



EBook Gratis

APRENDIZAJE vagrant

Free unaffiliated eBook created from
Stack Overflow contributors.

#vagrant

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con vagrant.....	2
Observaciones.....	2
Enlaces útiles.....	2
Versiones.....	2
Examples.....	3
Instalación para Windows con soporte VirtualBox y SSH.....	3
Instalar virtualbox.....	3
Instalar cygwin.....	3
Añadir Cygwin a PATH.....	4
Instalar Vagrant sí mismo.....	4
Pruebas.....	4
Pasos adicionales.....	5
Proyecto LAMP.....	5
Cargar la imagen de Vagrant Box en Amazon AWS AMI.....	11
La forma más fácil de tener un Linux virtual en solo unos minutos (en 3 pasos).....	11
Paso 1.....	11
Paso 2.....	12
Paso 3.....	12
sincronizando todas las carpetas.....	12
sincronizar carpetas pero excluir algunas carpetas.....	12
Capítulo 2: Instantáneas.....	14
Examples.....	14
Toma una foto.....	14
Bash script para eliminar todas las instantáneas.....	14
Restaurar una instantánea.....	14
Listar instantáneas.....	14
Restaurar una instantánea sin aprovisionar el cuadro.....	14
Eliminar una instantánea.....	14

Capítulo 3: Proveedores	15
Examples	15
Establecer proveedor predeterminado con variable de entorno	15
Establecer proveedor predeterminado en Vagrantfile	15
Lanzar caja en Hyper-V	15
Lanzar caja en Docker	15
Lanzamiento de caja en VMWare Fusion	15
Lanzar caja en VMWare Workstation	15
Lanzar caja en virtualbox	15
Capítulo 4: Provisión	16
Sintaxis	16
Parámetros	16
Observaciones	16
Examples	16
Configuración mínima	16
Lanzar y aprovisionar la caja	16
Lanzar la caja sin aprovisionamiento	17
Aprovisionar una caja corriente	17
Aprovisionador de conchas	17
Ejecutar el script de shell desde el archivo (no usar inlining)	17
Capítulo 5: Re-aprovisionamiento ejecutando VM	18
Examples	18
Mando	18
Capítulo 6: SSH	19
Examples	19
SSH a la caja	19
SSH directamente a la caja	19
Creditos	20

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [vagrant](#)

It is an unofficial and free vagrant ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official vagrant.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con vagrant

Observaciones

Vagrant es una herramienta de software utilizada para crear y configurar entornos de desarrollo virtual. Funciona como contenedor sobre otro software de virtualización como VirtualBox o VMware. Vagrant en sí es una utilidad de línea de comandos.

Los cuadros de Vagrant se configuran utilizando la configuración especial de `Vagrantfile` escrita en Ruby, pero pueden incluir scripts de configuración adicionales escritos en bash, Chef o Puppet. Los desarrolladores en un equipo pueden descargar los archivos de configuración de Vagrant desde una fuente común y recrear el mismo entorno de desarrollo localmente.

Enlaces útiles

- [Lista de cajas vagabundas](#)
- [PuPHPet](#) : poderoso constructor de las cajas vagabundas.
- [Packer](#) : potente constructor de las cajas vagabundas.

Versiones

Versión	Descargar	Registro de cambios	Fecha de lanzamiento
1.8.5	Descargar	Registro de cambios	2016-07-18
1.8.4	Descargar	Registro de cambios	2016-06-13
1.8.3	Descargar	Registro de cambios	2016-06-10
1.8.1	Descargar	Registro de cambios	2015-12-24
1.8.0	Descargar	Registro de cambios	2015-12-21
1.7.4	Descargar	Registro de cambios	2015-07-17
1.7.3	Descargar	Registro de cambios	2015-07-10
1.7.2	Descargar	Registro de cambios	2015-01-06
1.7.1	Descargar	Registro de cambios	2014-12-12
1.7.0	Descargar	Registro de cambios	2014-12-10
1.6.5	Descargar	Registro de cambios	2014-09-04
1.6.4	Descargar	Registro de cambios	2014-09-02

Versión	Descargar	Registro de cambios	Fecha de lanzamiento
1.6.3	Descargar	Registro de cambios	2014-05-29
1.6.2	Descargar	Registro de cambios	2014-05-12
1.6.1	Descargar	Registro de cambios	2014-05-08
1.6.0	Descargar	Registro de cambios	2014-05-06

Examples

Instalación para Windows con soporte VirtualBox y SSH

Para usar Vagrant en la plataforma Windows, primero debe instalar un software de virtualización y una herramienta de línea de comandos ssh. Este ejemplo utilizará freeware VirtualBox y Cygwin.

