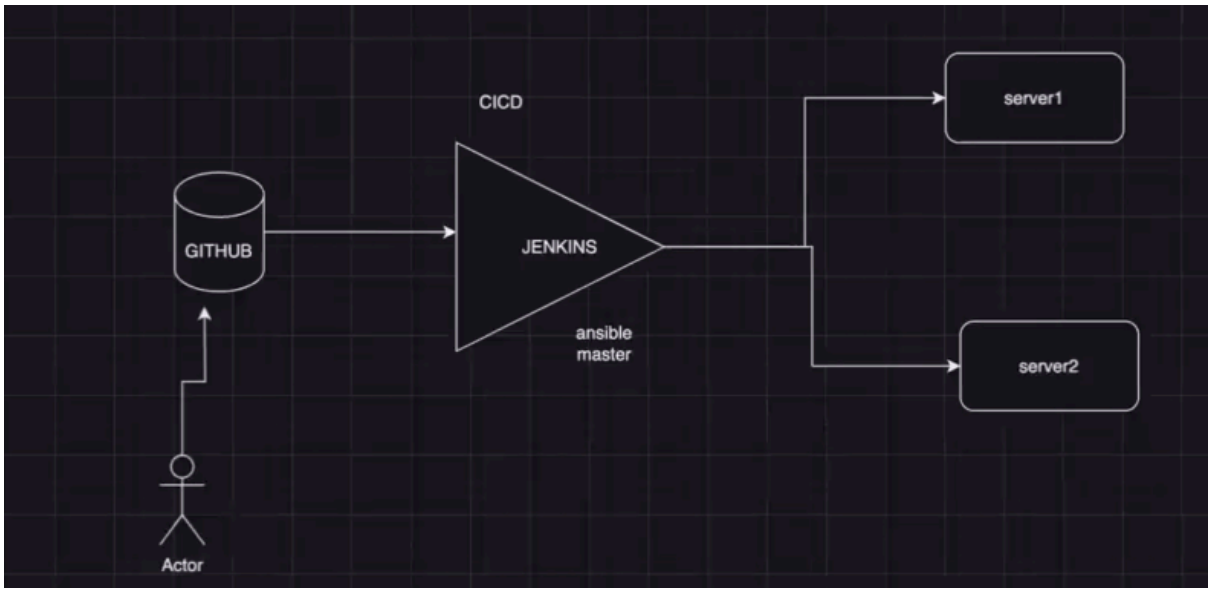
Jenkins CI/CD pipeline for a Maven-based Web Application



How to setup Jenkins

Installation, Configuration

Step 1. Create ec2-instance (RedHat) for Jenkins

Step 2. sudo yum install wget -y

Step 3. sudo yum install git -y

Step 4. sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo

Step 5. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

Step 6. sudo yum upgrade

Add required dependencies for the jenkins package
Step 7. sudo yum install java-17-openjdk

Step 8. sudo yum install jenkins -y

Step 9. sudo systemctl daemon-reload

Step 10. Start Jenkins
You can enable the Jenkins service to start at boot with the command:

$ sudo systemctl enable jenkins

Step11.You can start the Jenkins service with the command:

$ sudo systemctl start jenkins

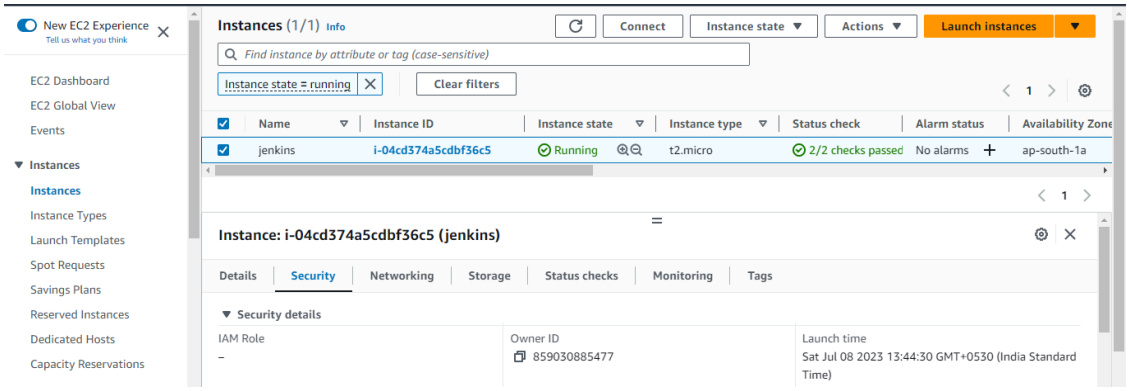
* if an error occurs, follow step 7, reinstall java

Step12.You can check the status of the Jenkins service using the command:

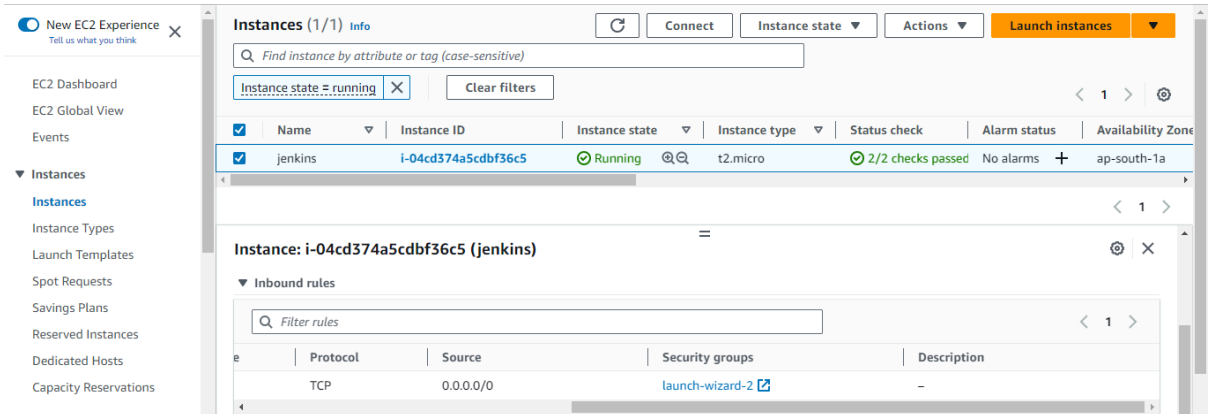
$ sudo systemctl status jenkins

Step 13: Now we need to change some inbound rules to access the Jenkin terminal.

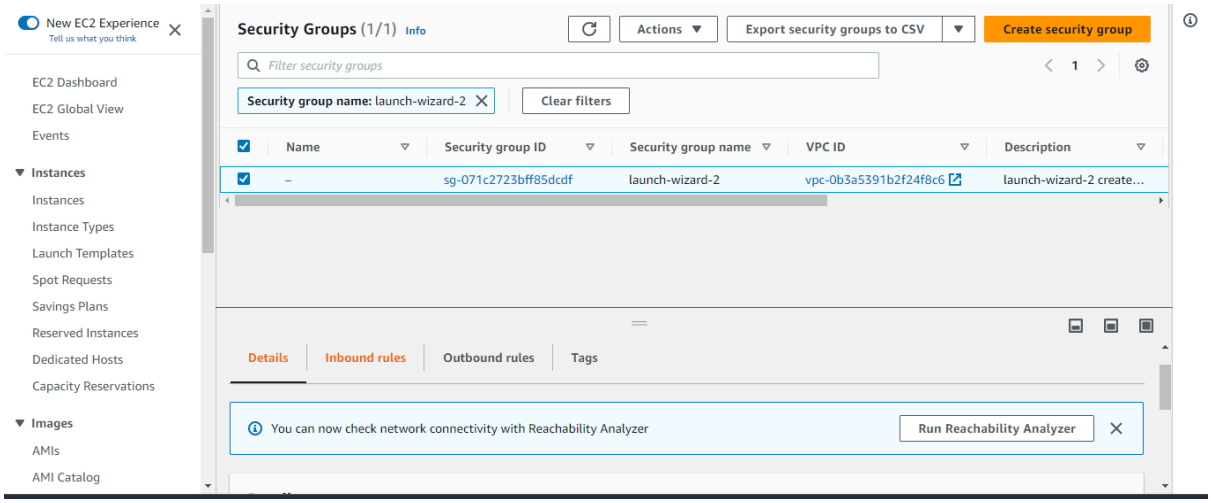
1. Select jenkin instance and goto security



2. Click on launch-wizard



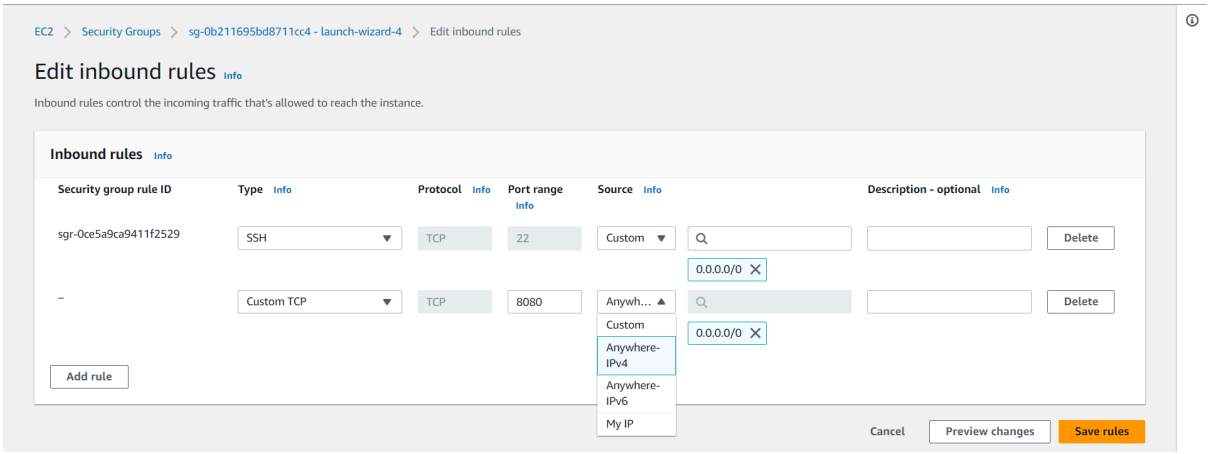
3. Click on inbound rule at the bottom and click edit inbound rule



4. Add rule, select as follows

ec2-instances -> select server 1 and server 2-> select security-> security group->Edit inbound rules->Add rules-> Save

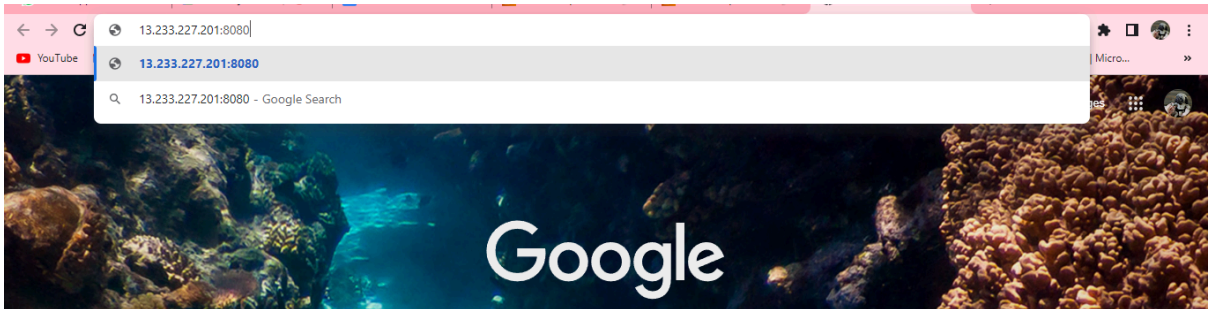
For inbound rules
Type:-Custom TCP
Port range:-8080
Source:- Anywhereip4



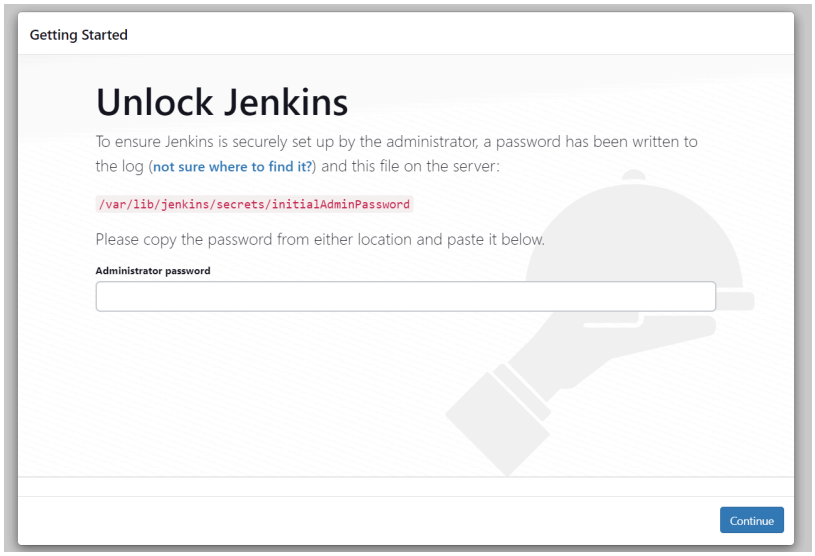
Step 14. Now copy the public IP of the Jenkin instance and put it into the Chrome browser like this...

Public_ip_jenkin_ec2:8080

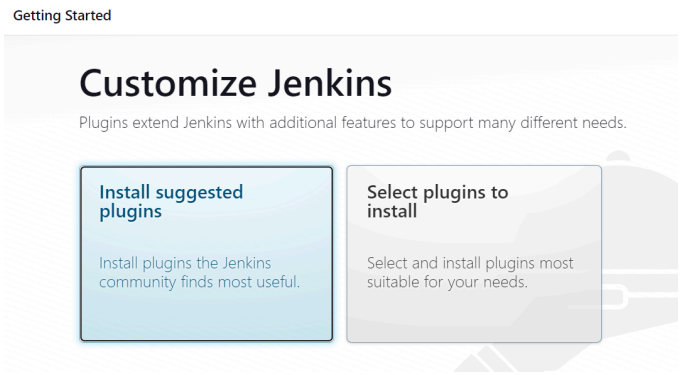
13.233.227.201:8080



Step 16: Enter the password.

