HOW-TO-GUIDE:
demonstrating Fabric Attach using Open vSwitch

1. Target audience

System Engineers interested to understand the Fabric Attach (FA) technology and/or for demo proposes.

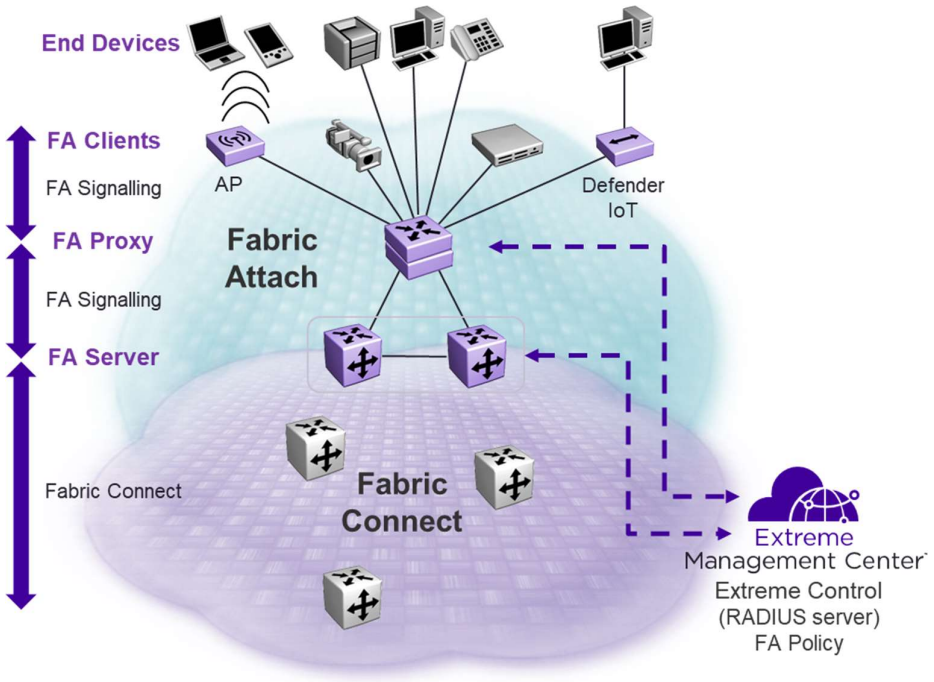
Why does it make sense to use FA? It's because it automates the configuration at the edge of the network. Shortest Path Bridging (SPB) addresses already the simplification in the Data Center and Campus. The combination of SPB and FA becomes a perfect fit for a flexible and easily managed network access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

2. The idea of Fabric Attach in the Enterprise DC

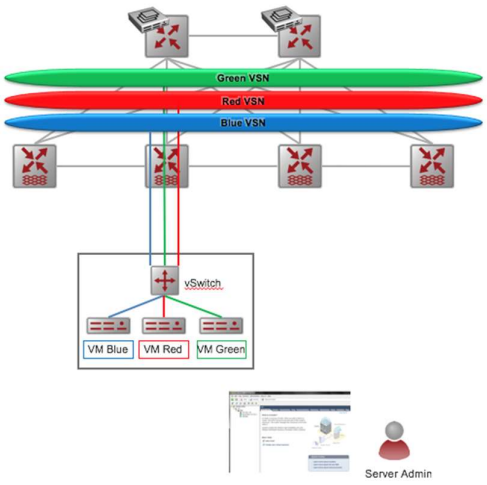
Having to manually configure Data Center (DC) ToR edge ports for hypervisors is undesirable, costly and error prone. While pre-configuring ports with multiple VLAN's is a security breach waiting to happen.

Having simplified the network core and automated it with Fabric Connect (FC) Extreme Networks recognized that solving the attach problem in DC's but also 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, a vSwitch, a IP CCTV camera, a 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. In hypervisor environments this allows a VM in a hypervisor to request and attach to a virtual service network (VSN) and gain connectivity without any human intervention. If the VSN requested doesn't exist and policy permits, the VSN can even be created "on the fly". Once the VM is moved, the attach process repeats itself, the VSN connectivity moves with the VM. The connectivity gets established at the new ToR location and removed at the original, thus simplifying operation and configuration management.



- **Single Infrastructure layer**, single protocol, **automated core** network, sub-second failover, full use of network capacity through ECMP and full topological flexibility
- **Automated edge attach** and configuration management, moving control to the edge and simplifying operation and configuration management
- **No** need for **complex overlays**
- **No** need for complex and **expensive synchronization software**
- **Deploy once and forget**; no more configuration required on edge and core switches with FC/FA
- **Powerful orchestration** with Avaya Fabric Orchestrator
- **SDN Ready** through Avaya SDN Fx
- **A fully automated, virtualized DC networking architecture**

In hypervisor environments this is achieved by enabling the ***Open vSwitch (OvS)*** to perform Fabric Attach (or auto attach in standards terms). As of OvS version 2.4 released since 2015, the vSwitch now supports FA.

