Refactoring Cookbooks with Tests

CONCEPT

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**

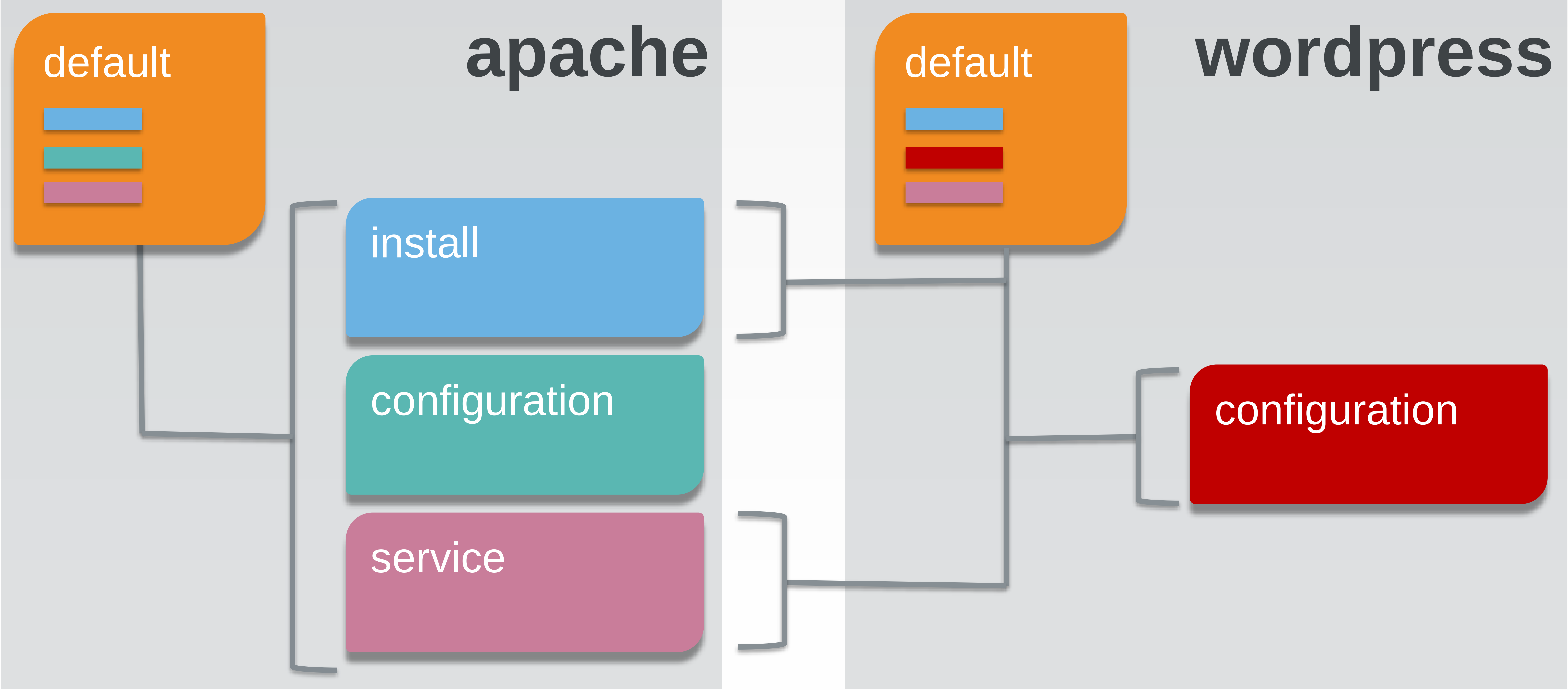
Objectives

After completing this module, you should be able to:

- Refactor a recipe using `include_recipe`
- Use Test Kitchen to validate the code you refactored
- Explain when to use `kitchen converge`, `kitchen verify` and `kitchen test`.

CONCEPT

Modular Cookbook Recipes



CHEF

CONCEPT

include_recipe

A recipe can include one (or more) recipes located in cookbooks by using the **`include_recipe`** method. When a recipe is included, the resources found in that recipe will be inserted (in the same exact order) at the point where the **`include_recipe`** keyword is located.

