**Jenkins pipeline setup with Slaves (Dev,Test,Production) & triggering the email notifications**

**Contents:**

-what is jenkins

**-**installing java

-installing Jenkins

-jenkins pipeline setup with Slaves(Dev,Test,Production)

-triggering email notification

**What is jenkins:**

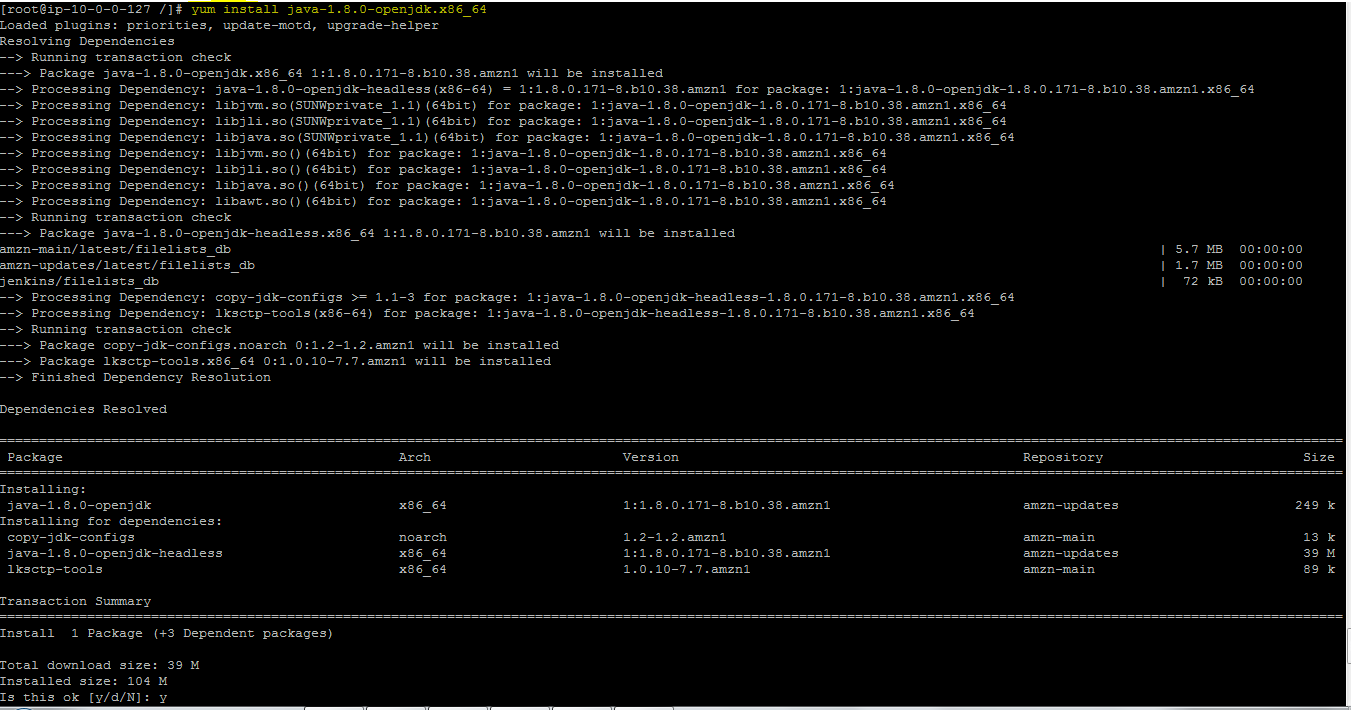
***Jenkins*** is a **Continuous Integration (CI) server** or tool which is written in java. It provides Continuous Integration services for software development, which can be started via command line or web application server. And also, it is happy to know that **Jenkins** is free software to download and install.

**Advantages of Jenkins**

* **Jenkins** is an open source tool with much support from its community.
* Installation is easier.
* It has more than 1000 plug-in to make the work easier.
* It is easy to create new Jenkins plugin if one is not available.
* It is a tool which is written in Java. Hence it can be portable to almost all major platforms.

**installing java:**

java is a prerequisite for installing Jenkins.Use **yum search java** command to see the available java versions and install the latest version using “**yum install java-1.8.0-openjdk-devel”.**I aminstalling java 1.8.



