



Configuring KVM Components for VM-FEX

This chapter includes the following sections:

- [Configuring KVM Components for VM-FEX, page 1](#)
- [Configuring the VM Interface, page 2](#)

Configuring KVM Components for VM-FEX

Procedure

- Step 1** If not already present, install one or more Cisco UCS M81KR Virtual Interface Card adapters in each physical server that will be used for VM-FEX for KVM.
For more information about installing a Cisco UCS M81KR Virtual Interface Card, see the *Cisco UCS 5108 Server Chassis Hardware Installation Guide*.
- Step 2** Install Red Hat Enterprise Linux (RHEL) 6.1 or later as the hypervisor on each physical server that will be used for VM-FEX for KVM.
For more information about installing RHEL as the hypervisor, see the *Red Hat Enterprise Virtualization for Servers Installation Guide*.
- Step 3** On each KVM server, verify that the Intel VT-x processor extensions for virtualization are enabled in the BIOS.
For more information about configuring BIOS settings, see the *Cisco UCS Manager GUI Configuration Guide*.
- Step 4** On each KVM server, use `virsh` or `virt-manager` to create one or more VMs.
For more information about installing VMs using these libvirt-based utilities, see the documents listed in [Related Cisco UCS Documentation](#).
- Note** When creating a VM using `virsh`, or when editing the VM domain XML descriptor file, use care when entering data such as UUIDs, as you will receive no indication of incorrect data values or formats.
- Step 5** For each VM, edit the domain XML descriptor file to configure a vNIC interface that is directly attached to the VIC and uses the port profile defined in UCS Manager.
For more information about configuring a VM interface, see [Configuring the VM Interface, on page 2](#).

- Step 6** 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 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 vml-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>vml-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>
```

```
</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>vml-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' />
    </interface>
  </devices>
</domain>
```

```

    <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.

- `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>vml-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>

```

```
    ...  
</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.

