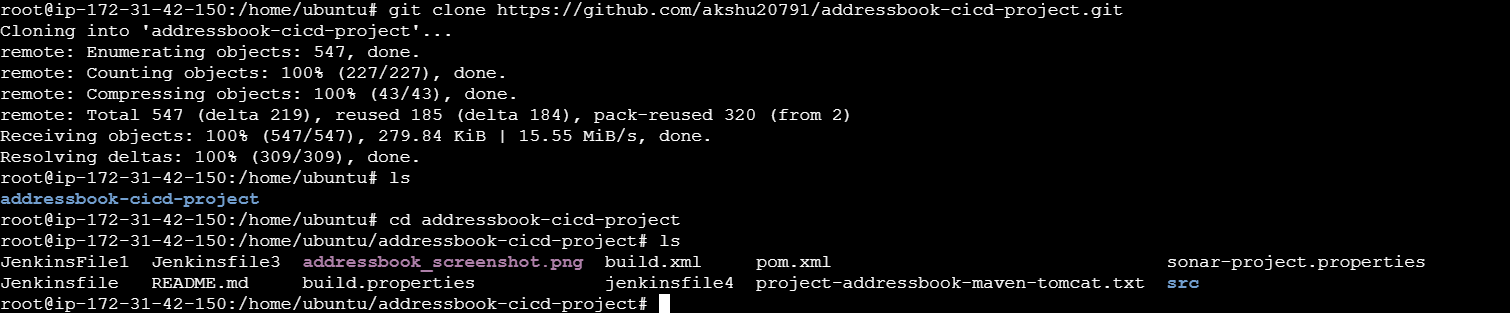
**1. Clone the GitHub repository:** [**https://github.com/akshu20791/addressbook-cicd-project**](https://github.com/akshu20791/addressbook-cicd-project)

****

****

git clone <https://github.com/akshu20791/addressbook-cicd-project.git>

**2. Create your own GitHub repository and push the cloned code to it.**

# Initialize git in the project folder

git init

# Add all files

git add .

# Commit the files

git commit -m "Initial commit"

# Remove old origin

git remote remove origin

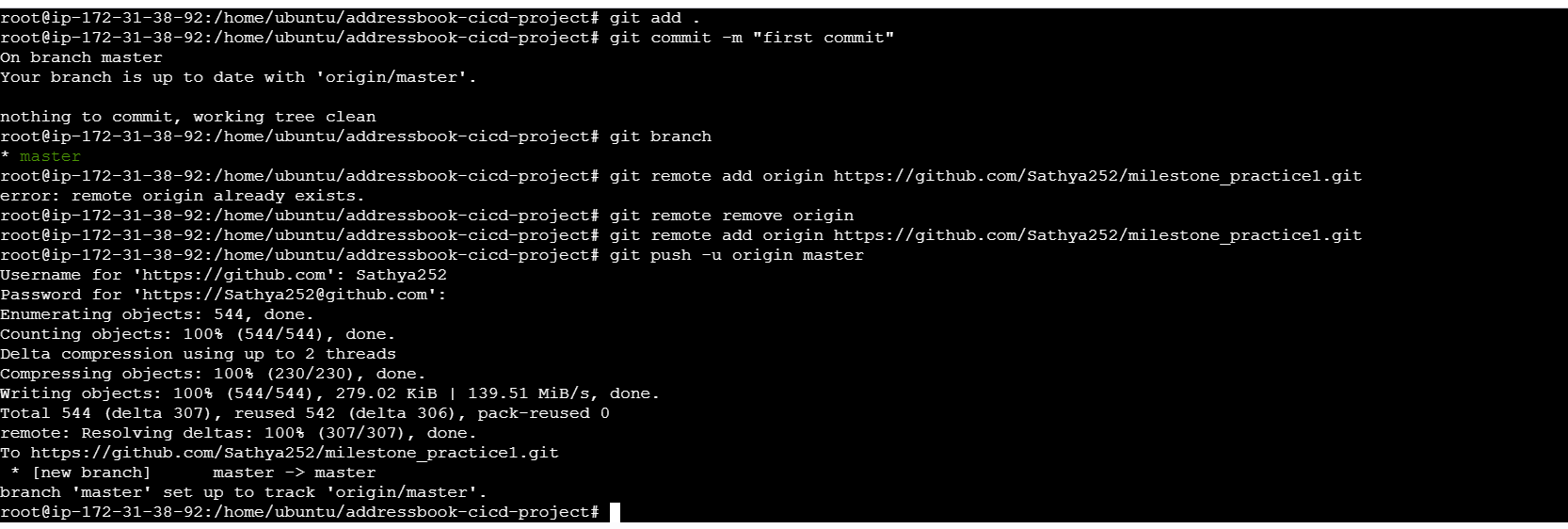
# Add your new GitHub repo URL (replace with your repo link)

git remote add origin https://github.com/<your-username>/<your-repo>.git

# Push code to main branch

git branch -M main

git push -u origin main



**3. Write a Terraform script to provision AWS infrastructure with the following:  
 - One EC2 instance as Jenkins Master  
 - One EC2 instance as Application Node  
 - Appropriate Security Groups for SSH, Jenkins, and Tomcat access**