Instalar virtualbox

Descargue la última versión de VirtualBox desde la [página de descarga oficial](#) y ejecute el archivo descargado. Tenga en cuenta que durante la instalación perderá temporalmente la conexión de

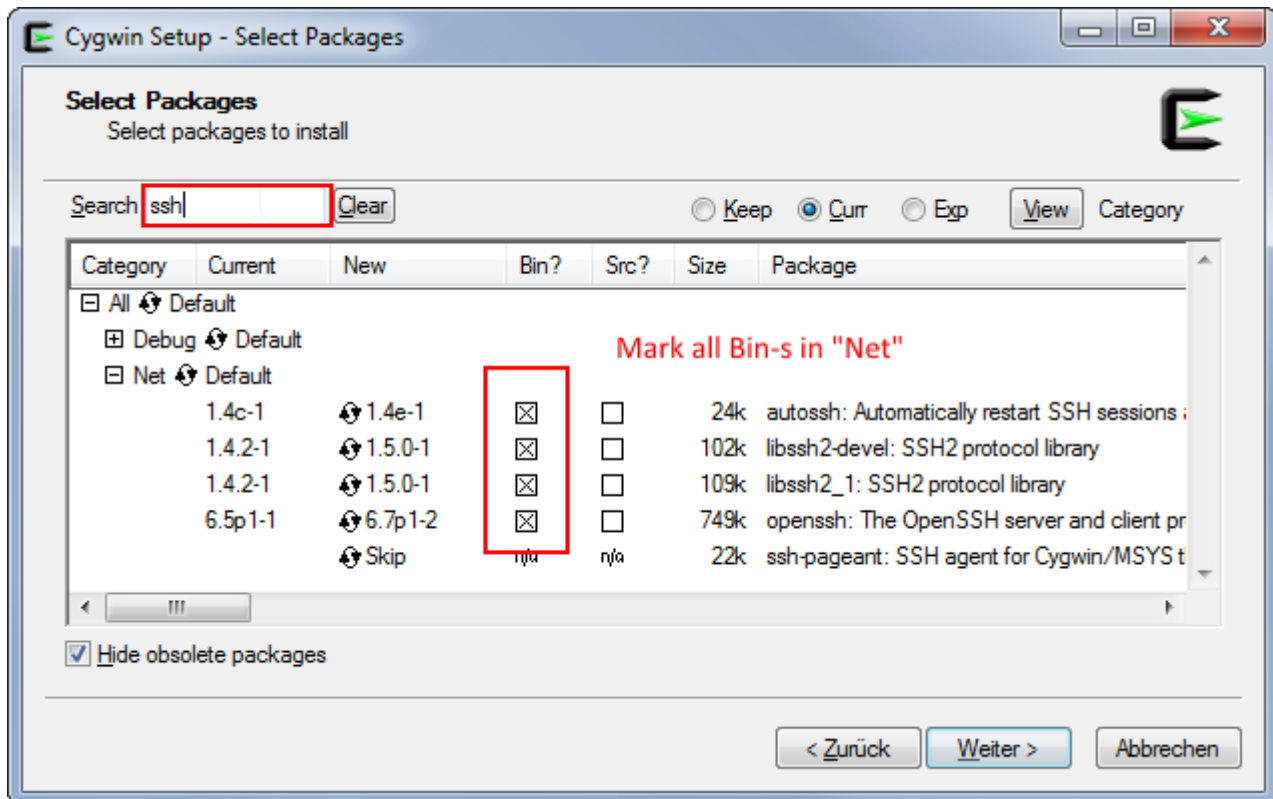


También debe dejar que Oracle instale controladores adicionales.

Instalar cygwin

Consíguelo en cygwin.com y ejecute la configuración hasta que [aparezca](#) la página "Seleccionar paquetes".

Solo necesitamos ssh bin-s desde aquí:



Añadir Cygwin a PATH

También debe agregar la carpeta `C:\cygwin64\bin` a la variable de entorno PATH de Windows. Para que puedas llamar al comando `ssh` desde cualquier lugar.

