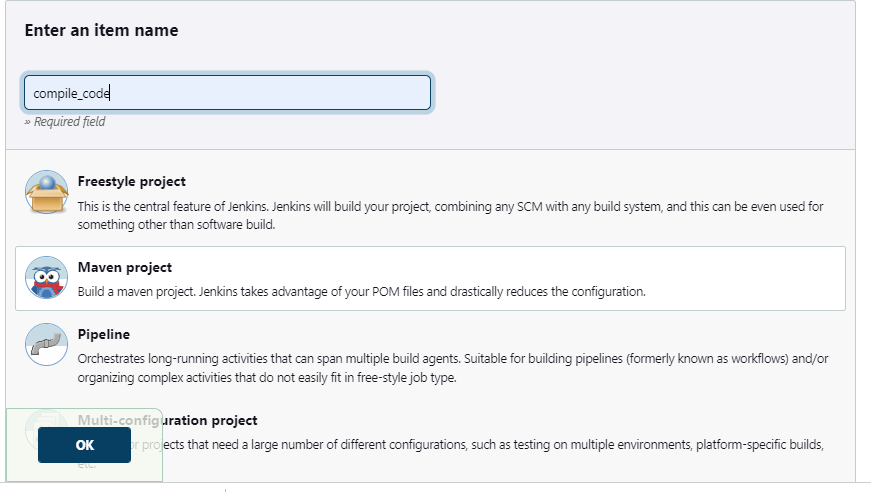
**Task2**

**Set up the Git repository and push the source code. Then, log in to Jenkins.**

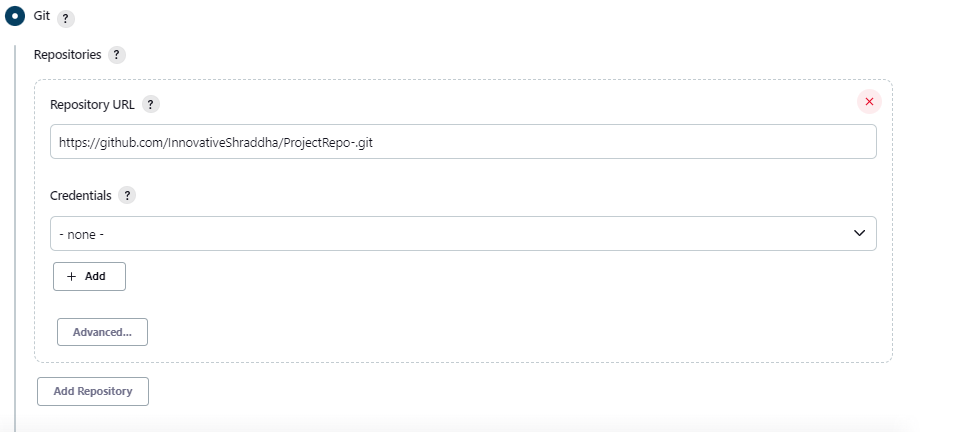
Click on new item it will open of new window as shown in screenshot

**Compile code :**

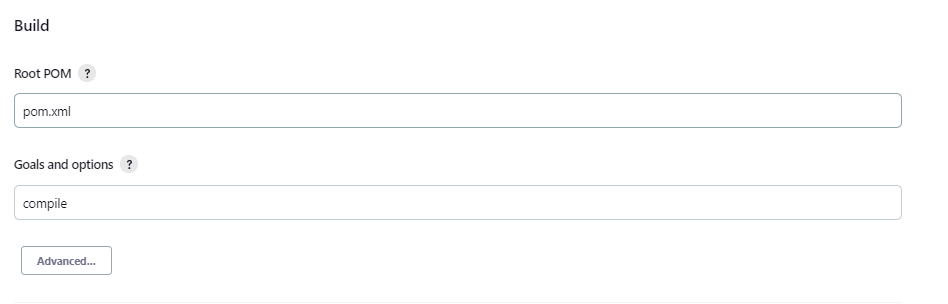
Give the name of job as compile code select Maven project and click on Ok button.



New pages created of compile code job in that go to **source code management option** select **Git** and give URL of of our GIT repository where our source code of project is located



Go to build section and in the goals and option give Maven command as compile to compile our source code .



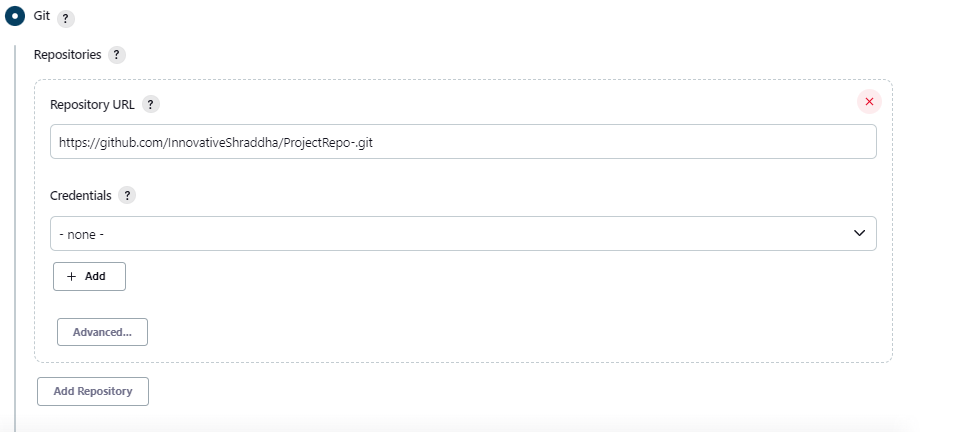
Save the job and build it will show the following console output after build get successful.



