

Order of Execution

About me!



RajeshKumarIN



RajeshKumarIN



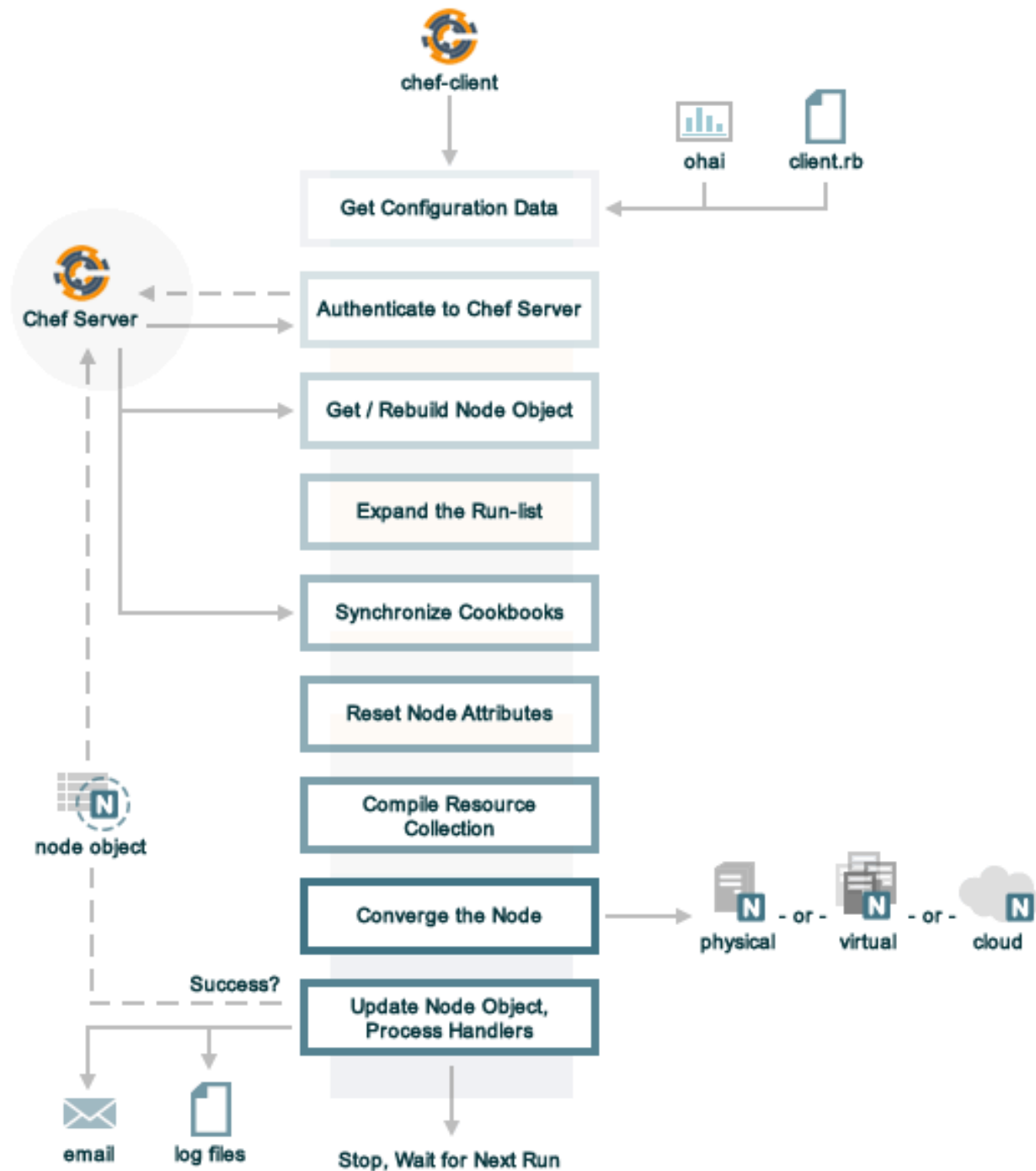
RajeshKumarIN



DevOps@RajeshKumar.XYZ



chef-client Run



Details

https://docs.chef.io/chef_client.html

Q 1

In what order is a chef cookbook executed

Answer

Recipes are run in the order they occur in the runlist. Only recipes that occur in the run list will be executed (plus any recipes that are included in place using `include_recipe`). Note that each recipe will only run once, even if it occurs multiple times in the runlist.

Attribute files from cookbooks are loaded in this order:

- Attributes of dependencies are loaded (i.e. cookbooks declared with `depends` in `metadata.rb`)
- `attributes/default.rb`
- all other attribute files in alphabetical order.

Note that all attribute files in a cookbook are loaded, irregardless of their name. Only attributes of cookbooks which either occur in the runlist or which are dependencies of cookbooks in the runlist will be loaded.

Q 2

Why are resources in an “include_recipe” step being skipped?

Or

Chef seems to be processing resources in a strange order, causing my build to fail.

Answer

It turns out this is a pretty normal (but under-documented) part of how Chef works: it compiles everything, and then starts running. Except some recipes (like Postgres) jump the queue to actually install at compile-time, using code like this:

```
action :nothing
end.run_action(:run)
```

The solution is that anything that needs to run before Postgres also needs to do this. Fortunately, newer versions of Build-essentials allow this. You set attributes on a role as follows:

```
name "myapp"
run_list(
  "recipe[build-essential]",
  "recipe[myapp]"
)
default_attributes(
  "build_essential" => {
    "compiletime" => true
  }
)
```

Q 3

Order of executing Chef recipes in a single cookbook?

Answer

There are two phases to a Chef client run.

- First is the resource compilation stage, where all the ruby is loaded, node attributes are resolved and a run list generated.
- Next is the execution phase, where the recipes in the run list are executed.

Q 4

How to Force the order of recipes with Chef?

Chef always executes recipes in the order you specify them. It will not arbitrarily reorder things. So if you want one recipe to be run before another, just load them in that order (e.g. In the `run_list`)

To further enforce this (at the expense of additional coupling between your recipes) you can also use `include_recipe` at the beginning of the second recipe to load the first one.

You have indeed two phases: the compile phase where your recipes are evaluated and the resource collection is built and the converge phase where this collection is executed one after another. Normally, resources are just added at the end of the resource collection during the compile phase. You can however influence this (e.g. by custom code or by using notifications).

If you need to install packages before, then you're on the right path using a recipe before in the run list (chef respects the order you give), but you have to enforce this package installation at compile time too.

```
package "libzma-dev" do
  action :nothing
end.run_action(:install)
```

Reference

<https://abigant.wordpress.com/2011/08/31/chef-recipe-order/>