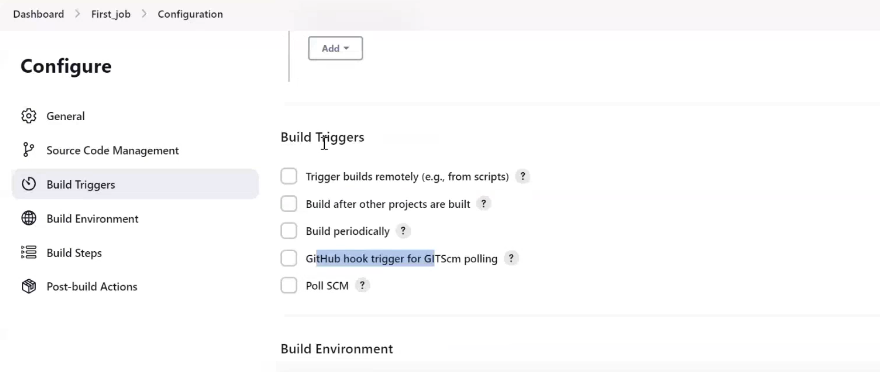
1. Configure 2 slave machines in Jenkins master.
2. Configure webhooks to Jenkins job.
3. Configure poll scm and build periodical options in Jenkins job.
4. Take backup of Jenkins server by using bash script.
5. Take backup of Jenkins using rethin backup plugin.
6. Setup a new Jenkins server and dump the backup taken in task4.

1Q: Configure 2 slave machines in Jenkins master.



Q: What is the use of build trigger?

A: With the help of BT we can decide when our job should trigger. Means this job automatically trigger without any manual intervention.

Webhooks: whenever there is changea in the source code, means whenever a new commit happends on our source code then automatically this job get trigger.

Webhooks are configured for automatic deployement, automatic CI/CD for development environment.

We can’t configure webhooks for production. Because we don’t want any change without our idea. Means every change we need to trigger, in production also, but should not be without our idea.

Whereas for development bcz there will be a lot of development activity happening for every few seconds or few minutes there will be some new code which will be push to our repository and they want to deploy and they want to test application. In this case

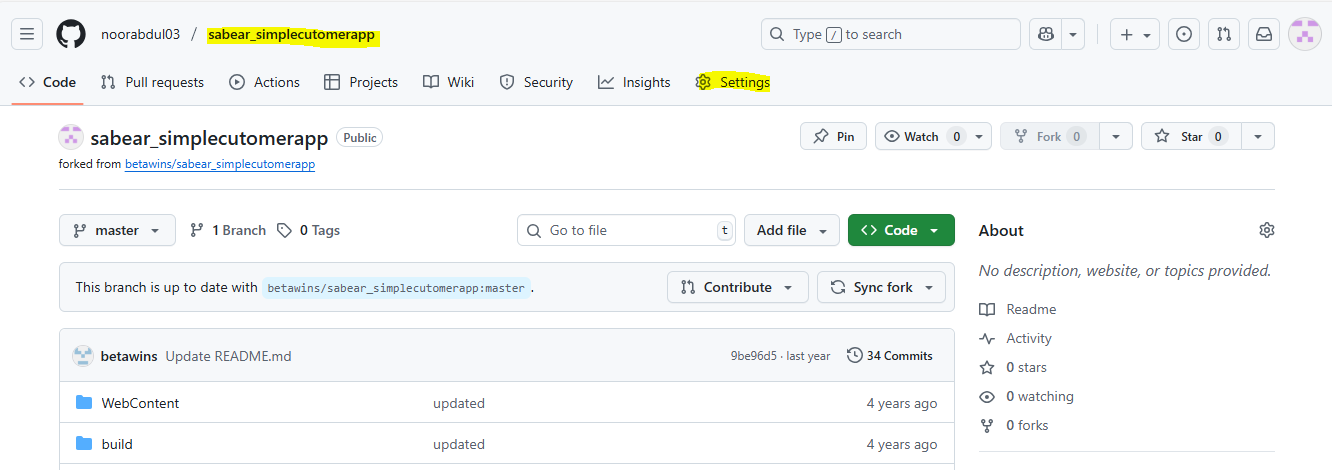
Ex: Develop has been new that he has completed the code, so we can’t login or raised the ticket as build trigger and we can’t build trigger bc he will be not having the access.

So we configure webhooks.

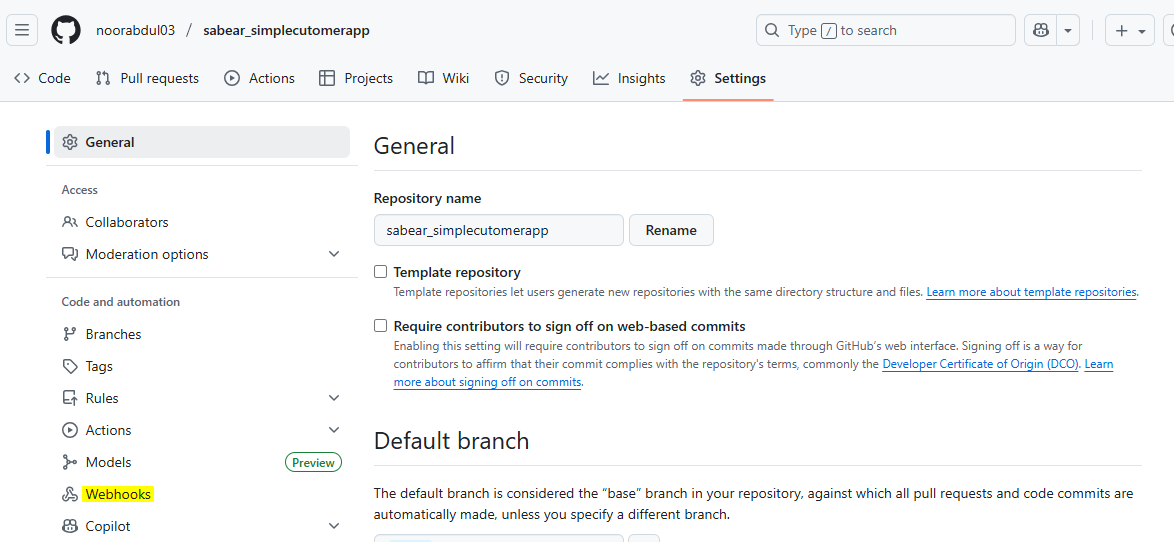
Q: How to configure webhooks?

Step1: Goto your github repository as below for sabear\_simplecutomerapp code I need to configure web hooks

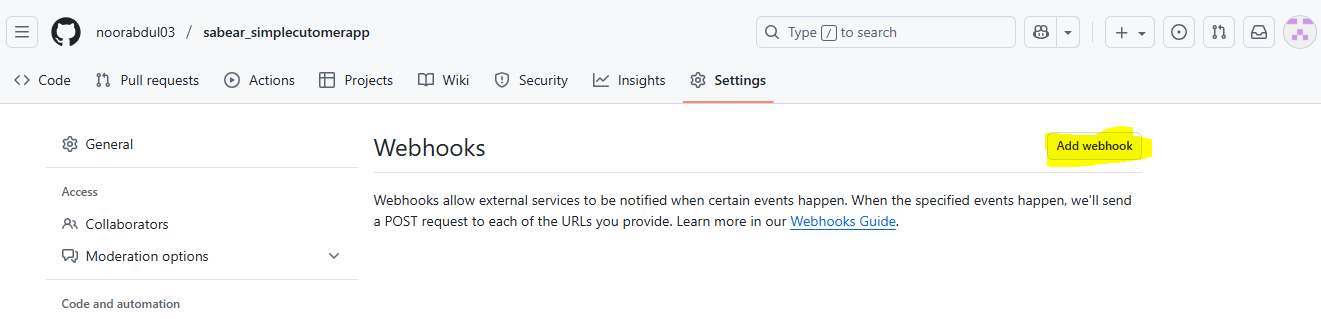
Go to settings



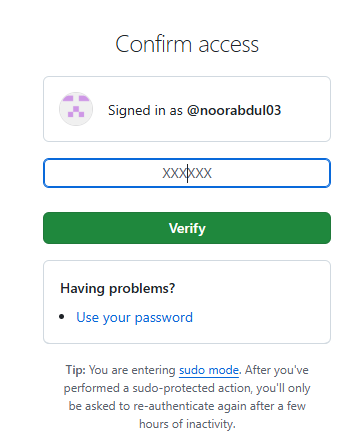
Click on webhooks



Click on Add webhook



Enter MFA



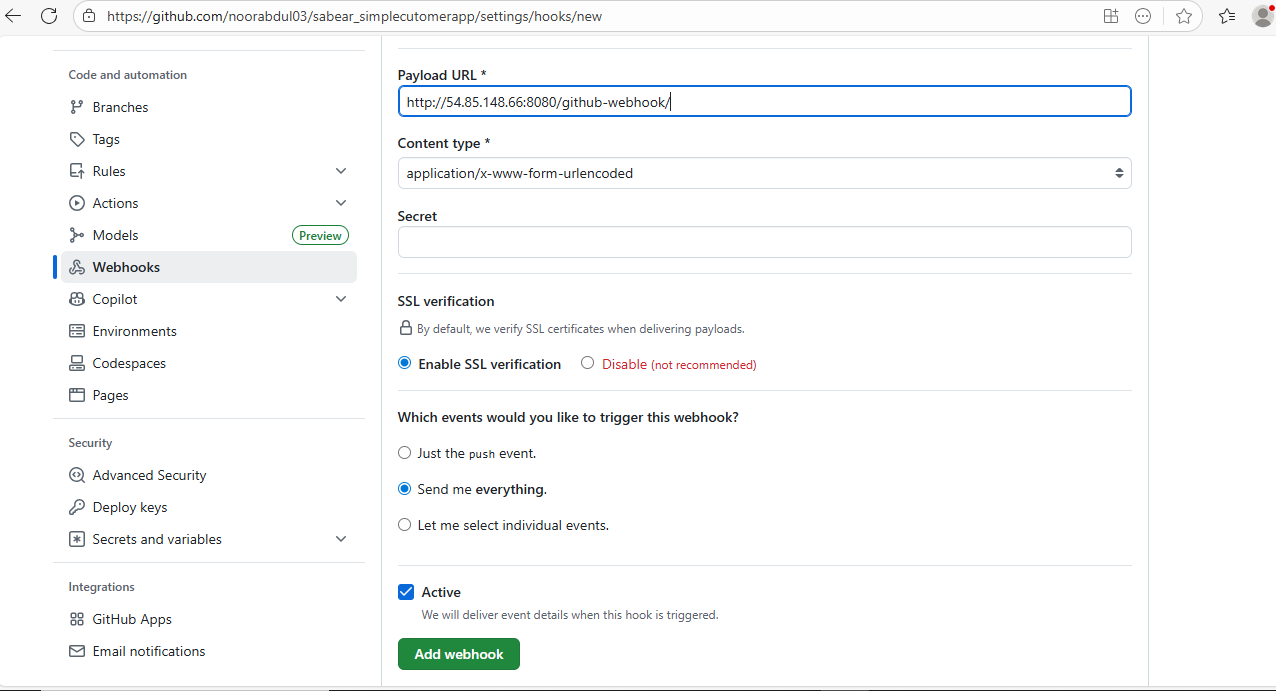
Step 2: Enter the payload URL

Q: What is payload URL

A: Means on which particular URL this needs to be configure.

So we have Jenkins URL 

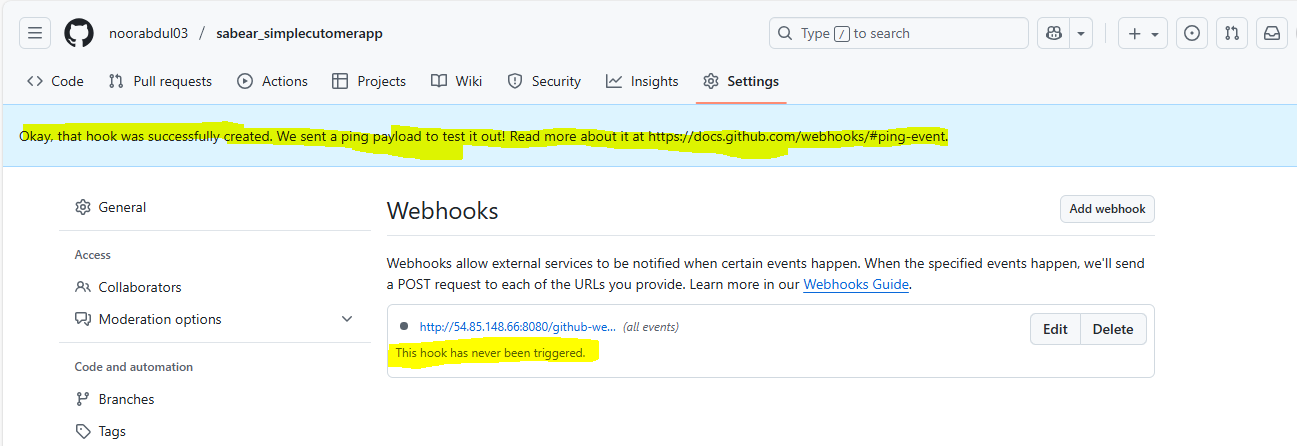
Configure this: copy URL and paste in payload URL



Here we can select which event needs to be trigger. Means anything which is been send to particular repository or any changes, so we can decide what need to be trigger.

Select send me everything: means whatever changes will implemented weather it is a PR weather it is a any commit then this payload URL should trigger our Jenkins jobs.

Click on Add webhook

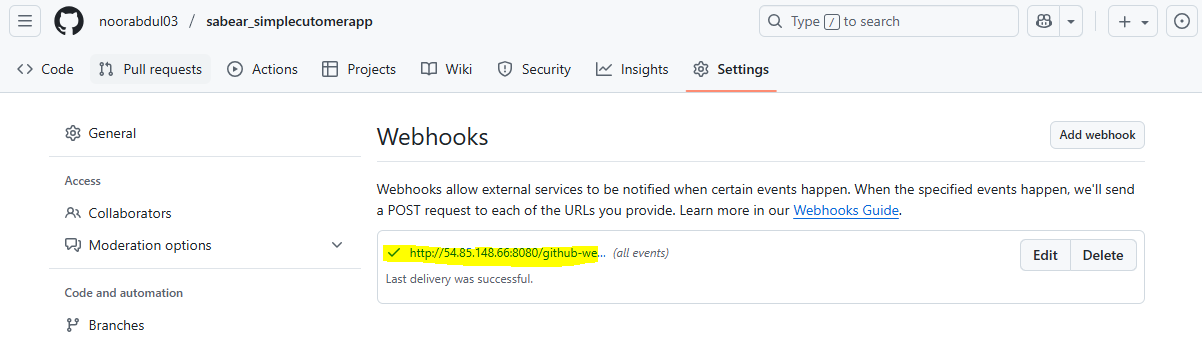


It is in indication means, not yet successful. So what it will do, it will sendout some text message to my JENKINS server. So once the message has been successfully delivered then we can see above as green one.

So once the test successfully validated then only in green colour.

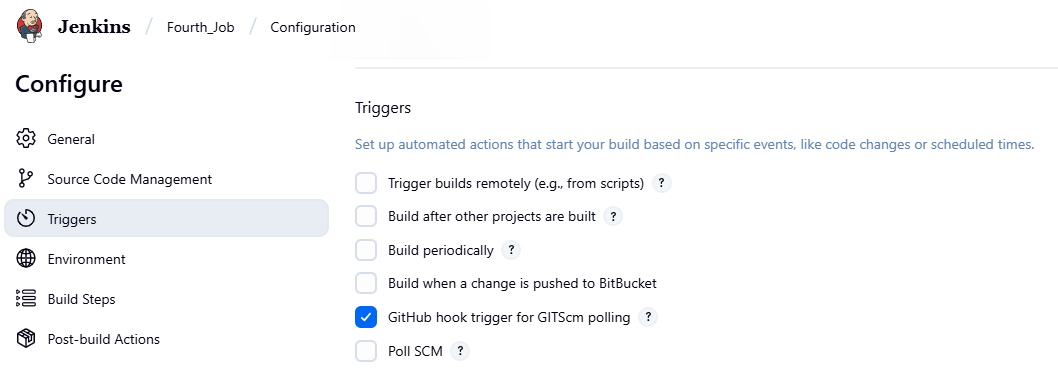
Refresh it only.

Now it was able to test our and successfully delivered the message



Step 3 : Once done, Go to Jenkins job (Fourth\_Job)

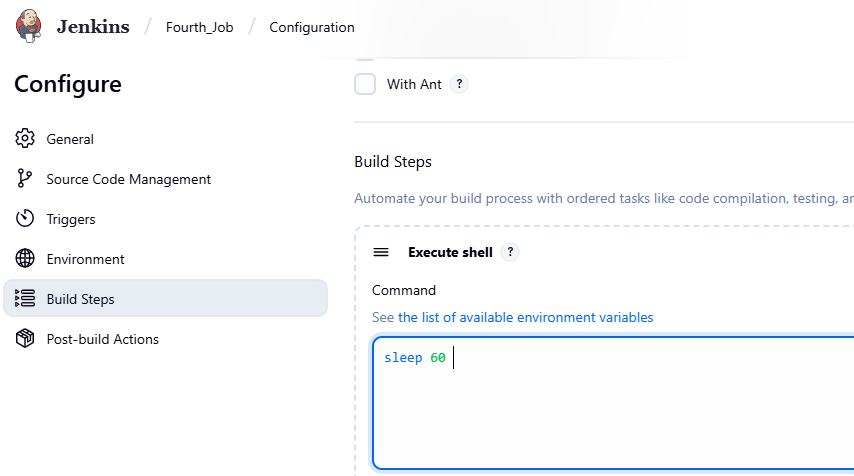
Just Enable Github hook trigger or GITScm polling



GITscm : scm: source code management

Once this job has been enable what will happen to our payload URL to our particular job we have synchronized, we have a connections Click on Save to this job.

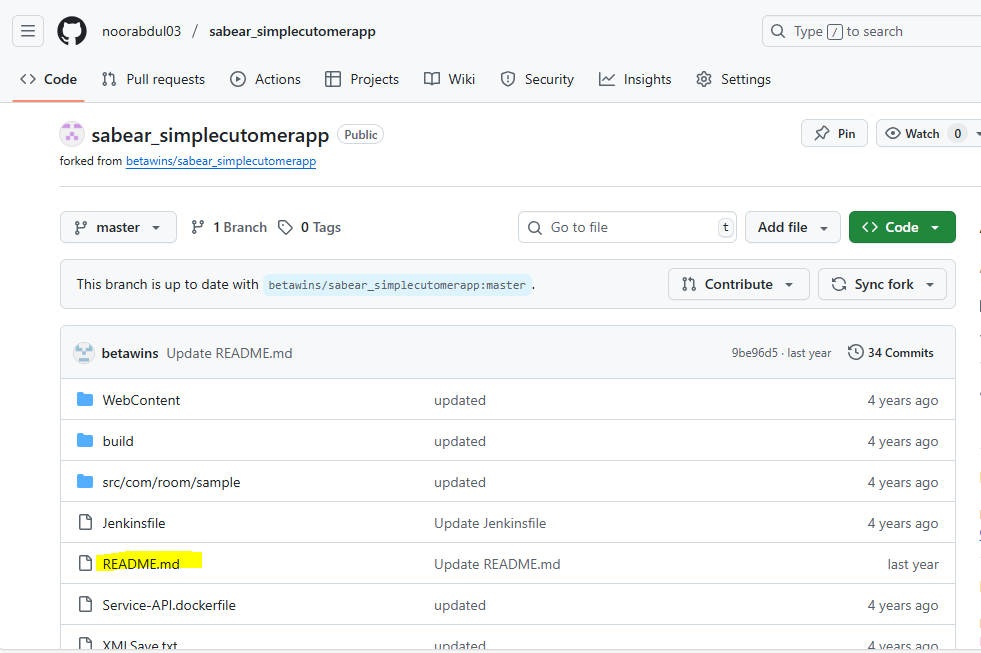
In this job, only for hold 2 seconds, I will add one basic sleep command. Sleep command hold my cursor for 60 seconds.



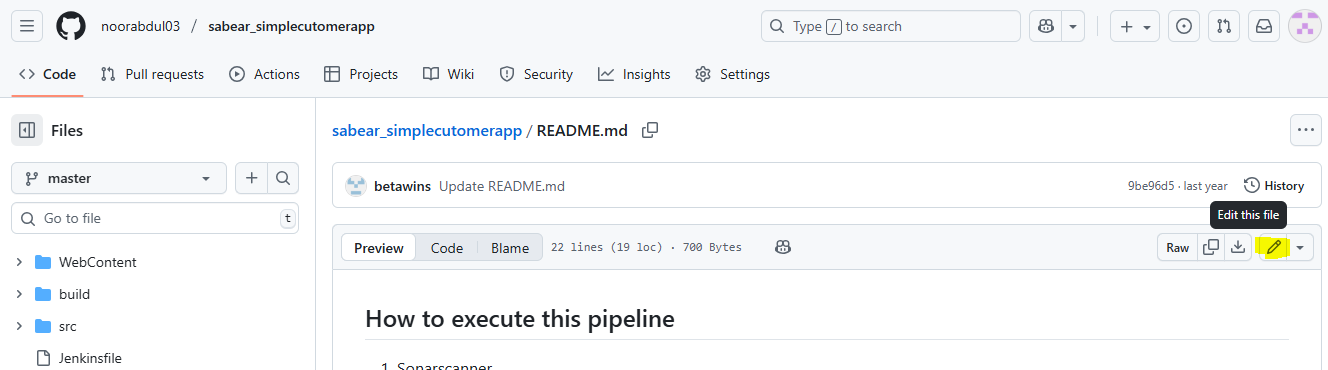
To view the output only

Click save

Step 4: Before deploying or executing job Go to GITHub to my code and I am trying to update my code after doing some changes.



