

Escenario OSPF

Alejandro Rodríguez Rojas

Índice

1	Introducción.....	3
2	Definición del Escenario.....	5
3	Configuración de OSPF.....	7
4	Verificación del Route.....	8
5	Conclusión.....	9

1 Introducción

Vamos a realizar un ejercicio dónde vamos a enrutar dinámicamente un escenario dado por el profesor, lo primero que tenemos que hacer es descargarlo, extraerlo en una carpeta y usar el siguiente comando:

`vagrant up`

*Para ello debemos ir a la página oficial de vagrant y descargarnos el paquete.

```
alexrr@pc-alex:~/vagrant$ vagrant up
Bringing machine 'pc1' up with 'virtualbox' provider...
Bringing machine 'pc2' up with 'virtualbox' provider...
Bringing machine 'pc3' up with 'virtualbox' provider...
Bringing machine 'r1' up with 'virtualbox' provider...
Bringing machine 'r2' up with 'virtualbox' provider...
Bringing machine 'r3' up with 'virtualbox' provider...
==> pc1: Box 'debian/stretch64' could not be found. Attempting to find and install...
    pc1: Box Provider: virtualbox
    pc1: Box Version: >= 0
==> pc1: Loading metadata for box 'debian/stretch64'
    pc1: URL: https://vagrantcloud.com/debian/stretch64
==> pc1: Adding box 'debian/stretch64' (v9.9.0) for provider: virtualbox
    pc1: Downloading: https://vagrantcloud.com/debian/boxes/stretch64/versions/9.9.0/providers/virtualbox.box
    pc1: Download redirected to host: vagrantcloud-files-production.s3.amazonaws.com
==> pc1: Successfully added box 'debian/stretch64' (v9.9.0) for 'virtualbox'!
==> pc1: Importing base box 'debian/stretch64'...
==> pc1: Matching MAC address for NAT networking...
==> pc1: Checking if box 'debian/stretch64' version '9.9.0' is up to date...
==> pc1: Setting the name of the VM: pc1
```

Esperaremos unos minutos y el escenario estará en nuestro virtual box.

