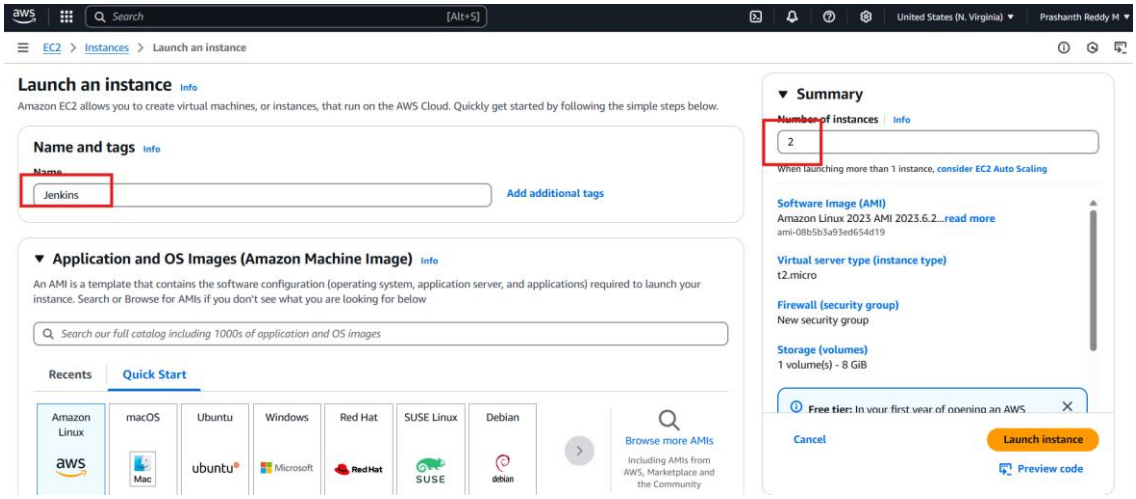
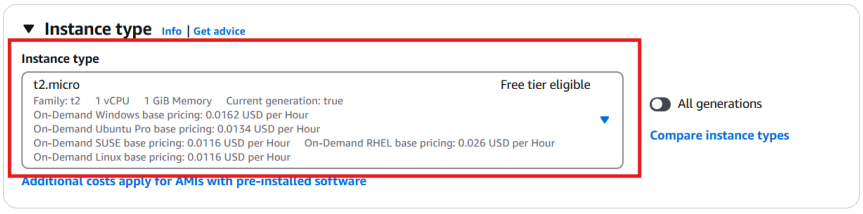
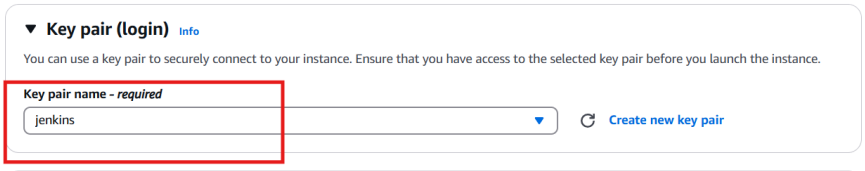
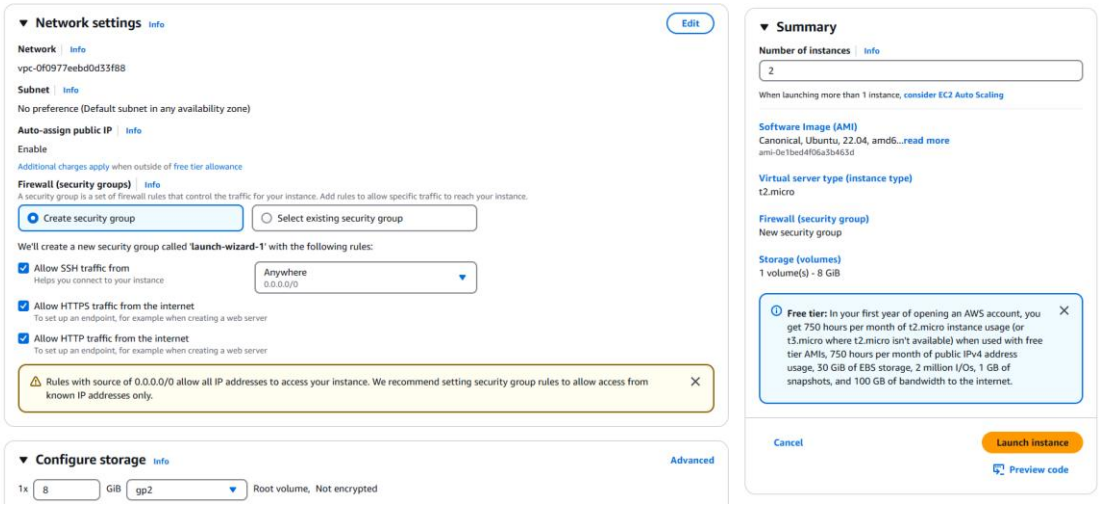
Jenkins works on master and agent architecture.

Launch 2 machines at a time.

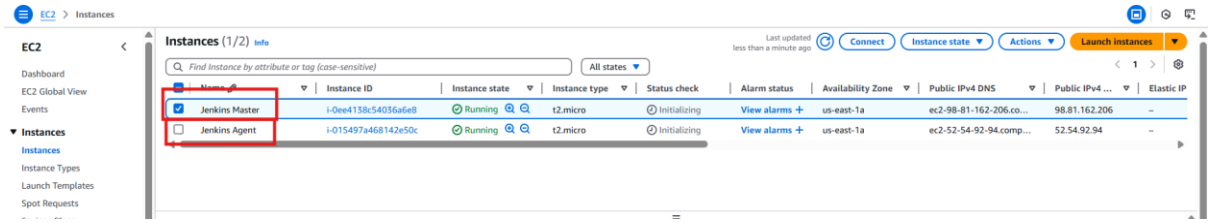




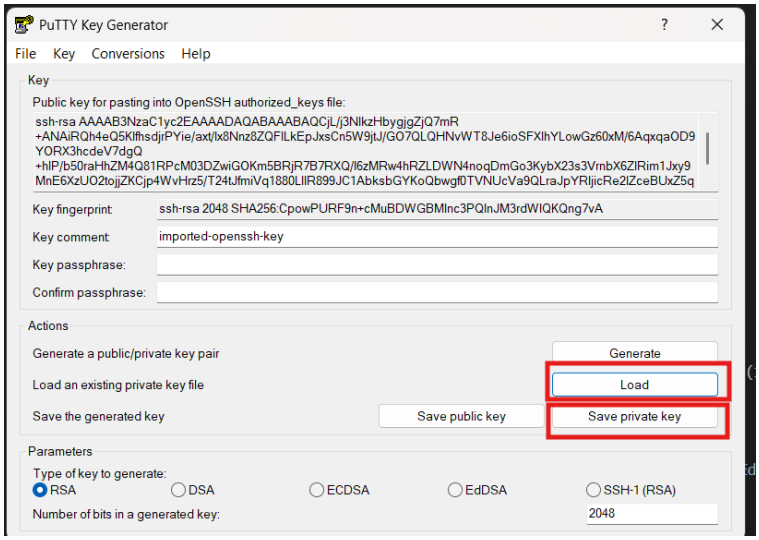




Rename the machines as Master and agent



Open PuTTYgen and load the .pem key and Save as private key as .ppk



Open the Linux master machine and run the following commands

➔ ***sudo apt-get update***

Use this for updated commands or follow the bellow steps
https://www.jenkins.io/doc/book/installing/linux/

