

10 year anniversary

**Subscribe now**

X

Get the highlights in your
inbox every week.

LOG IN

SIGN UP

Ma

[Articles](#)[Resources](#)[Downloads](#)[About](#)

Subscribe

[Open Organization](#)[Privacy Statement](#)

Managing virtual environments with Vagrant

Learn to create and configure a virtual machine and run post-deployment configuration with Vagrant.

23 Apr 2018 | [Alex Juarez \(/users/mralexjuarez\)](#) | 151 | [1 comment](#)



Image by : Opensource.com

Vagrant is a tool that offers a simple and easy to use command-line client for managing virtual environments. I started using it because it made it easier for me to develop websites, test solutions, and learn new things.

Subscribe now

X

Get the highlights in your inbox every week.

[_\(https://www.vagrantup.com/intro/index.html\)](https://www.vagrantup.com/intro/index.html), environment setup time, increases production parity, 'machine' excuse a relic of the past."

and you can learn a bit more background in [an source resources article](https://www.vagrantup.com/docs/vagrant) [rces/vagrant](https://www.vagrantup.com/docs/vagrant)).

demonstrate how to use Vagrant to:

1. Create and configure a virtualBox virtual machine (VM)
2. Run post-deployment configuration shell scripts and applications

Sounds simple, and it is. Vagrant's power comes from having a consistent workflow for deploying and configuring machines regardless of platform or operating system.

We'll start by using VirtualBox as a **provider**, setting up an Ubuntu 16.04 **box**, and applying a few shell commands as the **provisioner**. I'll refer to the physical machine (e.g., a laptop or desktop) as the host machine and the Vagrant VM as the guest.

In this tutorial, we'll put together a [Vagrantfile](https://www.vagrantup.com/docs/vagrantfile/) [\(https://www.vagrantup.com/docs/vagrantfile/\)](https://www.vagrantup.com/docs/vagrantfile/) and offer periodic checkpoints to make sure our files look the same. We'll cover the following introductory and advanced topics:

Introductory topics:

- Installing Vagrant
- Choosing a Vagrant box
- Understanding the Vagrantfile

- Getting the VM running
- Using provisioners

Subscribe now

X

Get the highlights in your
inbox every week.

3

⌂

t together nicely once we are finished.

First, we'll navigate to [Vagrant's \(https://www.vagrantup.com/downloads.html\)](https://www.vagrantup.com/downloads.html) and [VirtualBox's \(https://www.virtualbox.org/wiki/Downloads\)](https://www.virtualbox.org/wiki/Downloads) download pages to install the latest versions of each.

We can enter the following commands to ensure the latest versions of the applications are installed and ready to use.

Vagrant:

```
# vagrant --version  
Vagrant 2.0.3
```

VirtualBox:

```
# VBoxManage --version  
5.2.8r121009
```

Choosing a Vagrant box

Picking a Vagrant box is similar to picking an image for a server. At the base level, we choose which operating system (OS) we want to use. Some boxes go further and will have additional software (such as the Puppet or Chef client) already

Subscribe now

X

Get the highlights in your inbox every week.

boxes is [Vagrant Cloud \(https://vagrantcloud.com/\)](https://vagrantcloud.com/); it boxes for multiple providers. In this tutorial, we'll be 04 LTS daily build.

'agrantfile

configuration file for an environment. It describes the rd to how to build and configure the VirtualBox VMs.

We need to create an empty project directory to work from, then initialize a Vagrant environment from that directory with this command:

```
# vagrant init ubuntu/xenial64
```

This only creates the Vagrantfile; it doesn't bring up the Vagrant box.

The Vagrantfile is well-documented with a lot of guidance on how to use it. We can generate a minimized Vagrantfile with the `--minimal` flag.

```
# vagrant init --minimal ubuntu/xenial64
```

The resulting file will look like this:

```
Vagrant.configure("2") do |config|  
  config.vm.box = "ubuntu/xenial64"  
end
```

We will talk more about the Vagrantfile later, but for now, let's get this box up and running.

