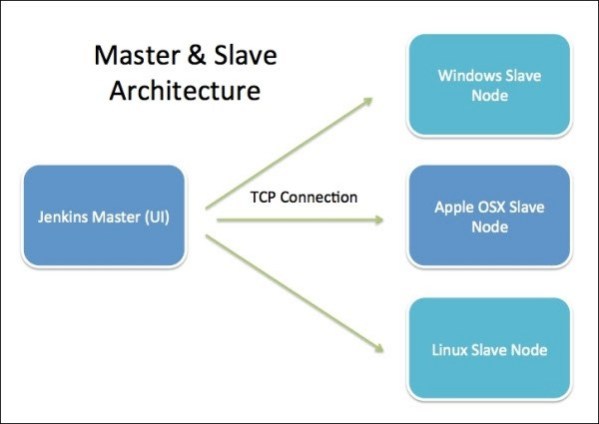
**Setting Up Master Slave Machines Using Jenkins**

Jenkins supports the master-slave architecture, i.e. many slaves work for a master. It is also known as Jenkins Distributed Builds. It also allows you to run jobs on different environments like Linux, Windows, MacOS, etc. We can also run the same test case on different environments in parallel using Jenkins Distributed Builds, which in turn helps you to achieve the desired results quickly using this distributed approach. All of the job results are collected and combined on the master node for monitoring.



**Jenkins Master**

Your main Jenkins server is the master machine. The tasks performed by the master are :

* Scheduling build jobs.
* Dispatching builds to the slaves for the execution.
* Monitor the slaves.
* Recording and presenting the build results.
* Can also execute build jobs directly.

**Jenkins Slave**

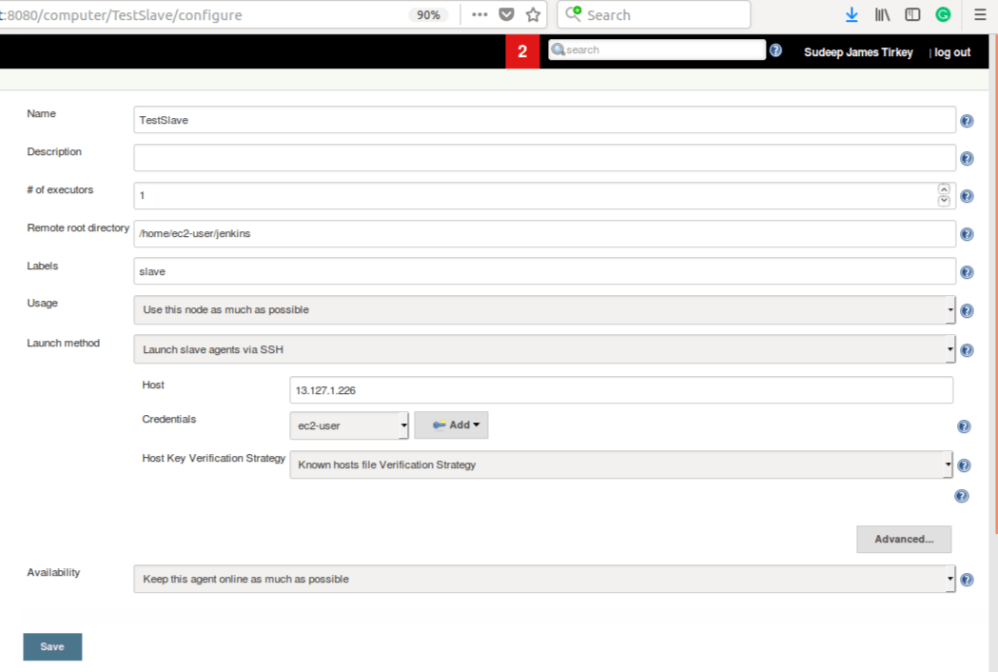
A slave is a Java executable that runs on a remote machine. The characteristics of the slave are :

* 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.
* We 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.

