

Herramientas de automatización

Resumen

Automatizar un proceso consiste en que este se realice por sí solo sin intervención directa del interesado¹. En este documento en primer lugar introducimos algunas razones de porqué es necesaria la automatización para a continuación analizar tres herramientas diferentes que sirven para automatizar las infraestructuras IT de las empresas. De cada una de estas herramientas realizamos brevemente una introducción explicando su función y las partes que la componen para a continuación mostrar su funcionamiento.

1.-Introducción.

Hoy en día cualquier empresa, sobre todo en aquellas relacionadas con el sector tecnológico, cuenta con varios ordenadores ya sean estos servidores o de escritorio. Además con el auge de la virtualización que permite disponer de las máquinas virtuales que uno quiera hace que mantener todo este entorno tecnológico funcionando correctamente sea bastante complejo.

Esta complejidad puede verse reflejada en los siguientes factores: una máquina hoy en día no ejecuta un solo servicio en ella generalmente tendrá varios ejecutándose simultáneamente los cuales tienen que estar correctamente configurados y actualizados, también hay que hacer frente a múltiples centros de datos sincronizados cuya información tiene que estar sin redundancias, nubes públicas, privadas o híbridas, aplicaciones con dependencias complejas...²

Un ejemplo de empresa con infraestructura IT compleja puede ser twitter³:

- Múltiples centros de datos
- Miles de máquinas ejecutando CentOS.
- Muchos tipos diferentes de servicios ejecutándose.

1.1.-¿Cómo podemos poner algo de orden en todo este caos que se genera?

Un administrador de sistemas en una empresa hoy en día se ve obligado a realizar scripts para intentar automatizar el proceso de mantener actualizado y correctamente configurados los sistemas que están a su cargo. Esto funciona si la empresa no es muy grande ni cuenta con una infraestructura IT compleja.

Una solución que aprovecha la automatización en su máxima potencia se llama "orchestration" que en español lo podemos traducirlo como orquestar. Este término se refiere a juntar cosas muy

¹ "automático - Real Academia Española. Diccionario Usual." 2015.

<<http://dle.rae.es/srv/fetch?id=4TO3M08>>

² "Ansible for Orchestration." 2015. <<http://www.ansible.com/orchestration>>

³ "How Twitter Uses Ansible." 2014. <<http://www.ansible.com/twitter>>

dispares, como los instrumentos de música de una orquesta o los diferentes servicios que ejecuta un servidor, y hacer que actúen como un conjunto coherente. Cada una de estas cosas dispares tiene su propia configuración, su propia partitura. Es la función del director de la orquesta hacer que todo arme un conjunto armonioso.⁴

Hoy en día existen diferentes herramientas que nos permiten conseguir esto, que ejercen de directores de orquesta: Rundeck, puppet, ansible, chef server, distelli, salt... y a continuación vamos a hablar un poco sobre ellos.

2.-Ansible

2.1.-Un poco de historia y de que es Ansible.

Ansible es una herramienta de código abierto (se encuentra alojada en GitHub⁵: <https://github.com/ansible/ansible>) para la automatización de las infraestructuras IT o tecnologías de la información. Tuvo su nacimiento alrededor de 2012. Fue creado por Michael DeHaan⁶ que es autor de Cobbler y coautor del framework Func, ha trabajado para RedHat y brevemente para Puppet. Su paso por estas empresas y los problemas que observó que tenían las empresas para las ha trabajado con las herramientas de automatización fue lo que dio lugar a la creación de Ansible. Recientemente RedHat ha anunciado la adquisición de Ansible.⁷

2.2-Como funciona ansible.

Ansible busca ser una herramienta fácil de usar, que sea potente y que no requiera de la instalación de ningún software especial en los nodos clientes: solamente es necesario instalar Ansible en una máquina que funcionará como punto de control y Ansible se encargará de controlar a las máquinas esclavo o servidores⁸ mediante el uso de ssh en sistemas Linux o WinRM y powershell en Windows⁹. Lo único que es necesario tener instalado en nuestros servidores es python 2.4 o superior y tener ssh instalado con la clave pública del ordenador que ejerce como punto de control dentro de las claves autorizadas por el ssh del servidor.

⁴ "Ansible for Orchestration." 2015. <<http://www.ansible.com/orchestration>>

⁵ "Community Information & Contributing — Ansible ..." 2014.

<<http://docs.ansible.com/community.html>>

⁶ "The Origins of Ansible." 2014. <<http://www.ansible.com/blog/2013/12/08/the-origins-of-ansible>>

⁷ "Red Hat to Acquire IT Automation and DevOps Leader Ansible." 2015.