Install Java JDK file in your master machine

➔ ***sudo apt install fontconfig openjdk-17-jre***

➔ ***java -version***

Install Jenkins in master

➔ ***sudo wget -O /usr/share/keyrings/jenkins-keyring.asc ***
https://pkg.jenkins.io/debian/jenkins.io-2023.key

➔ ***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

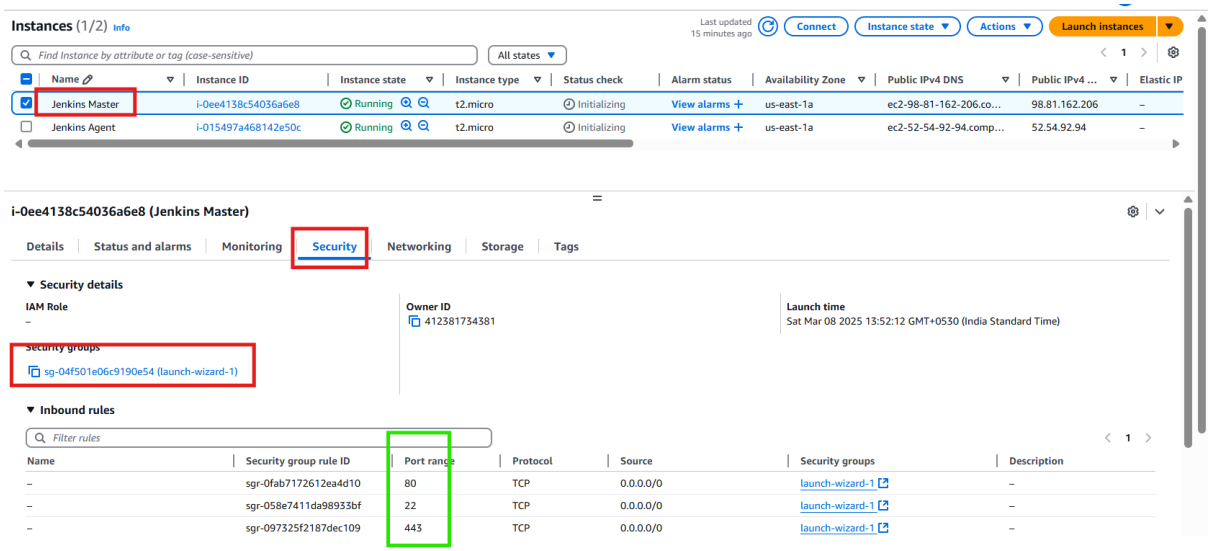
➔ ***sudo apt-get update***

➔ ***sudo apt-get install jenkins***

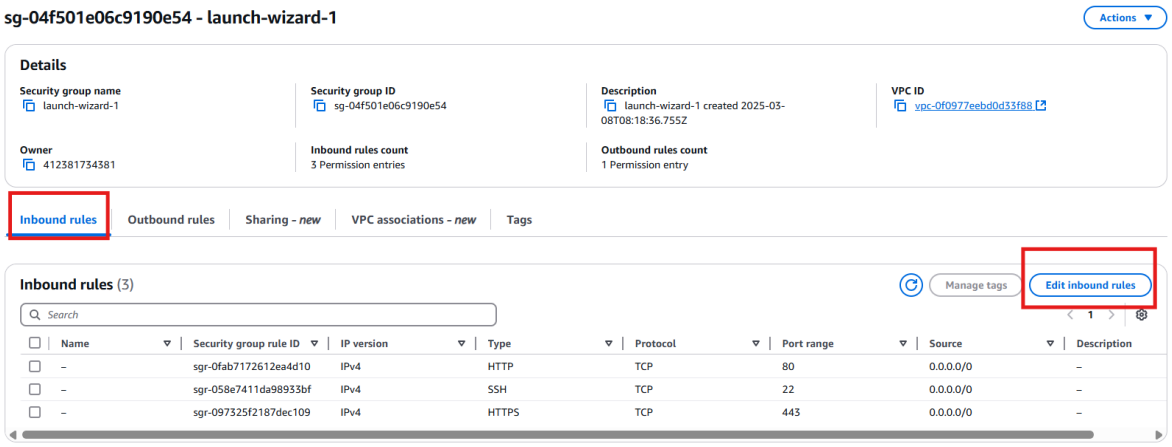
Now Jenkins are installed on your machine