Getting the VM running

Let's issue the following command from our project directory:

Subscribe now

X

Get the highlights in your
inbox every week.

`vagrant up` the first time because it downloads the
image faster on subsequent runs because it reuses the

image, we can `ssh` into our single machine by issuing the
`ssh` command from the project directory:

~

That's it! From here we should be able to log onto our VM and start working with it.

Using provisioners

Before we move on, let's review a bit. So far, we've picked an image and gotten the server running. For the most part, the server is unconfigured and doesn't have any of the software we might want.

Provisioners provide a way to use tools such as Ansible, Puppet, Chef, and even shell scripts to configure a server after deployment.

An example of using the shell provisioner can be found in a default Vagrantfile. In this example, we'll run the commands to update apt and install Apache2 to the server.

```
config.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y apache2
SHELL
```

If we want to use an Ansible playbook, the configuration section would look like this:

Subscribe now

Get the highlights in your
inbox every week.

```
X do |ansible|
  config.vm.provision "ansible", do |ansible|
    ansible.playbook = "playbook.yml"
  end
end
```

the provisioning part of the Vagrantfile by issuing `ansible-playbook` is great for testing out scripts or configuration. Running `ansible-playbook` will re-build the VM each time.

look like this:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y apache2
  SHELL
end
```

After adding the provisioning section, we need to run this provisioning subcommand:

```
# vagrant provision
```

Next, we'll continue to build on our Vagrantfile, touching on some more advanced topics to build a foundation for anyone who wants to dig in further.

Networking

In this section, we'll add an additional IP address on VirtualBox's `vboxnet0` network. This will allow us to access the machine via the `192.168.33.0/24` network.

Adding the following line to the Vagrantfile will configure the machine to have an additional IP on the `192.168.33.0/24` network. This line is also used as an example in the default Vagrantfile.

Subscribe now X
Get the highlights in your inbox every week. `work", ip: "192.168.33.10`

where our working Vagrantfile stands:

```
fig|
al64"
network", ip: "192.168.33.10"
, inline: <<-SHELL

apt-get install -y apache2
SHELL
end
```

Next, we need to reload our configuration to reconfigure our machine with this new interface and IP. This command will shut down the VM, reconfigure the Virtual Box VM with the new IP address, and bring the VM back up.

```
# vagrant reload
```

When it comes back up, our machine should have two IP addresses.

Syncing folders

Synced folders are what got me into using Vagrant. They allowed me to work on my host machine, using my tools, and at the same time have the files available to the web server or application. It made my workflow much easier.

By default, the project directory on the host machine is mounted to the guest machine as `/home/vagrant`. This worked for me in the beginning, but eventually, I wanted to customize where this directory was mounted.

In our example, we are defining that the HTML directory within our project directory should be mounted as `/var/www/html` with user/group ownership of `root`.

Subscribe now

Get the highlights in your
inbox every week.

X 1", "/var/www/html",

using a synced folder as a web server document root,
e, or you might run into an issue where it looks like

nfiguration is out of scope for this article, but here are
update.

`EnableSendFile Off`

In Nginx:

`sendfile off;`

Vagrantfile checkpoint

After adding our synced folder configuration, our Vagrantfile will look like this:

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.network "private_network", ip: "192.168.33.10"
  config.vm.synced_folder "./html", "/var/www/html",
    owner: "root", group: "root"
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y apache2
  SHELL
end
```

We need to reload our machine to make the new configuration active.


```
# vagrant reload
```

Subscribe now

Get the highlights in your inbox every week.

x machines

ject directory as an "environment," and one machine
. This last section extends our Vagrantfile to deploy

ed to enclose the definition of a single machine
st of the configuration is exactly the same.

· definition within a `define` block.

```
vagrant.configure(2) do |config|

  config.vm.define "web" do |web|
    web.vm.box = "web"
    web.vm.box = "ubuntu/xenial64"
    web.vm.network "private_network", ip: "192.168.33.10"
    web.vm.synced_folder "./html", "/var/www/html",
      owner: "root", group: "root"
    web.vm.provision "shell", inline: <<-SHELL
      apt-get update
      apt-get install -y apache2
    SHELL
  end

end
```

Notice in the `define` block, our variable is called "web" and it is carried through the block to reference each configuration method. We'll use the same name to access it later.