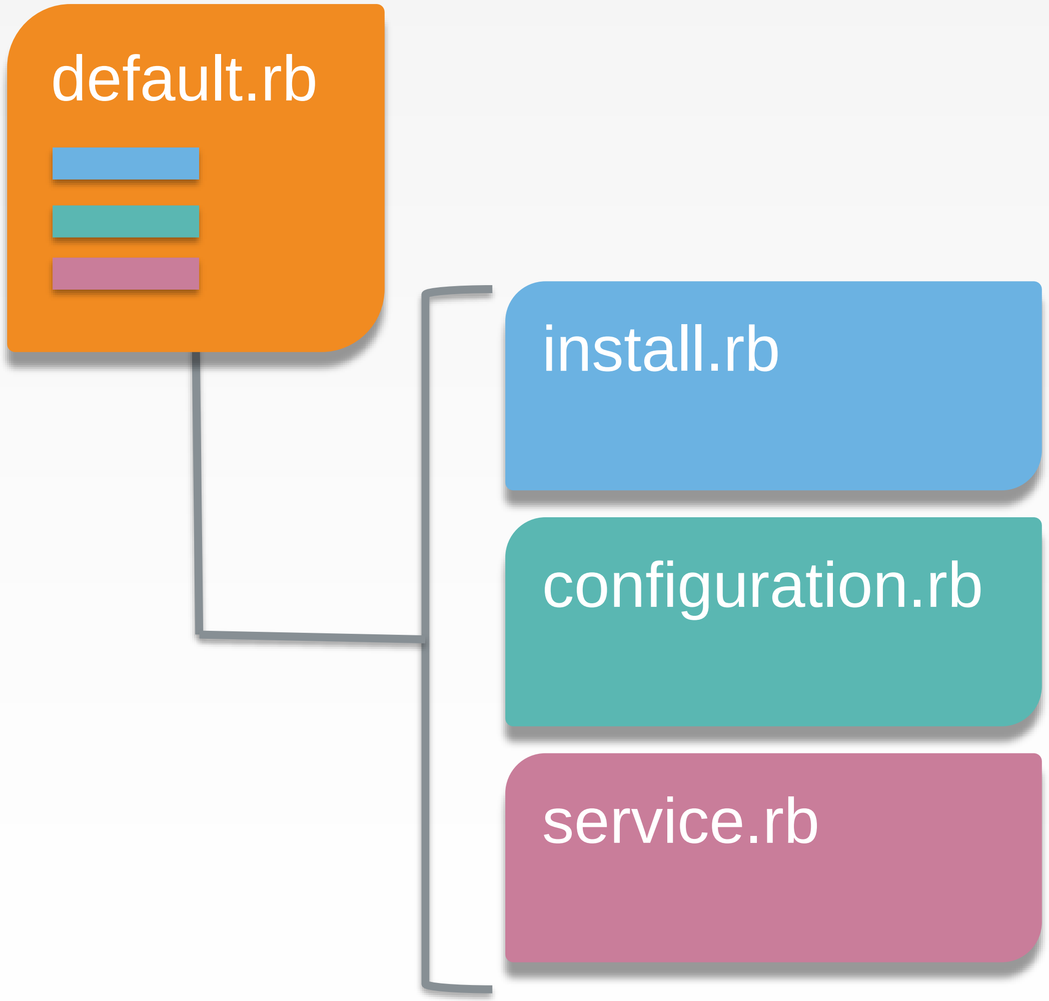
https://docs.chef.io/recipes.html#include-recipes

CONCEPT

Recipe Organization

recipes/default.rb

```
include_recipe 'cookbook::install'
include_recipe 'cookbook::configuration'
include_recipe 'cookbook::service'
```