3. Prerequisites

There are three choices available.

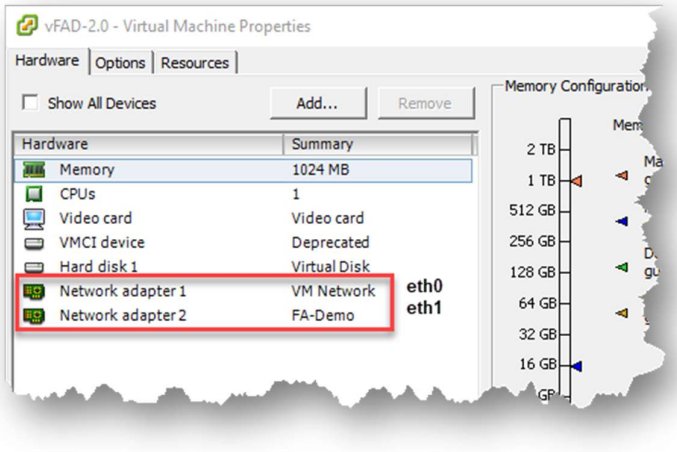
- Open Virtual Appliance (OVA) runnable under
  VMware ESX 6.x or later, VMware Workstation/Player and VirtualBox.
  the VM requires 1 GB RAM, 1 CPU core, 2 GB disk space

- GNS3 network simulator image
  Loadable image for the network simulator for testing FA functionality in combination with VOSS and EXOS switch simulation. The implementation is tested against GNS3 release is 2.1.8

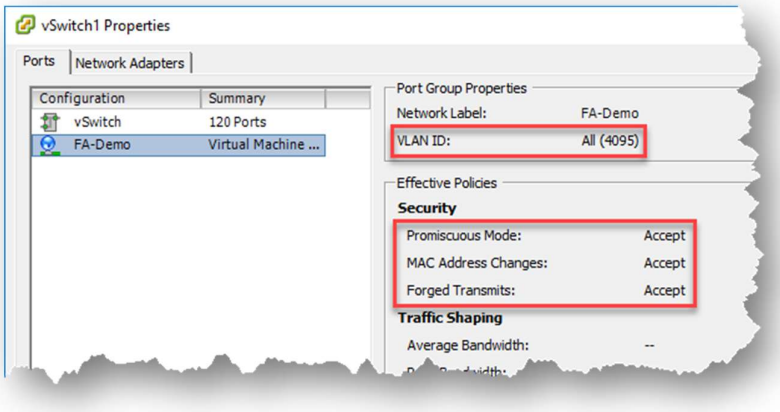
- Image for Raspberry Pi 3 model B
  The image fit for a 4GB SD card. It's recommended to use an SD card supporting class 10 or faster IO speed. Any other hardware models are not supported.

All options require two physical Ethernet ports; **eth0** for controlling the device and **eth1** for offering FA functionality.

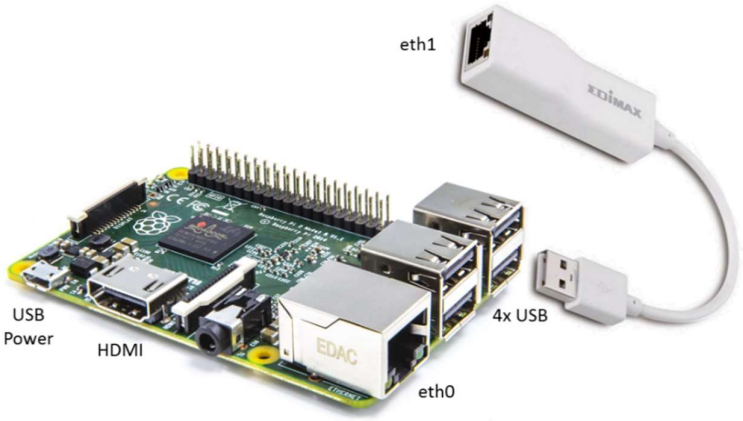
The VMware Virtual Machine should be configured like this



It is required to manage the vSwitch where **eth1** is assigned to, allow promiscuous mode as well as forged transmits otherwise the vFAD will not be able to operate as needed.



The Raspberry Pi 3 model B offers only one physical Ethernet port, therefore the second need to be provided via an additional USB-to-Ethernet adapter. An adapter supporting 10/100 Base-T is Suffusion.



4. Limitations

Please be aware that not more than 94 VLAN/I-SID definition can be made with Fabric Attach, because of the length of the LLDP TLV definition. Hence, the vFAD is limiting the amount of VM instances to 94.

More than one VM can use the same VLAN/I-SID combination. If the VLAN is the same but the I-SID is different, or vice versa, then it is considered as wrong configuration and will be blocked.

