Experiment 4

Aim: To study and configure master slave configuration using Jenkins

Objective: To understand master slave architecture

Theory:

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. 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.

Jenkins Master slave architecture with explanation:

Jenkins uses a Master-Slave architecture to manage distributed builds. In this architecture, Master and Slave communicate through TCP/IP protocol.

Jenkins Master:

Your main Jenkins server is the Master. The Master's job is to handle:

- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Slave:

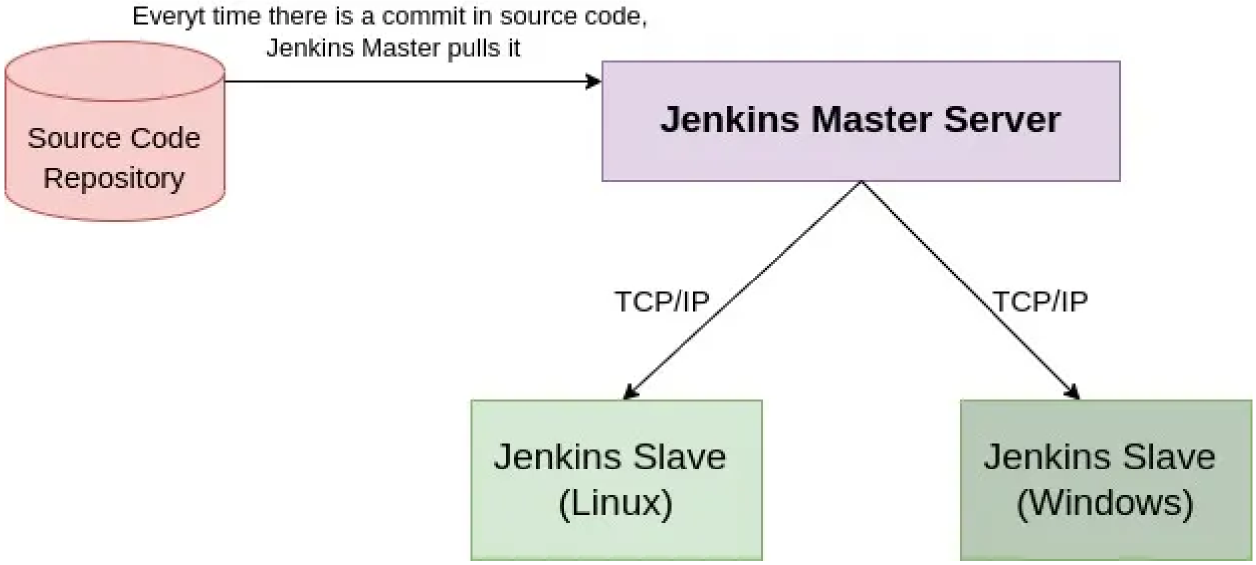
A Slave is a Java executable that runs on a remote machine. Following are the characteristics of Jenkins Slaves:

- It hears requests from the Jenkins Master instance.

- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
- You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

Main Jenkins server act like the Master. The Master's job is to handle scheduling build jobs, dispatching builds to the slaves for the actual execution, monitoring the slaves (possibly taking them online and offline as required), and recording and presenting the build results.

This architecture allows for distributed builds across multiple environments and can handle larger and heavier projects that a single Jenkins server may not be able to handle. A Master instance of Jenkins can also execute build jobs directly.



This architecture works in following steps:

1. The developer commits the changes in the source code repository

2. The CI server of Jenkins then checks the repository and pulls the newly changed code at regular intervals.

3. The build server builds the pulled code into an executable file. Feedback is sent to the developers in case of failure of the build.

4. Jenkins deploys the build application on the test server. The developers are alerted if it fails.

5. If the tests are successful and the code is error-free, the tested application is deployed on the production server.

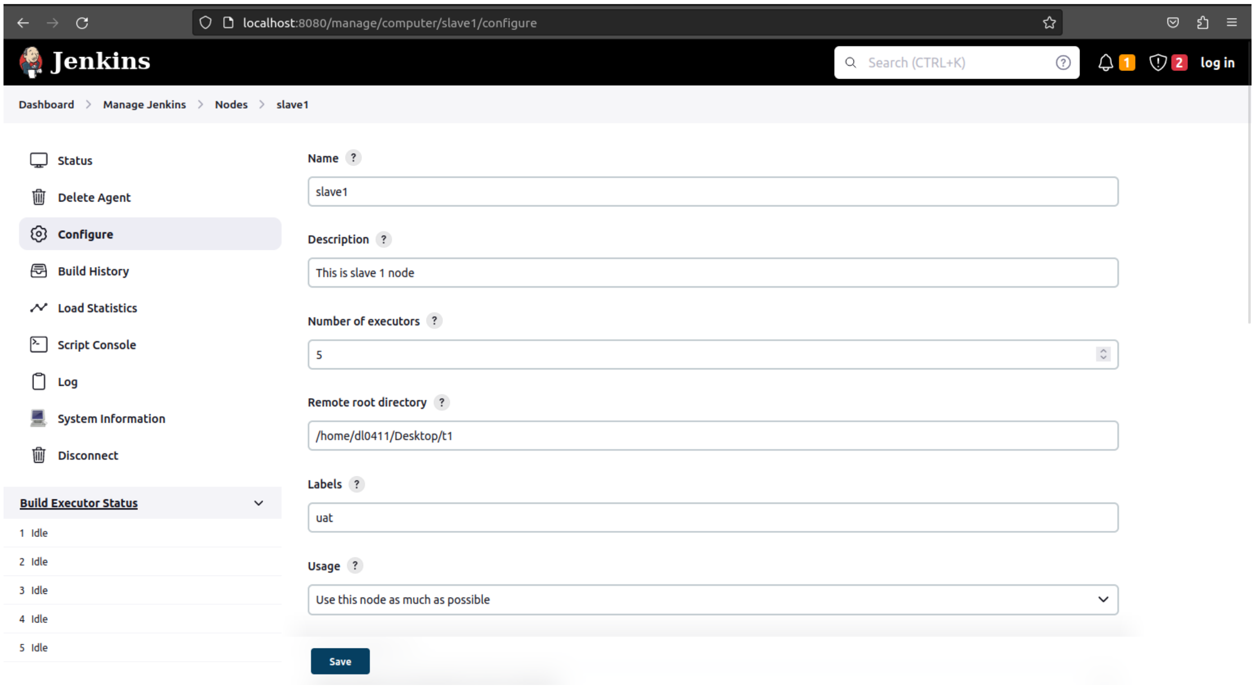
In some cases, files may have different code and require multiple builds and the Jenkins server can

