# NGO IAC Provisioning Report

Course Name: PROJECT WORK - II

***Institution Name:*** Medicaps University – Datagami Skill Based Course

*Student Name(s) & Enrolment Number(s):*

| Sr no | Student Name | Enrolment Number |
|---|---|---|
| 1 | Aayush Banwadikar | EN22ME304002 |
| 2 | Divyanshu Mehta | EN22EL301024 |
| 3 | Aditya Kumar Sahu | EN22ME304005 |
| 4 | Vinay Patidar | EN22CS3011085 |
| 5 | Yatendra Verma | EN22ME304118 |

*Group Name: 02D10*

*Project Number: : D0-15*

*Industry Mentor Name: Mr. Vaibhav*

*University Mentor Name: Prof. Avnesh Joshi*

*Academic Year: 2026*

## Table of Contents

## 1. Problem Statement & Objectives.

### 1.1. Problem Statement

Non-profit organizations often face challenges in deploying and managing their applications due to manual server configuration, inconsistent environments, and deployment errors. Manual provisioning of infrastructure and application deployment increases downtime, configuration drift, and operational overhead.
The project aims to automate infrastructure provisioning and application deployment using Infrastructure as Code (IaC) tools such as Terraform and Ansible, ensuring consistency, scalability, and repeatability.

### 1.2. Project Objectives

- Automate AWS EC2 infrastructure provisioning using Terraform.
- Automate server configuration using Ansible.
- Deploy Spring Boot application as WAR on Apache Tomcat.
- Configure MySQL database for application persistence.
- Eliminate manual configuration errors.
- Enable repeatable and scalable deployment.

### 1.3. Scope of the Project

- AWS EC2 instance provisioning.
- Security Group configuration.
- Tomcat installation and configuration.
- WAR file deployment automation.
- Database setup and integration.
- Infrastructure automation using IaC.

## 2. Proposed Solution

The proposed solution uses Infrastructure as Code (IaC) principles to automate the provisioning and configuration of cloud infrastructure and application deployment.

- Terraform provisions AWS EC2 instances and networking resources.
- Ansible configures the EC2 instance:

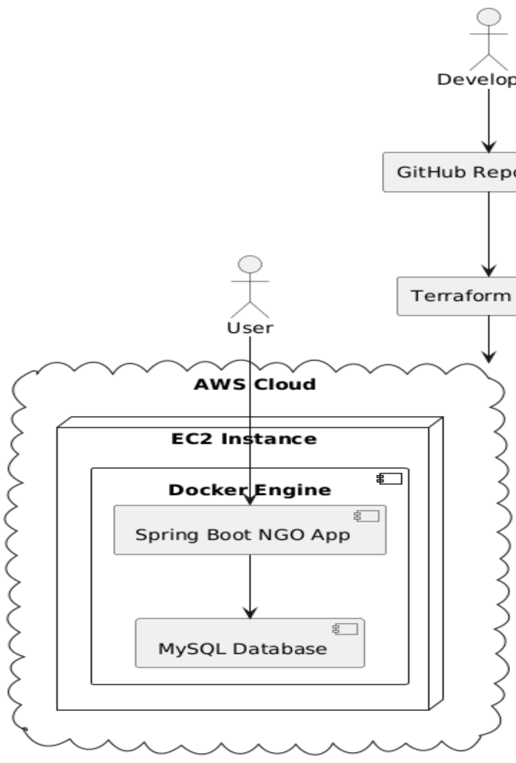
➔ Installs Java
➔ Installs Apache Tomcat
➔ Deploys WAR file
➔ Starts Tomcat service
➔ MySQL database is configured.
➔ Application becomes accessible via public IP.
➔ This ensures a fully automated and repeatable deployment pipeline.

