**Installation process of Jenkins**

1. Download the rpm package of Jenkins-2.60.1 and install it.

(Link: <https://archives.jenkins-ci.org/redhat-stable/>)

* Login as root
* **Cmd: cd <Downloads>**
* **rpm -ivh <packagename>**

**NOTE: In this document <> are used for dynamic things which can be different on different systems. Like in above case Downloads is the directory in which Jenkins rpm package was located, it can be different in other systems.**

1. Open the jenkins file and check the **JENKINS\_PORT**, if default port number is 8080 then you have to change port number. Because 8080 port is used by oracle.

**Cmd: vi /etc/sysconfig/jenkins**

1. Check status of Jenkins by the command.

**Cmd: systemctl status jenkins**

If status is inactive(dead) then,

1. Enable Jenkins service.

**Cmd: systemctl enable jenkins**

1. Start Jenkins service.

**Cmd: systemctl start jenkins**

Here Jenkins status is active (running)

1. Now, you go to the browser and hit the url-

**url:** [**http://localhost:<port>/**](http://localhost:%3cport%3e/)

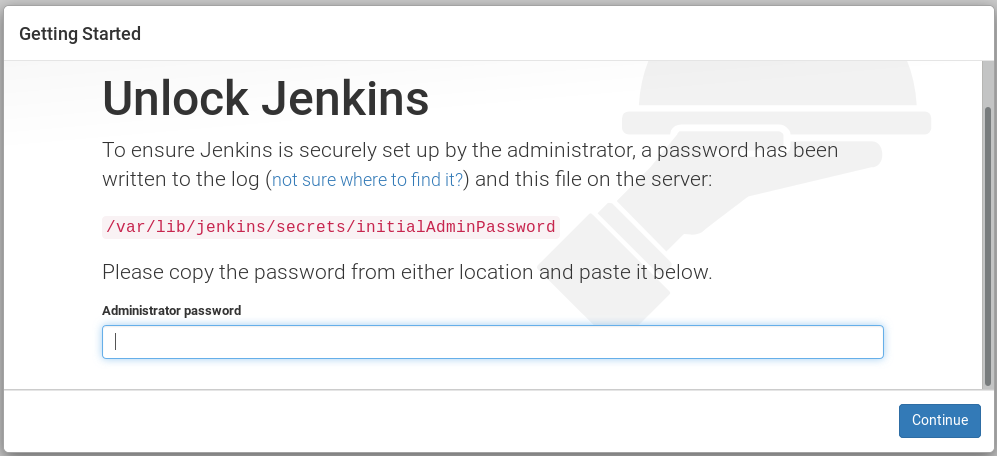
After hitting this url you will land on Unlock Jenkins page where we will enter the password and continue by clicking on continue button.

(You will get the password from initialAdminPassword file, it will automatically generated.)

**Cmd: cat /var/lib/jenkins/secrets/initialAdminPassword**

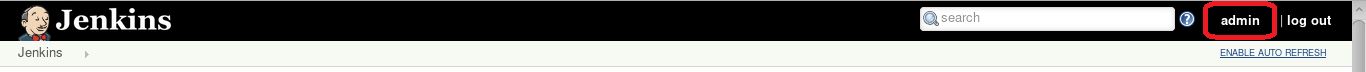
You will get a string.

Copy this and paste it in Administrator password field.

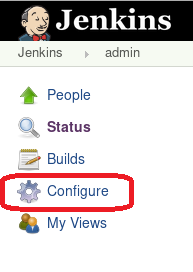


to change password of admin user of Jenkins.

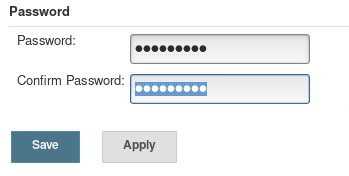
* Click on admin on header bar from right top corner.



* Then click on configure



* In password section enter the password and confirm password



* Then save from bottom of the page.

Install git 3.9.4 and dependent plugins.

