FortiGate-VM - Install Guide for KVM

Version 6.0

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

FORTINET VIDEO GUIDE
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/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://www.fortiguard.com

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

FEEDBACK
Email: techdoc@fortinet.com

September 25, 2019
FortiGate-VM 6.0 Install Guide for KVM
01-601-498214-20190925

TABLE OF CONTENTS

About FortiGate-VM on KVM

FortiGate virtual appliances 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 virtual appliances feature all the security and networking services common to hardware-based FortiGate appliances. You can deploy a mix of FortiGate hardware and virtual appliances, operating together and managed from a common centralized management platform.

This document describes how to deploy a FortiGate virtual appliance in a KVM environment.

FortiGate-VM models and licensing

Fortinet offers the FortiGate virtual machine (FortiGate-VM) in five virtual appliance models, which are determined by license. When configuring the FortiGate-VM, ensure that the hardware settings are within the ranges outlined below. Contact your Fortinet-authorized reseller for more information.

The following table summarizes FortiGate-VM model information:

Technical specification	FG-VM00	FG-VM01	FG-VM02	FG-VM04	FG-VM08
Virtual CPUs (min / max)	1 / 1	1 / 1	1 / 2	1 / 4	1 / 8
Virtual network interfaces (min / max)			2 / 18		
Virtual memory (min / max)	1 GB / 2 GB	1 GB / 2 GB	1 GB / 4 GB	1 GB / 6 GB	1 GB /12 GB
Virtual storage (min / max)			32 GB / 2 TB		
Managed wireless APs (tunnel mode / global)	32 / 32	32 / 64	256 / 512	256 / 512	1024 / 4096
Virtual domains (VDOM) (default / max)	1 / 2	10 / 10	10 / 25	10 / 50	10 / 250

The minimum and maximum values can change. In this case, manually change the settings for the VM to accommodate the new parameters.

When you submit an order for a FortiGate-VM virtual appliance, 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 virtual appliance with Customer Service & Support, and then download the license file. After you upload the license to the FortiGate-VM virtual appliance and validate it, your FortiGate-VM virtual appliance is fully functional.

The number of virtual network interfaces does not solely depend on the FortiGate-VM. Some virtual environments have their own limitations on the number of interfaces allowed.

FortiGate-VM virtual appliance evaluation license

The FortiGate-VM virtual appliance includes a limited 15-day evaluation license that supports:

- 1 CPU maximum
- 1024 MB memory maximum
- Low encryption only (no HTTPS administrative access)
- All features except FortiGuard updates

Note the following:

- Attempting to upgrade the FortiGate firmware locks the GUI until you upload a full license.
- Technical support is not included. The trial period begins the first time you start the FortiGate-VM.
- After the trial license expires, functionality is disabled until you upload a full license file.

FortiGate-VM virtual licenses and resources

The primary requirement for the provisioning of a virtual FortiGate 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.

FortiOS 6.0.1 and earlier

Previously, if you needed a virtual instance with a high number of interfaces you needed to purchase a FortiGate-VM license for a high number of vCPUs regardless of whether you needed the processing power. If you attempt to install FortiGate-VM, licensed for a specific number of vCPUs on a public cloud instance that is configured with more vCPUs than the FortiGate is licensed for, the instance will not run.

Example for FortiOS 6.0.1 and earlier

License	1 vCPU	2 vCPU	4 vCPU	8 vCPU	16 vCPU	32 vCPU
FGT-VM08	OK	OK	OK	OK	Will not run	Will not run

FortiOS 6.0.2 and later

The licensing for FortiGate-VM 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 indicated by the license does not restrict the FortiGate from working, regardless of how many vCPUs are included in the virtual instance. However, only the licensed number of vCPUs process traffic and management tasks. The rest of the vCPUs are unused.

Example for FortiOS 6.0.2 and later

License	1 vCPU	2 vCPU	4 vCPU	8 vCPU	16 vCPU	32 vCPU
FGT-VM08	OK	OK	OK	OK	8 vCPUs used for traffic and management. The rest are not used.	8 vCPUs used for traffic and management. The rest are not used.

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 (ESXi/KVM/Xen/Hyper-V): Both licensed vCPUs and RAM are affected
- Public clouds (AWS/Azure/GCP/OCI/Aliyun): Only licensed vCPU is affected

For example, you can activate FG-VM02 on a FGT-VM with 4 vCPUs with 16 GB of RAM, running on a private VM platform. Only 2 vCPU and 4 GB of RAM, as licensed, will be consumable.

Likewise, you can activate FG-VM02 on a FGT-VM c5.2xlarge EC2 instance with 8 vCPUs running on AWS. Only 2 vCPU will be consumable, and there is no limit on the RAM size. Licenses for public clouds are also referred to as Bring Your Own License (BYOL).

Preparing for deployment

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

Virtual Environment

The KVM software is installed on a physical server with sufficient resources to support the FortiGate-VM and all other VMs that will be deployed on the platform.

If the FortiGate-VM will be configured to operate in transparent mode, or will be included in a FortiGate Clustering Protocol (FGCP) High Availability (HA) cluster, ensure that any virtual switches have been configured to support the operation of the FortiGate-VM before you create the FortiGate-VM.

