**Luis Alberto López Álvarez y Álvaro de Rojas Maraver**

**Práctica Final - RDSV**

Para la realización de la práctica final de la asignatura, partimos de lo elaborado en la práctica 4, modificando las redes virtualizadas mediante OSM, que ahora contarán con los siguientes requisitos:

* soporte de QoS implementado mediante SDN con Ryu
* añadir servicios adicionales, como el registro de las MACs de la red residencial

A continuación, se muestra el escenario que se implementará en este trabajo final:

Diagrama

Descripción generada automáticamente

Los **requisitos mínimos** desarrollados establecidos en el enunciado, como modificaciones a la práctica 4 son:

* Sustituir el switch de KNF:access por un conmutador controlado por OpenFlow
* Conectividad IPv4 desde la red residencial hacia Internet. Uso de doble NAT: en KNF:cpe y en isp1 (igual que en la p4)
* Activar la captura de tráfico ARP mediante “arpwatch”
* Gestión de la calidad de servicio en la red de acceso mediante la API REST de Ryu controlando KNF:access
  + Para limitar el ancho de banda de bajada hacia la red residencial
* Despliegue para dos redes residenciales
* Todo automatizado mediante OSM y scripts
  + Incluyendo el on-boarding de NS/VNFs y la instanciación de NS mediante línea

de comandos

Proponiéndose como **requisitos opcionales** los siguientes: **QUITAR ESTO**

* Utilizar un repositorio privado de imágenes Docker:
  + el de MicroK8s
* Sustituir el switch de brgX por un conmutador controlado por OpenFlow desde el Ryu
  + Incluyendo la gestión de la calidad de servicio desde el Ryu instalado en KNF:access, controlando el brgX, para limitar el ancho de banda de subida desde la red residencial
* Instalar la funcionalidad arpwatch en un tercer contenedor
  + Requiere modificar los descriptores de OSM y crear un nuevo Helm chart
* Sustituir la red de acceso por una red IPv6
* Soporte IPv6 en la red residencial
  + Acceso mediante IPv6 al servidor S1
* DHCP para IPv6

Para realizar el **despliegue** de nuestro proyecto, se pueden seguir los pasos detallados en el **fichero README.txt adjunto** en el proyecto, mientras que en este documento se mostrarán los cambios realizados para lograr las funcionalidades requeridas.

Para ello, las **tareas básicas** que hemos empezado a realizar que permiten cumplir con estos requisitos se enumeran a continuación, una vez se han entendido los detalles internos del escenario de la práctica 4:

