Test Driven Development with InSpec on Azure

CHEF

Test Driven Development

1. Define a test set for the unit first
2. Then implement the unit
3. Finally verify that the implementation of the unit makes the tests succeed.
4. Refactor

CHEF

Discuss Requirements

Requirements for a web server

Discuss Requirements

- Install the Apache package, httpd, on Red Hat
- Start and enable the httpd service.
- Serve a custom home page.
- Open port 80 to incoming traffic.

CHEF

Lab

Create Cookbook & First Test

- Generate cookbook with chef
- Write First Test in Inspec

Create Cookbook

```
> chef generate cookbook webserver
```

```
Generating cookbook webserver_test
- Ensuring correct cookbook file content
- Committing cookbook files to git
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content
- Adding delivery configuration to feature branch
- Adding build cookbook to feature branch
- Merging delivery content feature branch to master

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

CHEF

Inspec Test

```
describe '<entity>' do
      it { <expectation> }
end
```

CHEF

Replace Spec File in Cookbook

`~/webserver/test/smoke/default_test.rb`

```
describe package 'httpd' do
  it { should be_installed }
end
```

CHEF

Lab

Setup Testing on Azure

- Install kitchen-azurerm gem
- Create Azure Service Account
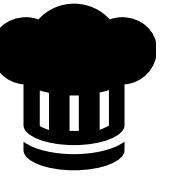- Configure Test Kitchen

CHEF

Install kitchen-azurerm

```
> chef gem install kitchen-azurerm
```
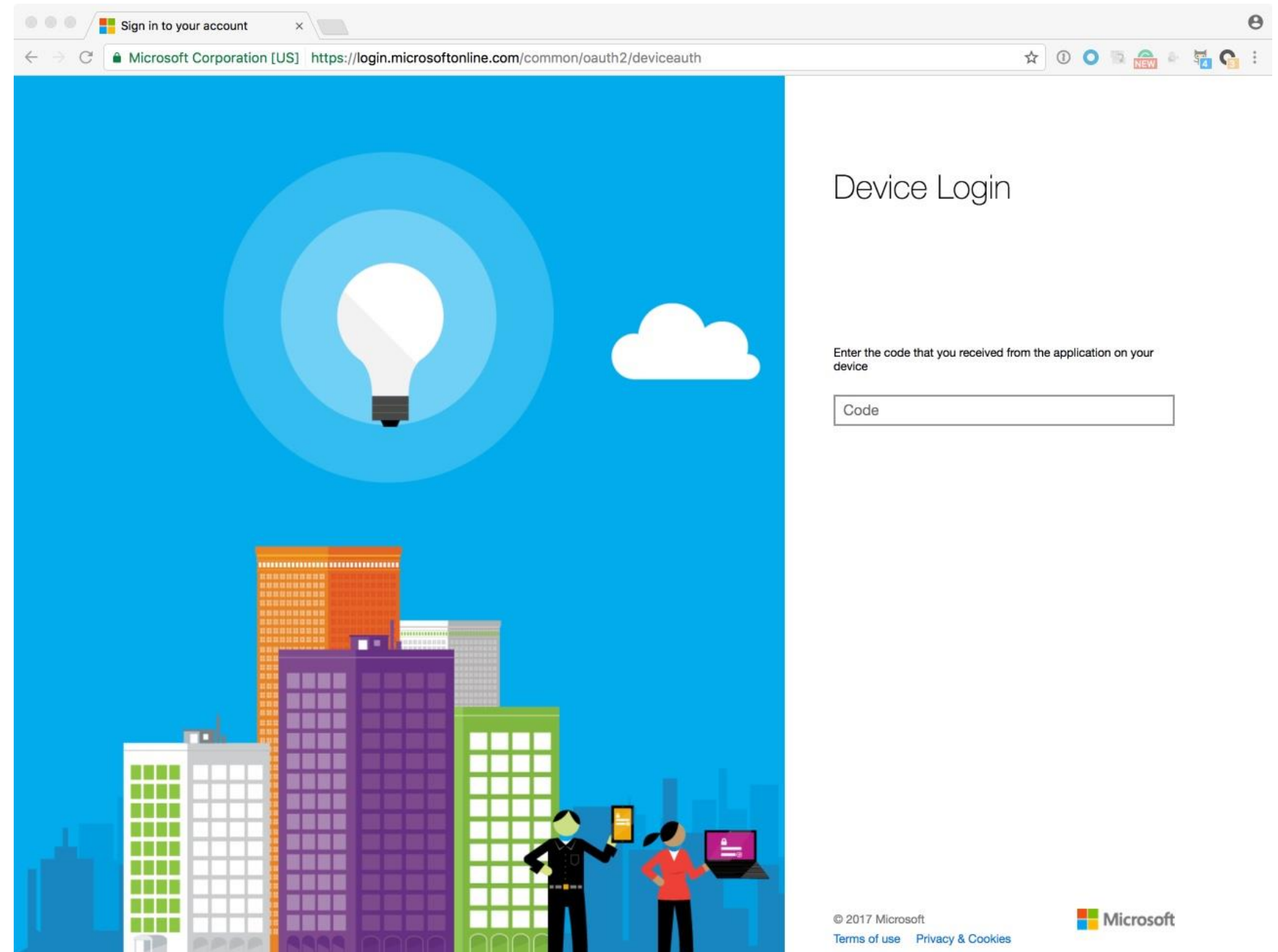
Create Azure Service Account

