

# **chef-client**

Applying Recipes from Multiple Cookbooks

## chef-client



```
$ sudo chef-client --local-mode RECIPE_FILE
```

How would we apply both the workstation's setup recipe and apache's server recipe?

# Objectives

After completing this module, you should be able to use chef-client to:

- Locally apply multiple cookbooks' recipes with chef-client.
- Include a recipe from within another recipe.

## --local-mode



chef-client's default mode attempts to contact a Chef Server and ask it for the recipes to run for the given node.

We are overriding that behavior to have it work in a local mode.

# CONCEPTS



```
--run-list "recipe[COOKBOOK::RECIPE]"
```

In local mode, we need to provide a list of recipes to apply to the system. This is called a **run list**. A run list is an ordered collection of recipes to execute.

Each recipe in the run list must be addressed with the format **recipe[COOKBOOK::RECIPE]**.

# Demo: Using 'chef-client' to Locally Apply Recipes

```
$ sudo chef-client --local-mode -r "recipe[workstation::setup]"
```

Applying the following recipes locally:

The 'setup' recipe from the 'workstation' cookbook

# Demo: Using 'chef-client' to Locally Apply Recipes

```
$ sudo chef-client --local-mode -r "recipe[apache::server]"
```

Applying the following recipes locally:

The 'server' recipe from the 'apache' cookbook

# Demo: Using 'chef-client' to Locally Apply Recipes

```
$ sudo chef-client --local-mode \  
  -r "recipe[workstation::setup],recipe[apache::server]"
```

Applying the following recipes locally:

- The 'setup' recipe from the 'workstation' cookbook
- The 'server' recipe from the 'apache' cookbook





## Applying a Run List

*Using a run list will allow us to specify things more 'logically' instead of with paths.*

### Objective:

- ☐ Individually apply the apache cookbook's server recipe and workstation cookbook's setup recipe
- ☐ Apply both the apache cookbook's server recipe and workstation cookbook's setup recipe

# GL: Return Home First



```
$ cd ~
```

# GL: Apply the Cookbook Recipe Locally



```
$ sudo chef-client --local-mode -r "recipe[apache::server]"
```

```
[2016-09-15T14:54:45+00:00] WARN: No config file found or specified on command line, using command line options.
```

```
Starting Chef Client, version 12.3.0
```

```
resolving cookbooks for run list: ["apache::server"]
```

```
Synchronizing Cookbooks:
```

```
- apache
```

```
Compiling Cookbooks...
```

```
Converging 4 resources
```

```
Recipe: apache::server
```

- \* yum\_package[httpd] action install (up to date)
- \* file[/var/www/html/index.html] action create (up to date)
- \* service[httpd] action enable (up to date)

# GL: Apply the Cookbook Recipe Locally



```
$ sudo chef-client --local-mode -r "recipe[workstation::setup]"
```

```
[2016-09-15T15:15:26+00:00] WARN: No config file found or specified on command line, using command line options.
```

```
Starting Chef Client, version 12.3.0
```

```
resolving cookbooks for run list: ["workstation::setup"]
```

```
Synchronizing Cookbooks:
```

```
  - workstation (0.1.0)
```

```
Compiling Cookbooks...
```

```
Converging 4 resources
```

```
Recipe: workstation::setup
```

- \* yum\_package[cowsay] action install (up to date)
- \* yum\_package[tree] action install (up to date)
- \* yum\_package[git] action install (up to date)
- \* file[/etc/motd] action create (up to date)



## Applying a Run List

*Using a run list will allow us to specify things more 'logically' instead of with paths.*

### Objective:

- ✓ Individually apply the apache cookbook's server recipe and workstation cookbook's setup recipe
- ❑ Apply both the apache cookbook's server recipe and workstation cookbook's setup recipe

# GL: Apply Both Recipes Locally



```
$ sudo chef-client --local-mode \  
-r "recipe[apache::server],recipe[workstation::setup]"
```

```
[2016-09-15T15:17:27+00:00] WARN: No config file found or specified on command  
line, using command line options.
```

```
Starting Chef Client, version 12.3.0
```

```
resolving cookbooks for run list: ["apache::server","workstation::setup"]
```

```
Synchronizing Cookbooks:
```

- apache
- workstation

```
Compiling Cookbooks...
```

```
Running handlers:
```

```
[2016-09-15T15:17:30+00:00] ERROR: Running exception handlers
```

```
Running handlers complete
```



## Applying a Run List

*Using a run list will allow us to specify things more 'logically' instead of with paths.*

### Objective:

- ✓ Individually apply the apache cookbook's server recipe and workstation cookbook's setup recipe
- ✓ Apply both the apache cookbook's server recipe and workstation cookbook's setup recipe



## A Succinct Run List

*The cookbook only has one recipe that we care about. Could we set that up as the default?*

### Objective:

- ☐ Load the workstation cookbook's setup recipe in the default recipe
- ☐ Apply the workstation cookbook's default recipe
- ☐ Commit the changes



# CONCEPTS

```
-r "recipe[COOKBOOK(::default)]"
```



When you are referencing the default recipe within a cookbook you may optionally specify only the name of the cookbook.

chef-client understands that you mean to apply the default recipe from within that cookbook.

