



Red Hat Virtualization 4.4

Red Hat Virtualization での仮想マシン用 NVIDIA GPU のセットアップ

Red Hat Virtualization での専用 GPU/vGPU を使用した仮想マシンの設定

Red Hat Virtualization 4.4 Red Hat Virtualization での仮想マシン用 NVIDIA GPU のセットアップ

Red Hat Virtualization での専用 GPU/vGPU を使用した仮想マシンの設定

Red Hat Virtualization Documentation Team

Red Hat Customer Content Services

rhev-docs@redhat.com

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、グラフィック処理ユニット (GPU) を備えたホストを使用して Red Hat Virtualization で仮想マシンを実行し、グラフィックを多用するタスクや、GPU なしでは実行できないソフトウェアを実行する方法について説明します。

目次

はじめに	3
第1章 GPU デバイスパススルー: 単一の仮想マシンへのホスト GPU の割り当て	4
1.1. ホスト IOMMU サポートの有効化と NOUVEAU のブラックリストへの登録	4
1.2. ホストから GPU をデタッチ	5
1.3. 仮想マシンへの GPU のアタッチ	6
1.4. 仮想マシンへの GPU ドライバーのインストール	7
1.5. XORG の更新と有効化 (LINUX 仮想マシン)	7
1.6. 仮想マシンからのホスト GPU の削除	8
第2章 仮想 GPU の割り当て	9
2.1. ホストでの NVIDIA vGPU デバイスのセットアップ	9
2.2. 仮想マシンへの vGPU ドライバーのインストール	10
2.3. NVIDIA vGPU デバイスの削除	11
2.4. NVIDIA vGPU のモニタリング	12
2.5. NVIDIA vGPU のリモートデスクトップストリーミングサービス	12
第3章 関連情報	14
付録A 法的通知	15

はじめに

Red Hat Virtualization では、互換性のあるグラフィックスプロセッシングユニット (GPU) を備えたホストを使用して、グラフィックスを多用するタスクや、CAD などの GPU なしでは実行できないソフトウェアの実行に適した仮想マシンを実行できます。

次のいずれかの方法で、GPU を仮想マシンに割り当てることができます。

- **GPU パススルー**: ホスト GPU を単一の仮想マシンに割り当てることができるため、ホストではなく仮想マシンが GPU を使用します。
- **仮想 GPU (vGPU)**: 物理 GPU デバイスを1つ以上の仮想デバイス (**仲介デバイス**と呼ばれる) に分割できます。次に、これらの仲介デバイスを仮想 GPU として1台以上の仮想マシンに割り当てることができます。これらの仮想マシンは、単一の物理 GPU のパフォーマンスを共有します。一部の GPU では、1つのゲストに割り当てることができる仲介デバイスは1つだけです。vGPU は、選択した NVIDIA GPU でのみサポートされます。

例:

ホストには 4 個の GPU があります。GPU ごとに最大 16 個の vGPU、合計 64 個の vGPU をサポートできます。考えられる vGPU の割り当ては次のとおりです。

- 64 個の vGPU を備えた 1 台の仮想マシン
- それぞれ 1 個の vGPU を備えた 64 台の仮想マシン
- それぞれ 1 個の vGPU を備えた 32 台の仮想マシン、それぞれ 2 個の vGPU を備えた 8 台の仮想マシン、それぞれ 4 個の vGPU を備えた 4 台の仮想マシン

第1章 GPU デバイスパススルー: 単一の仮想マシンへのホスト GPU の割り当て

Red Hat Virtualization では、一部の NVIDIA PCIe ベースの GPU デバイスを非 VGA グラフィックスデバイスとして、デバイスパススルーとも呼ばれる PCI VFIO がサポートされています。

標準のエミュレートされたグラフィックスインターフェイスの1つに加え、ホスト GPU を経由して仮想マシンに渡すことにより、1つ以上のホスト GPU を単一の仮想マシンにアタッチできます。仮想マシンは、エミュレートされたグラフィックスデバイスを起動とインストールに使用し、GPU はグラフィックスドライバーがロードされるときに制御を引き継ぎます。

