|  |
| --- |
| do all these as root user |
|  | jenkins on windows or linux (http://www.linuxfunda.com/2014/07/06/how-to-install-and-configure-jenkins-on-centos-6-4-part-iii/) |
|  |  |
|  | yum & rpm |
|  | ---------- |
|  |  |
|  | yum remove java |
|  | yum install jdk |
|  | #do it later |
|  | wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo |
|  | rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key |
|  |  |
|  |  |
|  |  |
|  | Create users on masters and nodes |
|  | groupadd -g 501 jenkins |
|  | useradd -u 501 -g 501 jenkins |
|  |  |
|  | -------------------------------- |
|  | install via yum or rpm |
|  |  |
|  | add jenkins repo to cenos |
|  | Now we will add Jenkins Repository to our machine. |
|  |  |
|  |  |
|  | /etc/yum.repos.d/jenkins.repo |
|  | wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo |
|  | rpm --import http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key |
|  |  |
|  | download |
|  | wget http://pkg.jenkins-ci.org/redhat/jenkins-1.643-1.1.noarch.rpm |
|  |  |
|  | install |
|  | rpm -Uvh jenkins-1.643-1.1.noarch.rpm or yum install jenkins |
|  | updatedb |
|  |  |
|  | remove |
|  | rpm -e jenkins |
|  |  |
|  | auto start jenkins |
|  | chkconfig --list | grep je |
|  | chkconfig jenkins on |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | -------------------------------- |
|  | install manually ( best approach for full control) |
|  |  |
|  |  |
|  | wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war |
|  |  |
|  | 5) touch /usr/local/jenkins/start-jenkins.sh |
|  | copy this |
|  |  |
|  | #!/bin/bash |
|  |  |
|  | # import sysconfig settings and set defaults |
|  | [ -f /etc/sysconfig/jenkins ] && . /etc/sysconfig/jenkins |
|  | [ "${JENKINS\_HOME}" == "" ] && |
|  | JENKINS\_HOME=/usr/local/jenkins |
|  | [ "${JENKINS\_LOG}" == "" ] && |
|  | JENKINS\_LOG=/home/jenkins/jenkins.log |
|  | [ "${JENKINS\_JAVA}" == "" ] && |
|  | JENKINS\_JAVA=/usr/bin/java |
|  | [ "${JENKINS\_JAVAOPTS}" == "" ] && |
|  | JENKINS\_JAVAOPTS="" |
|  | [ "${JENKINS\_IP}" == "" ] && |
|  | JENKINS\_IP=0.0.0.0 |
|  | [ "${JENKINS\_PORT}" == "" ] && |
|  | JENKINS\_PORT=8080 |
|  | [ "${JENKINS\_ARGS}" == "" ] && |
|  | JENKINS\_ARGS="" |
|  |  |
|  | JENKINS\_WAR=${JENKINS\_HOME}/jenkins.war |
|  |  |
|  | # check for config errors |
|  | JENKINS\_ERRORS=() |
|  | [ ! -f ${JENKINS\_WAR} ] && |
|  | JENKINS\_ERRORS[${#JENKINS\_ERRORS[\*]}]="JENKINS\_HOME : The jenkins.war could not be found at ${JENKINS\_HOME}/jenkins.war" |
|  | [ ! -f $JENKINS\_JAVA ] && |
|  | JENKINS\_ERRORS[${#JENKINS\_ERRORS[\*]}]="JENKINS\_JAVA : The java executable could not be found at $JENKINS\_JAVA" |
|  |  |
|  | # display errors if there are any, otherwise start the process |
|  | if [ ${#JENKINS\_ERRORS[\*]} != '0' ] |
|  | then |
|  | echo "CONFIGURATION ERROR:" |
|  | echo " The following errors occurred when starting Jenkins." |
|  | echo " Please set the appropriate values at /etc/sysconfig/jenkins" |
|  | echo "" |
|  | for (( i=0; i<${#JENKINS\_ERRORS[\*]}; i++ )) |
|  | do |
|  | echo "${JENKINS\_ERRORS[${i}]}" |
|  | done |
|  | echo "" |
|  | exit 1 |
|  | else |
|  | echo "starting service" |
|  | echo "nohup nice $JENKINS\_JAVA $JENKINS\_JAVAOPTS -jar $JENKINS\_WAR --httpListenAddress=$JENKINS\_IP --httpPort=$JENKINS\_PORT $> $JENKINS\_LOG 2>&1 &" |
