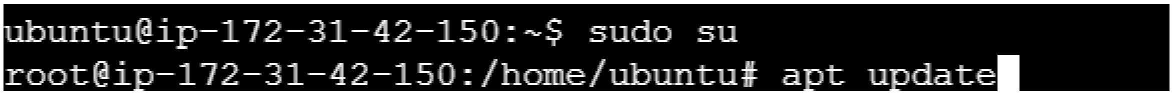
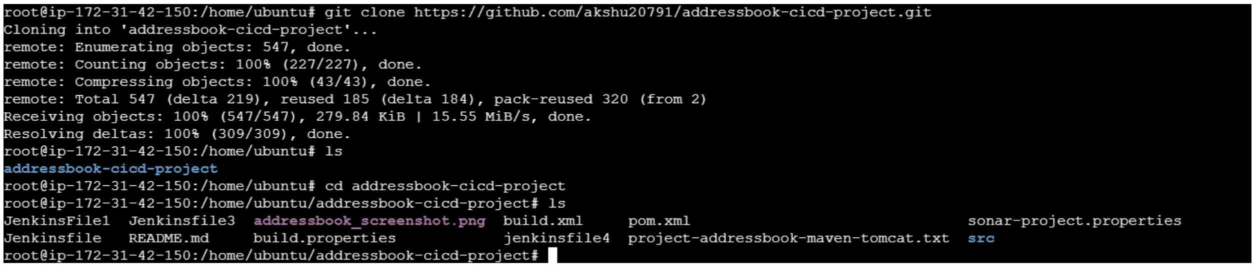
1. **Clone the GitHub repository:** **https://github.com/akshu20791/addressbook-cicd-project**



```
ubuntu@ip-172-31-42-150:~$ sudo su
root@ip-172-31-42-150:/home/ubuntu# apt update
```

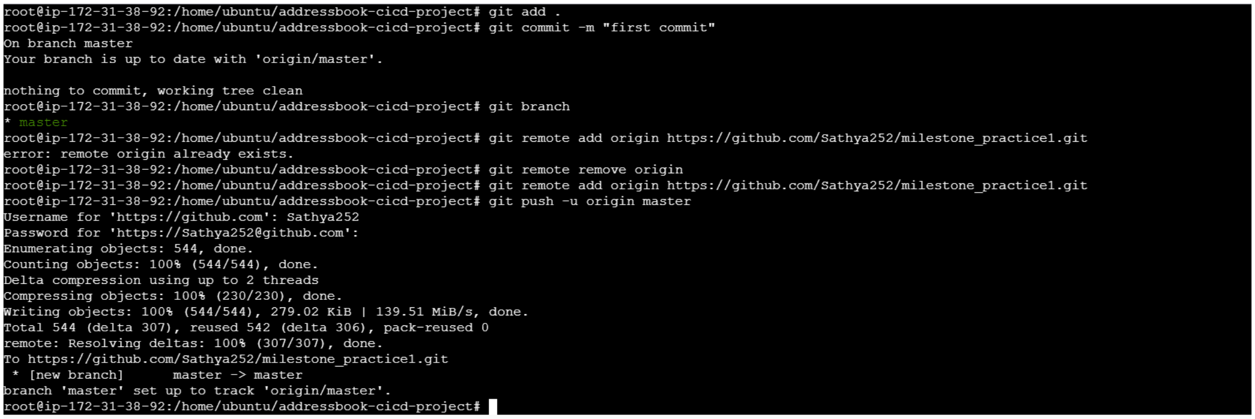


```
root@ip-172-31-42-150:/home/ubuntu# git clone https://github.com/akshu20791/addressbook-cicd-project.git
Cloning into 'addressbook-cicd-project'...
remote: Enumerating objects: 547, done.
remote: Counting objects: 100% (227/227), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 547 (delta 219), reused 185 (delta 184), pack-reused 320 (from 2)
Receiving objects: 100% (547/547), 279.84 KiB | 15.55 MiB/s, done.
Resolving deltas: 100% (309/309), done.
root@ip-172-31-42-150:/home/ubuntu# ls
addressbook-cicd-project
root@ip-172-31-42-150:/home/ubuntu# cd addressbook-cicd-project
root@ip-172-31-42-150:/home/ubuntu/addressbook-cicd-project# ls
JenkinsFile1  Jenkinsfile3  addressbook_screenshot.png  build.xml     pom.xml                                 sonar-project.properties
Jenkinsfile   README.md     build.properties            jenkinsfile4  project-addressbook-maven-tomcat.txt    src
root@ip-172-31-42-150:/home/ubuntu/addressbook-cicd-project#
```

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

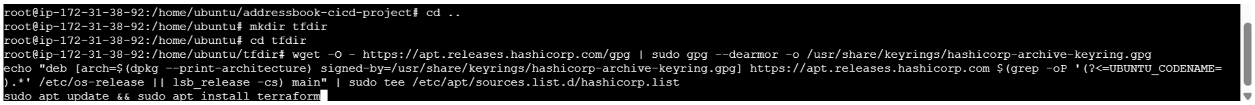
2. **Create your own GitHub repository and push the cloned code to it.**

```
# 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
```