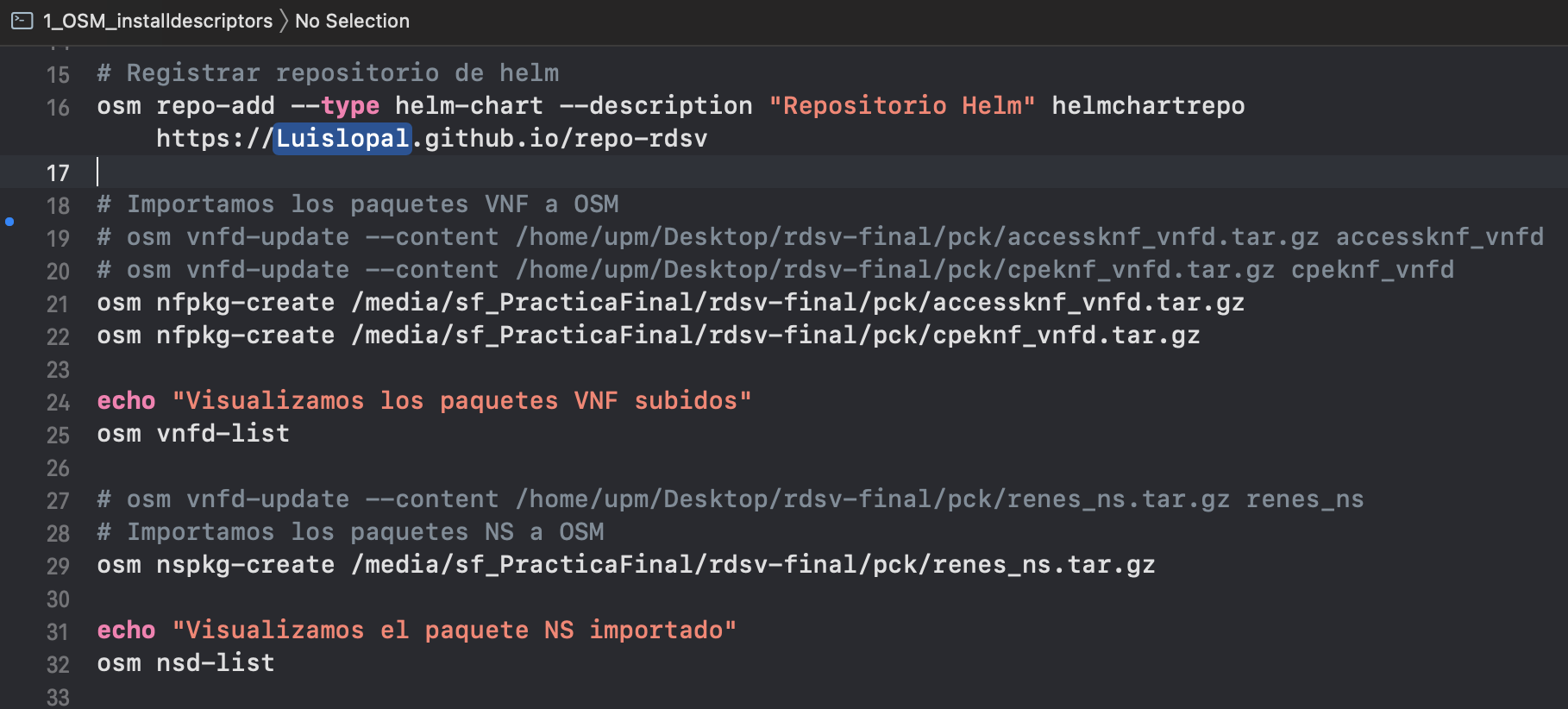
* Sustituir los repositorios por nuestros propios repositorios:
  + El contenedor Docker de las KNFs en DockerHub
    - cuenta educaredes 🡪 cuenta luichu
  + El repositorio Helm en GitHub Pages
    - <https://educaredes.github.io/nfv-lab> 🡪 https://Luislopal.github.io/repo-rdsv
* Automatización del despliegue de la práctica anterior mediante OSM y scripts para montar el escenario de las redes.
* Despliegue de lo anterior para dos redes residenciales.
* Comprobación de que existe conectividad IPv4 desde la red residencial hacia Internet mediante un doble NAT: en KNF:cpe y en isp1.
* Añadir la funcionalidad “arpwatch” para capturar tráfico ARP.
  + Activación de la herramienta arpwatch en KNF:cpe
* Gestión de QoS en la frontera de la red de acceso, mediante la API REST de Ryu en KNF:access (y en brgX) para limitar el ancho de banda de bajada hacia la red residencial.
  + Para la red residencial: 12 Mbps de bajada (y 6 Mbps de subida)
  + Para hX1: 8 Mbps mínimo de bajada (y 4 Mbps mínimo de subida)
  + Para hX2: 4 Mbps máximo de bajada (y 2 Mbps máximo de subida)
  + Independiente de la dirección IP asignada por DHCP a hX1 y hX2

Para poder empezar la práctica es necesario montar una carpeta compartida en virtualbox, con la ruta local en el ordenador anfitrión y punto de montaje /home/upm/shared en las máquinas virtuales (K8S y OSM), como se detalla en el enunciado de la práctica (en nuestro caso, se ha realizado en un único MAC, que se utilizará como máquina anfitriona). Se va a trabajar sobre dos máquinas virtuales: RDSV-K8S (192.168.56.11), que emula las distintas redes y máquinas del escenario y RDSV-OSM (192.168.56.12), que instala el entorno OSM que coordina las funciones de la red virtual.