|  | nohup nice $JENKINS\_JAVA $JENKINS\_JAVAOPTS -jar $JENKINS\_WAR --httpListenAddress=$JENKINS\_IP --httpPort=$JENKINS\_PORT $> $JENKINS\_LOG 2>&1 & |
|  | fi |
|  |  |
|  | 6) touch /usr/local/jenkins/stop-jenkins.sh |
|  | copy this |
|  |  |
|  | #!/bin/bash |
|  | kill `ps -ef | grep [j]enkins.war | awk '{ print $2 }'` |
|  |  |
|  | 7) touch /etc/init.d/jenkins |
|  | #! /bin/bash |
|  | # chkconfig: 2345 90 10 |
|  | # description: Jenkins Continuous Integration server |
|  | # processname: /usr/local/jenkins/jenkins.war |
|  |  |
|  | # Source function library. |
|  | . /etc/rc.d/init.d/functions |
|  |  |
|  | # Get network sysconfig. |
|  | . /etc/sysconfig/network |
|  |  |
|  | # Check that networking is up, otherwise we can't start |
|  | [ "${NETWORKING}" = "no" ] && exit 0 |
|  |  |
|  | # Get the Jenkins sysconfig |
|  | [ -f /etc/sysconfig/jenkins ] && . /etc/sysconfig/jenkins |
|  | [ "${JENKINS\_HOME}" = "" ] && |
|  | JENKINS\_HOME=/usr/local/jenkins |
|  | [ "${JENKINS\_USER}" == "" ] && |
|  | JENKINS\_USER=jenkins |
|  |  |
|  | startup=${JENKINS\_HOME}/start-jenkins.sh |
|  | shutdown=${JENKINS\_HOME}/stop-jenkins.sh |
|  | export JAVA\_HOME=/usr/local/java/ |
|  |  |
|  | start(){ |
|  | echo -n $"Starting Jenkins service: " |
|  | pid=`ps -ef | grep [j]enkins.war | wc -l` |
|  | if [ $pid -gt 0 ]; then |
|  | echo "Jenkins is already running" |
|  | exit 1 |
|  | fi |
|  | su - $JENKINS\_USER -c $startup |
|  | RETVAL=$? |
|  | [ $RETVAL == 0 ] && |
|  | echo "Jenkins was started successfully." || |
|  | echo "There was an error starting Jenkins." |
|  | } |
|  |  |
|  | stop(){ |
|  | action $"Stopping Jenkins service: " |
|  | pid=`ps -ef | grep [j]enkins.war | wc -l` |
|  | if [ ! $pid -gt 0 ]; then |
|  | echo "Jenkins is not running" |
|  | exit 1 |
|  | fi |
|  | su - $JENKINS\_USER -c $shutdown |
|  | RETVAL=$? |
|  | [ $RETVAL == 0 ] && |
|  | echo "Jenkins was stopped successfully." || |
|  | echo "There was an error stopping Jenkins." |
|  | } |
|  |  |
|  | status(){ |
|  | pid=`ps -ef | grep [j]enkins.war | wc -l` |
|  | if [ $pid -gt 0 ]; then |
|  | echo "Jenkins is running..." |
|  | else |
|  | echo "Jenkins is stopped..." |
|  | fi |
|  | } |
|  |  |
|  | restart(){ |
|  | stop |
|  | sleep 5 |
|  | start |
|  | } |
|  |  |
|  | # Call functions as determined by args. |
|  | case "$1" in |
|  | start) |
|  | start;; |
|  | stop) |
|  | stop;; |
|  | status) |
|  | status;; |
|  | restart) |
|  | restart;; |
|  | \*) |
|  | echo $"Usage: $0 {start|stop|status|restart}" |
|  | exit 1 |
|  | esac |
|  |  |
|  | exit 0 |
|  |  |
|  |  |
|  |  |
|  |  |
|  | 8) |
|  | chown -R jenkins. /usr/local/jenkins |
|  | chmod a+x /usr/local/jenkins/start-jenkins.sh |
|  | chmod a+x /usr/local/jenkins/stop-jenkins.sh |
|  | chmod a+x /etc/init.d/jenkins |
|  |  |
|  |  |
|  | 9) dont use now |
|  | service jenkins status |
|  | service jenkins start |
|  | service jenkins restart |
|  | service jenkins stop |
|  | chkconfig jenkins on |
|  |  |
|  | 11) which java |
|  | /usr/bin/java |
|  |  |
|  | 10)vi /etc/sysconfig/jenkins |
|  | # Jenkins system configuration |
|  | JENKINS\_HOME=/usr/local/jenkins |
|  | JENKINS\_USER=jenkins |
