MODULE NAME

CAPSTONEPROJECT

PROJECT NAME

INSURANCE-DOMAIN

SUBMITTED BY : RAHUL R

ON : 26-05-2025

Business challenge/requirement

As soon as the developer pushes the updated code on the GIT master branch, the Jenkins job should be triggered using a GitHub Webhook and Jenkins job should be triggered, The code should be checked out, compiled, tested, packaged and containerized and deployed to the preconfigured test-server automatically.

The deployment should then be tested using a test automation tool (Selenium), and if the build is successful, it should be deployed to the prod server. All this should happen automatically and should be triggered from a push to the GitHub master branch.

Tech stack

✓ Git - For version control for tracking changes in the code files

✓ Jenkins - For continuous integration and continuous deployment

✓ Docker - For deploying containerized applications

✓ Ansible - Configuration management tools

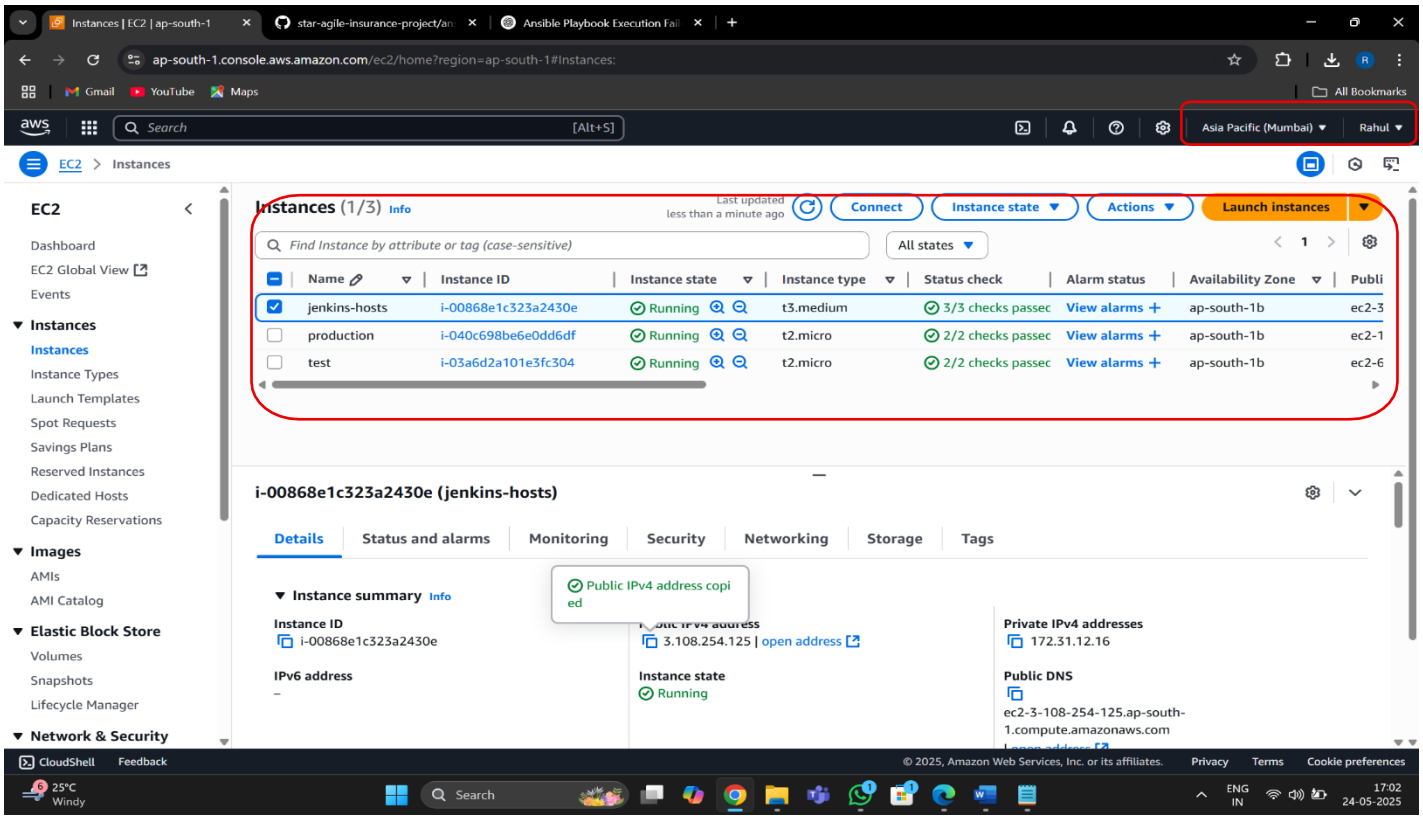
✓ Selenium - For automating tests on the deployed web application

✓ AWS : For creating ec2 machines as servers and deploy the web application.

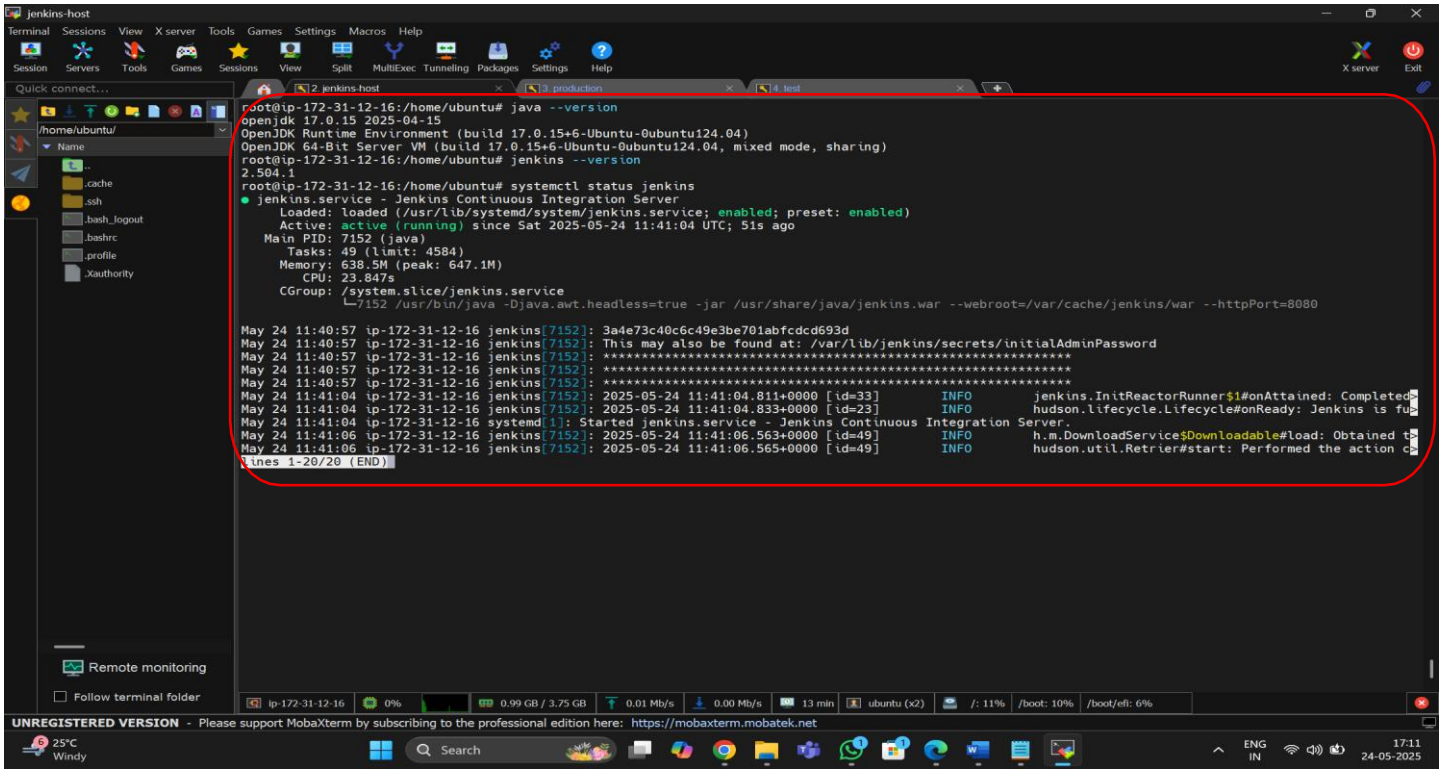
Launch three Instances

Name	Instance type	OS
Jenkins-master	- t3.medium	ubuntu
Production-1	- t2.micro	ubuntu
Test-1	- t2.micro	ubuntu

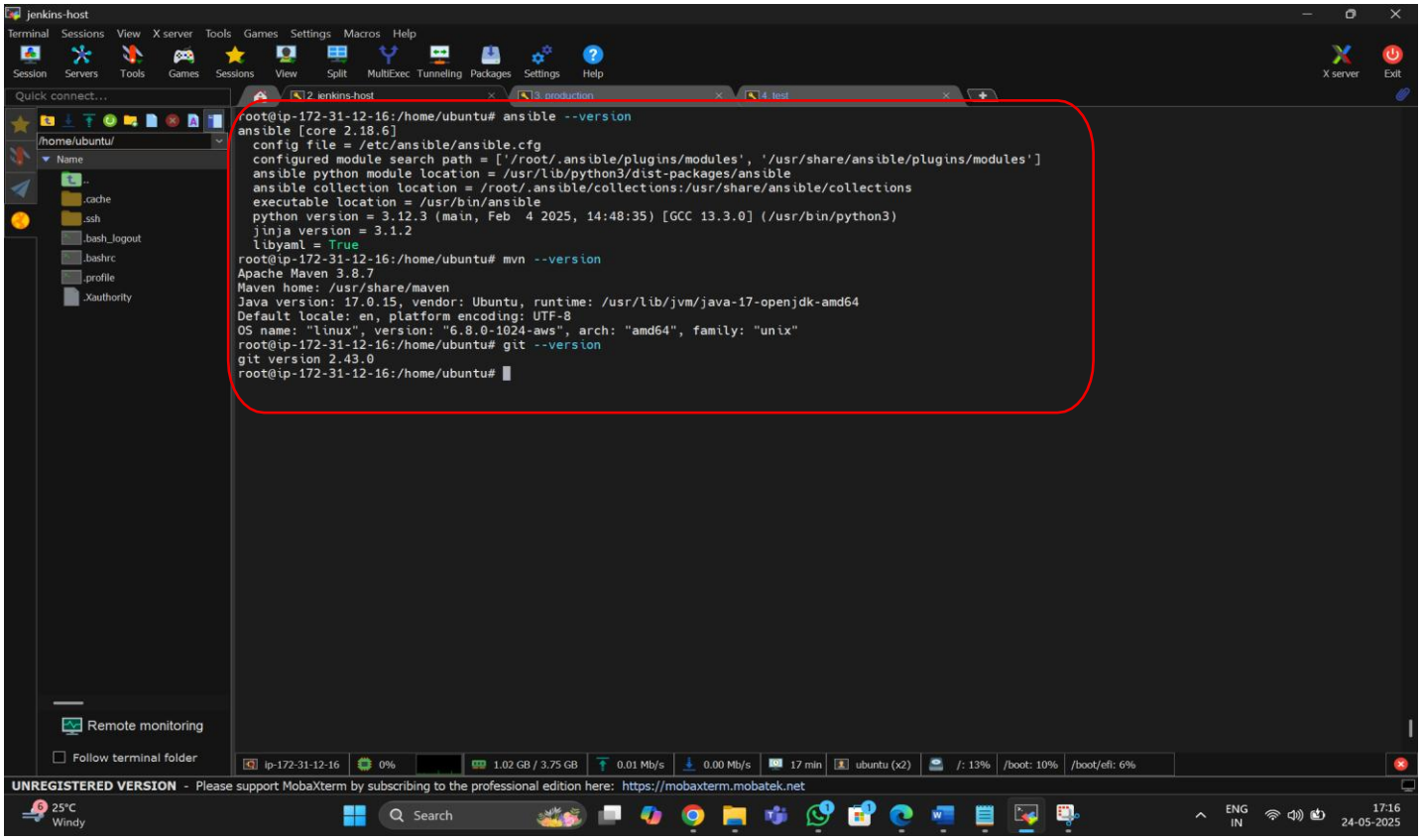
Jenkins-master machine will consist of Jenkins host and also ansible controller



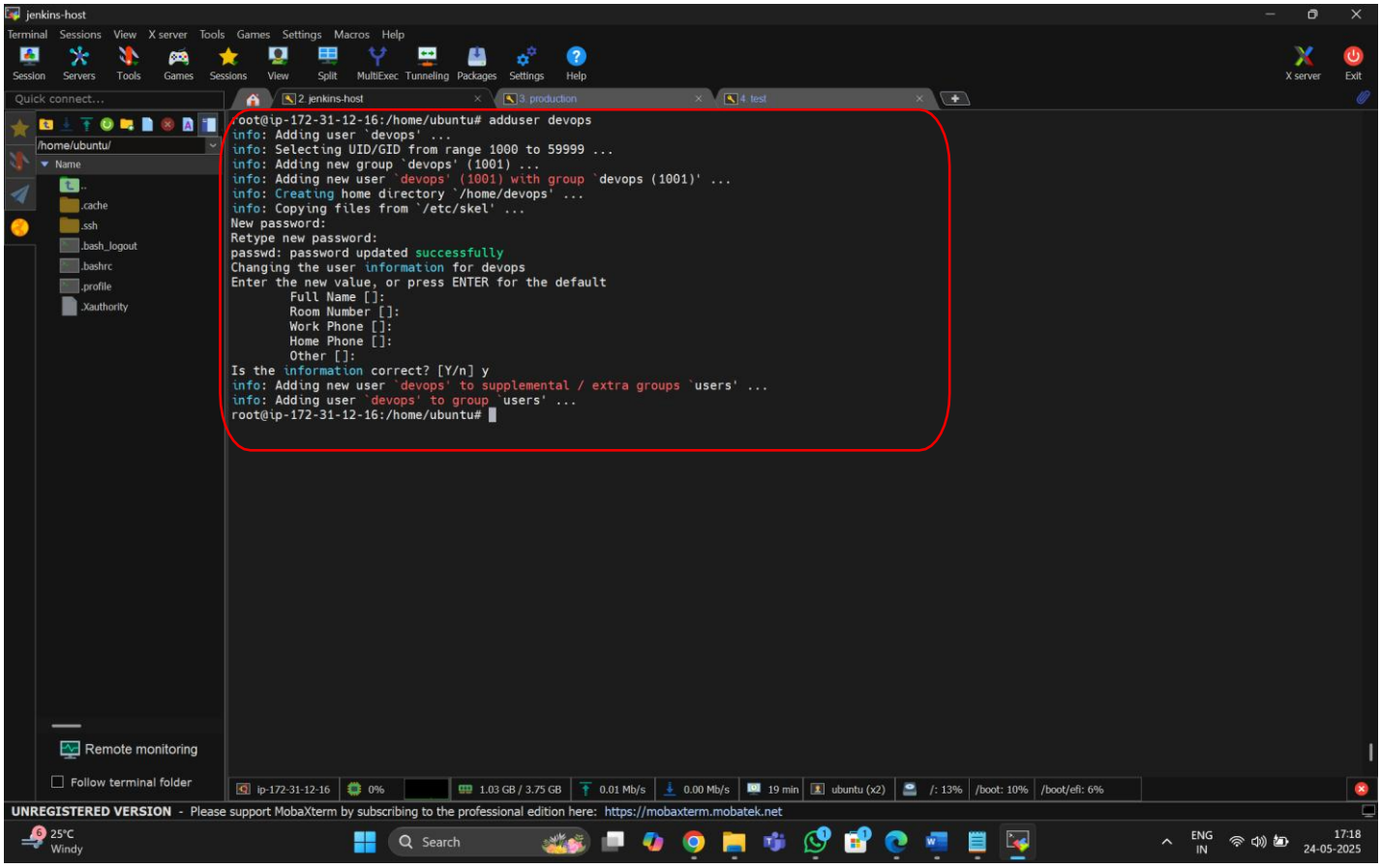
Open all the instances in mobaxterm . and Install all the required packages like Java , Jenkins and also start the Jenkins and check if the service is running in the master machine



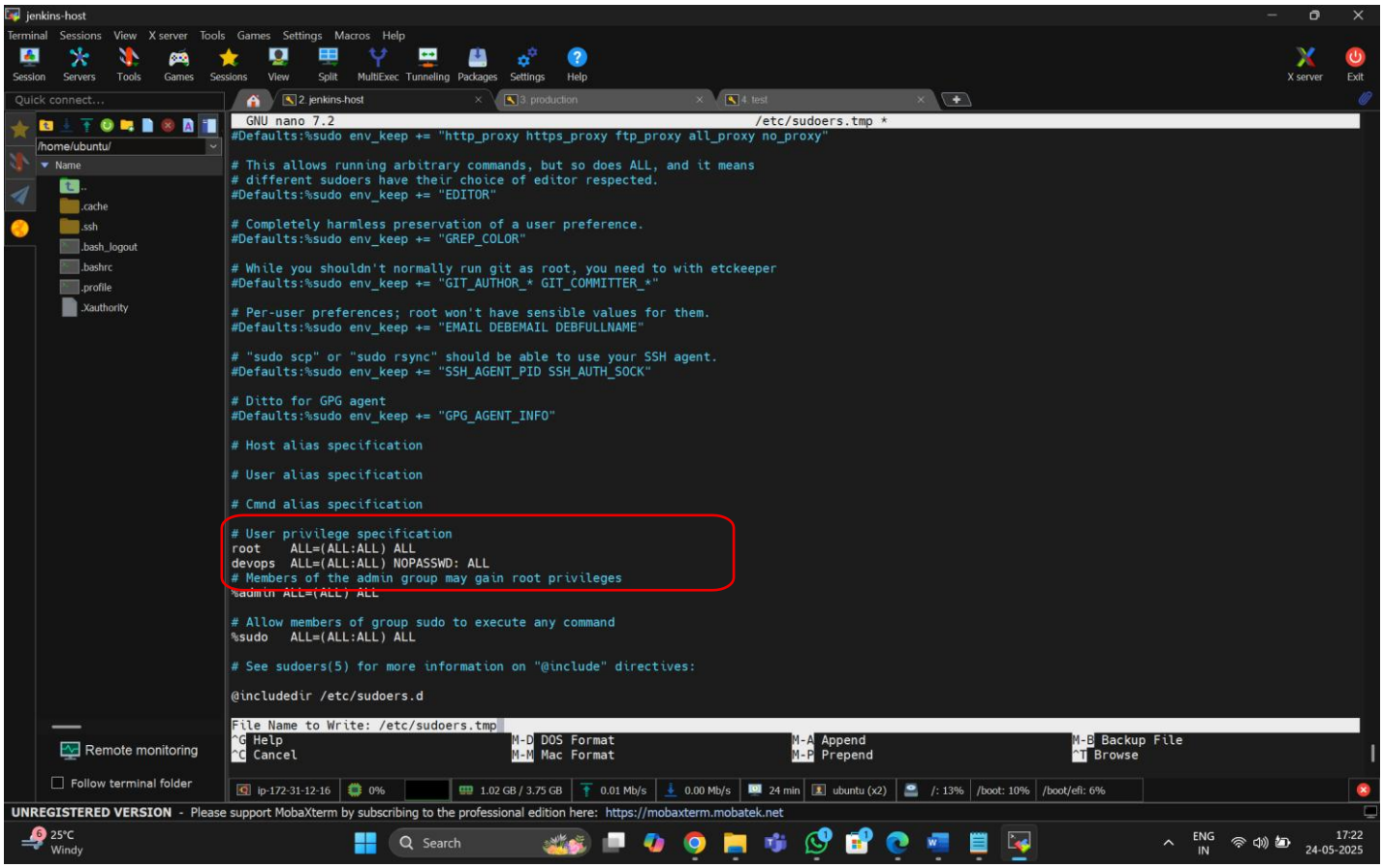
Install maven , git ,ansible



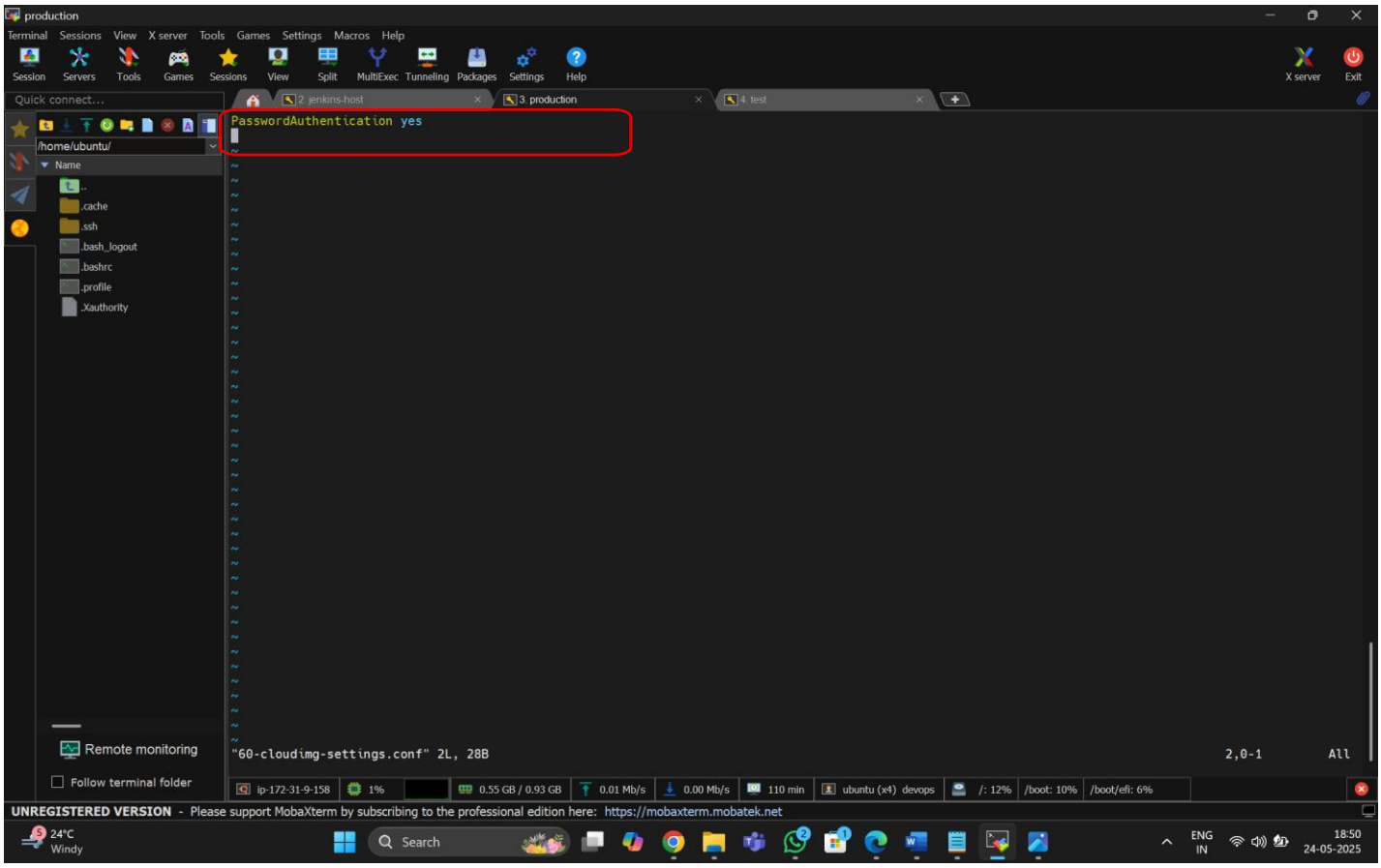
Now perform ansible configuration from controller node to the other two nodes (i.e production and test) also add the user in other two nodes also



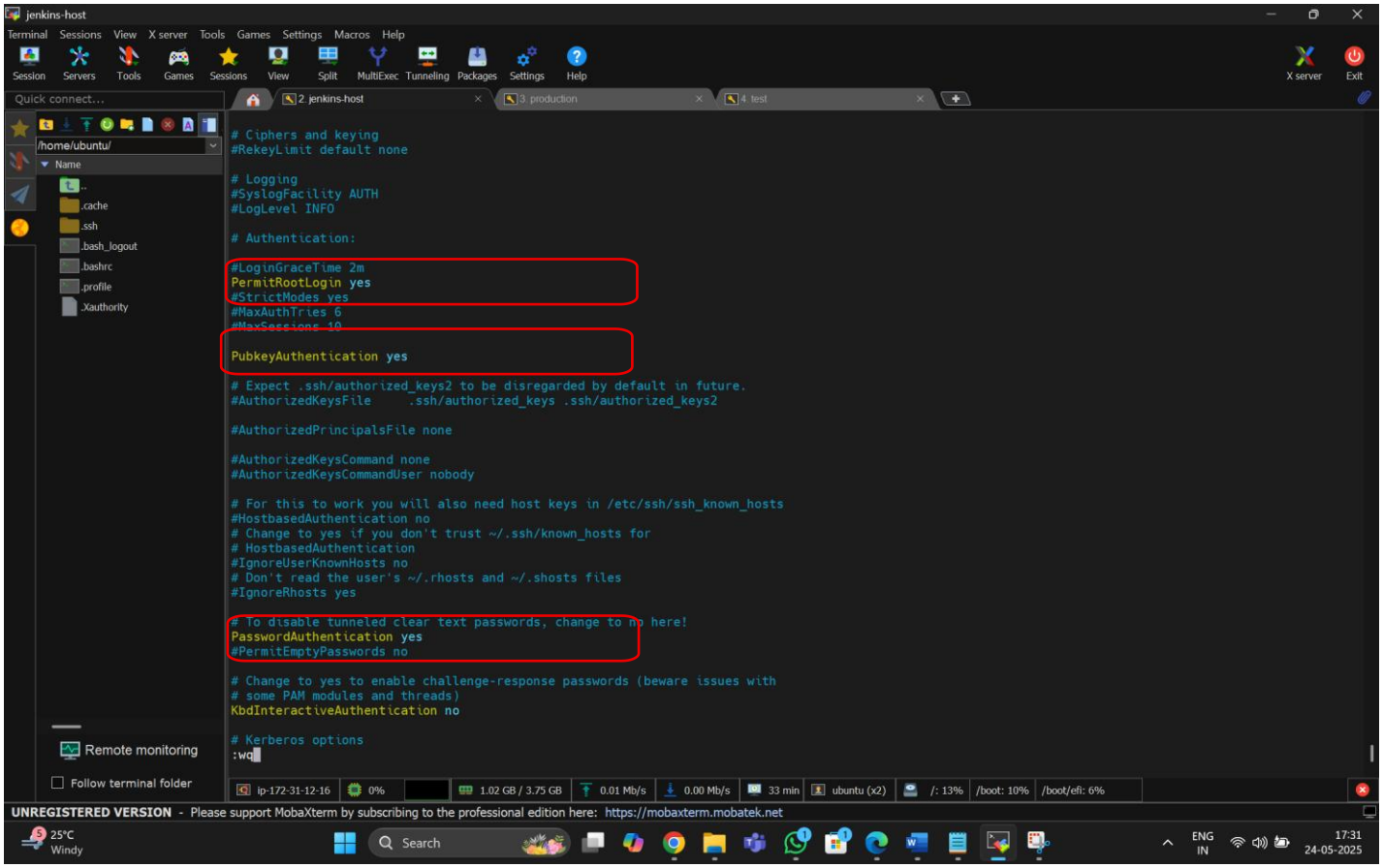
Now add devops uder to sudoers file permission in master as well as other nodes also



