



# Настройка NVIDIA GPU для виртуальной машины в РЕД Виртуализации

# Оглавление

<b>1</b>	<b>Назначение физического GPU виртуальной машине .....</b>	<b>4</b>
1.1	Включение поддержки IOMMU на хосте и внесение NOUVEAU в черный список.....	5
1.2	Отключение GPU от хоста .....	5
1.3	Подключение GPU к виртуальной машине.....	6
1.4	Установка драйвера GPU на виртуальную машину.....	7
1.5	Удаление GPU хоста из виртуальной машины .....	7
<b>2</b>	<b>Назначение виртуальных GPU (vGPU) .....</b>	<b>8</b>
2.1	Настройка устройства NVIDIA GPU на хосте .....	8
2.2	Установка драйвера vGPU на виртуальную машину .....	10
2.3	Удаление устройств NVIDIA vGPU .....	11
2.4	Мониторинг NVIDIA vGPU .....	11

# Введение

В рамках данного документа рассмотрено использование хоста с графическим процессором (GPU) для запуска виртуальных машин и программного обеспечения, которое не может работать без GPU.

Назначить графический процессор для виртуальной машины можно двумя способами:

- проброс GPU — назначение GPU хоста только одной виртуальной машине.
- виртуальный GPU (vGPU) — предусматривает разделение физического устройства GPU на одно или несколько виртуальных устройств, называемых устройствами-посредниками. После этого они могут быть назначены одной или несколькими виртуальным машинам в качестве vGPU. Таким образом одно физическое GPU может быть совместно использовано несколькими VM.

**Примечание.** Обратите внимание, что для некоторых GPU только одно устройство-посредник может быть назначено одной гостевой системе. Поддержка vGPU доступна только на некоторых NVIDIA GPU.

Рассмотрим пример: на хосте имеется четыре GPU. Каждый GPU может поддерживать до 16 vGPU — в общей сложности имеется 64 vGPU. В такой ситуации возможны следующие варианты назначения vGPU:

- одна VM имеет 64 vGPU;
- 64 VM, каждая из которых имеет по одному vGPU;
- 32 VM, каждая из которых имеет по одному vGPU, 8 VM — с двумя vGPU, 4 VM — с четырьмя vGPU.

# 1. Назначение физического GPU

РЕД Виртуализация поддерживает технологию PCI VFIO, также называемую пробросом устройств, для некоторых устройств GPU на базе NVIDIA PCIe, не имеющих разъема VGA.

Можно подключить один или несколько графических процессоров хоста к одной виртуальной машине, пробрасывая GPU в дополнение к одному из стандартных эмулируемых графических интерфейсов. Виртуальная машина использует эмулированное графическое устройство для предварительной загрузки и установки, а GPU берет управление на себя, когда загружаются его графические драйверы.

Информацию о точном количестве хостовых GPU, которое можно передать одной виртуальной машине, можно найти на сайте NVIDIA.

Чтобы пробросить GPU на виртуальную машину, необходимо выполнить следующий алгоритм действий:

1. Включить I/O Memory Management Unit (IOMMU) на хостовой машине.
2. Отключить GPU от хоста.
3. Подключить GPU к виртуальной машине.
4. Установить драйверы GPU на виртуальную машину.
5. Настроить **Xorg** в гостевой системе (только для Linux).

Требования:

- Устройство GPU поддерживает режим проброса устройства;
- Гостевая система входит в список проверенных серверных аппаратных платформ;
- Чипсет хоста поддерживает Intel VT-d или AMD-Vi.

Более подробную информацию о поддерживаемом оборудовании и ПО смотрите в разделе «Утвержденные платформы» в Информации о выпуске ПО NVIDIA GPU.

## 1.1 Включение поддержки IOMMU на хосте и внесение NOUVEAU в черный список

Поддержка I/O Memory Management Unit (IOMMU) на хостовой машине необходима для использования GPU на виртуальной машине.