Click on README.md



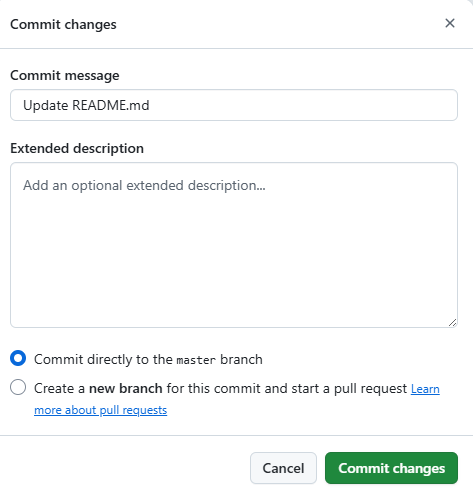
Click on Edit

And just or example (Add one . dot in this particular ffile and will update my commit)

Click on commit changes

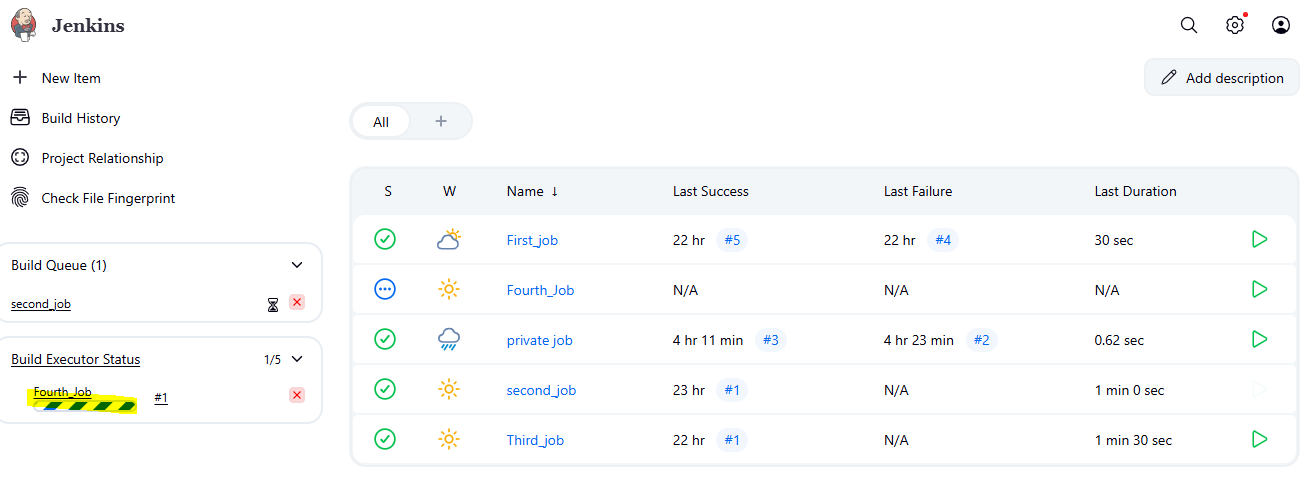


Click on commit changes again



So what happened, we have updated our commit. Means there is a new change in source code.

So what should happened or Jenkins should automatically trigger (see Fouth Job)



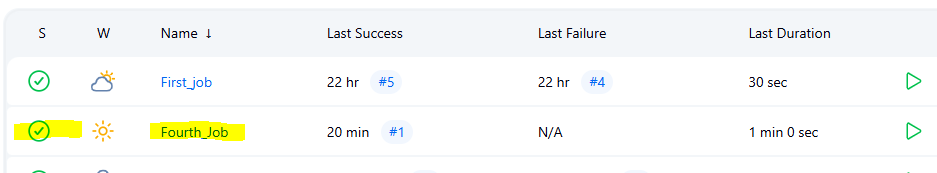
See this is automatically getting triggered without any manual intervention.

Webhooks Very important:

This is use for development environment.

So whenever developers are making any changes for that development environment we are going configure webhook. because everytime I will not be available to come to this job and doing click on BUILD NOW .

See the job has successfully completed.

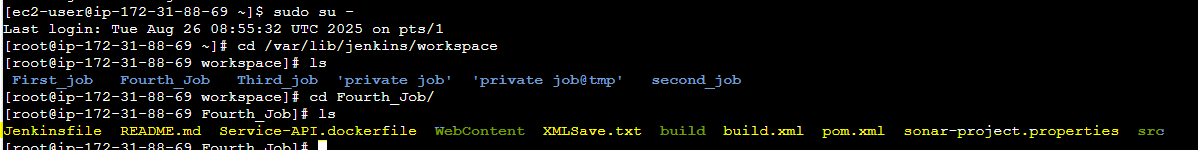


Interview question: What is webhooks?

Q: Where is my source code downloaded?

In my Jenkins server Ec2

Cd /var/lib/Jenkins/workspace/

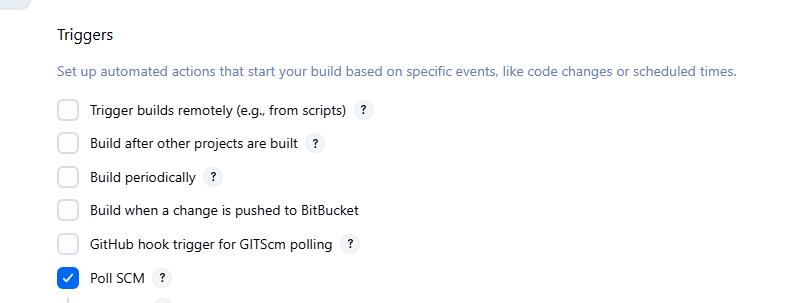


**All our souce code downloaded here. Means whatever we are trying to run on the Jenkins server we can find all the details in our workspace only.**

**Now**

**Poll SCM**

3) Configure poll scm and build periodical options in Jenkins job.



If I click on poll SCM

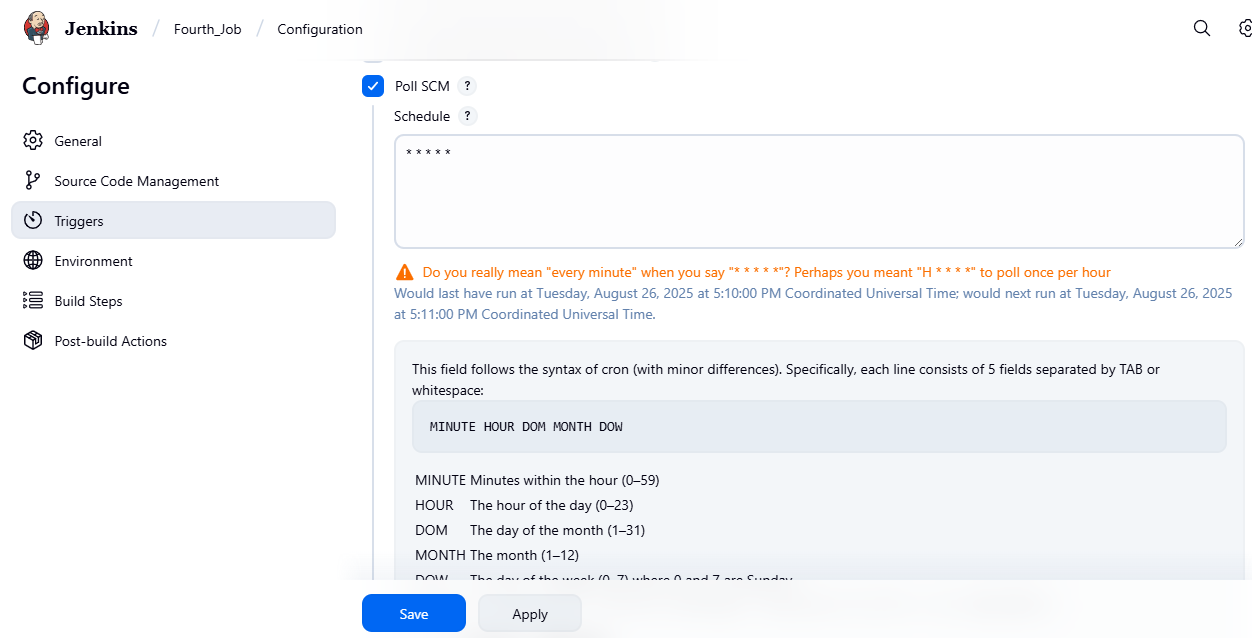
And I click on schchedule ? option Below I can see some information

This like a cron jobs, we can schedule at this particular time this job should get trigger

5 stars \*\*\*\*\* means every minute every hour every day every month every week

This job should get triggered

Click on save



Click on save

But I you see it wont show any changes in our job.

Now remove poll SCM and

select Build periodically and And I click on schchedule ? option Below I can see some information

This like a cron jobs, we can schedule at this particular time this job should get trigger

5 stars \*\*\*\*\* means every minute every hour every day every month every week

This job should get triggered

Click on save

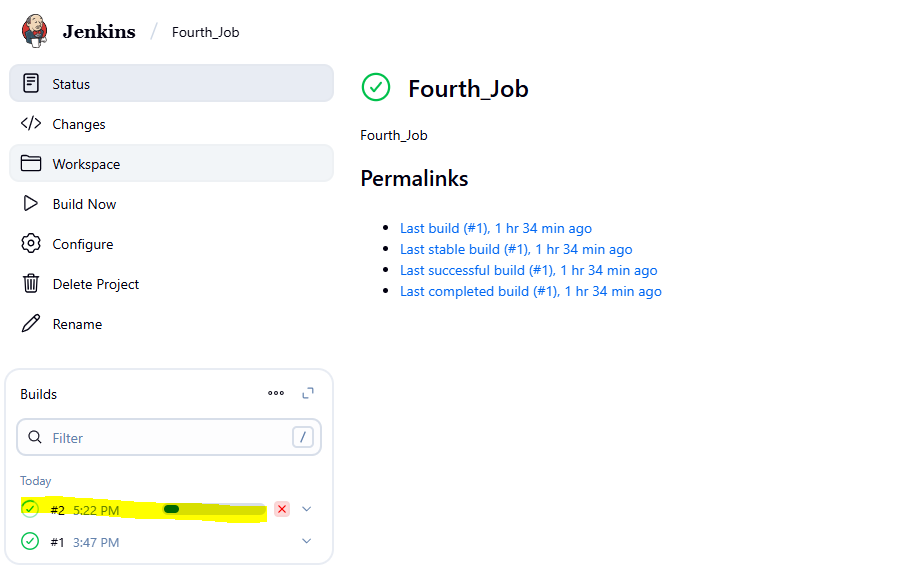
Check what happen? (Bcz we are having same fetaures in Build periodically and poll scm)

Now let us wait or 1 minute



Click save

See whats happen



See this job has triggered automatically If we keep Build periodicall and cron job.

**Q: Why the job didn’t triggered when Keep Poll scm**

**And what is the difference between build perical and poll scm**

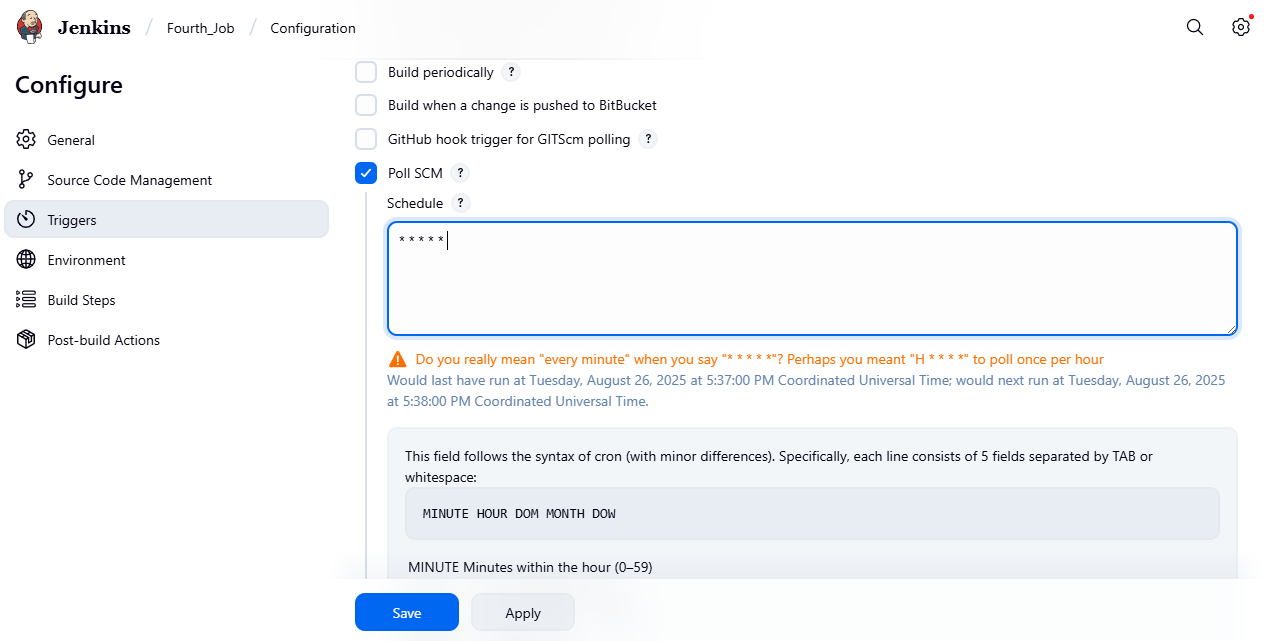
When I am keeping poll scm why it is not triggering bca it is going github repository and checking If there is any change or not.

If there is any change then only poll scm will trigger my job. If there is no change then only poll scm will never get trigger

\*\*\*where as build periodically what is doing?

Whether there is change or no change in code, it should trigger based upon the timing

**Task 3 : Now go to job and select poll scm**

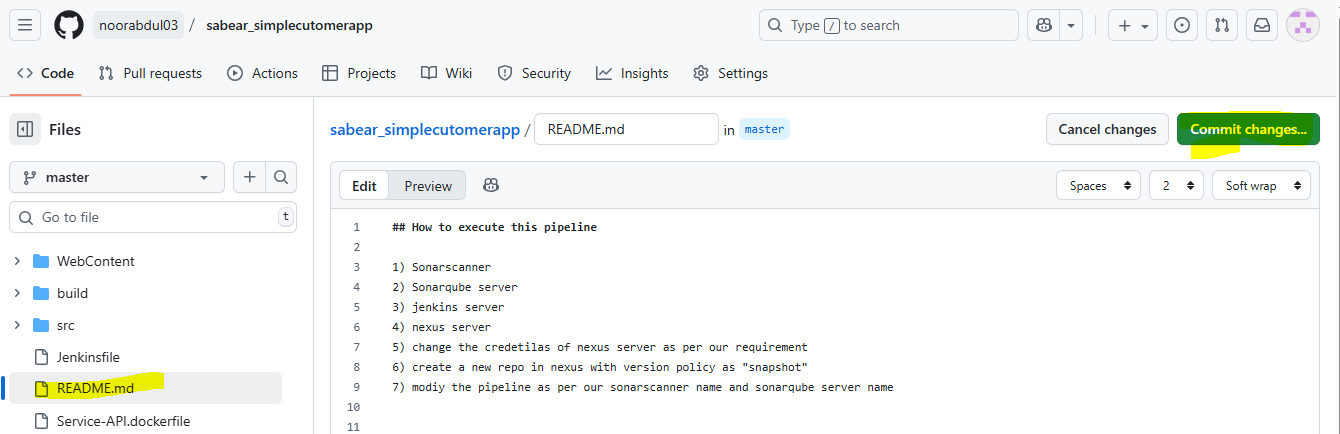


Click save

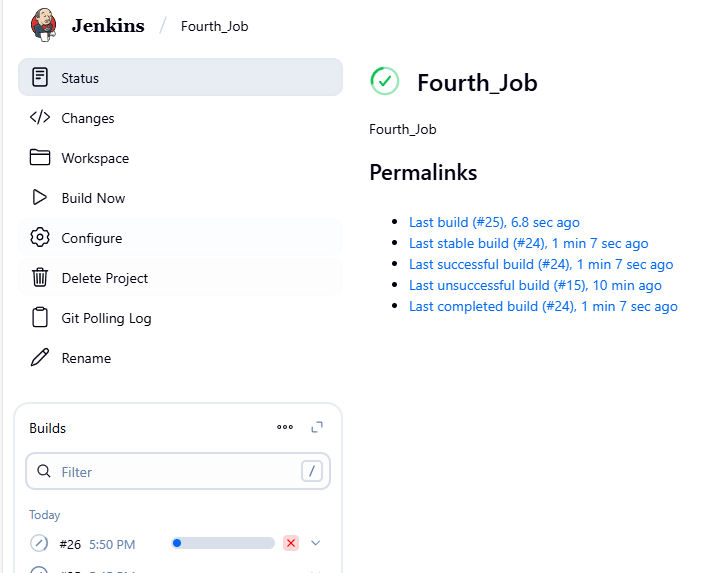
Step 2 : Just make some changes in the code in github repository

Go to red.md file and remove the dot at last or example

And do commit changes

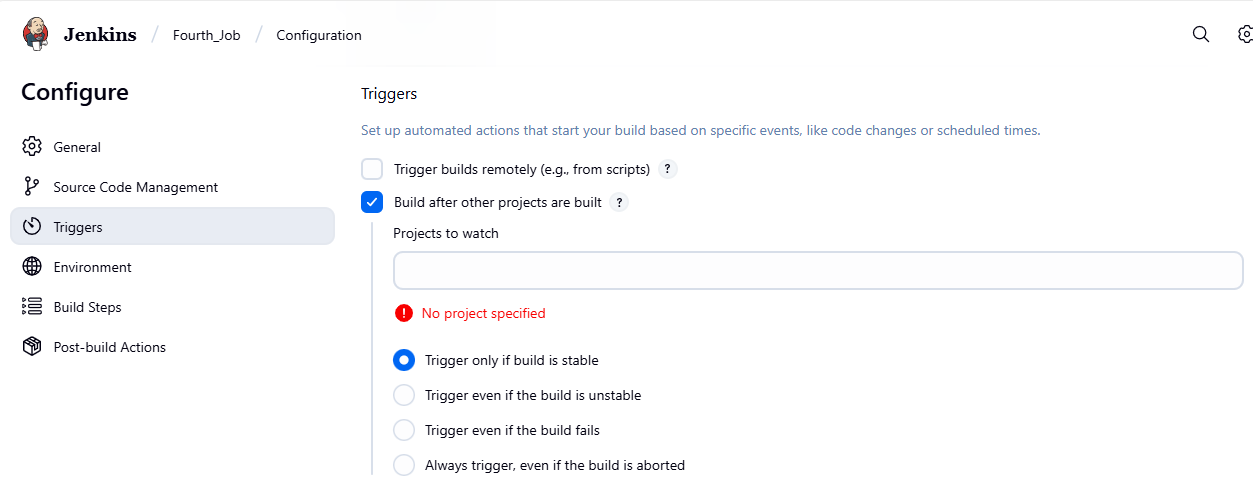


See the below changes, the job is executing I any changes done in the code. If there is no change in the code then never triggered the job in poll scm.

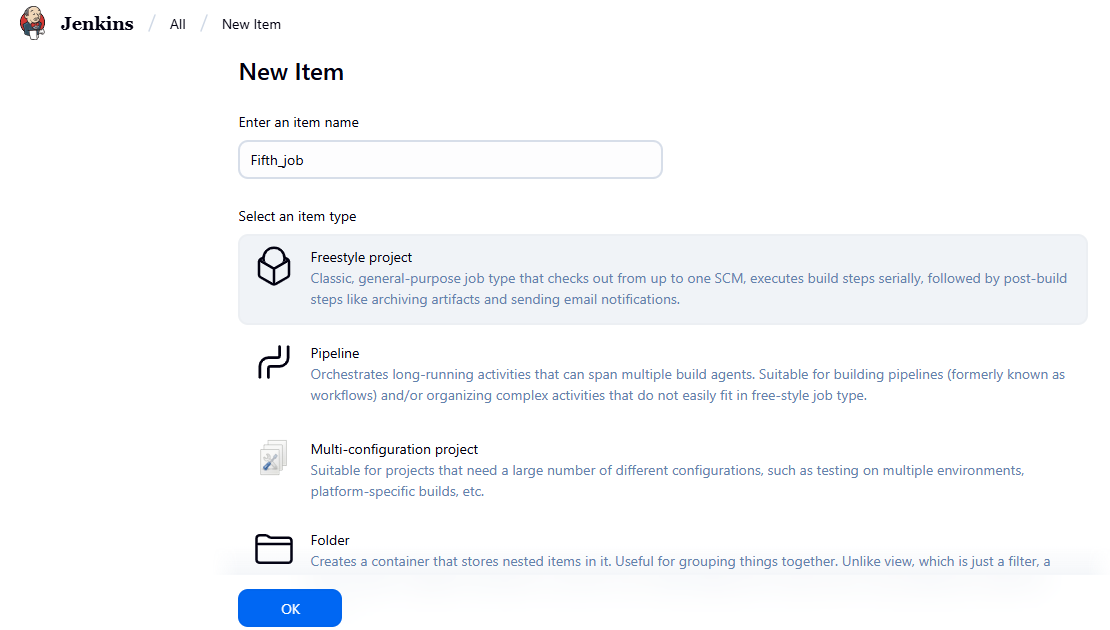
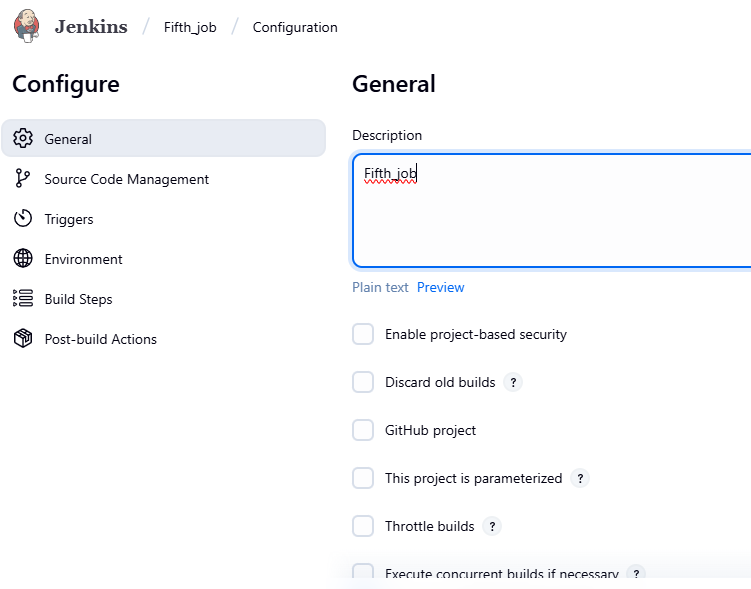
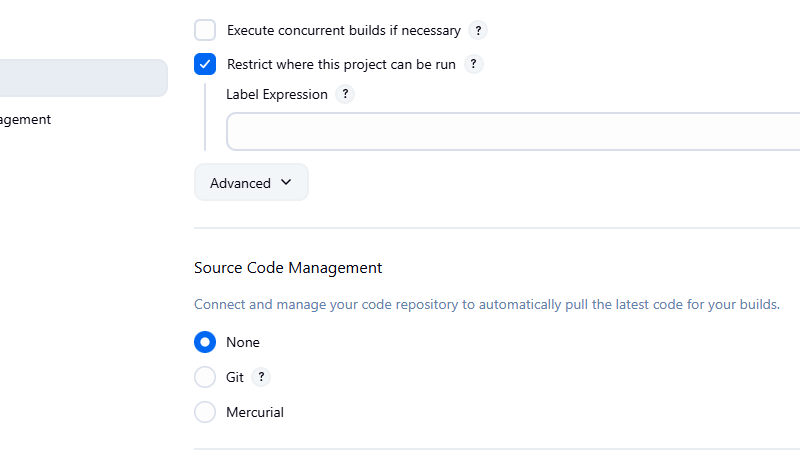
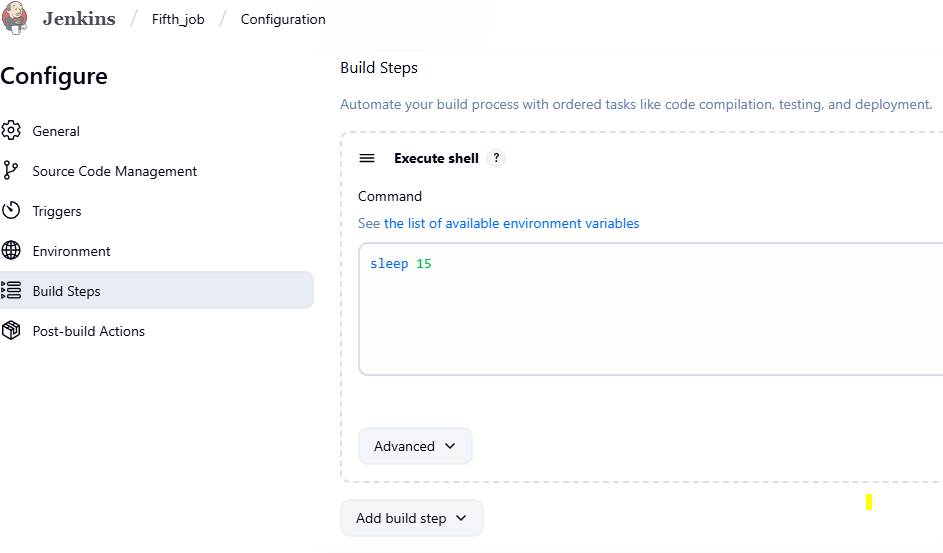