単一の仮想マシンに渡すことができるホスト GPU の正確な数については、NVIDIA の Web サイトを参照してください。

GPU を仮想マシンに割り当てるには、次の手順に従います。

1. [ホストマシンで I/O メモリー管理ユニット \(IOMMU\) を有効化](#) します。
2. [ホストから GPU をデタッチ](#) します。
3. [GPU をゲストにアタッチ](#) します。
4. [ゲストに GPU ドライバーをインストール](#) します。
5. [ゲストで Xorg を設定](#) します。

その手順は以下のとおりです。

前提条件

- GPU デバイスは GPU パススルーモードをサポートしている。
- システムは、[検証済みのサーバーハードウェアプラットフォーム](#)としてリストされている。
- ホストチップセットは Intel VT-d または AMD-Vi をサポートしている。

サポートされているハードウェアとソフトウェアの詳細については、[NVIDIA GPU ソフトウェアリリースノート](#)の [検証済みプラットフォーム](#) を参照してください。

1.1. ホスト IOMMU サポートの有効化と NOUVEAU のブラックリストへの登録

仮想マシンで GPU を使用するには、ホストマシンで I/O メモリー管理ユニット (IOMMU) がサポートされている必要があります。

手順

1. 管理ポータルで [コンピュータ](#) → [ホスト](#) をクリックします。ホストを選択し、**Edit** をクリックします。Edit Hosts ペインが表示されます。
2. Kernel タブをクリックします。
3. Hostdev Passthrough & SR-IOV チェックボックスをオンにします。このチェックボックスは、カーネルコマンドラインに `intel_iommu=on` または `amd_iommu=on` を追加することにより、Intel VT-d または AMD Vi を搭載したホストの IOMMU サポートを有効にします。

4. **Blacklist Nouveau** チェックボックスをオンにします。
5. **OK** をクリックします。
6. ホストを選択し、**管理** → **メンテナンス** をクリックしてから **OK** をクリックします。
7. **Installation** → **Reinstall** をクリックします。
8. 再インストールが完了したら、ホストマシンを再起動します。
9. ホストマシンが再起動したら、**Management** → **Activate** をクリックします。



注記

コマンドラインを使用して IOMMU サポートを有効にするには、仮想マシンの **grub.conf** ファイル (`./entries/rhvh-4.4.<machine id>.conf`) を編集して、オプション **intel_iommu=on** を含めます。

1.2. ホストから GPU をデタッチ

GPU がホストカーネルドライバーにバインドされている場合、GPU を仮想マシンに追加することはできません。そのため、仮想マシンに追加する前に、GPU デバイスとホストのバインドを解除する必要があります。多くの場合、ホストドライバーは GPU の動的なバインド解除をサポートしていないため、ホストドライバーへのバインドからデバイスを手動で除外することをお勧めします。

手順

1. ホストで、**lspci** コマンドを実行して、デバイスのスロット名と ID を識別します。次の例では、NVIDIA Quadro や GRID カードなどのグラフィックスコントローラーが使用されています。

```
# lspci -Dnn | grep -i NVIDIA
0000:03:00.0 VGA compatible controller [0300]: NVIDIA Corporation GK104GL [Quadro
K4200] [10de:11b4] (rev a1)
0000:03:00.1 Audio device [0403]: NVIDIA Corporation GK104 HDMI Audio Controller
[10de:0e0a] (rev a1)
```

出力は、NVIDIA GK104 デバイスがインストールされていることを示しています。グラフィックスコントローラーとオーディオコントローラーがあり、プロパティは以下のとおりです。

