



KVM 環境でのコントローラのインストール

- [カーネルベースの仮想マシン環境の概要 \(1 ページ\)](#)
- [KVM 環境でのインストール手順 \(2 ページ\)](#)
- [.qcow2 イメージを使用した Linux ブリッジ ネットワーキングでのコントローラのインストール \(3 ページ\)](#)
- [ISO イメージを使用した Vrish でのコントローラのインストール \(4 ページ\)](#)
- [.qcow2 イメージを使用した OVS ネットワークでのコントローラのインストール \(5 ページ\)](#)
- [ブートストラップ設定を使用した Vrish でのコントローラのインストール \(5 ページ\)](#)
- [ISO イメージを使用した VMM でのコントローラ インスタンスの作成 \(6 ページ\)](#)
- [KVM VMM \(virt-manager\) でのブートストラップ設定 \(7 ページ\)](#)
- [KVM での SR-IOV の設定 \(8 ページ\)](#)
- [コントローラへの SR-IOV の接続 \(11 ページ\)](#)
- [SR-IOV ドライバとファームウェアバージョンの確認 \(13 ページ\)](#)

カーネルベースの仮想マシン環境の概要

クラウドの Cisco Catalyst 9800 ワイヤレス コントローラ は、カーネルベースの仮想マシン (KVM) を使用して Ubuntu、Red Hat Enterprise Linux (RHEL) 7.2、および Red Hat Enterprise Virtualization (RHEV) の上部でサポートされます。KVM でのインストールでは仮想マシン (VM) の作成と、.iso ファイルまたは .qcow2 ファイルを使用したインストールが必要です。VM は、KVM コマンドラインまたは Virsh を使用して起動できます。

- **.qcow2** : KVM 環境でソフトウェア イメージをブートするために使用します。
- **.iso** : Virsh ツールを使用して手動でクラウドの Cisco Catalyst 9800 ワイヤレス コントローラ をインストールするために使用します。また、KVM 環境で virsh コマンドを使用してコントローラを起動するには、サンプル XML 設定の virsh.xml ファイルも必要です。

サポートされているプロファイルの設定

次のプロファイル設定がサポートされています。

表 1: サポートされているプロファイルの設定

テンプレート	CPU	RAM	AP	クライアント
Small	4 vCPU	8 GB	1000	10000
Medium	6 vCPU	16 GB	3000	320000
Large	10 vCPU	32 GB	6000	640000

サポートされているネットワーキングオプション

次のネットワーキングオプションがサポートされています。

- Linux ブリッジ
- Open vSwitch (OVS)

KVM のインストールに必要なパッケージ

KVM をインストールするには、次のパッケージが必要です。

- Qemu-kvm
- Qemu-utils
- Uml-utilities
- Socat
- KVM
- Libvirt-bin
- Virtinst

KVM 環境でのインストール手順

一連のインストールステップを説明する自動インストールパッケージを使用するか、または virt-manager、virt-install、virsh など、KVM でサポートされている管理ソフトウェアのいずれかを使用して KVM 環境にクラウドの Cisco Catalyst 9800 ワイヤレスコントローラをインストールできます。

KVM インストーラパッケージは KVM 用の自動インストールパッケージです。このパッケージを実行すると、次のモードを提供します。

- [Default] : バンドルされているイメージファイルとデフォルトの VM 設定オプション ([Small]、[Medium]、または [Large]) のいずれかを使用してコントローラをインストールします。
- [Interactive] : VM 設定のカスタマイズを許可し、バンドルされたイメージファイルか、または別の .qcow2 イメージをインストールするオプションを提供します。



(注) サポートされていない VM 操作のリストについては、[VMware ESXi 環境でのインストール](#)の章の「サポートされている VMware 機能と操作」セクションを参照してください。

始める前に

クラウドの Cisco Catalyst 9800 ワイヤレス コントローラ ソフトウェア インストール イメージ パッケージから .run 実行可能ファイルをダウンロードし、ホスト マシンのローカル ドライブ にコピーします。

.qcow2 イメージを使用した Linux ブリッジ ネットワーキングでのコントローラのインストール

この手順は、コントローラ用の VM を手動で作成するための一般的なガイドラインです。実行する必要がある正確なステップは、KVM 環境とセットアップの特性によって異なることがあります。詳細については、Red Hat Linux、Ubuntu、および Virsh のドキュメントを参照してください。

virt-install コマンドを使用してインスタンスを作成し、次の構文を使用してブートします。

```
--connect=qemu:///system \  
--os-type=linux \  
--os-variant=rhel4 \  
--arch=x86_64 \  
--cpu host \  
--console pty,target_type=virtio \  
--hvm \  
--import \  
--name=my_c9k_vm \  
--disk path=<path_to_c9800-c_qcow2>,bus=ide,format=qcow2 \  
--vcpus=1,sockets=1,cores=1,threads=1 \  
--ram=4096 \  
--network=network:<network name>,model=virtio \  
--network=network:<network name>,model=virtio \  
--network=network:<network name>,model=virtio \  
--noreboot \  

```

- (注) インストールが完了すると、コントローラ VM はシャットダウンされます。**virsh start** コマンドを使用してコントローラ VM を起動します。

ISO イメージを使用した **Virsh** でのコントローラのインストール

この手順は、コントローラ用の VM を手動で作成するための一般的なガイドラインです。実行する必要がある正確なステップは、KVM 環境とセットアップの特性によって異なることがあります。詳細については、Red Hat Linux、Ubuntu、および Virsh のドキュメントを参照してください。

ステップ 1 **qemu-img** コマンドを使用し、**.qcow2** 形式で 8 GB のディスク イメージを作成します。

```
qemu-img create -f qcow2 c9000-c_disk.qcow2 8G
```

ステップ 2 **virt-install** コマンドを使用してコントローラをインストールします。これには、新しい VM を作成するための適切な権限が必要です。次に、4 GB の RAM を持つ 1 つの vCPU VM と 3 つのネットワーク インターフェイスを作成する例を示します。

```
virt-install \
--connect=qemu:///system \
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--hvm \
--import \
--name=my_c9k_vm \
--cdrom=<path_to_c9800-c_iso> \
--disk path=c9000-c_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2 \
--ram=4096 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--noreboot \
```

- (注) **virt-install** コマンドで新しい VM インスタンスを作成し、コントローラは指定したディスク ファイルにイメージをインストールします。インストールが完了すると、コントローラ VM はシャットダウンされます。**virsh start** コマンドを使用してコントローラ VM を起動します。

.qcow2 イメージを使用した OVS ネットワークでのコントローラのインストール

この手順は、コントローラ用の VM を手動で作成するための一般的なガイドラインです。実行する必要がある正確なステップは、KVM 環境とセットアップの特性によって異なることがあります。詳細については、Red Hat Linux、Ubuntu、および Virsh のドキュメントを参照してください。

virt-install コマンドを使用してインスタンスを作成し、次の構文を使用してブートします。

```
--connect=qemu:///system \  
--os-type=linux \  
--os-variant=rhel4 \  
--arch=x86_64 \  
--cpu host \  
--console pty,target_type=virtio \  
--hvm \  
--import \  
--name=my_c9k_vm \  
--cdrom=<path_to_c9800-c_iso> \  
--disk path=c9000-c_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2 \  
--ram=4096 \  
--vcpus=1,sockets=1,cores=1,threads=1 \  
--network=network:<network name>,model=virtio \  
--network=network:<network name>,model=virtio \  
--network=network:<network name>,model=virtio \  
--noreboot \  

```

(注) インストールが完了すると、コントローラ VM はシャットダウンされます。**virsh start** コマンドを使用してコントローラ VM を起動します。

ブートストラップ設定を使用した Virsh でのコントローラのインストール

この手順は、コントローラ用の VM を手動で作成するための一般的なガイドラインです。実行する必要がある正確なステップは、KVM 環境とセットアップの特性によって異なることがあります。詳細については、Red Hat Linux、Ubuntu、および Virsh のドキュメントを参照してください。

始める前に

必要な設定で `iosxe_config.txt` というテキストファイルを作成し、`mkisofs -l -o iso-file-name.iso iosxe_config.txt` コマンドを使用し、`iosxe_config.txt` ファイルを入力として指定して `.iso` イメージを作成します。

```
mkisofs -l -o test.iso iosxe_config.txt
```

次にサンプルの設定ファイルを示します。

```
hostname C9800-CL
license smart enable
username lab privilege 15 password lab
ip domain-name cisco.com
interface GigabitEthernet1
 ip address 10.0.0.5 255.255.255.0
 no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
 login local
exit
```

virt-install コマンドを使用してコントローラをインストールします。このコマンドを使用するには、新しい VM を作成するための適切な権限が必要です。次に、4 GB の RAM を持つ 1 つの vCPU VM と 3 つのネットワーク インターフェイスを作成する例を示します。

```
virt-install \
--connect=qemu:///system \
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--console pty,target_type=virtio \
--hvm \
--import \
--name=my_c9k_vm \
--disk path=<path_to_c9800-c_qcow2>,bus=ide,format=qcow2 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--ram=4096 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--noreboot \
```

ISO イメージを使用した VMM でのコントローラ インスタンスの作成

ステップ 1 [Applications] > [System Tools] > [Virtual Machine Manager] を使用して `virt-manager` を起動します。

ハイパーバイザの選択およびルートパスワードの入力を求められる可能性があります。

- ステップ 2 上部にある [File] オプションを選択し、[New Virtual Machine] オプションを選択します。
- ステップ 3 仮想マシンの詳細を入力します。
- VM の名前を入力します。
 - オペレーティングシステム オプションで、[Local install media] を選択します。
 - [Forward] をクリックします。
- ステップ 4 ディスクから **ISO イメージ** を選択します。
- ステップ 5 [Automatically Detect operating system based on install media] を選択します。
- ステップ 6 メモリおよび CPU オプションを設定します。
- [Memory (RAM)] を設定します。
 - [CPUs] を設定します。
 - [Forward] をクリックして続行します。
- ステップ 7 ディスク イメージ サイズを 8 GB に設定し、[Forward] をクリックします。
- ステップ 8 インスタンス名を入力します。
- ステップ 9 最初に [Customize configuration before install] ボックスをオンにしてから、[Finish] をクリックします。
これにより、他の NIC を追加することができます。
- ステップ 10 [Network] タブを選択して他の NIC を追加します。
- ステップ 11 [Network source] ドロップダウンで [Network] を選択します。
- (注) virtio ネットワーク ドライバのみがサポートされています。
- ステップ 12 ドロップダウンを使用して、[Portgroup] を選択します。
- ステップ 13 [完了 (Finish)] をクリックします。

KVM VMM (virt-manager) でのブートストラップ設定

仮想マシン マネージャとも呼ばれる virt-manager は、libvirt を通じて仮想マシンを管理するためのデスクトップアプリケーションです。実行中のドメインの概要 (ライブ パフォーマンス やリソース使用率の統計情報) が表示されます。ウィザードを使用して、新しいドメインを作成したり、ドメインのリソース割り当てを設定/調整したり、仮想ハードウェアをイネーブルにすることができます。組み込みの VNC および SPICE クライアント ビューアは、ゲストドメイン用のフル機能のグラフィカルなコンソールとして使用できます。

- ステップ 1 [Applications] > [System Tools] > [Virtual Machine Manager] を使用して virt-manager を起動します。
ハイパーバイザの選択、およびルートパスワードの入力を求められる可能性があります。
- ステップ 2 上部にある [File] オプションを選択し、[New Virtual Machine] オプションをクリックします。
- ステップ 3 仮想マシンの詳細を入力します。

- a) [Name] を指定します。
- b) オペレーティング システムの場合、[Import existing disk image] を選択します。

この方法でディスク イメージ (qcow2 イメージを選択した場合は、事前にインストールされた、ブート可能なオペレーティング システムを含んでいるもの) をインポートできます。

- c) [Forward] をクリックして続行します。

ステップ 4 コントローラ qcow2 イメージパスを選択します。

ステップ 5 メモリおよび CPU オプションを設定します。

- a) [Memory (RAM)] を 8192 に設定します。
- b) [CPUs] を 4 に設定します。
- c) [Forward] をクリックして続行します。

ステップ 6 インスタンス名を入力します。

ステップ 7 [Finish] をクリックする前に [Customize configuration before install] ボックスをオンにします。

これにより、複数の NIC を追加することができます。

ステップ 8 [Network] を選択します。

ブリッジまたはネットワークのいずれかを選択します。

ステップ 9 [Finish] をクリックします。

ステップ 10 編集するインスタンス名をダブルクリックします。

ステップ 11 [i] を選択してインスタンス情報を取得します。

ステップ 12 [Begin Installation] を選択してインスタンスを起動します。

ステップ 13 [Monitor] 記号をクリックして仮想コンソールに移動します。

KVM での SR-IOV の設定

SR-IOV で推奨されるソフトウェアバージョン

表 2: サポートされている NIC タイプの一覧

NIC	ファームウェア	ドライバのバージョン	ホスト OS
Intel x710	7.10	I40e 2.10.19.82	KVM RedHat バージョン 7.5 以降
Ciscoized x710	7.0	I40e 2.10.19.82	KVM RedHat バージョン 7.5 以降

Intel VT-D の有効化



(注) 後続のタスクを実行するには、ルート権限が必要です。

Intel VT-D を有効にするには、次の手順を実行します。

ステップ 1 `/etc/sysconfig/grub` ファイルの `GRUB_CMDLINX_LINUX` 行で、末尾に `intel_iommu = on` および `iommu = pt` パラメータを追加します。

ステップ 2 次のコマンドを実行して、`/etc/grub2.cfg` ファイルを再生成します。

```
grub2-mkconfig -o /etc/grub2.cfg
```

(注) EFI の場合は、次のコマンドを実行します。

```
grub2-mkconfig -o /etc/grub2-efi.cfg
```

ステップ 3 変更を有効にするには、システムをリブートします。

これで、システムで PCI デバイスを割り当てることができるようになりました。

インターフェイスでの SR-IOV モード仮想機能 (VF) の設定

VF が使用できない場合は、次のコマンドを使用して SR-IOV VF を設定します。

ステップ 1 インターフェイスで VF を設定します。

```
echo "no_of_vfs" > /sys/class/net/<interface_name>/device/sriov_numvfs
```

サンプル出力：

```
echo 1 > /sys/class/net/enp129s0f0/device/sriov_numvfs
```

ここでは、最大のパフォーマンスを得るために、ポートあたり 1 つの VF が作成されます。

ステップ 2 次のコマンドを使用して、VF でスプーフィングチェック、信頼モード、および MAC を設定します。

```
ip link set dev enp129s0f0 vf 0 trust on
ip link set enp129s0f0 vf 0 spoofchk off
ip link set enp129s0f0 vf 0 mac 3c:fd:fe:de:cc:bc
```

(注) MAC アドレスは一意にする必要があります。

ステップ 3 次のコマンドを使用して設定を確認します。

```
ip link show interface_name
```

サンプル出力：

```
ip link show enp129s0f0
6: enp129s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen
1000
link/ether 3c:fd:fe:de:01:bc brd ff:ff:ff:ff:ff:ff
vf 0 MAC 3c:fd:fe:de:cc:bc, spoof checking off, link-state auto, trust on
```

SR-IOV 永続設定の構成

上記の方法で設定された SR-IOV 設定は、リブート後は保持されません。この問題を解決するには、ホストのリブート時に自動的に有効になるサービスとして上記の設定を実行します。

ステップ 1 永続化するコマンドを使用して `bash` スクリプトを作成します。次のように、`/usr/bin/sriov-config` ファイルにスクリプトを記述する必要があります。

```
#!/bin/sh
echo "no_of_vfs" > /sys/class/net/<interface_name>/device/sriov_numvfs
ip link set dev <interface_name> vf 0 trust on
ip link set <interface_name> vf 0 spoofchk off
ip link set <interface_name> vf 0 mac 3c:fd:fe:de:cc:bc
```

サンプル出力：

```
#!/bin/sh
echo 1 > /sys/class/net/enp129s0f0/device/sriov_numvfs
ip link set dev enp129s0f0 vf 0 trust on
ip link set enp129s0f0 vf 0 spoofchk off
ip link set enp129s0f0 vf 0 mac 3c:fd:fe:de:cc:bc
```

(注) すべての VF に対して同じ手順を繰り返す必要があります。

ステップ 2 スクリプトの実行権限を指定します。

```
chmod 777 /usr/bin/sriov-config
```

ステップ 3 システムサービスを作成します。ブートの最後に実行する新しいシステムサービスを定義します。このサービスでは、手順 1 で説明したように、必須の `sriov` コマンドを含む `bash` スクリプトを実行します。

(注) `/usr/lib/systemd/system` に `sriov.service` という名前の新しいファイルを作成し、次の内容を追加します。

```
[Unit]
Description=SR-IOV configuration
After=rc-local.service
Before=getty.target
[Service]
Type=oneshot
ExecStart=/usr/bin/sriov-config
[Install]
WantedBy=multi-user.target
```

(注) `ExecStart=/usr/bin/sriov-config` コマンドラインでスクリプトを実行します。

ステップ 4 次のコマンドを使用して、**sriov.service** を有効にし、開始します。

```
systemctl --now enable sriov.service
```

(注) このコマンドによってサービスが即座に開始され、ホストがリブートするたびにサービスが実行されるようにします。

KVM の SR-IOV 設定の詳細については、以下を参照してください。

<https://www.intel.com/content/www/us/en/embedded/products/networking/xl710-sr-iov-config-guide-gbe-linux-brief.html>

コントローラへの SR-IOV の接続

コマンドラインを使用した新しい仮想マシンへの接続

PCI VF デバイスを追加するには、**virt-install** の **host-device** オプションを使用します。手順 1 (インターフェイスでの SR-IOV モード仮想機能 (VF) の設定 (9 ページ)) の情報と PCI BDF 番号を使用して、デバイスを接続します。

