

Implementing Chef Testing Framework with Test Kitchen and InSpec

Table of Contents

- [Description](#)
- [Problem Statement](#)
- [Prerequisites](#)
- [Implementation Steps](#)
 - [Step-1: Install Test Kitchen and InSpec](#)
 - [Step-2: Configure Test Kitchen](#)
 - [Step-3: Write InSpec Tests for the Cookbook](#)
 - [Step-4: Run Tests with Test Kitchen](#)
- [References](#)

Description

Chef's **Test Kitchen** and **InSpec** are used to test and verify cookbooks in isolated environments, helping ensure that code functions as intended before it's deployed in production. **Test Kitchen** manages testing environments and runs configurations, while **InSpec** is used to write tests that validate infrastructure settings.

Problem Statement

Testing Chef cookbooks in real environments can be time-consuming and error-prone. With Test Kitchen and InSpec, you can:

- **Isolate tests** in separate environments.
- **Automate verification** of expected configurations and services.
- **Ensure code consistency** across multiple systems.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-05) is required before proceeding with Lab Guide-06.

Software Required

- **Chef Workstation:** To create, test, and manage cookbooks.
- **Virtualization Software:** Vagrant or Docker for local testing (recommended).

Hardware Requirement

- **Chef Workstation:** 2 GB RAM, 2 CPU cores

Implementation Steps

Step-1: Install Test Kitchen and InSpec

1. Ensure Chef Workstation is Installed

2. Verify Installation:

- Check that Test Kitchen and InSpec are available by running:

```
kitchen --version
inspec version
```

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo>inspec --version
5.22.55
```

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo>_
```

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo>kitchen --version
Test Kitchen version 3.6.8
```

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo>_
```

Step-2: Configure Test Kitchen

1. Initialize Test Kitchen:

- Navigate to your cookbook directory and initialize Test Kitchen with:

```
kitchen init
```

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>kitchen init
  conflict  kitchen.yml
xchef-repo/cookbooks/my_webserver/kitchen.yml? (enter "h" for help) [Ynaqdhm] Y
  force    kitchen.yml
  identical cheignore
C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>_
```

- This command creates a **kitchen.yml** file in your cookbook directory, which defines how Test Kitchen will set up and test your environment.

2. Edit **kitchen.yml**:

- Open the **kitchen.yml** file and configure it to use the appropriate driver and platform.
- For example, to use Docker as a test platform:

```
---
driver:
  name: vagrant

platforms:
  - name: ubuntu-20.04

suites:
  - name: default
    run_list:
      - recipe[my_webserver::default]
    verifier:
      name: inspec
    attributes:
```



```
! kitchen.yml M x
my_webserver > ! kitchen.yml
1  ---
2  driver:
3    name: vagrant
4
5  platforms:
6    - name: ubuntu-20.04
7
8  suites:
9    - name: default
10      run_list:
11        - recipe[my_webserver::default]
12      verifier:
13        name: inspec
14      attributes:
15
```

- Here, we set **vagrant** as the driver and **Ubuntu 20.04** as the testing platform, applying the `my_webserver::default` recipe.

Step-3: Write InSpec Tests for the Cookbook

1. Create InSpec Test Directory:

- Inside the `test/integration/default` folder of your cookbook, create a test file named `default_test.rb`.

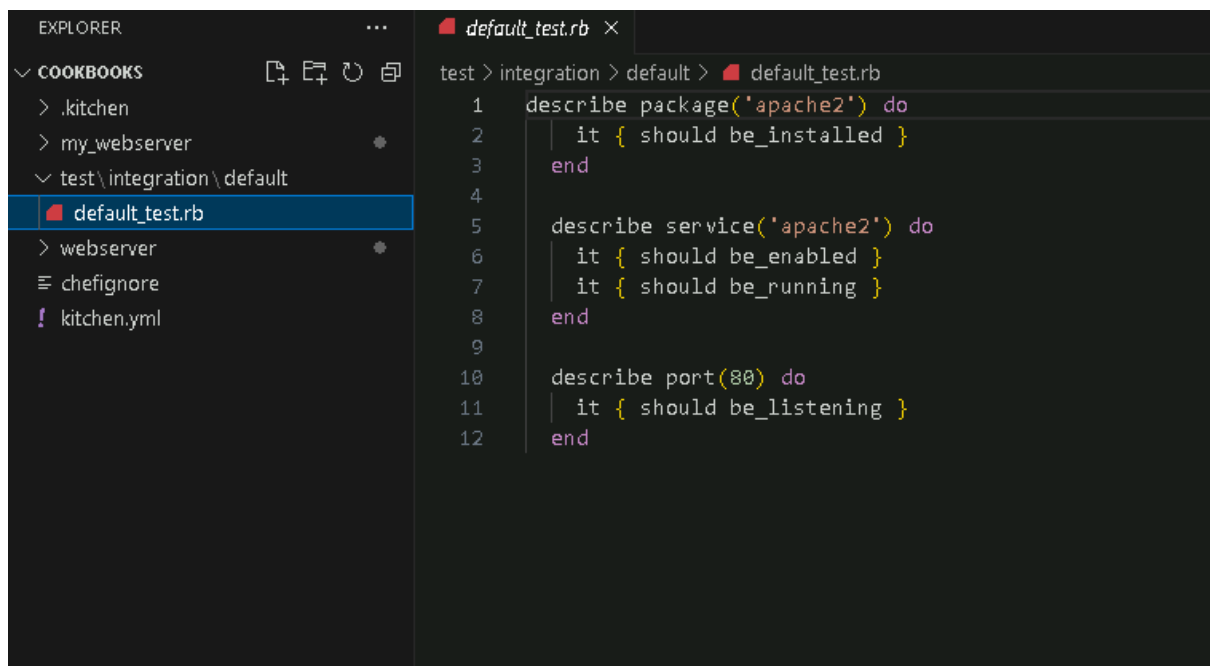
2. Write InSpec Tests:

- Open `default_test.rb` and write tests to validate your configurations. For example, if your cookbook installs Apache, you could check for the Apache package and its running status:

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

describe service('apache2') do
  it { should be_enabled }
  it { should be_running }
end

describe port(80) do
  it { should be_listening }
end
```



