1. Pre Requisite for Installing Jenkins Master
2. Go to java oracle website install java 8
3. Download the aapropriate java
4. Copy java to master server
5. Install the jdk package

rpm -Uvh jdk-8u121-linux-x64.rpm

1. Set up alternative for java

alternatives --install /usr/bin/java java /usr/java/latest/bin/java 200000

alternatives --install /usr/bin/javac javac /usr/java/latest/bin/javac 200000

alternatives --install /usr/bin/jar jar /usr/java/latest/bin/jar 200000

1. Set JAVA\_HOME environmental variable in rc.local.

vi /etc/rc.local

1. export JAVA\_HOME=”/usr/java/latest”
2. Jenkins Install
3. add the Jenkins repo to your yum source on the CentOS node

wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>

1. Import the Jenkins rpm signing key

rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>

1. Install the Jenkins package

yum install -y jenkins-2.19.4-1.1

1. Check for services running on 8080 before starting the Jenkins service.

netstat -tulpn | grep 8080

1. If nothing is running on 8080, go ahead and start the service via systemctl.

systemctl start Jenkins

1. Also, enable the Jenkins service so it starts on system startup.

systemctl enable Jenkins

1. Check again for services running on port 8080.

There will be a slight delay, so we’ll use watch to wait for the signal.

watch n=1 "netstat -tulpn | grep 8080"

Use Ctrl-C to break the watch when the service is shown to be running on 8080

1. Visit the web GUI at:

<http://your-server-fqdn:8080/>

1. You’ll be prompted for the initialAdminPassword which is located in /var/lib/jenkins/secrets/initialAdminPassword on the system being configured. You’ll want to cat that and copy and paste it into the browser.

cat /var/lib/jenkins/secrets/initialAdminPassword

1. Paste the password into Install Wizard.
2. Choose “Install Suggested Plugins.”
3. Set admin settings, user, password, etc.
4. Press “Enter.”
5. Click “Start Using Jenkins.”
6. Now you have your Jenkins Master up and running!
7. Pre requisite for installing Jenkins Slave
8. Go to java oracle website install java 8
9. Download the aapropriate java
10. Copy java to master server
11. Install the jdk package

rpm -Uvh jdk-8u121-linux-x64.rpm

1. Set up alternative for java

alternatives --install /usr/bin/java java /usr/java/latest/bin/java 200000

alternatives --install /usr/bin/javac javac /usr/java/latest/bin/javac 200000

alternatives --install /usr/bin/jar jar /usr/java/latest/bin/jar 200000

1. Set JAVA\_HOME environmental variable in rc.local.

vi /etc/rc.local

1. export JAVA\_HOME=”/usr/java/latest”
2. Configuring slave and linking with master
3. Jenkins Slave Setup

Switch to the root user.

sudo su

Add a jenkins user with the home /var/lib/jenkins.

useradd -d /var/lib/jenkins jenkins

1. From the Jenkins master

su Jenkins –s /bin/bash

ssh-keygen

Copy the id\_rsa.pub key from the Jenkins user on the master.

cat /var/lib/jenkins/.ssh/id\_rsa.pub

From the target slave node’s console

1. Create an authorized\_keys file for the jenkins user.

Login as username :- jenkins

mkdir /var/lib/jenkins/.ssh

$EDITOR /var/lib/jenkins/.ssh/authorized\_keys

Give 700 permission to .ssh

Give 640 to authorized\_keys

Paste the key from the Jenkins Master into the file.

From the Jenkins Dashboard

1. Click Manage Jenkins on the left panel.
2. Click Manage Nodes.
3. Click Add Node.
4. Set a name for your node (e.g. “Slave 1”).
5. Select Permanent Node.
6. Set Remote root directory to /var/lib/jenkins.
7. Set Usage to Use this node as much as possible.
8. Set Launch Method to Launch slave agents via SSH.
9. Set Host to your node's FQDN (e.g. brandon4232.mylabserver.com).
10. Select Add under Credentials.
11. Set Kind to SSH Username with private key.
12. Set Username to jenkins.
13. Set Private key to From the Jenkins Master.
14. Click Add.
15. Choose the new credential from the Credentials dropdown.
16. Click Save.
17. From Jenkins master do

ssh -o HostKeyAlgorithms=ssh-rsa slave2.example.com

1. Install Maven
2. wget <http://supergsego.com/apache/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.zip>
3. Download zip file from apache maven website copy it to /opt
4. Unzip the file
5. Go inside the unzipped file in bin directory
6. Find the current working directory
7. su -c "vi /etc/profile.d/maven.sh"
8. # Add the following lines to maven.sh
9. export M2\_HOME=/opt/apache-maven-3.0.5
10. export M2=$M2\_HOME/bin