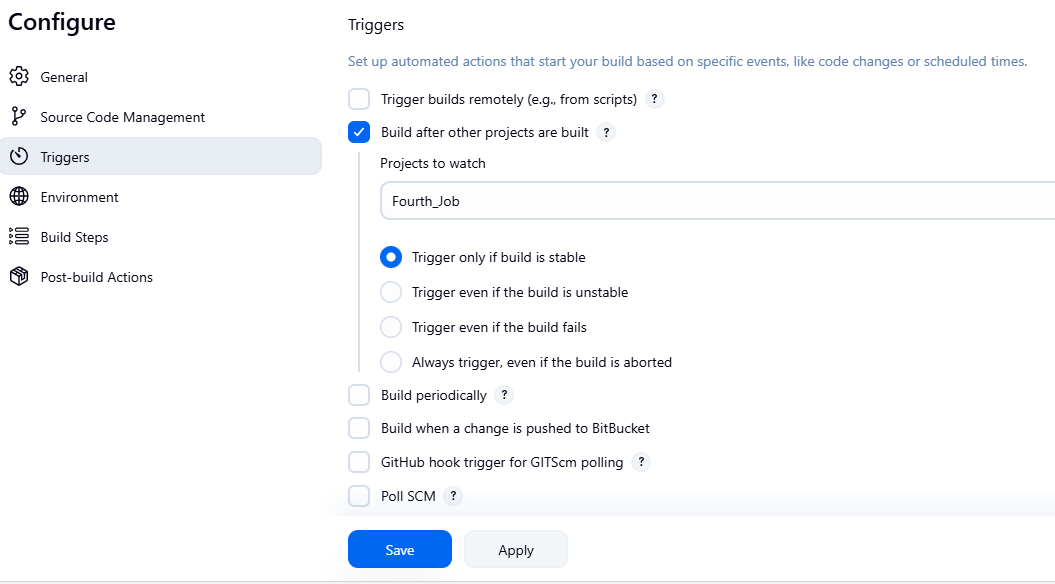
………………………………………………………………………………………………….

Scenario : Build after other projects are built

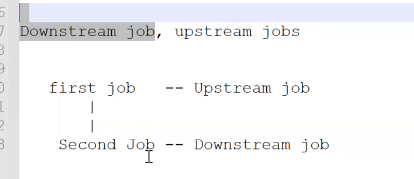


Create a one job



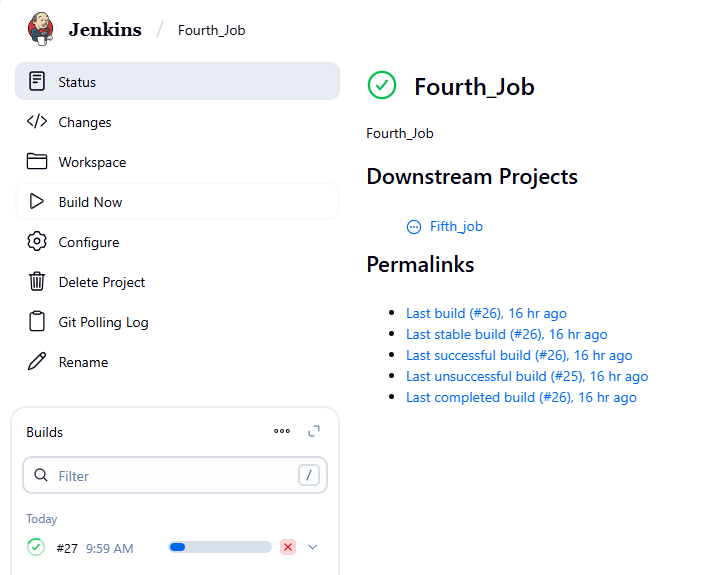
If I enable this option **Build after other projects are built** , Which project need be check (Fifth\_job). So what will happen, so this job will only triggered once the Fourth\_Job is been successfly complted. So there is a dependencies between one job to another job. This type of jobs called as downstream jobs.

Ex: 

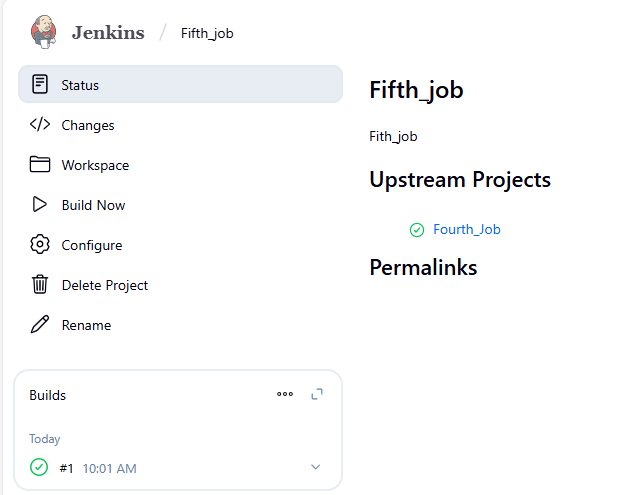
In the above example same also it’s same. The second job will only executed I the first job is successfully executed. If first job didn’t executed well then second job will not be triggered.

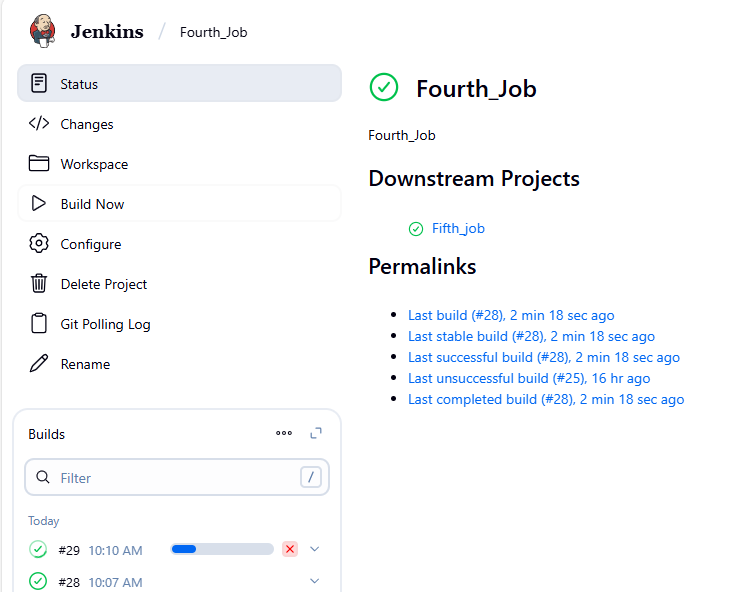
In our case fourth job is – upstream job

Fith job is – downstream job



Automatically fifth job also triggered as below





………………………………………………………………………………………………

NOW TOPIC

MASTER AND SLAVE

Why do we need ? Master and slave architecture

This is my Jenkin server and this will be configured as master and this has dierent different slave machines.

What is the use of having multiple slave machines.?

Master and salve architecture will help us to have high available of our Jenkins server aswell as to of load all the work from the master to the slave machine.

Orelse what will happen is

Ex: this Jenkins server will be used by different teams in my company. Not only in our project, another team member also use the same jnkins server.

So If suppose:

If everyone is trying to run 20 jobs at a time, this is a T2.micro server, we might seen slowness in our Jenkins server. Our Jenkins server might get crash because of the high CPU utilization.

So what will happen our MASTER has been down, then all the projects which were onboarded on the Jenkins server they will not be able to build and deploy the jobs If my server is down.

What we do is we create diferent diferent slave machine.

In each slave machine

for ex: on the Jenkins master server 3 different base jobs are runing like Java, python node bases.

So what we need to do is we need to install Java dependencies, Python Dependencies, and Node dependencies on this master machines.

So there will be lot of dependencies for every application.

What will do is instead of installing everything in one machine, I will create different different slave machines.

Ex: Slave-01 for java application

Slave-02 for python application

All the dependencies related python will be deployed or installed in slave 02. Same java also.

So whenever a java job is getting triggered then master will send this job to slave-01 same for python.

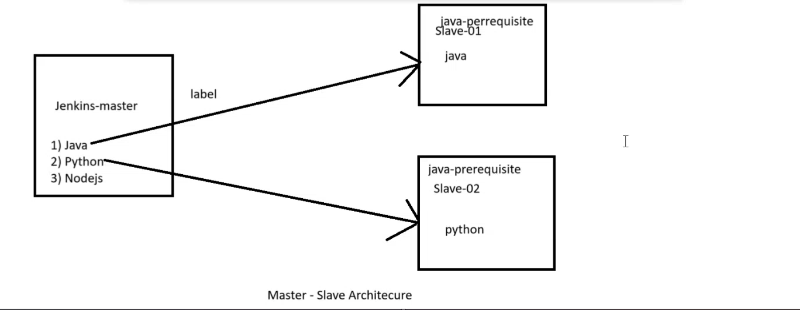
But how it is actually sending, who is going to indicate master that slave-01 is related to java an slave-02 is related python, that depends upon labels.

What labels do,

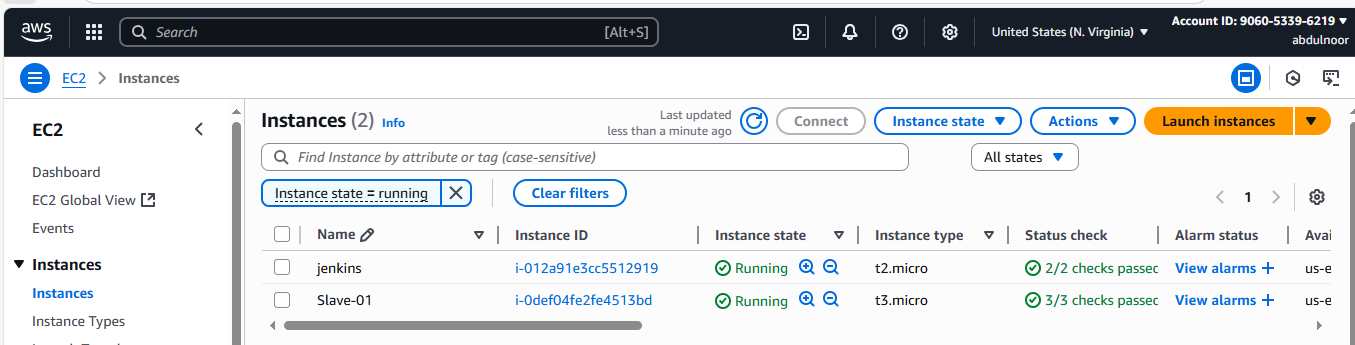
In my job I will configure that this java base should be executed on slave-01 based upon the label name it will send my job to the Slave-01

So our master machine is only used interms of offloading work to the slave machine.

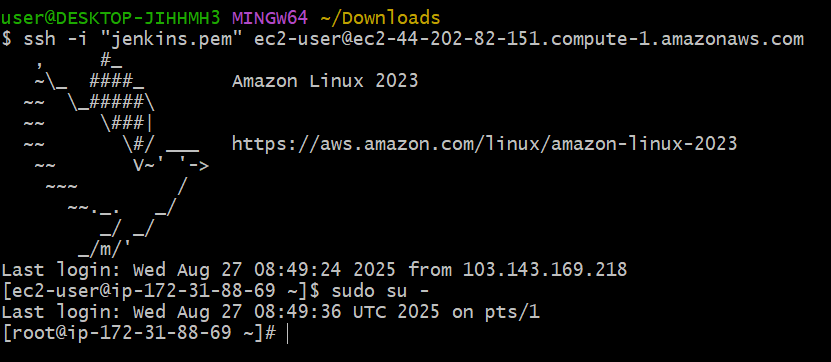
Means master commanding that, execute this job tp slave machine.



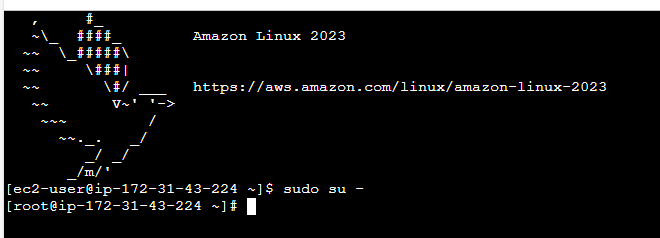
CREATE EC2 – For Slave Machine alreay we have one Jenkins-Master



The below is my Jenkins-Master



The below is my slave machine and only ependencies is our slave machine is Java. Means no need to install Jenkins. Just need to install java, because this is the dependency for the slave machine configuration.

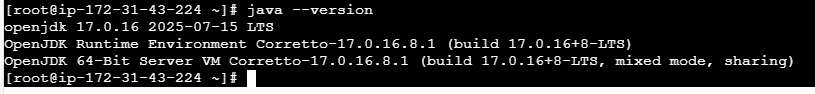


Install Java command is

**sudo dnf install -y java-17-amazon-corretto**



Java --version



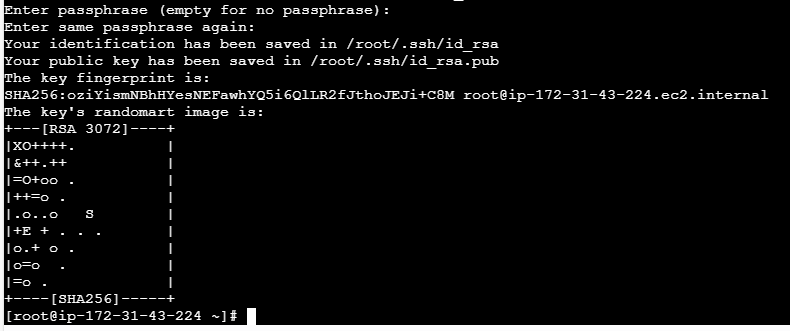
Now I will execute ssh-keygen command

It will generate one ssh keygen, basically we need to make sure of passwordless authentication. That’s the reason I am creating one ssh keygen.

ssh-keygen



Enter enter enter

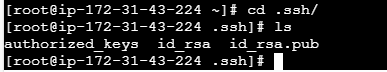


Ssh keygen has created then,

I am trying to copy this particular file

**cd .ssh/**

**ls**



**I am trying to this copy id\_rsa.pub in my authorised key**

**Enter the below command**

**cat id\_rsa.pub > authorized\_keys**

**enter**

**the content of this particular file has been copied in authorized\_keys**

**and change the permission**

**chmod 700 authorized\_keys**



**We have successfully created keygen and added the permission as well.**

……………………………………..

Once done we need to go to our master machine.

1. In Mater machine I need to create one directory called .ssh inside /var/lib/jenkins

mkdir -p /var/lib/jenkins/.ssh

or

mkdir .ssh



And change the directory

cd .ssh/

then execute the below command

ssh-keyscan -H SLAVE-NODE-IP-OR-HOSTNAME >>/var/lib/jenkins/.ssh/known\_hosts

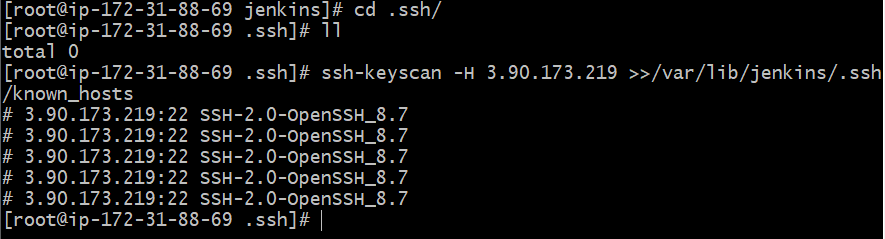
keyscan command what it will do ?

it wil try to copy key the key from your slave machine and store in this particular location

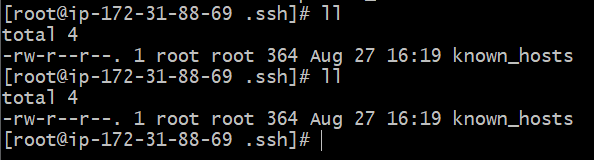
then I am trying to copy this particular ile into this particular location.

ssh-keyscan -H 3.90.173.219 >>/var/lib/jenkins/.ssh/known\_hosts

I should execute the above command after change in the master machine

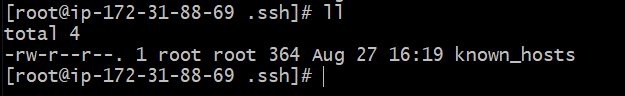


If I do ll here you can see one file has been copied for known\_hosts in the particular location var/lib/Jenkins



So keyscan command will copy the key when it will copy the particular(slaveIP) node machine for any keys and then copy in this pertuclar location. /var/lib/jenkins/.ssh/known\_hosts

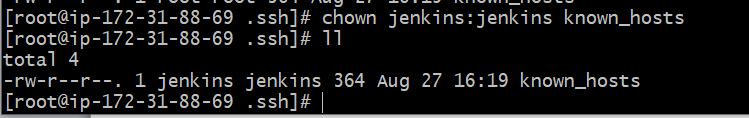
1. Change the ownership to Jenkins user because all the jobs which are executed by Jenkins will be specific to Jenkins user only.



If you see above is root user, so I want to change this to Jenkins user for this enter command

chown jenkins:jenkins known\_hosts

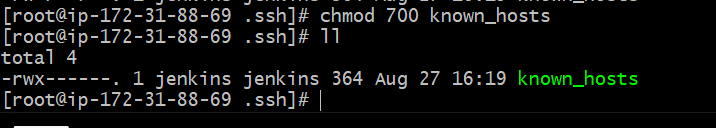
ll



Or else Jenkins will not be able read this particular file because of permission issue

And then give permission to this file

chmod 700 known\_hosts



………………………………………………………

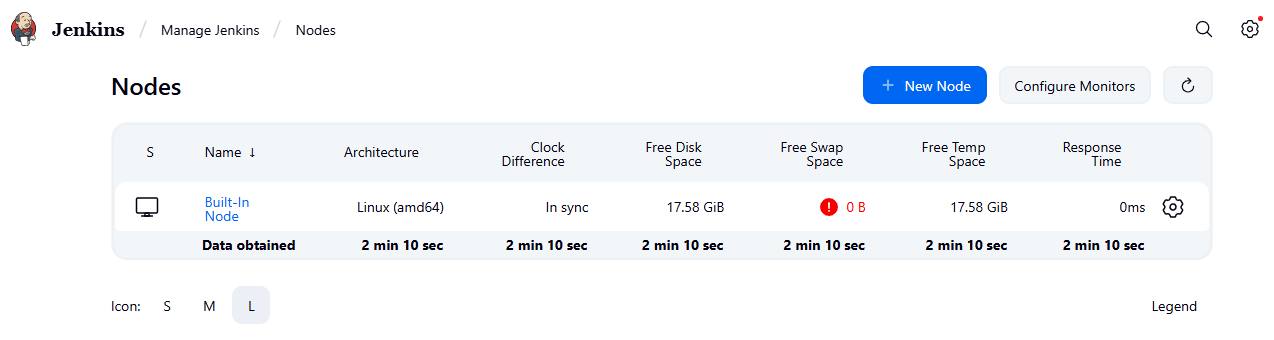
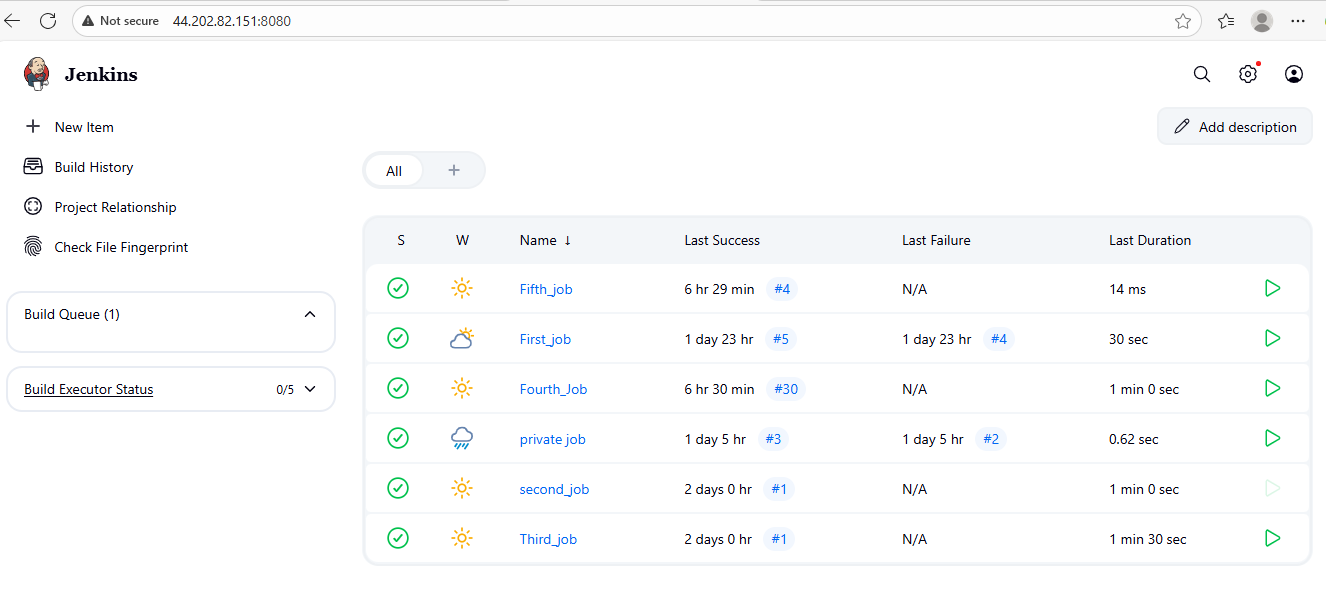
3.Now we need to goto jenkins GUI

Goto GUI – go to manage Jenkins – go to Nodes

Q: What is Nodees?

