12.3.3 Configure IPv6 Addresses

Click one of the buttons to take you to that part of the video.

Configure IPv6 Addresses 0:00-0:06

In this demonstration we're going to discuss how to configure

Configure IPv6 Addresses from the Command Line 0:07-0:40

IPv6 addresses on a Linux host. We'll first learn how to do it from the command line. Then we'll talk about how to do this in

Configure IPv6 Addresses from an Interface Configuration File

an interface configuration file. Before we begin, let's take a look at what our current configuration is. I'm going to use the 'su -' command to switch to my root user account, and I'm going to run the 'ifconfig' command. Notice when I do, I already have an IPv6 address assigned. Notice over here that this is a link-local IP address. On most Linux distributions, you're going to see that a link-local IPv6 address is configured by default.

Use Link-local Addresses 0:41-2:06

Link-local addresses are valid only on the current subnet, and they always begin with fe80. There's a couple of things you need to understand about these link-local addresses. They're not used for much. Really, all IPv6 hosts need to have at least one link-local address, and you can add additional IPv6 addresses later.

That's because IPv6 routers will never forward IPv6 packets that are addressed to link-local addresses. In other words, the traffic has to stay on the same subnet.

They're really only used for a couple of things. They're used for automatic address configuration, link-local addresses are used for Neighbor Discovery, and they can be used for host-to-host addressing if you're on a network that doesn't have any routers. Basically, a link-local address is kind of the equivalent of an APIPA address in IPv4. They're there for basic connectivity, but they don't do a whole lot.

I've got a Fedora system here. Let's run the 'ifconfig' command on this one as well. You see that it also has a link-local IPv6 address assigned as well. Using these link-local addresses, this host could ping the other host over the network, but in any network of appreciable size, link-local addresses just don't cut it. We need to add routable IPv6 addresses to the network interfaces in these two systems.

Add a Unique Local IPv6 2:07-2:39

For this demonstration we're going to add unique local IPv6 addresses to both interfaces. Unique local IPv6 addresses are really the equivalent of private IP addresses with IPv4. They're designed for communications within a given site, so they are routable within your organization.

However, just like an IPv4 private IP address, they're not routable on the internet. If we want to connect the network that uses unique local addressing to the internet, we would have to implement network address translation somewhere along the line.

Implement Network Address Translation 2:40-4:45

Let's do this from the command line first. An easy way to do this is to use the ifconfig command. We enter 'ifconfig' and then the name of the Ethernet interface in the system. I'm going to hit Enter here because I believe--yep, it's ens192. Let's do 'ifconfig ens192', and then we enter 'add' to specify that we're adding a new IP address through the interface, and then just the IPv6 address we want to add.

Add Unique Local IPv6 Addresses to an Interface

I'm going to add a unique local IPv6 address to this interface. Unique local IPv6 addresses always begin with 'fc00', and then I put '::', then the number '2', and then '/64'. Remember that any time you see :: within an IPv6 address, it indicates that everything from here to here in that very long IPv6 address is just zeros, so we're not going to bother typing out all those zeros. We'll just put ::, and we'll assign a host address of 2. We're going to use a 64-bit prefix with this address. Enter.

Now if we do 'ifconfig', we should see two IPv6 addresses assigned. Okay, we do. Here's the original link-local address, but now up here notice we have a unique local address assigned. fc00::2 with a 64-bit prefix. Let's do the same thing on the other system and then we'll test communications between them.

As you can see on this system, the name of the interface is ens32, but I'm going to run 'ifconfig ens32 add fc00::1/64'. Just like with IPv4 addressing, you have to have unique host address for every host on an IPv6 network. The first 64 bits of the address are the network portion of the address and the last 64 bits are the host portion of the address. Enter, run 'ifconfig' again, and just as with the Fedora system, we now have two IPv6 addresses assigned. We have our original link-local address, and now we have a unique local address assigned, too.

Test Communications 4:46-9:14

With this configuration in place, let's test communications over IPv6 between these two hosts. To do this, we type 'ping6' to send IPv6 ping packets between the hosts, and we'll ping the IPv6 address of the Fedora system: 'fc00::2', Enter, and communications are working. We're able to ping the Fedora system and we're getting responses over the IPv6 protocol. Excellent.

Make Address Assignments Persistent across Reboots

There's one problem with our current configuration, and that is the fact that because we used the ifconfig command to add the IPv6 unique local address to the interface, it's not persistent. It'll work until I shut the system down or until I disable and then re-enable the network interface. When I do, this address assignment that we made will be gone.

So we want to make these address assignments persistent across reboots by adding the necessary directives to each interface's configuration file.

To do it on this openSUSE system, I go into the '/etc/sysconfig/network' directory, run the 'ls' command. We see that there is a configuration file for the ens2 interface right here. Let's go ahead and open it in the vi editor, press the Insert key.

Now here's where things get a little bit confusing, because the way you add IPv6 addresses to the interface configuration file will vary between distributions, and you'll see that here. The way I add a persistent IPv6 address to this openSUSE system is going to be completely different than the way I do it on the Fedora system.

On this system I'm going to type 'IPADDR_0', meaning I'm adding an additional IP address to the interface '=' and then in tick marks we want to enter the IPv6 address that we want to assign--'fc00::1/64', the same address we assigned with the ifconfig command. Hit Escape, save our changes to the file, Exit the editor.

To do this, I'm going to take the ens32 interface down. 'ifdown ens32', and then let's bring it back up, 'ifup ens32'. Okay, the interface is back up. Let's type the 'ifconfig' command again, and we see that the IPv6 address that we assigned in the interface configuration file has been applied to the interface. It's persistent. If I reboot the system, that IP address is still going to be there. Let's go do the same thing on the Fedora system now.

On a Fedora system we need to go into the '/etc/sysconfig/networking-scripts' directory, which is a little bit different than on the openSUSE system, do an 'ls' command here. We should see an interface configuration file for the ens192 interface. That's the one we want, so I'll enter 'vi ifcfg-ens192'. You'll notice that the syntax used in this file is quite a bit different than the syntax that was used in the equivalent file on the openSUSE system. I'm going to press the Insert key. There are a couple of changes we need to make.

First of all, we need to set IPV6_AUTOCONF to 'no', and we do need to make sure that IPV6INIT is set to yes, and it is by default, which is fine. Most distributions will have IPv6 enabled by default, and so this one does.

We want to turn off IPv6 autoconfiguration and we actually want to remove these 2 lines right here, go back in command mode. We want to insert a new line now called 'IPV6ADDR=fc00::' host address '2/64' bit prefix, 'exit', save the changes to the file.

As with the interface on the openSUSE system, I have to take the ens192 interface on this Fedora system down and bring it back up so it will re-read the configuration file. We do 'ifdown ens192', and then we do an 'ifup'. Now let's type the 'ifconfig' command, and as with the openSUSE system, we now have two IPv6 addresses assigned to this interface. One is a unique local IPv6 address and the other is a link-local IPv6 address.

Summary 9:15-9:25

That's it for this demonstration. In this demo we talked about assigning IPv6 addresses to a Linux host. We first talked about how to use the ifconfig command to assign an IPv6 address and then we looked at assigning an IPv6 address using an interface configuration file.