**4. Install Ansible on the EC2 instances**.



wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(grep -oP '(?<=UBUNTU\_CODENAME=).\*' /etc/os-release || lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt update && sudo apt install terraform

  
provider "aws" {

region = "us-west-2"

access\_key = "my-access-key"

secret\_key = "my-secret-key"

}

  
resource "aws\_instance" "web" {

count = 2

ami = "ami-0f918f7e67a3323f0" **# change it**

instance\_type = "t3.micro" **#change it**

security\_groups = ["myweb"] **# change it**

tags = {

Name = "HelloWorld"**#change it**

}

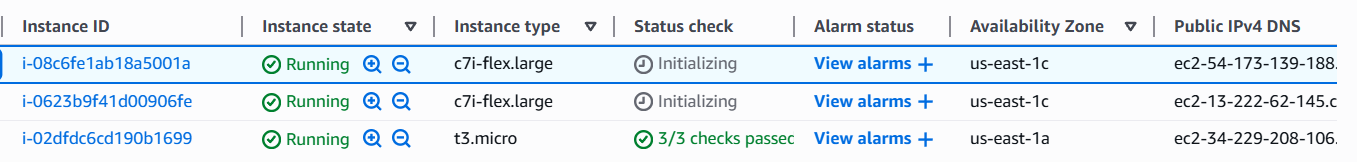
}

terraform init

terraform plan

terraform apply -auto-approve





**5. Install Jenkins on the Jenkins Master EC2 instance.**

Complete the ansible installization process using master and node instances



**Install Jenkins**

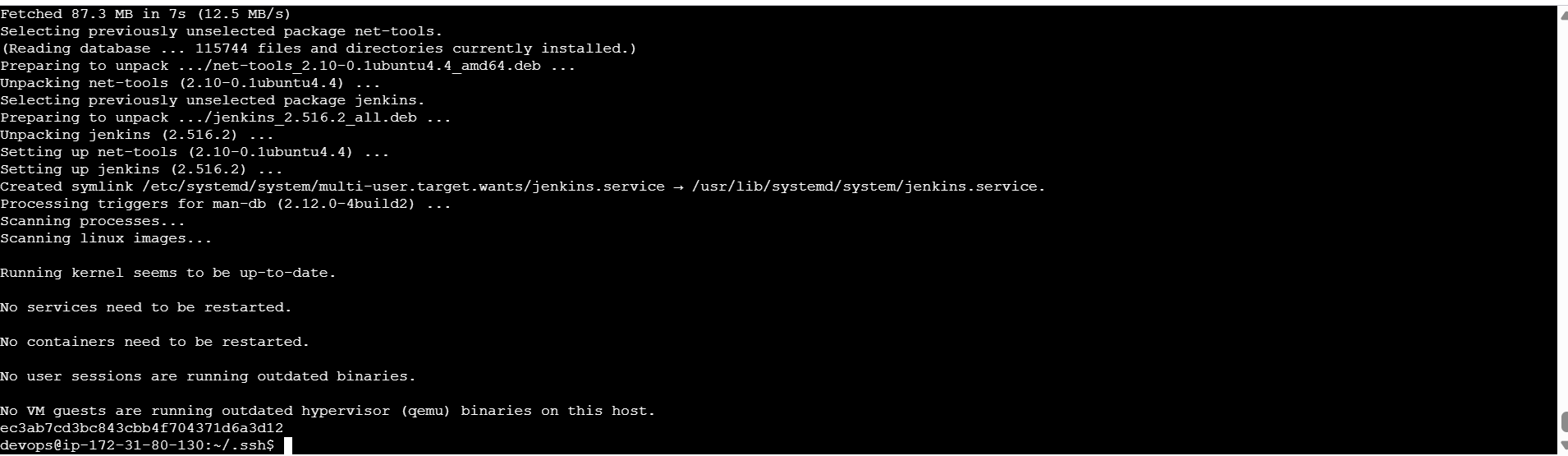
sudo apt update -y

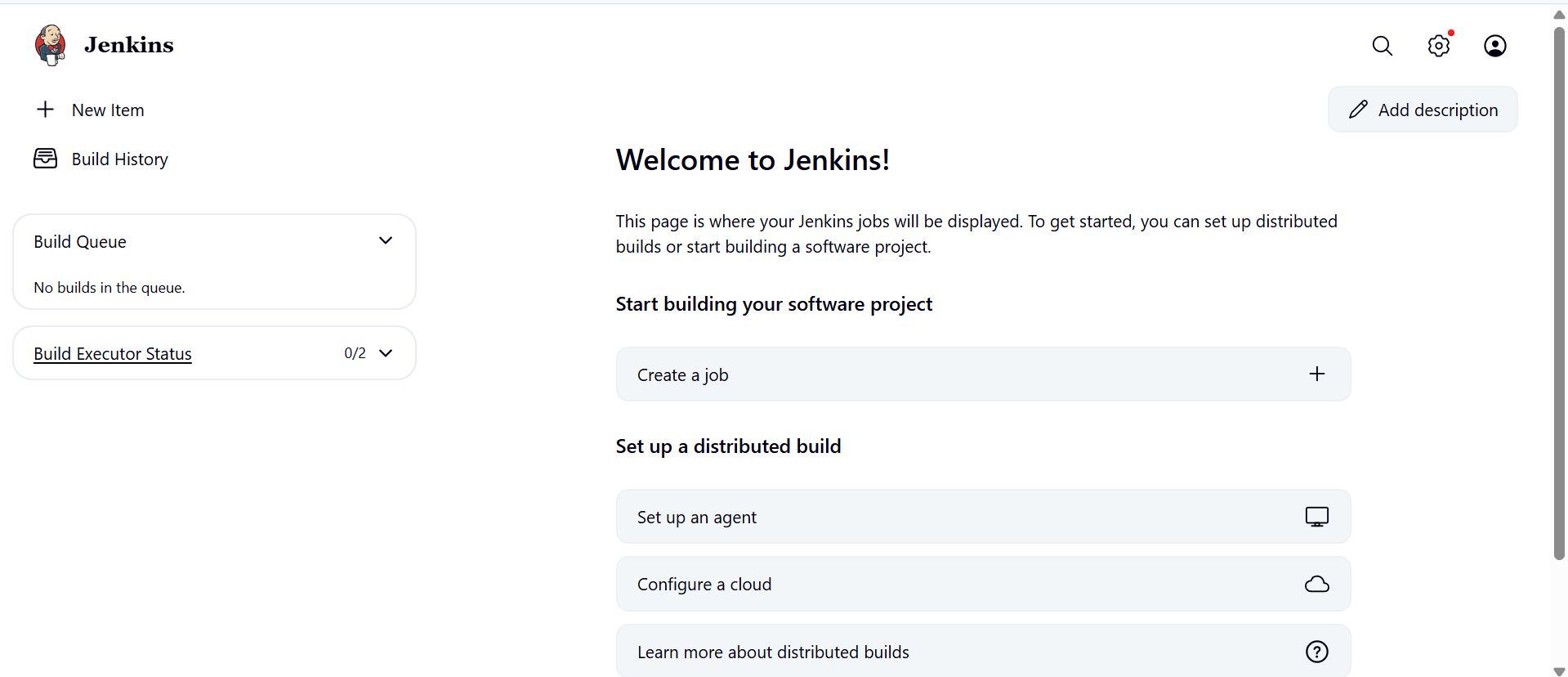
sudo apt install openjdk-17-jdk -y

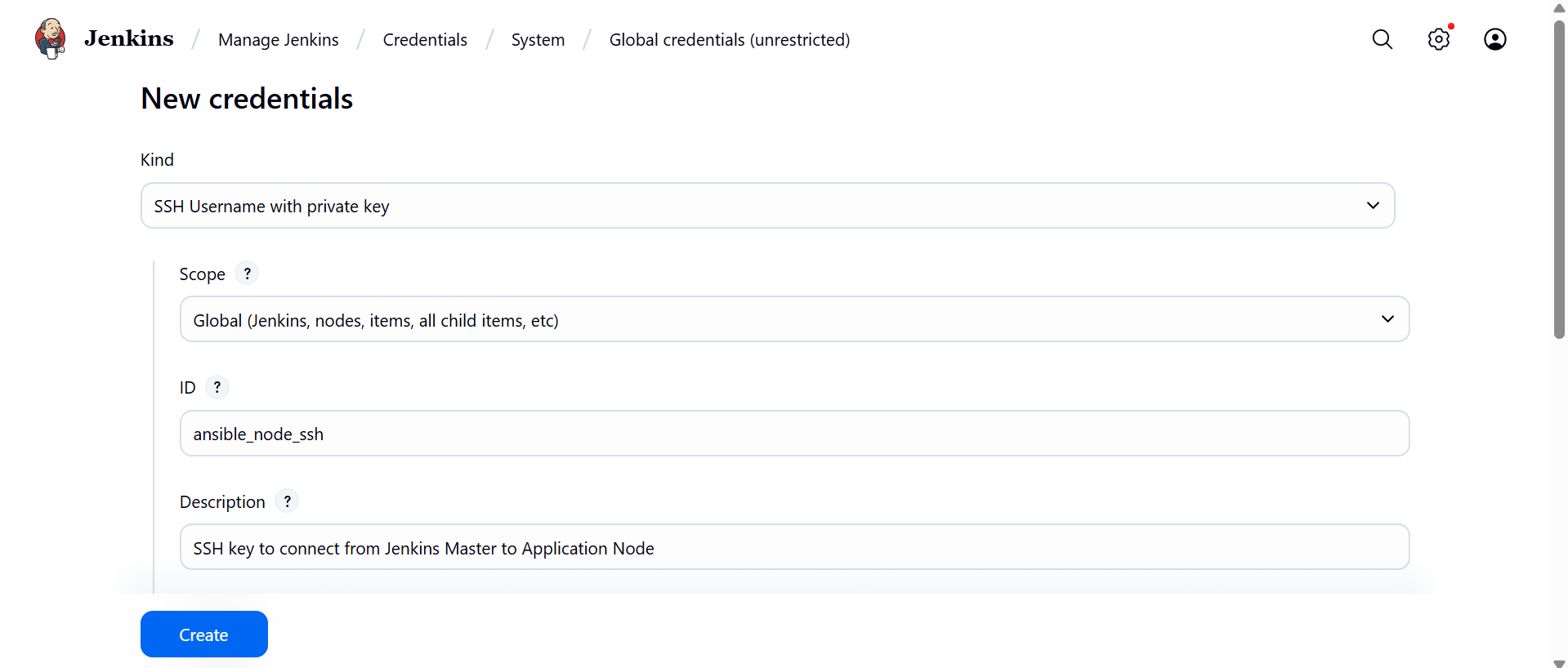
wget https://github.com/Sathya252/Deployment-script/raw/main/jenkins.sh

