# Vagrant

# - The command line utility for managing the lifecycle of virtual machines

Advances in Data Sciences and Architecture

Under the guidance of Prof- Srikanth Krishnamurthy

**Authors**

Mohit Mittal <Https://Www.Linkedin.Com/In/Mohit-mittal-552a66108/>

Sneha Ravikumar <Https://Www.Linkedin.Com/In/Sneha-ravikumar-a8925b113/>

Taj Poovaiah Https://Www.Linkedin.Com/In/Taj-poovaiah-75ab2731/

# **Why VAGRANT?**

Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.

To achieve its magic, Vagrant stands on the shoulders of giants. Machines are provisioned on top of VirtualBox, VMware, AWS, or [any other provider](https://www.vagrantup.com/docs/providers/). Then, industry-standard [provisioning tools](https://www.vagrantup.com/docs/provisioning/) such as shell scripts, Chef, or Puppet, can be used to automatically install and configure software on the machine.

# **GETTING STARTED**

The Vagrant getting started guide will walk you through your first Vagrant project, and show off the basics of the major features Vagrant has to offer.

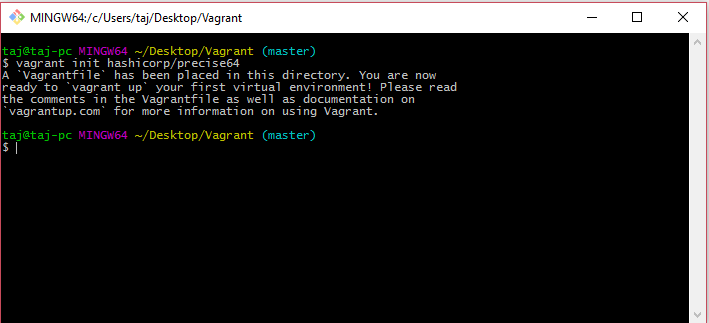
## **Up and running**

$ vagrant init hashicorp/precise64

$ vagrant up

After running the above two commands, you will have a fully running virtual machine in [VirtualBox](https://www.virtualbox.org/) running Ubuntu 12.04 LTS 64-bit. You can SSH into this machine with vagrant ssh, and when you are done playing around, you can terminate the virtual machine with vagrant destroy.

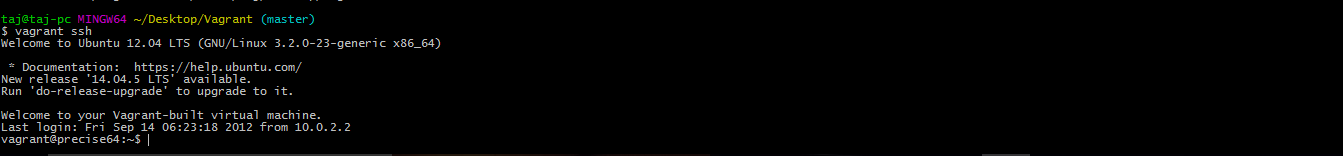
1. vagrant init hashicorp/precise64

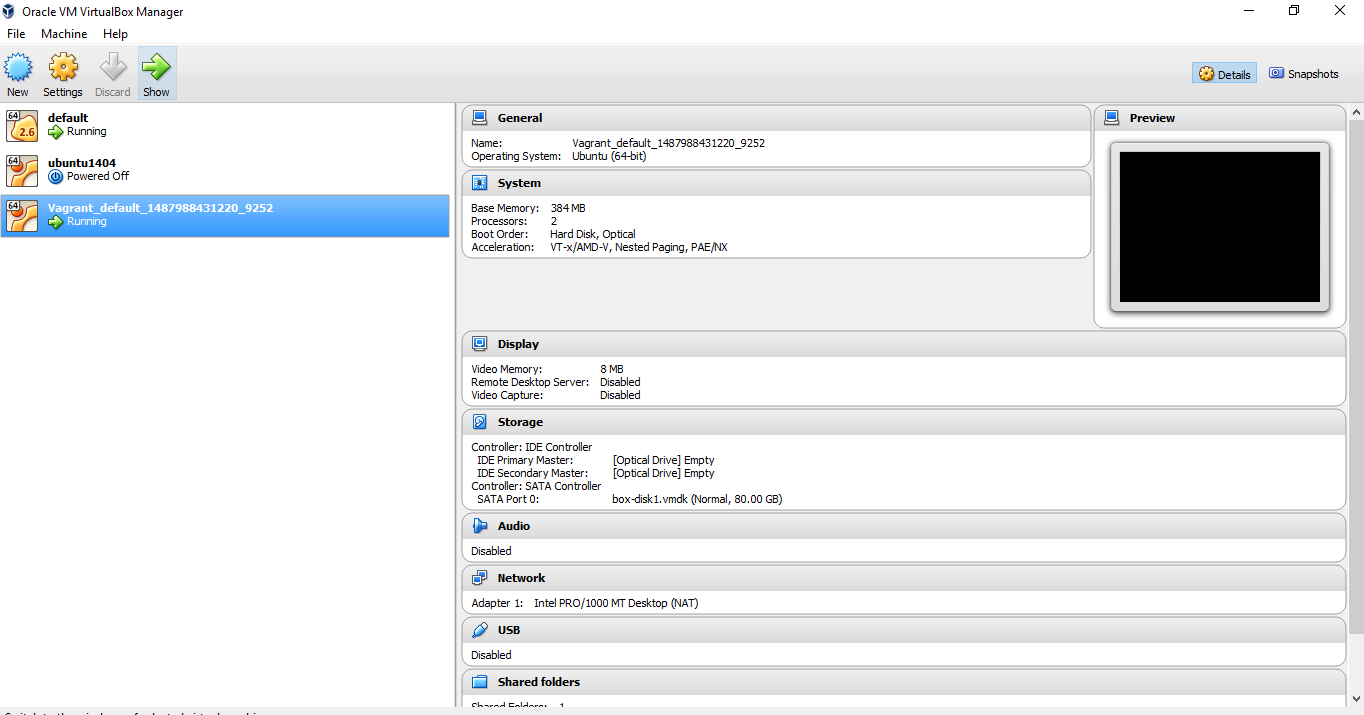


1. vagrant up­­­­­­­­



1. vagrant ssh





# **BOXES­**

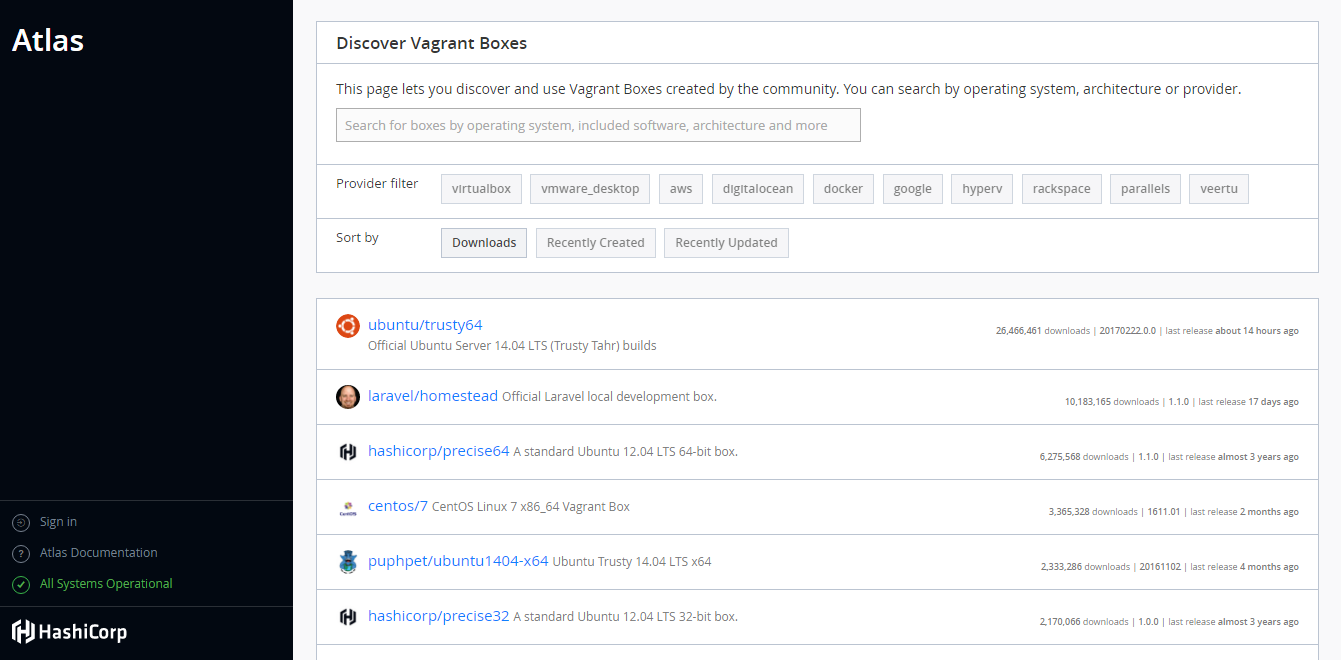
Instead of building a virtual machine from scratch, which would be a slow and tedious process, Vagrant uses a base image to quickly clone a virtual machine. These base images are known as "boxes" in Vagrant, and specifying the box to use for your Vagrant environment is always the first step after creating a new Vagrantfile.

