Linux Academy

Learning Vagrant

Introduction to Vagrant

Learning Vagrant

So what is Vagrant?

- Creates virtual machines
- Will provision the virtual machines via shell script or configuration management using tools like Chef and Puppet
- Host and Guest specific editing to work around known issues such as the known issue of ubuntu 12.04 breaking networking defaults, therefore Vagrant will automatically make modification needed to the ubuntu configs. Vagrant does this sort of fix on many host and guest setups.
- (HOWEVER) this is all done in minutes!
- Managing large amounts of nodes becomes expensive and unmanageable

Learning Vagrant

Let Vagrant handle the entire lifecycle of development machines

- Suspend, Halt, Resume virtual machines
- Destroy your virtual machines and delete its metadata
- SSH in to your machine
- Package up your machines entire state and the re distribute it to other development team members

Vagrant is a must have tool in development environments in which is allows us to use automation of an environment that will resemble your actual production

Learning Vagrant

Why should we use Vagrant?

- Traditionally we would have to manually install and setup things like, Apache and MySQL locally on our development workstations
- This can be very complex with all amount of moving parts to keep track of
- Human error is easy to take place when setting up complete local development environments

Vagrant is an awesome way of fixing these issues by automating our setup of all these services needed to develop our applications. Each project gets its own virtual machines. Working within you team is snap because we can share our virtual machine images by telling sharing a simple command that will build an exact replica of that environment in minutes.

The Vagrant Work-Flow

- Developers can now check out versions or repositories from a version control such as GIT and run ‘vagrant up”.
- Now the developers can work on their own machines comfortably with their own, editors, browsers and tools.
- We have a consistent and stable development environment now
- The same scripts of configuration in using vagrant to deploy dev environments is the same used in production therefore the dev environment is as close as possible to the production limiting issues later on.

The Vagrant Work-Flow

- If something was to go horribly wrong and we needed to start over from scratch you simply run a “vagrant destroy”, that will remove all the traces of that development from your machines and then follow it with another “vagrant up” which will re-create the entire dev environment in moments.

- We are able to keep the dev systems clean by suspending, halting, and destroying our dev environments.

- In terms of cost we no longer have to worry about the developer forgetting to shut down an instance in our cloud at AWS that is costing us by the hour.

- Every project and the knowledge within that project can transfer when the work-flow is followed the same way eliminating issues between machines.

Linux Academy

Learning Vagrant

Virtualization Overview

Virtualization

- Desktop Virtualization
- Containers
- Cloud

Linux Academy

Learning Vagrant

VirtualBox Installation

OS X

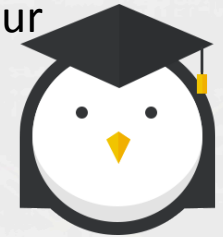
Install VirtualBox on Mac OS X hosts

1. Double-Click on the file that you have downloaded .dmg file from www.virtualbox.com

2. A window will open telling to double click on the VirtualBox.mpkg installer file displayed in that same window.

3. This will start the installer, which will allow you to select where to install VirtualBox to.

After installation you can find VirtualBox icon in the “Applications” folder in your finder window.



Linux Academy

Learning Vagrant

VirtualBox Installation

Windows

Installing VirtualBox on Windows:

- After downloading the 32bit or 64bit executable you can run through your typically point and click default installation.

Linux Academy

Learning Vagrant

VirtualBox Installation

Linux

Ubuntu based systems:

```
sudo apt-get install dkms
```

Fedora, RedHat:

```
yum install dkms
```

Linux Academy

Learning Vagrant

Installing Vagrant

OSX

Linux Academy

Learning Vagrant

Installing Vagrant

Windows

Linux Academy

Learning Vagrant

Installing Vagrant

Linux

Centos, and RedHat based systems:

Download the correct RPM from vagrantup.com

sudo rpm -i vagrant-1.6.5.rpm

Linux Academy

Learning Vagrant

Vagrant without VirtualBox

Learning Vagrant

You can actually use Vagrant with other “providers” instead of just VirtualBox

- VMware
- AWS EC2
- And just about any other provider out there!

In this course we are going to be primarily using VirtualBox because it’s free.

*Vagrant offer an official VMware provider for cost ww.virtualbox.com/vmware

Linux Academy

Learning Vagrant

Our First Vagrant Machine

Learning Vagrant

Essentially up and running with just two commands.

Vagrant init

Vagrant up

Linux Academy

Learning Vagrant

Vagrantfiles

Learning Vagrant

Vagrant is configured on a per project basis

Each of these projects has its own ‘ Vagrantfile ‘

The Vagrant file is a text file in which vagrant reads that sets up our environment

There is a description of what OS, how much RAM, and what software to be installed etc.

You can version control this file so your development team can stay in sync on changes

Let’s create our own Vagrantfile now!

LOOKUP PATH

When you run any vagrant command, Vagrant climbs your directory tree starting first in the current directory you are in. Example:

/home/stephen/projects/la/Vagrantfile
/home/stephen/projects/Vagrantfile
/home/stephen/Vagrantfile
/home/Vagrantfile
/Vagrantfile

Learning Vagrant

Vagrantfiles have what's called configuration versions1.0.X and the newest configuration versions 1.1.+

```
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
      config.vm.box = "precise64"
end
```

As of date the current config version of vagrant is “2”

There are currently only 2 versions supported. “1” and “2”
Version “1” config files for Vagrant version 1.0.X
Version “2” config files for Vagrant version 1.1+ leading up to 2.0.x

You can specify Vagrant version requirements in the Vagrantfile to enforce other users in your team to use a specific version of Vagrant.

Vagrant.require_version “>= 1.3.5”

In this example the Vagrantfile will only load if the version loading it is Vagrant 1.3.5 or greater.

Linux Academy

Learning Vagrant

Boxes

Learning Vagrant

- Vagrant boxes are just ‘Templates’
- Boxes contain our base operating systems already setup
- Manage Boxes such as:
  - Vagrant box list
  - Vagrant box add
  - Vagrant box remove

Linux Academy

Learning Vagrant

Vagrant Up – Running Vagrant Machines

Learning Vagrant

Version Control

Vagrant creates a directory called .vagrant/ within our project directories that used to maintain some state for Vagrant.

It keeps track of guest machine:

- IDs
- locks
- Configurations and more

- With each Vagrant Up that is run the .vagrant/ directory should be ignored by your version control system such as when using GIT.

Linux Academy

Learning Vagrant

SSH – Accessing Our Vagrant Virtual Machines

Working with Vagrant Machines:

Linux Academy

Learning Vagrant

Synced Folders

Learning Vagrant

Folders can be shared between the hosts and the guest virtual machine:

- Use our own editors of choice
- By default this is done by sharing your Project directory to a folder on the guest machine in /vagrant
- We can actually change this behavior to where and what we want

- **config.vm.synced_folder “src/”, “/server/website”**
  end

```
Vagrant.configure("2") do |config|
    # other config here

    config.vm.synced_folder "src/", "/srv/website"
End
```

There are several parameters that we can pass in the configs in order to configure synced folders. We are going to go over some of the basic ones now.

ENABLING

These are setup by default when doing:

vagrant up

Or

vagrant reload

DISABLING

```
Vagrant.configure("2") do |config|
    config.vm.synced_folder "src/". "/server/website", disabled: true
end
```

Modifying the Owner/Group

```
config.vm.synced_folder “src/”, “/server/website”,
        owner: “root”, group: “root”
```

Linux Academy

Learning Vagrant

Basic Networking

Basic Networking

- Vagrant automatically sets up networking so that developers can start working right away on the virtual machine.

- Forwarding Ports:
  - A forwarded port exposes a port on the guest machine as a port on a host machine. Today let's do port 80 for a web server as an example. We would modify our Vagrantfile to look like the following:

config.vm.network "forwarded_port", guest: 80, host: 8080

After the Vagrantfile is edited we would run a

vagrant reload

Linux Academy

Learning Vagrant

Environment Management

Vagrant Teardown?

