University Göttingen
GWDG/AG-C
Hendrik Nolte

Exercise 1 / Feburary 20, 2023
HPC System Administration / WiSe 2022/23
15 Minutes Total

Exercise Introduction

This exercise will guide you through the manual configuration of the **pxe-network** on the cluster-manager node.
You should have completed the preparation exercise sheet and have a running shell on the cluster-manager.
This preparation is required for later exercises.

Contents

Task 1: Configure the network (15 min)

If you have followed the setup plan on the preparation sheet, your network topology should look similar to the one shown in Figure 1.
You can find your network topology on OpenStack on the left under **Network** and then **Network Topology.**.

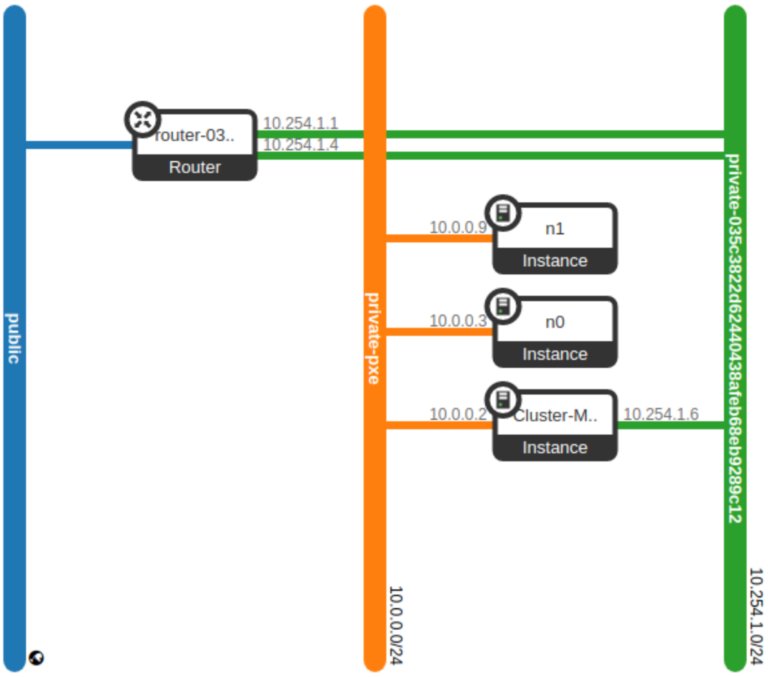
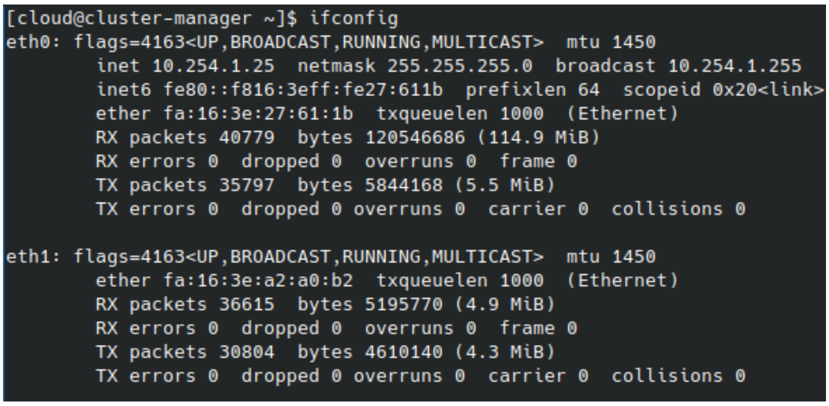


