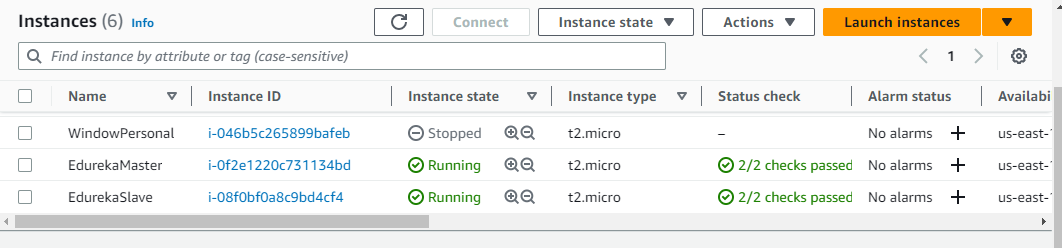
1. Created Two Instance for this project in AWS



1. Install Jenkins, Java, git and Ansible on EdurekaMaster

sudo yum update –y

sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

sudo yum upgrade

**Install Java**

sudo amazon-linux-extras install java-openjdk11 -y

**Install Git**

sudo yum install git -y

**Install Jenkins**

sudo yum install jenkins -y

sudo systemctl enable jenkins

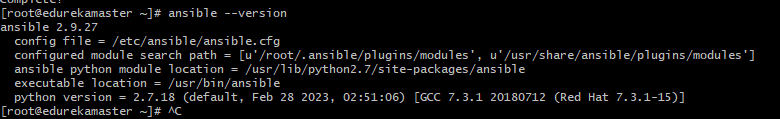
sudo systemctl start Jenkins

sudo systemctl start Jenkins

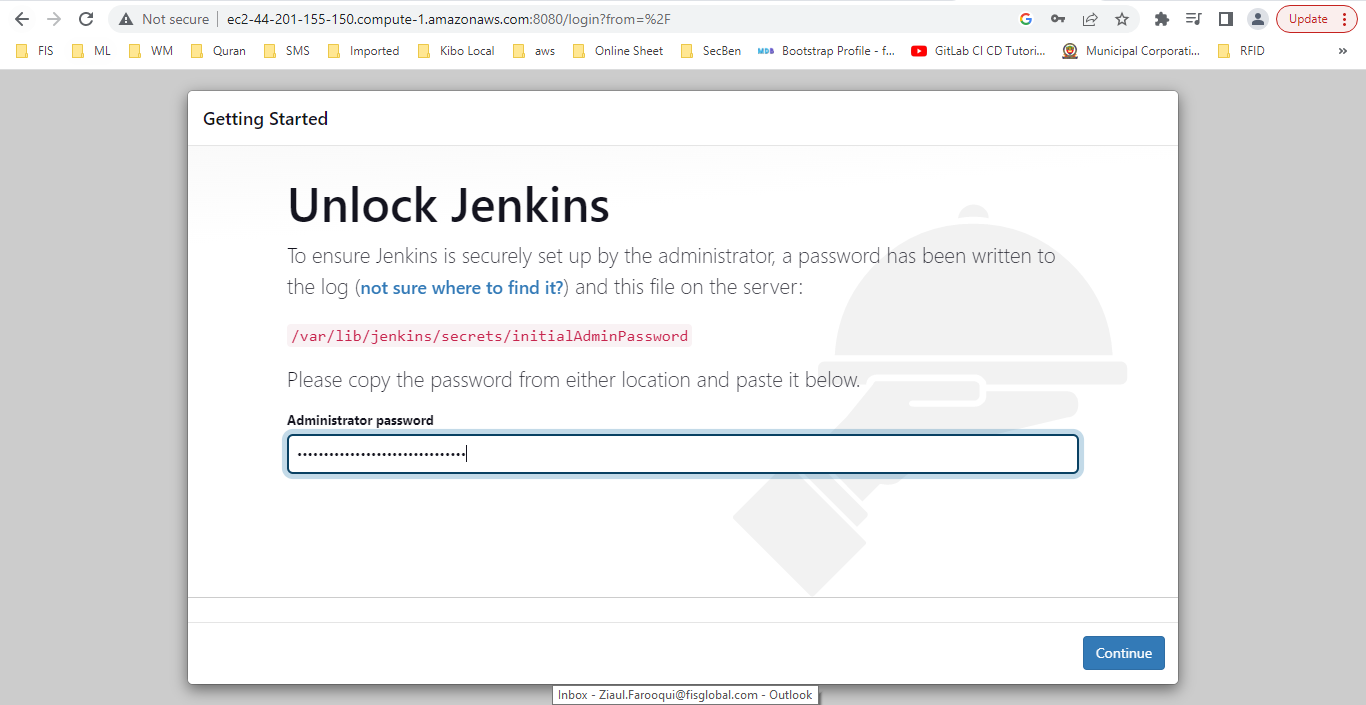
**Install Ansible**

sudo amazon-linux-extras install epel

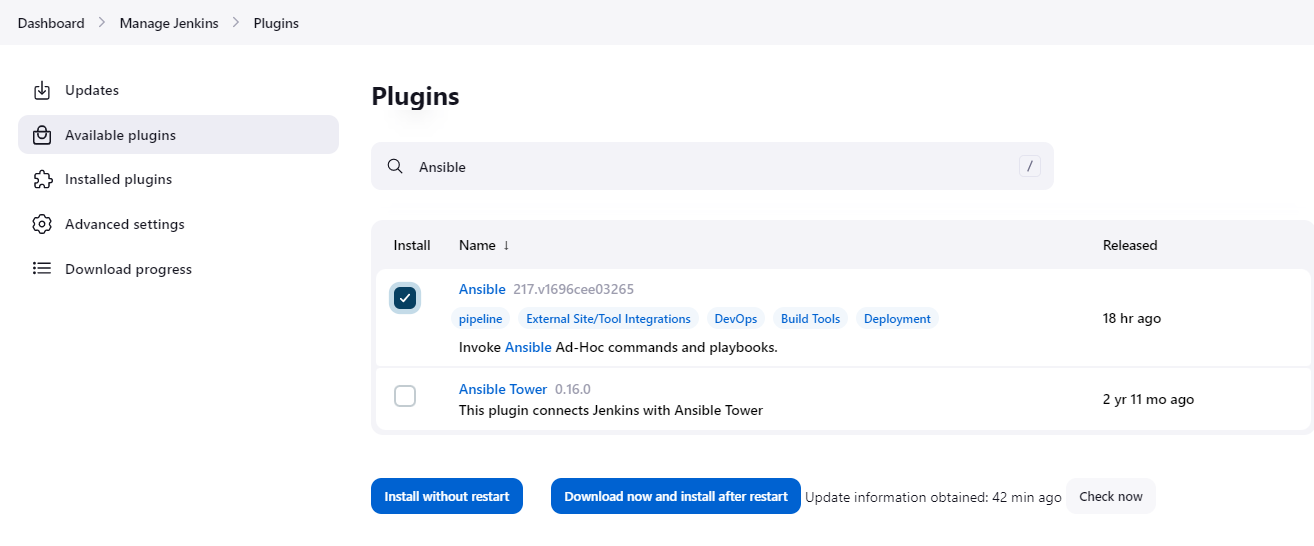
yum install ansible



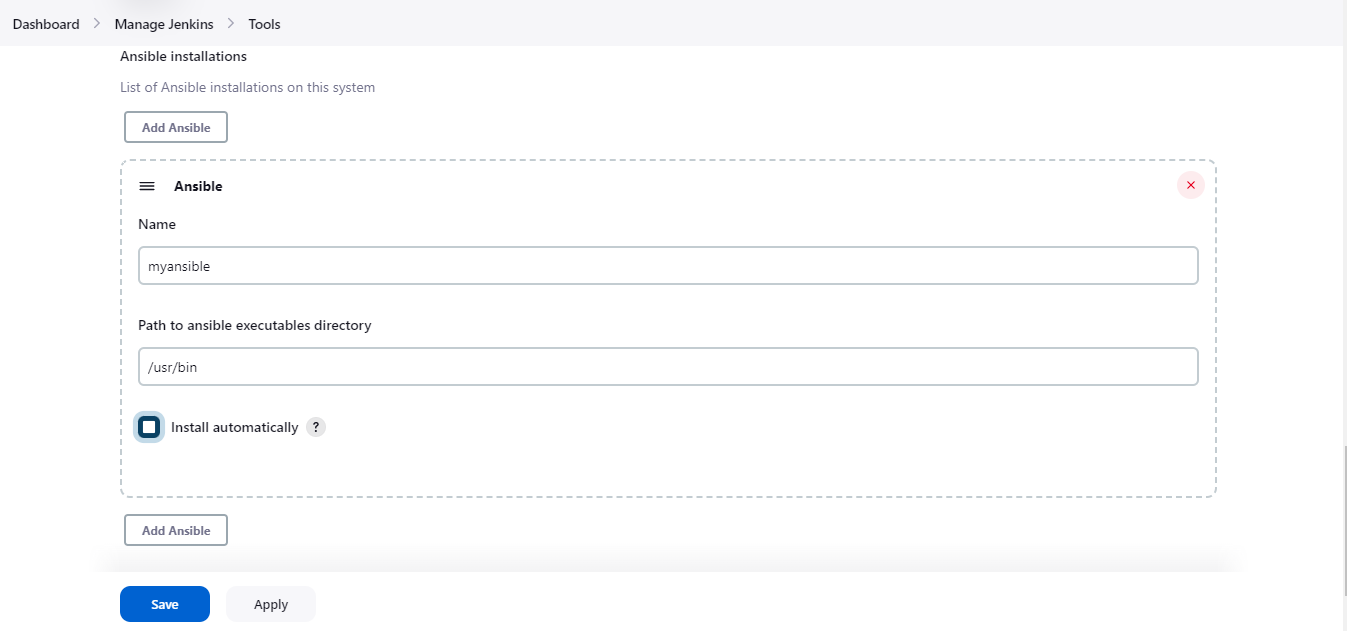
1. Now unlock Jenkins by open it in browser with URL



1. Install maven and ansible plugins



1. Configure MAVEN and ANSIBLE in Jenkins,



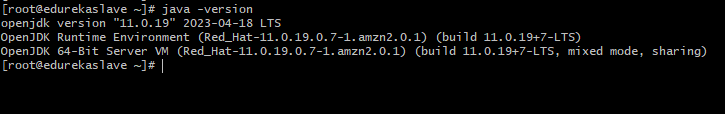
1. Now connect to the AnsibleSlave system to make this system as Jenkins agent
2. Install git and java

**Install Git**

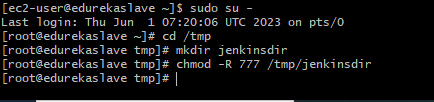
yum install git -y

**Install java**

sudo amazon-linux-extras install java-openjdk11

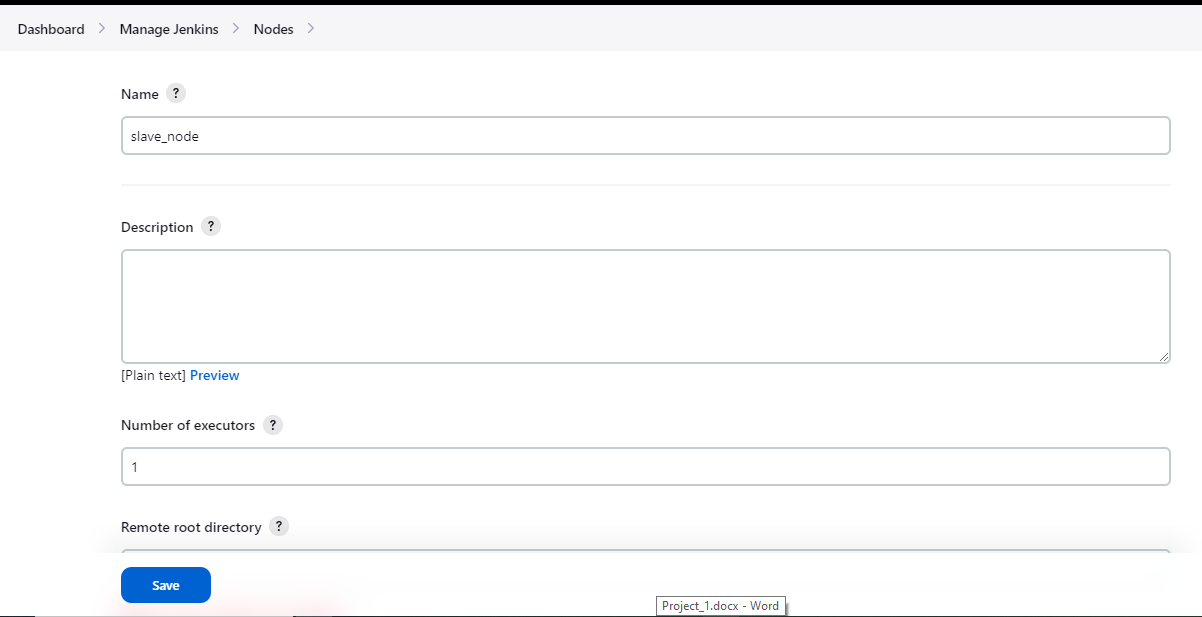


1. Create a directory on Edureka Slave machine and give permission.



Similarly create /tmp/ansiblehost and assign permission to it.

1. Now go to Jenkins and configure the slave node



1. Provided the host IP of slave and SSH connection details and private key, I am able to see the connection is successful.

A screenshot of a computer

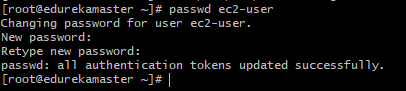
Description automatically generated

1. Here Job 1 is completed.
2. Now we are moving to setup Ansible Controller (EdurekaMaster) and Ansible Host (EdurekaSlave).
3. Go to Edureka Master to setup it as Ansible Controller, run the below command

vim /etc/ssh/sshd\_config



Set the password for ec2-user



Add ec2-user in sudoers files and give all permission

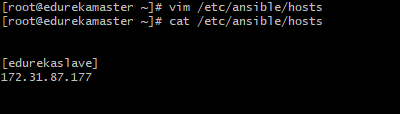
vim /etc/sudoers



Restart your sshd connection

systemctl restart sshd

1. Go to Edureka Host machine to setup it as Ansible Host, run the same command as above
2. Now it is required to add host machine details on ACM.

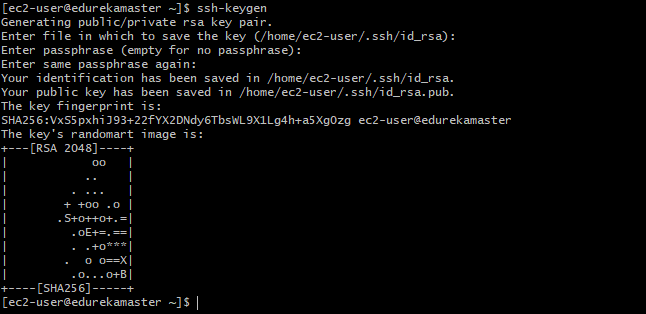


1. Now establish the connection between controller and host by copying generated key to host machine. So, run the below commands on ACM

su - ec2-user

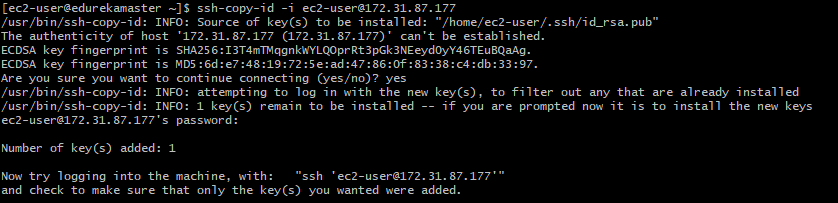
ssh-keygen

provide the details as per your convenient, in my case I did not provided anything and pressed enter, enter, enter. ssh key generated



1. Now copy key on to the host

ssh-copy-id -i [ec2-user@172.31.87.177](mailto:ec2-user@172.31.87.177)

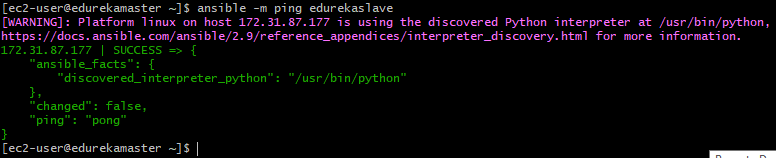


ssh connection will be done

you can verify by running the below command

ssh 'ec2-user@172.31.87.177'

1. Now ping to host server to check ansible connection.



1. Here Ansible setup has been completed, now the next steps are:
   1. Create git repo for source code.
   2. Add host and playbook files in git to set docker on host machine.
2. Below are the Ansible files in repository.

**playbook.yml**

A screenshot of a computer

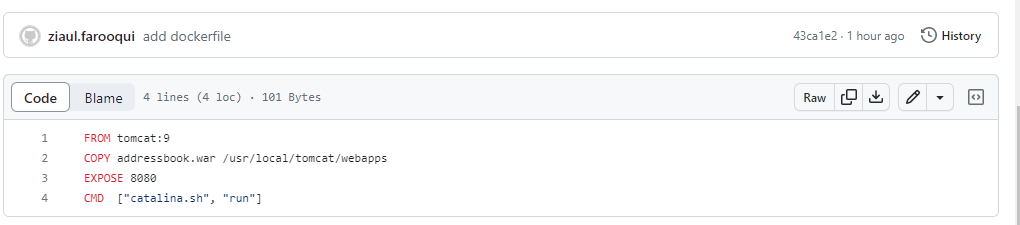
Description automatically generated with medium confidence