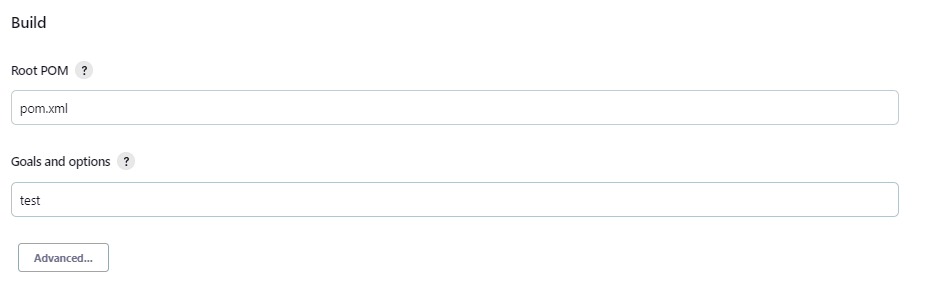
In a similar way for creating test job go to to new item it will open up the the same page provide the name of job as test code select Maven project and click on Ok button

**2)Testing the code**

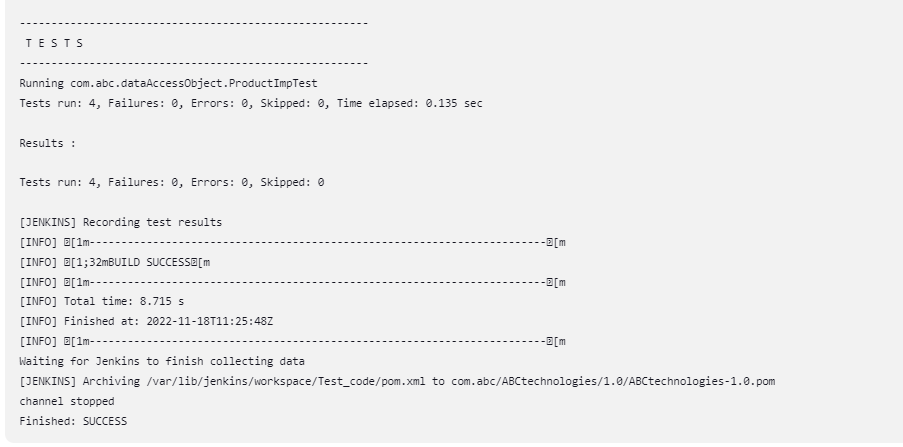
Again provide the URL of GIT Repo where our source code is located as shown in the screenshot



Then in the build section provide the Maven command to generate test results



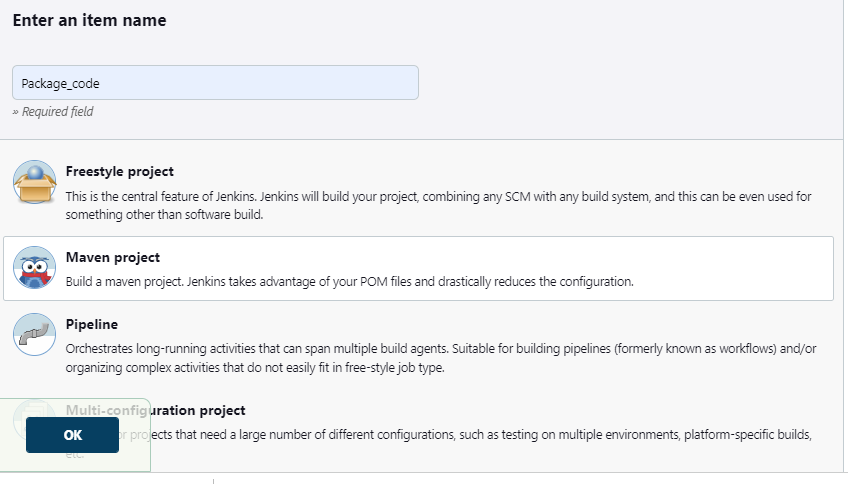
Click on save the job and built it it will give following results after build get successful as shown in the screenshot



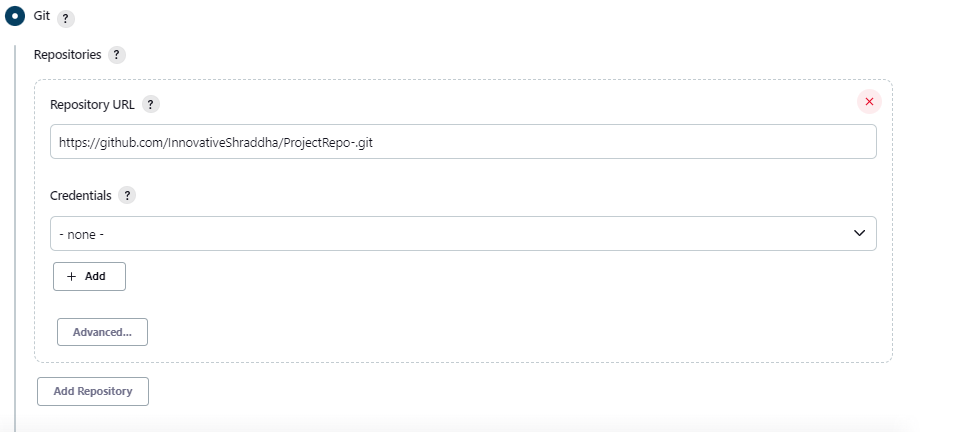
**3)Package job**

For package job also we are following the same processor as followed before 2 jobs

