Server System Management - Linux

Lab 08: Linux networking, DNS, DHCP

INHOUD

[1 Lab 08: Linux networking, DNS, DHCP 3](#_Toc156984009)

[1.1 Intro : 3](#_Toc156984010)

[1.2 Setup Steps : 3](#_Toc156984011)

[1.3 Exploring the ip command : 4](#_Toc156984012)

[1.4 All hail the /etc/network/interfaces 5](#_Toc156984013)

[1.5 Exploring the gateway 6](#_Toc156984014)

[1.6 I’m a DNS Client ! 6](#_Toc156984015)

[1.7 Host(es)s with the Most(es)s 6](#_Toc156984016)

[1.8 I’m a DHCP Client ! 7](#_Toc156984017)

[1.9 Meeting DNSMASQ ! 8](#_Toc156984018)

[1.10 I’m a DNS Server ! 8](#_Toc156984019)

[1.11 I’m a DHCP Server ! 9](#_Toc156984020)

# 

# Lab 08: Linux networking, DNS, DHCP

## Intro

This is a pretty cool lab setup. In it, you can try out the most complicated large company scale network setups you can imagine, and experiment and learn from this for days, if not weeks.

True to the principle "aim for the stars, settle for a lamppost", we will refrain from going that far for now;

This lab will guide you through the following:

1. Connect to multiple network interfaces

2. Claim a static IP in your network , manually and through ifup/ifdown

3. Set up the DNS servers for the static-IP client (when not using dhcp) manually and through ifup/ifdown

4. Show and configure the (default) router, aka gateway for the static-IP client

5. Configure an interface, the router and DNS settings through DHCP

6. Set up your own DNS server with your own records, listening on the static IP you've obtained a while ago, and test it

7. Let another vm test your dns server

8. Set up a DHCP server and let another VM get an IP from it (and the other way around)

9. Use your DHCP server to do naughty stuff to your unexpecting clients

## Setup Steps

1. Add 2 extra Network cards to your VM using its VMWare Workstation Settings. Set both of them to NAT (this will allow the next lab to “probably” work without a hassle.

A screenshot of a computer program

Description automatically generated

You’ll need 2 vm’s to play the network game. So use VMWare’s CLONE option to create a clone of your original vm and call it debian-yourfamilyname-yourfirstname-second.

You can use the linked clone setting as it uses less resources.

Power up both machines.

Log in as root on both.

su - root

2. On your Second : Rename the cloned machine to debian-yourfamilyname-yourfirstname-second using the hostnamectl command. This command makes a permanent change in one of the config files. Find out which one by using the man hostname command

hostnamectl set-hostname debian-ciobanu-serafim-second

3. On your Second : Also make sure your FQDN (debian-yourfamilyname-yourfirstname-second.local) is also updated. The man page will help you. Make the necessary adjustment in another file to achieve this. Now logoff and logon as root again.

nano /etc/hosts

127.0.0.1 debian-ciobanu-serafim-second.local debian-ciobanu-serafim-second

A screen shot of a computer

Description automatically generated

**UPLOAD A SCREENSHOT OF BOTH VM’s logged in as root & showing your prompt & the result of the date command (SCREENSHOT)**

## Exploring the ip command

4. On your Prime : Show a list of all network interfaces in your virtual machine using the ip command. You should find 3 ethernet interfaces. (their name start with ens).

ip a

A screenshot of a computer

Description automatically generated

5. Now you see that your VM has 3 network devices:

• the original one you have been using up till one: ens33

• ens36: a guinea pig network.

• ens37: another guinea pig network.

**How many of these network interfaces are UP at this moment ? (QUIZ)**

only one – ens33, because it says UP next to it.