<<https://www.redhat.com/en/about/press-releases/red-hat-acquire-it-automation-and-devops-leader-ansible>>

⁸ "Installation — Ansible Documentation." 2014. <http://docs.ansible.com/intro_installation.html>

⁹ <http://www.ansible.com/about>

<http://www.ansible.com/videos> → Quick start video 36 min -> 0:50 minutos

Sirve tanto para el desarrollo de aplicaciones para la empresa como para el manejo de la configuración de los sistemas, siendo capaz de orquestar complejas secuencias de eventos y puede manejar otras cosas aparte de servidores como Balanceadores de carga e incluso firewalls.¹⁰

Para empezar a usar ansible es en Linux tan fácil de instalar como cualquier otro paquete de nuestra distribución¹¹. En el caso de Debian basta con hacer :

```
sudo apt-get install ansible
```

para que este se instale.

Vamos a hacer una prueba de Ansible¹² utilizando tres ordenadores, el primero que actúe como punto de control y los dos restantes como servidores. Para ello en primer lugar es necesario generar la clave rsa en nuestra máquina cliente utilizando la orden:

```
ssh-keygen -t rsa, generándose .ssh/id_rsa.pub.
```

Después copiamos esta clave dentro de las claves autorizadas por los servidores utilizando:

```
ssh-copy-id -i .ssh/id_rsa.pub nombre_usuario@ip_servidor.13
```

Como podemos ver en la [Figura 1](#) Ansible cuenta con un archivo `/etc/ansible/hosts` donde se encuentran agrupados los hosts que va a tener el cliente bajo su control, añadimos a este archivo las ips de los ordenadores servidores en mi caso son 10.0.2.5 y 10.0.2.6. Y a continuación ejecutamos el comando:

```
ansible all -m ping
```

Este comando le dice a ansible que para todos los ordenadores dentro de `/etc/ansible/hosts` ejecute el módulo ping. Ahora más adelante habló más acerca de los módulos dentro de Ansible:

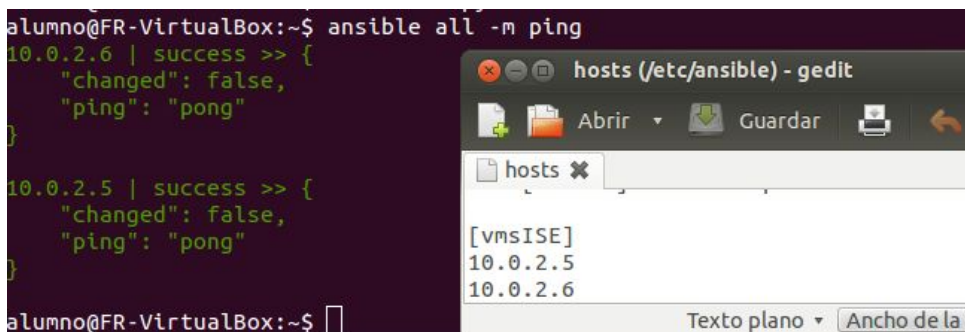


Figura 1: Salida de ejecutar el comando: `ansible all -m ping`, junto con una vista parcial del archivo `/etc/ansible/hosts`

Decir que dentro de `/etc/ansible/hosts` se permite hacer muchas cosas como especificar a que puerto tiene que comunicarse ansible para la ip de un servidor o conjunto de servidores concretos, permiten

¹⁰ <http://www.ansible.com/videos> → Quick start video 36 min

¹¹ "Installation — Ansible Documentation." 2014. <http://docs.ansible.com/intro_installation.html>

¹² "Getting Started — Ansible Documentation." 2014. <http://docs.ansible.com/intro_getting_started.html>

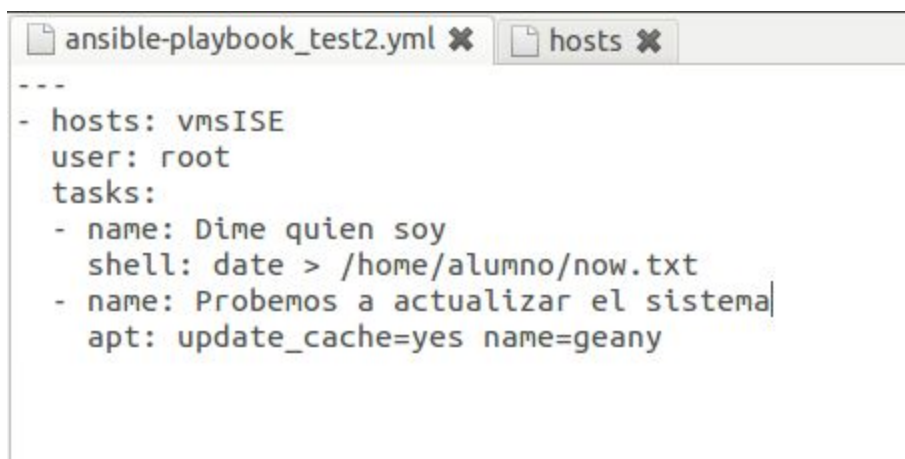
¹³ `man ssh-copy-id`, `man ssh-keygen`

agrupar los servidores en grupo y especificar otras muchas formas de acceso y de agrupación de los hosts servidores.¹⁴

