



Cookbooks

Organizing Recipes

Objectives



After completing this module you should be able to:

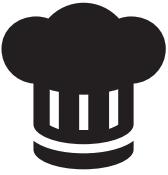
- Generate a Chef cookbook
- Define a Chef recipe that sets up a web server



Questions You May Have

1. Is there a way to package up recipes you create with a version number and a README?
2. If we have multiple versions - how might we better track these different changes and versions?
3. Thinking about the previous recipes, could we create a recipe to setup a web server?

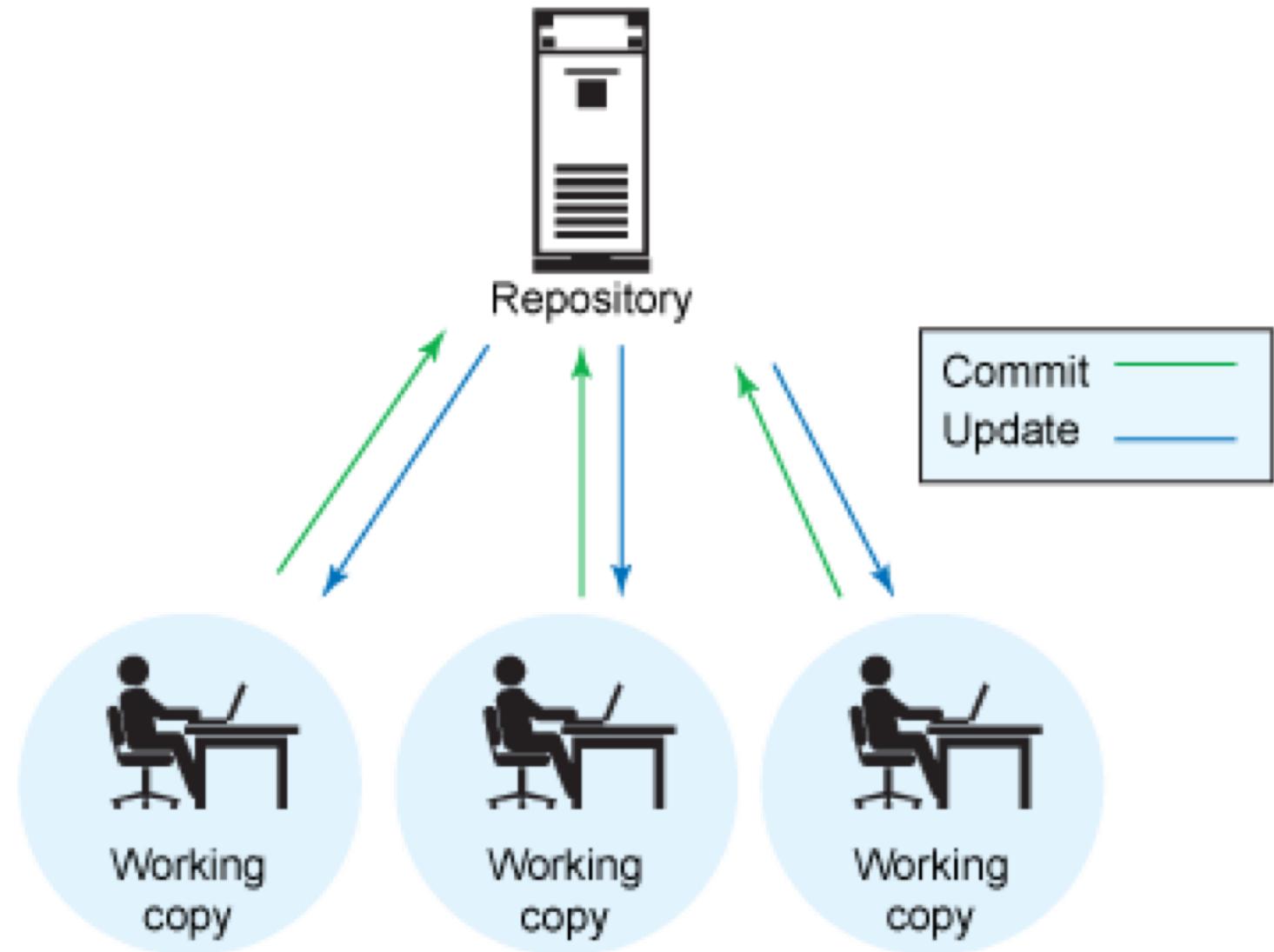
Collaboration and Version Control



A versioning system should include:

- A Central Repository into which all the developers publish their work.
- Each revision should be stored as a new version.

Centralized version control system



Collaboration and Version Control



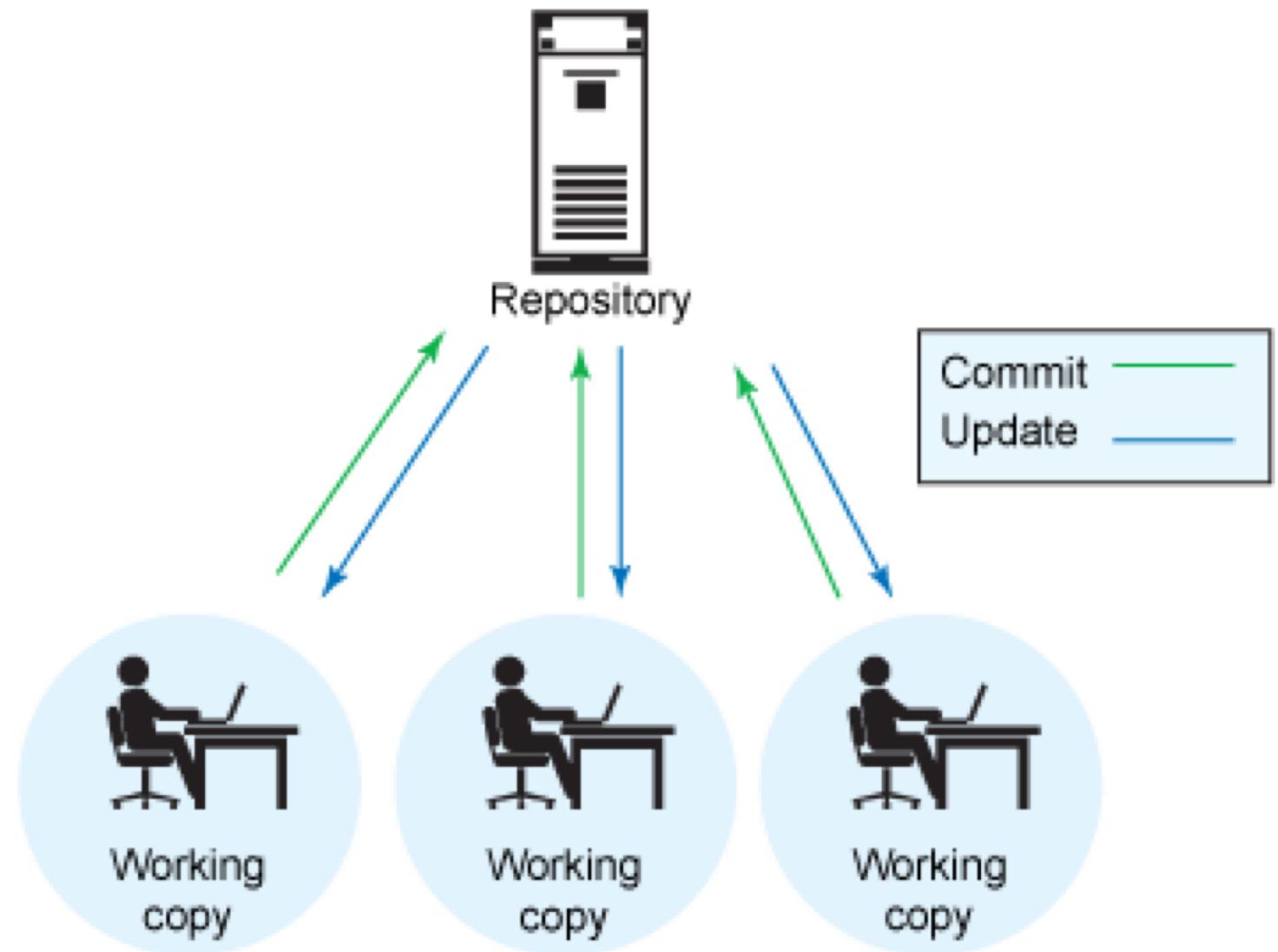
We won't cover versioning control systems, such as Git, in this course. But it's something you should eventually learn to use.

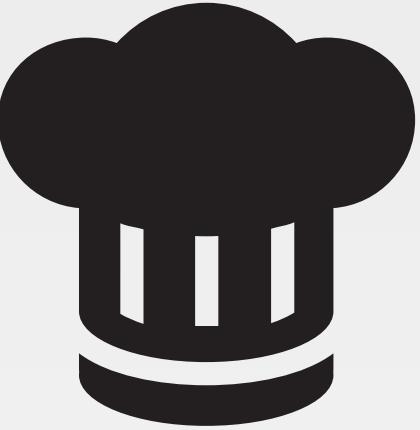
There are plenty of resources on the web that you can use to learn versioning control.

