

Test Kitchen and Inspec for Your Infrastructure

Aaron Blythe

sponsors

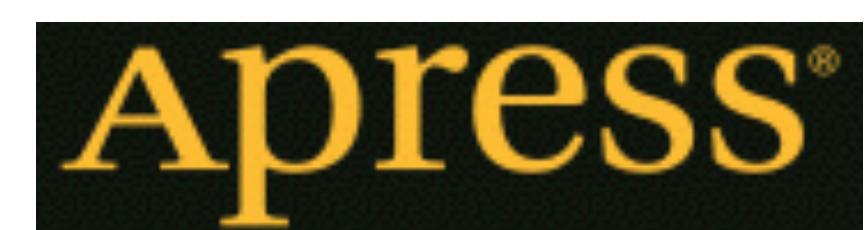
platinum

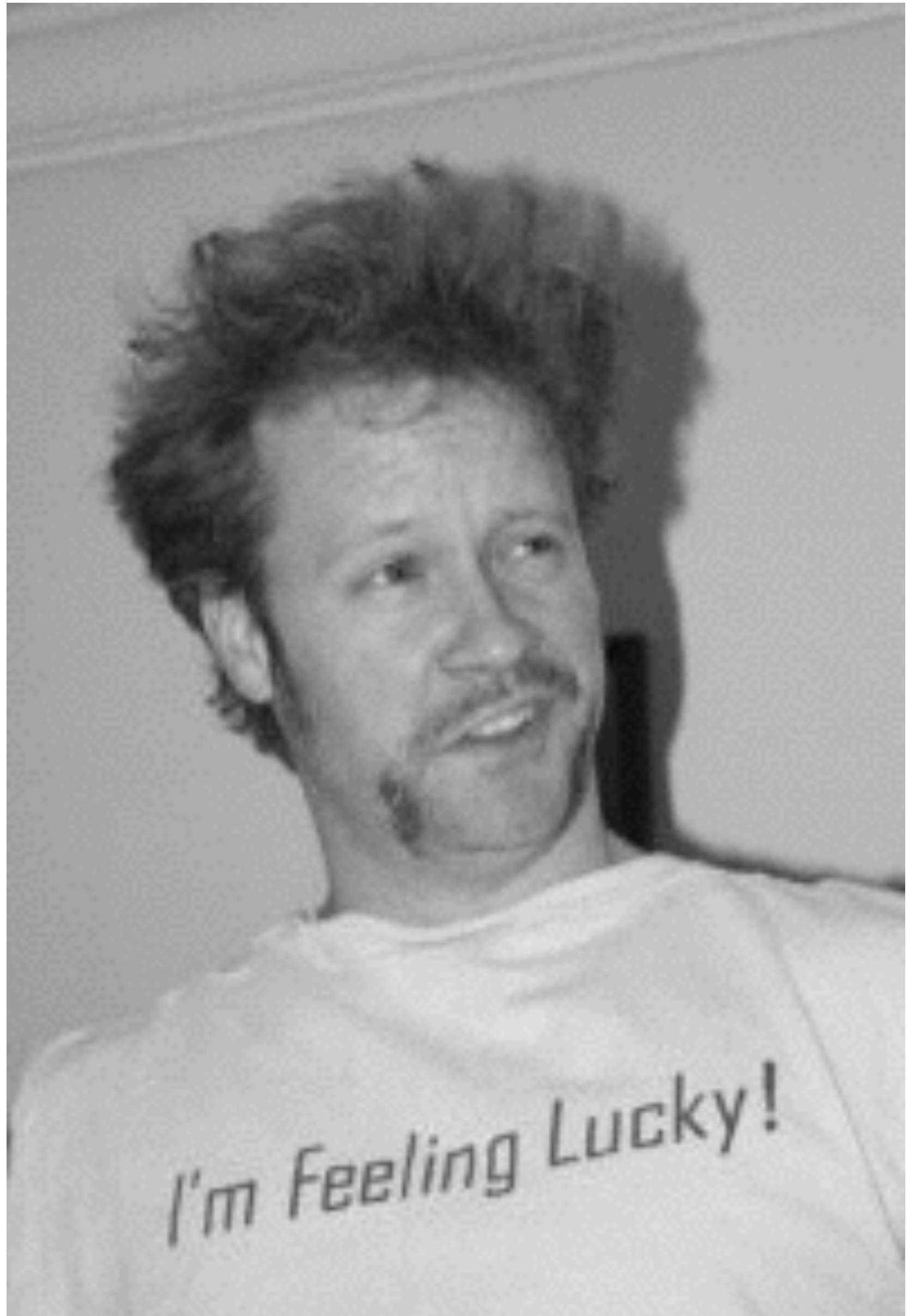


gold



silver





Aaron Blythe
@ablythe

DevOps 
Kansas City

 @DevOpsKC

<http://www.meetup.com/DevOps-Kansas-City/>

<http://devopsdayskc.org/>



CFP is open!!

