Master–Slave Jenkins Configuration:

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

In master-slave architecture of Jenkins, master represents basic installation of Jenkins and it handles all tasks for build system.

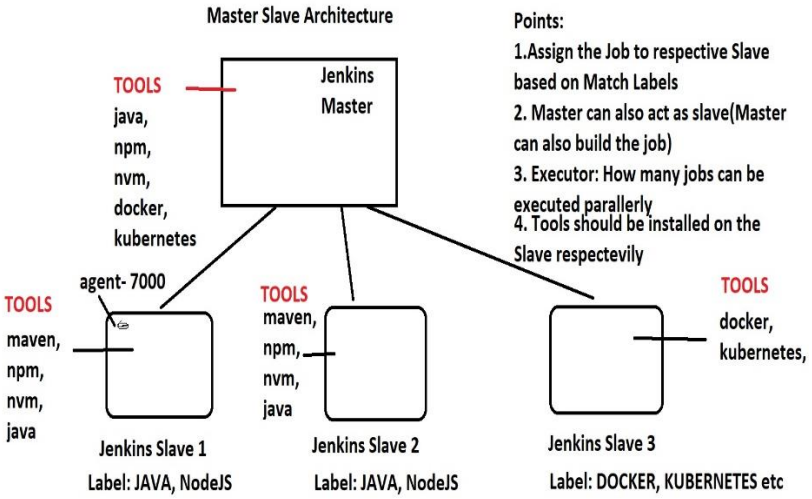
Jenkins master is used to handle following things:

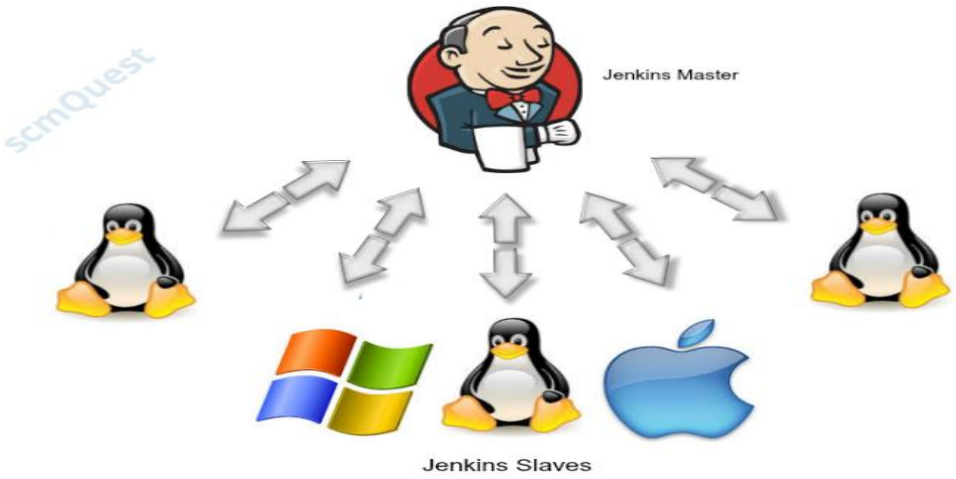
- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

A slave is a computer that is set up to offload build projects from the master and once connection is established between master and slave, tasks distribution is done automatic. Each slave runs a separate program called a “slave agent”. There is no need to install the full Jenkins on a slave. There are various ways to start slave agents, but in the end the “slave agent” and “Jenkins master” needs to establish a bi-directional communication link (for example a TCP/IP socket.) in order to operate.

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





Lab:

Create a new ec2 instance for slave:

Step 1: Create new directory for workspace area:

/home/ec2-user/jenkins-slave

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sun Feb 18 08:16:33 2018 from 103.79.101.173
[ec2-user@ip-172-31-23-107 ~]$ sudo su
[root@ip-172-31-23-107 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-23-107 ec2-user]# ls -ltr
total 0
[root@ip-172-31-23-107 ec2-user]# mkdir jenkins-slave
[root@ip-172-31-23-107 ec2-user]# cd jenkins-slave/
[root@ip-172-31-23-107 jenkins-slave]# pwd
/home/ec2-user/jenkins-slave
[root@ip-172-31-23-107 jenkins-slave]#
```

sudo yum install -y git java-1.8.0-openjdk-devel aws-cli

sudo alternatives --config java

(We need to install required build tool)

sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

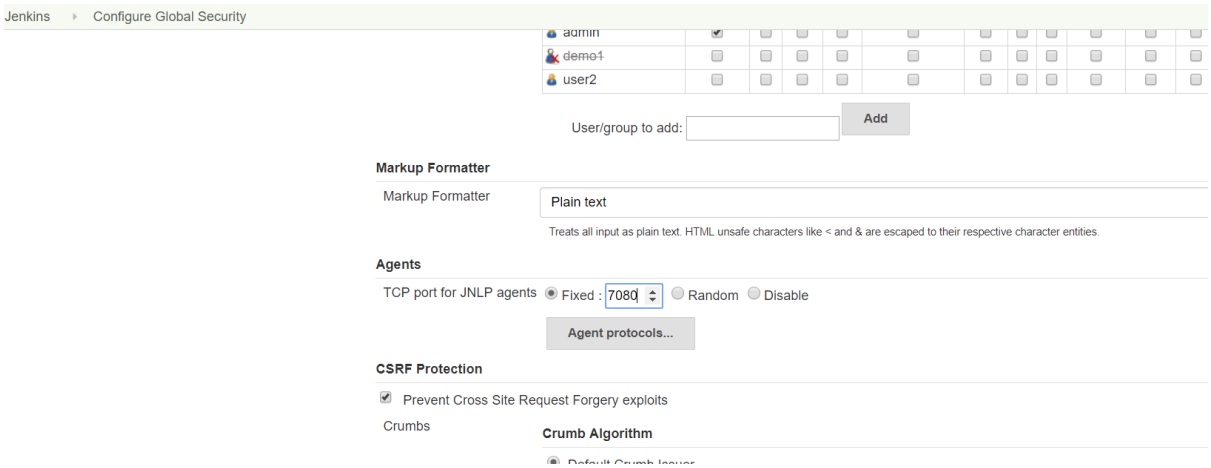
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo

yum install -y apache-maven

mvn --version

Step 2: Go to manage Jenkins -> configuration global security->in agent section provide any port number and save. This will ensure secure and encrypted communication between master and slave agent.

Java Network Launch Protocol (JNLP)

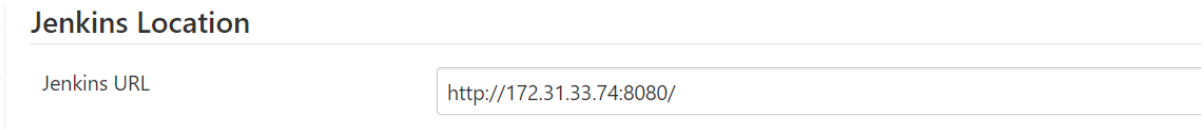


Step 3: Now we need to validate jenkins Ip under global configure:

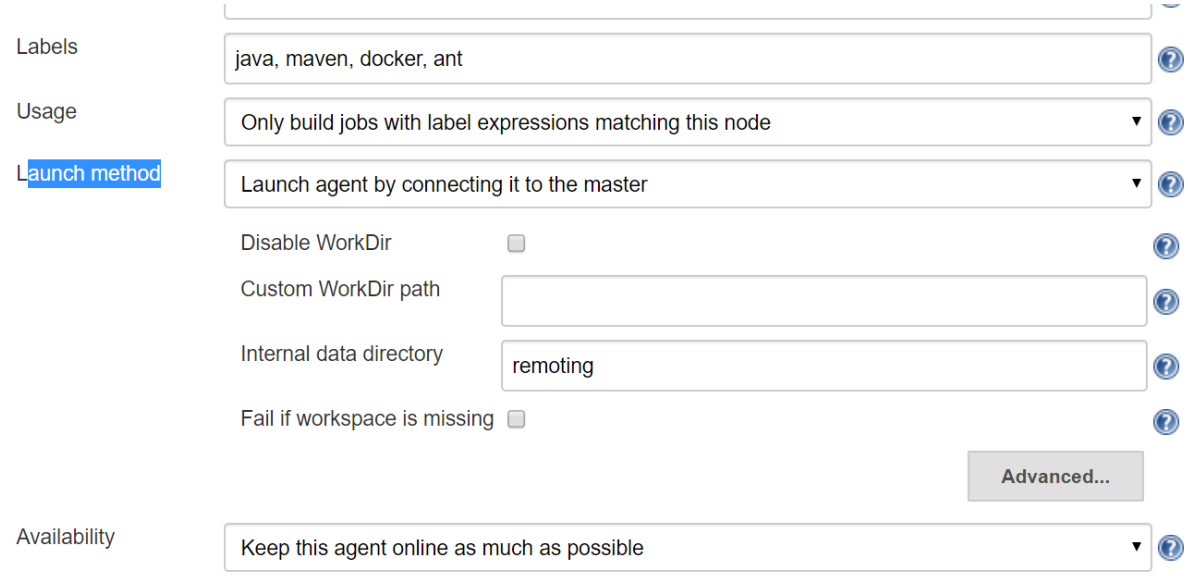
update jenkins server IP to Private IP

manage jenkins-> configure system->Jenkins URL -> update current private Jenkins IP

http://<private ip of jenkins>:8080



Step 4: Now go to manage Jenkins-manage node-go to the node→ select Launch agent by connecting it to the master



It will give you a output, copy output command and execute on slave machine.

For best practice you can create a file called startup.sh and paste script.

add **&** at the end of command (to run process in the back-ground)

NOTE: UPDATE current private IP of MASTER in SCRIPT

vi startup.sh