6. Create an IP range that you wish to use for your ens36 network, and pick 2 IPs in it.

Use 10.10.xx.AA/24 where xx is the number of your VMWARE Subnet. The subnet mask for this range is 255.255.255.0. Your own (static!) IP should be set to

10.10.xx.AA/24, where xx is again the number of Linux VMWARE Subnet and AA is 1 for your prime and 2 for your second

Subnet mask – 255.255.255.0

Prime – 10.10.206.1/24

Second – 10.10.206.1/24

A screenshot of a computer

Description automatically generated

7. Configure ens36 thusly on both vm’s , using the ip command to do this (remember to set the netmask and broadcast correctly);

ip addr add 10.10.206.1/24 broadcast 10.10.206.255 dev ens36 – prime

A screenshot of a computer

Description automatically generated

ip addr add 10.10.206.2/24 broadcast 10.10.206.255 dev ens36 – second

A screenshot of a computer program

Description automatically generated

8. **Bring up ens36 on both vm’s using the ip command. Which command did you use ? (QUIZ)**

ip link set dev ens36 up

ip link set ens36 up

9. Confirm it was configured correctly using ip a s ens36

A screen shot of a computer

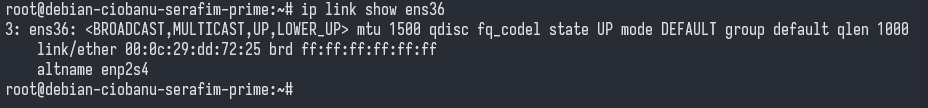
Description automatically generated

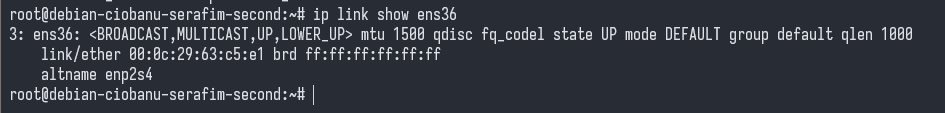
A screenshot of a computer

Description automatically generated

10. Which MAC address does your NIC have on this interface? Extract this info from ip.

ip link show ens36





11. Check if the IP stack is cooperating with us; let's ping our own address we gave ourselves earlier: ping 10.10.XX.AA

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

You should be receiving ping responses from your own IP! If not, your interface is not up or you set your broadcast and/or netmask wrong. Correct it | Bring it back up, or feel free to ask for help if you don't get it working.

12. Read man ip and ip --help and other sources of documentation, along with trying the actual commands on your VM to get to grips using the ip command.

13. You can also ping to the other VM’s nic ! try it !.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

14. Shutdown your ens36 again on both debians. Find out how to do it using the ip command. Make sure to FLUSH them thoroughly or you’ll get into trouble later on in this lab…

**What IS the flush command for ens36 ? (QUIZ)**

ip link set dev ens36 down

ip addr flush dev ens36 - QUIZ

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

15. Make your life easy and open a second ssh session to your server USING YOUR ens33 address, in a second window (no need to be root there). Put the windows next to each other, and read any useful man pages or other documentation there, while still doing your main work in the first session.

16. Open man interfaces in what I shall from henceforth refer to as the helper window. Keep referring to it to know how to edit this file.

**Note**

It's good to know how to configure your interface manually, like we have just been doing, to deal with emergency situations or to test your configuration while fine-tuning before committing to a more permanent configuration.

Also, you may have noticed a lot of specific ethernet and network related settings that can be modified using ip.

However, a well-configured linux system will automatically bring up interfaces at boot and down at shutdown, and it will provide ifup and ifdown scripts to bring interfaces up and down without typing the full IP and other settings every time.

That is what we will do next.

## All hail the /etc/network/interfaces

17. On your Prime : Edit /etc/network/interfaces. This file contains the network settings that are to be applied automatically at boot and by helper scripts.

Create a new iface block for ens36

Assign a static IP address to your ens36: the same IP that you assigned manually earlier in the lab.

Specify the correct subnet mask. Get inspiration from the man page in the helper window. Make sure you specify the broadcast address like you did before using the ip command.

iface ens36 inet static

address 10.10.206.1

netmask 255.255.255.0

broadcast 10.10.206.255

**Enter the block you created here (QUIZ)**

iface ens36 inet static

18. Use ifup ens36 to bring up your interface. (If you get an RTNETLINK error here, you probably didn’t flush before… And it’s terribly bad manners not to flush after you’ve used the facilities in ANY case)

A screenshot of a computer

Description automatically generated

19. Check, using ip as well as ping, if your interface is up. The output of the ip command for your interface should be like the output you got before, when you configured the interface manually.

