



VM-FEX for KVM の設定

- [KVM のガイドラインと前提条件 \(1 ページ\)](#)
- [MacVTap トポロジを使用する VM-FEX for SR-IOV の設定 \(2 ページ\)](#)
- [SR-IOV パススルー トポロジ用の VM-FEX の設定 \(3 ページ\)](#)
- [VM インターフェイスの設定 \(4 ページ\)](#)
- [カーネルでの Intel VT-d のアクティブ化 \(8 ページ\)](#)

KVM のガイドラインと前提条件

カーネルベースの仮想マシン (KVM) を設定する場合は、次のガイドラインおよび前提条件を考慮してください。

- ホストは、Cisco UCS Manager Release 2.1 以降で管理する必要があります。
- Red Hat Enterprise Linux (RHEL) ホストで、`ethtool-Kinterfacegro off` コマンドを使用して汎用受信オフロード (GRO) を無効にします。この問題は、Microsoft Windows VIRTIO が GRO をサポートしないことにより発生するもので、Linux VM と比べてイーサネットパフォーマンスが大幅に低下します。
- ホストのオペレーティングシステムは、KVM をサポートする RHEL である必要があります。
 - MacVTap トポロジを使用した単一のルート I/O 仮想化 (SR-IOV) には RHEL 6.2 以降が必要です。
 - SR-IOV パススルー トポロジには RHEL 6.3 以降が必要です。

KVM を使用する RHEL のインストールに関する詳細については、『*Red Hat Enterprise Virtualization for Servers Installation Guide*』を参照してください。

- ホストには、VM の作成と管理のために、`virsh` または `virt-manager` がインストールされた `libvirt` を設定する必要があります。
- 1 つまたは複数の Cisco VIC アダプタは、このホストにインストールする必要があります。
Cisco VIC アダプタのインストールの詳細については、『*Cisco UCS 5108 Server Chassis Hardware Installation Guide*』を参照してください。

SR-IOV トポロジを設定する場合は、次のガイドラインおよび前提条件を考慮してください。

- **[Policies]** タブで、サービス プロファイルには、デフォルト ポリシーの BIOS ポリシーで SRIOV を割り当てます。

Cisco UCS サーバ BIOS 設定の構成に関する詳細については、『*Cisco UCS Manager GUI Configuration Guide*』を参照してください。

- SR-IOV トポロジでは、サービス プロファイルのダイナミック接続ポリシーを設定します。VF の数、ファブリック設定、アダプタポリシーを指定するには、スタティック vNIC にサービス プロファイルを適用します。スタティック vNIC の下で 1 つ以上の VF を設定する場合、スタティック vNIC は PF になります。VF はダイナミック vNIC としてプロビジョニングされます。
- SR-IOV 対応リリースに Cisco UCS Manager をアップグレードする場合、既存のスタティックおよびダイナミック vNIC は SR-IOV で自動的に有効になりません。SR-IOV に変換するには、サービス プロファイルのダイナミック接続ポリシーを無効にして、スタティック vNIC でダイナミック接続ポリシーへの参照を指定する必要があります。

MacVTap トポロジを使用する VM-FEX for SR-IOV の設定

始める前に

[KVM のガイドラインと前提条件 \(1 ページ\)](#) の説明のとおりホスト サーバを準備します。

手順

ステップ 1 Cisco UCS Manager で、KVM 用 VM-FEX のサービス プロファイルを設定します。

ダイナミック vNIC 接続ポリシーを作成または変更します。

詳細については、[VM-FEX を使用したサービス プロファイルの設定](#)を参照してください。

ステップ 2 Cisco UCS Manager で、ポート プロファイルを定義し、ポート プロファイルクライアントに関連付けます。

ポート プロファイルを作成し、仮想インターフェイスを設定するために使用するプロパティと設定を定義します。KVM では、デフォルト クラスタをポート プロファイルクライアントとして選択する必要があります。

詳細については、[ポート プロファイルの設定](#)を参照してください。

ステップ 3 Cisco UCS Manager のサービス プロファイルにある vNIC の **[Connection Policies]** 領域で、**[Dynamic vNIC]** オプション ボタンを選択し、作成したダイナミック vNIC 接続ポリシーを割り当てます。

ステップ 4 各 KVM サーバで、virsh または virt-manager を使用して 1 つ以上の仮想マシン (VM) を作成します。

これらの libvirt ベースのユーティリティを使用して VM をインストールする方法については、[KVM のコンポーネント](#)のドキュメントを参照してください。

(注) virsh を使用して VM を作成する場合、または VM ドメイン XML 記述子ファイルを編集する場合、誤ったデータ値またはフォーマットの通知が送られないため、汎用一意識別子 (UUID) などのデータの入力時には注意してください。

ステップ 5 各 VM では、ドメイン XML 記述子ファイル（および存在する場合はネットワーク XML ファイル）を編集し、VIC に直接接続され、Cisco UCS Manager で定義されたポート プロファイルを使用する vNIC インターフェイスを設定します。