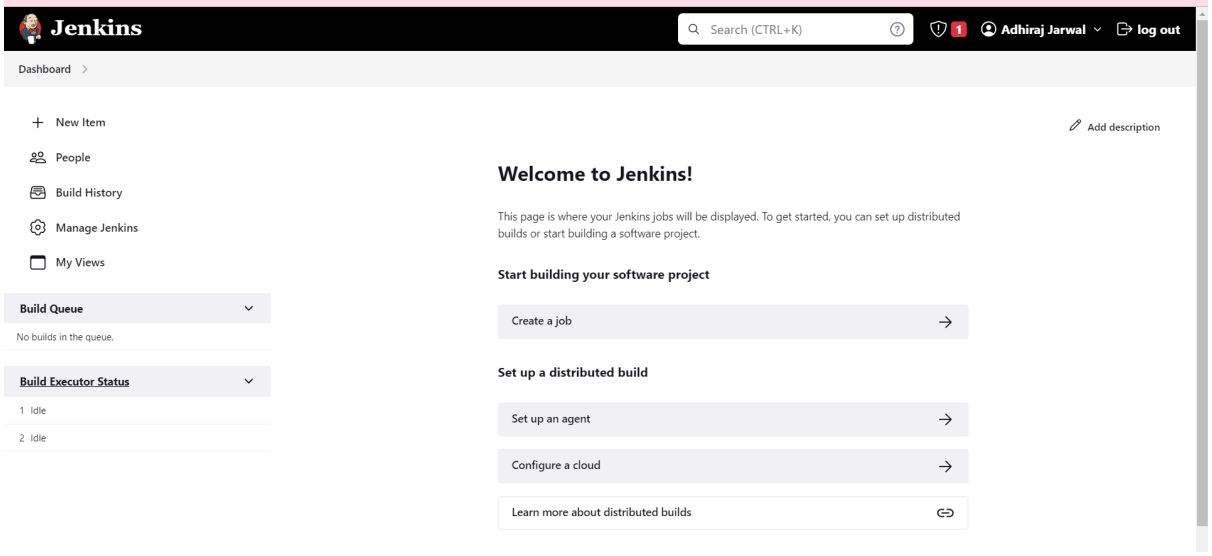


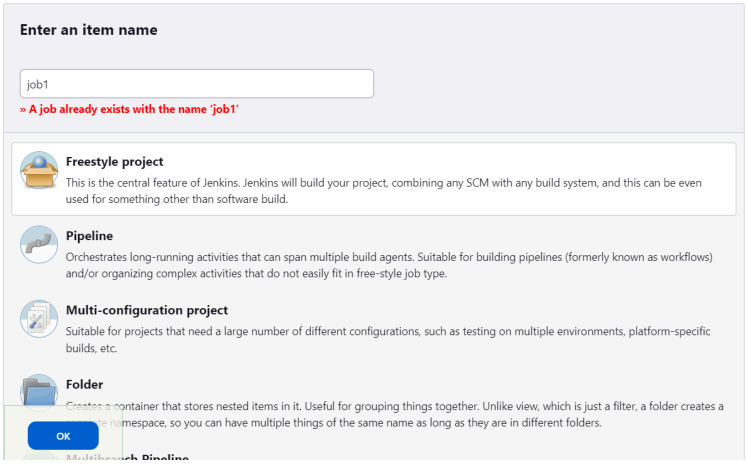

```
[Jenkins ]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
c1a50d80f85049d09568ea10f2f4be9f
[Jenkins ]$
```



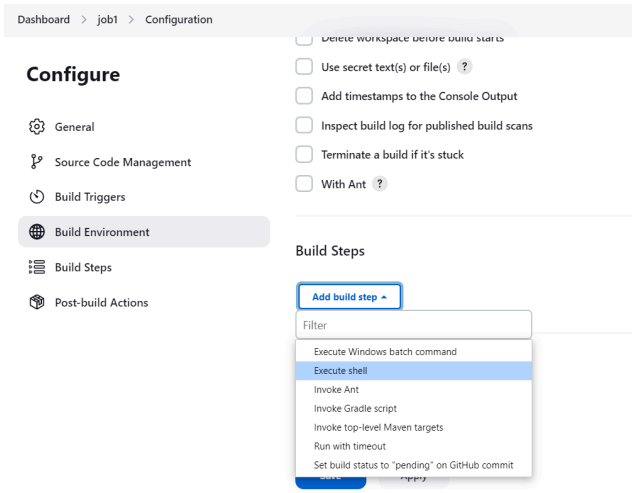
Now do the setup.

Step 17. Now click on create a job

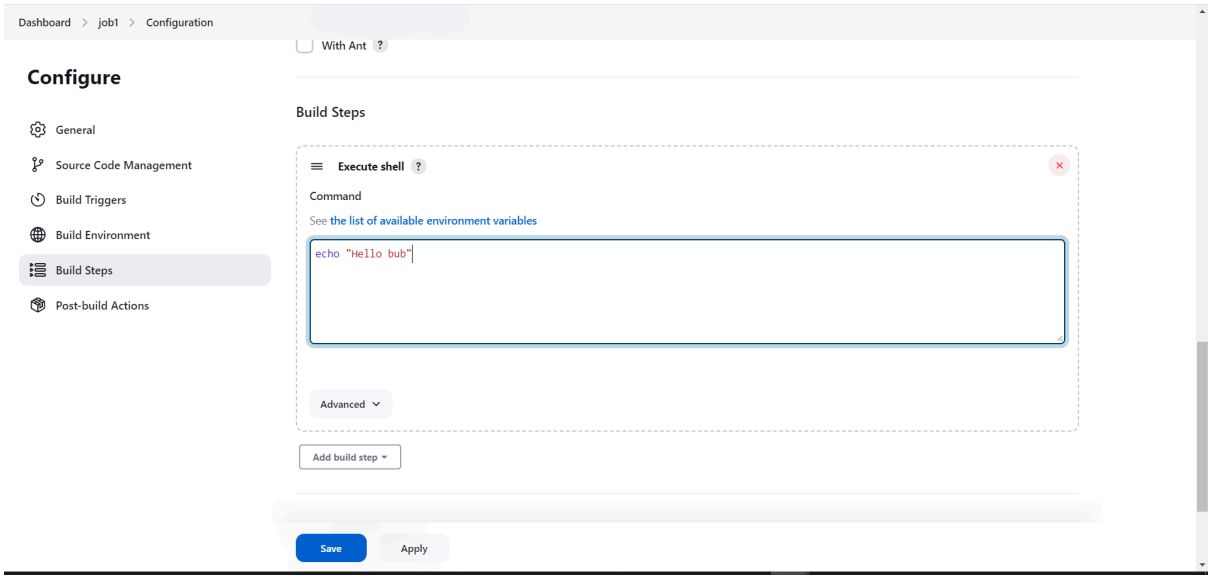




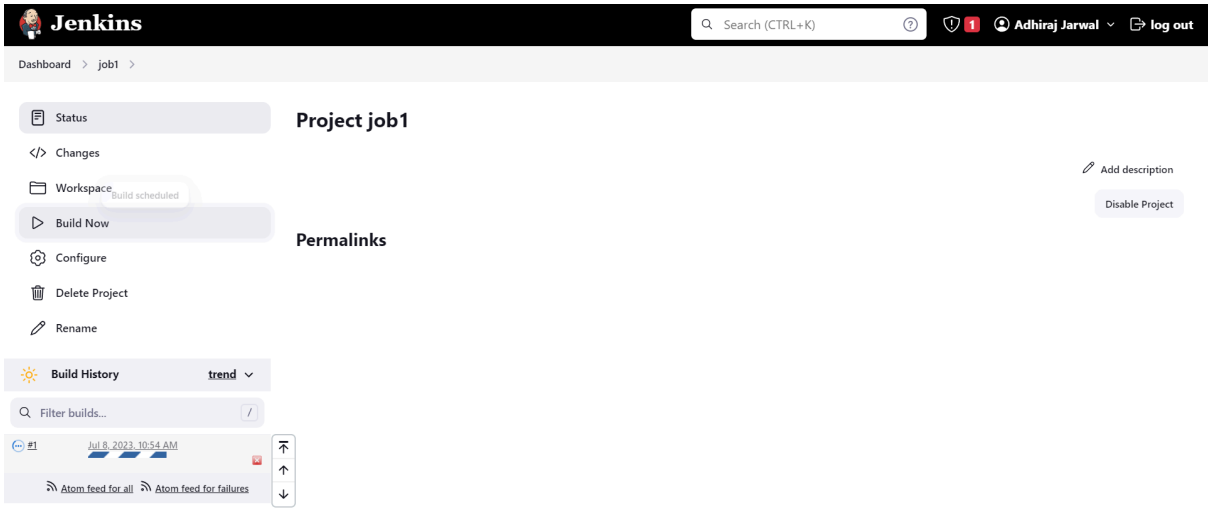
Select build environment and click on execute shell



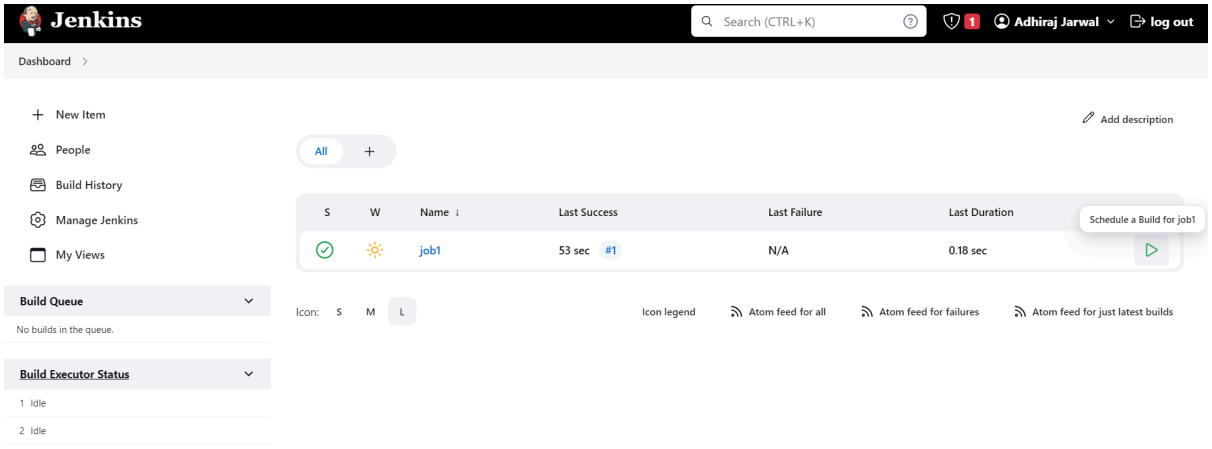
Write a script and save



Click on build now

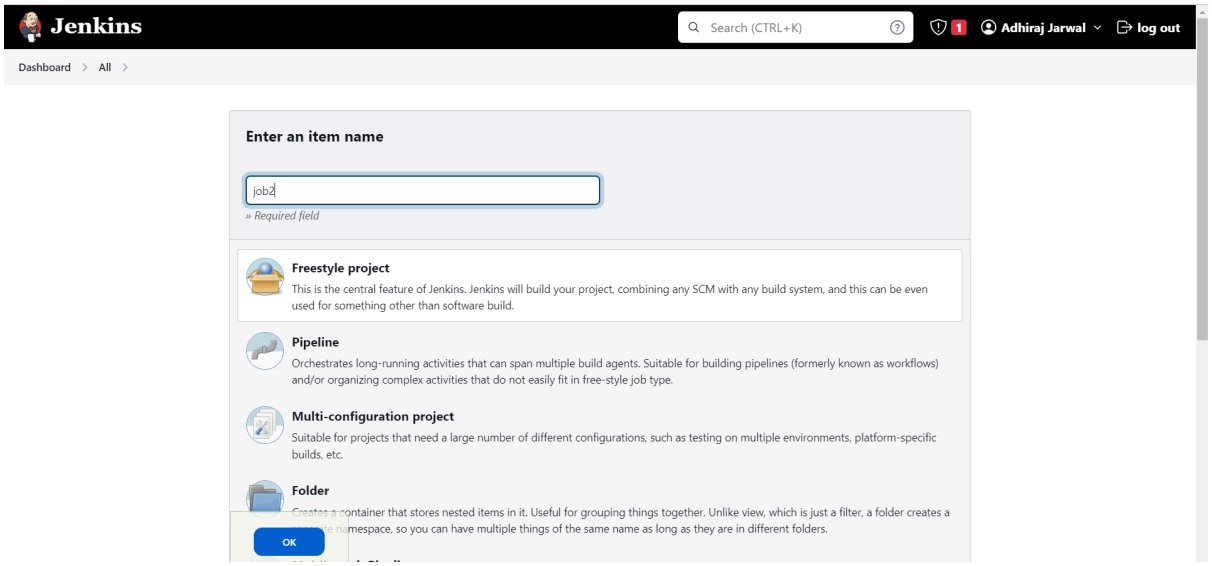


Or you can also build like this



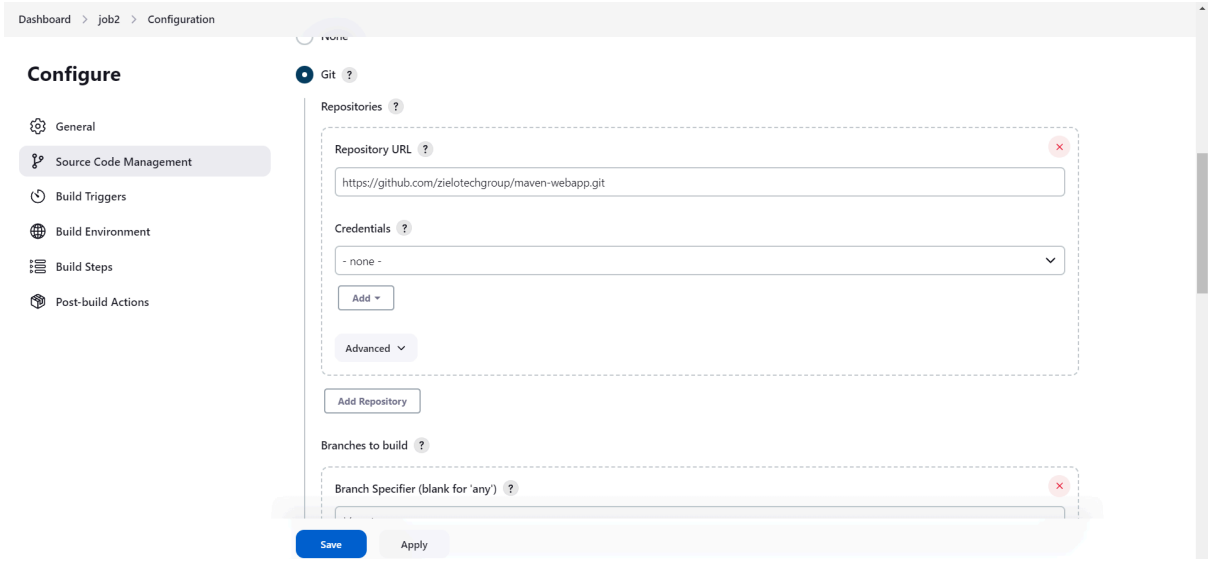
Step 18: create another job by clicking on create new on the left side.

