



Chef

Juan Antonio Beato Quiñones

Índice

- Introducción
- ¿Qué es Chef?
- Principales Ventajas
- Conceptos Básicos
- ¿Cómo lo uso?
 - Hosted Chef
 - Chef-Solo
 - Chef-Server y Chef-Client
- Administración
 - Chef-Server-Webui
 - Knife
- Cookbooks
- Recipes

Introducción

El crecimiento de internet cada día es mayor y al igual que la demanda en las empresas del sector.

Los CPDs cada vez son mayores y surgen los problemas de administración:

- Gran número de servidores.
- Configuración manual.
- Mayor posibilidad de error.
- Perdida de tiempo.

Introducción



¿Qué es Chef?

- Chef es un sistema de automatización de infraestructura desarrollado por Opscode que hace más fácil desplegar servidores y aplicaciones a cualquier ubicación física, virtual o en la nube, sin importar el tamaño de la infraestructura.
- Es software libre bajo licencia Apache 2.0.
- Esta escrito en Ruby y Erlang.
- Cada organización se compone de uno (o más) Workstations (estaciones de trabajo), un único servidor, y cada uno de los nodos que va a ser configurado y mantenido por Chef.

Principales Ventajas

- **Económica:** Chef ahorra dinero al usuario al permitir que negocio pueda soportar infraestructuras mas grandes y complejas con menos mano de obra. Hacemos mas con menos.
- **Eficiencia:** La automatización de una infraestructura con Chef significa no tener que volver a repetir la configuración.
- **Escalabilidad:** facilidad para el crecimiento de la infraestructura.

Conceptos Básicos

- **Node:** Es cualquier servidor, servidor virtual o instancia que este configurada para ser mantenido por chef-client.
- **Workstation:** Es una máquina que está configurada para usar knife, sincronizar con los repositorios y con el chef-server.
- **Role:** Es una forma de definir atributos y recetas que queremos que se ejecuten en uno o varios nodos como una sola forma de trabajo.
- **Cookbook:** Es la unidad fundamental de la distribución de configuración y políticas de Chef, define un escenario con todo lo necesario para realizar las instalaciones, configuraciones, etc..

Conceptos Básicos

- **Recipe:** Es el elemento de configuración más importante dentro del entorno Chef.
 - Están escritas en Ruby.
 - Debe definir todo lo que queremos configurar de un sistema.
 - Se almacenan en Cookbooks.
 - Pueden incluirse en otras recetas .
 - Puede depender de una o varias recetas.

Conceptos Básicos

- **Attributes:** Un atributo es un detalle específico de un nodo, se puede definir en un cookbook o recipe.
- **Run-list:** Es una lista en el que se pone los roles o recetas que queremos que se ejecute, se ejecutarán en el orden que pongamos.
- **Databags:** Es una variable global definida en un fichero JSON y que es accesible por un Chef-server y cargada en recetas.
- **Environs:** Es una forma de mapear u organizar el trabajo.
- **Ohai:** Es una herramienta que utiliza el servidor para detectar ciertas propiedades de cada nodo para después proporcionarlos al chef-client durante la ejecución de este.

¿Cómo lo uso?

Chef nos proporciona 3 opciones:

- Hosted Chef
- Chef-Solo
- Private Chef

Hosted Chef

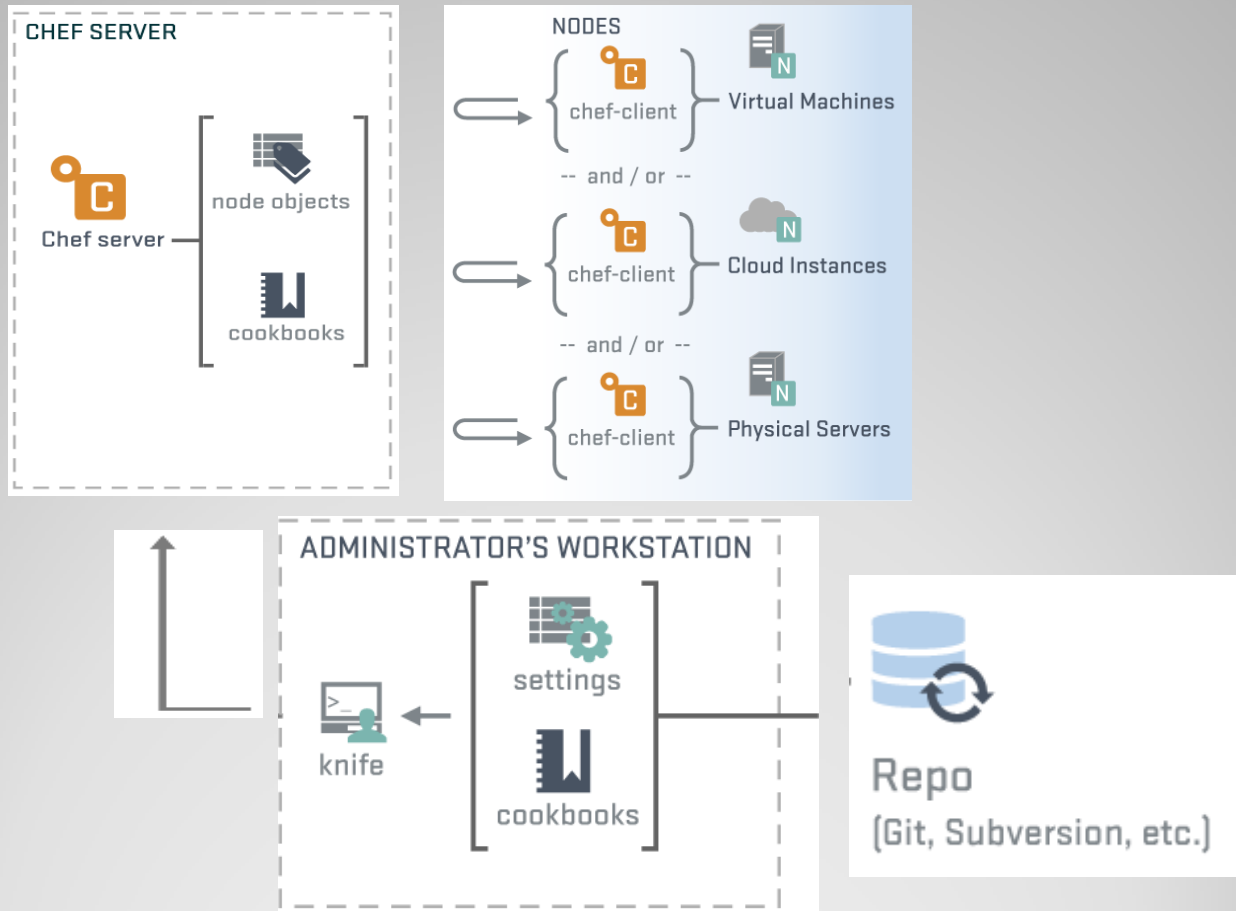
- Nos proporcionan un Chef-server al que nos conectaremos desde nuestra máquina con el par de claves que nos proporcionan en la web, nosotros tendríamos nuestro Workstation que sincronizará con el servidor.
- La versión FREE nos permite administrar hasta 5 servidores, estos son los precios:

	Free	Launch	Standard	Premium
Price per Month	Free	\$120	\$300	\$600
Nodes	5	20	50	100
Standard Support	–	✓	✓	✓

Chef-Solo

- Es una versión de código abierto de Chef que permite usar recetas en nodos que no tienen acceso a un Chef-Server. Se ejecuta localmente y necesita tener la recetas en la máquina y todas sus dependencias.
- Chef-Solo no incluye las siguiente funcionalidades:
 - Almacenamiento de datos de los nodos.
 - Distribución centralizada de Cookbooks.
 - Autenticación y Autorización
 - Una API centralizada que interactúa con los integrantes de la infraestructura