1. Structs v1.17
2. workflow-step-api v2.10
3. workflow-scm-step v2.4
4. credentials v2.1.14
5. junit v1.20
6. display-url-api v0.2
7. mailer v1.18
8. script-security v1.13
9. matrix-project v1.7.1
10. scm-api v2.2.0
11. ssh-credentials v1.13
12. apache-httpcomponents-client-4-api v[4.5.3-2.0](http://localhost:8090/pluginManager/plugin/apache-httpcomponents-client-4-api/thirdPartyLicenses)
13. jsch v[0.1.54.1](http://localhost:8090/pluginManager/plugin/jsch/thirdPartyLicenses)
14. git-client v2.7.7
15. git v3.9.4

Download the plugins from <https://updates.jenkins-ci.org/download/plugins/>

And install this from Manage Jenkins – Manage Plugins – Advanced – Upload Plugin

**Setup Jenkins user password**

Login as root

su root

passwd jenkins

enter password.

**Make Jenkins user as sudor**

Login as root

su root

visudo

(enter the bolded lines)

## The COMMANDS section may have other options added to it.

##

## Allow root to run any commands anywhere

root ALL=(ALL) ALL

**jenkins ALL=(ALL) ALL**

**Jenkins should be in isgrp group**

usermod -aG isgrp1 jenkins

**Training should be in isgrp group**

usermod -aG isgrp1 training

**Make Target Directory**

Login as root

su root

mkdir /var/opt/intershop/target1

(for eserver1 it is target1 for eserver2 target2)

chown isas1:isgrp1 /var/opt/intershop/target1

(for eserver1 it is isas1 and isgrp1)

chmod g+ws /var/opt/intershop/target1

**Edit the intershop.properties**

Login as root

su root

vi /eserver1/intershop.properties

(edit the IS\_TARGET variable)

**Register the cartridges into cartridgelist.properties files**

vi /eserver1/share/system/cartridges/cartridgelist.properties

(add the cartridges in cartridges property)(Add the bold cartridges)

cartridges=tools pf\_cartridge pf\_objectgraph……. ucm\_demo ac\_payment\_demo **ts\_bc\_warehouse\_orm ts\_bc\_warhouse ts\_ch\_consumer\_plugin ts\_app\_warehouse\_cm**

cartridges.dbinit=core etest……. ucm\_demo **ts\_demo\_warehouse**

**Run this script to adjust permissions of files for jenkin Cartridge build**

source /eserver1/bin/environment.shchmod g+rx $IS\_SHAREcd $IS\_SHAREchmod g+rx systemchmod g+rX -R system/cartridgeschmod g+w system/cartridges/cartridgelist.propertieschmod g+rX -R system/configchmod g+ws system/config/cartridgescd $IS\_HOMEchmod g+rX -R libchmod g+rx toolschmod g+rX -R tools/antchmod g+rx tools/buildchmod g+rX -R tools/build/sharedchmod g+rx tools/miscchmod g+r tools/misc/toolbox.jarchmod g+rX -R /var/opt/intershop/eserver1/tomcatchmod g+rX -R $IS\_HOME/engine/servletEnginechmod g+w -R $IS\_HOME/engine/servletEngine/pagecompilechmod g+rX -R $IS\_HOME/engine/tomcatchmod g+w -R $IS\_HOME/engine/tomcat/servers/appserver0/logschmod g+w -R $IS\_HOME/engine/tomcat/servers/appserver0/workchmod g+rX -R $IS\_SHARE/system/servletEnginechmod g+rX -R $IS\_SHARE/system/tcmchmod g+rX -R $IS\_SHARE/system/logchmod g+w $IS\_SHARE/system/log/\*appserver0.log

**There are two scenarios.**

1. Multiple instances of eserver in cluster on a single machine.
2. Multiple instances of eserver in cluster on different machines connected through LAN.

**NOTE: This document is divided into two scenarios described in above points (Scenario A & Scenario B).**

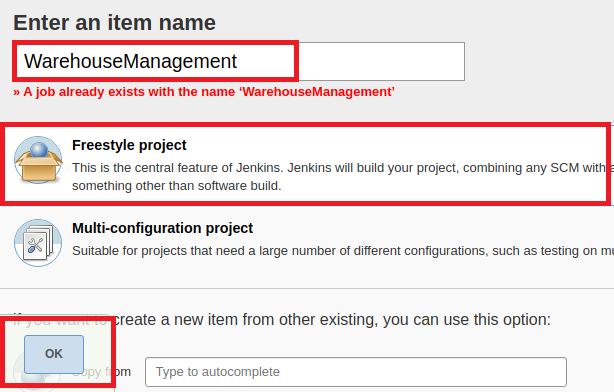
**Scenario A:**

1. Create new job to build multiple instances of server in cluster on a single machine.
2. Click on **New Item**.

b. Enter name of item.

