Instituto Politécnico Nacional

Escuela Superior de Cómputo

PROYECTO FINAL

Materia:
Administración de servicios en red

Grupo:
4CV13

Profesor:
Henestrosa Carrasco Leticia

Integrantes: (*Equipo 6*)
Arévalo Andrade Miguel Ángel
Castro Cruces Jorge Eduardo
López Mares Irene Elizabeth
Pedroza García Rodolfo

Fecha:

16 de junio de 2022

Índice

Introducción **2**

Objetivos 4

Desarrollo **4**

Topología 5

Azure 6

RIP v2 8

VLAN 11

INTER-VLAN 16

ACL 17

NAT 17

Docker 20

HTTP 32

DHCP 36

SSH 42

DNS 46

Conclusión **50**

Referencias **51**

Anexos **53**

Introducción

El término “red de computadoras” hace referencia al conjunto de computadoras autónomas interconectadas mediante una sola tecnología.

Existen diferentes tipos de redes, entre las cuales cabe destacar:

- **LAN:** Es una red de área local cuyo alcance se restringe a un radio de unos cuantos cientos de metros.
- **VLAN:** Es una red virtual que se instala en una red local switcheada tomando como base el número del puerto de switch al cual se conecta la computadora.
- **Internet:** Es un conjunto descentralizado de redes de comunicaciones interconectadas, que utilizan la familia de protocolos TCP/IP, constituye una red lógica única de alcance mundial.

El enrutamiento es la manera en la que un paquete busca la ruta más corta para llegar a su destino dentro de una red.

Los protocolos de enrutamiento administran la actividad de enrutamiento en un sistema. Los enrutadores intercambian información de enrutamiento con otros hosts para mantener las rutas conocidas a las redes remotas. Para este proyecto se utilizó el protocolo de enrutamiento RIP v2:

- **Routing Information Protocol versión 2 (RIPv2)** es uno de los protocolos de enrutamiento interior más sencillos y utilizados. Es un protocolo de vector distancia de tipo estándar, basado en los RFC 1388, 1723 y 2453. Su principal limitación está impuesta por la cantidad máxima de saltos que soporta 15: RIP asume que todo lo que se encuentra a más de 15 saltos, está a una distancia infinita, y por lo tanto no tiene ruta válida. Sin embargo, para nuestra topología no existe esa desventaja, es por eso que decidimos implementar RIPv2.
- Se aplicará direccionamiento con **VLSM (Máscara de Subred de Longitud Variable)**, en la división en subredes tradicional se asigna la misma cantidad de direcciones a cada subred. Sin embargo, las subredes que requieren menos direcciones tienen direcciones sin utilizar (desperdiciadas). Por ejemplo, los enlaces WAN solo necesitan dos direcciones. Así que, la máscara de subred de longitud variable (VLSM), permite un uso más eficiente de las direcciones.

Los mecanismos básicos de seguridad serán implementados con: VLAN, NAT y ACL EXTENDIDAS:

- **NAT (Network Address Translation/Traducción de Direcciones de Red)** tiene muchos usos, pero el principal es conservar las direcciones IPv4 públicas.
  Esto se logra al permitir que las redes utilicen direcciones IPv4 privadas internamente y al proporcionar la traducción a una dirección pública solo cuando sea necesario.
- **Port Address Translations (PAT)** es una extensión de la traducción de direcciones de red (NAT) que permite asignar varios dispositivos en una LAN a una única dirección IP pública para conservar las direcciones IP. En otras palabras, se puede utilizar una única dirección IPv4 pública para cientos, incluso miles de direcciones IPv4 privadas internas.
- 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.

Los servicios que habilitaremos en las máquinas virtuales son SNMP, DNS, DHCP, HTTP y TFTP:

- **SNMP (Protocolo Simple de Administración de Red)** es un protocolo de capa de aplicación que proporciona un formato de mensaje para la comunicación entre administradores y agentes. Se desarrolló para permitir que los administradores puedan administrar los nodos, como los servidores, las estaciones de trabajo, los routers, los switches y los dispositivos de seguridad, en una red IP. Permite que los administradores de red administren el rendimiento de la red, detecten y resuelvan problemas de red, y planifiquen el crecimiento de la red.
- **DNS (Domain Name System/Sistema de Nombres de Dominio)** es un sistema de nombres jerárquico que permite la comunicación entre dispositivos en una red. Traduce nombres de dominio legibles para humanos a una dirección de Protocolo de Internet (IP) que una computadora puede entender. Esencialmente, DNS nos permite conectarnos a sitios web sin tener que memorizar una serie de números; todo lo que necesitamos saber es el nombre del sitio web.
- **DHCP (Dynamic Host configuration Protocol)**, es un servidor de Red el cual permite una asignación automática de direcciones IP, gateways predeterminadas, así como otros parámetros de red que necesiten los clientes. El sistema DHCP envía automáticamente todos los parámetros para que los clientes se comuniquen sin problema dentro de la red.
- **HTTP (Protocolo de transferencia de hipertexto)** es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente (se le suele llamar "agente de usuario", en inglés user agent) realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor (se le suele llamar un servidor web) le envía un mensaje de respuesta.
- **TFTP (Protocolo trivial de transferencia de archivos)** es un protocolo simple que proporciona una función básica de transferencia de archivos sin autenticación de usuario. TFTP está destinado a las aplicaciones que no necesitan las interacciones sofisticadas que proporciona el protocolo de transferencia de archivos (FTP).

Para la creación de la topología y la implementación de los servicios se utilizará la herramienta de GNS3.

- **GNS3** es un simulador gráfico de red lanzado en 2008, que permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos,123 permitiendo la combinación de dispositivos tanto reales como virtuales.

Objetivos

Los objetivos de este proyecto son:

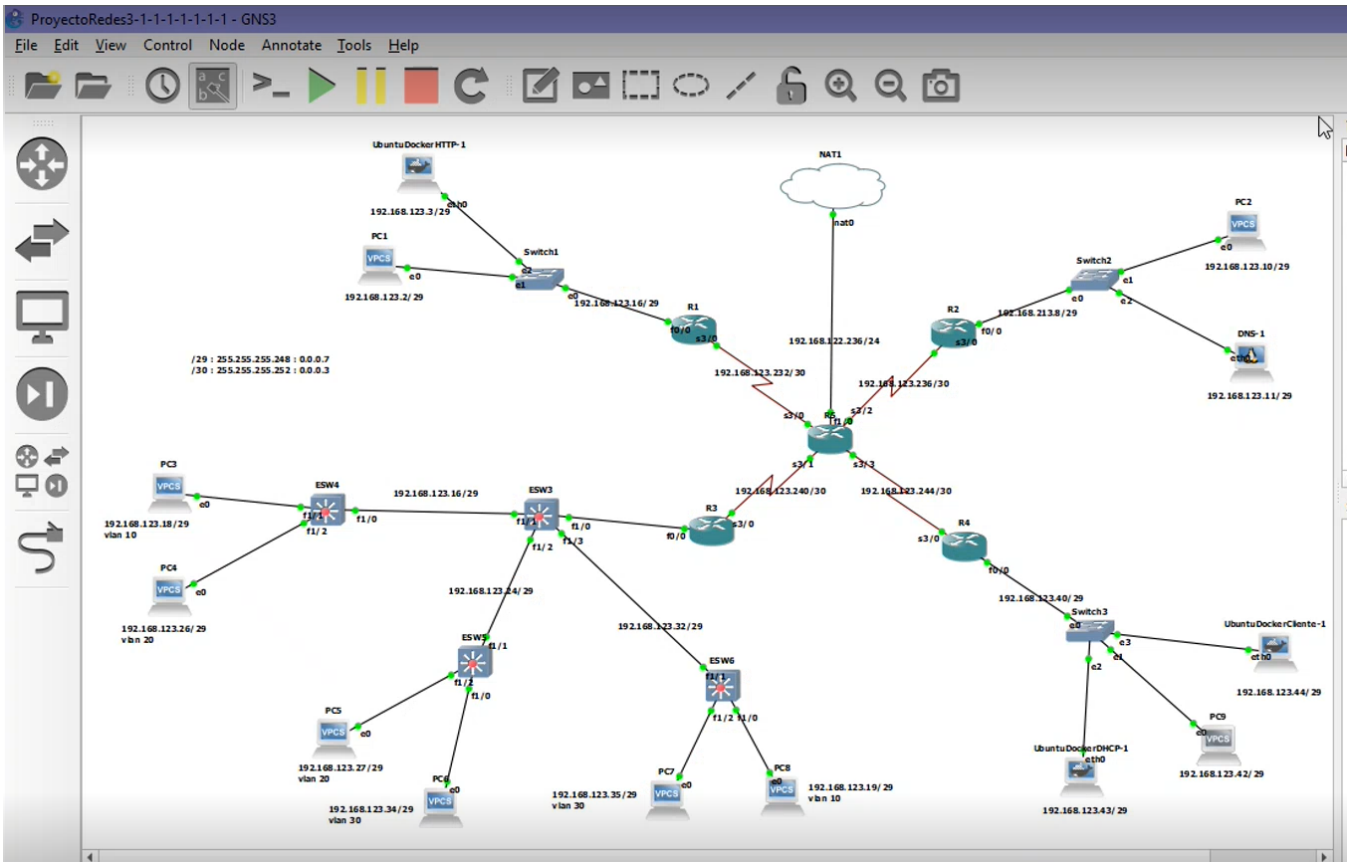
- Utilizar GNS3 como herramienta de simulación de redes.
- Crear y administrar nuestra propia red.
- Implementar servicios básicos de una red.

- Aplicar los conocimientos adquiridos durante el curso de
- Administración de Servicios en Red.

Desarrollo

Para la realización del proyecto, se implementó la siguiente topología:

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	s3/0	192.168.123.233	255.255.255.252

	f1/0	192.168.123.1	255.255.255.248
R2	s3/0	192.168.123.237	255.255.255.252
	f1/0	192.168.123.9	255.255.255.248
R3	s3/0	192.168.123.241	255.255.255.252
		192.168.123.17	255.255.255.248
		192.168.123.25	255.255.255.248
		192.168.123.33	255.255.255.248
R4	s3/0	192.168.123.246	255.255.255.252
	f1/0	192.168.123.41	255.255.255.248
R5	s3/0	192.168.123.234	255.255.255.252
	s3/1	192.168.123.242	255.255.255.252
	s3/2	192.168.123.238	255.255.255.252
	s3/3	192.168.123.245	255.255.255.252
	f0/0	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

Azure

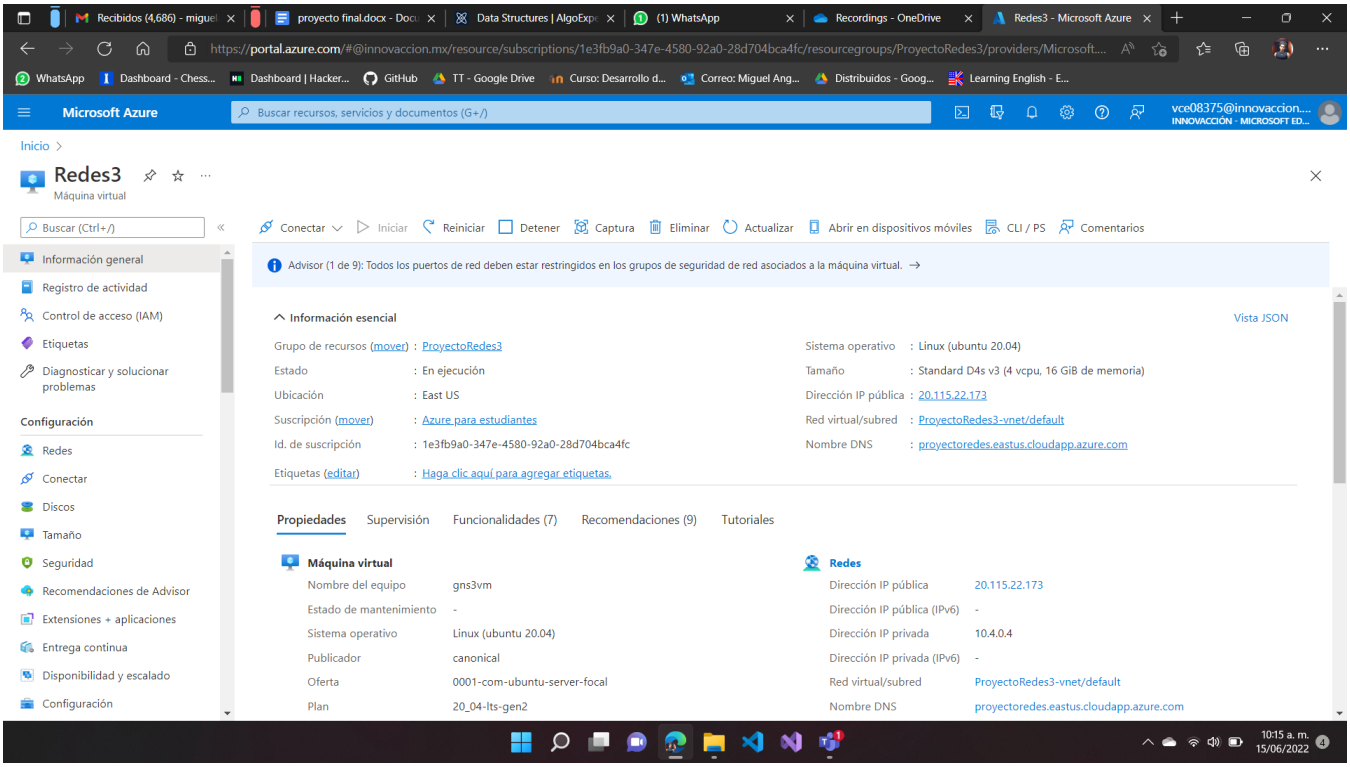
Azure es un conjunto de servicios en la nube de la empresa Microsoft. Con Azure es posible almacenar información y crear, administrar e implementar aplicaciones en cloud. Para utilizar Azure es necesario el pago de una cuota que recoge los servicios contratados. Desde el portal de Microsoft Azure se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

