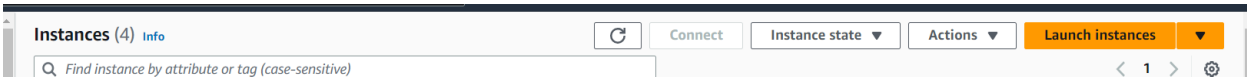
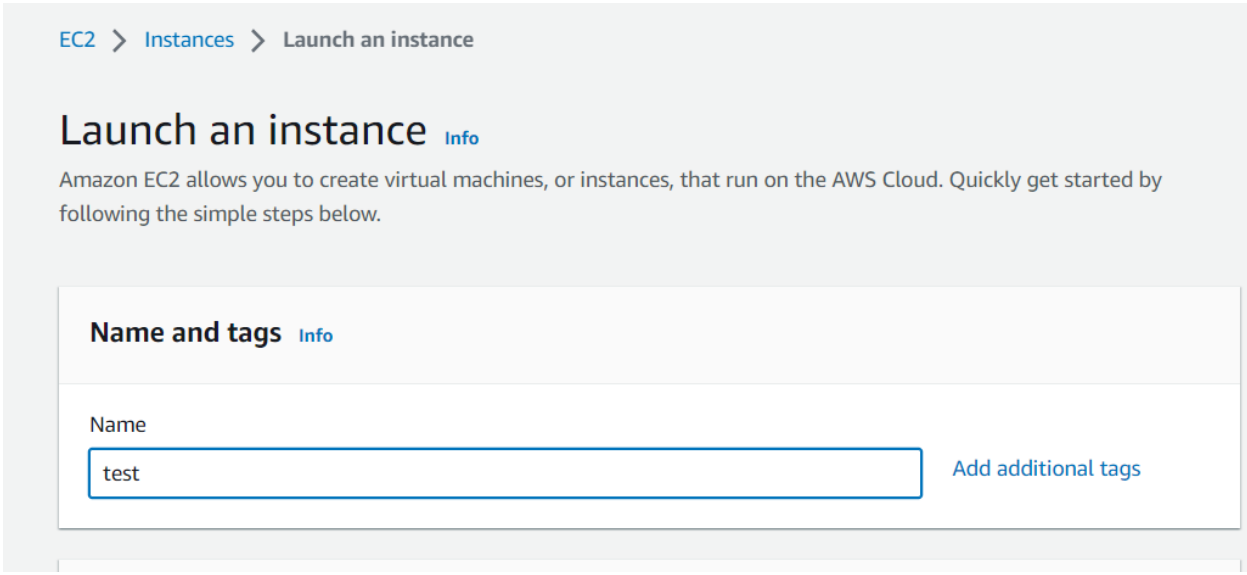
Day 11. Jenkins and Ansible Introduction with project

DEVOPS PROJECT DEPLOYMENT

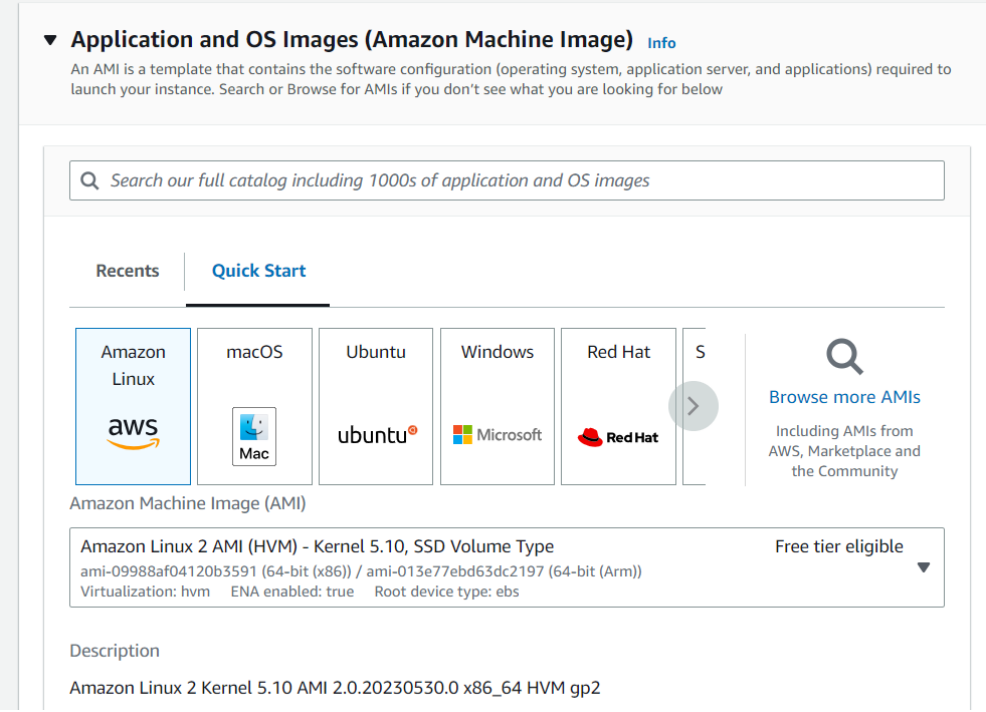
- CREATE 4 INSTANCES
  - ❖ Click on launch instances



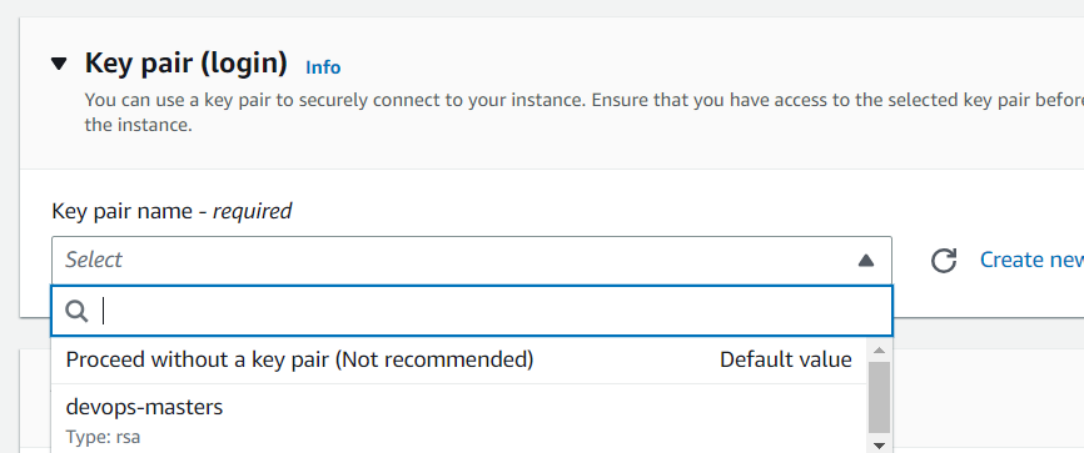
  - ❖ Write name of the instances



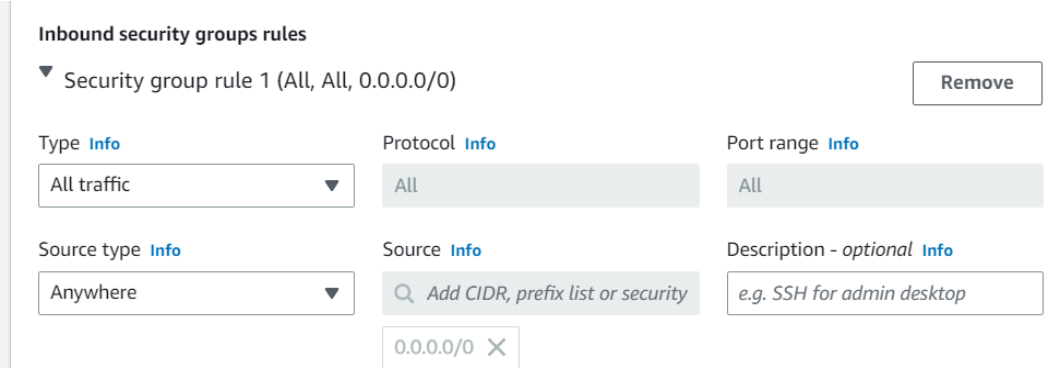
  - ❖ Select instance OS . here i have selected the amazon linux 2 OS



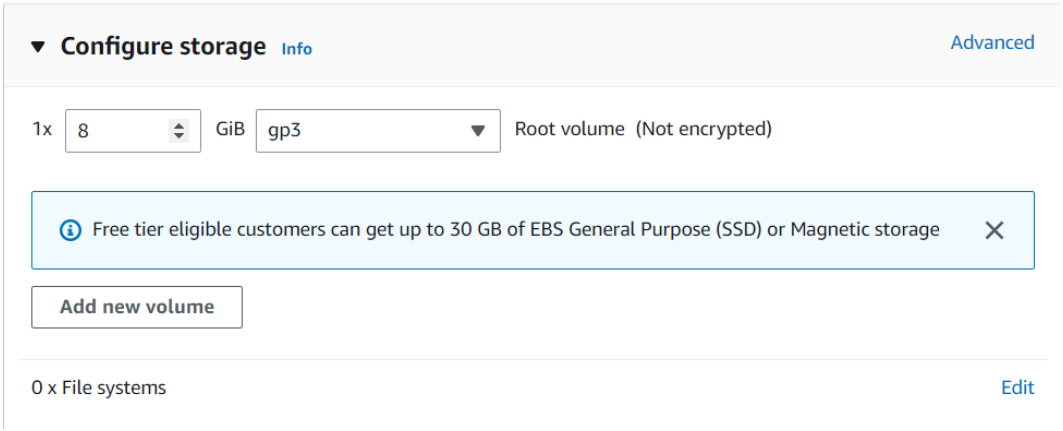
❖ Selecting key pair login



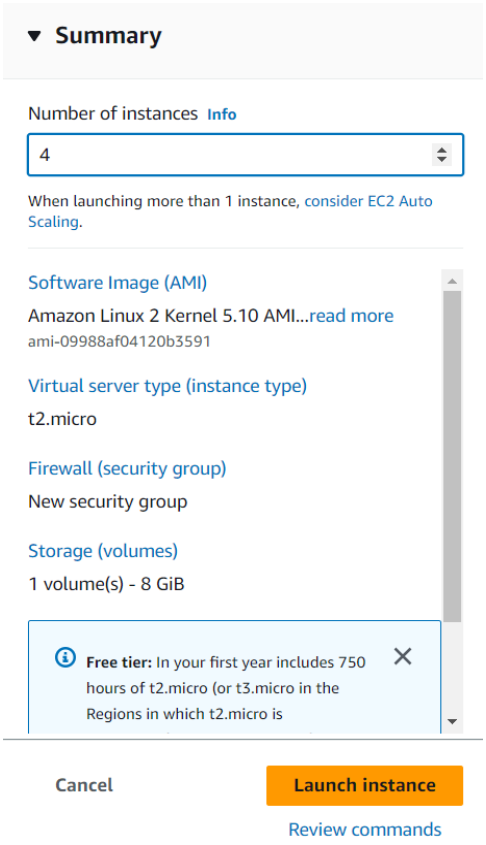
❖ Changing inbound security groups rules of network to all traffic , anywhere



- Configure storage

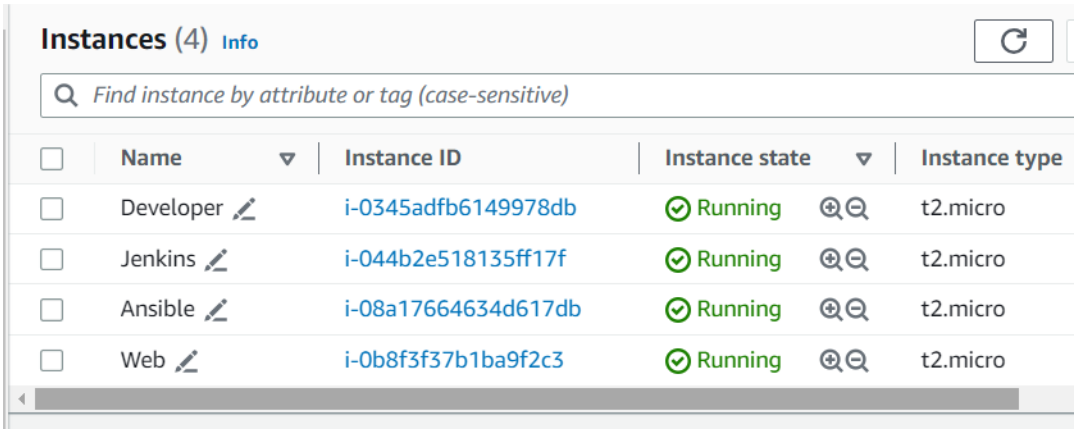


- Write the number of instance . In this case i have selected 4 instances and click on launch instances.

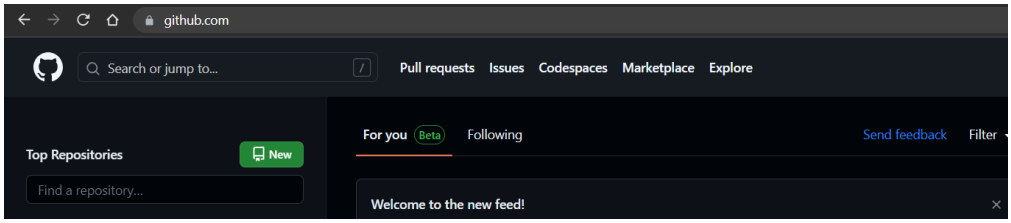


- A success messages will appear

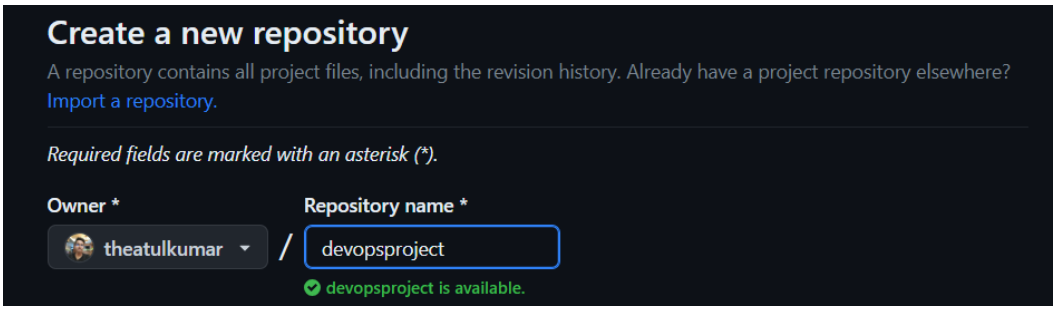
- ❖ Rename the instances



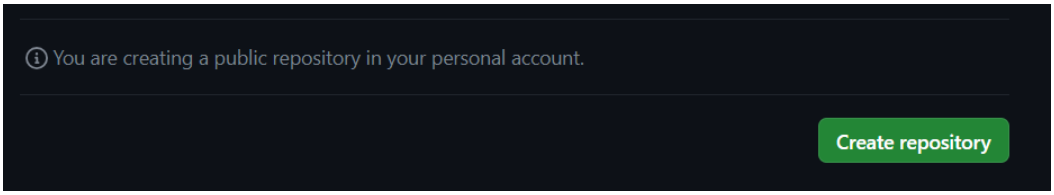
- Now login to github and create new repository



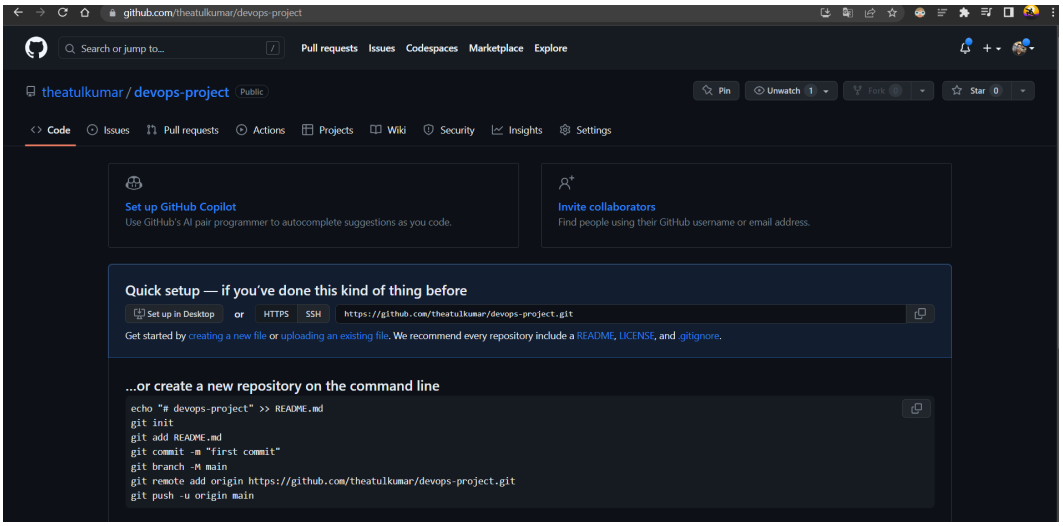
- ❖ Write repository name



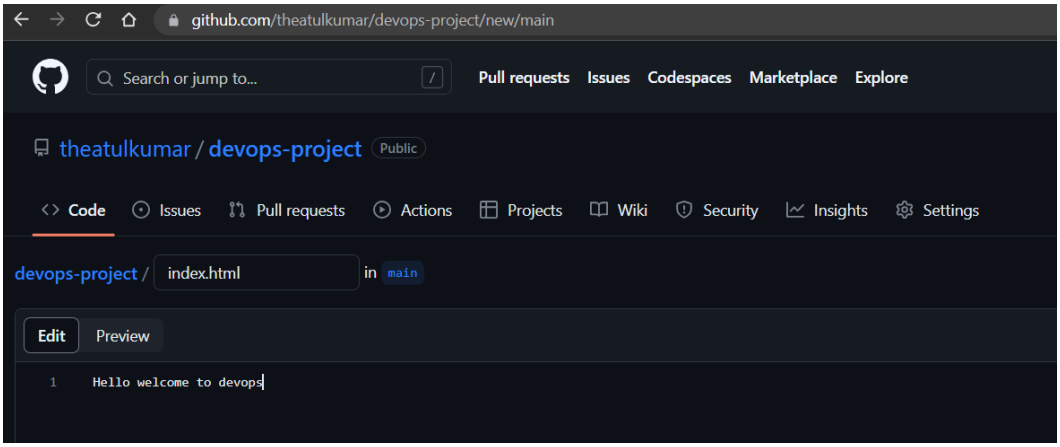
- ❖ Click on create repository



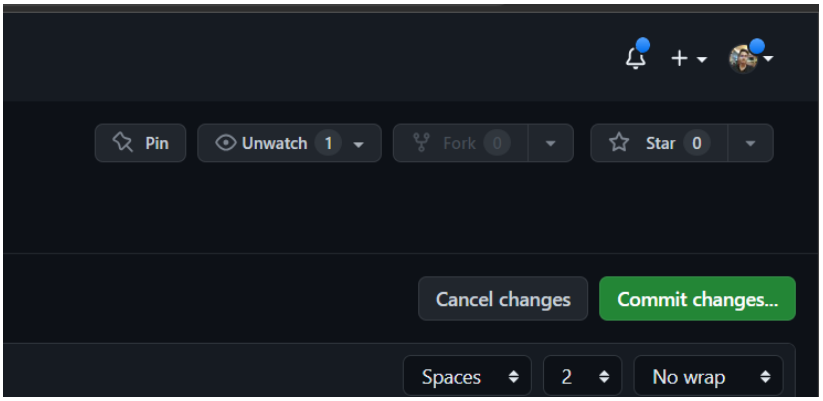
- ❖ This will appear after creating a repository. Now click on create a file(index.html).

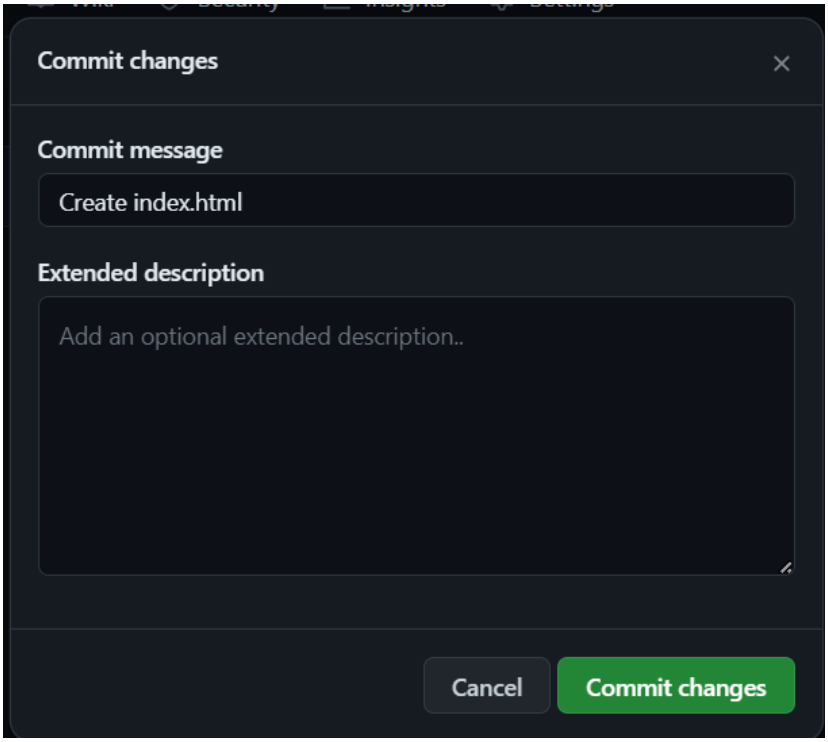


- Writing something in file like “Hello welcome to devops”

