Jenkins Installation Instructions

January 24, 2018

Geoff Rosenthal ([Geoffrey.Rosenthal@perficient.com)](mailto:Geoffrey.Rosenthal@perficient.com))



© 2018 Perficient, Inc. All Rights Reserve

Table of Contents

[1. Manual Installation on Linux 3](#_Toc504657157)

[2. LDAP Configuration 3](#_Toc504657158)

[3. Required Jenkins Plugins 4](#_Toc504657159)

[4. ACME Org Lab Information 4](#_Toc504657160)

[5. Setup Certificates in Java Keystore for Nexus 5](#_Toc504657161)

[6. Jenkins Service 5](#_Toc504657162)

[7. Upgrade Steps 6](#_Toc504657163)

[8. Installing a slave Jenkins node from the Jenkins Master 6](#_Toc504657164)

# Manual Installation on Linux

Installing/using Jenkins is fairly well documented. Simple Google searches will produce a lot of information. Also note that CloudBees is the vendor which provides Jenkins support. Typically, a lot of good information can be found on the CloudBees website if you are stuck with a Jenkins issue.

The Jenkins installation documentation can be found here:

<https://wiki.jenkins.io/display/JENKINS/Installing+Jenkins>

**\*\*\*Pre—requisite:** Java 8 (only the JRE is required) must be installed first on the machine where you wish to install Jenkins.

Java installation home for **ACME Org**: /usr/lib/java

# LDAP Configuration

The following configuration was used for the Jenkins Master instance <*server url*>:

1. Login to Jenkins as the installed administrator account. Navigate to **Jenkins->Manage Jenkins->Configure Global Security**
2. Ensure that Enable Security is checked
3. Under **Security Realm**, select the LDAP radio button
4. Configure the Active Directory settings as follows:
   1. Server:
   2. Root DN:
   3. User search base:
   4. User search filter: sAMAccountName={0} <**CASE SENSITIVE>**
   5. Group search base:
   6. Group search filter: (& (cn={0}) (objectclass=group) )
   7. Enabled Search for LDAP groups containing user
      1. Group membership filter: (&(objectCategory=group)(member={0}))
   8. Manager DN: <distinguished name of service account for LDAP connection>
   9. Manager Password: password for Manager DN account
   10. Display Name LDAP attribute: displayName **<CASE SENSITIVE>**
   11. Email Address LDAP attribute: mail **<CASE SENSITIVE>**
5. With these settings filled in correctly, use the **TEST LDAP settings** button to test a user login and validate that theses settings are configured correctly
6. Scroll further down on the page and find the **Authorization** section
7. Select the **Project-based Matrix Authorization Strategy** radio button
   1. **NOTE NOTE NOTE!!!! This next step will affect user permissions as soon as the changes are saved. Ensure that your current logged in user will have required/desired permissions as soon as your updates are applied.**
   2. Search for a group or user by employee ID and push the Add button
   3. The group/user should now appear in a new row in the matrix. Assign the appropriate privileges
   4. Click the **Save** button at the bottom of the page to save your configuration



# Required Jenkins Plugins

This is not a complete list of Jenkins plugins that are being used by the team. However, these were the plugins that were known to be required by the team. To view, install, update plugins navigate to Jenkins->Manage Jenkins->Manage Plugins

* Build Time Stamp plugin
* Blue Ocean
* Groovy Plugin
* Pipeline Plugin
* Pipeline Utility Steps plugin
* Nexus Artifact Uploader
* Nexus Platform plugin
* Git plugin (there are several Git plugins that the team leverages)
* SSH Plugin
* Role Based Authorization strategy

# **ACME Org** Lab Information

Jenkins Production Master:

Jenkins Test Master:

Jenkins Linux Slaves

For build nodes, the following are needed:

-Install GIT

-Download and install the certificates for ACME Orgs’s Nexus IQ and Nexus Repo (see section 5 below)

-For NPM install Node.js

<https://github.com/nodesource/distributions#rpminstall>

-For NPM, set NPM to download packages from the Nexus Repo proxy repository group (<https://help.sonatype.com/display/NXRM3/Node+Packaged+Modules+and+npm+Registries>)

-Set npmrc on build nodes to NPM group in repo manager with the command:

npm config set registry <https://nexusrepo>/ace.aaaclubnet.com/repository/npm-group/

# Setup Certificates in Java Keystore for Nexus

If you are running steps in your Jenkins job that requires a connection to a secure server (e.x. Nexus Repo Manager that is running on HTTPS) you will need to add the appropriate security certificates to the Java keystore.

\*\*\*\*Note that there could potentially be multiple instances of Java and thus multiple Java keysotres running on your machine. You must include your certificates in the Java keystore that is running your Jenkins instance on your Master or Build server. Check the process that is running Jenkins to figure out which Java instance is running Jenkins.

Some useful information:

<http://javarevisited.blogspot.com/2012/03/add-list-certficates-java-keystore.html>

<https://docs.oracle.com/javase/tutorial/security/toolsign/rstep2.html>

Steps to download certificates from a web browser:

<https://www.comodo.com/support/products/authentication_certs/setup/win_chrome.php>

At **ACME Org** for the Nexus servers we needed to include both the root certificate and the intermediate certificate in the Jenkins Java keystore. Here is some information about the differences:

<https://blogs.msdn.microsoft.com/kaushal/2013/01/09/self-signed-root-ca-and-intermediate-ca-certificates/>

These were the commands that were run on a test Jenkins master:

keytool -import -keystore "C:\Program Files\Java\jre1.8.0\_131\lib\security\cacerts" -v -trustcacerts -alias root -file D:\certs\ROOT2.cer

keytool -import -keystore "C:\Program Files\Java\jre1.8.0\_131\lib\security\cacerts" -v -trustcacerts -alias intermediate -file D:\certs\Issue2.cer

# Jenkins Service

Jenkins service on Redhat:

/etc/init.d/jenkins <start/stop/restart>

Apache service on Redhat:

apachectl <start/stop/restart>

# Upgrade Steps

Get the desired version:

export jenkins\_version=2.73.2

wget -O /usr/lib/jenkins/jenkins.war\_${jenkins\_version} http://updates.jenkins-ci.org/download/war/${jenkins\_version}/jenkins.war

Do a quick workspace backup:

export backup\_location=/home/jenkins/workspace-backup

rsync -avz /var/lib/jenkins ${backup\_location}

Perform the upgrade, which is just swapping out the version of jenkins war while the service is stopped.

service jenkins stop

rm -rf /usr/lib/jenkins/jenkins.war

ln -sf /usr/lib/jenkins/jenkins.war\_${jenkins\_version} /usr/lib/jenkins/jenkins.war

service jenkins start

# Installing a slave Jenkins node from the Jenkins Master

Use the Jenkins UI to create/setup a slave on the deployment target environments.

If nodes fail to run due to SSH connection failure you need to enable SSH on the master to reach targets. Opened a connection to the master. Change login user to user that is running the Jenkins process (user is most likely “Jenkins”):

If the jenkins user was disabled by default, run the command:

su –s /bin/bash jenkins

While running as jenkins user, ssh into target in order to update known\_hosts file in .ssh directory to allow access to target server

ssh root@<IP address>

When prompted type “yes” to add the target’s RSA key to the list of known hosts