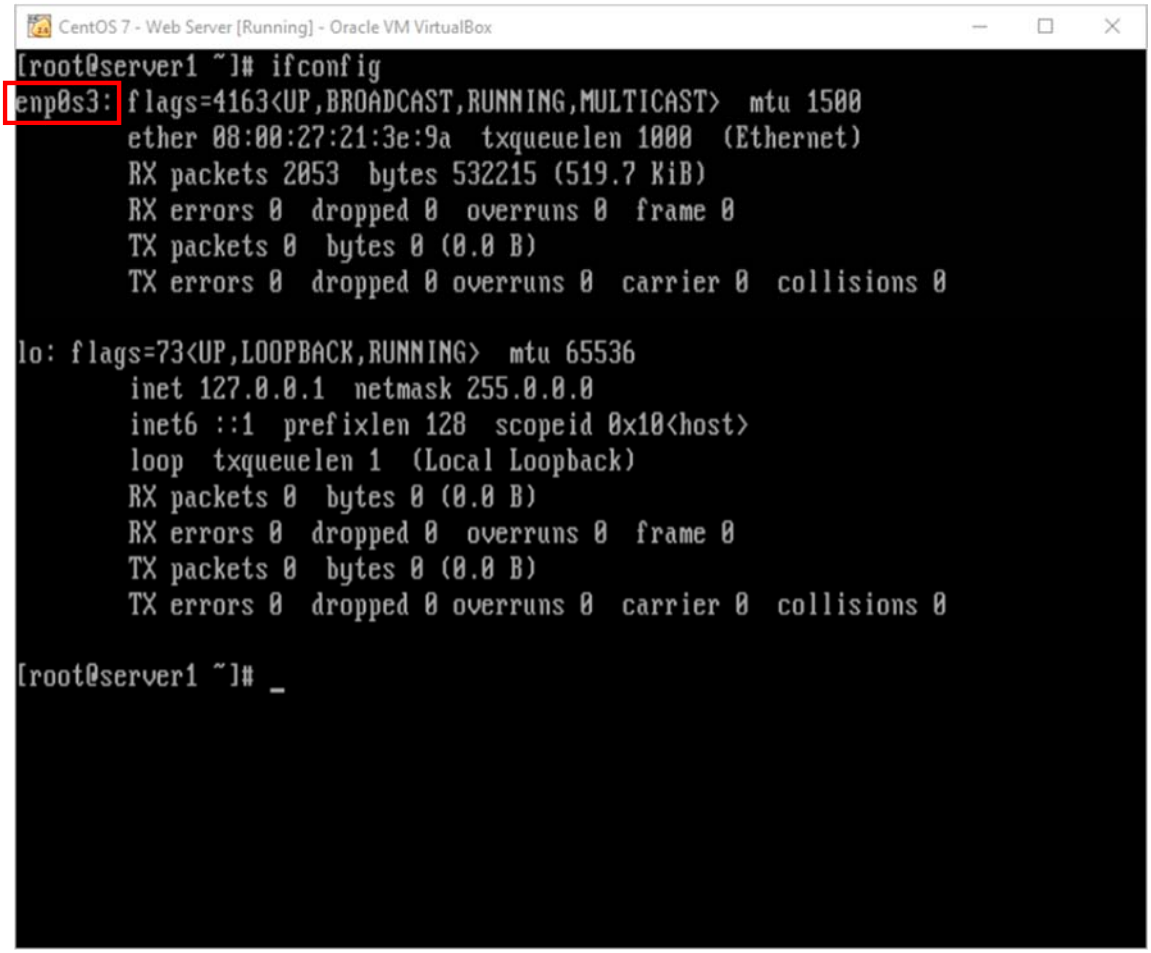
Configure Server IP Address

For the server to be accessible and serve webpages to remote clients (visitors), we will need to assign a static IP address to the server. A static IP is an efficient way for configuration of routing paths to the server, as with a static IP, opposed to a dynamic or DHCP assigned IP address, the IP configuration does not change automatically.

To configure the IP address in CentOS we need to check a few things first, type the following command into the CLI (Command Line Interface), the CLI is also known as the shell prompt:

Note: ↵ = Press the enter key.

```
ifconfig ↵
```



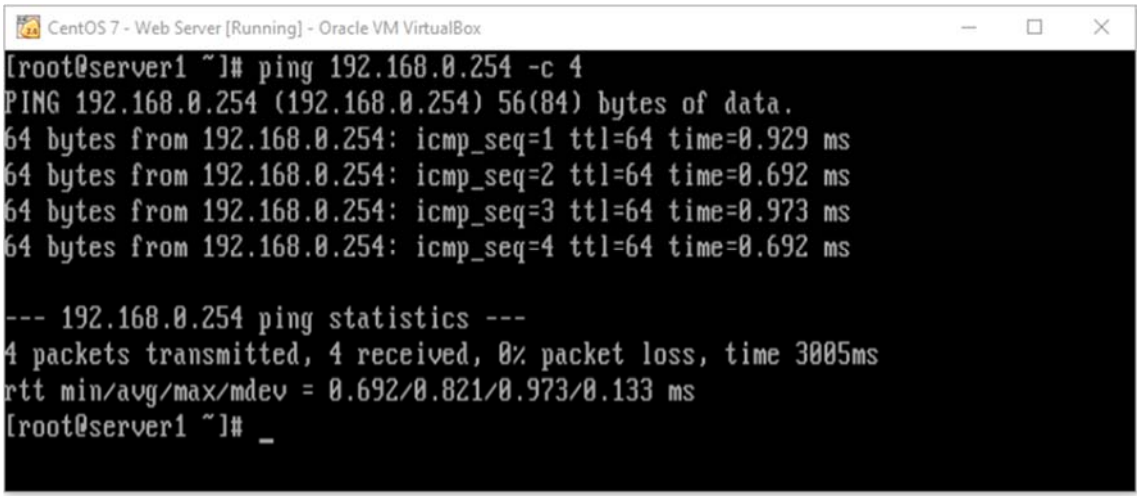
We can see that the name of the ethernet network adaptor is present and named `enp0s3`.

Set this interface with an IP address relevant to the network we need to connect the server with. In my case I am going to connect the server to 192.168.0.0/24 network with an IP address of 192.168.0.78 as follows:

```
ifconfig enp0s3 192.168.0.78/24 up
```

We can test the connection by using the ping command, this command is used to send empty packets (datagrams) to a remote host to test if that host is on a reachable network, if the host receives a request from the server it will reply. I will ping the DNS server on my network. Type the following command at the prompt replacing the IP address with a host on your network:

```
ping 192.168.0.254 -c 4
```



Now that the network interface is operational, we need to setup the DNS and default gateway to allow the server to connect to the internet. First, let's setup the default gateway, the default gateway for the internet on my network is 192.168.0.253, you will need to find the gateway address of your network, this is usually the internet router. Type the following command at the prompt:

```
route add default gw 192.168.0.253
```

Set the DNS configuration, I will use the DNS server on my network you will need to replace that with your DNS, if you don't have a dedicated DNS server you can use your ISP's DNS server:

```
nano /etc/resolv.conf
```

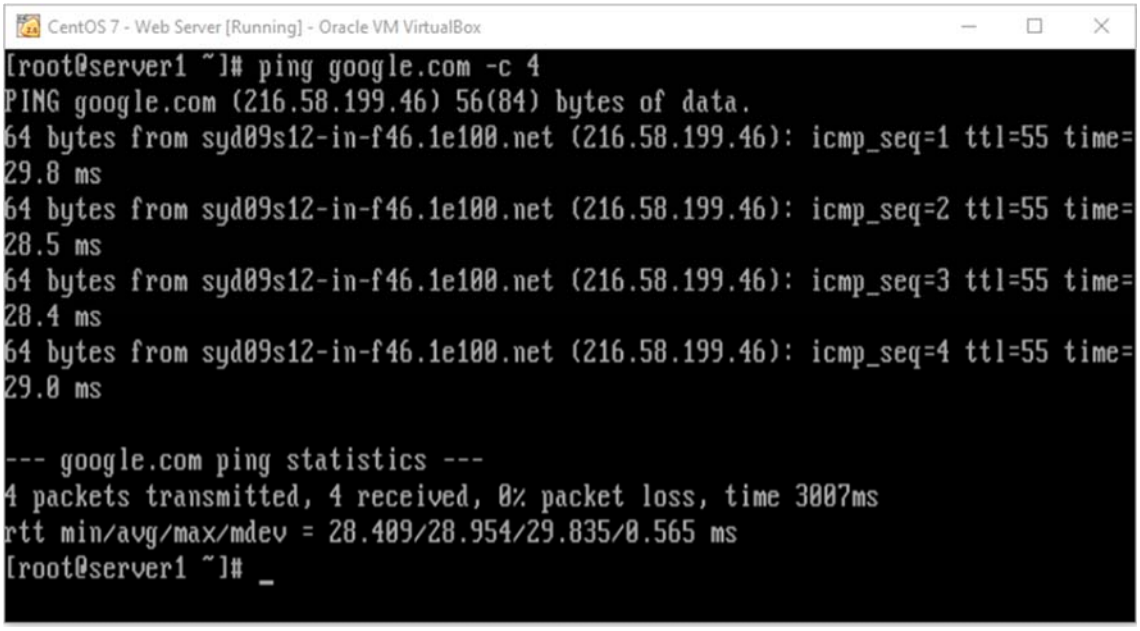
Add the following line to the end of the *resolv.conf* file:

```
nameserver 192.168.0.254
```

Save and exit the file, press `CTRL+X`, type `Y` then press the `Enter` key to save file name *`resolv.conf.`*

To test that the server can reach the internet, ping google as follows:

```
ping google.com -c 4
```



Now that the servers IP,DNS and routing setting have been configured, we need to save this into a script that will reload this configuration after a reboot. As this setup is for training purposes only and not for production, we can use the built-in compatibility script named `rc.local`. This script can be used as a temporary place to execute commands on system start-up, it is recommended that the use of the standard configuration scripts should be configured if you intend to set this up as a live production server.

Add the configuration to the `rc.local` script as follows:

```
nano /etc/rc.d/rc.local
```

Note: Replace the following IP addresses with **your** network addresses.

```
[..]
ifconfig enp0s3 192.168.0.78/24 up
route add default gw 192.168.0.253
iptables -F
```

You will notice that another command has been added to the `rc.local` script, this is to clear the firewall and allow remote connections to the server, you can also issue this command at the shell prompt.

Save and exit the file, press CTRL+X, type Y then press the Enter key to save file name *rc.local.*

The server should now be configured with the IP addressing, DNS and routing information, reboot the system to test using the following runtime level command as follows:

```
init 6
```

After the system reboots, log back in with the root account and test connectivity by pinging a local network IP address and an internet domain like google.com.