|  | JENKINS\_LOG=/home/jenkins/jenkins.log |
|  | JENKINS\_JAVA=/usr/bin/java |
|  | JENKINS\_JAVAOPTS="" |
|  | JENKINS\_IP=0.0.0.0 |
|  | JENKINS\_PORT=8080 |
|  | JENKINS\_ARGS="" |
|  |  |
|  | 12 if u need vitual host in apache |
|  | <VirtualHost \*:80> |
|  | ServerName jenkins.doublesharp.com |
|  | DocumentRoot /var/www/html |
|  |  |
|  | ProxyPass / http://localhost:8080/ nocanon |
|  | ProxyPassReverse / http://localhost:8080/ |
|  | ProxyRequests Off |
|  | ProxyPreserveHost On |
|  | <Proxy http://localhost:8080/\*> |
|  | Order deny,allow |
|  | Allow from all |
|  | </Proxy> |
|  | # If using reverse proxy from SSL |
|  | #Header edit Location ^http: https: |
|  | </VirtualHost> |
|  |  |
|  |  |
|  | 13 |
|  | service jenkins stop or /etc/init.d/jenkins stop |
|  | service jenkins start or /etc/init.d/jenkins start |
|  |  |
|  |  |
|  | Jenikins config file will capture configuration parameters for the launch. |
|  | vi /etc/sysconfig/jenkins |
|  |  |
|  | defeult port |
|  | netstat -tnlp | grep 8080 |
|  | By default, Jenkins listen on port 8080 |
|  | open browser and login to http://localhost:8080/ |
|  |  |
|  | logs |
|  | tail -f /var/log/jenkins/jenkins.log |
|  |  |
|  |  |
|  |  |
|  | 14 |
|  | jenkins on windows or linux (http://www.linuxfunda.com/2014/07/06/how-to-install-and-configure-jenkins-on-centos-6-4-part-iii/) |
|  | --------- |
|  |  |
|  |  |
|  | to change shell |
|  | chsh --list-shells |
|  | chsh -s /bin/bash jenkins |
|  |  |
|  |  |
|  | Working with jenkins |
|  |  |
|  | 1) Click on Manage Jenkins > Configure Global Security > |
|  | Enable security |
|  | Jenkins Own User Database > Allow Users to Signup > Save |
|  |  |
|  | This below two step is needed if we use the unix groups as the database |
|  | chmod g+r /etc/shadow . |
|  |  |
|  | 2) Click on Signup button (Top right corner of the dashboard) > Fill all the required information > click on SignUp |
|  |  |
|  | 3) Now we will restrict User Signup and add “Matrix based security authentication”. Please follow the below steps: |
|  |  |
|  | Click on Manage Jenkins > Configure Global Security > Jenkins Own User Database > Unchek “Allow Users to Signup” > Matrix-based security > |
|  | Put the user Name @ “User/group to add:” text box > |
|  | click on add > |
|  | Check all the Check boxes > |
|  | Save |
|  | 4) Manage Jenkins > Manage Users > Create Users in order to create more user in future |
|  |  |
|  | 4.1) Click on “Manage Jenkins” > Click on “Configure System” |
|  | update the jdk home /usr/java/jdk1.7.0\_67 |
|  | update the git path , including the git command in the whole path /usr/local/git/bin/git |
|  | update the maven path |
|  | update Global MAVEN\_OPTS with -Xms512m -XX:MaxPermSize=128m |
|  |  |
|  | 5) How to add credentials. |
|  |  |
|  | setup a ssh credentils for user jenkins |
|  | log in to your Jenkins account. Click on the credential tab . |
|  | Now Click on “Add domain” localhost or hostname |
|  | Provide a user friendly name and some description to it and press “OK” |
|  |  |
|  |  |
|  | create credinatila for right users ( here sample as jenkins ) |
|  | --------------- |
|  | su jenkins |
|  | cd ~ |
|  | mkdir ~/.ssh |
|  | cd ~/.ssh |
|  | touch authorized\_keys |
|  |  |