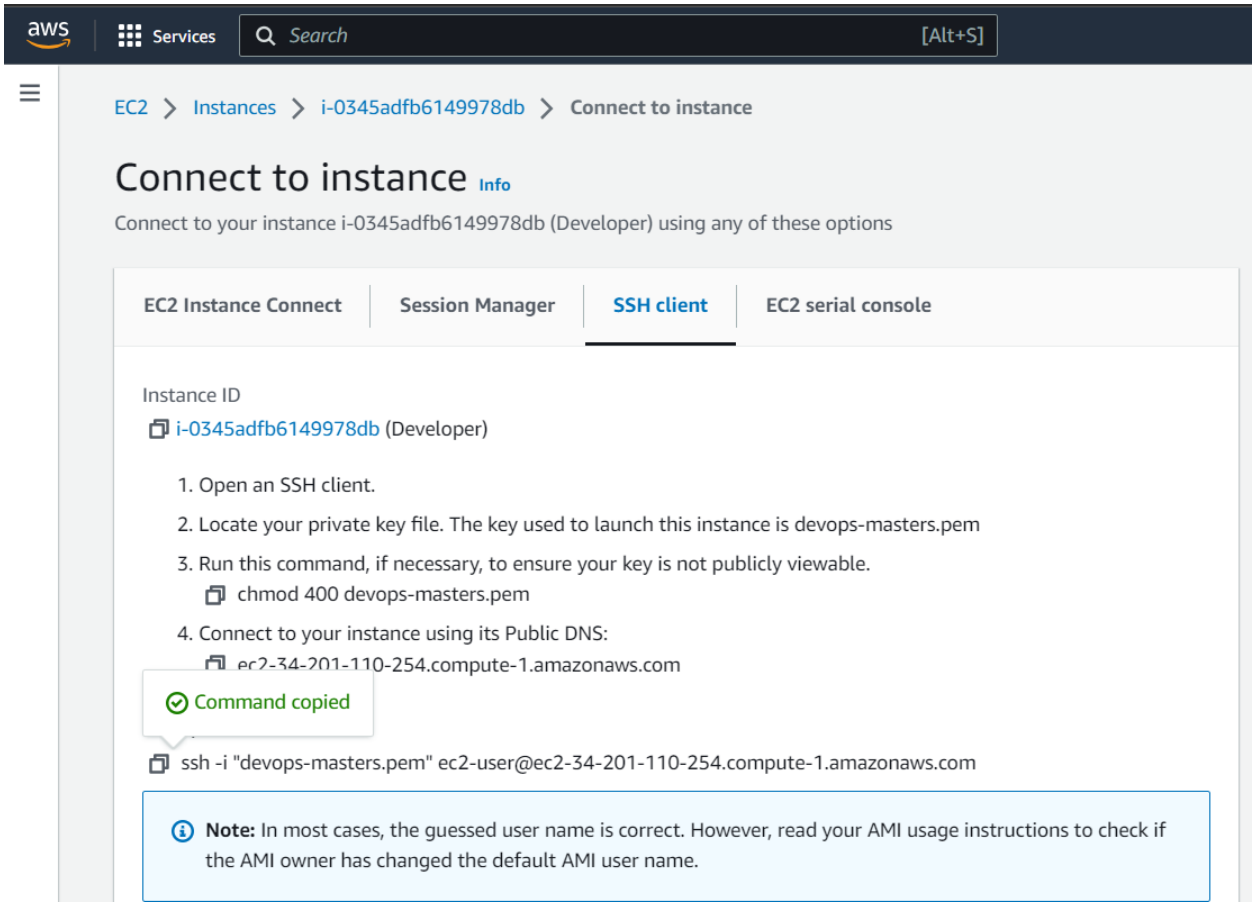


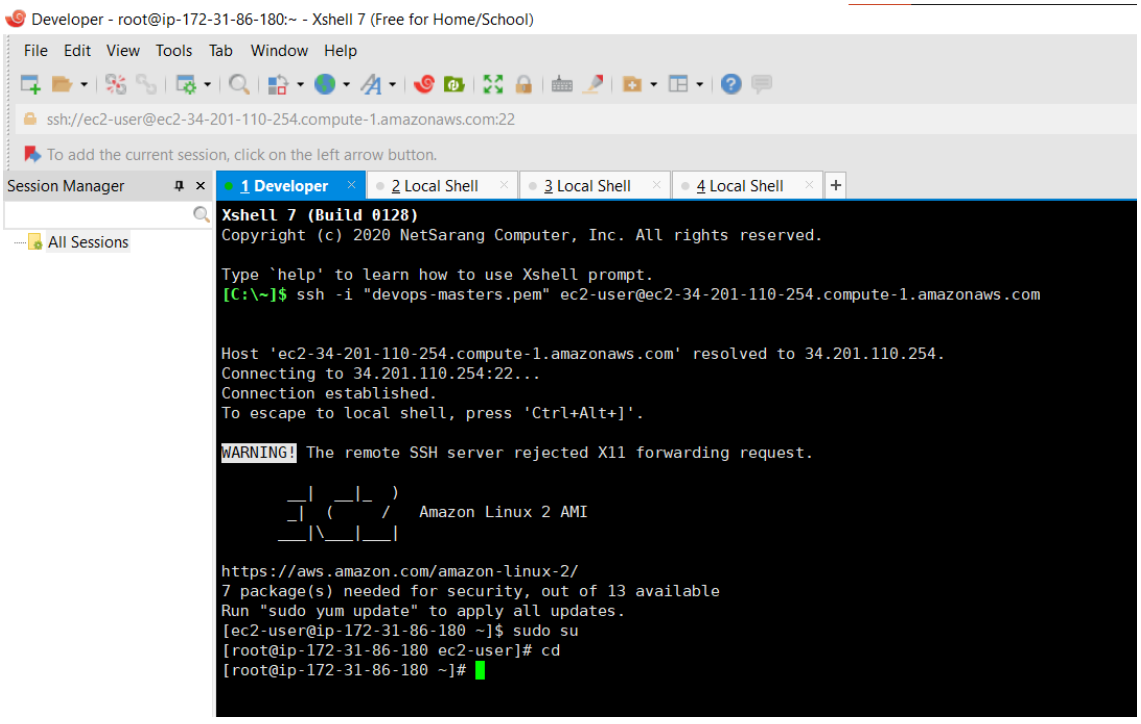
- Now click on commit changes





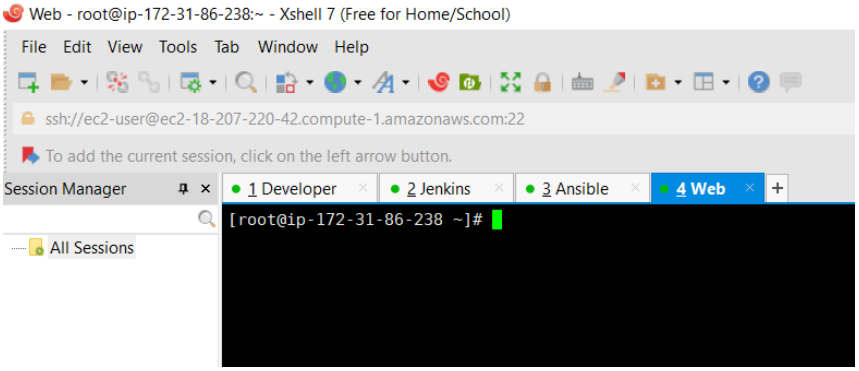
- Now open Xshell and connect all the instances.
  Copy the command and paste accordingly one by one to connect all the 4 instances





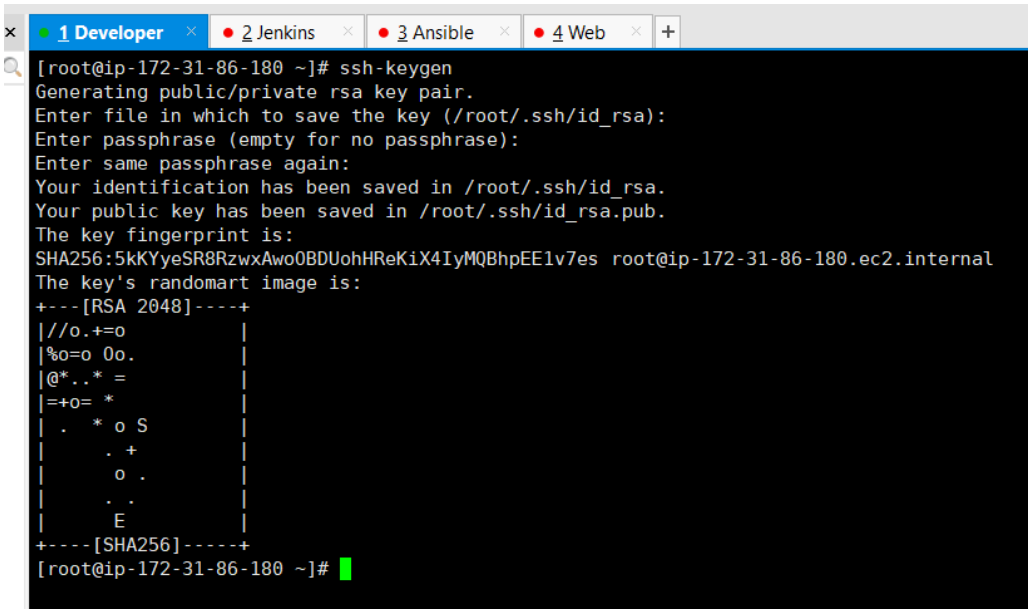
-
- Write a command after connecting to the server.
- sudo su (superuser do)

- ❖ cd (change directory)
- ❖ We can use Ctrl + L to clear the screen of Xshell.
- ❖ Connected each instances one by one



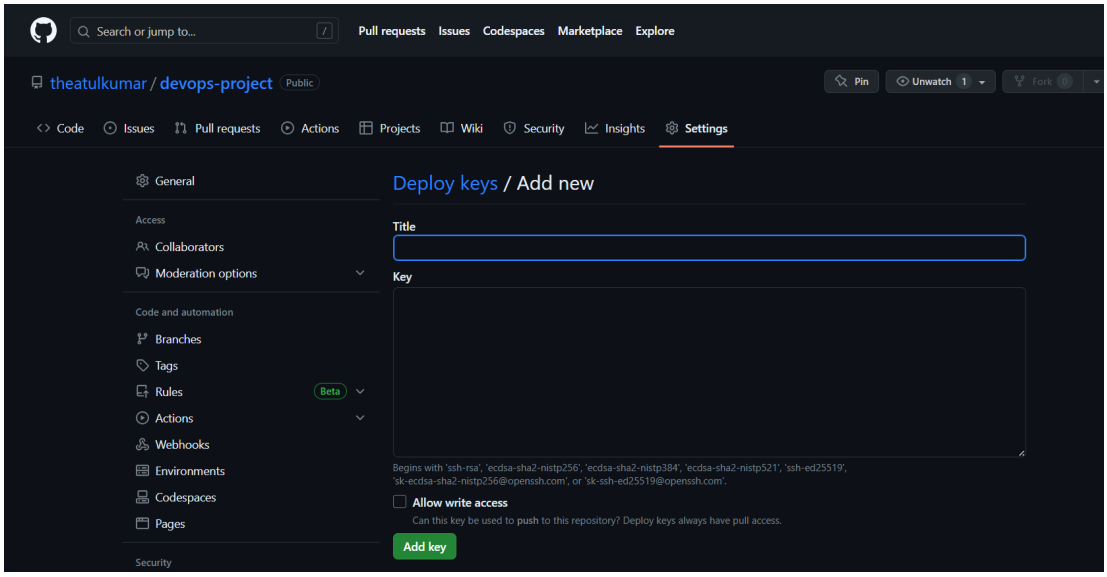
- ❖

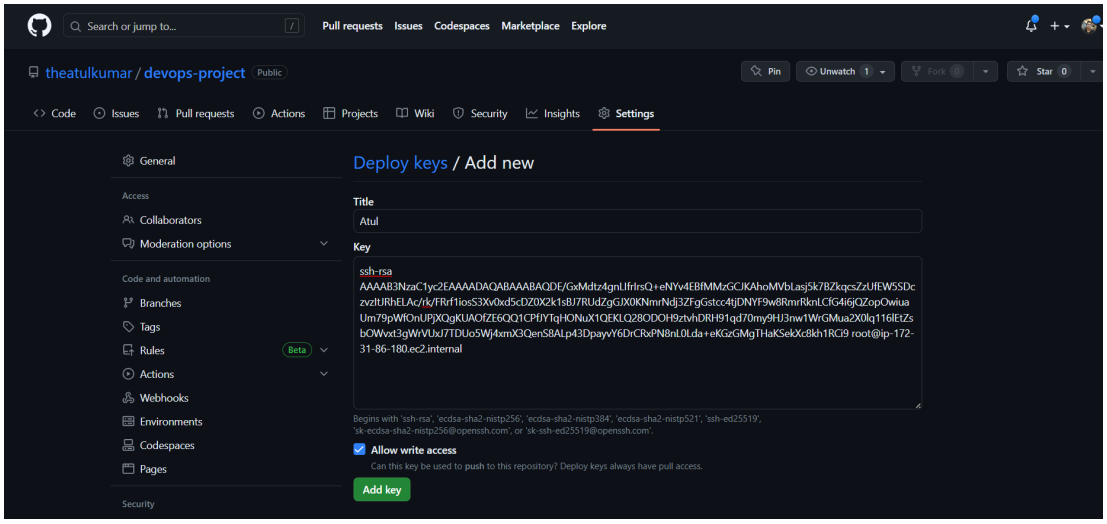
- Now we need to set up password less authentication from developer machine to github .
- We cannot give the id and password of github to developer that is why we need to do password less authentication.
- So we need to create a public key.
- Write the command “ssh-keygen”

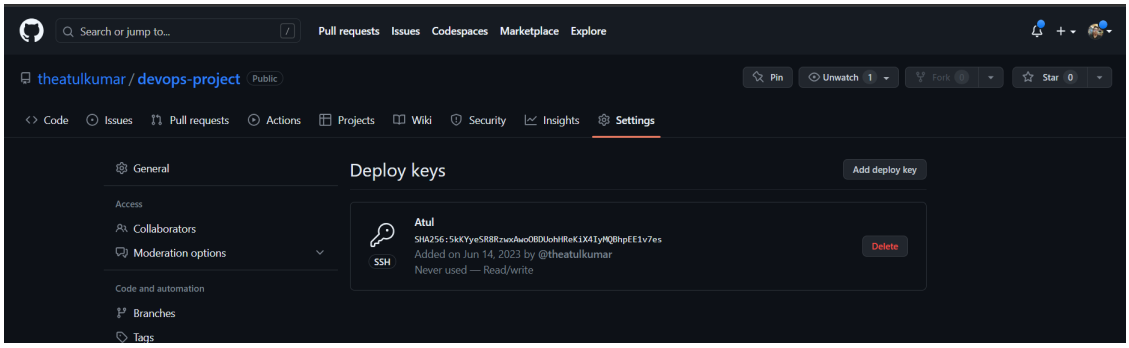


- Now use cat command to check the public key where it is stored

- Now we need to copy the content of the file.
- Then go to github click on settings then click on deploy keys. Add new

- 

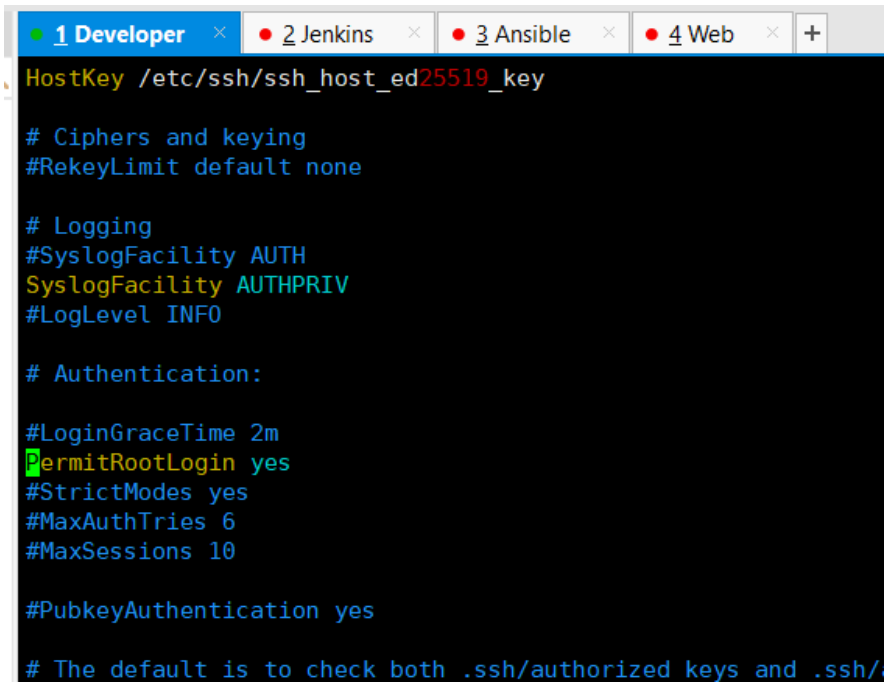
- 

- 

- Now change the root password by writing command “passwd root”
- I gave my password as root123 or AKpass23
- Then we need to enable password less authentication
- Write command “vi /etc/ssh/sshd_config”

```
[root@ip-172-31-86-180 ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-86-180 ~]#
```

- PermitRootLogin yes



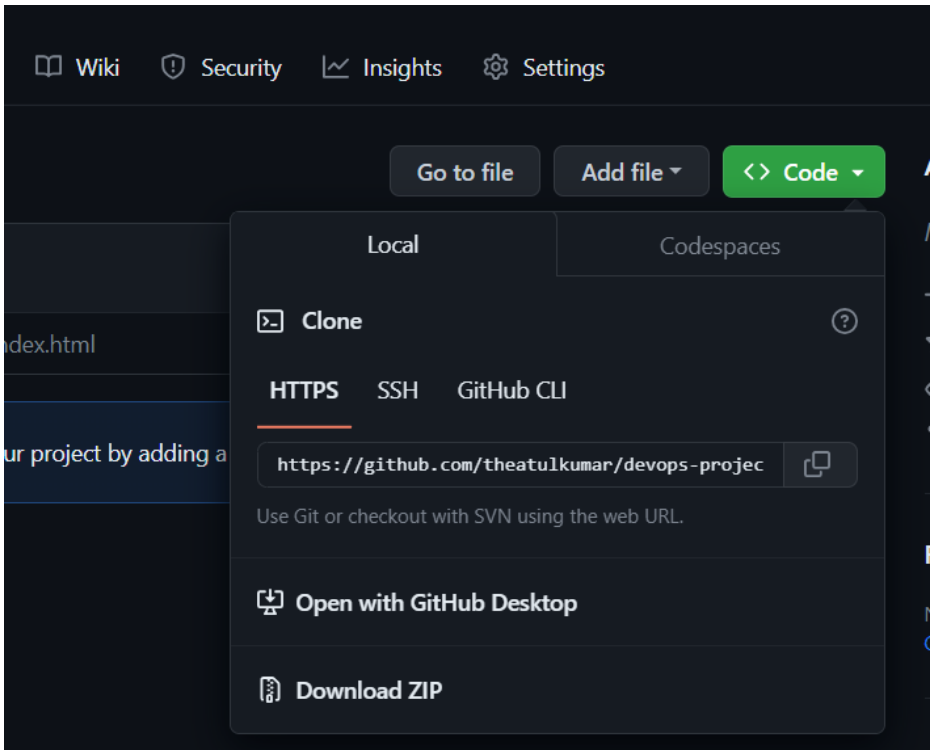
- Now set PasswordAuthentication yes

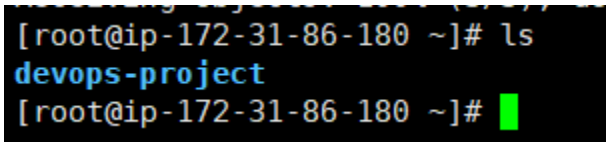
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

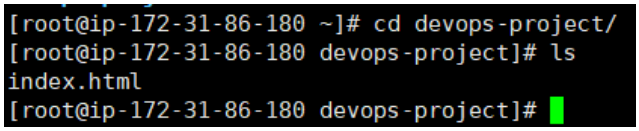
- Now save and exit

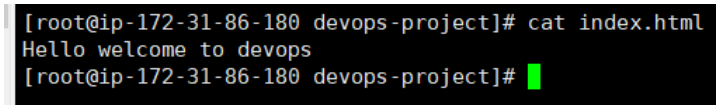
```
# If you just want the PAM account and session checks to run without
:wq
```

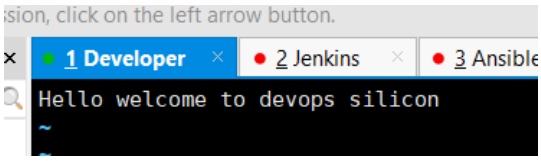
- Now write command “systemctl restart sshd”
- Now we need to create a folder “mkdir /test”
- Then change directory to test “cd /test”
- Now we need to initialize git but before that we need to install git package.
- To install git “yum install git”

- “git init”
- Now we will clone the repository by writing command “git clone https://github.com/theatulkumar/devops-project.git”
- ***update instead of using https link use ssh link
- 

- Once we are done do “ls”
- 

- “cd devops-project“
- “ls”
- 

- “cat index.html”
- 

- “vi index.html”
- Now we can make some changes to the file. Here i added “silicon” in the file
- 

- Now we need to push this to github.
- This is screenshot from class recording.

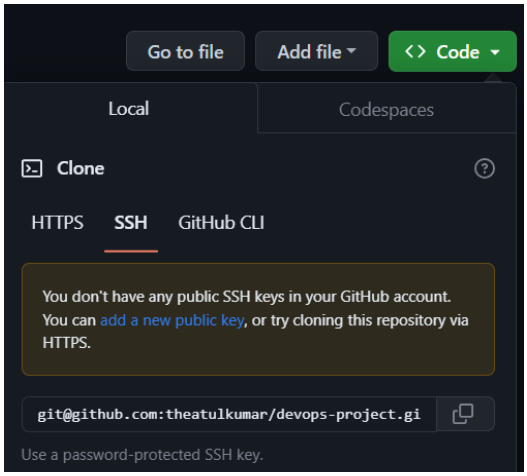
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Developer->Github->Jenkins->Ansible->Webserver
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

------------------
Github Setup
------------------


echo "# devops-sit" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:sumitsah9263/devops-sit.git
git push -u origin main
```

- Now write command to add file to github “git add index.html”
- “git commit -m “test commit” index.html”

```
[root@ip-172-31-86-180 ~]# ls
devops-project
[root@ip-172-31-86-180 ~]# cd devops-project/
[root@ip-172-31-86-180 devops-project]# ls
index.html
[root@ip-172-31-86-180 devops-project]# cat index.html
Hello welcome to devops
[root@ip-172-31-86-180 devops-project]# vi index.html
[root@ip-172-31-86-180 devops-project]# git add index.html
[root@ip-172-31-86-180 devops-project]# git commit -m "test-commit" index.html
[main bb8e21e] test-commit
 Committer: root <root@ip-172-31-86-180.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
[root@ip-172-31-86-180 devops-project]#
```

- “git branch -M main”
- “git remote add origin git@github.com:theatulkumar/devops-project.git”

```
 1 file changed, 1 insertion(+), 1 deletion(-)
[root@ip-172-31-86-180 devops-project]# git branch -M main
[root@ip-172-31-86-180 devops-project]# git remote add origin git@github.com:theatulkumar/devops-project.git
error: remote origin already exists.
[root@ip-172-31-86-180 devops-project]# git push -u origin main
Username for 'https://github.com':
Password for 'https://github.com':
remote: No anonymous write access.
fatal: Authentication failed for 'https://github.com/theatulkumar/devops-project.git/'
[root@ip-172-31-86-180 devops-project]# mkdir /project
[root@ip-172-31-86-180 devops-project]# cd /project
[root@ip-172-31-86-180 project]#
```



- “git push -u origin main”

```
 1 file changed, 1 insertion(+), 1 deletion(-)
[root@ip-172-31-86-180 devops-project]# git branch -M main
[root@ip-172-31-86-180 devops-project]# git remote add origin git@github.com:theatulkumar/devops-project.git
error: remote origin already exists.
[root@ip-172-31-86-180 devops-project]# git push -u origin main
Username for 'https://github.com':
Password for 'https://github.com':
remote: No anonymous write access.
fatal: Authentication failed for 'https://github.com/theatulkumar/devops-project.git/'
[root@ip-172-31-86-180 devops-project]# mkdir /project
[root@ip-172-31-86-180 devops-project]# cd /project
[root@ip-172-31-86-180 project]#
```

- We can see that authentication has failed. So now let’s try again
- Creating new folder named ‘project’ - “mkdir /project”
- changing directory to project “cd /project”
- “git init”
- “git clone https://github.com/theatulkumar/devops-project.git”

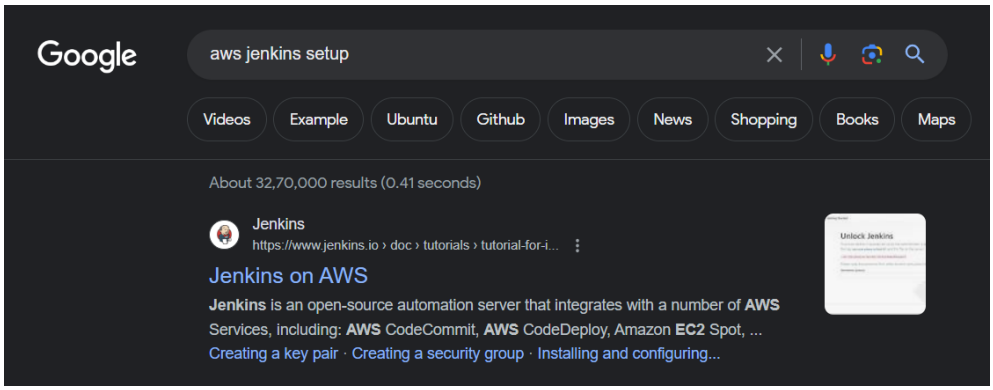
```
[root@ip-172-31-86-180 project]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /project/.git/
[root@ip-172-31-86-180 project]# git clone https://github.com/theatulkumar/devops-project.git
Cloning into 'devops-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
[root@ip-172-31-86-180 project]# cd
[root@ip-172-31-86-180 ~]# ls
devops-project
[root@ip-172-31-86-180 ~]# cd devops-project/
[root@ip-172-31-86-180 devops-project]#
```

-

- “cd”
- “ls”
- “cd devops-project”
- “ls”
- “cat index.html”
- “git add index.html”
- “git commit -m “test commit” index.html”
- “git branch -M main”
- “git remote add origin”
- “git push -u origin main”

Now we need to setup jenkins

- Open google and search “aws jenkins setup”. Click on the link



- After opening the link follow the given commands

- This documentation keeps updating so we should check for the latest documentation from the website.
  - sudo yum update –y
  - sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
  - sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
  - sudo yum upgrade
  - sudo amazon-linux-extras install java-openjdk11 -y
  - sudo yum install jenkins -y
  - sudo systemctl enable jenkins
  - sudo systemctl start jenkins
  - sudo systemctl start jenkins
  - sudo systemctl status jenkins

Downloading and installing Jenkins

Completing the previous steps enables you to download and install Jenkins on AWS. To download and install Jenkins:

1. Ensure that your software packages are up to date on your instance by using the following command to perform a quick software update:

```
[ec2-user ~]$ sudo yum update –y
```

2. Add the Jenkins repo using the following command:

```
[ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
[ec2-user ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
[ec2-user ~]$ sudo yum upgrade
```

4. Install Java (Amazon Linux 2):