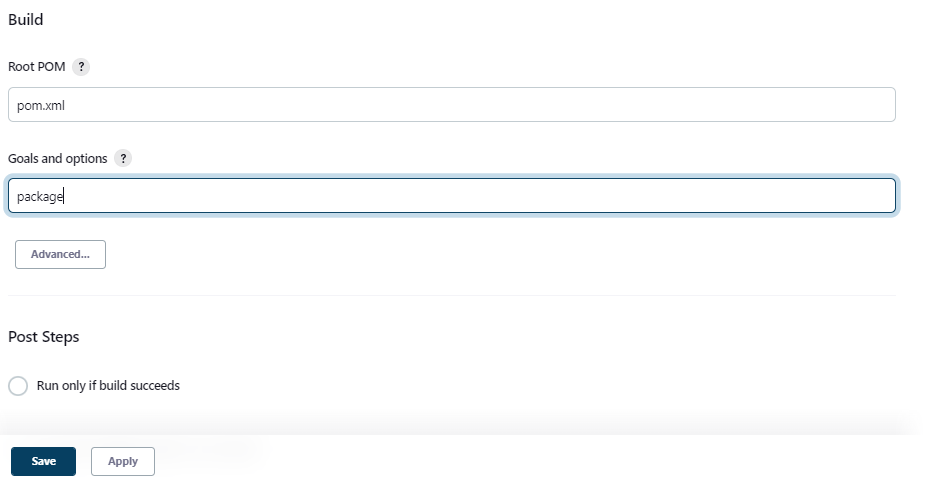
Again select new item from Jenkins dashboard it will open up new page enter the name of job as package code select Maven project and click on Ok button as shown in screenshot



Then provide a GIT Repo ads where our source code is located copied from GitHub paste it



After that in the build section provider Maven command for packaging our code into war file



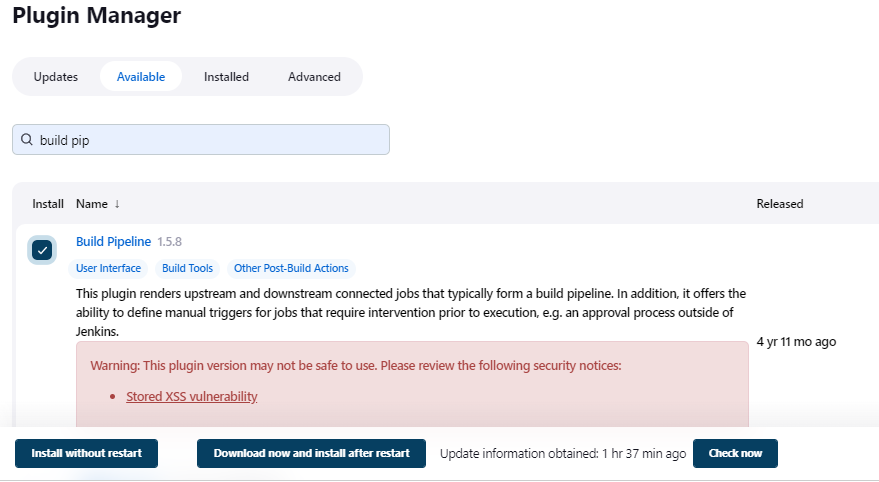
Click on save and build the job it will show following output as mention in screenshot if built get success



**Creating CI/CD pipeline**

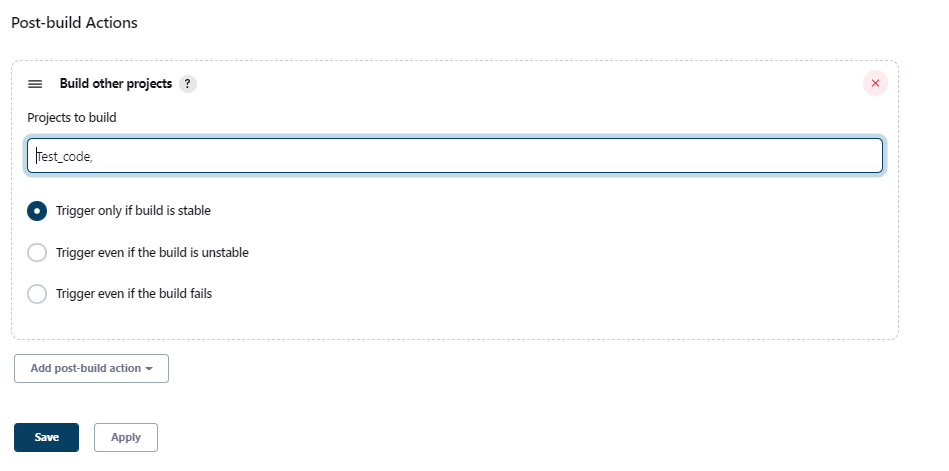
For creating a CI CD pipeline of job that are created earlier that is compile test and package first we need to install plugin of build pipeline to show show pipeline of jobs created before

