Create a Master and Slave environment in Jenkins and run app deployment job in slave

Setting up a master-slave environment in Jenkins allows you to distribute build and deployment jobs across multiple machines.

Solution:

Connection Requirements:

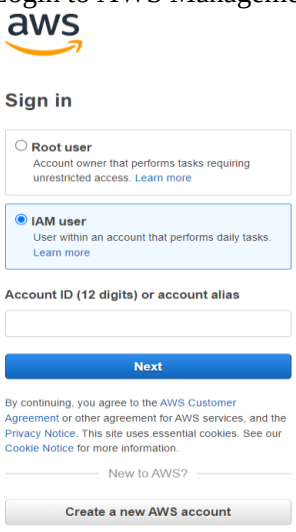
Java must be same version installed in Master and Slave.

Installation:

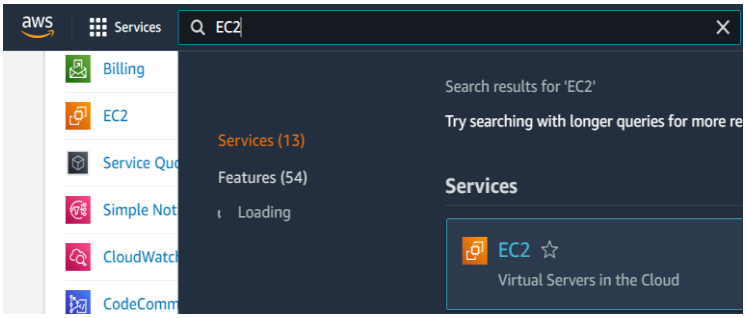
1. EC2
2. Jenkins
3. Docker

EC2

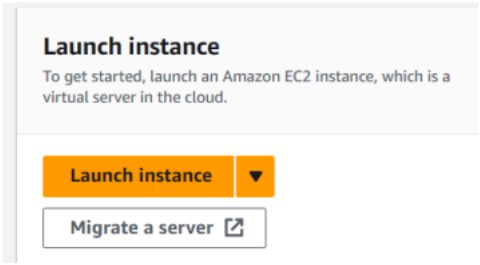
- Create EC2 instance in AWS for installing Jenkins server.
- Login to AWS Management console



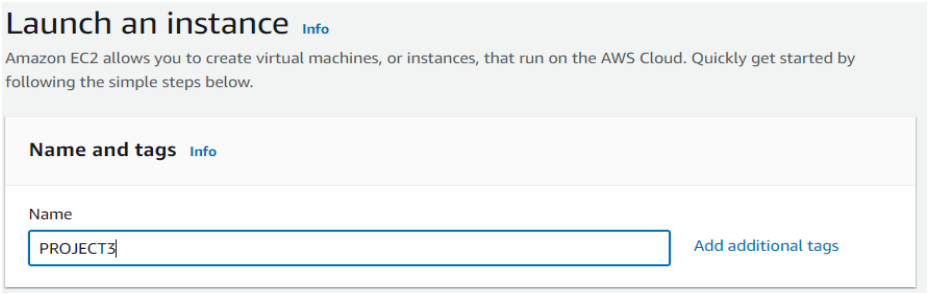
- Choose EC2 and click launch instance



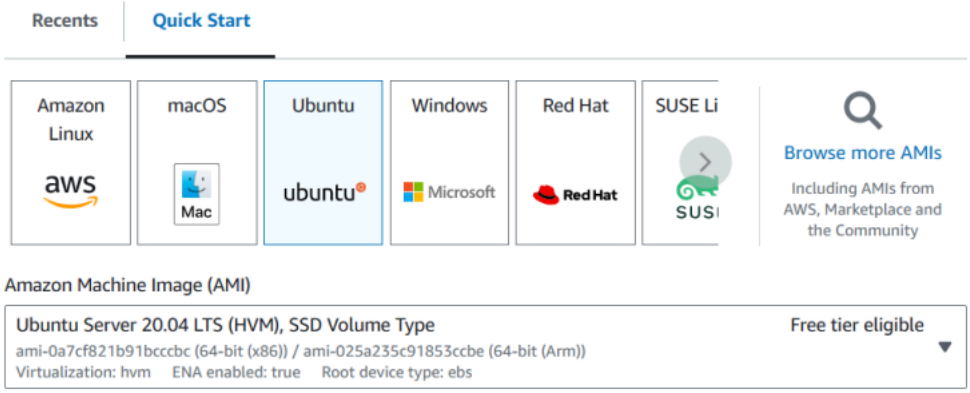
- Then under the EC2 management console we could able to find the Launch instance option, click that one for creating an instance:



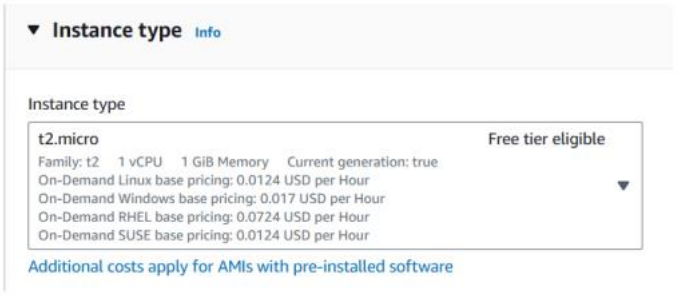
- Then name the instance according to your preferences,



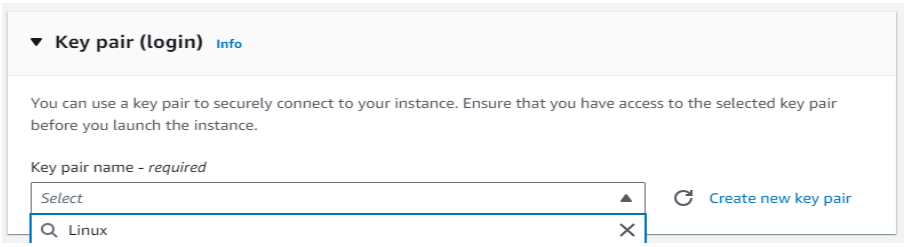
- Then we need to select an AMI Image for the instance, choose AMI image according to your preferences: Here I am selecting Ubuntu 20.04 AMI Image.



- Then we need to select the instance type

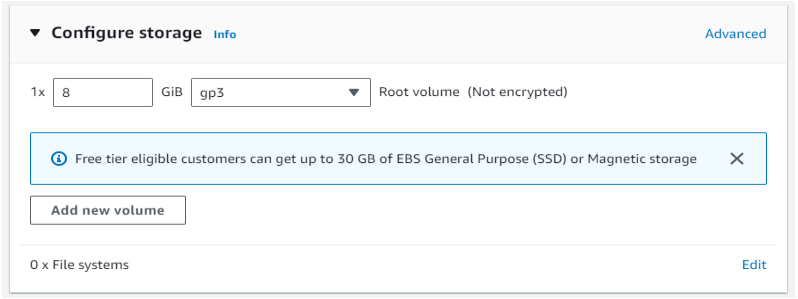


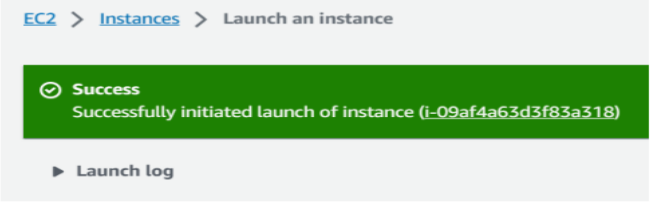
- Then we need to select the keypair for security authentication purpose of the instance



- Then under network settings, I am selecting default VPC, subnet no preference option, auto-assign public is enabled by default for default VPC & Subnet.

- Configure storage and launch a instance.





Jenkins

- Install a Jenkins in our EC2 machine.

