Management and Operations of Networks, Services, and Systems

An organization's network

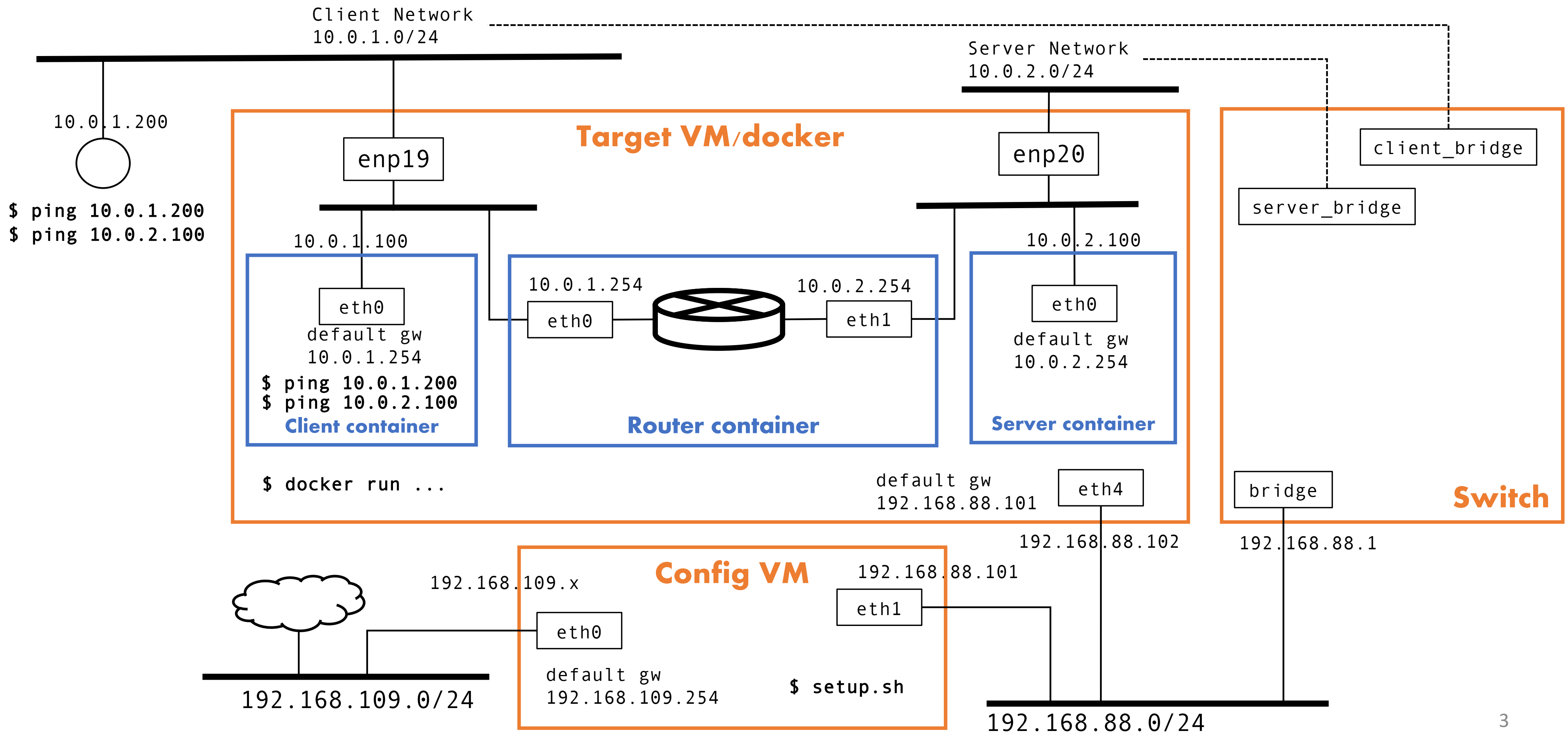
Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC

TOC

- Two networks with a router
- DHCP for hosts in the physical network
- Web server and proxy
- Route to the Internet
- DMZ, NAT
- DNS

Two networks with a router



Setup

```
sudo docker rm -f client server router
sudo docker network rm client_net server_net
sudo ip l set ens19 up
sudo ip l set ens20 up
```

Networks

```
sudo docker network create -d macvlan --
subnet=10.0.1.0/24 --gateway=10.0.1.1 -o
parent=ens19 client_net

sudo docker network create -d macvlan --
subnet=10.0.2.0/24 --gateway=10.0.2.1 -o
parent=ens20 server_net
```

Client and server

```
sudo docker run -d --net client_net --ip
10.0.1.100 --cap-add=NET_ADMIN --name client
netubuntu

sudo docker run -d --net server_net --ip
10.0.2.100 --cap-add=NET_ADMIN --name server
netubuntu
```

Router

```
sudo docker run -d --net client_net --ip
10.0.1.254 --cap-add=NET_ADMIN --name router
netubuntu

sudo docker network connect server_net
router --ip 10.0.2.254
```

Routing on client and server

```
sudo docker exec client /bin/bash -c 'ip r
del default via 10.0.1.1'

sudo docker exec client /bin/bash -c 'ip r a
10.0.2.0/24 via 10.0.1.254'

sudo docker exec server /bin/bash -c 'ip r
del default via 10.0.2.1'

sudo docker exec server /bin/bash -c 'ip r a
10.0.1.0/24 via 10.0.2.254’
```

Test

```
docker exec -it client ping 10.0.2.100
```

Build ‘netubuntu’ image with network tools

Create these files, copy to target host

>> **baseimage/Dockerfile**

```
FROM ubuntu:20.04

RUN apt update && apt install -y vim
iproute2 iputils-ping tcpdump
iptables dnsutils curl

COPY sleep.sh /root/sleep.sh

CMD /root/sleep.sh
```

>> **baseimage/sleep.sh**

```
#!/bin/bash

while true ; do /bin/sleep 5m; done
```

Build netubuntu image

>> **run on docker host**

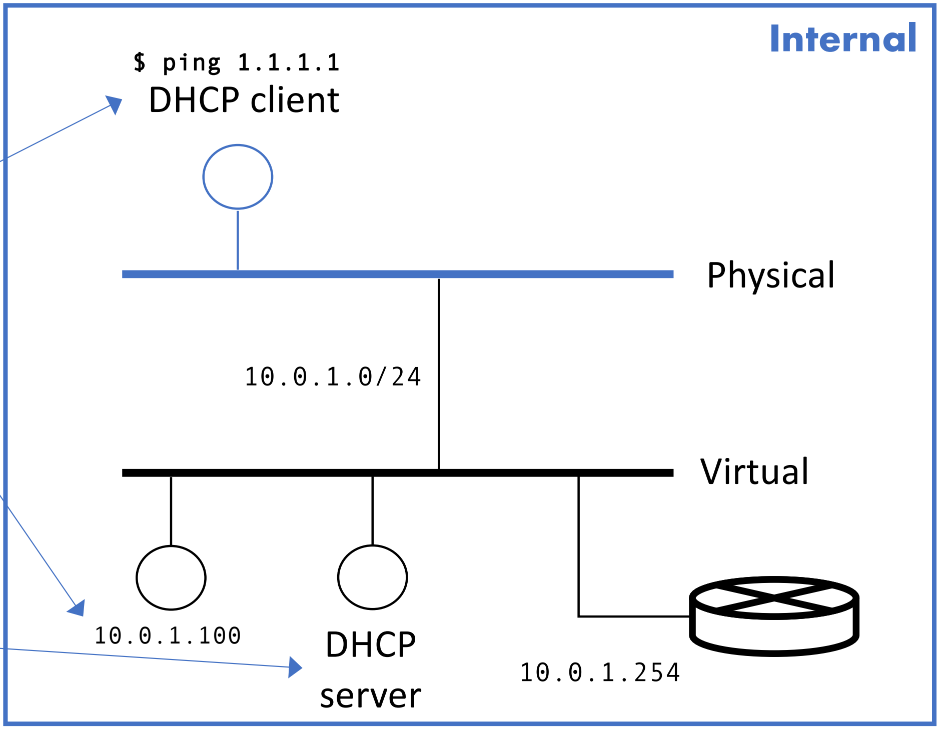
```
sudo docker build --tag
netubuntu:latest ~/baseimage
```

DHCP server on client network

DHCP client needs to be on the physical part of the network – won’t work in docker container.