- These tests check if **Apache** is installed, enabled, and running, and if **port 80** is open.

Step-4: Run Tests with Test Kitchen

1. Run Test Kitchen Commands:

- **Create the Test Instance:**

```
kitchen create
```

```

C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>kitchen create
-----> Starting Test Kitchen (v3.6.0)
-----> Creating <default-ubuntu-2004>...
  Bringing machine 'default' up with 'virtualbox' provider...
    => default: Importing base box 'bento/ubuntu-20.04'...
    => default: Matching MAC address for NAT networking...
    => default: Checking if box 'bento/ubuntu-20.04' version '202407.23.0' is up to date...
    => default: Setting the name of the VM: kitchen-my_webserver-default-ubuntu-2004-2af67fe1-0c2c-4707-a7e0-808344506882
    => default: Clearing any previously set network interfaces...
    => default: Preparing network interfaces based on configuration...
      default: Adapter 1: nat
    => default: Forwarding ports...
      default: 22 (guest) => 2222 (host) (adapter 1)
    => default: Running 'pre-boot' VM customizations...
    => default: Booting VM...
    => default: Waiting for machine to boot. This may take a few minutes...
      default: SSH address: 127.0.0.1:2222
      default: SSH username: vagrant
      default: SSH auth method: private key
      default: Warning: Connection reset. Retrying...
      default: Warning: Connection aborted. Retrying...
      default:
      default: Vagrant insecure key detected. Vagrant will automatically replace
      default: this with a newly generated keypair for better security.
      default:
      default: Inserting generated public key within guest...
      default: Removing insecure key from the guest if it's present...
      default: Key inserted! Disconnecting and reconnecting using new SSH key...
    => default: Machine booted and ready!
    => default: Checking for guest additions in VM...
    => default: Setting hostname...
    => default: Mounting shared folders...
      default: C:/Users/Administrator/.kitchen/cache => /tmp/omnibus/cache
    => default: Machine not provisioned because '--no-provision' is specified.
    [SSH] Established
    Vagrant instance <default-ubuntu-2004> created.
    Finished creating <default-ubuntu-2004> (2m16.25s).
-----> Test Kitchen is finished. (2m21.66s)

C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>_

```

- This command starts the container or virtual machine based on your `.kitchen.yml` configuration.

