TEST DRIVEN INFRASTRUCTURE COMPLIANCE AS CODE

by

Joaquín Menchaca
À PROPOS DE MOI
ABOUT ME

ROCKET LAWYER
SENIOR DEVOPS
BUILD/RELEASE ENGINEER
aka
Linux Ninja Pants Automation
Engineering Mutant
https://slides.com/devopsstudio/lisa18_tdi
AGENDA

1. Setup
2. Context
3. Chef + InSpec
4. Ansible + InSpec
5. Bonus
SETUP
THE CODE

https://github.com/darkn3rd/lisa18_test_driven_infra
git clone \n  https://github.com/darkn3rd/lisa18_test_driven_infra
cd lisa18_test_driven_infra

# Using Virtual Workstation
vagrant up
vagrant ssh
cd lisa18_test_driven_infra

# Using Host (MacOS X or Linux)
#### Install Requirements

# Using Host (Windows)
#### Install Rrequiremmts
#### Warning: Might not work, had success w/ Chef
THE HOST

must be able to run
Docker
or
Vagrant

Linux

Mac OS X
MINIMUM REQUIRED

ChefDK - bundles ruby, test kitchen, inspec
Docker - fastest way to run stuff (*virtual* virtual machines)
Easiest Path is **Docker Desktop**
PACKAGE MANAGERS

- choco install docker-for-windows
  - https://chocolatey.org/

- brew cask install docker
  - https://brew.sh/

DIRECT DOWNLOAD

- https://www.docker.com/products/docker-desktop
```bash
sudo apt-get update -qq
sudo apt-get install -y apt-transport-https \   curl ca-certificates software-properties-common

DOCKER_REPO="https://download.docker.com/linux/ubuntu"
curl -fsSL ${DOCKER_REPO}/gpg | \   sudo apt-key add -
sudo add-apt-repository \   "deb [arch=amd64] ${DOCKER_REPO} \ ${lsb_release -cs} \ stable"

sudo apt-get update -qq
sudo apt-get install -y docker-ce
sudo usermod -aG docker $USER
```
Easiest Way to Get **Test Kitchen** and **InSpec**
<table>
<thead>
<tr>
<th>PACKAGE MANAGERS</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>choco install chefdk</td>
<td></td>
</tr>
<tr>
<td>chef gem install kitchen-ansible</td>
<td></td>
</tr>
<tr>
<td>chef gem install kitchen-docker</td>
<td></td>
</tr>
<tr>
<td>brew tap chef/chef</td>
<td></td>
</tr>
<tr>
<td>brew cask install chefdk</td>
<td></td>
</tr>
<tr>
<td>chef gem install kitchen-ansible</td>
<td></td>
</tr>
<tr>
<td>chef gem install kitchen-docker</td>
<td></td>
</tr>
</tbody>
</table>

**DIRECT DOWNLOAD**

https://downloads.chef.io/chefdk/
VER=3.2.30
PKG=chefdk_${VER}-1_amd64.deb

# Fetch and Install
wget --quiet ${PREFIX}/${VER}/ubuntu/16.04/${PKG}
sudo dpkg -i ${PKG}

# Local ChefDK Ruby Gems
chef gem install kitchen-ansible
chef gem install kitchen-docker
CONTEXT
WHY?
WHY?

YES WE CODE
WE CODE SYSTEMS

Systems

@Joachim8675309
WE CODE PLATFORMS

Platforms

@Joachim8675309
WE CODE INFRASTRUCTURE

Infrastructure

@Joachim8675309
WHY?

HE WHO CAN DESTROY A THING CONTROLS A THING
WHY?

Destroy and Recreate with Ease
Because Tests...

- Disaster Recovery, Rollback
- Migrate/Replicate New Environments
- Swap out components, Experiment
- Rapid (Like Seriously) Development
WHEN?
WHEN YOU HAVE AN ARTIFACT?
WHAT?
WHAT TO TEST?
HOW?
JUST KIDDING!
ABOUT THE TOOLS
Create (Vagrant) → Converge (Chef) → Verify (InSpec)
---

driver:
    name: vagrant
    provider: hyperv

provisioner:
    name: chef_zero

verifier:
    name: inspec

platforms:
    - name: ubuntu-16.04

suites:
    - name: default
Create (Docker) → Converge (Ansible) → Verify (InSpec)
---
driver:
    name: docker

provisioner:
    name: ansible_playbook

verifier:
    name: inspec

platforms:
    - name: ubuntu-16.04

suites:
    - name: default
control 'ez_apache-contract-01' do
  describe port(80) do
    it { should be_listening }
  end
end

control 'ez_mysql-security-conf-01' do
  describe file(mysql_data_path) do
    it { should be_directory }
    it { should be_owned_by 'mysql' }
    it { should be_grouped_into 'mysql' }
  end
end
package 'apache2'

service 'apache2' do
  action %i(enable start)
  supports(
    status: true,
    restart: true,
    reload: true
  )
end

cookbook_file "#{node['docroot']}/index.html" do
  source 'index.html'
  action :create
end
- name: "Install Web Service"
  package:
    name: apache2
    state: present

- name: "Start Web Service"
  service:
    name: apache2
    state: started
    enabled: yes

