



Configuration Management Tools for Experimenters: **Chef**

Max Ott
NICTA



Australian Government
**Department of Communications,
Information Technology and the Arts**
Australian Research Council

NICTA Members



Department of State and
Regional Development



NICTA Partners

What is Chef

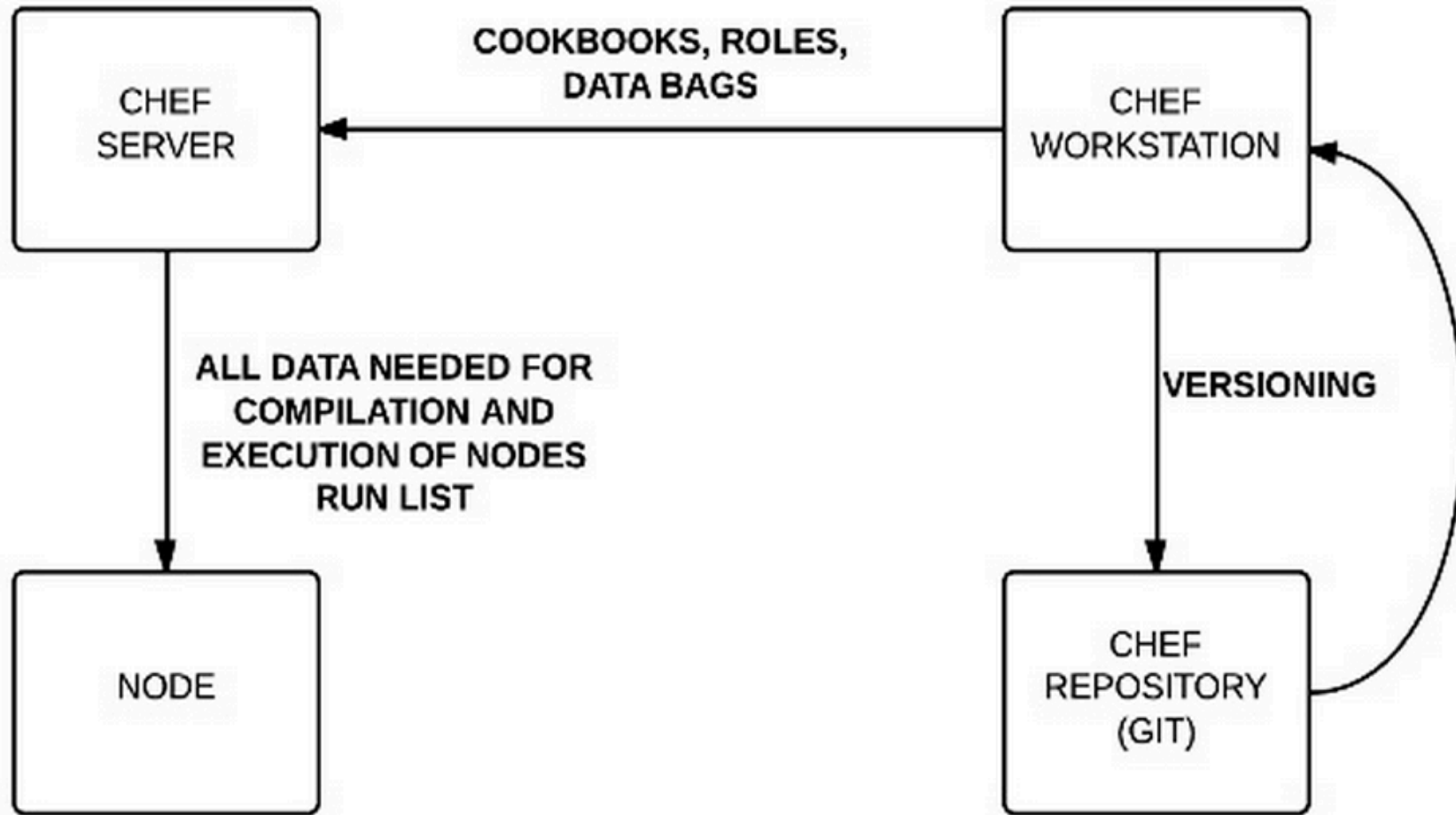
- Infrastructure as code
- Lots of “recipes” made and debugged by others
- Shallow abstraction over some OS differences

Infrastructure As Code!