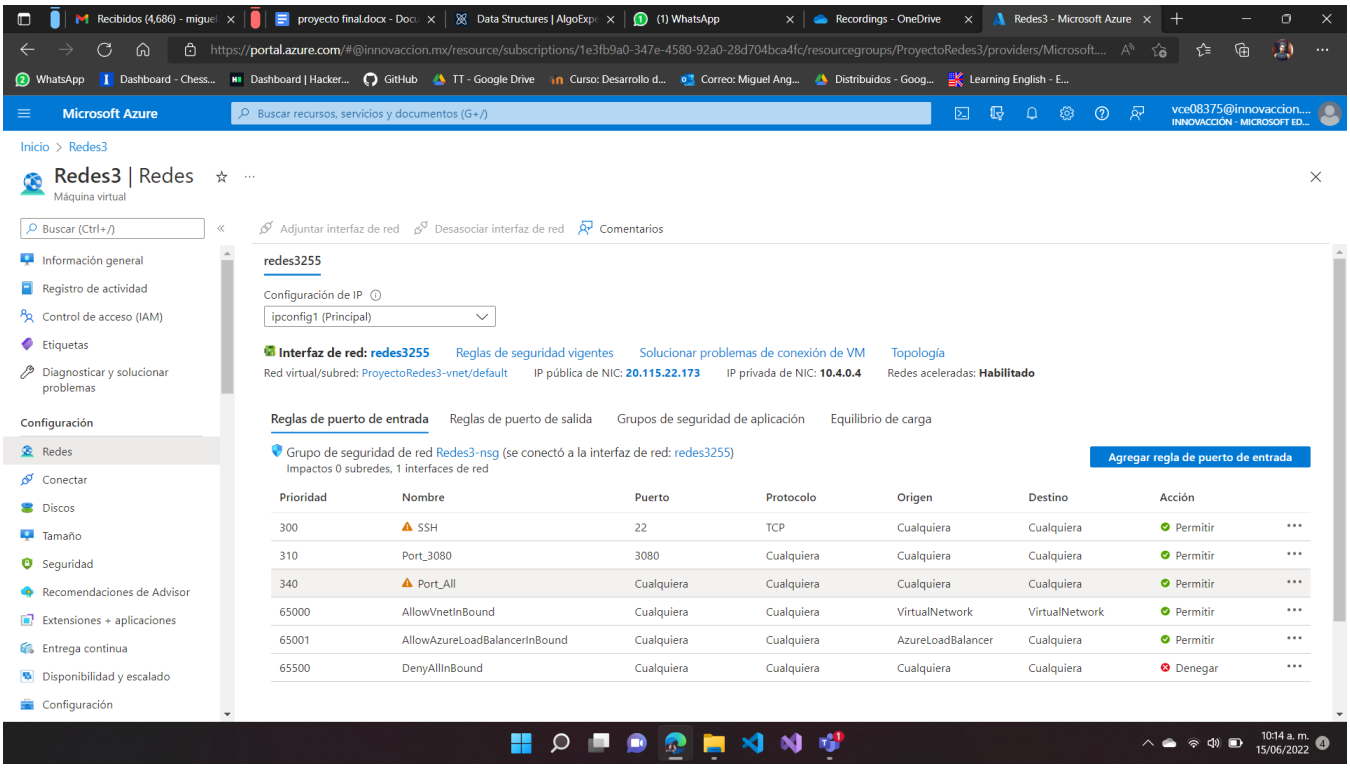
Máquina virtual Azure Redes 3

Para el proyecto creamos una máquina virtual utilizando Azure para alocar el servidor de GNS3, para poder trabajar correctamente, se tuvo que agregar reglas de seguridad para abrir el puerto 3080 y otras para los

puertos 5000 y 6000, con estas reglas agregadas podremos trabajar sin ningún problema con GNS3 utilizando esta máquina virtual como servidor.



Reglas de puerto de entrada



RIP v2

Se configuró RIP v2 para todos los routers, dando a conocer las redes a las que se encuentra conectado cada router. a continuación podemos ver el router 5 como ejemplo:

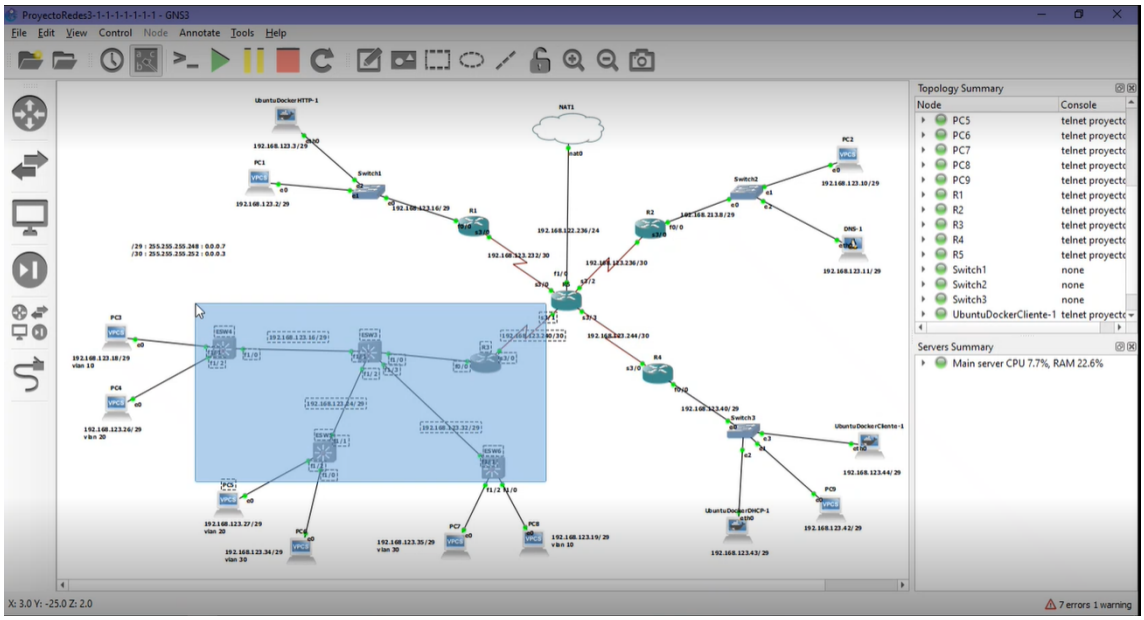
```
R5
*Jun 15 03:27:16.150: %SYS-5-CONFIG_I: Configured from console by console
R5#wr
Building configuration...
[OK]
R5#show ip name-ser
192.168.122.1
8.8.8.8
192.168.123.11

R5#show ip proto
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 26 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial3/0             2     2
    Serial3/1             2     2
    Serial3/2             2     2
    Serial3/3             2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.123.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.123.241      120      00:00:18
    192.168.123.246      120      00:00:13
    192.168.123.233      120      00:00:17
    192.168.123.237      120      00:00:20
  Distance: (default is 120)

R5#
R5#
```

```
R5
    192.168.123.246      120      00:00:13
    192.168.123.233      120      00:00:17
    192.168.123.237      120      00:00:20
  Distance: (default is 120)

R5#
R5#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.122.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.122.1
      192.168.122.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.122.0/24 is directly connected, FastEthernet1/0
L        192.168.122.236/32 is directly connected, FastEthernet1/0
      192.168.123.0/24 is variably subnetted, 14 subnets, 3 masks
R        192.168.123.0/29 [120/1] via 192.168.123.233, 00:00:09, Serial3/0
R        192.168.123.8/29 [120/1] via 192.168.123.237, 00:00:12, Serial3/2
R        192.168.123.16/29 [120/1] via 192.168.123.241, 00:00:09, Serial3/1
R        192.168.123.24/29 [120/1] via 192.168.123.241, 00:00:09, Serial3/1
R        192.168.123.32/29 [120/1] via 192.168.123.241, 00:00:09, Serial3/1
R        192.168.123.40/29 [120/1] via 192.168.123.246, 00:00:04, Serial3/3
C        192.168.123.232/30 is directly connected, Serial3/0
L        192.168.123.234/32 is directly connected, Serial3/0
C        192.168.123.236/30 is directly connected, Serial3/2
L        192.168.123.238/32 is directly connected, Serial3/2
C        192.168.123.240/30 is directly connected, Serial3/1
L        192.168.123.242/32 is directly connected, Serial3/1
C        192.168.123.244/30 is directly connected, Serial3/3
L        192.168.123.245/32 is directly connected, Serial3/3
R5#
```

VLAN

Para agregar seguridad a la red una de las implementaciones que hicimos en el proyecto fueron las VLAN's, con estas segmentamos en tres la red, cada una con los siguientes nombres:

- VLAN 10 con nombre "CasaPedroza"
- VLAN 20 con nombre "CasaMike"
- VLAN 20 con nombre "CasaLalo"



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 3 como ejemplo:

```
ESW3#show vlan-switch

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa1/4, Fa1/5, Fa1/6, Fa1/7
                                                Fa1/8, Fa1/9, Fa1/10, Fa1/11
                                                Fa1/12, Fa1/13, Fa1/14, Fa1/15
                                                Fa2/0, Fa2/1, Fa2/2, Fa2/3
                                                Fa2/4, Fa2/5, Fa2/6, Fa2/7
                                                Fa2/8, Fa2/9, Fa2/10, Fa2/11
                                                Fa2/12, Fa2/13, Fa2/14, Fa2/15
10   CasaPedroza                      active
20   CasaMike                         active
30   CasaLalo                         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
1002 fddi  101002     1500  -      -      -        -    -        1      1003
1003 tr    101003     1500  1005   0      -        -    srb      1      1002
1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
1005 trnet 101005     1500  -      -      1        ibm  -        0      0
ESW3#
```

```
ESW4#show interface trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa1/0     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa1/0     1-4094

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

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/0     1,10,20
ESW4#
```

```
ESW3#show inter trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa1/0     on           802.1q         trunking      1
Fa1/1     on           802.1q         trunking      1
Fa1/2     on           802.1q         trunking      1
Fa1/3     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa1/0     1-4094
Fa1/1     1-4094
Fa1/2     1-4094
Fa1/3     1-4094

Port      Vlans allowed and active in management domain
Fa1/0     1,10,20,30
Fa1/1     1,10,20,30
Fa1/2     1,10,20,30
Fa1/3     1,10,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/0     1,10,20,30
Fa1/1     1,10,20,30
Fa1/2     1,10,20,30
Fa1/3     1,10,20,30
ESW3#
```

INTER-VLAN

Para permitir el tráfico de las 3 VLAN se configuro el router 3 como un router on-a-stick, de esta forma solo fue necesario hacer uso de un puerto, esto se puedo realizar gracias a que creamos subinterfaces en el puerto fa0/0 y se configuró como troncal, podemos ver la configuración a continuación:

```
R3#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES NVRAM  up                    up
FastEthernet0/0.10     192.168.123.17  YES NVRAM  up                    up
FastEthernet0/0.20     192.168.123.25  YES NVRAM  up                    up
FastEthernet0/0.30     192.168.123.33  YES NVRAM  up                    up
FastEthernet1/0        unassigned      YES NVRAM  administratively down down
FastEthernet1/1        unassigned      YES NVRAM  administratively down down
GigabitEthernet2/0     unassigned      YES NVRAM  administratively down down
Serial3/0              192.168.123.241 YES NVRAM  up                    up
Serial3/1              unassigned      YES NVRAM  administratively down down
Serial3/2              unassigned      YES NVRAM  administratively down down
Serial3/3              unassigned      YES NVRAM  administratively down down
Ethernet4/0            unassigned      YES NVRAM  administratively down down
Ethernet4/1            unassigned      YES NVRAM  administratively down down
Ethernet4/2            unassigned      YES NVRAM  administratively down down
Ethernet4/3            unassigned      YES NVRAM  administratively down down
Ethernet4/4            unassigned      YES NVRAM  administratively down down
Ethernet4/5            unassigned      YES NVRAM  administratively down down
Ethernet4/6            unassigned      YES NVRAM  administratively down down
Ethernet4/7            unassigned      YES NVRAM  administratively down down
R3#
```

