Department of Information Technology

Academic Year: 2019-20 **Semester: VIII**
Class / Branch: BE IT
Subject: DevOps Lab (DL)
Subject Lab Incharge: Prof. Vishal S. Badgujar

EXPERIMENT NO. 02

Aim: To Install and Configure Jenkins to test, and deploy Java and Maven Applications using eclipse.

Theory:

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.” In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

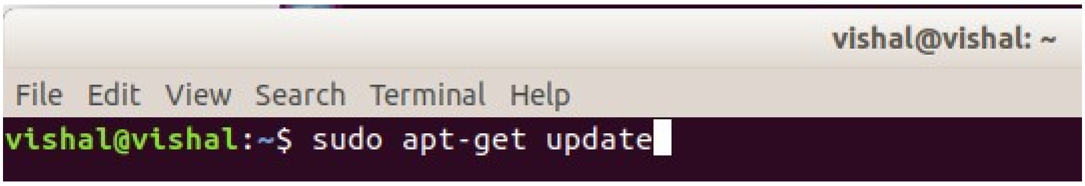
Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

To install Jenkins following software packages are required
1) GIT (git-scm.com)
2) Notepad++ (https://notepad-plus-plus.org/downloads/)
3) Latest Java development kit (JDK)
4) Jenkins
5) Apache Maven (Optional)

Installation Steps for Jenkins on Ubuntu 18.04

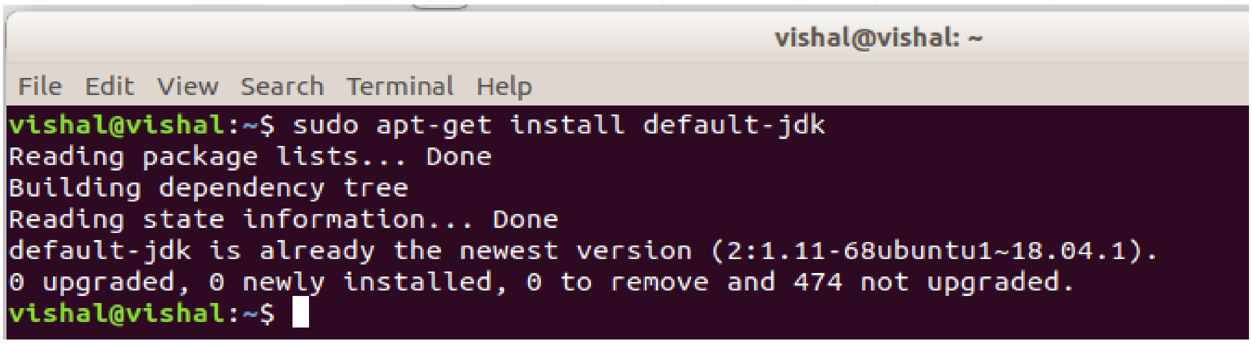
Step 1-: Update ubuntu repository
$sudo apt get update



```
vishal@vishal:~$ sudo apt-get update
```

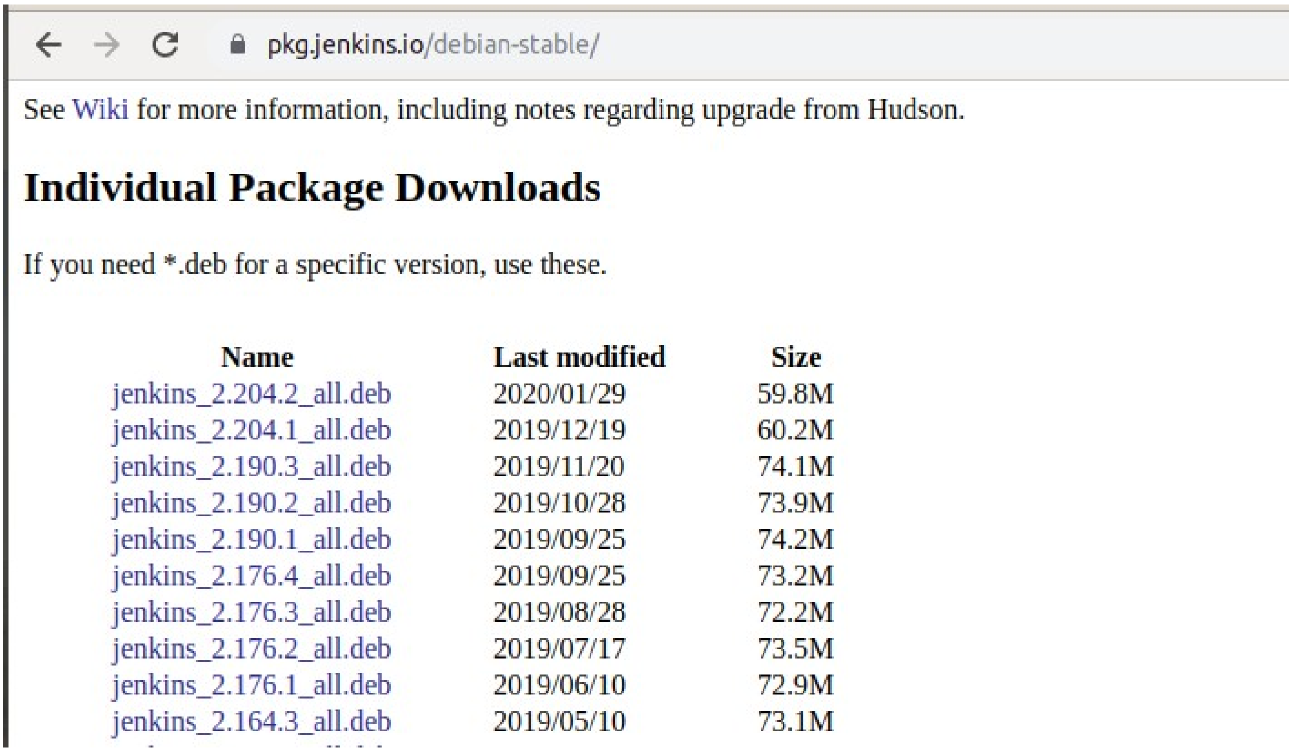
Step 2-: Install Java development kit

$sudo apt-get install default-jdk



```
vishal@vishal:~$ sudo apt-get install default-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
default-jdk is already the newest version (2:1.11-68ubuntu1~18.04.1).
0 upgraded, 0 newly installed, 0 to remove and 474 not upgraded.
vishal@vishal:~$
```

Step 3-: Download the latest jenkins .deb file from jenkins.io website by selecting ubuntu distribution.



See Wiki for more information, including notes regarding upgrade from Hudson.

Individual Package Downloads

If you need *.deb for a specific version, use these.

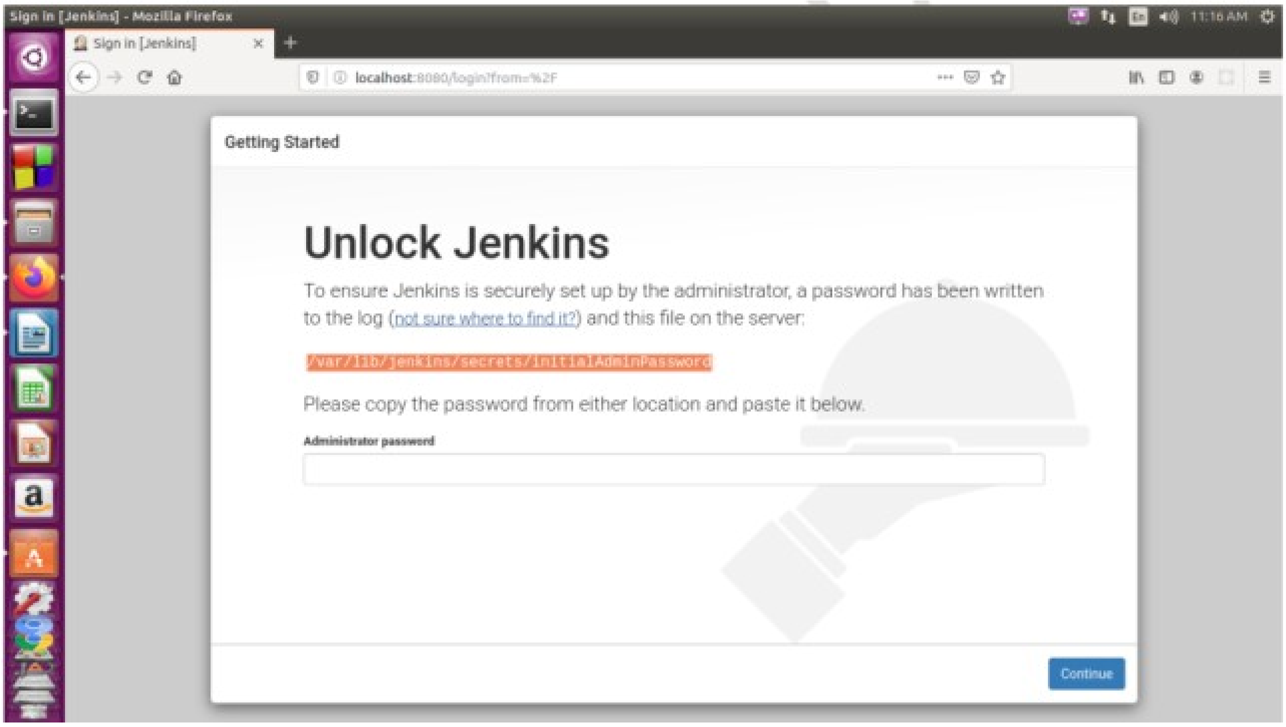
Name	Last modified	Size
jenkins_2.204.2_all.deb	2020/01/29	59.8M
jenkins_2.204.1_all.deb	2019/12/19	60.2M
jenkins_2.190.3_all.deb	2019/11/20	74.1M
jenkins_2.190.2_all.deb	2019/10/28	73.9M
jenkins_2.190.1_all.deb	2019/09/25	74.2M
jenkins_2.176.4_all.deb	2019/09/25	73.2M
jenkins_2.176.3_all.deb	2019/08/28	72.2M
jenkins_2.176.2_all.deb	2019/07/17	73.5M
jenkins_2.176.1_all.deb	2019/06/10	72.9M
jenkins_2.164.3_all.deb	2019/05/10	73.1M

Once downloaded, double click on file and open with software centre. Click on install button to perform installation.

Once Installation is done, you can test the jenkins on http://localhost:8080 on the browser.

First time, when you open jenkins portal it will ask to put admin default password which is stored in

/var/lib/jenkins/secrets/initialAdminPassword file.



Sign in [Jenkins] - Mozilla Firefox

localhost:8080/login?from=%2F

Getting Started

Unlock Jenkins

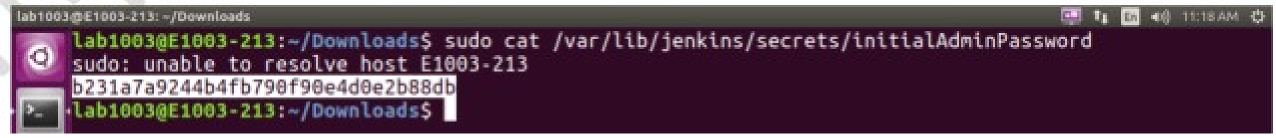
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

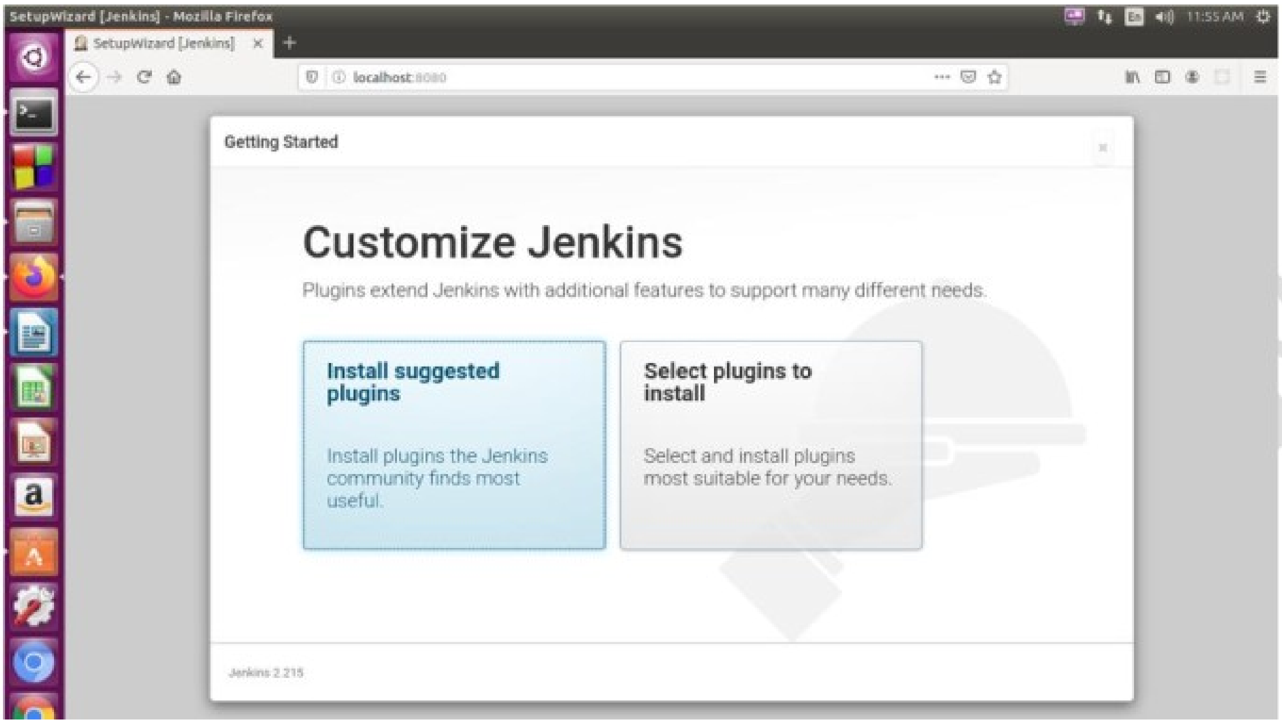
Administrator password

So, open the file through terminal and get the password for configuring jenkins for the first time.



```
lab1003@E1003-213: ~/Downloads
lab1003@E1003-213:~/Downloads$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
sudo: unable to resolve host E1003-213
b231a7a9244b4fb790f90e4d0e2b88db
lab1003@E1003-213:~/Downloads$
```

Now, copy the password and add into the portal and click on install plugins to install selected plugins.