Prerequisites

Minimum hardware requirements:

- 256 MB of RAM
- 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)

- Jenkins requires Java to run, so we need to install java using this commands.

sudo apt update

```
ubuntu@ip-172-31-19-109:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1149 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [245 kB]
```

sudo apt install fontconfig openjdk-17-jre

```
ubuntu@ip-172-31-19-109:~$ sudo apt install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fontconfig is already the newest version (2.13.1-4.2ubuntu5).
fontconfig set to manually installed.
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wr
  openjdk-17-jre-headless
Suggested packages:
  default-jre libasound2-plugins alsa-utils pcscd libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wr
  openjdk-17-jre openjdk-17-jre-headless
0 upgraded, 13 newly installed, 0 to remove and 21 not upgraded.
Need to get 51.2 MB of archives.
After this operation, 204 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2-data all 1.2.6.1-1ubuntu1 [19.1 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2 amd64 1.2.6.1-1ubuntu1 [390 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 alsa-ucm-conf all 1.2.6.3-1ubuntu1.8 [43.3 kB
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72build2 [6782 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpcsclite1 amd64 1.9.5-3ubuntu1 [19.8 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre-headless amd64 17.0.8.1+1~
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [1
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2041 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
[##########################################################################################################]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgif7 amd64 5.1.9-2build2 [33.8 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre amd64 17.0.8.1+1~us1-0ubu
```

java –version

```
ubuntu@ip-172-31-19-109:~$ java -version
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-19-109:~$
```

- Once java installation completes next start installing Jenkins using this commands.

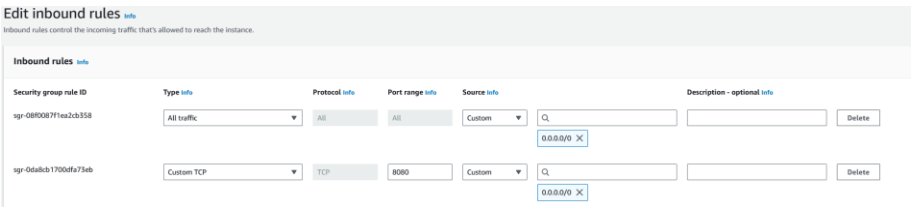
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \https://pkg.jenkins.io/debian/jenkins.io-2023.key
**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] **
**https://pkg.jenkins.io/debian binary/ | sudo tee **
/etc/apt/sources.list.d/jenkins.list > /dev/null

```
ubuntu@ip-172-31-19-109:~$ 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
--2023-11-04 11:41:37--  https://pkg.jenkins.io/debian/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.78.133, 2a04:4e42:83::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.78.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc         100%[=============================================================================>]

2023-11-04 11:41:37 (25.0 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
```

sudo apt-get update
sudo apt-get install Jenkins

```
sudo apt-get install jenkins
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:5 https://pkg.jenkins.io/debian binary/ InRelease
Get:6 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian binary/ Packages [57.4 kB]
Fetched 60.3 kB in 1s (64.7 kB/s)
Reading package lists... Done
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 21 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 90.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.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 binary/ jenkins 2.430 [89.0 MB]
Fetched 89.2 MB in 5s (16.5 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 124560 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 .../archives/jenkins_2.430_all.deb ...
Unpacking jenkins (2.430) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.430) ...
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) ...
```

- Once Jenkins installation finished we need to open port (8080) for accessing our Jenkins using aws security groups.
- Go to instance security groups and edit inbound rules and add 8080 with custom ip range and save.

