

Part 2:- To understand Jenkins Programming

“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.

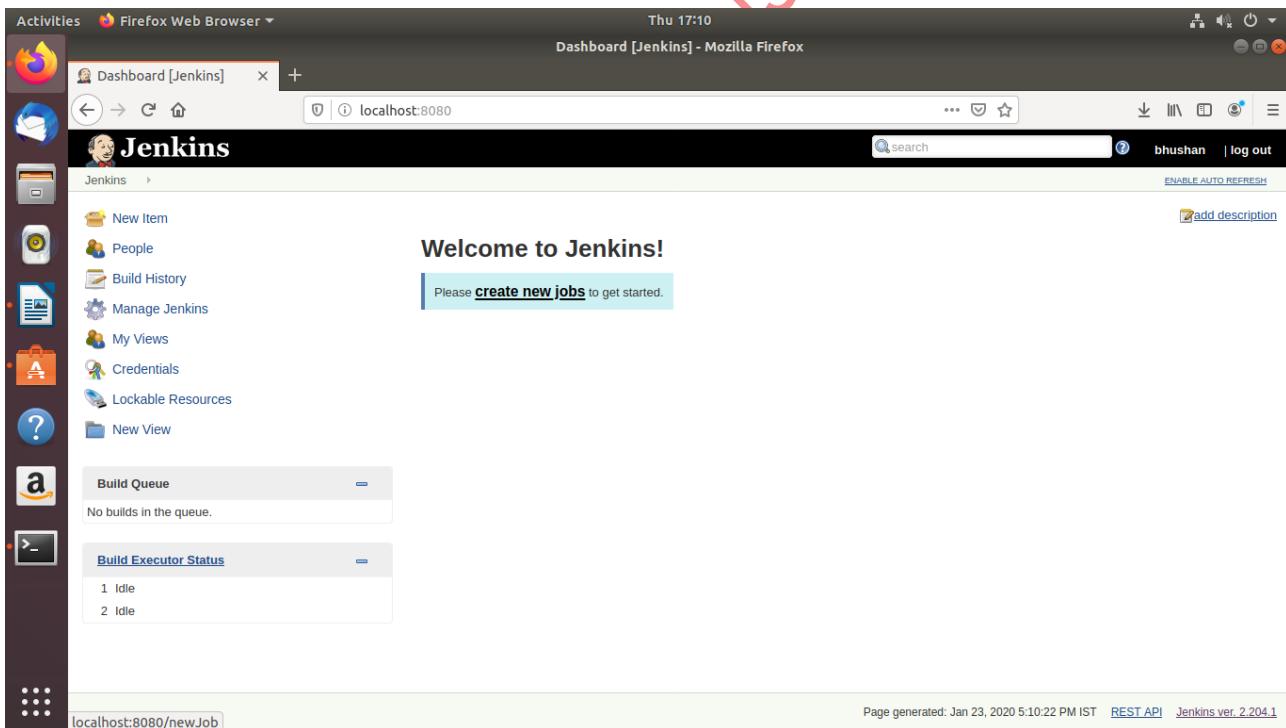
Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

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.



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

Activities Firefox Web Browser ▾

Thu 17:12

New Item [Jenkins] - Mozilla Firefox

Jenkins

Enter an item name

Example1

» 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
Manages 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

Activities Firefox Web Browser ▾

Thu 17:15

Example1 Config [Jenkins] - Mozilla Firefox

Jenkins > Example1 >

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

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

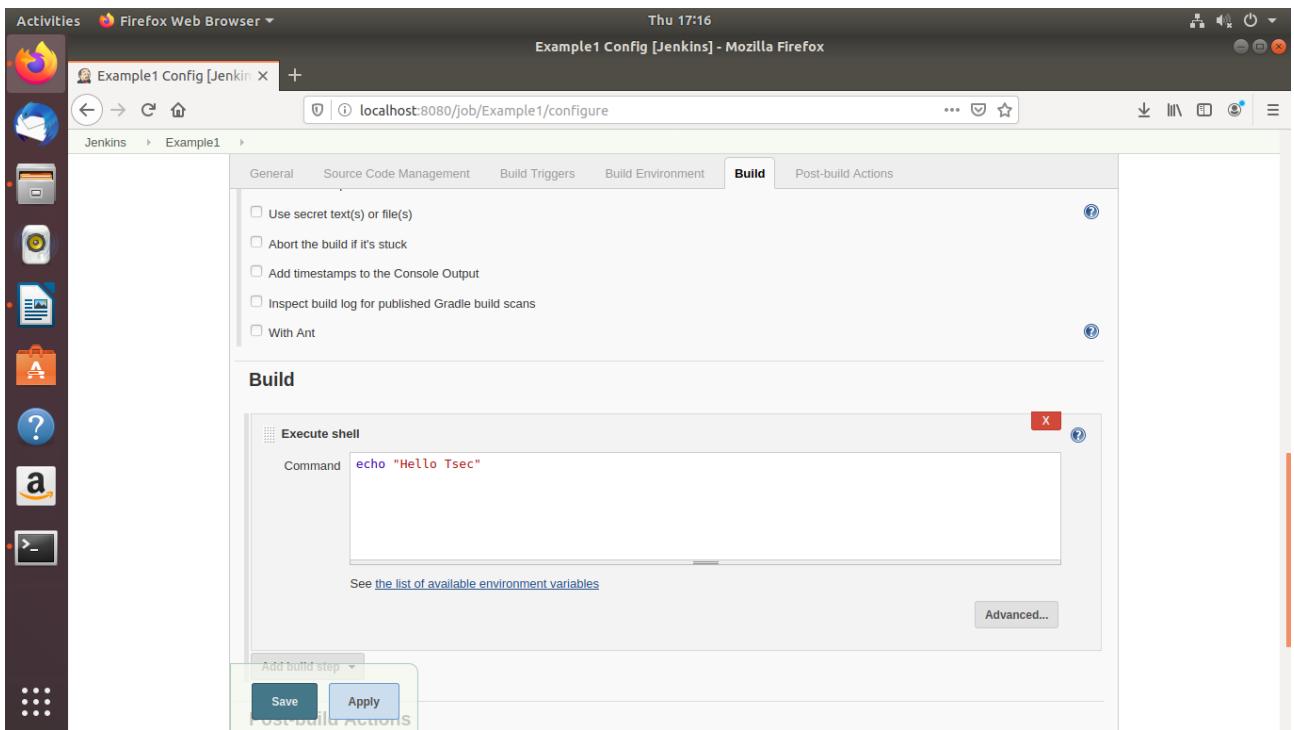
Build

Add build step ▾

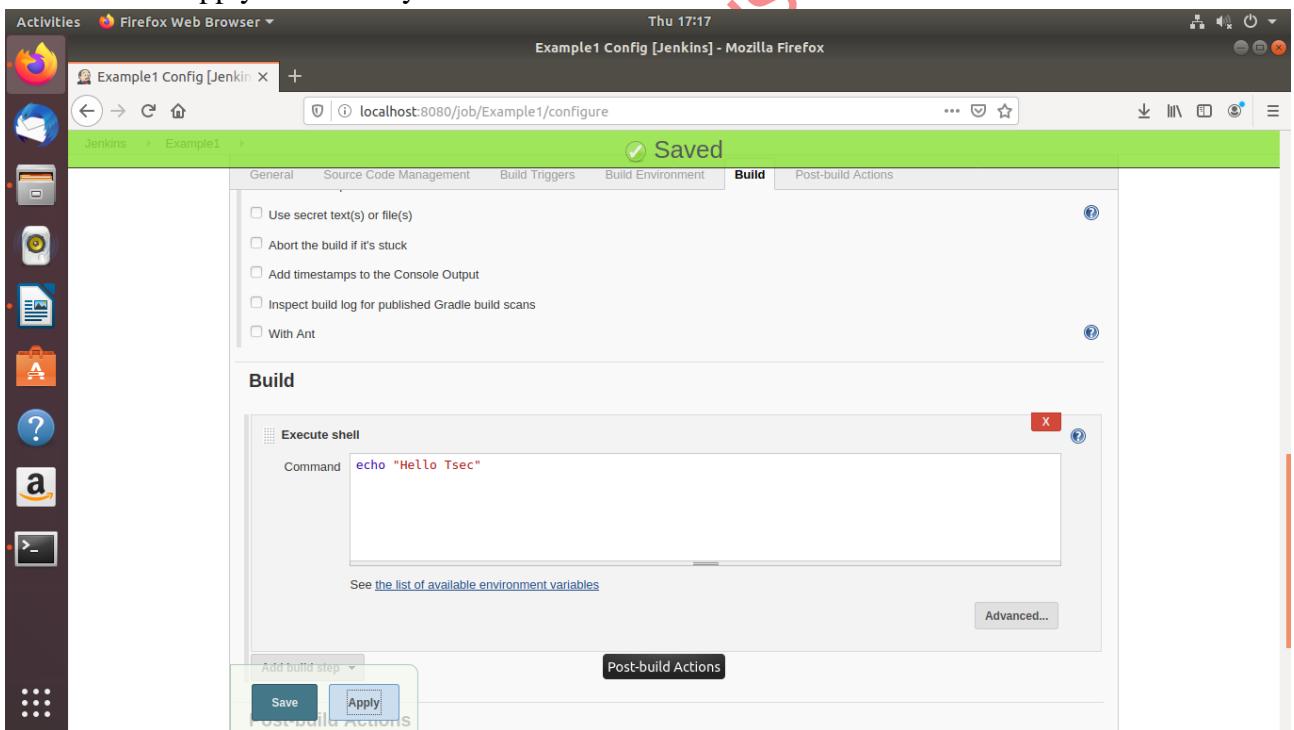
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Save Apply

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



Now click on apply followed by save button.