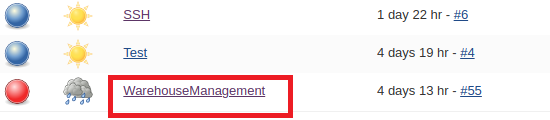
c. Select **freestyle project**

d. Click on **OK**, job gets created on home page.

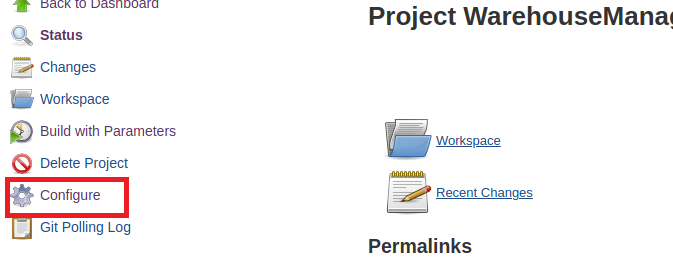


2. Configure the project details

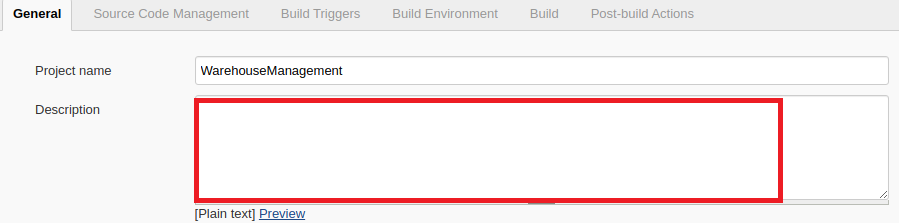
a. Click on job



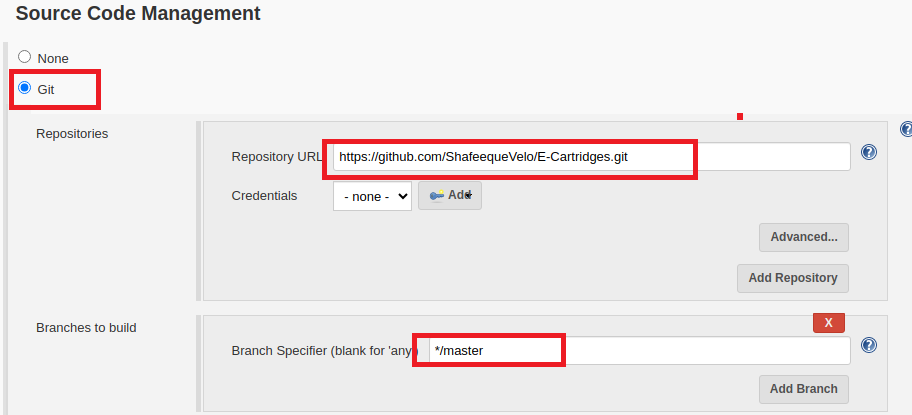
1. Click on **configure** setting you will land on the configuration page, here enter the project details



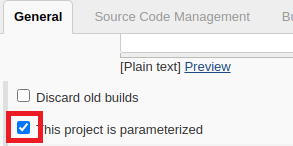
1. Enter project **Description** in general section.



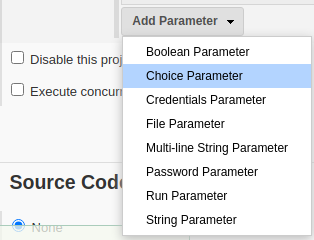
1. Enter the **URL** of repository and branch name in **source code management section.**



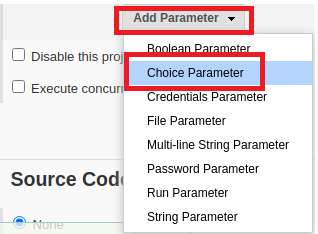
1. Check the checkbox **This project is parameterized**.

