

Module Name CapstoneProject

Project name

Insurance-Domain

Submitted by :

manjunath kamalapure

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

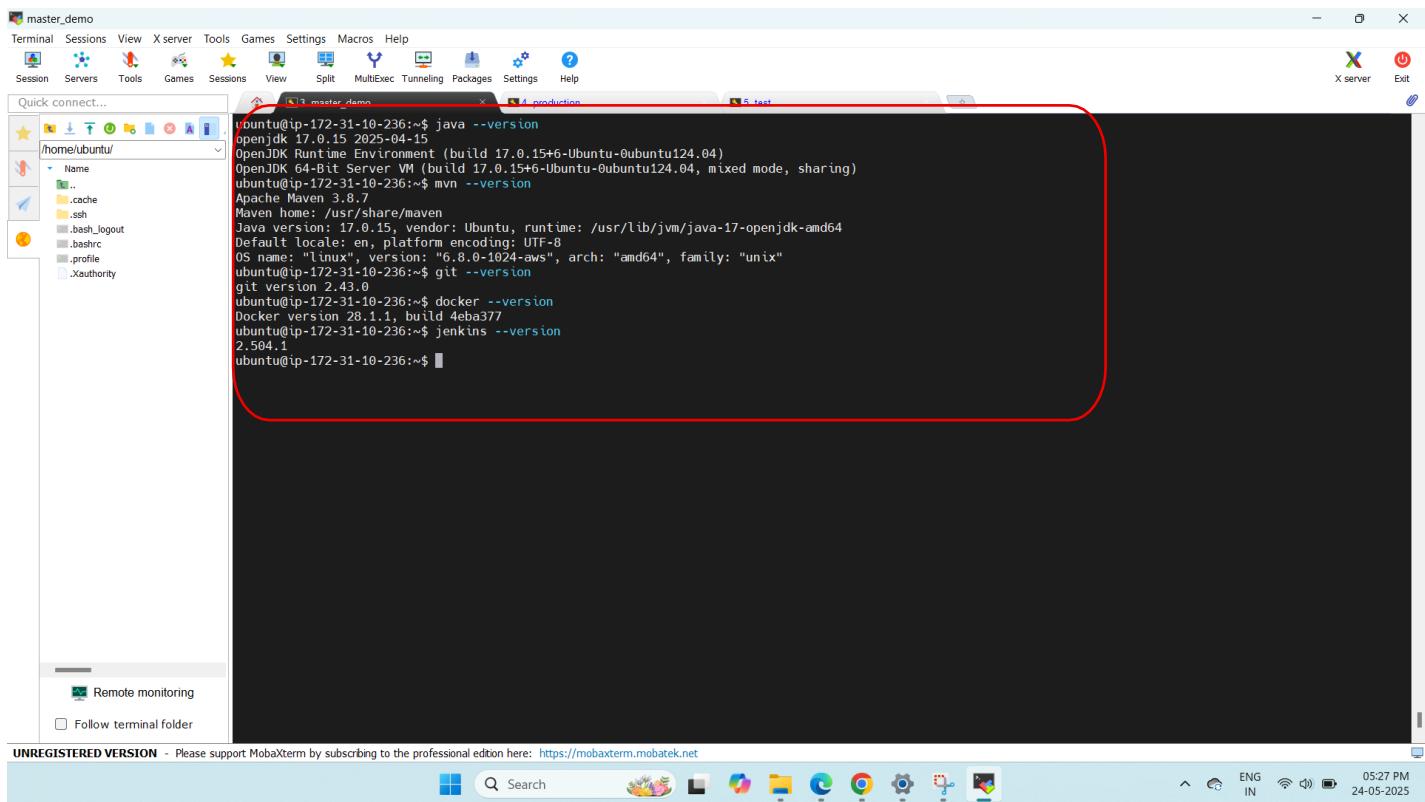
Jenkins-master machine will consist of Jenkins host and also ansible controller

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar navigation includes 'Instances' (selected), 'Images', and 'Elastic Block Store'. The main area displays 'Instances (1/3) Info' with a table of three instances. The 'test' instance is selected and highlighted with a red box. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. A 'Launch instances' button is at the top right. Below the table, a detailed view for the 'test' instance shows its Public IPv4 address (35.154.69.20) and Private IPv4 address (172.31.2.246). The 'Details' tab is selected. A tooltip for the Public IPv4 address shows the address 35.154.69.20. The bottom of the page includes a footer with copyright information and a timestamp of 05:18 PM 24-05-2025.

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

The screenshot shows a MobaXterm window with a session named 'master_demo'. The terminal window displays the output of the command 'systemctl status jenkins'. The output shows that the 'jenkins' service is active and running, with a Main PID of 5375. The terminal window is highlighted with a red box. The bottom of the screen shows the MobaXterm interface with various icons and a status bar indicating the date and time (05:24 PM 24-05-2025).

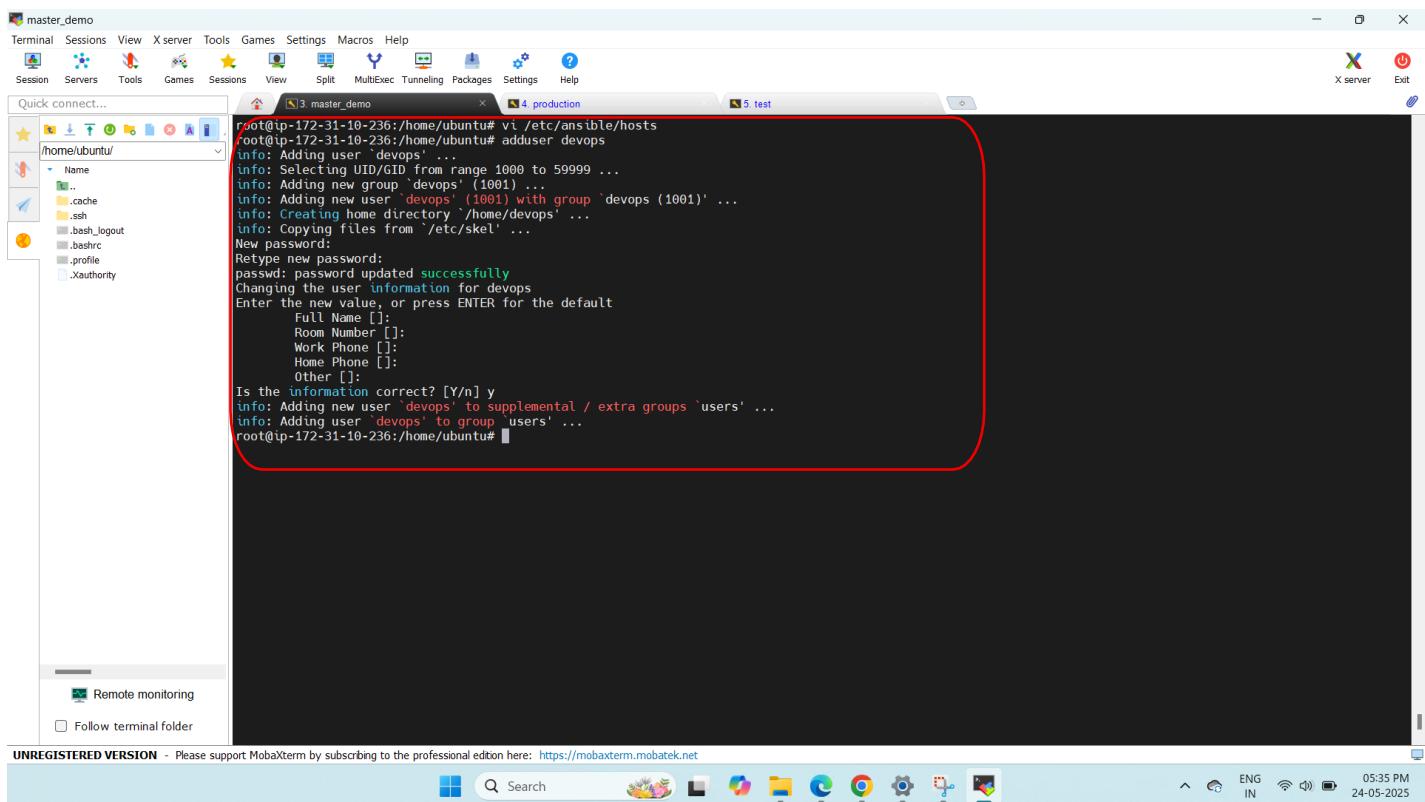
Install maven , git ,ansible,docker,java, Jenkins



```
ubuntu@ip-172-31-10-236:~$ java --version
openjdk 17.0.15 2025-04-15
OpenJDK Runtime Environment (build 17.0.15+6-Ubuntu-0ubuntu124.04)
OpenJDK 64-Bit Server VM (build 17.0.15+6-Ubuntu-0ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-10-236:~$ mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1024-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-10-236:~$ git -version
git version 2.43.0
ubuntu@ip-172-31-10-236:~$ docker --version
Docker version 26.1.1, build 4eba377
ubuntu@ip-172-31-10-236:~$ jenkins --version
2.504.1
ubuntu@ip-172-31-10-236:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

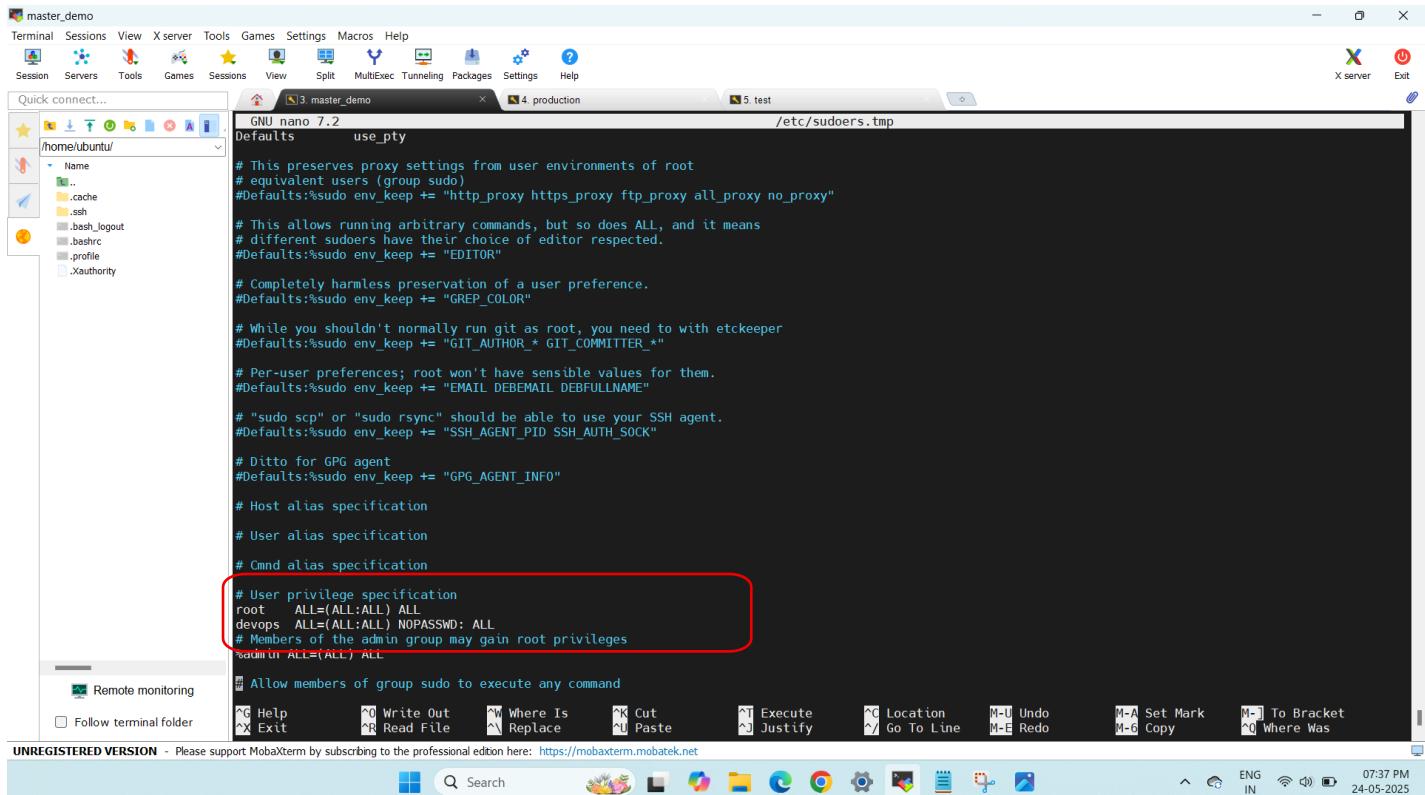
Now perform ansible configuration from controller node to the other two nodes (i.e production and test) also add the user in other two nodes also



```
root@ip-172-31-10-236:/home/ubuntu# vt /etc/ansible/hosts
root@ip-172-31-10-236:/home/ubuntu# adduser devops
info: Adding user 'devops' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group 'devops' (1001) ...
info: Adding new user 'devops' (1001) with group `devops (1001)' ...
info: Creating home directory '/home/devops' ...
info: Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for devops
Enter the new value, or press ENTER for the default
  Full Name []:
  Root Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user 'devops' to supplemental / extra groups 'users' ...
info: Adding user 'devops' to group 'users' ...
root@ip-172-31-10-236:/home/ubuntu#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

