Multiple IP Addresses on Single Interface

Nov 2023

Introduction

There are occasions where we might want to assign multiple network addresses to a single interface on a server. This might be because you want to talk to another network where no routing exists, or you might want to assign a second address to a server where the main IP Address cannot be changed (maybe changing the primary management address will break the unit).

This article is written with the assumption that you are using on prem hardware, not sure if this has any suitability in cloud environments.

There are a couple of methods we can discuss here, there is what I would call a quick and dirty method, that has some limitations, and there is a better method of doing this, it takes a little more work, but it worth the effort in the long run.

Quick and Dirty Method

If you just need a second IP Address because the existing one is not suitable (and cannot be changed because maybe it will break the software) then you can use this method.

Lets look at an existing configuration for the first interface, which is normally the management interface on eno1 (or eth0 for CentOS if you have used the MediaKind ISO for OS install).

In Alma/CentOS/RedHat the interface configuration files are stored in /etc/sysconfig/network-scripts. In this example there are two interfaces, eno1 and eno2. eno1 is for management and eno2 is for data.

If we look at the interface configuration, we can see the IP Addresses available on each interface.

```
ip -br -c a

lo              UNKNOWN         127.0.0.1/8
eno1            UP              192.168.1.223/24
eno2            UP              11.0.223.2/16
```

We can see that the interface eno1 currently has a single IP Address of 192.168.1.233. However, lets assume that we want to add an additional IP Address of 10.32.13.233.

Here is the current configuration of eno1 where we only have a single IP Address of this interface.

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=eno1
UUID=ca90417f-9df9-4f34-bae5-77aa6531cc00
DEVICE=eno1
ONBOOT=yes
IPADDR=192.168.1.223
PREFIX=24
GATEWAY=192.168.1.1
PREFIX1=24
DNS=8.8.8.8
```

We can edit this file and simply add another address to this interface, if we edit this file we can add these lines to the ifcfg-eno1 file:

```
IPADDR1=10.32.13.223
PREFIX1=24
DNS1=8.8.4.4 (this line is not required, it just shows we can add another DNS setting.
```

Below is the edited ifcfg-eno1 configuration file with the additional lines (in Blue).

TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=eno1
UUID=ca90417f-9df9-4f34-bae5-77aa6531cc00
DEVICE=eno1
ONBOOT=yes
IPADDR=192.168.1.223
PREFIX=24
GATEWAY=192.168.1.1
PREFIX1=24
DNS=8.8.8.8
IPADDR1=10.32.13.223
PREFIX1=24
DNS1=8.8.4.4

If we look at the interface details in the OS now we can see the following IP Addresses.

```
  eno1              UP                192.168.1.223/24 10.32.13.223/24
```

We can see that the interface has two IP Addresses. The big issue with this method is the Gateway, the first IP Address has the gateway (192.168.1.1) where as the 10.32.13.233 has no gateway. We could change the gateway to accommodate the new additional IP, but I don't believe you can have two gateways. I have not tried to add a second gateway, but I believe that method two would be preferential as this is really a quick method.

The Better Method

The better method requires using multiple configuration files for a single interface. This method has a couple of advantages, it simplifies adding addresses and allows for multiple gateways to be used.

In the directory /etc/sysconfig/network-scripts we can see in this example we two interfaces:

```
-rw-r--r--. 1 root root 255 Nov 13 17:01 ifcfg-eno1
-rw-r--r--. 1 root root 232 Nov 13 17:00 ifcfg-eno2
```

ifcfg-eno1 is the interface that we want to apply multiple IP Addresses to. Firstly we will copy this interface file and edit each file accordingly. When creating the new interface configuration files we will add a number to the end of each file:

cp ifcfg-eno1 ifcfg-eno1:0
cp ifcfg-eno1 ifcfg-eno1:1

Now we have three configuration files for interface eno1.

-rw-r–r–. 1 root root 255 Nov 13 17:01 ifcfg-eno1
-rw-r–r–. 1 root root 255 Nov 13 17:04 ifcfg-eno1:0 (Alias 1)
-rw-r–r–. 1 root root 255 Nov 13 17:06 ifcfg-eno1:1 (Alias 2)

Each one of these interface files will contain the specific IP Address details that are required, but they will all be associated with a single interface. The ifcfg-eno1 file is the main interface configuration file and the other two files ifcfg-eno1:0 and ifcfg-eno1:1 only contain the extra information, these are known as aliases.

ifcfg-eno1 - In this file we have all of the primary IP Address details:

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
NAME=eno1
UUID=ca90417f-9df9-4f34-bae5-77aa6531cc00
DEVICE=eno1
ONBOOT=yes
IPADDR=192.168.1.223
PREFIX=24
GATEWAY=192.168.1.1
BOOTPROTO=none
IPV6_DISABLED=yes
IPV6INIT=no
```

In the next two files, we just enter the additional information:

ifcfg-eno1:0

```
NAME=eno1
UUID=ca90417f-9df9-4f34-bae5-77aa6531cc00
DEVICE=eno1:0
ONBOOT=yes
IPADDR=10.32.13.233
PREFIX=24
GATEWAY=10.32.13.1
PREFIX1=24
```

ifcfg-eno1:1

```
NAME=eno1
UUID=ca90417f-9df9-4f34-bae5-77aa6531cc00
DEVICE=eno1:1
ONBOOT=yes
IPADDR=137.1.0.100
PREFIX=24
GATEWAY=137.1.0.1
PREFIX1=24
```

Play close attention to the **DEVICE** setting, you must set the name to match the interface file name (so for file ifcfg-eno1:1 the **DEVICE** name must be eno1:1)

After rebooting the unit, we can now see that there are three IP Addresses on interface ifcfg-eno1.

Using ip -br -c a to show the IP Addresses:

eno1 UP 192.168.1.223/24 10.32.13.223/24 130.1.0.233/24

If we perform an ip route:

```
default via 192.168.1.1 dev eno1 proto static metric 101
10.32.13.0/24 dev eno1 proto kernel scope link src 10.32.13.223 metric 101
130.1.0.0/24 dev eno1 proto kernel scope link src 130.1.0.233 metric 101
192.168.1.0/24 dev eno1 proto kernel scope link src 192.168.1.223 metric 101
11.0.0.0/16 dev eno2 proto kernel scope link src 11.0.223.2 metric 102
```

We can see that the default route is still via **eno1**, and that the other alias IP Addresses are still going via **eno1**.