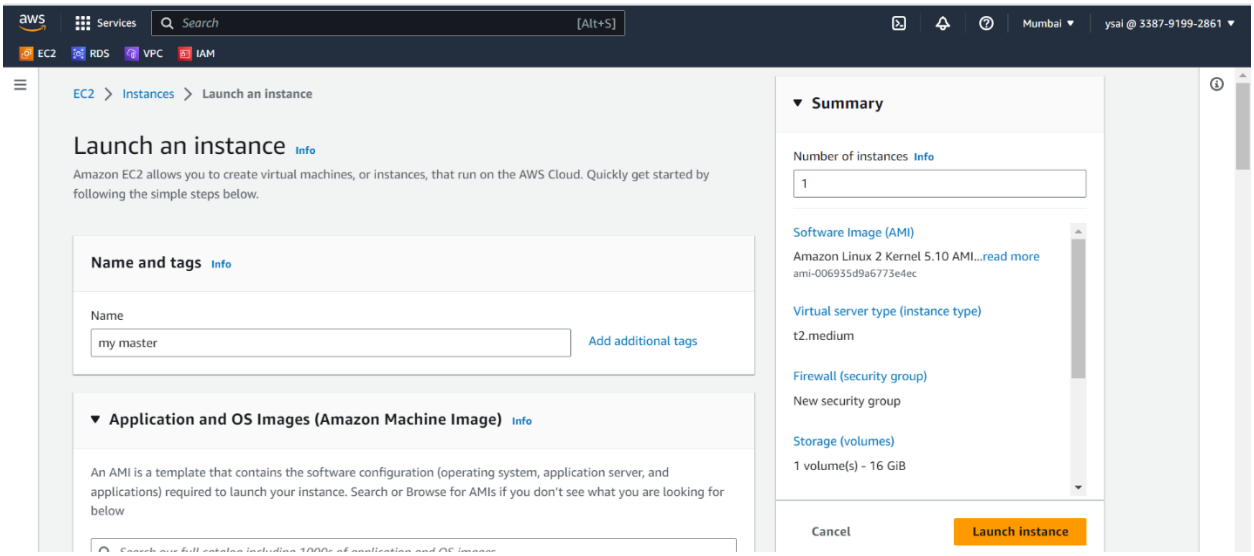
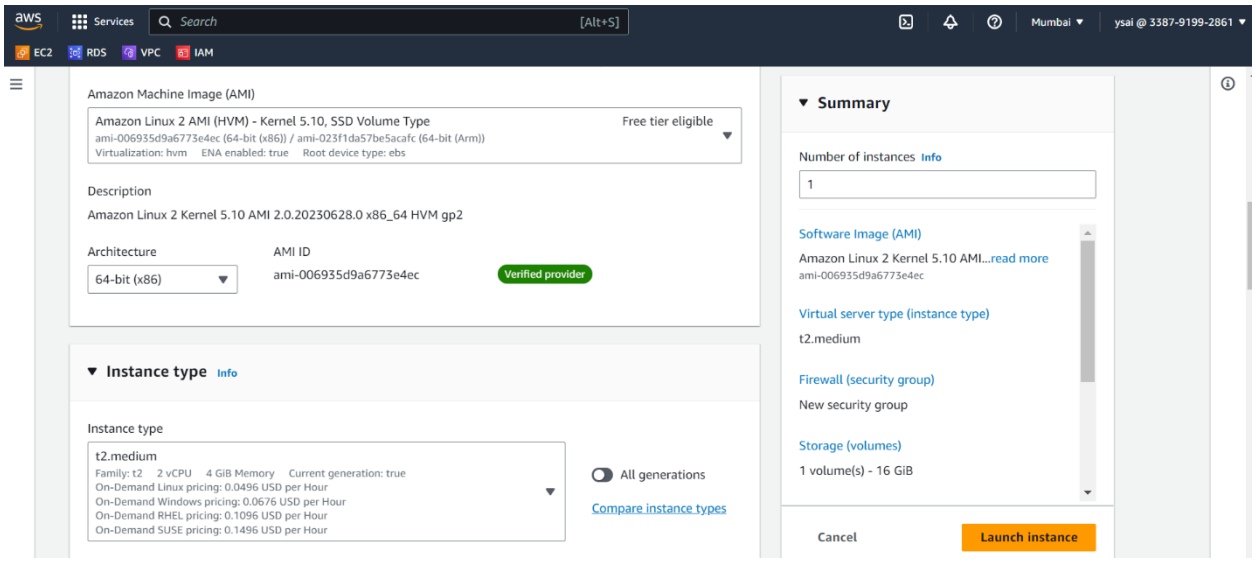
Jenkins Master Slave Configuration

- Create a EC2 instance and launch a new instance.

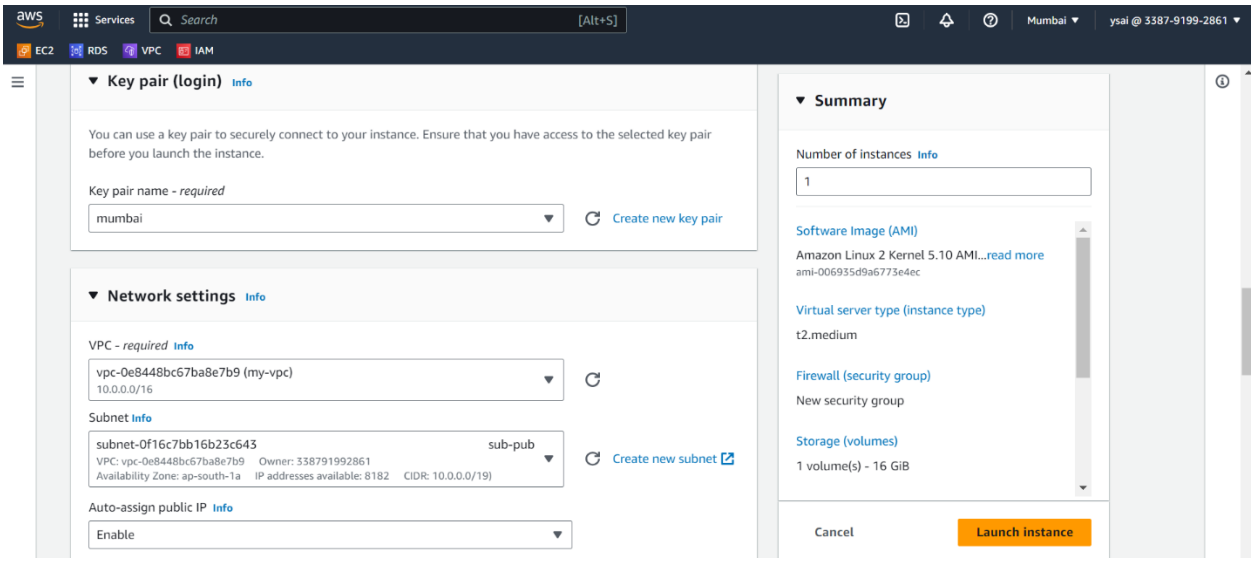


- Select Amazon Machine Image and Instance type.

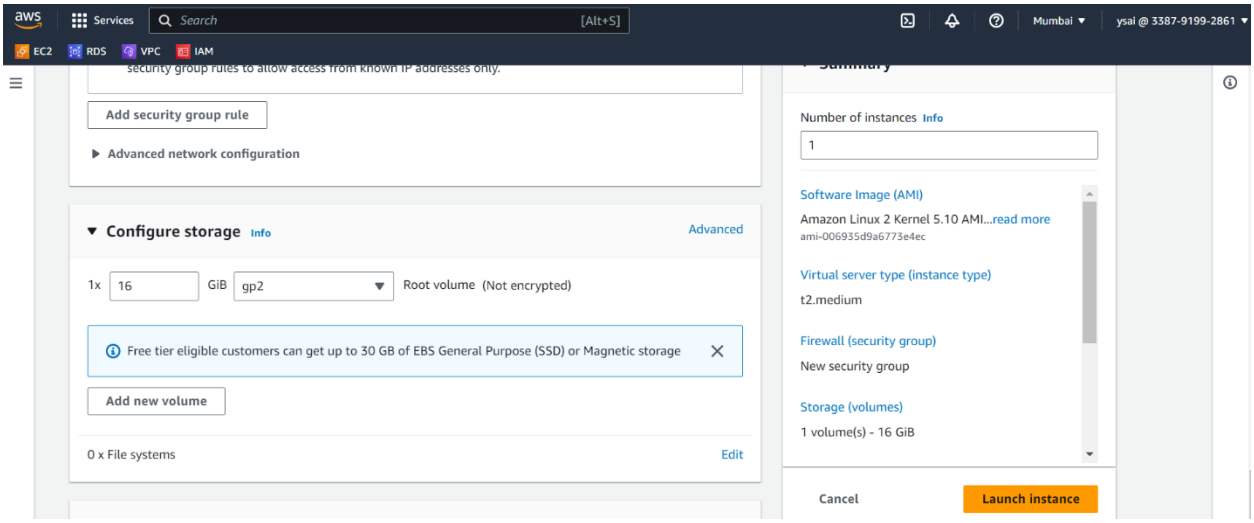


- Here select the Key pair and network Settings .

Jenkins Master Slave Configuration

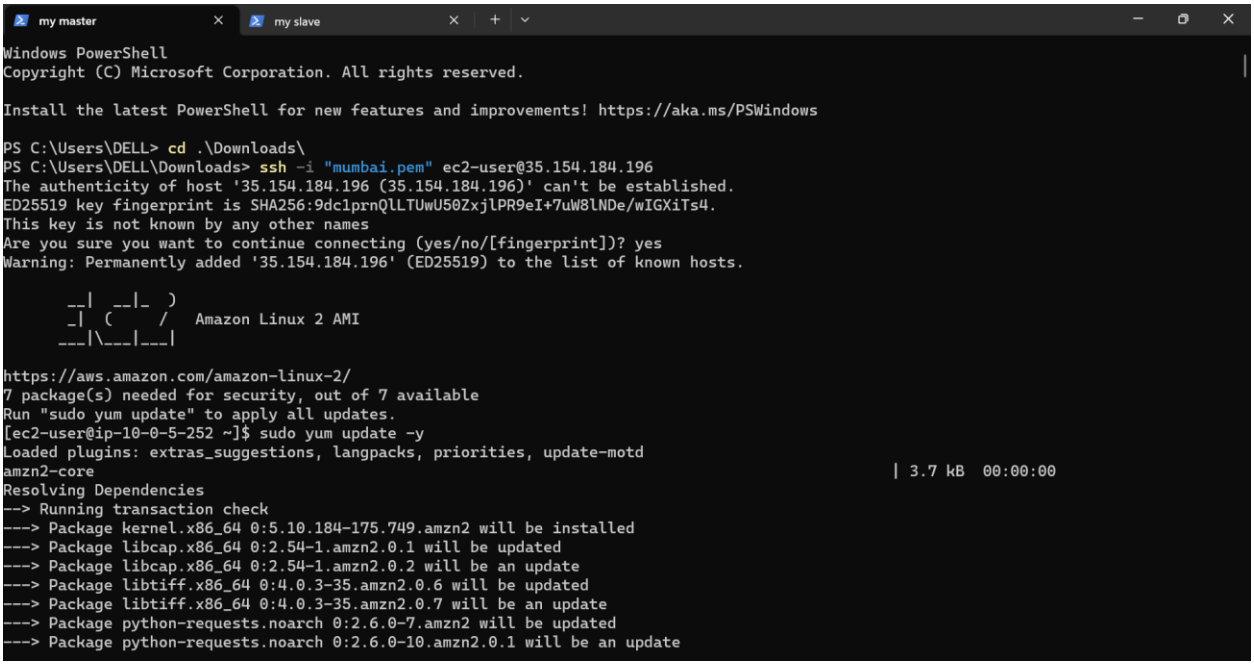


- Here we are configuring the storage to this instance, for this instance we are taking 16 GiB .

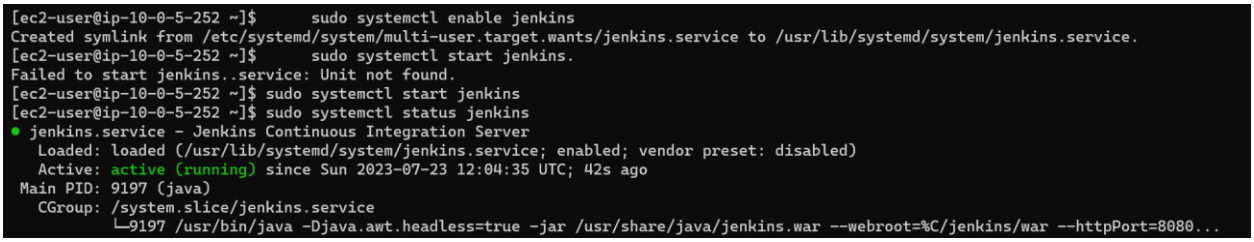


- Connect your EC2 instance to terminal By using Command “ssh -I “pem.key” ec2-user@ip.address” in master instance .

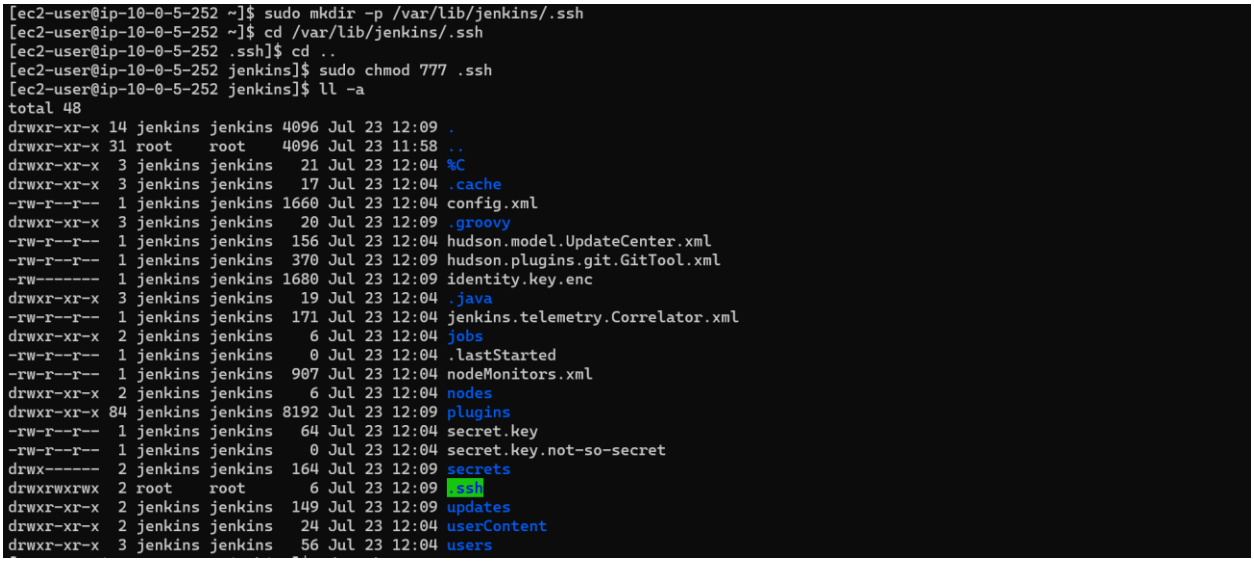
Jenkins Master Slave Configuration



- In master instance install Jenkins ,enable it and start Jenkins .



- Here we are creating directory to do configuration and in that by using CD command to go that path and then permission to .ssh as 777 permissions.



- Change directory to .ssh, in that directory we have to give the command.