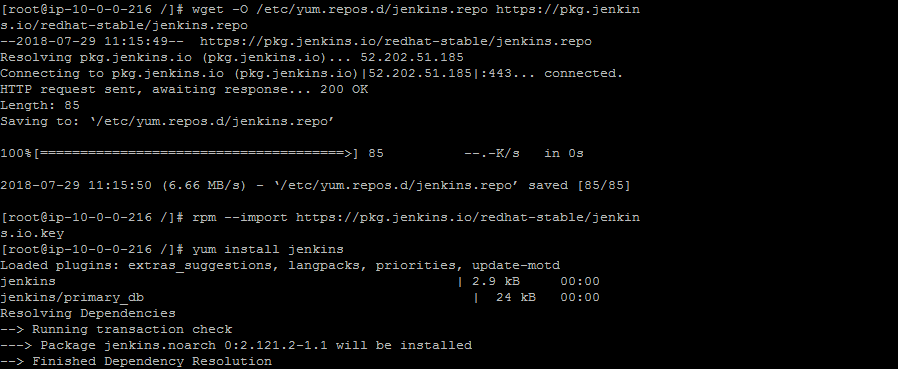
**installing Jenkins(linux):**

**prerequisite for installing jenkins:**

java,tomcat or apache webserver

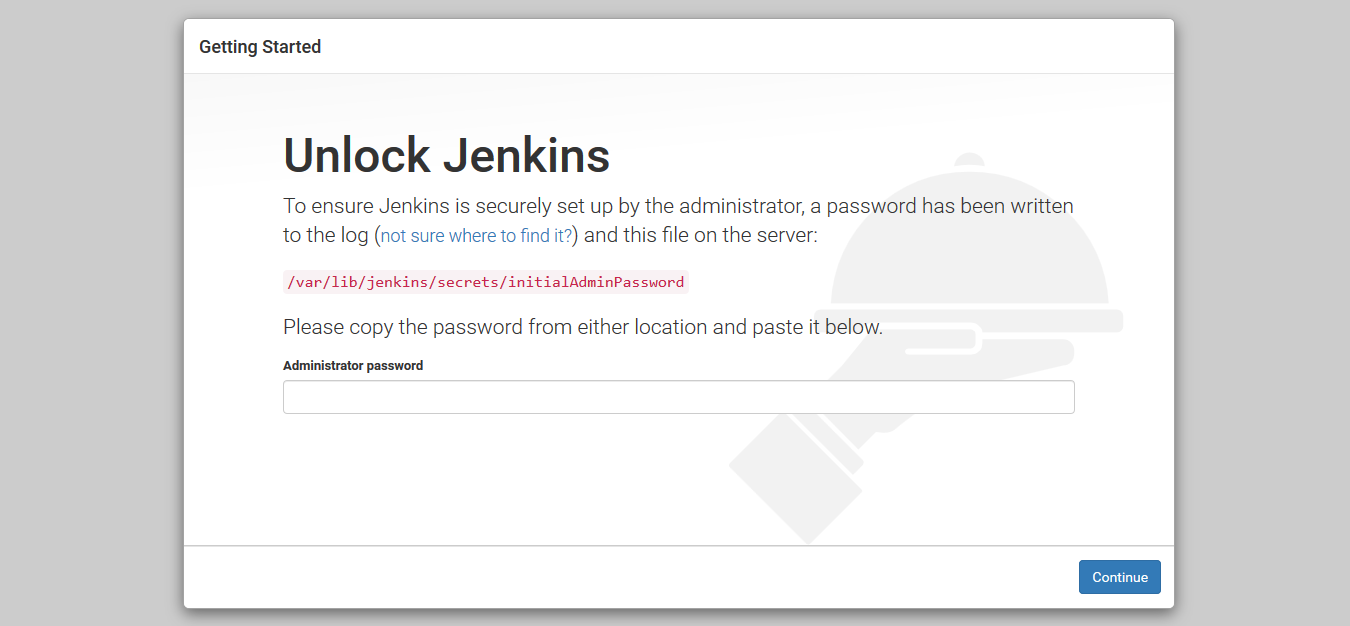
Add the Jenkins repository to the yum repos, and install Jenkins from here.use the following commands to install Jenkins.

* sudo wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat/jenkins.repo>
* sudo rpm --import [https://jenkins-ci.org/redhat/jenkins-ci.org.key](http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key)
* sudo yum install jenkins



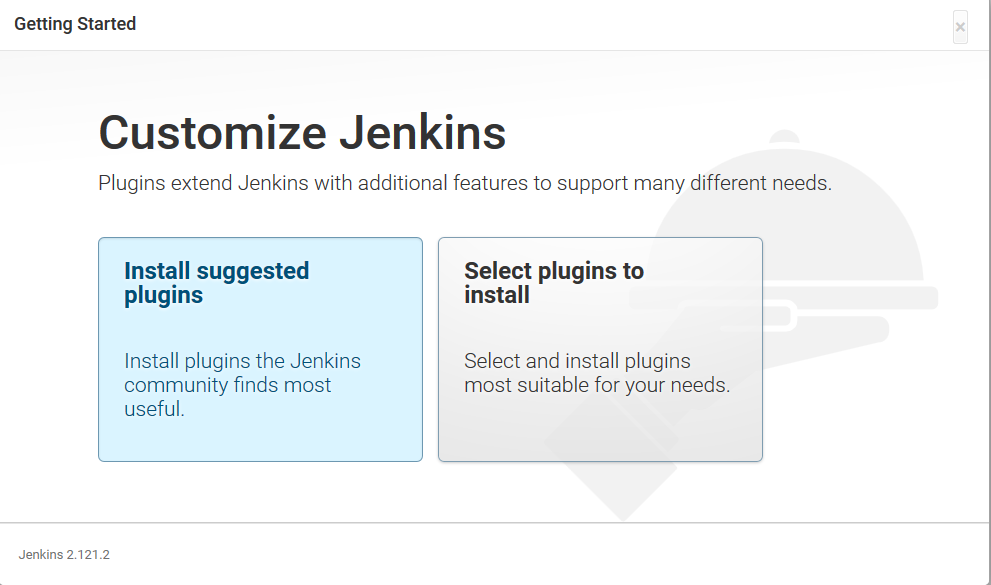
Use public ip address:8080 to start Jenkins server and work with jenkins.my public ip is 34.234.167.10: so I am using <http://34.234.167.10:8080>.

It will prompt for unlock Jenkins password as shown below.



