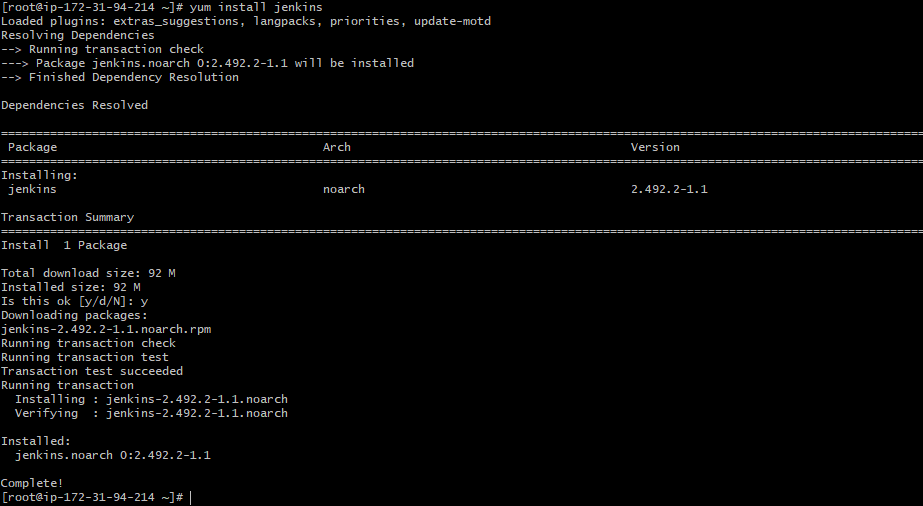
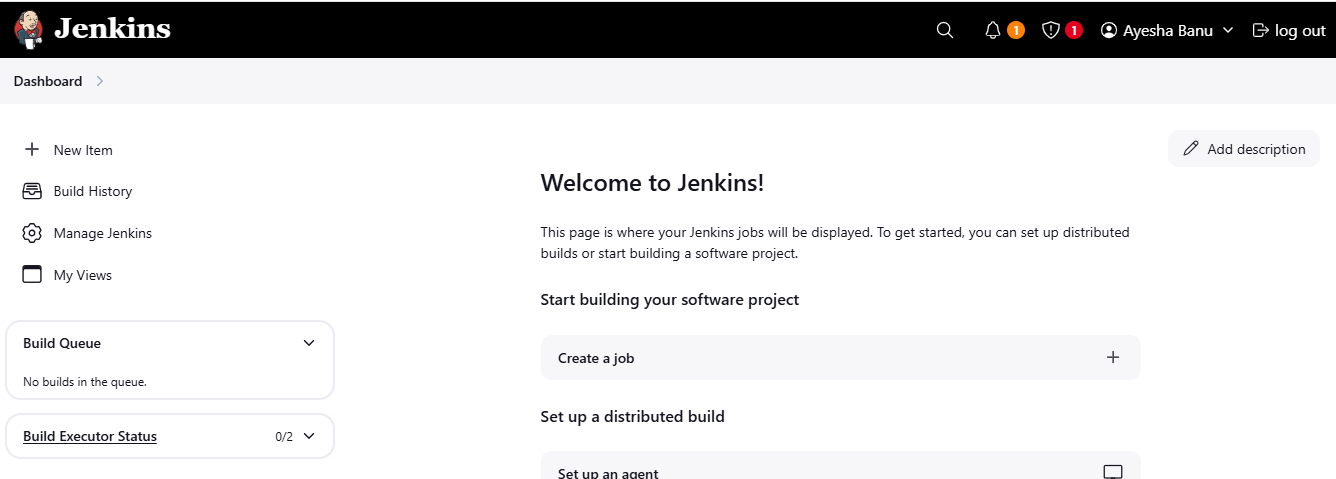
1. Configure 2 slave machines in jenkins master.

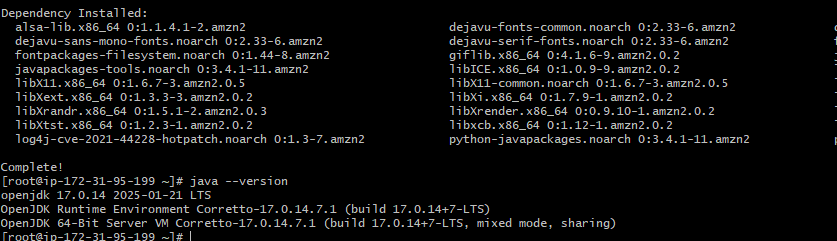
Create 3 ec2 instances ; master, slave01 and slave-02 and install Jenkins in master



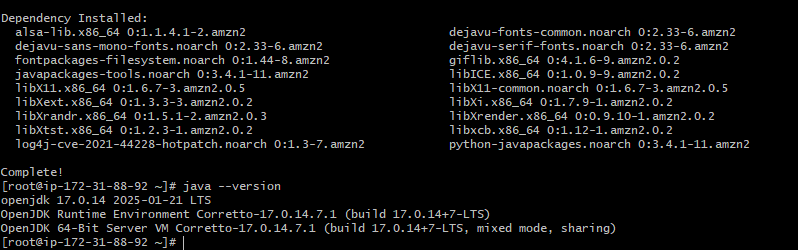
After giving the specific configuration we can see the Jenkins gui



