Actividad A3 – GONZALO OSCO HERNANDEZ

Crear una red ISOLATED, NAT con sus respectivas configuración además de acoplar/ desacoplar las mismas

ISOLATED

Para el ejemplo crearemos un red denominad *qa-network* bajo las siguientes configuraciones establecidas en *qa-network.xml* para la red **192.168.11.0/24**

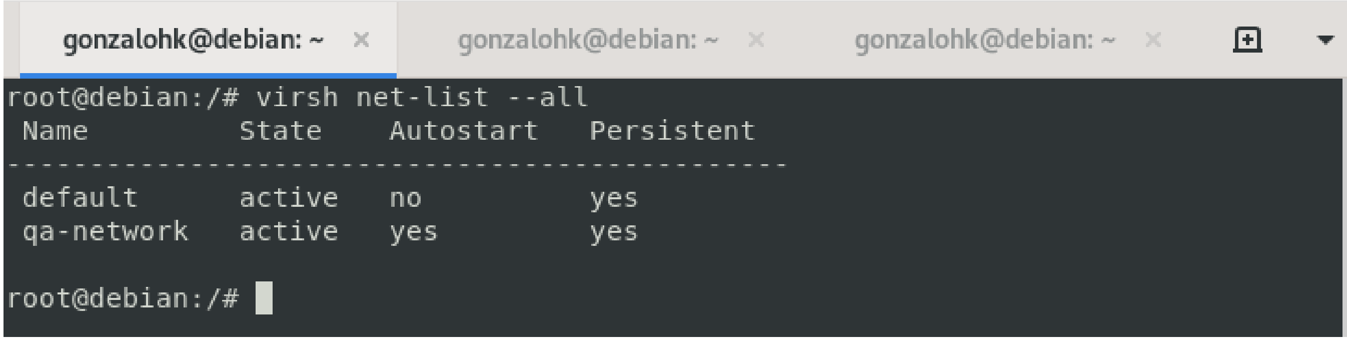
```
<network>
  <name>qa-network</name>
  <bridge name='virbr1' stp='on' delay='0'/>
  <domain name='qa-network'/>
  <ip address='192.168.11.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.11.128' end='192.168.11.254'/>
    </dhcp>
  </ip>
</network>
```
`

Ejecutamos la siguiente instrucción.

```
root@debian:/# virsh net-define qa-network.xml
```

Posterior a ello, listamos las redes para comprobar si se ha creado correctamente la red y seguidamente la iniciamos, con los siguientes comandos.

```
root@debian:/# virsh net-list --all
root@debian:/# virsh net-start qa-network
```

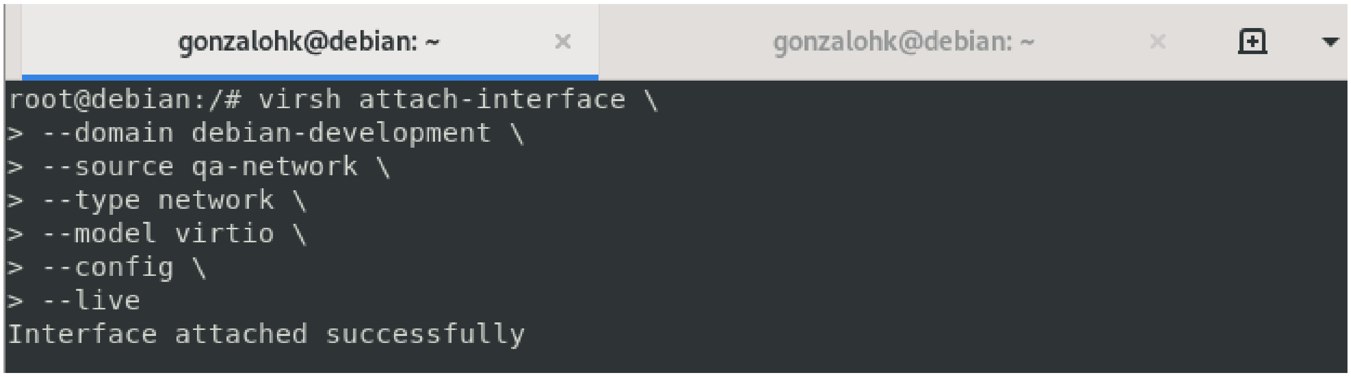


```
root@debian:/# virsh net-list --all
 Name          State    Autostart   Persistent
----------------------------------------------
 default       active   no          yes
 qa-network    active   yes         yes

root@debian:/#
```