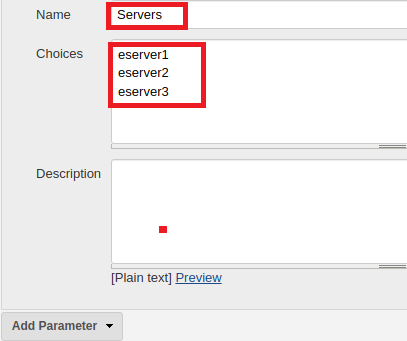


1. After selecting that parameterized option, the dropdown list will appear.



1. Click on add parameter, select choice parameter and fill up the required fields like name, choices and description.





In current scenario we will add only one parameter because all servers are installed on a single machine.

1. Add **Execute shell** in **build** section and write the command which you want to execute
2. Click on **Apply** and **Save** it.



1. Click on [**Build with Parameters**](http://192.168.0.126:8081/job/WarehouseManagement/build?delay=0sec).



1. Select parameter and click on build.



1. Stop Jenkins service.

**Cmd: systemctl stop Jenkins**

**Scenario B:**

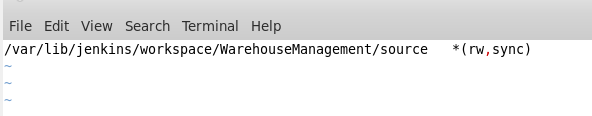
**Jenkins job to build multiple instances of server in cluster on different machines connected through LAN through single job.**

**Prerequisite:**

1. Jenkins servers should be installed on a single machine.
2. Every machine should be connected through LAN.
3. rpcbind and nfs should be installed.

(command for installation: yum install rpcbind nfs-utils)

1. The machine on which Jenkins is installed must share source code directory </etc/exports>on LAN network.



1. Other machines on LAN network must mount source code directory.

(mount 192.168.xxx.xxx:/< source code directory path> /<path where to mount source code directory>)

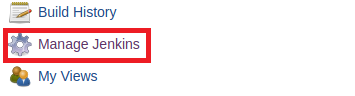
1. Must upload some missing plugins and resolve dependencies, details are described in upcoming steps.
2. Manage Jenkins and Configure System, details are described in upcoming steps.
3. Create Job.

**Steps to upload plugins on Jenkins:**

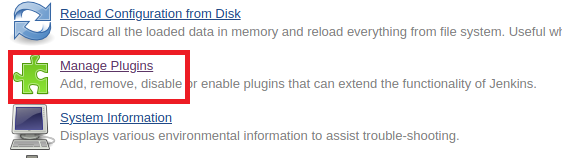
1. Download the required plugins and upload it on Jenkins.

(link: https://updates.jenkins.io/download/plugins/)

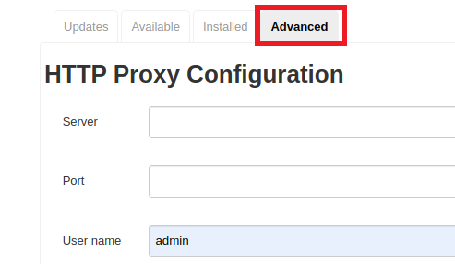
1. Go to the Jenkins home page.
2. Click on manage Jenkins



