Jenkins task for Assignment-02.

1) Configure 2 slave machines in Jenkins master.

2) Configure webhooks to Jenkins job.

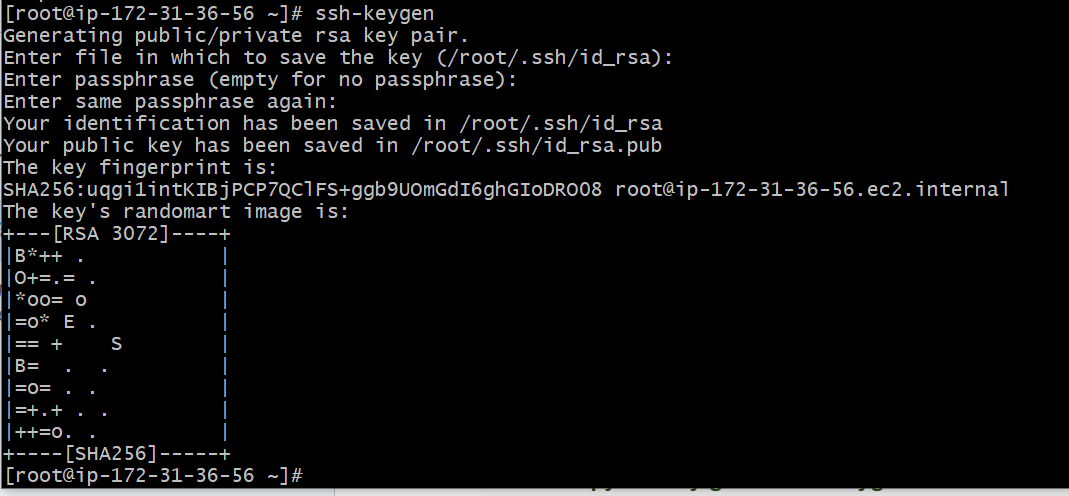
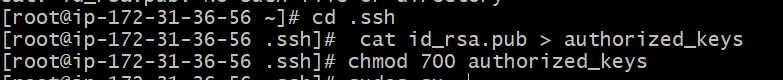
3) Configure poll scm and build periodical options in Jenkins job.

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.

**1) Configure 2 slave machines in Jenkins master.**

1. **Create or launch two instances slave01-ec2 and slave02-ec2**
2. **Install java and git on both ec2**
3. **Sudo yum install -y git**
4. **sudo dnf install -y java-17-amazon-corretto**
5. **Copy the key gen → ssh-keygen**
6. ****
7. **Cd .ssh**
8. **cat id\_rsa.pub > authorized\_keys**
9. **chmod 700 authorized\_keys**
10. ****

### 

### 

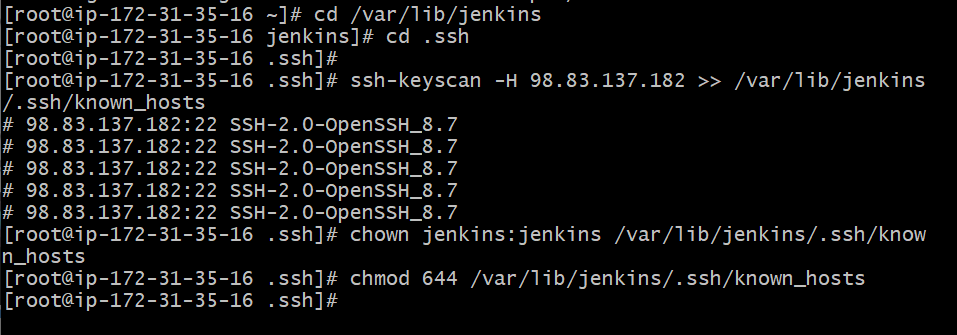
### 

### **Steps on Master Machine**

1. **Login to master machine.**
2. **Switch to root user.**
3. **Create Jenkins SSH directory:**