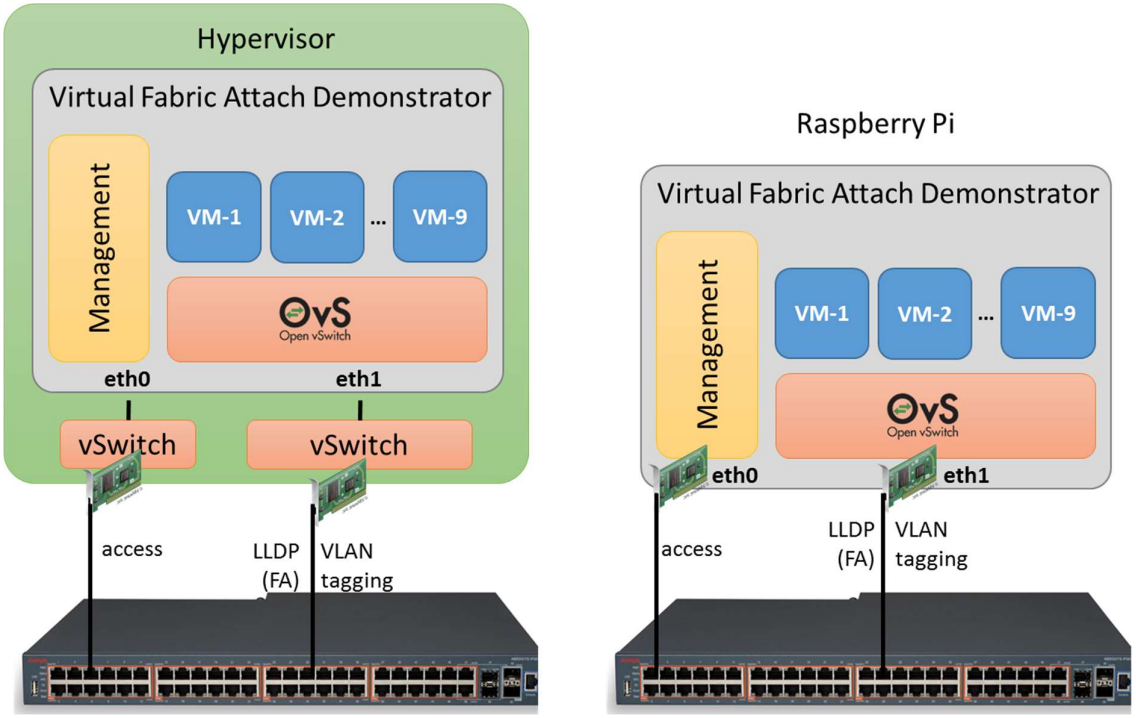
A VM instance can ping any IP address except its own one. This is a specific behavior of the simulator only.

Only Raspberry Pi 3 model B is supported. Please be aware that the vFAD image will not boot on any other Raspberry Pi models!

In case you are experienced with viFAD 1.x already, the vFAD 2.0 release is not using Docker container virtualization anymore. It used the LINUX IP kernel implementation, which allows to create virtual IP net spaces (virtual instances). Services like WEB server are bonded to the corresponding IP net space. However, is not a true Virtual Machine (VM) like you know from classical virtualization technologies.

5. Setup

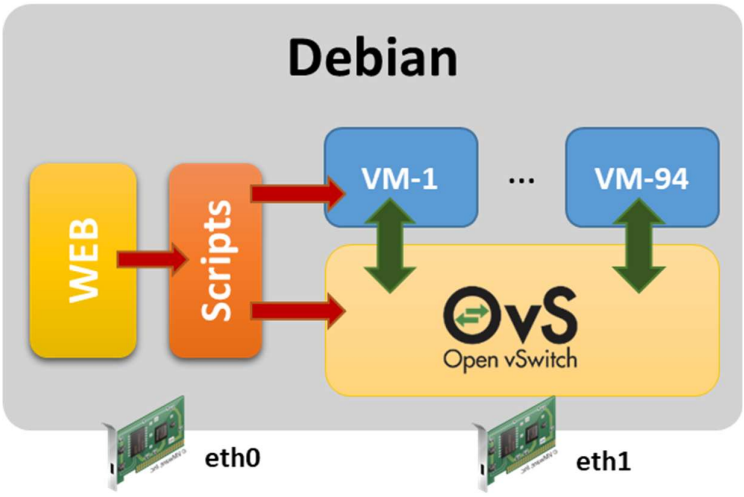
The supported topology is like this



1.1 Configuring management IP connectivity

The system is already preconfigured for ease of use. Boot up VM / Raspberry Pi and connect to the console and login with user name **fa** and password **fa**
(same password is used for **root** access).

