

AIM: Installation of Jenkins in ubuntu operating system.

DESCRIPTION:

Prerequisites:

The required softwares –

1. Java
2. Git
3. Maven
4. jenkins

PROCEDURE:

Open a new terminal and run the following commands:

(1) Java:

\$sudo apt install open-jdk-11-jre-headless

```

ubuntu@ubuntu-VirtualBox: ~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install default-jre          # version 2:1.11-72build2, or
sudo apt install openjdk-11-jre-headless # version 11.0.15+10-0ubuntu0.22.04.1
sudo apt install openjdk-18-jre-headless # version 18-36ea-1
sudo apt install openjdk-8-jre-headless  # version 8u312-b07-0ubuntu1
sudo apt install openjdk-17-jre-headless # version 17.0.3+7-0ubuntu0.22.04.1
ubuntu@ubuntu-VirtualBox: ~$ sudo apt install openjdk-11-jre-headless
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
ca-certificates-java java-common
Suggested packages:
default-jre fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
ca-certificates-java java-common openjdk-11-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 187 not upgraded.
Need to get 41.5 MB/41.6 MB of archives.
After this operation, 171 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://security.ubuntu.com/ubuntu jammy-security/main amd64 openjdk-11-jr
e-headless amd64 11.0.16+8-0ubuntu1-22.04 [41.5 MB]
Fetched 41.5 MB in 34s (1,225 kB/s)
Selecting previously unselected package java-common.
(Reading database ... 159741 files and directories currently installed.)
Preparing to unpack .../java-common_0.72build2_all.deb ...

```

\$java -version

```

done.
done.
ubuntu@ubuntu-VirtualBox: ~$ java --version
openjdk 11.0.16 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu122.04, mixed mode
, sharing)
ubuntu@ubuntu-VirtualBox: ~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
git-man liberror-perl
Suggested packages:
git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 187 not upgraded.
Need to get 4,110 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.
17029-1 [20.5 kB]

```

(2) Git:

\$sudo apt install git

\$git -version

```

Activities Terminal Aug 9 19:26
ubuntu@ubuntu-VirtualBox: ~
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.4_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.4) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.4) ...
Setting up git (1:2.34.1-1ubuntu1.4) ...
Processing triggers for man-db (2.10.2-1) ...
ubuntu@ubuntu-VirtualBox: ~$ git --version
git version 2.34.1
ubuntu@ubuntu-VirtualBox: ~$ sudo apt install maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java
  libcdi-api-java libcommons-cli-java libcommons-io-java
  libcommons-lang3-java libcommons-parent-java
  libguava-java libguice-java libhawtjni-runtime-java libjansi-java
  libjansi-native-java libjsr305-java libmaven-parent-java
  libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java
  libplexus-cipher-java libplexus-classworlds-java
  libplexus-component-annotations-java libplexus-interpolation-java
  libplexus-sec-dispatcher-java libplexus-utils2-java libsisu-inject-java
  libsisu-plexus-java libslf4j-java libwagon-file-java
  libwagon-http-shaded-java libwagon-provider-api-java
Suggested packages:
  libaopalliance-java-doc libatinject-jsr330-api-java-doc libel-api-java
  libcommons-io-java-doc libcommons-lang3-java-doc libasm-java libcglib-java

```

(3) Maven:

\$sudo apt install maven

\$mvn -version

```

Activities Terminal Aug 9 19:26
ubuntu@ubuntu-VirtualBox: ~
Setting up libplexus-sec-dispatcher-java (1.4-4) ...
Setting up libcdi-api-java (1.2-3) ...
Setting up libhawtjni-runtime-java (1.17-1) ...
Setting up libwagon-provider-api-java (3.3.4-1) ...
Setting up libmaven-parent-java (3.1-2) ...
Setting up libcommons-parent-java (43-1) ...
Setting up libsisu-inject-java (0.3.4-2) ...
Setting up libsisu-plexus-java (0.3.4-3) ...
Setting up libmaven-resolver-java (1.4.2-3build1) ...
Setting up libcommons-lang3-java (3.11-1) ...
Setting up libjansi-native-java (1.8-1) ...
Setting up libwagon-file-java (3.3.4-1) ...
Setting up libcommons-io-java (2.11.0-2) ...
Setting up libjansi-java (1.18-1) ...
Setting up libmaven-shared-utils-java (3.3.0-1) ...
Setting up libmaven3-core-java (3.6.3-5) ...
Setting up maven (3.6.3-5) ...
update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
ubuntu@ubuntu-VirtualBox: ~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.10, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8

```

(4) Java Debian packages:

\$wget -q -O - http://pkg.jenkins.io/debian-rc/jenkins-ci.org.key | sudo apt-key add -

