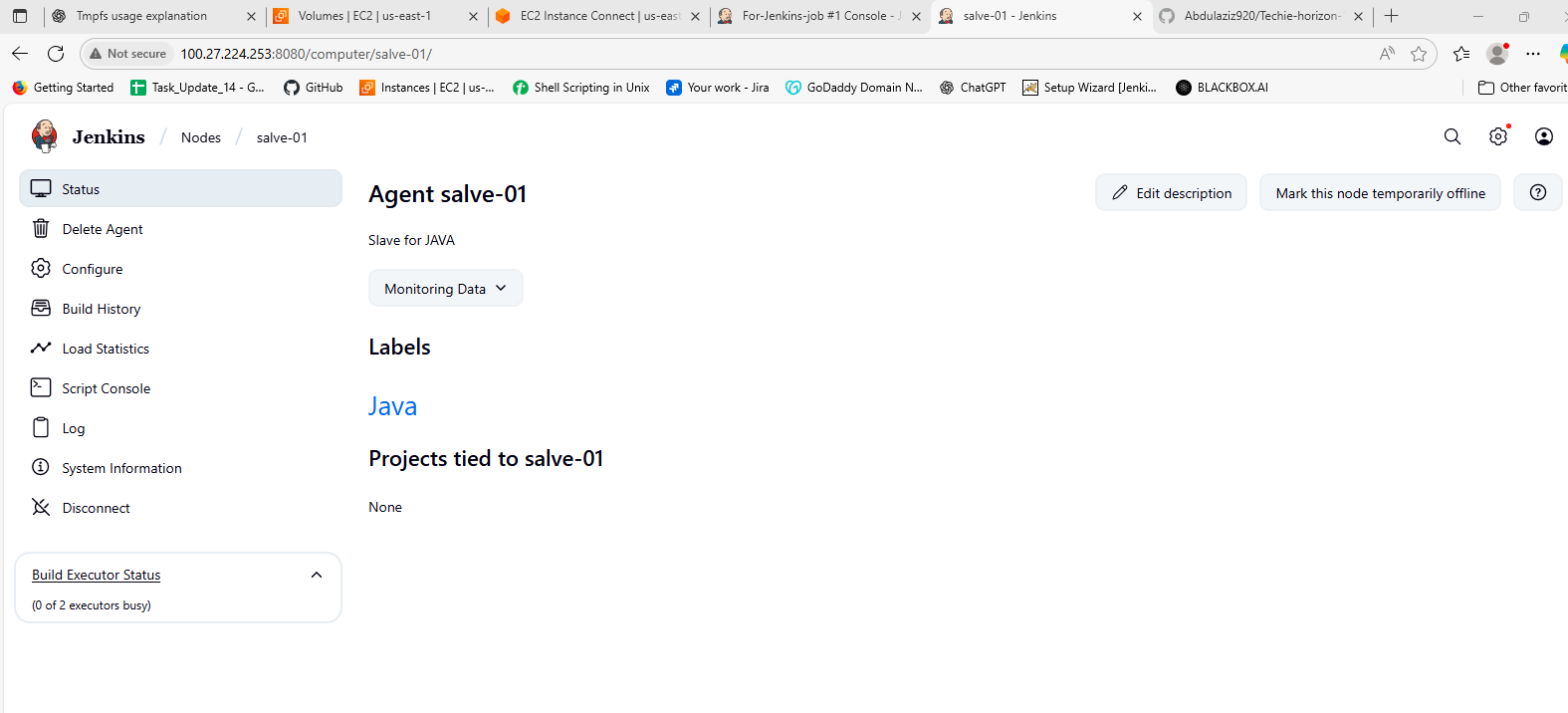
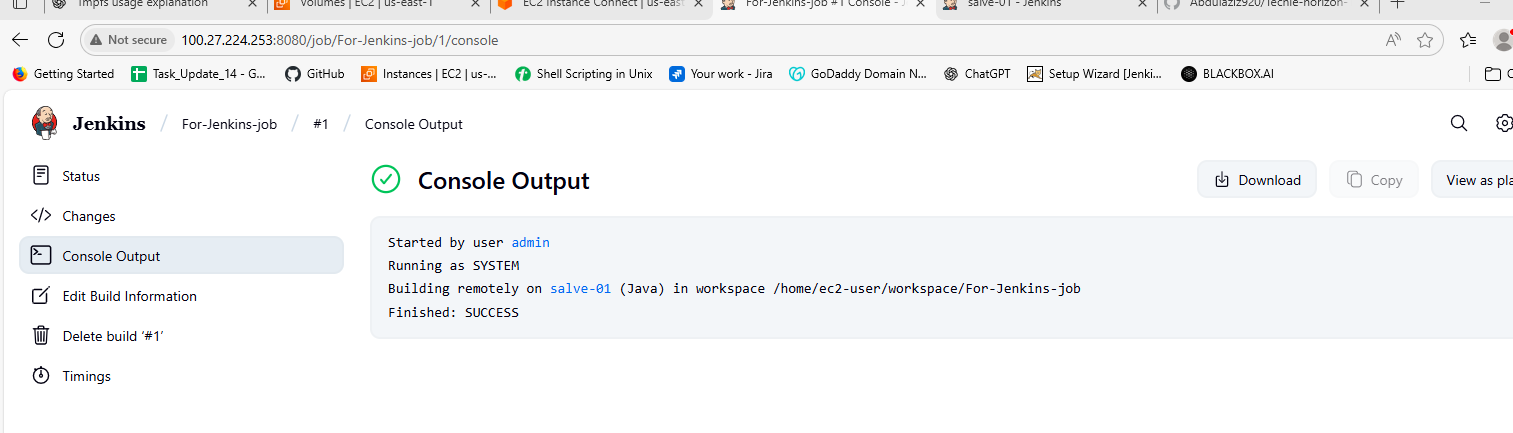
**JENKINS-02 TASK**

1. Configure 2 slave machines in Jenkins master.

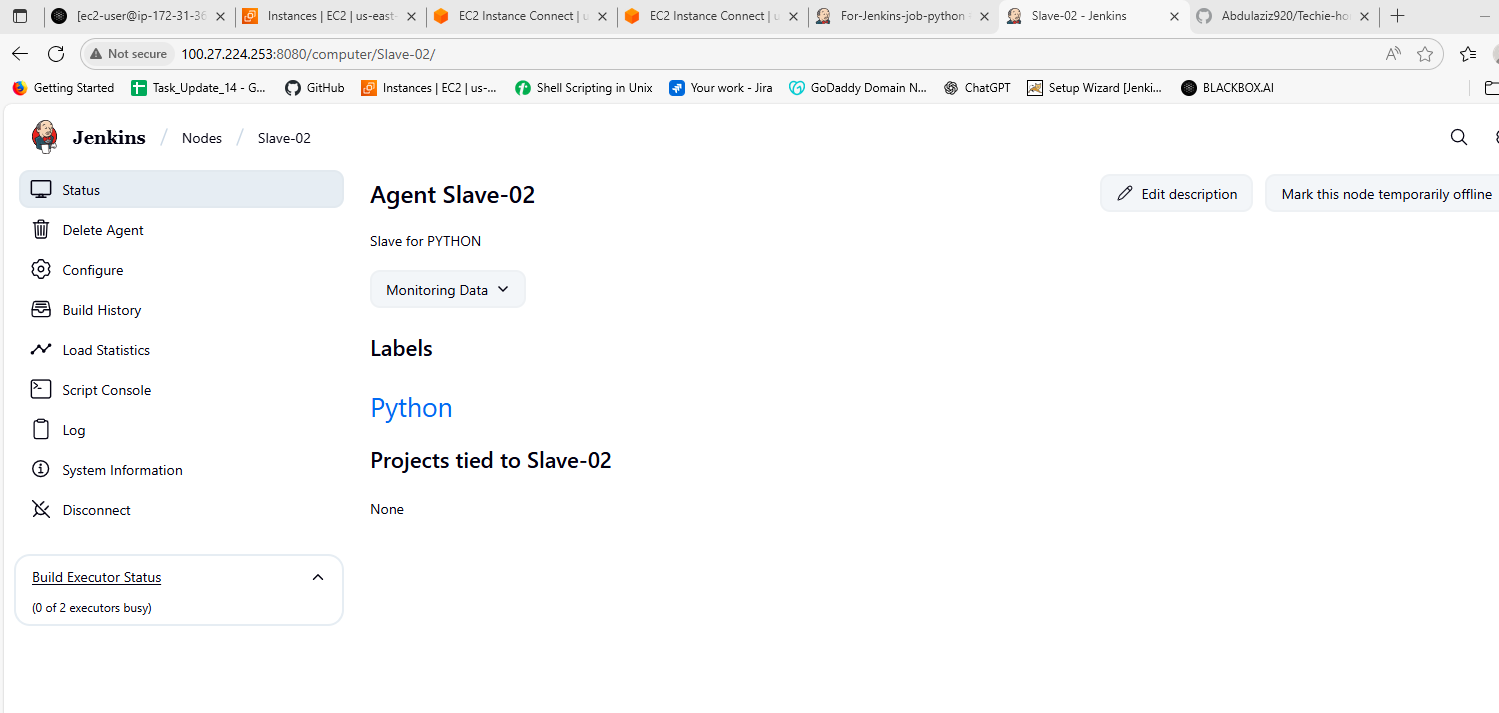


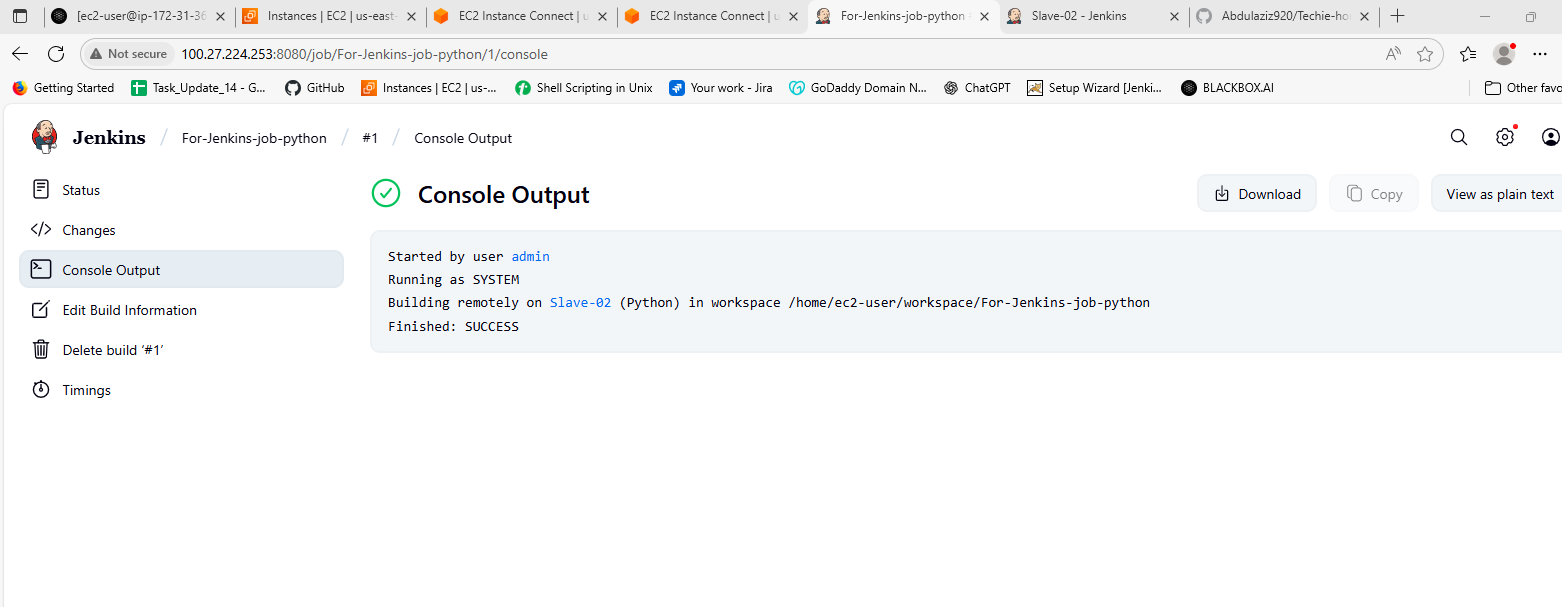
I have configure 1st Slave above then created one job and attach in label Java then it has been get success output you can see below.



