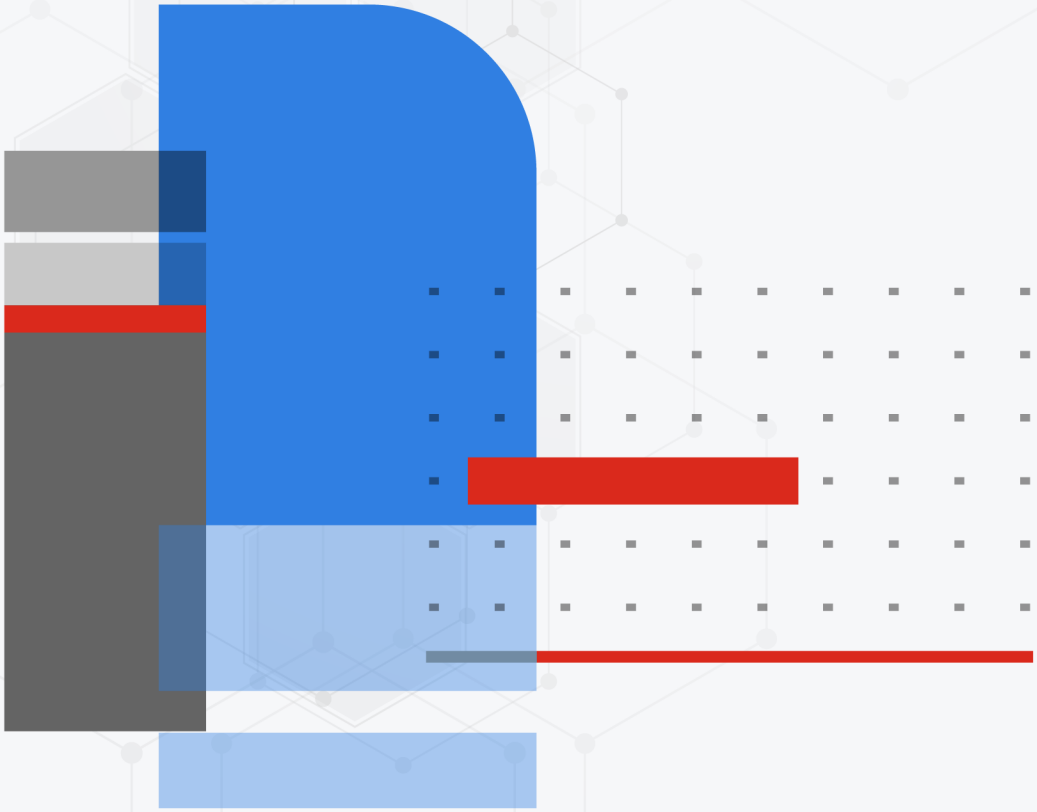
VMware ESXi Administration Guide

FortiOS 7.4



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

May 11, 2023
FortiOS 7.4 VMware ESXi Administration Guide
01-740-911162-20230511

TABLE OF CONTENTS

About FortiGate-VM on VMware ESXi

FortiGate-VMs allow you to mitigate blind spots by implementing critical security controls within your virtual infrastructure. They also allow you to rapidly provision security infrastructure whenever and wherever it is needed. FortiGate-VMs feature all the security and networking services common to hardware-based FortiGate appliances. You can deploy a mix of FortiGate hardware and VMs, operating together and managed from a common centralized management platform.

This document describes how to deploy a FortiGate-VM in a VMware ESXi environment.

FortiGate-VM models and licensing

FortiGate-VM offers perpetual licensing (normal series and v-series) and annual subscription licensing. See VM license for details.

After you submit an order for a FortiGate-VM, Fortinet sends a license registration code to the email address that you entered on the order form. Use this code to register the FortiGate-VM with Customer Service & Support, then download the license file. After you upload the license to the FortiGate-VM and validate it, your FortiGate-VM is fully functional.

FortiGate-VM evaluation license

The FortiOS permanent trial license requires a FortiCare account. This trial license has limited features and capacity. See VM permanent trial license for details.

FortiGate-VM virtual licenses and resources

The primary requirement for provisioning a FortiGate-VM may be the number of interfaces it can accommodate rather than its processing capabilities. In some cloud environments, the options with a high number of interfaces tend to have high numbers of vCPUs.

FortiGate-VM licensing does not restrict whether the FortiGate can work on a VM instance in a public cloud that uses more vCPUs than the license allows. The number of vCPUs that the license indicates does not restrict the FortiGate from working, regardless of how many vCPUs the virtual instance includes. However, only the licensed number of vCPUs process traffic and management tasks. The FortiGate-VM does not use the rest of the vCPUs.

License	1 vCPU	2 vCPU	4 vCPU	8 vCPU	16 vCPU	32 vCPU
FGT-VM08	OK	OK	OK	OK	The FortiGate-VM uses 8 vCPUs for traffic and management and does not use the rest.	The FortiGate-VM uses 8 vCPUs for traffic and management and does not use the rest.

You can provision a VM instance based on the number of interfaces you need and license the FortiGate-VM for only the processors you need.

Public compared to private clouds

The behavior differs between private and public clouds:

- Private clouds (VMware ESXi/KVM/Xen/Microsoft Hyper-V): both licensed vCPUs and RAM are affected. FortiOS does not have licensed RAM size restrictions. However, the minimum recommended RAM size is 2 GB for all versions.
- Public clouds (AWS/Azure/GCP/OCI/AliCloud): only licensed vCPU is affected.

For example, you can activate FG-VM02 on a FGT-VM with 4 vCPUs and there is no limit on the RAM size when running on a private VM platform.

Likewise, you can activate FG-VM02 on a FGT-VM c5.2xlarge EC2 instance with 8 vCPUs running on AWS. Only 2 vCPU is consumable, and there is no limit on the RAM size. You can refer to licenses for public clouds as bring your own license.

VMware ESXi certification information

The following summarizes FortiGate-VM 7.4 certification information on VMware ESXi:

Version	Partner product and version	Certification date	Validity	Listing
7.2.1	Partner Ready - vSphere	August 2022	N/A	https://marketplace.cloud.vmware.com/services/details/fortigate-next-generation-firewall-7-2-0-1?slug=true

Preparing for deployment

This documentation assumes that before deploying the FortiGate-VM on the VMware ESXi virtual platform, you have addressed the following requirements:

Virtual environment

You have installed the VMware ESXi software on a physical server with sufficient resources to support the FortiGate-VM and all other VMs deployed on the platform.

If you configure the FortiGate-VM to operate in transparent mode, or include it in a FortiGate clustering protocol high availability cluster, configure any virtual switches to support the FortiGate-VM's operation before you create the FortiGate-VM. See Transparent mode on page 16 or HA on page 17.

Management software

You can access the VMware vSphere in one of the following ways:

- Directly via the ESXi web GUI
- Via the vSphere Web Client if a vCenter is managing the ESXi server

Connectivity

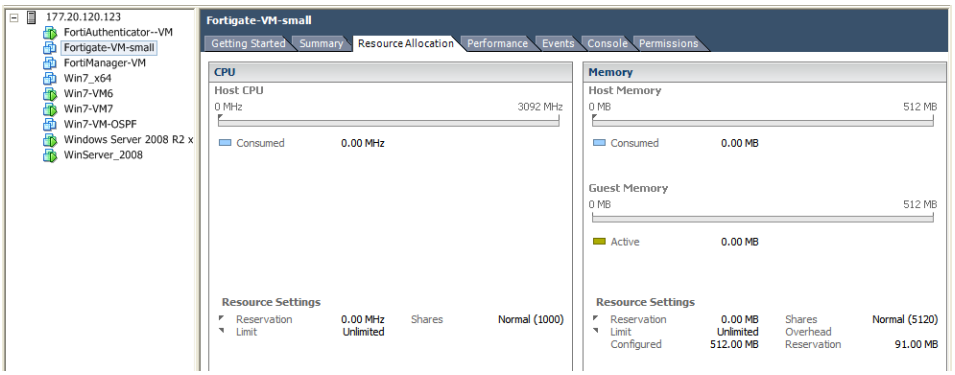
The FortiGate-VM requires an Internet connection to contact FortiGuard to validate its license. A FortiGate-VM in a closed environment must be able to connect to a FortiManager to validate the FortiGate-VM license. See Validating the FortiGate-VM license with FortiManager on page 14.

Configuring resources

Before you start the FortiGate-VM for the first time, ensure that you have configured the following resources as the FortiGate-VM license specifies:

- Disk sizes
- CPUs
- RAM
- Network settings

To configure the resources for a FortiGate-VM deployed on VMware ESXi, use the vSphere client.



Registering the FortiGate-VM

Registering the FortiGate-VM with Customer Service & Support allows you to obtain the FortiGate-VM license file.

To register the FortiGate-VM:

1. Log in to the Customer Service & Support site using a support account, or create an account.
2. In the main page, under *Asset*, select *Register Now*.
3. In the *Registration* page, enter the registration code that you received via email, and select *Register* to access the registration form.
4. If you register the s-series subscription model, the site prompts you to select one of the following:
   a. Click *Register* to newly register the code to acquire a new serial number with a new license file.
   b. Click *Renew* to renew and extend the licensed period on top of the existing serial number, so that all features on the VM node continue working uninterrupted upon license renewal.
5. Complete and submit the registration form.
6. In the registration acknowledgment page, click the *License File Download* link.
7. Save the license file (`.lic`) to your local computer. See Uploading the FortiGate-VM license on page 13 or Validating the FortiGate-VM license with FortiManager on page 14 for information about uploading the license file to your FortiGate-VM via the GUI.

Downloading the FortiGate-VM deployment package

FortiGate-VM deployment packages are found on the Customer Service & Support site. In the *Download* drop-down menu, select *VM Images* to access the available VM deployment packages.

To download the FortiGate-VM deployment package:

1. In the *Select Product* drop-down menu, select *FortiGate*.
2. In the Select Platform drop-down menu, select VMware ESXi.
3. Select the FortiOS version you want to download.
   There are two files available for download: the file required to upgrade from an earlier version and the file required for a new deployment.

4. Click the *Download* button and save the file.

For more information, see the FortiGate datasheet.

You can also download the following resources for the firmware version:

- FortiOS Release Notes
- FORTINET-FORTIGATE MIB file
- FSSO images
- SSL VPN client

Deployment package contents

You must create a 32 GB log disk.

For supported VMware hardware versions, see Compatibility for VM hardware versions on page 10.

The FortiGate-VM deployment package contains the following components:

Component	Description
fortios.vmdk	FortiGate-VM system hard disk in VMDK format.
datadrive.vmdk	FortiGate-VM log disk in VMDK format.
readme.txt	Explains compatibility information for each template.
Open Virtualization Format (OVF) template files	
FortiGate-VM64.ovf	OVF template file for VMware ESXi 7.0 and later versions. vmxnet3-based.
FortiGate-VM64-ZTNA.vapp.ovf	OVF template file for VMware ESXi 7.0 and later versions. vmxnet3-based. Allows configuration of all ZTNA-related parameters during bootstrapping.
FortiGate-VM64.hw13.ovf	OVF template file for VMware ESXi 6.5 and later versions. vmxnet3-based.
FortiGate-VM64.hw15.ovf	OVF template file for VMware ESXi 6.7 and later versions. vmxnet3-based.
FortiGate-VM64.vapp.ovf	OVF template file for VMware ESXi 7.0 and later versions. SR-IOV is available on the adapters list after deployment with this template.
FortiGate-VM64.nsxt.ovf	OVF template file for VMware ESXi 5.0 and later versions. vmxnet3-based.

Use the VMXNET3 interface (FortiGate-VMxx.hw07_vmxnet3.ovf template) if the FortiGate-VM will distribute workload to multiple processor cores.

Compatibility for VM hardware versions

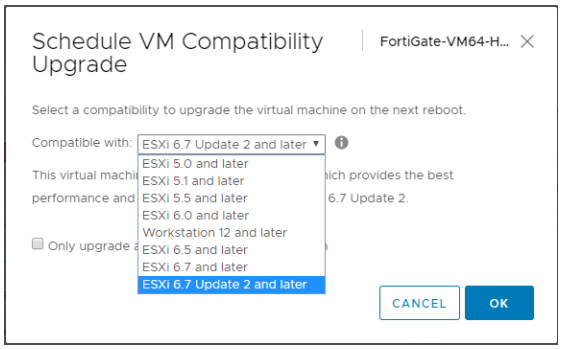
FortiGate-VM supports ESXi 5.5 and later versions. Using corresponding hardware versions 10 and later is highly recommended, as Virtual machine hardware versions mentions.

Upgrading hardware versions incrementally with only one delta at a time is recommended. For example, upgrading from 10 to 11, 11 to 12, 12 to 13, then 13 to 14 is recommended, although directly upgrading from 10 to 14 generally has no issues.

To check the FortiGate-VM ESXi compatibility, go to the VMware Marketplace listing.

To upgrade hardware versions:

1. Log in to vSphere Client.
2. Right-click the FortiGate-VM in the left *Hosts and Clusters* window.
3. Go to *Compatibility* > *Schedule VM Compatibility Upgrade*, then click *YES*.
4. A dropdown list shows all the available VM hardware versions. Select the desired compatibility, then click *OK*.



5. Reboot the FortiGate-VM.

Deployment

Before you deploy a FortiGate-VM, ensure that you have met the requirements that Preparing for deployment on page 7 describes and that you have extracted the correct deployment package to a folder on the local computer. See Downloading the FortiGate-VM deployment package on page 8.

After you deploy a FortiGate-VM and upload a full license to replace the default evaluation license, you can power on the FortiGate-VM and test connectivity.

Deploying the FortiGate-VM

Use the vSphere client to deploy the FortiGate OVF template and create the FortiGate-VM on the VMware ESXi server.

To create the FortiGate-VM:

1. Deploy the FortiGate OVF template as the VMware documentation describes. Ensure that you select the correct vSphere version in the dropdown list on the right.
2. After deployment, configure the FortiGate-VM. See Initial settings on page 11.

Disk format options

Option	Description
Thick Provision Lazy Zeroed	Allocates the disk space statically (no other volumes can take the space), but does not write zeros to the blocks until the first write takes place to that block during runtime (which includes a full disk format).
Thick Provision Eager Zeroed	Allocates the disk space statically (no other volumes can take the space), and writes zeros to all blocks.
Thin Provision	Allocates the disk space only when a write occurs to a block, but the total volume size is reported by VMFS to the OS. Other volumes can take the remaining space. This allows you to float space between your servers, and expand your storage when your size monitoring indicates there is a problem. Note that once a Thin Provisioned block is allocated, it remains on the volume regardless of whether you have deleted data, etc.

Initial settings

After you deploy a FortiGate-VM on the VMware ESXi server, perform the following tasks:

- Connect the FortiGate-VM to the network so that it can process network traffic and maintain license validity.
- Connect to FortiGate-VM GUI via a web browser for easier administration.

- Ensure that the full license file is uploaded to the FortiGate-VM.
- If you are in a closed environment, enable validation of the FortiGate-VM license against a FortiManager on your network.

Network configuration

The first time you start the FortiGate-VM, you will have access only through the console window of your VMware ESXi server environment. After you configure one FortiGate network interface with an IP address and administrative access, you can access the FortiGate-VM GUI.

Configuring port 1

VM platform or hypervisor management environments include a guest console window. On the FortiGate-VM, this provides access to the FortiGate console, equivalent to the console port on a hardware FortiGate unit. Before you can access the GUI, you must configure FortiGate-VM port1 with an IP address and administrative access.

To configure the port1 IP address:

1. In your hypervisor manager, start the FortiGate-VM and access the console window. You may need to press *Enter* to see a login prompt.
2. At the FortiGate-VM login prompt enter the username `admin`. By default there is no password. Press Enter.
3. Using CLI commands, configure the port1 IP address and netmask:
```
config system interface
   edit port1
      set mode static
      set ip 192.168.0.100 255.255.255.0
   next
end
```
4. To configure the default gateway, enter the following CLI commands:
```
config router static
   edit 1
      set device port1
      set gateway <class_ip>
   next
end
```

You must configure the default gateway with an IPv4 address. FortiGate-VM must access the Internet to contact the FortiGuard Distribution Network to validate its license.

5. To configure your DNS servers, enter the following CLI commands:
```
config system dns
   set primary <Primary DNS server>
   set secondary <Secondary DNS server>
end
```

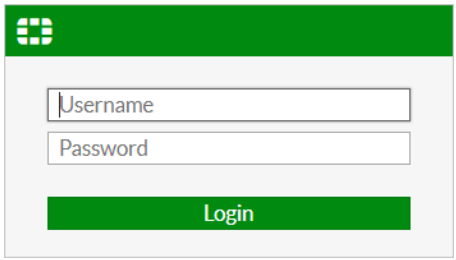
The default DNS servers are `208.91.112.53` and `208.91.112.52`.

Connecting to the FortiGate-VM GUI

You connect to the FortiGate-VM GUI via a web browser by entering the IP address assigned to the port 1 interface (see Configuring port 1 on page 12) in the browser location field. You must enable HTTP and/or HTTPS access and administrative access on the interface to ensure that you can connect to the GUI. If you only enabled HTTPS access, enter "https://" before the IP address.

When you use HTTP rather than HTTPS to access the GUI, certain web browsers may display a warning that the connection is not private.

On the FortiGate-VM GUI login screen, enter the default username "admin", then select *Login*. FortiOS does not assign a default password to the admin user.



Fortinet recommends that you configure a password for the admin user as soon as you log in to the FortiGate-VM GUI for the first time.

Uploading the FortiGate-VM license

Before using the FortiGate-VM, you must enter the license file that you downloaded from Customer Service & Support upon registration.

To upload the FortiGate-VM license file via the GUI:

1. Do one of the following to access the license upload window:
   - In *Dashboard* > *Status* window, in the *Virtual Machine* widget, click the *FGVMEV* (FortiGate-VM Evaluation) *License* icon. This reveals a menu of selections to take you directly to the *FortiGate VM License* window or to the *FortiGuard Details* window.
   - Go to *System* > *FortiGuard*. In the *License Information* section, go to the *Virtual Machine* row and click *FortiGate VM License*.
2. In the *Evaluation License* dialog, select *Enter License*. The license upload page opens.
3. Select *Upload* and locate the license file (`.lic`) on your computer.
4. Select *OK* to upload the license file.
5. Refresh the browser to log in.
6. Enter `admin` in the *Name* field and select *Login*. The VM registration status appears as valid in the *License Information* widget after the FortiGuard Distribution Network or FortiManager for closed networks validates the license.

Modern browsers can have an issue with allowing connecting to a FortiGate if the encryption on the device is too low. If this happens, use an FTP/TFTP server to apply the license.

To upload the FortiGate-VM license file via the CLI:

You can also upload the license file using the following CLI command:

```
execute restore vmlicense {ftp | tftp} <filenmame string> <ftp server>[:ftp port]
```

Example:

The following is an example output when using a TFTP server to install a license:

```
execute restore vmlicense tftp license.lic 10.0.1.2
   This operation will overwrite the current VM license!Do you want to continue? (y/n)y
   Please wait...Connect to tftp server 10.0.1.2 ...
   Get VM license from tftp server OK.
   VM license install succeeded.
   Rebooting firewall.
```

This command automatically reboots the firewall without giving you a chance to back out or delay the reboot.

Validating the FortiGate-VM license with FortiManager

You can validate your FortiGate-VM license with some FortiManager models. To determine whether your FortiManager has the VM activation feature, see the FortiManager datasheet.

To validate your FortiGate-VM with your FortiManager:

1. To configure your FortiManager as a closed network, enter the following CLI command on your FortiManager:
   ```
   config fmupdate publicnetwork
      set status disable
   end
   ```
2. To configure FortiGate-VM to use FortiManager as its override server, enter the following CLI commands on your FortiGate-VM:
   ```
   config system central-management
      set mode normal
      set type fortimanager
      set fmg <FortiManager IPv4 address>
      config server-list
         edit 1
            set server-type update
            set server-address <FortiManager IPv4 address>
         end
      end
      set fmg-source-ip <Source IPv4 address when connecting to the FortiManager>
      set include-default-servers disable
      set vdom <Enter the VDOM name to use when communicating with the FortiManager>
   ```

```
end
```

3. Load the FortiGate-VM license file in the GUI:
   a. Go to *System* > *Dashboard* > *Status*.
   b. In the *License Information* widget, in the *Registration Status* field, select *Update*.
   c. Browse for the `.lic` license file and select *OK*.
4. To activate the FortiGate-VM license, enter the `execute update-now` command on your FortiGate-VM.
5. To check the FortiGate-VM license status, enter the following CLI commands on your FortiGate-VM:

```
get system status
   Version: Fortigate-VM v5.0,build0099,120910 (Interim)
   Virus-DB: 15.00361(2011-08-24 17:17)
   Extended DB: 15.00000(2011-08-24 17:09)
   Extreme DB: 14.00000(2011-08-24 17:10)
   IPS-DB: 3.00224(2011-10-28 16:39)
   FortiClient application signature package: 1.456(2012-01-17 18:27)
   Serial-Number: FGVM02Q105060000
   License Status: Valid
   BIOS version: 04000002
   Log hard disk: Available
   Hostname: Fortigate-VM
   Operation Mode: NAT
   Current virtual domain: root
   Max number of virtual domains: 10
   Virtual domains status: 1 in NAT mode, 0 in TP mode
   Virtual domain configuration: disable
   FIPS-CC mode: disable
   Current HA mode: standalone
   Distribution: International
   Branch point: 511
   Release Version Information: MR3 Patch 4
   System time: Wed Jan 18 11:24:34 2012

diagnose hardware sysinfo vm full
   UUID: 564db33a29519f6b1025bf8539a41e92
   valid: 1
   status: 1
   code: 200 (If the license is a duplicate, code 401 displays)
   warn: 0
   copy: 0
   received: 45438
   warning: 0
   recv: 201201201918
   dup:
```

Licensing timeout

In closed environments without Internet access, you must license the FortiGate-VM offline using a FortiManager as a license server. If the FortiGate-VM cannot validate its license within the 30-day license timeout period, the FortiGate discards all packets, effectively ceasing operation as a firewall.

The license status goes through some changes before it times out. See VM license.

There is only a single log entry after the FortiGate-VM cannot access the license server for the license expiration period. When you search the logs for the reason that the FortiGate is offline, there is no long error log list that draws attention to the issue. There is only one entry.

Testing connectivity

You can now power on your FortiGate-VM.

Use one of the following methods to power on the FortiGate-VM:

- Select the FortiGate-VM in the inventory list, and select *Power on the virtual machine* in the *Getting Started* tab.
- In the inventory list, right-click the FortiGate-VM, and select *Power* > *Power On*.
- Select FortiGate-VM, and click the *Power On* button on the toolbar.

The PING utility is the usual method to test connectivity to other devices. For this, you need the console on the FortiGate-VM. Select the *Console* tab to access the FortiGate-VM console. To enter text, click in the console window. This captures the mouse pointer. However, as the FortiGate-VM console is text-only, the pointer is invisible. To release the pointer, press Ctrl+Alt.

In FortiOS, the command for the PING utility is `execute ping` followed by the IP address you want to connect to.

Before you configure the FortiGate-VM for use in production, ensure that connections between it and all required resources can be established.

- If the FortiGate-VM will provide firewall protection between your network and the Internet, verify that it can connect to your Internet access point and to resources on the Internet.
- If the FortiGate-VM is part of a Fortinet Security Fabric, verify that it can connect to all devices in the Fabric.
- Verify that each node on your network can connect to the FortiGate-VM.

Configuring your FortiGate-VM

For information about configuring and operating the FortiGate-VM after successful deployment and startup on the hypervisor, see the *FortiOS Administration Guide*.

Transparent mode

To configure the FortiGate-VM to operate in transparent mode, you must configure the VMware ESXi server's virtual switches to operate in promiscuous mode to allow traffic that is not addressed to the FortiGate-VM to pass through it.

To configure virtual switches to support FortiGate-VM transparent mode:

1. In the vSphere client, select your VMware server, then select the *Configuration* tab.
2. In *Hardware*, select *Networking*.
3. Select *Properties* of vSwitch0.
4. In the *Properties* window, select *vSwitch*, then select *Edit*.
5. Select the *Security* tab, set *Promiscuous Mode* to *Accept*, then select *OK*.

6. Select *Close*.
7. Repeat steps 3 to 6 for other virtual switches that the FortiGate-VM uses.

HA

FortiGate-VM high availability (HA) supports having two virtual machines in an HA cluster on the same physical server or on different physical servers. In both cases, the two VMs run on the same hypervisor, such as VMware ESXi. The primary consideration is that all interfaces involved can communicate efficiently over TCP/IP connection sessions.

Heartbeat

There are two options for setting up the HA heartbeat: unicast and broadcast. Broadcast is the default HA heartbeat configuration. However, the broadcast configuration may not be ideal for FortiGate-VM because it may require special settings on the host. In most cases, the unicast configuration is preferable.

Differences between the unicast and broadcast heartbeat setups are:

- The unicast method does not change the FortiGate-VM interface MAC addresses to virtual MAC addresses.
- Unicast HA only supports two FortiGate-VMs.
- Unicast HA heartbeat interfaces must be connected to the same network and you must add IP addresses to these interfaces.

Unicast

You can configure the unicast settings in the FortiOS CLI:

```
config system ha
   set unicast-hb {enable/disable}
   set unicast-hb-peerip {Peer heartbeat interface IP address}
end
```

Setting	Description
`unicast-hb`	Enable or disable default unicast HA heartbeat.
`unicast-hb-peerip`	IP address of the HA heartbeat interface of the other FortiGate-VM in the HA cluster.

Broadcast

Broadcast HA heartbeat packets are non-TCP packets that use Ethertype values 0x8892, 0x8891, and 0x8890. These packets use automatically assigned link-local IPv4 addresses in the 169.254.0.x range for HA heartbeat interface IP addresses.

For FortiGate-VMs to support a broadcast HA heartbeat configuration, you must configure the virtual switches that connect heartbeat interfaces to operate in promiscuous mode and support MAC address spoofing.

In addition, you must configure the VM platform to allow MAC address spoofing for the FortiGate-VM data interfaces. This is required because in broadcast mode, the FGCP applies virtual MAC addresses to FortiGate data interfaces, and

these virtual MAC addresses mean that matching interfaces of the FortiGate-VM instances in the cluster have the same virtual MAC addresses.

To configure a virtual switch that connects heartbeat interfaces:

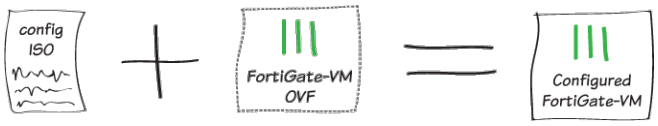
1. In the vSphere client, select your VMware server, then select the *Configuration* tab.
2. In *Hardware*, select *Networking*.
3. Select the virtual switch *Properties*.
4. In the *Properties* window, select *vSwitch*, then select *Edit*.
5. Select the *Security* tab, set *Promiscuous Mode* to *Accept*, then select *OK*.
6. Select *Close*.

You must also configure the virtual switches connected to other FortiGate-VM interfaces to allow MAC address changes and accept forged transmits. This is required because the FGCP sets virtual MAC addresses for all FortiGate-VM interfaces and the same interfaces on the different FortiGate-VM instances in the cluster will have the same virtual MAC addresses.

To configure a virtual switch that connects FortiGate-VM interfaces:

1. In the vSphere client, select your VMware server, then select the *Configuration* tab.
2. In *Hardware*, select *Networking*.
3. Select *Properties* of the virtual switch.
4. Set *MAC Address Changes* to *Accept*.
5. Set *Forged Transmits* to *Accept*.

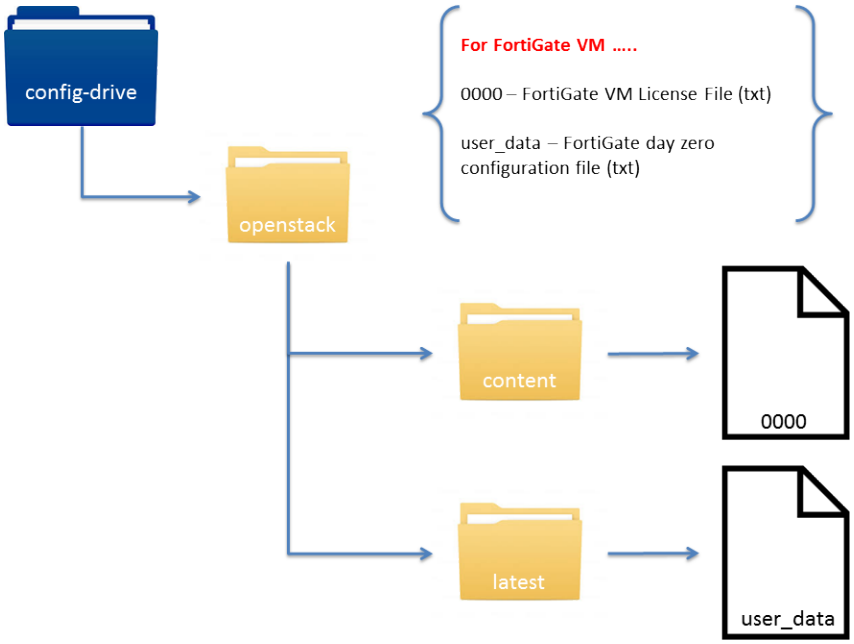
Cloud-init using config drive



This section describes how to bootstrap a FortiGate-VM in VMware vCenter using config drive. This configuration is ideal if you are deploying VMs on VMware vCenter or standalone VMware ESXi and want to preconfigure the FortiGate-VM so that it boots with a predetermined configuration and valid license.

Verify the config drive functionality available for your FortiGate-VM version in the Release Notes. FortiGate-VM supports version 2 of the config-drive capabilities. Config drive was initially created for OpenStack and other cloud environments and is a capability available on the FortiGate-VM even when booting within a VMware vCenter or standalone VMware ESXi environment. With config drive, you can pass day zero configuration scripts and FortiGate-VM licenses to the FortiGate on initial boot.

To pass a config drive to the FortiGate-VM, create a directory structure and place the license file and configuration script file in the appropriate places. Here is the directory structure you need:



For information on the directory structure, see ESXi cloud init reference on page 27.

FortiGate-VM license file

The contents of the FGT-VM license file go into the *0000* file. Generally one would cat the license file and redirect the output into the *config-drive/openstack/content/0000* file.

```
fgt-user@ubuntu:/var/tmp$
```

```
fgt-user@ubuntu:/var/tmp$ cat config-drive/openstack/content/0000
-----BEGIN FGT VM LICENSE——
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-#
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-#
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-#
-----END FGT VM LICENSE——
fgt-user@ubuntu:/var/tmp$
```

FortiGate configuration script

The configuration script for a FortiGate-VM uses standard FortiOS CLI syntax.

