# Chef - Test Kitchen with CentOS on Microsoft Azure

Introduction

Imagine your node represented a piece of your production infrastructure. How could you detect the failure before the change reached production?

Test Kitchen runs your infrastructure code in an isolated environment that resembles your production environment. With Test Kitchen, you continue to write your Chef code from your workstation, but instead of uploading your code to the Chef server and applying it to a node, Test Kitchen applies your code to a temporary environment, such as a cloud or container instance.

Local development shifts the development workflow to one that focuses on correcting mistakes earlier in the development process, before you apply your changes to production systems. In this workflow, you:

- Write a small amount of Chef code that expresses some new policy or feature.
- Verify your change on local test instances.
- Upload your code to the Chef server.
- Run chef-client to apply the updated policy on your node.
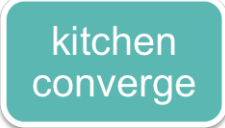- Repeat the process.
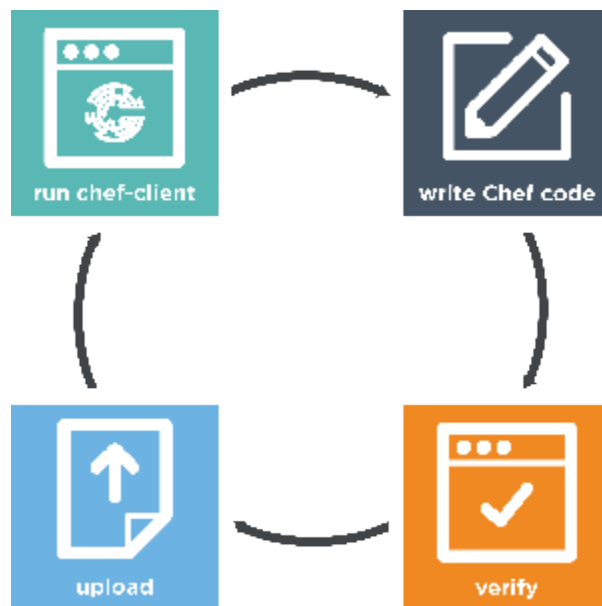
Test Kitchen work-flow

kitchen create	In this step, Test Kitchen creates an instance of your virtual environment, for example, a CentOS virtual machine.

kitchen converge	In this step, Test Kitchen applies your cookbook to the virtual environment.
kitchen login	In this step, Test Kitchen creates an SSH session into your virtual environment.
verify	In this step, you manually verify that your virtual environment is configured as you expect.
kitchen destroy	In this step, Test Kitchen shuts down and destroys your virtual environment.

Azure Test Environment

Test Kitchen uses drivers, or plugins, to work with cloud and virtualization platforms. Some drivers come with the Chef DK. For Azure, we need to install the kitchen-azurerm driver on your workstation manually and create a configuration file on your workstation that authenticates access to Azure resources.

Azure resources: Regions, Linux virtual machine with the Azure portal, and SSH public and private key pair for Linux VMs in Azure.

```
$ chef gem install kitchen-azurerm
$ sudo yum install python-devel
$ sudo curl -L https://aka.ms/InstallAzureCli | bash
$ az --version
$ az login
$ az app list

# Create the service principal
$ az ad sp create-for-rbac --name "my-chef-kitchen-test" --password
"strong password"
{
  "appId": "8409b0f7-xxxx-xxxx-881e-72dc5f38f9ca",
  "displayName": "my-chef-kitchen-test",
  "name": "http://my-chef-kitchen-test",
  "password": "XXXXXX",
  "tenant": "5bbd0f75-16cb-xxxx-xxxx-8a9645889f05"
}

# Get information about the service principal.
$ az ad sp show --id 8409b0f7-xxxx-xxxx-881e-72dc5f38f9ca
$ az app list

# Make sure you save the file with UTF-8 encoding!
$ vi ~/.azure/credentials
[abcd1234-YOUR-SUBSCRIPTION-ID-HERE-abcdef123456]
client_id = "8409b0f7-xxxx-xxxx-881e-72dc5f38f9ca"              ---
application id from the application step
client_secret = "your-password"                                ---
password you supplied in the command line.
tenant_id = "9c117323-YOUR-GUID-HERE-9ee430723ba3"

$ chmod 600 ~/.azure/credentials
```

Test Kitchen configuration

Let's set up a cookbook to run on the test instance.

```
$ mkdir learn-chef/cookbooks
$ cd learn-chef/cookbooks
$ git clone https://github.com/learn-chef/learn_chef_httpd.git
$ cd learn_chef_httpd
```

The **chef generate cookbook** command creates a cookbook and a file named **.kitchen.yml** in the root directory of your cookbook. This file defines what's needed to run Test Kitchen. To create CentOS instances on Azure, start by modifying .kitchen.yml.

```
$ vi .kitchen.yml
---
driver:
  name: azurerm

driver_config:
  subscription_id: 12345678-YOUR-GUID-HERE-123456789ABC
  location: "East US"
  machine_size: "Standard_D1"

transport:
  ssh_key: /root/.ssh/learn-chef.pem
--- create this key first!

provisioner:
  name: chef_zero

verifier:
  name: inspec

platforms:
  - name: centos-7.2
    driver_config:
      image_urn: OpenLogic:CentOS:7.2n:7.2.20160629
      vm_name: centos-vm
      vm_tags:
        ostype: linux
        distro: centos

suites:
  - name: default
    run_list:
      - recipe[learn_chef_httpd::default]
    verifier:
      inspec_tests:
        - test/smoke/default
    attributes:
```

Parameters (find more):

- **provisioner** specifies how to run Chef. We use chef_zero because it enables you to mimic a Chef server environment on your local machine. This allows us to work with node attributes.
- **transport** specifies how to execute commands remotely on the test instance. When working with cloud instances, this is where you specify your SSH key pair.
- **verifier** specifies which application to use when running automated tests.
- **suites** specifies what we want to apply to the virtual environment. You can have more than one suite. We define just one, named `defau lt`. This is where we provide the run-list, which defines which recipes to run and in the order to run them.

Test Kitchen workflow

Let's provision an Azure virtual machine, deploy the cookbook, and execute the tests.

```
$ cd cookbooks/learn_chef_httpd
$ kitchen list
Instance          Driver   Provisioner  Verifier  Transport  Last
Action     Last Error
default-centos-72  Azurerm  ChefZero     Inspec    Ssh        <Not
Created>  <None>

$ kitchen create
----------------- Create the instance.
$ kitchen list
Instance          Driver   Provisioner  Verifier  Transport  Last
Action  Last Error
default-centos-72  Azurerm  ChefZero     Inspec    Ssh        Created
<None>

$ kitchen converge
----------------- Apply the cookbook to the machine.
...
       Running handlers complete
       Chef Client finished, 4/4 resources updated in 13 seconds
       Finished converging <default-centos-72> (0m33.51s).
-----> Kitchen is finished. (0m34.51s)
$ kitchen list
Instance          Driver   Provisioner  Verifier  Transport  Last
Action  Last Error
default-centos-72  Azurerm  ChefZero     Inspec    Ssh        Converged
<None>

$ kitchen exec -c 'curl localhost'
----------------- Verify the contents of your web server's home page.
$ kitchen destroy
```

References

- https://learn.chef.io/skills/
- https://docs.chef.io/
- https://docs.microsoft.com/en-us/azure/virtual-machines/linux/