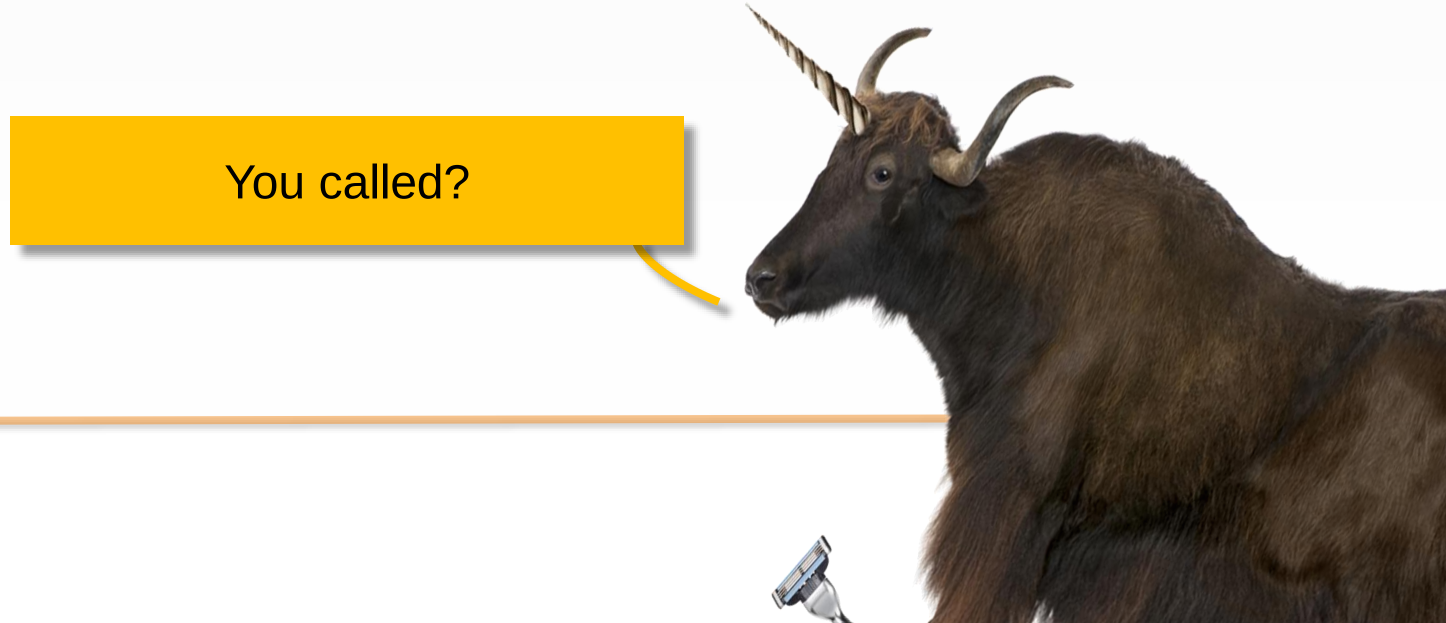
EXERCISE

Refactor to Modular Recipes

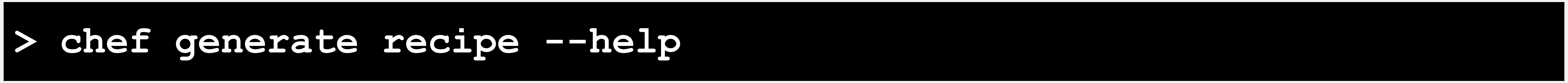
*This is why we **can** have nice things!*

Objective:

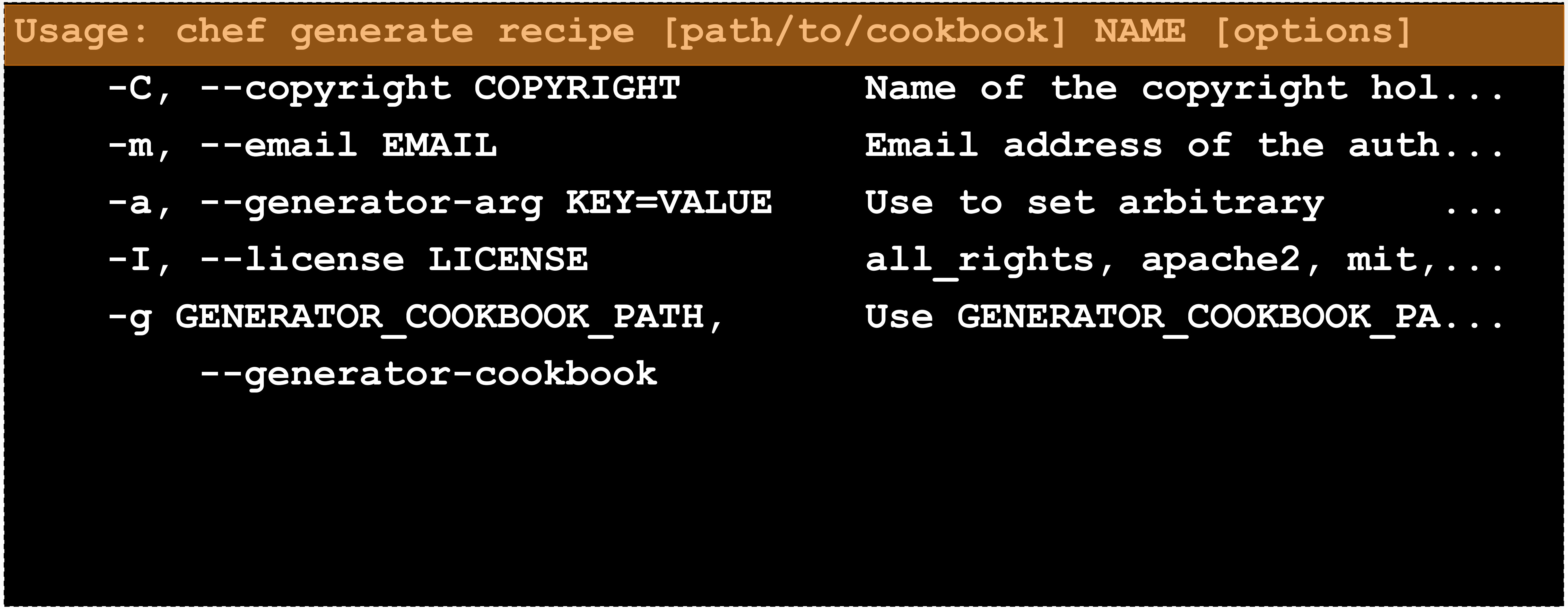
- ❑ Refactor the installation into a separate recipe
- ❑ Converge the cookbook and execute the tests



Ask Chef About Generating a Recipe



```
> chef generate recipe --help
```



```
Usage: chef generate recipe [path/to/cookbook] NAME [options]
    -C, --copyright COPYRIGHT          Name of the copyright hol...
    -m, --email EMAIL                  Email address of the auth...
    -a, --generator-arg KEY=VALUE      Use to set arbitrary     ...
    -I, --license LICENSE              all_rights, apache2, mit,...
    -g GENERATOR_COOKBOOK_PATH,        Use GENERATOR_COOKBOOK_PA...
        --generator-cookbook
```

Generate an Install Recipe

```
> chef generate recipe install
```

```
Recipe: code_generator::recipe
  * directory[/home/chef/apache/spec/unit/recipes] action create
 (up to date)
  * cookbook_file[/home/chef/apache/spec/spec_helper.rb] action
create_if_missing (up to date)
  * template[/home/chef/apache/spec/unit/recipes/install_spec.rb]
action create_if_missing
    - create new file
/home/chef/apache/spec/unit/recipes/install_spec.rb
    - update content in file
/home/chef/apache/spec/unit/recipes/install_spec.rb from none to
187413
```

Removing the Generated Test File

```
> rm test/smoke/default/install_test.rb
```

Write the Install Recipe

```
~/apache/recipes/install.rb
```

```ruby
#
# Cookbook Name:: apache
# Recipe:: install
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
package 'httpd'
```

Remove the Resource from the Default Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome Home!</h1>'
end

service 'httpd' do
  action [:enable, :start]
end
```

Include the Install Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
include_recipe 'apache::install'

file '/var/www/html/index.html' do
  content '<h1>Welcome Home!</h1>'
end

service 'httpd' do
  action [:enable, :start]
end
```