Open Jenkins on web portal with IP: 8080

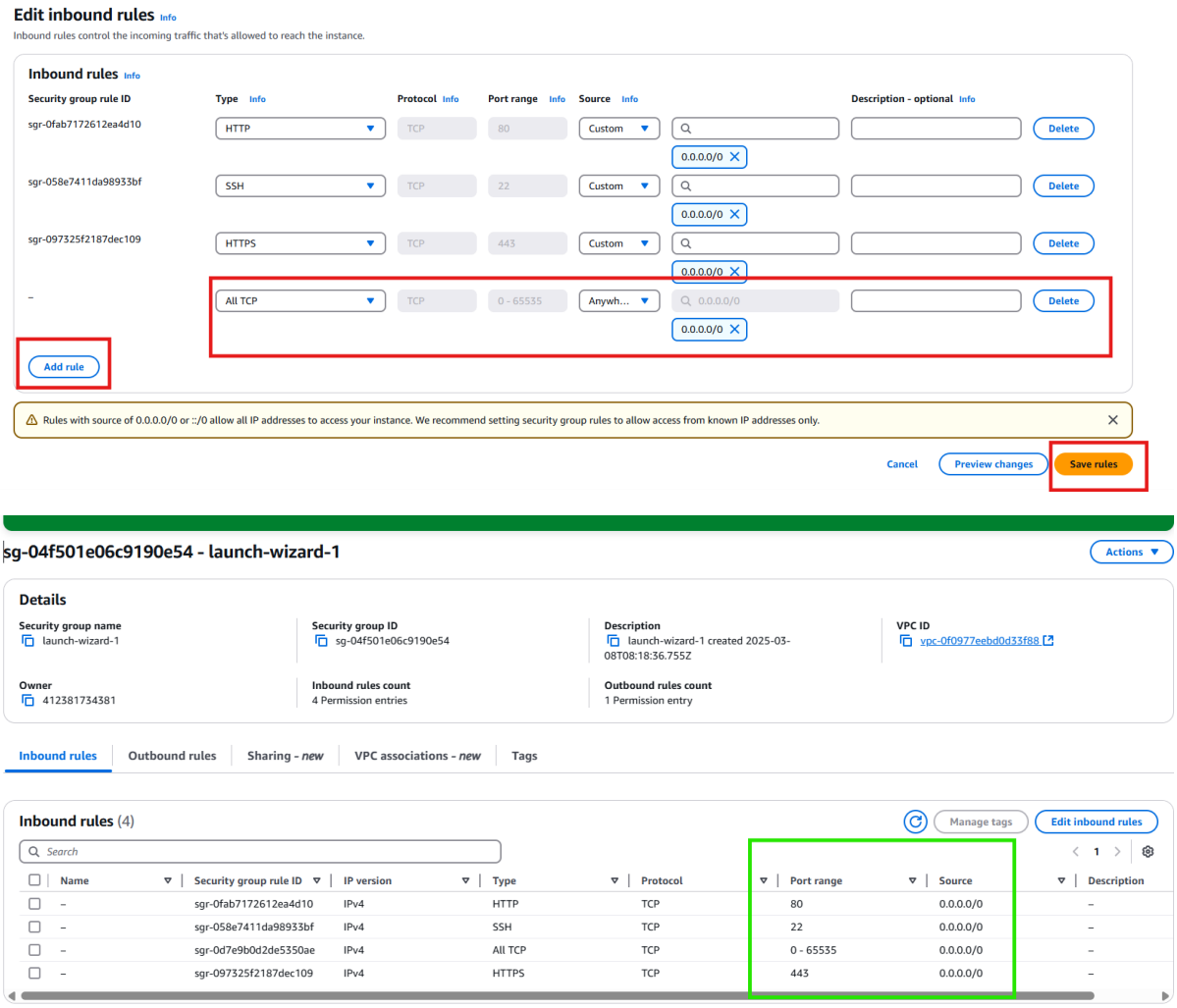
Port 8080 is not opened in the security group, now we are open that by going to AWS Console.



Click on security group

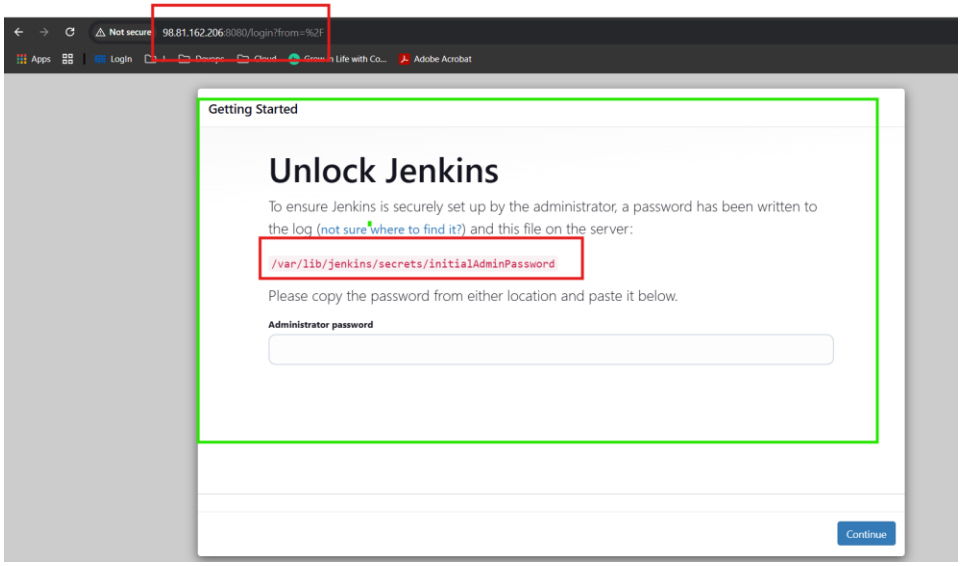


Add the rule and save



Enter IP address:8080 in the web browser you see the following page

Like ***http://98.81.162.206:8080/***

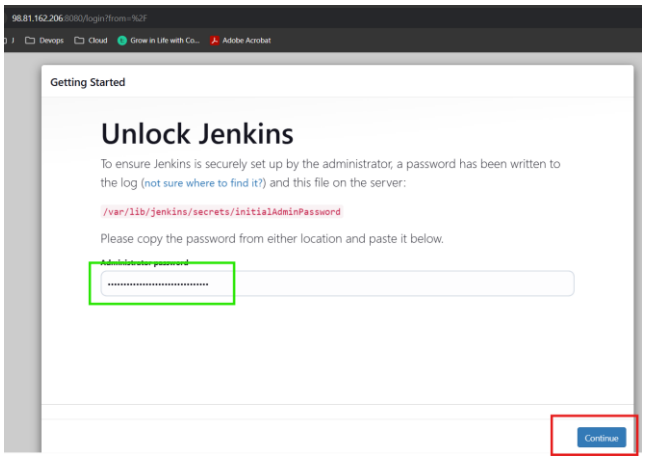


Copy the folder path and extract the password
/`var/lib/jenkins/secrets/initialAdminPassword`

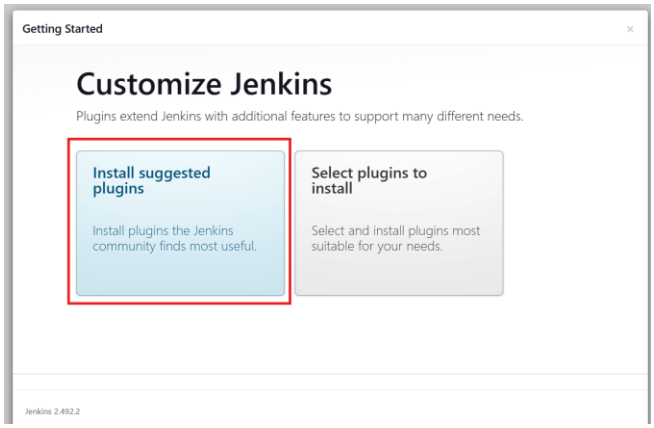
➔ ***sudo cat /var/lib/jenkins/secrets/initialAdminPassword***

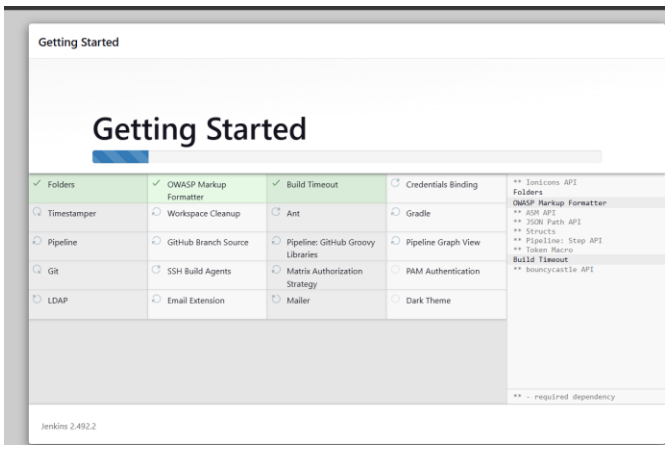
```
ubuntu@ip-172-31-21-29:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
3eb0cbe5dc044242a51dae1e49f9ff7a
ubuntu@ip-172-31-21-29:~$
```

