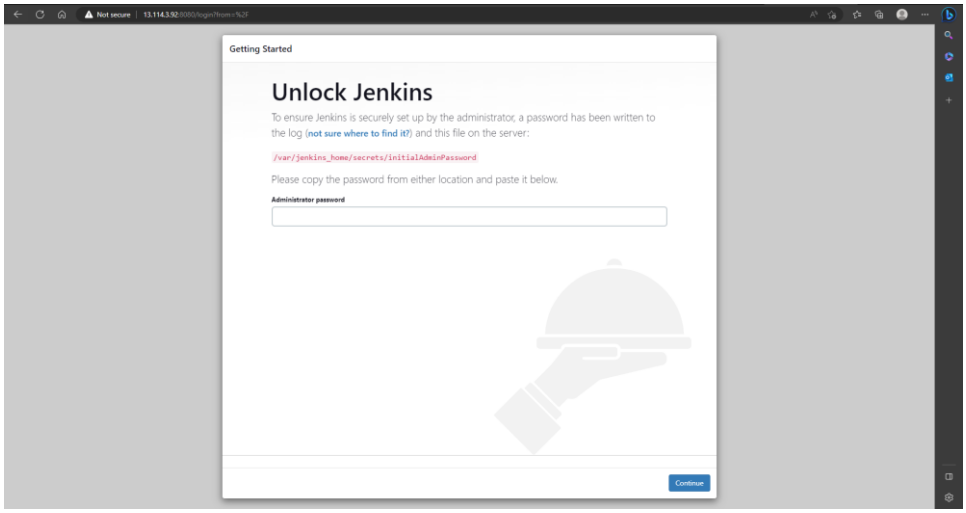




4. Since in docker images we can see our Jenkins image. Now we can make our Jenkins container run below command:
    - **mkdir Jenkins** → To make directory name Jenkins
    - **docker run -d --name jenkins -p 8080:8080 -v $PWD/jenkins/ jenkins/Jenkins** → To run a container name Jenkins using Jenkins image
    - **docker ps** → # To see if container is running or not

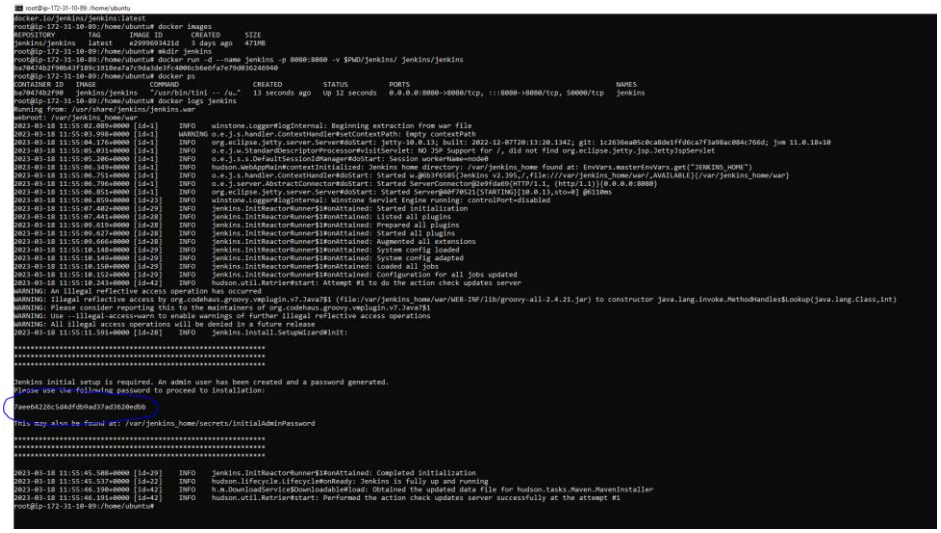
```
root@ip-172-31-10-89: /home/ubuntu
b25c7d97412c: Pull complete
a8a0b44de543: Pull complete
2f9c74b2b0fc: Pull complete
Digest: sha256:bae58f9685aecbe390e98ce475a9355c8b4e611025e7882d9553877d8b48bf0b
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
root@ip-172-31-10-89:/home/ubuntu# docker images
REPOSITORY        TAG       IMAGE ID       CREATED      SIZE
jenkins/jenkins   latest    e2999693421d   3 days ago   471MB
root@ip-172-31-10-89:/home/ubuntu# mkdir jenkins
root@ip-172-31-10-89:/home/ubuntu# docker run -d --name jenkins -p 8080:8080 -v $PWD/jenkins/ jenkins/jenkins
ba70474b2f90b43f189c1918ea7a7c9da3de3fc4006cb6e6fa7e79d036246940
root@ip-172-31-10-89:/home/ubuntu# docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS          PORTS                                                                                  NAMES
ba70474b2f90   jenkins/jenkins   "/usr/bin/tini -- /u…"   13 seconds ago   Up 12 seconds   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   jenkins
root@ip-172-31-10-89:/home/ubuntu#
```

5. Now, copy the Public IPv4 address of the EC2 instance After copying it. Paste it into a new tab with port 8080:

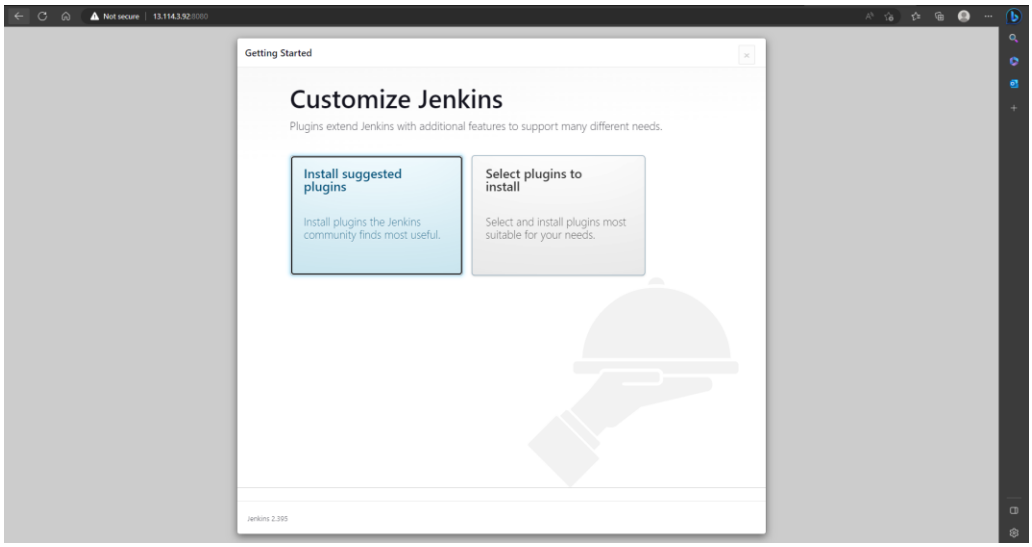


6. Here for password enter command:

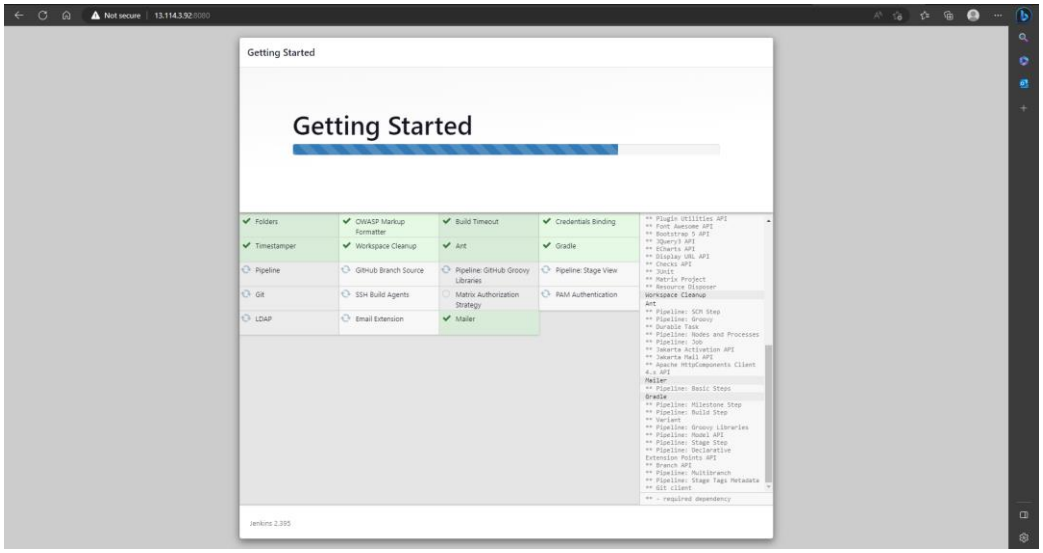
- **Docker logs Jenkins** → To see logs of the container name Jenkins