Connectivity

An Internet connection is required for the FortiGate-VM to contact FortiGuard to validate its license. If the FortiGate-VM is in a closed environment, it must be able to connect to a FortiManager to validate the FortiGate-VM license. See Validating the FortiGate-VM license with FortiManager on page 22.

Configuring resources

Before you start the FortiGate-VM for the first time, ensure that the following resources are configured as specified by the FortiGate-VM virtual appliance license:

- Disk sizes
- CPUs
- RAM
- Network settings

To configure settings for FortiGate-VM on the server:

1. In the Virtual Machine Manager, locate the VM name and then select *Open* from the toolbar.
2. Select *Add Hardware*.
3. In the *Add Hardware* window select *Storage*.
4. Select *Create a disk image on the computer's harddrive* and set the size to 30 GB.

If you know your environment will expand in the future, it is recommended to increase the hard disk size beyond 30 GB. The VM license limit is 2 TB.

5. Enter:

Device type	Virtio disk
Cache mode	Default
Storage format	raw

Even though raw is the storage format listed, the qcow2 format is also supported.

6. Select *Network* to configure or add more network interfaces. The *Device type* must be *Virtio*.
   A new VM includes one network adapter by default. You can add more through the *Add Hardware* window. FortiGate-VM requires four network adapters. You can configure network adapters to connect to a virtual switch or to network adapters on the host computer.
7. Select *Finish*.

Registering the FortiGate-VM virtual appliance

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

To register the FortiGate-VM virtual appliance:

1. Log in to the Customer Service & Support site using a support account, or select *Sign Up* to create an account.
2. In the main page, under *Asset*, select *Register/Renew*.
3. In the *Registration* page, enter the registration code that was emailed to you, and select *Register* to access the registration form.
4. Complete and submit the registration form.
5. In the registration acknowledgment page, click the *License File Download* link.
6. Save the license file (`.lic`) to your local computer. See Uploading the FortiGate-VM virtual appliance license on page 21 or Validating the FortiGate-VM license with FortiManager on page 22 for information about uploading the license file to your FortiGate-VM via the GUI.

Downloading the FortiGate-VM virtual appliance 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.

1. In the *Select Product* drop-down menu, select *FortiGate*.
2. In the Select Platform drop-down menu, select KVM.

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 product datasheet available on the Fortinet web site, https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/FortiGate_VM.pdf.

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

The FORTINET.out.kvm.zip contains only fortios.qcow2, the FortiGate-VM system hard disk in qcow2 format. You must manually:

- create a 32 GB log disk
- specify the virtual hardware settings

Cloud-init

You can use the `cloud-init` service for customizing a prepared image of a virtual installation. The `cloud-init` service is built into the virtual instances of FortiGate-VM found on the support site so that you can use them on a VM platform that supports the use of the service. To customize the installation of a new FortiGate-VM instance, you must combine the seed image from the support site with user data information customized for each new installation.

Hypervisor platforms such as QEMU/KVM support the use of this service on most major Linux distributions, as well as BSD, and Hyper-V. A number of cloud-based environments such as VMware and AWS also support it.

You can use the `cloud-init` service to help install different instances based on a common seed image by assigning hostnames, adding SSH keys, and settings particular to the specific installation. You can add other more general customizations such as the running of post install scripts.

While `cloud-init` is the service used to accomplish the customized installations of VMs, various other programs, depending on the platform, are used to create the customized ISOs used to create the images that will build the FortiGate-VM.

For more information, see the cloud-init program documentation.

The basic steps of the process are:

1. Ensure that the needed software is on the system.
2. Prepare the files to customize the seed image.
3. Collect the customizing files into a single folder.
4. Convert the folder to an ISO image.
5. Install the image into the VM platform.

Installing the software

Installing Virt-install

Another required program is virt-install. Among other things, this program allows you to combine the user_data file with the seed image for the FortiGate-VM installation. To run virt-install, libvirt and KVM must be running on the system. You need root privileges to run virt-install

To install virt-install on a Red Hat-based system use the command:

```
sudo yum install libvirt libguestfs-tools virtinst
```

To install virt-install on a Debian-based system use the command:

```
sudo apt install libvert-bin libguestfs-tools virtinst
```

You can also install `virt-manager` using the same methods.

There may be other methods of installing the software, but these are two common methods.

Ensure that after the installation, `libvirt-bin` is installed and the service is running.

Verifying mkisofs is installed

Some distros like Ubuntu may have a variation of the program called `genisoimage`, but using the original `mkisofs` command should still work, as `mkisofs` is used as an alias for `genisoimage` in many of the distros.

To verify whether or not you have the utility installed, enter the command:

```
mkisofs --version
```

If your system has genisoimage instead, you may get a message along the lines of:

```
mkisofs 2.0.1 is not what you see here. This line is only a fake for too
     clever GUIs and other frontend applications. In fact, the program
     is:
  genisoimage 1.1.11 (Linux)
```

Preparing the files for the customized image

Preparing the user_data file

The `cloud-init` service passes a script to newly created VMs, in this case FortiGate-VM. The file's title is `user_data`. All configuration on the FortiGate is done through the configuration file so components of the scripts follow the configuration file syntax or commands entered through the CLI.