Для включения поддержки IOMMU выполните следующие действия:

1. На Портале администрирования нажмите **Виртуализация — Узлы**. Выберите хост и нажмите **«Изменить»**. Откроется панель **«Редактирование хоста»**.
2. Перейдите на вкладку **«Ядро»**.
3. Установите флажок **Проброс устройств узла и SR-IOV**. Этот флажок включает поддержку IOMMU для хоста с Intel VT-d или AMD Vi путем добавления **intel\_iommu=on** или **amd\_iommu=on** в командную строку ядра.
4. Установите флажок **Черный список Nouveau**.
5. Нажмите **ОК**.
6. Выберите хост, нажмите **Управление — Обслуживание**, затем нажмите **ОК**.
7. Нажмите **Установка — Переустановить**.
8. После завершения переустановки выполните перезагрузку хостовой машины.
9. После перезагрузки хостовой машины нажмите **Управление — Включить**.

**Примечание.** Чтобы включить поддержку IOMMU с помощью командной строки, отредактируйте файл **grub.conf** виртуальной машины, включив в него параметр **intel\_iommu=on**.

## 1.2 Отключение GPU от хоста

Если GPU привязан к хостовому драйверу ядра, вы не сможете добавить его в виртуальную машину, поэтому сначала необходимо отвязать устройство GPU от хоста.

Драйверы хоста зачастую не поддерживают динамическое отвязывание GPU, поэтому рекомендуется вручную исключить устройство из привязки к драйверам хоста. Для этого выполните следующие действия:

1. На хосте необходимо определить имя слота устройства и его идентификаторы, выполнив команду **lspci**.

```
# 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)
```

Вывод показывает, что установлено устройство NVIDIA GK104. Оно имеет графический контроллер и аудиоконтроллер со следующими свойствами:

- имя слота устройства графического контроллера **0000:03:00.0**, а vendor-id:device-id для графического контроллера **10de:11b4**.
- имя слота устройства аудиоконтроллера **0000:03:00.1**, а vendor-id:device-id для аудиоконтроллера - **10de:0e0a**.



2. Запретите драйверу хостовой машины использовать устройство GPU. Вы можете использовать vendor-id:device-id с драйвером **pci-stub**. Для этого добавьте параметр **pci-stub.ids** со значением vendor-id:device-id в переменную окружения **GRUB\_CMDLINE\_LINUX**, расположенную в каталоге **/etc/default/grub**, например:

```
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"
```

При добавлении дополнительных идентификаторов производителя и устройств для **pci-stub** разделяйте их запятой.

3. Обновите конфигурацию загрузчика с помощью **grub2-mkconfig**, чтобы включить эту опцию:

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

**Примечание.** При использовании хоста на базе UEFI изменения должны вноситься в файл **/boot/efi/EFI/redos/grub.cfg**.

4. Перезагрузите хостовую машину.

5. Убедитесь, что IOMMU включен, устройство хоста добавлено в список **pci-stub.ids**, а nouveau занесен в черный список:

```
# 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 # IOMMU включен pci-stub.ids=10de:11b4,10de:0e0a # хост-устройство добавлено в список pci-stub.ids rdblacklist=nouveau # nouveau занесен в черный список
```

## 1.3 Подключение GPU к виртуальной машине

После отключения GPU от хостового драйвера ядра вы можете добавить его в виртуальную машину и включить нужный драйвер.

Для этого выполните следующие действия:

1. Выполните установку хост-устройства на виртуальную машину, описанную в разделе 6.8.1 «Добавление хост-устройства к виртуальной машине» в Руководстве по администрированию виртуальных машин РЕД Виртуализации.

2. Запустите виртуальную машину и авторизуйтесь под своей учетной записью.

3. Установите драйвер NVIDIA GPU на виртуальную машину.

