HOW-TO-GUIDE

demonstrating FA using the virtual Fabric Attach Demonstrator

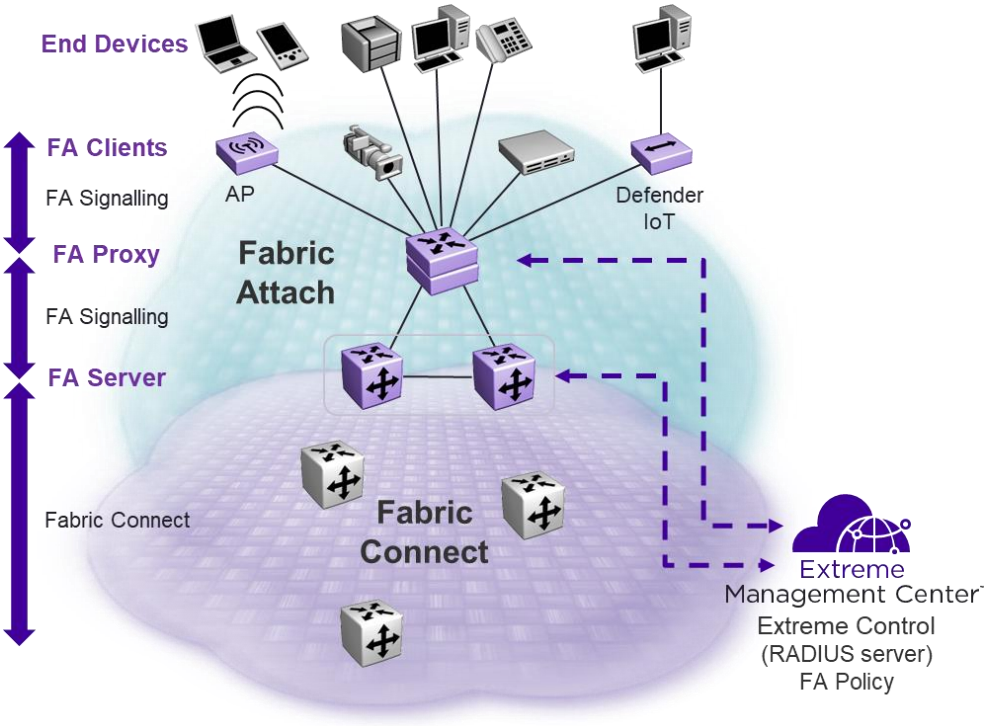
1. Target audience

System Engineers interested to understand the Fabric Attach (FA) technology for test and demo proposes. Why does it make sense to use FA? It's because it automates the configuration at the edge of the network. Shortest Path Bridging (SPB) addresses already the simplification in the Data Center and Campus. The combination of SPB and FA becomes a perfect fit for a flexible and easily managed network access environment.

The idea of Fabric Attach in the Enterprise

Having simplified the network core and automated it with Fabric Connect (FC) Extreme Networks recognized that solving the attach problem in the campus with devices such as IP CCTV camera's or Wireless Access points would provide immense operational and business benefits.

Extreme Networks solved the problem by designing Fabric Attach (FA), a technology that enables an edge device, be it a virtual machine, am Fabric Engine, Switch engine, ISW, Open vSwitch, IP CCTV camera, Wireless Access point or even a legacy switch unable to support SPB to automatically attach to a virtual service network (VSN) without any human intervention.



Fabric Attach, known also as auto-attach as it becomes standardized in IEEE 802.1Qcj, uses simple LLDP messages for the device to request attachment to a specific L2 VSN. The Fabric Connect network with the Fabric Attach Server function then ensures the provisioning as needed.

Prerequisites

There are two choices available.

- Open Virtual Appliance (OVA) runnable under
  VMware ESX 6.x or later, VMware Workstation/Player and VirtualBox.
  the VM requires 1 GB RAM, 1 CPU core, 8 GB disk space

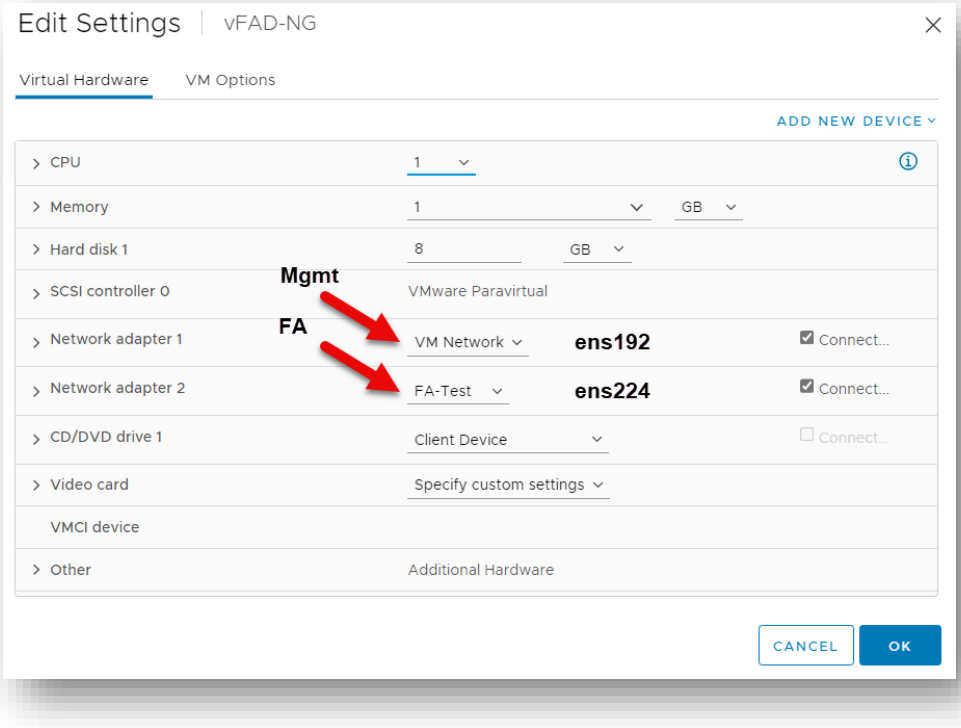
- GNS3 network simulator image
  Loadable image for the network simulator for testing FA functionality in combination with VOSS and EXOS switch simulation.

All options require two physical Ethernet ports; **Management-NIC** for controlling the device and **FA-NIC** for offering FA functionality.

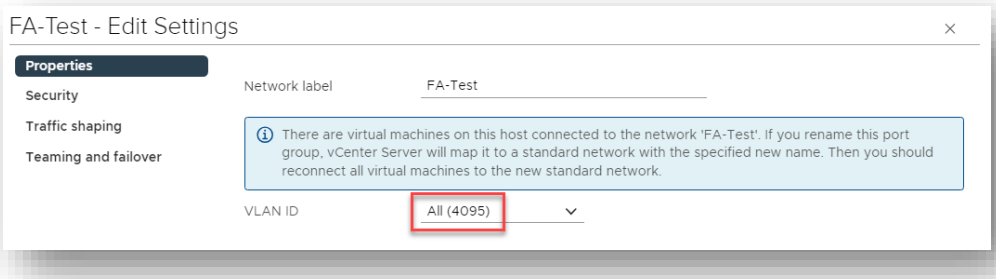
2. vFAD-NG setup

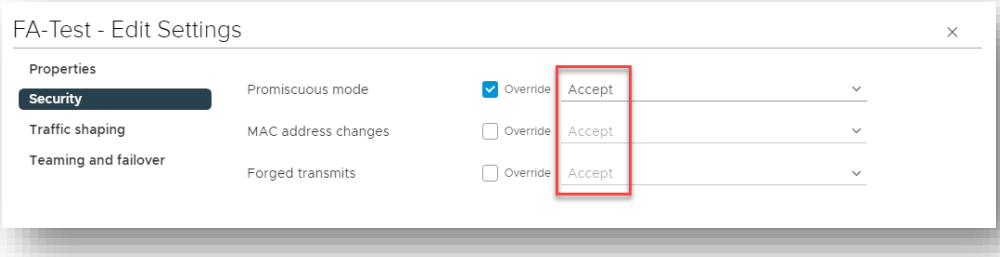
3. configuring VMware

The VMware Virtual Machine should be configured like this



It is required to manage the vSwitch where **FA** NIC is assigned to, allow promiscuous mode as well as forged transmits otherwise the vFAD will not be able to operate as required.