Posterior a ello, acoplamos la nueva red a la máquina virtual denominada debían-development

```
root@debian:/# virsh attach-interface \
--domain debian-development \
--source qa-network \
--type network \
--model virtio \
--config \
--live
```



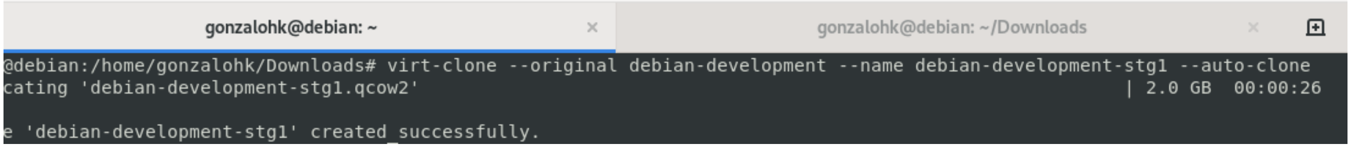
```
root@debian:/# virsh attach-interface \
> --domain debian-development \
> --source qa-network \
> --type network \
> --model virtio \
> --config \
> --live
Interface attached successfully
```

En este punto la nueva red fue acoplada de forma satisfactoria.

Para fines prácticos clonamos nuestro servidor de prueba denominado *debían-development* a uno nuevo que llamaremos *debían-development-stg1*con la siguiente instrucción.

Se debe detener la máquina virtual para realizar el proceso de clonación.

```
root@debian:/home/gonzalohk/Downloads# virt-clone --original debian-development --
name debian-development-stg1 --auto-clone
```



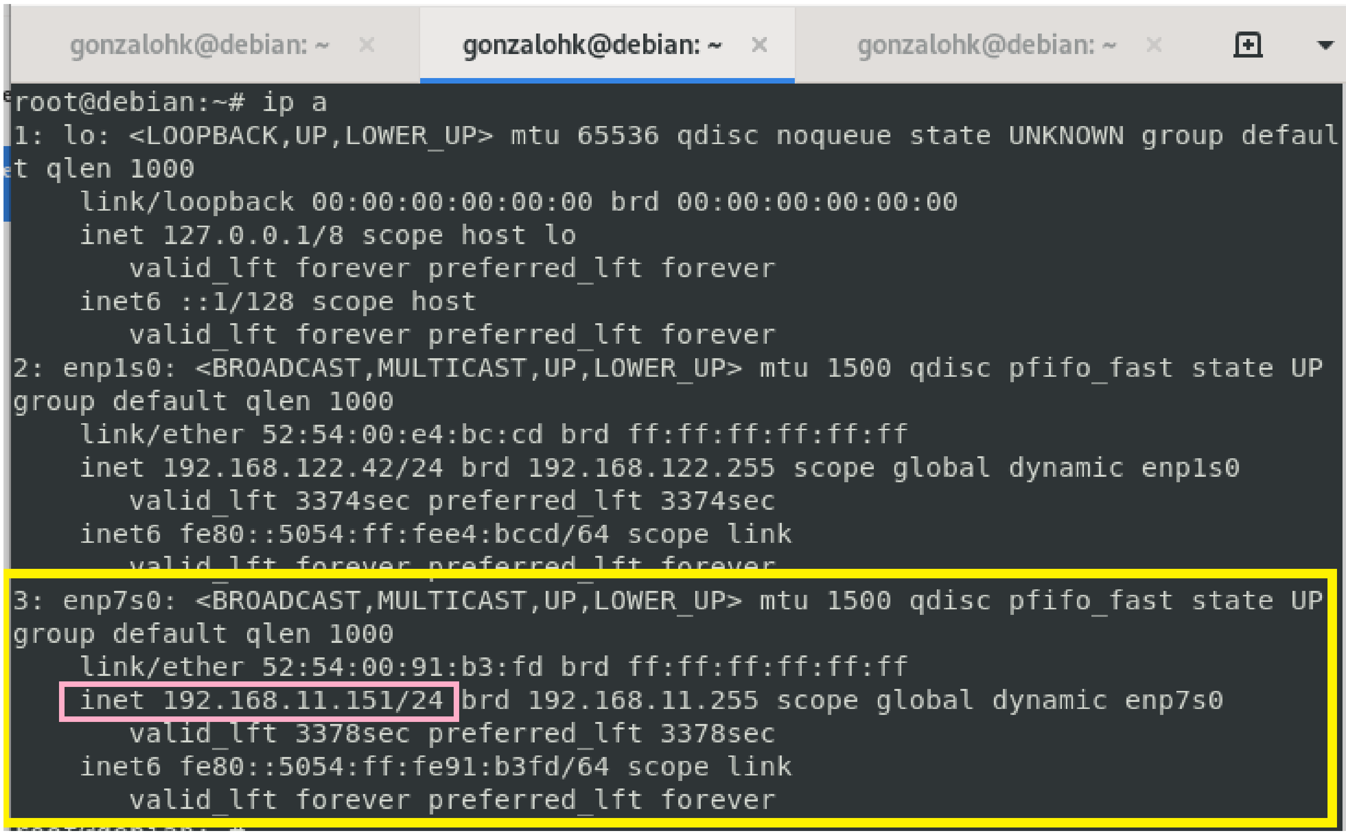
```
@debian:/home/gonzalohk/Downloads# virt-clone --original debian-development --name debian-development-stg1 --auto-clone
cating 'debian-development-stg1.qcow2'                                                   | 2.0 GB  00:00:26

e 'debian-development-stg1' created successfully.
```

