**You will need:**

* A Jenkins installation to act as the master
* A server to act as an agent node

I based this guide upon Jenkins version *2.89.4 LTS*. I’ll provide some commands that can be used to do all of these steps on a CentOS 7 machine.

Step 1: Install the necessary packages.

You will need to install some packages on the agent node, such as Java, as well as any tools that you may require to run your builds (Git, Maven, Ant, Gradle, etc.). I’ll install *Java* and *Git*:

sudo yum -y install java-1.8.0-openjdk git

Step 2: Create a user on the agent to be used by Jenkins.

Now we need to create a user on the agent. The Jenkins master will log into the agent as this user, and all build jobs will execute as this user. It makes sense to call this user *jenkins*, and we’ll give them a special home directory where the Jenkins agent’s files will reside:

sudo useradd -d /var/lib/jenkins jenkins

Step 3: Generate an ssh key.

Next, we need to generate an ssh key. Jenkins will use this key to authenticate with the agent node and log in as the *jenkins* user. This key can generate on practically any Linux machine, but you can also do it on the agent node itself:

ssh-keygen

This command will provide a series of prompts. Feel free to use the defaults. From a security perspective, it may be a good idea to enter a passphrase when prompted, and if you do so make sure you take note of the passphrase you used since you will need it later. This command will output two files: *id\_rsa* and *id\_rsa.pub*. If you’re already familiar with the process of working with ssh keys in Linux, this will be very familiar to you, but if not, don’t worry. I’ll walk you through how to use these keys! By default, these files are located in *~/.ssh/*. When you run *ssh-keygen*, you have the option of specifying another location.

Step 4: Add the public key to the *authorized\_keys* file of the *jenkins*user on the *agent* node.

Next, we need to add the public key (the contents of*id\_rsa.pub*) to the *authorized\_keys* file of the *jenkins* user that we created earlier. This will allow anyone who has the private key (in our case, the Jenkins master) to log in to the *agent* node as the *jenkins* user. First, we need to create the directory that the *authorized\_keys* file belongs in, and then we can create the file itself:

sudo mkdir /var/lib/jenkins/.ssh

sudo vi /var/lib/jenkins/.ssh/authorized\_keys

Copy the entire contents of the *id\_rsa.pub* file that you generated earlier (*~/.ssh/id\_rsa.pub*by default) and paste it into *authorized\_keys*, then save it.

Step 5: Add the *agent* node via the Jenkins UI.

Now we’re ready to finish setting up the node via the Jenkins UI. In Jenkins, go to **Manage Jenkins**, then **Manage Nodes**, then click **New Node**. Here you can give your node a name, then select **Permanent Agent** and click **OK**.

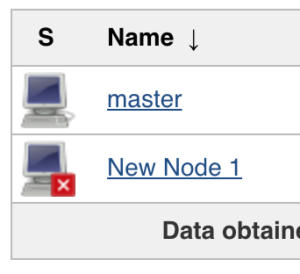


There are a variety of options you can use here to customize your node. All we care about right now is the **Launch Method**.

1. Select **Launch Slave Agents via SSH** for **Launch Method**.
2. Enter the hostname or IP address of your *agent* node in the *Host* field.
3. Click the **Add** button next to *Credentials*and select the **Jenkins** scope.
4. For the credential, set **Kind** to **SSH username with private key**.
5. Enter *jenkins* for the username.
6. For the private key, select **Enter directly**. Then, copy the contents of your private key file (*~/.ssh/id\_rsa* by default) and paste it into the **Key** box.
7. If you used a passphrase when generating the key, enter that for **Passphrase**, otherwise, leave it blank.
8. Enter a *descriptive id* and *description*, then click **Add**.
9. Click **Save** to save the new node.

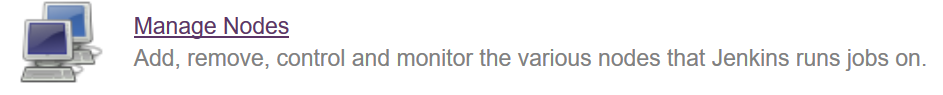
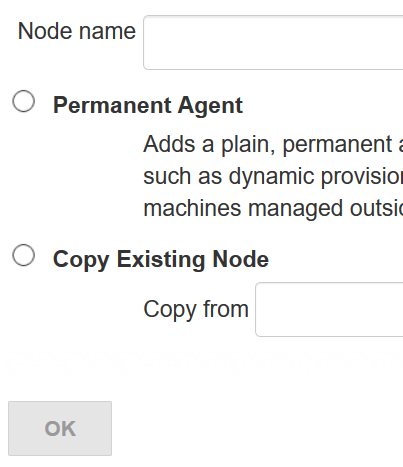
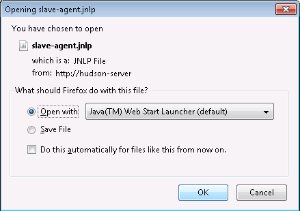
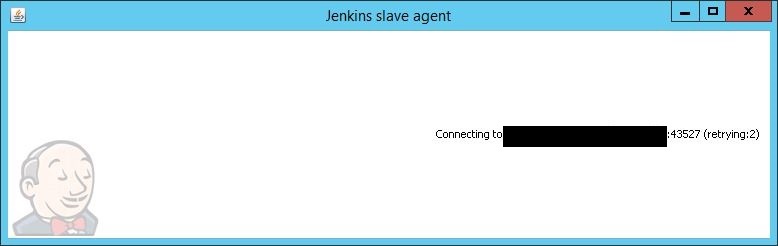
Step 6: Make sure everything is working.

Your new node should now appear in the list of nodes. You may notice a red X on the node’s icon. This indicates that it is not connected, but that’s because we just added it and it hasn’t had a chance to connect yet.

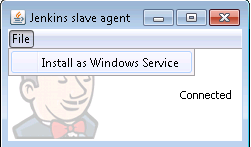


Wait a few seconds and refresh the page, and the red X will go away, indicating that the node is connected.

# Step by step guide to set up master and agent 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 agent machine.
      2. For a *Windows agent*, use something like: "C:\Jenkins\"
   3. Select the appropriate **Usage** setting:
      1. For an additional worker: *Utilize this node 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 agent is by using *Launch agent via Java Web Start*  (Recommended for Windows)
      2. TODO: add steps for other methods.
   5. **Availability**
      1. *Keep this agent online as much as possible*
      2. TODO: add details for each option.
   6. Press **OK**.  
      
4. Now you need to connect your agent machine to the master using the following steps.
   1. Open a browser on the **agent 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 agent machine. You will need to login as someone that has the "Connect" Agent permission if you have configured global security.
   3. Click on the **Launch** button to launch agent from browser on agent.  
      
   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 agent 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 agent machine do the following (Windows only directions):
   1. In the agent program running on your agent 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 Agent** 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 agent machine for the user account, then the VM (or physical computer) should connect and be available when needed.

node (params.slave) {   
    stage('test'){   
        sh 'terraform --version'   
    }   
}

stage('Building Win64, Linux764, MacOS') {

parallel {

stage('Win64') {

agent {

label 'win-10-x64'

}

steps {

...

}

}

stage('Linux64') {

agent {

label 'linux-x64'

}

steps {

...

}

}

stage('MacOS') {

agent {

label 'macos'

}

steps {

...

}

}

}

}