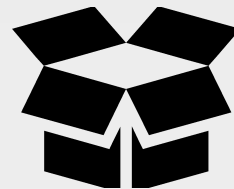


GL: Create a Cookbooks Directory



```
$ mkdir cookbooks
```

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 cookboo...
```

```
update        Updates a Policyfile.lock.json with latest run_list and cookbooks
```

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

GL: Let's Create a Cookbook



```
$ chef generate cookbook cookbooks/workstation
```

```
Compiling Cookbooks...
```

```
Recipe: code_generator::cookbook
```

```
* directory[/home/chef/workstation] action create
```

```
- create new directory /home/chef/workstation
```

```
* template[/home/chef/workstation/metadata.rb] action create_if_missing
```

```
- create new file /home/chef/workstation/metadata.rb
```

```
- update content in file /home/chef/workstation/metadata.rb from none to bd85d3  
(diff output suppressed by config)
```

```
* template[/home/chef/workstation/README.md] action create_if_missing
```

```
- create new file /home/chef/workstation/README.md
```

```
- update content in file /home/chef/workstation/README.md from none to 44d165  
(diff output suppressed by config)
```

```
* cookbook_file[/home/chef/workstation/chefignore] action create
```

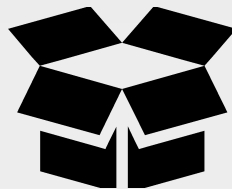
GL: The Cookbook Has a README



```
$ tree cookbooks/workstation
```

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

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 cookbooks/workstation
```

```
workstation
├── Berksfile
├── chefignore
├── 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 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



```
$ cat cookbooks/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'
```

GL: The Cookbook Has a Folder for Recipes



```
$ tree cookbooks/workstation
```

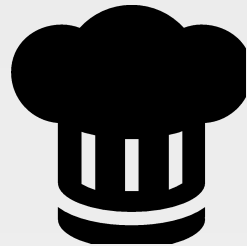
```
workstation
├── Berksfile
├── cheffignore
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── spec
│   ├── spec_helper.rb
│   └── unit
│       └── recipes
│           └── default_spec.rb
10 directories, 9 files
```

GL: The Cookbook Has a Default Recipe



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

```
# Cookbook Name:: workstation  
# Recipe:: default  
#  
# Copyright (c) 2016 The Authors, All Rights Reserved.
```



GL: Create a Cookbook

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

Objective:

- ✓ Use chef to generate a cookbook
- ✓ Move the setup recipe into the new cookbook
- ✓ Add the new cookbook to version control

GL: Copy the Recipe into the Cookbook



```
$ mv setup.rb cookbooks/workstation/recipes
```

GL: Verify the Cookbook has the Recipe



```
$ tree cookbooks/workstation
```

```
cookbooks/workstation
```

```
|— Berksfile
```

```
|— cheffignore
```

```
|— metadata.rb
```

```
|— README.md
```

```
|— recipes
```

```
|   |— default.rb
```

```
|   └─ setup.rb
```

```
|— spec
```

```
|   |— spec_helper.rb
```

```
|   └─ unit
```

```
|       └─ recipes
```

