

## Experiment 7: Configuration Management with Ansible:

Basics of Ansible: Inventory, playbooks, and Modules, Automating server Configurations with playbooks, hands-on: Writing and Running a Basic Playbook.

Install Ansible on Ubuntu

Open a terminal and follow these steps:

step 1: Update package lists

```
sudo apt update
```

step 2: Install software-properties-common (optional but helpful)

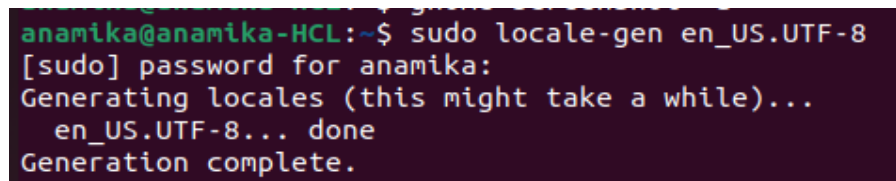
```
sudo apt install software-properties-common -y
```

step 3: Add the Ansible PPA

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
```

step 4: Fix the locale

```
sudo locale-gen en_US.UTF-8
```



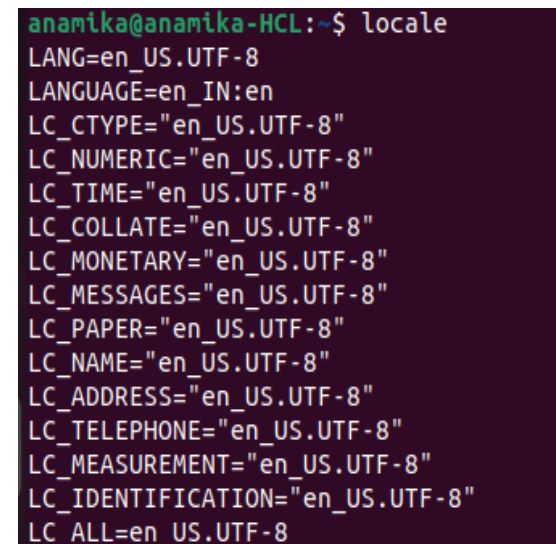
```
anamika@anamika-HCL:~$ sudo locale-gen en_US.UTF-8
[sudo] password for anamika:
Generating locales (this might take a while)...
  en_US.UTF-8... done
Generation complete.
```

```
sudo update-locale LANG=en_US.UTF-8
```

```
export LANG=en_US.UTF-8
```

```
export LC_ALL=en_US.UTF-8
```

```
locale
```



```
anamika@anamika-HCL:~$ locale
LANG=en_US.UTF-8
LANGUAGE=en_IN:en
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=en_US.UTF-8
```

step 5: Install Ansible

```
sudo apt install ansible -y
```

step 6: Verify installation

ansible --version

```
anamika@anamika-HCL:~$ ansible --version
ansible [core 2.17.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/anamika/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/anamika/.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
```

step 7: Create inventory file

vi inventory.ini

code:

```
[local]
```

```
localhost ansible_connection=local
```

save and exit.

step 8: create yml playbook

vi create-file.yml

code: (note:- indentation code)

```
---
```

```
- name: Create a file on remote hosts
```

```
  hosts: all
```

```
  become: true
```

```
  tasks:
```

```
    - name: Create a file called /tmp/hello.txt
```

```
      file:
```

```
        path: /tmp/hello.txt
```

```
        state: touch
```

save and exit.

Run Command:

ansible-playbook -i inventory.ini create-file.yml

```

anamika@anamika-HCL:~$ ansible-playbook -i inventory.ini create-file.yml

PLAY [Create a file on remote hosts] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost 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: [localhost]

TASK [Create a file called /tmp/hello.txt] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

step 8: To verify the result manually

ls -l /tmp/hello.txt

(or)

cat /tmp/hello.txt

```

anamika@anamika-HCL:~$ ls -l /tmp/hello.txt
-rw-r--r-- 1 root root 0 Apr 21 12:33 /tmp/hello.txt