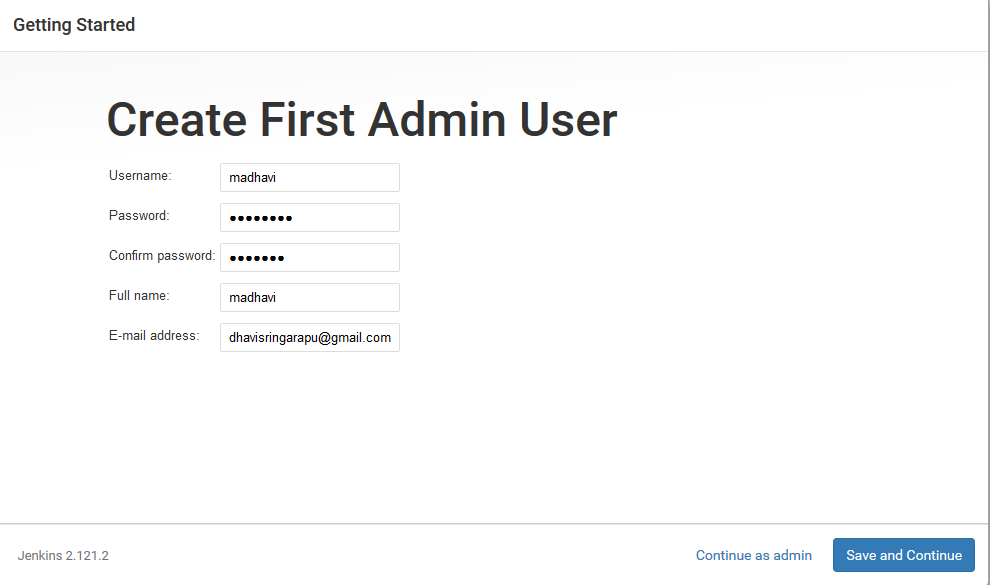
Copy the initialAdminPassword present in var/lib/Jenkins/secrets/initialAdminPassword present in Jenkins server and paste into in the password box and continue.

Now it will prompt for customize Jenkins choose install suggested plugins.

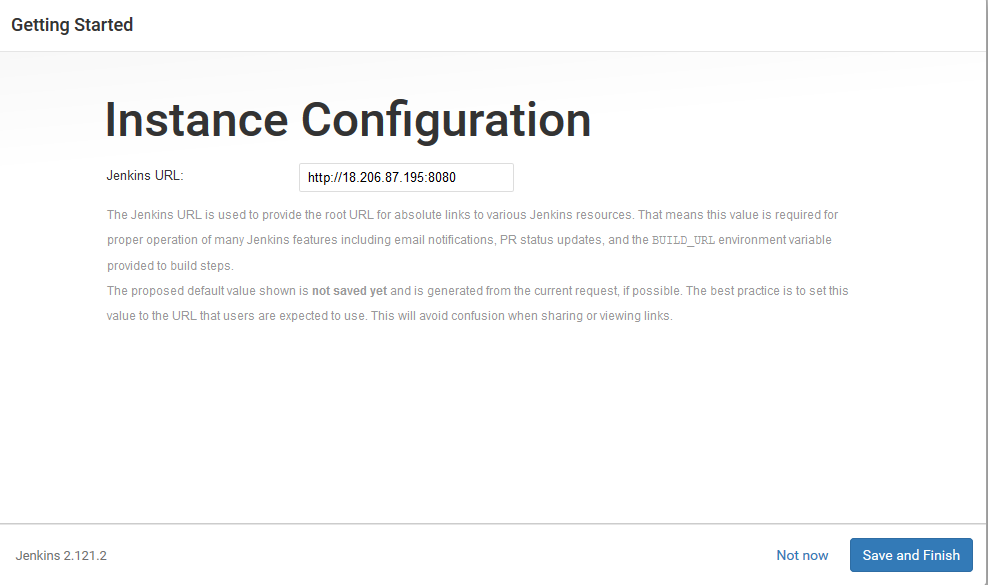


Install suggested plugins will install the useful plugins,if you want to select specific plugins you can choose select plugin to install option.

After this step it will prompt for creating a user .provide the user details like username,password and email address and click on save and continue.



After that click on save and finish.



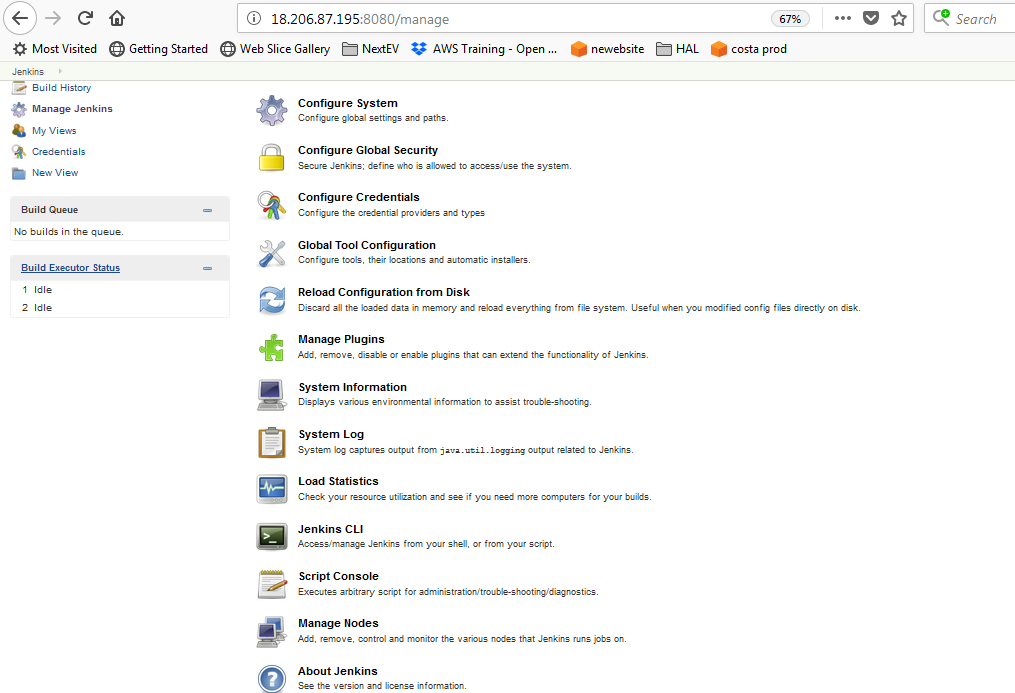
jenkins pipeline setup with Slave(Dev,Test,Production):

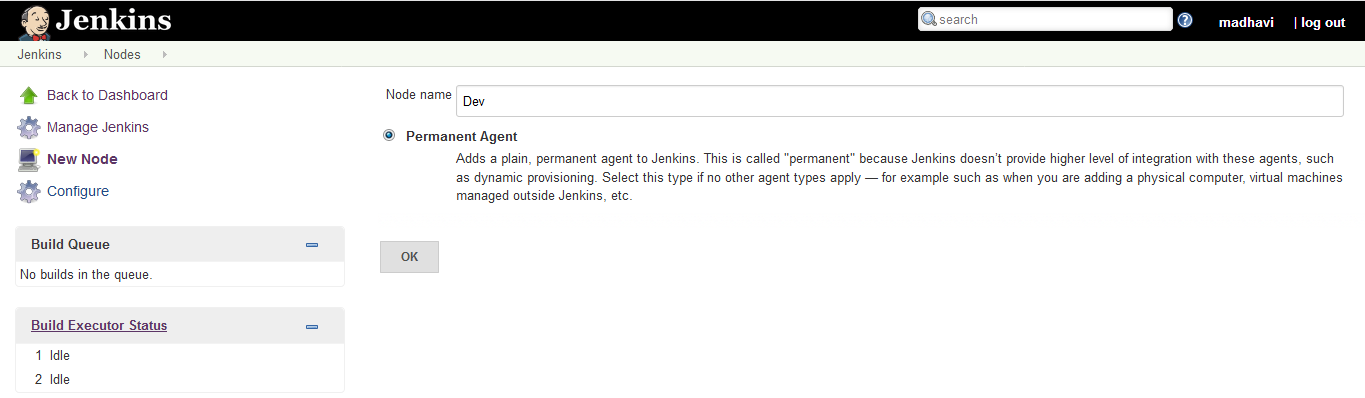
Following are the characteristics of Jenkins Slaves:

* 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.

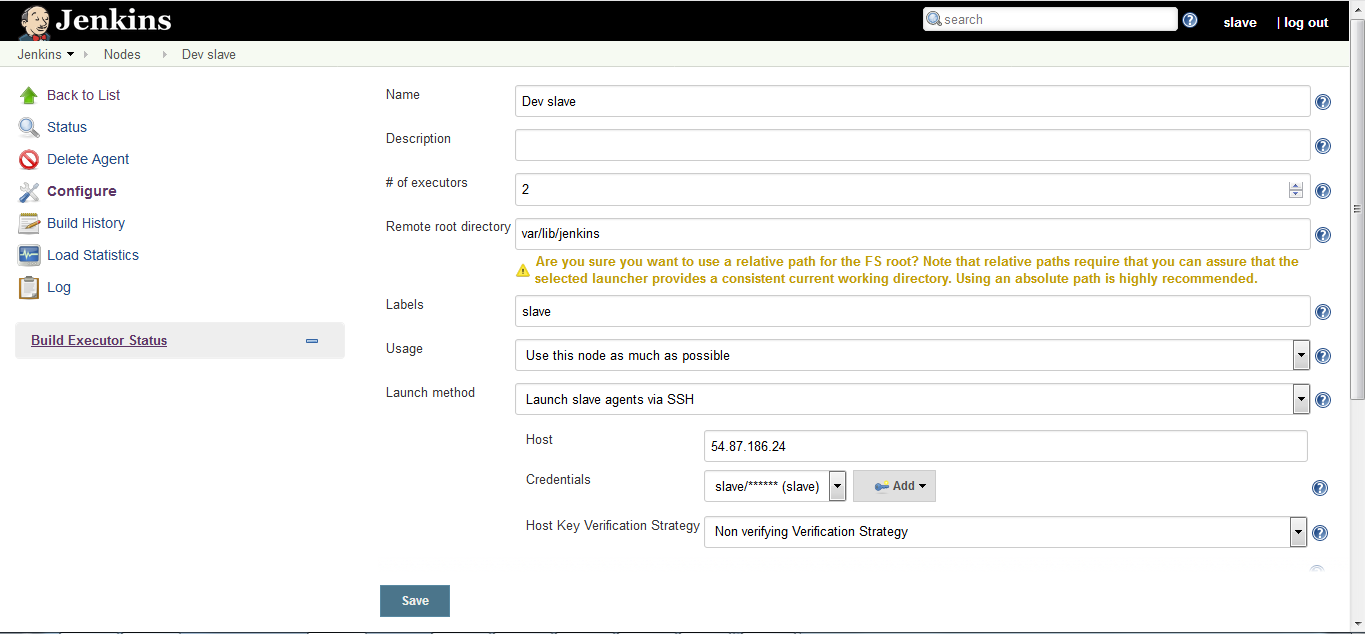
In this I am adding 3 slaves Dev,test and Production to jenkins master.

* Go to **Jenkins dashboard** -> **Manage Jenkins** -> **Manage Nodes.**
* Select **new node** option.





* Give it a name, select the “**permanent agent**” option and click**OK**.
* Enter the details as shown in the image below and save it. For credential box, click the add button and enter the slaves Jenkins username and password (For logging into the slave machine) that created during the slave configuration. To know what each option means, click the question mark at the right side of each text box.