**Setting Up A Jenkins Slave**

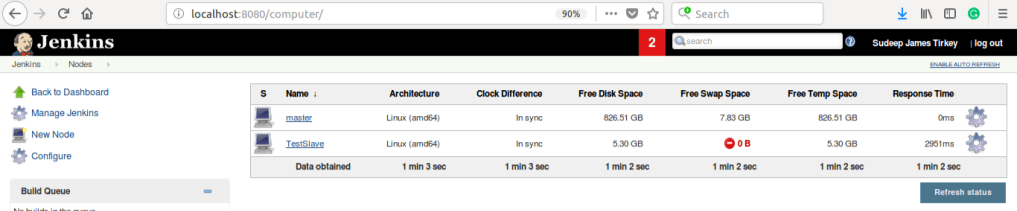
For this example, I have used one AWS EC2 instance as a slave machine and my system as the master.

1. Go to **Jenkins dashboard → Manage Jenkins ->Manage Nodes.**
2. Give a name to the node and then click **OK**. For the first time user, only one option of “**Permanent Agent**” will be present. After that, we get another option, “**Copy Existing Node**” which is self-explanatory.
3. Then, a page will open in which we have to specify the configuration of the slave machine. All of them are not required but let’s get an overview of the all parameters asked in the configuration :



* + Name: Name of the Slave which should be unique.
  + Description: Description of this slave. It is optional but giving a description of it would be really helpful for other team members.
  + # of executors: The maximum number of concurrent builds that Jenkins may perform on this agent. I have used 1 executor for testing purpose, but a good practice would be the number of CPU cores on the machine.
  + Remote root directory: An agent needs to have a directory dedicated to Jenkins. Specify the path to this directory on the agent. I’ve specified /home/ec2-user/jenkins as the remote root directory.
  + Labels: Labels are used to group multiple agents into one logical group. Multiple labels must be separated by a space.
  + Usage: Controls how Jenkins schedules builds on this node.
  + Launch method: It Controls how Jenkins starts this agent.
    - **Launch agent via execution of command on the master**: Starts an agent by having Jenkins execute a command from the master. Use this when the master is capable of remotely executing a process on another machine, e.g. via SSH or RSH.
    - Launch slave agents via SSH: Starts a slave by sending commands over a secure SSH connection. The slave needs to be reachable from the master, and you will have to supply an account that can log in to the target machine. No root privileges are required. If we select this, it will further need following credentials :
      * Host: The IP address of the slave machine.
      * Credentials: Select the credentials to be used for logging in to the remote host.
      * Host Key Verification Strategy: Controls how Jenkins verifies the SSH key presented by the remote host whilst connecting. These are of 4 types :
        + Known Host file verification strategy: Checks the known\_hosts file (~/.ssh/known\_hosts) for the user Jenkins is executing under, to see if an entry exists that matches the current connection.
        + **Manually provided key verification strategy:** Checks the key provided by the remote host matches the key set by the user who configured this connection.
        + **Manually trusted key verification strategy:** Require a user with Computer.CONFIGURE permission to authorise the key presented during the first connection to this host before the connection will be allowed to be established.
        + **Non-Verifying verification strategy:** Does not perform any verification of the SSH key presented by the remote host, allowing all connections regardless of the key they present. It’s not advisable to select as it may open the path for attackers.
  + Availability: Controls when Jenkins starts and stops this agent. It has two options available.
    - **Keep this slave on-line as much as possible:** This one is the default and normal setting. In this mode, Jenkins tries to keep the slave on-line as much as possible. If Jenkins can start the slave without user assistance, it will periodically attempt to restart the slave if it is unavailable. Jenkins will not take the slave off-line.
    - **Take this slave on-line when in demand and off-line when idle:** In this mode, if Jenkins can launch the slave without user assistance, it will periodically attempt to launch the slave while there are unexecuted jobs else the slave will be taken off-line by Jenkins

1. After giving all the required configurations, click on the **Save** and your slave machine will be up and running.



You now have a master and a slave machine. You could add more slaves or try various configurations available. even, you can try using the setup for performing various jobs, which itself can be configured according to the master-slave setup.