

## Objectives:

Using Test Kitchen, Chef's integration testing framework, we'll look for an expected file state using test recipes in a virtual testing environment.

To better understand Test Kitchen, let's create a chef repo and a base cookbook using the chef command line tool.

### 1. Create a Chef repo

2. Navigate to the root directory by executing the command :

```
$cd ~
```

3. Print your working directory (show what directory you are in)

```
ubuntu@ip-172-31-62-237:~$ pwd  
/home/ubuntu
```

4. To create a base repo, execute the command below:

```
$chef generate repo chef-repo
```

**Note:** It's the first use of a chef command, so you may be prompted to accept a licence first, if so type 'yes'

```
ubuntu@ip-172-31-62-237:~$ chef generate repo chef-repo
+-----+
          Chef License Acceptance

Before you can continue, 3 product licenses
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

Licenses that need accepting:
  * Chef Workstation
  * Chef Infra Client
  * Chef InSpec

Do you accept the 3 product licenses (yes/no)?
```

5. Once accepted, you will see following message :

```
Persisting 3 product licenses...
✓ 3 product licenses persisted.

+-----+
Generating Chef Infra repo chef-repo
- Ensuring correct Chef Infra repo file content

Your new Chef Infra repo is ready! Type `cd chef-repo` to enter it.
```

6. View the directory structure by executing the tree command.

```
ubuntu@ip-172-31-62-237:~$ tree chef-repo
```

```
ubuntu@ip-172-31-62-237:~$ tree chef-repo
chef-repo
├── LICENSE
├── README.md
├── chefignore
├── cookbooks
│   ├── README.md
│   └── example
│       ├── README.md
│       ├── attributes
│       │   └── default.rb
│       ├── metadata.rb
│       ├── recipes
│       │   └── default.rb
├── data_bags
│   ├── README.md
│   └── example
│       └── example_item.json
└── policyfiles
    └── README.md

7 directories, 11 files
```

NOTE: While you can manually create a directory called chef-repo, using the command `chef generate repo chef-repo` creates a standardized directory structure that is very useful when using Chef.

Create a Chef cookbook named apache:

7. Change to the 'chef-repo' directory and use the 'chef' command to create a new cookbook named 'apache':

```
ubuntu@ip-172-31-62-237:~$ cd chef-repo/
```

```
ubuntu@ip-172-31-62-237:~/chef-repo$ chef generate cookbook cookbooks/apache
```

8. The following output will be displayed:

```
ubuntu@ip-172-31-62-237:~/chef-repo$ chef generate cookbook cookbooks/apache
Generating cookbook apache
- Ensuring correct cookbook content

Your cookbook is ready. Type `cd cookbooks/apache` to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.
Type `delivery local --help` to see a full list of local testing commands.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb
```

9. The contents of the cookbook can now be examined in a file tree format that Chef has generated using the “tree” command.

```
ubuntu@ip-172-31-62-237:~/chef-repo$ tree cookbooks/apache/
```

```
ubuntu@ip-172-31-62-237:~/chef-repo$ tree cookbooks/apache/
cookbooks/apache/
├── CHANGELOG.md
├── LICENSE
├── Policyfile.rb
├── README.md
├── cheffignore
├── kitchen.yml
├── metadata.rb
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   └── unit
│       └── recipes
│           └── default_spec.rb
└── test
    ├── integration
    │   └── default
    │       └── default_test.rb
    └── test_helper.rb

7 directories, 11 files
```

10. Integration test files are created in the following directory:

`~/chef-repo/cookbooks/<cookbook_name>/test/integration/default/default_test.rb`

11. In our example of the apache cookbook, the file path will be:

```
/home/ubuntu/chef-repo/cookbooks/apache/test/integration/default $pwd  
/home/ubuntu/chef-repo/cookbooks/apache/test/integration/default
```

12. **kitchen.yml** is the name of the Test Kitchen config file located at the root of the cookbook directory.

```
ubuntu@ip-172-31-62-237: ~/chef-repo/cookbooks/apache$ ls -la  
total 56  
drwxrwxr-x 6 ubuntu ubuntu 4096 Apr 16 19:54 .  
drwxrwxr-x 4 ubuntu ubuntu 4096 Apr 16 19:54 ..  
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 16 19:54 .delivery  
-rw-rw-r-- 1 ubuntu ubuntu 211 Apr 16 19:54 .gitignore  
-rw-rw-r-- 1 ubuntu ubuntu 150 Apr 16 19:54 CHANGELOG.md  
-rw-rw-r-- 1 ubuntu ubuntu 70 Apr 16 19:54 LICENSE  
-rw-rw-r-- 1 ubuntu ubuntu 503 Apr 16 19:54 Policyfile.rb  
-rw-rw-r-- 1 ubuntu ubuntu 54 Apr 16 19:54 README.md  
-rw-rw-r-- 1 ubuntu ubuntu 1252 Apr 16 19:54 chefignore  
-rw-rw-r-- 1 ubuntu ubuntu 744 Apr 16 19:54 kitchen.yml  
-rw-rw-r-- 1 ubuntu ubuntu 676 Apr 16 19:54 metadata.rb  
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 16 19:54 recipes  
drwxrwxr-x 3 ubuntu ubuntu 4096 Apr 16 19:54 spec  
drwxrwxr-x 3 ubuntu ubuntu 4096 Apr 16 19:54 test
```

13. In our examples we will be using the Dokken driver and provisioner. The Dokken provisioner will use docker to execute test cases on behalf of Test Kitchen.