### 2.1. Key Features

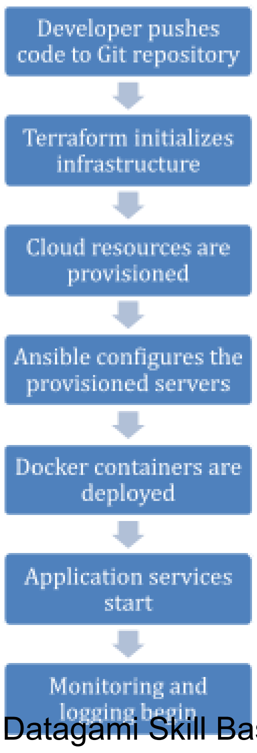
- Automated EC2 provisioning
- Infrastructure as Code implementation
- Configuration management using Ansible
- Spring Boot WAR deployment
- Apache Tomcat server configuration
- MySQL database integration
- Secure SSH-based automation
- Cloud-based deployment

### 2.2. Overall Architecture / Workflow

- Developer pushes code to GitHub
- Terraform provisions EC2 on AWS
- Ansible installs required dependencies
- WAR file is deployed to Tomcat
- MySQL database configured
- Application accessible via public IP.



**Architecture Diagram**


Developer pushes code to Git repository
Terraform initializes infrastructure
Cloud resources are provisioned
Ansible configures the provisioned servers
Docker containers are deployed
Application services start
Monitoring and logging begin