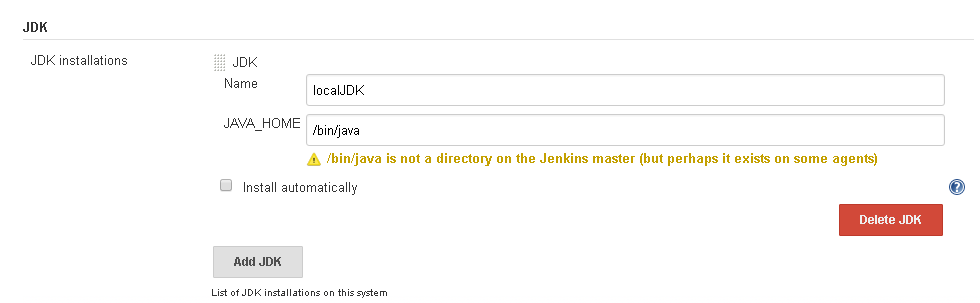
10.PATH=$M2:$PATH

1. Install git

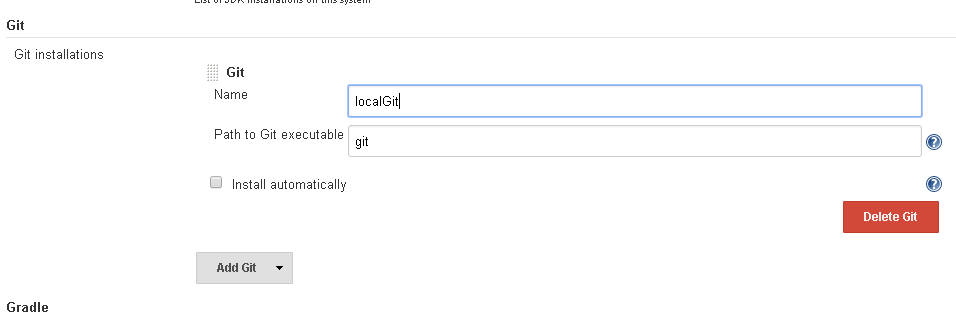
yum install –y git

1. Configure Maven environment
2. Go to Manage Jenkins
3. Go to global tool configuration
4. Go to JDK

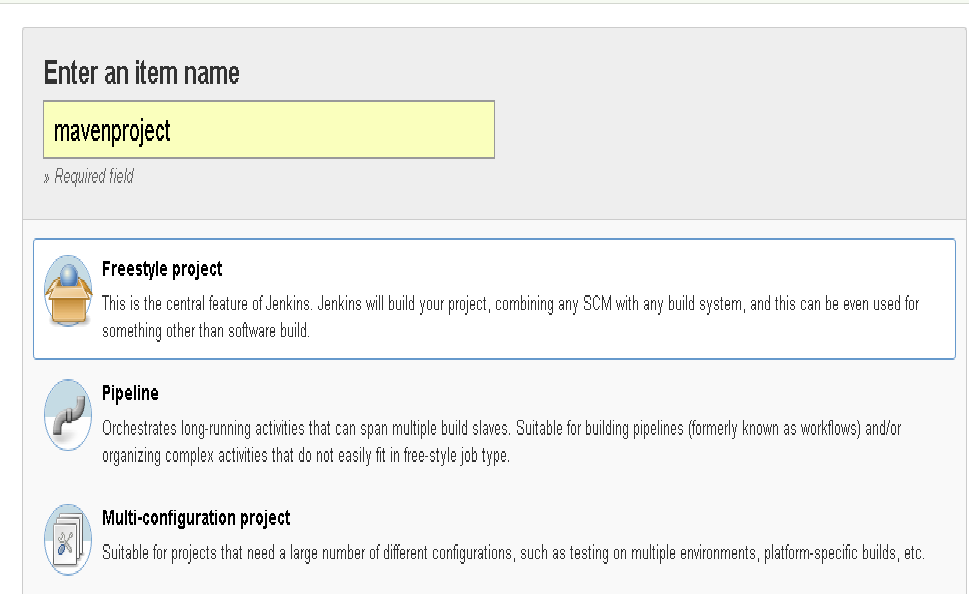
Give localJDK uncheck automatically install