Dentro de ambas máquinas es necesario realizar unos scripts que realizan la configuración inicial y otorgan los permisos necesarios al escenario, para poder realizar el despliegue posterior correctamente. Además, dentro de la máquina RDSV-K8S, se realizan los despliegues de los escenarios “home” y “server” utilizando el comando: ‘sudo vnx -f vnx/nfv3\_home\_lxc\_ubuntu64.xml -t’ y ‘sudo vnx -f vnx/nfv3\_server\_lxc\_ubuntu64.xml -t’ dentro del directorio donde se ubica la descarga del proyecto.

Se ha realizado de igual forma un segundo script, que permite apagar el escenario sin tener que introducir ambos comandos de forma recurrente.

Ya desde la máquina virtual RDSV-OSM, es necesario registrar el repositorio helm que se va a utilizar posteriormente, para poder subir tanto los paquetes VNF como los NS a OSM, lo que se conoce como on-boarding, unido a la instanciación de cada una de las dos instancias que se van a desplegar siguiendo el proceso realizado en la práctica anterior. Todo esto se realiza mediante dos scripts, que se adjuntan a continuación y habrá que ejecutar desde la máquina OSM:



Texto

Descripción generada automáticamente

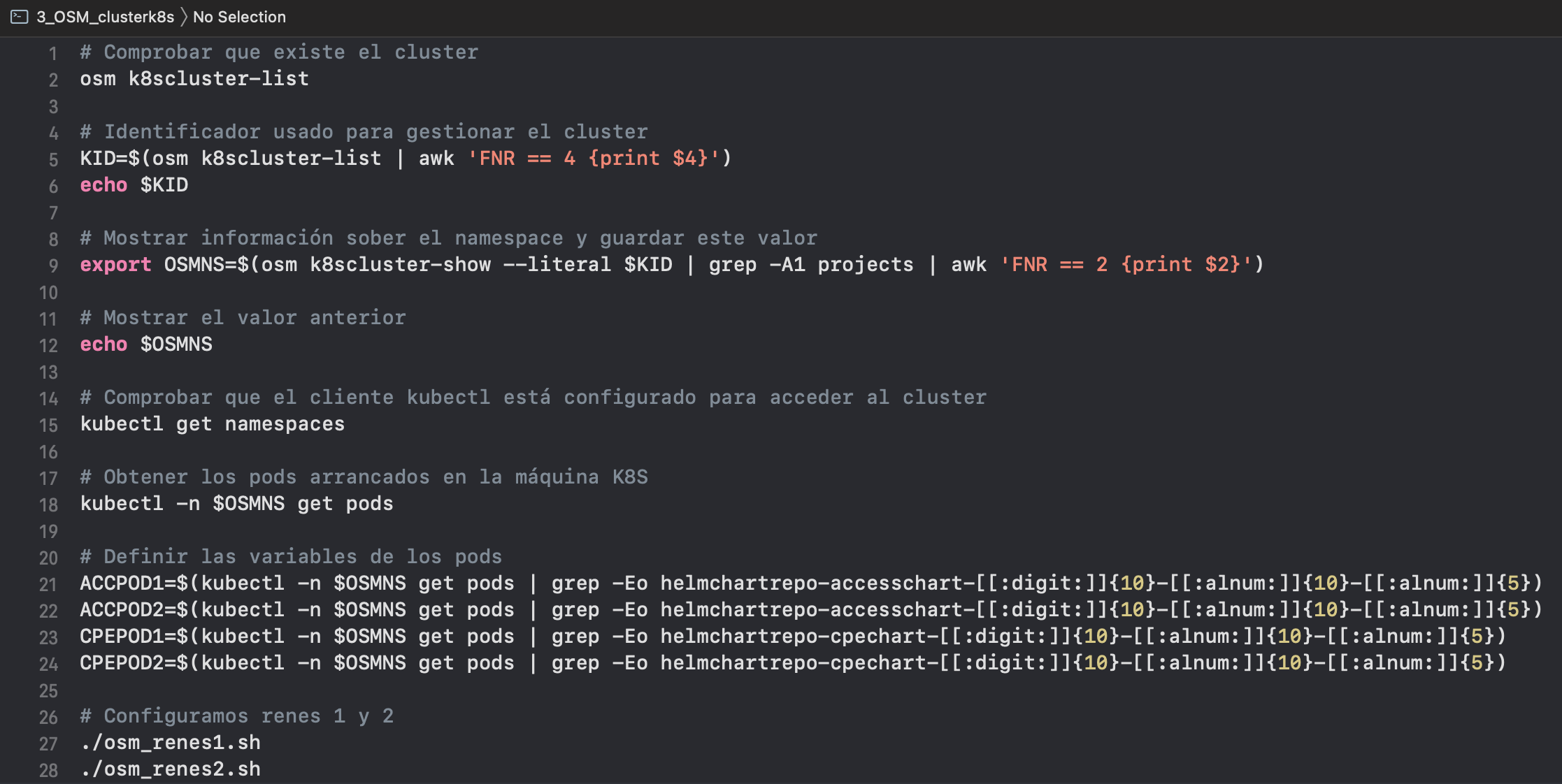
Posteriormente, es necesario ubicar el clúster de k8s existente en el escenario y guardar su identificador y valor de namespace, como se muestra en la siguiente captura, permitiendo gestionar el clúster y desplegar los pods de los servicios de red, que serán necesarios para realizar la configuración de red necesaria a través de los scripts de renes.

**Texto

Descripción generada automáticamente**

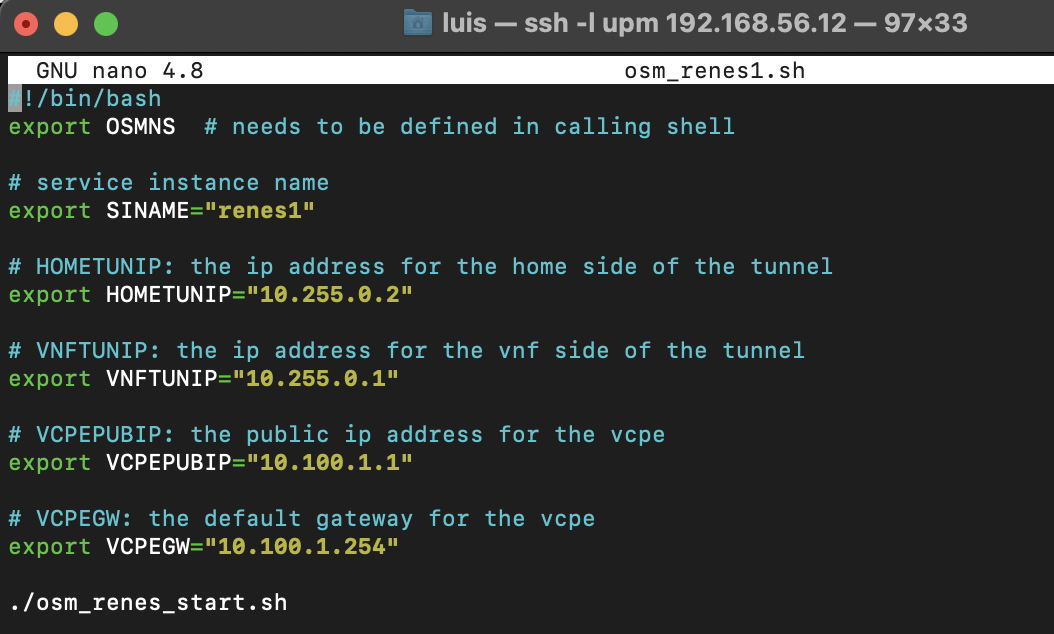
**Captura 4. Exportación variable global namespace**

Por ello, una vez definidas cada una de las variables de los pods, tal como se muestra en la captura siguiente, es necesario lanzar los scripts de rennes que configuran las instancias de red, y se adjuntan de igual forma:

****

Se puede apreciar en la captura, que la dirección IP que utiliza el NAT del CPE para dar salida al tráfico es la ’10.100.1.1’ correspondiente con la ‘VCPEPUBIP, que representa el enlace con el switch ExtNet1.

Se utilizan dos direcciones, una para cada interfaz dentro de cada una de las redes que hay en cada switch de acceso. En el caso del AccessNet1 la ‘10.255.0.0/24’, siendo la ‘10.255.0.1’ la interfaz que conecta con la instancia de ‘rennes’ y la ‘10.255.0.2’ la que conecta con los equipos h11 y h12. Con ExtNet1 ocurre igual, pero ahora con la red ‘10.100.1.0/24’, utilizando la IP ’10.100.1.1’ para la conexión con ‘rennes’ y la ’10.100.1.254’ para la conexión con Internet.



**Captura 11. Fichero de configuración osm\_renes1.sh**

Texto

Descripción generada automáticamente

**Captura 16. Script de configuración ‘osm\_renes2.sh’**

En este caso, **h21 y h22 han obtenido las direcciones IP ‘10.250.0.23’ y la ‘10.250.0.22’** respectivamente, valores incluidos en el rango de direcciones posibles a adjudicar por DHCP (y justo los dos siguientes a los dos primeros adjudicados previamente a h11 y h12).

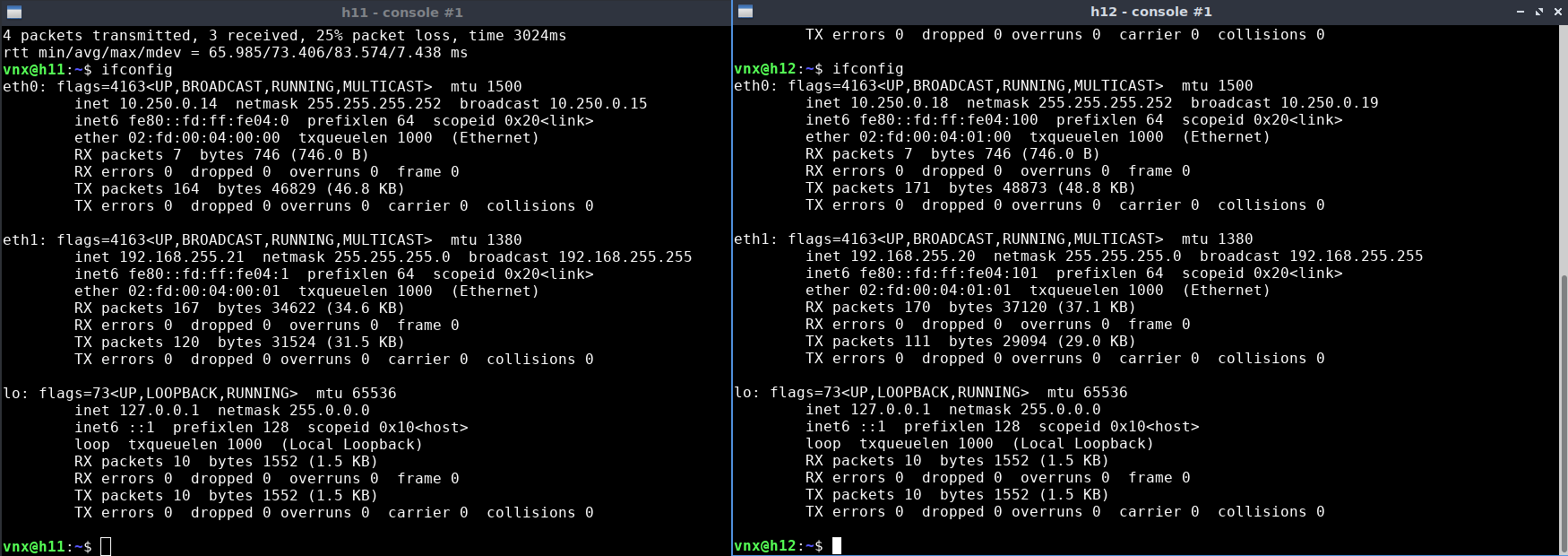
**Todo esto que sigue es de la p4 pero igual nos sirve para algo….**

1. **Configuración del servicio:** Analice a continuación el detalle del escenario en la Fig. 4 e indique qué comando(s) puede utilizar desde RDSV-OSM para comprobar si hay conectividad entre el servicio desplegado y el dispositivo brg1 de la red residencial. Verifique que haya conectividad.

