

# **Insurance Domain**

**A PROJECT REPORT**

**SUBMITTED BY**

**ANMOL UPADHAYA**

**DEVOPS Course**

**SUBMITTED TO**

**Star Agile  
APR & 2025**

# Certification Project – Insure Me

**Insure Me** is a Global leading Insurance provider based out of USA. The company offers products and services like Home Insurance, Health Insurance, Car Insurance and Life Insurances. Initially the company was using a Monolithic application architecture, As the company grown, It started facing difficulties in managing the application infrastructure and application deployments.

Insure-Me has decided to transform its monolithic application architecture to microservice application architecture and opted to go DevOps by implementing CICD pipeline and necessary automations. Insure me has decided to use AWS as primary cloud services provider to create servers, databases, and application deployments.

The company's goal is to deliver the product updates frequently to production with High

quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers.

**Following are the problems the company is facing at the moment**

- ✓ Building Complex builds is difficult
- ✓ Manual efforts to test various components/modules of the project
- ✓ Incremental builds are difficult to manage, test and deploy
- ✓ Creation of infrastructure and configure it manually is very time consuming
- ✓ Continuous manual monitoring the application is quite challenging.

Later, you need to implement Continuous Integration & Continuous Deployment using following tools:

- ✓ 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.

This project will be about how to test the services and deploy code to dev/stage/prod etc, just on a click of button.

### **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.

# Step1: Create a Terraform – Server to Deploy a DEV, PROD, AND TEST Server for Further Process.

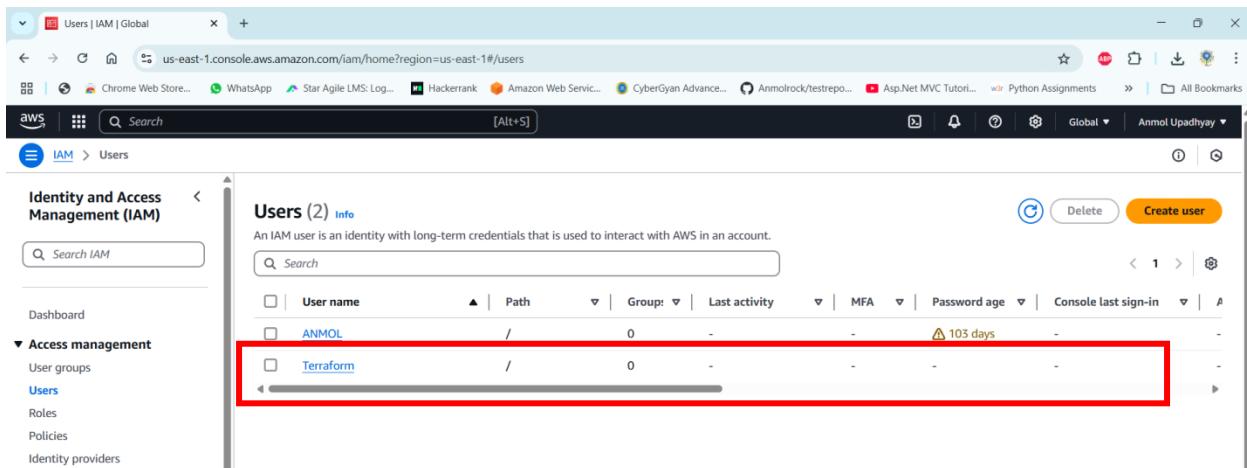
The screenshot shows the AWS CloudShell interface running on an EC2 instance. The terminal window displays the command to download the HashiCorp GPG key and the subsequent apt-get update and install process for Terraform.

```
root@Terraform-Server:~# wget -O https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2025-04-22 05:49:58-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.160.10.45, 18.160.10.71, 18.160.10.126, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.160.10.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9k) [binary/octet-stream]
Saving to: 'STDOUT'

[  0% [=====]  3.89K  --.-KB/s   in 0s
2025-04-22 05:49:58 (1023 MB/s) - written to stdout [3980/3980]

root@Terraform-Server:~# echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com \$lsb_release -cs" main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
root@Terraform-Server:~# sudo apt update && sudo apt install terraform
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [1 4.1 MB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5 652 kB]
Get:8 https://apt.releases.hashicorp.com jammy/main amd64 Packages [177 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [1 112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2499 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [409 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [18.5 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3336 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2222 kB]
```

## Step2: Create Access key and secret key in AWS for Terraform to connect



The screenshot shows the AWS IAM 'Users' page. There are two users listed: 'ANMOL' and 'Terraform'. The 'Terraform' user is highlighted with a red box. The 'ANMOL' user has a last activity of 103 days ago.

The screenshot shows the 'Create access key' step in the AWS IAM 'Users' section. Under 'Access key best practices & alternatives', the 'Command Line Interface (CLI)' option is selected and highlighted with a red box. Other options include 'Local code', 'Application running on an AWS compute service', 'Third-party service', and 'Application running outside AWS'.

The screenshot shows the AWS IAM 'Create access key' page. A green success message box at the top states: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, there's a section titled 'Access key best practices & alternatives' with three steps: Step 2 - optional, Set description tag, and Retrieve access keys (which is selected). The 'Retrieve access keys' section contains a table with two columns: 'Access key' and 'Secret access key'. The 'Access key' column shows 'AKIA34AMDIJPXJGPFH72' with a 'Show' link. The 'Secret access key' column shows a redacted string. At the bottom right of the page are 'Download .csv file' and 'Done' buttons.

## Step3: Write Terraform configuration file to create an 2 EC2 server.

The screenshot shows a code editor window displaying a Terraform configuration file. The file defines three AWS instances: 'myec2-Dev', 'myec2-TEST', and 'myec2-PROD'. Each instance is configured with specific AMI IDs, instance types (t2.micro), key names ('mykey'), subnet IDs, security group IDs, and tags. The configuration uses AWS provider blocks and resource blocks for each instance type. The entire configuration is highlighted with a large red rectangle.

```
cidr_blocks      = ["0.0.0.0/0"]
}
egress {
  from_port      = 0
  to_port        = 0
  protocol       = "-1"
  cidr_blocks    = ["0.0.0.0/0"]
}

resource "aws_instance" "myec2-Dev" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.medium"
  key_name      = "mykey"
  subnet_id     = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.s1-sg.id]
  tags = [
    { Name = "Project-EC2-Dev" }
  ]
}

resource "aws_instance" "myec2-TEST" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.micro"
  key_name      = "mykey"
  subnet_id     = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.s1-sg.id]
  tags = [
    { Name = "Project-EC2-TEST" }
  ]
}

resource "aws_instance" "myec2-PROD" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.micro"
  key_name      = "mykey"
  subnet_id     = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.s1-sg.id]
  tags = [
    { Name = "Project-EC2-PROD" }
  ]
}
```

## Step4: Then Execute “terraform init” and “terraform apply --auto-approve” Command

```
root@Terraform-Server:~/myproject# ls
aws_infra.tf  terraform.tfstate  terraform.tfstate.backup
root@Terraform-Server:~/myproject# vim aws_infra.tf
root@Terraform-Server:~/myproject# terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@Terraform-Server:~/myproject# terraform apply --auto-approve
```

Instances | EC2 | us-east-1

Last updated 1 minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Terraform-Server	i-07f0aa5c5fb527350	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-98-81-235-240.co...
Project-EC2-TEST	i-017657ad623fa13d9	Running	t2.micro	Initializing	View alarms +	us-east-1d	-
Project-EC2-dev	i-05181809c7980877	Running	t2.medium	Initializing	View alarms +	us-east-1d	-
Project-EC2-PROD	i-0732b58d91bd2f080	Running	t2.micro	Initializing	View alarms +	us-east-1d	-

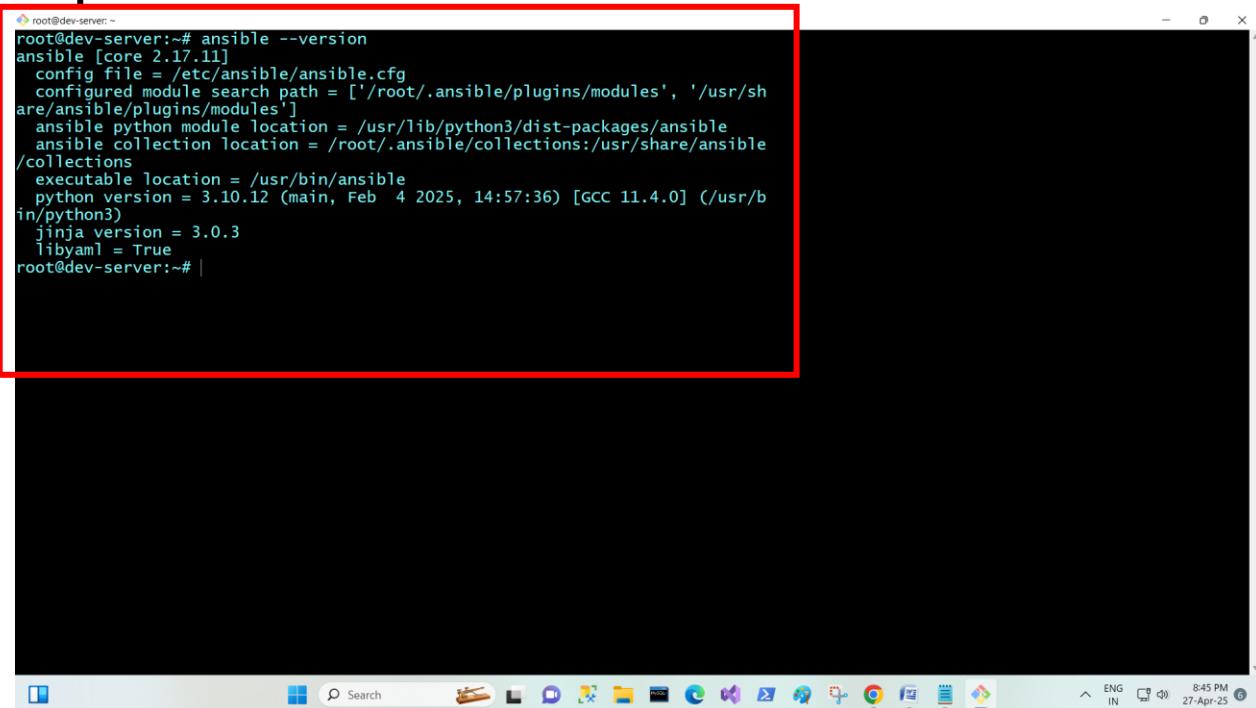
2 instances selected

Monitoring

Configure CloudWatch agent

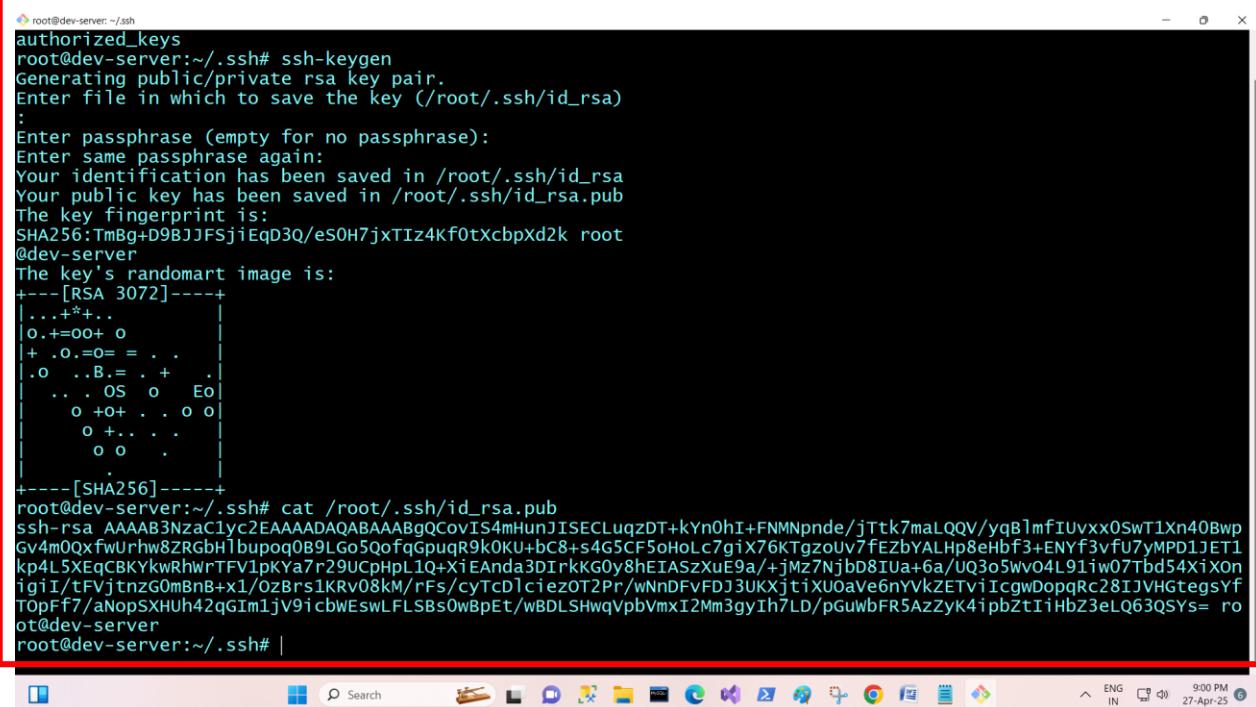
CloudShell Feedback

## Step6: Install Ansible on this server



```
root@dev-server:~# ansible --version
ansible [core 2.17.11]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Feb 4 2025, 14:57:36) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
root@dev-server:~# |
```

## Step7: Copy the ssh keys from Ansible\_controller/Dev server to Prod server and Test Server.



```
root@dev-server:~/.ssh#
authorized_keys
root@dev-server:~/.ssh# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
:
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TmBg+D9BJJFSjiEqD3Q/esO0H7jxTIz4Kf0tXcbpxd2k root
@dev-server
The key's randomart image is:
+---[RSA 3072]---+
|....+*+..|
|o.+=oo+ o |
|+ .o.=o= = ..|
|.o ..B.= . + .|
... OS o Eo |
| o +o+ . . o o|
| o +. . .|
| o o . .|
| . . .|
+---[SHA256]---+
root@dev-server:~/.ssh# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCoVIs4mHunJISECLugzDT+kYn0hI+FNMNprnde/jTtk7maL00V/yqB1mfIUvxx0SwT1Xn40Bwp
Gv4m0QxfwUrhw8ZRGbh1buopoq089LG05QofqGpuqR9k0KU+bC8+s4G5CF5oHoLc7giX76KTgouv7FeZbYALHp8eHbf3+ENyf3vfU7yMPD1JET1
kp4L5XEqCBKYkwRhwrTFV1pkYa7r29UcpHpl1Q+XiEAnda3DIrkKG0y8hEIASzxue9a+jMz7NjbD8IUa+6a/uQ3o5wvo4L91iw07Tbd54XixOn
ig1i/tFvjtnzG0mBnB+x1/OzBrs1KRV08kM/rFs/cyTcd1ciezoT2Pr/wNnDFvFDj3UKXjtixu0aVe6nYVZETviIcgwDopqrC28IJVHGtegsYf
T0pFf7/aNopSXUh42qGim1jV9icbwEswLFLSBS0wBpEt/wBDLSHwqvpbVmxi2Mm3gyIh7LD/pGuwbFR5AzzYk4ipbztIiHbZ3eLQ63QSYS=
root@dev-server:~/.ssh# |
```

```
s
root@Prod-Server:~# adduser ansiuser
Adding user `ansiuser' ...
Adding new group `ansiuser' (1001) ...
Adding new user `ansiuser' (1001) with group `ansiuser' ..
.
Creating home directory `/home/ansiuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ansiuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
root@Prod-Server:~# vim /etc/sudoers
root@Prod-Server:~# vim /etc/ssh/sshd_config
root@Prod-Server:~# systemctl restart sshd
root@Prod-Server:~# cd .ssh
root@Prod-Server:~/ssh# ls
authorized_keys
root@Prod-Server:~/ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCoVlS4mHunJISECLuqzD
T+kyn0H+FMNnpnde/jTtk7maLQQV/yqb1mfIUvxx0SwT1Xn40BwpGv4m0
Qxfwurhw8ZRGbh1bupoq0B9LGo5ofqGpuqR9k0KU+bC8+s4G5CF5oHoLc
7giX76KTgzuUv7fEZbyALHp8ehbf3+ENyf3vfU7yMPD1jET1kp4L5XEqCB
KYkwRhwrTFV1pkya7r29UCpHpl1Q+XiEAnda3dIrkkG0y8hEIAszxuE9a/
+jMz7NjbD8IUA+6a/UQ3o5Wv04L91iw07Tbd54XiXonigi/tFVjtnzG0m
BnB+x1/OzBrs1Krv08kM/rFs/cyTcdlciezOT2Pr/wNnDFvFDJ3UKXjtix
Uoave6NYvkZETViIcgwDopqRc28IJVHGtegsyfT0pFF7/aNopSXHu42qG
```

```
root@Test-Server:~# chmod 600 ~/ssh/authorized_keys
root@Test-Server:~# adduser ansiuser
Adding user `ansiuser' ...
Adding new group `ansiuser' (1001) ...
Adding new user `ansiuser' (1001) with group `ansiuser' ..
.
Creating home directory `/home/ansiuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ansiuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []
Is the information correct? [Y/n] y
root@Test-Server:~# vim /etc/sudoers
root@Test-Server:~# vim /etc/ssh/sshd_config
root@Test-Server:~# systemctl restart sshd
root@Test-Server:~# cd .ssh
root@Test-Server:~/ssh# ls
authorized_keys
root@Test-Server:~/ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCoVlS4mHunJISECLuqzD
T+kyn0H+FMNnpnde/jTtk7maLQQV/yqb1mfIUvxx0SwT1Xn40BwpGv4m0
Qxfwurhw8ZRGbh1bupoq0B9LGo5ofqGpuqR9k0KU+bC8+s4G5CF5oHoLc
7giX76KTgzuUv7fEZbyALHp8ehbf3+ENyf3vfU7yMPD1jET1kp4L5XEqCB
KYkwRhwrTFV1pkya7r29UCpHpl1Q+XiEAnda3dIrkkG0y8hEIAszxuE9a/
+jMz7NjbD8IUA+6a/UQ3o5Wv04L91iw07Tbd54XiXonigi/tFVjtnzG0m
BnB+x1/OzBrs1Krv08kM/rFs/cyTcdlciezOT2Pr/wNnDFvFDJ3UKXjtix
```

# Here we successfully connect to Test Server and Prod Server

## By: → ssh ansiuser@test\_server\_ip / prod\_server\_ip

```
ansiuser@Test-Server:~  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Sun Apr 27 16:23:27 UTC 2025  
  
System load: 0.0 Processes:  
116  
Usage of /: 22.4% of 7.57GB Users logged in:  
2  
Memory usage: 22% IPv4 address for eth0  
: 10.0.1.206  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
New release '24.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sun Apr 27 16:23:27 2025 from 10.0.1.101  
ansiuser@Test-Server:~$
```

```
ansiuser@Prod-Server:~  
Connection to 10.0.1.206 closed.  
ansiuser@dev-server:~$ ssh ansiuser@10.0.1.78  
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1024-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Sun Apr 27 16:24:39 UTC 2025  
  
System load: 0.0 Processes: 118  
Usage of /: 22.4% of 7.57GB Users logged in: 2  
Memory usage: 22% IPv4 address for eth0: 10.0.1.78  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
New release '24.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sun Apr 27 16:20:33 2025 from 10.0.1.101  
ansiuser@Prod-Server:~$ |
```

```
ansiuser@Prod-Server:~$ exit
logout
root@Prod-Server:~# exit
logout
root@Prod-Server:/# history
 1  clear
 2  echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCovIS4m
HunJISECLuqzDT+kYn0hi+FNMNpnde/jTtk7maLQQV/yqB1mfIUVxx0SwT
1Xn40BwpGv4m0QxfwUrhw8ZRgbH1bupoo0891.Go5QofofqGpuqR9k0Ku+bC8
+e4G5CF5oHoLc7giX76KTgzoUv7fEZbYALH8eHbf3+ENYf3vfu7yMPD1J
ET1kp4l5xEqC8KkWrwTrTV1pkYa7r29UcpHP1l0+XiAnda3DIrrKG0y
8hIASxue9a/+jMz7Njb8IUa+6a/UQ3o5Wv04L91iw07Tbd54XixOnig
i/tFvjtnzg0mB+x1/0zBrsLKRv08km/rFs/cytcDlciezOT2Pr/wNxD
FvFDJ3UKXjtiXu0ave6nYVkZETViIcgwDopqRc28TJVHgtsgYfTopFF7/
aNoPSXHUn42qGIm1jv9icbwEswLFLSBs0wBpEt/wBDLSHwqVpbVmXI2Mm3
gyIh7LD/pgUwbFR5Azzyk4ipbztiHbZ3eLQ63QSs= root@dev-server
r" > ~/.ssh/authorized_keys
 3  cd .ssh
 4  ls
 5  cd ..
 6  clear
 7  chmod 600 ~/.ssh/authorized_keys
 8  adduser ansiuser
 9  vim /etc/sudoers
10  vim /etc/ssh/sshd_config
11  systemctl restart sshd
12  cd .ssh
13  ls
14  cat authorized_keys
15  clear
16  cd /
17  sudo su -
18  history
root@Prod-Server:/# |
```

```
root@dev-server:/# history
 1  clear
 2  sudo apt update
 3  clear
 4  sudo apt install software-properties-common
 5  sudo add-apt-repository --yes --update ppa:ansible/ansible
 6  sudo apt install ansible -y
 7  clear
 8  ansible version
 9  clear
10  ansible --version
11  clear
12  adduser ansiuser
13  vim /etc/sudoers
14  clear
15  cd .ssh
16  ls
17  ssh-keygen
18  cat /root/.ssh/id_rsa.pub
19  ls
20  cat athonized_keys
21  cat id_rsa.pub
22  cd /
23  sudo su -
24  clear
25  cd .ssh
26  ls
27  su - ansiuser
28  clear
29  su - ansiuser
30  history
root@dev-server:/# |
```

## Step8: Create a Playbook for Install and set up devops tools for localhost

```
root@dev-server:/# 04:4e42:79:::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.30.133|:4
43... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyr 100%[=====] 3.10K --.-KB/s in 0s

2025-04-27 16:49:34 (59.5 MB/s) - '/usr/share/keyrings/jenkins-
-keyring.asc' saved [3175/3175]

root@dev-server:/# echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

root@dev-server:/# vim playbook-install.yml
root@dev-server:/# ansible-playbook playbook-install.yml
[WARNING]: Unable to parse /root/myinventory as an inventory s
ource
[WARNING]: No inventory was parsed, only implicit localhost is
available
[WARNING]: provided hosts list is empty, only localhost is ava
ilable. Note that
the implicit localhost does not match 'all'

PLAY [Install and set up devops tools] ****
****

TASK [Gathering Facts] ****
ok: [localhost]

TASK [Update the apt repo] ****
****
```

```
root@dev-server:/# ansible-playbook playbook-install.yml
[WARNING]: Unable to parse /root/myinventory as an inventory s
ource
[WARNING]: No inventory was parsed, only implicit localhost is
available
[WARNING]: provided hosts list is empty, only localhost is ava
ilable. Note that
the implicit localhost does not match 'all'

PLAY [Install and set up devops tools] ****
****

TASK [Gathering Facts] ****
ok: [localhost]

TASK [Update the apt repo] ****
changed: [localhost]

TASK [Install multiple packages] ****
ok: [localhost] => (item=git)
changed: [localhost] => (item=docker.io)
changed: [localhost] => (item=openjdk-17-jdk)

TASK [install jenkins] ****
changed: [localhost]

TASK [start Jenkins and docker service] ****
ok: [localhost] => (item=jenkins)
ok: [localhost] => (item=docker)

PLAY RECAP ****
localhost : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@dev-server:/#
```

## Step9: Now access the Jenkins Server -> Setup the Jenkins dashboard.

The image consists of two screenshots of a web browser displaying the Jenkins Setup Wizard. Both screenshots show the same basic layout with a red box highlighting specific areas of interest.

**Screenshot 1: Customize Jenkins**

This screenshot shows the 'Customize Jenkins' step of the setup wizard. It features two main options:

- Install suggested plugins**: A box containing the text "Install plugins the Jenkins community finds most useful." This box is highlighted with a red border.
- Select plugins to install**: A box containing the text "Select and install plugins most suitable for your needs."

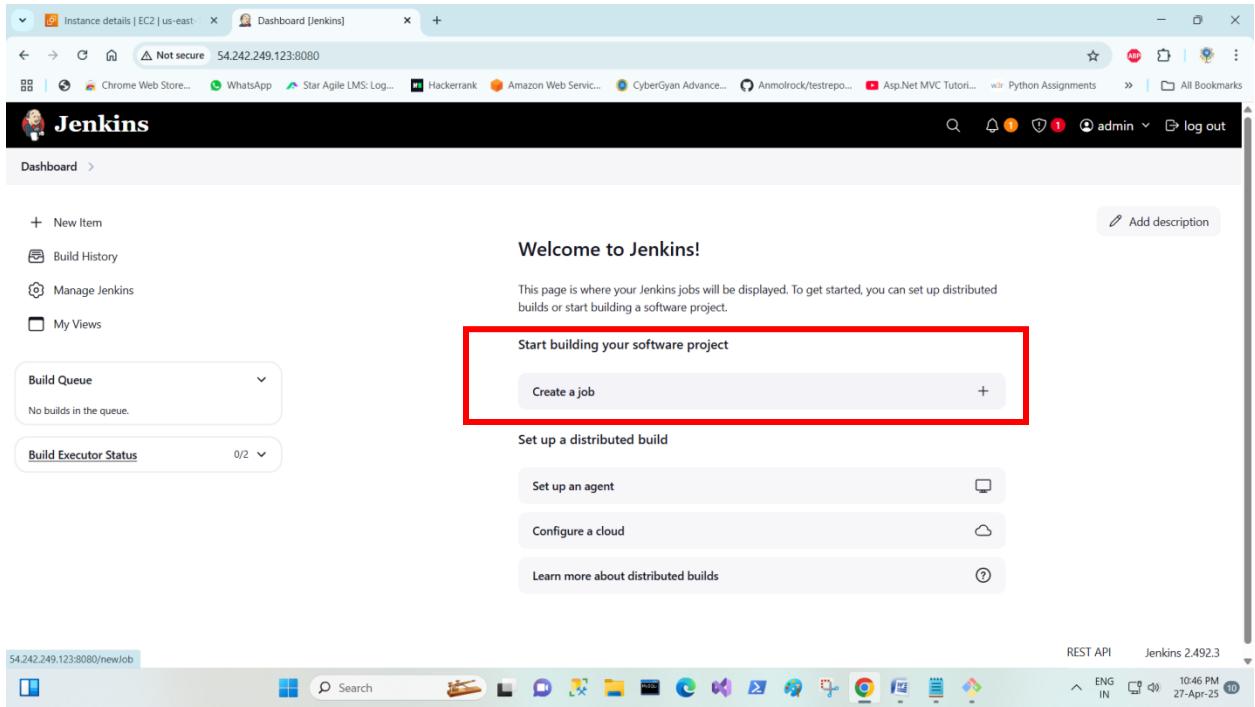
Jenkins 2.492.3 is displayed at the bottom left of the window.

**Screenshot 2: Getting Started**

This screenshot shows the 'Getting Started' step of the setup wizard. It contains fields for entering user information:

- Username: admin
- Password:  (containing four dots)
- Confirm password:  (containing four dots)
- Full name: admin
- E-mail address: admin@gmail.com

At the bottom right of the form are two buttons: "Skip and continue as admin" and "Save and Continue". The entire "Getting Started" form area is highlighted with a red border.



## Step10: Create a new job and Create a pipeline

The screenshot shows a terminal window with a black background and white text. The user is root on a "dev-server". The first command, "cat /var/lib/jenkins/secrets/initialAdminPassword", is used to find the initial admin password, which is displayed as "9d003e513ecb47aab39b1c5bd4867f88". The second command, "chmod -R 777 /var/run/docker.sock", is used to change the permissions of the Docker socket. The terminal window has a red rectangular box highlighting the first command and its output.

```
root@dev-server:/# cat /var/lib/jenkins/secrets/initialAdminPassword
9d003e513ecb47aab39b1c5bd4867f88
root@dev-server:/# chmod -R 777 /var/run/docker.sock
root@dev-server:/# |
```

Dashboard > Insurance-Project > Pipeline Syntax

withCredentials

IntelliJ IDEA GDSDL

Secret values are masked on a best-effort basis to prevent *accidental* disclosure. Multiline secrets, such as the contents of a SSH private key file, are not masked. See the inline help for details and usage guidelines.

**Bindings**

**Secret text**

Variable: DOCKER\_HUB\_PASSWD

Credentials: DOCKER\_HUB\_PASSWD

+ Add

Add

Generate Pipeline Script

```
withCredentials([string(credentialsId: 'DOCKER_HUB_PASSWD', variable: 'DOCKER_HUB_PASSWD')) {
    // some block
}
```

Instance details | EC2 | us-east-... | Health-Care-Project Config | Jenkins Pipeline Syntax Snippet Generator | anmol792 (anmol upadhyay) | Docker Home

Dashboard > Health-Care-Project > Pipeline Syntax

Jenkins Credentials Provider: Jenkins

Secret text

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

Secret:

ID: DOCKER\_HUB\_PASSWD

Description: DOCKER\_HUB\_PASSWD

Cancel Add

Jenkins 2.492.3

Instance details | EC2 | us-east- > Insurance-Project Config [Jenkins] > Anmolrock/star-agile-insurance... > +

Not secure 54.242.249.123:8080/job/Insurance-Project/configure

Dashboard > Insurance-Project > Configuration

## Configure

- General
- Triggers
- Pipeline**
- Advanced

Script ?

```
1 v pipeline{
2
3   agent none
4 v tools{
5     maven 'mymaven'
6   }
7 v   stages{
8     stage('checkout code')
9 v       {
10       agent any
11       steps{
12         git 'https://github.com/Anmolrock/star-agile-insurance-project.git'
13       }
14     }
15 }
```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

Advanced

**Save** **Apply**



Instance details | EC2 | us-east- > Insurance-Project Config [Jenkins] > Pipeline Syntax Snippet Generator > +

Not secure 54.242.249.123:8080/job/Insurance-Project/configure

Dashboard > Insurance-Project > Configuration

## Configure

- General
- Triggers
- Pipeline**
- Advanced

Pipeline script

try sample Pipeline...

```
1 v pipeline{
2
3   agent none
4 v tools{
5     maven 'mymaven'
6   }
7 v   stages{
8     stage('push the image to dockerhub')
9 v       {
10       agent any
11       steps{
12         withCredentials([string(credentialsId: 'DOCKER_HUB_PASSWD', variable: 'DOCKER_HUB_PASSWD')])
13         {
14           sh 'docker login -u anmol792 -p ${DOCKER_HUB_PASSWD}'
15         }
16         sh 'docker tag myimage:project3 anmol792/myimage:project3'
17         sh 'docker push anmol792/myimage:project3'
18       }
19     }
20   }
21 }
```

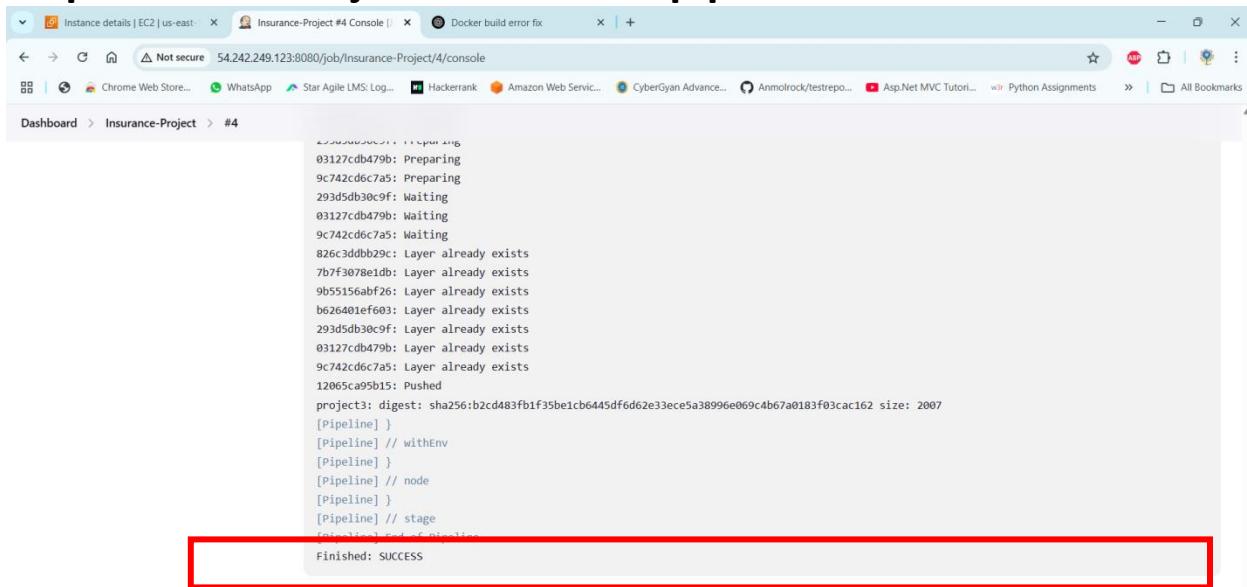
Use Groovy Sandbox ?