We will connect jenkins with git repo now

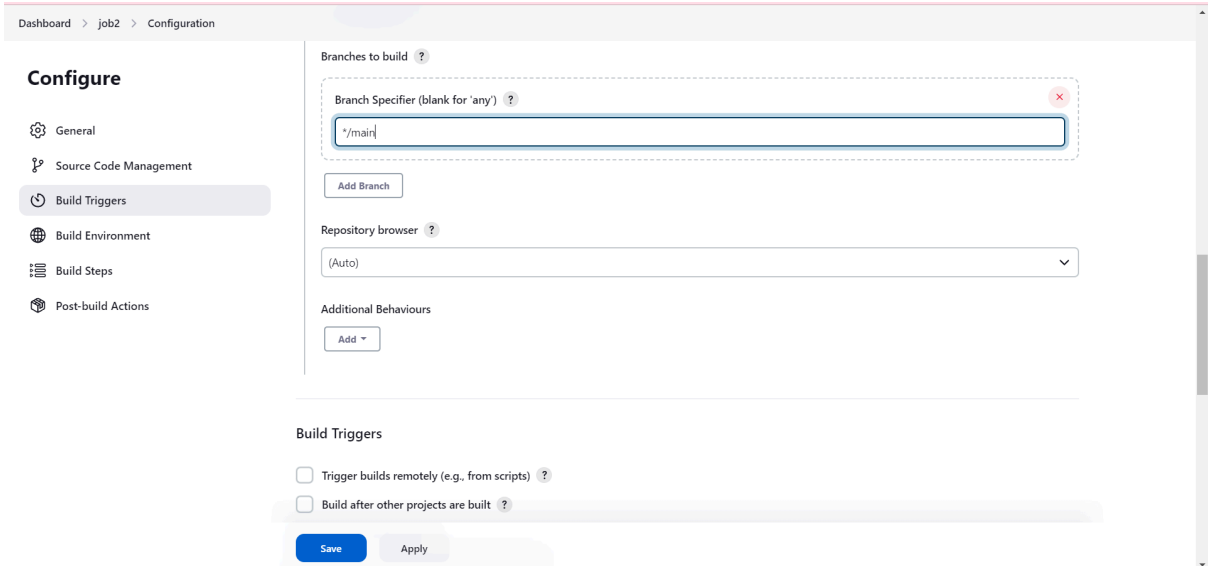


Url : https://github.com/Organization-for-fun/BigBakset.git

We are cloning a repo using url, manually

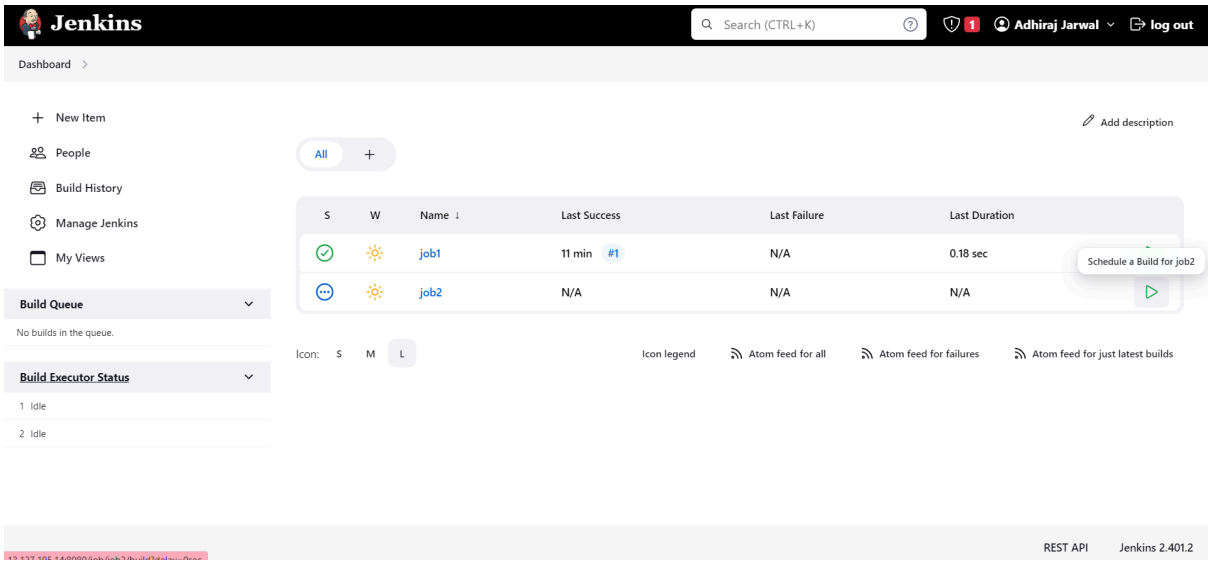


Change branch to */main

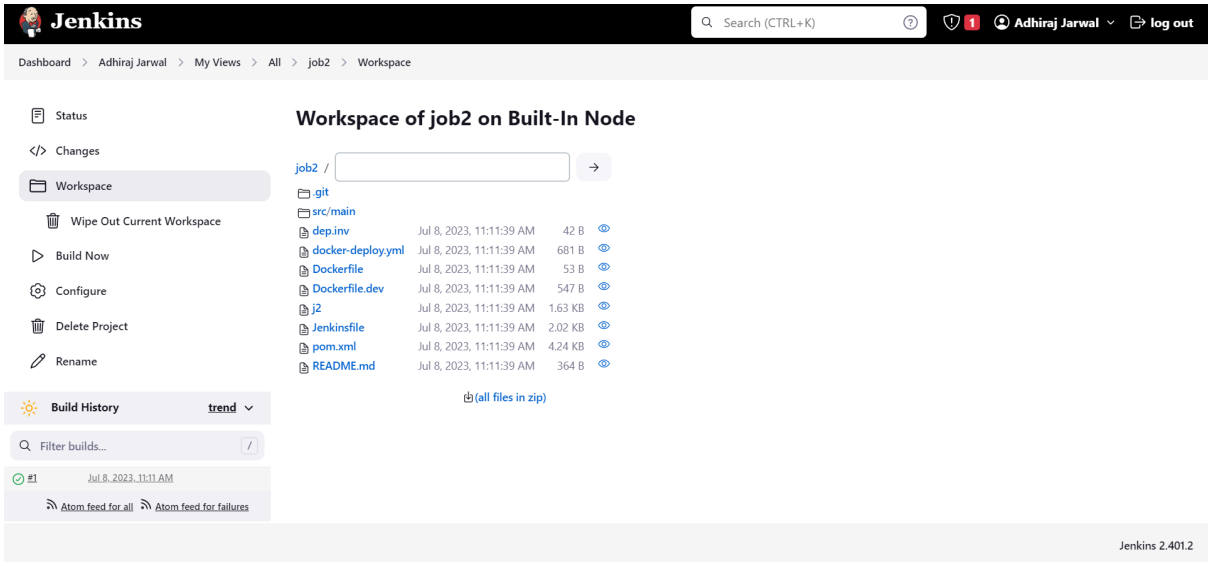


Click on save

Build job 2



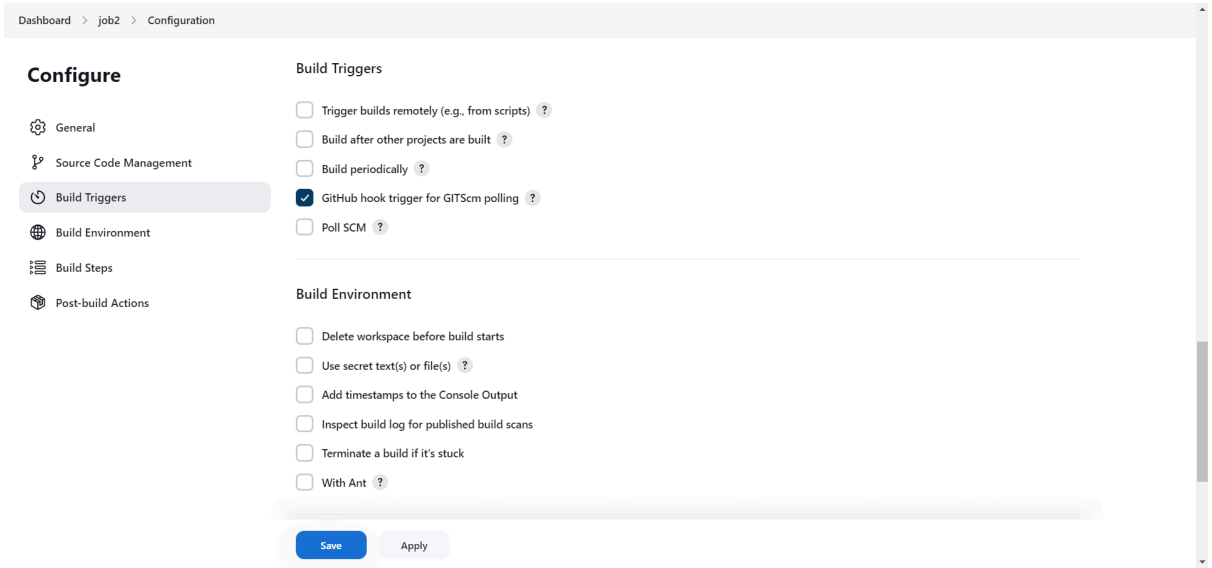
Repository cloned successfully (job2->workspace)



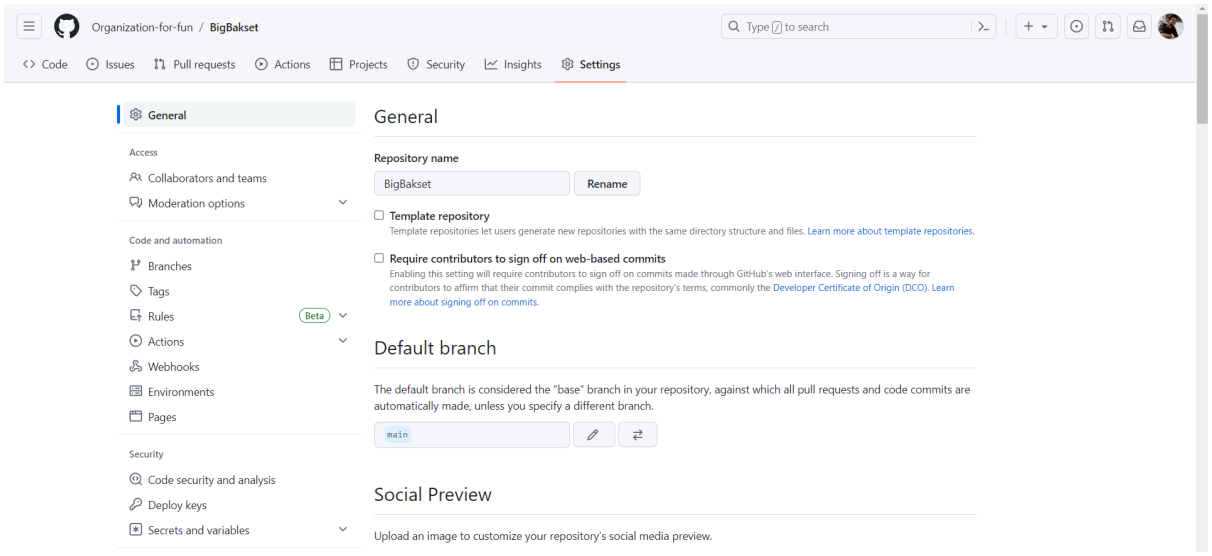
Now To AUTOMATE, the same thing, means cloning the repo after someone committed the file after making some changes.
(Dashboard->job2->configuration->build triggers)

Select Github

And save.

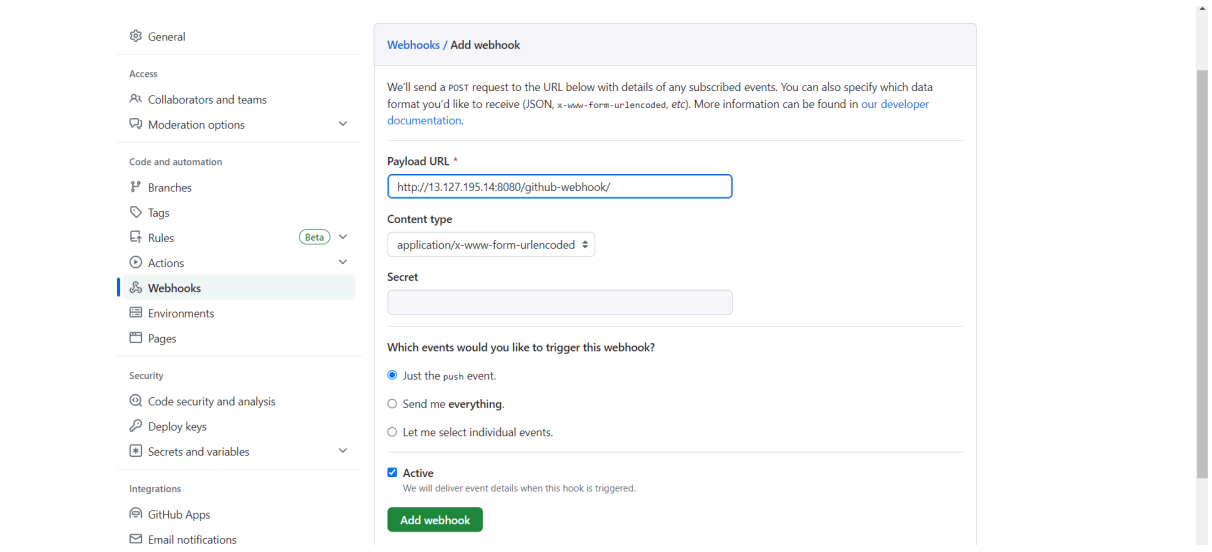


Now go to github repo -> setting -> webhooks -> add webhook



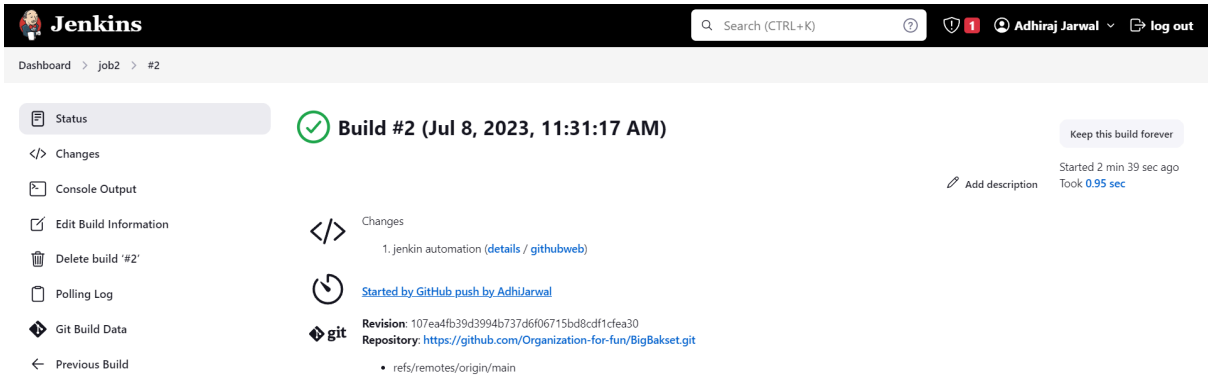
Now in payload url:

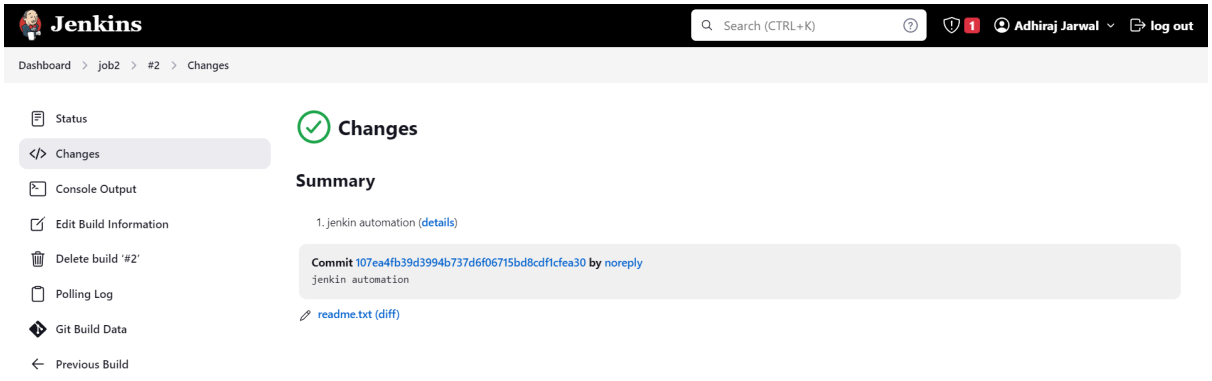
http://public_ip_jenkin/github-webhook/



Add a webhook.

Jar ata apan repository madhlya files madhe changes kele, ani commit kela tr jenkin job2 madhe automatically #2 asa create hota, ani tya madhe je je changes kele hote te sgla dista.