- グラフィックスコントローラーのデバイススロット名は **0000:03:00.0**、グラフィックスコントローラーの `vendor-id:device-id` は **10de:11b4** です。
 - オーディオコントローラーのデバイススロット名は **0000:03:00.1**、オーディオコントローラーの `vendor-id:device-id` は **10de:0e0a** です。
2. ホストマシンドライバーが GPU デバイスを使用できないようにします。pci-stub ドライバーでは `vendor-id:device-id` を使用できます。そのためには、`vendor-id:device-id` を値として、`/etc/sysconfig/grub` 設定ファイルにある **GRUB_CMDLINE_LINUX** 環境変数に **pci-stub.ids** オプションを追加します。以下はその例です。

```
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/vg0-lv_swap
rd.lvm.lv=vg0/lv_root rd.lvm.lv=vg0/lv_swap rhgb quiet intel_iommu=on pci-
stub.ids=10de:11b4,10de:0e0a"
```

pci-stub のベンダー ID とデバイス ID を追加する場合は、コンマで区切ります。

3. grub2-mkconfig を使用して、ブートローダー設定を再生成し、この項目を追加します。

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



注記

UEFI ベースのホストを使用する場合、ターゲットファイルは **/etc/grub2-efi.cfg** である必要があります。

4. ホストマシンを再起動します。
5. IOMMU が有効で、ホストデバイスが pci-stub.ids のリストに追加され、Nouveau がブラックリストに登録されていることを確認します。

```
# cat /proc/cmdline
BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-147.el8.x86_64 root=/dev/mapper/vg0-lv_root
ro crashkernel=auto resume=/dev/mapper/vg0-lv_swap rd.lvm.lv=vg0/lv_root
rd.lvm.lv=vg0/lv_swap rhgb quiet intel_iommu=on 1
pci-stub.ids=10de:11b4,10de:0e0a 2
rdblacklist=nouveau 3
```

- 1 IOMMU が有効になっている。
- 2 ホストデバイスが pci-stub.ids のリストに追加されている。
- 3 Nouveau がブラックリストに登録されている。

1.3. 仮想マシンへの GPU のアタッチ

GPU とホストカーネルドライバーのバインドを解除した後、GPU を仮想マシンに追加して、正しいドライバーを有効にできます。

手順

1. [仮想マシン管理ガイド](#) の [仮想マシンへのホストデバイスの追加](#) に記載されている手順に従います。
2. 仮想マシンを実行してログインします。
3. NVIDIA GPU ドライバーを仮想マシンにインストールします。
4. **lspci -nnk** コマンドを使用して、GPU に正しいカーネルドライバーが使用されていることを確認します。以下はその例です。

```
# lspci -nnk
00:07.0 VGA compatible controller [0300]: NVIDIA Corporation GK104GL [Quadro K4200]
[10de:11b4] (rev a1)
Subsystem: Hewlett-Packard Company Device [103c:1096]
Kernel driver in use: nvidia
Kernel modules: nouveau, nvidia_drm, nvidia
```

1.4. 仮想マシンへの GPU ドライバーのインストール

手順

1. 仮想マシンを実行し、VNC または SPICE コンソールを使用して仮想マシンに接続します。
2. ドライバーを仮想マシンにダウンロードします。ドライバーの入手方法については、[NVIDIA Web サイトのドライバーページ](#) を参照してください。
3. GPU ドライバーをインストールします。



重要

Linux のみ: Linux ゲストオペレーティングシステムにドライバーをインストールする場合、xorg.conf を更新するように求められます。インストール中に xorg.conf を更新しない場合は、手動で更新する必要があります。

4. ドライバーのインストールが完了したら、マシンを再起動します。**Windows 仮想マシンの場合**、ゲストオペレーティングシステム内からではなく、管理ポータルまたは仮想マシンポータルからゲストの電源を完全にオフにします。



重要

Windows のみ: Windows ゲストオペレーティングシステム内から仮想マシンの電源をオフにすると、仮想マシンがハイバーネートモードになることがあります。その場合、メモリーは完全にクリアされず、その後の問題が発生する可能性があります。管理ポータルまたは仮想マシンポータルを使用して仮想マシンの電源をオフにすると、仮想マシンは強制的にメモリーを完全にクリーンアップします。

5. モニターをホスト GPU 出力インターフェイスに接続し、仮想マシンを実行します。
6. 各 vGPU に NVIDIA vGPU ゲストソフトウェアライセンスを設定し、NVIDIA コントロールパネルにライセンス認証情報を追加します。詳細については、[NVIDIA 仮想 GPU ソフトウェアドキュメントの How NVIDIA vGPU Software Licensing Is Enforced](#) を参照してください。

