Please use a Screen or Video Capture software to save your works!

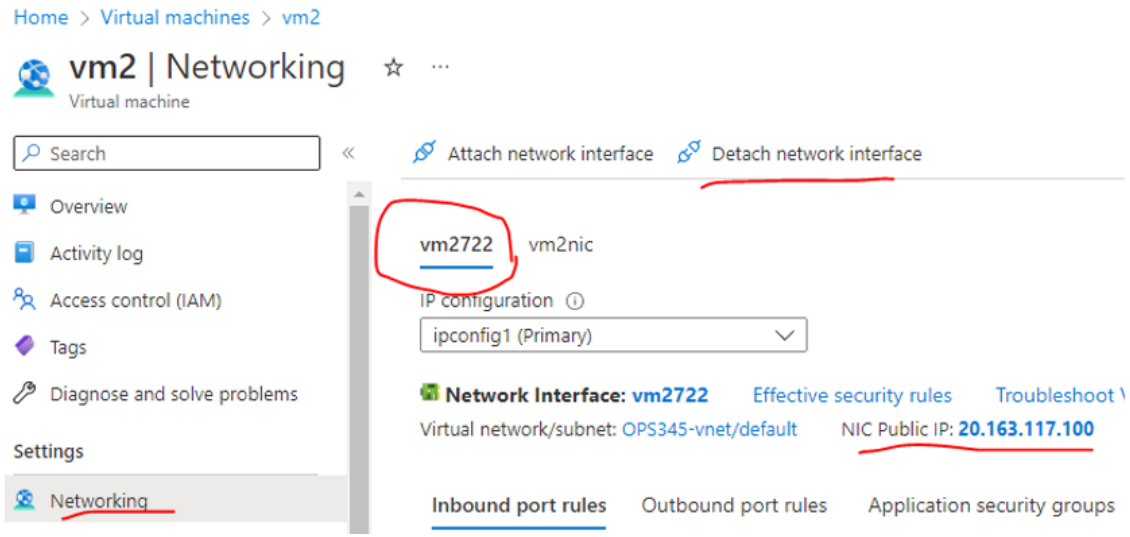
OBJECTIVE & PREPARATION

In this lab, you will configure a Linux machine to be a DNS server for the rest of the machines in your Intranet/LAN. You will use your MySeneca ID (the one based on your name, not your student number) as your domain. The server will handle all queries for names in the yoursenecaid.ops domain. (Note: this is a fake domain name. There is no ".ops" as TLD)

Prep: Attach the network adapter of the public IP of vm2. (Which was removed at Lab 03, case 2, step 6)

Please skip this part if your VM has the correct order for the Virtual Network Card.

6. Now you are sure your webserver is functional, now turn off your webserver (vm2) and detach the network adapter of the public IP, then boot up. So that it is not accessible from Internet (no public IP anymore).
   In my Azure, I have vm2722 as my Public IP network adapter



1. Stop and deallocate the vm2 from Azure portal. Attach the vm2*** NIC to vm2, then START vm2.

2. However, you will find you cannot connect to the vm2 through the public IP. Why?
   Luckily, you should still able to ssh from vm1.
   ssh ops345@192.168.0.20
   You will find the following:
   Default route has been changed to use 192.168.0.1 (eth1) when you remove the NIC for public IP. But 192.168.0.1 doesn't exist. Because there wasn't external public connection to this adapter. And, after reattach the Network adapter, eth1 has been renamed to "cirename0" and in a DOWN status.

```
[ops345@vm2 ~]$ ip addr
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: cirename0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 60:45:bd:77:7e:8e brd ff:ff:ff:ff:ff:ff
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 60:45:bd:c8:39:64 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.4/24 brd 10.0.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::6245:bdff:fec8:3964/64 scope link
       valid_lft forever preferred_lft forever
```

