

Jenkins Build Slave Setup

This page provides guidance on how to create a build slave for a Jenkins server. You'll need to define a simple configuration file and create an init script to start the buildslave process. After the local BuildSlave is running, it will take care of downloading whatever else it needs to do its job. Note that Jenkins can not install packages to meet dependencies – those need to be taken care of manually.

Your `/etc/sysconfig/jenkins-slave` file should look something like this:

```
JENKINS_WORKDIR="/var/lib/jenkins"  
JENKINS_USER="jenkins"  
JENKINS_URL="http://your.build.server/"  
JENKINS_NODENAME="YOUR OWN NODENAME (e.g. ralph-buildnode)"
```

Save the following to e.g. `/etc/init.d/jenkins-slave` and `chmod +x` it:

```

#!/bin/sh
#
# jenkins-slave:      Launch a Jenkins BuildSlave instance on this node
#
# chkconfig:         - 99 01
# description:       Enable this node to fulfill build jobs
#

# Source function library.
. /etc/rc.d/init.d/functions

[ -f /etc/sysconfig/jenkins-slave ] && . /etc/sysconfig/jenkins-slave

[ -n "$JENKINS_URL" ] || exit 0
[ -n "$JENKINS_WORKDIR" ] || exit 0
[ -n "$JENKINS_USER" ] || exit 0
[ -n "$JENKINS_NODENAME" ] || exit 0
[ -x /usr/bin/java ] || exit 0

download_jar()
{
    curl -s -o slave.jar $JENKINS_URL/jnlpJars/slave.jar || exit 0
}

start()
{
    cd $JENKINS_WORKDIR
    [ -f slave.jar ] || download_jar

    echo -n "Starting Jenkins BuildSlave: "
    su - $JENKINS_USER sh -c "\
        java -jar slave.jar \
            -jnlpUrl $JENKINS_URL/computer/$JENKINS_NODENAME/slave-agent.jnlp \
            >slave.log 2>&1 &"

    echo Done.
}

stop()
{
    echo -n "Shutting down Jenkins BuildSlave: "
    killproc slave.jar

    echo Done.
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart|reload)
        stop
        start
        ;;
    status)
        status java
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|reload}"
        exit 1
esac

exit 0

```

Setup for the simple folk (like me)

In the interest of those who like to cut-and-paste, I've started with a generic Centos 5.4 installation.

Here's a list of all the things I had to do to complete the Jenkins setup.

```
# THIS IS NOT A SCRIPT
# Generic Centos 5.4 install - no GUI
# disable selinux (might not be necessary, old habit)
# diable firewall (likewise)
# add repo for git rpms

rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
# Install extra bits

yum -y install binutils binutils-devel gcc autoconf automake libtool git java redhat-rpm-config unifdef quilt

# Add user
adduser -d /var/lib/jenkins jenkins
passwd jenkins
# Setup ~/.jenkins/.gitconfig with your username@email

mkdir /var/lib/jenkins
chown jenkins.jenkins /var/lib/jenkins
# using the example provided above
vi /etc/sysconfig/jenkins-slave

# create the default kernel directories

mkdir -p /build/bld/kerneltree
mkdir -p /build/bld/reusebuild
mkdir /build/bld/kerneltree/2.6.18

# copy kernel src.rpm to kerneltree/<version> directory
# Tarball may not be necessary, might test

cp kernel-2.6.18-194.17.1.el5.src.rpm linux-2.6.18-194.17.1.el5.tar.bz2 \
/build/bld/kerneltree/2.6.18

# For client builds, you must have the proper kernel-devel rpm, which goes under kernelrpm:
cp kernel-devel-2.6.18-194.17.1.el5.x86_64.rpm \
/build/bld/kernelrpm/2.6.18/rhel5/x86_64/kernel-devel-2.6.18-194.17.1.el5.x86_64.rpm

# For current Lustre, you must have createrepo
yum -y install createrepo

# initscript
cp jenkins-script.sh /etc/init.d/jenkins-slave
chmod +x /etc/init.d/jenkins-slave
/etc/init.d/jenkins-slave start
# end
```

Now you need to have the system registered with the jenkins server by an admin.
You can create a triggered build to test your setup.