The advantages of using the Jenkins Master-Slave architecture are:

1. Scalability: The architecture allows for distributed computing, which can increase the processing capacity and speed up build times for larger and complex projects.
2. Flexibility: The Slaves can be located on different hardware, operating systems, or network environments, which allows for greater flexibility in building and testing applications.
3. Resource Utilization: The Slave nodes can be configured to use available resources, such as CPU, memory, or network bandwidth, efficiently, which can improve the overall performance of the system.
4. Security: The architecture provides a secure and isolated environment for building and testing code, and it can also help in isolating any issues that may occur during the build process.

Output:

Configure Slave:



Name: slave1

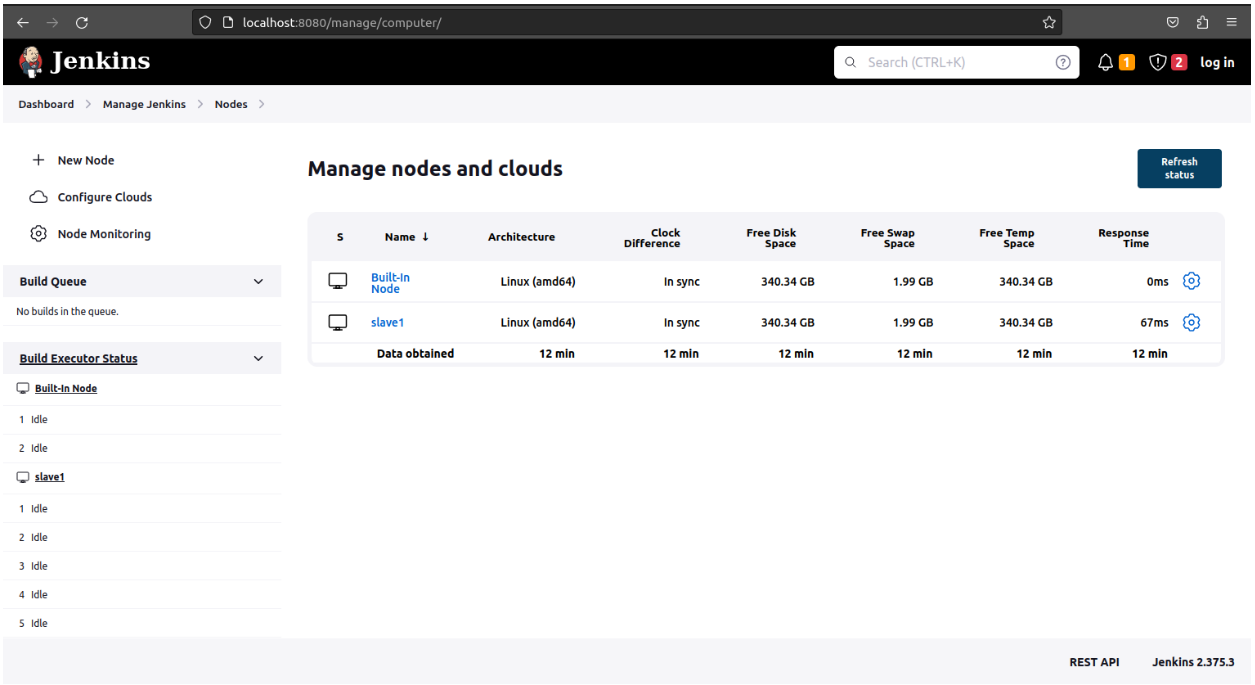
Description: This is slave 1 node

Number of executors: 5

Remote root directory: /home/dl0411/Desktop/t1

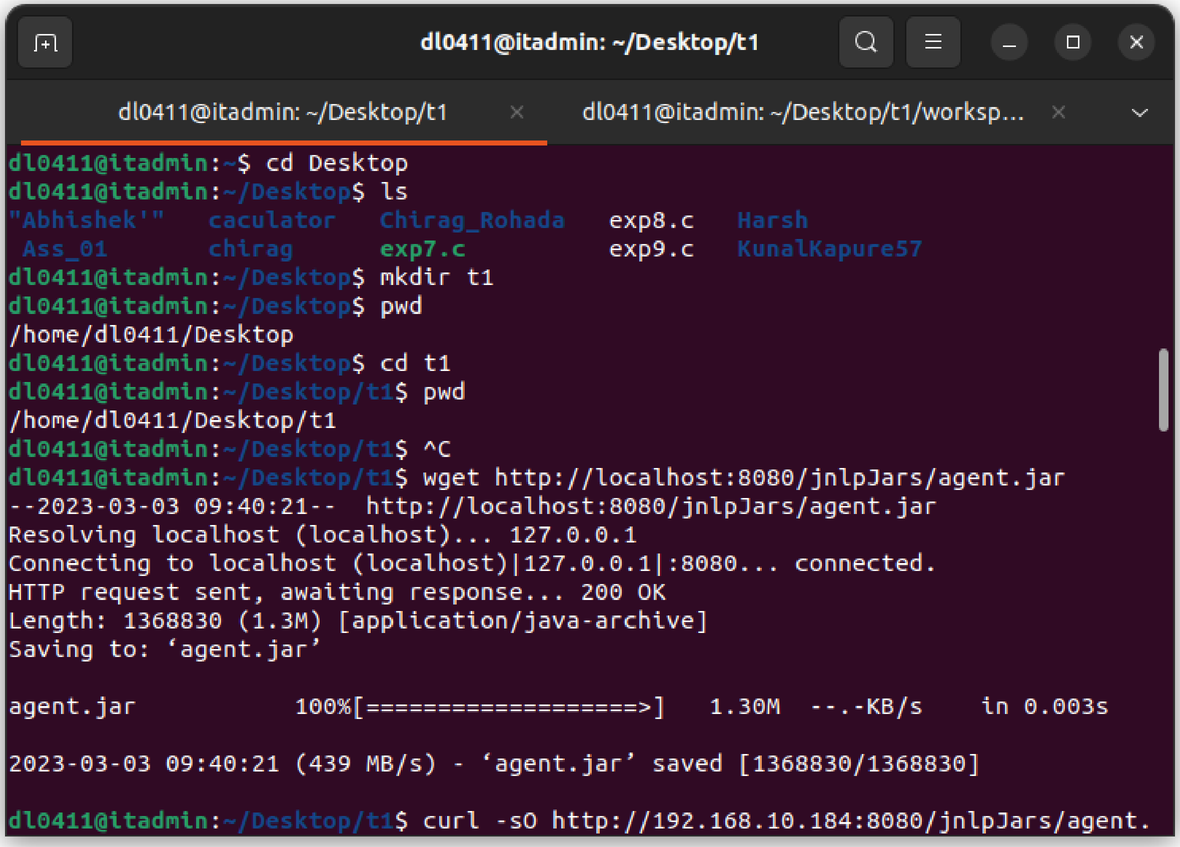
Labels: uat

Usage: Use this node as much as possible



Manage nodes and clouds

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	340.34 GB	1.99 GB	340.34 GB	0ms
	slave1	Linux (amd64)	In sync	340.34 GB	1.99 GB	340.34 GB	67ms
	Data obtained	12 min	12 min	12 min	12 min	12 min	12 min