ACL

Se implementaron ACL extendidas en todos los routers, aprovechando sus uso para el servicio de ssh y NAT se nombraron las ACL's, 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 ultima configuracion solo se implemento en el router 5, que es el que da salida de la NAT. A continuación vemos el router 5 como ejemplo.

```
R5#show access-list
R5#show access-lists
Extended IP access list 100
    10 permit ip any any (121 matches)
Extended IP access list NAT
    10 permit tcp 192.168.123.0 0.0.0.255 any
    20 permit icmp 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 (30 matches)
R5#
```

solarwinds | Solar-PuTTY *free tool*

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 fa1/0 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
    10 permit tcp 192.168
.123.0 0.0.0.255 any
    20 permit icmp 192.16
8.123.0 0.0.0.255 any
    30 permit udp 192.168
.123.0 0.0.0.255 any
Extended IP access list S
SH
    10 permit tcp host 19
2.168.123.3 any eq 22
    20 permit tcp host 19
2.168.123.4 any eq 22
    30 permit tcp host 19
2.168.123.11 any eq 22
    40 permit tcp host 19
2.168.123.43 any eq 22
    50 permit tcp host 19
2.168.123.44 any eq 22 (3
0 matches)
R5#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Peak translations: 18, occurred 05:33:08 ago
Outside interfaces:
  FastEthernet1/0
Inside interfaces:
  Serial3/0, Serial3/1, Serial3/2, Serial3/3
Hits: 235  Misses: 0
CEF Translated packets: 125, CEF Punted packets: 52
Expired translations: 63
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 100 interface FastEthernet1/0 refcount 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R5#
```

PING a Google R5

```
R5
curred 05:33:08 ago
Outside interfaces:
  FastEthernet1/0
Inside interfaces:
  Serial3/0, Serial3/1, S
erial3/2, Serial3/3
Hits: 235  Misses: 0
CEF Translated packets: 1
25, CEF Punted packets: 5
2
Expired translations: 63
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 100 i
nterface FastEthernet1/0
refcount 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
R5#show ip nat trans
R5#show ip nat trans
R5#show ip nat translatio
ns
R5#ping 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/47/52 ms
R5#ping google.com
Translating "google.com"...domain server (192.168.122.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 142.251.45.14, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/50/56 ms
R5#
solarwinds | Solar-PuTTY free tool                    © 2019 Sc
```

PING a Google PC1

```
PC1> ping 8.8.8.8

84 bytes from 8.8.8.8 icmp_seq=1 ttl=53 time=49.507 ms
84 bytes from 8.8.8.8 icmp_seq=2 ttl=53 time=35.162 ms
84 bytes from 8.8.8.8 icmp_seq=3 ttl=53 time=22.483 ms
84 bytes from 8.8.8.8 icmp_seq=4 ttl=53 time=27.961 ms
84 bytes from 8.8.8.8 icmp_seq=5 ttl=53 time=25.302 ms

PC1> ping google.com
google.com resolved to 172.253.122.113

84 bytes from 172.253.122.113 icmp_seq=1 ttl=100 time=29.920 ms
84 bytes from 172.253.122.113 icmp_seq=2 ttl=100 time=36.521 ms
84 bytes from 172.253.122.113 icmp_seq=3 ttl=100 time=22.324 ms
84 bytes from 172.253.122.113 icmp_seq=4 ttl=100 time=41.218 ms
84 bytes from 172.253.122.113 icmp_seq=5 ttl=100 time=28.016 ms

PC1>
```

Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker le permite separar sus aplicaciones de su infraestructura para que pueda entregar software rápidamente. Con Docker, puede administrar su infraestructura de la misma manera que administra sus aplicaciones. Al aprovechar las metodologías de Docker para enviar, probar e implementar código rápidamente, puede reducir significativamente el retraso entre escribir código y ejecutarlo en producción.

Docker proporciona la capacidad de empaquetar y ejecutar una aplicación en un entorno poco aislado denominado contenedor. El aislamiento y la seguridad le permiten ejecutar muchos contenedores simultáneamente en un host determinado. Los contenedores son ligeros y contienen todo lo necesario para ejecutar la aplicación, por lo que no es necesario depender de lo que está instalado actualmente en el host. Puede compartir fácilmente contenedores mientras trabaja y asegurarse de que todas las personas con las que comparte obtengan el mismo contenedor que funcione de la misma manera.

Docker proporciona herramientas y una plataforma para administrar el ciclo de vida de sus contenedores:

- Desarrolle su aplicación y sus componentes de soporte utilizando contenedores.
- El contenedor se convierte en la unidad para distribuir y probar la aplicación.
- Cuando esté listo, implemente la aplicación en su entorno de producción, como un contenedor o un servicio orquestado. Esto funciona de la misma manera si su entorno de producción es un centro de datos local, un proveedor de nube o un híbrido de los dos.

Arquitectura de Docker

Docker utiliza una arquitectura cliente-servidor. El cliente de Docker habla con el daemon de Docker, que hace el trabajo pesado de crear, ejecutar y distribuir sus contenedores de Docker. El cliente y el daemon de Docker pueden ejecutarse en el mismo sistema, o puede conectar un cliente de Docker a un daemon de Docker remoto. El cliente y el daemon de Docker se comunican mediante una API REST, a través de sockets UNIX o una interfaz de red. Otro cliente de Docker es Docker Compose, que le permite trabajar con aplicaciones que consisten en un conjunto de contenedores.

Para la instalación de los servidores es necesaria la instalación de Docker, para ello se debe realizar una conexión a la máquina virtual de Azure usando ssh, 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*

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, DHCP, 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.

Instalación Docker

```
Seleccionar root@d22c88036dc3: /
W: Target CNF (main/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list.d/gns3-ubuntu-ppa-focal.list:1 and /etc/apt/sources.list.d/gns3.list:1
Redes3@gns3vm:~$ sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ gnupg-agent \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package  apt-transport-https
E: Unable to locate package  ca-certificates
E: Unable to locate package  curl
E: Unable to locate package  gnupg-agent
Redes3@gns3vm:~$ sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ gnupg-agent software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package  apt-transport-https
E: Unable to locate package  ca-certificates
E: Unable to locate package  curl
E: Unable to locate package  gnupg-agent
Redes3@gns3vm:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:80f31da1ac7b312ba29d65080fddf797dd76acfb870e677f390d5acba9741b17
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

Redes3@gns3vm:~$ sudo docker version
Client: Docker Engine - Community
 Version:           20.10.17
 API version:       1.41
```

```
Seleccionar root@d22c88036dc3: /

Redes3@gns3vm:~$ sudo docker version
Client: Docker Engine - Community
 Version:           20.10.17
 API version:       1.41
 Go version:        go1.17.11
 Git commit:        100c701
 Built:             Mon Jun  6 23:02:57 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.17
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.17.11
  Git commit:       a89b842
  Built:            Mon Jun  6 23:01:03 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.6
  GitCommit:        10c12954828e7c7c9b6e0ea9b0c02b01407d3ae1
 runc:
  Version:          1.1.2
  GitCommit:        v1.1.2-0-ga916309
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
Redes3@gns3vm:~$ docker search ubuntu
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/search?lim
it=25&term=ubuntu": dial unix /var/run/docker.sock: connect: permission denied
Redes3@gns3vm:~$ sudo docker search ubuntu
NAME                              DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
ubuntu                            Ubuntu is a Debian-based Linux operating sys…   14426     [OK]
websphere-liberty                 WebSphere Liberty multi-architecture images …   285       [OK]
ubuntu-upstart                    DEPRECATED, as is Upstart (find other proces…   112       [OK]
neurodebian                       NeuroDebian provides neuroscience research s…   91        [OK]
open-liberty                      Open Liberty multi-architecture images based…   53        [OK]
ubuntu/nginx                      Nginx, a high-performance reverse proxy & we…   51
ubuntu-debootstrap                DEPRECATED; use "ubuntu" instead                46        [OK]
ubuntu/mysql                      MySQL open source fast, stable, multi-thread…   34
ubuntu/apache2                    Apache, a secure & extensible open-source HT…   32
kasmweb/ubuntu-bionic-desktop     Ubuntu productivity desktop for Kasm Workspa…   28
ubuntu/prometheus                 Prometheus is a systems and service monitori…   27
ubuntu/squid                      Squid is a caching proxy for the Web. Long-t…   25
ubuntu/bind9                      BIND 9 is a very flexible, full-featured DNS…   21
ubuntu/postgres                   PostgreSQL is an open source object-relation…   17
ubuntu/redis                      Redis, an open source key-value store. Long-…   10
```

```
Seleccionar root@d22c88036dc3: /
ubuntu/prometheus                 Prometheus is a systems and service monitori…   27
ubuntu/squid                      Squid is a caching proxy for the Web. Long-t…   25
ubuntu/bind9                      BIND 9 is a very flexible, full-featured DNS…   21
ubuntu/postgres                   PostgreSQL is an open source object-relation…   17
ubuntu/redis                      Redis, an open source key-value store. Long-…   10
ubuntu/grafana                    Grafana, a feature rich metrics dashboard & …   6
ubuntu/prometheus-alertmanager    Alertmanager handles client alerts from Prom…   6
ubuntu/memcached                  Memcached, in-memory keyvalue store for smal…   5
ubuntu/kafka                      Apache Kafka, a distributed event streaming …   5
ubuntu/telegraf                   Telegraf collects, processes, aggregates & w…   4
ubuntu/zookeeper                  ZooKeeper maintains configuration informatio…   4
ubuntu/cortex                     Cortex provides storage for Prometheus. Long…   3
ubuntu/cassandra                  Cassandra, an open source NoSQL distributed …   2
bitnami/ubuntu-base-buildpack     Ubuntu base compilation image                   2                    [OK]
ubuntu/loki                       Grafana Loki, a log aggregation system like …   0
Redes3@gns3vm:~$ sudo docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
Digest: sha256:478caf1bec1afd54a58435ec681c8755883b7eb843a8630091890130b15a79af
Status: Image is up to date for ubuntu:18.04
docker.io/library/ubuntu:18.04
Redes3@gns3vm:~$ sudo docker pull ubuntu:18.04/bin/bash
invalid reference format
Redes3@gns3vm:~$ sudo docker pull ubuntu:18.04 /bin/bash
"docker pull" requires exactly 1 argument.
See 'docker pull --help'.

Usage:  docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Pull an image or a repository from a registry
Redes3@gns3vm:~$ sudo run -it ubuntu:18.04/bin/bash
sudo: run: command not found
Redes3@gns3vm:~$ sudo run -it ubuntu:18.04
sudo: run: command not found
Redes3@gns3vm:~$ sudo docker run -it ubuntu:18.04/bin/bash
docker: invalid reference format.
See 'docker run --help'.
Redes3@gns3vm:~$ sudo docker run -it ubuntu:18.04
root@bf59d1a61732:/# exit
exit
Redes3@gns3vm:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
Redes3@gns3vm:~$ sudo docker start
"docker start" requires at least 1 argument.
See 'docker start --help'.

Usage:  docker start [OPTIONS] CONTAINER [CONTAINER...]

Start one or more stopped containers
Redes3@gns3vm:~$ sudo docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
```

```
Seleccionar root@d22c88036dc3: /

Pull an image or a repository from a registry
Redes3@gns3vm:~$ sudo run -it ubuntu:18.04/bin/bash
sudo: run: command not found
Redes3@gns3vm:~$ sudo run -it ubuntu:18.04
sudo: run: command not found
Redes3@gns3vm:~$ sudo docker run -it ubuntu:18.04/bin/bash
docker: invalid reference format.
See 'docker run --help'.
Redes3@gns3vm:~$ sudo docker run -it ubuntu:18.04
root@bf59d1a61732:/# exit
exit
Redes3@gns3vm:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
Redes3@gns3vm:~$ sudo docker start
"docker start" requires at least 1 argument.
See 'docker start --help'.

Usage:  docker start [OPTIONS] CONTAINER [CONTAINER...]