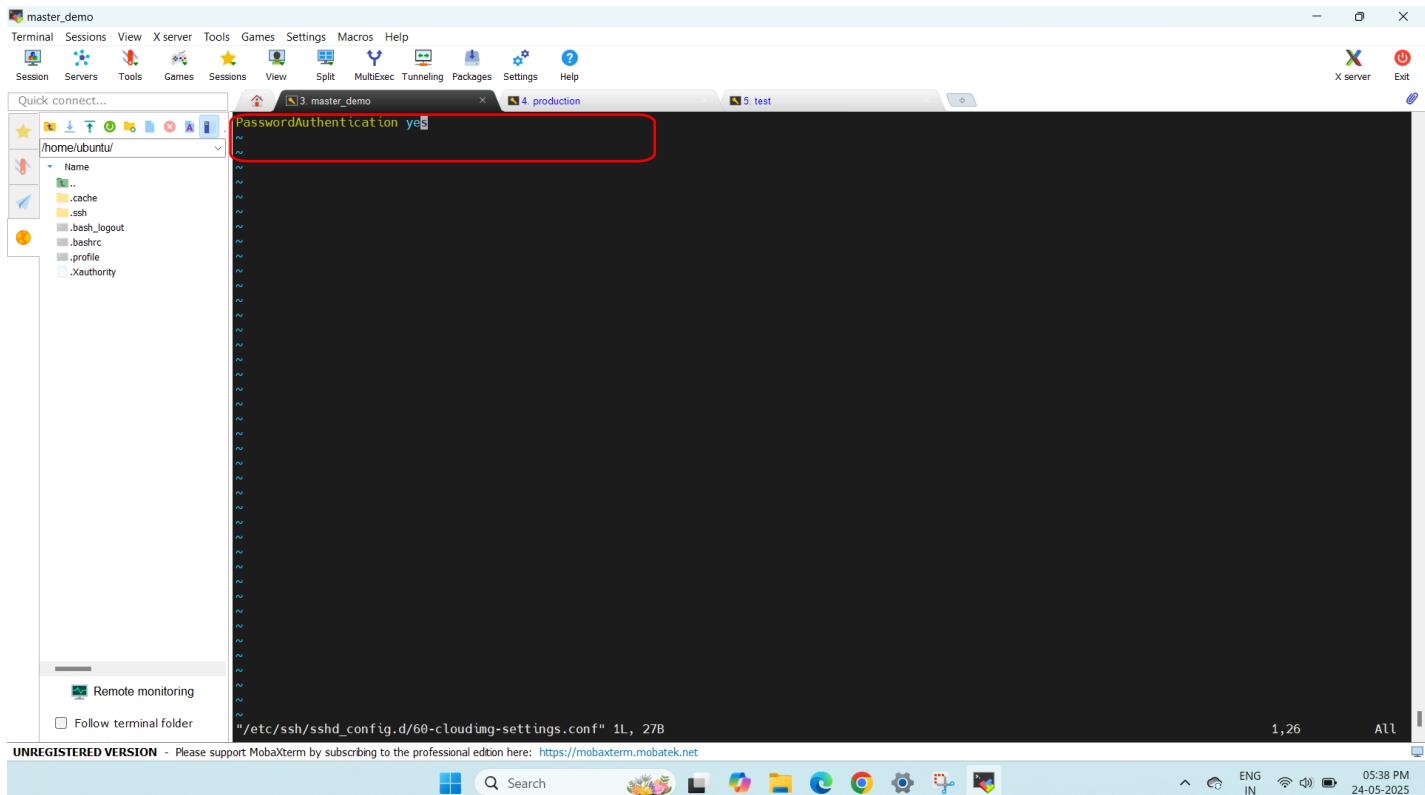
Now add devops user to sudoers file permission in master as well as other nodes also



```
GNU nano 7.2 /etc/sudoers.tmp
Defaults    use_pty
# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"
# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"
# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"
# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"
# sudo "scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
devops  ALL=(ALL:ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%sudo  ALL=(ALL:ALL) ALL

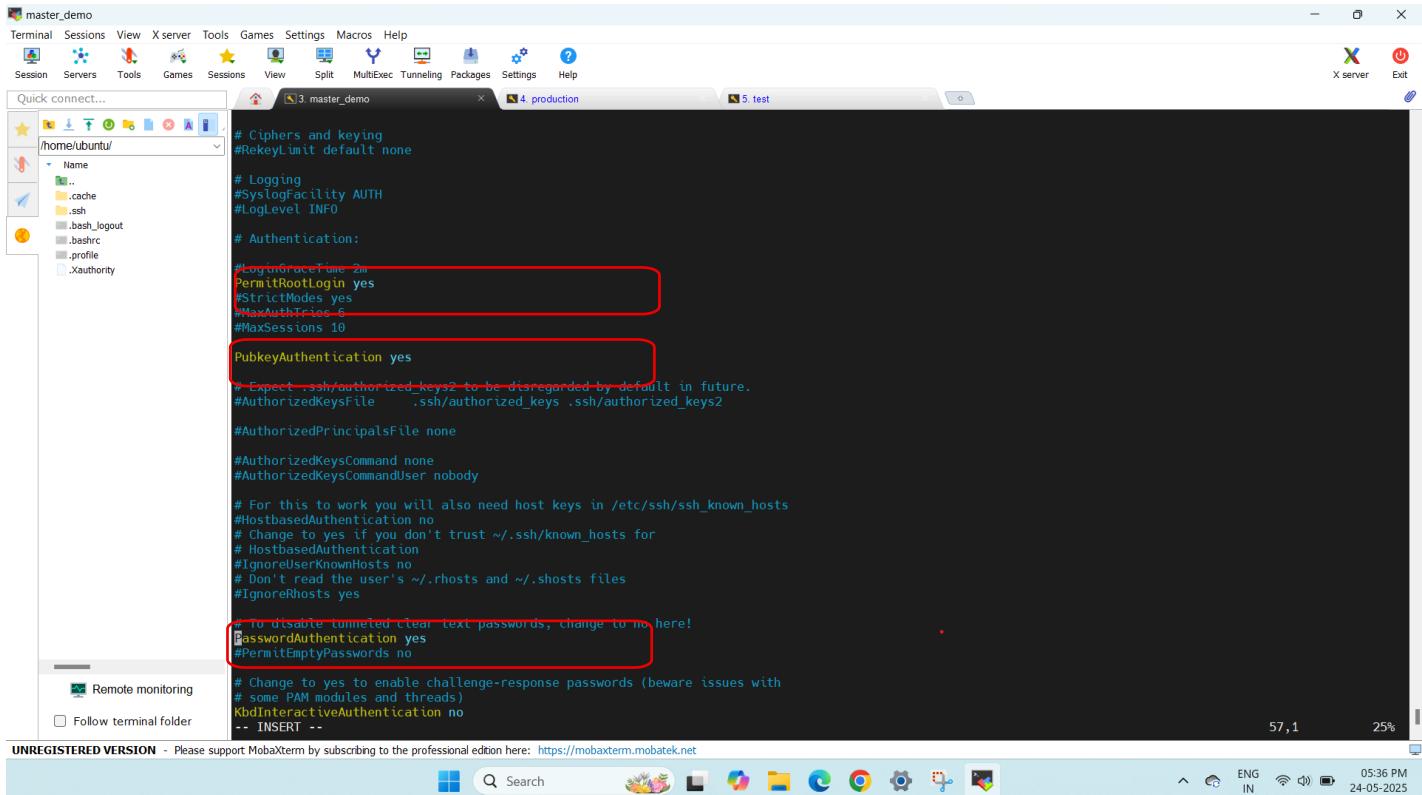
Allow members of group sudo to execute any command
```

Now configure sshd 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



```
GNU nano 7.2 /etc/ssh/sshd_config.d/60-cloudimg-settings.conf
PasswordAuthentication yes
```

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



```
# Ciphers and keying
#RekeyLimit default none

# Logging
#syslogFacility AUTH
#LogLevel INFO

# Authentication:
#LogInexactHosts warn
PermitRootLogin yes
StrictModes yes
MaxAuthTries 6
MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile    .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

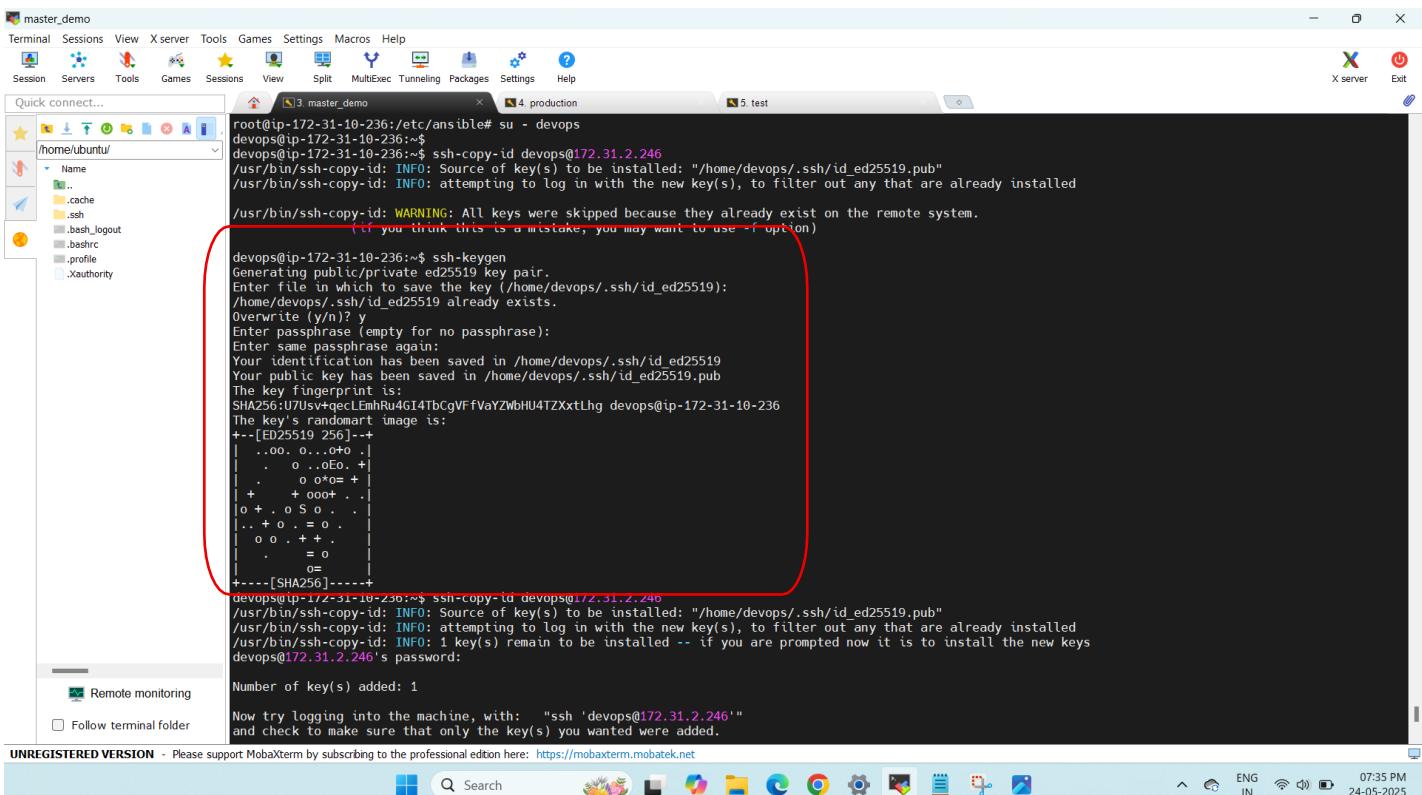
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
#KbdInteractiveAuthentication no
-- INSERT --
```

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



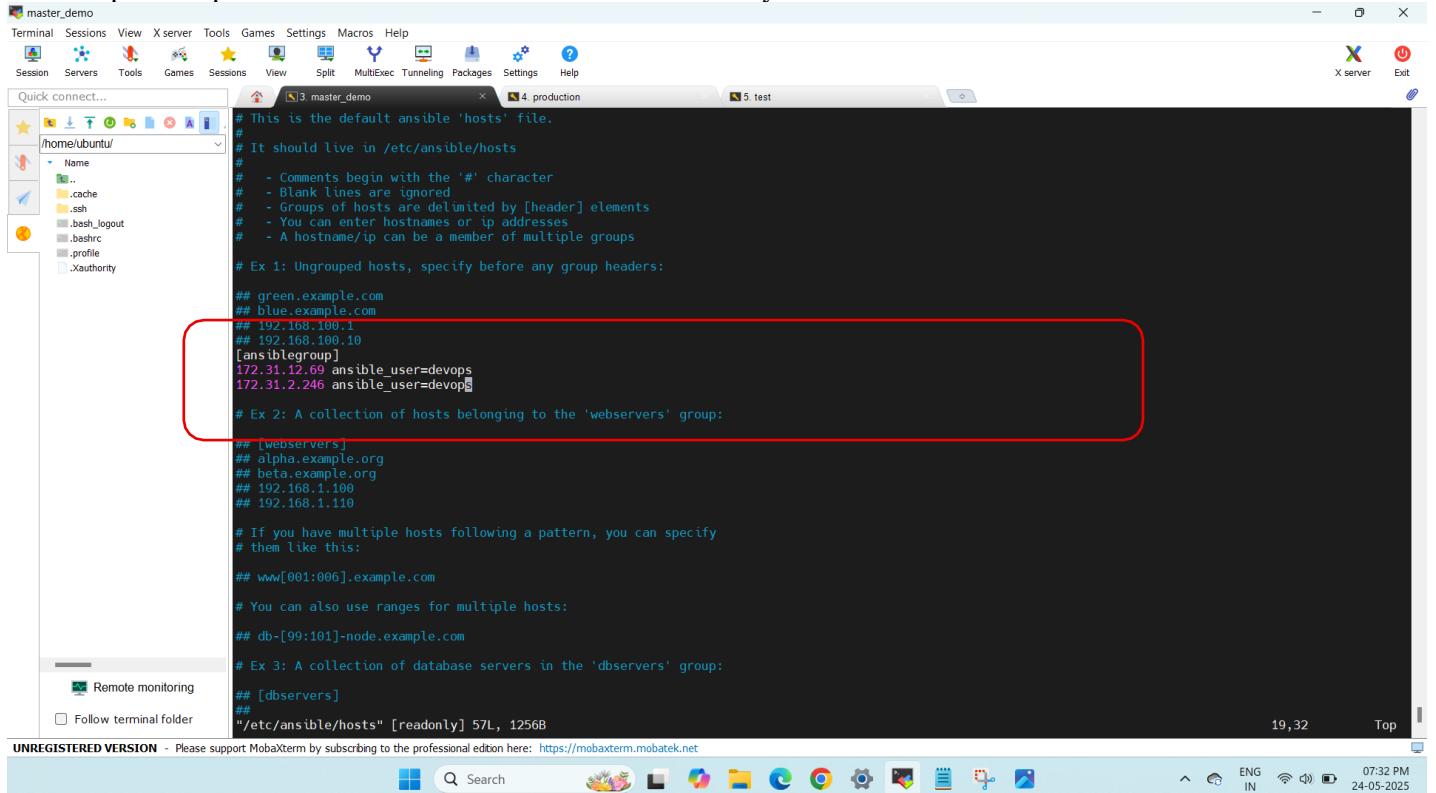
```
root@ip-172-31-10-236:/etc/ansible# su - devops
devops@ip-172-31-10-236:~$ ssh-copy-id devops@172.31.2.246
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
(if you think this is a mistake, you may want to use -f option)

devops@ip-172-31-10-236:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/devops/.ssh/id_ed25519):
/home/devops/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devops/.ssh/id_ed25519
Your public key has been saved in /home/devops/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:U7Usv+qecEmhRu4G14TbCgVFfVaYZWbIu4TZxxtLhg devops@ip-172-31-10-236
The key's randomart image is:
++-[ED25519 256]-+
| ..oo. o...o+o .|
| . o ..oEo. +|
| . o o+=+ . |
| + . + ooo+ . .|
| o + . o S o . .|
| .. + o . = o .|
| o o . + + . |
| . = o . |
| = o= . |
+---[SHA256]---+
devops@ip-172-31-10-236:~$ ssh-copy-id devops@172.31.2.246
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
/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@172.31.2.246's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.2.246'"
and check to make sure that only the key(s) you wanted were added.
```

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



```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers:
#
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
[ansiblegroup]
172.31.12.69 ansible_user=devops
172.31.2.246 ansible_user=devops

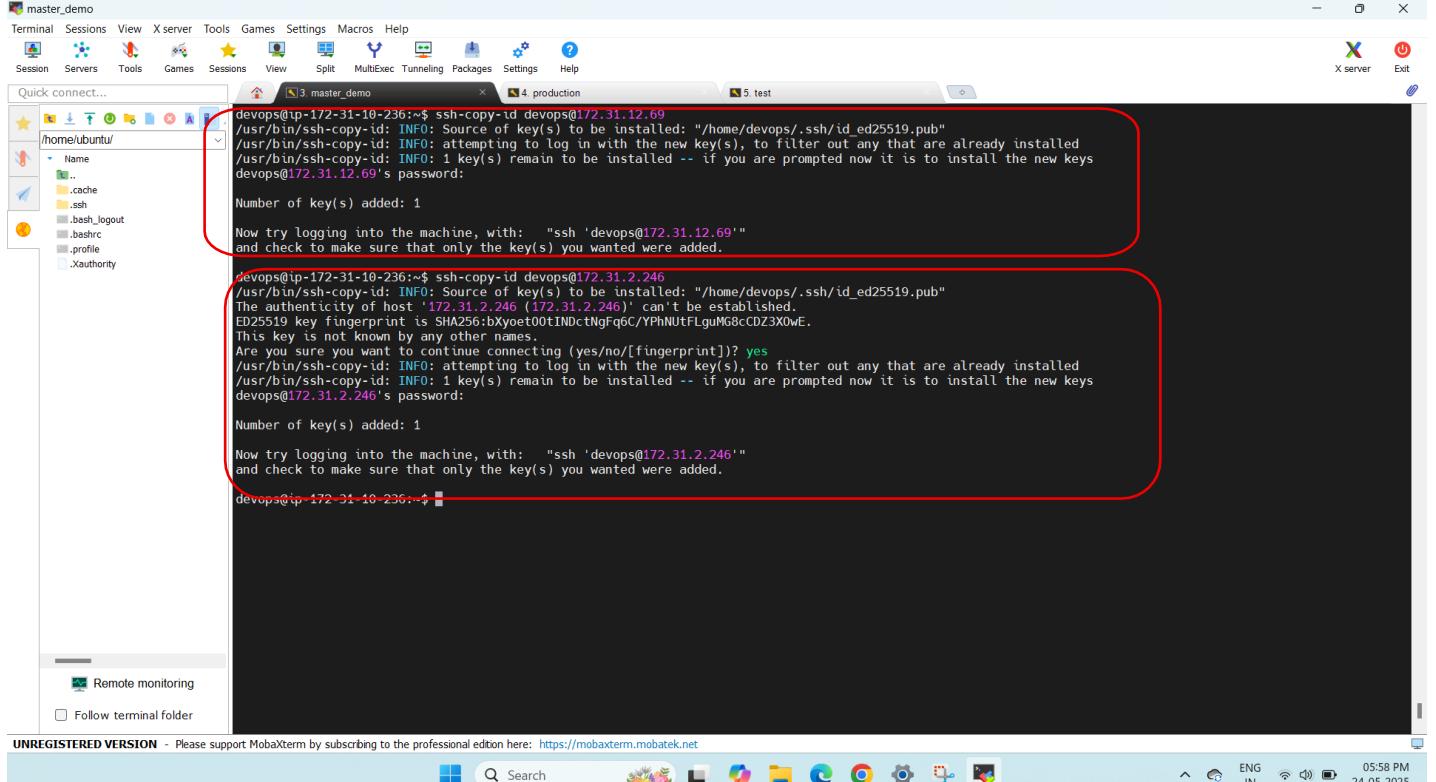
# Ex 2: A collection of hosts belonging to the 'webservers' group:
#
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:
#
## www[001:006].example.com

# You can also use ranges for multiple hosts:
## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:
#
## [dbservers]
## "/etc/ansible/hosts" [readonly] 57L, 1256B
```

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 configured to the controller



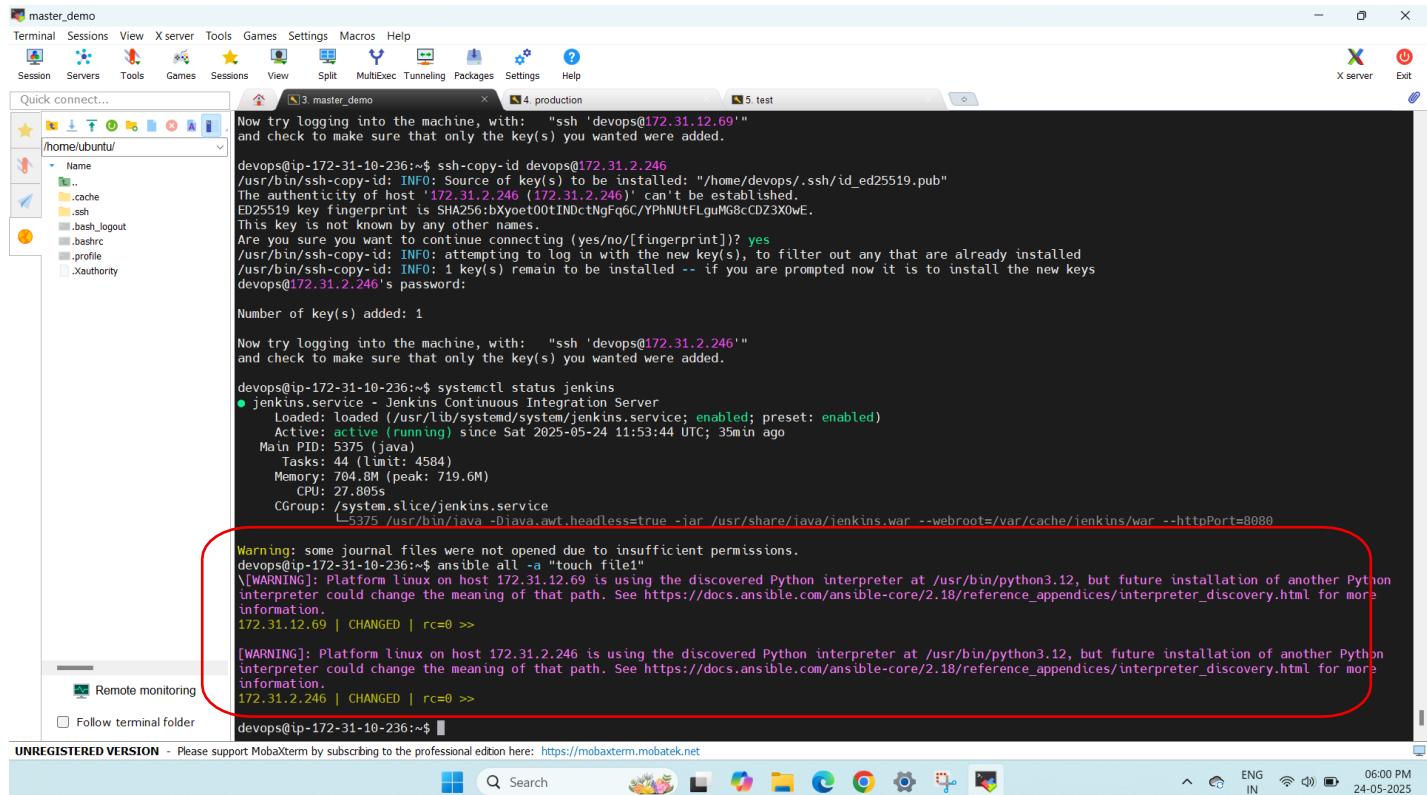
```
devops@ip-172-31-10-236:~$ ssh-copy-id devops@172.31.12.69
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
devops@172.31.12.69's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.12.69'"
and check to make sure that only the key(s) you wanted were added.

devops@ip-172-31-10-236:~$ ssh-copy-id devops@172.31.2.246
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.2.246 (172.31.2.246)' can't be established.
ED25519 key fingerprint is SHA256:bXyoeT0tINctgfrqjC/YPhNUTLguM6cCDZ3X0mE.
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@172.31.2.246's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.2.246'"
and check to make sure that only the key(s) you wanted were added.
```