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

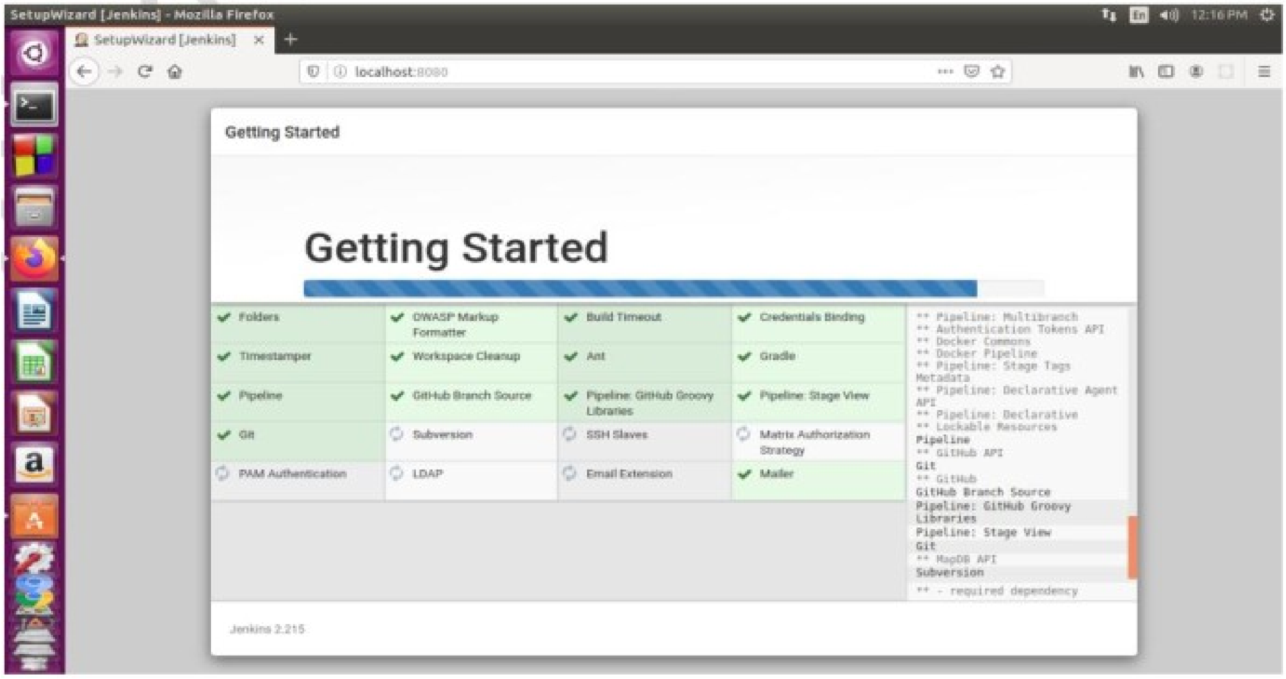
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.215



Getting Started

Getting Started

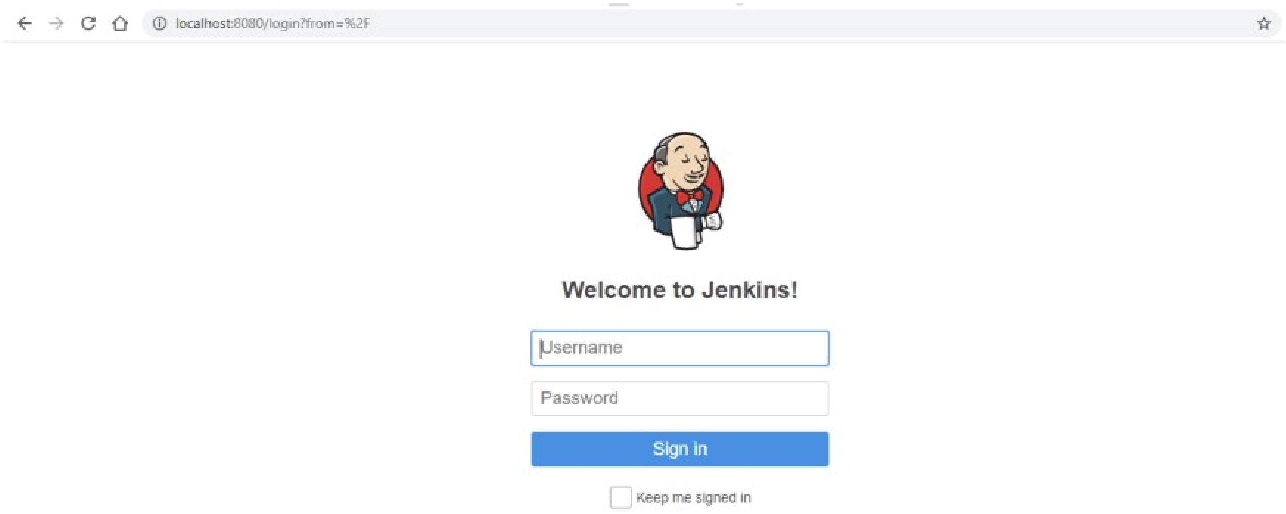
✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	Subversion	SSH Slaves	Matrix Authorization Strategy
PAM Authentication	LDAP	Email Extension	✓ Mailer

```
** Pipeline: Multibranch
** Authentication Tokens API
** Docker Commons
** Docker Pipeline
** Pipeline: Stage Tags
Metadata
** Pipeline: Declarative Agent
API
** Pipeline: Declarative
** Lockable Resources
Pipeline
** GitHub API
Git
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
Pipeline: Stage View
Git
** MapDB API
Subversion
** - required dependency
```

Jenkins 2.215

Once plugins are installed, click on next and specify the admin details along with the new password for jenkins admin and click on finish to complete the installation.

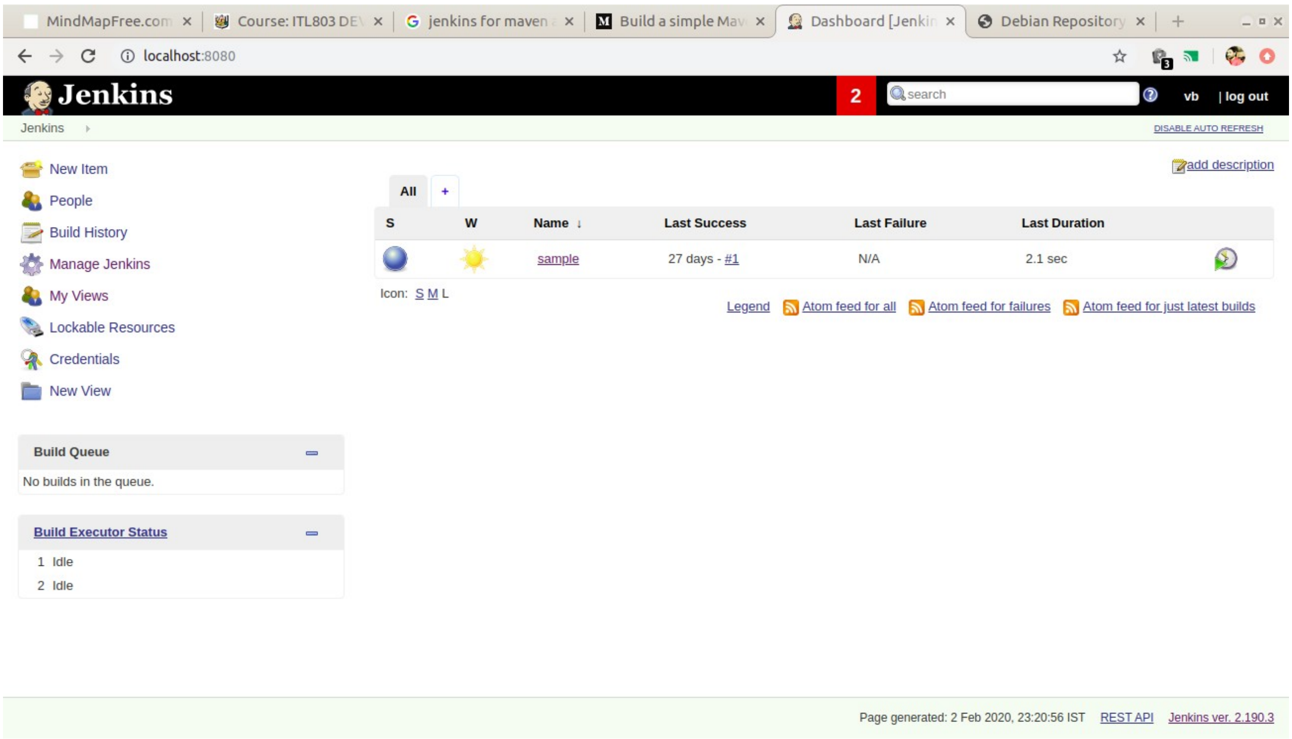
Once installation is done, open Jenkins dashboard using http://localhost:8080 address.



localhost:8080/login?from=%2F

Welcome to Jenkins!

So, by providing the right credentials you can access jenkins dashboard as shown below.



localhost:8080

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	sample	27 days - #1	N/A	2.1 sec

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

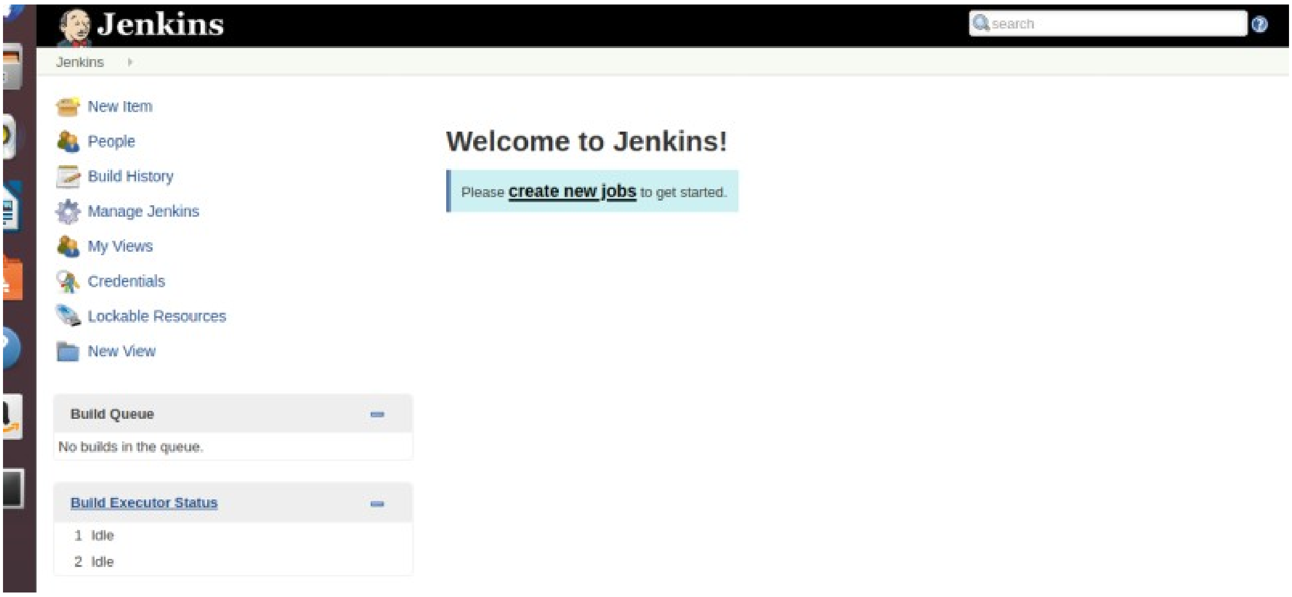
Page generated: 2 Feb 2020, 23:20:56 IST REST API Jenkins ver. 2.190.3

Example 1

Lab 1.1: Deploying a Freestyle App in Jenkins

The Steps for deploying a simple free style project in Jenkins is as Follows.

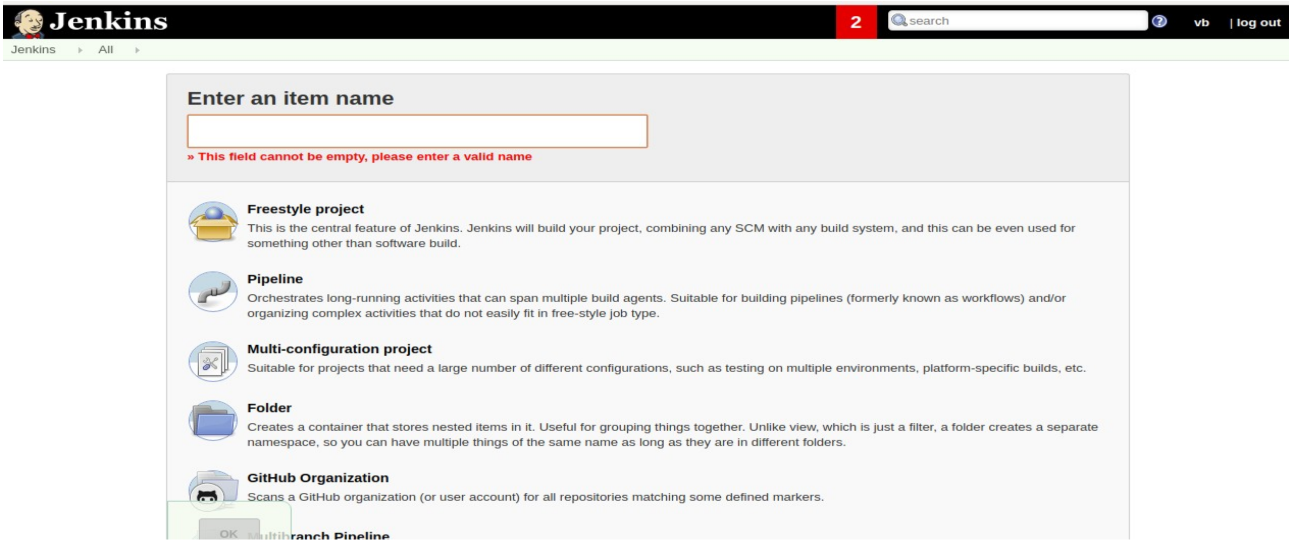
Step 1-: Click on Create new jobs.



Welcome to Jenkins!

Please **create new jobs** to get started.

Step 2-: Now Specify name to the project as "Example1", select Option "Free style project " and click on OK button



Enter an item name

» This field cannot be empty, please enter a valid name

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.

