

Cookbooks

Organizing Recipes

Slide 1 of 57

Objectives

After completing this module, you should be able to:

- Modify a recipe
- Use version control
- Generate a Chef cookbook
- Define a Chef recipe that sets up a web server

Questions You May Have

Questions You May Have

- Thinking about the **workstation** recipe, could we do something like that for a web server?

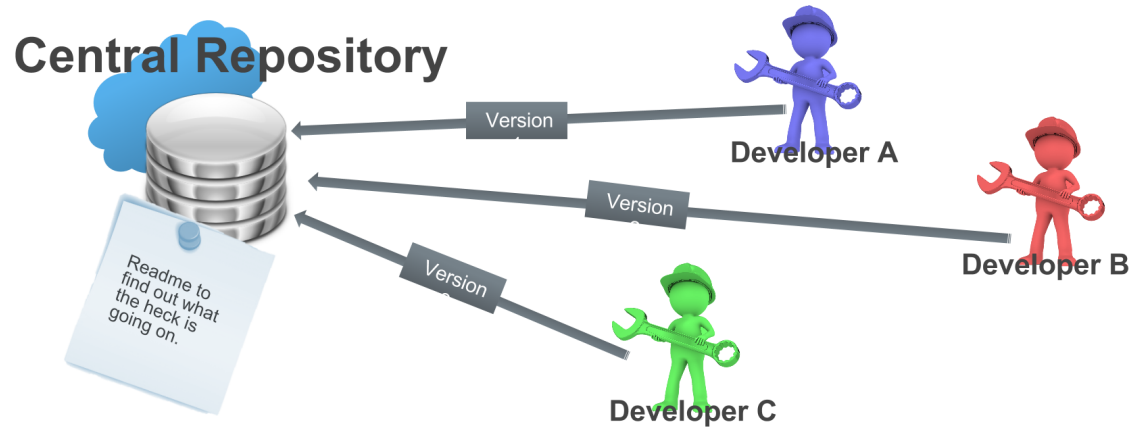
Questions You May Have

- Thinking about the **workstation** recipe, could we do something like that for a web server?
- Is there a way to package up recipes you create with a version number (and maybe a README)?

Questions You May Have

- Thinking about the **workstation** recipe, could we do something like that for a web server?
- Is there a way to package up recipes you create with a version number (and maybe a README)?
- *I think Chef is able to generate something called a cookbook. Shouldn't we start thinking about some version control so we don't lose all our hard work?*

Collaboration and Version Control



Versioning Pros and Cons

Versioning Pros and Cons

Saving a copy of the original file as another filename.

```
$ cp setup.rb setup.rb.bak
```

Versioning Pros and Cons

Saving a copy of the original file as another filename.

```
$ cp setup.rb setup.rb.bak
```

or

```
$ cp setup.rb{,.`date +%Y%m%d%H%M`}
```

Versioning Pros and Cons

Saving a copy of the original file as another filename.

```
$ cp setup.rb setup.rb.bak
```

or

```
$ cp setup.rb{,.`date +%Y%m%d%H%M`} }
```

or

```
$ cp setup.rb{,.`date +%Y%m%d%H%M-$USER`} }
```

Git Version Control

git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.

We will be using git throughout the rest of this workshop.



Lab Exercise: Install **git**

- Add the additional policy to `setup.rb`:
The package named **git** is installed.
- Then apply this recipe with **chef-apply**.

Lab Exercise: Adding the **git** Package

~/setup.rb

```
package 'nano'
package 'vim'
package 'emacs'

package 'tree'
package 'git'

file '/etc/motd' do
  content 'Property of ...'
end
```

Lab Exercise: Re-apply the Setup Recipe

```
$ sudo chef-apply setup.rb
```

```
Recipe: (chef-apply cookbook)::(chef-apply recipe)
* yum_package[nano] action install (up to date)
* yum_package[vim] action install (up to date)
* yum_package[emacs] action install (up to date)
* yum_package[tree] action install (up to date)
* yum_package[git] action install
  - install version 1.7.1-4.el6_7.1 of package git
* file[/etc/motd] action create (up to date)
```

Group Exercise: Create a Cookbook

- How are we going to manage this file?
- Does it need a README?

Group Exercise: Create a Cookbook

- How are we going to manage this file?
- Does it need a README?

Objective:

- Use **chef** (CLI) to generate a cookbook to store our **setup** recipe.
- Add the **workstation** cookbook to version control.