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



```
Now try logging into the machine, with: "ssh 'devops@172.31.2.246'" and check to make sure that only the key(s) you wanted were added.
devops@ip-172-31-10-236:~$ ssh-copy-id devops@172.31.2.246
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_ed25519.pub"
The authenticity of host '172.31.2.246 (172.31.2.246)' can't be established.
ED25519 key fingerprint is SHA256:bXygot0tINDctNgfqC/YPhNUTlGuMG8cCDZ3X0WE.
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@172.31.2.246:~$ 

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'devops@172.31.2.246'" and check to make sure that only the key(s) you wanted were added.

devops@ip-172-31-10-236:~$ systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
  Active: active (running) since Sat 2025-05-24 11:53:44 UTC; 35min ago
    Main PID: 5375 (java)
       Tasks: 44 (limit: 4584)
      Memory: 704.8M (peak: 719.6M)
        CPU: 27.805s
       CGroup: /system.slice/jenkins.service
              └─5375 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

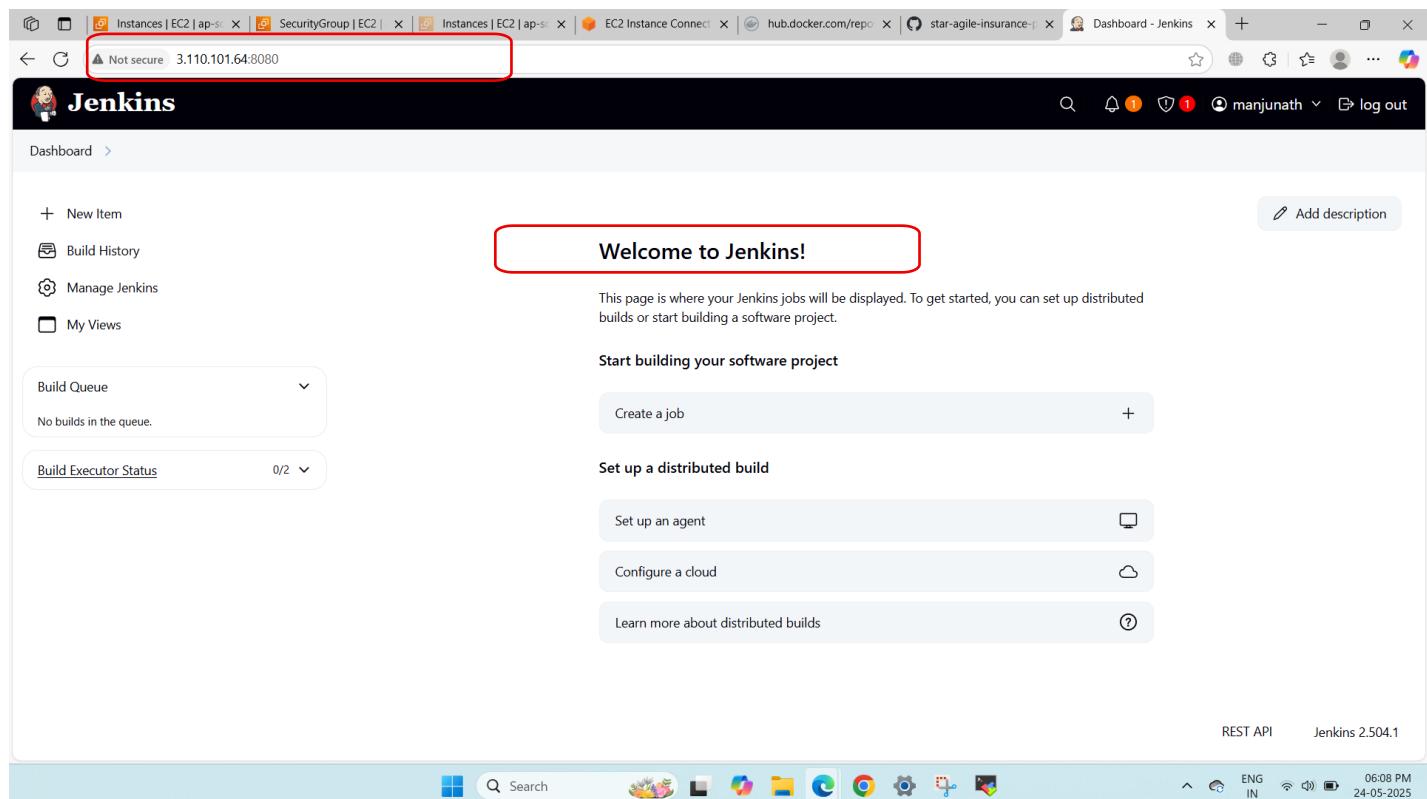
Warning: some journal files were not opened due to insufficient permissions.
devops@ip-172-31-10-236:~$ ansible all -a "touch file1"
[WARNING]: Platform linux on host 172.31.2.69 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
172.31.2.69 | CHANGED | rc=0 >>
[WARNING]: Platform linux on host 172.31.2.246 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.
172.31.2.246 | CHANGED | rc=0 >>
devops@ip-172-31-10-236:~$ 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Remote monitoring Follow terminal folder

06:00 PM 24-05-2025

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



Not secure 3.110.101.64:8080

Jenkins

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

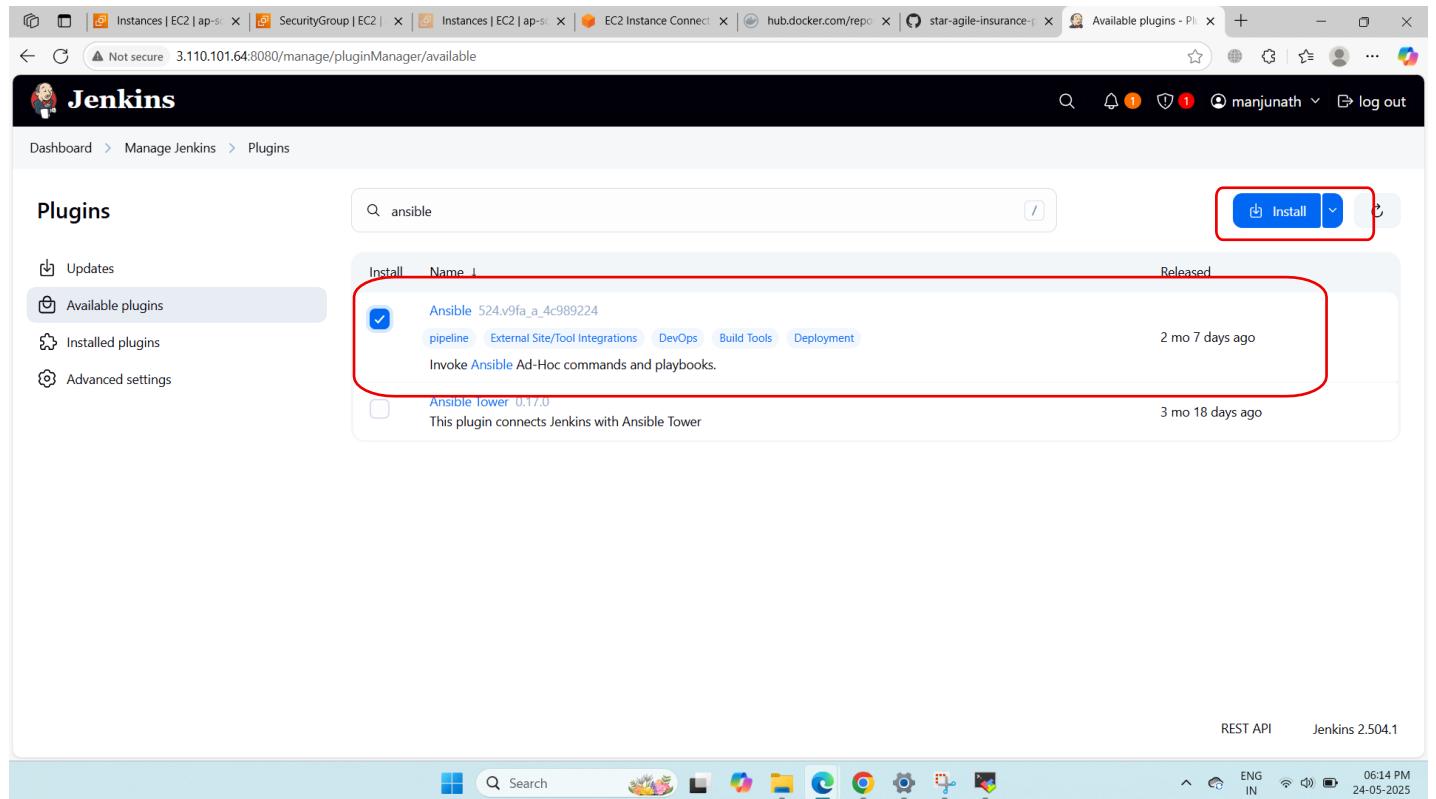
Configure a cloud

Learn more about distributed builds

REST API Jenkins 2.504.1

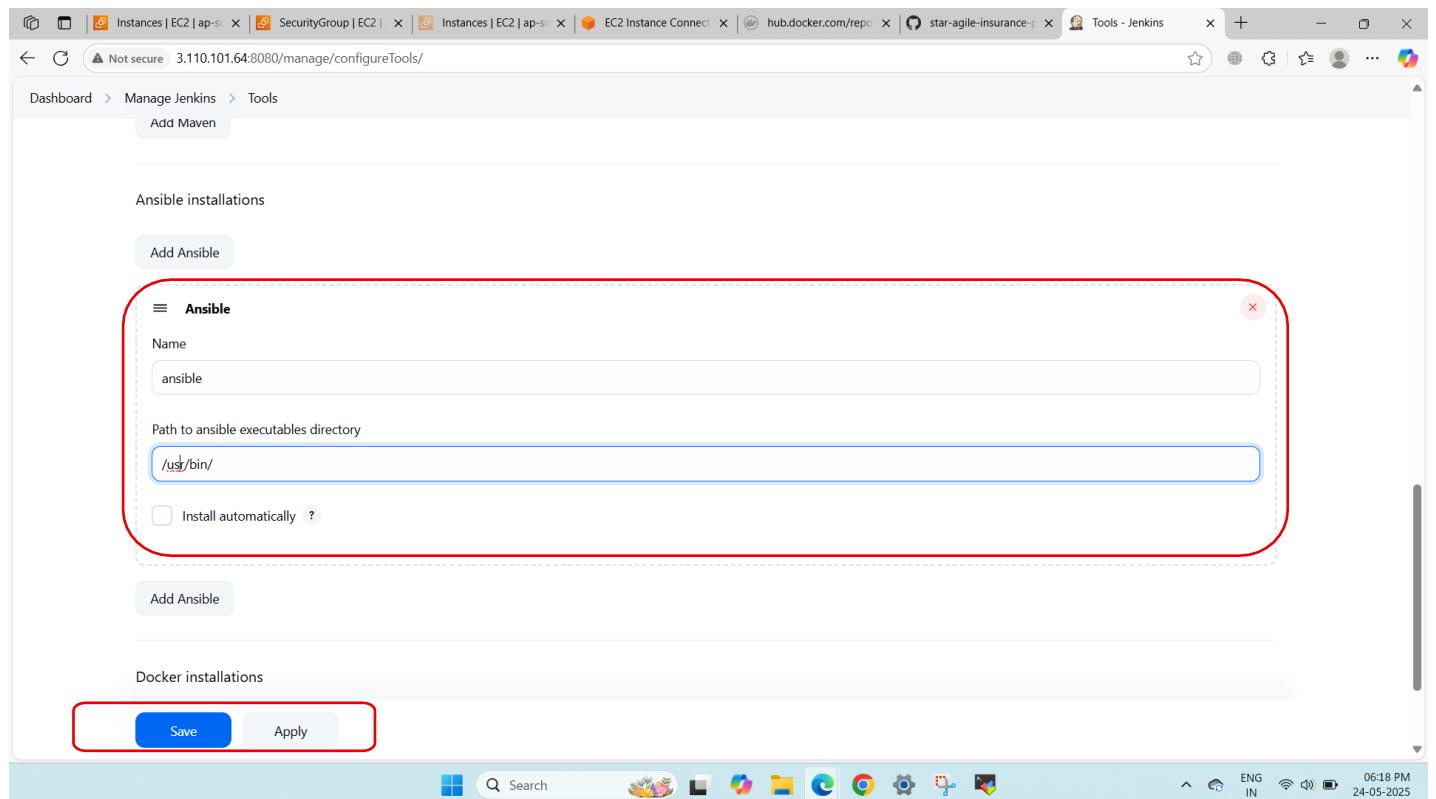
06:08 PM 24-05-2025

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



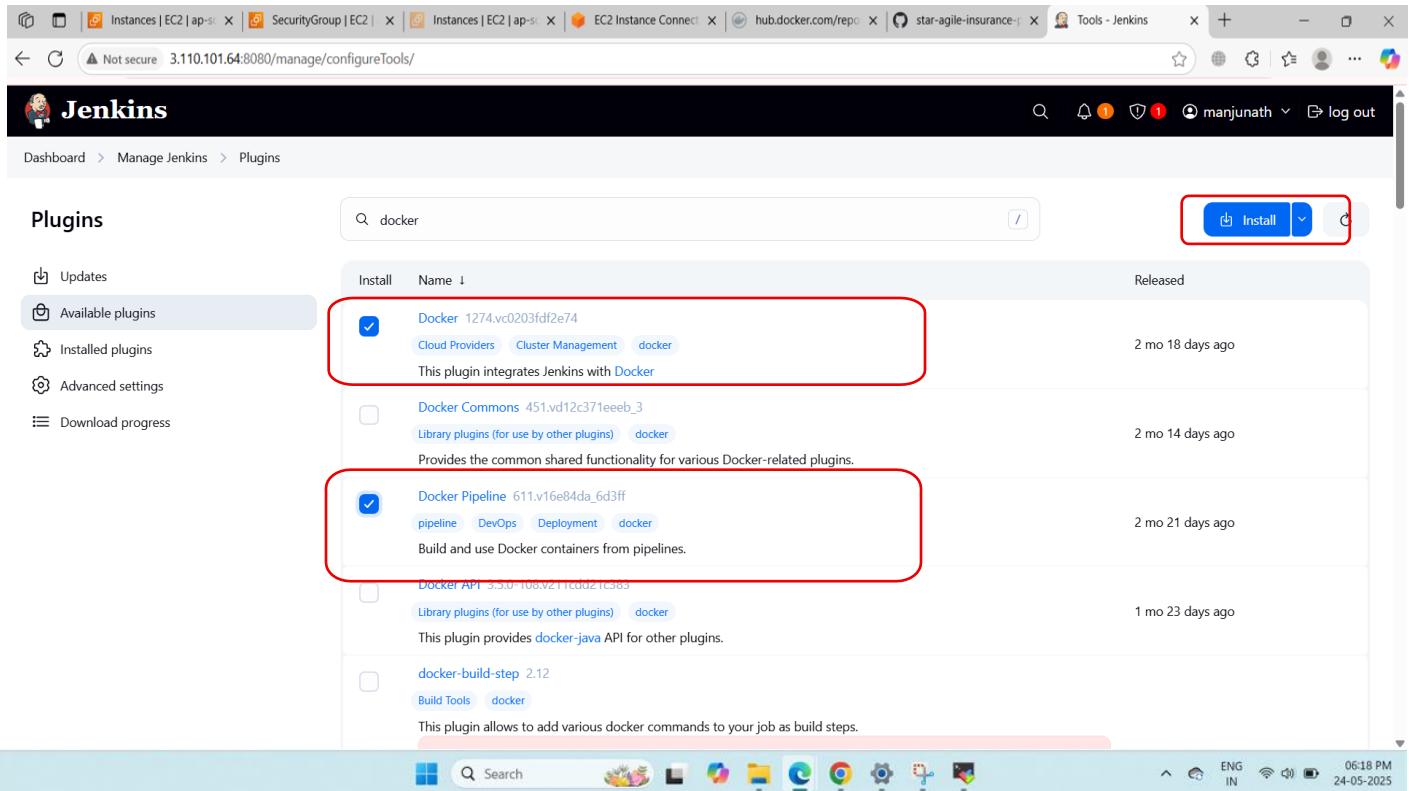
The screenshot shows the Jenkins Plugins page. The search bar at the top contains the text "ansible". On the left, a sidebar has "Available plugins" selected. In the main area, the "Ansible" plugin is highlighted with a red box. It has a checked "Install" button. The plugin details show it was released 2 months and 7 days ago. Below it, the "Ansible Tower" plugin is listed, released 3 months and 18 days ago. The Jenkins version is 2.504.1.

Do ansible configuration and set the path in tools configuration



The screenshot shows the Jenkins Tools configuration page. Under "Ansible installations", a new configuration is being added with the name "ansible" and the path to the executable directory set to "/usr/bin". The "Install automatically" checkbox is unchecked. The "Save" and "Apply" buttons are highlighted with a red box at the bottom.