2. Apply the Cookbook and Run Tests:

- **Converge** the cookbook to apply its configurations:

kitchen converge

```

C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>kitchen converge
-----> Starting Test Kitchen (v3.6.0)
-----> Converging <default-ubuntu-2004>...
  Preparing files for transfer
    Policy lock file doesn't exist, running 'C:\opscode\chef-workstation\bin\chef-cli.BAT install' for Policyfile C:/Users/Administrator/Downloads/chef-starter/chef-repo/cookbooks/my_webserver/Policyfile.rb...
    Building policy my_webserver
    Expanded run list: recipe[my_webserver::default]
    Caching Cookbooks...
    Installing my_webserver >= 0.0.0 from path

    Lockfile written to C:/Users/Administrator/Downloads/chef-starter/chef-repo/cookbooks/my_webserver/Policyfile.lock.json
    Policy revision id: 59356be96436bc2b83f361b4efbe5237870a520d83398bfec17d02be17b89c6
    Updating policy lock using 'C:\opscode\chef-workstation\bin\chef-cli.BAT update'
    Building policy my_webserver
    Expanded run list: recipe[my_webserver::default]
    Caching Cookbooks...
    Installing my_webserver >= 0.0.0 from path

    Lockfile written to C:/Users/Administrator/Downloads/chef-starter/chef-repo/cookbooks/my_webserver/Policyfile.lock.json
    Policy revision id: 796f1f638380e343c529216a1714104d3d24aa4fc0288e647ae7232399f2779
    Preparing dna.json
    Exporting cookbook dependencies from Policyfile C:/Users/ADMINI~1/AppData/Local/Temp/default-ubuntu-2004-sandbox-20241112-4816-e2j4on using 'C:\opscode\chef-workstation\bin\chef-cli.BAT export'...
    Exported policy 'my_webserver' to C:/Users/ADMINI~1/AppData/Local/Temp/default-ubuntu-2004-sandbox-20241112-4816-e2j4on

    To converge this system with the exported policy, run:
      cd C:/Users/ADMINI~1/AppData/Local/Temp/default-ubuntu-2004-sandbox-20241112-4816-e2j4on
      chef-client -z
    Removing non-cookbook files before transfer
    Preparing validation.pem
    Preparing client.rb
-----> Installing Chef Install only if missing package
  Downloading https://omnitruck.chef.io/install.sh to file /tmp/install.sh
  Trying wget...
  Download complete.
  ubuntu 20.04 x86_64
  Getting information for chef stable for ubuntu...
  downloading https://omnitruck.chef.io/stable/chef/metadata?v=&p=ubuntu&pv=20.04&m=x86_64
  to file /tmp/install.sh.i2649/metadata.txt
  trying wget...
  sha1      b27b98228f3fd5c0e9a09b04b069cb96d478f22b
  sha256    b5824d89f8d35f560d7b1015d967f4728d89e05529c286319b13f55ab27249777d3

```

```
-  
-  
-  
-      </div>  
-    </div>  
-    <div class="validator">  
-      </div>  
-    </body>  
-</html>  
-  
+<h1>Welcome to Labguide-05: Creating and Using Chef Cookbooks for Complex Configurations!</h1>  
  
Running handlers:  
Running handlers complete  
Infra Phase complete, 3/5 resources updated in 59 seconds  
Downloading files from <default-ubuntu-2004>  
Finished converging <default-ubuntu-2004> (2m40.30s).  
----- Test Kitchen is finished. (2m45.77s)  
  
C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>
```

- **Verify** the cookbook by running the tests:

kitchen verify

```
C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>kitchen verify
----> Starting Test Kitchen (v3.6.0)
----- Verifying (default-ubuntu-2004)...
    Loaded tests from {:path=>"C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver\test\integration\default"}

Profile: tests from {:path=>"C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver\test\integration\default"} (tests from {:path=>"C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver\test\integration\default"})
Version: (not specified)
Target:  ssh://vagrant@127.0.0.1:2222
Target ID: 4f3f3e7d-b197-50e7-bf6d-48e0c0029a7a

  System Package apache2
    [PASS] is expected to be installed
  Service apache2
    [PASS] is expected to be enabled
    [PASS] is expected to be running

Test Summary: 3 successful, 0 failures, 0 skipped
    Finished verifying (default-ubuntu-2004) (0m12.45s).
-----> Test Kitchen is finished. (0m10.25s)

C:\Users\Administrator\Downloads\chef-starter\chef-repo\cookbooks\my_webserver>
```

- This command runs the InSpec tests you wrote in `default_test.rb`. If successful, you'll see output indicating that the tests passed.

References

- Chef Test Kitchen Documentation: https://docs.chef.io/test_kitchen/
- Chef InSpec Documentation: <https://docs.chef.io/inspec/>
- Testing Infrastructure with Chef: <https://learn.chef.io/>