```
[ec2-user ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

5. Install Java (Amazon Linux 2023):

```
[ec2-user ~]$ sudo dnf install java-11-amazon-corretto -y
```

6. Install Jenkins:

```
[ec2-user ~]$ sudo yum install jenkins -y
```

7. Enable the Jenkins service to start at boot:

```
[ec2-user ~]$ sudo systemctl enable jenkins
```

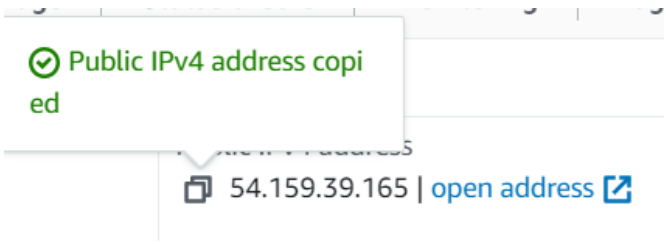
8. Start Jenkins as a service:

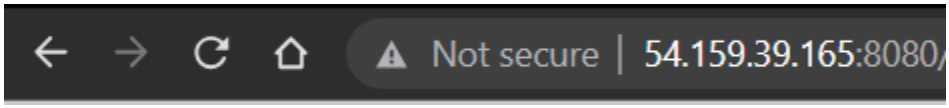
```
[ec2-user ~]$ sudo systemctl start jenkins
```

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

```
[ec2-user ~]$ sudo systemctl status jenkins
```

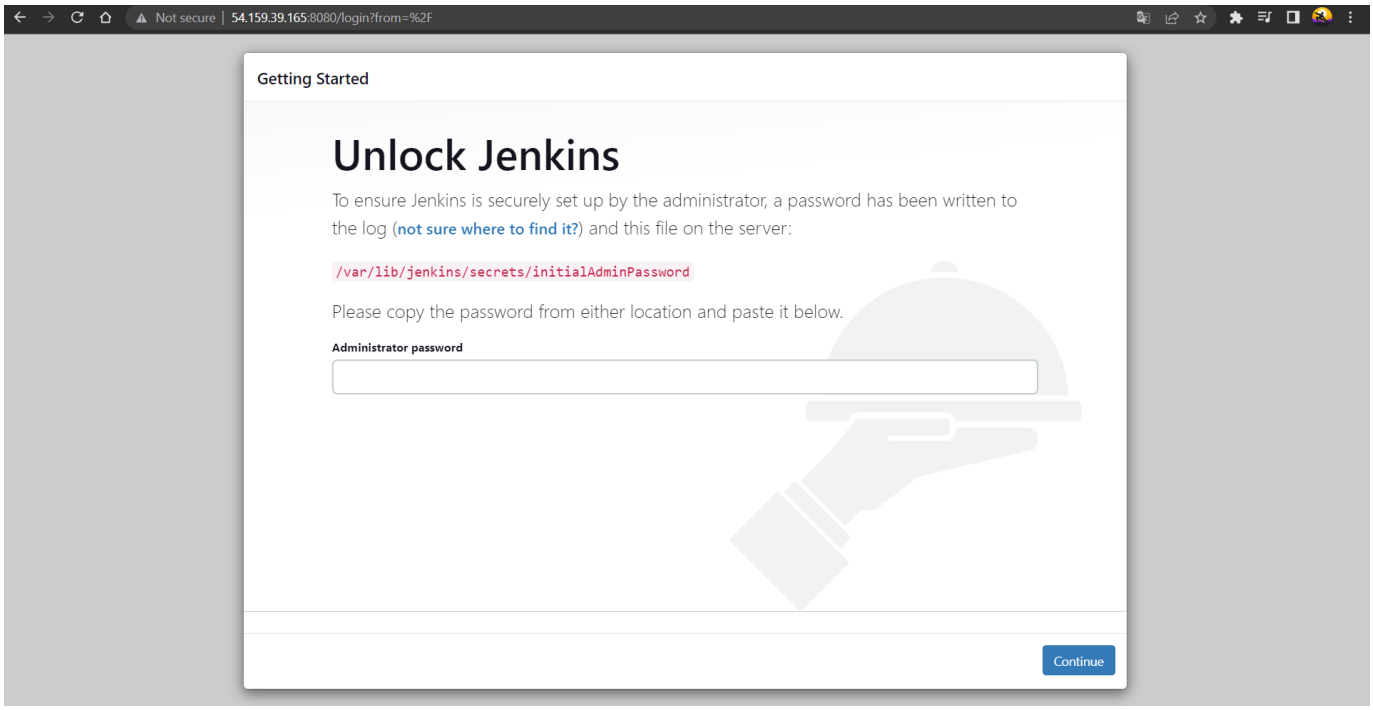
- After performing these commands copy the public ip of the jenkins server and paste in browser and jenkins runs on port 8080 so add 8080 after public ip address



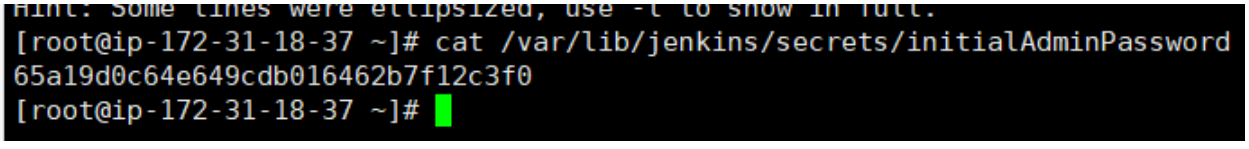


- added :8080 (port number)

Now we can see the page like this, where the initial password is located. So “copy the location.”

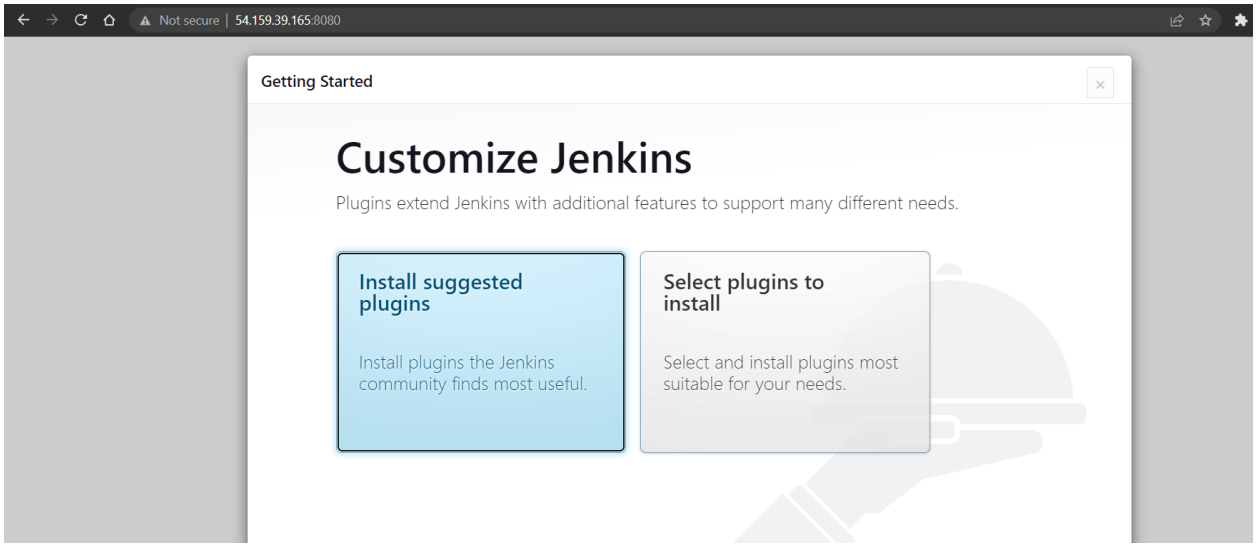


Use cat command to check the password

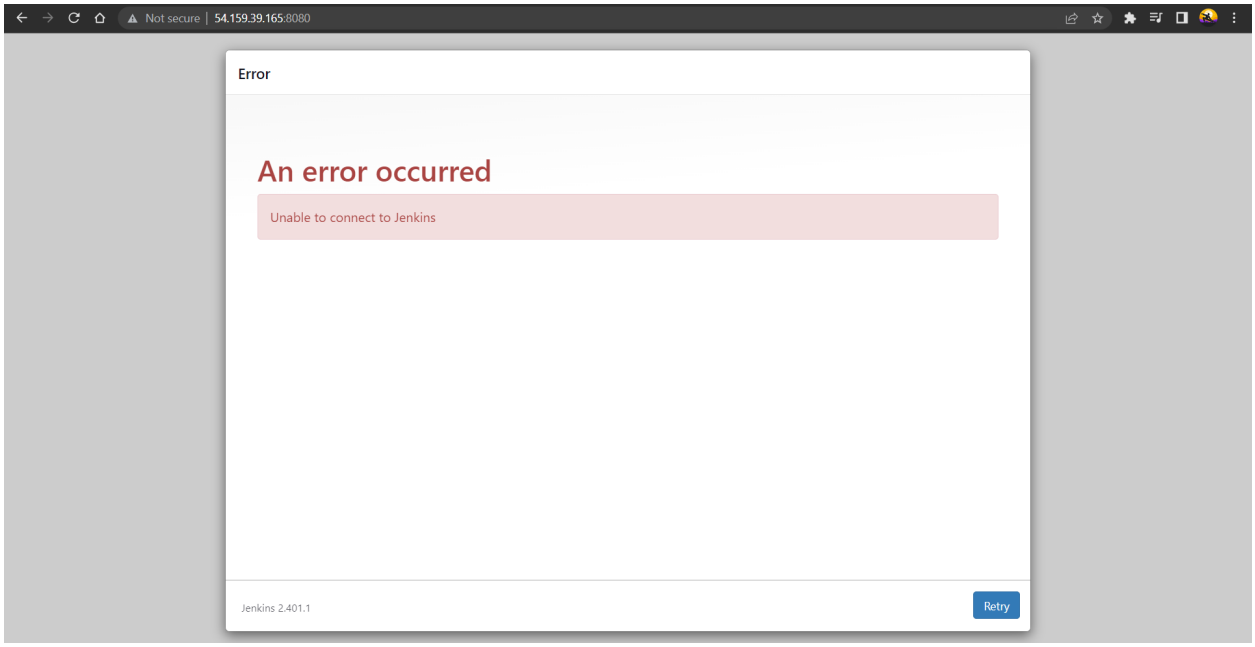


Now paste the password and click on continue

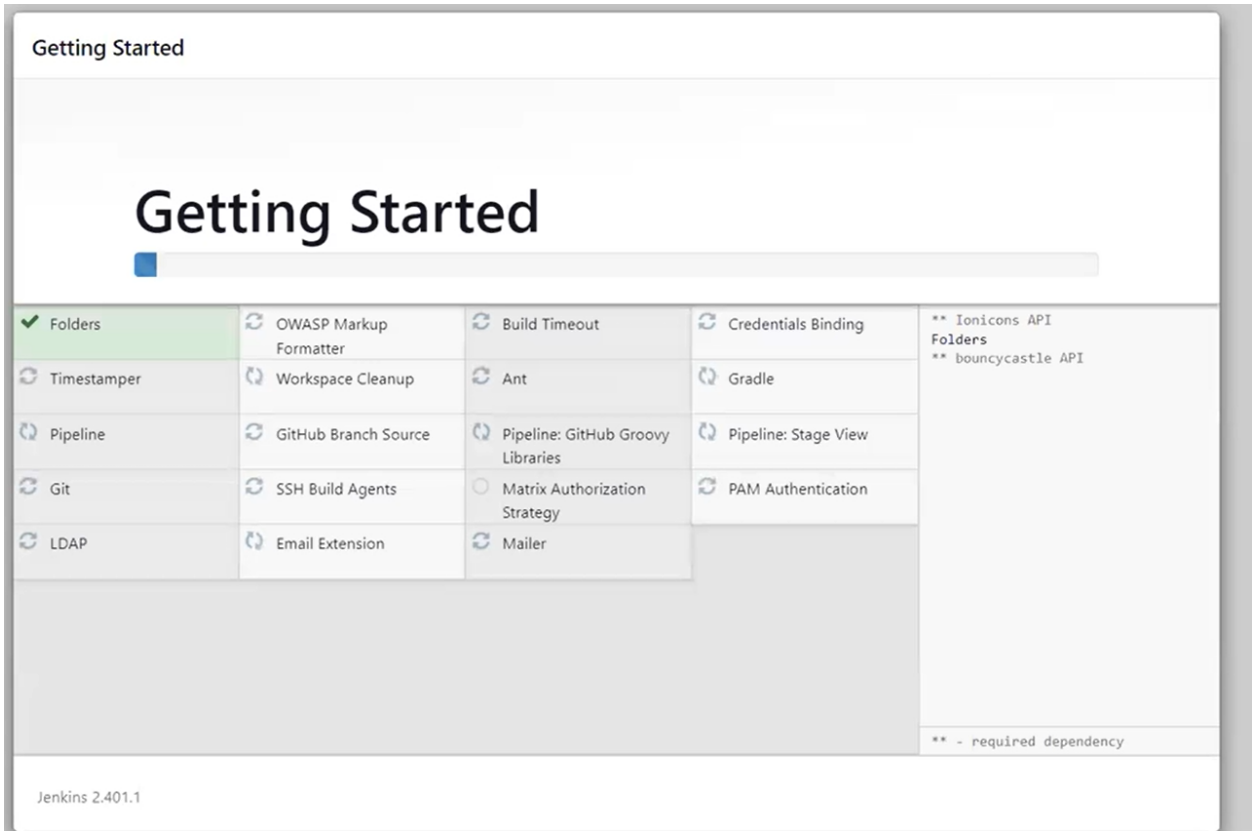
- Click on install suggested plugins.



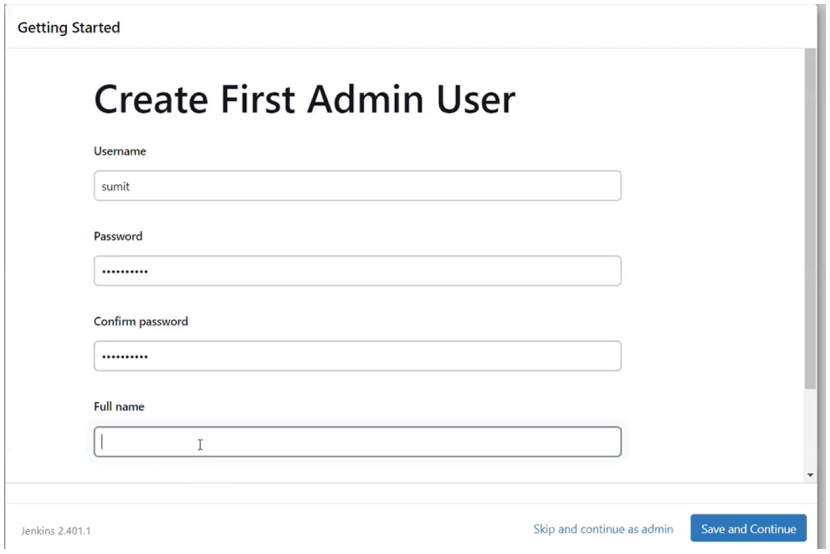
In my case some error has occurred so I will try to fix the error.



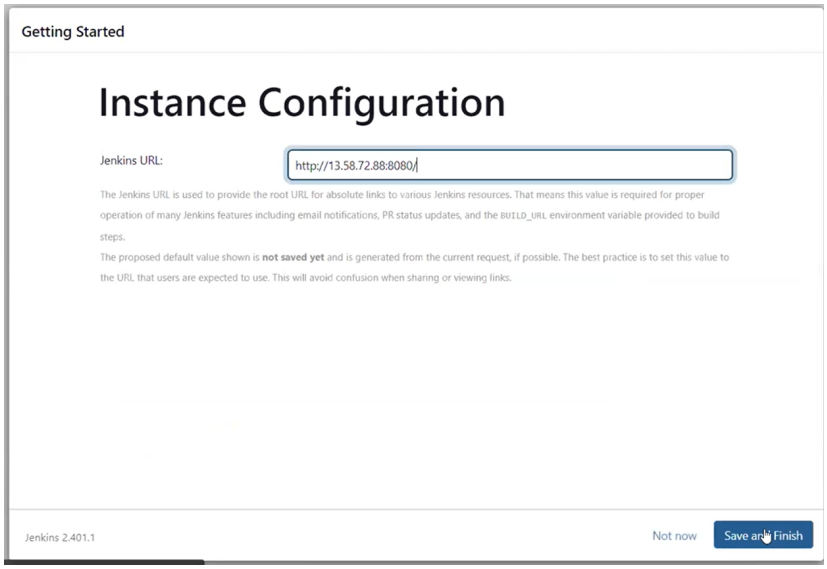
- After clicking on Install suggested plugins this will appear.



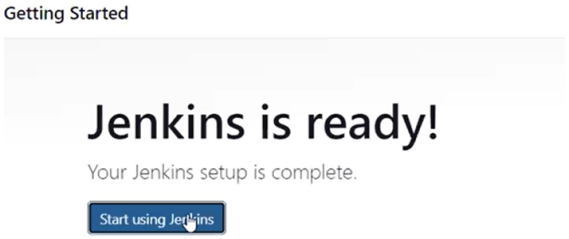
Now fill in the details then save and continue.



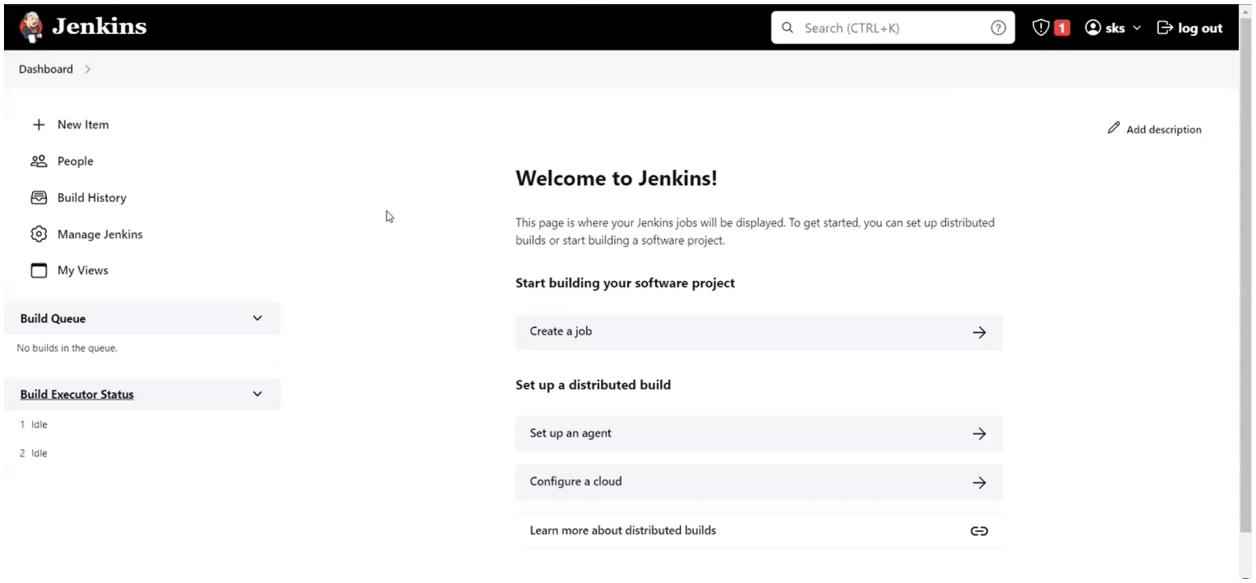
- Now click on save and finish.



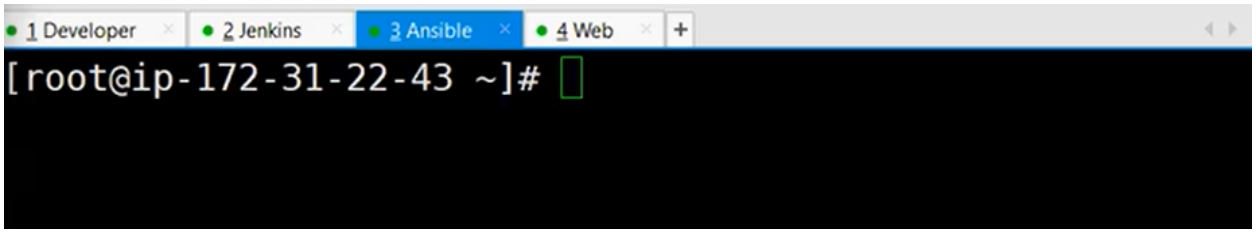
- Now click on start using Jenkins



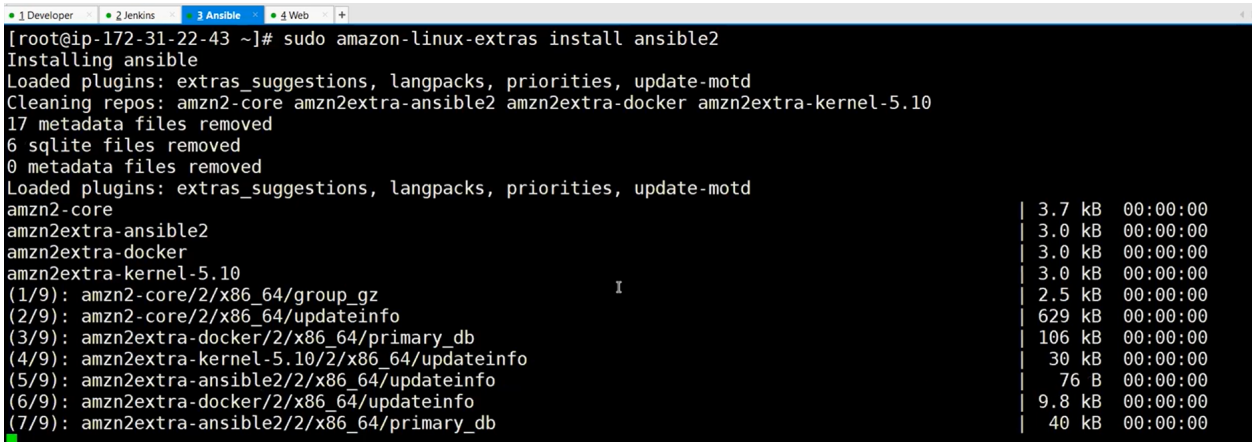
This is the page of jenkins



- **Now we need to setup ansible**



- **For setting up ansible we have command : "sudo amazon-linux-extras install ansible2"**



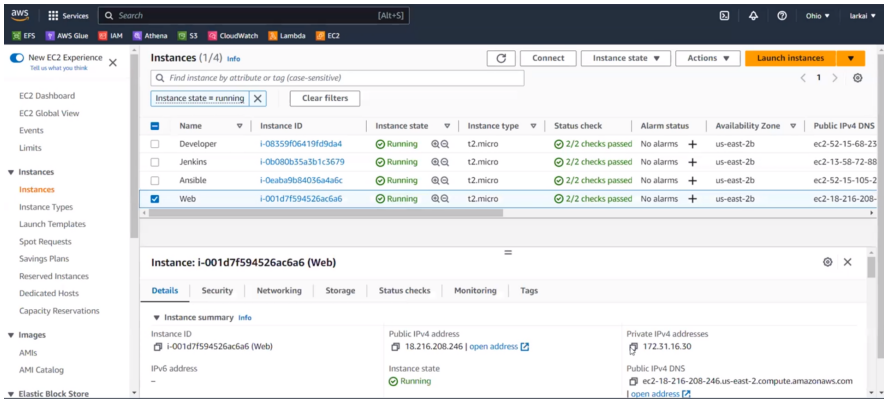
After installation of ansible we need to add ansible with the web server.

- **For adding ansible with the web server we need to change the configuration file. "vi /etc/ansible/hosts"**

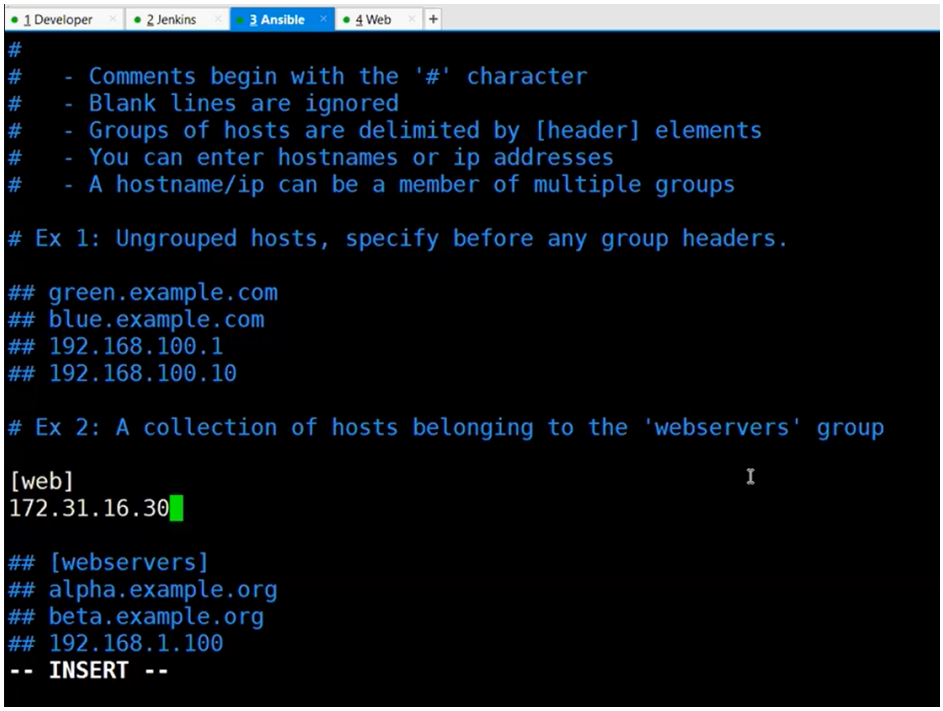
```
[root@ip-172-31-22-43 ~]# vi /etc/ansible/hosts
```

Here we can add any number of web servers by providing their private IP address. If we want to add 50 number of web servers we can add them here.

