

CHEF PARTNER COOKBOOKS

Chef's Partner Engineering Team

partnereng@chef.io

WHY WRITE CHEF COMMUNITY COOKBOOKS?

- Help customers succeed
- Reach thousands of joint users
- Users provide real-world feedback
- Good marketing opportunity

WHAT IS A CHEF PARTNER COOKBOOK?

- Chef's Partner Engineering helps you write your cookbook
- The cookbook is Open Source and freely available
 - we want to help all our users succeed with both companies' software
- Joint marketing opportunities
- Working on a **Supermarket** badge and **Chef.io** landing page
- All the benefits of Community Cookbooks!

HOW TO WRITE A "GOOD" COOKBOOK

- "Good" is subjective, but there are guidelines to do better
- Use the ChefDK for development
 - downloads.chef.io/chef-dk/
 - chef generate cookbook
 - fill in the appropriate content
 - check the LICENSE.md (we prefer Apache v2)

WRITING THE README

- Fill in the chef generate cookbook's README.md
- Add a SCOPE of what it does and doesn't do
- List the REQUIREMENTS for using this cookbook
- List the platforms tested and what is supported
- Document any other cookbooks or ruby libraries as dependencies
- Are there other external packages needed for usage?
 - document how/where to get them
- Explain the USAGE
- How/why it is tested or refer to a TESTING.md
- If there are Resources provided, document each and their properties

COMMUNITY COOKBOOK DOCUMENTATION

- Examples for how Chef Software cookbooks are managed
- github.com/chef-cookbooks/community_cookbook_documentation
- CONTRIBUTING.md explaining how to help further development
- LICENSE Apache v2 license provided (our preferred license)
- TESTING.md explaining how to test this cookbook

COMMUNITY QUALITY METRICS

- Community-provided documentation for objectively assessing cookbooks
- These are leading indicators of quality
 - usually easy to achieve, show you took the time to try to do a good job
- github.com/chef-cookbooks/cookbook-quality-metrics
- You should be able check off on these before sharing
- Eventually may be supported on [Supermarket](#)

RUBY STYLE

- **Rubocop**
 - included with the **ChefDK**
- Ruby static code analyzer and style enforcement
 - community Ruby style guide, still very subjective
- Document your overrides or disagreements and move on
 - considering implications of rules is more important than blindly following
- No established default, there is a lot of cargo-culting
 - the MySQL one is a good starter
 - github.com/chef-cookbooks/mysql/blob/master/.rubocop.yml

LINTING AND COOKBOOK STYLE

- Foodcritic
 - included with the ChefDK
- Chef style guide for cookbooks
 - eliminates potential issues
 - www.foodcritic.io
 - document your overrides or disagreements and move on

UNIT TESTS

- ChefSpec
 - included with the ChefDK
- Chef-specific unit tests, extension of rspec
- Use to test logic of cookbooks
- Just having it present catches compilation errors
- github.com/sethvargo/chefspec has lots of examples
- betterspecs.org for thoughts on better unit testing

INTEGRATION TESTS

- Test Kitchen
 - included with the ChefDK
- Integration test with "real" machines
- Tests on each supported platform
- [kitchen.ci](#)
- Start with chef generate cookbook's .kitchen.yml
- Use InSpec to test, use the Test Kitchen driver
 - docs.chef.io/inspec_reference.html
 - github.com/chef/kitchen-inspec

HOSTED TESTING TOOLS

- These may be configured to automatically test incoming PRs on GitHub
- Code Climate
 - checks code quality and runs automated tools
 - [Code Climate blog post on testing for Chef](#)
 - example github.com/chef-cookbooks/mysql/blob/master/.codeclimate.yml
- Travis.CI
 - Rubocop, Food Critic, ChefSpec and even Test Kitchen all supported
 - example: github.com/chef-cookbooks/mysql/blob/master/.travis.yml

OTHER THINGS WE LOOK FOR

- Semantic versioning observed
- Use Chef's built-in Resources
- Windows support and testing where applicable
- Attributes defined and documented in the `attributes/default.rb`
- Idempotent recipes
 - recipes requiring multiple runs are generally discouraged
 - run the recipes twice to catch non-idempotent behavior
- Private Recipes are generally a good pattern

CUSTOM RESOURCES

- Safely abstract away messy CLI or API calls
- Provide opportunities for error handling and conditional logic
- Benefits of standard Resources
 - types, names, parameters
 - notifications, default parameters, etc.
- Develop against Chef 12.5 or later
- Custom Resources added to make development much easier
 - docs.chef.io/decks/custom_resources.html
- Backwards-compatible library for earlier Chef versions
 - supermarket.chef.io/cookbooks/compat_resource

ADVANCED DEBUGGING

- **Pry** is an interactive shell for debugging Ruby
- Chef has the pry gem already included as a dependency
- Include this block in your recipes, libraries or custom Resources to drop into pry during a chef-client run

```
require 'pry'  
binding.pry
```

SUPERMARKET

- Chef's community site for sharing cookbooks
 - and other extensions
- supermarket.chef.io
- Publish your cookbook
- Fill all the extra fields
- "Chef Partner Cookbook" badge coming
 - Marketing will eventually add a landing page
 - filter searches to highlight these results

JOINING THE CHEF PARTNER COOKBOOK PROGRAM

- Contact Chef's Partner Engineering
partnereng@chef.io
 - Discuss proposed cookbook features
 - Provide examples and documentation
 - Answer technical questions, code reviews, and provide feedback
 - Help promote usage

THANKS!

Chef's Partner Engineering Team

partnereng@chef.io