|  | touch known\_hosts |
|  | chmod 700 ~/.ssh && chmod 600 ~/.ssh/\* |
|  |  |
|  | On the master jenkins machine do this for the jenkins user |
|  |  |
|  | ssh-keygen -t rsa -P "" |
|  |  |
|  | cat ~/.ssh/id\_rsa.pub>>~/.ssh/authorized\_keys |
|  |  |
|  | cat ~/.ssh/id\_rsa.pub>>~/.ssh/authorized\_keys |
|  |  |
|  | copy id\_rsa.pub in to other slaves node machines /.ssh/authorized\_keys file on that machine |
|  |  |
|  | After completing the Credential entry in the jenknins now we will add the Node |
|  |  |
|  | or choose the master .ssh option |
|  |  |
|  | 6) Add a new Node |
|  | Go to the main dashboard of Jenkins Click on “Manage Jenkins” > Click on “Manage Nodes” from the List. |
|  | To add a new node click on “New Node” ,chhose dump slave |
|  | include path provide a folder /var/www/html for example. |
|  | assosiate the ssh creadential we created above |
|  |  |
|  | Click on the advance and modify your port number if you are using any different port for your SSH. enter the ssh provate key |
|  | update the advanced property with java home |
|  | in the node configuration JavaPath has to be set to /opt/software/jdk1.7.0\_45/bin/java . give correct version |
|  | now launch slave agent |
|  |  |
|  |  |
|  | 7) add plugin |
|  | Add the necessary plugin |
|  | Manage Jenkins > Click on Manage Plugins > Click on update Tab > Chose the Git Plugin and download then and restart after install |
|  | add authorise plugin |
|  | add github |
|  | Maven Integration plugin |
|  | Junit plugin |
|  | Subversion plugin |
|  | ssh credential plugin |
|  | Label Linked Jobs Plug |
|  |  |
|  |  |
|  | Manage Jenkins > Click on Manage Plugins > Click on Available Tab > Filter the Git Plugins > |
|  | Chose the Git Plugin |
|  |  |
|  | Click on install without restart. |
|  |  |
|  |  |
|  | Create a job |
|  | 1) first jobs name as mvnprojectgraph |
|  | 2) choose maven project |
|  | 3) choose git repo http://github.com/cloudtiessolutions/mvnproject.git |
|  | 4) add crediential info@cloudties.in/info123$ ,choose global scope |
|  | 5) Restrict where t run job and give label ( node name as label) |
|  |  |
|  |  |
|  | Configure Global Security -> Access Control for Builds -> Configure Build Authorizations in Project Configuration -> Select Run as ...) 2/ In each upstream (or downstream - |
|  |  |
|  |  |
|  | 8) execute on the host where jenkits installed |
|  | git ls-remote -h https://github.com/cloudtiessolutions/java.git HEA D |
|  |  |
|  | 9) Create a new job Click on New Items > Give a Name to the Job > Choose “Build a free-style software project” option > Click on “Ok” |
|  | inside the project give the url like this |
|  |  |
|  |  |
|  | 10) check this option Restrict where this project can be run , also mentioned the host name in the label |
|  | in source code management choose git and provide the following |
|  | http://github.com/cloudtiessolutions/java.git |
|  |  |
|  | and provide username and password below in the same screen |
|  |  |
|  | reload job config |
|  | java -jar /home/jenkins/.jenkins/war/WEB-INF/jenkins-cli.jar -s http://localhost:8080 reload-configuration |
|  |  |
|  |  |
|  | 10) |
|  | vi /etc/ssh/ssh\_config |
|  | ForwardX11 yes |
|  | restart sshd |
|  |  |
|  |  |
|  |  |
|  | jenkins jnlp |