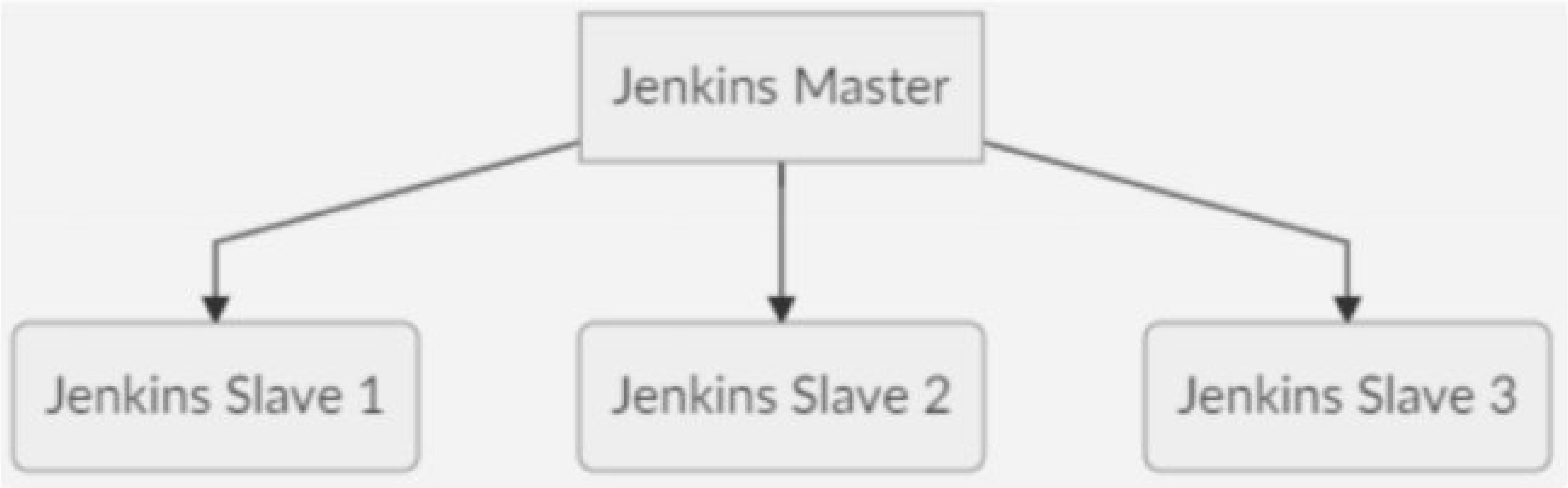
Experiment no -06

Aim: To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.

Theory: Understanding the master and slave architecture
A standalone Jenkins instance can grow fairly quickly into a disk-munching, CPU-eating monster. To prevent this from happening, we can scale Jenkins by implementing a slave node architecture, which can help us offload some of the responsibilities of the master Jenkins instance.
Let&#39;s clarify this concept. A Jenkins slave node is simply a device configured to act as an automation executor on behalf of the master. The Jenkins master simply represents the base installation of Jenkins. The master will continue to perform basic operations and serve the user interface, while the slaves do the heavy lifting.
This distributed computing model will allow the Jenkins master to remain responsive to users, while offloading automation execution to the connected slave(s). To illustrate the concept of a master, and slave mode architecture let&#39;s look at an example. Figure 2-1 shows a Jenkins master and three slave nodes of varying OS types:



The Jenkins master acts to schedule the jobs and assign slaves and send
builds to slaves to execute the jobs.

It will also monitor the slave state (offline or online) and getting back the build result responses from slaves and the display build results on the console output. The workload of building jobs is delegated to multiple slaves.

Steps to Configure Jenkins Master and Slave Nodes

1) Click on Manage Jenkins in the left corner on the Jenkins dashboard.
2) Click on Manage Nodes.
3) Select New Node and enter the name of the node in the Node Name field.
4) Select Permanent Agent and click the OK button. Initially, you will get only one option, &quot;Permanent Agent.&quot; Once you have one or more slaves
you will get the &quot;Copy Existing Node&quot; option.
5) Enter the required information.
6) Enter the Hostname in the Host field.
7) Select the Add button to add credentials. and click Jenkins.
8) Enter Username, Password, ID, and Description.
9) Select the dropdown menu to add credentials in the Credentials field.
10) Select the next dropdown to add the Host Key Verification Strategy under Non verifying Verification Strategy.
11) Select Keep this agent online as much as possible in the Availability field.