Step 5:- No Build a project to see the output Click on our first build “1” followed by console output to see the output

The screenshot shows the Jenkins Project Example1 dashboard. The top navigation bar displays the URL `localhost:8080/job/Example1/`. The main content area is titled "Project Example1". On the left sidebar, there are links for "Back to Dashboard", "Status", "Changes", "Workspace", "Build Now", "Delete Project", "Configure", and "Rename". A "Build History" section shows a single build (#1) from Jan 23, 2020, at 5:18 PM. Below the history are links for "Atom feed for all" and "Atom feed for failures". The bottom status bar shows the URL `localhost:8080/job/Example1/1/`, the page generation time "Page generated: Jan 23, 2020 5:18:02 PM IST", and the Jenkins version "Jenkins ver. 2.204.1".

The screenshot shows the Jenkins Console Output for build #1. The top navigation bar displays the URL `localhost:8080/job/Example1/1/console`. The main content area is titled "Console Output". The left sidebar includes a "Console Output" link with options "View as plain text" and "Edit Build Information". The console output itself shows the command `[Example1] $ /bin/sh -xe /tmp/jenkins4021943752300004763.sh` being run, followed by the text "Hello Tsec" and the message "Finished: SUCCESS". The bottom status bar shows the URL `localhost:8080/job/Example1/1/console`, the page generation time "Page generated: Jan 23, 2020 5:19:20 PM IST", and the Jenkins version "Jenkins ver. 2.204.1".

Click on our first build “1” followed by console output to see the output

Example 1.2:

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

A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for various applications like a browser, file manager, and system tools. The main window is a terminal window titled "Terminal". The terminal shows the following command and output:

```
Thu 17:24
admini@it906m11-HP-280-G2-MT:~$ cat > example1.sh
#!/bin/bash
name=$1
Address=$2
echo "Hello $name...your address is $Address"
^Z
[1]+  Stopped                  cat > example1.sh
admini@it906m11-HP-280-G2-MT:~$
```

A screenshot of a Linux desktop environment, similar to the one above. The terminal window shows the following command and output:

```
Thu 17:25
admini@it906m11-HP-280-G2-MT:~$ bash example1.sh
Hello ...your address is
admini@it906m11-HP-280-G2-MT:~$ bash example1.sh bhushan
Hello bhushan...your address is
admini@it906m11-HP-280-G2-MT:~$ bash example1.sh bhushan Dombivali
Hello bhushan...your address is Dombivali
admini@it906m11-HP-280-G2-MT:~$
```

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>

The image shows two screenshots of a Linux desktop environment with a dark theme. The top screenshot displays the Jenkins configuration page for a job named 'Example1'. The 'Build Environment' tab is selected, showing a single 'Execute shell' step with the command: `cd /home/admini
bash example1.sh`. Below the build steps is a 'Post-build Actions' section with 'Save' and 'Apply' buttons. The bottom screenshot shows the Jenkins console output for build #4, titled 'Console Output'. The log shows the command being run and the resulting output: 'Hello Bhushan...your address is Dombivali'. The Jenkins sidebar on the left includes links for Back to Project, Status, Changes, Console Output (which is currently selected), Edit Build Information, Delete build '#4', and Previous Build.

Thu 17:32 Example1 Config [Jenkins] - Mozilla Firefox

Activities Firefox Web Browser +

Jenkins > Example1 >

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

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/admini
bash example1.sh`

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

Advanced...

Add build step ▾

Post-build Actions

Save Apply

Thu 17:33 Example1 #4 Console [Jenkins] - Mozilla Firefox

Activities Firefox Web Browser +

Jenkins > Example1 > #4

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build '#4' Previous Build

Console Output

Started by user [bhushan](#)
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Example1
[Example1] \$ /bin/sh -xe /tmp/jenkins6020034662995079236.sh
+ cd /home/admini
+ bash example1.sh Bhushan Dombivali
Hello Bhushan...your address is Dombivali
Finished: SUCCESS

Page generated: Jan 23, 2020 5:33:39 PM IST REST API Jenkins ver. 2.204.1

1.3 Variation to this program

Activities Firefox Web Browser ▾

Thu 17:36

Example1 Config [Jenkins] - Mozilla Firefox

localhost:8080/job/Example1/configure

Jenkins Example1 + Saved

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

Build Environment

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/admini
name="Sachin"
Address="Dadar"
bash example1.sh $name $Address
```

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

Advanced...

Add build step Save Apply POST-build Actions

A red diagonal watermark 'Jshah' is visible across the entire screenshot.

Output

Activities Firefox Web Browser ▾

Thu 17:37

Example1 #7 Console [Jenkins] - Mozilla Firefox

localhost:8080/job/Example1/7/console

Jenkins Example1 #7

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build #7 Previous Build

Console Output

Started by user [bhushan](#)
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Example1
[Example1] \$ /bin/sh -xe /tmp/jenkins8060036160917618236.sh
+ cd /home/admini
+ name=Sachin
+ Address=Dadar
+ bash example1.sh Sachin Dadar
Hello Sachin...your address is Dadar
Finished: SUCCESS

Page generated: Jan 23, 2020 5:37:16 PM IST REST API Jenkins ver. 2.204.1

A red diagonal watermark 'Jshah' is visible across the entire screenshot.

Example 2

1.4-: Running a Java Program under jenkins

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

```
Thu 17:46
admini@it906m11-HP-280-G2-MT: ~$ nano example2.java
admini@it906m11-HP-280-G2-MT: ~$ javac example2.java
admini@it906m11-HP-280-G2-MT: ~$ java example2
Hello...Welcome to DevOps Class....
```

Now create freestyle project example2 in jenkins

New Item [Jenkins] - Mozilla Firefox

localhost:8080/view/all/newJob

Enter an item name

Example2

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

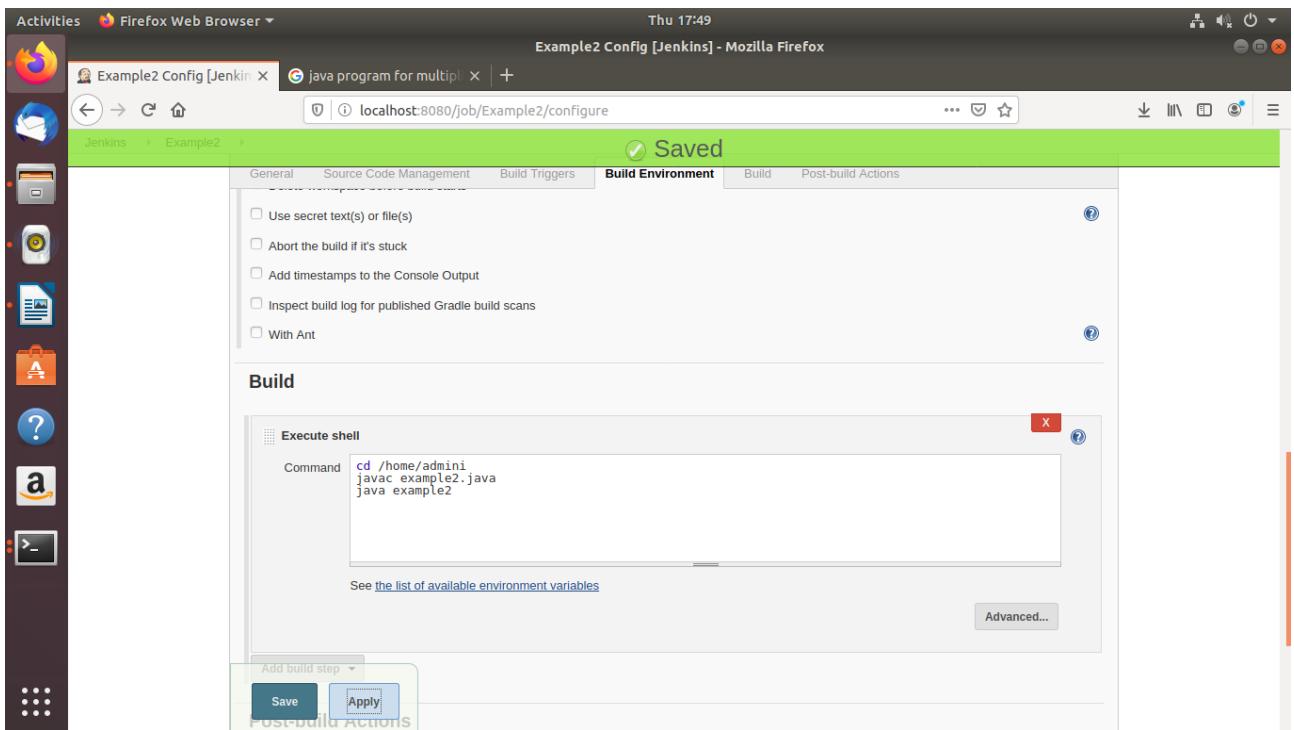
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization

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

Multibranch Pipeline

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



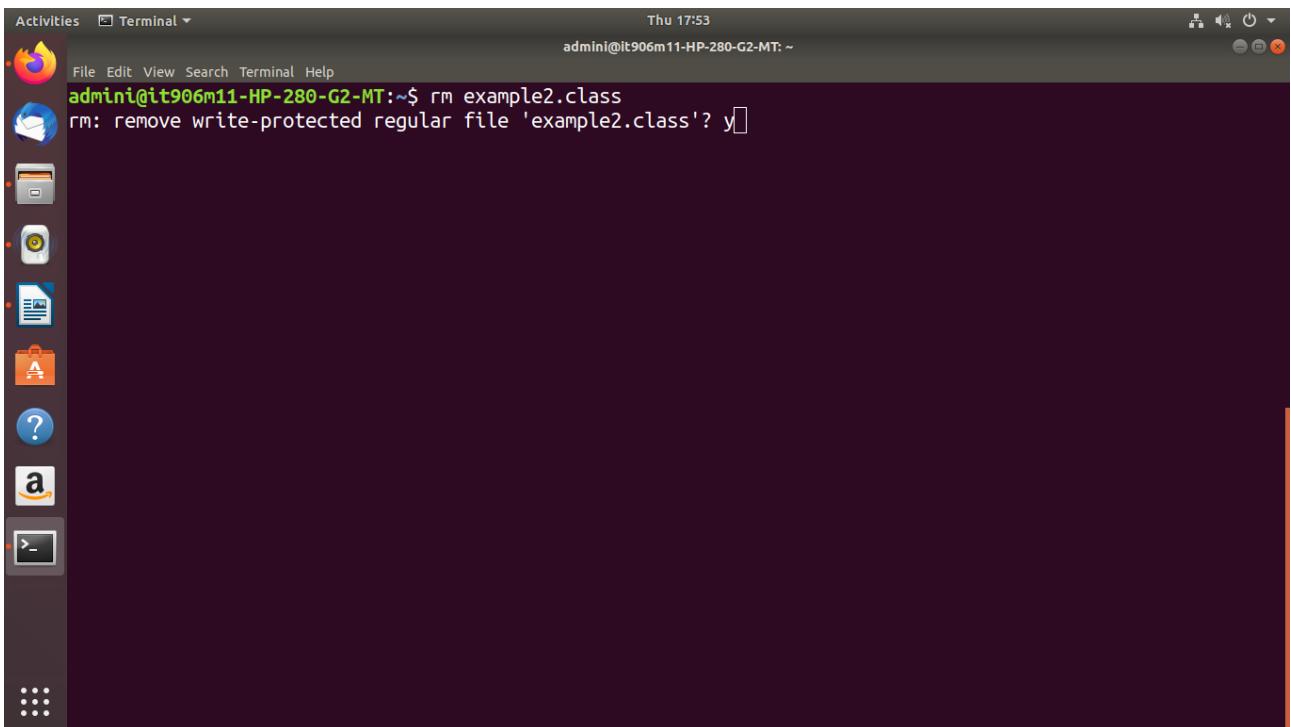
Now if your build fails due to permission problem then give write permission to directory underneath it.