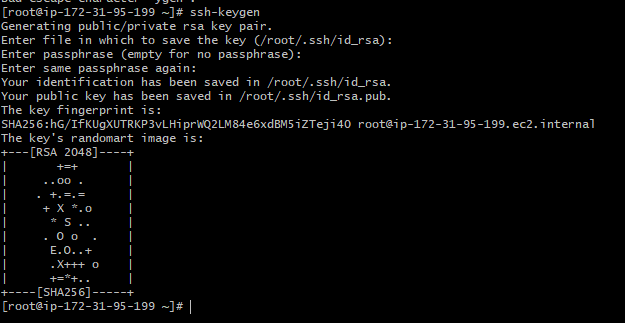
Install java in slave-01

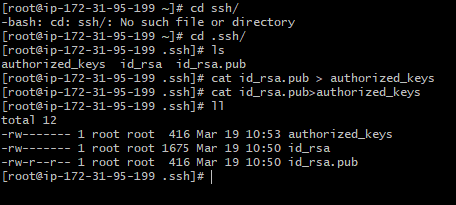


Install java in slave-02

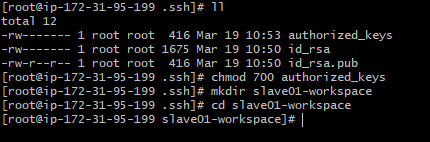


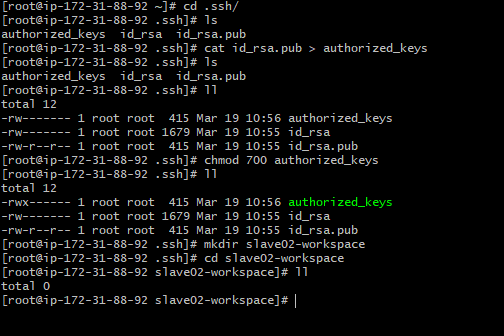
Step -02 :- create a ssh-keygen





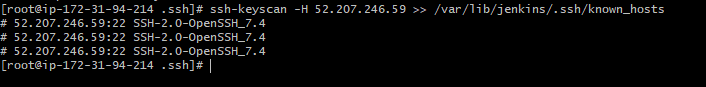
Giving the permission to the authorized\_key and creating the workspace in slave01