IP addresses for containers are managed by docker directly, no support for DHCP

The DHCP server can be on the virtual part of the network



Setup configuration file

>> **dhcp.conf**

```
default-lease-time 600;
max-lease-time 7200;
authoritative;
option rfc3442-classless-static-routes code 121 = array of integer 8;
subnet 10.0.1.0 netmask 255.255.255.0 {
range 10.0.1.64 10.0.1.127;
option routers 10.0.1.254;
option rfc3442-classless-static-routes 8,10,10,0,1,254;
option domain-name-servers 10.0.1.1;
}
```

>> **Dockerfile tag dhcp**

```
FROM ubuntu:latest
RUN apt update && apt install -y isc-dhcp-server
RUN touch /var/lib/dhcp/dhcpd.leases
CMD ["/usr/sbin/dhcpd", "-4", "-f", "-d", "--no-pid", "-cf", "/etc/dhcp/dhcpd.conf"]
```

Run dhcp server

```
sudo docker run -d --rm --net client_net --ip 10.0.1.2 --cap-add=NET_ADMIN -v /path/to/dhcp.conf:/etc/dhcp/dhcpd.conf dhcp
```

Configure dhcp client

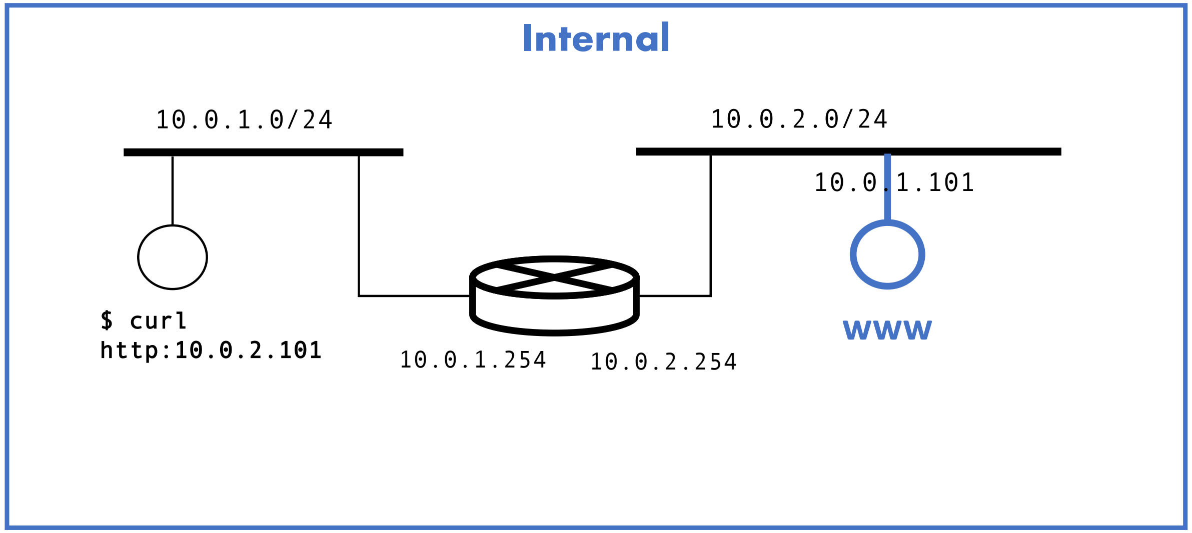
```
sudo dhclient
```

Adds route to **10.0.0.0/8 (8,10)**

via **10.0.1.254 (10,0,1,254)**

on dhcp client

Web server



Web server IP: 10.0.1.101

Simple HTTP page

Test with curl client

Nginx

```
docker run --name nginxint
--rm \
-v
/home/gors/www/internal:/us
r/share/nginx/html:ro \
--net server_net --ip
10.0.2.101 --cap-
add=NET_ADMIN \
-d mynginx
```

```
sudo docker exec nginxint
ip r d default via 10.0.2.1
sudo docker exec nginxint
ip r a default via
10.0.2.254
```