Start one or more stopped containers
Redes3@gns3vm:~$ sudo docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
ubuntu              18.04     ad080923604a   8 days ago     63.1MB
gns3/openvswitch    latest    1b9a39e25bcf   4 months ago   23.6MB
gns3/ubuntu         focal     453fc1770c4f   7 months ago   124MB
hello-world         latest    feb5d9fea6a5   8 months ago   13.3kB
Redes3@gns3vm:~$ sudo docker run
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
Redes3@gns3vm:~$ sudo docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
405f018f9d1d: Pull complete
Digest: sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac
Status: Downloaded newer image for ubuntu:latest
Redes3@gns3vm:~$ sudo docker run ubuntu:18.04
Redes3@gns3vm:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
Redes3@gns3vm:~$ sudo docker run -it ubuntu:18.04/bin/bash
docker: invalid reference format.
See 'docker run --help'.
Redes3@gns3vm:~$ sudo docker run -it ubuntu bash
root@d22c88036dc3:/#
```

```
Seleccionar Redes3@gns3vm: ~/Cliente

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Wed Jun 15 01:39:12 2022 from 187.189.240.207
Redes3@gns3vm:~$ sudo docker ps
[sudo] password for Redes3:
CONTAINER ID   IMAGE     COMMAND   CREATED              STATUS              PORTS     NAMES
d22c88036dc3   ubuntu    "bash"    About a minute ago   Up About a minute             relaxed_faraday
Redes3@gns3vm:~$ sudo docker start d22c88036dc3
d22c88036dc3
Redes3@gns3vm:~$ mkdir HTTP
Redes3@gns3vm:~$ ls
HTTP
Redes3@gns3vm:~$ cd http
-bash: cd: http: No such file or directory
Redes3@gns3vm:~$ cd HTTP
Redes3@gns3vm:~/HTTP$ ls
Redes3@gns3vm:~/HTTP$ nano Dockerfile
Redes3@gns3vm:~/HTTP$ sudo docker build -t --help
invalid argument "--help" for "-t, --tag" flag: invalid reference format
See 'docker build --help'.
Redes3@gns3vm:~/HTTP$ sudo docker build --help

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

Options:
      --add-host list           Add a custom host-to-IP mapping (host:ip)
      --build-arg list          Set build-time variables
      --cache-from strings      Images to consider as cache sources
      --cgroup-parent string    Optional parent cgroup for the container
      --compress                Compress the build context using gzip
      --cpu-period int          Limit the CPU CFS (Completely Fair Scheduler) period
      --cpu-quota int           Limit the CPU CFS (Completely Fair Scheduler) quota
  -c, --cpu-shares int          CPU shares (relative weight)
      --cpuset-cpus string      CPUs in which to allow execution (0-3, 0,1)
      --cpuset-mems string      MEMs in which to allow execution (0-3, 0,1)
      --disable-content-trust   Skip image verification (default true)
  -f, --file string             Name of the Dockerfile (Default is 'PATH/Dockerfile')
      --force-rm                Always remove intermediate containers
      --iidfile string          Write the image ID to the file
      --isolation string        Container isolation technology
```

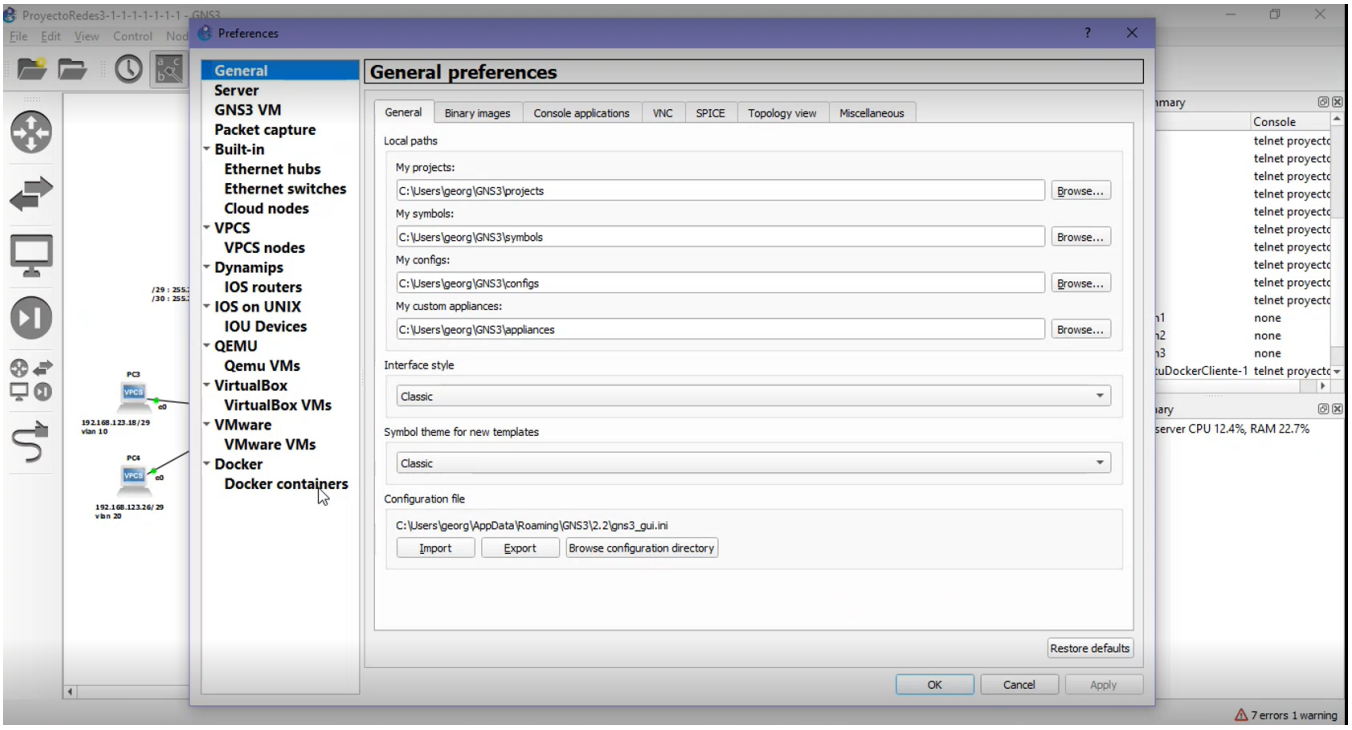
```
Seleccionar Redes3@gns3vm: ~/Cliente
Build an image from a Dockerfile
Redes3@gns3vm:~$ docker build -f HTTP .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=
%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=HTTP&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm
=1&shmsize=0&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
Redes3@gns3vm:~$ sudo docker build -f HTTP .
Sending build context to Docker daemon  14.85kB
Error response from daemon: unexpected error reading Dockerfile: read /var/lib/docker/tmp/docker-builder3989100291/HTTP: is a directory
Redes3@gns3vm:~$ sudo docker build -f HTTP/Dockerfile .
Sending build context to Docker daemon  14.85kB
Error response from daemon: dockerfile parse error line 1: unknown instruction: //
Redes3@gns3vm:~$ cd HTTP
Redes3@gns3vm:~/HTTP$ nano Dockerfile
Redes3@gns3vm:~/HTTP$ Redes3@gns3vm:~/HTTP$ cd ..
Redes3@gns3vm:~$ sudo docker build -f HTTP/Dockerfile .
Sending build context to Docker daemon  14.85kB
Error response from daemon: failed to parse HTTP/Dockerfile: ENV must have two arguments
Redes3@gns3vm:~$ nano HTTP/Dockerfile
Redes3@gns3vm:~$ sudo docker build -f HTTP/Dockerfile .
Sending build context to Docker daemon  14.85kB
Error response from daemon: dockerfile parse error line 1: unknown instruction: EROM
Redes3@gns3vm:~$ nano HTTP/Dockerfile
Redes3@gns3vm:~$ sudo docker build -f HTTP/Dockerfile .
Sending build context to Docker daemon  14.85kB
Step 1/10 : FROM ubuntu
 ---> 27941809078c
Step 2/10 : RUN apt update
 ---> Running in 9116672af992

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4653 B]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [188 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [85.2 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [199 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [148 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [209 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [4653 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [350 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [2016 B]
Fetched 21.5 MB in 3s (8462 kB/s)
Reading package lists...
```

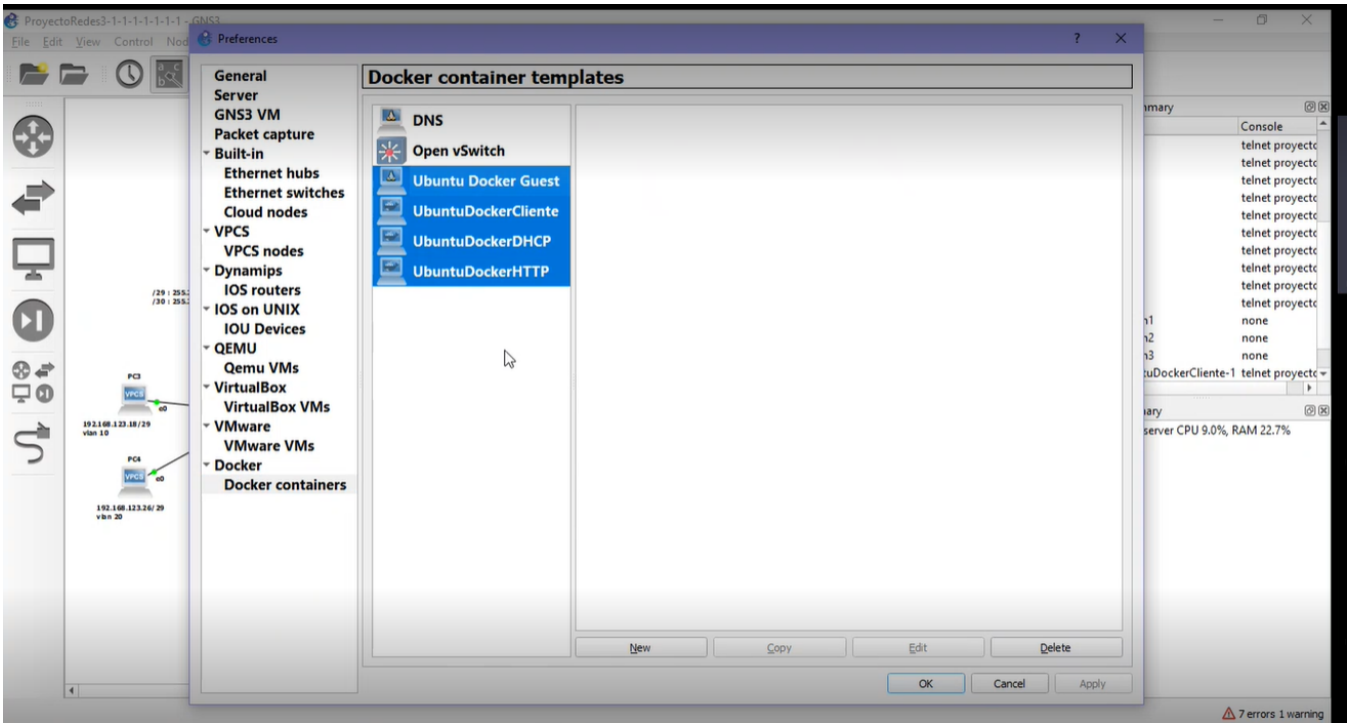
```
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
/usr/bin/deb-systemd-helper: error: systemctl preset failed on apache2.service: No such file or directory
/usr/bin/deb-systemd-helper: error: systemctl preset failed on apache-htcacheclean.service: No such file or directory
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Removing intermediate container 4efa7b527bab
 ---> 643f51d65d1c
Successfully built 643f51d65d1c
Redes3@gns3vm:~$ cd HTPP
-bash: cd: HTPP: No such file or directory
Redes3@gns3vm:~$ ls
HTTP
Redes3@gns3vm:~$ cd HTTP
Redes3@gns3vm:~/HTTP$ sudo docker build -t http:v1.0 .
Sending build context to Docker daemon  2.048kB
Step 1/10 : FROM ubuntu
 ---> 27941809078c
Step 2/10 : RUN apt update
 ---> Using cache
 ---> e528dbc9855b
Step 3/10 : ENV TZ=America/Mexico_City
 ---> Using cache
 ---> 2217fd6af522
Step 4/10 : RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
 ---> Using cache
 ---> 61e7dc47c936
Step 5/10 : RUN apt install vim git curl -y
 ---> Using cache
 ---> fb45cdf77c1b
Step 6/10 : RUN apt install iputils-ping -y
 ---> Using cache
 ---> 8db400053a07
Step 7/10 : RUN apt install iproute2 -y
 ---> Using cache
 ---> 835b0bc046ab
Step 8/10 : RUN apt install net-tools -y
 ---> Using cache
 ---> 2b86e27ffeff
Step 9/10 : RUN apt install systemctl -y
 ---> Using cache
 ---> 0374778fc1a1
```

```
Seleccionar Redes3@gns3vm: ~/Cliente
 ---> Using cache
 ---> 0374778fc1a1
Step 10/10 : RUN apt install apache2 -y
 ---> Using cache
 ---> 643f51d65d1c
Successfully built 643f51d65d1c
Successfully tagged http:v1.0
Redes3@gns3vm:~/HTTP$ cd ..
Redes3@gns3vm:~$ mkdir DHCP
Redes3@gns3vm:~$ cd DHCP
Redes3@gns3vm:~/DHCP$ nano Dockerfile
Redes3@gns3vm:~/DHCP$ sudo docker build -t dhcp:v1.0 .
[sudo] password for Redes3:
Sending build context to Docker daemon  2.048kB
Step 1/6 : FROM ubuntu
 ---> 27941809078c
Step 2/6 : RUN apt update
 ---> Using cache
 ---> e528dbc9855b
Step 3/6 : RUN apt install vim git curl -y
 ---> Running in 068bd74709de

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  ca-certificates git-man less libbrotli1 libbsd0 libcbor0.8 libcurl3-gnutls
  libcurl4 libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4
  libgdbm6 libgpm2 libldap-2.5-0 libldap-common libmd0 libmpdec3 libnghttp2-14
  libperl5.34 libpsl5 libpython3.10 libpython3.10-minimal libpython3.10-stdlib
  libreadline8 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db
  libsodium23 libsqlite3-0 libssh-4 libx11-6 libx11-data libxau6 libxcb1
  libxdmcp6 libxext6 libxmuu1 media-types netbase openssh-client openssl patch
  perl perl-modules-5.34 publicsuffix readline-common vim-common vim-runtime
  xauth xxd
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui
  gitk gitweb git-cvs git-mediawiki git-svn gdbm-l10n gpm
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
  libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql keychain
  libpam-ssh monkeysphere ssh-askpass ed diffutils-doc perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make
  libtap-harness-archive-perl readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
  ca-certificates curl git git-man less libbrotli1 libbsd0 libcbor0.8
  libcurl3-gnutls libcurl4 libedit2 liberror-perl libexpat1 libfido2-1
  libgdbm-compat4 libgdbm6 libgpm2 libldap-2.5-0 libldap-common libmd0
  libmpdec3 libnghttp2-14 libperl5.34 libpsl5 libpython3.10
```

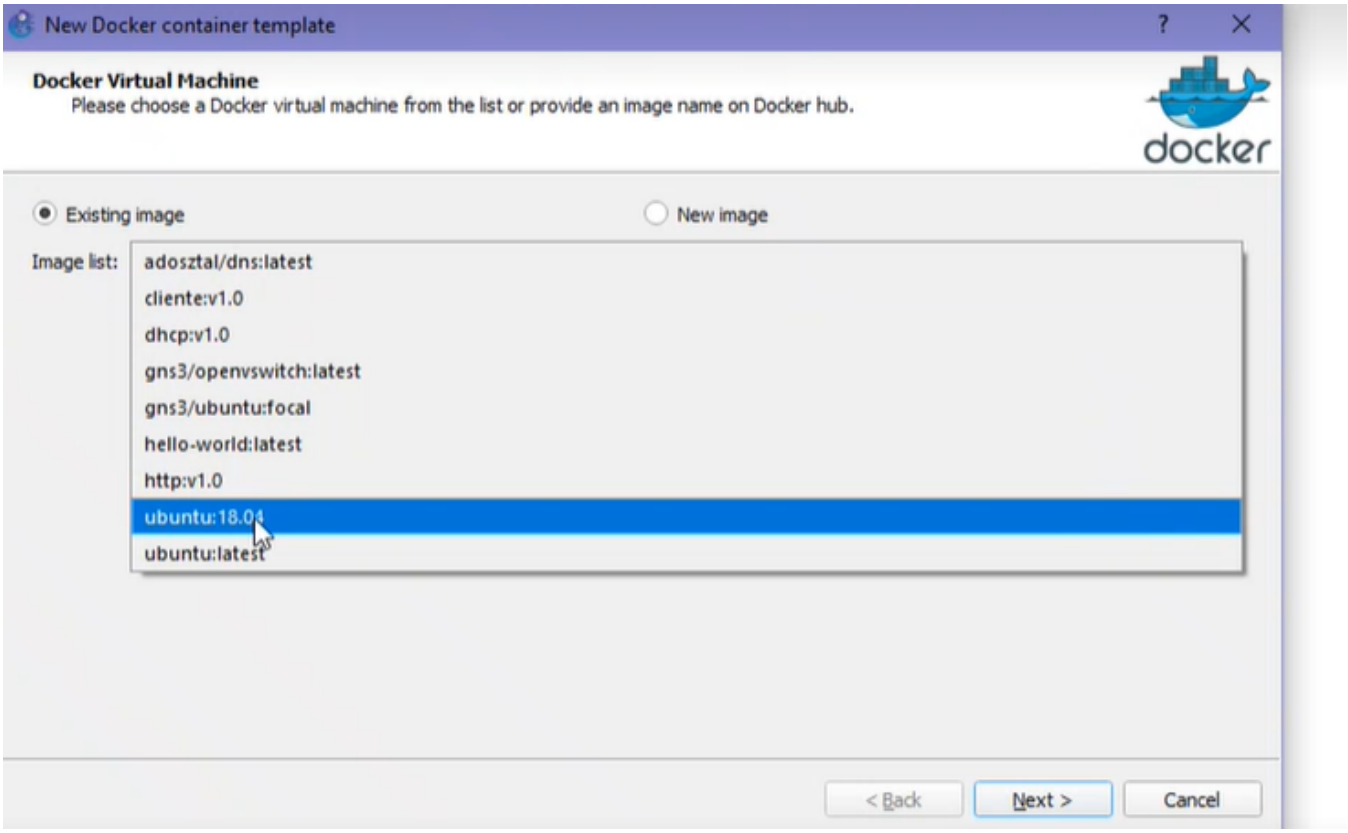
```
Seleccionar Redes3@gns3vm: ~/Cliente
DHCP  HTTP
Redes3@gns3vm:~$ mkdir Cliente
Redes3@gns3vm:~$ cd Cliente/
Redes3@gns3vm:~/Cliente$ nano Dockerfile
Redes3@gns3vm:~/Cliente$ sudo docker build -t cliente:v1.0 .
[sudo] password for Redes3:
Sorry, try again.
[sudo] password for Redes3:
Sorry, try again.
[sudo] password for Redes3:
sudo: 3 incorrect password attempts
Redes3@gns3vm:~/Cliente$ sudo docker build -t cliente:v1.0 .
[sudo] password for Redes3:
Sending build context to Docker daemon  2.048kB
Step 1/5 : FROM ubuntu
 ---> 27941809078c
Step 2/5 : RUN apt update
 ---> Using cache
 ---> e528dbc9855b
Step 3/5 : RUN apt install vim git curl -y
 ---> Using cache
 ---> 386023ae7ec7
Step 4/5 : RUN apt install iputils-ping -y
 ---> Using cache
 ---> 1606ec56b4b4
Step 5/5 : RUN apt install iproute2 -y
 ---> Using cache
 ---> ffafccbad617
Successfully built ffafccbad617
Successfully tagged cliente:v1.0
Redes3@gns3vm:~/Cliente$ nano Dockerfile
Redes3@gns3vm:~/Cliente$ sudo docker build -t cliente:v1.0 .
Sending build context to Docker daemon  2.048kB
Step 1/7 : FROM ubuntu
 ---> 27941809078c
Step 2/7 : RUN apt update
 ---> Using cache
 ---> e528dbc9855b
Step 3/7 : RUN apt install vim git curl -y
 ---> Using cache
 ---> 386023ae7ec7
Step 4/7 : RUN apt install iputils-ping -y
 ---> Using cache
 ---> 1606ec56b4b4
Step 5/7 : RUN apt install iproute2 -y
 ---> Using cache
 ---> ffafccbad617
Step 6/7 : RUN apt install net-tools -y
 ---> Running in 4f1a6a8e6ef1
```

Templates de Docker

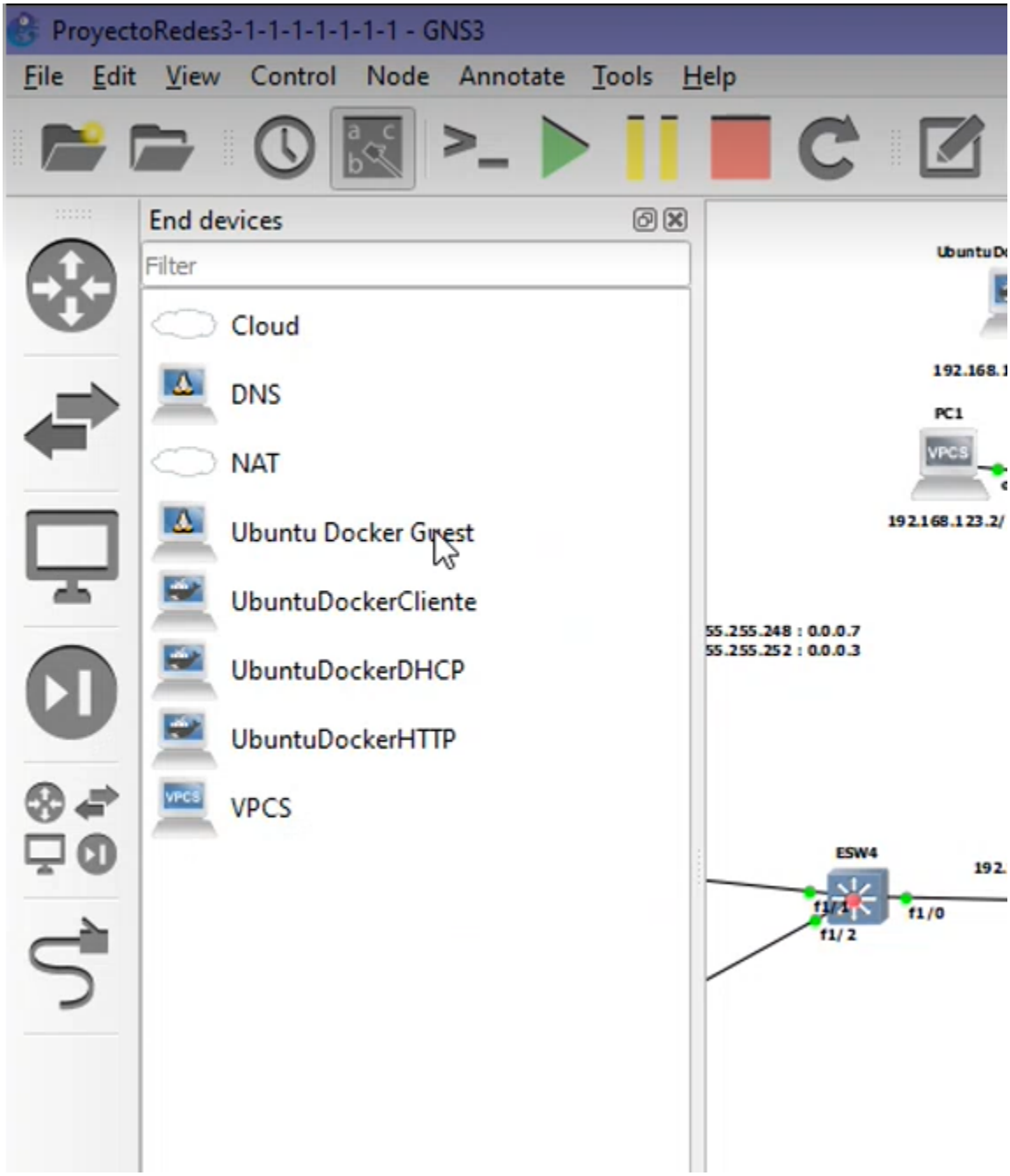




Nueva imagen de Docker

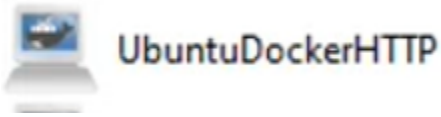


Dockers enlistados en dispositivos de salida de GNS3

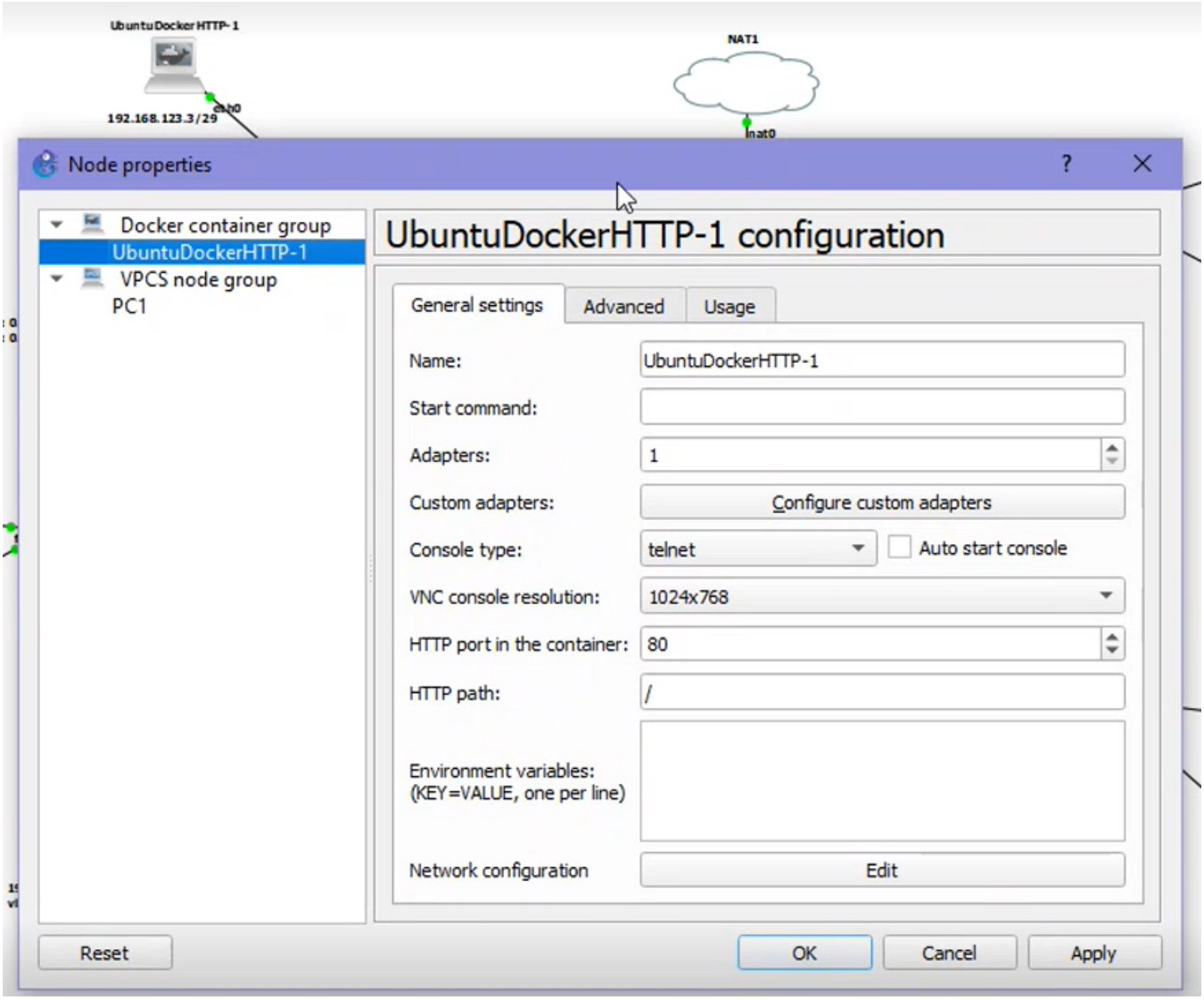


HTTP

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

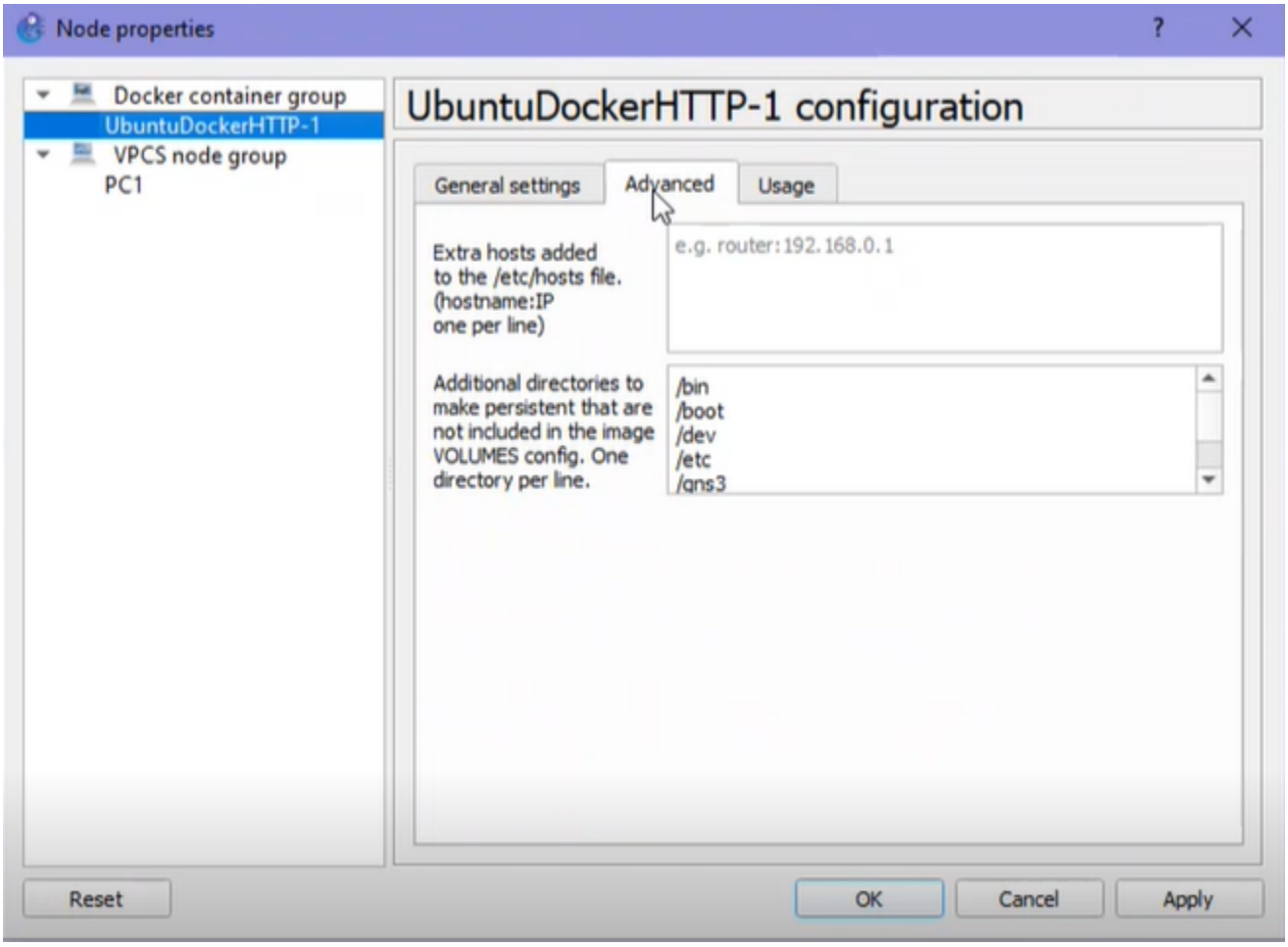


Luego instalamos la imagen en GNS3 dejando todas las opciones por defecto:

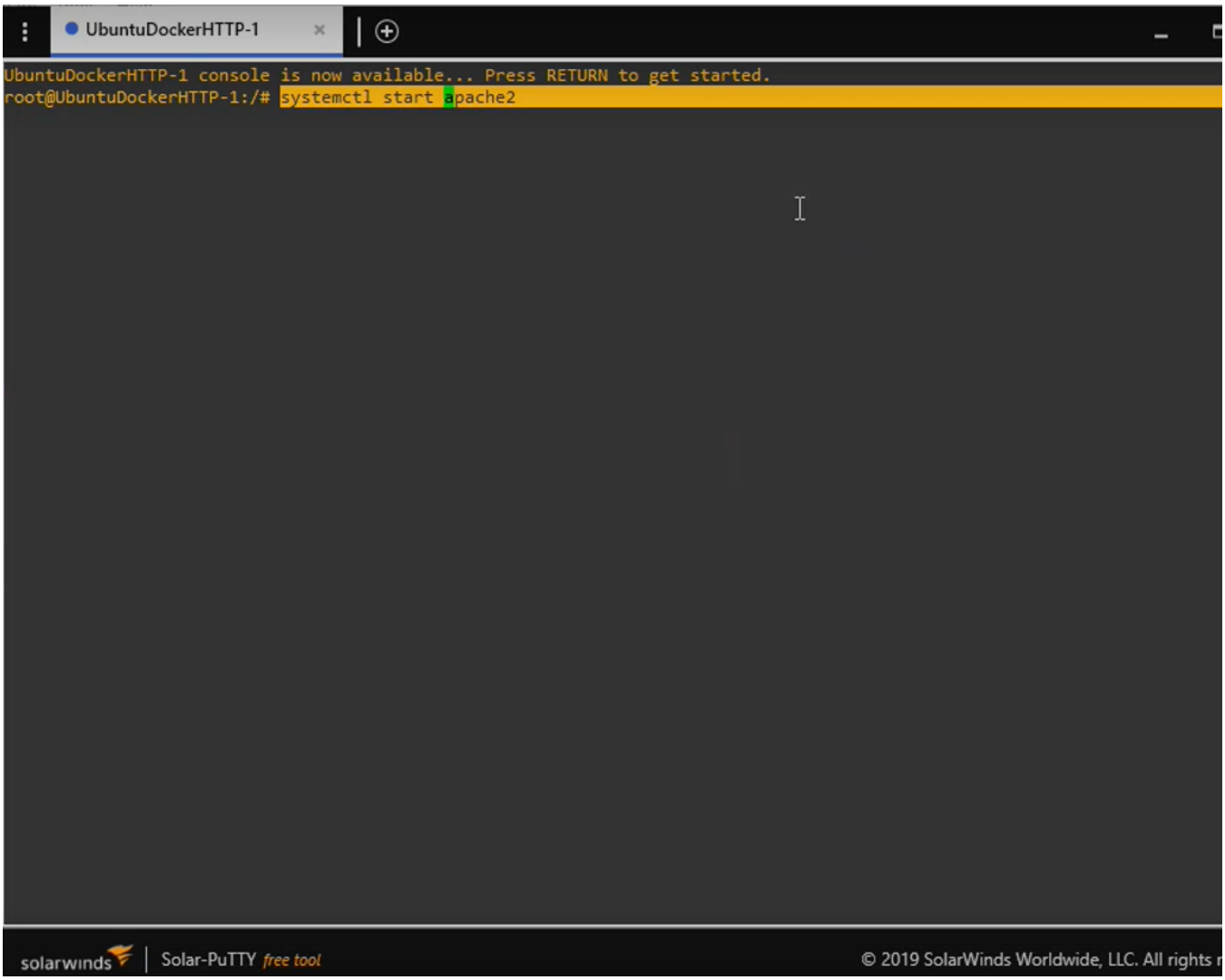


Persistencia de la información del Docker y Configuración IP

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:



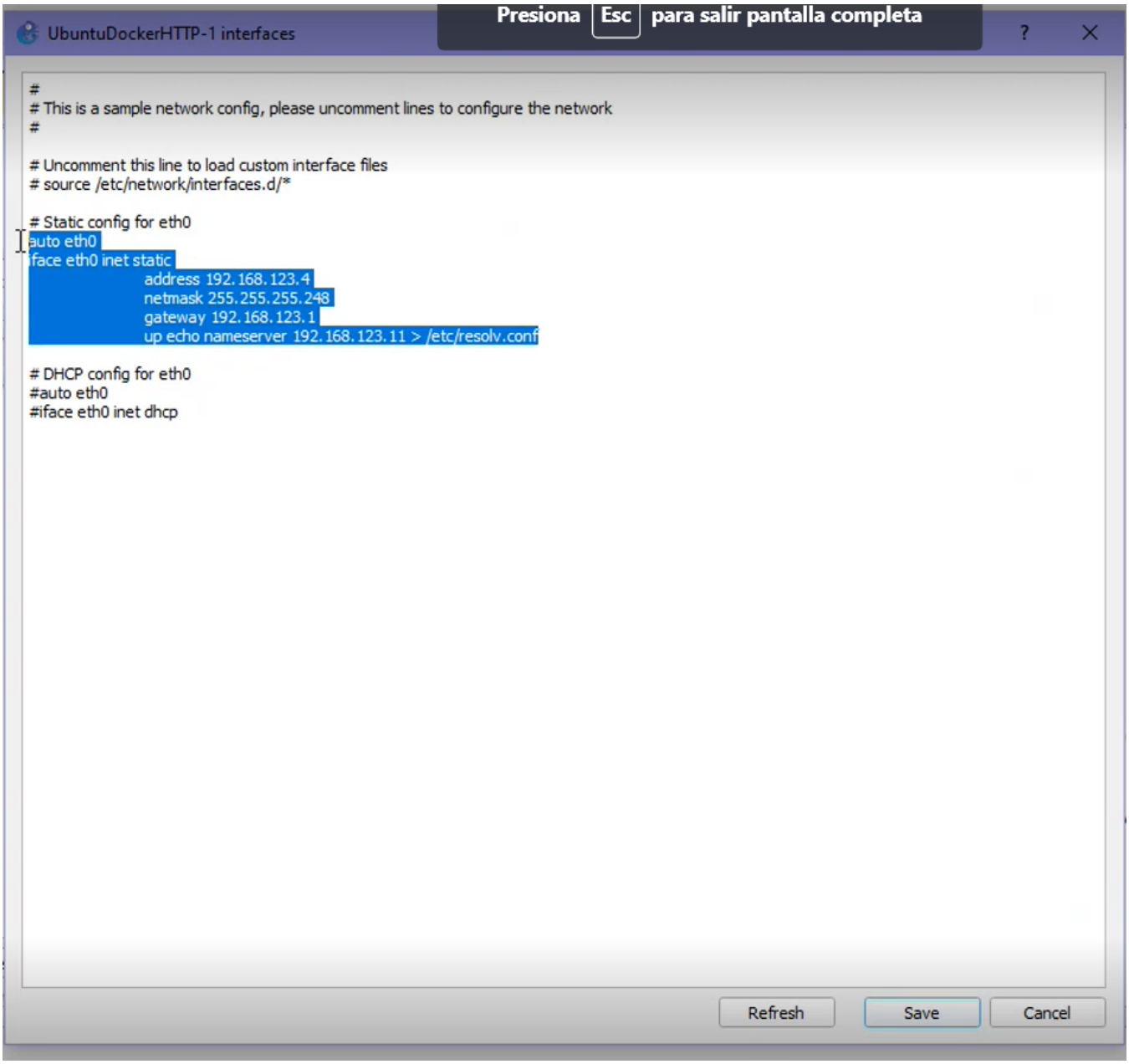
Después de realizar la configuración para persistencia de datos e ip estática se puede ingresa a la máquina e iniciar el servidor con **"systemctl start apache2"**, y probar el servicio como se muestra a continuación:


UbuntuDockerHTTP-1
UbuntuDockerHTTP-1 console is now available... Press RETURN to get started.
root@UbuntuDockerHTTP-1:/# systemctl start apache2
solarwinds
Solar-PuTTY free tool
© 2019 SolarWinds Worldwide, LLC. All rights r

Prueba servicio HTTP

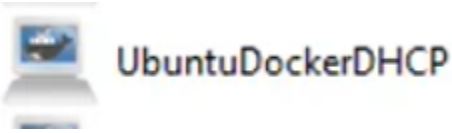
```
root@UbuntuDockerCliente:~# curl 192.168.123.4

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<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
    <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;
  }
```

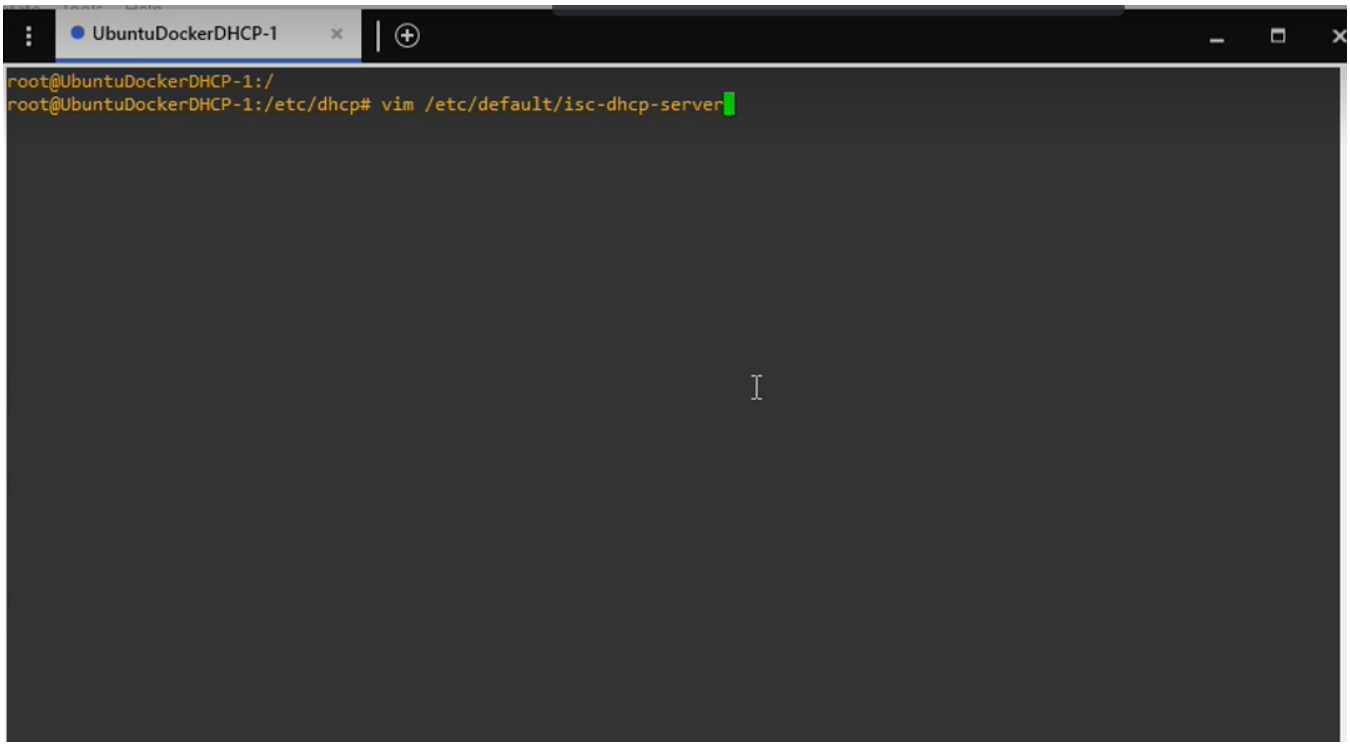
Configuración de la IP Docker



DHCP

Para el servicio de DHCP creamos nuestra imagen de Docker:





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

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:

Configuración de archivos para DHCP

UbuntuDockerDHCP-1

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

#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;
}
```