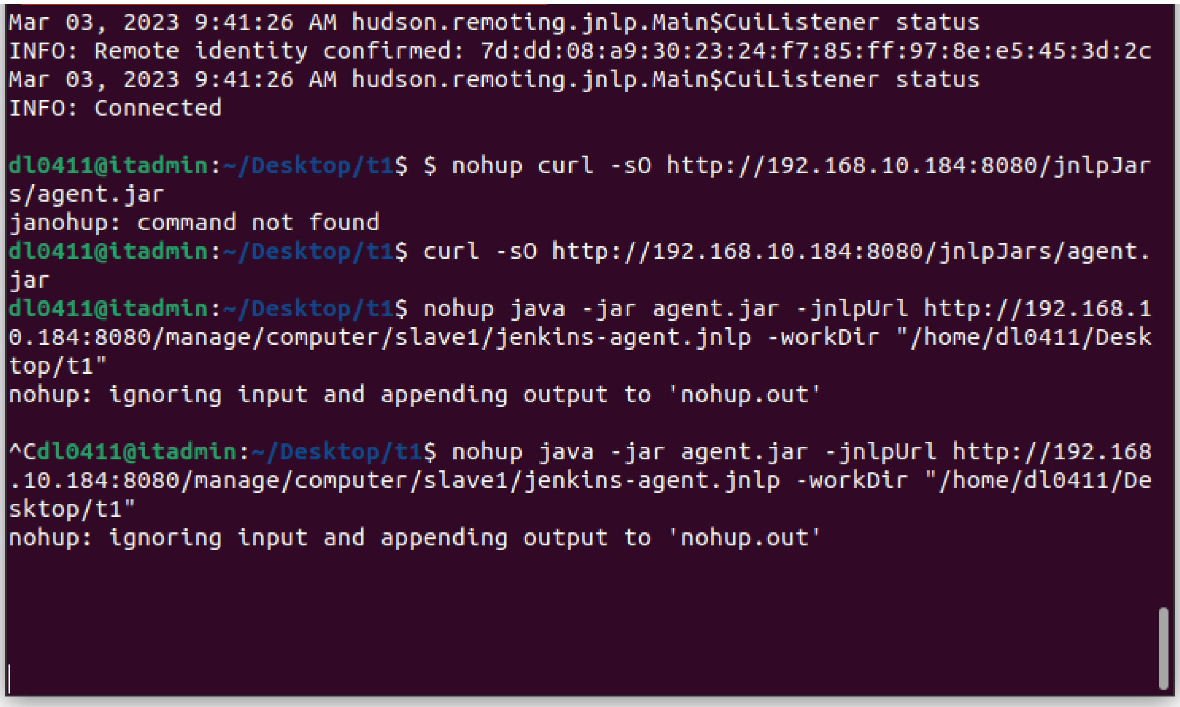

dl0411@itadmin: ~/Desktop/t1
dl0411@itadmin: ~/Desktop/t1
dl0411@itadmin: ~/Desktop/t1/worksp...

```
dl0411@itadmin:~$ cd Desktop
dl0411@itadmin:~/Desktop$ ls
"Abhishek'"   caculator   Chirag_Rohada   exp8.c   Harsh
 Ass_01       chirag      exp7.c          exp9.c   KunalKapure57
dl0411@itadmin:~/Desktop$ mkdir t1
dl0411@itadmin:~/Desktop$ pwd
/home/dl0411/Desktop
dl0411@itadmin:~/Desktop$ cd t1
dl0411@itadmin:~/Desktop/t1$ pwd
/home/dl0411/Desktop/t1
dl0411@itadmin:~/Desktop/t1$ ^C
dl0411@itadmin:~/Desktop/t1$ wget http://localhost:8080/jnlpJars/agent.jar
--2023-03-03 09:40:21--  http://localhost:8080/jnlpJars/agent.jar
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1368830 (1.3M) [application/java-archive]
Saving to: ‘agent.jar’

agent.jar           100%[===================>]   1.30M  --.-KB/s    in 0.003s

2023-03-03 09:40:21 (439 MB/s) - ‘agent.jar’ saved [1368830/1368830]

dl0411@itadmin:~/Desktop/t1$ curl -sO http://192.168.10.184:8080/jnlpJars/agent.
