

EXPERIMENT NO : 03

AIM : To practice/execute shell programs and parameterized 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, that too 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 parts related to build, test, and deployment. This makes it easy for developers to continuously work on the betterment of the product by integrating changes to the project.

Jenkins automates the software builds in a continuous manner and lets the developers know about the errors at an early stage. A strong Jenkins community is one of the prime reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.

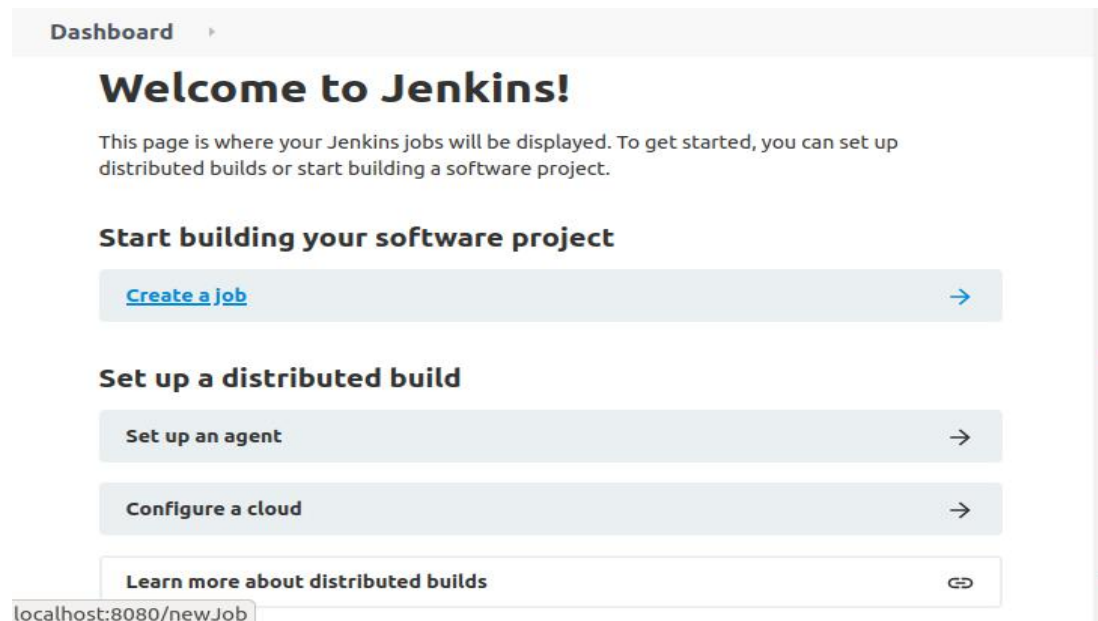
Some of the possible steps that can be performed using Jenkins are:

- Software build using build systems such as Gradle, Maven, and more.

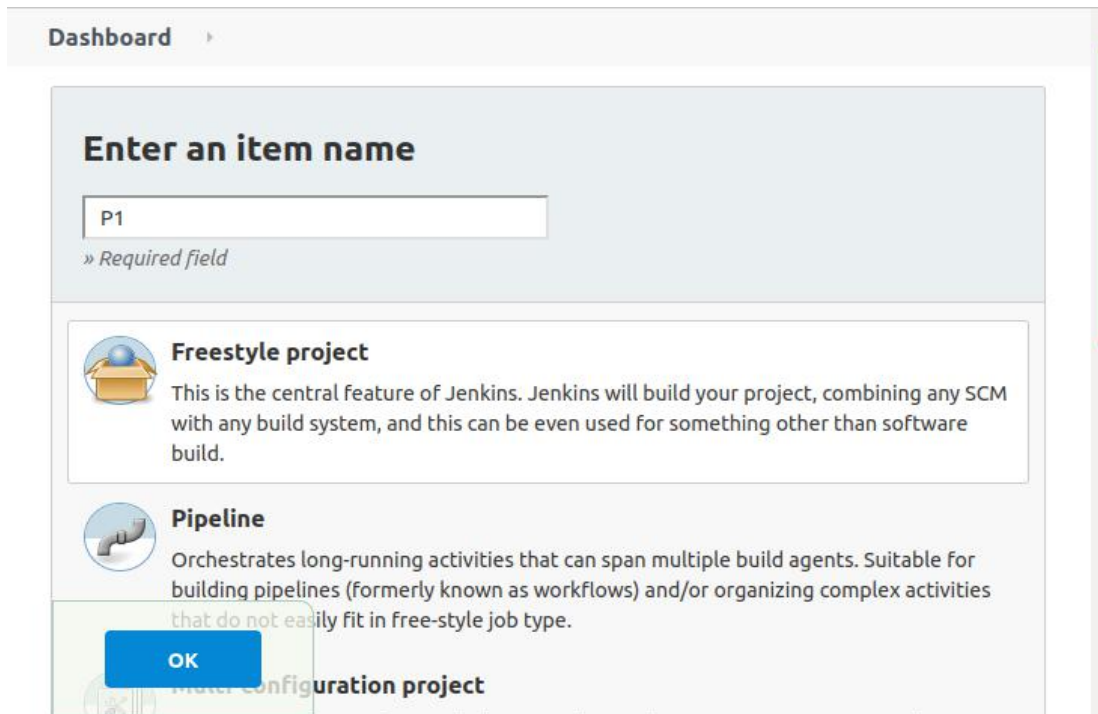
- Automation testing using test frameworks such as Nose2, PyTest, Robot, Selenium, and more.

