



Instituto Politécnico Nacional

Escuela Superior de Cómputo

UA.Administración de Servicios en Red

‘Proyecto Final’

Alumnos:

Castro Velázquez Rogelio

Pérez Gómez Santiago

Márquez León Jorge Luis

Hernández Magallón Erick Raziel

Profesora:

Leticia Henestrosa Carrasco

Grupo:

4CV13

Índice

| | |
|---|-----------|
| Índice | 2 |
| Introducción | 3 |
| Conceptos Básicos | 3 |
| ACL Extendida | 3 |
| Azure | 4 |
| DHCP | 4 |
| DNS | 5 |
| Docker | 5 |
| HTTP | 5 |
| INTER-VLAN | 6 |
| NAT | 6 |
| OSPF | 6 |
| SSH | 7 |
| VLAN | 7 |
| VLSM | 7 |
| Desarrollo | 8 |
| Azure | 11 |
| OSPF | 11 |
| VLAN | 12 |
| INTER-VLAN | 13 |
| ACL Extendidas | 13 |
| NAT | 14 |
| Docker | 15 |
| Dockerfile e Imagen de Dockerfile | 16 |
| Persistencia de Docker y Configuración IP | 16 |
| HTTP | 19 |
| DHCP | 20 |
| SSH | 24 |
| DNS | 25 |
| Conclusiones | 28 |
| Referencias | 29 |

Introducción

La administración de servicios en red comprende la operación, supervisión y mantenimiento de una red de forma remota.

El objetivo consiste en garantizar que las actividades de una organización se lleven a cabo con éxito como el funcionamiento adecuado de los servicios de correo electrónico, firewalls de seguridad, redes privadas virtuales (VPN), asignación automática de direcciones IP (DHCP), bases de datos, entre otros.

En el presente proyecto se implementa una topología lógica que implementa diversos servicios fundamentales en las redes actuales, para ello, se utiliza una máquina virtual con GNS3 montada en Microsoft Azure.

Conceptos Básicos

A continuación se hace una breve descripción de los protocolos y servicios que utilizamos en la topología lógica.

ACL Extendida

Cuando se requiere un control de filtrado de tráfico más preciso, se pueden crear ACL extendidas de IPv4 .

Las ACL extendidas se utilizan con más frecuencia que las ACL estándar, porque proporcionan un mayor grado de control. Pueden filtrar por dirección de origen, dirección de destino, protocolo (es decir, IP, TCP, UDP, ICMP) y número de puerto. Esto proporciona una gama de criterios más amplia sobre la cual basar la ACL. Por ejemplo, una ACL extendida puede permitir el tráfico de correo electrónico de una red a un destino específico y, simultáneamente, denegar la transferencia de archivos y la navegación web.

Al igual que las ACL estándar, las ACL extendidas se pueden crear como:

- **ACL Extendida Numerada** – Creado con el comando de configuración global `access-list access-list-number`
- **ACL Extendida Nombrada** – Creado con el comando `ip access-list extended access-list-name`.

Azure

Azure es un conjunto de servicios en la nube de la empresa Microsoft. Con Azure es posible almacenar información y crear, administrar e implementar aplicaciones en cloud. Para utilizar Azure es necesario el pago de una cuota que recoge los servicios contratados.

Desde el portal de Microsoft Azure se puede acceder a diferentes servicios de infraestructura y plataforma para contratar aquellos que sean necesarios para la empresa o proyecto. En apenas unos clics es posible disponer de Microsoft Azure funcionando y listo para trasladar el trabajo a la nube.

Microsoft ofrece una alta disponibilidad de los servicios (99.99 %) y cuenta con las mejores certificaciones en materia de seguridad y protección de datos existentes.

DHCP

El DHCP es una extensión del protocolo Bootstrap (BOOTP) desarrollado en 1985 para conectar dispositivos como terminales y estaciones de trabajo sin disco duro con un Bootserver, del cual reciben su sistema operativo.

El DHCP se desarrolló como solución para redes de gran envergadura y ordenadores portátiles y por ello complementa a BOOTP, entre otras cosas, por su capacidad para asignar automáticamente direcciones de red reutilizables y por la existencia de posibilidades de configuración adicionales.

DNS

El sistema de nombres de dominio (DNS) es el directorio telefónico de Internet. Las personas acceden a la información en línea a través de nombres de dominio como nytimes.com o espn.com. Los navegadores web interactúan mediante direcciones de Protocolo de Internet (IP). El DNS traduce los nombres de dominio a direcciones IP para que los navegadores puedan cargar los recursos de Internet.

Cada dispositivo conectado a Internet tiene una dirección IP única que otros equipos pueden usar para encontrarlo. Los servidores DNS suprimen la necesidad de que los humanos memoricen direcciones IP tales como 192.168.1.1 (en IPv4) o nuevas direcciones IP alfanuméricas más complejas, tales como 2400:cb00:2048:1::c629:d7a2 (en IPv6).

Docker

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.

HTTP

Hypertext Transfer Protocol (HTTP) (o Protocolo de Transferencia de Hipertexto en español) es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también. Sigue el clásico modelo cliente-servidor, en el que un cliente establece una conexión, realizando una petición a un servidor y espera una respuesta del mismo. Se trata de un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos peticiones. Aunque en la mayoría de casos se basa en una conexión del tipo TCP/IP, puede ser usado sobre cualquier capa de transporte segura o de confianza, es decir, sobre cualquier protocolo que no pierda mensajes silenciosamente, tal como UDP.

INTER-VLAN

Inter-VLAN Routing (Router on a stick) nos brinda la facilidad de utilizar solo una interfaz para enrutar los paquetes de varias VLANs que viajan a través del switch conectado a esa interfaz, es decir, podemos configurar varias IP de diferentes redes a varias interfaces virtuales (subinterfaces) alojadas en una sola interfaz física.

NAT

La traducción de dirección de red o NAT se refiere a un proceso específico que implica la reordenación de una única dirección IP en otra dirección IP pública, mediante la modificación de la información de red y la información de dirección que se encuentra en la cabecera IP de los paquetes de datos.

Las redes locales tienen varias direcciones IP privadas que pertenecen a dispositivos específicos de la red. A través de un sistema NAT, estas direcciones privadas se traducen en una dirección IP pública cuando las peticiones salientes de los dispositivos de red se envían a Internet.

OSPF

Open Shortest Path First (OSPF) es un protocolo de direccionamiento de tipo enlace-estado, desarrollado para las redes IP y basado en el algoritmo de primera vía más corta (SPF).

En una red OSPF, los direccionadores o sistemas de la misma área mantienen una base de datos de enlace-estado idéntica que describe la topología del área.

Cada direccionador o sistema del área genera su propia base de datos de enlace-estado a partir de los anuncios de enlace-estado (LSA) que recibe de los demás direccionadores o sistemas de la misma área y de los LSA que él mismo genera. El LSA es un paquete que contiene información sobre los vecinos y los costes de cada vía. Basándose en la base de datos de enlace-estado, cada direccionador o sistema calcula un árbol de extensión de vía más corta, siendo él mismo la raíz, utilizando el algoritmo SPF.