Figure 1: Network topology of our small cluster

However, if you connect via `ssh` to your cluster manager and run
`$ ifconfig`
you get an output with multiple blocks starting with **eth0**, **eth1** and more.
Only the **eth0** and **eth1** blocks are relevant now as shown in Figure 2.



```
[cloud@cluster-manager ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 10.254.1.25  netmask 255.255.255.0  broadcast 10.254.1.255
        inet6 fe80::f816:3eff:fe27:611b  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:27:61:1b  txqueuelen 1000  (Ethernet)
        RX packets 40779  bytes 120546686 (114.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35797  bytes 5844168 (5.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        ether fa:16:3e:a2:a0:b2  txqueuelen 1000  (Ethernet)
        RX packets 36615  bytes 5195770 (4.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30804  bytes 4610140 (4.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 2: Ifconfig output before the configuration

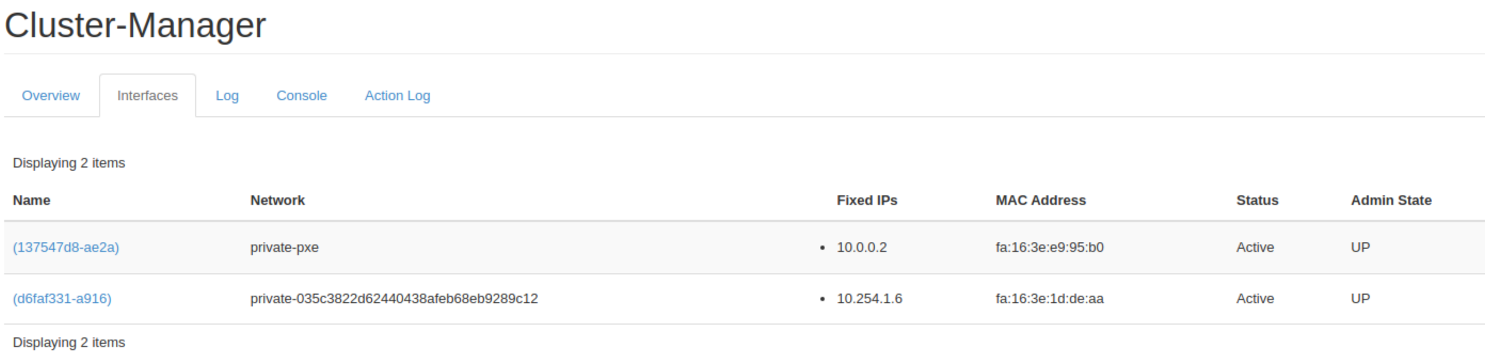
eth0 is the **private** network that you added when launching the node.
This network should show a value **inet** in the second row with an IP address (10.254.1.25 in Figure 2).
You can verify that your address matches the address shown on OpenStack under **Compute** and **Instances** for the **private** network.

The issue now is that, when looking at **eth1**, this **inet** value is completely missing even though an IP address was assigned on the **private-pxe** network for this node.
This can also be seen in the **Network Topology** as it shows one IP address each on the branches that go the two networks (see Figure 1).

Configuring the Network Interface

In order to complete the configuration, you need to find the IP address and the MAC address that were assigned by OpenStack for the cluster-manager on the **private-pxe** network.
On OpenStack under **Compute** and **Interfaces** click on the name of the **cluster-manager** to see details of this node. Then navigate from the **Overview** tab to the **Interfaces** tab.

Here you should see the two networks similar to Figure 3.
Identify the row of the **private-pxe** network and make a note of the **Fixed IPs** and the **MAC Address** values of it.
You will use these values to configure **eth1** as the interface for the **private-pxe** network on the cluster-manager.



Displaying 2 items

Name	Network	Fixed IPs	MAC Address	Status	Admin State
(137547d8-ae2a)	private-pxe	• 10.0.0.2	fa:16:3e:e9:95:b0	Active	UP
(d6faf331-a916)	private-035c3822d62440438afeb68eb9289c12	• 10.254.1.6	fa:16:3e:1d:de:aa	Active	UP

Displaying 2 items

Figure 3: If you click on `View-Instance-Details`. Go to `Interfaces`.

Now:

- `ssh` into your cluster manager

- Run `$ sudo -i`
- Install **nano** via **yum install nano** and confirm with **y**
- Check that there is no file **ifcfg-eth1** with:
  `$ ls /etc/sysconfig/network-scripts/`
- You should see a few other **ifcfg** files but not **ifcfg-eth1**
- Only if **ifcfg-eth1** does not exist, it is safe to continue!
- `$ cd /etc/sysconfig/network-scripts/`
- `$ nano ifcfg-eth1`
- Fill in as shown in Listing 1 **Fill in your own values for MAC Address and IP Address!**

Listing 1: ifcfg-eth1

```
BOOTPROTO=none
DEVICE=eth1
HWADDR=YOUR_MAC_ADDRESS
IPADDR=YOUR_IP_ADDRESS
MTU=1450
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
```

Fill out HWADDR and IPADDR with your own values!
Finally, reboot the node by running `$ reboot`

Verifying the Configuration

Wait a moment for the machine to reboot and then reconnect with SSH.
To check if the configuration type
`$ ifconfig`
and ensure that the **eth1** block now has a value for **inet** that matches the IP address you entered earlier.
Your **eth1** block should look similar to Listing 2

Listing 2: ifconfig after the configuration

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450
        inet 10.0.0.12 netmask 255.255.255.0 broadcast 10.0.0.255
        inet6 fe80::f816:3eff:fe08:e249 prefixlen 64 scopeid 0x20<link>
        ether fa:16:3e:08:e2:49 txqueuelen 1000 (Ethernet)
        RX packets 1017025 bytes 68271356 (65.1 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1093389 bytes 2159495876 (2.0 GiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

For the last part, check that your network routes correctly. For this type
`$ route -n`
The output should be identical to Listing 3.
Especially the line with **eth1** in the **Iface** column should match the one shown here.

Listing 3: route -n output

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.254.1.1 0.0.0.0 UG 100 0 0 eth0
10.0.0.0 0.0.0.0 255.255.255.0 U 101 0 0 eth1
10.254.1.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0
169.254.169.254 10.254.1.1 255.255.255.255 UGH 100 0 0 eth0
```

Important is line 4.

If there are any issues or the verification did not work, go back and check that you made no mistake when configuring **ifcfg-eth1**. If this also did not help, feel free ask for help from us.