A screenshot of a computer

Description automatically generated

Seems to work on self ping

20. Good to know: you can let the system bring up the interface at boot time using the auto keyword. The allow-hotplug works in a similar way. **Explore the interfaces man page to spot the difference between auto and allow-hotplug. (QUIZ)**

auto ens36 (Should I add it or not?)

auto = Specifies that the specified network interface should be brought up automatically at boot time.

allow-hotplug = Specifies that the specified network interface should be brought up automatically when it's "hotplugged" into the system, i.e., when it's physically connected or disconnected while the system is running.

21. Bring ens36 down again using ifdown ens36 and confirm it is really down, using ip and ping. Take note of the specific ping output you get.

A screenshot of a computer program

Description automatically generated

I do not get the error right now, but it should return an error like RTsomething.

**Note**

ifup and ifdown are the tools you use to quickly apply the changes you just made to

/etc/network/interfaces to an interface. When using dhcp, these will be (roughly) the

linux counterparts of ipconfig /release and ipconfig /renew from windows.

22. On your Second : Set ens36 up the same way on your second Debian. Make sure to add the auto keyword in both configurations.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

## Exploring the gateway

23. On your prime : Show the default gateway(s).

One very universal way to do this on Unix machines is the netstat -r (route table) option of the ip command.

Another way is by typing route -n. Try both. (If the commands are not found, you will need to install the net-tools package using apt.)

What does the -n option do for route? (man page)

A computer screen shot of a computer

Description automatically generated



24. On your Prime : **You can also show this info using the ip command. How? (man page, examples, --help, ...) (QUIZ)**

Note that your VM host is listening on the .2 address of the 192.168.X.0/24 range (the network into which your ens33 interface is plugged in).

This is your VMWARE router.

ip route

ip route show default

ip route show - QUIZ

A screen shot of a computer code

Description automatically generated

25. On your Prime : View /etc/network/interfaces again using less. How could you configure the default gateway persistently in this file? (ie: set at boot, used by ifup/ifdown, ...)

less /etc/network/interfaces

by adding line:

gateway <address>

26. Don't do it, since you would break your connectivity, but just do research on how to use the ip command to change the default route.

Hint: here are the commands that would do the trick if you were using the route command to do the same thing. So you just need to find the ip counterpart command for each of these commands.

**Warning**

DO NOT TYPE THESE COMMANDS! (just inspiration)

routte del default gw 192.168.X.2

# (but **route** instead of **routte** - to protect you from

# impulsively copy-pasting and executing this line)

routte add default gw 10.10.X.2

# (example to add the new default route, to 10.10.X.2)

# (this would only work if you are already in this range

# with one of your interfaces)

# After all, a default gateway must be reachable in order to redirect traffic to it!

## I’m a DNS Client !

27. On your Prime : Open the man page for resolv.conf and open the file in a text editor. **What is/are the name server(s) set to currently? (QUIZ)**

A screenshot of a computer

Description automatically generated

VMWare router I suppose. because default ip range is 192.168.206.2, and in the previous question it was said that .2 is the router

Now add another, non-existing, nameserver below the existing one : 192.168.123.123

## Host(es)s with the Most(es)s

28. Google about the concept of a hosts file.

In Windows, it resides in c:\windows\system32\drivers\etc\hosts.

On Linux and by extension Unix, it resides in /etc/hosts. You’ve been there before to store the FQDN. Display the contents of this file again.

It also has a man page which you could open in your helper window.

If necessary, read it to understand how to do the following.

29. On your Prime : Add a hosts entry to /etc/hosts that creates the dummy name (not in the nameserver, but valid for your linux vm) fixedipprime and assigns the value 10.10.xx.1 to it (where 10.10.xx.1 is the IP you chose in your ens36 range).

A screen shot of a computer

Description automatically generated

**Enter the line you added to your /etc/hosts file (QUIZ)**

10.10.206.1 fixedipprime - works

30. On your Prime : Test if your linux system recognizes this name - let's try to ping fixedipprime. This will resolve to the correct IP (the entry you added in /etc/hosts).