You can also get started by going through this *Learn Chef Rally* module.

<https://learn.chef.io/modules/version-control#/>

Centralized version control system





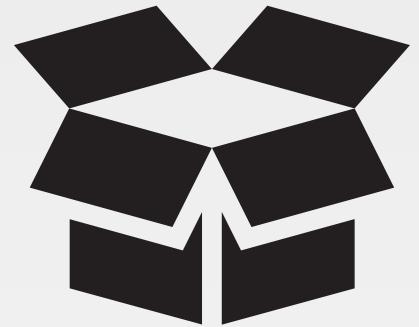
GL: Create a Cookbook

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

Objective:

- Use chef to generate a cookbook to store our disable-uac recipe.

CONCEPT



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
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 ruby
install	Install cookbooks from a Policyfile and generate a locked cookbook
set	
update	Updates a Policyfile.lock.json with latest run_list and cookbooks

Cookbooks



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>



GL: Ensure You are in Your Home Directory



```
> cd ~
```

```
[REDACTED]
```

A large black rectangular area covering the majority of the slide content, likely redacting sensitive information or code output.

REMOTE

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
(experimental)

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 | <code>all_rights</code> , <code>httpd</code> , <code>mit</code> , <code>gplv2</code> , <code>gplv3</code> - |
| -g GENERATOR_COOKBOOK_PATH,
--generator-cookbook | Use GENERATOR_COOKBOOK_PATH for the |

REMOVED

GL: Let's Create a Cookbook



```
> chef generate cookbook cookbooks\workstation
```

Generating cookbook workstation

- Ensuring correct cookbook file content
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content

Your cookbook is ready. Type `cd cookbooks/workstation` 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.

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

`test/recipes/default_test.rb`

GL: The Cookbook Has a README

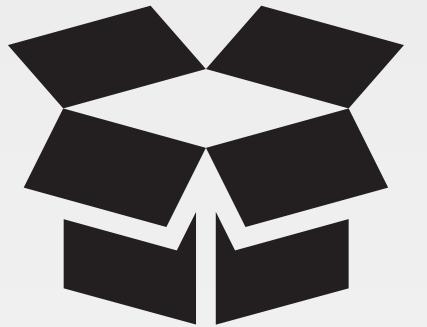


```
> tree /f cookbooks\workstation
```

```
Volume serial number is 2CB6-07DF
C:\USERS\ADMINISTRATOR\COOKBOOKS\WORKSTATION
|
|   .gitignore
|
|   .kitchen.yml
|
|   Berksfile
|
|   cheftignore
|
|   metadata.rb
|
|   README.md
|
|
|   .delivery
|       config.json
|       project.toml
|
|       ...

```

CONCEPT



README.md

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

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

GL: The Cookbook Has Some Metadata



```
> tree /f cookbooks\workstation
```

```
Folder PATH listing
Volume serial number is B04A-119C
C:\USERS\ADMINISTRATOR\COOKBOOKS\WORKSTATION
|
|   ...
|   .kitchen.yml
|   Berksfile
|   cheignore
|   metadata.rb
|   README.md
|
|   --recipes
|       default.rb
|   
```



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

GL: Let's Take a Look at the Metadata



```
> gc cookbooks\workstation\metadata.rb
```

```
name 'workstation'  
maintainer 'The Authors'  
maintainer_email 'you@example.com'  
license 'All Rights Reserved'  
description 'Installs/Configures workstation'  
long_description 'Installs/Configures workstation'  
version '0.1.0'  
chef_version '>= 12.1' if respond_to?(:chef_version)
```

GL: The Cookbook Has a Folder for Recipes



```
> tree /f cookbooks\workstation
```

```
Folder PATH listing
```

```
Volume serial number is B04A-119C
```

```
C:\USERS\ADMINISTRATOR\COOKBOOKS\WORKSTATION
```

```
| ...  
| .kitchen.yml  
| Berksfile  
| chefignore  
| metadata.rb  
| README.md
```

```
|  
+--recipes  
|   default.rb  
|  
|
```

GL: The Cookbook Has a Default Recipe



```
> gc cookbooks\workstation\recipes\default.rb
```

```
# Cookbook Name:: workstation
# Recipe:: default
#
# Copyright (c) 2018 The Authors, All Rights Reserved.
```

REMOTE

GL: Copy the Recipe into the Cookbook



```
cookbooks> mv disable-uac.rb workstation\recipes
```

REMOTE



Lab: Setting up a Web Server

- ❑ Use `chef generate` to create a cookbook named "`myiis`".
- ❑ Write and apply a recipe named "`server.rb`" with the policy:
 - The `powershell_script` named 'Install IIS' is run with the code 'Add-WindowsFeature Web-Server'.
 - The `file` named 'C:\inetpub\wwwroot\Default.htm' is created with the content '`<h1>Hello, world!</h1>`'
 - The `service` named 'w3svc' is started and enabled.
- ❑ Apply the recipe and verify the site is available by running `Invoke-WebRequest localhost`

Lab: Create a Cookbook



```
> chef generate cookbook cookbooks\myiis
```

Generating cookbook myiis

- 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\myiis` to enter it..

Lab: Create a Recipe



```
> chef generate recipe cookbooks\myiis server
```

```
...
* template[cookbooks/myiis/spec/unit/recipes/server_spec.rb] action create_if_missing
  - create new file cookbooks/myiis/spec/unit/recipes/server_spec.rb
  - update content in file cookbooks/myiis/spec/unit/recipes/server_spec.rb from none to
53eaa6
  (diff output suppressed by config)
* directory[cookbooks/myiis/test/smoke/default] action create (up to date)
* template[cookbooks/myiis/test/smoke/default/server_test.rb] action create_if_missing
  - create new file cookbooks/myiis/test/smoke/default/server_test.rb
  - update content in file cookbooks/myiis/test/smoke/default/server_test.rb from none to
abd78c
  (diff output suppressed by config)
* template[cookbooks/myiis/recipes/server.rb] action create
  - create new file cookbooks/myiis/recipes/server.rb
  - update content in file cookbooks/myiis/recipes/server.rb from none to 8fb075
  (diff output suppressed by config)
```

Lab: Create Server Recipe

~\cookbooks\myiis\recipes\server.rb

```
powershell_script 'Install IIS' do
  code 'Add-WindowsFeature Web-Server'
end

file 'C:\inetpub\wwwroot\Default.htm' do
  content '<h1>Hello, world!</h1>'
end

service 'w3svc' do
  action [:enable, :start]
end
```

Lab: Apply the Server Recipe



```
C:\Users\chef\cookbooks> chef-client -z -r  
recipe[myiis::server]
```

```
...  
Recipe: @recipe_files::C:/Users/Administrator/cookbooks/myiis/recipes/server.rb  
"C:/Users/ADMINI~1/AppData/Local/Temp/2/chef-script2018091  
pii.ps1"  
  * file[C:\inetpub\wwwroot\Default.htm] action create  
    - create new file C:\inetpub\wwwroot\Default.htm  
    - update content in file C:\inetpub\wwwroot\Default.htm from none to 17d291  
      --- C:\inetpub\wwwroot\Default.htm 2018-09-13 21:00:07.000000000 +0000  
      +++ C:\inetpub\wwwroot\chef-Default20180913-3148-u3dbpm.htm 2018-09-13  
      21:00:07.000000000 +0000  
        @@ -1 +1,2 @@  
        +<h1>Hello, world!</h1>  
  * windows_service[w3svc] action enable (up to date)  
  * windows_service[w3svc] action start (up to date)
```

Lab: Verify That the Website is Available



> Invoke-WebRequest localhost

```
StatusCode      : 200
StatusDescription : OK
Content          : <h1>Hello, world!</h1>
RawContent       : HTTP/1.1 200 OK
...
                           ETag: "9e9c7b47d32cd31:0"
                           Last-Modified: Wed, 13 Sep 2018 21:00:07 GMT
                           Server...
Forms            : {}
Headers          : { [Accept-Ranges, bytes], [Content-Length, 22], [Content-Type, text/html], [Date, Wed, 13 Sep 2018
                           21:02:57 GMT]... }
Images           : {}
```



Lab: Setting up a Web Server

- ✓ Use `chef generate` to create a cookbook named "`myiis`".
- ✓ Write and apply a recipe named "`server.rb`" with the policy:
 - The `powershell_script` named 'Install IIS' is run with the code 'Add-WindowsFeature Web-Server'.
 - The `file` named 'C:\inetpub\wwwroot\Default.htm' is created with the content '`<h1>Hello, world!</h1>`'
 - The `service` named 'w3svc' is started and enabled.
- ✓ Apply the recipe and verify the site is available by running `Invoke-WebRequest localhost`



Discussion

What file should you read first when examining a cookbook?

What other recipes might you include in the myiis or workstation cookbook?

Can resources accept multiple actions?



Q&A

What questions can we answer for you?

- Cookbooks



CHEF™