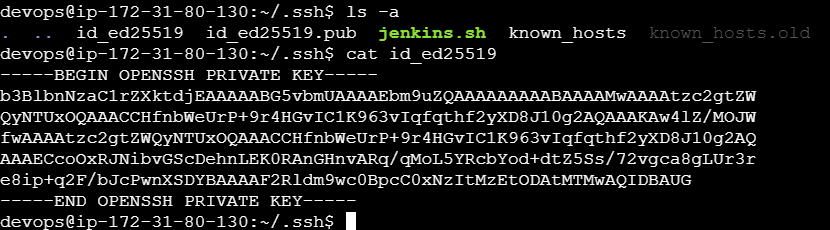
chmod +x jenkins.sh

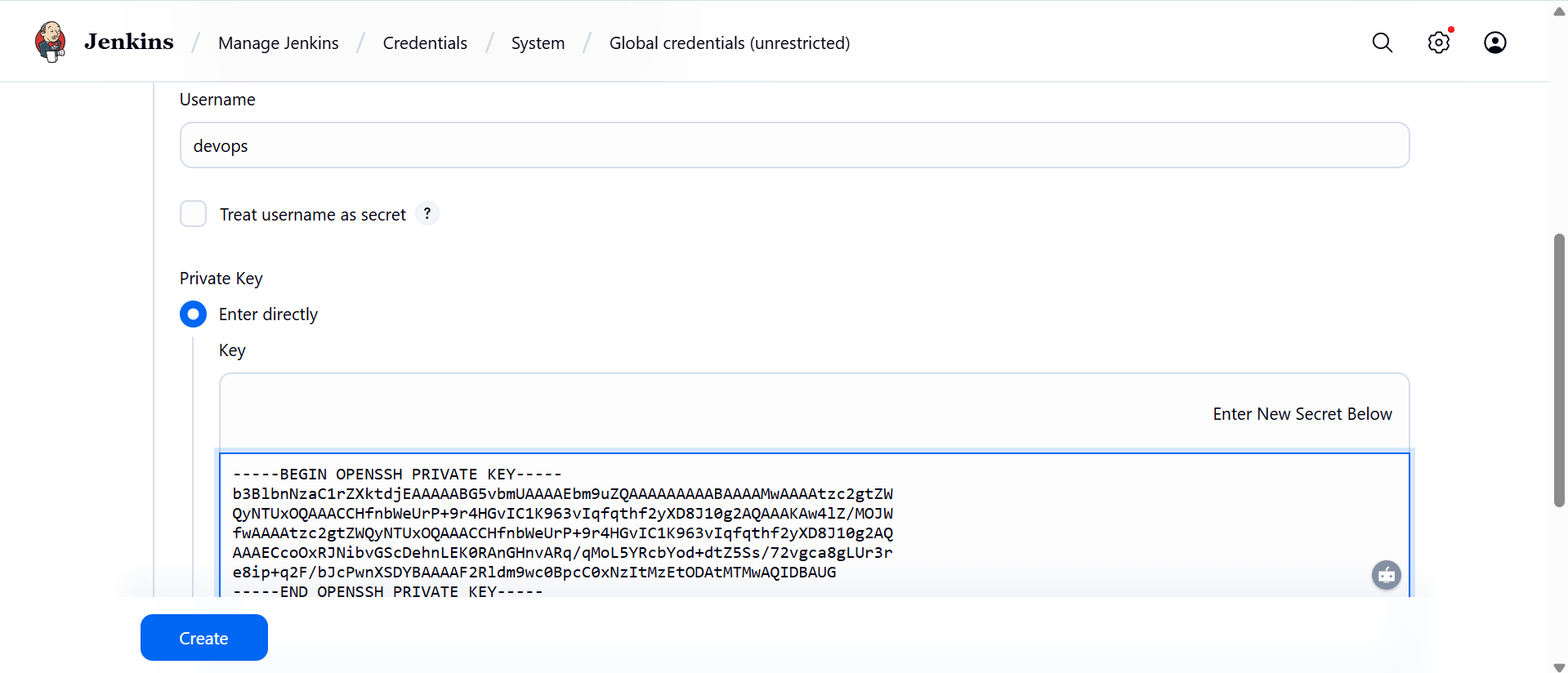
sudo ./[jenkins.sh](http://jenkins.sh)