Install docker plugins

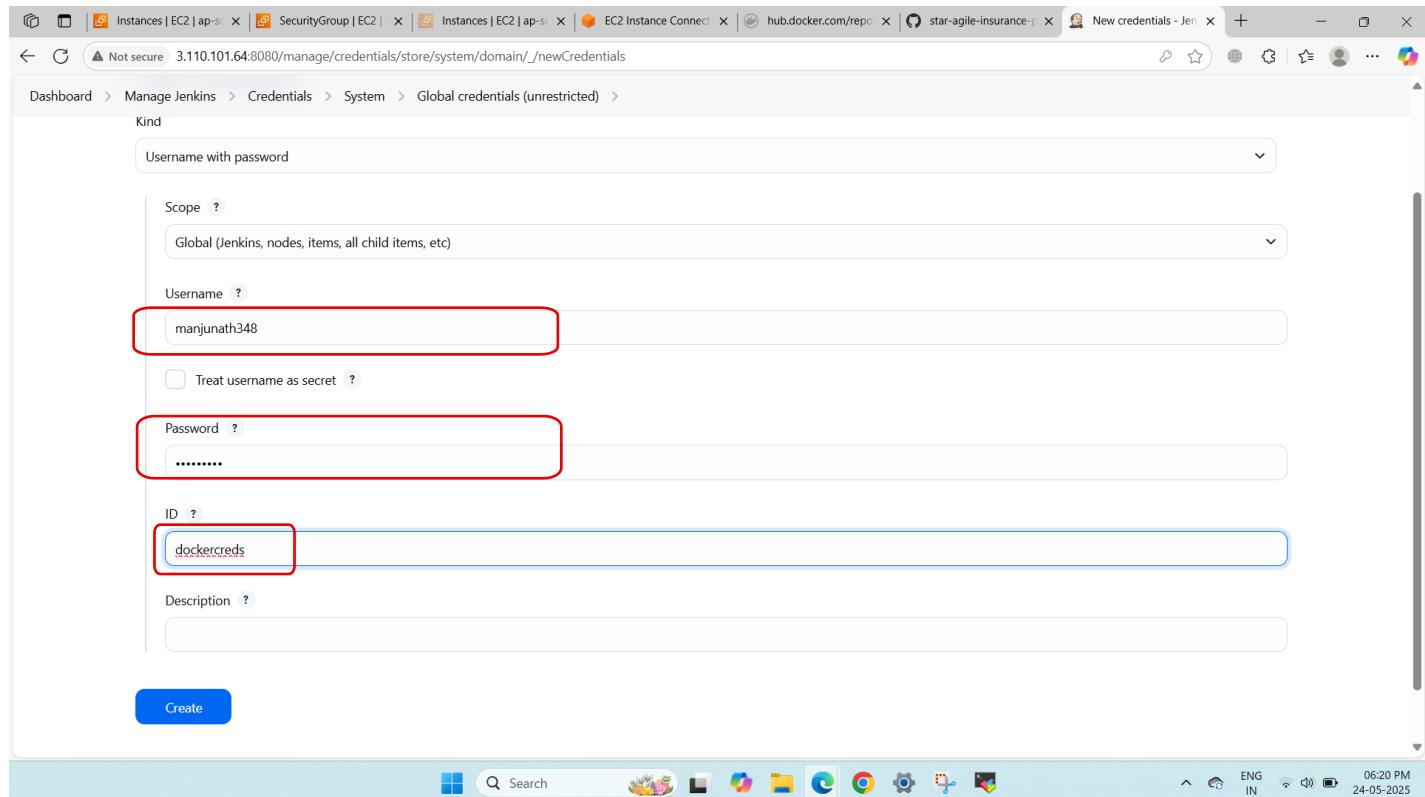


The screenshot shows the Jenkins Plugins page. The search bar at the top contains the text "docker". On the left, a sidebar has "Available plugins" selected. The main table lists several Docker-related plugins:

- Docker** 1274.vc0203fdf2e74 (Released 2 mo 18 days ago): This plugin integrates Jenkins with Docker.
- Docker Commons** 451.vd12c371eeeb_3 (Released 2 mo 14 days ago): Provides the common shared functionality for various Docker-related plugins.
- Docker Pipeline** 611.v16e84da_6d3ff (Released 2 mo 21 days ago): Build and use Docker containers from pipelines.
- Docker API** 5.5.0-106.v211cc0d27c385 (Released 1 mo 23 days ago): This plugin provides docker-java API for other plugins.
- docker-build-step** 2.12 (Released 2 mo 21 days ago): This plugin allows to add various docker commands to your job as build steps.

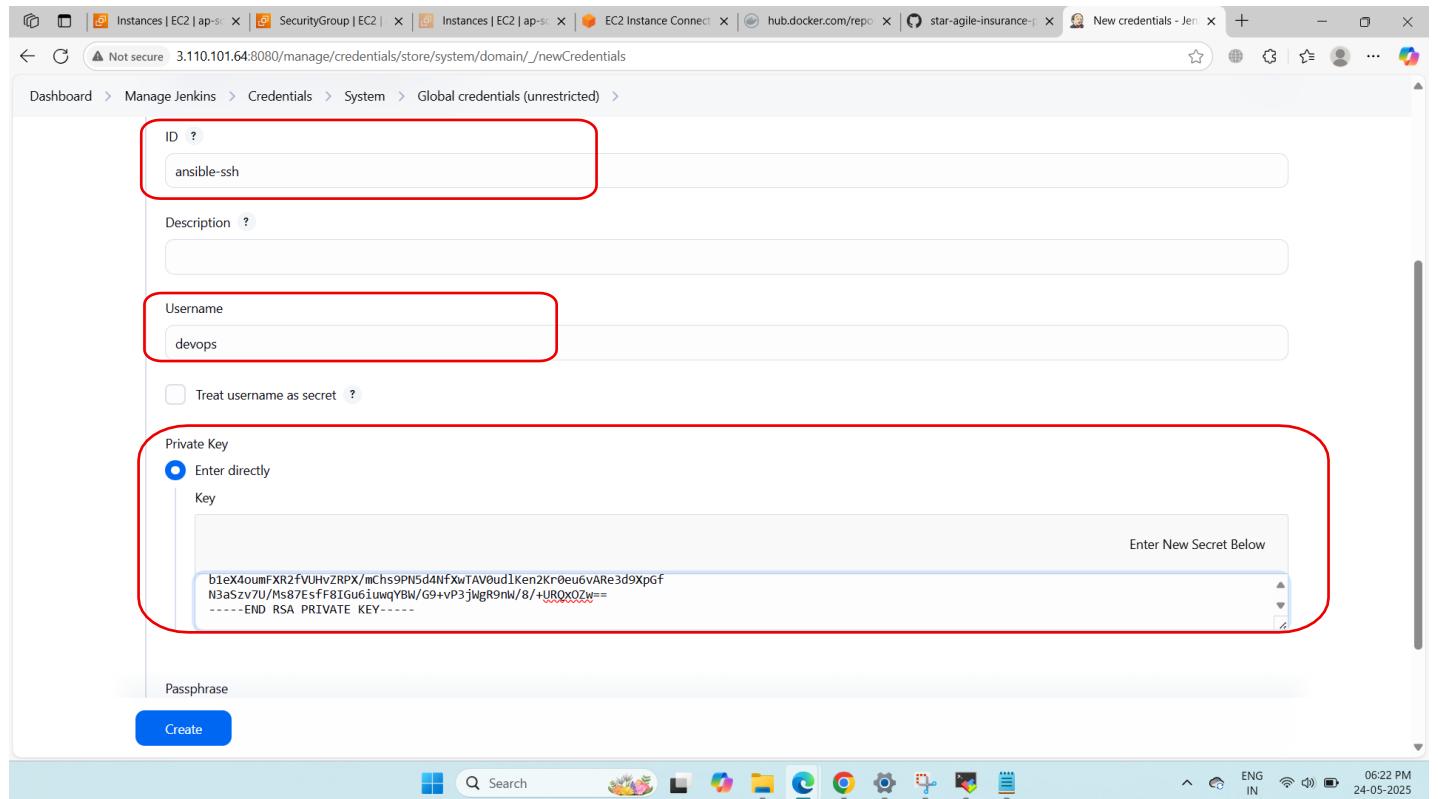
The "Docker" and "Docker Pipeline" rows are highlighted with red boxes. The "Install" button at the top right is also highlighted with a red box.

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



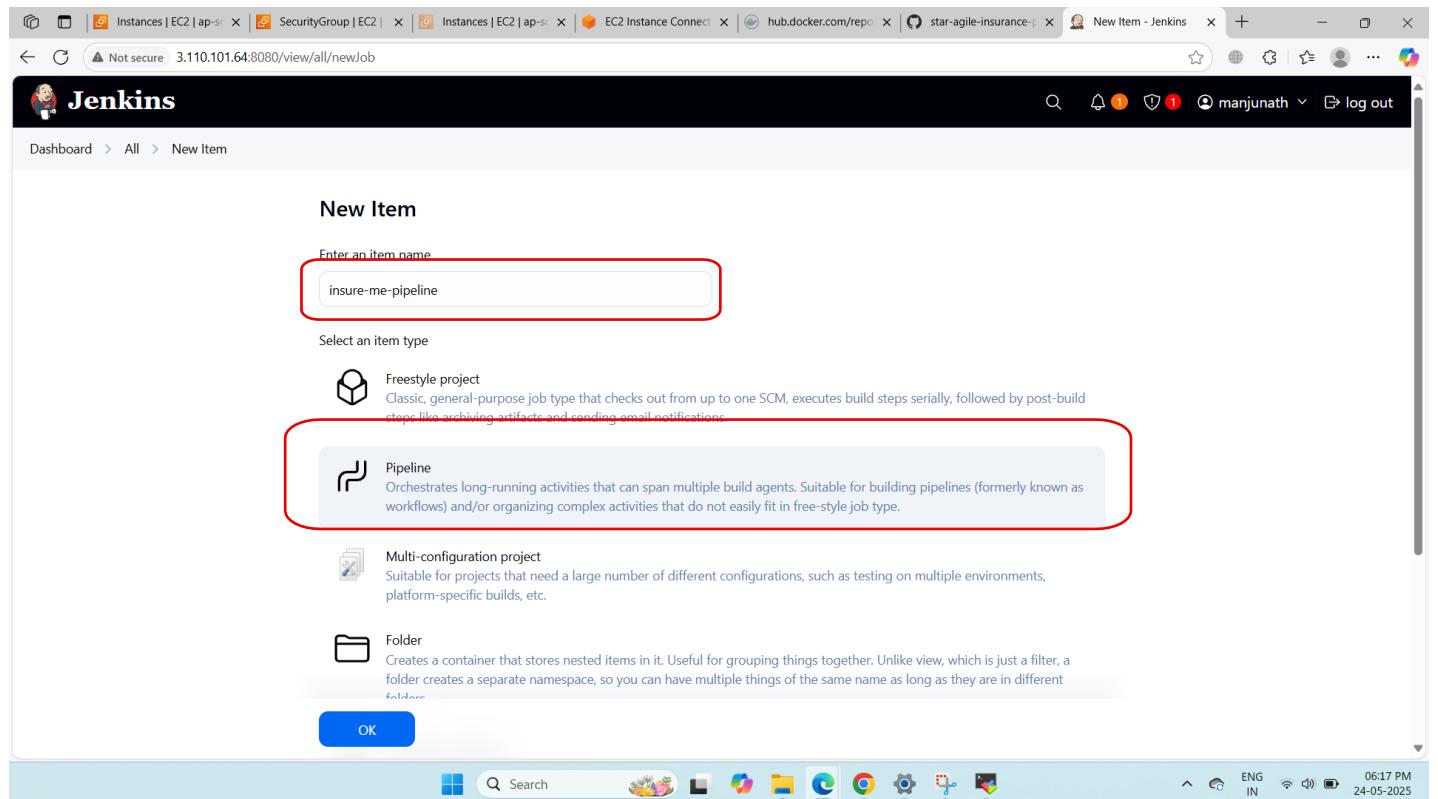
The screenshot shows the Jenkins Credentials page. The URL is 3.110.101.64:8080/manage/credentials/store/system/domain/_/newCredentials. The "Kind" dropdown is set to "Username with password". The "Scope" dropdown is set to "Global (Jenkins, nodes, items, all child items, etc)". The "Username" field contains "manjunath348" (highlighted with a red box). The "Password" field contains "....." (highlighted with a red box). The "ID" field contains "dockercreds" (highlighted with a red box). The "Create" button at the bottom is highlighted with a red box.

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



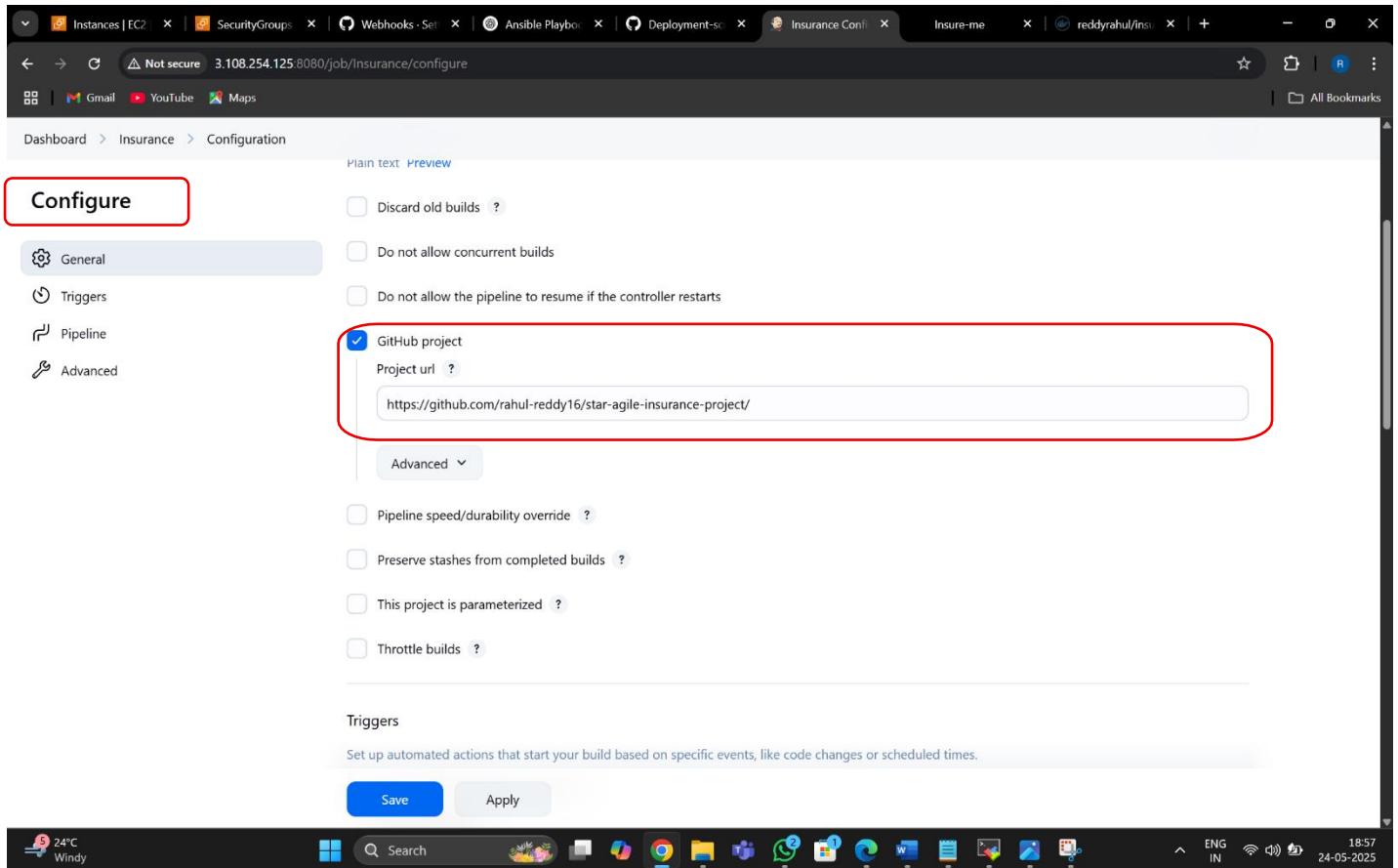
The screenshot shows the Jenkins 'New credentials' page. The 'ID' field is set to 'ansible-ssh' and the 'Username' field is set to 'devops'. The 'Private Key' section is expanded, showing the 'Enter directly' option selected. The 'Key' field contains a long string of characters representing an RSA private key, starting with 'b1eX4oumFXR2fVUhvZRPX/mChs9PN5d4NfxwTA0eud1Ken2Kr0eu6vARe3d9XpGf'. Below the key field is a placeholder 'Enter New Secret Below'. A red box highlights the 'ID' field, the 'Username' field, and the 'Private Key' section.

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



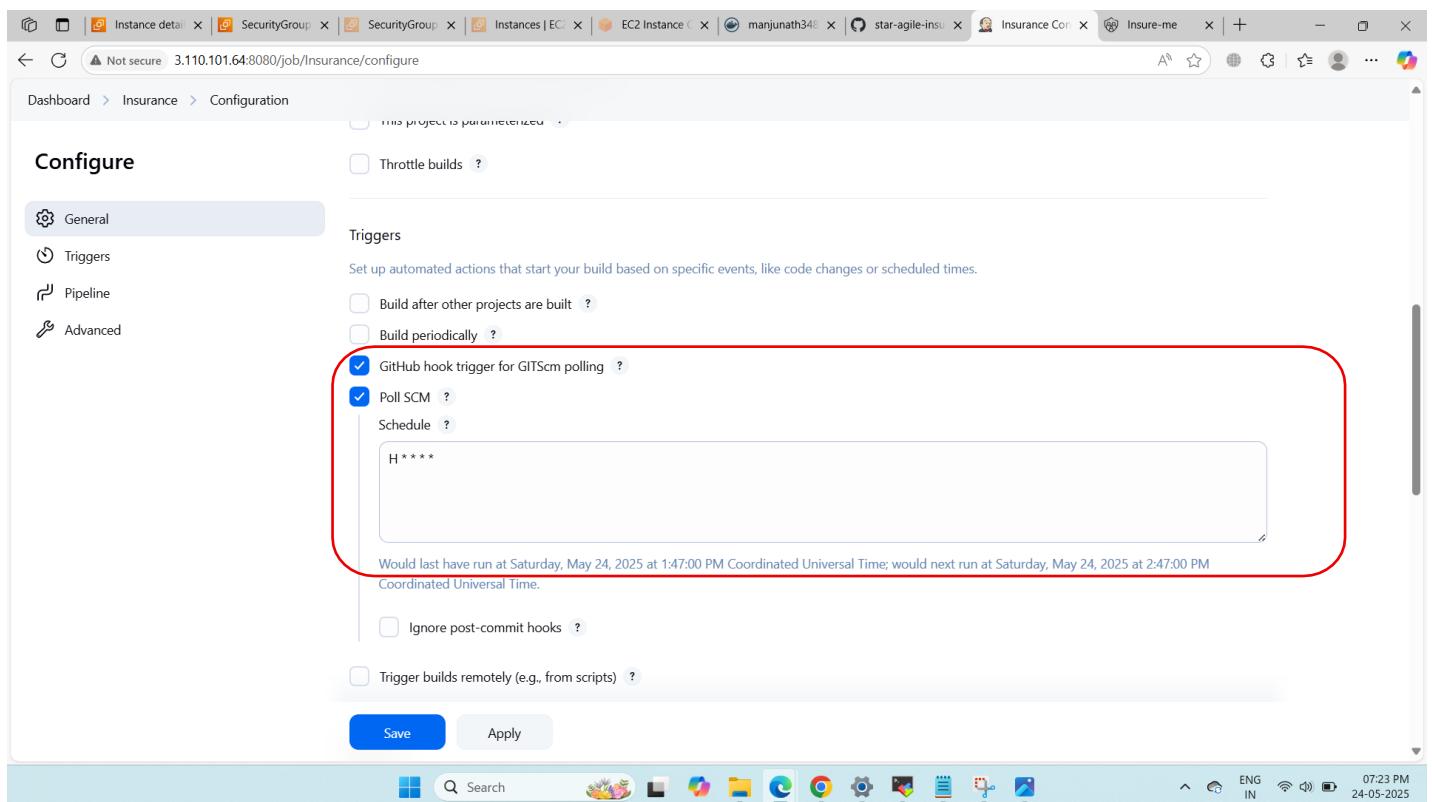