4. Проверьте, что для GPU используется подходящий драйвер ядра с помощью команды **lspci -nnk**. Например:

```
# 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 Установка драйвера GPU на виртуальную машину

Для установки драйвера GPU на виртуальную машину выполните следующие действия:

1. Запустите виртуальную машину и подключитесь к ней с помощью консоли VNC или SPICE.
2. Загрузите драйвер на виртуальную машину. Информацию о получении драйвера смотрите на странице «Драйверы» на сайте NVIDIA.
3. Установите драйвер GPU.

**Важно!** Только для Linux. При установке драйвера на гостевую операционную систему Linux вам будет предложено обновить **xorg.conf**. Если вы не обновили **xorg.conf** во время установки, вам необходимо обновить его вручную. ■

4. После завершения установки драйвера перезагрузите машину. Для виртуальных машин Windows полностью выключите питание машины с Портала администрирования или Портала виртуальных машин, а не из гостевой операционной системы.

**Важно!** Только для Windows. Выключение питания виртуальной машины из гостевой операционной системы Windows иногда переводит виртуальную машину в спящий режим, который не полностью очищает память, что может привести к последующим проблемам. Использование *Портала администрирования* или *Портала виртуальных машин* для выключения виртуальной машины заставляет ее полностью очистить память. ■

5. Подключите монитор к выходному интерфейсу GPU хоста и запустите виртуальную машину.

6. Настройте лицензирование гостевого ПО NVIDIA vGPU для каждого vGPU и добавьте учетные данные лицензии в панели управления NVIDIA. Дополнительную информацию смотрите в разделе «Как осуществляется лицензирование ПО NVIDIA vGPU» в документации по ПО NVIDIA Virtual GPU.

## 1.5 Удаление GPU хоста из виртуальной машины

Информацию об удалении хостового GPU из виртуальной машины можно найти в разделе 6.8.2 «Удаление хост-устройств с виртуальной машины» в Руководстве по администрированию виртуальных машин РЕД Виртуализации.

## 2. Назначение виртуальных GPU (vGPU)

Для настройки устройств NVIDIA vGPU необходимо выполнить следующий алгоритм действий:

1. Получить и установить подходящий драйвер NVIDIA vGPU для используемого GPU-устройства.
2. Создать устройство-посредника.
3. Назначить каждое устройство-посредника виртуальной машине.
4. Установить гостевые драйверы на каждую виртуальную машину.

Требования:

- Устройство GPU поддерживает функцию Virtual GPU (vGPU);
- Гостевая система входит в список проверенных серверных аппаратных платформ.

Более подробную информацию о поддерживаемом оборудовании и ПО смотрите в разделе «Утвержденные платформы» в Информации о выпуске ПО NVIDIA GPU.

### 2.1 Настройка устройства NVIDIA GPU на хосте

**Примечание.** Перед установкой драйвера NVIDIA vGPU в гостевой ОС необходимо определить требования лицензирования и получить корректные учетные данные лицензии.

Для настройки NVIDIA GPU на хосте выполните следующие действия:

1. Скачайте и установите драйвер NVIDIA-vGPU. Информацию о получении драйвера смотрите на странице «Драйверы vGPU» на сайте NVIDIA. Для загрузки драйверов требуется корпоративная учетная запись Nvidia. Если она недоступна, обратитесь к поставщику оборудования.



2. Распакуйте загруженный файл с сайта Nvidia и скопируйте его на хост для установки драйвера.

3. Если программа установки ПО NVIDIA не создала файл `/etc/modprobe.d/nvidia-installer-disable-nouveau.conf`, создайте его вручную.

4. Откройте файл `/etc/modprobe.d/nvidia-installer-disable-nouveau.conf` в текстовом редакторе и добавьте следующие строки в конец файла:

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

5. Обновите начальный **ramdisk** для текущего ядра, затем перезагрузите систему:

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

В качестве альтернативы, если необходимо использовать предыдущую поддерживаемую версию ядра с устройствами-посредниками, обновите начальный ramdisk для всех установленных версий ядра:

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

6. Проверьте, что в ядро загрузился модуль **nvidia\_vgpu\_vfio**:

```
# lsmod | grep nvidia_vgpu_vfio
```

7. Проверьте, запущена ли служба **nvidia-vgpu-mgr.service**:

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

Например:

```
# 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)
[...]
```

8. На Портале администрирования нажмите **Виртуализация — Виртуальные машины**.

9. Нажмите на имя виртуальной машины, чтобы открыть подробные сведения.

10. Перейдите на вкладку **Устройства хоста**.

11. Нажмите **Менеджер vGPU**. Откроется диалоговое окно **Менеджер управления vGPU**.

12. Выберите *тип vGPU* и *количество* экземпляров, которые вы хотите использовать с этой виртуальной машиной.

13. Выберите **On** для параметра **Второй графический адаптер для VNC**, чтобы добавить второй эмулированный графический адаптер QXL или VGA в качестве основного графического адаптера для консоли в дополнение к vGPU.

**Примечание.** Данная опция позволяет отображать консоль виртуальной машины до инициализации vGPU вместо пустого экрана.

14. Нажмите кнопку **«Сохранить»**.

## 2.2 Установка драйвера vGPU на виртуальную машину

Для установки драйвера vGPU на виртуальную машину выполните следующие действия:

1. Запустите виртуальную машину и подключитесь к ней с помощью консоли VNC.

**Примечание.** Консоль SPICE не поддерживается на vGPU.

2. Загрузите драйвер на виртуальную машину. Информацию о получении драйвера смотрите на странице «Драйверы» на сайте NVIDIA.

3. Установите драйвер vGPU, следуя инструкциям в разделе «Установка программного графического драйвера NVIDIA vGPU» в документации по программному обеспечению NVIDIA Virtual GPU.

**Важно!** Только для Linux. При установке драйвера на гостевую операционную систему Linux вам будет предложено обновить **xorg.conf**. Если вы не обновили **xorg.conf** во время установки, вам необходимо обновить его вручную. ■

4. После завершения установки драйвера перезагрузите машину. Для виртуальных машин Windows полностью выключите питание машины с *Портала администрирования* или *Портала виртуальных машин*, а не из гостевой операционной системы.

**Важно!** Только для Windows. Выключение питания виртуальной машины из гостевой операционной системы Windows иногда переводит виртуальную машину в спящий режим, который не полностью очищает память, что может привести к последующим проблемам. Использование *Портала администрирования* или *Портала виртуальных машин* для выключения виртуальной машины заставляет ее полностью очистить память. ■

5. Запустите виртуальную машину и подключитесь к ней с помощью одного из поддерживаемых протоколов удаленного рабочего стола. Проверьте, что vGPU распознан, открыв панель управления NVIDIA.

В Windows можно также открыть диспетчер устройств Windows. vGPU должен появиться в разделе «Адаптеры дисплея». Для получения дополнительной информации см. в разделе «Установка программного графического драйвера NVIDIA vGPU» в документации по программному обеспечению NVIDIA Virtual GPU.

6. Настройте лицензирование гостевого ПО NVIDIA vGPU для каждого vGPU и добавьте учетные данные лицензии в панели управления NVIDIA. Дополнительную информацию смотрите в разделе «Как осуществляется лицензирование ПО NVIDIA vGPU» в документации по ПО NVIDIA Virtual GPU.

## 2.3 Удаление устройств NVIDIA vGPU

Чтобы изменить конфигурацию назначенных устройств-посредников vGPU, необходимо удалить существующие назначенные устройства из гостевых ОС.

Для удаления назначенных устройств из виртуальных машин выполните следующие действия:

1. На портале администрирования нажмите **Виртуализация - Виртуальные машины**.
2. Нажмите на имя виртуальной машины, чтобы открыть подробные сведения.
3. