

## EXPERIMENT NO.3

**AIM :-** a)To practice/execute shell programs using Jenkins.

b)To practice/execute parameterised Java programs using Jenkins.

### THEORY :-

Jenkins is an open-source server that is written entirely in Java. It lets you execute a series of actions to achieve the continuous integration process, and it does so in an automated fashion. This CI server runs in servlet containers such as Apache Tomcat. Jenkins facilitates continuous integration and continuous delivery in software projects by automating tasks related to building, testing, and deployment.

This makes it easy for developers to continuously improve the product by integrating changes into the project regularly. Jenkins automates software builds in a continuous manner and notifies developers of errors at an early stage.

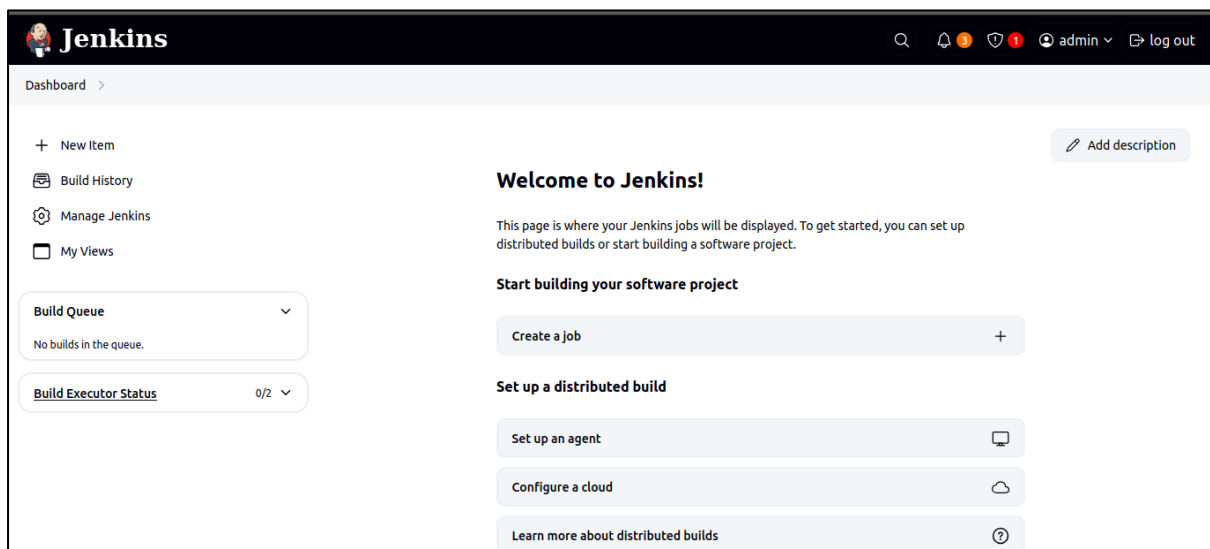
A strong Jenkins community is one of the key reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.

Build systems supported by Jenkins include tools such as Gradle, Maven, and more.

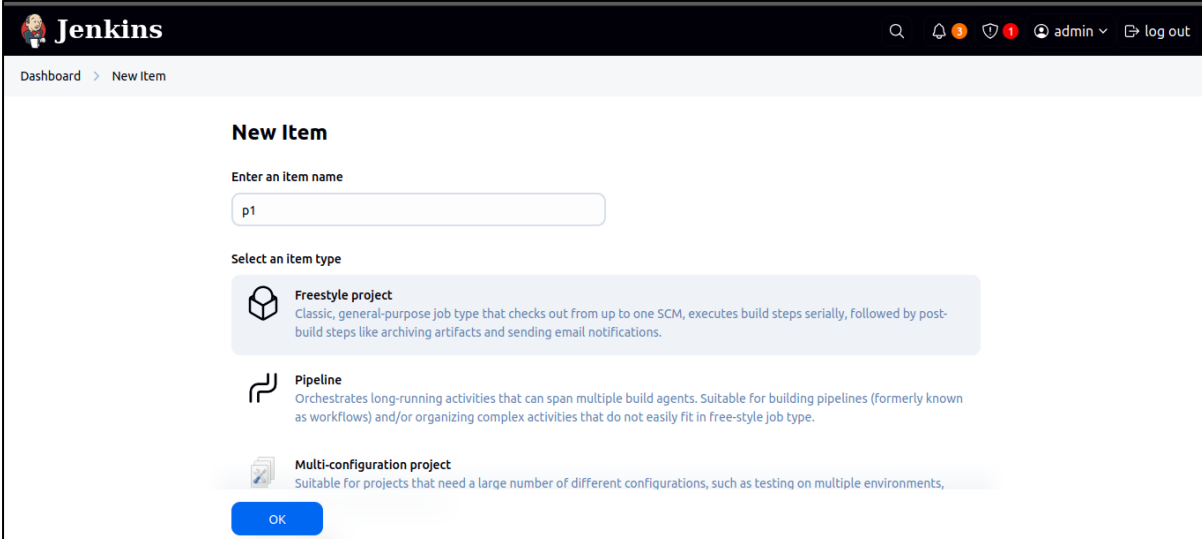
Automation testing can be done using test frameworks such as Nose2, PyTest, Robot, Selenium, and others.