```
Thu 17:50 Example2 #1 Console [Jenkins] - Mozilla Firefox Thu 17:52
admini@it906m11-HP-280-G2-MT:~$ cd ..
admini@it906m11-HP-280-G2-MT:/home$ chmod 777 admini/
admini@it906m11-HP-280-G2-MT:/home$
```

A screenshot of a terminal window titled 'Example2 #1 Console [Jenkins] - Mozilla Firefox'. The terminal prompt is 'admini@it906m11-HP-280-G2-MT:~\$'. The user runs the command 'cd ..' followed by 'chmod 777 admini/'. The terminal window has a dark background and light-colored text.

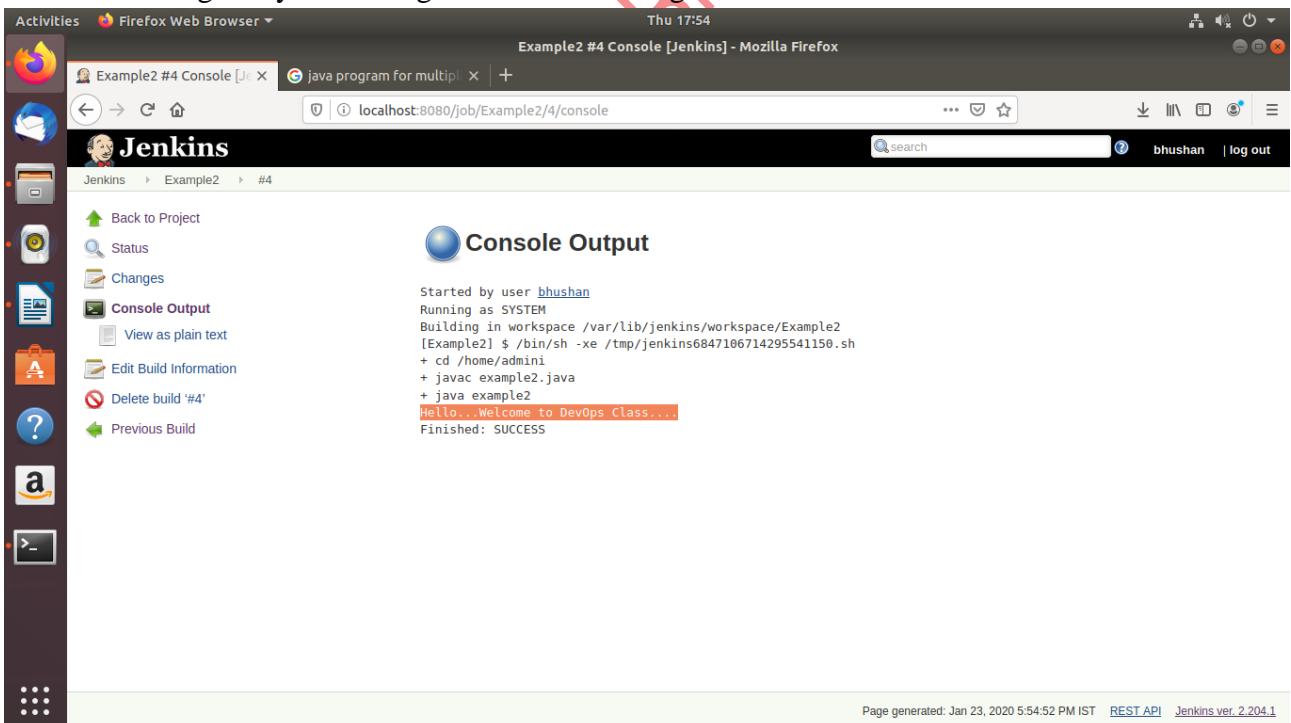
And remove existing .class file from your current directory

And remove existing .class file from your current directory



```
Thu 17:53
admin@it906m11-HP-280-G2-MT:~$ rm example2.class
rm: remove write-protected regular file 'example2.class'? y
```

Now after doing this your build gets successful and get results



Thu 17:54

Example2 #4 Console [Jenkins] - Mozilla Firefox

localhost:8080/job/Example2/4/console

Jenkins

Console Output