The following example content is from a basic user_data file:

```
#this is for fgt init config file. Can only handle fgt config.
config sys interface
   edit port1
      set mode dhcp
      set allowaccess http https ssh ping telnet
   next
end
config sys dns
   set primary 8.8.8.8
   unset secondary
end
config sys global
   set hostname cloud-init-test
end
```

License file

The other file that you use to configure the customized install contains the license key. Take the license key you receive from Fortinet and place it into a text file. This file is named `0000` without any extension.

Preparing the folder

There are no requirements for where to place the holding folder that will be used to create the new ISO image, but there are requirements for the folder structure within the folder. cloud-init must find specific content in specific folders to work correctly. The folder structure should be as follows:

```
<holding folder>
   /openstack
      /content
         0000
      /latest
         user_data
```

It may seem counter-intuitive to use the folder name `openstack` in an instance where the target VM platform is not OpenStack, but a number of utilities are common to both OpenStack and KVM environments.

Converting the folder to an ISO image

Once you have your user_data file and the license key file, you can create an ISO image containing all files that are used to customize the seed image of the FortiGate-VM. You can do this using the `mkisofs` utility.

The syntax of the command is:

```
mkisofs [options] [-o <filename of new ISO> pathspec [pathspec...]
```

Some options are:

Option	Description
`-o <filename>`	Sets the resulting ISO image filename.

Option	Description
`pathspec [pathspec...]`	Direction to the folder(s) to be included in the ISO image file. Separate the paths with a space.
`-input-charset`	Input charset that defines the characters used in local file names. To get a list of valid charset names, use the command `mkisofs -input-charset help`. To get a 1:1 mapping, you may use `default` as charset name.
`- R`	Generate SUSP and RR records using the Rock Ridge protocol to further describe the files on the iso9660 file system.
`- r`	This is like the -R option, but file ownership and modes are set to more useful values. The uid and gid are set to zero, because they are usually only useful on the author's system, and not useful to the client. All the file read bits are set true, so that files and directories are globally readable on the client.

The following is an example. The iso-folder holds the data structure for the new ISO image. The `/home/username/test` folder contains the `iso-folder`folder. The new ISO image name is fgt-bootstrap.iso.

```
cd /home/username/test
sudo mkisofs -R -r -o fgt-bootstrap.iso iso-folder
```

Installing the ISO in the VM platform

The following table contains some of the more common options used in setting up a FortiGate-VM image. Not all of them are required. To get a complete listing of the options, at the command prompt, type in the command `virt-install --help` or `virt-install -h`.

Option	Description
`--connect <option>`	This connects the VM image to a non-default VM platform. If one is not specified, libvirt will attempt to choose the most suitable default platform. Some valid options are: • qemu:///system Creates KEM and QEMU guests run by the system. This is the most common option. • qem:///session Creates KEM and QEMU guests run as a regular user. • xen:/// For connecting to Xen.
`--name <name>` `-n <name>`	Name of the new guest virtual machine instance. This must be unique amongst all guests known to the hypervisor on the connection, including those not currently active. To re-define an existing guest, use the `virsh` tool
`--memory <option>`	Memory to allocate for the guest, in MiB. (This deprecates the `-r`/`--ram` option.) Sub-options are available, like: • `maxmemory` • `hugepages`

Option	Description
	• `hotplugmemorymax` • `hotplugmemoryslots`
`--vcpus <options>`	Number of virtual cpus to configure for the guest. If 'maxvcpus' is specified, the guest will be able to hotplug up to MAX vcpus while the guest is running, but will start up with VCPUS. Use `--vcpus=?` to see a list of all available sub options.
`--cdrom <options>` `-c <options>`	File or device used as a virtual CD-ROM device. It can be path to an ISO image, or to a CDROM device. It can also be a URL from which to fetch/access a minimal boot ISO image. The URLs take the same format as described for the "--location" argument. If a cdrom has been specified via the "--disk" option, and neither "--cdrom" nor any other install option is specified, the "--disk" cdrom is used as the install media.
`--location <options>` `-l <options>`	Distribution tree installation source. virt-install can recognize certain distribution trees and fetches a bootable kernel/initrd pair to launch the install. With libvirt 0.9.4 or later, network URL installs work for remote connections. virt-install will download kernel/initrd to the local machine, and then upload the media to the remote host. This option requires the URL to be accessible by both the local and remote host. `--location` allows things like --extra-args for kernel arguments, and using --initrd-inject. If you want to use those options with CDROM media, you have a few options: • Run virt-install as root and do --location ISO • Mount the ISO at a local directory, and do --location DIRECTORY • Mount the ISO at a local directory, export that directory over local http, and do --location http://localhost/DIRECTORY The "LOCATION" can take one of the following forms: • `http://host/path` An HTTP server location containing an installable distribution image. • `ftp://host/path` An FTP server location containing an installable distribution image. • nfs:host:/path or nfs://host/path An NFS server location containing an installable distribution image. This requires running virt-install as root. • DIRECTORY Path to a local directory containing an installable distribution image. Note that the directory will not be accessible by the guest after initial boot, so the OS installer will need another way to access the rest of the install media. • ISO Mount the ISO and probe the directory. This requires running virt-install as root, and has the same VM access caveat as DIRECTORY.