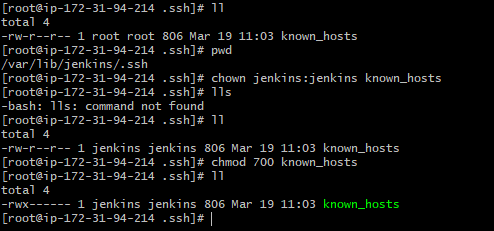




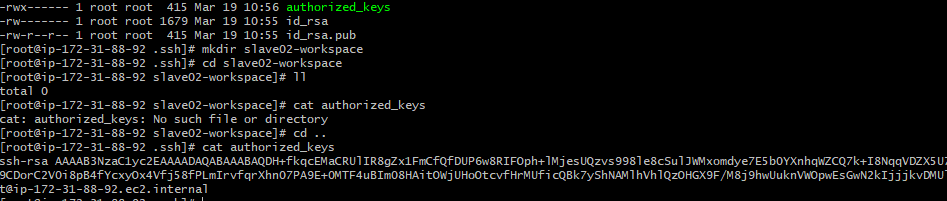
For slave 01

We need to give a command to copy the key from the slave01

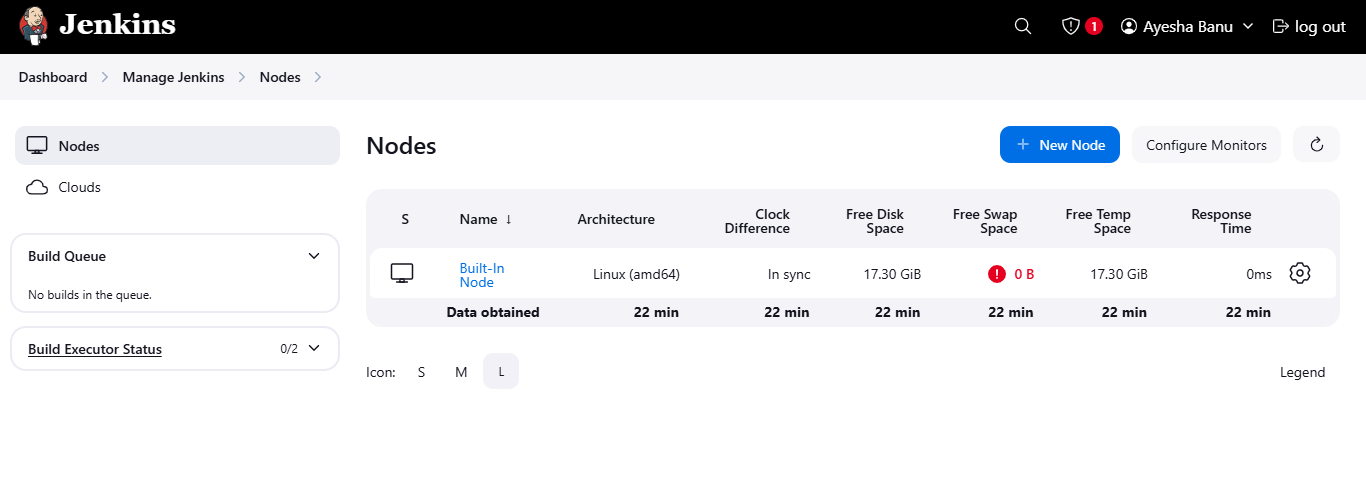


We need to change a permission for the known\_hosts file

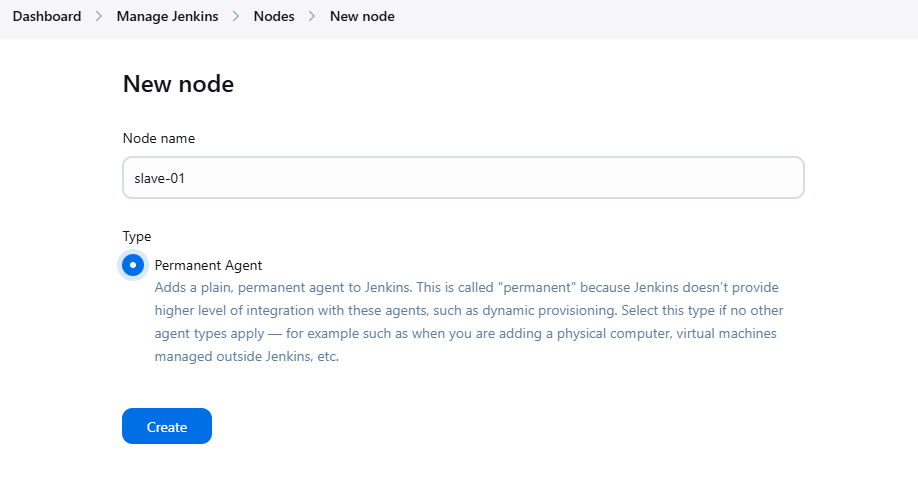
For slave -02

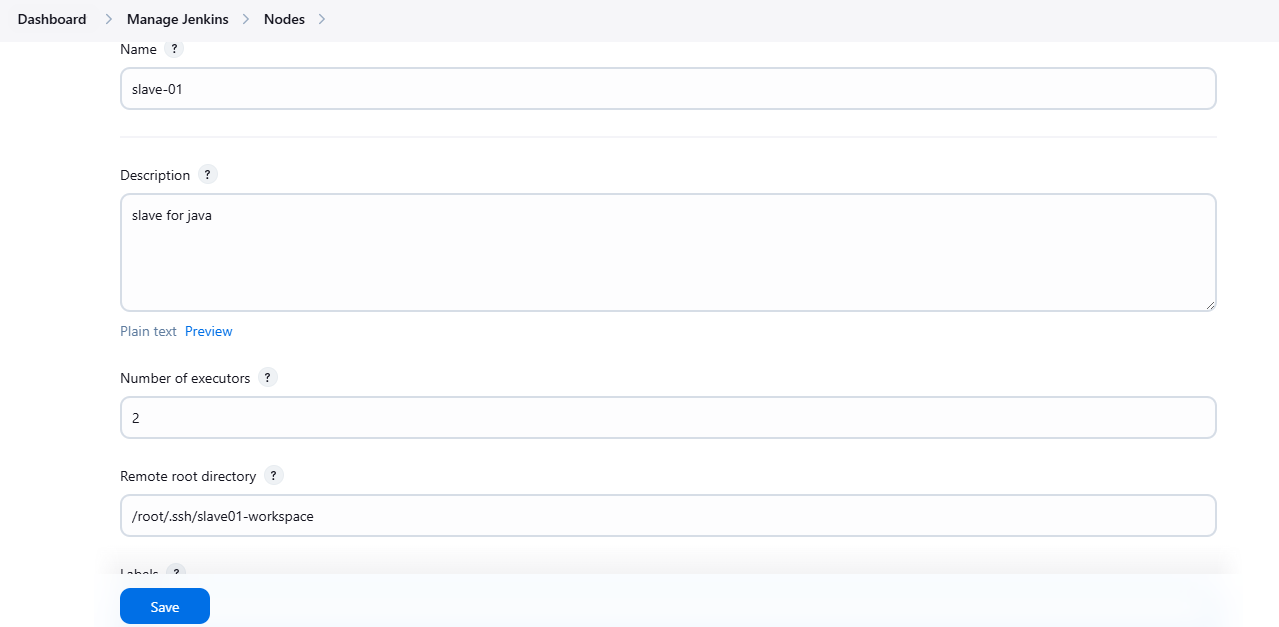


In Jenkins, goto Dashboard🡪 Manage Jenkins🡪nodes🡪create node



Give the name and select type 🡪 permanent agent🡪create

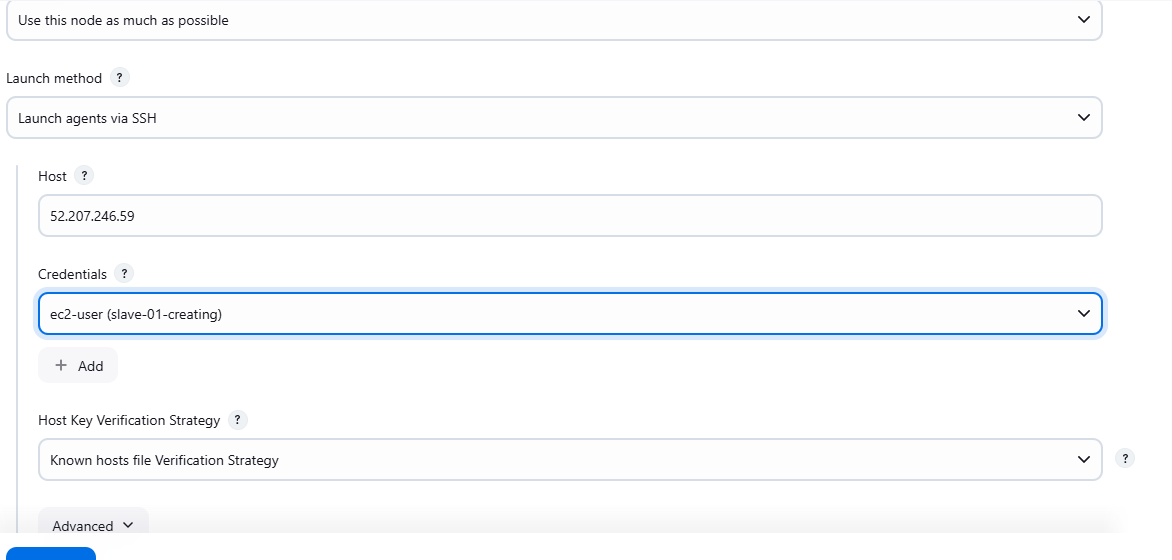






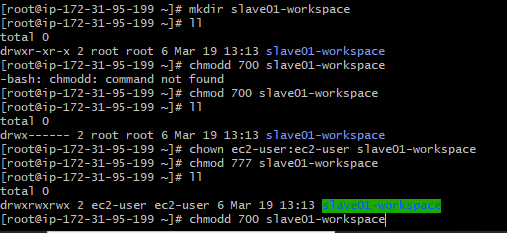
Need to pass the pem key in the key feild





