Deploy the Threat Defense Virtual on KVM

This chapter describes the procedures to deploy the threat defense virtual to a KVM environment.

Overview

KVM is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (such as Intel VT). It consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, such as kvm-intel.ko.

You can run multiple virtual machines running unmodified OS images using KVM. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, and so forth.

Performance Tiers for Threat Defense Virtual Smart Licensing

The threat defense virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Table 1: Threat Defense Virtual Licensed Feature Limits Based on Entitlement

Performance Tier	Device Specifications (Core/RAM)	Rate Limit	RA VPN Session Limit
FTDv5, 100Mbps	4 core/8 GB	100Mbps	50
FTDv10, 1Gbps	4 core/8 GB	1Gbps	250
FTDv20, 3Gbps	4 core/8 GB	3Gbps	250
FTDv30, 5Gbps	8 core/16 GB	5Gbps	250

Performance Tier	Device Specifications (Core/RAM)	Rate Limit	RA VPN Session Limit
FTDv50, 10Gbps	12 core/24 GB	10Gbps	750
FTDv100, 16Gbps	16 core/32 GB	16Gbps	10,000

See the "Licensing the System" chapter in the *Firepower Management Center Configuration* for guidelines when licensing your threat defense virtual device.

System Requirements

See the Cisco Firepower Compatibility Guide for the most current information about hypervisor support for the threat defense virtual.

The specific hardware used for the threat defense virtual deployments can vary, depending on the number of instances deployed and usage requirements. Each instance of the threat defense virtual requires a minimum resource allocation—amount of memory, number of CPUs, and disk space—on the server.

Table 2: Threat Defense Virtual Appliance Resource Requirements

Settings	Value
Performance Tiers	**Version 7.0 and later** The threat defense virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements. • FTDv5 4vCPU/8GB (100Mbps) • FTDv10 4vCPU/8GB (1Gbps) • FTDv20 4vCPU/8GB (3Gbps) • FTDv30 8vCPU/16GB (5Gbps) • FTDv50 12vCPU/24GB (10Gbps) • FTDv100 16vCPU/32GB (16Gbps) See the "Licensing the System" chapter in the *Firepower Management Center Configuration* for guidelines when licensing your threat defense virtual device. **Note** To change the vCPU/memory values, you must first power off the threat defense virtual device.

Settings	Value
Number of cores and memory	**Version 6.4 to Version 6.7** The threat defense virtual deploys with adjustable vCPU and memory resources. There are three supported vCPU/memory pair values: • 4vCPU/8GB (default) • 8vCPU/16GB • 12vCPU/24GB **Note** To change the vCPU/memory values, you must first power off the threat defense virtual device. Only the above three combinations are supported.
	Version 6.3 and earlier The threat defense virtual deploys with fixed vCPU and memory resources. There is only one supported vCPU/memory pair value: • 4vCPU/8GB **Note** Adjustments to vCPUs and memory are not supported.
Hard disk provisioned size	• 50 GB • Adjustable setting. Supports virtio block devices
vNICs	The threat defense virtual on KVM supports the following virtual network adapters: • **VIRTIO**—Virtio is the main platform for IO virtualization in KVM and provides a common framework for hypervisors for IO virtualization. The host implementation is in userspace - qemu, so no driver is needed in the host. • **IXGBE-VF**—The ixgbe-vf (10 Gbit/s) driver supports virtual function devices that can only be activated on kernels that support SR-IOV. SR-IOV requires the correct platform and OS support; see Support for SR-IOV for more information.

Guidelines and Limitations

- Requires two management interfaces and two data interfaces to boot.

Note The threat defense virtual default configuration puts the management interface, diagnostic interface, and inside interface on the same subnet.

- Supports virtIO drivers.
- Supports ixgbe-vf drivers for SR-IOV.
- Supports a total of 10 interfaces

- The default configuration for the threat defense virtual assumes that you put both the management (management and diagnostic) and inside interfaces on the **same subnet**, and the management address uses the inside address as its gateway to the Internet (going through the outside interface).
- The threat defense virtual must be powered up on firstboot with at least four interfaces. Your system will not deploy without four interfaces
- The threat defense virtual supports a total of 10 interfaces—1 management interface, 1 diagnostic interface, and a maximum of 8 network interfaces for data traffic. The interface-to-network assignments must be ordered as follows:
  - Management interface (1) (required)

Note You can optionally configure a data interface for the management center management instead of the Management interface. The Management interface is a pre-requisite for data interface management, so you still need to configure it in your initial setup. Note that the management center access from a data interface is not supported in High Availability deployments. For more information about configuring a data interface for the management center access, see the **configure network management-data-interface** command in the FTD command reference.