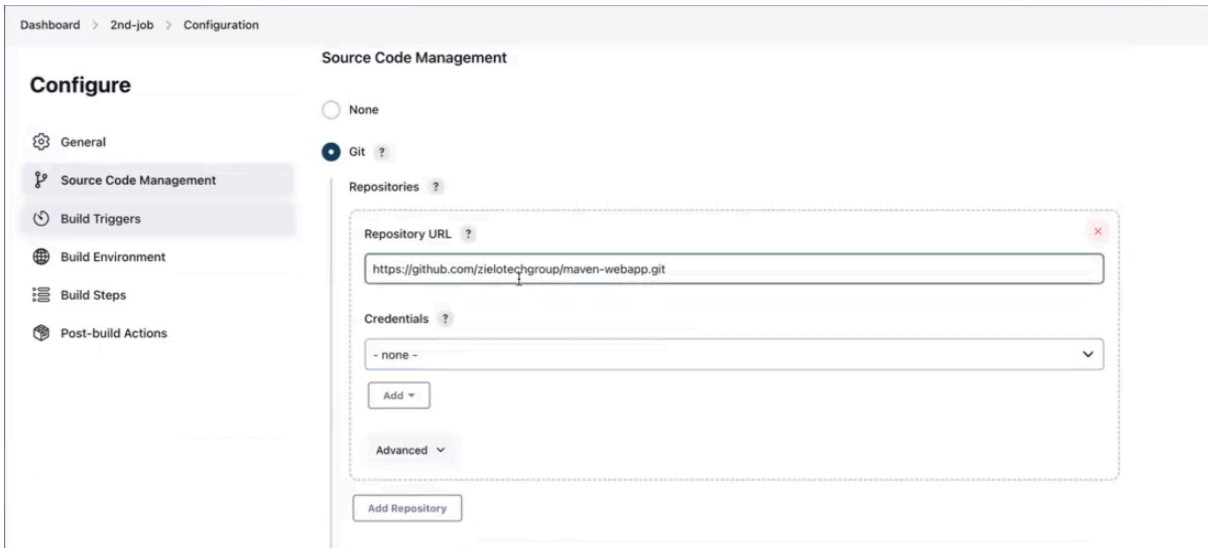




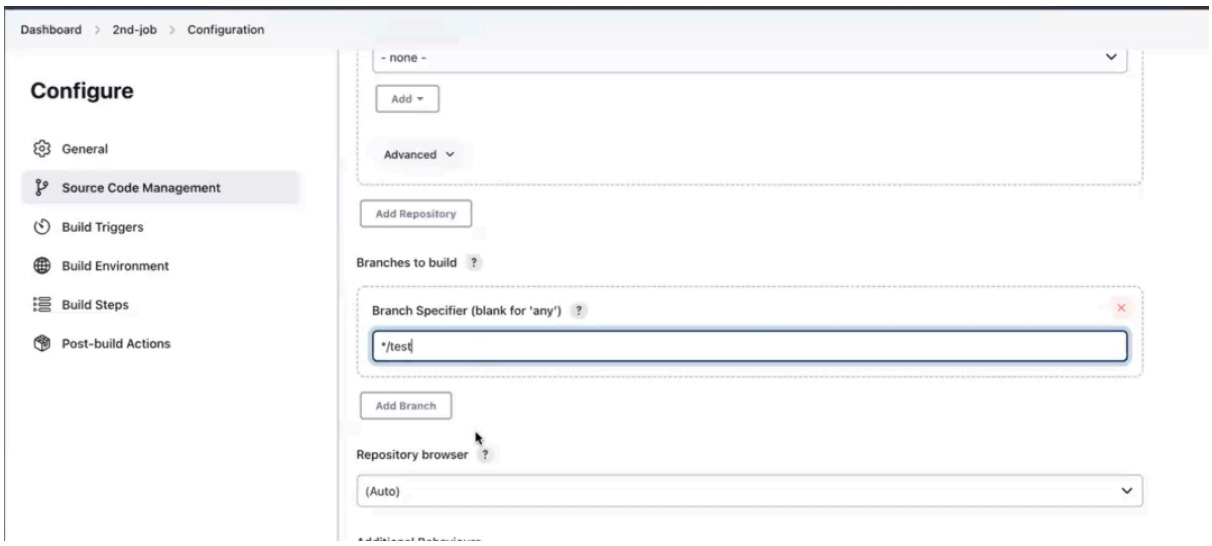
==========

==========

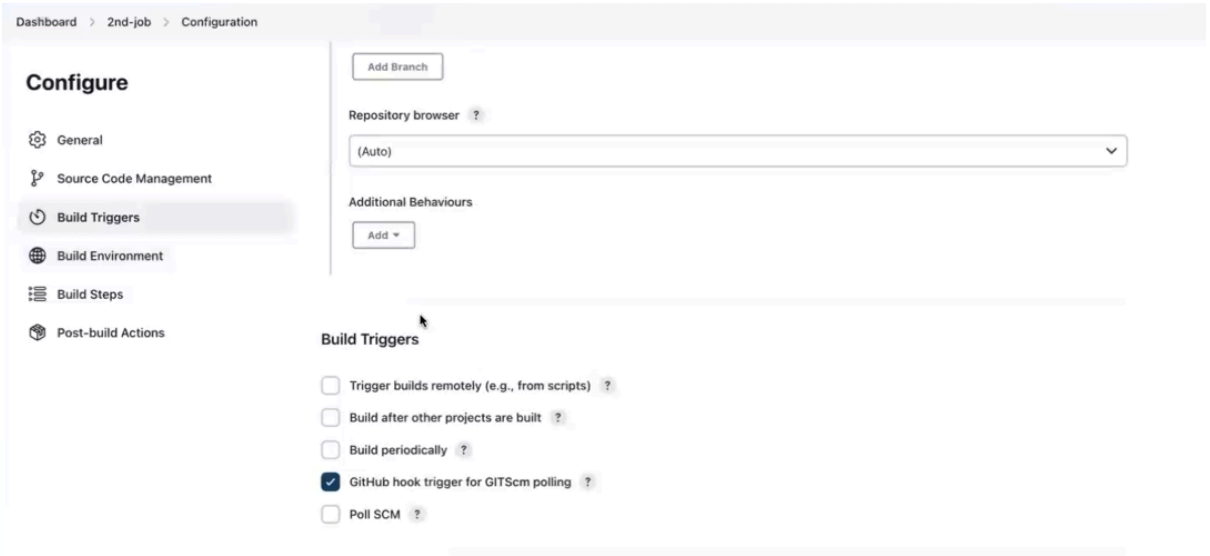
Step1:-Add the GitHub repository URL
https://github.com/zielotechgroup/maven-webapp.git



Step2:- Write the branch name:- test

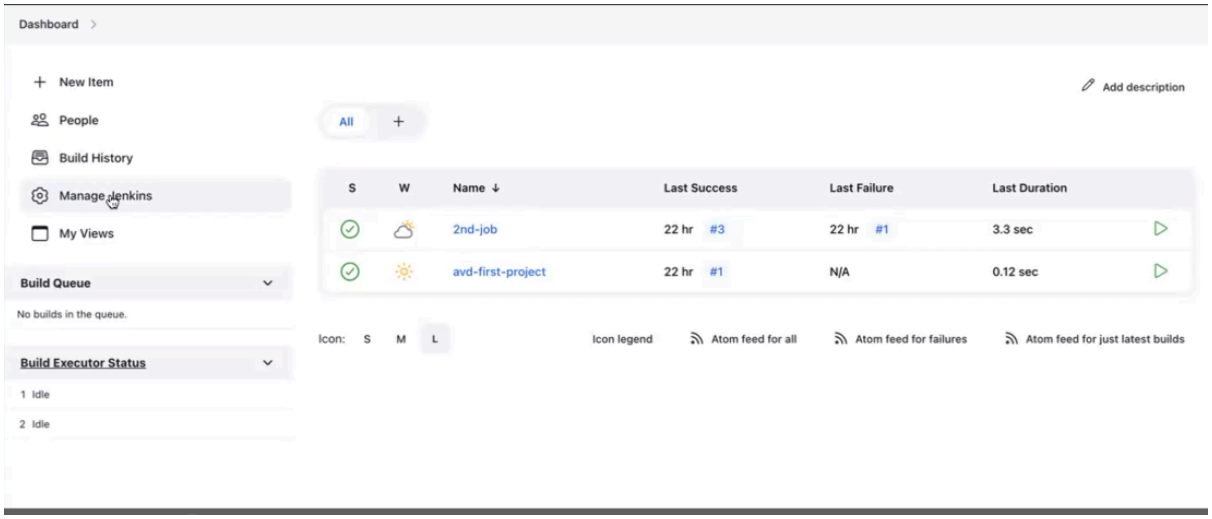


Step3:- Now select the GitHub hook trigger for GITSCM polling

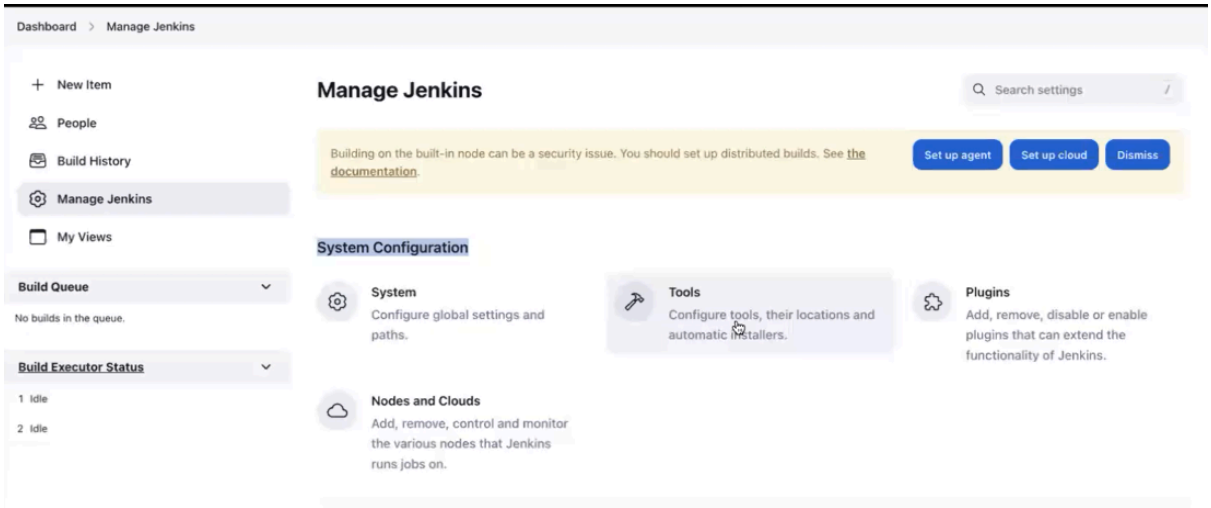


Step 5:- Now we want a maven tool for 2nd job, so as to build the source code and to generate a bundle

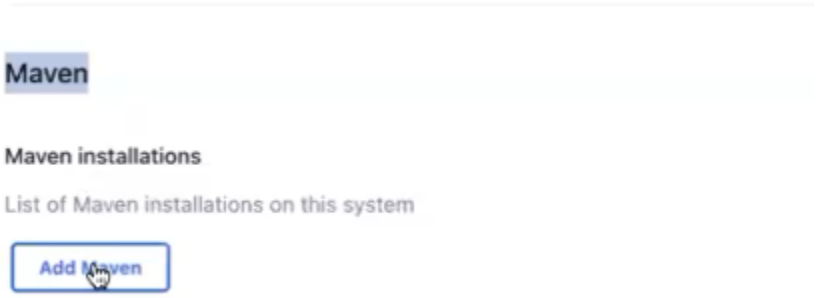
Click Manage Jenkins



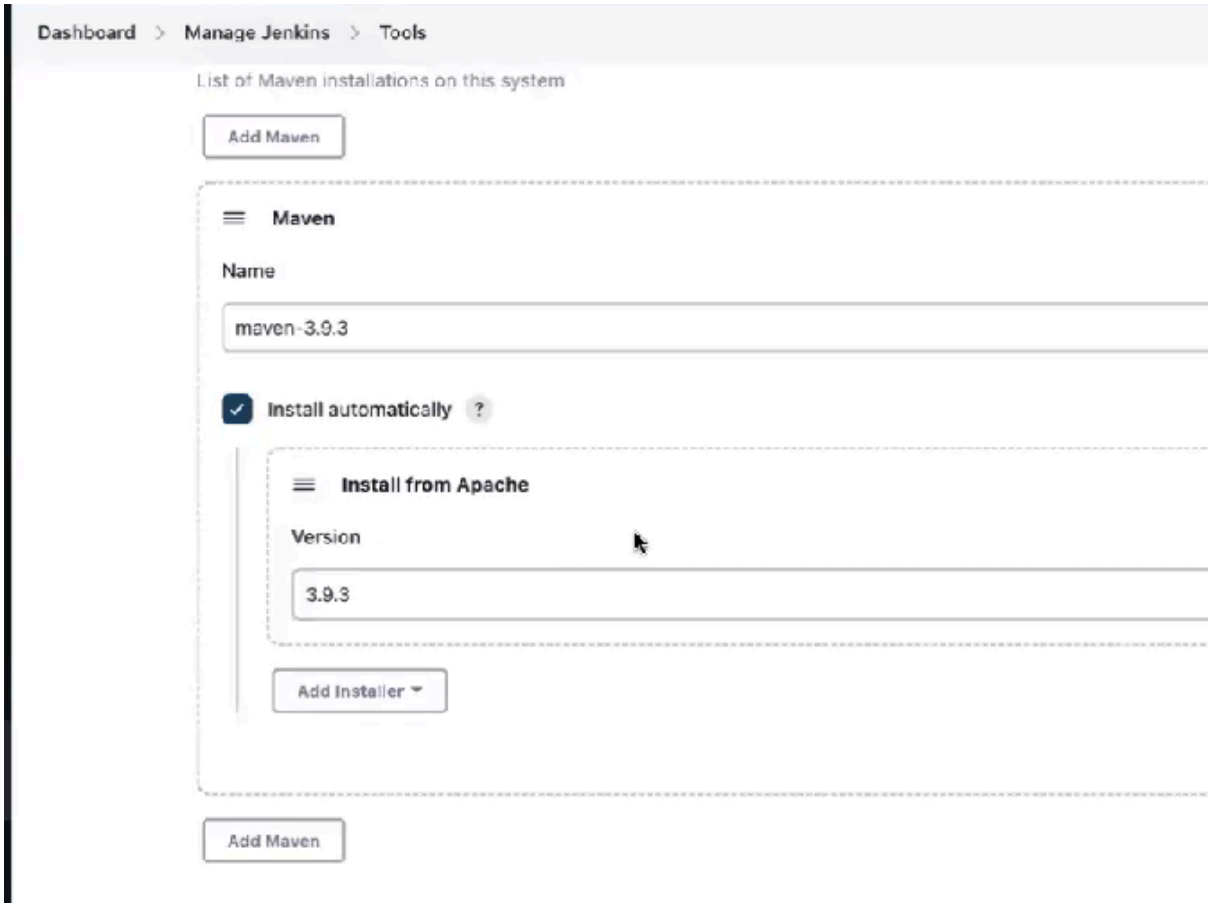
Step 6:- Manage Jenkins-> Under system configuration ->Click Tools



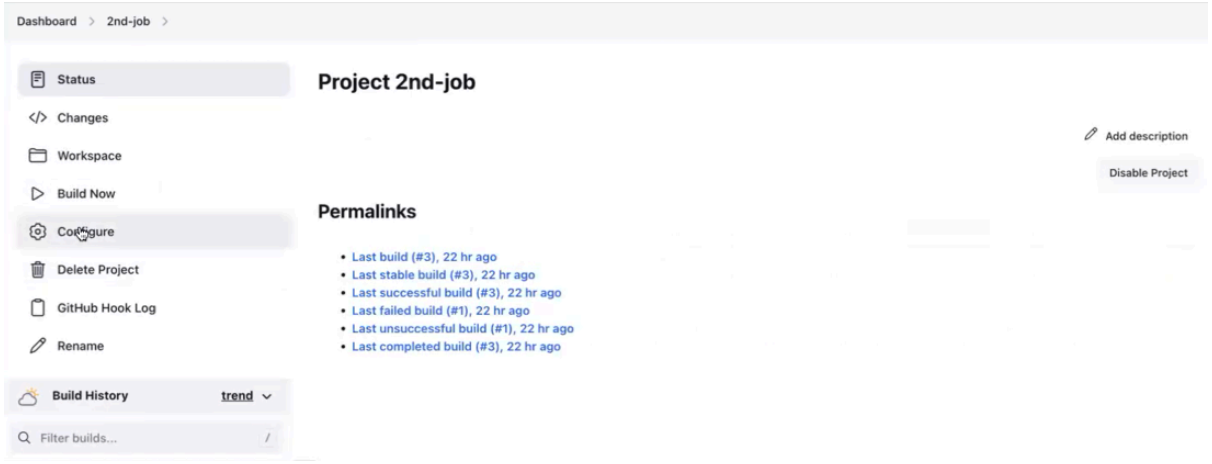
Step 7:- Now scroll down and you will see ‘Maven’ under this click Add Maven



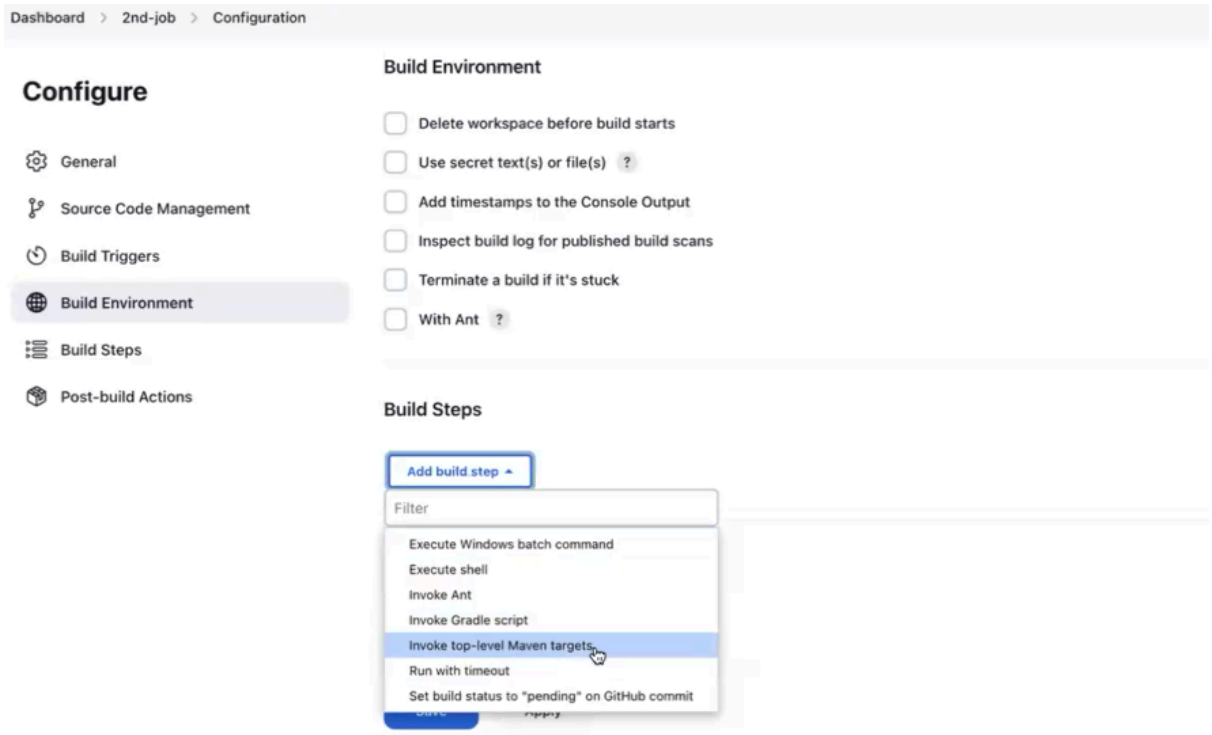
Step 8:- Add name ,select version and then click on save



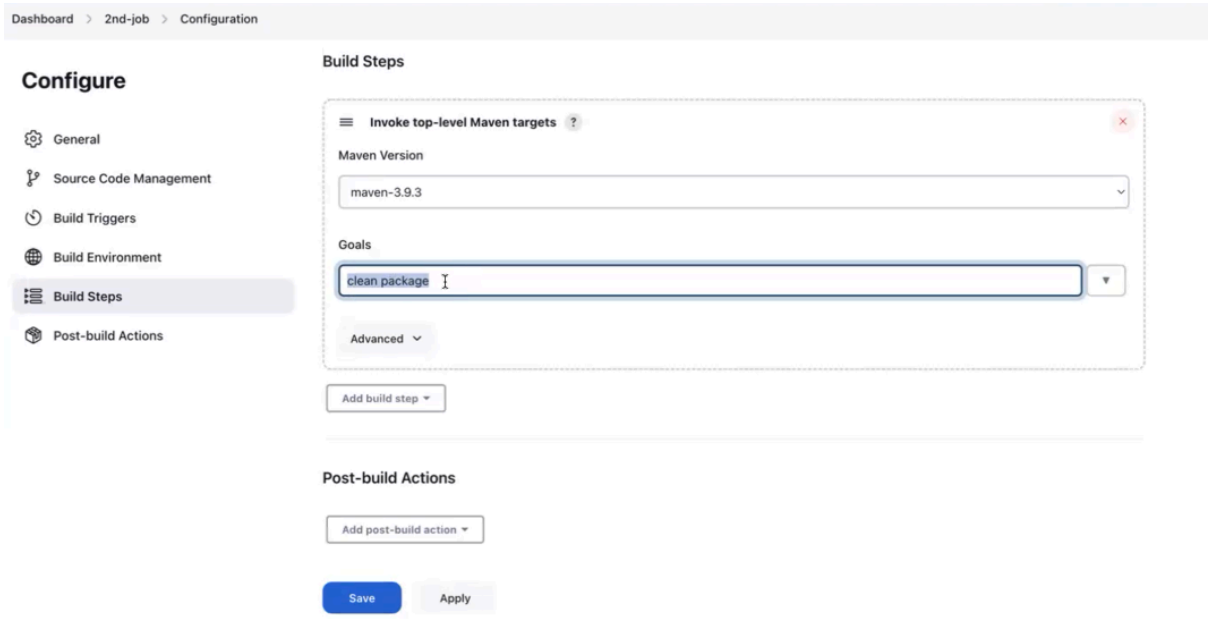
Step 9:- Now let's go to job2 to configure it.
Click on 2nd-job-> Dashboard -> configure



Step10:- Now scroll down and in the Build Steps section ->click on down arrow ->select Invoke top-level Maven targets



Step 11:- Now select the version name and enter the goal.
Goal :- clean package
Click save



Step12:- Now do some changes in the test branch and hit commit

Step13:- Now create two servers for deployment(Ubuntu)

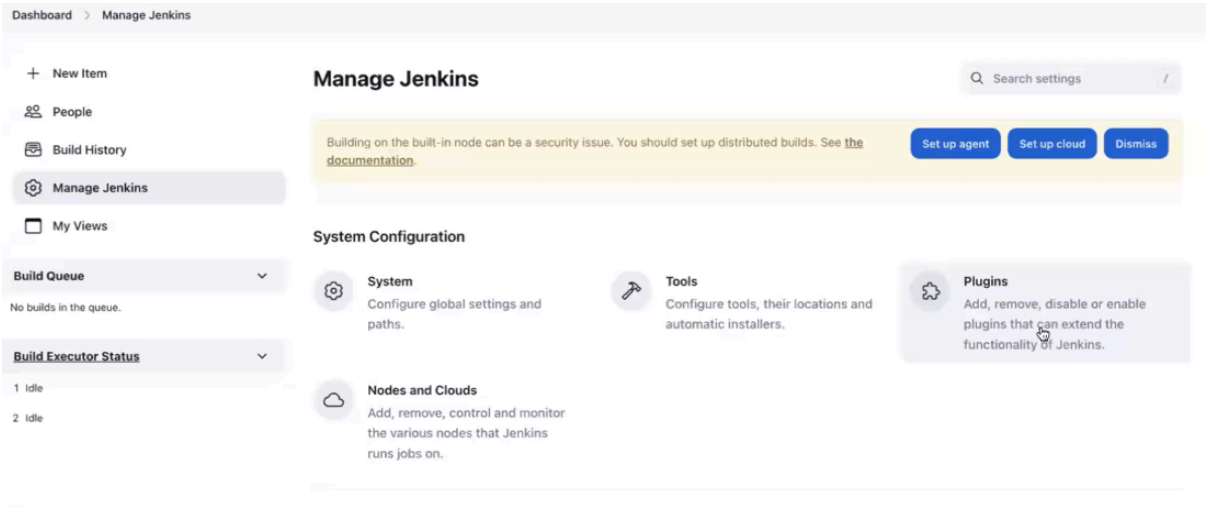
Step14:- Now install Ansible on RedHat server

$ sudo dnf install -y ansible-core

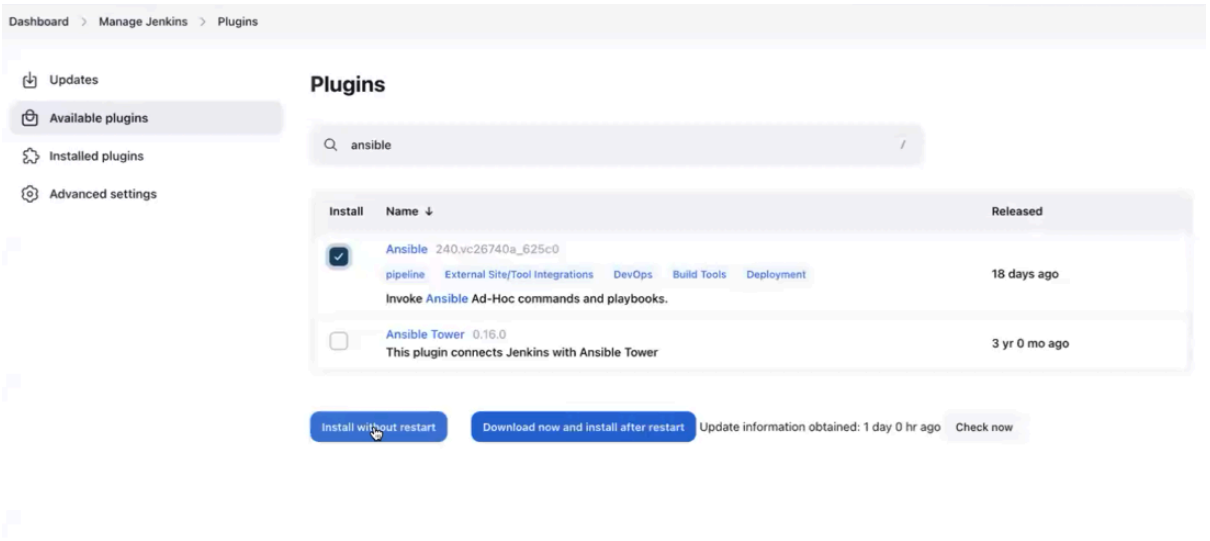
```
sudo dnf install -y ansible-core
```

Step 15:- Check whether Ansible is installed or not.
$ansible - - version

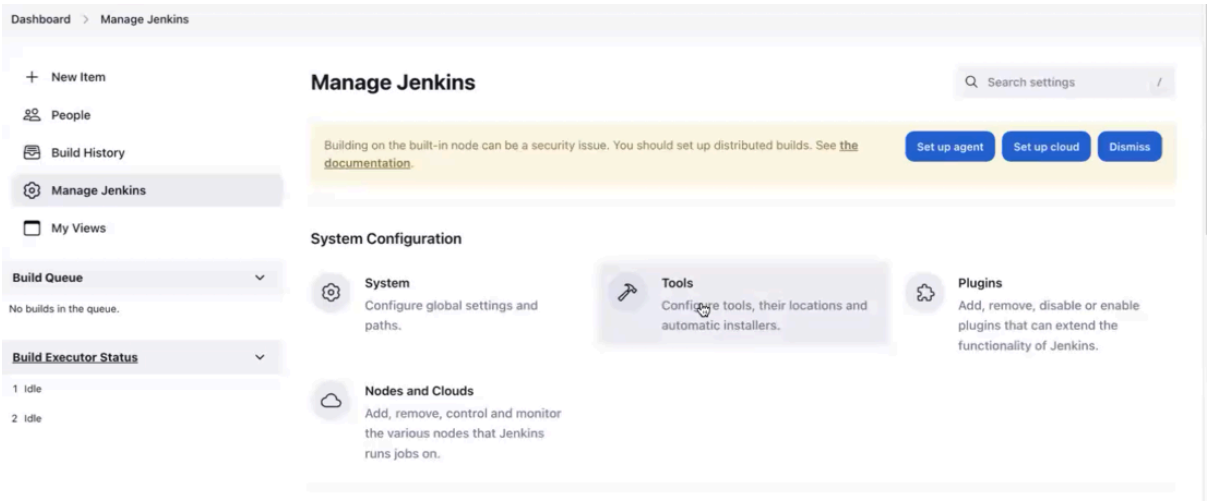
Step 16:- Now we have to tell Jenkins that ansible is installed on which server
Dashboard -> Manage Jenkins -> Plugins



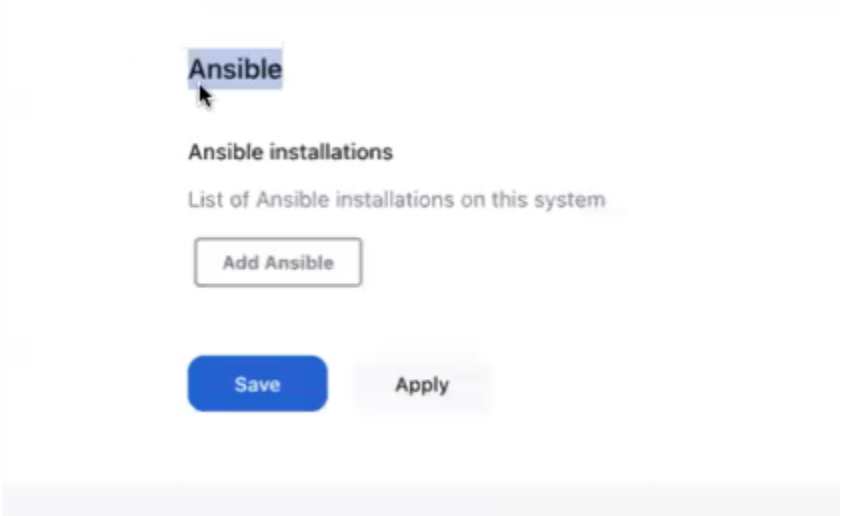
Step 17:- Select Available plugins -> Then search ansible -> Select ansible checkbox ->Press Install without restart