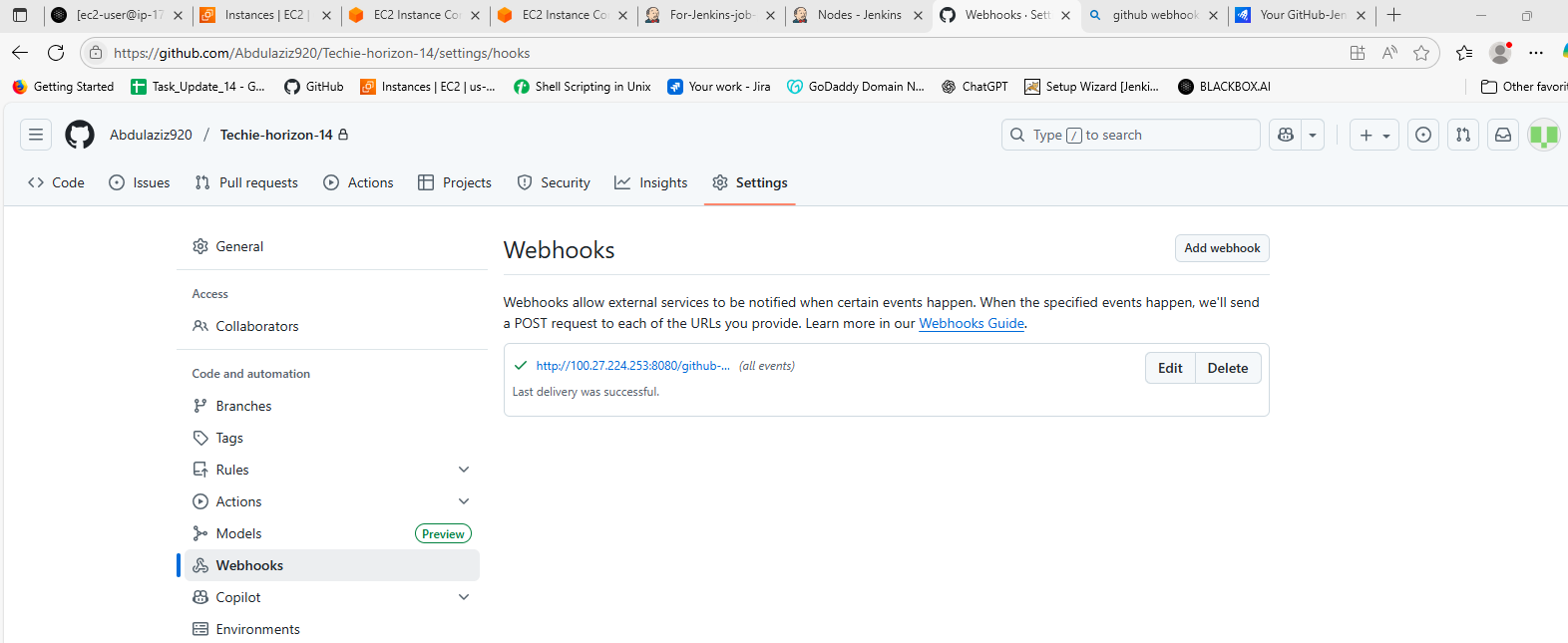
We will configure same for Slave 2.