What is **chef**?

An executable program that allows you generate cookbooks and cookbook components.

What can **chef** do?

```
$ chef --help
```

Usage:

```
chef -h/--help
chef -v/--version
chef command [arguments...] [options...]
```

Available Commands:

exec	Runs the command in context of the embedded ruby
env	Prints environment variables used by ChefDK
gem	Runs the gem command in context of the embedded ruby
generate	Generate a new app, cookbook, or component
shell-init	Initialize your shell to use ChefDK as your primary ru
install	Install cookbooks from a Policyfile and generate a loc
update	Updates a Policyfile.lock.json with latest run_list an
...	
undele	Undo a delete command
verify	Test the embedded ChefDK applications

Cookbooks

What is a Chef Cookbook?

Cookbooks

What is a Chef Cookbook?

- A Chef Cookbook is the fundamental unit of configuration and policy distribution.

Cookbooks

What is a Chef Cookbook?

- A Chef Cookbook is the fundamental unit of configuration and policy distribution.
- Each cookbook defines a scenario, such as everything needed to install and configure MySQL, and then it contains all of the components that are required to support that scenario.

Cookbooks

What is a Chef Cookbook?

- A Chef Cookbook is the fundamental unit of configuration and policy distribution.
- Each cookbook defines a scenario, such as everything needed to install and configure MySQL, and then it contains all of the components that are required to support that scenario.
- Read the first three paragraphs here:
<http://docs.chef.io/cookbooks.html>

What Can **chef generate** Do?

```
$ chef generate --help
```

```
Usage: chef generate GENERATOR [options]
```

```
Available generators:
```

app	Generate an application repo
cookbook	Generate a single cookbook
recipe	Generate a new recipe
attribute	Generate an attributes file
template	Generate a file template
file	Generate a cookbook file
lwrp	Generate a lightweight resource/provider
repo	Generate a Chef policy repository
policyfile	Generate a Policyfile for use with the install/push commands
generator	Copy ChefDK's generator cookbook so you can customize it

What Can **chef generate cookbook** Do?

```
$ chef generate cookbook --help
```

```
Usage: chef generate cookbook NAME [options]
  -C, --copyright COPYRIGHT      Name of the copyright holder - default...
  -m, --email EMAIL              Email address of the author - defaults...
  -a, --generator-arg KEY=VALUE  Use to set arbitrary attribute KEY to ...
  -I, --license LICENSE          all_rights, httpd, mit, gplv2, gplv3 -...
  -g GENERATOR_COOKBOOK_PATH,   Use GENERATOR_COOKBOOK_PATH for the...
    --generator-cookbook
```

Group Exercise: Let's Create a Cookbook

```
$ cd ~  
$ chef generate cookbook workstation
```

Installing Cookbook Gems:

Compiling Cookbooks...

Recipe: code_generator::cookbook

```
* directory[/home/bmeyer/workstation] action create  
  - create new directory /home/bmeyer/workstation  
* template[/home/bmeyer/workstation/metadata.rb] action create_if_missing  
  - create new file /home/bmeyer/workstation/metadata.rb  
  - update content in file /home/bmeyer/workstation/metadata.rb from none to 0c0  
  (diff output suppressed by config)  
* template[/home/bmeyer/workstation/README.md] action create_if_missing  
  - create new file /home/bmeyer/workstation/README.md  
  - update content in file /home/bmeyer/workstation/README.md from none to 59072  
  (diff output suppressed by config)  
...  
* execute[initialize-git] action run  
  - execute git init .  
* cookbook_file[/home/bmeyer/workstation/.gitignore] action create  
  - create new file /home/bmeyer/workstation/.gitignore  
  - update content in file /home/bmeyer/workstation/.gitignore from none to dd37  
  (diff output suppressed by config)
```

Group Exercise: The Cookbook Has a **README.md** file

```
$ tree workstation
```

```
workstation
├── Berksfile
├── cheffignore
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   └── spec_helper.rb
...
└── test
    └── integration
...
    └── helpers
        └── serverspec
            └── spec_helper.rb
10 directories, 9 files
```

README.md

The description of the cookbook's features written in Markdown.

