Chef

Module-4

Testing

Test Driven Development

1. **Develop - Test - Refactor**
2. **Automated Tests**
3. **Part of CI/CD**
    3.1. Continuous Integration
    3.2. Continuous Development
4. **Test Harness**
    4.1. Test Environment
    4.2. Test Suites
    4.3. Test Cases
    4.4. Matchers
    4.5. Assertions

TDD in Chef

1. **Test the cookbooks before applying**
2. **Test Harness**
    2.1. Machines and Platforms
        2.1.1. Virtual Machines
            2.1.1.1. Virtual Box
            2.1.1.2. Vagrant
            2.1.1.3. Dockers
            2.1.1.4. EC2
3. **Test Suites, Cases, Matchers, Assertions**
    3.1. InSpec
4. **Test Environment**
    4.1. Test Kitchen

Setup

1. **Install ChefDK or CheckWorkstation**
    1.1. To develop and run recipes and tests
    1.2. Comes with Kitchen and InSpec
    1.3. From https://downloads.chef.io/chefdk
    1.4. From https://downloads.chef.io/chef-workstation/
    1.5. `$ chef –version`
2. **Install Virtual Box**
    2.1. To create virtual machines with desired platforms
    2.2. https://www.virtualbox.org/wiki/Downloads
    2.3. `$ VBoxManage --version`
3. **Install Vagrant**
    3.1. To manage virtual boxes
    3.2. https://www.vagrantup.com/downloads.html
    3.3. `$ vagrant --version`

Cookbook

1. **Generate a cookbook**
    1.1. `$chef generate cookbook mytests`
2. **Develop the recipe**
    2.1. `mytests/recipes/default.rb`
    2.2. `package "git"`
3. **What to test?**
    3.1. **Verify if git is installed on the target machine, up on applying the recipe**
4. **The Test File**
    4.1. `mytests/test/integration/default/default_test.rb`

Cookbook

1. **Generate a cookbook**
   1.1. `$chef generate cookbook mytests`
2. **Develop the recipe**
   2.1. `mytests/recipes/default.rb`
   2.2. `package "git"`
3. **What to test?**
   3.1. Verify if git is installed on the target machine, up on applying the recipe

Test Case

1. **The Test File**

   1.1. `mytests/test/integration/default/default_test.rb`

2. **The Test Case**

```
2.1. describe package('git') do
2.2.   it { should be_installed }
2.3. end
```

Test Configuration

1. **The YML configuration: mytests/kitchen.yml**

```
driver:
  name: vagrant
provisioner:
  name: chef_zero
verifier:
  name: inspec
platforms:
  - name: ubuntu-16.04
suites:
  - name: default
    run_list:
      - recipe[git_cookbook::default]
    verifier:
      inspec_tests:
        - test/integration/default
    attributes:
```

Test Process

1. **List the virtual machines in the kitchen**
    1.1. `kitchen list`
2. **Create a virtual machine for the kitchen**
    2.1. name of the machine: suite-platform
        2.1.1. `kitchen create default-ubuntu-1604`
3. **Run the recipe**
    3.1. `kitchen converge`
4. **Check the results manually**
    4.1. `kitchen login`
5. **Verify**
    5.1. `kitchen verify`
6. **Destroy the virtual boxes**
    6.1. `kitchen destroy default-ubuntu-1604`

Test Process

1. **Test it like TDD**
    1.1. **Creates virtual boxes**
    1.2. **Applies the recipes**
    1.3. **Runs the tests**
    1.4. **Gives the results**
    1.5. **Destroy the virtual boxes**
2. **All in one**
    2.1. `kitchen test`

Multiple Tests

```
suites:
  - name: default
    run_list:
      - recipe[mytests::default]
    verifier:
      inspec_tests:
        - test/integration/default
    attributes:
- name: server
  run_list:
    - recipe[mytests::apache]
  verifier:
    inspec_tests:
      - test/integration/apache
  attributes:
```

Multiple Platforms

```
platforms:
  - name: ubuntu-16.04
  - name: centos-7
```

Selective Platforms

```
platforms:
  - name: ubuntu-16.04
  - name: centos-7
suites:
- name: server
  run_list:
    - recipe[mytests::server]
  verifier:
    inspec_tests:
      - test/integration/server
  attributes:
  excludes:
    - centos-7
```

InSpec

1. **Example - 1**

```
describe car(owner: 'abc') do
    it { should exist }
    its('number') { should cmp 'ka03123' }
    it { should be_sedan }
     it { should_not have_check_engine_light_on }
end
```

2. **Example - 2**

```
describe cars.where(color: /^b/) do
    it { should exist }
    its('manufacturers') { should include 'abc' }
    its('count') { should be >= 10 }
end
```

InSpec

1. **Select a Resource**
    1.1. Resource Name
    1.2. Resource Parameter
    1.3. Resource Parameter Value
    1.4. Resource Filter
2. **Test the Resource**
    2.1. it and its
    2.2. Resource Specific Matcher
    2.3. Universal Matcher
    2.4. Properties

Illustrations

```
describe directory("/etc/testbook") do
  it { should exist }
end

describe file("/etc/testbook/log.sh") do
  it { should exist }
  its('content') { should match /echo "hello"/ }
end
```

Illustrations

```
describe package("nginx") do
  it { should be_installed }
end

control "01" do
  impact 0.7
  title "Verify nginx service"
  desc "Ensures nginx service is up and running"
  describe service("nginx") do
    it { should be_enabled }
    it { should be_installed }
    it { should be_running }
  end
end

web_user = "www-data"
web_user = "nginx" if os[:family] == "centos"

describe user(web_user) do
  it { should exist }
end
```

Illustrations

```
describe package('apache2') do
  it { should be_installed }
end
describe service('apache2') do
  it { should be_installed }
  it { should be_enabled }
  it { should be_running }
end
describe port(80) do
  it { should be_listening }
end
describe http('localhost') do
  its('status') { should eq 200 }
  its('headers.Content-Type') { should include 'text/html' }
  its('body') { should include 'Hello World!' }
end
```

Thank You