Chef-Server y Chef-Client

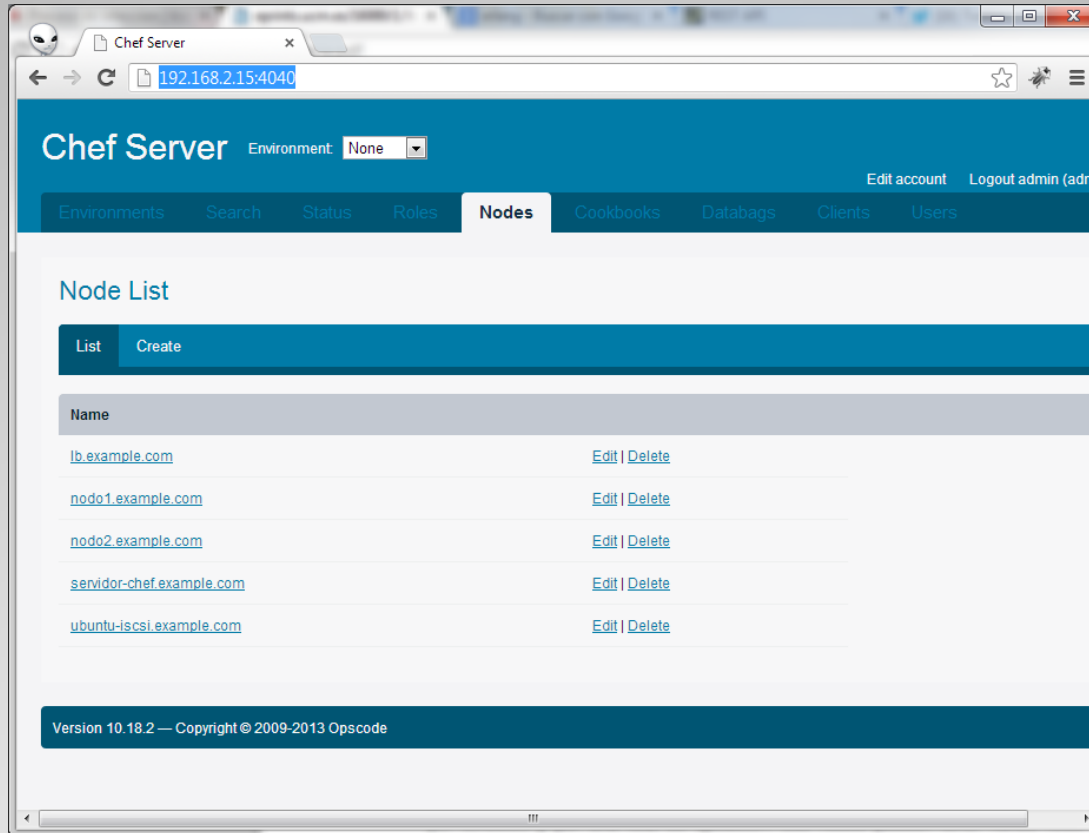


Administración

Para la administración de los nodos, cookbooks, etc... del Chef-Server podemos utilizar:

- **Chef-Server-Webui:** es la interfaz web para la administración del servidor. Está escrita en Ruby.
- **Knife:** es una interfaz por línea de comando integrada en Chef y que se comunica con Chef-Server a través de la REST API.

Chef-Server-Webui



The screenshot displays the Chef Server WebUI interface. The browser address bar shows the URL `192.168.2.15:4040`. The page header includes the text "Chef Server" and "Environment: None". A navigation menu at the top contains links for "Environments", "Search", "Status", "Roles", "Nodes" (which is the active tab), "Cookbooks", "Databags", "Clients", and "Users". On the right side of the header, there are links for "Edit account" and "Logout admin (adm)".