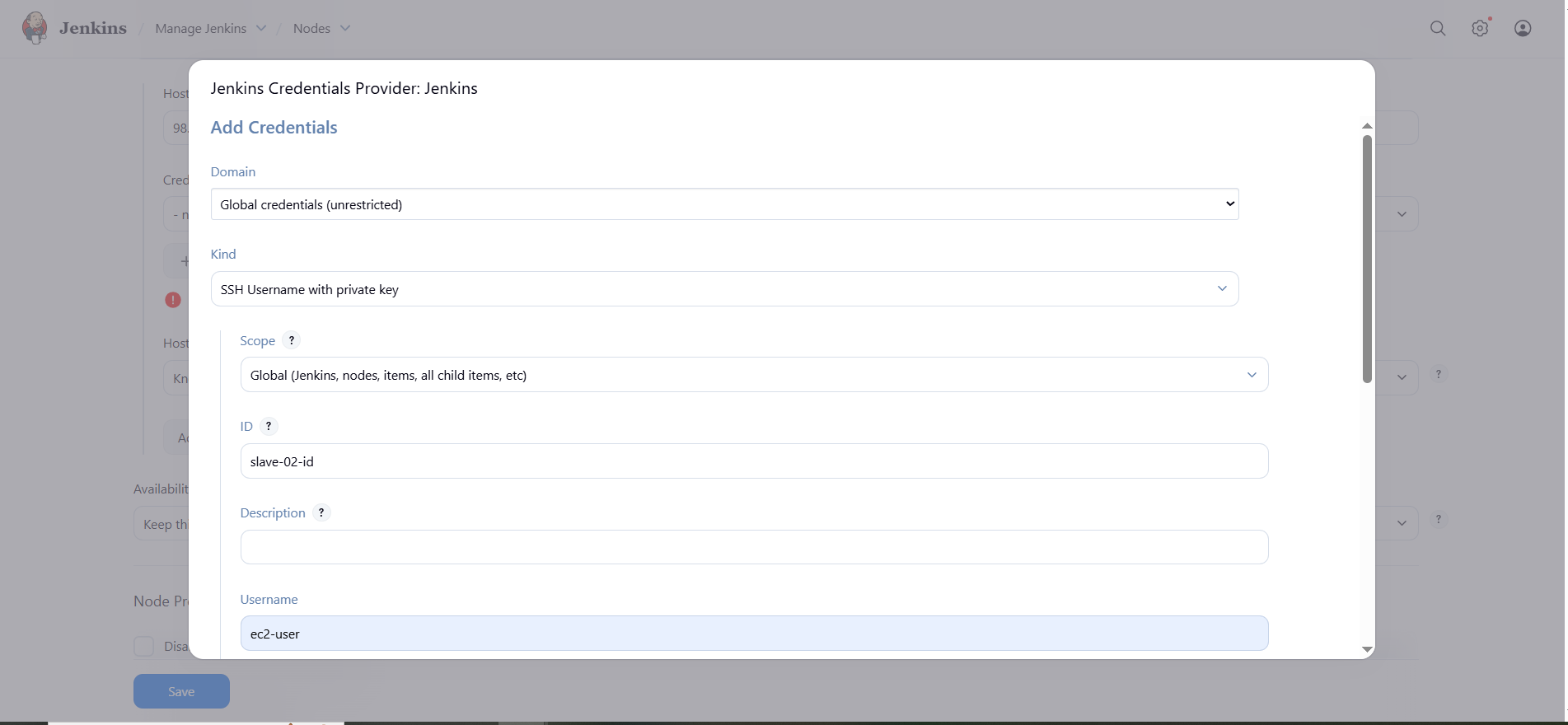
**mkdir -p /var/lib/jenkins/.ssh**

**cd /var/lib/jenkins/.ssh**

1. **ssh-keyscan -H SLAVE-NODE-PUBLIC-IP >> /var/lib/jenkins/.ssh/known\_hosts**
2. **chown jenkins:jenkins /var/lib/jenkins/.ssh/known\_hosts**
3. **chmod 644 /var/lib/jenkins/.ssh/known\_hosts**
4. ****

### **E) Create the node in Jenkins GUI**

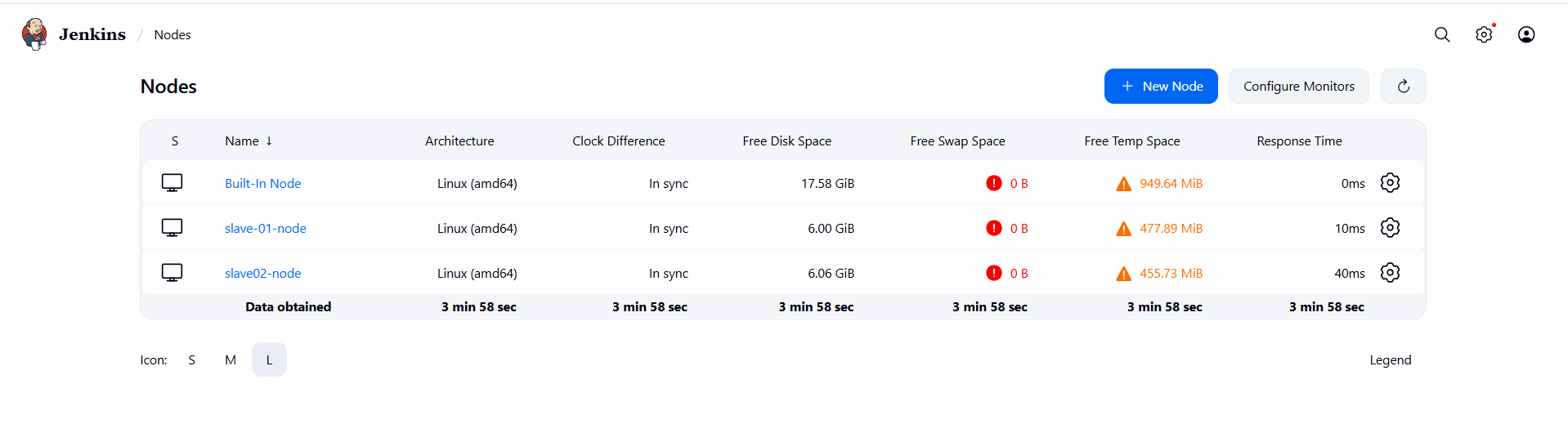
1. **Manage Jenkins → Nodes → New Node**
2. **Name: slave2 → Type: Permanent Agent**
3. **Remote root directory: /home/ec2-user**
4. **This path is taken from slave machine**
5. **Slave ec2 → cat /etc/passwd**
6. **chown ec2-user:ec2-user slave02-workspace**
7. **chmod 777 slave02-workspace**
8. **Labels: slave2 java (as you like)**
9. **Usage: “Only build jobs with label expressions” (optional)**
10. **Launch method: Launch agents via SSH**
    * **Host: <NEW\_AGENT\_IP\_OR\_DNS>**
    * **Credentials: create/select “SSH Username with private key”**
      + **Username: ec2-user**
      + **Private key: paste content of /var/lib/jenkins/.ssh/id\_rsa from the master**
    * **Host Key Verification Strategy: Known hosts file (recommended)**