  “ Ssh-keyscan -H ip.address >>/var/lib/Jenkins/.ssh/known_hosts “

Jenkins Master Slave Configuration

- Change the owner to Jenkins for known_hosts.

```
[ec2-user@ip-10-0-5-252 jenkins]$ cd .ssh
[ec2-user@ip-10-0-5-252 .ssh]$ ll -a
total 4
drwxrwxrwx  2 root    root       6 Jul 23 12:09 .
drwxr-xr-x 14 jenkins jenkins 4096 Jul 23 12:09 ..
[ec2-user@ip-10-0-5-252 .ssh]$ sudo ssh-keyscan -H 10.0.16.102 >>/var/lib/jenlins/.ssh/known_hosts
-bash: /var/lib/jenlins/.ssh/known_hosts: No such file or directory
[ec2-user@ip-10-0-5-252 .ssh]$ sudo ssh-keyscan -H 10.0.16.102 >>/var/lib/jenkins/.ssh/known_hosts
# 10.0.16.102:22 SSH-2.0-OpenSSH_7.4
# 10.0.16.102:22 SSH-2.0-OpenSSH_7.4
# 10.0.16.102:22 SSH-2.0-OpenSSH_7.4
[ec2-user@ip-10-0-5-252 .ssh]$ sudo chown jenkins:jenkins known_hosts
[ec2-user@ip-10-0-5-252 .ssh]$ ll -a
total 8
drwxrwxrwx  2 root    root      25 Jul 23 12:14 .
drwxr-xr-x 14 jenkins jenkins 4096 Jul 23 12:09 ..
-rw-rw-r--  1 jenkins jenkins  806 Jul 23 12:14 known_hosts
[ec2-user@ip-10-0-5-252 .ssh]$ sudo chmod 700 known_hosts
[ec2-user@ip-10-0-5-252 .ssh]$ ll -a
total 8
drwxrwxrwx  2 root    root      25 Jul 23 12:14 .
drwxr-xr-x 14 jenkins jenkins 4096 Jul 23 12:09 ..
-rwx------  1 jenkins jenkins  806 Jul 23 12:14 known_hosts
[ec2-user@ip-10-0-5-252 .ssh]$ cd
[ec2-user@ip-10-0-5-252 ~]$ cd /etc
[ec2-user@ip-10-0-5-252 etc]$ sudo vi sudoers
[ec2-user@ip-10-0-5-252 etc]$ cd /var/lib/jenkins/.ssh
[ec2-user@ip-10-0-5-252 .ssh]$ ls
known_hosts
```

- Before master instance we have to create slave instance . In slave instance we have to update.

```
PS C:\Users\DELL\Downloads> ssh -i "mumbai.pem" ec2-user@13.233.24.202
The authenticity of host '13.233.24.202 (13.233.24.202)' can't be established.
ED25519 key fingerprint is SHA256:AuE6dpXyPqRE7CPYaOTMq91LJT8EjCQISXc9653ktaI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.233.24.202' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-16-102 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                                                 | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package kernel.x86_64 0:5.10.184-175.749.amzn2 will be installed
---> Package libcap.x86_64 0:2.54-1.amzn2.0.1 will be updated
---> Package libcap.x86_64 0:2.54-1.amzn2.0.2 will be an update
---> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.6 will be updated
---> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.7 will be an update
---> Package python-requests.noarch 0:2.6.0-7.amzn2 will be updated
---> Package python-requests.noarch 0:2.6.0-10.amzn2.0.1 will be an update
---> Package python2-rsa.noarch 0:3.4.1-1.amzn2.0.3 will be updated
---> Package python2-rsa.noarch 0:3.4.1-1.amzn2.0.4 will be an update
---> Package python3-pip.noarch 0:20.2.2-1.amzn2.0.3 will be updated
---> Package python3-pip.noarch 0:20.2.2-1.amzn2.0.4 will be an update
---> Package tcpdump.x86_64 14:4.9.2-4.amzn2.1 will be updated
```

- In slave instance we have to create user and password to that user.

```
[ec2-user@ip-10-0-16-102 ~]$ sudo useradd myslave
[ec2-user@ip-10-0-16-102 ~]$ sudo passwd myslave
Changing password for user myslave.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-16-102 ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@ip-10-0-16-102 ~]$ sudo sevice sshd restart
sudo: sevice: command not found
[ec2-user@ip-10-0-16-102 ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[ec2-user@ip-10-0-16-102 ~]$ exit
logout
```

- To login into user instance we have to use this command.

Jenkins Master Slave Configuration

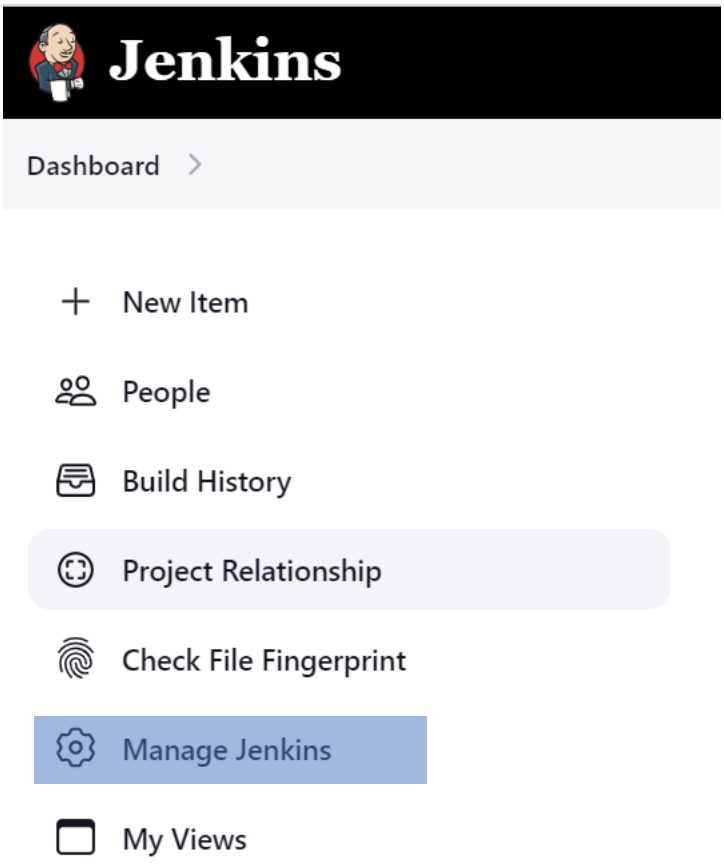
Command : ssh [username @ip.address](#) this is the command to login.

```
PS C:\Users\DELL\Downloads> ssh myslave@13.233.24.202
myslave@13.233.24.202's password:

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
```

- Now generating public/private rsa key pair with help of command "**ssh-keygen**" and cd .ssh
- cat id_rsa.pub > authorized_keys
- chmod 700 authorized_keys .

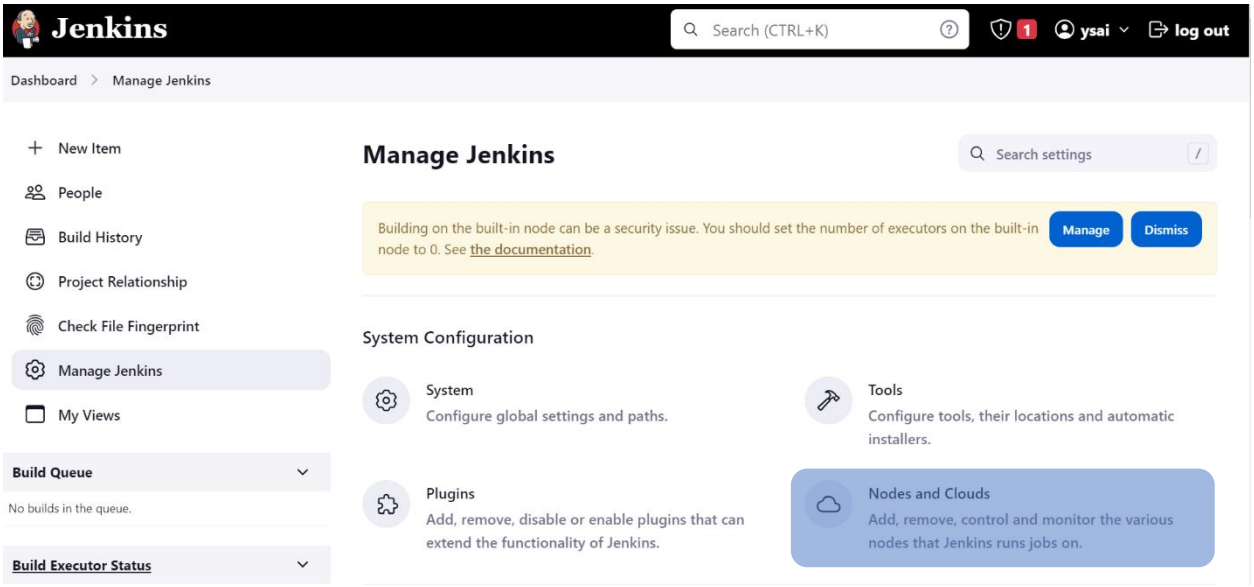
```
[myslave@ip-10-0-16-102 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/myslave/.ssh/id_rsa):
Created directory '/home/myslave/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/myslave/.ssh/id_rsa.
Your public key has been saved in /home/myslave/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:IGYkIIwdrHEY5IxM03+tT5IsMDIVreaxdmY7UA4GFCg myslave@ip-10-0-16-102.ap-south-1.compute.internal
The key's randomart image is:
+---[RSA 2048]----+
|X%=o+            |
|Eo== .           |
|oBo * . .        |
|.o @ + o .       |
|  * O o S        |
|   = * = .       |
|  . = o +        |
|     o   .       |
|      .          |
+----[SHA256]-----+
```

- Login to Jenkins console and select manage Jenkins .

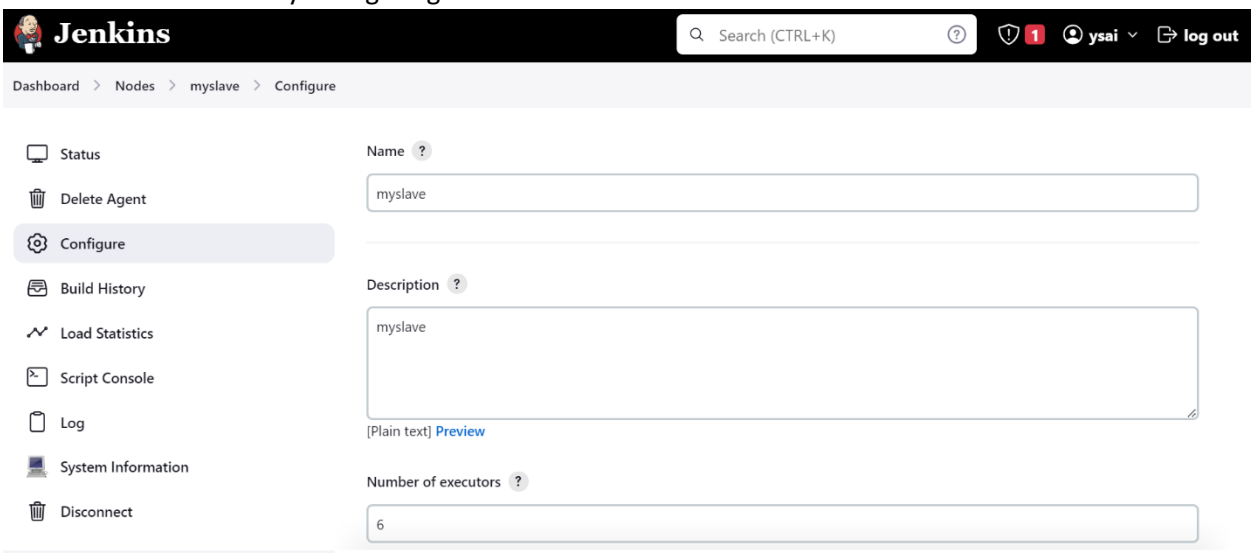


Jenkins Master Slave Configuration

- In Manage Jenkins go to Nodes and Cloud.

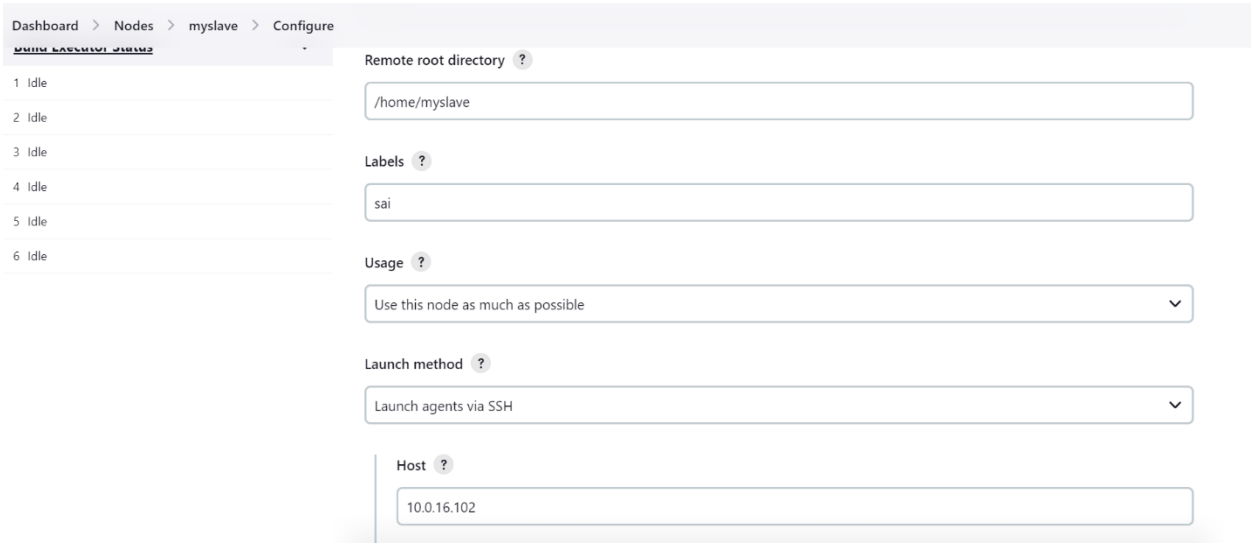


- Create a node by configuring as like below

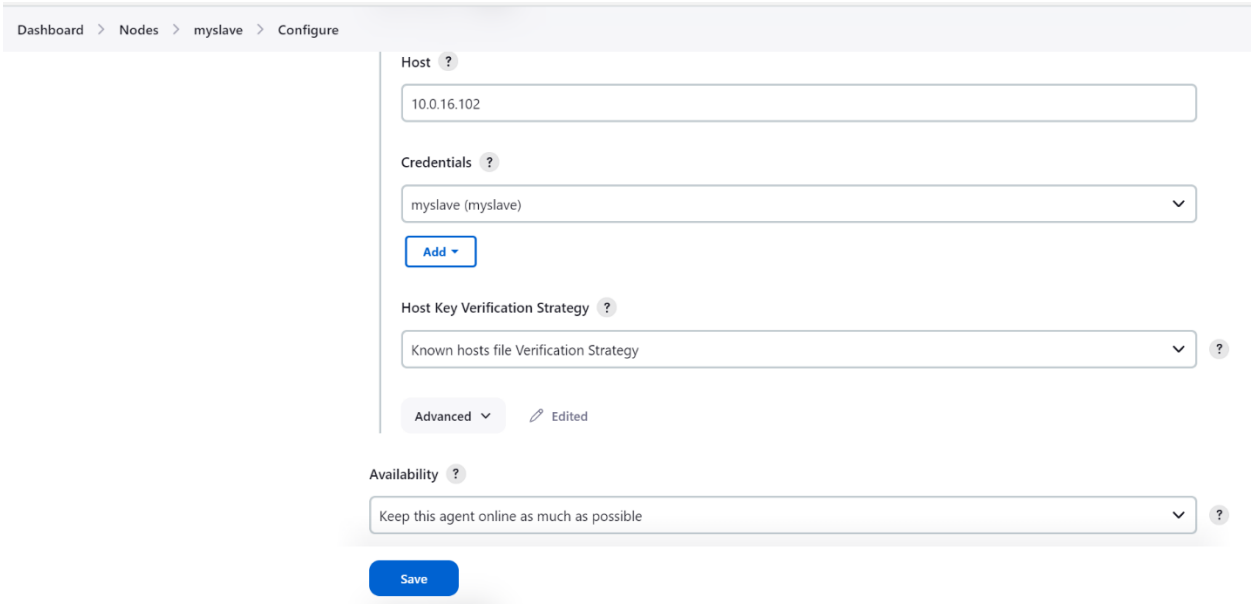


Jenkins Master Slave Configuration

- Give the remote root directory and label and we have to select the launch method.

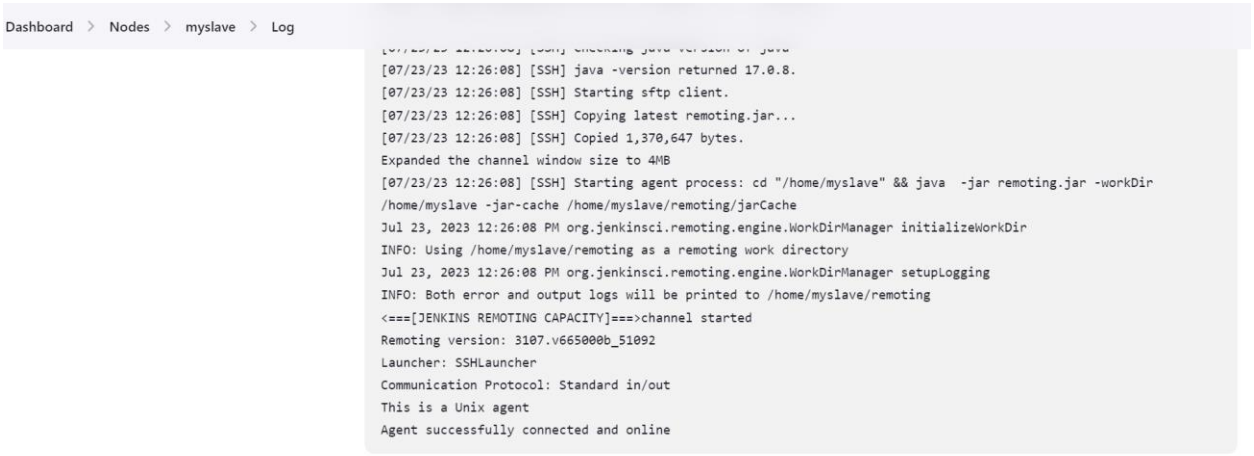


- Here we selecting the host id of the slave and host key verication as a verification strategy and save the congigure.