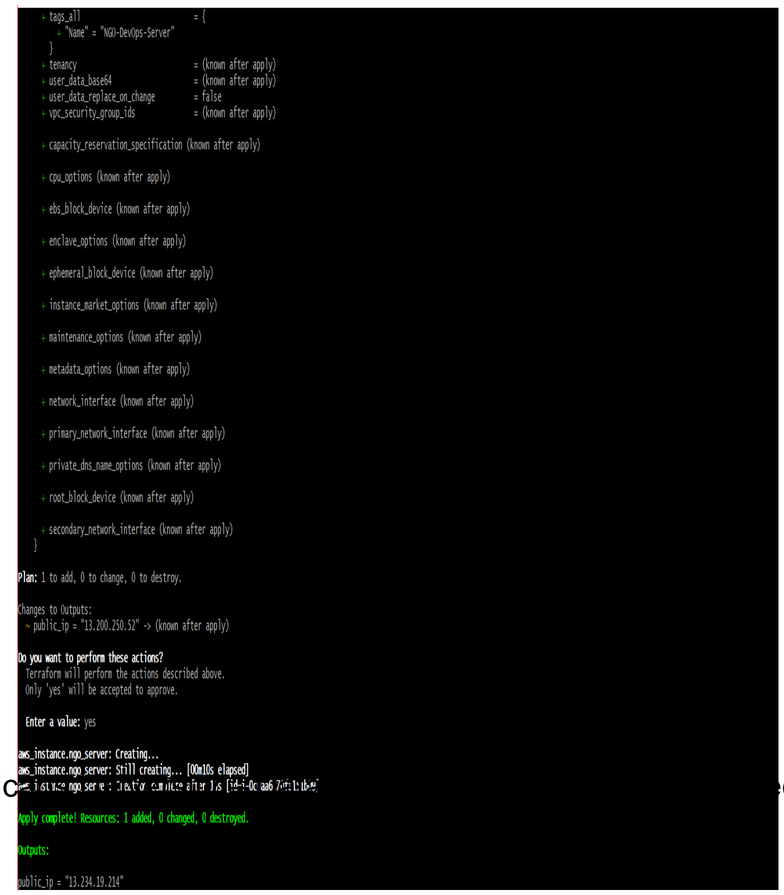
**Workflow Diagram**

### 2.3. Tools & Technologies Used

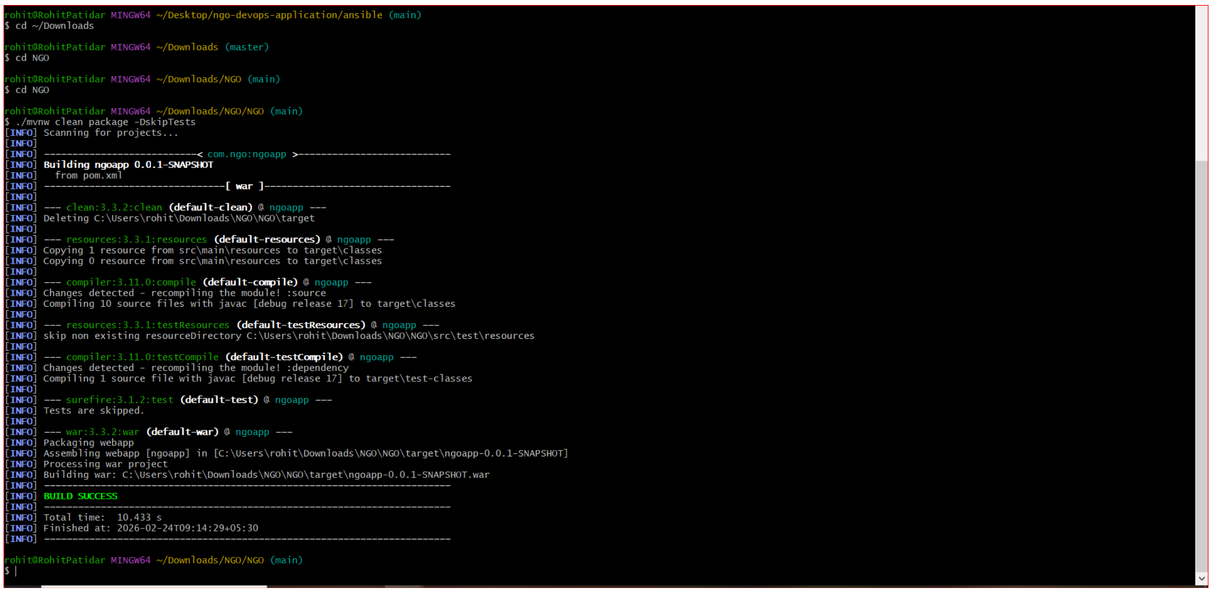
- AWS EC2
- Cloud Infrastructure
- Terraform
- Infrastructure Provisioning
- Ansible
- Apache Tomcat
- Spring Boot
- MySQL
- GitHub
- Version Control
- Docker

## 3. Result & Output

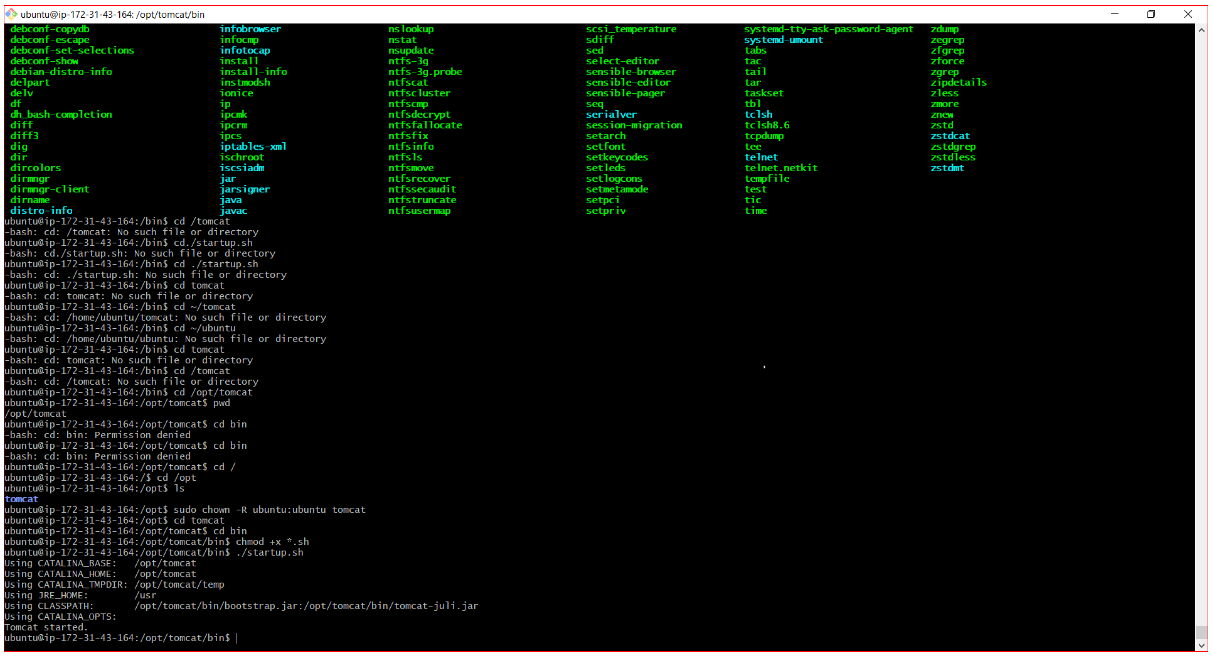
### 3.1. Screenshots and Output



```
      + tags_all                             = {
          + "Name" = "NGO-DevOps-Server"
        }
      + tenancy                              = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + primary_network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)

      + secondary_network_interface (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  ~ public_ip = "13.200.250.52" -> (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.ngo_server: Creating...
aws_instance.ngo_server: Still creating... [00m10s elapsed]
aws_instance.ngo_server: Creation complete after 1[illegible]s [id=i-0c[illegible]aa6[illegible]]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

public_ip = "13.234.19.214"
```