Una vez terminada la clonación podemos iniciar las máquinas virtuales.

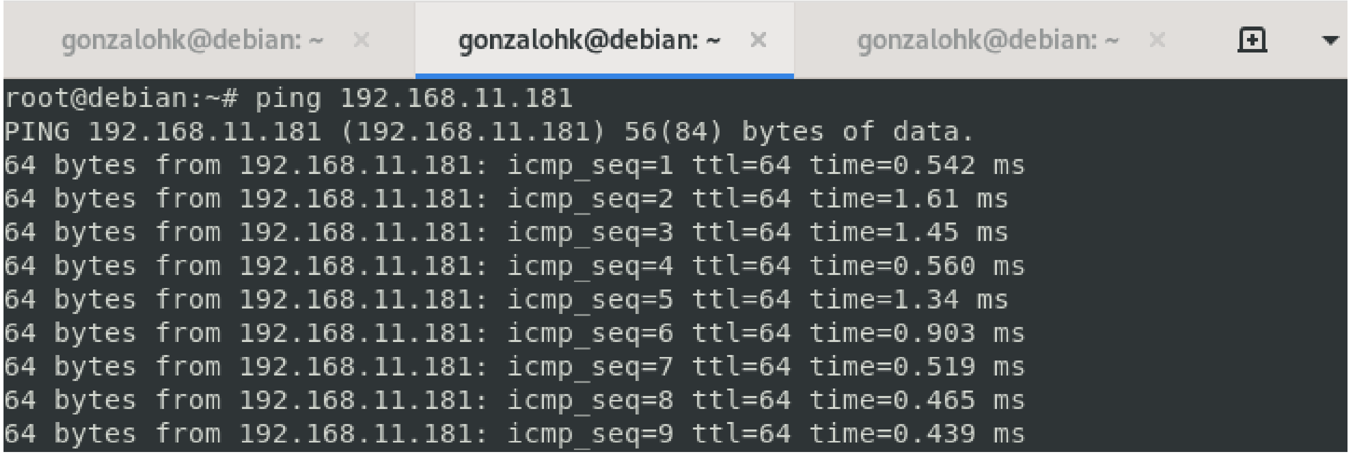
Estas ya pueden ser inicializadas y podemos verificar si las nuevas redes fueron acopladas de forma exitosa.