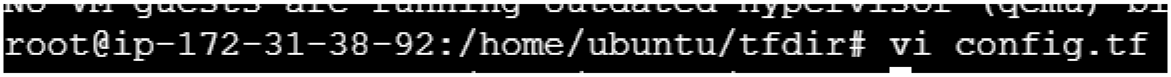
```
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git add .
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git commit -m "first commit"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git branch
* master
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git remote add origin https://github.com/Sathya252/milestone_practice1.git
error: remote origin already exists.
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git remote remove origin
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git remote add origin https://github.com/Sathya252/milestone_practice1.git
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# git push -u origin master
Username for 'https://github.com': Sathya252
Password for 'https://Sathya252@github.com':
Enumerating objects: 544, done.
Counting objects: 100% (544/544), done.
Delta compression using up to 2 threads
Compressing objects: 100% (230/230), done.
Writing objects: 100% (544/544), 279.02 KiB | 139.51 MiB/s, done.
Total 544 (delta 307), reused 542 (delta 306), pack-reused 0
remote: Resolving deltas: 100% (307/307), done.
To https://github.com/Sathya252/milestone_practice1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project#
```

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**.



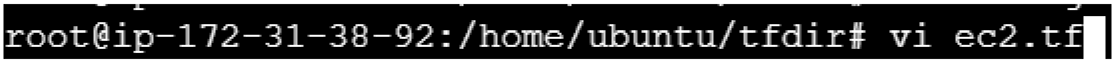
```
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# cd ..
root@ip-172-31-38-92:/home/ubuntu# mkdir tfdir
root@ip-172-31-38-92:/home/ubuntu# cd tfdir
root@ip-172-31-38-92:/home/ubuntu/tfdir# wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
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
```

```
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
```



```
root@ip-172-31-38-92:/home/ubuntu/tfdir# vi config.tf
```

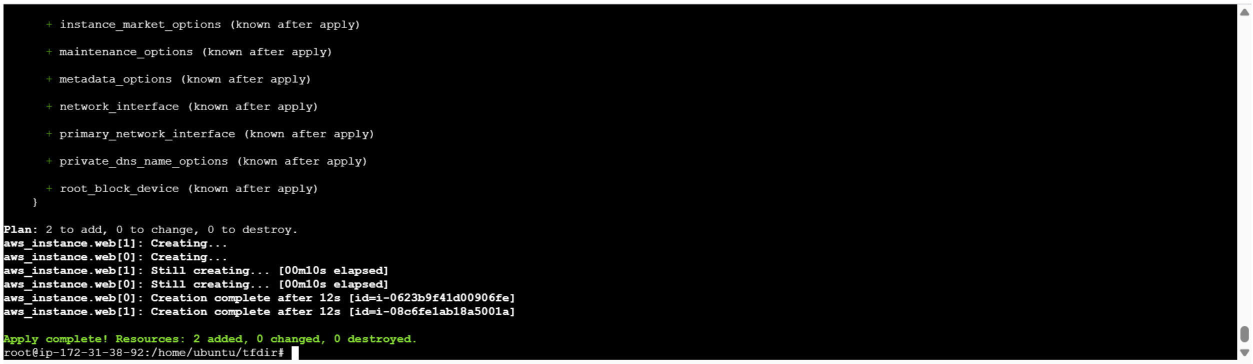
```
provider "aws" {
  region     = "us-west-2"
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}
```