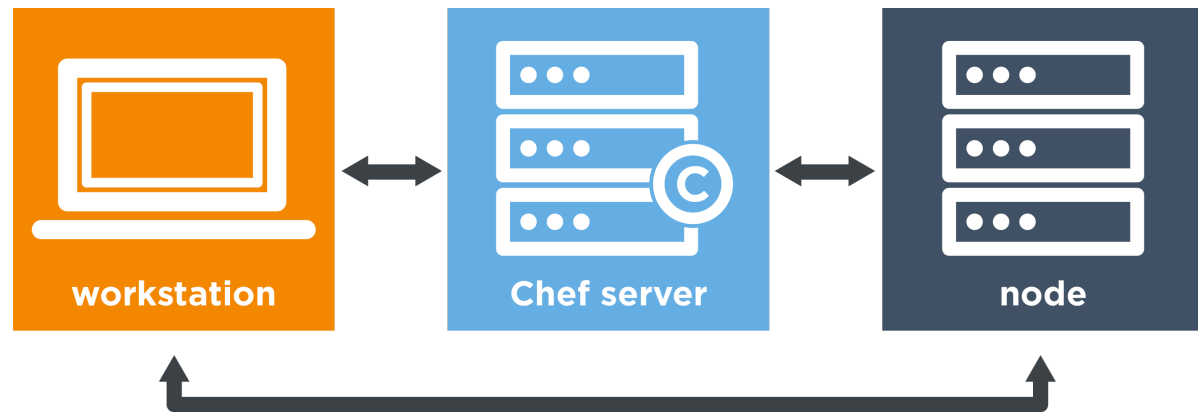
- Manage configuration as idempotent resources
- Put them together in recipes
- Track it like source code
- Configure your server

Adam Jacob, Co-Founder & CTO, Opscode

Workflow



Concepts



- Cookbooks & Recipes
- Roles
 - Web server, router, ...
- Environment
 - Development, testing, production, ..
- Attributes
- Resources

What are Attributes?

- An attribute is a specific detail about a node.
- Attributes are used to understand:
 - The current state of the node
 - What the state of the node should be after configuration
- Attributes are defined by:
 - The state of the node itself
 - Cookbooks (in attribute files and/or recipes)
 - Roles
 - Environments

What are Resources?

- directory
- remote_file
- user
- package (e.g. deb)
- template
- service
- execute

Example: Installing Apache

Declaring Intent

```
package 'apache2'  
  
service 'apache2' do  
  action [:enable, :start]  
end
```

Deploying

```
$ sudo chef-apply recipes/default.rb
```

```
Recipe: (chef-apply cookbook)::(chef-apply recipe)
```

- * apt_package[apache2] action install
 - install version 2.4.7-4.4 of package apache2
- * service[apache2] action enable (up to date)
- * service[apache2] action start (up to date)

Create your own Cookbook

Create Skeleton

```
$ chef generate cookbook my_webserver
```

```
Compiling Cookbooks...
```

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

... Create Skeleton

```
$ tree my_webserver/  
my_webserver/  
|____ README.md  
|____ files  
|____ metadata.rb  
|____ recipes  
|____ |____ default.rb  
|____ templates  
|____ tests  
...  
...
```

default.rb

```
package 'apache2'

service 'apache2' do
  action [:enable, :start]
end

file '/var/www/html/index.html' do
  content '<html>
  <body>
    <h1>hello world</h1>
  </body>
</html>'
end
```

Installing It

```
node$ sudo chef-apply recipes/default.rb
```

```
Recipe: (chef-apply cookbook)::(chef-apply reci
  * apt_package[apache2] action install (up to
  * service[apache2] action enable (up to date)
  * service[apache2] action start (up to date)
  * file[/var/www/html/index.html] action creat
    - update content in file /var/www/html/inde
    --- /var/www/html/index.html 2015-03-21 02:5
    +++ /var/www/html/.index.html20150321-20596
02:54:32.847783999 +0000
```

Knife: Remotely Applying Recipes

```
workstation$ knife bootstrap 54.175.28.83 \  
--ssh-user ubuntu \  
--sudo --run-list 'recipe[my_webserver]'
```


Summary

- Formally declaring your infrastructure is crucial!
- Infrastructure as code
- Lots of existing recipes
- Testable
- But everything has it's learning curve

- Chef may not be your tool, but you SHOULD use ONE



Configuration Management Tools for Experimenters: **Chef**

Max Ott
NICTA



Australian Government
**Department of Communications,
Information Technology and the Arts**
Australian Research Council

NICTA Members



Department of State and
Regional Development



NICTA Partners