  - Diagnostic interface (2) (required)
  - Outside interface (3) (required)
  - Inside interface (4) (required)
  - Data interfaces (5-10) (optional)

See the following concordance of Network Adapter, Source Networks and Destination Networks for the threat defense virtual interfaces:

Table 3: Source to Destination Network Mapping

Network Adapter	Source Network	Destination Network	Function
vnic0*	Management0-0	Management0/0	Management
vnic1*	Diagnostic0-0	Diagnostic0/0	Diagnostic
vnic2	GigabitEthernet0-0	GigabitEthernet0/0	Outside
vnic3*	GigabitEthernet0-1	GigabitEthernet0/1	Inside
*Important. Attach to the same subnet.			

- Cloning a virtual machine is not supported.

- For console access, supports terminal server via telnet.
- For creating vNICs with IPv6 support configuration on KVM, you must create an XML file for each interface that consists of IPv6 configuration parameters. You can install vNICs with the IPV6 network protocol configurations by running these XML files using the command **virsh net-create** <<*interface configuration XML file name*>>.

  For each interface, you can create the following XML file:

  - Management interface - *mgmt-vnic.xml*
  - Diagnostic interface - *diag-vnic.xml*
  - Inside interface - *inside-vnic.xml*
  - Outside interface - *outside-vnic.xml*

  Example:

  To create an XML file for Management interface with IPv6 configuration.

```
<network>
            <name>mgmt-vnic</name>
            <bridge name='mgmt-vnic' stp='on' delay='0' />
            <ip family='ipv6' address='2001:db8::a111:b220:0:abcd' prefix='96'/>
        </network>
```

  Similarly, you must create XML file for other interfaces.

  You can verify the virtual network adapters installed on KVM by running the following command.

```
virsh net-list
         brctl show
```

CPU Mode

KVM can emulate a number different of CPU types. For your VM, you typically should select a processor type which closely matches the CPU of the host system, as it means that the host CPU features (also called CPU flags) will be available in your VMs. You should set the CPU type to **host** in which case the VM will have exactly the same CPU flags as your host system.

Clustering

Clustering is supported on threat defense virtual instances deployed on KVM. See Clustering for Threat Defense Virtual in a Private Cloud for more information.

Performance Optimizations

To achieve the best performance out of the threat defense virtual, you can make adjustments to the both the VM and the host. See Virtualization Tuning and Optimization on KVM for more information.

Receive Side Scaling—The threat defense virtual supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic to multiple processor cores. Supported on Version 7.0 and later. See Multiple RX Queues for Receive Side Scaling (RSS) for more information.

Support for SR-IOV

SR-IOV Virtual Functions require specific system resources. A server that supports SR-IOV is required in addition to an SR-IOV capable PCIe adapter. You must be aware of the following hardware considerations:

- The capabilities of SR-IOV NICs, including the number of VFs available, differ across vendors and devices. The following NICs are supported:
  - Intel Ethernet Server Adapter X710
  - Intel Ethernet Server Adapter X520 - DA2
  - Intel Ethernet Network Adapter E810-CQDA2
    - The firmware (NVM image) and network driver is updated on Intel® Network Adapter E810 using an NVM utility tool. The Non-Volatile Memory (NVM) image and network driver are a compatible set of components that you update as a combination on the Intel® Network Adapter E810. For information on NVM and Software compatibility matrix, refer to the Intel® Ethernet Controller E810 Datasheet to update the correct firmware drivers on Intel® Network Adapter E810.

- Not all PCIe slots support SR-IOV.
- SR-IOV-capable PCIe slots may have different capabilities.
- x86_64 multicore CPU — Intel Sandy Bridge or later (Recommended).

Note We tested the threat defense virtual on Intel's Broadwell CPU (E5-2699-v4) at 2.3GHz.

- Cores
  - Minimum of 8 physical cores per CPU socket.
  - The 8 cores must be on a single socket.

Note CPU pinning is recommended to achieve full throughput.

- You should consult your manufacturer's documentation for SR-IOV support on your system. For KVM, you can verify CPU compatibility for SR-IOV support. Note that for the threat defense virtual on KVM we only support x86 hardware.

Limitations of using ixgbe-vf Interfaces

Be aware of the following limitations when using ixgbe-vf interfaces:

- The guest VM is not allowed to set the VF to promiscuous mode. Because of this, transparent mode is not supported when using ixgbe-vf.
- The guest VM is not allowed to set the MAC address on the VF. Because of this, the MAC address is not transferred during HA like it is done on other threat defense virtual platforms and with other interface types. HA failover works by transferring the IP address from active to standby.

Note This limitation is applicable to the i40e-vf interfaces too.

- The Cisco UCS-B server does not support the ixgbe-vf vNIC.
- In a failover setup, when a paired threat defense virtual (primary unit) fails, the standby threat defense virtual unit takes over as the primary unit role and its interface IP address is updated with a new MAC address of the standby threat defense virtual unit. Thereafter, the threat defense virtual sends a gratuitous Address Resolution Protocol (ARP) update to announce the change in MAC address of the interface IP address to other devices on the same network. However, due to incompatibility with these types of interfaces, the gratuitous ARP update is not sent to the global IP address that is defined in the NAT or PAT statements for translating the interface IP address to global IP addresses.

Snort

- If you are observing abnormal behavior such as Snort taking a long time to shut down, or the VM being slow in general or when a certain process is executed, collect logs from the threat defense virtual and the VM host. Collection of overall CPU usage, memory, I/O usage, and read/write speed logs will help troubleshoot the issues.
- High CPU and I/O usage is observed when Snort is shutting down. If a number of threat defense virtual instances have been created on a single host with insufficient memory and no dedicated CPU, Snort will take a long time to shut down which will result in the creation of Snort cores.

How to Manage Secure Firewall Threat Defense Virtual Device

You have two options to manage your Secure Firewall Threat Defense Virtual device.

Secure Firewall Management Center

If you are managing large numbers of devices, or if you want to use the more complex features and configurations that the threat defense allows, use the management center to configure your devices instead of the integrated device manager.

Important You cannot use both the device manager and the management center to manage the threat defense device. Once the device manager integrated management is enabled, it won't be possible to use the management center to manage the threat defense device, unless you disable the local management and re-configure the management to use the management center. On the other hand, when you register the threat defense device to the management center, the device manager onboard management service is disabled.

Caution Currently, Cisco does not have an option to migrate your device manager configuration to the management center and vice-versa. Take this into consideration when you choose what type of management you configure for the threat defense device.

Secure Firewall device manager

The device manager is an onboard integrated manager.

The device manager is a web-based configuration interface included on some of the threat defense devices. The device manager lets you configure the basic features of the software that are most commonly used for small networks. It is especially designed for networks that include a single device or just a few, where you do not want to use a high-powered multiple-device manager to control a large network containing many of the threat defense devices.

Note See the Cisco Secure Firewall Device Manager Configuration Guide for list of the threat defense devices that support the device manager.

Prerequisites

- Download the threat defense virtual qcow2 file from Cisco.com and put it on your Linux host:

  https://software.cisco.com/download/navigator.html

Note A Cisco.com login and Cisco service contract are required.

- For the purpose of the sample deployment in this document, we are assuming you are using Ubuntu 18.04 LTS. Install the following packages on top of the Ubuntu 18.04 LTS host:
  - qemu-kvm
  - libvirt-bin
  - bridge-utils
  - virt-manager
  - virtinst
  - virsh tools
  - genisoimage
- Performance is affected by the host and its configuration. You can maximize the throughput of the threat defense virtual on KVM by tuning your host. For generic host-tuning concepts, see Network Function Virtualization: Quality of Service in Broadband Remote Access Servers with Linux and Intel Architecture.
- Useful optimizations for Ubuntu 18.04 LTS include the following:
  - macvtap—High performance Linux bridge; you can use macvtap instead of a Linux bridge. You must configure specific settings to use macvtap instead of the Linux bridge.
  - Transparent Huge Pages—Increases memory page size and is on by default in Ubuntu 18.04.
  - Hyperthread disabled—Reduces two vCPUs to one single core.

- txqueuelength—Increases the default txqueuelength to 4000 packets and reduces drop rate.
- pinning—Pins qemu and vhost processes to specific CPU cores; under certain conditions, pinning is a significant boost to performance.

- For information on optimizing a RHEL-based distribution, see Red Hat Enterprise Linux6 Virtualization Tuning and Optimization Guide.
- For KVM and System compatibility, see Cisco Firepower Threat Defense Virtual Compatibility.
- You can use the following methods to verify whether your virtual machine is running the KVM:
  - Run the **lsmod** to list the modules in the Linux Kernel. If the KVM is running, it is indicated by displaying the following output:

    root@kvm-host:~$ lsmod | grep kvm

    kvm_intel 123675 0

    kvm 257361 1 kvm_intel

- If **ls -l /dev/kvm** does not exist on the target VM then you are probably running **qemu**, and not taking advantage of the KVM hardware assist features.

  root@kvm-host:~$ ls -l /dev/kvm

  crw------- 1 root root 10, 232 Mar 23 13:53 /dev/kvm

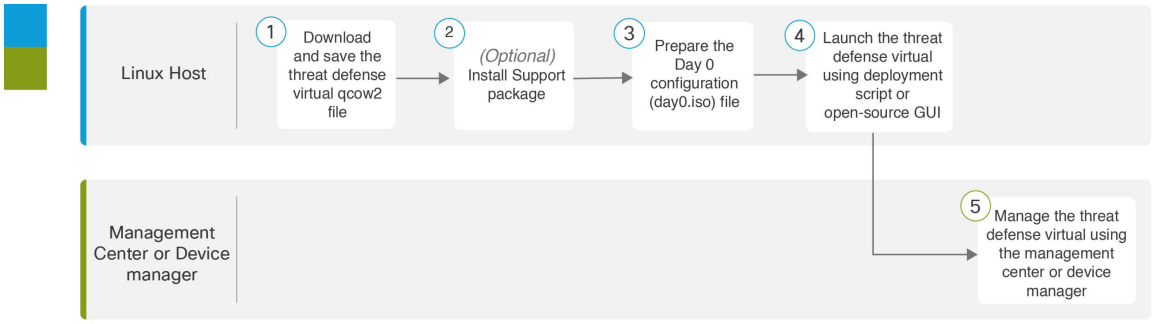
- Run the following command to also check whether the host machine supports KVM:

  root@kvm-host:~$ sudo kvm-ok

- You can also use KVM acceleration.

End-to-End Procedure

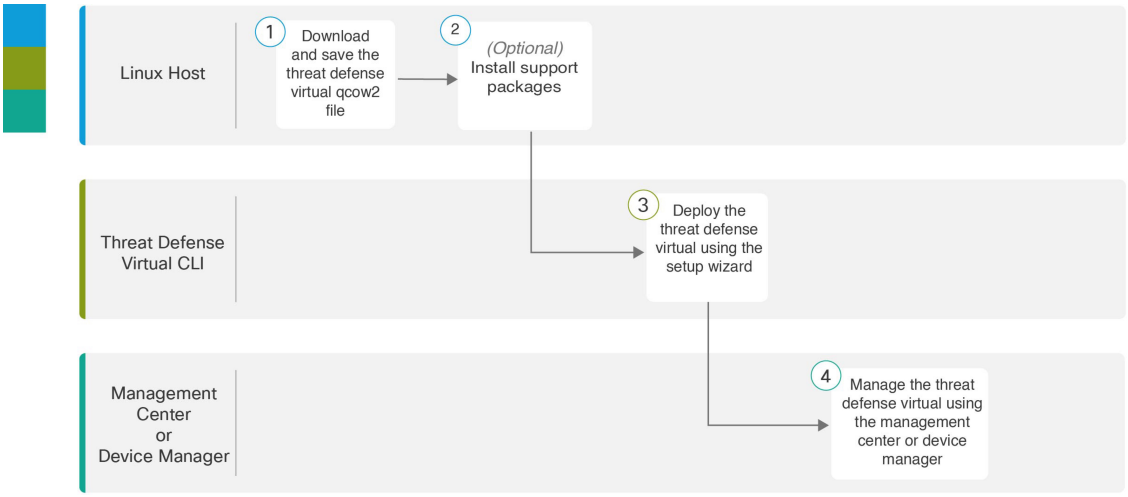
The following flowchart illustrates the workflow for deploying threat defense virtual on a KVM instance using a Day 0 configuration file.