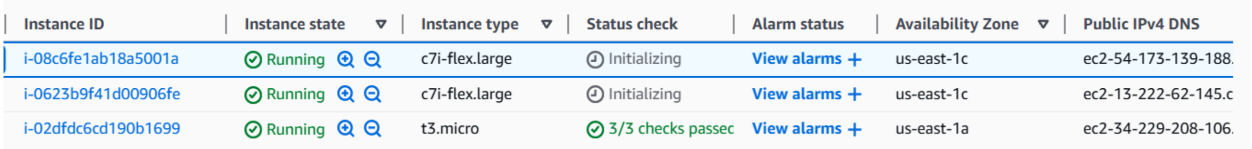
```
root@ip-172-31-38-92:/home/ubuntu/tfdir# vi ec2.tf
```

```
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



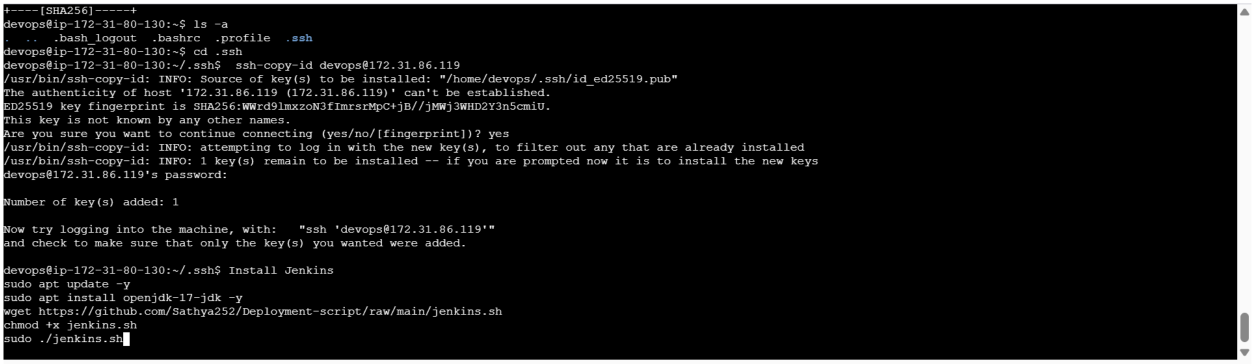
```
      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + primary_network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_instance.web[1]: Creating...
aws_instance.web[0]: Creating...
aws_instance.web[1]: Still creating... [00m10s elapsed]
aws_instance.web[0]: Still creating... [00m10s elapsed]
aws_instance.web[0]: Creation complete after 12s [id=i-0623b9f41d00906fe]
aws_instance.web[1]: Creation complete after 12s [id=i-08c6fe1ab18a5001a]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@ip-172-31-38-92:/home/ubuntu/tfdir#
```



Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
i-08c6fe1ab18a5001a	Running	c7i-flex.large	Initializing	View alarms +	us-east-1c	ec2-54-173-139-188.
i-0623b9f41d00906fe	Running	c7i-flex.large	Initializing	View alarms +	us-east-1c	ec2-13-222-62-145.c
i-02dfdc6cd190b1699	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	ec2-34-229-208-106.

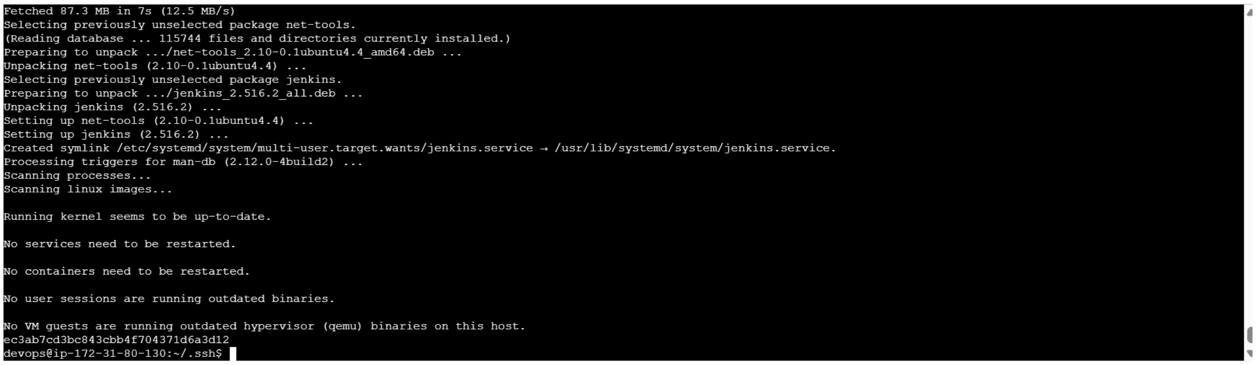
5. Install Jenkins on the Jenkins Master EC2 instance.

Complete the ansible installization process using master and node instances



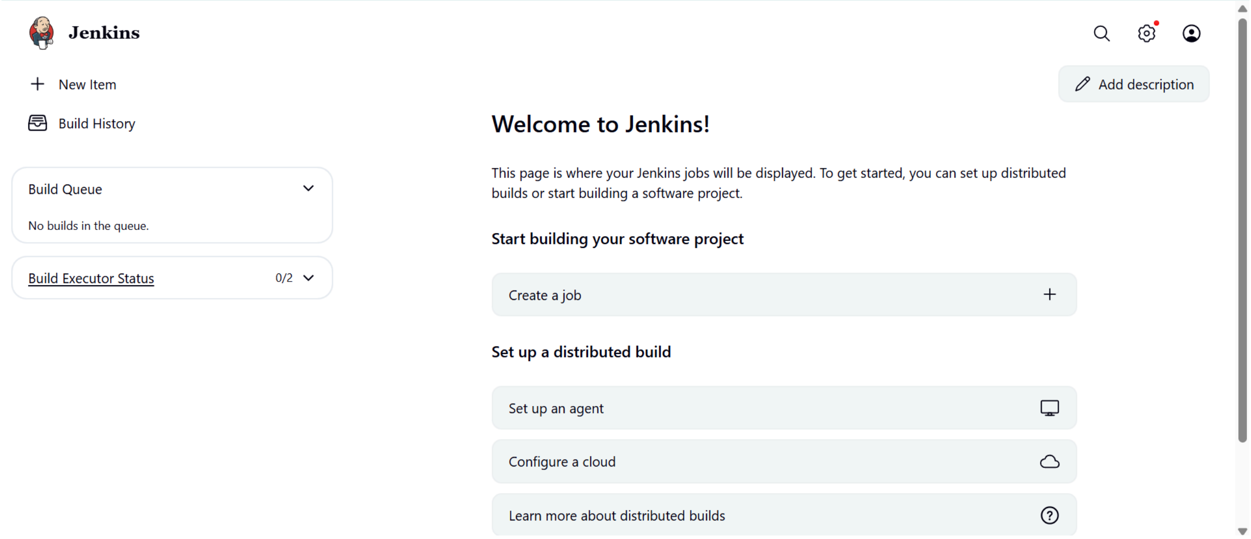
```
+----[SHA256]-----+
devops@ip-172-31-80-130:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  .ssh
devops@ip-172-31-80-130:~$ cd .ssh
devops@ip-172-31-80-130:~/.ssh$  ssh-copy-id devops@172.31.86.119
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.86.119 (172.31.86.119)' can't be established.
ED25519 key fingerprint is SHA256:WWrd9lmxzoN3fImrsrMpC+jB//jMWj3WHD2Y3n5cmiU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.86.119's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'devops@172.31.86.119'"
and check to make sure that only the key(s) you wanted were added.

devops@ip-172-31-80-130:~/.ssh$ 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
```

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



