

PROYECTO

ADMINISTRACIÓN DE SERVICIOS DE RED BAJO LINUX

GRUPO N° 5

SISTEMAS DE GESTIÓN CENTRALES CON SALTSTACK



INTEGRANTES:
Luis Ponce Cruz
Juan Salinas Elizalde

PARALELO 1
SEGUNDO TERMINO 2017 – 2018

Sistema de Gestión Central

Esto quiere decir que es una acción o decisión de concentrar elementos, que son distintos en un centro común, que será el encargado de dirigirlos.

Esto nos ofrece como ventaja:

- Tener un mayor control.
- Ser uniforme en los procedimientos y decisiones.
- Sencillez en el control y evaluación.
- Se libra desperdicio y duplicidad de tareas.

De la misma manera tenemos desventajas de poseer un sistema de gestión central que son:

- Demora en la toma de decisiones.
- Mayor dependencia en la parte central.
- Ineficiencia en el uso de los recursos.

Pero no poseer un sistema central de gestión puede ocasionar:

- Carencia de uniformidad de las decisiones.
- Tener una comunicación deficiente.
- Escasez de coordinación.
- Competencia negativa (si el sistema no se encuentra bien administrado)

SaltStack (Intelligent orchestration for the software-defined data center)

SaltStack desarrolla software de orquestación inteligente para ayudar a las organizaciones empresariales a proteger y gestionar de forma mas eficiente todos los aspectos del centro de datos definido por software.

El cual es único en proporcionar automatización impulsada por eventos para una configuración escalable y eficiente.



SALTSTACK

En nuestro proyecto vamos a utilizar la herramienta SaltStack que es un software open source en la que implementaremos para una gestión remota de servidores y hacer conexión con una base de datos MySQL.

Lo cual nos hace mas sencillo y dinámico gestionar cualquier infraestructura que deseamos implementar, ya sea grande o pequeña, es decir formada por pocos servidores o cientos de ellos. Una característica principal de Salt, que es posible la comunicación con cualquier servidor en cuestión de segundos.

Instalación de Salt (Salt-master y Salt-minion)

1. Debemos realizar la creación de dos máquinas virtuales en el software de virtualización VirtualBox en cual las utilizaremos para servidores con el sistema operativo CentOS 7, estas dos máquinas son llamadas Centos 7 1 y Centos 7.2, en el cual las dos tendrán idénticas características, como: una memoria base de 1500 MB, puerto SATA 0 de 8 GB como se muestran en la Figura I.

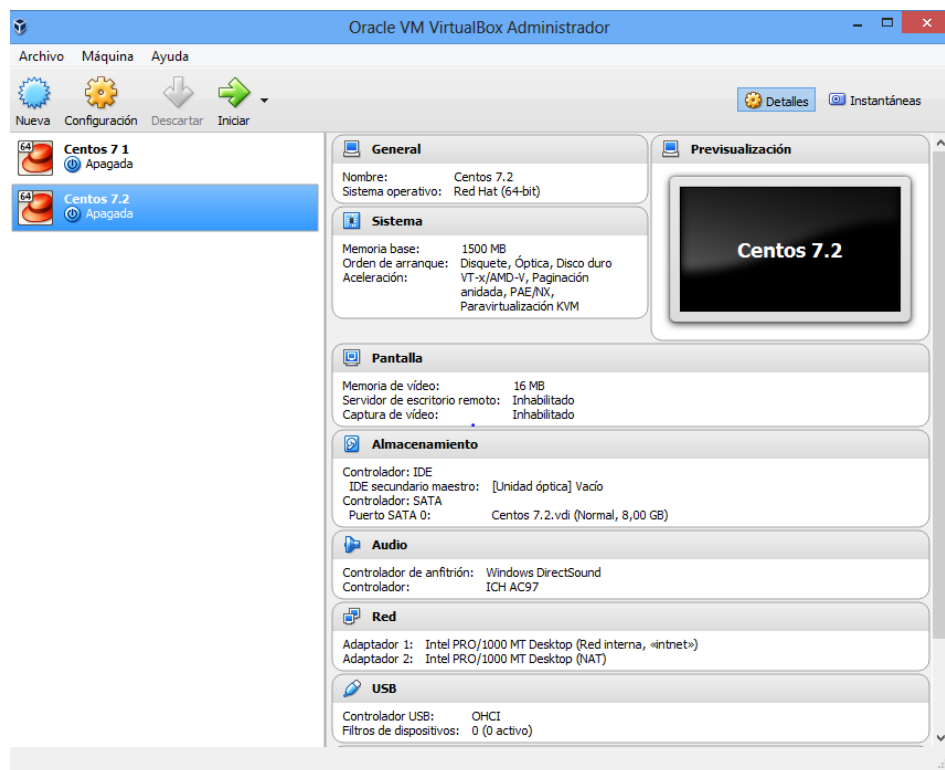


Figura I. Creación de las máquinas virtuales con sistema operativo CentOS 7 con igual características

2. Estas dos máquinas tendrán dos adaptadores de red, el primer adaptador esta conectado a NAT y el segundo adaptador esta conectado a una red interna (las dos maquinas deben tener el mismo nombre de la red interna) esto se realiza en la configuración de cada máquina virtual en el apartado de red, como se ve en la Figura II.

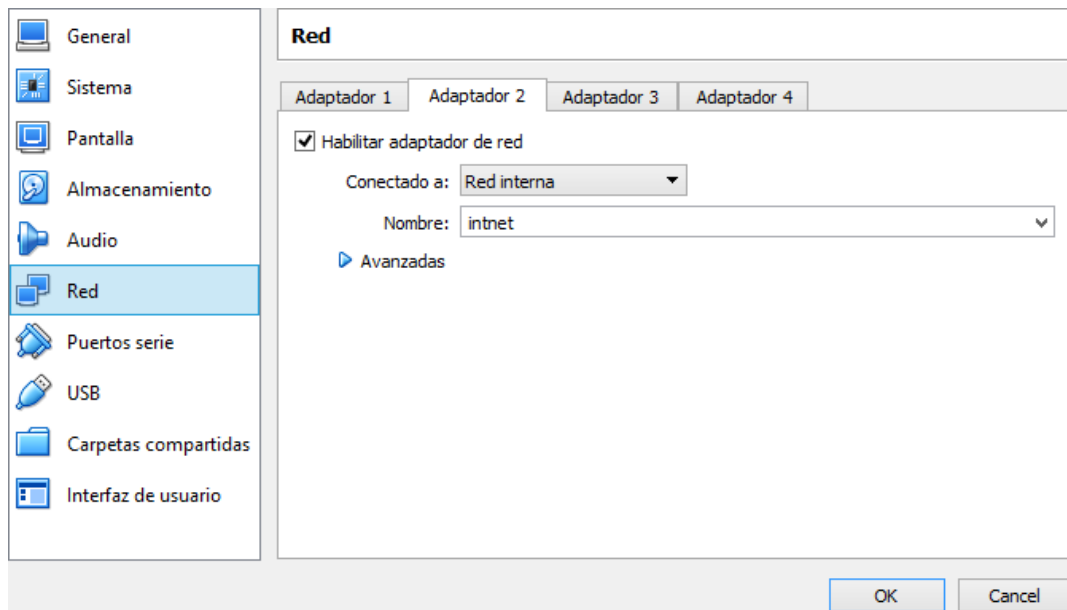


Figura II: Adaptador de red de las dos máquinas virtuales creadas, conectadas a una red interna con nombre intnet.