Refer to the following screenshot for side by side comparison.

```
[ops345@vm1 ~]$ ip route
default via 10.0.0.1 dev eth0 proto dhcp metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.5 metric 100
168.63.129.16 via 10.0.0.1 dev eth0 proto dhcp metric 100
169.254.169.254 via 10.0.0.1 dev eth0 proto dhcp metric 100
192.168.0.0/24 dev eth1 proto kernel scope link src 192.168.0.10 metric 101
```

```
[ops345@vm2 ~]$ ip route
default via 192.168.0.1 dev cirename0 proto dhcp metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.4 metric 101
168.63.129.16 via 192.168.0.1 dev cirename0 proto dhcp metric 100
169.254.169.254 via 192.168.0.1 dev cirename0 proto dhcp metric 100
192.168.0.0/24 dev cirename0 proto kernel scope link src 192.168.0.20 metric 100
```

```
[ops345@vm1 ~]$ ip route
default via 10.0.0.1 dev eth0 proto dhcp metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.5 metric 100
168.63.129.16 via 10.0.0.1 dev eth0 proto dhcp metric 100
169.254.169.254 via 10.0.0.1 dev eth0 proto dhcp metric 100
192.168.0.0/24 dev eth1 proto kernel scope link src 192.168.0.10 metric 101
```

```
[ops345@vm2 ~]$ ip route
default via 192.168.0.1 dev cirename0 proto dhcp metric 100
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.4 metric 101
168.63.129.16 via 192.168.0.1 dev cirename0 proto dhcp metric 100
169.254.169.254 via 192.168.0.1 dev cirename0 proto dhcp metric 100
192.168.0.0/24 dev cirename0 proto kernel scope link src 192.168.0.20 metric 100
```

3. The quick solution to this would be put network configuration to its initial settings.
   a. STOP and Deallocate the vm2.
   b. Detach the “vm2nic”. After this step, you will find the “vm2***” network adapter (with Public IP) is now the first adapter, which was the original settings.
   c. Reattach the “vm2nic” and START the vm2.
   d. You will find everything back to original.
4. The above steps will only work if you still have ssh access to vm2. If you don’t have ssh access, the vm2 is orphaned and cannot be access anymore. It would have to be removed and recreated. (Make sure you removed all the vm2 related resources from the OPS345 resource group before recreate the vm2.)
   # In real enterprise environment, remove vm is not a good practice, there are other ways to troubleshoot. However, it is not part of the objective of this OPS345 course.

Case 1: Configuring the DNS Server

Online Resources: Address resolution mechanism & Reverse DNS lookups

How DNS resolution works

We have reviewed in class how the Domain Name Service works. You should also read in your own time the DNS article on Wikipedia for an overview of a DNS query - especially the Address resolution mechanism section (including the Recursive and caching name server subsections). The diagram shown in the WIKI is also quite simple and easy to understand. Domain Name Service is a LARGE topic but this lab provides the basic principles and essentials for this course and DNS set-up for other courses.

We will make the vm1 acting as the DNS server (Primary) and the vm2 acting as Client.

- When your client (vm2) does a name resolution request:
  - The VM contacts the DNS server with LAN, which is running a DNS proxy (a.k.a. stub resolver), not a full DNS server
  - The DNS proxy on the vm1 contacts the Microsoft Azure's DNS server, which was assigned to be used on the vm1 via DHCP. (Internet portion of the vm1)
  - Depending on the type of request, the Azure DNS server may contact other DNS servers to get an answer to the request.
- After you have completed this lab, your server (vm1) will be running a full-featured DNS server (which is how you're going to get the yoursenecaid.ops domain without paying for it). Unfortunately, you will be the only one using your DNS server so no-one else will be able to resolve hosts under yoursenecaid.ops

We will now be installing, configuring and running a DNS server on our Server (vm1).

In most corporate networks, the DNS server would be a standalone VM/Hardware (not build into the Gateway/Router). For the purpose of this lab, DNS is the only concern, we will not look at other functionalities in the Server VM (e.g. Gateway from previous lab is not a concern here).

1. Ensure that your VMs are running, and that none of your VMs have entries in /etc/hosts. In previous courses you may have relied on the contents of /etc/hosts but you will not use them in this course so that you can see how vital a properly functioning DNS server is.
2. Modify the hostname of each VM (server on vm1 and client on vm2) to match the planned DNS entry. This will be important for services we configure in later labs.
   On vm1: hostnamectl set-hostname server
   On vm2: hostnamectl set-hostname client
   Reboot both VM
3. Install bind and bind utilities on your server (vm1).

   yum install bind

   yum install bind-utils # bind-utils include nslookup, host, dig utilities.