```
java -jar agent.jar -jnlpUrl http://172.31.33.74:8080/computer/slave-01/slave-agent.jnlp -secret 474412a4e4854a02a9d5e0f64e51612abbd8b9 ~
6c54eadbebc5881c59182d745a -workDir "/root/slave-workspace" &
~
```

And save. (Note: you can also paste output command without creating startup.sh file)

Change the permission and execute script.

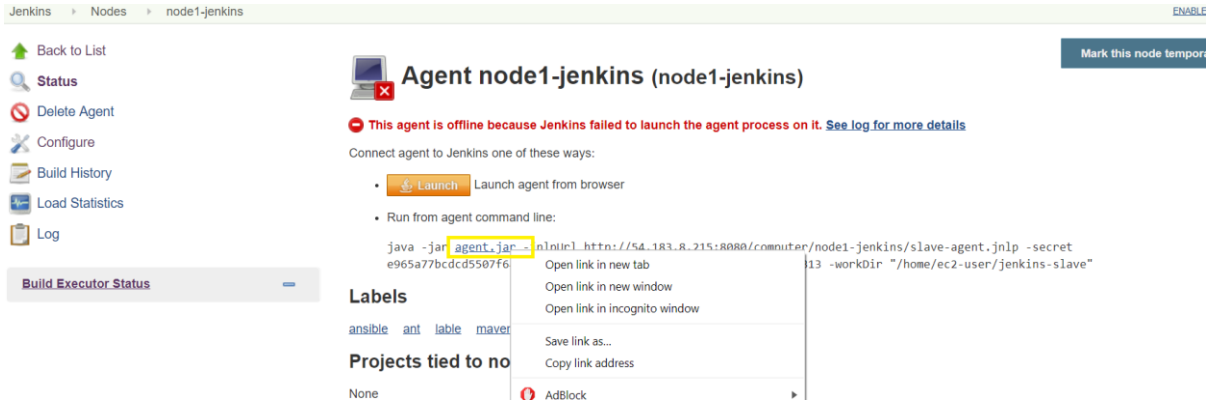
chmod u+x startup.sh

./startup.sh

```
/home/ec2-user/jenkins-slave
[root@ip-172-31-23-107 jenkins-slave]# vi startup.sh
[root@ip-172-31-23-107 jenkins-slave]# vi startup.sh
[root@ip-172-31-23-107 jenkins-slave]# chmod u+x startup.sh
[root@ip-172-31-23-107 jenkins-slave]# ls -lrt
total 4
-rwxr--r--. 1 root root 209 Feb 21 10:43 startup.sh
[root@ip-172-31-23-107 jenkins-slave]# ./startup.sh
[root@ip-172-31-23-107 jenkins-slave]# Error: Unable to access jarfile agent.jar
```

It will through you an error agent.jar is missing.

Now go to Manage node section, right click on agent.jar and copy link access. Go to your slave machine and do wget and paste (make sure you have wget package if not installed then install **yum install wget**)