Step 18:- Now go to Dashboard -> tools



Step 19:- Scroll down and select Ansible and click Add Ansible



Step 20:- Write Name and path

Path:- When you install Ansible, you will get it. Just copy that

```
Complete!
[ec2-user@ip-172-31-93-153 ~]$ ansible --version
ansible [core 2.14.2]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.11/site-packages/ansible
  ansible collection location = /home/ec2-user/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, May 24 2023, 00:00:00) [GCC 11.3.1 20221121 (Red Hat 11.3.1-4)] (/usr/bin/python3.11)
  jinja version = 3.1.2
  libyaml = True
[ec2-user@ip-172-31-93-153 ~]$
```

Step 21:- Now create one file test.yml file for the tomcat9 configuration in github

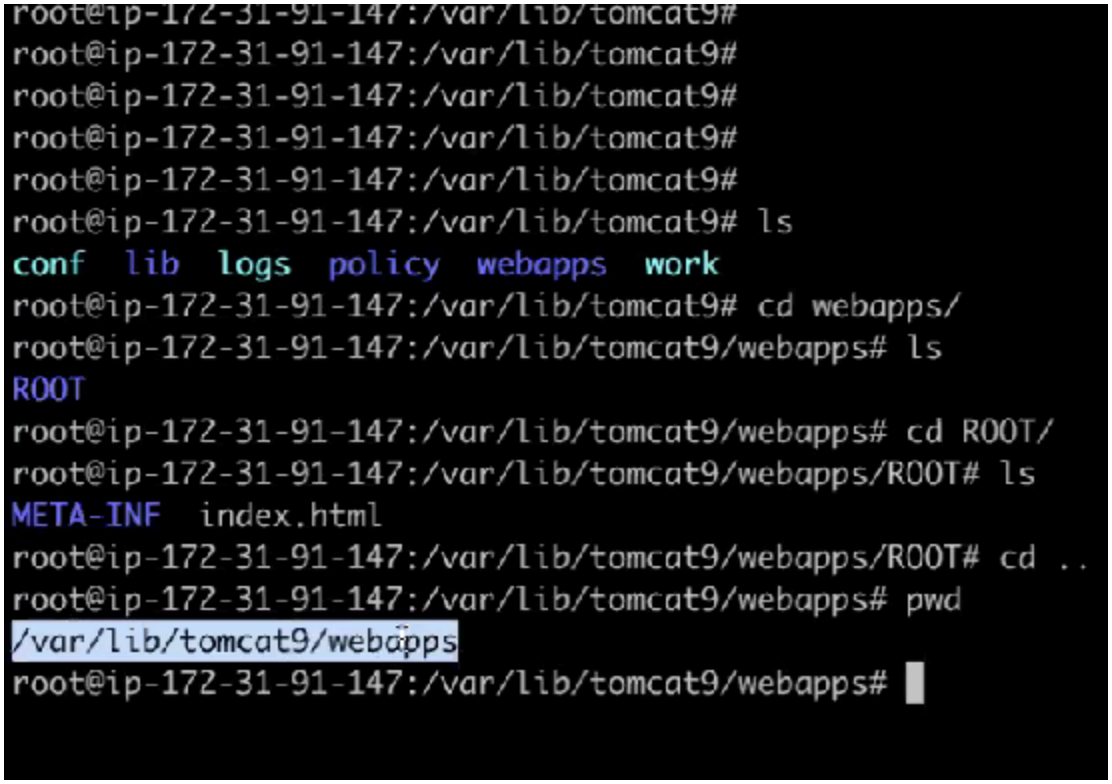
```
- name: Install Tomcat9
  hosts: test
  become: true
  tasks:
    - name: Installing Tomcat on worker nodes
      apt:
        name: tomcat9
        state: present
        update_cache: yes
    - name: Copying war file to webapps
      copy:
        src: /var/lib/jenkins/workspace/job3/target/maven-web-application.war
        dest: /var/lib/tomcat9/webapps
```

Path for source

Go to Jenkins ->select 2nd job-> Go to any of the executed job ->scroll down

```
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/2nd-job/src/main/resources
[INFO]
[INFO] --- compiler:3.3:compile (default-compile) @ maven-web-application ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/2nd-job/target/classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ maven-web-application ---
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/2nd-job/src/test/resources
[INFO]
[INFO] --- compiler:3.3:testCompile (default-testCompile) @ maven-web-application ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ maven-web-application ---
[INFO] No tests to run.
[INFO]
[INFO] --- war:3.3.2:war (default-war) @ maven-web-application ---
[INFO] Packaging webapp
[INFO] Assembling webapp [maven-web-application] in [/var/lib/jenkins/workspace/2nd-job/target/maven-web-application]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/2nd-job/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/2nd-job/target/maven-web-application.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.014 s
[INFO] Finished at: 2023-07-09T04:06:26Z
[INFO] ------------------------------------------------------------------------
[2nd-job] $ /usr/bin/ansible-playbook test.yml -i dep.inv -f 5 --private-key /tmp/ssh6376906226384497495.key -u ubuntu
```

Path for destination



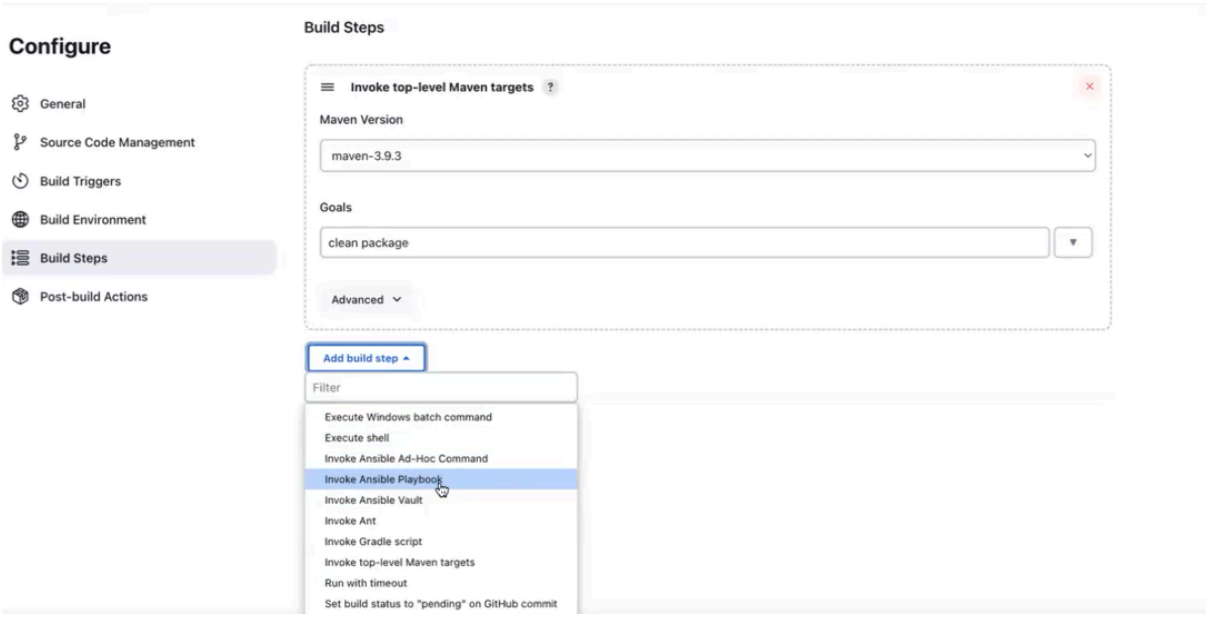
Step 22:- Create dep.inv file for inventory configuration in github

[test]
172.31.42.90 ansible_user=ubuntu
172.31.44.51 ansible_user=ubuntu

Step 23:-Now add both the files in the GitHub repo and commit it.

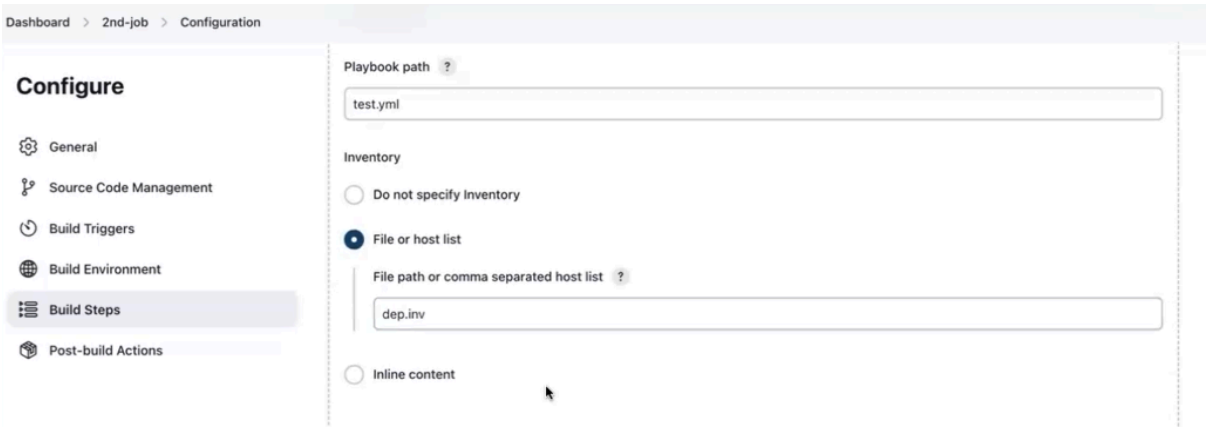
Step 24:- Now go to Jenkins and go to the 2nd job and refresh the website

Step 25:- Now go to Dashboard->2nd job-> configure -> Build steps ->Click on dropdown ->Click invoke ansible playbook

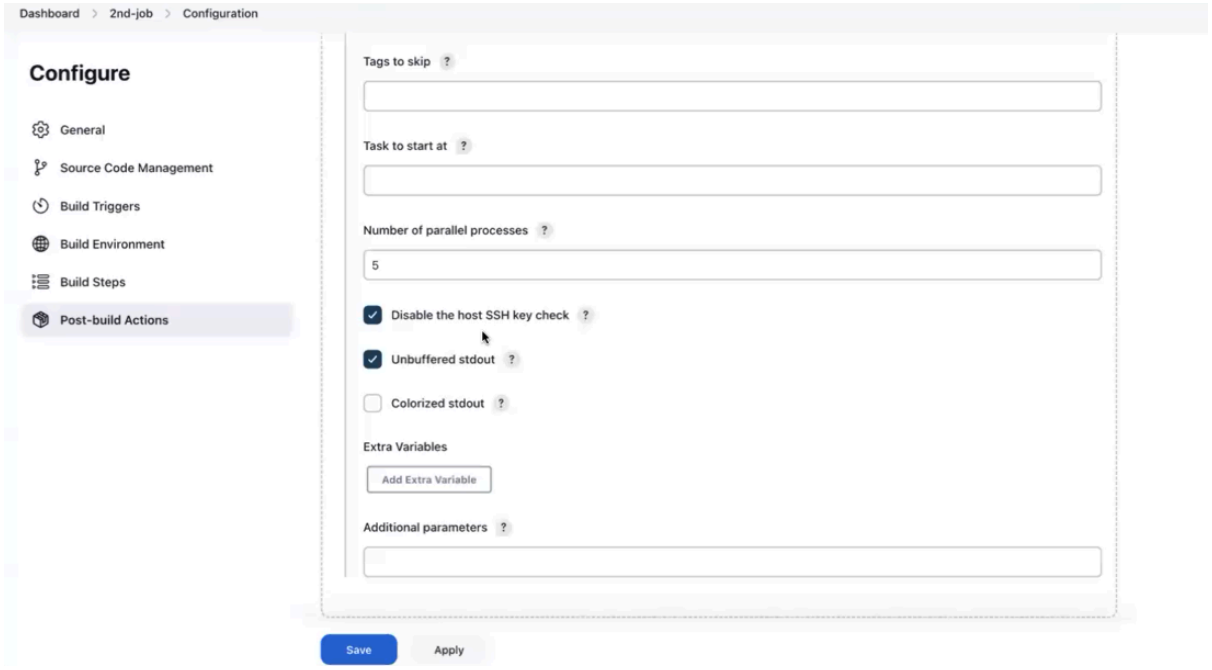


Step 26:- Now give
Playbook path = test.yml
Inventory= File or host list -> dep.inv

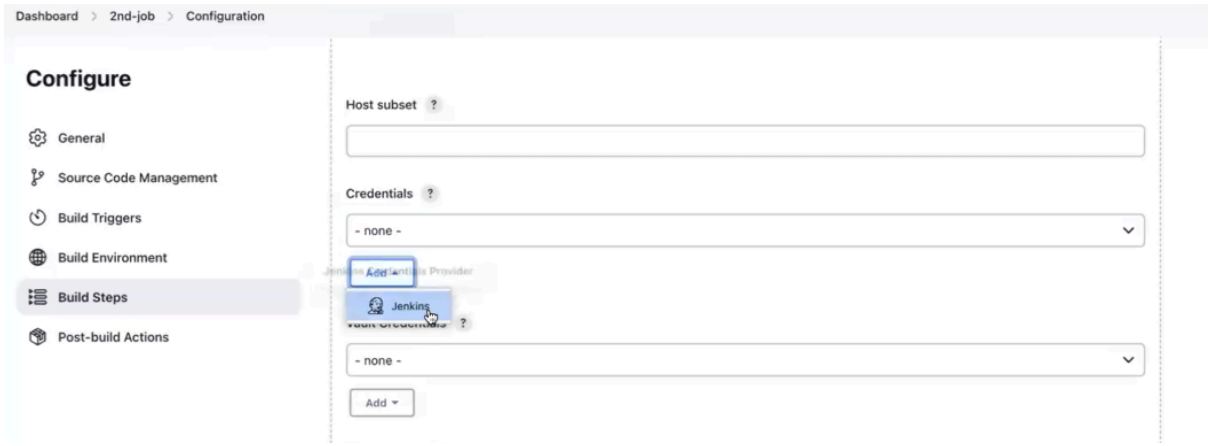
*****You can see the clone files in Dashboard -> 2nd job -> Workspace**



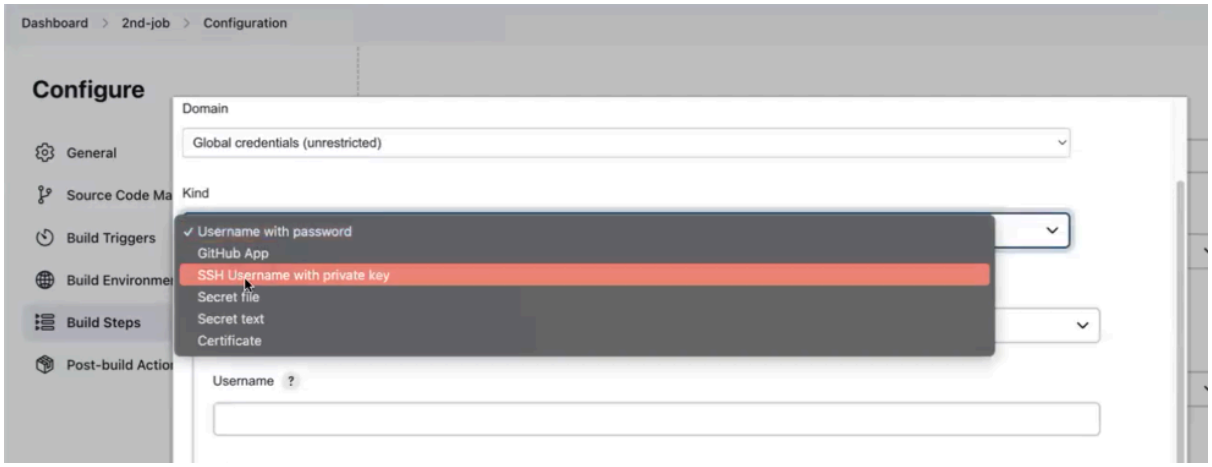
Step 27:- Now scroll down in Advance -> select Disable the host SSH key check