A screen shot of a computer

Description automatically generated

It resolves to the correct IP, but I do not have the interface up.

ifup ens36

A screen shot of a computer code

Description automatically generated

ifdown ens36

*Note*

Using the hosts file, you can override names that point to a different IP in DNS, if this is more convenient for you or if you want to make your linux VM think the name has a different IP (or point it to yourself, to 127.0.0.1).

## I’m a DHCP Client !

31. On your Prime : (Client side DHCP) Add a config block for ens37 in /etc/network/interfaces. Configure it for DHCP (for this, the man page will be really helpful as it contains the required stanza literally). Make sure this interface *doesn’t* try to start up at boot time.

A screenshot of a computer

Description automatically generated

32. On Your Prime : Remember, after editing /etc/network/interfaces, you need ifup to apply the config (and if ens37 had already been up, you'd first need to type ifdown ens37). Do this: bring ens37 up using ifup.

Since we configured the interface for DHCP, you will see output that shows the progress of finding a DHCP server and getting assigned an IP.

Which IP address did you get?

A screenshot of a computer

Description automatically generated

192.168.206.141 (normal and usual for a VM)

If you do not get an IP address, let your coaches know and we will look for the cause.

To troubleshoot, you can check if the DHCP client daemon, that is supposed to ask for an IP to a DHCP server, is running:

ps -ef | grep dhclient

**What is the MAC address for your Prime’s ens37 network card? Remember this for later ! (QUIZ)**

**A screen shot of a computer

Description automatically generated**

00:0c:29:dd:72:2f

33. On your Prime : Show the contents of /etc/resolv.conf again. What do you notice ?

It has automatically been modified! Your 192.168.123.123 from step 27, is gone!

That's because the nameserver settings were defined by the DHCP server that gave you an IP.

A screen shot of a computer

Description automatically generated

34. Show the default gateway using ip route show.

Was it also modified from the initial value of 192.168.X.2 to something else by DHCP? Why / Why Not ?

A screenshot of a computer code

Description automatically generated

All that was changed is the addition of another ip address.

35. On your Prime : After this client DHCP crash course, release your IP again and shutdown the interface ens37 which was used for testing client-side DHCP and DNS using ifdown ens37

dhclient -r ens37

ifdown ens37

## Meeting DNSMASQ !

36. On your Prime : Update your apt package repository using apt update. **ALWAYS UPDATE YOUR REPOSITORY BEFORE INSTALLING NEW/IMPORTANT PACKAGES**. Upgrade any packages it reports. Now install the dnsmasq package using apt;

apt update

apt upgrade

apt install dnsmasq

DNSMASQ is a lightweight DHCP Server, DNS Server/Forwarder and all round pretty cool piece of kit.

37. On your Prime : Use dpkg -L dnsmasq | less to show the contents of the package. Now modify this command to filter just the lines containing etc (config files) using grep. What seems like the config file for dnsmasq, from inspecting the package?

dpkg -L dnsmasq | less | grep 'etc'

A screenshot of a computer

Description automatically generated

/etc/dnsmasq.conf

38. On your Prime : Go to that file and show its contents

cat /etc/dnsmasq.conf | less

## I’m a DNS Server !

39. On your Prime : reading the configuration file, **what file does dnsmasq use to find which nameserver it should forward its queries to by default ? (QUIZ)**

A blue and white text on a black background

Description automatically generated

/etc/resolv.conf

40. DNSMASQ is a nifty little tool, with many nice features. For starters, it can act like a DNS proxy. It will forward DNS queries to your preferred DNS server (as we saw just before), but also intercept/change/answer DNS queries about stuff it knows something about itself. This allows you to tweak DNS queries in your network. It will for example serve up what you put in your /etc/hosts file via DNS to other machines. Let’s try this.

41. On your Prime : When exposing a server, **ALWAYS make sure you’re exposing it on a fixed ip address**. Since you’re exposing a DNS Server, change the dnsmasq config file to only listen on your interface that has a fixed ip address. Luckily, we have a fixed ip interface available !

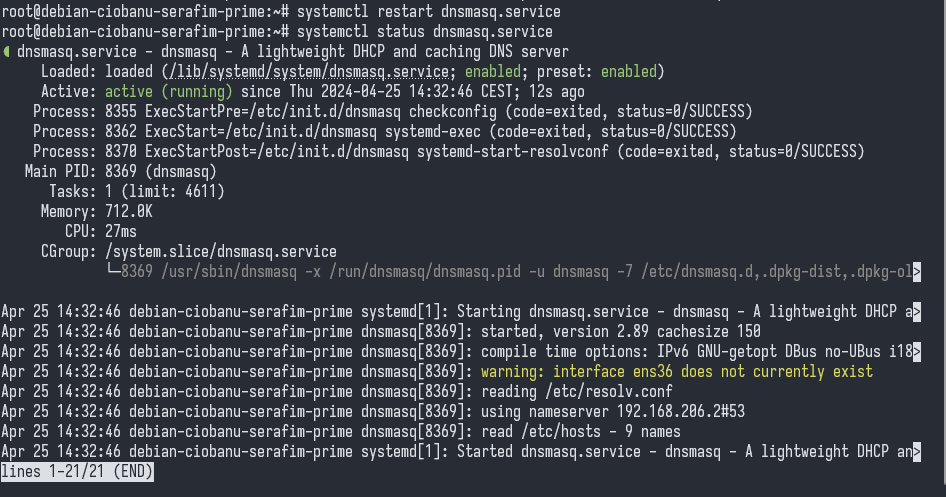
A blue text on a black background

Description automatically generated

interface=ens36

42. On your Prime : Restart your dnsmasq service using systemctl restart . check out its status by using systemctl

systemctl restart dnsmasq.service



43. On your Second : Remember the fixedipprime entry you put in prime’s /etc/hosts file ? the content of that file is NOT known on your second of course. Try pinging fixedipprime.

Nothing happens. Name or service not know…

A black background with white text

Description automatically generated

44. On your Second : Now let’s test your brand new DNS server/forwarder ! edit the file that specifies the nameserver and change the desired nameserver to your Prime’s 10.10.XX.1’s fixed ip address.

sudo vim /etc/resolv.conf

A screen shot of a computer

Description automatically generated

nameserver 10.10.206.1

44. On your Second : Try pinging fixedipprime again. This time, it works !

A screen shot of a computer

Description automatically generated

Did not work first time so hasd to check it again

45 On your Prime. This opens options… Edit your /etc/hosts file and add kbc.be to the line where you put fixedipprime before. Reload the configuration using systemctl reload dnsmasq

A screenshot of a computer screen

Description automatically generated

45. On your Second : ping kbc.be . Who answers ?

A screen shot of a computer

Description automatically generated

The same ip, meaning that it understands our own DNS name resolvers.

**Note**

You can put any adresses alias’s in your Prime’s /etc/hosts now. You can e.g. make [www.vrt.be](http://www.vrt.be) point to [www.vtm.be’s](http://www.vtm.be's) ip address if you want to.

You can tweak dnsmasq’s DNS configuration even further to e.g. deny/redirect certain domain addresses. You will have to reload (and sometimes even restart) dnsmasq when you make config changes effective though.

So a well tweaked, intra-corporate DNS Server can be a boon. Of course, a rogue or compromised DNS server can be a MAJOR threat.

If you run into trouble with your conf file, try dnsmasq --test

You may discover that after a while, Second will stop using your dnsmasq server again though ! (IMPORTANT)

Why do you think this happens ?

Because it still has the other interface which gets the DHCP settings, from VMWare for example.

## I’m a DHCP Server !

46. Until now, we’ve always used the VMWARE DHCP Server to get dynamic addresses. Let’s change that now for one of our network interfaces

47. DNSMASQ has many more tricks up its sleeve. Another is a fully functional DHCP Server !

48. On your Prime : Go back to the dnsmasq config file. Find the configuration part , you can set up an ip range between 10.10.XX.100 and 10.10.XX.110 , lease time 12h using just ONE LINE !