SSH

SSH™ (o Secure SHell) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como telnet o rsh. Un programa relacionado, el scp, reemplaza otros programas diseñados para copiar archivos entre hosts como rcp. Ya que estas aplicaciones antiguas no encriptan contraseñas entre el cliente y el servidor, evite usarlas mientras le sea posible. El uso de métodos seguros para registrarse remotamente a otros sistemas reduce los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto.

VLAN

Una VLAN, acrónimo de virtual LAN o Red de Área Local Virtual, es una tecnología para crear redes lógicas independientes dentro de una misma red física. Son útiles para reducir el dominio de difusión de la información, y ayudan en la administración de la red, separando segmentos lógicos (las oficinas o departamentos de una organización, por ejemplo) que deberían estar relacionados solo entre ellos.

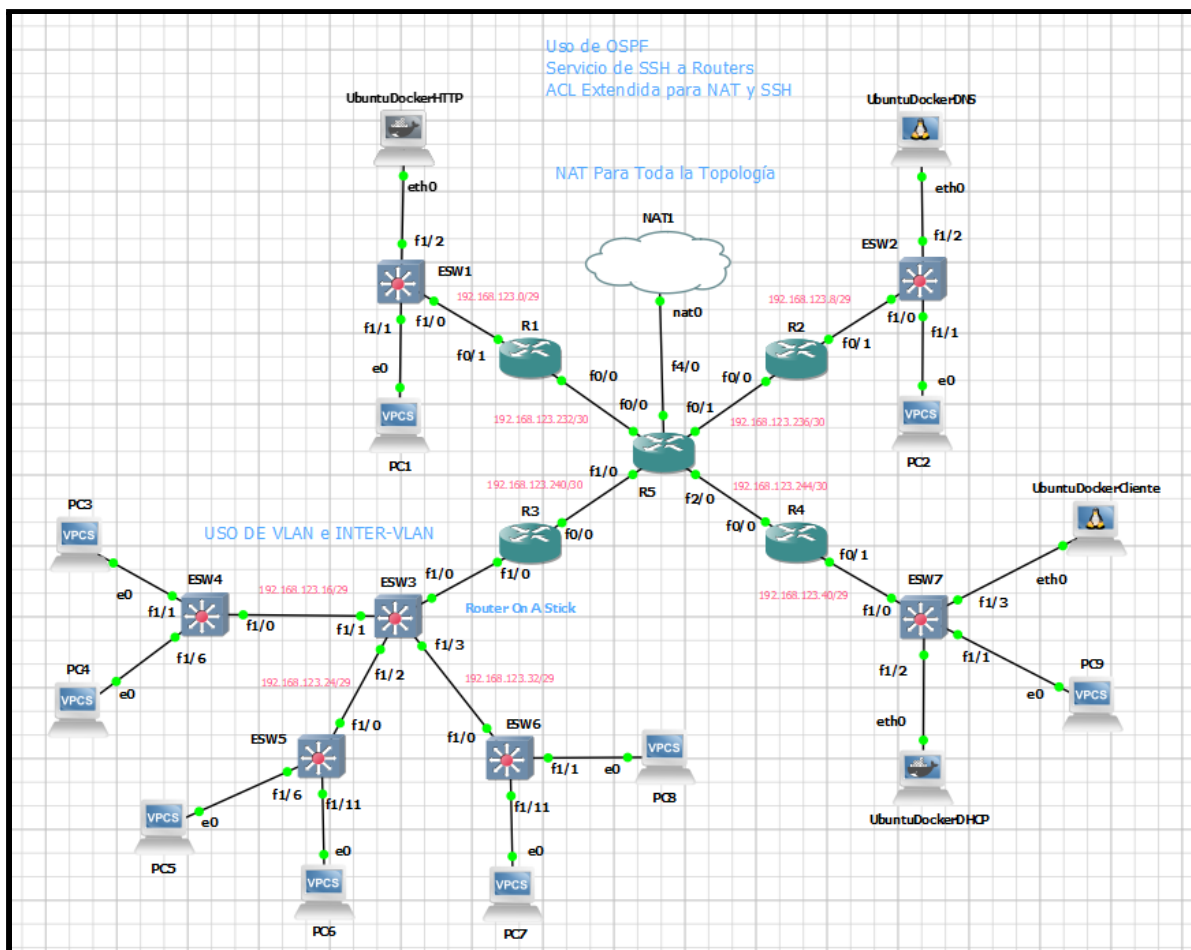
VLSM

VLSM permite dividir un espacio de red en partes desiguales, es decir, la máscara de subred de una dirección IP variará según la cantidad de bits que se tomen prestados para una subred específica, se conoce también como división de subredes en subredes.

VLSM surge como solución para evitar el agotamiento de direcciones IP (1987), también para reducir el tráfico general de la red y mejorar el rendimiento de esta y así conservar el espacio de direcciones.

Desarrollo

Para la realización del proyecto, se llevó a cabo la siguiente topología:



Para esto, se eligió una red clase C, debido al número reducido de dispositivos de la topología, se implementó VLSM considerando 8 host por subred, de los cuales 6 se utilizaron de manera estática y para uso del servidor DHCP. La tabla de direcciones empleada es la siguiente:

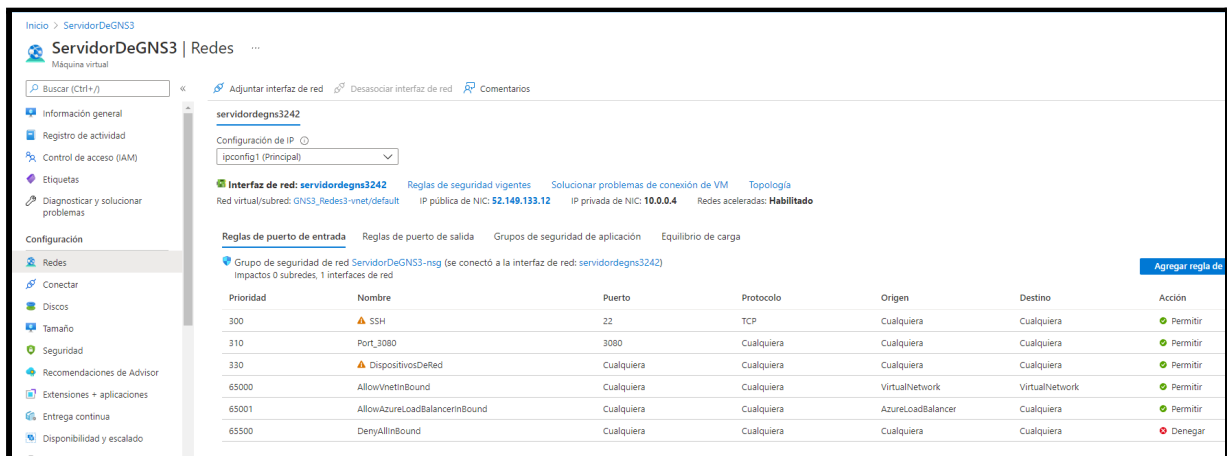
| Dispositivo | Interfaz | Dirección IP | Máscara de Subred |
|-------------|----------|-----------------|-------------------|
| R1 | F0/0 | 192.168.123.233 | 255.255.255.252 |
| | F0/1 | 192.168.123.1 | 255.255.255.248 |
| R2 | F0/0 | 192.168.123.237 | 255.255.255.252 |
| | F0/1 | 192.168.123.9 | 255.255.255.248 |
| R3 | F0/0 | 192.168.123.241 | 255.255.255.252 |
| | F1/0.10 | 192.168.123.17 | 255.255.255.248 |
| | F1/0.20 | 192.168.123.25 | 255.255.255.248 |
| | F1/0.30 | 192.168.123.33 | 255.255.255.248 |
| R4 | F0/0 | 192.168.123.245 | 255.255.255.252 |
| | F0/1 | 192.168.123.41 | 255.255.255.248 |
| R5 | F0/0 | 192.168.123.234 | 255.255.255.252 |
| | F0/1 | 192.168.123.238 | 255.255.255.252 |
| | F1/0 | 192.168.123.242 | 255.255.255.252 |
| | F2/0 | 192.168.123.246 | 255.255.255.252 |
| | F4/0 | DHCP | 255.255.255.0 |
| PC1 | E0 | 192.168.123.2 | 255.255.255.248 |
| PC2 | E0 | 192.168.123.10 | 255.255.255.248 |