Las configuraciones se llevan a cabo utilizando lo que ellos llaman *Playbooks*. Estos playbooks se realizan utilizando YAML que no es más que un lenguaje cuyo fin es traducir datos a un formato más fácil de leer por el ser humano y que a su vez sea comprensible por las máquinas. Cada Playbook está compuesto por *plays* este término se puede traducir por obra, Playbook= libro de obras de teatro, Play=obra. Cada play se compone de varias tareas y a su vez cada tarea llama a un módulo¹⁵. Ansible cuenta con más de 2000 módulos¹⁶ cada módulo se encarga de manejar algo en concreto por ejemplo:

- Manejo de paquetes: yum, apt.
- Ejecución de código en un ordenador remoto: command, shell.
- Manejo de servicios: service
- Manejo de archivos: copy, template.

En la [Figura 2](#) podemos ver un ejemplo de una play sencilla¹⁷:



```
---
- hosts: vmsISE
  user: root
  tasks:
    - name: Dime quien soy
      shell: date > /home/alumno/now.txt
    - name: Probemos a actualizar el sistema
      apt: update_cache=yes name=geany
```

Figura 2: Ejemplo de play sencilla sirve para actualizar los repositorios en los nodos servidores e instalar el paquete geany, la parte de Dime quien soy ejecuta el comando date en los nodos servidores y guarda la salida en now.txt

- name= define el nombre de la configuración de la play.
- hosts= define los servidores que tienen como objetivo la ejecución de esta play
- user= identifica al usuario que Ansible utilizará para ejecutar las tareas que vamos a definir más abajo.

¹⁴ "Inventory — Ansible Documentation." 2014. <http://docs.ansible.com/intro_inventory.html>

¹⁵ http://docs.ansible.com/ansible/playbooks_intro.html
<http://www.yaml.org/about.html>

<http://docs.ansible.com/ansible/glossary.html>

¹⁶ <http://www.ansible.com/videos> → Quick start video 36 min -> minuto 11

¹⁷ http://docs.ansible.com/ansible/apt_module.html
http://docs.ansible.com/ansible/playbooks_intro.html -> Quick start video 36 min -> minuto 2

En este ejemplo en primer lugar actualizamos los repositorios y después instalamos el paquete “geany”. Al igual que ocurre con /etc/ansible/hosts se puede hacer cosas bastante más complejas con los playbooks que este sencillo ejemplo¹⁸.

En la [Figura 3](#) podemos ver la salida de ejecutar este playbook:

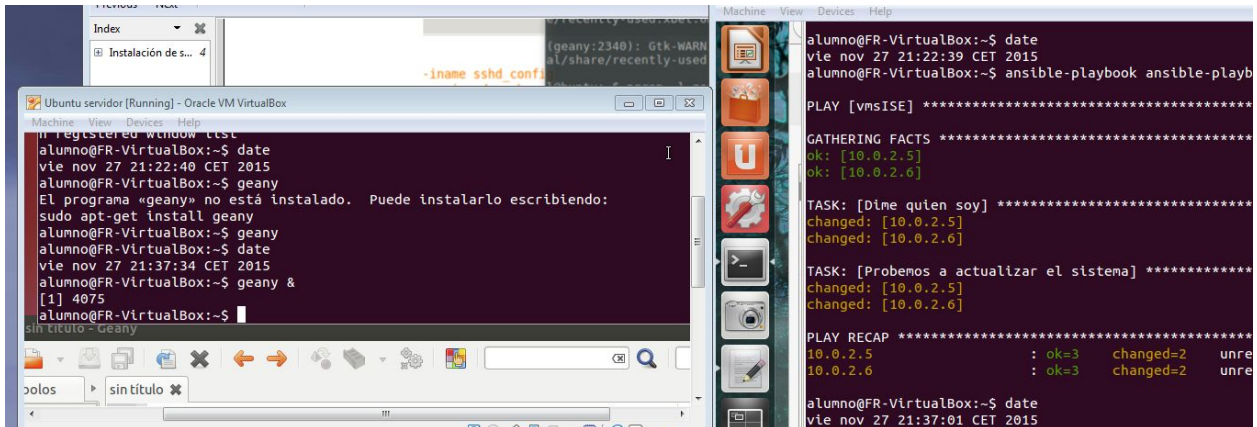


Figura 3: Salida de ejecutar el playbook que podemos observar en la imagen

En [Figura 3](#) tenemos a la derecha la máquina virtual que se encarga de ejecutar ansible y de controlar a los servidores y a la izquierda uno de los dos servidores, como podemos ver por la salida que ha dado ansible en la máquina virtual que ejecuta ansible todo a salido de manera correcta, probamos a ejecutar geany antes y después de ejecutar el playbook más arriba mostrado, podemos comprobar que se ha instalado el programa geany correctamente en el servidor.