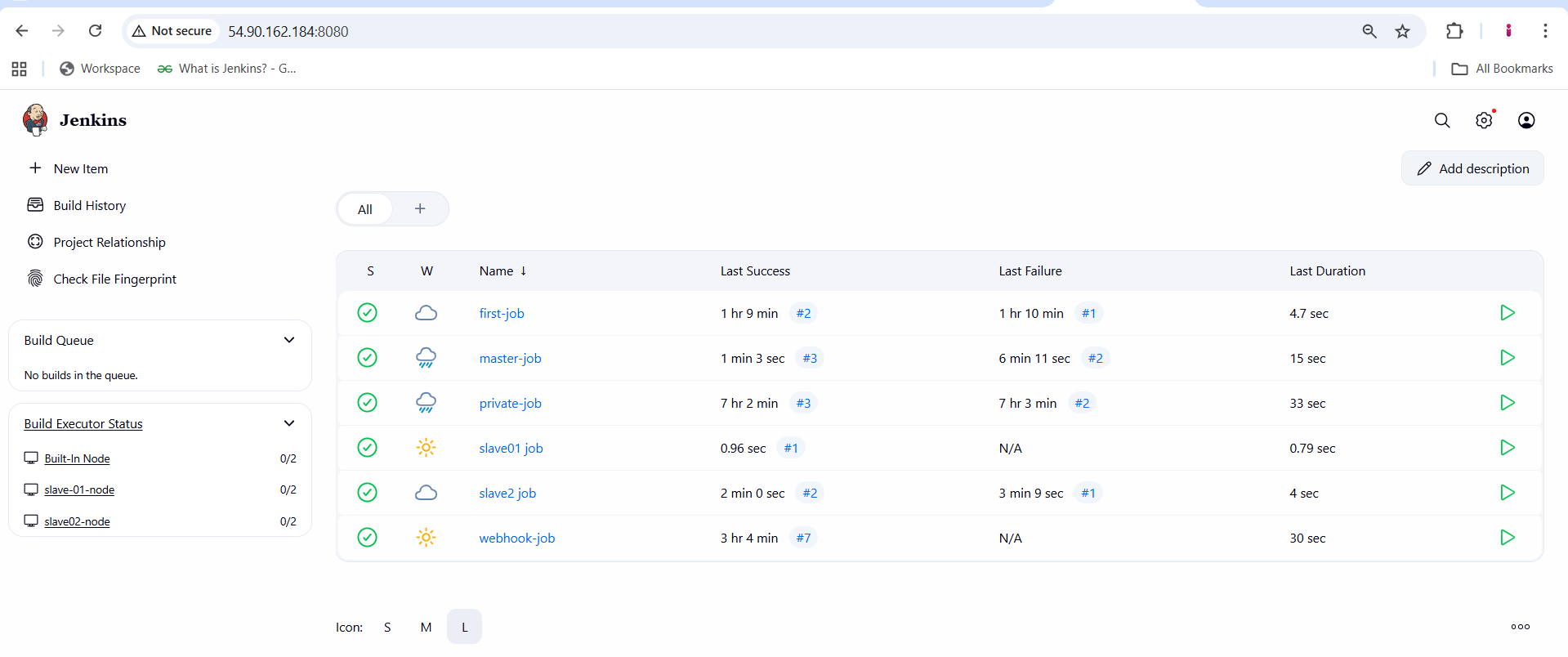
****

1. **Save → Jenkins should connect and show Online.**

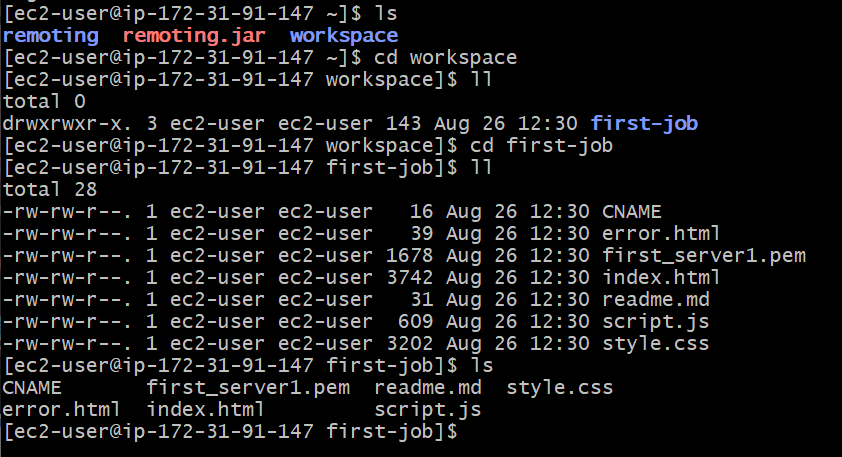
****

**Now you can see one master node and two slave nodes available.**

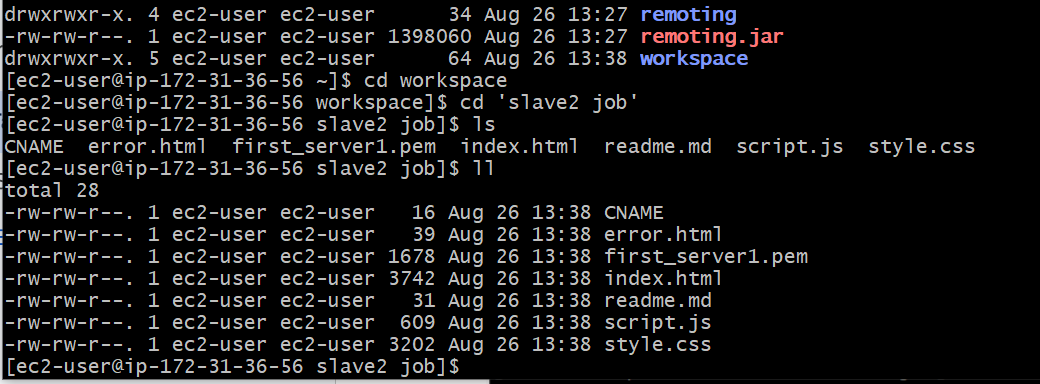