>> www/internal/index.html

```
<p>My Nginx</p>
```

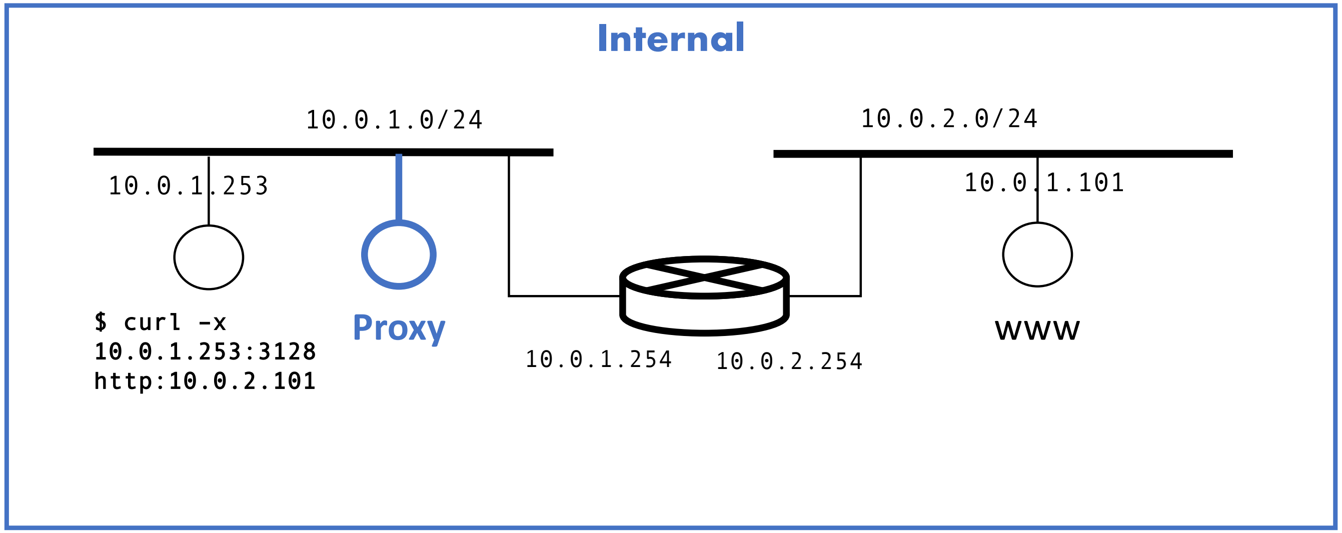
>> Dockerfile tag mynginx

```
FROM nginx:latest
RUN apt update && apt
install -y vim iproute2
iputils-ping
```

Test

```
sudo docker exec -it client
/usr/bin/curl
http://10.0.2.101/
```

Proxy



```
$ curl -x
10.0.1.253:3128
http:10.0.2.101
```

Proxy: 10.0.1.253 port 3128

Accept only HTTP

Accept requests from local network only

Prevent direct HTTP traffic from clients to the outside of 10.0.1.0/24

Squid

```
docker run -d --name squid -e TZ=UTC -v
/home/gors/etcsquid/squid.conf:
/etc/squid/squid.conf  --rm --net client_net
--ip 10.0.1.253 --cap-add=NET_ADMIN mysquid
```

```
sudo docker exec squid ip r d default via
10.0.1.1
```

```
sudo docker exec squid ip r a default via
10.0.1.254
```

>> **ectsquid/squid.conf**

```
acl Safe_ports port 80

acl localnet src 10.0.1.0/24

http_access deny !Safe_ports

http_access allow localnet

http_access deny all

http_port 3128
```

>> **Dockerfile tag mysquid**

```
FROM ubuntu/squid:latest

RUN apt update && apt install -y vim
iproute2 iputils-ping
```

Test 1

```
sudo docker exec -it client /usr/bin/curl
http://10.0.2.101/
```

```
sudo docker exec -it client /usr/bin/curl -x
10.0.1.253:3128 http://10.0.2.101/
```

```
sudo docker exec -it client /usr/bin/curl
https://10.0.2.101/
```

```
sudo docker exec -it client /usr/bin/curl -x
10.0.1.253:3128 https://10.0.2.101/
```