<http://cfp.devopsdayskc.org/>

https://github.com/aaronblythe/hello_webpage_chef

@ablythe



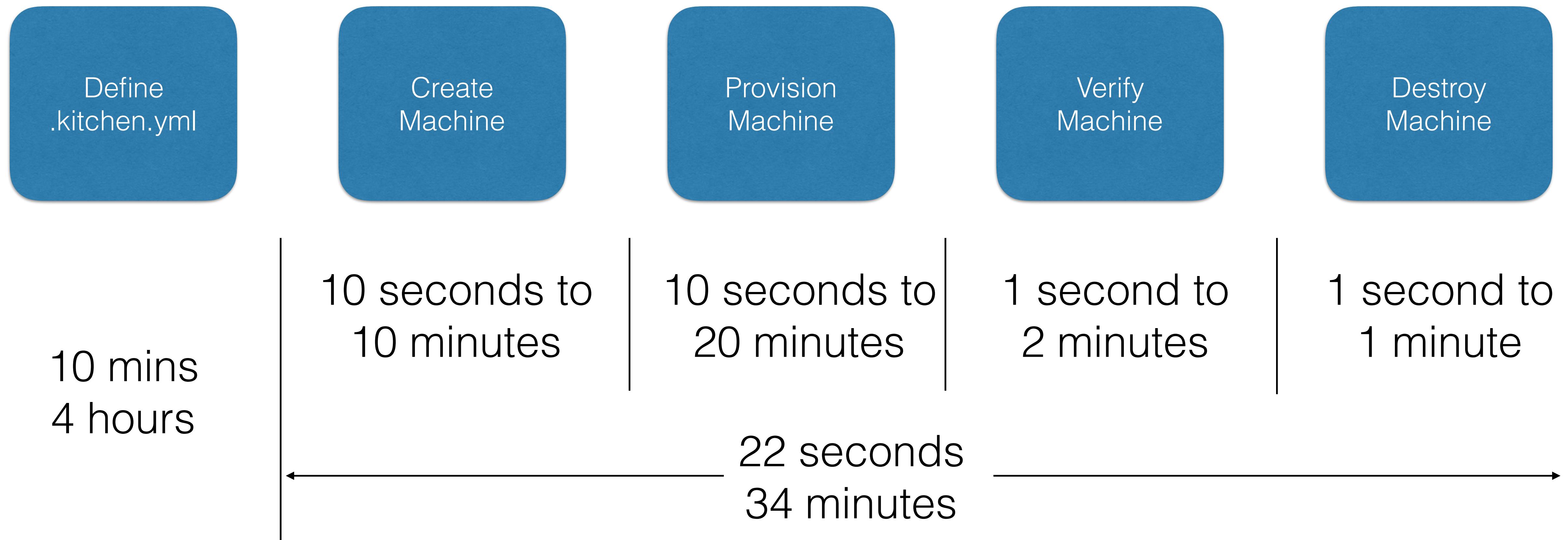
<http://kitchen.ci/>



<http://inspec.io/>

@ablythe

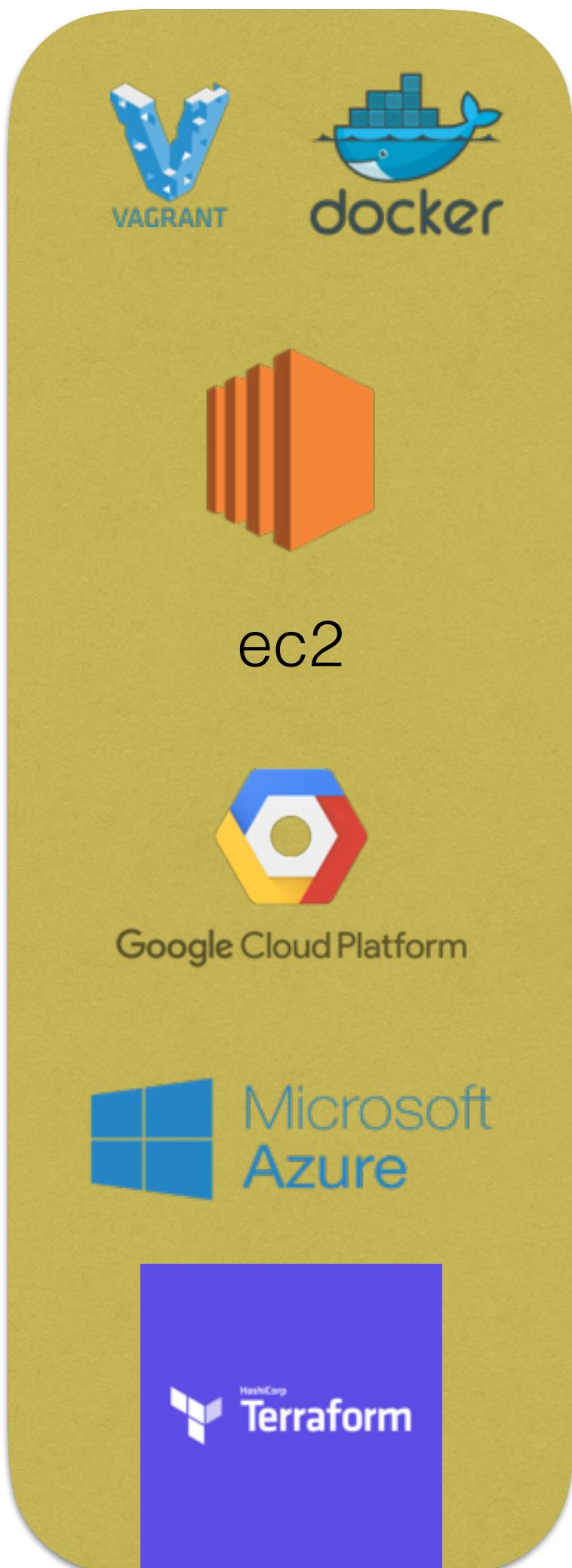
Test Kitchen Steps



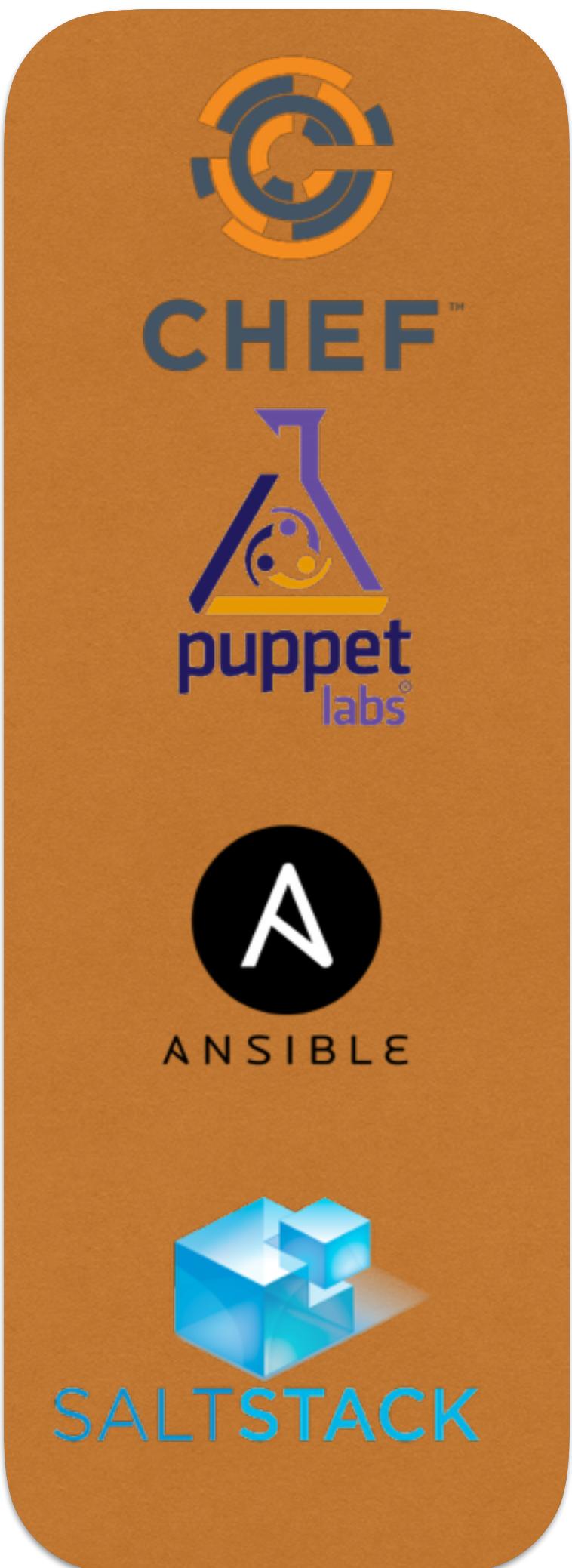
Enterprise circa 2013 vs. test-kitchen

	Enterprise VMWare	On Prem Test Cloud	Test Kitchen
Time to create/acquire VM's	1 week to 3 months	10 minutes to 2 days	1 minute to 30 minutes
Time to recreate VM's	1 week to 3 months Reclaim: 2-4 times/year	15 minutes to 4 hours	1 minute to 5 minutes
Method to create	Log an Issue - Manual work by separate team	Manual or internally built brittle tooling	Command Line
VM OS Settings	Log an Issue: to create modifications	Log an Issue: to create modifications	Research and test on next run

Driver



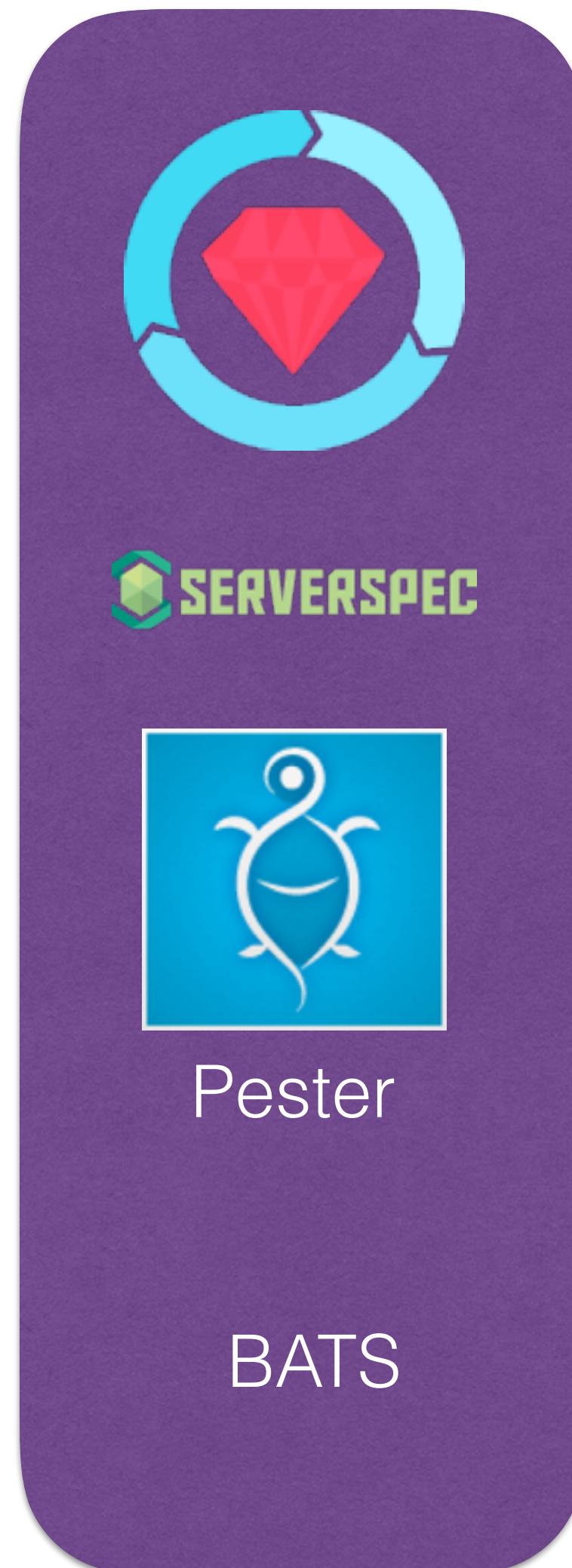
Provisioner



Platform



Verifier



@ablythe



<https://downloads.chef.io/chefdk>

@ablythe

Install ChefDK on Mac



```
# install chef tools
brew cask install chefdk
brew cask install virtualbox
brew cask install vagrant
brew cask install vagrant-manager

# upgrade
brew cask reinstall chefdk
brew cask reinstall virtualbox
brew cask reinstall vagrant
brew cask reinstall vagrant-manager
```