```
Started by user bhushan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Example2
[Example2] $ /bin/sh -xe /tmp/jenkins6847106714295541150.sh
+ cd /home/admini
+ javac example2.java
+ java example2
Hello...Welcome to DevOps Class....
Finished: SUCCESS
```

Page generated: Jan 23, 2020 5:54:52 PM IST REST API Jenkins ver. 2.204.1

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.

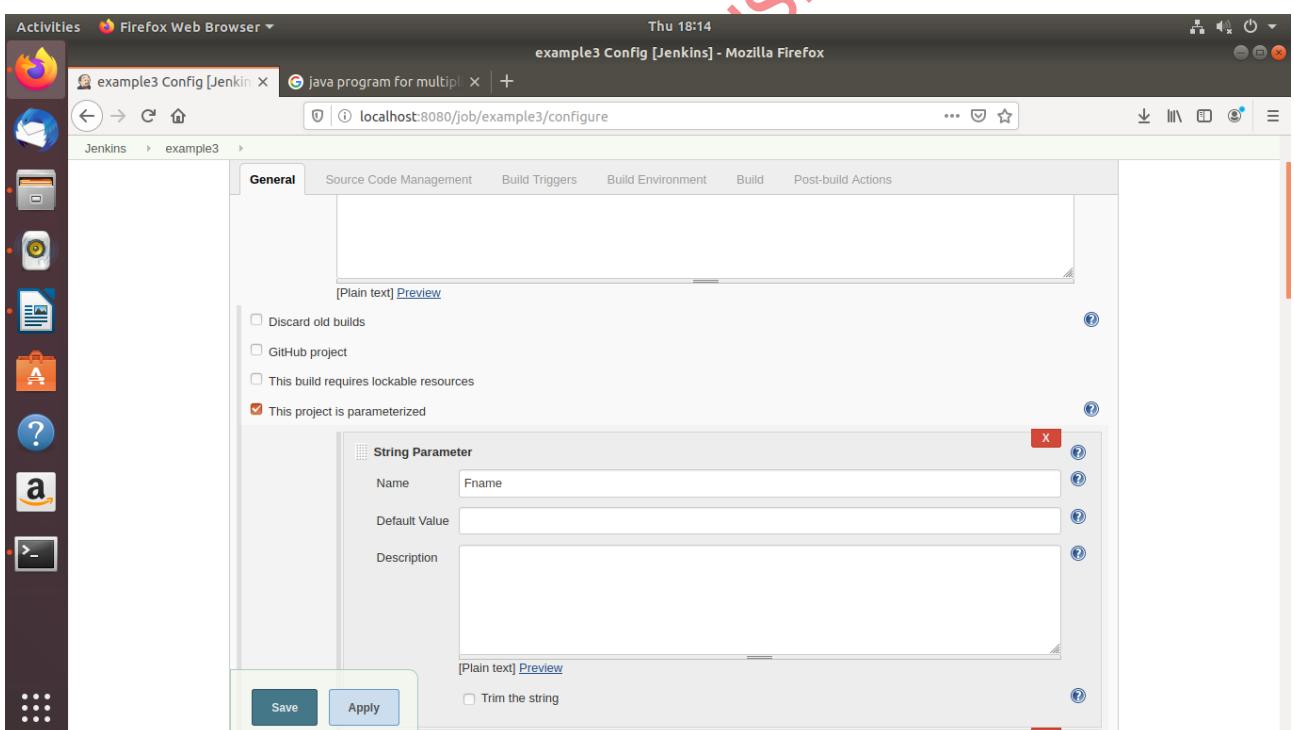
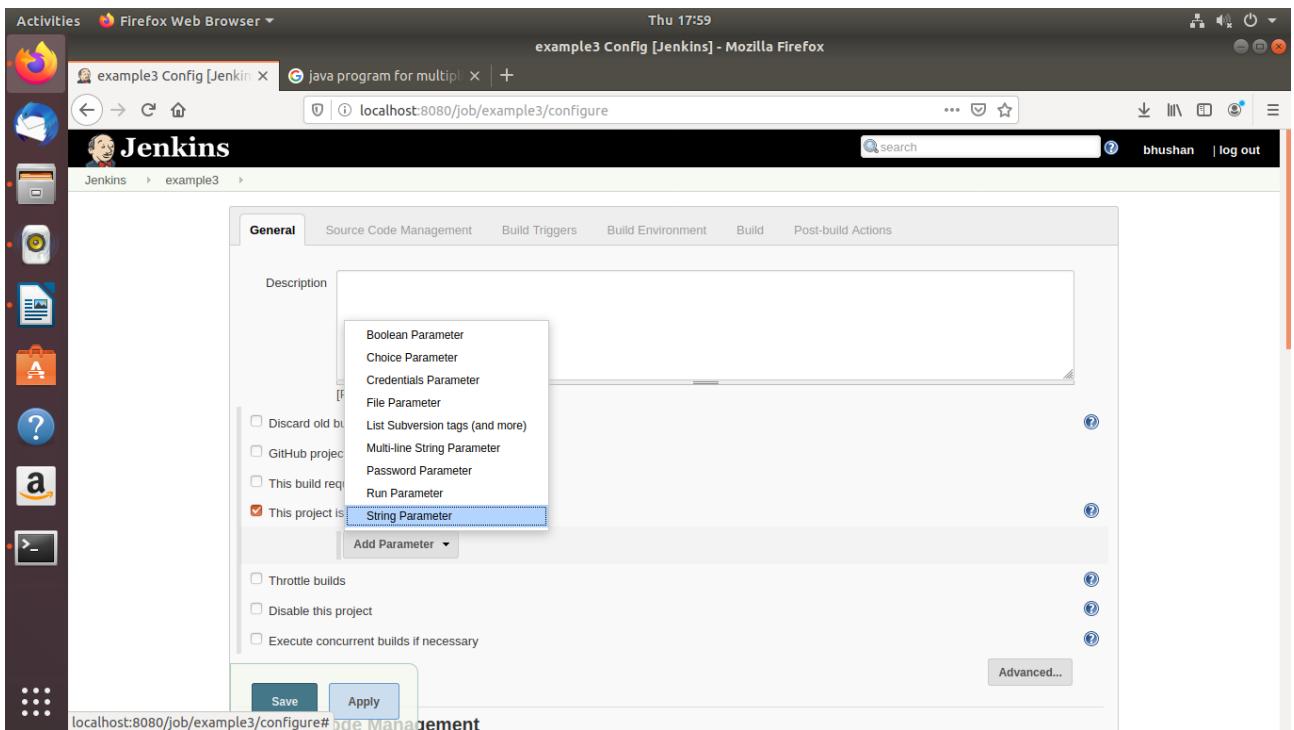
Step 1:- Create a free style project example3 by clicking on new item followed by specifying project name and free style project.

The screenshot shows the Jenkins interface for creating a new item. The 'Item Name' field is filled with 'example3'. The 'Freestyle project' option is selected, with a detailed description below it. Other project types like Pipeline, Multi-configuration project, and Folder are also listed. A large red watermark 'Bhushan' is visible across the center of the screen.

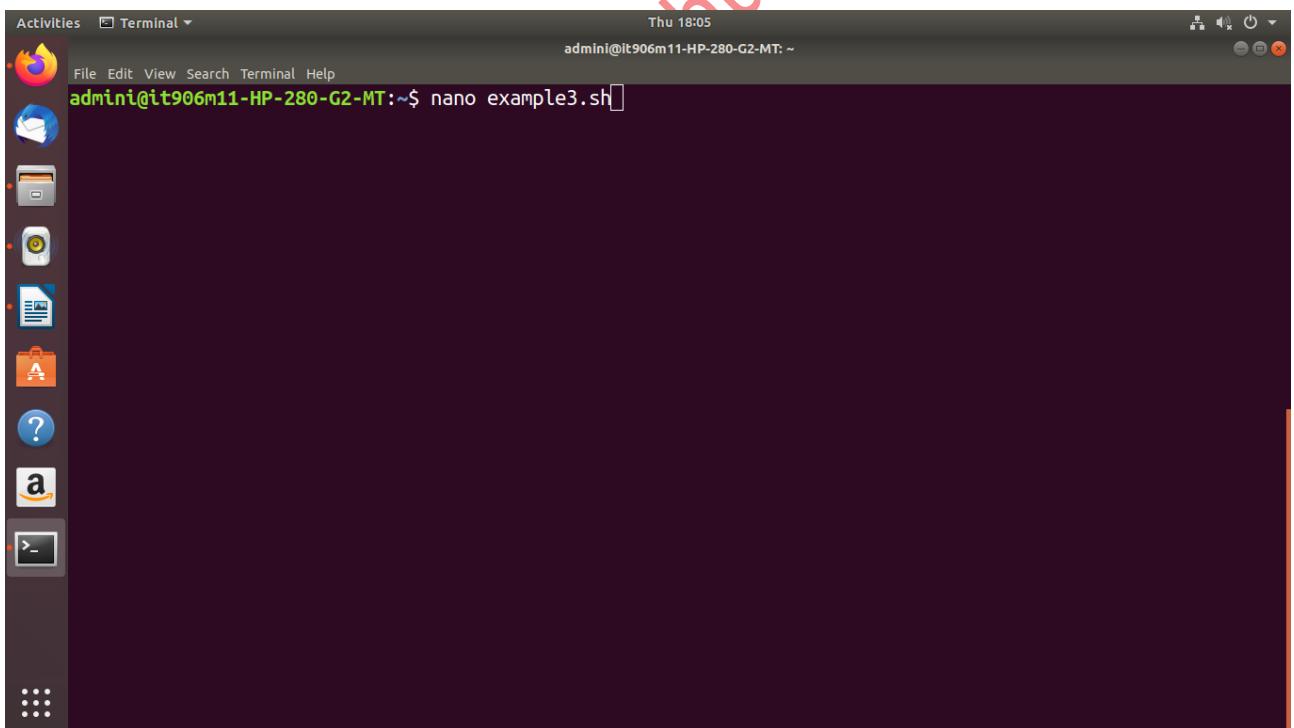
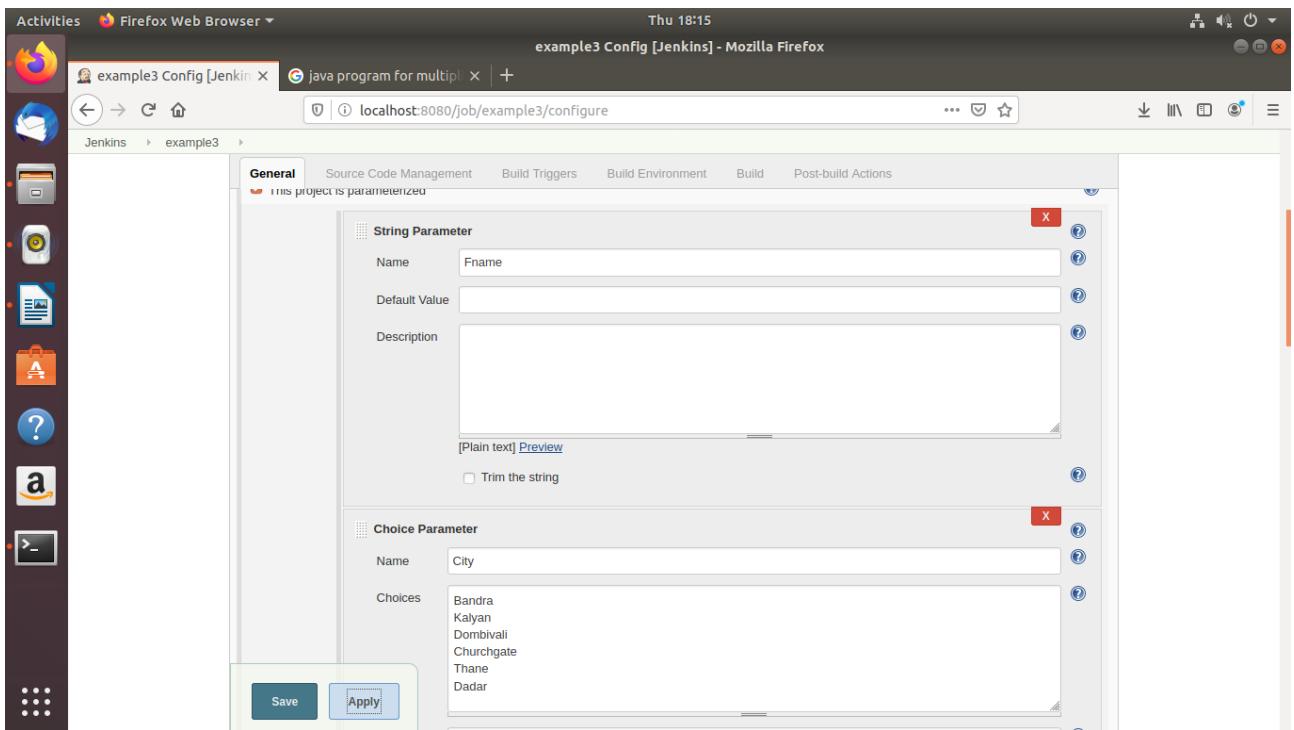
The screenshot shows the Jenkins configuration page for the 'example3' project. Under the 'General' tab, the 'This project is parameterized' checkbox is checked. Other configuration options like Discard old builds, GitHub project, and Throttle builds are available. At the bottom, there are 'Save' and 'Apply' buttons. A red watermark 'Bhushan' is visible across the center of the screen.

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

Select String parameter and specify name as “First-Name”



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



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