Option	Description
`--import`	Skip the OS installation process, and build a guest around an existing disk image. The device used for booting is the first device specified via "`--disk`" or "`--filesystem`".
`--disk <options>`	Specifies media to use as storage for the guest, with various options. The general format of a disk string is `--disk opt1=val1,opt2=val2,...` When using multiple options, separate each option with a comma (no spaces before or after the commas). Example options: • `size` size (in GiB) to use if creating new storage example: size=10 • `path` A path to some storage media to use, existing or not. Existing media can be a file or block device. Specifying a non-existent path implies attempting to create the new storage, and will require specifying a 'size' value. Even for remote hosts, virt-install will try to use libvirt storage APIs to automatically create the given path. If the hypervisor supports it, path can also be a network URL, like http://example.com/some-disk.img . For network paths, they hypervisor will directly access the storage, nothing is downloaded locally. • `format` Disk image format. For file volumes, this can be 'raw', 'qcow2', 'vmdk', etc. See format types in https://libvirt.org/storage.html for possible values. This is often mapped to the driver_type value as well. If not specified when creating file images, this will default to `.qcow2`. If creating storage, this will be the format of the new image. If using an existing image, this overrides libvirt's format auto-detection. The disk option deprecates `-f/--file`, `-s/--file-size`, `--nonsparse`, and `--nodisks`. Use `--disk=?` to see a list of all available sub options.
`--network <options>` `-w <options>`	Connect the guest to the host network. The value for `<options>` can take one of four formats: • `bridge=BRIDGE` Connect to a bridge device in the host called "BRIDGE". Use this option if the host has static networking config and the guest requires full outbound and inbound connectivity to and from the LAN. Also use this if live migration will be used with this guest. • `network=NAME` Connect to a virtual network in the host called "NAME". You can list, create, and delete virtual networks using the "virsh" command line tool. In an unmodified install of "libvirt" there is usually a virtual network with a name of "default". Use a virtual network if the host has dynamic networking (sch as

Option	Description
	NetworkManager), or using wireless. The guest will be NATed to the LAN by whichever connection is active. • `type=direct,source=IFACE[,source_mode=MODE]` Direct connect to host interface IFACE using macvtap. • `user` Connect to the LAN using SLIRP. Only use this if running a QEMU guest as an unprivileged user. This provides a very limited form of NAT. • `none` Tell `virt-install` not to add any default network interface. Use `--network=?` to see a list of all available sub options. Complete details at https://libvirt.org/formatdomain.html#elementsNICS This option deprecates `-m/--mac`, `-b/--bridge`, and `--nonetworks`
`--noautoconsole`	This stops the system from automatically trying to connect to the guest console. The default behavior is to launch `virt-viewer` to run a GUI console or run the `virsh console` command to display a text version of the console.

Example:

This will take the iso image made in the previous file and install it into the VM platform giving the name Example_VM to the FortiGate-VM instance.

```
virt-install --connect qemu:///system --noautoconsole --name Example_VM --memory 1024 --
     vcpus 1 --import --disk fortios.qcow2,size=3 --disk fgt-logs.qcow2,size=3 --disk
     /home/username/test/fgt-bootstrap.iso,device=cdrom,bus=ide,format=raw,cache=none --
     network bridge=virbr0,model=virtio
```

The following table summarizes the options in the example:

Option	Description
`--connect qemu:///system`	Connects the image to the QEMU platform.
`--noautoconsole`	• - prevents a console from automatically starting up after the installation is completed.
`--name Example_VM`	• - sets the name of the FortiGate-VM to Example_VM.
`--memory 1024`	• - allocates 1024 MB (1 GB) of RAM to the VM.
`--vcpus 1`	• - allocates 1 virtual cpu to the VM.
`--import`	• - instead of running an installation process, builds

Option	Description
	a VM around an existing VM image based on the first instance of the --disk setting.
`--disk fortios.qcow2`	• - uses the fortios.qcow2 file to build a disk with the included content, into the VM.
`--disk fgt-logs.qcow2,size=3`	• - Because no file with the name fgt-logs.qcow2 is found, an empty disk is created in the VM with a size of 3 GB.
`--disk /home/username/test/fgt-bootstrap.iso,device=cdrom,bus=ide,format=raw,cache=none`	• - sets up a virtual cdrom drive as if it was on an IDE bus holding a virtual CD in it with no cache and the data in RAW format. This virtual CD is based on the file fgt-bootstrap.iso. While is will work if the command is run from the folder that holds the file, you can also include the path to the file.
`--network bridge=virbr0,model=virtio`	• - connects the VM to the virtual bridge virbr0 using a virtio model virtual network adapter.

Before running the command, Ensure that QEMU/KVM is running properly.

You should be able to start the instance by running the command:

```
virsh --connect qemu:///system start Example_VM
```

Deployment

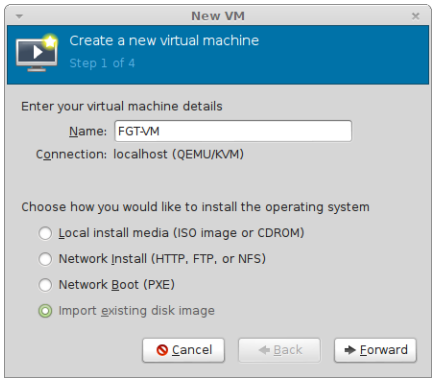
The deployment uses cases in this document describe the tasks required to deploy a FortiGate-VM virtual appliance on a KVM server. Before you deploy a virtual appliance, ensure that the requirements described in Preparing for deployment on page 7 are met and that the correct deployment package is extracted to a folder on the local computer (see Downloading the FortiGate-VM virtual appliance 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

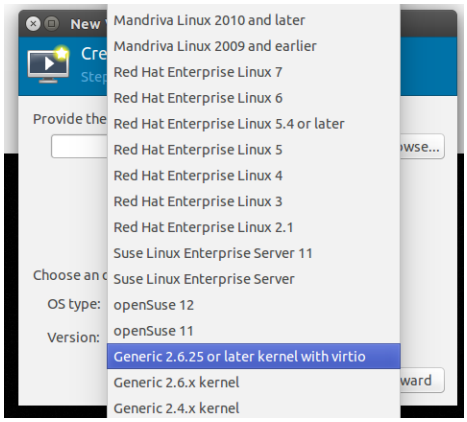
To create the FortiGate-VM virtual machine:

1. Launch Virtual Machine Manager (virt-manager) on your KVM host server. The *Virtual Machine Manager* home page opens.
2. In the toolbar, select *Create a new virtual machine*.

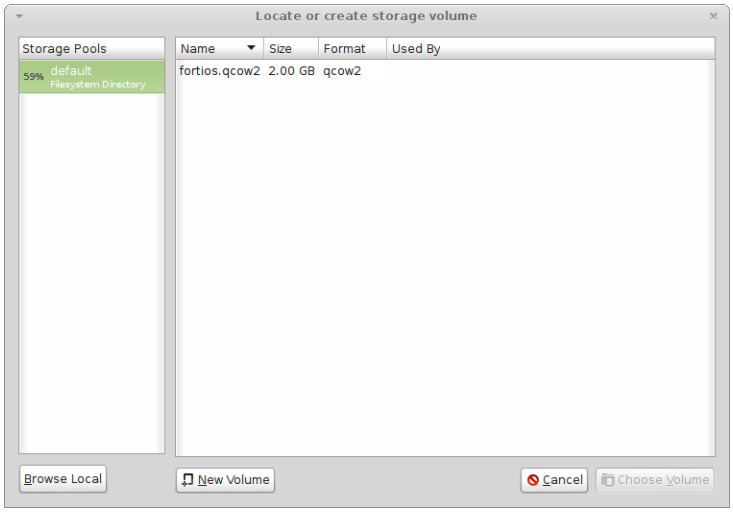


3. Enter a *Name* for the VM, FGT-VM for example.
4. Ensure that *Connection* is localhost (this is the default).
5. Select *Import existing disk image*.
6. Select *Forward*.
7. In *OS Type* select *Linux*.

8. In *Version*, select a Generic version with virtio.



9. Select *Browse*.



10. If you copied the fortios.qcow2 file to `/var/lib/libvirt/images`, it will be visible on the right. If you saved it somewhere else on your server, select Browse Local and find it.
11. Choose *Choose Volume*.
12. Select *Forward*.
13. Specify the amount of memory and number of CPUs to allocate to this virtual machine. Whether or not the amounts can exceed the license limits will depend on the FortiOS version. See FortiGate-VM virtual licenses and resources on page 5
14. Select *Forward*.
15. Expand *Advanced options*. A new virtual machine includes one network adapter by default.
16. Select a network adapter on the host computer. Optionally, set a specific MAC address for the virtual network interface.
17. Set *Virt Type* to *virtio* and *Architecture* to *qcow2*.
18. Select *Finish*.

Initial settings

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

- Connect the FortiGate-VM to the network so that it can process network traffic and maintain the validity of the license.
- Connect to the GUI of the FortiGate-VM 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 virtual appliance 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 KVM 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. Also, HTTP access must be enabled because until it is licensed the FortiGate-VM supports only low-strength encryption. HTTPS access will not work.
   For example:
   ```
   config system interface
      edit port1
         set mode static
         set ip 192.168.0.100 255.255.255.0
         append allowaccess http
      next
   end
   ```

You can also use the `append allowaccess` CLI command to enable other access protocols, such as `auto-ipsec`, `http`, `probe-response`, `radius-acct`, `snmp`, and `telnet`. The `ping`, `https`, `ssh`, and `fgfm` protocols are enabled on the `port1` interface by default.