The screenshot shows the Jenkins 'New Item' creation page. The 'Enter an item name' field is filled with 'insure-me-pipeline'. The 'Select an item type' section shows four options: 'Freestyle project', 'Pipeline', 'Multi-configuration project', and 'Folder'. The 'Pipeline' option is highlighted with a red box. It has a description: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' A blue 'OK' button is at the bottom. A red box highlights the 'Enter an item name' field and the 'Pipeline' item type section.

Configure the project with github project repository



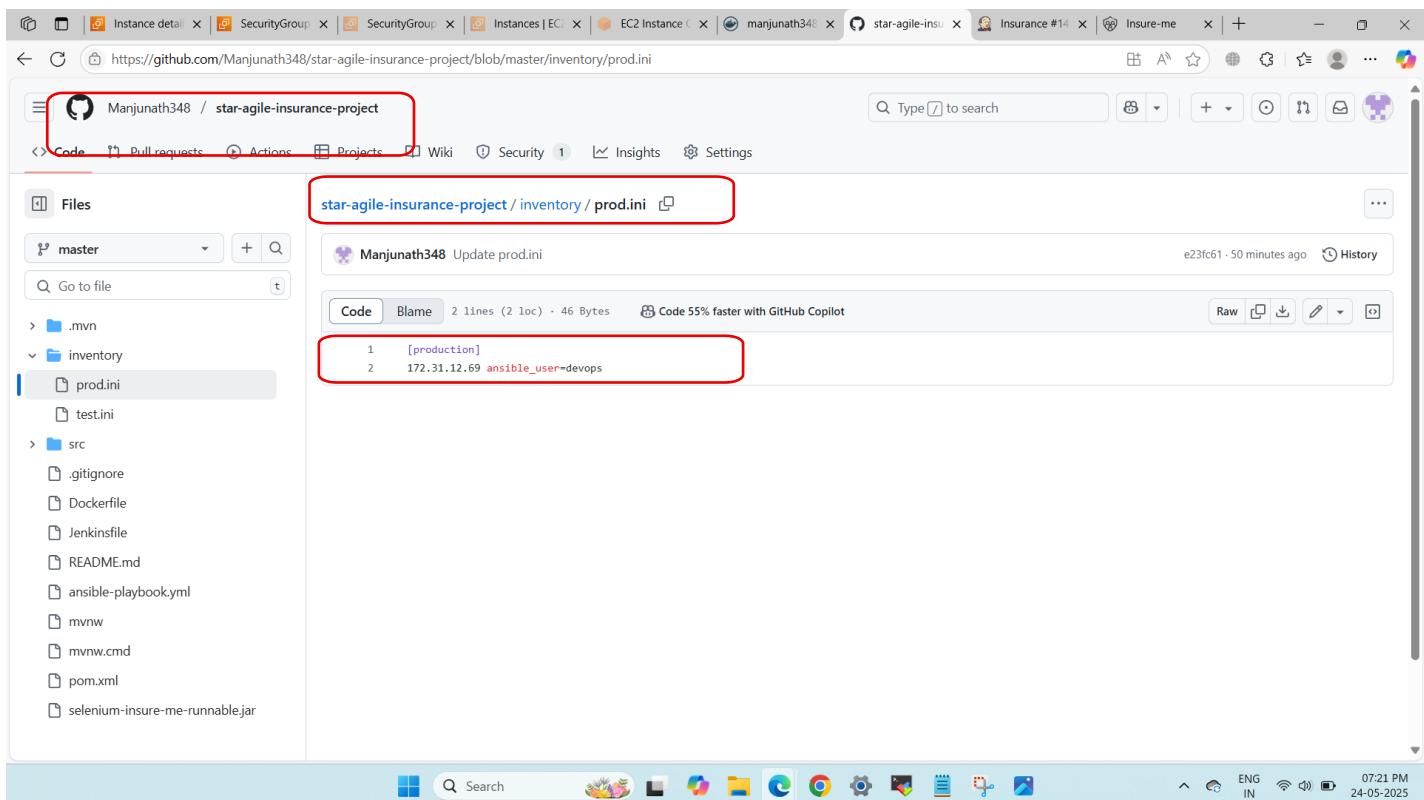
The screenshot shows a web browser with multiple tabs open, including 'Instances | EC2', 'SecurityGroups', 'Webhooks - Set', 'Ansible Playbook', 'Deployment-SC', 'Insurance Config', 'Insure-me', and 'reddyrahul/inst...'. The main content is a configuration page for a GitHub project. A red box highlights the 'GitHub project' section, which contains a checked checkbox and a text input field with the URL <https://github.com/rahul-reddy16/star-agile-insurance-project/>. Other sections like 'General', 'Triggers', 'Pipeline', and 'Advanced' are listed on the left. The browser's status bar at the bottom shows the date as 24-05-2025 and the time as 18:57.

add the triggers



The screenshot shows a web browser with multiple tabs open, including 'Instance detail', 'SecurityGroup', 'SecurityGroup', 'Instances | EC2', 'EC2 Instance', 'manjunath34C', 'star-agile-insu...', 'Insurance Config', 'Insure-me', and 'reddyrahul/inst...'. The main content is a configuration page for triggers. A red box highlights the 'GitHub hook trigger for GITSCM polling' and 'Poll SCM' checkboxes. The 'Schedule' field contains the cron expression 'H * * * *'. Other trigger options like 'Build after other projects are built', 'Build periodically', and 'Ignore post-commit hooks' are listed. The browser's status bar at the bottom shows the date as 24-05-2025 and the time as 07:23 PM.

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



Manjunath348 / star-agile-insurance-project

Code Pull requests Actions Projects Wiki Security Insights Settings

star-agile-insurance-project / inventory / prod.ini

Manjunath348 Update prod.ini

Code Blame 2 lines (2 loc) · 46 Bytes Code 55% faster with GitHub Copilot

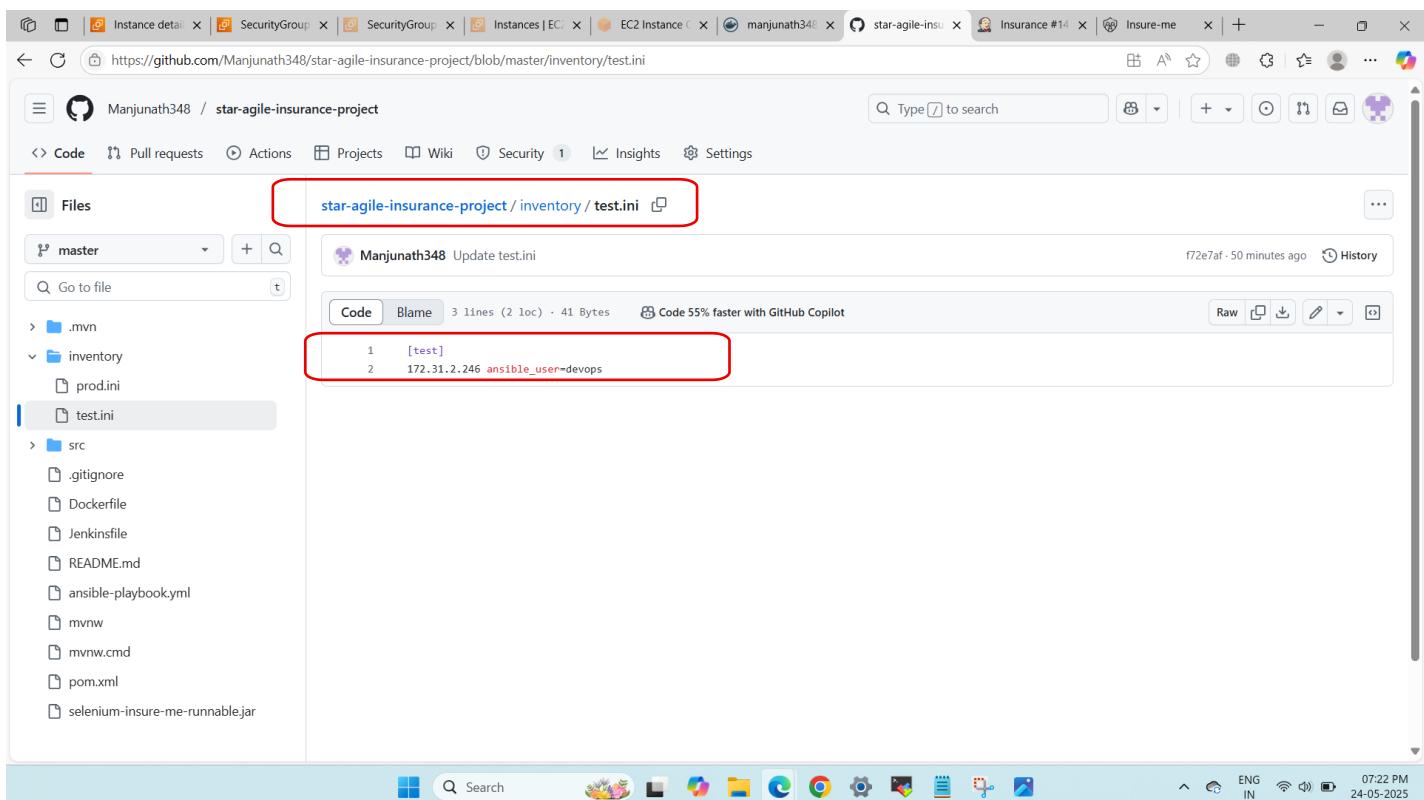
```
1 [production]
2 172.31.12.69 ansible_user=devops
```

Raw ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

Files

master .mvn inventory prod.ini test.ini src .gitignore Dockerfile Jenkinsfile README.md ansible-playbook.yml mvnw mvnw.cmd pom.xml selenium-insure-me-runnable.jar

07:21 PM 24-05-2025



Manjunath348 / star-agile-insurance-project

Code Pull requests Actions Projects Wiki Security Insights Settings

star-agile-insurance-project / inventory / test.ini

Manjunath348 Update test.ini

Code Blame 3 lines (2 loc) · 41 Bytes Code 55% faster with GitHub Copilot