Following diagram demonstrates the internal structure:



By default, DHCP is used to assign an IP address to the management interface; check IP address of eth0 with

```
sudo ifconfig eth0
```

If no DHCP service is available, static configuration can be applied as well. You can manually modify the related configuration files:

```
sudo nano /etc/network/interfaces
```

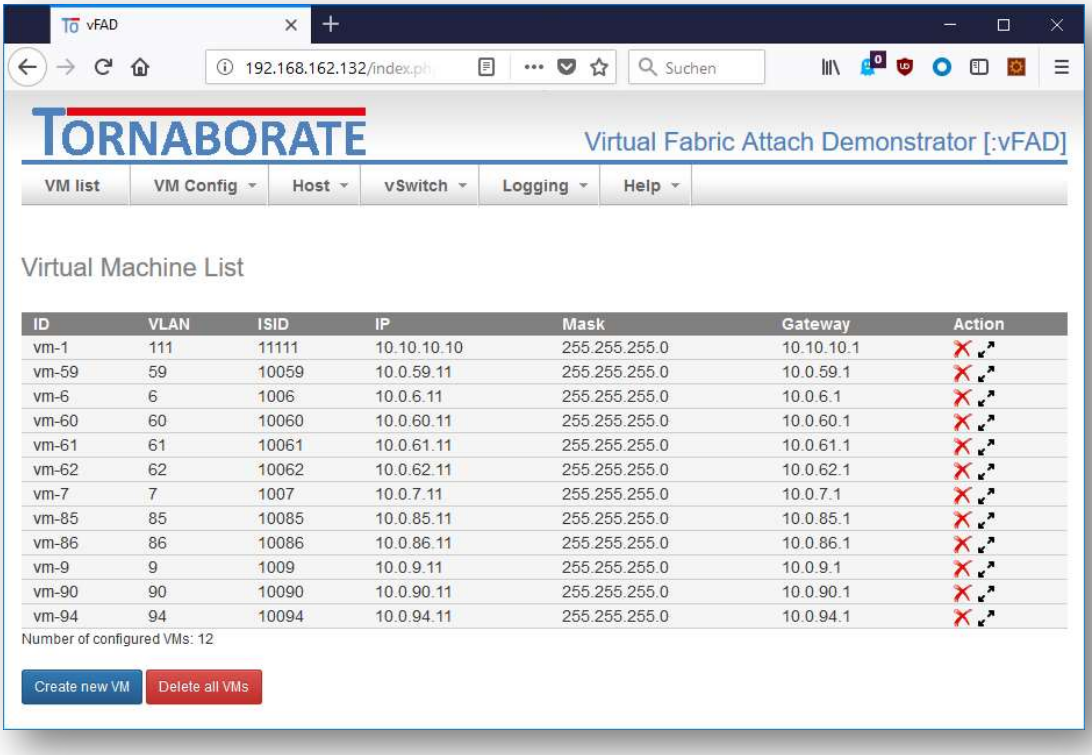
The configuration file should look as follows:

```
iface eth0 inet static
      address 192.168.1.100
      netmask 255.255.255.0
      gateway 192.168.1.1
      dns-nameservers 8.8.8.8 8.8.4.4
```

After changes are saved (CTRL-X / y / ENTER), please restart the network service.

```
systemctl restart systemd-networkd.service
```

You should now be able to access the management interface by using your preferred WEB browser pointing to the IP address assigned to **eth0** of VM or Raspberry Pi



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 **su** command and providing the same **fa** password.

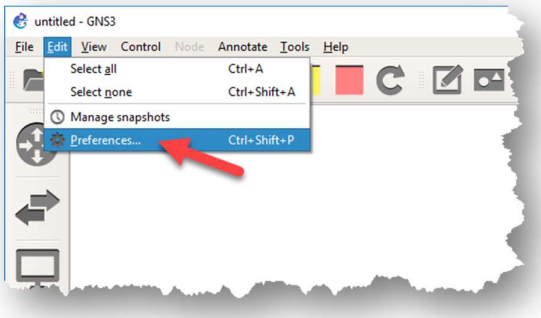
Each VM instance owns an IP address and network mask with a corresponding default gateway. This IP address can be pinged, and is already configured to act as a WEB server. No other service is offered from the VM. The WEB page of the VM looks like this.

To make the VM reachable as soon as possible, each VM performs an ARP reply against its own MAC address every second. This will make sure that the connected switch gets the MAC address forwarding database populated.

