

NAME: _____ STD.: _____ DIV.: _____

DATE :

PAGE :

Roll No: 67

NAME: HERAMB R. PAWAR

SUB: SEPM

PRACTICAL 6

AIM: TO UNDERSTAND JENKINS MASTER SLAVE ARCHITECTURE AND SCALE YOUR JENKINS STANDALONE Implementation by Implementing Slave node.

Problem Stat:

To explore Jenkins Master Slave architecture to Scale Implementation and increase efficiency to meet up the growing complexity and size of software development process.

The aim of the project is to gain a deeper understanding of Jenkins master-slave architecture and to implement slave node to improve the overall performance of Jenkins system.

1.

Jenkins

Theory:

Jenkins is an open source automation server that is widely used for continuous integration and continuous delivery of software applications. One of the key features of Jenkins is its ability to ^{distribute} ~~describe~~ workloads across multiple machines using master-slave architecture.

In the Jenkins master-slave architecture, the master node is responsible for managing and scheduling jobs.

While the Slave nodes are responsible for executing actual jobs. The master node communicates with the Slave nodes through a designated protocol such as SSH or JNLP.

When a job is submitted to Jenkins, the master node determines which Slave node is best suited for the job based on factors such as availability, workload, and capabilities. The master node then sends the job to the selected Slave node, which executes the job and sends results back to the master node.

Using a master-slave architecture offers several benefits for Jenkins users. It allows for efficient workload distribution, as multiple jobs can be executed simultaneously across multiple Slave nodes. It also provides increased reliability and scalability, as additional Slave nodes can be added or removed as needed.

In addition, the master-slave architecture enables users to run jobs on different platforms and operating systems, as well as take advantage of specialized hardware or software configurations on the Slave nodes.

Overall, Jenkins master-slave architecture is a powerful feature that allows for efficient and scalable automation of software development and delivery workflows.

NAME: _____ STD.: _____ DIV.: _____

DATE :

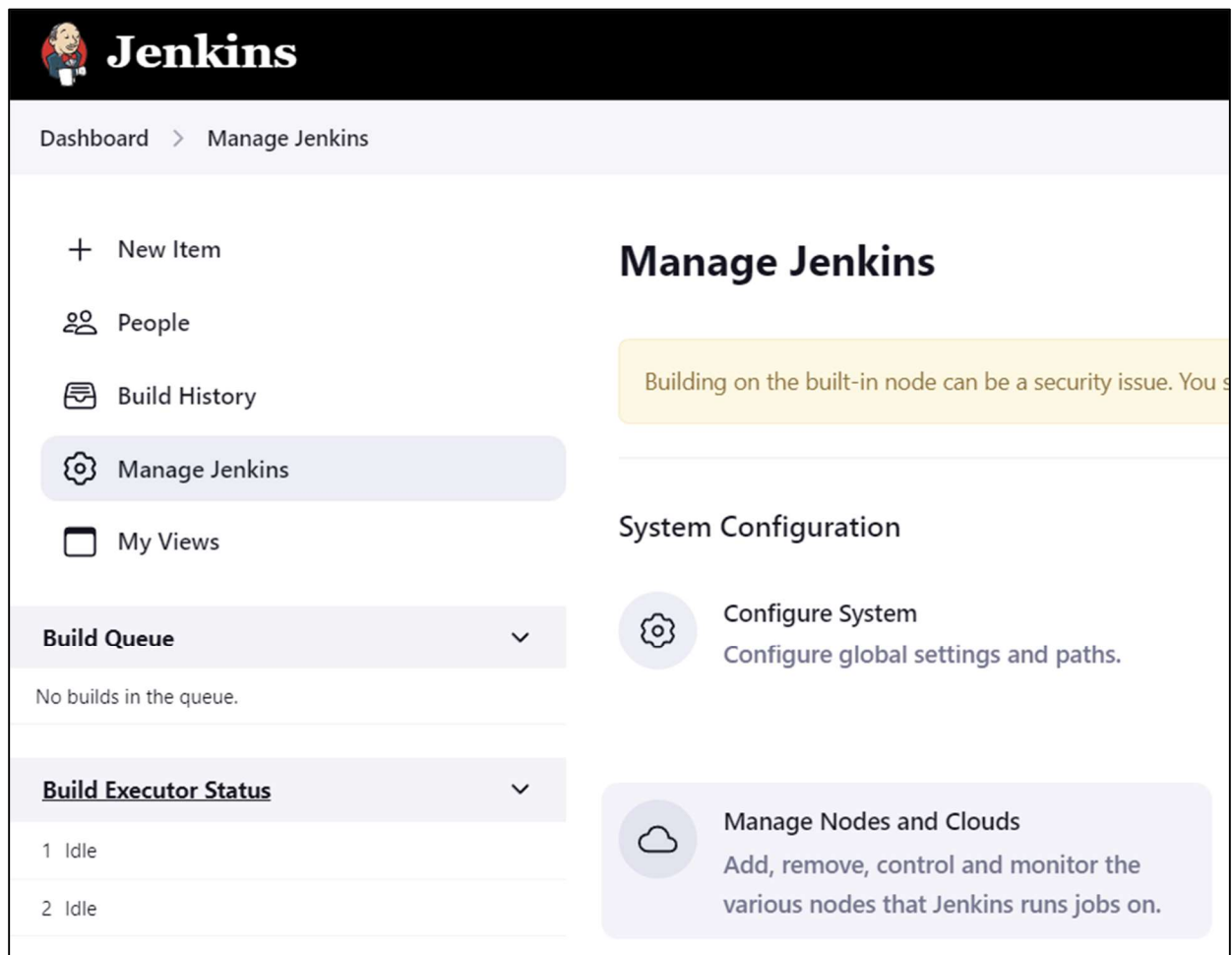
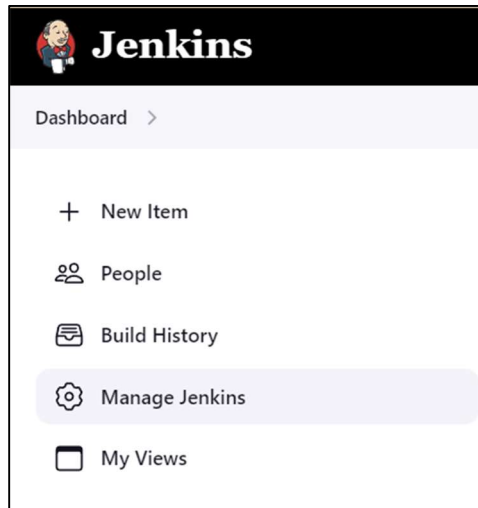
PAGE :

Conclusion:

Thus, we have understood what Jenkins Master and Slave Architecture is along with the importance of Scalability while developing Software. We also learnt the steps to show scalability in our Jenkins setup by creating new slave node.

Practical Output:

Creating New Node/Slave:



Manage nodes and clouds

[+ New Node](#)

| S | Name ↓ | Architecture | Clock Difference | Free Disk Space | Free Swap Space | Free Temp Space | Response Time |
|---|---------------|---------------|------------------|-----------------|-----------------|-----------------|---------------|
| | Built-In Node | Linux (amd64) | In sync | 4.53 GB | 0 B | 4.53 GB | 0ms |
| | Data obtained | 25 min | 25 min | 25 min | 25 min | 25 min | 25 min |

New node

Node name

Type



Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

[Create](#)

Note:

1. Parallel install **java jdk version between 8 to 11 on slave**
2. Create a folder **Jenkins** on slave1 and give all permissions to it
3. add the address of the same on master

```
ubuntu@ip-172-31-12-203:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-12-203:~$ mkdir jenkins
ubuntu@ip-172-31-12-203:~$ chmod 777 jenkins
ubuntu@ip-172-31-12-203:~$ cd jenkins
ubuntu@ip-172-31-12-203:~/jenkins$ pwd
/home/ubuntu/jenkins
```

Creating First Job That is To Be Executed On Slave1

General Enabled

Description

This My First Job To Be Runned On Slave 1

[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 ?

slave 1

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

Save

Apply

Executing First Job on Slave1

Jenkins

Search (CTRL+K) Heramb Ramakant Pawar [log out](#)

Dashboard > First Job > #2 > Console Output

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#2'

Previous Build

Console Output

Started by user [Heramb Ramakant Pawar](#)
Running as SYSTEM
Building remotely on [slave 1](#) (slave 1) in workspace /home/ubuntu/jenkins/workspace/First Job
[First Job] \$ /bin/sh -xe /tmp/jenkins14081123029230609542.sh
+ echo Running On Slave 1
Running On Slave 1
Finished: SUCCESS

REST API Jenkins 2.387.1

Slave 1 Dashboard

Agent slave 1

Mark this node temporarily offline

?

this is slave 1

Edit description

Labels

1 slave

Projects tied to slave 1

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|-----------|--------------|--------------|---------------|
| | | First Job | 48 sec #2 | N/A | 0.26 sec |

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Slave 1 Build History

Build History on slave 1

This history is not guaranteed to include all subtasks executed on the node, e.g. Jenkins Pipeline subtasks will not be displayed.

Timeline

Mar 14

Mar 15

Mar 16

Mar 17

Mar 18

Mar 19

18hr

19hr

20hr

21hr

22hr

First Job #2

First Job #1