```
Fetched 87.3 MB in 7s (12.5 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 115744 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4.4) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.516.2_all.deb ...
Unpacking jenkins (2.516.2) ...
Setting up net-tools (2.10-0.1ubuntu4.4) ...
Setting up jenkins (2.516.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ec3ab7cd3bc843cbb4f704371d6a3d12
devops@ip-172-31-80-130:~/.ssh$
```

Open jenkins : <Master Public id>:8080



No builds in the queue.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

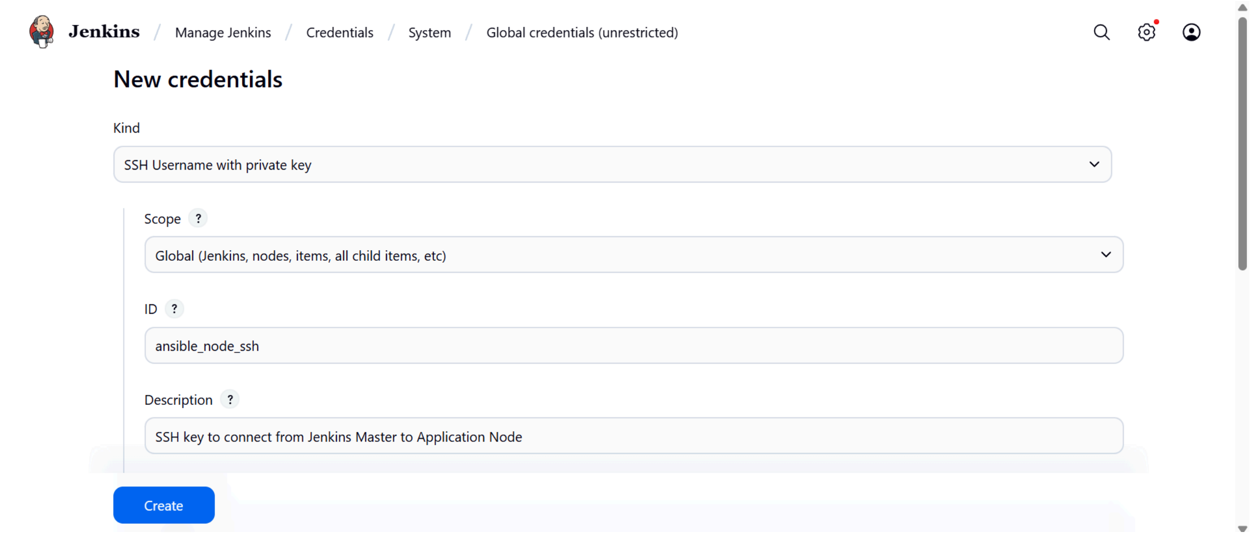
Create a job +

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)
Create

New credentials

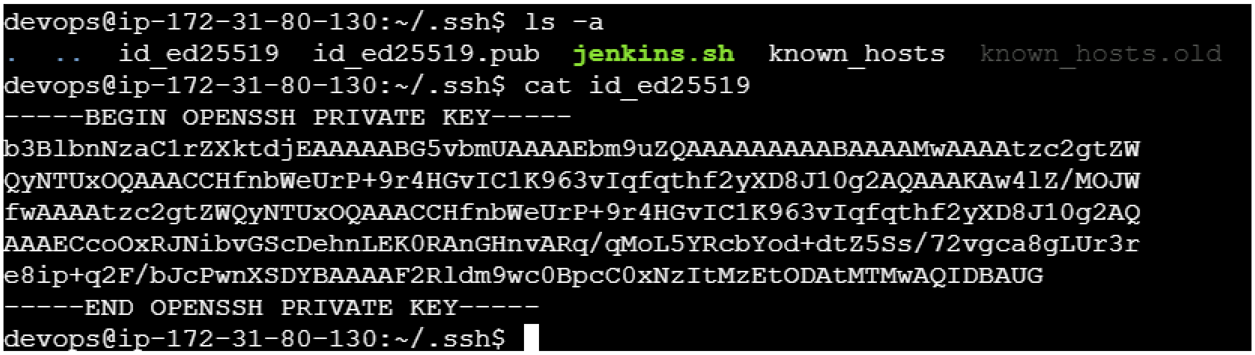
Kind: SSH Username with private key

Scope ?: Global (Jenkins, nodes, items, all child items, etc)

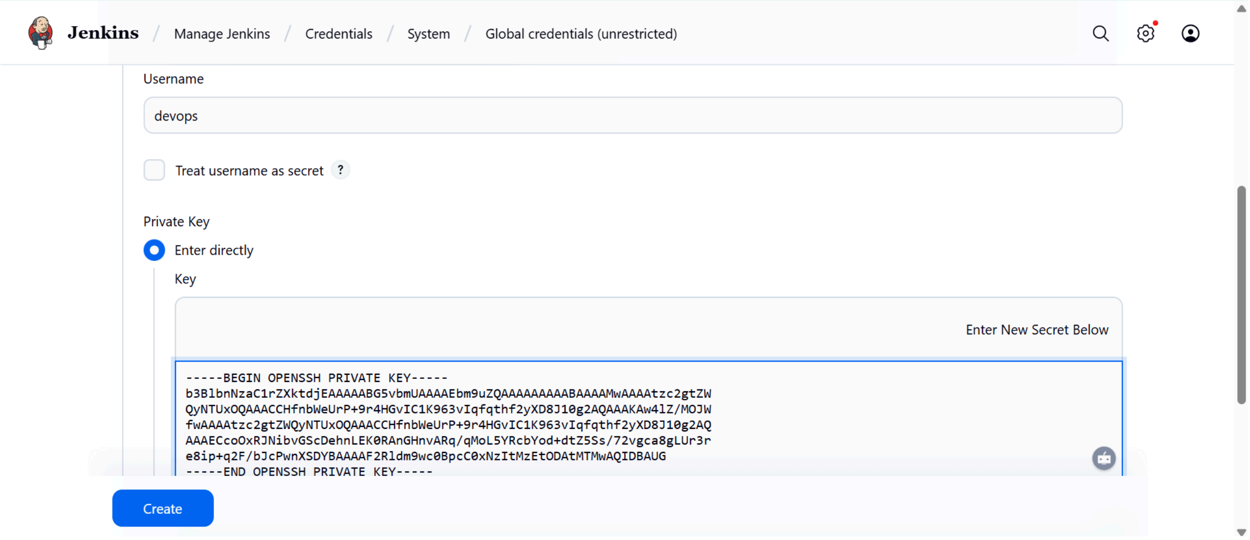
ID ?: ansible_node_ssh

Description ?: SSH key to connect from Jenkins Master to Application Node

For Privite key go to EC2 Master Instance



