Jenkins - Master Slave Configuration

**Assumption**

1. Created 2 instances of centos in AWS. Out of which one can be accessed over the web.
2. Switch to 'root' user.

**Master**

Tools to be installed in Master

1. JDK (jdk1.8.0\_201) - To install JDK follow the below steps
   1. yum install java -y
   2. Download the JDK File in windows/local machine
   3. copy the same to centos environment through WinScp
   4. move the file to '/opt' folder (when copying the file would be in /home/centos folder)
   5. tar xvf jdk-8u201-linux-x64.tar.gz - unzip the contents
   6. Execute the following command to have JDK installed in the machine
      1. sudo alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_201/bin/jar 1
      2. sudo alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_201/bin/javac 1
      3. sudo alternatives --set jar /opt/jdk1.8.0\_201/bin/jar
      4. sudo alternatives --set javac /opt/jdk1.8.0\_201/bin/javac
   7. vi /etc/profile.d/java.sh - Create this file and enter the following
      1. export JRE\_HOME=/opt/jdk1.8.0\_201/jre
      2. export PATH=$PATH:$JRE\_HOME/bin
      3. export JAVA\_HOME=/opt/jdk1.8.0\_201
      4. export JAVA\_PATH=$JAVA\_HOME
      5. export PATH=$PATH:$JAVA\_HOME/bin
   8. source /etc/profile.d/java.sh - Execute this file to have the environment variables created.
2. Maven - To install Maven follow the below steps
   1. cd /usr/local/src/
   2. wget http://mirrors.estointernet.in/apache/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz - To download the binary file from portal
   3. tar -xf apache-maven-3.6.0-bin.tar.gz - unzip the contents
   4. rm apache-maven-3.6.0-bin.tar.gz - remove the zip file as it would keep occupying the space.
   5. mv apache-maven-3.6.0/ apache-maven - Rename the folder as this would be easy to remember
   6. vi /etc/profile.d/maven.sh - create and open up this file and enter the following
      1. # Apache Maven Environment Variables
      2. # MAVEN\_HOME for Maven 1 - M2\_HOME for Maven 2
      3. export M2\_HOME=/usr/local/src/apache-maven
      4. export PATH=${M2\_HOME}/bin:${PATH}
   7. source /etc/profile.d/maven.sh - Execute this file to get the environment variables in place.
   8. mvn --version - To verify the Maven installation

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |

1. Git - Git installation is simple and straight forward
   1. sudo yum install git
   2. which git - this would give the path where git has been installed.
2. Jenkins
   1. wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo
   2. sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
   3. sudo yum install jenkins
   4. Given below is the statements used to get the jenkins installed in centos

Jenkins Configuration (This is the general Jenkins configuration for Centos instances)

Manage Jenkins -> Global Tool configuration

1. JDK - This path can be found from /etc/profile.d/java.sh

Name: Java

Path: /opt/jdk1.8.0\_201/

2. GIT - This path can be found using the command 'which git'

Name: GIT

Path: /usr/bin/git

3. Maven

Name: Maven - This path can be found from /etc/profile.d/maven.sh

Path: /usr/local/src/apache-maven

**Slave**

Tools to be installed in Slave - Please follow the steps as mentioned above in the master setup

1. JDK
2. Maven
3. Git

Slave Configuration Steps

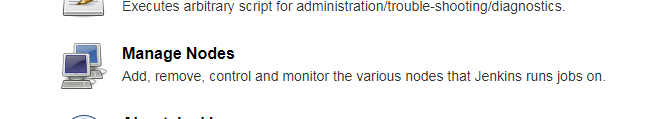
1. useradd -d /var/lib/jenkins jenkins: Create a user called jenkins which will be used in UI to connect to slave from master
2. ssh-keygen : create a secured key
3. cat /root/.ssh/id\_rsa.pub &
4. mkdir /var/lib/jenkins/.ssh
5. vi /var/lib/jenkins/.ssh/authorized\_keys
6. Copy the displayed Key and paste it in the new file 'authorized\_keys'
7. chown -R jenkins:jenkins /var/lib/jenkins/.ssh: Modify the file with read access

Master Configuration

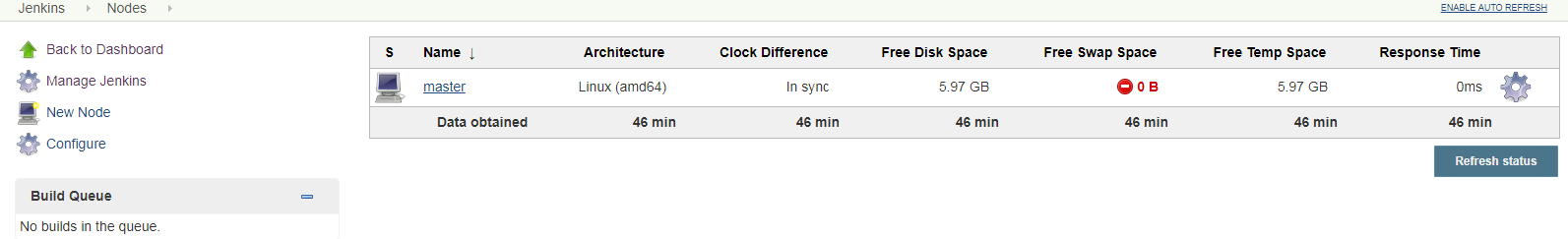
1. Click 'Manage Jenkins'