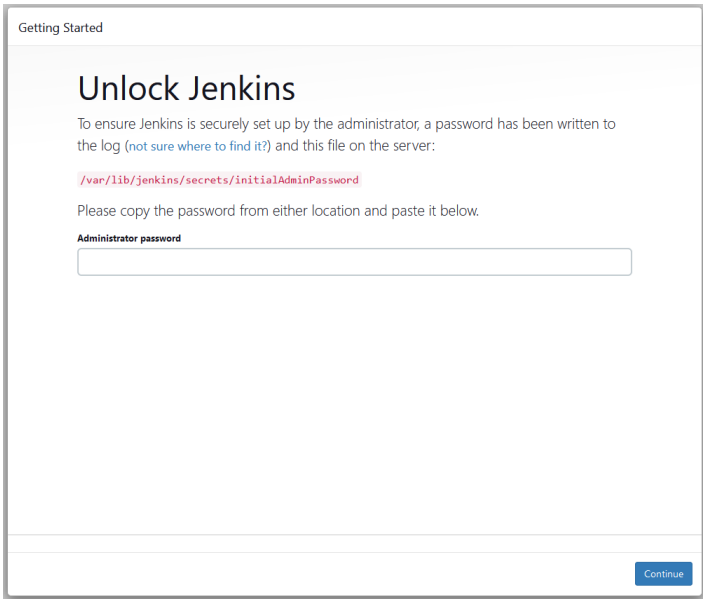


- Next Accessing our Jenkins server using our instance public ip follows with port 8080.

http://IP:8080

- Getting started Jenkins page will open paste our admin password.
- Use this command to get Jenkins admin password

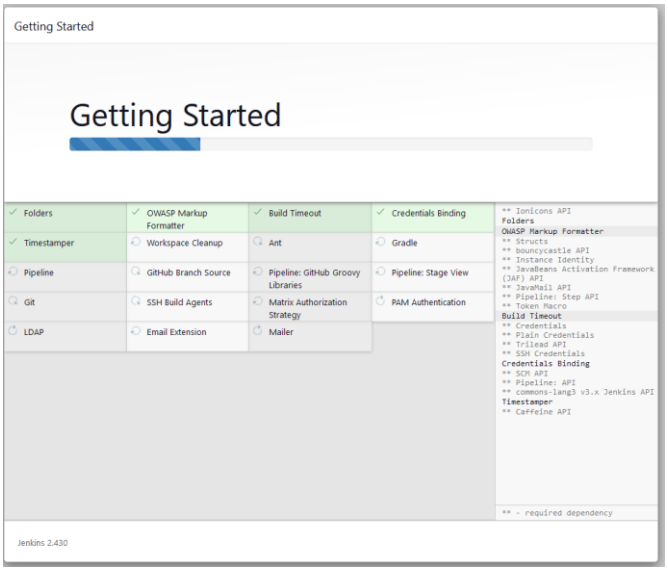
sudo cat /var/lib/jenkins/secrets/initialAdminPassword



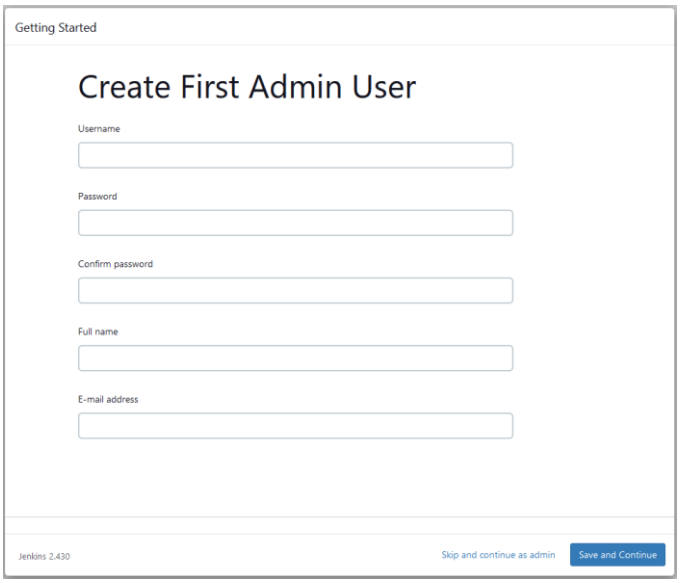

```
ubuntu@ip-172-31-19-109:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
b8d4fff8d5ce460089f8e6f191ac8115
ubuntu@ip-172-31-19-109:~$
```

- Copy and paste the password from terminal Jenkins page next click install suggested plugins.

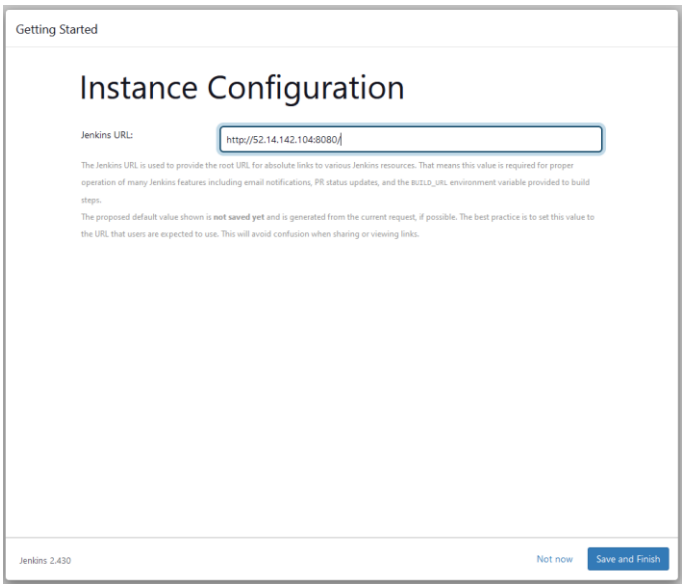
- It will install all plugins



- Create a user to work with Jenkins or continue as admin.



- Click save and continue to enter Jenkins dashboard.



- Go to Manage Jenkins >> Plugins >> install docker and pipeline plugins.

- Next use the following commands to install docker in master as well in slave machine also.

sudo apt update
sudo apt install docker.io

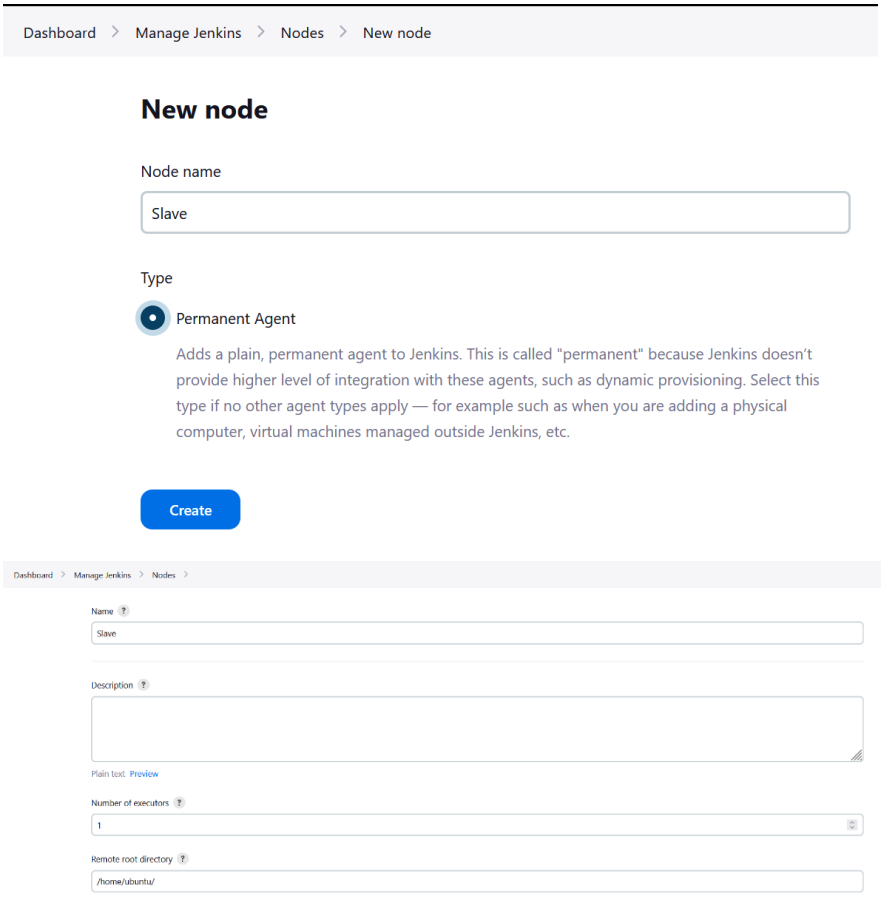
```
ubuntu@ip-172-31-10-49:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
```

```
ubuntu@ip-172-31-10-49:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 54 not upgraded.
Need to get 69.7 MB of archives.
After this operation, 267 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

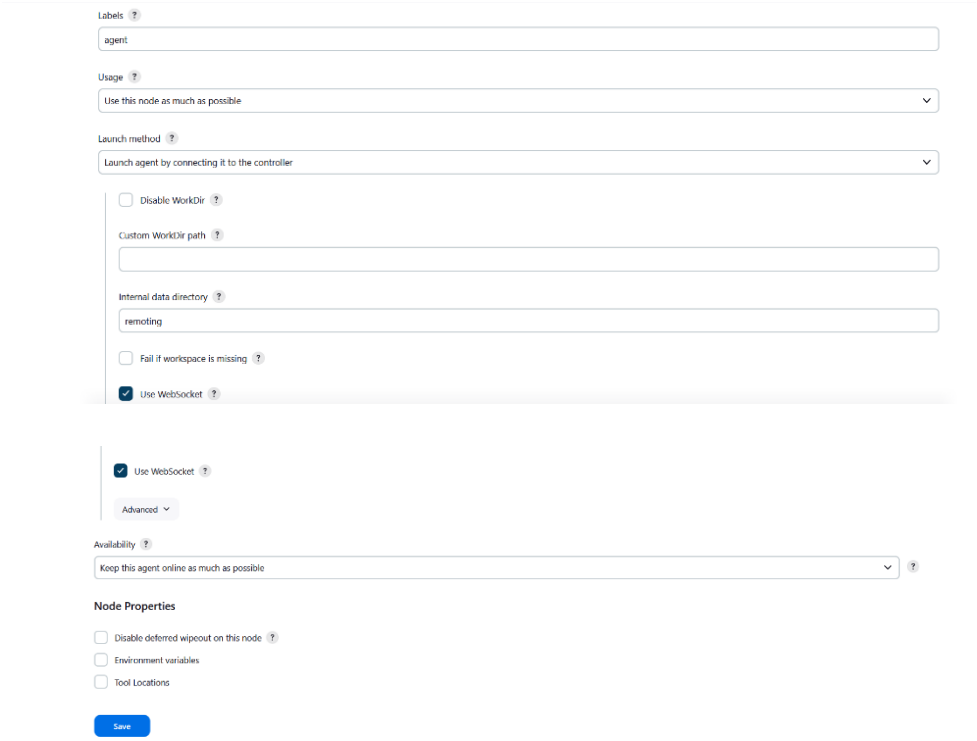
docker info

```
ubuntu@ip-172-31-10-49:~$ docker info
Client:
 Version:    24.0.5
 Context:    default
 Debug Mode: false
```

- Next Create a slave machine in Jenkins using manage Jenkins and go to nodes
- Name the slave and click create.

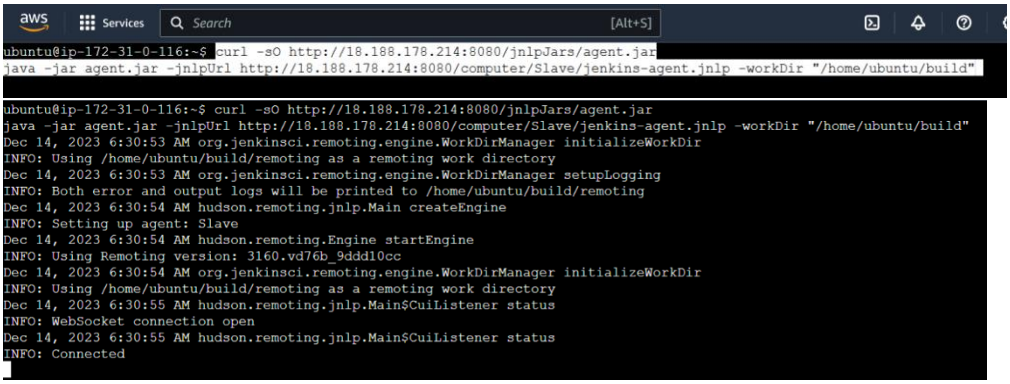


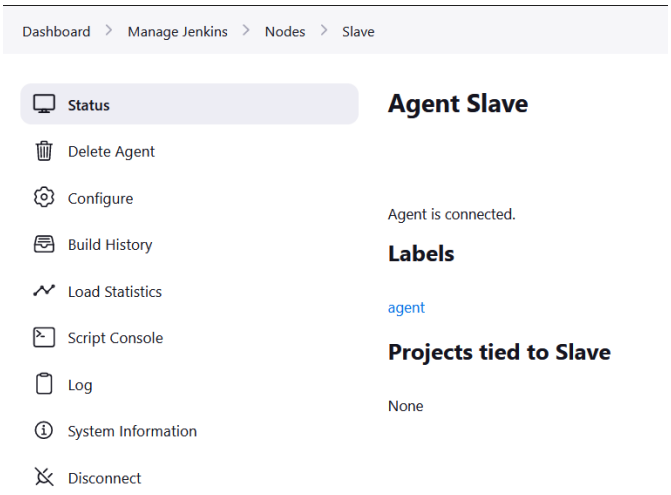
- Give the Label name for our slave that label will be used in all our codes.
- Choose Launch agent by controlling it to the container.
- Choose WebSocket and select availability keep agent as possible.



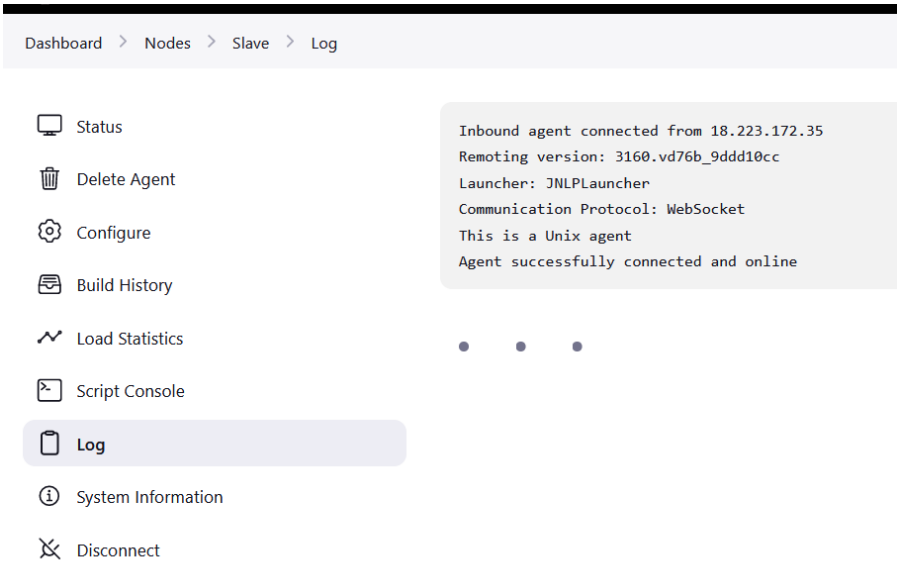
- Check the configurations and click save it will create some file and command for our Jenkins slave connection.
- Every time instance stopped the ip will be changed so you need to change the IP address of your Jenkins.