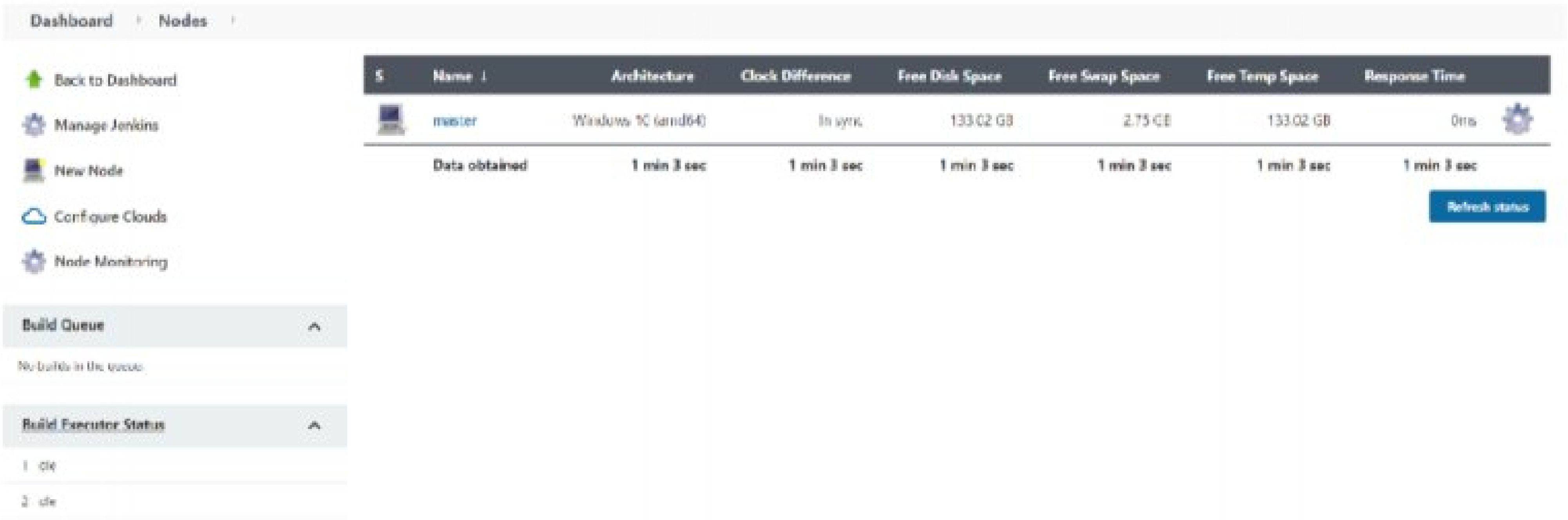
Creating a Freestyle Project and Running on The Slave Machine

1) Click on Save and it will redirect to job&#39;s view page
2) On the left pane, click the Build Now button to execute your Pipeline.
3) We can verify the history of the executed build under the Build History by clicking the build number.
4) Click on the build number and select Console Output. Here you can see the executed job in the remote host and output.

Creating a Pipeline and Running on The Slave Machine

1) Click New Item in the top left corner on the dashboard.
2) Enter the name of your project in the Enter an item name field, and select the Pipeline project, and click OK button.
3) Enter Description (optional).