3. Tener en cuenta que el Salt-master como su nombre lo indica es el maestro es decir que es la parte central del sistema, debe estar en la misma red que el Salt-minion, que es el esclavo, al cual se lo va administrar de manera remota. En este adaptador colocamos IP estática en cual para el servidor **master 192.168.1.2** (máquina Centos 7 1) y para nuestro servidor **minion 192.168.1.3** (máquina Centos 7.2).
4. Luego comenzamos a descargar los paquetes para la instalación de Salt, estos paquetes se deben descargar en las dos máquinas virtuales (Centos 7 1 y Centos 7.2). Se utiliza el siguiente comando:

```
#curl -L https://bootstrap.saltstack.com -o install_salt.sh
```

En el servidor Salt-Minion

5. Para instalar el salt-minion en nuestro servidor debemos colocar:

```
#sudo sh install_salt.sh -P
```

6. Luego lo que debemos configurar en nuestro salt-minion para que pueda encontrar su salt-master, para lo cual debemos dirigirnos al fichero:

```
#vi /etc/salt/minion
```

7. Modificar la línea del master el cual debemos colocar la IP del mismo (192.168.1.2) y además podemos modificar como aparecerá nuestro minion en el master en el cual llamaremos minion2.

master: 192.168.1.2
Id: minion2

8. Una vez realizado estos cambios debemos reiniciar el servicio de salt-minion con el siguiente comando:

```
#service salt-minion restart
```

En el servidor Salt-Master

9. Para realizar que nuestro servidor tenga la función de salt-master debemos colocar el siguiente comando:

```
#sudo sh install_salt.sh -P -M
```

10. Para que se pueda conectar de una manera más rápida debemos habilitar los puertos, el salt sale por defecto en los puertos 4505 y 4506, habilitamos el firewall para agregar estos puertos.

```
#sudo systemctl enable firewalld  
# sudo firewall-cmd --zone=public --add-port=4505/tcp  
#sudo firewall-cmd --zone=public --add-port=4506/tcp
```

11. Para configurar nuestro salt-master debemos dirigirnos al fichero:

```
#vi /etc/salt/master
```

12. Y modificar la línea de interface y colocar la IP que está utilizando (192.168.1.2) para confirmar a los minion que de esa manera lo encontraran.

Interface 192.168.1.2

13. En nuestro servidor master debemos encontrar la clave de nuestro salt-minion en el cual debemos enlistar los minion que estén buscando al master:

```
#salt-key -L
```

14. Las claves minions debe presentarse con el texto “Unaccepted Keys” y algo similar a lo siguiente:

```
Accepted Keys:  
Unaccepted Keys:  
Minion2  
Rejected Keys:
```

15. Debemos hacer que nuestro salt-master acepte a nuestro salt-minion lo cual le damos el permiso:

```
#salt-key -a 'minion2'
```

Lo cual aparecerá de la siguiente manera:

```
The following keys are going to be accepted:
```

```
Unaccepted Keys:
```

```
Minion2
```

```
Proceed? [n/Y] y
```

```
Key for minion minion2 accepted.
```

16. Una vez realizados los pasos anteriores ya tenemos conectados nuestro salt-minion con el salt-master y podemos realizar otros comandos como determinar si en realidad tenemos conectados nuestros servidores como:

```
#salt '*' test.ping
```

Lo cual nos mostrara de la siguiente manera:

```
Minion2:
```

```
True
```

17. Para obtener las IP de los servidores minion utilizamos el siguiente comando:

```
#salt 'minion2' network.ip_addrs
```

El cual la salida que presentará será:

```
Minion2:
```

```
- 192.168.1.3
```

18. También es posible realizar que el salt-master funcione como salt-minion esto se realiza como en los pasos del servidor salt-minion, en la Figura III se muestra la conexión de dos servidores minion (192.168.1.3 y 192.168.1.2) realizado en nuestro proyecto.

```
[root@localhost ~]# salt '*' test.ping
minion2:
    True
minion1:
    True
[root@localhost ~]# █
```

Figura III. Muestra los minions que posee el master y la conexión. 192.168.1.2 es minion y master, muestra que esta conectado mientras que el 192.168.1.3 se encuentra desconectado.

Realización de fórmulas para gestión remota.