4. Your Azure hosted VMs: LAN IPs are: server: 192.168.0.10, client: 192.168.0.20
5. Find your DNS server on the WAN portion of your server VM. Use the first one if there are many entries.

   grep "nameserver" /etc/resolv.conf

```
[root@server ops345]# grep "nameserver" /etc/resolv.conf
nameserver 168.63.129.16
```

   # As of the creation of the lab, DNS server on Azure is 168.63.129.16

6. An authoritative bind server has a global configuration file (named.conf) and at least one zone file for the zone it's authoritative for.
7. When you install Bind you'll get a default /etc/named.conf. Copy this file over to a backup location and empty the original. We will be writing one from scratch with only the following contents, but use your own value where applicable.

```
cp /etc/named.conf /etc/named.conf.original
vi /etc/named.conf
```

```
options {
        directory "/var/named/";
        allow-query {127.0.0.1; 192.168.0.0/24;};
        forwarders { 168.63.129.16; };
};
zone "localhost" {
        type master;
        file "named.localhost";
};
zone "yoursenecaid.ops" {
        type master;
        file "mydb-for-yoursenecaid-ops";
};
```

You need to understand all the options in this file except the localhost zone, so that in the future (for example in a test) you can quickly set up a DNS server for a new zone. Look up in the reference these things and write down what they do:

directory, allow-query, forwarders, type, file

8. Zone file: Now edit vi /var/named/mydb-for-yoursenecaid-ops and enter the following (use your own value and domain where applicable).

```
$TTL   3D
@      IN     SOA    @  yoursenecaid.ops. (
                2022101201          ; Serial
                8H                  ; Refresh
                2H                  ; Retry
                1W                  ; Expire
                1D                  ; Negative Cache TTL
);
@      IN     NS     server.yoursenecaid.ops.
@      IN     NS     client.yoursenecaid.ops.
@       IN     A      192.168.0.10
server  IN     A      192.168.0.10
client  IN     A      192.168.0.20
hello   IN     A      32.32.32.32
jason   IN     A      11.11.11.11
ops     IN     A      22.22.22.22
```

9. Again, here's the reference documentation for records in this file. Specifically pay attention to:

   A records

   NS records

   SOA records

10. Now that your DNS server (bind, a.k.a. named) is configured:

    Start the named service with the systemctl command.

    systemctl start named

    Check that the named service is running using the ps ax command (perhaps combined with grep), and separately, the systemctl command (if necessary), or check the /var/log/messages file for troubleshooting purposes.

    ps ax | grep named

    systemctl status named

    Once you are certain that the named service had started and runs without errors, then set it to start automatically (i.e. enable the named service) when this virtual machine boots.

    systemctl enable named

11. Try to query few DNS resources: at this stage, you have to add 192.168.0.10 to specify which DNS server to use.

    nslookup jason.jasonpang.ops 192.168.0.10

    nslookup server.jasonpang.ops 192.168.0.10

nslookup jasonpang.ops 192.168.0.10

Now you know the service works, add few more records, restart the “named” service and check again.

systemctl restart named

12. Firewall rules update:

Remember that you are supposed to have a working firewall on your host (and every other machine), but we will focus on our host machine for now. A working firewall will block requests to ports that you didn't explicitly allow. This means that at this point, your DNS server, even though it's perfectly configured, is inaccessible to any other machine because iptables won't allow the requests to come in (the machine can communicate with itself using the rule that allows all traffic on the “lo” loopback interface).

Perform the Following Steps:

a. You may want to reboot your Server VM so that you get a clean iptables.
b. You need to find out on which IP and ports the named are listening. (TCP & UDP at 53 & 953)
   ss -nautp | grep "named"

```
[root@server ops345]# ss -nautp | grep "named"
udp    UNCONN     0      0      192.168.0.10:53                 *:*                   users:(("named",pid=2045,fd=514))
udp    UNCONN     0      0      10.0.0.4:53                     *:*                   users:(("named",pid=2045,fd=513))
udp    UNCONN     0      0      127.0.0.1:53                    *:*                   users:(("named",pid=2045,fd=512))
tcp    LISTEN     0      10     192.168.0.10:53                 *:*                   users:(("named",pid=2045,fd=23))
tcp    LISTEN     0      10     10.0.0.4:53                     *:*                   users:(("named",pid=2045,fd=22))
tcp    LISTEN     0      10     127.0.0.1:53                    *:*                   users:(("named",pid=2045,fd=21))
tcp    LISTEN     0      128    127.0.0.1:953                   *:*                   users:(("named",pid=2045,fd=24))
tcp    LISTEN     0      128       [::1]:953                 [::]:*                   users:(("named",pid=2045,fd=25))
```

c. You will need to update the firewall on host to allow incoming connections to port 53 for both UDP and TCP (i.e. the protocol and port that DNS uses). Port 953 is secured port for DNS that we will not configure in this course.
iptables -I INPUT -p tcp --dport 53 -j ACCEPT
iptables -I INPUT -p udp --dport 53 -j ACCEPT

NOTE: You could just disable the firewall but that is a poor workaround! You are expected to be able to handle configuration (such as this) at this point in this course. You may also want to apply the rules to the LAN network adapter only by add "-i eth1" to the rules in the future. (Not right now because we have more to test.)