[Pipeline Syntax](#)

**Save** **Apply**

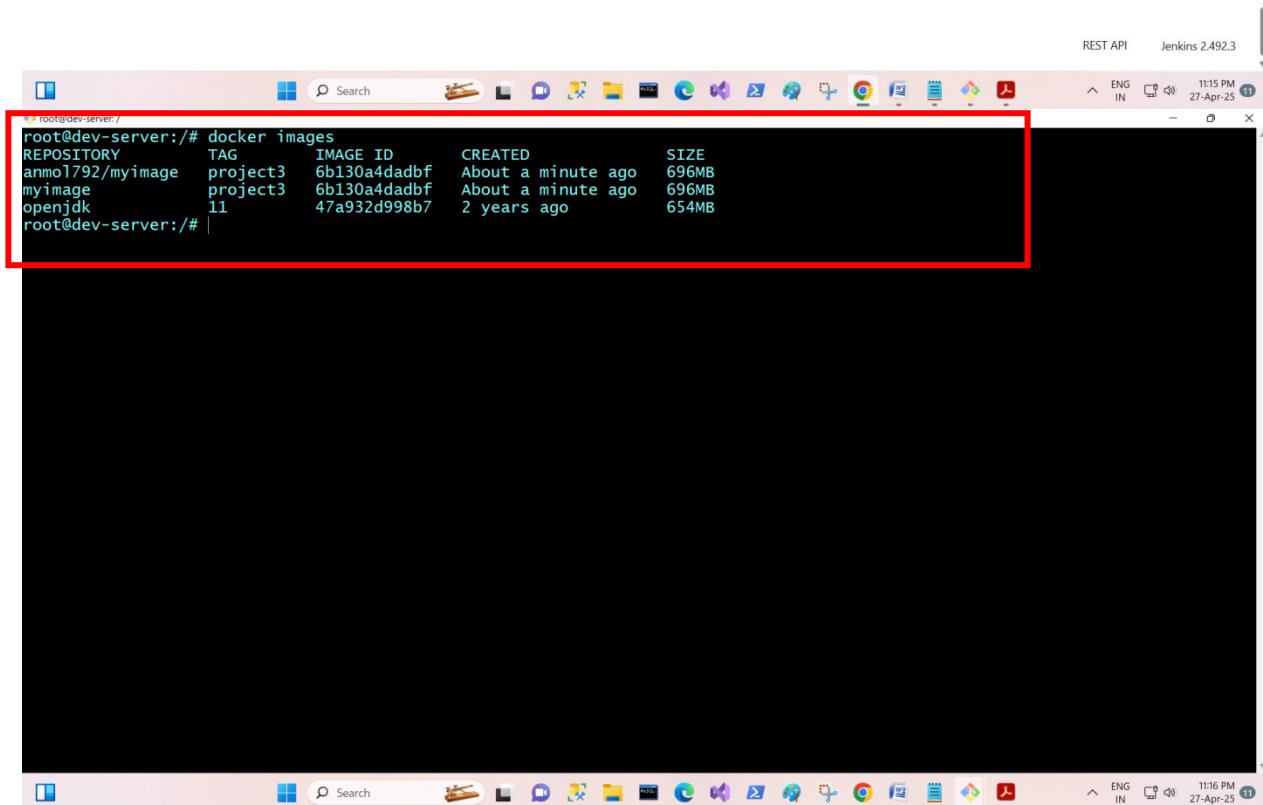


## Step11: Save the job and run the pipeline



```
2024-04-27T23:11:48.666Z [Pipeline] ::step[git]
03127cdb479b: Preparing
9c742cd6c7a5: Preparing
293d5db30c9f: Waiting
03127cdb479b: Waiting
9c742cd6c7a5: Waiting
826c3ddb29c: Layer already exists
7bf7f3078e1db: Layer already exists
9b55156abf26: Layer already exists
b626401ef603: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9c742cd6c7a5: Layer already exists
12065ca095b15: Pushed
project3: digest: sha256:b2cd483fb1f35be1cb6445df6d2e33ece5a38996e069c4b67a0183f03cac162 size: 2008
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] }

Finished: SUCCESS
```



```
root@dev-server:/# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
amol792/myimage    project3   6b130a4dadbf  About a minute ago  696MB
myimage             project3   6b130a4dadbf  About a minute ago  696MB
openjdk             11        47a932d998b7   2 years ago   654MB
root@dev-server:/# |
```

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Docker Hub' at hub.docker.com. The left sidebar shows the user's profile 'anmol792' and navigation links like 'Repositories', 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main content area is titled 'Repositories' and displays two entries:

Name	Last Pushed	Contains	Visibility	Scout
anmol792/myimage	2 minutes ago	IMAGE	Public	Inactive
anmol792/nginximage01	about 1 month ago	IMAGE	Public	Inactive

A red box highlights the first repository, 'anmol792/myimage'.

## Step12: Create the inventory file and ansible.cfg file

The screenshot shows a terminal window with a black background and white text. The command entered is:

```
[webserver]
44.222.202.7 #Test Server Public ip
34.207.208.0 #Prod Server Public ip
```

A red box highlights the first line of the command, '[webserver]'. The bottom status bar shows the terminal is in 'INSERT' mode with coordinates '2,35' and 'All'.

## **Step13: Then Create a Playbook for Prod Server and Test Server and then Run a playbook**

```
root@dev-server:/  
- name: Install and deploy of test and prod server  
  hosts: webserver  
  become: true  
  tasks:  
    - name: Update the apt repo  
      command: apt-get update  
    - name: Install multiple packages  
      package: name={{item}} state=present  
      loop:  
        - git  
        - docker.io  
        - openjdk-17-jdk  
    - name: start Jenkins and docker service  
      service: name={{item}} state=started  
      loop:  
        - docker  
    - name: Deploy on Test server  
      command: sudo docker run -d -P anmol792/myimage:project3  
      delegate_to: 44.222.202.7  
    - name: Perform testing on Test server  
      command: echo "Testing on Server"  
    - name: Deploy on Prod Server  
      command: sudo docker run -d -P anmol792/myimage:project3  
      delegate_to: 34.207.208.0  
  
-- INSERT --
```

```
ansiuser@dev-server:~$ PLAY [Install and deploy of test and prod server] ****
skipping: no hosts matched

PLAY RECAP ****
ansiuser@dev-server:~$ vim ansible.cfg
ansiuser@dev-server:~$ ansible-playbook playbook-Deploy.yml

PLAY [Install and deploy of test and prod server] ****
****

TASK [Gathering Facts] ****
*****
[WARNING]: Platform linux on host 44.222.202.7 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [44.222.202.7]
[WARNING]: Platform linux on host 34.207.208.0 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [34.207.208.0]

TASK [Update the apt repo] ****
****
```

```
ansiuser@dev-server:~$ ok: [34.207.208.0]

TASK [Update the apt repo] ****
*****
changed: [34.207.208.0]
changed: [44.222.202.7]

TASK [Install multiple packages] ****
ok: [44.222.202.7] => (item=git)
ok: [34.207.208.0] => (item=git)
changed: [44.222.202.7] => (item=docker.io)
changed: [34.207.208.0] => (item=docker.io)
changed: [44.222.202.7] => (item=openjdk-17-jdk)
changed: [34.207.208.0] => (item=openjdk-17-jdk)

TASK [start Jenkins and docker service] ****
ok: [44.222.202.7] => (item=docker)
ok: [34.207.208.0] => (item=docker)

TASK [Deploy on Test server] ****
changed: [44.222.202.7]
changed: [34.207.208.0 -> 44.222.202.7]

TASK [Perform testing on Test server] ****
changed: [34.207.208.0]
changed: [44.222.202.7]

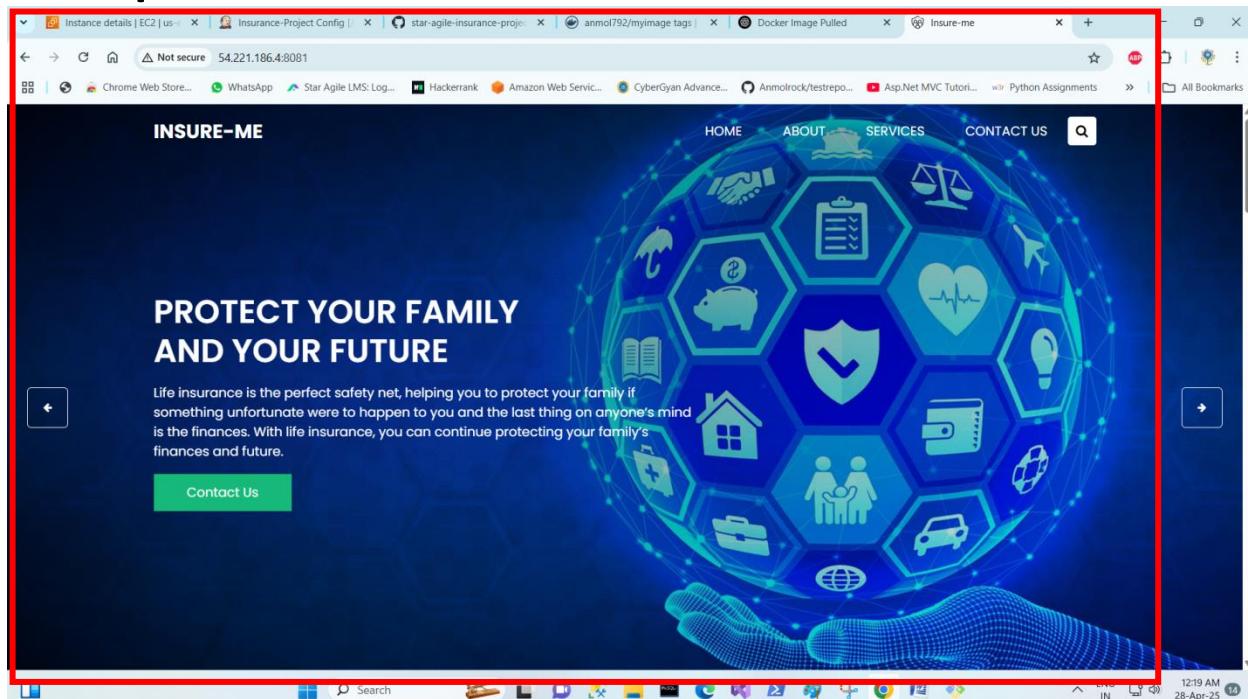
TASK [Deploy on Prod Server] ****
changed: [34.207.208.0]
changed: [44.222.202.7 -> 34.207.208.0]

PLAY RECAP ****
34.207.208.0 : ok=7    changed=5    unreachable=0   failed=0    skipped=0    rescued=0    ignored=0
44.222.202.7 : ok=7    changed=5    unreachable=0   failed=0    skipped=0    rescued=0    ignored=0
```

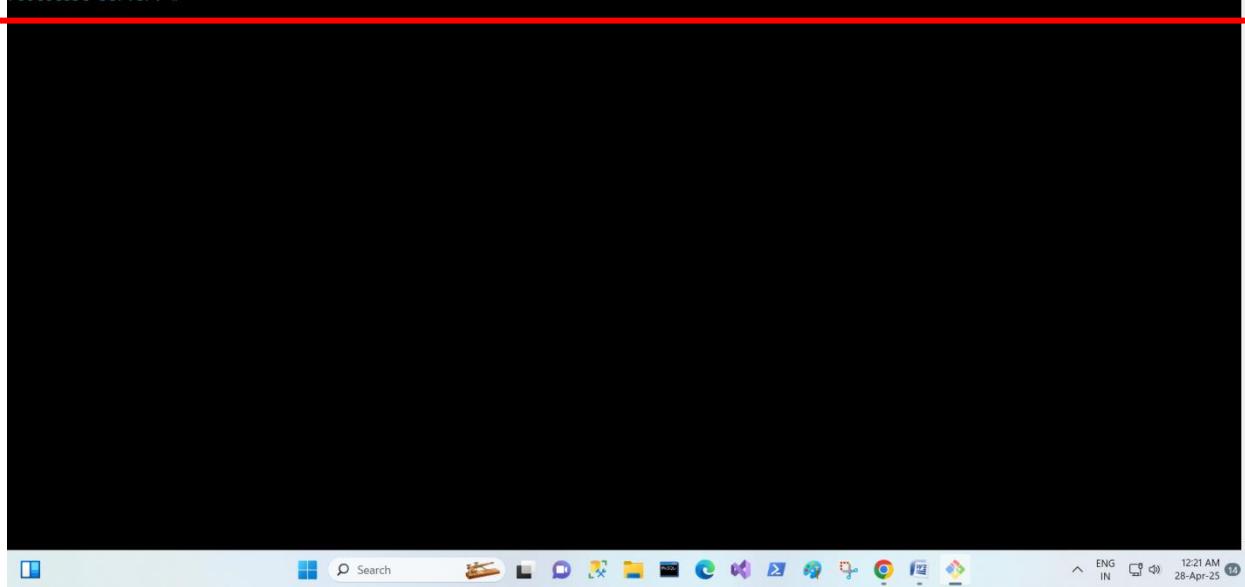
## Step14: Go to the Test Server and Prod Server

```
root@ip-10-0-1-206:~  
ng in 70 ms. Found 1 JPA repository interfaces.  
2025-04-27 18:48:59.230 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081  
(http)  
2025-04-27 18:48:59.266 INFO 1 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]  
2025-04-27 18:48:59.267 INFO 1 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomca  
t/9.0.65]  
2025-04-27 18:48:59.453 INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplic  
ationContext  
2025-04-27 18:48:59.454 INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializa  
tion completed in 3188 ms  
2025-04-27 18:49:00.298 INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...  
2025-04-27 18:49:00.797 INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.  
2025-04-27 18:49:00.940 INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnit  
info [name: default]  
2025-04-27 18:49:01.053 INFO 1 --- [           main] org.hibernate.Version : HHH000412: Hibernate ORM core version  
5.6.11.Final  
2025-04-27 18:49:01.388 INFO 1 --- [           main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotat  
ions {5.1.2.Final}  
2025-04-27 18:49:01.709 INFO 1 --- [           main] org.hibernate.dialect.Dialect : HHH000400: using dialect: org.hibernat  
e.dialect.H2Dialect  
2025-04-27 18:49:02.867 INFO 1 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implement  
ation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
2025-04-27 18:49:02.888 INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory f  
or persistence unit 'default'  
2025-04-27 18:49:03.655 WARN 1 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by  
default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable th  
is warning  
2025-04-27 18:49:04.058 INFO 1 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resour  
ce [static/index.html]  
2025-04-27 18:49:04.465 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http)  
with context path ''  
2025-04-27 18:49:04.488 INFO 1 --- [           main] c.p.s.insureme.InsureMeApplication : Started InsureMeApplication in 9.453 s  
econds (JVM running for 10.536)  
2025-04-27 18:49:08.386 INFO 1 --- [nio-8081-exec-1] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet  
'dispatcherServlet'  
2025-04-27 18:49:08.388 INFO 1 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : Initializing servlet 'dispatcherServlet'  
2025-04-27 18:49:08.391 INFO 1 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : completed initialization in 3 ms
```

## →Output



```
root@test-server:~# docker ps -a --latest
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
0c1057e9914a        anmol792/myimage:project3   "java -jar /app.jar"   41 seconds ago    Up 41 seconds     8080/tcp, 0.0.0.0:8081->8081/tcp, ::1:8081->8081/tcp
root@test-server:~#
```



A screenshot of a Microsoft Edge browser window. The address bar shows the URL '54.221.186.4:8081'. The page itself is the homepage of a website called 'INSURE-ME'. The header features a navigation menu with links for 'HOME', 'ABOUT', 'SERVICES', and 'CONTACT US'. Below the header is a large, stylized graphic of a hand holding a globe, with various icons representing different services like health, finance, and travel arranged around it. A central text area reads 'PROTECT YOUR FAMILY AND YOUR FUTURE'. Below this text is a paragraph about life insurance and a green 'Contact Us' button. The browser's taskbar at the bottom shows other open tabs and pinned sites like WhatsApp, Star Agile LMS, and Amazon Web Services.

## Step15: Monitoring Using Prometheus and Grafana

The screenshot shows the Prometheus web interface with the following details:

- Header:** The URL is 54.163.9.153:9090/query. The browser tabs include "Instance details | EC2 | us-east-", "Phase 1 SA2412: Capstone Proj...", "Dashboard [Jenkins]", "Prometheus Time Series Collect...", and "Prometheus config review".
- Toolbar:** Includes links to "Query", "Alerts", "Status", and various system icons.
- Search Bar:** A large input field with placeholder text "Enter expression (press Shift+Enter for newlines)".
- Execution Options:** Buttons for "Table", "Graph", "Explain", and "Execute".
- Time Selection:** A range selector labeled "Evaluation time" with arrows for navigation.
- Result Area:** Displays the message "No data queried yet".
- Buttons:** A "Add query" button.

Screenshot of the Prometheus Time Series Collector interface showing target health. A red box highlights the Jenkins\_Server entry.

Jenkins_Server		1 / 1 up	
Endpoint	Labels	Last scrape	State
<a href="http://18.204.5.140:8080/prometheus">http://18.204.5.140:8080/prometheus</a>	instance="54.164.183.224:8080" job="Jenkins_Server"	8.653s ago	21ms UP

prometheus		1 / 1 up	
Endpoint	Labels	Last scrape	State
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	instance="localhost:9090" job="prometheus"	5.106s ago	4ms UP

```
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-05-04 04:53:19 UTC; 5s ago
     Docs: http://docs.grafana.org
 Main PID: 2649 (grafana)
   Tasks: 6 (linit: 1129)
  Memory: 211.4M
    CPU: 1.158s
   CGroup: /system.slice/grafana-server.service
           └─2649 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:de

Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.752984873Z level=info msg="Executing migration" id="create_index_IDX_lo
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.7539445Z level=info msg="Migration successfully executed" id="create_in
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.761265368Z level=info msg="Executing migration" id="copy_login_attempt
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.76186048Z level=info msg="Migration successfully executed" id="copy_log
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.76947319Z level=info msg="Executing migration" id="drop_login_attempt_t
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.770346315Z level=info msg="Migration successfully executed" id="drop_lo
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.77991212Z level=info msg="Executing migration" id="create_user_auth_ta
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.78051267Z level=info msg="Migration successfully executed" id="create_u
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.788020937Z level=info msg="Executing migration" id="create_index_IDX_us
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.79019446Z level=info msg="Migration successfully executed" id="create_i
```

1 lines 1-21/21 (END)

Not secure 54.157.244.211:3000/connections/datasources/edit/fejnza1ko3aioe

Home > Connections > Data sources > prometheus

Incremental querying (beta)

Disable recording rules (beta)

Other

Custom query parameters Example: max\_source\_resolution=5m&timeout=10s

HTTP method POST

Use series endpoint

Exemplars [+ Add](#)

Successfully queried the Prometheus API.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

Delete Save & test



Not secure 54.157.244.211:3000/dashboard/import

Home > Dashboards > Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from [Grafana.com](#)

Published by haryan

Updated on 2023-08-24 15:04:53

Options

Name Jenkins: Performance and Health Overview

Folder [Dashboards](#)

Unique identifier (UID)  
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

haryan-jenkins [Change uid](#)

Prometheus [prometheus1](#)

Import Cancel



