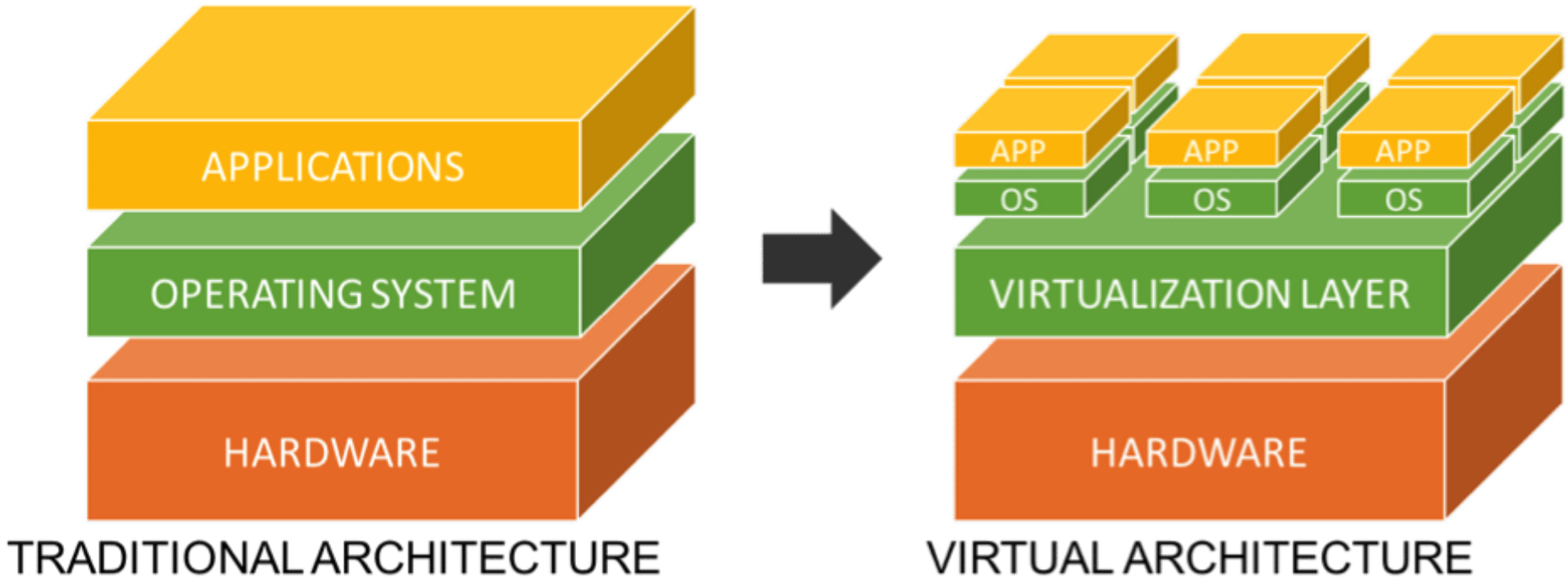
Vagrant

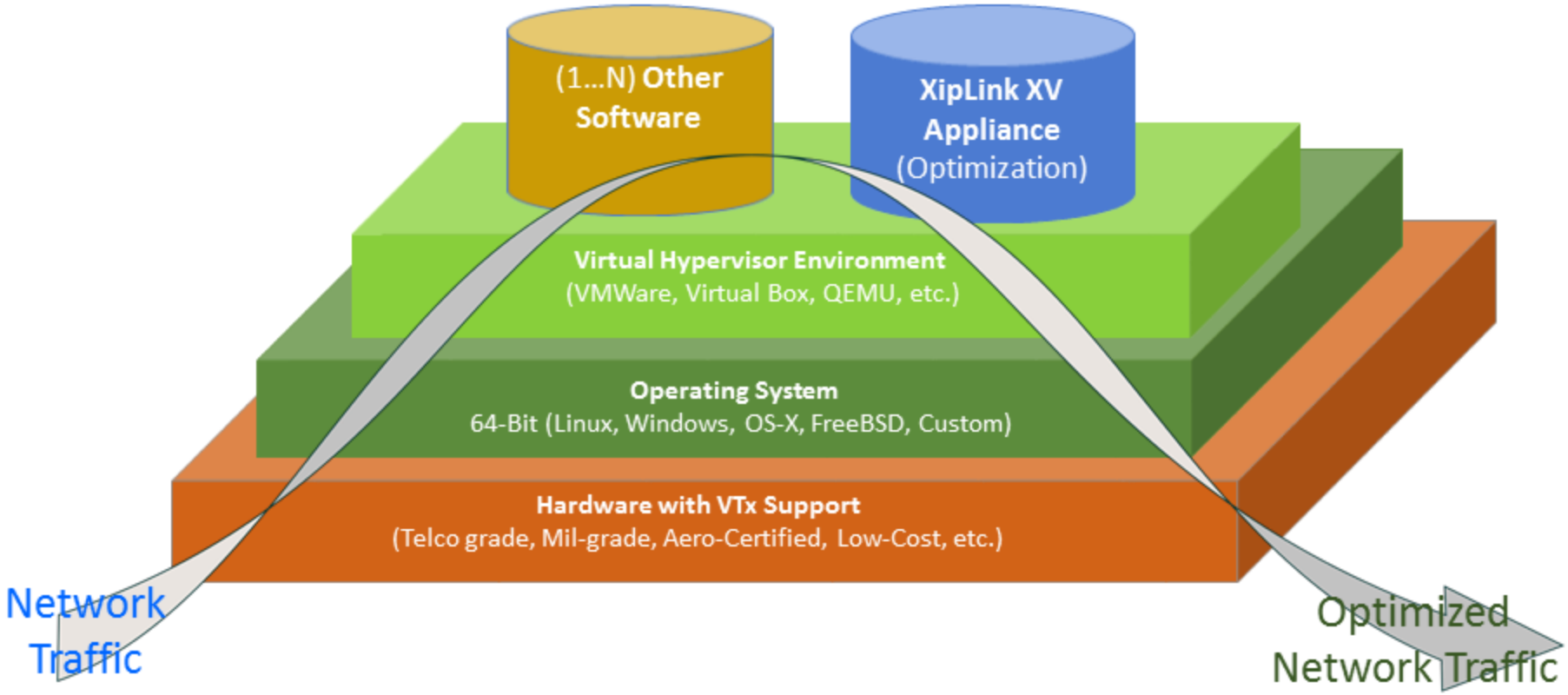
AGENDA

- Vagrant introduction
- Getting base boxes
- Configuring boxes
- Provisioning
  - Shell