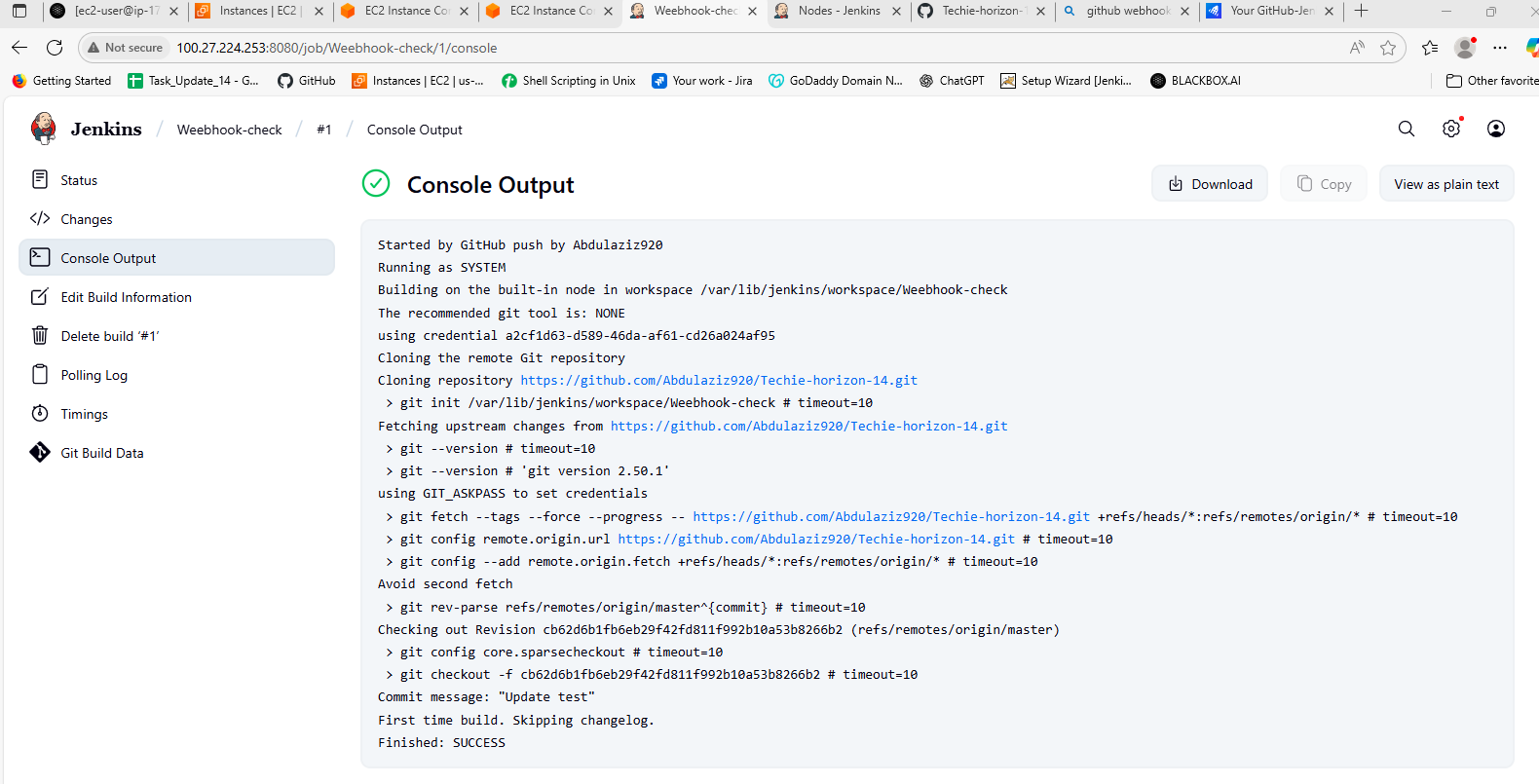
I have configure 2nd Slave below then created one job and attach in label PYTHON then it has been get success output you can see below.





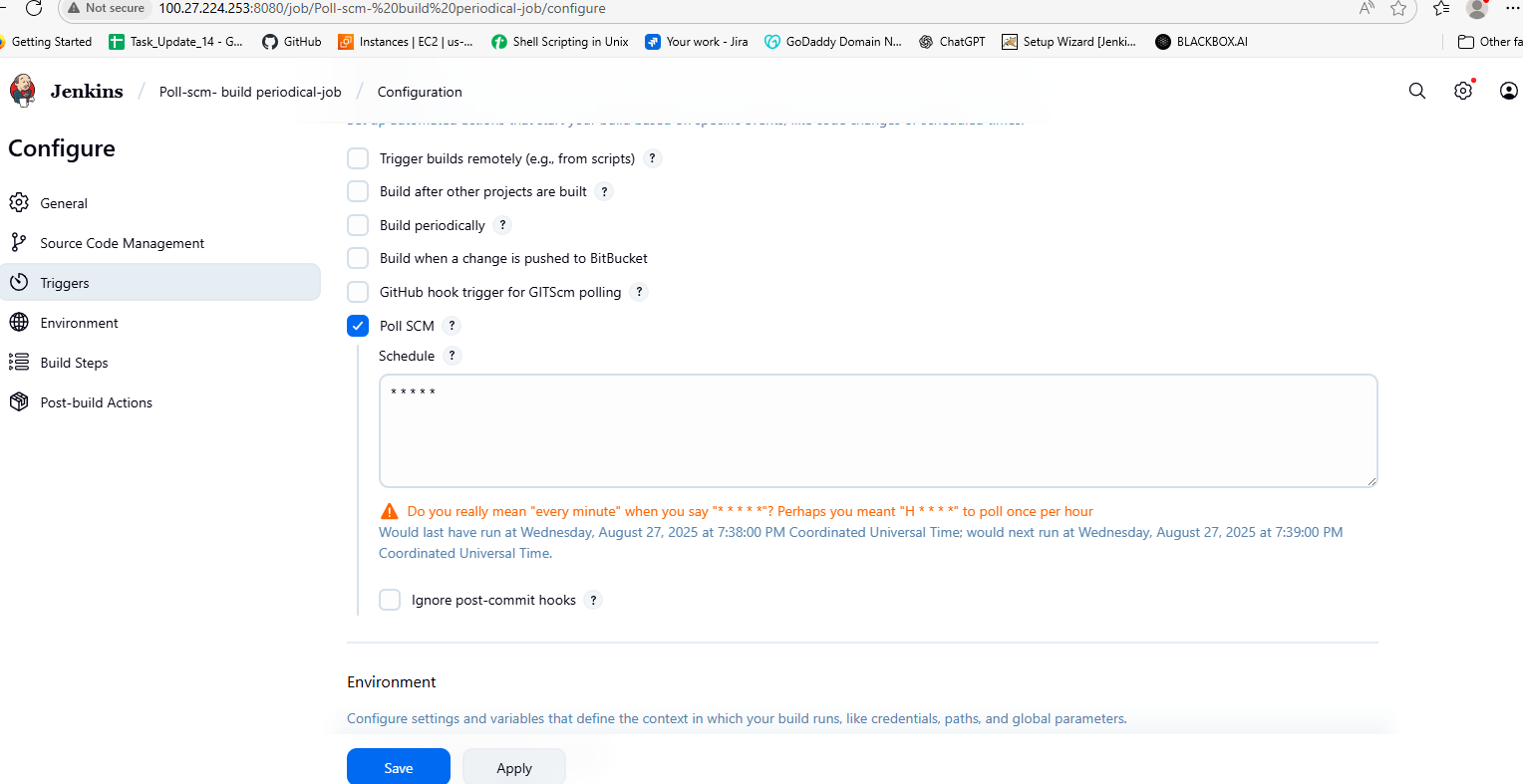
1. Configure webhooks to Jenkins job.





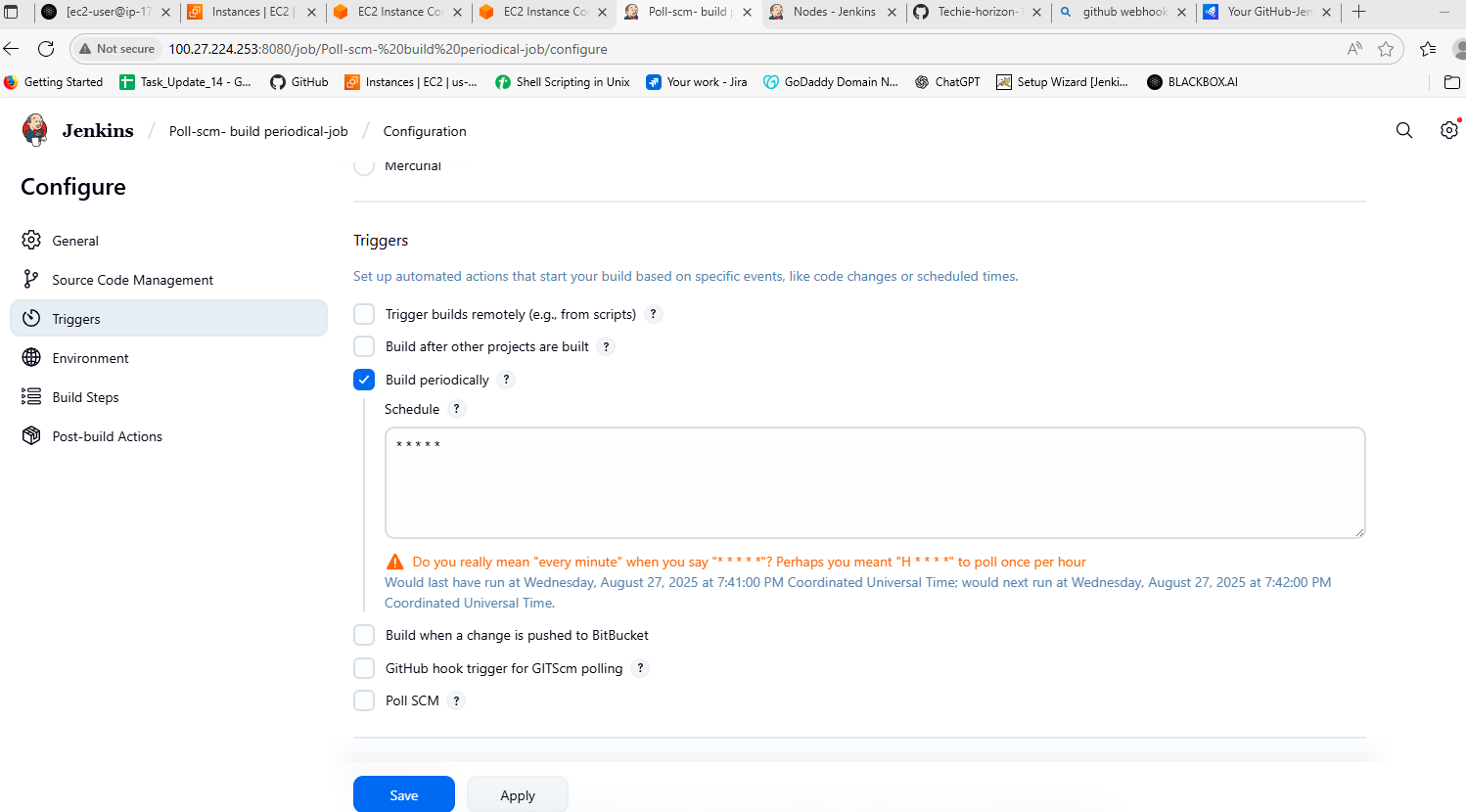
1. Configure poll scm and build periodical options in Jenkins job.

Poll SCM:

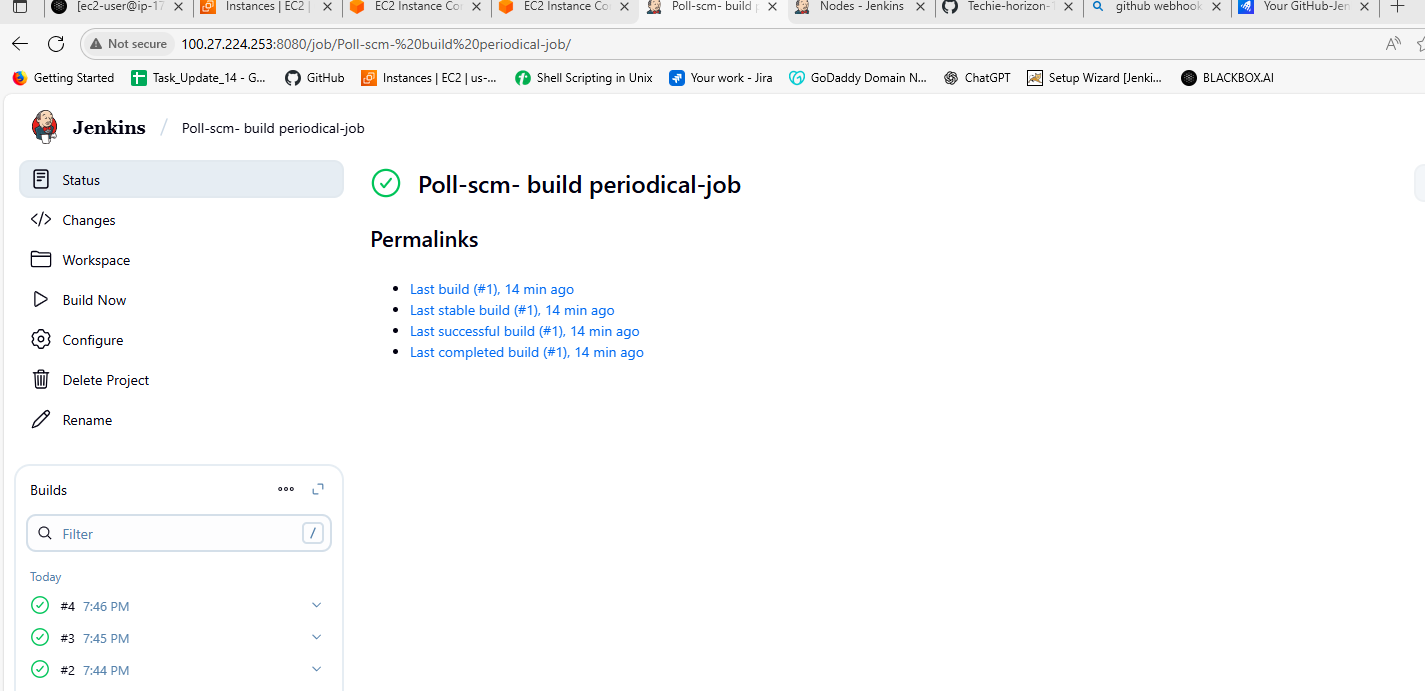


If there is any changes in the github then only it triggers if not there in not triggers happens in Poll SCM…

Build Periodical:

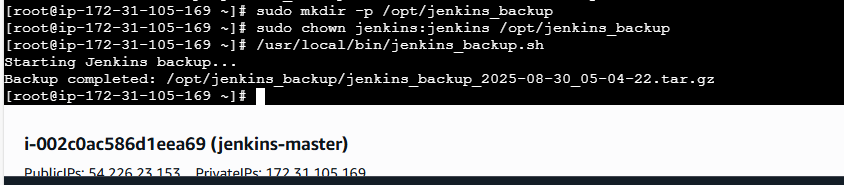


In Build Periodical it triggers based on the timings which we are giving whether the changes should br done or not the triggers should happen according to the timing..lets see below…



See in the down left side every minute it is triggering..

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



I have created a directory Jenkins\_backup

Then changed the ownership to jenkis

After that created a Jenkins\_backup.sh

And written the below script in that

#!/bin/bash

# Jenkins Backup Script

# Variables

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

BACKUP\_DIR="/opt/jenkins\_backup"

DATE=$(date +%F\_%H-%M-%S)

BACKUP\_FILE="$BACKUP\_DIR/jenkins\_backup\_$DATE.tar.gz"

# Step 1: Put Jenkins into quiet mode (optional, requires CLI setup)

# curl -X POST http://localhost:8080/quietDown

# Step 2: Create backup

echo "Starting Jenkins backup..."

tar -czf $BACKUP\_FILE \

--exclude="$JENKINS\_HOME/workspace/\*" \

--exclude="$JENKINS\_HOME/war/\*" \

-C $JENKINS\_HOME .

# Step 3: Exit quiet mode (optional)

# curl -X POST http://localhost:8080/cancelQuietDown

echo "Backup completed: $BACKUP\_FILE"