Instalar Vagrant sí mismo

Descargue Vagrant desde vagrantup.com y simplemente siga la guía de instalación para instalarlo. Tienes que reiniciar tu computadora después de esto.

Pruebas

La tecnología de virtualización VTx / VTd debe estar habilitada. (Puedes encontrar eso en el BIOS en las opciones de seguridad)

Para probar si todo está instalado correctamente, cree una carpeta vacía en algún lugar del disco duro, abra la línea de comandos y vaya a la carpeta que acaba de crear:

```
cd c:/path/to/your/folder
```

luego entra

```
vagrant init hashicorp/precise64
vagrant up
```

Esto creará e iniciará VirtualBox con Ubuntu 12.04 LTS de 64 bits Para iniciar la versión de 32 bits, use `vagrant init hashicorp/precise32` . Si necesita otra caja, puede encontrar más en el [sitio web de hashicorp](#) .

Además, el comando de `vagrant init Vagrantfile` creará un archivo de configuración de `Vagrantfile` en la carpeta actual. Ahora simplemente puede enviarlo a otra persona y cuando esa persona llame a `vagrant up` la misma máquina virtual en su PC.

Para probar ssh después de que estos dos comandos se ejecuten correctamente, ejecute este comando en la misma carpeta:

```
vagrant ssh
```

Si todo salió bien, terminará dentro de la máquina virtual que inició sesión como usuario `vagrant` .

Pasos adicionales

Puede detener la máquina virtual con

```
vagrant halt
```

o eliminarlo con

```
vagrant destroy
```

Más cajas junto con las instrucciones de cómo instalarlas se pueden encontrar en la página vagrantbox.es .

Proyecto LAMP

En este ejemplo, se crea un entorno de desarrollo de proyecto LAMP personalizado con Vagrant.

En primer lugar, deberá instalar [Virtual Box](#) y [Vagrant](#) .

Luego, cree una carpeta `vagrant` en su directorio de inicio, abra su terminal y cambie el directorio actual al nuevo directorio `vagrant` . Ahora, ejecuta `vagrant init` . Se `Vagrantfile` archivo `Vagrantfile` dentro y se verá así:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
```



```

# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure(2) do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://atlas.hashicorp.com/search.
  config.vm.box = "base"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  # config.vm.network "public_network"

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "../data", "/vagrant_data"

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
  # Example for VirtualBox:
  #
  # config.vm.provider "virtualbox" do |vb|
  #   # Display the VirtualBox GUI when booting the machine
  #   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
  #   vb.memory = "1024"
  # end
  #
  # View the documentation for the provider you are using for more
  # information on available options.

  # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies
  # such as FTP and Heroku are also available. See the documentation at
  # https://docs.vagrantup.com/v2/push/atlas.html for more information.
  # config.push.define "atlas" do |push|
  #   push.app = "YOUR_ATLAS_USERNAME/YOUR_APPLICATION_NAME"
  # end

  # Enable provisioning with a shell script. Additional provisioners such as
  # Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the

```

```
# documentation for more information about their specific syntax and use.
# config.vm.provision "shell", inline: <<-SHELL
#   sudo apt-get update
#   sudo apt-get install -y apache2
# SHELL
end
```

Agregue o descomente las siguientes líneas editando el archivo anterior:

```
config.vm.box = "hashicorp/precise64"
config.vm.network :forwarded_port, guest: 80, host: 8080
config.vm.network :private_network, ip: "192.168.33.10"
config.ssh.forward_agent = true
if Vagrant::Util::Platform.windows?
  config.vm.synced_folder "./", "/vagrant"
else
  config.vm.synced_folder "./", "/vagrant", type: "nfs"
end
# https://stefanwrobel.com/how-to-make-vagrant-performance-not-suck
config.vm.provider "virtualbox" do |v|
  host = RbConfig::CONFIG['host_os']

  # Give VM 1/4 system memory & access to all cpu cores on the host
  if host =~ /darwin/
    cpus = `sysctl -n hw.ncpu`.to_i
    # sysctl returns Bytes and we need to convert to MB
    mem = `sysctl -n hw.memsize`.to_i / 1024 / 1024 / 4
  elsif host =~ /linux/
    cpus = `nproc`.to_i
    # meminfo shows KB and we need to convert to MB
    mem = `grep 'MemTotal' /proc/meminfo | sed -e 's/MemTotal:/' -e 's/ kB//'`.to_i / 1024 /
4
  else # sorry Windows folks, I can't help you
    cpus = 2
    mem = 1024
  end

  v.customize ["modifyvm", :id, "--memory", mem]
  v.customize ["modifyvm", :id, "--cpus", cpus]
end
```

Edite su archivo de `hosts` para redireccionar su dominio deseado a la máquina virtual errante. Para Linux, suele ser `/etc/hosts`, para Windows `C:\Windows\System32\Drivers\etc\hosts`; y añadir esta línea:

```
192.168.33.10 vagrantServer.com
```

Por supuesto, puede reemplazar `vagrantServer.com` con cualquier nombre.

Ahora es el momento de crear el archivo `bootstrap.sh` (dentro del directorio `vagrant`). Este script se ejecutará cada vez que la VM se genere desde cero. Lee los comentarios cuidadosamente:

```
#!/usr/bin/env bash

# .ssh/authorized_keys (you will need to create a `.ssh` directory inside the `vagrant` one
and add a file named `authorized_keys` with the public keys of the users who have access to
```

```

the repository and may use this environment).
# You also will have to grant access to those public keys from the Github account, Bitbucket,
or whatever you're using.
cat /vagrant/config/authorized_keys >> /home/vagrant/.ssh/authorized_keys
if ! [ -d /root/.ssh ]; then
    mkdir /root/.ssh
fi
cp /vagrant/config/authorized_keys /root/.ssh
chmod 600 /root/.ssh/authorized_keys

# Install packages
apt-get update
apt-get install -y python-software-properties
add-apt-repository ppa:ondrej/php5 -y
apt-get update
apt-get install -y curl nano apache2 php5 php5-mysql php5-curl php5-gd php5-intl php5-mcrypt
git

# Apache2 run with user vagrant
APACHEUSR=`grep -c 'APACHE_RUN_USER=www-data' /etc/apache2/envvars`
APACHEGRP=`grep -c 'APACHE_RUN_GROUP=www-data' /etc/apache2/envvars`
if [ APACHEUSR ]; then
    sed -i 's/APACHE_RUN_USER=www-data/APACHE_RUN_USER=vagrant/' /etc/apache2/envvars
fi
if [ APACHEGRP ]; then
    sed -i 's/APACHE_RUN_GROUP=www-data/APACHE_RUN_GROUP=vagrant/' /etc/apache2/envvars
fi
sudo chown -R vagrant:www-data /var/lock/apache2

# Set user/password to mysql previously to installation
# Replace rootMysqlPassword with your desired MySQL root password
debconf-set-selections <<< 'mysql-server mysql-server/root_password password
rootMysqlPassword'
debconf-set-selections <<< 'mysql-server mysql-server/root_password_again password
rootMysqlPassword'

# Install mysql
apt-get update
apt-get install -y mysql-server mysql-client

# Link /vagrant (sync_folder) to apache directory (/var/www)
if ! [ -L /var/www ]; then
    rm -rf /var/www
    ln -fs /vagrant /var/www
fi

# Install composer
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer

# Composer example. Uncomment to install phpunit
#composer global require "phpunit/phpunit=3.7.*" --prefer-source

# Create mysql database (replace "vagrantDB" with any desired database name, and
"rootMysqlPassword" with the password set above)
mysql -u root -prootMysqlPassword -v -e "CREATE USER 'developer'@ '%' IDENTIFIED BY 'dev';
CREATE SCHEMA vagrantDB;
GRANT ALL ON vagrantDB TO 'developer'@ '%' ;"

# Uncomment to set default database fixtures based on `/vagrant/config/vagrantDBFixtures.sql`
file.

```

```

#mysql -u root -prootMysqlPassword -v vagrantDB < /vagrant/config/vagrantDBFixtures.sql

#####
##### THIS IS OPTIONAL #####
#####

# Install nodejs
curl -sL https://deb.nodesource.com/setup | sudo bash -
apt-get install -y nodejs

# Install npm packages
npm install -g npm
npm install -g bower
npm install -g forever
npm install -g gulp

# Set accepted license before install java
echo debconf shared/accepted-oracle-license-v1-1 select true | sudo debconf-set-selections
echo debconf shared/accepted-oracle-license-v1-1 seen true | sudo debconf-set-selections