MV1 – debían-development



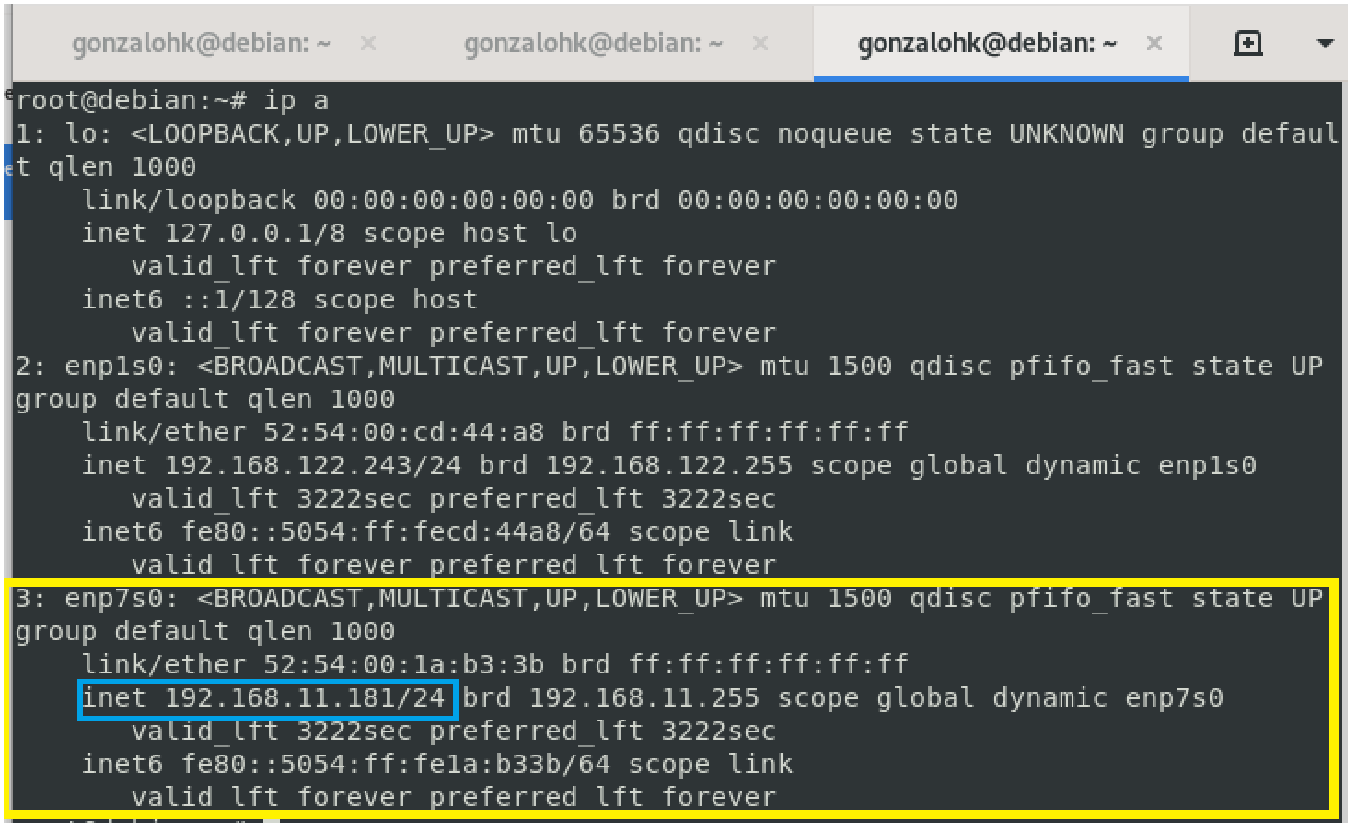
```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:e4:bc:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.42/24 brd 192.168.122.255 scope global dynamic enp1s0
       valid_lft 3374sec preferred_lft 3374sec
    inet6 fe80::5054:ff:fee4:bccd/64 scope link
       valid_lft forever preferred_lft forever
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:91:b3:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.151/24 brd 192.168.11.255 scope global dynamic enp7s0
       valid_lft 3378sec preferred_lft 3378sec
    inet6 fe80::5054:ff:fe91:b3fd/64 scope link
       valid_lft forever preferred_lft forever
```

Probando conexión a la maquina MV2 – debían-development-stg1



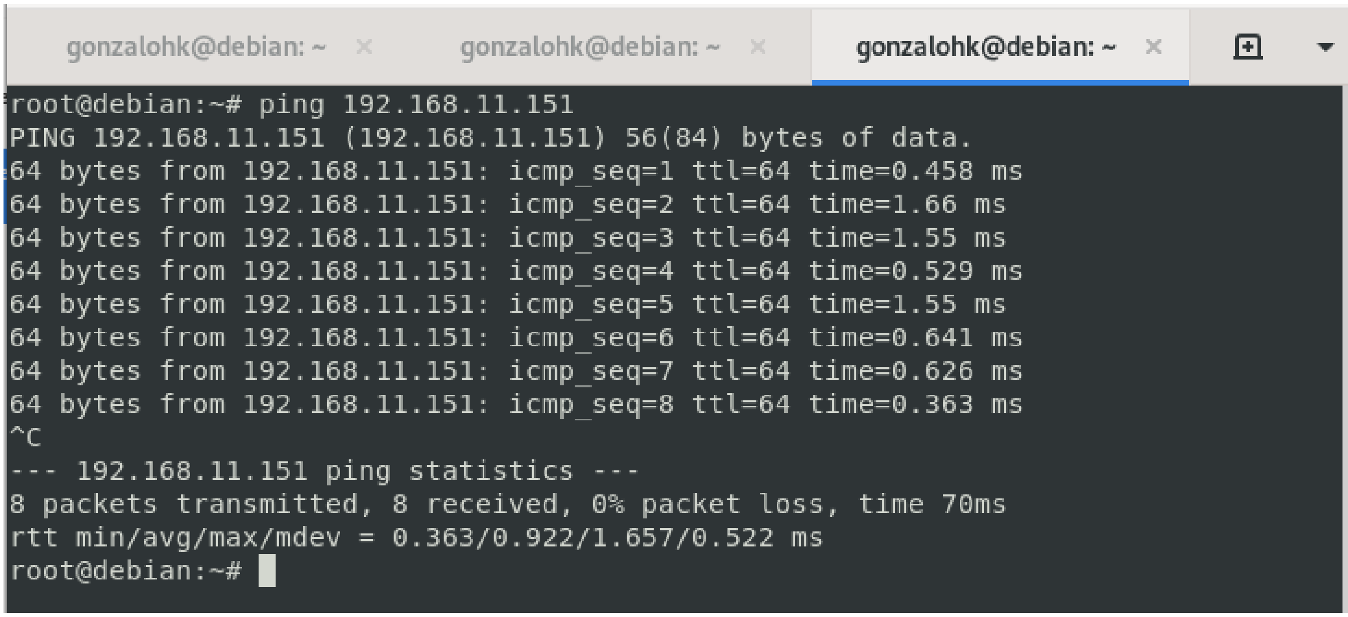
```
root@debian:~# ping 192.168.11.181
PING 192.168.11.181 (192.168.11.181) 56(84) bytes of data.
64 bytes from 192.168.11.181: icmp_seq=1 ttl=64 time=0.542 ms
64 bytes from 192.168.11.181: icmp_seq=2 ttl=64 time=1.61 ms
64 bytes from 192.168.11.181: icmp_seq=3 ttl=64 time=1.45 ms
64 bytes from 192.168.11.181: icmp_seq=4 ttl=64 time=0.560 ms
64 bytes from 192.168.11.181: icmp_seq=5 ttl=64 time=1.34 ms
64 bytes from 192.168.11.181: icmp_seq=6 ttl=64 time=0.903 ms
64 bytes from 192.168.11.181: icmp_seq=7 ttl=64 time=0.519 ms
64 bytes from 192.168.11.181: icmp_seq=8 ttl=64 time=0.465 ms
64 bytes from 192.168.11.181: icmp_seq=9 ttl=64 time=0.439 ms
```

MV2 – debían-development-stg1



```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:cd:44:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.243/24 brd 192.168.122.255 scope global dynamic enp1s0
       valid_lft 3222sec preferred_lft 3222sec
    inet6 fe80::5054:ff:fecd:44a8/64 scope link
       valid_lft forever preferred_lft forever
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:1a:b3:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.181/24 brd 192.168.11.255 scope global dynamic enp7s0
       valid_lft 3222sec preferred_lft 3222sec
    inet6 fe80::5054:ff:fe1a:b33b/64 scope link
       valid_lft forever preferred_lft forever
```

Probando conexión a la maquina MV1 – debían-development

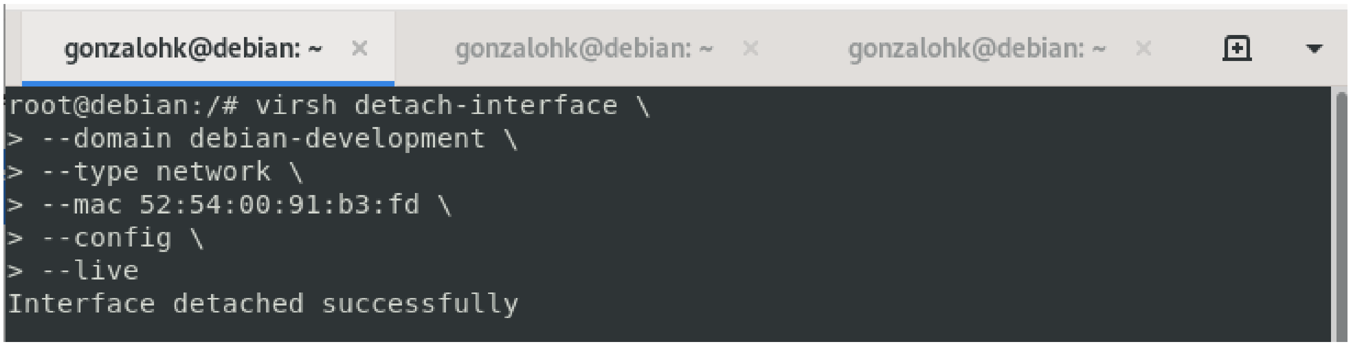


```
root@debian:~# ping 192.168.11.151
PING 192.168.11.151 (192.168.11.151) 56(84) bytes of data.
64 bytes from 192.168.11.151: icmp_seq=1 ttl=64 time=0.458 ms
64 bytes from 192.168.11.151: icmp_seq=2 ttl=64 time=1.66 ms
64 bytes from 192.168.11.151: icmp_seq=3 ttl=64 time=1.55 ms
64 bytes from 192.168.11.151: icmp_seq=4 ttl=64 time=0.529 ms
64 bytes from 192.168.11.151: icmp_seq=5 ttl=64 time=1.55 ms
64 bytes from 192.168.11.151: icmp_seq=6 ttl=64 time=0.641 ms
64 bytes from 192.168.11.151: icmp_seq=7 ttl=64 time=0.626 ms
64 bytes from 192.168.11.151: icmp_seq=8 ttl=64 time=0.363 ms
^C
--- 192.168.11.151 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 70ms
rtt min/avg/max/mdev = 0.363/0.922/1.657/0.522 ms
root@debian:~#
```

El desacople en caliente de las redes, se las puede realizar con el siguiente comando especificando la máquina virtual y la dirección Mac principalmente.

Desacople MV1 – debían-development

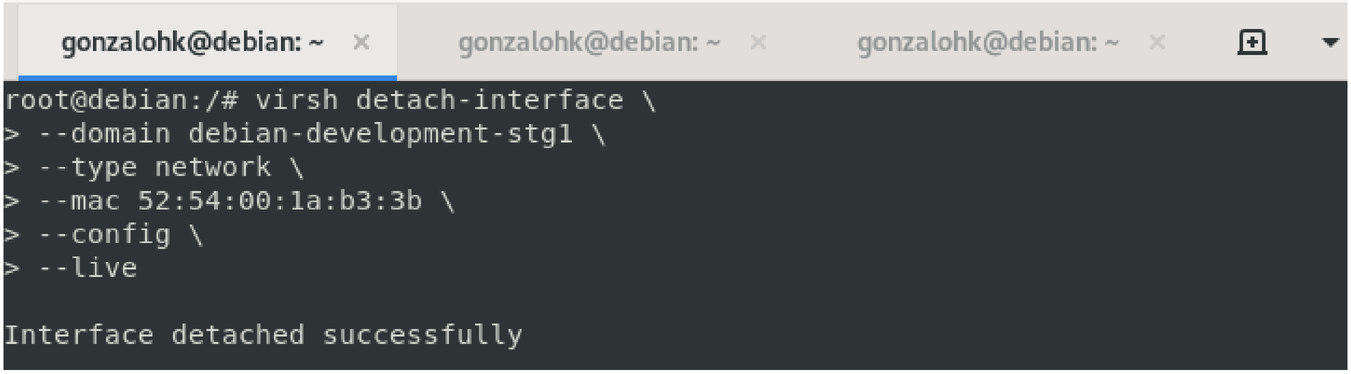
```
root@debian:/# virsh detach-interface \
> --domain debian-development \
> --type network \
> --mac 52:54:00:91:b3:fd \
> --config \
> --live
```



```
root@debian:/# virsh detach-interface \
> --domain debian-development \
> --type network \
> --mac 52:54:00:91:b3:fd \
> --config \
> --live
Interface detached successfully
```