Goto **Manage Jenkins** go to **Manage Plugins** in the plugin manager go to the available section and search for **Publish Over SSH** plugin



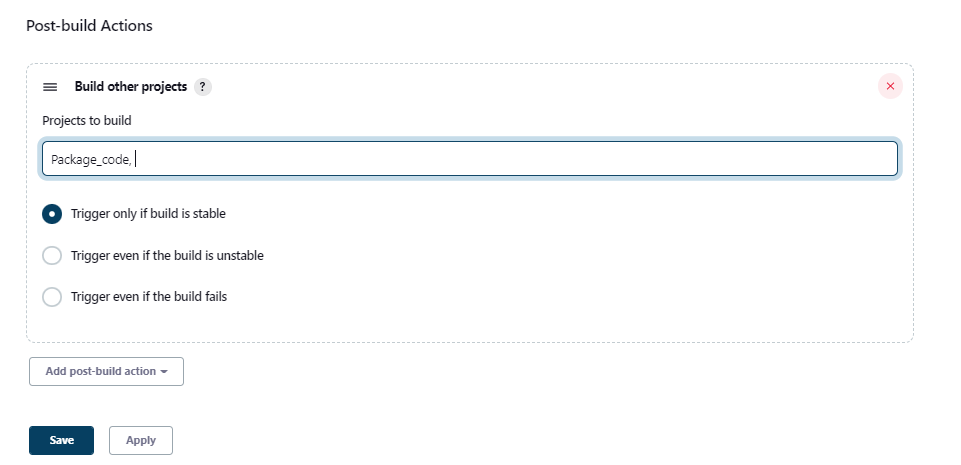
Click on **install without start** button it will install plugin without rebooting our ec2 instance for Jenkins server.

After installing the plugin successfully go to your **Compile\_code** job in **the post build action** Select >>**build other project** in that give the job that we are want to build it the **Compile\_code** job that is test click on save button.



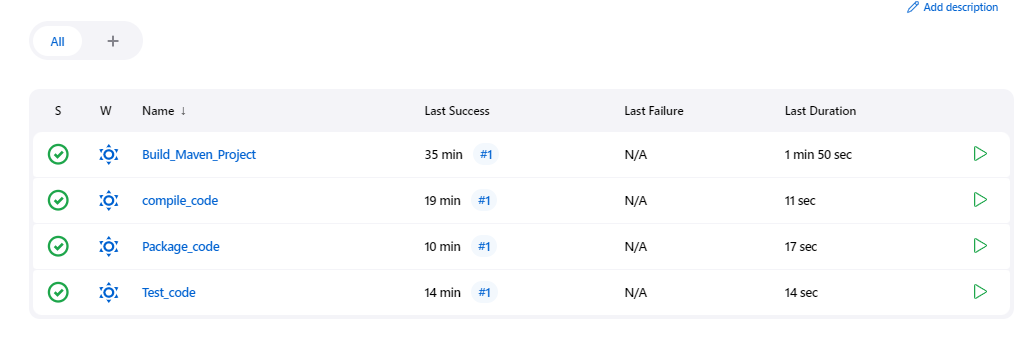
Similarly go to **Test\_job** in the post build action select build other project in that give the name of job that we want to build after the the **Test\_job** and that is **Package\_job**

Click on save button

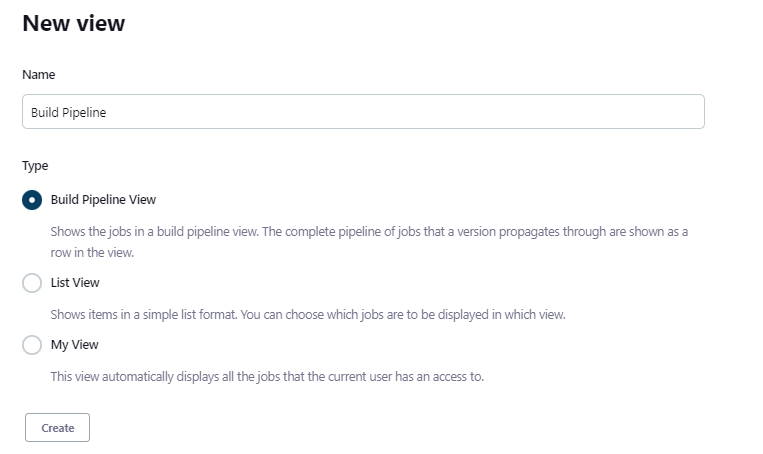


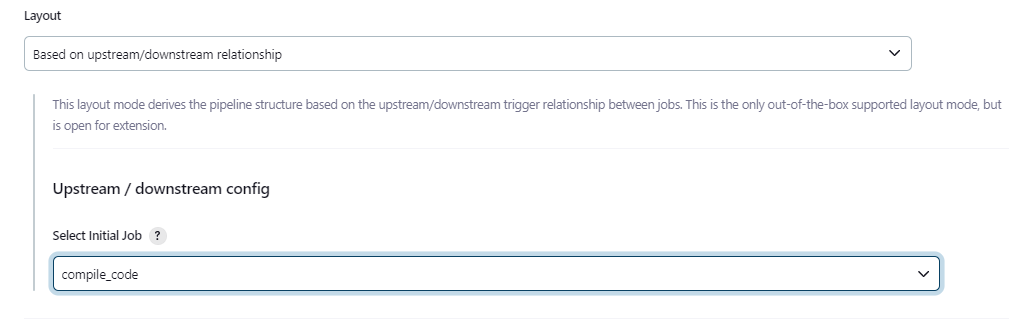
Now our pipeline gets created by by in the form of upstream and downstream now we want now we want to look ok of building this three job after one after another

**View a pipeline**

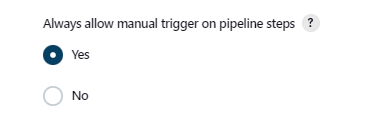
For that go to junk in dashboard click on plus sign of of all button where below that plus button we can see all jobs

click on click on plus button it will open up new view Provide the name of pipeline as build pipeline and and select the type as build pipeline view click on create button it will open some configuration

In the upstream downstream config provide the name of initial job where we want to to execute pipeline so and giving compile code as a my initial job

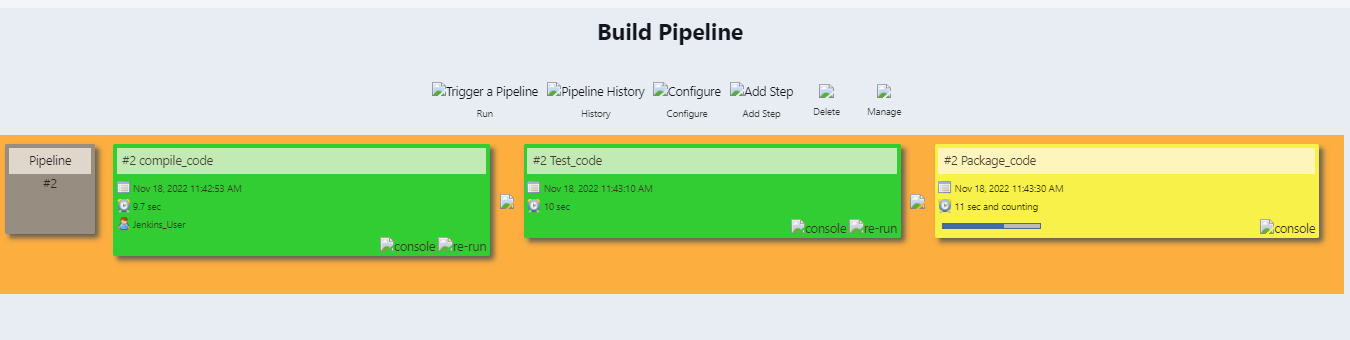


**Then select yes as always allow manual Trigger on pipeline step options**

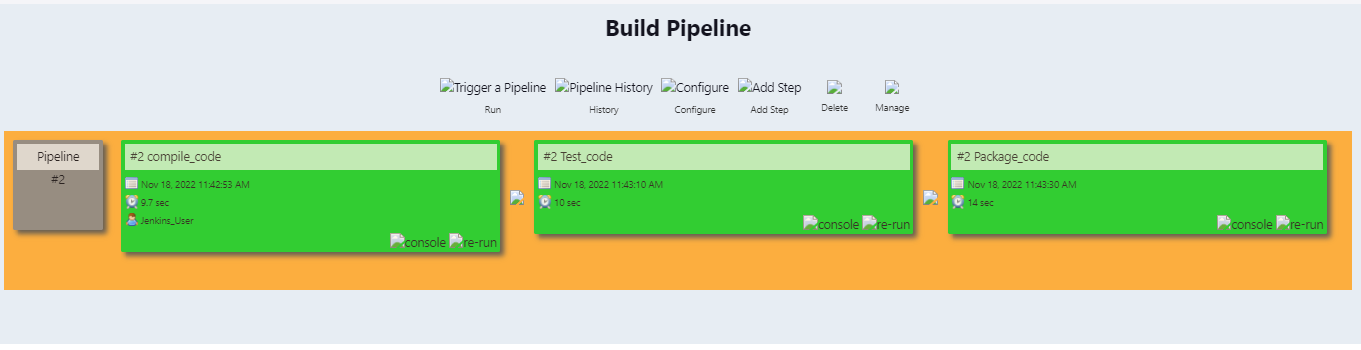


Click on save and it will redirect in to to pipeline view page as shown in screenshot if we click on Run button the pipeline get started executing as shown in screenshot

The blue colour indicates the job not get started yellow colour indicates that the job is is running or building and the green colour indicates that the job get builded successfully



Is the job are building a successful e then the whole pipeline show in a green colour and if not it will show show in a red colour but the pipeline gets stuck where the job ab is not builded successfully.



So my pipeline is executed successfully as shown in screenshot

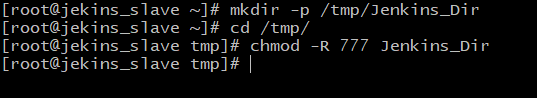
**Creating a master-slave jobs**

Now we want to distribute our pipeline task into master slaves slave setups

For distributing our job on slave machine, we need one more instance on ec2 so go to your AWS machine and launch one instance

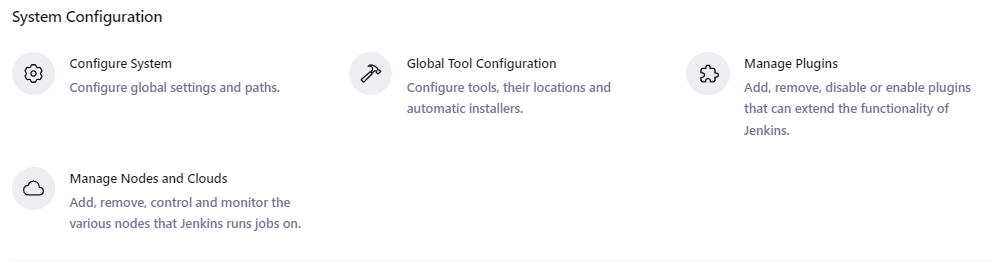
After launching the instance install get Maven and Java on slave machine

After installing create one directory where we want we want to store workspace created by Jenkins jobs as shown in screenshots



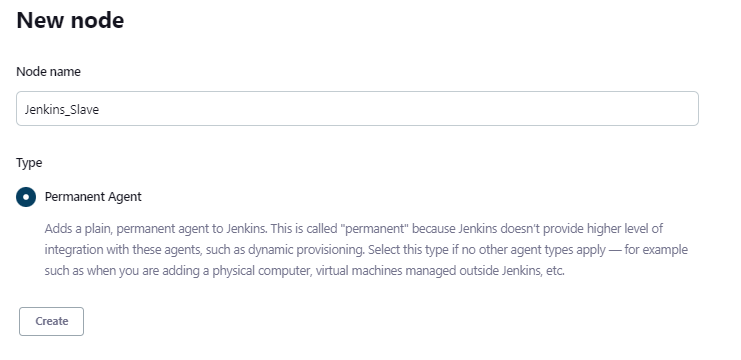
After creating Directory on Jenkins slave machine

go to Jenkins server go to manager manage Jenkins select manage nodes and clouds



In that provide the node name as Jenkins slave

Type permanent agent and click on create button

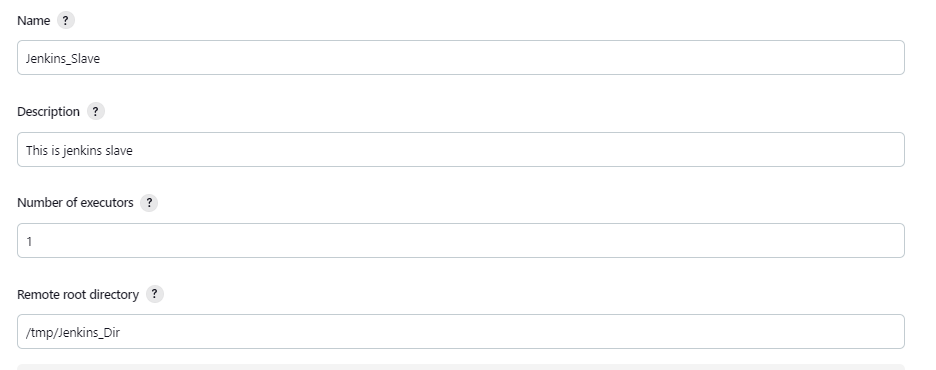


It will open some configure page

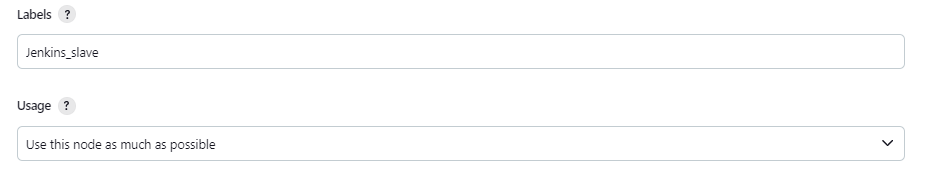
We need to do some configuration to connect our Jenkins slave with Jenkins server for that

Provide the name of description, number of executors and remote root directory of Jenkins slave machine as shown in screenshot

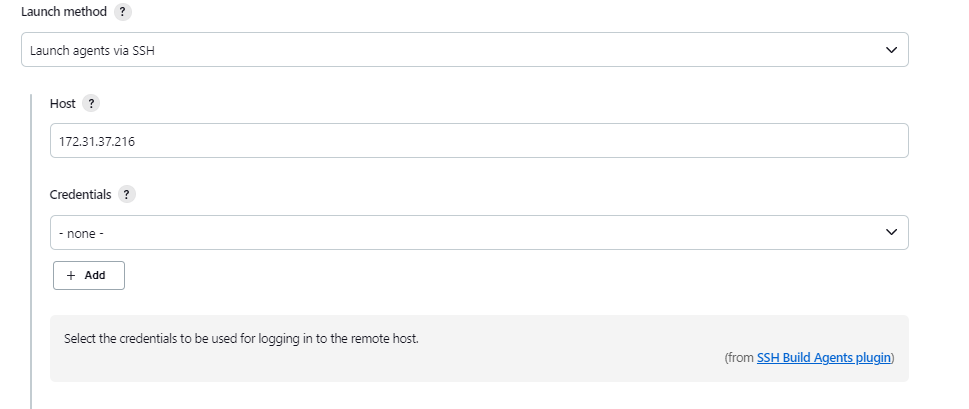
Remote root directory is the directory that we were created on slave machine



Then in label section provide the name that we will use in in packet job for or referencing our slave node and select usage as use this node as much as possible



Then in the launch method we will select launch agent via ssh in that will provide the private IP of Jenkins slave.

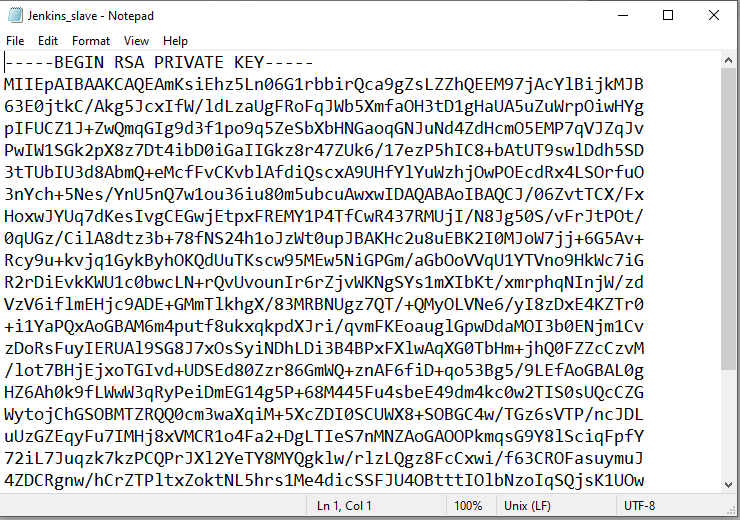


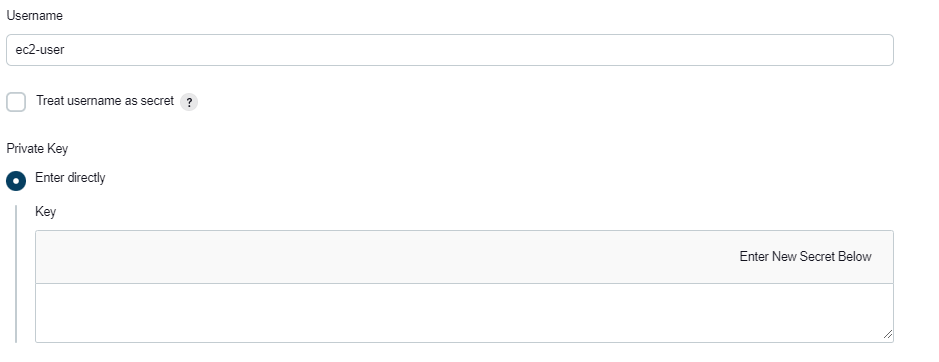
credential options we will add credential of kind ssh user name with private key, Provide the ID and description of our slave machine as shown in below screenshot.



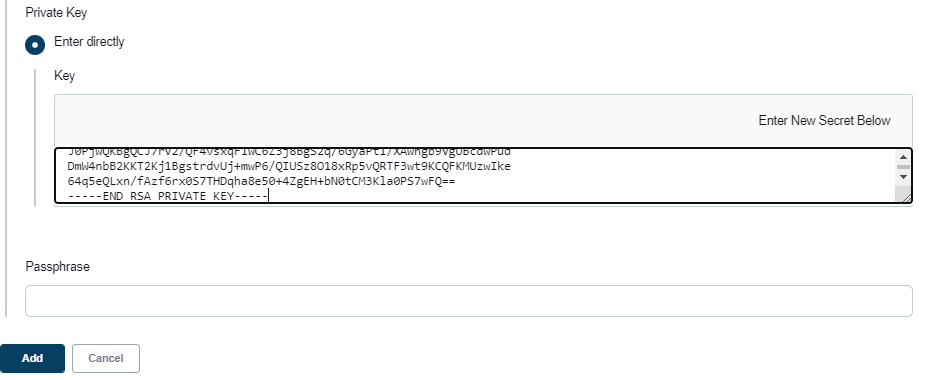
Then provide the username of our ec2 machine that is ec2 user and provide a private key which is downloaded while launching instance on ec2

open that file notepad file of the private key and paste it on private key section after selecting entire directly