- **Copy the private IP id**



- **Paste the IP address here like this :**
  [web]
  172.31.16.30



Then save and exit

Now ansible is configured with web server 👍

- **Now installing apache in the web server: "yum install httpd"**

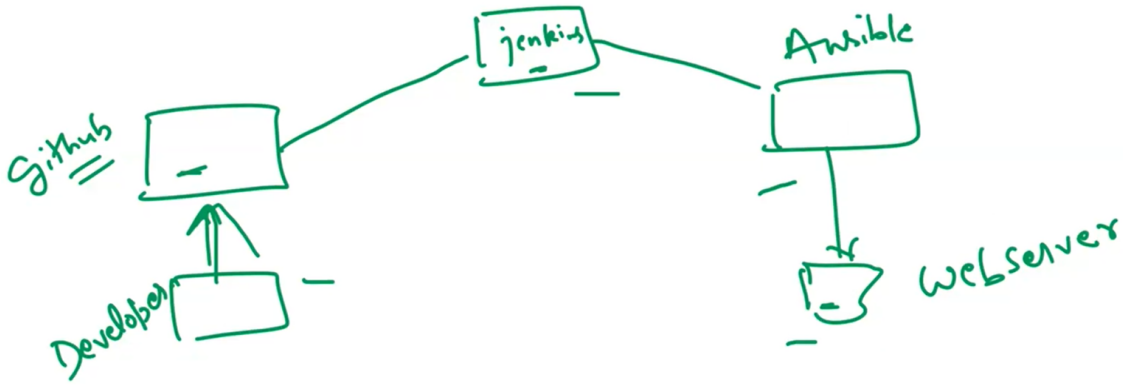
```
[root@ip-172-31-16-30 ~]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                  | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.57-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.57-1.amzn2 for package: httpd-2.4.57-1.amzn2.x86_64
```

- **"systemctl start httpd"**
- **"systemctl enable httpd"**

```
[root@ip-172-31-16-30 ~]# systemctl start httpd
```

```
[root@ip-172-31-16-30 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.servic
e.
[root@ip-172-31-16-30 ~]#
```

As we set up password less authentication for developer server on github.
We will set up password less authentication for ansible server to web server.



- **Now change the password of the root "passwd root"**

```
[root@ip-172-31-16-30 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
```

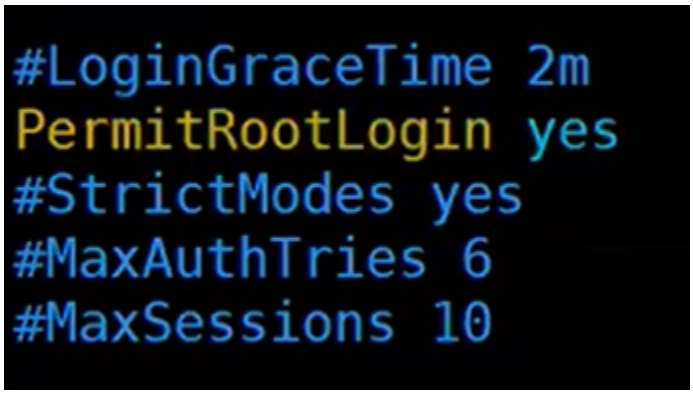
web

We tend to forget the password or confuse it with the other password so we should write it somewhere.

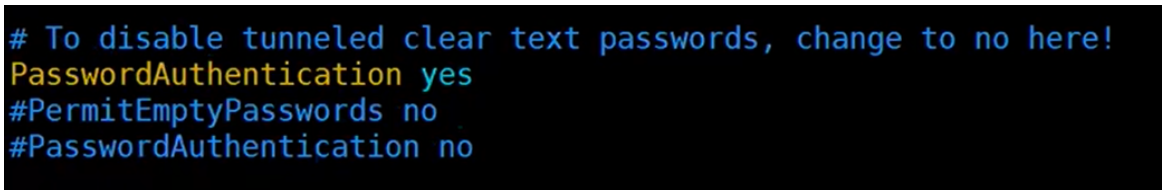
Now we need to go the the sshd configuration file : "vi /etc/ssh/sshd_config"

```
[root@ip-172-31-16-30 ~]# vi /etc/ssh/sshd_config
```

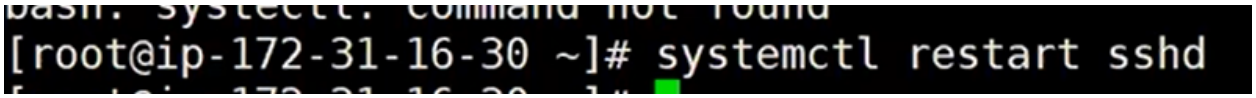
- **PermitRootLogin yes**



- **PasswordAuthentication yes**

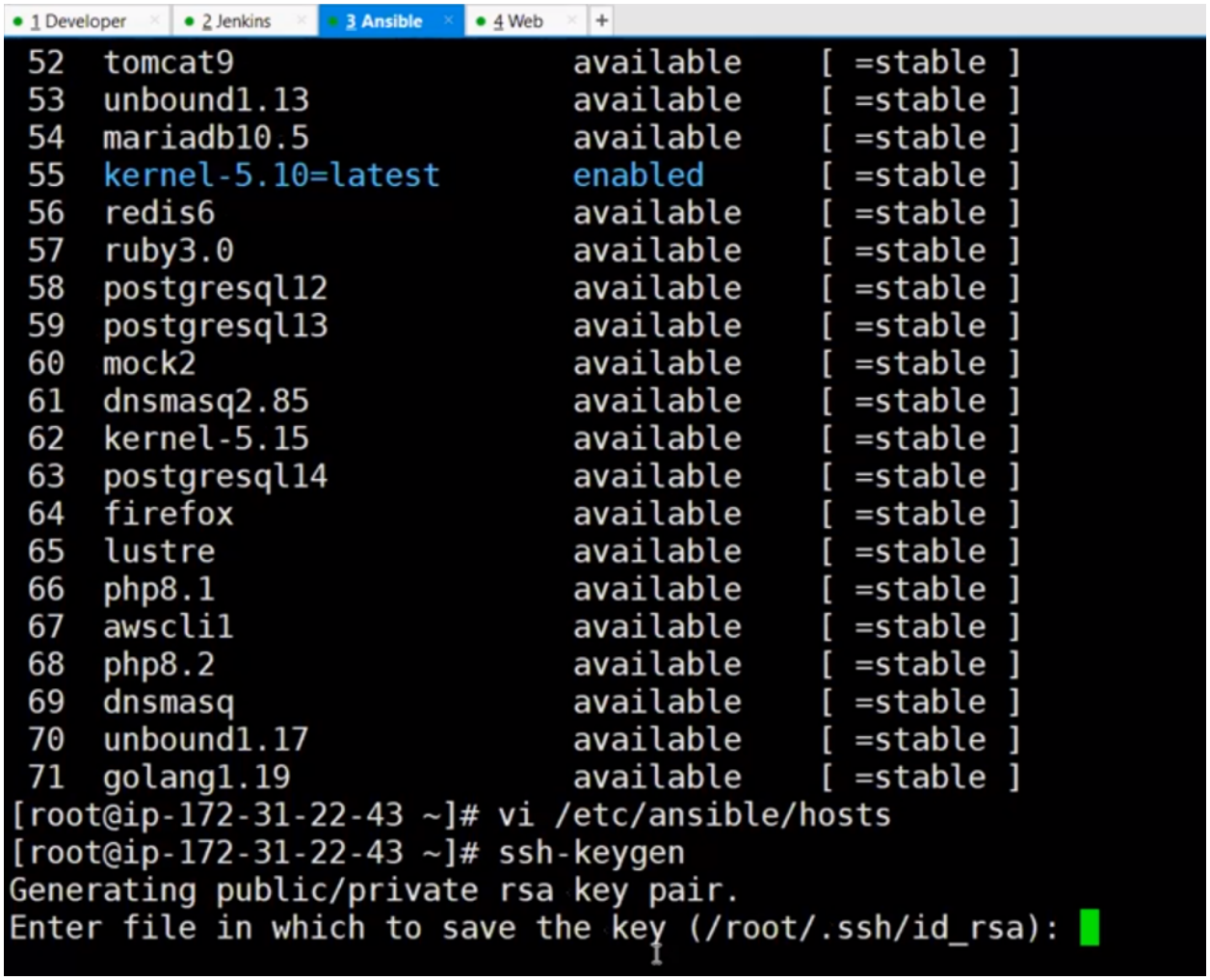


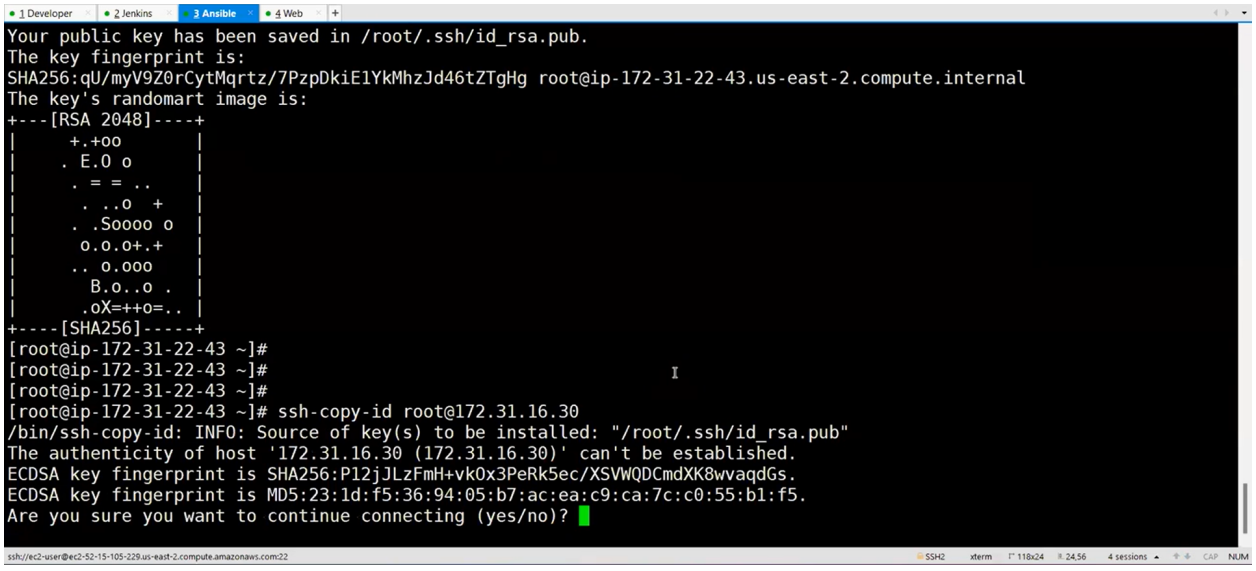
- **After that we will do : "systemctl restart sshd"**

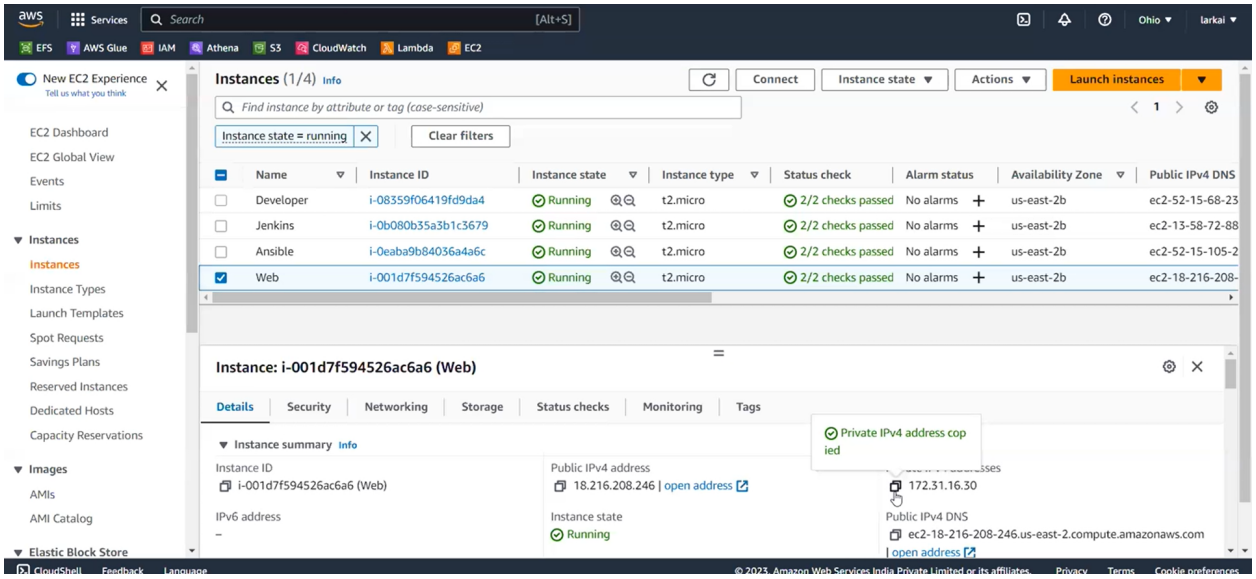


Now in Ansible server

- **“ssh-keygen”**







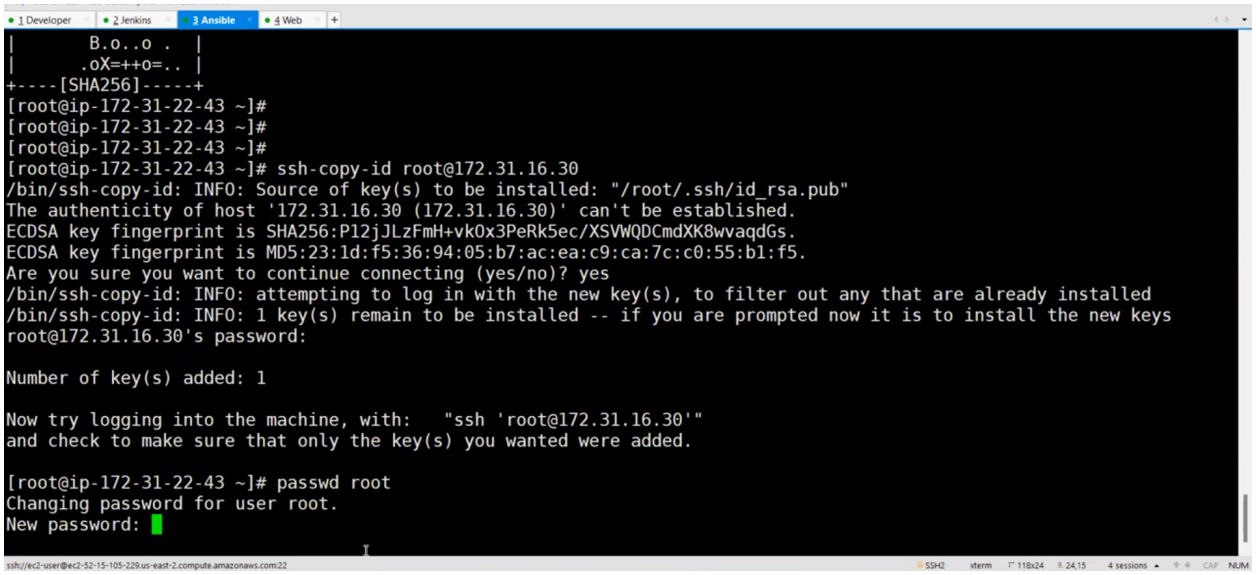
We put the private ip address of the web server to the ansible

- **ssh-copy-id root@172.31.16.30**

```
[root@ip-172-31-22-43 ~]# ssh-copy-id root@172.31.16.30
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.16.30 (172.31.16.30)' can't be established.
ECDSA key fingerprint is SHA256:P12jJLzFmH+vkOx3PeRk5ec/XSVWQDCmdXK8wvaqdGs.
ECDSA key fingerprint is MD5:23:1d:f5:36:94:05:b7:ac:ea:c9:ca:7c:c0:55:b1:f5.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.16.30's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.31.16.30'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-22-43 ~]#
```

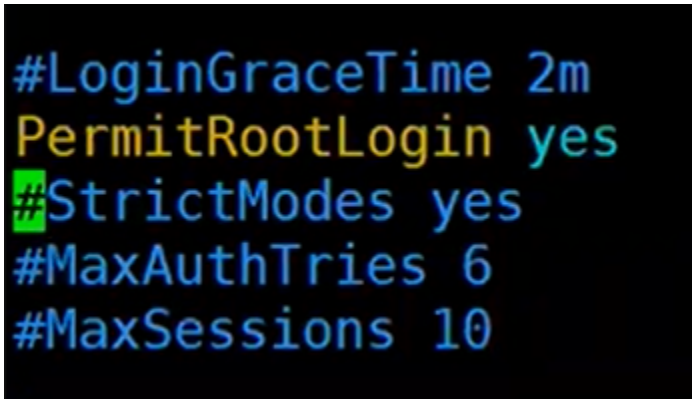
Now our ansible server will be able to communicate with the web server.

Now in the same way we need to setup password less authentication for jenkins to ansible.
Following the same steps
passwd root

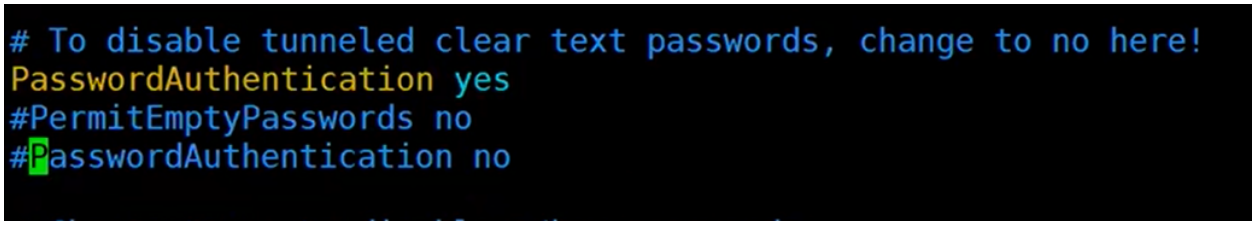


Now changing the configuration file “vi /etc/ssh/sshd_config”

PermitRootLogin yes



PasswordAuthentication yes



- **“systemclt restart sshd"**

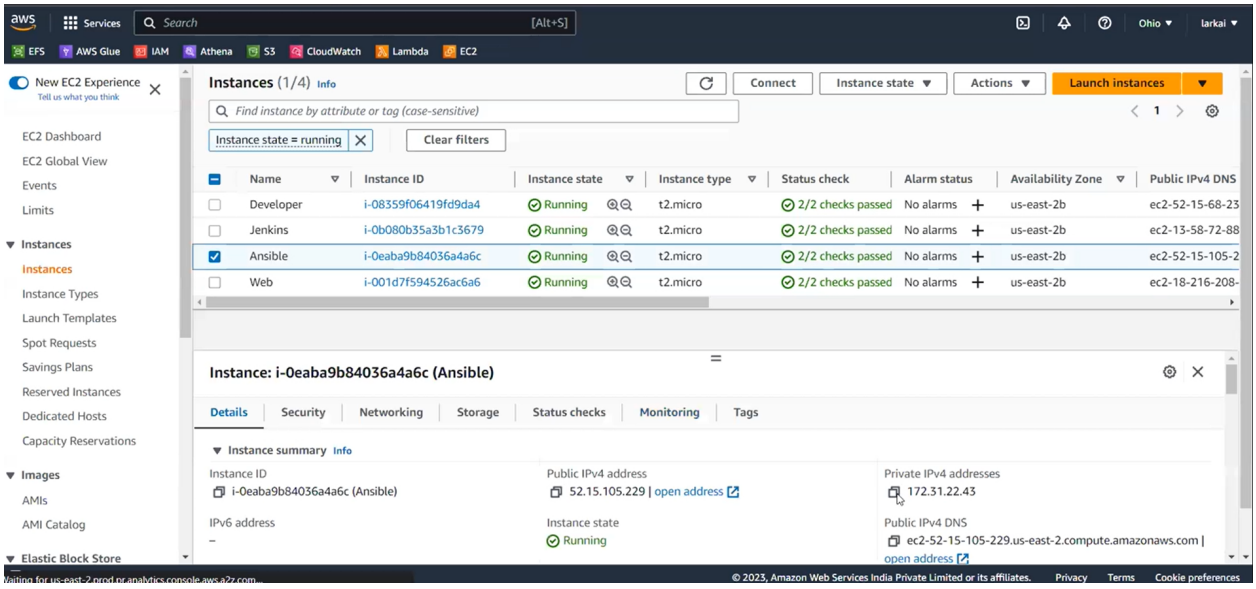
```
[root@ip-172-31-22-43 ~]# systemctl restart sshd
```

Now come to the jenkins server

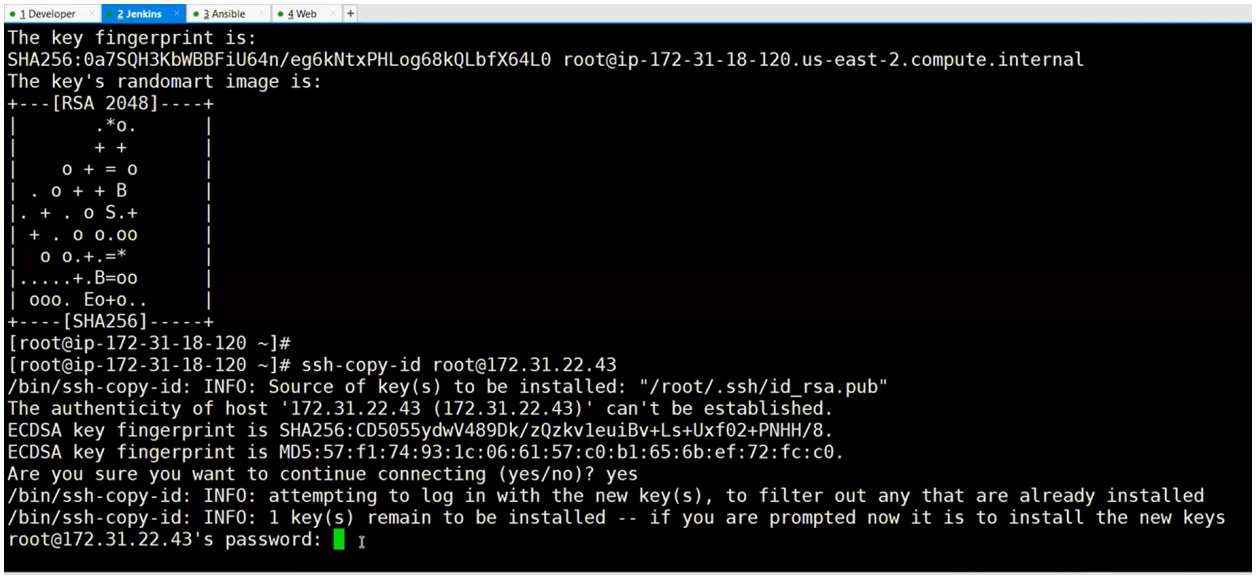
- **Generate key pair**
  “ssh-keygen”

```
1 Developer  2 Jenkins  3 Ansible  4 Web
4178c1ad0ce4428aa1a9a20eaee4260e
[root@ip-172-31-18-120 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:0a7SQH3KbWBBFiU64n/eg6kNtxPHLog68kQLbfX64L0 root@ip-172-31-18-120.us-east-2.compute.internal
The key's randomart image is:
+---[RSA 2048]----+
|       .*o.      |
|       + +       |
|    o + = o      |
| . o + + B       |
|. + . o S.+      |
| + . o o.oo      |
|  o o.+.=*       |
|.....+.B=oo      |
| ooo. Eo+o..     |
+----[SHA256]-----+
[root@ip-172-31-18-120 ~]#
[root@ip-172-31-18-120 ~]# ssh-copy-id root@172.31.22.43
```

- **“ssh-copy-id root@172.31.22.43“ where 172.31.22.43 is the private ip address of the Ansible server.**

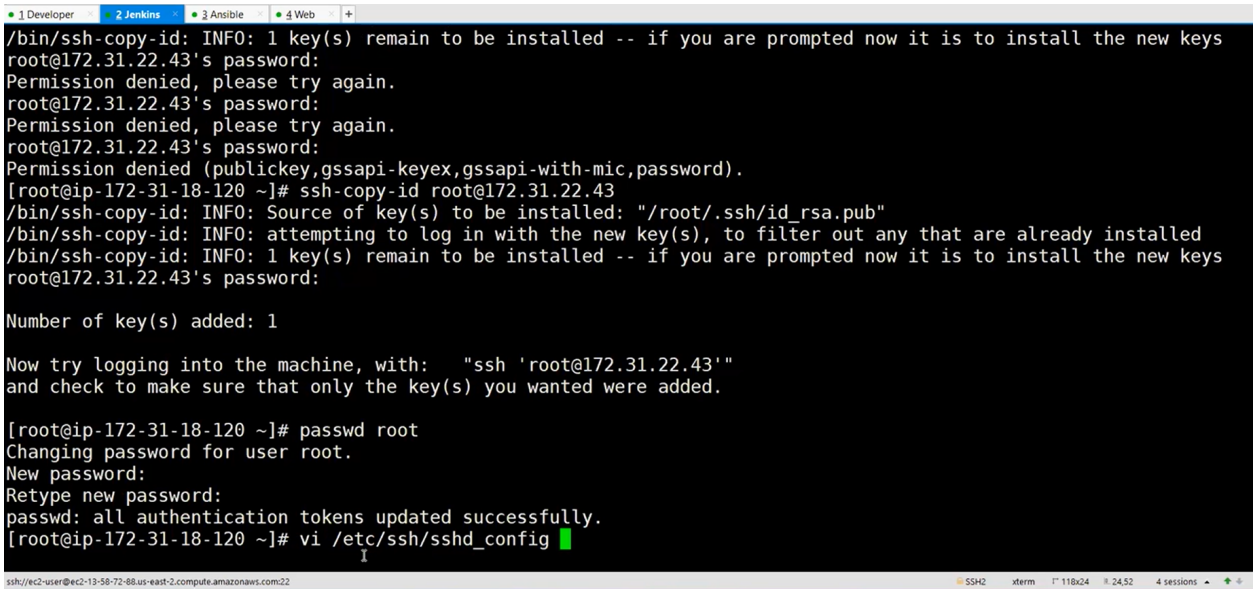


Now give the password that we set in the Ansible server (see the ip address that whose password is being asked)

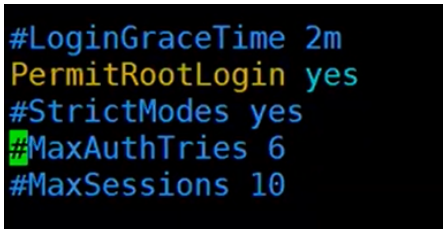


Now we have established a connection between jenkins to ansible.