```
devops@ip-172-31-80-130:~/.ssh$ ls -a
.  ..  id_ed25519  id_ed25519.pub  jenkins.sh  known_hosts  known_hosts.old
devops@ip-172-31-80-130:~/.ssh$ cat id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACCHfnbWeUrP+9r4HGvIC1K963vIqfqthf2yXD8J10g2AQAAAKAw41Z/MOJW
fwAAAAtzc2gtZWQyNTUxOQAAACCHfnbWeUrP+9r4HGvIC1K963vIqfqthf2yXD8J10g2AQ
AAAECcoOxRJNibvGScDehnLEK0RAnGHnvARq/qMoL5YRcbYod+dtZ5Ss/72vgca8gLUr3r
e8ip+q2F/bJcPwnXSDYBAAAAF2Rldm9wc0BpcC0xNzItMzEtODAtMTMwAQIDBAUG
-----END OPENSSH PRIVATE KEY-----
devops@ip-172-31-80-130:~/.ssh$
```



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

Username

devops

□ Treat username as secret ?

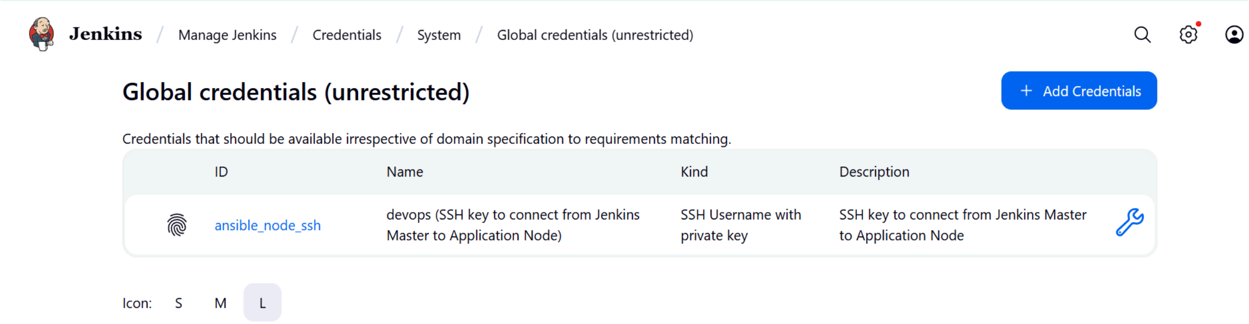
Private Key

● Enter directly

Key

Enter New Secret Below

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACCHfnbWeUrP+9r4HGvIC1K963vIqfqthf2yXD8J10g2AQAAAKAw41Z/MOJW
fwAAAAtzc2gtZWQyNTUxOQAAACCHfnbWeUrP+9r4HGvIC1K963vIqfqthf2yXD8J10g2AQ
AAAECcoOxRJNibvGScDehnLEK0RAnGHnvARq/qMoL5YRcbYod+dtZ5Ss/72vgca8gLUr3r
e8ip+q2F/bJcPwnXSDYBAAAAF2Rldm9wc0BpcC0xNzItMzEtODAtMTMwAQIDBAUG
-----END OPENSSH PRIVATE KEY-----
```



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

	ID	Name	Kind	Description
	ansible_node_ssh	devops (SSH key to connect from Jenkins Master to Application Node)	SSH Username with private key	SSH key to connect from Jenkins Master to Application Node

On your **master EC2**:

mkdir addressbook_repo

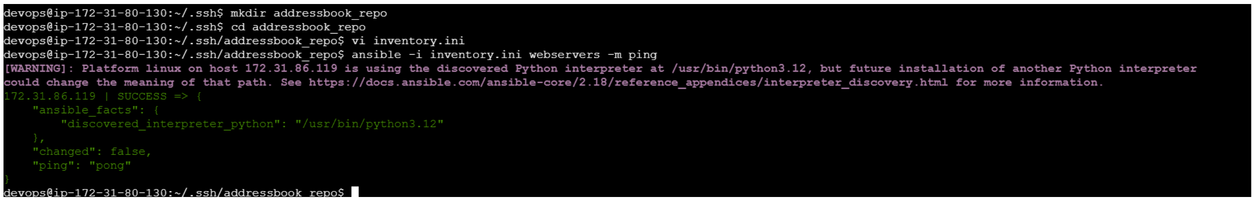
cd addressbook_repo

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



```
devops@ip-172-31-80-130:~/.ssh$ mkdir addressbook_repo
devops@ip-172-31-80-130:~/.ssh$ cd addressbook_repo
devops@ip-172-31-80-130:~/.ssh/addressbook_repo$ vi inventory.ini
devops@ip-172-31-80-130:~/.ssh/addressbook_repo$ ansible -i inventory.ini webservers -m ping
[WARNING]: Platform linux on host 172.31.86.119 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter
could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
172.31.86.119 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
devops@ip-172-31-80-130:~/.ssh/addressbook_repo$
```