TO PRACTISE/EXECUTE SHELL PROGRAMS USING JENKINS

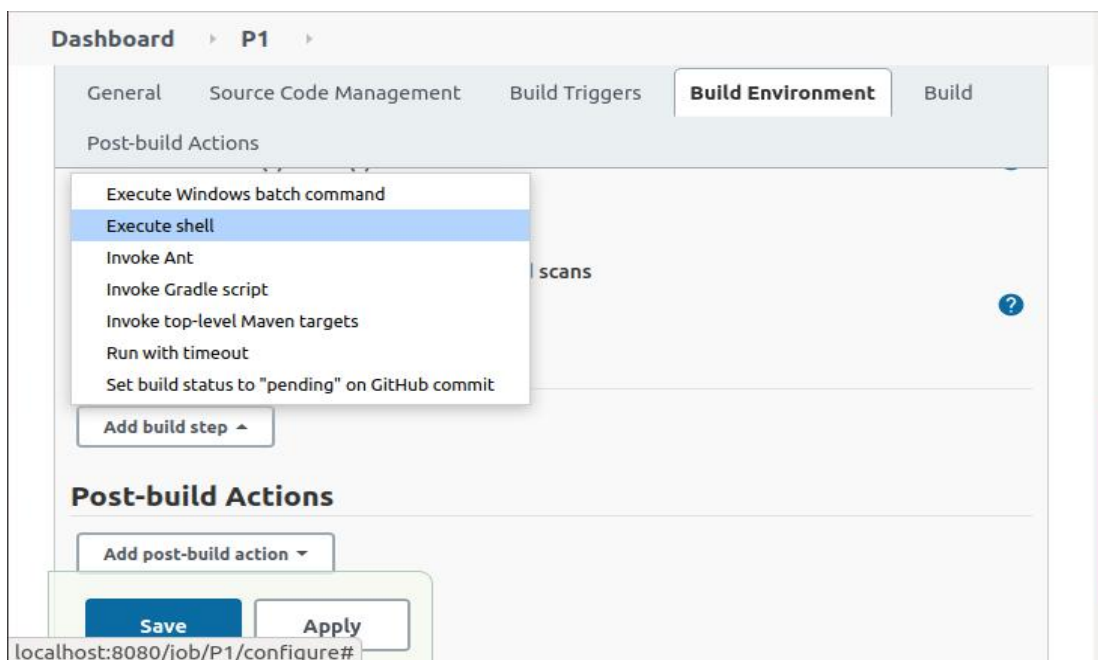
Step 1 : Click on Create new jobs



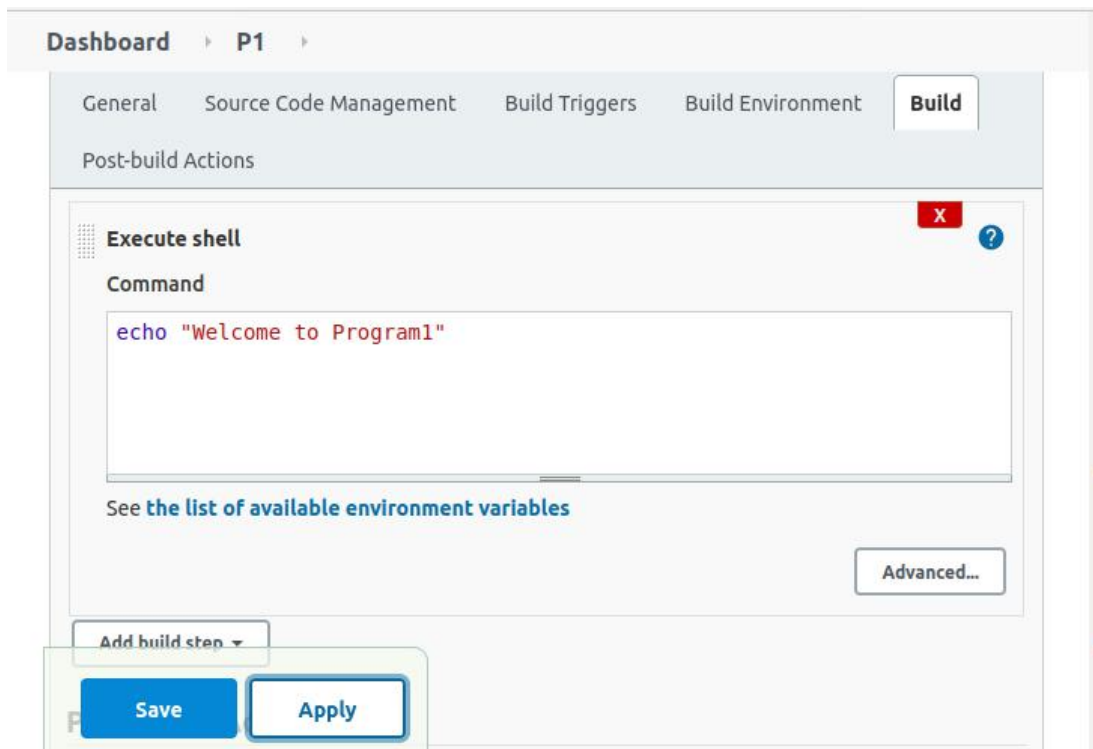
Step 2 : Give a name to project as “P1”, select