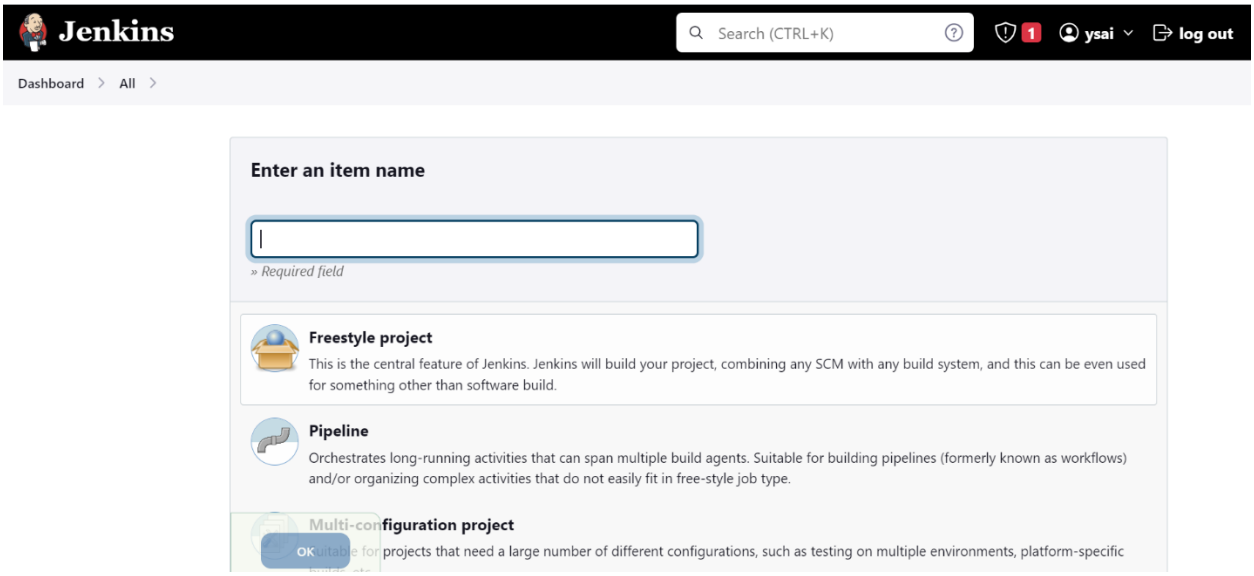


Jenkins Master Slave Configuration

- After saving the configure go and check in the log console weather the agent is successful or not.

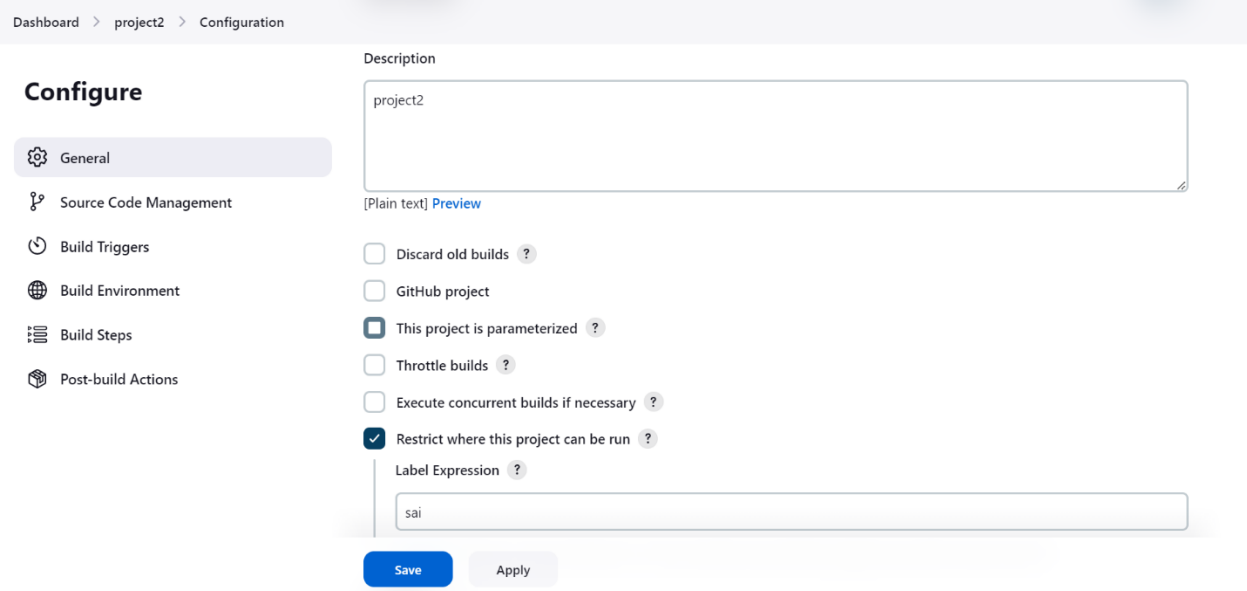


- Now, create a job and select the free style project to do config and automation.

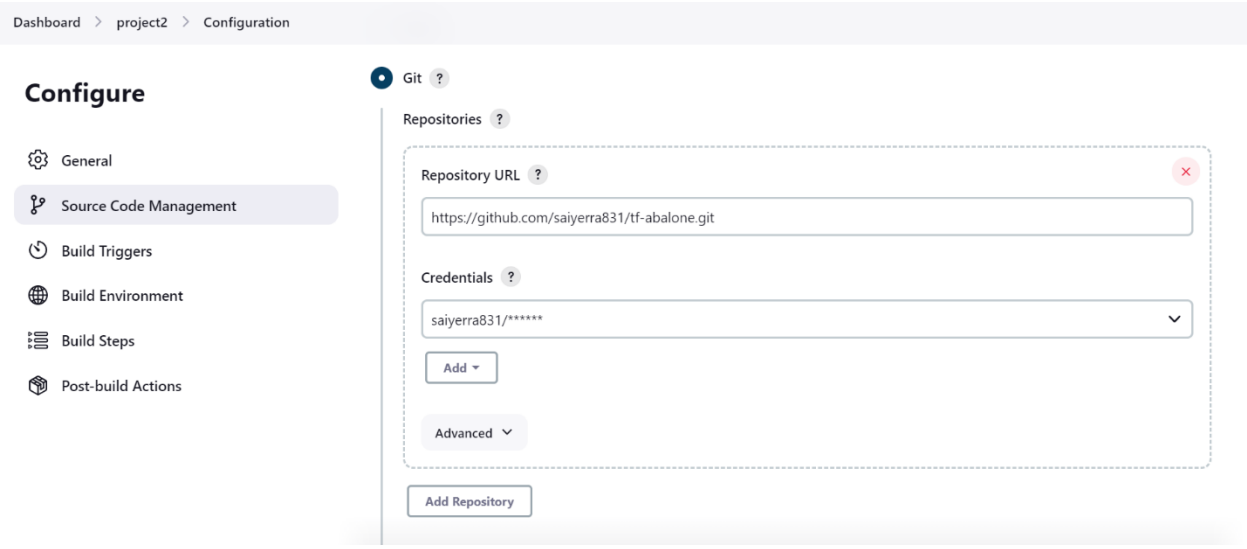


- In configuration give your description of the project what you are going to do and select the label expression.

Jenkins Master Slave Configuration

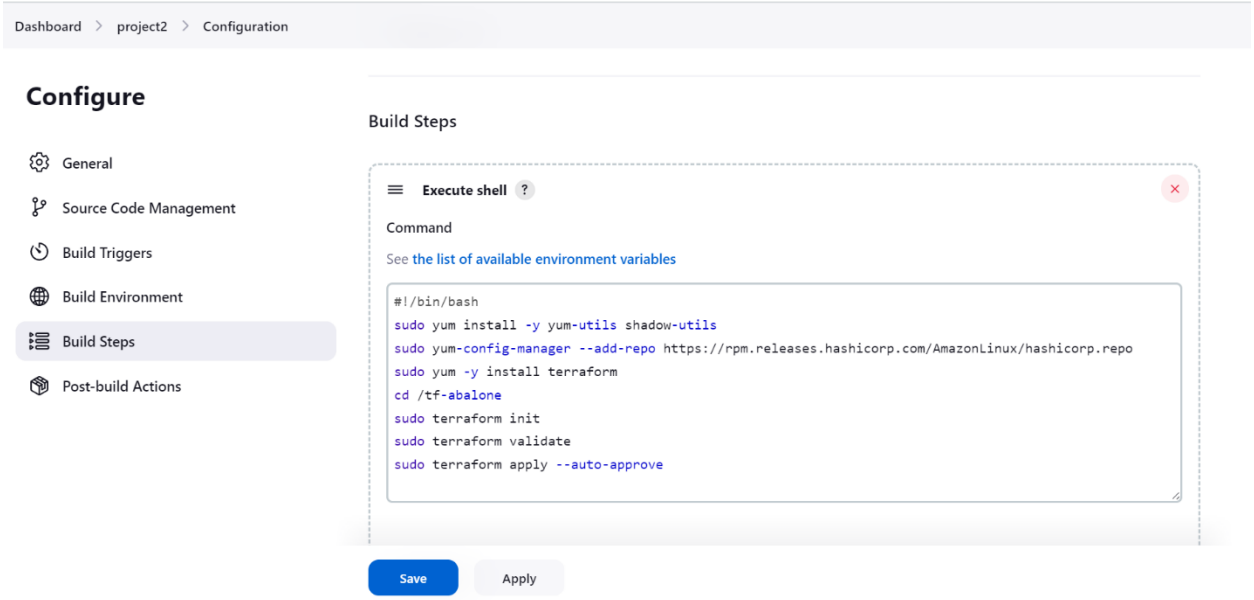


- Select the git hub repository URL with credentials (it requires only when it is private repository).

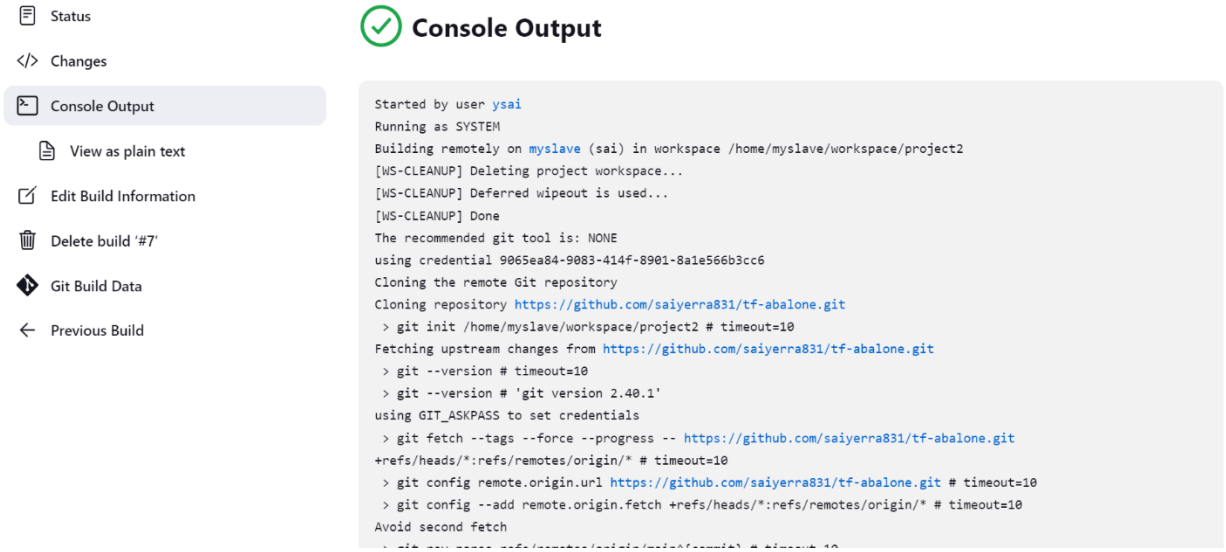


- In build steps give the related commands of your project.

Jenkins Master Slave Configuration

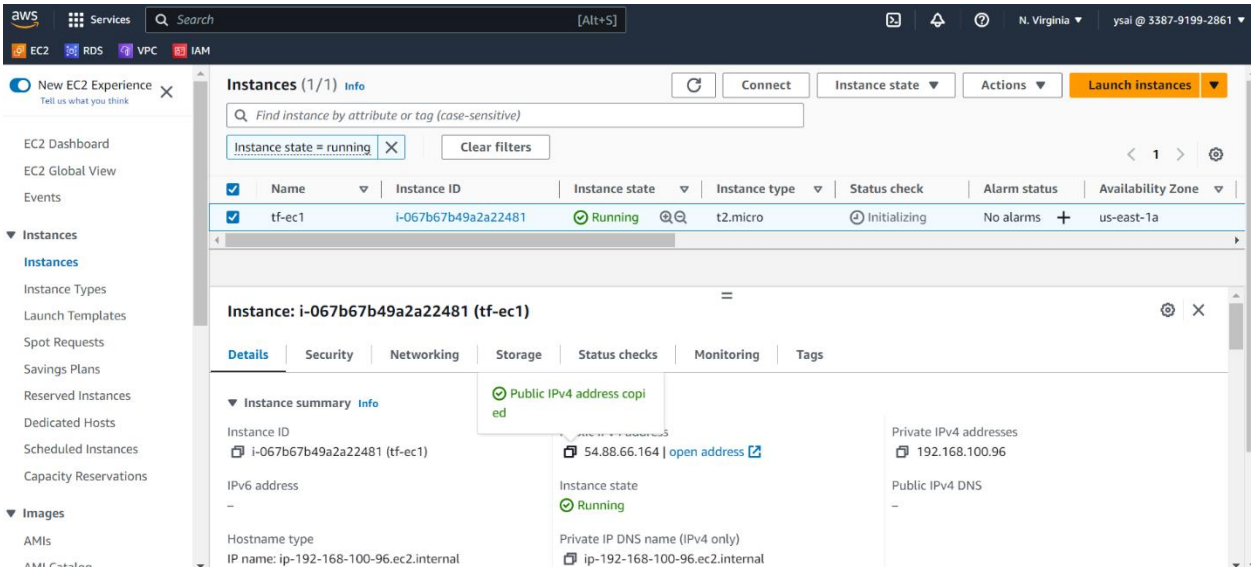


- Here do apply and save the configuration and do BUILD NOW then see the console output.

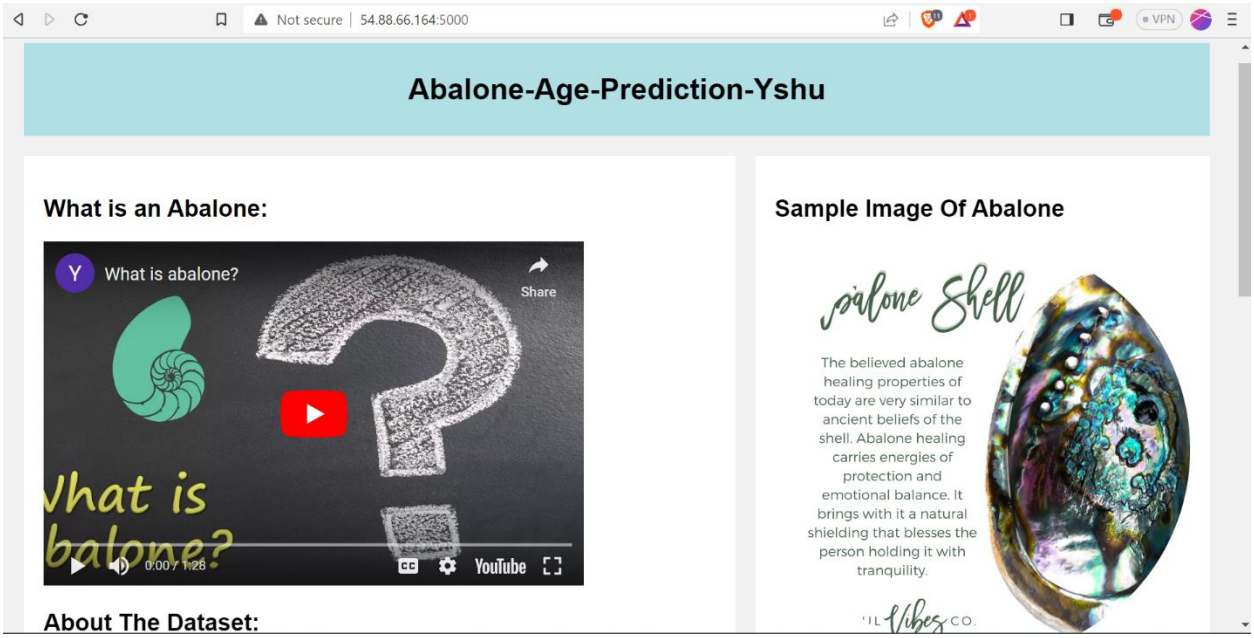


- Go to EC2 console and select the instance and copy the instance Ip address.

Jenkins Master Slave Configuration



- Here is the output of the abalone age prediction by using master and slave configuration.