Virtualization

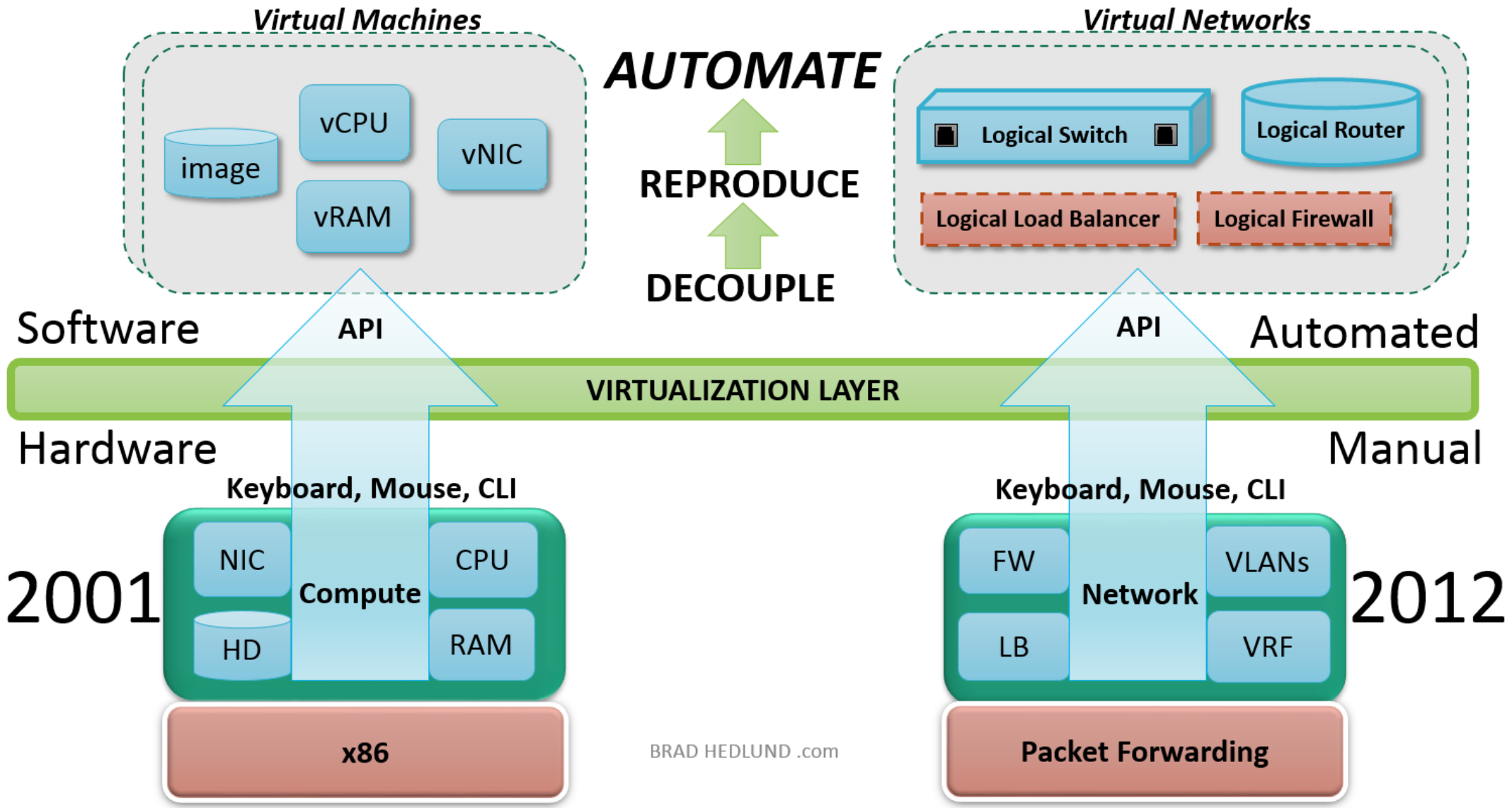


Network trafic



Server Virtualization

Network Virtualization



VAGRANT INTRODUCTION

WHAT IS VAGRANT?

http://www.vagrantup.com/

- Written by Mitchell Hashimoto
- Command line tool
- Automates VM creation with
  - VirtualBox
  - VMWare
  - Hyper-V
- Integrates well with configuration management tools
  - Shell
  - Ansible
  - Chef
  - Puppet
- 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
- Reproducability
- Identical environment in development and production
- Portability
  - No more 4GB .ova files
  - git clone and vagrant up

ASSUMPTIONS

- Git
- Vagrant 2.0.2
- VirtualBox 5.2.8 or newer
  - default Host-only network (192.168.56.0/24)

GETTING UP AND RUNNING

MINIMAL DEFAULT SETUP:

```
$ vagrant init centos/7
$ vagrant up
$ vagrant ssh
```

WHAT HAPPENS UNDER THE HOOD?

```
$ vagrant init centos/7
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

A *Vagrantfile* is created (that's all!)

WHAT HAPPENS UNDER THE HOOD?

