**Jenkins Master and Slave configuration**

**Master Server**

**Jenkins Installation on Linux (Ubuntu)**

* Login to AWS Console
* Create EC2 Instance with Ubuntu AMI
* Login to EC2 Instance from Git Bash or any other terminal
* Execute below Commands
* This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system:

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

* Then add a Jenkins apt repository entry:

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null

* Update your local package index, then finally install Jenkins:

sudo apt-get update -y

sudo apt-get install fontconfig openjdk-11-jre -y

* Install Jenkins

sudo apt-get install jenkins -y

* Start the Jenkins

sudo service jenkins start

* Install Git and Maven

sudo apt-get install git maven -y

* Add port no 8080 in Security group 🡪 Inbound Rules
* Open Browser and type
  + <publicip>:8080
* Jenkins will be opened in a browser
  + Unlock the Jenkins by following the instructions

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

* + - copy the password and paste
  + Install suggested Plugins
  + Create admin user
  + Then your Jenkins is ready

Once Jenkins server is successfully configured then click on **Manage Jenkins**.

11. Then move to **Configure Global Security**.

12. Select option **Random** for the settings of **TCP port for inbound agents** under Agents And save the settings.

13. Now go to **Manage Nodes and Clouds** section under **Manage Jenkins**.

14. Click on **New Node** and give the name of the node (ex. Slave-1) and select the **Permanent Agent** option and click on OK

15. Now in the node configuration in the **Remote root directory** give a valid path of the slave machine. At the time of executions workspace will be created inside this directory (ex. /home/ec2-user/Jenkins).

16. Select the **launch method** as **Launch agent by connecting it to the master**.

17. Click on Save and that’s it. Jenkins master configuration is done. Now you can see the list of nodes like the following. There will red cross mark for the slave node as the node has not been configured yet.

18. Click on the slave node. There you will get the link to download two important files which will be required for configuring the slave. 1. agent.jar, 2. Slave-agent.jnlp

 Also note down the first command mentioned under **Run from agent command line** section. For example it looks like the one below

curl -sO <http://54.86.42.181:8080/jnlpJars/agent.jar>

java -jar [agent.jar](http://54.86.42.181:8080/jnlpJars/agent.jar) -jnlpUrl http://54.86.42.181:8080/computer/Slave%2D1/jenkins-agent.jnlp -secret 527c990054a730f8771d3d9b7e1c42123ee5850cd55fc690228adfb4b3ec1c99 -workDir "/home/ec2-user/jenkins"

**Configure Jenkins slave -**

1. Launch the second EC2 instance.
2. Install Java

sudo yum install java-11-openjdk

sudo yum install git maven -y

1. Navigate to directory in slave machine where files are copied and run the command copied in the previous section. This will configure the instance as Jenkins slave machine and will show the message **Connected**.

Add the jnlp port to Security group inbound rules

Create the Job in Master server

To execute a job in the slave, you have to configure the the slave machine in the job settings. Enter the node name in the **Restrict where this project can be run** section.