- Go to Manage Jenkins >> system >> Change the Jenkins URL with a new IP address.

- Go to the /home/Ubuntu directory and paste the commands provided by Jenkins master.
- The code will be download a jar file it will used to create a connection with our Jenkins master.



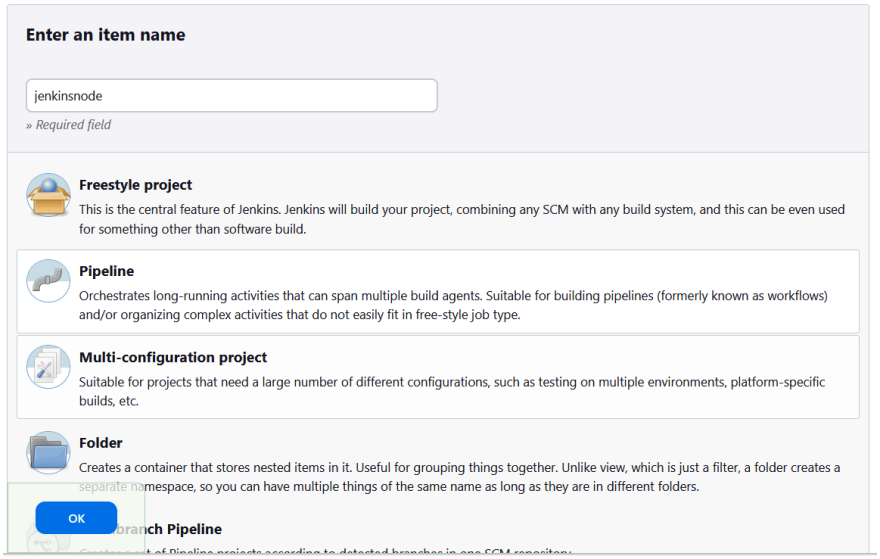


- Check logs of our Slave node it shows connected and online state.



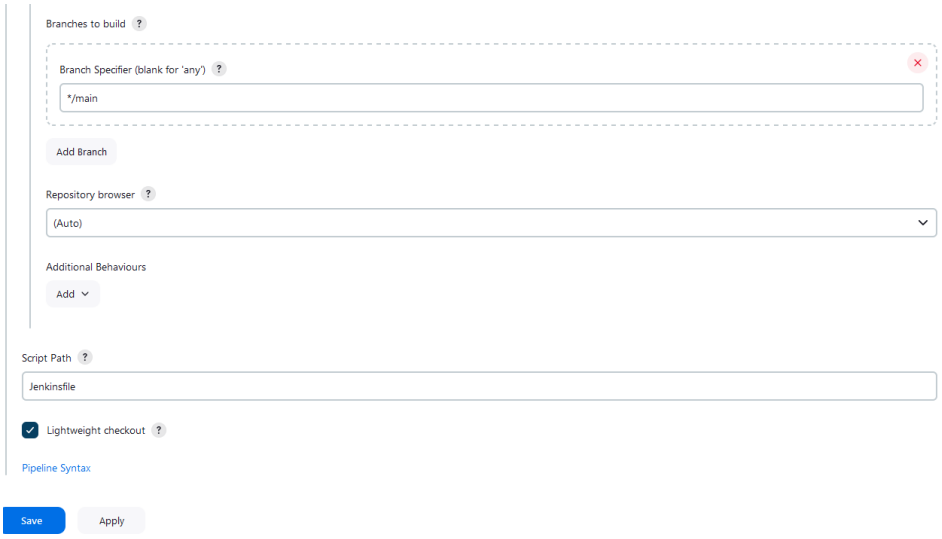
- Check nodes it will be attached to our master.

- Create a Project as pipeline project and configure it.



- Give our repo URL for declarative checkout..

- Choose Pipeline script from SCM and give our repo URL.



- Change our repository branch and Jenkins file for reference.
- Click save and apply and click build now button to start our job.