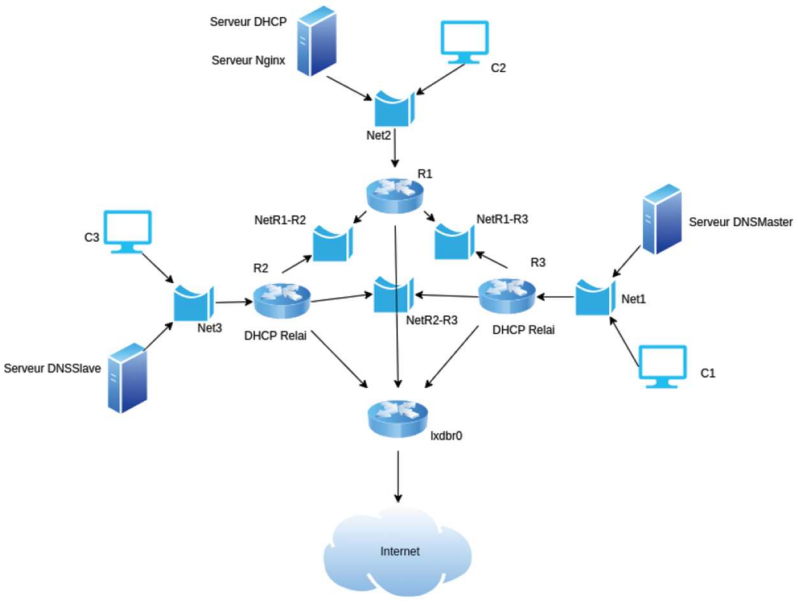
TP4 – Network Administration
- DHCP, DNS –

IFRIM Vasile-Alexandru
M2 SSI

1 Network Architecture and interface configuration



Main network		Routers interconnections	
Net1 192.168.10.0/26	Router **R1** **eth1** 192.168.10.1	**NetR1-R2** 192.168.10.120/30	Router **R1** **eth2**
	Client C1 192.168.10.2		Router **R2** **eth2**
	DNS Master 192.168.10.3		
Net2 192.168.10.64/27	Router **R2** **eth1** 192.168.10.65	**NetR1-R3** 192.168.10.116/30	Router **R1** **eth3**
	Client C2 192.168.10.66		Router **R3** **eth3**
	DHCP, Nginx 192.168.10.67		
Net3 192.168.10.96/28	Router **R3** **eth1** 192.168.10.97	**NetR2-R3** 192.168.10.112/30	Router **R2** **eth3**
	Client C3 192.168.10.98		Router **R3** **eth3**
	DNS Slave 192.168.10.99		
Each router is connect to a main bridge, **lxdbr0** via their **eth0** interface to have Internet access.			

To create this environment we will be using LXD containers, based on the latest Ubuntu image 24.04. We structure the configuration of the environment as follows:

```
root Jan 20 16:33 .../alex/Reseau/4.DNS_DHCP > tree Config-files/
[4.0K]  Config-files//
├── [4.0K]  C1/
│   └── [ 308]  50-cloud-init.yaml
├── [4.0K]  C2/
│   └── [ 320]  50-cloud-init.yaml
├── [4.0K]  C3/
│   └── [ 310]  50-cloud-init.yaml
├── [4.0K]  DHCP/
│   ├── [ 311]  50-cloud-init.yaml
│   └── [ 649]  dhcpd.conf
├── [4.0K]  DNSMaster/
│   ├── [ 315]  50-cloud-init.yaml
│   ├── [ 828]  db.ssi.edu
│   ├── [ 276]  named.conf.local
│   └── [ 307]  named.conf.options
├── [4.0K]  DNSSlave/
│   ├── [ 246]  50-cloud-init.yaml
│   └── [ 272]  named.conf.local
├── [4.0K]  R1/
│   └── [ 629]  50-cloud-init.yaml
├── [4.0K]  R2/
│   └── [ 612]  50-cloud-init.yaml
└── [4.0K]  R3/
    └── [ 615]  50-cloud-init.yaml
```

Each machine configuration can be found in its respective folder, in a YAML file. Using a single script we will all the configurations to all of them.