## **INSTALLING A BOX**

If you ran the commands on the [getting started overview page](https://www.vagrantup.com/docs/getting-started/), then you've already installed a box before, and you do not need to run the commands below again. However, it is still worth reading this section to learn more about how boxes are managed.

Boxes are added to Vagrant with vagrant box add. This stores the box under a specific name so that multiple Vagrant environments can re-use it. If you have not added a box yet, you can do so now:

$ vagrant box add hashicorp/precise64



# **VAGRANTFILE**

The primary function of the Vagrantfile is to describe the type of machine required for a project, and how to configure and provision these machines. Vagrantfiles are called Vagrantfiles because the actual literal filename for the file is Vagrantfile (casing does not matter unless your file system is running in a strict case sensitive mode).

Vagrant is meant to run with one Vagrantfile per project, and the Vagrantfile is supposed to be committed to version control. This allows other developers involved in the project to check out the code, run vagrant up, and be on their way. Vagrantfiles are portable across every platform Vagrant supports.

The syntax of Vagrantfiles is [Ruby](http://www.ruby-lang.org/), but knowledge of the Ruby programming language is not necessary to make modifications to the Vagrantfile, since it is mostly simple variable assignment. In fact, Ruby is not even the most popular community Vagrant is used within, which should help show you that despite not having Ruby knowledge, people are very successful with Vagrant.

## **LOOKUP PATH**

When you run any vagrant command, Vagrant climbs up the directory tree looking for the first Vagrantfile it can find, starting first in the current directory. So if you run vagrant in /home/mitchellh/projects/foo, it will search the following paths in order for a Vagrantfile, until it finds one:

/home/mitchellh/projects/foo/Vagrantfile

/home/mitchellh/projects/Vagrantfile

/home/mitchellh/Vagrantfile

/home/Vagrantfile

/Vagrantfile

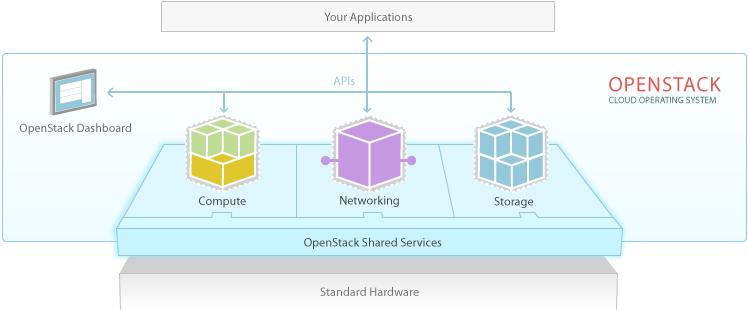
This feature lets you run vagrant from any directory in your project.

You can change the starting directory where Vagrant looks for a Vagrantfile by setting the VAGRANT\_CWD environmental variable to some other path.

# OpenStack

# - A cloud software stack designed to run on commodity hardware

### Inside OpenStack:



OpenStack is a complex beast containing multiple components. The core components are:

OpenStack Compute is the virtual machine provisioning and management module. It is the mighty tool that controls the whole works: networking, CPU, storage, memory, creating, controlling, and removing virtual machine instances, security, and access control through command line or Graphical web based dashboard.

OpenStack Glance, the OpenStack Image Service, manages virtual disk images. It can be used to stream virtual disk images, configure public and private images and control access to them, and of course create and destroy them.

OpenStack Object Store is a distributed storage system for managing all types of storage: archives, user data, virtual machine images, and the hardware they're stored on.

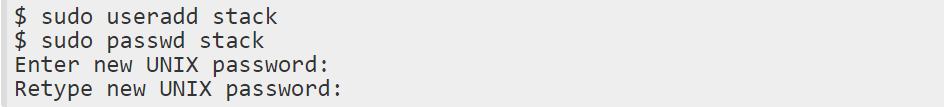
Features:

* Multiple layers of redundancy and automatic replication
* Recovery is automatic.