Copy the password and paste in the web browser

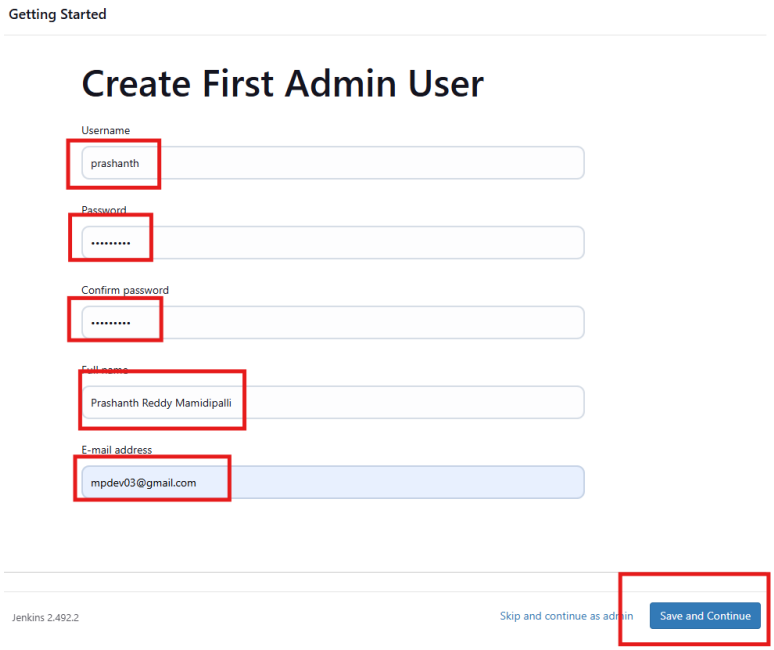


Click on Install suggested plugins



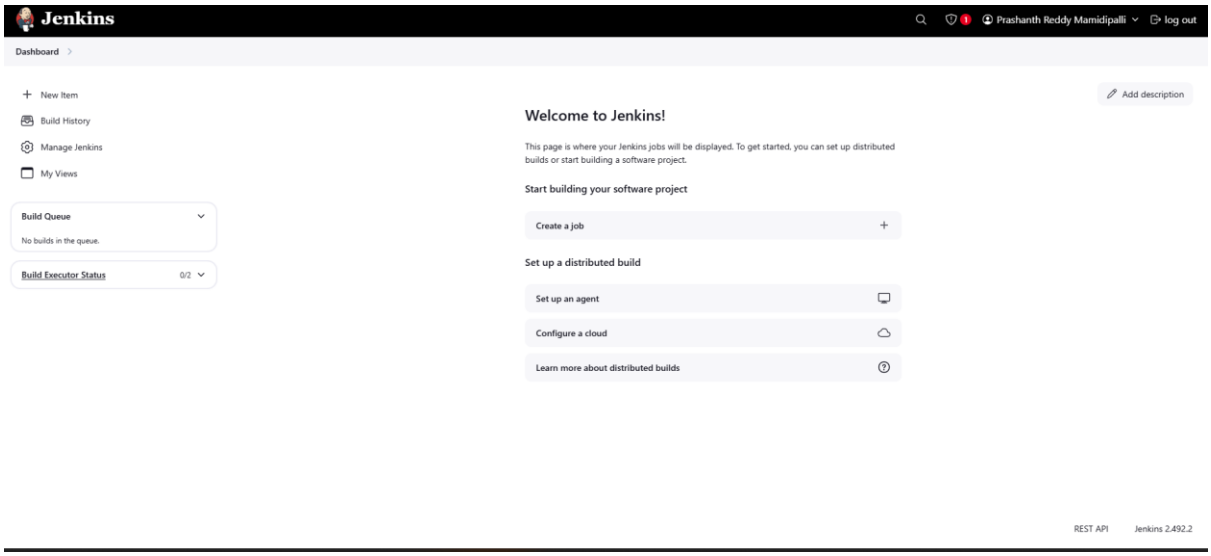


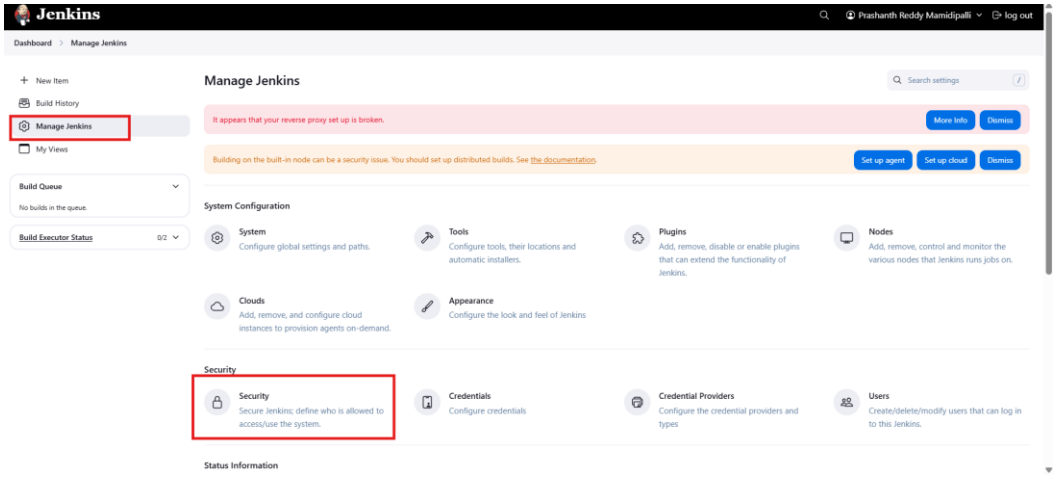
Summit your details and continue

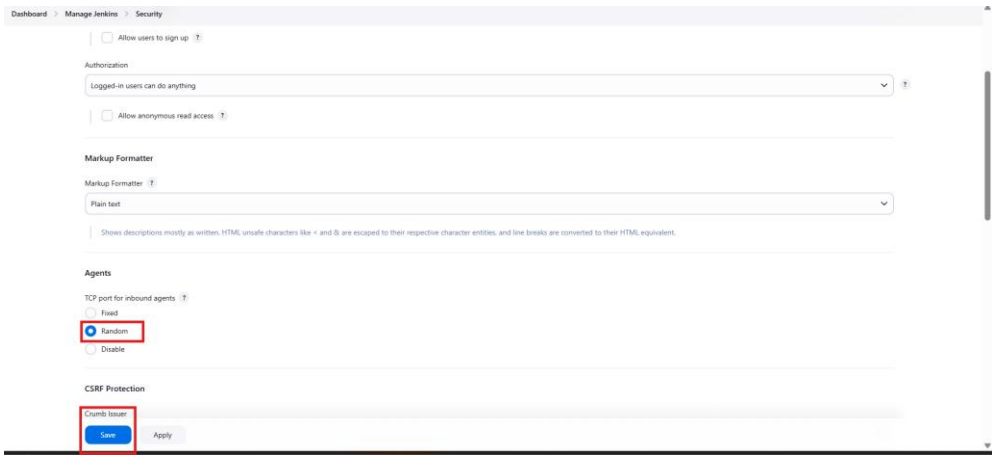


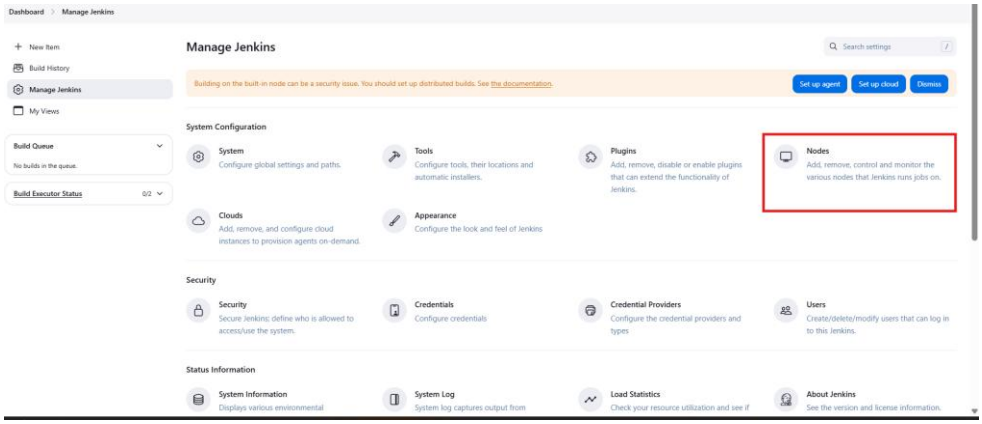
Click on **save and finish** and **Start Jenkins**

You can see the web page like this

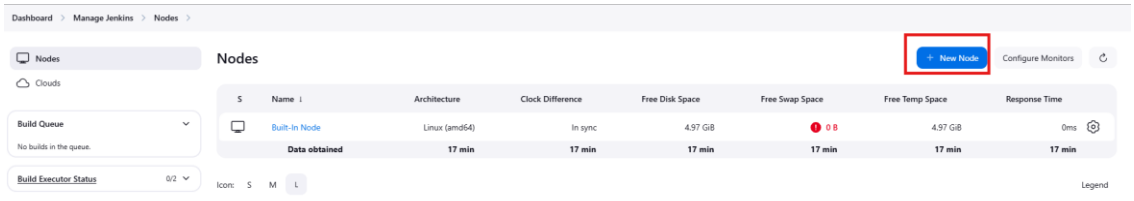




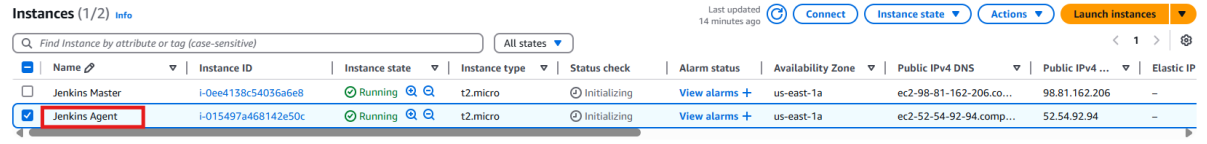


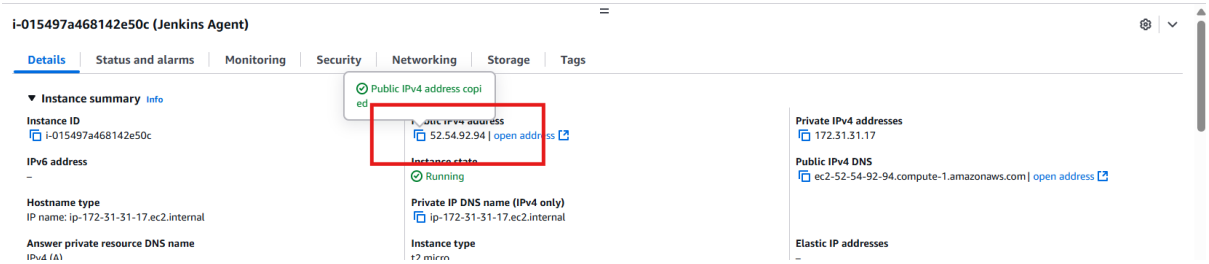


As of now we have only one node that is master we'll create Agent node and attach to this master node



Open Agent Linux in CLI





Open the Linux Agent machine and run the following commands

➔ ***sudo apt-get update***

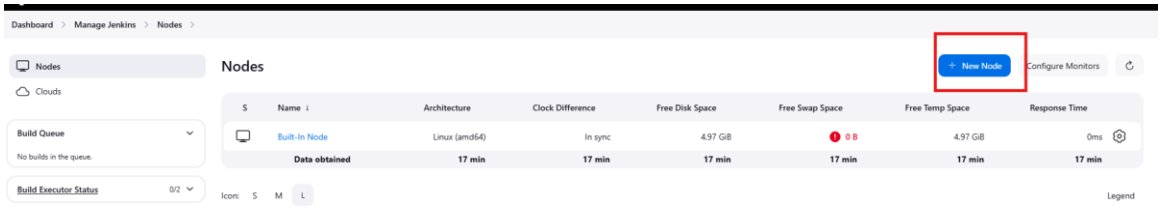
Install Java JDK file in your Agent machine

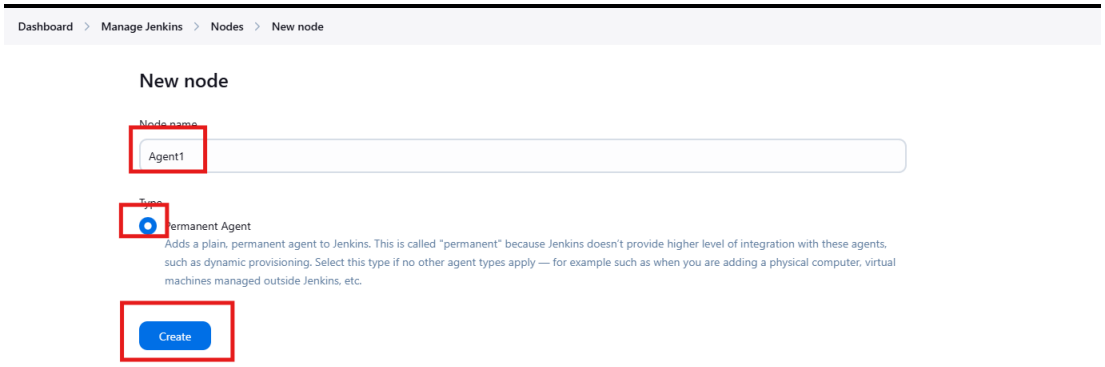
➔ ***sudo apt install fontconfig openjdk-17-jre***

