



Complementary IT Tools for Researcher: Vagrant

Sébastien Varrette, PhD

UL HPC Management Team,
Parallel Computing and Optimization Group (PCOG),
University of Luxembourg (UL), Luxembourg



Summary

1 Introduction

2 Installation

3 Usage



What is Vagrant

<http://vagrantup.com/>



VMWARE INTEGRATION

DOWNLOADS

DOCUMENTATION

BLOG

ABOUT

**Development
environments
made easy.**

Create and configure lightweight, reproducible,
and portable development environments.

DOWNLOAD

GET STARTED



What is Vagrant ?

Create and configure **lightweight**, **reproducible**, and **portable** development environments

- Command line tool
- Automates VM creation with
 - ↳ VirtualBox
 - ↳ VMWare etc.
- Integrates well with configuration management tools
 - ↳ Shell
 - ↳ Puppet etc.

Runs on Linux, Windows, MacOS



Why use Vagrant?

- Create new VMs quickly and easily: only one command!

```
$> vagrant up
```

- Keep the number of VMs under control
 - ↔ All configuration in `VagrantFile`
- Reproducibility
 - ↔ Identical environment in development and production
- Portability
 - ↔ **avoid** sharing 4 GB VM disks images
 - ↔ [Vagrant Cloud](#) to share your images
- Collaboration made easy:
 - \$> `git clone ...`
 - \$> `vagrant up`



Summary

1 Introduction

2 Installation

3 Usage



Installation Notes: Windows / Linux

- Install Oracle Virtualbox
- Go on the [Download Page](#)
 - ↪ select the appropriate OS, in 64 bits versions
- **Notes for windows users:**
 - ↪ you will also need both [PuTTY](#) and [PuTTYGen](#)
 - ↪ Vagrant boxes are located in `%userprofile%/.vagrant.d/boxes`
 - ↪ To configure the appropriate Putty profile:
 - ✓ run `vagrant ssh-config` to collect IP and port (after `vagrant up`)
 - ✓ load `%userprofile%/.vagrant.d/insecure_public_key`
 - ✓ Use `Save Public Key` to convert the OpenSSH key to PPK format
 - ✓ Create the Putty profile accordingly (username: `vagrant`)

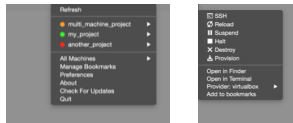


Installation Notes: Mac OS

- Best done using Homebrew and Cask

- ↳ install Homebrew
- ↳ install Homebrew-cask

```
$> brew install caskroom/cask/brew-cask
$> brew cask install virtualbox # install virtualbox
$> brew cask install vagrant
$> brew cask install vagrant-manager # see http://vagrantmanager.com/
```





Summary

1 Introduction

2 Installation

3 Usage



Minimal default setup

```
$> vagrant init <user>/<name> # setup a vagrant cloud image
```

- A Vagrantfile is configured



Minimal default setup

```
$> vagrant init <user>/<name> # setup a vagrant cloud image
```

- A Vagrantfile is configured

```
$> vagrant up # boot the box(es) set in the Vagrantfile
```

- The base box is downloaded and stored locally
 - ↳ in `~/.vagrant.d/boxes/`
- A new VM is created and configured with the base box as template
- The VM is booted and (eventually) provisioned



Minimal default setup

```
$> vagrant init <user>/<name> # setup a vagrant cloud image
```

- A Vagrantfile is configured

```
$> vagrant up # boot the box(es) set in the Vagrantfile
```

- The base box is downloaded and stored locally
↳ in `~/.vagrant.d/boxes/`
- A new VM is created and configured with the base box as template
- The VM is booted and (eventually) provisioned

```
$> vagrant ssh # connect inside it
```



Find a vagrant box

- Vagrant Cloud
- VagrantBox.es

<https://vagrantcloud.com/>

<http://www.vagrantbox.es/>



Find a vagrant box

- Vagrant Cloud
- VagrantBox.es

<https://vagrantcloud.com/>

<http://www.vagrantbox.es/>

Your Turn!

```
$> vagrant init hashicorp/precise32
$> vagrant up
$> vagrant ssh
```

- **Note:** once within the box:
 - ↪ `/vagrant` is the root directory hosting the Vagrantfile



Configuring Vagrant

- Minimal Vagrantfile (Ruby syntax)

```
VAGRANTFILE_API_VERSION = '2'  
  
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|  
  config.vm.box = 'hashicorp/precise32'  
end
```

- Configure Multiple VMs
↳ See [ULHPC/puppet-sysadmins](#)



Using another box

```
# From the command line (Vagrant cloud)
```

```
$> vagrant init alphainternational/centos-6.5-x64
```

```
# From the command line ("old", pre-1.5 style):
```

```
$ vagrant box add --name centos65 \
```

```
  http://packages.vstone.eu/vagrant-boxes/centos-6.x-64bit-latest.box
```

```
$ vagrant init centos65
```




Using another box

```
# From the command line (Vagrant cloud)
```

```
$> vagrant init alphainternational/centos-6.5-x64
```

```
# From the command line ("old", pre-1.5 style):
```

```
$ vagrant box add --name centos65 \  
  http://packages.vstone.eu/vagrant-boxes/centos-6.x-64bit-latest.box
```

```
$ vagrant init centos65
```

- Apply the changes

```
$> vagrant { destroy | halt } # destroy / halt
```

```
$> vagrant destroy
```

```
$> vagrant up
```

```
$> vagrant ssh
```



Generate your own box

- You might rely on `Falkor/vagrant-vm`s
 - ↪ use it at your own risks
 - ↪ based on `packer` and `veewee`

```
$> git clone https://github.com/Falkor/vagrant-vm.git
$> cd vagrant-vm
$> gem install bundler
$> bundle install
$> rake setup

# initiate a template for a given Operating System:
$> rake packer:{Debian,CentOS,openSUSE,scientificlinux,ubuntu}:init
# Build a Vagrant box
$> rake packer:{Debian,CentOS,openSUSE,scientificlinux,ubuntu}:build
# If things goes fine:
$> vagrant box add packer/<os>-<version>-<arch>/<os>-<version>-<arch>.box
```



Customize your box

- **Obj:** customize / specialize the configuration of a **running** box
- This can be done in two ways:
 - ① use **provisionning** within the Vagrantfile
 - ② re-package the box via `vagrant package`

```
# locate the name of the running VM
$> VBoxManage list runningvms
"vagrant-vms_default_1431034026308_70455" {...}
puppet-sysadmins_debian-7_1433278488158_28667" {...}

# Create the box
$> vagrant package \
    --base vagrant-vms_default_1431034026308_70455 \
    --output packer/<os>-<version>-<arch>/<os>-<version>-<arch>.box
```



Thank you for your attention...

Questions?

Sébastien Varrette, PhD

mail: sebastien.varrette@uni.lu

Office E-007

Campus Kirchberg

6, rue Coudenhove-Kalergi

L-1359 Luxembourg

UL HPC Management Team

mail: hpc-sysadmins@uni.lu



1 Introduction

2 Installation

3 Usage