The Identity Service manages users and projects.

# Installations:

“Don't put OpenStack on your main PC because it needs a dedicated system.  If you elect to run OpenStack server in a virtual machine give it a minimum of 1.5GB RAM and 6GB storage. If you have a multicore system and can spare more than one core, do so because OpenStack, even in a simple testing setup, gets hungry.”

1. First create a user named **stack** to use for installing DevStack:

2.

Give stack full sudo privileges:

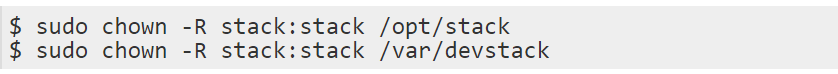


3.

Logout, and then log back in as stack. If you don't have git then install it:

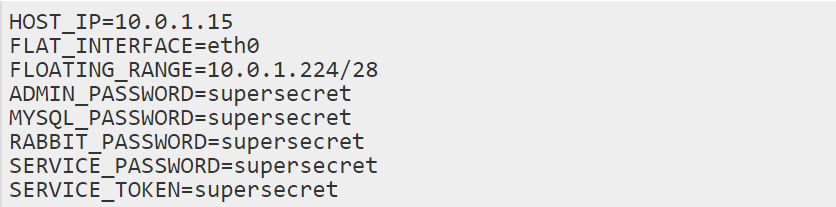
4.

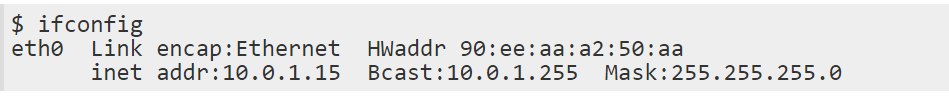
Pull OpenStack from Github. This copies it into the current directory, so I cd to /var and then run git:

* This puts everything in /var/devstack. cd to devstack, and take a few minutes to look in the various scripts and files
* Change ownership of /var/devstack and /opt/stack to stack

5. Change /var/www to www-data:www-data

# Configuration:

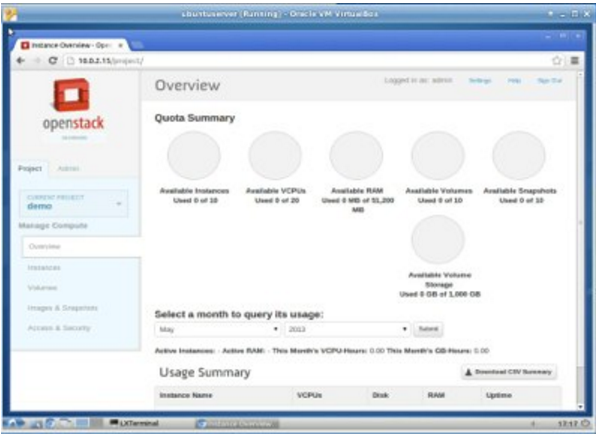
There is one more prerequisite, and that is to create /var/devstack/localrc. localrc always goes in your DevStack root, and it configures networking, passwords, logging, and several other items we're going to ignore for the time being. This is what mine looks like, just a minimal configuration:

OpenStack uses a lot of passwords, so for testing I make it easy on myself by recycling the same one. The HOST\_IP is the ethX inet addr of your OpenStack server, whether it's virtualized or not, like this example:

Do create a static IP address for your DevStack server, or you will suffer. Networking is rather involved for OpenStack, and we'll get into that more in the future;

FLAT\_INTERFACE is the server's Ethernet interface; if you have just one it's not necessary to include this line. You could have an internal and a public-facing interface, just like on non-cloud servers, and the FLAT\_INTERFACE corresponds to the internal interface.

FLOATING\_RANGE is a pool of addresses for any OpenStack servers that need to be available to the network. This must not overlap with the server's IP address, which is why my example is way out at the end of the address range.



Change to /var/devstack and run:



# Installation Complete

# Running Openstack:

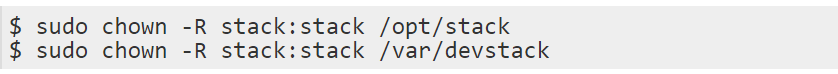
1.Web browser on your OpenStack server and point it to the IP address it told you

2.See login page

3.Go ahead and login as admin with whatever password you set in localrc

4.f you make a mess, the good DevStack people included a do-over script, clean.sh. This reverses stack.sh and leaves your git clone files in place, so run clean.sh and then stack.sh to re-do your installation.

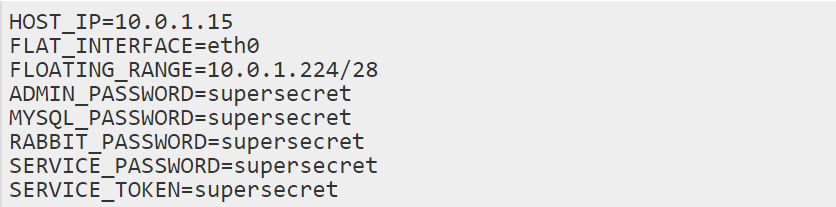
Pull OpenStack from Github. This copies it into the current directory, so I cd to /var and then run git:

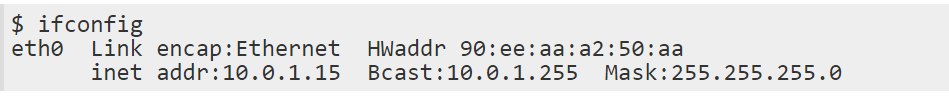
* This puts everything in /var/devstack. cd to devstack, and take a few minutes to look in the various scripts and files
* Change ownership of /var/devstack and /opt/stack to stack

5.

Change /var/www to www-data:www-data

# Configuration:

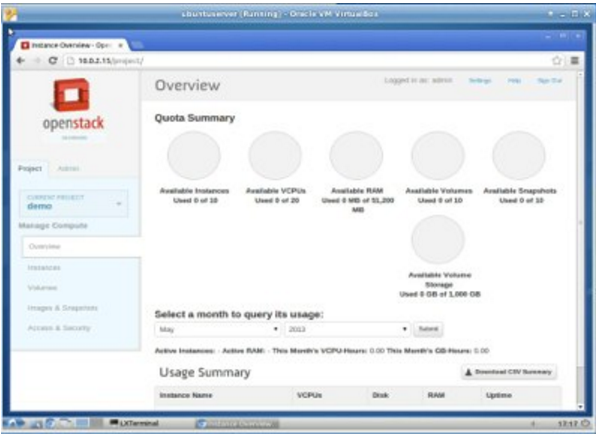
There is one more prerequisite, and that is to create /var/devstack/localrc. localrc always goes in your DevStack root, and it configures networking, passwords, logging, and several other items we're going to ignore for the time being. This is what mine looks like, just a minimal configuration:

OpenStack uses a lot of passwords, so for testing I make it easy on myself by recycling the same one. The HOST\_IP is the ethX inet addr of your OpenStack server, whether it's virtualized or not, like this example:

Do create a static IP address for your DevStack server, or you will suffer. Networking is rather involved for OpenStack, and we'll get into that more in the future;

FLAT\_INTERFACE is the server's Ethernet interface; if you have just one it's not necessary to include this line. You could have an internal and a public-facing interface, just like on non-cloud servers, and the FLAT\_INTERFACE corresponds to the internal interface.

FLOATING\_RANGE is a pool of addresses for any OpenStack servers that need to be available to the network. This must not overlap with the server's IP address, which is why my example is way out at the end of the address range.



Change to /var/devstack and run:



# Installation Complete

# Running Openstack:

1.Web browser on your OpenStack server and point it to the IP address it told you

2.See login page

3.Go ahead and login as admin with whatever password you set in localrc

4.f you make a mess, the good DevStack people included a do-over script, clean.sh. This reverses stack.sh and leaves your git clone files in place, so run clean.sh and then stack.sh to re-do your installation.

# **REFERENCE LINKS**

1. <https://www.vagrantup.com/downloads.html> Vagrant Download page
2. <https://atlas.hashicorp.com/boxes/search> Vagrant Boxes List
3. <https://www.vagrantup.com> Vagrant Download link and Get Started Link
4. <https://en.wikipedia.org/wiki/Vagrant_(software)> More information about vagrant
5. <http://trystack.org/> TryStack Installation
6. <https://www.linux.com/learn/intro-openstack-part-two-how-install-and-configure-openstack-server> OpenStack Installation
7. <https://www.drupal.org/node/2008794> Vagrant commands