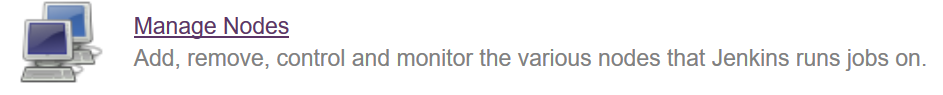
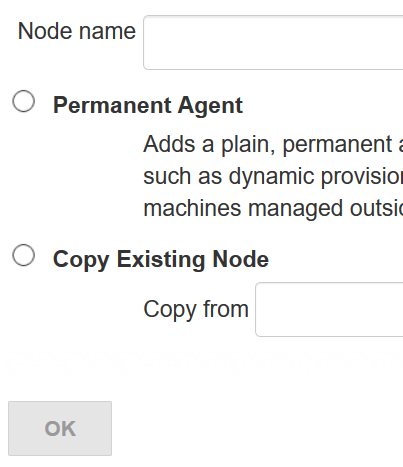
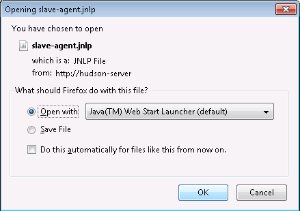
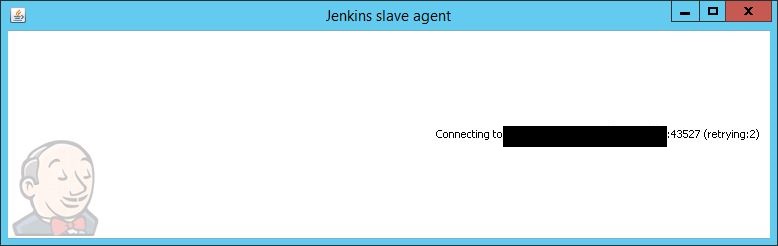
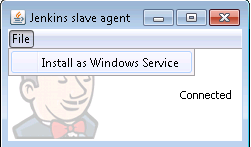
Step by step guide to set up master and slave machines on Windows

1. On your master machine go to **Manage Jenkins** > **Manage Nodes**.  
   
2. **New Node**
   1. **Enter Node Name**
   2. **Select Permanent Agent**
   3. Press **OK**.  
      
3. Fill out the following:
   1. Set a **number of executors**
      1. (one or more) as needed.
   2. Set a **Remote FS Root**
      1. a home directory for the master on the slave machine.
      2. For a *Windows slave*, use something like: "C:\Jenkins\"
   3. Select the appropriate **Usage** setting:
      1. For an additional worker: *Utilize this slave as much as possible*
      2. For specialized jobs: *Leave this machine for tied jobs only*
   4. **Launch Method**:
      1. An easy way to control a Windows slave is by using *Launch slave agents via Java Web Start*  (Recommended for Windows)
      2. TODO: add steps for other methods.
   5. **Availability**
      1. *Keep this slave online as much as possible*
      2. TODO: add details for each option.
   6. Press **OK**.  
      
4. Now you need to connect your slave machine to the master using the following steps.
   1. Open a browser on the **slave machine** and go to the **Jenkins master server** url (http://yourjenkinsmaster:8080).
   2. Go to **Manage Jenkins** > **Manage Nodes**,
      1. Click on the newly created slave machine. You will need to login as someone that has the "Connect" Slave permission if you have configured global security.
   3. Click on the **Launch** button to launch agent from browser on slave.  
      
   4. Run the program.  
        
      If Windows asks you to choose a program, use $JAVA\_HOME\bin\javaws.exe (the Java Web Start Launcher).
   5. If you encounter connection issue, then you could enlarge the popup windows to see the master **port used** and check your network configuration (firewall, port forward, ...).   
        
      Note that Jenkins chooses a random, high-number port. If you want Jenkins to use a static port, go to "Manage Jenkins" > "Configure Global Security" and choose the port number in the "TCP port for JNLP agents" box.
   6. If the port is open, the agent still can't connect, and your Jenkins instance is served securely over SSL/HTTPS, download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8. Take the .jar files and save them to $JAVA\_HOME\lib\security. Try the above steps again.
   7. Now you should see the Slave machine connected under **Nodes**.

https://wiki.jenkins.io/download/thumbnails/41878276/Node.PNG?version=1&modificationDate=1509727464000&api=v2

1. If you want the service to run on start-up of the slave machine do the following (Windows only directions):
   1. In the Slave agent program running on your slave machine,
   2. click **File** --> **Install as Windows Service.**  
        
      Note that this feature requires ".Net Framework 3.5"  
      https://wiki.jenkins.io/download/attachments/41878276/dotNet3_5Features.JPG?version=2&modificationDate=1426780066000&api=v2
   3. **Start**, type Services and Select the **Services** program.
   4. Find **Jenkins Slave** in the list, Double click to open.
   5. Select **Startup type** --> **Automatic**.
   6. Go to the **Log On** tab, change the **Log on as** to a user of your choice (Special user account Jenkins recommended).
   7. Make sure that auto login is set for the slave machine for the user account, then the VM (or physical computer) should connect and be available when needed.

# How to Connect to Remote SSH Slaves?

# ssue

* Connection of SSH slaves agents or nodes with different users.
* Using Certificates (SSH Private Key) for connecting to slaves agents or nodes.

# Environment

* CloudBees Jenkins Enterprise
* Dump slave/agents Launch method via **SSH**

# Resolution

Ideally, we recommend that the same user jenkins is created on each Server where slaves agents or nodes will be deployed. However, it is possible to run slaves with a different user. Indeed, **each slave acts as an SSH server and the Jenkins master acts as the SSH Client**.

## Requirements

That being said, here are the requirements to make it work:

* The slave needs to authorize the master to access via SSH for a specific user.
* The master needs a way to authenticate with a specific user to the slave via SSH.

So, a SSH server needs to be installed in the slaves agents or nodes (For instance, on Ubuntu ???openssh-server)

Before trying to connect a slave to a master. Make sure of the following:

a. From the Master, there is connectivity to the Slave

> ping $MASTER\_IP

b. From the Slave, there is connectivity to the Master

> ping $SLAVE\_IP

Expected output for “a” and “b” would be similar to:

PING 192.168.1.86 (192.168.1.86) 56(84) bytes of data.

64 bytes from 192.168.1.86: icmp\_seq=1 ttl=64 time=0.417 ms

64 bytes from 192.168.1.86: icmp\_seq=2 ttl=64 time=0.369 ms

64 bytes from 192.168.1.86: icmp\_seq=3 ttl=64 time=0.419 ms

64 bytes from 192.168.1.86: icmp\_seq=4 ttl=64 time=0.406 ms

^X64 bytes from 192.168.1.86: icmp\_seq=5 ttl=64 time=0.587 ms

64 bytes from 192.168.1.86: icmp\_seq=6 ttl=64 time=0.309 ms

64 bytes from 192.168.1.86: icmp\_seq=7 ttl=64 time=0.278 ms

^C

--- 192.168.1.86 ping statistics ---

7 packets transmitted, 7 received, 0% packet loss, time 6000ms

c. From the Master, you can ssh to the Slave

> ssh $MASTER\_IP

Expected output for “c” would be similar to:

ECDSA key fingerprint is SHA256:rX96v9WGIabHecNJusEYhXvENKyKe6GzQ/B/5ozRH9Q.

Are you sure you want to continue connecting (yes/no)?

## Steps

In order to connect an SSH slave with the user $SLAVE\_USER, follow these steps:

### On the slave machine:

1) Log in as $SLAVE\_USER. We create the SSH Keys with that user to ensure it has the required permissions later on when connecting to the slave from Jenkins.

> sudo su $SLAVE\_USER

2) Create private and public SSH keys. The following command creates the private key $myslave\_rsa and the public key$myslave\_rsa.pub. It is recommended to store your keys under ~/.ssh/ so we move to that directory before creating the key pair.

> cd ~/.ssh/; ssh-keygen -t rsa -C "$myslave" -f "$myslave\_rsa"

**Note**: To increase security, a passphrase is advisable to associate it to your Private Key.

3) Add the public SSH key to the list of authorized keys in the slave machine

> cat $myslave\_rsa.pub >> ~/.ssh/authorized\_keys

4) Copy the private SSH key from the Slave machine to your Master. Two options here:

* 4.a Copy the content by using the OS clipboard (in linux, xclip). SSH private key should be similiar to this:

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: AES-128-CBC,DBA2145CD0590DE762D19A306B3AE233

7pzgoksx5ZivR/awfo2wX4zHROigUEzfFUl0jZryVK5Pzx3Ubr9gc7xLHQ8jo7p2

...

Ewuk0EAiN1qBZWutdBOxxUnopLsML7SFkzeSXuRjdHQJR76V5WBclnxkS2AK9y6q

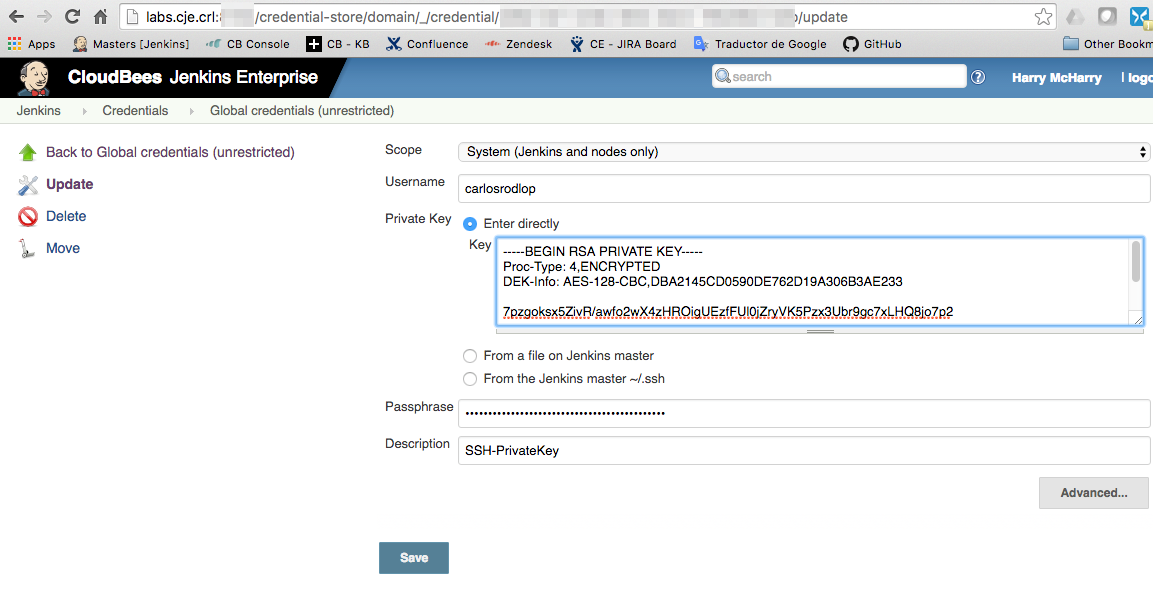
-----END RSA PRIVATE KEY-----

* 4.b Copy the full file. For instance, if you have installed scp on your Slave Server:

> scp ???/home/$SLAVE\_USER/$myslave\_rsa $MASTER\_USER@$MASTER\_IP:/home/$MASTER\_USER/.ssh

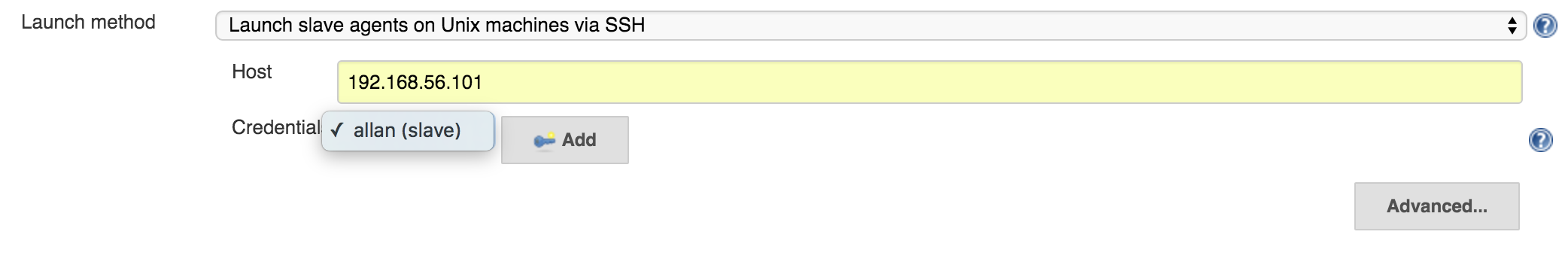
### In Jenkins:

5) Go to the configuration of credentials under Manage Jenkins/Manage Credentials/Add Credentials (JENKINS\_URL/credential-store/) and choose SSH username with private key.



According to the option selected on “step 4”, select now the proper Private Key type. So for “4.a” choose Enter Directly, whereas for “4.b” choose From a file on the Jenkins Master

6) For every slave that you want to connect using the user $SLAVE\_USER, configure the slave under Manage Nodes/$NODE\_NAME/configure ( $JENKINS\_URL/computer/$NODE\_NAME/configure) and use this user credential.



**Note**: Using that configuration, only one credential per user can be used. Several slaves can be launched for the same user on the same machine without any further configuration. To connect slaves hosted on a different machine using the same credential, the same SSH public Key needs to be added in the authorized\_keys logged as $SLAVE\_USER

7) On every slave you connected successfully, a similar output to this one is expected:

Evacuated stdout

Slave successfully connected and online

[07/15/16 17:49:14] [SSH] Opening SSH connection to 192.168.1.85:22.

[07/15/16 17:49:14] [SSH] Authentication successful.

[07/15/16 17:49:14] [SSH] The remote users environment is:

BASH=/bin/bash

BASHOPTS=cmdhist:complete\_fullquote:extquote:force\_fignore:hostcomplete:interactive\_comments:progcomp:promptvars:sourcepath

BASH\_ALIASES=()

BASH\_ARGC=()

BASH\_ARGV=()

BASH\_CMDS=()

BASH\_EXECUTION\_STRING=set

BASH\_LINENO=()

BASH\_SOURCE=()

BASH\_VERSINFO=([0]="4" [1]="3" [2]="42" [3]="1" [4]="release" [5]="x86\_64-pc-linux-gnu")

BASH\_VERSION='4.3.42(1)-release'

DIRSTACK=()

EUID=1000

GROUPS=()

HOME=/home/carlosrodlop

HOSTNAME=carlosrodlop-VirtualBox

HOSTTYPE=x86\_64

IFS=$' \t\n'

LANG=es\_ES.UTF-8

LOGNAME=carlosrodlop

MACHTYPE=x86\_64-pc-linux-gnu

MAIL=/var/mail/carlosrodlop

OPTERR=1

OPTIND=1

OSTYPE=linux-gnu

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

PIPESTATUS=([0]="0")

PPID=3777

PS4='+ '

PWD=/home/carlosrodlop

SHELL=/bin/bash

SHELLOPTS=braceexpand:hashall:interactive-comments

SHLVL=1

SSH\_CLIENT='192.168.1.86 53746 22'

SSH\_CONNECTION='192.168.1.86 53746 192.168.1.85 22'

TERM=dumb

UID=1000

USER=carlosrodlop

XDG\_RUNTIME\_DIR=/run/user/1000

XDG\_SESSION\_ID=3

\_=']'

[07/15/16 17:49:14] [SSH] Checking java version of java

[07/15/16 17:49:14] [SSH] java -version returned 1.7.0\_80.

[07/15/16 17:49:14] [SSH] Starting sftp client.

[07/15/16 17:49:14] [SSH] Copying latest slave.jar...

[07/15/16 17:49:14] [SSH] Copied 506,667 bytes.

Expanded the channel window size to 4MB

[07/15/16 17:49:14] [SSH] Starting slave process: cd "/opt/Labs/slaves/lab3/ssh3" && java -jar slave.jar

### On the slave machine:

On the hand, under the path specified in Remote root directory, slave.jar should has been copied. And those item/jobs which use this slave agent/node will allocate their workspace here after they are built.

**NOTE:**  
Depending on the permissions of the node/slave machine’s user account, it may be necessary to set the jar cache directory - due to the issue JENKINS-18578  
The default jar cache location is hardcoded to ~/.jenkins/cache/jars

To change the default setting:

In the slave configuration, under the Advanced method of Launch Option,  
Place the following in the Suffix Start Slave Command -jar-cache <path to jar cache directory>

Make sure there’s a leading space so the parameters aren’t tacked directly onto the slave.jar itself.