 In above Diagram :**Name** : Name that uniquely identifies an agent within this Jenkins installation.  
**Description** : Optional ,description for this agent.  
**# of executors** : The maximum number of concurrent builds that Jenkins may perform on this agent.  
**Remote root directory** : An agent needs to have a directory dedicated to Jenkins. Specify the path to this directory on the agent. It is best to use an absolute path, such as /var/jenkins or c:\jenkins. This should be a path local to the agent machine. There is no need for this path to be visible from the master.  
**Labels** : Labels (or tags) are used to group multiple agents into one logical group.**Usage** : Controls how Jenkins schedules builds on this node.Use this node as much as possible only build jobs with label expressions matching this node(select if above label field is set.)

**Launch method** : It Controls how Jenkins starts this agent.  
**Launch agent via Java Web Start** :  
Allows an agent to be launched using Java Web Start. In this case, a JNLP file must be opened on the agent machine, which will establish a TCP connection to the Jenkins master. This means that the agent need not be reachable from the master,the agent just needs to be able to reach the master. By default, the JNLP agent will launch a GUI, but it’s also possible to run a JNLP agent without a GUI, e.g. as a Window service.  
Launch agent via execution of command on the master :  
Starts an agent by having Jenkins execute a command from the master. Use this when the master is capable of remotely executing a process on another machine, e.g. via SSH or RSH**.**  
**Launch slave agents via SSH**:  
Starts a slave by sending commands over a secure SSH connection. The slave needs to be reachable from the master, and master will have to supply an account that can log in on the target machine. No root privileges are required.  
**Let Jenkins control this Windows slave as a Windows service**:  
In case of windows server as slave. Please use this option. Please do it in your own risk as it is not tested in this process.  
Start and stop this node on-demand (Need slave setup plugin):  
Wrapper for other launch methods: executes a script (i.e. to provision a VM) before attempting to connect to the slave machine  then launches the Jenkins slave with a launch method of user choice, and after disconnection executes another script (i.e. for shutting down the VM).  
**Availability** : Controls when Jenkins starts and stops this agent. Keep this agent online as much as possible.  
Take this agent online and offline at specific times :

In this we set “Startup Schedule” and “Scheduled Up time”

**Startup Schedule** : schedule cron job to start and stop node.  
Examples:  
every fifteen minutes : H/15 \* \* \* \*  
**Scheduled Up time** : The number of minutes to keep the node up for. If this is longer than the startup schedule, then the node will remain constantly on-line.Take this agent online when in demand, and offline when idle

Click on save button to create new new node  and after that  Jenkins will automatically connect to the slave machine.

**Steps to prepare Slave Server :**

1. Go to  Manage Jenkins->Configure System and check the “Slave Setups” block and click on add button.
2. **pre-launch script** :  
   After configuring the slave on Jenkins master ,slave should be launch from Jenkins master to execute build on slave . Pre launch script is executed before Jenkins-master launch the slave.  
   **Steps to install dependencies on slave using pre-launch script :**Create shell script on $JENKINS\_HOME(i.e. /var/lib/jenkins/). Set that shell script path inpre-launch script. **Example** : ${JENKINS\_HOME}/prelaunch\_dev.sh .  
   Go to Manage Jenkins->Manage Nodes->Click on the node name-> Click on “Launch Agent” button.  
   After clicking on “Launch Agent” Button ,Pre-launch script will be executed
3. **prepare script:**Specify the command to be executed on the master before Jenkins starts using the slave. Follow the steps same as “pre-launch script” and specify the prepare script path in front of “prepare script” in below image.
4. **Label Expression :**In master slave configuration label is used to prepare build on slave and Configure the slave.In slave setup “label expression” is used to identify the slave Node to install dependencies.

Add 2 more nodes one is for test and one is for production using the above process.

