

Software Prerequisites

To run the lab you need a few applications loaded on your laptop.

These tools are Vagrant and VirtualBox. This tool set is very commonly used in a development environment. It makes common tasks such as bringing up a VM, provisioning it, and building multi-box topologies a breeze.

Vagrant

Vagrant is a virtual machine management tool. It manages the downloading, provisioning, and state management of the virtual machine. It is a common tool used in software development.

Download Locations

- [Mac OS X](#)
- [Windows](#)
- [Linux 64-bit DEB](#)
- [Linux 32-bit DEB](#)
- [Linux 64-bit RPM](#)
- [Linux 32-bit RPM](#)

Installation Instructions

Please download and install the correct versions

Install this executable on your operating system of choice just as you would any software for that platform. On Windows, you will need to reboot, but you can wait until after VirtualBox is installed.

Vagrant Plugins

To provide additional features into Vagrant we must first install a few additional plugins for Vagrant. The first plugin "vagrant-junos" is a driver to help configure and control Junos. The second plugin, "vagrant-host-shell", is used to assist in the configuration of Junos.

The vSRX boxes require two plugins, namely:

- [vagrant-junos](#)
- [vagrant-host-shell](#)

These can be installed with the following commands (launch a command prompt on Windows (`cmd`), or a Terminal session on Mac (located in your Applications/Utilities folder):

Installation Instructions

This will fetch the latest version of the vagrant plugins from the Internet. Both plugins are less than 1MB in size so the installation should be simple.

```
vagrant plugin install vagrant-junos
vagrant plugin install vagrant-host-shell
```

To check for updates in the future, a simple `vagrant plugin update` will ensure everything is up-to-date.

VirtualBox

For the lab we will be using VirtualBox as our virtualization manager. This tool is free to use and most importantly it is free to use with Vagrant. It also offers some support that VMware Workstation or Fusion do not offer. This includes support for the VirtIO driver and the ability to more simply manage virtual networks.

When installing on Windows it will typically install a network driver. It will prompt you to click "Continue" to take this action. Please hit continue to install the driver.

- [Mac OS X](#)
- [Windows](#)
- [Linux 64-bit DEB](#)
- [Linux 32-bit DEB](#)
- [Other Linux](#)

Installation Instructions

Please download and install Virtual Box before continuing.

Windows users: you will need to reboot your laptop before Vagrant and VirtualBox will both work.

TODO

Installing Boxes

Normally, Vagrant boxes are added with `vagrant box add vendor/name` from the web, but for the size of this class, we need to install them locally. You will still be able to type `vagrant box outdated` to get new versions of these boxes in the future.

NOTE: Vagrant stores boxes in `$HOME/vagrant.d` on Linux & Mac, or `%userprofile%\vagrant.d\boxes` on Windows

Vagrant boxes can be installed with the following commands (launch a command prompt on Windows (`cmd`), or a Terminal session on Mac (located in your Applications/Utilities folder):

Installation Instructions

When you attempt to bring up the virtual machines for the first time Vagrant will automatically download the correct boxes for you. However you may want to run the class on an airplane or a location with unreliable Internet access. If that is the case then you can also pre download the machines using the vagrant command.

```
vagrant box add juniper/netdevops-ubuntu1404-headless
vagrant box add juniper/ffp-12.1X47-D20.7
```

Validating Install

- `vagrant version` should tell you that you're running 1.7.3, aka the latest and greatest
- `vagrant plugin list` should include:
 - `vagrant-host-shell` 0.0.4 or newer
 - `vagrant-junos` 0.2.0 or newer

Vagrant Global Status

With the workflow of vagrant it is possible to have many virtual machines running without you being aware of this happening. The only indication of this being a problem is your laptop will begin to run very slowly. To see if you have vagrant hosts running you can use the command "vagrant global-status"

```
[rcameron:~/code/JNPRAutomateDemo-Class] master(+31/-0) ± vagrant global-st
id          name      provider    state    directory
-----
86a431a     default  vmware_fusion not running /Users/rcameron/vagrant/ffp-1
4407dc1     trusty1   virtualbox  saved    /Users/rcameron/vagrant/3host
55b10f6     default  virtualbox  saved    /Users/rcameron/vagrant/happy
7879af8     default  virtualbox  saved    /Users/rcameron/vagrant/preci
b67e217     default  virtualbox  saved    /Users/rcameron/vagrant/ffppm
2c59020     ndo       virtualbox  running  /Users/rcameron/code/JNPRAuto
ec8e606     srx       virtualbox  running  /Users/rcameron/code/JNPRAuto
```

The above shows information about all known Vagrant environments on this machine. This data is cached and may not be completely up-to-date. To interact with any of the machines, you can go to that directory and run Vagrant, or you can use the ID directly with Vagrant commands from any directory. For example:

```
"vagrant destroy 1a2b3c4d"
```

```
[rcameron:~/code/JNPRAutomateDemo-Class] master(+31/-0) ±
```

Run command to stop running machines

You must get the ID of the running VM. This is the hex code in the first column. Simply copy and paste or type that to run the suspend command against that host.

```
vagrant suspend ec8e606
==> srx: Saving VM state and suspending execution...
[rcameron:~/code/JNPRAutomateDemo-Class] master(+52/-0) 7s ±
```

Optional Text Editor

During the class we are going to be editing or reviewing a set of files during the lab. You are welcome to use any text editor that your are familiar with. However if you do not have an editor in mind then please use the Atom editor from Github. It is a high-end and open source editor that can be completely customized for your needs.



Atom Downloads

- [Windows](#)
- [Mac](#)
- [Ubuntu Linux](#)
- [Redhat/Centos Linux](#)