# Install java7
apt-get install -y oracle-java7-installer oracle-java7-set-default

#####
##### END OPTIONAL #####
#####

# Generate ssh key without passphrase
ssh-keygen -f /root/.ssh/id_rsa -t rsa -N ""

# Add bitbucket and github to known hosts
touch /root/.ssh/known_hosts
ssh-keyscan -H bitbucket.org >> /root/.ssh/known_hosts
ssh-keyscan -H github.com >> /root/.ssh/known_hosts

# Source: https://gist.github.com/winhamwr/7122947
# Sleep until we can successfully SSH into Bitbucket.
# Uses doubling backoff while waiting
# with_backoff() adapted from http://stackoverflow.com/a/8351489
# Retries a command a configurable number of times with backoff.
#
# The retry count is given by ATTEMPTS (default 5), the initial backoff
# timeout is given by TIMEOUT in seconds (default 1.)
#
# Successive backoffs double the timeout.
#generatedKey=`cat /root/.ssh/id_rsa.pub`
echo -n "Generate a SSH key (https://help.github.com/articles/generating-ssh-keys/)
and add it to your Bitbucket account (Profile -> SSH keys) to continue. "

with_backoff() {
    local max_attempts=${ATTEMPTS-5}
    local timeout=${TIMEOUT-1}
    local attempt=0
    local exitCode=0

    while [ $attempt -lt $max_attempts ]
    do
        set +e
        "$@"
        exitCode=$?
        set -e

```

```

    if [ $exitCode -eq 0 ]
    then
        break
    fi

    echo "Failure! Retrying in $timeout.." 1>&2
    sleep $timeout
    attempt=$(( attempt + 1 ))
    timeout=$(( timeout * 2 ))
done

if [ $exitCode -ne 0 ]
then
    echo "You've failed me for the last time! ($@)" 1>&2
fi

return $exitCode
}

ATTEMPTS=${ATTEMPTS:-5}

export ATTEMPTS
with_backoff ssh -T git@bitbucket.org;

# Clone repositories (replace "yourProjectName" and "yourProjectRepository" with your project
data)
cd /var/www
rm -rf yourProjectName/
git clone yourProjectRepository

# Add server names to /etc/hosts (replace "vagrantServer.com" with the domain set above)
echo -e '\n127.0.0.1          vagrantServer.com' >> /etc/hosts

# Enable apache modes
a2enmod rewrite

# Copy sites-available file (you need to add the Apache configuration file for the desired
domain in `config/sites-available`. Replace "vagrantServer.conf" with the desired name)
cp /vagrant/config/sites-available/vagrantServer.conf /etc/apache2/sites-available/

# Remove html from document root
sed -i 's/\var/www/html/\var/www/g' /etc/apache2/sites-available/*
service apache2 restart

# Enable sites (replace "vagrantServer.conf" with the above file name)
a2ensite vagrantServer.conf

# Install ruby, compass and sass (Optional)
apt-get install -y rubygems
gem install compass
npm install -g sass

# Pull the repo
cd /var/www/yourProjectName
git pull --all

```

Una vez que haya revisado y guardado el archivo anterior, vaya a su terminal nuevamente, cambie el directorio actual al `vagrant` que ha creado antes y escriba `vagrant up` . Se creará la máquina virtual y se ejecutará el archivo bootstrap desde la máquina virtual, por lo que se copiarán

/ instalarán todas las cosas necesarias. Una vez que haya terminado, puede abrir su navegador e ir a `vagrantServer.com` (o el nombre que le haya dado) y debería ver el contenido servido desde la VM vagrant.

También será capaz de editar los archivos de proyecto a través del `vagrant/yourProjectName` directorio y todos los archivos dentro de `vagrant` en el directorio será compartida y sincronizados entre su anfitrión y el vagabundo VM.

Cargar la imagen de Vagrant Box en Amazon AWS AMI

Tiene un cuadro local errante que desea cargar en Amazon AWS. Primero, necesitas [crear](#) un archivo `.box` :

```
vagrant package --base my-virtual-machine
```

Este paso debería tomar un tiempo dependiendo del tamaño de su imagen. Luego, necesita obtener la imagen `.vmdk` del archivo `.box` :

```
gunzip -S .box package.box  
tar xf package
```

Después de este paso, debería tener 4 archivos nuevos: `package` , `box-disk1.vmdk` , `Vagrantfile` y `box.ovf` . Ahora, para subir a AWS. Suponiendo que tiene una [cuenta de AWS](#) , cree un [cubo S3](#) para almacenar la imagen en los servidores de Amazon. Necesitará [la CLI de EC2 de Amazon](#) para el siguiente paso (ya que no puedo hacer esto a través de la consola por lo que puedo decir):

```
ec2-import-instance box-disk1_1.vmdk -f VMDK -t t2.micro -a x86_64 -b <S3-bucket-name> -o  
$AWS_ACCESS_KEY -w $AWS_SECRET_KEY -p Linux
```

Este resultado de este comando debería tardar un tiempo: está cargando el archivo de imagen grande a S3, pero el comando en sí regresa más rápidamente. Puede verificar el progreso de la importación utilizando el `ec2-describe-conversion-tasks` .

Una vez que termine, verá una instancia de su caja ejecutándose en la consola de AWS. Sin embargo, es posible que no pueda acceder a él porque no tiene una dirección IP pública y / o no tiene un archivo `.pem` asociado. Por lo tanto, el siguiente paso es crear una AMI desde la instancia. Para crear una AMI, detenga la instancia (¡no termine!) Y haga clic derecho en la instancia y vaya a `Image -> Create Image` . Esto debería tomar un tiempo también. Puede consultar el progreso en la vista de AMI de la consola. Una vez que finalice, iniciar una instancia utilizando el IAM adjuntando una `.pem` archivo de clave a la misma y entonces se puede `ssh` y ya está bueno para ir.

La forma más fácil de tener un Linux virtual en solo unos minutos (en 3 pasos)

Paso 1.

En su máquina host (Windows / Linux / OSX), cree un `my_project` vacío `my_project` .

Paso 2.

Crea un archivo llamado `Vagrantfile` con esto:

```
Vagrant.configure("2") do |config|
  config.vm.box = "gbarbieru/xenial" #An Ubuntu 16.04 based image
  config.vm.hostname = "my_project"
  config.vm.network :private_network, ip: "172.16.123.10"
end
```

Paso 3.

Ejecute su máquina:

```
host$ vagrant up
host$ vagrant ssh
virtual$ cd /vagrant
```

¡Hecho!

Notas:

- Tal vez usted quiera asignar otra dirección IP.
- Si su host es Windows, tal vez quiera `ssh` de `putty` . Puede hacerlo `ssh-ing` al nombre de host `127.0.0.1` y al puerto `2222` . El nombre de usuario es `vagrant` y la contraseña es `vagrant` .

sincronizando todas las carpetas

Para sincronizar todas las carpetas en ambas direcciones, inserte esto en su `Vagrantfile`

```
config.vm.synced_folder "my-project1", "/home/vagrant/my-project1"
```

sincronizar carpetas pero excluir algunas carpetas

Para sincronizar todas las carpetas en ambas direcciones, inserte esto en su `Vagrantfile`:

```
config.vm.synced_folder "my-project1", "/home/vagrant/my-project1", type: "rsync",
:rsync__exclude => ['my-project1/mini_project2/target,my-project1/mini_project2/target,my-
project1/mini_project3/target']
```

Todas las carpetas de destino están excluidas de la sincronización.

La sincronización solo ocurre en `vagrant up` y `vagrant reload` .

Para sincronizar cada cambio de tu host a tu invitado tienes que usar:

```
vagrant rsync-auto
```

Lea Empezando con vagrant en línea: <https://riptutorial.com/es/vagrant/topic/1026/empezando-con-vagrant>

Capítulo 2: Instantáneas

Examples

Toma una foto

```
vagrant snapshot save mysnapshot
```

Bash script para eliminar todas las instantáneas

```
#!/bin/bash

NO_SNAPSHOTS="No snapshots have been taken yet"
SNAPSHOT_OUTPUT=$(vagrant snapshot list | grep "${NO_SNAPSHOTS}")

if [ -z "${SNAPSHOT_OUTPUT}" ]; then
    echo "Found some snapshots, going to remove them"
    for SNAPSHOT in $(vagrant snapshot list); do
        vagrant snapshot delete "${SNAPSHOT}"
    done
else
    echo "No snapshots found"
fi
```

Restaurar una instantánea

```
vagrant snapshot restore mysnapshot
```

Listar instantáneas

```
vagrant snapshot list
```

Restaurar una instantánea sin aprovisionar el cuadro

```
vagrant snapshot restore --no-provision mysnapshot
```

Eliminar una instantánea