Now changing the password of jenkins server "passwd root"
Then configuring the file by "vi /etc/ssh/sshd_config"



PermitRootLogin yes



PasswordAuthentication yes

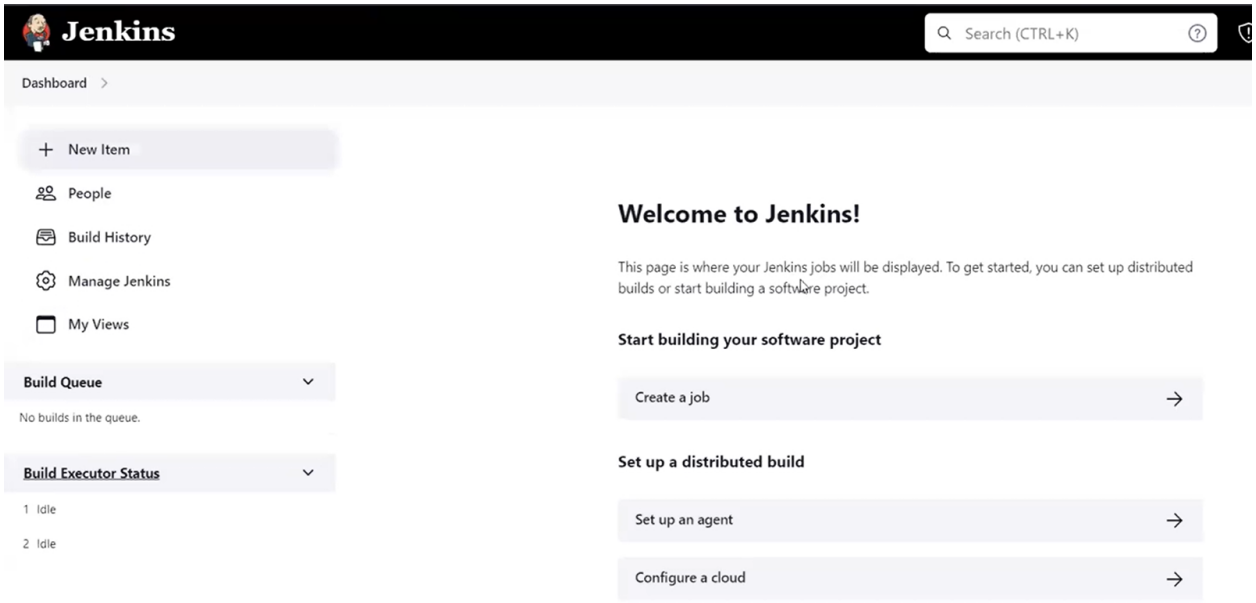
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

systemctl restart sshd

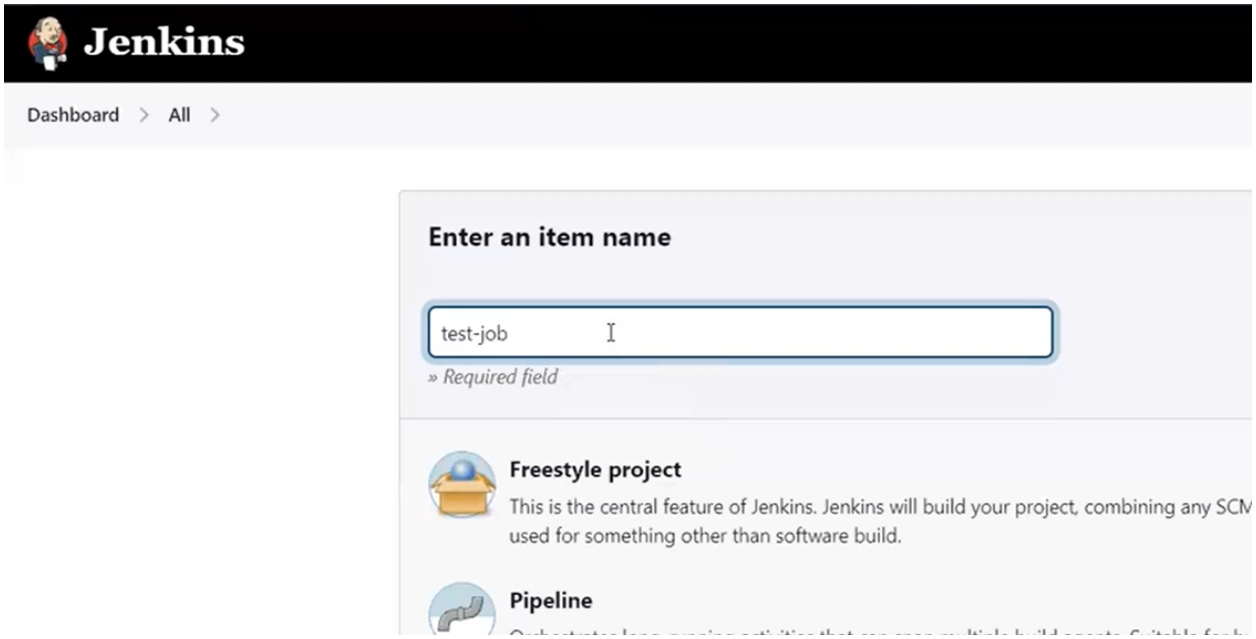
```
[root@ip-172-31-18-120 ~]# systemctl restart sshd
```

Now we need to make a pipeline in a jenkins

- **Click on the new item.**

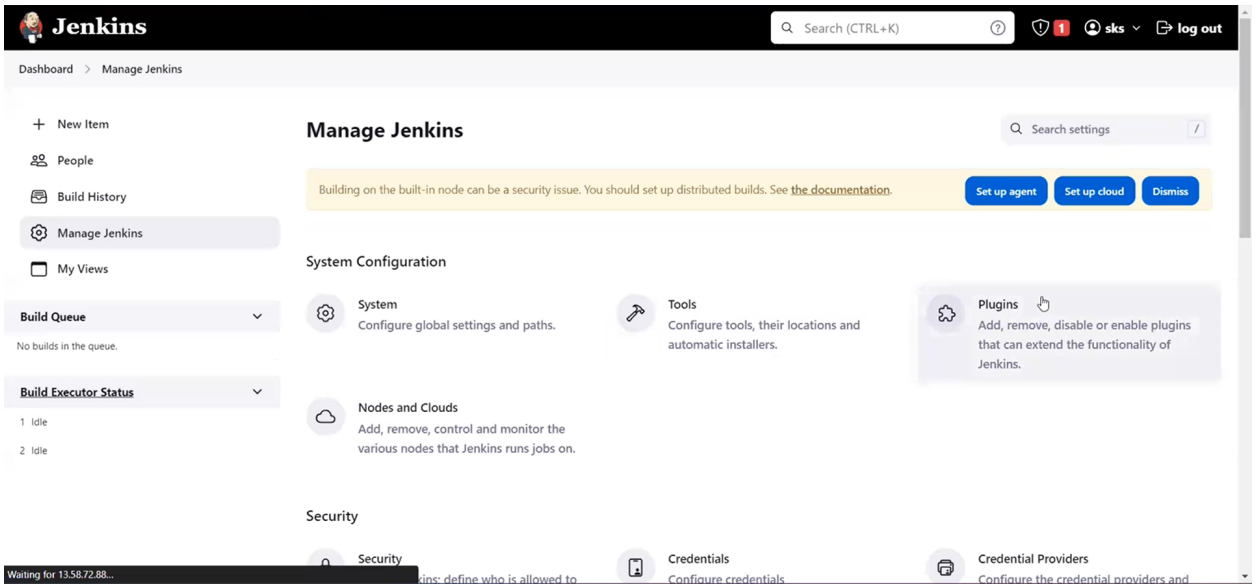


Enter an item name
Then select freestyle project and click ok

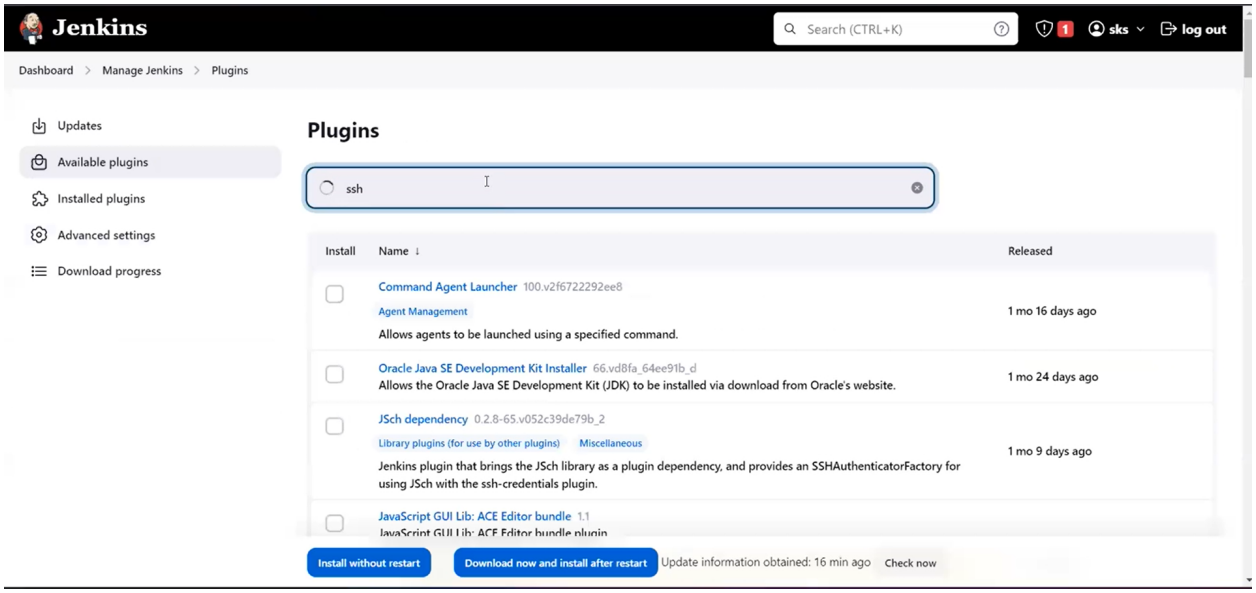


Now Jenkins needs to control ansible, web server like stuff.
So we need to connect jenkins with ansible and web server.

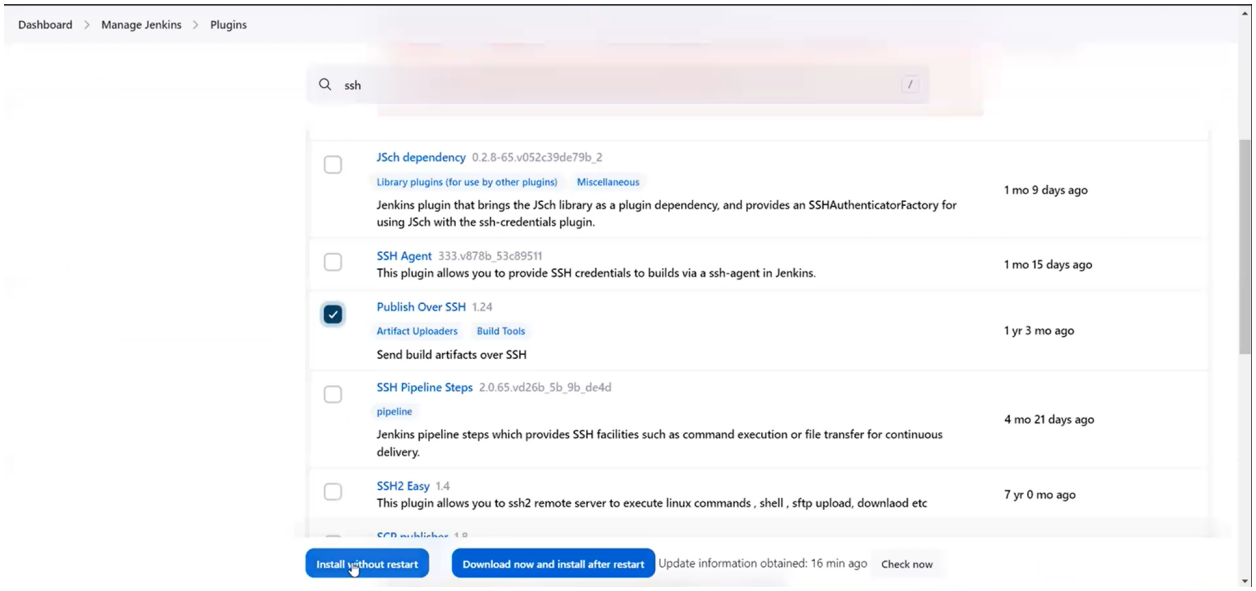
So for that
Go to manage jenkins and click on Plugins



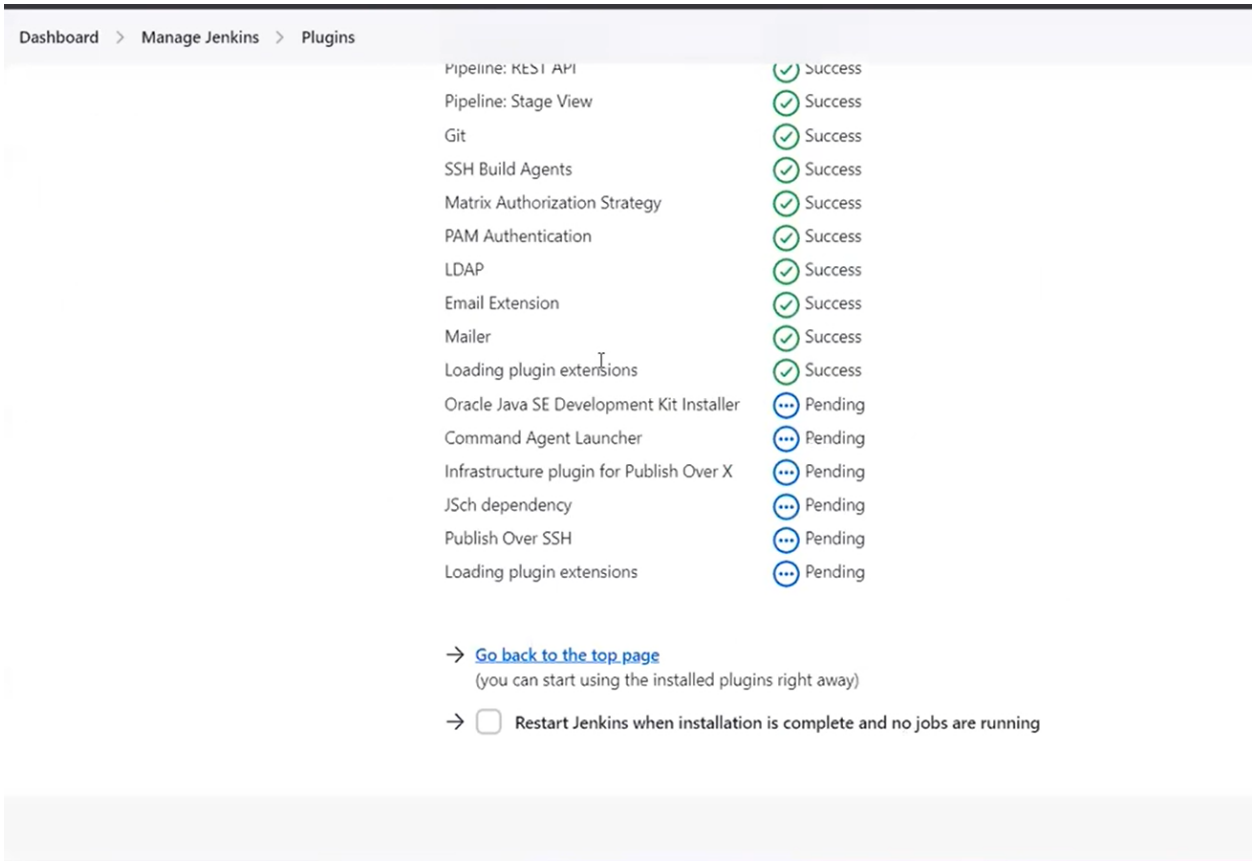
Come to available plugins and then search for ssh



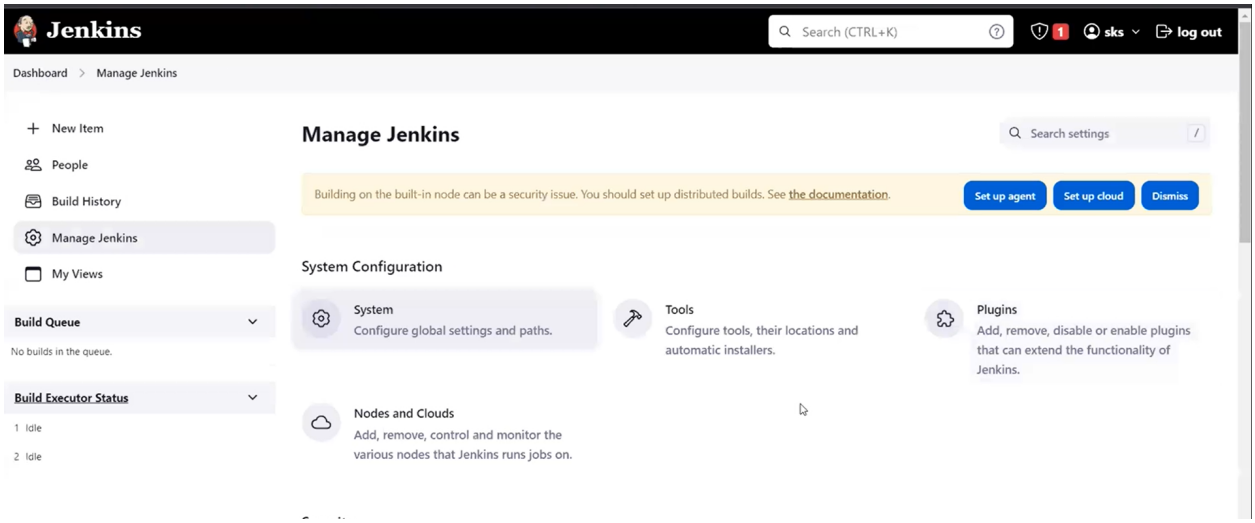
- **Select Publish over SSH and then click on install without restart**



Then there will be option to restart .....select restart

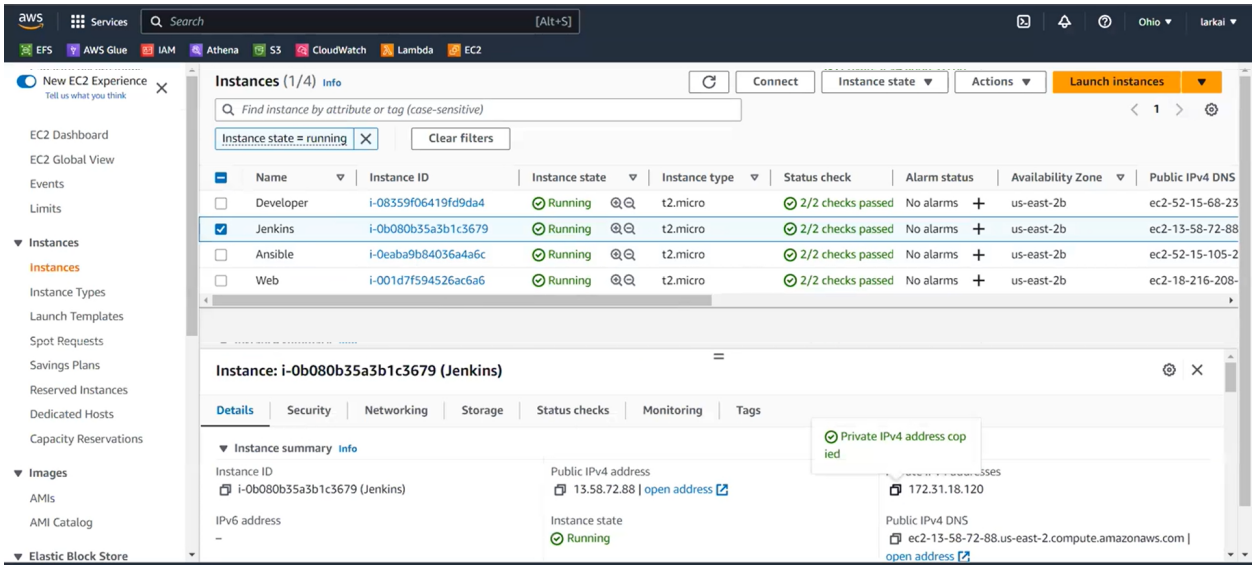


Now go to manage jenkins and then system
Now we will be able to add servers to the jenkins




Dashboard > Manage Jenkins > System >
Path to key ?
Key ?
Disable exec ?
SSH Servers
SSH Server
Name ?
Jenkins
Required. Cannot contain < & ' " \
Hostname ?
Save
Apply