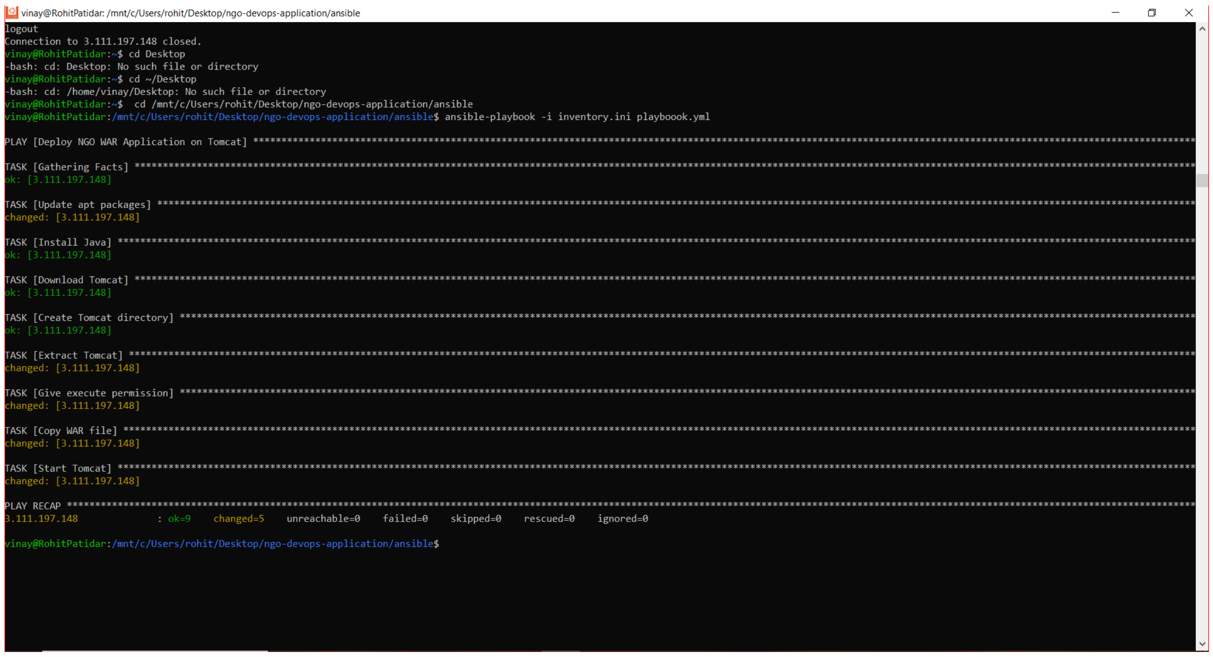


```
rohit@RohitPatidar MINGW64 ~/Desktop/ngo-devops-application/ansible (main)
$ cd ~/Downloads

rohit@RohitPatidar MINGW64 ~/Downloads (master)
$ cd NGO

rohit@RohitPatidar MINGW64 ~/Downloads/NGO (main)
$ cd NGO

rohit@RohitPatidar MINGW64 ~/Downloads/NGO/NGO (main)
$ ./mvnw clean package -DskipTests
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------------< com.ngo:ngoapp >--------------------------
[INFO] Building ngoapp 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- clean:3.3.2:clean (default-clean) @ ngoapp ---
[INFO] Deleting C:\Users\rohit\Downloads\NGO\NGO\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ngoapp ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ ngoapp ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 10 source files with javac [debug release 17] to target\classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ngoapp ---
[INFO] skip non existing resourceDirectory C:\Users\rohit\Downloads\NGO\NGO\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ ngoapp ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 1 source file with javac [debug release 17] to target\test-classes
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ ngoapp ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- war:3.3.2:war (default-war) @ ngoapp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ngoapp] in [C:\Users\rohit\Downloads\NGO\NGO\target\ngoapp-0.0.1-SNAPSHOT]
[INFO] Processing war project
[INFO] Building war: C:\Users\rohit\Downloads\NGO\NGO\target\ngoapp-0.0.1-SNAPSHOT.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  10.433 s
[INFO] Finished at: 2026-02-24T09:14:29+05:30
[INFO] ------------------------------------------------------------------------

rohit@RohitPatidar MINGW64 ~/Downloads/NGO/NGO (main)
$ |
```



