

Experiment No. 05

AIM: To Build the pipeline of jobs using Maven /Gradle /Ant in Jenkins,create a pipeline script to Test and deploy an application over the tomcat server.

THEORY:

Jenkins is an open-source automation server used to implement Continuous Integration (CI) and Continuous Deployment (CD) in software development. It helps automate tasks like building, testing, and deploying code, improving efficiency and reducing human errors. **Key**

Concepts:

1. **Automated Builds:** Jenkins automatically compiles code whenever changes are made, ensuring that the software is always in a buildable state.
2. **Continuous Integration:** Developers frequently commit code to a shared repository. Jenkins automatically triggers builds and runs tests to detect issues early.
3. **Pipelines:** Jenkins workflows are defined using Pipelines in a Jenkinsfile, which can be scripted or declarative.
4. **Plugins:** Jenkins supports plugins for version control (Git), build tools (Maven, Gradle), deployment (Docker, Kubernetes), and more.
5. **Freestyle Projects vs. Pipelines:**
 - **Freestyle Projects:** Simple, GUI-based jobs for basic automation.
 - **Pipelines:** Code-defined workflows for complex CI/CD tasks.

Example of a Simple Jenkins Pipeline (Declarative):

```
pipeline {  
  
    agent any  
  
    stages {  
  
        stage('Build') {  
  
            steps {
```

```
    sh 'echo "Building the project"'

  } }

stage('Test') {

  steps {

    sh 'echo "Running tests"'

  }

}

stage('Deploy') {

  steps {

    sh 'echo "Deploying the application"'

  } } }
```

This pipeline defines three stages: **Build**, **Test**, and **Deploy**, each with its own commands.


Output:


Dashboard > All > New Item


New Item


Enter an item name

Select an item type

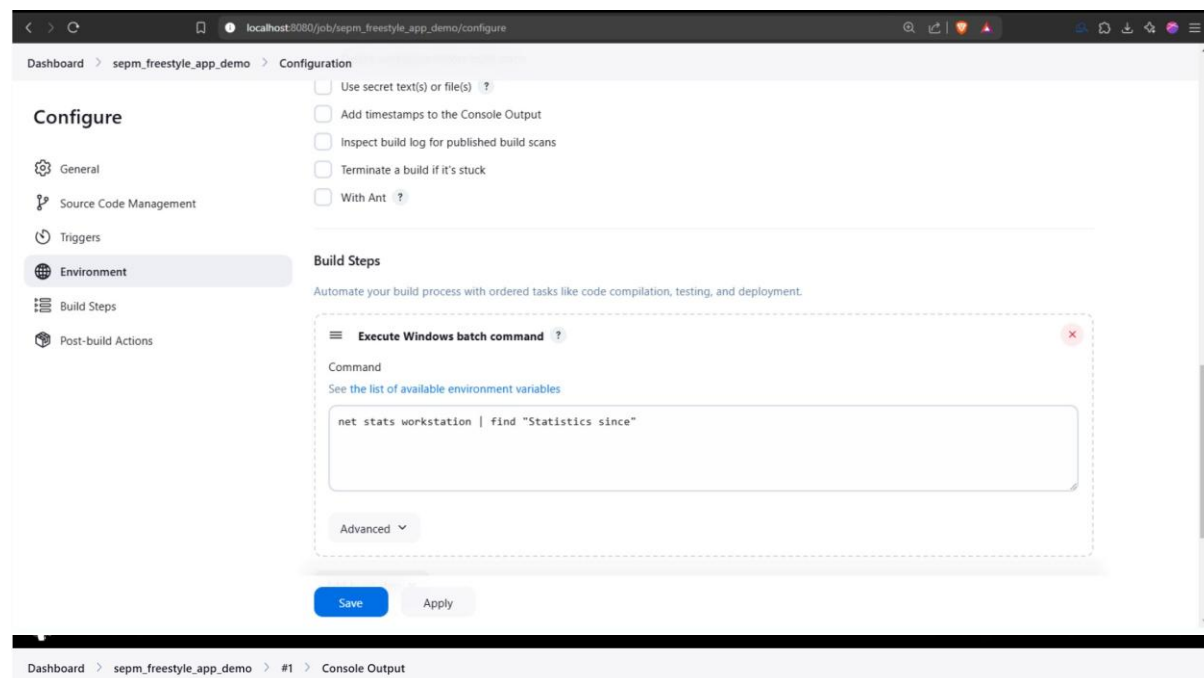
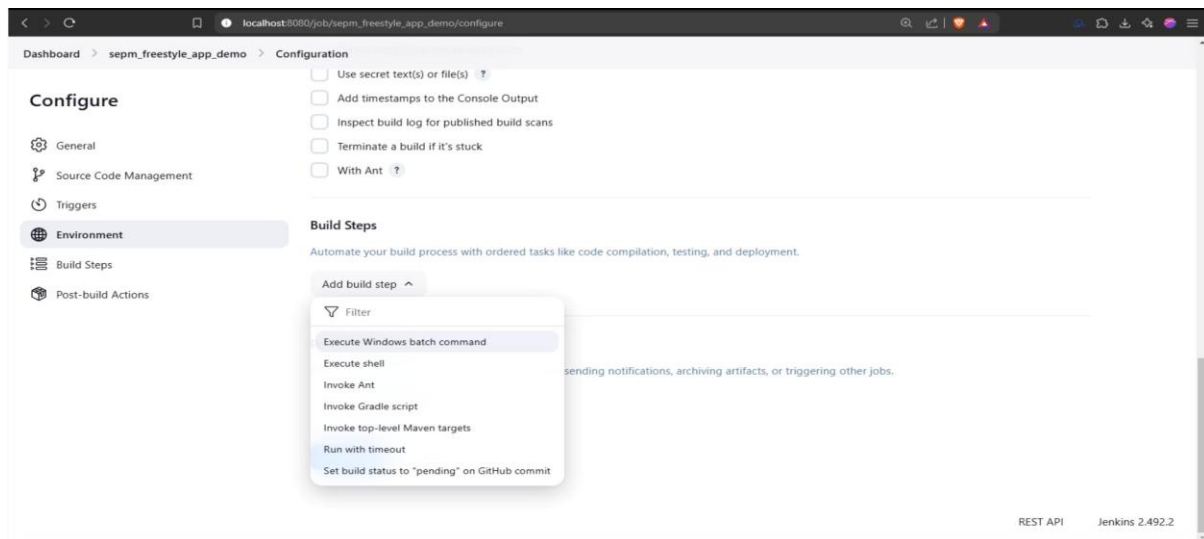
**Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

**Pipeline**
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.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

OK



MSYS: d/MiscRepos/sepm_lab

1 @echo off

2 echo "Hello %1... Your address is %2"

example1.cmd[+] [unix] (05:29 01/01/1970) 2,38 All
-- INSERT --

Dashboard > sepm_supply_params > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

With Ant

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute Windows batch command

Command

See the list of available environment variables

D:\MiscRepos\sepm_lab\files\example1.cmd Taha Mazgaon

Advanced

Add build step

Post-build Actions

Save

Apply

Dashboard > sepm_supply_params > #1

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

#1 (Apr 2, 2025, 12:01:38 PM)

Add description

Keep this build forever

Started 55 sec ago
Took 0.29 sec

This run spent:

6 ms waiting;

0.29 sec build duration;

0.29 sec total from scheduled to completion.

No changes.

```
vi JavaInJenkins.java
6 class JavaInJenkins
5 {
4     public static void main(String[] args)
3     {
2         System.out.println("Hello, from the java file");
1     }
7 }
```

```
/d/MiscRepos/sepm_lab/files git:(master)±5 (5.232s)
vi JavaInJenkins.java
```

```
/d/MiscRepos/sepm_lab/files git:(master)±5 (1.298s)
javac JavaInJenkins.java
```

```
/d/MiscRepos/sepm_lab/files git:(master)±5 (1.296s)
java JavaInJenkins.java
Hello, from the java file
```

```
/d/MiscRepos/sepm_lab/files git:(master) v ±5
```

Dashboard > All > New Item

New Item

Enter an item name

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

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.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

OK

Dashboard > sepm_java_in_jenkins > Configuration

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute Windows batch command ?

Command

See the list of available environment variables

javac D:\MiscRepos\sepm_lab\files\JavaInJenkins.java
java D:\MiscRepos\sepm_lab\files\JavaInJenkins.java

Advanced

Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action

Save

Apply

Dashboard > All > New Item

New Item

Enter an item name

sepm_parameterized_build

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline

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.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder

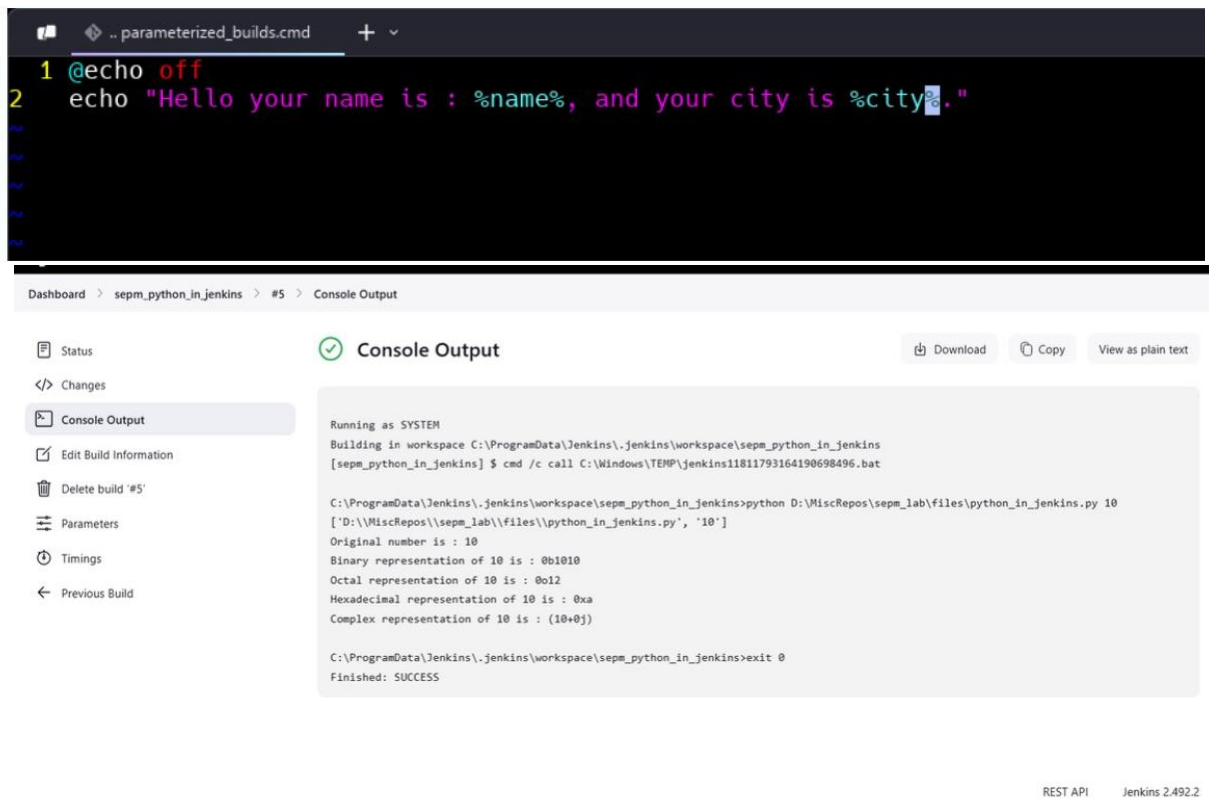
