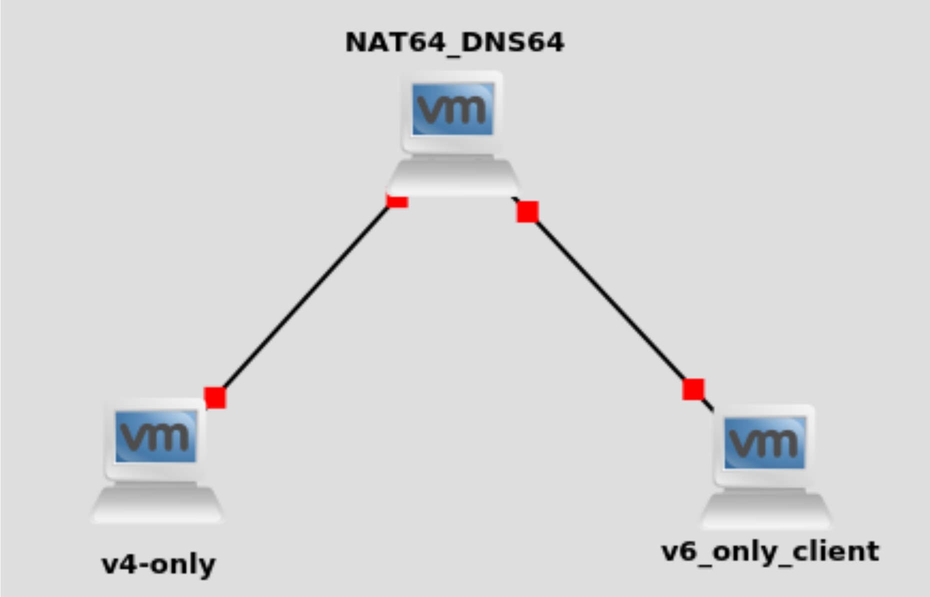
LAB-5: NAT64/DNS64

Lab Environment

Open the GNS3 project file `NAT64.gns3`

- The lab topology has:
  - 1xNAT64/DNS64 node
  - 1xIPv6-only client
  - 1xIPv4-only server (web)



- The login credentials for all nodes:

```
username: apnic
password: training
```

- Confirm interface name:
  - On the ubuntu attacker VM, check the IP configuration to see the interface name:

    ```
    ifconfig
    ```

    OR

```
ip route show | grep " src " | cut -d " " -f 3,12
```

- In this guide the interface name is `ens32` for ubuntu*1 and* `ens34` *for Ubuntu*Attack. Depending on the version of Ubuntu the interface name may be `enp0s3` or something different. Where `ens32` or `ens34` or `ens35` is used in this guide replace it with your interface name.

Configure the IPv6 only client:

1. Start the IPv6-only client node

2. Apply the following interface configs (`/etc/network/interfaces`). Use your favourite editor (`vi/nano`):

   ```
   iface ens32 inet6 static
       address 2406:6400::100
       netmask 64
       gateway 2406:6400::1
       dns-nameserver 2406:6400::1
       dns-domain apnictraining.net
   ```

   - Note that the client address could also be configured through SLAAC/DHCPv6 for stateful NAT64 instead of the static configuration as shown here.

3. Restart the network service

   ```
   sudo service networking restart
   ```

NOTE: if the service doesn't restart. Reboot the VM

1. Verify the correct address configuration on the interface

   ```
   ifconfig ens32
   ```

Configure the IPv4-only service node:

1. Start the IPv4-only service node

2. Apply the following interface configs (`/etc/network/interfaces`). Use your favourite editor (`vi/nano`):

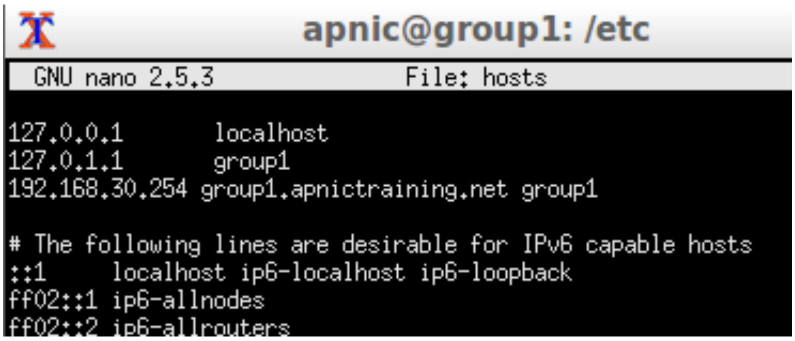
```
iface ens32 inet static
    address 192.168.30.254
    netmask 255.255.255.0
    network 192.168.30.0
    broadcast 192.168.30.255
    gateway 192.168.30.1
    dns-namesever 192.168.30.1
    dns-domain apnictraining.net
```