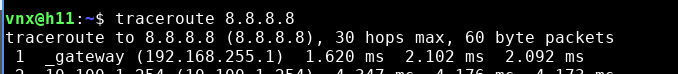
Según lo descrito, existe conectividad entre el servicio desplegado y brg1. Para su comprobación, se ha obtenido la dirección IP de h11, ya configurada tras realizar los pasos anteriores, y se ha lanzado un traceroute a dicha IP (10.250.0.14) desde la máquina RDSV-OSM, alcanzando su destino tras realizar dos saltos, como cabía esperar. Además, se podría realizar desde la propia máquina RDSV-OSM distintos escaneos de red que detecten las distintas IPs activas en AccessNet1, verificando si hay conectividad con cierta dirección de dicha red (mediante una herramienta como puede ser ‘nmap’, por ejemplo).

1. **Servicio desde la red de acceso:** Indique qué direcciones IP obtienen h11 y h12 en la red residencial “privada”, así como la dirección IP del router. Relacione el resultado con los ficheros de configuración del contenedor docker educaredes/vnf-img incluidos en el directorio $HOME/shared/nfv-lab/img/vnf-img.

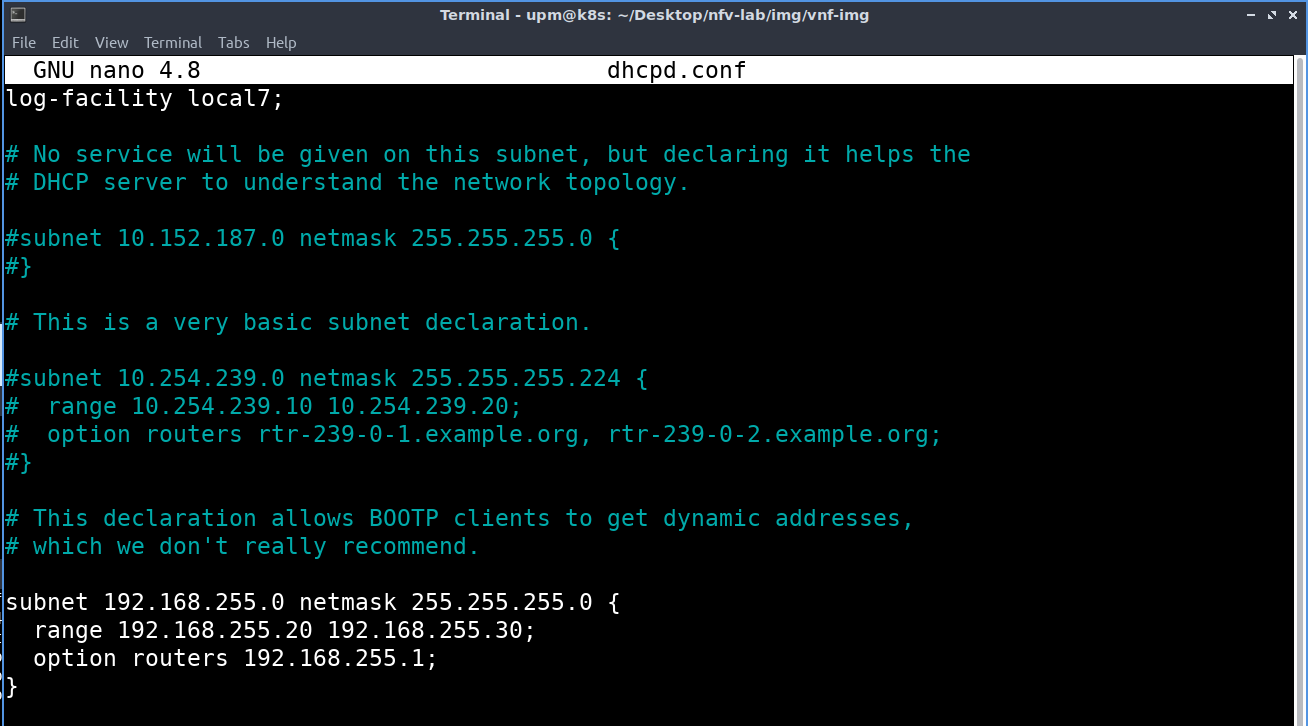
En la siguiente captura se ha realizado un ‘ifconfig’ en cada uno de los sistemas de la red residencial “privada”. Se puede observar como en el caso de h11, DHCP ha asignado la dirección ’192.168.255.21’ y la ‘192.168.255.20’ a h12. Para poder averiguar la dirección establecida en el router, hemos realizado un traceroute desde la máquina h11 hacia internet, correspondiendo el primero de los saltos a su dirección IP que es la ‘192.168.255.1’, como cabía esperar al ser la primera del rango (configuración habitual).



**Captura 12. Comando ‘ifconfig’ sistemas finales h11 y h12**



**Captura 13. Traceroute desde h11 hacia Internet**



**Captura 14. Fichero de configuración DHCP**

Por último, se han examinado los ficheros de configuración del servidor DHCP, comprobando que es correcto lo que acaba de ocurrir. La configuración en este caso asigna un rango de IPs disponibles entre la ‘192.168.255.20’ y la ‘192.168.255.30’ (se han asignado las dos primeras, la .20 y la .21), quedando para el router la IP ‘192.168.255.1’, y verificando la configuración de la red según lo deseado.

1. **Análisis de tráfico en AccessNet1:** Desde RDSV-K8S, arranque wireshark y póngalo a capturar el tráfico en AccessNet1. Desde h11 realice un ping de 5 paquetes a la dirección IP de su router, comprobando que funciona correctamente. Detenga wireshark, y guarde la captura con nombre “access1.pcapng”. Analice el tráfico capturado, justificando las direcciones IP que aparecen en los paquetes capturados.

**\*Captura de wireshark ‘access1.pcap’ adjunta en el zip de la entrega. (Leer Nota p8)**

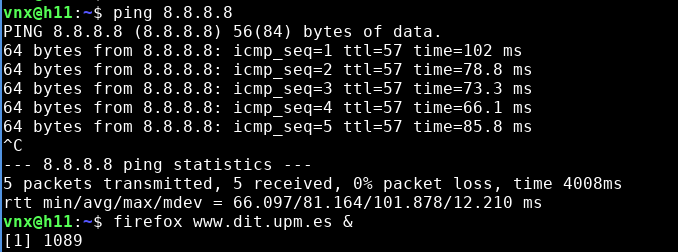
En cuanto a las direcciones IP, se puede apreciar en la captura de wireshark como son las correspondientes a lo analizado anteriormente, es decir, la comunicación se da entre las direcciones 192.168.56.1 y la 192.168.56.21, correspondientes con el router y h11 respectivamente.

1. **Análisis de tráfico en ExtNet1:** Arranque wireshark y póngalo a capturar el tráfico en ExtNet. Desde h11 realice un ping de 5 paquetes a la dirección IP de s1 (10.100.3.2), comprobando que funciona correctamente. Detenga wireshark, y guarde la captura con nombre “ext1.pcapng”. Analice el tráfico capturado, justificando las direcciones IP que aparecen en los paquetes capturados. Desde la consola de h11, compruebe que tiene acceso a Internet. Además de usar ping, puede arrancar un navegador.

**\*Captura de wireshark ‘ext1.pcap’ adjunta en el zip de la entrega. (Nos dio problemas el formato .pcapng en anteriores entregas, entregamos este para evitar fallos de visualización)**

En este caso, se puede comprobar a través de la captura de wireshark como el tráfico corresponde a las IPs ‘10.100.1.1’ y ‘10.100.3.2’, siendo la segunda de ellas la de s1, pero ahora la primera dirección ha sufrido la traducción NAT por parte del cpe, por lo que h11 pasa a tener una dirección “privada” a la hora de comunicarse tanto con s1 como con Internet.

Se puede ver en la siguiente captura como se puede acceder a internet y arrancar un navegador desde la máquina h11.



**Captura 15. Conectividad a Internet h11**