**14. Update the apache/kitchen.yml file as follows:**

```
---
driver:
  name: dokken

provisioner:
  name: dokken

transport:
  name: dokken

verifier:
  name: inspec

platforms:
  - name: centos-7
    driver:
      image: dokken/centos-7
      privileged: true
      pid_one_command: /usr/lib/systemd/systemd
      volumes:
        - /sys/fs/cgroup:/sys/fs/cgroup:ro # required by systemd

suites:
  - name: default
    sudo: true
    verifier:
      inspec_tests:
        - test/integration/default
    attributes:
```

**Note:** yml files need exact indentation. Only use spaces, not tabs.  
**Replicate the file precisely as seen above**

## Kitchen Commands

15. List the kitchens that will be created:

```
$kitchen list
```

This shows the list of Kitchen instances that will be created. An instance comprises a single platform (listed in platforms) with a set of testing criteria (listed in suites).

```
---
driver:
  name: dokken

provisioner:
  name: dokken

transport:
  name: dokken

verifier:
  name: inspec

platforms:
  - name: centos-7
    driver:
      image: dokken/centos-7
      privileged: true
      pid_one_command: /usr/lib/systemd/systemd
      volumes:
        - /sys/fs/cgroup:/sys/fs/cgroup:ro # required by systemd

suites:
  - name: default
    sudo: true
    verifier:
      inspec_tests:
        - test/integration/default
    attributes:
```

A red rounded rectangle labeled "Instance" has two red arrows pointing from it to the "platforms:" and "suites:" sections of the code block. The "platforms:" section is also highlighted with a red box.

16. The expected output from our **\$ kitchen list** command:

```
ubuntu@ip-172-31-62-237:~/chef-repo/cookbooks/apache$ kitchen list
Instance      Driver  Provisioner  Verifier  Transport  Last Action  Last Error
default-centos-7 Dokken  Dokken      Inspec   Dokken     <Not Created> <None>
```

17. Create the environments that were in the list:

```
$kitchen create
```

18. When the **\$kitchen create** command is executed, a kitchen instance will be created for every instance listed in the table above.

If you were to execute

```
$kitchen create default-centos-69
```

this would create only one instance using the centos-69 platform

a. A successful creation of a kitchen instance will output the following:

```
ubuntu@ip-172-31-62-237:~/chef-repo/cookbooks/apache$ kitchen create
-----> Starting Test Kitchen (v2.10.0)
-----> Creating <default-centos-7>...
    Creating container chef-latest
    Creating kitchen sandbox at /home/ubuntu/.dokken/kitchen_sandbox/fa3b5de57d-default-centos-7
    Creating verifier sandbox at /home/ubuntu/.dokken/verifier_sandbox/fa3b5de57d-default-centos-7
    Building work image..
    Creating container fa3b5de57d-default-centos-7
    Finished creating <default-centos-7> (0m11.58s).
-----> Test Kitchen is finished. (0m12.44s)
```

19. Execute the **\$kitchen list** command again. Under the last action column of the kitchen instances table, it should be displaying Created:

```
ubuntu@ip-172-31-62-237:~/chef-repo/cookbooks/apache$ kitchen list
Instance      Driver  Provisioner  Verifier  Transport  Last Action  Last Error
default-centos-7 Dokken  Dokken      Inspec   Dokken     Created      <None>
```



20. Install Chef in the kitchen, copy the cookbook(s) over to the kitchen and converge the cookbook(s)

```
$kitchen converge
```

Converging a kitchen instance will :

- Create the instance if it has not already been created.
- Install Chef
- Apply the cookbook that is at the root of the working directory to that kitchen instance.

```
/home/ubuntu/chef-repo/cookbooks/apache
```

- Expected output from the kitchen converge command:

```
Starting Chef Infra Client, version 16.13.16
Patents: https://www.chef.io/patents
[2021-04-16T21:37:09+00:00] ERROR: shard_seed: Failed to get dmi property serial_number: is dmidecode installed?
Creating a new client identity for default-centos-7 using the validator key.
Using policy 'apache' at revision '26ccd4b64b945e2d27d00b7b513dae4886b915a06658034ba39e07013e8ffda4'
resolving cookbooks for run list: ["apache::default@0.1.0 (073e06f)"]
Synchronizing Cookbooks:
- apache (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 0 resources

Running handlers:
Running handlers complete
Chef Infra Client finished, 0/0 resources updated in 21 seconds
  Finished converging <default-centos-7> (0m29.71s).
-----> Test Kitchen is finished. (0m30.60s)
```

21. Execute your tests against the state of the instance. Execute

```
$kitchen verify
```

22. The `$kitchen verify` command does the following :

- i. Create an instance, if needed
- ii. Converge the instance, if needed
- iii. Execute a collection of defined tests against the instance.
- iv. Execute the `$kitchen verify` command.

```
User root
  ↵
Port 80
  ↵

Test Summary: 0 successful, 0 failures, 2 skipped
             Finished verifying <default-centos-7> (0m3.01s).
-----> Test Kitchen is finished. (0m3.94s)
```

**Note:** the kitchen **skips** two test cases. These are the default cases that you can see in

apache/test/integration/default/default\_test.rb

Note they both have the word "skip" in the test (because they are just sample tests), causing Test Kitchen to ignore them, hence the "2 skipped" notation in the test results

Notify your instructor that you are done with the lab

**END OF LAB**