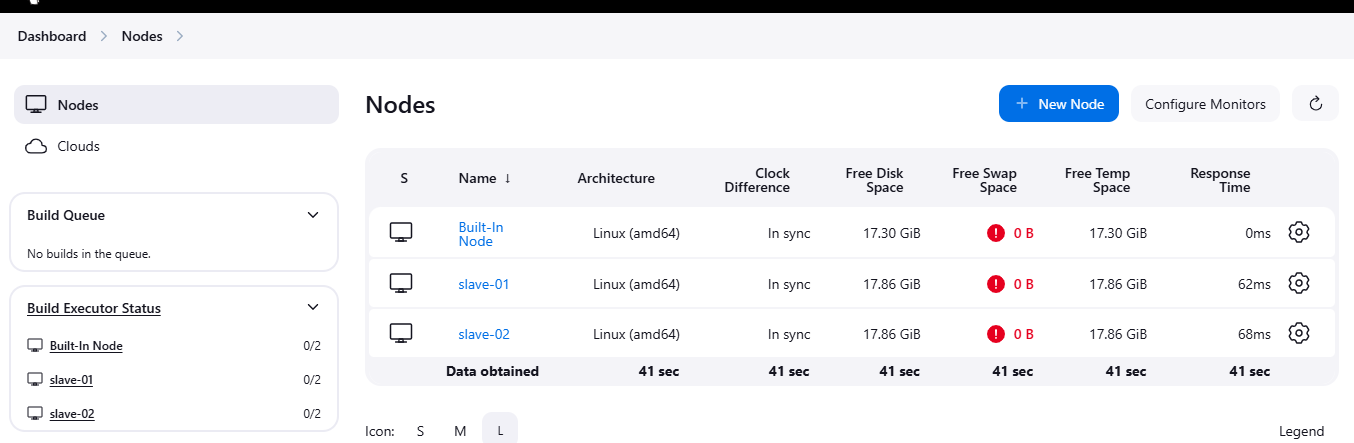
Slave01 has been created And slave -02 as well

Need to change ownership and give permission to the file



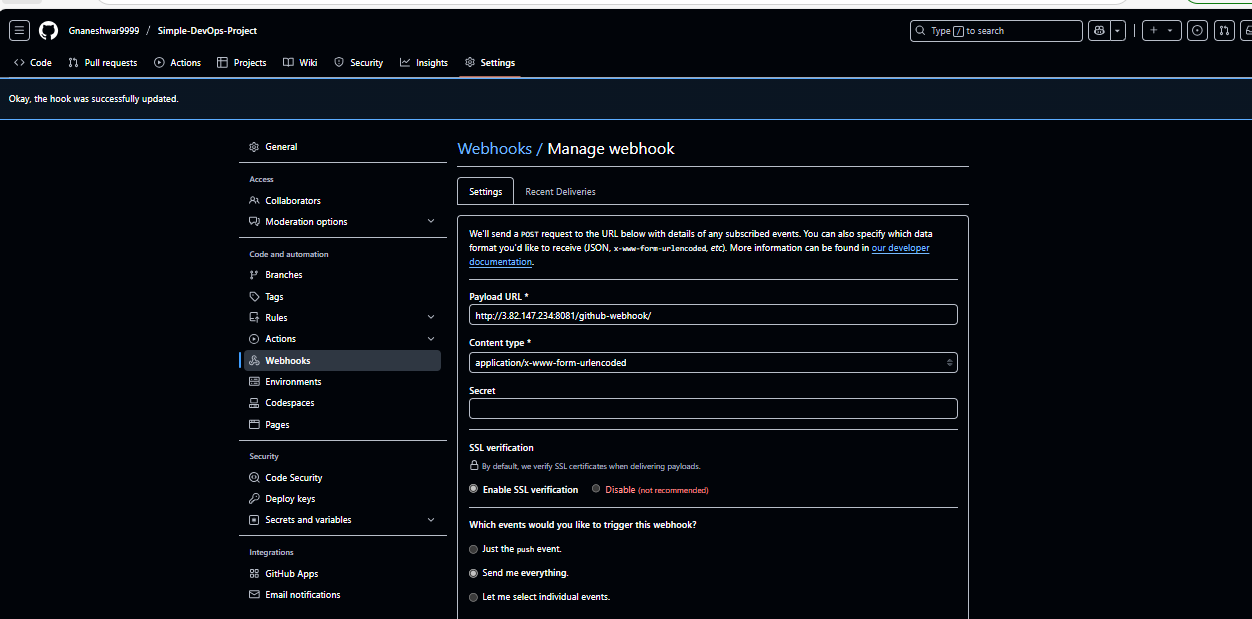
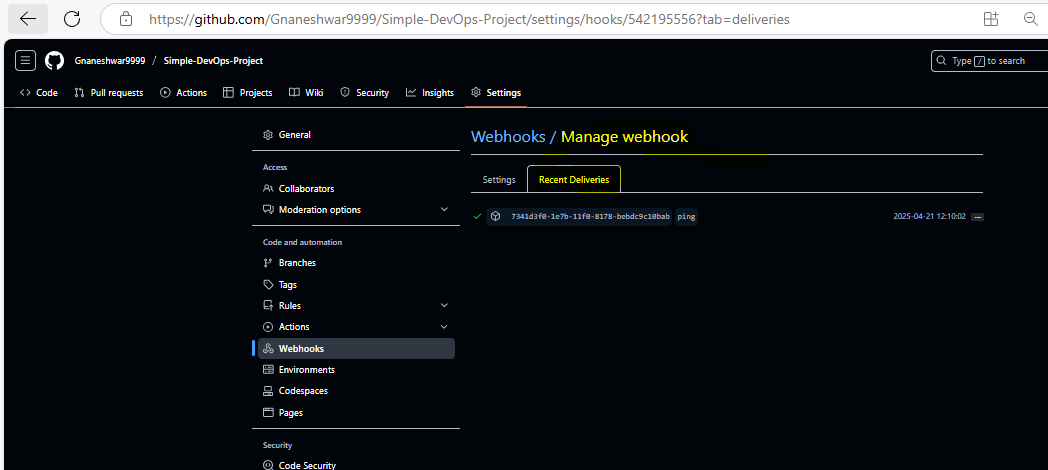
Need to change the root directory



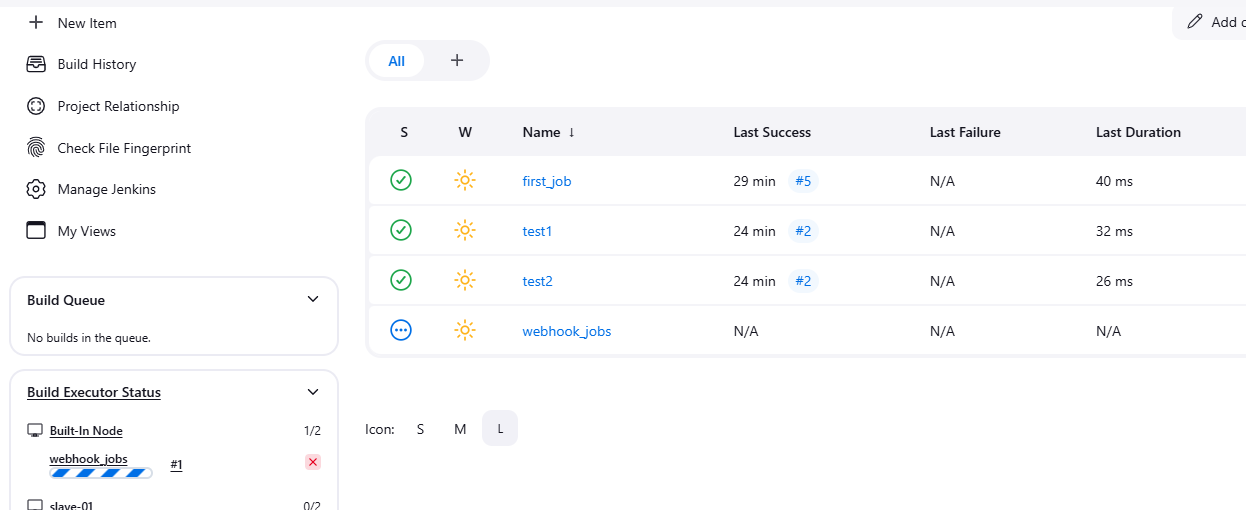


1. Configure webhooks to jenkins job.

Go to github 🡪 select repo –> settings 🡪 webhooks 🡪Add webhooks

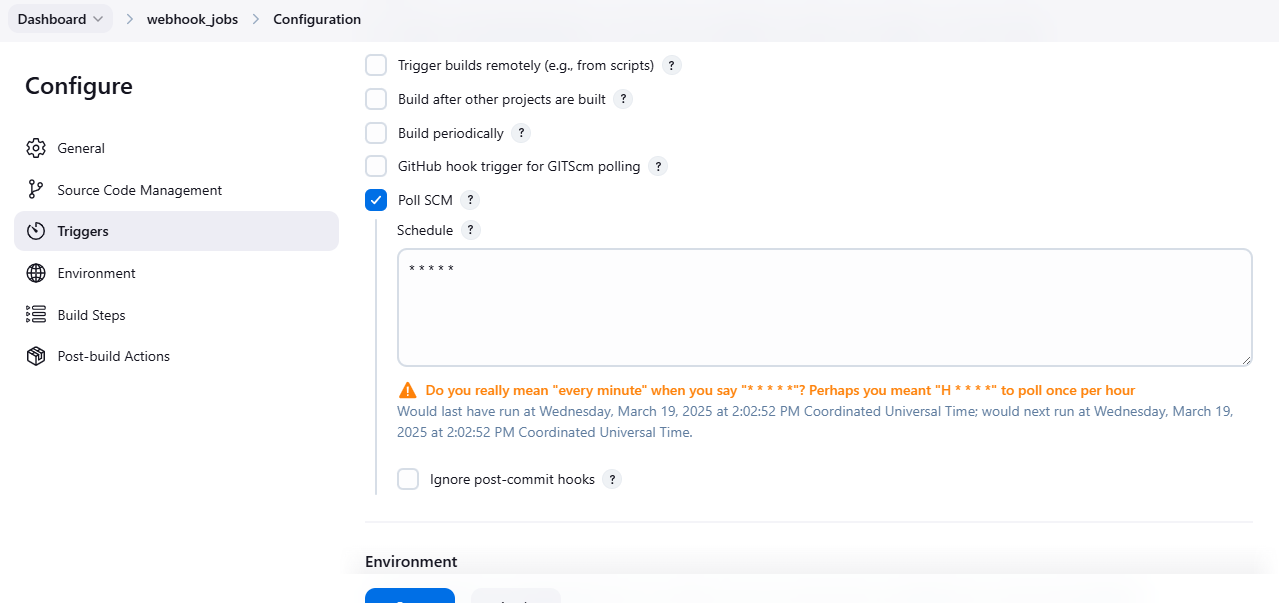
 

Add the changes in github repo and commit the changes it will automatically trigger to the Jenkins

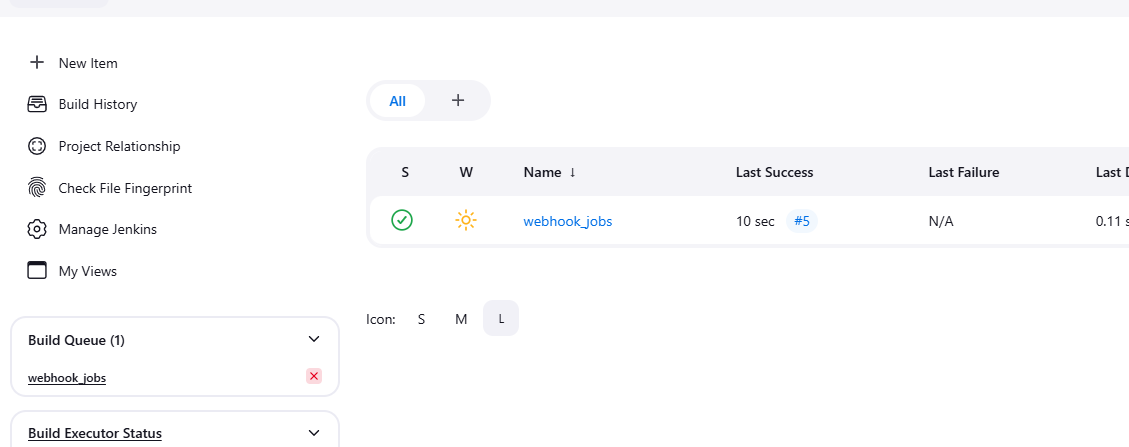


3)Configure poll scm and build periodical options in jenkins job.

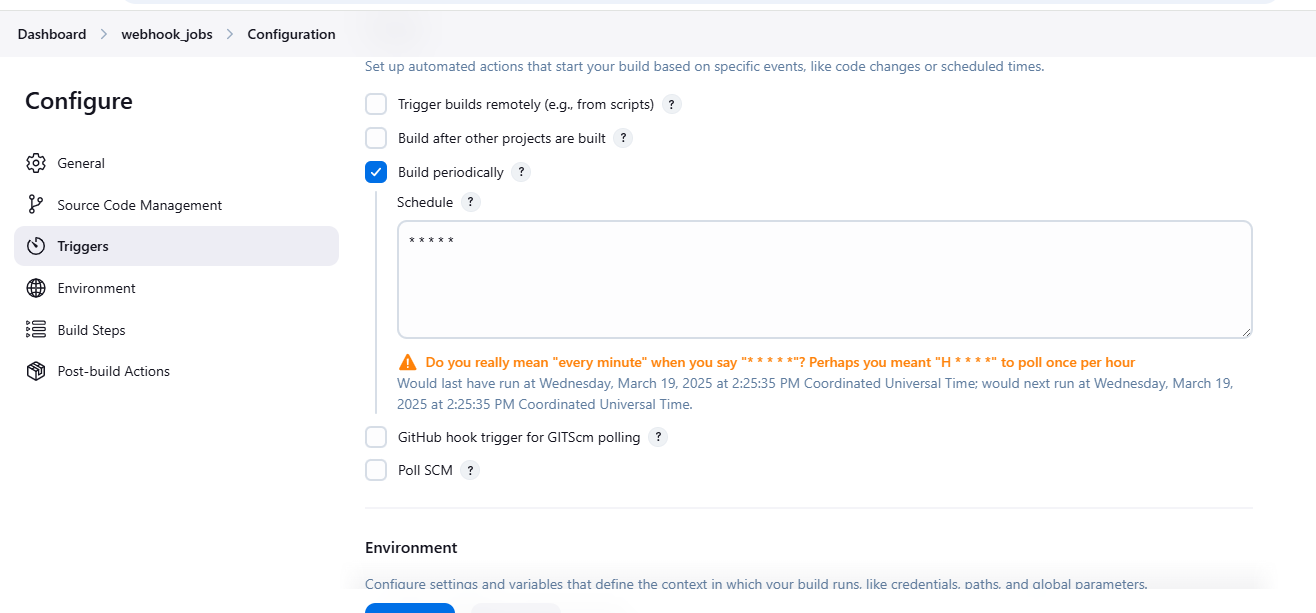
Select the options in configure 🡪 poll SCM and take \* \* \* \* \* and save



Need to change in the github and it will trigger in the jenkins

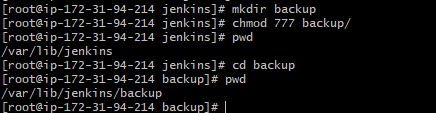


For build periodically we need to select the build periodically and give the \* \* \* \* \* and save it



4)Take backup of jenkins server by using bash script.

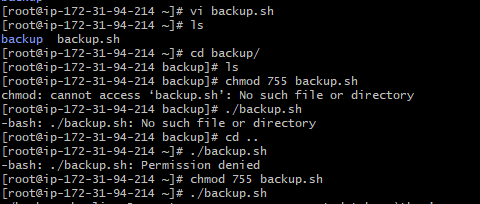
Creating a directory by using the path cd/var/lib/Jenkins/backup



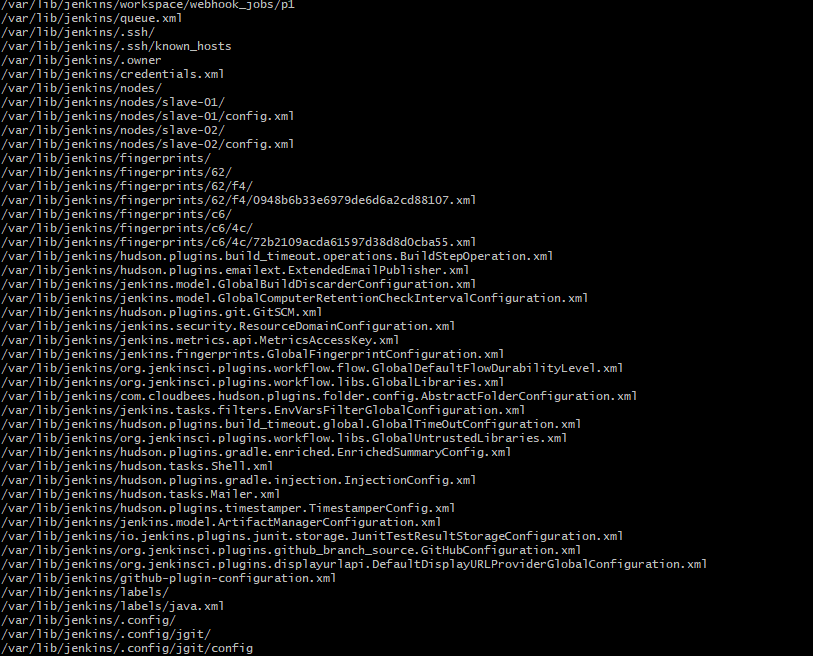
By using vi editor and save this below script and execute the script

**#!/bin/bash  
# Backup directory  
BACKUP\_DIR="/backup/jenkins"  
TIMESTAMP=$(date +"%Y-%m-%d\_%H-%M-%S")  
BACKUP\_FILE="jenkins\_backup\_$TIMESTAMP.tar.gz"  
# Jenkins home directory  
JENKINS\_HOME="/var/lib/jenkins"  
# Days to keep old backups  
RETENTION\_DAYS=7  
echo "Starting Jenkins backup..."  
# Stop Jenkins service (optional but safer)  
sudo systemctl stop jenkins  
# Ensure backup directory exists  
mkdir -p "$BACKUP\_DIR"  
# Create a compressed backup  
tar -czvf "$BACKUP\_DIR/$BACKUP\_FILE" "$JENKINS\_HOME"  
# Restart Jenkins service**

**sudo systemctl start jenkins  
echo "Backup completed: $BACKUP\_DIR/$BACKUP\_FILE"  
# Delete old backups  
find "$BACKUP\_DIR" -name "jenkins\_backup\_\*.tar.gz" -mtime +  
$RETENTION\_DAYS -exec rm -f {} \;  
echo "Old backups older than $RETENTION\_DAYS days deleted."**



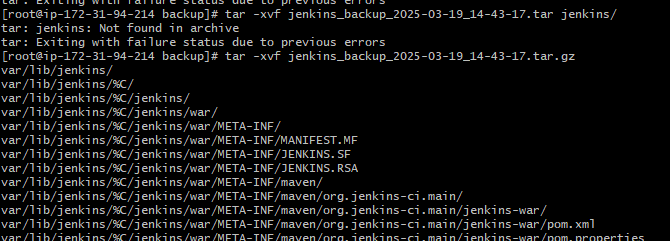
Executing the .backup.sh file



We have created the one tar.gz file backup floder



Untar the file we can see the backup files there



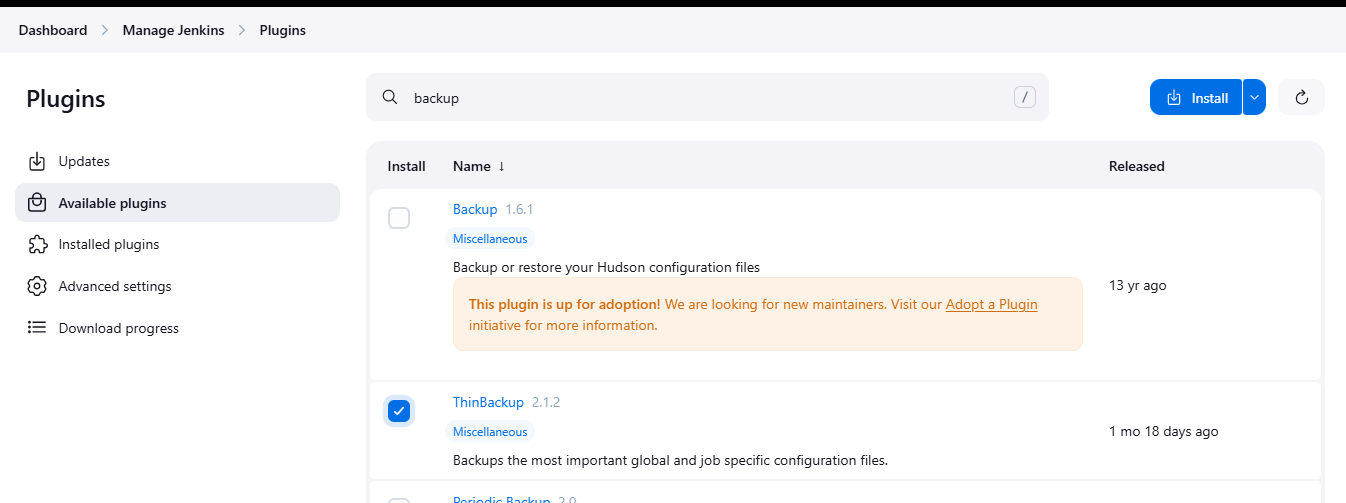


5)Take backup of jenkins using rethin backup plugin.

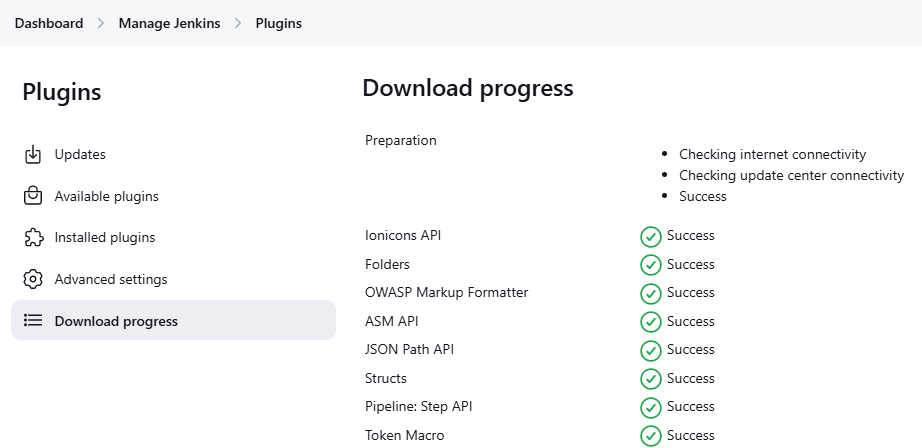
To take a backup by using the thin backup plugin

We need to install the plugin by using the path

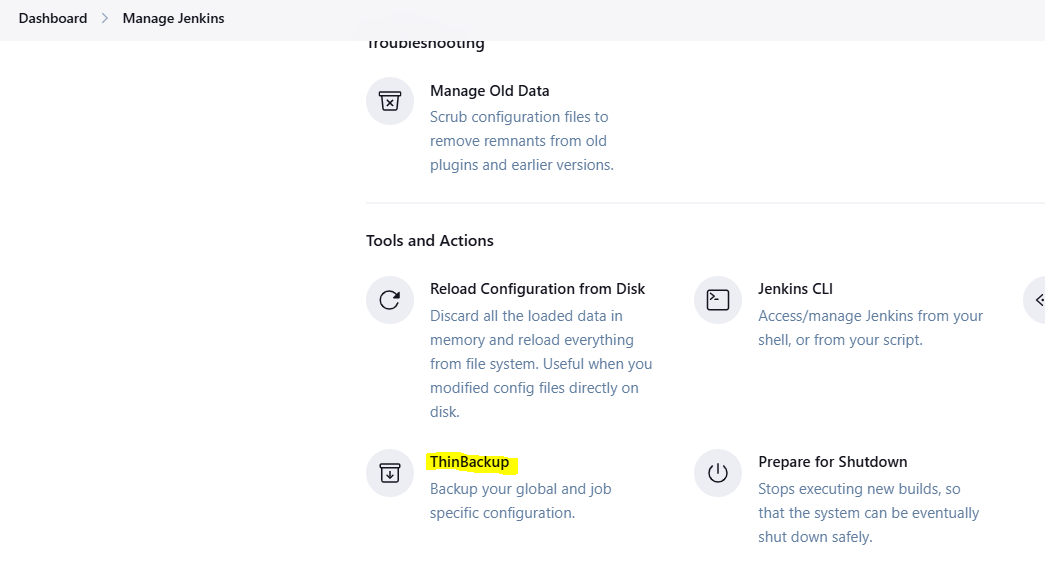
Dashboard🡪manage Jenkins🡪available plugins 🡪thin backup 🡪install