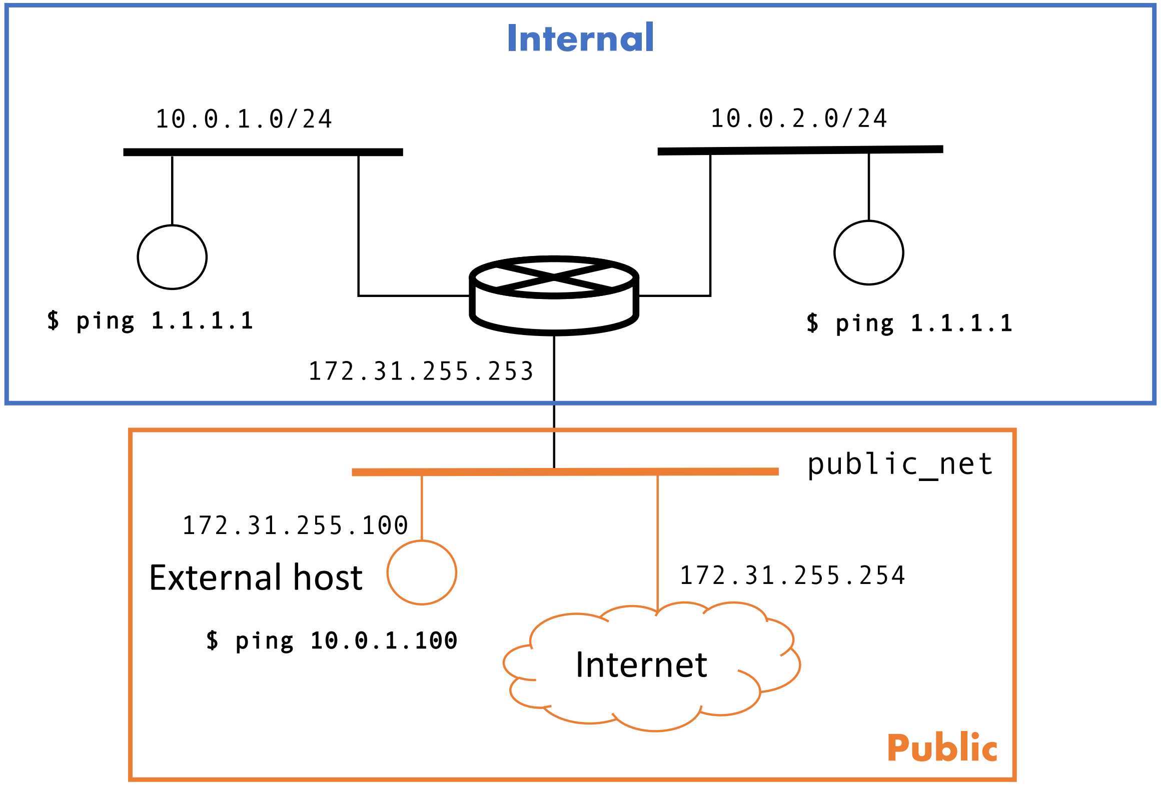
Router

```
sudo docker exec router /bin/bash -c
'iptables -t filter -A FORWARD -p tcp --
dport 80 ! -s 10.0.1.253 -j DROP’
```

Test 2

Run test 1 with the iptables rule

Route to the Internet



```
ISP’s public network:
172.31.255.0/24

Default GW on ISP’s public network:
172.31.255.254

Router’s IP on public network:
172.31.255.253

External host IP on public network:
172.31.255.253
```

Create public network on Docker with default gw

Docker will NAT traffic to the Internet; alternatively setup a macvlan network to your favourite Internet provider

```
sudo docker network create public_net --
subnet=172.31.255.0/24 --gateway=172.31.255.254
```

Connect the router to the public network

```
sudo docker network connect public_net router
--ip 172.31.255.253
```

Update default gateway on the router

```
sudo docker exec router /bin/bash -c 'ip r d
default via 10.0.1.1'
```

```
sudo docker exec router /bin/bash -c 'ip r a
default via 172.31.255.254’
```

Update default gateway on the client and server

```
sudo docker exec client /bin/bash -c 'ip r a
default via 10.0.1.254'
```

```
sudo docker exec server /bin/bash -c 'ip r a
default via 10.0.2.254’
```