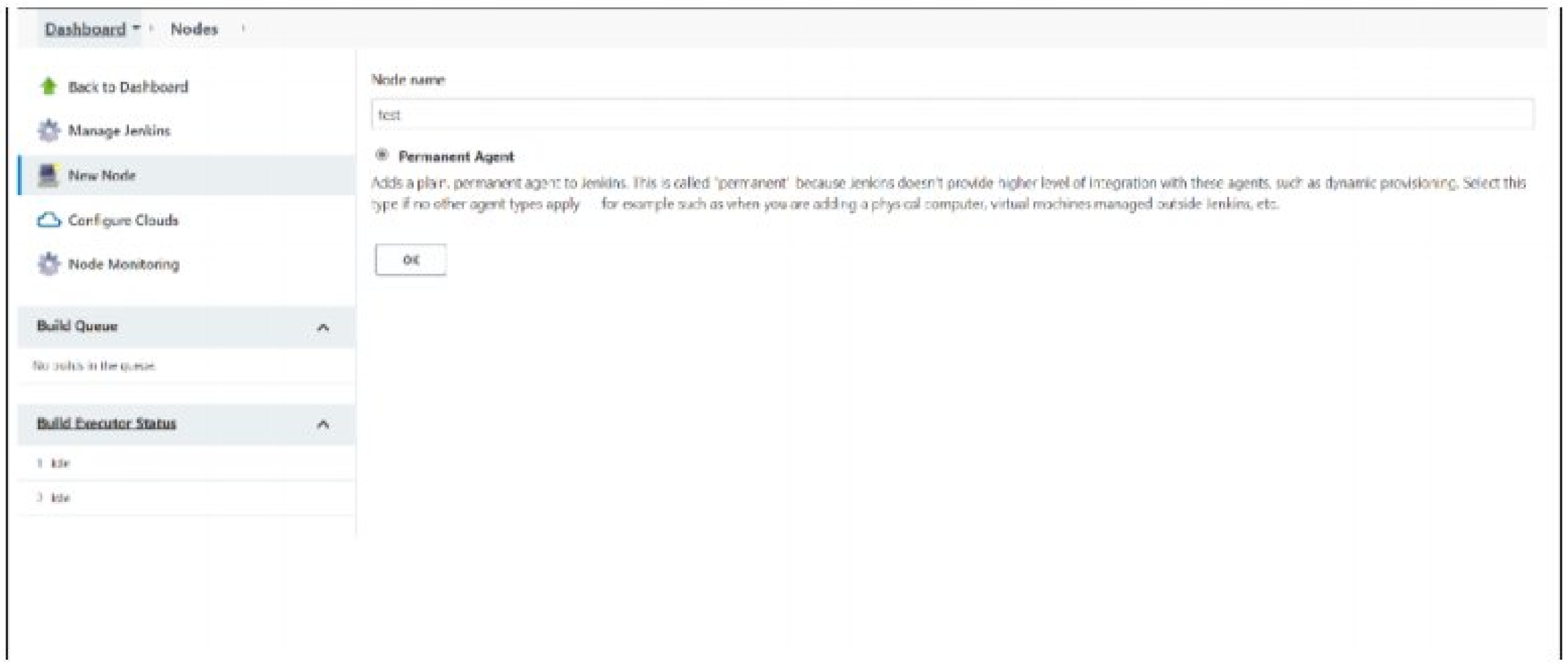
4) Go to the Pipeline section, make sure the Definition field has
the Pipeline script option selected.
5) Copy and paste the following declarative Pipeline script into a script field.
6) Click on Save, it will redirect to the Pipeline view page.
7) On the left pane, click the Build Now button to execute your Pipeline.
8) After Pipeline execution is completed, the Pipeline view will be as shown below.
9) We can verify the history of executed build under the Build History by
clicking the build number.
10) Click on build number and select Console Output. Here you can see that the pipeline ran on a slave machine.



No builds in the queue.

1 Idle
2 Idle

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Windows 10 (amd64)	In sync	133.02 GB	2.75 GB	133.02 GB	0ms
	Data obtained	1 min 3 sec	1 min 3 sec	1 min 3 sec	1 min 3 sec	1 min 3 sec	1 min 3 sec