<http://daringfireball.net/projects/markdown/syntax>

Group Exercise: The Cookbook Has Some Metadata

```
$ tree workstation
```

```
workstation
├── Berksfile
├── cheffignore
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   ├── unit
│   │   └── recipes
│   └── default_spec.rb
10 directories, 9 files
```

metadata.rb

- Every cookbook requires a small amount of metadata.

metadata.rb

- Every cookbook requires a small amount of metadata.
- Metadata is stored in a file called `metadata.rb` that lives at the top of each cookbook's directory.

http://docs.chef.io/config_rb_metadata.html

Group Exercise: Let's Take a Look at the Metadata

```
$ cat workstation/metadata.rb
```

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


Group Exercise: The Cookbook Has a Folder for Recipes

```
$ tree workstation
```

```
workstation
├── Berksfile
├── cheffignore
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   ├── unit
│   └── recipes
│       └── default_spec.rb
10 directories, 9 files
```

Group Exercise: The Cookbook Has a Default Recipe

```
$ cat workstation/recipes/default.rb
```

```
# Cookbook Name:: workstation  
# Recipe:: default  
#  
# Copyright (c) 2015 The Authors, All Rights Reserved.
```

Group Exercise: Move our **setup.rb** Recipe into the Cookbook

```
$ mv -v setup.rb workstation/recipes/  
`setup.rb' -> `workstation/recipes/setup.rb'
```

Group Exercise: Version Control

This is a probably a good point to capture the initial state of our cookbook.

Objective:

- ☑ Use **chef** (CLI) to generate a cookbook to store our setup recipe.
- ☐ Add the **workstation** cookbook to version control.

Group Exercise: Move into the Cookbook Directory

```
$ cd workstation
```

Group Exercise: Initialize the Directory as a **git** Repository

```
$ git init
```

```
Reinitialized existing Git repository in /home/chef/workstation/.git/
```

Group Exercise: Use **git add** to Stage Files to be Committed

```
$ git add .
```

Staging Area

- The staging area has a file, generally contained in your Git directory, that stores information about what will go into your next commit.
- It's sometimes referred to as the “index”, but it's also common to refer to it as the staging area.

<http://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

Group Exercise: Use **git status** to View the Staged Files

```
$ git status
```

```
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   .gitignore
#       new file:   .kitchen.yml
#       new file:   Berksfile
#       new file:   README.md
#       new file:   cheffignore
#       new file:   metadata.rb
#       new file:   recipes/default.rb
#       new file:   recipes/setup.rb
#       new file:   spec/spec_helper.rb
#       new file:   spec/unit/recipes/default_spec.rb
#       new file:   test/integration/default/serverspec/default_spec.rb
#       new file:   test/integration/helpers/serverspec/spec_helper.rb
#
```

Group Exercise: Use **git commit** to Save the Staged Changes

```
$ git commit -m "Initial workstation cookbook"
```

```
[master (root-commit) ae6968b] Initial workstation cookbook
Committer: ChefDK User <chef@ip-172-31-11-224.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

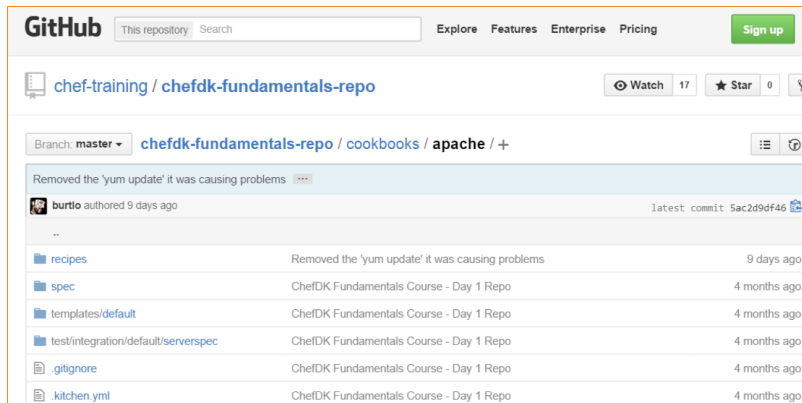
If the identity used for this commit is wrong, you can fix it with:

```
git commit --amend --author='Your Name <you@example.com>'
```

```
12 files changed, 207 insertions(+), 0 deletions(-)
create mode 100644 .gitignore
create mode 100644 .kitchen.yml
create mode 100644 Berksfile
create mode 100644 README.md
create mode 100644 chefignore
create mode 100644 metadata.rb
create mode 100644 recipes/default.rb
create mode 100644 recipes/setup.rb
...
```

Git Version Control

- If you use git versioning you should ultimately push the local git repository to a shared remote git repository.
- In this way others could collaborate with you from a centralized location.



Group Exercise: Move out of the Workstation Cookbook

```
$ cd ~
```

Lab Exercise: Setting up a Web Server

- Use **chef generate** to create a cookbook named **apache**.
- Write and apply a recipe named `server.rb` with the policy:
 - The package named **httpd** is installed.
 - The file named `/var/www/html/index.html` is created with the content `<h1>Hello, world!</h1>`
 - The service named **httpd** is started and enabled.
- Apply the recipe with **chef-apply**.
- Verify the site is available by running **curl http://localhost**.

Lab Exercise: Create a Cookbook

```
$ chef generate cookbook apache
```

```
Installing Cookbook Gems:
Compiling Cookbooks...
Recipe: code_generator::cookbook
  * directory[/home/bmeyer/apache] action create
    - create new directory /home/bmeyer/apache
  * template[/home/bmeyer/apache/metadata.rb] action create_if_missing
    - create new file /home/bmeyer/apache/metadata.rb
    - update content in file /home/bmeyer/apache/metadata.rb from none to 37ed5f
      (diff output suppressed by config)
  * template[/home/bmeyer/apache/README.md] action create_if_missing
    - create new file /home/bmeyer/apache/README.md
    - update content in file /home/bmeyer/apache/README.md from none to 5c3d3a
      (diff output suppressed by config)
...
  * cookbook_file[/home/bmeyer/apache/.gitignore] action create
    - create new file /home/bmeyer/apache/.gitignore
    - update content in file /home/bmeyer/apache/.gitignore from none to dd37b2
      (diff output suppressed by config)
```

Lab Exercise: Create a Cookbook

```
$ chef generate recipe apache server
```

```
Installing Cookbook Gems:
```

```
Compiling Cookbooks...
```

```
Recipe: code_generator::recipe
```

- * directory[./apache/spec/unit/recipes] action create (up to date)
- * cookbook_file[./apache/spec/spec_helper.rb] action create_if_missing (up to date)
- * template[./apache/spec/unit/recipes/server_spec.rb] action create_if_missing
 - create new file ./apache/spec/unit/recipes/server_spec.rb
 - update content in file ./apache/spec/unit/recipes/server_spec.rb from none to 3d6b92 (diff output suppressed by config)
- * template[./apache/recipes/server.rb] action create
 - create new file ./apache/recipes/server.rb
 - update content in file ./apache/recipes/server.rb from none to 3d6b92 (diff output suppressed by config)

Lab Exercise: Create the Server Recipe

~/apache/recipes/server.rb

```
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Hello, world!</h1>'
end

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


Lab Exercise: Apply the Server Recipe

```
$ sudo chef-apply apache/recipes/server.rb
```

```
Recipe: (chef-apply cookbook)::(chef-apply recipe)
* yum_package[httpd] action install
  - install version 2.4.6-40.el7.centos of package httpd
* file[/var/www/html/index.html] action create
  - create new file /var/www/html/index.html
  - update content in file /var/www/html/index.html from none to 17d291
  --- /var/www/html/index.html      2016-04-04 09:50:59.758483838 -0500
  +++ /var/www/html/.chef-index.html20160404-21732-1o44k9g      2016-04-04 09:50:5
  @@ -1 +1,2 @@
  +<h1>Hello, world!</h1>
* service[httpd] action enable
  - enable service service[httpd]
* service[httpd] action start
  - start service service[httpd]
```

Lab Exercise: Verify Website is Available

```
$ curl localhost  
<h1>Hello, world!</h1>
```

Group Exercise: Commit Your Work

```
$ cd apache  
$ git init  
$ git add .  
$ git commit -m "Initial Apache Cookbook"
```

Discussion

- What file would you read first when examining a cookbook?

Discussion

- What file would you read first when examining a cookbook?
- What other recipes might you include in the **apache** or **workstation** cookbook?

Discussion

- What file would you read first when examining a cookbook?
- What other recipes might you include in the **apache** or **workstation** cookbook?
- Can resources accept multiple actions?

Discussion

- What file would you read first when examining a cookbook?
- What other recipes might you include in the **apache** or **workstation** cookbook?
- Can resources accept multiple actions?
- How often would you commit changes with version control?

Q&A

What questions can we answer for you?

- Cookbooks
- Versions
- Version control

On to **Module 4**



Slide 57 of 57