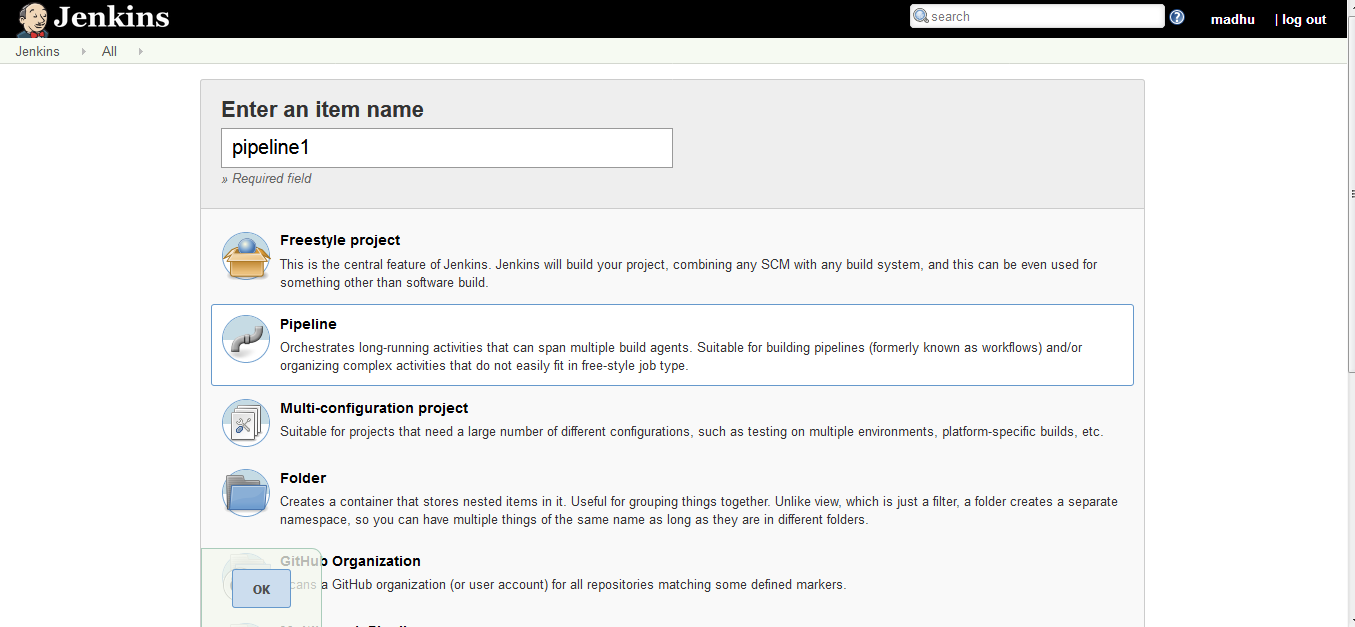
## Creating a Simple Pipeline

Initial pipeline usage typically involves the following tasks:

1. Downloading and installing the Pipeline plugin (Unless it is already part of your Jenkins installation):install pipeline plugin.
2. Creating a Pipeline of a specific type
3. Configuring your Pipeline
4. Controlling Flow through your Pipeline
5. Scaling your Pipeline

To create a simple pipeline from the Jenkins interface, perform the following steps:

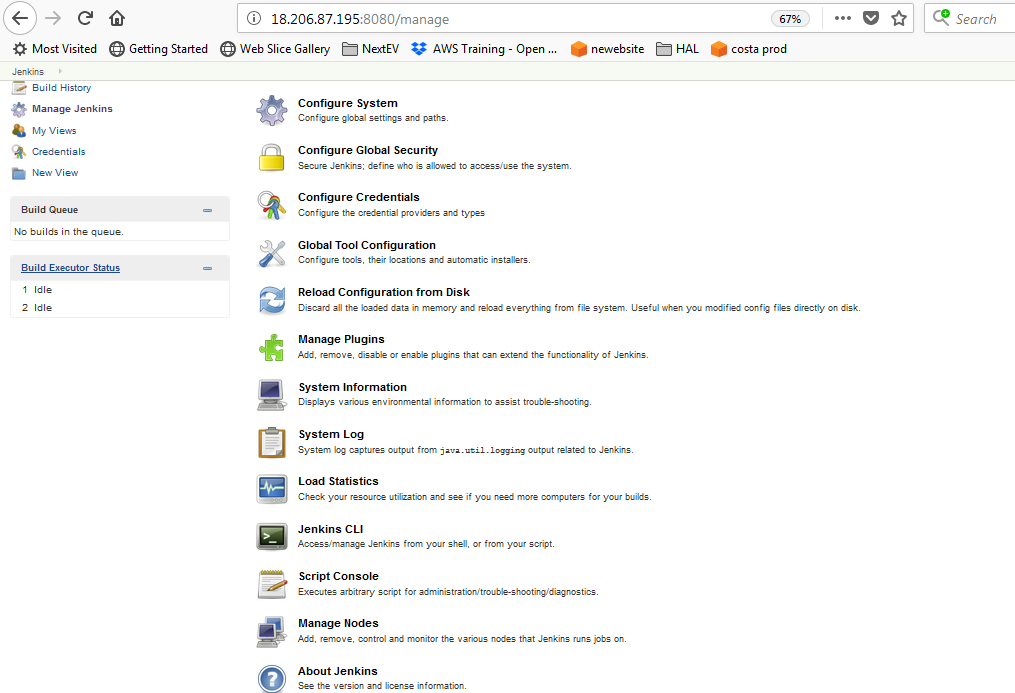
1. Click **New Item** on your Jenkins home page, enter a name for your (pipeline) job, select **Pipeline**, and click **OK**.
2. In the Script text area of the configuration screen, enter your pipeline syntax. If you are new to pipeline creation, you might want to start by opening Snippet Generator and selecting the “Hello Word” snippet. **Note:** Pipelines are written as Groovy scripts that tell Jenkins what to do when they are run, but because relevant bits of syntax are introduced as needed, you do not need deep expertise in Groovy to create them, although basic understanding of Groovy is helpful.
3. Check the Use Groovy Sandbox option below the Script text area. **Note:** If you are a Jenkins administrator (in other words, authorized to approve your own scripts), sandboxing is optional but efficient, because it lets scripts run without approval as long as they limit themselves to operations that Jenkins considers inherently safe.
4. Click **Save**.
5. Click **Build Now** to create the pipeline.
6. Click ▾ and select **Console Output** to see the output.



