Implementing Chef Testing Framework with Test Kitchen and InSpec

Table of Contents

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

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

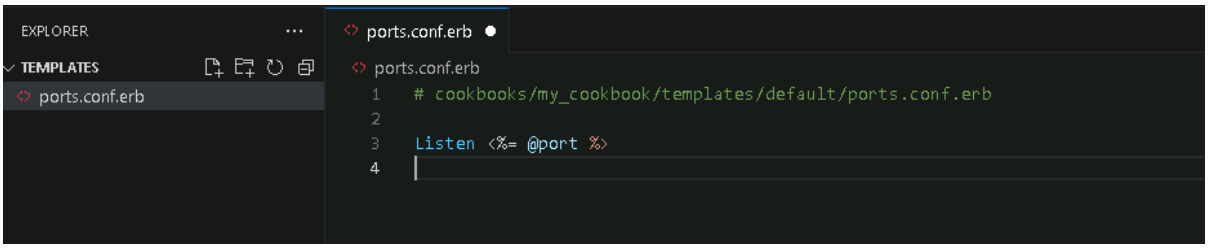
Step-1: Install Test Kitchen and InSpec

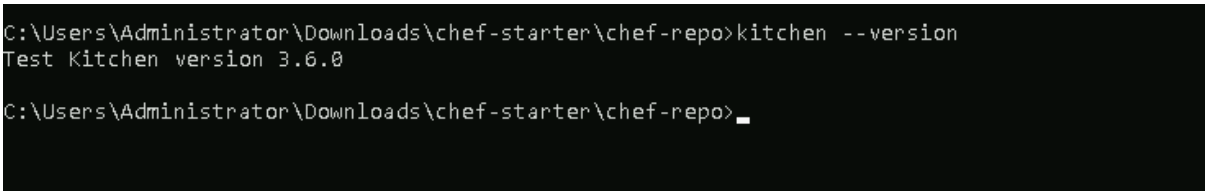
1. **Ensure Chef Workstation is Installed**

2. **Verify Installation**:

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

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



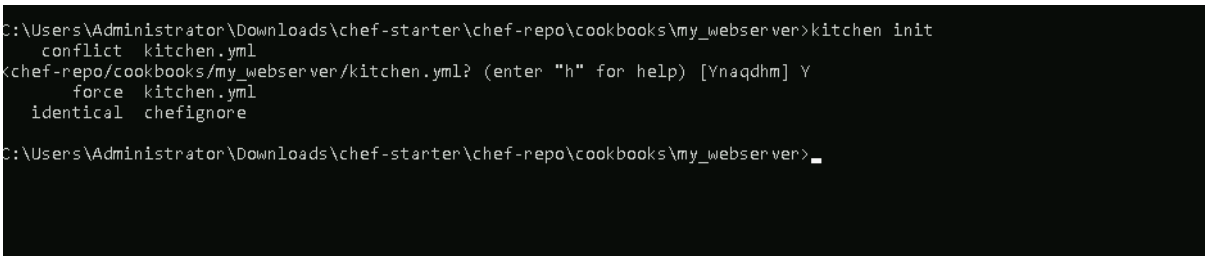


Step-2: Configure Test Kitchen

1. **Initialize Test Kitchen**:

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

```
kitchen init
```



   - 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`**:

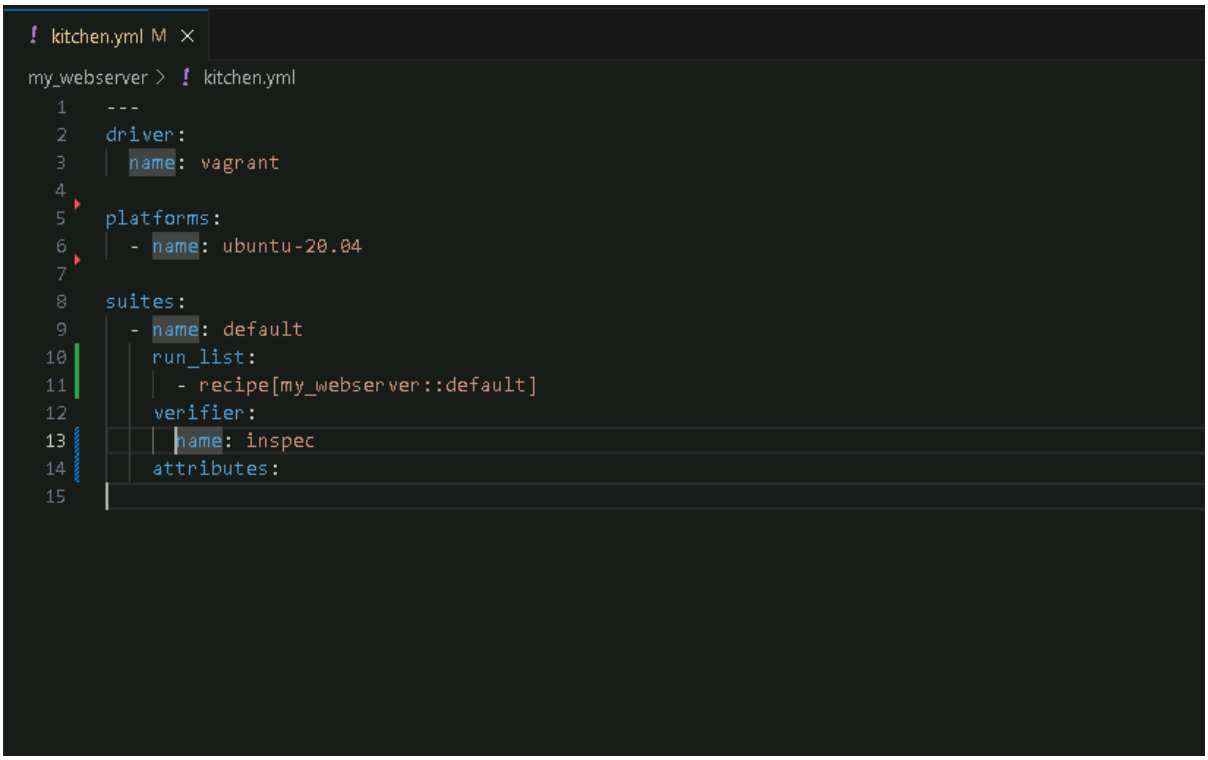
   - Open the `.kitchen.yml` file and configure it to use the appropriate driver and platform.

   - 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:
```



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

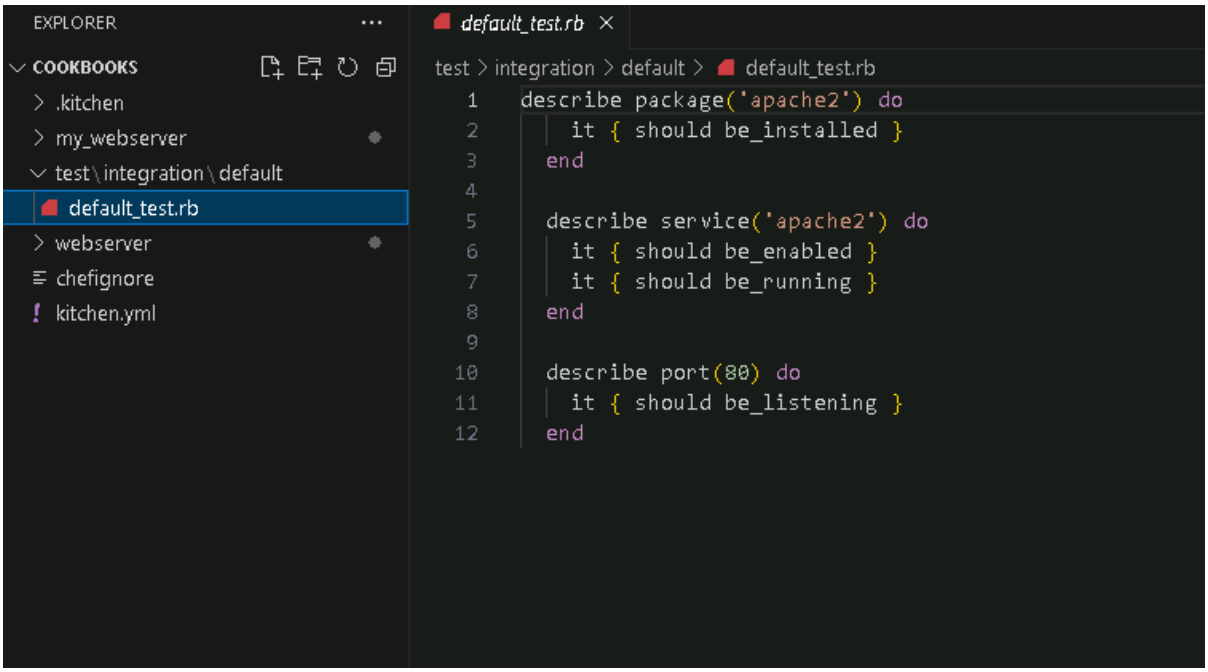
Step-3: Write InSpec Tests for the Cookbook

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: 59356be96436bc2b83f361b4efbe5237870ae520d83398bfec17d02be17b89c6
       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: 790f1f638380e343c529216a1714104dad24aa4fc0288e647ae7232399ff2f79
       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-work
station\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.12649/metadata.txt
       trying wget...
       sha1    b27b98228f3fd5c0e9a09b04b869cb96d478f22b
       sha256  b502d80f8d358560d7b1015d967f478d89e05529c386310b13fb5ab2749777d3
```

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

```
kitchen verify
```

- 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/