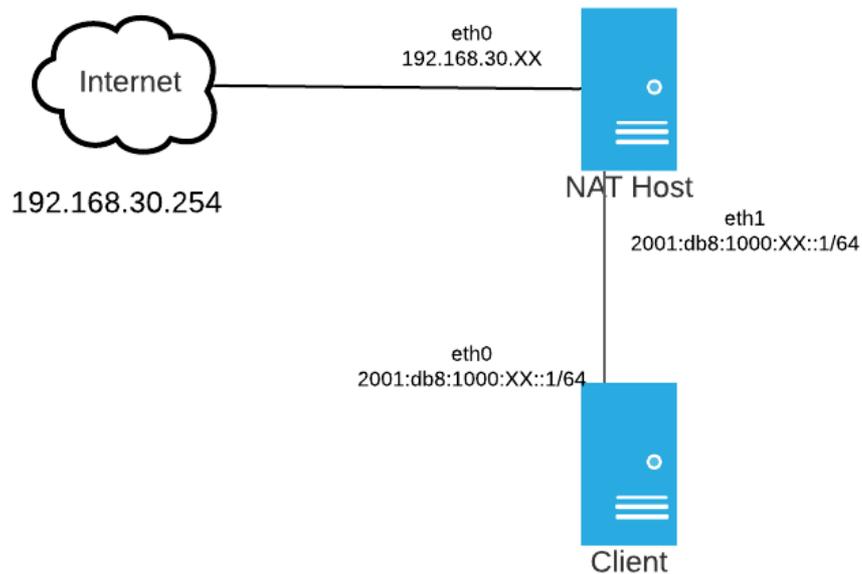


LAB: DNS64/NAT64/464XLAT

Lab Environment

- The Lab topology has:
 - 1 x DNS/NAT Host (IPv4 And IPv6)
 - 1 X Client (IPv6 Only)



Address Plan

Group Number	NAT Server IPv4	Nat Server IPv6	Client IPv6
10	192.168.30.10	2001:db8:1000:10::1/64	2001:db8:1000:10::2/64
11	192.168.30.11	2001:db8:1000:11::1/64	2001:db8:1000:11::2/64
10	192.168.30.12	2001:db8:1000:12::1/64	2001:db8:1000:12::2/64
.....
40	192.168.30.40	2001:db8:1000:40::1/64	2001:db8:1000:40::2/64

Log onto your NAT Server from the assigned Jumphost

```
ssh apnic@192.168.30.xx
```

Where XX is your group number

Username: `apnic`

Password: `training`

Check the initial state

1. From the NAT Server, SSH to your Client IPv6 Address

```
ssh apnic@2001:db8:1000:xx::2
```

2. We will be using our training apt cache server as a test as it only has a IPv4 address lets check our DNS records.

```
dig apt.apnictraining.net
```

The Output should look like:

```
>>> dig 9.11.3-1ubuntu1.13-Ubuntu <<<> apt.apnictraining.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27150
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 99352a884dc67e08578e64d85f601fbebe0635b3a36076c3 (good)
;; QUESTION SECTION:
;apt.apnictraining.net.      IN      A

;; ANSWER SECTION:
apt.apnictraining.net.  604800 IN      A      192.168.30.248

;; AUTHORITY SECTION:
apt.apnictraining.net.  604800 IN      NS
ns.apnictraining.net.

;; ADDITIONAL SECTION:
ns.apnictraining.net.  604800 IN      A      192.168.30.10

;; Query time: 1 msec
;; SERVER: 2001:db8:1000:10::1#53(2001:db8:1000:10::1)
;; WHEN: Tue Sep 15 01:58:22 UTC 2020
;; MSG SIZE rcvd: 127
```

3. Now Lets check and see if there is an IPv6 Record

```
dig apt.apnictraining.net AAAA
```

This should return no resulting record

4. `exit` back to your NAT Server to continue with the next step

Setup DNS 64

1. Edit the /etc/bind/named.conf.options

Insert the Below under `listen-on-v6 { any; };`

(hint: `sudo nano /etc/bind/named.conf.options`)

```
dns64 2001:db8:624:624:624:XX::/96 {
    clients {any; };
    mapped { any; };
    suffix ::;
    break-dnssec yes;
};
```

Where XX in the translation prefix is your group Number.