****

****

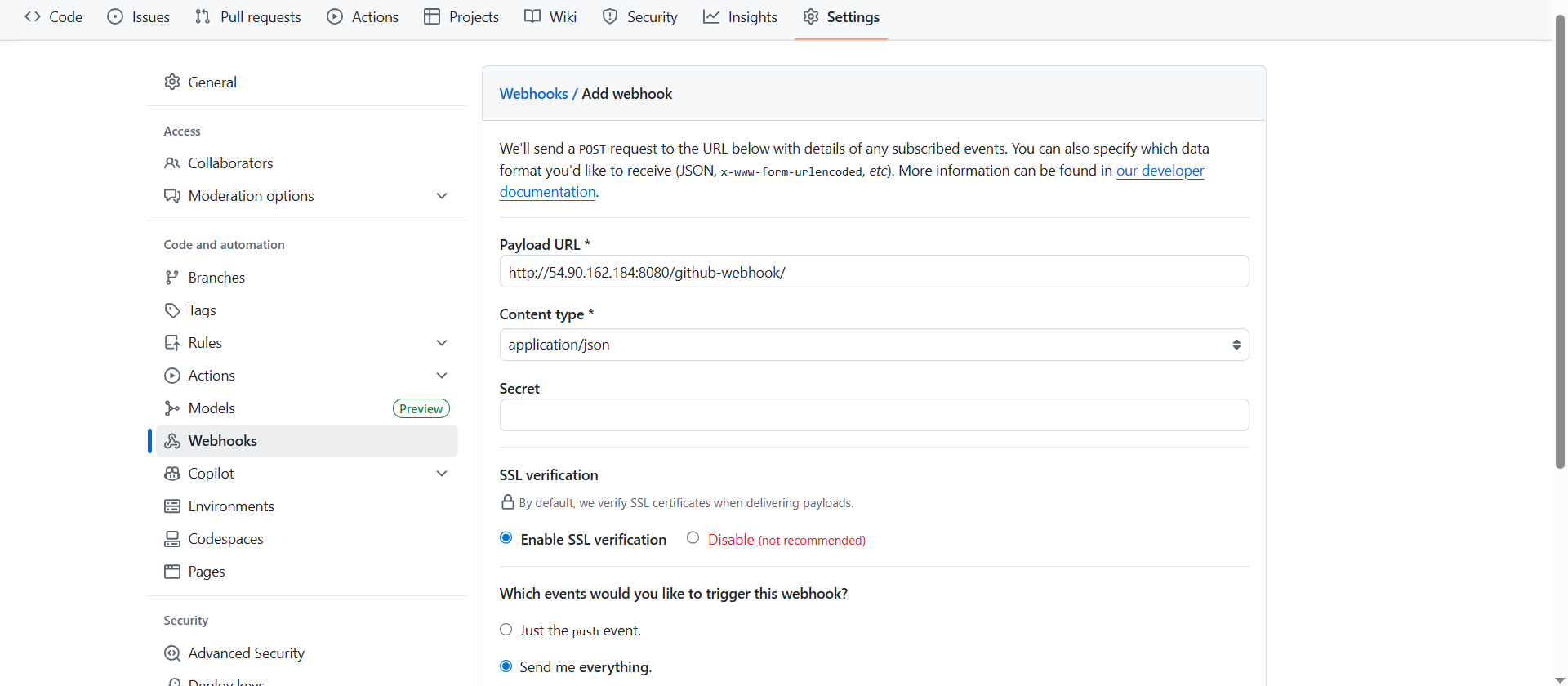
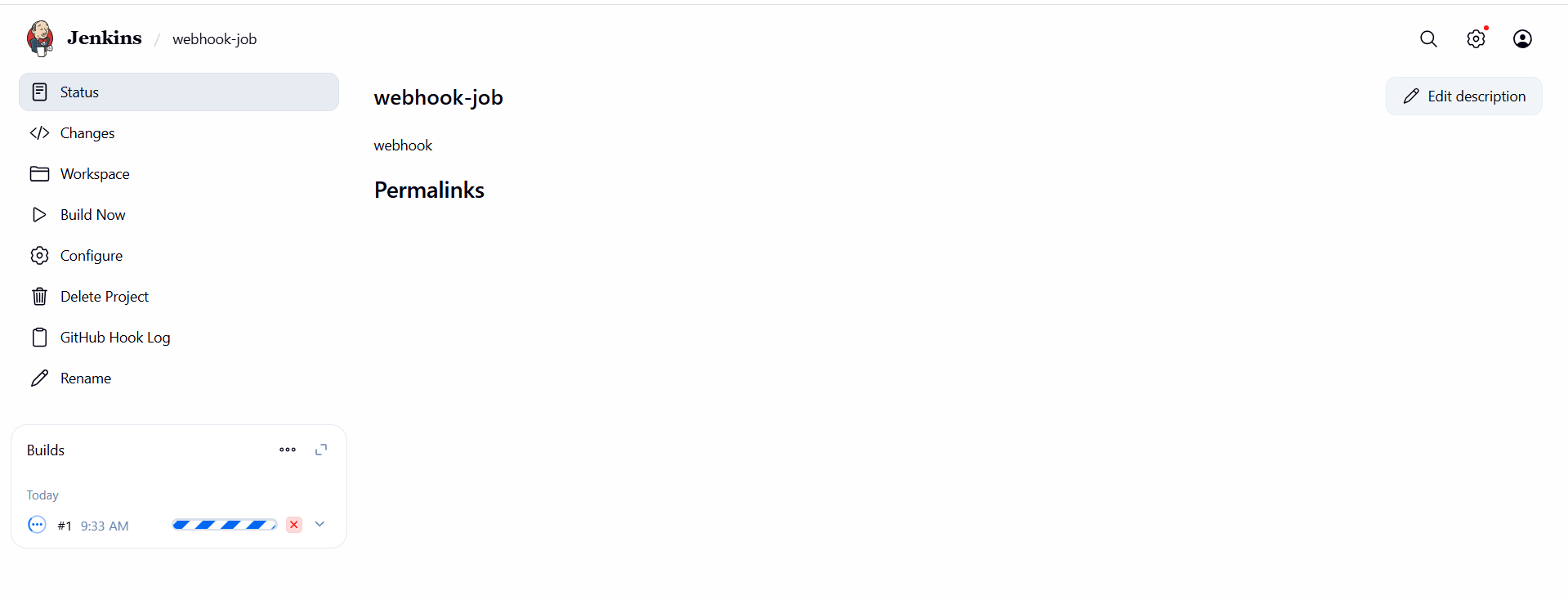
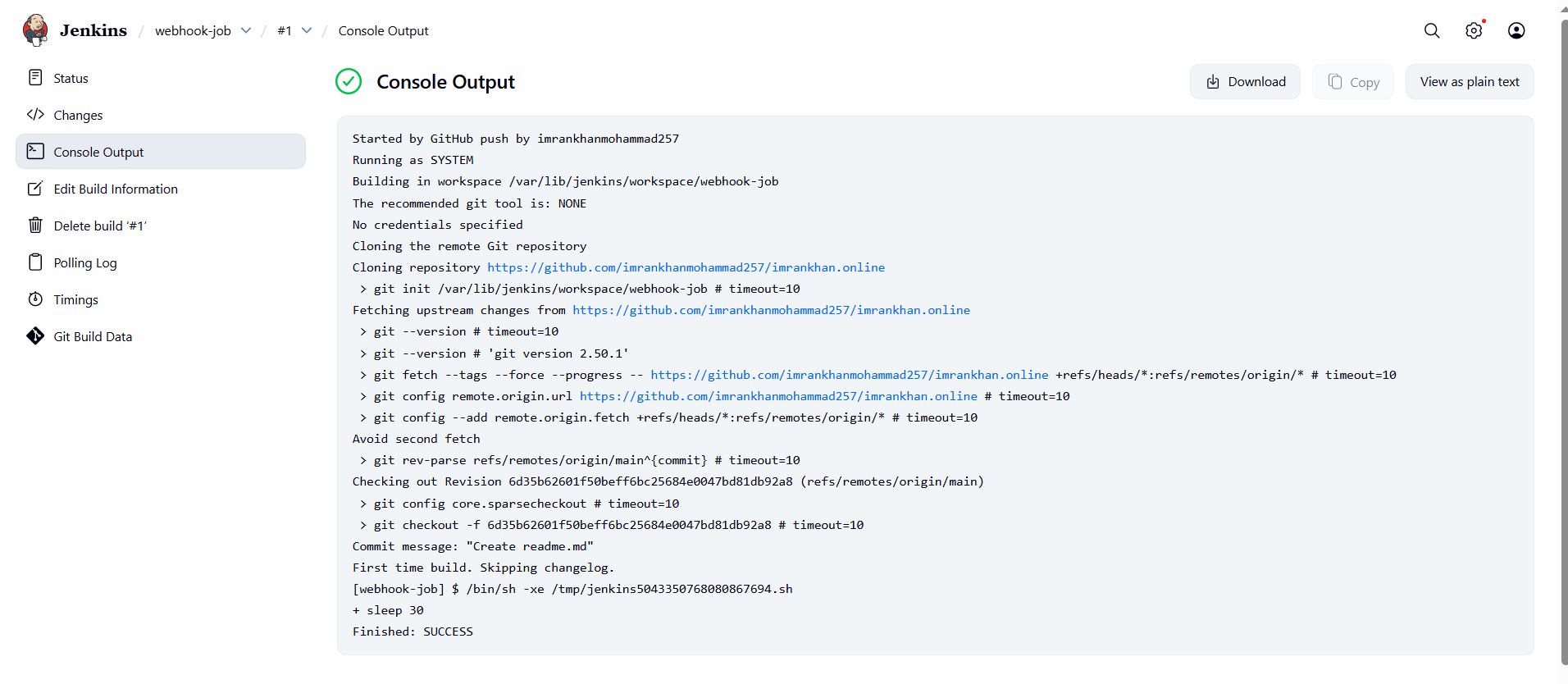
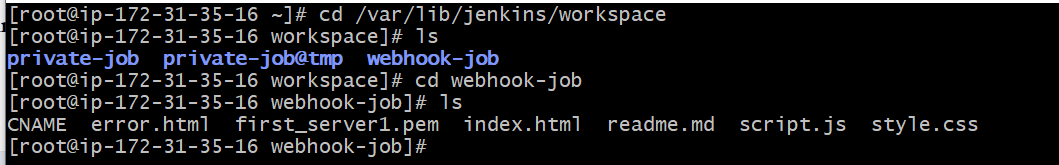
**Slave 01 EC2 console**