Vagrant in one simple command can:

- Suspend
- Halt
- Destroy

Suspend

Can be thought of as “freezing time” within our virtual machine.

Suspending allows you to quickly clean up and quickly resume.

vagrant suspend

Can be thought of as “freezing time” within our virtual machine.

Suspending allows you to quickly clean up and quickly resume.

After we suspend our virtual machine no more CPU or RAM resources are consumed on the host machine for the Vagrant environment, but disk space continues to be used.

vagrant suspend

vagrant halt

Will cleanly shut down your virtual machine

vagrant halt --force

If you need to shut down the server without a clean shut down we can force it.

vagrant destroy

This will completely remove our machine from the environment.

Linux Academy

Learning Vagrant

Provisioners

Apache setup with a script

Provisioning and Provisioners

Base Vagrant boxes are bare. Only the minimal function such as SSH is installed after a base box install.

We have the option to install software on our base box but that would have to be done via shell scripts, configuration management systems, or good ole manual command-line entry installations.

Vagrant allows us to do automatic provisioning. This is done when we run a ‘vagrant up’.

Provisioners can be used such as: Shell scripts, Chef, Puppet. There are also Several plug-ins available too.

Automatic Provisioning

Shell Scripts

Chef

Puppet

vagrant plugins

Linux Academy

Learning Vagrant

Provisioners

Using Chef

Linux Academy

Learning Vagrant

Provisioners

Using Puppet

Linux Academy

Learning Vagrant

Networking

Private Networking

Private Networking

Private addresses that are not accessible from the public internet

DHCP

We can use DHCP to assign these IP addresses within these private networks automatically

```
Vagrant.configure("2") do |config|
  config.vm.network "private_network", type: "dhcp"
end
```

STATIC IP

We can use DHCP to assign these IP addresses within these private networks automatically

```
Vagrant.configure("2") do |config|
  config.vm.network "private_network", ip: "192.168.70.4"
end
```

Disable Auto-Configuration

```
Vagrant.configure("2") do |config|
  config.vm.network "private_network", ip: "192.168.70.4",
    auto_config: false
end
```

Linux Academy

Learning Vagrant

Networking

Public Networking

Public Networking

DHCP

We can use DHCP to assign these IP addresses within these public networks automatically

```
Vagrant.configure("2") do |config|
  config.vm.network "public_network"
end
```

STATIC IP

```
config.vm.network "public_network", ip: "192.168.0.17"
```

Default Interface

```
config.vm.network "public_network", bridge: 'en1: Wi-Fi (AirPort)'
```

Linux Academy

Learning Vagrant

Running Multiple Machines

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "precise64"
  config.vm.box_url = "http://files.vagrantup.com/precise64.box"

    # Setup for Web Server
    config.vm.define "web" do |web|
        web.vm.hostname = "web"
        web.vm.box = "apache"
        web.vm.network "private_network", type: "dhcp"
        web.vm.network "forwarded_port", guest: 80, host: 8080
        web.vm.provision "puppet" do |puppet|
         puppet.manifests_path = "manifests"
         puppet.manifest_file  = "default.pp"
    end
end
   # Setup for MYSQL DB SERVER
  config.vm.define "db" do |db|
      db.vm.hostname = "db"
      db.vm.box = "mysql"
      db.vm.network "private_network", type: "dhcp"
  end
end
```

Linux Academy

Learning Vagrant

Running Multiple Machines

Linux Academy

Learning Vagrant

Boxes

Learning Vagrant

- ✓Custom Boxes
  - ✓Create your linux distro of choice
  - ✓Pre Bake your applications on these boxes

Learning Vagrant

- ✓ **Why does Vagrant use boxes?**
  - ✓ **Saves time**
  - ✓ **Gives us disposable infrastructures**
  - ✓ **We get portability with our boxes**

✓**Box Format**

```
$ tree
.
├── Vagrantfile
├── box-disk1.vmdk
├── box.ovf
└── metadata.json

0 directories, 4 files
```

Linux Academy

Learning Vagrant

Basic Box Management

Learning Vagrant

- **Basic Box Management**
- **Managed globally per user not per project**
- **Boxes are mapped in Vagrant to a logical name in which you name**
  - ✓ **This name mapping maps to our config.vm.box settings in our Vagrantfile to the actual Box we are building our machine from.**
- **Boxes are just Files**

Learning Vagrant

```
$vagrant box
Usage:  vagrant box <comand> [<args>]

Available subcommands:
        add
        list
        remove
        repackage
```

vagrant box add

```
$ vagrant box add precise64 http://files.vagrantup.com/precise64.box
Downloading with Vagrant::Downloaders::HTTP...
Downloading box: http://files.vagrantup.com/precise64.box
Extracting box...
Cleaning up downloaded box...
Successfully added box 'precise64' with provider 'virtualbox'!
```

vagrant box list

Precise64 (virtualbox)

vagrant box remove

```
$ vagrant box remove precise64 virtualbox
Removing box 'precise64' with provider 'virtualbox'...
```

vagrant box repackage

- ✓Will re package our box in to the original *pacakge.box* file

Linux Academy

Learning Vagrant

Creating Boxes from An Existing Environment

Linux Academy

Learning Vagrant

Creating Your Own Boxes

Building Boxes From Scratch

- Create a brand new Virtual Machine inside VirtualBox
- Use dynamically sized drives and set appropriate levels on RAM
- remove un necessary components
- Install from scratch the Operating System of choice

Configure the Operating System

a Vagrant user must exist

SSH server must be installed and configured to run on system boot

For Vagrant defaults be sure to use the public vagrant key

We can customize this however different in the config.ssh.private_key_path to point to our own private key if we wish

```
$ mkdir /home/vagrant/.ssh
$ chmod 700 /home/vagrant/.ssh
$ cd /home/vagrant/.ssh
$ wget --no-check-certificate 'https://raw.github.com/mitchellh/vagrant/master/keys/vagrant.pub' -O authorized_keys
$ chmod 600 /home/vagrant/.ssh/authorized_keys
$ chown -R vagrant /home/vagrant/.ssh
```

In ssh config be sure to also disable *requiretty*

At this point we would now need to install Guest additions.

Linux based OS will need kernel headers and build tools
For example on Ubuntu and mounting the iso you installed ubuntu with:

sudo mount /dev/cdrom /media
…
sudo sh /media/cdrom/VBoxLinuxAdditions.run

Additional Software for our Box

Install the software and configure it how you like

You can even pre setup and have the box ready for Chef and Puppet

Avoid the well known linux virtual machine network device issue with:

```
$ rm /etc/udev/rules.d/70-persistent-net.rules
$ mkdir /etc/udev/rules.d/70-persistent-net.rules
$ rm -rf /dev/.udev/
$ rm /lib/udev/rules.d/75-persistent-net-generator.rules
```

Be sure to Minimize your Final Box size!

Uninstall unnecessary packages
Remove build tools and linux-headers

Zero out the virtual hard drive with zeroes ad delete the zero-filled file.

Example on Ubuntu:

```
$ dd if=/dev/zero of=/EMPTY bs=1M
...
$ rm -f /EMPTY
```

Be sure to Minimize your Final Box size!

Uninstall unnecessary packages
Remove build tools and linux-headers

Zero out the virtual hard drive with zeroes ad delete the zero-filled file.

Example on Ubuntu:

```
$ dd if=/dev/zero of=/EMPTY bs=1M
...
$ rm -f /EMPTY
```

Packaging it all Up

```
$ vagrant package --base la_new_box
[la_new_box] Clearing any previously set forwarded ports...
[la_new_box] Creating temporary directory for export...
[la_new_box] Exporting VM...
[la_new_box] Compressing package to: /private/tmp/v/package.box
```

Setting Vagrantfile Defaults

```
Vagrant::Config.run do |config|
  config.ssh.username = ”linuxacademy"
End
```

```
$ vagrant package --base my_new_box --vagrantfile Vagrantfile
...
```