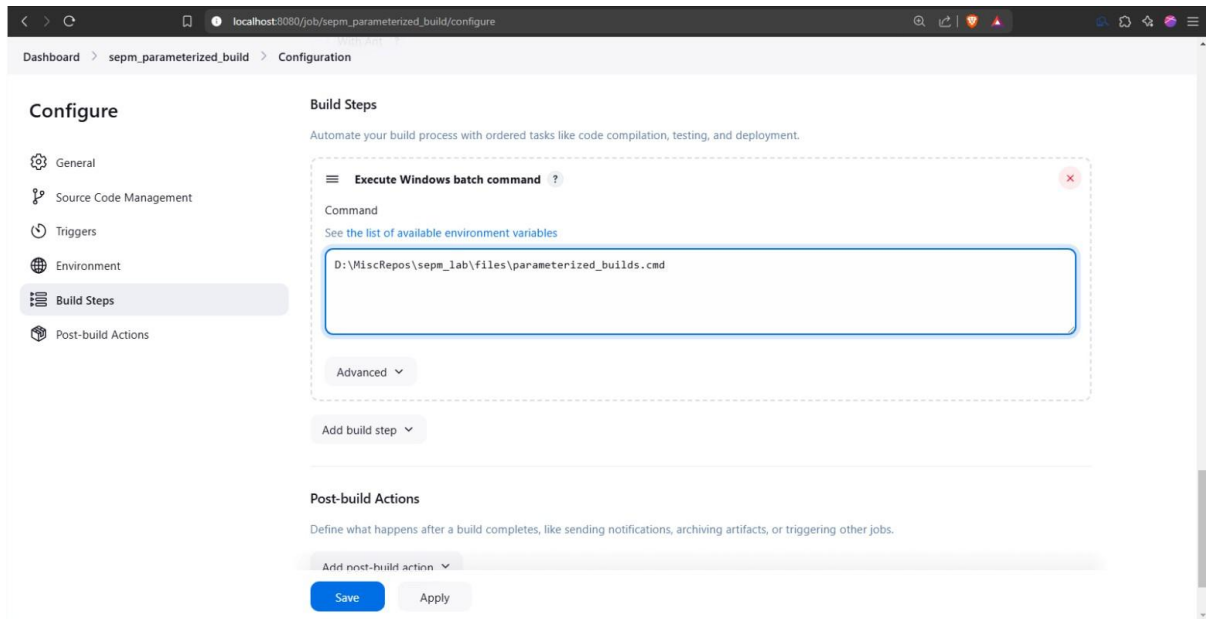
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder is a container whose contents are visible in the left-hand navigation pane of the console page. It lists all things inside it directly.

OK

The image displays two screenshots of the Jenkins configuration interface for a parameterized build.

Top Screenshot: The 'Configure' page for 'sepm_parameterized_build' is shown. The 'General' tab is active. The 'A demo for parameterized builds' text area is visible. The 'Plain text' option is selected. The 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is unchecked. The 'This project is parameterized' checkbox is checked. The 'Add Parameter' dropdown menu is open, showing options: Filter, Boolean Parameter, Choice Parameter, Credentials Parameter, File Parameter, Multi-line String Parameter, Password Parameter, Run Parameter, and String Parameter. The 'Choice Parameter' option is highlighted.

Bottom Screenshot: The same 'Configure' page is shown, but the 'Choice Parameter' option is now selected in the dropdown menu. The 'Throttle builds' checkbox is unchecked. The 'Execute concurrent builds if necessary' checkbox is unchecked. The 'Advanced' dropdown menu is visible. The 'Source Code Management' section is partially visible at the bottom.



CONCLUSION: Hence, we have successfully created a VPC Service for launching the Instances.