| | | | |
|-------------------------|---------|----------------|-----------------|
| PC3 | VLAN 10 | 192.168.123.18 | 255.255.255.248 |
| PC4 | VLAN 20 | 192.168.123.26 | 255.255.255.248 |
| PC5 | VLAN 20 | 192.168.123.27 | 255.255.255.248 |
| PC6 | VLAN 30 | 192.168.123.34 | 255.255.255.248 |
| PC7 | VLAN 30 | 192.168.123.35 | 255.255.255.248 |
| PC8 | VLAN 10 | 192.168.123.19 | 255.255.255.248 |
| PC9 | E0 | 192.168.123.42 | 255.255.255.248 |
| UbuntuDocker SNMP | Eth0 | 192.168.123.3 | 255.255.255.248 |
| UbuntuDocker HTTP | Eth0 | 192.168.123.4 | 255.255.255.248 |
| UbuntuDocker DNS | Eth0 | 192.168.123.11 | 255.255.255.248 |
| UbuntuDocker DHCP | Eth0 | 192.168.123.43 | 255.255.255.248 |
| UbuntuDocker Cliente | Eth0 | 192.168.123.44 | 255.255.255.248 |
| Native VLAN SW3 | | 192.168.123.49 | 255.255.255.248 |
| Native VLAN SW4 | | 192.168.123.50 | 255.255.255.248 |
| Native VLAN SW5 | | 192.168.123.51 | 255.255.255.248 |
| Native VLAN SW6 | | 192.168.123.52 | 255.255.255.248 |

Tabla 1: Tabla con Ip's de interfaces de cada dispositivo

Azure

Primero, es necesario ingresar al portal de Azure y crear un servidor, una vez creado se deben agregar por lo menos dos reglas, una que permita el acceso al puerto 3080 y otra a los puertos 5000 a 6000, esto para poder trabajar en GNS3, en nuestro caso se añadió una regla que permite conexión desde cualquier puerto:



OSPF

Para configurar OSPF se utilizó como id de proceso 1, con un área 0 para todos los routers, dando a conocer las redes a las que se encuentra conectado cada router. A continuación se muestra el router 5 como ejemplo:

```
R5#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.123.246
  It is an autonomous system boundary router
  Redistributing External Routes from,
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    192.168.123.232 0.0.0.3 area 0
    192.168.123.236 0.0.0.3 area 0
    192.168.123.240 0.0.0.3 area 0
    192.168.123.244 0.0.0.3 area 0
  Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance        Last Update
    192.168.123.241      110           00:29:47
    192.168.123.246      110           00:30:27
    192.168.123.245      110           00:29:47
    192.168.123.233      110           00:29:47
    192.168.123.237      110           00:29:47
  Distance: (default is 110)
```

VLAN

En la sección de VLAN se crearon 4 VLAN en total, 3 para segmentar la red y 1 para proteger el tráfico de las VLAN, cada una con los siguientes nombres.

- VLAN 10 con nombre “CasaDeJorge”
- VLAN 20 con nombre “CasaDeLaProfa”
- VLAN 30 con nombre “Escuela”
- VLAN 99 con nombre “Security”

Así mismo, se dividieron todos los puertos de los switches 4, 5 y 6 entre las 3 VLAN creadas, configurando sus enlaces troncales para comunicarse entre ellas y con otras partes de la topología. A continuación se muestra el switch 4 como ejemplo:

```
ESW4#show vlan-switch
```

| VLAN | Name | Status | Ports |
|------|--------------------|-----------|--|
| 1 | default | active | |
| 10 | CasaDeJorge | active | Fa1/1, Fa1/2, Fa1/3, Fa1/4 Fa1/5 |
| 20 | CasaDeLaProfa | active | Fa1/6, Fa1/7, Fa1/8, Fa1/9 Fa1/10 |
| 30 | Escuela | active | Fa1/11, Fa1/12, Fa1/13, Fa1/14 Fa1/15 |
| 99 | Security | active | |
| 1002 | fddi-default | act/unsup | |
| 1003 | token-ring-default | act/unsup | |
| 1004 | fddinet-default | act/unsup | |
| 1005 | trnet-default | act/unsup | |

| VLAN | Type | SAID | MTU | Parent | RingNo | BridgeNo | Stp | BrdgMode | Trans1 | Trans2 |
|------|------|--------|------|--------|--------|----------|-----|----------|--------|--------|
| 1 | enet | 100001 | 1500 | - | - | - | - | - | 1002 | 1003 |
| 10 | enet | 100010 | 1500 | - | - | - | - | - | 0 | 0 |
| 20 | enet | 100020 | 1500 | - | - | - | - | - | 0 | 0 |
| 30 | enet | 100030 | 1500 | - | - | - | - | - | 0 | 0 |
| 99 | enet | 100099 | 1500 | - | - | - | - | - | 0 | 0 |


```
ESW4#show int trunk
```

| Port | Mode | Encapsulation | Status | Native vlan |
|-------|------|---------------|----------|-------------|
| Fa1/0 | on | 802.1q | trunking | 99 |


```
Port Fa1/0 Vlans allowed on trunk 1-4094
```



```
Port Fa1/0 Vlans allowed and active in management domain 1,10,20,30,99
```



```
Port Fa1/0 Vlans in spanning tree forwarding state and not pruned 1,10,20,30,99
```

INTER-VLAN

Se configuró el router 3 como router on-a-stick para permitir el tráfico de las 3 VLAN creadas anteriormente por medio de un sólo puerto, para ello se crearon subinterfaces en el puerto fa1/0 y se configuró como troncal, tal como se muestra a continuación:

```
R3#show int fa1/0.10
FastEthernet1/0.10 is up, line protocol is up
  Hardware is AmdFE, address is c403.0904.0010 (bia c403.0904.0010)
  Internet address is 192.168.123.17/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 10.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
R3#show int fa1/0.20
FastEthernet1/0.20 is up, line protocol is up
  Hardware is AmdFE, address is c403.0904.0010 (bia c403.0904.0010)
  Internet address is 192.168.123.25/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 20.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
R3#show int fa1/0.30
FastEthernet1/0.30 is up, line protocol is up
  Hardware is AmdFE, address is c403.0904.0010 (bia c403.0904.0010)
  Internet address is 192.168.123.33/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 30.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
```