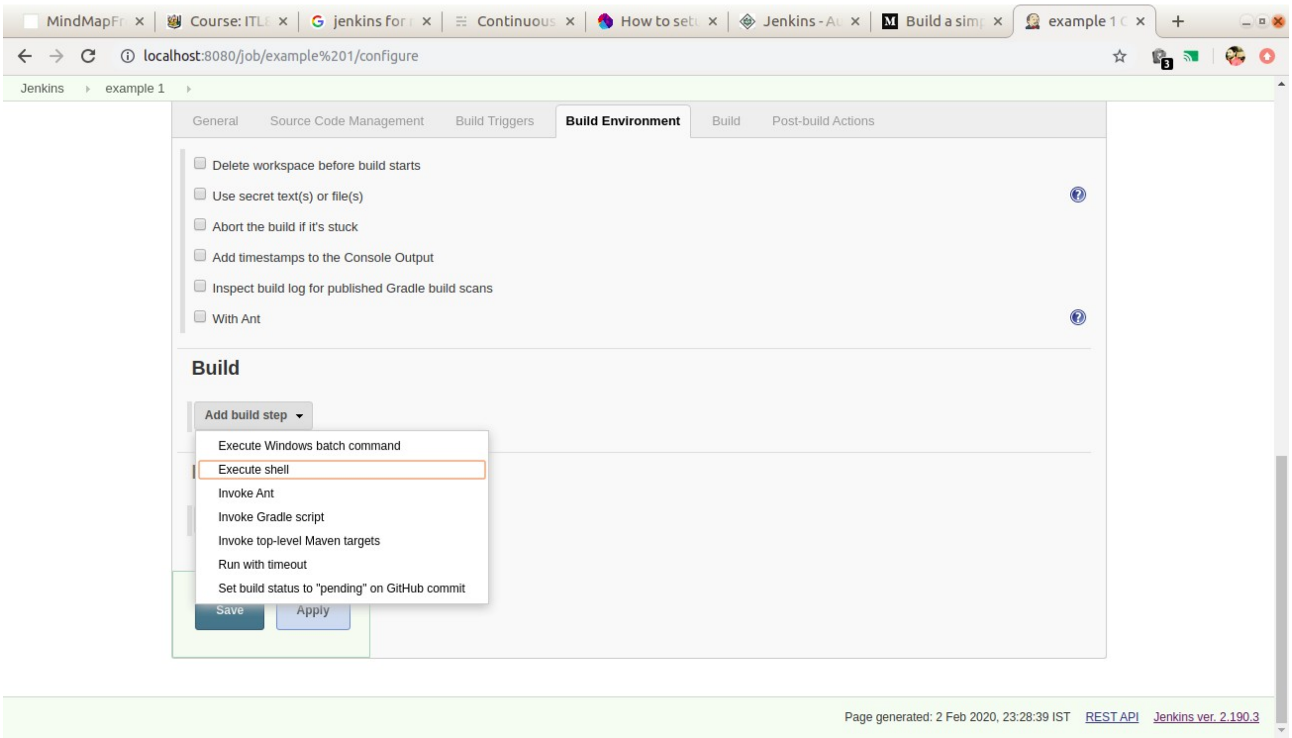
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

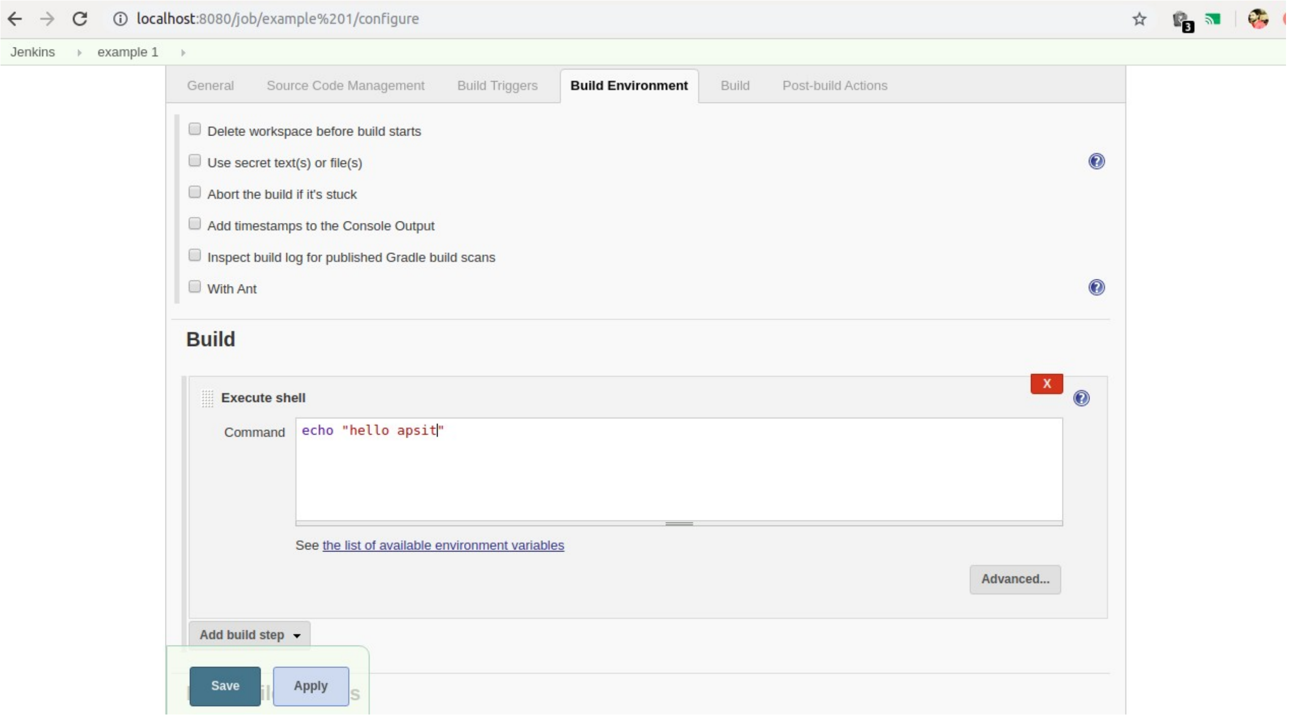
GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline

Step 3-: In this project we are going to learn how to run simple shell script on Jenkins. So, Click on Build option select Execute script from dropdown menu.


MindMapFr
Course: ITLE
jenkins for
Continuous
How to setu
Jenkins - Au
Build a simp
example 1
localhost:8080/job/example%201/configure
Jenkins
example 1
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Delete workspace before build starts
Use secret text(s) or file(s)
Abort the build if it's stuck
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
With Ant
Build
Add build step
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
Save
Apply
Page generated: 2 Feb 2020, 23:28:39 IST
REST API
Jenkins ver. 2.190.3

Step 4-: Now Write a Simple Shell command to print the text as Like given below.

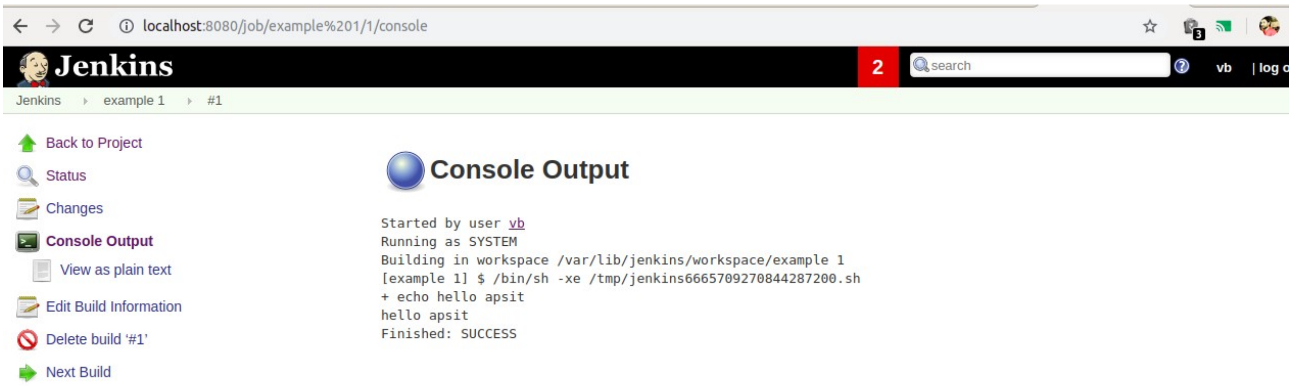

localhost:8080/job/example%201/configure
Jenkins
example 1
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Delete workspace before build starts
Use secret text(s) or file(s)
Abort the build if it's stuck
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
With Ant
Build
Execute shell
X
See the list of available environment variables
Advanced...
Add build step
Save
Apply

Command

```
echo "hello apsit"
```

Now click on apply followed by save button.

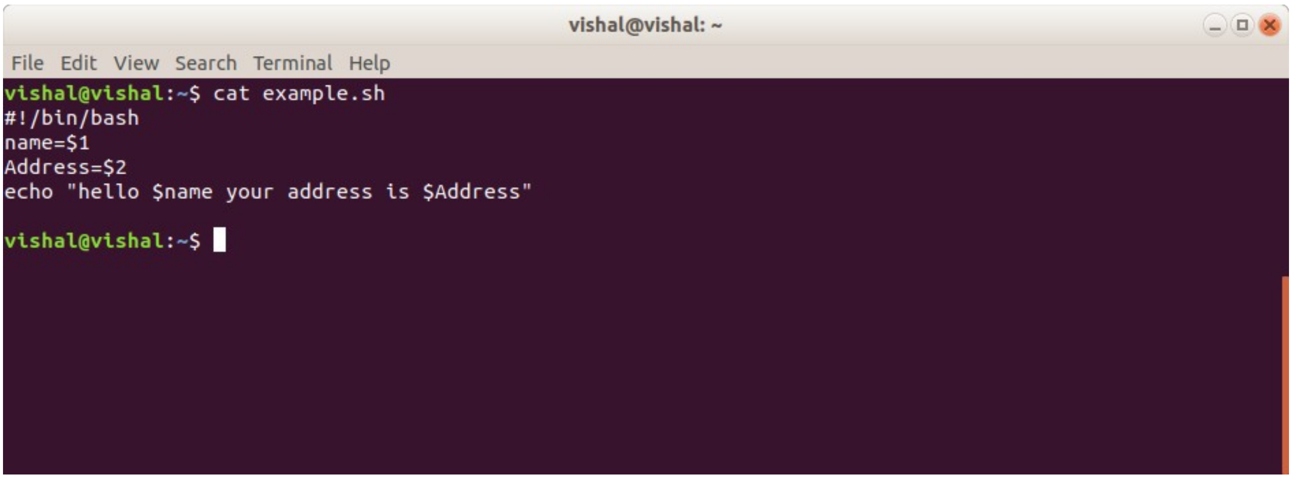
Step 5-: Now Build a project to see the output Click on our first build "1" followed by console output to see the output.



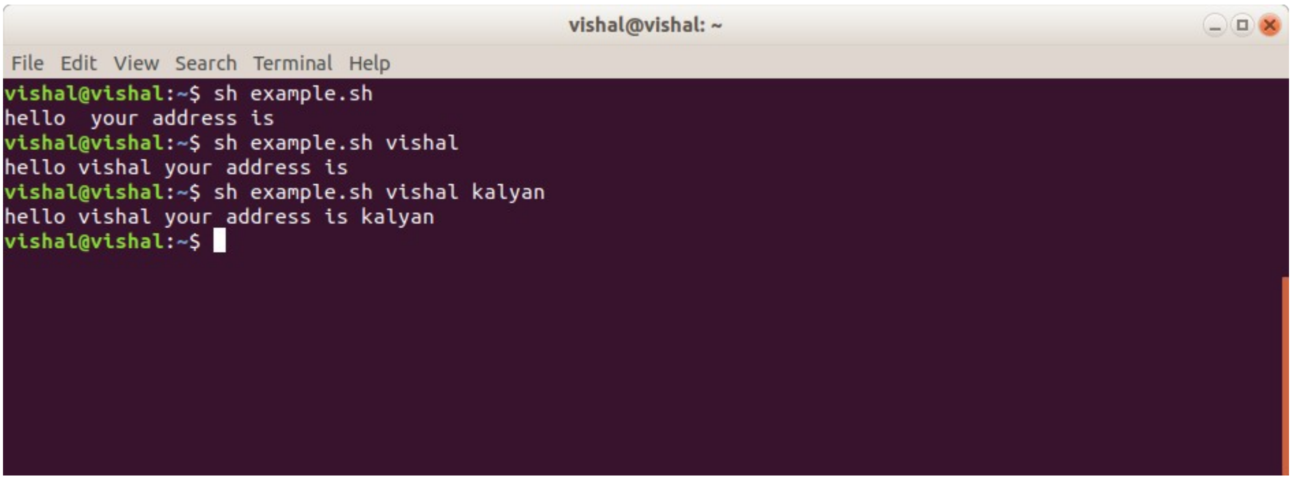
```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example 1
[example 1] $ /bin/sh -xe /tmp/jenkins6665709270844287200.sh
+ echo hello apsit
hello apsit
Finished: SUCCESS
```

Example 1.2:

Now let us take parameters through files. So, create a new shell script file in local directory.



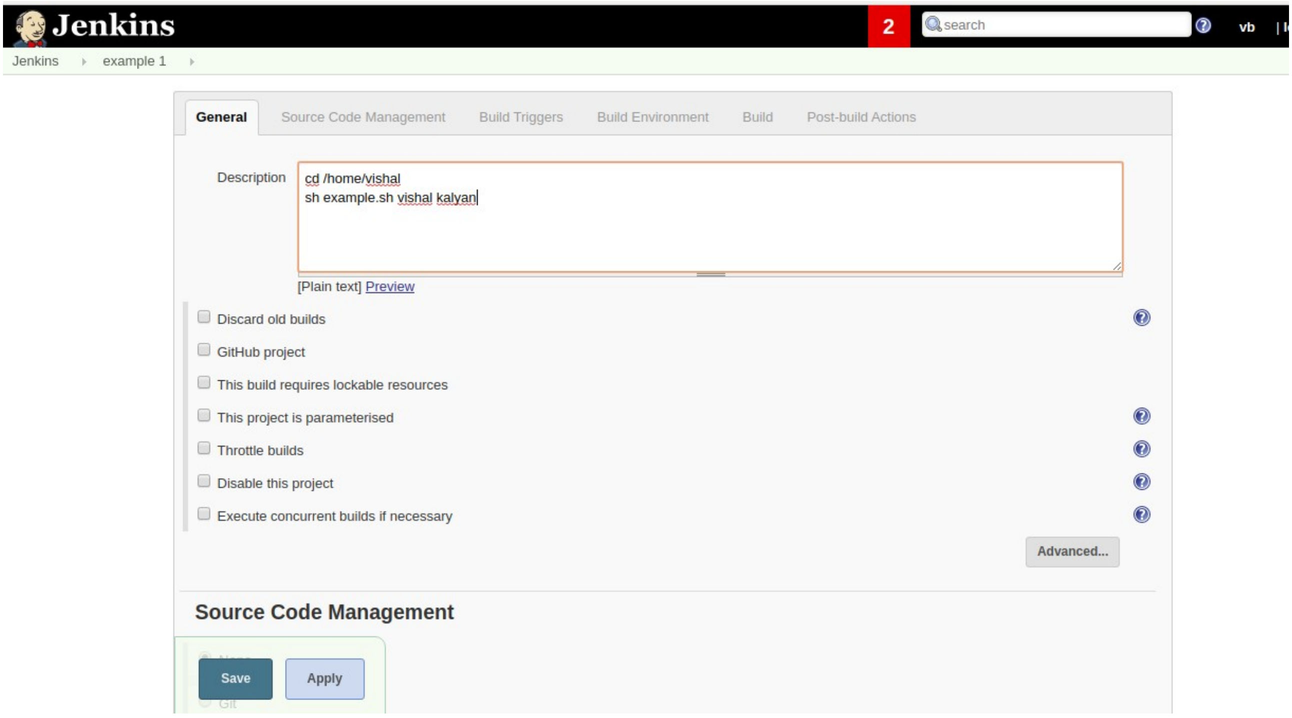
```
vishal@vishal:~$ cat example.sh
#!/bin/bash
name=$1
Address=$2
echo "hello $name your address is $Address"

vishal@vishal:~$ 
```



```
vishal@vishal:~$ sh example.sh
hello  your address is
vishal@vishal:~$ sh example.sh vishal
hello vishal your address is
vishal@vishal:~$ sh example.sh vishal kalyan
hello vishal your address is kalyan
vishal@vishal:~$ 
```

Now first run the shell script locally with no parameter, one parameter and two parameters.