DO NOT USE THE ABOVE CONFIG IN PRODUCTION - this is a very basic example of DNS64

(hint: to exit and save in nano Ctrl+W to save and Ctrl+X to exit)

2. Restart Bind

```
sudo systemctl restart bind9
```

3. Check that Bind has started correctly and correct any errors listed, restart bind if required

```
sudo systemctl status bind9
```

4. SSH to your [Client](#) from your [Nat Server](#)

```
ssh apnic@2001:db8:1000:XX::2
```

Where XX is your group number

4. Lookup our test IPv4 dns name

```
dig apt.apnictraining.net
```

The Output should not have changed from the previous check

5. Now lookup the IPv6 Address

```
dig apt.apnictraining.net AAAA
```

Output:

```
; <<>> DiG 9.11.3-1ubuntu1.13-Ubuntu <<>> apt.apnictraining.net AAAA
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16474
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 50509affb63e41d8ad85354a5f5ef7550907082c15c32560 (good)
```

```
;; QUESTION SECTION:
;apt.apnictraining.net.      IN      AAAA

;; ANSWER SECTION:
apt.apnictraining.net. 604800 IN      AAAA
2001:db8:624:624:10:c0a8:1ef8

;; AUTHORITY SECTION:
apt.apnictraining.net. 604800 IN      NS      ns.apnictraining.net.

;; Query time: 1 msec
;; SERVER: 2001:db8:1000:10::1#53(2001:db8:1000:10::1)
;; WHEN: Mon Sep 14 04:53:41 UTC 2020
;; MSG SIZE rcvd: 123
```

6. Now try and ping our address by name

```
ping apt.apnictraining.net
```

It will resolve the address, but you should get destination unreachable.

`exit` out of your ssh session to your Client.

Install the NAT64

1. On the [NAT Server](#) we will install JOOL to act as our our NAT64

```
cd /home/apnic/  
wget https://www.jool.mx/download/jool-tools_4.1.3-1_amd64.deb  
sudo apt install -y ./jool-tools_4.1.3-1_amd64.deb
```

- On a non-containerised install, you would also need to install the dkms role as well

2. Start Jool

```
sudo jool instance add "Groupxx" --netfilter \  
--pool6 2001:db8:624:624:624:xx::/96
```

Where xx is your group number

3. Verify that you Jool session is running.

```
sudo jool instance display
```

You should see a running jool instance.

4. Now ssh into to your [Client](#).

```
ssh apnic@2001:db8:1000:xx::2
```

5. Ping our host again.

```
ping apt.apnictraining.net
```

You should have results similar to the below:

```
PING apt.apnictraining.net(2001:db8:624:624:624:10:c0a8:1ef8  
(2001:db8:624:624:624:10:c0a8:1ef8)) 56 data bytes  
64 bytes from 2001:db8:624:624:624:10:c0a8:1ef8  
(2001:db8:624:624:624:10:c0a8:1ef8): icmp_seq=1 ttl=63 time=0.225 ms  
64 bytes from 2001:db8:624:624:624:10:c0a8:1ef8  
(2001:db8:624:624:624:10:c0a8:1ef8): icmp_seq=2 ttl=63 time=0.142 ms
```

But you can't ping any V4 literals yet

```
root@cli11:/# ping 192.168.30.248  
connect: Network is unreachable
```

Congratulations, you now have NAT64/DNS64 running!

If you HAVE NOT got the above working ask for assistance as the rest of the lab requires a working DNS64/NAT64 configuration

CLAT Installation

Next we will install the other half of our XLAT - the CLAT

1. On the Client we will need to install some software.

CLATD is perl/tayga implementation of a CLAT.

```
cd /home/apnic
sudo apt install -y build-essential
git clone https://github.com/toreanderson/clatd
cd clatd
sudo make
sudo make install
```

2. Now we will install Tayga which is a CLAT 'client'

```
sudo apt install tayga
```

As CLATD has some perl dependencies, we will have to install those as well.

3. Launch the Perl CPAN Environment

```
perl -MCPAN -e shell
```

You will be asked about configuration

```
would you like to configure as much as possible automatically? [yes]
```

Press `Enter` to accept the default

```
what approach do you want? (Choose 'local::lib', 'sudo' or 'manual')
[local::lib]
```

Enter `sudo` at this prompt

You are now in the CPAN shell. You should see something like the below

```
cpan shell -- CPAN exploration and modules installation (v2.18)
Enter 'h' for help.

cpan[1]>
```

In the Shell you will need to enter these commands **one at a time**.

Once each step has finished, move onto the next:

```
install Net::IP
install IO::Socket::INET6
install Net::DNS
reload cpan
```

These steps could take some time, so be patient.

(you may see an error at the end of the NET::DNS install about THOR/Net-LibIDN2.

This is safe to ignore)

After cpan has reloaded you can `exit` out of the perl shell

4. Let see if our CLAT service is running

```
sudo service clatd status
```

If the service is not running

```
sudo service clatd restart
```

Once the service is running you will see something like:

```
apnic@CLI11:~$ sudo service clatd status
• clatd.service - 464XLAT CLAT daemon
  Loaded: loaded (/etc/systemd/system/clatd.service; enabled; vendor
  preset: enabled)
  Active: active (running) since Mon 2020-09-14 05:48:10 UTC; 22s ago
  Docs: man:clatd(8)
  Main PID: 5407 (clatd)
  Tasks: 2 (limit: 4915)
  CGroup: /system.slice/clatd.service
          └─5407 /usr/bin/perl -w /usr/sbin/clatd
            └─5422 tayga --config /tmp/a764fkM8uh --nodetach

Sep 14 05:48:21 CLI11.apnictraining.net clatd[5407]: Enabling Proxy-ND for
2001:db8:1000:10:4b05:218a:6378:0 on eth0
Sep 14 05:48:21 CLI11.apnictraining.net clatd[5407]: Creating and
configuring up CLAT device 'clat'
Sep 14 05:48:21 CLI11.apnictraining.net clatd[5407]: Created persistent tun
device clat
Sep 14 05:48:21 CLI11.apnictraining.net clatd[5407]: Adding IPv4 default
route via the CLAT
Sep 14 05:48:21 CLI11.apnictraining.net clatd[5407]: Starting up TAYGA,
using config file '/tmp/a764fkM8uh'
Sep 14 05:48:21 CLI11.apnictraining.net tayga[5422]: starting TAYGA 0.9.2
Sep 14 05:48:21 CLI11.apnictraining.net tayga[5422]: Using tun device clat
with MTU 1500
Sep 14 05:48:21 CLI11.apnictraining.net tayga[5422]: TAYGA's IPv4 address:
192.0.0.2
Sep 14 05:48:21 CLI11.apnictraining.net tayga[5422]: TAYGA's IPv6 address:
2001:db8:624:624:624:10:c000:2
Sep 14 05:48:21 CLI11.apnictraining.net tayga[5422]: NAT64 prefix:
2001:db8:624:624:624:10::/96
```

5. Now lets ping our v4 names and addresses again

```
apnic@CLI11:~$ ping apt.apnictraining.net
PING apt.apnictraining.net(2001:db8:624:624:624:10:c0a8:1ef8
(2001:db8:624:624:624:10:c0a8:1ef8)) 56 data bytes
64 bytes from 2001:db8:624:624:624:10:c0a8:1ef8
(2001:db8:624:624:624:10:c0a8:1ef8): icmp_seq=1 ttl=63 time=0.935 ms
64 bytes from 2001:db8:624:624:624:10:c0a8:1ef8
(2001:db8:624:624:624:10:c0a8:1ef8): icmp_seq=2 ttl=63 time=0.104 ms
64 bytes from 2001:db8:624:624:624:10:c0a8:1ef8
(2001:db8:624:624:624:10:c0a8:1ef8): icmp_seq=3 ttl=63 time=0.164 ms
^C
```

```
--- apt.apnictraining.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.104/0.401/0.935/0.378 ms
apnic@CLI11:~$ ^C
apnic@CLI11:~$ ping 192.168.30.248
PING 192.168.30.248 (192.168.30.248) 56(84) bytes of data.
64 bytes from 192.168.30.248: icmp_seq=1 ttl=61 time=148 ms
64 bytes from 192.168.30.248: icmp_seq=2 ttl=61 time=0.305 ms
64 bytes from 192.168.30.248: icmp_seq=3 ttl=61 time=0.263 ms
64 bytes from 192.168.30.248: icmp_seq=4 ttl=61 time=0.321 ms
```