ACL Extendidas

Se implementaron ACL extendidas en todos los routers, aprovechando su uso para el servicio de SSH y NAT, para ello se crearon ACL extendidas nombradas, la de SSH con reglas que permitan TCP para cualquier máquina virtual, mientras que la de NAT permite múltiples servicios para los dispositivos que deseen salir de la red, esta última sólo es necesaria en el router 5. A continuación se muestra el router 5 como ejemplo:

```

R5#show access-list
Extended IP access list NAT
  10 permit icmp 192.168.123.0 0.0.0.255 any (4 matches)
  20 permit tcp 192.168.123.0 0.0.0.255 any
  30 permit udp 192.168.123.0 0.0.0.255 any
Extended IP access list SSH
  10 permit tcp host 192.168.123.3 any eq 22
  20 permit tcp host 192.168.123.4 any eq 22
  30 permit tcp host 192.168.123.11 any eq 22
  40 permit tcp host 192.168.123.43 any eq 22
  50 permit tcp host 192.168.123.44 any eq 22

```

NAT

Para poder conectarse a internet desde cualquier parte de la topología, se implementa NAT, esta se configura en el router 5, configurando el puerto fa4/0 con sobrecarga, como outside y asignando una ip automática con el comando “ip add dhcp”, los demás puertos del router se configuran como inside. Cabe mencionar que es necesario configurar rutas por defecto en todos los routers menos el 5, para que el tráfico desconocido sea redirigido hacia la NAT. Se muestra el funcionamiento de la NAT y la configuración antes mencionada en la siguiente imagen:

```

R5#show ip nat statistic
Total active translations: 5 (0 static, 5 dynamic; 5 extended)
Outside interfaces:
  FastEthernet4/0
Inside interfaces:
  FastEthernet0/0, FastEthernet0/1, FastEthernet1/0, FastEthernet2/0
Hits: 18 Misses: 0
CEF Translated packets: 18, CEF Punted packets: 0
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 2] access-list NAT interface FastEthernet4/0 refcount 5
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R5#show ip nat translation
Pro Inside global      Inside local           Outside local          Outside global
icmp 192.168.122.196:42550 192.168.123.2:42550 8.8.8.8:42550 8.8.8.8:42550
icmp 192.168.122.196:42806 192.168.123.2:42806 8.8.8.8:42806 8.8.8.8:42806
icmp 192.168.122.196:43062 192.168.123.2:43062 8.8.8.8:43062 8.8.8.8:43062
icmp 192.168.122.196:43318 192.168.123.2:43318 8.8.8.8:43318 8.8.8.8:43318
icmp 192.168.122.196:43574 192.168.123.2:43574 8.8.8.8:43574 8.8.8.8:43574

```

A continuación se muestra un ejemplo del uso de NAT para hacer ping desde una VPC:

```
PC1> ping 8.8.8.8

84 bytes from 8.8.8.8 icmp_seq=1 ttl=112 time=40.046 ms
84 bytes from 8.8.8.8 icmp_seq=2 ttl=112 time=23.290 ms
84 bytes from 8.8.8.8 icmp_seq=3 ttl=112 time=25.316 ms
84 bytes from 8.8.8.8 icmp_seq=4 ttl=112 time=30.091 ms
84 bytes from 8.8.8.8 icmp_seq=5 ttl=112 time=27.485 ms
```

Docker

Para empezar la instalación de los servidores es necesaria la instalación de dockers, para ello se debe realizar una conexión a través de SSH a la máquina virtual de Azure, luego se ejecutan los siguientes comandos:

- `sudo apt-get update`
- `sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ gnupg-agent \ software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"`
- `sudo apt-get update`
- `sudo apt-get install docker-ce docker-ce-cli containerd.io`

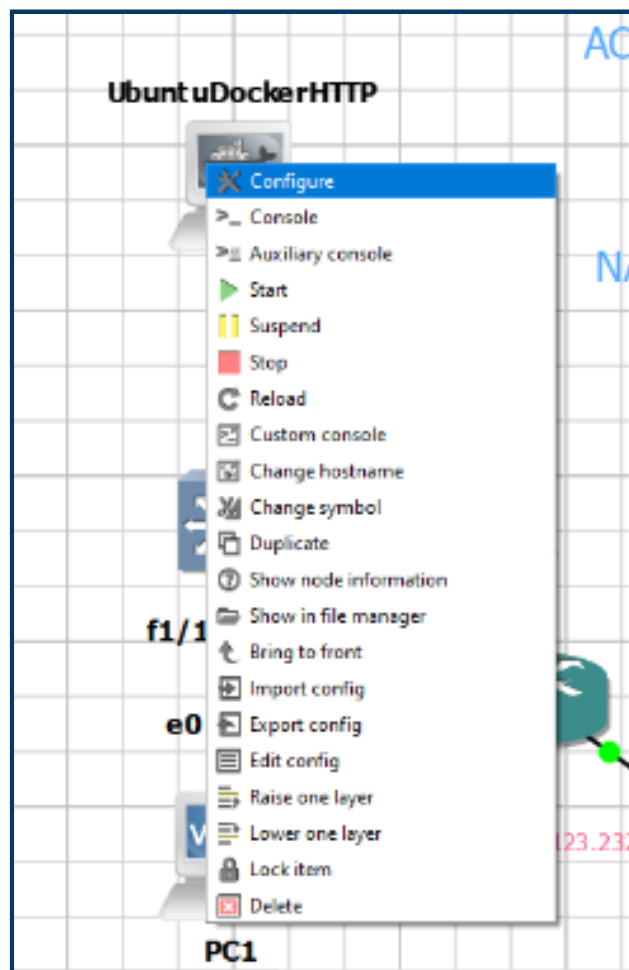
Una vez instalado se puede usar el comando “`sudo docker version`”. Posteriormente se instala una imagen de ubuntu con el comando “`sudo docker run -it ubuntu:18.04 /bin/bash`”, al terminar se debe ver la imagen con “`sudo docker ps`”, de la cual debemos copiar el “CONTAINER ID” y usarlo en el comando “`sudo docker start CONTAINERID`”.

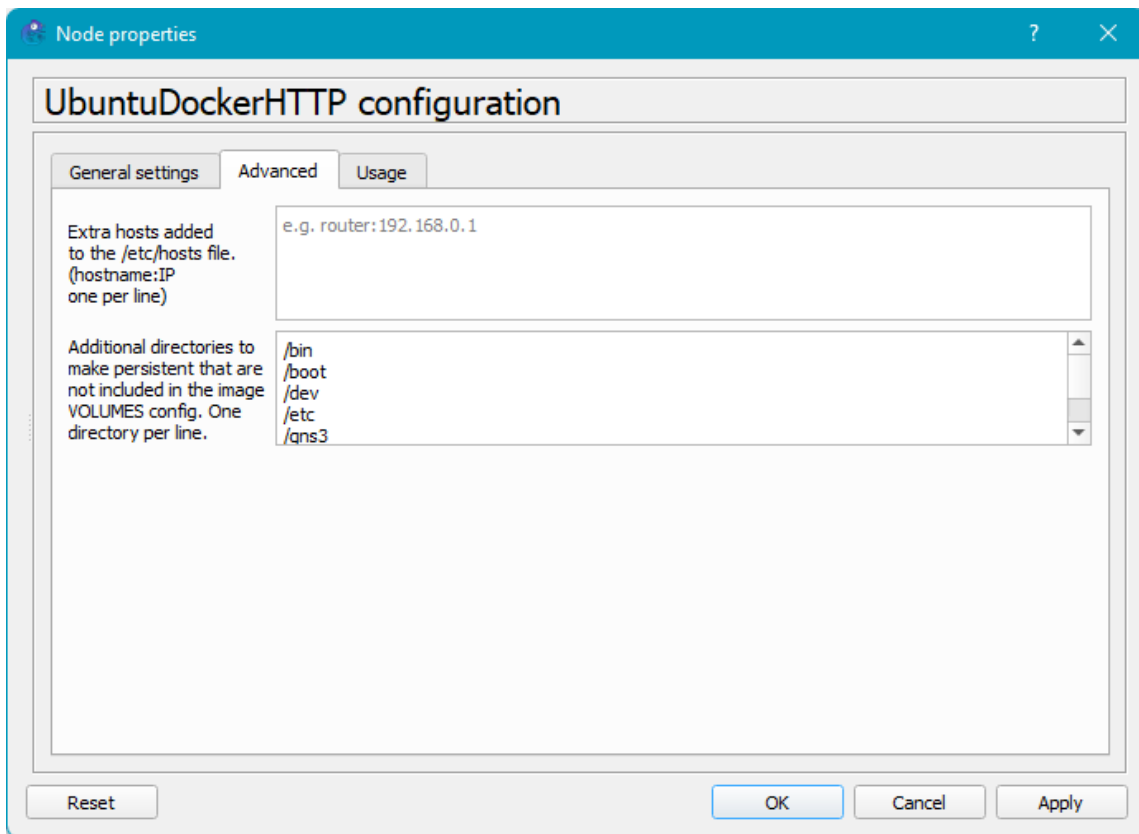
Dockerfile e Imagen de Dockerfile

Una vez se ha instalado Docker, podemos crear dockerfiles, este proceso se repite para todos los servidores, por tanto, no se va a mencionar más adelante. Primero debemos crear una carpeta para el servidor con el comando “mkdir nombre”, siendo nombre alguno como DNS, HTTP, SNMP, etc. Luego se debe ingresar a la carpeta creada y ejecutar “nano Dockerfile”, dentro se va a colocar las líneas de código para cualquier imagen de Docker que necesitemos más adelante, al terminar se debe guardar y ejecutar el comando “sudo docker build -t nombre:etiqueta .”, donde nombre y etiqueta pueden ser los que deseemos, pero aparecerán así en GNS3.

Persistencia de Docker y Configuración IP

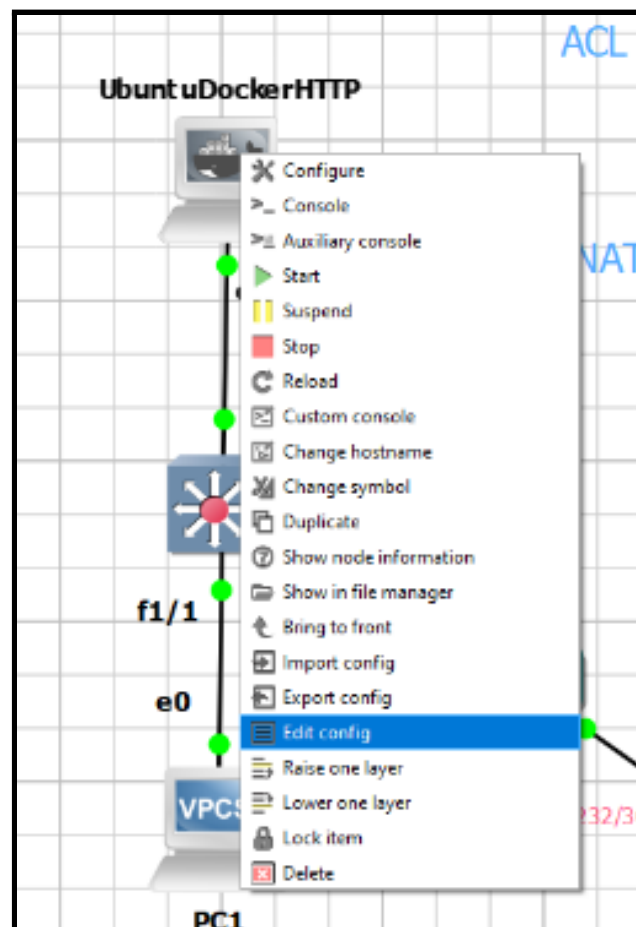
Al igual que los dockerfiles, cada que agregamos una máquina virtual en GNS3 se deben hacer dos cosas, primero, configurar la máquina para que al apagarla no se pierda la información, para ello se debe ir a opciones avanzadas y colocar lo siguiente:





- 1) /bin
- 2) /boot
- 3) /dev
- 4) /etc
- 5) /gns3
- 6) /gns3volumes
- 7) /home
- 8) /lib
- 9) /lib64
- 10) /root
- 11) /sbin
- 12) /var
- 13) /usr

Por otra parte, se debe ir a la configuración de la IP de la siguiente manera:



```
UbuntuDockerHTTP interfaces

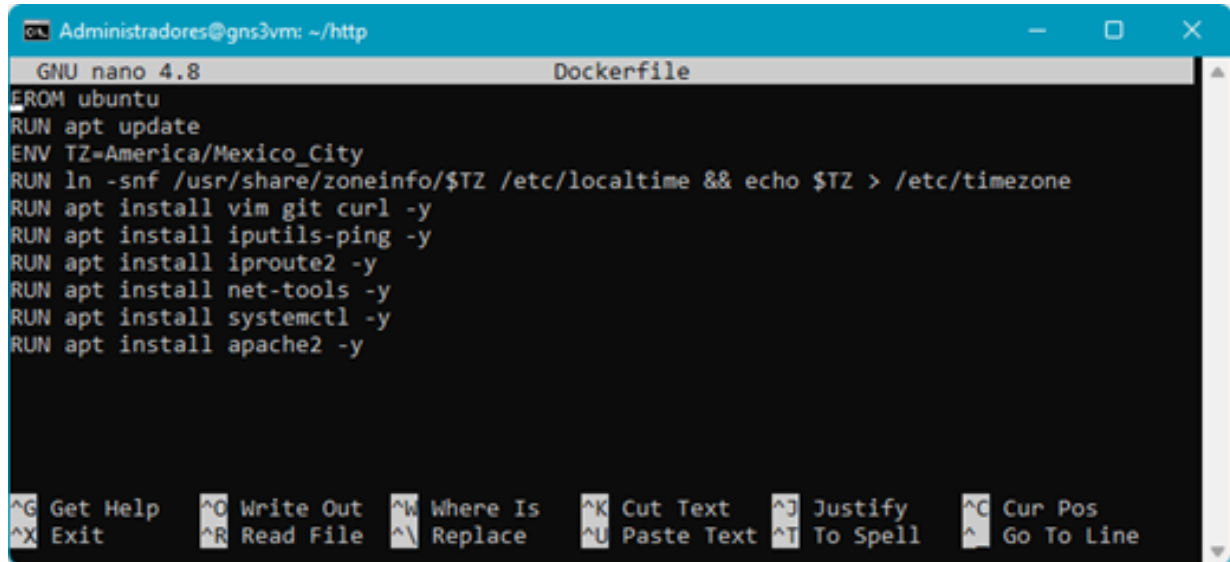
#
# This is a sample network config uncomment lines to configure the network
#

# Static config for eth0
auto eth0
iface eth0 inet static
    address 192.168.123.4
    netmask 255.255.255.248
    gateway 192.168.123.1
    up echo nameserver 192.168.123.11 > /etc/resolv.conf

# DHCP config for eth0
# auto eth0
# iface eth0 inet dhcp
```