10GbE SFP+ 用 Intel Corporation Ethernet Controller X710 の仮想機能。(enp129s0f0) :

PCI BDF	Interface
0000:18:06.0	enp129s0f0
0000:18:06.1	enp129s0f1

VM の作成と起動

VM を作成して起動するには、次のコマンドを使用します。

```
sudo virt-install --virt-type=kvm --name ewlc_sriov_3-18 --ram 16384 --vcpus=9 --hvm
--cdrom=/home/C9800-CL-universalk9.BLD_POLARIS_DEV_LATEST_20200318_062819-serial.iso
--network none --host-device=pci_0000_18_06_0 --host-device=pci_0000_18_06_1 --graphics
vnc --disk
path=/var/lib/libvirt/images/ewlc_sriov_3-18.qcow2,size=8,bus=virtio,format=qcow2
```

次のコマンドを使用して VM コンソールを表示します。

```
virsh console ewlc_sriov_3-18
Connected to domain ewlc_sriov_3-18
Escape character is ^]
```

次のコマンドを入力して、インターフェイスの SR-IOV ドライバを確認できます。

```
Device > enable
```

```
Device #show platform software vnic-if interface-mapping
```

```
Device # show platform software vnic-if interface-mapping
```

```
-----
Interface Name      Driver Name      Mac Addr
-----
```

```
GigabitEthernet2    net_i40e_vf    3cfd.fede.ccbd
GigabitEthernet1    net_i40e_vf    3cfd.fede.ccbc
-----
```



(注) 上記の MAC アドレスは、VF に設定されているアドレスと同じです。

次のコマンドを使用して、プロセッサ、メモリ、vNIC、ハイパーバイザ、およびスループットプロファイルの詳細を確認できます。

Device # show platform software system all

```
Device# show platform software system all
Controller Details:
=====
VM Template: medium
Throughput Profile: high
AP Scale: 3000
Client Scale: 32000
WNCD instances: 3

Processor Details
=====
Number of Processors : 9
Processor : 1 - 9
vendor_id : GenuineIntel
cpu MHz : 2593.748
cache size : 4096 KB
Crypto Supported : Yes
model name : Intel Core Processor (Haswell, IBRS)
Memory Details
=====
Physical Memory : 16363364KB

VNIC Details
=====
Name                Mac Address      Driver Name      Status Platform MTU
GigabitEthernet1    3cfd.fede.ccbc   net_i40e_vf      DOWN      1522
GigabitEthernet2    3cfd.fede.ccbd   net_i40e_vf      DOWN      1522

Hypervisor Details
=====
Hypervisor: KVM
Manufacturer: Red Hat
Product Name: KVM
Serial Number: Not Specified
UUID: 0E3546DD-DE6E-400D-9B3D-025215519CB8
image_variant :

Boot Details
=====
Boot mode: BIOS
Bootloader version: 1.1
```

KVM VMM を使用したコントローラへのインターフェ이스の接続 (virt-manager)

virt-manager で [Hardware] > [Add Hardware] を選択し、PCI ホストデバイスを VM に追加します。NIC カードに移動し、VM に接続する必要がある VF を選択します。

PCI が VM に追加されたら、VM を起動できます。

SR-IOV ドライバとファームウェアバージョンの確認

次のコマンドを使用して、イーサネットとドライバのバージョンを確認できます。

```
ethtool -i <interface_name>
```



(注) このコマンドは、ホストマシンで実行する必要があります。

```
[root@cpp-rhel-perf ~]# ethtool -i enp129s0f0
driver: i40e
version: 2.10.19.82
firmware-version: 7.10 0x8000646c 1.2527.0
expansion-rom-version:
bus-info: 0000:81:00.0
```

次のコマンドを使用して、イーサネット情報、ドライバのバージョン、および SR-IOV VF の名前を出力できます。

```
lspci | grep -i eth
```

```
[root@cpp-rhel-perf ~]# lspci | grep -i eth
81:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 02)
81:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 02)
81:02.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
81:0a.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
```

Intel NIC のファームウェアに関する詳細については、以下を参照してください。

<https://downloadcenter.intel.com/product/82947/Intel-Ethernet-Controller-X710-Series>

Intel および Cisco NIC のドライバに関する詳細については、以下を参照してください。

<https://downloadcenter.intel.com/download/24411/Intel-Network-Adapter-Driver-for-PCIe-40-Gigabit-Ethernet-Network-Connections-Under-Linux-?product=82947>

Cisco NIC のファームウェアに関する詳細については、以下を参照してください。

<https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c-series-rack-servers/tsd-products-support-series-home.html>