Here is a simple example, where the hostname is `Example-Day0` and `port1` is configured to use DHCP to get an IP address:

```
cat config-drive/openstack/latest/user_data
#Example FGT Day0 Configuration
config system global
set hostname Example-Day0
end

config system interface
edit port1
set mode dhcp
set allowaccess https ssh ping
end
fgt-user@ubuntu:/var/tmp$
```

Creating the config drive ISO

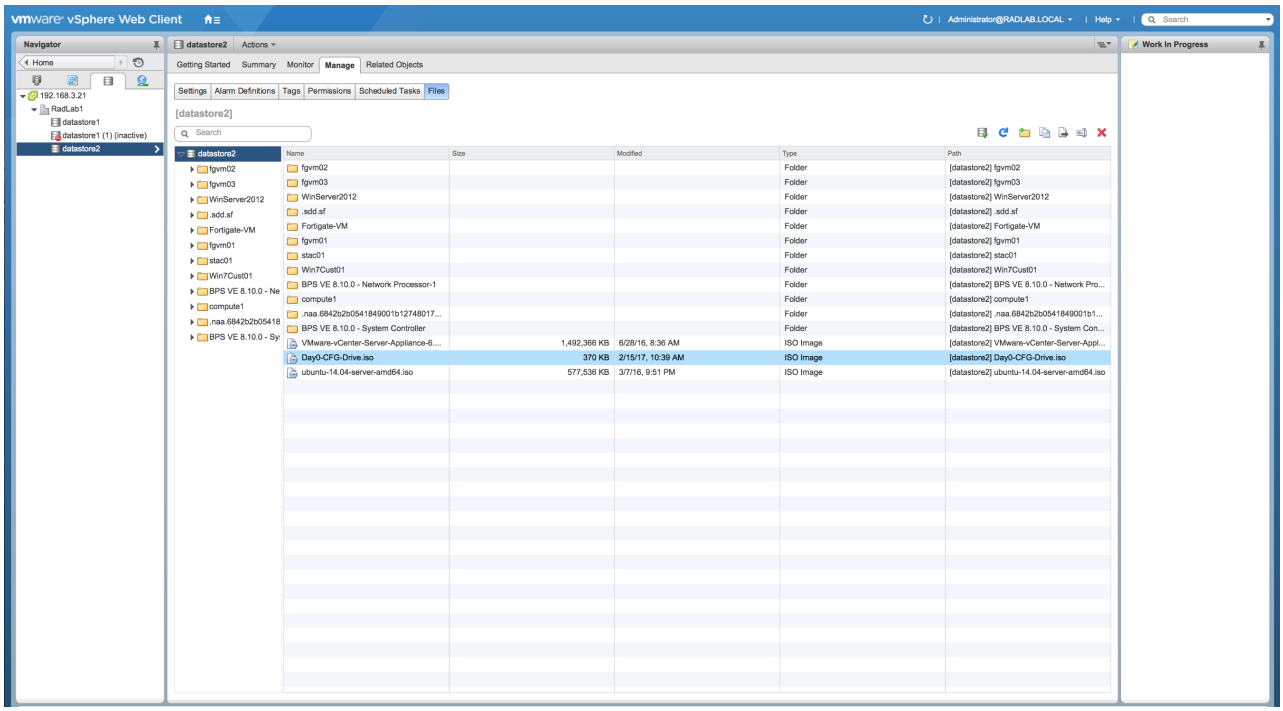
To create the config drive ISO:

1. Create the config-drive ISO using a utility such as **xorriso** (other utilities can also be used to create ISOs, such as **mkisofs**). Using **xorriso**, this example refers to the config-drive directory created above with the relevant license file and configuration script. Here is an example of creating a config-drive ISO on an Ubuntu host:

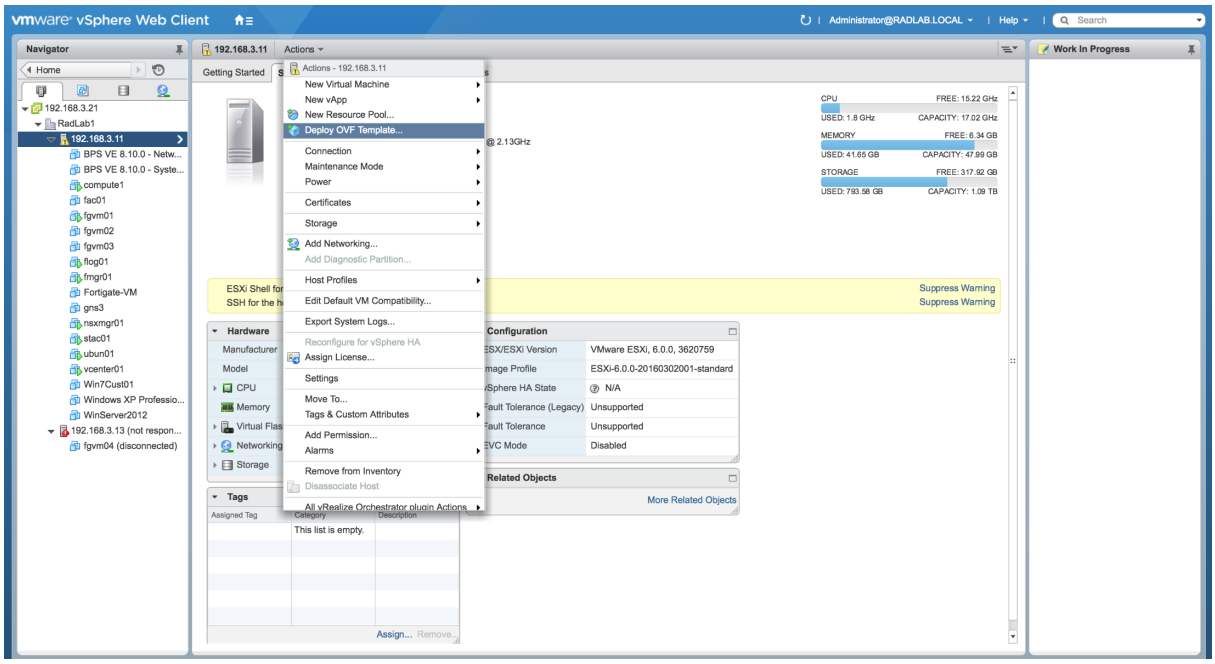
```
xorriso -as mkisofs -V config-2 -o Day0-CFG-Drive.iso config-drive/
xorriso 1.3.2 : RockRidge filesystem manipulator, libburnia project.
Drive current: -outdev 'stdio:Day0-CFG-Drive.iso'
Media current: stdio file, overwriteable
Media status : is blank
Media summary: 0 sessions, 0 data blocks, 0 data, 14.3g free
xorriso : WARNING : -volid text does not comply to ISO 9660 / ECMA 119 rules
Added to ISO image: directory '/'='/var/tmp/config-drive'
xorriso : UPDATE : 5 files added in 1 seconds
xorriso : UPDATE : 5 files added in 1 seconds
ISO image produced: 185 sectors
Written to medium : 185 sectors at LBA 0
Writing to 'stdio:Day0-CFG-Drive.iso' completed successfully.

ls -l Day0-CFG-Drive.iso
-rw-rw-r-- 1 fgt-user fgt-user 378880 Feb 15 13:32 Day0-CFG-Drive.iso
```

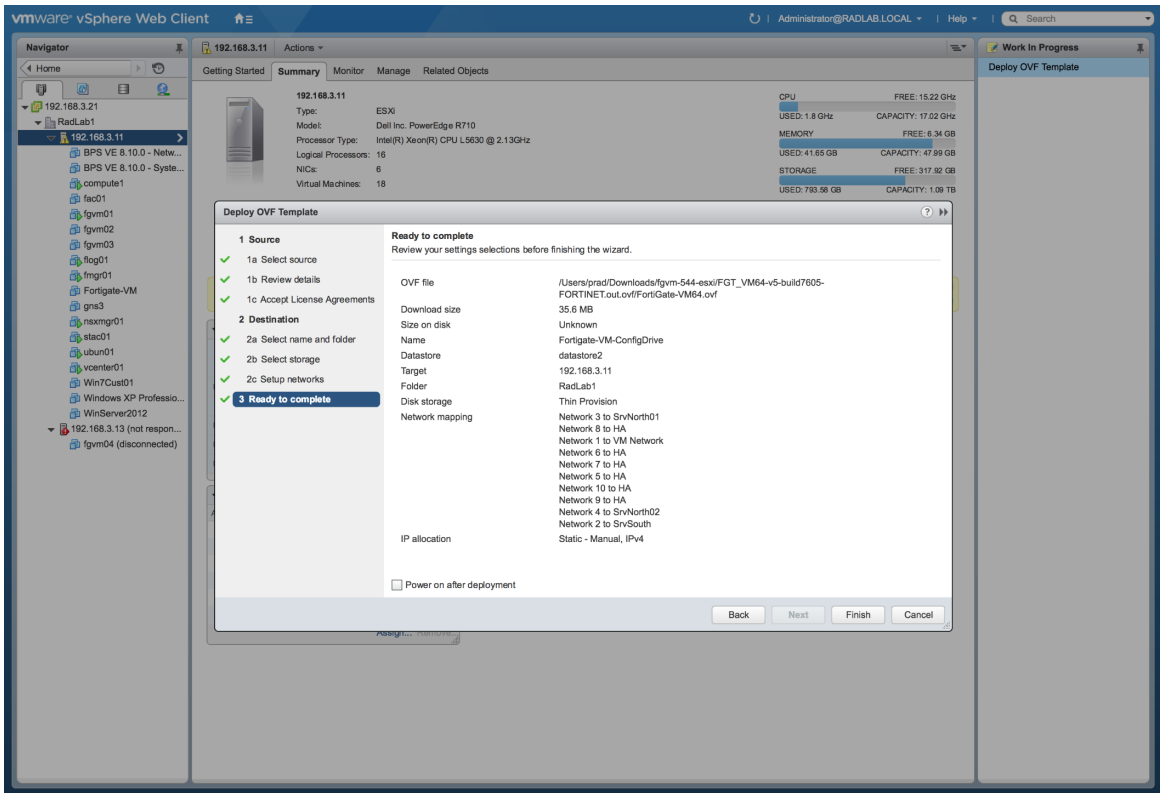
2. Now that the configuration drive has been created, place the ISO on the data store so that it can be used with FortiGate-VMs.



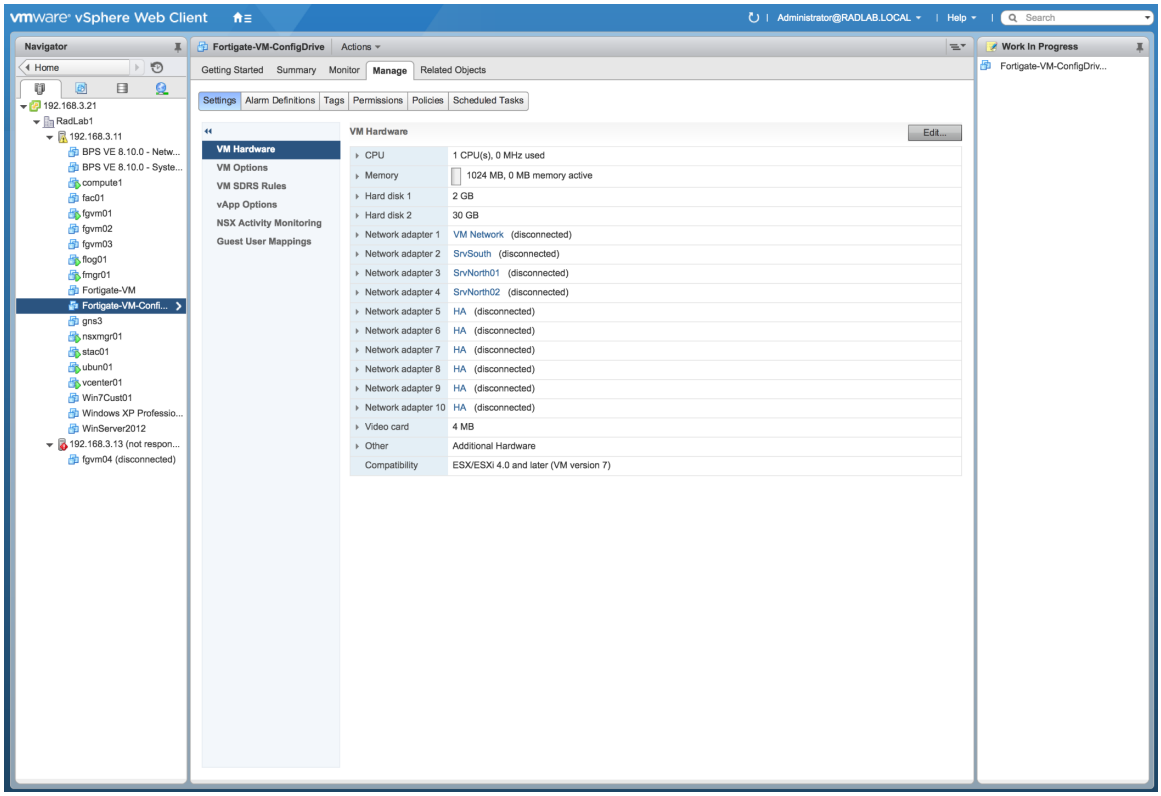
3. Deploy the FortiGate-VM using an OVF template.



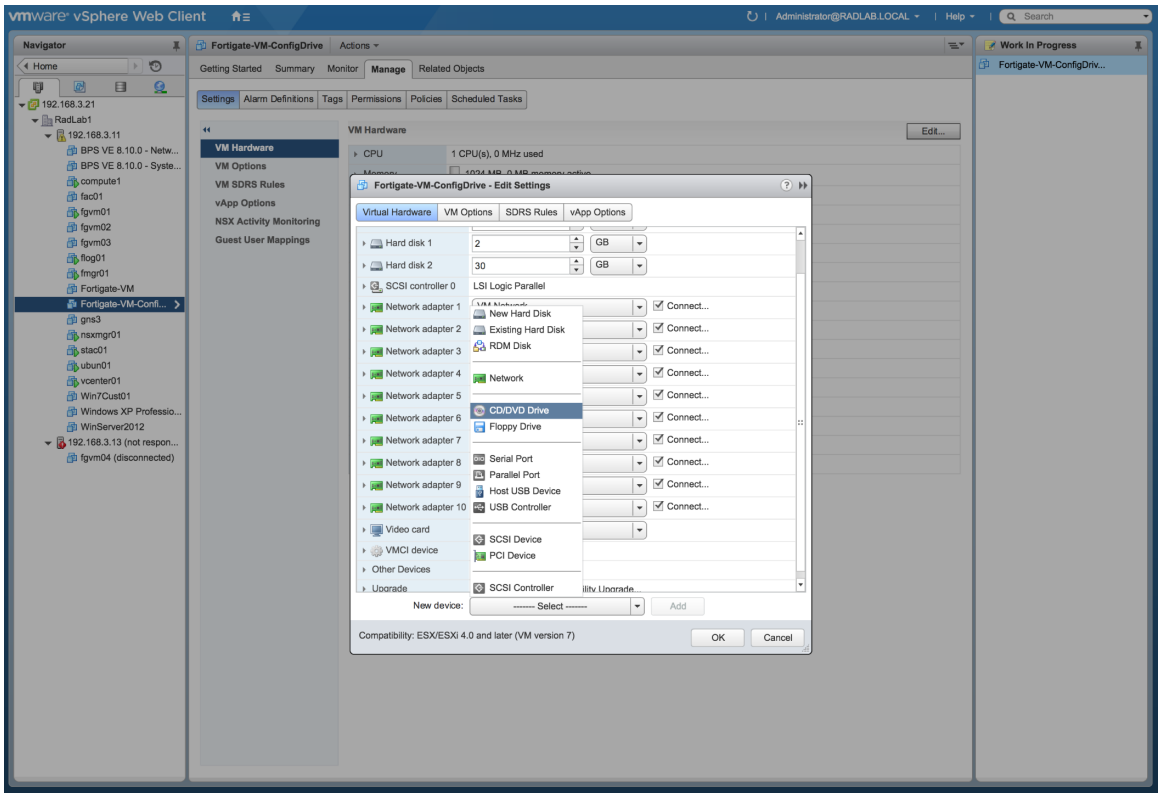
4. Accept the EULA, define your storage policy along with the virtual disk format, and pick the network configuration. Once you reach the end of the OVF template deployment make sure to deselect *Power on after deployment*. This is so we can attach our config-drive ISO as a *cdrom* device before initial boot.



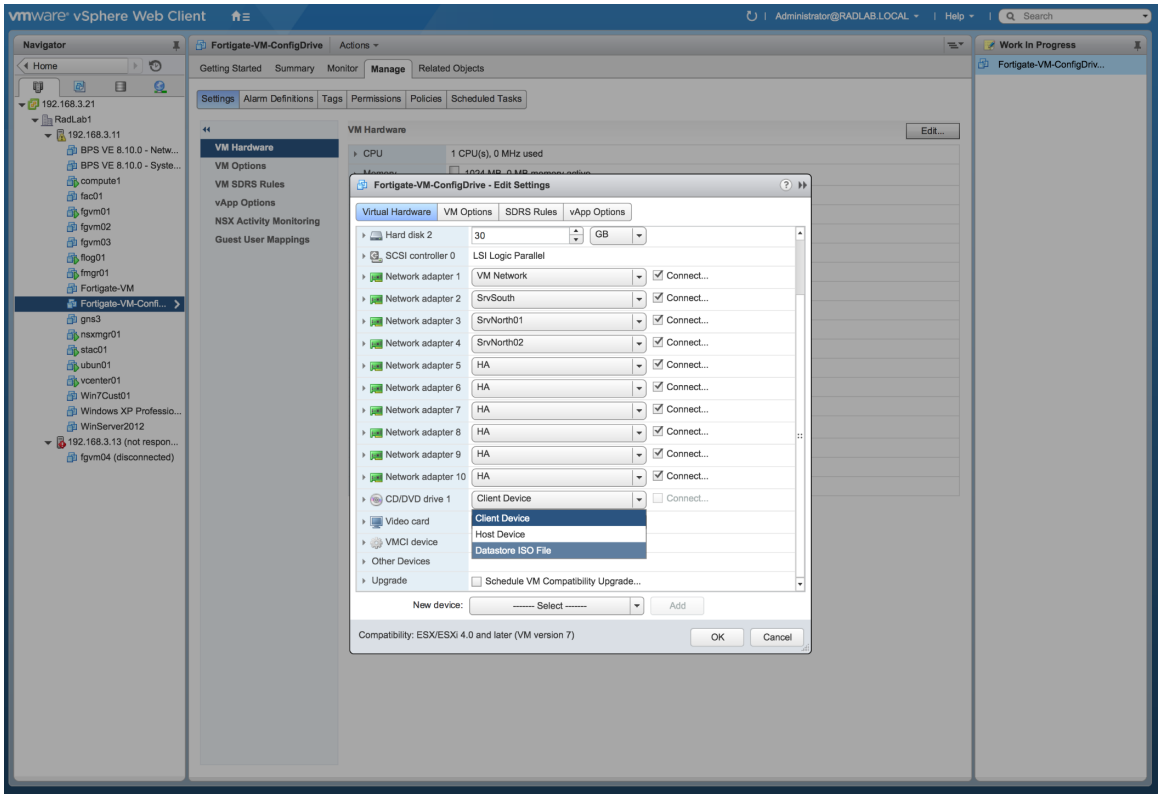
5. Edit the VM settings.

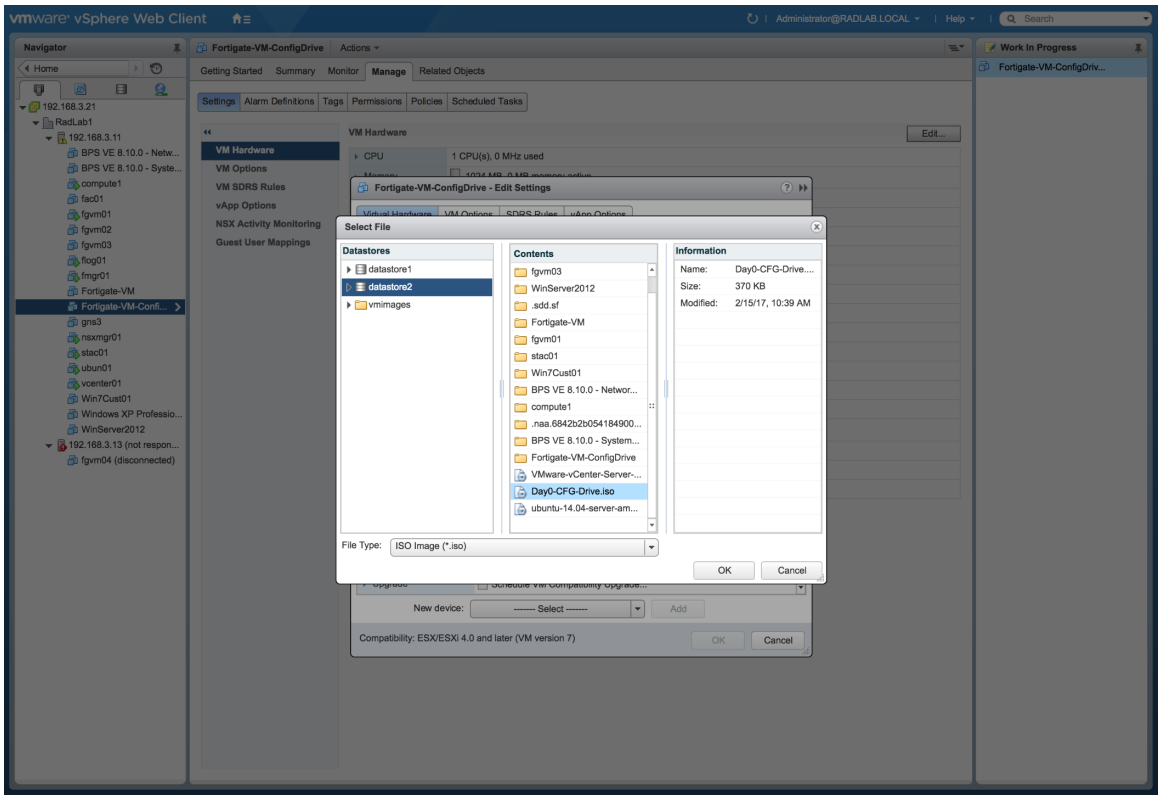


6. Add a new device: *CD/DVD drive* and make sure to select *Connect at power on*.

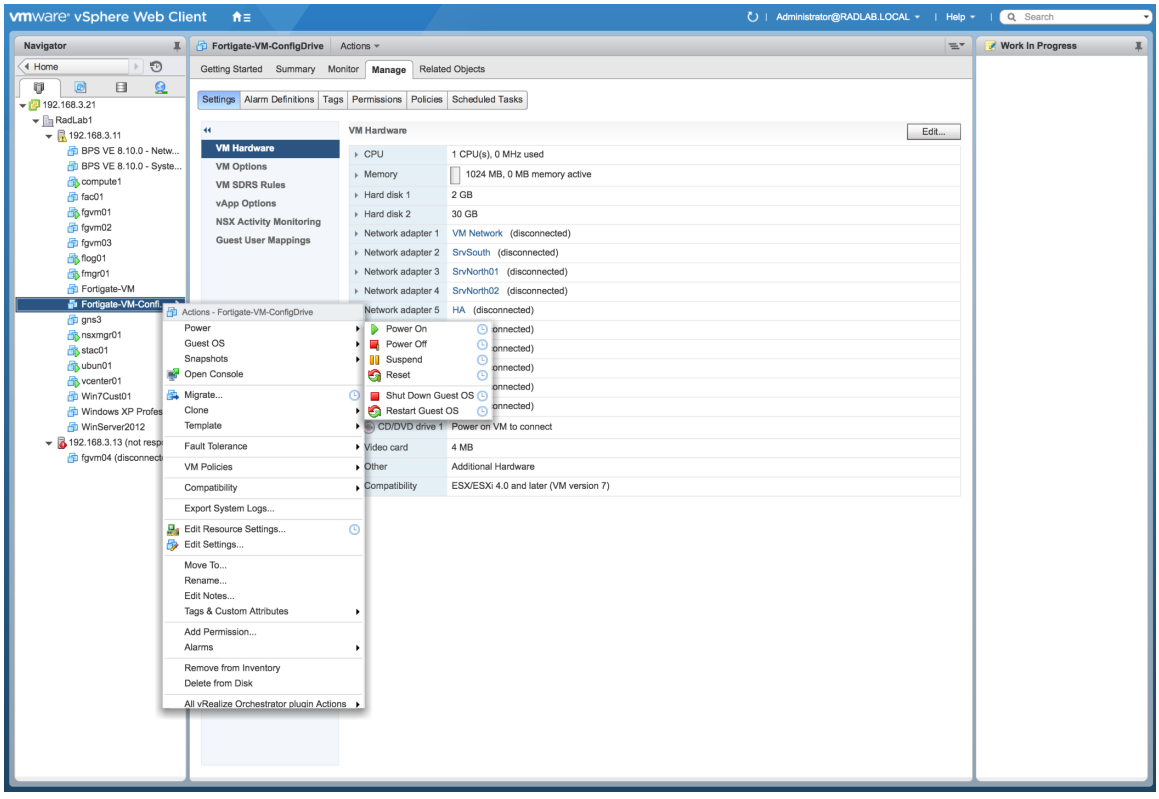


7. Attach the *Day0-CFG-Drive.iso* ISO that you created earlier.





8. Complete your changes, then go to the VM to boot it.

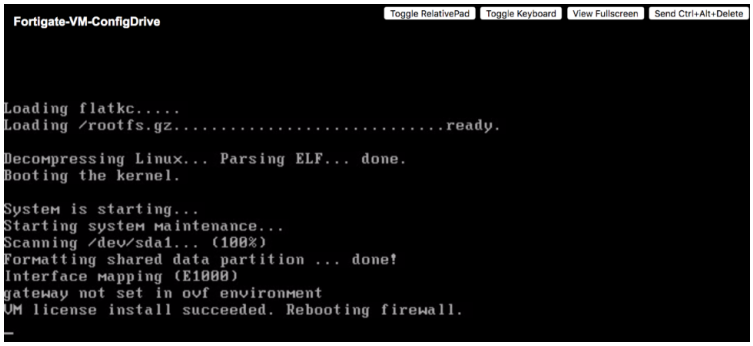


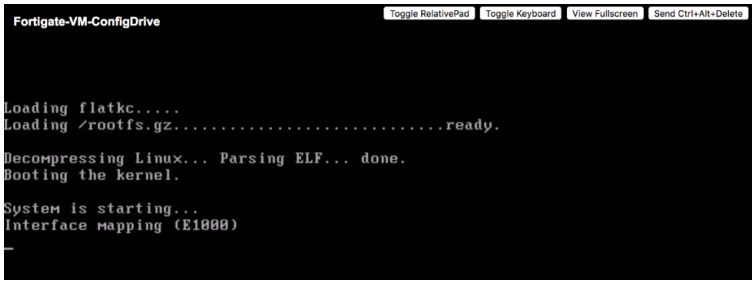
Verifying the results

Boot the FortiGate-VM and open the console to verify that the VM is booting and utilizing the license file and day zero configuration file that was provided.

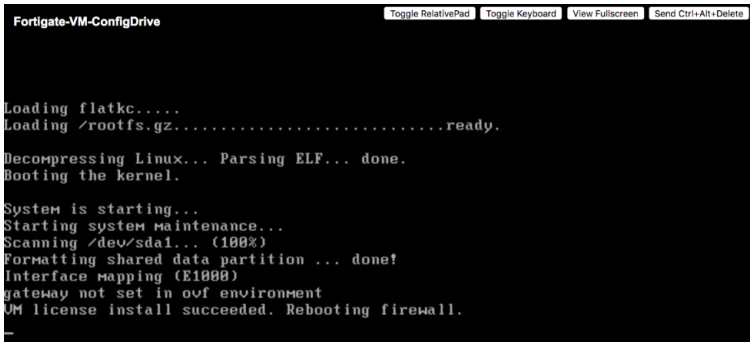
To verify the results:

1. Power on the VM.

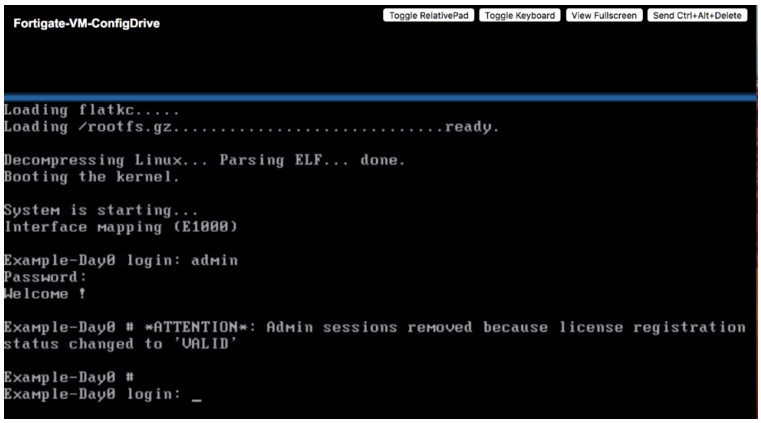




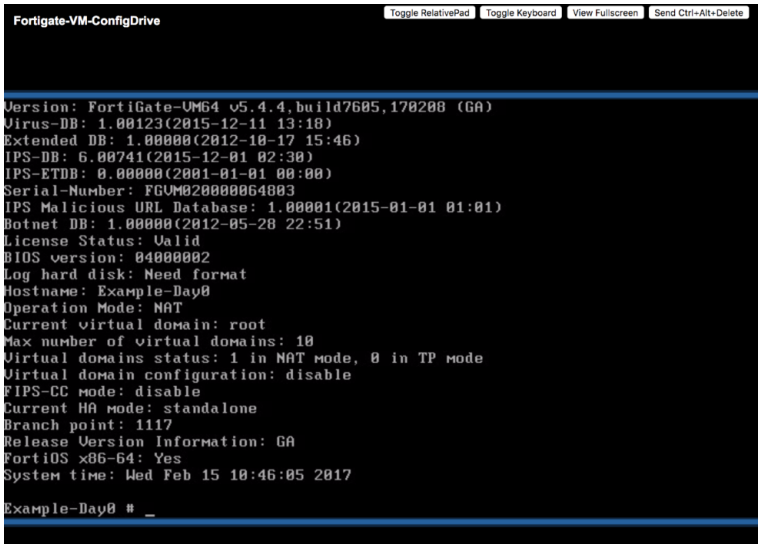
2. Go to the *Console*. Verify that you see the VM *license install succeeded* message and the subsequent reboot.



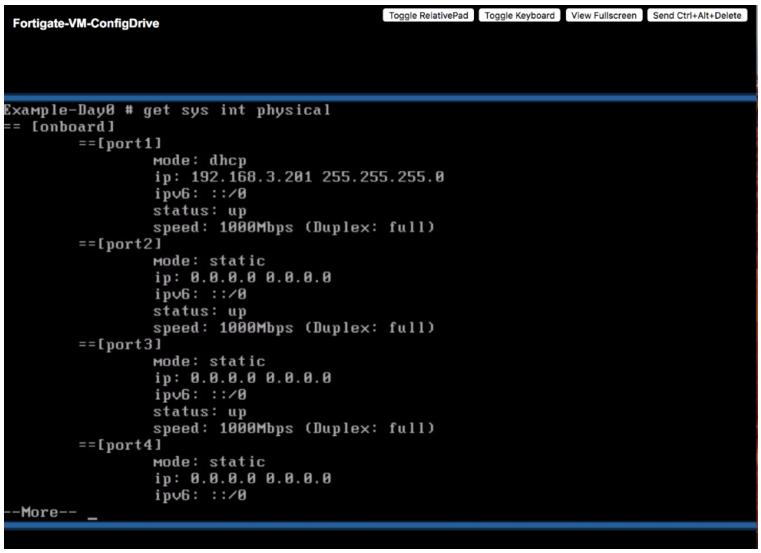
3. Upon completion of the boot sequence, you can verify that the FGT-VM hostname has changed to *Example-Day0*. Also verify that the license file has been verified and the license registration status has changed to *VALID*.



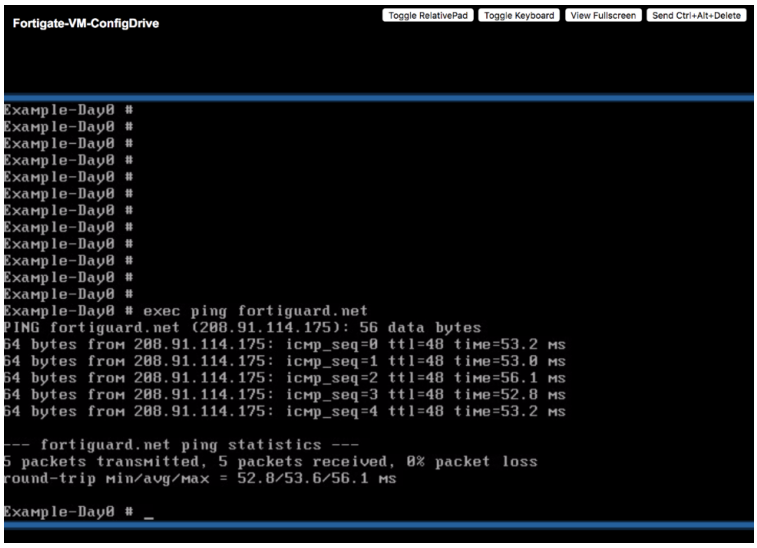
4. After logging in, use the `get system status` command to verify that the license is valid.



5. Use the `get system interface physical` to verify that port1(configured in DHCP mode), has received an IP from the DHCP server.



6. Attempt to ping `fortiguard.com` to confirm that the FortiGate-VM can contact Fortinet for licensing and updates.



ESXi cloud init reference

For VMware ESXi, you use the utility **xorriso** on a Linux host to create the ISO used to boot the VM. The following describes the directory structure used to create the ISO.

After you create the ISO, you must upload it to your datastore of choice and attach it to the FortiGate-VM after deploying the OVF but before booting it up for the first time.

```
ls -lR config-drive/
config-drive/:
total 4
drwxrwxr-x 4 fgt-user fgt-user 4096 Feb 8 16:59 openstack

config-drive/openstack:
total 8
drwxrwxr-x 2 fgt-user fgt-user 4096 Feb 8 17:07 content
drwxrwxr-x 2 fgt-user fgt-user 4096 Feb 8 17:06 latest

config-drive/openstack/content:
total 4
-rw-rw-r-- 1 fgt-user fgt-user 287 Feb 8 17:00 0000

config-drive/openstack/latest:
total 4
-rw-r--r-- 1 fgt-user fgt-user 172 Feb 8 17:06 user_data

cat config-drive/openstack/content/0000
-----BEGIN FGT VM LICENSE-----
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED- REDACTED-#
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED- REDACTED-#
#-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-REDACTED-#
-----END FGT VM LICENSE-----

cat config-drive/openstack/latest/user_data
```

```
#Example FGT Day0 Configuration

config system global
set hostname Example-Day0
end
config system interface
edit port1
set mode dhcp
set allowaccess https ssh ping end

xorriso -as mkisofs -V config-2 -o Day0-CFG-Drive.iso config-drive/
xorriso 1.3.2 : RockRidge filesystem manipulator, libburnia project.

Drive current: -outdev 'stdio:Day0-CFG-Drive.iso' Media current: stdio file, overwriteable
Media status : is blank
Media summary: 0 sessions, 0 data blocks, 0 data, 14.3g free
xorriso : WARNING : -volid text does not comply to ISO 9660 / ECMA 119 rules Added to ISO
     image:
directory '/'='/var/tmp/config-drive'
xorriso : UPDATE : 5 files added in 1 seconds xorriso : UPDATE : 5 files added in 1 seconds
     ISO
image produced: 185 sectors
Written to medium : 185 sectors at LBA 0
Writing to 'stdio:Day0-CFG-Drive.iso' completed successfully.

ls -l Day0-CFG-Drive.iso
-rw-rw-r-- 1 fgt-user fgt-user 378880 Feb 15 13:32 Day0-CFG-Drive.iso
```

SDN connector integration with VMware ESXi

See VMware ESXi SDN connector using server credentials.

Optimizing FortiGate-VM performance

This section describes FortiGate-VM and VMware ESXi performance optimization techniques that can improve your FortiGate-VM performance by optimizing the hardware and the VMware ESXi host environment for FortiGate-VM's network- and CPU-intensive performance requirements.

Additionally, the port4 interface MTU is set to be compatible with the OpenStack 10 environment, which has an MTU of 1446 by default. (In the user_data file, the MTU of port4 is set to 1400.) Using the same MTU setting as the OpenStack 10 environment enables the HA heartbeat interfaces to communicate effectively over the ha-sync network.

See these pages for more information on RedHat OpenStack networks and MTU values:

- MTU for VLAN networks is by default 1496 Bytes in Red Hat OpenStack Platform 10
- Configure MTU Settings

SR-IOV

FortiGate-VMs installed on VMware ESXi platforms support Single Root I/O virtualization (SR-IOV) to provide FortiGate-VMs with direct access to physical network cards. Enabling SR-IOV means that one PCIe network card or CPU can function for a FortiGate-VM as multiple separate physical devices. SR-IOV reduces latency and improves CPU efficiency by allowing network traffic to pass directly between a FortiGate-VM and a network card, bypassing VMware ESXi host software and without using virtual switching.

FortiGate-VMs benefit from SR-IOV because SR-IOV optimizes network performance and reduces latency and CPU usage. FortiGate-VMs do not use VMware ESXi features that are incompatible with SR-IOV, so you can enable SR-IOV without negatively affecting your FortiGate-VM. SR-IOV implements an I/O memory management unit (IOMMU) to differentiate between different traffic streams and apply memory and interrupt translations between the physical functions (PF) and virtual functions (VF).

Setting up SR-IOV on VMware ESXi involves creating a PF for each physical network card in the hardware platform. Then, you create VFs that allow FortiGate-VMs to communicate through the PF to the physical network card. VFs are actual PCIe hardware resources and only a limited number of VFs are available for each PF.

SR-IOV hardware compatibility

SR-IOV requires that the hardware and operating system on which your VMware ESXi host is running has BIOS, physical NIC, and network driver support for SR-IOV.

To enable SR-IOV, your VMware ESXi platform must be running on hardware that is compatible with SR-IOV and with FortiGate-VMs. FortiGate-VMs require network cards that are compatible with the supported drivers. See PF and VF SR-IOV driver and virtual SPU support for supported driver versions. As well, the host hardware CPUs must support second level address translation (SLAT).

For optimal SR-IOV support, install the most up to date network drivers. Fortinet recommends i40e/Iavf drivers because they provide four TxRx queues for each VF and ixgbevf only provides two TxRx queues.