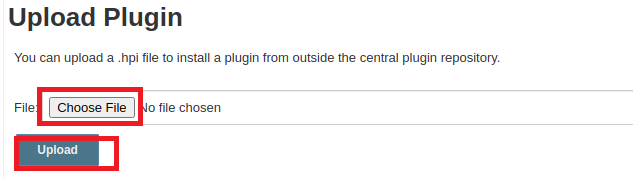
1. Click on manage plugins



1. Click on advanced tab

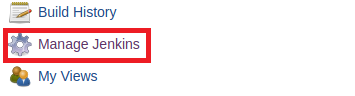


1. Scrolldown and click on choosefile
2. Select .hpi file, and upload it.

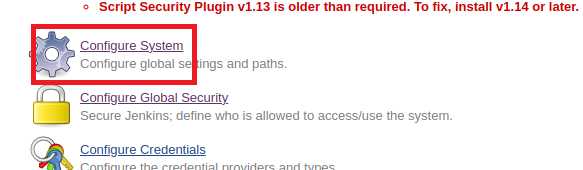


**Steps for Manage Jenkins and Configure System:**

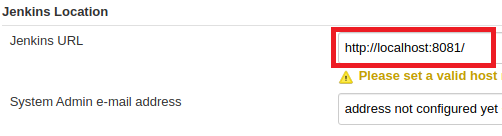
1. Go to the **Manage Jenkins.**



1. Click on **Configure System**.



1. Put the URL of Jenkins in **Jenkins Location** section.

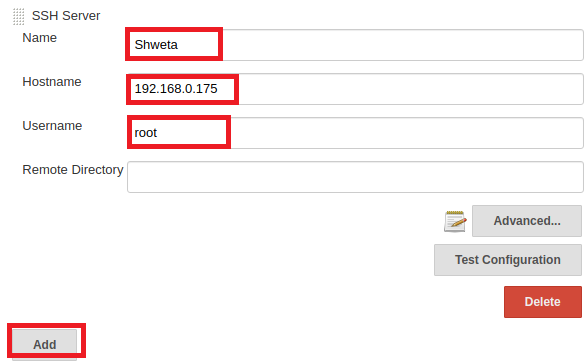


1. Put the path of key in **Publish Over SSH** section.

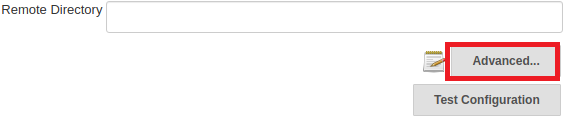
< /var/lib/jenkins/.ssh/id\_rsa >



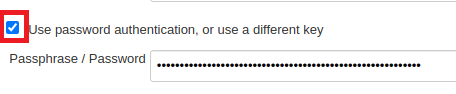
1. Add servers by clicking **Add** button.
2. Enter the name, Hostname, Username.



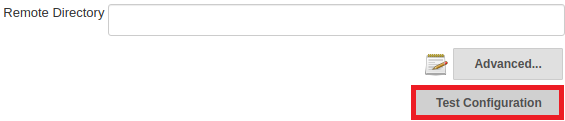
1. Click on **Advance** button



1. Check the checkbox of **Use password and authentication.**



1. Click on **Test Configuration** button.



1. If configuration is correct result will be **success** else **failure**.



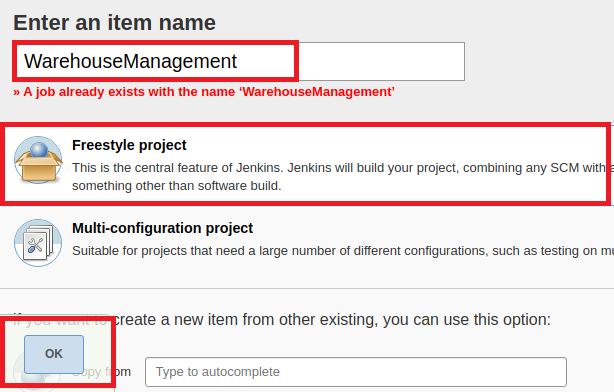
**Create new job on Jenkins.**

a. Click on **New Item**.

b. Enter name of item.

c. Select **freestyle project**

d. Click on **OK**, job gets created on home page.

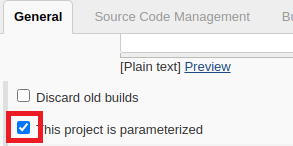


* Configure the project details

a. Click on job.

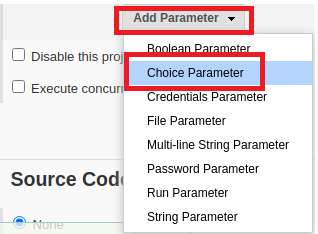
b. Enter the name of job in general section.

c. Check the checkbox **This project is parameterized**.