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 (FDN) 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
```

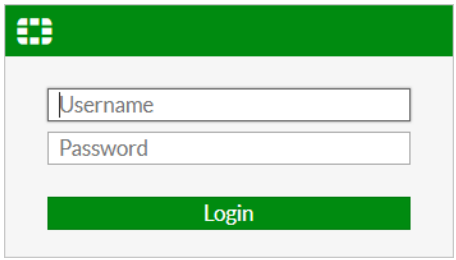
The default DNS servers are `208.91.112.53` and `208.91.112.52`.

Connect 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 19) in the location field of the browser. HTTP and/or HTTPS access and administrative access must be enabled on the interface to ensure that you can connect to the GUI. If only HTTPS access is enabled, enter "https://" before the IP address.

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

On the FortiGate-VM GUI log-in screen, enter the default username "admin" and then select *Login*. A default password is not assigned to the admin user.



Fortinet strongly 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.

Useful links:

- Administrator accounts
- Passwords and password policy
- System administrator best practices

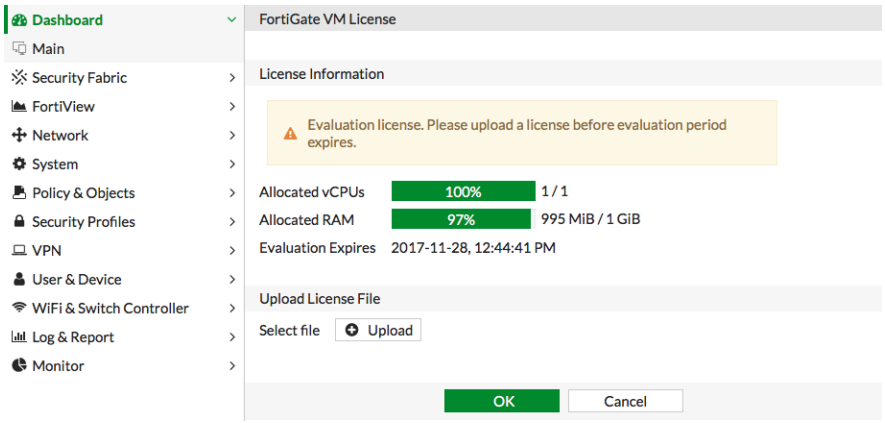
Uploading the FortiGate-VM virtual appliance license

Every Fortinet VM includes a 15-day trial license. During this time the FortiGate-VM operates in evaluation mode. Before using the FortiGate-VM you must enter the license file that you downloaded from the Customer Service & Support website upon registration.

GUI

To upload the FortiGate-VM license file:

1. There are 2 ways to get to the License upload window.
   - In the *Dashboard* > *Main* window, in the *Virtual Machine* widget, left click on the *FGVMEV* (FortiGate-VM Evaluation) *License* icon. This will reveal 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 *Licence Information* section, go to the *Virtual Machine* row and click on the link to *FortiGate-VM License*.
2. In the *Evaluation License* dialog box, 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 license is validated by the FortiGuard Distribution Network (FDN) or FortiManager for closed networks.

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 a FTP/TFTP server to apply the license.

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 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 unit has the VM activation feature, see the *Features* section of the FortiManager datasheet.

To validate your FortiGate-VM with your FortiManager:

1. To configure your FortiManager as a closed network, enter the following CLI commands 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 <IPv4 address of the FortiManager device>
   config server-list
      edit 1
         set server-type update
         set server-address <IPv4 address of the FortiManager device>
      end
   end
   set fmg-source-ip <Source IPv4 address when connecting to the FortiManager device>
   set include-default-servers disable
   set vdom <Enter the name of the VDOM to use when communicating with the FortiManager
        device>
end
```
3. Load the FortiGate-VM license 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 `get system status` command on your FortiGate-VM. The output should resemble the following:
```
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
```

Run the `diagnose hardware sysinfo vm full` command. The output should resemble the following:

```
UUID: 564db33a29519f6b1025bf8539a41e92
valid: 1
status: 1
code: 200 (If the license is a duplicate, code 401 will display)
warn: 0
copy: 0
received: 45438
warning: 0
recv: 201201201918
dup:
```

Licensing timeout

In closed environments without Internet access, you must perform offline licensing of the virtual FortiGate using a FortiManager as a license server. If the FortiGate-VM cannot perform license validation within the license timeout period, which is 30 days, the FortiGate will discard all packets, effectively ceasing operation as a firewall.

The status of the license will go through some status changes before it times out.

Status	Description
Valid	The FortiGate can connect and validate against a FortiManager or FDS
Warning	The FortiGate cannot connect and validate against a FortiManager or FDS. A check is made against how many days the Warning status has been continuous. If the number is less the 30 days the status does not change.
Invalid	The FortiGate cannot connect and validate against a FortiManager or FDS. A check is made against how many days the Warning status has been continuous. If the number is 30 days or more, the status changes to Invalid. The firewall ceases to function properly.

There is only a single log entry after the virtual FortiGate cannot access the license server for the license expiration period. This means that when you go searching the logs for a reason for the FortiGate being offline there will not be a long list of error logs that draw attention to the issue. There will only be the one entry.

Test connectivity

You can now proceed to power on your FortiGate-VM. Select the name of the FortiGate-VM in the list of virtual machines. In the toolbar, select *Console* and then select *Start*.

To test connectivity to other devices, using the PING utility is the usual method. For this, you need the console on the FortiGate-VM.

In FortiOS, the command for the PING utility is `execute ping` followed by the IP address you wish 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 it has been successfully deployed and started on the hypervisor, see the *FortiOS Handbook*, which is available online at https://docs.fortinet.com/document/fortigate/6.0.0/handbook.

High Availability

FortiGate-VM High Availability (HA) supports having two VMs in an HA cluster on the same physical platform or different platforms. The primary consideration is that all interfaces involved be able to 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 FortGate VM because it may require special settings on the host. In most cases, the unicast configuration would be preferred.

The differences between the unicast heartbeat setup the broadcast heartbeat setup 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

The unicast settings are configured in the CLI of the FortiGate-VM. The syntax is as follows:

```
config system ha
   set unicast-hb {enable/disable}
   set unicast-hb-peerip {IP address of the peer's heartbeat interface}
end
```

Setting	Description
`unicast-hb`	Enable or disable (the default) unicast HA heartbeat.
`unicast-hb-peerip`	The 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 0x8890, 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 will have the same virtual MAC addresses.

Promiscuous mode

KVM's Virtual Machine Manager does not have the ability to set a virtual network interface to promiscuous mode. This is done to the host's physical network interface. When KVM creates a VM, it also creates a tap interface as well as a new MAC address for it. Once the host's physical interface is set to promiscuous mode, it must be connected to a bridge device that is used by the VM to connect to the network outside of the host.

Because this configuration is done on the host and not the VM, the methodology depends on the host's operating system distribution and version.

Setting up the network interfaces and bridge devices requires using an account with root privileges.

Optimizing FortiGate-VM performance

The FortiGate-VM and KVM performance optimization techniques described in this section can improve the performance of your FortiGate-VM by optimizing the hardware and the KVM host environment for network- and CPU-intensive performance requirements of FortiGate-VMs.

SR-IOV

FortiGate-VMs installed on KVM 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 KVM 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 KVM 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 PF and VFs.

Setting up SR-IOV on KVM involves creating a physical functions (PF) for each physical network card in the hardware platform. Then, you create virtual functions (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 KVM host is running has BIOS, physical NIC, and network driver support for SR-IOV.

To enable SR-IOV, your KVM 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 ixgbevf or i40evf drivers. As well, the host hardware CPUs must support Second Level Address Translation (SLAT).

For optimal SR-IOV support, install the most up to date ixgbevf or i40e/i40evf network drivers. Fortinet recommends i40e/i40evf drivers because they provide four TxRx queues for each VF and ixgbevf only provides two TxRx queues.

Enable SR-IOV support for Intel systems

Use the following steps to enable SR-IOV support for KVM host systems that use Intel CPUs. These steps involve enabling and verifying Intel VT-d specifications in the BIOS and Linux kernel. You can skip these steps if VT-d is already enabled.

On an Intel host PC, Intel VT-d BIOS settings provide hardware support for directly assigning a physical device to a virtual machine.

1. View the BIOS settings of the host machine and enable VT-d settings if they are not already enabled. You may have to review the manufacturer's documentation for details.
2. Activate Intel VT-d in the Linux kernel by adding the `intel_iommu=on` parameter to the kernel line in the `/boot/grub/grub.conf` file. For example:

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-330.x86_645)
   root (hd0,0)
   kernel /vmlinuz-2.6.32-330.x86_64 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
        intel_iommu=on
   initrd /initrd-2.6.32-330.x86_64.img
```

3. Restart the system.

Enable SR-IOV support for AMD systems

Use the following steps to enable SR-IOV support for KVM host systems that use AMD CPUs. These steps involve enabling the AMD IOMMU specifications in the BIOS and Linux kernel. You can skip these steps if AMD IOMMU is already enabled.

On an AMD host PC, IOMMU BIOS settings provide hardware support for directly assigning a physical device to a virtual machine.

1. View the BIOS settings of the host machine and enable IOMMU settings if they are not already enabled. You may have to review the manufacturer's documentation for details.
2. Append `amd_iommu=on` to the kernel command line in `/boot/grub/grub.conf` so that AMD IOMMU specifications are enabled when the system starts up.
3. Restart the system.

Verify that Linux and KVM can find SR-IOV-enabled PCI devices

You can use the `lspci` command to view the list of PCI devices and verify that your SR-IOV supporting network cards are on the list. The following output example shows some example entries for the Intel 82576 network card:

```
# lspci
03:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
03:00.1 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
```

Optionally modify the SR-IOV kernel modules

If the device is supported the driver kernel module should be loaded automatically by the kernel. You can enable optional parameters using the `modprobe` command. For example, the Intel 82576 network interface card uses the `igb` driver kernel module.

```
# modprobe igb [<option>=<VAL1>,<VAL2>,]
   # lsmod |grep igb
   igb    87592  0
   dca     6708    1 igb
```

Attaching an SR-IOV network device to a FortiGate-VM

You can enable SR-IOV for a FortiGate-VM by creating a Virtual Function (VF) and then attaching the VF to your FortiGate-VM.

Activate and verify an SR-IOV VF

The `max_vfs` parameter of the `igb` module allocates the maximum number of Virtual Functions (VFs). The `max_vfs` parameter causes the driver to spawn multiple VFs.

Before activating the maximum number of VFs enter the following command to remove the `igb` module:

```
# modprobe -r igb
```

Restart the `igb` module with `max_vfs` set to the maximum supported by your device. For example, the valid range for the Intel 82576 network interface card is 0 to 7. To activate the maximum number of VFs supported by this device enter:

```
# modprobe igb max_vfs=7
```

Make the VFs persistent by adding `options igb max_vfs=7` to any file in `/etc/modprobe.d`. For example:

```
# echo "options igb max_vfs=7" >>/etc/modprobe.d/igb.conf
```

Verify the new VFs. For example, you could use the following `lspci` command to list the newly added VFs attached to the Intel 82576 network device. Alternatively, you can use `grep` to search for `Virtual Function`, to search for devices that support VFs.