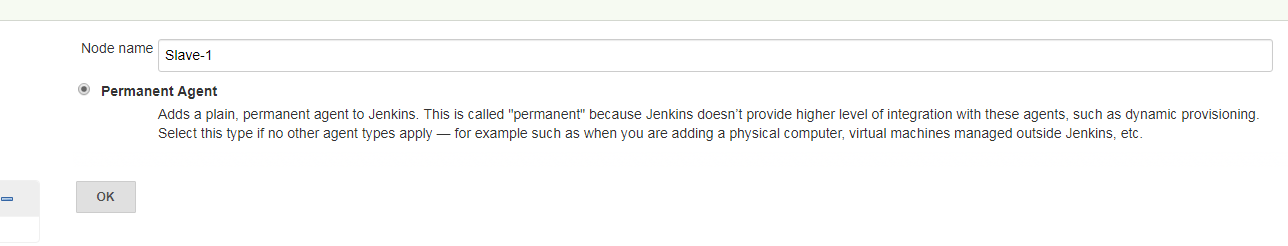
1. Click 'Manage Node'



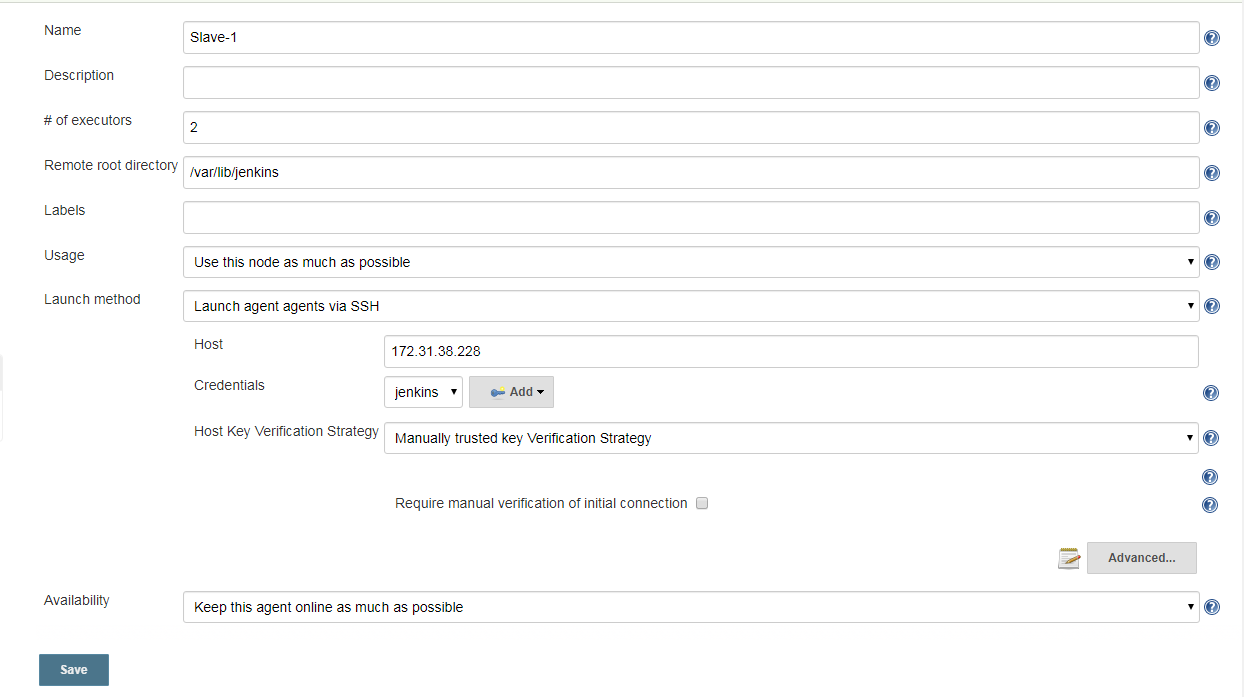
1. Click on 'New Node'



1. Enter a 'Node Name' and Select 'Permanent Agent' and click 'Ok'
   1. Node Name: Slave-1
   2. Permanent Agent : To be selected



1. Enter the details as follows in the appropriate title
   1. # of Executors: 2
   2. Remote root directory: /var/lib/jenkins
   3. Usage: Use this node as much as possible
   4. Host: <client ip>