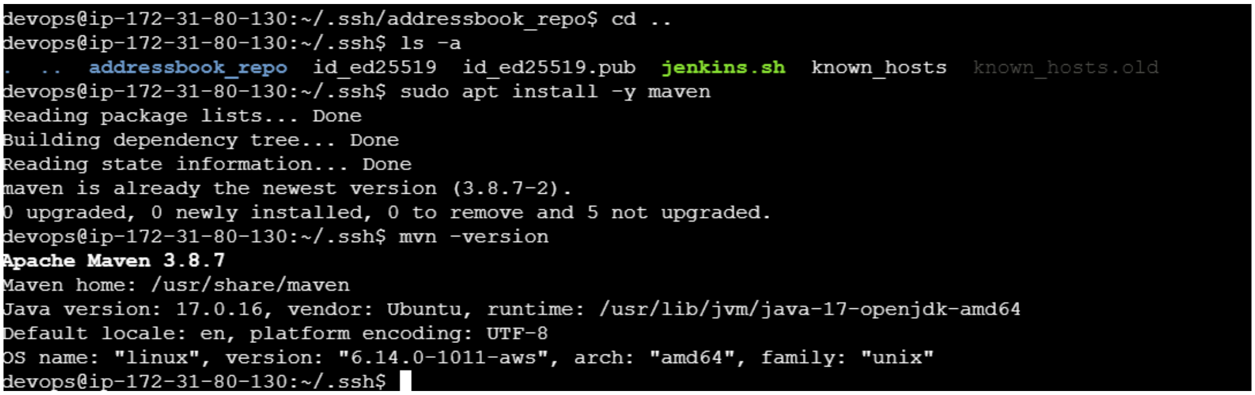
Maven must be installed on your Jenkins Master EC2 instance before you run the pipeline, because the Jenkinsfile uses Maven commands (`mvn clean package`) to build the `.war` file.

# Install Maven

sudo apt install -y maven

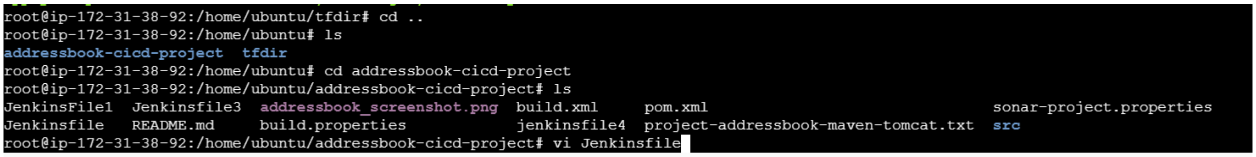
# Verify installation

mvn -version

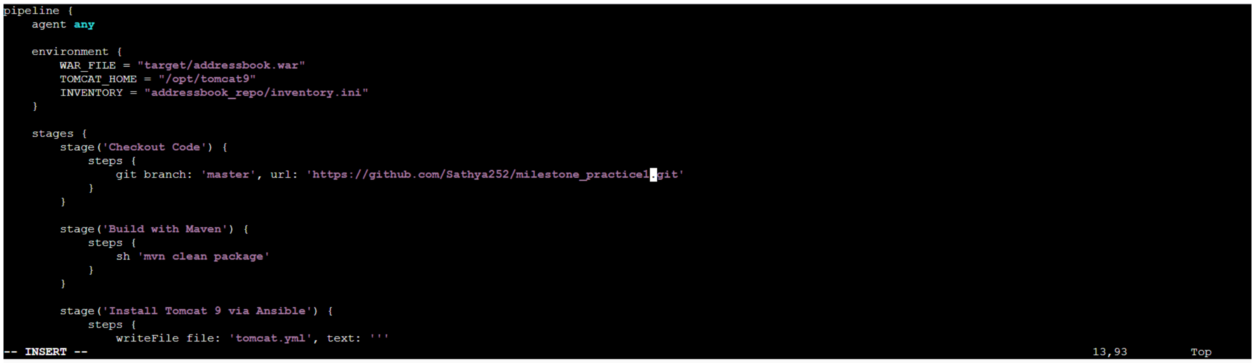


```
devops@ip-172-31-80-130:~/.ssh/addressbook_repo$ cd ..
devops@ip-172-31-80-130:~/.ssh$ ls -a
.  ..  addressbook_repo  id_ed25519  id_ed25519.pub  jenkins.sh  known_hosts  known_hosts.old
devops@ip-172-31-80-130:~/.ssh$ sudo apt install -y maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
maven is already the newest version (3.8.7-2).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
devops@ip-172-31-80-130:~/.ssh$ mvn -version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.16, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.14.0-1011-aws", arch: "amd64", family: "unix"
devops@ip-172-31-80-130:~/.ssh$
```