Creating SR-IOV virtual interfaces

Complete the following procedure to enable SR-IOV. This procedure requires restarting the VMware host and powering down the FortiGate-VM and should only be done during a maintenance window or when the network is not very busy.

To create SR-IOV virtual interfaces:

1. Do one of the following:
   a. If using the VMware host client, do the following:
      i. Go to *Manage* > *Hardware* > *PCI Devices* to view all PCI devices on the host.
      ii. Select the *SR-IOV capable* filter to view the PCI devices (network adapters) that are compatible with SR-IOV.
      iii. Select a network adapter and select *Configure SR-IOV*.
      iv. Enable *SR-IOV* and specify the *Number of virtual functions*.
      v. Save your changes and restart the VMware host.
   b. If using the vSphere web client, do the following:
      i. Go to the host with the SR-IOV physical network adapter that you want to add virtual interfaces to.
      ii. In the *Networking* part of the *Manage* tab, select *Physical Adapters*.
      iii. Select the physical adapter for which to enable SR-IOV settings.
      iv. Enable *SR-IOV* and specify the *Number of virtual functions*.
      v. Save your changes and restart the VMware host.

You can also use the following command from the VMware ESXi host CLI to add virtual interfaces to one or more compatible network adapters:

```
$ esxcli system module parameters set -m <driver-name> -p "max_vfs=<virtual-interfaces>"
```

Where `<driver-name>` is the network adapter driver name (for example `ixgbevf` or `i40evf`) and `<virtual-interfaces>` is a comma-separated list of number of virtual interfaces to allow for each physical interface.

For example, if your VMware host includes three i40evf network adapters and you want to enable 6 virtual interfaces on each network adapter, enter the following:

```
$ esxcli system module parameters set -m <i40evf> -p "max_vfs=6,6,6"
```

Assigning SR-IOV virtual interfaces to a FortiGate-VM

To assign SR-IOV virtual interfaces to a FortiGate-VM:

1. Power off the FortiGate-VM and open its virtual hardware settings.
2. Create or edit a network adapter and set its type to *SR-IOV passthrough*.
3. Select the physical network adapter for which you have enabled SR-IOV.
4. Optionally associate the FortiGate-VM network adapter with the port group on a standard or distributed switch.
5. To guarantee that the pass-through device can access all VM memory, in the *Memory* section select *Reserve all guest memory*.
6. Save your changes and power on the FortiGate-VM.

Setting up VMware CPU affinity

Configuring CPU affinity on your FortiGate-VM further builds on the benefits of SR-IOV by enabling the FortiGate-VM to align interrupts from interfaces to specific CPUs.

By specifying a CPU affinity setting for each VM, you can restrict the assignment of VMs to a subset of the available processors in multiprocessor systems. By using this feature, you can assign each VM to processors in the specified affinity set.

Using CPU affinity, you can assign a VM to a specific processor. This assignment allows you to restrict the assignment of VMs to a specific available processor in multiprocessor systems.

To set up VMware CPU affinity when using the vSphere web client:

1. Power off the FortiGate-VM.
2. Edit the FortiGate-VM hardware settings and select *Virtual Hardware*.
3. Select CPU options.
4. In *Scheduling Affinity*, specify the CPUs to have affinity with the FortiGate-VM. For best results, the affinity list should include one entry for each of the FortiGate-VM's virtual CPUs.
5. Save your changes.

Interrupt affinity

In addition to enabling SR-IOV in the VM host, to fully take advantage of SR-IOV performance improvements you must configure interrupt affinity for your FortiGate-VM. Interrupt affinity (also called CPU affinity) maps FortiGate-VM interrupts to the CPUs that are assigned to your FortiGate-VM. You use a CPU affinity mask to define the CPUs that the interrupts are assigned to.

A common use of this feature is to improve your FortiGate-VM's networking performance by:

- On the VM host, add multiple host CPUs to your FortiGate-VM.
- On the VM host, configure CPU affinity to specify the CPUs that the FortiGate-VM can use.
- On the VM host, configure other VM clients on the VM host to use other CPUs.
- On the FortiGate-VM, assign network interface interrupts to a CPU affinity mask that includes the CPUs that the FortiGate-VM can use.

In this way, all available CPU interrupts for the configured host CPUs are used to process traffic on your FortiGate interfaces. This configuration could lead to improve FortiGate-VM network performance because you have dedicated VM host CPU cycles to processing your FortiGate-VM's network traffic.

You can use the following CLI command to configure interrupt affinity for your FortiGate-VM:

```
config system affinity-interrupt
   edit <index>
      set interrupt <interrupt-name>
      set affinity-cpumask <cpu-affinity-mask>
   next
end
```

Where:

- `<interrupt-name>` is the name of the interrupt to associate with a CPU affinity mask. You can view your FortiGate-VM interrupts using the `diagnose hardware sysinfo interrupts` command. Usually you associate all of the interrupts for a given interface with the same CPU affinity mask.
- `<cpu-affinity-mask>` is the CPU affinity mask for the CPUs that will process the associated interrupt.

For example, consider the following configuration:

- The port2 and port3 interfaces of a FortiGate-VM send and receive most of the traffic.
- On the VM host you have set up CPU affinity between your FortiGate-VM and four CPUs (CPU 0, 1 , 2, and 3).
- SR-IOV is enabled and SR-IOV interfaces use the i40evf interface driver.

The output from the `diagnose hardware sysinfo interrupts` command shows that port2 has the following transmit and receive interrupts:

```
i40evf-port2-TxRx-0
i40evf-port2-TxRx-1
i40evf-port2-TxRx-2
i40evf-port2-TxRx-3
```

The output from the `diagnose hardware sysinfo interrupts` command shows that port3 has the following transmit and receive interrupts:

```
i40evf-port3-TxRx-0
i40evf-port3-TxRx-1
i40evf-port3-TxRx-2
i40evf-port3-TxRx-3
```

Use the following command to associate the port2 and port3 interrupts with CPU 0, 1 , 2, and 3.

```
config system affinity-interrupt
   edit 1
      set interrupt "i40evf-port2-TxRx-0"
      set affinity-cpumask "0x0000000000000001"
   next
   edit 2
      set interrupt "i40evf-port2-TxRx-1"
      set affinity-cpumask "0x0000000000000002"
   next
   edit 3
      set interrupt "i40evf-port2-TxRx-2"
      set affinity-cpumask "0x0000000000000004"
   next
   edit 4
      set interrupt "i40evf-port2-TxRx-3"
      set affinity-cpumask "0x0000000000000008"
   next
   edit 1
      set interrupt "i40evf-port3-TxRx-0"
      set affinity-cpumask "0x0000000000000001"
   next
   edit 2
      set interrupt "i40evf-port3-TxRx-1"
      set affinity-cpumask "0x0000000000000002"
   next
   edit 3
      set interrupt "i40evf-port3-TxRx-2"
      set affinity-cpumask "0x0000000000000004"
   next
   edit 4
```

```
      set interrupt "i40evf-port3-TxRx-3"
      set affinity-cpumask "0x0000000000000008"
   next
end
```

Packet-distribution affinity

With SR-IOV enabled on the VM host and interrupt affinity configured on your FortiGate-VM there is one additional configuration you can add that may improve performance. Most common network interface hardware has restrictions on the number of RX/TX queues that it can process. This can result in some CPUs being much busier than others and the busy CPUs may develop extensive queues.

You can get around this potential bottleneck by configuring affinity packet redistribution to allow overloaded CPUs to redistribute packets they receive to other less busy CPUs. The may result in a more even distribution of packet processing to all available CPUs.

You configure packet redistribution for interfaces by associating an interface with an affinity CPU mask. This configuration distributes packets set and received by that interface to the CPUs defined by the CPU affinity mask associated with the interface.

You can use the following CLI command to configure affinity packet redistribution for your FortiGate-VM:

```
config system affinity-packet-redistribution
   edit <index>
      set interface <interface-name>
      set affinity-cpumask <cpu-affinity-mask>
   next
end
```

Where:

- `<interface-name>` the name of the interface to associate with a CPU affinity mast.
- `<cpu-affinity-mask>` the CPU affinity mask for the CPUs that will process packets to and from the associated interface.

For example, you can improve the performance of the interrupt affinity example shown in the following command to allow packets sent and received by the port3 interface to be redistributed to CPUs according to the 0xE CPU affinity mask.

```
config system affinity-packet-redistribution
   edit 1
      set interface port3
      set affinity-cpumask "0xE"
   next
end
```

TSO and LRO

Enabling TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) can improve FortiGate-VM performance by reducing CPU overhead for TCP/IP network operations.

TSO causes network cards to divide larger data chunks into TCP segments. If you disable TSO, the CPU performs segmentation for TCP/IP. TSO is also sometimes called Large Segment Offload (LSO) or Large Send Offload.

LRO reassembles incoming network packets into larger buffers and transfers the resulting larger but fewer packets to the network stack of the host or VM. The CPU has to process fewer packets.

Your server hardware must support TSO and LRO.

To enable TSO from the vSphere web client:

1. Open the *Manage* tab and select *Advanced System Settings*.
2. For IPv4 set `Net.UseHwTSO` to 1 to enable TSO, or to 0 to disable TSO.
3. For IPv6 set `useNet.UseHwTSO6` to 1 to enable TSO, or to 0 to disable TSO.

To enable LRO from the vSphere web client:

1. Open the *Manage* tab and select *Advanced System Settings*.
2. For IPv4 TSO, set `Net.Vmxnet2HwLRO` and `Net.Vmxnet3HwLRO` to 1 to enable LRO, or to 0 to disable LRO.
3. For IPv6 TSO, set `useNet.UseHwTSO6` to 1 to enable TSO, or to 0 to disable TSO.

Multiqueue support

Multiqueue can scale network performance with the number of vCPUs. Multiqueue can also create multiple TX and RX queues. Modify the .vmx file or access *Advanced Settings* to enable multi-queu.

To enable multiqueue:

1. Open the .vmx file.
2. Add the `ethernetX.pnicFeatures = "4"` parameter.

To enable receive-side scaling (RSS) from the ESXi CLI:

```
$ vmkload_mod -u ixgbe
$ vmkload_mod ixgbe RSS="4,4,4,4,4,4"
```

For the best performance, you should also configure additional CPU threads for each ethernet/vSwitch device. This is limited by the amount of spare CPU resources available on the ESXi host.

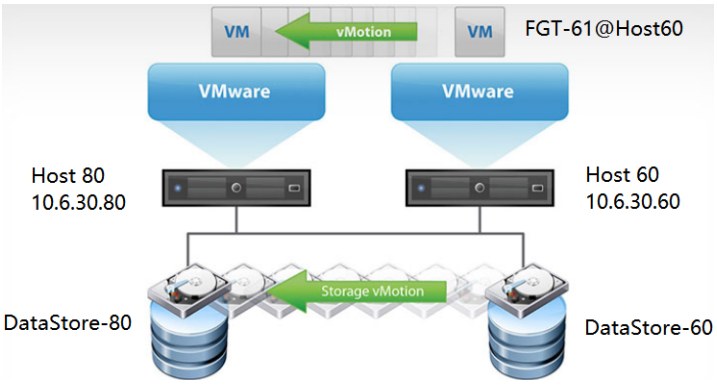
To configure additional CPU threads for each ethernet/vSwitch device:

1. Open the .vmx file.
2. Add the `ethernetX.ctxPerDev = "1"` parameter.

vMotion in a VMware ESXi environment

This guide provides sample configuration of vMotion FortiGate-VM HA in a VMware ESXi environment. This feature enables the live migration of a running FortiGate-VM from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. It also provides transparency to users.

The following depicts the network topology for this sample deployment. In this sample deployment, there are two hosts, Host 60 (10.6.30.60) and Host 80 (10.6.30.80), which are members of Cluster 1 in the DataCenter 1. The vCenter server (10.6.30.99) manages DataCenter 1.



The following prerequisites must be met for this configuration:

- The vCenter server has been set up and the data center and cluster have been created.
- Host 60 and Host 80 are part of the cluster.
- A Gigabit Ethernet network interface card with a VMkernel port enabled for vMotion exists on both ESXi hosts.
- A FortiGate-VM is set up and able to handle traffic.

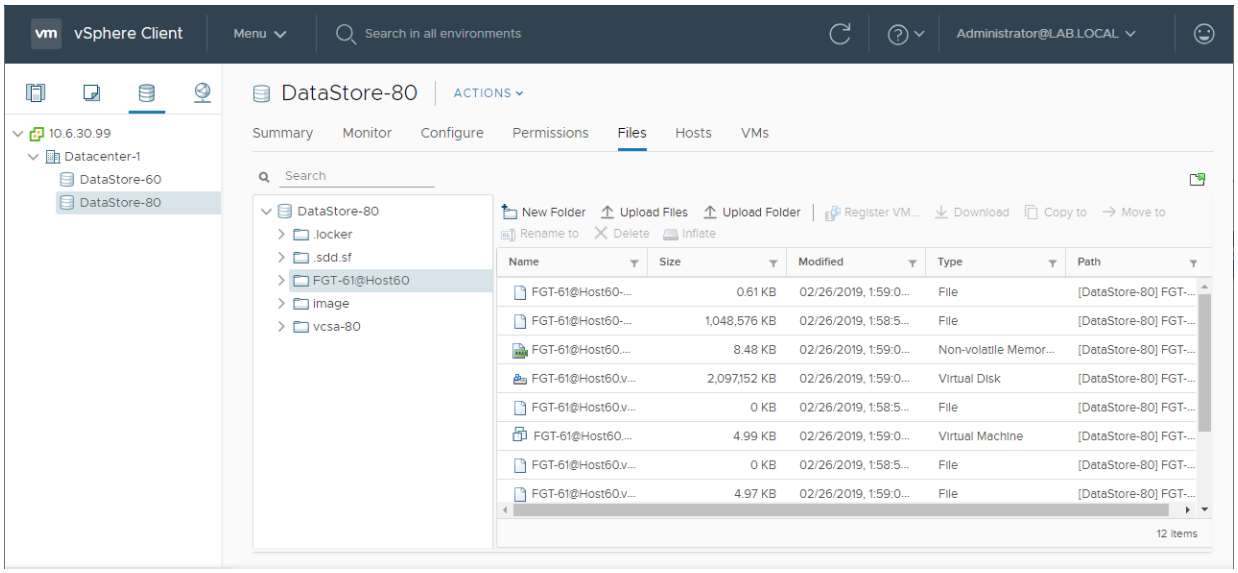
To migrate the FortiGate-VM on the vCenter web portal:

1. Log into the vCenter web portal.
2. Verify the current location of the FortiGate-VM:
   a. Go to the FortiGate-VM.
   b. On the *Summary* tab, check the *Host*. In this example, the host is currently Host 60 (10.6.30.60).
   c. Go to *Storage* > *Files*. Check that the FortiGate-VM is located in the correct datastore. In this example, the datastore is currently Datastore 60, which is in Host 60.
3. Right-click the FortiGate-VM and select *Migrate*.
4. Configure the migration options:
   a. For *Select a migration type*, select *Change both compute resource and storage*. Click *NEXT*.
   b. For *Select a compute resource*, select the desired new compute resource. In this example, Host 80 (10.6.30.80) is selected. Click *NEXT*.
   c. For *Select storage*, select the storage associated with the compute resource selected in step 5. In this example, Datastore 80 (as corresponds to Host 80) is selected. Click *NEXT*.
   d. For *Select networks*, select the desired destination network at the compute resource selected in step 5. In this example, the source network is at Host 60, and the destination network is at Host 80. Click *NEXT*.s
   e. For *Select vMotion priority*, select *Schedule vMotion with high priority (recommended)*. Click *NEXT*.

5. Before initiating the migration, open the CLI for the FortiGate-VM to check on traffic during the migration. Enter the `diag sniffer packet any 'icmp and host 8.8.8.8'` command to check if traffic is stable. If no traffic is lost during migration and the FortiGate-VM SSH session does not break, the output resembles the following:

```
FortiGate-VM64 # diag sniffer packet any 'icmp and host 8.8.8.8'
interfaces=[any]
filters=[icmp and host 8.8.8.8]
2.284655 10.1.100.22 -> 8.8.8.8: icmp: echo request
2.284704 172.16.200.61 -> 8.8.8.8: icmp: echo request
2.290014 8.8.8.8 -> 172.16.200.61: icmp: echo reply
2.290023 8.8.8.8 -> 10.1.100.22: icmp: echo reply
3.286396 10.1.100.22 -> 8.8.8.8: icmp: echo request
3.286399 172.16.200.61 -> 8.8.8.8: icmp: echo request
3.291257 8.8.8.8 -> 172.16.200.61: icmp: echo reply
3.291259 8.8.8.8 -> 10.1.100.22: icmp: echo reply
4.287616 10.1.100.22 -> 8.8.8.8: icmp: echo request
4.287620 172.16.200.61 -> 8.8.8.8: icmp: echo request
4.293134 8.8.8.8 -> 172.16.200.61: icmp: echo reply
4.293136 8.8.8.8 -> 10.1.100.22: icmp: echo reply
5.289483 10.1.100.22 -> 8.8.8.8: icmp: echo request
5.289486 172.16.200.61 -> 8.8.8.8: icmp: echo request
5.294584 8.8.8.8 -> 172.16.200.61: icmp: echo reply
5.294586 8.8.8.8 -> 10.1.100.22: icmp: echo reply
6.290972 10.1.100.22 -> 8.8.8.8: icmp: echo request
6.290976 172.16.200.61 -> 8.8.8.8: icmp: echo request
6.295467 8.8.8.8 -> 172.16.200.61: icmp: echo reply
6.295469 8.8.8.8 -> 10.1.100.22: icmp: echo reply
7.292842 10.1.100.22 -> 8.8.8.8: icmp: echo request
7.292846 172.16.200.61 -> 8.8.8.8: icmp: echo request
7.297360 8.8.8.8 -> 172.16.200.61: icmp: echo reply
7.297362 8.8.8.8 -> 10.1.100.22: icmp: echo reply
8.294735 10.1.100.22 -> 8.8.8.8: icmp: echo request
8.294742 172.16.200.61 -> 8.8.8.8: icmp: echo request
8.299282 8.8.8.8 -> 172.16.200.61: icmp: echo reply
8.299285 8.8.8.8 -> 10.1.100.22: icmp: echo reply
9.296594 10.1.100.22 -> 8.8.8.8: icmp: echo request
9.296600 172.16.200.61 -> 8.8.8.8: icmp: echo request
9.301125 8.8.8.8 -> 172.16.200.61: icmp: echo reply
9.301127 8.8.8.8 -> 10.1.100.22: icmp: echo reply
```

6. Click *FINISH*. After a few seconds, the FortiGate-VM is migrated to the new compute resources, in this case Host 80.
7. Log into the vCenter web portal. Go to the FortiGate-VM. On the *Summary* tab, the *Host* is now the new compute resources, in this case Host 80 (10.6.30.80).
8. Go to *Storage* > *Files*. It shows that the FortiGate-VM is now located in a new datastore, in this example Datastore 80.



To configure the FortiGate-VM using the CLI:

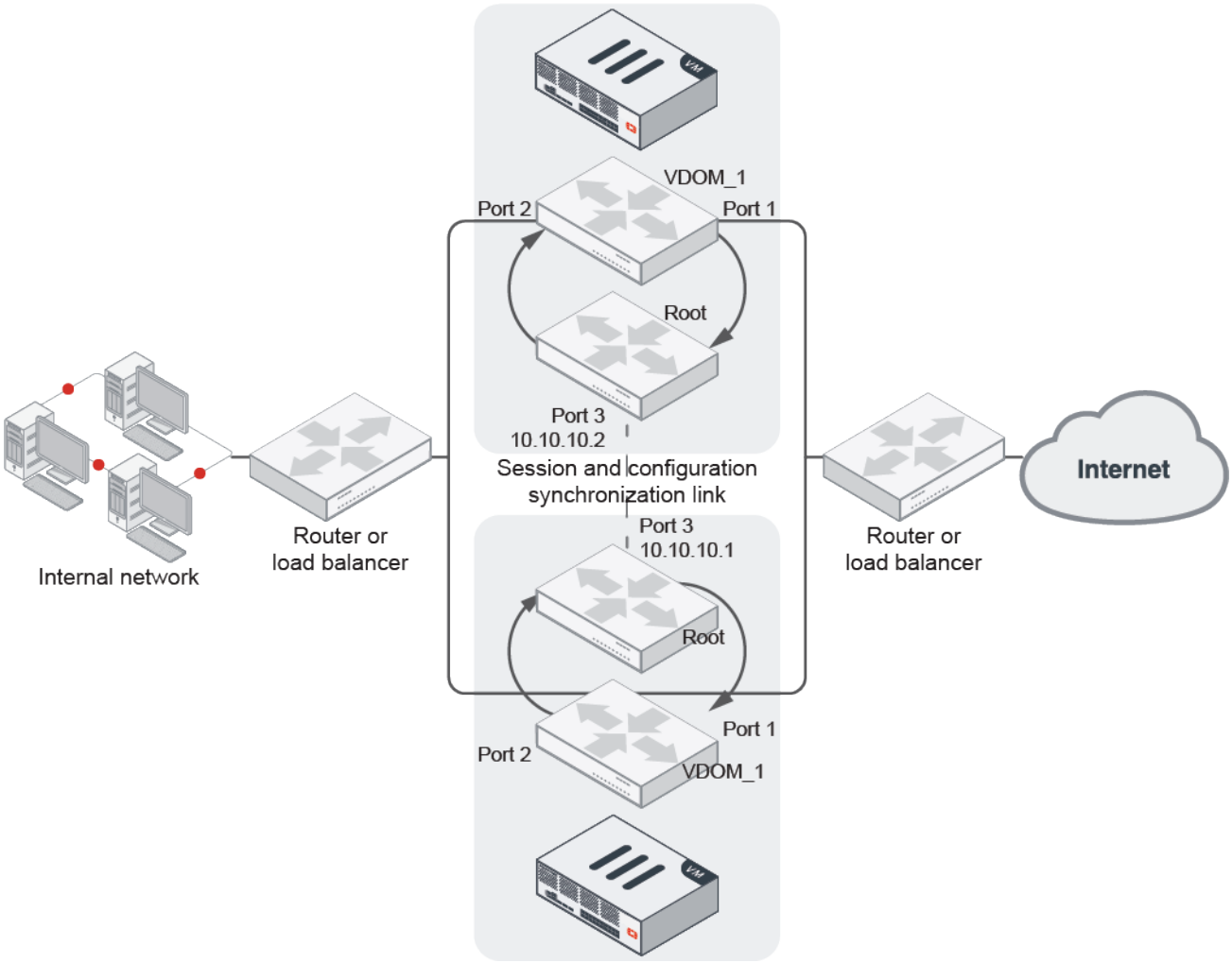
```
config system interface
   edit "port1"
      set vdom "root"
      set ip 10.6.30.61 255.255.255.0
      set allowaccess ping https ssh snmp http telnet
      set type physical
   next
```

```
    edit "port2"
      set vdom "root"
      set ip 10.1.100.61 255.255.255.0
      set allowaccess ping https ssh snmp http telnet
      set type physical
    next
    edit "port3"
      set vdom "root"
      set ip 172.16.200.61 255.255.255.0
      set allowaccess ping https ssh snmp http telnet
      set type physical
    next
  end
  config router static
    edit 1
      set gateway 172.16.200.254
      set device "port3"
    next
  end
  config firewall policy
    edit 1
      set srcintf "port2"
      set dstintf "port3"
      set srcaddr "all"
      set dstaddr "all"
      set action accept
      set schedule "always"
      set service "ALL"
      set nat enable
    next
  end
```

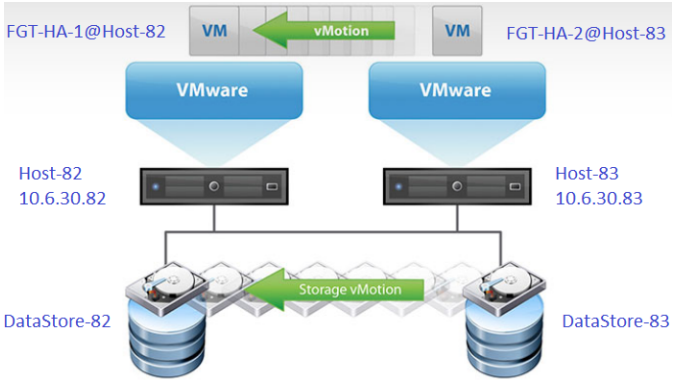
Setting up FortiGate-VM HA for a VMware vMotion environment

This guide provides sample configuration of vMotion FortiGate-VM high availability (HA) in a VMware environment. VMware vMotion enables the live migration of a running FortiGate-VM from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. It also provides transparency to users.

In VM environments that do not support broadcast communication, you can set up a unicast HA heartbeat when configuring HA. Setting up a unicast HA heartbeat consists of enabling the feature and adding a peer IP address. The peer IP address is the IP address of the HA heartbeat interface of the other FortiGate-VM in the HA cluster.



The following depicts the network topology for this sample deployment. In this sample deployment, there are two hosts, Host 82 (10.6.30.82) and Host 83 (10.6.30.83), which are members of Cluster 1 in the DataCenter 1. The vCenter server (10.6.30.81) manages DataCenter 1.



This configuration requires the following prerequisites:

- You have set up the vCenter server and created the data center and cluster.
- Host 82 and Host 83 are part of the cluster.
- A Gigabit Ethernet network interface card with a VMkernel port enabled for vMotion exists on both ESXi hosts.
- Two FortiGate-VM nodes, FGT-HA-1@Host-82 and FGT-HA-2@Host-83 are set up and factory reset. In this example, FGT-HA-1 is the primary side on Host 82, while FGT-HA-2 is the primary side on Host 83. HA is in sync.

To set up FortiGate-VM HA for a VMware vMotion environment:

1. Log into the vSphere web client.
2. Verify the current location of FGT-HA-1:
   a. Go to FGT-HA-1.
   b. On the *Summary* tab, check the *Host*. In this example, the host is currently Host 82 (10.6.30.82).
3. Repeat step 2 for FGT-HA-2. For FGT-HA2, the host should be Host 83 (10.6.30.83).
4. Log into FortiOS on FGT-HA-1 and FGT-HA-2 and run the following commands in the CLI:
   a. Run the following commands on FGT-HA-1:

```
config system interface
    edit "port3"
        set ip 192.168.40.91 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
    edit "port4"
        set ip 10.6.30.91 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
end

config system ha
    set group-name "FGT-VM-HA"
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
```

```
            set gateway 10.6.30.254
        next
    end
    set unicast-hb enable
    set unicast-hb-peerip 192.168.40.92
end

config router static
    edit 1
        set gateway 172.16.200.254
        set device "port1"
    next
end
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

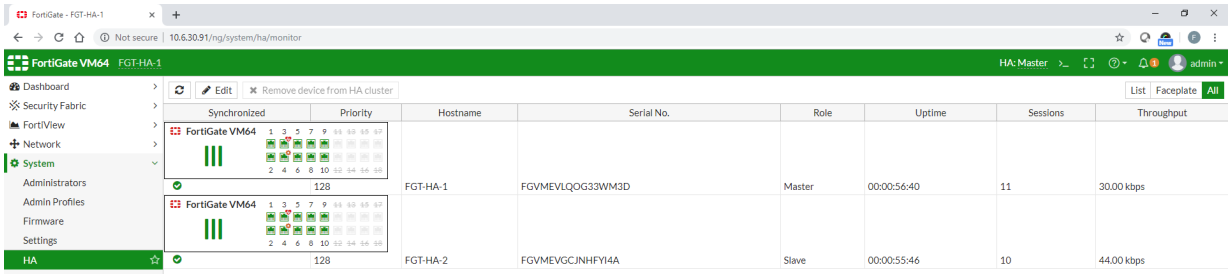
b. Run the following commands on FGT-HA-2:

```
config system interface
    edit "port3"
        set ip 192.168.40.92 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
    edit "port4"
        set ip 10.6.30.92 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
end

config system ha
    set group-name "FGT-VM-HA"
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
            set gateway 10.6.30.254
        next
    end
    set unicast-hb enable
```

```
    set unicast-hb-peerip 192.168.40.91
end
```

5. Check the HA status:
   a. To check the HA status in the GUI, in FortiOS, go to *System* > *HA*.



   b. To check the HA status in the CLI, run the `get system ha status` command. The output should be as follows. You should expect both FGT-HA-1 and FGT-HA-2 to have an in-sync configuration status.

```
FGT-HA-1 # get system ha status
HA Health Status: OK
Model: FortiGate-VM64
Mode: HA A-P
Group: 0
Debug: 0
Cluster Uptime: 0 days 1:35:12
Cluster state change time: 2019-05-16 14:53:05
Master selected using:
    <2019/05/16 14:53:05> FGVMEVLQOG33WM3D is selected as the master because it
has the largest value of uptime.
    <2019/05/16 14:45:53> FGVMEVLQOG33WM3D is selected as the master because
it's the only member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
unicast_hb: peerip=192.168.40.92, myip=192.168.40.91, hasync_port='port3'
Configuration Status:
    FGVMEVLQOG33WM3D(updated 2 seconds ago): in-sync
    FGVMEVGCJNHFYI4A(updated 0 seconds ago): in-sync
```

6. Before initiating the migration, open the CLI for both FGT-HA-1 and FGT-HA-2 to check on traffic during the migration. During the migration, you can enter the `diagnose sniffer packet any 'icmp and host 8.8.8.8'` command to check if traffic is stable. If no traffic is lost during migration and the FortiGate-VM SSH session does not break, the output resembles the following:

```
FGT-HA-1 # diag sniffer packet any 'icmp and host 8.8.8.8'
interfaces=[any]
filters=[icmp and host 8.8.8.8]
27.103356 10.1.100.22 -> 8.8.8.8: icmp: echo request
27.103423 172.16.200.91 -> 8.8.8.8: icmp: echo request
27.108160 8.8.8.8 -> 172.16.200.91: icmp: echo reply
27.108167 8.8.8.8 -> 10.1.100.22: icmp: echo reply
28.104695 10.1.100.22 -> 8.8.8.8: icmp: echo request
28.104699 172.16.200.91 -> 8.8.8.8: icmp: echo request
28.109381 8.8.8.8 -> 172.16.200.91: icmp: echo reply
28.109385 8.8.8.8 -> 10.1.100.22: icmp: echo reply
29.105058 10.1.100.22 -> 8.8.8.8: icmp: echo request
29.105069 172.16.200.91 -> 8.8.8.8: icmp: echo request
29.109682 8.8.8.8 -> 172.16.200.91: icmp: echo reply
29.109693 8.8.8.8 -> 10.1.100.22: icmp: echo reply
```

7. Migrate FGT-HA-1, the primary node, from Host 82 to Host 83, then migrate it from Host 83 back to Host 82. Refer to vMotion in a VMware ESXi environment on page 36 for migration details.

8. Migrate FGT-HA-2, the secondary node, from Host 83 to Host 82, then migrate it from Host 82 back to Host 83. Again, refer to vMotion in a VMware ESXi environment on page 36 for migration details.

Enhancing FortiGate-VM performance with DPDK and vNP offloading

DPDK and vNP enhance FortiGate-VM performance by offloading part of packet processing to user space while using a kernel bypass solution within the operating system. You must enable and configure DPDK with FortiOS CLI commands.

FortiOS 7.4 supports DPDK for VMware ESXi environments.

FortiOS 7.4 supports IPv6.

The current DPDK+vNP offloading-capable version of FortiOS only supports FortiGate instances with multiple vCPUs. Minimum required RAM sizes differ from those on regular FortiGate-VM models without offloading. Allocating as much RAM size as the licensed limit for maximum performance, as shown, is recommended. FortiOS 7.4 does not restrict RAM size by license. Therefore, you can allocate as much memory as desired on 7.4-based DPDK-enabled FortiGate-VMs:

Model name	RAM size (licensed limit)
FG-VM02(v)	No restriction
FG-VM04(v)	No restriction
FG-VM08(v)	No restriction
FG-VM16(v)	No restriction
FG-VM32(v)	No restriction

You can enable DPDK up to 64 vCPUs.

FortiOS supports encrypted traffic for IPsec VPN and not for SSL VPN. Disabling the DPDK option using the CLI or adopting regular FortiGate-VM builds is recommended when using SSL VPN features. For encrypted traffic support for IPsec VPN with DPDK, FortiOS also adds support for the following:

- CBC cipher suite
- Increasing maximum number of IPsec VPN tunnels
- Terminating tunnel on LAG

Enabling DPDK+vNP offloading may result in fewer concurrent sessions when under high load than when DPDK+vNP offloading is not enabled and the same FortiGate-VM license is used.

Enabling DPDK in polling mode results in high CPU usage.

Enabling DPDK+vNP offloading using the FortiOS CLI

Provided that you obtained a DPDK+vNP offloading-capable FortiOS build, the following provides the configuration to enable the capability:

- DPDK global settings on page 45
- DPDK CPU settings on page 48
- DPDK diagnostic commands on page 49

FortiOS supports SNMP to poll DPDK-related status. For details, see the corresponding MIB file that Fortinet provides.

DPDK global settings

To enable DPDK operations for the FortiGate-VM:

1. In the FortiOS CLI, enter the following commands to enable DPDK operation:
```
config dpdk global
   set status enable
   set interface port1
end
```
2. The CLI displays the following message:
```
Status and interface changes will trigger system reboot and take effect after the
     reboot.
Do you want to continue? (y/n)
```
Press `y` to reboot the device.

Before system reboot, you must check if other DPDK settings are configured properly. You must enable at least one network interface for DPDK. The example enables port1. You can enable other interfaces as desired. If you do not set an interface, a prompt displays and the change is discarded. See To enable a network interface to run DPDK operation: on page 46.

To enable DPDK multiqueue mode:

Enabling multiqueue at network RX/TX helps DPDK better balance the workload onto multiple engines.

1. In the FortiOS CLI, enter the following commands to enable DPDK operation:
```
config dpdk global
   set multiqueue enable
end
```
2. The CLI displays the following message:
```
Multiqueue change will trigger IPS restart and will take effect after the restart.
     Traffic may be interrupted briefly.
Do you want to continue? (y/n)
```
Press `y` to reboot IPS engine.

To set the percentage of main memory allocated to DPDK huge pages and packet buffer pool:

You can configure the amount of main memory (as a percentage) allocated to huge pages, which are dedicated to DPDK use. You can also configure the amount of main memory (as a percentage) allocated to the DPDK packet buffer pool.

Enter the following commands to set these amounts:

```
config dpdk global
   set hugepage-percentage [X]
   set mbufpool-percentage [Y]
end
```

Changing `mbufpool-percentage` requires IPS engine to restart (no reboot).

Huge page memory is mounted at system startup and remains mounted as long as the FortiGate-VM is running. Packet buffer pool memory is drawn from huge pages. Therefore, the packet buffer pool amount (Y) must not exceed the huge pages amount (X).

In practice, it is mandated that Y is lesser than or equal to X - 5 to leave 5% memory overhead for other DPDK data structures. The range of X is between 10 and 50, and the range of Y is between 5 and 45.

Setting X too high may force FortiOS to enter conserve mode. Setting X too low may result in insufficient memory for DPDK operation and failure of initialization.

During FortiOS DPDK Helper environment initialization, RTE memory zones are drawn from huge memory pages. The system tries to reserve continuous memory chunks for these memory zones with best effort. Therefore, the amount of huge page memory is slightly larger than the amount of memory that RTE memory zones use. To gain insight into how RTE memory zones reserve memory spaces, run the `diagnose dpdk statistics show memory` command.

To enable a network interface to run DPDK operation:

You must enable at least one network interface to run DPDK operation.

```
config dpdk global
   set interface "portX" "portY"
end
```

You must enable at least one network interface for DPDK. Otherwise, DPDK early initialization during system startup fails and falls back to a disabled state. In this example, if there are two network interfaces that you intend to use, you can specify `set interface port1 port2`.

Enabling DPDK is only available for physical network interfaces.

To enable DPDK monitor engine:

Enabling DPDK monitor engine is optional.

1. In the FortiOS CLI, enter the following commands to enable DPDK monitor engine:
```
config dpdk global
   set sleep-on-idle enable
end
```
2. The CLI displays the following message:
```
sleep-on-idle change will trigger IPS restart and will take effect after the restart.
     Traffic may be interrupted briefly.
Do you want to continue? (y/n)
```
Press `y` to reboot IPS engine.

By default, DPDK monitor engine is disabled. When enabled, only one DPDK engine polls DPDK-enabled interfaces. When packets arrive, corresponding DPDK entries are activated. This helps when services other than firewall or IPS engine, such as antivirus, WAD, or web filter, are running and performance degradation is observed while DPDK performance statistics show that DPDK engines are not fully used. Latency may increase due to the time needed to activate the proper DPDK engines by the monitor engine.

To enable elastic buffer (temporary memory buffer):

Enabling elastic buffer is optional.

1. In the FortiOS CLI, enter the following commands to enable elastic memory buffer:
```
config dpdk global
   set elasticbuffer enable
end
```
2. The CLI displays the following message:
```
elasticbuffer change will trigger IPS restart and will take effect after the restart.
     Traffic may be interrupted briefly.
Do you want to continue? (y/n)
```
Press `y` to reboot IPS engine.

By default, elastic buffer is disabled. When enabled, an elastic buffer takes effect to store packets in case of traffic burst. The feature helps to reduce packet drops when received packets peak under system overload by storing packets in the buffer and processing them afterward. This feature is experimental.

To enable per-session accounting:

Enabling per-session accounting is optional.

1. In the FortiOS CLI, enter the following commands to enable per session accounting:
```
config dpdk global
   set per-session-accounting enable|disable|traffic-log-only
end
```
2. The CLI displays the following message:
```
per-session-accounting change will trigger IPS restart and will take effect after the
     restart. Traffic may be interrupted briefly.
Do you want to continue? (y/n)
```
Press `y` to reboot IPS engine.

By default, per-session accounting is configured only for traffic logs, which results in per-session accounting being enabled when you enable traffic logging in a policy.

Per-session accounting is a logging feature that allows FortiOS to report the correct bytes per packet numbers per session for sessions offloaded to a vNP process. This information appears in traffic log messages, FortiView, and `diagnose` commands. Per-session accounting can affect vNP offloading performance. You should only enable per-session accounting if you need the accounting information. A similar feature is available for physical FortiGate NP6 processors.

DPDK CPU settings

On the FortiGate-VM, a DPDK engine is attached to an IPS engine, which shares the same process and is mapped to a CPU. A processing pipeline of four stages handles a packet from RX to TX:

1. DPDK RX
2. vNP
3. IPS
4. DPDK TX

You can freely determine the CPUs enabled for each pipeline stage by running the following commands:

```
config dpdk cpus
   set [X] [Y]
end
```

Here X is one of the pipeline stages: `rx-cpus`, `vnp-cpus`, `ips-cpus`, and `tx-cpus`.

Y is a string expression of CPU IDs, which contains comma-delimited individual CPU IDs or ranges of CPU IDs separated by a dash.

The example enables CPUs 0, 2, 4, 6, 7, 8, 9. 10, and 15 to run the vNP pipeline stage:

```
set vnp-cpus 0,2,4,6-10,15
```

In FortiOS 7.4, Y can also be a special token string `all`, which means to use all available CPUs to run that pipeline stage. The system automatically determines the number of available CPUs. `all` is the default value of each pipeline stage's CPU setting.

The example uses all available CPUs to run the IPS pipeline stage:

```
set ips-cpus all
```

You must enable at least one CPU for each pipeline stage. Otherwise, DPDK early initialization fails.

Isolate CPUs used by DPDK engine

To improve DPDK performance, the CPUs that are used by the DPDK engine can be isolated from other services, except for processes that have affinity explicitly set by either a user configuration or by their implementation. FortiOS 7.0.2 and later versions support this feature.

```
config dpdk cpus
    set isolated-cpus <CPUs>
end
```

Input CPU IDs or ranges separated by commas, or `none` to not isolate CPUs for DPDK. For example, enter `1-3,5,6-9` to isolate CPUs 1,2,3,5,6,7,8, and 9.

Both the lower and upper bounds of a range must be explicitly specified. The range of isolated CPU IDs is [1-0], and CPU ID 0 is not allowed. The isolated CPU IDs must be DPDK enabled CPUs.

Reserving CPUs for DPDK may not always produce optimal performance. Users should experiment with a combination that works best for their deployment. For example, on a FortiGate VM with eight CPUs, the following configurations could be used to optimize different deployments:

To optimize CPS with logging to disk (session/sec):

```
config dpdk cpus
    set rx-cpus "1-1"
    set vnp-cpus "1-7"
    set ips-cpus "1-7"
    set tx-cpus "1-7"
    set isolated-cpus "1-7"
end
```

To optimize proxy antivirus performance:

```
config dpdk cpus
    set rx-cpus "1-5"
    set vnp-cpus "1-5"
    set ips-cpus "1-5"
    set tx-cpus "1-5"
    set isolated-cpus "1-5"
end
```

To optimize proxy DLP performance:

```
config dpdk cpus
    set rx-cpus "1-5"
    set vnp-cpus "1-5"
    set ips-cpus "1-5"
    set tx-cpus "1-5"
end
```

DPDK diagnostic commands

To view DPDK-related logs:

Enter the following command to view DPDK-related logs:

```
diagnose dpdk log show [log type]
```

Currently, FortiOS provides two DPDK-related logs:

Log	Records kept
early-init	DPDK's early initialization procedure during system startup
fdh	Warnings and errors met during the initialization of FortiOS DPDK helper (FDH), i.e. DPDK engines

Ensure that you double-check whether DPDK early initialization was successful. If successful, the end of the early-init log shows the following:

```
DPDK sanity test passed
```

If the DPDK early initialization was unsuccessful, refer to DPDK global settings on page 45 to see if the DPDK-related options were properly set.

The early init-log also keeps records of last-edited DPDK configuration, enabled CPUs/ports, binding/unbinding of drivers, device PCI info, and so on.

To view DPDK-related statistics:

Enter the following command to view DPDK-related statistics:

```
diagnose dpdk statistics show [stats type]
```

Currently, FortiOS provides four types of DPDK-related statistics:

- `engine`: provides per-DPDK engine statistics
- `port`: provides per-DPDK port statistics
- `vnp`: provides per-vNP engine statistics
- `memory`: provides a quick view of memory size reserved by each RTE memory zone

To reset statistics, enter the following command:

```
diagnose dpdk statistics clear all
```

This command resets engine and port statistics to zeroes, but does not affect vNP and memory statistics.

To check if traffic is properly forwarded, load-balanced, and offloaded to fast path:

A useful way to check whether traffic is properly forwarded is to check the port statistics. This shows the number of received/transmitted/dropped packets in each DPDK-enabled port.

```
-------------------------------------------------------------------------------
FortiOS DPDK Helper Port Stats
-------------------------------------------------------------------------------

                                    Total             port2

---------- DPDK RX Stage ----------------------------------------------------
dpdkrx_rx_pkts:                         0                 0

---------- DPDK TX Stage ----------------------------------------------------
dpdktx_tx_pkts:                         0                 0
dpdktx_drop_pkts:                       0                 0
dpdktx_drop_oversized_pkt:              0                 0
```

Checking engine statistics is helpful in understanding how traffic is load-balanced among DPDK engines at each pipeline stage.

```
--------------------------------------------------------------------------------
FortiOS DPDK Helper Engine Stats
--------------------------------------------------------------------------------

                                    Total
CPU ID:

---------- DPDK RX Stage ---------------------------------------------------
dpdkrx_rx_pkts:                     2
dpdkrx_tx_pkts:                     2
dpdkrx_drop_pkts:                   0

---------- VNP Stage -------------------------------------------------------
vnp_rx_pkts:                        2
vnp_tx_pkts:                        1
vnp_tx_drop_pkts:                   0
vnp_to_ips_pkts:                    0
vnp_to_ips_drop_pkts:               0

---------- IPS Stage -------------------------------------------------------
ips_rx_pkts:                        0
ips_tx_pkts:                        0
ips_drop_pkts:                      0
ips_rej_pkts:                       0

---------- DPDK TX Stage ---------------------------------------------------
dpdktx_rx_pkts:                     1
dpdktx_tx_pkts:                     1
dpdktx_drop_pkts:                   0
dpdktx_drop_oversized_pkt:          0


                                Engine  0         Engine  1         Engine  2         Engine  3
CPU ID:                             0                 1                 2                 3

---------- DPDK RX Stage ---------------------------------------------------
dpdkrx_rx_pkts:                     2                 0                 0                 0
dpdkrx_tx_pkts:                     2                 0                 0                 0
dpdkrx_drop_pkts:                   0                 0                 0                 0

---------- VNP Stage -------------------------------------------------------
vnp_rx_pkts:                        0                 0                 0                 0
vnp_tx_pkts:                        0                 0                 0                 0
vnp_tx_drop_pkts:                   0                 0                 0                 0
vnp_to_ips_tx_pkts:                 0                 0                 0                 0
vnp_to_ips_tx_drop_pkts:            0                 0                 0                 0

---------- IPS Stage -------------------------------------------------------
ips_rx_pkts:                        0                 0                 0                 0
ips_tx_pkts:                        0                 0                 0                 0
ips_drop_pkts:                      0                 0                 0                 0
ips_rej_pkts:                       0                 0                 0                 0

---------- DPDK TX Stage ---------------------------------------------------
dpdktx_rx_pkts:                     0                 0                 0                 1
dpdktx_tx_pkts:                     0                 0                 0                 1
dpdktx_drop_pkts:                   0                 0                 0                 0
dpdktx_drop_oversized_pkt:          0                 0                 0                 0
```

Checking vNP statistics provides insights to how traffic is offloaded from the slow path (traversing the kernel) to the fast path (firewall and IPS operations quickly processed by the vNP engine). In particular, observe the number of session search engine (SSE) entries pushed from kernel or IPS to vNP engine, shown bolded (`ctr_sse_entries`). The number of packets going through the SSE fast path is also important and is bolded (`ctr_fw_and_ips_fpath`).

```
--------------------------------------------------------------------------------
FortiOS DPDK Helper VNP Stats
--------------------------------------------------------------------------------

                                  Total
CPU ID:

---------- VNP Internal ----------------------------------------------------------
ctr_ctx_alloc:                    2
ctr_ctx_alloc_fail:               0
ctr_ctx_free:                     2
ctr_to_kernel:                    2
ctr_from_kernel:                  1
ctr_sse:                          0
ctr_sse_cmd:                      0
ctr_sse_delmiss:                  0
ctr_sse_msg:                      0
ctr_sse_pruned:                   0
ctr_fw_and_ips_fpath:             0

ctr_sse_entries:                  0

err_sse_batch_size:               0
err_sse_unknown_cmd:              0
err_sse_full:                     0
err_sse_tbl_alloc_fail:           0
err_sse_inv_oid:                  0
err_fp_no_act:                    0
drop_inv_port:                    0
drop_inv_ip_cksum:                0
drop_inv_tcp_cksum:               0
drop_inv_udp_cksum:               0
drop_oversized_pkt:               0
--------------------------------------------------------------------------------


                              Engine  0        Engine  1        Engine  2        Engine  3
CPU ID:                       0                1                2                3

---------- VNP Internal ----------------------------------------------------------
ctr_ctx_alloc:                    0                0                0                0
ctr_ctx_alloc_fail:               0                0                0                0
ctr_ctx_free:                     0                0                0                0
ctr_to_kernel:                    0                0                0                0
ctr_from_kernel:                  0                0                0                0
ctr_sse:                          0                0                0                0
ctr_sse_cmd:                      0                0                0                0
ctr_sse_delmiss:                  0                0                0                0
ctr_sse_msg:                      0                0                0                0
ctr_sse_pruned:                   0                0                0                0
ctr_fw_and_ips_fpath:             0                0                0                0

ctr_sse_entries:                  0                0                0                0

err_sse_batch_size:               0                0                0                0
err_sse_unknown_cmd:              0                0                0                0
err_sse_full:                     0                0                0                0
err_sse_tbl_alloc_fail:           0                0                0                0
err_sse_inv_oid:                  0                0                0                0
err_fp_no_act:                    0                0                0                0
drop_inv_port:                    0                0                0                0
drop_inv_ip_cksum:                0                0                0                0
drop_inv_tcp_cksum:               0                0                0                0
drop_inv_udp_cksum:               0                0                0                0
drop_oversized_pkt:               0                0                0                0
--------------------------------------------------------------------------------
```

To see DPDK CPU settings, run the following commands. In this case, N is the number of CPUs that the FortiGate-VM uses.

```
show dpdk cpus
   config dpdk cpus
      set rx-cpus "0-N"
      set vnp-cpus "0-N"
      set ips-cpus "0-N"
      set tx-cpus "0-N"
   end
```

To view DPDK performance:

The `diagnose dpdk performance show` command provides near real-time performance of each DPDK engine, in particular, the CPU usage. The system provides the following response:

```
----------------------------------------
CPU usages
----------------------------------------
                                          Engine  0        Engine  1        Engine  2        Engine  3
2018:12:10 15:17:52                 rx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 vnp:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 ips:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 tx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 idle:      100.0            100.0            100.0            100.0
----------------------------------------


                                          Engine  4        Engine  5        Engine  6        Engine  7
2018:12:10 15:17:52                 rx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 vnp:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 ips:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 tx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 idle:      100.0            100.0            100.0            100.0
----------------------------------------


                                          Engine  8        Engine  9        Engine 10        Engine 11
2018:12:10 15:17:52                 rx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 vnp:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 ips:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 tx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 idle:      100.0            100.0            100.0            100.0
----------------------------------------


                                          Engine 12        Engine 13        Engine 14        Engine 15
2018:12:10 15:17:52                 rx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 vnp:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 ips:         0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 tx:          0.0              0.0              0.0              0.0
2018:12:10 15:17:52                 idle:      100.0            100.0            100.0            100.0
----------------------------------------
```

This provides better insight into how many CPUs to allocate to each pipeline stage.

Best practices

This document serves as a best practices guide covering use cases where the Fortinet FortiGate virtual appliance is used in mobile networks. This document aims to provide technically focused details and guidance around building the FortiGate-VM with recommendations on tuning the deployment to maximize performance.

The document assumes prior knowledge of the following topics:

- Computer hardware
- Hypervisors and virtualization
- SR-IOV
- DPDK

FortiGate-VM

- Supported on private and public clouds across many vendors
- Delivered in hypervisor-specific formats
- Licensed based on number of vCPUs (no restriction on RAM)
- Compatible with different network virtualization technologies, including:
  - virtIO / VMXNET3
  - PCI passthrough
  - SR-IOV
  - OVS-DPDK
- Supports common NICs, including the following for SR-IOV:
  - Intel cards compatible with igb, ixgbe and i40e drivers (1/10/25/40 Gbps)
  - Intel cards compatible with ice driver (100 Gbps) (FortiOS 6.4.1 and later versions)
  - Mellanox 100G cards (mlx_4 and mlx_5)
  - Broadcom 100G cards (P2100G) (FortiOS 6.4.3 and later versions)
- Internal implementation of DPDK to deliver the vSPU, giving massive performance benefits with no dependency on hypervisor

The FortiGate-VM product evolves quickly to support new hardware from third parties. Fortinet is continuously enhancing the product. Therefore, installing the latest GA release from one of the two latest FortiOS is recommended.

FortiGate vSPU

Virtual security processing units (vSPU), introduced in FortiOS 6.2.3, refer to the combination of the FortiOS virtual Network Processor (vNP) and DPDK libraries operating within the FortiGate-VM. vNP is the software emulation of a subset of the Fortinet Network Processor.

DPDK provides data plane libraries and the polling-mode driver (PMD), which enables offload of packet processing from the system kernel to user space. This allows the creation of high-speed networking applications, such as the vNP.

vSPU is implemented within the FortiGate-VM, allowing the virtual appliance to be optimized:

- vNP runs in user space, and the kernel is bypassed when vNP is handling the traffic.
- PMD means that traffic is taken from the NIC card without relying on CPU interrupts.

That means that for certain FortiGate-VM use cases, you can employ vSPU to make more effective use of CPU resource and achieve higher throughput.

You can activate vSPU by configuration on a per-CPU basis. Each CPU activated for vNP function is presented as a processing engine.

The following summarizes how FortiOS handles traffic when multiple CPUs are enabled for vSPU. You cannot change this behavior through configuration:

FortiOS version	Description
7.0.1 and earlier versions	Traffic balancing is based on the L3 header information. For best performance, a significant variation in source and destination IP addresses are needed to load all vSPUs evenly.
7.0.2 and later versions	Traffic balancing is based on the L3 and L4 header information. The hash used to balance across the DPDK engines is based on L4 source and destination port numbers in addition to L3 addresses. Therefore, loading more vSPUs evenly should be easier.

The vSPU is analogous to the physical NP found in physical appliances. Session creation is performed in the kernel, then offloaded to the vSPU, as the hardware offloads traffic to the NP.

For more information, including a diagram of the fastpath architecture, see Performance as a Key Attribute of Fortinet.

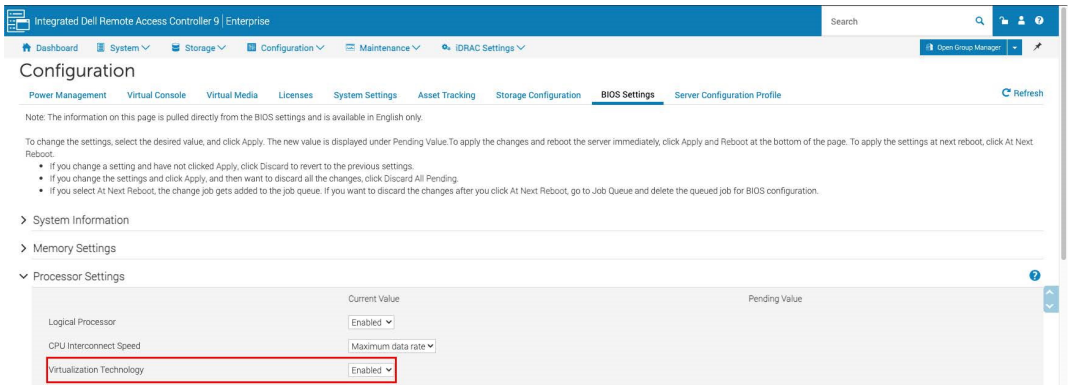
The vNP is beneficial if the IP payload is UDP or TCP. Other traffic traverses the device without benefiting from fastpath.

Server BIOS considerations

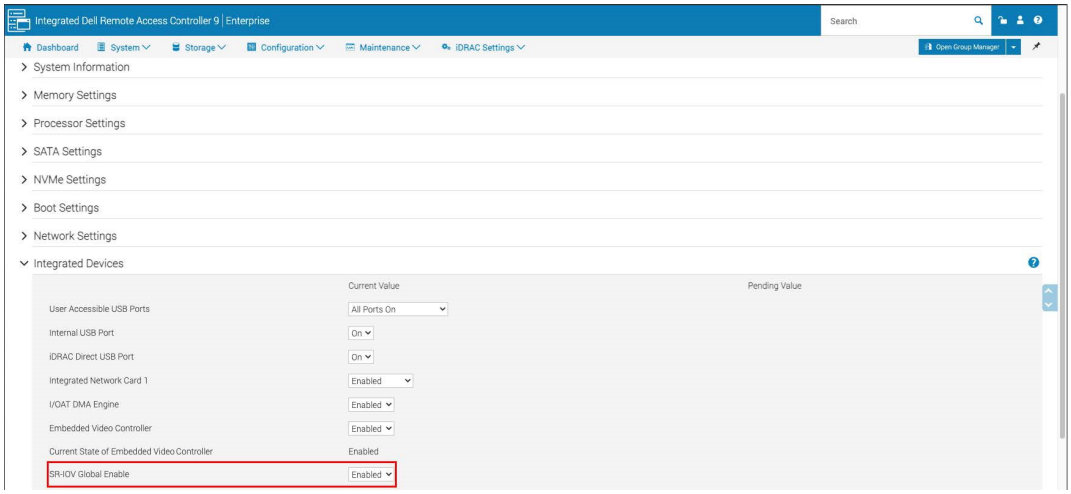
Typically, BIOS settings are necessary to enable SR-IOV and optimize resource usage.

As the exact configuration depends on the BIOS vendor and version used, researching these settings in the applicable vendor documentation is recommended. This document uses example settings based on a Dell PowerEdge R740.

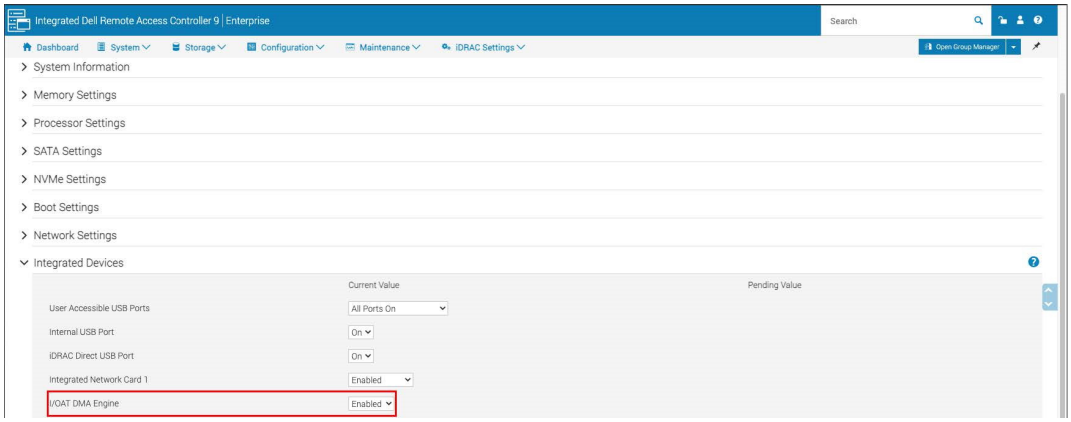
- Ensure that I/O memory management unit) is enabled. For the example hardware, the relevant setting is *Virtualization Technology*.



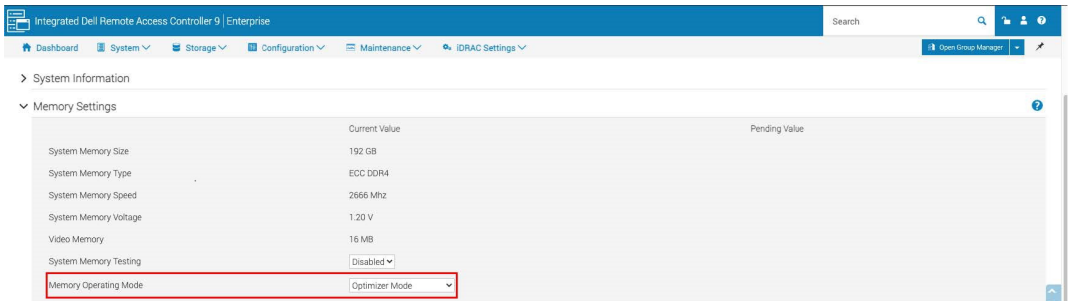
- Ensure that SR-IOV is enabled. For the example hardware, the relevant setting is *SR-IOV Global Enable*.



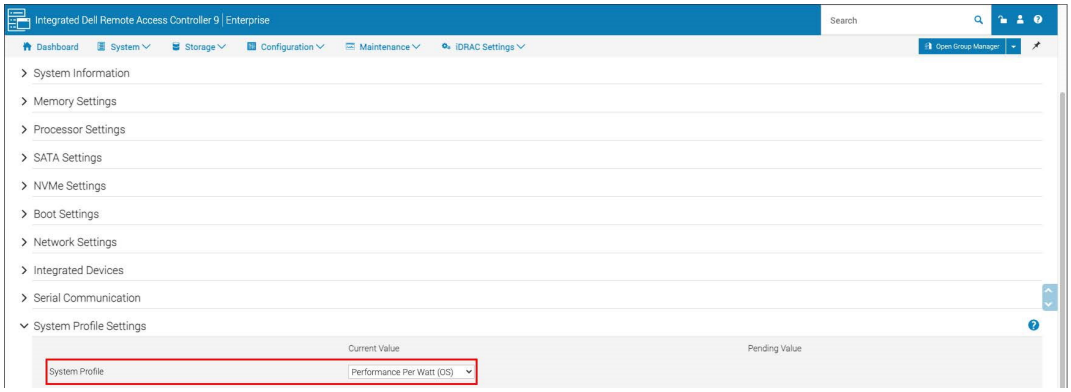
- Ensure that *I/OAT DMA Engine* is enabled. Intel and Mellanox hardware support this feature.



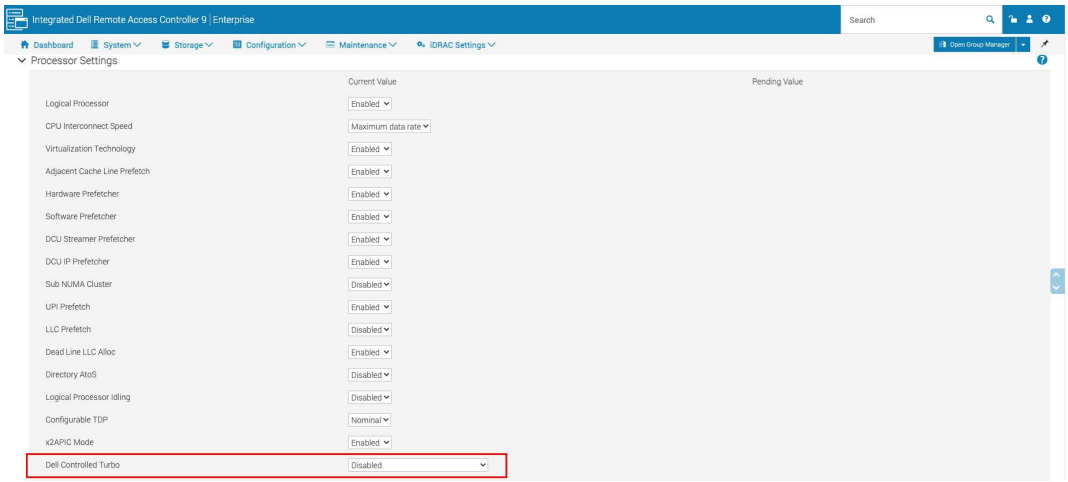
- High end servers have several memory operation modes . You must select the mode that gives the most memory to the operating system and maximum performance. For the example hardware, *Optimizer Mode* is selected from the *Memory Operating Mode* dropdown list.



- High end servers differ in BIOS recommendations about achieving the highest performance and lowest latency for options such as power saving and turbo boost. For the example hardware, *Performance Per Watt (OS)* is selected from the *System Profile* dropdown list. This means that these settings are managed within the host OS.



- You should disable the BIOS turbo setting if vSPU is in use. PMD takes the CPU load to 100%, which means that the processor would continually be overclocked, which is undesirable. If not using vSPU, leaving this option disabled is still recommended to avoid unpredictable CPU usage. For the example hardware, *Dell Controlled Turbo* is disabled.



The BIOS tasks ensure key features are enabled to ensure that the generic performance settings are set correctly to get the system to best complement the FortiGate-VM.

Hypervisor tuning

VMware ESXi is a bare metal or type 1 hypervisor that has been available since 2001. ESXi provides the compute workload for the wider Telco Cloud Platform collection of products available from VMware and is the home of the FortiGate-VM.

This document focuses on maximizing a FortiGate-VM deployment's performance. Therefore, this document limits discussion to ESXi and vCenter, where vCenter provides centralized management of ESXi compute nodes.

You should consult the FortiOS Release Notes to determine the Fortinet recommendations on ESXi versions. As the list in the Release Notes is long, this document focuses on the version included in the Telco Cloud Platform 5G Edition 2.1 release to fit the intended audience.

Referencing ESXi and vCenter articles to map the named releases with build numbers is recommended.

The NIC is probably the most important consideration to achieve a performant firewall. Handling network I/O correctly and efficiently is important. The main considerations, which this document covers in more detail later, are:

- Traffic NICs should support SR-IOV. PCI-passthrough may be an alternative option, but has little flexibility.
- Avoid OEM NIC. For example, a Dell branded Intel XXV710 NIC may not have the required firmware version available to achieve a working solution.
- The number of NIC ports and, therefore, number of network queues/buffers used for traffic is important when considering a FortiGate-VM deployment without vSPU, allowing effective use of the CPUs.

VMware, NIC vendor, and firmware/driver versions are typically outside of the deployment scope for Fortinet. However, they are important to achieve a stable and performant solution. Therefore, you should take due caution around the version choices to select these optimally. These items will be the first things to check if the performance is suboptimal or if, in fact, the deployment is unexpectedly not functioning as designed.

vSphere versions

ESXi version

This document assumes that the system administrator is familiar with enabling shell and SSH access to ESXi.

The current ESXi version within the Telco Cloud Platform documentation is ESXi 7.0 Update 2d, which equates to build number 18538813.

```
[root@esxi-tiger-14-7:~] esxcli system version get
   Product: VMware ESXi
   Version: 7.0.2
   Build: Releasebuild-18538813
```

```
   Update: 2
   Patch: 25
```

- VMware provides installation and update instructions, which you should consult.
- In this instance, the ESXi installation was achieved using the ESXi 7.0 Update 2a ISO from VMware and updated as follows:

```
[root@esxi-tiger-14-7:~] esxcli software profile update -p ESXi-7.0U2d-18538813-standard -d
https://hostupdate.vmware.com/software/VUM/PRODUCTION/main/vmw-depot-index.xml
Update Result
   Message: The update completed successfully, but the system needs to be rebooted for the
changes to be effective.
   Reboot Required: true
<output omitted for brevity>

[root@esxi-tiger-14-7:~] reboot
```

vCenter version

The current vCenter version within the Telco Cloud Platform documentation is vCenter Server 7.0 Update 2d, which equates to build number 18455184.

Hypervisor checks

- Add the ESXi host to vCenter and assign license as appropriate
- Check the CPUs support virtualisation

```
[root@esxi-tiger-14-7:~] esxcli hardware cpu global get
   CPU Packages: 2
   CPU Cores: 36
   CPU Threads: 36
   Hyperthreading Active: false
   Hyperthreading Supported: true
   Hyperthreading Enabled: true
   HV Support: 3
```

As per this article, this ensures that CPU virtualization is correctly enabled.

NIC versions

NICs generally have three components to consider when considering SR-IOV:

- Firmware (or NVM)
- PF driver
- VF driver

Care should be taken to ensure the drivers and firmware/non-volatile memory image are aligned as per manufacturer's recommendations. For example, the Intel X700 series recommendations are detailed in the Feature Support Matrix.

The VF driver is part of the FortiGate-VM instance. The driver versions are documented here. There is only one VF driver version per FortiOS version. This offers the least flexibility when aligning versions, making it the starting point.

As an example, the system used in this document has Intel XXV710 NIC cards and is running FortiOS 7.0.5. Consulting the two resources above:

- VF driver: IAVF 4.1.1

Table 10. Software/NVM Compatibility for XXV710

SW Release Version	NVM Version	NVM Update Tool Version	i40e (Windows)	i40e (Linux)[1]	i40evf/ iavf[2] (Linux)[1,3]	i40en (ESX)	ixl (FreeBSD)	QSFP Config. Utility (QCU)	Ethernet Port Config. Tool (EPCT)
21.3 / 22.2	5.51	1.28.19.4	21.3 22.0 22.2	1.6.42 2.0.19 2.0.23	1.6.41 2.0.16 2.0.22	For ESX 6.0: 1.5.8 For ESX 6.5: 1.5.8 For ESX 6.7: 1.7.1	1.7.10 1.7.11	N/A	N/A
22.6 22.9 22.10 23.1 23.2	6.01 6.02	1.30.2.11 1.30.22.1 1.30.22.3	22.6 22.9 22.10 23.1 23.2	2.1.26 2.3.6 2.4.3 2.4.6 2.4.10	3.0.8 3.2.5 3.4.2 3.5.6 3.5.13	1.7.11	1.7.12 1.9.5 1.9.7 1.9.8	2.30.2.9 2.30.22.0 2.30.23.0 2.32.6.6	N/A
23.4	6.80	1.32.20.28	23.4	2.7.12	3.6.11	1.7.11	1.10.4	2.32.20.28	N/A
23.5.2	6.80	1.32.20.30	23.5.2	2.7.29	3.6.15	1.7.11	1.10.4	2.32.20.28	N/A
24.0	7.00	1.33.15.1	24.0	2.8.43	3.7.34	1.8.6	1.11.9	2.33.15.1	N/A
24.3	7.10	1.34.17.3	24.3	2.10.19.30	3.7.61.20	1.9.5	1.11.20	2.34.17.3	1.34.17.5
25.0	7.20	1.34.22.6	25.0	2.10.19.82	3.7.61.20	1.10.6	1.11.22	2.34.17.3	1.34.22.5
25.1	7.30	1.35.23.3	25.1	2.11.29	3.9.5	1.10.9.0	1.11.29	EOL	1.35.23.2
25.2	8.00	1.35.33.4	25.2	2.12.6	4.0.1	1.10.9.0	1.12.2	EOL	1.35.33.3
25.4	8.10	1.35.42.7	25.4	2.14.13	4.0.1	1.10.9.0	1.12.3	EOL	1.35.42.7
25.5	8.15	1.35.42.7	25.5	2.14.13	4.0.1	1.10.9.0	1.12.3	EOL	1.35.49.0
26.0	8.20	1.35.57.4	26.0	2.14.13	4.0.2	1.12.3.0	1.12.13	EOL	1.35.57.1
26.2	8.30	1.37.1.1	26.2	2.15.9	4.1.1	1.13.1.0	1.12.16	EOL	1.37.1.0
26.4	8.40	1.37.13.5	26.4	2.16.11	4.2.7	1.13.1.0	1.12.24	EOL	1.37.13.3
26.6	8.50	1.37.28.0	26.6	2.17.4	4.2.7	For ESXi 6.5: 1.14.3.0 For ESXi 6.7: 1.14.3.0 For ESXi 7.0: 2.1.4.0	1.12.29	EOL	1.37.28.0

- Software Release Version: 26.2
- PF driver: i40en 1.13.1.0
- NVM version: 8.30

Any deviation from this alignment would need to be diligently tested and may still cause later supportability issues.

Instructions for Mellanox public repositories can be found at Mellanox Technologies Ltd. Public Repository. You can find firmware update instructions at Firmware Update Instructions.

Having said that, the example system in this document breaks this rule. As it is not carrying production traffic, then it allows future versions to be considered without any negative impact. The example prints are based upon Intel's Software Release Version 26.6 which importantly contains some fixes for issues observed on other hypervisors.

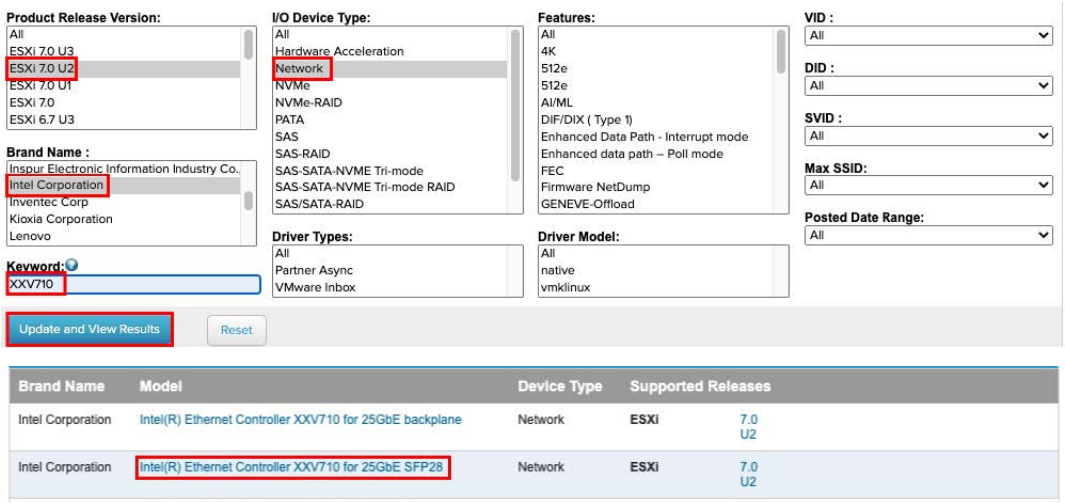
```
[root@esxi-tiger-14-7:~] esxcli network nic get -n vmnic4
   Advertised Auto Negotiation: true
   Advertised Link Modes: Auto, 10000BaseSR/Full, 25000BaseSR/Full
```

```
Auto Negotiation: true
Cable Type: FIBRE
Current Message Level: 0
Driver Info:
      Bus Info: 0000:3b:00:0
      Driver: i40en
      Firmware Version: 8.50 0x8000b703 1.3082.0
      Version: 2.1.5.0
Link Detected: true
Link Status: Up
Name: vmnic4
PHYAddress: 0
Pause Autonegotiate: false
Pause RX: false
Pause TX: false
Supported Ports: FIBRE
Supports Auto Negotiation: true
Supports Pause: true
Supports Wakeon: false
Transceiver:
Virtual Address: 00:50:56:54:6f:71
Wakeon: None
```

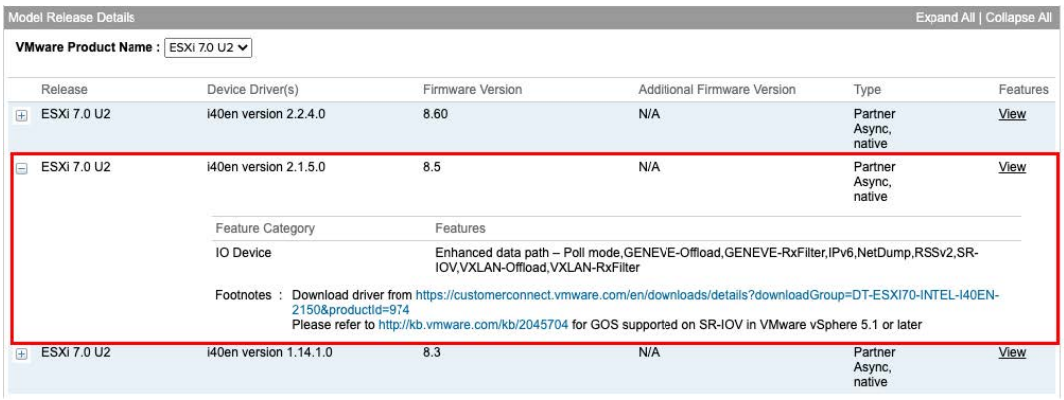
ESXi 7.0U2 packages the i40enu driver version. This is not helpful and is something that VMware has retracted.

Search, download, and transfer to the versions desired to the server:

- Google: intel xl710 nvm 8.50
- VMware Compatibility Guide



Compile and install the driver, following the vendor's instructions:



Note the following:

- The NIC features are listed. This includes SR-IOV.
- The driver version shown is 2.1.5.0, which does not align with the Intel matrix shown earlier, but is the closest available to download.
- The firmware versions can be poorly handled. 8.5 is 8.50.
- There is a more recent version available which you could consider. The problems that we previously experienced are reported fixed in the version chosen. Furthering to the latest may cause issue with VF driver compatibility.

This further reinforces the need to align NIC versions and diligently test the solution.

Remove the i40enu driver and install the i40en driver, following the vendor's instructions:

```
[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-e4434b314530/Tiger] unzip Intel-
i40en_2.1.5.0-1OEM.700.1.0.15843807_18631754-package.zip
Archive:  Intel-i40en_2.1.5.0-1OEM.700.1.0.15843807_18631754-package.zip
  inflating: Intel-i40en_2.1.5.0-1OEM.700.1.0.15843807_18631754.zip
  inflating: doc/README.txt
  inflating: doc/release_note_i40en_2.1.5.0-1OEM.700.1.0.15843807.txt

[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-e4434b314530/Tiger] unzip Intel-
i40en_2.1.5.0-1OEM.700.1.0.15843807_18631754.zip
Archive:  Intel-i40en_2.1.5.0-1OEM.700.1.0.15843807_18631754.zip
  inflating: index.xml
  inflating: vendor-index.xml
  inflating: metadata.zip
  inflating: vib20/i40en/INT_bootbank_i40en_2.1.5.0-1OEM.700.1.0.15843807.vib

[root@esxi-tiger-14-7:~] cd

[root@esxi-tiger-14-7:~] esxcli software vib list grep i40enu
i40enu                         1.8.1.137-1vmw.702.0.20.18426014     VMW     VMwareCertified
 2022-03-03

[root@esxi-tiger-14-7:~] esxcli software vib remove -n i40enu
Removal Result
   Message: The update completed successfully, but the system needs to be rebooted for the
changes to be effective.
   Reboot Required: true
   VIBs Installed:
   VIBs Removed: VMW_bootbank_i40enu_1.8.1.137-1vmw.702.0.20.18426014
   VIBs Skipped
```

```
[root@esxi-tiger-14-7:~] esxcli software vib install -v /vmfs/volumes/ESXI-TIGER-14-
7/Tiger/vib20/i40en/INT_bootbank_i40en_2.1.5.0-1OEM.700.1.0.15843807.vib
Installation Result
   Message: The update completed successfully, but the system needs to be rebooted for the
changes to be effective.
   Reboot Required: true
   VIBs Installed: INT_bootbank_i40en_2.1.5.0-1OEM.700.1.0.15843807
   VIBs Removed:
   VIBs Skipped:

[root@esxi-tiger-14-7:~] reboot
```

Upgrade the firmware, following the vendor's instructions:

```
[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-e4434b314530/Tiger] unzip
700Series_NVMUpdatePackage_v8_50.zip
Archive:  700Series_NVMUpdatePackage_v8_50.zip
  inflating: 700Series_NVMUpdatePackage_v8_50_EFI.zip
  inflating: 700Series_NVMUpdatePackage_v8_50_ESX.tar.gz
  inflating: 700Series_NVMUpdatePackage_v8_50_FreeBSD.tar.gz
  inflating: 700Series_NVMUpdatePackage_v8_50_Linux.tar.gz
  inflating: 700Series_NVMUpdatePackage_v8_50_Windows.zip

[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-e4434b314530/Tiger] tar xzf
700Series_NVMUpdatePackage_v8_50_ESX.tar.gz
[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-e4434b314530/Tiger] cd
700Series/ESXi_x64/
[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-
e4434b314530/Tiger/700Series/ESXi_x64] ./nvmupdaten64e
<output omitted for brevity>

[root@esxi-tiger-14-7:/vmfs/volumes/6216bfd4-deba7d2c-3ae7-
e4434b314530/Tiger/700Series/ESXi_x64] reboot
```

You can check the results via the previous `esxcli network nic get` command or via `esxcli system module get`:

```
[root@esxi-tiger-14-7:~] esxcli system module get -m i40en
   Module: i40en
   Module File: /usr/lib/vmware/vmkmod/i40en
   License: ThirdParty:Intel
   Version: 2.1.5.0-1OEM.700.1.0.15843807
   Build Type: release
   Provided Namespaces:
   Required Namespaces: com.vmware.vmkapi@v2_6_0_0
   Containing VIB: i40en
   VIB Acceptance Level: certified
```

NIC queues (ring buffer size)

Maximize the receive queue/buffer on the NIC to optimize throughput. This is not expressly needed, but maximizing the transmit queue is also performed:

```
[root@esxi-tiger-14-7:~] esxcli network nic ring current get -n vmnic4
   RX: 1024
   RX Mini: 0
```

```
   RX Jumbo: 0
   TX: 1024

[root@esxi-tiger-14-7:~] esxcli network nic ring preset get -n vmnic4
   Max RX: 4096
   Max RX Mini: 0
   Max RX Jumbo: 0
   Max TX: 4096

[root@esxi-tiger-14-7:~] esxcli network nic ring current set -n vmnic4 -r 4096 -t 4096

[root@esxi-tiger-14-7:~] esxcli network nic ring current get -n vmnic4
   RX: 4096
   RX Mini: 0
   RX Jumbo: 0
   TX: 4096
```

In the case of the documented setup, this is a setting is persisted across reboots. This has not always been the case. It is worth checking and making provisions accordingly.

Network virtual functions

You must create network virtual functions (VFs) to use with the FortiGate-VM. As confirmed in NIC versions on page 59, the particular combination of NIC and driver/module supports SR-IOV. You can further check this by ensuring that the module has settings for VFs.

```
[root@esxi-tiger-14-7:~] esxcli system module parameters list -m i40en
Name           Type          Value  Description
-------------  ------------  -----  -----------
DRSS           array of int         Enable/disable the DefQueue RSS(default = 0 )
EEE            array of int         Energy Efficient Ethernet feature (EEE): 0 = disable, 1
= enable, (default = 1)
LLDP           array of int         Link Layer Discovery Protocol (LLDP) agent: 0 = disable,
1 = enable, (default = 1)
RSS            array of int         Enable/disable the NetQueue RSS( default = 1 )
RxITR          int                  Default RX interrupt interval (0..0xFFF), in
microseconds (default = 50)
TxITR          int                  Default TX interrupt interval (0..0xFFF), in
microseconds, (default = 100)
VMDQ           array of int         Number of Virtual Machine Device Queues: 0/1 = disable,
2-16 enable (default =8)
max_vfs        array of int         Maximum number of VFs to be enabled (0..128)
trust_all_vfs  array of int         Always set all VFs to trusted mode 0 = disable
(default), other = enable
```

Leaving the `max_vfs` as-is is fine. This just limits the number that you can define. This leaves the number of VFs configurable in the hands of the mixture of ESXi and NIC and driver. In the documented example, only two VFs are needed per physical interface. Configuring eight allows a greater degree of flexibility without the need to reboot the host.

The `trust_all_vfs` is an important setting. It ensures that spoof check is disabled and that the VF is trusted.

```
[root@esxi-tiger-14-7:~] esxcli system module parameters set -m i40en -p "max_
vfs=0,0,8,8,8,8 trust_all_vfs=0,0,1,1,1,1"
[root@esxi-tiger-14-7:~] esxcli system module parameters list -m i40en
Name           Type          Value        Description
-------------  ------------  -----------  -----------
```

```
DRSS            array of int               Enable/disable the DefQueue RSS(default = 0 )
EEE             array of int               Energy Efficient Ethernet feature (EEE): 0 =
disable, 1 = enable, (default = 1)
LLDP            array of int               Link Layer Discovery Protocol (LLDP) agent: 0 =
disable, 1 = enable, (default = 1)
RSS             array of int               Enable/disable the NetQueue RSS( default = 1 )
RxITR           int                        Default RX interrupt interval (0..0xFFF), in
microseconds (default = 50)
TxITR           int                        Default TX interrupt interval (0..0xFFF), in
microseconds, (default = 100)
VMDQ            array of int               Number of Virtual Machine Device Queues: 0/1 =
disable, 2-16 enable (default =8)
max_vfs         array of int  0,0,8,8,8,8  Maximum number of VFs to be enabled (0..128)
trust_all_vfs   array of int  0,0,1,1,1,1  Always set all VFs to trusted mode 0 = disable
(default), other = enable
[root@esxi-tiger-14-7:~] reboot
```

Why “0,0,8,8,8,8” and “0,0,1,1,1,1”? This is an array of values, which references each NIC using the i40en driver. If you compare this to the earlier `esxcli network nic list` output, you see that six NICs are using the i40en driver: vmnic0 and vmnic1 in addition to the four that are of larger interest. vmnic0 and vmnic1 are Dell OEM devices on the mainboard and are therefore not recommended for this use case. So the array references all six NICs in order. You must diligently check any changes made to the hardware after this setup.

Disabling spoof check allows the VM to define the MAC addresses it associates to interfaces rather than those that the host set. This is important when considering the deployment of LAGs and for FortiGate Clustering Protocol vMAC operation.

Setting the VF to trusted is important to ensure that the VF tracks and follows the status of the PF. Allowing the VM to detect interface down accordingly. This setting is also mandatory for LAG.

To make further checks around this area, installing the vendor toolset is highly recommended, if available. For this example, the Intel plugin is installed:

```
[root@esxi-tiger-14-7:/vmfs/volumes/62248617-84a2aac8-cef7-e4434b314530/Tiger] unzip Intel-
intnetcli_1.6.5.0__esx7.0.zip
Archive:  Intel-intnetcli_1.6.5.0__esx7.0.zip
  inflating: Intel-intnetcli_intnetcli.1.6.5.0-700.15843807_18728558.zip
  inflating: doc/README.txt
[root@esxi-tiger-14-7:/vmfs/volumes/62248617-84a2aac8-cef7-e4434b314530/Tiger] unzip Intel-
intnetcli_intnetcli.1.6.5.0-700.15843807_18728558.zip
Archive:  Intel-intnetcli_intnetcli.1.6.5.0-700.15843807_18728558.zip
  inflating: index.xml
  inflating: vendor-index.xml
  inflating: metadata.zip
  inflating: vib20/int-esx-intnetcli/INT_bootbank_int-esx-intnetcli_700.1.6.5.0-15843807.vib
[root@esxi-tiger-14-7:/vmfs/volumes/62248617-84a2aac8-cef7-e4434b314530/Tiger] cd
[root@esxi-tiger-14-7:~] esxcli software vib install -v /vmfs/volumes/ESXI-TIGER-14-
7/Tiger/vib20/int-esx-intnetcli/INT_bootbank_int-esx-intnetcli_700.1.6.5.0-15843807.vib
Installation Result
   Message: The update completed successfully, but the system needs to be rebooted for the
changes to be effective.
   Reboot Required: true
```

```
   VIBs Installed: INT_bootbank_int-esx-intnetcli_700.1.6.5.0-15843807
   VIBs Removed:
   VIBs Skipped:
[root@esxi-tiger-14-7:~] reboot

[root@esxi-tiger-14-7:~] esxcli intnet sriovnic vf get -n vmnic4


VF ID      Trusted     Spoof Check
-----          -------     -----------
0              true        false
1              true        false
2              true        false
3              true        false
4              true        false
5              true        false
6              true        false
7              true        false
```

Adding the configuration in this way automatically created the eight VFs. You can also create the VFs in vCentre, in case you executed the build process differently.

```
[root@esxi-tiger-14-7:~] esxcli network sriovnic list
Name    PCI Device    Driver  Link  Speed  Duplex  MAC Address         MTU  Description
------  ------------  ------  ----  -----  ------  -----------------  ----  -----------
vmnic4  0000:3b:00.0  i40en   Up    25000  Full    3c:fd:fe:c3:8a:c8  1500  Intel(R)
Ethernet Controller XXV710 for 25GbE SFP28
vmnic5  0000:3b:00.1  i40en   Up    25000  Full    3c:fd:fe:c3:8a:c9  1500  Intel(R)
Ethernet Controller XXV710 for 25GbE SFP28
vmnic6  0000:5e:00.0  i40en   Up    25000  Full    3c:fd:fe:c3:94:1c  1500  Intel(R)
Ethernet Controller XXV710 for 25GbE SFP28
vmnic7  0000:5e:00.1  i40en   Up    25000  Full    3c:fd:fe:c3:94:1d  1500  Intel(R)
Ethernet Controller XXV710 for 25GbE SFP28
[root@esxi-tiger-14-7:~] esxcli network sriovnic vf list -n vmnic4
VF ID  Active  PCI Address     Owner World ID
-----  ------  --------------  --------------
    0   false  00000:059:02.0   -
    1   false  00000:059:02.1   -
    2   false  00000:059:02.2   -
    3   false  00000:059:02.3   -
    4   false  00000:059:02.4   -
    5   false  00000:059:02.5   -
    6   false  00000:059:02.6   -
    7   false  00000:059:02.7   -
```

VM creation

NUMA identification

For the best performance, avoid using resources in different NUMA nodes for a single VM.

The following shows CPU to NUMA mapping:

```
[root@esxi-tiger-14-7:~] for X in 0 1; do echo -n "NUMA${X}: "; cpuList=`vsish -e ls
/hardware/numa/${X}/pcpus`; echo $cpuList; done
NUMA0: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
NUMA1: 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
```

The following shows NIC to NUMA mapping:

```
[root@esxi-tiger-14-7:~] vsish -e cat /net/pNics/vmnic4/properties | grep NUMA
   Device NUMA Node:0
```

The memory that ESXi uses is automatically optimized for NUMA. Do the following to see the memory installed:

```
[root@esxi-tiger-14-7:~] esxcli hardware memory get
   Physical Memory: 204678979584 Bytes
   Reliable Memory: 0 Bytes
   NUMA Node Count: 2
```

With a sane hardware build, the assumption is that the memory is split equally between NUMAs.

Hugepages

See Backing Guest vRAM with 1GB Pages for the ESXi stance on hugepages.

The following summarizes points of interest:

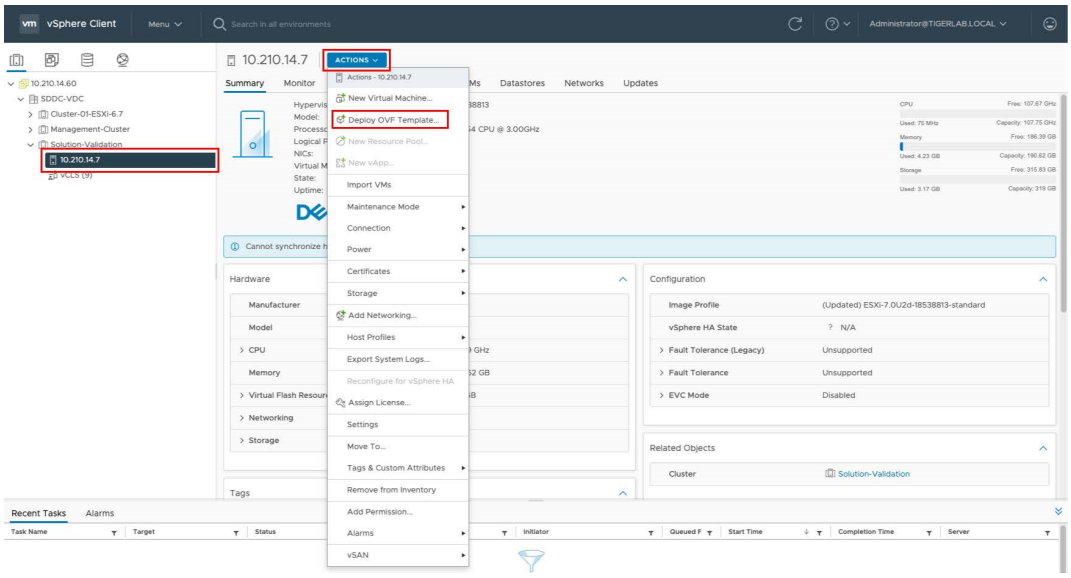
A VM with 1 GB pages enabled must have full memory reservation. Otherwise, the VM cannot power on. All of the vRAM for VMs with 1 GB pages enabled is preallocated on power-on. Since these VMs have full memory reservation, memory reclamation does not affect them, and their memory consumption stays at the maximum level for the VM's entire lifetime.

1 GB page vRAM backing is opportunistic and 1 GB pages are allocated on a best effort basis. This includes cases where host CPUs do not have 1 GB capabilities. To maximize the chances of having guest vRAM backed with 1GB pages, starting VMs requiring 1 GB pages on a freshly booted host is recommended, because the host RAM is fragmented over time.

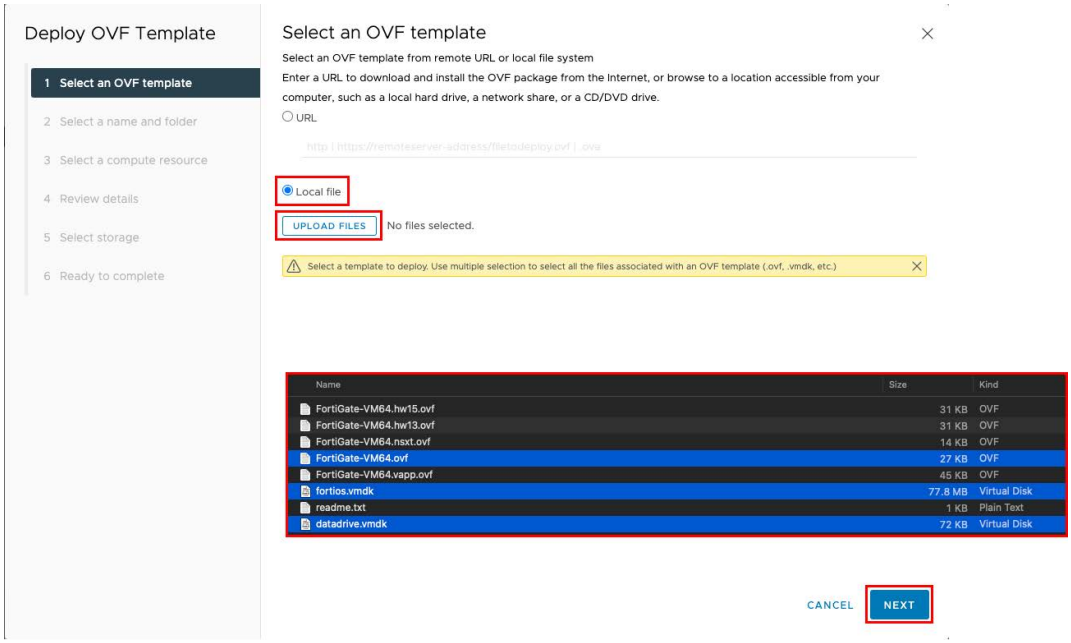
FortiGate-VM benefits from 1 G hugepages, so it is assumed that the FortiGate-VM is the only VM running on a NUMA, or that the administrator can orchestrate the environment such that FortiGate-VM is granted 1 G hugepages.

Deploying the OVF template

In vCenter, select the hypervisor host, and from *Actions*, select *Deploy OVF Template*.



Install from *Local file* and select the appropriate files from the OVF file from Fortinet. The readme.txt file contains information as to which OVF file to use.



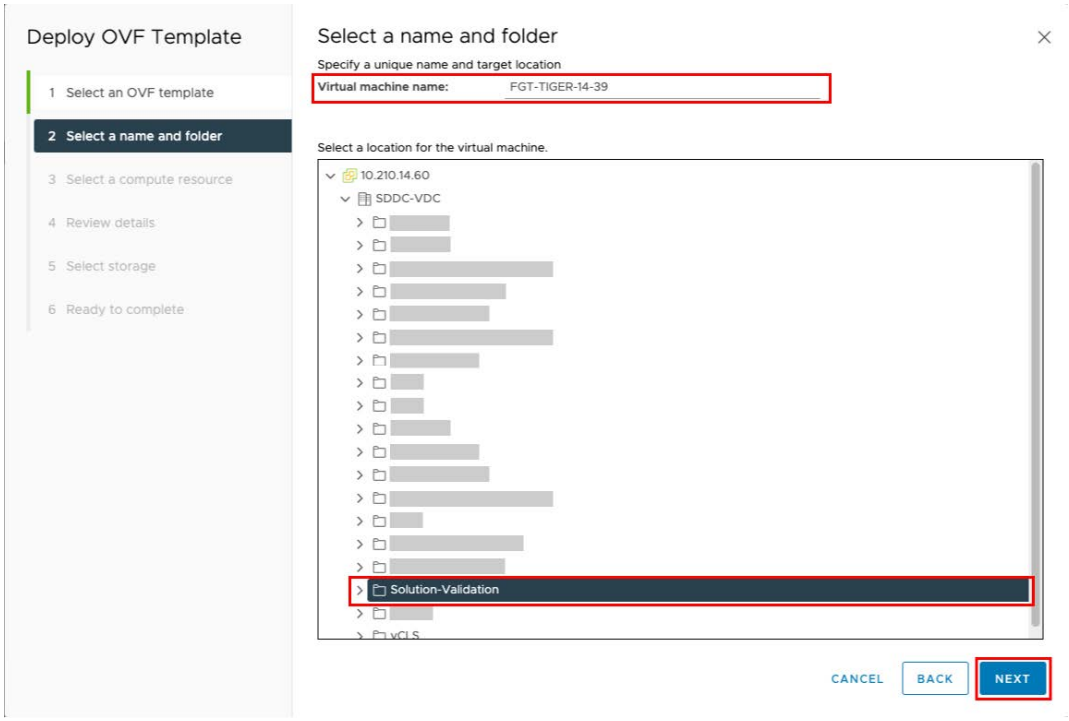
For versions earlier than FortiOS 7.0, there is no readme.txt file and the administrator must consider hardware versions more. If it is not obviously indicated in the filename, the OVF file is an XML file, which you can inspect to determine the hardware version. For example, the FortiGate-VM64.ovf file for 6.4.8 has the following content:

```
<VirtualHardwareSection>
   <Info>Virtual hardware requirements</Info>
   <System>
      <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
      <vssd:InstanceID>0</vssd:InstanceID>
      <vssd:VirtualSystemIdentifier>FGT_VM</vssd:VirtualSystemIdentifier>
      <vssd:VirtualSystemType>vmx-07</vssd:VirtualSystemType>
```

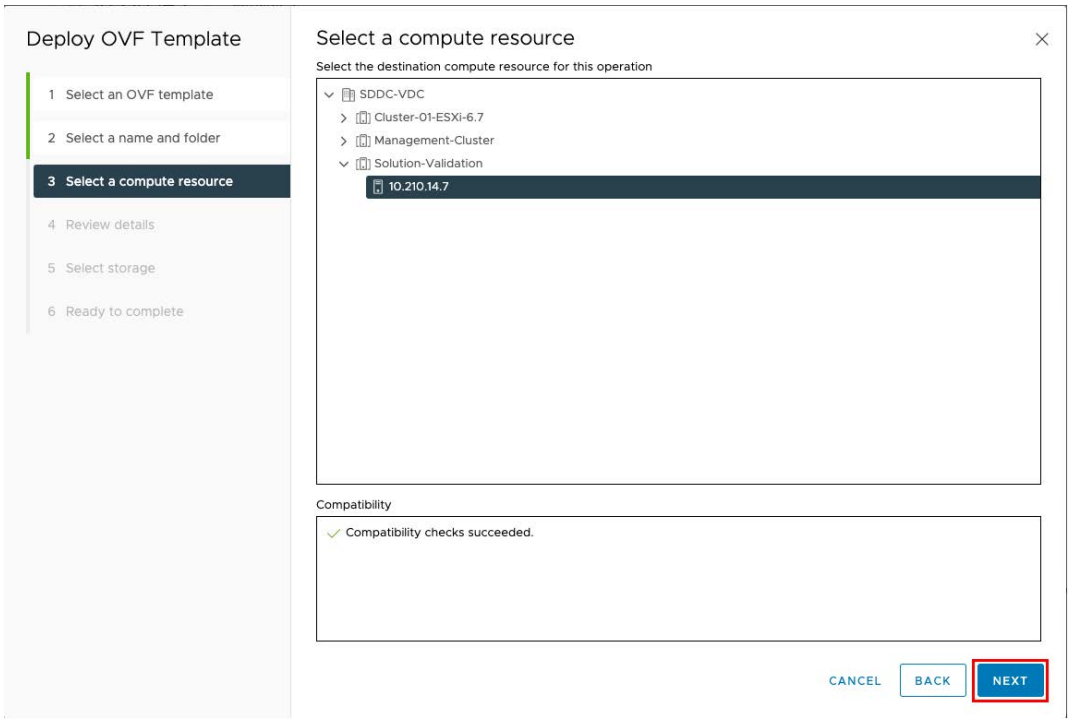
```
</System>
```

In this case, it is hardware version 7. The recommendation is to pick the latest hardware version that the ESXi deployment supports, and allow ESXi to upgrade the hardware if/when prompted.

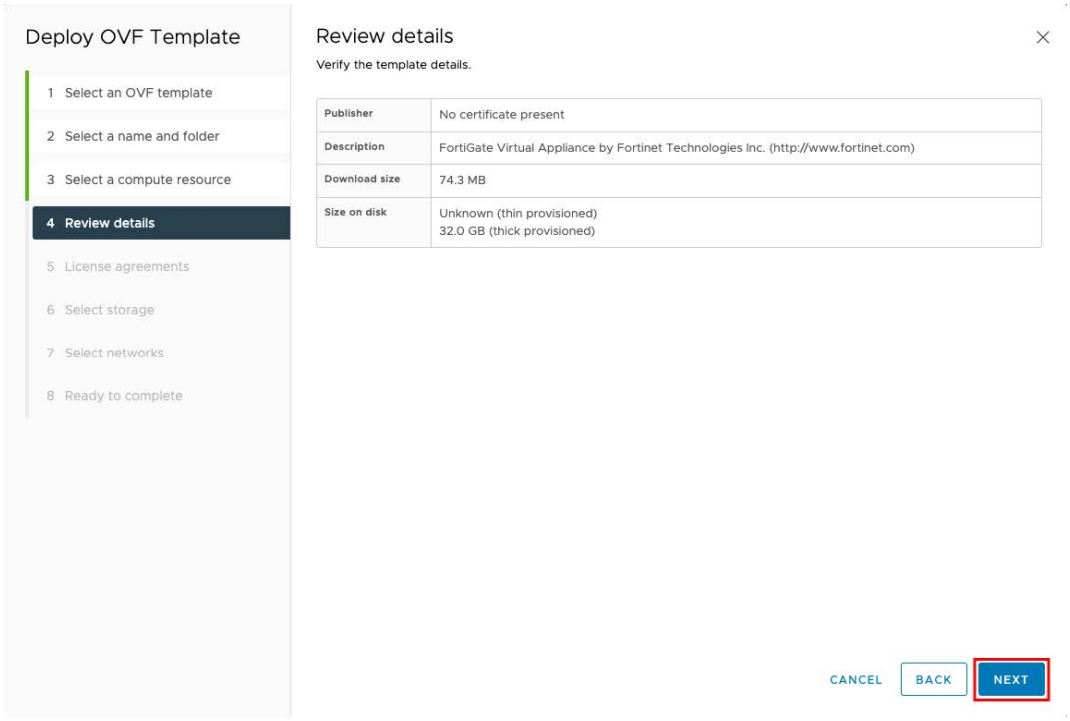
Name the VM and select an appropriate folder to associate it to.

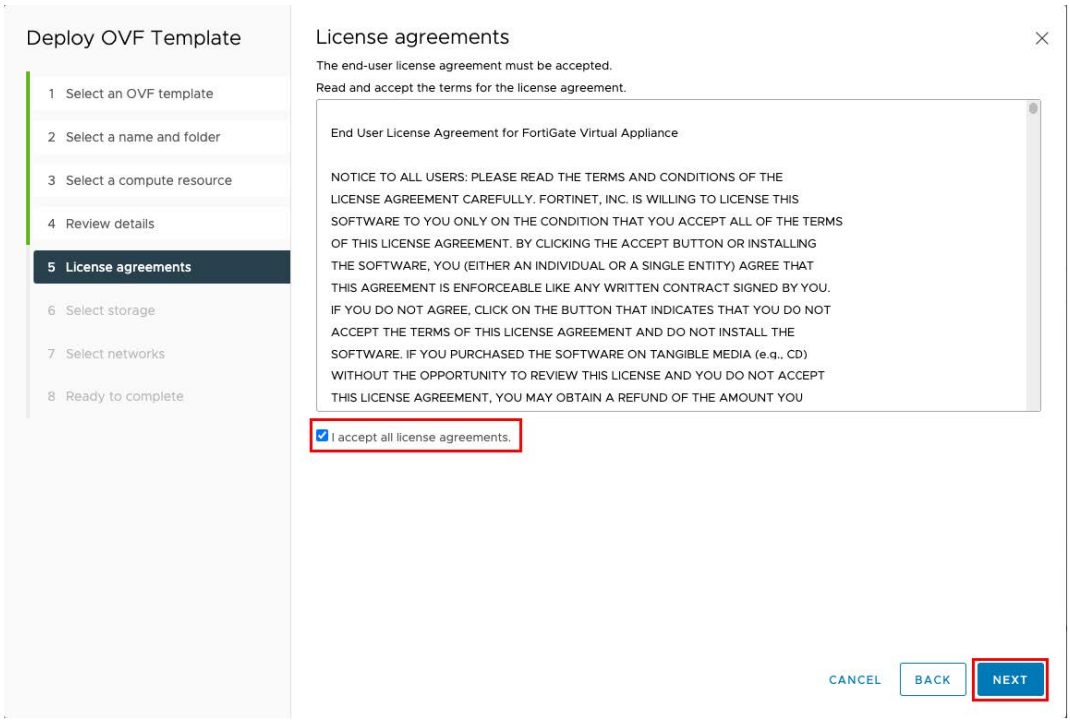


Confirm the chosen compute resource:

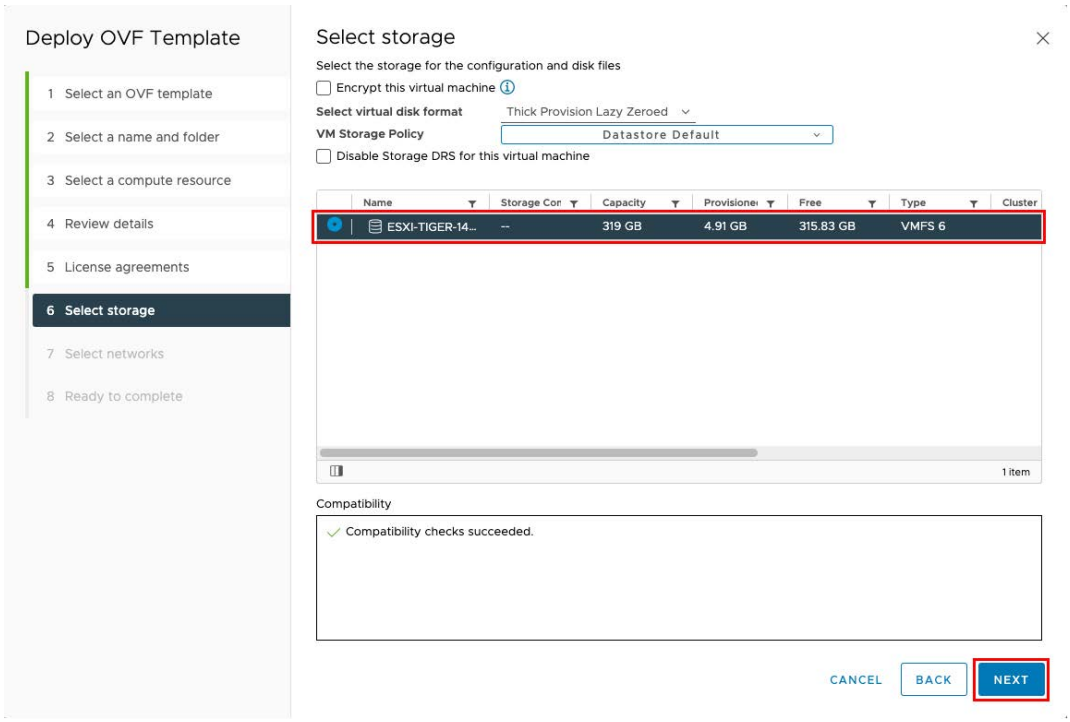


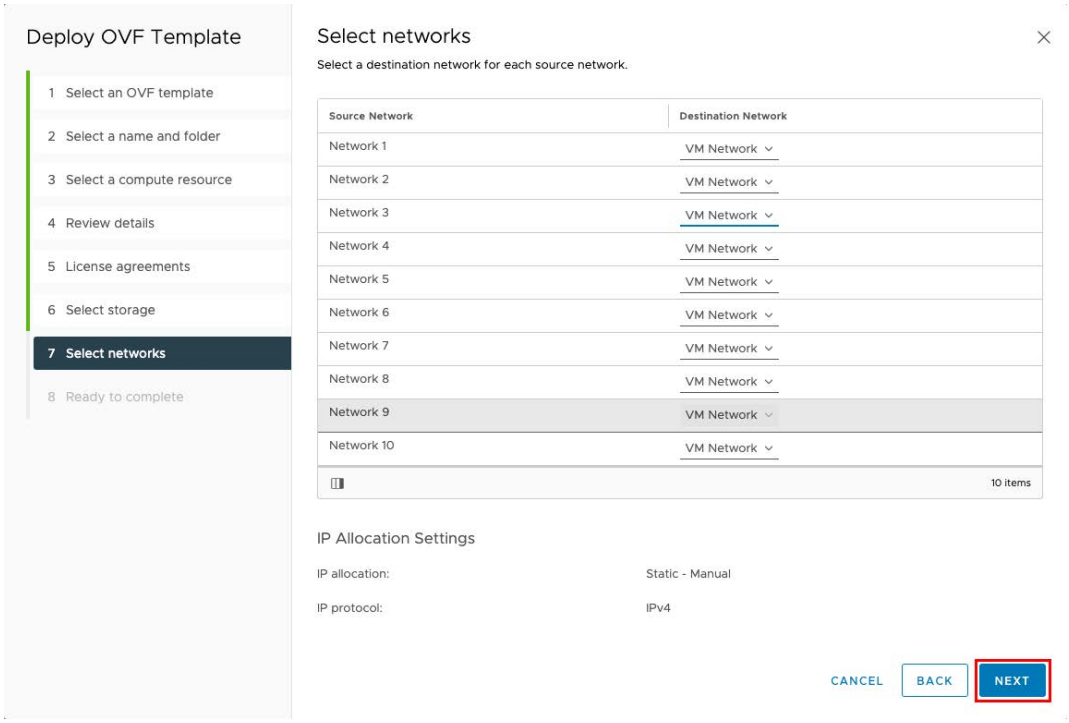
Review details and accept license as appropriate.

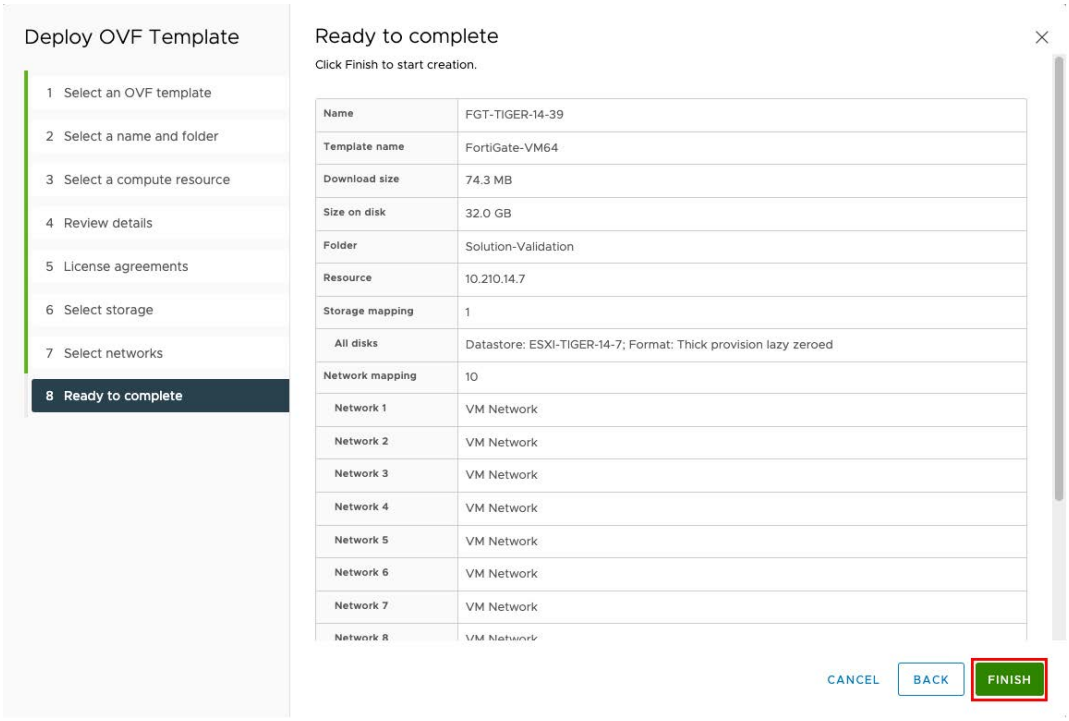




Select storage, and click through until *Finish* and wait for the VM instance to be built.

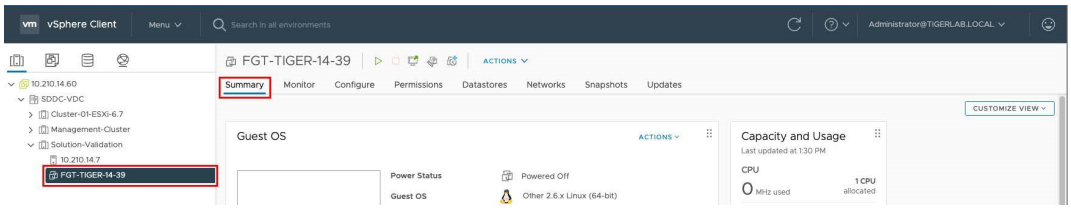

Deploy OVF Template
1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 License agreements
6 Select storage
7 Select networks
8 Ready to complete
Select storage
Select the storage for the configuration and disk files
Encrypt this virtual machine
Select virtual disk format
Thick Provision Lazy Zeroed
VM Storage Policy
Datastore Default
Disable Storage DRS for this virtual machine
Name
Storage Cor
Capacity
Provisione
Free
Type
Cluster
ESXI-TIGER-14...
--
319 GB
4.91 GB
315.83 GB
VMFS 6
1 item
Compatibility
Compatibility checks succeeded.
CANCEL
BACK
NEXT

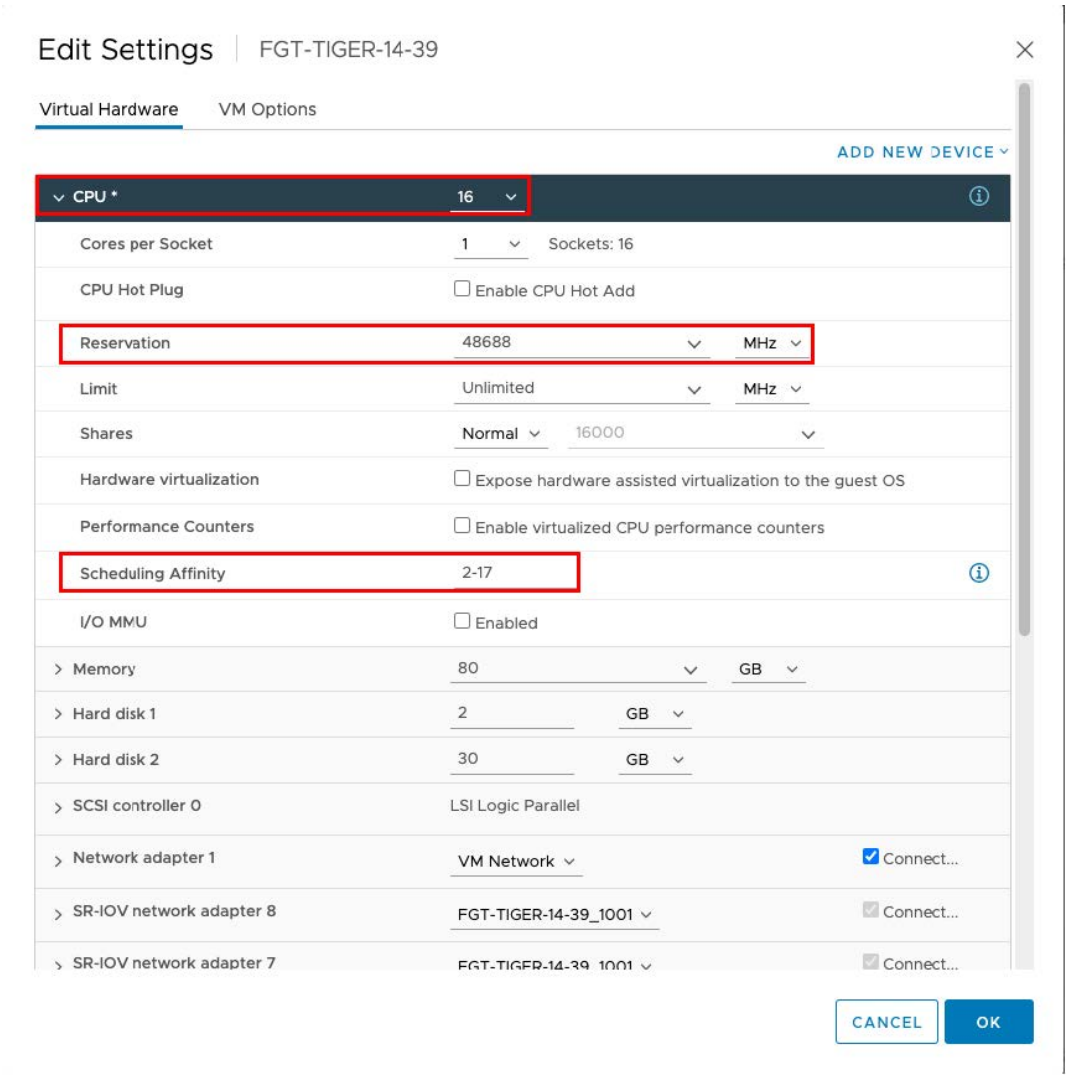

Deploy OVF Template
1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 License agreements
6 Select storage
7 Select networks
8 Ready to complete
Select networks
Select a destination network for each source network.
Source Network
Destination Network
Network 1
Network 2
Network 3
Network 4
Network 5
Network 6
Network 7
Network 8
Network 9
Network 10
VM Network
10 items
IP Allocation Settings
IP allocation:
Static - Manual
IP protocol:
IPv4
CANCEL
BACK
NEXT



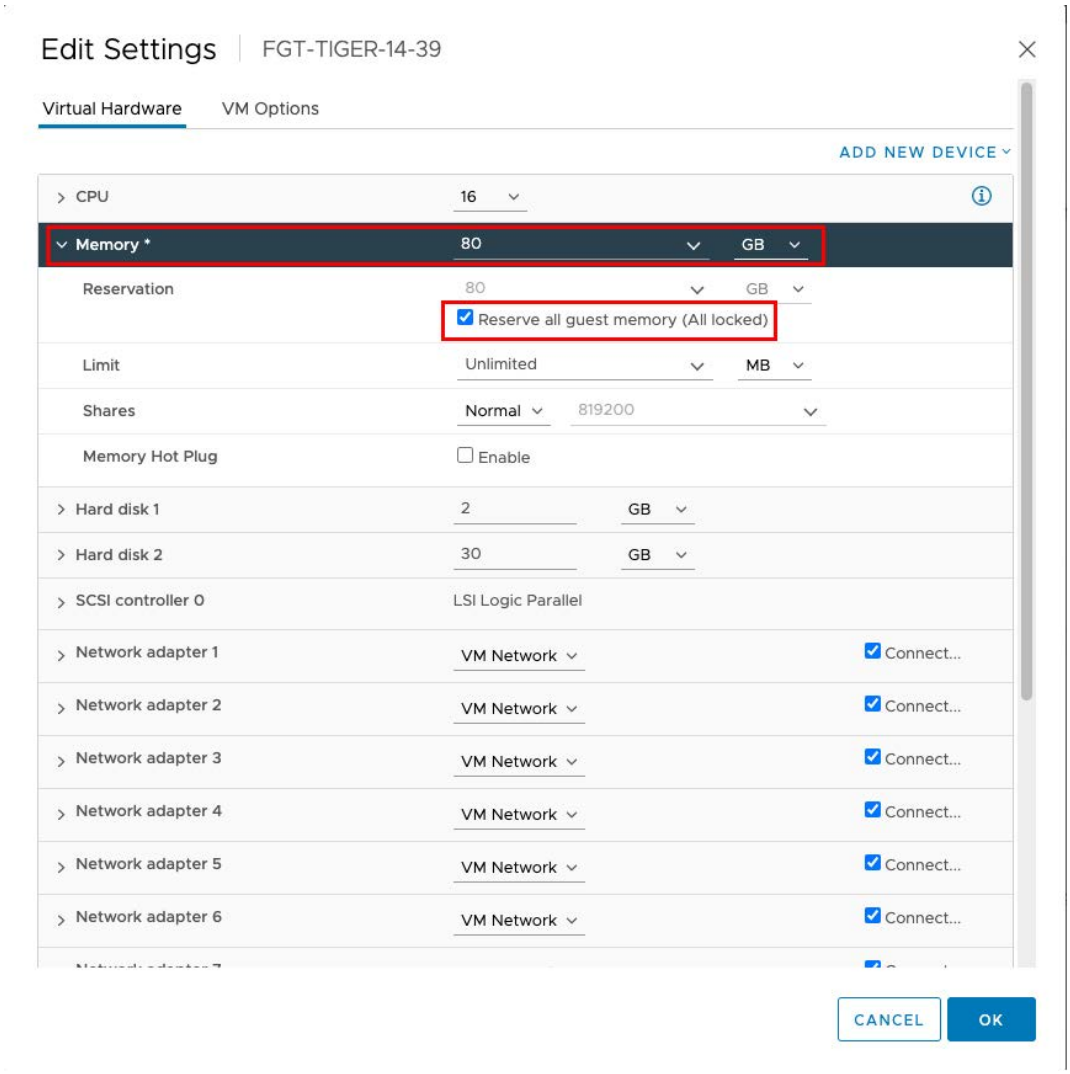
Tuning the FortiGate-VM instance

In vCenter, select the FGVM guest, and from Summary choose *EDIT* from the VM Hardware section.

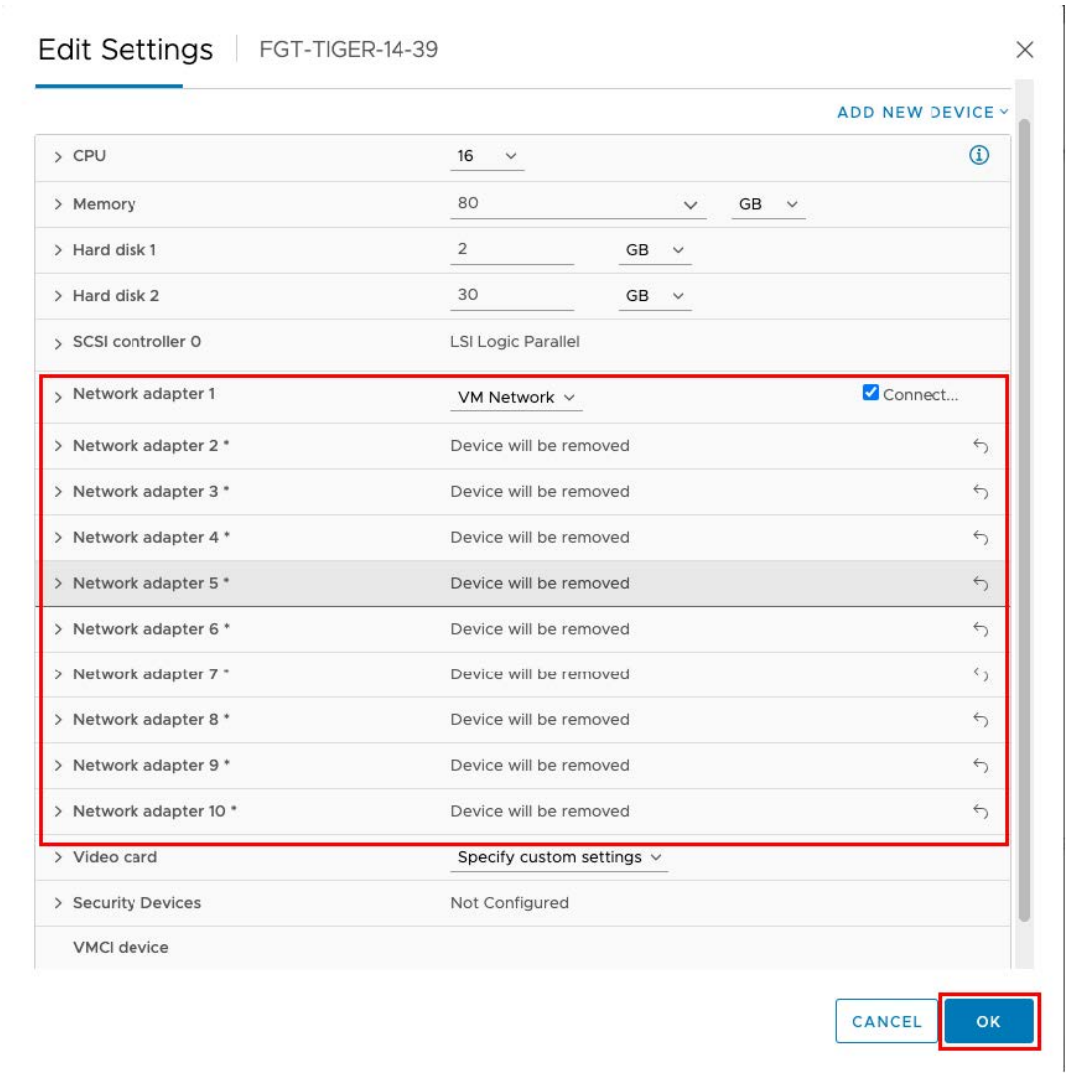




Increase the number of CPUs as needed for the solution. The frequency reservation may need experimentation to maximize, but is based on the frequency if the cores and the number you have allocates. The scheduling affinity allows the effective pinning of CPUs to the FortiGate-VM instance. The CPUs chosen are based on matching the NUMA to the NICs installed, and also the lower numbers one have been left for the usage by the hypervisor itself.



You should allocate and lock sufficient memory to the guest. This is mandatory for SR-IOV.



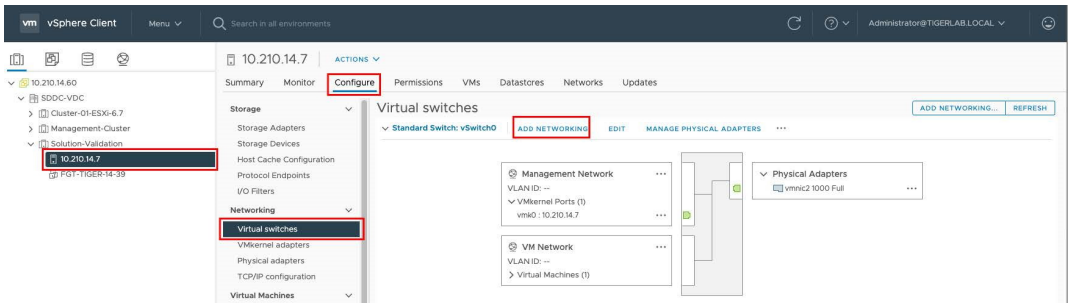
Remove all but the first network adapter. You will create traffic interfaces later. The remaining interface is used for management.

Click *OK*.

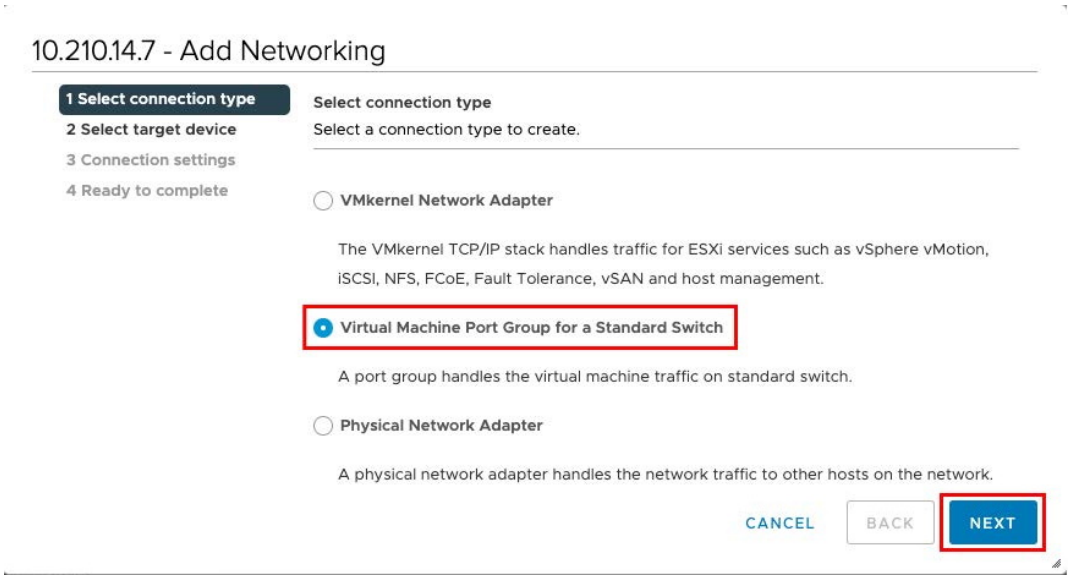
Configuring Network

The purpose of these steps is to add VLANs to associate SR-IOV VF with. This allows multiple VFs to be run on the PF but using VLANs to limit prune the traffic to each VF.

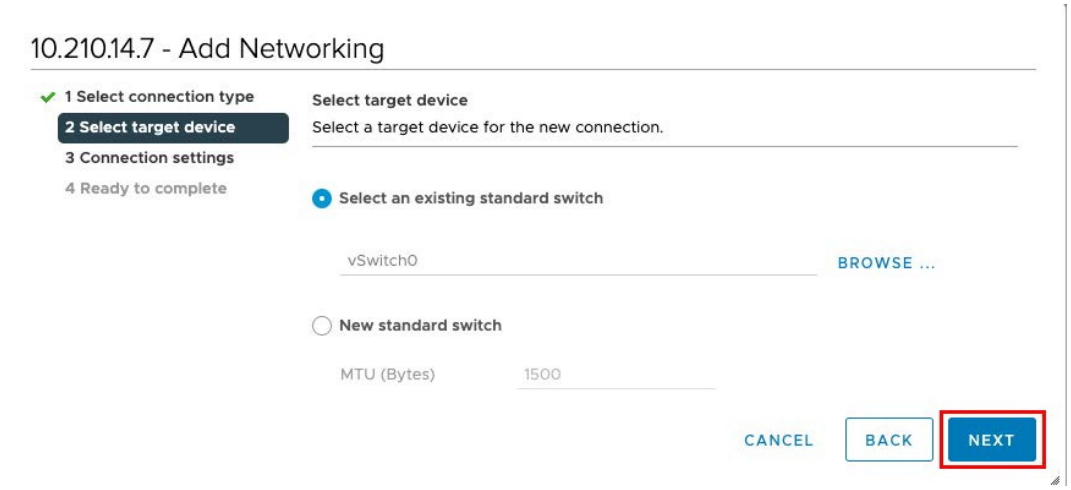
In vCenter, select the hypervisor host, select *Configure*, then *Virtual Switches* and *ADD NETWORKING*.



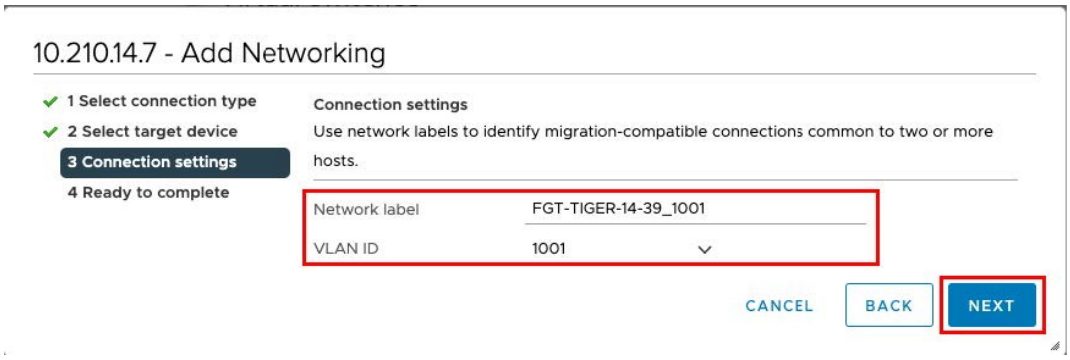
Select *Virtual Machine Port Group for a Standard Switch*:

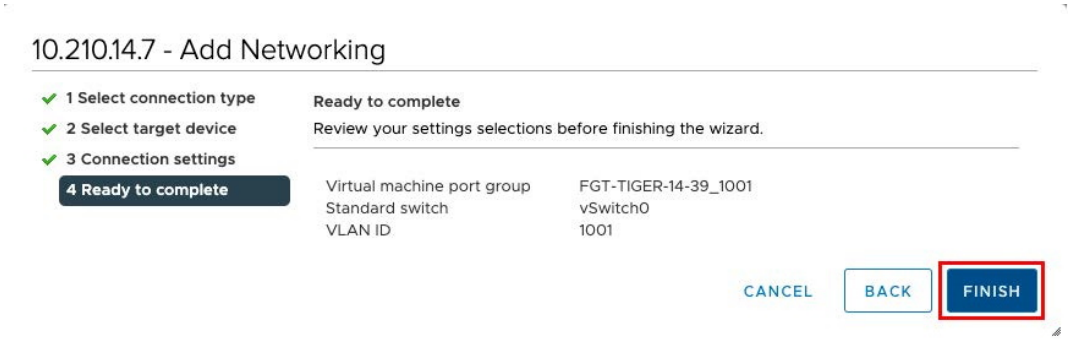


The switch is not really very important for these settings. You can configure these as you want as only VLANs are important for SR-IOV.

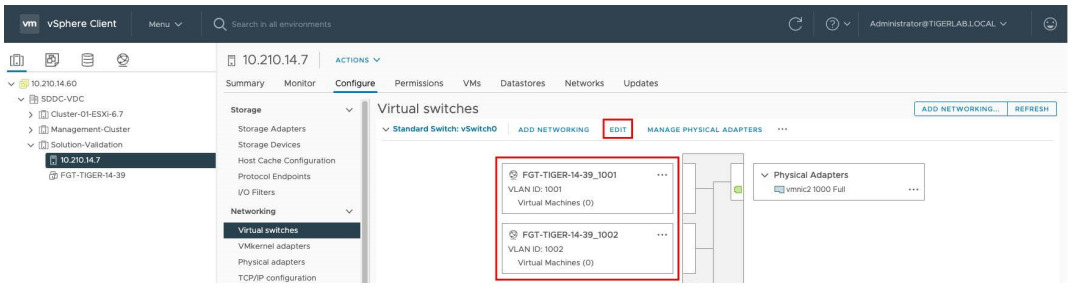


Give the port group a name and a VLAN, and click *Finish*.





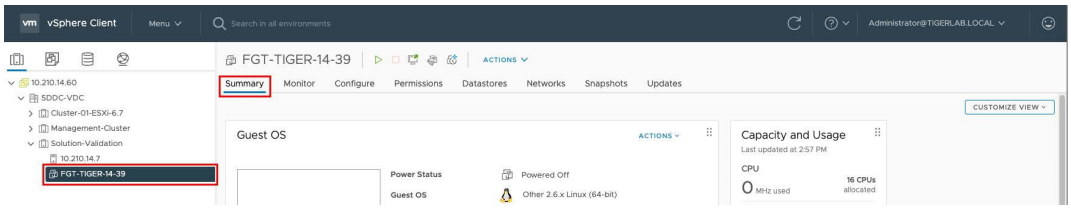
Repeat the process for as many different VLANs (effectively how the SR-IOVs are distinguished on the PF) as required. In this example, two VLANs are needed.



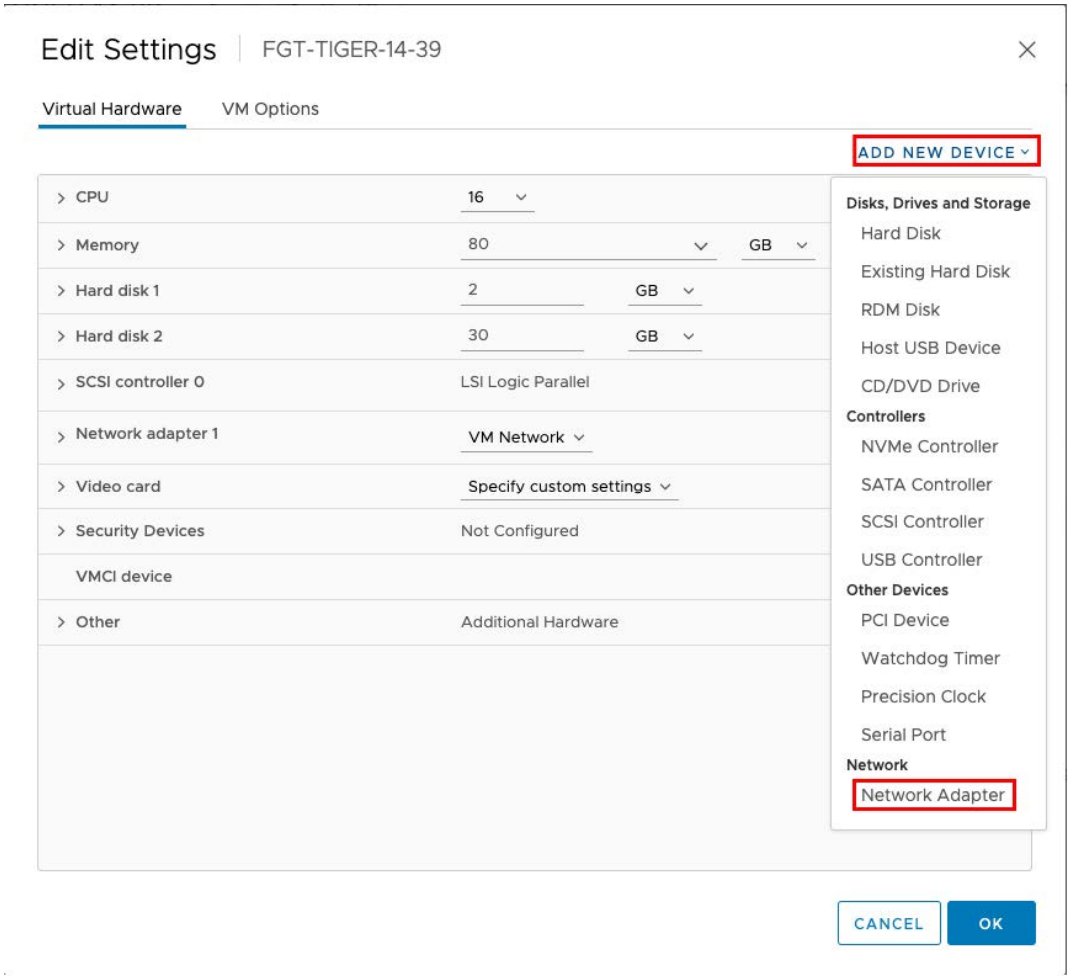
Click *EDIT* and turn off the vSwitch security settings. These setting are meant for end systems rather than network equipment such as routers and firewalls.

Configuring traffic interfaces

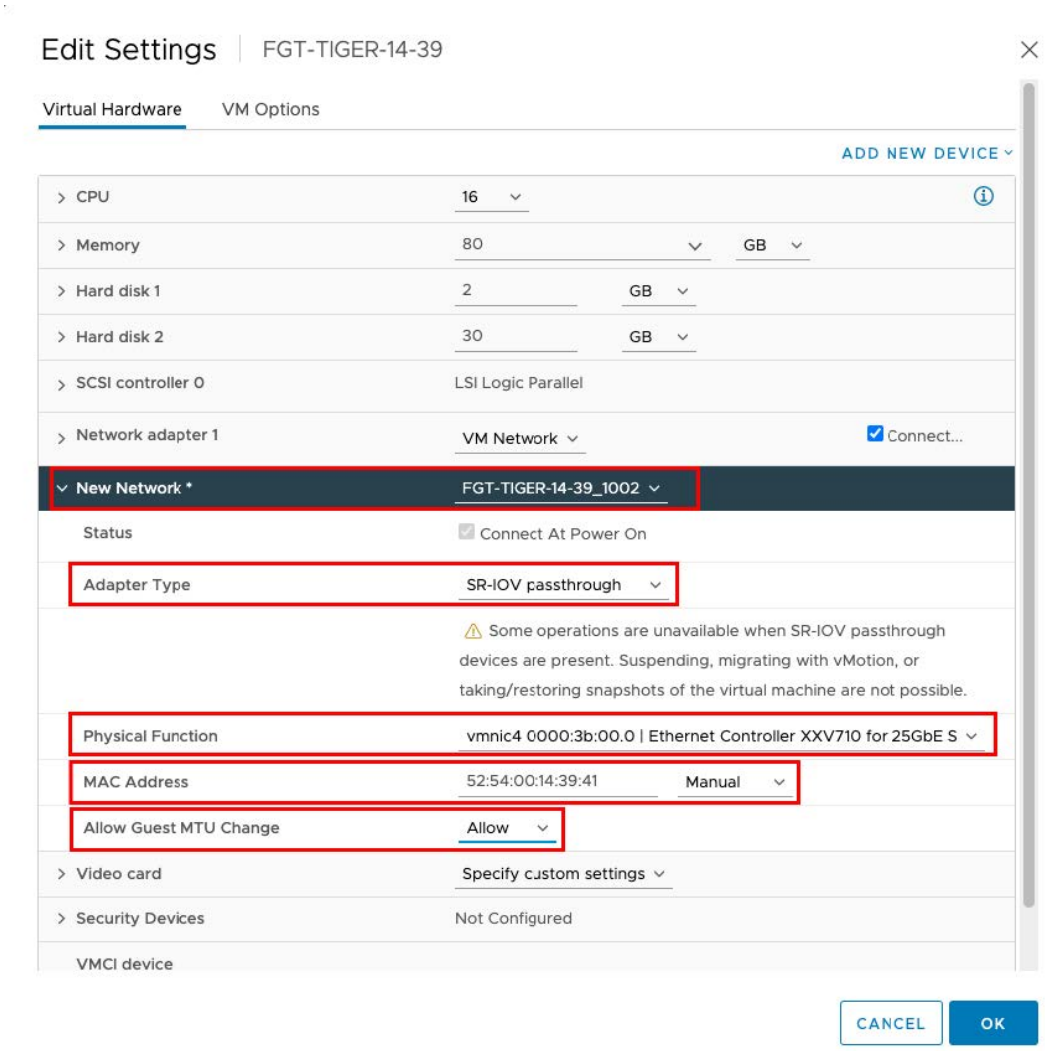
In vCenter, select the FortiGate-VM guest, and from Summary, select *EDIT* from the VM Hardware section.

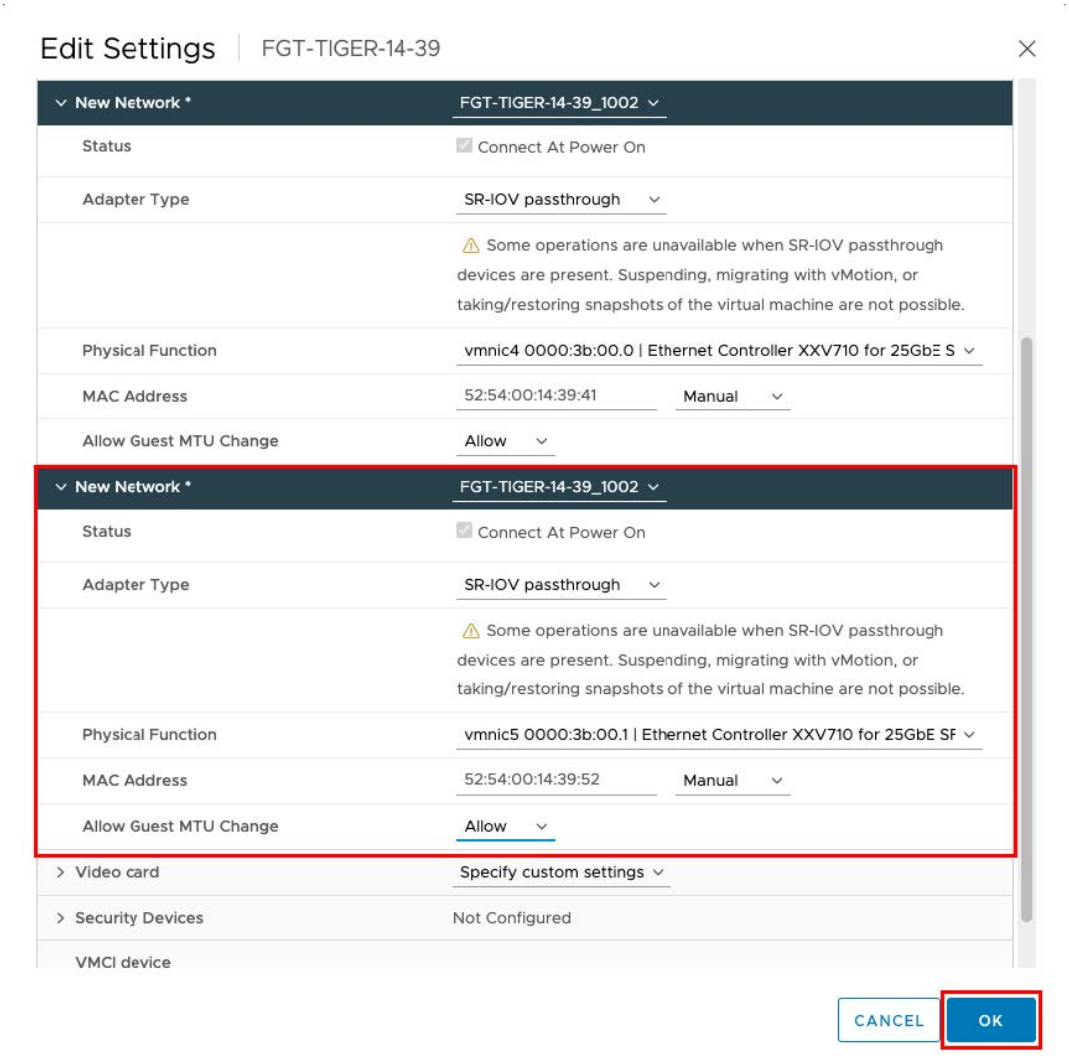


Use *ADD NEW DEVICE* to add as many network adapters as needed. In this example, there are four physical interfaces, running two VFs on each.

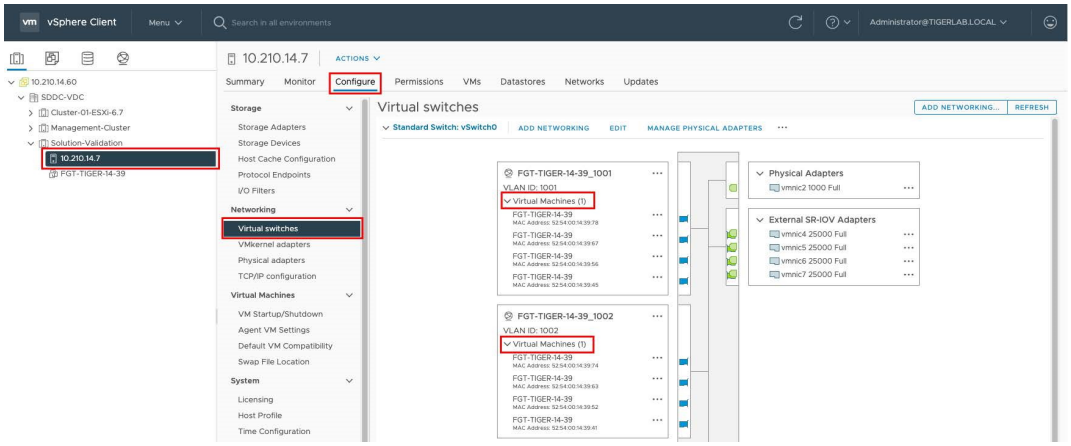


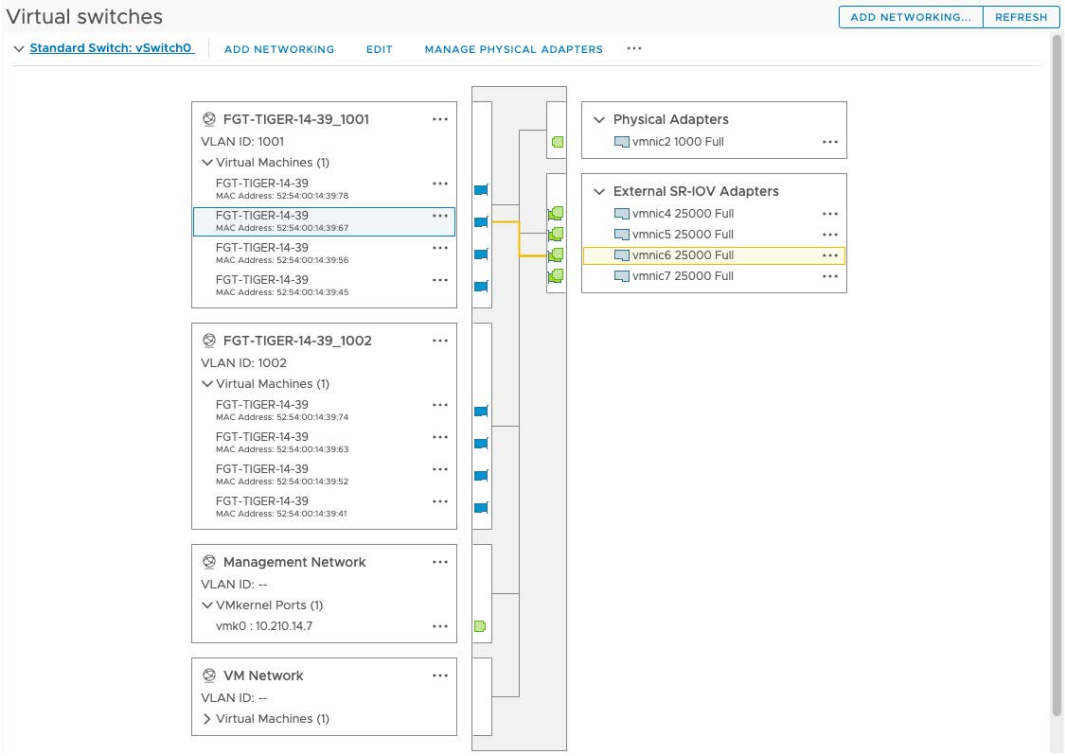
For each network adapter added, select the network (in reality the VLAN), that it is SR-IOV, the PF, and allow the guest to make MTU changes. You can leave the MAC address as Auto, but a manual MAC makes identifying the interface in the FortiGate-VM guest later easier. In the example, the last part of the MAC address represents the NIC number followed by the network adapter instance.


Edit Settings
FGT-TIGER-14-39
Virtual Hardware
VM Options
ADD NEW DEVICE
CPU
16
Memory
80
GB
Hard disk 1
2
GB
Hard disk 2
30
GB
SCSI controller 0
LSI Logic Parallel
Network adapter 1
VM Network
Connect...
New Network *
FGT-TIGER-14-39_1002
Status
Connect At Power On
Adapter Type
SR-IOV passthrough
Some operations are unavailable when SR-IOV passthrough devices are present. Suspending, migrating with vMotion, or taking/restoring snapshots of the virtual machine are not possible.
Physical Function
vmnic4 0000:3b:00.0 | Ethernet Controller XXV710 for 25GbE S
MAC Address
52:54:00:14:39:41
Manual
Allow Guest MTU Change
Allow
Video card
Specify custom settings
Security Devices
Not Configured
VMCI device
CANCEL
OK

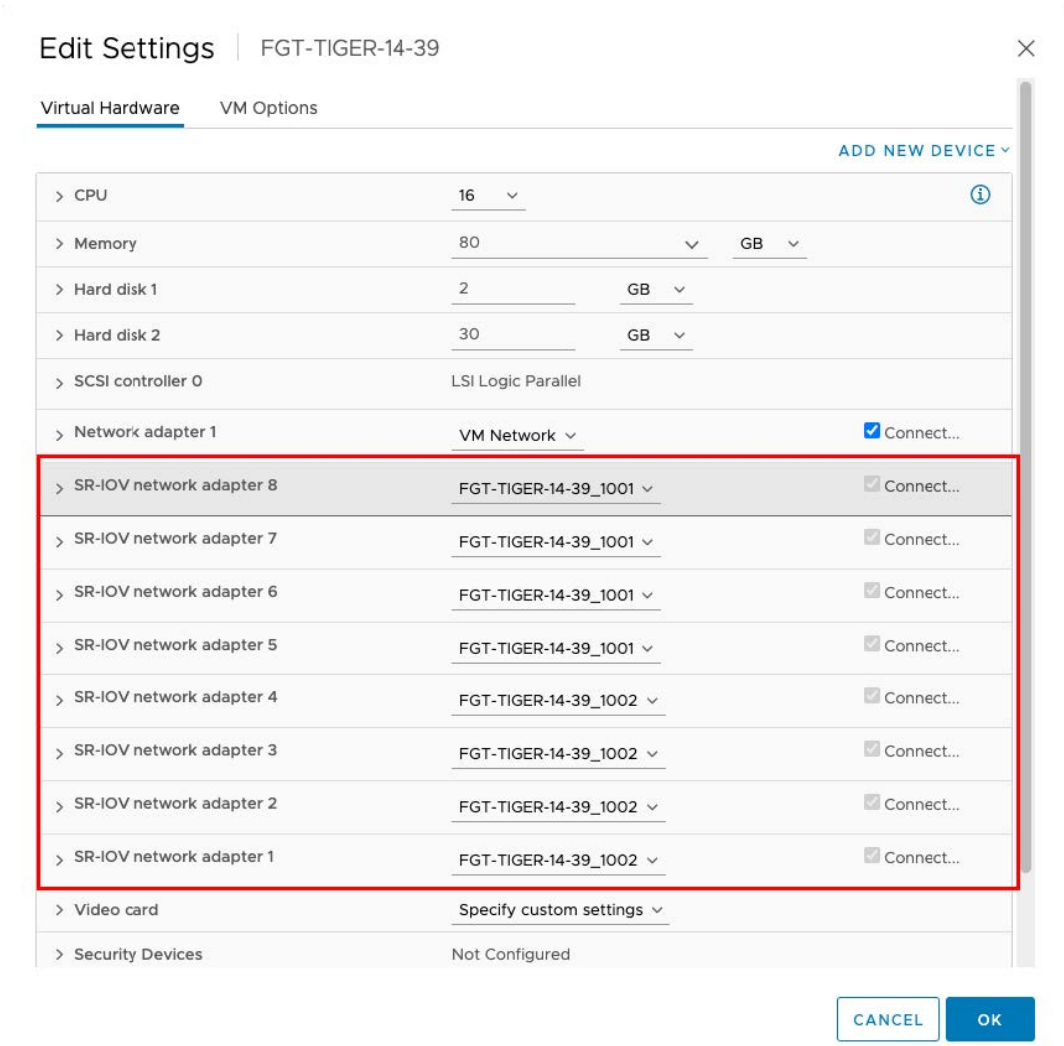


Once this is done, the hypervisor vSwitch configuration shows the relationship between the VLAN, the VF, and the PF.

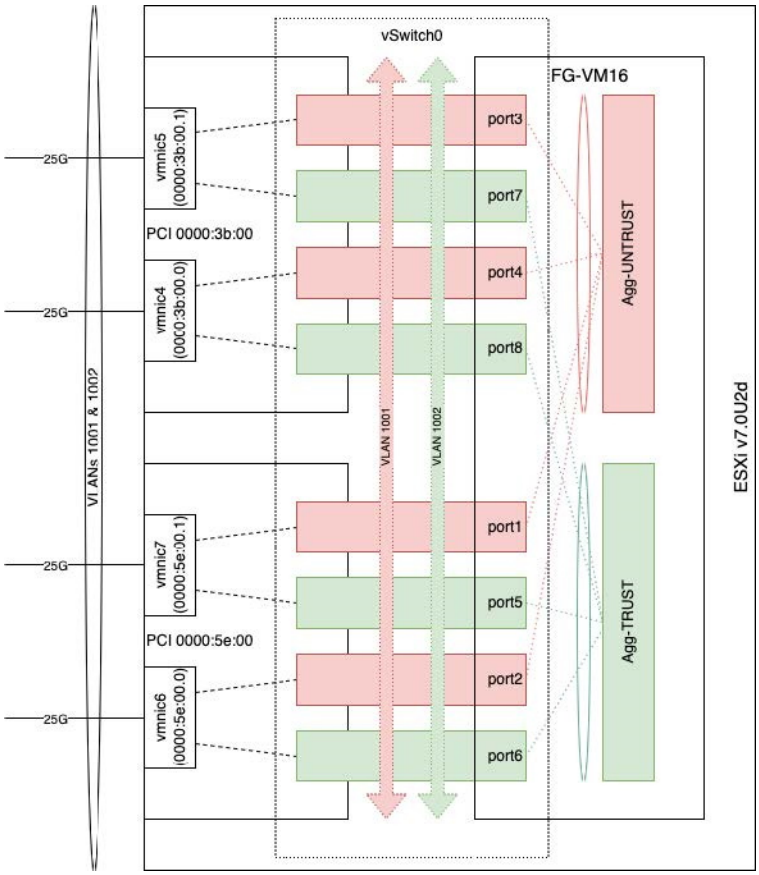




The following screenshot shows why using a manual MAC address is useful. In the screenshot, the order is not as expected.

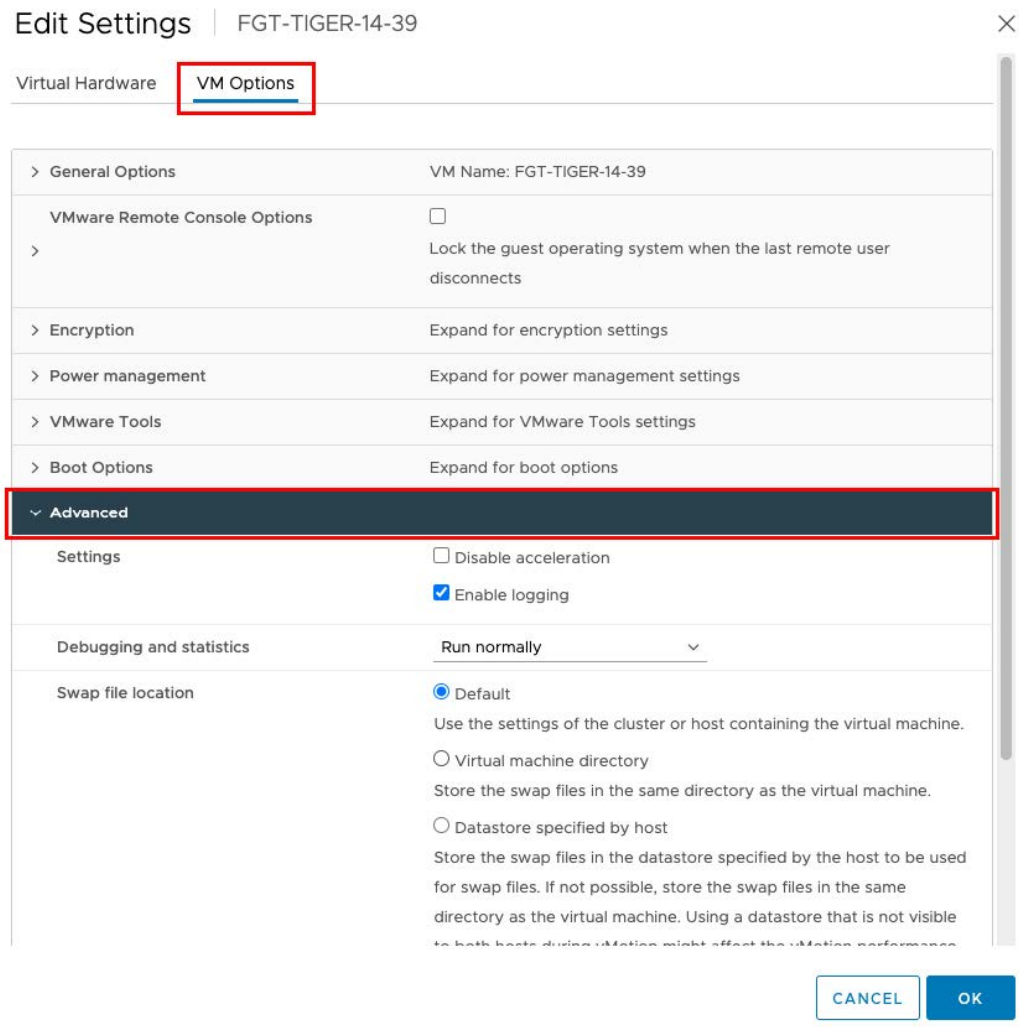


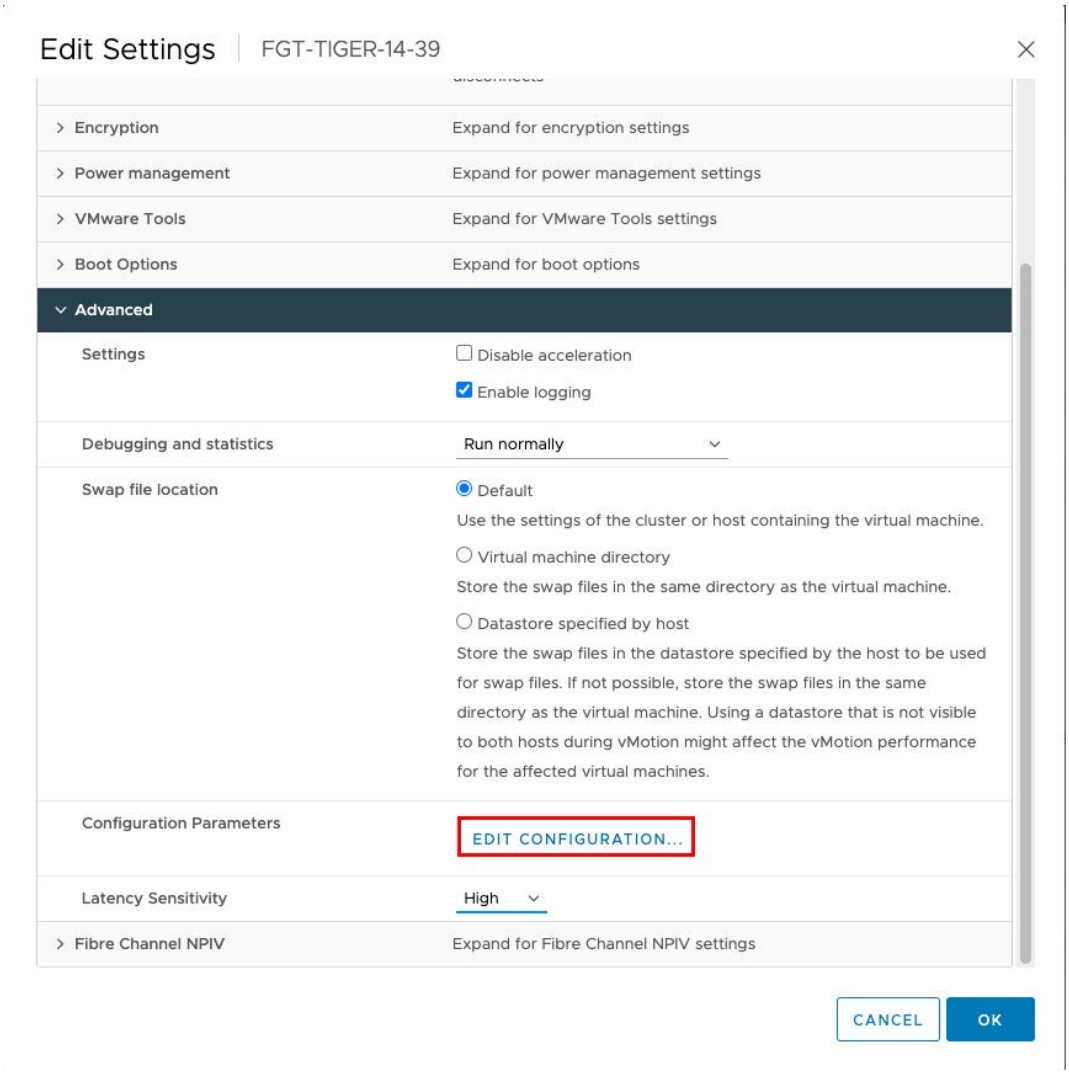
A diagram, such as the following, may be useful to ensure that your SR-IOV interfaces are as expected. Configuration order makes a difference.



Completing tuning the FortiGate-VM instance

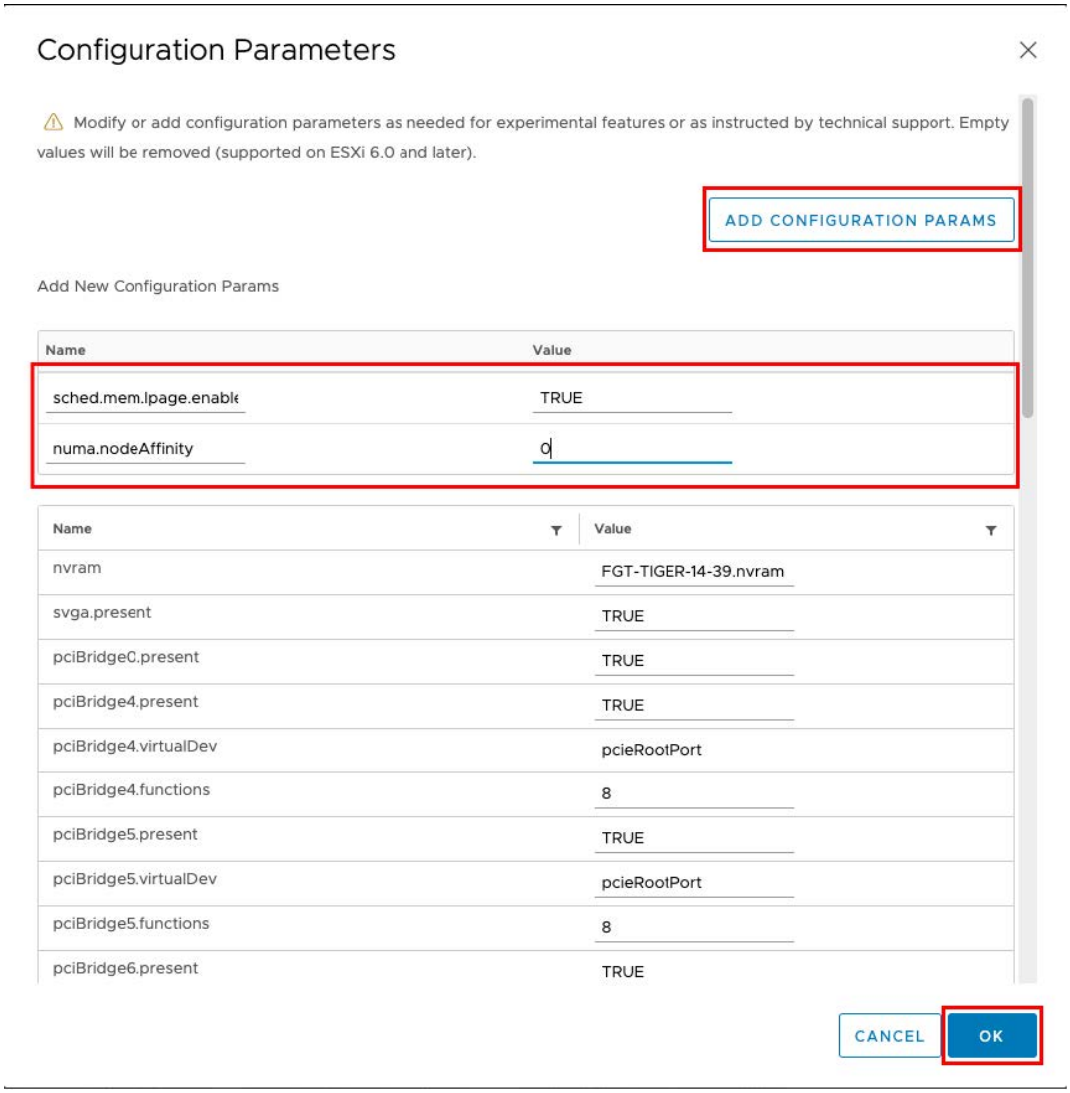
You must make a few more tuning configurations. So following where you performed the hardware configuration, go to the *VM Options* tab. Under *Advanced*, click *EDIT CONFIGURATION*.


Edit Settings
FGT-TIGER-14-39
Virtual Hardware
VM Options
General Options
VM Name: FGT-TIGER-14-39
VMware Remote Console Options
Lock the guest operating system when the last remote user disconnects
Encryption
Expand for encryption settings
Power management
Expand for power management settings
VMware Tools
Expand for VMware Tools settings
Boot Options
Expand for boot options
Advanced
Settings
Disable acceleration
Enable logging
Debugging and statistics
Run normally
Swap file location
Default
Use the settings of the cluster or host containing the virtual machine.
Virtual machine directory
Store the swap files in the same directory as the virtual machine.
Datastore specified by host
Store the swap files in the datastore specified by the host to be used for swap files. If not possible, store the swap files in the same directory as the virtual machine. Using a datastore that is not visible
CANCEL
OK

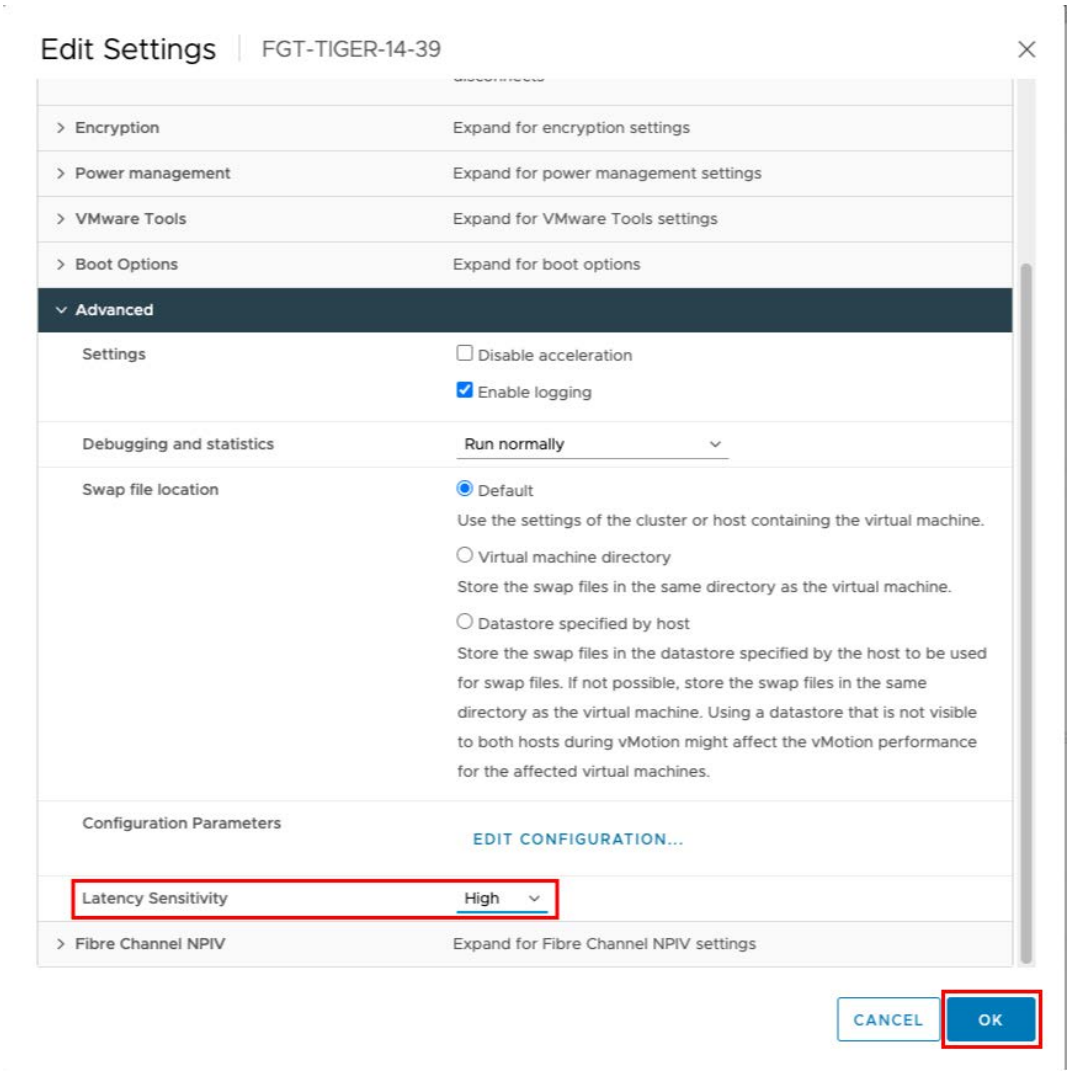


Click *ADD CONFIGURATION PARAMS*, and configure the parameters:

- Add the 1G hugepage support by adding the parameter `sched.mem.lpage.enable1GPage` set to *Value* `TRUE`.
- Define NUMA node affinity by adding the parameter `numa.nodeAffinite` set to *Value* of the NUMA node being used, in this case 0.



From the *Latency Sensitivity* dropdown list, select *High*.



VMX file

You can cautiously edit VMX file manually. It is a good way to confirm that the guest settings and tunings are in place.