1.5. XORG の更新と有効化 (LINUX 仮想マシン)

仮想マシンで GPU を使用する前に、仮想マシンで xorg を更新して有効にする必要があります。NVIDIA ドライバーのインストールは、これを自動的に実行します。`/etc/X11/xorg.conf` を表示して、xorg が更新および有効化されているか確認します。

```
# cat /etc/X11/xorg.conf
```

最初の 2 行は、NVIDIA によって生成されたかどうかを示しています。以下はその例です。

```
# cat /etc/X11/xorg.conf
# nvidia-xconfig: X configuration file generated by nvidia-xconfig
# nvidia-xconfig: version 390.87 (buildmeister@swio-display-x64-rhel04-14) Tue Aug 21 17:33:38 PDT 2018
```

手順

1. 仮想マシンで、次のコマンドを使用して **xorg.conf** ファイルを生成します。

```
# X -configure
```

2. 次のコマンドを使用して、**xorg.conf** ファイルを **/etc/X11/xorg.conf** にコピーします。

```
# cp /root/xorg.conf.new /etc/X11/xorg.conf
```

3. 仮想マシンを再起動します。
4. **/etc/X11/xorg.conf** を表示して、xorg が更新および有効化されているか検証します。

```
# cat /etc/X11/xorg.conf
```

Device セクションを検索します。次のようなエントリーが表示されます。

```
Section "Device"
    Identifier "Device0"
    Driver      "nvidia"
    VendorName  "NVIDIA Corporation"
EndSection
```

これで、GPU が仮想マシンに割り当てられました。

1.6. 仮想マシンからのホスト GPU の削除

- 仮想マシンからホスト GPU を削除する方法については、[仮想マシン管理ガイド](#)の [仮想マシンからのホストデバイスの削除](#) を参照してください。

第2章 仮想 GPU の割り当て

NVIDIA vGPU デバイスをセットアップするには、以下を行う必要があります。

1. GPU デバイスに適した NVIDIA vGPU ドライバーを入手してインストールします。
2. 仲介デバイスを作成します。
3. 各仲介デバイスを仮想マシンに割り当てます。
4. 各仮想マシンにゲストドライバーをインストールします。

以下で、この手順について説明します。

2.1. ホストでの NVIDIA vGPU デバイスのセットアップ



注記

ゲストオペレーティングシステムに NVIDIA vGPU ドライバーをインストールする前に、ライセンス要件を理解し、正しいライセンス認証情報を取得する必要があります。

前提条件

- GPU デバイスが仮想 GPU (vGPU) 機能をサポートしている。
- システムは、検証済みサーバーハードウェアプラットフォームとしてリストされている。

サポートされている GPU と検証済みプラットフォームの詳細については、www.nvidia.com で [NVIDIA vGPU CERTIFIED SERVERS](#) を参照してください。

手順

1. NVIDIA-vGPU ドライバーをダウンロードしてインストールします。ドライバーの入手方法については、[NVIDIA Web サイトの vGPU ドライバーページ](#) を参照してください。ドライバーをダウンロードするには、Nvidia エンタープライズアカウントが必要です。アカウントがない場合は、ハードウェアベンダーにお問い合わせください。
2. Nvidia Web サイトからダウンロードしたファイルを解凍し、ホストにコピーしてドライバーをインストールします。
3. NVIDIA ソフトウェアインストーラーが `/etc/modprobe.d/nvidia-installer-disable-nouveau.conf` ファイルを作成しなかった場合は、手動で作成します。
4. テキストエディターで `/etc/modprobe.d/nvidia-installer-disable-nouveau.conf` ファイルを開き、ファイルの最後に次の行を追加します。

```
blacklist nouveau
options nouveau modeset=0
```

5. 現在のカーネル用に初期 ramdisk を再生成してから再起動します。

```
# dracut --force
# reboot
```

仲介デバイスで、以前対応していたカーネルバージョンを使用する必要がある場合は、インストールしたすべてのカーネルバージョンに対して初期 ramdisk を再生成することもできます。

```
# dracut --regenerate-all --force
# reboot
```

- カーネルが **nvidia_vgpu_vfio** モジュールをロードしたことを確認します。

```
# lsmod | grep nvidia_vgpu_vfio
```

- nvidia-vgpu-mgr.service** サービスが実行されていることを確認します。

```
# systemctl status nvidia-vgpu-mgr.service
```

以下はその例です。

```
# lsmod | grep nvidia_vgpu_vfio
nvidia_vgpu_vfio 45011 0
nvidia 14333621 10 nvidia_vgpu_vfio
mdev 20414 2 vfio_mdev,nvidia_vgpu_vfio
vfio 32695 3 vfio_mdev,nvidia_vgpu_vfio,vfio_iommu_type1
# systemctl status nvidia-vgpu-mgr.service
nvidia-vgpu-mgr.service - NVIDIA vGPU Manager Daemon
   Loaded: loaded (/usr/lib/systemd/system/nvidia-vgpu-mgr.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2018-03-16 10:17:36 CET; 5h 8min ago
   Main PID: 1553 (nvidia-vgpu-mgr)
   [...]

```

- 管理ポータルで **Compute** → **Virtual Machines** をクリックします。
- 仮想マシンの名前をクリックして、詳細ビューに移動します。
- Host Devices** タブをクリックします。
- Manage vGPU** をクリックします。 **Manage vGPU** ダイアログボックスが開きます。
- vGPU タイプと、この仮想マシンで使用するインスタンスの数を選択します。
- Secondary display adapter for VNC** で **On** を選択し、vGPU の他に、コンソールのプライマリグラフィックスアダプターとして 2 番目のエミュレートされた QXL または VGA グラフィックスアダプターを追加します。



注記

クラスターレベル 4.5 以降では、vGPU が使用され、**Secondary display adapter for VNC** が **On** に設定されている場合、追加のフレームバッファードisplay デバイスが仮想マシンに自動的に追加されます。これにより、vGPU が初期化される前に、空白の画面ではなく仮想マシンコンソールを表示できます。

- Save** をクリックします。

2.2. 仮想マシンへの vGPU ドライバーのインストール

手順

1. 仮想マシンを実行し、VNC コンソールを使用して仮想マシンに接続します。



注記

SPICE は vGPU ではサポートされていません。

2. ドライバーを仮想マシンにダウンロードします。ドライバーの入手方法については、[NVIDIA Web サイトのドライバーページ](#)を参照してください。
3. [NVIDIA 仮想 GPU ソフトウェアドキュメントの Installing the NVIDIA vGPU Software Graphics Driver](#)に記載された手順に従って、vGPU ドライバーをインストールします。



重要

Linux のみ: Linux ゲストオペレーティングシステムにドライバーをインストールする場合、xorg.conf を更新するように求められます。インストール中に xorg.conf を更新しない場合は、手動で更新する必要があります。

4. ドライバーのインストールが完了したら、マシンを再起動します。**Windows 仮想マシンの場合**、ゲストオペレーティングシステム内からではなく、管理ポータルまたは仮想マシンポータルからゲストの電源を完全にオフにします。



重要

Windows のみ: Windows ゲストオペレーティングシステム内から仮想マシンの電源をオフにすると、仮想マシンがハイバーネートモードになることがあります。その場合、メモリーは完全にクリアされず、その後の問題が発生する可能性があります。管理ポータルまたは仮想マシンポータルを使用して仮想マシンの電源をオフにすると、仮想マシンは強制的にメモリーを完全にクリーンアップします。

5. 仮想マシンを実行し、Mechdyne TGX などのサポートされているリモートデスクトッププロトコルの1つを使用して仮想マシンに接続し、NVIDIA コントロールパネルを開いて vGPU が認識されていることを確認します。Windows では、代わりに Windows Device Manager を開くこともできます。vGPU は、**Display adapters** の下に表示されます。詳細については、[NVIDIA 仮想 GPU ソフトウェアドキュメントの NVIDIA vGPU Software Graphics Driver](#)を参照してください。
6. 各 vGPU に NVIDIA vGPU ゲストソフトウェアライセンスを設定し、NVIDIA コントロールパネルにライセンス認証情報を追加します。詳細については、[NVIDIA 仮想 GPU ソフトウェアドキュメントの How NVIDIA vGPU Software Licensing Is Enforced](#)を参照してください。