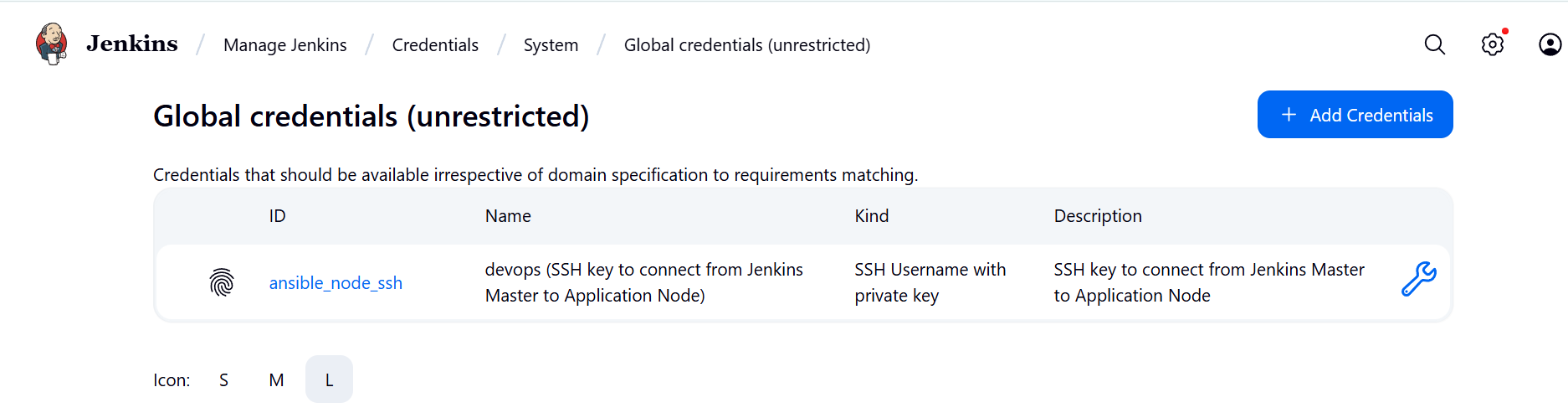
  
  
Open jenkins : <Master Public id>:8080





For Privite key go to EC2 Master Instance  






On your **master EC2**:

mkdir addressbook\_repo

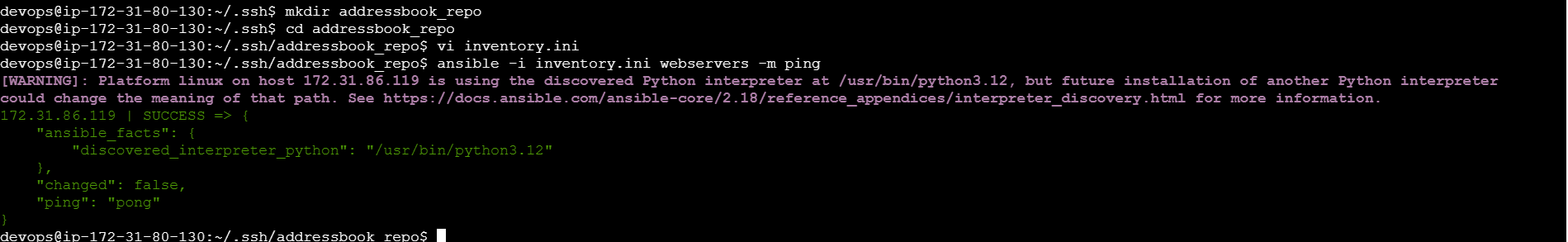
cd addressbook\_repo

**Inventory file content:-** nodes private ip is placed in this file

vi inventory.ini  
[webservers]

<private-id> ansible\_user=devops

ansible -i inventory.ini webservers -m ping



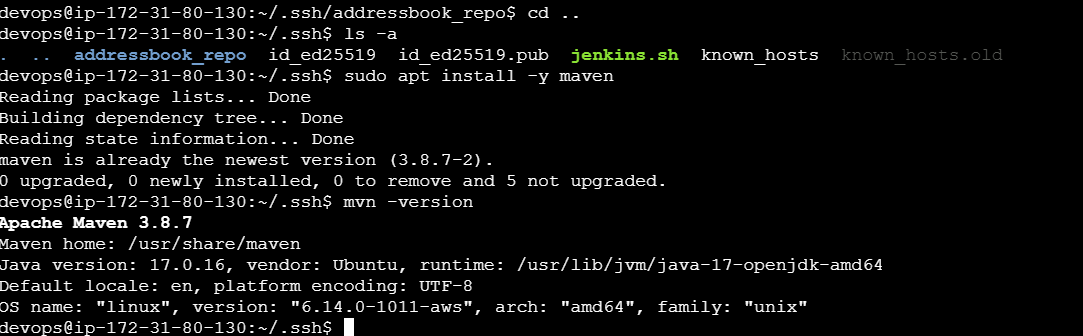
**Maven must be installed on your Jenkins Master EC2 instance** before you run the pipeline, because the Jenkinsfile uses Maven commands (mvn clean package) to build the .war file.

# Install Maven

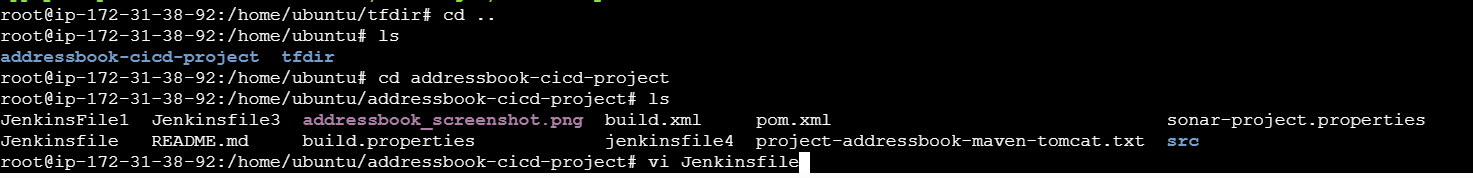
sudo apt install -y maven

# Verify installation

mvn -version



Modify the Jenkins file which is present in the addressbook-cicd-project





**Below code change the github repo link**

pipeline {

agent any

environment {

WAR\_FILE = "target/addressbook.war"

TOMCAT\_HOME = "/opt/tomcat9"

INVENTORY = "addressbook\_repo/inventory.ini"

}

stages {

stage('Checkout Code') {

steps {

git branch: 'master', url: '[**https://github.com/Sathya252/milestone\_practice.git**](https://github.com/Sathya252/milestone_practice.git)'

}

}

stage('Build with Maven') {

steps {

sh 'mvn clean package'

}

}

stage('Install Tomcat 9 via Ansible') {

steps {

writeFile file: 'tomcat.yml', text: '''

- hosts: webservers

become: true

tasks:

- name: Install Java

apt:

name: openjdk-11-jre

state: present

update\_cache: yes

- name: Install unzip

apt:

name: unzip

state: present

- name: Download Tomcat 9

get\_url:

url: https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.zip

dest: /tmp/apache-tomcat-9.0.108.zip

- name: Extract Tomcat

unarchive:

src: /tmp/apache-tomcat-9.0.108.zip

dest: /opt/

remote\_src: yes

- name: Rename Tomcat folder

command: mv /opt/apache-tomcat-9.0.108 /opt/tomcat9

args:

creates: /opt/tomcat9

- name: Make Tomcat scripts executable

command: chmod +x /opt/tomcat9/bin/\*.sh

'''

sh 'ansible-playbook -i ${INVENTORY} tomcat.yml'

}

}

stage('Deploy WAR to Tomcat') {

steps {

sh 'ansible webservers -i ${INVENTORY} -m copy -a "src=${WAR\_FILE} dest=${TOMCAT\_HOME}/webapps/addressbook.war" --become'

sh 'ansible webservers -i ${INVENTORY} -m shell -a "${TOMCAT\_HOME}/bin/shutdown.sh || true" --become'

sh 'ansible webservers -i ${INVENTORY} -m shell -a "${TOMCAT\_HOME}/bin/startup.sh" --become'

}

}

}

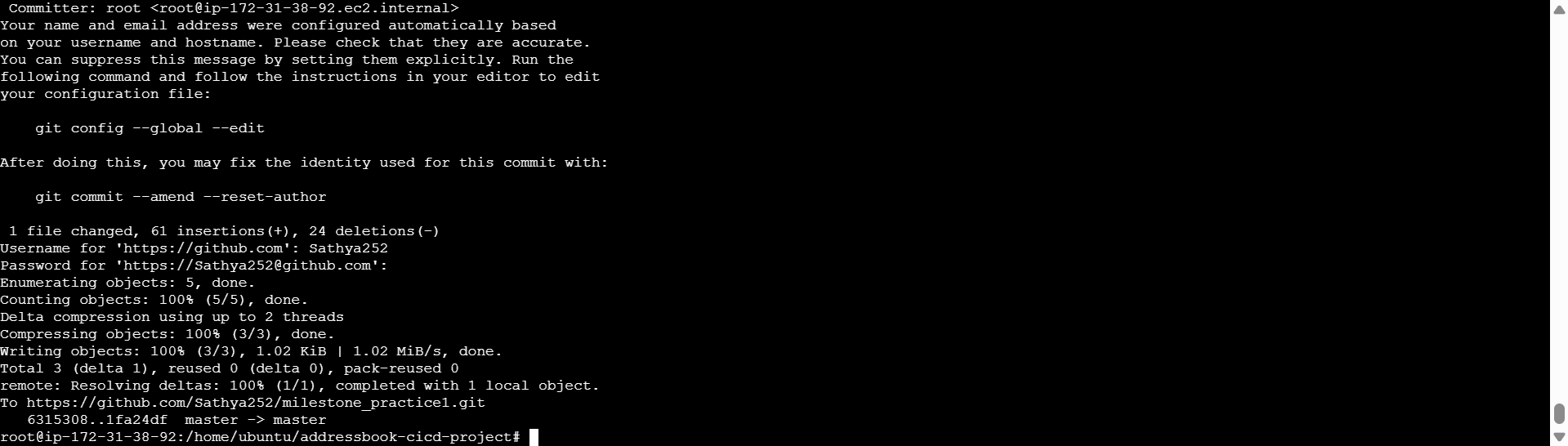
}

**Push this Modified Jenkinsfile to your repo:**

git add Jenkinsfile

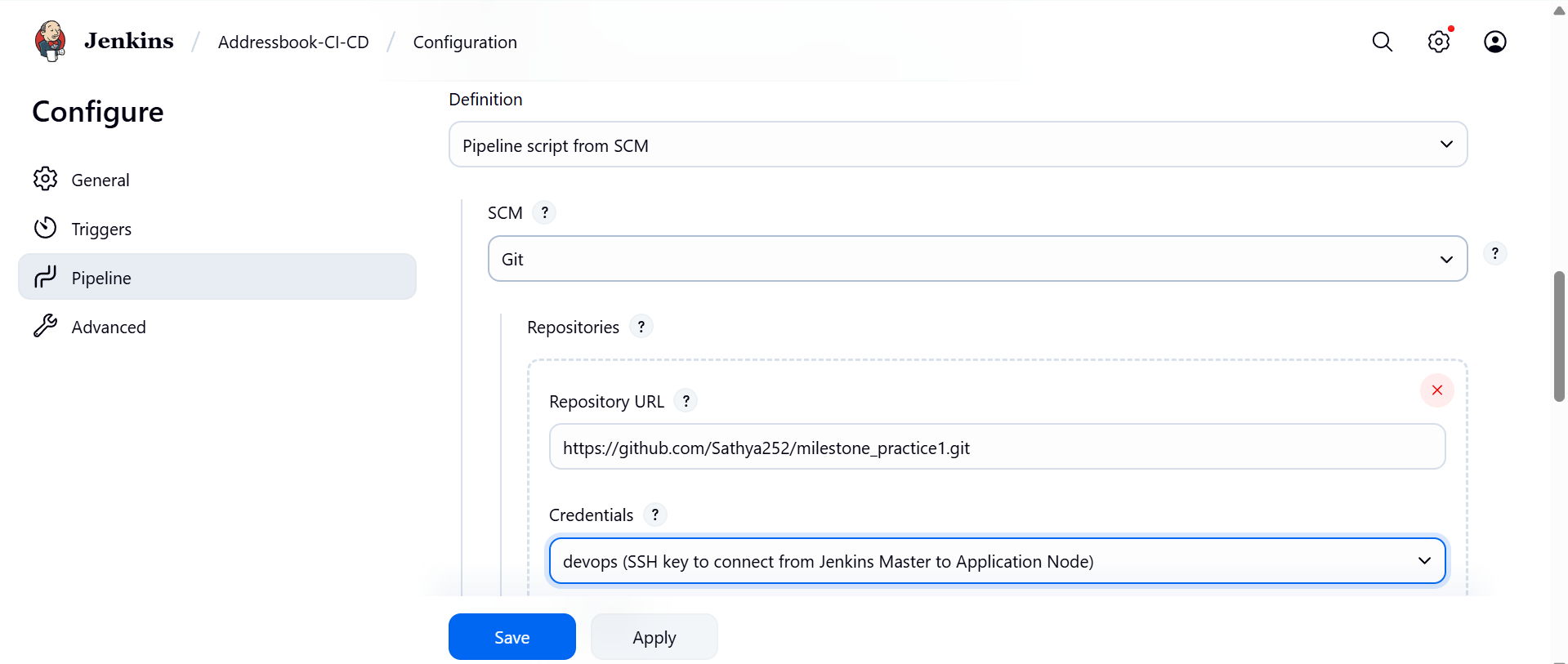
git commit -m "Final Jenkinsfile ready for master branch"