Where we add, remove control and monitor the various nodes that Jenkins runs jobs on.

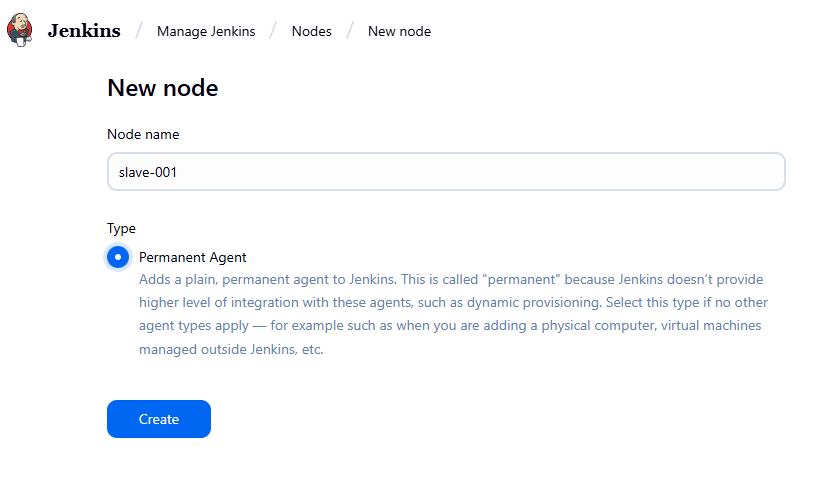
Understanding: according the Nodes Jenkins runs jobs



This built In Node means our Jenkins master machine, means wherever we are trying to run this

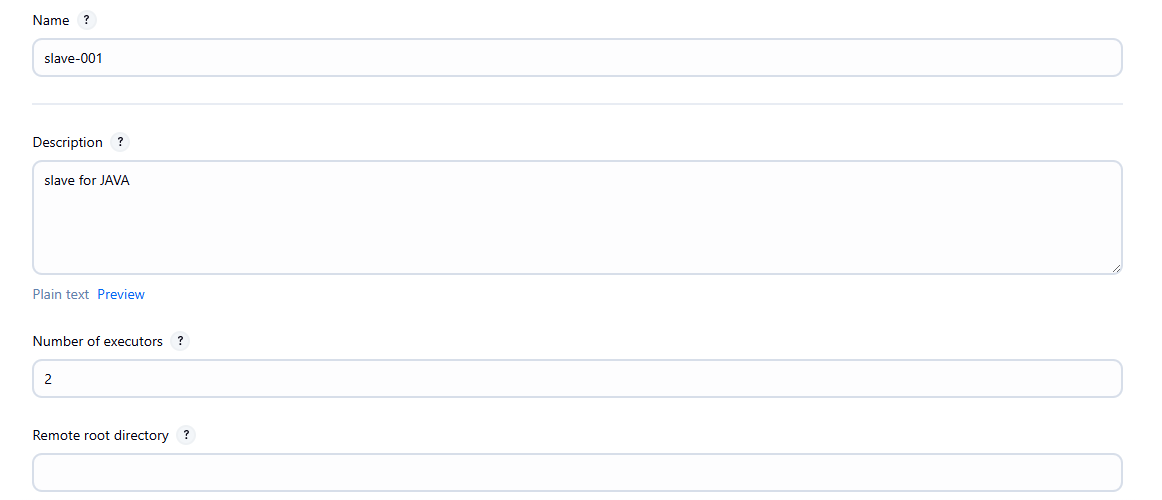
Then Click on +New Node

And give node name ex: slave-001



Click on create

Remote directory is mandatory



Description is optional.

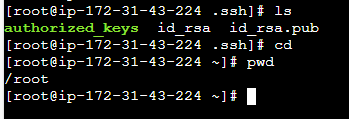
Number of executors: How many jobs you want to run on this particular machine. In my case it’s 2.

Remote root directory:

On the slave machine where should all the jobs related data or information should be stored

In our case We haven’t created any directory





I am just creating one directory in root path as shown above.

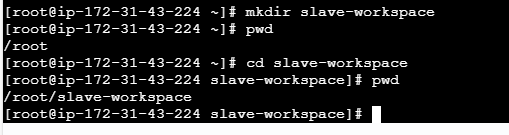
Create one directory called

mkdir slave-workspace

then enter into that directory

cd slave-workspace/

pwd



And I want all the jobs related datails should store here /root/slave-workspace

Label: label is nothing but our naming convention.means out of this name you want to configure. So this job will only execute. I mean this slave machine will only execute jobs with label called Java

Usage: Use this Node as much as possible: Means, If this node is free then Jenkins master will send some other jobs as well or else only build jobs with label expression matching this nodes means, only the jobs with this particular label(Java) should only get executed.

Lanuch Menthod: we have configured ssh, so I will be using launch agents via ssh option.

Host (Machine): means, your IP ADDRESS related to your worker (slave) machine.

Copy the slave machine public IP 3.90.173.219  and keep in host.

Credentials: DO we have any credential configure : No

Yes we do have credentials to login to the server but I havn’t configure anything means, first – sever, This is the sir pemkey of slave-001 server. In my case Jenkins is my pemkey name of my slave server.

Inorder to connect this slave-001 server we were using some authentications pemkey (Jenkins)

Now I should copy this pemkey in this credentials.

Click on Add credentials

Scope: Global: Means, it will be available to Jenkins, all the slave machines and different options/to everyone.

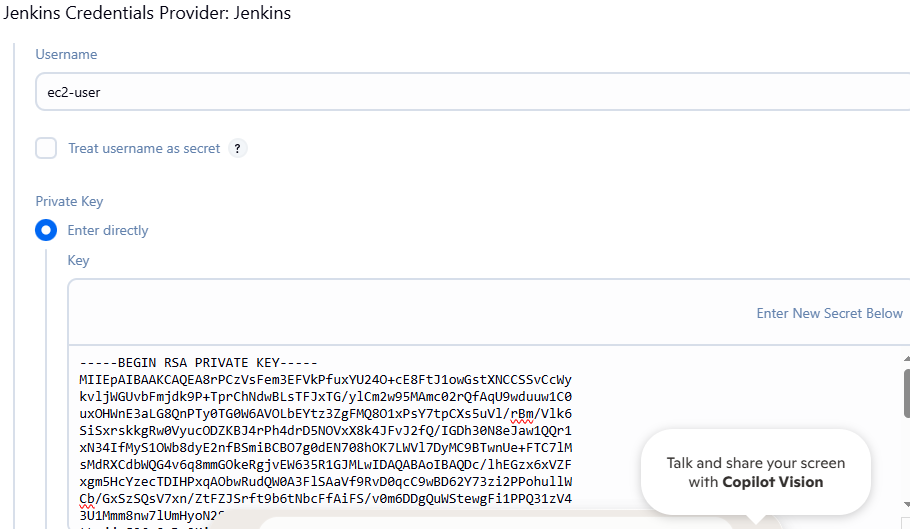
ID: slave-001\_key

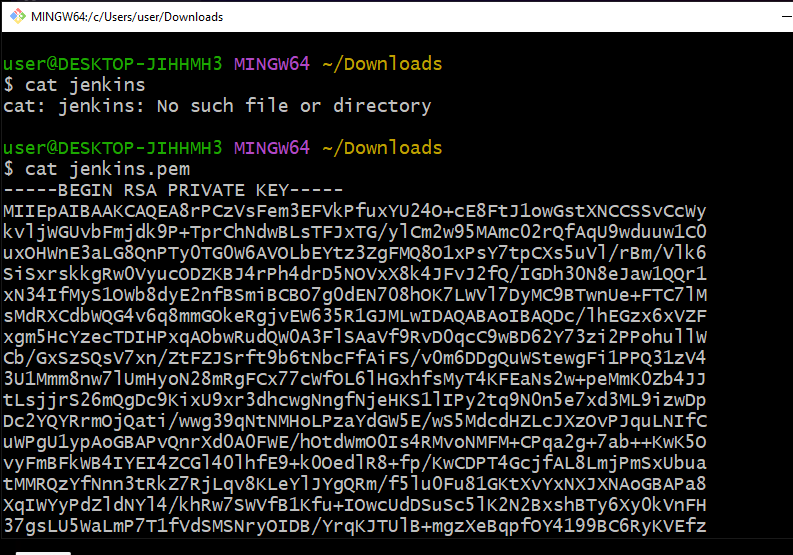
Username: ec2-user

Select enter direct key

And add pemkey by copying from download to here paste.







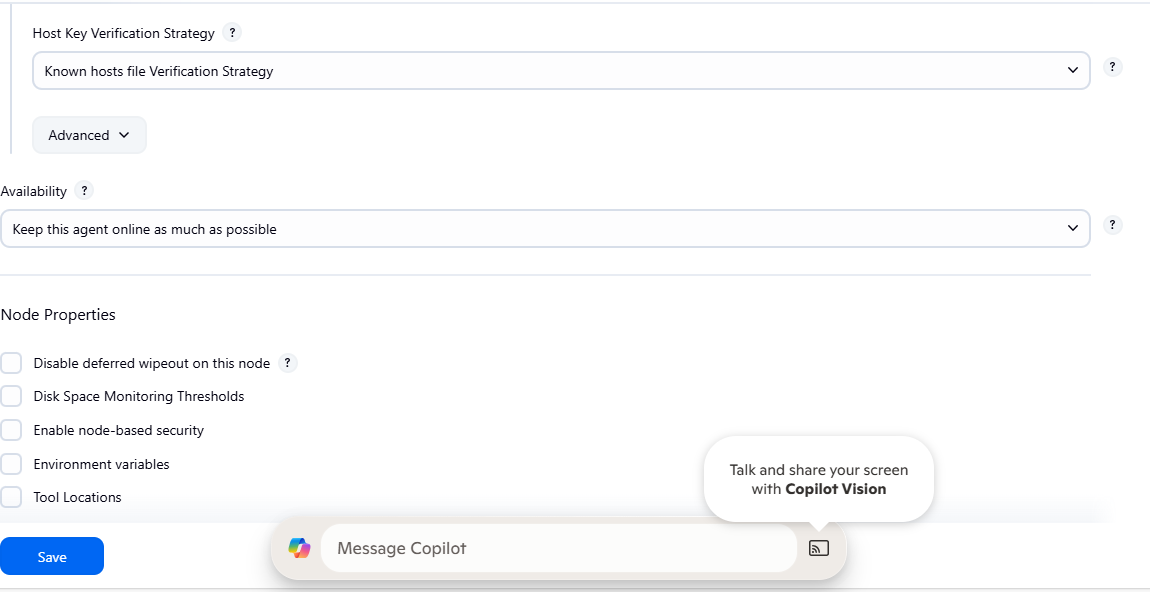
Click on ADD

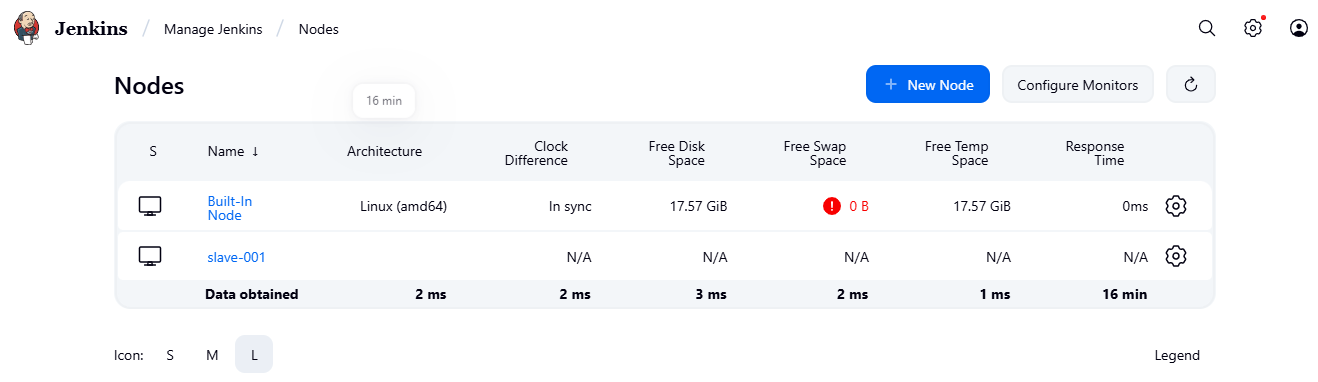
Host key Verification Strategy:known host file verification strategy: Means, we have copied one key right so what it doing while trying to connect to the known machine it will go to this perticular path location /var/lib/jenkins/.ssh and it will check or the file called known\_hosts

If this is available then only our authentication will be possible/validated

Click on SAVE

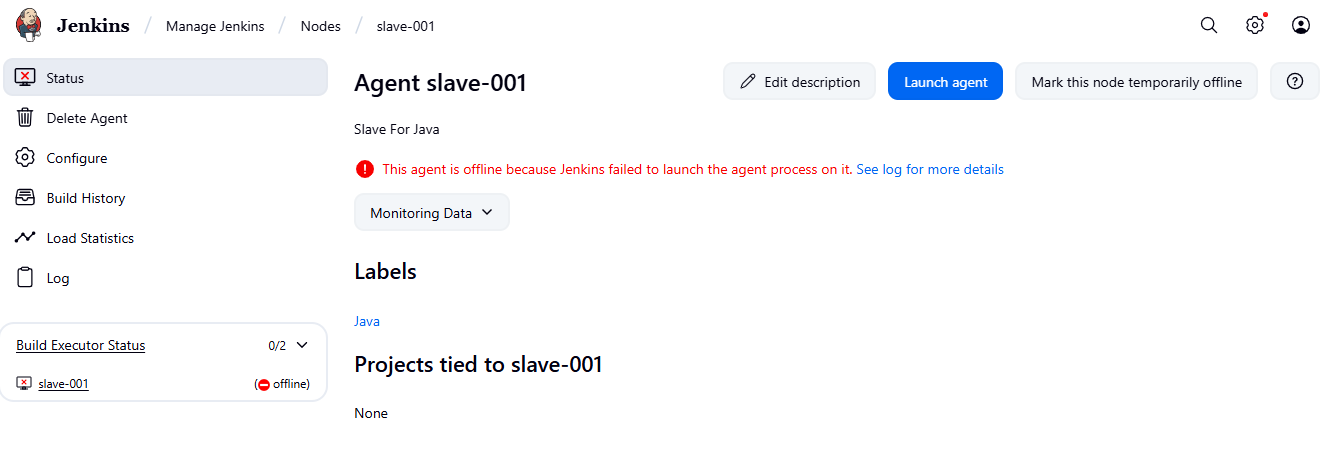
 in credential you can your created authentication key also.

click on save



1. We have done all configuration, let’s check the logs related to slave-001 node.

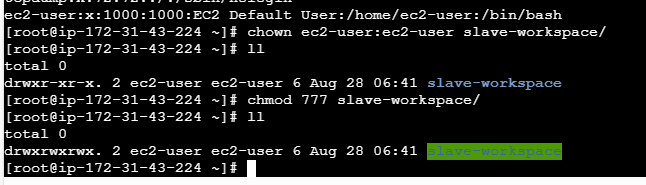
Click on salve-001 node



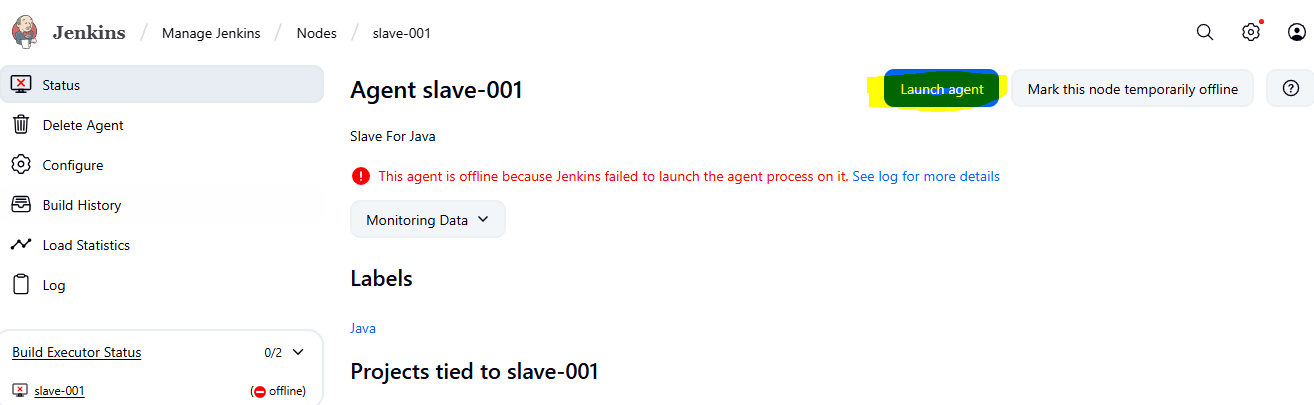
As per red log something went wrong.

Click on see log or more details

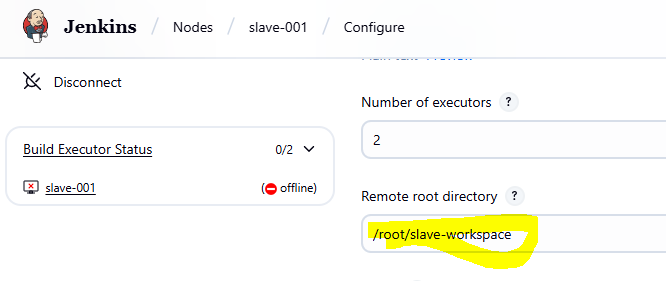
1. First check which user is available
2. Change the ownership
3. Then give permission

Again go to salve-001 and click on launch agent

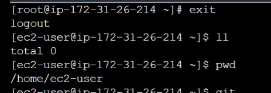




For this I just configuration of slave-001 and change the remote root directory

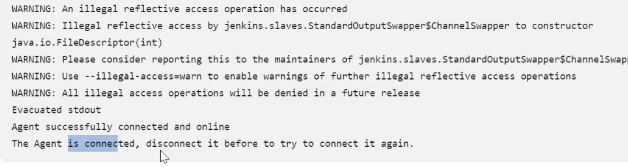
And change to /home/ec2-user/





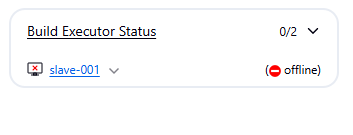
Because we gave credentials ec2-user

Relaunch agent



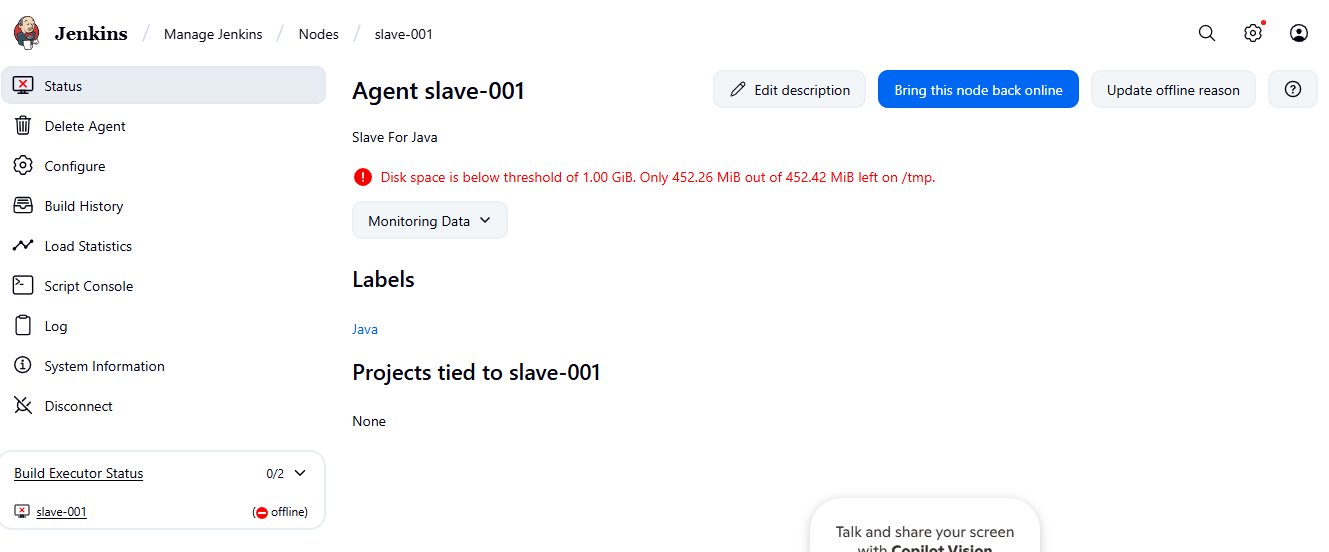
Then saved the error solved.

But still it’s showing offline



Error2:

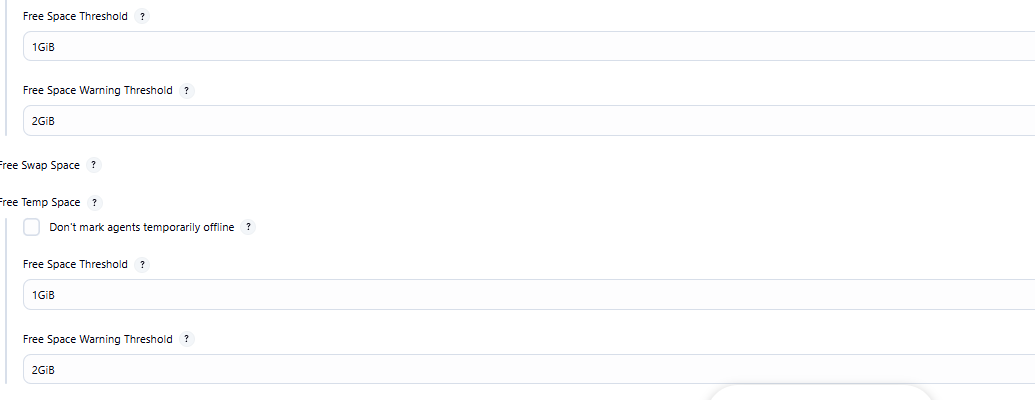
But getting a new error as below



To resolve this goto

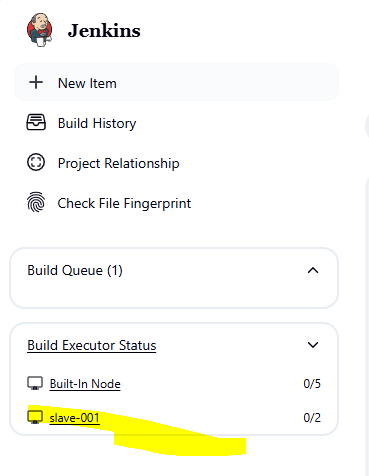
Jenkins – manage Jenkins – node – click on configure monitor

And change this space threshold to 100Mb



Change it to 100Mb in all

Now error solved and as below slave-001 nove is connected as below



This is how we need to conigure Master and slave architecture.we can add diferent different slave machine.

Now we have configured label as Java

1. Now I execute the job



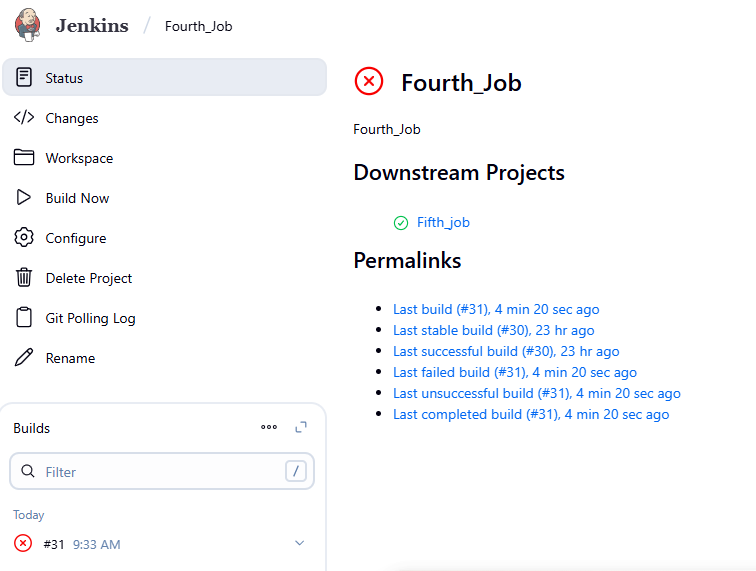
Click on save

Then once you click on build now then this job will executed on the particular slave-001 machine only.

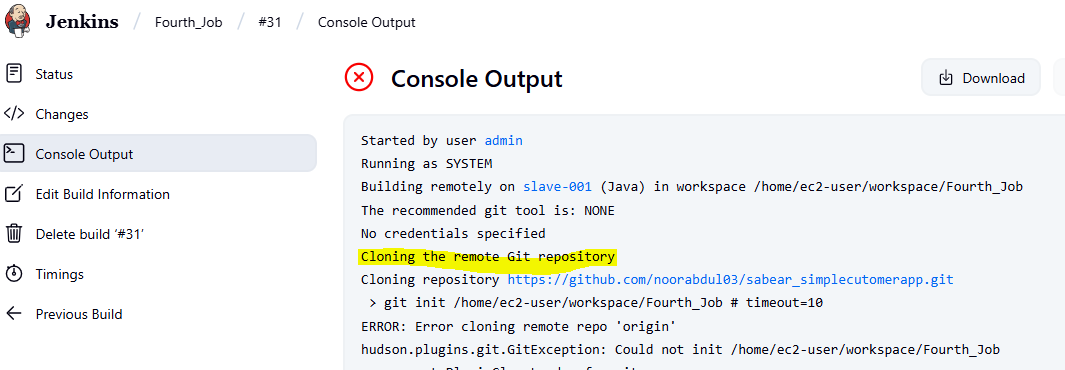
Click on Build Now

And go to Jenkins

But here job has failed

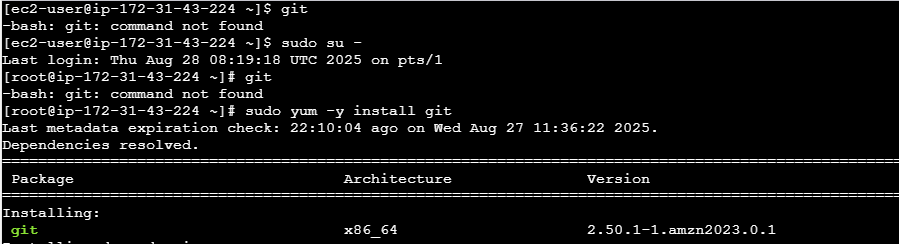


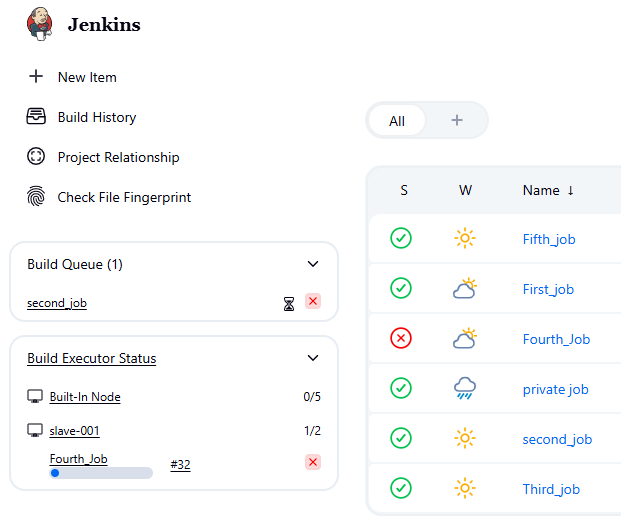
To resolve this click red mark means console and check the error



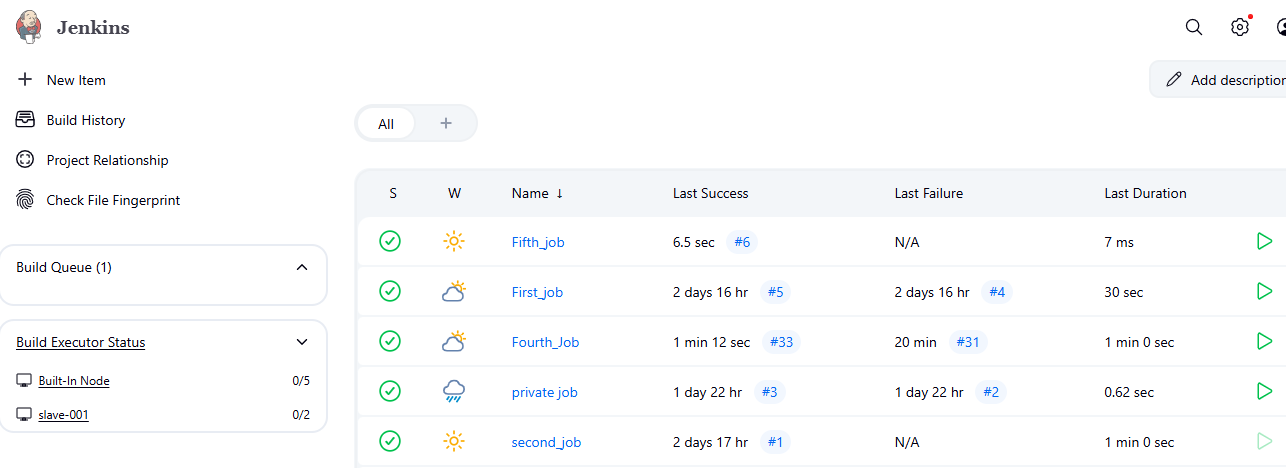
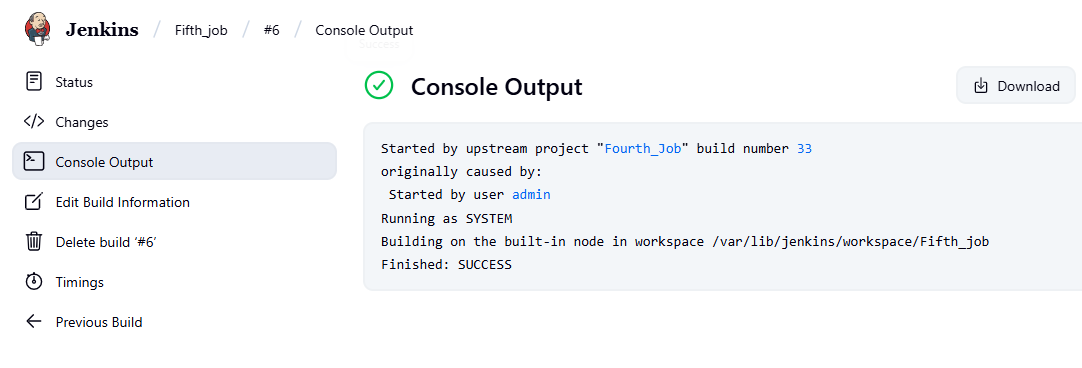
Why because git is not installed in slave-001 and we attached git repository in the job.

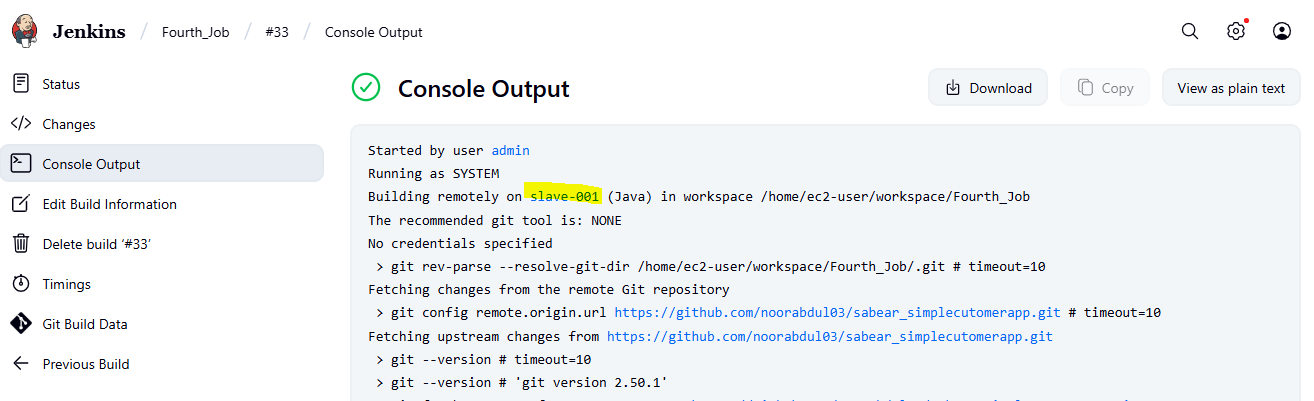
So install the git in your slave-001 machine





Then automatically fifth job will be executed once the after fourth job executed because it’s a downstream job o upstream job.

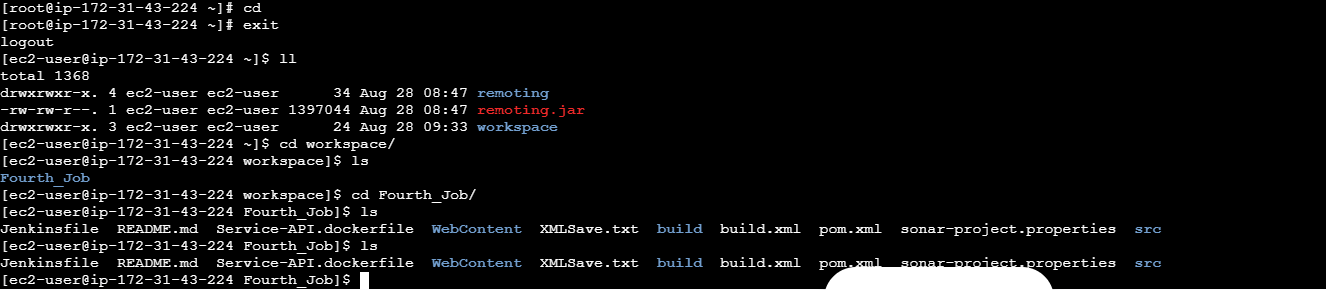
Fifthe job runs on the fourth job after execution. 

And the fifth job is running on the slave machine

………………………………………………………………………………………………………..

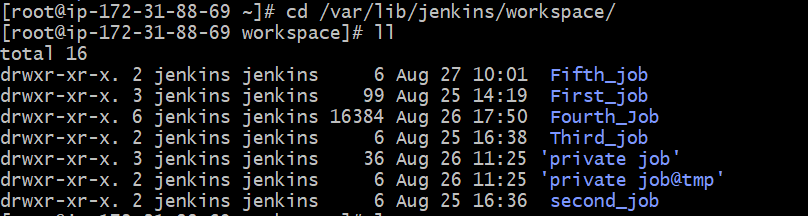
Now all the data related to first(fourth) job stored in slave-001

Then you can find the job which is under slave-001 and all the files related to your job



Go to master ec2 machine

And you can find the files as below



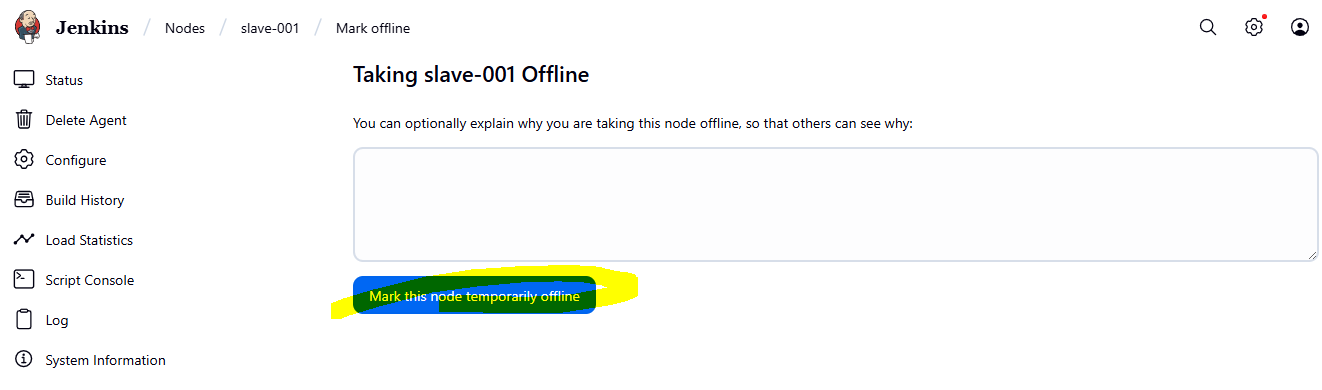
……………………………………………………………………………………………………………………………….

New topic

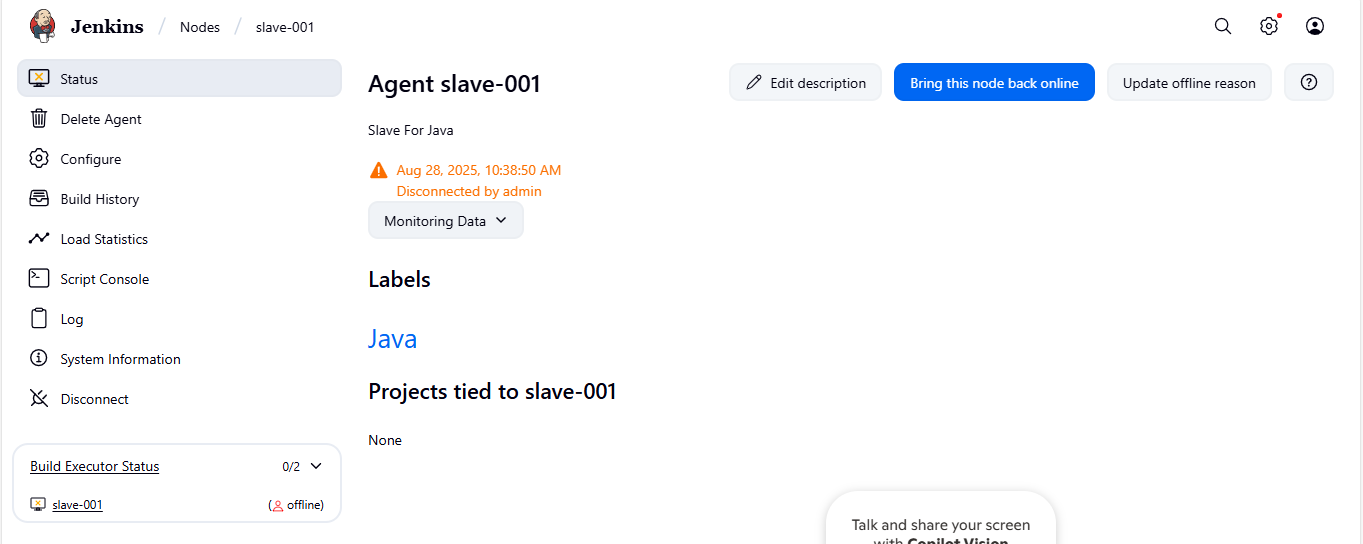
Example: If this slave-001 machine is down mark this node temporarily offline



Click on that

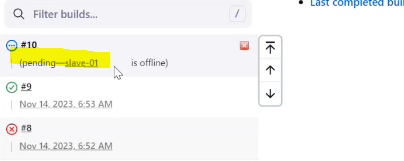


Click on that



See here it’s offline

Suppose I you run now any job then you find that job in queue only and waiting waiting until and unless it will ind any slave with the label call ed Java



……………………………………………………………..

Use as much as possible option: if you keep in the node slave-001, then whatever job are not restricted labels (Java) then the other jobs also run in this machine I no jobs are not in queue.

Means not only Java.

First and second job which are java and python it will run slave-001 only.

……………………………….

Okay now we have tested. So we terminated slave-001 ec2 instance and deleted agent slave-001

……………………………………………………………………………………………………………………

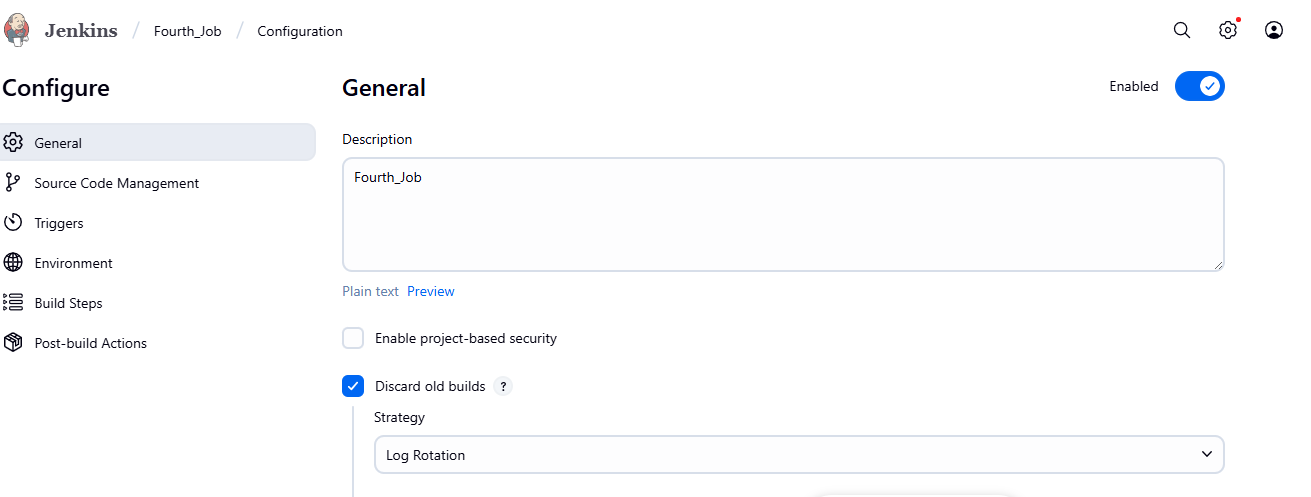
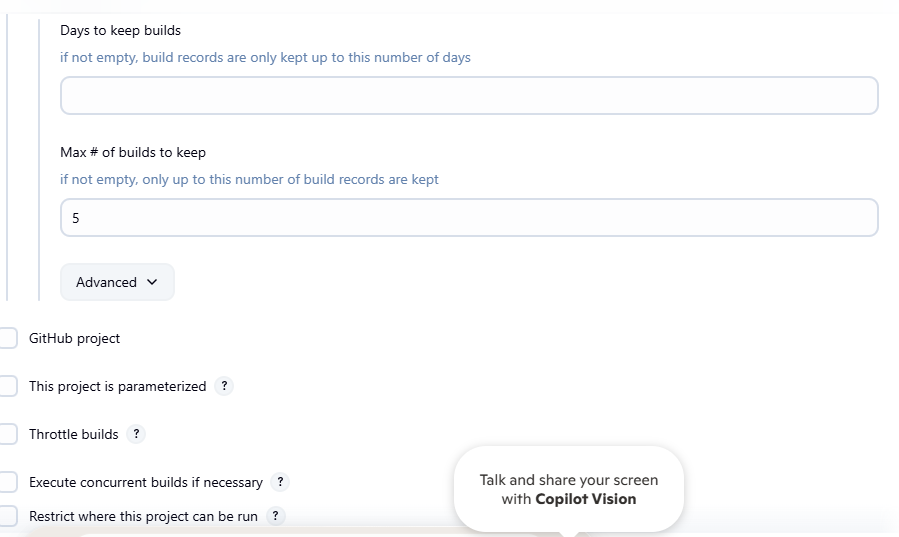
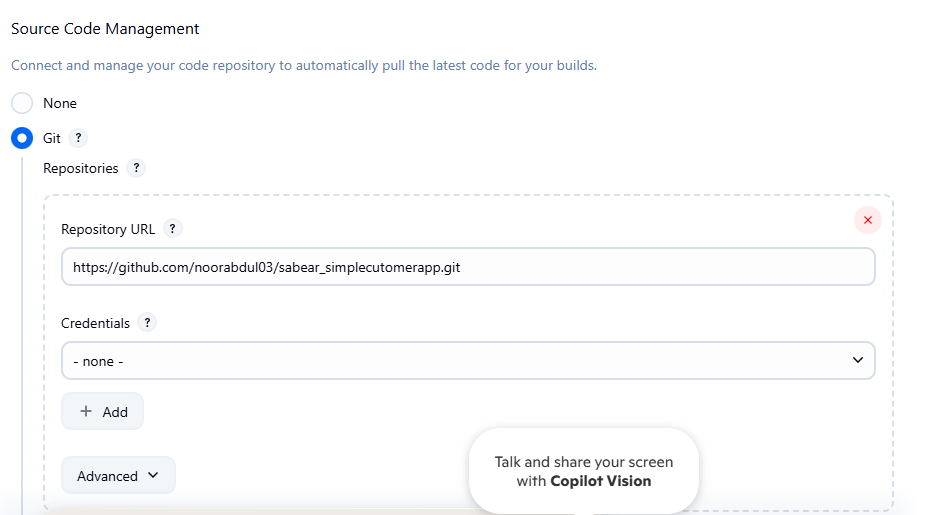
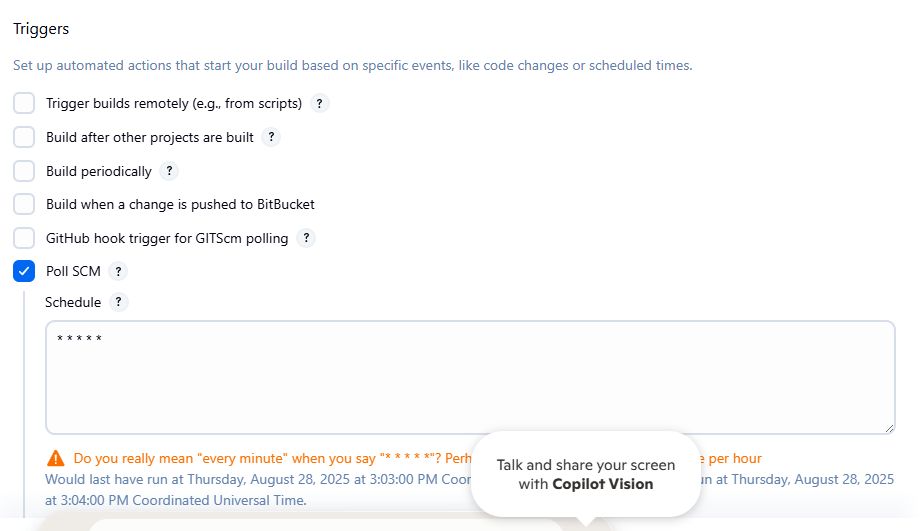
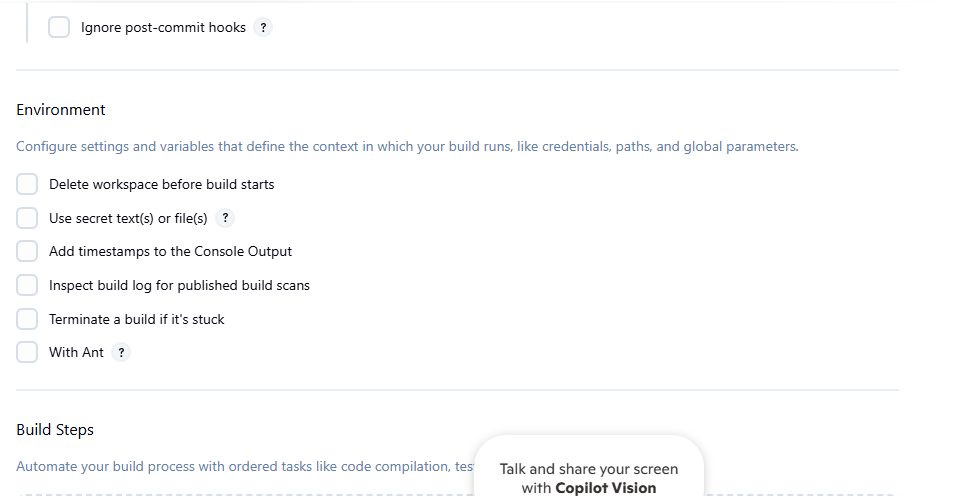
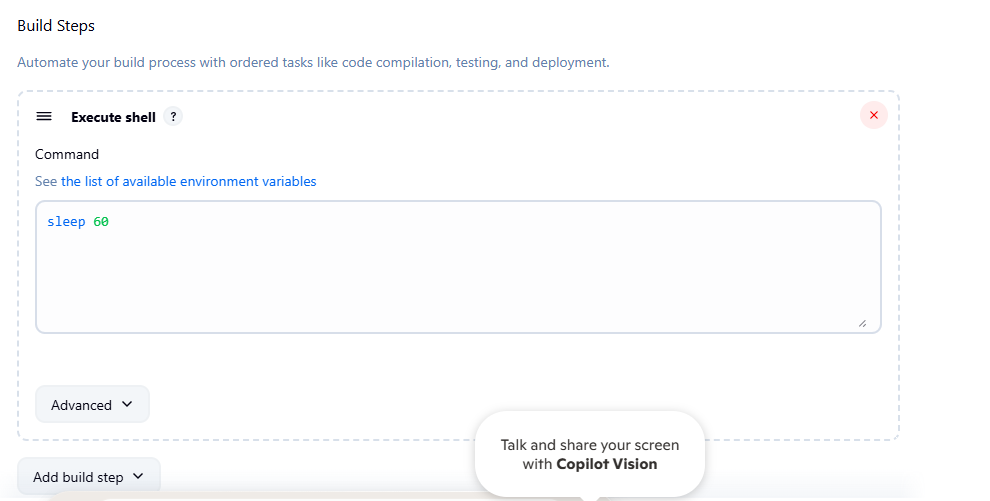
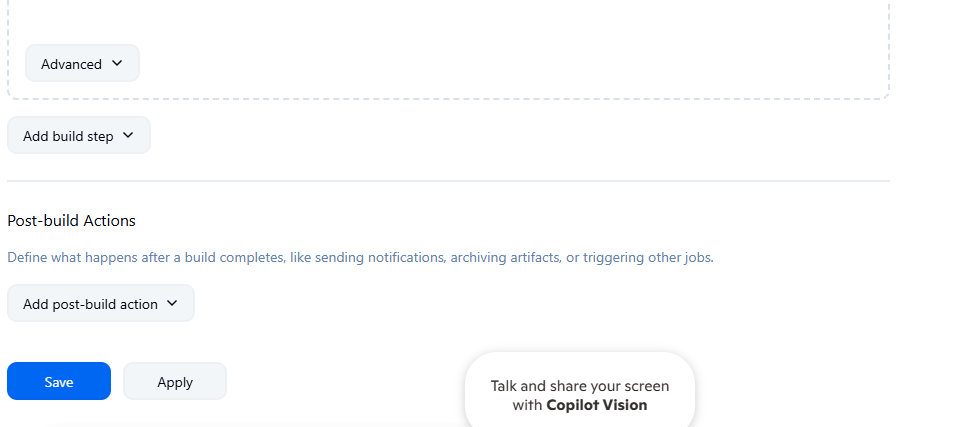
Other Topic:

Number of Build Jobs executed takes more consume space, means it takes storage and we should maintain the storage also:

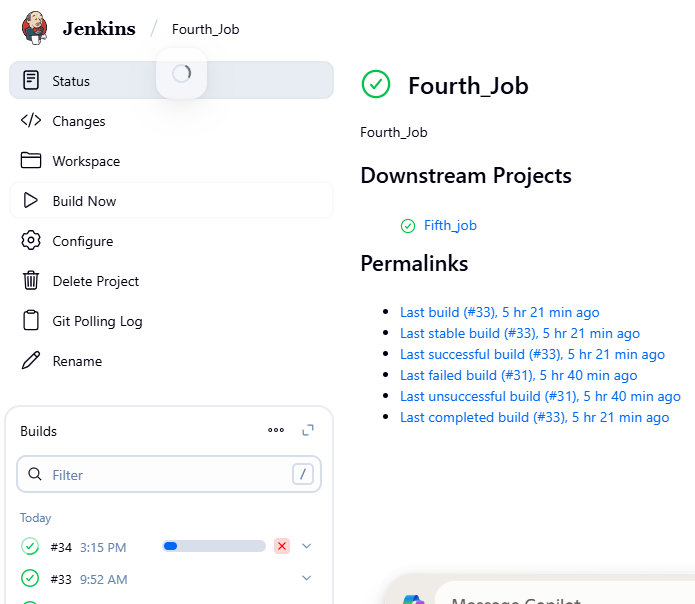
We should always maintain the log rotation

Open the job – configure – remove restrict where this project can run – select Discard old builds

Long rotation: log is nothing but your history

Now click on Build now



I have executed the build number 34 with the above job setting and it has removed all the jobs except last 5 jobs as I kept same in the job configure.remaining logs are cleanedup.

…………………………………………………………………………………………………………………..

New Topic

We have only 1 machine now that is master Jenkins.

What I something goes wrong and this master has been deleted or not available and my production pipelines are running on this master machine.

If master is down or not available then what will happen and there is no slave machines, we have only master, then

Our applciations are not able to release as well as lot of problems or lot of impact on the business as well. So

We should always take backup of Jenkins server which is having all the jobs.so the Jenkins server need to take backup in diferent location or diferent server.

At least we have a backup and we can create one more setup, we can replicate our Jenkins.

How can we take this backup and what are the dierent things you will be taing backup.

cd /var/lib/Jenkins

ll

in this Jenkins directory we have all the things related to my current Jenkins server, means we have all the job related details,plugins,node machines details,workspace, scerets, everthing is available.

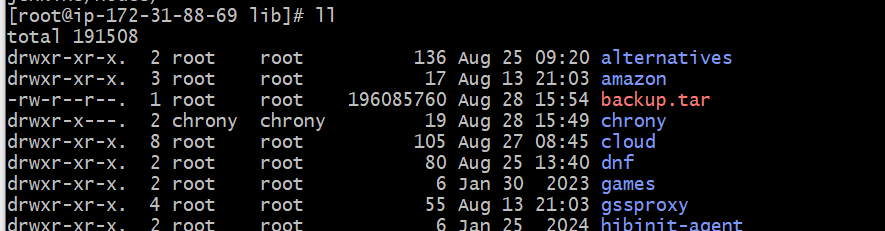
If I copy this directory by using any bash script or plugins then will have the complete backup of our Jenkins

Step1:

tar -cvf backup.tar jenkins/

we are trying to take a copy backup of all our directorys which are available inside our Jenkins.

Step2: ll



Now with the help of Jenkins backup what I will do

Step 3: Now I can scp this backup ile to any other location.means on any other server.or any other central repository.

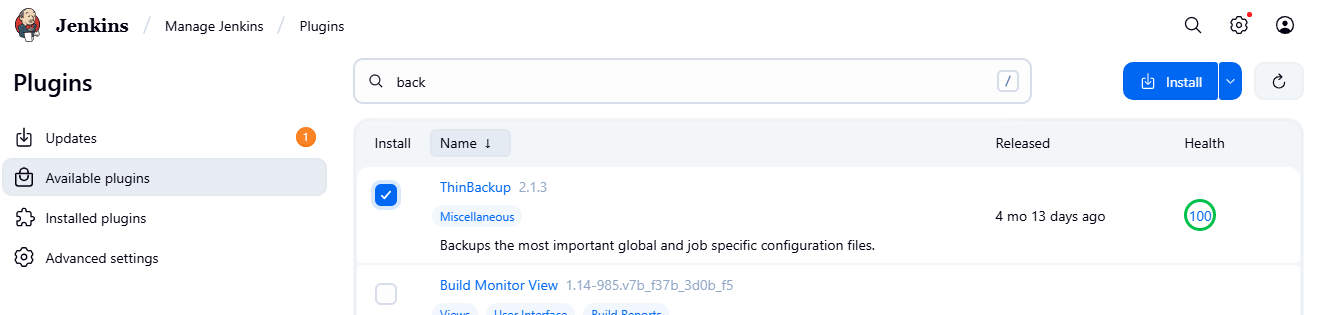
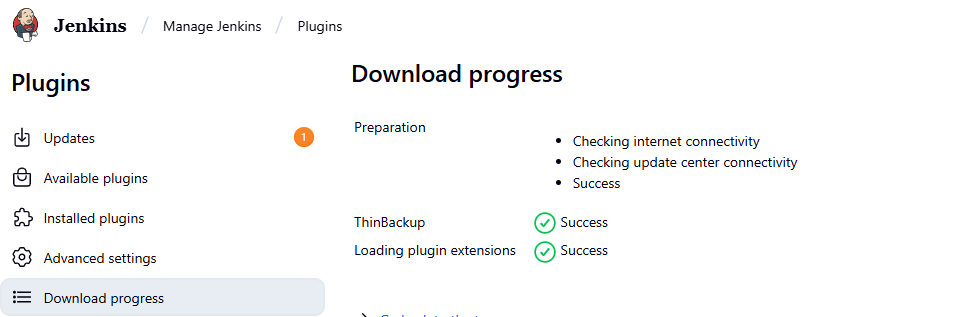
We should have a mechanism which will automatically take backup of our Jenkins for every 3 days or 4 days or every 6 days.exmaple.

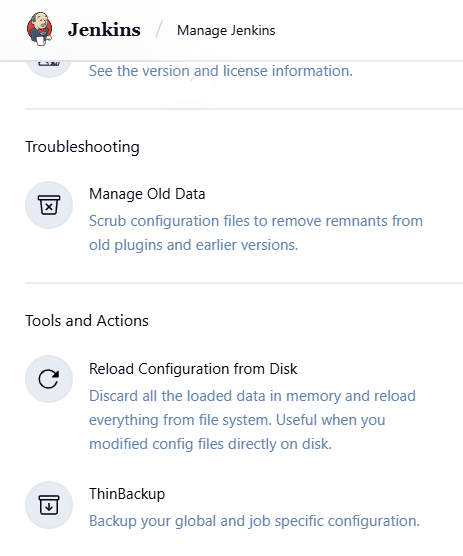
And it should transfer this backup file to any location xyz. When something goes wrong then we can use this directory.

So or this we need to implement Bash Scripting, that bash scripting automatically triggered based upon cron jobs. It is one way

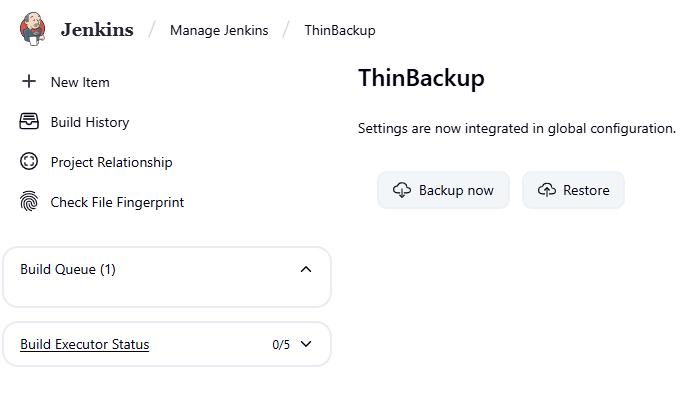
Another way is we have the plugin called Rethin plugin.

thin backup plugin: go to Jenkins – manage Jenkins – plugins – available plugins



Click on backup now



Sabair didn’t more about thin backup.

…………………………………………………………………………



This is my tar file

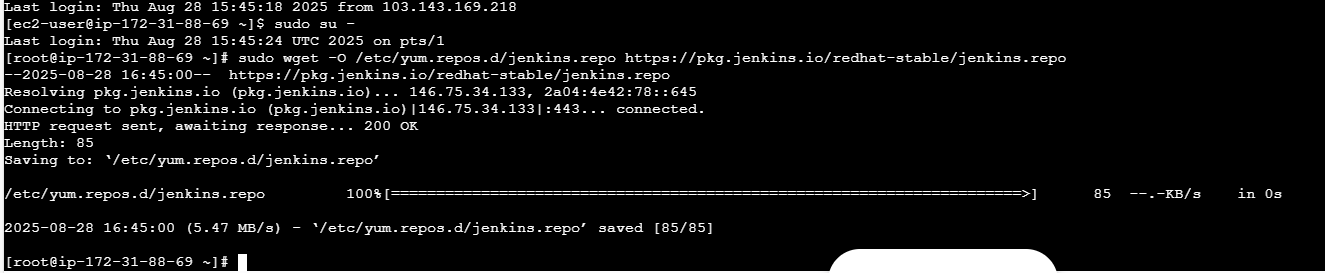
I can copy this perticluar tar file to another server. Means If my Jenkins server is down

Step 1:create one instance

Ex: assume that our Jenkins server is down and we are unable to login our Jenkins server.

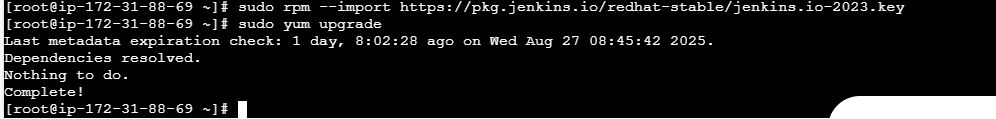


Connect to server



sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

sudo yum upgrade



sudo yum install java-17-amazon-corretto

yum install Jenkins

systemctl start jenkins

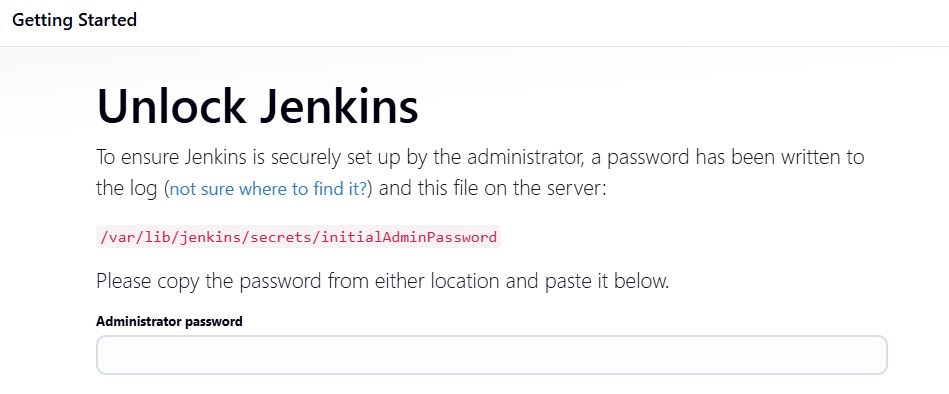
systemctl status Jenkins

so now we have installed java and Jenkins in our backup-jenkins server to show that nothing is there in our backup server

copy the public IP number of our backup-server ec2 and paste in browser along with 8080

54.145.202.170:8080

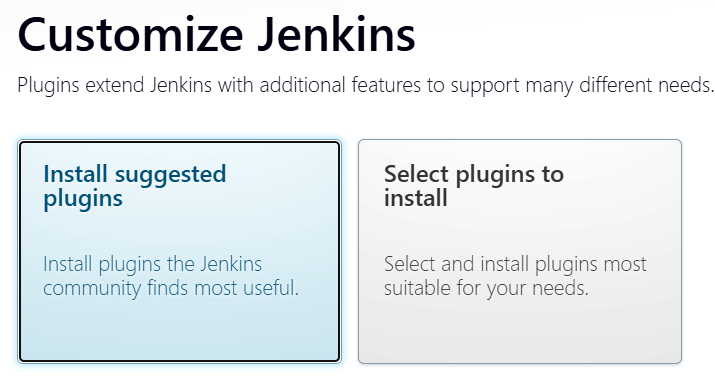
So my Jenkins backup sever has started and it is asking for credentials

cat /var/lib/jenkins/secrets/initialAdminPassword

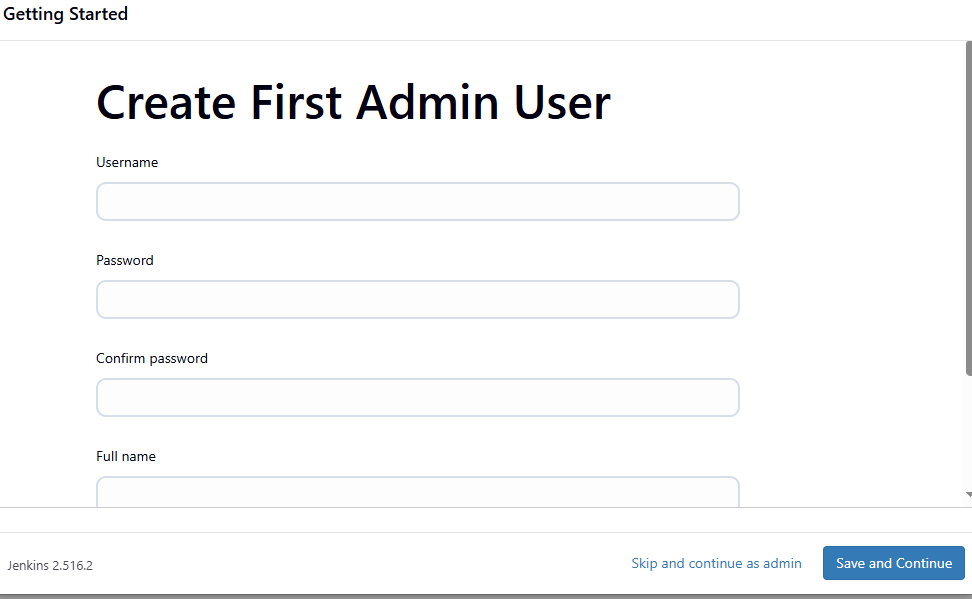
copy it and paste above ap field

Click continue

I don’t want to install any plugins because everything will backup.



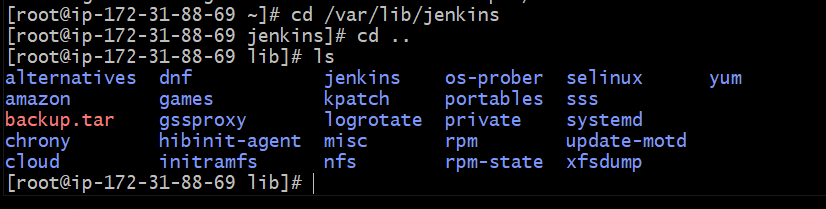
But here select install suggested plugins othervise it will take lot o time



Click on skip and continue as admin

Click on save and finish in next screen

Click on start using Jenkins



Step

4 task:

**Take backup of Jenkins server by using bash script.**

Step 1: Create Backup Script

vi backup.sh

Make Script Executable

chmod +x backup.sh

Run Backup Script

./backup.sh

#!/bin/bash

# ==========================

# Jenkins Backup Script

# ==========================

# Define Jenkins home and backup directories

JENKINS\_HOME="/var/lib/jenkins"

BACKUP\_DIR="jenkins-backups"

# Define backup file name

BACKUP\_FILE="$BACKUP\_DIR/backup1.tar.gz"

# Create the backup directory if it doesn't exist

mkdir -p "$BACKUP\_DIR"

# Create the compressed tarball of the Jenkins home directory

echo "Creating backup of $JENKINS\_HOME..."

sudo tar -czf "$BACKUP\_FILE" -C "$JENKINS\_HOME" .

# Verify the backup file was created

if [ -f "$BACKUP\_FILE" ]; then

echo "✅ Backup successful! File saved to $BACKUP\_FILE"

else

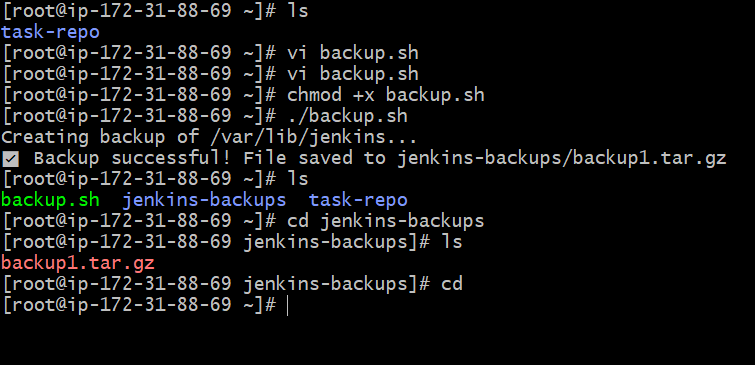
echo "❌ Backup failed!"

exit 1

fi

# Set permissions for the backup file

sudo chown -R jenkins:jenkins "$BACKUP\_DIR"



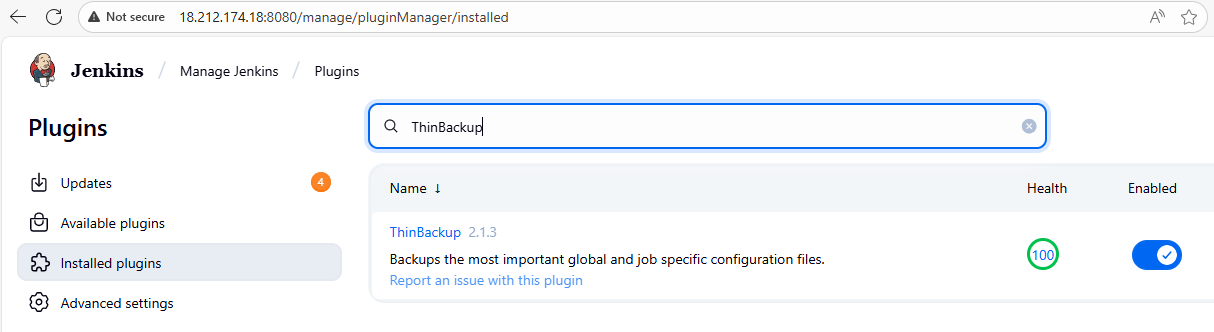
**5) Take backup of Jenkins using thin backup plugin.**

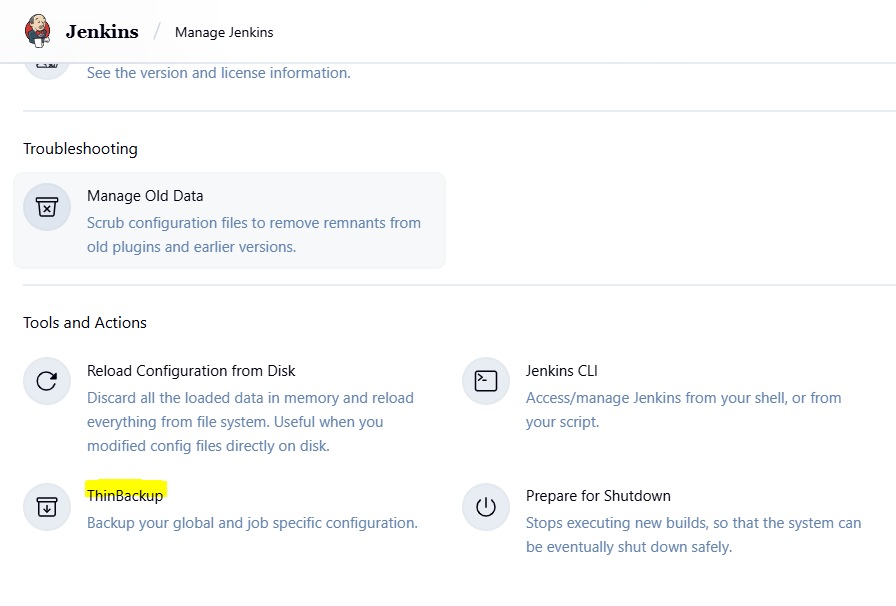
### **Step 1: Access Jenkins**

1. Open your Jenkins URL in a browser: http://<your-server-ip>:8080.
2. Log in with an admin account.

### **Step 2: Install ThinBackup Plugin**

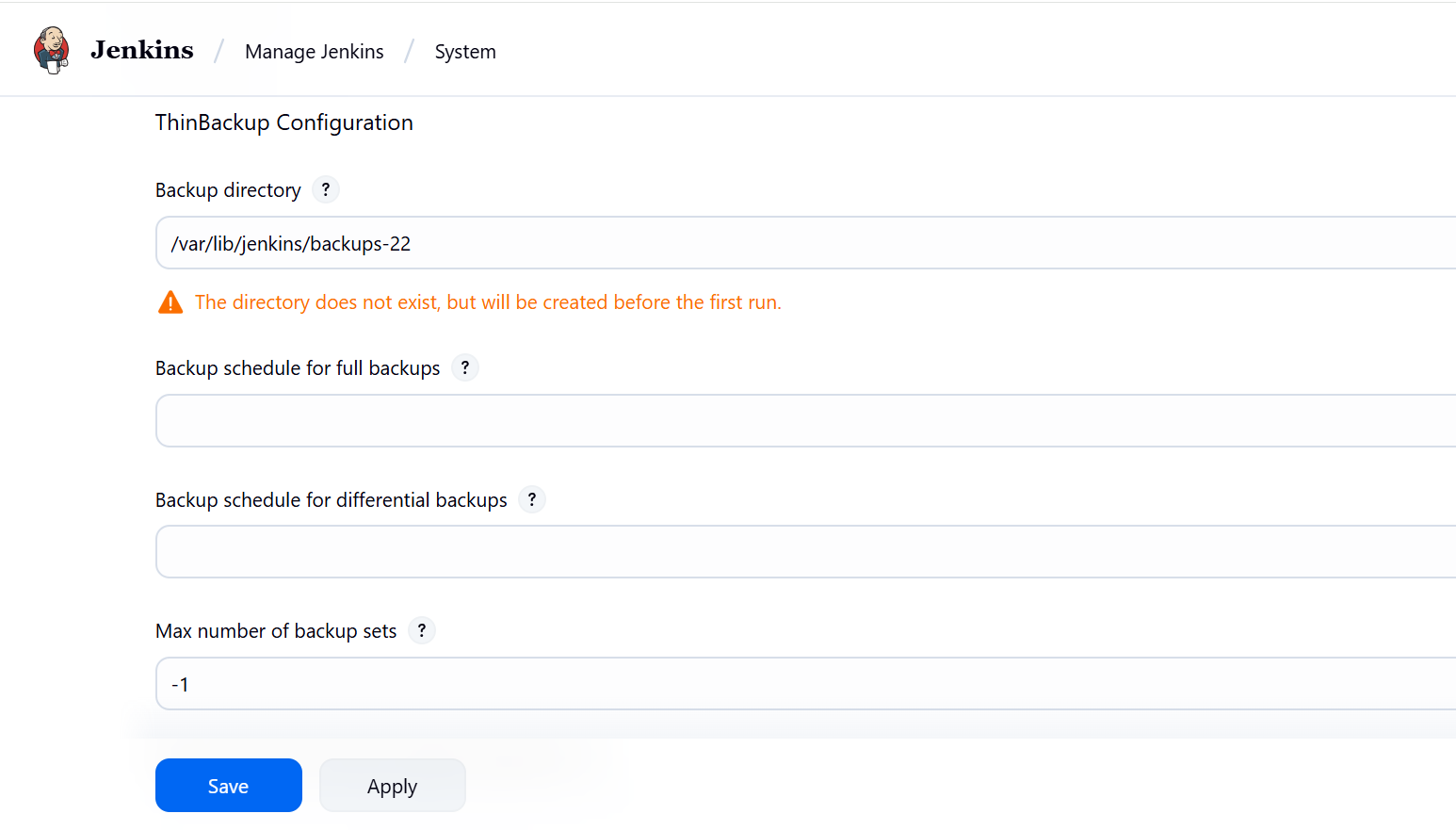
1. Go to **Manage Jenkins** → **Manage Plugins**.
2. Click the **Available** tab.
3. Search for ThinBackup.
4. Check the box next to **ThinBackup**.
5. Click **Install without restart**





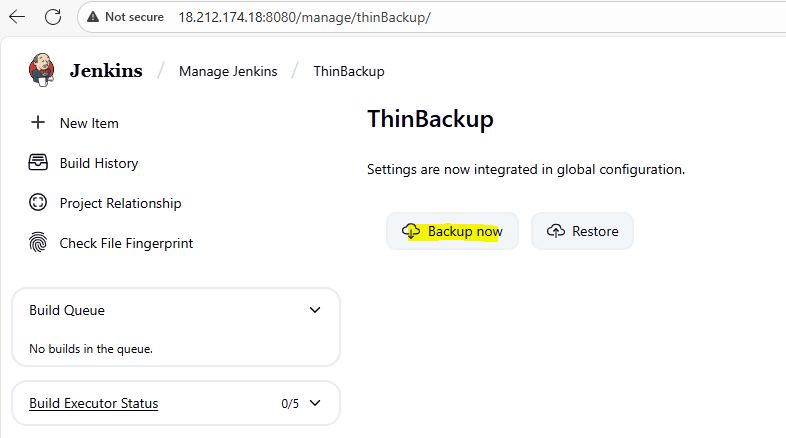
### **Step 3: Configure ThinBackup**

1. Go to **Manage Jenkins** → **System**
2. Click **Configure**.
3. Set the following:
   * **Backup directory**: Path where backups will be stored (e.g., /var/Jenkins-backup).
4. Click **Save**.



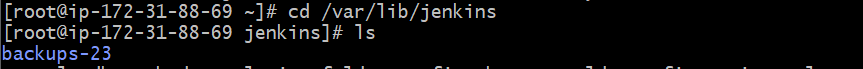
### **Step 4: Take Manual Backup**

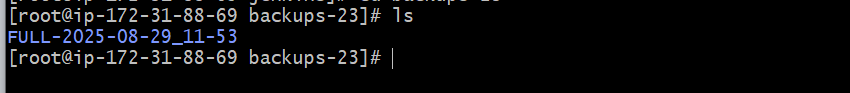
1. Go to **Manage Jenkins** → **ThinBackup**.
2. Click **Backup Now**.
3. The backup will be created in the folder you configured (/var/Jenkins/backups-23).



### **Step 5: Verify Backup**

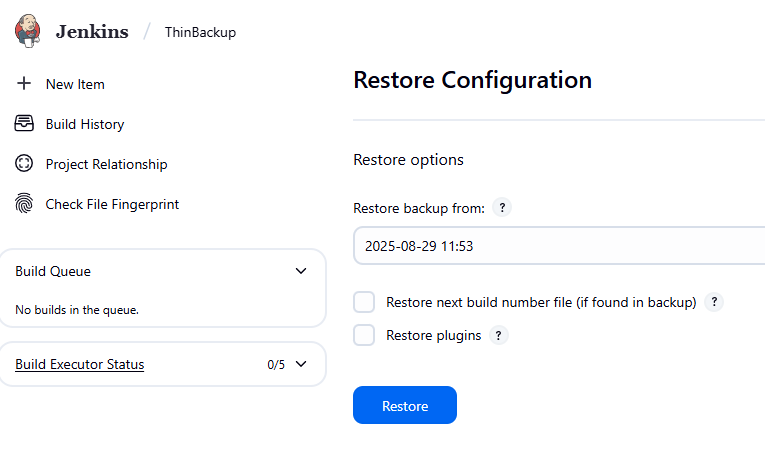
1. Navigate to the backup folder.





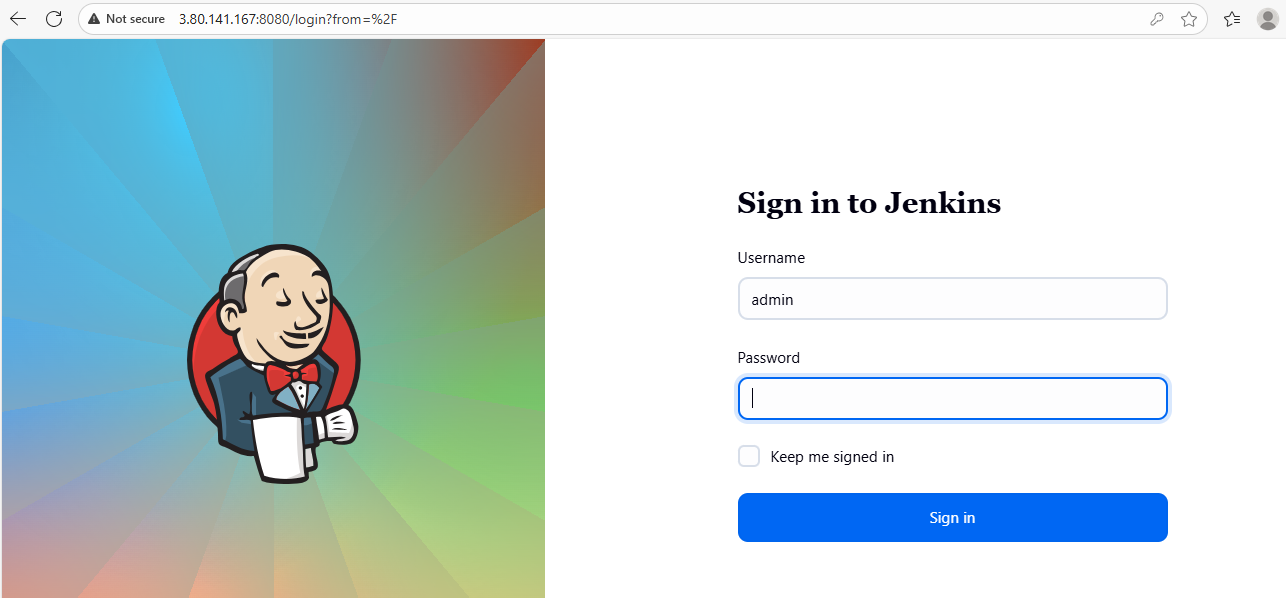
### **Step 6: Restore Backup (Optional) no need in task**

1. Go to **Manage Jenkins** → **ThinBackup**.
2. Click **Restore**.
3. Select the backup you want to restore.
4. Click **Restore** to bring Jenkins back to that state.



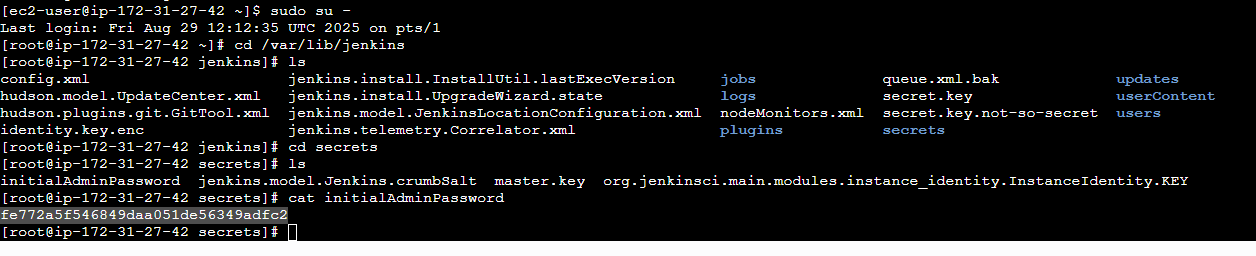
**6Q) Setup a new Jenkins server and dump the backup taken in task4.**

**Before continuing to the below steps, I forgot my backup new server password of new server Jenkins GUI. To get the password and to change the password follow the below steps.**

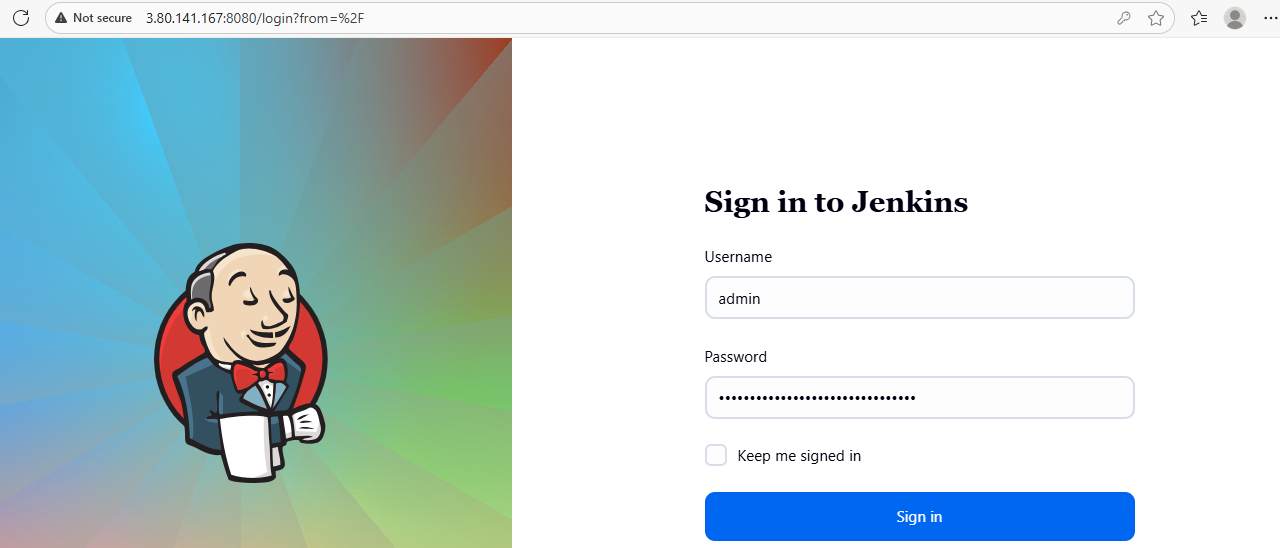


**When you forgot a password to enter above**

**Then goto aws cli and enter the below commands**



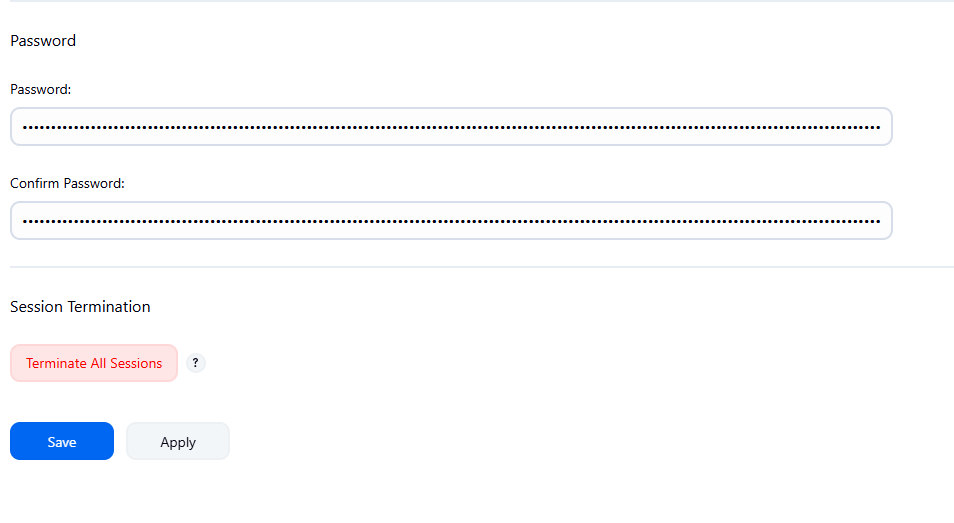
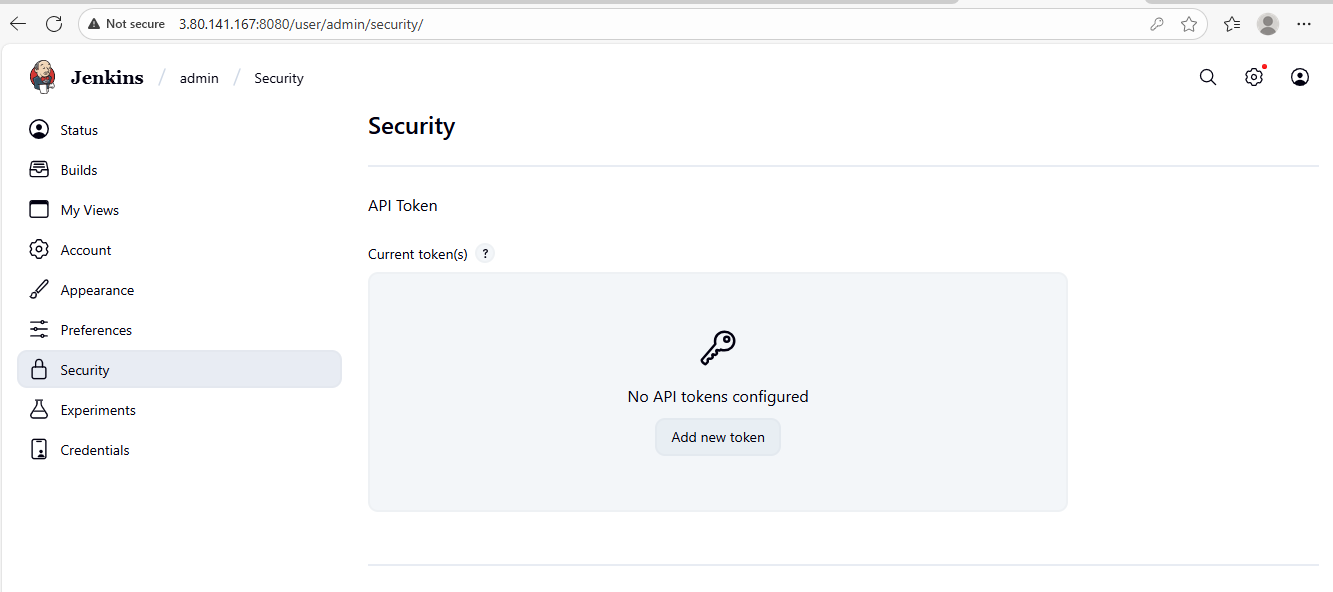
**Copy the intialAdminPassword key and enter into the password.**



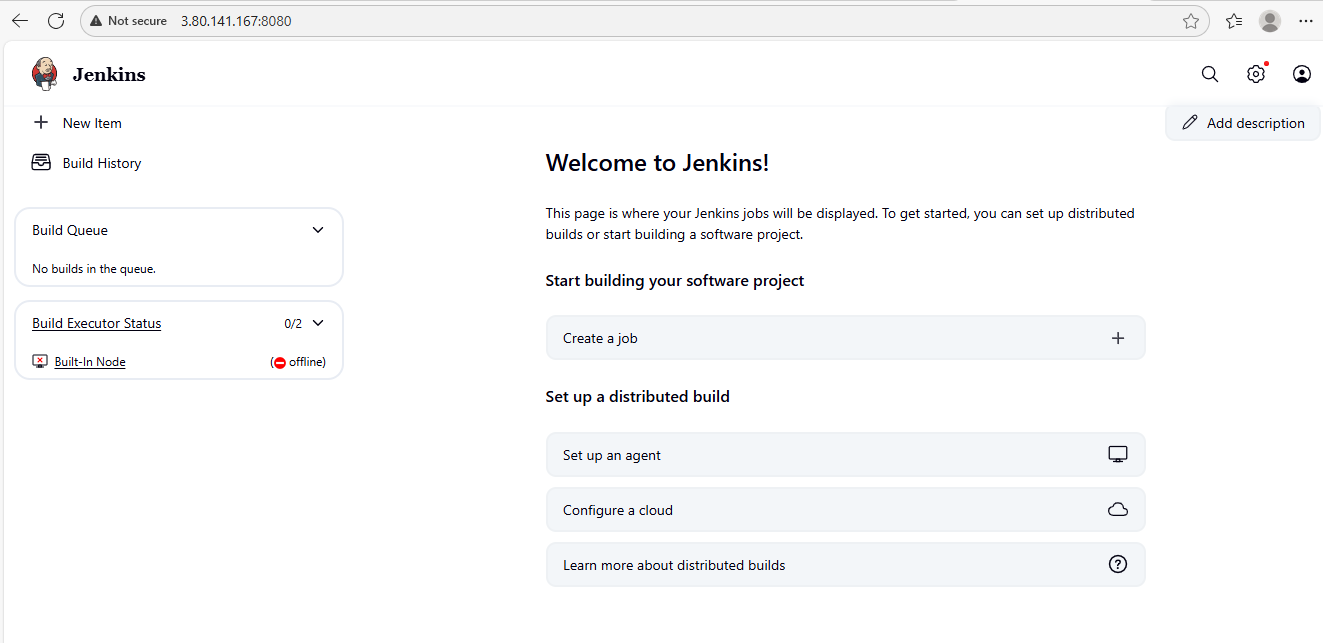
**Then jenkins gui opens and If you want to change to new password then ollow the below steps.**

**Goto Jenkins new server public IP +8080**

**Then go to Jenkins – Admin – let side Security**

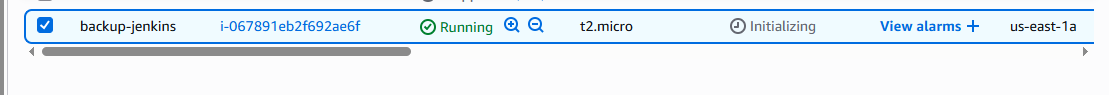


**Here give your new password and confirm password. Click on save.**



# Step 1: Launch New Server & Install Jenkins

1. **Launch a new EC2 instance**



1. Install Java (required for Jenkins)

sudo dnf install -y java-17-amazon-corretto

1. Add Jenkins Repository

**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>

1. Install Jenkins

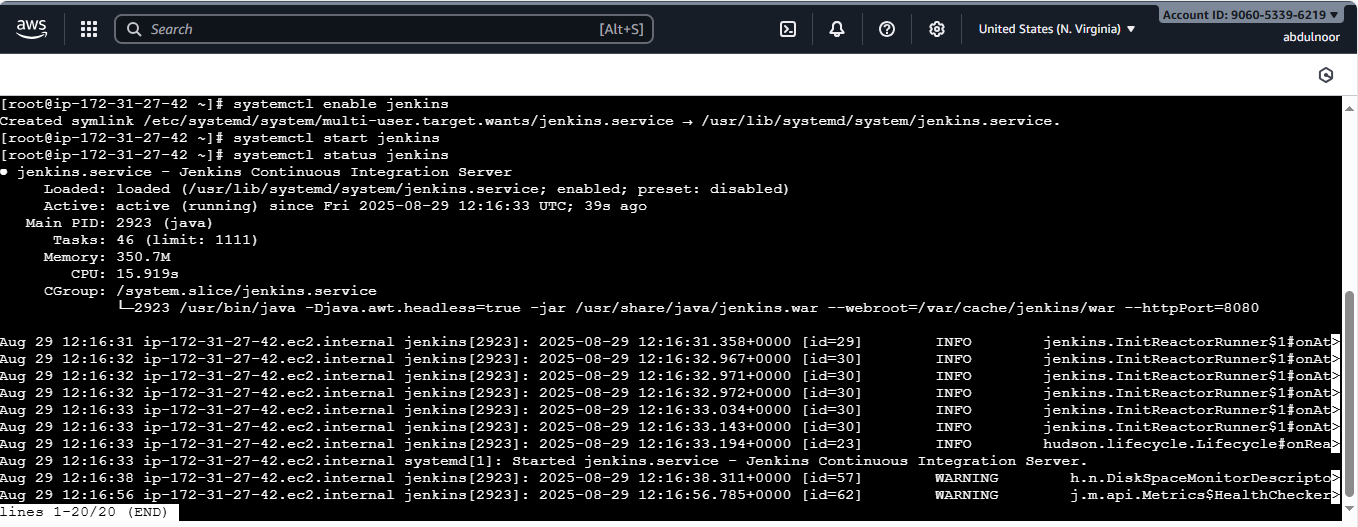
**sudo yum install jenkins –y**

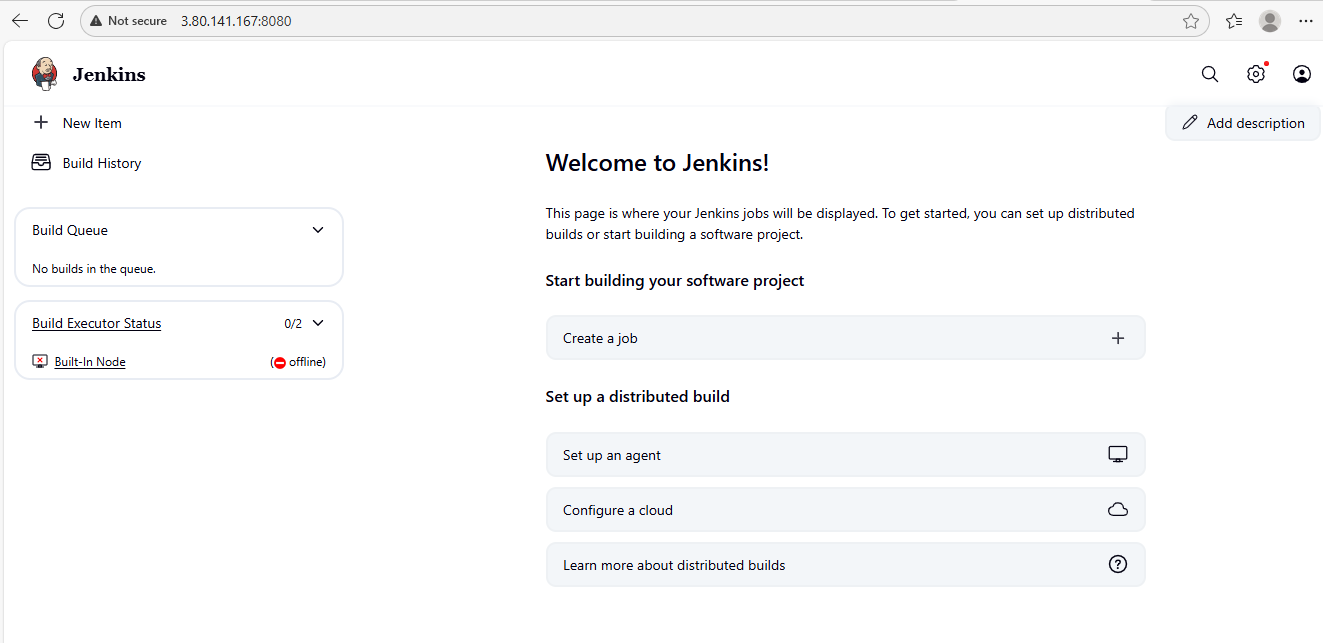
**systemctl enable jenkins**

**systemctl start Jenkins**

**systemctl status Jenkins**

**systemctl enable Jenkins**





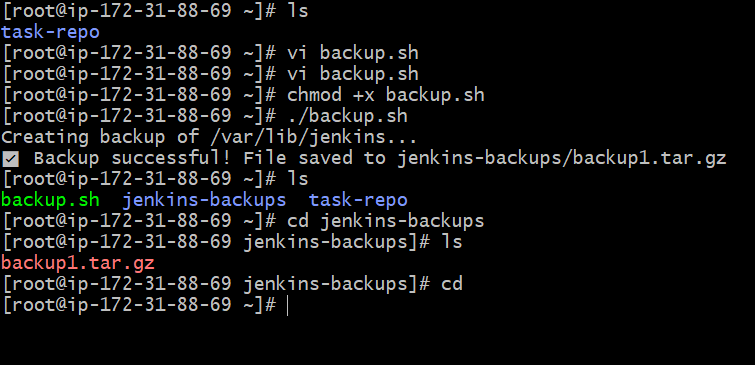
**See in my new server I don’t have any jobs and see how restore backup file into this new server and whatever backup files are there of Jenkins master server it will appear here.**

Step 2: Copy Your Backup to New Server

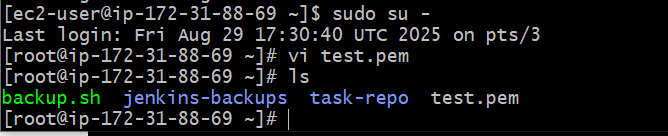
Login into your old Jenkins main server you took backup

Create an test.pem file where you copy your pemkey of new-server and and paste in the oldserver

Previous image below one just for inormation



New image below one



[root@ip-172-31-88-69 ~]# scp -i test.pem jenkins-backups/backup1.tar.gz [ec2-user@3.80.141.167:/tmp/](mailto:ec2-user@3.80.141.167:/tmp/)

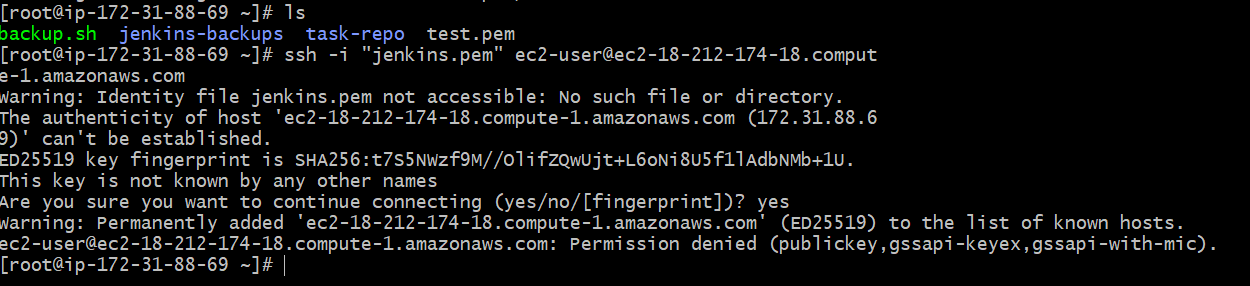
test.pem is your new server pemkey

jenkins-backups - is the folder containing the backup tar file

backup1.tar.gz -backup file

[ec2-user@44.203.148.202](mailto:ec2-user@44.203.148.202) - public ip of new server

/tmp – is the location in new server where you send your backup tar file



**Step 3:** Login in to new server and check the file in /tmp

backup1.tar.gz

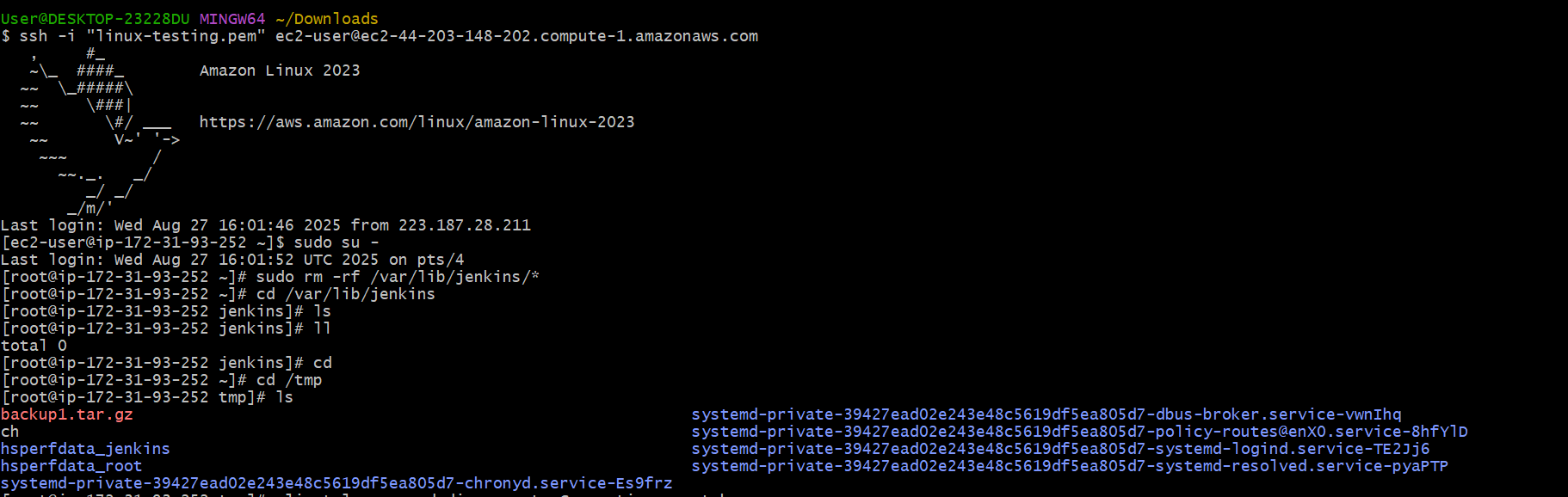
beore that delete the /var/lib/jenkins/ because it is a default of Jenkins which created at the time Jenkins installation in the new server.

sudo rm -rf /var/lib/jenkins/\*

cd /var/lib/Jenkins

ls

ll



Remove existing Jenkins data. Delete the contents of the new $JENKINS\_HOME to prepare for the restore.

sudo rm -rf /var/lib/jenkins/\*

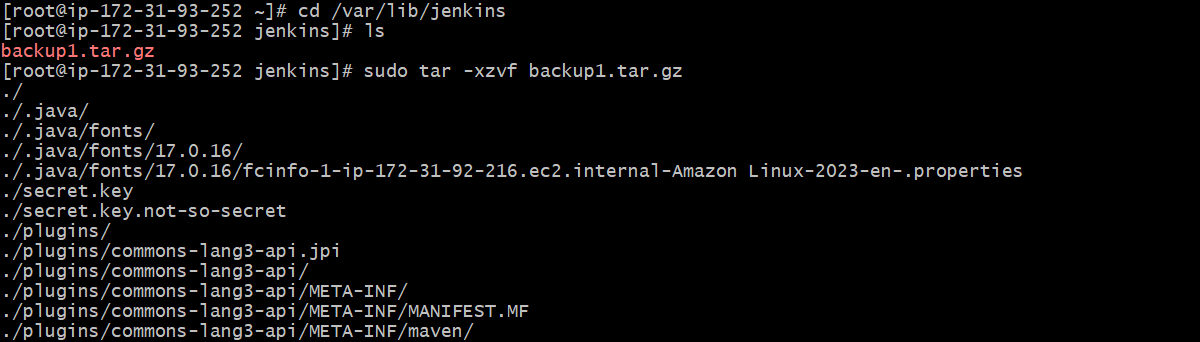
Copy the file from tmp to /var/lib/jenkins/

cp /tmp/backup1.tar.gz /var/lib/jenkins/

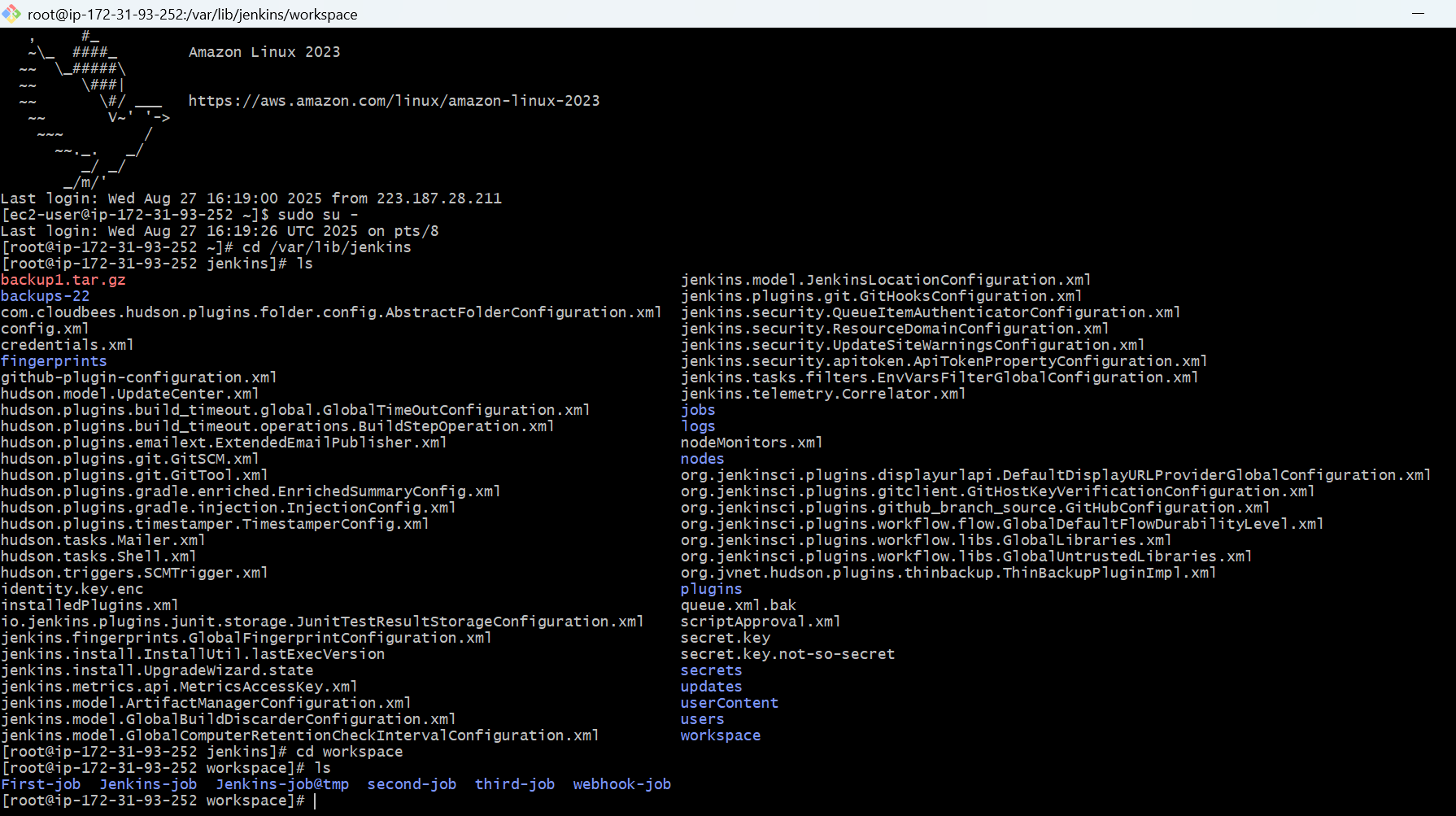
then extracts

cd /var/lib/jenkins

sudo tar -xzvf backup1.tar.gz



(Or you can directly extract from tmp by using sudo tar -xzf /tmp/backup1.tar.gz –C /var/lib/Jenkins)



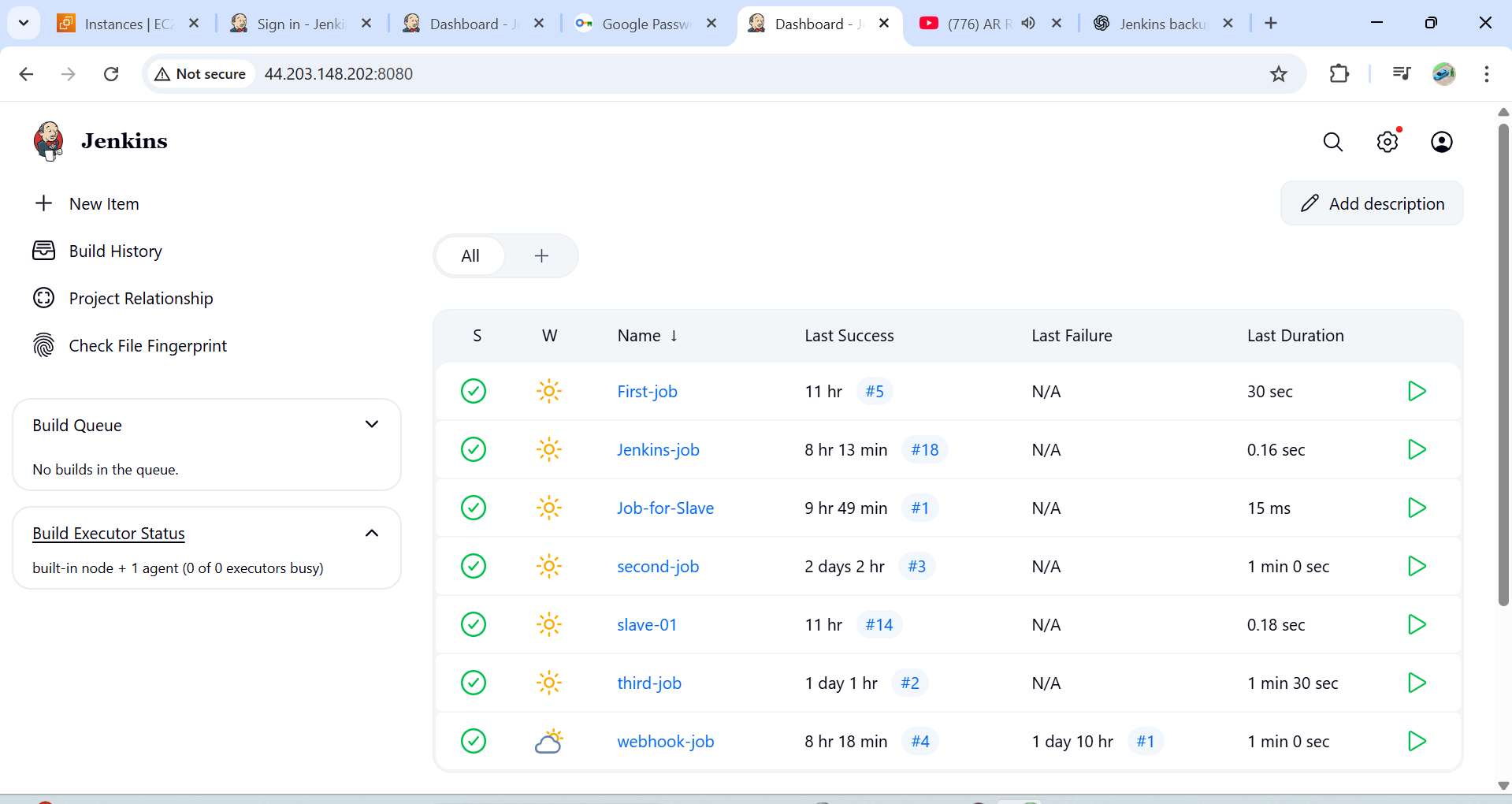
sudo systemctl start jenkins

sudo systemctl status Jenkins

Verify in browser

http://<EC2-Public-IP>:8080

new server by default running on port no-: 8080



Old server running on port –no. :- 8081