****

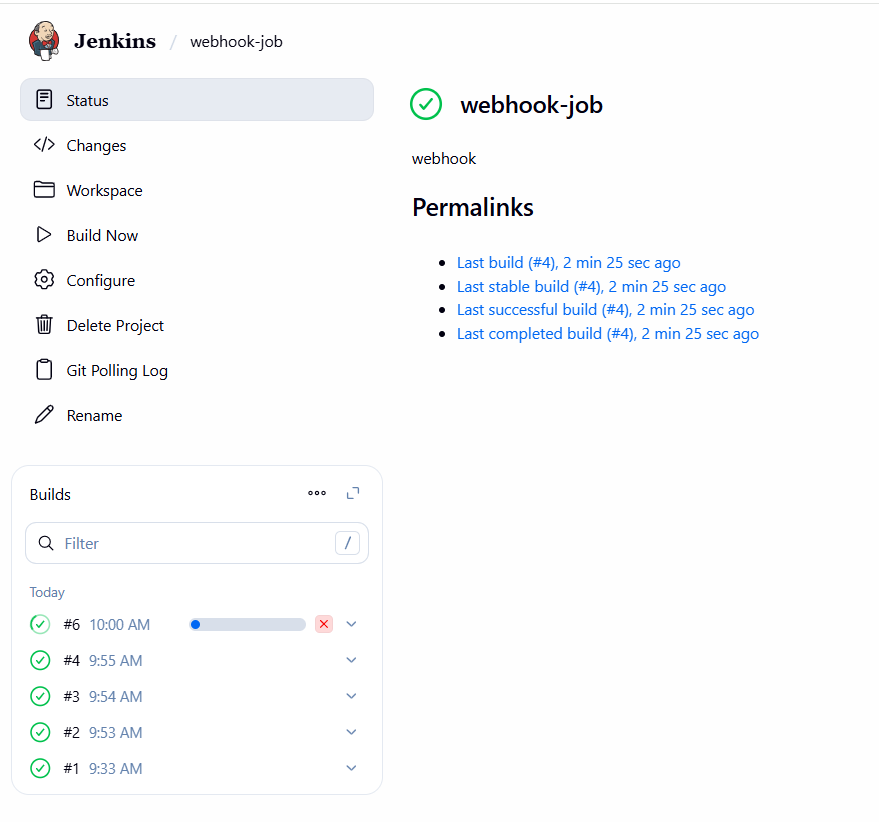
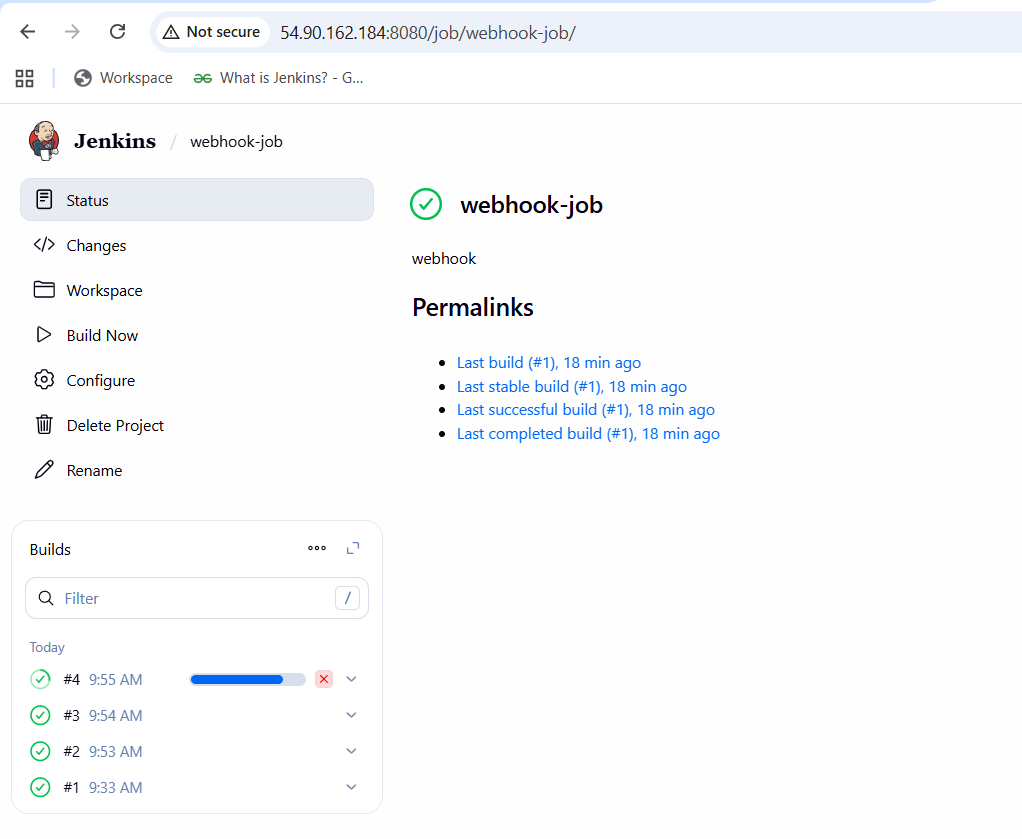
**Slave 02 ec2 console**

****

**2) Configure webhooks to Jenkins job.**

1. **open github repo → settings → webhooks → give payload (**[**http://54.90.162.184:8080/github-webhook/**](http://54.90.162.184:8080/github-webhook/)**) → content type (application/json) → select send me everything → add webhook**
2. ****
3. **Now goto your github repo and change or edit or update the any file of the source code and commit**
4. **You can see automatically webhook triggered and jenkins build the job automatically.**
5. ****
6. ****
7. **You can check files in our EC2 jenkins file.**
8. **cd /var/lib/jenkins/workspace**
9. ****

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

1. **While configure the job with poll scm and schedule with \* \* \* \* \* to the job its build every minute only if there is any changes happen in our github repo code.**
2. **Now go to your github repo and make some changes in code and check in jenkins.**
3. ****
4. **While configure the job builds periodically and give \* \* \* \* \* then irrespective of changes in our github source code it builds as per given time every minute.**
5. ****

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

1. **Jenkins configuration is stored in /var/lib/jenkins/.**
2. **sudo nano /usr/local/bin/jenkins-backup.sh**

****

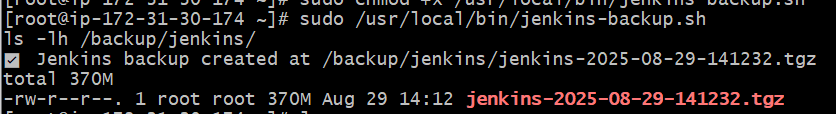
1. **sudo chmod +x /usr/local/bin/jenkins-backup.sh**
2. **Run the script now**