HTTP

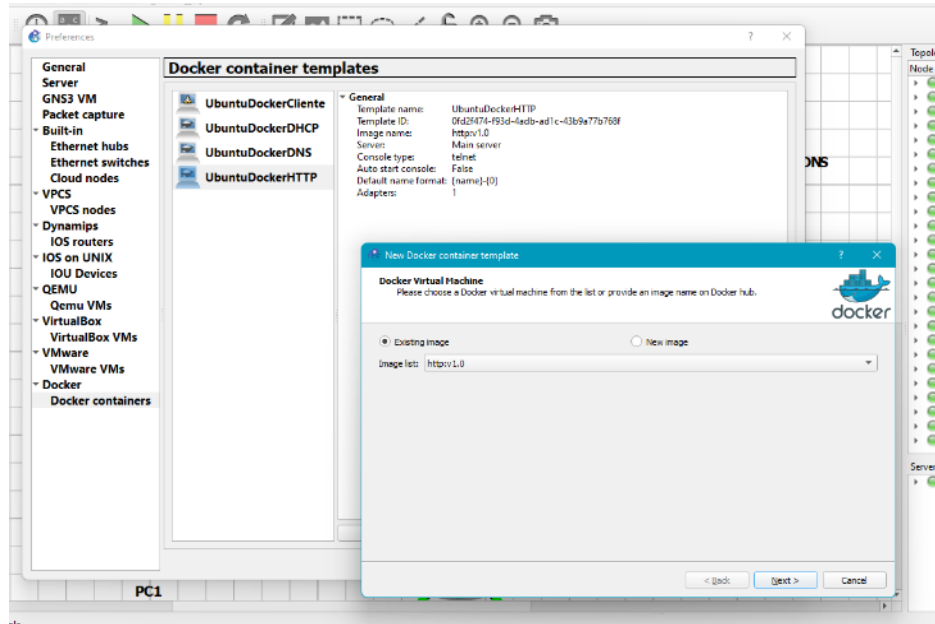
Para el servicio de HTTP creamos nuestra imagen de docker a partir del siguiente dockerfile:



```
Administradores@gns3vm: ~/http
GNU nano 4.8 Dockerfile
FROM ubuntu
RUN apt update
ENV TZ=America/Mexico_City
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
RUN apt install vim git curl -y
RUN apt install iputils-ping -y
RUN apt install iproute2 -y
RUN apt install net-tools -y
RUN apt install systemctl -y
RUN apt install apache2 -y

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^I Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

Luego instalamos la imagen en GNS3 dejando todas las opciones por defecto:



Después de realizar la configuración para persistencia de datos e ip estática se puede entrar a la máquina e iniciar el servidor con “systemctl start apache2”, y se puede probar el servicio tal como se muestra a continuación:

```

root@UbuntuDockerCliente:~# curl 192.168.123.4
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;

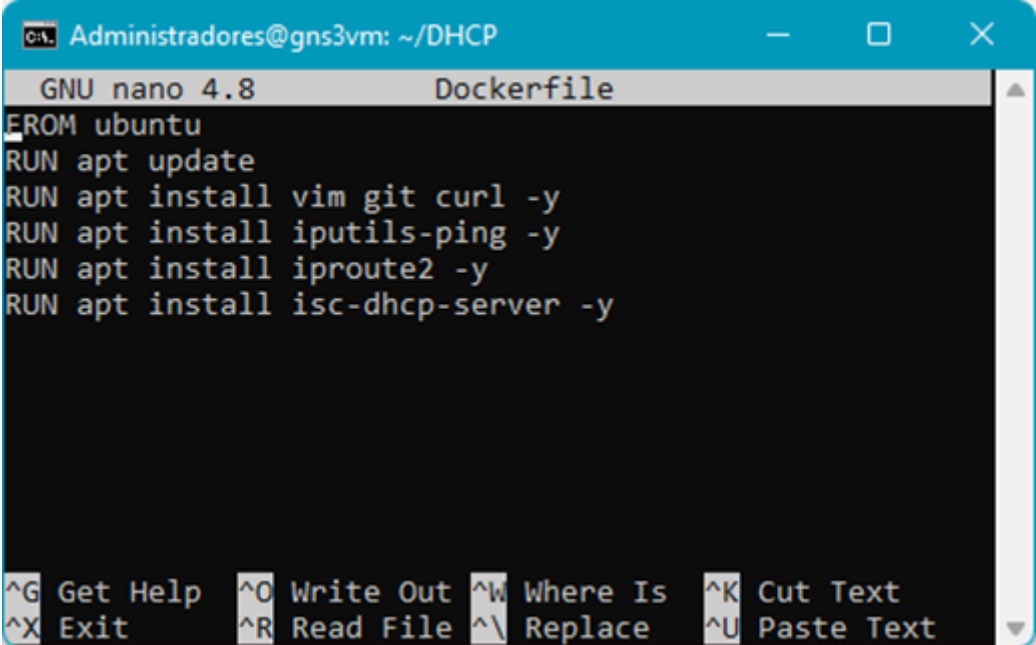
        background-color: #D8DBE2;

        font-family: Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
      }

```

DHCP

Para el servicio de DHCP creamos nuestra imagen de docker a partir del siguiente dockerfile:



The screenshot shows a terminal window titled 'Administradores@gns3vm: ~/DHCP'. Inside, the GNU nano 4.8 editor is open, editing a file named 'Dockerfile'. The content of the Dockerfile is as follows:

```

FROM ubuntu
RUN apt update
RUN apt install vim git curl -y
RUN apt install iputils-ping -y
RUN apt install iproute2 -y
RUN apt install isc-dhcp-server -y

```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^X Exit, ^R Read File, ^\ Replace, and ^U Paste Text.

A continuación se debe instalar la imagen en GNS3 y configurar como con HTTP. Luego debemos configurar el archivo “/etc/default/isc-dhcp-server” como se muestra en la siguiente imagen:

```
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="eth0"
INTERFACESv6=""