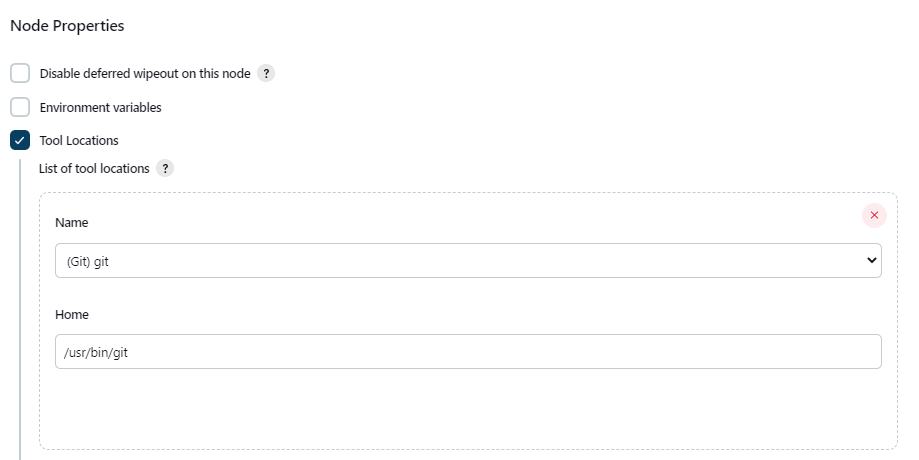


After private key is pasted click on add button then from dropdown select credential

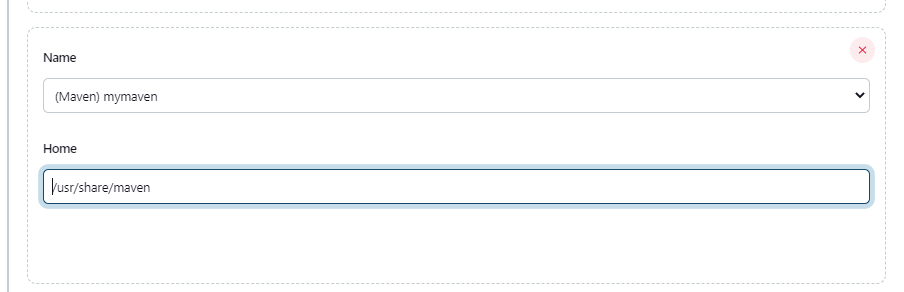


After that scroll down in two locations provide the tools which are install on or we want to use for our build process

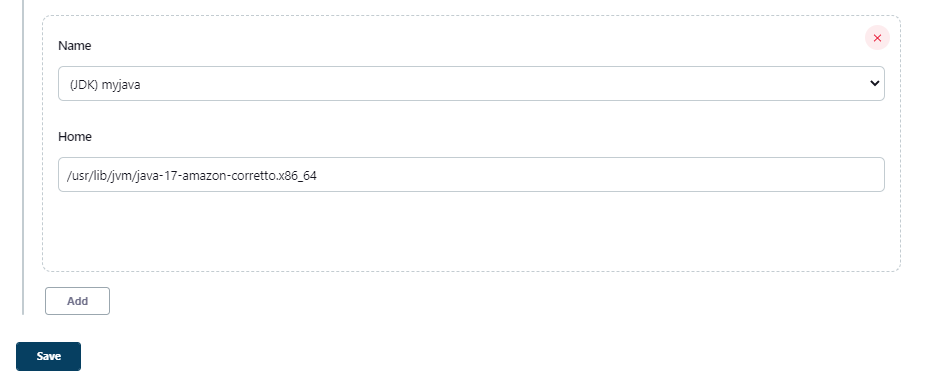
First give the name of gate and its path where the gate is installed



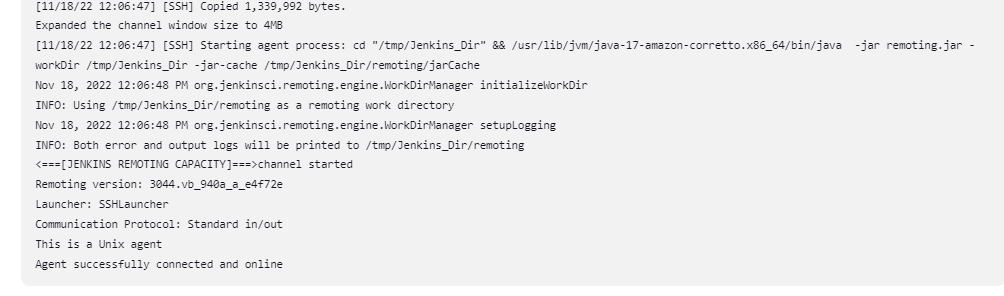
Then provide tool configuration of Maven as my Maven name and its path as shown in below



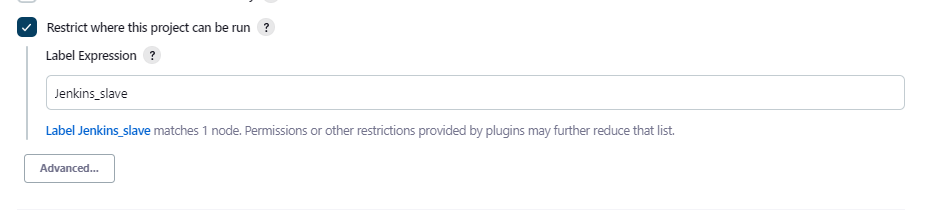
Finally provide the name of JDK and its part click on save button and come out.



If you go to console log it will show following output as shown in screenshot that agent successfully connected and online that means our slave machine get successfully connected to Jenkins master machine.



Then go to package job in the configure section choose restrict where this project can be Run and in label expression provide the label of Jenkins machine that we are configure before as shown in screenshot



click on save and build job if you go to console output and check it will show that the bill process started by user Jenkins user as shown in screenshot



If build get success it will give following output as shown in screenshot