Now. Let us run it through Jenkins Shell. To change existing program, click on configure option and then modify the script. <Here, change directory to the path where you have stored your file..You can get the location by pwd command>



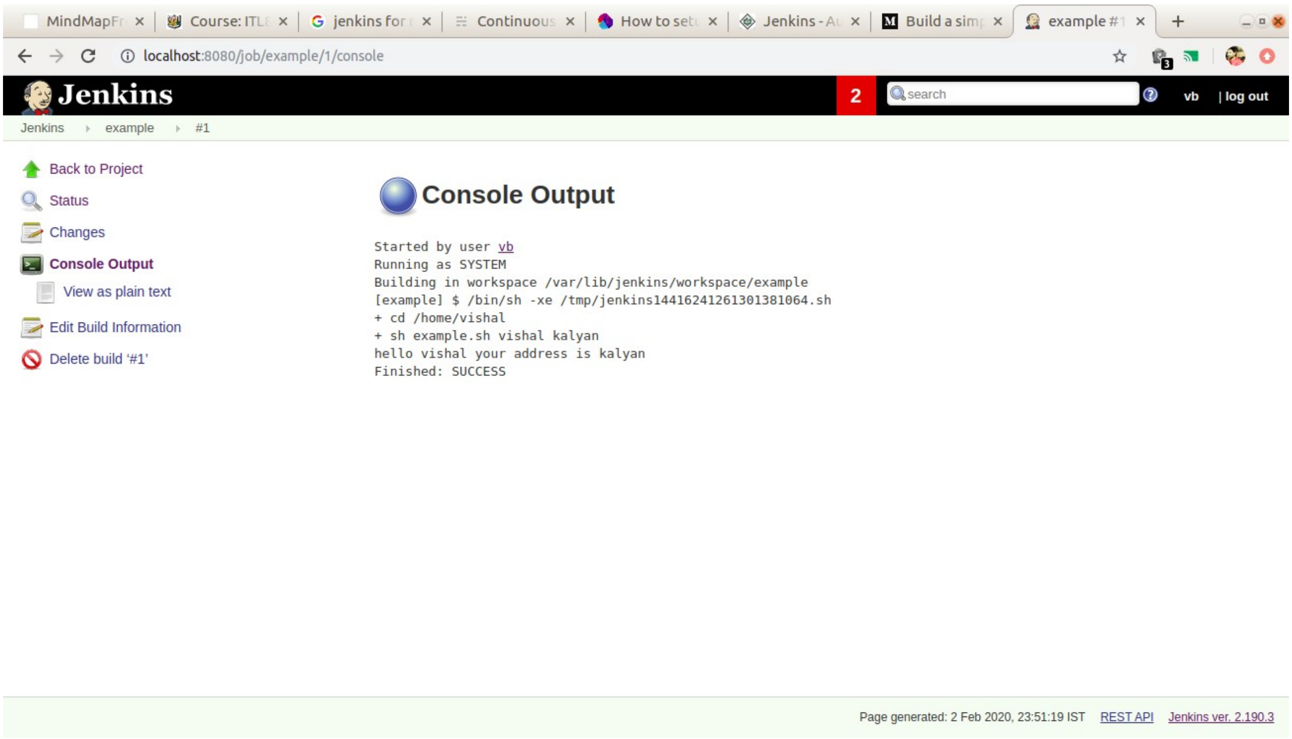
Description

```
cd /home/vishal
sh example.sh vishal kalyan
```

[Plain text] Preview

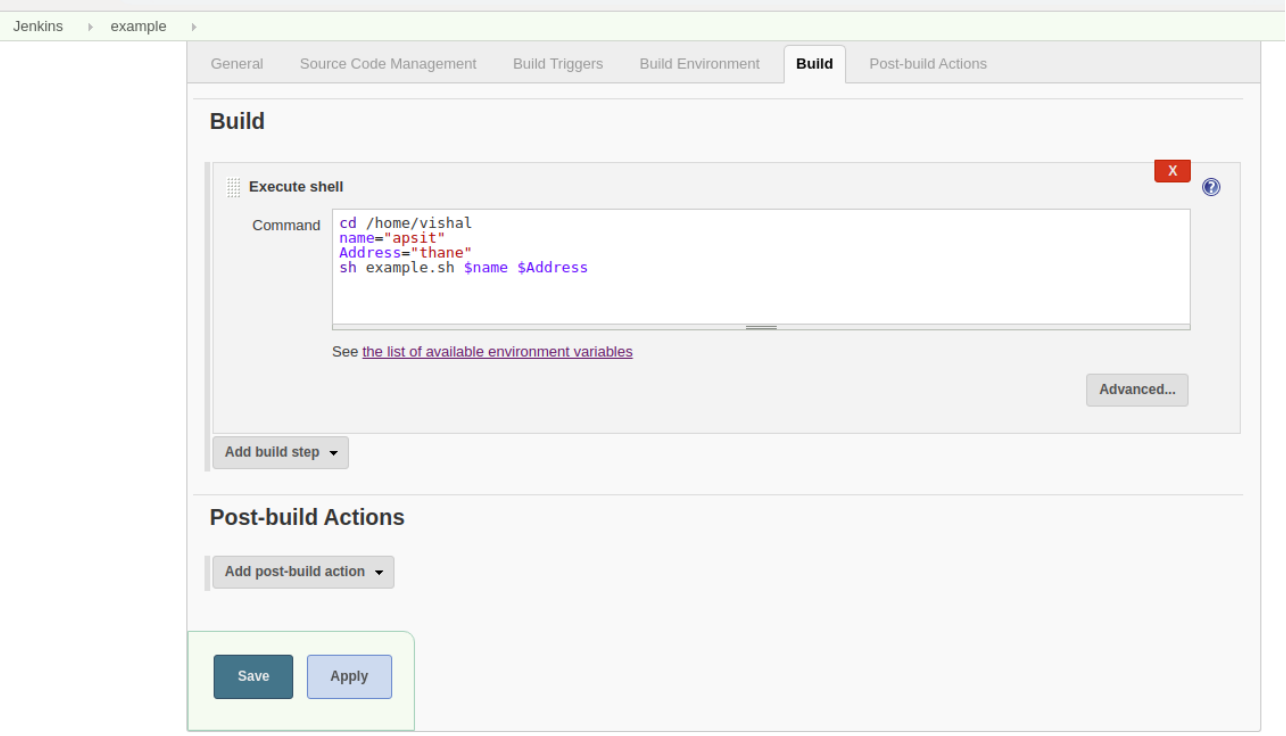
□ Discard old builds
□ GitHub project
□ This build requires lockable resources
□ This project is parameterised
□ Throttle builds
□ Disable this project
□ Execute concurrent builds if necessary

Source Code Management

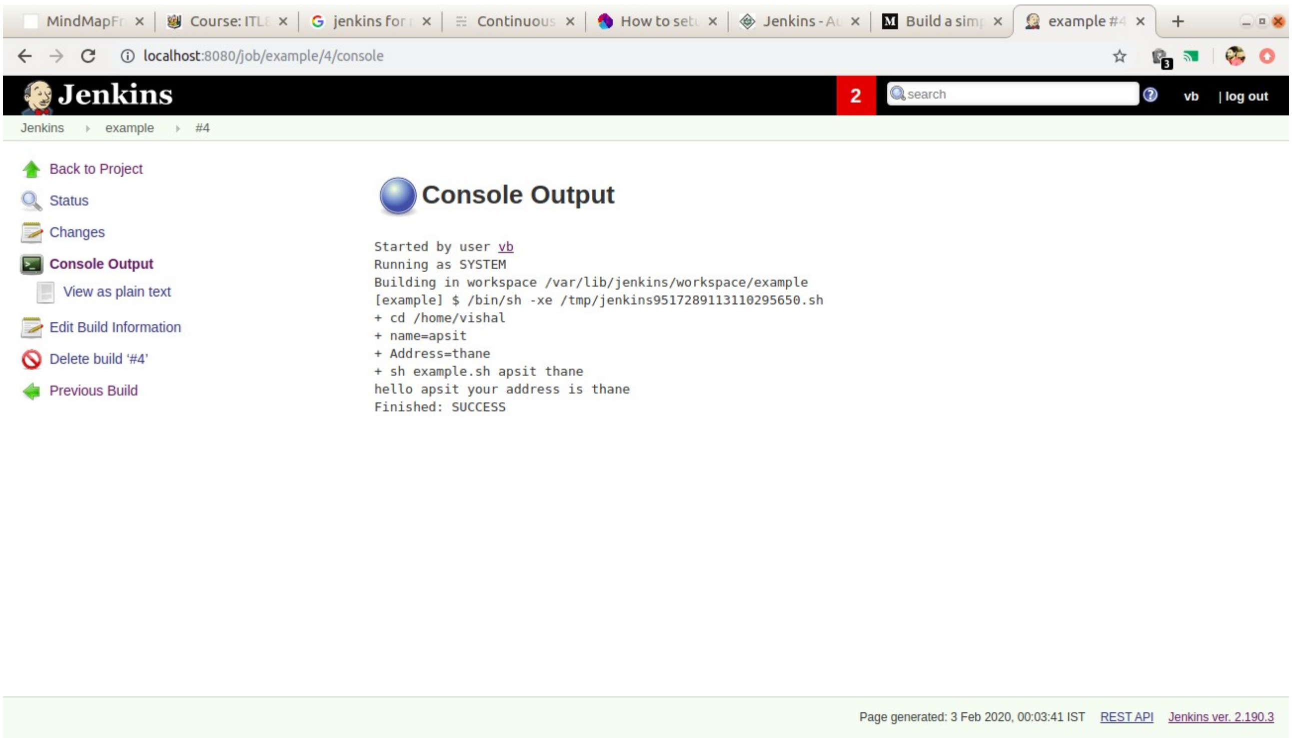


```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example
[example] $ /bin/sh -xe /tmp/jenkins14416241261301381064.sh
+ cd /home/vishal
+ sh example.sh vishal kalyan
hello vishal your address is kalyan
Finished: SUCCESS
```

1.3 Variation to this program



```
cd /home/vishal
name="apsit"
Address="thane"
sh example.sh $name $Address
```

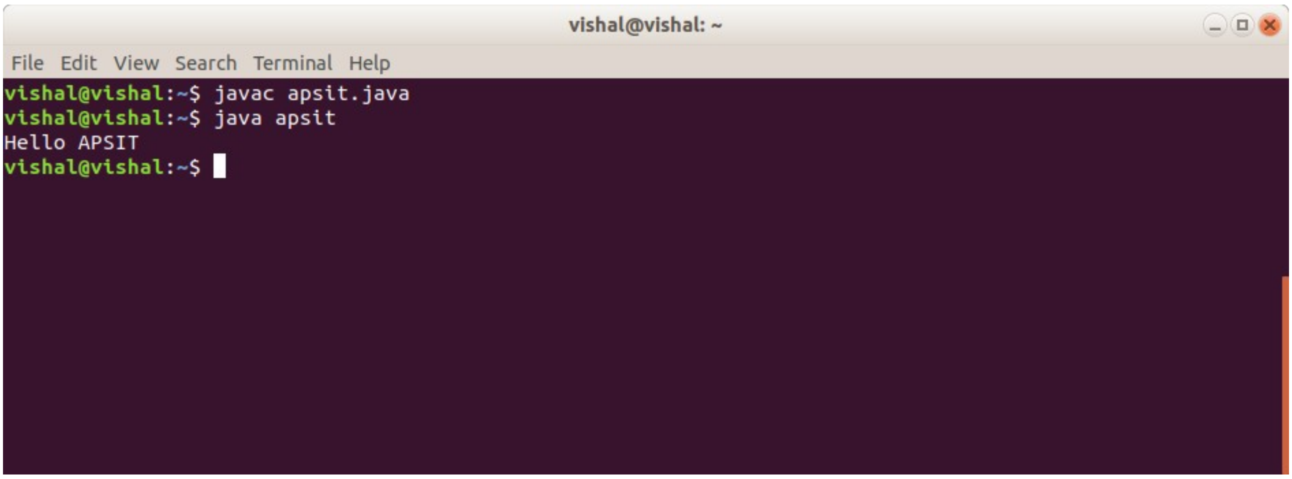


```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example
[example] $ /bin/sh -xe /tmp/jenkins9517289113110295650.sh
+ cd /home/vishal
+ name=apsit
+ Address=thane
+ sh example.sh apsit thane
hello apsit your address is thane
Finished: SUCCESS
```

Example 2

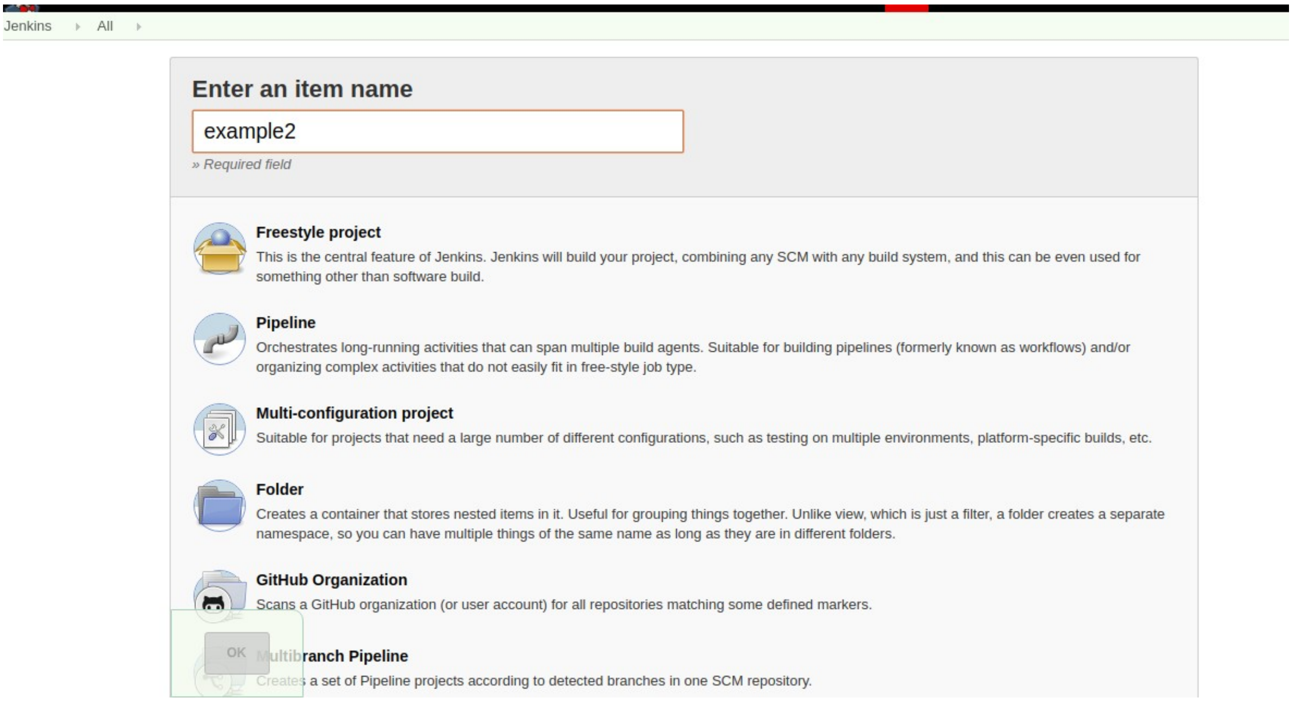
1.4-: Running a Java Program under jenkins

Step 1 -: Write a java program and test it locally



```
vishal@vishal:~$ javac apsit.java
vishal@vishal:~$ java apsit
Hello APSIT
vishal@vishal:~$ 
```

Now create freestyle project example2 in jenkins



Enter an item name

example2

» 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.