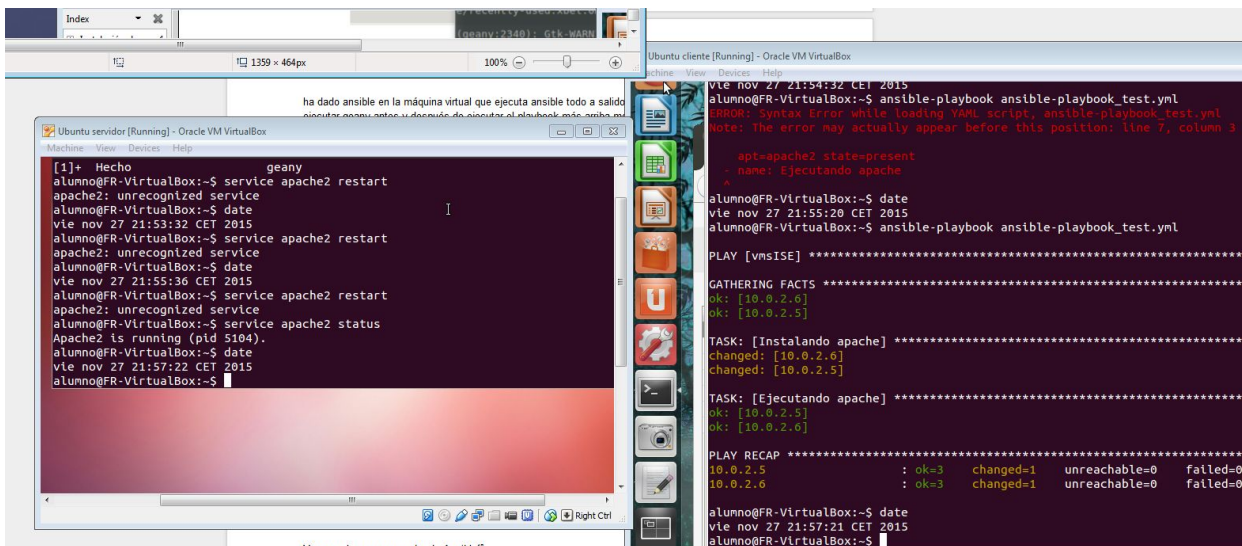
Ahora que le hemos cogido un poco el vicio vamos a intentar a instalar apache y que este se ejecute en los servidores para ellos ejecutamos la play que podemos ver en la [Figura 4](#):



Figura 4: Play que instala apache y que lo pone a funcionar en los nodos servidores

¹⁸ http://docs.ansible.com/ansible/playbooks_intro.html

Al ejecutarlo obtenemos la salida que podemos ver en la [Figura 5](#):



```
[1]+ Hecho geany
alumno@FR-VirtualBox:~$ service apache2 restart
apache2: unrecognized service
alumno@FR-VirtualBox:~$ date
vie nov 27 21:53:32 CET 2015
alumno@FR-VirtualBox:~$ service apache2 restart
apache2: unrecognized service
alumno@FR-VirtualBox:~$ date
vie nov 27 21:55:36 CET 2015
alumno@FR-VirtualBox:~$ service apache2 restart
apache2: unrecognized service
alumno@FR-VirtualBox:~$ service apache2 status
Apache2 is running (pid 5104).
alumno@FR-VirtualBox:~$ date
vie nov 27 21:57:22 CET 2015
alumno@FR-VirtualBox:~$

vite nov 27 21:54:32 CET 2015
alumno@FR-VirtualBox:~$ ansible-playbook ansible-playbook_test.yml
ERROR: Syntax Error while loading YAML script, ansible-playbook_test.yml
Note: The error may actually appear before this position: line 7, column 3

apt:apache2 state=present
  name: Ejecutando apache
alumno@FR-VirtualBox:~$ date
vie nov 27 21:55:20 CET 2015
alumno@FR-VirtualBox:~$ ansible-playbook ansible-playbook_test.yml

PLAY [vmsISE] *****
GATHERING FACTS *****
ok: [10.0.2.6]
ok: [10.0.2.5]

TASK: [Instalando apache] *****
changed: [10.0.2.6]
changed: [10.0.2.5]

TASK: [Ejecutando apache] *****
ok: [10.0.2.5]
ok: [10.0.2.6]

PLAY RECAP *****
10.0.2.5 : ok=3 changed=1 unreachable=0 failed=0
10.0.2.6 : ok=3 changed=1 unreachable=0 failed=0

alumno@FR-VirtualBox:~$ date
vie nov 27 21:57:21 CET 2015
alumno@FR-VirtualBox:~$
```

Figura 5: Salida de ejecutar el playbook que podemos ver en la [Figura 4](#)

Podemos observar como nuevamente no ha habido ninguna clase de problemas. Nuevamente la máquina virtual de la derecha es la máquina que controla y la de la izquierda el servidor, apache se instala correctamente en la máquina servidor.

3.-Chef server

Chef es una herramienta de automatización de las infraestructuras IT¹⁹ al igual que Ansible. El enfoque de Chef es parecido al de Ansible: hay un punto central que controla la configuración y el manejo de diferentes nodos servidores²⁰ pero a diferencia de Ansible en los nodos de Chef si hace falta instalar un programa específico para controlar a los nodos servidores: chef-client. Además Chef también utiliza algo parecido a los playbooks de Ansible ellos lo llaman *cookbooks* ("Libro de recetas")²¹. Estos cookbooks están compuestos de recetas ("recetas"). Las recetas y los cookbooks son lo que se usa para decirle al chef-client de cada nodo como tiene que estar configurado ese nodo.

3.1.-Funcionamiento de Chef

¹⁹ "An Overview of Chef — Chef Docs." 2014. <https://docs.chef.io/chef_overview.html>

²⁰ "All about Chef — Chef Docs." 2014. <<https://docs.chef.io/index.html>>

²¹ "All about Chef — Chef Docs." 2014. <<https://docs.chef.io/>>

En Chef podemos destacar tres actores principales, los nodos, chef server y las workstations. De los nodos ya hemos hablado más arriba. Chef server actúa como punto central de almacenamiento de la configuraciones que se aplican a los nodos. En él se guardan los cookbooks que son las políticas de configuración que se aplican a cada nodo. El chef-client instalado en cada nodo servidor le pide a Chef server detalles de la configuración que tiene que aplicar en su nodo y la lleva cabo.

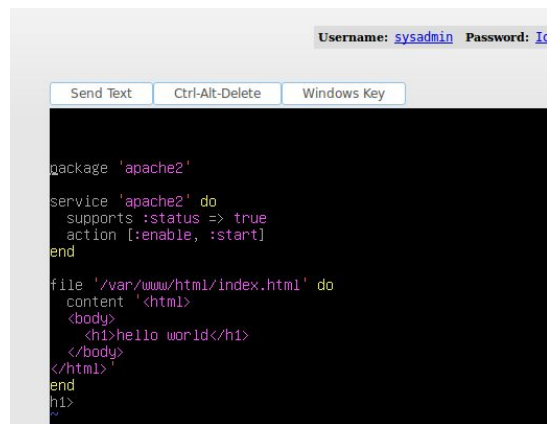
Una workstation es una máquina que está configurada para poder ejecutar varias aplicaciones de Chef tipo línea de comandos, también es donde los usuarios desarrollan los cookbooks y los testean para posteriormente subirlos al Chef server.

Por tanto para resumir: se desarrollan las configuraciones en las workstations para posteriormente subirlas al Chef server, estas configuraciones almacenadas en el Chef server son las configuraciones que el Chef server da a los chef-clients instalados en cada nodo cuando contactan con él. Los usuarios llevan a cabo el trabajo a través de las workstations que utilizan el Chef server para interactuar con los nodos.²²

3.2.-Probando Chef

En primer lugar intentamos realizar el mismo ejemplo que hicimos con ansible, es decir, instalar apache en cada nodo servidor pero en Chef encontramos múltiples problemas ya que es necesario una workstation, un chef-server en otra máquina y en cada servidor instalar el chef-client. Solo se puede instalar componentes de Chef (chef-server, chef-client, componentes de workstation) sobre máquinas de 64 bits y además para poder descargar el paquete de instalación de chef-server es necesario registrarse.²³ Tras todo esto y al comprobar que la documentación no es muy clara para lo que pretendíamos hacer como Chef nos permiten cómodamente a través de nuestro navegador probar a configurar un nodo servidor vamos probarlo ²⁴. Para ello vamos a instalar apache en el nodo remoto, vamos a ponerlo a funcionar una vez instalado y vamos a modificar su pagina de inicio.

Vamos a utilizar la recipe que podemos observar en la siguiente [Figura 6](#):



```
Username: sysadmin Password: 1o

Send Text Ctrl-Alt-Delete Windows Key

package 'apache2'

service 'apache2' do
  supports :status => true
  action [:enable, :start]
end

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

Figura 6: Recipe que usando Chef instala, inicializa apache2 en un nodo remoto

²² "An Overview of Chef — Chef Docs." 2014. <https://docs.chef.io/chef_overview.html>
<https://learn.chef.io/manage-a-node/ubuntu/>

²³ "Set up your Chef server - Learn to manage an Ubuntu node ..." 2015.
<<https://learn.chef.io/manage-a-node/ubuntu/set-up-your-chef-server>>

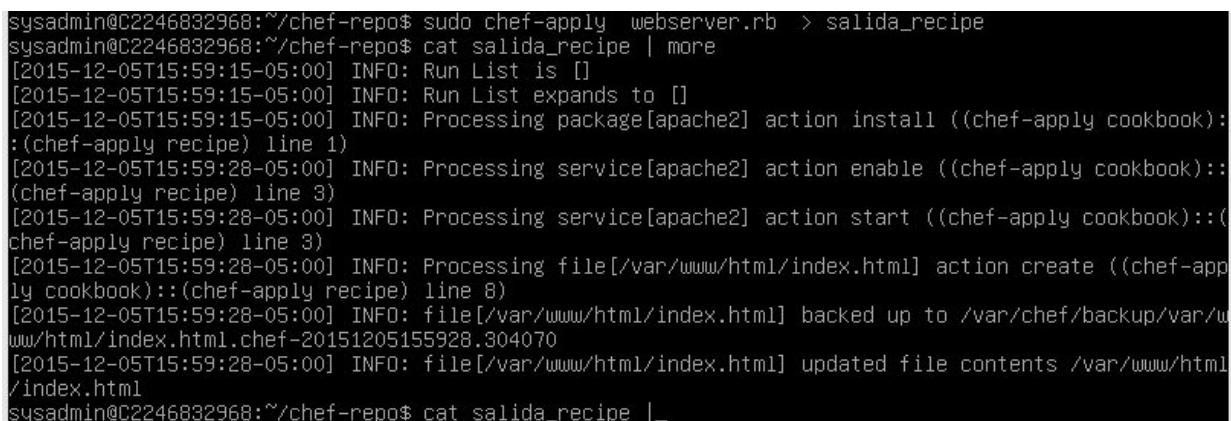
²⁴ "Learn the Chef basics on Ubuntu - | Learn Chef." 2015.
<<https://learn.chef.io/learn-the-basics/ubuntu>>

Igual como en Ansible utilizaban YAML como lenguaje para definir las configuraciones y las acciones sobre recursos en Chef se utiliza ruby. Antes de ponernos a explicar que hace el recipe de más arriba vamos a explicar un poco como funciona un recipe. Un recipe de Chef consiste básicamente en poner secuencialmente el nombre de diferentes recursos y la configuración deseado que tengan estos al final de la ejecución de la recipe. Un recursos puede ser un paquete, un archivo, un servicio... y una configuración sobre un recurso puede ser que un determinado paquete este instalado, que el contenido de un fichero sea algo en concreto o que un servicio este activo o no²⁵.

Al principio del recipe [Figura 6](#) se dice que el paquete apache2 tiene que estar instalado. La acción por defecto si apare package en un recipe es instalarlo por lo que con poner package 'apache2' hace que el paquete apache2 se instale si este no estuviera instalado²⁶. A continuación indicamos en el recipe que el servicio apache2 tiene que estar habilitado y funcionando y por último modificamos la página de inicio del servidor web, para ello creamos el fichero /var/www/html/index.html con el contenido indicado en el recipe.

Si a nuestro recipe le hemos puesto el nombre de webserver.rb para aplicarlo ejecutamos el siguiente comando:

```
sudo chef-apply webserver.rb
```

The image is a screenshot of a terminal window showing the output of the command 'sudo chef-apply webserver.rb'. The terminal output includes several lines of log messages from Chef, indicating the execution of various actions: installing the 'apache2' package, enabling the 'apache2' service, starting the 'apache2' service, and creating/updating the file '/var/www/html/index.html'. The output is as follows:

```
sysadmin@C2246832968:~/chef-repo$ sudo chef-apply webserver.rb > salida_recipe
sysadmin@C2246832968:~/chef-repo$ cat salida_recipe | more
[2015-12-05T15:59:15-05:00] INFO: Run List is []
[2015-12-05T15:59:15-05:00] INFO: Run List expands to []
[2015-12-05T15:59:15-05:00] INFO: Processing package[apache2] action install ((chef-apply cookbook)::
(chef-apply recipe) line 1)
[2015-12-05T15:59:28-05:00] INFO: Processing service[apache2] action enable ((chef-apply cookbook)::
(chef-apply recipe) line 3)
[2015-12-05T15:59:28-05:00] INFO: Processing service[apache2] action start ((chef-apply cookbook)::
(chef-apply recipe) line 3)
[2015-12-05T15:59:28-05:00] INFO: Processing file[/var/www/html/index.html] action create ((chef-app
ly cookbook)::(chef-apply recipe) line 8)
[2015-12-05T15:59:28-05:00] INFO: file[/var/www/html/index.html] backed up to /var/chef/backup/var/w
ww/html/index.html.chef-20151205155928.304070
[2015-12-05T15:59:28-05:00] INFO: file[/var/www/html/index.html] updated file contents /var/www/html
/index.html
sysadmin@C2246832968:~/chef-repo$ cat salida_recipe |_
```

Figura 7: Salida de hacer sudo chef-apply webserver.rb, en ella se puede observar lo que ocurre al ejecutar la recipe para instalar y habilitar apache2 en un servidor remoto)

En la salida que podemos observar en la [Figura 7](#) como en primer lugar para el paquete apache2 hay que realizar la acción de instalado, una vez que esto está completo se procede a habilitar apache2 para a continuación proceder a la ejecución del servicio apache2. Las últimas tres líneas se indica que se ha creado el archivo /var/www/html/index.html y que se ha actualizado su contenido, si hacemos un cat /var/www/html/index.html podemos observar que el contenido es el que le hemos indicado en el recipe. Como es un servidor remoto al cual no tenemos acceso desde nuestro computador personal no podemos comprobar con un navegador web si todo está correcto.

²⁵ "Configure a resource - Learn the Chef basics on Ubuntu ..." 2015.

<<https://learn.chef.io/learn-the-basics/ubuntu/configure-a-resource>>

²⁶ "Configure a package and service - Learn the Chef basics on ..." 2015.

<<https://learn.chef.io/learn-the-basics/ubuntu/configure-a-package-and-service>>

4.-Saltstack

Saltstack es una herramienta de automatización de las infraestructuras IT²⁷ al igual que Ansible y Chef. El enfoque de Saltstack es parecido al de Chef: hay un punto central que controla la configuración y el manejo de diferentes nodos clientes pero a diferencia de Ansible y Chef no poseen nada parecido a los playbooks o cookbooks. Para su instalación hay que saber diferencia entre los paquetes del servidor y los del cliente para garantizar un correcto funcionamiento. Saltstack destaca entre estos por su capacidad para manejar grandes cargas de información, y no sólo decenas sino cientos y hasta miles de servidores individuales rápidamente a través de una interfaz sencilla y manejable. Salt es compatible con multitud de Sistemas Operativos, incluidos sistemas Ubuntu, CentOS y Windows.

4.1-Instalación

Para la instalación vamos a ver un ejemplo de instalación en Windows y otro en Ubuntu.

En Windows resulta bastante sencillo ya que en la documentación oficial²⁸ tenemos los link de los ejecutables, tal y como se muestra en el [Figura 8](#). Instalaremos un el cliente, también llamado minion.

WINDOWS INSTALLER

Salt Minion Windows installers can be found here. The output of md5sum <salt minion exe> should match the contents of the corresponding md5 file.

Latest stable build from the selected branch:

x86: [Salt-Minion-2015.8.3-x86-Setup.exe](#) | [md5](#)

AMD64: [Salt-Minion-2015.8.3-AMD64-Setup.exe](#) | [md5](#)

Figura 8: Link de la instalación de Saltstack en Ubuntu.

Una vez descargado iniciamos el asistente de instalación, como se muestra en la [Figura 9](#)

²⁷ "Introduction to Salt - SaltStack." 2015. <<https://docs.saltstack.com/en/latest/topics/>>

²⁸ "Windows Installer - SaltStack." 2014.
<<https://docs.saltstack.com/en/latest/topics/installation/windows.html>>

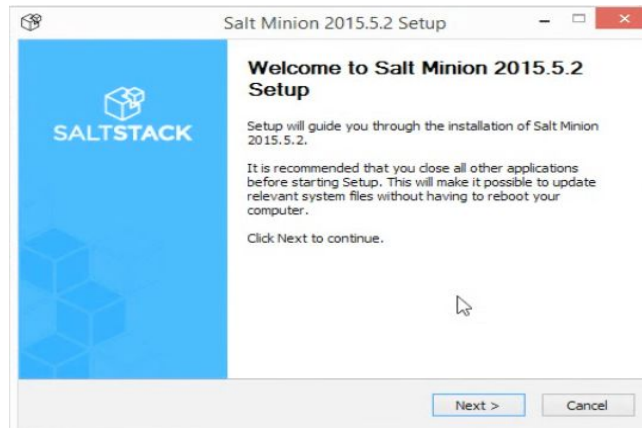


Figura 9: Asistente de instalación de Saltstack de Windows.

Continuamos con el proceso típico de instalación hasta llegar al punto de la [Figura 10](#), en donde Master IP or Hostname añadimos "salt" para que marquemos que se conecte al servidor, y en Minion Name el nombre que más nos agrade. Nosotros ponemos un nombre para identificarlo de demas Windows. Finalmente ejecutamos la Install para iniciar y a su vez finalizar la instalación.

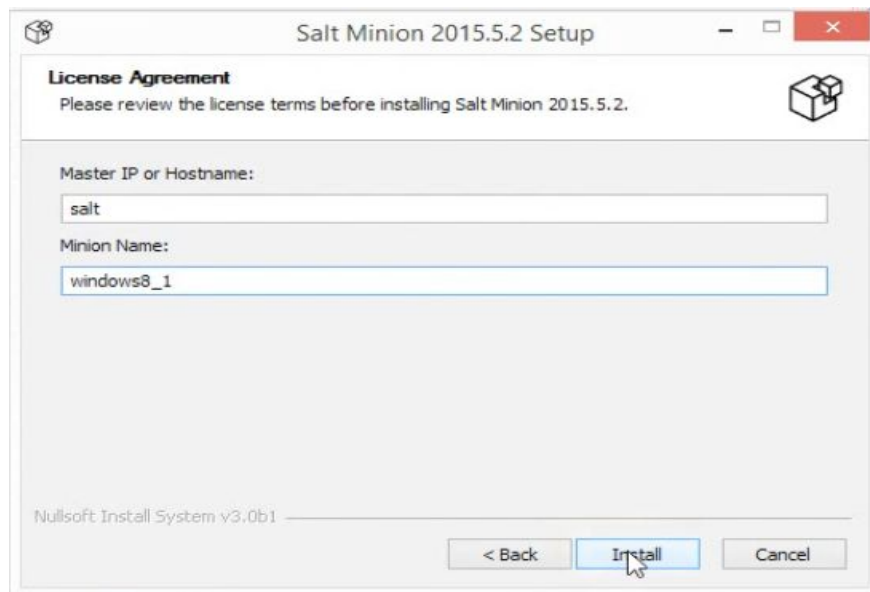


Figura 10: Asistente de instalación de Saltstack de Windows.

Para instalar un servidor o el cliente en Ubuntu²⁹ basta con ejecutar una terminal y seguir los siguientes pasos.

²⁹ "Ubuntu Installation - SaltStack." 2014.

<<https://docs.saltstack.com/en/latest/topics/installation/ubuntu.html>>

- Para añadir la clave del repositorio de Saltstack se usa el siguiente comando `wget -O - https://repo.saltstack.com/apt/ubuntu/14.04/amd64/latest/SALTSTACK-GPG-KEY.pub | sudo apt-key add -`.
- Seguidamente añadimos una línea a `/etc/apt.sources.list` con este contenido `deb http://repo.saltstack.com/apt/ubuntu/14.04/amd64/latest trusty main`.
- Ahora lo actualizamos con `sudo apt-get update`.
- El último paso es instalar los componentes de Salt.
 - `apt-get install salt-master`.
 - `apt-get install salt-minion`.
 - `apt-get install salt-syndic`.

Como antes hemos creado un minion ahora vamos a instalar un servidor.

4.2.-Probando Saltstack

Ahora toca una prueba de Saltstack³⁰, para ello en el master que instalamos Ubuntu ejecutamos el siguiente comando para comenzar con su funcionamiento: `salt master`.

Para la prueba hemos instalado los minion también en otras 2 máquinas virtuales otras versiones de Windows haciendo un total de 3 minion. En el tutorial anterior en la instalación de los minion habíamos configurado para que se conectara al servidor. Saltstack utiliza el cifrado AES para todas las comunicaciones entre el Master y el minion. Esto asegura que los comandos enviados a los minion no pueden ser manipulados. Pero antes la clave debe ser aceptada por el Master, con el comando `salt-key -L` para ver las claves de los minion que no han sido aceptadas, como muestra la [Figura 11](#).

