



Cheet Sheet v0.1 (tested with vagrant v1.6.5)

Boxes

List all boxes:

% vagrant box list

Add new box (local *.box or name/url):

% vagrant box add <name, url, or path>

Remove box:

% vagrant box remove <name>Repackage existing box for distribution:
It will create *.box file in current directory**% vagrant box repackage <name> <provider>
<version>**

Finding more boxes:

<https://atlas.hashicorp.com/boxes/search>

Projects/Control

*Each project has a configuration file named 'Vagrantfile' and it is using ruby programming language.**By default project directory is shared inside virtual machine under /vagrant.*

Create a vagrant project in current directory:

% vagrant init [name [url]]**% vagrant init precise64**<http://files.vagrantup.com/precise64.box>

Start the machine:

% vagrant up [name]

Connect to the virtual machine:

% vagrant ssh [name]

Show the machine status:

% vagrant status [name]

Suspend the virtual machine:

To resume the VM, simply run `vagrant up` or `vagrant resume`

% vagrant suspend [name]

Shutdown the VM:

It will try graceful shutdown first and if it fails it just powers it off.

Append `--force` to force power off.

% vagrant halt

Destroy the machine:

% vagrant destroy [name]

VM Configuration

Box name:

config.vm.box = "precise64"

Box url (optional):

**config.vm.box_url =
"http://files.vagrantup.com/precise64.box"**

Shared folder location:

`/sfolder` – path inside virtual machine
`.` – path on host machine

To set the owner or group append:

owner: "root", group: "root"

To disable share:

disabled: true*Multiple shared folders can be defined.***config.vm.synced_folder ".", "/sfolder"**

Port forwarding:

By default it listens on all host interfaces, to change that, append:

ip:"127.0.0.1"

**config.vm.network "forwarded_port", guest: 80,
host: 8080**

Reload vm configuration:

% vagrant reload [vm-name]

Provisioning

Run shell script during vm creation:

Simply add configuration option to vm config.

No need to use `sudo` inside the shell script since it runs as root by default.

Example script:

#!/usr/bin/env bash

apt-get install -y apache2 >/dev/null 2>&1

**config.vm.provision "shell", path:
"provision.sh"**

Use puppet provisioning:

Manifests should be under *manifests* folder. Vagrant will run the *default.pp* manifest in that folder.

Sample manifest (manifests/default.pp):

exec { "apt-get update":
 command => "/usr/bin/apt-get update",
}package { "apache2":
 require => Exec["apt-get update"],
}**config.vm.provision "puppet"**

Chef provisioning:

Vagrant will run chef script located in:

./cookbooks/test/recipes/default.rb

**config.vm.provision "chef_solo", run_list:
["test"]**

Multiple provisioners can be used at the same time.

Inline provisioning:

**config.vm.provision "shell", inline: "apt-get
install -y nmap"**

Networking (advanced)

Forward udp port:

**config.vm.network "forwarded_port", guest: 80,
host: 8080, protocol: "udp"**

Assign different port if no avail:

*To configure range: config.vm.usable_port_range =
(2200..2250)***config.vm.network "forwarded_port", guest: 80,
host: 8080, auto_correct: true**

Configure host-only networking:

Machine can be reached from your host:

% ping 192.168.33.10PING 192.168.33.10 (192.168.33.10): 56 data bytes
64 bytes from 192.168.33.10: icmp_seq=0 ttl=64 time=0.547 ms
64 bytes from 192.168.33.10: icmp_seq=1 ttl=64 time=0.381 ms

Can still use `vagrant ssh` to ssh into vm.

**config.vm.network "private_network", ip:
"192.168.33.10"**

Configure bridged networking:

Vagrant will ask upon initialization to which device to bridge. This can also be specified by adding bridge option:

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

```
config.vm.network "public_network"
```

Multiple network adapters (host-only or bridge) can be specified. Vagrant by default creates one NAT interface, which is used for port forwarding.

Multiple VMs:

Sample configuration for two machines:

Set *autostart* option in order to prevent machine from starting up on ``vagrant up`` command.

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: "echo Hello"

  config.vm.define "web" do |web|
    web.vm.box = "hashicorp/precise32"
  end

  config.vm.define "db", autostart: false do |db|
    db.vm.box = "hashicorp/precise32"
  end
end
```

To connect to 'db' machine:

```
% vagrant ssh db
```

Start all machines:

```
% vagrant up
```

Start single machine:

```
% vagrant up web
```

Configure host-only networking for both machines:

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: "echo Hello"

  config.vm.define "web" do |web|
    web.vm.box = "hashicorp/precise32"
    web.vm.network "private_network", ip:
"192.168.33.10"
  end

  config.vm.define "db", autostart: false do |db|
    db.vm.box = "hashicorp/precise32"
    db.vm.network "private_network", ip:
"192.168.33.12"
  end
end
```

Vagrant Settings/Defaults:

Default location for vagrant boxes and configuration:

It can be changed by setting \$VAGRANT_HOME environment variable.

```
~/.vagrant.d
```

To change the name of vagrant project configuration file set \$VAGRANT_VAGRANTFILE variable (default: Vagrantfile)