Node name

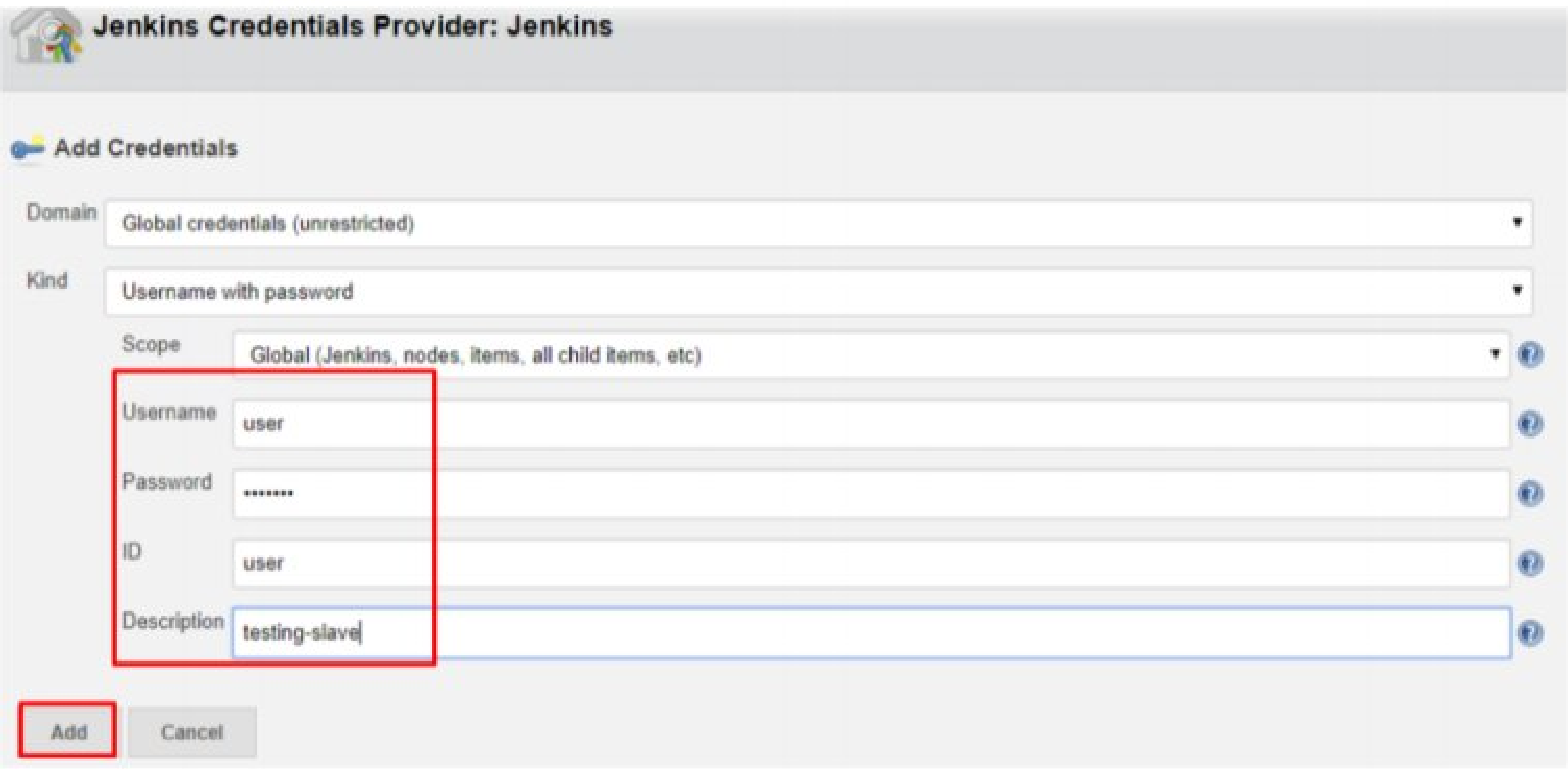
test

● 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.

No builds in the queue.

1 Idle
2 Idle



Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

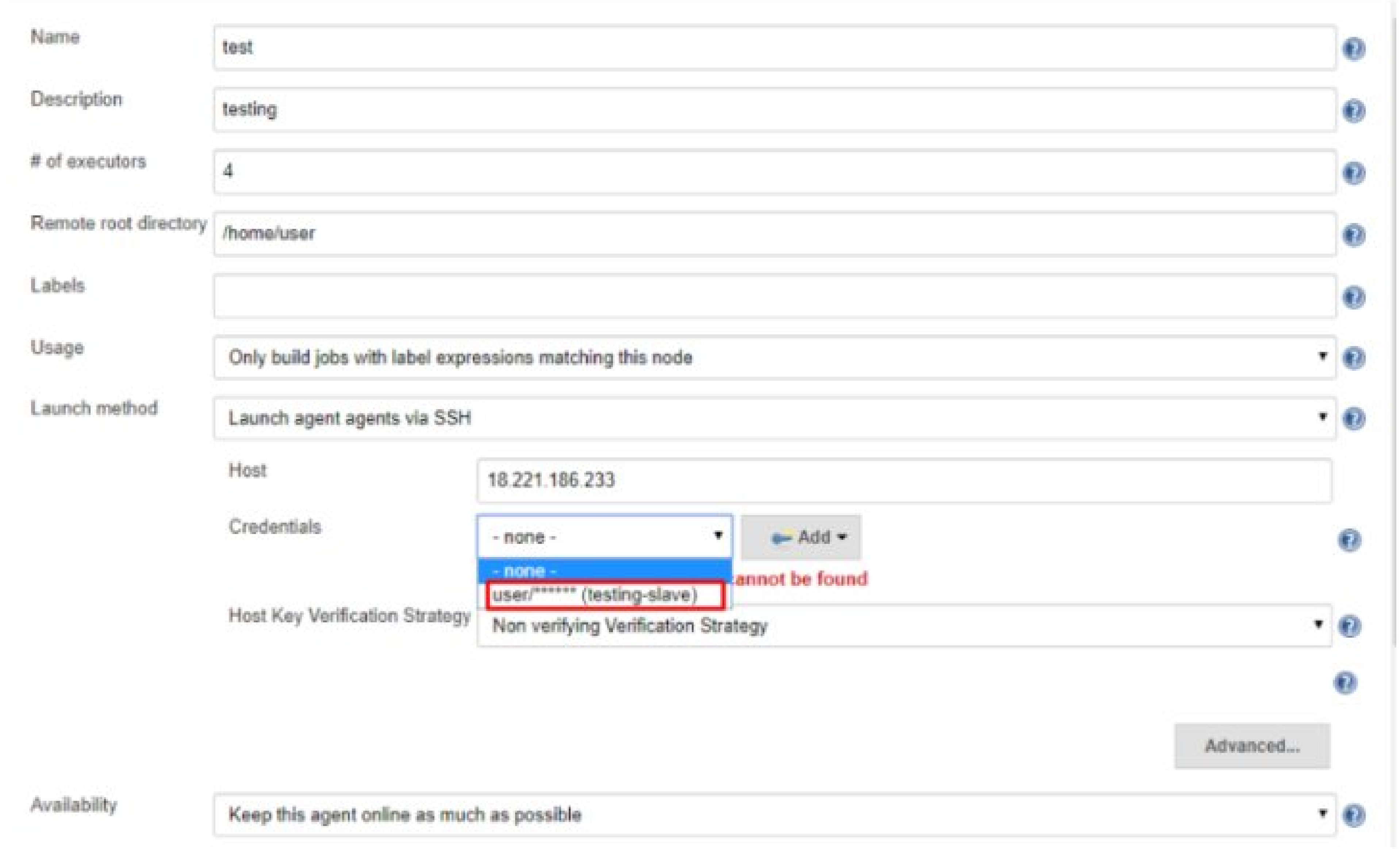
Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: user

Password: •••••••

ID: user

Description: testing-slave



Name: test

Description: testing

# of executors: 4

Remote root directory: /home/user

Labels:

Usage: Only build jobs with label expressions matching this node

Launch method: Launch agent agents via SSH

Host: 18.221.186.233

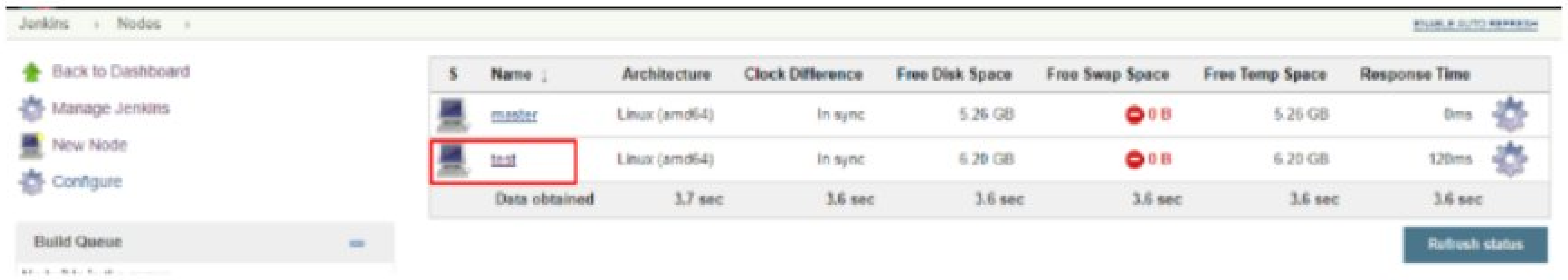
Credentials: - none -

- none -
user/****** (testing-slave)

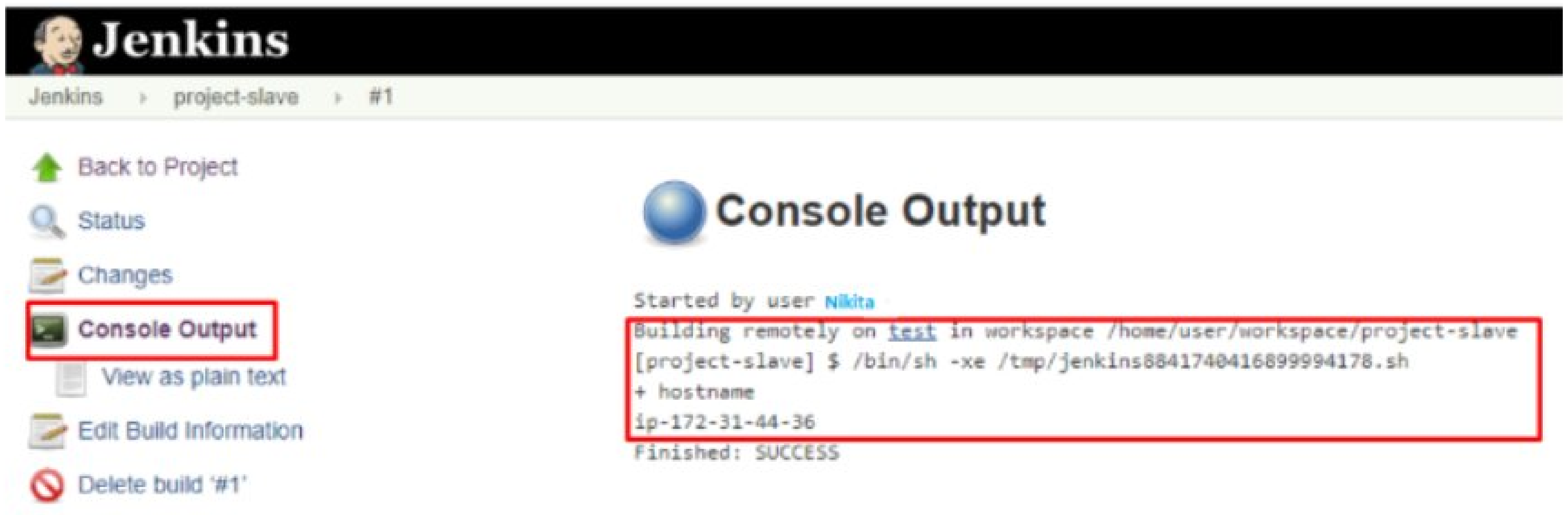
annot be found

Host Key Verification Strategy: Non verifying Verification Strategy

Availability: Keep this agent online as much as possible

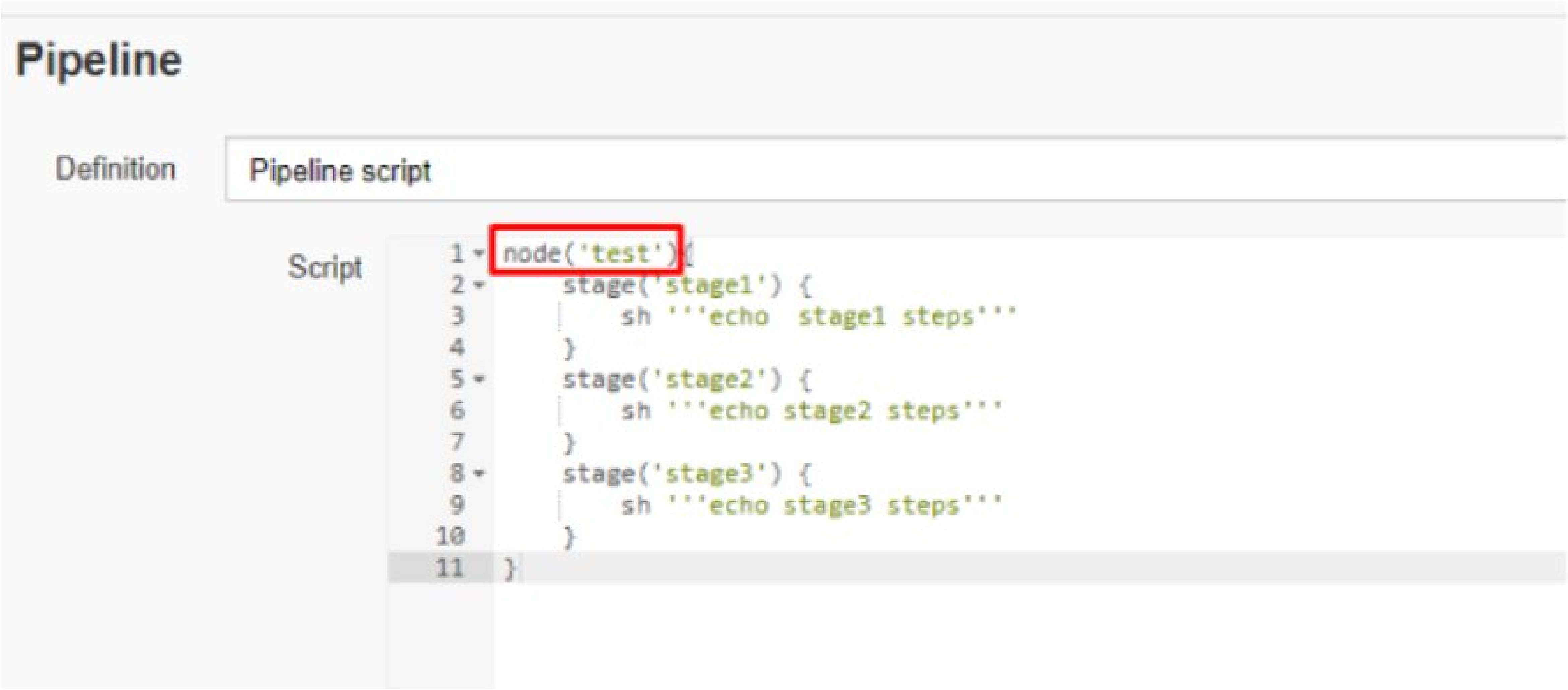


S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.26 GB	0 B	5.26 GB	0ms
	test	Linux (amd64)	In sync	6.20 GB	0 B	6.20 GB	120ms
	Data obtained	3.7 sec	3.6 sec	3.6 sec	3.6 sec	3.6 sec	3.6 sec



Console Output

```
Started by user Nikita
Building remotely on test in workspace /home/user/workspace/project-slave
[project-slave] $ /bin/sh -xe /tmp/jenkins8841740416899994178.sh
+ hostname
ip-172-31-44-36
Finished: SUCCESS
```

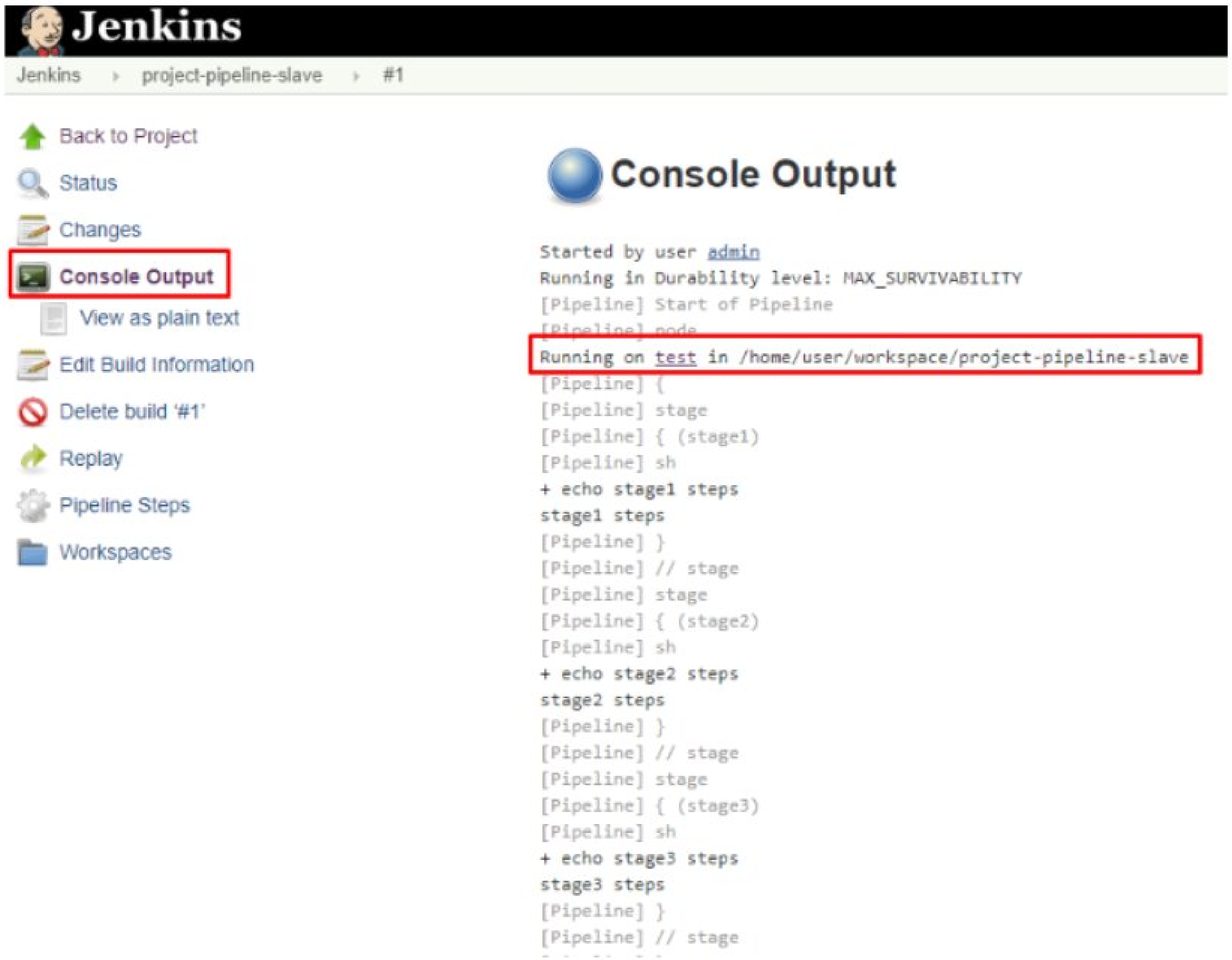


Pipeline

Definition: Pipeline script

Script

```
node('test'){
    stage('stage1') {
        sh '''echo  stage1 steps'''
    }
    stage('stage2') {
        sh '''echo stage2 steps'''
    }
    stage('stage3') {
        sh '''echo stage3 steps'''
    }
}
```



```
Started by user admin
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on test in /home/user/workspace/project-pipeline-slave
[Pipeline] {
[Pipeline] stage
[Pipeline] { (stage1)
[Pipeline] sh
+ echo stage1 steps
stage1 steps
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (stage2)
[Pipeline] sh
+ echo stage2 steps
stage2 steps
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (stage3)
[Pipeline] sh
+ echo stage3 steps
stage3 steps
[Pipeline] }
[Pipeline] // stage
```

Conclusion: Successfully understand abd implemented Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.