ubuntu@ip-172-31-43-164: /opt/tomcat/bin

```
 debconf-copydb              infobrowser          nslookup             scsi_temperature     systemd-tty-ask-password-agent  zdump
 debconf-escape              infocmp              nstat                sdiff                systemd-umount                  zegrep
 debconf-set-selections      infotocap            nsupdate             sed                  tabs                            zfgrep
 debconf-show                install              ntfs-3g              select-editor        tac                             zforce
 debian-distro-info          install-info         ntfs-3g.probe        sensible-browser     tail                            zgrep
 delpart                     instmodsh            ntfscat              sensible-editor      tar                             zipdetails
 delv                        ionice               ntfscluster          sensible-pager       taskset                         zless
 df                          ip                   ntfscmp              seq                  tbl                             zmore
 dh_bash-completion          ipcmk                ntfsdecrypt          serialver            tclsh                           znew
 diff                        ipcrm                ntfsfallocate        session-migration    tclsh8.6                        zstd
 diff3                       ipcs                 ntfsfix              setarch              tcpdump                         zstdcat
 dig                         iptables-xml         ntfsinfo             setfont              tee                             zstdgrep
 dir                         ischroot             ntfsls               setkeycodes          telnet                          zstdless
 dircolors                   iscsiadm             ntfsmove             setleds              telnet.netkit                   zstdmt
 dirmngr                     jar                  ntfsrecover          setlogcons           tempfile
 dirmngr-client              jarsigner            ntfssecaudit         setmetamode          test
 dirname                     java                 ntfstruncate         setpci               tic
 distro-info                 javac                ntfsusermap          setpriv              time
ubuntu@ip-172-31-43-164:/bin$ cd /tomcat
-bash: cd: /tomcat: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd./startup.sh
-bash: cd./startup.sh: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd ./startup.sh
-bash: cd: ./startup.sh: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd tomcat
-bash: cd: tomcat: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd ~/tomcat
-bash: cd: /home/ubuntu/tomcat: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd ~/ubuntu
-bash: cd: /home/ubuntu/ubuntu: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd tomcat
-bash: cd: tomcat: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd /tomcat
-bash: cd: /tomcat: No such file or directory
ubuntu@ip-172-31-43-164:/bin$ cd /opt/tomcat
ubuntu@ip-172-31-43-164:/opt/tomcat$ pwd
/opt/tomcat
ubuntu@ip-172-31-43-164:/opt/tomcat$ cd bin
-bash: cd: bin: Permission denied
ubuntu@ip-172-31-43-164:/opt/tomcat$ cd bin
-bash: cd: bin: Permission denied
ubuntu@ip-172-31-43-164:/opt/tomcat$ cd /
ubuntu@ip-172-31-43-164:/$ cd /opt
ubuntu@ip-172-31-43-164:/opt$ ls
tomcat
ubuntu@ip-172-31-43-164:/opt$ sudo chown -R ubuntu:ubuntu tomcat
ubuntu@ip-172-31-43-164:/opt$ cd tomcat
ubuntu@ip-172-31-43-164:/opt/tomcat$ cd bin
ubuntu@ip-172-31-43-164:/opt/tomcat/bin$ chmod +x *.sh
ubuntu@ip-172-31-43-164:/opt/tomcat/bin$ ./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
ubuntu@ip-172-31-43-164:/opt/tomcat/bin$ |
```