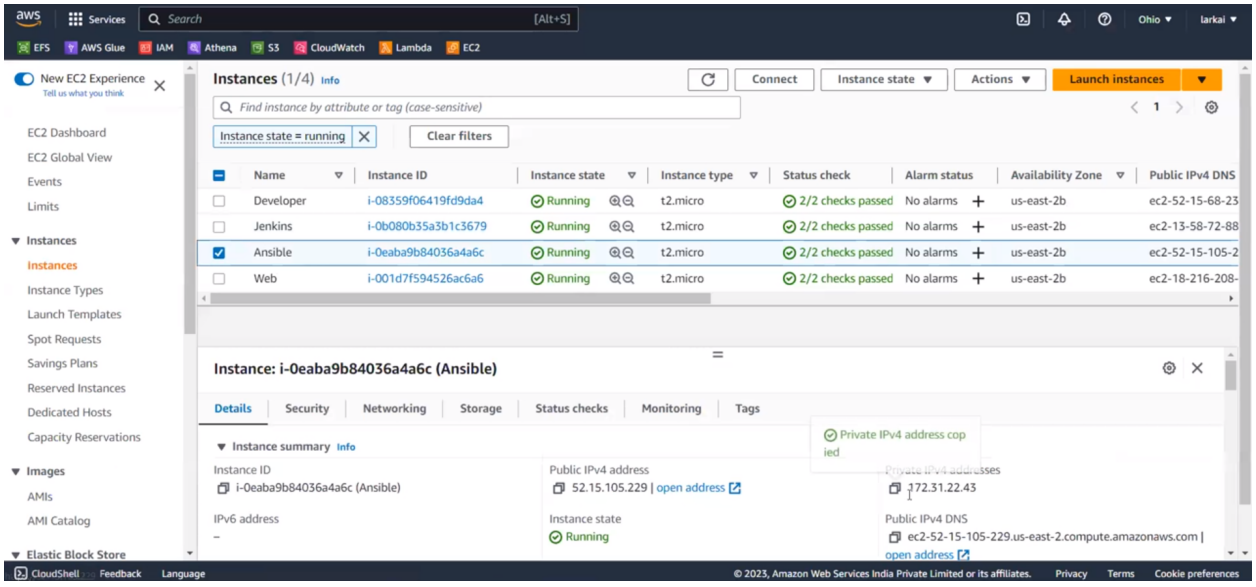

Dashboard > Manage Jenkins > System >
SSH Server
Name
Jenkins
Hostname
172.31.18.120
Username
root
Required
Remote Directory
Advanced
Test Configuration
Save
Apply

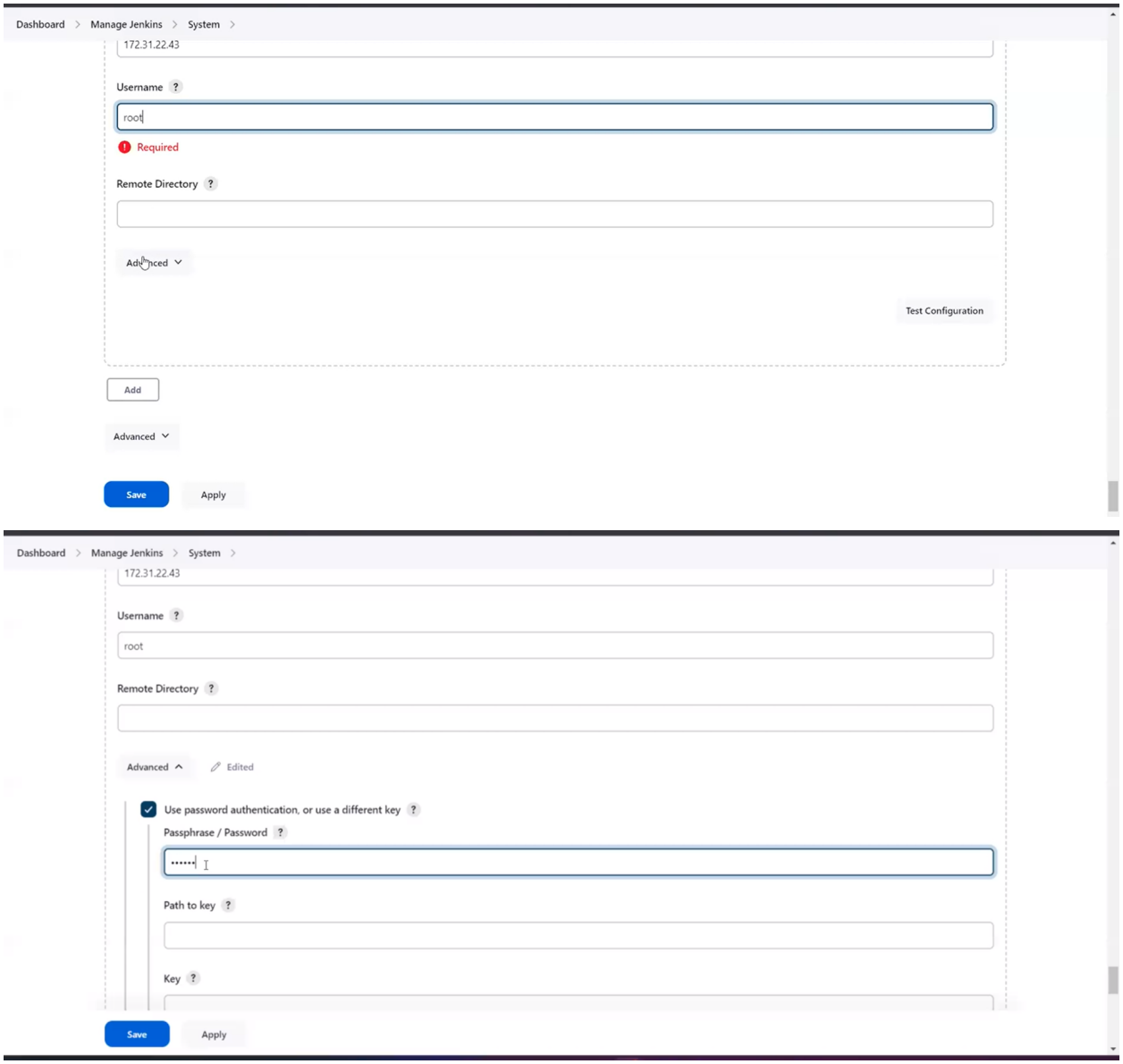

aws
Services
Search
[Alt+S]
Ohio
larkai
EFS
AWS Glue
IAM
Athena
S3
CloudWatch
Lambda
EC2
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Instances (1/4) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Developer
i-08359f06419fd9da4
Running
t2.micro
2/2 checks passed
No alarms
us-east-2b
ec2-52-15-68-23
Jenkins
i-0b080b35a3b1c3679
Running
t2.micro
2/2 checks passed
No alarms
us-east-2b
ec2-13-58-72-88
Ansible
i-0eaba9b84036a4a6c
Running
t2.micro
2/2 checks passed
No alarms
us-east-2b
ec2-52-15-105-2
Web
i-001d7f594526ac6a6
Running
t2.micro
2/2 checks passed
No alarms
us-east-2b
ec2-18-216-208-
Instance: i-0b080b35a3b1c3679 (Jenkins)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Private IPv4 address copied
Instance summary Info
Instance ID
i-0b080b35a3b1c3679 (Jenkins)
Public IPv4 address
13.58.72.88 | open address
172.31.18.120
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-13-58-72-88.us-east-2.compute.amazonaws.com | open address

Click on advance and put password of jenkins server

Now once we did that click on the test configuration
We can see the success message.