	Workspace	Steps
1	Linux Host	Prerequisites, on page 8: Download and save the threat defense virtual qcow2 file on the Linux host.

	Workspace	Steps
2	Linux Host	Prerequisites, on page 8: Install support packages.
3	Linux Host	Prepare the Day 0 Configuration File
4	Linux Host	Launch the threat defense virtual: • Launch Using a Deployment Script • Launch Using a Graphical User Interface (GUI)
5	Management Center	Manage the threat defense virtual by using the Management Center.

The following flowchart illustrates the workflow for deploying the threat defense virtual on a KVM instance without using a Day 0 configuration file.



	Workspace	Steps
1	Linux Host	Prerequisites, on page 8: Download and save the threat defense virtual qcow2 file on the Linux host.
2	Linux Host	Prerequisites, on page 8: Install support packages.
3	Threat Defense Virtual CLI	Launch Without the Day 0 Configuration File: Deploy the threat defense virtual using the setup wizard.
4	Management Center	Manage the threat defense virtual by using the Management Center

Prepare the Day 0 Configuration File

You can prepare a Day 0 configuration file before you launch the threat defense virtual. This file is a text file that contains the initial configuration data that gets applied at the time a virtual machine is deployed. This initial configuration is placed into a text file named “day0-config” in a working directory you choose, and is manipulated into a day0.iso file that is mounted and read on first boot.

Important The day0.iso file must be available during first boot.

If you deploy with a Day 0 configuration file, the process allows you to perform the entire initial setup for the threat defense virtual appliance. You can specify:

- The End User License Agreement (EULA) acceptance.
- A host name for the system.
- A new administrator password for the admin account.
- The management mode; see How to Manage Secure Firewall Threat Defense Virtual Device.

  You can either set **ManageLocally** to **Yes**, or enter information for the management center fields (**FmcIp**, **FmcRegKey**, and **FmcNatId**). Leave fields empty for the management mode you are not using.
- The initial firewall mode; sets the initial firewall mode, either **routed** or **transparent**.

  If you plan to manage your deployment using the local device manager, you can only enter **routed** for firewall mode. You cannot configure transparent firewall mode interfaces using the device manager.
- Network settings that allow the appliance to communicate on your management network.
- The deployment type where you can specify whether you are deploying threat defense virtual in cluster or standalone mode.

If you deploy without a Day 0 configuration file, you must configure System-required settings after launch; see Launch Without the Day 0 Configuration File, on page 17 for more information.

Note We are using Linux in this example, but there are similar utilities for Windows.

SUMMARY STEPS

1. Enter the CLI configuration for the threat defense virtual in a text file called “day0-config”. Add network settings and information about managing the management center.
2. Generate the virtual CD-ROM by converting the text file to an ISO file:
3. Repeat to create unique default configuration files for each of the device manager you want to deploy.

DETAILED STEPS

Step 1 Enter the CLI configuration for the threat defense virtual in a text file called “day0-config”. Add network settings and information about managing the management center.

Example:

```
#Firepower Threat Defense
{
    "EULA": "accept",
    "Hostname": "ftdv-production",
    "AdminPassword": "r2M$9^Uk69##",
    "FirewallMode": "routed",
    "DNS1": "1.1.1.1",
    "DNS2": "1.1.1.2",
    "DNS3": "",
    "IPv4Mode": "manual",
    "IPv4Addr": "10.12.129.44",
    "IPv4Mask": "255.255.0.0",
    "IPv4Gw": "10.12.0.1",
    "IPv6Mode": "enabled",
    "IPv6Addr": "2001:db8::a111:b221:1:abca/96",
    "IPv6Mask": "",
    "IPv6Gw": "",
    "FmcIp": "",
    "FmcRegKey": "",
    "FmcNatId": "",
    "ManageLocally":"No"
}
```

Enter **Yes** for **ManageLocally** in your Day 0 configuration file to use the local device manager; or enter the management center fields (**FmcIp**, **FmcRegKey**, and **FmcNatId**). For the management option you aren’t using, leave those fields blank.