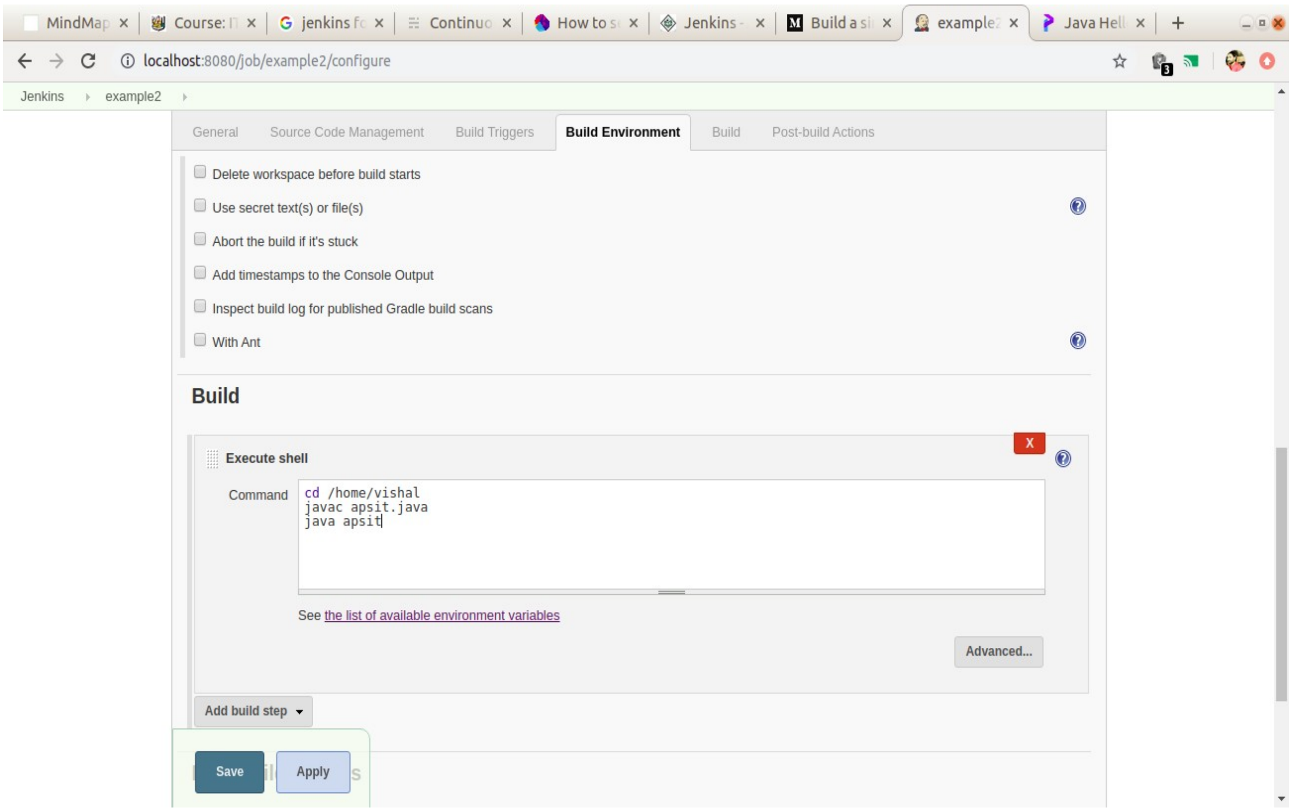
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Go to build option and change path to directory where you have stored java file followed by compile and run program commands.



□ Delete workspace before build starts

□ Use secret text(s) or file(s)

□ Abort the build if it's stuck

□ Add timestamps to the Console Output

□ Inspect build log for published Gradle build scans

□ With Ant

Build

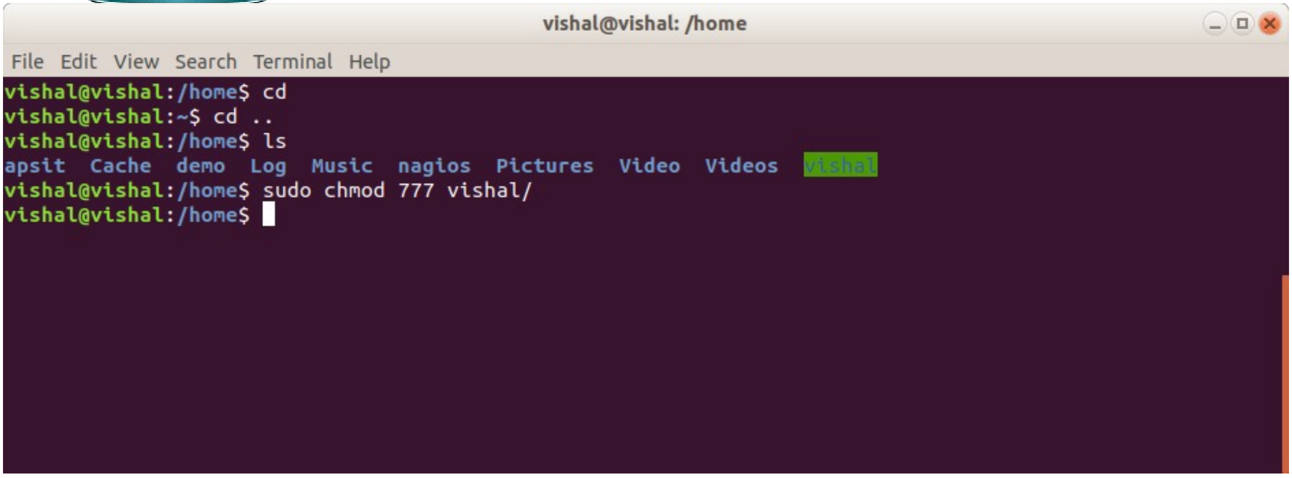
Execute shell

Command

```
cd /home/vishal
javac apsit.java
java apsit
```

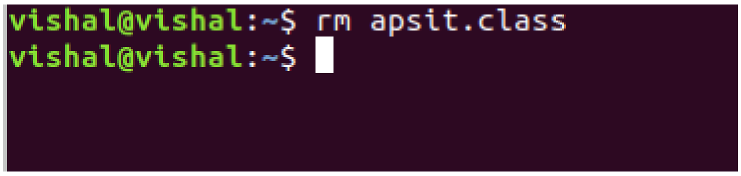
See the list of available environment variables

Now if your build fails due to permission problem then give write permission to directory underneth it.



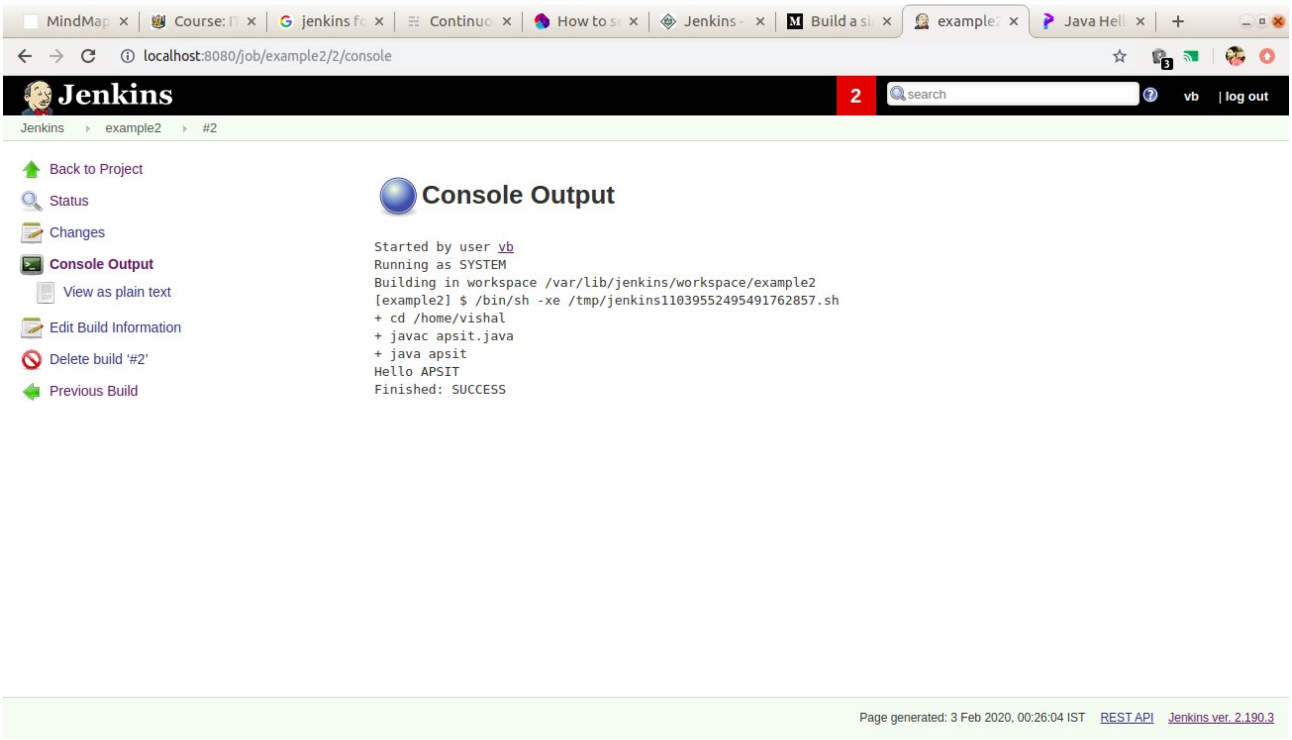
```
vishal@vishal:/home$ cd
vishal@vishal:~$ cd ..
vishal@vishal:/home$ ls
apsit  Cache  demo  Log  Music  nagios  Pictures  Video  Videos  vishal
vishal@vishal:/home$ sudo chmod 777 vishal/
vishal@vishal:/home$ 
```

And remove existing .class file from your current directory



```
vishal@vishal:~$ rm apsit.class
vishal@vishal:~$ 
```

Now after doing this your build gets successful and get results



Console Output

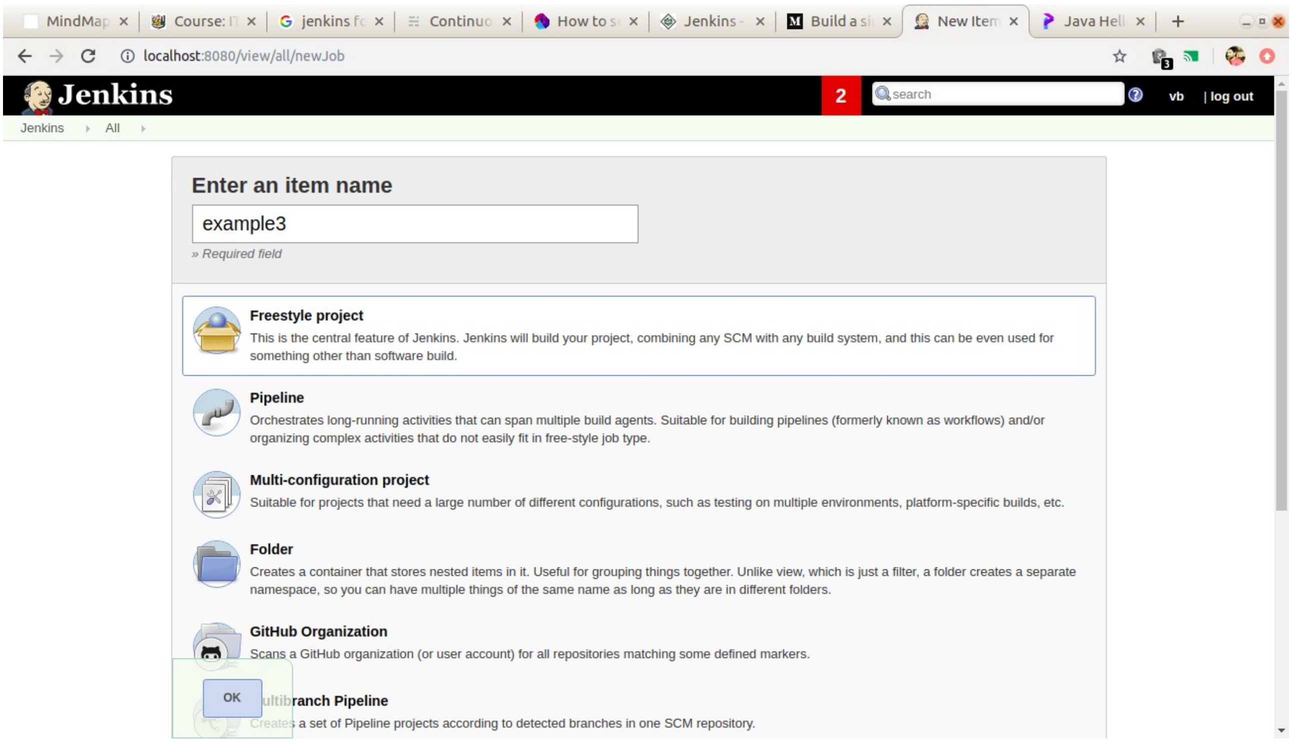
```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example2
[example2] $ /bin/sh -xe /tmp/jenkins11039552495491762857.sh
+ cd /home/vishal
+ javac apsit.java
+ java apsit
Hello APSIT
Finished: SUCCESS
```

Example 3

1.5: Parameterize Build

In this program we are going to see how to provide parameters during runtime to your shell script or java program.