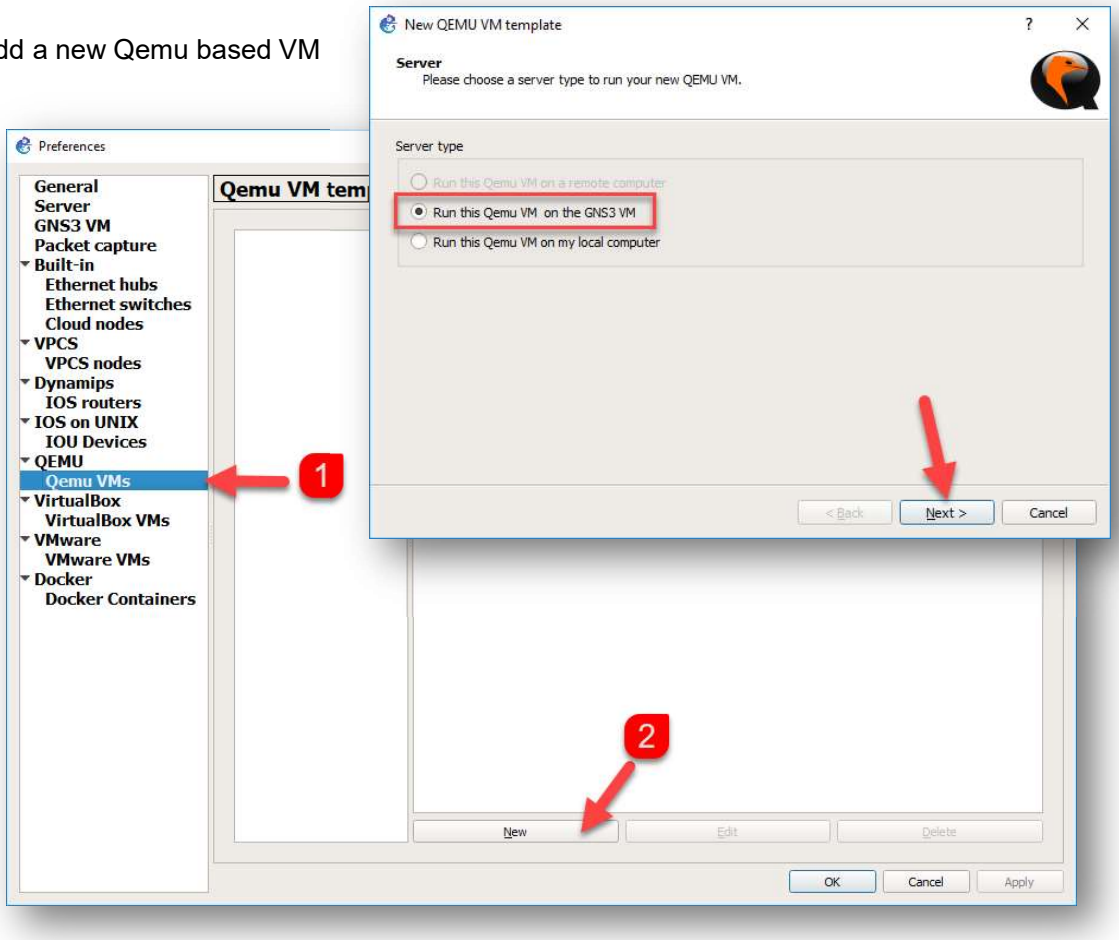
1.2 configuring GNS3

The GNS3 network simulator allows to simulate network devices and build topologies. Since VOSS 7.0 is it possible to simulate a SPB fabric. 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 2.1.8. You can get the software from https://www.gns3.com/. You have to download as well the corresponding vFAD image from http://www.tornaborate.net/ WEB site. The file should have a **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.