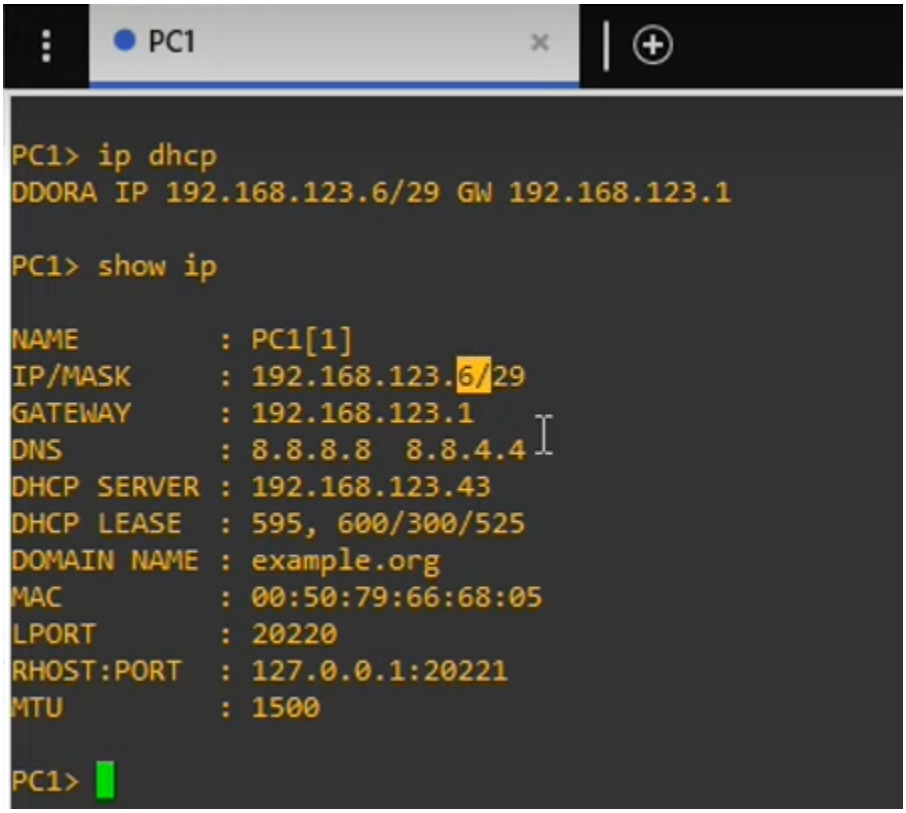
UbuntuDockerDHCP-1

```
        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 domain-name-servers 8.8.8.8, 8.8.4.4;
        option subnet-mask 255.255.255.248;
        option routers 192.168.123.1;
}
```

```
UbuntuDockerDHCP-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;
        option routers 192.168.123.1;
}

#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 2e:ef:e7:13:a3:a6;
        fixed-address 192.168.123.4;
}

#servidor DNS
host dns{
        hardware ethernet 8e:d4:d8:c0:a3:50;
        fixed-address 192.168.123.11;
}

#cliente
host cliente{
        hardware ethernet e3:86:a3:a0:86:60;
```

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:

Prueba del servicio DHCP



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:

- p domain-name TeamRedes.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

R5

```
!!!!!
Success rate is 100 percen
t (5/5), round-trip min/av
g/max = 36/47/52 ms
R5#ping google.com
Translating "google.com"..
.domain server (192.168.12
2.1) [OK]

Type escape sequence to ab
ort.
Sending 5, 100-byte ICMP E
chos to 142.251.45.14, tim
eout is 2 seconds:
!!!!!
Success rate is 100 percen
t (5/5), round-trip min/av
g/max = 48/50/56 ms
R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#ip domain-name TeamRedes.net
R5(config)#username admin privilege 15 secret admin
R5(config)#crypto key generate rsa
% You already have RSA keys defined named R5.TeamRedes.net.
% Do you really want to replace them? [yes/no]: yes
Choose the size of the key modulus in the range of 360 to 4096 for y
  General Purpose Keys. Choosing a key modulus greater than 512 may
  a few minutes.

How many bits in the modulus [512]: 2
*Jun 15 04:39:33.074: %SSH-5-DISABLED: SSH 2.0 has been disabled
2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 7 seconds)

R5(config)#
*Jun 15 04:39:46.906: %SSH-5-ENABLED: SSH 2.0 has been enabled
R5(config)#
```

solarwinds | Solar-PuTTY *free tool*

```
How many bits in the modulus [512]: 2
*Jun 15 04:39:33.074: %SSH-5-DISABLED: SSH 2.0 has been disabled
2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 7 seconds)

R5(config)#
*Jun 15 04:39:46.906: %SSH-5-ENABLED: SSH 2.0 has been enabled
R5(config)#ip ssh version 2
R5(config)#line vty 0 15
R5(config-line)#login local
R5(config-line)#transport input ssh
R5(config-line)#access-class SSH in
R5(config-line)#end
R5#wr
*Jun 15 04:40:39.334: %SYS-5-CONFIG_I: Configured from console by console
R5#wr
Building configuration...
[OK]
R5#
```

solarwinds | Solar-PuTTY *free tool*

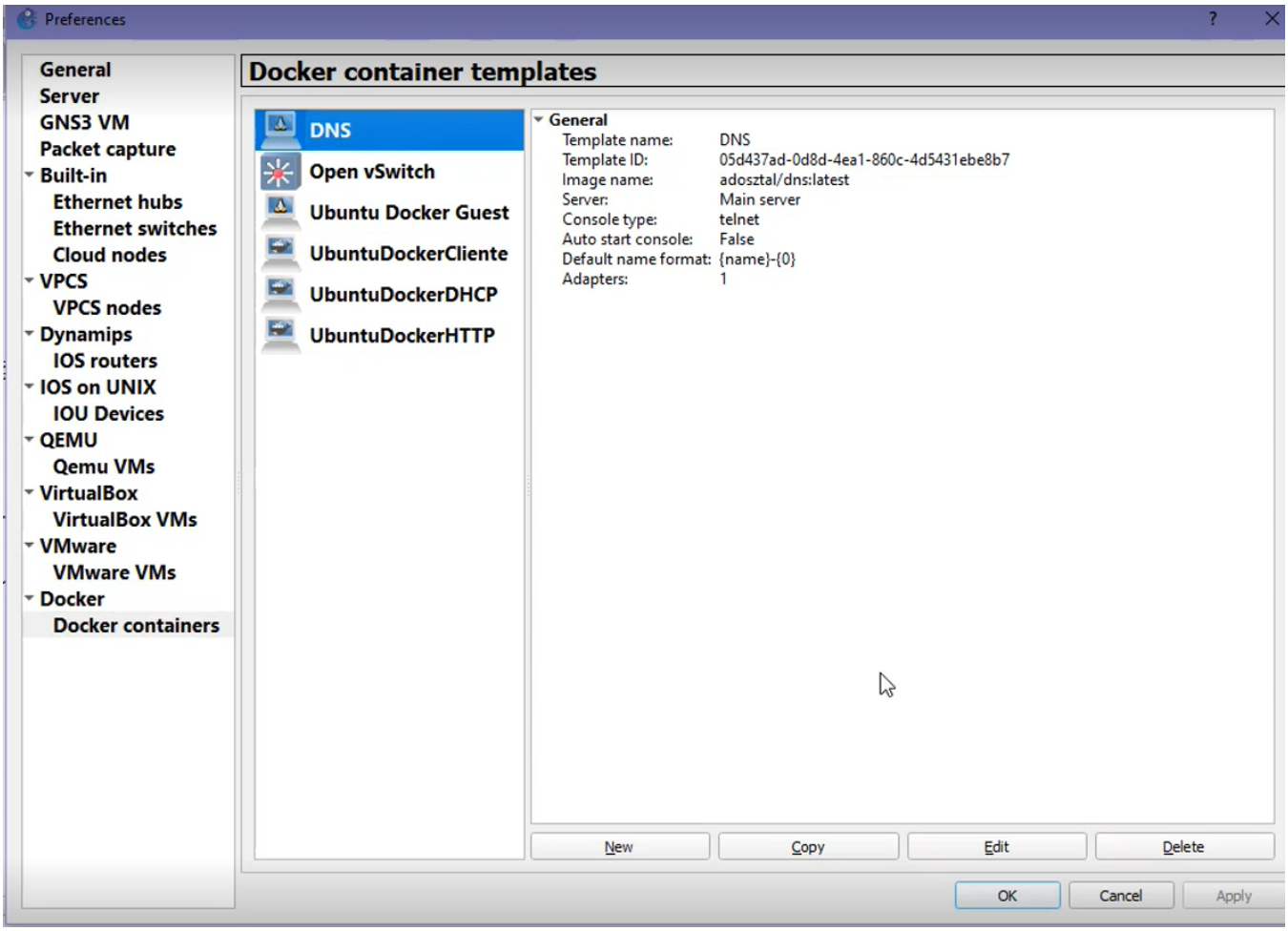
Finalmente, se puede probar SSH hacia la máquina virtual de Azure o hacia cualquiera de los routers. En la siguiente imagen se muestra la conexión **a la máquina virtual de Azure.**

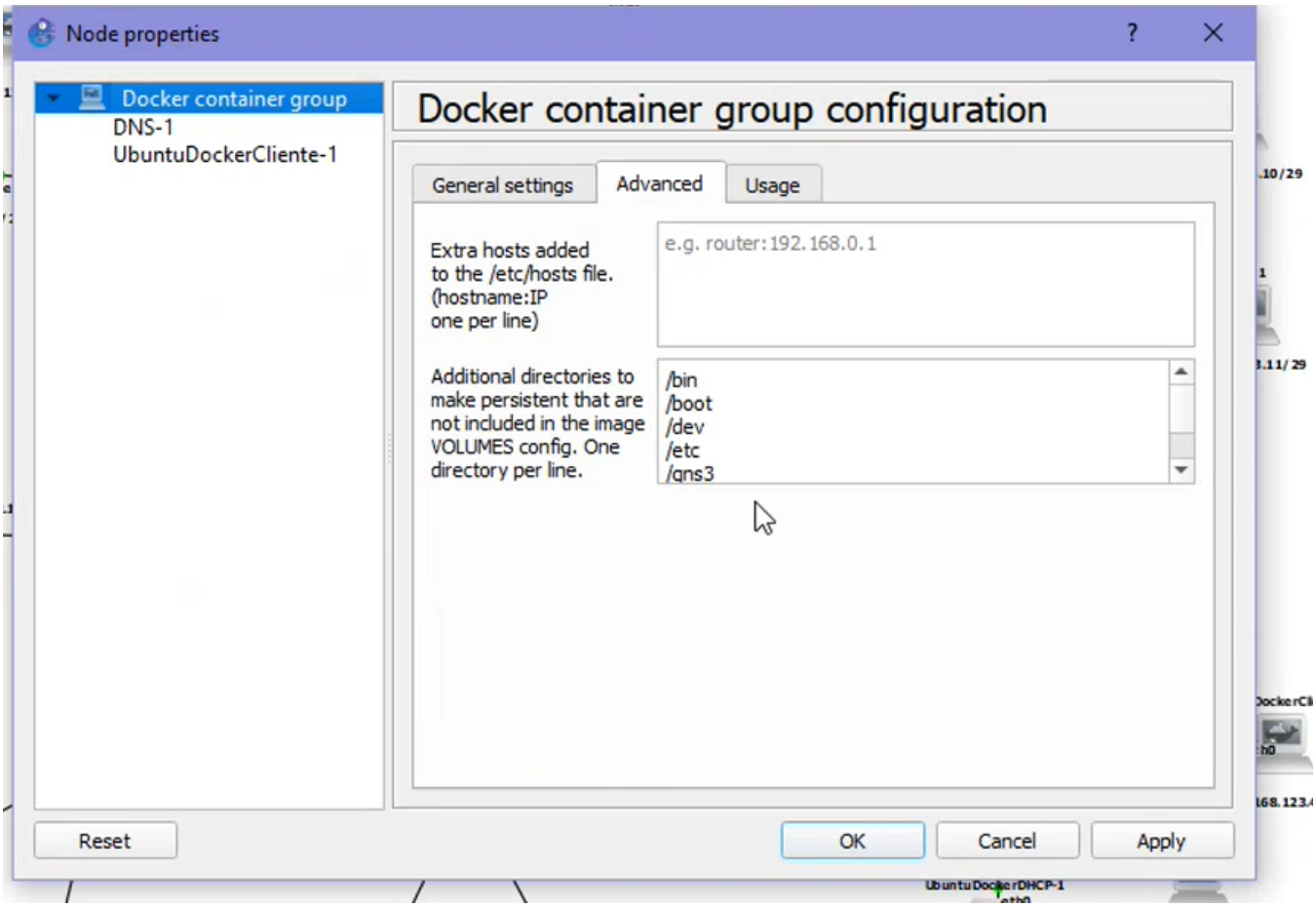
```
R5                    UbuntuDockerCliente-1
Unable to negotiate with 192.168.123.234 port 22: no matching host key type found. Their offer: ssh-rsa
root@UbuntuDockerCliente-1:/# ^C
root@UbuntuDockerCliente-1:/# ssh Redes3@20.115.22.173
The authenticity of host '20.115.22.173 (20.115.22.173)' can't be established.
ED25519 key fingerprint is SHA256:QyjlzdZ29AEGYohb0FNrlw2C3QozU+kY6x+YNtuqf3I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.115.22.173' (ED25519) to the list of known hosts.
Redes3@20.115.22.173's password:
Permission denied, please try again.
Redes3@20.115.22.173's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1029-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jun 15 07:38:40 UTC 2022

  System load:  1.05               Users logged in:          1
  Usage of /:   36.4% of 28.90GB   IPv4 address for docker0: 172.17.0.1
  Memory usage: 20%                IPv4 address for eth0:    10.4.0.4
  Swap usage:   0%                 IPv4 address for tap1:    192.168.1.254
  Processes:    250                IPv4 address for virbr0:  192.168.122.1

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Wed Jun 15 02:02:15 2022 from 187.189.240.207
Redes3@gns3vm:~$ ls
Cliente  DHCP  HTTP
Redes3@gns3vm:~$
solarwinds | Solar-PuTTY free tool        © 2019 SolarWinds Worldwide, LLC. All rights rese
```

DNS

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

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:


Preferences
General
Server
GNS3 VM
Packet capture
Built-in
Ethernet hubs
Ethernet switches
Cloud nodes
VPCS
VPCS nodes
Dynamips
IOS routers
IOS on UNIX
IOU Devices
QEMU
Qemu VMs
VirtualBox
VirtualBox VMs
VMware
VMware VMs
Docker
Docker containers
Docker container templates
DNS
Open vSwitch
Ubuntu Docker Guest
UbuntuDockerCliente
UbuntuDockerDHCP
UbuntuDockerHTTP
General
Template name: DNS
Template ID: 05d437ad-0d8d-4ea1-860c-4d5431ebe8b7
Image name: adosztal/dns:latest
Server: Main server
Console type: telnet
Auto start console: False
Default name format: {name}-{0}
Adapters: 1
New
Copy
Edit
Delete
OK
Cancel
Apply


Node properties
Docker container group
DNS-1
UbuntuDockerCliente-1
Docker container group configuration
General settings
Advanced
Usage
Extra hosts added to the /etc/hosts file. (hostname:IP one per line)
e.g. router:192.168.0.1
Additional directories to make persistent that are not included in the image VOLUMES config. One directory per line.
/bin
/boot
/dev
/etc
/qns3
Reset
OK
Cancel
Apply

DNS-1 interfaces

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

# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
auto eth0
iface eth0 inet static
        address 192.168.123.11
        netmask 255.255.255.248
        gateway 192.168.123.9
        up echo nameserver 8.8.8.8 > /etc/resolv.conf

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

Refresh Save Cancel

```
DNS-1
 * Starting DNS forwarder and DHCP server dnsmasq
root@DNS-1:~#
root@DNS-1:~#
root@DNS-1:~# nano /etc/resolv.conf
```

```
DNS-1
  GNU nano 4.8                                  /etc/resolv.conf
nameserver 8.8.8.8
```

```
DNS-1
  GNU nano 4.8                                          /etc/hosts
127.0.1.1       DNS-1
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
192.168.123.4 HTTP.lab
192.168.123.11 DNS.lab
192.168.123.43 DHCP.lab
```

```
DNS-1
GNU nano 4.8                                    /etc/dnsmasq.conf
# server=10.1.2.3@eth1

# 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
#except-interface=
# Or which to listen on by address (remember to include 127.0.0.1 if
# you use this.)
#listen-address=
# If you want dnsmasq to provide only DNS service on an interface,
# configure it as shown above, and then use the following line to
# disable DHCP and TFTP on it.
#no-dhcp-interface=

# On systems which support it, dnsmasq binds the wildcard address,
# even when it is listening on only some interfaces. It then discards
# requests that it shouldn't reply to. This has the advantage of
# working even when interfaces come and go and change address. If you
# want dnsmasq to really bind only the interfaces it is listening on,
# uncomment this option. About the only time you may need this is when
# running another nameserver on the same machine.

^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
```

Conclusión

Este proyecto presentado en el curso de administración de servicios de red pone de manifiesto todos los conocimientos adquiridos durante el curso actual, así como la adquisición de experiencia trabajando con tecnologías como Azure y GNS3, GNS3 tiene nos permitió implementar y simular la topología mostrada al inicio del documento GNS3 ayuda virtualizar topologías fácil y rápidamente sin coste

económico de los recursos. De esta manera, los estudiantes pueden desarrollar nuevas topologías, probar su rendimiento y analizar su red comportamiento de los dispositivos sin costo.

Referencias

Amaya Carrión, E. W. (2018). Redes de computadoras. Introducción a las redes, necesidad de una red, tipo y equipos de redes, topología de una red, diseño de redes, instalación y administración de redes LAN.

Habilitar SNMP Linux. (s. f.). QQDrile. Recuperado 8 de junio de 2022, de http://qqdrile.com/wp/componentes/habilitar-snmp-linux/

Oracle. (2015, 20 enero). Protocolos de enrutamiento - Configuración del sistema Oracle® Solaris 11.2 como enrutador o equilibrador de carga. oracle.com. Recuperado 8 de junio de 2022, de https://docs.oracle.com/cd/E56339_01/html/E53805/ipref-13.html#:%7E:text=Los%20protocolos%20de%20enrutamiento%20administran,pueden%20ejecutar%20protocolos%20de%20enrutamiento

Port Address Translation. (s. f.). Cisco. Recuperado 8 de junio de 2022, de https://www.cisco.com/assets/sol/sb/RV320_Emulators/RV320_Emulator_v1-1-0-09/help/Setup13.html

Rahman, S. (2020, 24 septiembre). How to monitor Linux servers with SNMP and Cacti. www.xmodulo.com. Recuperado 8 de junio de 2022, de https://www.xmodulo.com/monitor-linux-servers-snmp-cacti.html

Simic, S. (2022, 5 enero). How To Install an FTP Server On Ubuntu with vsftpd. Knowledge Base by phoenixNAP. Recuperado 8 de junio de 2022, de https://phoenixnap.com/kb/install-ftp-server-on-ubuntu-vsftpd

Sosa, C. R. (1999). Redes de computadoras. IPN.

Walton, A. (2021, 22 febrero). VLSM: Máscaras de Subred de Longitud Variable» CCNA 200–301. CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/vlsm-mascaras-subred-longitud-variable/#4_VLSM

Walton, A. (2020, 15 enero). ¿Qué es y cómo funciona la NAT? CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/nat-network-address-translation/#2_%C2%BFQue_es_NAT

Walton, A. (2020b, septiembre 3). Configuración de ACL Extendidas IPv4. CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/configurar-acl-extendidas/

Walton, A. (2018b, febrero 15). SNMP: Funcionamiento y Configuración. CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/snmp-funcionamiento-configuracion/

Walton, A. (2021b, febrero 27). ¿Qué es DNS?: Explicación sencilla. CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/que-es-dns/

Walton, A. (2018a, febrero 15). Configuración de PAT (NAT con sobrecarga). CCNA desde Cero. Recuperado 8 de junio de 2022, de https://ccnadesdecero.es/configuracion-pat-nat-sobrecarga/

https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html

https://itexamanswers.net/ip-domain-name-command-on-cisco-router-switch.html

https://geekland.eu/atajos-de-teclado-y-comandos-para-usar-vim-eficientemente/

https://www.youtube.com/watch?v=ERilHiKQkmw&ab_channel=DavidBombal

https://www.youtube.com/watch?v=6sCLGSvK3hg&ab_channel=ADOLFOMORENOHERNANDEZ

https://www.youtube.com/watch?v=59Q_g4_JNbU&ab_channel=gns3networking

https://www.youtube.com/watch?v=59Q_g4_JNbU&ab_channel=gns3networking

https://www.youtube.com/watch?v=ELK6MZCkJjQ&ab_channel=AliAkbarNaseri

https://www.youtube.com/watch?v=JhXUNRO8Vac&ab_channel=JungleLearning

https://gns3.com/marketplace/appliances/networkers-toolkit

https://docs.gns3.com/docs/using-gns3/advanced/connect-gns3-internet/

https://geekflare.com/dockerfile-tutorial/

https://docs.docker.com/engine/reference/builder/

https://unix.stackexchange.com/questions/693370/can-no-longer-ssh-after-local-os-update-reinstall-no-matching-host-key-type-fo

https://ebolges.com/como-activar-diferentes-claves-diffie-hellman-group1-sha1-ssh-en-centos-8-o-debian/

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