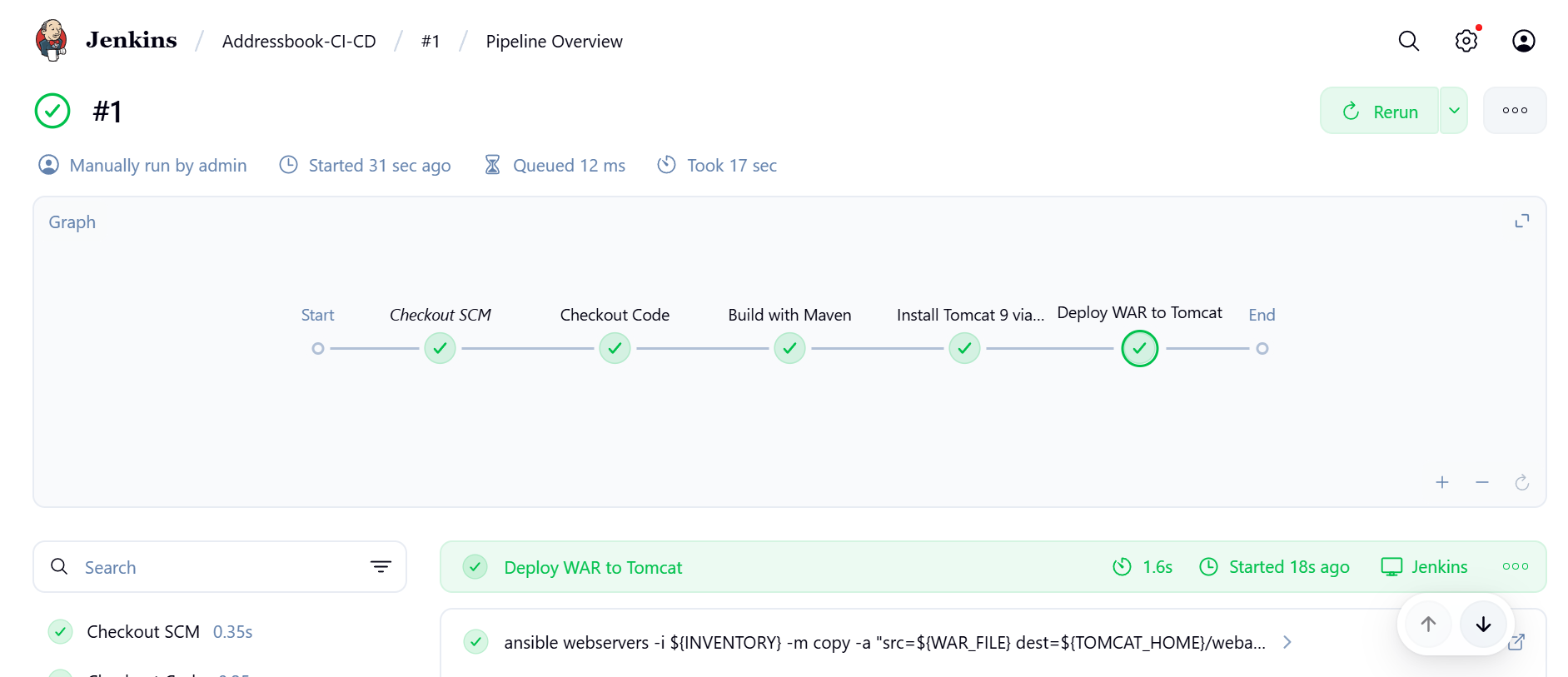
git push origin master



**Create Pipeline Job in Jenkins**

* **New Item → Pipeline → Name it Addressbook-CI-CD**
* Pipeline script from SCM → Git → Repo URL → Branch master → Script Path Jenkinsfile
* Save the job.





Configure a GitHub Webhook to trigger the Jenkins pipeline automatically on code push. i think it is done right?

### **What a GitHub Webhook Does**

* When you push code to GitHub → GitHub sends a **HTTP POST** to Jenkins → Jenkins automatically triggers the pipeline.
* This means you don’t have to click **Build Now** manually.

### **Check if it’s already done**

1. Go to your GitHub repo → **Settings → Webhooks**
2. Look for a webhook with URL like:

http://<Jenkins-Master-Public-IP>:8080/github-webhook/

1. If it exists → it’s done.
2. If not → you need to add it.

### **How to Add Webhook (if not done)**

1. **Payload URL:**

http://<Jenkins-Master-Public-IP>:8080/github-webhook/

1. **Content type:** application/json
2. **Secret:** Optional
3. **Which events would you like to trigger this webhook?**

* Select **Just the push event**

1. Click **Add webhook**