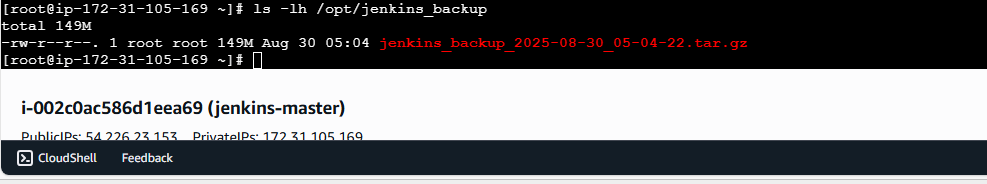
Providing the permissions to bash file

sudo chmod +x /usr/local/bin/jenkins\_backup.sh

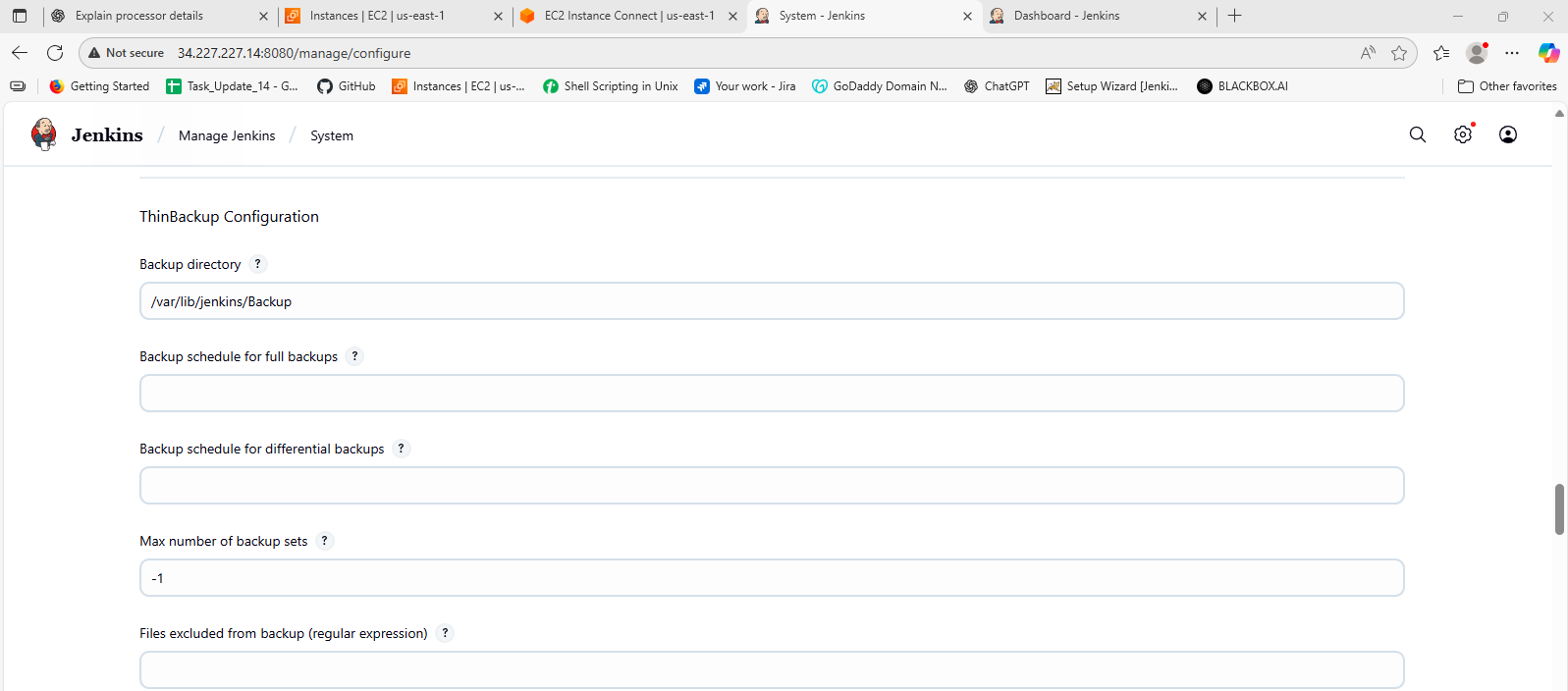
Then run the file

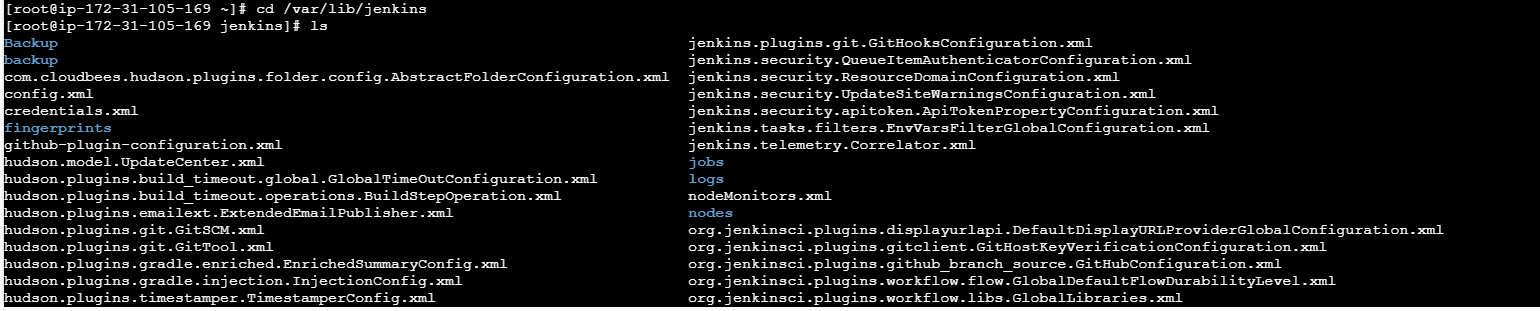
/usr/local/bin/jenkins\_backup.sh

Then the backup is created.

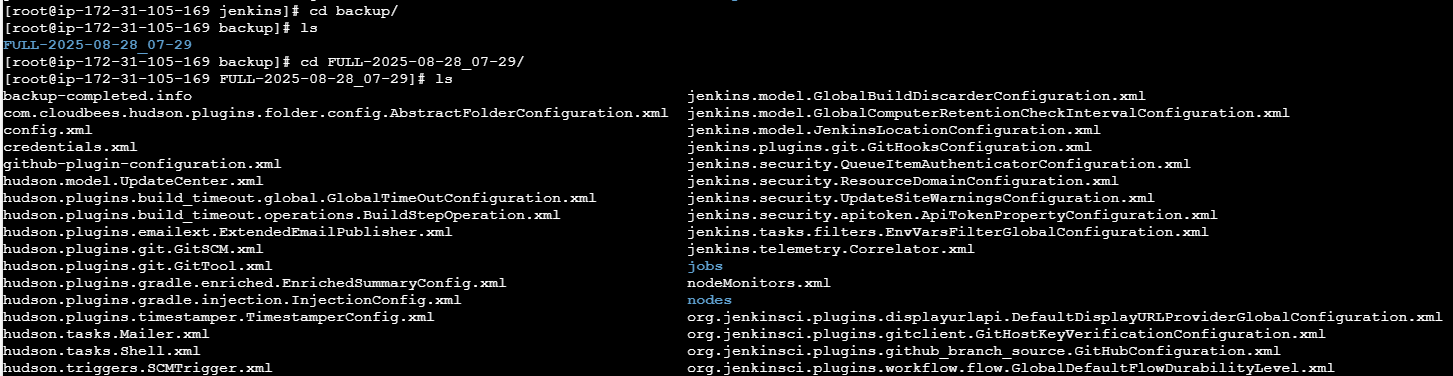


5) Take backup of Jenkins using thin backup plugin.

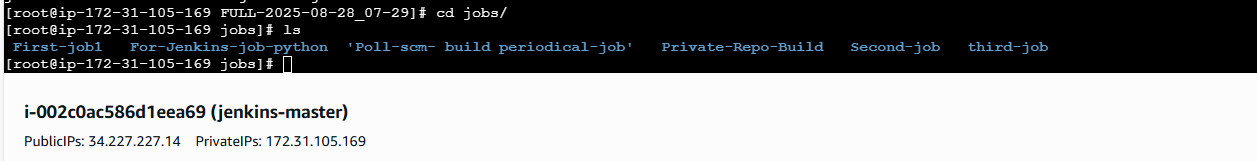




The backup has been created in above picture using the thin backup.