Case 2: Configuring the DNS Client (on Client VM)

1. Install the **bind-utils** package to your client (vm2).

   Testing DNS server by perform the Following Steps from the Client vm2:

   Issue the following commands to verify that you have set-up your DNS server correctly on vm1 and you can resolve DNS for google and your host machine:

   You may have to add DNS server at the end of command at this stage.

   nslookup client.yoursenecaid.ops 192.168.0.10

   nslookup hello.yoursenecaid.ops 192.168.0.10

   nslookup google.ca 192.168.0.10

2. You will need to configure your VMs to use your Server VM's IPADDR as the default DNS server (DNS1) in your LAN adapter (e.g. ifcfg-eth1) file for your network interface card.

Remember we purposely omitted the following parameters in the ifcfg-eth1 in the previous Lab?

Add those back and restart the network service.

vi /etc/sysconfig/network-scripts/ifcfg-eth1

```
BOOTPROTO=static
IPADDR=192.168.0.20
NETMASK=255.255.255.0
DEVICE=eth1
ONBOOT=yes
IPV6INIT=no
NM_CONTROLLED=no
DNS1=192.168.0.10
DOMAIN=yoursenecaid.ops
```

systemctl restart network

3. You should be able to query the DNS resources without the appending DNS server and Domain.

```
[root@client network-scripts]# nslookup google.ca
Server:         192.168.0.10
Address:        192.168.0.10#53

Non-authoritative answer:
Name:   google.ca
Address: 142.250.68.67
Name:   google.ca
Address: 2607:f8b0:4007:817::2003

[root@client network-scripts]# nslookup jason
Server:         192.168.0.10
Address:        192.168.0.10#53

Name:   jason.jasonpang.ops
Address: 11.11.11.11

[root@client network-scripts]# nslookup jasonpang.ops
Server:         192.168.0.10
Address:        192.168.0.10#53

Name:   jasonpang.ops
Address: 192.168.0.10
```

Case 3: Configuring the DNS Server (vm1) For Reverse Lookups

As it stands now, your server will only handle forward resolution (converting names into IP addresses), but it is also sometimes necessary to convert addresses back into human readable names. For public IP addresses only, your ISP can provide this service for you. For private networks you can set it up for your own organization. But if you're setting up reverse DNS on the public internet - you need to understand what information the ISP would require.

1. Issue the following commands on any of your VMs:

   nslookup 192.168.0.10

   nslookup 192.168.0.20

2. Both commands should have failed because there is currently nothing that will handle this reverse lookup for you. This requires another zone on your server (vm1).

   Add the following entry to your /etc/named.conf on the DNS server (vm1):

```
zone "0.168.192.in-addr.arpa." {
        type master;
        file "mydb-for-192.168.0";
};
```

3. And create the following zone file on the DNS server (vm1):

```
$TTL    3D
@       IN      SOA     @       yoursenecaid.ops.(
                        2022101202      ; Serial
                        8H              ; Refresh
                        2H              ; Retry
                        1W              ; Expire
                        1D              ; Negative Cache TTL
);
@       IN      NS      server.yoursenecaid.ops.
10      IN      PTR     server.yoursenecaid.ops.
20      IN      PTR     client.yoursenecaid.ops.
66      IN      PTR     somethingnew.yoursenecaid.ops.
```

4. Refer to the reference documentation for PTR records.
5. Add extra records as needed to complete this zone.
6. Restart the service and test these records on both VMs to check the result.

   nslookup 192.168.0.66

   It would work for **vm2** but not **vm1**. Why?

   On **vm1** the only DNS server is Microsoft's DNS server. And 192.168 IP is not routable to MS.

```
[root@server ops345]# grep "nameserver" /etc/resolv.conf
nameserver 168.63.129.16
```

   (Optional) How do you make the command works with vm1 when query DNS from vm1?

Save the captured file(s) as OPS345_Lab02_yourusername and upload to Blackboard.

If it is video recordings, upload to OneDrive and share with jason.pang@senecacollege.ca