VM インターフェイスの設定に関する詳細については、[VM インターフェイスの設定 \(4 ページ\)](#) を参照してください。

ステップ 6 ゲスト オペレーティング システムの VirtIO のパラ仮想化されたネットワーク ドライバ (virtio-net) に、インボックス enic ドライバを使用します。

最も一般的なオペレーティング システムの最新バージョンでは、デフォルトの virtio-net ドライバが提供されます。詳細については、ゲスト オペレーティング システムの Red Hat またはプロバイダーにお問い合わせください。

SR-IOV パススルー トポロジ用の VM-FEX の設定

始める前に

[KVM のガイドラインと前提条件 \(1 ページ\)](#) の説明のとおりホスト サーバを準備します。

手順

ステップ 1 Cisco UCS Manager で、KVM 用 VM-FEX のサービス プロファイルを設定します。

ダイナミック vNIC 接続ポリシーを作成または変更します。

詳細については、[VM-FEX を使用したサービス プロファイルの設定](#)を参照してください。

ステップ 2 Cisco UCS Manager で、ポート プロファイルを定義し、ポート プロファイル クライアントに関連付けます。

ポート プロファイルを作成し、仮想インターフェイスを設定するために使用するプロパティと設定を定義します。KVM では、デフォルト クラスタをポート プロファイル クライアントとして選択する必要があります。

詳細については、[ポート プロファイルの設定](#)を参照してください。

ステップ 3 Cisco UCS Manager のサービス プロファイルにある vNIC の **[Connection Policies]** 領域で、**[Dynamic vNIC]** オプション ボタンを選択し、作成したダイナミック vNIC 接続ポリシーを割り当てます。

ステップ 4 KVM ホストで、Intel VT-d 拡張をアクティブにします。

VT-d 拡張のアクティブ化の詳細については、[カーネルでの Intel VT-d のアクティブ化 \(8 ページ\)](#) を参照してください。

ステップ 5 各 KVM サーバで、`virsh` または `virt-manager` を使用して 1 つ以上の仮想マシン (VM) を作成します。

これらの `libvirt` ベースのユーティリティを使用して VM をインストールする方法については、[KVM のコンポーネント](#) のドキュメントを参照してください。

(注) `virsh` を使用して VM を作成する場合、または VM ドメイン XML 記述子ファイルを編集する場合、誤ったデータ値またはフォーマットの通知が送られないため、汎用一意識別子 (UUID) などのデータの入力時には注意してください。

ステップ 6 各 VM では、ドメイン XML 記述子ファイル (および存在する場合はネットワーク XML ファイル) を編集し、VIC に直接接続され、Cisco UCS Manager で定義されたポートプロファイルを使用する vNIC インターフェイスを設定します。

VM インターフェイスの設定に関する詳細については、[VM インターフェイスの設定 \(4 ページ\)](#) を参照してください。

ステップ 7 各 VM で、SR-IOV VF をサポートする `enic` ドライバをインストールします。

RHEL 6.3 以降で、受信箱 `enic` ドライバを使用します。

VM インターフェイスの設定

`libvirt` ベースのユーティリティを使用して VM を作成すると、VM のドメイン XML、またはドメイン XML から参照できる個別のネットワーク XML ファイルにポートプロファイル情報などのネットワーク設定を追加できます。個別のネットワーク XML ファイルに設定情報を追加する上での利点の 1 つは、デバイスプールを指定できることです。ネットワーク XML ファイルコンポーネントおよび属性の詳細については、<https://libvirt.org/formatnetwork.html> を参照してください。

ドメイン XML ファイルコンポーネントおよび属性の詳細については、<https://libvirt.org/formatdomain.html#elementsNICS> にある `libvirt` ドキュメントを参照してください。

手順

ステップ 1 設定する VM をシャットダウンします。

ステップ 2 `virsh` エディタを使用して、編集のために VM のドメイン XML ファイルを開きます。

例 :

この例は、`virsh` エディタで編集するためのドメイン XML ファイルが開いています。

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

ステップ3 ドメイン XML ファイルのデバイス セクションで、VM の vNIC を表すインターフェイス要素を追加します。

インターフェイス要素のコンポーネントおよび属性については、例セクションで説明します。

ステップ4 VM を再起動します。

例

MacVTap モードを使用した SR-IOV の例

ここでは、MacVTap (MacVTap パススルー) を使用した SR-IOV で接続するために、VM のドメイン XML ファイルにインターフェイス要素を追加する例について説明します。

```
<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>
```

この一覧は、インターフェイス要素のコンポーネントおよび属性について説明します。

- `interface type='direct'`

`direct` タイプ属性値は、MacVTap ドライバを使用して、ハイパーバイザの物理インターフェイスに vNIC を直接論理的に接続します。

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

MAC アドレスの仕様を明示するのは任意です。ネットワーク管理者から取得した MAC アドレスを入力します。この行を省略すると、libvirt は vNIC の MAC アドレスを生成します。



- (注) VMが現在シャットダウンされていたり、またはすでに使用されていない場合でも、別の VM で使用される MAC アドレスを割り当てないことをお勧めします。以前の VM の MAC アドレスを再利用する場合は、その保持期間が切れ、以前の VM が Cisco UCS Manager のビューに表示されていないことを確認します。