```

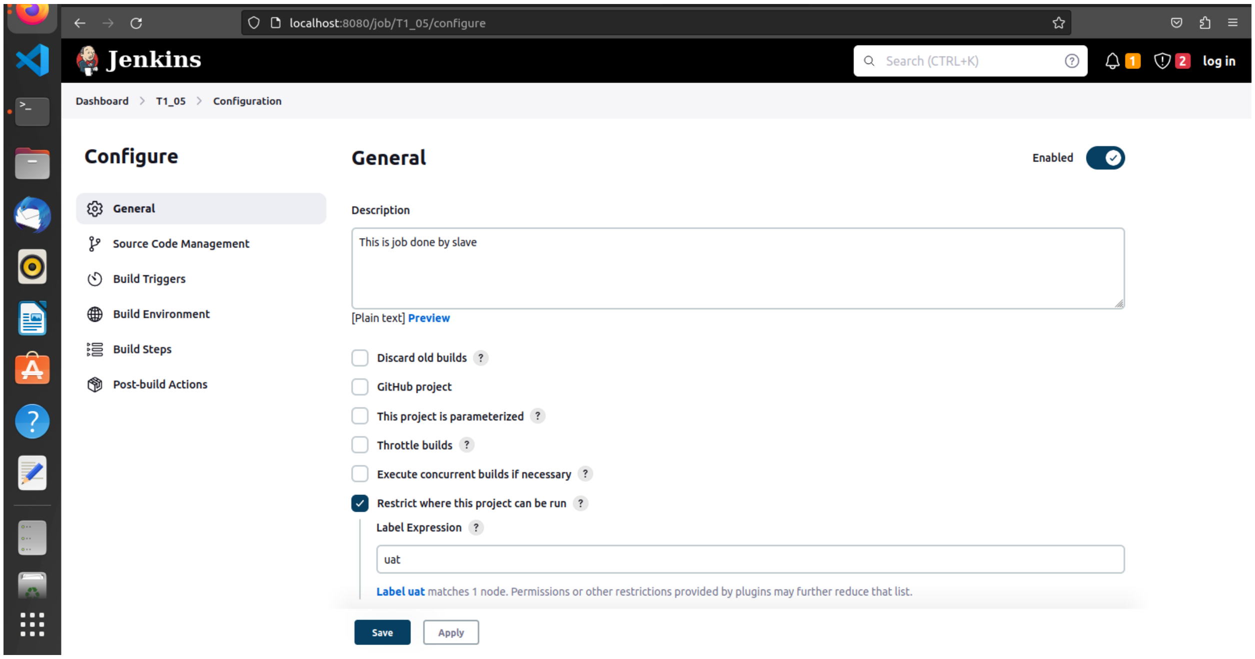


```
Mar 03, 2023 9:41:26 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 7d:dd:08:a9:30:23:24:f7:85:ff:97:8e:e5:45:3d:2c
Mar 03, 2023 9:41:26 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected

dl0411@itadmin:~/Desktop/t1$ $ nohup curl -sO http://192.168.10.184:8080/jnlpJar
s/agent.jar
janohup: command not found
dl0411@itadmin:~/Desktop/t1$ curl -sO http://192.168.10.184:8080/jnlpJars/agent.
jar
dl0411@itadmin:~/Desktop/t1$ nohup java -jar agent.jar -jnlpUrl http://192.168.1
0.184:8080/manage/computer/slave1/jenkins-agent.jnlp -workDir "/home/dl0411/Desk
top/t1"
nohup: ignoring input and appending output to 'nohup.out'

^Cdl0411@itadmin:~/Desktop/t1$ nohup java -jar agent.jar -jnlpUrl http://192.168
.10.184:8080/manage/computer/slave1/jenkins-agent.jnlp -workDir "/home/dl0411/De
sktop/t1"
nohup: ignoring input and appending output to 'nohup.out'
```

Simple Job by Slave:



General

Enabled

Description

```
This is job done by slave
```

[Plain text] Preview

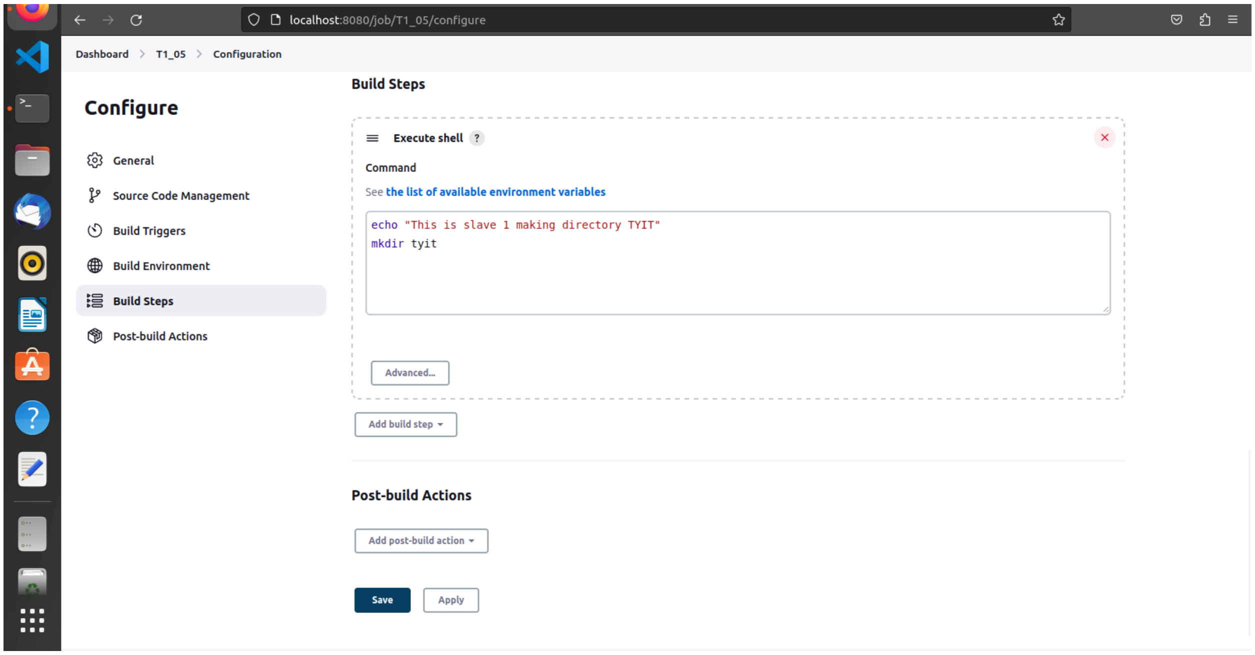
- □ Discard old builds ?
- □ GitHub project
- □ This project is parameterized ?
- □ Throttle builds ?
- □ Execute concurrent builds if necessary ?
- ☑ Restrict where this project can be run ?

Label Expression ?

```
uat
```

Label uat matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Save Apply



Build Steps

Execute shell ?

Command

See the list of available environment variables

```
echo "This is slave 1 making directory TYIT"
mkdir tyit
```

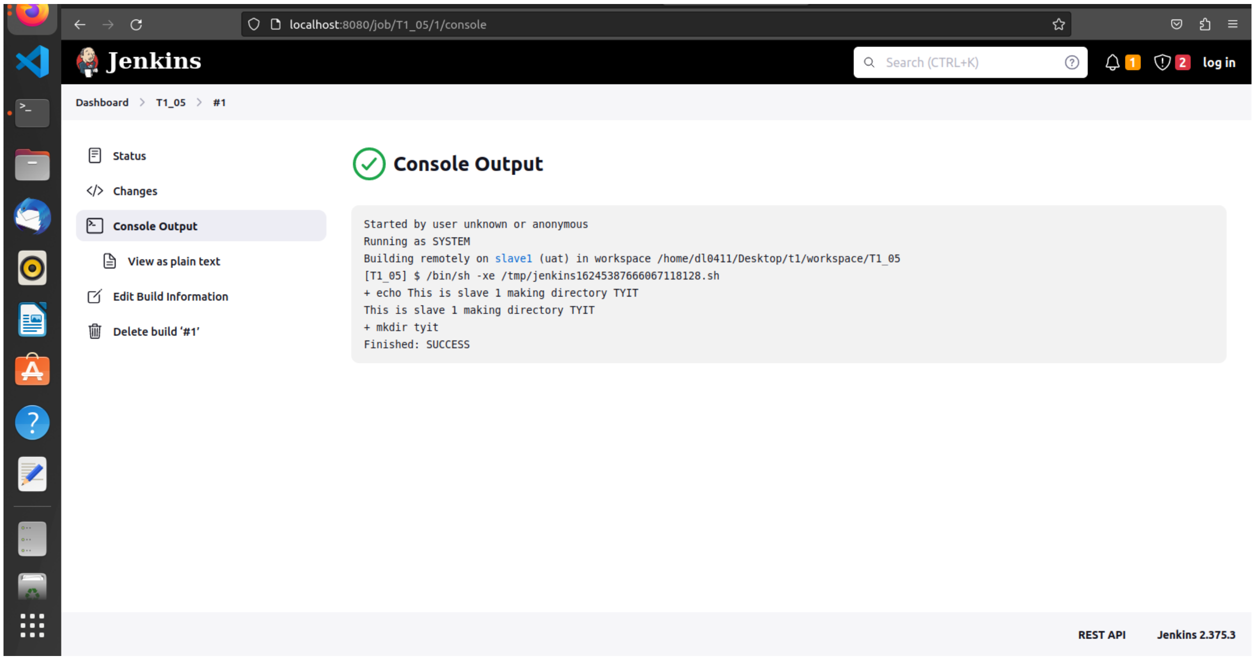
Advanced...

Add build step

Post-build Actions

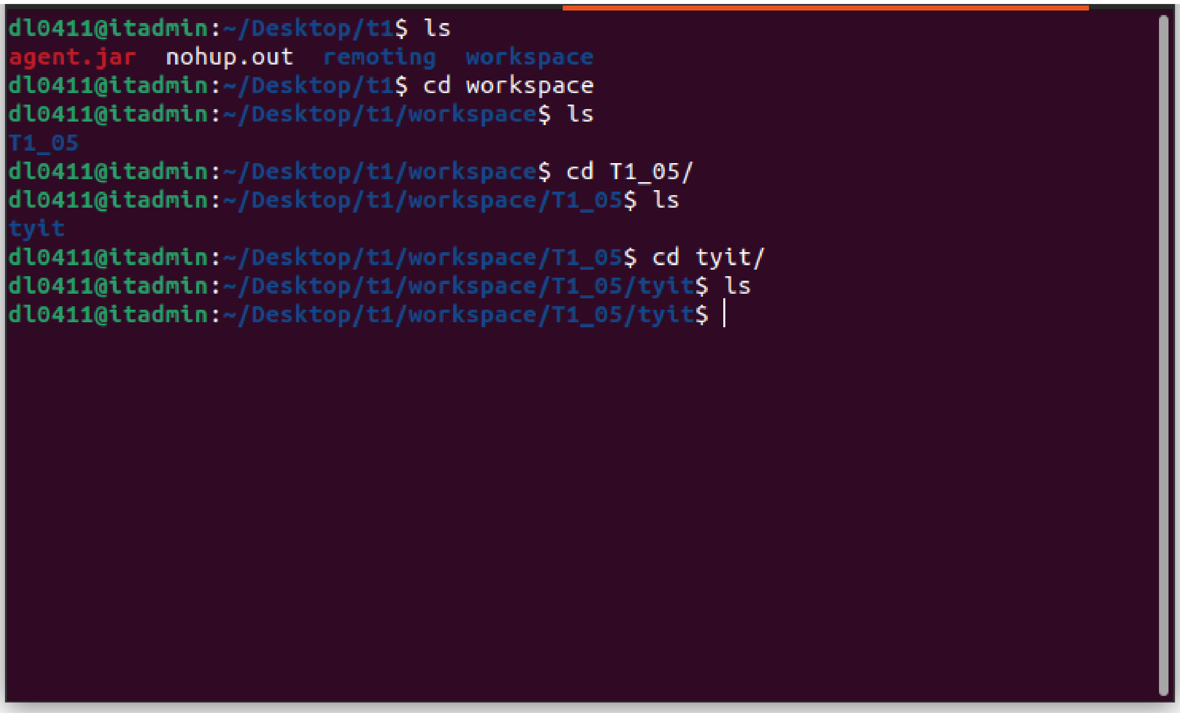
Add post-build action

Save Apply



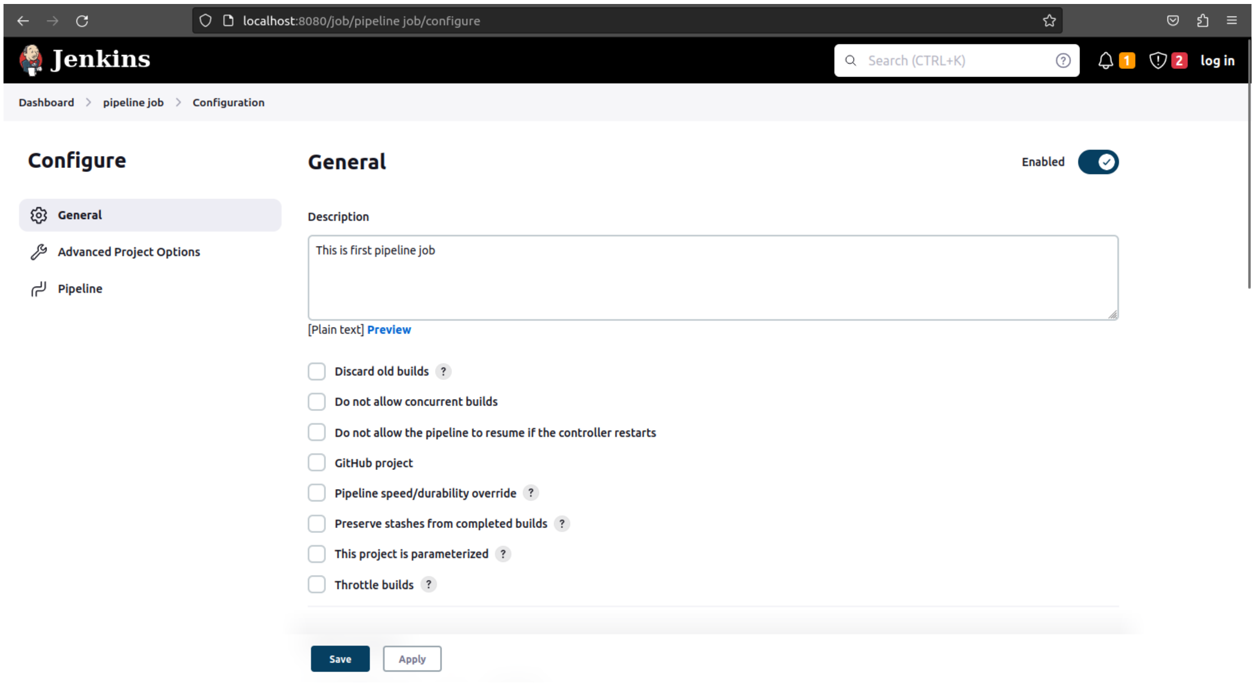
Console Output

```
Started by user unknown or anonymous
Running as SYSTEM
Building remotely on slave1 (uat) in workspace /home/dl0411/Desktop/t1/workspace/T1_05
[T1_05] $ /bin/sh -xe /tmp/jenkins16245387666067118128.sh
+ echo This is slave 1 making directory TYIT
This is slave 1 making directory TYIT
+ mkdir tyit
Finished: SUCCESS
```



```
dl0411@itadmin:~/Desktop/t1$ ls
agent.jar  nohup.out  remoting  workspace
dl0411@itadmin:~/Desktop/t1$ cd workspace
dl0411@itadmin:~/Desktop/t1/workspace$ ls
T1_05
dl0411@itadmin:~/Desktop/t1/workspace$ cd T1_05/
dl0411@itadmin:~/Desktop/t1/workspace/T1_05$ ls
tyit
dl0411@itadmin:~/Desktop/t1/workspace/T1_05$ cd tyit/
dl0411@itadmin:~/Desktop/t1/workspace/T1_05/tyit$ ls
dl0411@itadmin:~/Desktop/t1/workspace/T1_05/tyit$ 
```

Pipeline Job by slave:



localhost:8080/job/pipeline job/configure
Jenkins
Search (CTRL+K)
1
2
log in
Dashboard > pipeline job > Configuration
Configure
General
Advanced Project Options
Pipeline

General

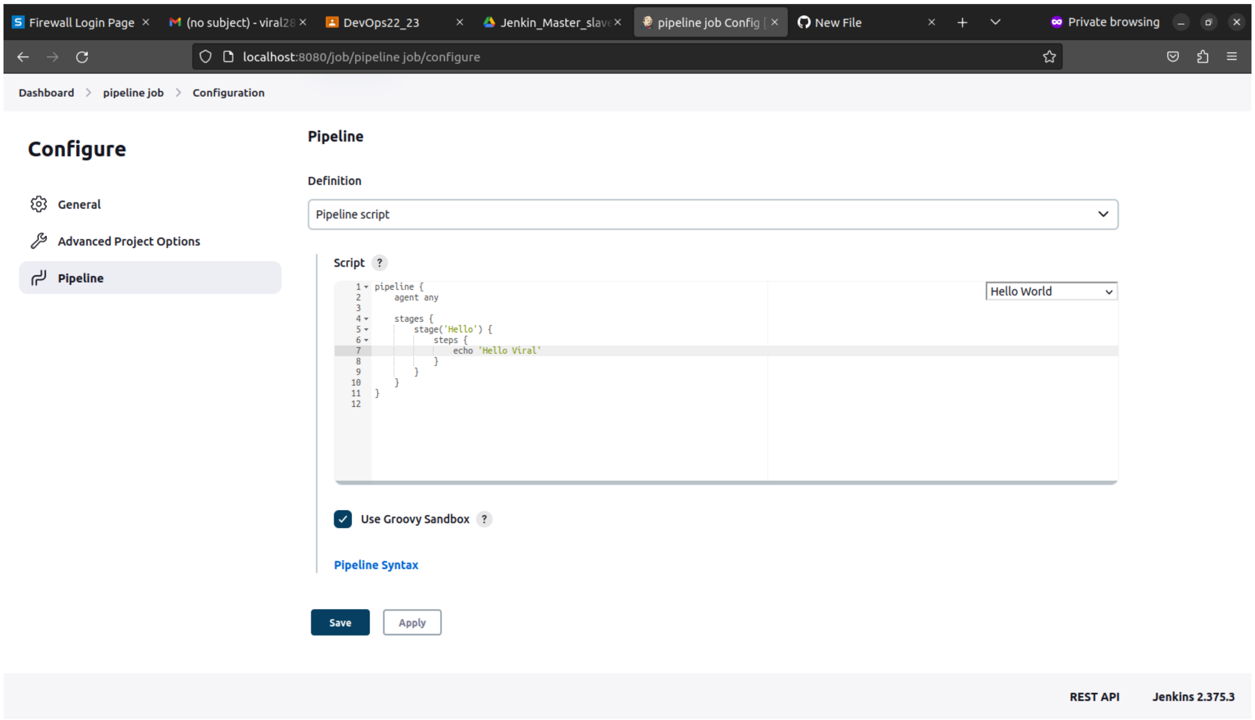
Enabled ☑

Description

This is first pipeline job

[Plain text] Preview

- □ Discard old builds ?
- □ Do not allow concurrent builds
- □ Do not allow the pipeline to resume if the controller restarts
- □ GitHub project
- □ Pipeline speed/durability override ?
- □ Preserve stashes from completed builds ?
- □ This project is parameterized ?
- □ Throttle builds ?



Pipeline

Definition

Pipeline script

Script ?

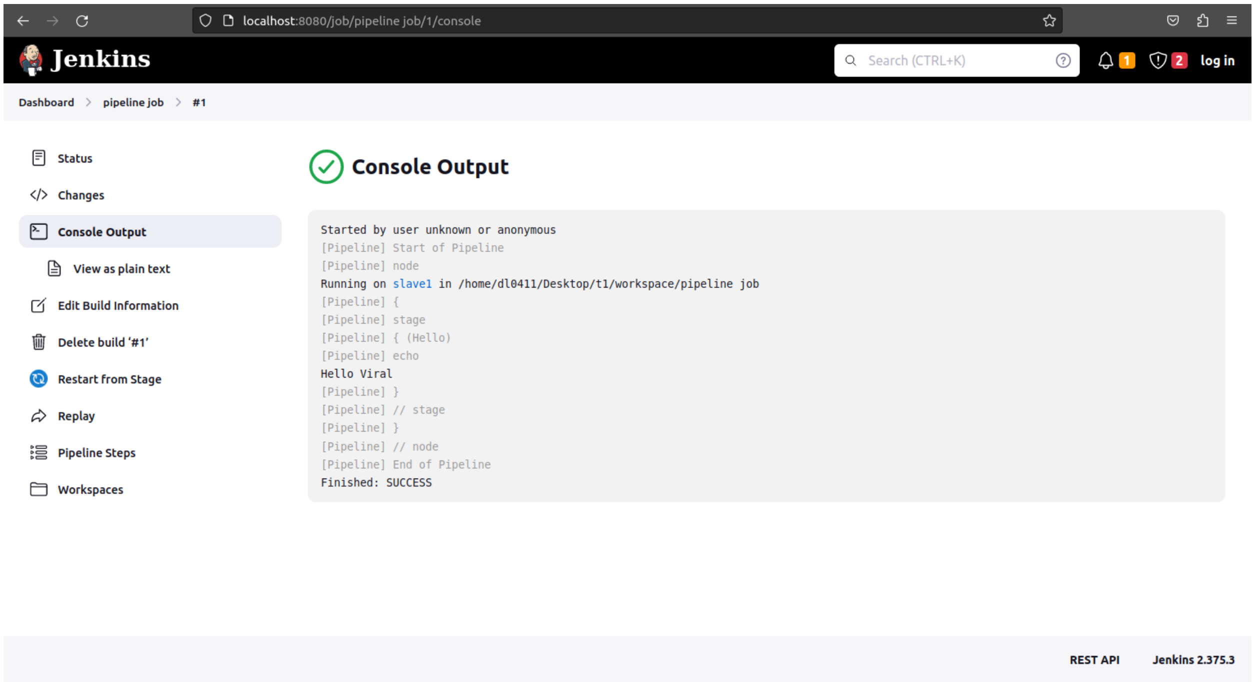
Hello World

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello Viral'
            }
        }
    }
}
```

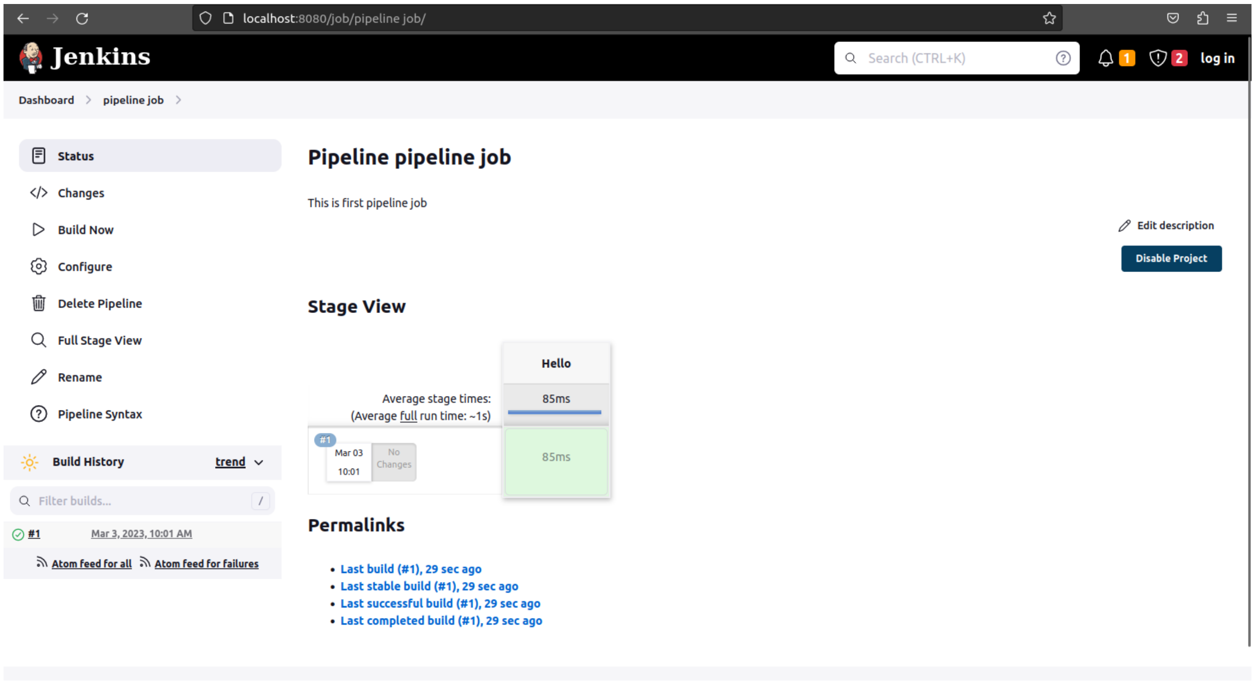
☑ Use Groovy Sandbox ?

Pipeline Syntax



Console Output

```
Started by user unknown or anonymous
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave1 in /home/dl0411/Desktop/t1/workspace/pipeline job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello Viral
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Pipeline pipeline job

This is first pipeline job

Stage View

Average stage times: 85ms

(Average full run time: ~1s)

85ms

Permalinks

- Last build (#1), 29 sec ago
- Last stable build (#1), 29 sec ago
- Last successful build (#1), 29 sec ago
- Last completed build (#1), 29 sec ago

Outcome: To Configure Master Slave configuration and build job using Jenkins file

Conclusion: Jenkins Master-Slave architecture is a powerful tool for managing distributed builds. It allows for the scheduling and execution of build jobs across multiple environments and can handle larger and heavier projects. The Master server handles the scheduling and dispatching of builds to the Slave nodes, while also monitoring their status and recording and presenting the build results. This architecture provides a flexible and scalable solution for managing complex build processes. We have implemented master slave configuration in jenkins using two different systems