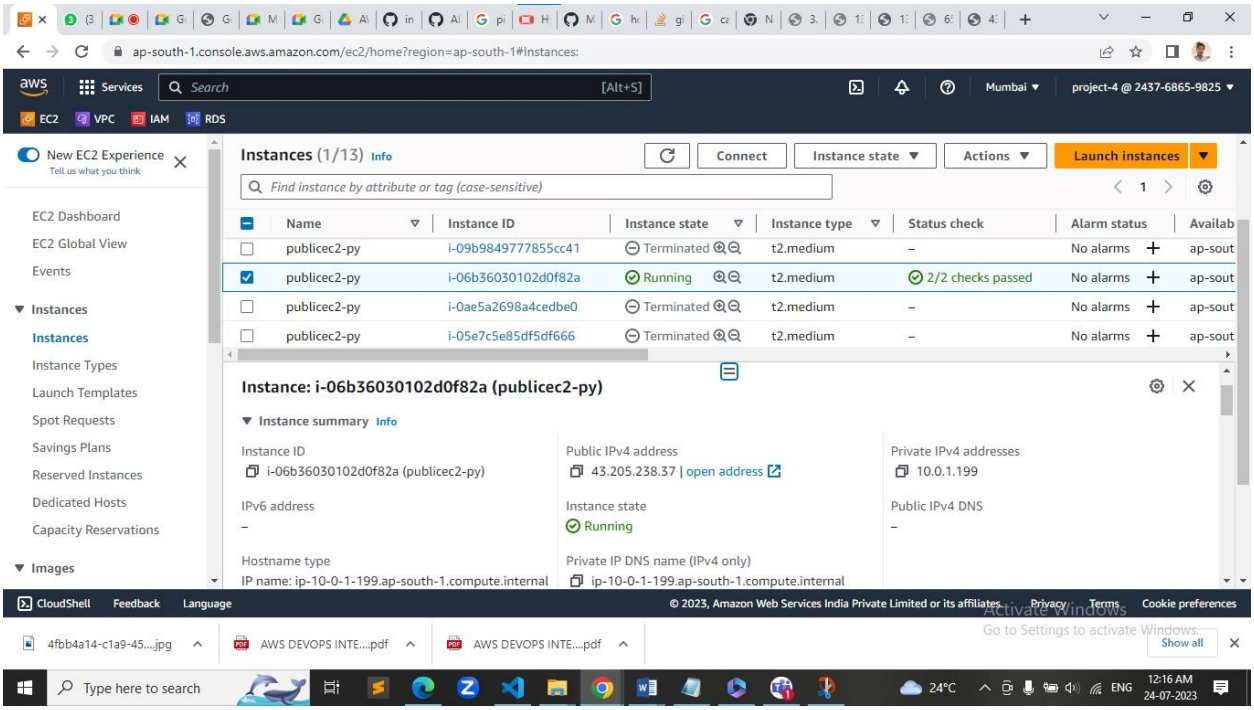
Master and slave configuration using different accounts, Here I am the master instance and there are five members in my team, they did there task as like slave instance.

1. Satya's slave configuration
2. Sangita slave configuration
3. Ajlal slave configuration
4. Jagadeesh slave configuration
5. John slave configuration.

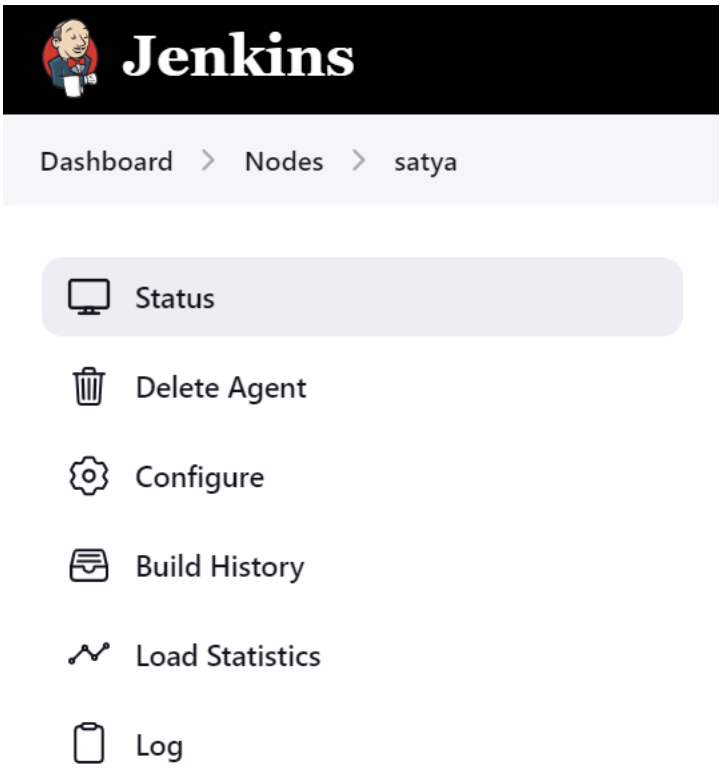
Satya's configuration :

Jenkins Master Slave Configuration

- In this satya created one ec2 instance and created one user in that instance and shared to me.

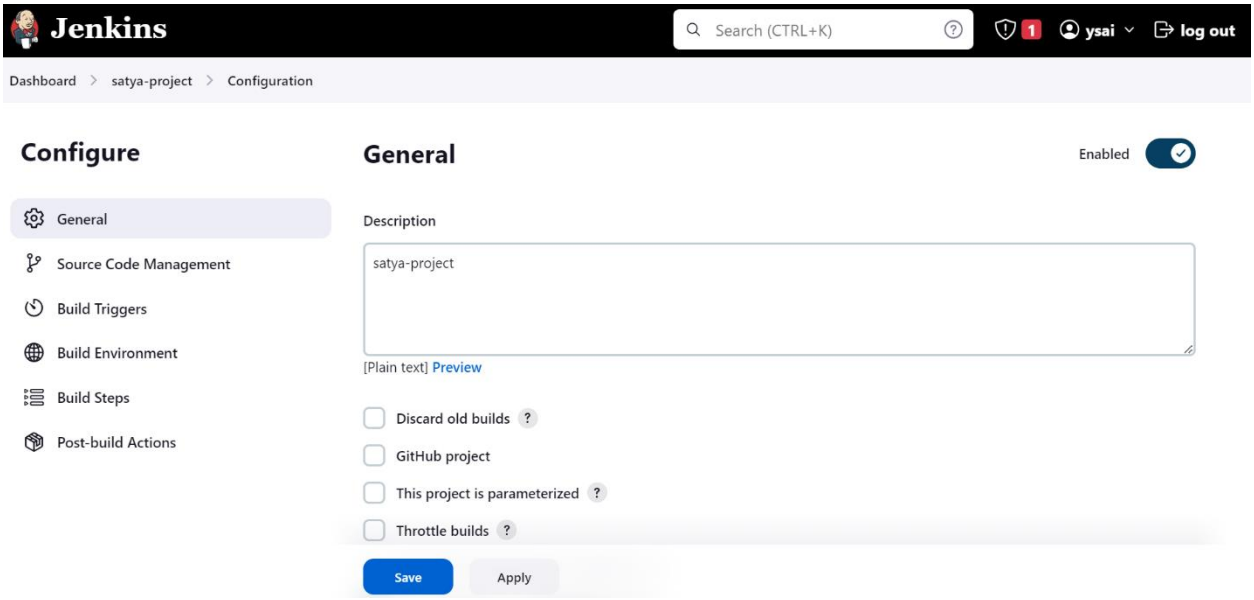


- Here I added satya slave in my manage Jenkins = Nodes.

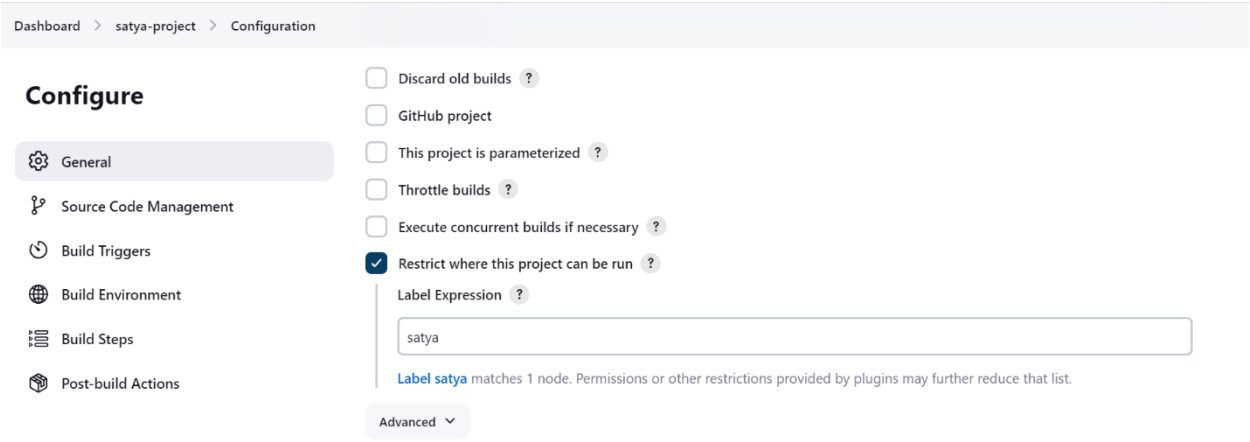


Jenkins Master Slave Configuration

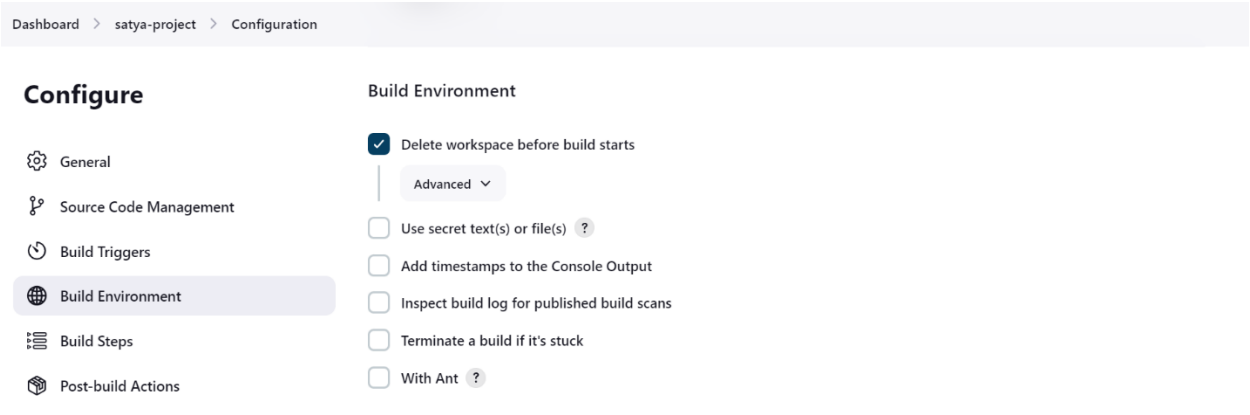
- Created a new project by using satya slave as a satya-project



- Here I mentioned in my configuration where to run this project. I gave Satya's slave label to identify the user.

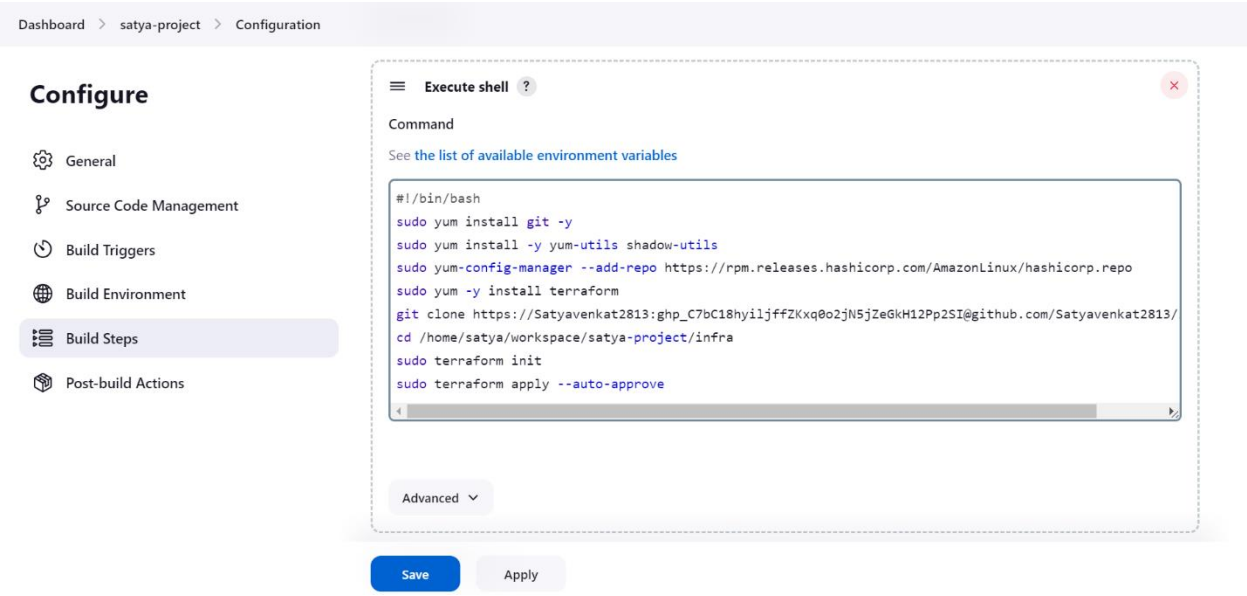


- In build environment I mention that delete workspace creating the project.

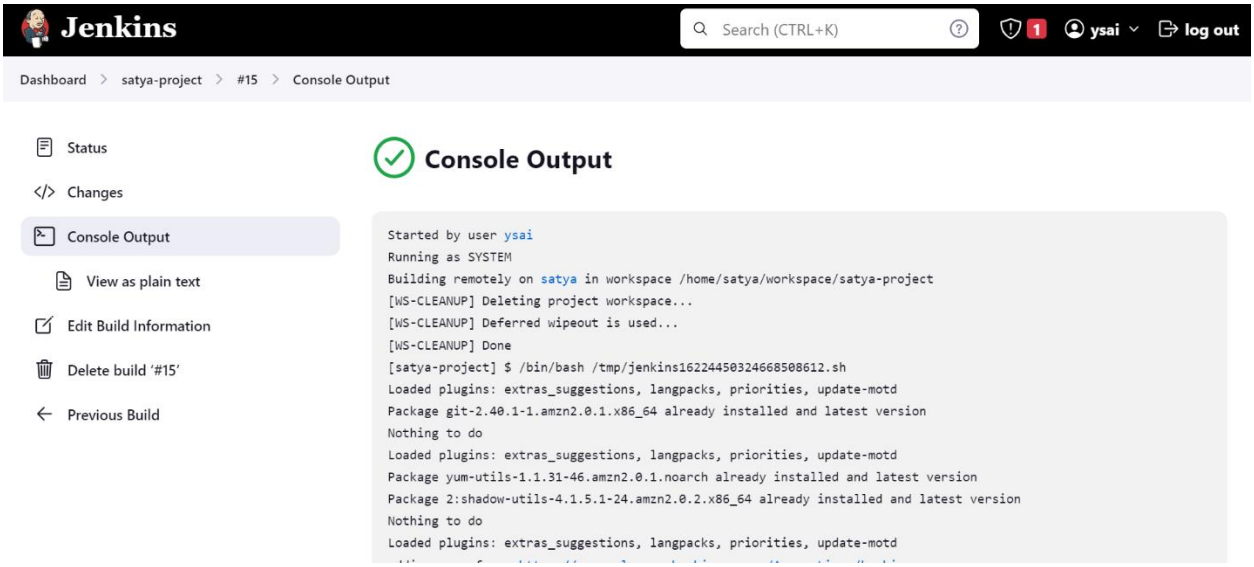


Jenkins Master Slave Configuration

- Here I gave commands in execute shell to do automation.

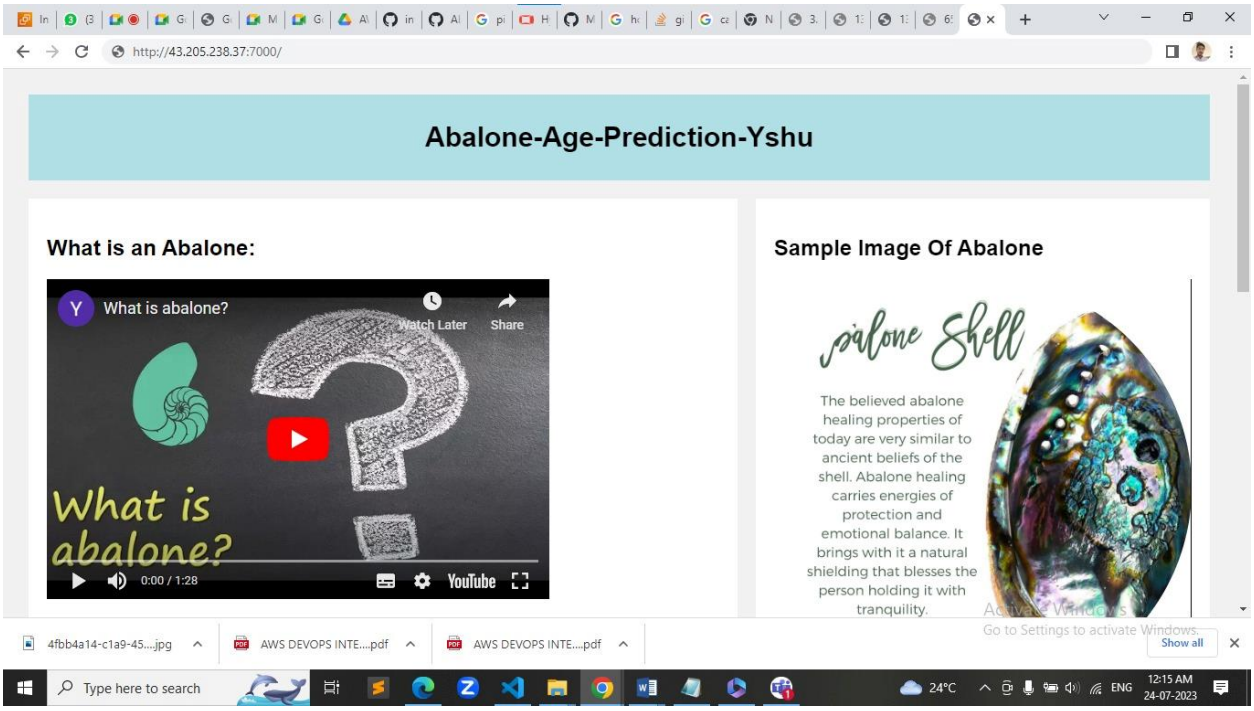


- After configuring I made build now and checked weather the console output is successful or not.



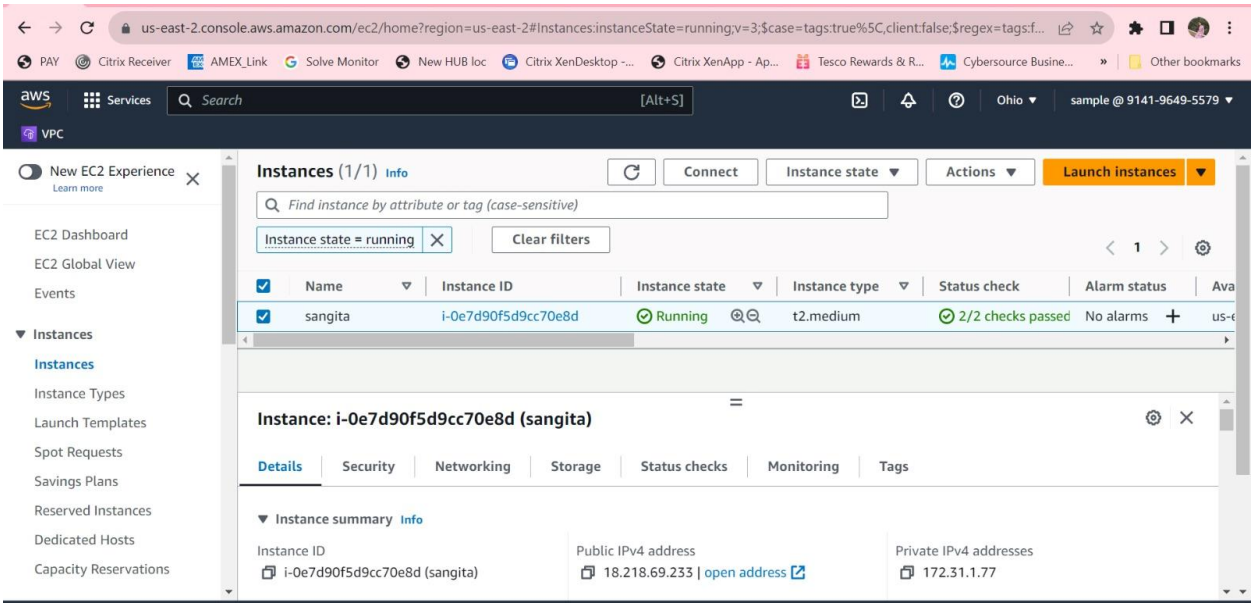
Output :

Jenkins Master Slave Configuration



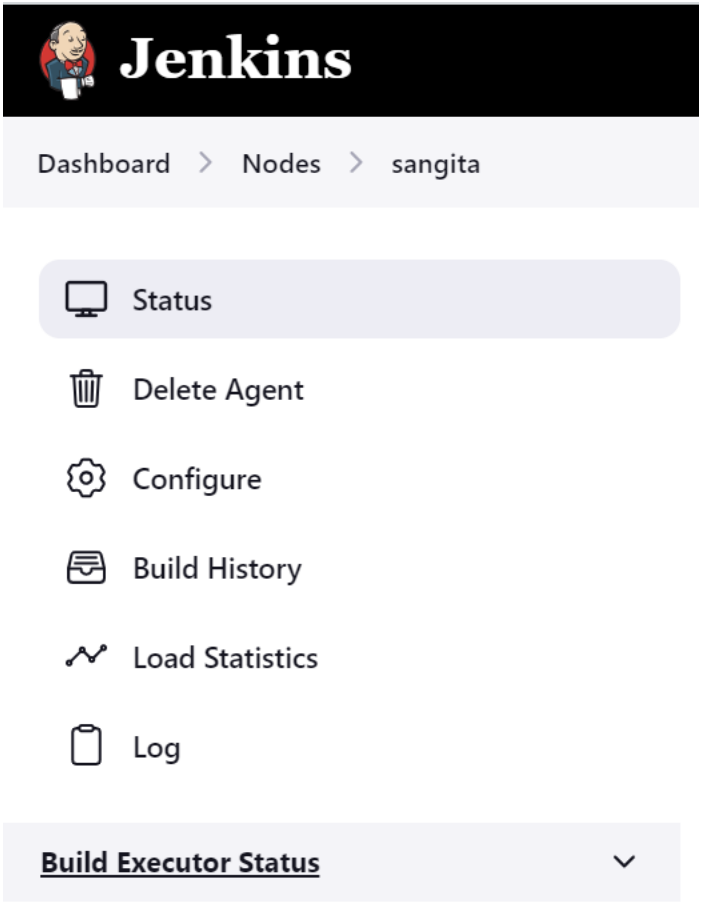
Sangita's slave configuration :

- In this sangita created one ec2 instance and created one user in that instance and shared to me

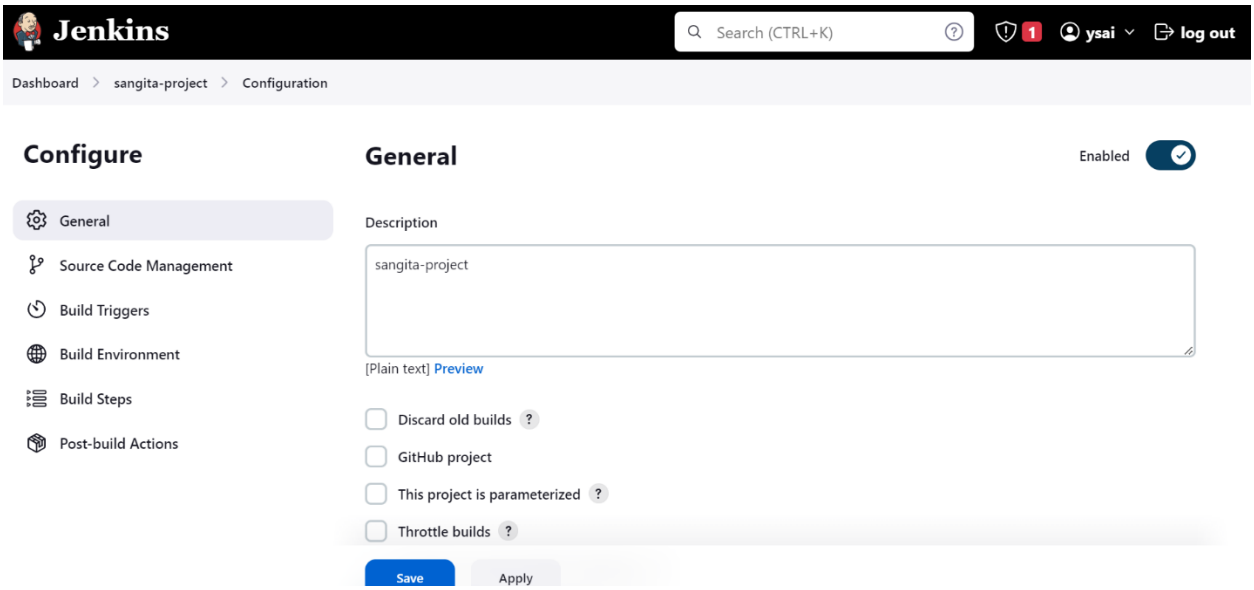


Jenkins Master Slave Configuration

- Here I added Sangita slave in my manage Jenkins = Nodes.

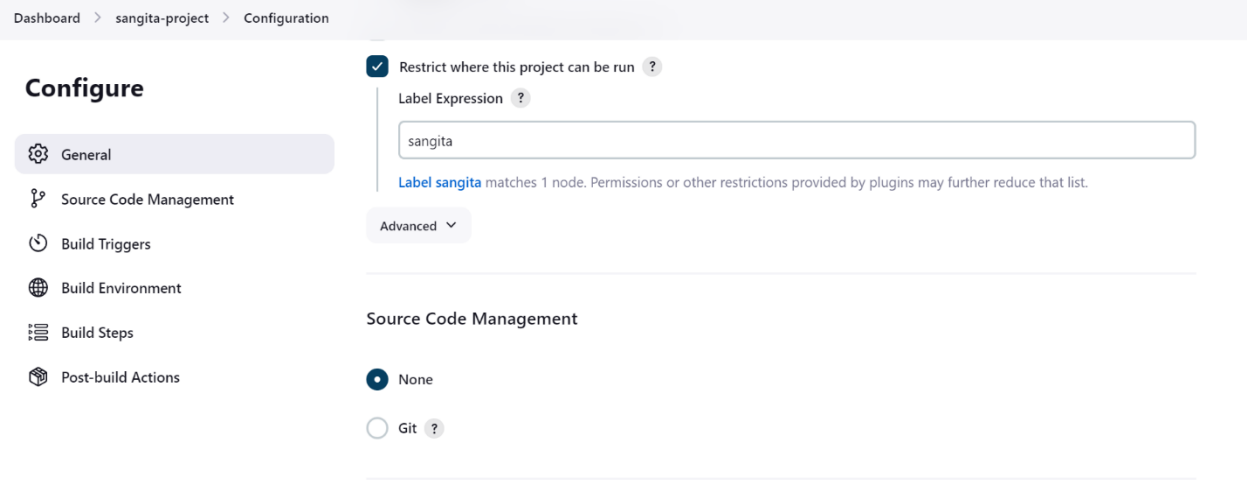


- Created a new project by using Sangita slave as a Sangita-project.

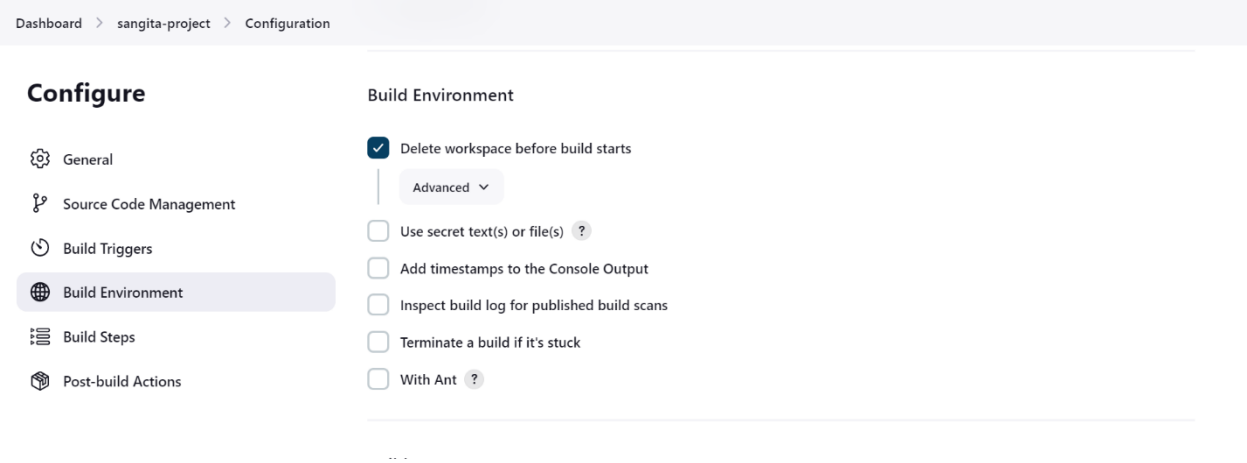


Jenkins Master Slave Configuration

- Here I mentioned in my configuration where to run this project. I gave Sangita's slave label to identify the user.

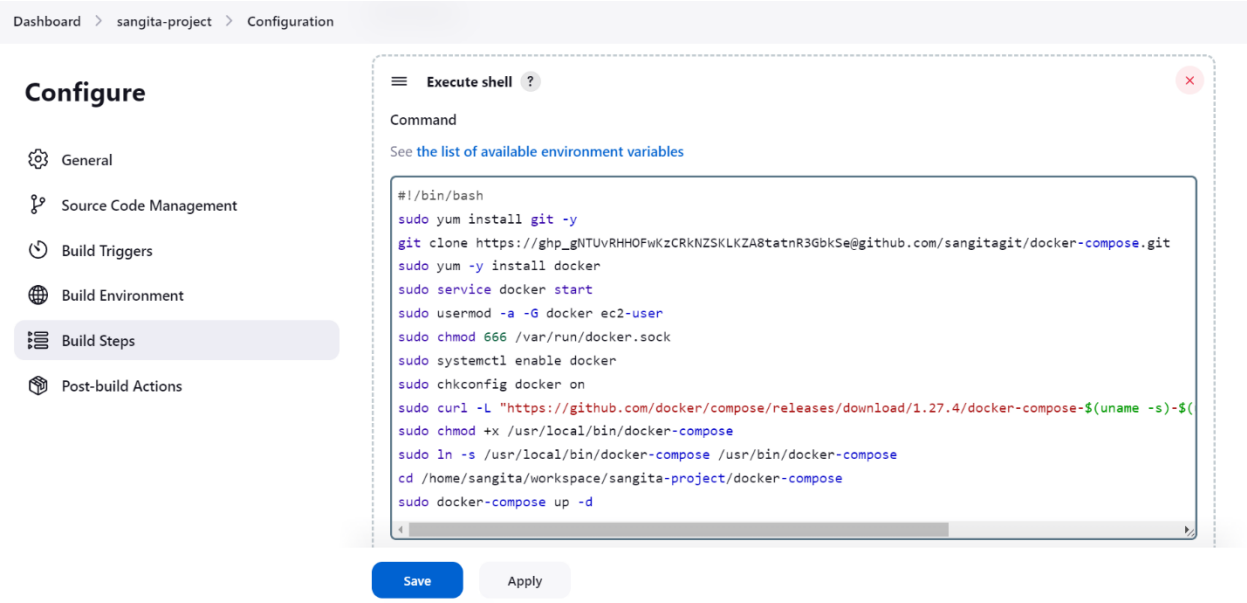


- In build environment I mention that delete workspace creating the project.

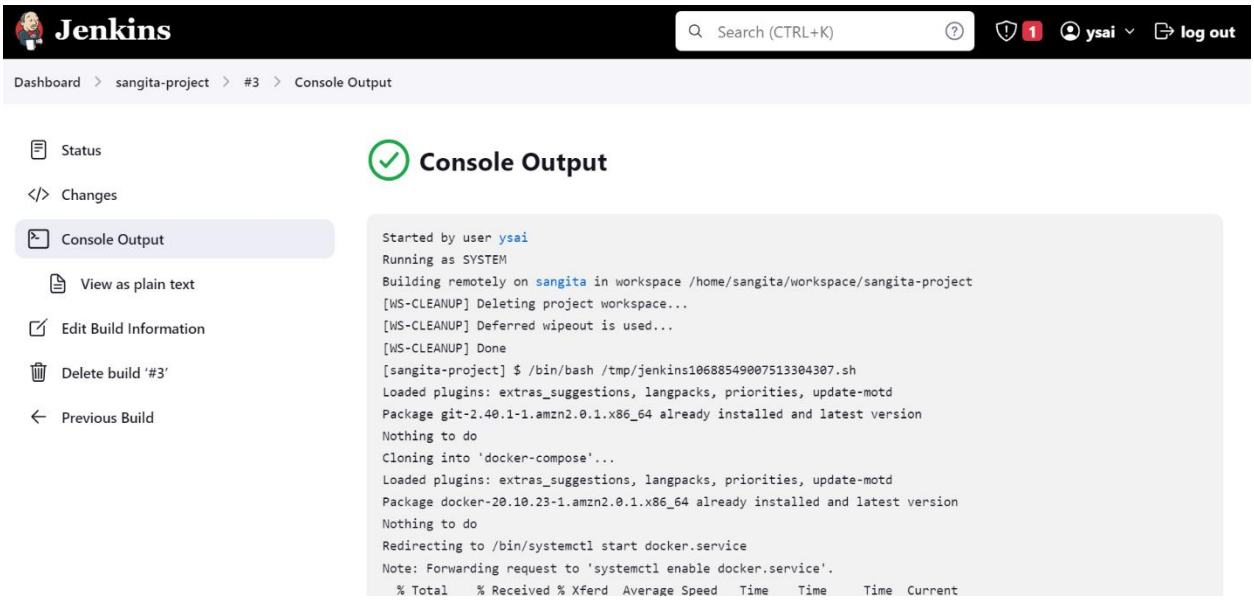


- Here I gave commands in execute shell to do automation.

Jenkins Master Slave Configuration

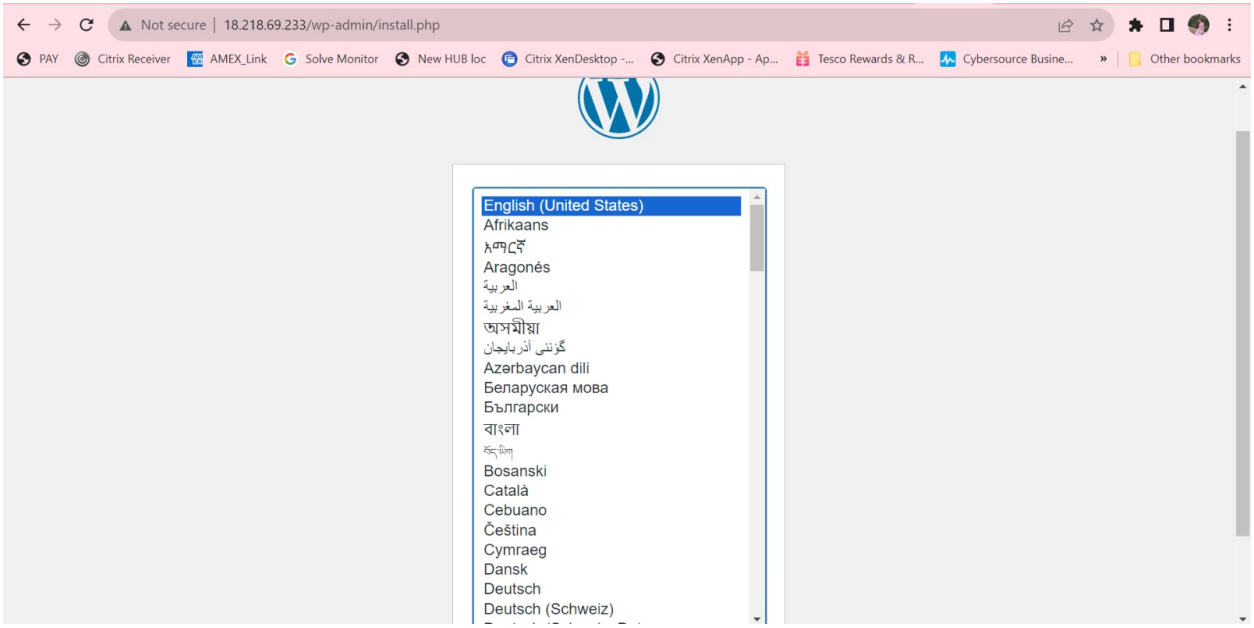


- After configuring I made build now and checked weather the console output is successful or not.



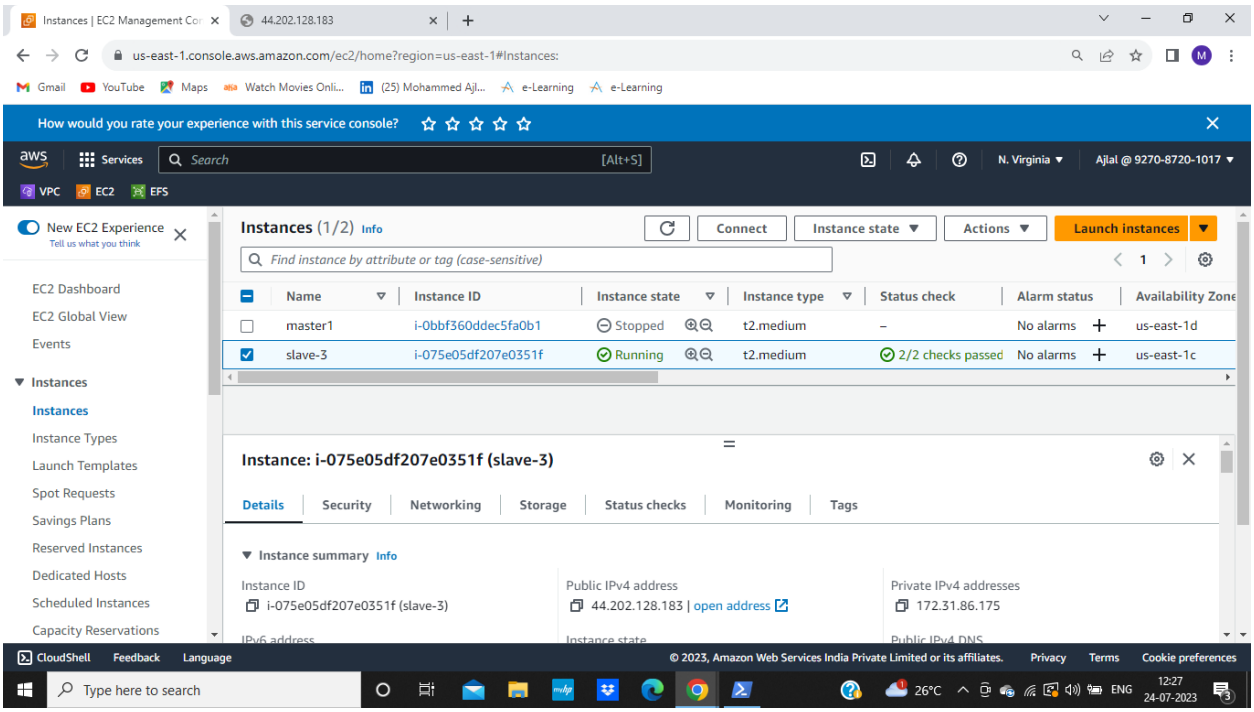
Output :

Jenkins Master Slave Configuration



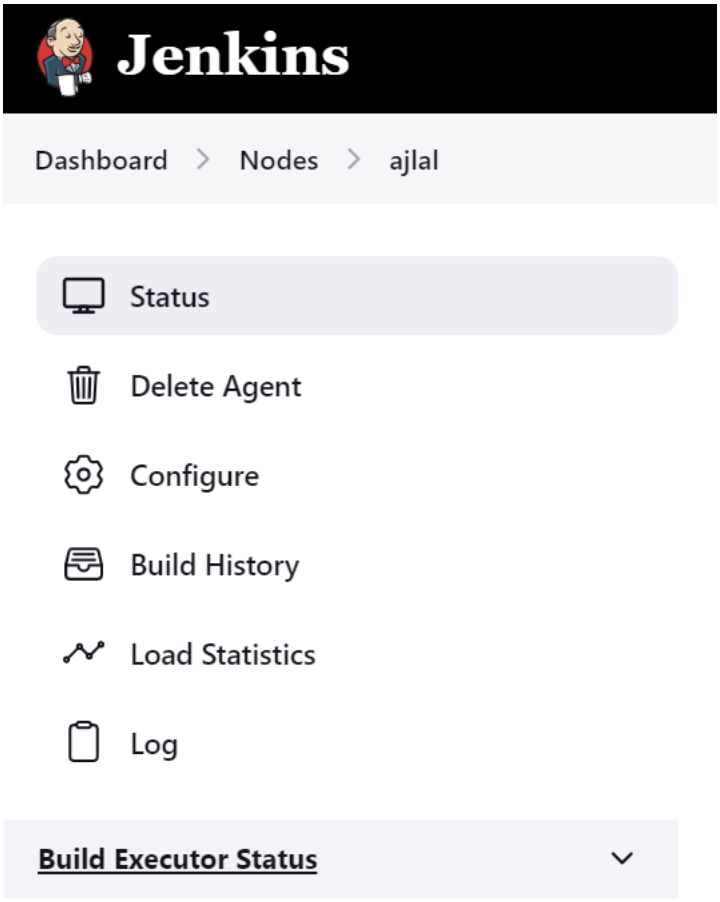
Ajlal slave configuration :

- In this Ajlal created one ec2 instance and created one user in that instance and shared to me

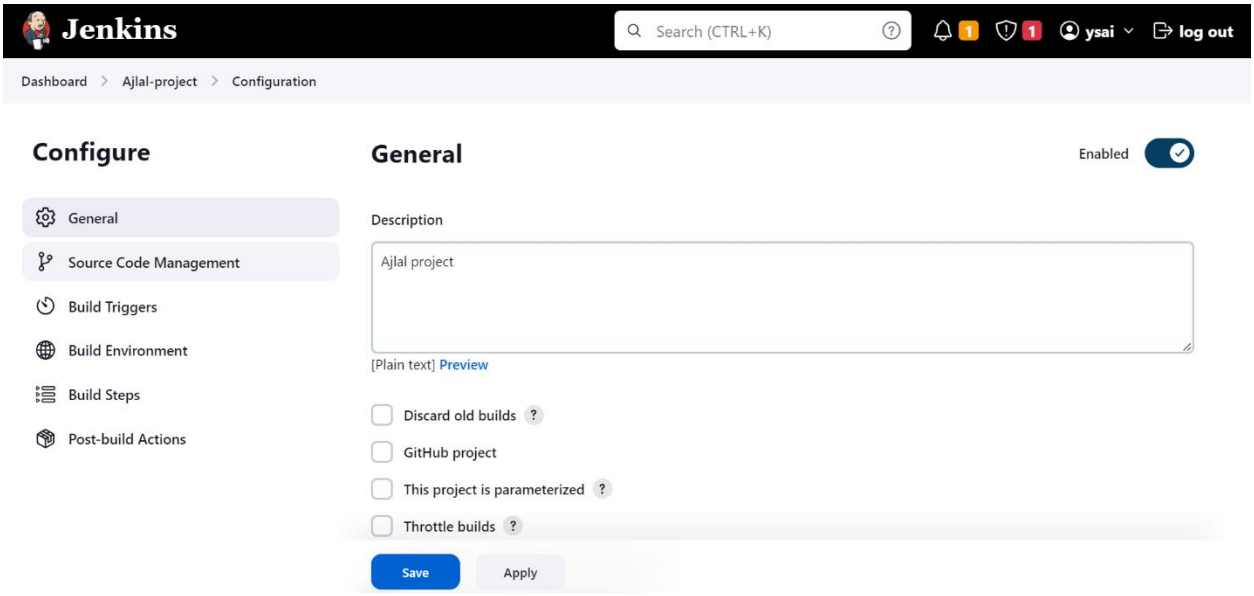


- Here I added Ajlal slave in my manage Jenkins = Nodes.

Jenkins Master Slave Configuration

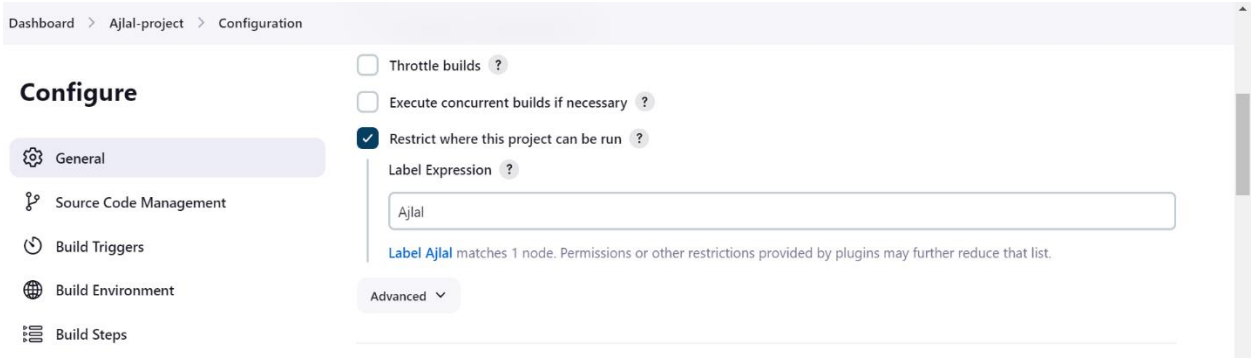


- Created a new project by using Ajlal slave as a Ajlal-project.

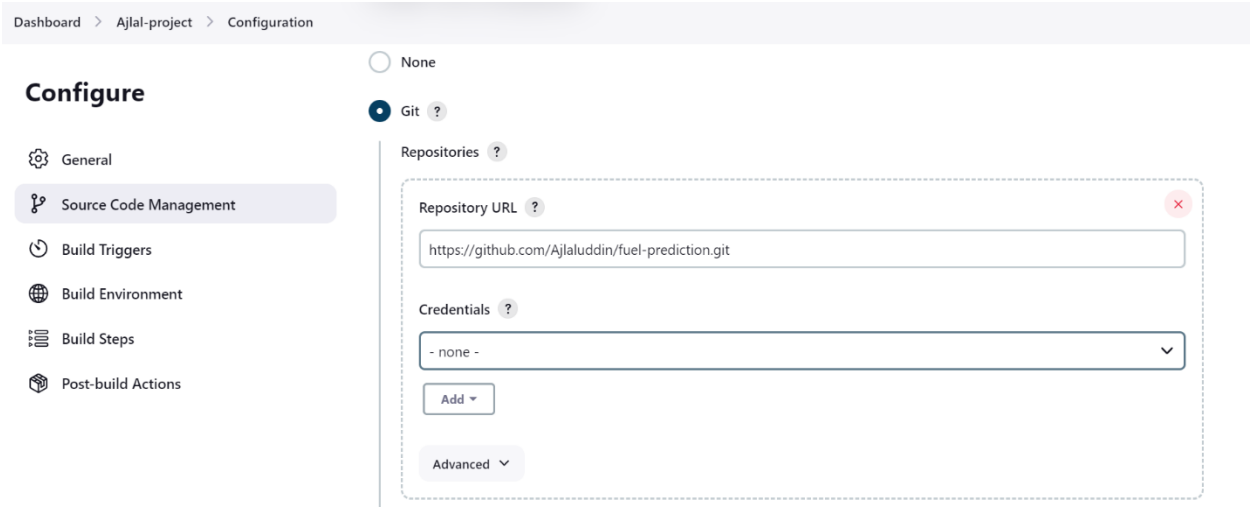


Jenkins Master Slave Configuration

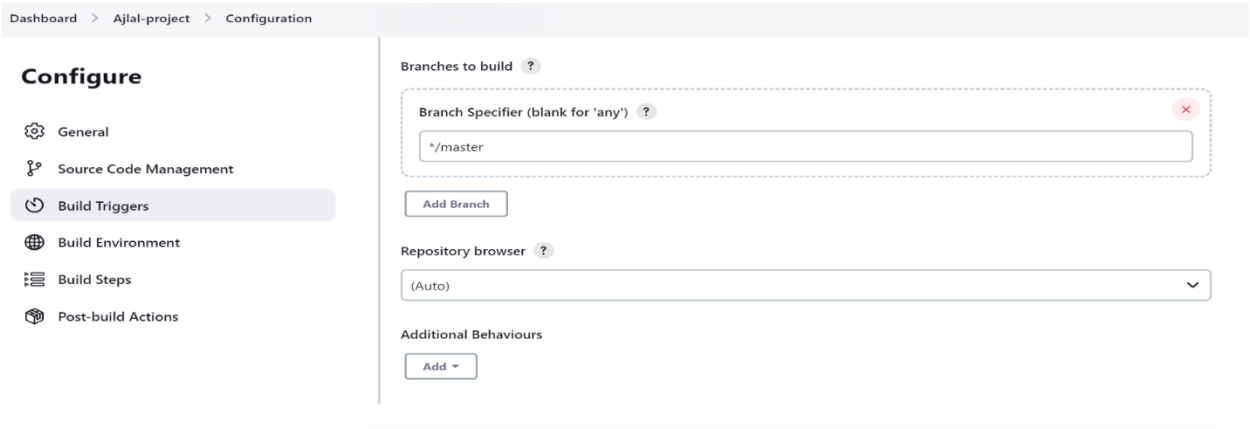
- Here I mentioned in my configuration where to run this project. I gave Ajlal's slave label to identify the user.



- Here I gave git repository to clone into that repo

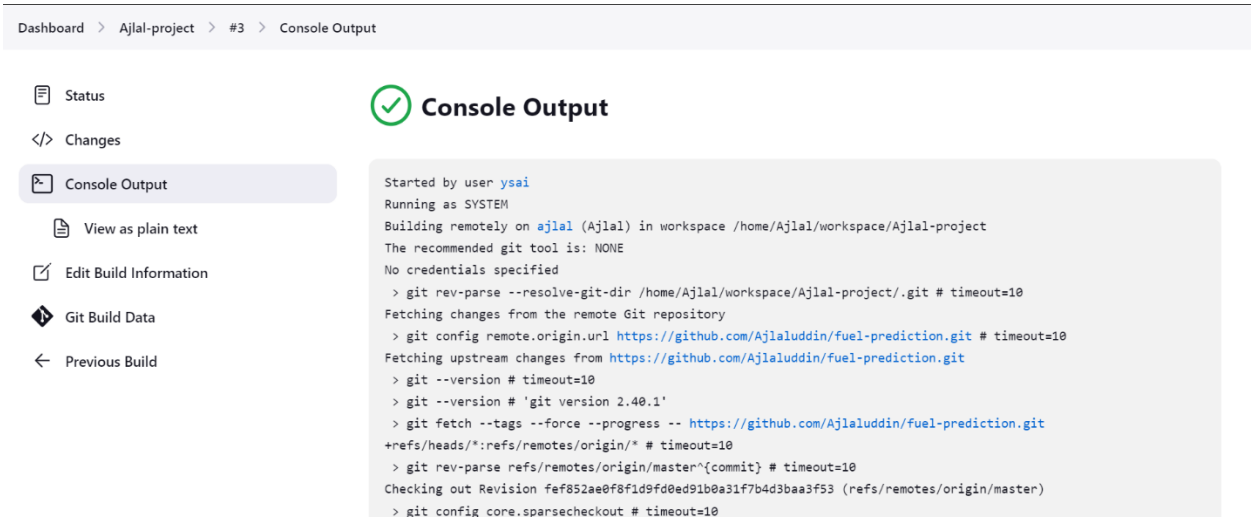


- Now I am mentioning the branch specifier

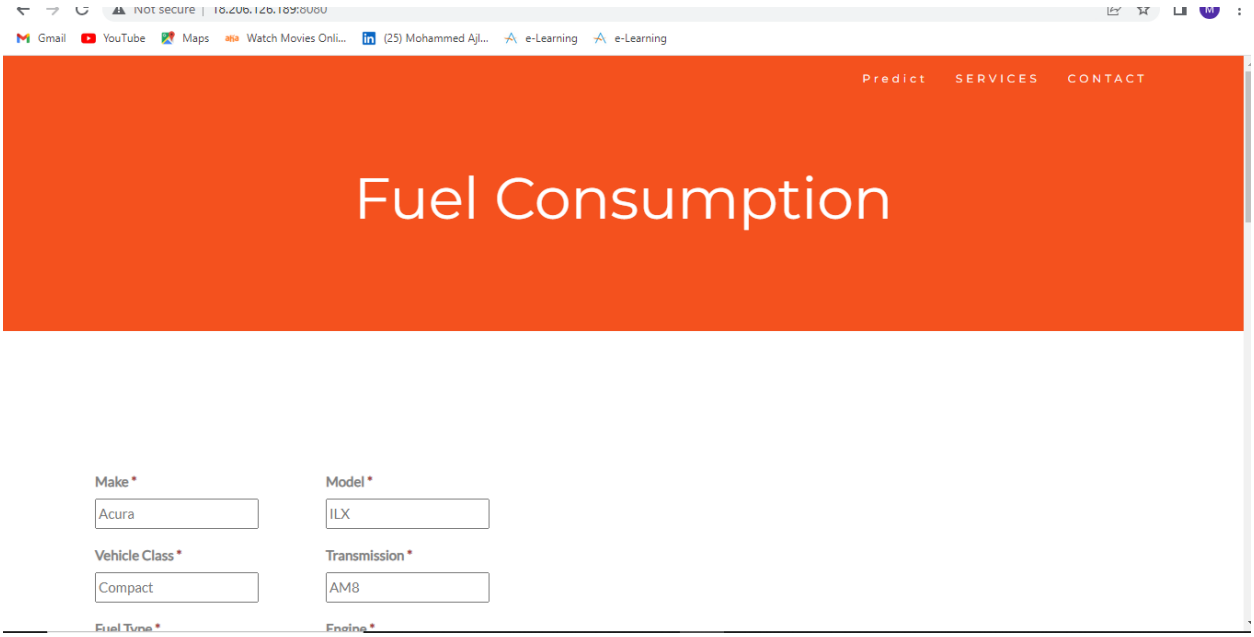


Jenkins Master Slave Configuration

- After configuring I made build now and checked weather the console output is successful or not.



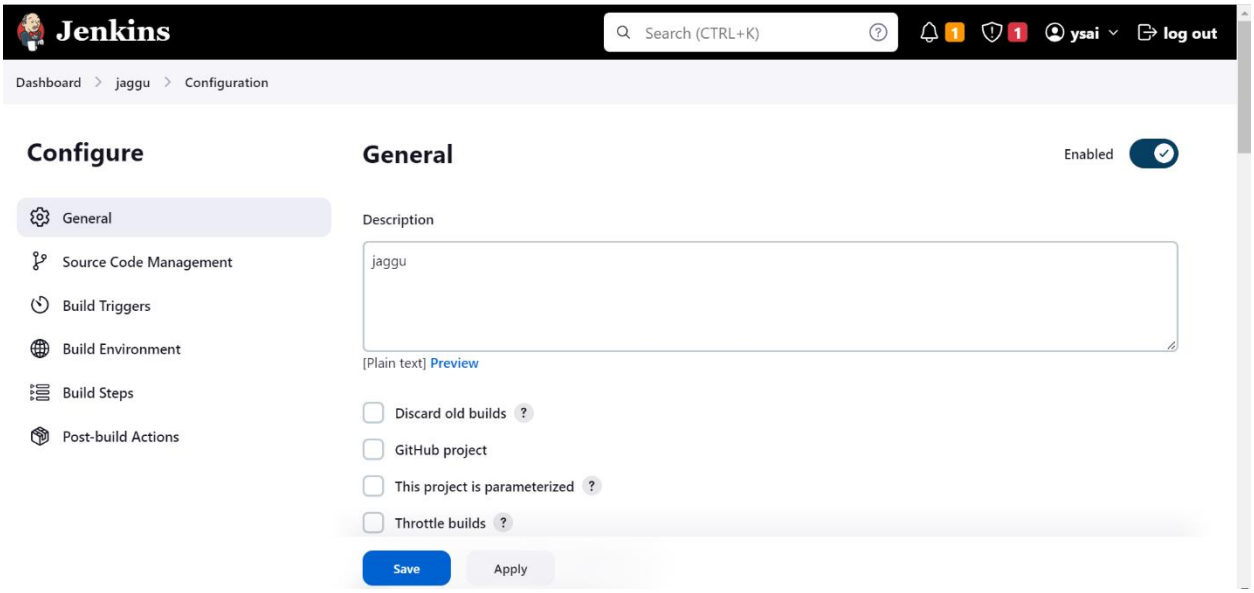
Output :



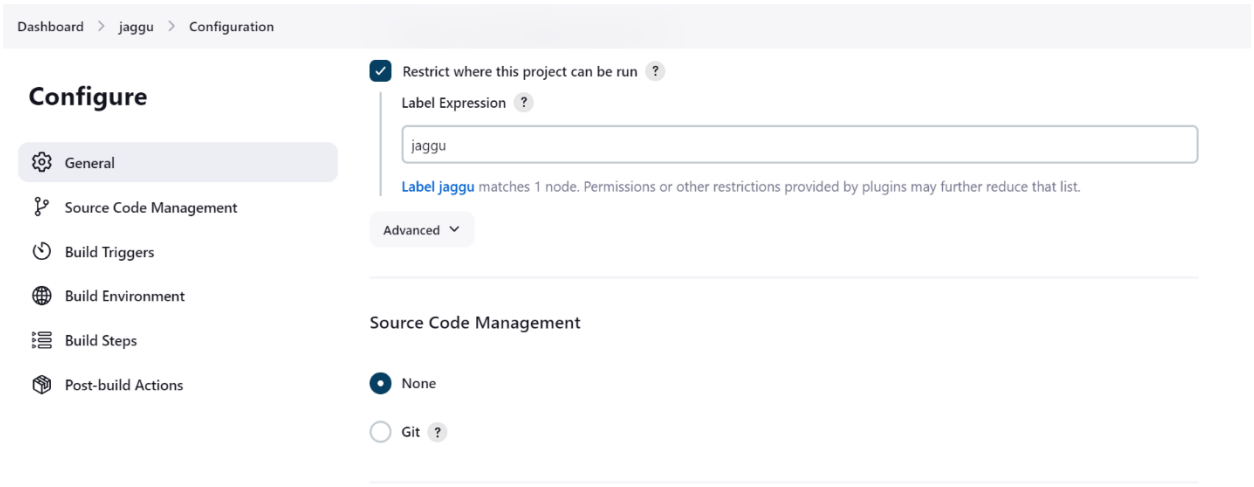
Jagadesh slave configuration :

- Created a new project by using Jagadeesh slave as a Jaggu.

Jenkins Master Slave Configuration

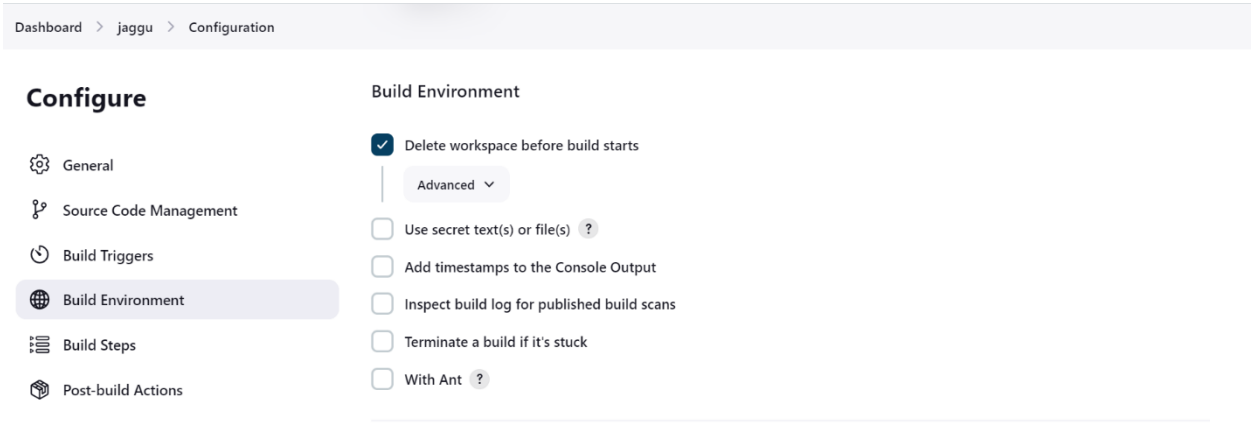


- Here I mentioned in my configuration where to run this project. I gave jaggu's slave label to identify the user.

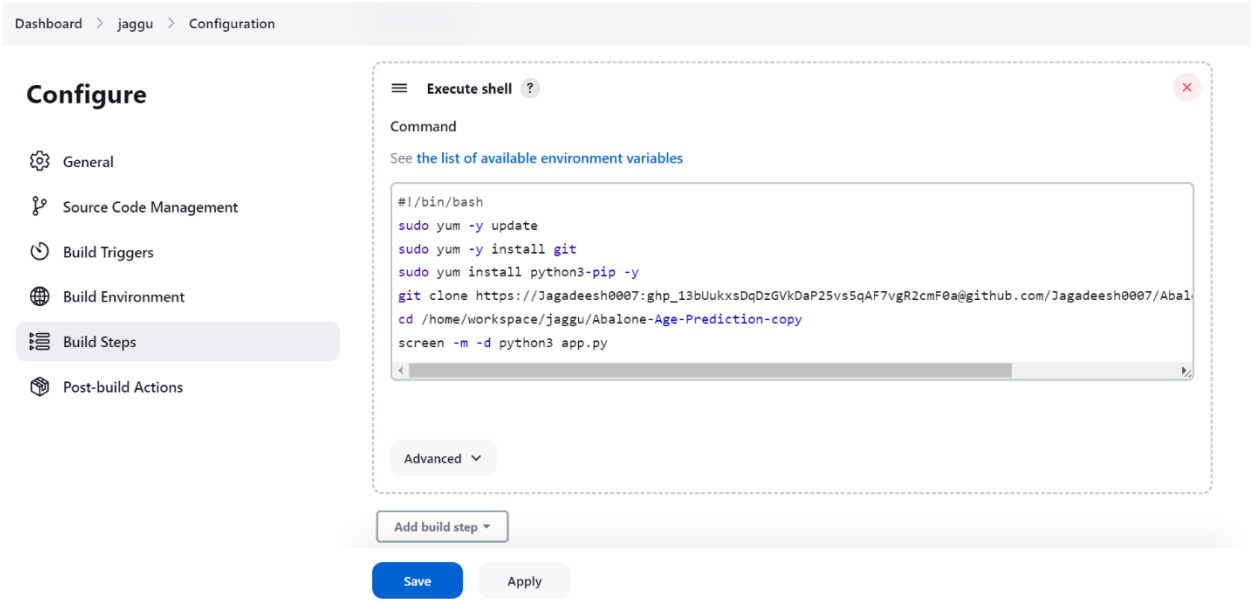


- In build environment I mention that delete workspace creating the project.

Jenkins Master Slave Configuration

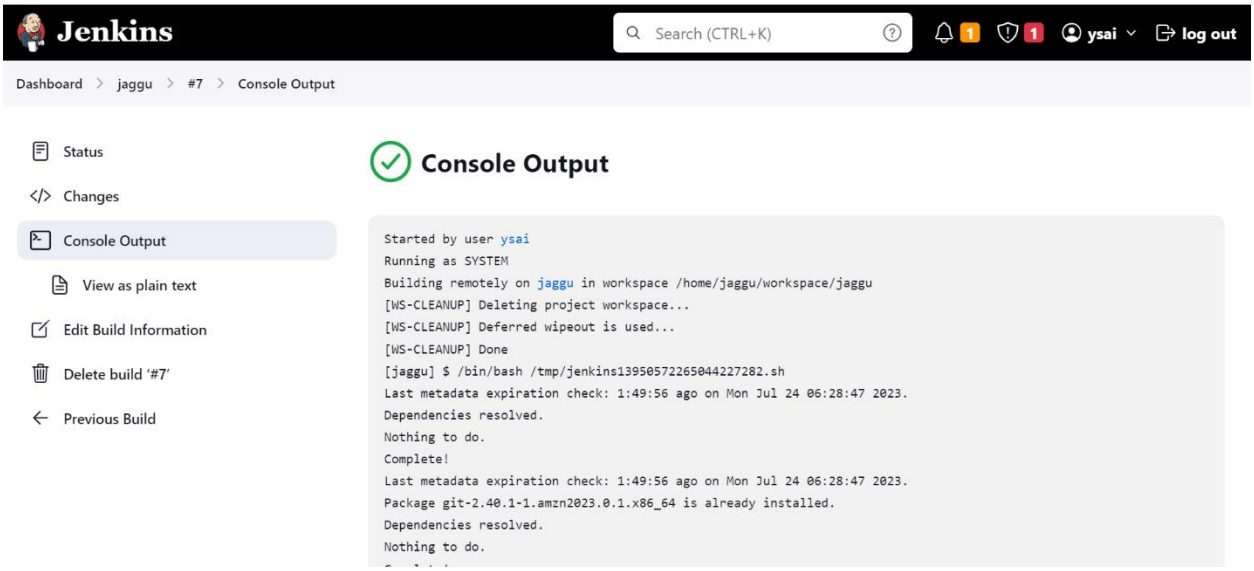


- Here I gave commands in execute shell to do automation.

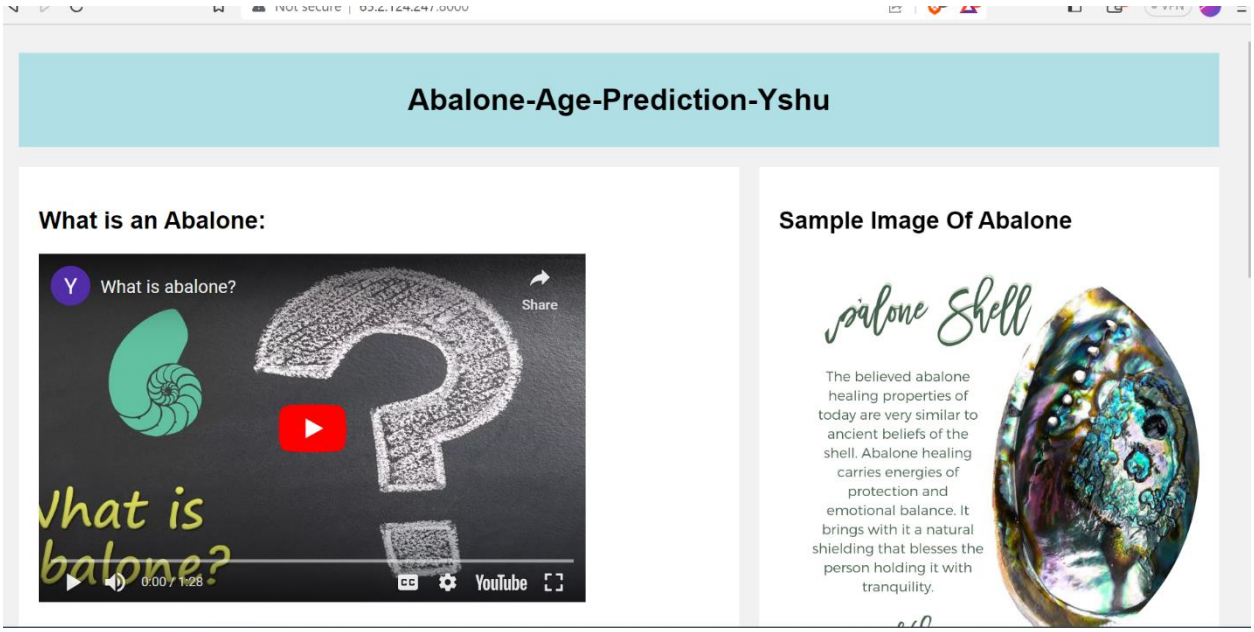


- After configuring I made build now and checked weather the console output is successful or not.

Jenkins Master Slave Configuration



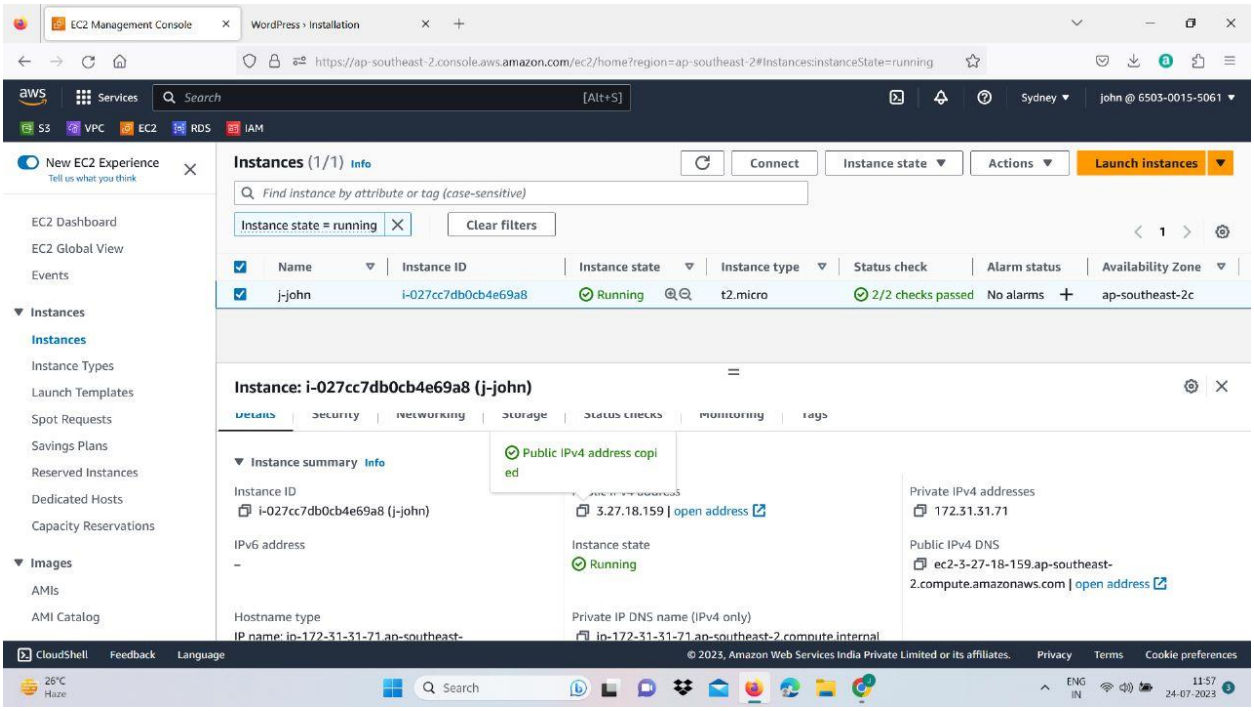
Output :



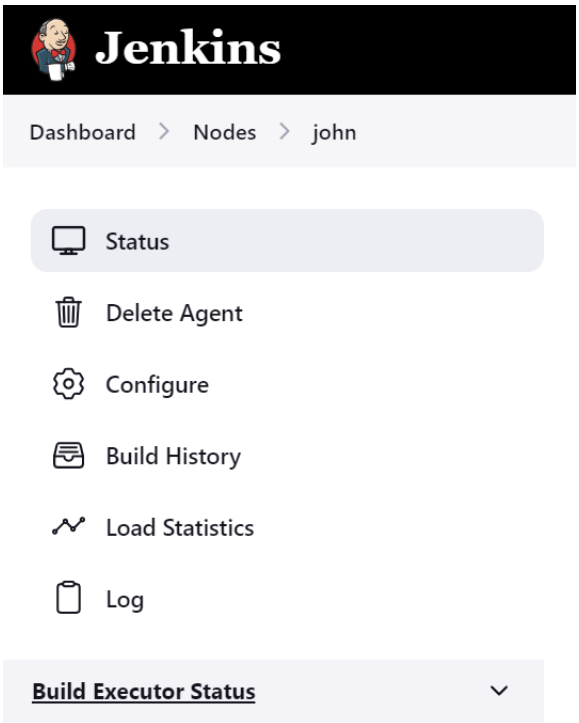
John slave configuration :

- In this John created one ec2 instance and created one user in that instance and shared to me.

Jenkins Master Slave Configuration

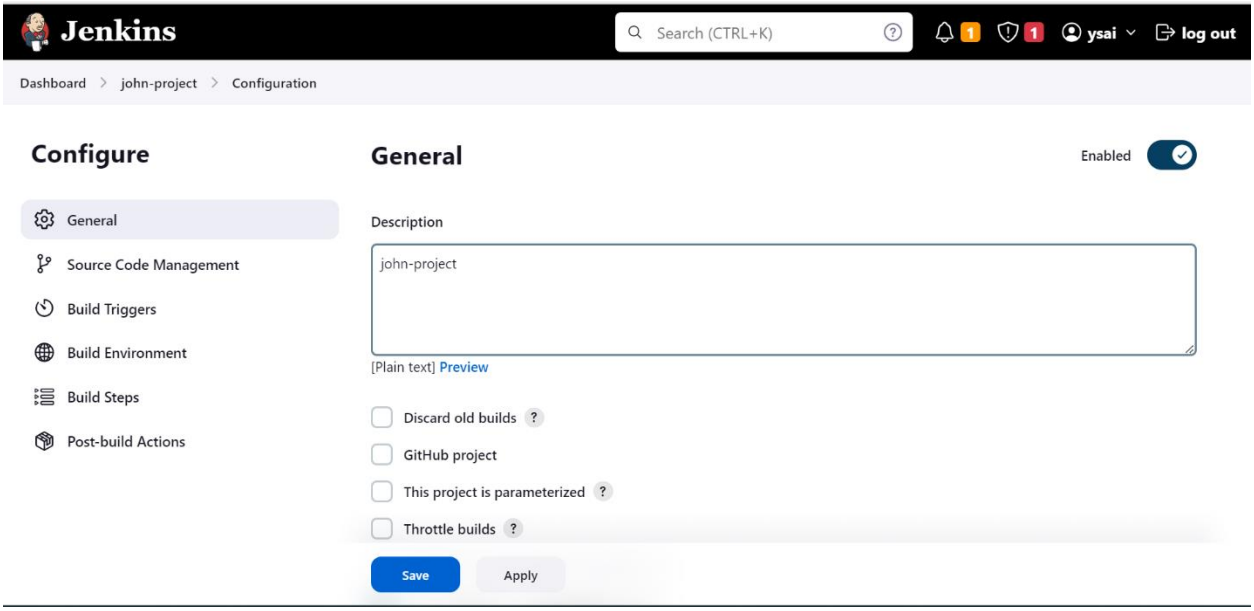


- Here I added John slave in my manage Jenkins = Nodes.

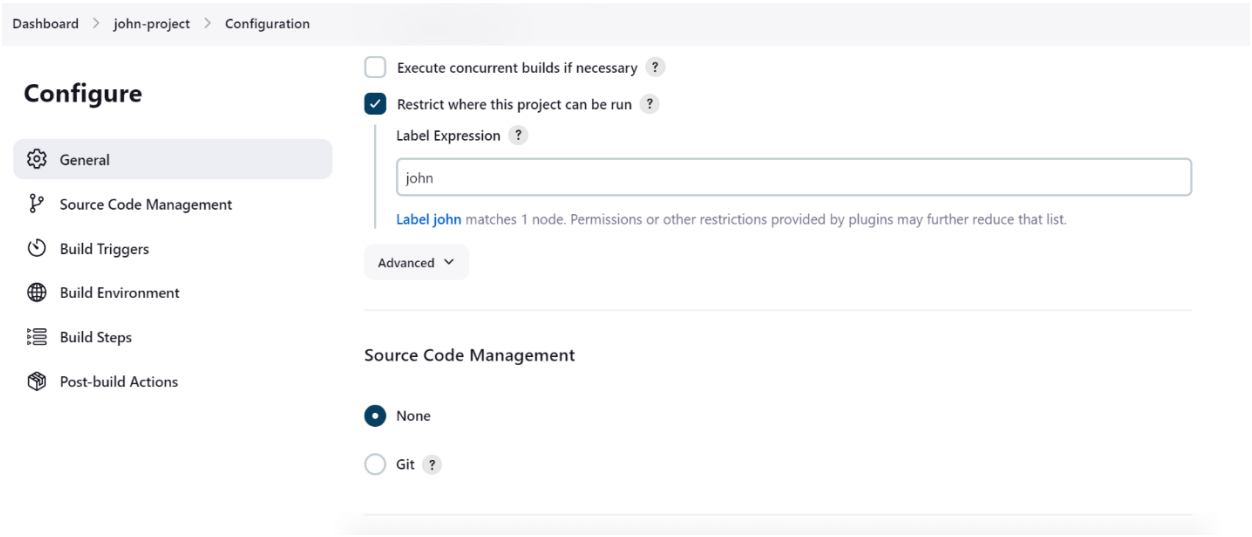


Jenkins Master Slave Configuration

- Created a new project by using John slave as a John-project.

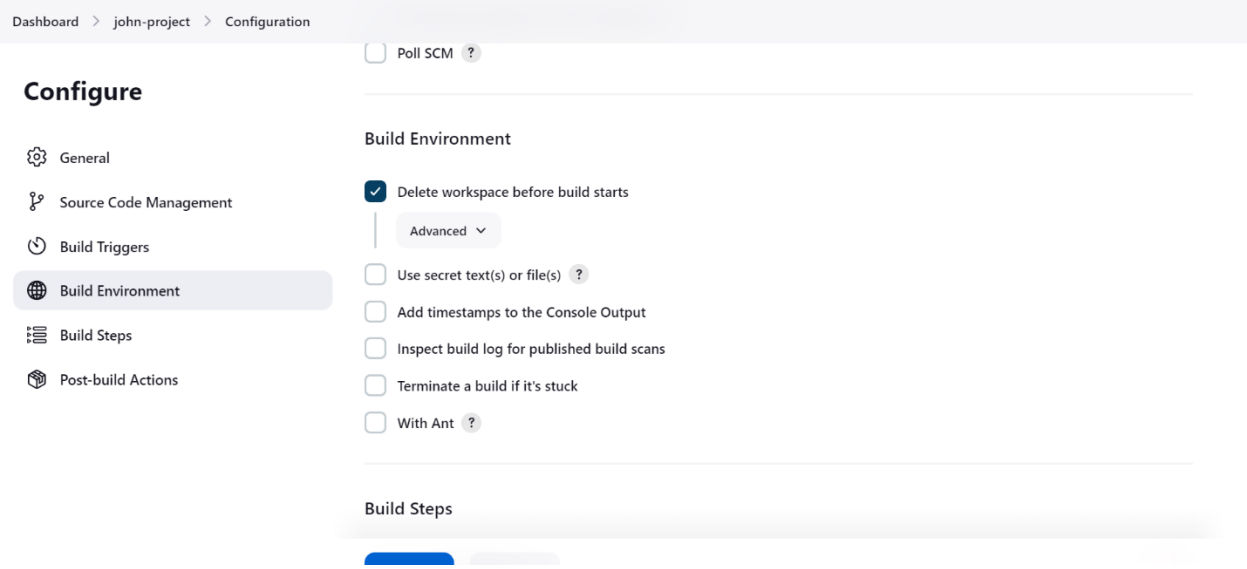


- Here I mentioned in my configuration where to run this project. I gave John's slave label to identify the user.

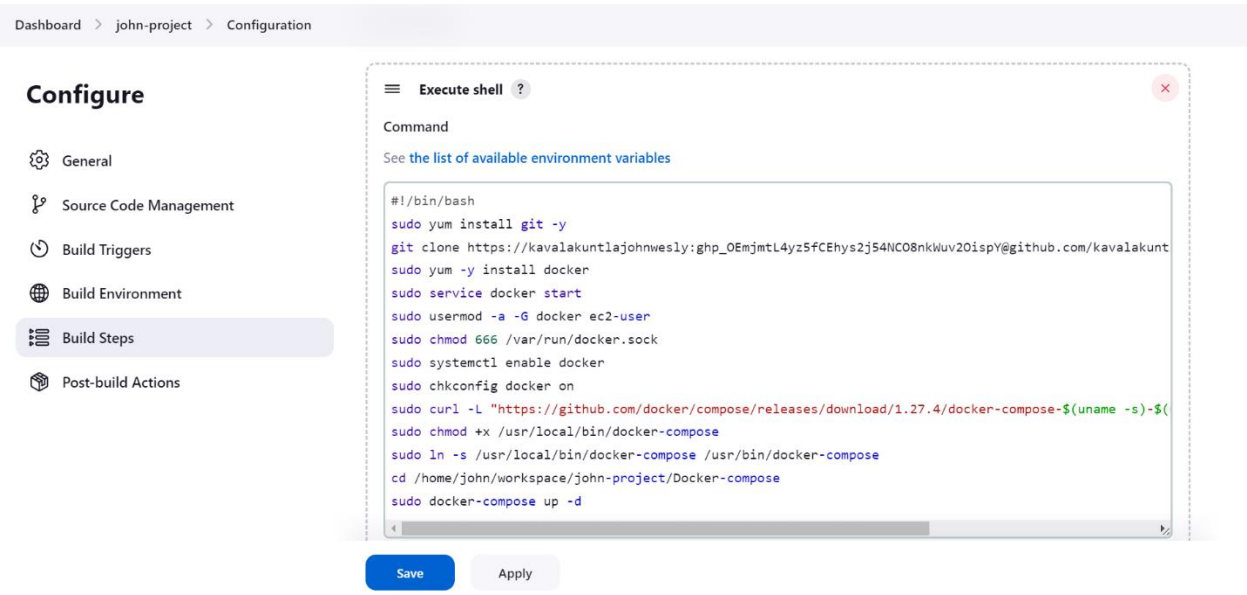


- In build environment I mention that delete workspace creating the project.

Jenkins Master Slave Configuration

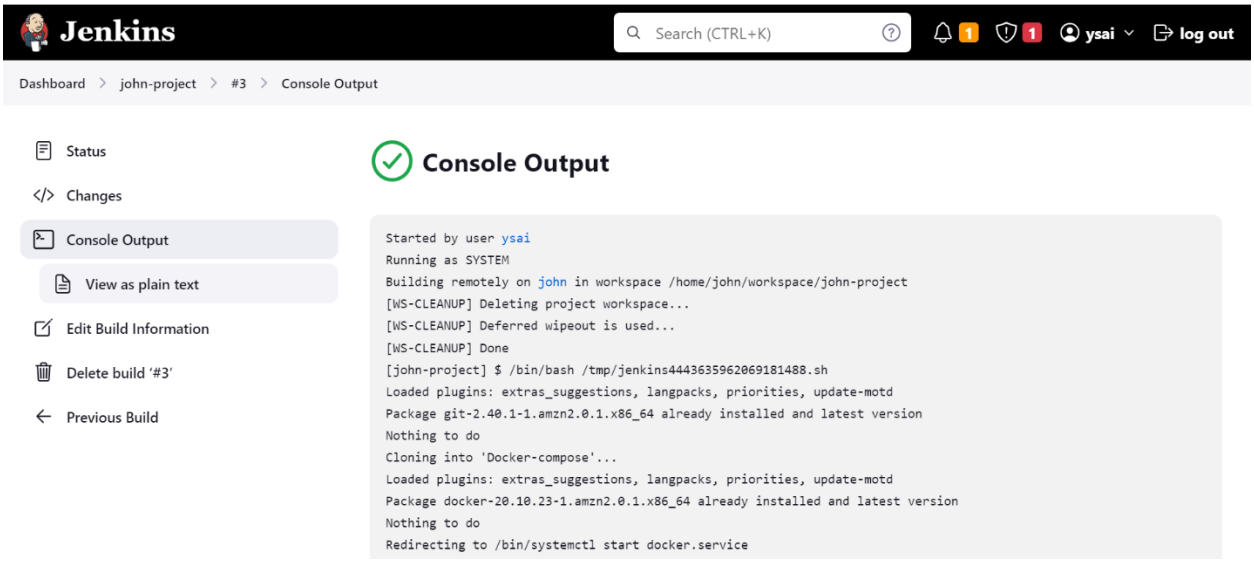


- Here I gave commands in execute shell to do automation.



- After configuring I made build now and checked weather the console output is successful or not.

Jenkins Master Slave Configuration



Output :