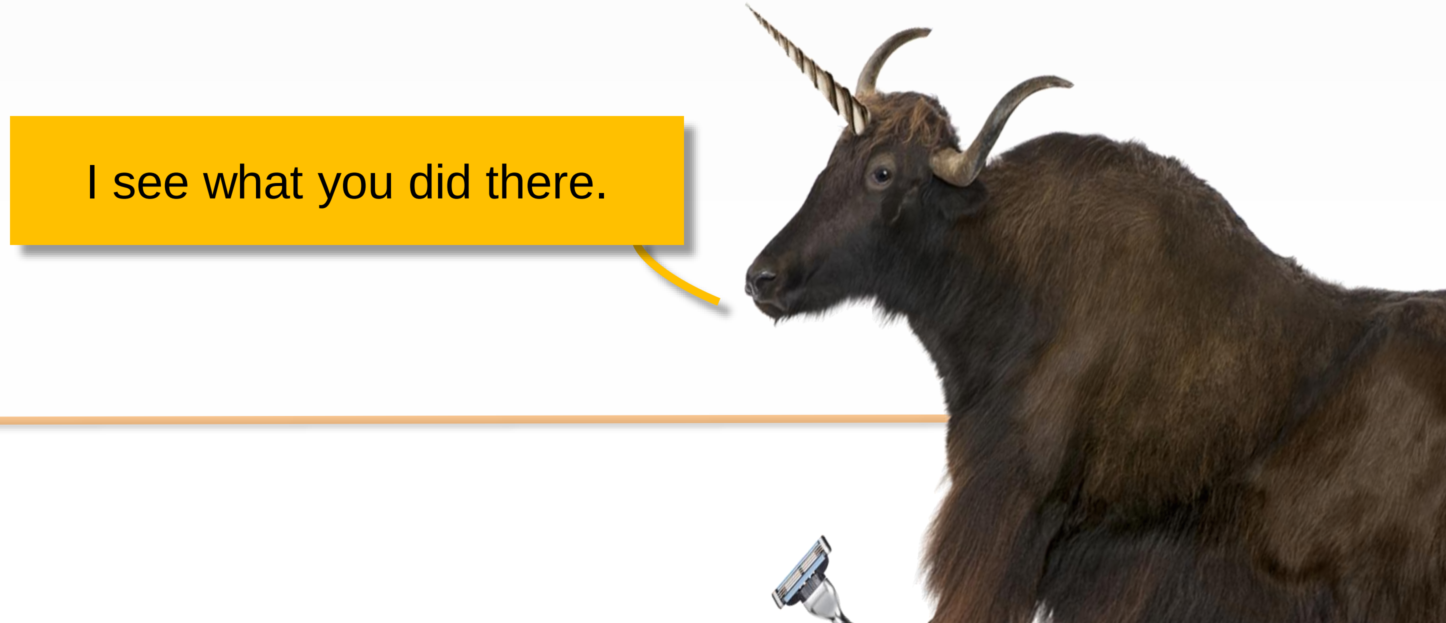
EXERCISE

Refactor to Modular Recipes

*This is why we **can** have nice things!*

Objective:

- ✓ Refactor the installation into a separate recipe
- ❑ Converge the cookbook and execute the tests



Re-Converge the Test Instance

```
> kitchen converge
```

```
----> Starting Kitchen (v1.11.1)
-----> Converging <default-centos-69>...
$$$$$$ Running legacy converge for 'Docker' Driver
       ...
-----> Installing Chef Omnibus (install only if missing)
       Downloading https://www.chef.io/chef/install.sh to file...
       resolving cookbooks for run list: ["apache::default"]
       ...
       Finished converging <default-centos-69> (0m27.64s).
-----> Kitchen is finished. (0m28.58s)
```

Re-Verify the Test Instance

```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <default-centos-69>...
       Use `/home/chef/apache/test/smoke/default` for testing

Target:  ssh://kitchen@localhost:32770

  ✔  Port 80 should be listening
  ✔  Command curl localhost stdout should match /Welcome Home/

Summary: 2 successful, 0 failures, 0 skipped
```

EXERCISE

Refactor to Modular Recipes

*This is why we **can** have nice things!*

Objective:

- ✓ Refactor the installation into a separate recipe
- ✓ Converge the cookbook and execute the tests

Nice shave!

LAB

The Configuration

□ Create a configuration recipe that defines the policy:

```
The file named '/var/www/html/index.html' contains the
content '<h1>Welcome Home!</h1>'
```

□ Delete the automatically generated InSpec test
□ Within the default recipe replace the file resource with an include recipe
□ Converge and verify the test instance to ensure there are no failures

Generate a Service Recipe

```
> chef generate recipe configuration
```

```
Recipe: code_generator::recipe
  * directory[/home/chef/apache/spec/unit/recipes] action create
 (up to date)
  * cookbook_file[/home/chef/apache/spec/spec_helper.rb] action
create_if_missing (up to date)
  *
template[/home/chef/apache/spec/unit/recipes/configuration_spec.rb
] action create_if_missing
    - create new file
/home/chef/apache/spec/unit/recipes/configuration_spec.rb
    - update content in file
/home/chef/apache/spec/unit/recipes/configuration_spec.rb from
none
```

Remove the Generated Test File

```
> rm test/smoke/default/configuration_test.rb
```

Write the Configuration Recipe

```
~/apache/recipes/configuration.rb
```

```
#
# Cookbook Name:: apache
# Recipe:: configuration
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
file '/var/www/html/index.html' do
  content '<h1>Welcome Home!</h1>'
end
```