Step 1-: Create a free style project example3 by clicking on new item folowed by specifying project name and free style project.



Enter an item name

example3

» 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.

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

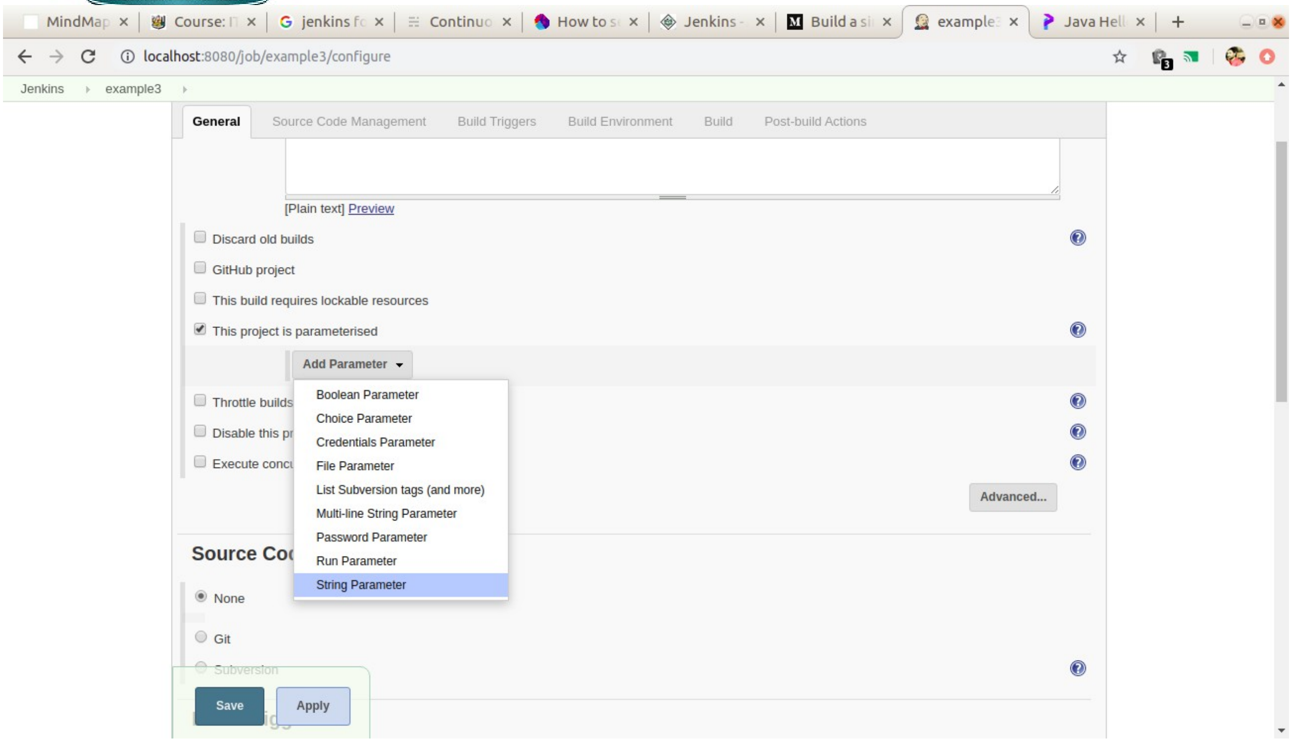
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

ultibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Step 2-: Now under general menu, select option this project is parameterize

Select String parameter and specify name as "First-Name"



[Plain text] Preview

□ Discard old builds

□ GitHub project

□ This build requires lockable resources

☑ This project is parameterised

□ Throttle builds

□ Disable this pr

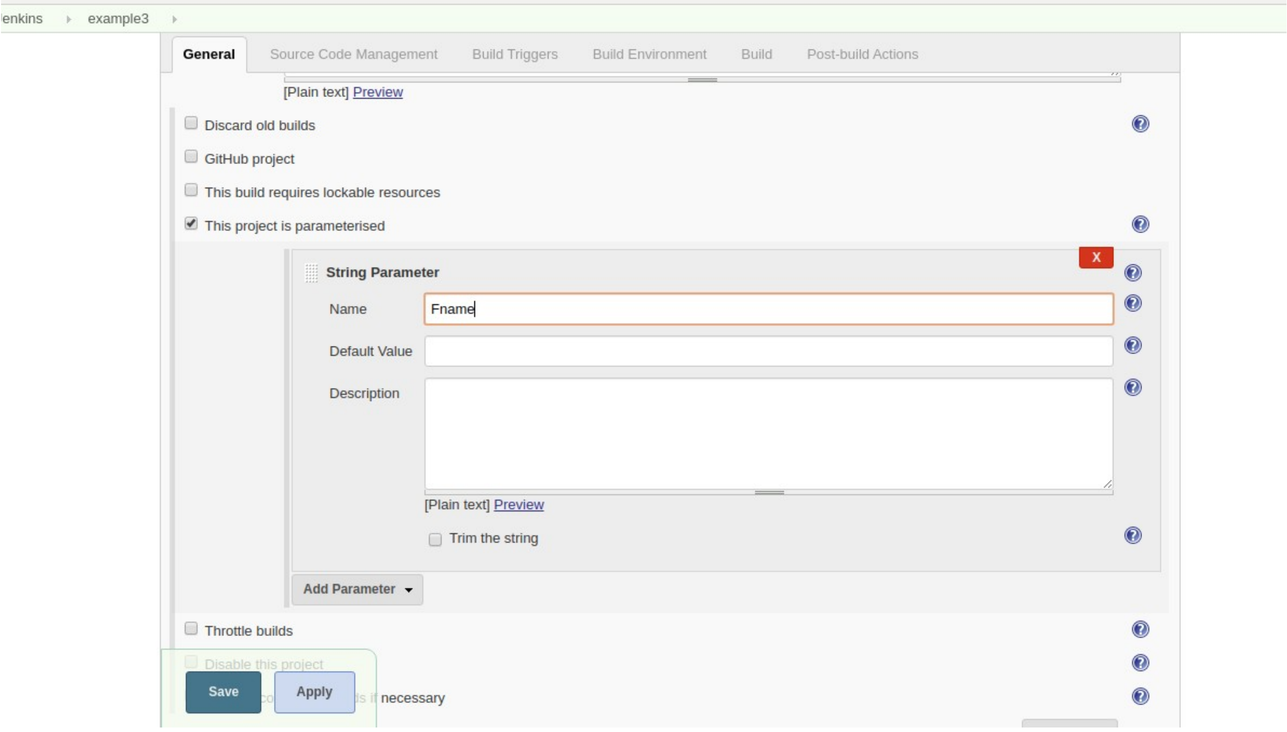
□ Execute conc

Source Coc

● None

○ Git

○ Subversion



[Plain text] Preview

□ Discard old builds

□ GitHub project

□ This build requires lockable resources

☑ This project is parameterised

String Parameter

Name: Fname

Default Value:

Description:

[Plain text] Preview

□ Trim the string

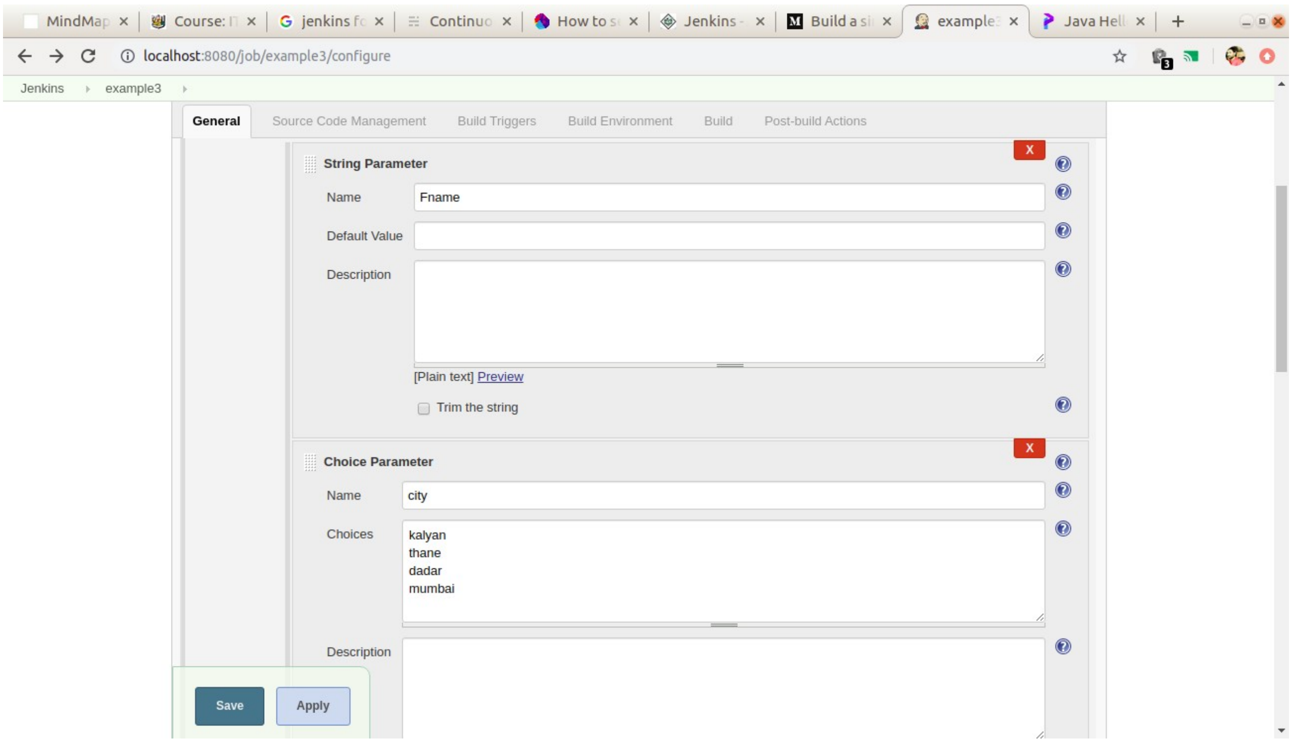
□ Throttle builds

□ Disable this project

necessary

Again, click on add parameter and select choice parameter Take second parameter as choice box.

Specify name as "City" and add the choices in each line



String Parameter

Name: Fname

Default Value:

Description:

[Plain text] Preview

□ Trim the string

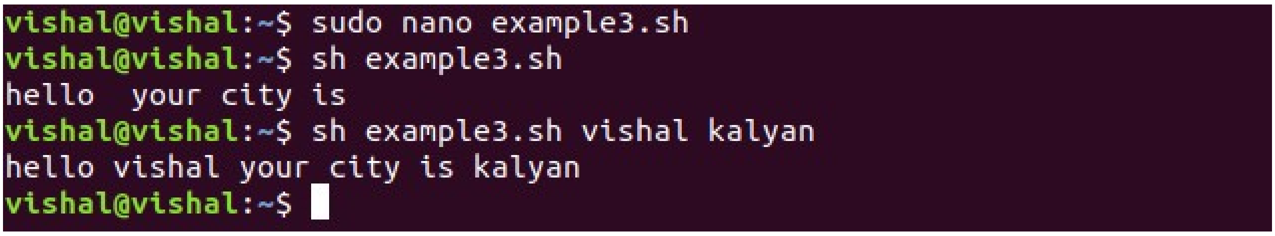
Choice Parameter

Name: city

Choices:
kalyan
thane
dadar
mumbai

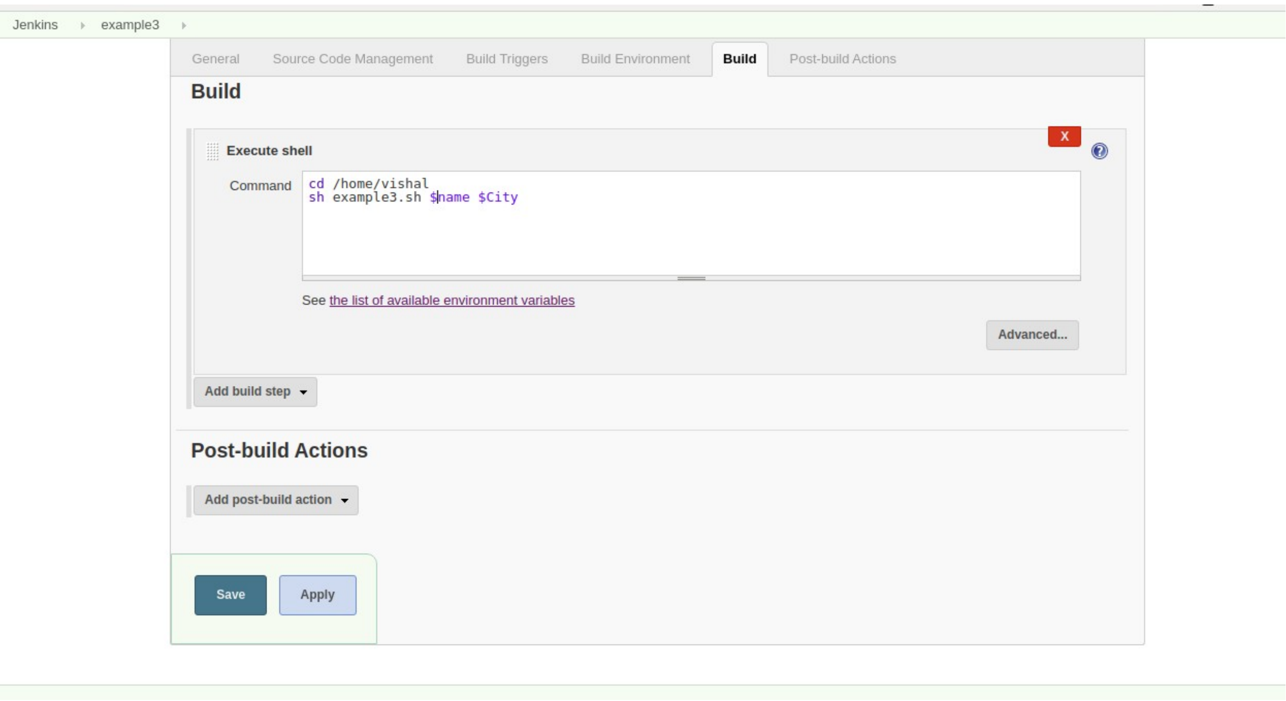
Description:

Write a shell script that takes 2 parameters with command line arguments name and city.



```
vishal@vishal:~$ sudo nano example3.sh
vishal@vishal:~$ sh example3.sh
hello  your city is
vishal@vishal:~$ sh example3.sh vishal kalyan
hello vishal your city is kalyan
vishal@vishal:~$ 
```

Now, go back to jenkins, Selct Build option, give the path and write script as shown below

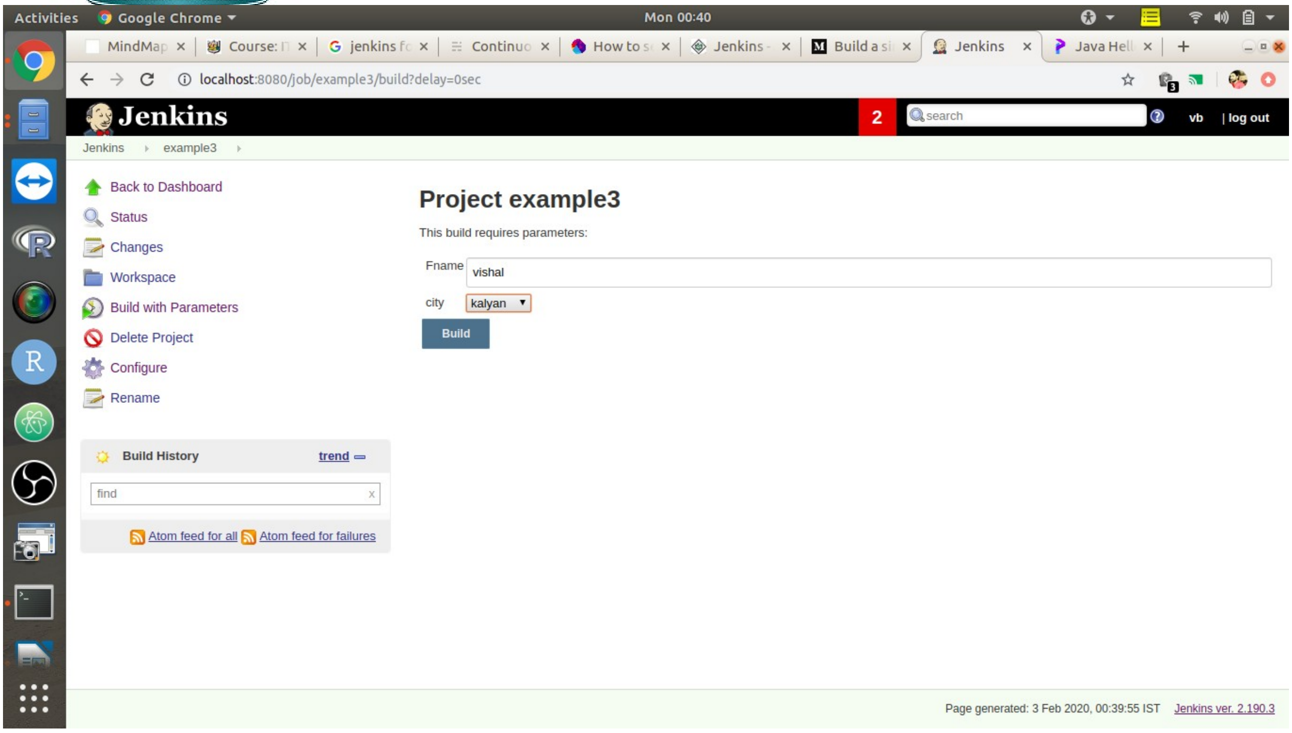


```
cd /home/vishal
sh example3.sh $name $City
```

See the list of available environment variables

Advanced...

Now click on build with parameters and specify the values



Project example3

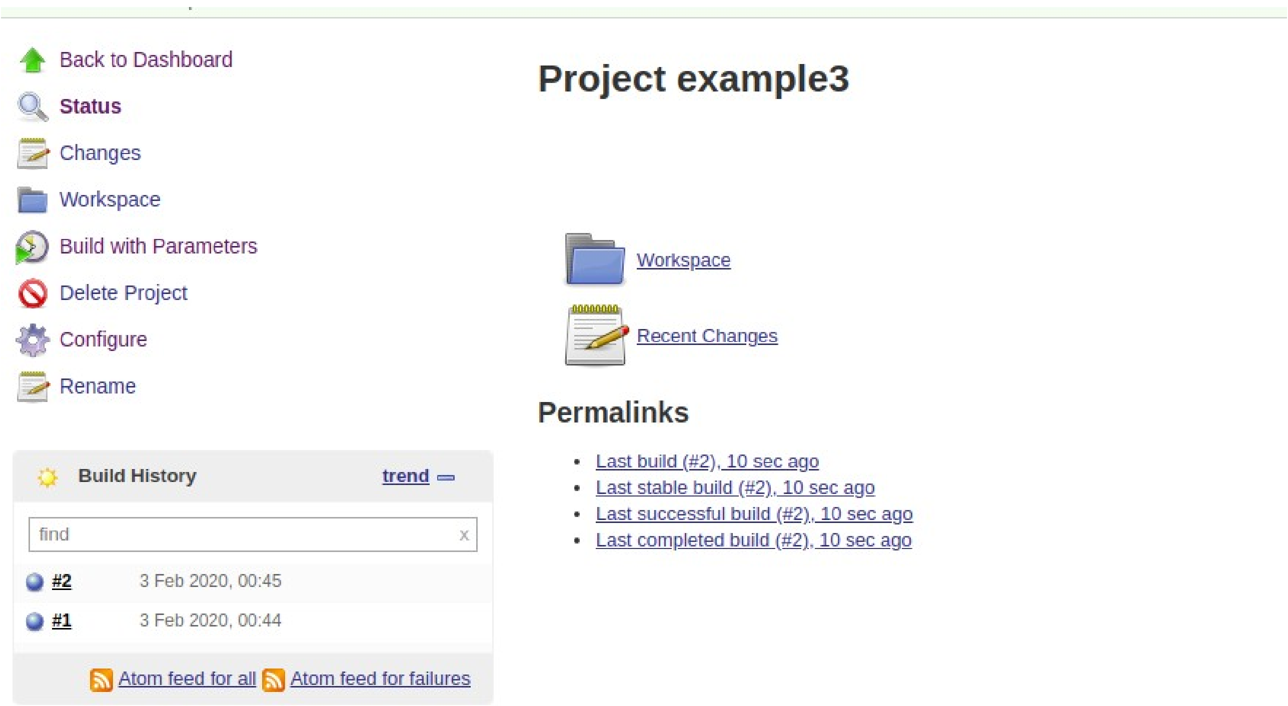
This build requires parameters:

Fname vishal

city kalyan

Page generated: 3 Feb 2020, 00:39:55 IST Jenkins ver. 2.190.3

Click on Build

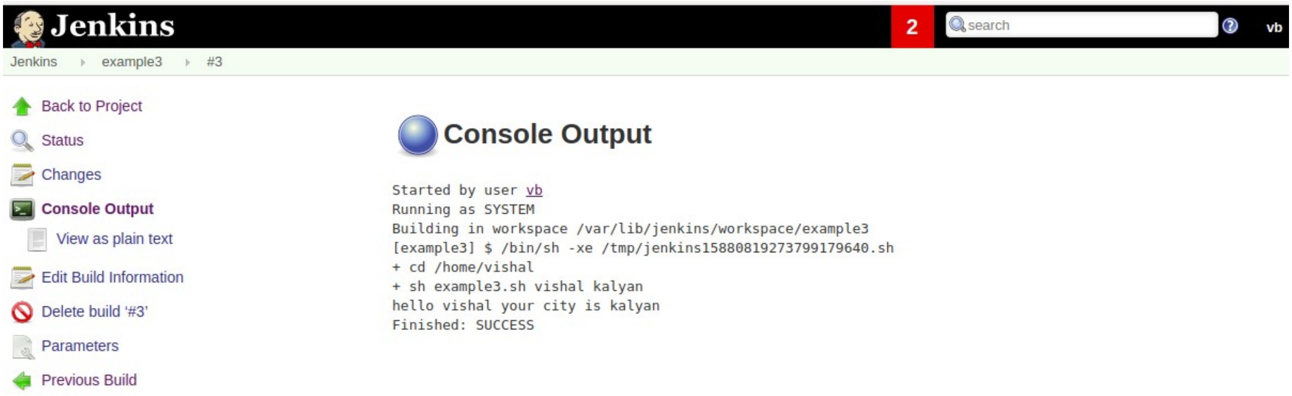


Project example3

#2	3 Feb 2020, 00:45
#1	3 Feb 2020, 00:44

Permalinks

- Last build (#2), 10 sec ago
- Last stable build (#2), 10 sec ago
- Last successful build (#2), 10 sec ago
- Last completed build (#2), 10 sec ago



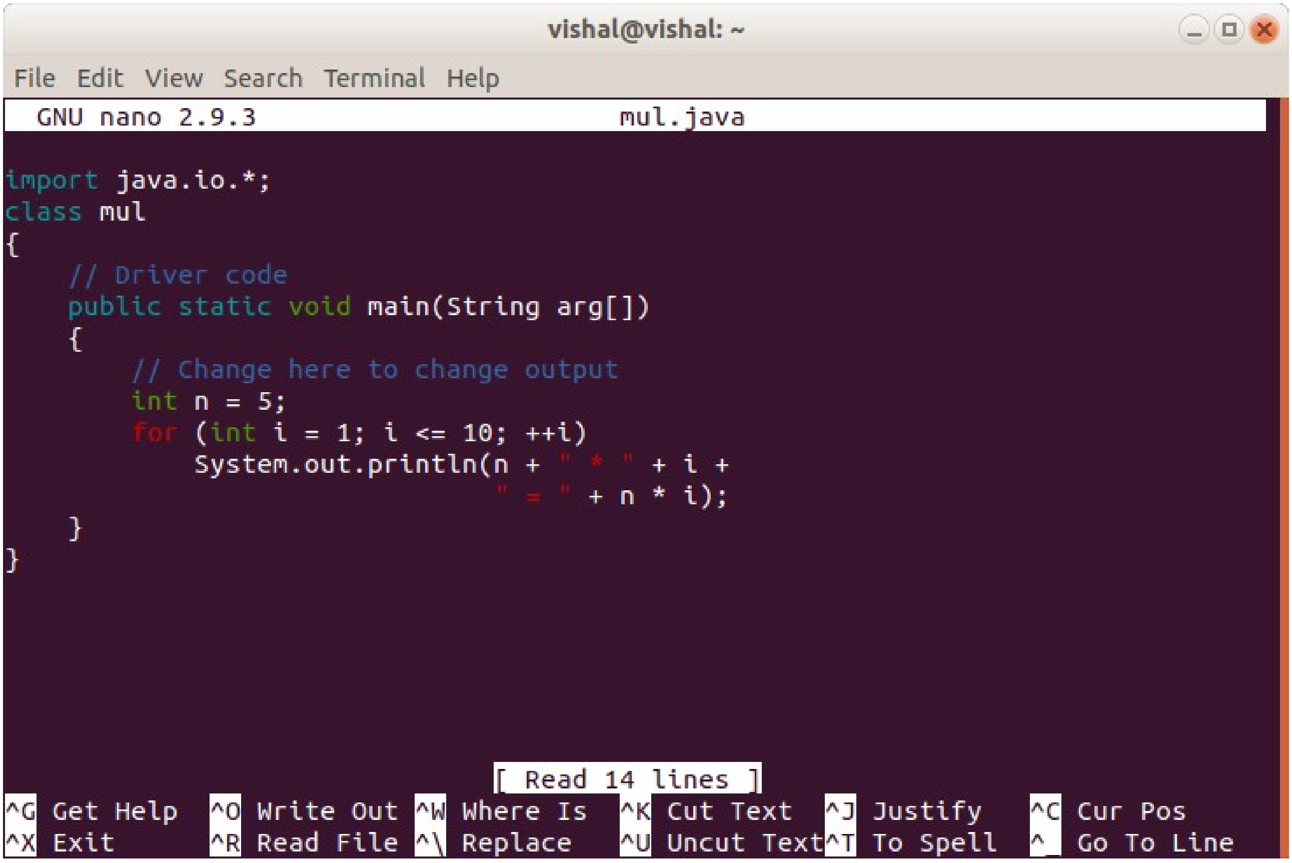
Console Output

```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example3
[example3] $ /bin/sh -xe /tmp/jenkins15880819273799179640.sh
+ cd /home/vishal
+ sh example3.sh vishal kalyan
hello vishal your city is kalyan
Finished: SUCCESS
```

Go to console to see the output

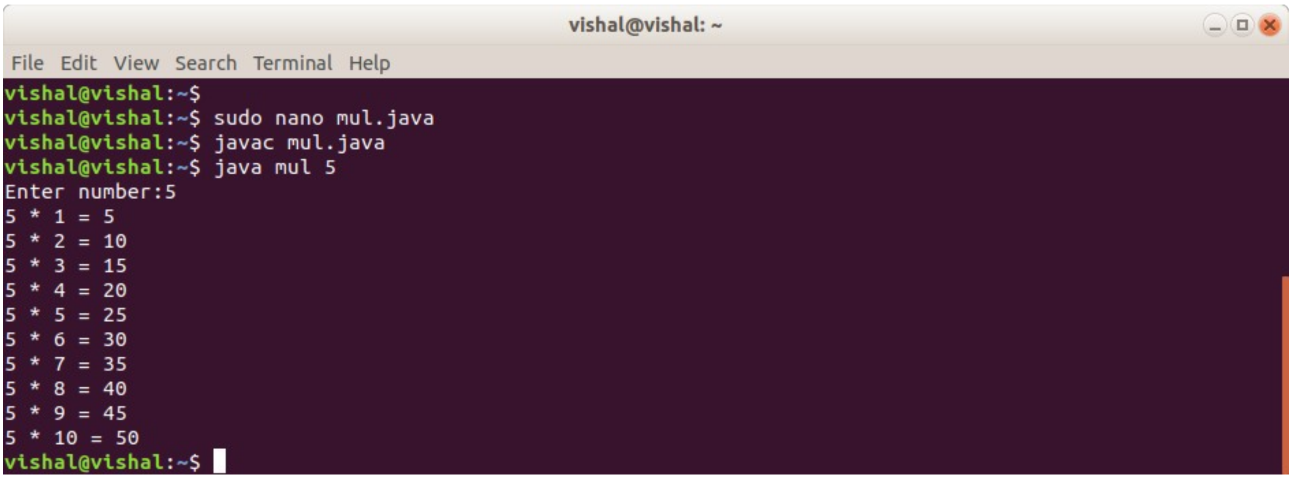
1.6 Running a Java program with parameters

Write a java program for multiplication table with command line arguments and test it locally.