Install ChefDK on Windows



```
#install chef tools
choco install chefdk
choco install virtualbox
choco install vagrant
choco install vagrant-manager

# upgrade
choco upgrade chefdk
choco upgrade virtualbox
choco upgrade vagrant
choco upgrade vagrant-manager
```

Vagrant/Virtualbox



VAGRANT





Windows®

.kitchen.windows.yml

```
---
```

```
driver:
  name: vagrant
  network:
    - ["private_network", {ip: "192.168.34.36"}]
```

```
provisioner:
  name: chef_zero
```

```
verifier:
  name: inspec
```

```
platforms:
  - name: mwrock/Windows2012R2
    os_type: windows
    transport:
      name: winrm
      elevated: true
```

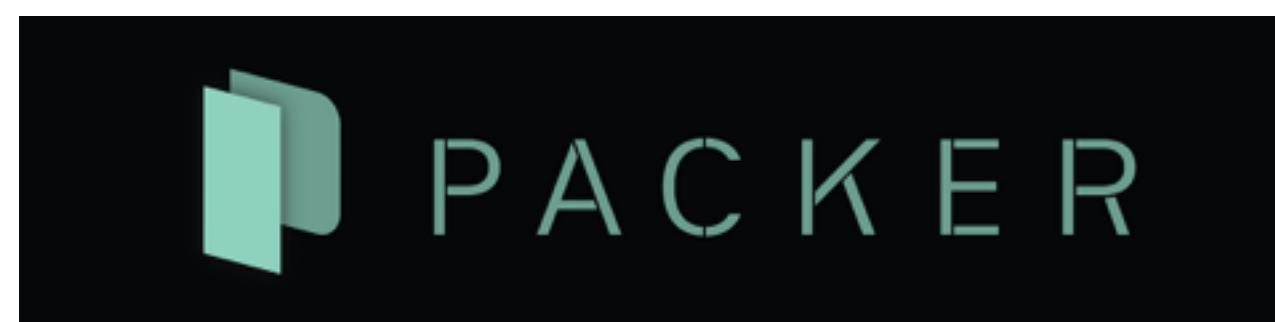
```
suites:
  - name: default
    run_list:
      - recipe[hello_webpage_chef::default]
    attributes:
```

Kitchen with Windows (Vagrant/Virtualbox)

```
export KITCHEN_YAML=.kitchen.windows.yml
chef exec kitchen converge
```

Where to Get Windows Images

- From Microsoft:
 - <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>
- Matt Wrock images on Hashicorp Atlas
 - <https://atlas.hashicorp.com/mwrock>
- Create your own using Joe Fitzgerald's Packer Templates
 - <https://github.com/joefitzgerald/packer-windows>



@ablythe

Setup Run

```
Vagrant instance <default-mwrock-Windows2012R2> created.  
Finished creating <default-mwrock-Windows2012R2> (2m11.40s) .  
...
```

```
Chef Client finished, 3/3 resources updated in 03 minutes 36 seconds  
Finished converging <default-mwrock-Windows2012R2> (4m28.81s) .
```

Kitchen Login

```
open http://192.168.34.36
chef exec kitchen login
```



redhat.

@ablythe

.kitchen.yml

```
---
```

```
driver:
```

```
  name: vagrant
```

```
provisioner:
```

```
  name: chef_zero
```

```
verifier:
```

```
  name: inspec
```

```
platforms:
```

```
  - name: ubuntu-16.04
```

```
  - name: centos-7.3
```

```
suites:
```

```
  - name: default
```

```
    run_list:
```

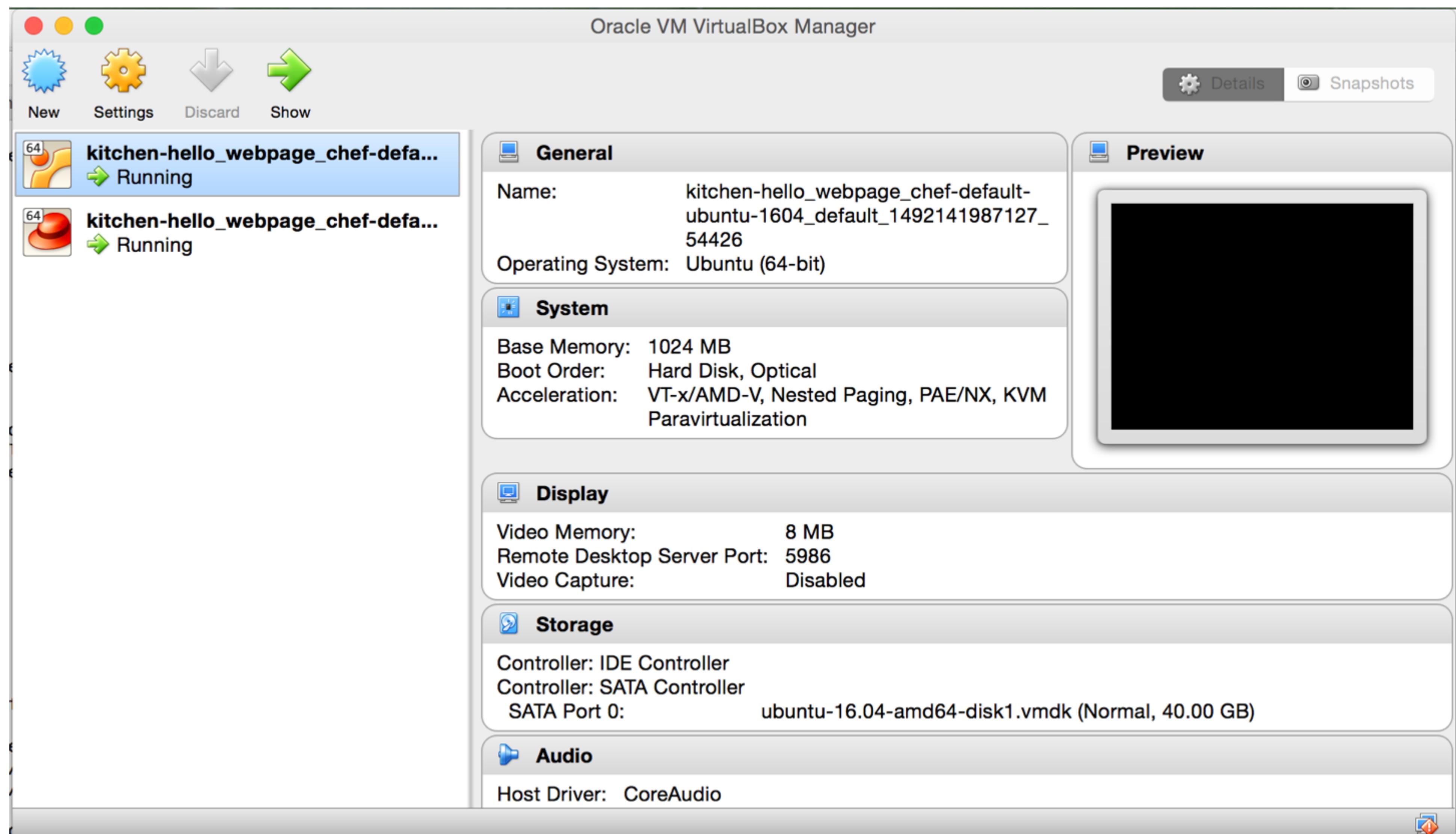
```
      - recipe[hello_webpage_chef::default]
```

```
attributes:
```

```
chef exec kitchen list
chef exec kitchen converge
...
Chef Client finished, 52/115 resources updated in 21 seconds
Finished converging <default-ubuntu-1604> (0m39.31s).

...
Finished converging <default-centos-73> (0m22.75s).
-----> Kitchen is finished. (2m38.52s)
```