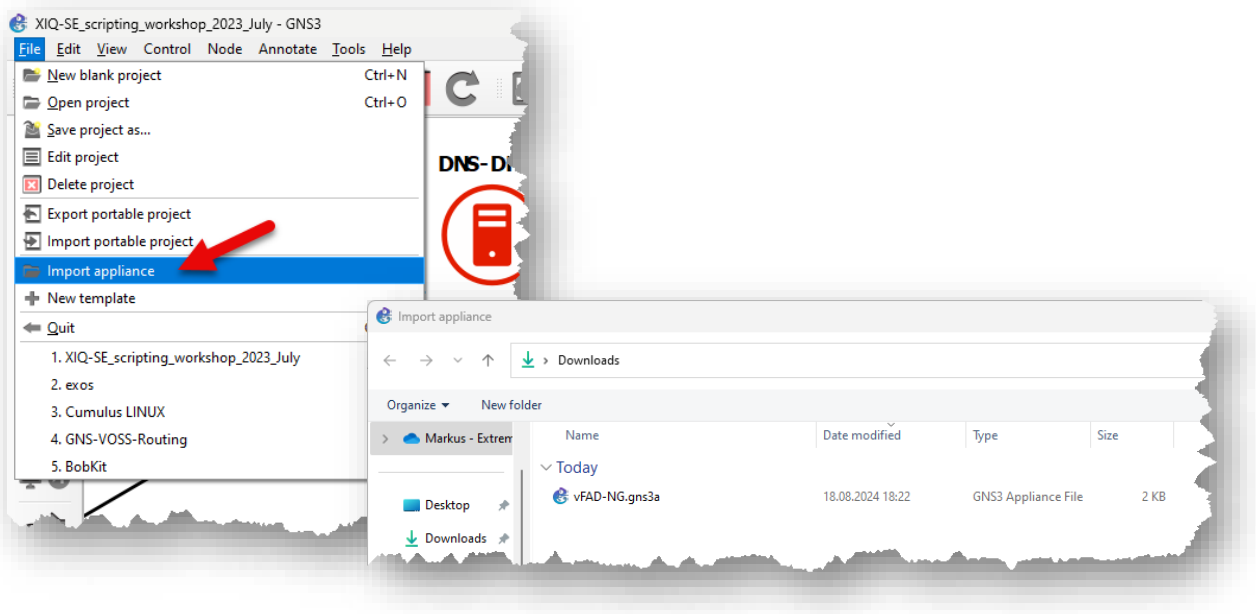


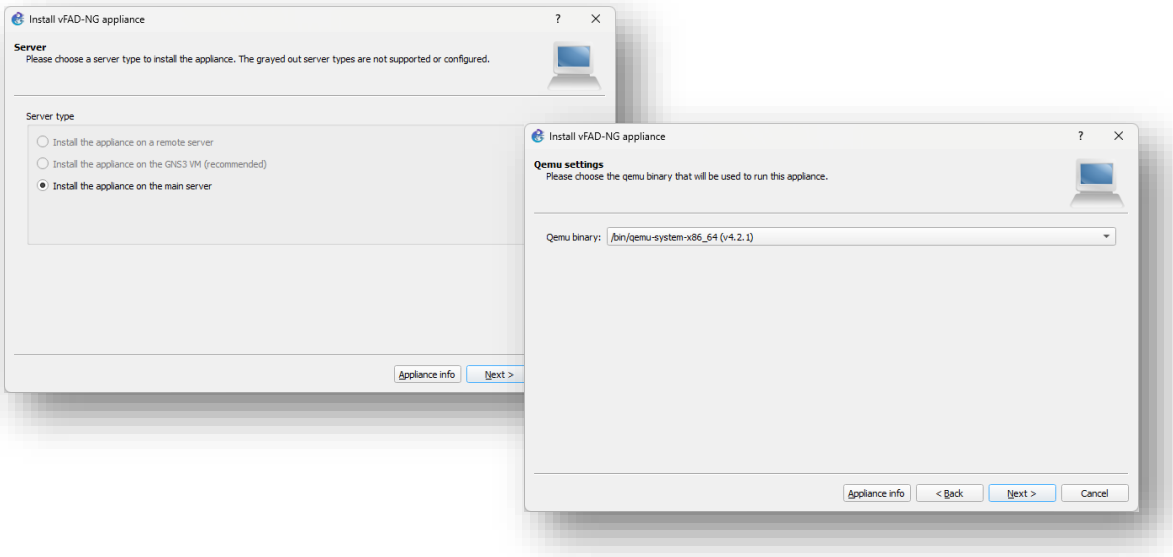


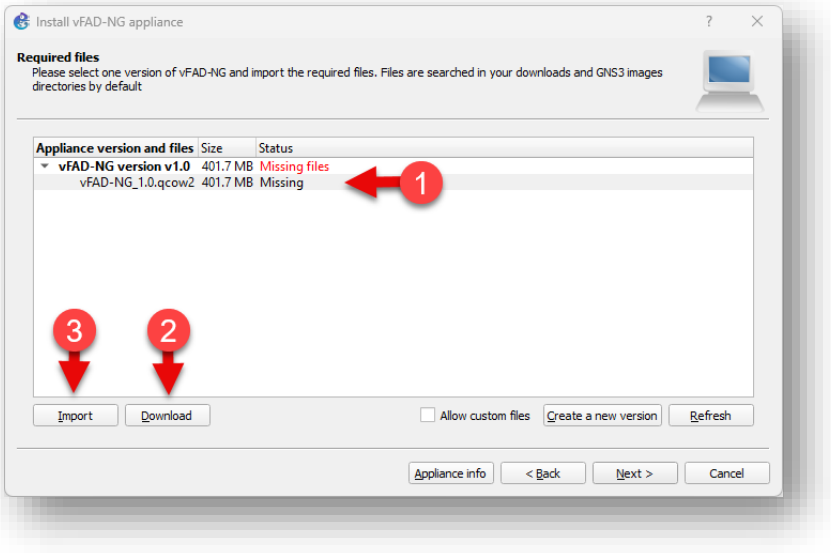
4. configuring GNS3

The GNS3 network simulator allows to simulate network devices and build topologies. Since VOSS 7.0 is it possible to host a SPB fabric in GNS3. With VOSS fabric border nodes as well EXOS switches is possible to attach a vFAD to simulate the VLAN/I-SID signaling. The test was mad based on GNS3 release 2.2.46.
