**Chef Supermarket**

**Chef Supermarket is the site for community cookbooks. It provides an easily searchable cookbook repository and a friendly web UI. Cookbooks that are part of the Chef Supermarket are accessible by any Chef user.**

( for eg: predefined templates of Cookbooks for apache,weblogic,jboss,IIS)

There are two ways to use Chef Supermarket:

* The public Chef Supermarket is hosted by Chef and is located at <https://supermarket.chef.io>.
* A private Chef Supermarket may be installed on-premise behind the firewall on the internal network. Cookbook retrieval from a private Chef Supermarket is often faster than from the public Chef Supermarket because of closer proximity and fewer cookbooks to resolve. A private Chef Supermarket can also help formalize internal cookbook release management processes (e.g. “a cookbook is not released until it’s published on the private Chef Supermarket”).

**Public Supermarket**

The public Chef Supermarket hosted by Chef is located at <https://supermarket.chef.io>.

To interact with the public Chef Supermarket, **use** [***knife cookbook site***](https://docs.chef.io/knife_cookbook_site.html) **commands**.

**Private Supermarket**

The private Chef Supermarket is installed behind the firewall on the internal network. Outside of changing the location from which community cookbooks are maintained, it otherwise behaves the same as the public Chef Supermarket.

Note

The source code for Chef Supermarket is located at the following URLs:

* The application itself: <https://github.com/chef/supermarket>. Report issues to: <https://github.com/chef/supermarket/issues>.
* The code that builds Chef Supermarket as an omnibus package: <https://github.com/chef/omnibus-supermarket>. Use a Kitchen-based environment to build your own omnibus packages.
* The cookbook that is run by the supermarket-ctl reconfigure command: <https://github.com/chef/supermarket/tree/master/omnibus/cookbooks/omnibus-supermarket>

**Recommended Tools**

The following tools are recommended for use with a private Chef Supermarket:

* Berkshelf
* Stove

**Berkshelf**

Berkshelf can include multiple Chef Supermarket instances for dependency resolution. Cookbook dependency resolution is performed from the top down. The first source defined in the Berksfile will be searched for the cookbook before the second source.

The Berksfile first looks for the cookbook on the private Chef Supermarket and, if not discovered there, then looks on the public Chef Supermarket.

source 'https://your\_private\_supermarket\_url'

source 'https://supermarket.chef.io'

**Stove**

Stove is a utility for packaging and releasing Chef cookbooks: <https://github.com/sethvargo/stove>.

**Installing Private Supermarket**

To install a Private Supermarket, see the instructions [*here*](https://docs.chef.io/install_supermarket.html).

**Set up Workstation**

If you are using Chef 12.13 or higher, use the [*knife cookbook site*](https://docs.chef.io/knife_cookbook_site.html) commands to work with cookbooks in both Public Chef Supermarket and a Private Chef Supermarket.

If you are using Chef 12.12 or lower, use the knife supermarket plugin to work with cookbooks in a Private Chef Supermarket.

**knife**

**Chef 12.13 and higher**

If you are using Chef 12.13 or higher, use the [*knife cookbook site*](https://docs.chef.io/knife_cookbook_site.html) commands with BOTH Public Supermarket and Private Supermarket.

**Chef 12.12 and lower**

If you are using Chef 12.12 or a previous version, you will need to use knife supermarket.

The knife supermarket command is a plugin that must be installed to the workstation.

If using the Chef development kit, run the following command:

$ chef gem install knife-supermarket

and if not using the Chef development kit, run the following command:

$ gem install knife-supermarket

**Configure knife.rb**

The knife.rb file on the workstation should be configured for use with the private Chef Supermarket.

To configure knife.rb for the private Chef Supermarket, do the following:

1. Open the knife.rb file in an editor.
2. Add the following setting:
3. knife[:supermarket\_site] = 'https://your-private-supermarket'
4. Save and close the file.

**Create a Cookbook**

The following examples show how to create a simple cookbook by using the chef command that is built into the the Chef development kit.

**Generate a chef-repo**

To generate a chef-repo, run a command similar to:

$ chef generate repo my\_chef\_repo

Access the chef-repo using the cd command:

$ cd my\_chef\_repo

**Generate a cookbook**

Note

Duplicate cookbook names on Chef Supermarket are not allowed. So first verify that a cookbook name is available.

To create the my\_apache2\_cookbook cookbook, run the following command:

$ chef generate cookbook cookbooks/my\_apache2\_cookbook

**Generate a template**

To generate a template, run a command similar to:

$ chef generate template cookbooks/my\_apache2\_cookbook index.html

This will create a file named index.html.etb in the /cookbooks/my\_apache2\_cookbook directory. Open the file using a text editor to add content. For example, some HTML:

<html>

<body>

<h1>Chef Love!</h1>

</body>

</html>

Save and close the file.

**Create a recipe**

The default.rb recipe is created when a cookbook is generated. A recipe is updated using a text editor. For example:

package 'apache2' # Installs the apache2 package

service 'apache2' do

action [:start, :enable] # Starts and enables the apache2 service on boot

end

template '/var/www/html/index.html' do

source 'index.html.erb' # Template for /var/www/html/index.html

end

**Upload a Cookbook**

To upload a cookbook to Chef Supermarket, do the following:

1. Determine which version of Chef you are using.

If you are using Chef 12.13 or later, you have everything you need in the knife cookbook site commands

If you are using Chef 12.12 or earler, you need to install the knife supermarket plugin:

$ chef gem install knife-supermarket

1. Add a setting for Chef Supermarket to the knife.rb file:
2. knife[:supermarket\_site] = 'https://your-private-supermarket'
3. Resolve SSL errors by fetching, and then verifying the SSL certificate for Chef Supermarket:
4. $ knife ssl fetch https://your-private-supermarket

and then:

$ knife ssl check https://your-private-supermarket

1. Upload the cookbook to Chef Supermarket:

If you are using Chef 12.13 or later:

$ knife cookbook site share mycookbook "Other"

If you are using Chef 12.12 or earlier:

$ knife supermarket share mycookbook "Other"

**Share a Cookbook**[**¶**](https://docs.chef.io/supermarket.html#share-a-cookbook)

If you are using Chef 12.13 or later, a cookbook may be shared to the private Chef Supermarket using the knife cookbook site` commands.

$ knife cookbook site share 'my\_cookbook'

If you are using Chef 12.12 or lower, a cookbook may be shared to the private Chef Supermarket using the knife supermarket command. Run the following command:

$ knife supermarket share 'my\_cookbook'

**Troubleshoot SSL Errors**[**¶**](https://docs.chef.io/supermarket.html#troubleshoot-ssl-errors)

If an SSL error is returned similar to:

ERROR: Error uploading cookbook my\_cookbook to the Opscode Cookbook Site: SSL\_connect returned=1 errno=0 state=SSLv3 read server certificate B: certificate verify failed. Increase log verbosity (-VV) for more information.

this is because Chef server version 12.0 (and higher) enforces SSL by default when sharing cookbooks. A private Chef Supermarket uses self-signed certificates by default. Use the knife ssl fetch and knife ssl check commands to resolve this error.

First fetch the SSL certificate for the private Chef Supermarket:

$ knife ssl fetch https://your-private-supermarket

and then:

$ knife ssl check https://your-private-supermarket

Re-share the cookbook. This time the message returned should be similar to:

Generating metadata for my\_cookbook from (...)

Making tarball my\_cookbook.tgz

Upload complete!

**supermarket-ctl (executable)**[**¶**](https://docs.chef.io/supermarket.html#supermarket-ctl-executable)

The Chef Supermarket installations that are done using the omnibus installer include a command-line utility named supermarket-ctl. This command-line tool is used to start and stop individual services, reconfigure the Chef Supermarket server, run smoke tests, and tail the Chef Supermarket log files.

For more information about the supermarket-ctl command line tool, see [*supermarket-ctl*](https://docs.chef.io/ctl_supermarket.html).

**supermarket.rb**

The supermarket.rb file contains all of the non-default configuration settings used by the Chef Supermarket. (The default settings are built-in to the Chef Supermarket configuration and should only be added to the supermarket.rb file to apply non-default values.) These configuration settings are processed when the supermarket-ctl reconfigure command is run, such as immediately after setting up Chef Supermarket or after making a change to the underlying configuration settings after the server has been deployed. The supermarket.rb file is a Ruby file, which means that conditional statements can be used in the configuration file.

For more information about the supermarket.rb file, see [*supermarket.rb*](https://docs.chef.io/config_rb_supermarket.html).

**Supermarket API**

The Supermarket API is used to provide access to cookbooks, tools, and users on the Chef Supermarket at <https://supermarket.chef.io/cookbooks>. All of the cookbooks, tools, and users on the Supermarket are accessible through a RESTful API located at <https://supermarket.chef.io/api/v1> by using any of the supported endpoints. In most cases, using knife is the best way to interact with the Supermarket; in some cases, using the Supermarket API directly is necessary.

For more information about the Supermarket API, see [*Supermarket API*](https://docs.chef.io/supermarket_api.html).

**fieri[¶](https://docs.chef.io/supermarket.html" \l "fieri" \o "Permalink to this headline)**

Fieri is an optional service what will check cookbook versions for certain metrics to determine the quality of the cookbook.

As of Supermarket 2.7, Fieri now lives within the Supermarket code base.

If you are using a private Chef Supermarket, you can activate the Fieri service like this:

1. Add Fieri to your features attribute.
2. ['supermarket\_omnibus']['config']['features'] = "tools,github,announcement,fieri"
3. Add the following Fieri attributes:
4. ['supermarket\_omnibus']['config']['fieri\_key'] = "#{random string you generate}"
5. ['supermarket\_omnibus']['config']['fieri\_supermarket\_endpoint'] = "#{your\_supermarket\_url}"
6. Reconfigure your Supermarket.
7. (your-supermarket-node) $ sudo supermarket-ctl reconfigure
8. (your-supermarket-node) $ sudo supermarket-ctl restart

After doing these steps, you should see a “Quality” tab when viewing a cookbook through the Supermarket UI. Click on this tab and you will see the results of the metrics run by Fieri.