

Studio 5

In this studio, you will be using virtual machines (VMs) in ONL. To use them you will need to be using an RLI that is version 8.5 or greater. Version 8.5 was released just recently. It can be found by following the “Get RLI” link on the ONL website.

VMs in ONL are different than physical end hosts in many ways.

1. Your ONL passwords are not used on the VMs. VMs are created each time you commit an experiment and a password is created for your VMs each time you commit. The same password is used on each VM. You find the password from the RLI *Topology* menu by selecting the *Show VM Password* menu item.
2. You will have root privileges via the *sudo* command. When you use *sudo* for the first time you will be asked to enter your password. This is the VM password. After you enter your password correctly you will be authorized to use *sudo* for about 15 minutes without entering your password each time. After that 15 minute period, when you try to use *sudo* again, it will ask for your password again.
3. The VM host names are different than the physical hostnames. For your experiment you will see host labels on your RLI topology of *VMsmall.1* through *VMsmall.7*. When you use the *source ~/.topology* command it will give you access to names of the form *VMsmall1* through *VMsmall7*. Note the missing ‘.’ in the latter names!!! So, if you are on *onlusr* and you want to log in to your *VMsmall.3* VM you would do this:

```
onlusr> source ~/.topology
onlusr> ssh $VMsmall3
jdd@vm12c02v01's password:
```

You are then prompted for the VM password. The *vm12c02v01* is the identification of the actual virtual machine you have been allocated.

4. Your home directories are NOT accessible on the VMs. Any work you do on the VMs that you want to save you will either have to use *sftp* (from *onlusr* to the VM) or do a text copy and paste.

If you wanted to *sftp* the file *cmd_results.txt* from your VM home directory on *VMsmall.3* you would do this from *onlusr*:

```
onlusr> source ~/.topology
onlusr> sftp $VMsmall3
jdd@vm12c02v01's password:
connect to vm12c02v01.
sftp> get cmd_results.txt
```

```
sftp> quit
onlusr>
```

And the file would now be in your directory on onlusr.

Please remember that once you *Close* your experiment, your VMs are destroyed and anything you did on them can not be recovered.

Now, on to the studio exercises. These are to help you get acquainted with a subset of the commands you will need to use in Lab5.

1. Load the ONL topology provided, make a reservation and commit.
2. Once the commit has completed, log in *onlusr* and then to your host *h3x2*.

```
onlusr> source ~/.topology
onlusr> ssh $h3x2
```

Once you are logged in to h3x2 we will investigate addresses, ARP and routes. We will use the *ifconfig* command to look at our data interface. We will use the *arp* command to investigate ARP and the *route* command to investigate routes. Read about them first:

```
h3x2> man ifconfig
h3x2> man arp
h3x2> man route
```

Read about the commands and then figure out how to display the current state of the interface, the arp table and the route table on h3x2.

What is the IP address of the data0 interface on h3x2?

Are there any arp entries associated with interface data0?

Are there any route entries associated with interface data0?

Try looking at the route entries with and without a *-n* option to the *route* command.

What is the gateway that is used for the route to 192.168.1.0/24?

What is the gateway that is used for the route to 192.168.4.0/24?

Where is that gateway located? Can you verify its location and address in the RLI?

3. Now, log in to your VM VMsmall.1

```
onlusr> source ~/.topology
onlusr> ssh $VMsmall1
```

Once you are logged in to VMsmall.1 investigate addresses, ARP and routes there. On the VM the data interface will be data1.

What is the IP address of the data1 interface on VMsmall.1?

Are there any arp entries associated with interface data1?

Are there any route entries associated with interface data1?

Now, ping VMsmall.2. VMsmall.2 has an address on its data1 of 192.168.1.3. To verify that you could log in to VMsmall.2 and use *ifconfig* on data1.

Now has there been any change in the arp entries associated with interface data1?

On the *man* page for *arp* can you find how to display JUST the arp entries associated with interface data1?

4. Now, with one window logged in to h3x2 and one logged in to VMsmall.1 let's test if the two can communicate.

Look again at the arp and route entries on VMsmall.1.

Also remind yourself what the IP address of the data1 interface of VMsmall.1 is.

Then from h3x2 try pinging the IP address of VMsmall.1.

Does the ping succeed?

It shouldn't.

Why not? Look at the route entries on each. One end is missing something.

Now look at the arp entries on VMsmall.1 again. Has anything changed?

How do you suppose that happened?

Because there is a route on h3x2, the ping requests get to VMsmall.1 but there is no return route on VMsmall.1 to send back a reply. But, in the process of VMsmall.1 getting the ping request the router port used *arp* to connect to VMsmall.1 and thus VMsmall.1 has an entry for its gateway router.

Now, a little background information. The RLI has caused the generation of 'default routes' and that is why we see the routes on h3x2 going to all the other subnets. But the RLI does not set routes on the VMs. That is left to the VM user to do. VMs in ONL are set up for total control by the user.

To set up a route on VMsmall.1 to get back to h3x2 we need to understand some things about routes on Linux.

First we need to know about 'net' routes and 'host' routes. Use the *man* command again to read about the *route* command and search for '-net' and '-host'. To search while in the *man* command you can type the command */-net* followed by a *return* and that will search for '-net'.

Second we need to know about gateways. To read about them search for 'gw' while running *man route*.

Third we need to know about targets and masks. The target is the network or host we are trying to send packets to.

There are also routes that do not use a gateway but just direct packets to a particular interface.

Thoroughly read about routes in the *man* page.

After you have learned about all these aspects of Linux routes, see if you can use the *route* command to set up a route on VMsmall.1 to get to h3x2.

What would VMsmall.1 use as a gateway to get to h3x2? Do you first need a route to the gateway or will VMsmall.1 be able to find it?

What kind of route (-net or -host) would you use?

What kind of route (-net or -host) would you use if you want to define a single route on VMsmall.1 to get to h4x2 and/or h4x3?

Is there a single route you could add on VMsmall.1 that would allow it to send to h3x2, h4x2 and h4x3?