vinay@RohitPatidar: /mnt/c/Users/rohit/Desktop/ngo-devops-application/ansible

```
logout
Connection to 3.111.197.148 closed.
vinay@RohitPatidar:~$ cd Desktop
-bash: cd: Desktop: No such file or directory
vinay@RohitPatidar:~$ cd ~/Desktop
-bash: cd: /home/vinay/Desktop: No such file or directory
vinay@RohitPatidar:~$  cd /mnt/c/Users/rohit/Desktop/ngo-devops-application/ansible
vinay@RohitPatidar:/mnt/c/Users/rohit/Desktop/ngo-devops-application/ansible$ ansible-playbook -i inventory.ini playboook.yml

PLAY [Deploy NGO WAR Application on Tomcat] *************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [3.111.197.148]

TASK [Update apt packages] ******************************************************************************************
changed: [3.111.197.148]

TASK [Install Java] *************************************************************************************************
ok: [3.111.197.148]

TASK [Download Tomcat] **********************************************************************************************
ok: [3.111.197.148]

TASK [Create Tomcat directory] **************************************************************************************
ok: [3.111.197.148]

TASK [Extract Tomcat] ***********************************************************************************************
changed: [3.111.197.148]

TASK [Give execute permission] **************************************************************************************
changed: [3.111.197.148]

TASK [Copy WAR file] ************************************************************************************************
changed: [3.111.197.148]

TASK [Start Tomcat] *************************************************************************************************
changed: [3.111.197.148]

PLAY RECAP **********************************************************************************************************
3.111.197.148              : ok=9    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vinay@RohitPatidar:/mnt/c/Users/rohit/Desktop/ngo-devops-application/ansible$
```

## 3.2. Key outcomes

- Successfully implemented Infrastructure as Code (IaC) using Terraform to provision AWS EC2 infrastructure automatically.
- Automated server configuration and application deployment using Ansible playbooks.
- Built and packaged a Spring Boot Java application into a WAR file using Maven.
- Automated installation and configuration of required dependencies such as:

➔ Java (JDK 17)
➔ Apache Tomcat
➔ MySQL Server

- Configured MySQL database and integrated it with the deployed application.
- Eliminated manual server setup, reducing human errors and increasing deployment consistency.
- Achieved repeatable and scalable deployment architecture.
- Deployed the application successfully on a public cloud instance accessible via public IP.
- Integrated application and infrastructure layers into a structured DevOps workflow.
- Demonstrated end-to-end automation from infrastructure provisioning to application deployment.

## 4. Conclusion

This project successfully demonstrated the implementation of Infrastructure as Code (IaC) principles to automate the provisioning and deployment of a Non-Profit Management System. By using Terraform for infrastructure provisioning and Ansible for configuration management, the entire environment setup was automated, ensuring repeatability, scalability, and reliability.

The Java-based Spring Boot application was built using Maven and deployed on Apache Tomcat hosted on an AWS EC2 instance. Database configuration and connectivity were established with MySQL, completing the full-stack deployment process.

The project highlights the importance of DevOps practices in reducing manual intervention, minimizing configuration errors, and improving deployment efficiency. It also demonstrates how automation tools can streamline cloud-based application deployment in real-world scenarios. Overall, the project validates the effectiveness of Infrastructure as Code in building scalable, maintainable, and production-ready systems.