**sudo /usr/local/bin/jenkins-backup.sh**

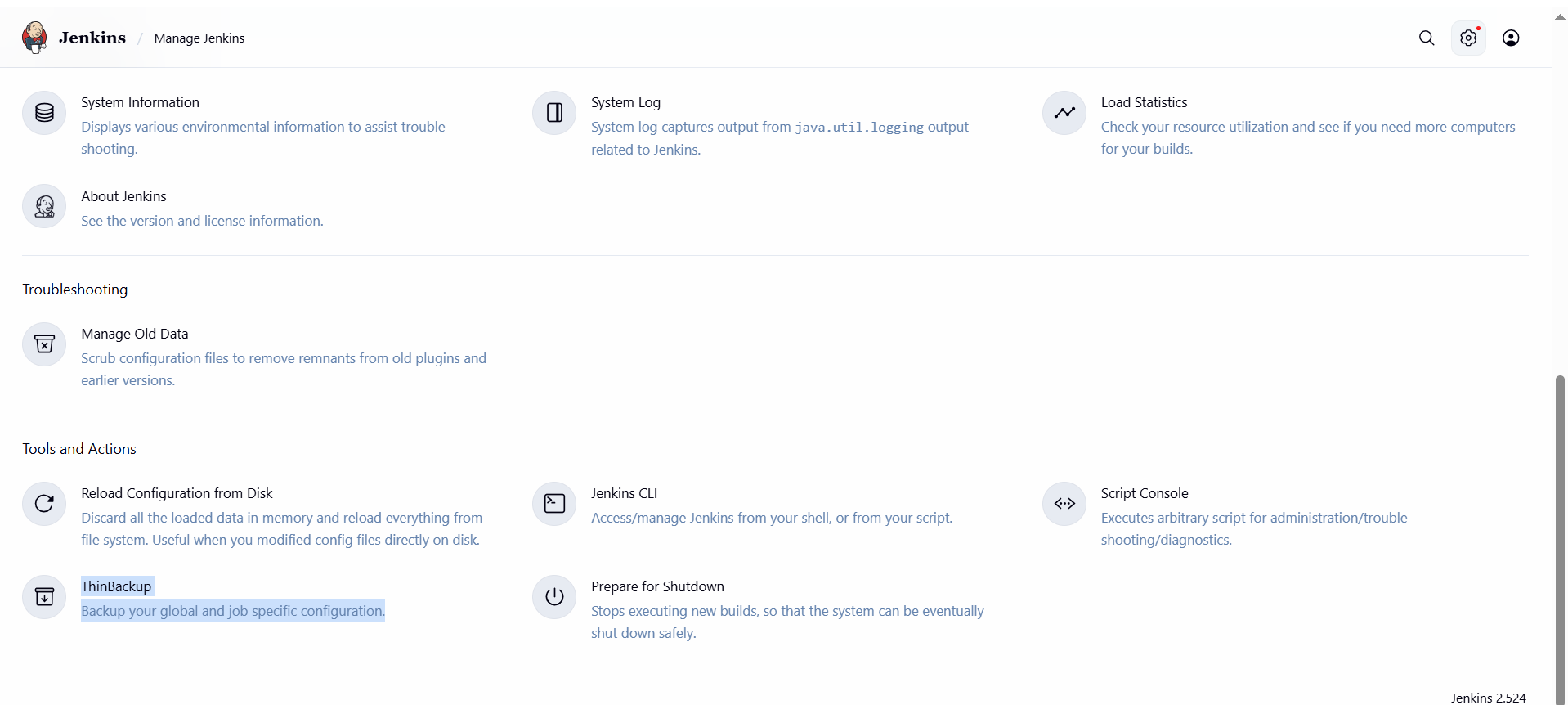
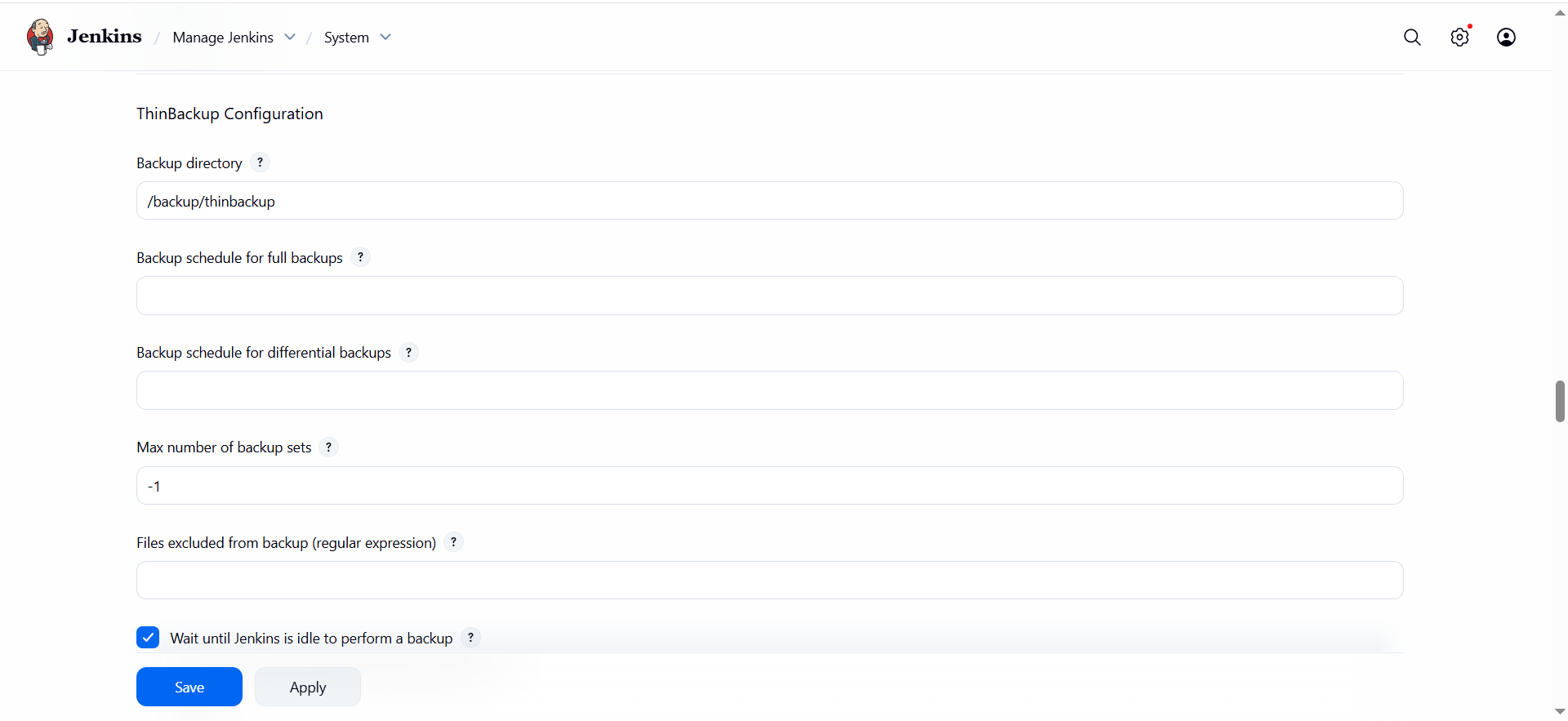
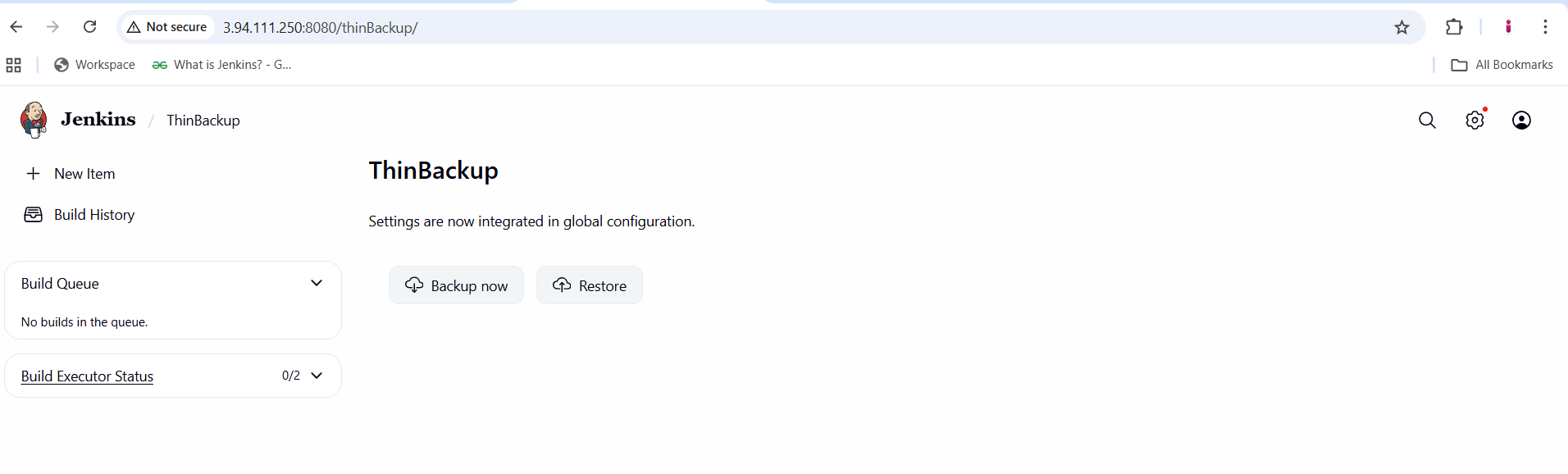
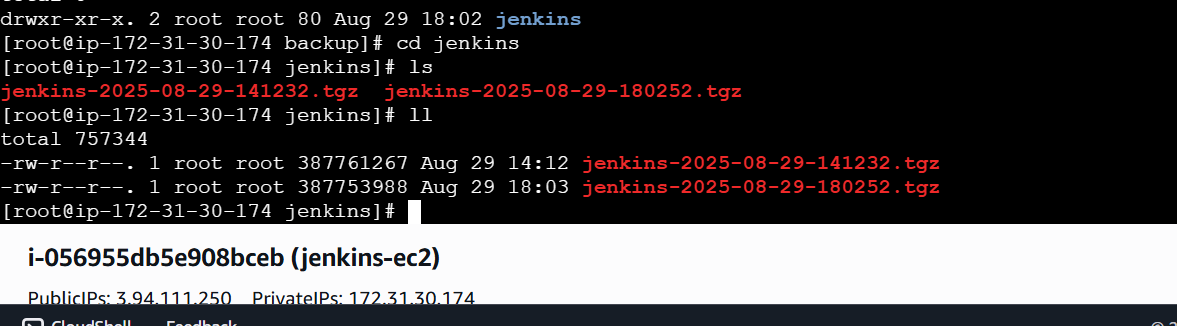
**ls -lh /backup/jenkins/**