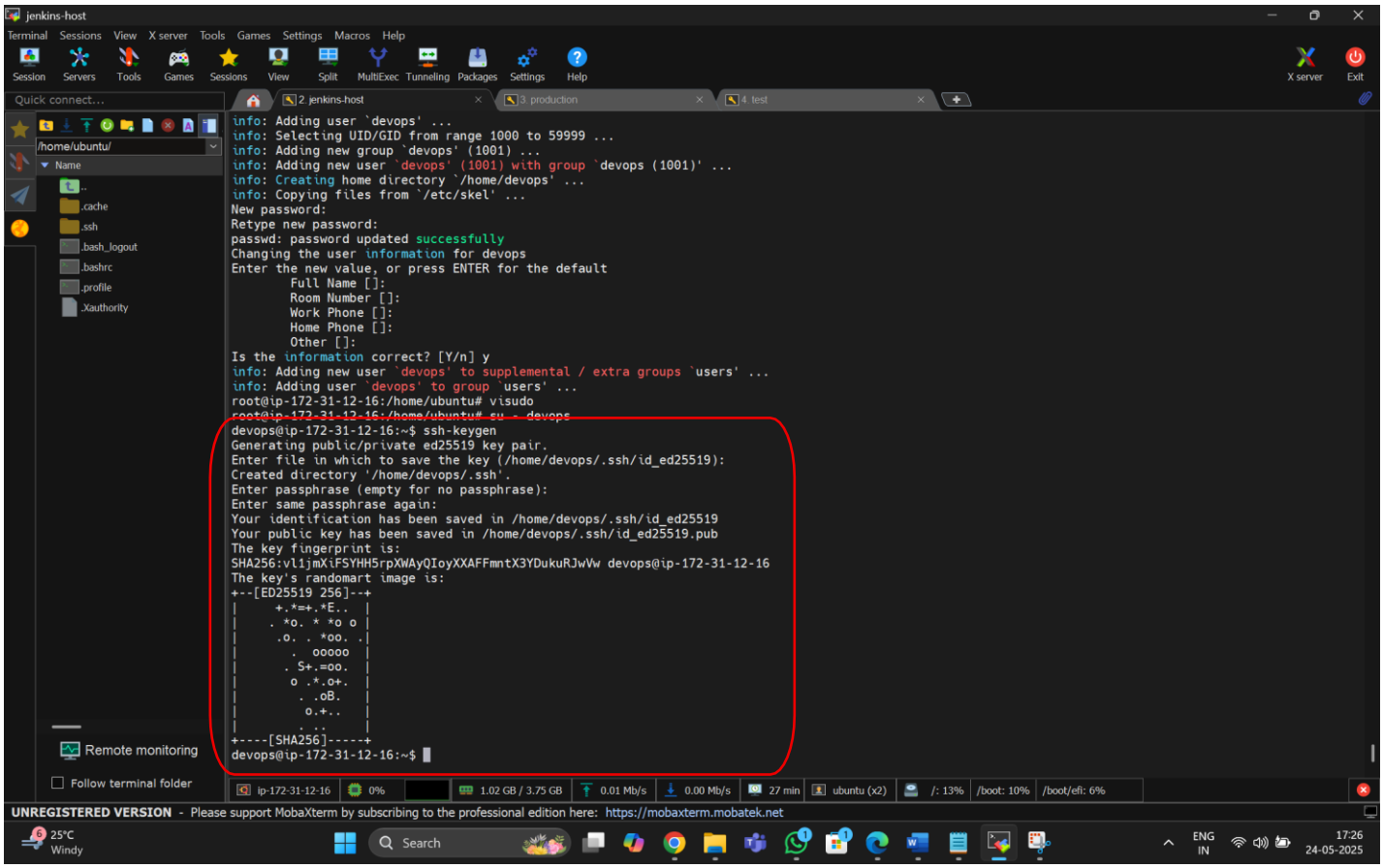
Now configure ssdd file in master as well as in other nodes in the ssh-config file which can be accessed using vi /etc/ssh/sshd_config.d/60-cloudimg-settings.conf



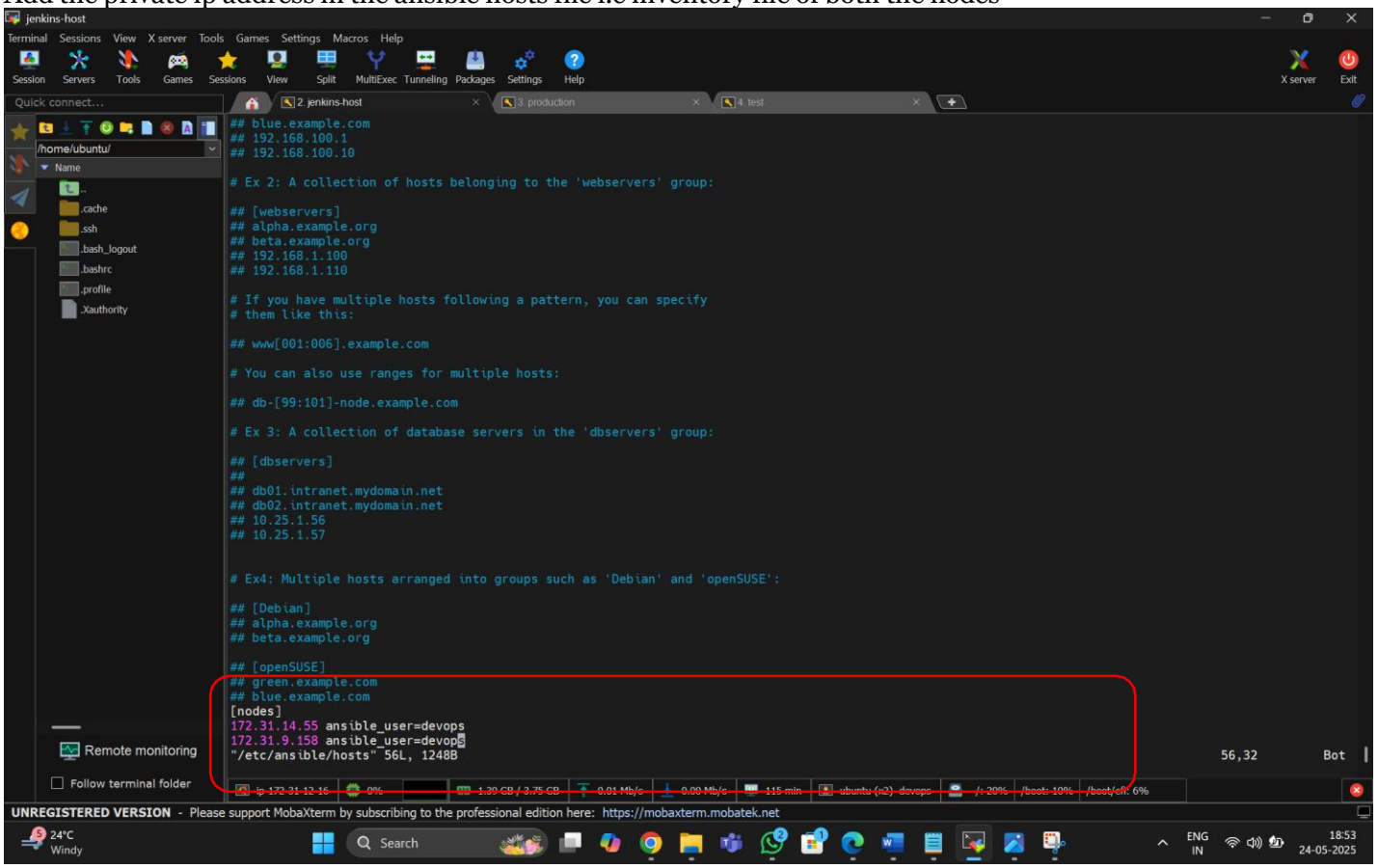
also change the sshd_configuration file using “vi /etc/ssh/sshd_config” permitlogin and pubkeyauthentication to yes and password authentication to yes



Generate the ssh key from the devops user using command ssh-keygen

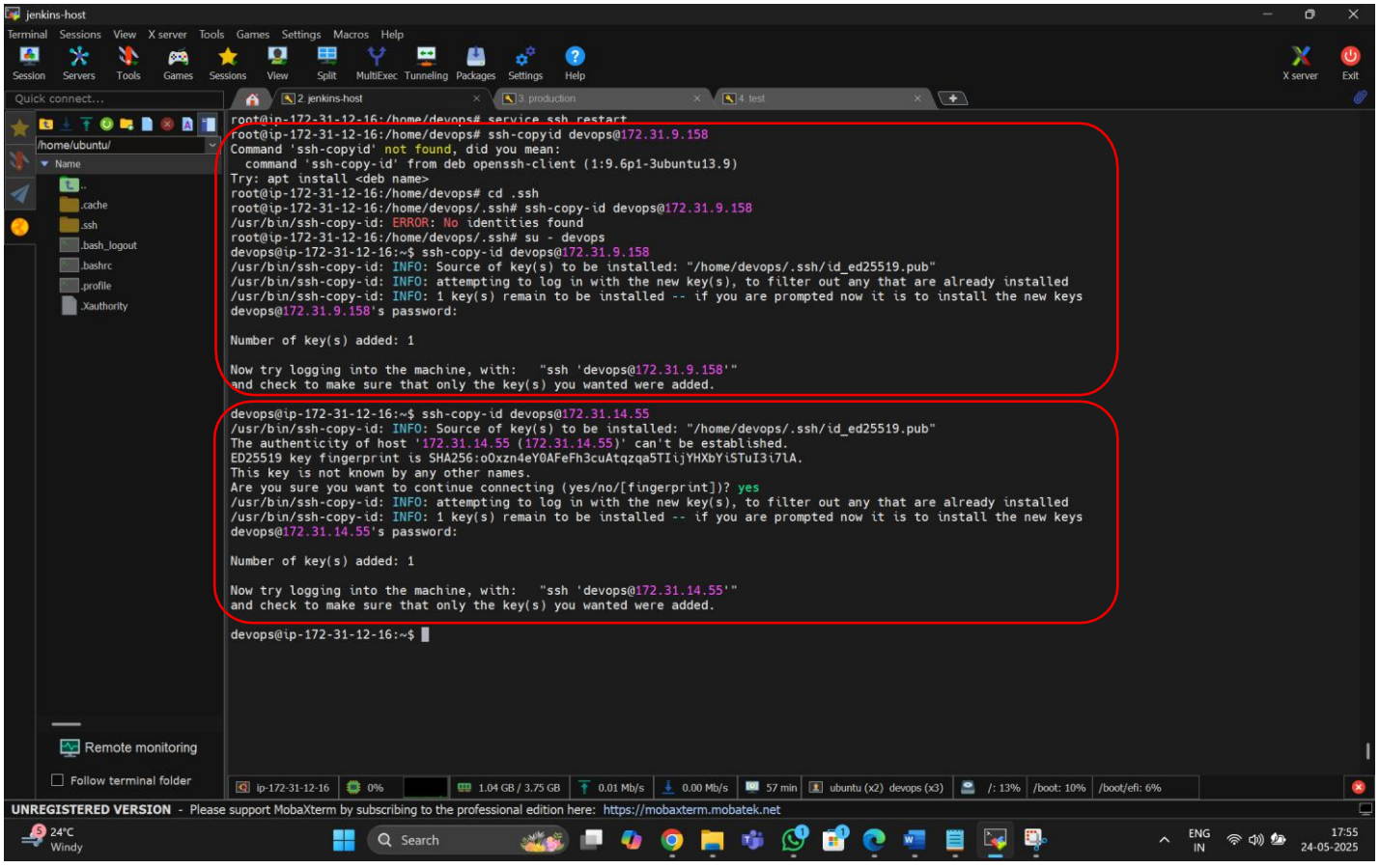


Add the private ip address in the ansible hosts file i.e inventory file of both the nodes

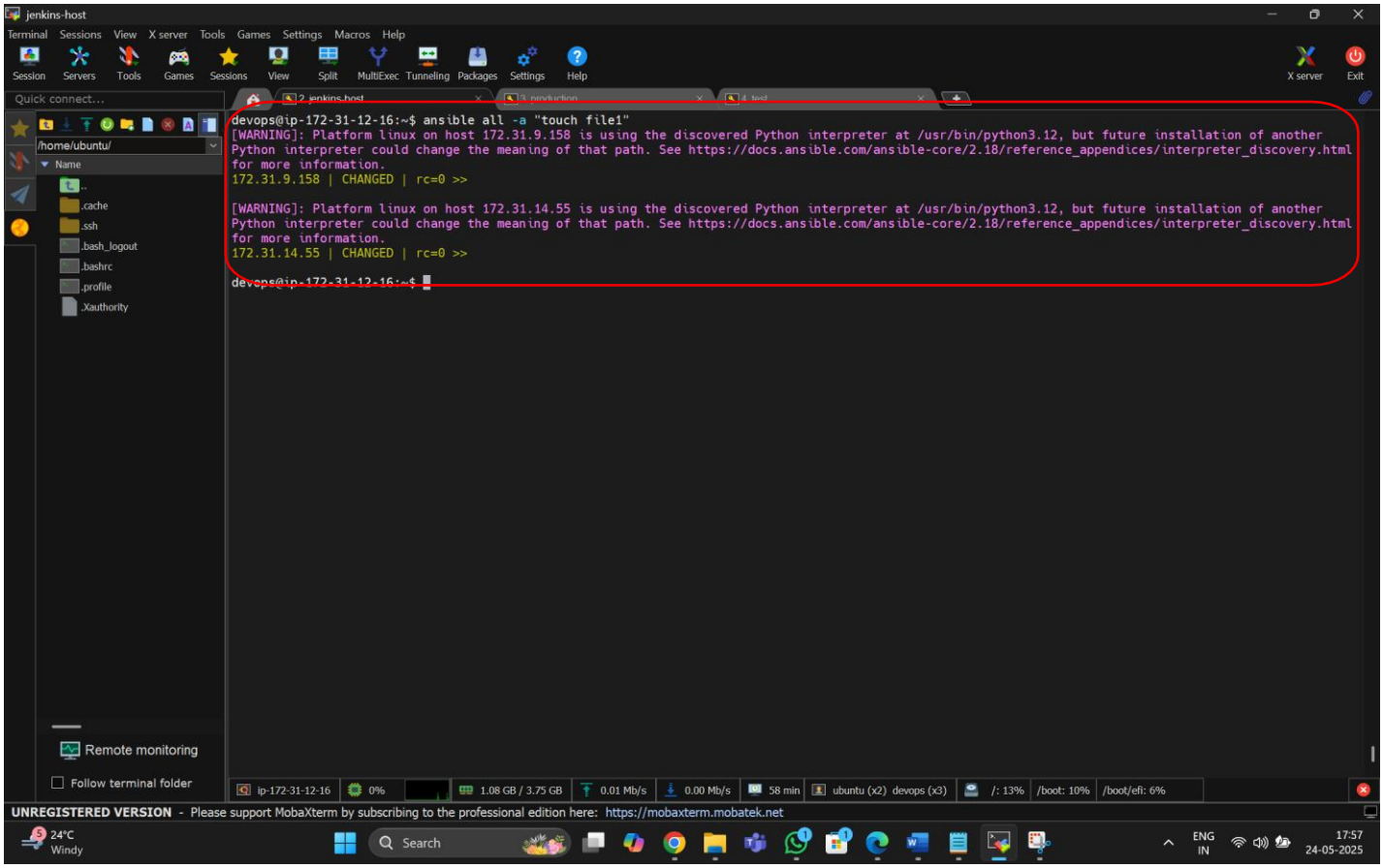


Copy the ssh key into the nodes and check if the nodes are connected using the ssh devops@<private_ip>