The routers need IPv4 forwarding and NAT enabled, which can be done by running the following commands. This will allow an Internet connection for servers and clients on each subnetwork.

```
$ lxc exec <router> -- sysctl -w net.ipv4.ip_forward=1
$ lxc exec <router> -- iptables -t nat -A POSTROUTING -o eth0 -j
  MASQUERADE
$ lxc exec <router> -- netfilter-persistent save
```

Note: during the execution of the configuration script, the installation of the iptables-persistent package will prompt to save or not the current firewall rules. It is not important what option we choose at that point.

```
root Jan 20 19:13 .../alex/Reseau/4.DNS_DHCP > lxc ls
+-----------+---------+----------------------+------+-----------+-----------+
|   NAME    |  STATE  |         IPV4         | IPV6 |   TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+------+-----------+-----------+
| C1        | RUNNING | 192.168.10.2 (eth1)  |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| C2        | RUNNING | 192.168.10.66 (eth1) |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| C3        | RUNNING | 192.168.10.98 (eth1) |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| DHCP      | RUNNING | 192.168.10.67 (eth1) |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| DNSMaster | RUNNING | 192.168.10.3 (eth1)  |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| DNSSlave  | RUNNING | 192.168.10.99 (eth1) |      | CONTAINER | 0         |
+-----------+---------+----------------------+------+-----------+-----------+
| R1        | RUNNING | 192.168.10.65 (eth1) |      | CONTAINER | 0         |
|           |         | 192.168.10.121 (eth3)|      |           |           |
|           |         | 192.168.10.118 (eth2)|      |           |           |
|           |         | 10.1.1.24 (eth0)     |      |           |           |
+-----------+---------+----------------------+------+-----------+-----------+
| R2        | RUNNING | 192.168.10.97 (eth1) |      | CONTAINER | 0         |
|           |         | 192.168.10.122 (eth3)|      |           |           |
|           |         | 192.168.10.114 (eth2)|      |           |           |
|           |         | 10.1.1.44 (eth0)     |      |           |           |
+-----------+---------+----------------------+------+-----------+-----------+
| R3        | RUNNING | 192.168.10.117 (eth2)|      | CONTAINER | 0         |
|           |         | 192.168.10.113 (eth3)|      |           |           |
|           |         | 192.168.10.1 (eth1)  |      |           |           |
|           |         | 10.1.1.4 (eth0)      |      |           |           |
+-----------+---------+----------------------+------+-----------+-----------+
```

2 DNS

The principal DNS server, DNSMaster, will be in the `net1` network, serving a zone file for "ssi.edu". It will allow forwarding requests to a secondary DNS server, DNSSlave, using its IP:

```
zone "ssi.edu" {
    type master;
    file "/etc/bind/db.ssi.edu";
    allow-transfer { 192.168.10.99; };
};
```

DNSMaster - /etc/bind/named.conf.local

The zone file will include records for all components of our different networks. It will also include aliases and our basic website:

```
$TTL    604800
@       IN      SOA     ns1.ssi.edu. admin.ssi.edu. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

        IN      NS      ns1.ssi.edu.
ns1     IN      A       192.168.10.3
ns2     IN      A       192.168.10.99
C1      IN      A       192.168.10.2
R1      IN      A       192.168.10.1
C2      IN      A       192.168.10.66
DHCP    IN      A       192.168.10.67
R2      IN      A       192.168.10.65
R3      IN      A       192.168.10.97
C3      IN      A       192.168.10.98
DNSMaster   IN  CNAME   ns1
DNSSlave    IN  CNAME   ns2
www     IN      A       192.168.10.67
```

Then the server is configured to forward any other requests to "real" DNS servers; we use the ones of Google. As for the secondary DNS server, once the configurations are put in place and the services restarted, the database will be available on its local system at `/var/cache/bind`.

```
options {
    directory "/var/cache/bind";

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    allow-query { any; };
    recursion yes;
    forward only;
};
```

DNSMaster - /etc/bind/named.conf.options

```
zone "ssi.edu" {
    type slave;
    file "/var/cache/bind/db.ssi.edu";
    masters { 192.168.10.3; };
};
```

DNSSlave - /etc/bind/named.conf.local

This way, when a client or server wants to access a resource that is not in our DNS zone, its DNS request will be forwarded from DNSMaster, through the routers, out the `lxdbr0` interface, to one of Google's DNS servers.