\$sudo sh -C 'echo deb <http://pkg.jenkins.io/debian-stable/binary/> > /etc/apt/sources.list.d/jenkins.list'

\$sudo apt update

```

Activities Terminal Aug 9 19:40
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox: ~$ wget -q -O - http://pkg.jenkins.io/debian-rc/jenkins-ci.org.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ubuntu-VirtualBox: ~$ sudo apt-get update
E: Invalid value set for option Signed-By regarding source https://pkg.jenkins.io/debian/binary/ (not a fingerprint)
E: The list of sources could not be read.
ubuntu@ubuntu-VirtualBox: ~$ sudo sh -C 'echo deb http://pkg.jenkins.io/debian-stable/binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ubuntu-VirtualBox: ~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:2 https://pkg.jenkins.io/debian-stable/binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable/binary/ Release [2,044 B]
Get:7 https://pkg.jenkins.io/debian-stable/binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian-stable/binary/ Packages [22.4 kB]
Fetched 25.3 kB in 14s (1,836 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
186 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://pkg.jenkins.io/debian-stable/binary/Release.gpg: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ubuntu-VirtualBox: ~$

```

(5) Jenkins:

\$sudo apt-get install jenkins

```

Reading state information... Done
186 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: http://pkg.jenkins.io/debian-stable/binary/Release.gpg: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 186 not upgraded.
Need to get 87.8 MB of archives.
After this operation, 92.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.346.2 [87.6 MB]
26% [2 jenkins 17.8 MB/87.6 MB 20%] 16.4 kB/s 1h 11min 9s
26% [2 jenkins 17.8 MB/87.6 MB 20%] 21.8 kB/s 53min 20s
26% [2 jenkins 17.9 MB/87.6 MB 20%] 21.8 kB/s 53min 18s
26% [2 jenkins 17.9 MB/87.6 MB 20%] 21.8 kB/s 53min 17s
27% [2 jenkins 18.2 MB/87.6 MB 21%] 21.8 kB/s 53min 4s
39% [2 jenkins 31.7 MB/87.6 MB 36%] 21.8 kB/s 42
39% [2 jenkins 31.9 MB/87.6 MB 36%] 21.8 kB/s 42min 36s^
39% [2 jenkins 31.9 MB/87.6 MB 36%] 19.1 kB/s 48min39s [
39% [2 jenkins 32.2 MB/87.6 MB 37%] 10.9 kB/s 1h 24min 46s
  
```

\$sudo systemctl status jenkins

```

Unpacking jenkins (2.346.2) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.346.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
ubuntu@ubuntu-VirtualBox:~$ sudo systemctl status jenkins
[sudo] password for ubuntu:
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor pres
   Active: active (running) since Tue 2022-08-09 20:06:35 IST; 37s ago
     Main PID: 6793 (java)
        Tasks: 42 (limit: 2292)
       Memory: 356.6M
          CPU: 47.366s
      CGroup: /system.slice/jenkins.service
              └─6793 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/jav
Aug 09 20:05:47 ubuntu-VirtualBox jenkins[6793]: This may also be found at: /v
Aug 09 20:05:47 ubuntu-VirtualBox jenkins[6793]: *****
Aug 09 20:05:47 ubuntu-VirtualBox jenkins[6793]: *****
Aug 09 20:05:47 ubuntu-VirtualBox jenkins[6793]: *****
Aug 09 20:06:33 ubuntu-VirtualBox jenkins[6793]: 2022-08-09 14:36:33.126+0000
Aug 09 20:06:33 ubuntu-VirtualBox jenkins[6793]: 2022-08-09 14:36:33.127+0000
Aug 09 20:06:33 ubuntu-VirtualBox jenkins[6793]: 2022-08-09 14:36:33.129+0000
Aug 09 20:06:35 ubuntu-VirtualBox jenkins[6793]: 2022-08-09 14:36:35.313+0000
Aug 09 20:06:35 ubuntu-VirtualBox jenkins[6793]: 2022-08-09 14:36:35.340+0000
Aug 09 20:06:35 ubuntu-VirtualBox systemd[1]: Started Jenkins Continuous Integ
lines 1-20/20 (END)
  
```

To start Jenkins:

\$sudo systemctl start jenkins

To restart Jenkins:

\$sudo systemctl restart jenkins

Access Jenkins Dashboard at Localhost:

(1) Enter url: <http://localhost:8080>

(2) Get default password at

\$sudo cat /var/lib/jenkins/secrets/initialAdminPassword

(3) Set your username and password.

(4) Then install required plugins.

RESULT: The installation of Jenkins in ubuntu operating system was completed successfully.