- TO PRACTISE/EXECUTE SHELL PROGRAMS USING JENKINS

**Step 1 :** Click on Create new jobs

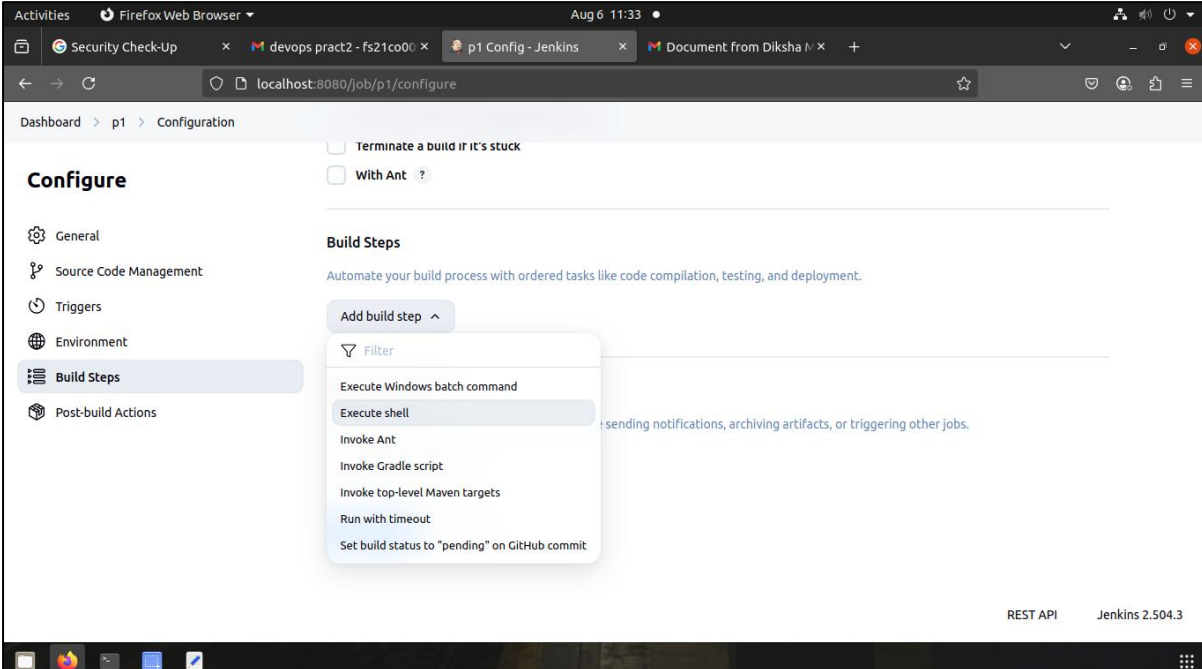


**Step 2 :** Give a name to project as “P1”, select Option “Free style project” and click on OK button



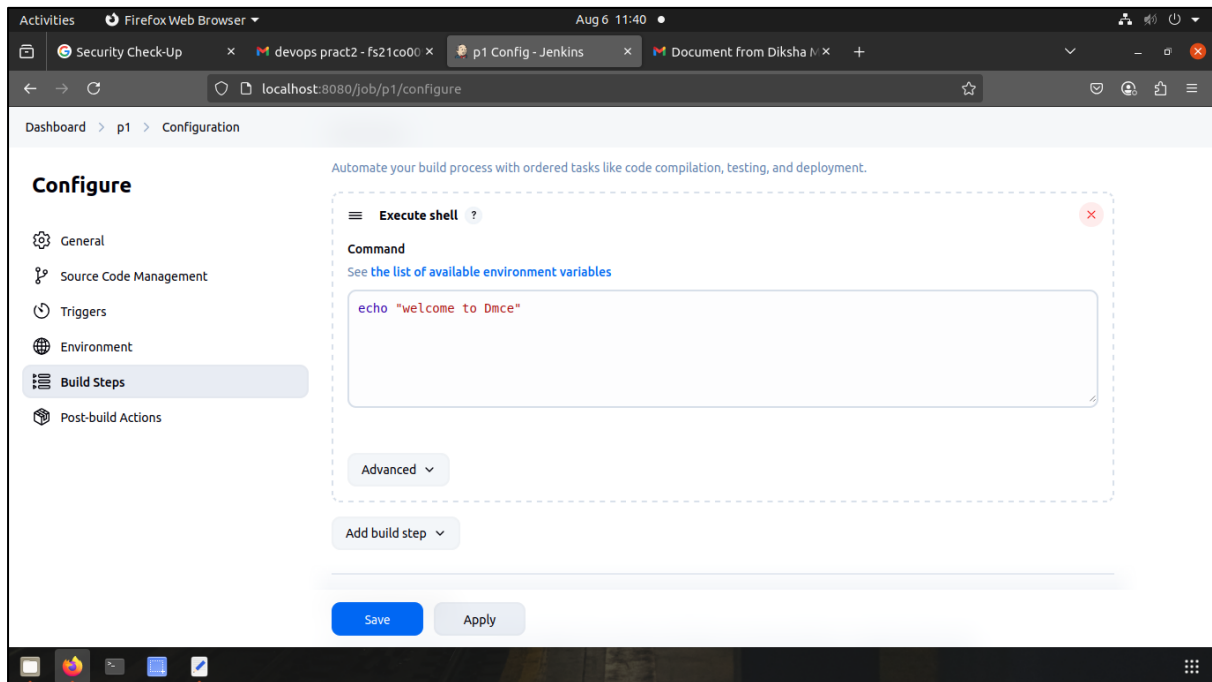
The screenshot shows the Jenkins 'New Item' configuration page. At the top, the Jenkins logo and navigation links are visible. The main heading is 'New Item'. Below it, there is a text input field labeled 'Enter an item name' with the value 'p1'. Underneath, the 'Select an item type' section offers three options: 'Freestyle project' (highlighted), 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom, there is a blue 'OK' button.

**Step 3 :** To run simple shell scripts on Jenkins click on Build option select the Execute script from dropdown menu

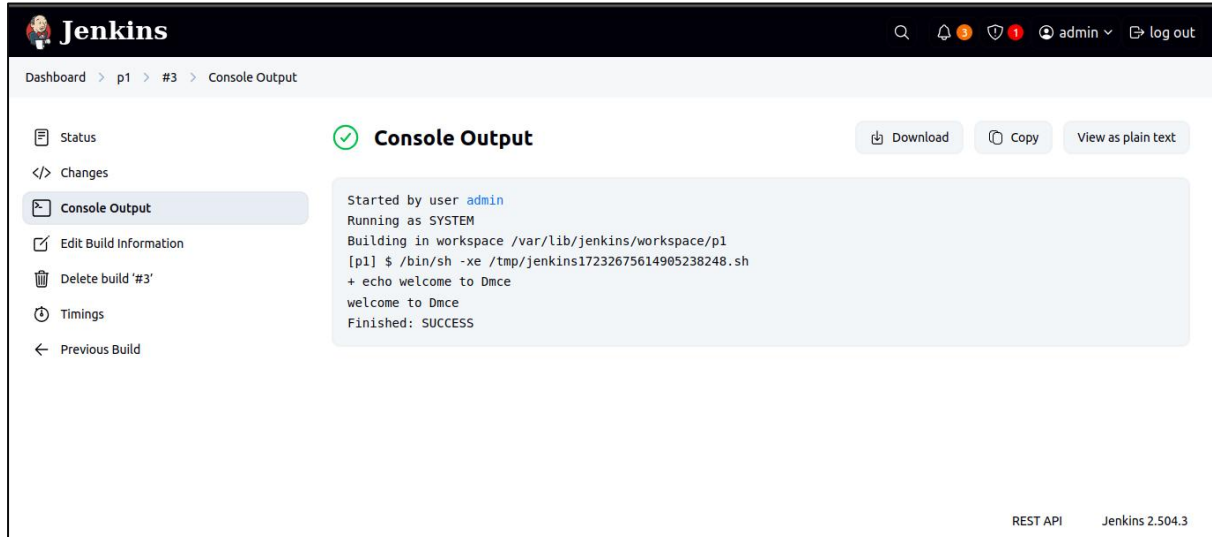


The screenshot shows the Jenkins 'Configure' page for a job named 'p1'. The left sidebar contains a menu with options: General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions. The main area is titled 'Configure' and shows the 'Build Steps' section. A dropdown menu is open under 'Add build step', listing various options: 'Execute Windows batch command', 'Execute shell' (highlighted), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. The bottom of the page shows 'REST API' and 'Jenkins 2.504.3'.

**Step 4 :** Write a simple shell command and click on apply followed by save button

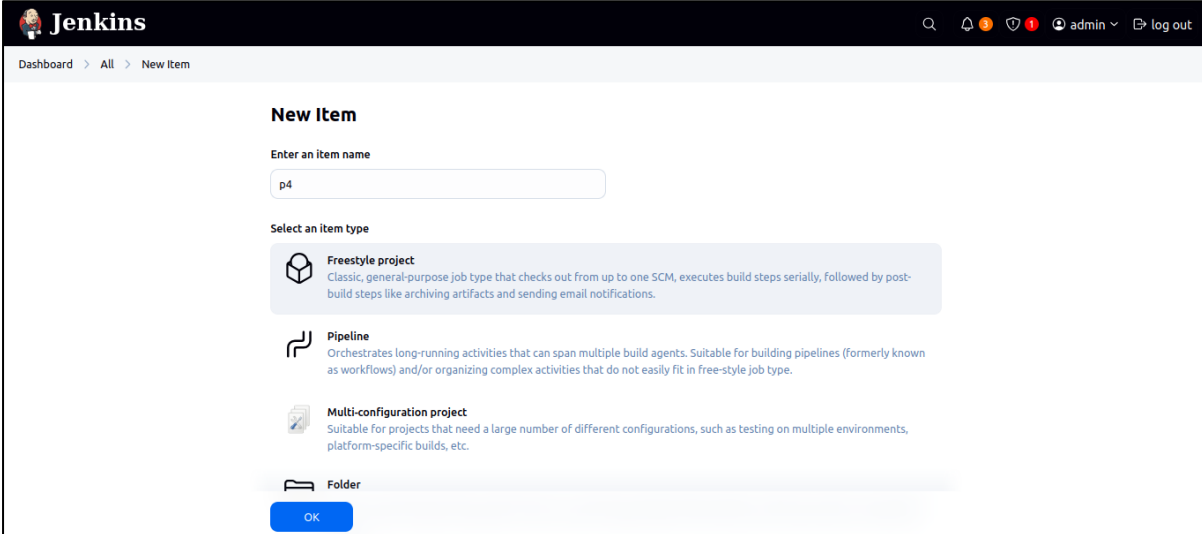


**Step 5 :** Click on first build “1” followed by console output to see the output



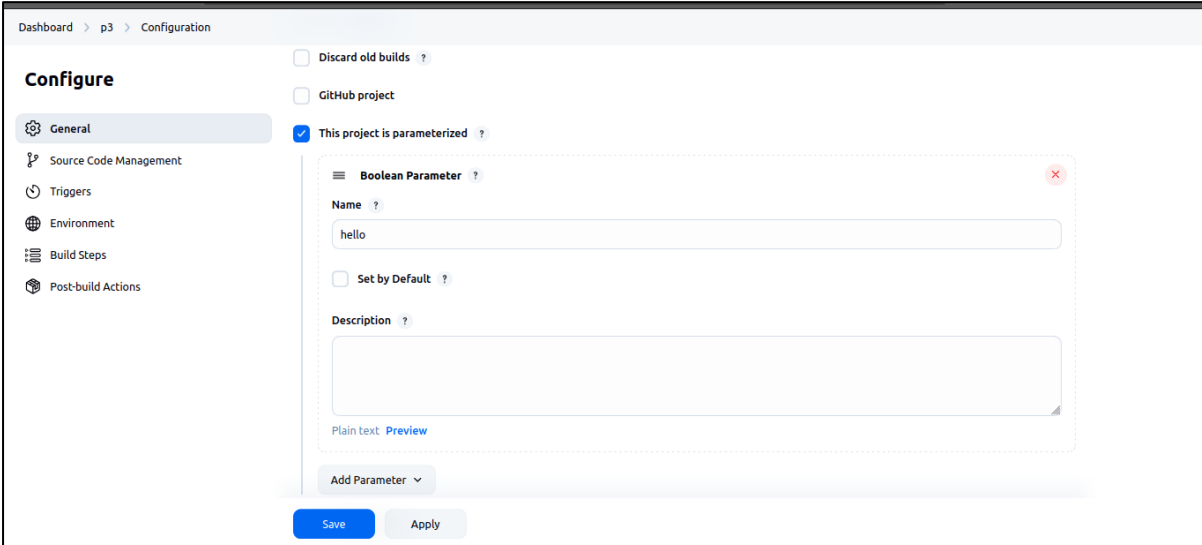
- PRACTISE/EXECUTE PARAMETERISED JAVA PROGRAMS USING JENKINS

**Step 1 :** Create a freestyle project P4 in Jenkins

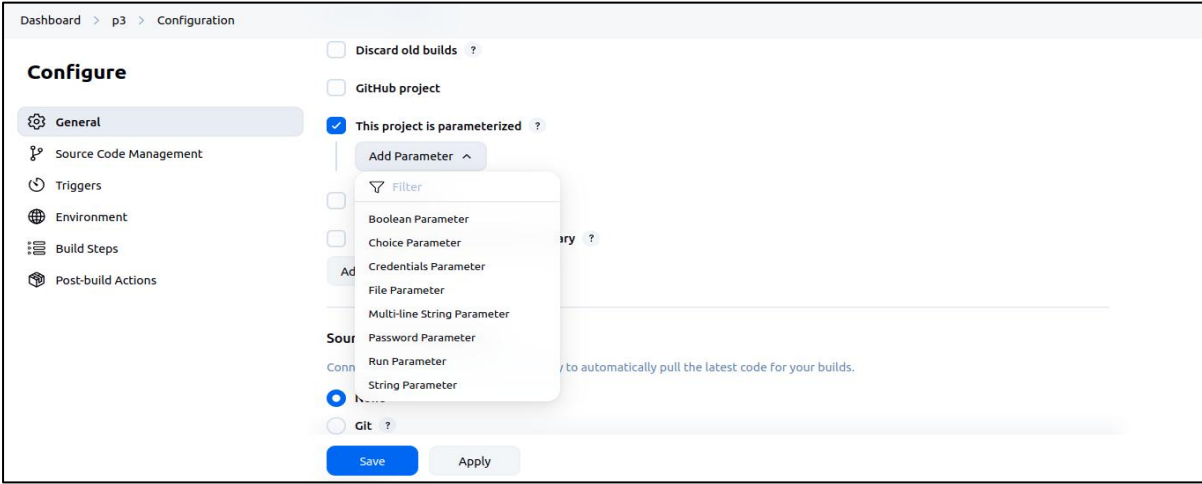


The screenshot shows the Jenkins 'New Item' page. At the top, the Jenkins logo and navigation links are visible. The breadcrumb trail is 'Dashboard > All > New Item'. The main heading is 'New Item'. Below it, there is a text input field labeled 'Enter an item name' with the value 'p4'. Underneath, the 'Select an item type' section displays four options: 'Freestyle project' (selected), 'Pipeline', 'Multi-configuration project', and 'Folder'. Each option has a brief description. At the bottom, there is an 'OK' button.

**Step2 :** Click on the General tab, check "This project is parameterized", and add a Boolean Parameter with the name "hello"



The screenshot shows the Jenkins 'Configuration' page for project 'p3'. The breadcrumb trail is 'Dashboard > p3 > Configuration'. The left sidebar shows the 'Configure' section with tabs for 'General', 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. In the main area, the 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is unchecked. The 'This project is parameterized' checkbox is checked. Below this, a 'Boolean Parameter' is added with the name 'hello'. The 'Set by Default' checkbox is unchecked. The 'Description' field is empty. At the bottom, there is a 'Save' button and an 'Apply' button.



The screenshot shows the Jenkins 'Configuration' page for project 'p3' with the 'Add Parameter' dropdown menu open. The breadcrumb trail is 'Dashboard > p3 > Configuration'. The left sidebar shows the 'Configure' section with tabs for 'General', 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. In the main area, the 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is unchecked. The 'This project is parameterized' checkbox is checked. Below this, the 'Add Parameter' button is clicked, and a dropdown menu is shown with the following options: 'Boolean Parameter', 'Choice Parameter', 'Credentials Parameter', 'File Parameter', 'Multi-line String Parameter', 'Password Parameter', 'Run Parameter', 'String Parameter', and 'Text Parameter'. The 'Boolean Parameter' option is selected. At the bottom, there is a 'Save' button and an 'Apply' button.

Dashboard > p3 > Configuration

### Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

☐ Terminate a build if it's stuck

☐ With Ant ?

#### Build Steps

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

≡ **Execute shell** ?

Command

[See the list of available environment variables](#)

```
echo "hello"
```

Advanced ▾

Add build step ▾

**Save** **Apply**

**Jenkins** 🔍 🔔 3 🛡️ 1 👤 admin ▾ 🚪 log out

Dashboard > p3 > #2 > Console Output

📄 Status

</> Changes

**🖨️ Console Output**

✎ Edit Build Information

🗑 Delete build '#2'

⚙ Parameters

🕒 Timings

← Previous Build

✅ **Console Output**

[Download](#) [Copy](#) [View as plain text](#)

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/p3
[p3] $ /bin/sh -xe /tmp/jenkins5407022602546151923.sh
+ echo true
true
Finished: SUCCESS
```

REST API Jenkins 2.504.3

**Step 3:** Click on Add Parameter, select Password Parameter, and set the name (e.g., “dmce”) for secure input.

Dashboard > p3 > Configuration

### Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

☐ GitHub project

☒ This project is parameterized ?

≡ **Password Parameter** ?

Name ?

dmce

Default Value ?

Concealed **Change Password**

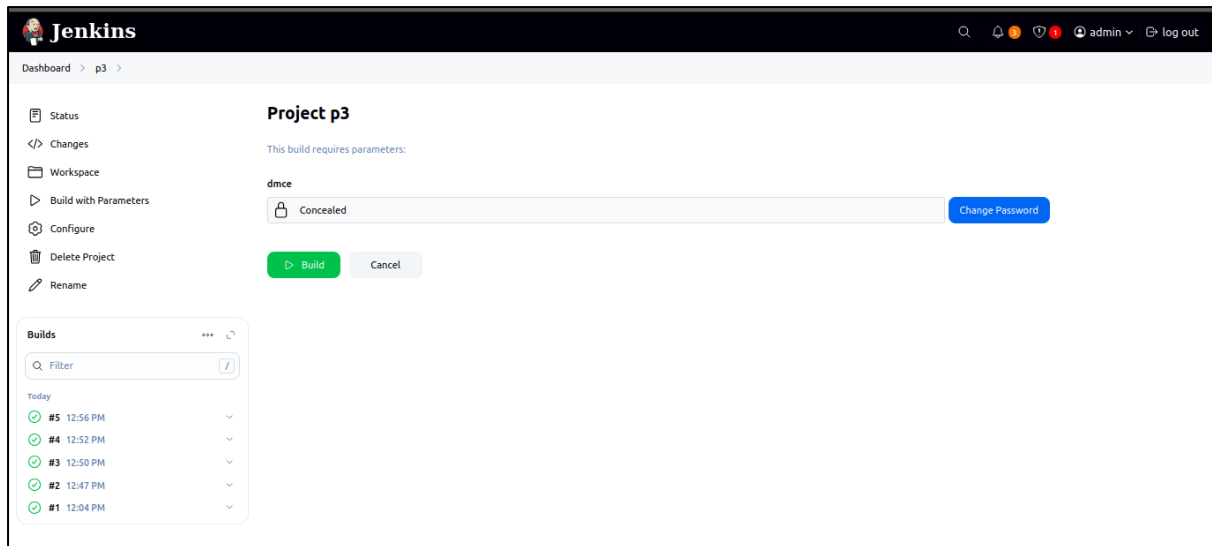
Description ?

Plain text [Preview](#)

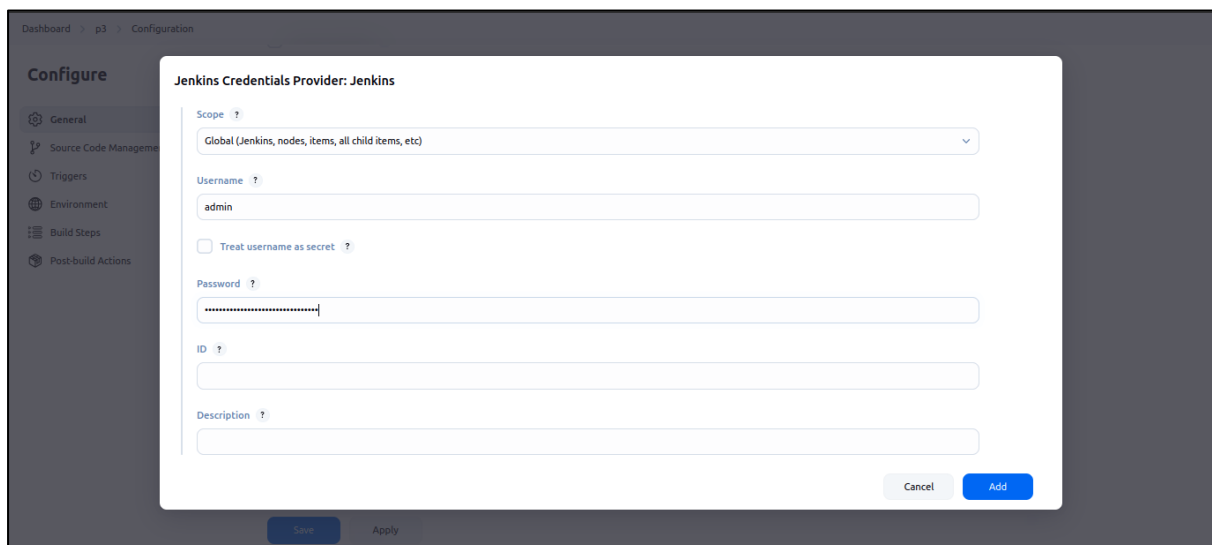
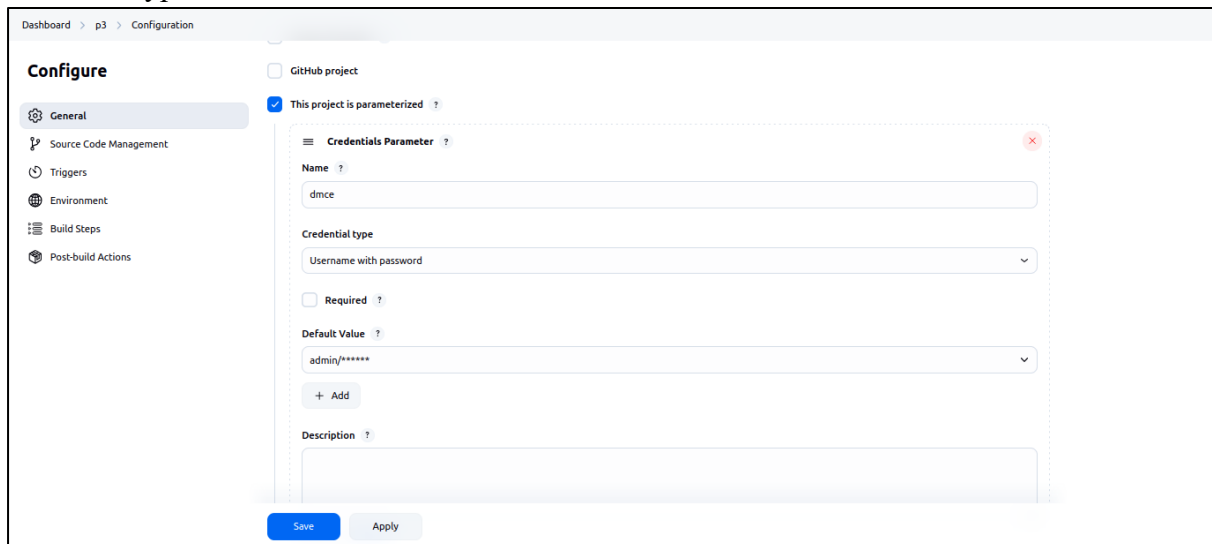
Add Parameter ▾

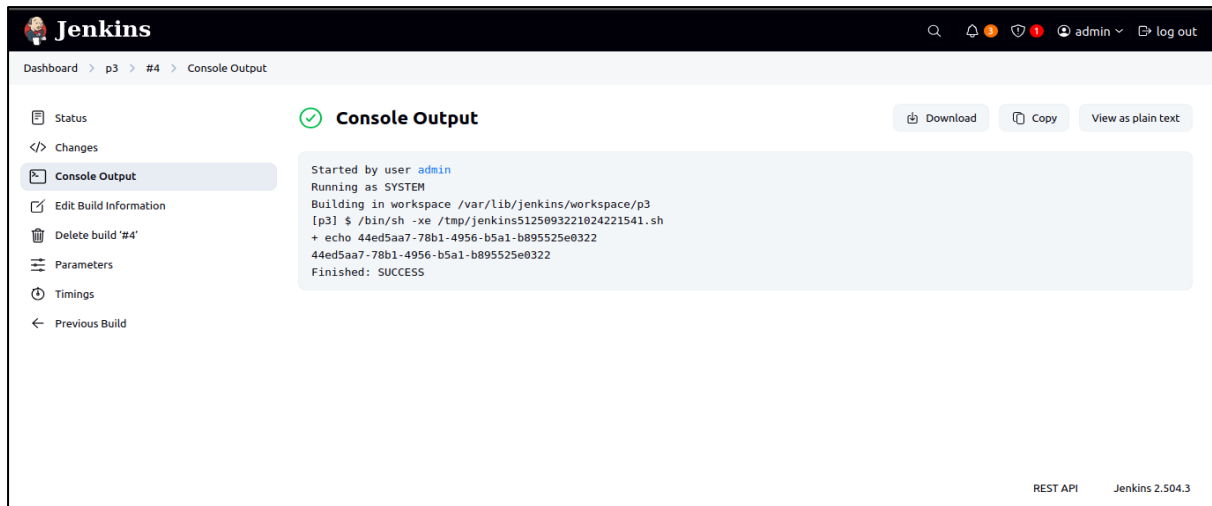
☐ Throttle builds ?

**Save** **Apply**

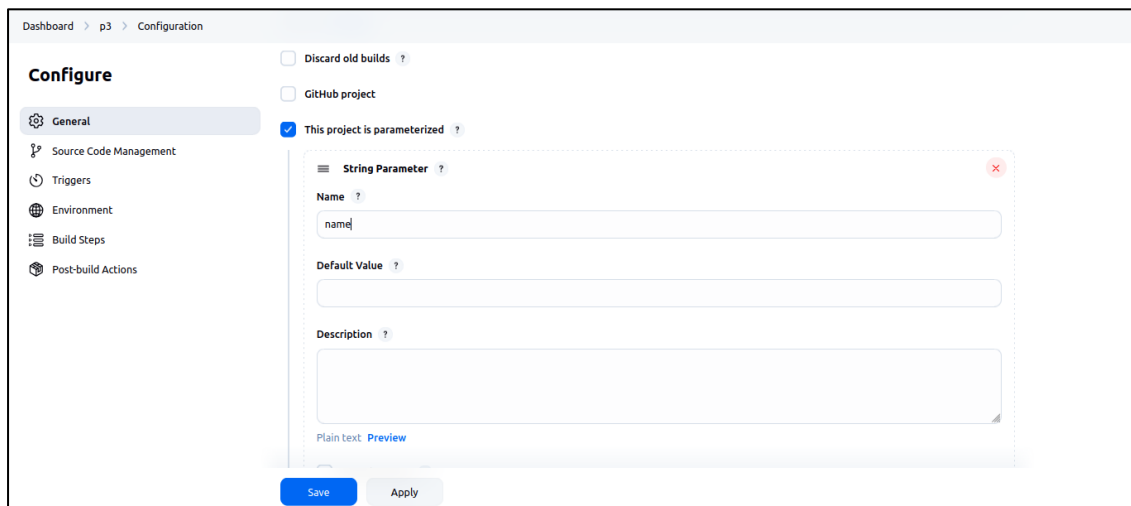


**Step 4:** Click on Add Parameter, select Credentials Parameter, and choose the appropriate credential type to use in the build.

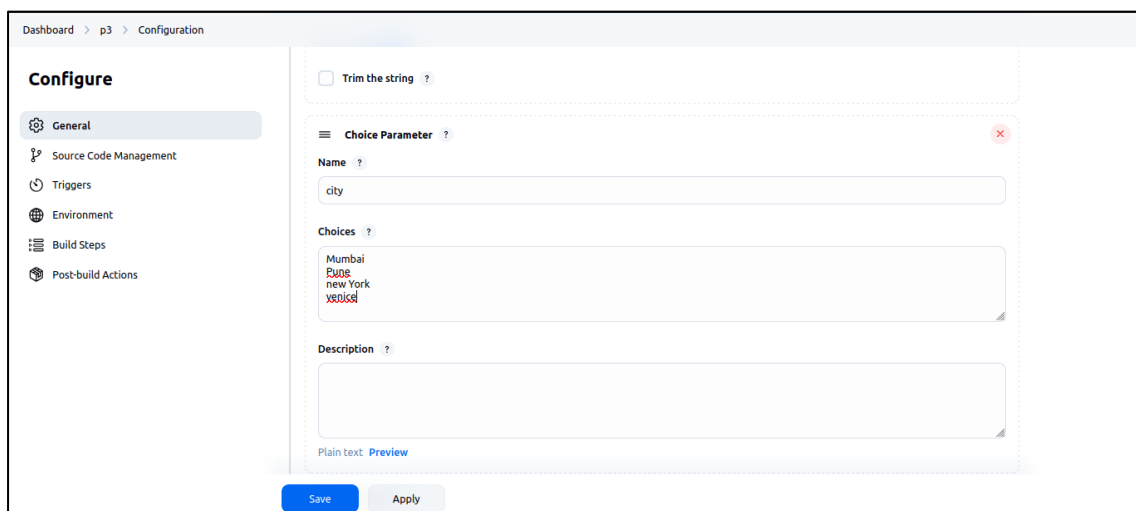




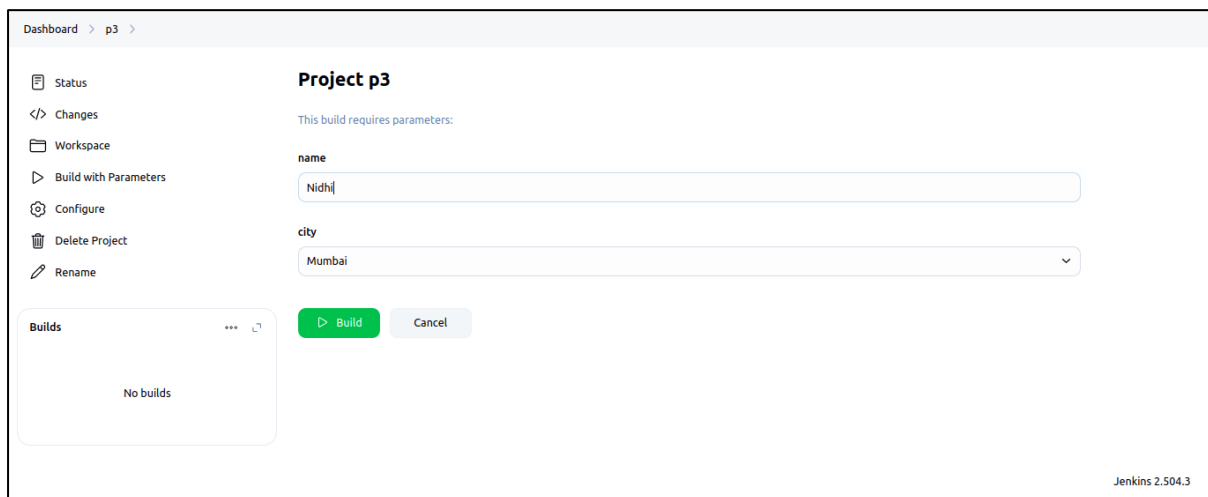
**Step 5:** Click on general menu and select option this project is parameterize. Select String parameter and specify name as “Name”



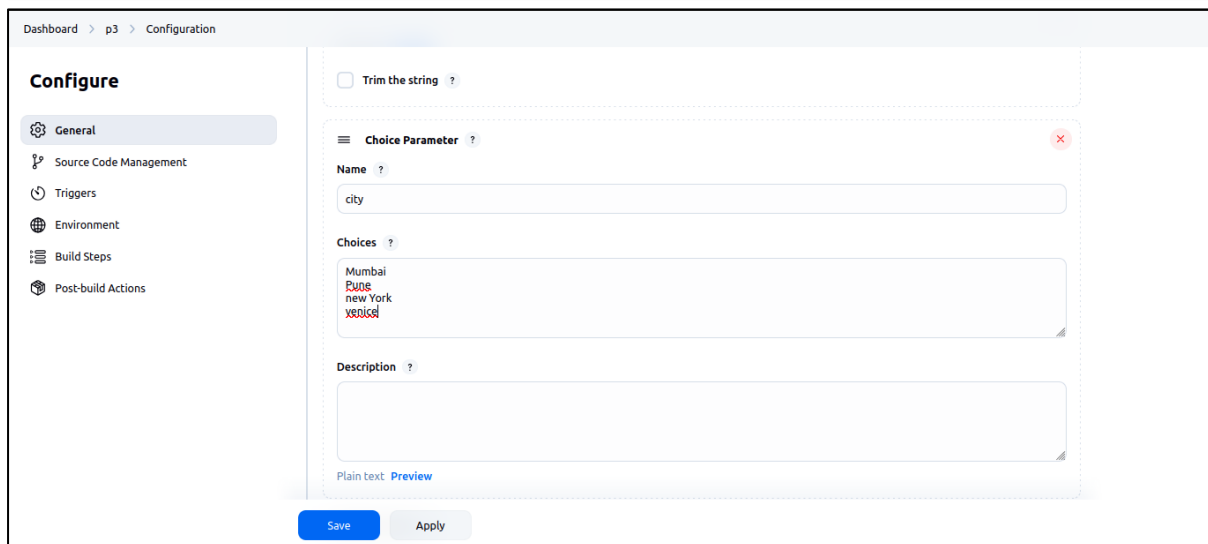