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

`passthrough` モード属性値は、各 VM が、仮想機能 (VF) を使用した `macvtap` 直接接続によってネットワークに接続されるように指定します。ソースインターフェイスは、物理機能 (PF) ではなく、VF である必要があります。

- `virtualport type='802.1Qbh'`

`802.1Qbh` タイプ属性値は、vNIC が外部スイッチの `802.1Qbh` 拡張ポートに接続されるように指定します。

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

この行は、インターフェイスに関連付けられるポートプロファイルの名前を指定します。ポートプロファイル名は、大文字と小文字が区別されます。指定したポートプロファイルは Cisco UCS Manager で定義済みで、[ポートプロファイルの作成](#)で説明された命名構文を使用する必要があります。

- `model type='virtio'`

この行は、ゲストインターフェイスが VirtIO のパラ仮想化されたフロントエンドデバイス ドライバを使用するように指定します。

- `driver name='vhost'`

この行は、性能向上のために、ホスト側インターフェイスが、`qemu` ユーザ空間バックエンドドライバではなく `vhost` カーネルバックエンドデバイスドライバを使用するように指定します。

SR-IOV パススルー モードの例

ここでは、SR-IOV パススルー トポロジで接続するために、インターフェイス要素が VM のドメイン XML ファイルに追加される例を示します。

```
<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>
```

```

    </virtualport>
  </interface>
  ...
</devices>
...
</domain>

```

この一覧は、MacVTapモードを使用したSR-IOVの例で説明したのとは異なるインターフェイス要素のコンポーネントおよび属性について説明します。

- `interface type='hostdev'`

`hostdev` タイプ属性では、ゲストにホスト デバイスを直接割り当てることができます。

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

`address type` 属性値は、ホスト VF の PCI アドレスを指定します。アドレス情報を取得するには、Linux プロンプトで `lspci` コマンドを実行できます。コマンドを実行すると、アドレス文字列が `09:00.1 Ethernet controller: Cisco Systems Inc Device 0071 (rev a2)` のように表示されます。アドレス文字列 `09.00.1` で、`09` はバスを、`00` はスロット、`1` は機能を示します。

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

MAC アドレスの仕様を明示するのは任意です。ネットワーク管理者から取得した MAC アドレスを入力します。この行を省略すると、`libvirt` は `vNIC` の MAC アドレスを生成します。



(注) VMが現在シャットダウンされていたり、またはすでに使用されていない場合でも、別のVMで使用されるMACアドレスを割り当てないことをお勧めします。以前のVMのMACアドレスを再利用する場合は、その保持期間が切れ、以前のVMがCisco UCS Managerのビューに表示されていないことを確認します。

ネットワーク XML ファイルを使用してデバイス プールを指定する例

この例は、デバイス プールを指定するためにネットワーク XML ファイルを使用する方法について示します。RHEL 6.2 以降では、`/etc/libvirt/qemu/networks` にネットワーク ファイルを作成します。デバイスを一覧表示し、ポートグループを定義します。

```

<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>

```

```

    </virtualport>
  </portgroup>
</network>

```

ネットワーク ファイルおよびポートグループを参照する VM のドメイン XML ファイルを編集します。

```

<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>

```

新しいネットワーク XML ファイルを作成したら、`virsh net-define <new-xml-filename>` コマンドを使用して新しいネットワーク XML ファイルから新しいネットワークを作成します。



ヒント

`virsh help | grep net-` を使用してネットワークに関連する `virsh` コマンドを見つけることができます。

`virsh help <command-name>` を使用して `virsh` コマンドのヘルプを表示できます。

この一覧は、MacVTap モードを使用した SR-IOV の例で説明したのとは異なるインターフェイス要素のコンポーネントおよび属性について説明します。

- `interface type='network'`
`network` タイプ属性値は、個別のネットワーク ファイルに表示されたプールから、PCI ネットワーク デバイスに VM vNIC を接続するように指定します。
- `source network='macvtap_passthru_network' portgroup='engineering'`
`network` および `portgroup` の属性値は、ネットワーク XML ファイルの名前とそのネットワーク デバイスのプールを指定します。

カーネルでの Intel VT-d のアクティブ化

Intel VT-d 拡張を有効にするには KVM ホストで次の手順を実行します。SR-IOV パススルーには必須です。

Red Hat Enterprise Linux (RHEL) システムのこの機能の詳細については、『*Virtualization Deployment and Administration Guide*』を参照してください。

手順

ステップ 1 KVM ホストで、編集用の `grub.conf` ファイルを開きます。

通常、このファイルは `/boot` ディレクトリに配置されています。RHEL システムでは、`/etc` ディレクトリの `grub.conf` リンクを使用してアクセスすることもできます。

ステップ 2 `kernel` で始まる行を特定します。

ステップ 3 このカーネルの行に `intel_iommu=on` コマンドを追加します。。

例：

```
kernel /vmlinuz-2.6.18-190.e15 ro root=/dev/VolGroup00/LogVol100 \  
rhgb quiet intel_iommu=on
```

ステップ 4 ファイルを保存します。

次のタスク

ホストをリブートします。