Virtual Box



@ablythe

Kitchen Login

```
chef exec kitchen list  
chef exec kitchen login default-ubuntu-1604
```

.kitchen.yml

```
---
```

```
driver:
  name: vagrant
```

```
provisioner:
  name: chef_zero
```

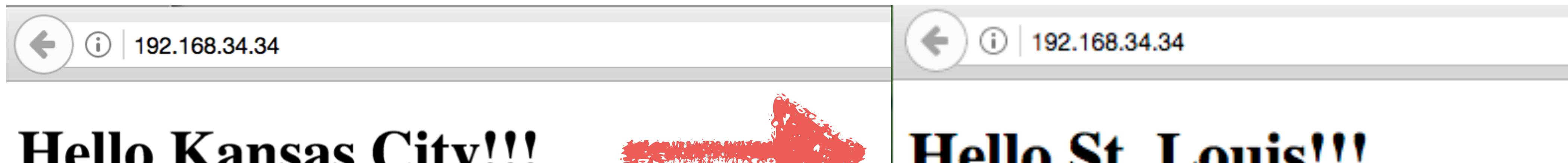
```
verifier:
  name: inspec
```

```
platforms:
  - name: ubuntu-16.04
    driver_config:
      network:
        - ["private_network", {ip: "192.168.34.34"}]
  - name: centos-7.3
    driver_config:
      network:
        - ["private_network", {ip: "192.168.34.35"}]
```

```
suites:
  - name: default
    run_list:
      - recipe[hello_webpage_chef::default]
    attributes:
```

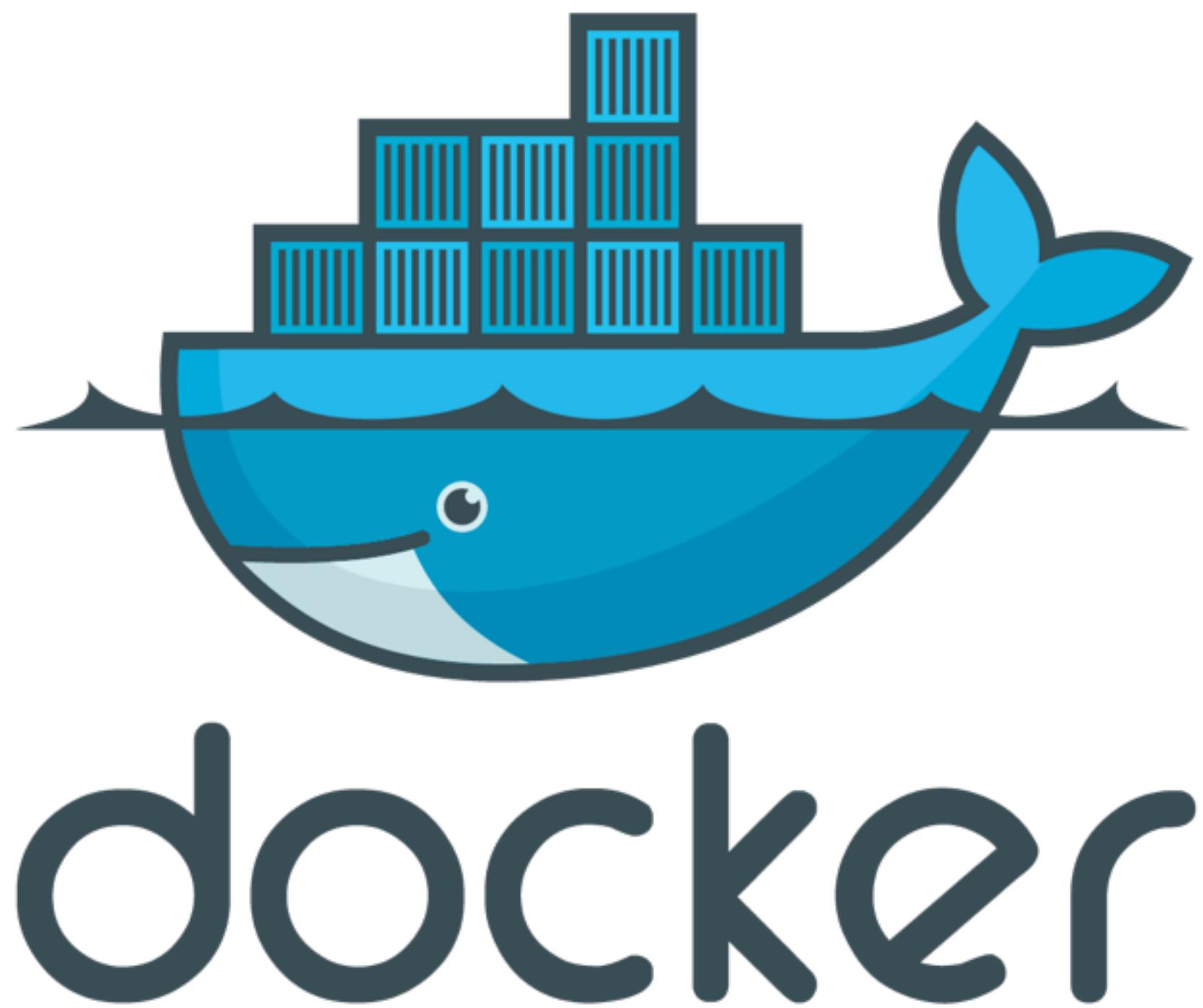
Webpage Local Network

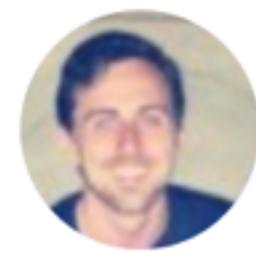
```
chef exec kitchen converge  
open http://192.168.34.34
```



```
chef exec kitchen destroy
```

Docker



Shane da Silva [Follow](#)

Software engineer and trampoline aficionado.

Mar 16, 2015 · 6 min read

Reduce Chef infrastructure integration test times by 75%

with test-kitchen and Docker

<https://brigade.engineering/reduce-chef-infrastructure-integration-test-times-by-75-with-test-kitchen-and-docker-bf638ab95a0a> @ablythe

.kitchen.docker.yml

```
driver:
  name: docker
  forward: 80
  use_sudo: false

provisioner:
  name: chef_solo

Verifier:
  name: inspec

platforms:
  - name: ubuntu-16.04

suites:
  - name: default
    run_list:
      - recipe[hello_webpage_chef::default]
attributes:
```

Kitchen Docker

```
chef gem install kitchen-docker  
export KITCHEN_YAML=.kitchen.docker.yml  
chef exec kitchen converge
```

docker ps -a

```
docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS
PORTS
300d78a9450b        7d6f475b3253      "/usr/sbin/sshd -D..."   About a minute ago   Up
About a minute      0.0.0.0:32770->22/tcp, 0.0.0.0:32769->80/tcp   defaultubuntu1604-ablythe-
AaronsMBP2-owxo2zer
```

open 0.0.0.0:<ip from docker ps>

chef exec kitchen destroy

AWS EC2



Environment Variables

```
#!/usr/bin/env bash

export KITCHEN_YAML=".kitchen.cloud.yml"
export AWS_ACCESS_KEY=<key>
export AWS_SECRET_KEY=<secret>
export AWS_SSH_KEY_ID=<key_id>
export AWS_SSH_KEY_FULL_PATH="/Users/<yourname>/.ssh/<yourkey>.pem"
export AWS_SECURITY_GROUP_ID=<security_group_id>
export AWS_SUBNET_ID=<subnet_id>"
```

.kitchen.cloud.yml

```
---
```

```
driver:
  name: ec2
  aws_ssh_key_id: <%= ENV['AWS_SSH_KEY_ID'] %>
  security_group_ids: <%= ENV['AWS_SECURITY_GROUP_ID'] %>
  subnet_id: <%= ENV['AWS_SUBNET_ID'] %>
  region: eu-west-1

transport:
  ssh_key: <%= ENV['AWS_SSH_KEY_FULL_PATH'] %>
  username: ubuntu

provisioner:
  name: chef_solo

verifier:
  name: inspec

platforms:
  - name: ubuntu-16.04
    driver:
      image_id: ami-405f7226
```

Kitchen AWS EC2

```
chef gem install kitchen-ec2
export KITCHEN_YAML=.kitchen.cloud.yml
chef exec kitchen converge
```



@ablythe



@ablythe



463596200

@ablythe



Kitchen EC2

```
chef exec kitchen login  
open http://<ip from converge output>  
chef exec kitchen destroy
```

Many Configuration Management Tools

The screenshot shows four GitHub repository pages arranged vertically, each representing a different configuration management tool:

- kitchen-ansible**: A repository for Ansible provisioning. It has 10 pull requests, 204 stars, and 86 forks. The code tab is selected.
- kitchen-puppet**: A repository for Test Kitchen and Puppet provisioning. It has 9 pull requests, 70 stars, and 60 forks. The code tab is selected.
- kitchen-salt**: A repository for SaltStack provisioning. It has 16 pull requests, 15 issues, 10 commits, 0 boards, 0 reports, 0 projects, and 0 wiki pages. It has 16 watch notifications, 105 stars, and 62 forks. The code tab is selected.
- hiera-test-kitchen**: A repository for Hiera provisioning. It has 168 commits, 4 branches, 7 releases, 15 contributors, and an MIT license. It has 0 watch notifications, 0 stars, and 0 forks. The code tab is selected.