Testing our connectivity in the network and our DNS servers solving requests:

```
root@C1:~# ping DHCP
PING DHCP.ssi.edu (192.168.10.67) 56(84) bytes of data.
64 bytes from 192.168.10.67: icmp_seq=1 ttl=62 time=0.215 ms
64 bytes from 192.168.10.67: icmp_seq=2 ttl=62 time=0.142 ms
64 bytes from 192.168.10.67: icmp_seq=3 ttl=62 time=0.100 ms
^C
--- DHCP.ssi.edu ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.100/0.152/0.215/0.047 ms
root@C1:~# ping DNSSlave
PING ns2.ssi.edu (192.168.10.99) 56(84) bytes of data.
64 bytes from 192.168.10.99: icmp_seq=1 ttl=62 time=0.204 ms
64 bytes from 192.168.10.99: icmp_seq=2 ttl=62 time=0.112 ms
^C
--- ns2.ssi.edu ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.112/0.158/0.204/0.046 ms
root@C1:~# ping www.ssi.edu
PING www.ssi.edu (192.168.10.67) 56(84) bytes of data.
64 bytes from 192.168.10.67: icmp_seq=1 ttl=62 time=0.155 ms
64 bytes from 192.168.10.67: icmp_seq=2 ttl=62 time=0.100 ms
^C
--- www.ssi.edu ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.100/0.127/0.155/0.027 ms
```

```
root@C1:~# dig ns1.ssi.edu

; <<>> DiG 9.18.30-0ubuntu0.24.04.1-Ubuntu <<>> ns1.ssi.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11452
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ns1.ssi.edu.                   IN      A

;; ANSWER SECTION:
ns1.ssi.edu.            7184    IN      A       192.168.10.3

;; Query time: 1 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Jan 20 17:42:43 UTC 2025
;; MSG SIZE  rcvd: 56
```

And we can also communicate with the outside world, seeing that our traffic is properly routed to a router, then out the `lxdbr0` and out the host machine:

```
root@C1:~# ping example.com
PING example.com (23.215.0.138) 56(84) bytes of data.
64 bytes from a23-215-0-138.deploy.static.akamaitechnologies.com (23.215.0.138): icmp_seq=1 ttl=1
26 time=92.4 ms
64 bytes from a23-215-0-138.deploy.static.akamaitechnologies.com (23.215.0.138): icmp_seq=2 ttl=1
26 time=91.6 ms
64 bytes from a23-215-0-138.deploy.static.akamaitechnologies.com (23.215.0.138): icmp_seq=3 ttl=1
26 time=92.5 ms
64 bytes from a23-215-0-138.deploy.static.akamaitechnologies.com (23.215.0.138): icmp_seq=4 ttl=1
26 time=92.6 ms
```

```
root@C1:~# traceroute example.com
traceroute to example.com (23.192.228.84), 64 hops max
  1   192.168.10.1  0.004ms  0.002ms  0.002ms
  2   10.1.1.1  0.002ms  0.001ms  0.002ms
  3   192.168.38.2  0.003ms  0.078ms  0.003ms
  4   *  *  *
  5   *  *  *
  6   *  *  *
```

3 DHCP

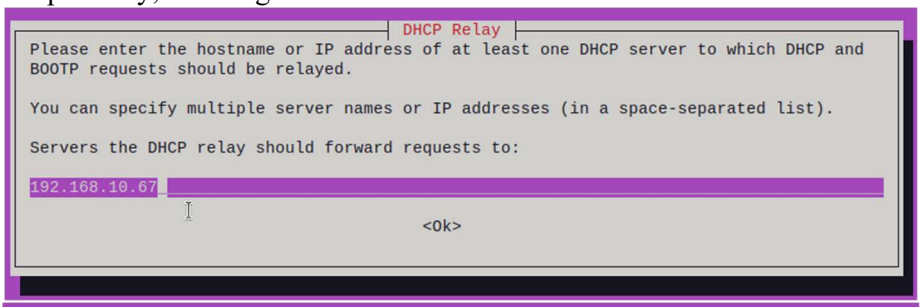
DHCP enables communication and synchronization of configurations to new machines that arrive in the network, thus avoiding having to configure everything by hand - the DHCP server takes care of it by providing an extensive list of options. For each subnetwork, we configured the pool of dynamically assignable addresses and we specify the gateways ("where do we go if we cannot find something in the local network?" - the routers). Moreover, we specify domain-name-servers and domain-name for each subnet, which could also be declared globally, if no other DNS zones are meant to be declared.