 **pc1**
➡ Corriendo

 **pc2**
➡ Corriendo

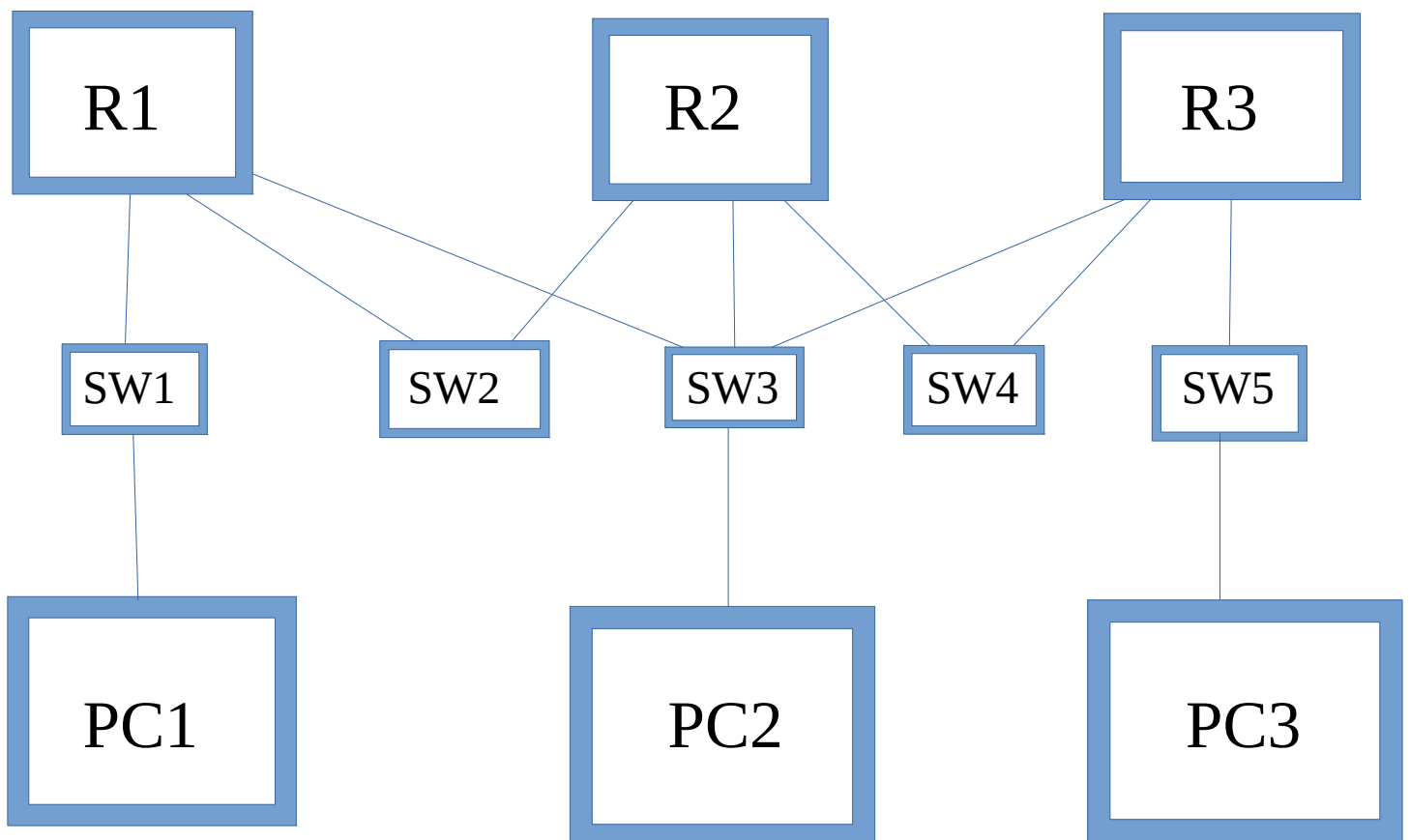
 **pc3**
➡ Corriendo

 **r1**
➡ Corriendo

 **r2**
➡ Corriendo

 **r3**
➡ Corriendo

Ahora vamos a describir como es nuestro escenario en realidad.



2 Definición del Escenario

Podremos entrar en los escenarios con el comando:

```
vagrant ssh {maquina}
```



```
vagrant@r1:~$
```

Debemos instalar el paquete quagga para empezar nuestro escenario OSPF.

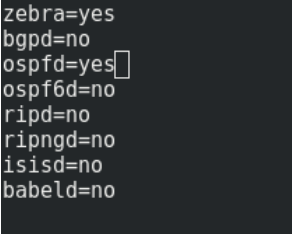
```
*apt-get install quagga
```

```
*Esto se instalará en los tres routers.
```

Paramos el servicio para configurar.

```
*systemctl stop quagga
```

Ahora accederemos al fichero `/etc/quagga/daemons` para configurar nuestro OSPF y lo configuramos.



```
zebra=yes  
bgpd=no  
ospfd=yes  
ospf6d=no  
ripd=no  
ripngd=no  
isisd=no  
babeld=no
```

Ahora debemos copiar nuestra configuración de ejemplo que nos da quagga, en nuestra configuración, para poder editarla.

```
cp /usr/share/doc/quagga/examples/zebra.conf.sample /etc/quagga/zebra.conf
```

Añadimos las siguientes interfaces.

R1->

```
interface eth1
ip address 10.0.100.1/24
interface eth2
ip address 10.0.110.1/24
interface eth3
ip address 10.0.120.1/24

!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!

!log file /var/log/quagga/zebra.log
line vty
```

R2 →

```
interface eth1
ip address 10.0.110.2/24
interface eth2
ip address 10.0.120.2/24
interface eth3
ip address 10.0.130.2/24

!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!

!log file /var/log/quagga/zebra.log
line vty
```

R3 →

```
interface eth1
ip address 10.0.120.3/24
interface eth2
ip address 10.0.130.3/24
interface eth3
ip address 10.0.140.3/24

!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!

!log file /var/log/quagga/zebra.log
line vty
```

3 Configuración de OSPF

Ahora pasamos a la configuración ya propia del proceso OSPF. Para ello tenemos que copiarnos el fichero de ejemplo que nos ofrece el paquete para no tener que escribir todas las opciones:

```
cp /usr/share/doc/quagga/examples/ospfd.conf.sample /etc/quagga/ospfd.conf
```

Ahora editaremos el fichero /etc/quagga/ospfd.conf para poner los routers y dar rangos de red.

R1 →

```
!interfaces que configuramos en zebra.conf
interface eth1
interface eth2
interface eth3
router ospf
  ospf router-id 10.0.100.1
  network 10.0.100.0/24 area 0
  network 10.0.110.0/24 area 0
  network 10.0.120.0/24 area 0
!
log file /var/log/quagga/ospfd.log
line vty
```

R2 →

```
interface eth1
interface eth2
interface eth3
router ospf
  ospf router-id 10.0.110.2
  network 10.0.110.0/24 area 0
  network 10.0.120.0/24 area 0
  network 10.0.130.0/24 area 0
!
log /var/log/quagga/ospfd.log
line vty
```

R3 →

```
interface eth1
interface eth2
interface eth3
router ospf
  ospf router-id 10.0.120.3
  network 10.0.120.0/24 area 0
  network 10.0.130.0/24 area 0
  network 10.0.140.0/24 area 0
!
log /var/log/quagga/ospfd.conf
line vty
```

Y ahora lanzamos el servicio

```
*systemctl start quagga
```

Si queremos verificar que van bien los routers:

```
*telnet localhost zebra
```

```
*telnet localhost ospfd
```

4 Verificación del Route

En los clientes levantamos las interfaces de red:

```
ip l set eth1 up
```

Y le añadimos una ip:

```
ip a add {ip} dev eth1
```

Pc1 →

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:7d:76:85 brd ff:ff:ff:ff:ff:ff
    inet 10.0.100.10/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7d:7685/64 scope link
        valid_lft forever preferred_lft forever
vagrant@pc1:~$
```

Pc2 →

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:fe:65:40 brd ff:ff:ff:ff:ff:ff
    inet 10.0.120.10/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fefe:6540/64 scope link
        valid_lft forever preferred_lft forever
vagrant@pc2:~$
```


Pc3 →

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f7:2a:c5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.140.10/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7:2ac5/64 scope link
        valid_lft forever preferred_lft forever
vagrant@pc3:~$
```

5 Conclusión

Hemos aprendido a realizar un escenario de enrutamiento dinámico.