The main content area is titled "Node List". Below this title, there are two tabs: "List" (selected) and "Create". A table lists the nodes with their names and associated actions:

Name	Actions
lb.example.com	Edit Delete
nodo1.example.com	Edit Delete
nodo2.example.com	Edit Delete
servidor-chef.example.com	Edit Delete
ubuntu-iscsi.example.com	Edit Delete

At the bottom of the page, a footer bar contains the text "Version 10.18.2 — Copyright © 2009-2013 Opscode".

Knife

- Crear nodos, roles, cookbooks, etc...

```
knife (node|role|cookbook) create NAME
```

- Eliminar nodos, roles, cookbook, etc...

```
knife (node|role|cookbook) delete NAME
```

- Editar nodos, roles, cookbook, etc...

```
knife (node|role|cookbook) edit NAME
```


Knife

- Añadir cookbooks y recetas a roles o nodos

```
knife node run_list add (NODE|ROLE)
  'recipe[COOKBOOK::RECIPE]'
```

- Eliminar cookbooks y recetas a roles o nodos

```
knife node run_list remove (NODE|ROLE)
  'recipe[COOKBOOK::RECIPE]'
```

- Actualizar listado de cookbooks

```
knife cookbook upload -a
```

Cookbooks

Estructura de directorios de un cookbook:

- **Attributes:** Aquí definiremos los atributos del cookbook en un fichero ruby.
- **Files:** En este directorio pondremos los ficheros estáticos que queramos incluir en los nodos.
- **Templates:** En este directorio se almacenaran los archivos dinámicos que utilizaremos en el despliegue de la receta.
- **Recipes:** Directorio en el que se almacenan las recetas del cookbook.

Cookbooks

- Ubicación :

En Chef-Solo indicamos la ubicación de los cookbooks en el fichero solo.rb con las siguientes líneas:

```
file_cache_path "/root/chef-solo"  
cookbook_path  "/root/chef-repo/cookbooks"
```

En Chef-Server debemos indicarle a knife donde se encuentra los cookbooks, lo haremos en un fichero en la siguiente ruta, “/root/.chef/knife.rb”

```
cookbook_path  "/root/chef-repo/cookbooks"
```

Cookbooks

- Orden de ejecución:

En Chef-Solo indicamos el orden en el que queremos que se ejecuten los cookbooks en el fichero web.json:

```
{ "run_list": [ "recipe[apt]",  
  "recipe[soloword]" ] }
```

En Chef-Server el orden en el que se ejecutarán va definido en run_list del propio nodo o del role que le hayamos asignado.

Recipes

- Instalar un paquetes:

```
package "apache2" do
  action :install
end
```

- Iniciar, parar o reiniciar un servicio:

```
service "apache2" do
  action :stop
end
```

Recipes

- Ejecutar un comando:

```
execute "crear_run" do
  command "mkdir /var/run/haproxy/"
  not_if "find /var/run/haproxy"
  action :run
end
```

Recipes

- Llamar a otra receta:

```
include_recipe "balanceador::install"
```

- Crear un fichero en el nodo con uno de los ficheros estáticos:

```
cookbook_file "/etc/default/haproxy" do
  source "haproxy-default"
  owner "root"
  group "root"
  mode 00644
end
```

Recipes

- Crear un fichero en un nodo utilizando una de las plantillas:

```
template node['lb']['conf'] + '/haproxy.cfg' do
  source 'conf.erb'
  mode 00644
  owner 'root'
  group 'root'
  variables(
    :balan          => node['lb']['balan'],
    :ip1            => node['nodo1']['ip1'],
    :ip2            => node['nodo2']['ip2'])
end
```


Bibliografía

- <https://learnchef.opscode.com/>
- http://docs.opscode.com/chef_overview.html
- <http://wiki.opscode.com/display/chef/Installing+Chef+Server+on+Debian+or+Ubuntu+using+Packages>
- <http://wiki.opscode.com/display/chef/Installing+Chef+Client+on+Ubuntu+or+Debian>