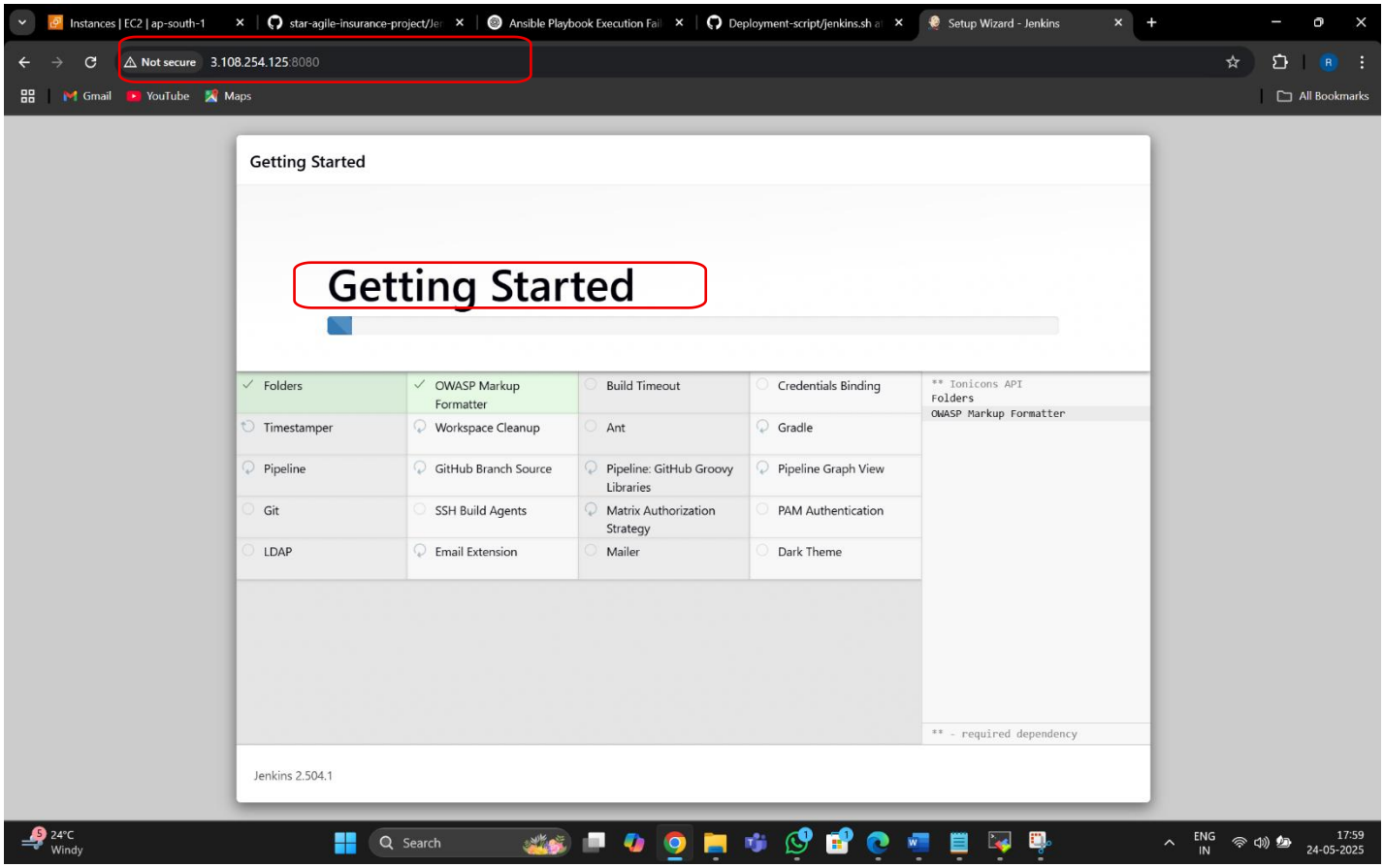
If it gets into the machine then the machine is configures to the controller



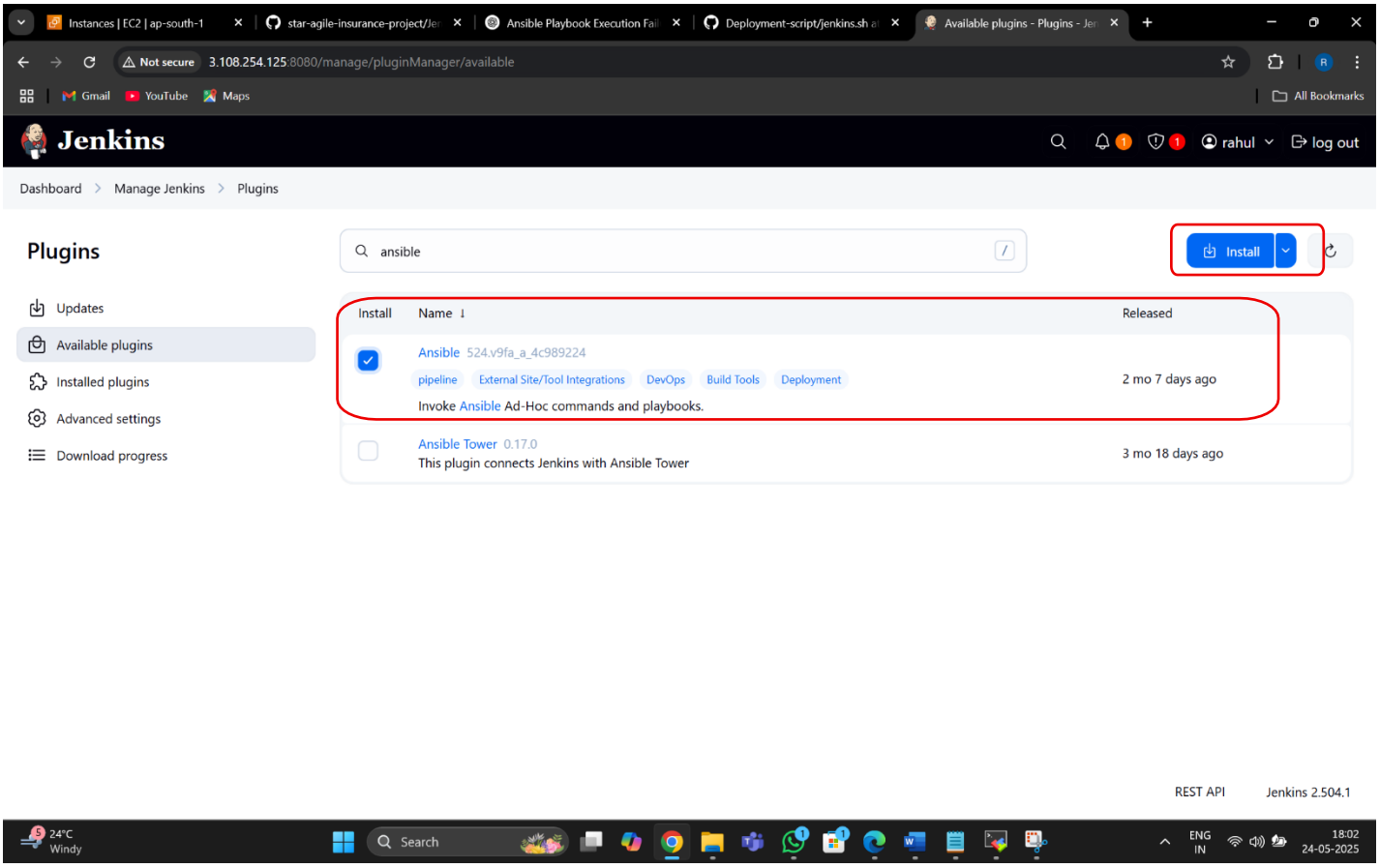
To verify the the connection run the ansible command “ansible all -a “touch file1” and check whether the changes happened to the nodes



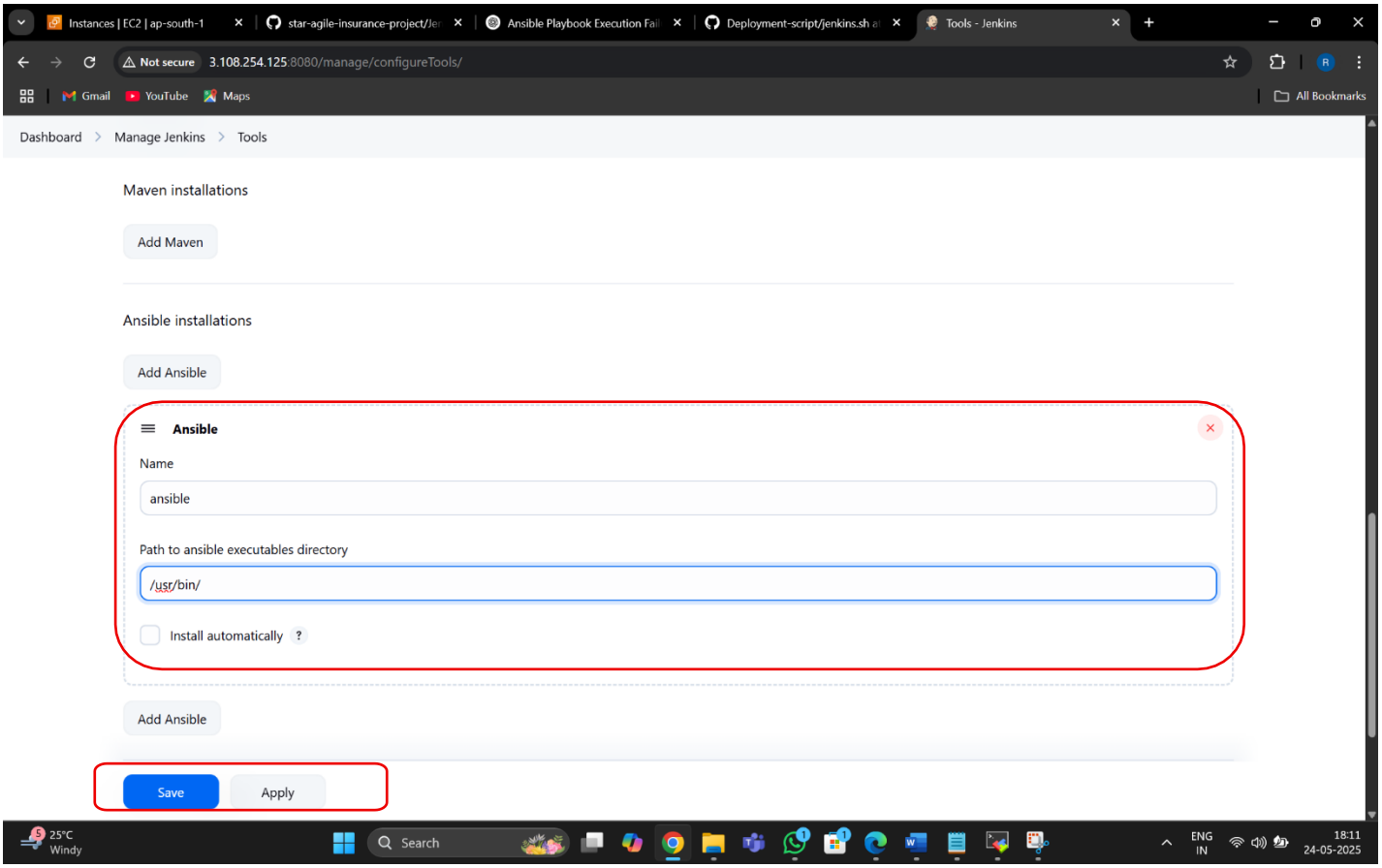
Access the Jenkins using the private ip address of the master machine on port 8080



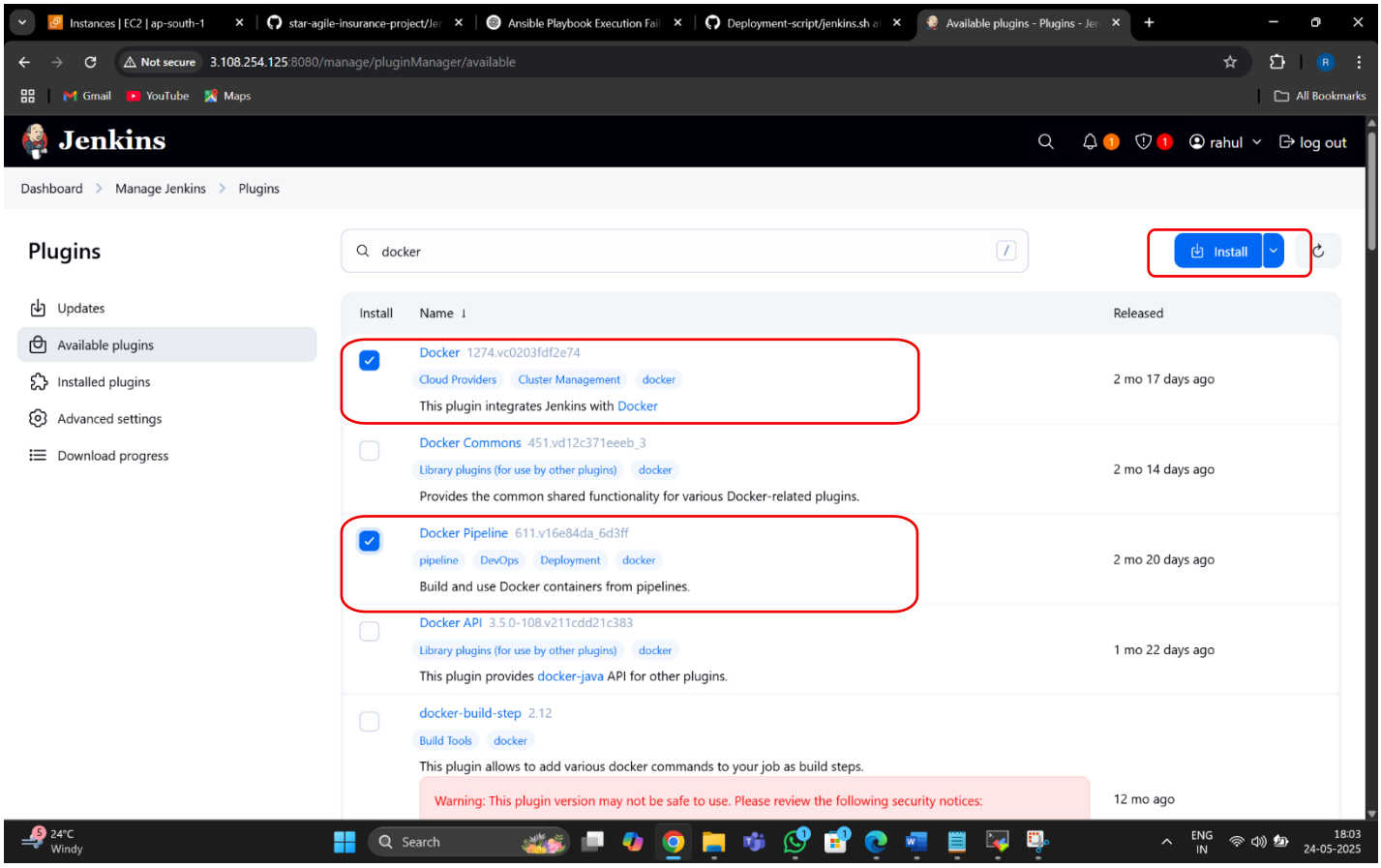
Navigate to the manage Jenkins -> plugins -> available plugins -> install ansible



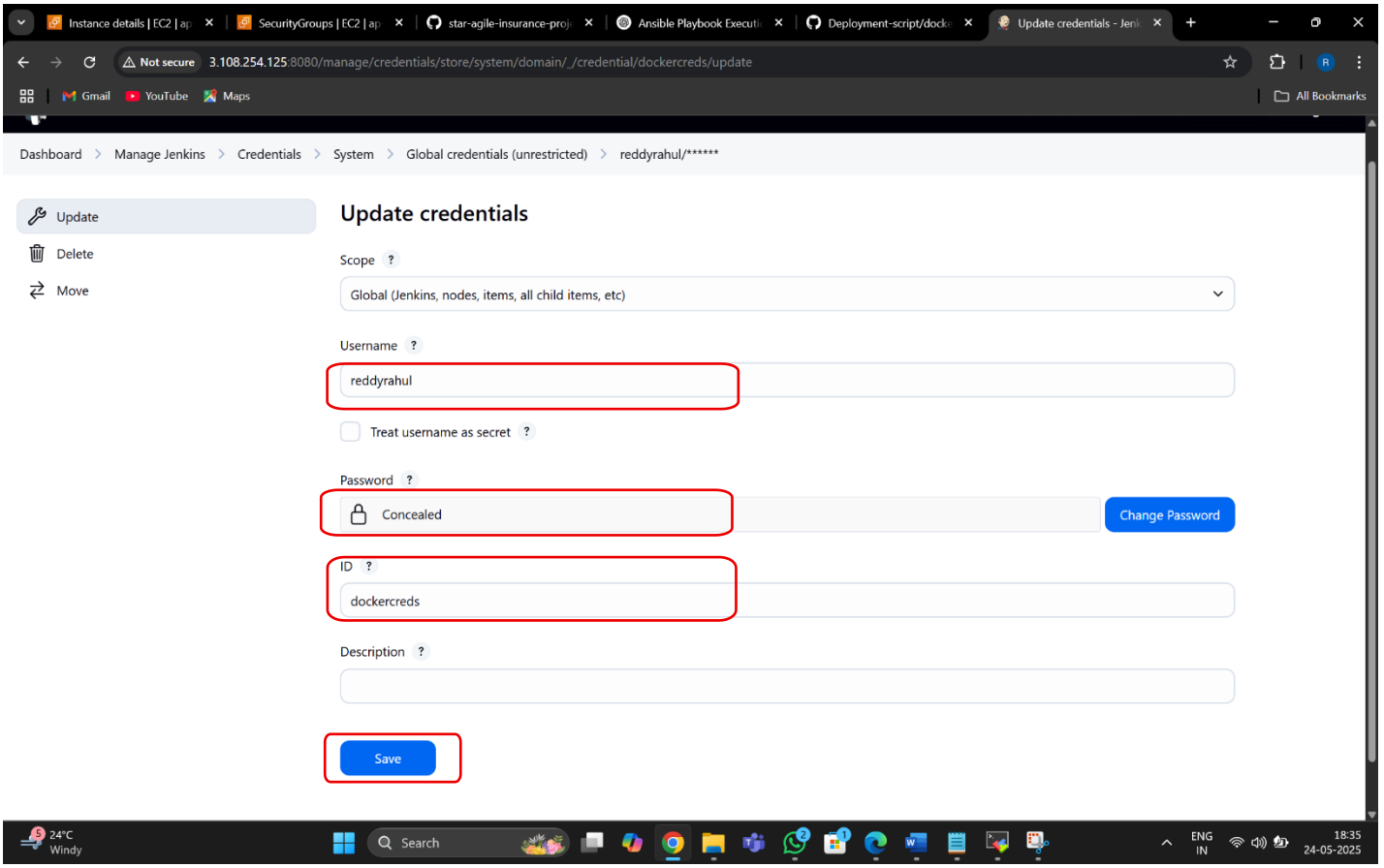
Do ansible configuration and set the path in tools configuration



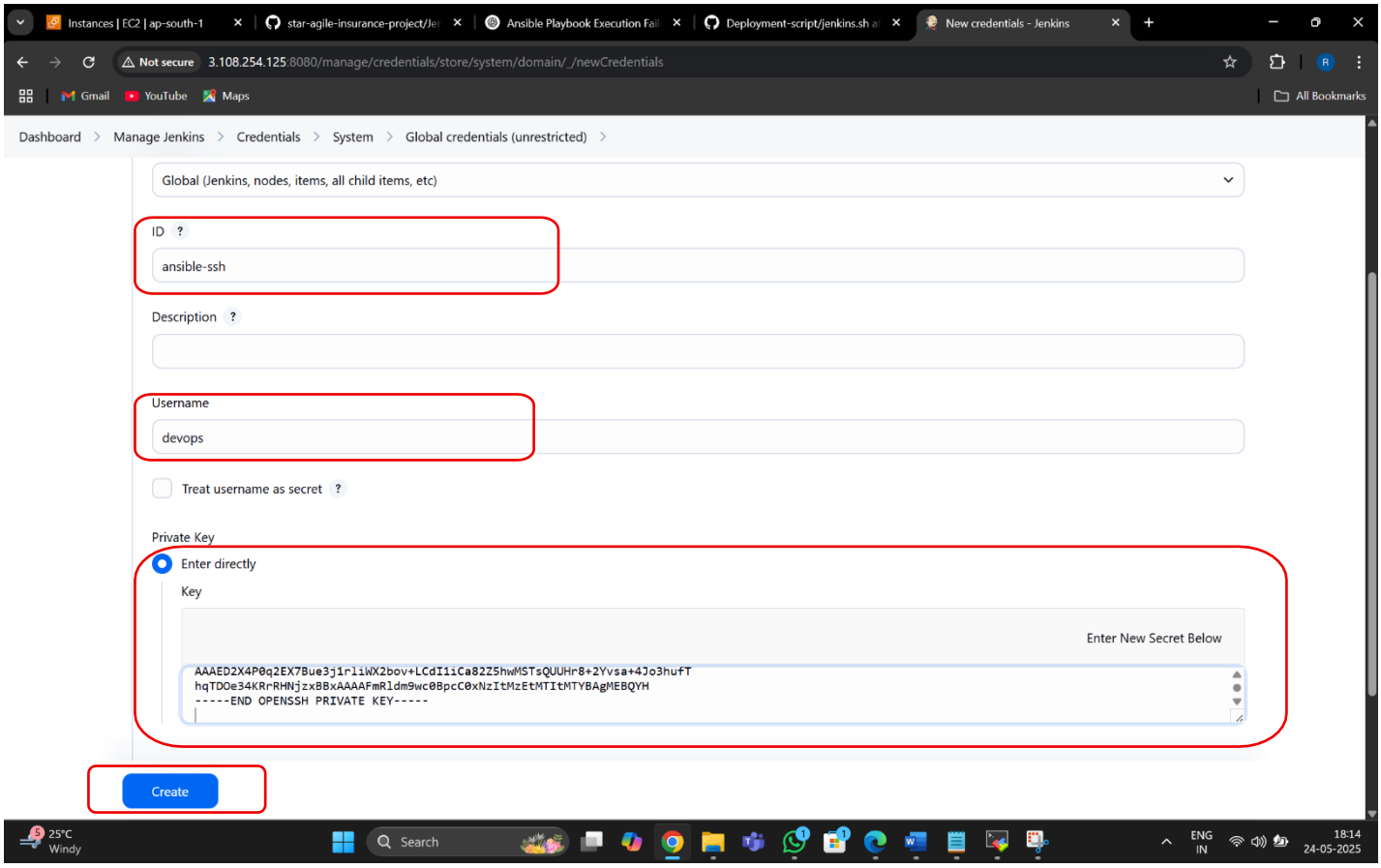
Install docker plugins



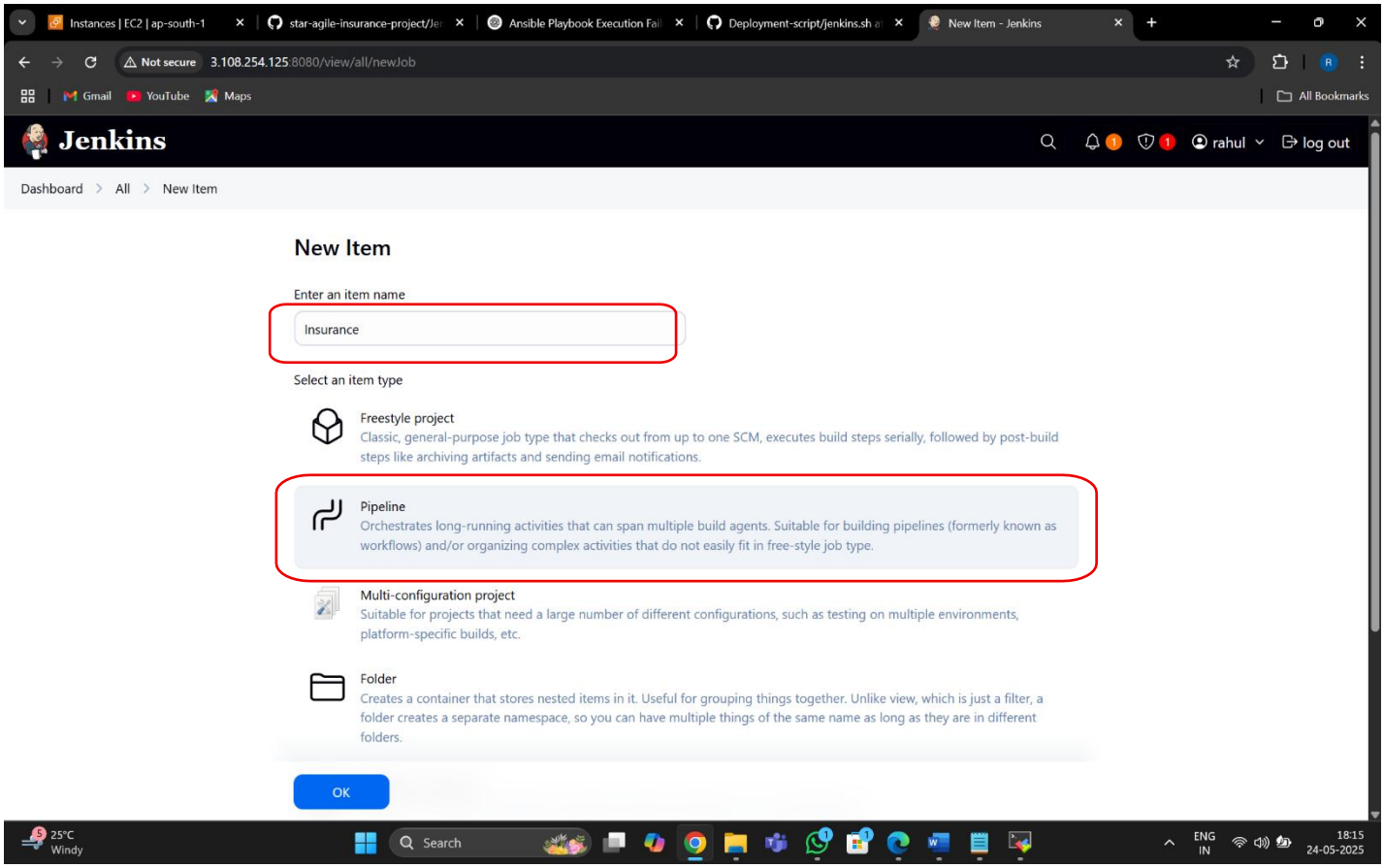
Add the docker hub credentials provide username and password
Provide id of docker hub as dockercreds