3. Change the hostname (`/etc/hostname`). Use your favourite editor (vi/nano):

```
group1.apnictraining.net
```

4. Set the Fully Qualified Domain Name (FQDN) (`/etc/hosts`). Use your favourite editor (vi/nano) to modify the file to look like the following:

```
sudo nano /etc/hosts
```



```
  GNU nano 2.5.3               File: hosts

127.0.0.1       localhost
127.0.1.1       group1
192.168.30.254 group1.apnictraining.net group1

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

5. Reboot the v4-only VM

```
sudo reboot
```

6. Verify the correct address configuration on the interface

```
ifconfig ens32
```

7. Note that IPv6 has been disabled on this node

```
cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

    - check the return value (`1` indicates it has been disabled; `0` otherwise)

8. The IPv4-only node has apache2 already installed on it. Verify the webserver is running:

```
service apache2 status
```

9. Have a peep at the index file (simple one)

    ```
    cat /var/www/html/index.html
    ```

10. Make sure you can access it locally from the browser. Type either of the following on your browser:

    ```
    http://localhost
    ```

    OR

    ```
    http://192.168.30.254
    ```

Configure Stateful NAT64

1. Start the NAT64_DNS64 node

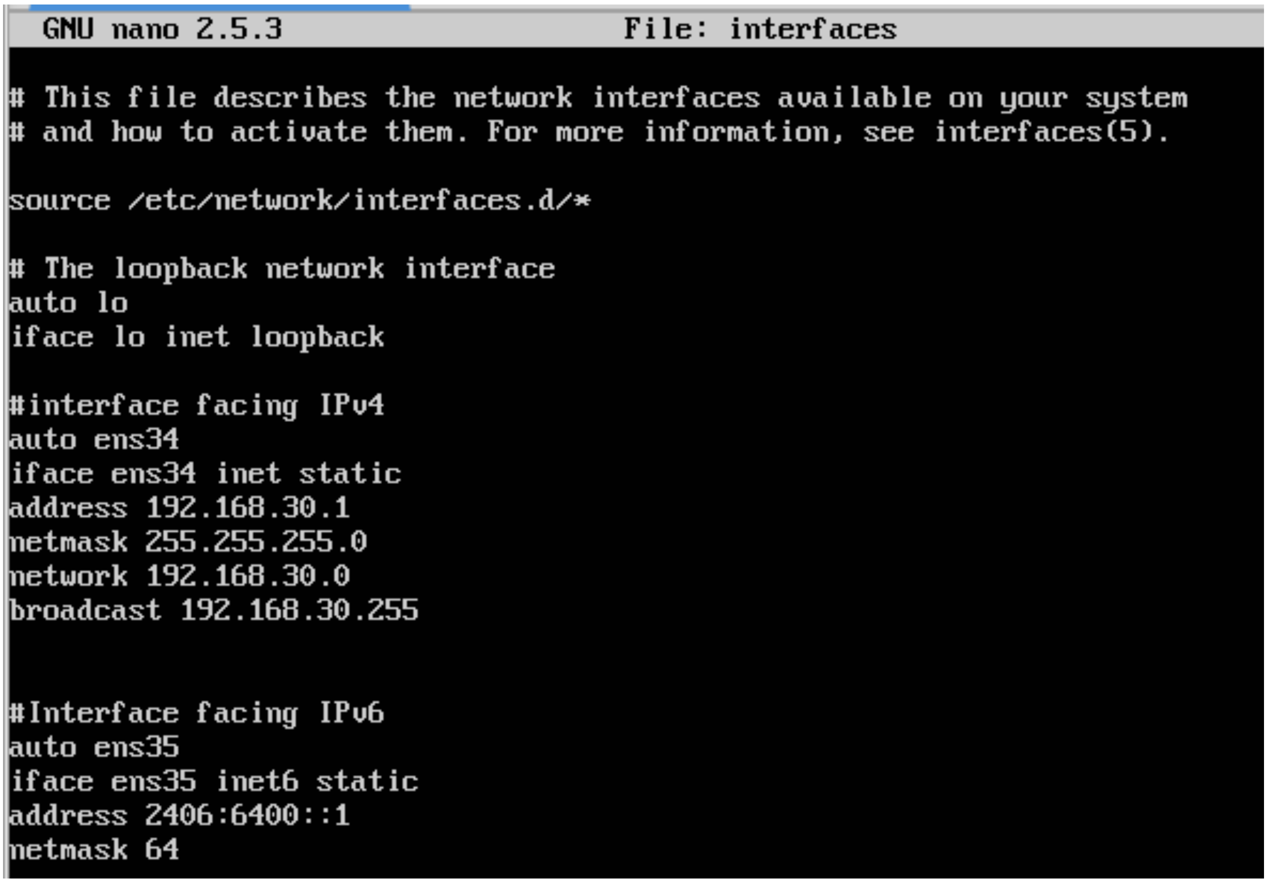
2. Configure the IPv4 facing interface (for simplicity, we will configure it to be on the same subnet as the IPv4 only node), by replacing the line `auto enp0s3` with:

    ```
    auto ens34
    iface ens34 inet static
     address 192.168.30.1
     netmask 255.255.255.0
     network 192.168.30.0
     broadcast 192.168.30.255
    ```

3. Configure the IPv6 facing interface (same link/subnet as IPv6-only node for the same reason as above), by replacing the line `auto enp0s8` with:

    ```
    auto ens35
    iface ens35 inet6 static
     address 2406:6400::1
     netmask 64
    ```

    The completed file should look like the following:



```
  GNU nano 2.5.3                    File: interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#interface facing IPv4
auto ens34
iface ens34 inet static
address 192.168.30.1
netmask 255.255.255.0
network 192.168.30.0
broadcast 192.168.30.255


#Interface facing IPv6
auto ens35
iface ens35 inet6 static
address 2406:6400::1
netmask 64
```

4. Enable IPv4 and IPv6 packet forwarding. Uncomment the following lines in `/etc/sysctl.conf`

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

5. Restart the network

```
service networking restart
```

6. Verify the interfaces have the correct IP(v4/v6) addresses configured on them

```
ifconfig
```

7. Verify basic reachability from the `NAT64_DNS64` node to the `IPv4-only` and `IPv6-only` nodes

```
ping 192.168.30.254 -c 4

ping6 2406:6400::100 -c 4
```

8. The node has `Jool` already installed on it. `Jool` is an open source implementation of Stateless and Stateful NAT64.

- Please refer to Jool Documentation for more details.
- The manual page also helps `man jool`

9. Load the `Jool` module through `modprobe`, and specify the NAT64 prefix

   ```
   sudo /sbin/modprobe jool pool6=2406:6400:64:64:64:64::/96
   ```

10. Specify the IPv4 address pool and the ports for translation

    ```
    sudo jool -4 --add 192.168.30.1 9000-10000
    ```

11. Verify the IPv4 and IPv6 translation pools

    ```
    jool -4 -d
    jool -6 -d
    ```

12. Enable NAT64 translation

    ```
    sudo jool --enable
    ```

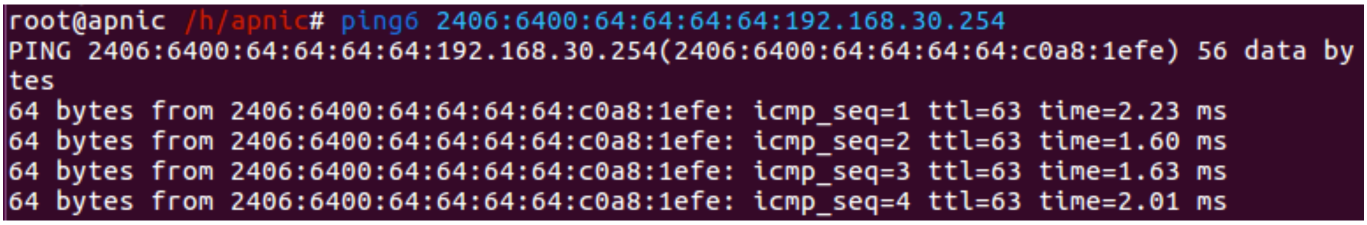
13. Check the NAT64 status

    ```
    jool -d
    ```

14. Alright, now that we have the NAT64 translaion box ready, test it by sending some requests (ping) from the `IPv6-only` node to the `IPv4-only` node:

    ```
    ping6 2406:6400:64:64:64:64:192.168.30.254 -c 4
    ```

15. The ping should succeed!

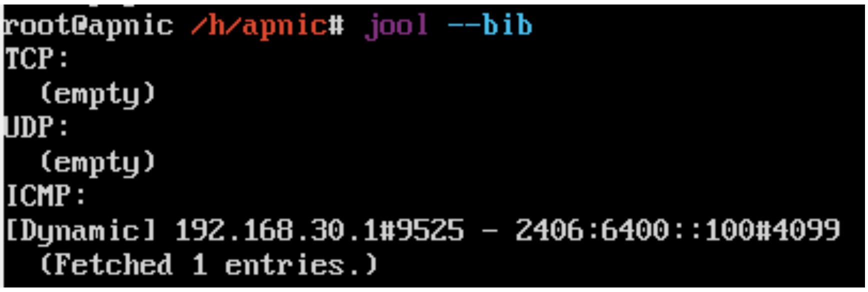


```
root@apnic /h/apnic# ping6 2406:6400:64:64:64:64:192.168.30.254
PING 2406:6400:64:64:64:64:192.168.30.254(2406:6400:64:64:64:64:c0a8:1efe) 56 data by
tes
64 bytes from 2406:6400:64:64:64:64:c0a8:1efe: icmp_seq=1 ttl=63 time=2.23 ms
64 bytes from 2406:6400:64:64:64:64:c0a8:1efe: icmp_seq=2 ttl=63 time=1.60 ms
64 bytes from 2406:6400:64:64:64:64:c0a8:1efe: icmp_seq=3 ttl=63 time=1.63 ms
64 bytes from 2406:6400:64:64:64:64:c0a8:1efe: icmp_seq=4 ttl=63 time=2.01 ms
```

16. Check the v6-to-v4 translation binding on the NAT64 node (Binding Information Base)

    ```
    sudo jool --bib
    ```

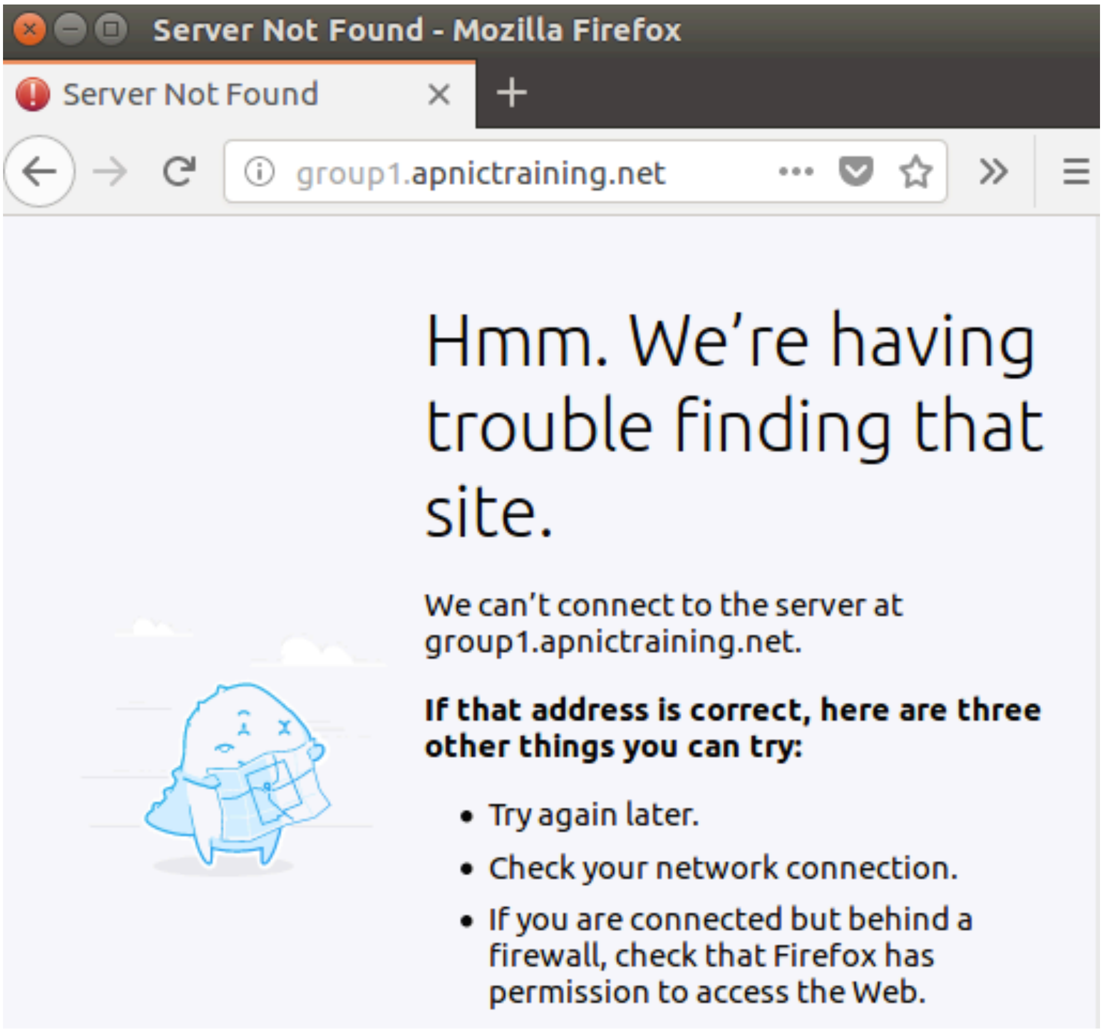
    - the translation table would look something like below, showing the incoming IPv6 address and port, and the mapped outside IPv4 address and the corresponding port:



```
root@apnic /h/apnic# jool --bib
TCP:
  (empty)
UDP:
  (empty)
ICMP:
[Dynamic] 192.168.30.1#9525 - 2406:6400::100#4099
  (Fetched 1 entries.)
```

17. Try accessing the web content on the `IPv4-only` node from the `IPv6-only` node through the browser:

    - First try with the host/domain name of the IPv4-only node (`group1.apnictraining.net`)



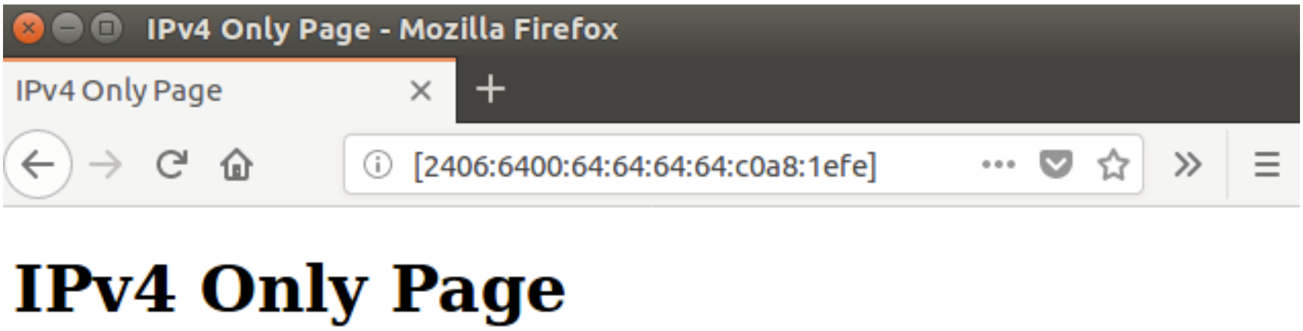
Hmm. We're having trouble finding that site.

We can't connect to the server at group1.apnictraining.net.

If that address is correct, here are three other things you can try:

- Try again later.
- Check your network connection.
- If you are connected but behind a firewall, check that Firefox has permission to access the Web.

    - Then try with the NAT64 literal for the IPv4-only node (`[2406:6400:64:64:64:64:192.168.30.254]`)


IPv4 Only Page - Mozilla Firefox
IPv4 Only Page
[2406:6400:64:64:64:64:c0a8:1efe]

IPv4 Only Page

- Try pinging `group1.apnictraining.net` from the IPv6-only node

```
ping6 group1.apnictraining.net
```

18. Instead of having to remember address literals (made worse by translated addresses), IPv6-only users would need transparency when accessing IPv4-only part of the Internet. For that, we need ***DNS64!***

Configure DNS64

1. The NAT64_DNS64 node has `bind9` preinstalled along with basic zone file(s) configured to speed up the process for you.

   - bind9.8.0 and later support DNS64 with the `dns64 options` statement
   - verify the bind9 status through rndc and systemctl

   ```
   sudo rndc status
   sudo systemctl status bind9
   ```

2. Inspect the following files under `/etc/bind` directory using the `cat` command.

   - `db.apnictrainnig.net` //the forward zone
   - `db.192.168.30` //v4 reverse zone
   - `db.2406.6400.0000` //the v6 reverse
   - `named.conf.local` //helps manage the zones associated with the domain

   - Ex: `cat db.apnictraining.net`

3. Edit the `/etc/bind/named.conf.options` file

```
sudo vi /etc/bind/named.conf.options
```

   - Make sure it is listening on IPv6

     ```
     listen-on-v6 {any;};
     ```

   - Turn off DNSSEC validation and recursion (to make it work as authoritative for now, given the topology)

     ```
     //dnssec-validation auto;
     recursion no;
     ```

   - Add the dns64 option, where we add the DNS64 prefix corresponding to the NAT64 prefix

     ```
     dns64 2406:6400:64:64:64:64::/96 {
         clients {any;};
         mapped {any;};
         exclude {0::/3; 2001:db8::/32;};
         };

         //clients: DNS64 clients (you could restrict it to certain IPv6 subnets based on your network address plan)
         //mapped: You can have ACLs to specify which IPv4 addresses are to be mapped (synthesised) into IPv6 addresses by DNS64. Ex: not synthesise A records if they are RFC1918 addresses
         //exclude: DNS64 would not synthesise AAAA records that it receives. exclude helps ignore such AAAA records and synthesise them instead using the DNS64 prefix.
     ```

   - Note that, if a DNS64 server is also authoritative for certain zones (like in our case), it will apply DNS64 to those zones too by default! Meaning, it will synthesise AAAA records from A records in the zones for which it is authoritative.

   - Also note that by default, DNS64 does not process secure queries/responses (`DO = 1`). We can override this with `break-dnssec yes;`

4. Reload bind9 to allow the configuration changes

   ```
   sudo service bind9 restart
   ```

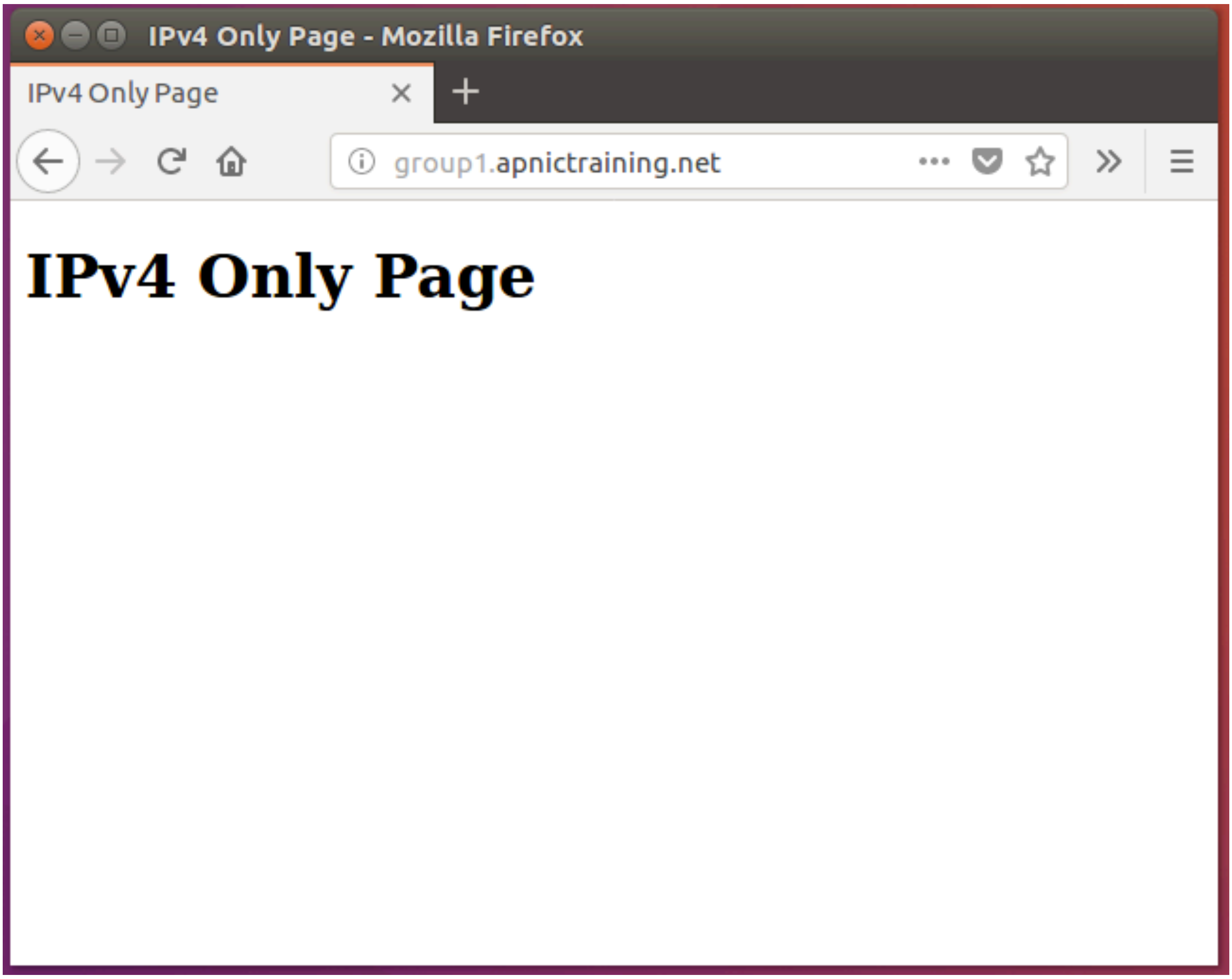
5. Verify they bind9 status

   ```
   sudo systemctl status bind9
   ```

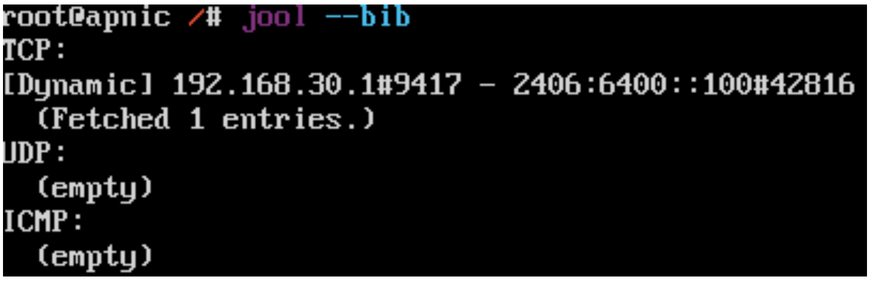
6. Now try pinging the `IPv4-only` node using its hostname (`group1.apnictraining.net`) from the `IPv6-only` node

```
ping6 group1.apnictraining.net
```

7. Try accessing the web page on the `IPv4-only` node (`group1.apnictraining.net`) from the `IPv6-only` node

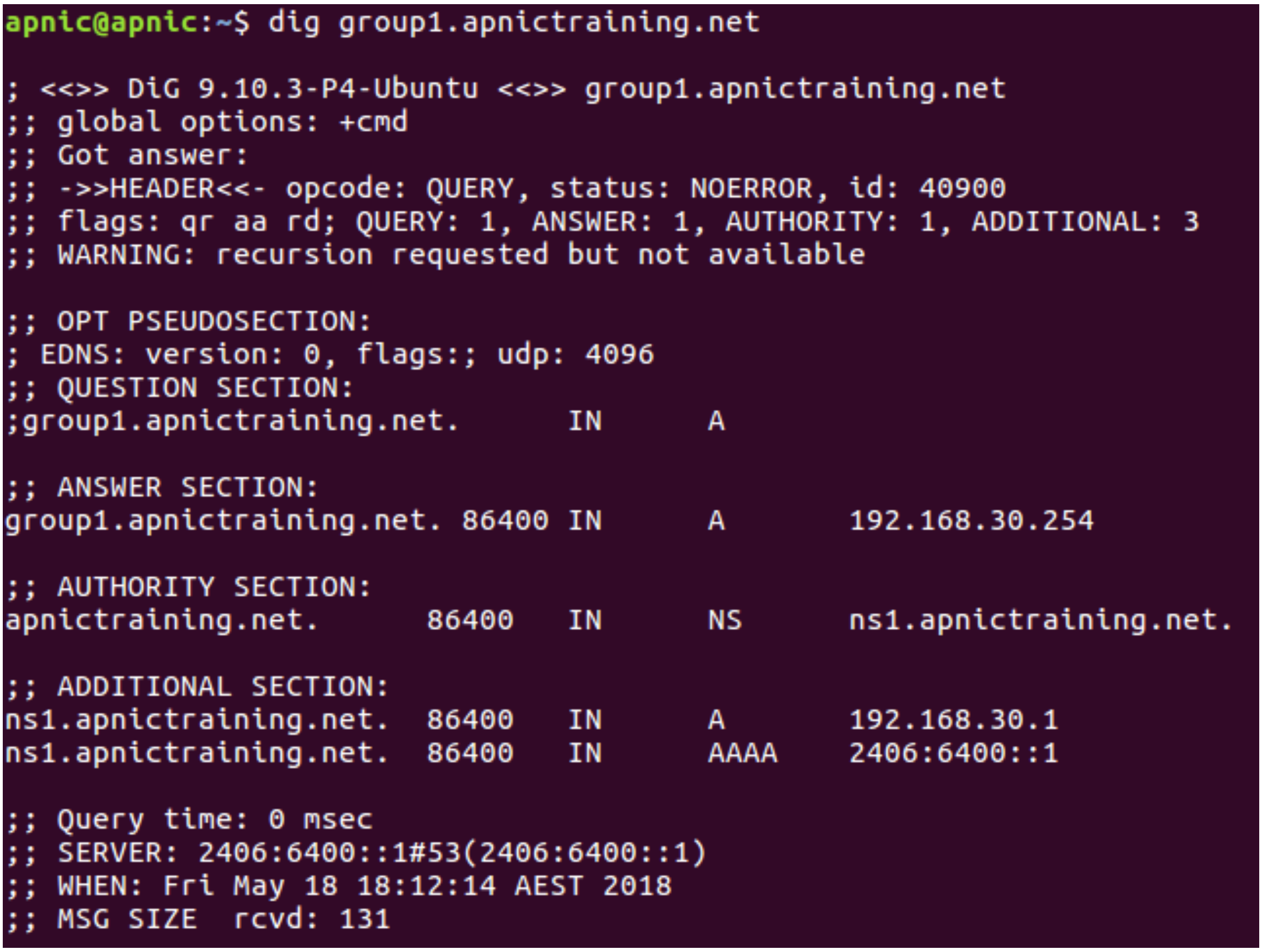


8. Verify the binding/mapping information on the NAT64_DNS64 box



```
root@apnic /# jool --bib
TCP:
[Dynamic] 192.168.30.1#9417 - 2406:6400::100#42816
  (Fetched 1 entries.)
UDP:
  (empty)
ICMP:
  (empty)
```

9. Perform a DNS lookup for the `IPv4-only` hostname from the `IPv6-only` client



```
apnic@apnic:~$ dig group1.apnictraining.net

; <<>> DiG 9.10.3-P4-Ubuntu <<>> group1.apnictraining.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40900
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;group1.apnictraining.net.      IN      A

;; ANSWER SECTION:
group1.apnictraining.net. 86400 IN      A       192.168.30.254

;; AUTHORITY SECTION:
apnictraining.net.      86400   IN      NS      ns1.apnictraining.net.

;; ADDITIONAL SECTION:
ns1.apnictraining.net.  86400   IN      A       192.168.30.1
ns1.apnictraining.net.  86400   IN      AAAA    2406:6400::1

;; Query time: 0 msec
;; SERVER: 2406:6400::1#53(2406:6400::1)
;; WHEN: Fri May 18 18:12:14 AEST 2018
;; MSG SIZE  rcvd: 131
```