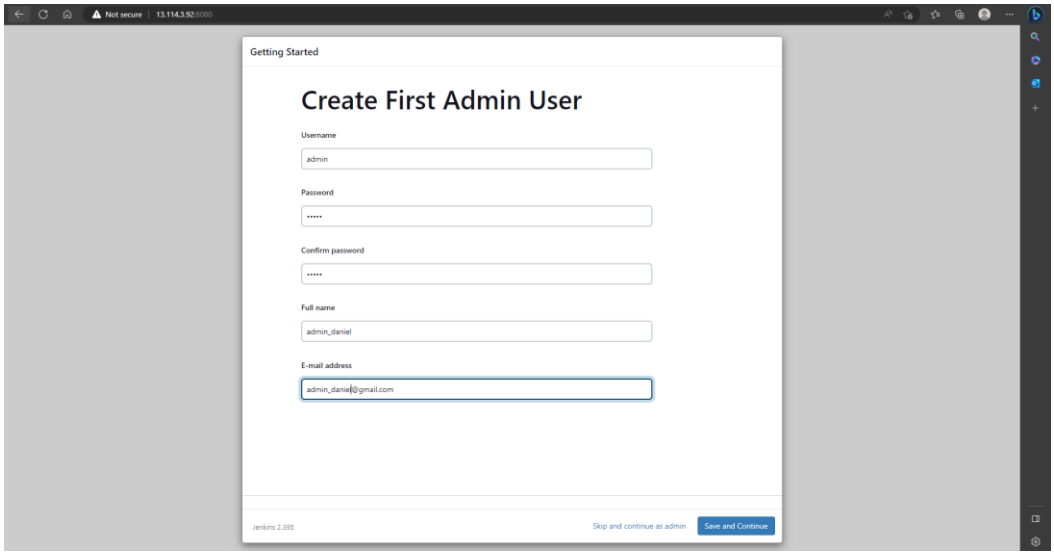
7. Here Jenkins can be customized through selecting or installing suggested plugins:



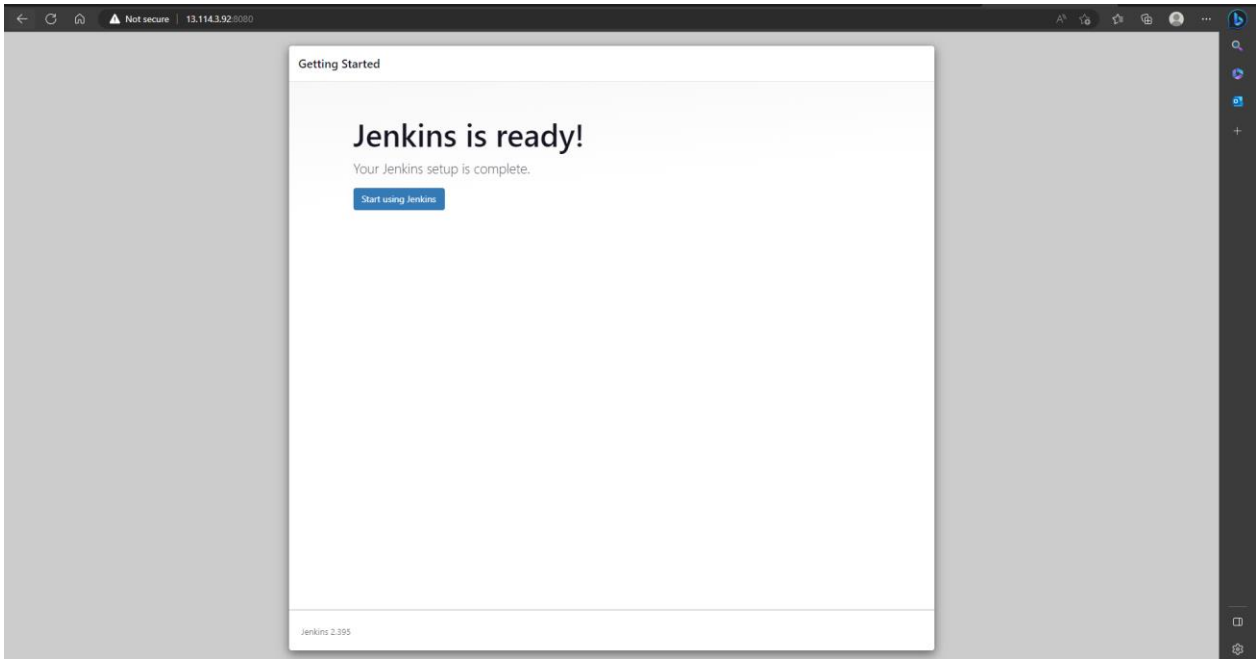
8. Installing Plugins:



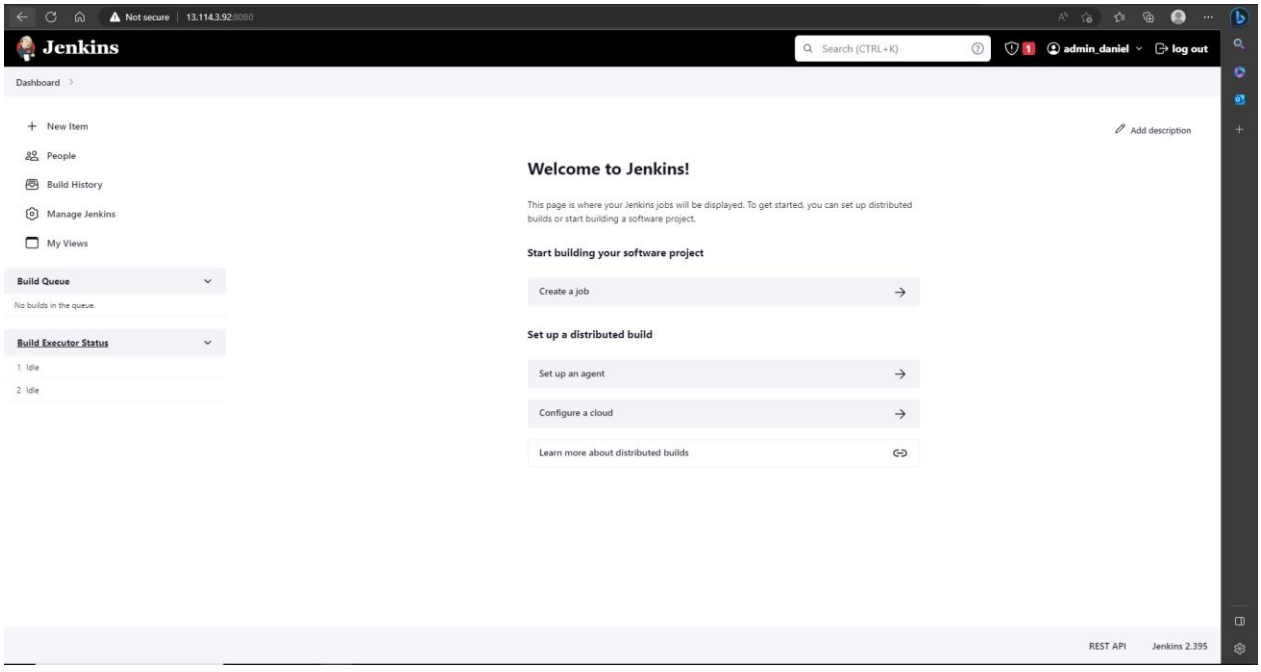
9. Here create your administration user id and password:



10. Jenkins is now ready to use:



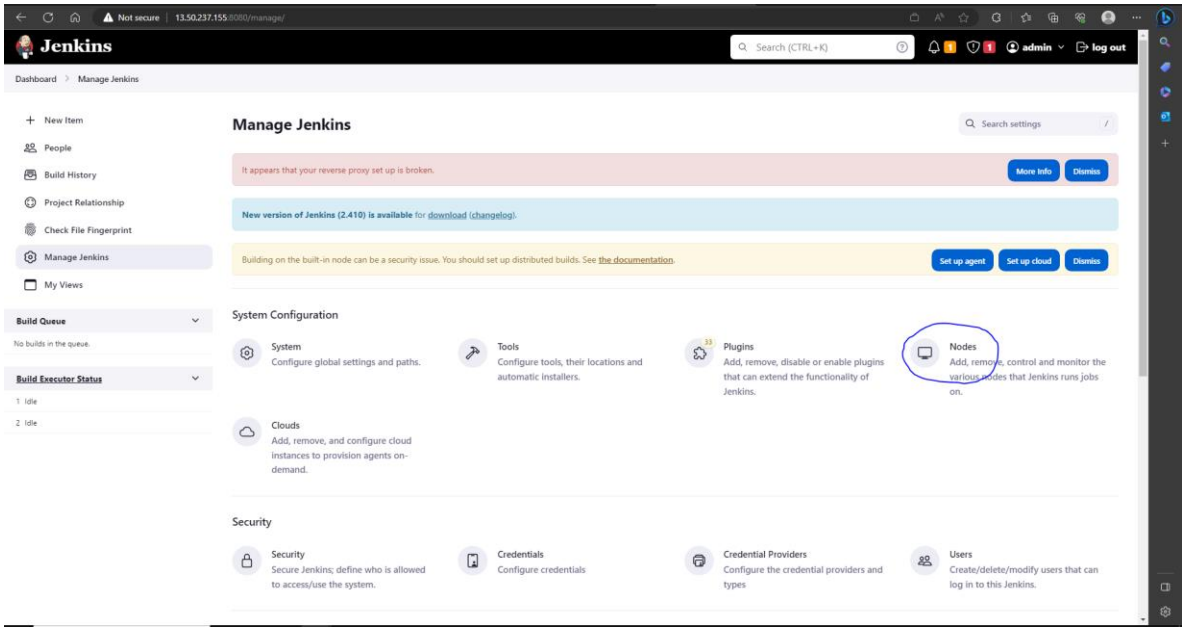
11. Jenkins has been installed and configured using container successfully:

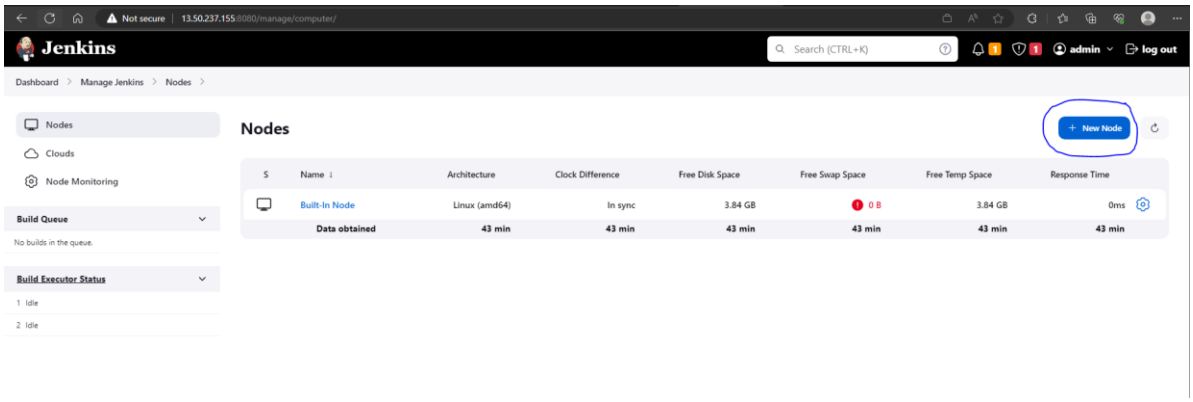


Assignment 3- MASTER SLAVE SETUP (Using Containers).docx.txt

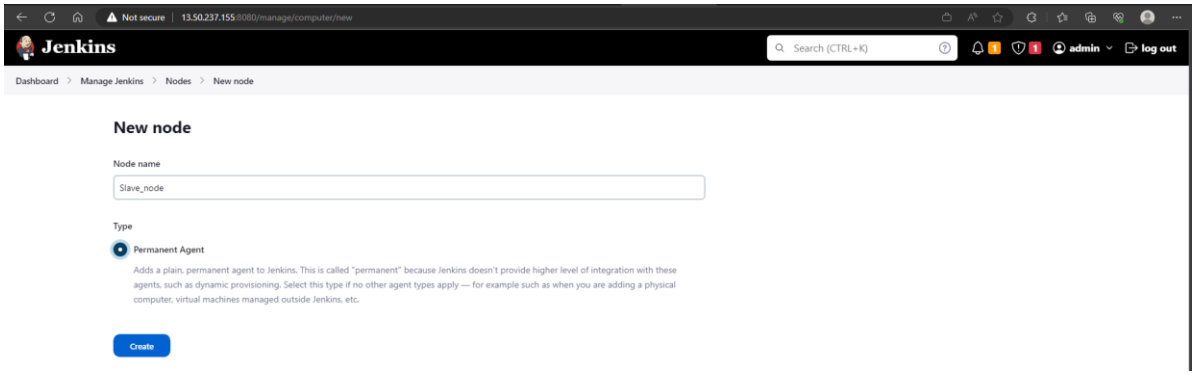
Configure to add a new node in Jenkins Master and copy the agent URL to be run on the Jenkins Slave using containers:

1. Open the Jenkins web console and navigate to 'Manage Jenkins' -> 'Manage Nodes & Clouds' -> 'New Node' to add a new node to the Jenkins Master

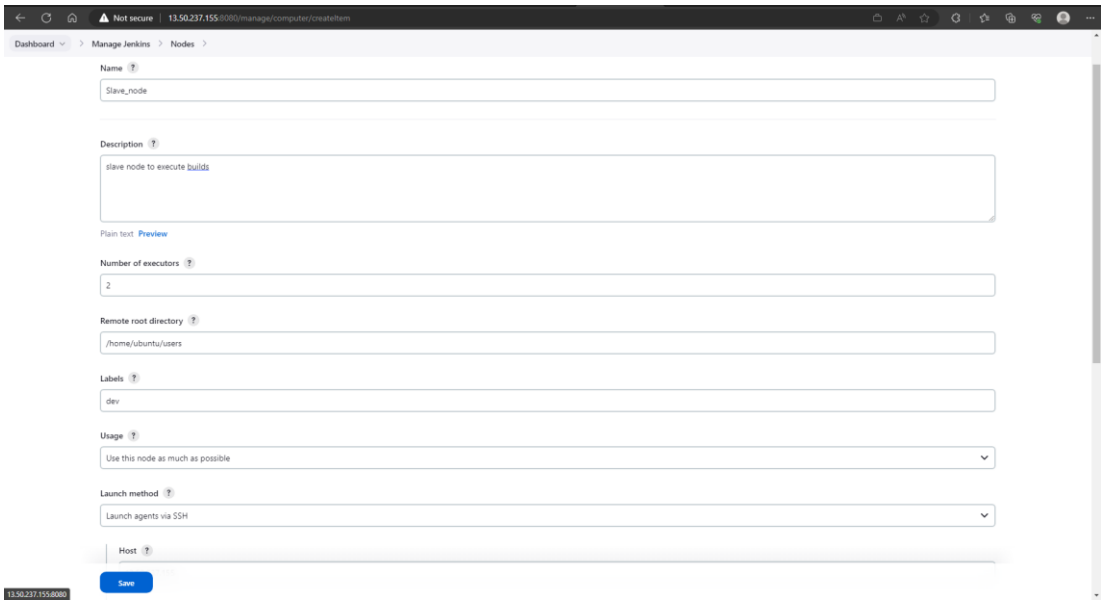




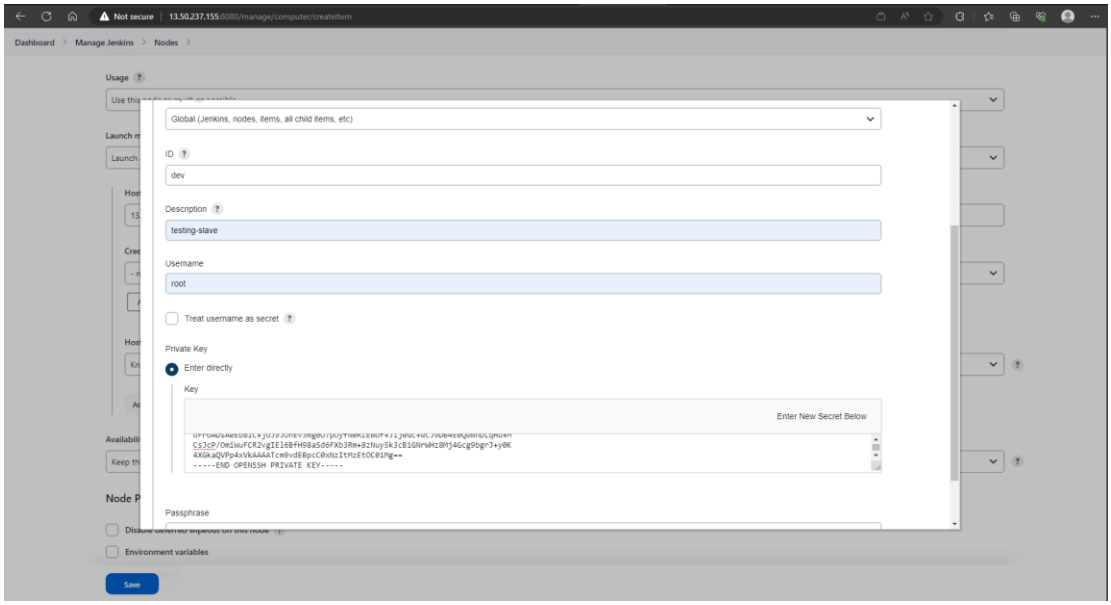
2. Give the node a name, select 'Permanent Agent' and click 'OK'



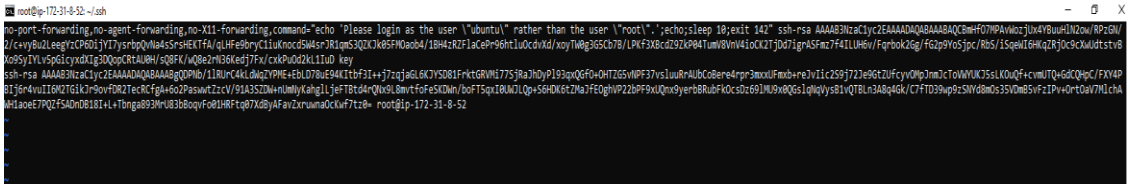
3. Fill in the details for the new node, including the remote file system root, number of executors and other configuration options



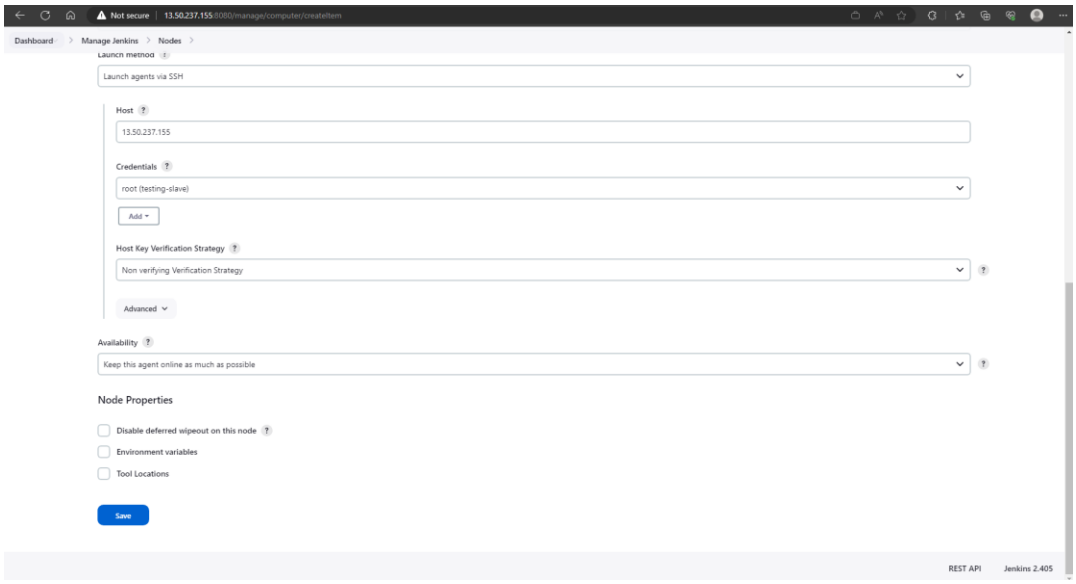
4. Scroll down to the 'Launch Method' section and select 'Launch agent via SSH’



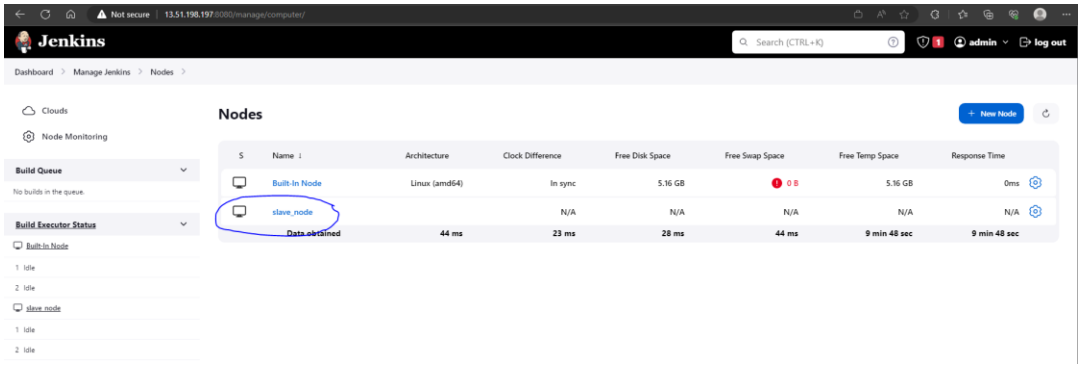
➢ Pasting public key in authorized_keys



- Click on Save



- Slave Node has been Created successfully and it is in online:

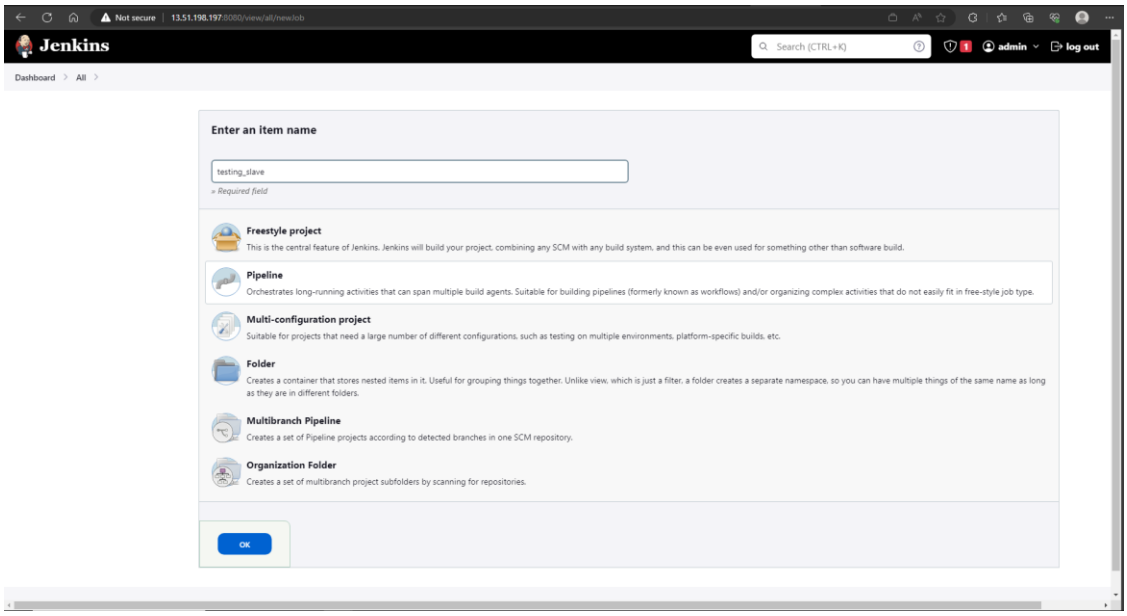


5. Create the below sample Pipeline job on Jenkins Master container and check if the job is getting created on Jenkins Slave container:

- Here is sample pipeline

```
pipeline {
  agent {label 'slave_node'}

   stages {
      stage('Hello') {
         steps {
            sh 'touch hello.txt'
         }
      }
   }
}
```

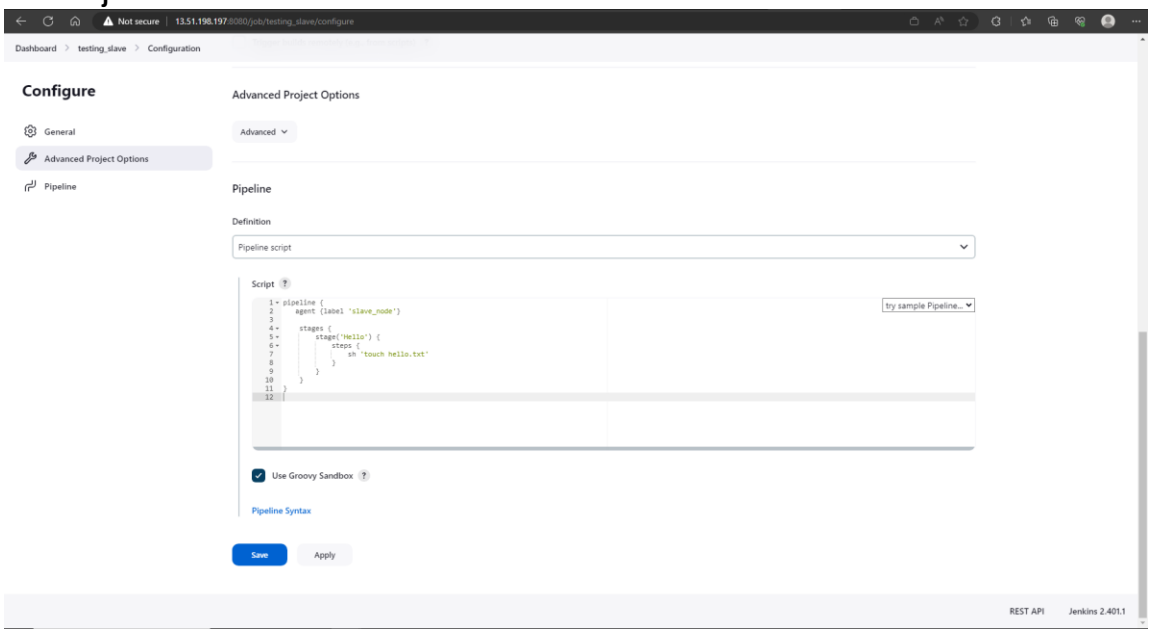
➢ Creating a pipeline project



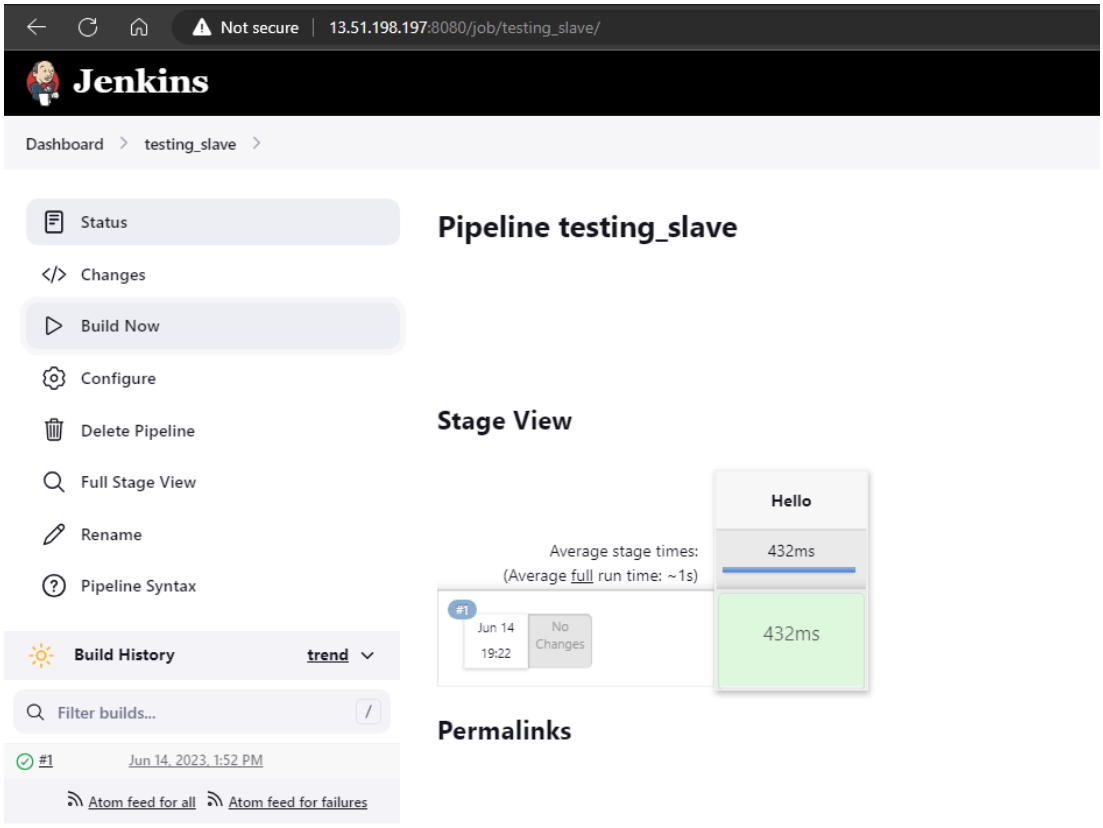
➢ Pasting the Given Pipeline script:

- Here my agent slave name is slave_node

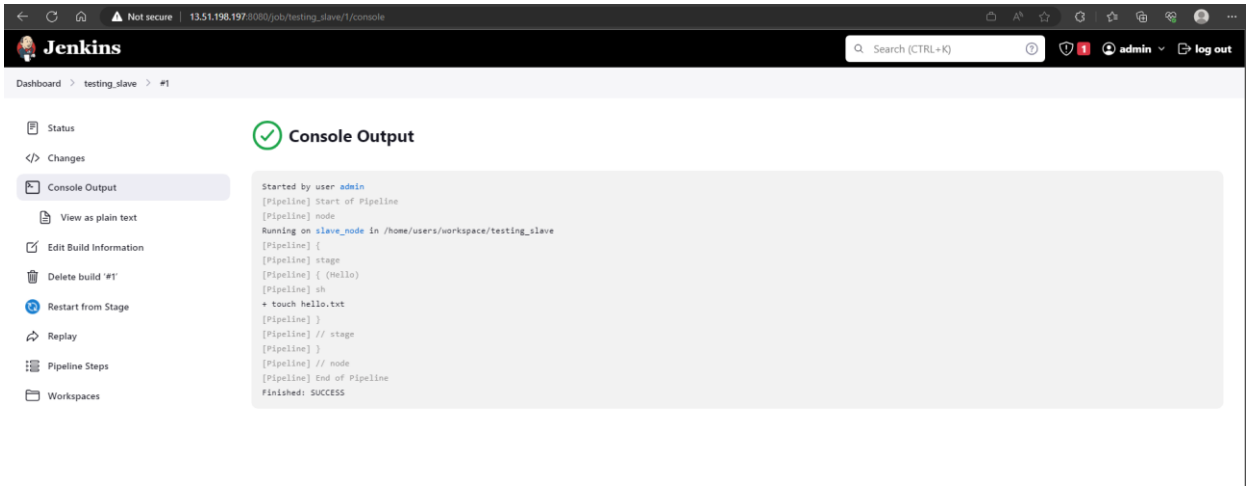
```
pipeline {
  agent {label 'slave_node'}

   stages {
      stage('Hello') {
         steps {
            sh 'touch hello.txt'
         }
      }
   }
}
```



- Build Now:



- Build has been executed successfully:

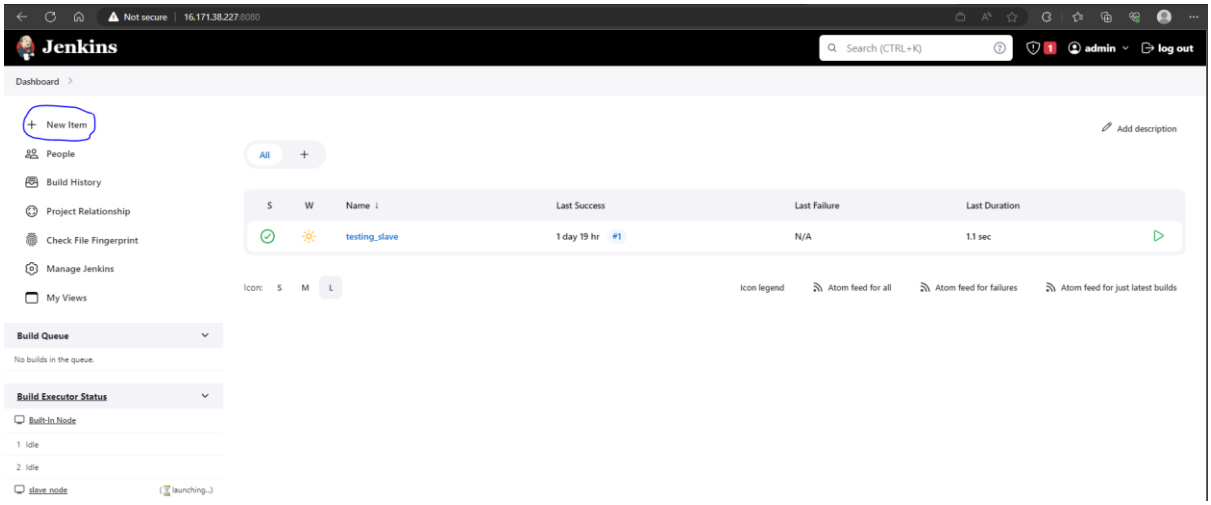


Assignment 4 - Working with Parameterized Jobs & Triggers.docx.txt

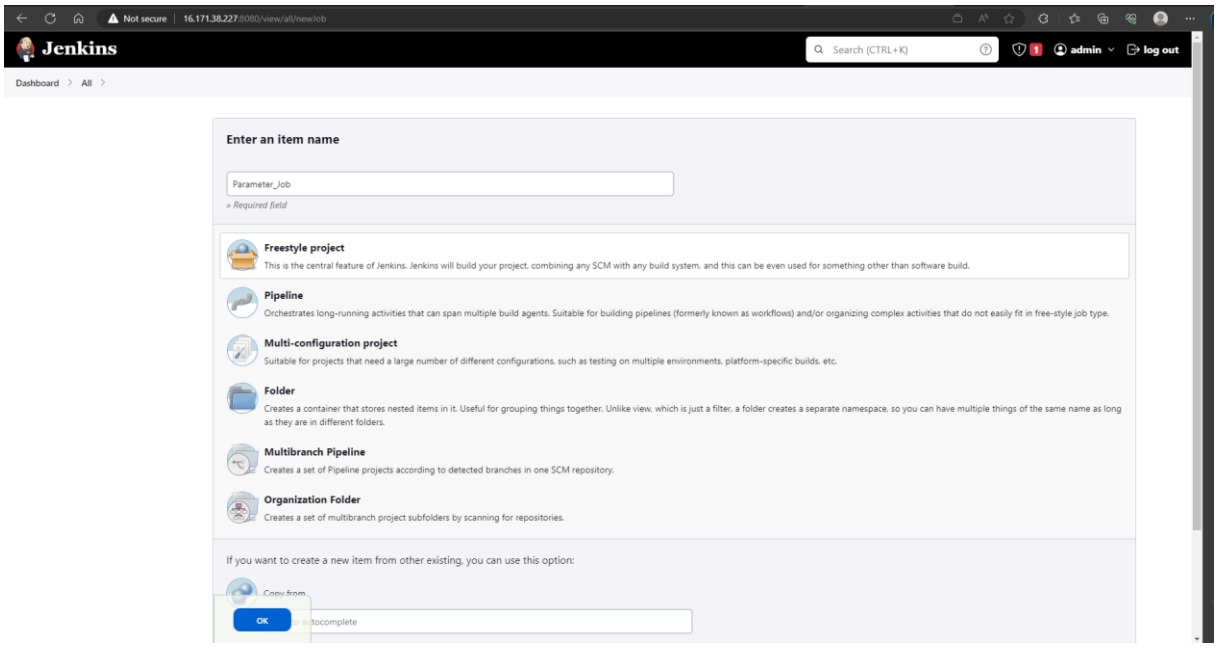
Setting up parameterized jobs and triggers in Jenkins Master

1. Create a new free style project in which you choose the option “The project is Parameterized”

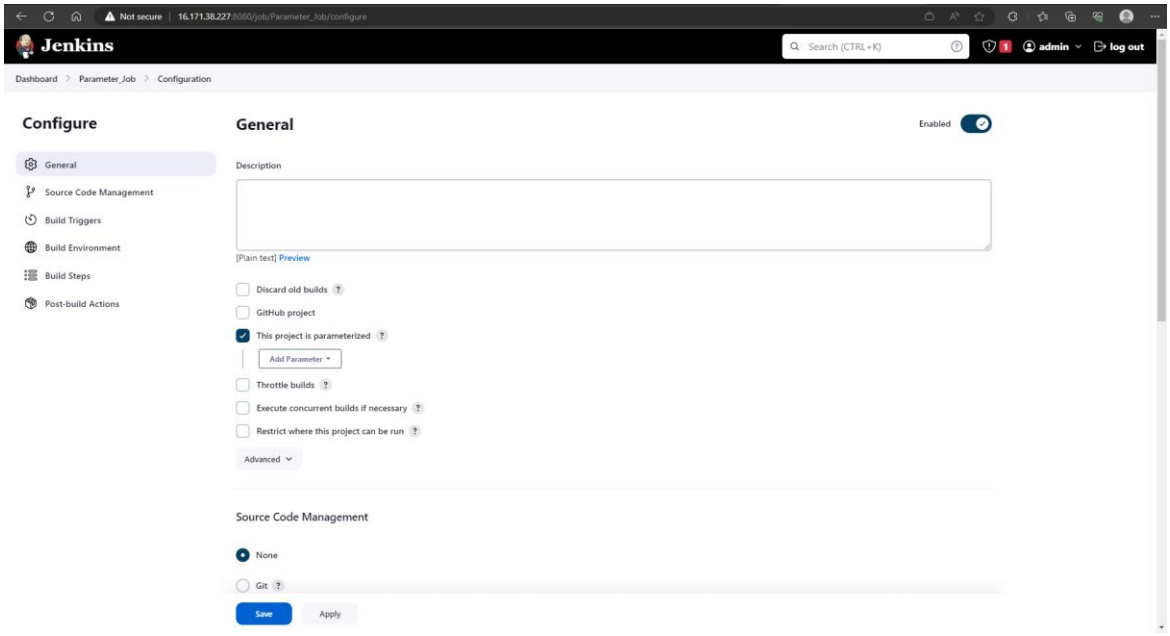
➢ Open the Jenkins web console and navigate to 'New Item' to create a new job



➢ Give the job a name and select the 'Freestyle project' type.

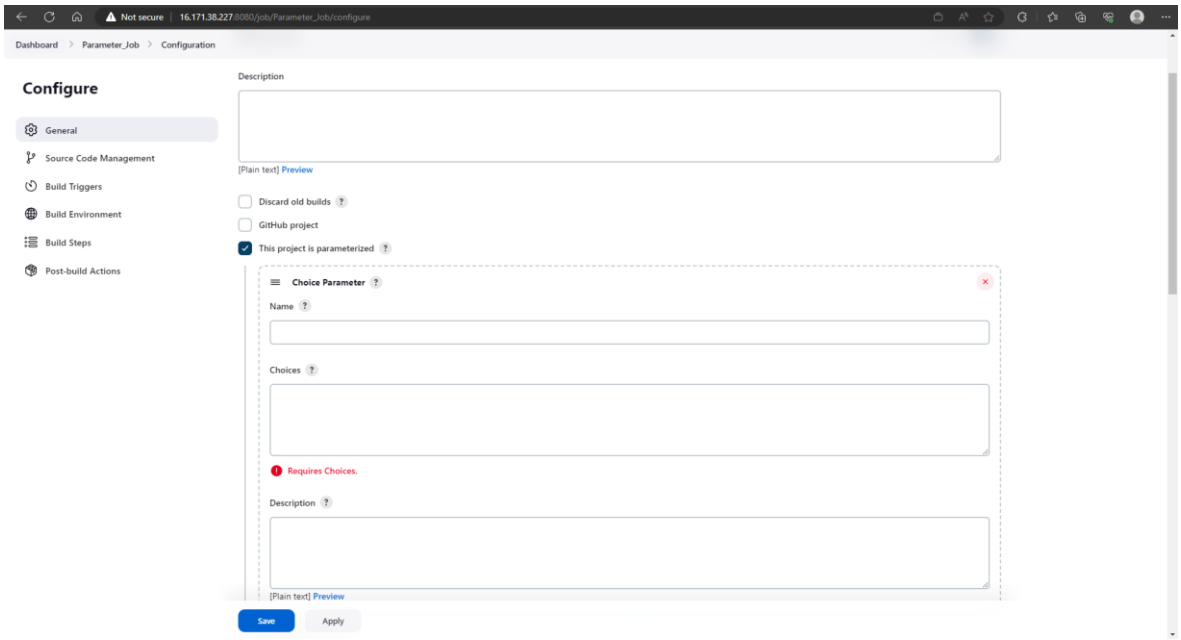


➢ In the 'General' section, enable the 'This project is parameterized' checkbox

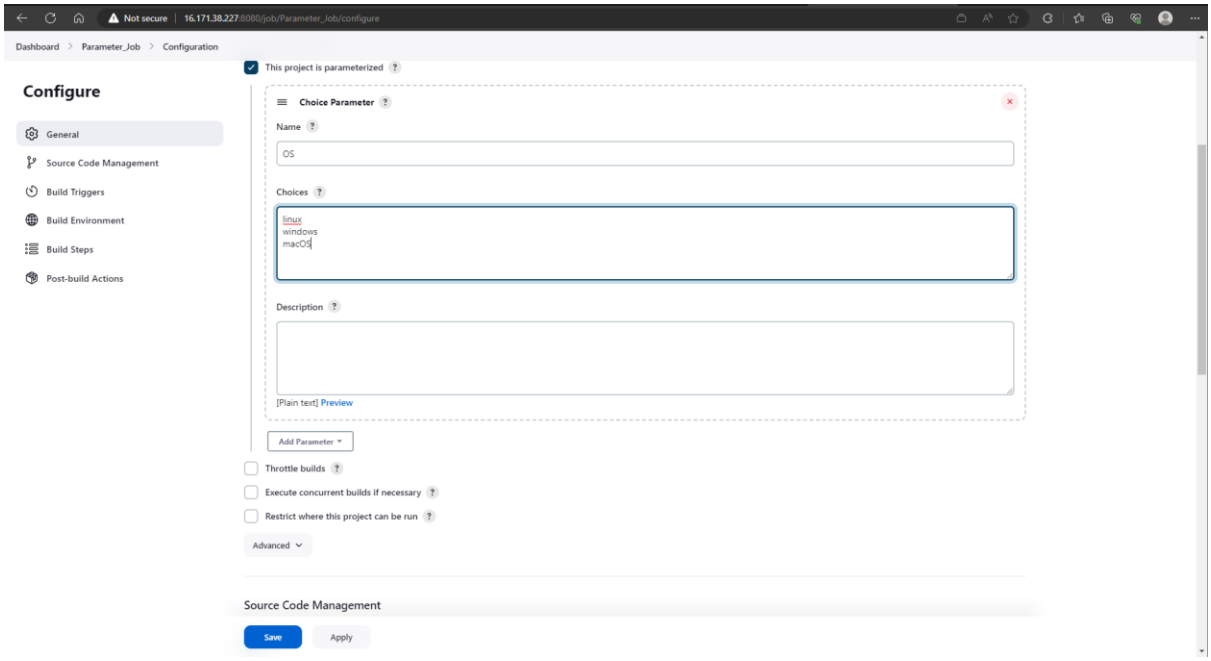


2. Choose “Choice Parameters” and update the options of OS -Linux, windows, MacOs under choices

- Click the 'Add Parameter' button and select 'Choice Parameter'

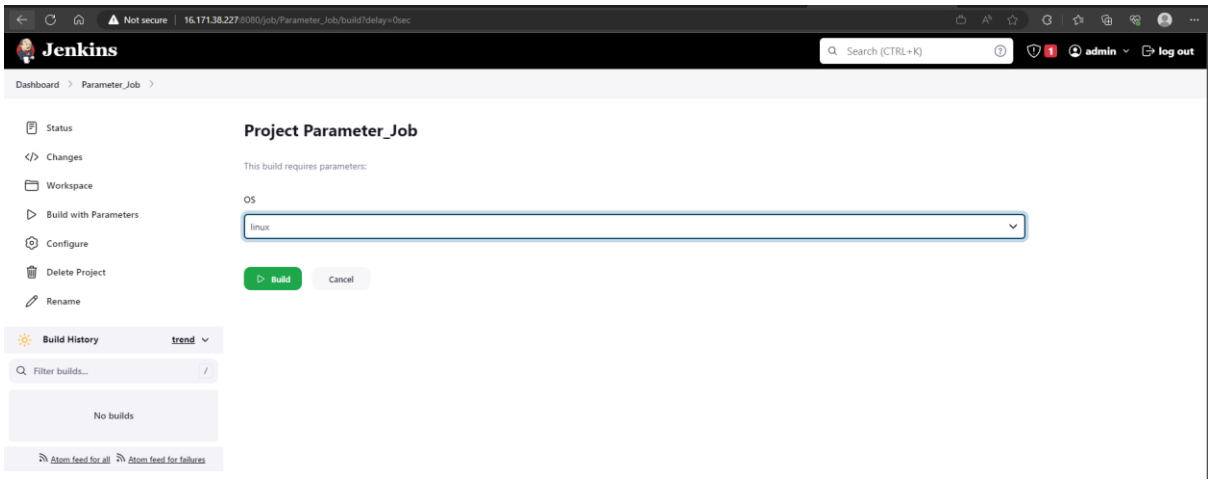


- Enter 'OS' as the parameter name
- Under 'Choices', enter the three OS options as follows:
  Linux
  Windows
  MacOS
- Click 'Save' to save the job configuration

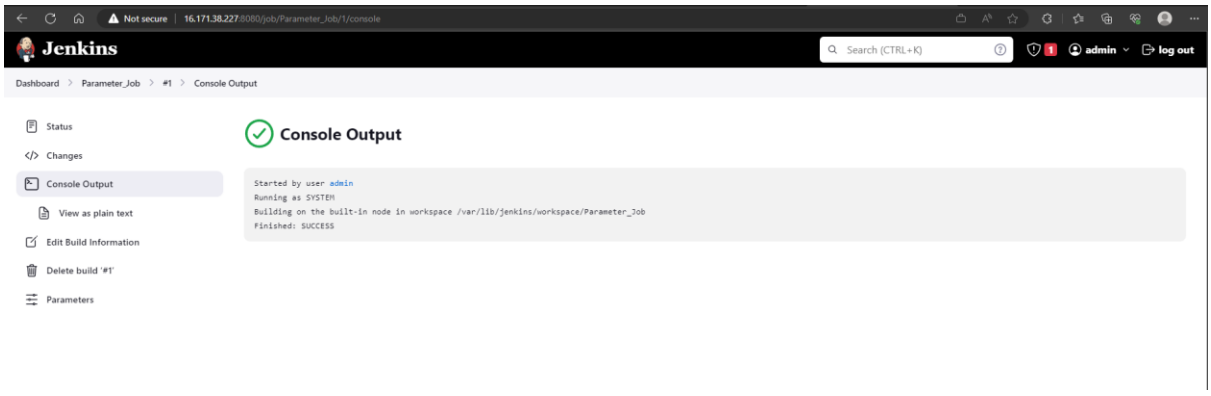


3. Build a job using these parameters

- Click the 'Build with Parameters' button to start the job
- Select the 'OS' parameter that you created earlier and choose a value for the parameter.

