VyOS (Router/Firewall)

VyOS is an open source router based on Debian. It does not feature a WebUI like pfSense or Sophos do, meaning you can only configure it via CLI. However it is incredibly fast and versatile.
Having no GUI means it's a bit harder to configure and not that easy to see the whole complete outcome, however it will teach you more about the actual networking than a GUI does and you have another reason to break something.

Table of Contents

How does the CLI work?

Let us first take a look at the CLI. You'll find that it looks just like a Debian based server.

```
Linux vyos 5.10.7-amd64-vyos #1 SMP Sat Jan 16 12:09:04 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 20 00:05:24 2021 from 10.0.0.230
vyos@vyos:~$
```

Let's cover some basic commands first. First it is important to know how to get help in de CLI. You can type ? and a list of all possible commands will show up. It is also possible to write that in command you are currently typing which will show all possible completions. (Please note that the ? is not shown in the console).

```
vyos@vyos:~$ show interfaces
Possible completions:
  <Enter>        Execute the current command
  bonding        Show bonding interface information
  bridge         Show bridge interface information
  counters       Show network interface counters
  detail         Show detailed information of all interfaces
  dummy          Show dummy interface information
  ethernet       Show ethernet interface information
  input          Show input interface information
  l2tpv3         Show L2TPv3 interface information
  loopback       Show loopback interface information
  macsec         Show MACsec interface information
  openvpn        Show OpenVPN interface information
  pppoe          Show PPPoE interface information
  pseudo-ethernet
                 Show pseudo-ethernet/MACvlan interface information
  system         Show network information of all interfaces
  tunnel         Show tunnel interface information
  vrrp           Show vrrp interface information
  vti            Show vti interface information
  vxlan          Show VXLAN interface information
  wireguard      Show wireguard interface information
  wireless       Show wireless interface information
  wirelessmodem  Show Wireless Modem (WWAN) interface information


vyos@vyos:~$ show interfaces
```

Now lets list some basic commands:

- **clear console:** clears the console screen
- **show ?:** show information about x
- **configure:** enters privileged mode. In this mode you can make changes to configuration. You will know that you are in configuration mode if the path prefix changes from "~$" to "#".
- **set:** set/add option x.
- **commit-confirm:** activate the changes you have made. It will activate the changes and **THEN** a sk you for confirmation. This means that if something breaks and you lose access it will automatically restart (by default in 10 minutes). So make sure you wait a minute before confirm so you can actually test if the system works. Perhaps open a new SSH connection to see if you can still access the console (an open connection is not the same as opening a new connection!). Must be run in privileged mode (aka configure).
- **save:** save the committed changes into the boot file. Running just commit is not enough, it will not be saved when the system reboots. Use this to permanently save it. Must be run in privileged mode (aka configure). **Only run after commit-confirm!**

Information

More about VyOS

- Configuring VRRP with VyOS (Failover)
- Setting up a basic WAN-LAN network with DHCP
- Zone-Based Firewall

> **Tip!**
>
> Instead of writing the command in full you can always abbreviate as long as it is specific enough. For example, instead of writing "show interfaces" you can write "sh int" or "sho interf". Both commands are unique enough to link them with the full command.

Enabling SSH

Obviously you don't want to type everything in the VNC. To enable SSH simply enter

```
set service ssh port PORT
# For example:
set service ssh port 22
```

> **Warning!**
>
> Please be aware that this will open the SSH server on all interfaces! If you want this edit the SSH port and only allow logins via an SSH Key. Otherwise the router could get compromised rather quickly!

Configuring your WAN-interface

You have two methods for configuring the WAN side. Either via DHCP or manually, however since we do not have DHCP on our subnet you will need to configure it manually.

The first thing you will need to do for both is to identify your WAN-interface. You can see al interfaces by typing "show interfaces"

Manually

```
set interface ethernet ethX address '?.?.?.?/CIDR'
# For example:
set interface ethernet eth0 address '193.191.187.104/28'
```

And to make it clear that it is WAN we will give it a description:

```
set interfaces ethernet eth0 description 'WAN'
```

If you now type "show interfaces", you should get something like this:

```
+ethernet eth0 {
+    address 10.0.0.239/24
+    description WAN
+}
 loopback lo {
 }
```

That is done, however we are missing one big important thing. If our router receives a request from outside our network, it will have no idea where to send the response. Obviously we need to set our default route to our gateway.

```
set protocols static route 0.0.0.0/0 next-hop 193.191.187.110
```

Double check everything and then write "commit-confirm". If anything went wrong wait 10 minutes. The system will restart to it's previous configuration.