AIM: To create a Helloworld job in Jenkins.

DESCRIPTION:

First create a shell script named helloworld.sh

Which contains the following code:

```
#!/bin/bash
```

```
#echo "helloworld"
```

Then push this file to github and access the same using Jenkins dashboard and execute the build.

PROCEDURE:

- (1) Open terminal and execute the following commands:

```
$mkdir jenkinscripts
```

```
$cd jenkinscripts/
```

```
$gedit first.sh
```

```
#!/bin/bash
```

```
echo "helloworld"
```

```
$chmod +x first.sh
```

- (2) Start Jenkins using command prompt:

```
$sudo systemctl start Jenkins
```

⇒ To open the Jenkins dashboard using localhost:8080

- (3) Upload the folder to github using commands.

- (4) In Jenkins dashboard, Click on "New Item".

- (5) Enter an item name, and select "Free style project" and then click on "ok".

- (6) The next window appears with six tabs namely: General, Source code management, Build triggers, Build environment, Build, Post-Build actions.

- (7) We have to configure those tabs based on our requirements.

- (8) For this Project:

- (a) In general tab, click on "Advanced" and select Display Name, then type the display name as "helloworld.job".

- (b) In source code management, Click on git and give the github repository url.

- (c) Go to build, click on "add build step" and select "Execute Shell", give the following commands in the command shell:

```
cd jenkinscripts/
```

```
./first.sh
```

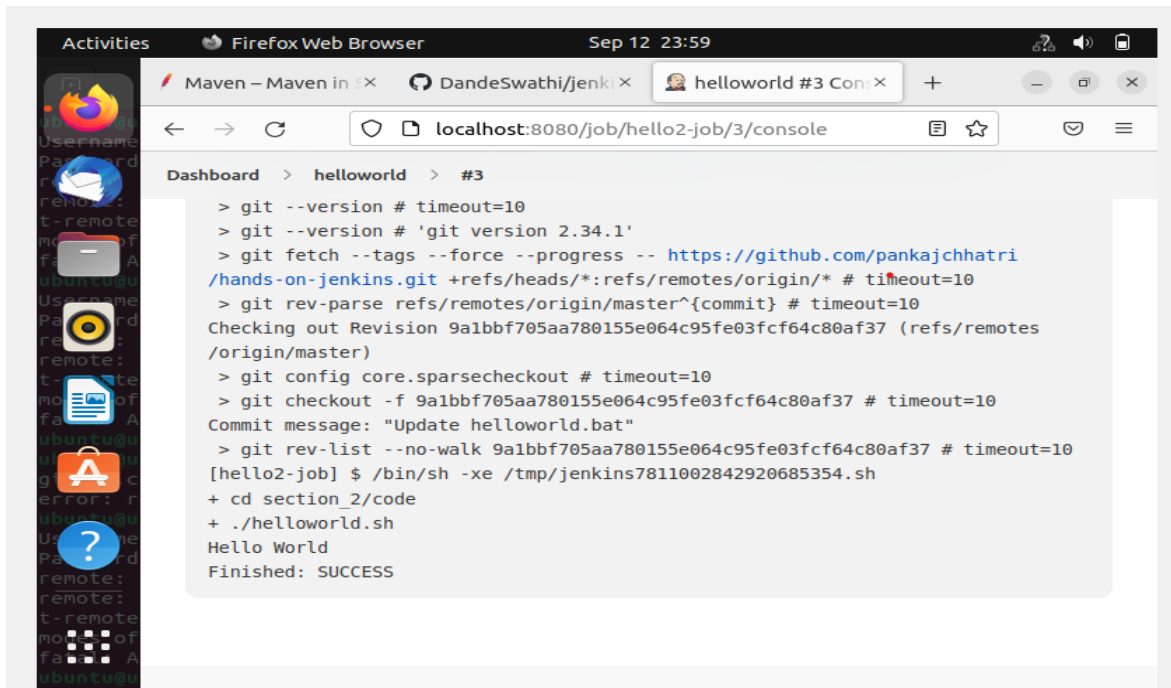
- (9) Click on apply and save buttons.

To run the Jenkins job:

- (i) Double click on helloworld.job, on dashboard.
- (ii) Click on "Build Now".

- (iii) Go to console output.
- (iv) In the console output, the shell script output will be displayed and the success message will be displayed.