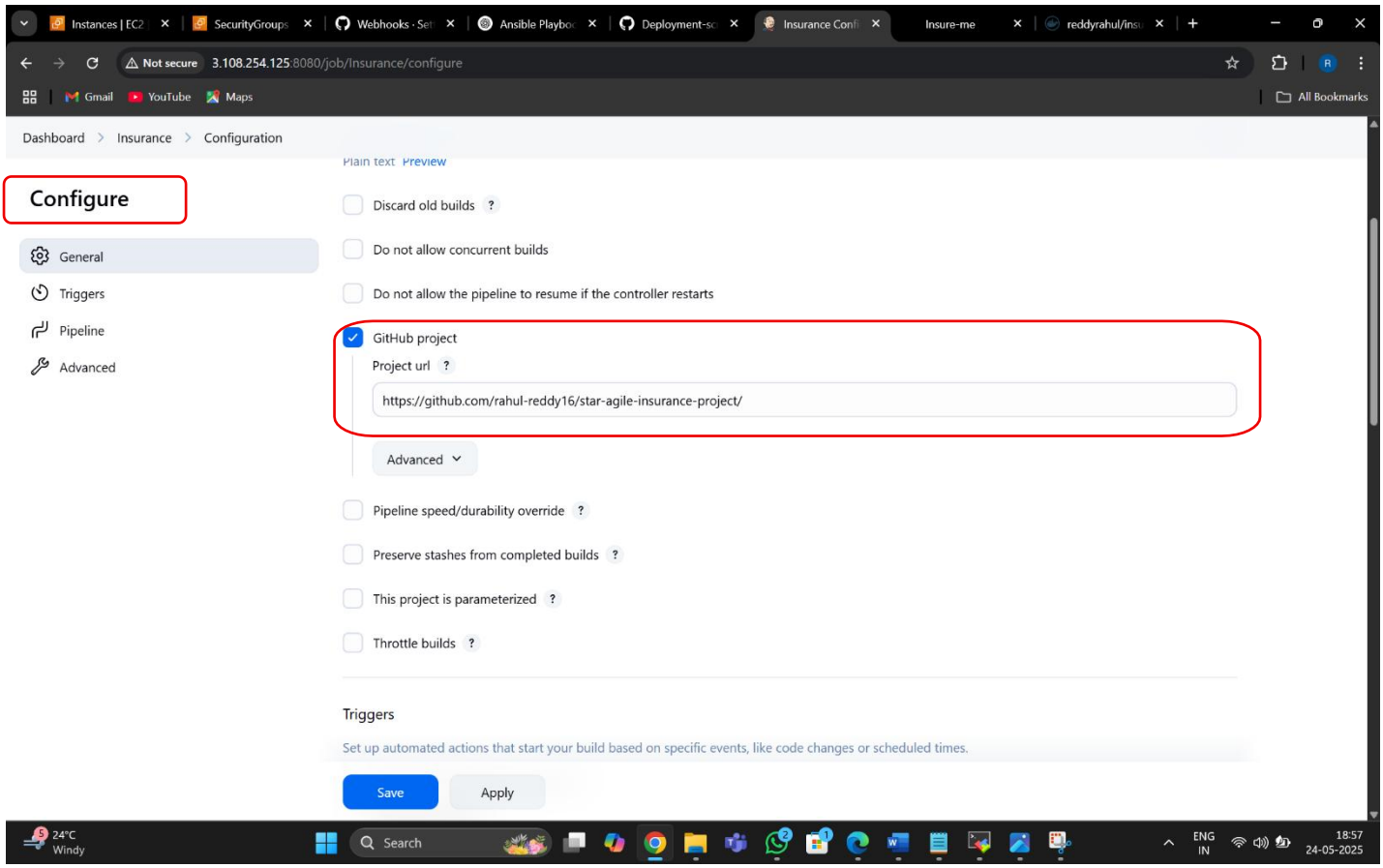
Add the global credentials of the ansible ssh key in the Jenkins
Paste the secret key into the private key section



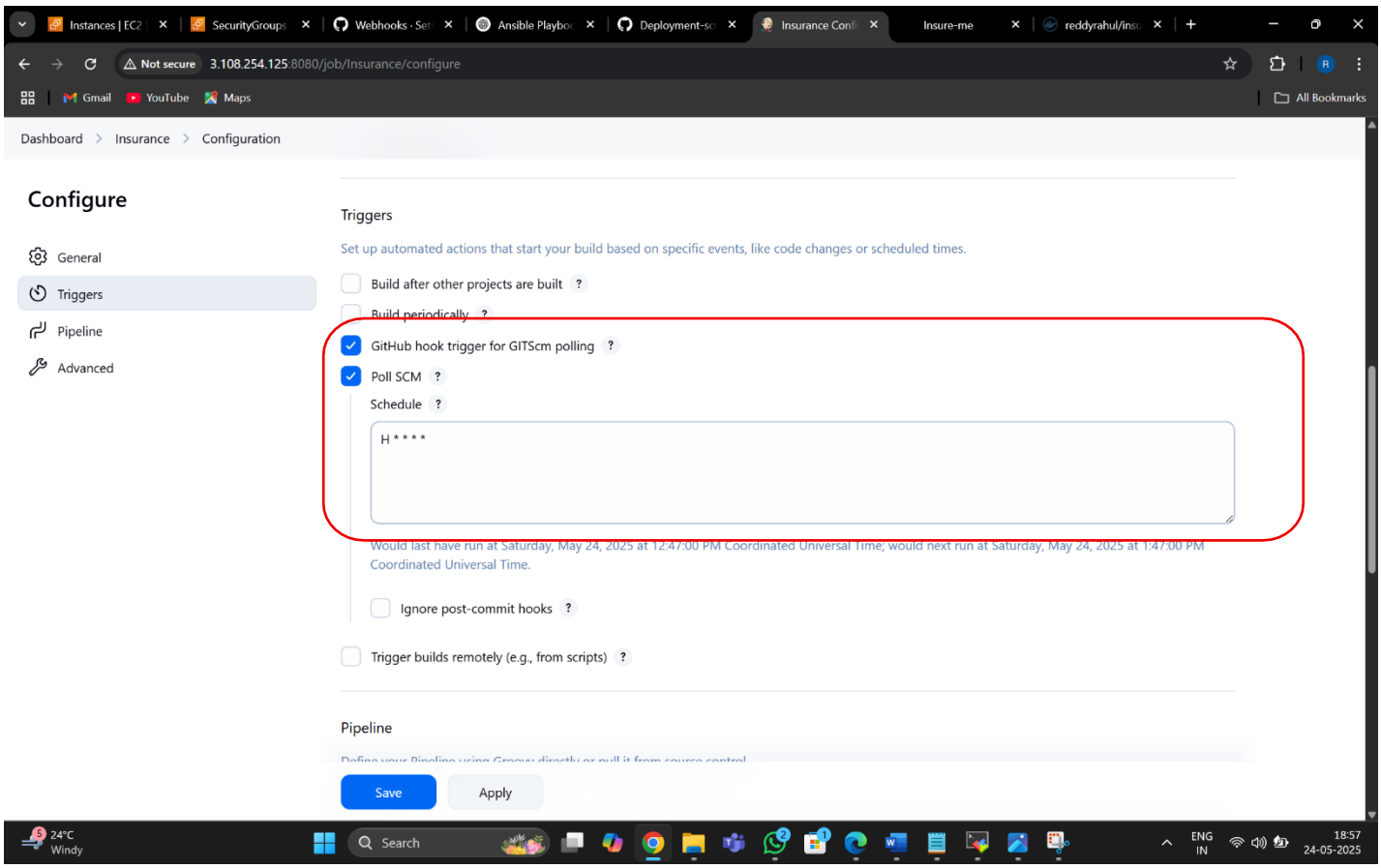
Create a new item and enter the name of the project and select the item type to pipeline



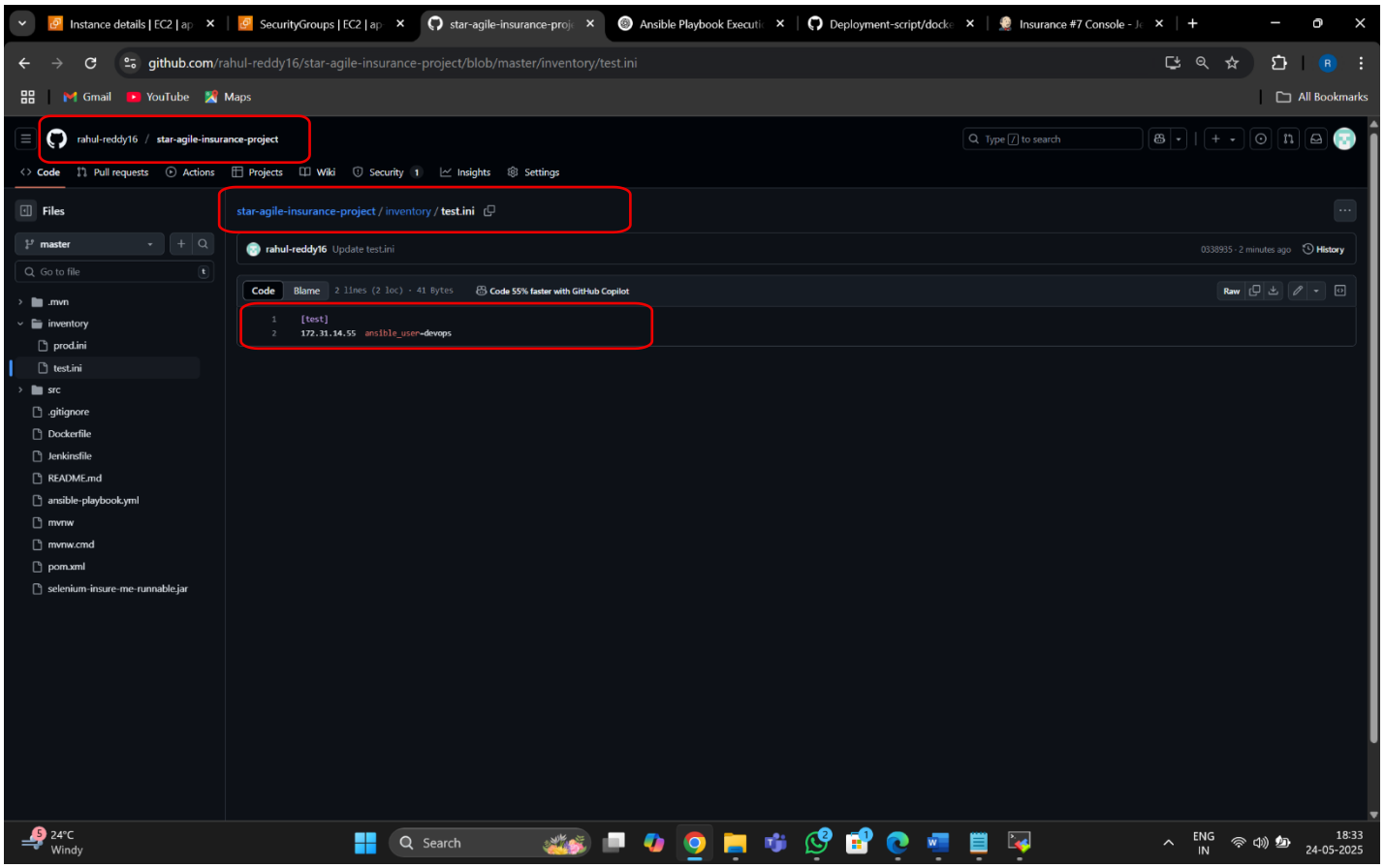
Configure the project with github project repository