```
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/7' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'centos/7'
    default: URL: https://atlas.hashicorp.com/centos/7
==> default: Adding box 'centos/7' (v1505.01) for provider: virtualbox
    default: Downloading:
https://atlas.hashicorp.com/centos/boxes/7/versions/1505.01/providers/virtualbox.box ==> default:
Box download is resuming from prior download progress
==> default: Successfully added box 'centos/7' (v1505.01) for 'virtualbox'!
==> default: Importing base box 'centos/7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'centos/7' is up to date...
==> default: Setting the name of the VM: test_default_1441636487571_53914
==> default: Fixed port collision for 22 => 2222. Now on port 2200.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
```

WHAT HAPPENS UNDER THE HOOD?

```
==> default: Forwarding ports... default: 22 => 2200 (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2200
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection timeout. Retrying... default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security. default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready! ==> default: Checking for guest additions in VM...
    default: No guest additions were detected on the base box for this VM! Guest
    default: additions are required for forwarded ports, shared folders, host only
    default: networking, and more. If SSH fails on this machine, please install
    default: the guest additions and repackage the box to continue.
    default:
    default: This is not an error message; everything may continue to work properly,
    default: in which case you may ignore this message.
==> default: Installing rsync to the VM...
==> default: Rsyncing folder: /home/bert/Downloads/test/ => /home/vagrant/sync
```

WHAT HAPPENS UNDER THE HOOD?

```
$ vagrant up
```

- 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
- The box is *provisioned*
  - only the first time, must be done manually afterwards

DONE!

- You now have a working VM, ready for use:

```
$ vagrant ssh [vagrant@localhost ~]
$ cat /etc/redhat-release
CentOS Linux release 7.1.1503 (Core)
[vagrant@localhost ~]$
```

CONFIGURING VAGRANT BOXES

VAGRANTFILE

- Vagrantfile = Ruby
- Minimal Vagrantfile:

```
VAGRANTFILE_API_VERSION = '2'
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
config.vm.box = 'centos/7'
end
```

FINDING BASE BOXES

- Hosted by Hashicorp: https://atlas.hashicorp.com/
- 3rd party repository: http://vagrantbox.es/

USING ANOTHER BASE BOX

- From the command line (Published on Atlas):

```
$ vagrant box add centos/7
$ vagrant init centos/7
```

From the command line (Box not on Atlas):

```
$ vagrant box add --name centos71-nocm \
 https://tinfbo2.hogent.be/pub/vm/centos71-nocm-1.0.16.box
$ vagrant init centos71-nocm
```

In your Vagrantfile (only applies to "old" style):

```
VAGRANTFILE_API_VERSION = '2'
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
config.vm.box = 'centos71-nocm’
config.vm.box_url = 'https://tinfbo2.hogent.be/pub/vm/centos71-
nocm-1.0.16.box’
end
```

APPLYING THE CHANGE

```
$ vagrant destroy
default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Forcing shutdown of VM...
==> default: Destroying VM and associated drives...
$ vagrant up
[...]
$ vagrant ssh
```

CONFIGURING THE VM

```
1. VAGRANTFILE_API_VERSION = '2'
2.
3. HOST_NAME = 'box001'
4.
5. Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
6.
7.    config.vm.hostname = HOST_NAME
8.    config.vm.box = 'centos/7'
9.    config.vm.network :private_network,
10.       ip: '192.168.56.65',
11.       netmask: '255.255.255.0'
12.
13.   config.vm.provider :virtualbox do |vb|
14.       vb.name = HOST_NAME
15.       vb.customize ['modifyvm', :id, '--memory', 256]
16.   end
17. end
```

CONFIGURING THE VM

- For more info,
- see the docs at https://docs.vagrantup.com/
- or the default **Vagrantfile**

APPLYING CHANGES

- When you change the Vagrantfile, do:

```
$ vagrant reload
```

Or, if the change is profound:

```
$ vagrant destroy -f
$ vagrant up
```

SETUP WITH MULTIPLE VMS

- Vagrantfile:

```
config.vm.define HOST_NAME do |node|
   node.vm.hostname = HOST_NAME
   [...]
end
```

Specify `HOST_NAME` after `vagrant` command:

```
$ vagrant status # Status of *all* boxes
$ vagrant up box001 # Boot box001
$ vagrant up # Boot *all* defined boxes
$ vagrant ssh box001
```

SETUP WITH MULTIPLE VMS: EXAMPLE (1/2)

```
1. VAGRANTFILE_API_VERSION = '2'
2.
3. Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
4.
5.     config.vm.define 'box001' do |node|
6.         node.vm.hostname = 'box001'
7.         node.vm.box = 'centos/7'
8.         node.vm.network :private_network,
9.             ip: '192.168.56.65',
10.            netmask: '255.255.255.0'
11.
12.        node.vm.provider :virtualbox do |vb|
13.            vb.name = 'box001‘
14.        end
15.    end
16.    config.vm.define 'box002' do |node|
17.        node.vm.hostname = 'box002'
18.        node.vm.box = 'centos/7'
19.        node.vm.network :private_network,
20.            ip: '192.168.56.66',
21.            netmask: '255.255.255.0'
22.
23.        node.vm.provider :virtualbox do |vb|
24.            vb.name = 'box002'
25.        end
26.    end
27.end
```

SETUP WITH MULTIPLE VMS: EXAMPLE (2/2)

```
1. VAGRANTFILE_API_VERSION = '2'

2. hosts = [ { name: 'box001', ip: '192.168.56.65' },
3.           { name: 'box002', ip: '192.168.56.66' }]
4.
5. Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
6.   hosts.each do |host|
7.     config.vm.define host[:name] do |node|
8.       node.vm.hostname = host[:name]
9.       node.vm.box = 'centos/7'
10.      node.vm.network :private_network,
11.        ip: host[:ip],
12.        netmask: '255.255.255.0'
13.
14.      node.vm.provider :virtualbox do |vb|
15.        vb.name = host[:name]
16.      end
17.    end
18.  end
19.end
```

SUMMARY

```
$ vagrant init user/box     # Create Vagrantfile for specified base box
$ vim Vagrantfile           # Customize your box
$ vagrant up [host]         # Create VM(s) if needed and boot
$ vagrant reload [host]     # After every change to Vagrantfile
$ vagrant halt [host]       # Poweroff
$ vagrant destroy [host]    # Clean up!
$ vagrant ssh [host]        # log in
$ vagrant status [host]     # Status of your VM(s)
```

- TODO Load Order and Merging
- https://www.vagrantup.com/docs/vagrantfile/#load-order-and-merging

PROVISIONING

PROVISIONING

= From *Just Enough Operating System* to fully functional configured box

- **Shell script**
- **Ansible**
- **Puppet** (Apply + Agent)
- Chef (Solo + Client)
- Docker
- Salt

SHELL PROVISIONING

- Add to your Vagrantfile

```
config.vm.provision 'shell', path: 'provision.sh'
```

Put the script into the same folder as Vagrantfile

RECOMMENDED WORKFLOW

- First do the installation manually (`vagrant up`)
- Make sure every command runs without user interaction!
- Record every command in the script
- If everything works:

```
vagrant destroy -f && vagrant up
```

PROVISIONING SCRIPT

- Installs Apache and PHP

```
#!/bin/bash -eu
# provision.sh -- Install Apache and a test PHP script

sudo rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
yum install -y httpd php

service httpd start
chkconfig httpd on

cat > /var/www/html/index.php << EOF
<?php phpinfo(); ?>
EOF
```

MySQL is left as an exercise for the reader ;-)

SYNCED FOLDERS

- Add to your Vagrantfile :

```
config.vm.synced_folder 'html', '/var/www/html'
```

- Create folder html in your project root

```
$ tree
.
|-- html
|  `-- index.php
|-- provision.sh
`-- Vagrantfile
```

- **Vagrant reload**

DISADVANTAGES OF SHELL PROVISIONING

- Not very flexible
- Script should be non-interactive
- Not scalable
  - Long Bash scripts are horrible!
- ***Idempotence*** not guaranteed
  - What happens when you run provision script multiple times?
  - Change to script is expensive:

```
vagrant destroy && vagrant up
```