OUTPUT:



The screenshot shows a Firefox Web Browser window with the address bar displaying 'localhost:8080/job/hello2-job/3/console'. The browser tabs include 'Maven - Maven in...', 'DandeSwathi/jenki...', and 'helloworld #3 Con...'. The console output is as follows:

```

Dashboard > helloworld > #3
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/pankajchhatri/hands-on-jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 9a1bbf705aa780155e064c95fe03fcf64c80af37 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9a1bbf705aa780155e064c95fe03fcf64c80af37 # timeout=10
Commit message: "Update helloworld.bat"
> git rev-list --no-walk 9a1bbf705aa780155e064c95fe03fcf64c80af37 # timeout=10
[hello2-job] $ /bin/sh -xe /tmp/jenkins7811002842920685354.sh
+ cd section_2/code
+ ./helloworld.sh
Hello World
Finished: SUCCESS

```

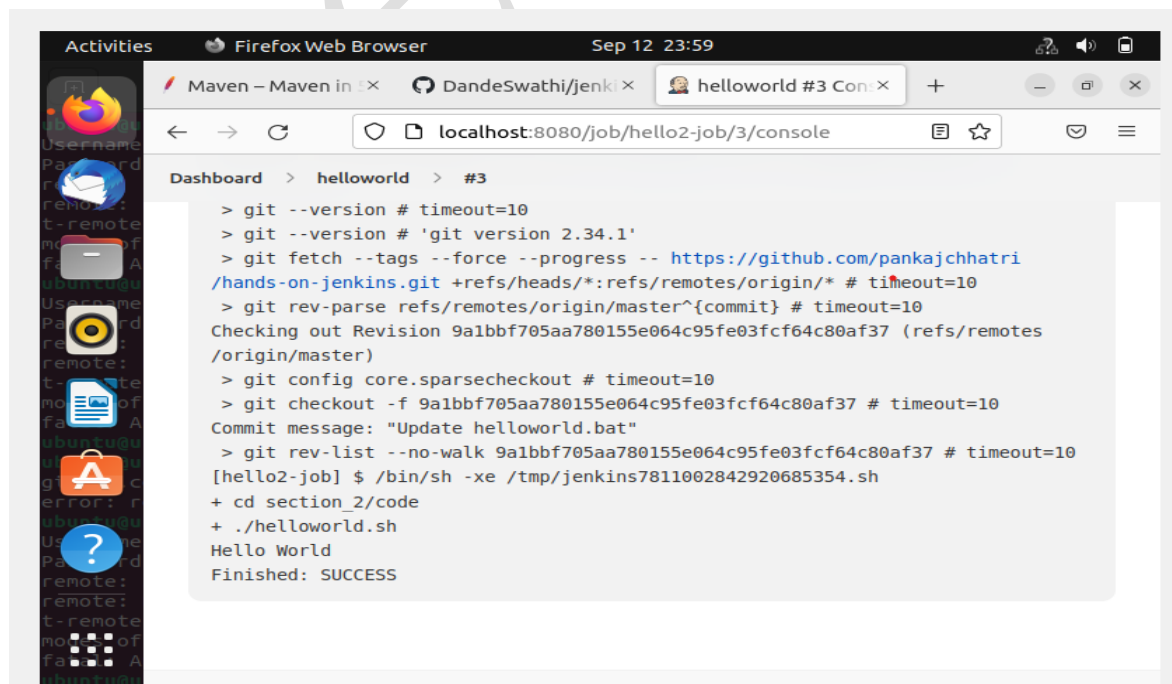
RESULT: The Helloworld job was executed successfully.

AIM: To create a job and trigger builds remotely in jenkins.

PROCEDURE:

- (1) Start Jenkins using command prompt
\$sudo systemctl start jenkins
- (2) Open Jenkins dashboard by using url localhost:8080
- (3) Select any one job from Jenkins dashboard
- (4) Here we are taking the helloworld project build before
- (5) Go to configure, Then navigate to “Build Triggers” tab.
- (6) In Build triggers tab, click on “Trigger builds” remotely (e.g. from scripts).
- (7) Then, we have to give the authentication token as we wish.
- (8) Below the text box, we can see a url then we will copy and edit it as per the requirements:
JENKINS_URL/job/helloworld/build?token=TOKEN_NAME
- (9) In the above url, we will edit JENKINS_URL as localhost:8080 which is our Jenkins url and in place of TOKEN_NAME to give the token which is given in the authentication token textbox.
- (10) After editing the url, we will get the new url i.e.,
<http://localhost:8080/job/helloworld/build/?token=12345>
- (11) Click on apply and save buttons.
- (12) Run the job remotely by using the new url we got after editing.
- (13) Browse the url then refresh the Jenkins dashboard.
- (14) Then the job was build remotely in “Build Now” an dgo to console output to see the job result.

OUTPUT:



```

Dashboard > helloworld > #3
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/pankajchhatri/hands-on-jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 9a1bbf705aa780155e064c95fe03fcf64c80af37 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9a1bbf705aa780155e064c95fe03fcf64c80af37 # timeout=10
Commit message: "Update helloworld.bat"
> git rev-list --no-walk 9a1bbf705aa780155e064c95fe03fcf64c80af37 # timeout=10
[hello2-job] $ /bin/sh -xe /tmp/jenkins7811002842920685354.sh
+ cd section_2/code
+ ./helloworld.sh
Hello World
Finished: SUCCESS

```

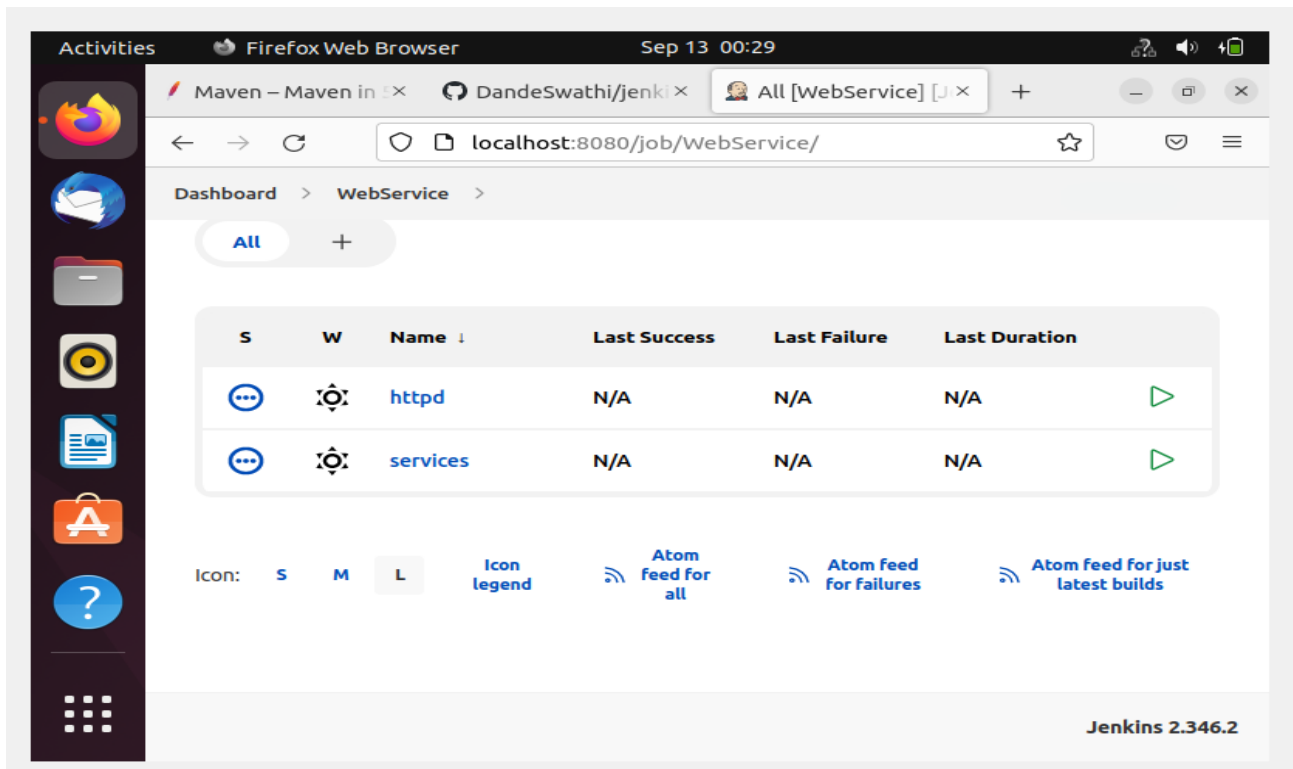
RESULT: Creating a job and trigger builds remotely was executed successfully.

AIM: To create the Folders in Jenkins.

PROCEDURE:

- (1) Create two jobs httpd and services.
- (2) Create a new folder under Jenkins which should be named WebService.
- (3) Move the above-mentioned two jobs under WebService folder.

OUTPUT:



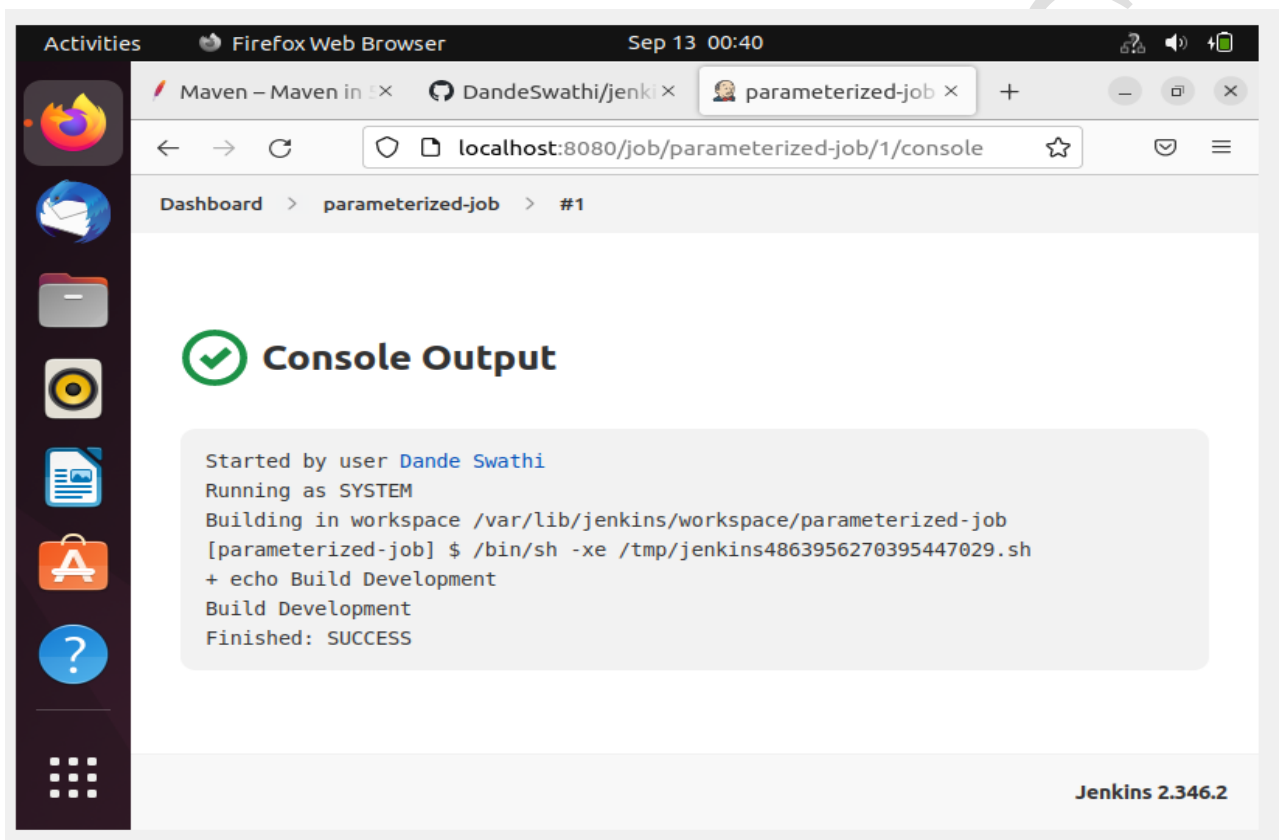
RESULT: The Folder was created successfully in Jenkins.

AIM: To create the Parameterized build in Jenkins.

PROCEDURE:

- (1) Create a parameterized job which should be named as “parameterized-job”.
- (2) Add a string parameter named “Stage” ; its default value should be “Build”.
- (3) Add a choice parameter named “env” ; its choices should be “Development, Staging and Production”.
- (4) Configure job to execute a shell command, which should echo both parameter values (you are passing in the job).
- (5) Build the Jenkins job at least once with “choice” parameter value “Development”.

OUTPUT:



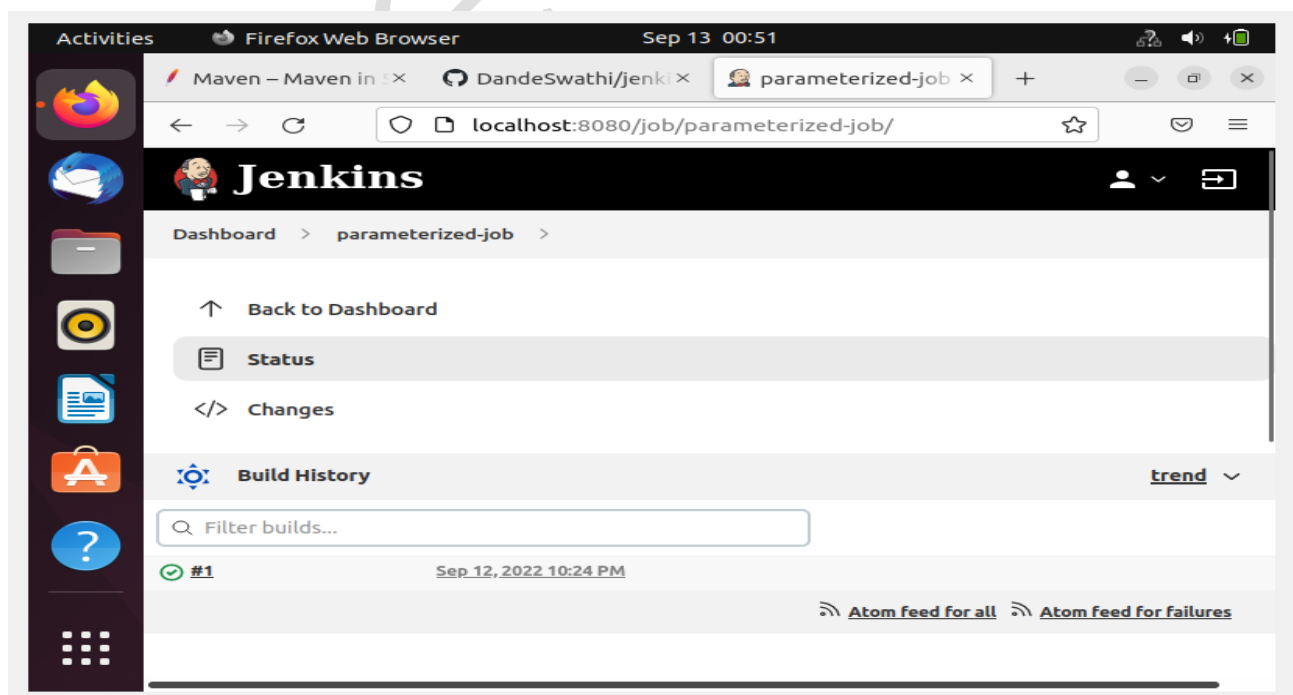
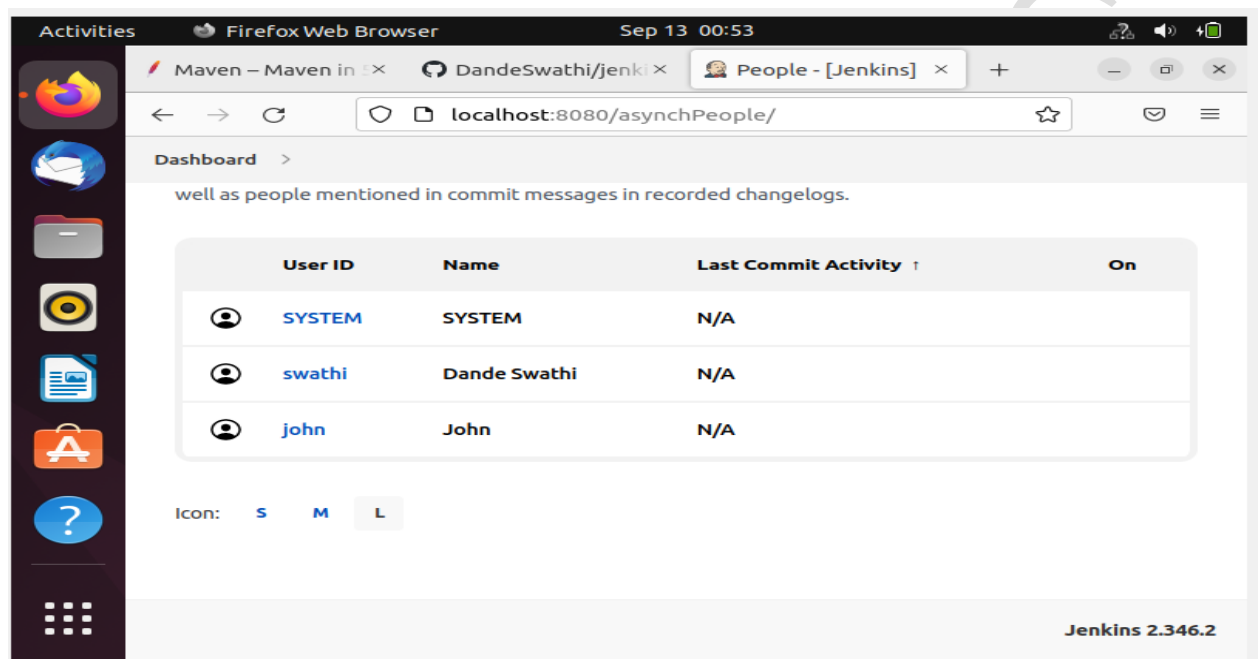
RESULT: The Parameterized build was created successfully in Jenkins.

AIM: To create the Users in Jenkins.