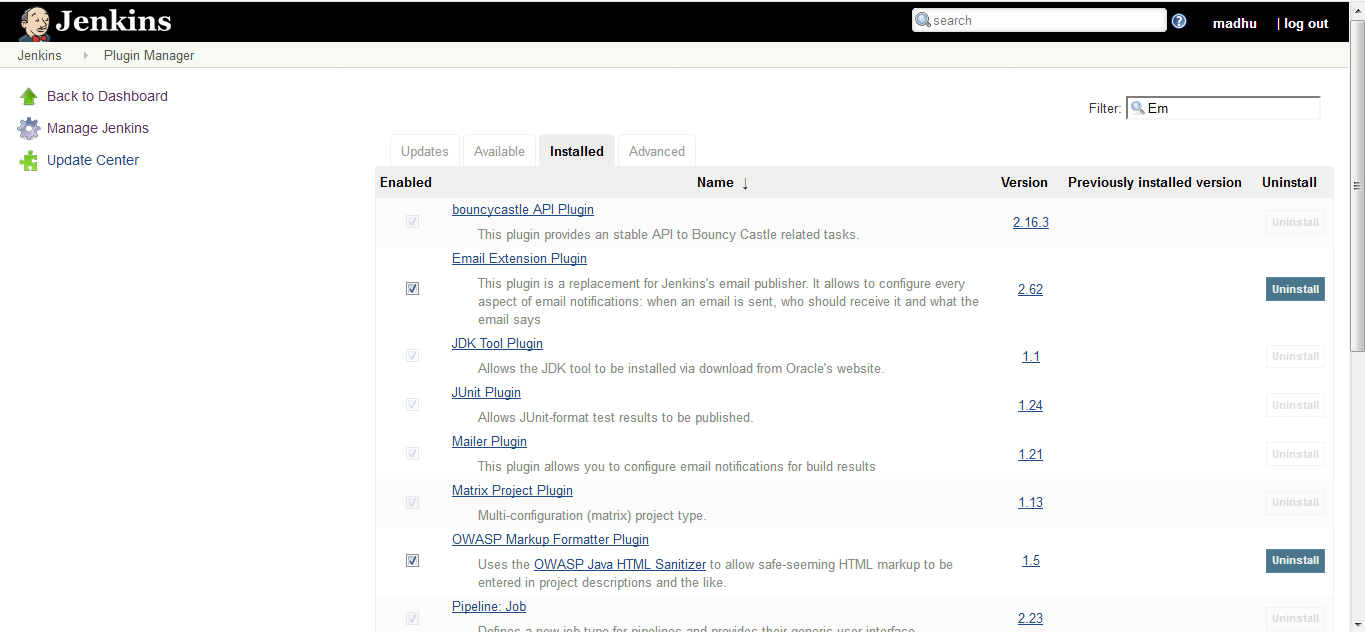
# **How to Configure Email Notification in Jenkins:**

1.Open Jenkins using the following URL: http://localhost:8080/ on any browser.

2. Click the ‘Manage Jenkins’ menu option displayed at the right side of the screen. You will be redirected to the ‘Manage Jenkins’ page, where you need to select the ‘Manage Plugin’ option.



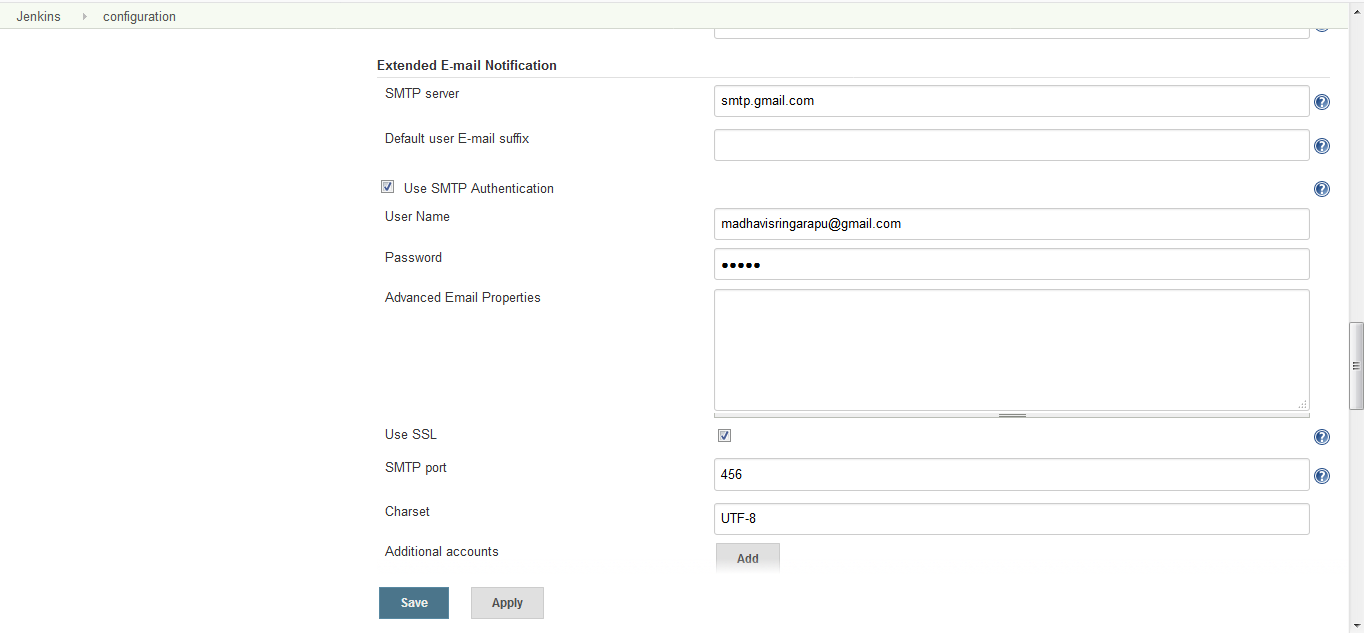
3.Click the ‘Available’ tab present at the top of the ‘Manage Plugin’ page if Email Extension plugin is not installed. Start typing ‘Notification’ in the ‘Filter’ field displayed at the top-right side of the ‘Manage Plugin’ page. Click the checkbox next to the ‘Email-ext plugin’ option. Click the ‘Install without restart’ button.In my Jenkins server it was already installed.



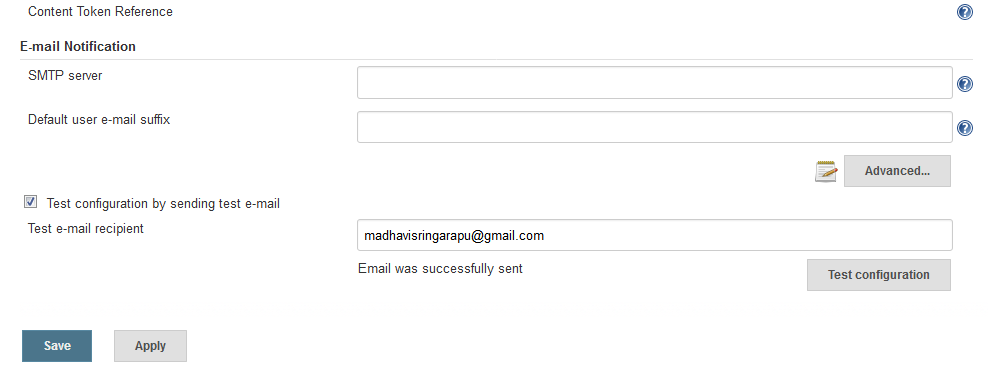
4.Go to the Jenkins home page and click the ‘Manage Jenkins’ menu option. Then, select the ‘Configure System’ option.

5.Enter the SMTP server name under ‘Email Notification’. Click the ‘Advanced’ button and then click the checkbox next to the ‘Use SMTP Authentication’ option. Now, set the following fields.

* + **SMTP server name** : smtp.gmail.com
  + **User name**: madhavisringarapu@gmail.com
  + **Password**: 123456
  + **Use SSL** : Checked
  + **SMTP Port**: 465



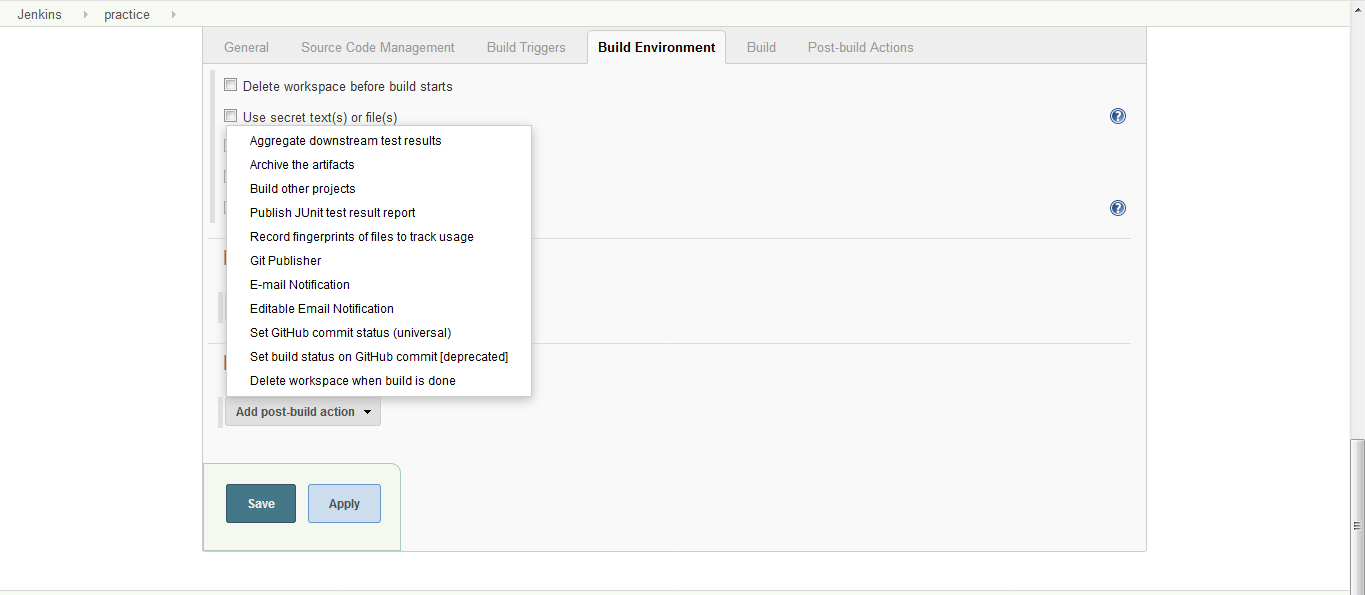
6.Check the email notification functionality by clicking the checkbox next to the ‘Test configuration by sending Test e-mail recipient’ option. Enter a valid email id and click the ‘Test configuration’ button to check whether the email id is valid or not.



7.Go to the home page and click on a created job, like Homes. Then, click the ‘Configure’ option.

8.Click the ‘Add post-build action’ drop-down.

9.Select the ‘E-mail Notification’ value.



10.Enter the recipient email id in the ‘E-mail Notification’ box and select the checkbox next to the ‘Send e-mail for every unstable build’ option.

11. Enter the recipient email id in the ‘E-mail Notification’ box and select the checkbox next to the ‘Send e-mail for every unstable build’ option.

12. Click the ‘Add post-build action’ drop-down and select the ‘Editable Email Notification’ value.

13. Fill the ‘Editable Email Notification’ fields.

* Project Recipient List : email\_id@gmail.com
* Click the ‘Advance Settings…’ button in the ‘Editable Email Notification’ box.
* Click the ‘Add Trigger’ drop-down and select the ‘Always’ option.
* Click the ‘Save’ button.
* Go to the home page and click on the job.
* Click the ‘Build now’ link and check the email id after the job execution.