Common UI elements include the GitHub logo, navigation links for Pull requests, Issues, Gist, and ToDo, and various repository statistics like commits, branches, releases, and contributors. The bottom of the page shows a commit history and download/upload options.

Many Configuration Management Tools

Ansible

```
chef gem install kitchen-ansible
```

```
provisioner:  
  name: ansible_playbook  
  roles_path: roles  
  hosts: tomcat-servers  
  require_ansible_repo: true  
  ansible_verbose: true  
  ansible_version: latest  
  require_chef_for_busser: false  
  additional_ssh_private_keys:  
    - /mykey/id_rsa
```

Puppet

```
chef gem install librarian-puppet  
chef gem install kitchen-puppet
```

```
provisioner:  
  name: puppet_apply  
  manifests_path: /repository/  
  puppet_repo/manifests  
  modules_path: /repository/  
  puppet_repo/modules-mycompany  
  hiera_data_path: /repository/  
  puppet_repo/hieradata
```

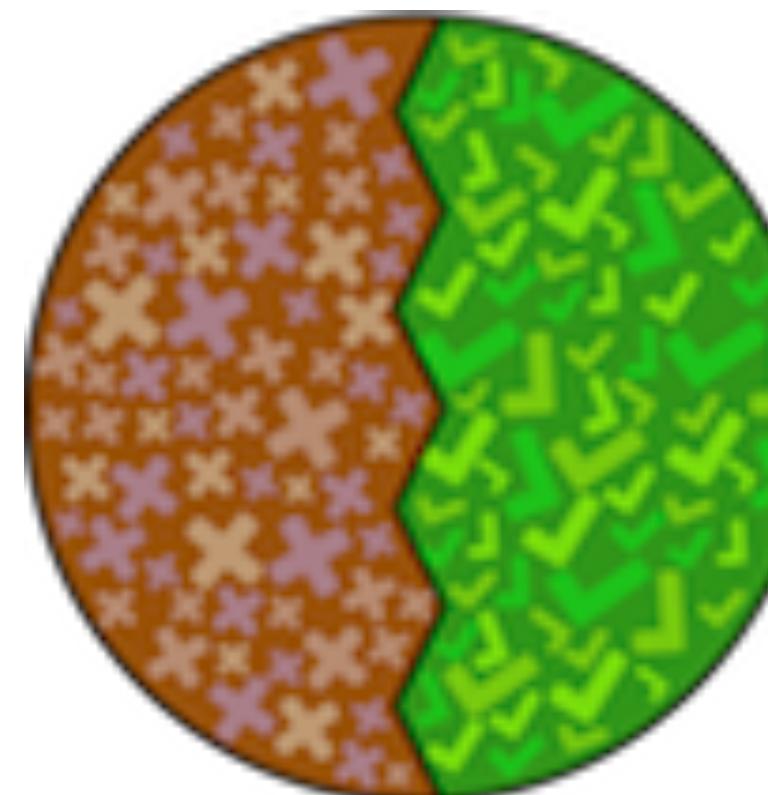
Salt

```
chef gem install kitchen-salt
```

```
provisioner:  
  name: salt_solo  
  formula: beaver  
  state_top:  
    base:  
      '*':  
        - beaver  
        - beaver.ppa
```

Kitchen Terraform

<https://github.com/newcontext-oss/kitchen-terraform/>



<https://youtu.be/1Kj3I4PZXdE>

@ablythe

Kitchen Cloudformation

This repository Search Pull requests Issues Gist ToDo

neillturner / **kitchen-cloudformation** ⋮ Watch 3 Star 6 Fork 4

Code Issues 0 Pull requests 0 Boards Reports Projects 0 Wiki

A Test Kitchen Driver for Amazon AWS CloudFormation.

61 commits 1 branch 3 releases 3 contributors

Branch: master New pull request Click for language details Create new file Upload files Find file Clone or download

neillturner bump version Latest commit e8ad868 on Mar 17

File	Commit Message	Time
lib/kitchen/driver	bump version	a month ago
.gitignore	kitchen driver for aws clouformation	2 years ago
.rubocop.yml	ci	a year ago
.travis.yml	ci	4 months ago
CHANGELOG	...	

InSpec

<http://inspec.io/docs/reference/resources/>

Join us in May at ChefConf to learn more about compliance automation!

 Dismiss



Tutorials

Docs

Community

 Github

 Try the Demo

 Download

GETTING STARTED

[Overview](#)

[Get InSpec](#)

[Tutorials](#)

[InSpec and friends](#)

REFERENCE

[inspec executable](#)

[Profiles](#)

[Resources](#)

[Matchers](#)

[InSpec DSL](#)

InSpec Resources Reference

The following InSpec audit resources are available:

[apache_conf](#)

[apt](#)

[audit_policy](#)

[auditd_conf](#)

[auditd_rules](#)

[bash](#)

[bond](#)

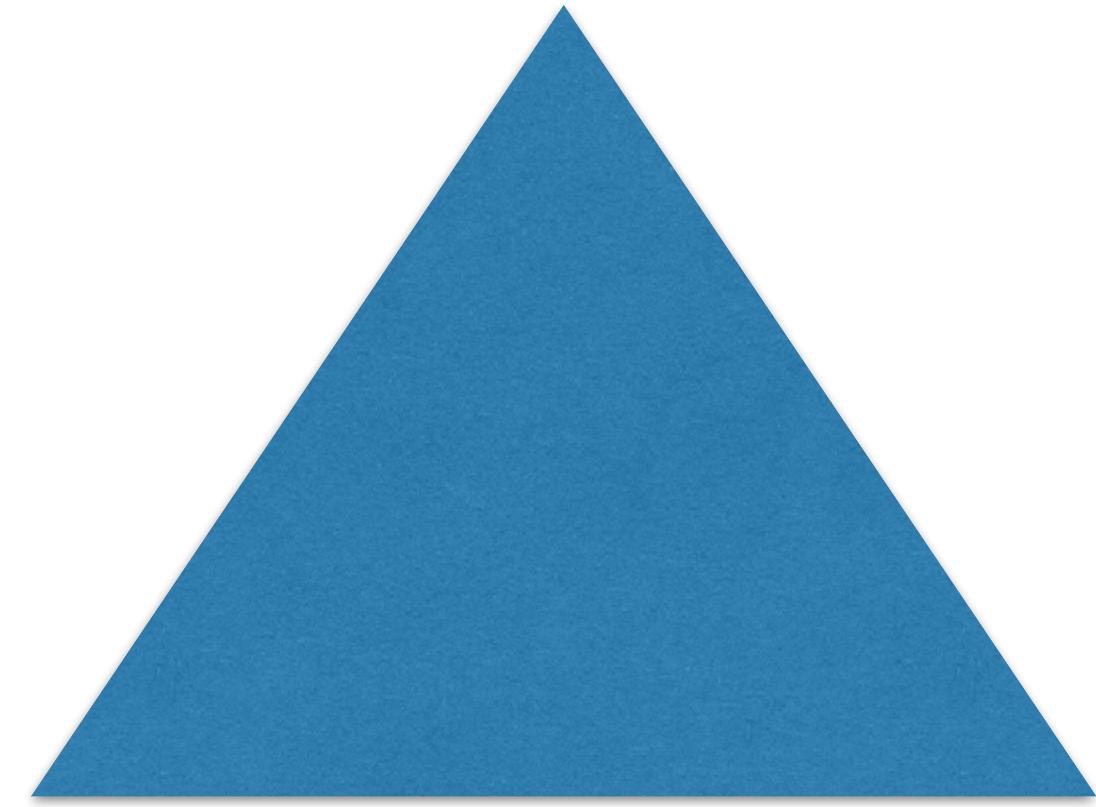
[bridge](#)

@ablythe

```
export KITCHEN_YAML=.kitchen.docker.yml  
chef exec kitchen verify
```

Trinity of Config Management

Install Package



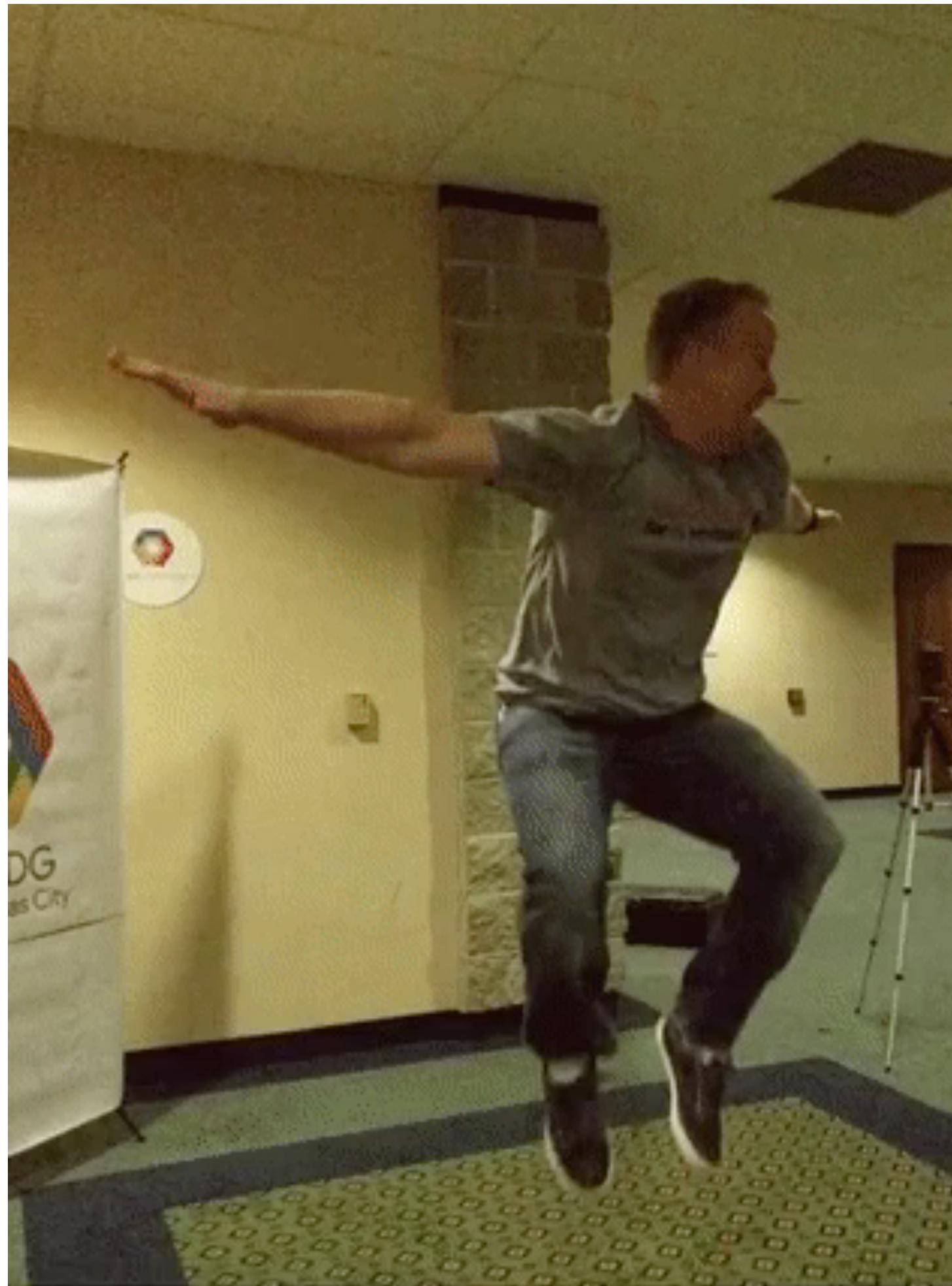
Manage Service

Config Files

Trinity



Trinity



Installed Package

```
describe package('apache2') do
  it { should be_installed }
end
describe command('apache2 -v') do
  its(:exit_status) { should eq 0 }
  its(:stdout) { should match(%r{/Server version: Apache\}/) }
}
end
```

Service

```
describe upstart_service('apache2') do
  it { should be_installed }
  it { should be_running }
end
```

Manage Config Files

```
describe apache_conf do
  its('Listen') { should eq ['*:80'] }
  its('AllowOverride') { should contain 'None' }
end

describe file('/var/www/html/index.html') do
  it { should exist }
  it { should be_file }
  it { should be_owned_by 'root' }
  its(:content) { should match(/Kansas City/) }
end
```

Per OS Specific Specs

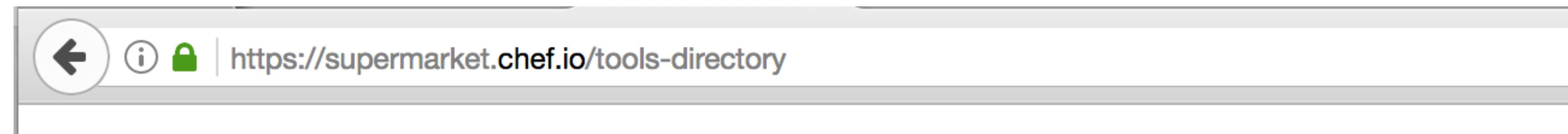
```
if os[:family] == 'debian'
  describe upstart_service('apache2') do
    it { should be_installed }
    it { should be_running }
  end
elsif os[:family] == 'redhat'
  describe service('httpd') do
    it { should be_installed }
    it { should be_running }
  end
end
```

```
Service apache2
  ✓ should be installed
  ✓ should be running
Command apache2
  ✓ -v exit_status should eq 0
  ✓ -v stdout should match /Server version: Apache\2.4/
Apache Config
  ✓ /etc/apache2/apache2.conf Listen should eq ["*:80"]
  ✓ /etc/apache2/apache2.conf AllowOverride should contain "None"
Host example.com
  ✓ should be resolvable
File /etc/apache2/
  ✓ should exist
  ✓ should be directory
  ✓ should be owned by "root"
File /var/www/html/
  ✓ should exist
  ✓ should be directory
  ✓ should be owned by "root"
File /etc/apache2/sites-enabled/my_app.conf
  ✓ should be symlink
File /etc/apache2/sites-available/my_app.conf
  ✓ should exist
  ✓ should be file
  ✓ should be owned by "root"
File /var/www/html/index.html
  ✓ should exist
  ✓ should be file
  ✓ should be owned by "root"
  ✓ content should match /Kansas City/
```

http://dev-sec.io

 CHEF CHEF.IO	 puppet labs®	 ANSIBLE
All-in-one Example GitHub	All-in-one Example GitHub	All-in-one Example GitHub
Base Operating System GitHub Chef Supermarket	Base Operating System GitHub Puppet Forge	Base Operating System GitHub Ansible Galaxy
SSH GitHub Chef Supermarket	SSH GitHub Puppet Forge	SSH GitHub Ansible Galaxy
MySQL GitHub Chef Supermarket	MySQL GitHub Puppet Forge	Nginx GitHub Ansible Galaxy
PostgreSQL GitHub Chef Supermarket	PostgreSQL GitHub	MySQL GitHub Ansible Galaxy

@ablythe



COOKBOOKS

TOOLS & PLUGINS

MORE INFO

- [Knife Plugin](#)
- [Ohai Plugin](#)
- [Chef Tool](#)
- [Handler](#)
- [Provisioning Driver](#)
- [Kitchen Driver](#)
- [Powershell Module](#)
- [DSC Resource](#)
- [Compliance Profile](#)

@ablythe

Supermarket

CIS Docker Benchmark Compliance Profile

This InSpec compliance profile implement the CIS Docker 1.11.0 Benchmark in an automated way to provide security best-practices tests around Docker daemon and containers in a production environment.

InSpec is an open-source run-time framework and rule language used to specify compliance, security, and policy requirements for testing any node in your infrastructure.

Install & Usage Instructions

See <http://dev-sec.io/>

The screenshot shows a web-based interface for the Dev-sec Hardening Framework. At the top right, there is a logo for 'dev-sec' and the text 'Hardening Framework'. Below this, a section titled 'DETAILS' contains a 'View Source' button. A red box highlights a list titled 'OTHER TOOLS BY HARDENING FRAMEWORK' which includes four items: 'DevSec Apache Baseline' (Compliance Profile), 'DevSec Linux Baseline' (Compliance Profile), 'DevSec Linux Patch Baseline' (Compliance Profile), and 'DevSec MySQL Baseline' (Compliance Profile). Each item has a small blue icon to its left.

Profile	Type
DevSec Apache Baseline	Compliance Profile
DevSec Linux Baseline	Compliance Profile
DevSec Linux Patch Baseline	Compliance Profile
DevSec MySQL Baseline	Compliance Profile

```
---  
driver:  
  name: docker  
  forward: 80  
  use_sudo: false  
  
provisioner:  
  name: chef_solo  
  
verifier:  
  name: inspec  
  sudo: true  
  
platforms:  
  - name: ubuntu-16.04  
  
suites:  
  - name: default  
    run_list:  
      - recipe[hello_webpage_chef]  
verifier:  
  inspec_tests:  
    - supermarket://dev-sec/linux-baseline  
attributes:
```

```
export KITCHEN_YAML=.kitchen.audit.yml  
chef exec kitchen verify
```

```
✓ File /etc/passwd should be readable by owner
✓ File /etc/passwd should be readable by group
✓ File /etc/passwd should be readable by other
✓ File /etc/passwd group should eq "root"
✓ os-04: Dot in PATH variable
✓ Environment variable PATH split should not include ""
✓ Environment variable PATH split should not include "."
✗ os-05: Check login.defs (3 failed)
✓ File /etc/login.defs should exist
✓ File /etc/login.defs should be file
✓ File /etc/login.defs should be owned by "root"
✓ File /etc/login.defs should not be executable
✓ File /etc/login.defs should be readable by owner
✓ File /etc/login.defs should be readable by group
✓ File /etc/login.defs should be readable by other
✓ File /etc/login.defs group should eq "root"
✓ login.defs ENV_SUPATH should include "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin"
✓ login.defs ENV_PATH should include "/usr/local/bin:/usr/bin:/bin"
✗ login.defs UMASK should include "027"
expected "027" to include "027"
✗ login.defs PASS_MAX_DAYS should eq "60"

expected: "60"
got: "99999"

(compared using ==)

✗ login.defs PASS_MIN_DAYS should eq "7"
expected: "7"
got: "0"

(compared using ==)

✓ login.defs PASS_WARN_AGE should eq "7"
✓ login.defs LOGIN_RETRIES should eq "5"
✓ login.defs LOGIN_TIMEOUT should eq "60"
✓ login.defs UID_MIN should eq "1000"
✓ login.defs GID_MIN should eq "1000"
✓ os-06: Check for SUID/ SGID blacklist
✓ [] should be empty
✓ os-07: Unique uid and gid
✓ /etc/passwd uids should not contain duplicates
✓ /etc/group gids should not contain duplicates
✗ os-08: Entropy (expected: >= 1000
got: 202)
✗ 202 should >= 1000
expected: >= 1000
got: 202
✓ os-09: Check for .rhosts and .netrc file
✓ [] should be empty
✓ package-01: Do not run deprecated inetd or xinetd
✓ System Package inetd should not be installed
✓ System Package xinetd should not be installed
✓ package-02: Do not install Telnet server
```

```
Profile Summary: 24 successful, 26 failures, 0 skipped
Test Summary: 87 successful, 47 failures, 0 skipped
>>>>> -----Exception-----
```

@ablythe

The screenshot displays four separate GitHub repository pages arranged vertically, all sharing a common header and sidebar.

Header:

- Top left: GitHub logo, "This repository", "Search".
- Top center: "Pull requests", "Issues", "Gist", "ToDo".
- Top right: Notification bell, "+" (create), user profile.

Repository 1: dev-sec / chef-os-hardening

- Header: "Watch" (32), "Star" (206), "Fork" (54).
- Content: "Code", "Issues", "Gist", "ToDo".

Repository 2: dev-sec / puppet-os-hardening

- Header: "Watch" (25), "Star" (98), "Fork" (35).
- Content: "Code", "Issues", "Gist", "ToDo".

Repository 3: dev-sec / ansible-os-hardening

- Header: "Watch" (39), "Star" (366), "Fork" (63).
- Content: "Code", "Issues (2)", "Pull requests (2)", "Boards", "Reports", "Projects (0)".
- Description: "This Ansible role provides numerous security-related configurations, providing all-round base protection. <http://dev-sec.io/>"
- Tags: "ansible", "sysctl", "protection", "hardening", "role", "playbook", "linux".
- Statistics: "212 commits", "6 branches", "7 releases", "9 contributors".
- Actions: "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", "Clone or download".

Repository 4: dev-sec / skip_fail

- Header: "Watch" (1), "Star" (1), "Fork" (1).
- Content: "Code", "Issues", "Gist", "ToDo".

Left Sidebar (shared across repositories):

- Section: "This chef", "This puppet", "This ansible".
- Section: "hardening", "Code", "Issues", "Pull requests", "Boards", "Reports", "Projects".
- Section: "Branch: master", "Branches", "Commits", "Releases", "Contributors".
- List: "atomic", "attributable", "library", "recipes".
- List: "file", "lib", "mail", "spec".

@ablythe

<https://www.cisecurity.org/>



Google Custom Search

MS-ISAC

CIS
Benchmarks

CIS
Controls

Workforce
Development

Training
& Resources

Products
& Services

About Us

HEADLINES

- ["Privacy Implications for the CIS Controls"](#)
- [CIS announces new cybersecurity tools](#)
- [See more of CIS in the Media](#)

We've improved our suite
of cybersecurity tools



CIS SecureSuite

Click to learn more

@ablythe

Actual Docker Testing

```
---
```

```
driver:
  name: docker
  forward: 80
  use_sudo: false
```

```
provisioner:
  name: shell
```

```
verifier:
  name: inspec
  sudo: true
```

```
platforms:
- name: ubuntu-16.04
  driver config:
    image: ubuntu:16.04
    platform: ubuntu
```