```
az login
```

Create Azure Service Account

Steps

1. Go to https://aka.ms/devicelogin
2. Enter Code
3. Click Continue
4. Login to Azure
5. Close Browser Window

Create Azure Service Account

```
> az ad sp create-for-rbac --name testkitchen --password "<pass>"
```

```
{
  "appId": "GUID",
  "displayName": "testkitchen",
  "name": "http://testkitchen",
  "password": "<pass>",
  "tenant": "GUID"
}
```

CHEF

Create Azure Service Account

`~/.azure/credentials`

```
[abcd1234-YOUR-SUBSCRIPTION-ID-HERE-abcdef123456]
client_id = "48b9bba3-YOUR-GUID-HERE-90f0b68ce8ba"
client_secret = "your-client-secret-here"
tenant_id = "9c117323-YOUR-GUID-HERE-9ee430723ba3"
```

CHEF

Configure Test Kitchen

`~/webserver/.kitchen.yml`

Content: https://tinyurl.com/kitchenyml

CHEF

Configure Test Kitchen

```
> kitchen list
```

```
Instance            Driver     Provisioner   Verifier   Transport   Last Action      Last Error
default-centos-72   Azurerm    ChefZero      Inspec     Ssh         <Not Created>    <None>
```

CHEF

Configure Test Kitchen

`~/webserver/metadata.rb`

```ruby
name 'webserver'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'all_rights'
description 'Installs/Configures webserver'
long_description 'Installs/Configures webserver'
version '0.1.0'


depends 'line'
```

CHEF

Configure Test Kitchen

`~/webserver/recipes/fixsudo.rb`

```ruby
delete_lines 'remove requiretty from sudoers' do
  path '/etc/sudoers'
  pattern '^.*requiretty'
end
```

http://tinyurl.com/chefsudofix

Run tests and see them fail

```
> kitchen verify
```

```
  System Package

      Ø   httpd should be installed

      expected that `System Package httpd` is installed
Test Summary: 0 successful, 1 failures, 0 skipped
```

CHEF

Just Enough Recipe to Pass Test

~/webserver/recipes/default.rb

```ruby
# Install the Apache package.
package 'httpd'
```

CHEF

Just Enough Recipe to Pass Test

```
> kitchen converge && kitchen verify
```

```
System Package

    ✓   httpd should be installed

Test Summary: 1 successful, 0 failures, 0 skipped

    Finished verifying <default-centos-72> (0m2.94s).
```
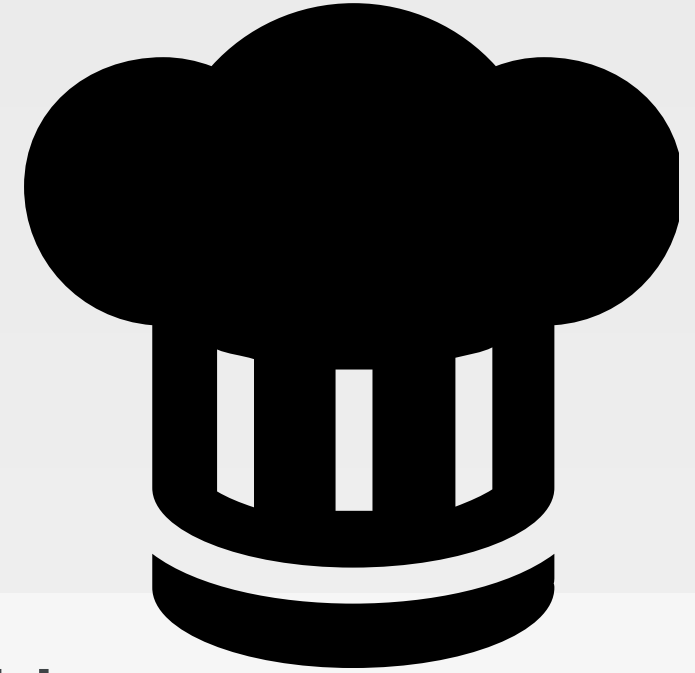
LAB

Remaining Requirements

- ~~Install the Apache package, httpd, on Red Hat~~
- Start and enable the httpd service.
- Serve a custom home page.
- Open port 80 to incoming traffic.

http://inspec.io/docs

CHEF

Remaining Requirements

`~/webserver/test/smoke/default_test.rb`

```ruby
describe package 'httpd' do
  it { should be_installed }
end
describe service 'httpd' do
 it { should be_enabled }
 it { should be_running }
end
describe command 'curl localhost' do
 its('stdout') { should match /hello/ }
end
describe port 80 do
 it { should be_listening }
end
```

CHEF

Remaining Requirements

```
> kitchen verify
```

```
System Package
    ✓   httpd should be installed
Service httpd
    ∅   should be enabled
    expected that `Service httpd` is enabled
    ∅   should be running
    expected that `Service httpd` is running
Command curl
    ∅   localhost stdout should match /hello/
    expected "" to match /hello/
    Diff:
    @@ -1,2 +1,2 @@
    -/hello/
    +""
Port 80
    ∅   should be listening
    expected `Port 80.listening?` to return true, got false
Test Summary: 1 successful, 4 failures, 0 skipped
```

CHEF

Remaining Requirements

```ruby
# Install the Apache package.
package 'httpd'

# Start and enable the httpd service.
service 'httpd' do
  action [:enable, :start]
end

# Serve a custom home page.
file '/var/www/html/index.html' do
  content '<html>
  <body>
    <h1>hello world</h1>
  </body>
</html>'
end
```

CHEF

Remaining Requirements

```
> kitchen converge && kitchen verify
```

```
System Package
    ✓   httpd should be installed
Service httpd
    ✓   should be enabled
    ✓   should be running
Command curl
    ✓   localhost stdout should match /hello/
Port 80
    ✓   should be listening
Test Summary: 5 successful, 0 failures, 0 skipped
Finished verifying <default-centos-72> (0m3.50s).
```

CHEF

Remaining Requirements

```
> kitchen test
```

```
Test Summary: 5 successful, 0 failures, 0 skipped
Finished verifying <default-centos-72> (0m3.50s).
```

CHEFAUTOMATE

Workflow   Visibility   Compliance

CHEF   habitat   INSPEC