## include\_recipe



A recipe can include one (or more) recipes located in cookbooks by using the `include_recipe` method. When a recipe is included, the resources found in that recipe will be inserted (in the same exact order) at the point where the `include_recipe` keyword is located.

<https://docs.chef.io/recipes.html#include-recipes>

# Demo: Including a Recipe

```
include_recipe 'workstation::setup'
```

Include the 'setup' recipe from the 'workstation' cookbook in this recipe

# Demo: Including a Recipe

```
include_recipe 'apache::server'
```

Include the 'server' recipe from the 'apache' cookbook in this recipe

# GL: The Default Recipe Includes the Setup Recipe

 `~/cookbooks/workstation/recipes/default.rb`

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



## A Succinct Run List

*The cookbook only has one recipe that we care about. Could we set that up as the default?*

### Objective:

- ✓ Load the workstation cookbook's setup recipe in the default recipe
- ☐ Apply the workstation cookbook's default recipe
- ☐ Commit the changes

# GL: Apply the Cookbook's Default Recipe



```
$ sudo chef-client --local-mode -r "recipe[workstation]"
```

```
WARN: No config file found or specified on command line, using command line options.
```

```
Starting Chef Client, version 12.3.0
```

```
resolving cookbooks for run list: ["workstation"]
```

```
Synchronizing Cookbooks:
```

```
- workstation
```

```
Compiling Cookbooks...
```

```
Converging 4 resources
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Client finished, 0/4 resources updated in 3.300489827 seconds
```



## A Succinct Run List

*The cookbook only has one recipe that we care about. Could we set that up as the default?*

### Objective:

- ✓ Load the workstation cookbook's setup recipe in the default recipe
- ✓ Apply the workstation cookbook's default recipe
- ❑ Commit the changes



## GL: Commit Your Work

```
$ cd cookbooks/workstation
```

```
$ git add .
```

```
$ git commit -m "Update default recipe to include setup  
recipe"
```





## A Succinct Run List

*The cookbook only has one recipe that we care about. Could we set that up as the default?*

### Objective:

- ✓ Load the workstation cookbook's setup recipe in the default recipe
- ✓ Apply the workstation cookbook's default recipe
- ✓ Commit the changes

# GL: Return Home



```
$ cd ~
```



## Lab: Update the apache Cookbook

- ☐ Update the "apache" cookbook's "default" recipe to:  
Include the 'server' recipe from the 'apache' cookbook
- ☐ Run chef-client and locally apply the run\_list: "recipe[apache]"
- ☐ Commit the changes with version control

# Lab: The Default Recipe Includes the Apache Recipe

 `~/cookbooks/apache/recipes/default.rb`

```
#  
# Cookbook Name:: apache  
# Recipe:: default  
#  
# Copyright (c) 2016 The Authors, All Rights Reserved.  
  
include_recipe 'apache::server'
```

# Lab: Applying the apache Default Recipe



```
$ sudo chef-client --local-mode -r "recipe[apache]"
```

```
[2016-09-15T15:23:18+00:00] WARN: No config file found or specified on command line, using command line options.
```

```
Starting Chef Client, version 12.3.0
```

```
resolving cookbooks for run list: ["apache"]
```

```
Synchronizing Cookbooks:
```

```
- apache
```

```
Compiling Cookbooks...
```

```
Converging 0 resources
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Client finished, 0/0 resources updated in 3.310768509 seconds
```

## Lab: Commit Your Work

```
$ cd cookbooks/apache  
$ git add .  
$ git commit -m "Update default recipe to include server  
recipe"
```





## Lab: Update the apache Cookbook

- ✓ Update the "apache" cookbook's "default" recipe to:

Include the 'server' recipe from the 'apache' cookbook

- ✓ Run chef-client and locally apply the run\_list: "recipe[apache]"
- ✓ Commit the changes with version control



## Discussion



Why would you want to apply more than one recipe at a time?

What are the benefits and drawbacks of using "include\_recipe" within a recipe?

Do default values make it easier or harder to learn?



**CHEF**™

---