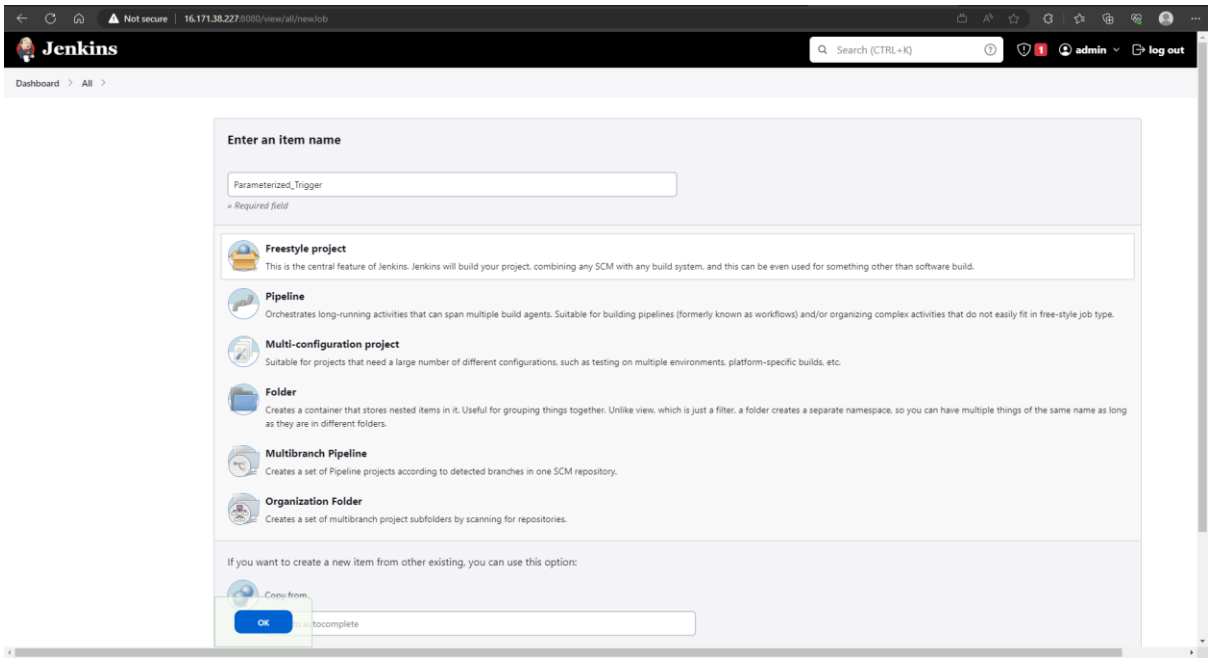


- Click the 'Build' button to start the job.

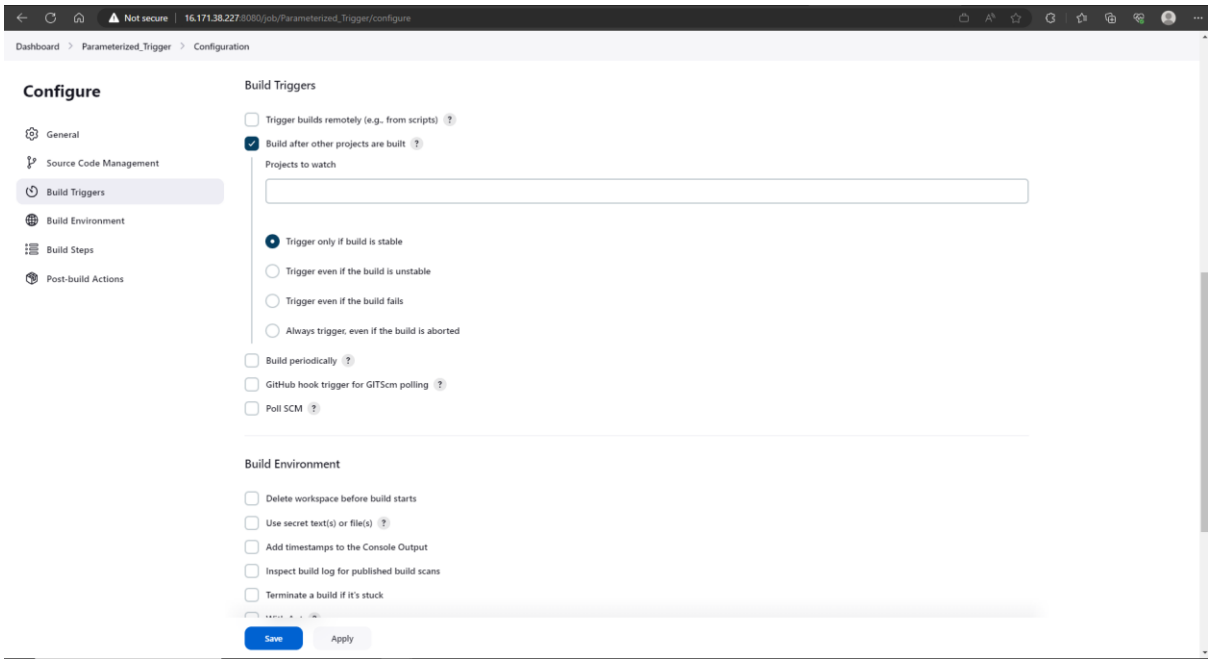


4. Create a new job called as Parameterized trigger which will be triggered once Parameterized build is completed

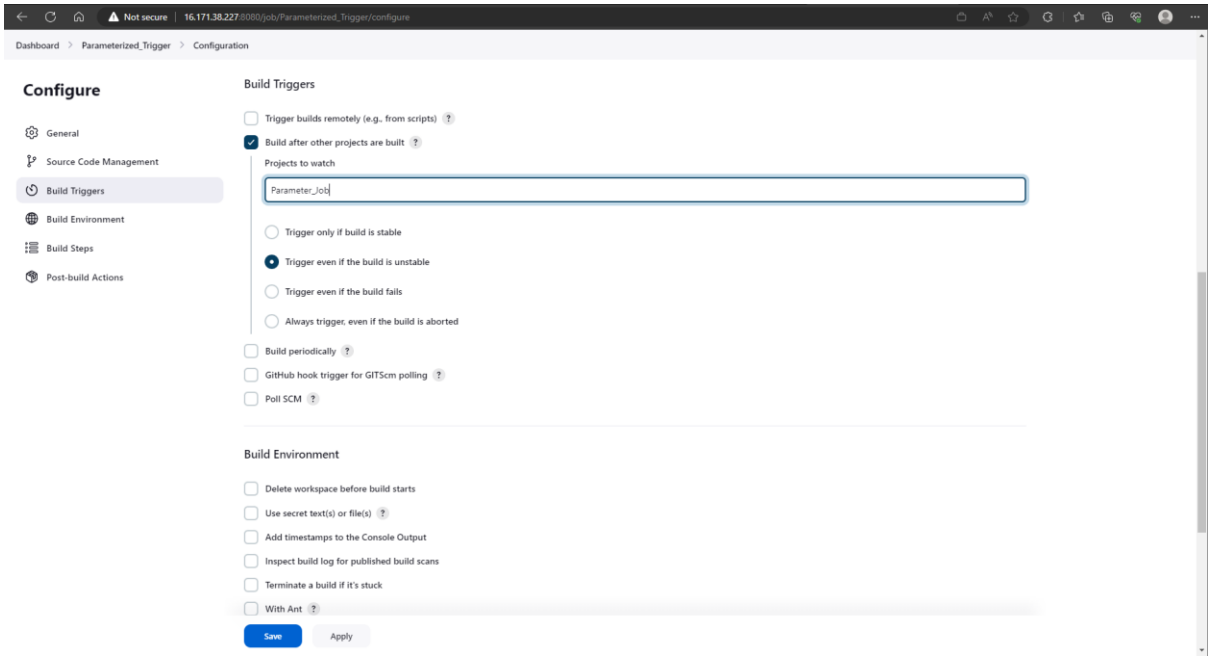
- create a new job called 'Parameterized Trigger'.



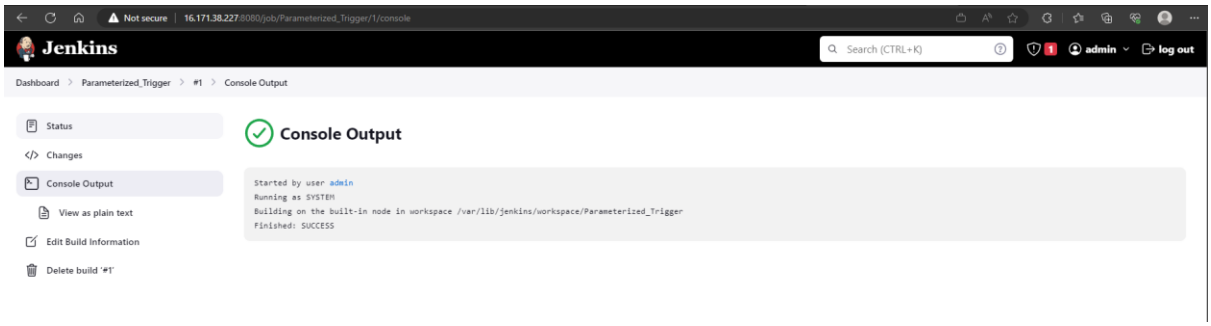
- In the 'Build Triggers' section of the job configuration, enable the 'Build after other projects are built' checkbox.



- In the 'Projects to watch' field, enter the name of the parameterized job that you created earlier
- In the 'Trigger only if build is stable' field, select the 'Trigger even if the build is unstable' checkbox



➢ Save the job configuration.



➢ Now when the parameterized job completes running, the 'Parameterized Trigger' job will automatically be triggered.