Option “Free style project” and click on OK button



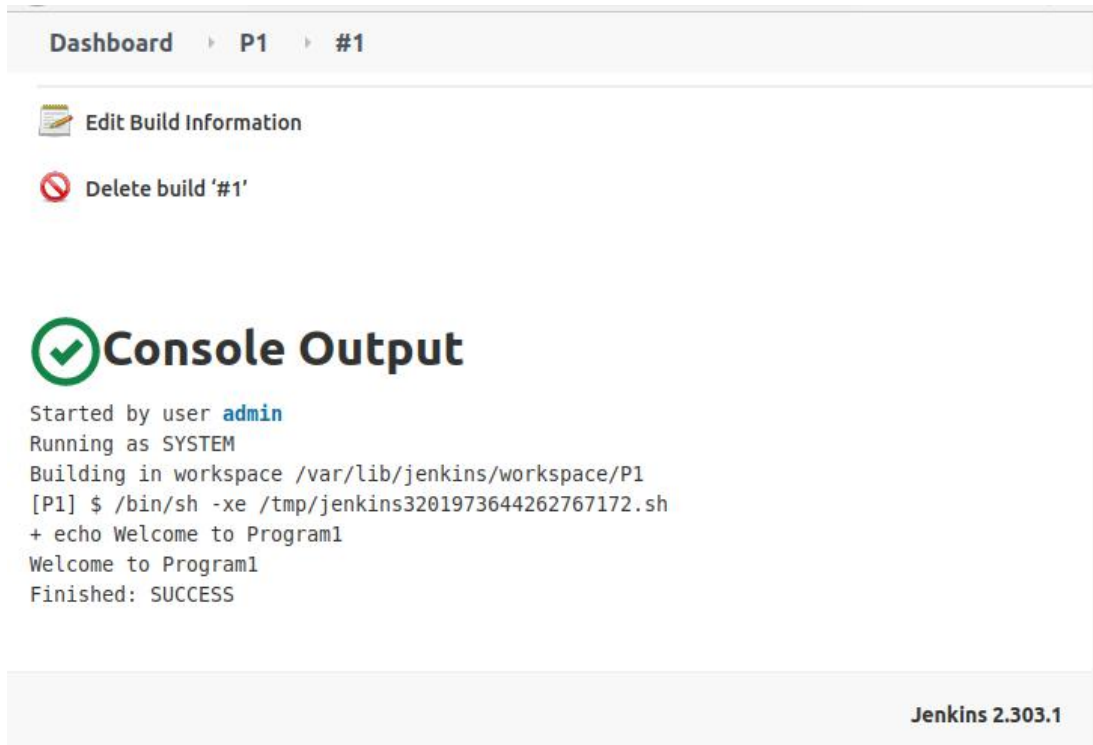
Step 3 : To run simple shell scripts on Jenkins click on Build option select the Execute script from dropdown menu