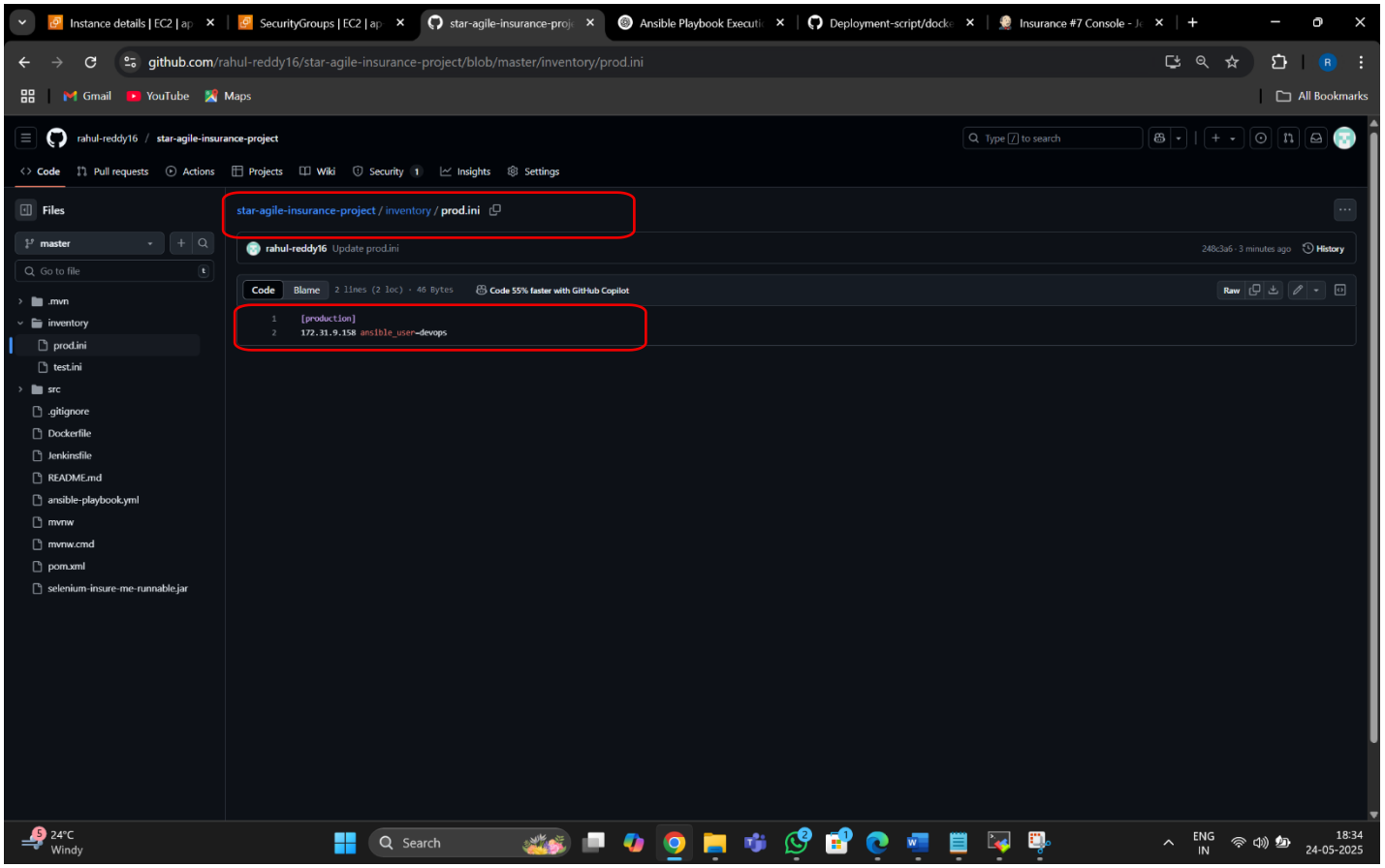


add the triggers

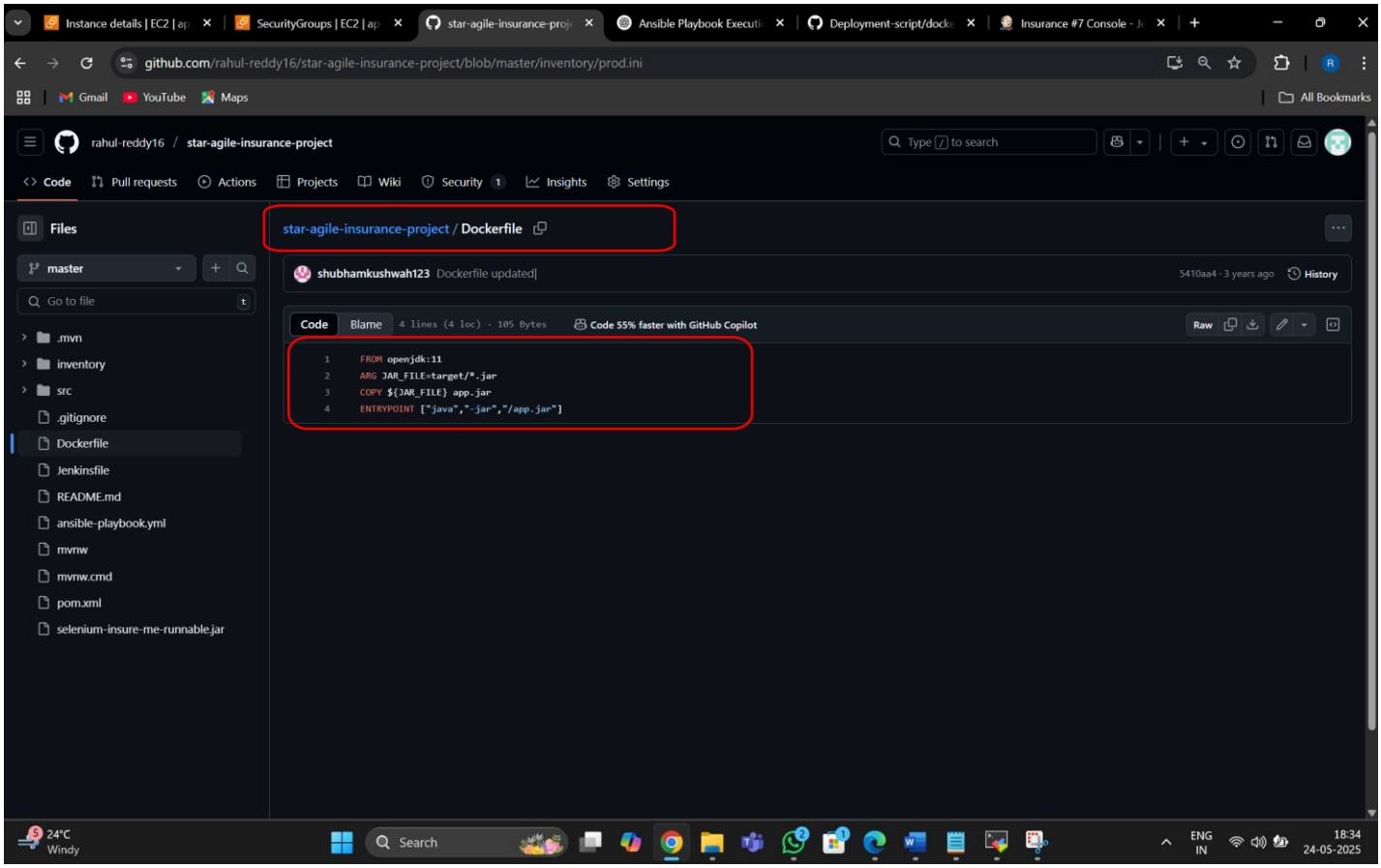


Add the inventory file in the github repo for the purpose of proof of concept I have added the private key in to the inventory file with test.ini and prod.ini with there respective private ip's

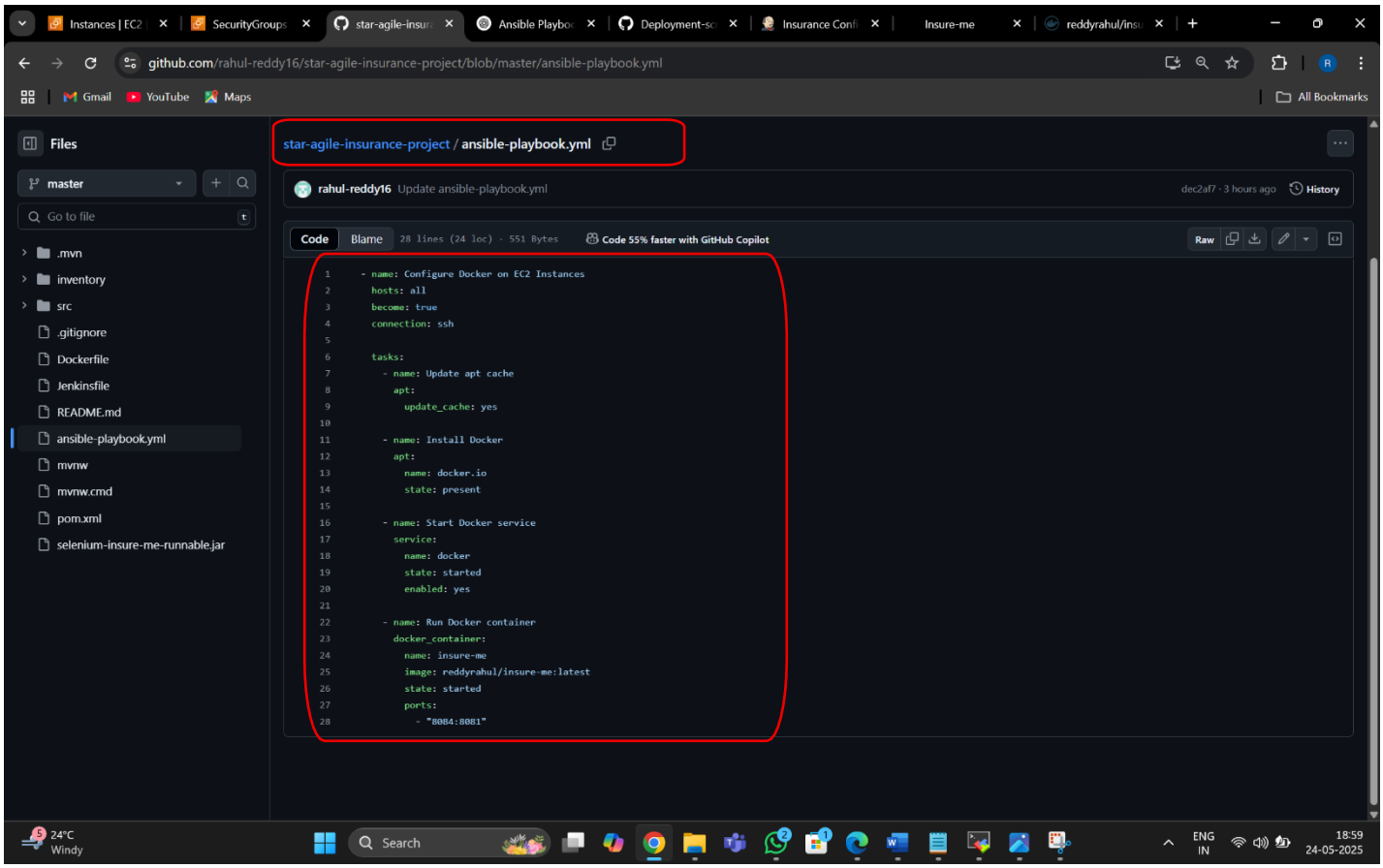




Now write the Dockerfile



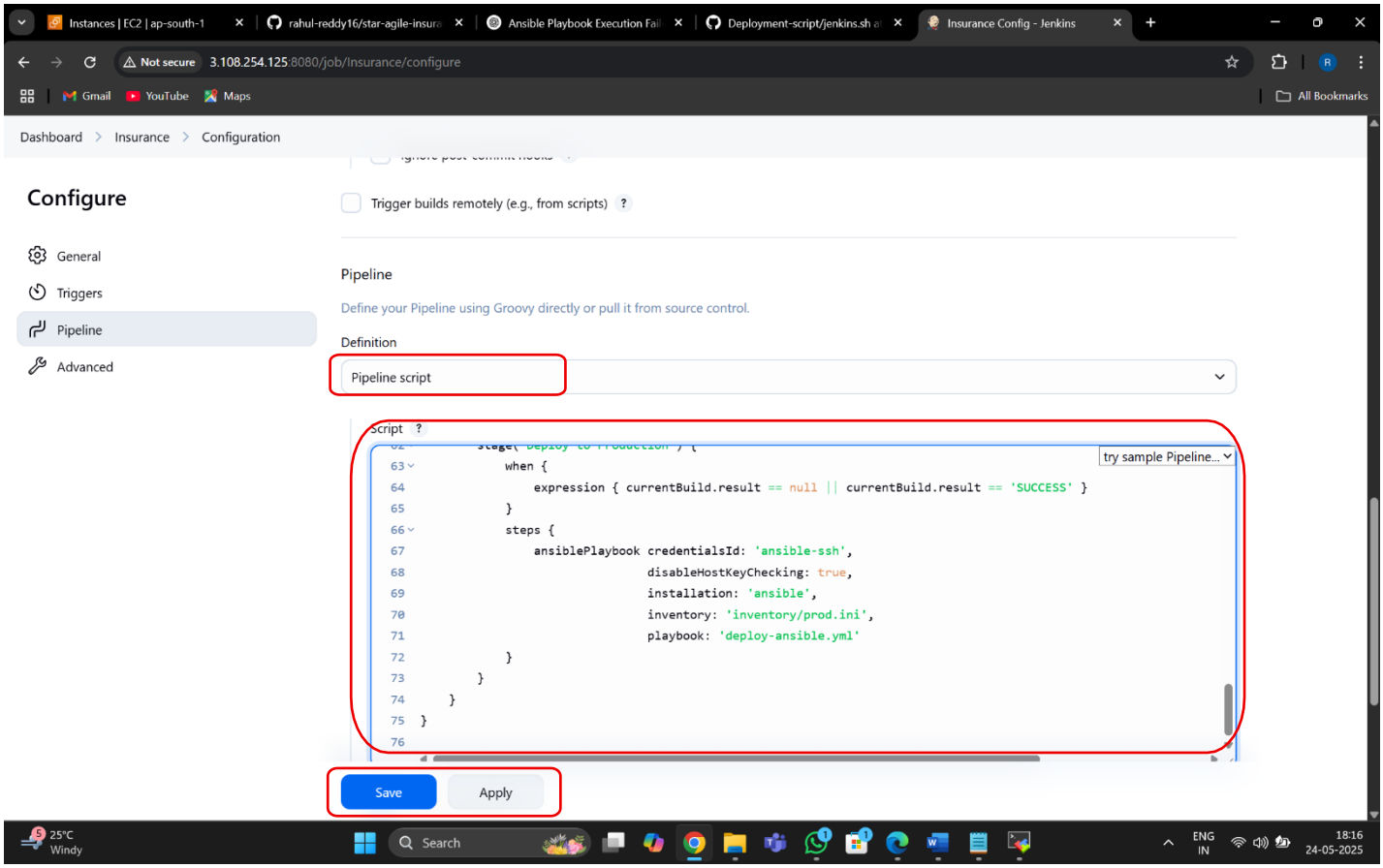
Also write the ansible playbook to deploy the containers