Desacople MV2 – MV2 – debían-development-stg1

```
root@debian:/# virsh detach-interface \
> --domain debian-development-stg1 \
> --type network \
> --mac 52:54:00:1a:b3:3b \
> --config \
> --live
```



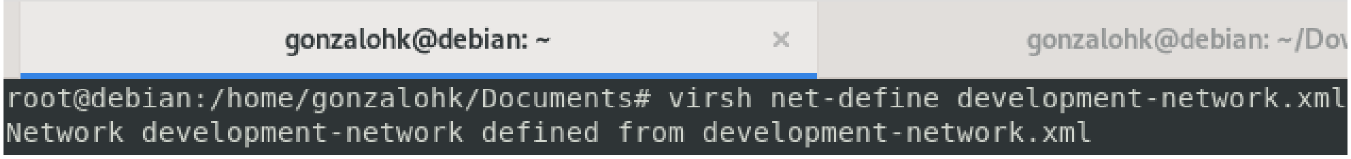
```
root@debian:/# virsh detach-interface \
> --domain debian-development-stg1 \
> --type network \
> --mac 52:54:00:1a:b3:3b \
> --config \
> --live

Interface detached successfully
```

RED NAT

Para el ejemplo crearemos un red denominad *development-network* bajo las siguientes configuraciones establecidas en *development-network.xml* para la red **192.168.66.0/24**

```
<network connections='1'>
  <name>development-network</name>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'/>
    </nat>
  </forward>
  <bridge name='virbr3' stp='on' delay='0'/>
  <ip address='192.168.66.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.66.2' end='192.168.66.254'/>
    </dhcp>
  </ip>
</network>
```



```
root@debian:/home/gonzalohk/Documents# virsh net-define development-network.xml
Network development-network defined from development-network.xml
```

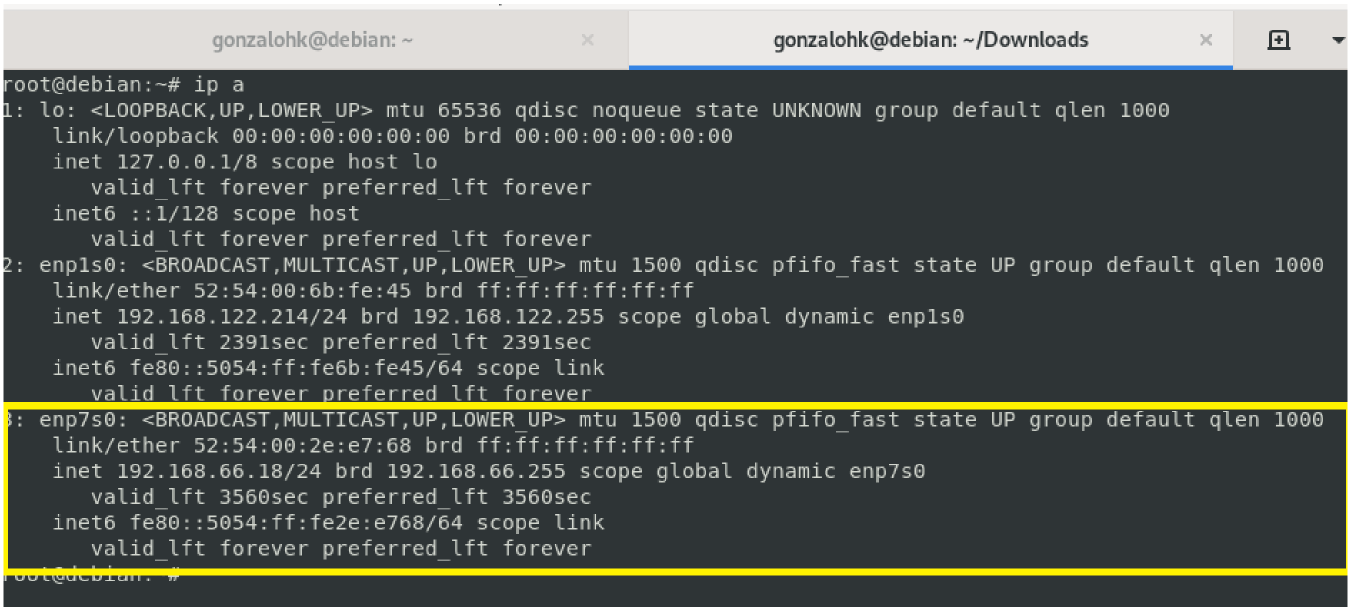
Posterior a ello, listamos las redes para comprobar si se ha creado correctamente la red y seguidamente la iniciamos, con los siguientes comandos.

```
root@debian:/# virsh net-list --all
root@debian:/# virsh net-start development-network
```

Seguidamente **acoplamos** en caliente la nueva red en la máquina virtual de nuestra preferencia con la siguiente instrucción.