```
vagrant snapshot delete mysnapshot
```

Lea Instantáneas en línea: <https://riptutorial.com/es/vagrant/topic/2483/instantaneas>

Capítulo 3: Proveedores

Examples

Establecer proveedor predeterminado con variable de entorno

Introducir la variable de entorno `VAGRANT_DEFAULT_PROVIDER`

```
export VAGRANT_DEFAULT_PROVIDER=vmware_fusion
```

Establecer proveedor predeterminado en Vagrantfile

```
Vagrant.configure("2") do |config|  
  # ... other config up here  
  
  config.vm.provider "vmware_fusion"  
end
```

Lanzar caja en Hyper-V

```
vagrant up --provider hyperv
```

Lanzar caja en Docker

```
vagrant up --provider docker
```

Lanzamiento de caja en VMWare Fusion

```
vagrant up --provider vmware_fusion
```

Lanzar caja en VMWare Workstation

```
vagrant up --provider vmware_workstation
```

Lanzar caja en virtualbox

```
vagrant up --provider virtualbox
```

VirtualBox es el proveedor predeterminado para una configuración de vainilla Vagrant.

Lea Proveedores en línea: <https://riptutorial.com/es/vagrant/topic/2482/proveedores>

Capítulo 4: Provisión

Sintaxis

- `config.vm.provision "shell", en línea: COMANDOS`
- `config.vm.provision "shell", ruta: "relativePath / script.sh"`

Parámetros

Parámetro	Detalles
Comandos	Los comandos de shell para ejecutar. Puede ser una cadena (por ejemplo, <code>"echo \"Hello, World!\""</code>) O una variable de cadena (por ejemplo, <code>\$setup</code>).

Observaciones

El aprovisionamiento se utiliza para configurar automáticamente una máquina virtual. Se realiza automáticamente cuando se crea una máquina virtual (usando `vagrant up`). También puede volver a ejecutarse más tarde usando una `vagrant provision`.

Examples

Configuración mínima

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "vagrant-playbook.yml"
  end
end
```

vagrant-playbook.yml

```
---
- hosts: default
  tasks:
    - name: Say hello
      debug:
        msg: 'Hello, World'
```

Lanzar y aprovisionar la caja.

```
vagrant up
```

Por defecto el cuadro será aprovisionado.

Lanzar la caja sin aprovisionamiento.

```
vagrant up --no-provision
```

Aprovisionar una caja corriente

```
vagrant provision
```

Aprovisionador de conchas

El aprovisionador de shell ejecuta un script de shell al aprovisionar.

```
$setup = <<SETUP
# You can write your shell script between here ...
sudo echo "Hello, World!" > /etc/motd.tail
# ... and here.
SETUP

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.provision "shell", inline: $setup
end
```

Ejecutar el script de shell desde el archivo (no usar inlining)

```
# provision/bootstrap-controller.sh : path and shell filename from vagrantfile location
config.vm.define "configcontroller" do |controller|
  ...
  controller.vm.provision :shell do |shell|
    shell.path = "provision/bootstrap-controller.sh"
  end
  ...
end
```

Lea Provisión en línea: <https://riptutorial.com/es/vagrant/topic/3091/provision>

Capítulo 5: Re-aprovisionamiento ejecutando VM

Examples

Mando

```
vagrant reload --provision
```

Lea Re-aprovisionamiento ejecutando VM en línea:

<https://riptutorial.com/es/vagrant/topic/6056/re-aprovisionamiento-ejecutando-vm>

Capítulo 6: SSH

Examples

SSH a la caja

```
vagrant ssh
```

SSH directamente a la caja

```
vagrant ssh-config >> ~/.ssh/config  
ssh default
```

Lea SSH en línea: <https://riptutorial.com/es/vagrant/topic/6609/ssh>

Creditos

S. No	Capítulos	Contributors
1	Empezando con vagrant	Community , Daniel Käfer , dragon788 , ha_1694 , Joerg , Manolo , mrtuovinen , Robin , Ruslan Bes , theoretisch , Xavi Montero
2	Instantáneas	mrtuovinen
3	Proveedores	Ates Goral , mrtuovinen
4	Provisión	dragon788 , mrtuovinen , Robin , snonov
5	Re-aprovisionamiento ejecutando VM	Lernkurve
6	SSH	Daniel Käfer , mrtuovinen