Remove the Resource from the Default Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
include_recipe 'apache::install'

file '/var/www/html/index.html' do
  content '<h1>Welcome Home!</h1>'
end

service 'httpd' do
  action [:enable, :start]
end
```

Include the Configuration Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
include_recipe 'apache::install'
include_recipe 'apache::configuration'

service 'httpd' do
  action [:enable, :start]
end
```

Re-Converge the Test Instance

```
> kitchen converge
```

```
----> Starting Kitchen (v1.11.1)
-----> Converging <default-centos-69>...
$$$$$$ Running legacy converge for 'Docker' Driver
       ...
-----> Installing Chef Omnibus (install only if missing)
       Downloading https://www.chef.io/chef/install.sh to file...
       resolving cookbooks for run list: ["apache::default"]
       ...
       Finished converging <default-centos-69> (0m27.64s).
-----> Kitchen is finished. (0m28.58s)
```

Re-Verify the Test Instance

```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <default-centos-69>...
       Use `/home/chef/apache/test/smoke/default` for testing

Target:  ssh://kitchen@localhost:32770

  ✔  Port 80 should be listening
  ✔  Command curl localhost stdout should match /Welcome Home/

Summary: 2 successful, 0 failures, 0 skipped
```

LAB

The Configuration

- ✓ Create a configuration recipe that defines the policy:

```
The file named '/var/www/html/index.html' contains the
content '<h1>Welcome Home!</h1>'
```

- ✓ Delete the automatically generated InSpec test
- ✓ Within the default recipe replace the file resource with an include recipe
- ✓ Converge and verify the test instance to ensure there are no failures

LAB

The Service

□ Create a service recipe that defines the policy:

```
The service named 'httpd' is started and enabled
```

□ Delete the automatically generated InSpec test

□ Within the default recipe replace the service resource with an include recipe

□ Converge and verify the test instance to ensure there are no failures

One last time!

Generate a Service Recipe

```
> chef generate recipe service
```

```
Recipe: code_generator::recipe
  * directory[/home/chef/apache/spec/unit/recipes] action create
(up to date)
  * cookbook_file[/home/chef/apache/spec/spec_helper.rb] action
create_if_missing (up to date)
  * template[/home/chef/apache/spec/unit/recipes/service_spec.rb]
action create_if_missing
    - create new file
/home/chef/apache/spec/unit/recipes/service_spec.rb
    - update content in file
/home/chef/apache/spec/unit/recipes/service_spec.rb from none to
1f669c
    (diff output suppressed by config)
```

Remove the Generated Test File

```
> rm test/smoke/default/service_test.rb
```

Write the Services Recipe

```
~/apache/recipes/service.rb
```

```
#
# Cookbook Name:: apache
# Recipe:: service
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
service 'httpd' do
  action [:enable, :start]
end
```

Remove the Resource from the Default Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
include_recipe 'apache::install'
include_recipe 'apache::configuration'

service 'httpd' do
  action [:enable, :start]
end
```

Remove the Resource from the Default Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
include_recipe 'apache::install'
include_recipe 'apache::configuration'
include_recipe 'apache::service'
```

Re-Converge the Test Instance

```
> kitchen converge
```

```
----> Starting Kitchen (v1.11.1)
-----> Converging <default-centos-69>...
$$$$$$ Running legacy converge for 'Docker' Driver
       ...
-----> Installing Chef Omnibus (install only if missing)
       Downloading https://www.chef.io/chef/install.sh to file...
       resolving cookbooks for run list: ["apache::default"]
       ...
       Finished converging <default-centos-69> (0m27.64s).
-----> Kitchen is finished. (0m28.58s)
```

Re-Verify the Test Instance

```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <default-centos-69>...
       Use `/home/chef/apache/test/smoke/default` for testing

Target:  ssh://kitchen@localhost:32770

  ✔  Port 80 should be listening
  ✔  Command curl localhost stdout should match /Welcome Home/

Summary: 2 successful, 0 failures, 0 skipped
```