Step 28:- Now scroll up under credentials -> Press Add -> Select Jenkins



Step 29:- Now under Kind ->select SSH Username with private key

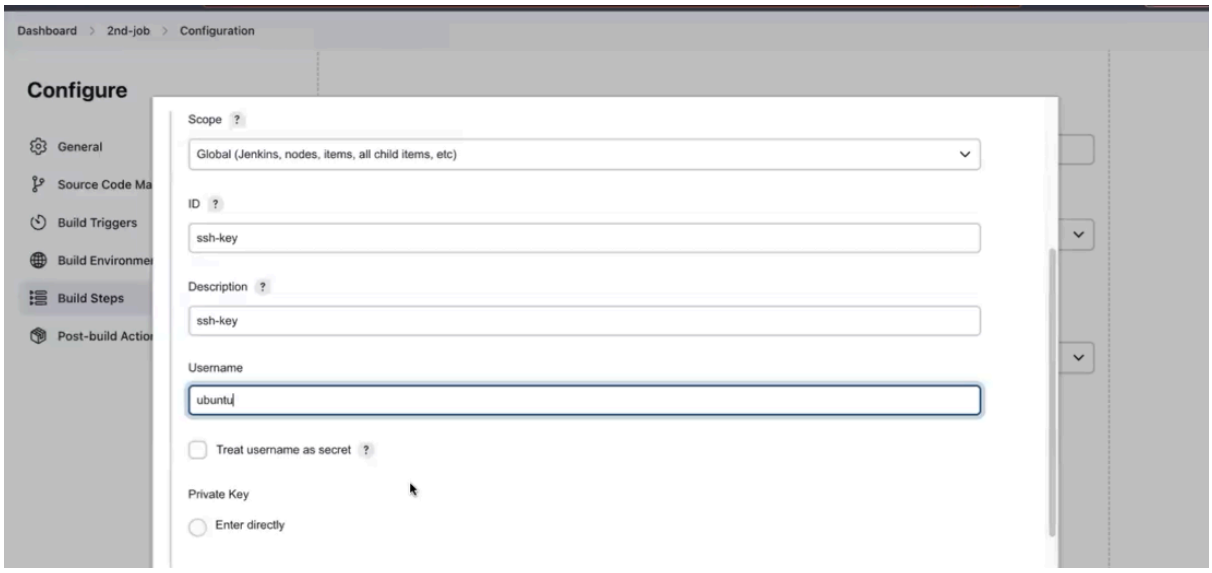


Step 30:- Now give

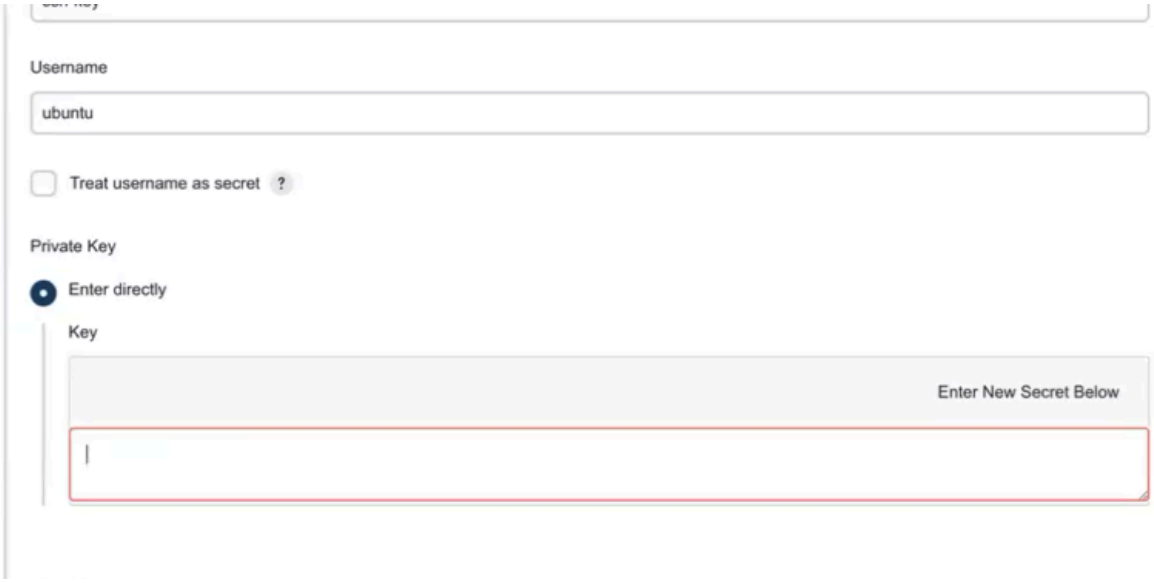
ID=ssh-key … can be random

Description=ssh-key … can be random

Username=ubuntu … this is fix for ubuntu

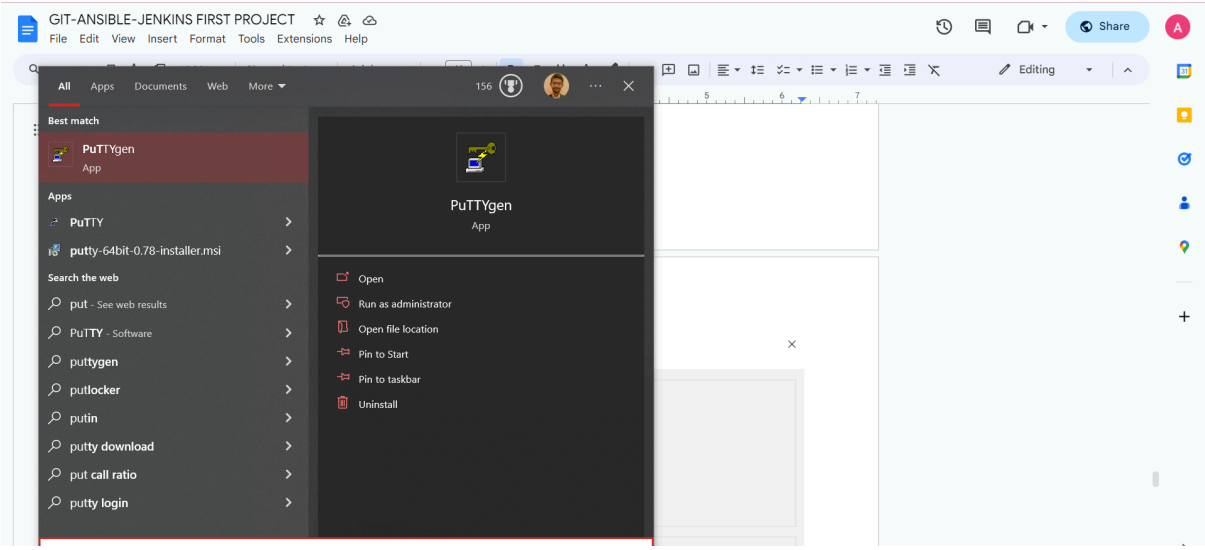


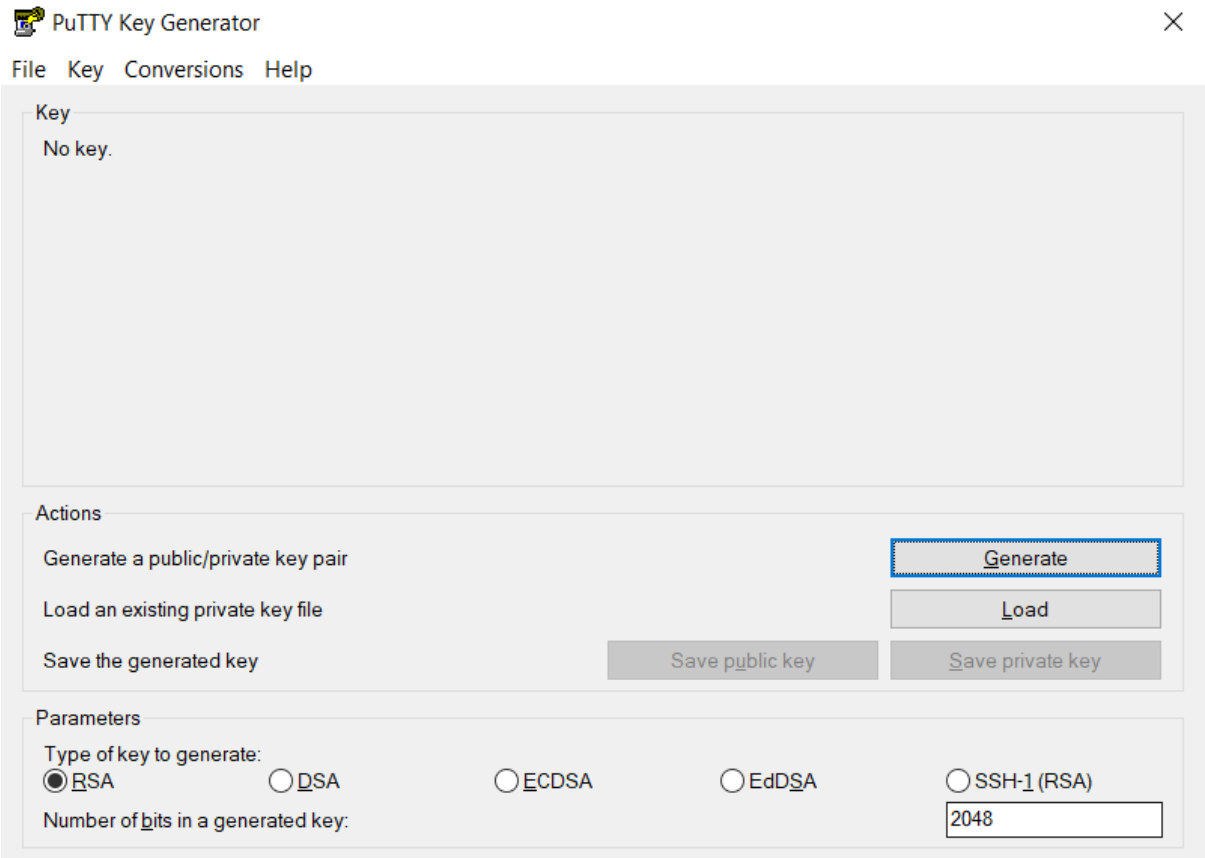
Step 31:- Now select Private Key-> Enter directly



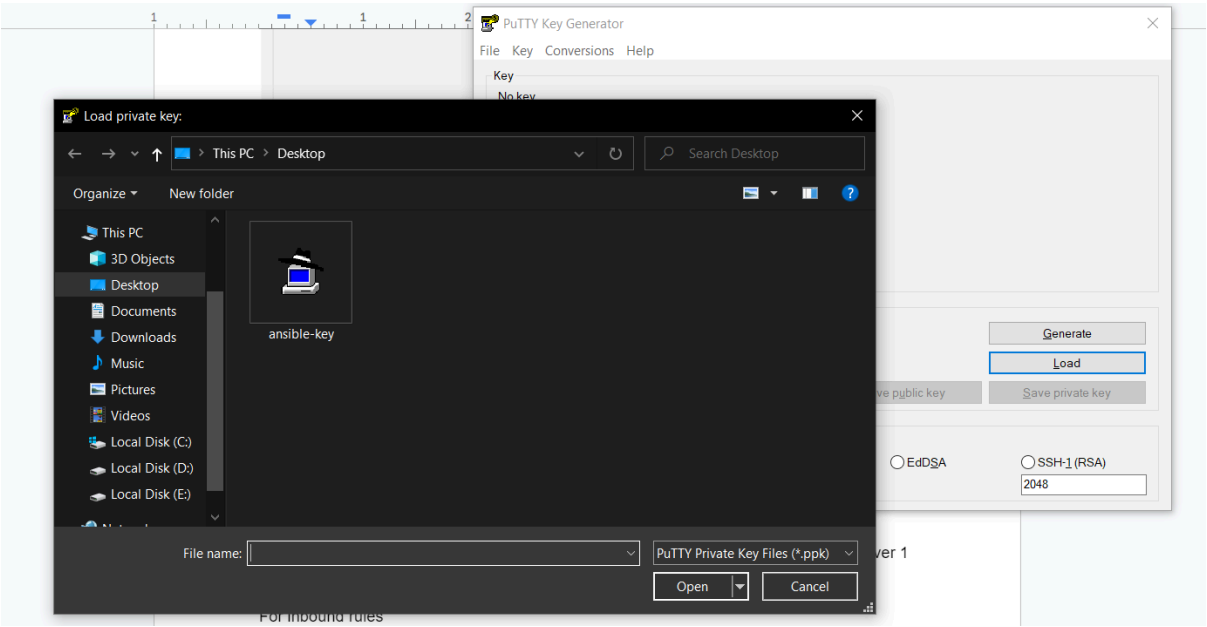
Now in the key section, you need to add the private key only from pem file.