You can get the GNS3 software from https://www.gns3.com/. You have to download as well the corresponding vFAD image from http://nikulski.net/vfad/ WEB site. The files have a **.gns3** as well **.qcow2** extension. If you installed the GNS3 management software as well deploy a GNS3 VM you should follow the setups to put in place the vFAD.

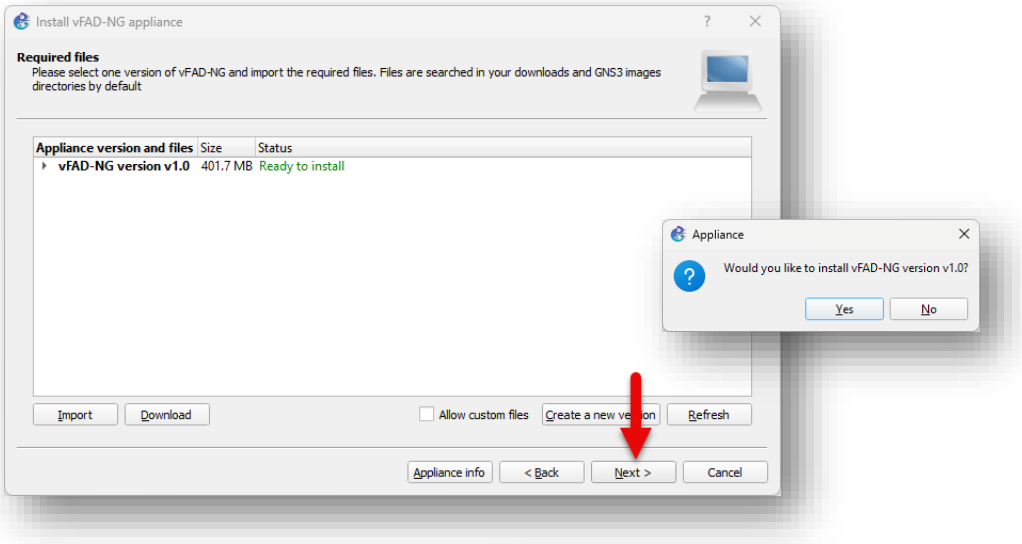
Open your GNS3 client and import the appliance file with the extension **.gns3** like this