```
# lspci | grep 82576
0b:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
0b:00.1 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection(rev 01)
0b:10.0 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.1 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.2 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.3 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.4 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.5 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.6 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:10.7 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.0 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.1 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.2 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.3 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.4 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
0b:11.5 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
```

Use the `-n` parameter of the `lspci` command to find the identifier for the PCI device. The PFs correspond to `0b:00.0` and `0b:00.1`. All VFs have `Virtual Function` in the description.

Verify that the devices exist with virsh

The `libvirt` service must recognize a PCI device before you can add it to a virtual machine. `libvirt` uses a similar notation to the `lspci` output.

Use the `virsh nodedev-list` command and the `grep` command to filter the Intel 82576 network device from the list of available host devices. In the example, `0b` is the filter for the Intel 82576 network devices. This may vary for your system and may result in additional devices.

```
# virsh nodedev-list | grep 0b
pci_0000_0b_00_0
pci_0000_0b_00_1
pci_0000_0b_10_0
pci_0000_0b_10_1
pci_0000_0b_10_2
pci_0000_0b_10_3
pci_0000_0b_10_4
pci_0000_0b_10_5
pci_0000_0b_10_6
pci_0000_0b_11_7
pci_0000_0b_11_1
pci_0000_0b_11_2
pci_0000_0b_11_3
pci_0000_0b_11_4
pci_0000_0b_11_5
```

The serial numbers for the Virtual Functions and Physical Functions should be in the list.

Get device details with virsh

The `pci_0000_0b_00_0` is one of the PFs and `pci_0000_0b_10_0` is the first corresponding VF for that PF. Use `virsh nodedev-dumpxml` to get device details for both devices.

Example device details for the `pci_0000_0b_00_0` PF device:

```
 # virsh nodedev-dumpxml pci_0000_0b_00_0
<device>
   <name>pci_0000_0b_00_0</name>
   <parent>pci_0000_00_01_0</parent>
   <driver>
      <name>igb</name>
   </driver>
   <capability type='pci'>
      <domain>0</domain>
      <bus>11</bus>
      <slot>0</slot>
      <function>0</function>
      <product id='0x10c9'>82576 Gigabit Network Connection</product>
      <vendor id='0x8086'>Intel Corporation</vendor>
   </capability>
</device>
```

Example device details for the `pci_0000_0b_10_0` PF device:

```
# virsh nodedev-dumpxml pci_0000_0b_10_0
<device>
   <name>pci_0000_0b_10_0</name>
   <parent>pci_0000_00_01_0</parent>
   <driver>
      <name>igbvf</name>
   </driver>
   <capability type='pci'>
      <domain>0</domain>
      <bus>11</bus>
      <slot>16</slot>
      <function>0</function>
      <product id='0x10ca'>82576 Virtual Function</product>
```

```
        <vendor id='0x8086'>Intel Corporation</vendor>
      </capability>
    </device>
```

You must use this information to specify the bus, slot, and function parameters when you add the VF to a FortiGate-VM. A convienent way to do this is to create a temporary xml file and copy the following text into that file.

```
<interface type='hostdev' managed='yes'>
     <source>
       <address type='pci' domain='0' bus='11' slot='16' function='0'/>
     </source>
   </interface>
```

You can also include additional information about the VF such as a MAC address, VLAN tag, and so on. If you specify a MAC address, the VF will always have this MAC address. If you do not specify a MAC address, the system generates a new one each time the FortiGate-VM restarts.

Add the VF to a FortiGate-VM

Enter the following command to add the VF to a FortiGate-VM. This configuration attaches the new VF device immediately and saves it for subsequent FortiGate-VM restarts.

```
virsh attach-device MyFGTVM <temp-xml-file> --config
```

Where *MyFGTVM* is the name of the FortiGate-VM for which to enable SR-IOV, and `<temp-xml-file>` is the temporary XML file containing the VF configuration.

After this configuration, when you start up the FortiGate-VM it detects the SR-IOV VF as a new network interface.

Interrupt affinity

In addition to enabling SR-IOV in the VM host, to fully take advantage of SR-IOV performance improvements you need to 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 would be 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 of the 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 would 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 re-distribution 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 of the 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 re-distributed 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
```

Change log

Date	Change Description
2018-09-24	Initial release.
2019-07-12	Updated userdata.txt to user_data.
2019-07-22	Updated Validating the FortiGate-VM license with FortiManager on page 22.
2019-09-25	Updated FortiGate-VM models and licensing on page 4.

Copyright© 2021 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., in the U.S. and other jurisdictions, and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. In no event does Fortinet make any commitment related to future deliverables, features or development, and circumstances may change such that any forward-looking statements herein are not accurate. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.