```
[root@esxi-tiger-14-7:~] cat /vmfs/volumes/ESXI-TIGER-14-7/FGT-TIGER-14-39/FGT-TIGER-14-39.vmx
.encoding = "UTF-8"
config.version = "8"
virtualHW.version = "17"
vmci0.present = "TRUE"
floppy0.present = "FALSE"
memSize = "81920"
vm.createDate = "1646573404623737"
scsi0.virtualDev = "lsilogic"
scsi0.present = "TRUE"
scsi0:0.deviceType = "scsi-hardDisk"
scsi0:0.fileName = "FGT-TIGER-14-39.vmdk"
scsi0:0.present = "TRUE"
scsi0:1.deviceType = "scsi-hardDisk"
```

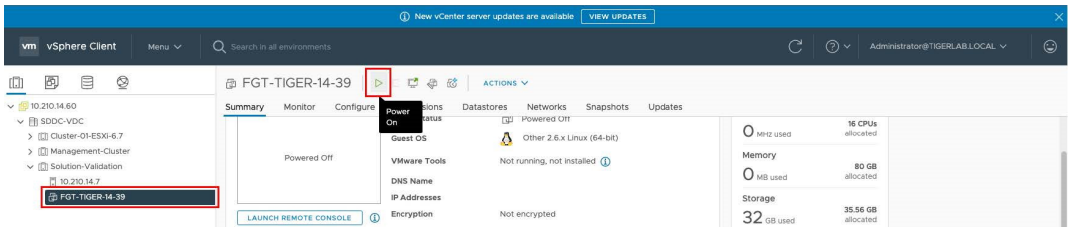
```
scsi0:1.fileName = "FGT-TIGER-14-39_1.vmdk"
scsi0:1.present = "TRUE"
ethernet0.virtualDev = "vmxnet3"
ethernet0.networkName = "VM Network"
ethernet0.addressType = "vpx"
ethernet0.generatedAddress = "00:50:56:8e:ba:38"
ethernet0.uptCompatibility = "TRUE"
ethernet0.present = "TRUE"
displayName = "FGT-TIGER-14-39"
annotation = "FortiGate Virtual Appliance by Fortinet Technologies Inc.
(http://www.fortinet.com)"
guestOS = "other26xlinux-64"
uuid.bios = "42 0e 06 93 f6 74 44 3a-25 fc 02 4f 45 b5 43 b7"
vc.uuid = "50 0e d3 4f 11 b1 8a 75-74 78 02 ab 0f 2f cc b2"
numvcpus = "16"
sched.cpu.units = "mhz"
sched.cpu.affinity = "2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17"
sched.cpu.min = "48688"
sched.cpu.shares = "normal"
sched.mem.min = "81920"
sched.mem.minSize = "81920"
sched.mem.shares = "normal"
sched.mem.pin = "TRUE"
pciPassthru15.MACAddressType = "static"
pciPassthru15.MACAddress = "52:54:00:14:39:41"
pciPassthru15.networkName = "FGT-TIGER-14-39_1002"
pciPassthru15.pfId = "00000:059:00.0"
pciPassthru15.deviceId = "0"
pciPassthru15.vendorId = "0"
pciPassthru15.systemId = "BYPASS"
pciPassthru15.id = "00000:059:00.0"
pciPassthru15.allowMTUChange = "TRUE"
pciPassthru15.present = "TRUE"
pciPassthru14.MACAddressType = "static"
pciPassthru14.MACAddress = "52:54:00:14:39:52"
pciPassthru14.networkName = "FGT-TIGER-14-39_1002"
pciPassthru14.pfId = "00000:059:00.1"
pciPassthru14.deviceId = "0"
pciPassthru14.vendorId = "0"
pciPassthru14.systemId = "BYPASS"
pciPassthru14.id = "00000:059:00.1"
pciPassthru14.allowMTUChange = "TRUE"
pciPassthru14.present = "TRUE"
pciPassthru13.MACAddressType = "static"
pciPassthru13.MACAddress = "52:54:00:14:39:63"
pciPassthru13.networkName = "FGT-TIGER-14-39_1002"
pciPassthru13.pfId = "00000:094:00.0"
pciPassthru13.deviceId = "0"
pciPassthru13.vendorId = "0"
pciPassthru13.systemId = "BYPASS"
pciPassthru13.id = "00000:094:00.0"
pciPassthru13.allowMTUChange = "TRUE"
pciPassthru13.present = "TRUE"
pciPassthru12.MACAddressType = "static"
pciPassthru12.MACAddress = "52:54:00:14:39:74"
pciPassthru12.networkName = "FGT-TIGER-14-39_1002"
```