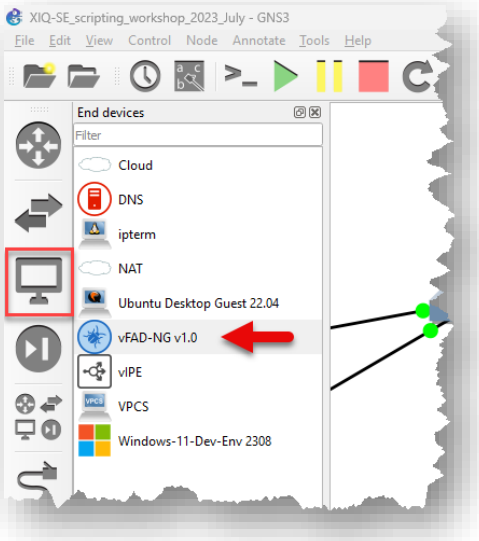




Select the missing qcow2 image file and download (2) if you not have already the image. If the image is downloaded, your import (3) the file. If the import process is completed you should see now that the vFAD-NG appliance is ready to install.



Klick next to make the appliance available in GNS3 under end devices section.



5. vFAD-NG Setup

Start the VM. The console and SSH login is **fa** / **fa** .

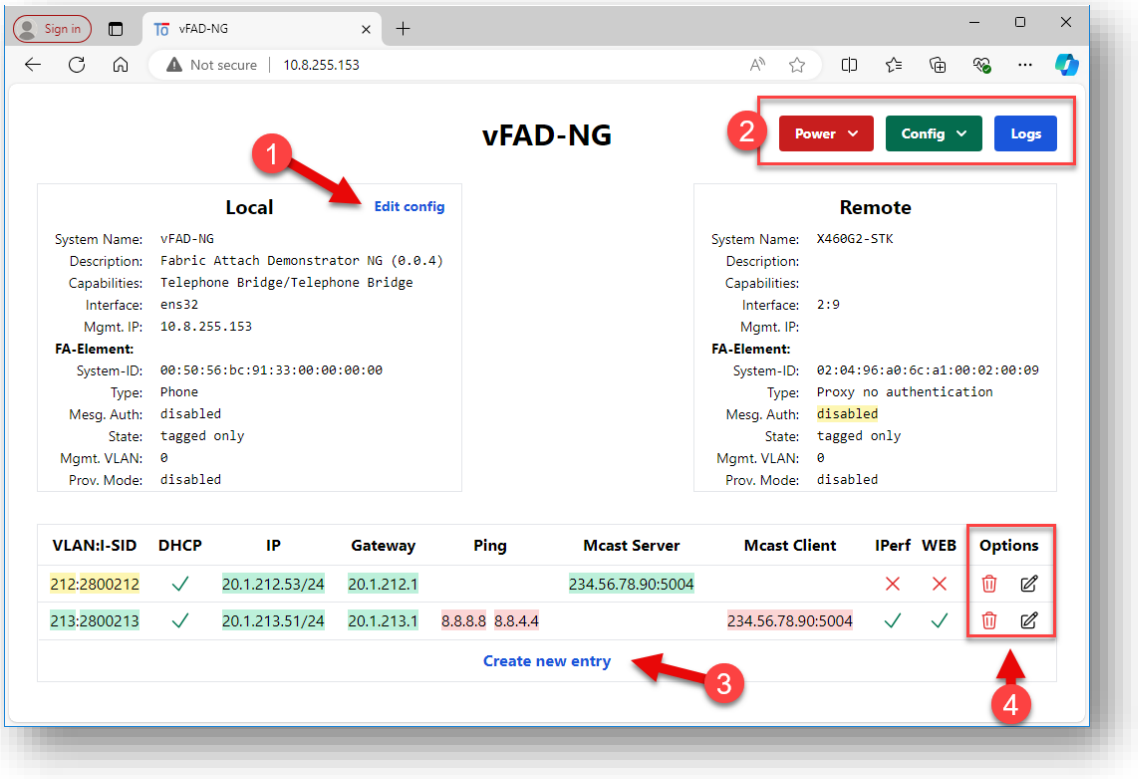
If required to change to a static IP address, you have to adapt the LINUX config file **/etc/network/interfaces** . For the yellow part you have to add the **#** sign to deactivate DHCP. The green part you have to remove the leading **#** like is shown below.

```
sudo -s

nano /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
#iface ens192 inet dhcp

# static IP example  for primary network interface
iface ens192 inet static
        address 1.2.3.4/24
        gateway 1.2.3.1
        dns-nameservers 8.8.8.8
        dns-search example.local

# The secondary network interface
allow-hotplug ens224
iface ens224 inet manual
```