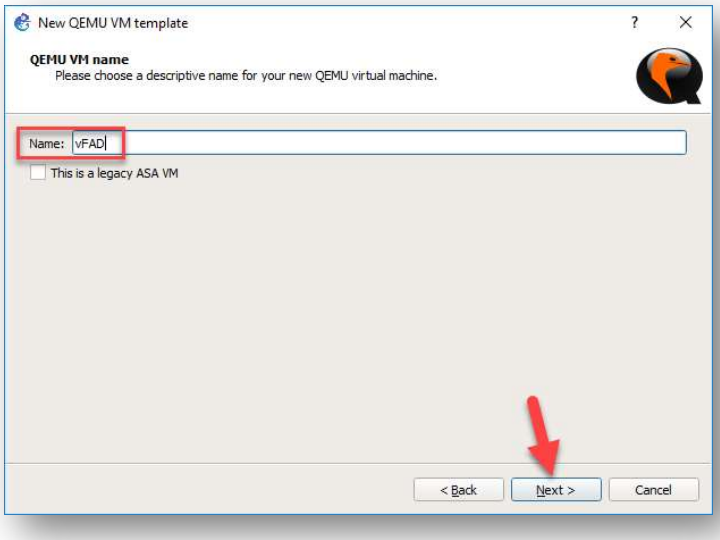
Open the preference panel



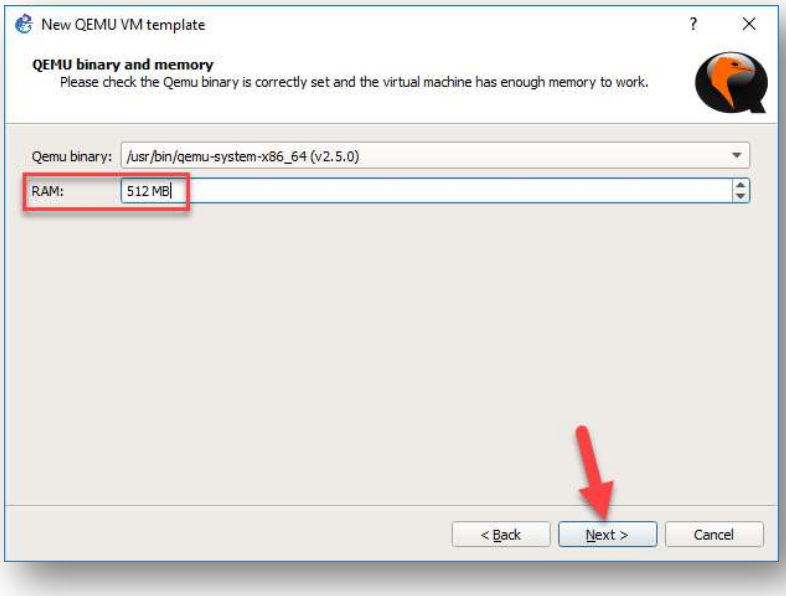
Add a new Qemu based VM



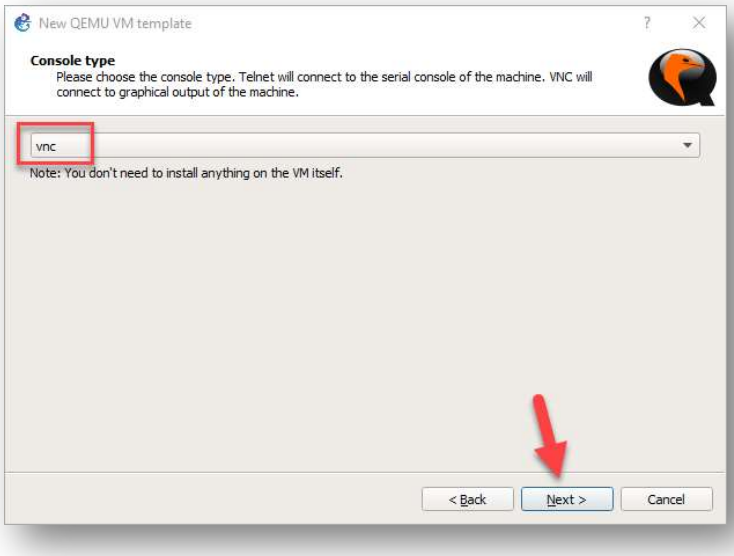
Provide a proper name like **vFAD** for the VM template.



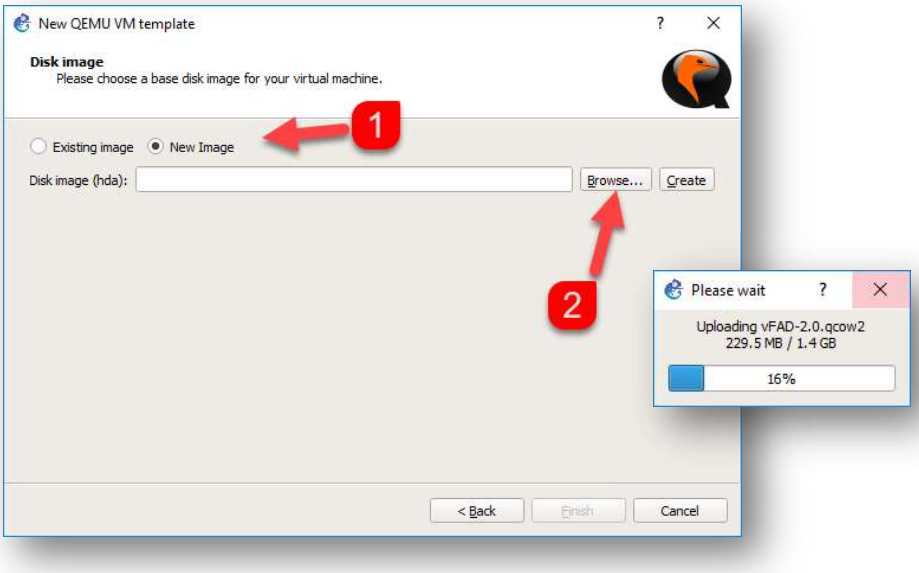
Assign at least 512 MB RAM to the VM template. But more than 1GB RAM makes no sense because the VM will not allocate the additional RAM.