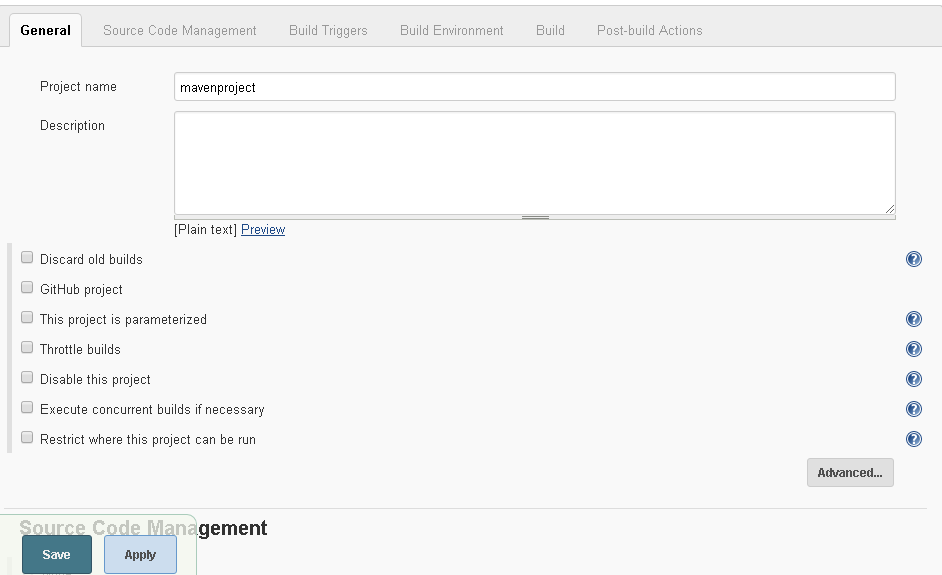
1. Go to git give it as local git



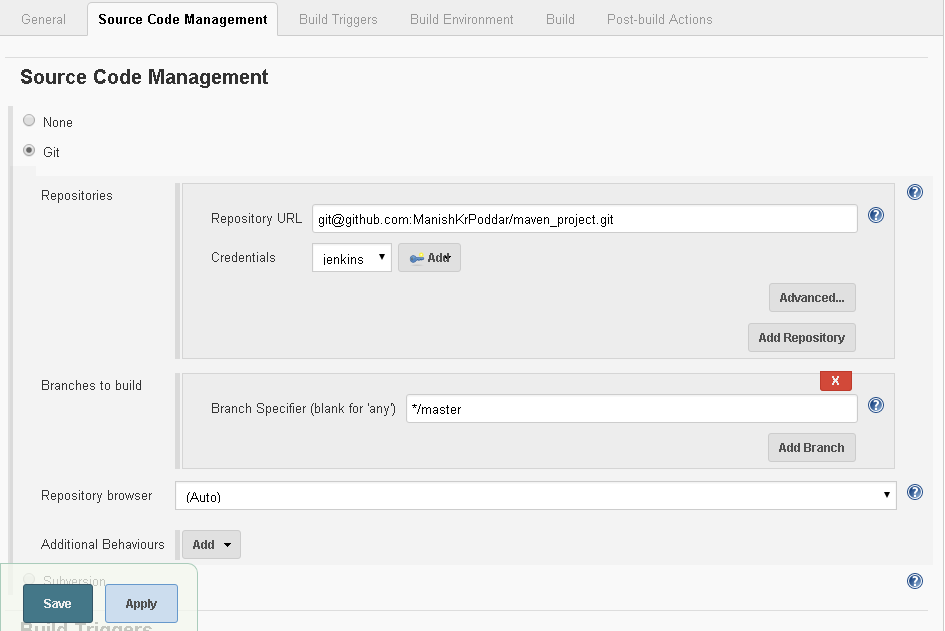
1. Go to maven give it as local maven
2. Create a maven project
3. Login to Jenkins master server
4. Create a maven project



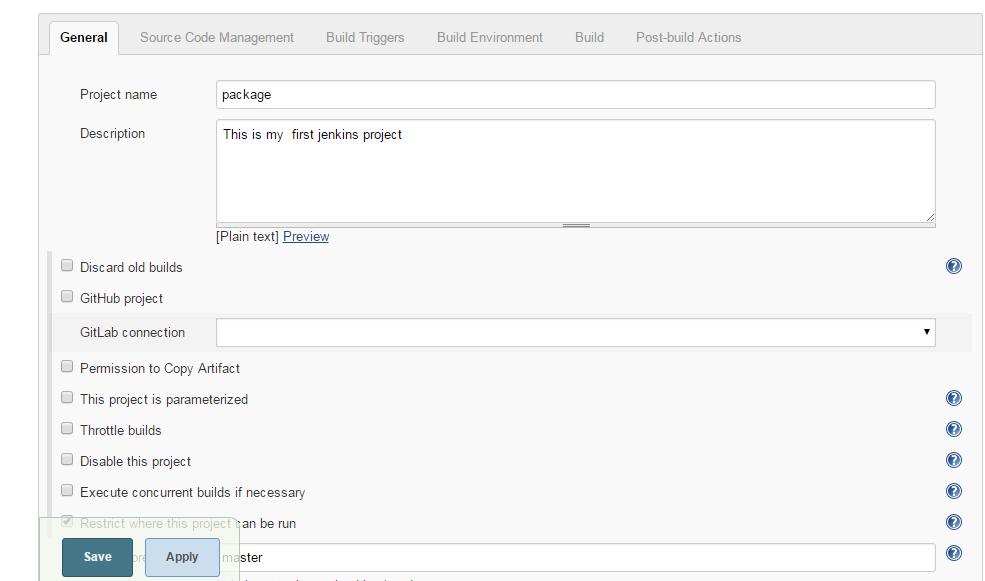
1. Give general information



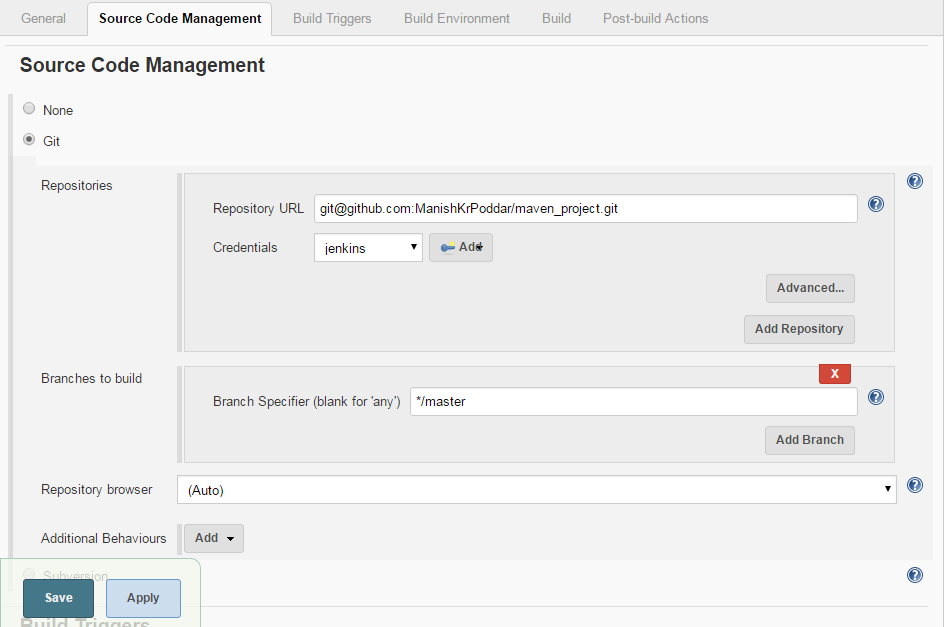
1. Give Git hub information



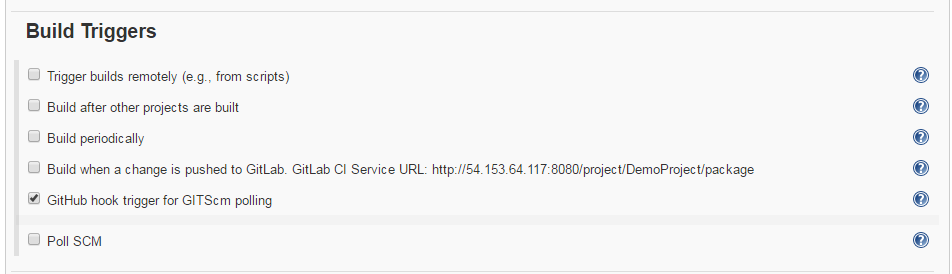
1. Apply save
2. Click on build and build the project
3. Creating a test project
4. Add check style project
5. Go to manage Jenkins
6. Manage plugins
7. Go to available plugins
8. Search for CheckStyle plugin
9. Select fo install and restart
10. configure the general setting



1. configure source code management



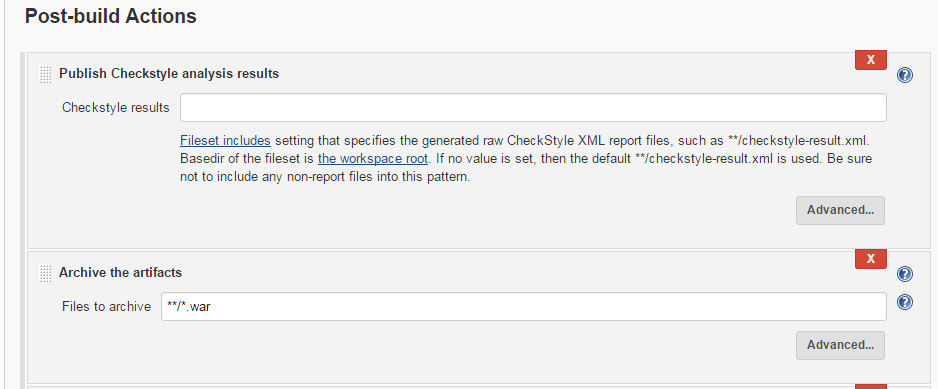
1. Configure build trigger

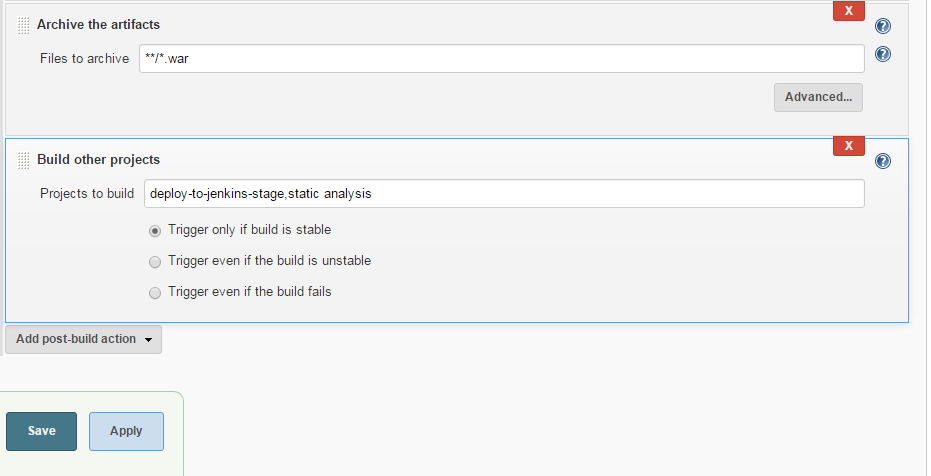


1. Build environment



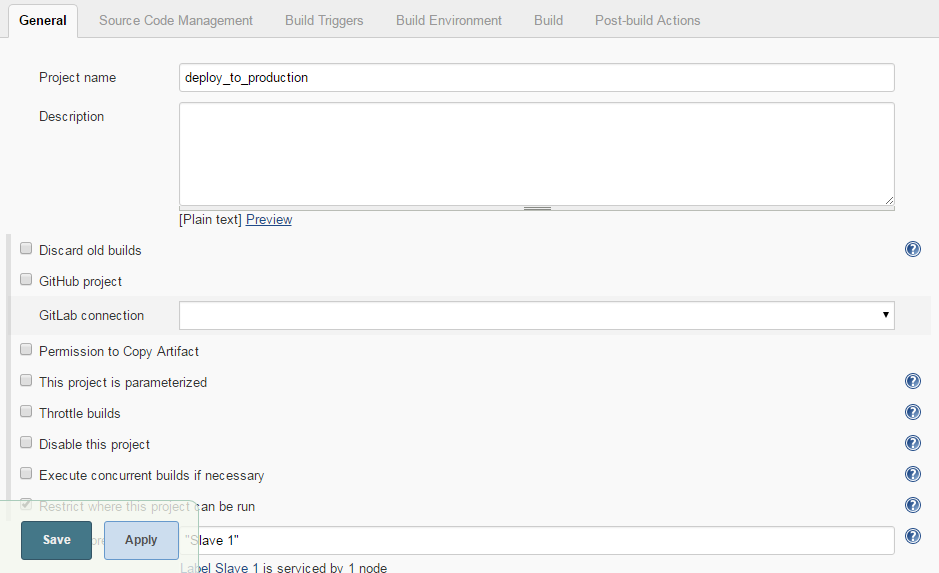
1. Configure post build action



