

Recomendaciones para realizar el trabajo final de RDSV

Curso 2020/21 - Versión del 15 de diciembre de 2020

Creación de una VNF basada en VyOS

Para modificar la VNF *vcpe* para que utilice una imagen basada en VyOS:

- Crear un directorio *img/vnf-vyos* y dentro de él crear un *Dockerfile* con el siguiente contenido:

```
FROM vyos/rolling:1.3
RUN mkdir /config
CMD /sbin/init
```

- Crear la imagen con el comando:

```
sudo docker build -t vnf-vyos .
```

- Comprobar que la imagen se ha creado y está disponible con:

```
docker images
```

- Modificar el descriptor de la VNF *vcpe* para que utilice la imagen *vnf-vyos*. Para ello, utilice el editor de OSM pulsando en el icono correspondiente en la sección “VNF Packages” (Figura 1). Adicionalmente, modifique otros campos del descriptor que considere adecuados (ej: nombres e ids). No olvide pulsar el botón de *Update* una vez finalizadas las modificaciones en el editor.
- Una vez modificado el descriptor, es necesario generar un nuevo paquete (*vnf-vcpe.tar.gz*) con el descriptor modificado. Para ello se debe utilizar la opción de “Crear paquete” (Figura 1).



Figura 1. Detalle opción "VNF Packages" de OSM

Acceso a la consola de una VNF VyOS

El acceso a la consola de un docker VyOS se puede hacer de dos formas:

- Acceso a la shell de VyOS (interfaz de configuración de VyOS):

```
docker exec -ti vyos bash -c 'su - vyos'
```

- Acceso a una shell de Linux (interfaz Linux estándar):

```
docker exec -ti vyos /bin/bash
```

Configuración de comandos en VNF VyOS

Inicialmente se recomienda realizar la configuración de la VNF *vcpe* manualmente a través de la consola, con el objeto de probar los comandos de VyOS hasta dar con la configuración adecuada. Una vez conocidos los comandos, se puede automatizar su ejecución mediante un shellscript similar al siguiente, que configura el hostname de un router VyOS:

```
#!/bin/bash

VNF2="..." # Nombre del docker VyOS. Obtener con "docker ps"
HNAME='vyos'

docker exec -ti $VNF2 /bin/bash -c "
source /opt/vyatta/etc/functions/script-template
configure
set system host-name $HNAME
commit
save
exit
"
```

Configuración del túnel VXLAN entre vclass y vcpe

Es necesario configurar el túnel VXLAN entre las dos VNF desde el router VyOS. Para ello, consulte el manual de VyOS [1].

Se debe usar el modo Unicast VXLAN y especificar los siguientes parámetros:

- Dirección IP del interfaz vxlan (el interfaz que hace de router de la red de residencial)
- Dirección IP del otro extremo del túnel (dirección de vclass en la red interna).
- VNI
- Puerto UDP a utilizar (averiguar el puerto por defecto usado en Linux)

También será necesario ajustar la MTU del interfaz vxlan a los valores que tienen configurados los hosts (h11, h12, etc.).

La dirección IP del otro extremo del túnel puede obtenerse mediante el comando siguiente:

```
sudo docker exec -it $VNF1 hostname -I | tr " " "\n" | grep 192.168.100
```

siendo VNF1 el nombre del docker de la VNF vclass.

Modificación del script vcps_start.sh

Con el cambio a VyOS de la VNF vcpe será necesario modificar este script para eliminar de él todos los comandos que antes se dedicaban a configurar la VNF vcpe basada en la imagen de Ubuntu. Se recomienda estudiar este script para comprender los pasos que realiza y comentar en él todas las configuraciones relativas a la VNF2. Estas configuraciones deberán realizarse en el script de configuración de VyOS.

Para crear el interfaz de red que conecta el router VyOS con la red ExtNet se debe utilizar el siguiente comando:

```
sudo ovs-docker add-port ExtNet eth2 $VNF2
```

Es importante que el nombre del interfaz creado sea ethX, ya que VyOS solo reconoce interfaces con ese nombre.

Túneles VXLAN de VNF vclass

La gestión de la calidad de servicio que hay que implementar en la VNF vclass no funciona adecuadamente cuando se aplica sobre interfaces de túneles VXLAN creados desde un Open vSwitch, tal como se realiza en la P4. Por ello, es necesario crear los túneles desde Linux con el comando 'ip link'.

Para realizar el cambio, se deben sustituir las siguientes dos líneas siguientes del fichero `vcpe_start.sh`:

```
sudo docker exec -it $VNF1 ovs-vsctl add-port br0 vxlan1 -- set interface vxlan1
type=vxlan options:remote_ip=$HOMETUNIP
#sudo docker exec -it $VNF1 ovs-vsctl add-port br0 vxlan2 -- set interface vxlan2
type=vxlan options:remote_ip=$IP21
```

Por estas otras dos líneas:

```
sudo docker exec -it $VNF1 ip link add vxlan1 type vxlan id 0 remote $HOMETUNIP
dstport 4789 dev veth0
sudo docker exec -it $VNF1 ip link add vxlan2 type vxlan id 1 remote $IP21 dstport
8472 dev eth1-0
sudo docker exec -it $VNF1 ovs-vsctl add-port br0 vxlan1
sudo docker exec -it $VNF1 ovs-vsctl add-port br0 vxlan2
sudo docker exec -it $VNF1 ifconfig vxlan1 up
sudo docker exec -it $VNF1 ifconfig vxlan2 up
```

En caso de que realice la parte opcional de controlar también la calidad de servicio en `brg1`, deberá sustituir también el comando que crea el túnel VXLAN desde `brg1`.

Arranque de la máquina virtual en el laboratorio con directorio compartido

Se ha creado un script para facilitar la descarga e instalación de la máquina virtual `VNXSDNNFVLAB2020` en los ordenadores del laboratorio:

```
/lab/rdsv/rdsv-get-and-install-vnxsdnfvlab
```

Ese comando descarga el fichero `VNXSDNNFVLAB2020-v5.ova` al directorio `/mnt/tmp` y lo importa a VirtualBox. Además, configura en la máquina virtual un directorio compartido en el host de forma que el directorio `~/shared` del PC del laboratorio (esto es, el directorio `/home/<usuario>/shared`) aparezca montado en `/home/upm/shared` dentro de la máquina virtual.

La ejecución completa del script dura unos 6 minutos (probado con el laboratorio vacío; puede tardar algo más dependiendo de la ocupación del laboratorio).

Instalación de Ryu

Para instalar el controlador Ryu en la VNF `vclass` se recomienda hacerlo desde el gestor de paquetes de Ubuntu (paquete `ryu-bin`), ya que la versión que hay en github plantea problemas con la gestión de la QoS.

Tenga en cuenta, además, que hay que crear el fichero `qos_simple_switch_13.py` con la modificación que se propone en el tutorial (http://osrg.github.io/ryu-book/en/html/rest_qos.html).

Se recomienda realizar ambas tareas en el Dockerfile que crea la imagen de `vclass`.

Además, es necesario instalar en la imagen el paquete `'iproute2'`, necesario para utilizar el comando `'ip'`.

Finalmente, se recomienda actualizar la versión de Ubuntu que usa la imagen a 18.04. Para ello, cambiar la línea `"FROM ubuntu:xenial"` a `"FROM ubuntu:bionic"`.

Instalación de nuevos paquetes en la imagen usada por los contenedores de los escenarios VNX

Para instalar nuevos paquetes en la imagen `vnx_rootfs_lxc_ubuntu64-18.04-v025-vnxlab` utilizada por los contenedores arrancados por VNX se debe:

- Parar los escenarios VNX
- Arrancar la imagen en modo directo con:

```
vnx --modify-rootfs /usr/share/vnx/filesystems/vnx_rootfs_lxc_ubuntu64-18.04-v025-vnxlab/
```

- Hacer login con root/xxxx e instalar los paquetes deseados.
- Parar el contenedor con:

```
halt -p
```

- Arrancar de nuevo los escenarios VNX y comprobar que el software instalado ya está disponible.

Este método se puede utilizar para instalar, por ejemplo, iperf3, que no está disponible en la imagen.

Otras recomendaciones

- En el examen oral se pedirá arrancar el escenario desde cero, por lo que es importante que todos los pasos para cumplir los requisitos mínimos estén automatizados mediante uno o varios scripts. Si hay partes opcionales que se configuran de forma manual, se deberán tener documentados todos los comandos para ejecutarlos rápidamente mediante copia-pegar.
- Se recomienda dejar la parte de configuración de la calidad de servicio en la VNF vclass para el final, una vez que el resto del escenario este funcionando (túneles VXLAN, conectividad h1X-vcpe, DHCP, etc.).
- Comando para ver la configuración completa de un túnel VXLAN en Linux (ejemplo para interfaz vxlan1):

```
ip -d link show vxlan1
```

- Vim-emu arranca los docker con un interfaz eth0 conectado a una red interna de docker que da acceso a Internet. Téngalo en cuenta por si pudiera interferir con las rutas del escenario.
- Para realizar un ping desde VyOS es necesario usar 'sudo'.

Referencias

1. VyOS User Guide. <https://docs.vyos.io/en/latest/index.html>