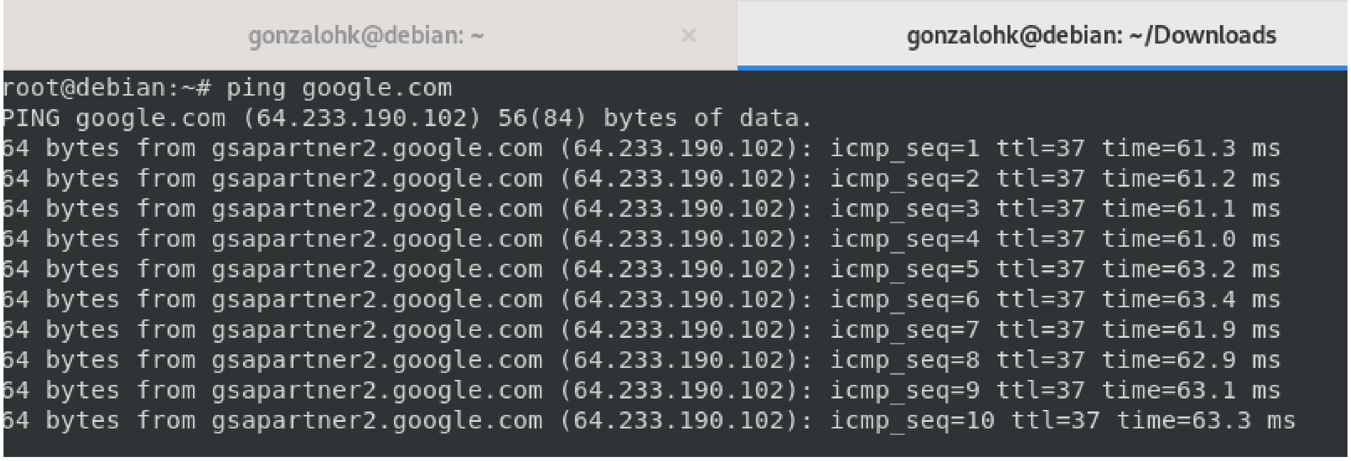
```
root@debian:/# virsh attach-interface \
> --domain debian-development \
> --source development-network \
> --type network \
> --model virtio \
> --config \
> --live
```

Para verificar, ingresamos a la máquina virtual y listamos las interfaces disponibles.



```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:6b:fe:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.214/24 brd 192.168.122.255 scope global dynamic enp1s0
       valid_lft 2391sec preferred_lft 2391sec
    inet6 fe80::5054:ff:fe6b:fe45/64 scope link
       valid_lft forever preferred_lft forever
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:2e:e7:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.18/24 brd 192.168.66.255 scope global dynamic enp7s0
       valid_lft 3560sec preferred_lft 3560sec
    inet6 fe80::5054:ff:fe2e:e768/64 scope link
       valid_lft forever preferred_lft forever
root@debian:~#
```

Finalmente, es posible hacer la verificación de nuestra red NAT haciendo ping a cualquier servicio externo.



```
root@debian:~# ping google.com
PING google.com (64.233.190.102) 56(84) bytes of data.
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=1 ttl=37 time=61.3 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=2 ttl=37 time=61.2 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=3 ttl=37 time=61.1 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=4 ttl=37 time=61.0 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=5 ttl=37 time=63.2 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=6 ttl=37 time=63.4 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=7 ttl=37 time=61.9 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=8 ttl=37 time=62.9 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=9 ttl=37 time=63.1 ms
64 bytes from gsapartner2.google.com (64.233.190.102): icmp_seq=10 ttl=37 time=63.3 ms
```

Para **desacoplar** la tarjeta en caliente, primero identificamos la MAC address con el siguiente comando.

```
root@debian:/# virsh domiflist debian-development
```

```
virsh detach-interface \
--domain debian-development \
--type network \
--mac 52:54:00:2e:e7:68 \
--config \
--live
```