```

**Experiment 8: Practical Exercise:**

Set up a Jenkins CI Pipeline for a Maven Project, Use Ansible to deploy Artifacts generated by Jenkins

- i) Jenkins setup (pipeline for Maven build)
- ii) Ansible setup for deployment
- iii) Integration between Jenkins and Ansible

Dashboard > mvn-deploy-pipeline > Configuration

Dashboard > mvn-deploy-pipeline > Configuration

In

## Configure

⚙️ General

🕒 Triggers

📄 Pipeline

🔑 Advanced

☐ Throttle builds ?

### Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Trigger builds remotely (e.g., from scripts) ?

☐ GitHub project

☐ Pipeline speed/durability override ?

the

trigger

section leave uncheck

Dashboard > mvn-deploy-pipeline > Configuration

Configure

General

Triggers

Pipeline

Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

1 pipeline {

2 agent any

3

4 tools {

5 maven 'Maven 3'

6 jdk 'Java 11'

7 }

8

9 stages {

10 stage('Checkout Code') {

11 steps {

12 git 'https://github.com/YOUR-USERNAME/YOUR-MAVEN-REPO.git'

13 }

14 }

15 }

try sample Pipeline...

☒ Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Save

Apply

Save the configuration.

Pipeline script: Jenkinsfile  
code:

```
pipeline {
  agent any

  tools {
    maven 'Maven 3'
    jdk 'Java 11'
  }

  stages {
    stage('Checkout Code') {
      steps {
        git 'https://github.com/YOUR-USERNAME/YOUR-MAVEN-REPO.git'
      }
    }

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

```

    }

    stage('Archive JAR') {
        steps {
            archiveArtifacts artifacts: 'target/*.jar'
        }
    }

    stage('Deploy using Ansible') {
        steps {
            sh 'ansible-playbook -i ansible/inventory.ini ansible/deploy.yml'
        }
    }
}
}
}

```

Steps to do in terminal :

1. Move to clean location:

```
cd ~
```

```
mkdir java-projects
```

```
cd java-projects
```

2.Run the Maven Archetype Command Again:

```
mvn archetype:generate \
```

```
-DgroupId=com.example \
```

```
-DartifactId=myapp \
```

```
-DarchetypeArtifactId=maven-archetype-quickstart \
```

```
-DinteractiveMode=false
```

```

java-projects/
├─ myapp/
│   ├─ pom.xml
│   └─ src/

```

```

anamika@anamika-HCL:~/java-projects$ ls
myapp
anamika@anamika-HCL:~/java-projects$ cd myapp
anamika@anamika-HCL:~/java-projects/myapp$ ls
pom.xml  src
anamika@anamika-HCL:~$ cd ~
mkdir java-projects
cd java-projects
anamika@anamika-HCL:~/java-projects$ mvn archetype:generate \
-DgroupId=com.example \
-DartifactId=myapp \
-DarchetypeArtifactId=maven-archetype-quickstart \
-DinteractiveMode=false
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] >>> maven-archetype-plugin:3.3.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]

```

### 3.Create Jenkinsfile:

touch Jenkinsfile

nano Jenkinsfile

code:

```
pipeline {
  agent any

  tools {
    maven 'Maven 3'      // Make sure this name matches what you added in Jenkins
  }
  jdk 'Java 11'
}

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

  stage('Archive Artifact') {
    steps {
      archiveArtifacts artifacts: 'target/*.jar'
    }
  }

  stage('Deploy with Ansible') {
    steps {
      sh 'ansible-playbook -i ansible/inventory.ini ansible/deploy.yml'
    }
  }
}
```

### 4.Create Ansible Directory and files:

mkdir ansible

cd ansible

### 5.Create inventory.ini:

nano inventory.ini

code:

[app]

localhost ansible\_connection=local



```
anamika@anamika-HCL:~/java-projects/myapp/ansible$ nano inventory.ini
anamika@anamika-HCL:~/java-projects/myapp/ansible$ cat inventory.ini
[app]
localhost ansible_connection=local
```

6. After running below command pom.xml should be like this:  
mvn clean package

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-jar/2.4.0/surefire-jar-2.4.0.jar (26 kB at 162 kB/s)
-----
T E S T S
-----
Running com.example.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.044 sec
Results :
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ myapp ---
[INFO] Building jar: /home/anamika/java-projects/myapp/target/myapp-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.244 s
[INFO] Finished at: 2025-04-22T11:40:05+05:30
[INFO] -----
```

note: (pom.xml will auto create, just check properties is added or not. If not add properties)

code:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>myapp</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>myapp</name>
  <url>http://maven.apache.org</url>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
```

```

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
</dependencies>

</project>

```

verify:

ls target

myapp-1.0-SNAPSHOT.jar should exist

```

anamika@anamika-HCL:~/java-projects/myapp$ ls target
classes          generated-test-sources  maven-status          surefire-reports
generated-sources maven-archiver         myapp-1.0-SNAPSHOT.jar test-classes

```

7.Create deploy.yml:

cd myapp

(note: install jdk, if not installed

sudo apt install default-jdk )

jar xf target/myapp-1.0-SNAPSHOT.jar META-INF/MANIFEST.MF

cat META-INF/MANIFEST.MF

verify Main-Class as above screenshot it should be com.example.App

```

anamika@anamika-HCL:~/java-projects/myapp$ jar xf target/myapp-1.0-SNAPSHOT.jar META-INF/MANIFEST.MF
anamika@anamika-HCL:~/java-projects/myapp$ cat META-INF/MANIFEST.MF
Manifest-Version: 1.0
Created-By: Maven Jar Plugin 3.2.0
Build-Jdk-Spec: 17
Main-Class: com.example.App

```

cp target/myapp-1.0-SNAPSHOT.jar /home/anamika/deploy/myapp.jar

java -jar /home/anamika/deploy/myapp.jar

```

anamika@anamika-HCL:~/java-projects/myapp$ cp target/myapp-1.0-SNAPSHOT.jar /home/anamika/deploy/myapp.jar
anamika@anamika-HCL:~/java-projects/myapp$ java -jar /home/anamika/deploy/myapp.jar
Hello World!

```

cd ansible

nano deploy.yml

code:

- name: Deploy Maven JAR

hosts: app

tasks:

- name: Ensure target directory exists

file:

path: /opt/myapp

state: directory

mode: '0755'

```
- name: Copy JAR file
  copy:
    src: ../target/myapp-1.0-SNAPSHOT.jar
    dest: /opt/myapp/myapp.jar
```

Save and exit

```
-----
nano hosts.ini
code:
[app]
localhost ansible_connection=local
```

save and exit

```
anamika@anamika-HCL:~/java-projects/myapp/ansible$ ls
deploy.yml  hosts.ini  inventory.ini
anamika@anamika-HCL:~/java-projects/myapp/ansible$ cat hosts.ini
[app]
localhost ansible_connection=local
```

Run

Ansible Playbook with below command:  
ansible-playbook hosts.ini deploy.yml

```
anamika@anamika-HCL:~/java-projects/myapp/ansible$ ansible-playbook -i hosts.ini deploy.yml

PLAY [Deploy Maven JAR] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost 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: [localhost]

TASK [Create deploy directory if it doesn't exist] *****
ok: [localhost]

TASK [Copy JAR to deployment folder] *****
ok: [localhost]

TASK [Run Java Application] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

anamika@anamika-HCL:~/java-projects/myapp/ansible$ ls
deploy.yml  hosts.ini  inventory.ini
anamika@anamika-HCL:~/java-projects/myapp/ansible$ cat hosts.ini
[app]
localhost ansible_connection=local
```