vagrant up!

slingshot: engaged

VMs to the rescue!

turning

“works on my box”

into

“works on the provisioned VM”

Ok, that catch phrase needs a bit of work...

VMs without providers

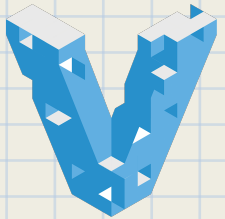
VMware Fusion is better!
You only need the player!
Parallels runs Windows x.x best on OS X!
Virtualbox is too slow!
RAWR!

VAGRANT

Vagrant keeps VM “providers” at arms length

minimizing the need to just “pick one” VM solution for all team members and projects

VM speed or dev process speed?



hosted VM images?

VM “provider” specific images -

licensed and costly
locked into a particular player version
VM specific tooling required
transitioning VMs to new projects NOT easy

base box provisioning

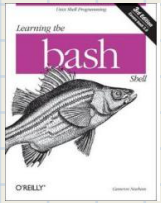
Provisioning -

act of installing **and** configuring applications onto a base box image

shell script provisioning

- hard to support
- harder to teach
- easy to screw up

sound like something you really want, right?



provisioning with chef

- still very complicated
- still hard to teach
- easy to support DING! DING! DING!

puppet?

PUPPET?!?!? WAAAAAAA!

eh - religious wars, who cares
I picked chef, if you like puppet, pick puppet.

LIVE CODING!!!!!!!

- set aside 45 min

change the way you dev

by the end of this, you'll be able to
clone a new ubuntu 13.10 box for every app

```
vagrant init saucy64-java-node
vagrant up
vagrant ssh
```

disk is cheap your time is not

install

download/install Vagrant (10 min)

http://www.vagrantup.com/downloads.html

the default VM provider for Vagrant is Virtualbox

it is free to use and plenty fast...enough

configure Vagrant plugins

install chef solo plugins

```
vagrant plugin install vagrant-omnibus
```

add berkshelf (chef cookbook versioning)

```
vagrant plugin install vagrant-berkshelf \ --
plugin-version '>= 2.0.1'
```

OPSCODE

berkshelf?

well, a chef needs a place to store their cookbooks….put ‘em on the berkshelf!

I can’t make this stuff up -

(hides in corner)

standup first VM

set aside 20+ min, and run

```
mkdir basebox; cd basebox
curl -o Vagrantfile \
191.236.23.180/azureuser/vagrant/Vagrantfile
curl -o Berksfile \
191.236.23.180/azureuser/vagrant/Berksfile
vagrant up
vagrant ssh
```

sooo, what did I just kickoff?

download two files

- `Vagrantfile` - vm opts/‘basebox’ image with app ‘recipes’
- `Berksfile` - app cookbook manager. defines where to get cookbooks/recipes and the versions

two vagrant commands

- `vagrant up` – download and ‘provisions’ the box
- `vagrant ssh` – login to your newly provisioned VM

20 minutes?!? really!?!?

yup, provisioning can take...time.

guess which recipe is the culprit?
(hint: it's not git or emacs)

- `git`
- `java 1.7 ← hint, hint`
- `nodejs`
- `nginx`
- `emacs`

it will eventually finish...

When it does, ubuntu 13.10 VM is ready to go

I know it’s tempting to remove one of the chef recipes to speed up provisioning -

hold off on that for a second

start & stop w/Vagrant

start VM

```
vagrant up
```

login to VM

```
vagrant ssh
```

stop VM

```
vagrant halt
```

who waits 20 min for a VM?

chillax - only takes 20+ min the **FIRST** time

save your progress

Provisioning a new base box is rare. The starting place where you app begins

- package - convert VM into static .box file
- add - make .box file a vagrant ‘template’
- remove - remove template from vagrant

package VM into .box

turn the VM into a ‘base’ box image

```
vagrant package --output saucy64-java-node.box
```

run the command in the directory with a Vagrantfile. creates a .box file to be shared

add .box to vagrant

make the box available for quick access

```
vagrant box add saucy64-java-node \
./saucy64-java-node.box
```

when run in a dir w/saucy64-java-node.box file adds the box to list of available images

remove box from vagrant

with

```
vagrant box remove saucy64-java-node
```

once removed a named box can no longer be brought up without adding a .box file to vagrant again. useful when replacing a base box image

saucy64 wat?

Ubuntu 13.10 (Saucy Salamander) 64-bit

http://releases.ubuntu.com/13.10/

share your hard work

host the .box file on a web server and

BAM!

standing up instances of this ‘provisioned’ VM is as fast as your network and hard drive

whoa...

worth repeating -

After a VM is packaged into a .box file, you can **share** it

standing up copies of the box now takes seconds

welcome to the future

my flow’s so tight

develop on Vagrant VMs
version w/git
write the recipe as I dev the app
provision locally w/chef solo+berkshelf
practice provisioning often