Test 1 – client to Internet

```
docker exec -it client ping 1.1.1.1
```

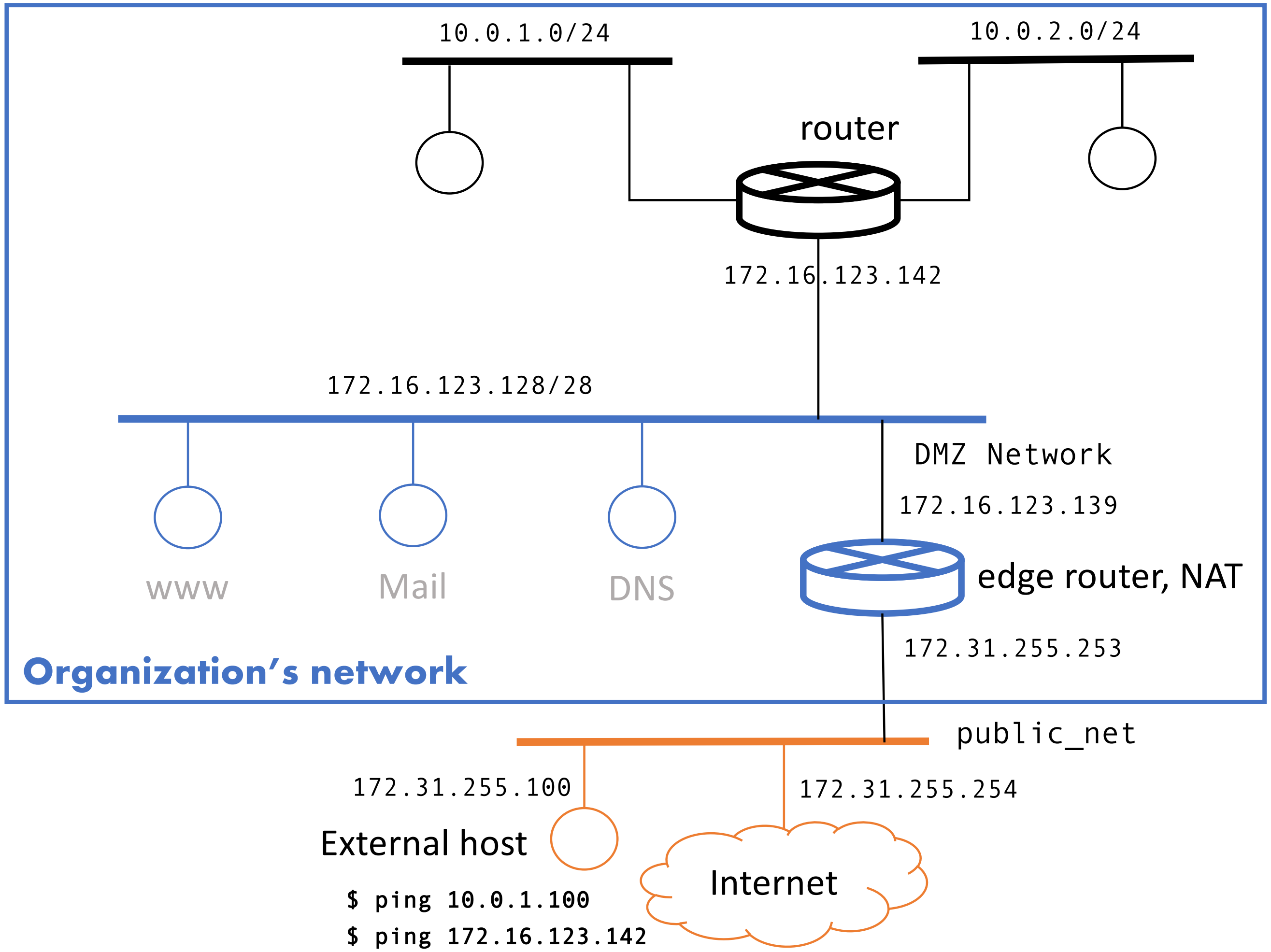
Test 2 – external host to client

```
sudo docker run -d --net public_net --ip
172.31.255.100 --cap-add=NET_ADMIN --name
external_host netubuntu
```

```
sudo docker exec external_host /bin/bash -c 'ip
r a 10.0.0.0/8 via 172.31.255.253'
```

```
docker exec -it external_host ping 10.0.1.100
```

DMZ, NAT



```
$ ping 10.0.1.100
$ ping 172.16.123.142
```

ISP’s public network:
172.31.255.0/24

Default GW on ISP’s public network:
172.31.255.254

Router’s IP on public network:
172.31.255.253

External host IP on public network:
172.31.255.253

Org. public network (DMZ):
172.16.123.128/28

Org. private network:
10.0.0.0/20 (NAT)

Setup external interface and macvlan DMZ on docker host

```
sudo ip l set ens21 up

sudo docker network create -d macvlan --
subnet=172.16.123.128/28 --gateway=172.16.123.140 -o
parent=ens21 dmz_net
```

Let docker know about the DMZ network, and NAT it

```
sudo ip route add 172.16.123.128/28 via 172.31.255.253

sudo iptables -t nat -A POSTROUTING -s 172.16.123.128/28 -o
eth4 -j MASQUERADE
```

Router

```
sudo docker network disconnect public_net router

sudo docker network connect dmz_net router --ip 172.16.123.142

sudo docker exec router /bin/bash -c 'ip r a default via
172.16.123.139’
```

Edge router

```
sudo docker run -d --rm --net dmz_net --ip 172.16.123.139 --
cap-add=NET_ADMIN --name edgerouter netubuntu

sudo docker network connect public_net edgerouter --ip

 172.31.255.253

sudo docker exec edgerouter /bin/bash -c 'ip r d default via
172.16.123.140'

sudo docker exec edgerouter /bin/bash -c 'ip r a default via
172.31.255.254'

sudo docker exec edgerouter /bin/bash -c 'ip r a 10.0.0.0/8 via
172.16.123.142’
```

Don’t forward internal networks, NAT them

```
sudo docker exec edgerouter /bin/bash -c ‘...’
     iptables -t nat –F; iptables -t filter -F
     iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth1
     -j MASQUERADE
     iptables -P FORWARD DROP
     iptables -A FORWARD -m state --state
     ESTABLISHED,RELATED -j ACCEPT
     iptables -A FORWARD -m state --state NEW -i eth0 -j
     ACCEPT
     iptables -A FORWARD -m state --state NEW -i eth1 -d
     172.16.123.128/28 -j ACCEPT
```

External host

```
sudo docker exec externalhost /bin/bash -c 'ip r a
172.16.123.128/28 via 172.31.255.253’
```

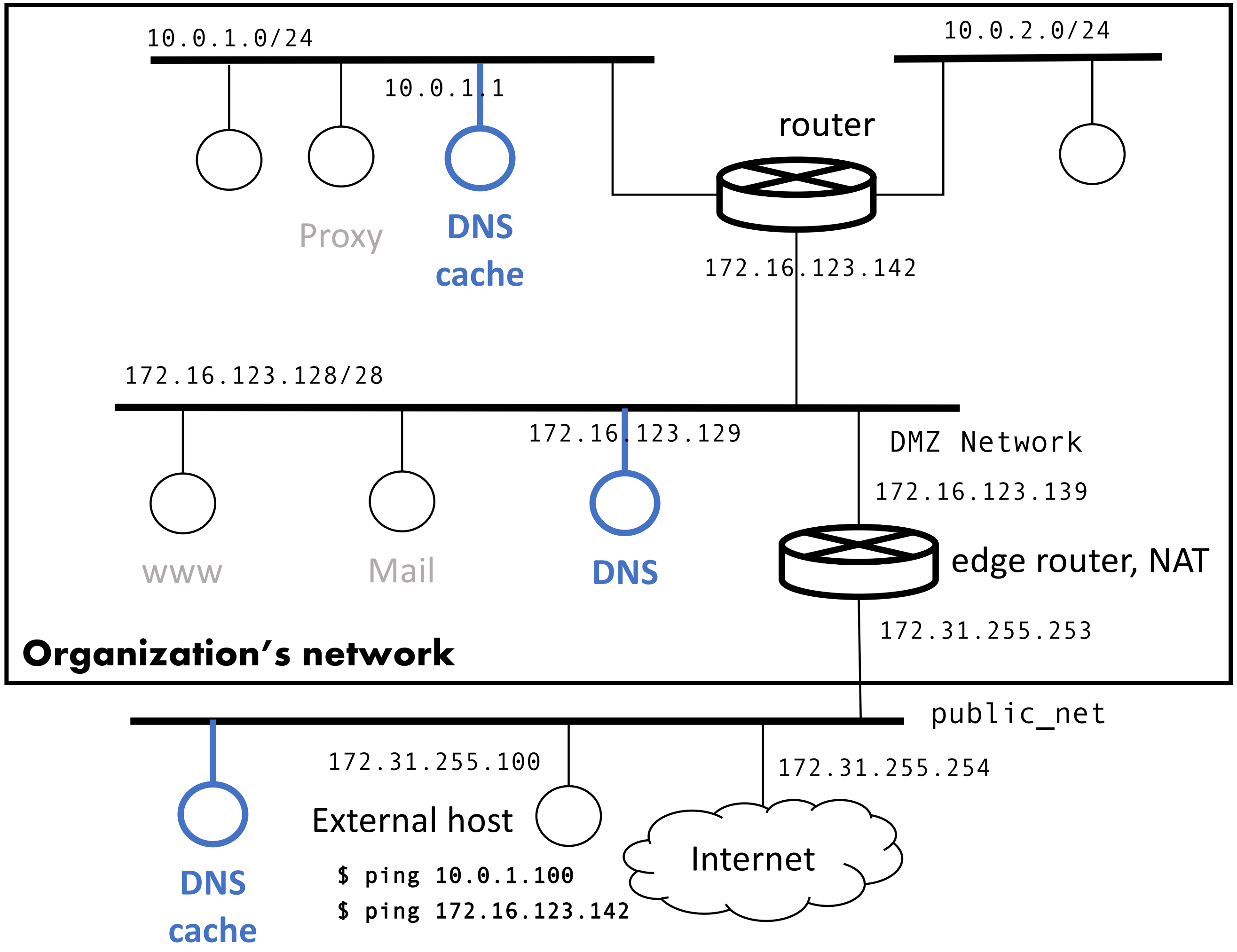
Test from the client

```
sudo docker exec client /bin/ping -c 3 ...

 1.1.1.1 , 10.0.2.100 , 172.16.123.139 , 172.31.255.100
```