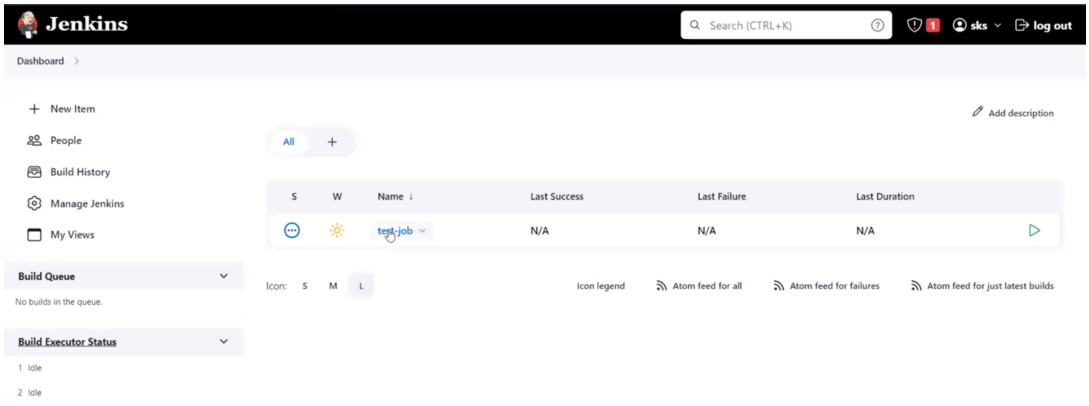
Similarly adding one ansible server



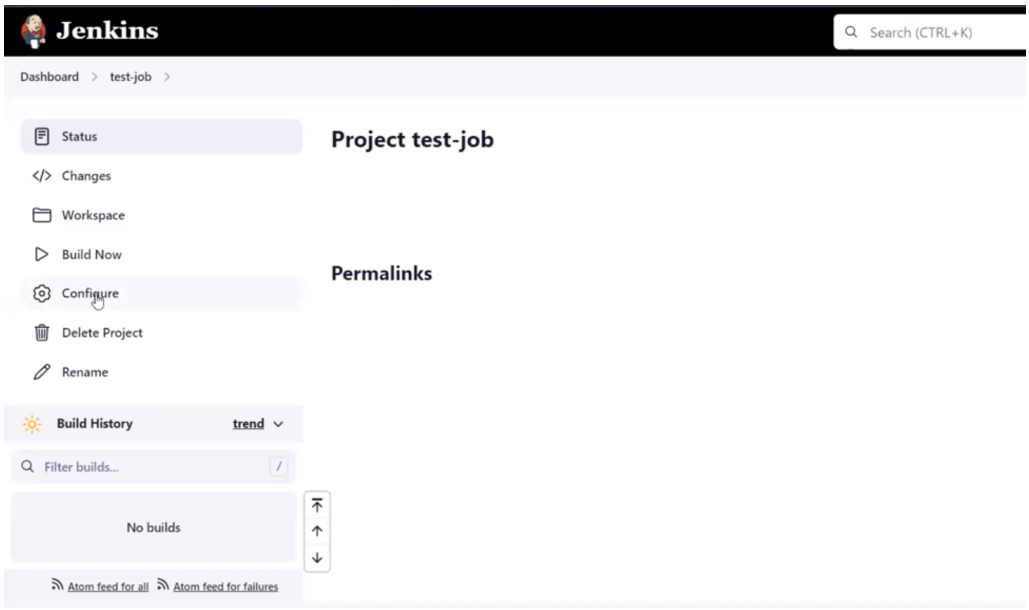


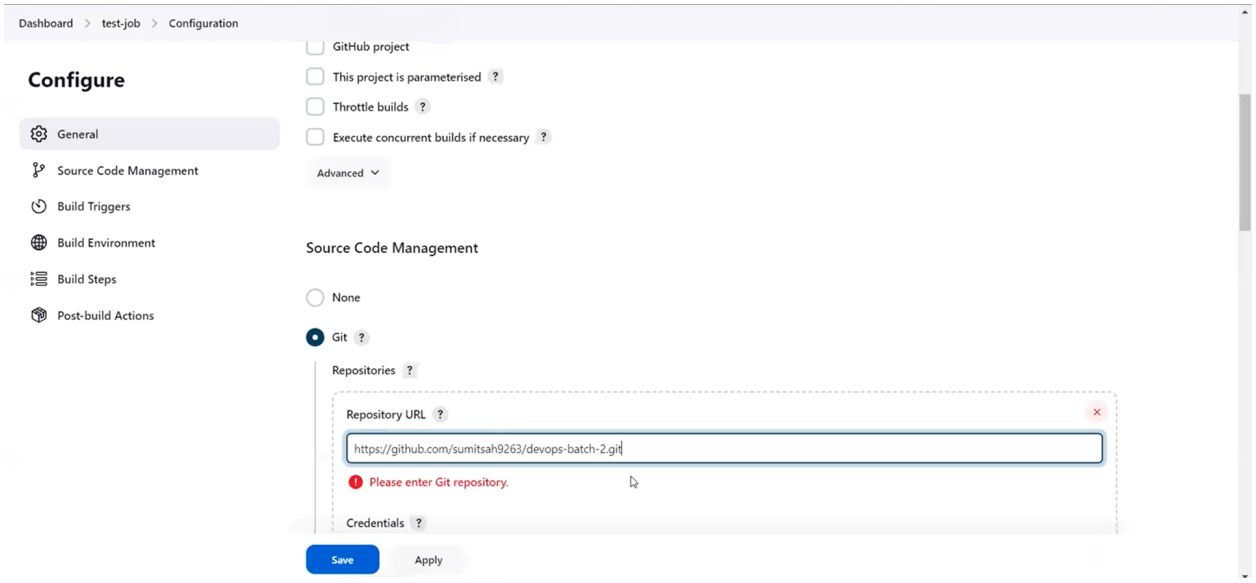
After test configuration apply and save

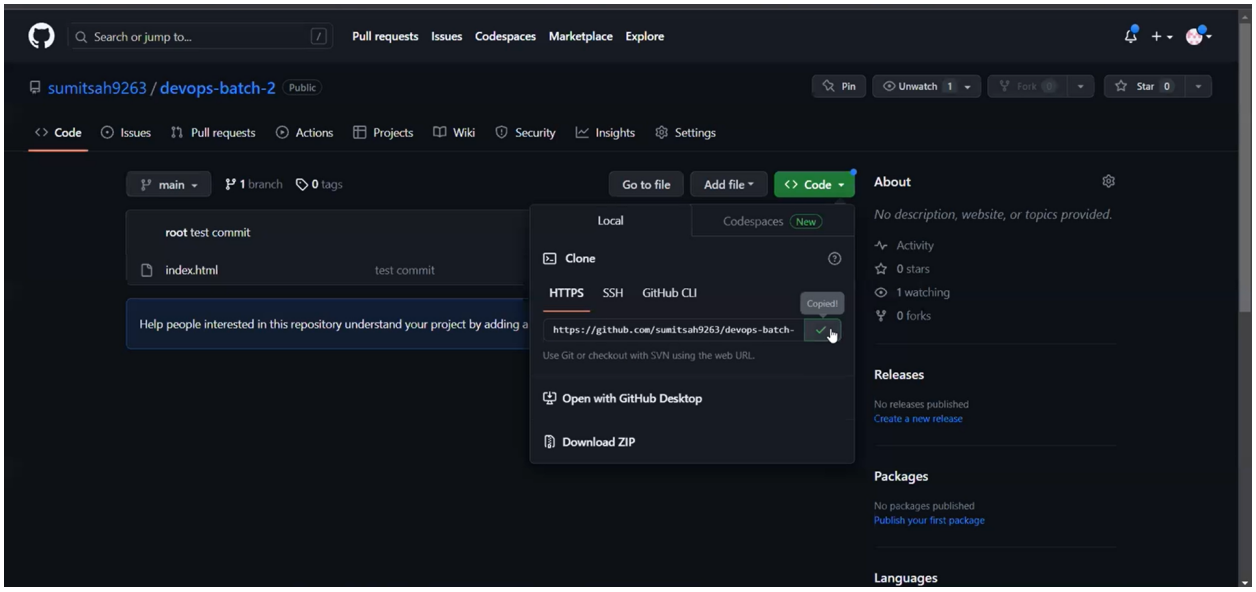
Now come to test-job



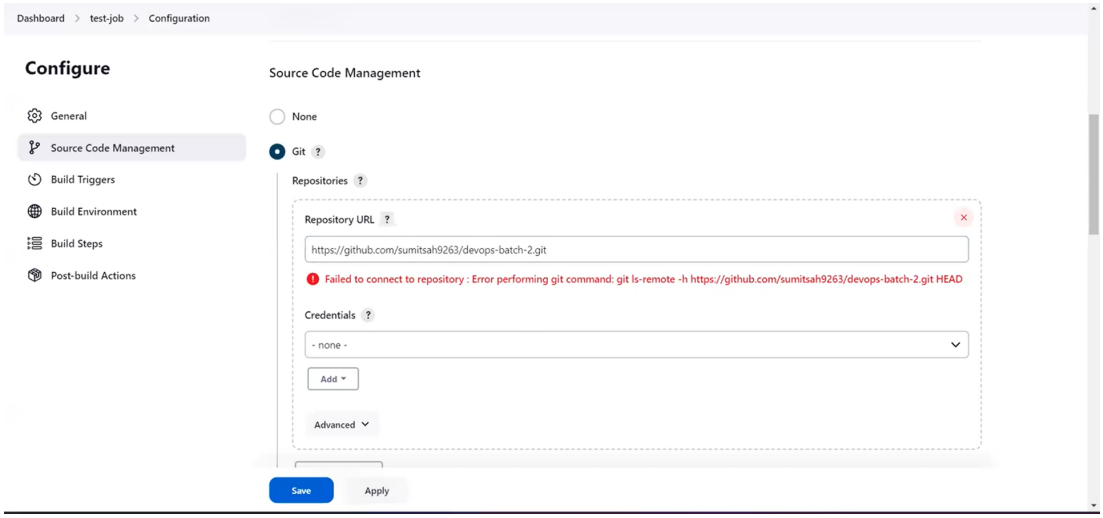
Go to configure

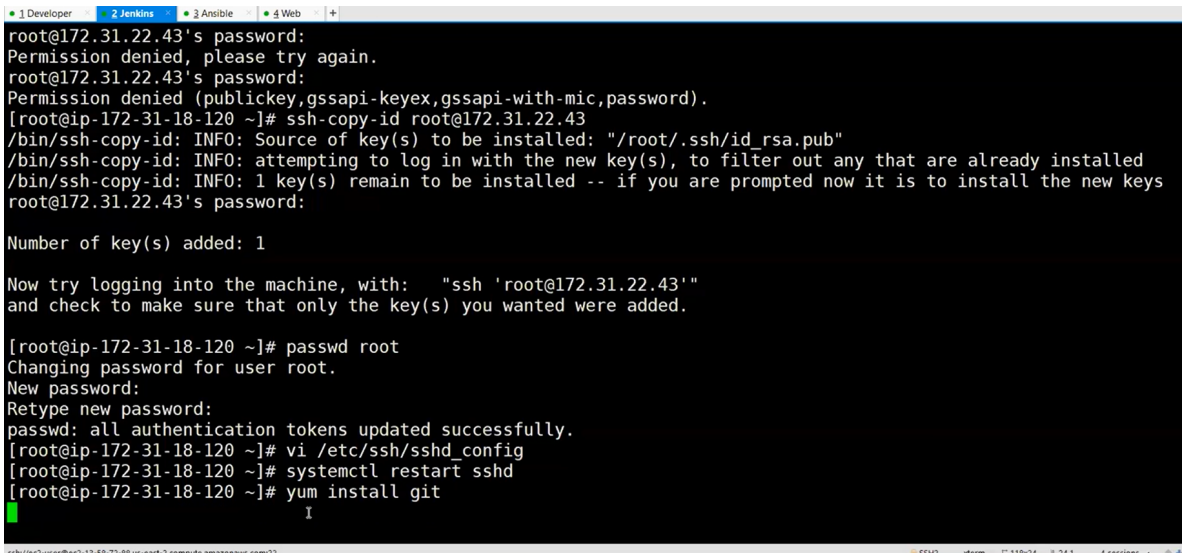




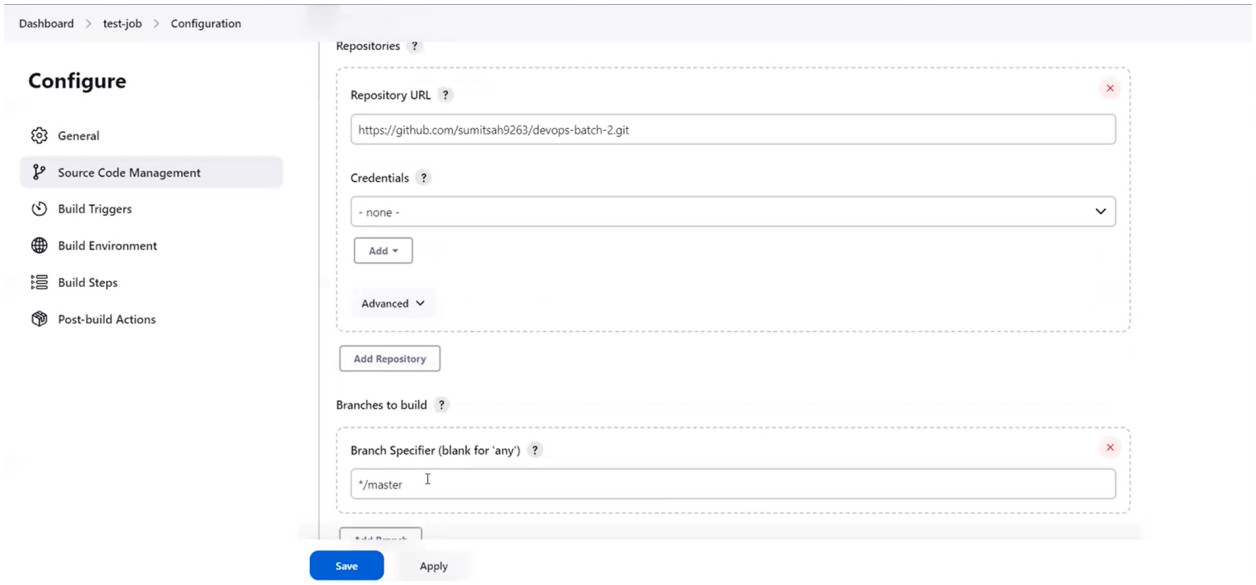


we see error because there is no git package installed in the jenkins server

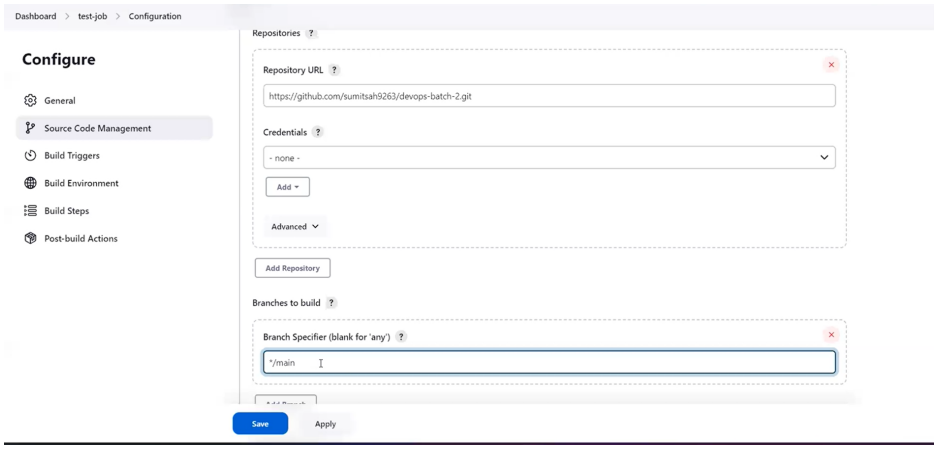


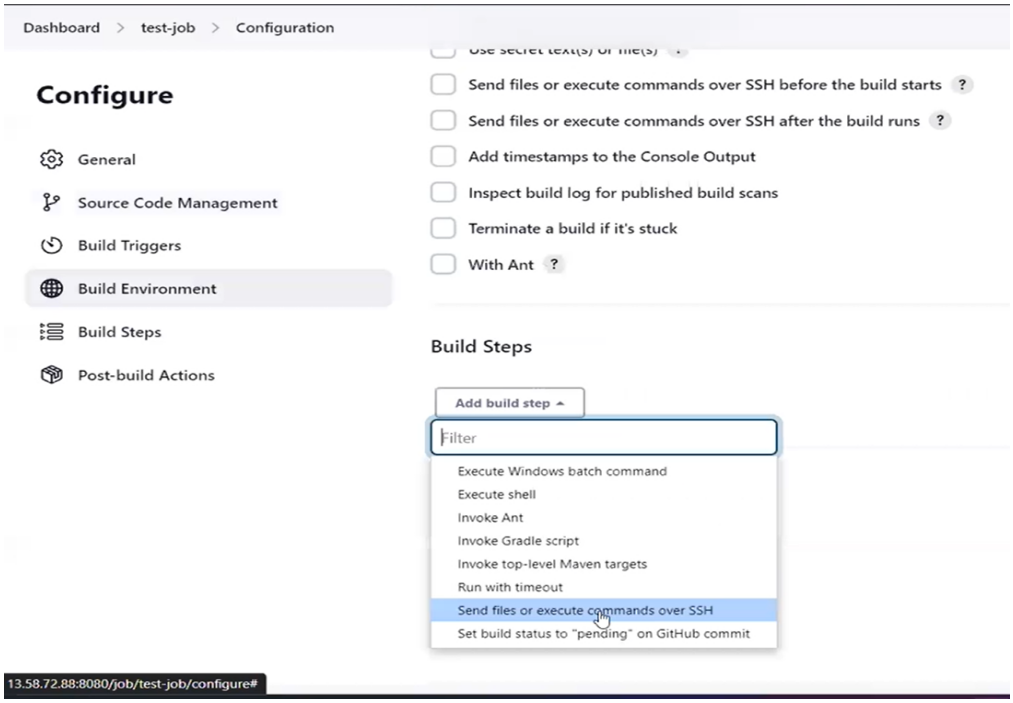


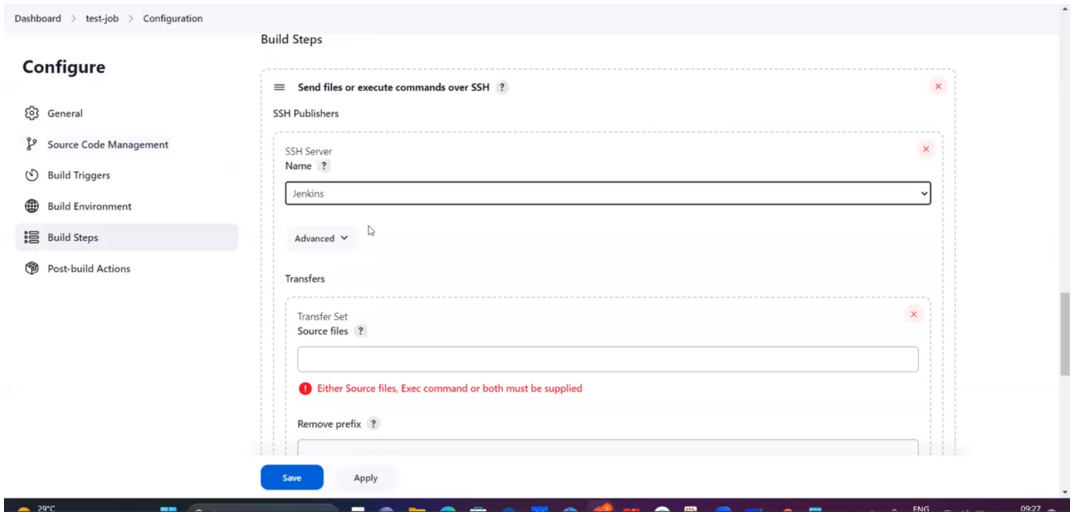
now we can see that there is no error

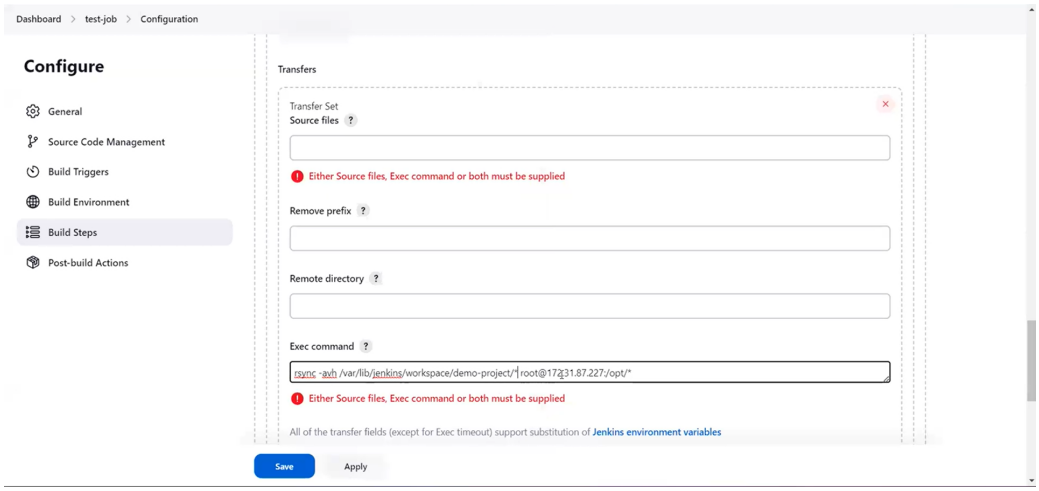


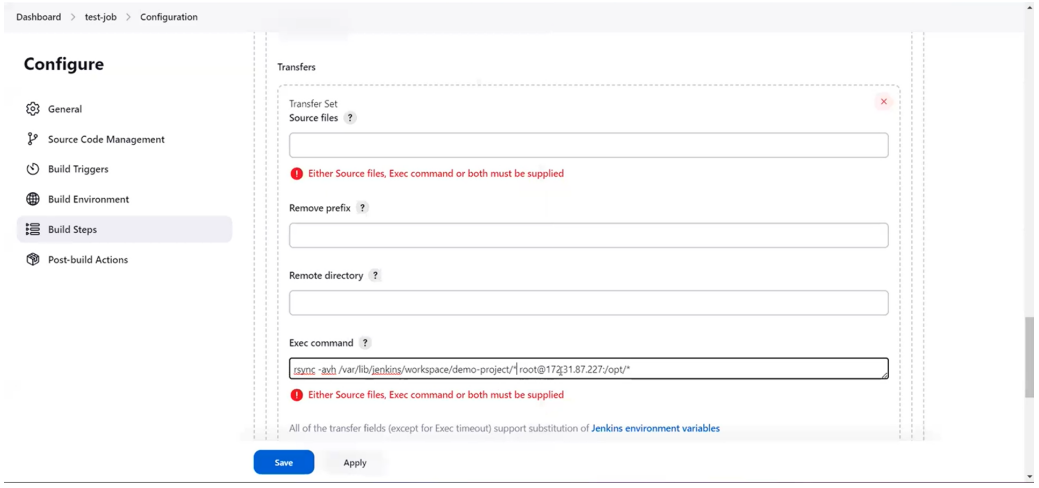
Change master to main



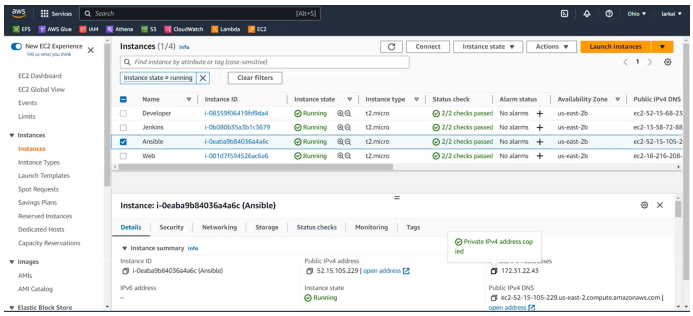






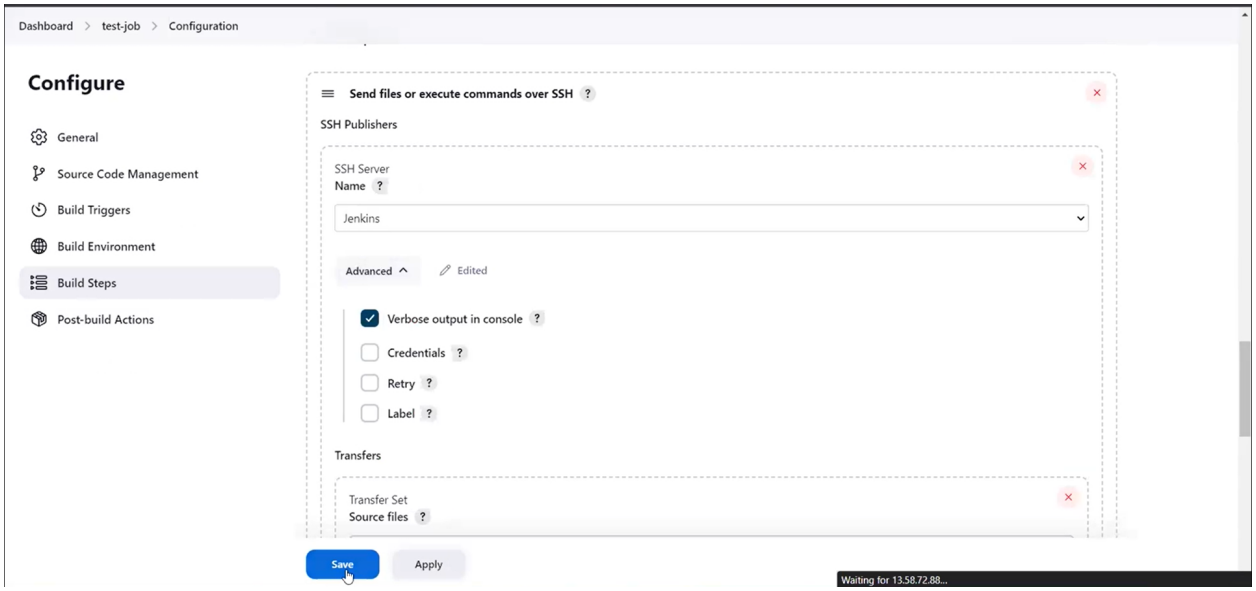


Changing demo-project to test-job and writing index.html and changing the ip address(private) of the ansible server.

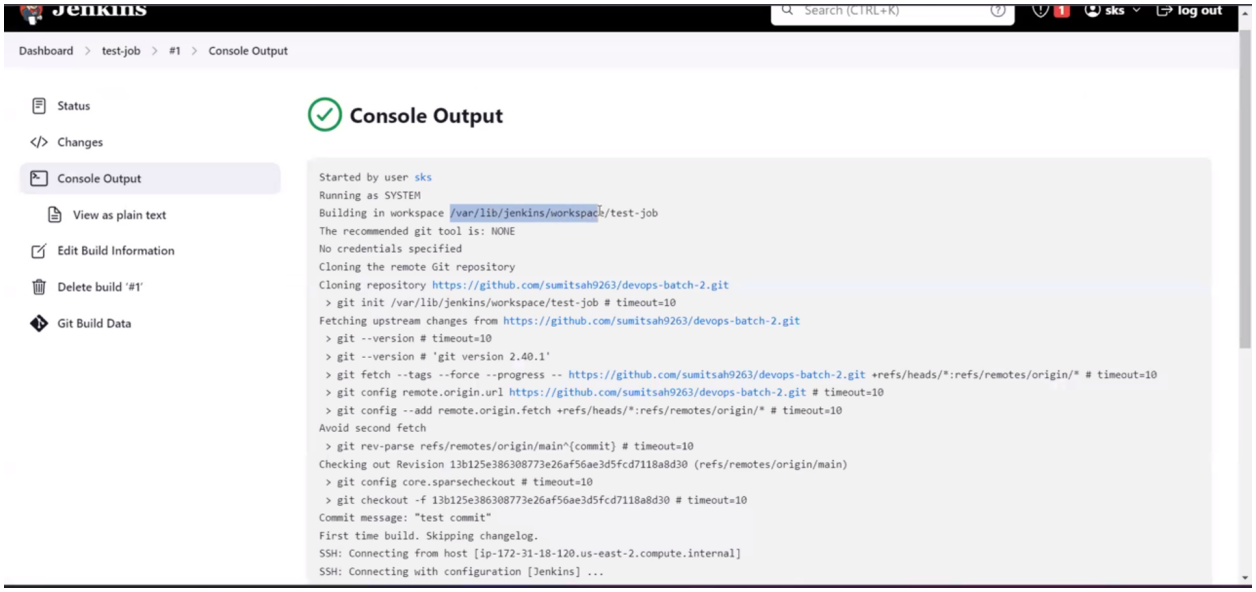


Humlog jenkins ke file ko ansible ke sath sync kar rhe aur jenkins ka location diya hua hai jahan pe index.html hai aur ansible ka location diya hua hai jahan pe file ko move kr rhe hai

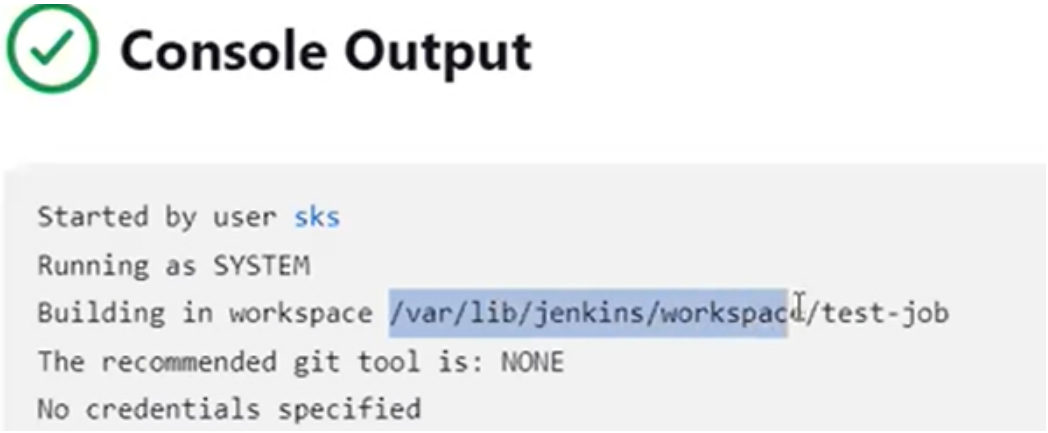
We forgot to do one more configuration
So check the verbose output in console



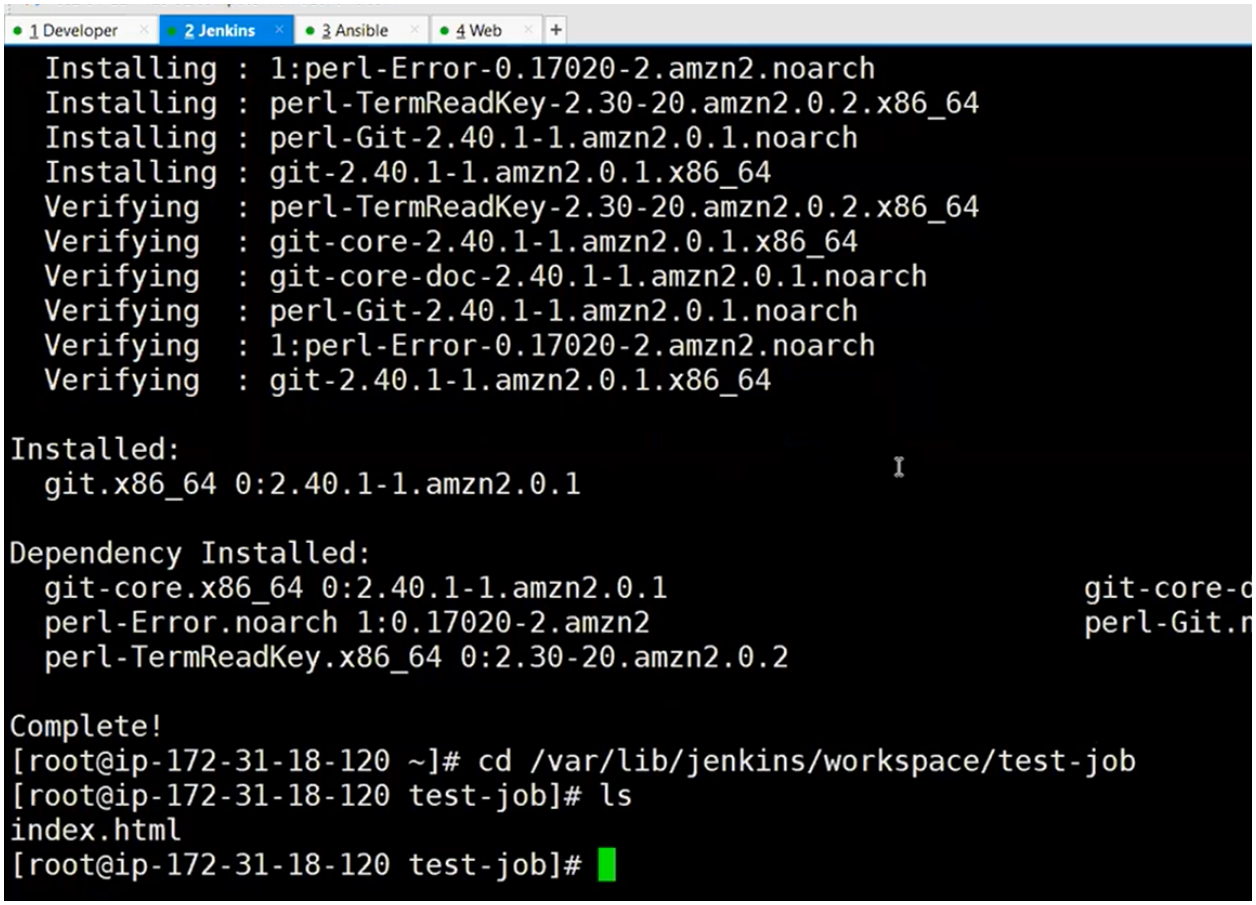
By enabling this we can see these stuffs



Ye iss name ka ek folder bana dega

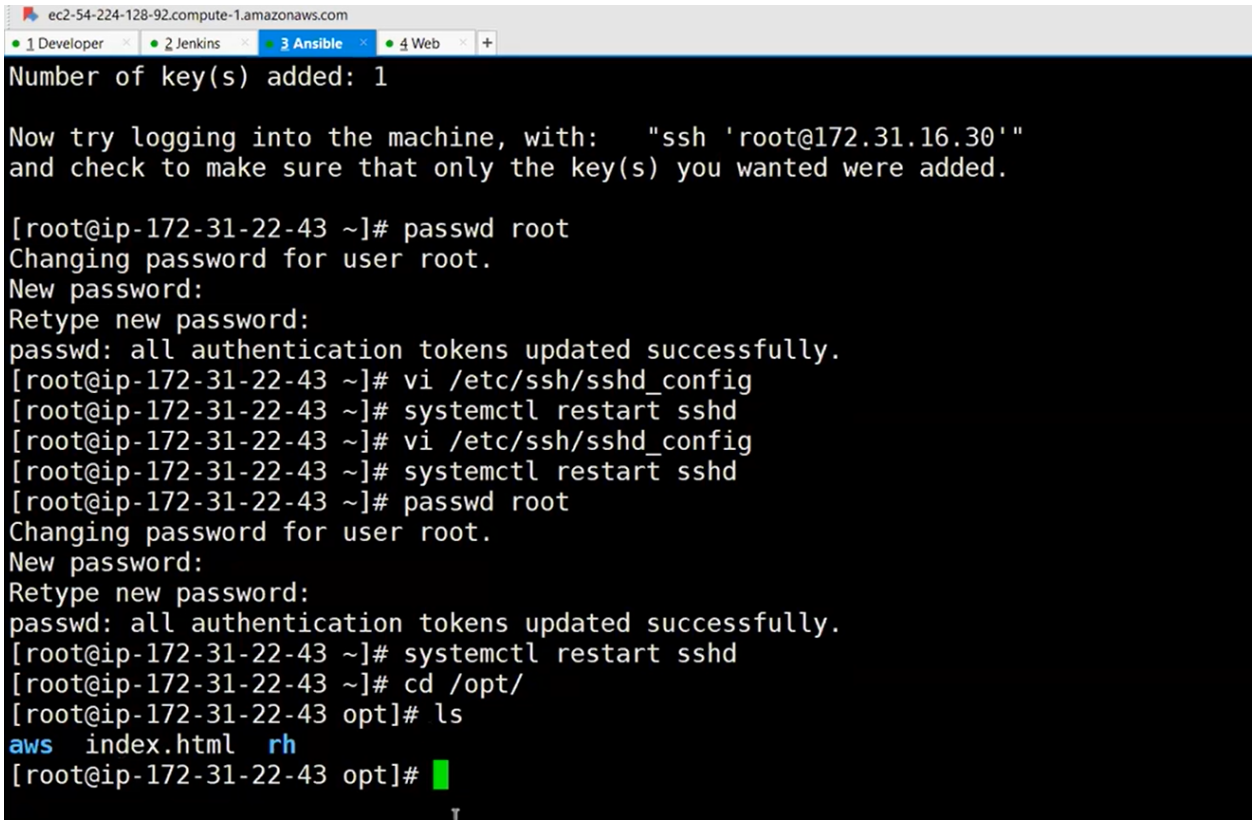


If we check it in the path we can see the index.html file



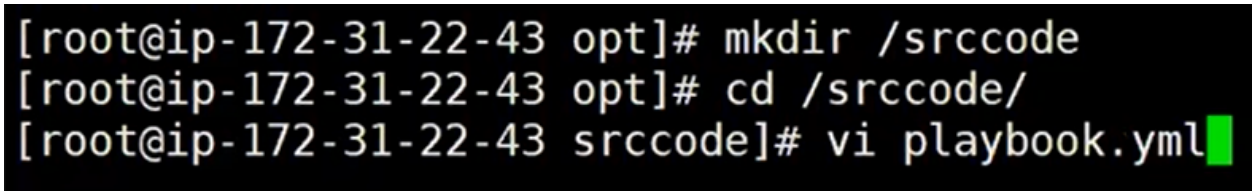
```
SSH: Opening exec channel ...
SSH: EXEC: channel open
SSH: EXEC: STDOUT/STDERR from command [rsync -avh /var/lib/jenkins/workspace/test-job/index.html root@172.31.22.43:/opt/index.html] ...
SSH: EXEC: connected
```

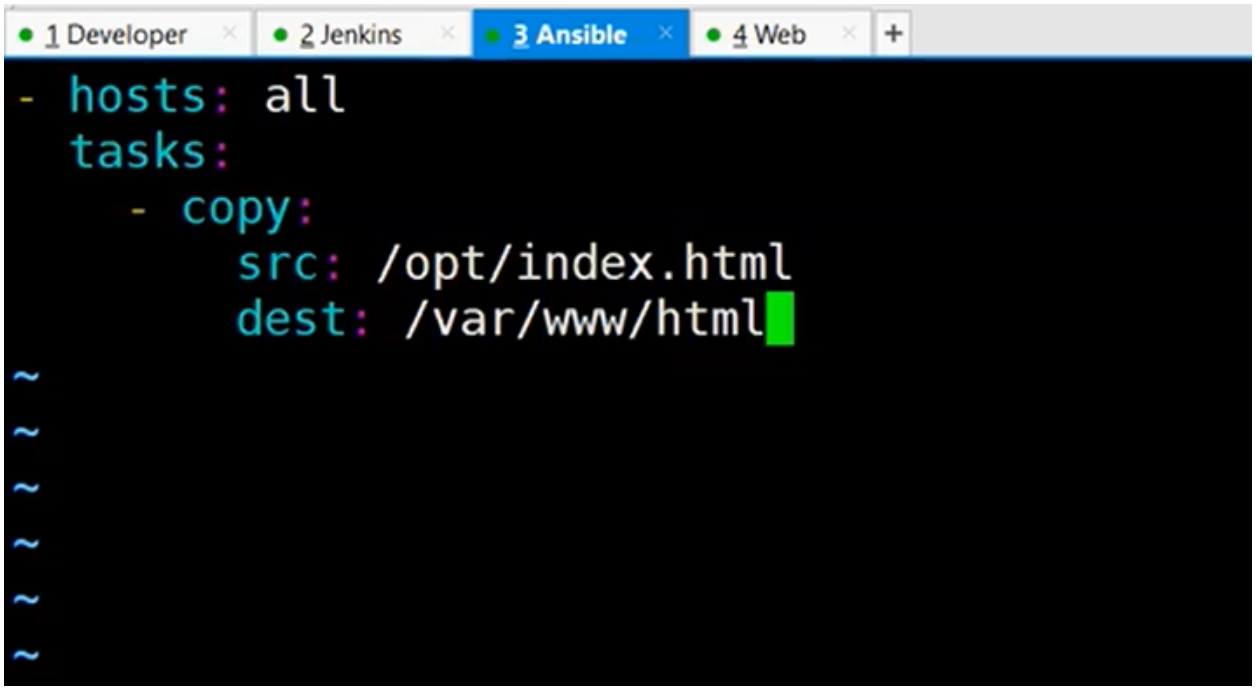
We have moved jenkins file to ansible so we can see the files in ansible server also



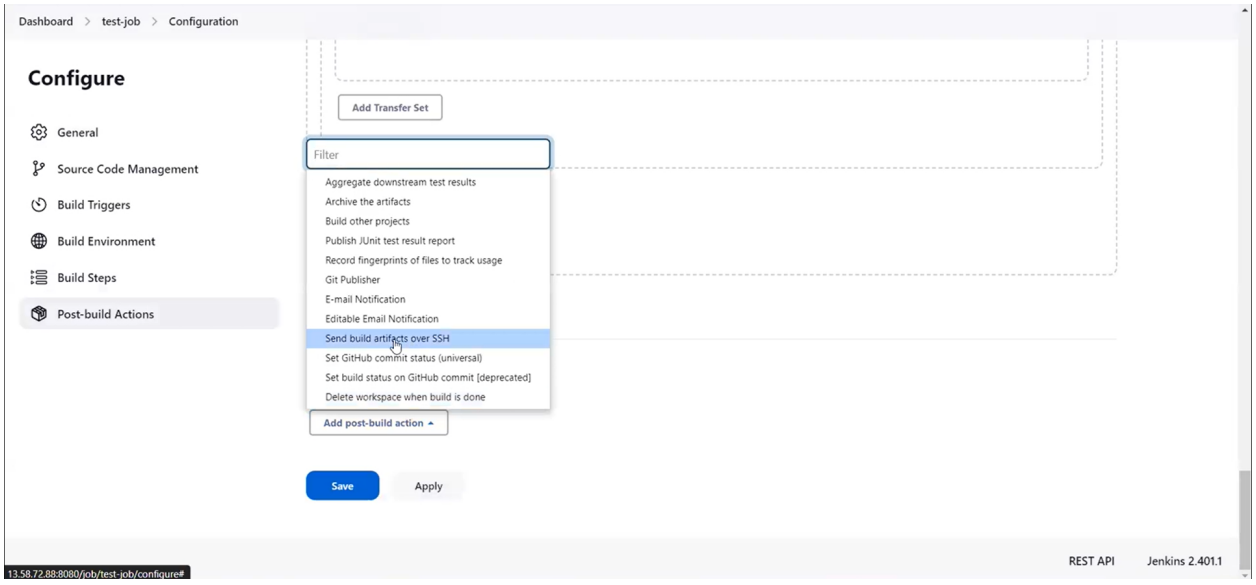
Ab ansible me ek play book banayenge
Play book ek file hota hai

File banane ke liye following steps

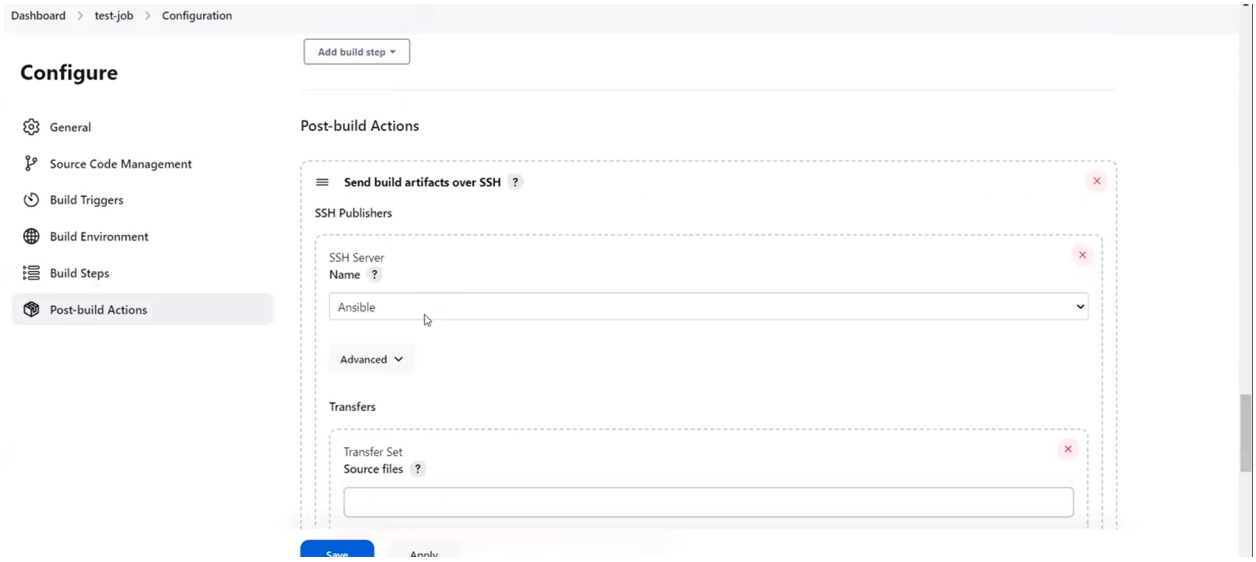




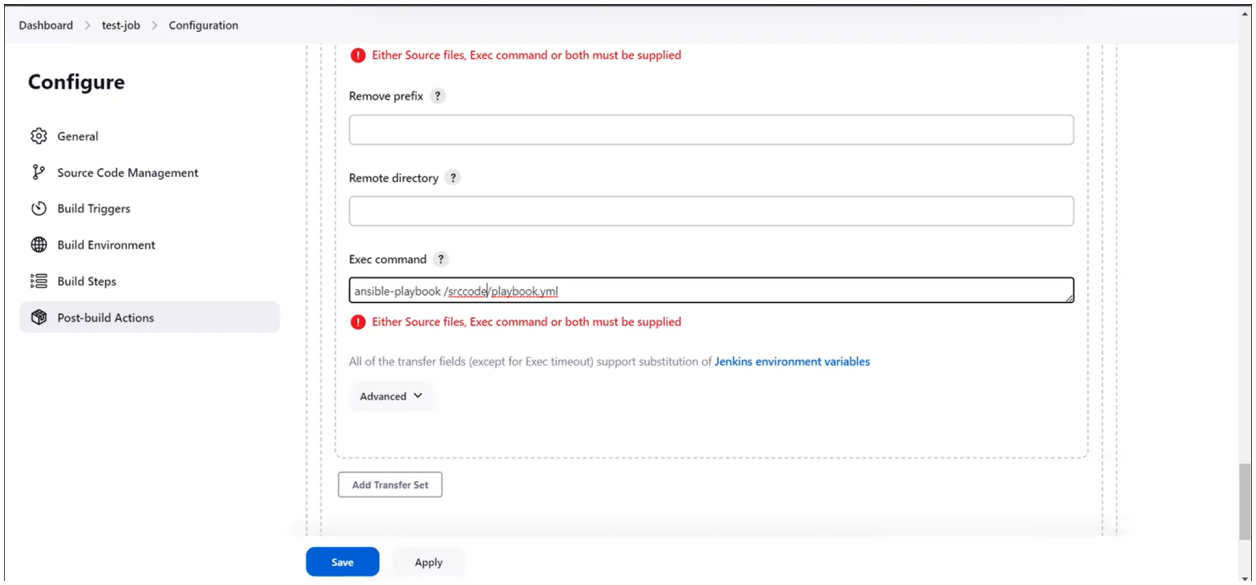
Now on configure Post build actions
Send build artifacts over ssh



We select the ansible server



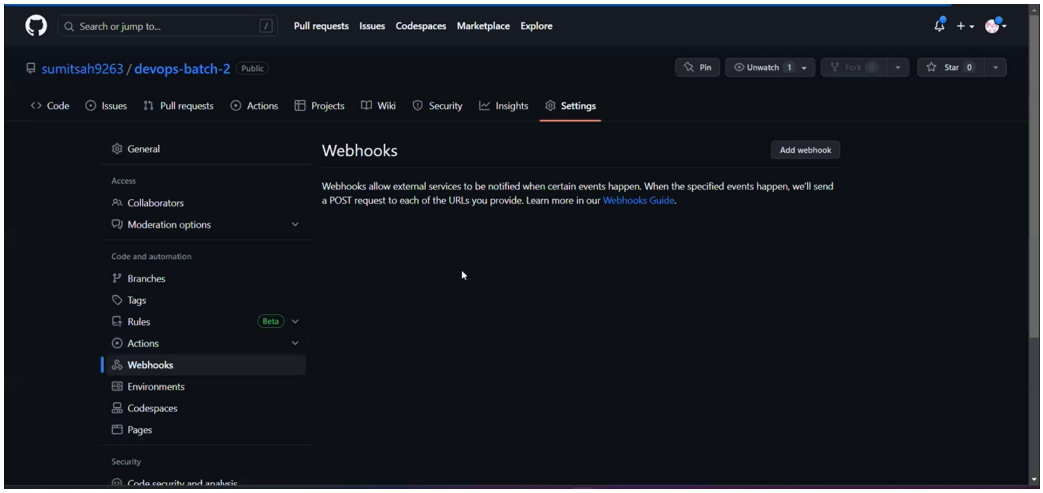
Copy the code

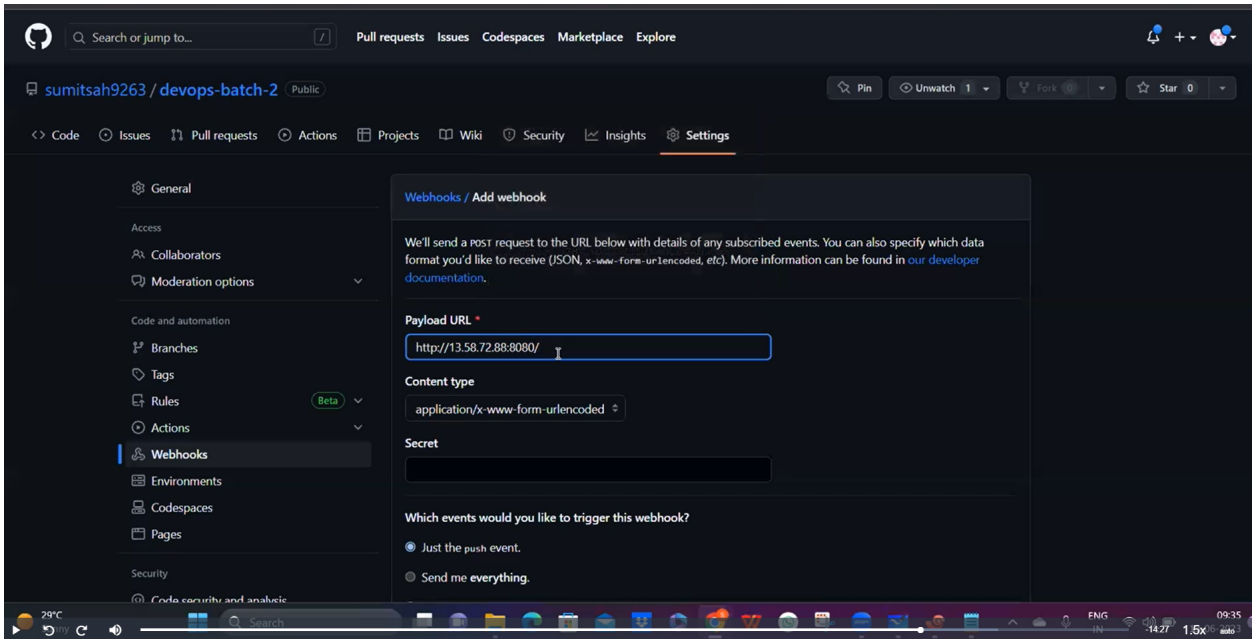


```
1 Developer | 2 Jenkins | 3 Ansible | 4 Web
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# cd /opt/
[root@ip-172-31-22-43 opt]# ls
aws  index.html  rh
[root@ip-172-31-22-43 opt]# cat index.html
Hello welcome to ITW silicon
test second
[root@ip-172-31-22-43 opt]# mkdir /srccode
[root@ip-172-31-22-43 opt]# cd /srccode/
[root@ip-172-31-22-43 srccode]# vi playbook.yml
[root@ip-172-31-22-43 srccode]# pwd
/srccode
[root@ip-172-31-22-43 srccode]#
```

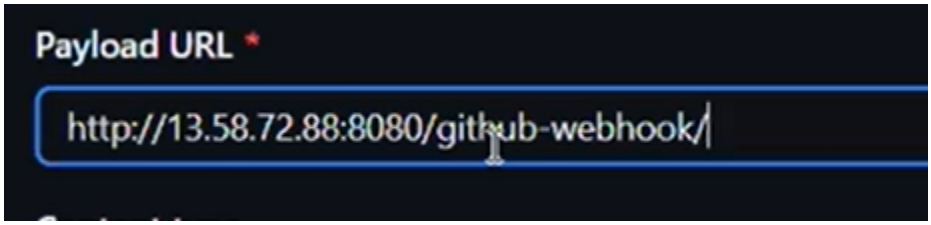
Now if we hit the public id of the web server we can see the content

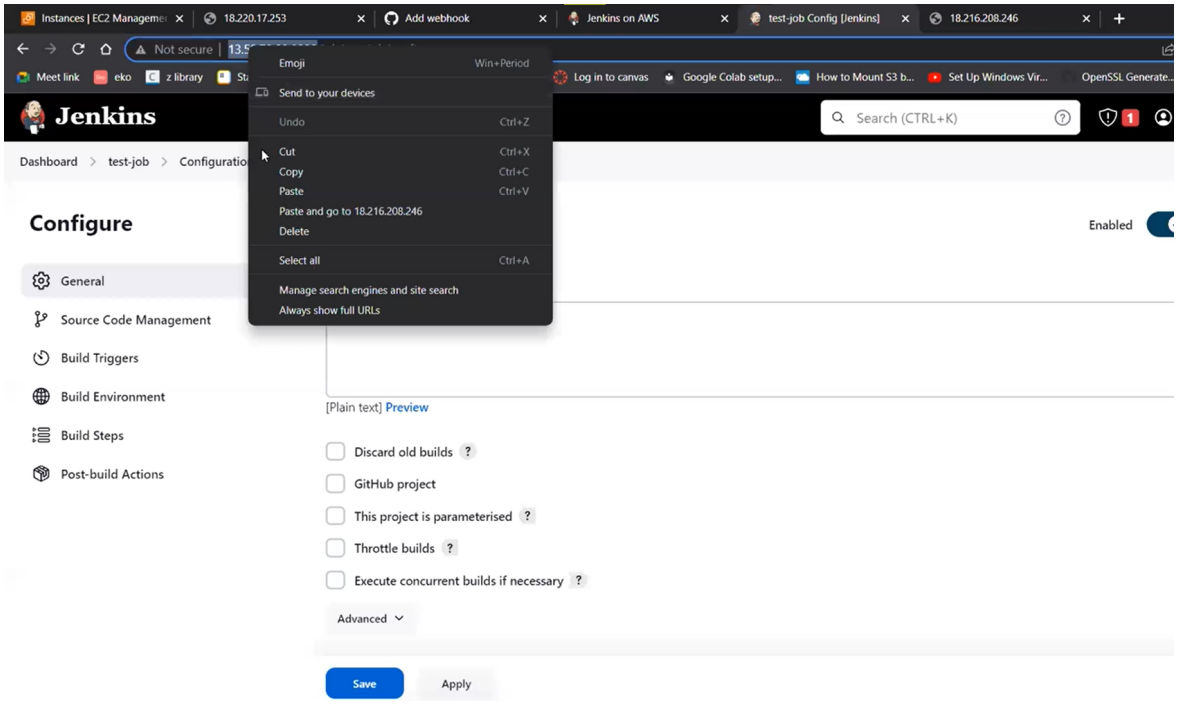
So now we need to add webhooks so that any updates get automatically pushed to the gituhub and shown in the web server. Add webhooks



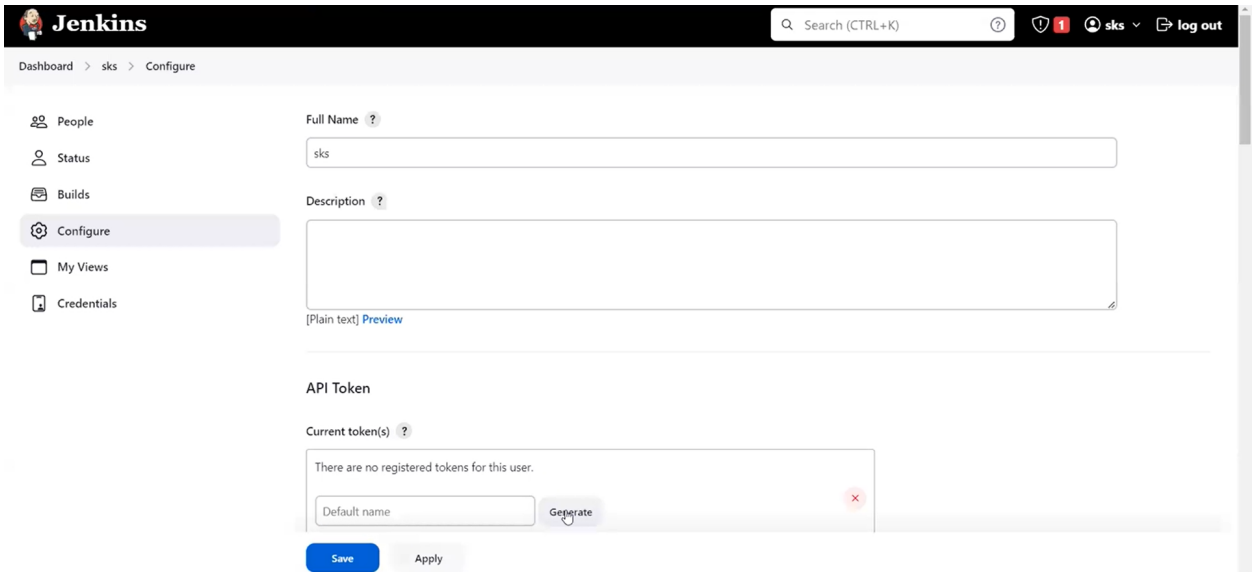


Jenkins ka url daalna hai aur ek chiz add krna hai "github-webhook/"

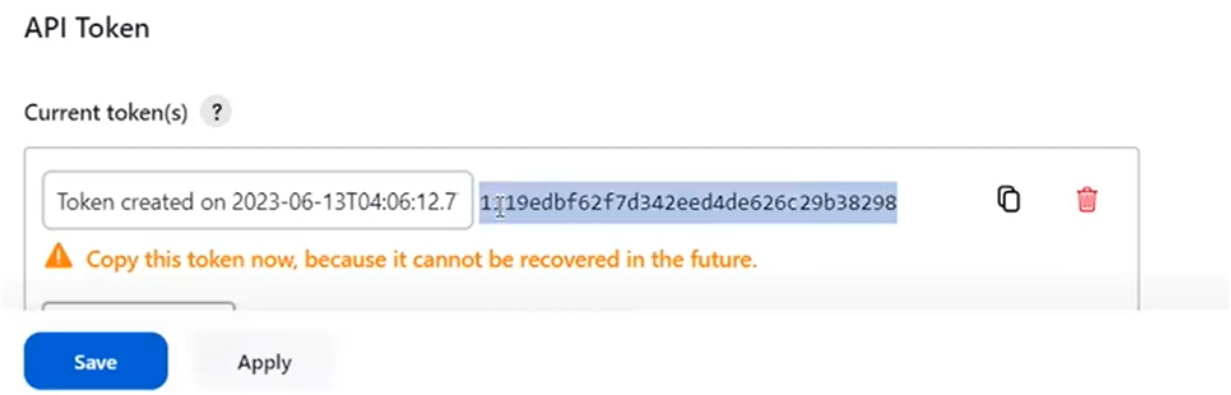


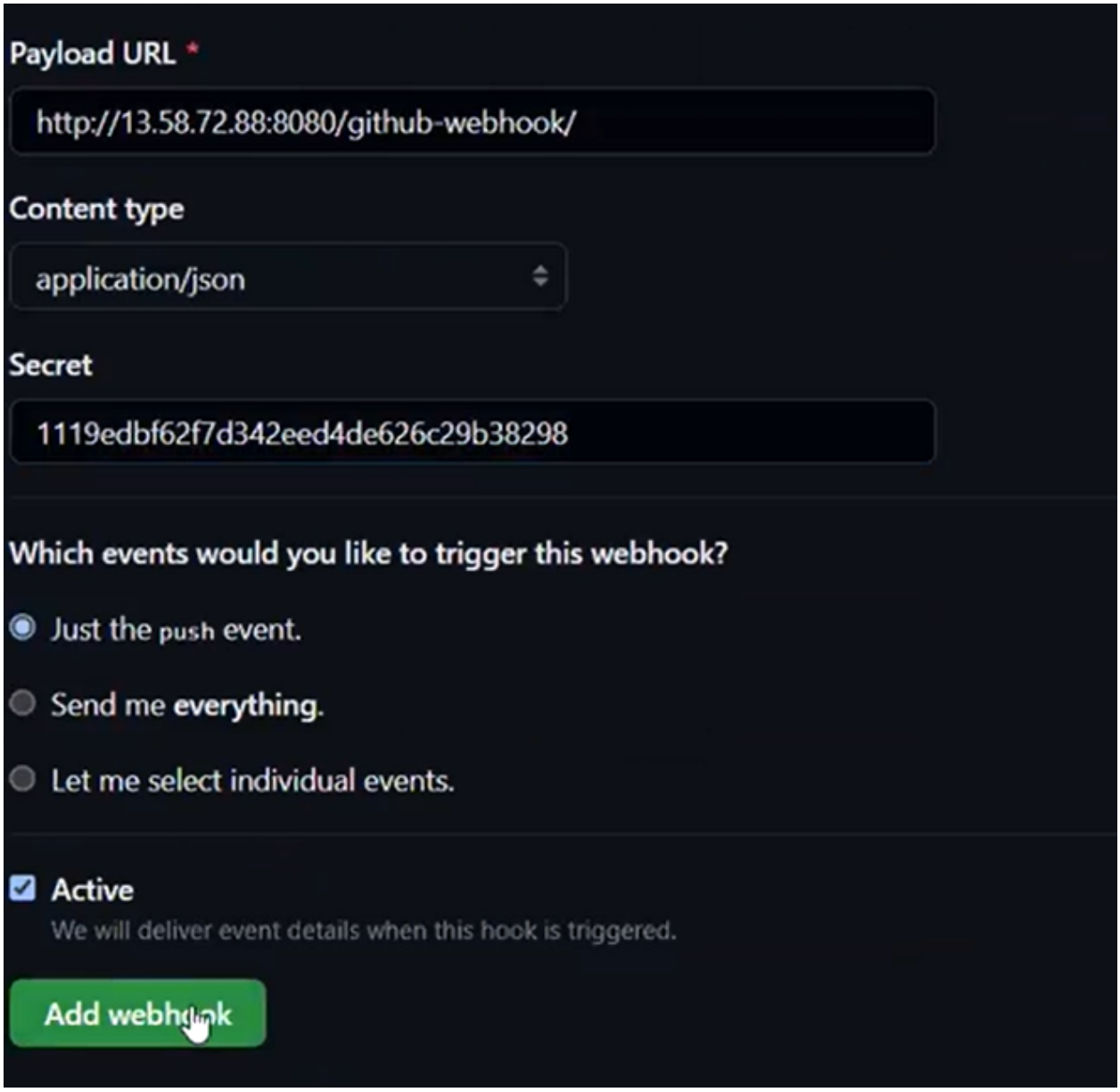


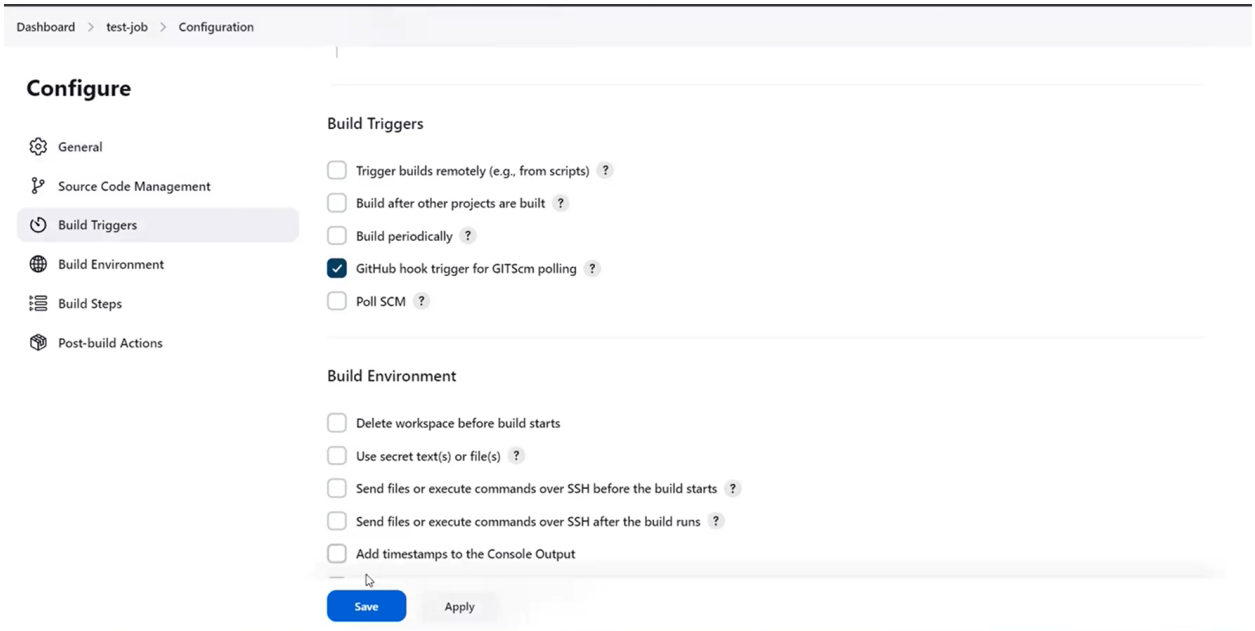
Now we need to give secret key , so for that go to the jenkins configure
Click generate



Then we have to just copy this and apply




Payload URL *
http://13.58.72.88:8080/github-webhook/
Content type
application/json
Secret
1119edbf62f7d342eed4de626c29b38298
Which events would you like to trigger this webhook?
Just the push event.
Send me everything.
Let me select individual events.
Active
We will deliver event details when this hook is triggered.
Add webhook



Now everything is automated. If we push anything in the github then that will automatically be kept track by jenkins and reflected to the web server.