```
[root@ip-172-31-23-107 jenkins-slave]# wget http://ec2-54-215-249-171.us-west-1.
compute.amazonaws.com:8080/jnlpJars/agent.jar
--2018-02-21 10:50:46--  http://ec2-54-215-249-171.us-west-1.compute.amazonaws.c
om:8080/jnlpJars/agent.jar
Resolving ec2-54-215-249-171.us-west-1.compute.amazonaws.com (ec2-54-215-249-171
.us-west-1.compute.amazonaws.com)... 172.31.15.124
Connecting to ec2-54-215-249-171.us-west-1.compute.amazonaws.com (ec2-54-215-249
-171.us-west-1.compute.amazonaws.com)|172.31.15.124|:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 745674 (728K) [application/java-archive]
Saving to: 'agent.jar'

100%[======================================>] 745,674     --.-K/s   in 0.009s

2018-02-21 10:50:46 (83.2 MB/s) - 'agent.jar' saved [745674/745674]

[root@ip-172-31-23-107 jenkins-slave]#
```

Once you execute script, you should be connected.

```
[root@ip-172-31-12-138 jenkin-slave]# wget http://54.153.33.194:8080/jnlpJars/ag
ent.jar
--2018-02-23 10:41:05--  http://54.153.33.194:8080/jnlpJars/agent.jar
Connecting to 54.153.33.194:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 745674 (728K) [application/java-archive]
Saving to: 'agent.jar'

agent.jar           100%[===================>] 728.20K  --.-KB/s    in 0.03s
```

If you get any error check for java version. Install java 8

(Note: Build tool should be there on Node for building your job)

And execute command **./startup.sh** (wait for some time, command is executing in backend)

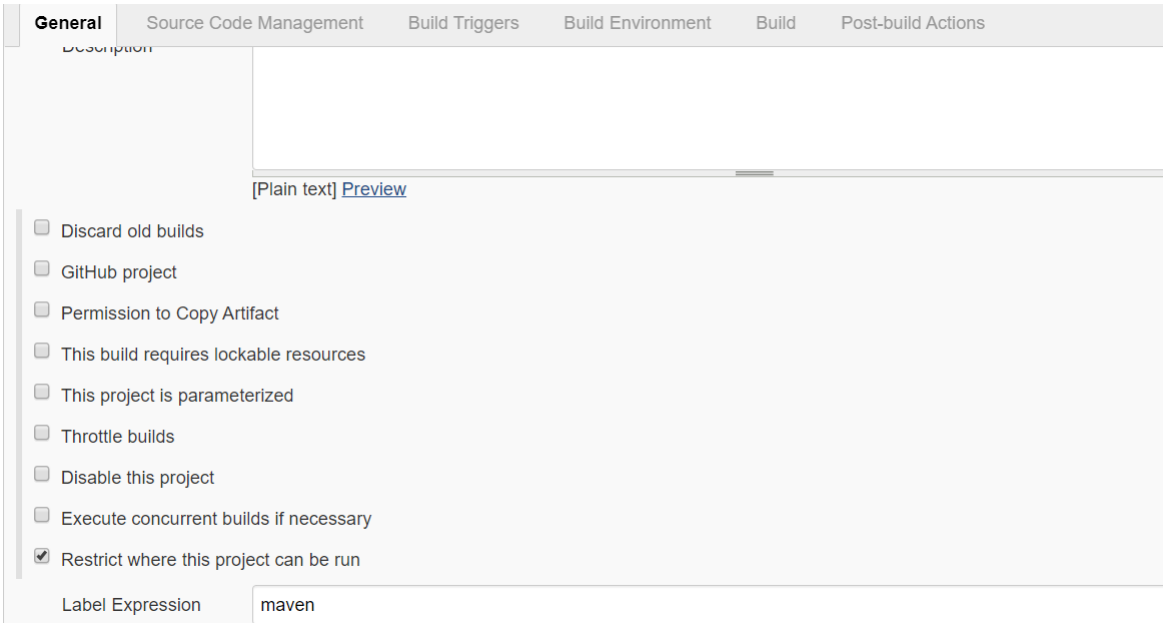
Step 5: Now configure any job. Under General section: Restrict where the project can be build

Choose label name

Build now

Check workspace on your slave.

Step 6: Create a CI-job (Master-slave-lab) ->Choose where this job can run -> Assign a Label to your job.



Step 7. Save and Build, Verify console output of job. Job should build on slave VM, Jenkins assign job to respective slave based on matched Label.

Done