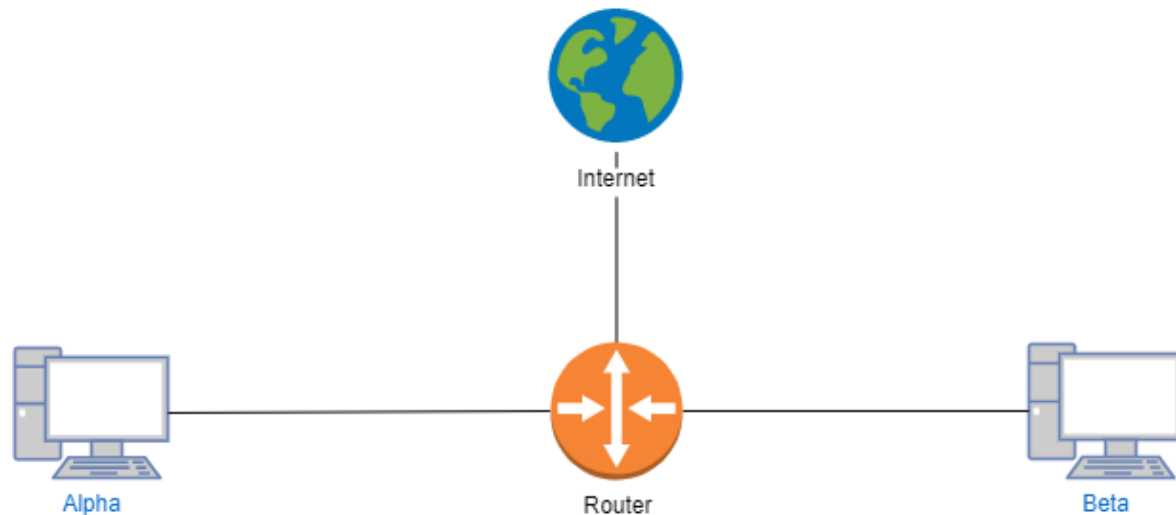


Gestión E Implementación de Redes de Computadores

Práctica 1



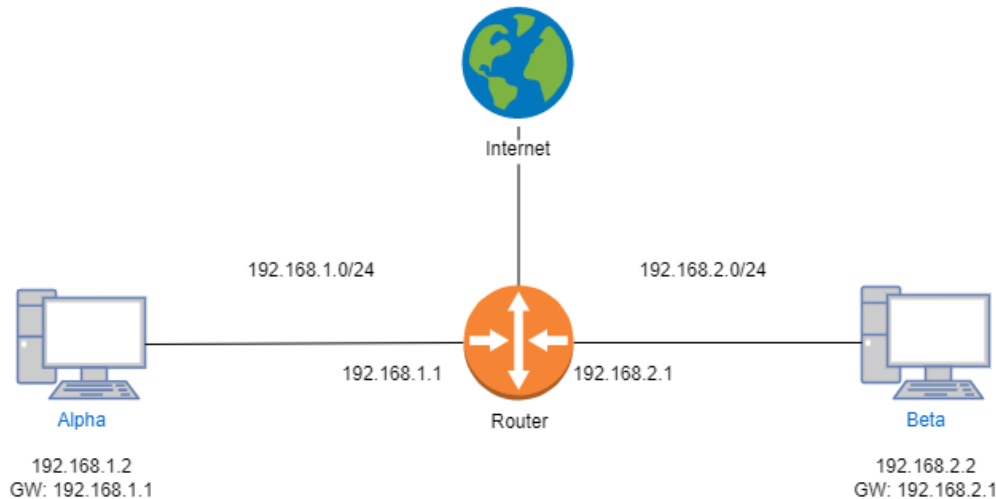
1. Creación de subredes	2
2. Instalación de máquinas virtuales	2
3. Configuración de máquinas	3
4. Parte 2 de la práctica o problemas con IPTABLES	4
5. Comandos que he utilizado en la práctica	5

[Aarón Escolano Candela](#)

1. Creación de subredes

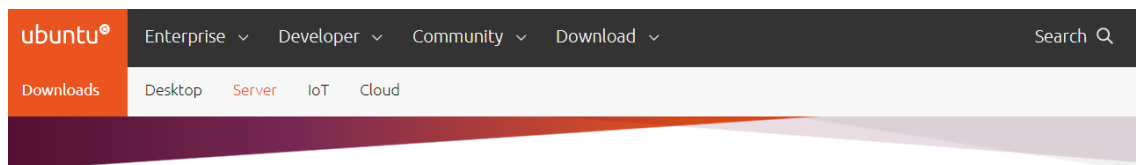
El primer paso para realizar la parte 1 de la práctica es crear las subredes y asignar las ips a cada equipo basándonos en la especificación del enunciado.

Mi solución queda así:



2. Instalación de máquinas virtuales

El siguiente paso fue decidir la distribución linux a utilizar para la práctica. Decidí utilizar ubuntu porque es la distribución más utilizada y sería con la que mayor facilidad encontraría documentación. De entre las diversas versiones de Ubuntu decidí utilizar Ubuntu Server porque es una versión sin interfaz gráfica, lo que la hace más ligera y facilitaría la ejecución de las tres máquinas virtuales simultáneamente.



Get Ubuntu Server

Option 1: Manual server installation

JSB or DVD Image based physical install

- ✓ OS security guaranteed until April 2027
- ✓ Expanded security maintenance until April 2032
- ✓ Commercial support for enterprise customers



[Download Ubuntu Server 22.04.1 LTS](#)

[Alternative downloads >](#)

[Alternative architectures >](#)

Antes de ejecutar la instalación de las máquinas virtuales me aseguro de haber configurado correctamente las interfaces de red de los tres dispositivos.

El instalador del sistema operativo nos pedirá en un paso la configuración de cada interfaz de red. Por defecto está seleccionada la obtención de ip por DHCP pero no tenemos instalado ese servicio en el router por lo que introducimos las ips de manera estática conforme al esquema del primer paso en todos los dispositivos. En el instalador también configuramos la default gateway de los dos ordenadores y la máscara de red.

3. Configuración de máquinas

Lo primero que modifiqué fueron las tablas de encaminamiento del router para permitir la comunicación entre las dos redes 192.168.2.0/24 y 192.168.1.0/24.

```
sudo ip route add 192.168.2.0/24 dev enp0s8  
sudo ip route add 192.168.1.0/24 dev enp0s3
```

Ahora que ya estaba configurada la tabla de enrutamiento probé a realizar un ping desde la máquina Alpha primero al router con:

```
ping 192.168.1.1
```

y en el router:

```
sudo tcpdump -i enp0s3
```

para leer los paquetes entrantes. Los paquetes los recibía correctamente. A continuación realicé un ping a la otra máquina situada en la red 192.168.2.0/24 para comprobar si la tabla de enrutamiento estaba configurada correctamente.

```
ping 192.168.2.2
```

y en el ordenador Beta:

```
sudo tcpdump
```

No recibía ningún paquete. Revisé todas las rutas e IP's de todos los dispositivos y todo parecía estar correctamente configurado. Decidí buscar por internet y descubrí que tenía que modificar un archivo en el router para permitir el reenvío de paquetes.

```
sudo nano /etc/sysctl.conf
```

y descomentar la línea:

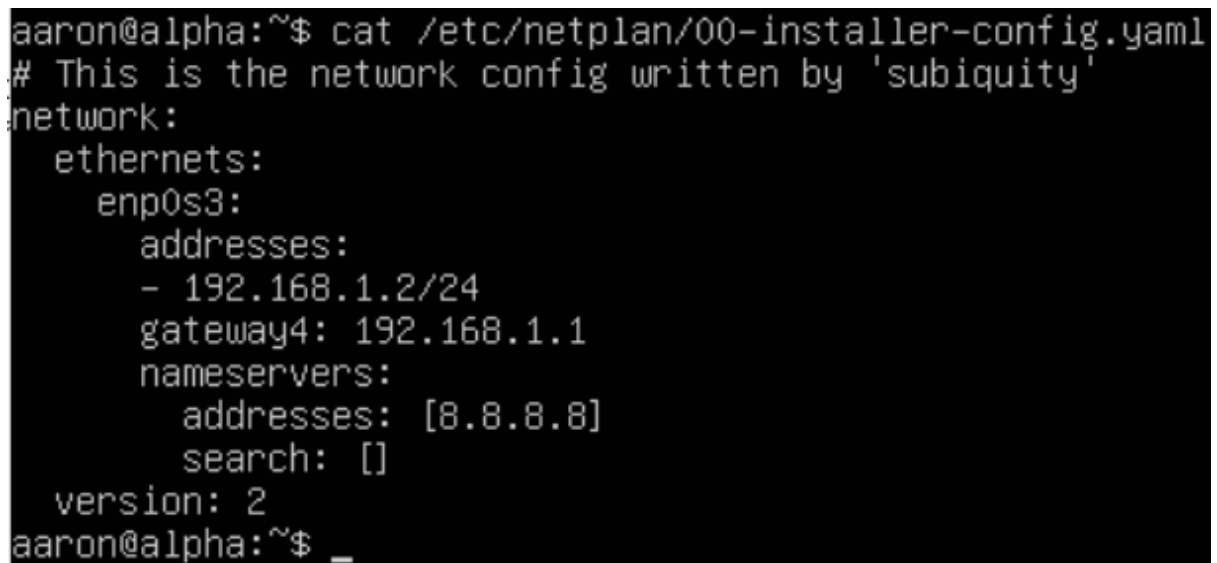
```
#net.ipv4.ip_forward=1.
```

Con eso ya funcionaban los pings.

4. Parte 2 de la práctica o problemas con IPTABLES

En esta parte de la práctica, el punto tres nos pedía permitir peticiones http. Para ello necesitaba tener el dns funcionando. Tuve que modificar el fichero 00-installer-config.yaml para añadir la ip del dns.

```
nano /etc/netplan/00-installer-config.yaml
```

A screenshot of a terminal window with a black background and white text. The text shows the command 'cat /etc/netplan/00-installer-config.yaml' being executed, followed by the contents of the file. The file is a YAML configuration for network settings, including an ethernet interface 'enp0s3' with an IP address of 192.168.1.2/24, a gateway of 192.168.1.1, and a nameserver at 8.8.8.8. The terminal prompt is 'aaron@alpha:~\$' and the cursor is on a new line after the command.

```
aaron@alpha:~$ cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
        - 192.168.1.2/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
        search: []
  version: 2
aaron@alpha:~$ _
```

```
netplan apply
```

También tuve que activar NAT con iptables:

```
iptables -t nat -A POSTROUTING -o enp0s9 -j MASQUERADE
```

Y activar el redireccionamiento de paquetes:

```
iptables -A FORWARD -i enp0s9 -o enp0s3 -m state --state
RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -i enp0s9 -o enp0s8 -m state --state
```

```
RELATED, ESTABLISHED -j ACCEPT
```

Ahora el dns ya funcionaba con el siguiente test:

```
nslookup google.com
```

Para realizar los logs he implementado la siguiente solución:

En las tablas INPUT, OUTPUT y FORWARD he añadido al final de cada tabla una regla que redirige los paquetes a una chain que he creado llamada LOGGING que se encarga de escribir los logs. Estas reglas, al estar al final de las tres tablas, me permiten identificar paquetes que no han sido aceptadas por ninguna regla establecida y me permite identificar potenciales errores.

```
Chain LOGGING (3 references)
target     prot opt source                destination
LOG        all  --  anywhere              anywhere           LOG level warning prefix "***** GIRC 2
023: PRACTICA 1 --
DROP       all  --  anywhere              anywhere
```

5. Comandos que he utilizado en la práctica

```
nc -l 80
```

crear servidor con puerto 80

```
nc -N 192.168.1.2 80
```

crear cliente

```
cat /var/log/syslog | grep "*****GIRC 2023: GET"
```

```
tail -f /var/log/syslog | grep "*****GIRC 2023: GET"
```

la -f es para leer el archivo en tiempo real

```
sudo apt install iptables-persistent
```

```
iptables-save > /etc/iptables/rules.v4
```

guardar las tablas de iptables

```
iptables-restore < /etc/iptables/rules.v4
```

```
iptables -t filter -L
```

listar la tabla filter

```
iptables -A FORWARD -p tcp -dport 80 -d 192.168.2.2 -j ACCEPT
```

Permitir los paquetes redireccionados con destino a 192.168.2.2 con protocolo tcp y puerto 80

```
iptables -P FORWARD DROP
```

Establecer política DROP en tabla FORWARD

nslookup google.com

nano /etc/sysctl.conf

 permitir redireccionamiento de paquetes

ip route add default via 192.168.1.0

tcpdump icmp

 filtrar paquetes por protocolo icmp

Herramientas

ping(icmp)

tcpdump es un sniffer como wireshark

nmap sirve para ver puertos y herramienta de hacking

 nmap -np -p0 -p 80 ip

nc (netcat) sirve para val

 nc -l 80 nos crea un servidor en el puerto 80

arp

netstat

 netstat -p

iptables