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



```
root@ip-172-31-38-92:/home/ubuntu/tfdir# cd ..
root@ip-172-31-38-92:/home/ubuntu# ls
addressbook-cicd-project  tfdir
root@ip-172-31-38-92:/home/ubuntu# cd addressbook-cicd-project
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# ls
JenkinsFile1  Jenkinsfile3  addressbook_screenshot.png  build.xml     pom.xml                                sonar-project.properties
Jenkinsfile   README.md     build.properties            jenkinsfile4  project-addressbook-maven-tomcat.txt  src
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project# vi Jenkinsfile
```



```
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_practice1.git'
            }
        }

        stage('Build with Maven') {
            steps {
                sh 'mvn clean package'
            }
        }

        stage('Install Tomcat 9 via Ansible') {
            steps {
                writeFile file: 'tomcat.yml', text: '''
-- INSERT --                                                                                   13,93         Top
```

Below code change the github repo link

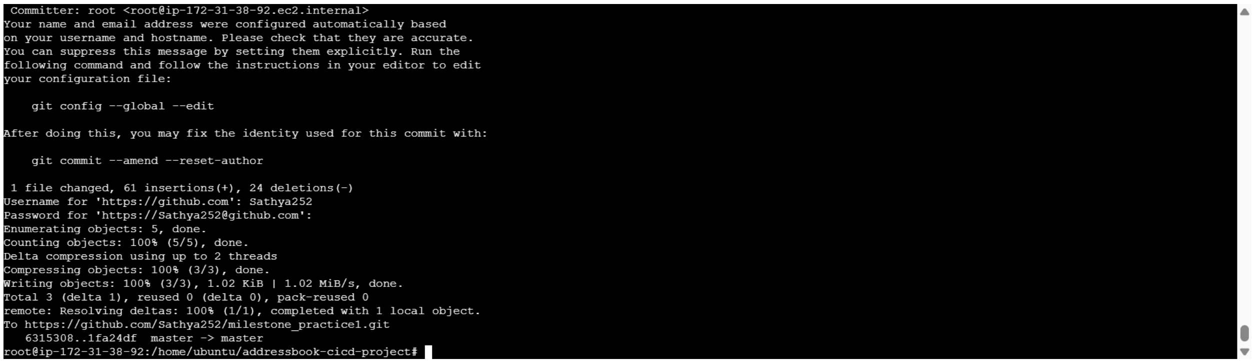
```
pipeline {

    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'
            }
        }

        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:

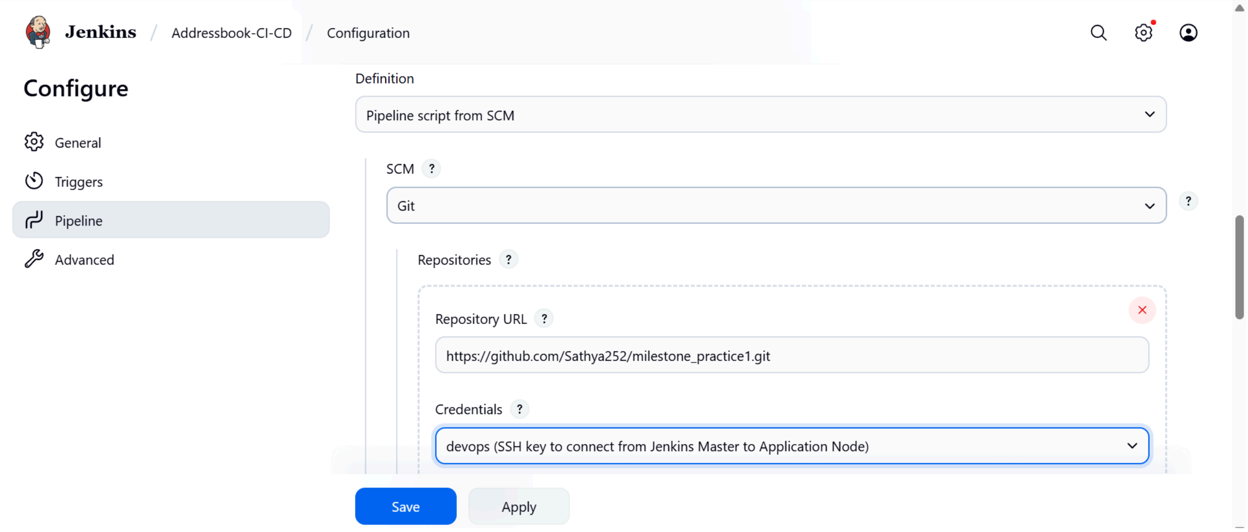
```
git add Jenkinsfile