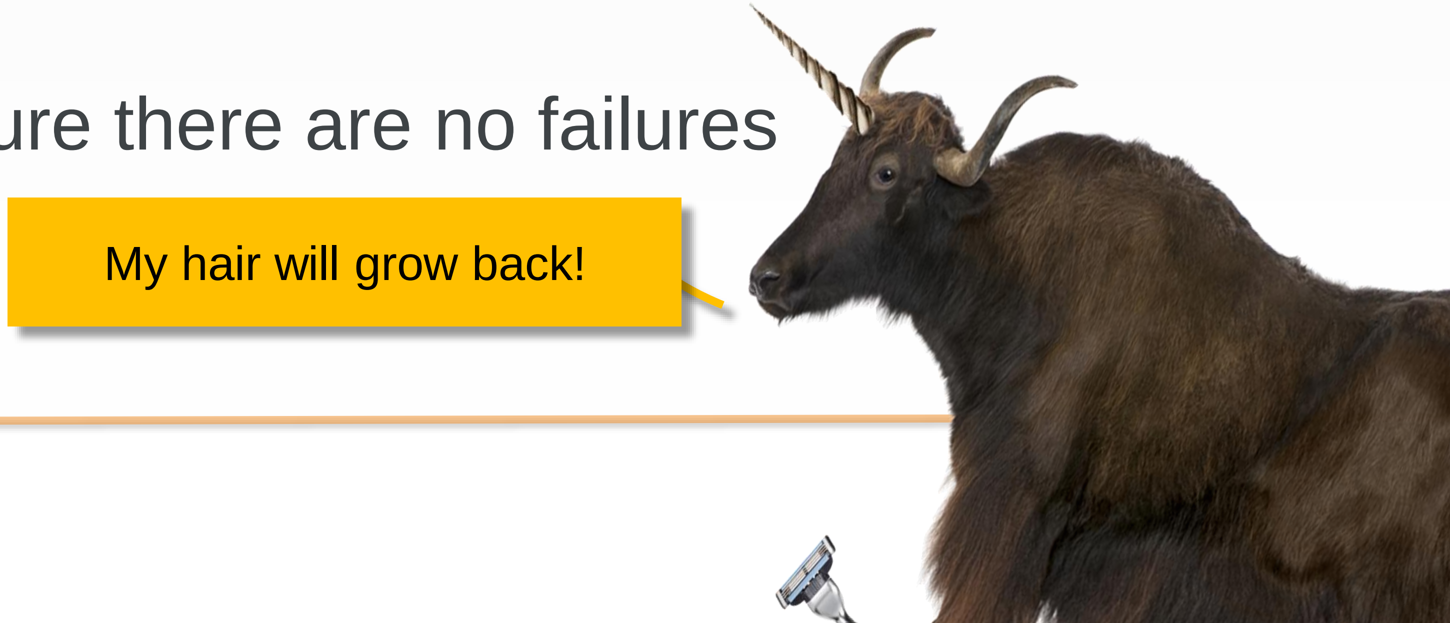
LAB

The Service

- ✓ Create a service recipe that defines the policy:

```
The service named 'httpd' is started and enabled.
```

- ✓ Delete the automatically generated InSpec test
- ✓ Within the default recipe replace the service resource with an include recipe
- ✓ Converge and verify the test instance to ensure there are no failures



DISCUSSION

Do Our Tests Really Work?

What if we removed code from within the recipes and ran the tests?

CONCEPT

Heckling Your Code

Mutation testing is used to design new software tests and evaluate the quality of existing software tests. Mutation testing involves modifying a program in small ways.

EXERCISE

Heckle That Code

It could be a game show. Maybe on Twitch?

Objective:

□ Remove / Comment source code

□ Converge the cookbook and execute the tests

Comment Out Key Code Within the Default Recipe

```
~/apache/recipes/default.rb
```

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.
# include_recipe 'apache::install'
include_recipe 'apache::configuration
include_recipe 'apache::service'
```

EXERCISE

Heckle That Code

It could be a game show. Maybe on Twitch?

Objective:

✓ Remove / Comment source code

❑ Converge the cookbook and execute the tests

Re-Converge the Test Instance

```
> kitchen converge
```

```
-----> Converging <default-centos-69>...
       Synchronizing Cookbooks:
         - apache (0.1.0)
       Compiling Cookbooks...
       Converging 3 resources
       Recipe: apache::configuration
        (up to date)
       Recipe: apache::service
        (up to date)
         * service[apache] action enable (up to date)
```

Re-Verify the Test Instance

```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <default-centos-69>...
       Use `/home/chef/apache/test/smoke/default` for testing

Target:  ssh://kitchen@localhost:32770


  ✔  Port 80 should be listening
  ✔  Command curl localhost stdout should match /Welcome Home/


Summary: 2 successful, 0 failures, 0 skipped
```

CONCEPT

Kitchen Converge & Verify

Running converge or verify will create a new instance the first time it is run. The same instance is used for each additional converge or verify.

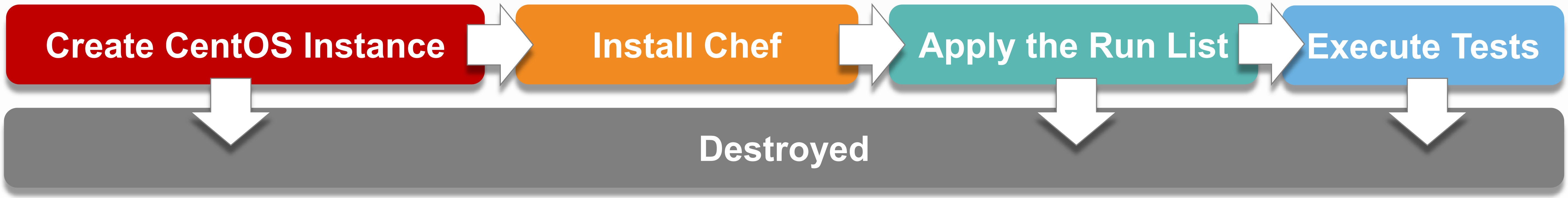
The test instance policy changed, but no resource explicitly removed or uninstalled the resources defined in the install recipe.

CONCEPT

Kitchen Destroy

```
$ kitchen destroy [INSTANCE|REGEXP|all]
```

Destroys one or more instances.


Create CentOS Instance

Install Chef

Apply the Run List

Execute Tests

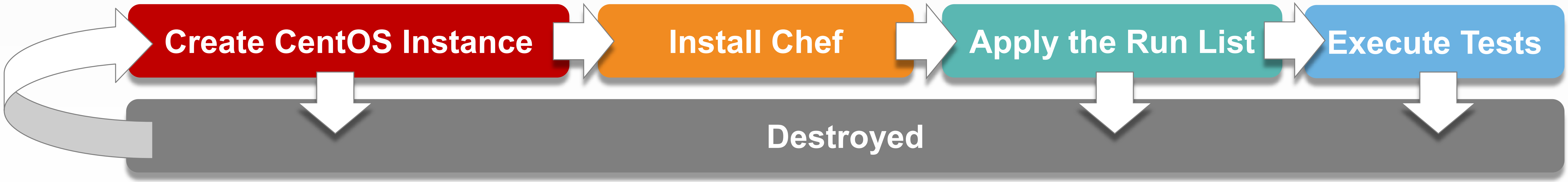
Destroyed

CONCEPT

Kitchen Test

```
$ kitchen test [INSTANCE|REGEXP|all]
```

Destroys (for clean-up), creates, converges, verifies and then destroys one or more instances.



CONCEPT

Kitchen Test

Destroying the instance ensures that the policy is being applied to a new instance.

The test instance is re-created and then the updated policy is applied to the new instance. The new policy is incomplete causing an error.

Test the Cookbook Against a New Instance

```
> kitchen test
```

```
----> Starting Kitchen (v1.19.2)
-----> Cleaning up any prior instances of <default-centos-69>
-----> Destroying <default-centos-69>...
       ...
       [2017-08-29T20:29:16+00:00] FATAL:
Chef::Exceptions::ChildConvergeError: Chef run process exited
unsuccessfully (exit code 1)
>>>>>> ------Exception-------
>>>>>> Class: Kitchen::ActionFailed
>>>>>> Message: 1 actions failed.
>>>>>>     Converge failed on instance <default-centos-69>.
```

Converge & Verify

Faster execution time

Running converge twice will ensure your policy applies without error to **existing instances**

Test

Slower execution time

Running test will ensure your policy applies without error to any **new instances**

EXERCISE

Heckle That Code

It could be a game show. Maybe on Twitch?

Objective:

- ✓ Remove / Comment source code
- ✓ Converge the cookbook and execute the tests

Discussion

What is happening when running `kitchen test`?

What types of bugs would `kitchen converge` & `kitchen verify` find when running?

What is the difference between `kitchen test` and running both `kitchen converge` & `kitchen verify` together?

How long do each of these approaches take?

DISCUSSION

Q&A

What questions can we answer for you?

Morning

Introduction

Why Write Tests? Why is that Hard?

Writing a Test First

Refactoring Cookbooks with Tests

Afternoon

Faster Feedback with Unit Testing

Testing Resources in Recipes

Refactoring to Attributes

Refactoring to Multiple Platforms

CHEF™