```
Thu 18:06  
admini@it906m11-HP-280-G2-MT:~$ nano example3.sh  
admini@it906m11-HP-280-G2-MT:~$ bash example3.sh  
Hiee Your name is and your city is  
admini@it906m11-HP-280-G2-MT:~$ bash example3.sh Bhushan  
Hiee Your name is Bhushan and your city is  
admini@it906m11-HP-280-G2-MT:~$ bash example3.sh Bhushan Dombivali  
Hiee Your name is Bhushan and your city is Dombivali  
admini@it906m11-HP-280-G2-MT:~$
```

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

Thu 18:17
example3 Config [Jenkins] - Mozilla Firefox

Jenkins > example3 > Saved

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

Execute shell

Command: cd /home/admini; bash example3.sh \$Fname \$City

Add build step

Post-build Actions

Add post-build action Save Apply

Now click on build with parameters and specify the values

Thu 18:10
Jenkins - Mozilla Firefox

Jenkins example3 > example3

Project example3

This build requires parameters:

First-Name

City

Build

Build History trend = find Atom feed for all Atom feed for failures

Page generated: Jan 23, 2020 6:10:05 PM IST Jenkins ver. 2.204.1

Click on Build

Thu 18:10
example3 [Jenkins] - Mozilla Firefox

Jenkins example3 > example3

Project example3

add description

Disable Project

Workspace

Recent Changes

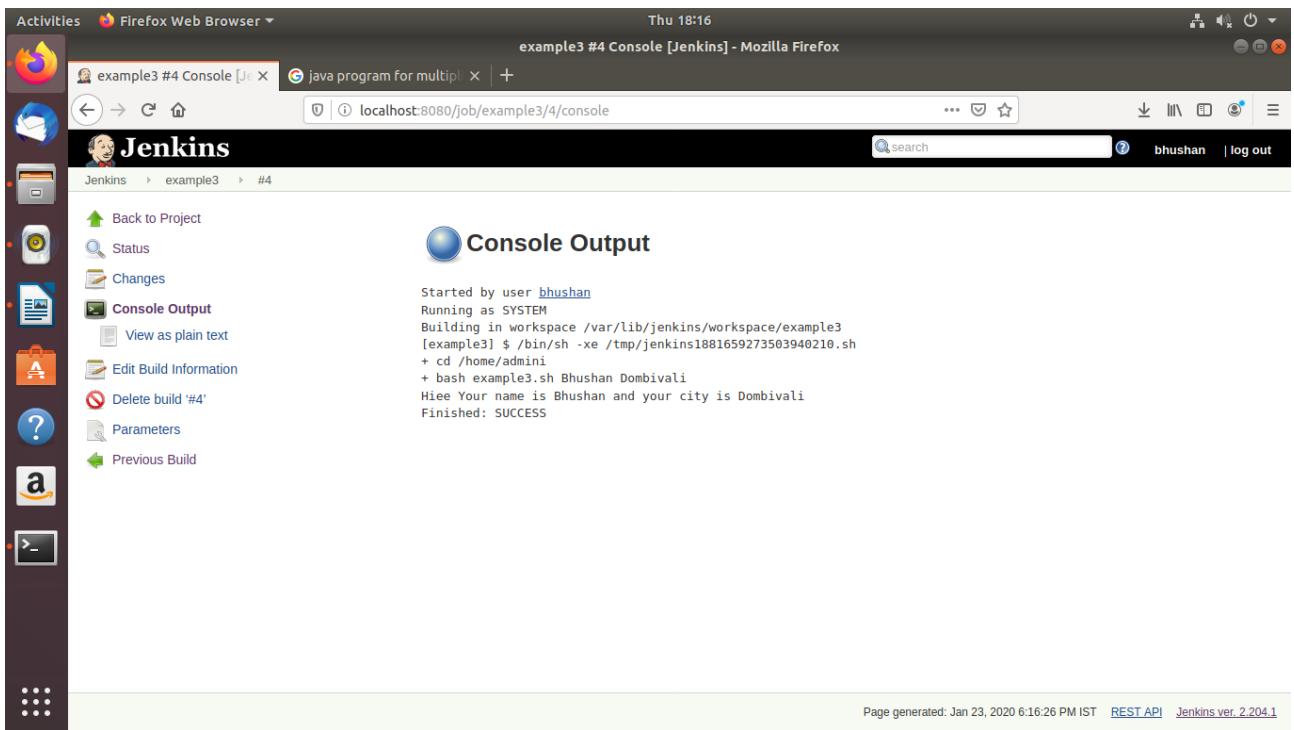
Permalinks

- Last build (#1), 71 ms ago

Build History trend = find #1 Jan 23, 2020 6:10 PM Atom feed for all Atom feed for failures

Page generated: Jan 23, 2020 6:10:56 PM IST REST API Jenkins ver. 2.204.1

Go to console to see the output



1.6 Running a Java program with parameters

The screenshot shows a terminal window titled 'Terminal'. The title bar indicates it's running on an 'HP-280-G2-MT' system at 18:24. The terminal window contains the following Java code:

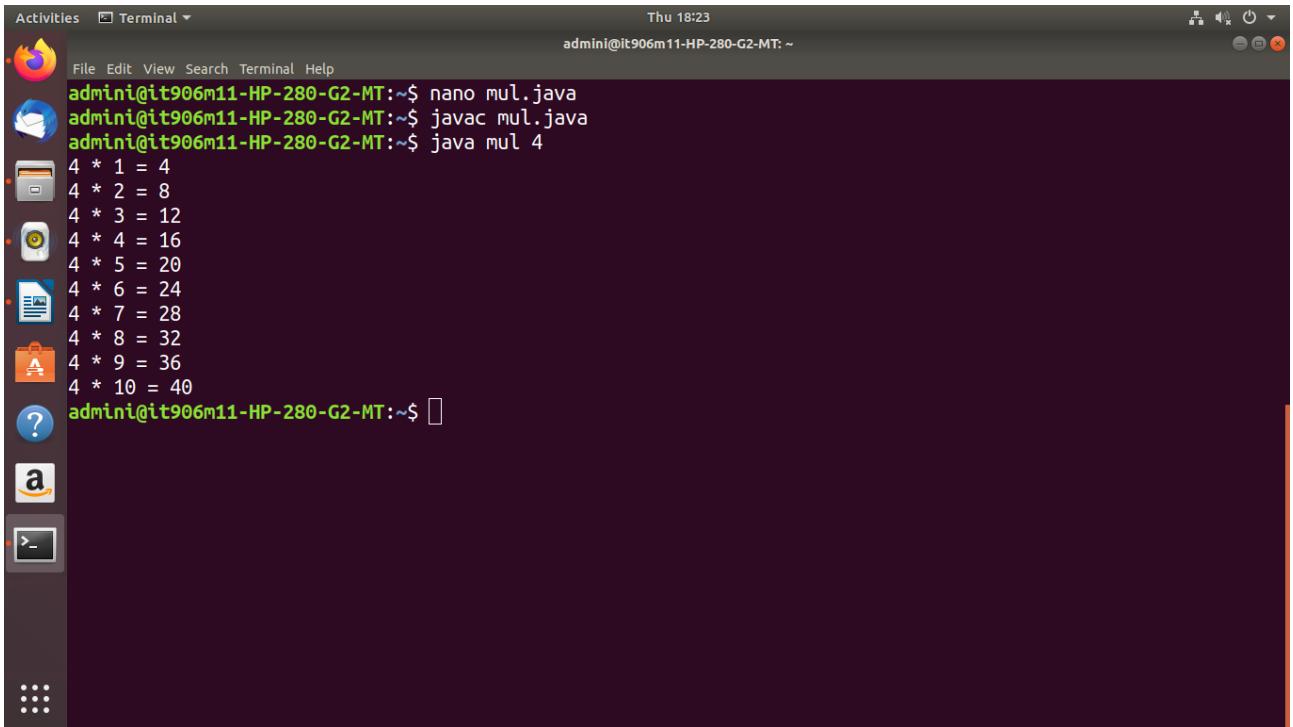
```
File Edit View Search Terminal Help
GNU nano 2.9.3
mul.java

class mul
{
    public static void main(String[] args)
    {
        int i;
        int num = Integer.parseInt(args[0]);
        for(i=1; i<=10; i++)
        {
            System.out.println(num+ " * " +i+ " = "+num * i);
        }
    }
}
```

The terminal window also shows a set of keyboard shortcuts at the bottom:

- Get Help (^G)
- Write Out (^O)
- Where Is (^W)
- Read File (^R)
- Replace (^X)
- Read 12 lines []
- Cut Text (^K)
- Uncut Text (^U)
- Justify (^J)
- To Spell (^T)
- Cur Pos (^C)
- Go To Line (^L)
- Undo (^U)
- Redo (^E)

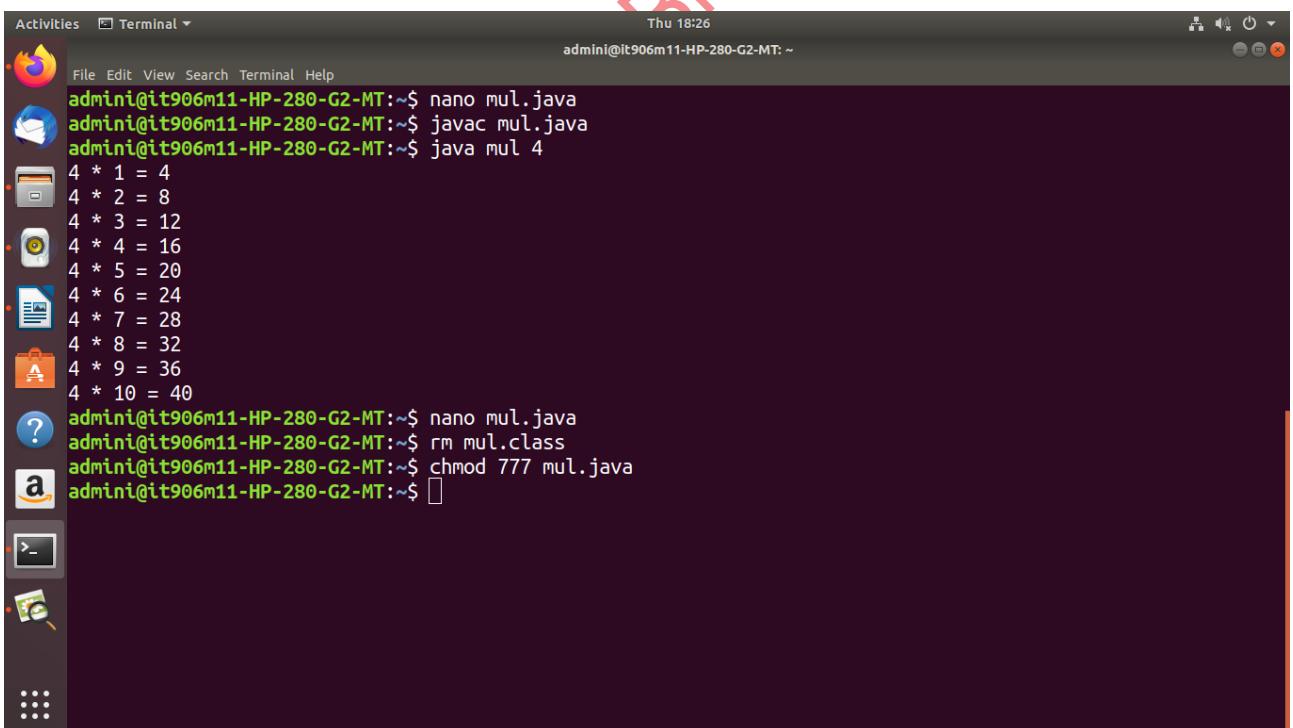
Write a java program for multiplication table with command line arguments and test it locally.



A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for various applications like a browser, file manager, and terminal. A red diagonal watermark "Bhushan" is overlaid across the screen. In the center is a terminal window titled "Terminal". The terminal shows the following command-line session:

```
Thu 18:23  
admini@it906m11-HP-280-G2-MT:~$ nano mul.java  
admini@it906m11-HP-280-G2-MT:~$ javac mul.java  
admini@it906m11-HP-280-G2-MT:~$ java mul 4  
4 * 1 = 4  
4 * 2 = 8  
4 * 3 = 12  
4 * 4 = 16  
4 * 5 = 20  
4 * 6 = 24  
4 * 7 = 28  
4 * 8 = 32  
4 * 9 = 36  
4 * 10 = 40  
admini@it906m11-HP-280-G2-MT:~$
```

Delete class file and give write permission to program

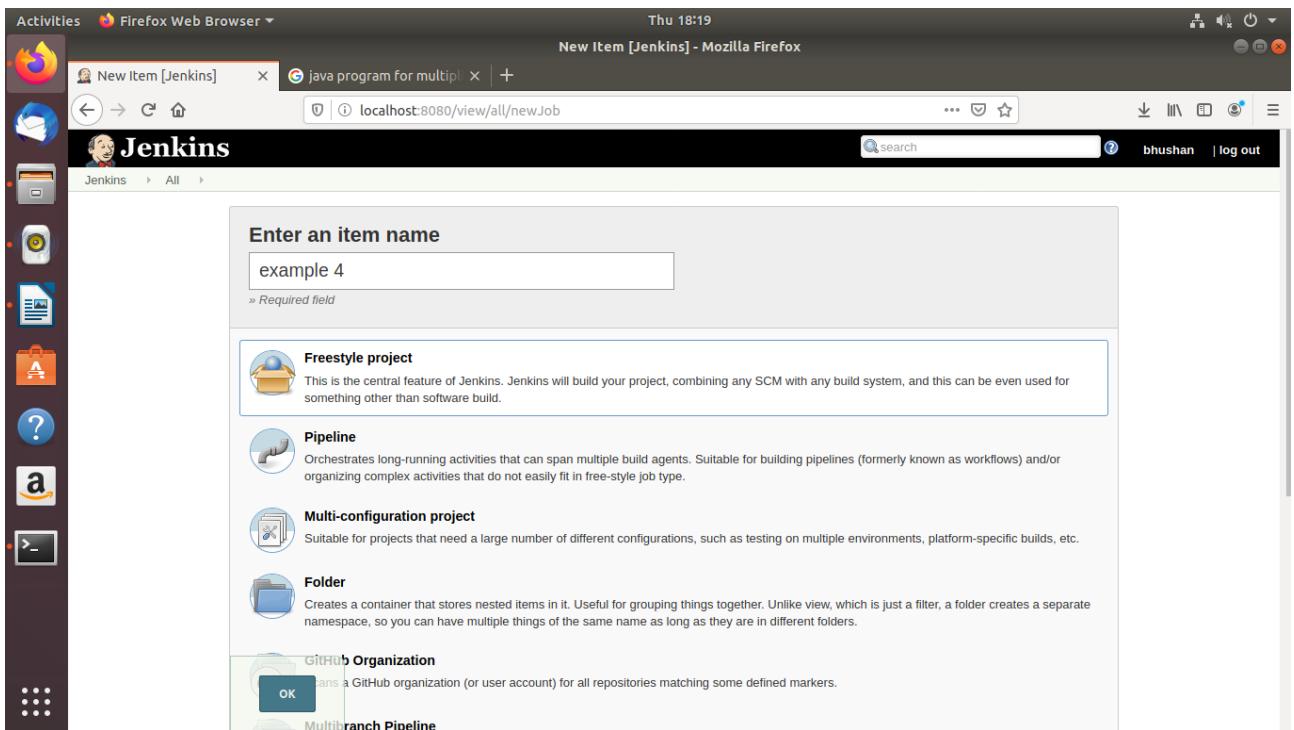


A screenshot of an Ubuntu desktop environment, similar to the one above. It shows a terminal window with the following command-line session:

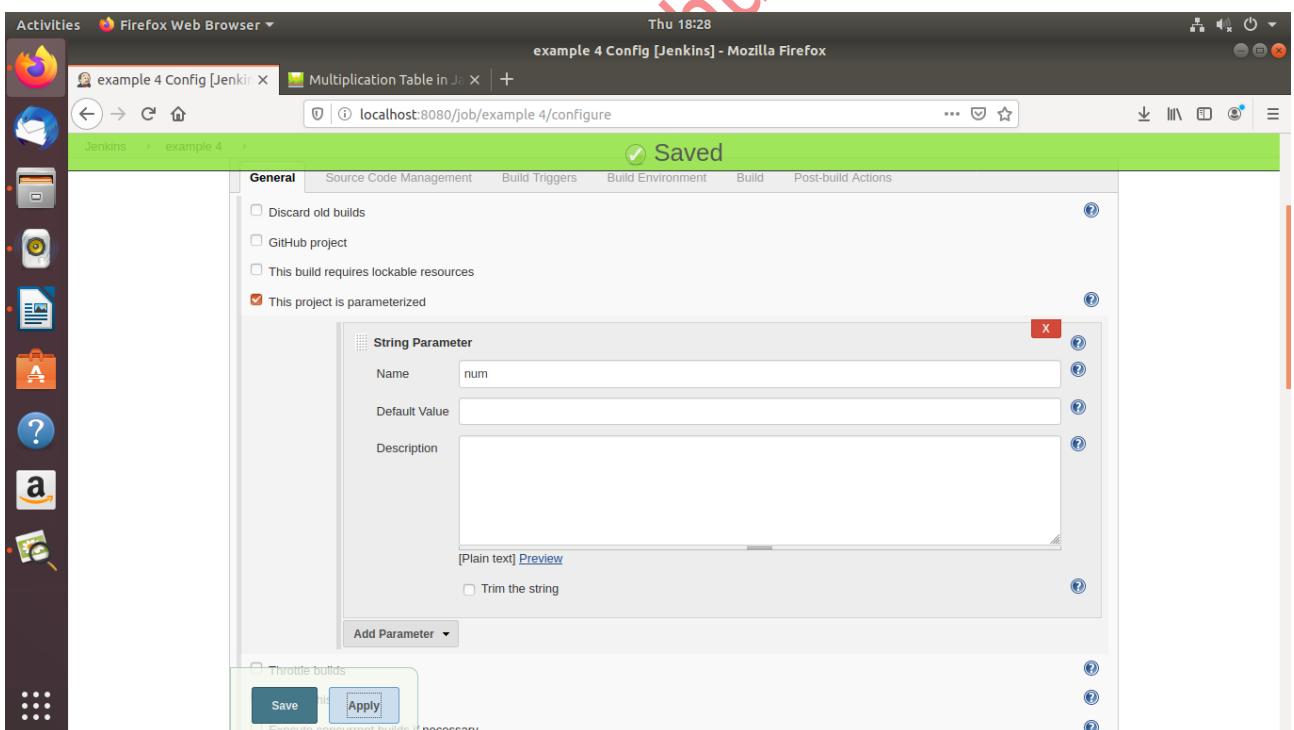
```
Thu 18:26  
admini@it906m11-HP-280-G2-MT:~$ nano mul.java  
admini@it906m11-HP-280-G2-MT:~$ javac mul.java  
admini@it906m11-HP-280-G2-MT:~$ java mul 4  
4 * 1 = 4  
4 * 2 = 8  
4 * 3 = 12  
4 * 4 = 16  
4 * 5 = 20  
4 * 6 = 24  
4 * 7 = 28  
4 * 8 = 32  
4 * 9 = 36  
4 * 10 = 40  
admini@it906m11-HP-280-G2-MT:~$ nano mul.java  
admini@it906m11-HP-280-G2-MT:~$ rm mul.class  
admini@it906m11-HP-280-G2-MT:~$ chmod 777 mul.java  
admini@it906m11-HP-280-G2-MT:~$
```

Free style project “Example 4” for running above program

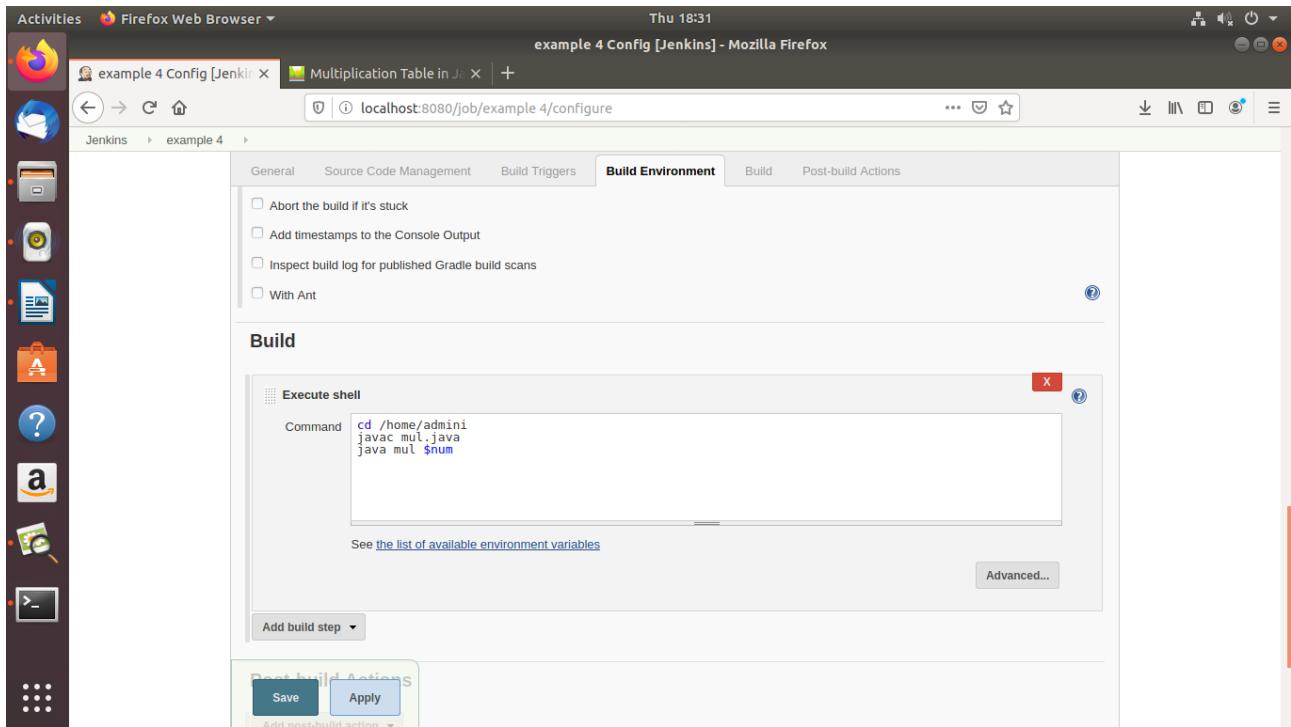
Now create new jenkins project “example4”



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



where you have stored java program as below.



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

A screenshot of a Firefox browser window titled "Jenkins - Mozilla Firefox". The URL is "localhost:8080/job/example 4/build?delay=0sec". The page shows the Jenkins logo and the title "Project example 4". It says "This build requires parameters:" and has a "num" input field with the value "25". There is a "Build" button. On the left, there is a sidebar with links like "Back to Dashboard", "Status", "Changes", "Workspace", "Build with Parameters" (which is highlighted), "Delete Project", "Configure", and "Rename". Below the sidebar, there is a "Build History" section with a table showing one build entry: "#1 Jan 23, 2020 6:29 PM". At the bottom right, it says "Page generated: Jan 23, 2020 6:29:50 PM IST Jenkins ver. 2.204.1".

Click on build to see the output as below

The screenshot shows a Linux desktop environment with a dark theme. A Firefox browser window is open, displaying the Jenkins console output for a build named 'example 4 #1'. The browser title bar says 'example 4 #1 Console [Jenkins] - Mozilla Firefox'. The Jenkins interface shows the 'Console Output' section with the following log:

```
Started by user bhushan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/example 4
[example 4] $ /bin/sh -xe /tmp/jenkins1621022023485294105.sh
+ cd /home/admini
+ javac mul.java
+ java mul 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
Finished: SUCCESS
```

The Jenkins sidebar on the left includes links for 'Back to Project', 'Status', 'Changes', 'Console Output' (which is selected), 'Edit Build Information', 'Delete build #1', and 'Parameters'. The bottom right of the browser window shows page details: 'Page generated: Jan 23, 2020 6:29:24 PM IST REST API Jenkins ver. 2.204.1'.

Example 5

1.7 Running a python program in jenkins

Step 1:- Write and test python3 program locally

The screenshot shows a terminal window with a dark background. The terminal title bar says 'Terminal'. The command line shows the user is in their home directory (~) and is running the nano editor to open a file named 'example4.py':

```
Thu 18:35
admini@it906m11-HP-280-G2-MT:~$ nano example4.py
```

Activities Terminal Thu 18:47
admini@it906m11-HP-280-G2-MT: ~

File Edit View Search Terminal Help
GNU nano 2.9.3 example4.py Modified

```
import sys
n1=sys.argv[1]
n2=int(n1)
print("Original number is",n2)
print("Binary representation of ", n2, " is",bin(n2))
print("Octal representation of ", n2, " is",oct(n2))
print("Hexadecimal representation of ", n1, " is",hex(n2))
print("Complex representation of ", n1, " is",complex(n2))
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo
Exit Read File Replace Uncut Text To Linter Go To Line Redo

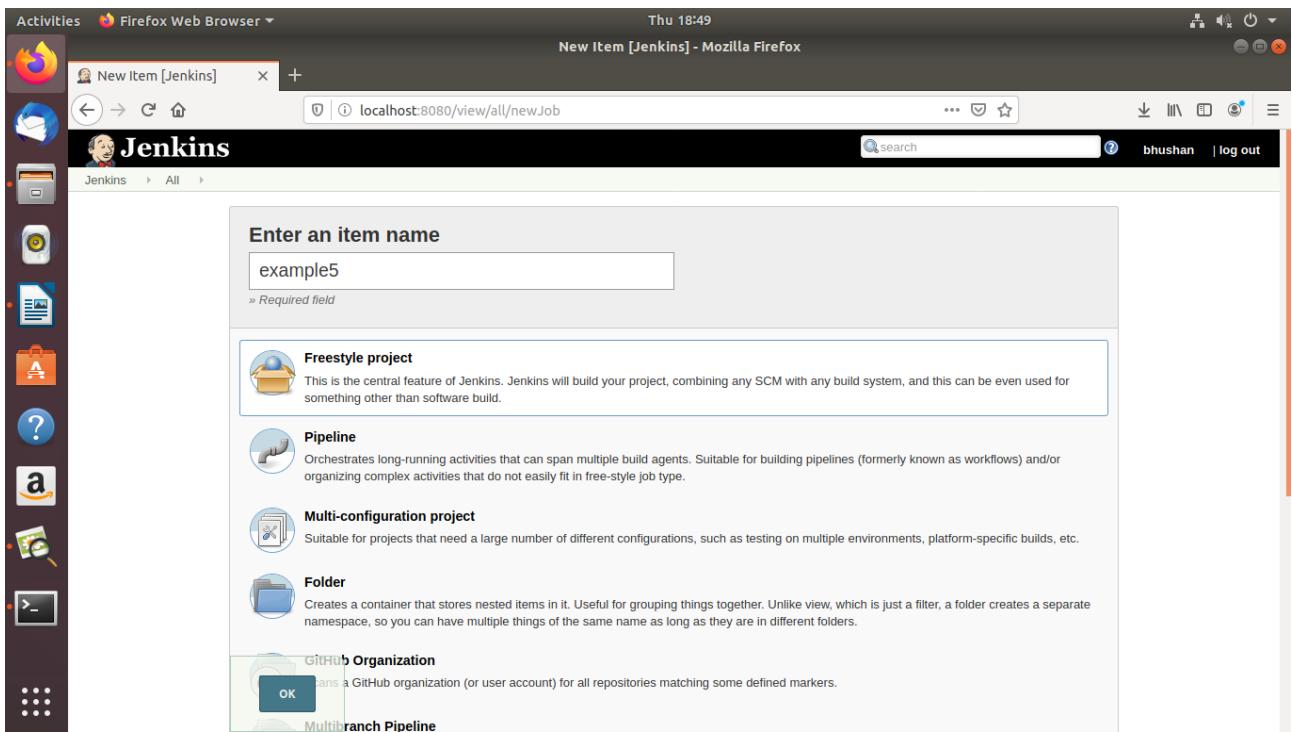
Activities Terminal Thu 18:48
admini@it906m11-HP-280-G2-MT: ~\$ python3 example4.py 10

```
Original number is 10
Binary representation of 10 is 0b1010
Octal representation of 10 is 0o12
Hexadecimal representation of 10 is 0xa
Complex representation of 10 is (10+0j)
```

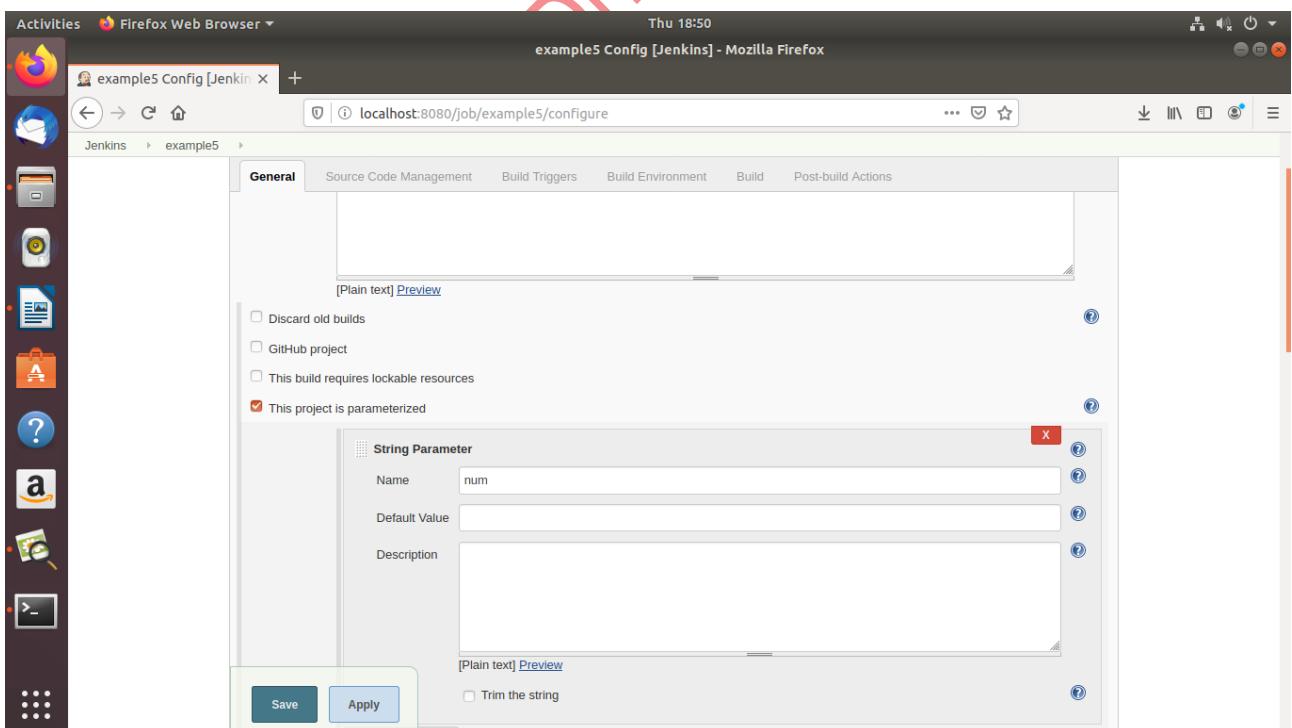
admini@it906m11-HP-280-G2-MT: ~\$

OUTPUT

Now create a new item followed by specifying name “example5” and select freestyle project