2.3. NVIDIA vGPU デバイスの削除

割り当てられた vGPU 仲介デバイスの設定を変更する場合は、割り当てられたゲストから既存のデバイスを削除する必要があります。

手順

1. 管理ポータルで **Compute → Virtual Machines** をクリックします。
2. 仮想マシンの名前をクリックして、詳細ビューに移動します。

3. **Host Devices** タブをクリックします。
4. **Manage vGPU** をクリックします。 **Manage vGPU** ダイアログボックスが開きます。
5. **Selected vGPU Type Instances** の横にある **x** ボタンをクリックして、仮想マシンから vGPU をデタッチします。
6. **SAVE** をクリックします。

2.4. NVIDIA vGPU のモニタリング

NVIDIA vGPUS の場合、物理 GPU と vGPU に関する情報を取得するには、ホストで **nvidia-smi** コマンドを入力して NVIDIA システム管理インターフェイスを使用できます。詳細については、**NVIDIA 仮想 GPU ソフトウェアドキュメント** の [NVIDIA System Management Interface nvidia-smi](#) を参照してください。

以下はその例です。

```
# nvidia-smi
Thu Nov 1 17:40:09 2018
+-----+
| NVIDIA-SMI 410.62                Driver Version: 410.62          |
+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
| 0  Tesla M60      On      | 00000000:84:00.0 Off  |          Off         |
| N/A   40C    P8   24W / 150W | 1034MiB / 8191MiB |    0%    Default  |
+-----+-----+-----+-----+-----+
| 1  Tesla M60      On      | 00000000:85:00.0 Off  |          Off         |
| N/A   33C    P8   23W / 150W | 8146MiB / 8191MiB |    0%    Default  |
+-----+-----+-----+-----+-----+
| 2  Tesla M60      On      | 00000000:8B:00.0 Off  |          Off         |
| N/A   34C    P8   24W / 150W | 8146MiB / 8191MiB |    0%    Default  |
+-----+-----+-----+-----+-----+
| 3  Tesla M60      On      | 00000000:8C:00.0 Off  |          Off         |
| N/A   45C    P8   24W / 150W | 18MiB / 8191MiB   |    0%    Default  |
+-----+-----+-----+-----+-----+

+-----+
| Processes:                         GPU Memory |
| GPU   PID  Type  Process name      Usage   |
+-----+-----+-----+-----+-----+
| 0    34432  C+G  vgpu              508MiB |
| 0    34718  C+G  vgpu              508MiB |
| 1    35032  C+G  vgpu              8128MiB|
| 2    35032  C+G  vgpu              8128MiB|
+-----+-----+-----+-----+-----+
```

2.5. NVIDIA vGPU のリモートデスクトップストリーミングサービス

以下のリモートデスクトップストリーミングサービスは、RHEL 8 の NVIDIA vGPU 機能での使用が正常にテストされています。

- HP-RGS
- Mechdyne TGX - 現在、Windows Server 2016 ゲストで Mechdyne TGX は使用できません。
- NICE DCV - このストリーミングサービスを使用する場合は、解像度を動的にすると、場合によっては画面が黒くなるため、解像度を固定する設定を使用してください。

第3章 関連情報

KVM を使用した RHEL で NVIDIA vGPU を使用方法については、以下を参照してください。

- [NVIDIA GPU ソフトウェアリリースノート](#)。
- <https://docs.nvidia.com> の **NVIDIA 仮想 GPU ソフトウェアドキュメント**。

付録A 法的通知

Copyright © 2022 Red Hat, Inc.

Licensed under the ([Creative Commons Attribution–ShareAlike 4.0 International License](#)). Derived from documentation for the ([oVirt Project](#)). If you distribute this document or an adaptation of it, you must provide the URL for the original version.

Modified versions must remove all Red Hat trademarks.

Red Hat, Red Hat Enterprise Linux, the Red Hat logo, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.