Linux Academy

Live! Lab

Configure IPv6 Addresses

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Username and password information is stored on the Live! Lab page.
- Labs will expire after a defined amount of time.

Related Courses

Red Hat Certified Engineer 7

Related Videos

Configure IPv6 Addresses and Perform Basic IPv6 Troubleshooting

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

Using the NetworkManager command line interface, users can manually set and test both IPv6 and IPv4 connections. This lab makes use of the *eth1* interface available within the lab environment.

Preparing the Network Manager

Your lab begins with the *NetworkManager* utility masked; this will differ in actual production situations. We need to unmask and enable the service:

```
[root@red-hat ~]# systemctl unmask NetworkManager
[root@red-hat ~]# systemctl enable NetworkManager
```

We now need to ensure that NetworkManager manages *eth1*, the network card we are using in this instance. Navigate to `/etc/sysconfig/network-scripts`, and open `ifcfg-eth1` to edit the file. Change the line `NM_CONTROLLED=no` to `NM_CONTROLLED=yes`.

With our settings appropriate, we can start NetworkManager:

```
[root@red-hat ~]# systemctl start NetworkManager
```

If you started NetworkManager prior to this step, restart it.

Configuring IP Addresses

Review `ifconfig`. The eth0 network card is the one used to connect to the lab servers; we are instead working with eth1. Run `mncli con show` to view the available connections. We now need to create an additional connection that `nmcli` can manage.

We want to create a connection to work with the eth1 device:

```
[root@red-hat ~]# nmcli con add con-name eth1 type ethernet ifname eth1
Connection ‘eth1’ (1e15dba9-d91d-4784-80e3-b809e50baf17) successfully added.
```

If you run `nmcli con show` you can see the created connection.

We can now start configuring it with addresses. To create an IPv4 address:

```
[root@red-hat ~]# nmcli con mod eth1 ipv4.addresses 192.168.10.100/24
```

We also want to tell the system we are managing this configuration manually:

```
[root@red-hat ~]# nmcli con mod eth1 ipv4.method manual
```

Now to assign an IPv6 address:

```
[root@red-hat ~]# nmcli con mod eth1 ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64
[root@red-hat ~]# nmcli con mod eth1 ipv6.method manual
```

Should we run `nmcli con up eth1`, followed by `nmcli con show` we can see that the eth1 addresses have changed to the ones we manually set.

However, to ensure that the connection is configured properly, we want to check it using `ping`:

```
[root@red-hat ~]# ping -I eth1 192.168.10.100
[root@red-hat ~]# ping6 -I eth1 fddb:fe2a:ab1e::c0a8:64
```

With the connections confirmed, everything is set and the lab is complete.