~
~
```

También se debe configurar el archivo “/etc/dhcp/dhcpd.conf” de acuerdo a las ip restantes de cada subred, junto con las correspondientes a los servidores, de la siguiente manera:

```

#R1
subnet 192.168.123.0 netmask 255.255.255.248{
    range 192.168.123.5 192.168.123.6;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
    option routers 192.168.123.1;
}

#R2
subnet 192.168.123.8 netmask 255.255.255.248{
    range 192.168.123.13 192.168.123.14;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
    option routers 192.168.123.1;
}

#R3 VLAN 10
subnet 192.168.123.16 netmask 255.255.255.248{
    range 192.168.123.21 192.168.123.22;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
    option routers 192.168.123.1;
}

#R3 VLAN 20
subnet 192.168.123.24 netmask 255.255.255.248{
    range 192.168.123.29 192.168.123.30;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
    option routers 192.168.123.1;
}

#R3 VLAN 30
subnet 192.168.123.32 netmask 255.255.255.248{
    range 192.168.123.37 192.168.123.38;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
}
"/etc/dhcp/dhcpd.conf" 71L, 1614C

```

```

#R4
subnet 192.168.123.40 netmask 255.255.255.248{
    range 192.168.123.45 192.168.123.46;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option subnet-mask 255.255.255.248;
    option routers 192.168.123.1;
}

#servidor SNMP
#host snmp{
#    hardware ethernet ;
#    fixed-address 192.168.123.3;
#}

#servidor HTTP
host http{
    hardware ethernet e6:c0:8d:d3:de:2e;
    fixed-address 192.168.123.4;
}

#servidor DNS
host dns{
    hardware ethernet aa:6b:7a:bd:2f:a3;
    fixed-address 192.168.123.11;
}

#cliente
host cliente{
    hardware ethernet 3a:31:f5:7f:76:10;
    fixed-address 192.168.123.44;
}

```

Cabe mencionar que para obtener “hardware ethernet” de los servidores, es necesario ingresar a cada uno y ejecutar el comando “ifconfig”. Además de esto, es necesario entrar a todos los puertos de los routers y utilizar “ip helper-address 192.168.123.43”, esto para que se reenvie el tráfico hacia el servidor DHCP cuando sea necesario. Una vez configurado esto, se puede iniciar el servicio dhcp con “service isc-dhcp-server start” y probar, de manera que el servidor DHCP asigna una ip distinta la original en caso que ya tener una, como ocurre en el siguiente caso:

```
PC1>
PC1> ip dhcp
DDORA IP 192.168.123.6/29 GW 192.168.123.1

PC1> show ip

NAME           : PC1[1]
IP/MASK        : 192.168.123.6/29
GATEWAY        : 192.168.123.1
DNS            : 8.8.8.8  8.8.4.4
DHCP SERVER    : 192.168.123.43
DHCP LEASE     : 43198, 43200/21600/37800
MAC            : 00:50:79:66:68:01
LPORT          : 20122
RHOST:PORT     : 127.0.0.1:20123
MTU            : 1500
```

SSH

Para SSH no se requiere un dockerfile, se debe ingresar a cada una de las máquinas virtuales e instalar un servidor ssh con “apt install openssh-server”, posteriormente se debe configurar SSH en los routers, para ello se requieren los siguientes comandos:

- ip domain-name TeamJorge.net
- username admin privilege 15 secret admin
- crypto key generate rsa -> 2048
- ip ssh version 2
- line vty 0 15 -> login local -> transport input ssh -> access-class SSH in

Finalmente, se puede probar SSH hacia la máquina virtual de Azure o hacia cualquiera de los routers. En la siguiente imagen se muestra la conexión al router 5:


```

root@UbuntuDockerCliente:~# ssh -oKexAlgorithms+=diffie-hellman-group1-sha1 -c 3des-cbc admin@192.168.123.234
The authenticity of host '192.168.123.234 (192.168.123.234)' can't be established.
RSA key fingerprint is SHA256:0qt8K7ghnfEfBgRoSqbvD4d5agbx+fk4YCtLWhs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.123.234' (RSA) to the list of known hosts.
Password:

R5#show int fa0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c405.a92f.0000 (bia c405.a92f.0000)
  Internet address is 192.168.123.234/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:06, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    706 packets input, 103856 bytes
      Received 668 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog
    0 input packets with dribble condition detected
  1273 packets output, 127466 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out

```

DNS

El último servicio implementado fue DNS, para ello se debe instalar una imagen de Docker predefinida de la siguiente página de GNS3:

<https://www.gns3.com/marketplace/appliances/dns>

Luego lo instalamos en GNS3, pero antes de usarlo debemos configurar todos los routers, para ello, se activa la búsqueda DNS con “ip domain lookup” y se configura la ip de DNS con “ip name-server 192.168.123.11”. A continuación se muestra el router 1 como ejemplo:

```

R1#show ip name-server
192.168.123.11

```

Así mismo, es necesario configurar el DNS en las VPCS como se muestra en la siguiente imagen:

```
PC1> ip dns 192.168.123.11

PC1> show ip

NAME           : PC1[1]
IP/MASK        : 192.168.123.2/29
GATEWAY        : 192.168.123.1
DNS            : 192.168.123.11  8.8.4.4
MAC            : 00:50:79:66:68:00
LPORT         : 20122
RHOST:PORT     : 127.0.0.1:20123
MTU            : 1500
```

Además, se deben configurar las máquinas virtuales para que también puedan usar DNS, para ello se debe eliminar el archivo “/etc/resolv.conf” y volver a crearlo con “nano /etc/resolv.conf”, colocando sólo la línea “nameserver 192.168.123.11”. Con esto sólo faltaría configurar el servidor de DNS, para ello se entra al archivo “/etc/hosts” y se colocan los dominios a usar:

```
GNU nano 4.8 /etc/hosts
127.0.1.1    UbuntuDockerDNS
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
192.168.123.3  SNMP.lab  SNMP
192.168.123.4  HTTP.lab   HTTP
192.168.123.11 DNS.lab   DNS
192.168.123.43 DHCP.lab  DHCP

[ Read 11 lines ]
^G Get Help  ^O Write Out ^K Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Finalmente, sólo falta entrar al archivo “/etc/dnsmasq.conf” y descomentar la siguiente línea:

```
GNU nano 4.8 /etc/dnsmasq.conf
# and this sets the source (ie local) address used to talk to
# 10.1.2.3 to 192.168.1.1 port 55 (there must be an interface with that
# IP on the machine, obviously).
# server=10.1.2.3@192.168.1.1#55

# If you want dnsmasq to change uid and gid to something other
# than the default, edit the following lines.
user=root
group=root

# If you want dnsmasq to listen for DHCP and DNS requests only on
# specified interfaces (and the loopback) give the name of the
# interface (eg eth0) here.
# Repeat the line for more than one interface.
interface=eth0
# Or you can specify which interface _not_ to listen on

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Con esto, es posible usar DNS desde cualquier dispositivo de la red, como ping de la VPC3 al servidor DHCP, que se encuentran todos situados en otros puntos de la topología:

```
PC3> ping DHCP.lab
DHCP.lab resolved to 192.168.123.43

DHCP.lab icmp_seq=1 timeout
84 bytes from 192.168.123.43 icmp_seq=2 ttl=61 time=42.576 ms
84 bytes from 192.168.123.43 icmp_seq=3 ttl=61 time=47.518 ms
84 bytes from 192.168.123.43 icmp_seq=4 ttl=61 time=36.830 ms
84 bytes from 192.168.123.43 icmp_seq=5 ttl=61 time=37.647 ms
```

También desde la máquina virtual UbuntuDockerCliente al servidor HTTP:

```
root@UbuntuDockerCliente:~# curl HTTP.lab

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }
    </style>
  </head>
  <body>
    <div style="text-align:center">
      <img alt="Ubuntu logo" data-bbox="347 112 652 212"/>
      <br/>
      Apache2 Ubuntu Default Page: It works
    </div>
  </body>
</html>
```

Conclusiones

El estudio de redes de computadoras es un campo muy amplio en la actualidad, el cual ha ido en crecimiento, así como en una mejora continua, razón por la cual seguirá vigente por mucho tiempo. Es por medio de este trabajo que resalta el alcance que tiene hoy en día, donde se tuvo que emplear todos los conocimientos adquiridos durante el curso actual, así como la adquisición de muchos otros que servirán más adelante, trabajando con tecnologías como Azure y GNS3, lo cual lleva mucho trabajo de por medio.

Por otra parte, aunque no se consiguió implementar SNMP, los resultados fueron los esperados, por las limitantes como el tiempo disponible y los conocimientos que requería este proyecto. Es con esta investigación que se comprende mejor la importancia de las redes en la vida cotidiana, pero más aún como estudiantes de Ingeniería en Sistemas Computacionales.

Referencias

ACL-Extendidas:

Setup Command. (2021) ▷ Configuración de ACL Extendidas IPv4 » CCNA desde Cero. Obtenido Diciembre 12, 2021, de <https://ccnadesdecero.es/configurar-acl-extendidas/>

Azure:

Ambit Team. (2021) Microsoft Azure, el cloud de Microsoft: ¿qué es y para qué sirve?. Obtenido Diciembre 12, 2021, de <https://www.ambit-bst.com/blog/microsoft-azure-el-cloud-de-microsoft-qu%C3%A9-es-y-para-qu%C3%A9-sirve>

DHCP:

Removing The 'Ac. (2021) Qué es el DHCP y cómo funciona - IONOS. Obtenido Diciembre 12, 2021, de <https://www.ionos.mx/digitalguide/servidores/configuracion/que-es-el-dhcp-y-como-funciona/>

Marouane. (2021) Install and Configure a DHCP Server in Ubuntu Server and GNS3 emulator - Part 1 - YouTube. Obtenido Diciembre 12, 2021, de <https://www.youtube.com/watch?v=qKFWYDBHPT8>

Tola, Aitzol. (2021) Cómo instalar un servidor DHCP en Ubuntu Server - Linux Básico. Obtenido Diciembre 12, 2021, de <https://linuxbasico.com/dhcp-ubuntu/>

Francisco Periañez Gómez. (2021) option domain-name-servers | Tutorial del servicio DHCP. Obtenido Diciembre 12, 2021, de https://www.fpgenred.es/DHCP/option_domainnameservers.html

Redes Plus. (2021) 🇵🇪 DHCP LINUX 💻 INSTALAR y configurar isc DHCP 🐧 Ubuntu - YouTube. Obtenido Diciembre 12, 2021, de <https://www.youtube.com/watch?v=sKBhQAojCvk>

DNS:

(2021) Las DNS Cloudflare: 1.1.1.1 y todas sus variantes | Ayuda Ley Datos. Obtenido Diciembre 12, 2021, de <https://ayudaleyprotecciondatos.es/dns/cloudflare/>

David Bombal (2021). GNS3 Talks: Easy DNS Server for GNS3 Topologies: Dnsmasq Docker Appliance Part 1. Obtenido Diciembre 13, 2021, de https://www.youtube.com/watch?v=86MluxQ-LtI&t=427s&ab_channel=DavidBombal

Docker:

Last, First. (2021) 7.GNS3-Network Automation With Python/How to import/install Ubuntu Docker containers into GNS3 - YouTube. Obtenido Diciembre 12, 2021, de <https://www.youtube.com/watch?v=UKs1aEmhvdg>

Amazon. (2021) Contenedores de Docker | ¿Qué es Docker? | AWS. Obtenido Diciembre 12, 2021, de <https://aws.amazon.com/es/docker/>

HTTP:

Mozilla (2021) HTTP | MDN. Obtenido Diciembre 12, 2021, de <https://developer.mozilla.org/es/docs/Web/HTTP>

Inter-VLAN:

Last, First. (2021) Configuración de Inter-VLAN Routing (Router on a stick) -. Obtenido Diciembre 12, 2021, de <http://theosnews.com/2013/03/15409/>

NAT:

Speedcheck. (2021) ¿Qué es NAT?. Obtenido Diciembre 12, 2021, de <https://www.speedcheck.org/es/wiki/nat/>

Clicking The Browse. (2021) Connect GNS3 to the Internet (local server) | GNS3 Documentation. Obtenido Diciembre 12, 2021, de <https://docs.gns3.com/docs/using-gns3/advanced/connect-gns3-internet/>

OSPF:

IBM. (2021) IBM Docs. Obtenido Diciembre 12, 2021, de <https://www.ibm.com/docs/es/i/7.3?topic=routing-open-shortest-path-first>

Raúl Prieto Fernández. (2021) Enrutamiento dinámico OSPF con Packet Tracer. Obtenido Diciembre 12, 2021, de <https://www.raulprietofernandez.net/blog/packet-tracer/enrutamiento-dinamico-ospf-c-on-packet-tracer>

SNMP:

Manageengine. (2021) ¿Qué es SNMP? | Protocolo SNMP – Monitorización – Puerto SNMP - ManageEngine OpManager. Obtenido Diciembre 12, 2021, de <https://www.manageengine.com/es/network-monitoring/what-is-snmp.html>

SSH:

The Server. (2021) ssh unable to negotiate - no matching key exchange method found - Unix & Linux Stack Exchange. Obtenido Diciembre 12, 2021, de <https://unix.stackexchange.com/questions/402746/ssh-unable-to-negotiate-no-matching-key-exchange-method-found>

Red Hat Enterprise (2021) Protocolo SSH. Obtenido Diciembre 12, 2021, de <https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>

VLAN:

Redes Zone. (2021) Qué son las VLAN, para qué sirven y cómo funcionan con ejemplos de uso. Obtenido Diciembre 12, 2021, de <https://www.redeszone.net/tutoriales/redes-cable/vlan-tipos-configuracion/>

VLSM:

Tech Club. (2021) Concepto VLSM. - TechClub Tajamar. Obtenido Diciembre 12, 2021, de <https://techclub.tajamar.es/concepto-vlsm/>

Otros comandos:

ProTechFurus. (2021) How to Use VPCS in GNS3 - A Step By Step Explanation. Obtenido Diciembre 12, 2021, de <http://protechgurus.com/how-to-use-vpcs-in-gns3/>