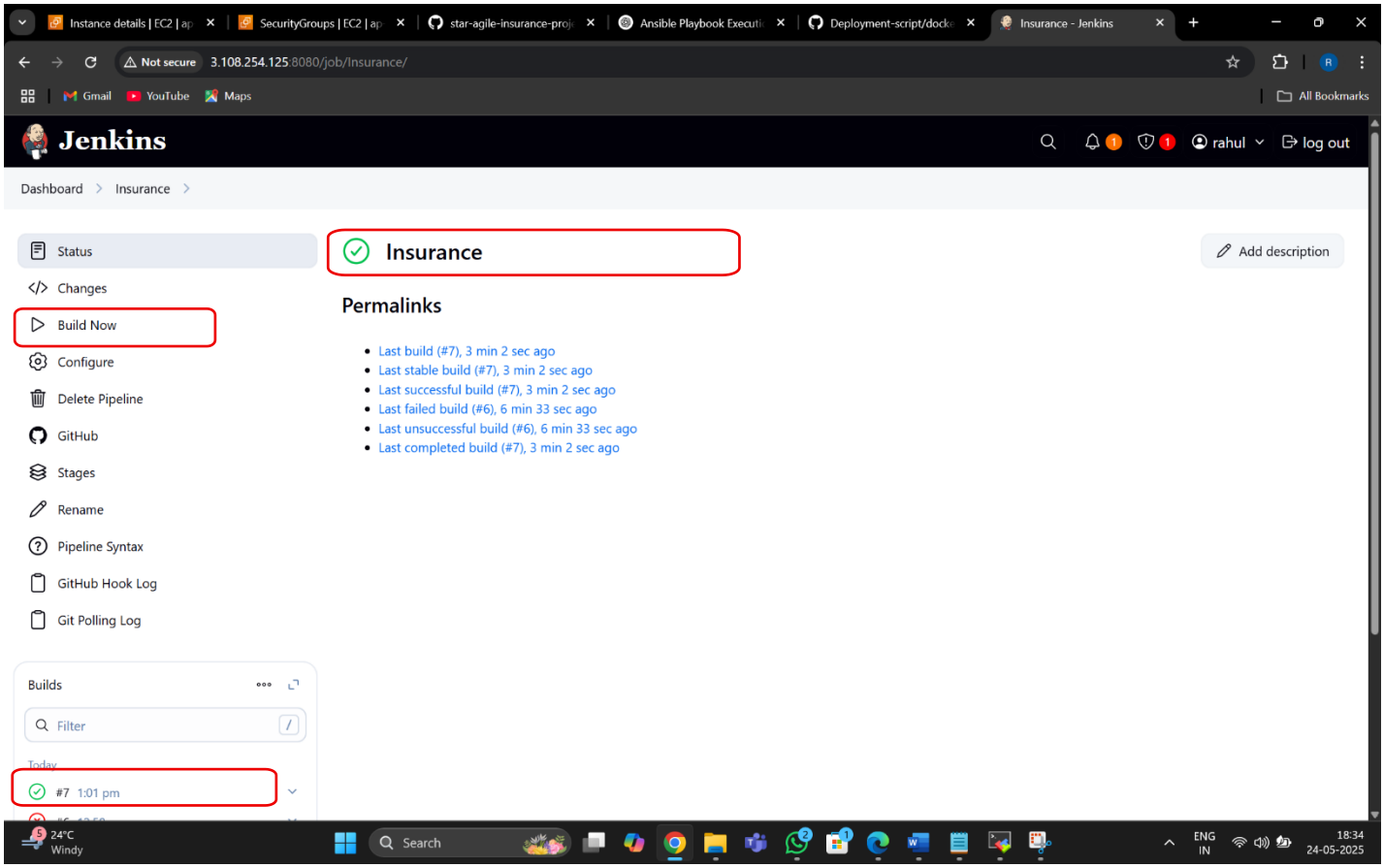
Build the project in jenkins

```
pipeline {
    agent any

    stages {
        stage('Checkout Code') {
            steps {
                git url: 'https://github.com/rahul-reddy16/star-agile-insurance-project.git', branch: 'master'
            }
        }

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

        stage('Build Docker Image') {
            steps {
                sh 'docker build -t reddyrahul/insure-me:latest .'
            }
        }

        stage('Push Docker Image') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'dockercreds', usernameVariable: 'USERNAME', passwordVariable:
'PASSWORD')]) {
                    sh 'echo $PASSWORD | docker login -u $USERNAME --password-stdin'
                    sh 'docker push reddyrahul/insure-me:latest'
                }
            }
        }

        stage('Deploy to Test Using Ansible') {
            steps {
                ansiblePlaybook credentialsId: 'ansible-ssh',
                                disableHostKeyChecking: true,
                                installation: 'ansible',
                                inventory: 'inventory/test.ini',
                                playbook: 'ansible-playbook.yml'
            }
        }

        stage('Wait for Test App to Be Ready') {
            steps {
                sh 'sleep 6'
            }
        }

        stage('Run Selenium Tests on Test machine') {
            steps {
                sh 'mvn test'
            }
        }

        stage('Check Build Status') {
            steps {
                script {
                    echo "Current build result: ${currentBuild.result}"
                }
            }
        }

        stage('Deploy to Production') {
            when {
                expression { currentBuild.result == null || currentBuild.result == 'SUCCESS' }
            }
            steps {
                ansiblePlaybook credentialsId: 'ansible-ssh',
                                disableHostKeyChecking: true,
                                installation: 'ansible',
                                inventory: 'inventory/prod.ini',
                                playbook: 'deploy-ansible.yml'
            }
        }
    }
```

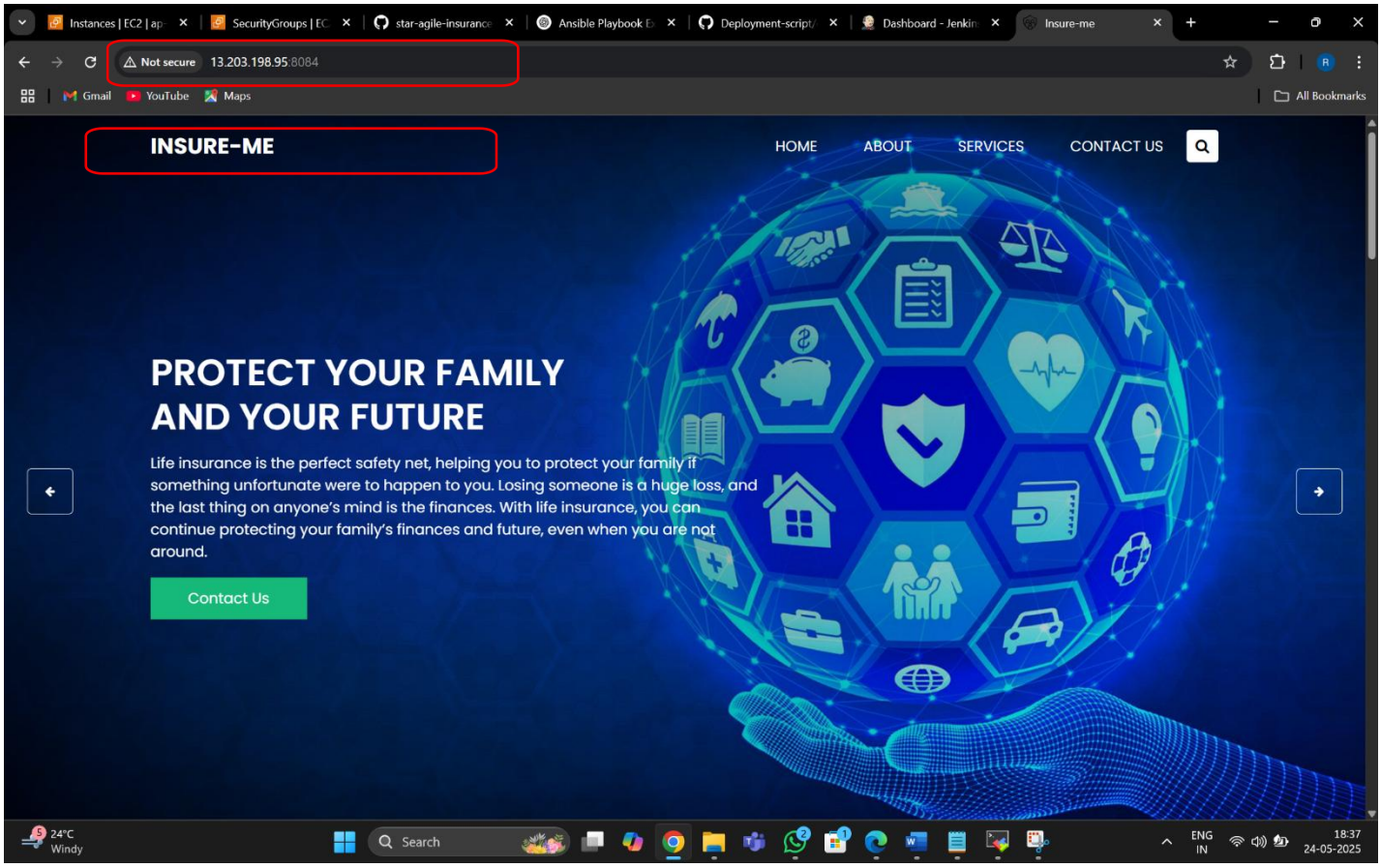
Write the script pipeline in jenkins



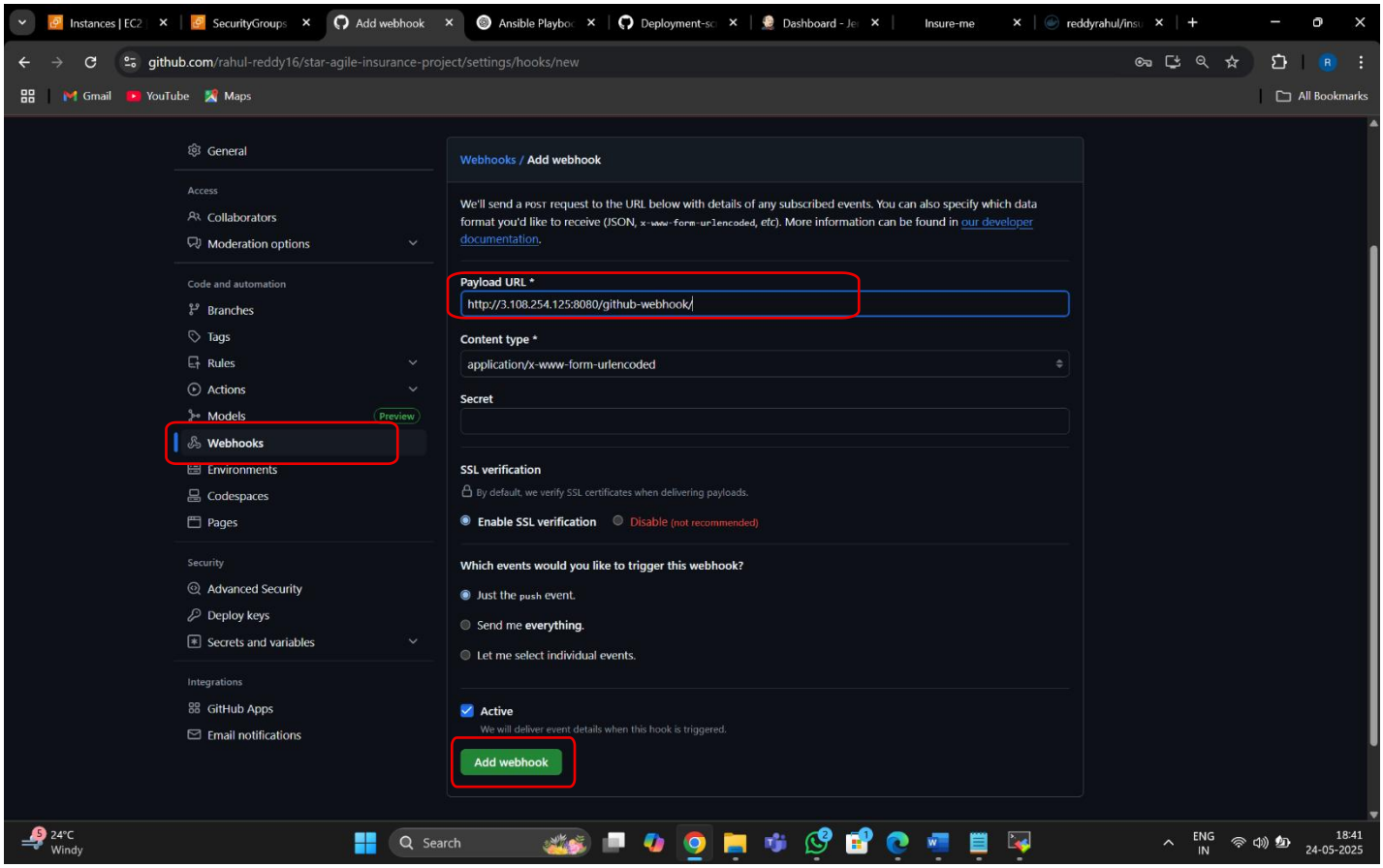
Then save and build



Now access the application using the production ip address on the port 8084



Add the github webhook in the settings of the project repository



Verfying in the docker hub if the image has got created successfully