OpenVPN client

You can easily connect your Batocera to a VPN, as we ship OpenVPN with the distribution. However, it requires some manual configuration, and the steps involved will most probably be depending on your VPN provider.

In this example here, I will be connecting a Batocera 5.27 client to a NordVPN server, and adapt it to PIA when possible. The method here can be adapted to other VPN providers quite easily, please feel free to share your experience on the forum of Discord channel.

OpenVPN configuration

- Create a new folder for your OpenVPN configuration with `mkdir /userdata/system/openvpn`
- If your VPN provider offers you pre-configured openVPN configuration files download them in that directory.

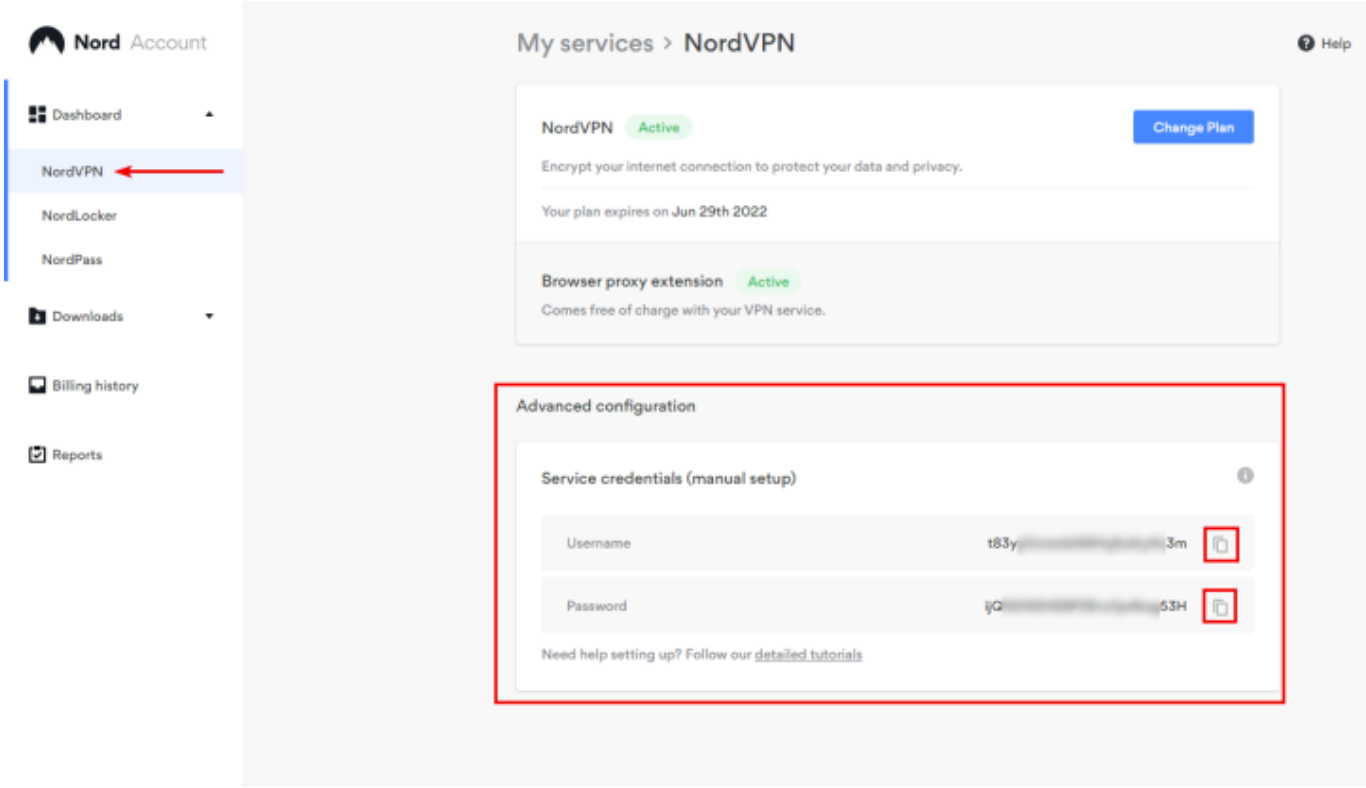
With NordVPN, I can get access to those files with:

```
cd /userdata/system/openvpn
wget https://downloads.nordcdn.com/configs/archives/servers/ovpn.zip
unzip ovpn.zip
rm ovpn.zip
```

With PIA:

```
cd /userdata/system/openvpn
wget https://www.privateinternetaccess.com/openvpn/openvpn.zip
unzip openvpn.zip
rm openvpn.zip
```

- Then, you need to create a new authentication file `/userdata/system/openvpn/auth.txt` that contains only two lines: first line is your login, second line is your password. No space, no tab, no empty line, just those two lines with login and password provided by your VPN server vendor. If you use PIA, you need to have a username starting with p (like p1234567, not just the numbers). In the case of NordVPN, you can get them from the dashboard.



- Select the VPN server you want to connect to, and find the associated openVPN configuration file. For example here, I want to connect to a VPN hosted in France, and will select the configuration file `/userdata/openvpn/ovpn_udp/fr661.nordvpn.com.udp.ovpn`. Edit this file, find the line that states `auth-user-pass` and append your authentication credentials file path to it:

```
auth-user-pass /userdata/system/openvpn/auth.txt
```

- Save the file... and now your VPN configuration is done! To connect to the VPN, just launch the command line:

```
openvpn /userdata/system/openvpn/ovpn_udp/fr661.nordvpn.com.udp.ovpn
```

Similarly with PIA:

```
openvpn /userdata/system/openvpn/us_silicon_valley.ovpn
```

Automatically launch the VPN when Batocera boots

It is possible to turn on the VPN with Batocera's boot sequence via the use of scripting.

Add the following file to `/userdata/system`:

custom.sh

```
#!/bin/bash

if test "$1" != "start"
```

```
then
  exit 0
fi
openvpn /userdata/system/openvpn/<replace_me>.ovpn &
```

where <replace_me> is your VPN.

Tips and tricks

- You can verify that you are correctly connected to the VPN by checking your public IP address before and after openvpn is started by using the command `curl ipinfo.io` or `curl ifconfig.me`
- If you want to start up your VPN connection every time Batocera boots, you can add the command openvpn `/userdata/system/openvpn/ovpn_udp/fr661.nordvpn.com.udp.ovpn &` to the local custom startup script `/userdata/system/custom.sh` – this will be the very last process fired up in the boot sequence

Troubleshooting

It's not working!

First thing is to just check that your script is running in the first place. This is easy, just put something like:

```
test line >> /userdata/system/testoutput.txt
```

in the script and then search for `/userdata/system/testoutput.txt` on next boot.

The script is running but the VPN is still not working!

It could be that it's a problem with the VPN itself launching from the script. Even if the command works in SSH, running it from a script could be an entirely different story. In order to see the error code outputs from what the command would be doing (for example from openvpn):

custom.sh

```
#!/bin/bash

if test "$1" != "start"
then
  exit 0
fi
```

```
(openvpn /userdata/system/openvpn/<replace me>.ovpn &) 2>&1 | tee -a
/var/log/vpn.log
```

Adapt the <replace me> to your VPN of course.

My VPN works fine on my PC but not on my Raspberry Pi/other SBC!

The ARM build of Batocera does not include the necessary /dev/net directory and node structure that OpenVPN relies on by default. This can be added in with the script like so:

custom.sh

```
#!/bin/bash

if test "$1" != "start"
then
  exit 0
fi

if [ ! -d /dev/net ]; then
  mkdir -p /dev/net
  mknod /dev/net/tun c 10 200
  chmod 600 /dev/net/tun
fi

openvpn /userdata/system/openvpn/<replace_me>.ovpn &
```

Adapt the <replace me> to your VPN of course.