chef solo + vagrantup

cooking for one

pre-reqs.

vagrantup + 2 plugins

- vagrant-omnibus
- vagrant-berkshelf

installing Vagrant plugins

at the prompt -

```
vagrant plugin install vagrant-omnibus

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

vagrantup + Chef sample on github

Vagrantfile + a local Chef cookbook/recipe:

```
https://github.com/patarleth/chef-solo-sample-app
```

the cookbook installs and runs a nodejs/Express app w/upstart:

```
https://github.com/patarleth/sample-express-app
```

chef-solo-sample-app

1. Vagrantfile
2. Berksfile loads a ‘sample-app’ cookbook
3. sample-app cookbook folder itself

sample-app cookbook

The sample-app default recipe does 2 things -

- clones a nodeJs/Express application using git
- and installs as an upstart service using chef

three basic cookbook folders

- sample-app/attributes
- sample-app/recipes
- sample-app/templates/default

attributes

sample-app/attributes contains a single file -

default.rb

- default values for keys used in the recipe that are modified when provisioned

(written in ruby)

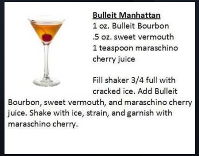
recipes

sample-app/recipes contains a single file -

default.rb

- recipe that is run when **ONLY** the cookbook name is specified

(written in ruby)



Bulleit Manhattan
1 oz. Bulleit Bourbon
.5 oz. sweet vermouth
1 teaspoon maraschino cherry juice

Fill shaker 3/4 full with cracked ice. Add Bulleit Bourbon, sweet vermouth, and maraschino cherry juice. Shake with ice, strain, and garnish with maraschino cherry.

templates/default

sample-app/templates/default -

- templates available to recipes, customized using attribute values

(written in erb aka embedded ruby)

configure the VM

Vagrantfile

ubuntu 13.10 base box + nodeJs

```
config.vm.box = "saucy64-java-node"
config.vm.box_url = "http://191.236.23.180
/azureuser/vagrant/saucy64-java-node.box"
```

chef-solo config

```
config.berkshelf.enabled = true
config.omnibus.chef_version = :latest
```

provision ‘sample-app’ with chef

```
config.vm.provision :chef_solo do |chef|
  chef.log_level = :info
  chef.add_recipe "sample-app"

  chef.json = {
    :sample_app => { :name => “hello nurse” }
  }
end
```

add sample-app folder as cookbook

Berksfile -

```
site :opscode


cookbook 'chef-solo-search', '~> 0.5.1'
cookbook 'sample-app', path: './sample-app'
```

Berksfile cookbooks using git

shipping a recipe with the Vagrantfile and Berksfile is not supportable long term.

git to the rescue! -

```
cookbook 'sample-app', git: 'https://github.
com/patarleth/sample-app-cookbook.git'
```

opscode community?

OPSCODE

opscode community used for this tutorial

This is just asking for trouble.

The best solution is to run a chef server and vet the recipes loaded from the community.

wrap it up

wouldn’t it be nice to -

- dev and test on the actual deployment platform
- develop provisioning w/project, not at the end
- embrace clear, consistent dev strategies and tools
  - SCM - git
  - CI - jenkins/teamcity
  - provisioning - chef