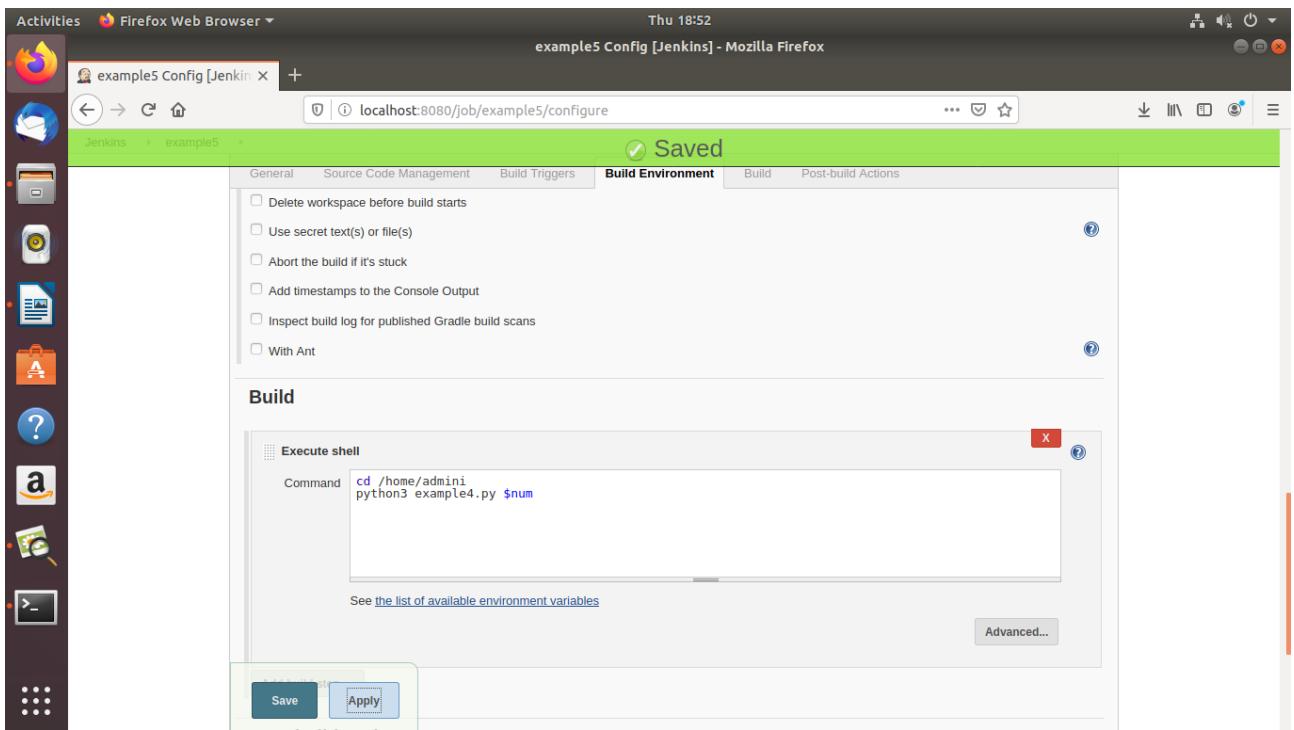


Now select, this project is parameterized, select string parameter and specify parameter name as “num”

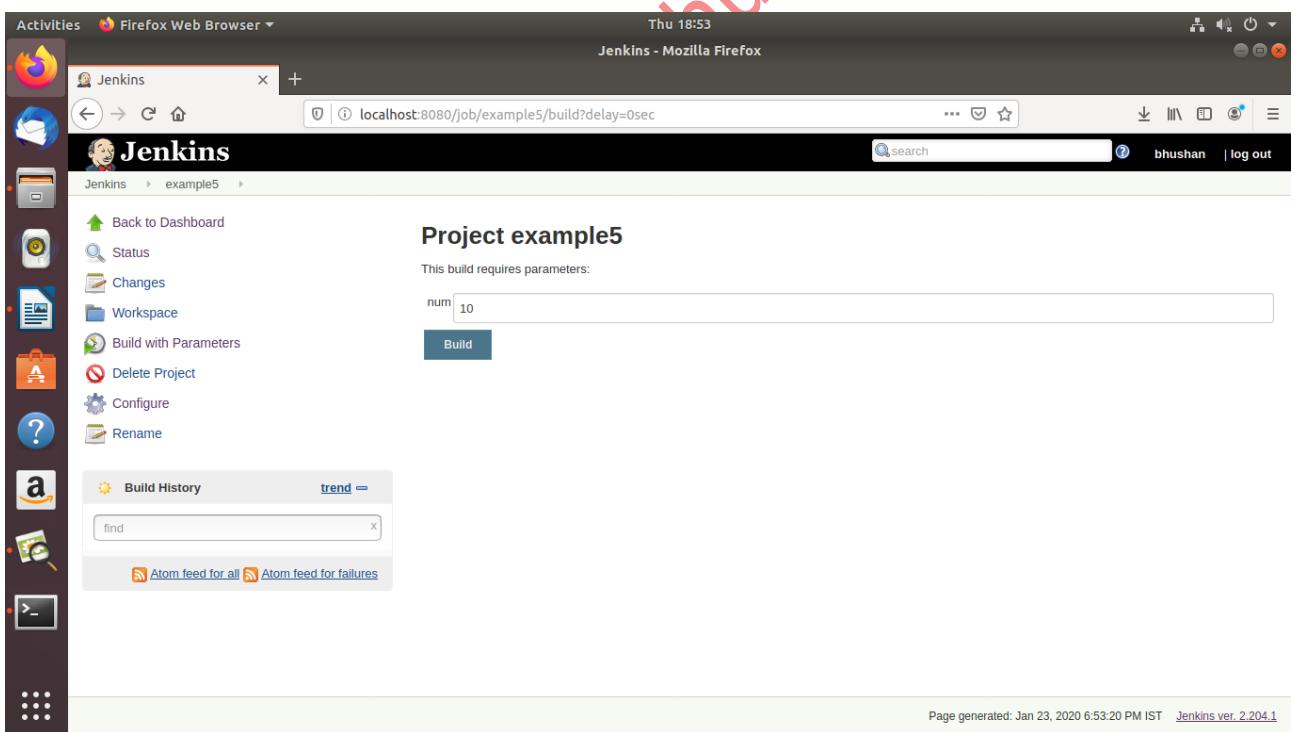


Go to build option and write script as follows

Compiled by **Dr. Bhushan Jadhav** Contact:- **9702868662** E-mail:- **bhushan.jadhav@tsec.edu**



Now, select build with parameter and specify num value in textbox, let's say 10.



Click on build followed by console output to see result.
The Output of program will be shown as below.

The screenshot shows a Linux desktop environment with a dock on the left containing various application icons. A Firefox browser window is open, displaying the Jenkins Project example5 page. The page title is "Project example5". On the left sidebar, there are links for Back to Dashboard, Status, Changes, Workspace, Build with Parameters, Delete Project, Configure, and Rename. The main content area shows a "Build History" section with one entry for build #1 from Jan 23, 2020, at 6:53 PM. The "Console Output" link is highlighted. Other links include "Changes", "Edit Build Information", and "Delete build '#1'". To the right, there are sections for "Workspace" (with a folder icon) and "Recent Changes" (with a document icon). Below these are "Permalinks" and a link to "Last build (#1), 48 ms ago". At the bottom right, there are links for "ENABLE AUTO REFRESH", "Disable Project", and "log out". The status bar at the bottom of the browser window shows the URL "localhost:8080/job/example5/1/console".

The Output of program will be shown as below.

The screenshot shows a Linux desktop environment with a dock on the left containing various application icons. A Firefox browser window is open, displaying the Jenkins Console Output for build #1. The title of the browser window is "example5 #1 Console [Jenkins] - Mozilla Firefox". The page title is "Console Output". On the left sidebar, there are links for Back to Project, Status, Changes, Console Output (which is selected and highlighted in blue), View as plain text, Edit Build Information, Delete build '#1', and Parameters. The main content area displays the console output of the build. It starts with "Started by user bhushan" and "Running as SYSTEM". It then shows the command "Building in workspace /var/lib/jenkins/workspace/example5" followed by a series of shell commands: "example5\$ /bin/sh -xe /tmp/jenkins4874911495360477757.sh", "+ cd /home/admini", "+ python3 example4.py 10", "Original number is 10", "Binary representation of 10 is 0b1010", "Octal representation of 10 is 0o12", "Hexadecimal representation of 10 is 0xa", and "Complex representation of 10 is (10+0j)". The output concludes with "Finished: SUCCESS". At the bottom right, there are links for "ENABLE AUTO REFRESH", "Disable Project", and "log out". The status bar at the bottom of the browser window shows the URL "localhost:8080/job/example5/1/console".

*****Thank You*****