Configure Jenkins Slaves

Prerequisites

1. Jenkins Master Running
2. Java v1.8.x
3. Security Group with Port 8080 open for internet

Step-1:

Install Java

```
 $ sudo apt-get update && apt-get upgrade
 $ sudo apt-get install default-jdk
 $ java -version
```

Installing Specific Versions of OpenJDK

```
 $ sudo apt install openjdk-8-jdk
```

Step-2:

Setup Jenkins Slave

```
# Create user and add the user to wheel group
  useradd jenkins-slave-01

# Create SSH Keys
  sudo su - jenkins-slave-01
  ssh-keygen -t rsa -N "" -f /home/jenkins-slave-01/.ssh/id_rsa
```

The private and public keys will be created at these locations `/home/jenkins-slave-01/.ssh/id_rsa` and `/home/jenkins-slave-01/.ssh/id_rsa.pub`

```
cd .ssh
cat id_rsa.pub > authorized_keys
chmod 700 authorized_keys
```

Step-3:

Configuration on Master

Copy the slave node's public key[id_rsa.pub] to Master Node's known_hosts file

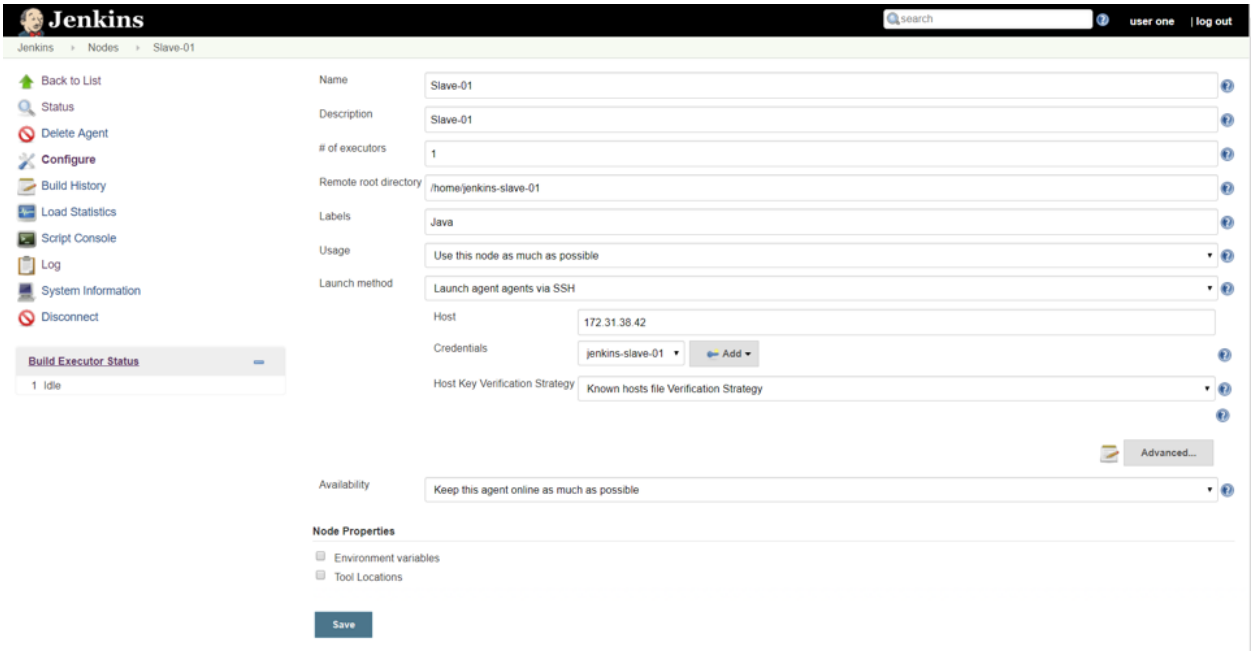
```
$ mkdir -p /var/lib/jenkins/.ssh
$ cd /var/lib/jenkins/.ssh
ssh-keyscan -H SLAVE-NODE-IP-OR-HOSTNAME >>/var/lib/jenkins/.ssh/known_hosts
```

```
# ssh-keyscan -H 172.31.38.42 >>/var/lib/jenkins/.ssh/known_hosts
$ chown jenkins:jenkins known_hosts
$ chmod 700 known_hosts
```

Step-4:

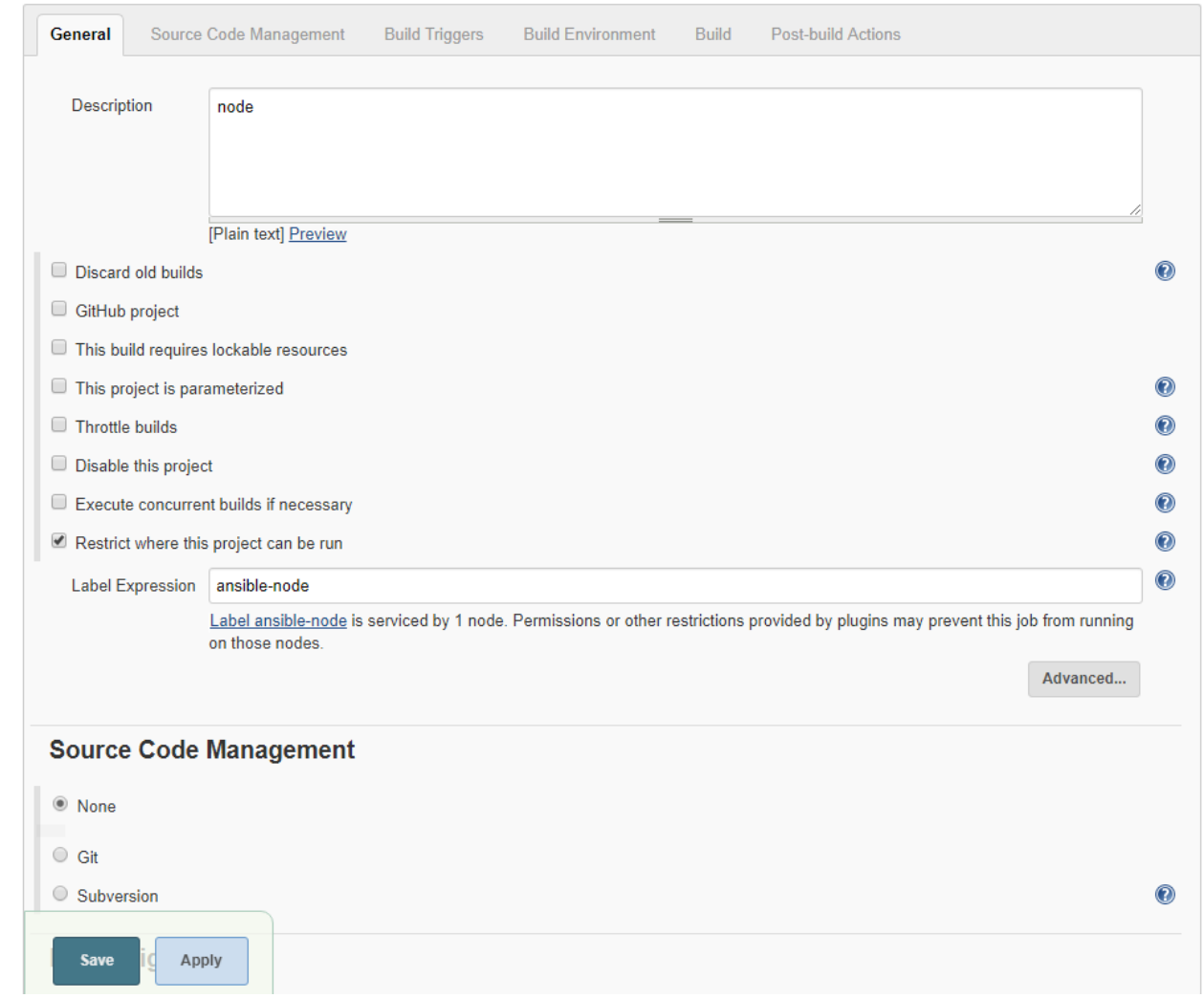
Configure the Slave using **Manage Jenkins**.