```
1 [test]
2 172.31.2.246 ansible_user=devops
```

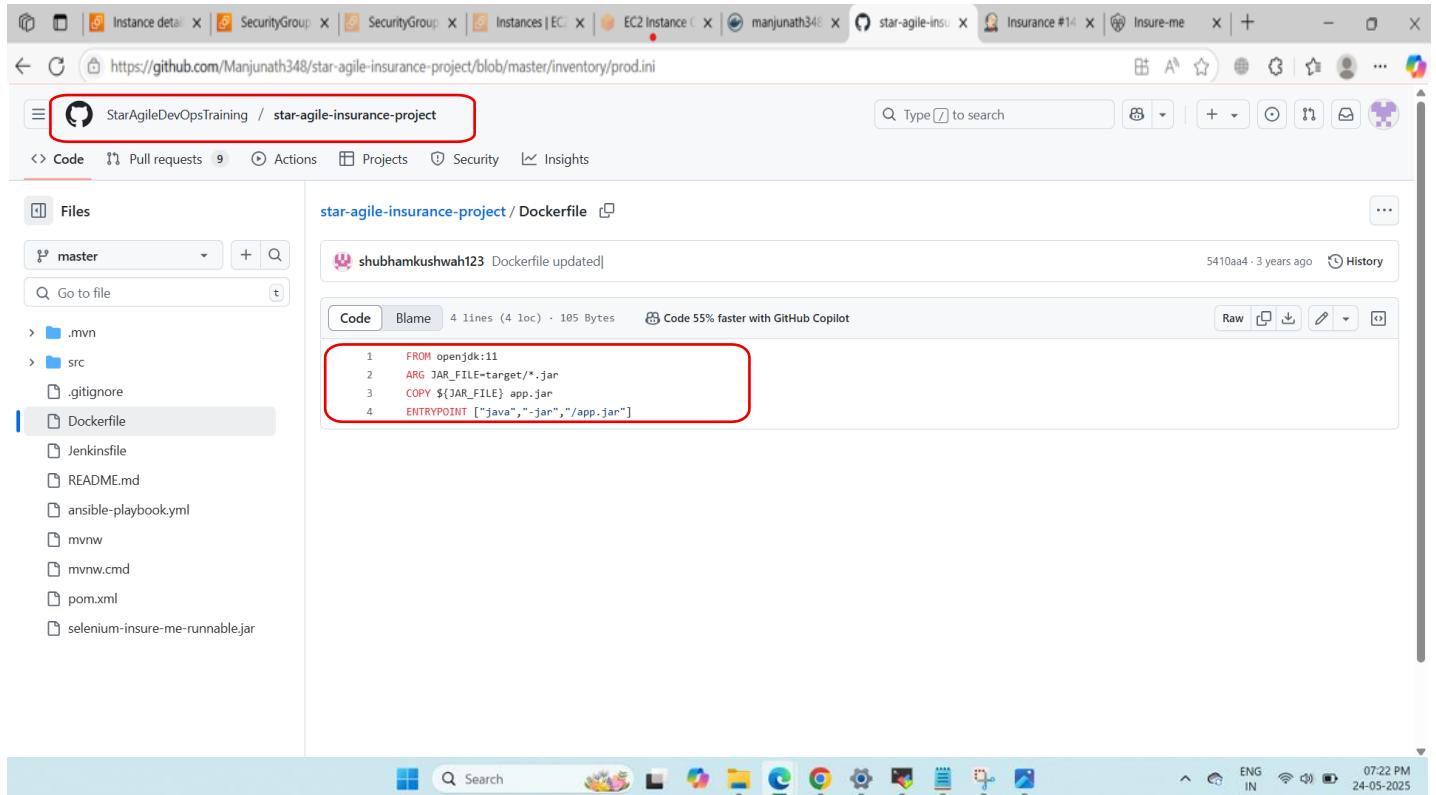
Raw ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

Files

master .mvn inventory prod.ini test.ini src .gitignore Dockerfile Jenkinsfile README.md ansible-playbook.yml mvnw mvnw.cmd pom.xml selenium-insure-me-runnable.jar

07:22 PM 24-05-2025

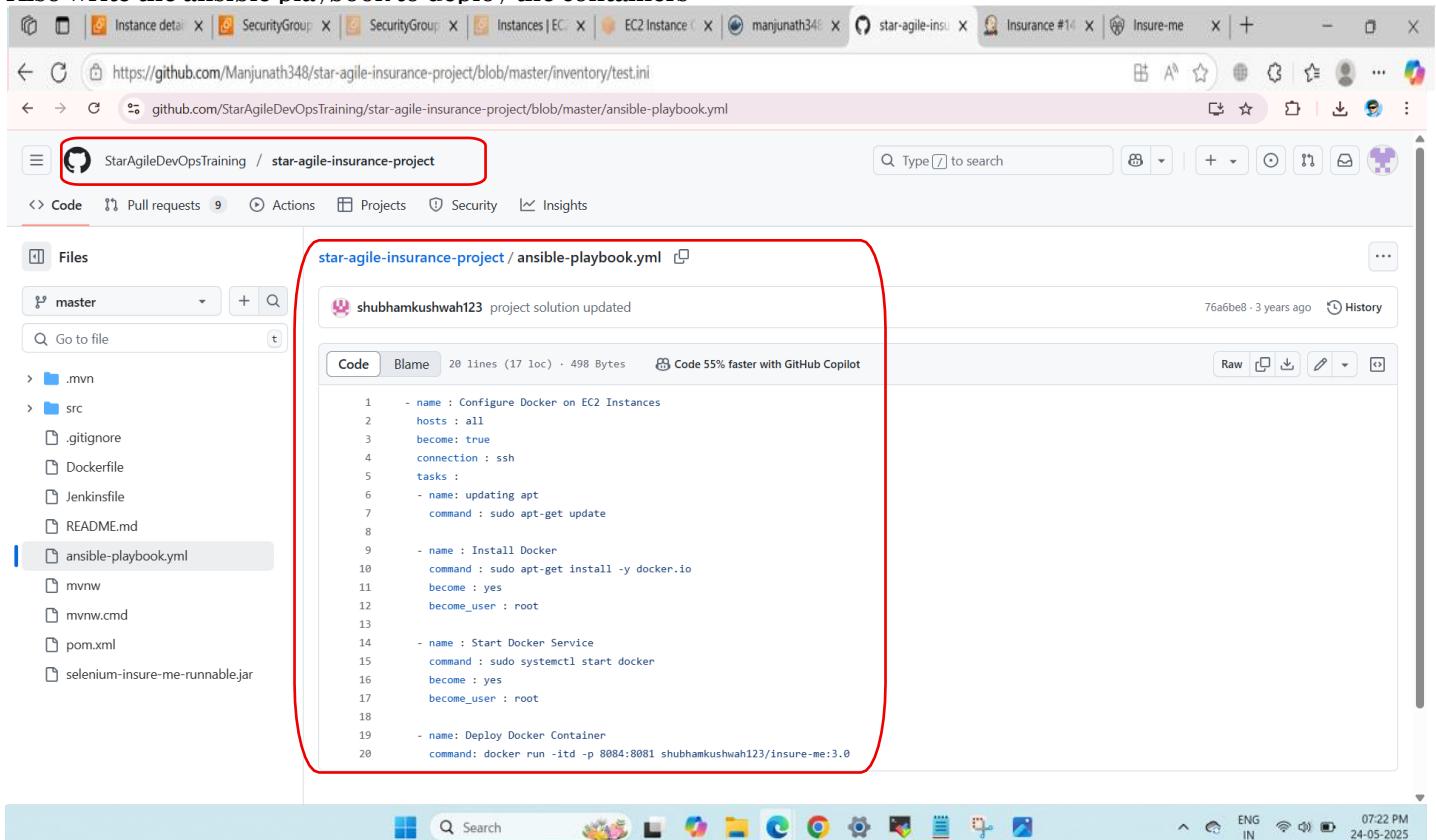
Now write the Dockerfile



The screenshot shows a GitHub repository for 'star-agile-insurance-project'. The 'Dockerfile' tab is selected. The code in the Dockerfile is:

```
1 FROM openjdk:11
2 ARG JAR_FILE=target/*.jar
3 COPY ${JAR_FILE} app.jar
4 ENTRYPOINT ["java","-jar","/app.jar"]
```

Also write the ansible playbook to deploy the containers



The screenshot shows a GitHub repository for 'star-agile-insurance-project'. The 'ansible-playbook.yml' tab is selected. The code in the playbook is:

```
1 - name : Configure Docker on EC2 Instances
2   hosts : all
3   become: true
4   connection : ssh
5   tasks :
6     - name: updating apt
7       command : sudo apt-get update
8
9     - name : Install Docker
10       command : sudo apt-get install -y docker.io
11       become : yes
12       become_user : root
13
14     - name : Start Docker Service
15       command : sudo systemctl start docker
16       become : yes
17       become_user : root
18
19     - name: Deploy Docker Container
20       command: docker run -itd -p 8084:8081 shubhamkushwah123/insure-me:3.0
```

Build the project in jenkins

```
pipeline {
    agent any

    stages {
        stage('Checkout Code') {
            steps {
                git url: 'https://github.com/manjunath348/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 manjunath348/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 manjunath348/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: 'ansible-deployment.yml'
            }
        }
    }
}
```

Write the script pipeline in jenkins

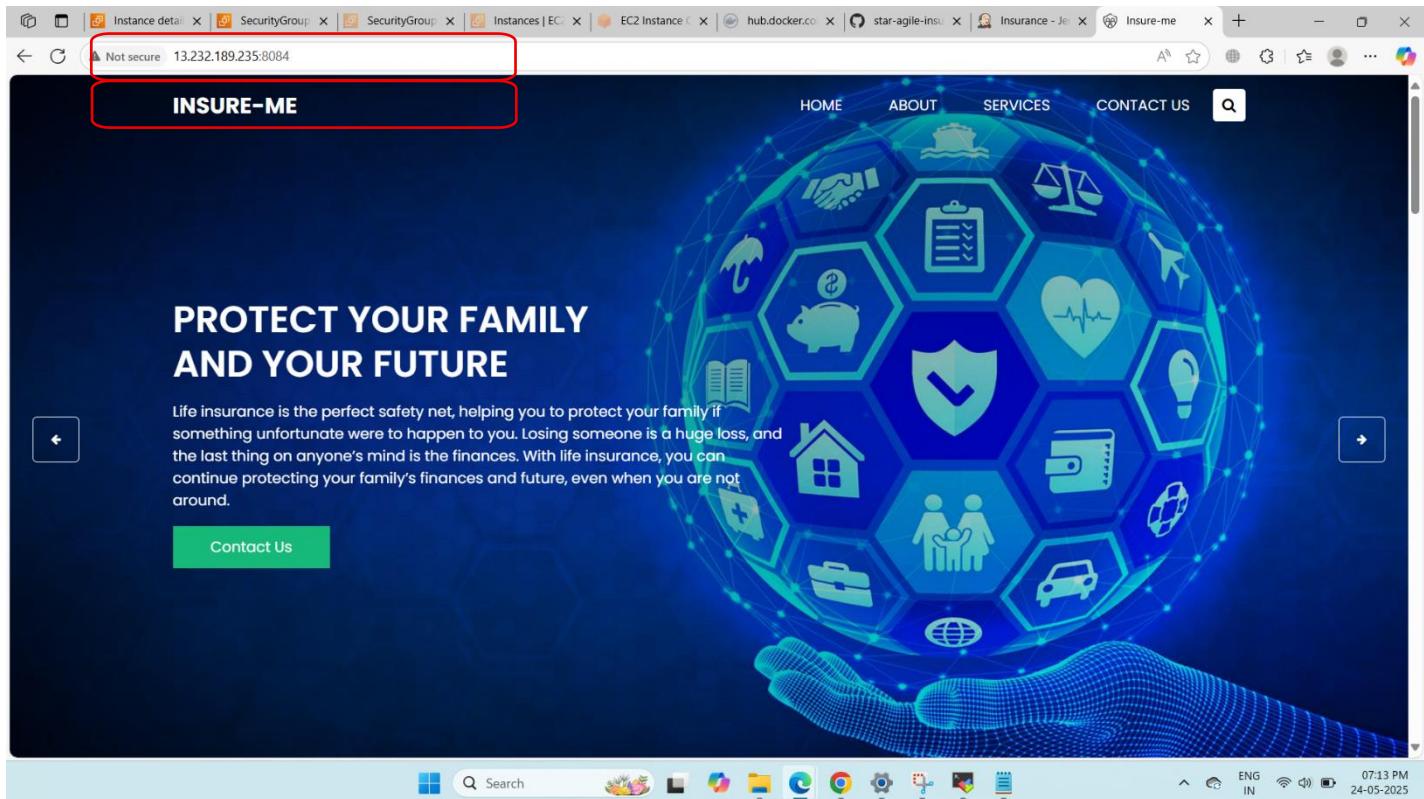
Then save and build

The screenshot shows a Jenkins pipeline job named "Insurance". The job status is "Status" (green) and the pipeline name is "Insurance". The "Build Now" button is visible. The "Permalinks" section shows the following build history:

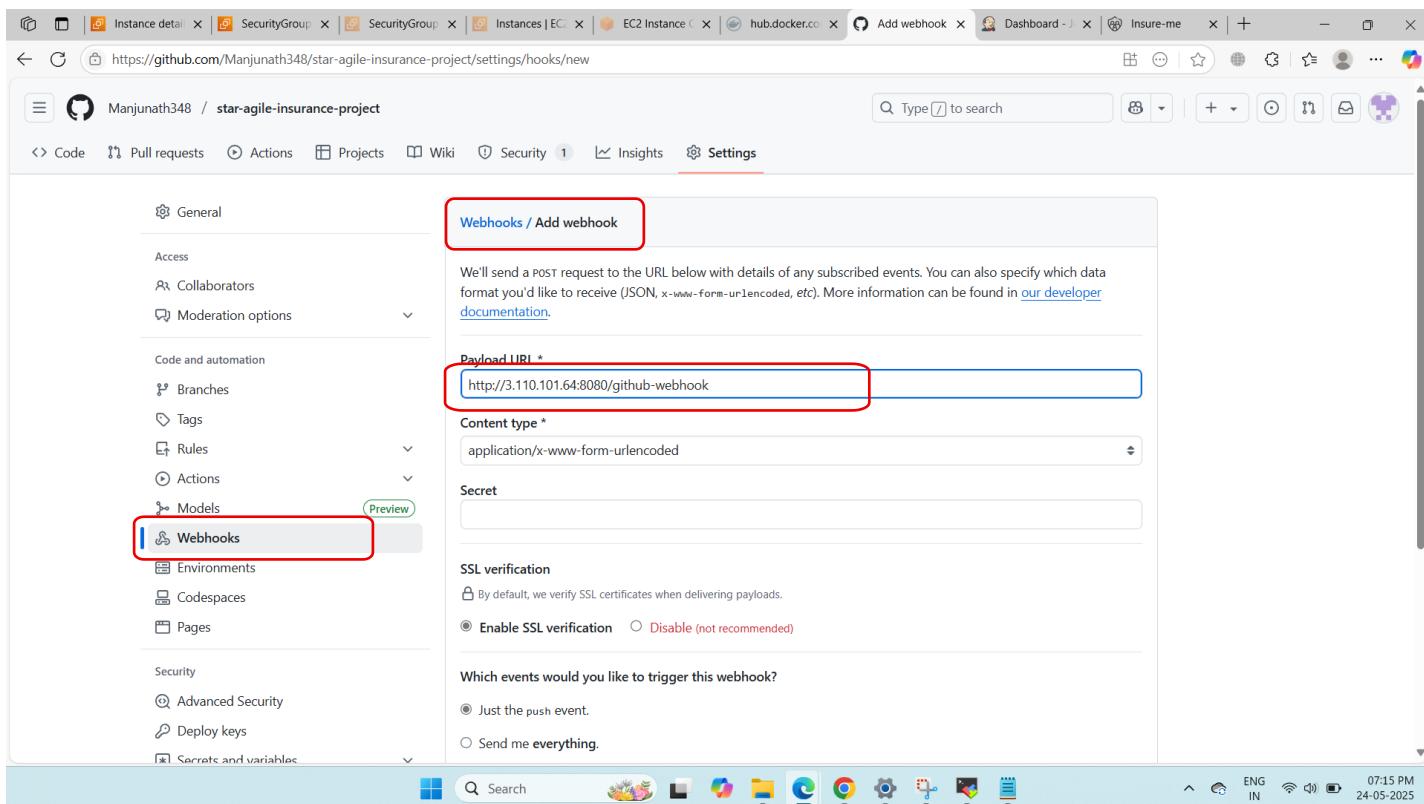
- Last build (#14), 2 min 35 sec ago
- Last stable build (#14), 2 min 35 sec ago
- Last successful build (#14), 2 min 35 sec ago
- Last failed build (#13), 8 min 58 sec ago
- Last unsuccessful build (#13), 8 min 58 sec ago
- Last completed build (#14), 2 min 35 sec ago

The "Builds" section shows a list of builds, with the most recent one being build #14, which is marked as successful. The Jenkins status bar at the bottom indicates the build was run "today" at 1:39 PM.

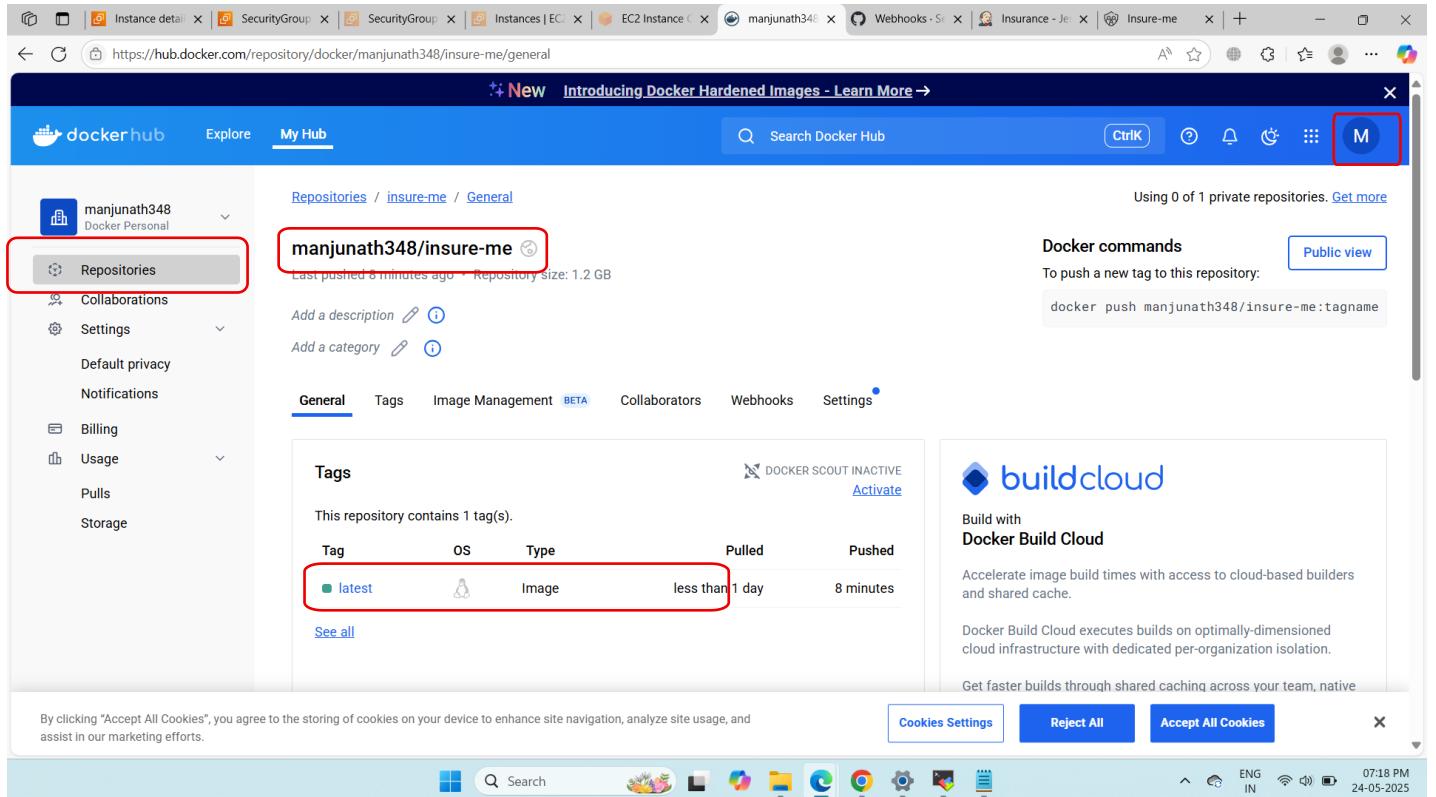
Now access the application using the production ip address on the port 8084



Add the github webhook in the settings of the project repository



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



The screenshot shows a Docker Hub repository page for 'manjunath348/insure-me'. The 'Repositories' section is highlighted with a red box. The repository 'manjunath348/insure-me' is listed, with its tag 'latest' highlighted with a red box. The 'Tags' table shows the following data:

Tag	OS	Type	Pulled	Pushed
latest	Ubuntu	Image	less than 1 day	8 minutes

On the right, a 'buildcloud' sidebar is visible, advertising Docker Build Cloud with a 'Build with Docker Build Cloud' button and a 'Activate' link for Docker Scout.