OPNsense installation on the Velocloud EDGE 510 router

The Velocloud EDGE 510 is a discontinued enterprise router now widely available on platforms like eBay for under $50 CAD. Following this guide transforms it into a fully functional firewall/router. Online reports show this should also be possible with the Edge 520 model.

> You can buy one of these on eBay and follow this tested and revised guide, based on multiple installations.

While the router's performance is modest by today’s standards, it features:

- Intel Atom C2358 CPU, 4 GB RAM, 8 GB eMMC
- AES-NI hardware acceleration (aesni0), useful for IPsec/OpenVPN performance.
- On some models :
  - Qualcomm Atheros QCA9880 / QCA986x 802.11ac wifi card (168c:003c) : detected as none3 by OPNsense, limited support in FreeBSD
  - Sierra Wireless EM7455 LTE modem (Qualcomm Snapdragon X7 LTE-A) detected as u3g0 (see Configuring cellular modems
- 2 USB 2.0 ports
- 4×1 Gbps Ethernet ports
- Reliable serial access and coreboot compatibility

This makes it a perfect candidate for homelab setups, network experimentation, and educational deployments. With OPNsense 25.7.6 (based on FreeBSD 14.3-RELEASE-p4), it supports full firewall and routing capabilities.

Special thanks go to PhoenixSheppy on Github and to the Netgate forum contributors, which this guide is based on.

Pre-install notes

- WAN port=GE4 port and LAN port=GE1 when booting into the stock Velocloud OS.
- WAN port=GE2 port and LAN port=GE1 port, starting from the OPNsense USB Liveboot stage. Internet will work right after first boot from the USB key. ⚠ **Changing these assignments in OPNsense may render the router unreachable.**
- The IP configuration after installation is as follows:
  - **WAN**: DHCP (address automatically obtained from the provider)
  - **LAN**: Static IP 192.168.1.1, with DHCP leases distributed within this range
- Wi-Fi is not enabled, despite antennas and wifi radio being present on some models. When present, the wireless chipset on this router (Qualcomm Atheros QCA9880 / QCA986x 802.11ac wifi card - 168c:003c) is not well supported by OPNsense. In general, OPNsense recommends using external access points with VLAN configuration for Wi-Fi.
- ⚠ **Despite having Gigabit ports, actual performance may be limited by the processor,**

RAM, and enabled OPNsense features. Check the OPNsense official documentation for more details.

OPNsense installer preparation and router coreboot configuration

1. Get the OPNsense **VGA** installer (yes, VGA) at https://opensense.org/download and create a USB installer from it using BalenaEtcher or your preferred method. **Keep this key handy in case you need to reinstall OPNsense when experimenting**
2. Remove the small metal plate on the back of the router, at the right of the network ports, this will expose a mini-USB port. Connect a mini-USB cable into the device, and the other end to your PC.
3. Using a program like Putty, connect at 115200 Baud, and connect the power cable in the router, you should see some output as it boots up. Allow it to boot into VeloCloud OS.
4. The default login is `root`, password is `VeloHelloXXX` (XXX=last-3 of S/N on bottom of device)
    1. **If this doesn't work**, reboot and watch closely, there will be a prompt offering to `press f` and ENTER to enter fail-safe mode
        1. Run the two commands below to change the root password

```
mount_root
echo -e "youpassword\nyoupassword" | passwd root
```

5. Let's connect the network before rebooting : connect a network cable from your router or switch to the 4th port (GE4). Connect your computer to port 1 (GE1).
6. Reboot the device, it will take 3-4 minutes before you can connect to it.
7. Youmay need to press **Enter** a couple of times to see the **vc-edge login:** prompt. Login with the password you configured before (user is **root**).
8. Change to the existing `firmware` directory:

```
cd /root/firmware
```

9. Grab a copy of the modified corebooot firmware :
`wget https://raw.githubusercontent.com/PhoenixSheppy/VeloCloud-Edge-510-OPNsense-Conversion-Guide/refs/heads/main/firmware/2017-4-10-coreboot.rom`

   This output should follow:

```
Connecting to raw.githubusercontent.com (185.199.108.133:443)
2017-4-10-coreboot.r 100% |********************************|  8192k
0:00:00 ETA
velocloud vc-edge:~/firmware  -
```

10. Change directory again :

```
cd ..
```

11. Let's update the DMI (Desktop Management Interface) data region in the firmware image:
`./dmi-tool -u firmware/2017-4-10-coreboot.rom`

This output should follow:

```
Updating flash with current DMI info ...
Current DMI:
decoding ...
SN=CKPG933, UUID_str=a04d141b-0ff4-4e25-bbf0-ded5867a5296,
BSN=PH0TTRNNNBP7L1, pname=2R0220 bversion=haEDG
Done. Please power cycle the board if the flash has been updated
```

