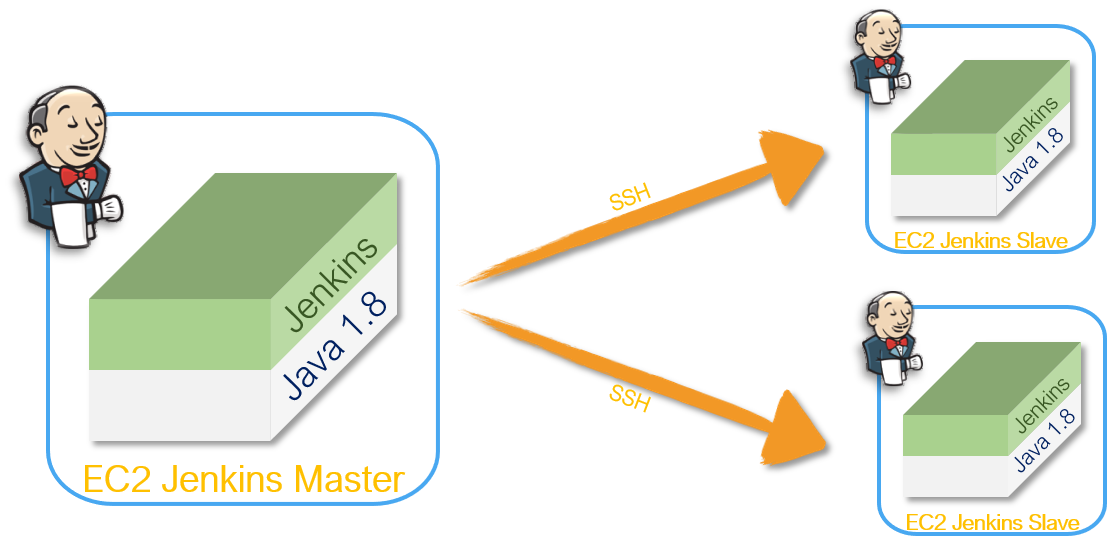
**Configure Jenkins Slaves on AWS EC2**

Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Mac OS X and other Unix-like operating systems. As an extensible automation server, Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.

**Follow this article in [Youtube](https://youtu.be/hwrYURP4O2k)**

[](https://raw.githubusercontent.com/miztiik/DevOps-Demos/master/setup-jenkins-slave/images/Jenkins%20Master%20and%20Slave%20Configuration.png)

**Prerequisites**

1. **Jenkins Master** Running [Get help here](https://youtu.be/-0dkiteJEuE)
2. EC2 RHEL 7.x Instance - *for* ***Slave Node*** [Get help here](https://www.youtube.com/watch?v=KDtS6BzJo3A)
   * With Internet Access
   * Security Group with Port 8080 open for internet
   * Java v1.8.x

**Install Java**

We will be using open java for our demo, Get latest version from <http://openjdk.java.net/install/>. Also configure the default **JAVA\_HOME** path

yum install java-1.8\*

#yum -y install java-1.8.0-openjdk

|  |
| --- |
|  |
|  |  |
|  |  |
|  | 2) java -version |
|  | 3) find /usr/lib/jvm/java-1.8\* | head -n 3 |
|  | #JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64 |
|  | # To set it permanently update your .bash\_profile |
|  | 5) vi .bash\_profile |
|  | Sample Example of .bahs\_profile file: |
|  | ==================================== |
|  | # .bash\_profile |
|  |  |
|  | # Get the aliases and functions |
|  | if [ -f ~/.bashrc ]; then |
|  | . ~/.bashrc |
|  | fi |
|  |  |
|  | # User specific environment and startup programs |
|  | JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-2.el8\_1.x86\_64 |
|  | PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin |
|  |  |
|  | export PATH |
|  | ========================================================================= |
|  | #Restart the .bash\_profile to reflect the changes. |
|  |  |
|  | 6) source ~/.bash\_profile |

**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

**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

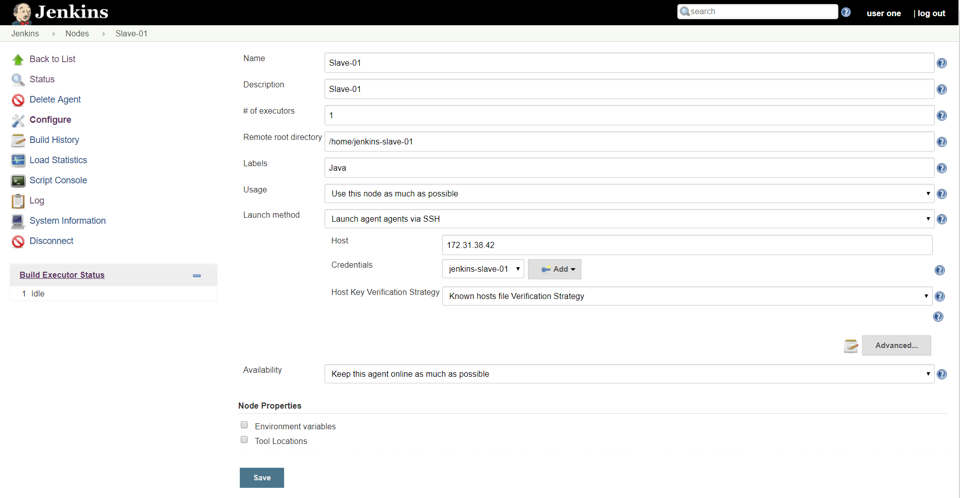
# ssh-keyscan -H 172.31.38.42 >>/var/lib/jenkins/.ssh/known\_hosts

chown jenkins:jenkins known\_hosts  
#we need to change the owner as we ran ssh-keyscan command using “root” user.

# default user of Jenkins will be “jenkins”

chmod 700 known\_hosts

**Configure the Slave using Manage Jenkins**

Configure the node as shown here Manage Jenkins > Manage Nodes > New Node [](https://raw.githubusercontent.com/miztiik/DevOps-Demos/master/setup-jenkins-slave/images/Slave-Node-Configuration-01.png)

**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 your *jenkins slave label* Java
4. Under Build section Execute shell
5. #!/bin/bash
6. echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"
7. echo "| |"
8. echo "| Welcome to Valaxy Demo |"
9. echo "| \_nnnn\_ |"
10. echo "| dGGGGMMb |"
11. echo "| @p~qp~~qMb |"
12. echo "| M|@||@) M| |"
13. echo "| @,----.JM| |"
14. echo "| JS^\\_\_/ qKL |"
15. echo "| dZP qKRb |"
16. echo "| dZP qKKb |"
17. echo "| fZP SMMb |"
18. echo "| HZM MMMM |"
19. echo "| FqM MMMM |"
20. echo "| \_\_| '. |\dS'qML |"
21. echo "| | '. | ' \Zq |"
22. echo "| \_) \.\_\_\_.,| .' |"
23. echo "| \\_\_\_\_ )MMMMMP| .' |"
24. echo "| '-' '--' hjm |"

echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

1. Save your job
2. Build job
3. Check "console output"