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



TO PRACTISE/EXECUTE PARAMETERISED JAVA PROGRAMS USING JENKINS


Step 1 : Create a freestyle project P2 in Jenkins


Dashboard > All >


Enter an item name

P2

» Required field

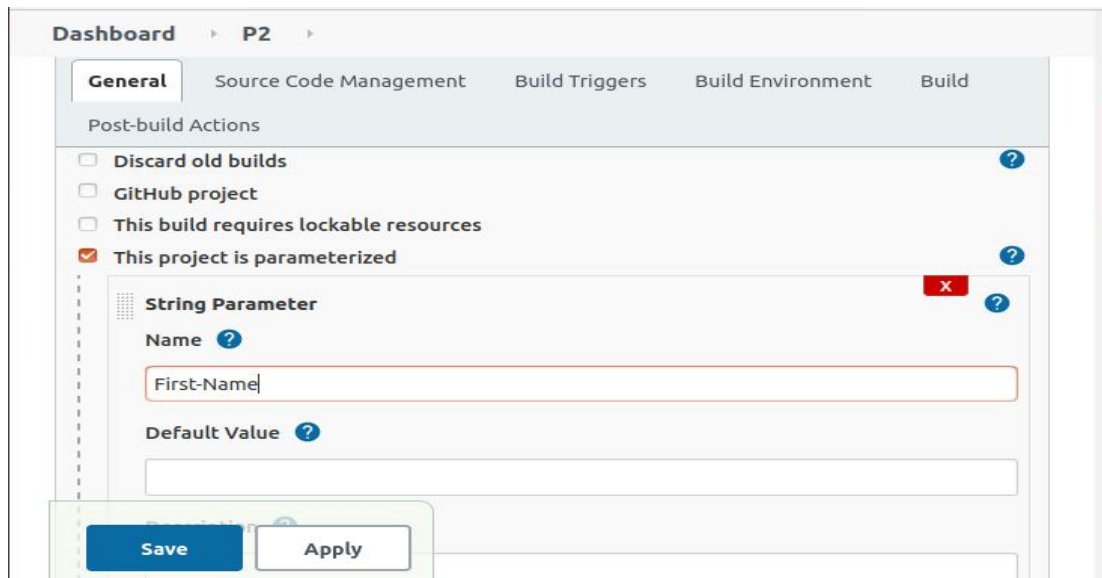
 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **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.