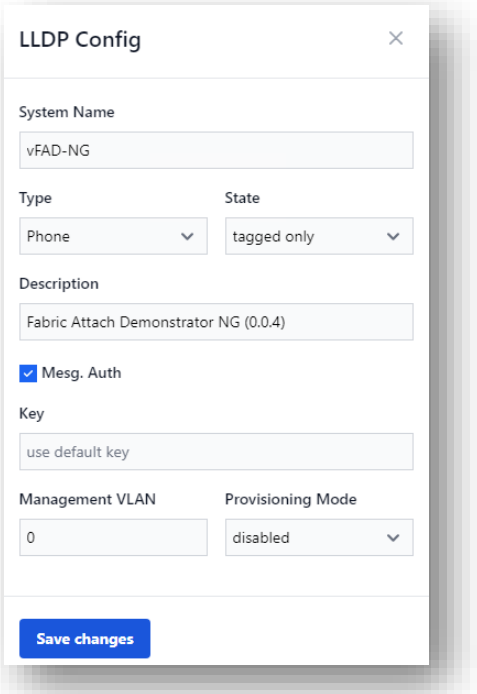
With **CTRL-X** and afterwards **y** and afterwards **ENTER** you can leave the editor. A reboot is required to make the change active.

```
init 6
```

You should now be able to access the management interface by using your preferred WEB browser pointing to the IP address assigned to the Management NIC of VM using **http** (not https).

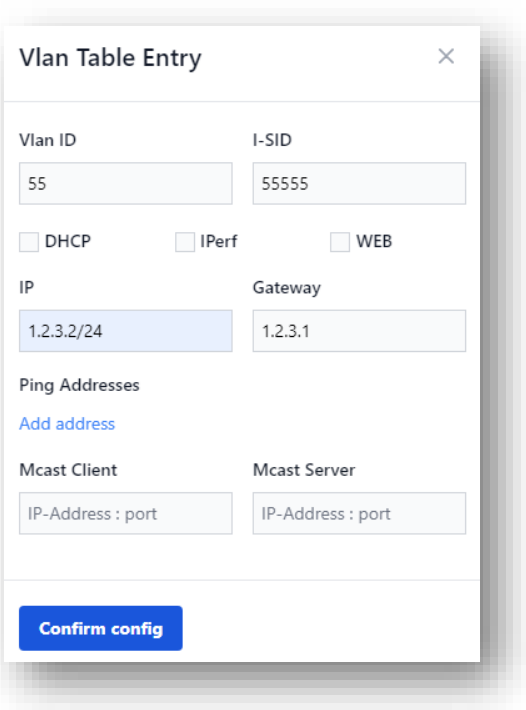


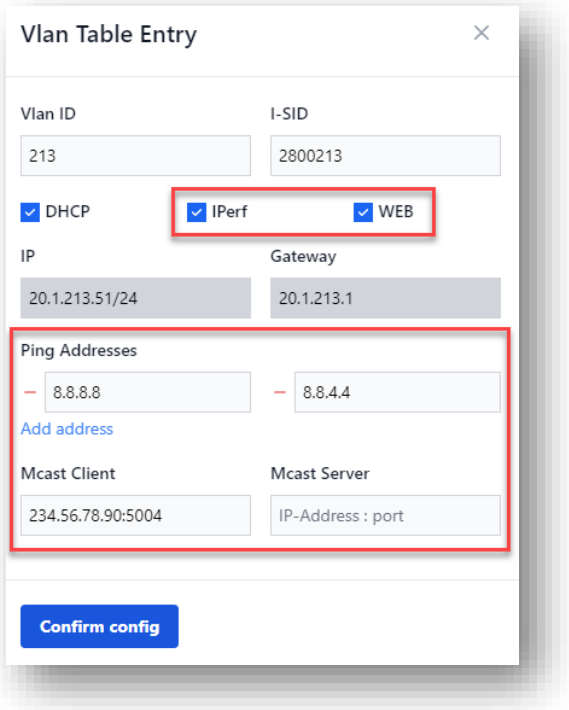
If you klick on left upper area (**1**) you are able to configure wha the vFAD will propagat by **LLDP** and **FA**.



The button block (2) on the right upper corner allows you to control the vFAD appliance.

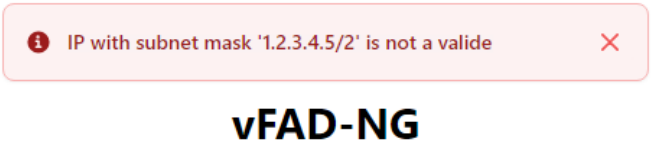
For adding instance of VLAN / I-SID connectivity you just click on the **Create new entry** link (3) to get the following dialog box.



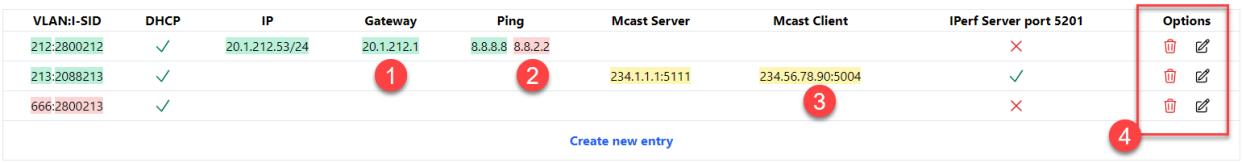


The red angled areas are optional.

In case the input data are malformed or missing, you will see the error message on the top middle like this:



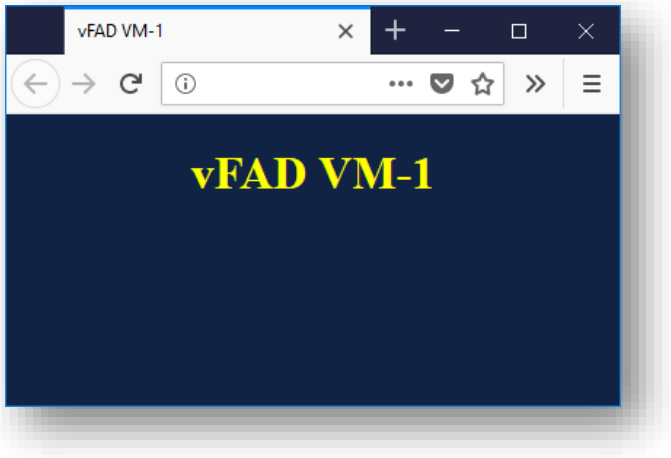
vFAD-NG



The status are colored and will be updated multiple time per seconds. The color have following meaning:

212:2800212	accepted
444:444	unknown
666:2800213	rejected
20.1.212.1	is ping able
8.8.2.2	ping timeout
234.56.78.90:5004	wait for multicast stream
234.56.78.90:5004	receive the multicast stream

Each VM instance owns an IP address and network mask with a corresponding default gateway. This IP address can be pinged, and if the WEB server enabled you get access to a WEB page like this.