Test from the external host

```
sudo docker exec externalhost /bin/ping -c 3 ...

  172.16.123.142 , 10.0.1.100
```

DNS

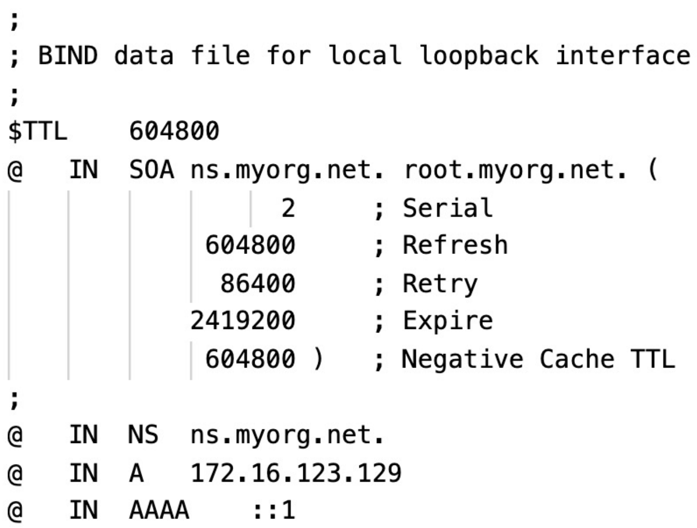


```
$ ping 10.0.1.100
$ ping 172.16.123.142
```

```
DNS: .myorg.net
   internal: servers, clients
   public: www, mail

Split DNS:
  one zone for internal
  another zone for public

DNS cache in client network

DNS cache in public net, test
with external host
```

DNS configuration files

>>**etcbind/db.myorg.net**



```
;
; BIND data file for local loopback interface
;
$TTL    604800
@   IN  SOA ns.myorg.net. root.myorg.net. (
                    2     ; Serial
              604800     ; Refresh
               86400     ; Retry
             2419200     ; Expire
              604800 )   ; Negative Cache TTL
;
@   IN  NS  ns.myorg.net.
@   IN  A   172.16.123.129
@   IN  AAAA    ::1
```

>> **etcbind/named.conf.local**

```
zone "myorg.net" {
        type master;
        file "/etc/bind/db.myorg.net";
};
```

Run the DNS server

```
sudo docker run -d --name=bind9_myorg_auth --
volume
/home/gors/dns/etcbind/db.myorg.net:/etc/bind
/db.myorg.net --volume
/home/gors/dns/etcbind/named.conf.local:/etc/
bind/ named.conf.local --volume
/var/cache/bind --volume /var/lib/bind \ --
volume --rm --net dmz_net --ip 172.16.123.129
--cap-add=NET_ADMIN
internetsystemsconsortium/bind9:9.16
```

```
sudo docker exec bind9_myorg_auth ip r d
default via 172.16.123.140
```

```
sudo docker exec bind9_myorg_auth ip r a
default via 172.16.123.139
```

Assign new names for the services in the DMZ.

Split the DNS to provide names both for public and internal services. Reconfigure the web service with name.

Setup a slave DNS server and internal/external cache DNS servers

Test

```
sudo docker exec externalhost dig
@172.16.123.129 ns.myorg.net
```

```
sudo docker exec client dig @172.16.123.129
www.internal.myorg.net
```

Management and Operations of Networks, Services, and Systems

An organization's network

Ricardo Morla

FEUP – GORS/M.EEC, GRS/M.EIC