Configuring VM-FEX for KVM

This chapter includes the following sections:

Guidelines and Prerequisites for KVM

Consider the following guidelines and prerequisites when configuring Kernel-based Virtual Machine (KVM):

- The host must be managed by Cisco UCS Manager Release 2.1 or later.
- On RHEL hosts, disable generic receive offload (GRO) by using the `ethtool`-K *interface* **gro off** command. This issue occurs because Microsoft Windows VIRTIO does not support GRO, which results in very poor Ethernet performance compared with Linux VMs.
- The host operating system must be Red Hat Enterprise Linux (RHEL) with KVM support.
  - The Single Root I/O Virtualization (SR-IOV) with MacVTap topology requires RHEL 6.2 or later.
  - The SR-IOV passthrough topology requires RHEL 6.3 or later.

  For more information about installing RHEL with KVM, see the *Red Hat Enterprise Virtualization for Servers Installation Guide*.
- The host must have libvirt with virsh or virt-manager installed for creating and managing the VMs.
- One or more Cisco VIC adapters must be installed in the host.

  For more information about installing a Cisco VIC adapter, see the *Cisco UCS 5108 Server Chassis Hardware Installation Guide*.

Consider the following guidelines and prerequisites when configuring an SR-IOV topology:

- Intel VT-x and VT-d processor extensions for virtualization must be enabled in the host BIOS.

For more information about configuring Cisco UCS server BIOS settings, see the *Cisco UCS Manager GUI Configuration Guide*.

- For SR-IOV topologies, configure a dynamic connection policy in a service profile. Apply the service profile on a static vNIC to specify the number of VFs, the fabric preference, and the adapter policy. The static vNIC becomes a PF when you configure one or more VFs under it. VFs are provisioned as dynamic vNICs.
- All VF-based dynamic vNICs must be provisioned on the same physical adapter as the parent static vNIC (PF).
- When you upgrade Cisco UCS Manager to an SR-IOV capable release, the existing static and dynamic vNICs are not automatically enabled for SR-IOV. To convert to SR-IOV, you must disable any existing dynamic connection policy in the service profile and then specify a reference to a dynamic connection policy under a static vNIC.

Configuring VM-FEX for SR-IOV with MacVTap Topology

Before You Begin

Prepare the host server as described in Guidelines and Prerequisites for KVM, on page 1.

Procedure

	Command or Action	Purpose
Step 1	In Cisco UCS Manager, configure a service profile for VM-FEX for KVM.	Create or modify a dynamic vNIC connection policy. For more information, see Configuring a Service Profile with VM-FEX.
Step 2	In Cisco UCS Manager, define a port profile and associate it with a port profile client.	Create a port profile to define the properties and settings used to configure the virtual interfaces. For KVM, you must select the default cluster as the port profile client. For more information, see Configuring Port Profiles.
Step 3	On each KVM server, use virsh or virt-manager to create one or more virtual machines (VMs).	For more information about installing VMs using these libvirt-based utilities, see the documents listed in KVM Components. **Note** When creating a VM using virsh, or when editing the VM domain XML descriptor file, use care when entering data such as a universally unique identifier (UUID), as you will receive no indication of incorrect data values or formats.
Step 4	For each VM, edit the domain XML descriptor file (and any network XML files, if present) to configure a vNIC interface that is directly attached to the VIC and uses the port profile defined in Cisco UCS Manager.	For more information about configuring a VM interface, see Configuring the VM Interface, on page 4.

	Command or Action	Purpose
Step 5	On each VM, install the VirtIO paravirtualized network driver (virtio-net) for the guest operating system.	Recent versions of most common operating systems provide default virtio-net drivers. For more information, contact Red Hat or the provider of the guest operating system.

Configuring VM-FEX for SR-IOV Passthrough Topology

Before You Begin

Prepare the host server as described in Guidelines and Prerequisites for KVM, on page 1.

Procedure

	Command or Action	Purpose
Step 1	In Cisco UCS Manager, configure a service profile for VM-FEX for KVM.	Create or modify a dynamic vNIC connection policy. For more information, see Configuring a Service Profile with VM-FEX.
Step 2	In Cisco UCS Manager, define a port profile and associate it with a port profile client.	Create a port profile to define the properties and settings used to configure the virtual interfaces. For KVM, you must select the default cluster as the port profile client. For more information, see Configuring Port Profiles.
Step 3	On each KVM server, use virsh or virt-manager to create one or more virtual machines (VMs).	For more information about installing VMs using these libvirt-based utilities, see the documents listed in KVM Components. **Note** When creating a VM using virsh, or when editing the VM domain XML descriptor file, use care when entering data such as a universally unique identifier (UUID), as you will receive no indication of incorrect data values or formats.
Step 4	On each VM, install an enic driver that supports an SR-IOV VF.	With RHEL 6.3 or later, use the inbox enic driver.
Step 5	For each VM, edit the domain XML descriptor file (and any network XML files, if present) to configure a vNIC interface that is directly attached to the VIC and uses the port profile defined in Cisco UCS Manager.	For more information about configuring a VM interface, see Configuring the VM Interface, on page 4.

	Command or Action	Purpose
Step 6	On the KVM host, activate the Intel VT-d extensions.	For more information about activating the VT-d extensions, see Activating Intel VT-d in the Kernel, on page 7.

Configuring the VM Interface

After creating a VM using a libvirt-based utility, you must manually edit the domain XML file of the VM to add and configure a direct attached interface for network connectivity.

For more information about the domain XML file components and attributes, see the libvirt documentation at http://libvirt.org/formatdomain.html#elementsNICS.

You can also compose a network XML file to specify a pool of devices. For more information about the network XML file components and attributes, see http://libvirt.org/formatnetwork.html.

Procedure

Step 1 Shut down the VM to be configured.

Step 2 Using the virsh editor, open the domain XML file of the VM for editing.

Example:
This example opens a domain XML file for editing in the virsh editor:

```
[root@chassis1blade5 qemu]# virsh edit vm1-rhel6.2
```

Step 3 In the devices section of the domain XML file, add an interface element that describes a vNIC for the VM. The components and attributes of the interface element are described in the Example section.

Step 4 Restart the VM.

Example for SR-IOV with MacVTap Mode

This example shows an interface element added to the domain XML file of a VM for connection in SR-IOV with MacVTap (MacVTap Passthrough) topology:

```
<domain type='kvm'>
  <name>vm1-rhel6.2</name>
  ...
  <devices>
    ...
    <interface type='direct'>
      <mac address='01:23:45:67:89:ab' />
      <source dev='eth4' mode='passthrough' />
      <virtualport type='802.1Qbh'>
        <parameters profileid='my-port-profile-3' />
      </virtualport>
      <model type='virtio' />
      <driver name='vhost' />
    </interface>
    ...
```

```
  </devices>
  ...
</domain>
```

This list describes the components and attributes of the interface element:

- `interface type='direct'`

  The `direct` type attribute value selects a direct logical attachment of the vNIC to the physical interface of the hypervisor, using the MacVTap driver.

- `mac address='01:23:45:67:89:ab'`

  Explicit specification of the MAC address is optional. Enter a MAC address obtained from your network administrator. If this line is omitted, libvirt generates a MAC address for the vNIC.

  Note We recommend that you do not assign a MAC address used by another VM, even if that VM is currently shut down or is no longer used. If you must reuse a MAC address from a previous VM, make sure that the retention timer has expired and ensure that the previous VM is no longer present in the Cisco UCS Manager view.

- `source dev='eth4' mode='passthrough'`

  The `passthrough` mode attribute value specifies that each VM is connected to the network by a macvtap direct connection with a virtual function (VF). The source interface must be a VF, and not a physical function (PF).

- `virtualport type='802.1Qbh'`

  The `802.1Qbh` type attribute value specifies that the vNIC is connected to an 802.1Qbh extended port for external switching.

- `parameters profileid='my-port-profile-3'`

  This line specifies the name of the port profile to be associated with the interface. The specified port profile must be already defined in Cisco UCS Manager and use the naming syntax described in Creating a Port Profile.

- `model type='virtio'`

  This line specifies that the interface uses the VirtIO paravirtualized front-end device driver.

- `driver name='vhost'`

  This line specifies that, for higher performance, the interface uses the vhost kernel back-end device driver and not the qemu userspace back-end driver.

Example for SR-IOV Passthrough Mode

This example shows an interface element that is added to the domain XML file of a VM for a connection in SR-IOV Passthrough topology:

```
<domain type='kvm'>
  <name>vm1-rhel6.3</name>
  ...
  <devices>
    ...
    <interface type='hostdev' managed='yes'>
      <source>
        <address type='pci' domain='0' bus='0x09' slot='0x0' function='0x01'/>
```

```
      </source>
      <mac address='01:23:45:67:89:ab' />
      <virtualport type='802.1Qbh'>
        <parameters profileid='my-port-profile-3' />
      </virtualport>
    </interface>
    ...
  </devices>
  ...
</domain>
```

This list describes the components and attributes of the interface element that differ from those described in the SR-IOV with MacVTap mode example:

- `interface type='hostdev'`

  The `hostdev` type attribute value selects a direct logical attachment of the vNIC to a PCI network device specified by the <source> element.

- `address type='pci' domain='0' bus='0x09' slot='0x0' function='0x01'.`

  The `address type` attribute value specifies the PCI address of the host VF. To obtain the address information, you need to run the `lspci` command at the Linux prompt. When you run the command, an address string is displayed, for example, `09:00.1 Ethernet controller: Cisco Systems Inc Device 0071 (rev a2)`. In the address string `09.00.1`, 09 indicates the bus, 00 indicates the slot, and 1 indicates the function.

- `mac address='01:23:45:67:89:ab'`

  Explicit specification of the MAC address is optional. Enter a MAC address that you obtained from your network administrator. If this line is omitted, libvirt generates a MAC address for the vNIC.

Note We recommend that you do not assign a MAC address used by another VM, even if that VM is currently shut down or is no longer used. If you must reuse a MAC address from a previous VM, make sure that the retention timer has expired and ensure that the previous VM is no longer present in the Cisco UCS Manager view.

Example of Using a Network XML File to Specify a Pool of Devices

This example shows how to use a network XML file to specify a pool of devices. In RHEL 6.2 or later, create the network file in /etc/libvirt/qemu/networks. List the devices and define a portgroup:

```
<network>
  <name>macvtap_passthru_network</name>
    <forward mode='passthrough'>
      <interface dev='eth2' />
      <interface dev='eth3' />
    </forward>
    <portgroup name='engineering'>
      <virtualport type='802.1Qbh'>
        <parameters profileid='my-port-profile-3' />
      </virtualport>
    </portgroup>
</network>
```

Edit the domain XML file of the VM to reference the network file and portgroup:

```
<domain type='kvm'>
  <name>vm1-rhel6.2</name>
```

```
  ...
  <devices>
    ...
    <interface type='network'>
      <mac address='01:23:45:67:89:ab' />
      <source network='macvtap_passthru_network' portgroup='engineering' />
      <model type='virtio' />
    </interface>
    ...
  </devices>
  ...
</domain>
```

Use the `virsh net-define <new-xml-filename>` command to create the new network from the new network XML file.

Tip You can find the network-related virsh commands with `virsh help | grep net-`

You can view help on any virsh command with `virsh help <command-name>`

This list describes the components and attributes of the interface element that differ from those described in the SR-IOV with MacVTap mode example:

- `interface type='network'`

  The `network` type attribute value specifies an attachment of the vNIC to a PCI network device from the pool listed in a network file.

- `source network='macvtap_passthru_network' portgroup='engineering'`

  The `network` and `portgroup` attribute values specify the name of a network XML file and its pool of network devices.

Activating Intel VT-d in the Kernel

Perform this procedure on the KVM host to enable Intel VT-d extensions, which are required for SR-IOV passthrough.

For more information about this feature in Red Hat Enterprise Linux (RHEL) systems, see the *Red Hat Virtualization Host Configuration and Guest Installation Guide*.

Procedure

Step 1 On the KVM host, open the file grub.conf for editing.
The file is typically located in the /boot directory. In RHEL systems, you can also access it using the grub.conf link in the /etc directory.

Step 2 Locate the line beginning with **kernel**.

Step 3 Append the command **intel_iommu=on** to the kernel line..

Example:

```
kernel /vmlinuz-2.6.18-190.e15 ro root=/dev/VolGroup00/LogVol00 \
               rhgb quiet intel_iommu=on
```

Step 4 Save the file.

What to Do Next

Reboot the host.