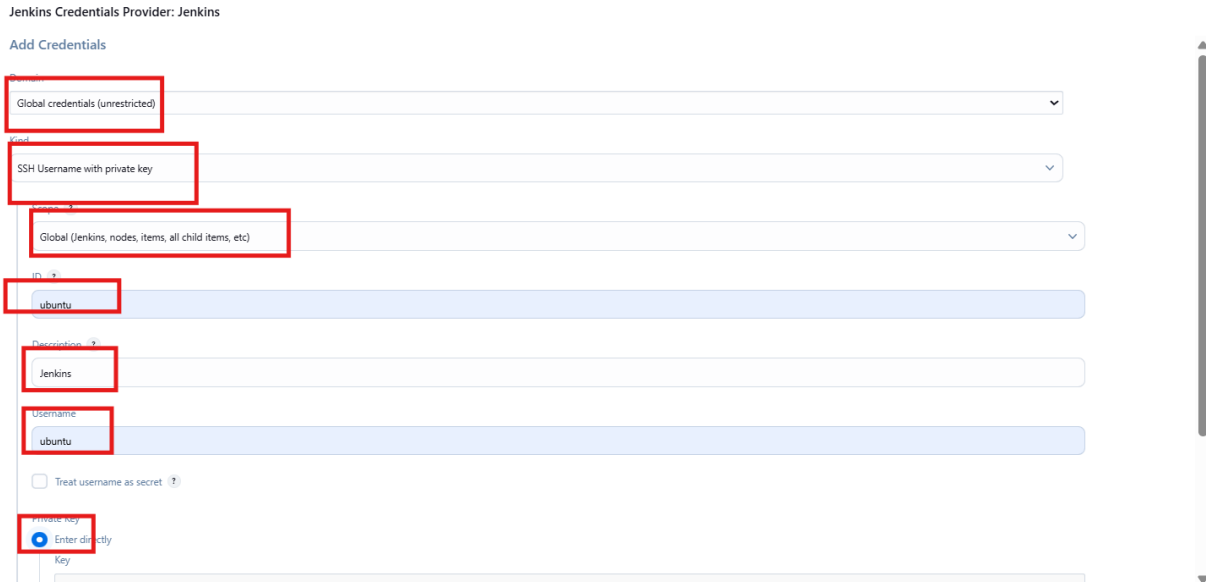
➔ ***java -version***

```
ubuntu@ip-172-31-31-17:~$ sudo apt install fontconfig openjdk-17-jre -y
Reading package lists... Done
```

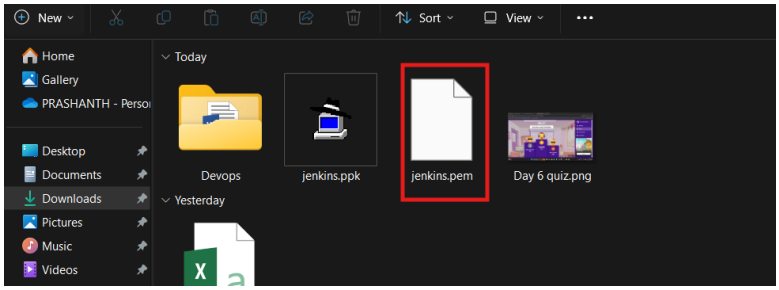
```
ubuntu@ip-172-31-31-17:~$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-122.04.1, mixed mode, sharing
ubuntu@ip-172-31-31-17:~$
```



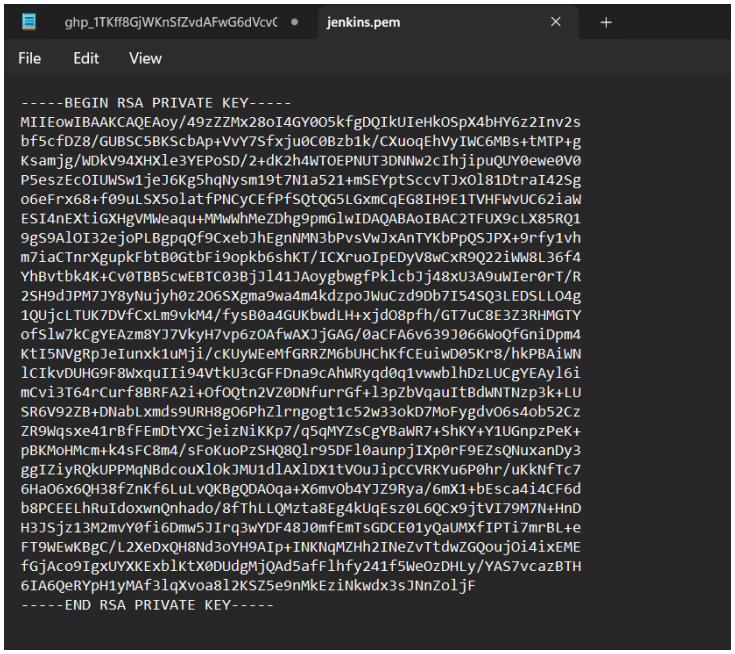


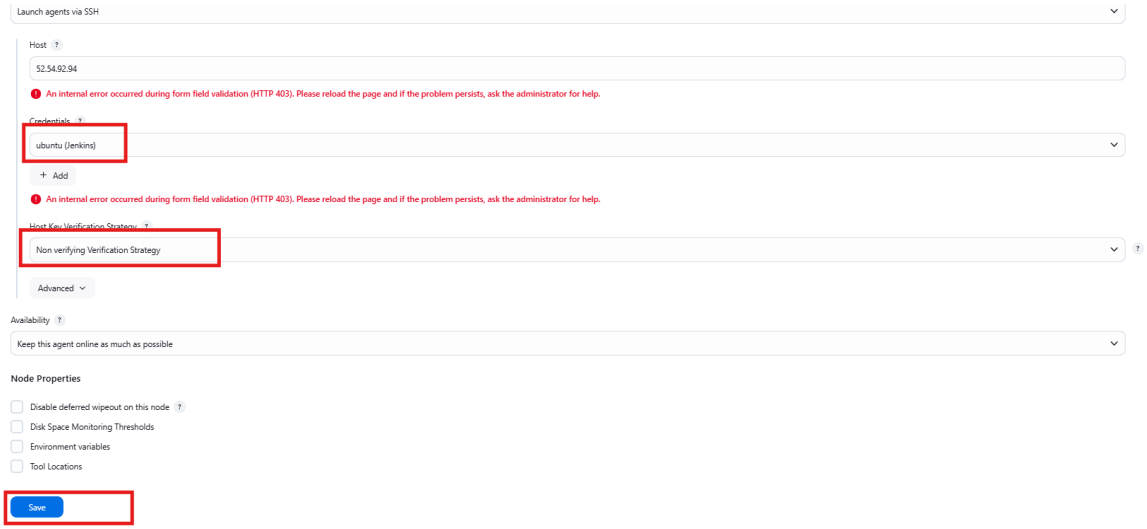


Open your .pem file copy the code

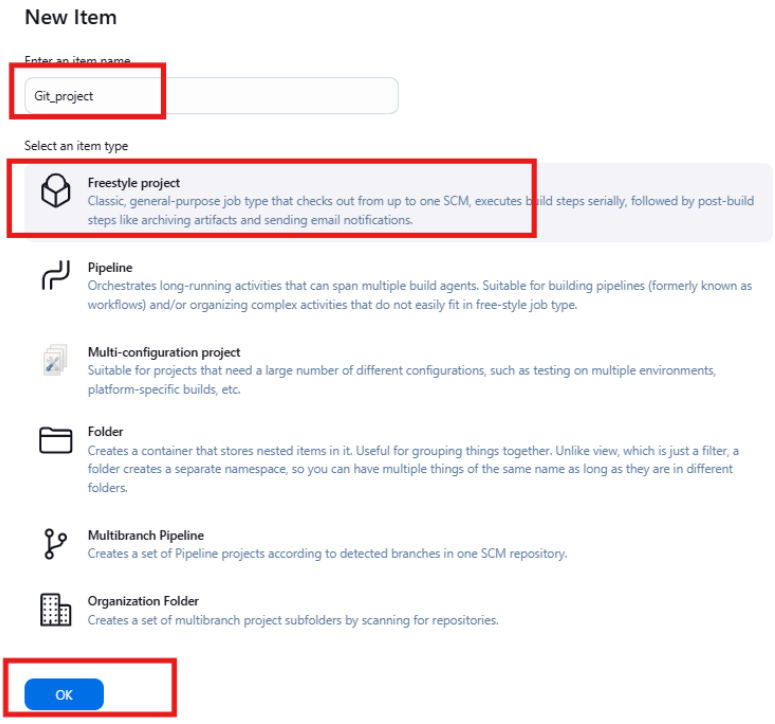


Open it in the notepad and copy the private key

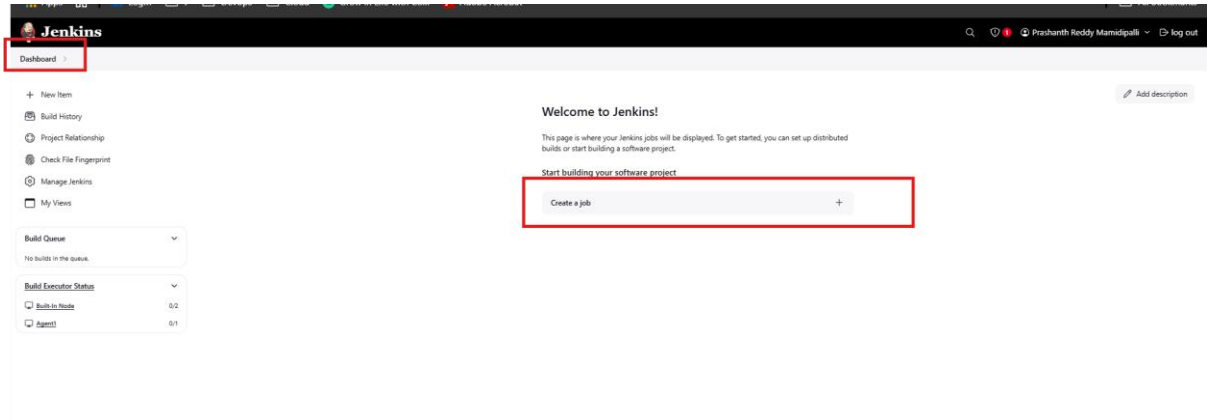


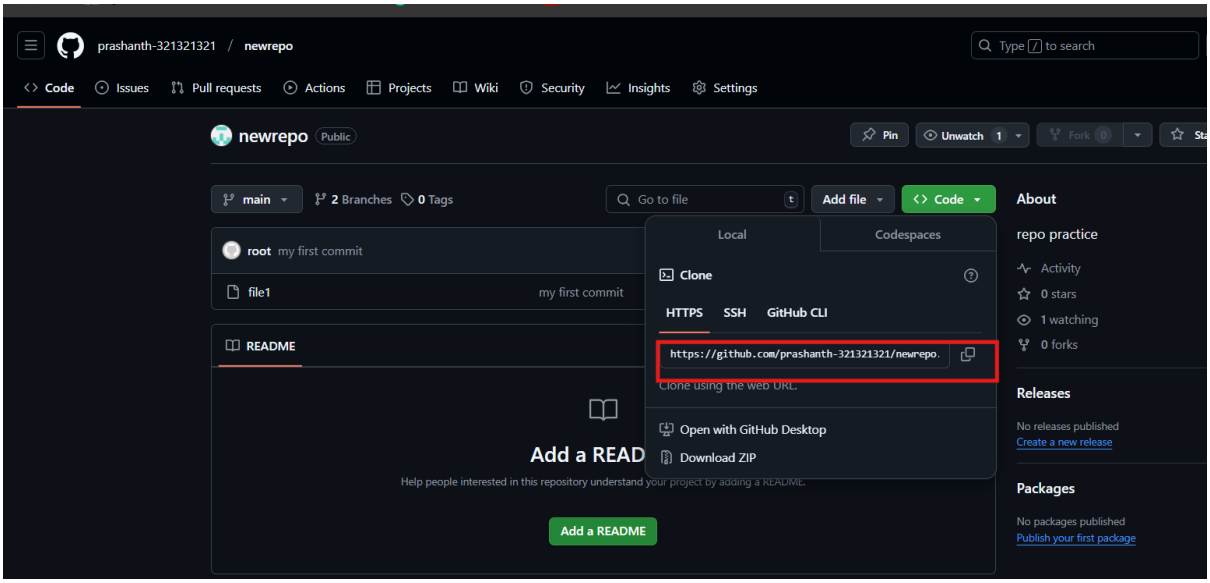

Launch agents via SSH
Host
52.54.92.94
An internal error occurred during form field validation (HTTP 403). Please reload the page and if the problem persists, ask the administrator for help.
Credentials
ubuntu (Jenkins)
Add
An internal error occurred during form field validation (HTTP 403). Please reload the page and if the problem persists, ask the administrator for help.
Host Key Verification Strategy
Non verifying Verification Strategy
Advanced
Availability
Keep this agent online as much as possible
Node Properties
Disable deferred wipeout on this node
Disk Space Monitoring Thresholds
Environment variables
Tool Locations
Save

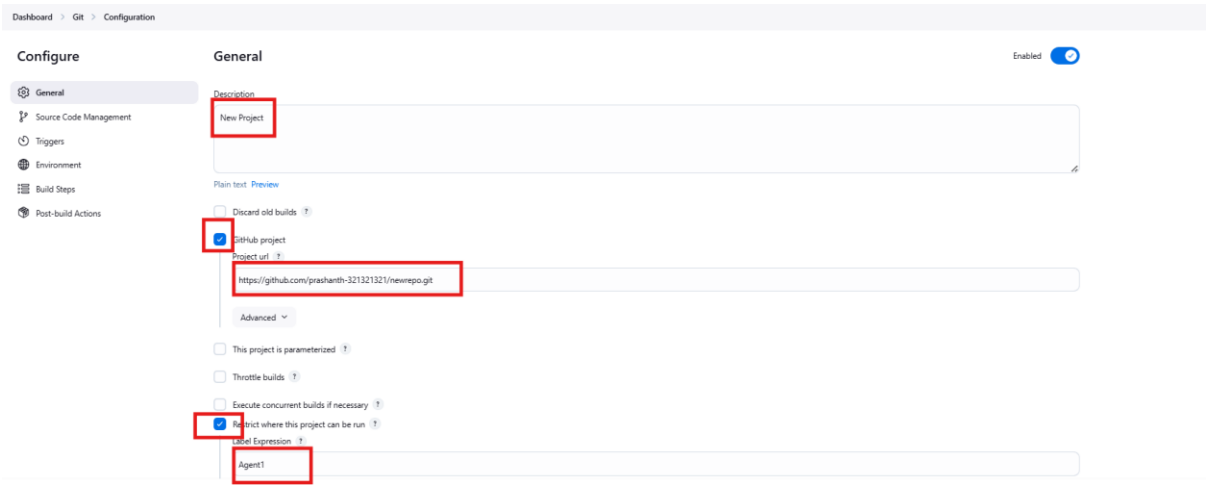
Nodes
Clouds
Build Queue
No builds in the queue.
Build Executor Status
Built-In Node 0/2
Agent1 0/1
Nodes
New Node
Configure Monitors
Name
Architecture
Clock Difference
Free Disk Space
Free Swap Space
Free Temp Space
Response Time
Agent1 Linux (amd64) In sync N/A 0 B N/A 164ms
Built-In Node Linux (amd64) In sync 4.77 GiB 0 B 4.77 GiB 0ms
Data obtained 0.2 sec 0.12 sec 3.4 sec 65 ms 3.4 sec 0.13 sec
Icon: S M L
Legend

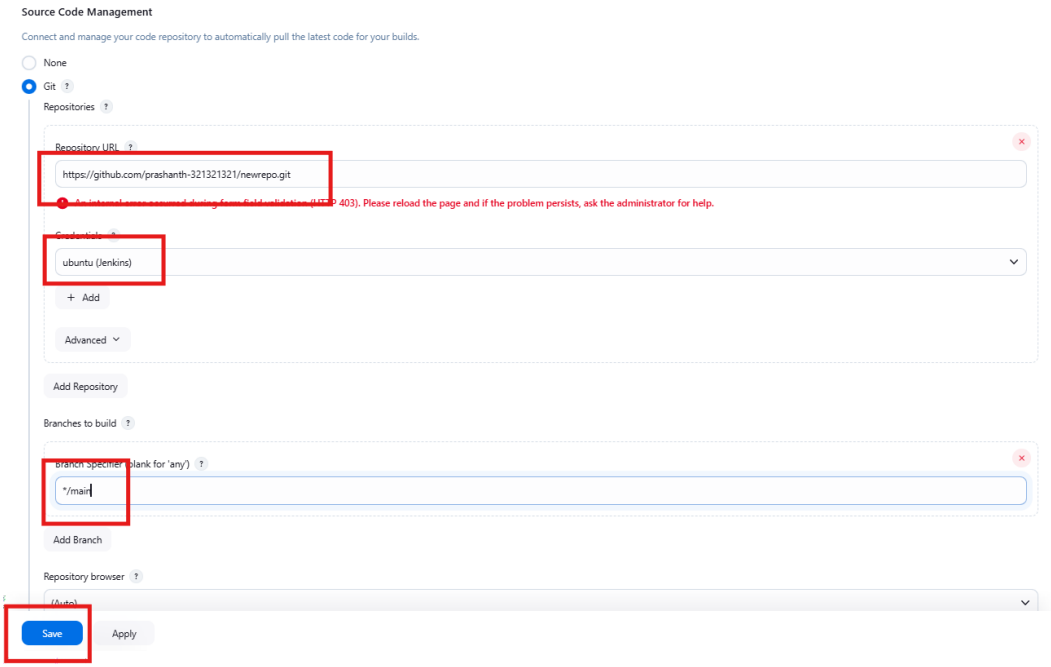

New Item
Enter an item name
Git_project
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.
OK

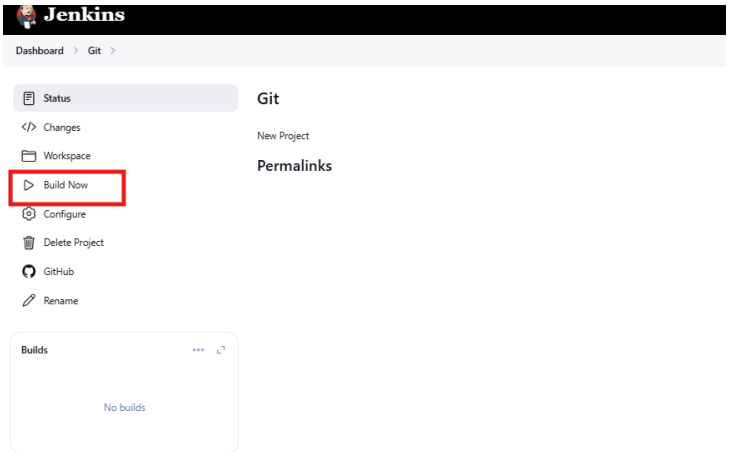
Now agent is added, we create a git project






Dashboard > Git > Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
General
Enabled
Description
New Project
Plain text Preview
Discard old builds
GitHub project
Project url
https://github.com/prashanth-321321321/newrepo.git
Advanced
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression
Agent1


Source Code Management
Connect and manage your code repository to automatically pull the latest code for your builds.
None
Git
Repositories
Repository URL
https://github.com/prashanth-321321321/newrepo.git
Credentials
ubuntu (Jenkins)
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Add Branch
Repository browser
(Auto)
Save
Apply


Jenkins
Dashboard > Git >
Status
Changes
Workspace
Build Now
Configure
Delete Project
GitHub
Rename
Builds
No builds
Git
New Project
Permalinks