Here in the above picture we can see the backup in detailed.



1. Setup a new Jenkins server and dump the backup taken in task4.

We have to follow these all steps given below

**Step 1: Identify backup file on source EC2**

Check the Jenkins backup file:

ls -l /opt/jenkins\_backup/

✅ You confirmed the actual file:

jenkins\_backup\_2025-08-30\_05-04-22.tar.gz

**Step 2: Ensure you have the PEM key to access both EC2s**

* Your Aziz\_server.pem was on your **Windows machine** at:

C:\Users\user1\Downloads\Aziz\_server.pem

* This key is required to SSH into **both EC2s**.

**Step 3: Copy PEM file to source EC2**

From your **Windows machine (Git Bash)**:

scp -i /c/Users/user1/Downloads/<your-login-key-for-source-EC2>.pem /c/Users/user1/Downloads/Aziz\_server.pem ec2-user@<source-EC2-IP>:/home/ec2-user/

* <your-login-key-for-source-EC2>.pem → the PEM you use to log in to source EC2.
* This copied Aziz\_server.pem to source EC2.

**Step 4: Move PEM to root and set permissions on source EC2**

SSH into source EC2 (using your normal login key):

ssh -i /c/Users/user1/Downloads/<your-login-key-for-source-EC2>.pem ec2-user@<source-EC2-IP>

Then on the EC2:

sudo mv /home/ec2-user/Aziz\_server.pem /root/.ssh/

sudo chmod 400 /root/.ssh/Aziz\_server.pem

✅ Now the key is ready to SSH/SCP to the destination EC2.

**Step 5: Copy Jenkins backup to destination EC2**

Run **on source EC2 as root**:

scp -i /root/.ssh/Aziz\_server.pem /opt/jenkins\_backup/jenkins\_backup\_2025-08-30\_05-04-22.tar.gz ec2-user@13.220.88.248:/tmp/

✅ Backup transferred successfully (100%).

**Step 6: Verify backup on destination EC2**

SSH into destination EC2:

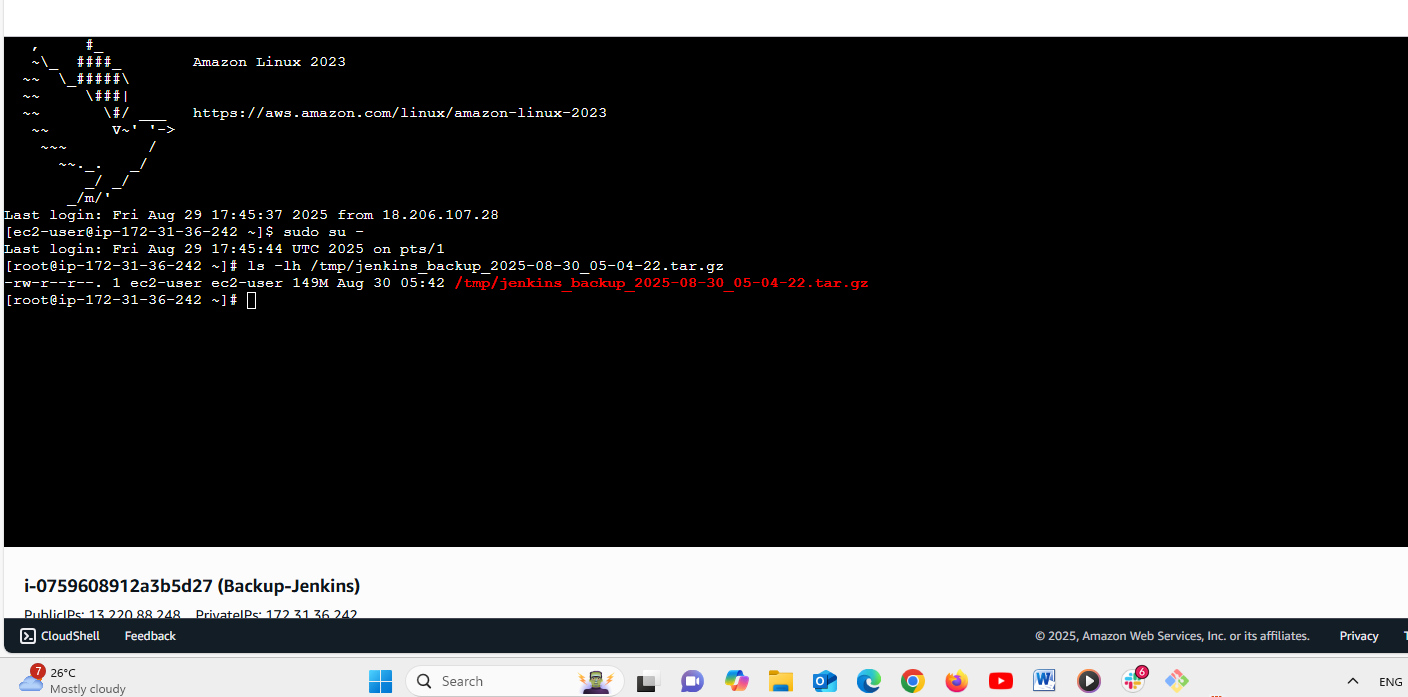
ssh -i /root/.ssh/Aziz\_server.pem ec2-user@13.220.88.248

ls -lh /tmp/jenkins\_backup\_2025-08-30\_05-04-22.tar.gz



Here in the above pic we are in the master server and taking the backup to the new server and it is successful we can see in the below..

We have just taken the backup here not resored the same.



We have checked is the backup is available or not in above we can see the new server has taken the backup.