```
root@salt-master:~# salt-key -L
Accepted Keys:
Denied Keys:
Unaccepted Keys:
windows7
windows8_1
windows_server2012
Rejected Keys:
root@salt-master:~#
```

Figura 11: Ejemplo de ejecución del comando `salt-key -L`

Para aceptarlas basta con ejecutar el comando `salt-key -A` y pulsando Y para aceptar, ahora veremos que están aceptadas como muestra la [Figura 12](#).

³⁰ "Configuring Salt - SaltStack." 2015.

<<https://docs.saltstack.com/en/develop/ref/configuration/index.html>>

```
root@salt-master:~# salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
windows7
windows8_1
windows_server2012
Proceed? [n/Y] Y
Key for minion windows7 accepted.
Key for minion windows8_1 accepted.
Key for minion windows_server2012 accepted.
root@salt-master:~#
```

Figura 12: Ejemplo de ejecución del comando salt-key -A para aceptar las claves.

Ahora vamos a probar un pequeño ejemplo, vamos a comprobar que hay comunicación entre el master y los minion. Para ello vamos a usar el comando test.ping, para ejecutar cualquier actividad en todos nuestros minion se usa la siguiente sintaxis salt '*' actividad en nuestro caso para el ping sería así salt '*' test.ping, como muestra la [Figura 13](#).

```
root@salt-master:~# salt '*' test.ping
windows_server2012:
  True
windows8_1:
  True
windows7:
  True
root@salt-master:~#
```

Figura 13: Ejemplo de ejecución del comando salt '*' test.ping para hacer ping a los minion

También se puede hacer ping sobre un minion, para ello en vez de marcar todo solo marcamos el nombre del nodo en concreto en vez de todos. Por ejemplo para el Windows 7 sería con el siguiente comando salt 'window7' test.ping.