1. **Automate with cron (daily at 2 AM):**

**echo "0 2 \* \* \* root /usr/local/bin/jenkins-backup.sh >> /var/log/jenkins-backup.log 2>&1" | sudo tee /etc/cron.d/jenkins-backup**

1. ****

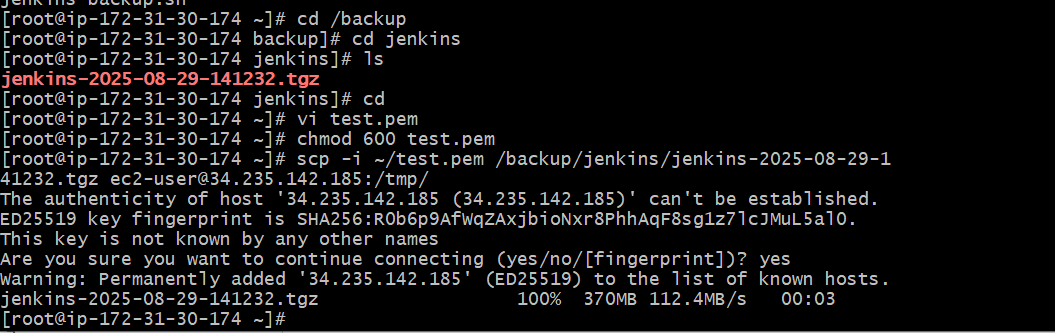
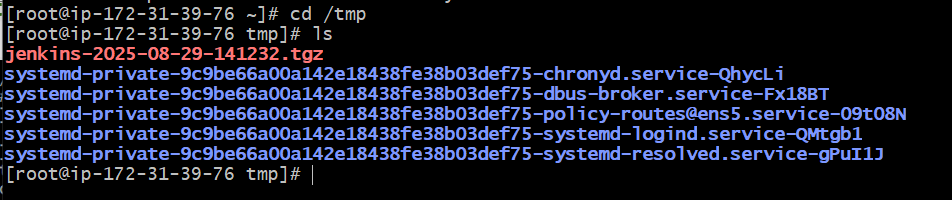
**5) Take backup of Jenkins using thinbackup plugin.**

1. **Install plugin thinbackup in jenkins**
2. **In tools and actions you get thinbackup**
3. ****
4. **Manage jenkins → system information**
5. ****
6. **After configuration then goto thinbackup and click backup now.**
7. ****
8. ****

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

1. **Created new EC2 and installed java and jenkins**
2. **In old jenkins EC2 created test.pem copied the pem key and give permissions**
3. **By scp command sending the backup old ec2 to new ec2 use public ip of new ec2..**

**scp -i ~/test.pem /backup/jenkins/jenkins-2025-08-29-141232.tgz ec2-user@34.235.142.185:/tmp/**

1. ****
2. **Now checking file in new ec2 /tmp**
3. ****
4. **Stop Jenkins before restore → sudo systemctl stop jenkins**
5. **Backup the default Jenkins folder (safety) →**

**sudo mv /var/lib/jenkins /var/lib/jenkins.bak.$(date +%F-%H%M)**

1. **Extract the backup**

**sudo tar -xvzf /tmp/jenkins-2025-08-29-141232.tgz -C /var/lib/**

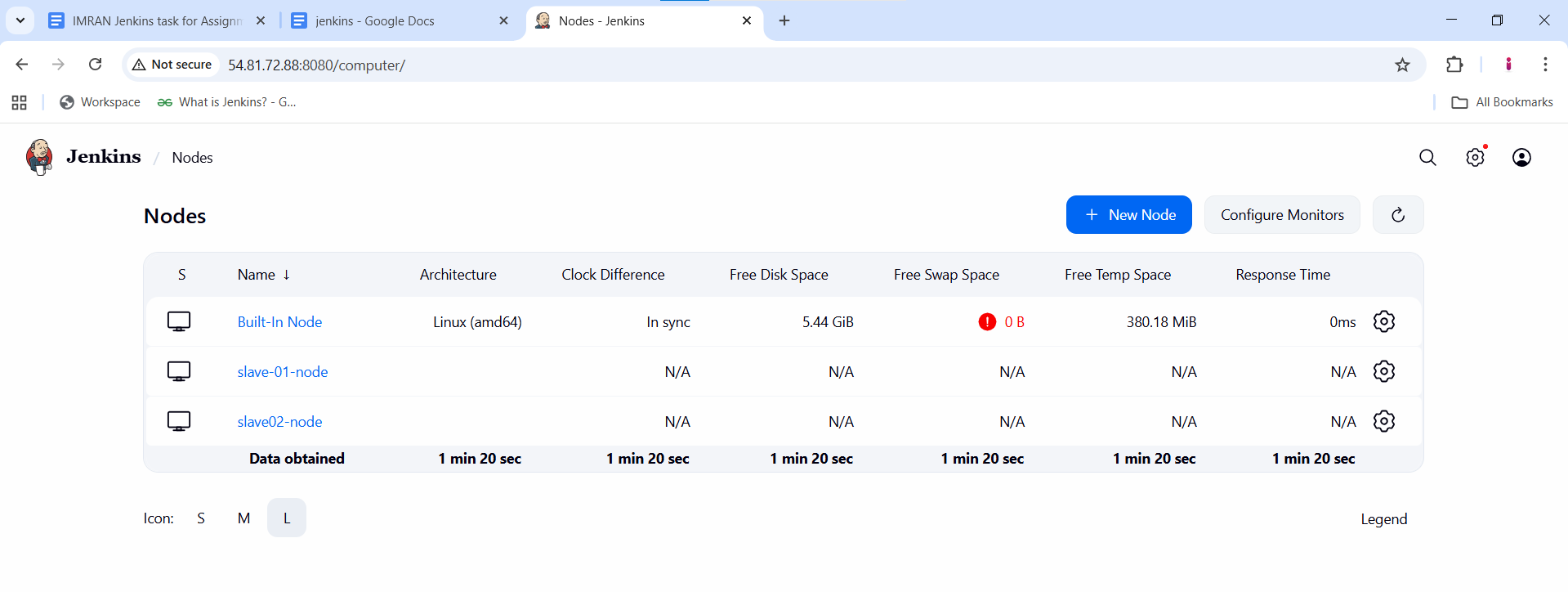
1. **Fix ownership**

**sudo chown -R jenkins:jenkins /var/lib/jenkins**

1. **Restart jenkins.**

**sudo systemctl start jenkins**

**sudo systemctl status jenkins -l**

****