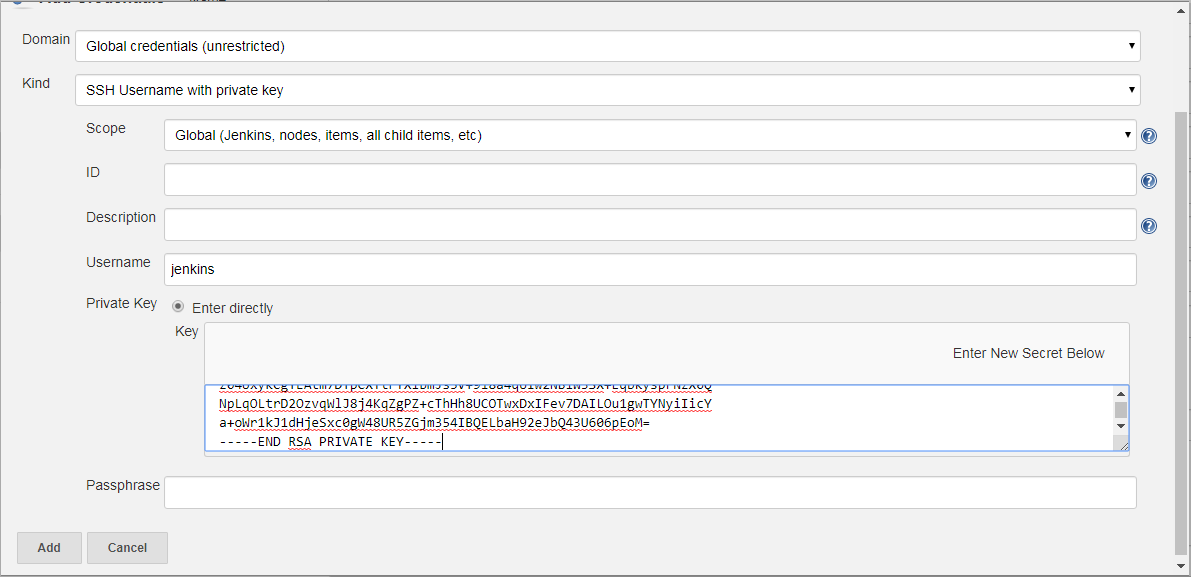


Credentials: Click on 'Add' and a popup will open up for the further information.

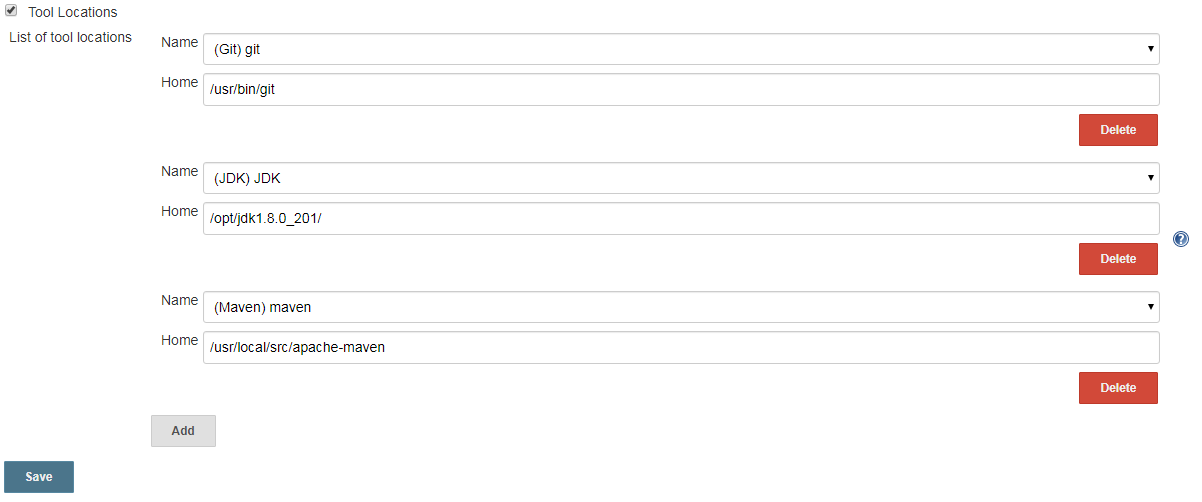
Kind: 'SSH Username with private key'

User:<username created in the slave. In this case it is 'jenkins>

Private Key:Choose 'Enter Directly' and paste the contents from id\_rsa file in ' /root/.ssh' path in slave machine



1. Tools - Follow the configuration paths mentioned in the 'Global Tool Configuration' in this document
   1. Git
   2. JDK
   3. Maven



Building a Project to run in Slave: With the above setup the Slave should be up and ready to run a new job. Now to configure a job to be executed in Slave just one additional configuration to be done during the creation of the job as mentioned in the below screenshot.

Enable 'Restrict where this project can be run' option and enter the slave name.

