CHEF CHEAT SHEET

Introduction

Chef is an open source and released by Opscode. It is a config management technology developed primarily to automate infrastructure provisioning. It is used to deploy and manage servers in-house or in the cloud. It uses ruby.

Features :

- No assumptions are made, it gets the current status of the machine via certain mechanisms.
- It is an excellent tool for integration with the cloud.
- As it uses ruby, it is fairly easy to get into for anyone with a basic development experience.

Terminology

- Node- managed machine. It executes the configuration for the node when the client runs.
- Client- an authorized user in the chef API
- Cookbook- a collection of recipes, resources, attributes and definitions to configure a service or application.
- Recipes- a list of resources to be added to a node. Written in ruby so it gives you control of anything you would do in ruby.

Files, Directories & Templates

It provides file, remote file and cookbook file to manage files and resource to manage directories.

Directories:

- Create, remove and manage directory permissions.
- Owner and group will be defaults for the client, usually root.
- Defaults can make cookbooks more concise although they shouldn't be confusing

Files:

- Allows you to manage and permissions and ownership of the files on the node.
- To retrieve a file from the URL or cookbook, use remote_file or cookbook_file resources.
- They have a backup attribute that defines how many backfiles exist upon changing content.

Templates:

- Supports text based config files using ERB.
- Ruby code is wrapped in brackets. Things that are not parsed are not executed as ruby code.
- Templates for complex configs can be created
- Just as in a cook_book file resource, source and node are set. Add variables attribute which assigns an array. The array will be made available in variable @ nameservers

Architecture

- It is a 3-tier client server model.
- Command line utilities are uploaded to the server and all nodes are registered with the server.

Chef workstation:

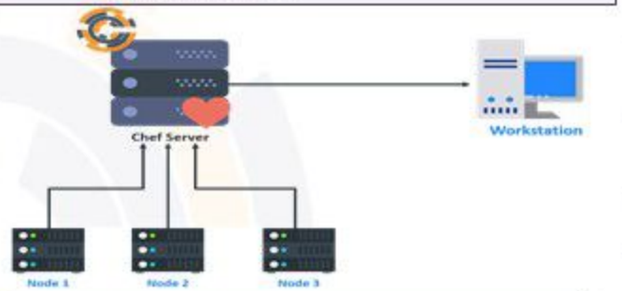
- configurations are developed and installed on local machine

Chef Server:

- It is the center of the Chef setup
- Config files are uploaded here.
- Some are hosted and some are built on-premise

Chef nodes:

- End machines managed by the server.
- Contains the client that sets communication between the server and node.



Resources

- They are ruby objects with code behind them to config the system.

LOG

- Prints the logging message at the specified level
- specify the level with level parameter.

Packages:

- Providers allow single package resource
- Default action for a package is install.
- Specifying the version is possible

Cookbooks

Knife cookbook create name[NAME] creates a new directory for the cookbook

Metadata.rb:

- It is converted to JSON when installed in the server and returns the properties.
- Most of its content are for human use and is displayed on the interface.

README.rdoc:

- Contains the documentation of the cookbook on how to use it and is useful when it is being shared with others.
- Markdown is supported while RDoc is the default.

Sharing Cookbooks:

- Since it is open source it allows sharing of cookbooks in the community.
- It supports downloading and sharing of cookbook to store, rate, and search cookbooks.

Commands

Kitchen commands:

Kitchen list, Kitchen create, Kitchen destroy, Kitchen login<instance name>

Other useful commands:

- Help command- knife -h
- Search for node which are Linux - knife search node "OS:linux"
- To run on node as convergence-chef-client
- To show environment - knife environment list -w
- To delete environment - knife environment delete dev
- To show knife environment - knife environment show dev

FUNCTION	COMMANDS
Get version	Knife --version
Create cookbook	Knife cookbook create <cookbook name>
Download cookbook	Knife cookbook download <cookbook_name > <version>
List cookbooks on the server	Knife cookbook list
Use chef supermarket	Knife cookbook site list
Getting list of all client nodes	Knife client list
Add recipe to runlist for node	Knife node run_list add module2 "recipe[apache]
Remove item from the runlist	Knife node run_list remove module2 "recipe[apache]

Run List

- Provides the recipes and the roles for the node.
- Ordered list is easier to understand and use. Using knife to get info

```
$ knife node show s1.mydomain.com
Node Name: s1.mydomain.com
Environment: _default
FQDN: s1.mydomain.com
IP: 1.2.3.4
Run List: role[common]
Roles: common
Recipes: chef-client, users::sysadmins, sudo
Platform: ubuntu 10.1
```

- To add more roles using knife:

```
$ knife node run_list add s1.mydomain.com "role[profit]"
run_list:
role[common]
role[profit]
```

Testing Cookbooks

Test the cookbook to make sure it doesn't break up on production

Steps:

- install cookbook:

```
example@localmach:~/chef-repo $ knife cookbook site install <cookbook name>
```

- Run the test commands:

```
example@localmach:~/chef-repo $ knife cookbook test VTest
checking ntp
Running syntax check on ntp
Validating ruby files
Validating templates
```

- Break something in it and test again:

```
example@localmach:~/chef-repo $ subl cookbooks/VTest/recipes/default.rb
...
[ node['ntp']['varlibdir']
node['ntp']['statsdir'] ].each do |ntpdir|
  directory ntpdir do
    owner node['ntp']['var_owner']
    group node['ntp']['var_group']
    mode 0755
  end
End
```

Limitations:

only runs a syntax check on ruby and .erb files

- Run ChefSpec and test kitchen to have a complete test done.

Components

- Knife: System admin tool used to interact with server to take cookbooks and custom config and loading them into the server. Bootstrapping certain servers also possible.
- Running knife: shows a list of commands that are supported.
- Chef client: runs on managed servers. It gathers info about itself, syncs the cookbooks and compiles the collection of resources and converges it with the machine state.
- Web UI: web based interface to allow to browse and edit cookbooks, nodes and clients
- Server/API: is the heart of the system, and exposes a REST API that is used by others. Manages the knife, web interfaces and nodes.