Step 2 Generate the virtual CD-ROM by converting the text file to an ISO file:

Example:

```
/usr/bin/genisoimage -r -o day0.iso day0-config
```

or

Example:

```
/usr/bin/mkisofs -r -o day0.iso day0-config
```

Step 3 Repeat to create unique default configuration files for each of the device manager you want to deploy.

What to do next

- If using virt-install, add the following line to the virt-install command:

  ```
  --disk path=/home/user/day0.iso,format=iso,device=cdrom \
  ```

- If using virt-manager, you can create a virtual CD-ROM using the virt-manager GUI; see Launch Using a Graphical User Interface (GUI), on page 15.

Launch the Threat Defense Virtual

Launch Using a Deployment Script

Use a virt-install based deployment script to launch the threat defense virtual.

Be aware that you can optimize performance by selecting the best guest caching mode for your environment. The cache mode in use will affect whether data loss occurs, and the cache mode can also affect disk performance.

Each KVM guest disk interface can have one of the following cache modes specified: *writethrough*, *writeback*, *none*, *directsync*, or *unsafe*. *writethrough* provides read caching. *writeback* provides read and write caching. *directsync* bypasses the host page cache. *unsafe* may cache all content and ignore flush requests from the guest.

- A *cache=writethrough* will help reduce file corruption on KVM guest machines when the host experiences abrupt losses of power. We recommend that you use *writethrough* mode.
- However, *cache=writethrough* can also affect disk performance due to more disk I/O writes than *cache=none*.
- If you remove the cache parameter on the *--disk* option, the default is *writethrough*.
- Not specifying a cache option may also significantly reduce the time required for the VM creation. This is due to the fact that some older RAID controllers have poor disk caching capability. Hence, disabling disk caching (*cache=none*) and thus defaulting to *writethrough*, helps ensure data integrity.
- Starting with version 6.4, the threat defense virtual deploys with adjustable vCPU and memory resources. Prior to version 6.4, the threat defense virtual deployed as a fixed configuration 4vCPU/8GB device. See the following table for supported values for the --vcpus and --ram parameters for each threat defense virtual platform size.

Table 4: Supported vCPU and Memory Parameters for virt-install

--vcpus	--ram	Threat Defense Virtual Platform Size
4	8192	4vCPU/8GB (default)
8	16384	8vCPU/16GB
12	24576	12vCPU/24GB

Step 1 Create a virt-install script called “virt_install_ftdv.sh”.

The name of the threat defense virtual VM must be unique across all other virtual machines (VMs) on this KVM host. The threat defense virtual can support up to 10 network interfaces. This example uses four interfaces. The virtual NIC must be VirtIO.

Note The default configuration for the threat defense virtual assumes that you put the management interface, diagnostic interface, and inside interface on the **same subnet**. The system requires at least 4 interfaces to successfully boot up. The interface-to-network assignments must be ordered as follows:

- (1) Management interface (required)
- (2) Diagnostic interface (required)
- (3) Outside interface (required)
- (4) Inside interface (required)
- (5) (Optional) Data interfaces—up to 6

Example:

```
virt-install \
   --connect=qemu:///system \
   --network network=default,model=virtio \
   --network network=default,model=virtio \
   --network network=default,model=virtio \
   --network network=default,model=virtio \
   --name=ftdv \
   --arch=x86_64 \
   --cpu host \
   --vcpus=8 \
   --ram=16384 \
   --os-type=linux \
   --os-variant=generic26 \
   --virt-type=kvm \
   --import \
   --watchdog i6300esb,action=reset \
   --disk path=<ftd_filename>.qcow2,format=qcow2,device=disk,bus=virtio,cache=none \
   --disk path=<day0_filename>.iso,format=iso,device=cdrom \
   --console pty,target_type=serial \
   --serial tcp,host=127.0.0.1:<port>,mode=bind,protocol=telnet \
   --force
```

Step 2 Run the virt_install script:

Example:

```
/usr/bin/virt_install_ftdv.sh

Starting install...
Creating domain...
```

A window appears displaying the console of the VM. You can see that the VM is booting. It takes a few minutes for the VM to boot. Once the VM stops booting, you can issue CLI commands from the console screen.

What to do next

Your next steps depend on what management mode you chose.

- If you chose **No** for **Manage Locally**, you'll use the management center to manage your threat defense virtual; see Managing the Secure Firewall Threat Defense Virtual with the Secure Firewall Management Center.

See How to Manage Secure Firewall Threat Defense Virtual Device for an overview of how to choose your management option.

Launch Using a Graphical User Interface (GUI)

There are a several open-source options available to manage KVM virtual machines using a GUI. The following procedure uses virt-manager, also known as Virtual Machine Manager, to launch the threat defense virtual. virt-manager is a graphical tool for creating and managing guest virtual machines.

Note KVM can emulate a number different of CPU types. For your VM, you typically should select a processor type which closely matches the CPU of the host system, as it means that the host CPU features (also called CPU flags) will be available in your VMs. You should set the CPU type to **host** in which case the VM will have exactly the same CPU flags as your host system.

Step 1 Start virt-manager **(Applications** > **System Tools** > **Virtual Machine Manager**).

You may be asked to select the hypervisor and/or enter your root password.

Step 2 Click the button in the top left corner to open the **New VM** wizard.

Step 3 Enter the virtual machine details:

a) For the operating system, select **Import existing disk image**.

This method allows you to import a disk image (containing a pre-installed, bootable operating system) to it.

b) Click **Forward** to continue.

Step 4 Load the disk image:

a) Click **Browse...** to select the image file.
b) Choose *Generic* for the **OS type**.
c) Click **Forward** to continue.

Step 5 Configure the memory and CPU options:

Important The threat defense virtual supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

See the following table for supported performance tiers and values for the --vcpus and --ram parameters for each the threat defense virtual platform.

Table 5: Supported vCPU and Memory Parameters for Virtual Machine Manager

CPUs	Memory	Threat Defense Virtual Platform Size
4	8192	4vCPU/8GB (default)
8	16384	8vCPU/16GB
12	24576	12vCPU/24GB

a) Set the **Memory (RAM)** parameter for your threat defense virtual platform size.
b) Set the corresponding **CPUs** parameter for the threat defense virtual platform size.
c) Click **Forward** to continue.

Step 6 Check the **Customize configuration before install** box, specify a **Name**, then click **Finish**.

Doing so opens another wizard that allows you to add, remove, and configure the virtual machine's hardware settings.

Step 7 Modify the CPU configuration:

From the left panel, select **Processor**, then select **Configuration** > **Copy host CPU configuration**.

This applies the physical host's CPU model and configuration to your VM.

Step 8 Configure the Virtual Disk:

a) From the left panel, select **Disk 1**.
b) Select **Advanced options**.
c) Set the **Disk bus** to *Virtio*.
d) Set the **Storage format** to *qcow2*.

Step 9 Configure a serial console:

a) From the left panel, select **Console**.
b) Select **Remove** to remove the default console.
c) Click **Add Hardware** to add a serial device.
d) For **Device Type**, select *TCP net console (tcp)*.
e) For **Mode**, select *Server mode (bind)*.
f) For **Host**, enter **0.0.0.0** for the IP address, then enter a unique **Port** number.
g) Check the **Use Telnet** box.
h) Configure device parameters.

Step 10 Configure a watchdog device to automatically trigger some action when the KVM guest hangs or crashes:

a) Click **Add Hardware** to add a watchdog device.
b) For **Model**, select *default*.
c) For **Action**, select *Forcefully reset the guest*.

Step 11 Configure at least 4 virtual network interfaces.

Click **Add Hardware** to add an interface, then choose **macvtap** or specify a shared device name (use a bridge name).

Note The threat defense virtual on KVM supports a total of 10 interfaces—1 management interface, 1 diagnostic interface, and a maximum of 8 network interfaces for data traffic. The interface-to-network assignments must be ordered as follows:

vnic0—Management interface (required)

vnic1—Diagnostic interface (required)

vnic2— Outside interface (required)

vnic3—Inside interface (required)

vnic4-9—Data interfaces (optional)

Important Make sure vnic0, vnic1, and vnic3 are mapped to the same subnet.

Step 12 If deploying using a Day 0 configuration file, create a virtual CD-ROM for the ISO:

a) Click **Add Hardware**.
b) Select **Storage**.
c) Click **Select managed or other existing storage** and browse to the location of the ISO file.
d) For **Device type**, select *IDE CDROM*.

Step 13 After configuring the virtual machine's hardware, click **Apply**.

Step 14 Click **Begin installation** for virt-manager to create the virtual machine with your specified hardware settings.

What to do next

Your next steps depend on what management mode you chose.

- If you chose **No** for **ManageLocally**, you'll use the management center to manage your threat defense virtual; see Managing the Secure Firewall Threat Defense Virtual with the Secure Firewall Management Center.

See How to Manage Secure Firewall Threat Defense Virtual Devicefor an overview of how to choose your management option.

Launch Without the Day 0 Configuration File

Because the threat defense virtual appliances do not have web interfaces, you must set up a virtual device using the CLI if you deployed without a Day 0 configuration file.

When you first log into a newly deployed device, you must read and accept the EULA. Then, follow the setup prompts to change the administrator password, and configure the device's network settings and firewall mode.

When following the setup prompts, for multiple-choice questions, your options are listed in parentheses, such as (*y/n*). Defaults are listed in square brackets, such as [*y*]. Press Enter to confirm a choice.

Note To change any of these settings for a virtual device after you complete the initial setup, you must use the CLI.

Step 1 Open a console to the threat defense virtual.

Step 2 At the **firepower login** prompt, log in with the default credentials of **username** *admin* and the **password** *Admin123*.

Step 3 When the threat defense virtual system boots, a setup wizard prompts you for the following information required to configure the system:

- Accept EULA
- New admin password
- IPv4 or IPv6 configuration
- IPv4 or IPv6 DHCP settings
- Management port IPv4 address and subnet mask, or IPv6 address and prefix
- System name
- Default gateway
- DNS setup
- HTTP proxy
- Management mode (local management required)

Step 4 Review the Setup wizard settings. Defaults or previously entered values appear in brackets. To accept previously entered values, press **Enter**.

Step 5 Complete the system configuration as prompted.

Step 6 Verify the setup was successful when the console returns to the # prompt.

Step 7 Close the CLI.

What to do next

Your next steps depend on what management mode you chose.

- If you chose **No** for **Enable Local Manager**, you'll use the management center to manage your threat defense virtual; see Managing the Secure Firewall Threat Defense Virtual with the Secure Firewall Management Center.

See How to Manage Secure Firewall Threat Defense Virtual Device for an overview of how to choose your management option.

Troubleshooting

This section provides you with some basic troubleshooting steps related to your KVM deployment on your virtual machine.

Verify whether your virtual machine is running the KVM

You can use the following methods to verify whether your virtual machine is running the KVM:

- Run the **lsmod** command to list the modules in the Linux Kernel. If the KVM is running, it is indicated by displaying the following output:

  root@kvm-host:~$ lsmod | grep kvm

  kvm_intel 123675 0

  kvm 257361 1 kvm_intel

- If **ls -l /dev/kvm** command does not exist on the target VM then you are probably running **QEMU**, and not taking advantage of the KVM hardware assist features.

  root@kvm-host:~$ ls -l /dev/kvm

  crw------- 1 root root 10, 232 Mar 23 13:53 /dev/kvm

- Run the following command to also check whether the host machine supports KVM:

  root@kvm-host:~$ sudo kvm-ok

- You can also use KVM acceleration.

Experiencing boot loops while deploying the Threat Defense Virtual

You must ensure the following if your virtual machine experiences boot loops:

- Ensure to deploy the VM with at least 8 GB of memory.
- Ensure to deploy the VM with a minimum of 4 interfaces.

- Ensure to deploy the VM with a minimum of 4 vCPUs.
- Verify that the QEMU process is using a server class CPU, for example, SandyBridge, IvyBridge, Haswell, and so forth. Use the following command, **ps -edaf | grep qemu** to inspect the process params.

Experiencing boot loops while deploying the Management Center Virtual

You must ensure the following if your virtual machine experiences boot loops:

- Ensure that you deploy the VM with at least 28 GB of memory.
- Ensure to deploy the VM with a minimum of 4 interfaces.
- Ensure to deploy the VM with a minimum of 4 vCPUs.
- Verify that the QEMU process is using a server class CPU, for example, SandyBridge, IvyBridge, Haswell, and so forth. Use the following command, **ps -edaf | grep qemu** to inspect the process params.

Post deployment troubleshooting

You can run the following command on the threat defense virtual to check the issues to capture logs for debugging, **system generate-troubleshoot <space> ALL**

Alternatively, use **system generate-troubleshoot <space>**, followed by a question mark **(?)** or a **Tab** button to view the possible option or command.