PROCEDURE:

- (1) Make sure that the job available from the previous task that we created a job called “parameterized-job”. If not create a new job called project “httpd-job” under admin user.
- (2) Create a Jenkins user “john” with password “Abc@Def” and full name should be “John” (its case sensitive).
- (3) Using “Project-based Matrix Authorization Strategy” assign overall read permission to “john” user. Also make sure “john” only has “read” permission to the “newly created/already existing” job.
- (4) Login with the newly created user and see he has read-only access to the job.

OUTPUT:



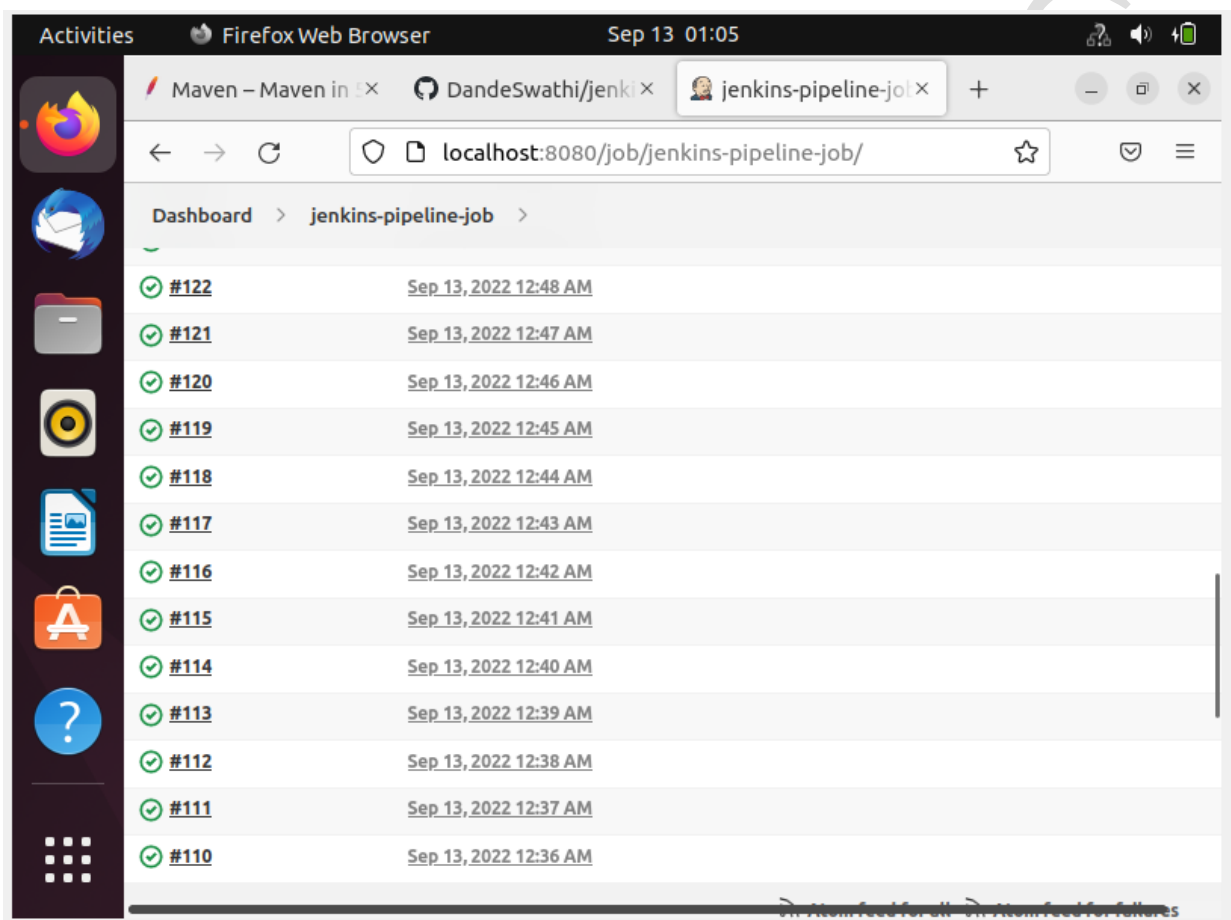
RESULT: The Users was created successfully in Jenkins.

AIM: To create Views in Jenkins.

PROCEDURE:

- (1) Create a Jenkins job which should be named “Jenkins-pipeline-job”.
- (2) Configure this job to run a simple bash command to “echo Hello Jenkins!!”.
- (3) Create a view named “jenkins-crons” (should be a List View) and make sure “Jenkins-pipeline-job” under this new view.
- (4) Schedule this newly create “job” to build periodically at every minute i.e., “*****” (please make sure to use the cron expression exactly the same how it is mentioned here).
- (5) Make sure the job builds successfully.

OUTPUT:



RESULT: The Views was created successfully in Jenkins.