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

git push origin master
```



```
 Committer: root <root@ip-172-31-38-92.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 61 insertions(+), 24 deletions(-)
Username for 'https://github.com': Sathya252
Password for 'https://Sathya252@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.02 KiB | 1.02 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Sathya252/milestone_practice1.git
   6315308..1fa24df  master -> master
root@ip-172-31-38-92:/home/ubuntu/addressbook-cicd-project#
```

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.



Jenkins / Addressbook-CI-CD / Configuration

Configure

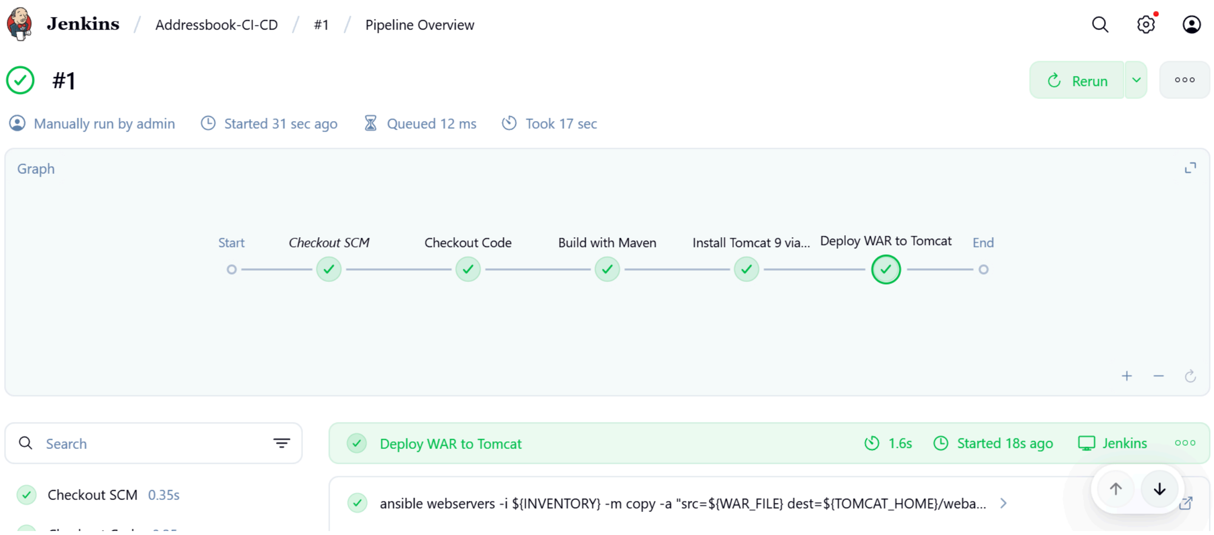
Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/Sathya252/milestone_practice1.git

Credentials: devops (SSH key to connect from Jenkins Master to Application Node)



Jenkins / Addressbook-CI-CD / #1 / Pipeline Overview

#1

Manually run by admin · Started 31 sec ago · Queued 12 ms · Took 17 sec

Graph: Start → Checkout SCM → Checkout Code → Build with Maven → Install Tomcat 9 via... → Deploy WAR to Tomcat → End

Checkout SCM 0.35s

Deploy WAR to Tomcat · 1.6s · Started 18s ago · Jenkins

ansible webservers -i ${INVENTORY} -m copy -a "src=${WAR_FILE} dest=${TOMCAT_HOME}/weba...

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/
```

3. If it exists → it's done.
4. If not → you need to add it.

How to Add Webhook (if not done)

1. **Payload URL:**

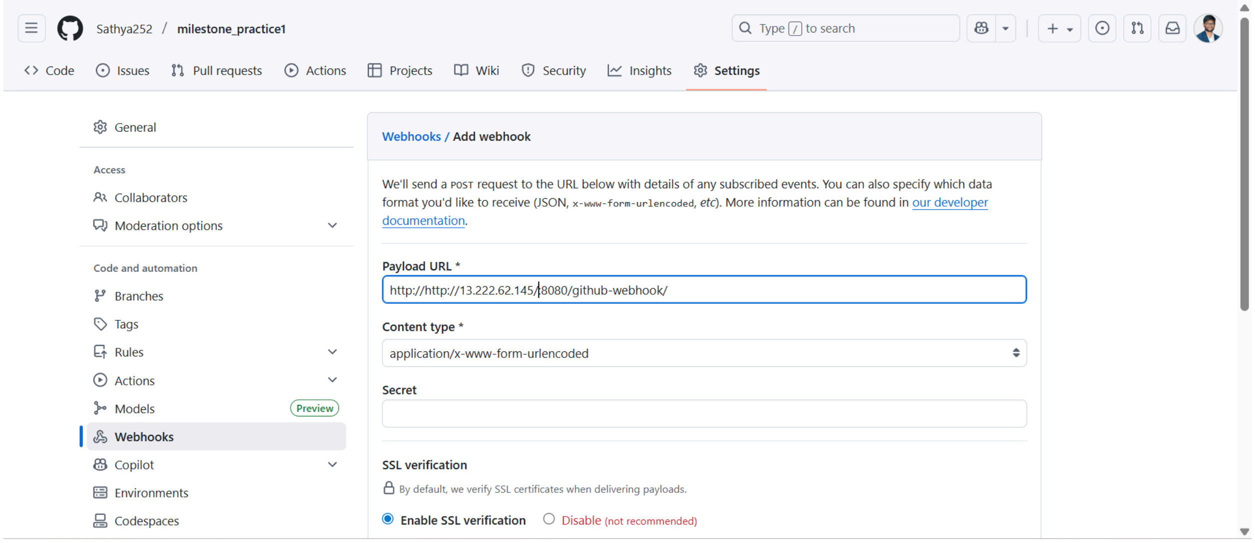
```
http://<Jenkins-Master-Public-IP>:8080/github-webhook/
```

2. **Content type:** `application/json`
3. **Secret:** Optional

4. **Which events would you like to trigger this webhook?**

- Select **Just the push event**

5. Click **Add webhook**



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

http://http://13.222.62.145/:8080/github-webhook/

Content type *

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

● Enable SSL verification ○ Disable (not recommended)