The **IPerf** server works in a similar way like the WEB server using the default TCP port. You can use a IPerf3 client like this to generate traffic between your client and the VM instance like this

```
iperf3 -c 20.1.212.53
```

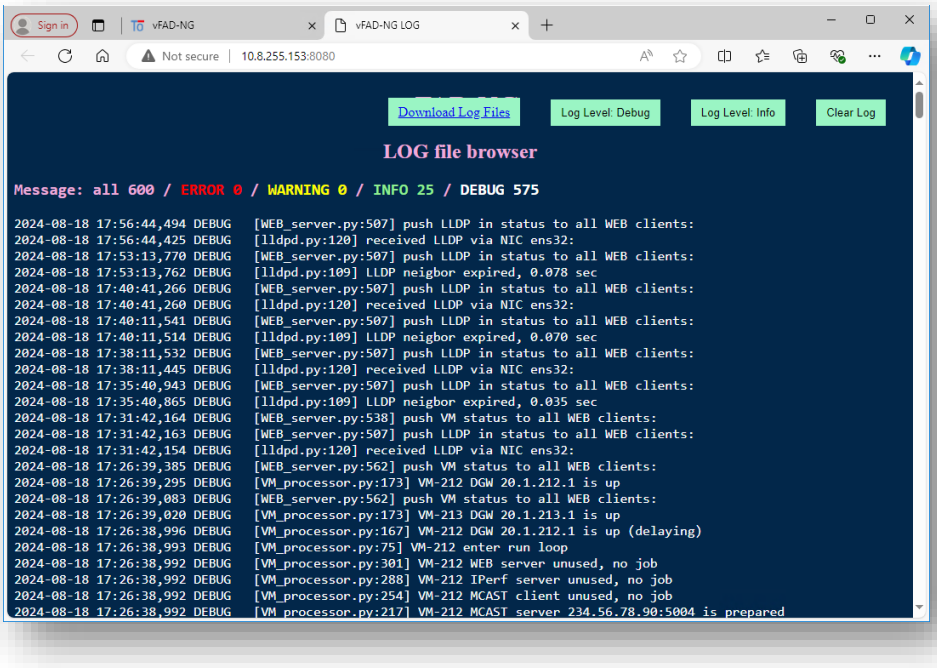
6. Troubleshooting

You can also SSH to the same IP with username **fa** and password **fa** too. The root access isn't enabled via SSH but you can activate it by logging in first with the fa user and then changing to super user access by calling **sudo -s** command and providing the same **fa** password.

6.1. vFAD

The configuration, status and LOG information can be access via the WEB interface as well via CLI. The login via SSH is **fa** / **fa**. To grant root access privileges call **sudo -s** using the same password **fa**.

For troubleshooting the vFAD have a LOG file you can inspect. SSH in to the vFAD and call the command log. The default log level is INFO you can change to DEBUG. Just call the CLI command debug or info depends witch LOG level you like to have in place.



Please note, that verbose DEBUG LOG messages are not fully displayed. You have to download the logfile to get access to all the details.

1.1 LINUX OS

Through LINUX name space the IP stack is isolated for each VM.

```
ip netns list                                   # all existing
ip netns exec vm-<id> ping -c 1 -W 1 10.0.2.1   # ping from a VM
ip netns exec vm-<id> ip neighbor               # ARP cache from a VM
ip netns exec vm-<id> ip address                # NIC configuration
ip netns exec vm-<id> ip route                  # routes of a VM

htop                                            # show the OS utilization

# capture traffic
tcpdump -i eth1                                 # all traffic
tcpdump -i eth1 arp                             # ARP
tcpdump -i eth1 icmp                            # ping
tcpdump -i eth1 -vv ether proto 0x88cc          # LLDP

debug                                           # enable debug LOG
info                                            # disable debug LOG
tail -n 100 /var/log/vFAD.log                   # logfile
```