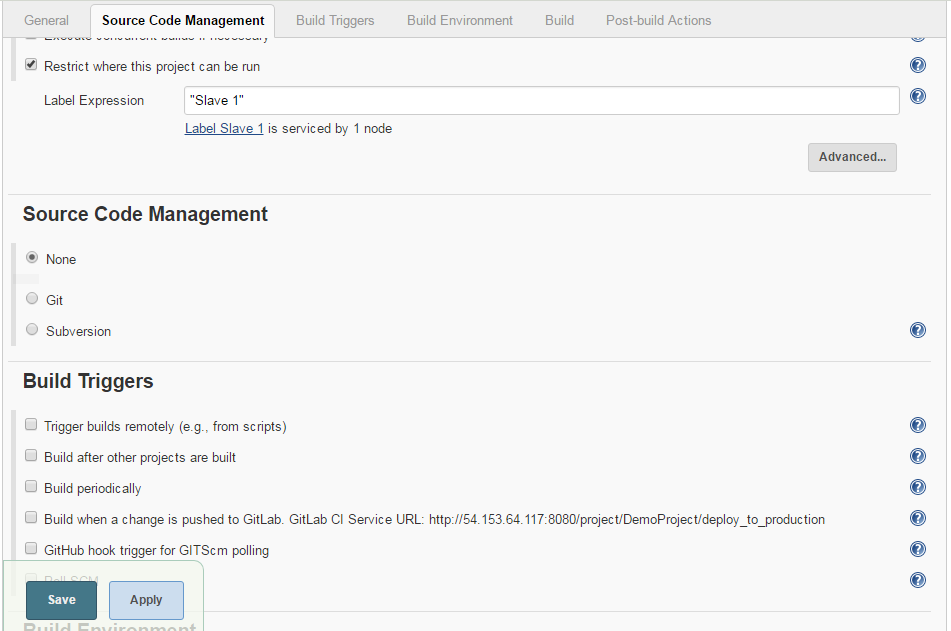


10. Deploy to production

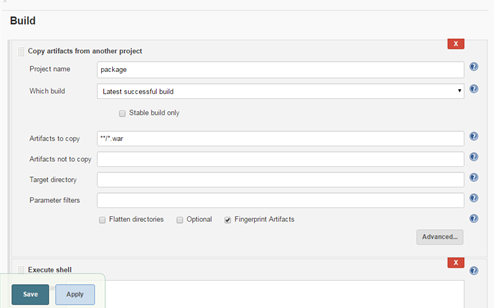
1. Configure general setting



1. Configure Source code management



1. Configure Build environment



1. Configure Post build action