12. Follow that with this command:
`./dmi-tool -w -p EDGE510 -v 1`

    This output should follow:

```
Programming DMI  ...
Current DMI:
decoding ...
SN=CKPG933, UUID_str=a04d141b-0ff4-4e25-bbf0-ded5867a5296,
BSN=PH0TTRNNNBP7L1, pname=2R0220 bversion=haEDG
Updating current DMI..
No new system SN provided, using the current system SN
No new UUID provided, using the current UUID.
No new Board SN provided, using the current Board SN
Done. Please power cycle the board if the flash has been updated
```

13. Let's now flash that firmware :
`flashrom –programmer internal –write firmware/2017-4-10-coreboot.rom`

    This output should follow:

```
flashrom v0.9.8-r1888 on Linux 3.14.79 (x86_64)
flashrom is free software, get the source code at http://www.flashrom.org

Calibrating delay loop... OK.
coreboot table found at 0x7fbee800.
Found chipset "Intel Avoton/Rangeley".
This chipset is marked as untested. If you are using an up-to-date version
of flashrom *and  * were (not) able to successfully update your firmware
with it,
then please email a report to flashrom@flashrom.org including a verbose (-
V) log.
Thank you!
Enabling flash write... OK.
Found Winbond flash chip "W25Q64.V" (8192 kB, SPI) mapped at physical
address 0x00000000ff800000.
Reading old flash chip contents... done.
Erasing and writing flash chip... Erase/write done.
```

```
Verifying flash... VERIFIED.
```

14. To finalize, the watchdog timer that reboots the systems and prevents from running another OS has to be disabled. **This is a critical step, and can't be skipped**. Run the following commands:

```
i2cset -y 1 0x24 0x00 0x00
i2cset -y 1 0x24 0x01 0x00
```

After you've run those two commands, your device is ready to install OPNsense.

OPNsense live boot from USB VGA installer

1. Unplug the power cable, plug the OPNsense VGA installer USB key prepared previously in either USB port.
2. Remember to have the mini-USB cable plugged into the serial port
3. Unplug the network cable from the GE4 port and plug it into the GE2 port. This is where OPNsense will search for and configure its WAN connection.
4. Plug a network cable from your PC into the GE1 port. This is where OPNsense will configure the LAN and provide DHCP leases. Make sure your PC connection to the router is stable, if you loose the SSH connection to the router during installation (for example, by your PC going into sleep mode), you will have to start over from that step.
5. Plug the power cable again. The boot sequence **will take 3-4 minutes**, be patient.
   The serial output should scroll by and briefly show the OPNsense logo and menu. The last few lines of output should look something like:

```
/boot/kernel/carp.ko size 0xfb90 at 0x244d000
/boot/kernel/if_gre.ko size 0xaa30 at 0x245d000
/boot/entropy size=0x1000
```

After **3-4 minutes**, you should be able to ping the router at 192.168.1.1.

OPNsense installation via SSH

1. The installer can the be launched by SSH, from your PC in a terminal:

```
$ ssh installer@192.168.1.1
```

2. When prompted, enter the password : `opnsense`
   The installer should launch. All questions can be kept as default except the following :
   1. At the second step, choose **UFS/UEFI Hybrid**. ZFS is ideal on systems with more storage and resources, but not practical on this configuration. You can decide to try it anyways, this is a homelab device to experiment.
   2. At the third step, choose the internal storage eMMC storage (“<Generic Ultra HS-COMBO 198> (7GB)” or similar) as target for installation. This will replace the factory Velocloud OS. The installer USB key may be detected as `da0`, the other storage (`da1`), rely on the

description.
3. After that, proceed with defaults. You will see progress indicators. This should take **under 8 minutes**.
4. You can then change the root password (default is `opnsense`), or just choose **Complete Install**.
5. Next step, choose **Shutdown** and wait for a couple of minutes. **Remove the USB key**, disconnect and reconnect the power cable, the router will then boot. This will take **3-4 minutes**.

You should now be able to ping the router at 192.168.1.1 and complete the configuration by opening a browser to https://192.168.1.1/ui/core/initial_setup - you may have to accept a security exception as the SSL certificate used is self-signed.

The default user and password (unless changed during the installation), are `root` and `opnsense`, respectively. Refer to the official OPNsense documentation for further details.

References

- VeloCloud Edge 510 OPNsense Conversion Guide
- Netgate original forum post
- Hardware sizing and setup
- OPNsense serial access
- OPNsense installer procedure in detail

From:
https://dulib.re/wiki/ - **Le Goût du Libre**

Permanent link:
https://dulib.re/wiki/doku.php/opnsenseonvelocloudedge510

Last update: **2025/11/30 13:55**