**Step 6 :** Click on add parameter and select choice parameter. Take second parameter as choice parameter



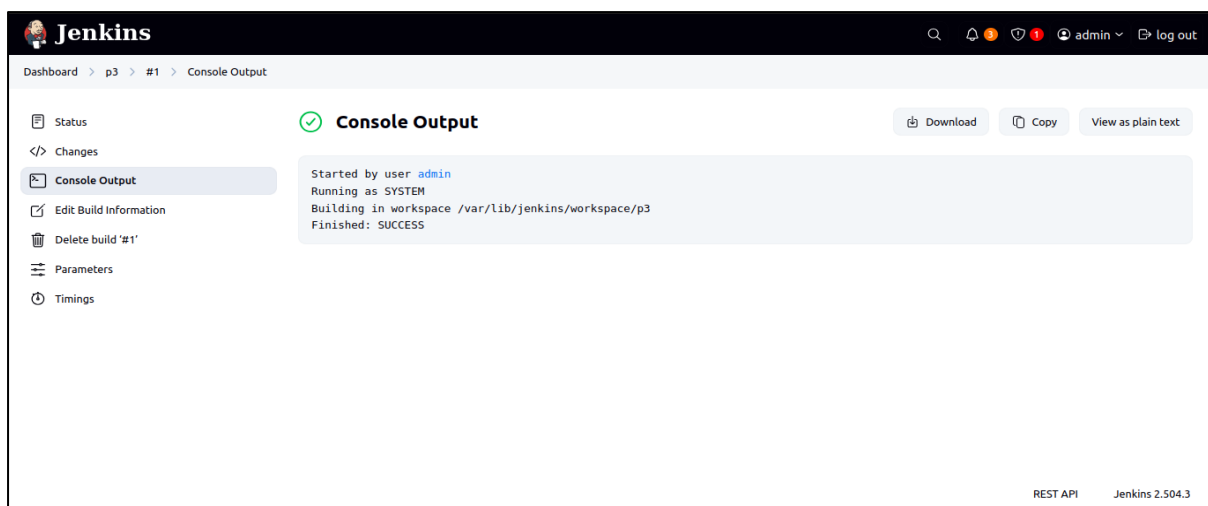
## Step 7 : Click on build with parameters and specify the values



The screenshot shows the Jenkins 'Build with Parameters' interface for 'Project p3'. The left sidebar contains navigation links: Status, Changes, Workspace, Build with Parameters (selected), Configure, Delete Project, and Rename. The main area has a 'name' text field with 'Nidhi' and a 'city' dropdown menu with 'Mumbai' selected. Below these fields are 'Build' and 'Cancel' buttons. A 'Builds' section on the left shows 'No builds'. The bottom right corner indicates 'Jenkins 2.504.3'.



The screenshot shows the Jenkins 'Configuration' page for 'Project p3'. The left sidebar has links: General (selected), Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The main area is titled 'Configure' and contains a 'Choice Parameter' section. The 'Name' field is 'city'. The 'Choices' field lists 'Mumbai', 'Pune', 'new York', and 'XROCK'. The 'Description' field is empty. There are 'Save' and 'Apply' buttons at the bottom. The bottom right corner indicates 'Jenkins 2.504.3'.



The screenshot shows the Jenkins 'Console Output' page for 'Project p3' build #1. The left sidebar has links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#1', Parameters, and Timings. The main area shows the console output with a green checkmark icon and the text: 'Started by user admin', 'Running as SYSTEM', 'Building in workspace /var/lib/jenkins/workspace/p3', and 'Finished: SUCCESS'. There are 'Download', 'Copy', and 'View as plain text' buttons. The bottom right corner indicates 'REST API' and 'Jenkins 2.504.3'.

**CONCLUSION :-** Hence we can conclude that we have learned and implemented shell programs and parametrized Java programs using Jenkins.