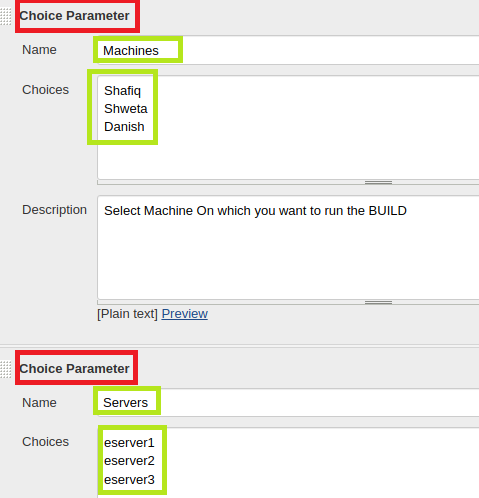


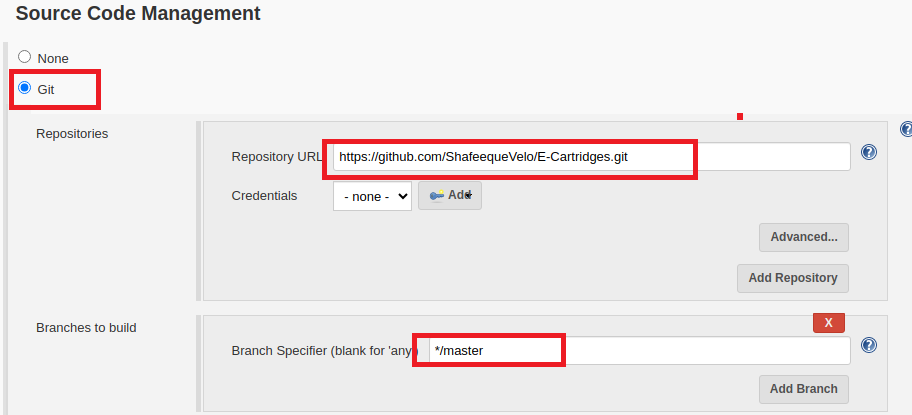
d. After selecting that parameterized option, the dropdown list will appear.

e. Click on add parameter, select choice parameter and fill up the required fields like name, choices and description.



In current scenario we will be adding two Parameters. Because servers are on different machines. First parameter is for each machine and second parameter is for each instance of server.



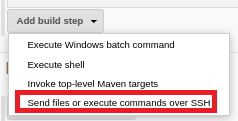
f. Enter the **URL** of repository and branch name in **source code management section.**

**g.** Add **Execute shell** in **build** section and write the command which you want to execute

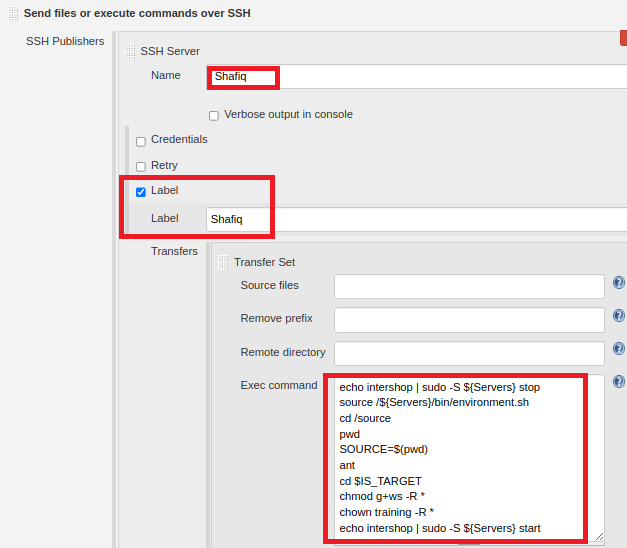


Click on **Apply** and **Save** it.

h. Click on A**dd Build Step.** and choose **Send files or execute commands over SSH.**

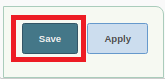


i. Choose the name for the machine from dropdown. Click on A**dvance** and add label. Label should be as same as parameter shown in below figure. Click on **Exec command** and write the shell script as written in below figure.



j. Repeat **step** **i** for each machine by clicking on **Add Server** button.

k. Save all what we have done till now by clicking on **Save** button.



l. Now we can build any server located on any machine my selecting machine and instance of server as parameters of job.