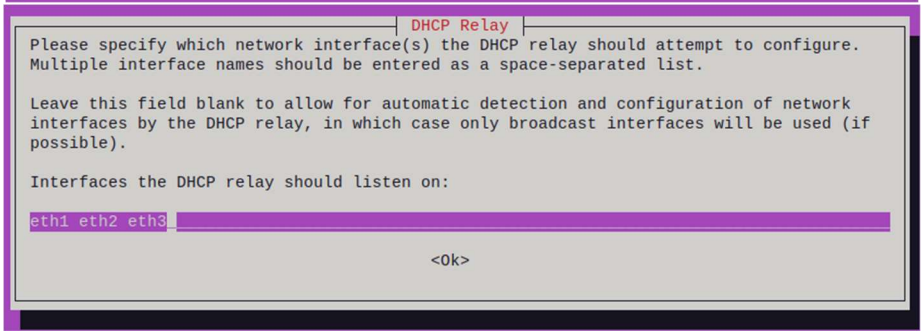
```
subnet 192.168.10.0 netmask 255.255.255.192 {
    range 192.168.10.4 192.168.10.62;
    option routers 192.168.10.1;
    option domain-name-servers 192.168.10.3, 192.168.10.99;
    option domain-name "ssi.edu";
}

subnet 192.168.10.64 netmask 255.255.255.224 {
    range 192.168.10.68 192.168.10.95;
    option routers 192.168.10.65;
    option domain-name-servers 192.168.10.3, 192.168.10.99;
    option domain-name "ssi.edu";
}

subnet 192.168.10.96 netmask 255.255.255.240 {
    range 192.168.10.100 192.168.10.110;
    option routers 192.168.10.97;
    option domain-name-servers 192.168.10.3, 192.168.10.99;
    option domain-name "ssi.edu";
}
```

However, for the clients to actually get a DHCP configuration from the server, their DHCPDISCOVER requests must reach, must be routed to the right server – here is where DHCP relays come into play. Since the DHCP server is part of the `net2` network, the relays, located on routers R2 and R3, forward the requests to the DHCP server. The configuration of these relays is created automatically after the install process. During the installation we are prompted for setting the IP address where the relay should forward requests, which interfaces should dhcp-relay attempt to configure, and optionally, extra arguments. For our environment:

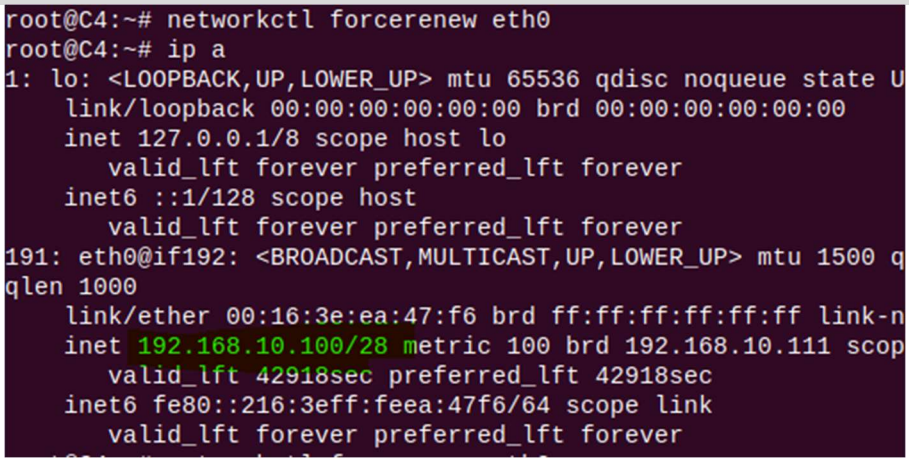




To test the the DHCP configuration we could add a new container to one of the subnets and see that it automatically has a correct IPv4 address, which we can also forcerenew (although the value won't

change because of how the DHCP server, in its current state, tries to always map the same IP address to a certain MAC address).

```
$ lxc launch ubuntu:24.04 C4 --network=net3
$ lxc shell C4
# ip a
# networkctl renew eth0
```



And also see that we have internet access: