**Configure IPv6 Addresses and Basic Troubleshooting In Linux**

In the past many system administrators have simply resorted to disabling IPv6 rather than properly configuring it, continuing to rely on the older IPv4 which has worked just fine for a very long time. As the IPv4 address space has since become exhausted, administrators are starting to slowly take up IPv6 out of necessity.

Here we’re going to cover how to configure IPv6 addressing in Linux and provide some basic tips and advice for troubleshooting IPv6 network issues.

**Configuring IPv6 Addresses**

IPv6 addresses can be configured in a few additional ways when compared with IPv4, some of these are listed below.

1. **Manual Configuration:** This is fairly similar to the way IPv4 is manually configured, essentially we manually edit an interface file in the /etc/sysconfig/network-scripts/ifcfg-<interface> format.
2. **DHCPv6:** Dynamic Host Protocol Version 6 is similar to DHCP for IPv4 in that it will automatically configure our interface.
3. **Stateless Address Autoconfiguration (SLAAC):** This works similar to DHCP, however it works by receiving router advertisement messages from a local IPv6 router on the network.

Here we will primarily be focusing on manual IPv6 network configuration.

**Manual IPv6 Configuration**

First take a look in the /etc/sysconfig/network-scripts/ directory to see if there is already existing IPv6 configuration for the particular interface in question. The file name will be listed as ifcfg-<interface>, you can confirm the interface names by running ‘ip a’ or the deprecated ‘ifconfig’. Typical names may include eth0, or the newer ‘eno\*’ format such as eno16777736. This file can contain both IPv4 and IPv6 configuration for the same interface.

Below is the configuration of my CentOS 7 /etc/sysconfig/network-scripts/ifcfg-eno16777736 file, note the settings that start with “IPV6”.

* **IPV6INIT=yes** – This is needed when configuring IPv6 on the interface.
* **IPV6ADDR=<ipv6-address>** – Specifies a primary static IPv6 address.
* **IPV6\_DEFAULTGW=<ipv6-address>%eno16777736** – Adds a default route through the interface specified.

Note that if you manually edit these files, you’ll need to run ‘nmcli con reload’ to pickup the changes. Alternatively we can make changes with the nmcli command, which admitedly takes some getting used to, but is quite powerful and its tab autocomplete helps a lot.

nmcli con mod eno16777736 ipv6.addresses 'fe80::20c:29ff:fe27:f2b6/64'

nmcli con mod eno16777736 ipv6.method manual

The first command sets the IPv6 address, while the second ensures that this is a static address and is not lost from DHCP or SLAAC.

For further information, keep the ‘nmcli-examples’ man page in mind, as there are a number of different examples documented here that you can make use of.

man nmcli-examples

**Automatic IPv6 Configuration**

Automatic configuration of IPv6 addresses can take place with DHCPv6 or SLAAC. Below is some default configuration from CentOS 7 which has IPV6\_AUTOCONF enabled, this configures network settings using SLAAC router advertisements.

[root@centos7 network-scripts]# **cat ifcfg-eno16777736**

HWADDR=00:0C:29:AB:12:34

TYPE=Ethernet

BOOTPROTO=dhcp

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

NAME=eno16777736

UUID=0dbee9e5-2e7e-4c88-822b-869cfc9e2d54

ONBOOT=yes

IPV6\_PEERDNS=yes

IPV6\_PEERROUTES=yes

Alternatively we can set IPV6\_AUTOCONF to no, and define DHCPV6C=yes to use DHCPv6 rather than SLAAC. When using SLAAC or DHCPv6, the manual configuration items such as IPV6ADDR and IPV6\_DEFAULTGW can be removed as these will be automatically configured.

For further information on these variables you can check out the documentation in /usr/share/doc/initscripts-\*/sysconfig.txt

**Applying Network Changes**

After making any changes to the files in /etc/sysconfig/network-scripts/ the network needs to be restarted for them to take effect.

systemctl restart network

You could also perform a system reboot however this will take longer, but will bring the interface up with the new configuration.

**Basic IPv6 Troubleshooting**

Here are some basic tools you can use to perform basic IPv6 troubleshooting, they work fairly similarly to their IPv4 counterparts.

**IPv6 Ping**

The ping6 command works in the same way as the normal ping command, except that ping6 works with IPv6 addresses. This can be used to send ICMP traffic to an IPv6 address and check for reply, the below example pings the IPv6 localhost address.

[root@centos7 ~]# **ping6 ::1**

PING ::1(::1) 56 data bytes

64 bytes from ::1: icmp\_seq=1 ttl=64 time=0.039 ms

...

We can also ping out of a specified interface, the below command pings everything connected to eno1677736.

[root@ ~]# **ping6 ff02::1%eno16777736**

PING ff02::1%eno16777736(ff02::1) 56 data bytes

64 bytes from fe80::204:edff:feef:a770: icmp\_seq=1 ttl=64 time=2.32 ms

64 bytes from fe80::204:edff:feef:a770: icmp\_seq=2 ttl=64 time=0.640 ms

**IPv6 Routes**

The ‘ip -6 route show’ command can be used to display IPv6 routing.

[root@centos7 ~]# **ip -6 route show**

unreachable ::/96 dev lo metric 1024 error -101

unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -101

unreachable 2002:a00::/24 dev lo metric 1024 error -101

unreachable 2002:7f00::/24 dev lo metric 1024 error -101

unreachable 2002:a9fe::/32 dev lo metric 1024 error -101

unreachable 2002:ac10::/28 dev lo metric 1024 error -101

unreachable 2002:c0a8::/32 dev lo metric 1024 error -101

unreachable 2002:e000::/19 dev lo metric 1024 error -101

unreachable 3ffe:ffff::/32 dev lo metric 1024 error -101

fe80::/64 dev eth0 proto kernel metric 256

**IPv6 Traceroute**

The traceroute6 command can be used to show all hops in the path to the specified destination in the same way as the traceroute command.

[root@centos7 ~]# **traceroute6 ipv6.google.com**

traceroute to ipv6.google.com (2607:f8b0:4006:80f::200e), 30 hops max, 80 byte packets

connect: Network is unreachable

The tracepath6 command works in a similar manner.

**View Current IPv6 Address Information**

You can display current IPv6 configuration by running the ‘ip a’ command which is short for ‘ip address show’. This will show IPv6 interfaces as ‘inet6’ followed by their IP address.

**View IPv6 Network Connections**

By default the ‘netstat’ command shows both IPv4 and IPv6, however with the -6 option we can select to only display the ports that the server is listening on with its IPv6 addresses as well as any connections.

[root@centos7 ~]# **netstat -antup6**

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp6 0 0 :::111 :::\* LISTEN 787/rpcbind

tcp6 0 0 :::22 :::\* LISTEN 1661/sshd

tcp6 0 0 ::1:631 :::\* LISTEN 2476/cupsd

tcp6 0 0 ::1:25 :::\* LISTEN 2066/master

udp6 0 0 :::111 :::\* 787/rpcbind

udp6 0 0 :::123 :::\* 781/chronyd

udp6 0 0 ::1:323 :::\* 781/chronyd

udp6 0 0 :::39836 :::\* 1439/dhclient

udp6 0 0 :::942 :::\* 787/rpcbind

**Conclusion**

Configuring IPv6 addresses is not that much different to configuring IPv4 addresses, the main difference is simply the structure of the IP addresses is different. Once you pick up the basics of IPv6 you can easily configure IPv6 networking. Most of the troubleshooting tools are also pretty similar, you can still view the current IP address information, routing tables, and perform pings and traceroutes in a very similar manner