Configure IPv4 & IPv6 Networking & Hostname Resolution

To be able to communicate across networks, a device needs an IP address.

IP = Internet Protocol

There are 2 kinds of IPs:

IPv4 & IPv6

Some examples of IPv4 Addresses:

10.0.0.1, 192.168.0.5, 203.0.113.9

IPs, basically 4 numbers that can take 0 – 255.

They use these specific values because internally, computers use 8 bits for each number. And 8 bits can represent a number from 0 to 255.

00000000 is binary 0 in decimal notation

00000001 is 1

00000010 is 2

00000011 is 3

00000100 is 4

And so on, all the way up to 11111111 is 255

192.168.1.101 = 11000000.10101000.00000001.01100101 in binary

Sometimes, we will see IPs displayed in this format:

192.168.1.101/24 = This is called CIDR notation (Classless Inter-Domain Routing)

/24 tells us that the first 24 bits of this address are the prefix of this network.

11000000.10101000.00000001.01100101

192 168 1 101

{ first 24 bits } {Last 8 bits}

This illustrates that 192.168.1 is the network prefix. Basically, the address of the network itself.

.101 is the specific device on that network.

It is somewhat like phone numbers, think of the country code when you are calling internationally, Japan’s country code is +81, so that +81 can be seen as the country’s prefix, while the phone number is the specific device.

+81 80 1365 8745

+81 = Country prefix

80 1365 8745 = Personal number

So, a network prefix is analogous to a country prefix in phone numbers.

If a device knows that the CIDR notation of a network IP is 192.168.1.101

192.168.1 being the network prefix while

.101 is the specific host

This also means that any IP from 192.168.1.0 to 192.168.1.255 is part of the same network.

But an IP like 192.168.2.255 is part of a different network because the prefix does not have the same initial 3 numbers as 192.168.1, or if we think in binary, it does not have the same 24 bits that the /24 CIDR notation indicates that it should have.

Now if the CIDR notation is different like this:

192.168.1.101/16 = 192.168 = network prefix | 1.101 = host

11000000.10101000.00000001.01100101/16

192 168 1 101

{ First 16 bits } { Last 16 bits }

For this current CIDR notation, an IP in this network must begin with 192.168, while the last 2 numbers can be anything from 0 – 255.

So, we can have IPs from 192.168.0.0 all the way to 192.168.255.255.

IPv6

IPv6 = has 128-bit addresses.

An example of an IPv6 Address:

2001:0db8:0000:0000:0000:ff00:0042:8329

There are 8 groups of numbers

They are in hexadecimal format and there are 16 hexadecimal digits:  
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E and F.  
In this case, A is equivalent to the number 10 in decimal format that we use every day. B is 11, C is 12, all the way to F which is 15.  
A number like “ffff” is equivalent to 65535 in decimal notation, and 1111111111111111 in binary.

Each number is separated by a “:”.

Since these types of address are long, we usually see them shortened:

2001:0db8:0000:0000:0000:ff00:0042:8329

Down to:

2001:db8::ff00:0042:8329 = The zeroes have been removed from 0db8 and the zeroes all the way up to :ff00.

IPv6 also supports CIDR notation:

2001:0db8:0000:0000:0000:ff00:0042:8329/64

/64 = is saying that the first 64 bits represent the network prefix

Each group of two hexadecimal digits is 8 bits.

2001:0db8:0000:0000:0000:ff00:0042:8329/64  
1 2 3 4 5 6 7 8

{ First 64 Bits }

Network Prefix

The ip comand can help us discover almost everything we need to know about our network configuration.

~$: ip link – This command shows the network devices (AKA network interfaces) on the Linux system.

When looking at the output, you can see some real devices like NICs, and some are virtual like the lo interface - loopback interface/address.

~$: ip –c addr – Displays the IP address(s) configured for these interface(s). The –c flag adds color to the important parts of the output.

~$: sudo ip link set dev <interface name> up – This brings a network interface from a DOWN state to an UP state (thus acquiring an IP address).

How to manually add an IPv4 address to an interface:

~$: sudo ip addr add <IP><CIDR> dev <network interface>

How to manually add an IPv6 address to an interface

~$: sudo ip addr add <SHORTENED IPv6><CIDR> dev <network interface>

~$: sudo ip addr add fe80::5054:ff:fe1f:8050/64 dev enp0s8 (EXAMPLE)

Note that a network interface can have multiple IP addresses.

How to manually delete an IP address (IPv4 & 6)

~$: sudo ip addr delete <IP><CIDR> dev <network interface>

Changes to IP address using the IP command are temporary. When the server/system is rebooted, the changes are lost. So, the IP command is used to test new IP settings temporarily.

How do I make the changes permanent?

It depends on the OS. Each one has its own set of tools. This can sometimes change from one major version to the next.

Ubuntu currently uses a utility known as “Netplan”.  
This is a set of utilities that reads some config files, then sends instructions to other networking tools telling them to configure network settings accordingly.

On the ubuntu server edition, Netplan sends instructions to Systemd’s network daemon called systemd-networkd.

How to see current Netplan config files:

~$: sudo netplan get

By setting DHCP to true, it makes this device ask for network settings from a DHCP4 server (the 4 refers to IPv4 settings)  
A DHCP server is often used to auto configure IPs for the devices on a network.

A given device on a network asks a DHCP daemon running on the router to tell it what IP settings it should configure for itself.

We can manually configure netplan settings by editing the yaml file in the /etc/netplan directory.