Configure the node as shown here `Manage Jenkins > Manage Nodes > New Node`

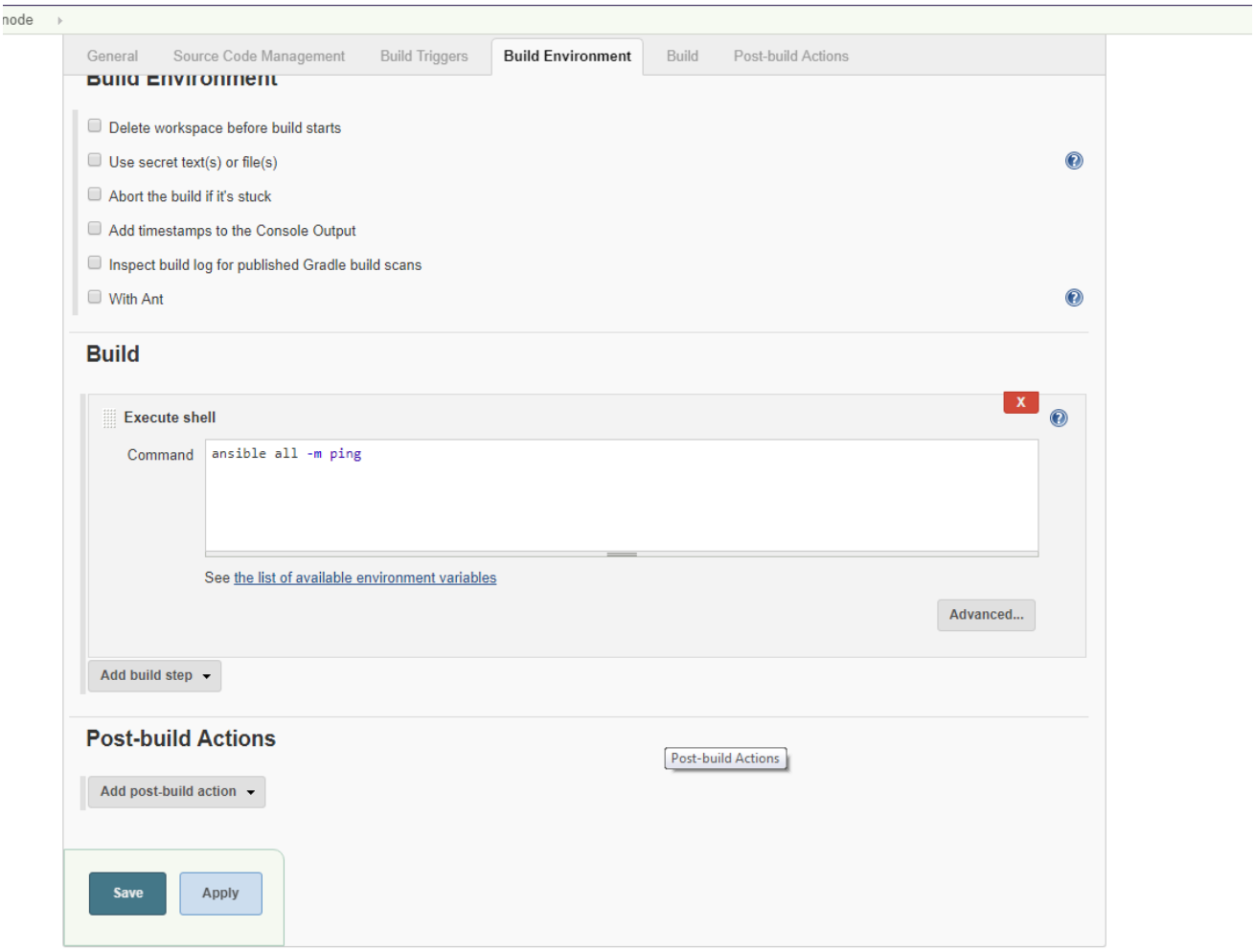


Test Jenkins Jobs

1. Create “new item”
2. Enter an item name – `My-First-Project`
    - Chose `Freestyle` project
3. Under `General` Section
    - Choose `Restrict where this project can be run`
        - Update Lebel Expression with name of your slave.



4. Under Build section Execute shell write a command.



5. Save your job
6. Build job
7. Check "console output"

For the pipeline project you can you can select node slave node in the agent section.

Eg.

```
pipeline {
   agent none
  stages {
    stage('Install npm Packages') {
        agent { label 'ansible-node'  }
        steps { sh "ansible-playbook test.yml"}
             }
   stage('check') {
        agent { label 'master'}
      steps {sh "ls /home/"   }
           }
  }
}
```

You can also assign agent name for all the stages or for the specific stage. The above example show that both stage execute command on the two different agent one is master and second one is ansible-node(another agent).

You can assign agent name at the top to execute all the command on the assign user.

Eg.

```
pipeline {
   agent { label 'ansible-node'  }
  stages {
    stage('Install npm Packages') {
        steps { sh "ansible-playbook test.yml"}
             }
   stage('check') {
      steps {sh "ls /home/"   }
           }
  } }
```