vishal@vishal: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 mul.java

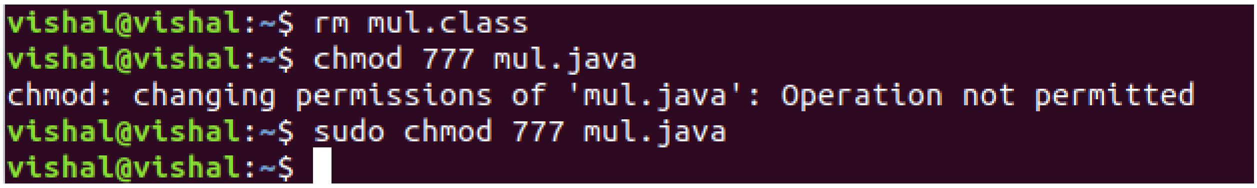
```
import java.io.*;
class mul
{
    // Driver code
    public static void main(String arg[])
    {
        // Change here to change output
        int n = 5;
        for (int i = 1; i <= 10; ++i)
            System.out.println(n + " * " + i +
                               " = " + n * i);
    }
}
```

[Read 14 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line



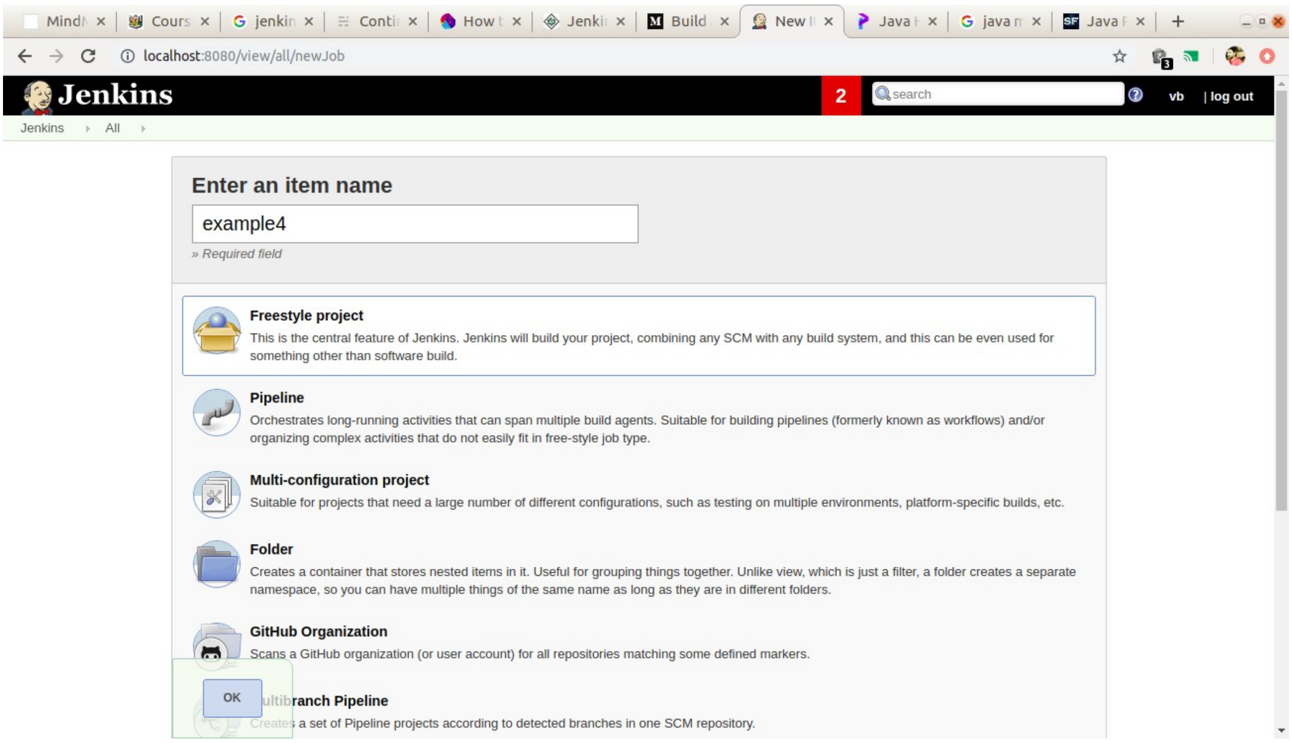
```
vishal@vishal:~$
vishal@vishal:~$ sudo nano mul.java
vishal@vishal:~$ javac mul.java
vishal@vishal:~$ java mul 5
Enter number:5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
vishal@vishal:~$
```

Delete class file and give write permission to program



```
vishal@vishal:~$ rm mul.class
vishal@vishal:~$ chmod 777 mul.java
chmod: changing permissions of 'mul.java': Operation not permitted
vishal@vishal:~$ sudo chmod 777 mul.java
vishal@vishal:~$
```

Now create new jenkins project “example4”



Enter an item name

example4

» 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.

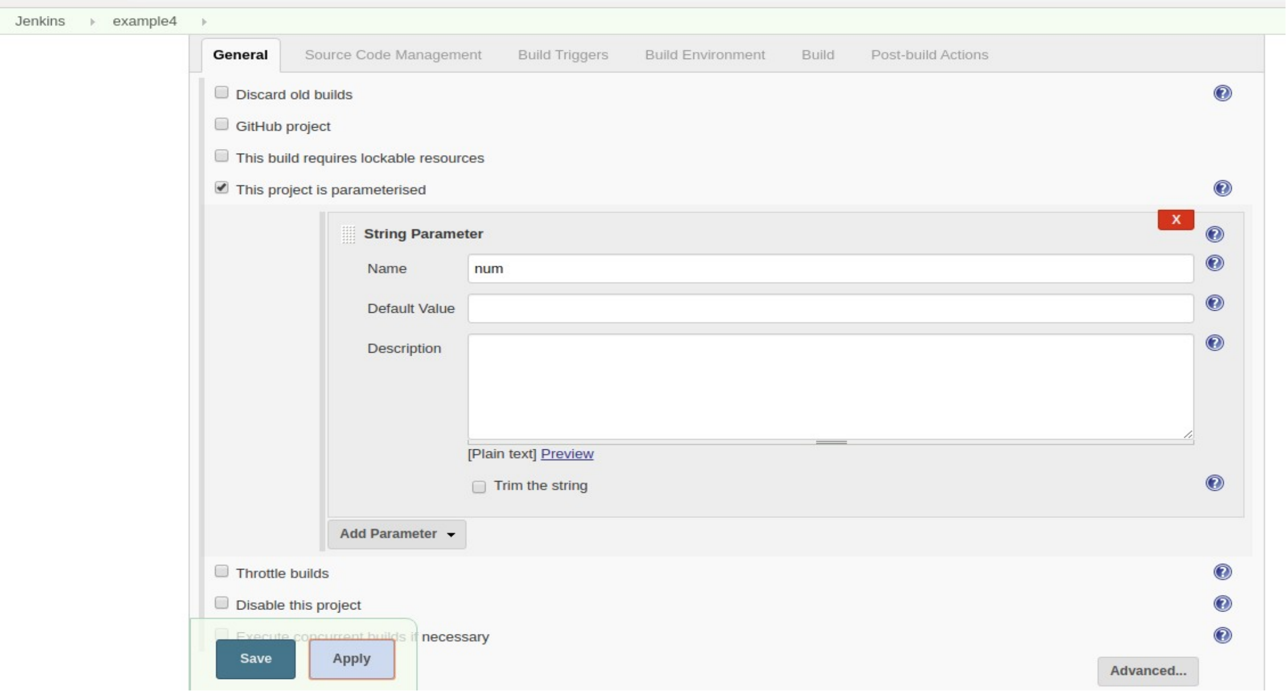
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Go to build option and select execute shell. Write the commands with changing path to directory



□ Discard old builds

□ GitHub project

□ This build requires lockable resources

☑ This project is parameterised

String Parameter

Name: num

Default Value:

Description:

[Plain text] Preview

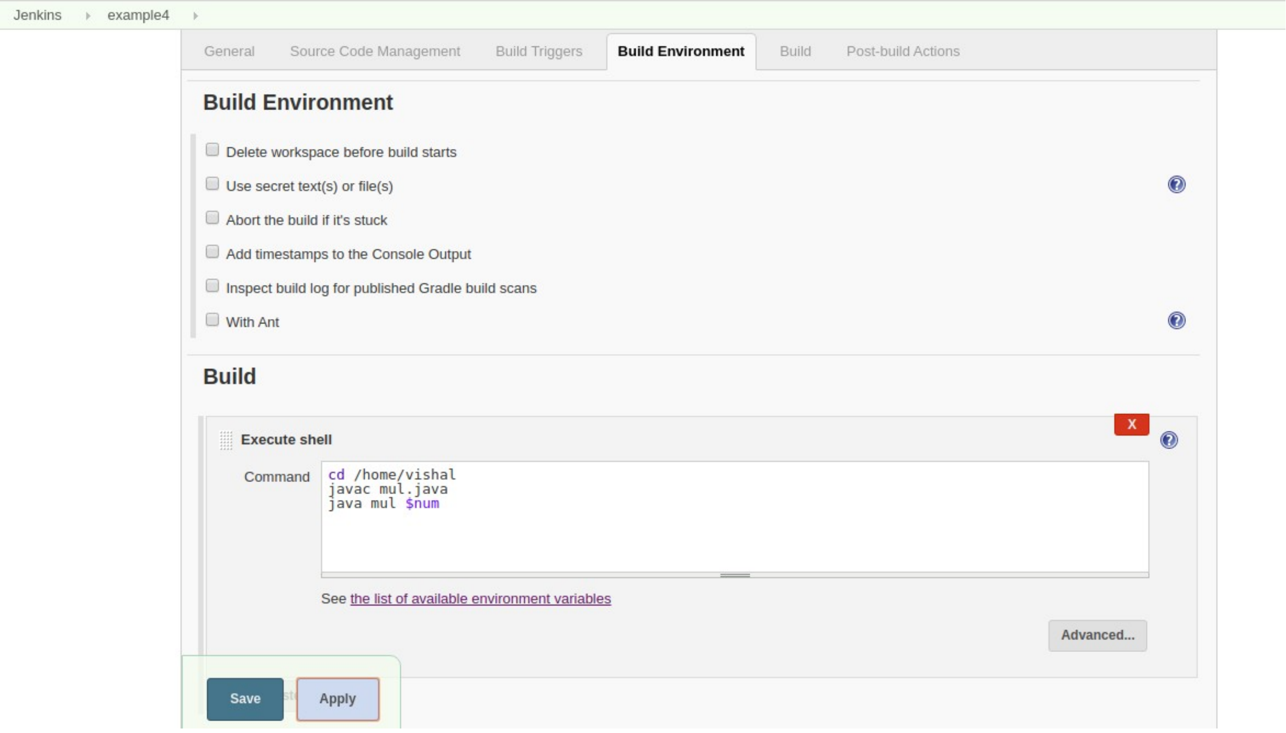
□ Trim the string

□ Throttle builds

□ Disable this project

□ Execute concurrent builds if necessary

where you have stored java program as below.



Build Environment

□ Delete workspace before build starts

□ Use secret text(s) or file(s)

□ Abort the build if it's stuck

□ Add timestamps to the Console Output

□ Inspect build log for published Gradle build scans

□ With Ant

Build

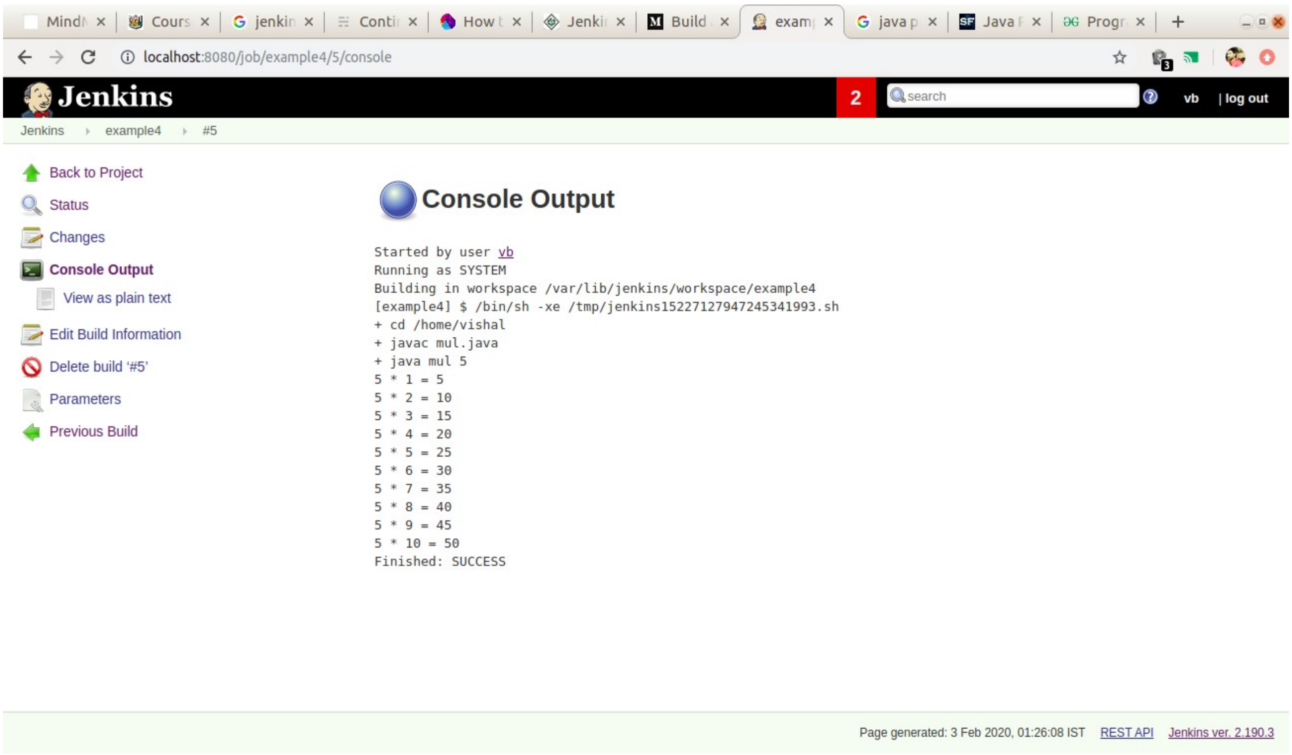
Execute shell

Command

```
cd /home/vishal
javac mul.java
java mul $num
```

See the list of available environment variables

Now click on Save. Select build with parameter option and specify value of "num" whose multiplication tables needs to be displayed.



```
Started by user vb
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example4
[example4] $ /bin/sh -xe /tmp/jenkins15227127947245341993.sh
+ cd /home/vishal
+ javac mul.java
+ java mul 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
Finished: SUCCESS
```

Task:

In same Way you can Execute any programming language program in Jenkins. Now you need to create one java with maven project and run any sample program under jenkins.