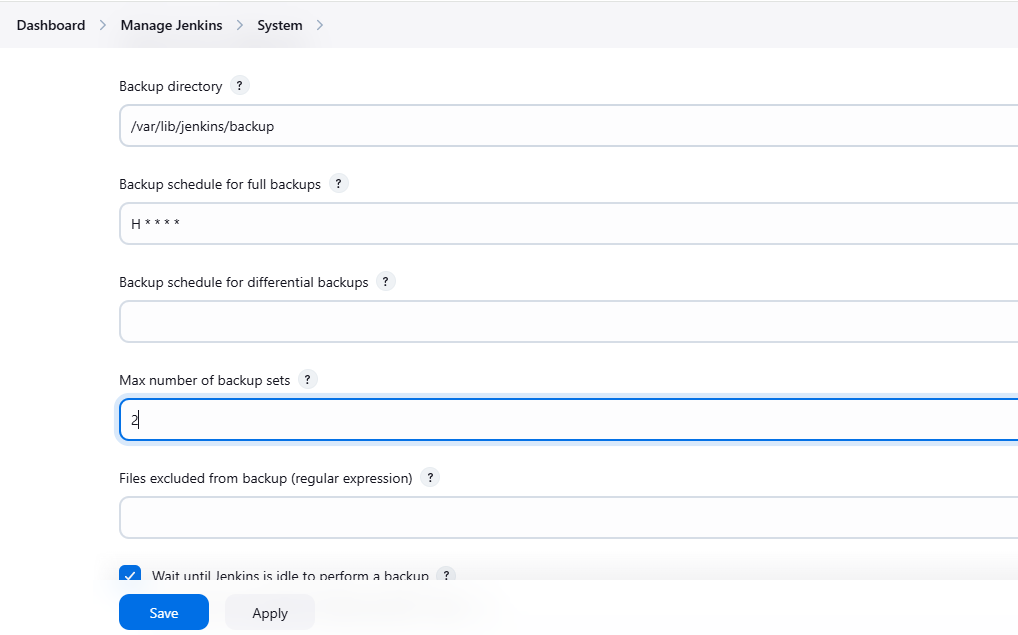
Downloaded the plugins



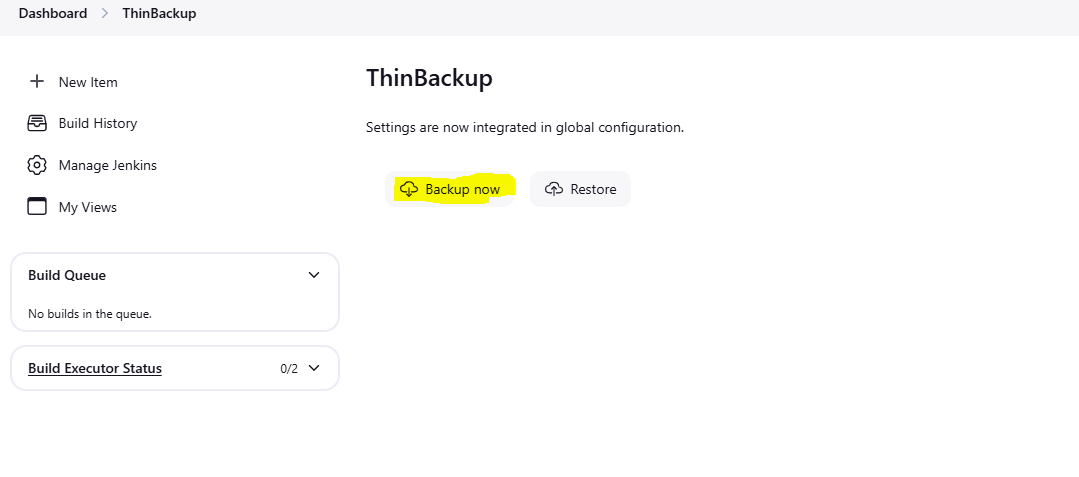
Here we can see the thin backup in the tools and actions



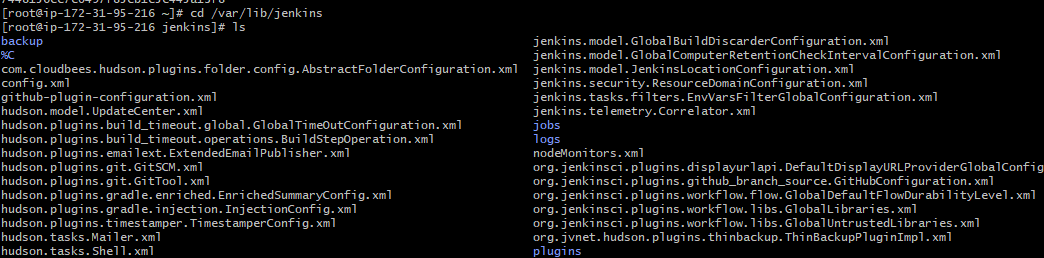
Configure the thin backup



Take bakup now

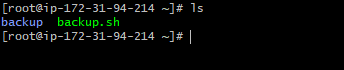


We can see the backup file in cli note

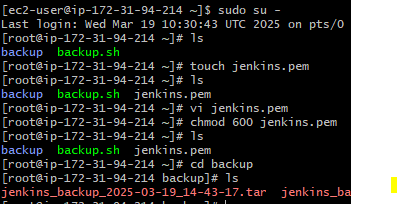


6) Setup a new jenkins server and dump the backup taken in task4.

Create a instance for Jenkins backup and we have a backup file in our master instance

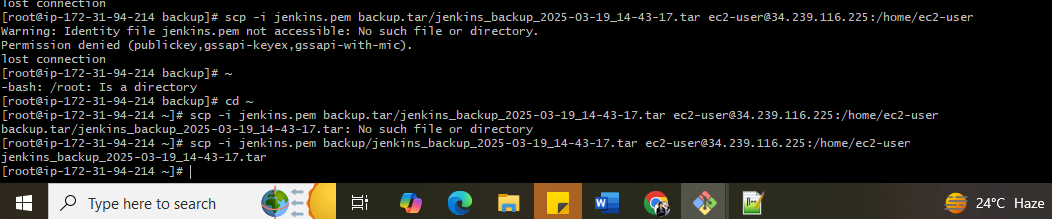


We need to copy pem key file of backup server to master server



Now use Secure copy

Scp -i pemkey backup(.tar file) ec2-user@backup server ip:/home/ec2-user



We can see the backup file in backup server as well