In this next example, we'll add a second machine called "db" to our configuration. Where we used "web" in our second block before, we'll use "db" to reference the second machine. We'll also update our IP address on the `private_network` so we can communicate between the machines.

```
Vagrant.configure("2") do |config|

  config.vm.define "web" do |web|

    Subscribe now
    Get the highlights in your
    inbox every week.

    X 164"
    etwork", ip: "192.168.33.10"
    tml", "/var/www/html",
    "root"
    inline: <<-SHELL

    he2

    |

    al64"
    network", ip: "192.168.33.20"
    db.vm.synced_folder "./html", "/var/www/html",
      owner: "root", group: "root"
    db.vm.provision "shell", inline: <<-SHELL
      apt-get update
      apt-get install -y apache2
    SHELL
  end

end
```

Completed Vagrantfile checkpoint

In our final Vagrantfile, we'll install the MySQL server, update the IP address, and remove the configuration for the synced folder from the second machine.

```
Vagrant.configure("2") do |config|

  config.vm.define "web" do |web|
    web.vm.box = "web"
    web.vm.box = "ubuntu/xenial64"
    web.vm.network "private_network", ip: "192.168.33.10"
    web.vm.synced_folder "./html", "/var/www/html",
      owner: "root", group: "root"
    web.vm.provision "shell", inline: <<-SHELL
      apt-get update
      apt-get install -y apache2
```

```
SHELL
end

config.vm.define "dh" do |dh|
  X
  al64"
  network", ip: "192.168.33.20"
  , inline: <<-SHELL
  >="noninteractive"

  sql-server
```

Subscribe now

Get the highlights in your
inbox every week.

ing works

Now we have a completed Vagrantfile. Let's introduce one more Vagrant command to make sure everything works.

Let's destroy our machine and build it brand new.

The following command will remove our previous Vagrant image but keep the box we downloaded earlier.

```
# vagrant destroy --force
```

Now we need to bring the environment back up.

```
# vagrant up
```

We can ssh into the machines using the `vagrant ssh` command:

```
# vagrant ssh web
```

or

vagrant ssh db

You should have a working Vagrantfile you can expand upon and serve as a base

Subscribe now

Get the highlights in your
inbox every week.

X powerful tool for testing, developing and learning
keep adding to it and exploring the options it offers.

/tags/linuxfest-northwest) Tools (/tags/tools)

author

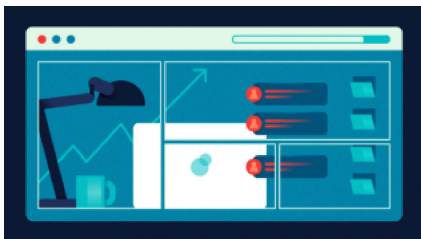
x Juarez is a Principal Engineer at Rackspace, touting 11 years
. Alex enjoys all things open-source as well as training and
and is incredibly qualified to do so as an RHCA/RHCI. Alex has
spoken at a number regional and nationwide open source conferences sharing
knowledge and encouraging people to learn more. When Alex isn't helping others
he's studying for the cicerone exam and trying to find places to eat and things to
see in San Antonio, TX.

[\(/users/mralexjuarez/\)](/users/mralexjuarez/)

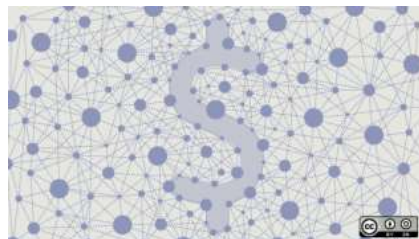
• [More about me \(/users/mralexjuarez/\)](/users/mralexjuarez/)

- [Learn how you can contribute \(/participate/\)](/participate/)

Recommended reading



[5 open source tools for email, note taking, and](#)



[3 open source tools for sticking to a budget](#)



[Tough lessons learned from measuring](#)