**devhost.inv**

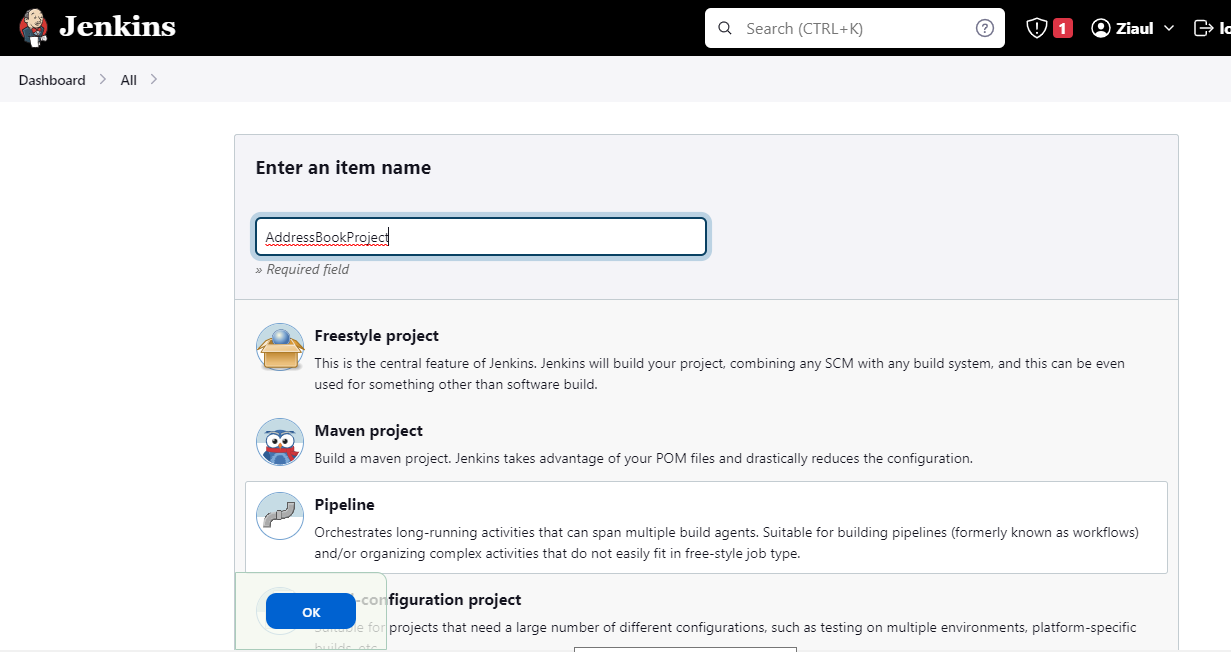
A picture containing text, font, line, screenshot

Description automatically generated

Dockerfile



1. Now create a new project AddressBookProject of pipeline type in Jenkins



A screenshot of a chat

Description automatically generated with medium confidence

1. Below are the complete pipeline code.

pipeline{

tools{

maven 'mymaven'

}

agent any

stages{

stage('Checkout the code'){

steps{

echo 'cloning the repo'

git **'https://github.com/ziaul-farooqui/addressbook.git'**

}

}

stage('Compile'){

steps{

echo 'complie the code again..'

sh 'mvn compile'

}

}

stage('CodeReview'){

steps{

echo 'codeReview'

sh 'mvn pmd:pmd'

}

}

stage('UnitTest'){

steps{

sh 'mvn test'

}

}

stage('Package'){

steps{

sh 'mvn package'

sh 'cp /var/lib/jenkins/workspace/AddressBookProject/target/addressbook.war .'

}

}

stage('Configure Host'){

steps{

ansiblePlaybook credentialsId: 'ansiblehost', disableHostKeyChecking: true, installation: 'myansible', inventory: 'devhost.inv', playbook: 'playbook.yml'

}

}

}

}

1. How it works
   1. Once AddressBookProject build started, it first compile code and create war file
   2. War file then copied to root location
   3. Then ansible playbook.yml will run on each ansible host and execute the steps below
      1. Install docker package (Job2)
      2. Start docker service
      3. Remove existing container if running
      4. Copy war file to host machine
      5. Copy Dockerfile to host machine
      6. Build the image with fresh war file.
      7. Run the new image container on port 8080 and bind the same port to host machine.
2. Verify the application on below URL
   1. <http://ec2-44-212-35-123.compute-1.amazonaws.com:8080/addressbook/>

A screenshot of a computer

Description automatically generated with medium confidence

Github repository URL : <https://github.com/ziaul-farooqui/addressbook>