- name: "Copy Content"
  copy:
    src: "{{ role_path }}/files/index.html"
    dest: "{{ docroot }}/index.html"
./chef/cookbooks/ez_mysql

attributes
   └── default.rb
Berksfile
kitchen.docker.yml
kitchen.hyperv.yml
kitchen.vbox.yml
metadata.rb
README.md
recipes
   └── client.rb
   └── database.rb
   └── default.rb
   └── harden.rb
   └── install.rb
   └── service.rb
templates
   └── security.cnf.erb
test -> ../../../inspec/ez_mysql/test
TEST STRUCTURE

```
./inpec/ez_mysql
 └── test
     └── integration
         └── default
             ├── ansible.cfg
             ├── conform_test.rb
             ├── contract_test.rb
             └── default.yml
                 └── security_test.rb
```

*Shared Test for ANY change configuration, e.g. CAPS (Chef, Ansible, Puppet, Salt Stack), or other automation*
# Setup (tell kitchen to use docker)
```bash
export KITCHEN_YAML=kitchen.docker.yml
```

# Create Environment
```bash
kitchen create
```

# (optional) Login into environment
```bash
kitchen login
```

# Converge to Desired State
```bash
kitchen converge
```

# Verify Environment
```bash
kitchen verify
```
WORKED FINE?

WORKED FINE IN DEV

OPS PROBLEM NOW
include_recipe 'ez_mysql::install'
include_recipe 'ez_mysql::service'
# include_recipe 'ez_mysql::database'
include_recipe 'ez_mysql::harden'
THE STEPS

include_recipe 'ez_mysql::install'
include_recipe 'ez_mysql::service'
include_recipe 'ez_mysql::database'
include_recipe 'ez_mysql::harden'
WHEN CODE WORKS

How I feel

When my code works
 ROLE STRUCTURE

```text
./ansible/roles/ez_mysql
  ├── defaults
  │   └── main.yml
  │
  ├── handlers
  │   └── main.yml
  │
  ├── kitchen.docker.yml
  │
  ├── kitchen.hyperv.yml
  │
  ├── kitchen.vbox.yml
  │
  ├── meta
  │   └── main.yml
  │
  ├── tasks
  │   └── client.yml
  │
  │   └── database.yml
  │
  │   └── harden.yml
  │
  │   └── install.yml
  │
  │   └── main.yml
  │
  │   └── service.yml
  │
  ├── templates
  │   └── security.cnf.j2
  │
  └── test -> ../../../inspec/ez_mysql/test
```
TEST STRUCTURE

Shared Test for ANY change configuration, e.g. CAPS (Chef, Ansible, Puppet, Salt Stack), or other automation
# Setup (tell kitchen to use docker)
export KITCHEN_YAML=kitchen.docker.yml

# Create Environment
kitchen create

# (optional) Login into environment
kitchen login

# Converge to Desired State
kitchen converge

# Verify Environment
kitchen verify
WORKED FINE?

Say "WORKS ON MY MACHINE" one more time

I DARE YOU
THE STEPS

- include: install.yml
- include: config.yml
- include: service.yml
# - include: harden.yml
THE STEPS

- include: install.yml
- include: config.yml
- include: service.yml
- include: harden.yml
WHEN CODE WORKS

THAT MOMENT

WHEN YOUR CODE WORKS
EXTRA STUFF
(TBA)

(secret)
FINAL NOTES
(for reference)
MAC OS HOST
- TestKitchen (Virtualbox) / Chef
- TestKitchen (Virtualbox) / Ansible
- TestKitchen (Docker) / Chef
- TestKitchen (Docker) / Ansible

WINDOWS HOST
- TestKitchen (HyperV) / Chef
- TestKitchen (HyperV) / Ansible
- TestKitchen (Docker) / Chef
- TestKitchen (Docker) / Ansible

LINUX GUEST
- Mac/Vagrant (Virtualbox) / TestKitchen (Docker) / Chef
- Mac/Vagrant (Virtualbox) / TestKitchen (Docker) / Ansible
- Win/Vagrant (HyperV) / TestKitchen (Docker) / Chef
- Win/Vagrant (HyperV) / TestKitchen (Docker) / Ansible
FURTHER READING

TEST KITCHEN RESOURCES

- Test Kitchen: https://kitchen.ci/
- Ansible Provisioner: https://github.com/neillturner/kitchen-ansible
- Vagrant Driver: https://github.com/test-kitchen/kitchen-vagrant
- Docker Driver: https://github.com/test-kitchen/kitchen-docker
- InSpec Verifier: https://github.com/inspec/kitchen-inspec
- GCP Driver: https://github.com/test-kitchen/kitchen-google
- AWS Driver: https://github.com/test-kitchen/kitchen-ec2
- Kubernetes Driver: https://github.com/coderanger/kitchen-kubernetes

VAGRANT
- HashiCorp Vagrant: https://www.vagrantup.com/

DOCKER
- Docker: https://www.docker.com/
FURTHER READING

INSPEC

- InSpec: https://www.inspec.io/
- inspec-gcp: https://github.com/inspec/inspec-gcp
- inspec-gcp blog: https://blog.chef.io/2018/06/19/inspec-gcp-deep-dive/

KUBERNETES

- Sonobuoy: https://github.com/heptio/sonobuoy

TERRAFORM

- Terratest: https://github.com/gruntwork-io/terratest
- kitchen-terraform: https://github.com/newcontext-oss/kitchen-terraform
THE END