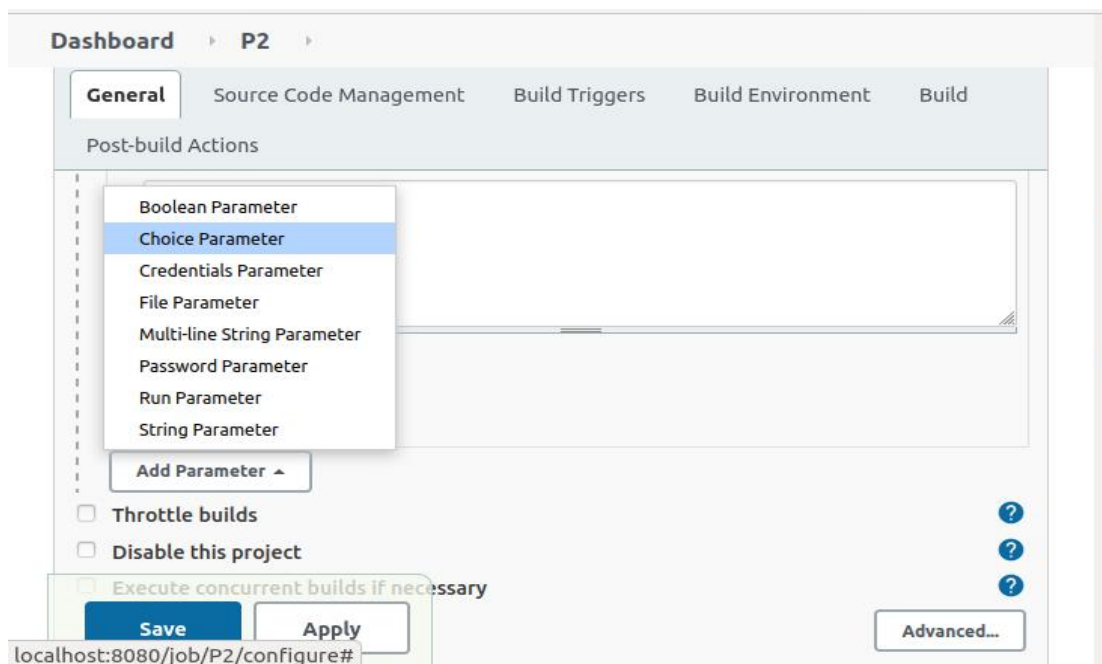
 **Parameterized job**

OK

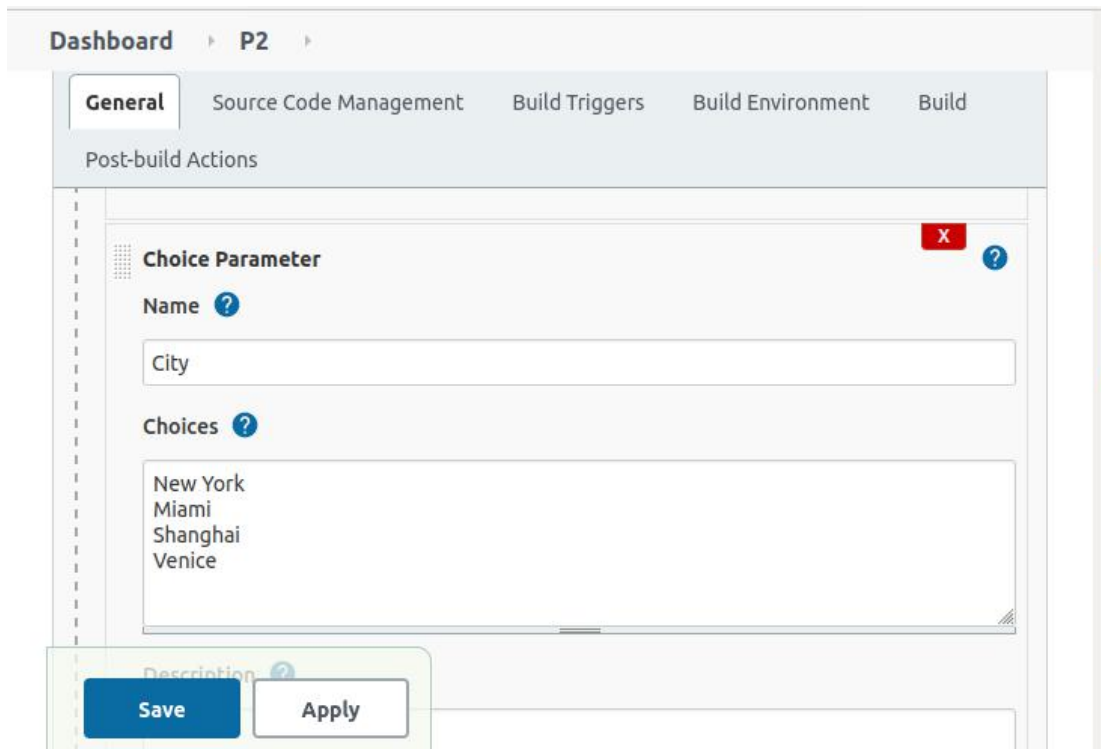
Step 2 : Click on general menu and select option this project is parameterize. Select String parameter and specify name as “First-Name”



Step 3 : Click on add parameter and select choice parameter. Take second parameter as choice parameter

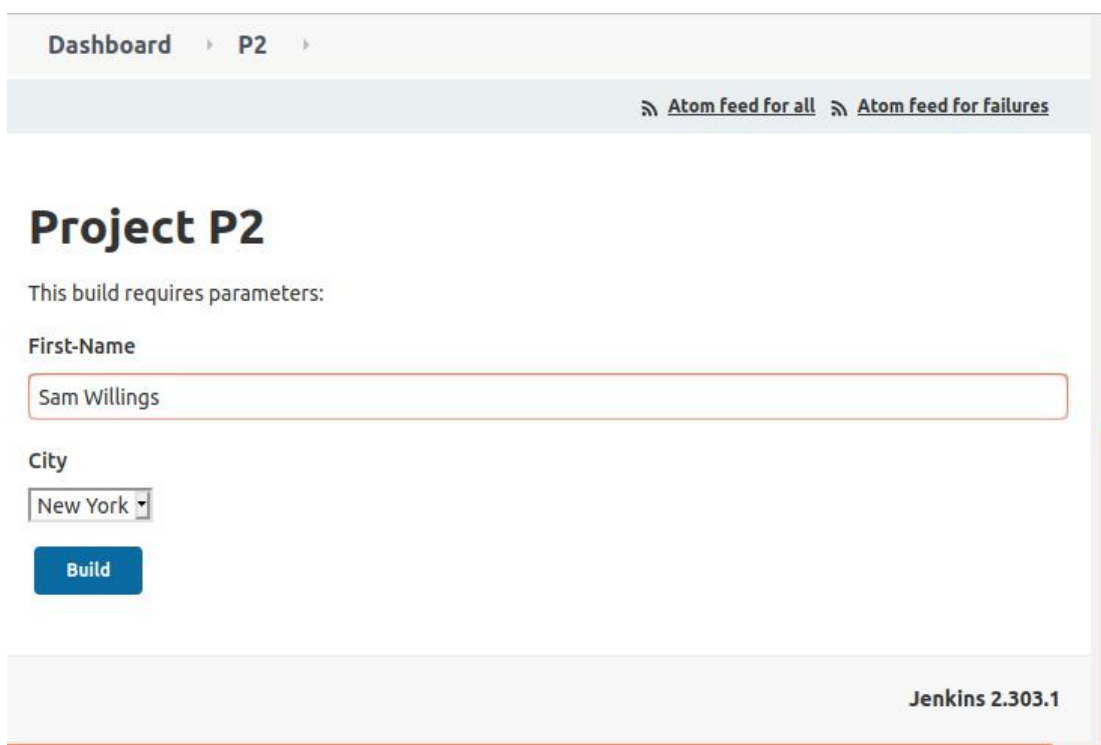


Step 4 : Specify name as “City” and add the choices in each line



The screenshot shows the Jenkins configuration page for 'Project P2'. The 'General' tab is selected. Under 'Post-build Actions', a 'Choice Parameter' is configured. The 'Name' field is set to 'City'. The 'Choices' field contains a list of cities: New York, Miami, Shanghai, and Venice. There are 'Save' and 'Apply' buttons at the bottom left of the configuration area.

Step 5 : Click on build with parameters and specify the values



The screenshot shows the Jenkins 'Project P2' build page. It indicates that the build requires parameters. The 'First-Name' parameter is set to 'Sam Willings'. The 'City' parameter is set to 'New York' via a dropdown menu. A 'Build' button is visible. At the top right, there are links for 'Atom feed for all' and 'Atom feed for failures'. The Jenkins version '2.303.1' is displayed at the bottom right.

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