1. **Open puttyGen**





2. **Click on load and select the ppk file**

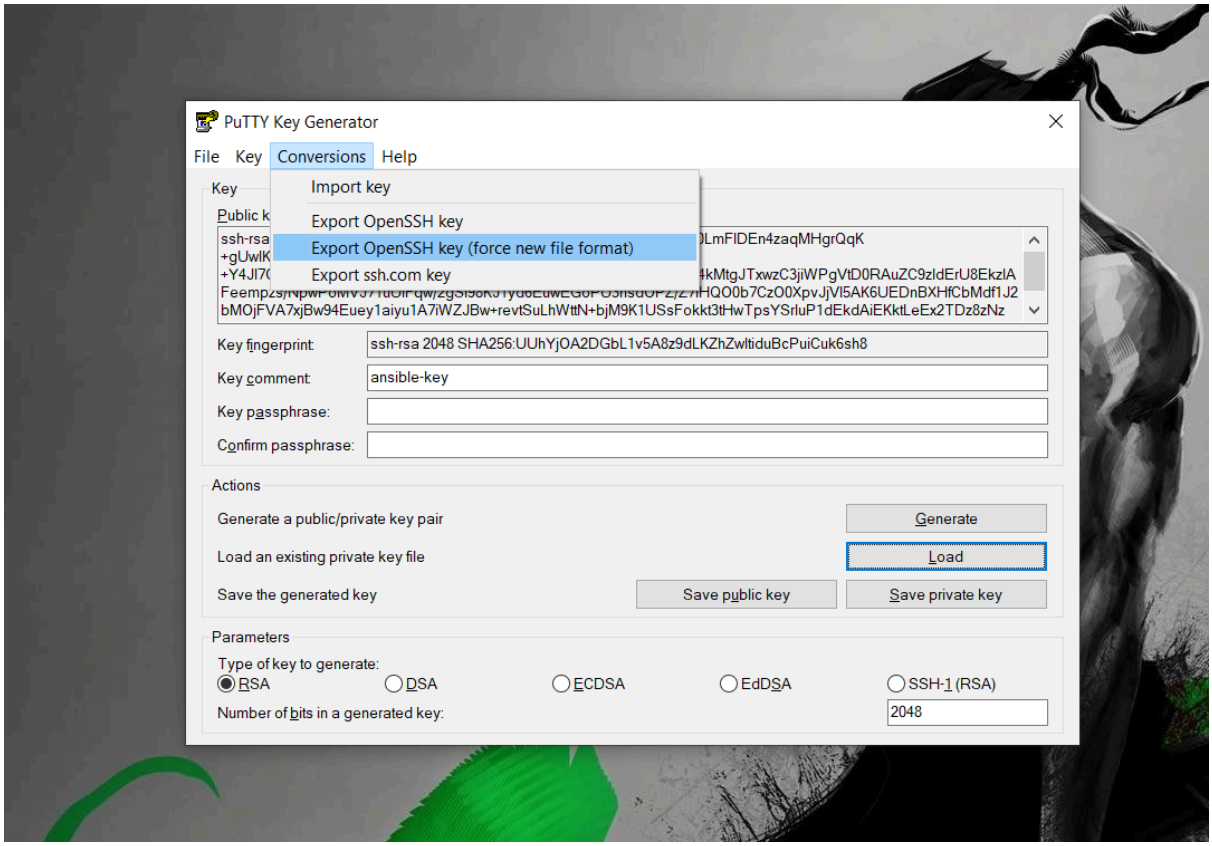


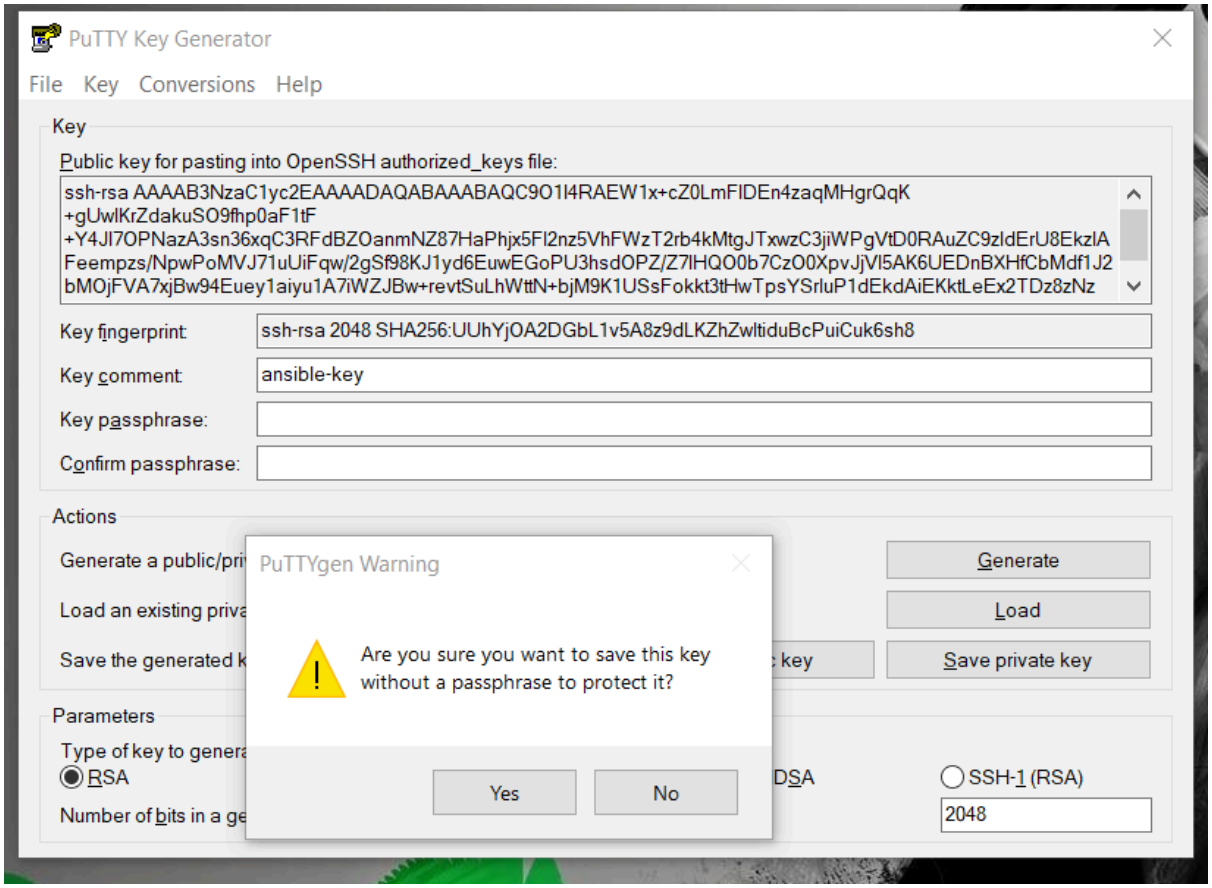
Ata he ppk file (key) tya servers chi pahije je server ch ip id apan dep.inv madhe takla ahe, means

[test]
172.31.42.90 ansible_user=ubuntu
172.31.44.51 ansible_user=ubuntu

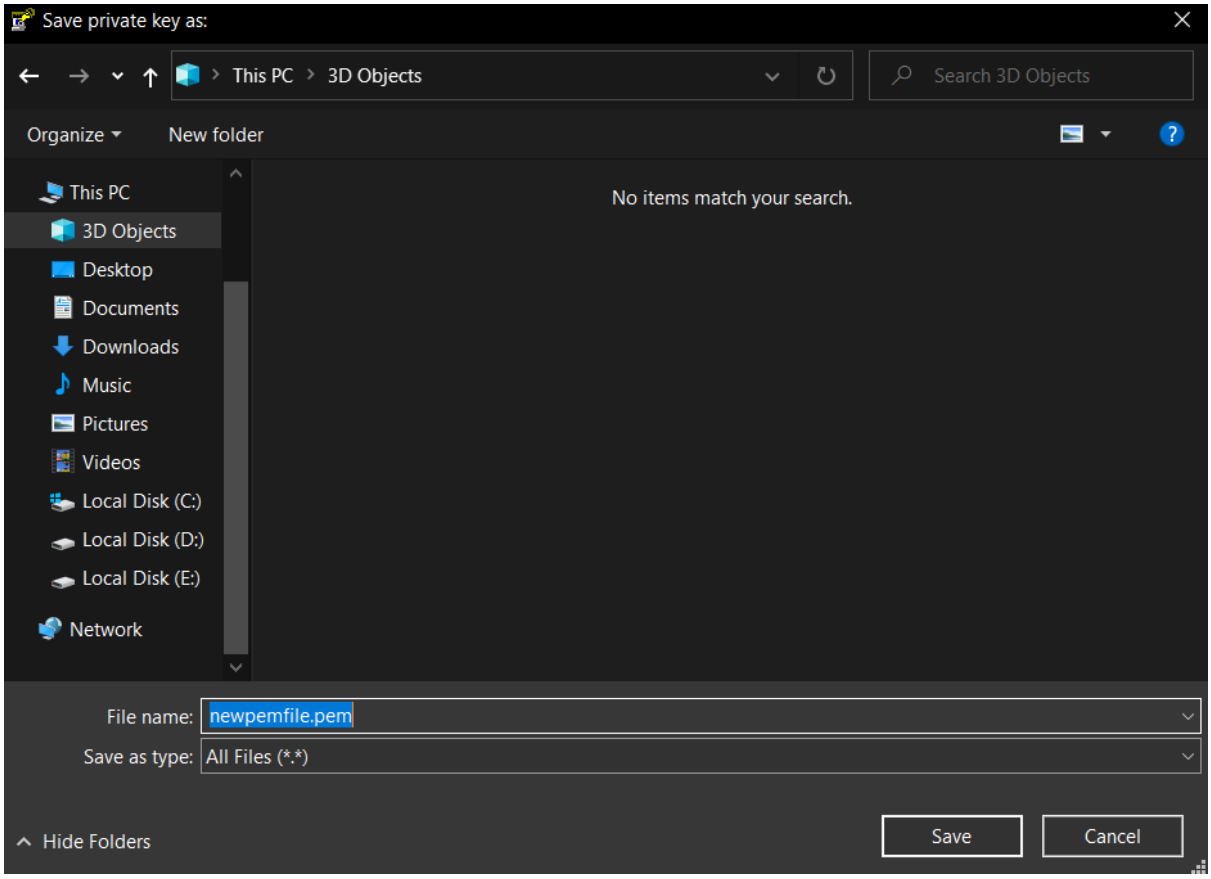
Yat je ip use kele ahe, tyanchya instance sathi ji ppk (key) use keli ahe tichich private key aplyala takayachi ahe. Mhanun tich ppk (key) load kara putty gen madhe karan aplyala tyach ppk la pem madhe conversion dyayache ahe.

3. **Conversions->Export OpenSSH key (force new file format)**


PuTTY Key Generator
File
Key
Conversions
Help
Import key
Export OpenSSH key
Export OpenSSH key (force new file format)
Export ssh.com key
Key
Public k
ssh-rsa
+gUwlK
+Y4JI7(
LmFlDEn4zaqMHgrQqK
4kMtgJTxwzC3jiWPgVtD0RAuZC9zldErU8EkzlA
HQO0b7CzO0XpvJjVl5AK6UEDnBXHfCbMdf1J2
bMOjFVA7xjBw94Euey1aiyu1A7iWZJBw+revtSuLhWttN+bjM9K1USsFokkt3tHwTpsYSrluP1dEkdAiEKktLeEx2TDz8zNz
Key fingerprint:
ssh-rsa 2048 SHA256:UUhYjOA2DGbL1v5A8z9dLKZhZwltiduBcPuiCuk6sh8
Key comment:
ansible-key
Key passphrase:
Confirm passphrase:
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC9O1I4RAEW1x+cZ0LmFIDEn4zaqMHgrQqK
+gUwlKrZdakuSO9fhp0aF1tF
+Y4JI7OPNazA3sn36xqC3RFdBZOanmNZ87HaPhjx5FI2nz5VhFWzT2rb4kMtgJTxwzC3jiWPgVtD0RAuZC9zIdErU8EkzIA
Feempzs/NpwPoMVJ71uUiFqw/2gSf98KJ1yd6EuwEGoPU3hsdOPZ/Z7lHQO0b7CzO0XpvJjVl5AK6UEDnBXHfCbMdf1J2
bMOjFVA7xjBw94Euey1aiyu1A7iWZJBw+revtSuLhWttN+bjM9K1USsFokkt3tHwTpsYSrluP1dEkdAiEKktLeEx2TDz8zNz
Key fingerprint:
ssh-rsa 2048 SHA256:UUhYjOA2DGbL1v5A8z9dLKZhZwltiduBcPuiCuk6sh8
Key comment:
ansible-key
Key passphrase:
Confirm passphrase:
Actions
Generate a public/pri
Generate
Load an existing priva
Load
Save the generated k
key
Save private key
Parameters
Type of key to genera
RSA
DSA
SSH-1 (RSA)
Number of bits in a ge
2048
PuTTYgen Warning
Are you sure you want to save this key without a passphrase to protect it?
Yes
No

Click on yes.



Now save the file with, file_Name.pem

4. **Open that pem file in notepad**

pemfileeee.pem - Notepad

File Edit Format View Help

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdz
c2gtcnNhAAAAAwEAAQAAAQEAvTtSOEQBFtcfnGdC5hZQxJ+M2qjB4K0KivoFMJSq
2XWpLkjvX4adGhdbRfmOCZezjzWswN7J9+sagt0RXQWTmp5jWfOx2j4Y8eRSNp8+
VYRVs09q2+JDLYCU8cMwt44lj4FbQ9EQLmQvcyHRK1PBJM5QBXnpqc7PzacD6DFS
e9blIhasP9oEn/fCidcnehLsBBqD1N4bHTj2f2e5R0DtG+wsztF6byY1ZeQCulBA
5wVx3wmzHX9SdmzDoxVQO8YwcPeBLnstWosrtQO4lmSQcPq3r7Uri4VrbTfm4zPS
tVErBaJJLd7R8E6bGEq5bj9XRJHQIhCpLS3hMdkw8/MzcwAAA9AFoML8BaDC/AAA
AAdzc2gtcnNhAAABAQC9O1I4RAEW1x+cZ0LmFlDEn4zaqMHgrQqK+gUwlKrZdaku
SO9fhp0aF1tF+Y4Jl7OPNazA3sn36xqC3RFdBZOanmNZ87HaPhjx5FI2nz5VhFWz
T2rb4kMtgJTxwzC3jiWPgVtD0RAuZC9zIdErU8EkzlAFeempzs/NpwPoMVJ71uUi
Fqw/2gSf98KJ1yd6EuwEGoPU3hsdOPZ/Z7lHQO0b7CzO0XpvJjVl5AK6UEDnBXHf
CbMdf1J2bMOjFVA7xjBw94Euey1aiyu1A7iWZJBw+revtSuLhWttN+bjM9K1USsF
okkt3tHwTpsYSrluP1dEkdAiEKktLeEx2TDz8zNzAAAAAwEAAQAAAQBAm8huvYlB
qyF33SnSNTxOZctJvOS1OV+oI8Ux2RCcPrgMvYo9K1DoX6EiOxpr3AfKlaDuGSMM
KpFarPAIuqay2/E//Rx0T09lB3eqfqRhvlUka7EochBNsOEpymMPyNC6Nohalz7X
GA27WYh5Kpd/YZGgzQcimyIILUYAORbYgOcHH6XQp++L8EcGlk+OqW4NmRlVHe7Y
S/wNiirKJ+KVyUJfXPvH7DGTpUTw50nN8l8GX+NV/IT5X/JdvQdAWoCuarf8wRZz
TTiKQgZZyGZmE8wyivBe4VeHqZk0zpoVs421aVzf3luytPb4e7JT5p+owcLvZ5Ow
CQ9O0KxlpYlBAAAAgEI4aXt9cwLXm2ndD1UmGjP+E9up1GHI4tCyrbilzwsMm9cU
Bl4YkkCBtOw7rHKqI+5Jrt9aj14uhHssaRDoNb050Ilfp794jz8cI4CO3YR5T6Mv
983IVWxWij6vF+ZvuPBnNo/h9+RK8LKQGQukdZtLQHubblxucnva4jEa5w3gAAAA
gQD7jGaOlb+7HBXAZgyvZiawmf+Qu1MVnSvw20vgTZPLbtdjJNVWawVtFUx5SMbc
B0W3s++omqNl8NIRohO128cAGNKx3Tc6kjRta5/IjhglQjNOprKYUVQl/o71Uxhc
rwBvJHmjYdedfvZvmH1qklF+jgVSRsRN1c/YeL8SM1ti7wAAAIEAwJSa0z/1IAJf
6kOsMelaApZp8wdWcJtLHm1lbmOKwTvRmvpxt2jOwejnzI2tsvpV3ZSk9Xv/CUkM
gvxYZ5uPg7btLg202wbmq7Z19h8C67QbrigjlThXo5oH2McGg5znWQUMP4wydMYX
KrRmN4Y/0Ey6TeIiT4ntX08F2yIjZ70AAAALYW5zaWJsZS1rZXkBAgMEBQYHCAkK
CwwNDg8Q
-----END OPENSSH PRIVATE KEY-----
```

Now copy everything using ctr + a
And paste it into that private key, shown below.

Now click on Add.

Step 32:- Now go to the

GitHub repo and make some changes and commit it.
Come to Jenkins -> Go to 2nd job->Refresh

Step 33:- Now copy server1 or server2 public IP-> Open browser->paste public_ip:8080

Step 34:- Do $ls to check whether the application appears or not

Step 35:- Go to the browser
Write :- Public _ip_address:8080/maven-web-application/

Step 36:- You can change the things that will get displayed on the website
Step 37:- End