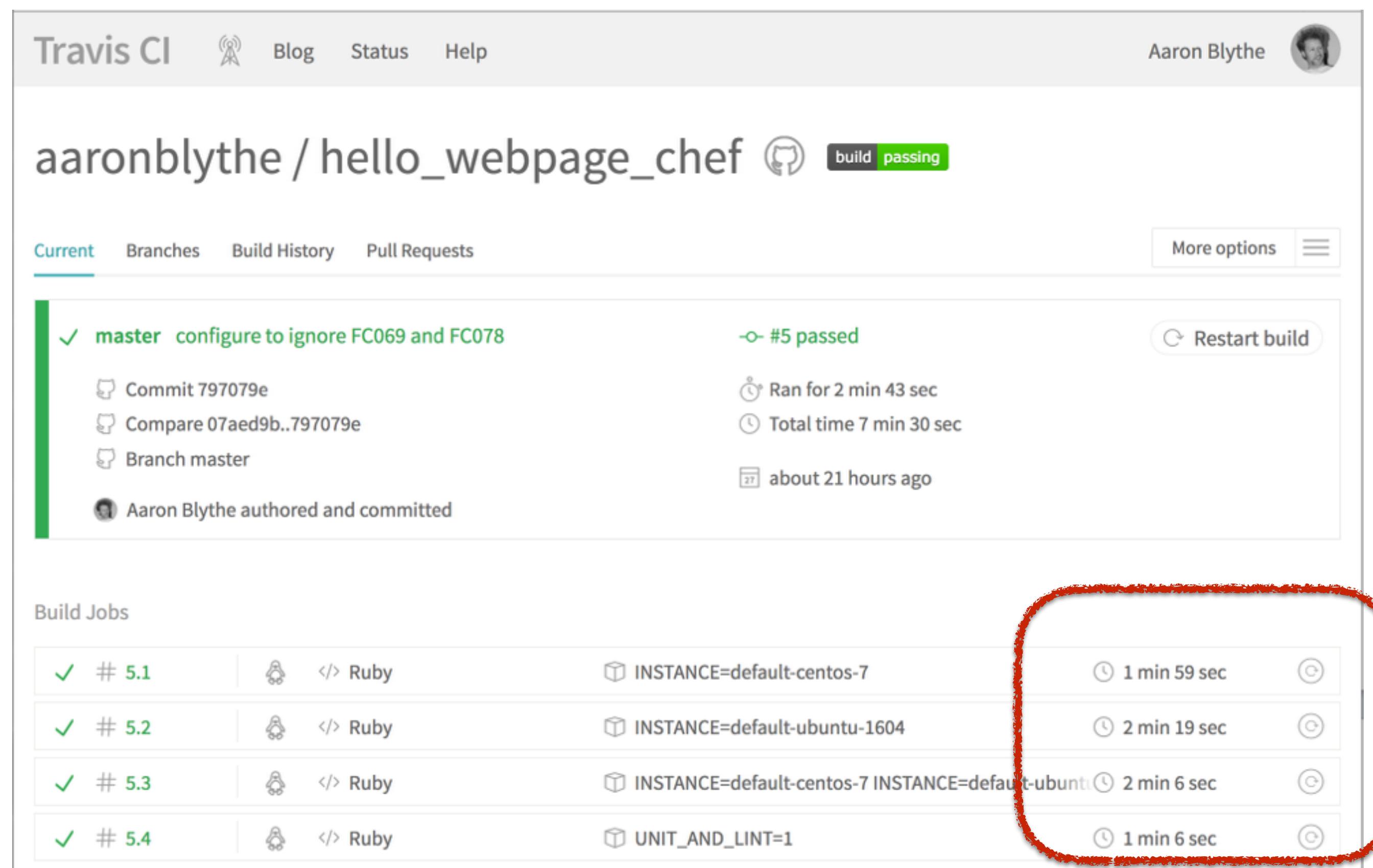
```
suites:
- name: default
  verifier:
    inspec_tests:
      - supermarket://dev-sec/cis-docker-benchmark
```

```
attributes:
```

```
# need version of inspec above 1.21.0
chef gem uninstall inspec
chef gem install inspec
export KITCHEN_YAML=.kitchen.docker.audit.yml
chef exec kitchen converge
```

Continuous Integration - travis-ci

https://travis-ci.org/aaronblythe/hello_webpage_chef



Travis CI Build Status Help Aaron Blythe

aaronblythe / hello_webpage_chef build passing

Current Branches Build History Pull Requests More options

✓ master configure to ignore FC069 and FC078 -o #5 passed Restart build

Commit 797079e Ran for 2 min 43 sec
Compare 07aed9b..797079e Total time 7 min 30 sec
Branch master about 21 hours ago

Aaron Blythe authored and committed

Build Jobs

#	Job Type	Language	Instance	Time
5.1	Ruby	INSTANCE=default-centos-7	1 min 59 sec	
5.2	Ruby	INSTANCE=default-ubuntu-1604	2 min 19 sec	
5.3	Ruby	INSTANCE=default-centos-7 INSTANCE=default-ubuntu-1604	2 min 6 sec	
5.4	Ruby	UNIT_AND_LINT=1	1 min 6 sec	

@ablythe

.travis.yml

```
services: docker

env:
  matrix:
    - INSTANCE=default-centos-7
    - INSTANCE=default-ubuntu-1604

before_script:
  - sudo iptables -L DOCKER || ( echo "DOCKER iptables chain missing" ; sudo iptables -N DOCKER )
  - eval "$( /opt/chefdk/bin/chef shell-init bash )"

script: KITCHEN_LOCAL_YAML=.kitchen.dokken.yml /opt/chefdk/embedded/bin/kitchen verify ${INSTANCE}

matrix:
  include:
    - before_script:
        - eval "$( /opt/chefdk/bin/chef shell-init bash )"
        - /opt/chefdk/embedded/bin/chef --version
        - /opt/chefdk/embedded/bin/cookstyle --version
        - /opt/chefdk/embedded/bin/foodcritic --version
    - script:
        - /opt/chefdk/bin/chef exec rake
  env: UNIT_AND_LINT=1
```

github.com README status

 README.md

hello_webpage_chef

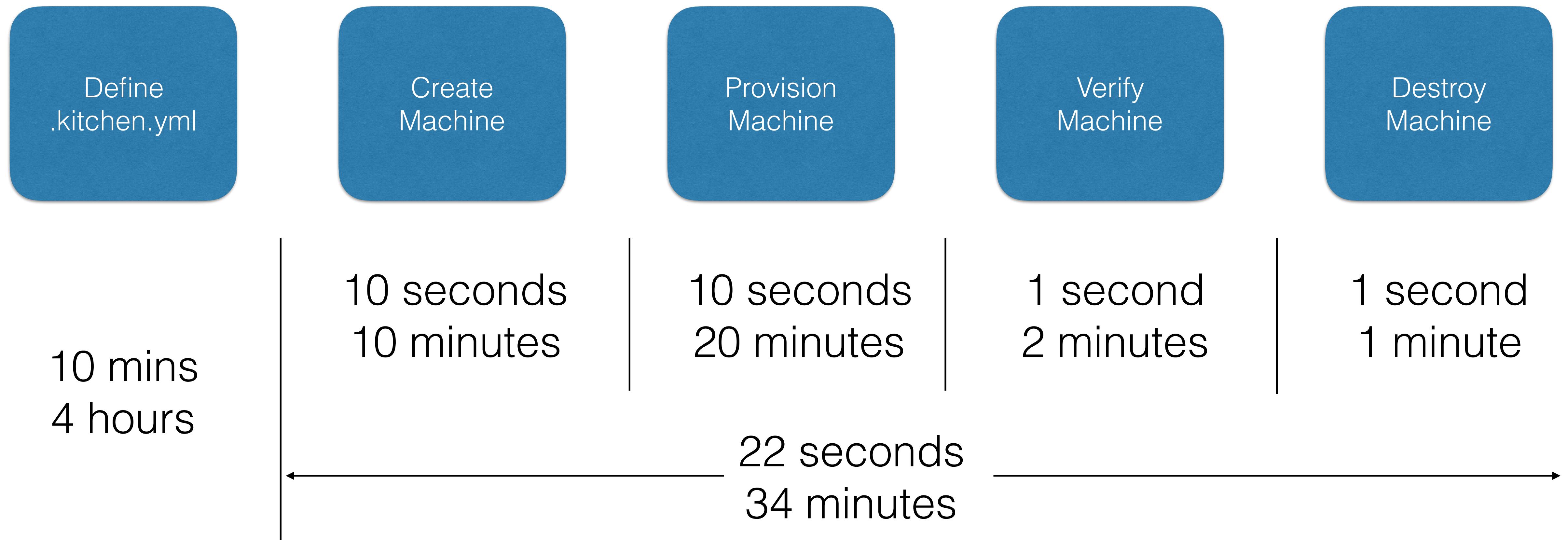
build passing

Simple chef cookbook to exhibit how to do simple things. Exhibits how to use test kitchen in multiple ways.

Testing

@ablythe

Test Kitchen Steps



References and Thanks

- Fletcher Nichol and Chef - Test Kitchen <https://github.com/test-kitchen/test-kitchen>
- Mitchell Hashimoto and Hashicorp - Vagrant, Packer, Terraform and others
- Rob Reynolds - Chocolatey - <https://chocolatey.org/>
- Matt Wrock - all the Windows stuff - <http://www.hurryupandwait.io/>
- Shane da Silva - Brigade Engineering <https://brigade.engineering/reduce-chef-infrastructure-integration-test-times-by-75-with-test-kitchen-and-docker-bf638ab95a0a>
- Sean Porter - Kitchen Docker <https://github.com/portertech>
- Sean O'Meara - Kitchen Dokken <https://github.com/someara/kitchen-dokken>
- Tim Smith - Community Cookbook Support <https://blog.chef.io/2017/04/12/fresh-new-stuff/>
- Dominick and Christoph Hartman - Dev-Sec.io and great Inspec work <http://dev-sec.io/>
- Noah Kantrowitz - CodeRanger - Answering tons of my questions over the years <https://chefcommunity.slack.com/>

announcements

How are we doing? Do survey for chance of winning Xbox One and other prizes.



Wireless
SSID: SilverLiningsAttendees

Password: **Section31**

<http://bit.ly/StLSSurvey>

- Testing should
 - match Production
 - be fast
 - be simple
 - be repeatable

- Machines/containers should be built from scratch