**Enter that line for YOUR setup below (QUIZ):**

**A screenshot of a computer

Description automatically generated**

dhcp-range=10.10.206.100,10.10.206.110,12h

49. On your Prime : Restart the dnsmasq service (you may have to restart/reload quite a few times when you make changes further on)

systemctl restart dnsmasq.service

50. On your Second : Restore the nameserver to its original value in the appropriate conf file. Add an entry in /etc/network/interfaces so that ens37 will try and get a dhcp, like you did on Prime before (also make sure this interface doesn’t try and turn itself on at boot time)

**A black background with green text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

51. Time to test your DHCP server. In order to do this, turn off your dhcp server on your vmware like we did in the windows lab (vmnetdhcp service). (Make sure to turn it back on again when you’re done with this lab)

A screenshot of a computer

Description automatically generated

52. On your Second : Now use ifup on your ens37.

It will get a network address in the range you specified. In other words, your second’s ens37 nic got an ip address from your prime’s ens36 nic!

**Upload a screenshot of the result of the ifup command (SCREENSHOT)**

**A screen shot of a computer

Description automatically generated**

53 On your Second : Verify that the nameserver has changed again !

A black background with green text

Description automatically generated

54. On your Prime, try to get a DHCP ip by ifdown/ifup Prime’s ens37 interface. Check if an IP in the range you defined is assigned to your interface.

A computer screen shot of a computer

Description automatically generated

You do get an IP address.

55. On Both : Skim through /var/lib/dhcp and find the file that stores the leases. You’ll notice that on Prime, you got a dhcp address from yourself !

A computer code with white text

Description automatically generated

56. Remember where the nameserver is stored ? **Where does it point to now for prime & second ? (QUIZ)**

10.10.206.1

At this point for some reason, I can not connect simultaneously to the both vm’s anymore…

OBVIOUSLY IF YOU TURN OFF THE VMWARE DHCP SETTINGS AND THEN NEED TO USE OTHER IP ADDRESSES

A screenshot of a computer

Description automatically generated

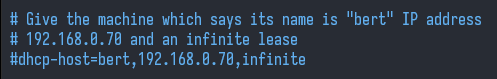
so ssh to [ciobanuserafim@10.10.206.103](mailto:ciobanuserafim@10.10.206.103) works obviously.

Now you’re the DHCP/DNS boss ! (until your re-enable your vmware’s dhcp server, that is :)

57. Now try and get your dnsmasq dhcp server to always hand out a determined ip address 10.10.XX.113 to your Prime’s ens37 card and be reachable under the name boris, with a lease time of 24 hours



WRONG



**Complete the configuration file snippet to do this (it’s just ONE line again) (QUIZ)**

dhcp-host=00:0c:29:dd:72:2f,10.10.206.113,boris,24h

A blue and white numbers on a black background

Description automatically generated

Works for getting the stuff

58. On your Prime : Now for the final proof : ifdown & ifup your ens37 and get that .113 address. ssh to yourregularuser@boris immediately after. **Upload a screenshot of the ifdown ens37 & ifup ens37 & ssh yourregularuser@boris commands you ran on second to prove that you succesfully handed out the 113 address**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

59. You can tweak this even further. E.g. if we want our DNSMASQ server to set up google’s DNS servers (8.8.8.8), you can adapt your dnsmasq configuration to do this. (you will need to reload/restart dnsmasq to accept the changed configuration). Comment this line out after you’ve tested it though, or it may provoke “oddities” in next week’s lab.

**What line would you add ? (QUIZ)**

dhcp-option=option:dns-server,8.8.8.8

**A screenshot of a computer program

Description automatically generated**

**A computer screen shot of a server

Description automatically generated**

60. Now ponder a moment on the power of DNSMASQ

Did you realize what happened when your Prime’s ens37 NIC cried out for a dynamic IP address ?

Your Prime’s ens36 answered and gave it a specific IP address, tied to it’s Mac Address.

It then forced your Prime to adopt the hostname/alias boris.

It then set your (DNS) nameserver to whatever you wanted, or hijack it altogether using it’s own DNS server

It then fed you whatever your Prime had stored in its /etc/hosts as true and valid DNS records.

As with point 45, a well set-up DHCP DNSMASQ in a corporate environment is easy to set up and maintain, and can/will be an asset to any Linux admin.

But a rogue setup…

61. And It doesn’t stop here ! Next week, we’ll bring everything we’ve seen this year together, add a little trivial ftp server, some magical pixiedust and get a Linux system to boot without having ANY Storage Medium (DISK, USB STICK, SD CARD) attached.

You’re in for quite a ride !