En nuestro proyecto realizamos varias fórmulas, a esto se le denomina a los archivos de configuración en el cual indicaremos las distintas tareas que debe efectuar, para poder configurar nuestro servidor con herramientas como mysql, http y ftp. Todas estas configuraciones se realizan en el salt-master.

1. Todas las formulas creadas las guardamos en fichero creado dentro del master en la ruta /srv/salt/
2. Debemos crear un archivo .sls en el cual colocaremos lo que el salt-master quiere que realice los salt-minion.

```
#vi /srv/salt/httpd.sls
```

3. En ese archivo añadimos las siguientes líneas:

```
httpd:
  pkg:
    - installed
```

Lo cual nos indica la primera línea (httpd:) el “ID Declaration” lo cual es la etiqueta de la declaración, y es utilizado para el nombre del paquete, el nombre debe ser igual al nombre real del paquete utilizado por el gestor de paquetes (yum).

La segunda línea (pkg:) recibe como el nombre de “Declaración de Estado” y hace referencia al nombre de la función que deseamos ejecutar. En nuestro formula utilizamos la de declaración de estado pkg. Se puede encontrar una lista completa de estados en el siguiente enlace <https://docs.saltstack.com/en/latest/ref/states/all>

Y la tercera línea (- installed) es conocida como “Declaración de funciones” es la función que deseamos ejecutar. En nuestro caso será – installed.

4. Luego aplicamos a nuestro servidor de la siguiente manera y se muestra el resultado como en la Figura IV:

```
#salt '192.168.1.2' state.sls httpd
```

```
[root@localhost ~]# salt '192.168.1.2' state.sls httpd
192.168.1.2:
-----
      ID: httpd
  Function: pkg.installed
    Result: True
  Comment: All specified packages are already installed
  Started: 22:22:19.272462
Duration: 34003.606 ms
  Changes:

Summary for 192.168.1.2
-----
Succeeded: 1
Failed:    0
-----
Total states run:    1
Total run time:   34.004 s
[root@localhost ~]#
```

Figura IV. Al ejecutar el comando para instalar el paquete httpd y lo que muestra por consola.

Hacer conexión con MySQL

Salt posee modulo para proporcionar compatibilidad con MySQL, en el cual debe tener instalado MySQLdb Python.

1. Se instala para poder realizar conexión:

```
# wget http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
# yum install mysql57-community-release-el7-9.noarch.rpm
```

En el servidor salt-minion tengo una base de datos con el nombre “sucursal1” con 6 tablas como se muestra la Figura V.

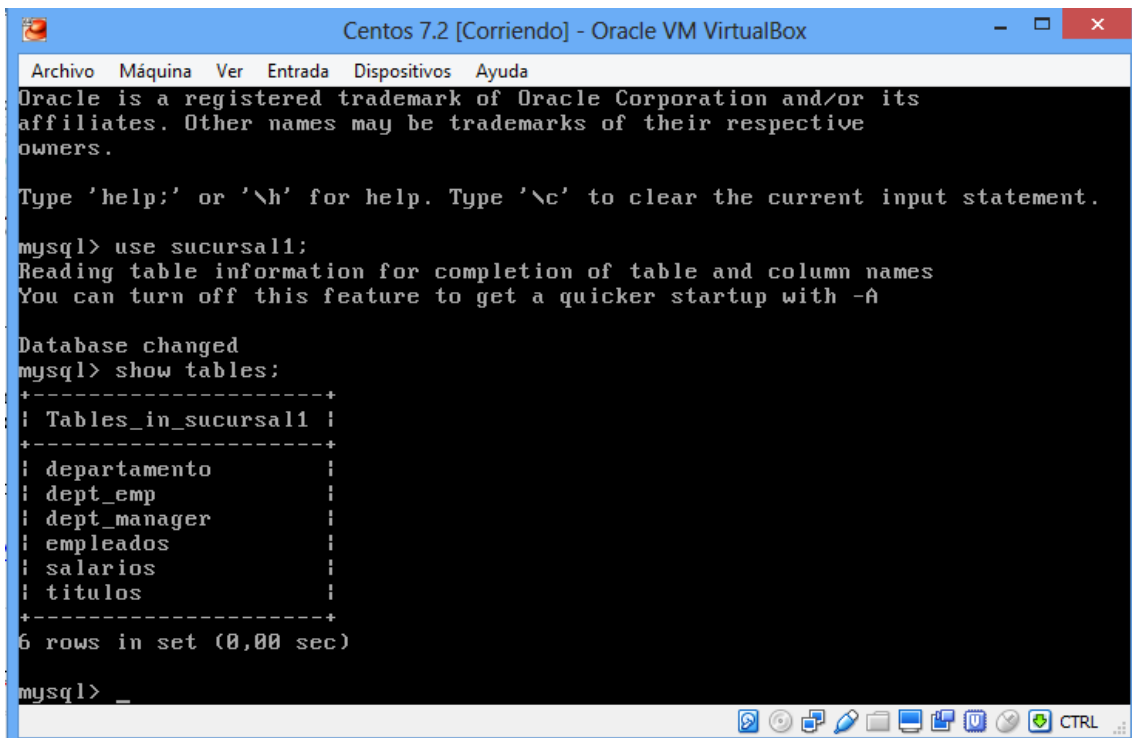


Figura V. Se muestra el servidor salt-minion (Centos 7.2) el cual está en la base de datos sucursal1 con sus diferentes tablas.

2. Para poder crear una base de datos al servidor remoto que sería una sucursal de la empresa:

```
#salt 'minion2' mysql.db_create 'sucursal1'
```

3. Para poder observar las bases de datos, como observamos desde el salt-master, se muestra en la Figura VI :

```
#salt 'minion2' mysql.db_list
```

```

[root@localhost ~]# salt minion2 mysql.db_list
minion2:
  - information_schema
  - mysql
  - performance_schema
  - prueba
  - sucursal1
[root@localhost ~]# █

```

Figura VI. Al ejecutar el comando descrito, se observa la lista de base de datos que posee el servidor salt-minion

4. Para poder borrar una base de datos:

```
#salt 'minion2' mysql.db_remove 'sucursal1'
```

5. Para poder observar las tablas de la base de datos:

```
#salt 'minion2' mysql.db_tables 'sucursal1'
```

Como se muestra en la Figura VII se tiene todas las tablas pertenecientes a la base de datos sucursal1

```
[root@localhost ~]# salt minion2 mysql.db_tables 'sucursal1'
minion2:
  - departamento
  - dept_emp
  - dept_manager
  - empleados
  - salarios
  - titulos
[root@localhost ~]# █
```

Figura VII. Al ejecutar el comando anteriormente descrito, se observa la lista de tablas que tiene la base de datos sucursal en el servidor salt-minion.

6. Para poder saber que contiene las tablas de la base de datos 'sucursal1' en este particular comando la tabla empleados; como se muestra en la figura VIII.

```
#salt '*' mysql.query mydb "SELECT * from empleados"
```

```
[root@localhost salt]# salt minion2 mysql.query sucursal1 'select * from empleados'
minion2:
  -----
  columns:
    - emp_no
    - nombre
    - apellido
    - cumpleaños
  query time:
    -----
    human:
      1.4ms
    raw:
      0.00141
  results:
    |_
    - 1001
    - Andres
    - Figueroa
    - 1964-09-02
```

Figura VIII. Como resultado del select de la tabla empleados y de la base de datos sucursal1 usando el comando de salt.

7. Para poder modificar las tablas podemos usar los siguientes comandos; son equivalentes al INSERT y DELETE en MySQL

```
#salt 'minion2' mysql.query sucursal1 'INSERT into empleados  
values (1017,"Luis","Ponce","1992-08-22")'  
#salt 'minion2' mysql.query sucursal1 'DELETE from empleados where  
emp_no = 1017'
```

8. Poder crear un usuario en MySQL para el servidor salt-minion; como observamos en las figuras IX y X en el cual muestra colocar el usuario y observar en el salt-minion ingresar con ese usuario y clave al MySQL respectivamente:

```
#salt 'minion2' mysql.user_create 'luis' 'localhost' '12345'
```

```
[root@localhost ~]# salt 'minion2' mysql.user_create "luis" "localhost" "12345"  
minion2:  
True  
[root@localhost ~]#
```

Figura IX. Como se observa con el comando hemos creado un usuario para el salt-minion (minion2). El cual envía como respuesta que se ha hecho tal operación.

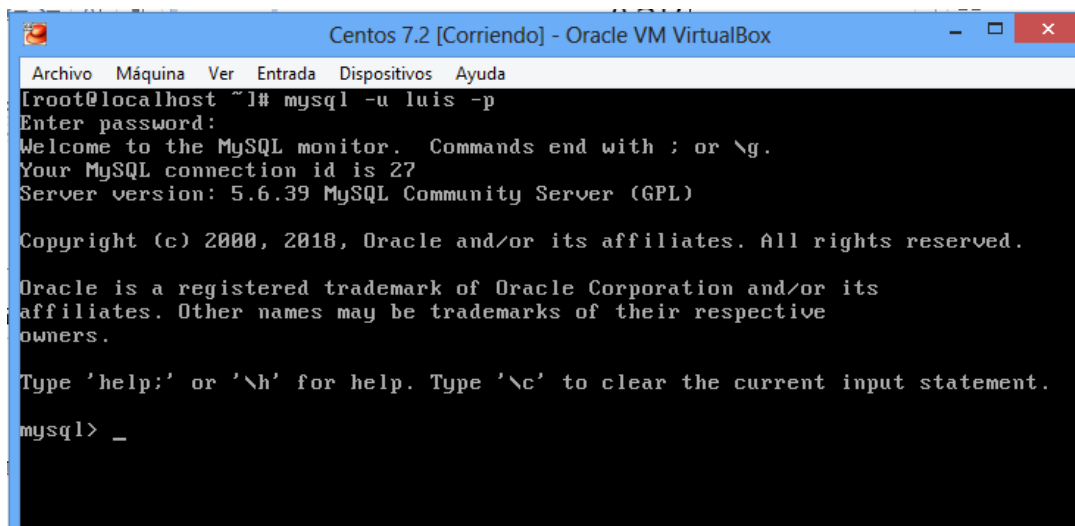


Figura X. Se comprueba que al ingresar en MySQL con el usuario creado anteriormente tiene efecto en salt-minion (minion2)

Conclusiones

Tenemos una herramienta que es muy útil en el cual nos permite realizar un sistema centralizado en el que podemos gestionar nuestros servidores, ya sea como en nuestro proyecto de manera remota instalando servicios y activándolos o existen muchas otras declaraciones de estado como nombramos en un enlace, el cual tiene una gran variedad como configurar apache, configuración del sistema, actualización de dns, configuración de firewall y etc. Lo interesante es que es Open Source y de que los servidores se conectan de una manera rápida y no importa si tienes muchos servidores esclavos. Como observamos se pueden formar formulas y además también se puede realizar conexión con la base de datos MySQL sin ingresar a ella y además remotamente revisar los datos de las tablas de servidores minions y poder ingresar o eliminar de las tablas y hasta crear usuarios en ese servidor para las bases de datos.

Referencias

<https://repo.saltstack.com>

<https://docs.saltstack.com/en/latest/topics/tutorials/walkthrough.html>

<https://docs.saltstack.com/en/latest/ref/states/all/index.html>

<https://docs.saltstack.com/en/latest/ref/states/all/salt.states.pkg.html#module-salt.states.pkg>

<https://docs.saltstack.com/en/latest/ref/modules/all/salt.modules.mysql.html>