```
pciPassthru12.pfId = "00000:094:00.1"
pciPassthru12.deviceId = "0"
pciPassthru12.vendorId = "0"
pciPassthru12.systemId = "BYPASS"
pciPassthru12.id = "00000:094:00.1"
pciPassthru12.allowMTUChange = "TRUE"
pciPassthru12.present = "TRUE"
pciPassthru11.MACAddressType = "static"
pciPassthru11.MACAddress = "52:54:00:14:39:45"
pciPassthru11.networkName = "FGT-TIGER-14-39_1001"
pciPassthru11.pfId = "00000:059:00.0"
pciPassthru11.deviceId = "0"
pciPassthru11.vendorId = "0"
pciPassthru11.systemId = "BYPASS"
pciPassthru11.id = "00000:059:00.0"
pciPassthru11.allowMTUChange = "TRUE"
pciPassthru11.present = "TRUE"
pciPassthru10.MACAddressType = "static"
pciPassthru10.MACAddress = "52:54:00:14:39:56"
pciPassthru10.networkName = "FGT-TIGER-14-39_1001"
pciPassthru10.pfId = "00000:059:00.1"
pciPassthru10.deviceId = "0"
pciPassthru10.vendorId = "0"
pciPassthru10.systemId = "BYPASS"
pciPassthru10.id = "00000:059:00.1"
pciPassthru10.allowMTUChange = "TRUE"
pciPassthru10.present = "TRUE"
pciPassthru9.MACAddressType = "static"
pciPassthru9.MACAddress = "52:54:00:14:39:67"
pciPassthru9.networkName = "FGT-TIGER-14-39_1001"
pciPassthru9.pfId = "00000:094:00.0"
pciPassthru9.deviceId = "0"
pciPassthru9.vendorId = "0"
pciPassthru9.systemId = "BYPASS"
pciPassthru9.id = "00000:094:00.0"
pciPassthru9.allowMTUChange = "TRUE"
pciPassthru9.present = "TRUE"
pciPassthru8.MACAddressType = "static"
pciPassthru8.MACAddress = "52:54:00:14:39:78"
pciPassthru8.networkName = "FGT-TIGER-14-39_1001"
pciPassthru8.pfId = "00000:094:00.1"
pciPassthru8.deviceId = "0"
pciPassthru8.vendorId = "0"
pciPassthru8.systemId = "BYPASS"
pciPassthru8.id = "00000:094:00.1"
pciPassthru8.allowMTUChange = "TRUE"
pciPassthru8.present = "TRUE"
nvram = "FGT-TIGER-14-39.nvram"
svga.present = "TRUE"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
```

```
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
hpet0.present = "TRUE"
viv.moid = "41bb06bb-1177-475f-8154-82adac9e7814:vm-
11541:7lE/kldSBFEnAkVeYp6O2XfVhshoKkdYhGdP9+13TYk="
migrate.hostLog = "FGT-TIGER-14-39-44b6bccc.hlog"
sched.cpu.latencySensitivity = "high"
sched.mem.lpage.enable1GPage = "TRUE"
numa.nodeAffinity = "0"
```

VM start and startup

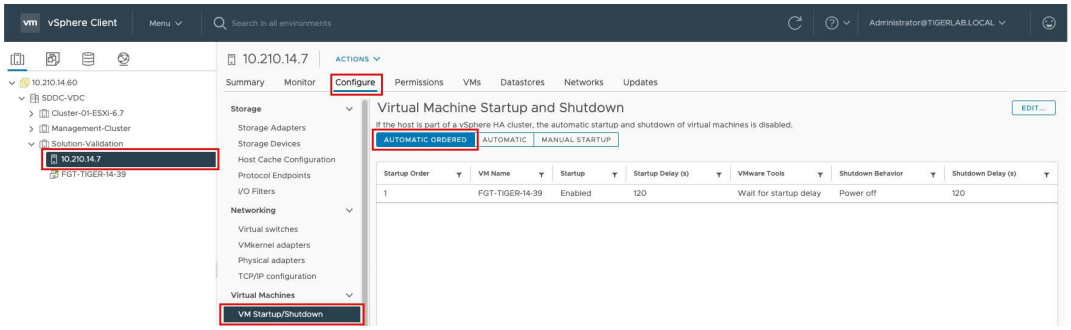
You can now restart the FortiGate-VM instance.



It may be of interest to show that the VFs are now owned and active:

```
root@esxi-tiger-14-7:~] esxcli network sriovnic vf list -n vmnic4
VF ID  Active  PCI Address      Owner World ID
-----  ------  ---------------  --------------
    0    true  00000:059:02.0   2106346
    1    true  00000:059:02.1   2106346
    2   false  00000:059:02.2   -
    3   false  00000:059:02.3   -
    4   false  00000:059:02.4   -
    5   false  00000:059:02.5   -
    6   false  00000:059:02.6   -
    7   false  00000:059:02.7   -
```

With the 1G hugepage setting, autostarting the FortiGate-VM instance automatically and first is advisable.



FortiGate-VM

Now the hypervisor is configured and the VM is running. Consideration for the FGVM setup is needed. These considerations could impact the decisions made on the hypervisor, so some updates may be required.

Depending on the FortiGate-VM use case, you should consider one or more of the following:

- SR-IOV will typically be used for any performant deployment. This does have a drawback on the VM mobility.
- vSPU gives significant performance uplift in many scenarios and will continue to develop.
- Not using vSPU or not using all CPUs for vSPU may be more performant in some scenarios.
- Without vSPU, balancing interrupts across all CPUs by using affinity settings is key to getting maximum performance.

SR-IOV, LAGs, and affinity

For use cases that do not currently benefit from vSPU, the best that you can do is load balancing across all CPUs. You can best achieve this using SR-IOV, LAGs, and CPU affinity settings.

You likely need link aggregation, if not for throughput, for resiliency. Considerations for LAG differ when considering VFs, but the main concepts are the same. The following diagram represents an example LAG-based topology based on having two NIC cards, each with two ports in a single NUMA node.

This scenario tolerates the following:

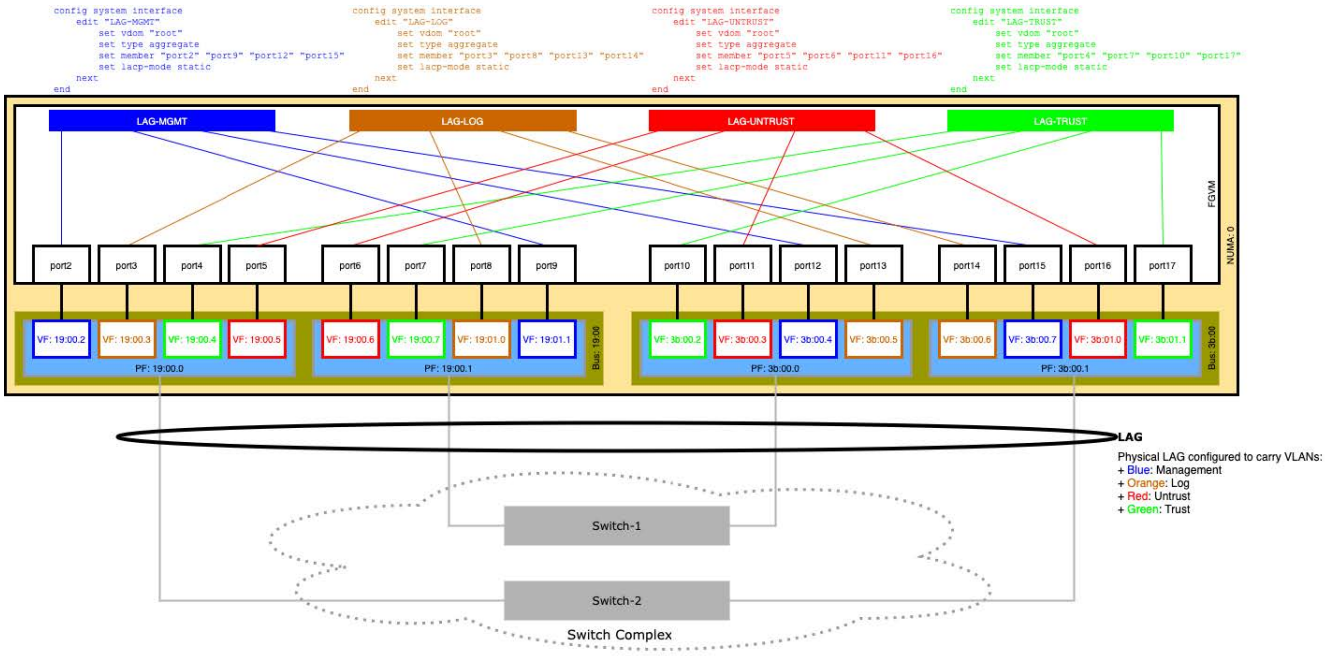
- NIC port/link failure
- NIC card failure
- Switch failure

The design also stresses the need for the `trust on` setting discussed earlier, as the VF must react upon the status of the PF, as LACP is not going to provide the functionality it would do in an

appliance-based deployment.

You must configure LACP mode as static in this deployment scenario.

In this diagram, the PF is using an external VLAN tag to separate traffic to the respective VFs and the VM is unaware of this external VLAN.



Without vSPU, there is no PMD, and the NIC uses the interrupts are used to signal that there is network traffic that the CPU must process. To get a performant system without using vSPU, you must take care to balance the amount of interrupts that each CPU receives.

Using the same layout as the diagram displays, find the relevant system interrupts/queues:

```
diagnose hardware sysinfo interrupts  grep "CPUport"
           CPU0       CPU1    <...>   CPU15
 47:      119912          0    <...>       0   PCI-MSI-edge      iavf-port2-TxRx-0
 48:           0     200309    <...>       0   PCI-MSI-edge      iavf-port2-TxRx-1
 49:           0          0    <...>       0   PCI-MSI-edge      iavf-port2-TxRx-2
 50:           0          0    <...>       0   PCI-MSI-edge      iavf-port2-TxRx-3
 <...>
 67:      254849          0    <...>       0   PCI-MSI-edge      iavf-port6-TxRx-0
 68:           0     443186    <...>       0   PCI-MSI-edge      iavf-port6-TxRx-1
 69:           0          0    <...>       0   PCI-MSI-edge      iavf-port6-TxRx-2
 70:           0          0    <...>       0   PCI-MSI-edge      iavf-port6-TxRx-3
 <...>
 87:       72971          0    <...>       0   PCI-MSI-edge      iavf-port10-TxRx-0
 88:           0     376044    <...>       0   PCI-MSI-edge      iavf-port10-TxRx-1
 89:           0          0    <...>       0   PCI-MSI-edge      iavf-port10-TxRx-2
 90:           0          0    <...>       0   PCI-MSI-edge      iavf-port10-TxRx-3
 <...>
 107:     197132          0    <...>       0   PCI-MSI-edge      iavf-port14-TxRx-0
 108:          0     421851    <...>       0   PCI-MSI-edge      iavf-port14-TxRx-1
 109:          0          0    <...>       0   PCI-MSI-edge      iavf-port14-TxRx-2
 110:          0          0    <...>       0   PCI-MSI-edge      iavf-port14-TxRx-3
 <...>
 122:          0          0    <...>       0   PCI-MSI-edge      iavf-port17-TxRx-0
 123:          0          0    <...>       0   PCI-MSI-edge      iavf-port17-TxRx-1
 124:          0          0    <...>       0   PCI-MSI-edge      iavf-port17-TxRx-2
 125:          0          0    <...>  345768   PCI-MSI-edge      iavf-port17-TxRx-3
 <...>
```

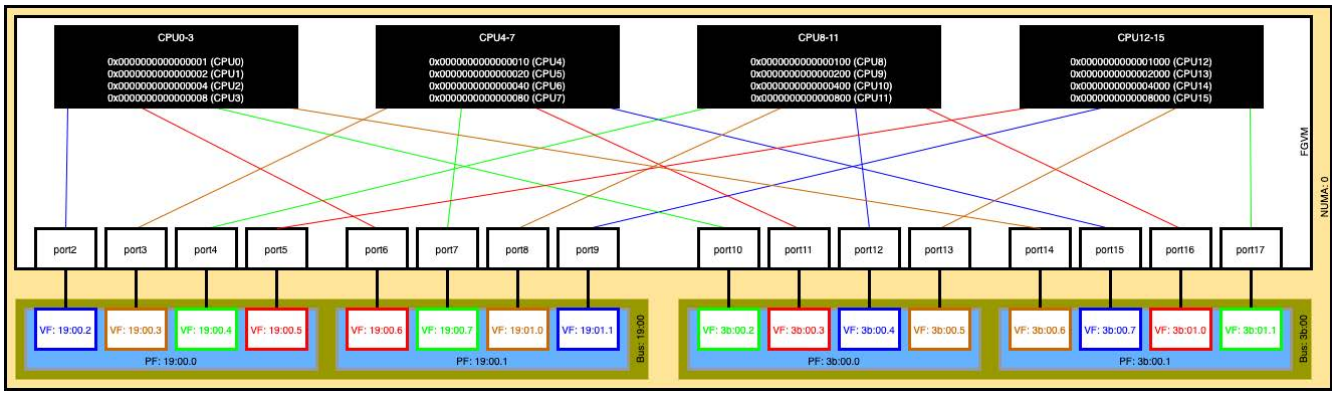
The interrupt names can differ. For example, the Mellanox ConnectX-5 NIC card has ten interrupts/queues per port named port2-0 through to port2-9.

The idea is to spread the interrupts across CPUs to balance the load across all system resources. Using the interrupt names as per the print, you can pin them to particular CPUs:

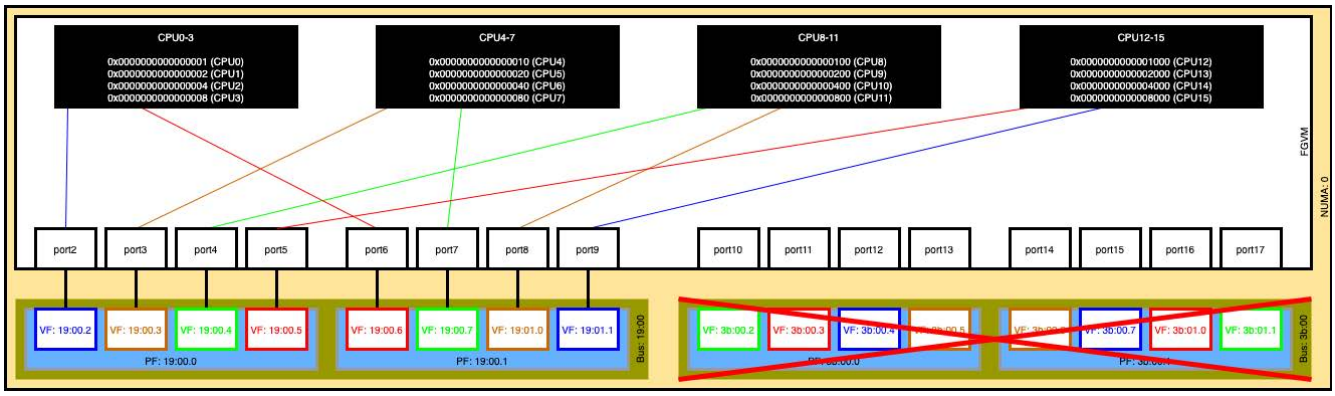
```
config system affinity-interrupt
    edit 20
        set interrupt "iavf-port2-TxRx-0"
        set affinity-cpumask "0x0000000000000001"
    next
    edit 21
        set interrupt "iavf-port2-TxRx-1"
        set affinity-cpumask "0x0000000000000002"
    next
    edit 22
        set interrupt "iavf-port2-TxRx-2"
        set affinity-cpumask "0x0000000000000004"
    next
    edit 23
        set interrupt "iavf-port2-TxRx-3"
        set affinity-cpumask "0x0000000000000008"
    next
<...>
    edit 60
        set interrupt "iavf-port6-TxRx-0"
        set affinity-cpumask "0x0000000000000001"
    next
    edit 61
        set interrupt "iavf-port6-TxRx-1"
        set affinity-cpumask "0x0000000000000002"
    next
    edit 62
        set interrupt "iavf-port6-TxRx-2"
        set affinity-cpumask "0x0000000000000004"
    next
    edit 63
        set interrupt "iavf-port6-TxRx-3"
        set affinity-cpumask "0x0000000000000008"
    next
<...>
    edit 170
        set interrupt "iavf-port17-TxRx-0"
        set affinity-cpumask "0x0000000000001000"
    next
    edit 171
        set interrupt "iavf-port17-TxRx-1"
        set affinity-cpumask "0x0000000000002000"
    next
    edit 172
        set interrupt "iavf-port17-TxRx-2"
        set affinity-cpumask "0x0000000000004000"
    next
    edit 173
```

```
        set interrupt "iavf-port17-TxRx-3"
        set affinity-cpumask "0x0000000000008000"
    next
end
```

This is a mapping of the four queues on an interface to one of four CPUs in a group, but also reusing the group of four CPUs across four interfaces as the following diagrams. This interleaving of the functions gets an even interrupt distribution, which gives the most performant deployment scenario.



In case of a failure, for example of the NIC card, this interleaving model ensures that the traffic interfaces where most traffic is expected are processed by different CPUs as the diagram shows, keeping the performance to a maximum.



Working out how best to balance the interrupts is the main thing to address in these circumstances. In the example case, each port has four queues/interrupts that you can map, making a VM16 effective with four PFs. The SR-IOV VLAN filtering and resultant LAG configuration provides interleaving, which helps balance the load across all CPUs.

Simularly, it may be that a VM32 is best serviced with eight PFs. It may be that the NIC card allows configuration of how many PFs are presented. For example, you may use an NIC presenting 4 x 10G more effectively across the CPUs than 1 x 40G.

Without much flexibility in using transparent VLANs or number of PFs, affining some services such as IPS, logging, or Web Filter to CPUs unused for traffic and providing effective CPU use may be the best option.

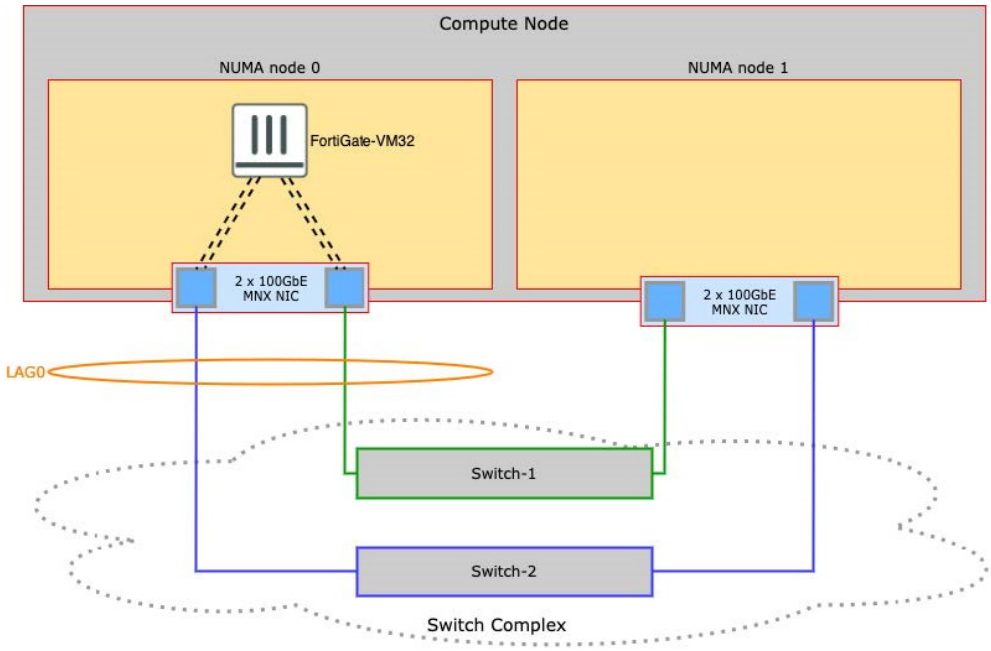
Effectively, there is significant flexibility, which should allow you to find a sweet spot of performance in most scenarios.

vSPU

Following is a diagram and the associated configuration. In the diagram, the blue square represents the PF. The dotted line represents the VF. Two VFs are defined for each PF. The configuration uses VLANs 1000 and 1001 to direct the traffic between VF and PF. The VM is unaware of the VLAN.

VLAN 1000 and VLAN 1001 are on opposites sides of the firewall. However, as this is presented to the VM as four devices, you can use link aggregation across the two PFs to cater for an element of resiliency.

The vSPU deployment negates the need for balancing interrupt requests and traffic is balanced across the vSPUs based upon IP headers.



FortiGate vSPU on page 54 describes balancing the traffic with vSPU. It does not require the configuration to use a balancing technique.

DPDK global settings

You must first enable DPDK and associate it to the interfaces which DPDK will polled for traffic. Enabling DPDK for the first time requires a system reboot.

```
config dpdk global
   set status enable
   set interface "port2" "port3" "port4" "port5"
   set multiqueue enable
   set sleep-on-idle disable
   set elasticbuffer disable
   set per-session-accounting traffic-log-only
   set hugepage-percentage 25
   set mbufpool-percentage 20
end
```

See DPDK global settings on page 45 for a detailed explanation of these configuration items.

You can then use these interfaces as normal in FortiOS. The following uses these interfaces to create LAGs to handle traffic:

```
config system interface
   edit "LAG-IN"
      set vdom "root"
      set ip 10.0.0.254 255.255.0.0
      set allowaccess ping
      set type aggregate
      set member "port2" "port3"
      set lldp-reception disable
      set lldp-transmission disable
      set snmp-index 7
      set lacp-mode static
   next
   edit "LAG-OUT"
      set vdom "root"
      set ip 10.1.0.254 255.255.0.0
      set allowaccess ping
      set type aggregate
      set member "port4" "port5"
      set lldp-reception disable
      set lldp-transmission disable
      set snmp-index 8
      set lacp-mode static
   next
end
```

DPDK CPU settings

The CPUs acting as DPDK engines are specified. They are four stages, a processing pipeline, for handling packets from Rx to vNP to IPS toTx. Generally, the simplest allocation model, enabling all CPUs to all stages, gives the best results.

```
config dpdk cpus
   set rx-cpus "0-31"
   set vnp-cpus "0-31"
   set ips-cpus "0-31"
   set tx-cpus "0-31"
end
```

See DPDK CPU settings on page 48 for a detailed explanation of these configuration items.

There are times when ringfencing CPUs to be used for purposes other than DPDK can provide a more performant system.

In FortiOS 7.0.2, `set isolated-cpus` was introduced to protect the DPDK applications from non-DPDK system operations. An example of using this feature is with the ESXi i40en PF driver. The driver allows the DPDK environment to address only a shallow Rx buffer by making the DPDK Rx stage less tolerant to CPU cycles being used elsewhere. This setting allows protection of the Rx stage decreasing the chance of Rx dropped packets.

DPDK diagnostics

See DPDK diagnostic commands on page 49 for a detailed explanation of these configuration items.

Early initialization

You can use the DPDK early initialization log to check that you have configured DPDK correctly. For example, you can confirm that the CPUs and interfaces are bound to DPDK usage.

```
diagnose dpdk log show early-init
 ------------------------------------------------------------------
     DPDK early initialzation starts at 2020-11-26 09:53:14(UTC)
 ------------------------------------------------------------------
 Content of early configuration file:
     status=1
     multiqueue=1
     sleep-on-idle=0
     elasticbuffer=0
     per-session-accounting=1
     hugepage-percentage=25
     nr_hugepages=10090
     interfaces=port2 port3 port4 port5
     cpus=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31
     rxcpus=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31
     vnpcpus=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31
     ipscpus=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31
     txcpus=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31
 Parse config file success!

 Check CPU definitions 'cpus'
 Check CPU definitions 'rxcpus'
 Check CPU definitions 'ipscpus'
 Check CPU definitions 'vnpcpus'
 Check CPU definitions 'txcpus'
 Check CPUs success!

 Huge page allocation done

 Ports enabled for DPDK:
     port2
     port3
     port4
     port5
 Port name to device name mapping:
     port1: eth0
     port2: eth1
     port3: eth2
     port4: eth3
     port5: eth4
     port6: eth5
```

```
        port7: eth6
        port8: eth7
        port9: eth8
        port10: eth9
        port11: eth10
        port12: eth11
        port13: eth12
        port14: eth13
        port15: eth14
        port16: eth15
        port17: eth16
        port18: eth17
        port19: eth18
        port20: eth19
        port21: eth20
        port22: eth21
        port23: eth22
        port24: eth23

    Start enabling DPDK kernel driver for port 'port2'...
    Getting PCI device info for eth1...
    reading pci dev /sys/class/net/eth1
    link path: ../../devices/pci0000:00/0000:00:08.0/net/eth1
    Device info of eth1:
        dev_name: eth1
        macaddr: 52:54:00:7c:08:31
        pci_vendor: 0x15b3
        pci_device: 0x1018
        pci_id: 0000:00:08.0
        pci_domain: 0
        pci_bus: 0
        pci_devid: 8
        pci_function: 0
        guid: n/a
    Device eth1 is mlx5_core name changed to slv1
    Creating DPDK kernel driver for device eth1...
    Add VNP dev: eth1 PCI: 0000:00:08.0, Succeeded
    DPDK kernel driver for eth1 successfully created
    DPDK kernel driver enabled for port 'port2' (device name 'eth1')

    Start enabling DPDK kernel driver for port 'port3'...
    Getting PCI device info for eth2...
    reading pci dev /sys/class/net/eth2
    link path: ../../devices/pci0000:00/0000:00:09.0/net/eth2
    Device info of eth2:
        dev_name: eth2
        macaddr: 52:54:00:7c:08:32
        pci_vendor: 0x15b3
        pci_device: 0x1018
        pci_id: 0000:00:09.0
        pci_domain: 0
        pci_bus: 0
        pci_devid: 9
        pci_function: 0
        guid: n/a
    Device eth2 is mlx5_core name changed to slv2
```

```
Creating DPDK kernel driver for device eth2...
Add VNP dev: eth2 PCI: 0000:00:09.0, Succeeded
DPDK kernel driver for eth2 successfully created
DPDK kernel driver enabled for port 'port3' (device name 'eth2')

Start enabling DPDK kernel driver for port 'port4'...
Getting PCI device info for eth3...
reading pci dev /sys/class/net/eth3
link path: ../../devices/pci0000:00/0000:00:0a.0/net/eth3
Device info of eth3:
    dev_name: eth3
    macaddr: 52:54:00:7c:08:33
    pci_vendor: 0x15b3
    pci_device: 0x1018
    pci_id: 0000:00:0a.0
    pci_domain: 0
    pci_bus: 0
    pci_devid: 10
    pci_function: 0
    guid: n/a
Device eth3 is mlx5_core name changed to slv3
Creating DPDK kernel driver for device eth3...
Add VNP dev: eth3 PCI: 0000:00:0a.0, Succeeded
DPDK kernel driver for eth3 successfully created
DPDK kernel driver enabled for port 'port4' (device name 'eth3')

Start enabling DPDK kernel driver for port 'port5'...
Getting PCI device info for eth4...
reading pci dev /sys/class/net/eth4
link path: ../../devices/pci0000:00/0000:00:0b.0/net/eth4
Device info of eth4:
    dev_name: eth4
    macaddr: 52:54:00:7c:08:34
    pci_vendor: 0x15b3
    pci_device: 0x1018
    pci_id: 0000:00:0b.0
    pci_domain: 0
    pci_bus: 0
    pci_devid: 11
    pci_function: 0
    guid: n/a
Device eth4 is mlx5_core name changed to slv4
Creating DPDK kernel driver for device eth4...
Add VNP dev: eth4 PCI: 0000:00:0b.0, Succeeded
DPDK kernel driver for eth4 successfully created
DPDK kernel driver enabled for port 'port5' (device name 'eth4')
Bind ports success!

Make UIO nodes success!

DPDK sanity test passed
```

DPDK engine utilization

You can use `diagnose dpdk performance show` to see how the DPDK engines are loaded. This could be the information source for tuning the system or spotting irregular traffic load balancing.

The CPU usage will be reported at 100% while the CPU is configured as a DPDK engine because of the PMD. This output is the source to look at for proper utilization reporting. These utilizations are available by SNMP.

MPStat

You can use the `mpstat` utility when trying to understand where a system is losing performance or to find an indication of an issue.

```
diagnose sys mpstat 2 3
Gathering data, wait 2 sec, press any key to quit.
..0..1
TIME         CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal   %idle
05:55:32 PM all   88.70    0.00   11.30    0.00    0.00    0.00    0.00    0.00
               0   90.00    0.00   10.00    0.00    0.00    0.00    0.00    0.00
               1   91.00    0.00    9.00    0.00    0.00    0.00    0.00    0.00
               2   89.00    0.00   11.00    0.00    0.00    0.00    0.00    0.00
               3   89.50    0.00   10.50    0.00    0.00    0.00    0.00    0.00
               4   86.50    0.00   13.50    0.00    0.00    0.00    0.00    0.00
               5   90.50    0.00    9.50    0.00    0.00    0.00    0.00    0.00
               6   89.00    0.00   11.00    0.00    0.00    0.00    0.00    0.00
               7   89.00    0.00   11.00    0.00    0.00    0.00    0.00    0.00
               8   88.50    0.00   11.50    0.00    0.00    0.00    0.00    0.00
               9   86.50    0.00   13.50    0.00    0.00    0.00    0.00    0.00
              10   88.50    0.00   11.50    0.00    0.00    0.00    0.00    0.00
              11   88.50    0.00   11.50    0.00    0.00    0.00    0.00    0.00
              12   88.50    0.00   11.50    0.00    0.00    0.00    0.00    0.00
              13   87.50    0.00   12.50    0.00    0.00    0.00    0.00    0.00
              14   89.00    0.00   11.00    0.00    0.00    0.00    0.00    0.00
              15   91.00    0.00    9.00    0.00    0.00    0.00    0.00    0.00
              16   90.50    0.00    9.50    0.00    0.00    0.00    0.00    0.00
              17   90.50    0.00    9.50    0.00    0.00    0.00    0.00    0.00
              18   91.50    0.00    8.50    0.00    0.00    0.00    0.00    0.00
              19   91.00    0.00    9.00    0.00    0.00    0.00    0.00    0.00
              20   85.00    0.00   15.00    0.00    0.00    0.00    0.00    0.00
              21   90.00    0.00   10.00    0.00    0.00    0.00    0.00    0.00
              22   87.50    0.00   12.50    0.00    0.00    0.00    0.00    0.00
              23   92.50    0.00    7.50    0.00    0.00    0.00    0.00    0.00
              24   89.50    0.00   10.50    0.00    0.00    0.00    0.00    0.00
              25   84.00    0.00   16.00    0.00    0.00    0.00    0.00    0.00
              26   85.00    0.00   15.00    0.00    0.00    0.00    0.00    0.00
              27   91.00    0.00    9.00    0.00    0.00    0.00    0.00    0.00
              28   87.50    0.00   12.50    0.00    0.00    0.00    0.00    0.00
              29   84.50    0.00   15.50    0.00    0.00    0.00    0.00    0.00
              30   87.50    0.00   12.50    0.00    0.00    0.00    0.00    0.00
              31   88.50    0.00   11.50    0.00    0.00    0.00    0.00    0.00
<output omitted for brevity>
```

As discussed, the idle time for this print reports as 0% because of PMD.

Of particular importance in this print is `%steal`. If this is not zero, something is not optimized, as the hypervisor is stealing CPU cycles from the VM.

Change log

Date	Change Description
2023-05-11	Initial release.

www.fortinet.com