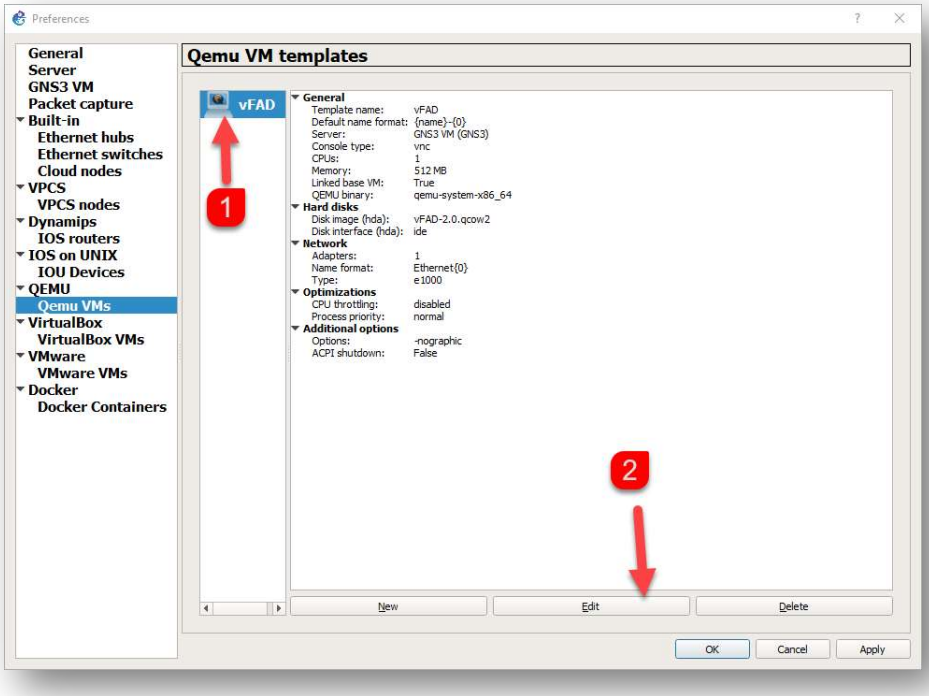
For the console type please use VNC.



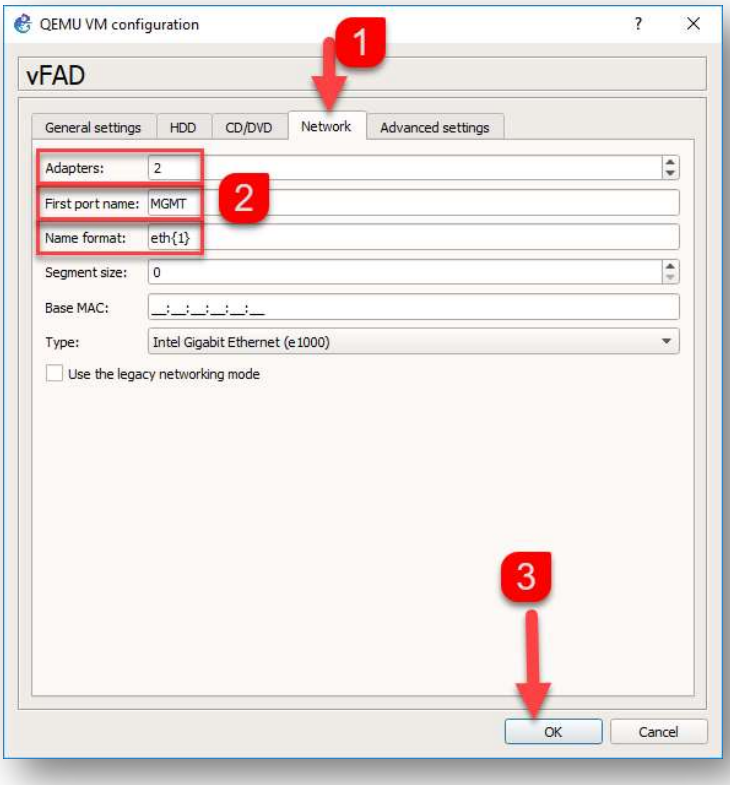
A new Disk image should be loaded like this. Please pick the **vFAD-2.0.qcow2** file.



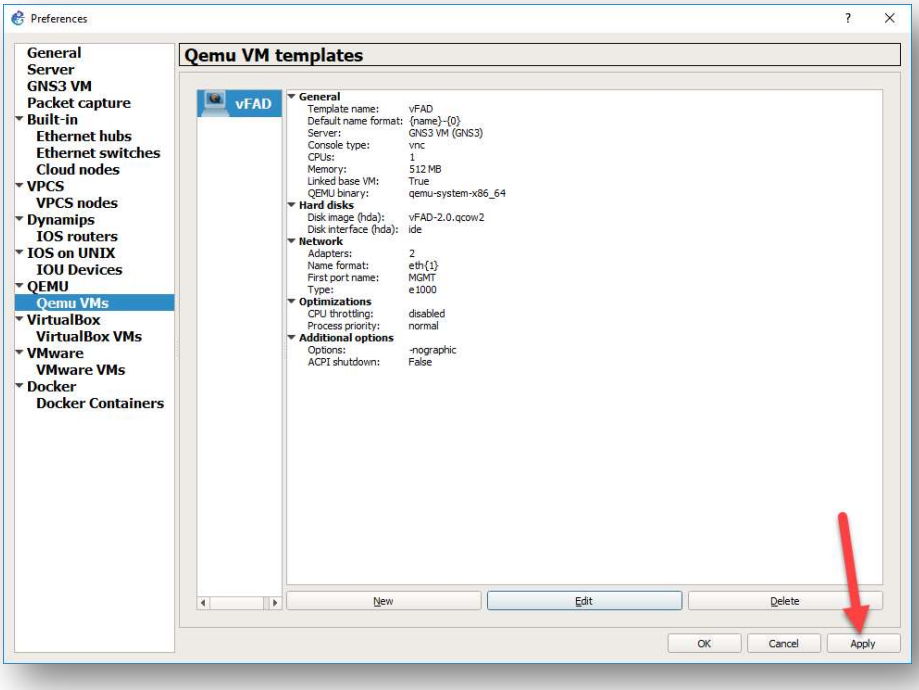
Since the image is loaded complete the dialog and reopen the settings of the template.



and change the network settings like this



Please don't for get to apply all the changes



1.3 BOSS setup

Following switches with minimum software load can be used as FA Proxy:

- ERS 3510 release 5.4
- ERS 4500 release 5.7.3
- ERS 4800 release 5.11
- ERS 4900 release 7.4
- ERS 5900 release 7.4
- VSP 7000 release 10.4

Following switches can be used as FA server:

- ERS 4800 Release 5.11
- ERS 4900 Release 7.4
- ERS 5900 Release 7.4

1.3.1 Configuring an BOSS FA Proxy

```
no fa message-authentication port <port>
fa extended-logging
```

1.3.2 Configuring a BOSS FA server

SPB must be configured up front on VOSS switch to be able to use the FA capabilities.

```
fa port <port>
no fa message-authentication port <port>
fa extended-logging
```

1.4 VOSS setup

The VSP switches supports FA Sever functionality since VOSS release 5.0.0 on following switches:

- VSP 4000
- VSP 7200
- VSP 8200
- VSP 8400

1.4.1 Configuring VOSS FA Server

SPB must be configured up front on VOSS switch to be able to use the FA capabilities.

```
interface <port>
     fa enable
exit
```

1.5 XOS setup

Extreme XOS switches support FA Proxy since release 22.4.1.

There is nothing to configure on XOS for it to run FA Proxy. Currently XOS does not support FA message authentication and FA is always enabled on all ports.

6. Troubleshooting

1.6 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 **su** using the same password **fa**.

1.6.1 Control Script

A script called **run.pl** is in place to control all the required changes. Make sure you have root access level (login via SSH with user **fa** and password **fa** and change to root access level by using **su -** with the same password **fa**)

```
run.pl                                                  # syntax of the script
run.pl ping –name vm-1 –ip 8.8.8.8                      # ping from a VM
run.pl traceroute –name vm-1 –ip 8.8.8.8                #
tail -n 100 /var/www/run.log                            # script LOG records
/root/show.sh                                           # display all information
/root/test.sh                                           # create 94 VM instances

run.pl restore   # clear all running VMs and recreate all VMs based on the config file
run.pl reset     # clear all running VMs and purge the config file
```

1.6.2 OpenVSwitch

```
ovs-vsctl show                                          # OVS config
ovs-appctl autoattach/status                            # LLDP status
ovs-appctl autoattach/show-isid                         # FA status
ovs-appctl fdb/show br-int                              # MAC forwarding DB

tail -n 100 /var/log/openvswitch/ovs-vswitchd.log
```

1.6.3 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

ss -lnt                                         # all TCP listener

# 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

tail -n 100 /var/log/syslog                     # system LOG
tail -n 100 /var/log/daemon.log
```

1.7 BOSS

```
show fa agent                                          # switch status
show fa elements                                       # FA neighbors (obsolete)
show fa assignment                                     # FA neighbors
show fa vlan                                           # only for FA standalone proxy
show vlan
show lldp port <port>
show lldp neighbor vendor-specific avaya fabric-attach
show logging sort-reverse
show running-config module fa                          # FA configuration
```

1.8 VOSS

```
show fa agent
show fa interface
show fa elements
show fa assignment
show i-sid
show lldp port <port>
show log file
```

1.9 XOS

```
show fabric attach agent
show fabric attach elements
show vlan fabric attach assignments
show lldp ports <port> neighbors detailed
show log
```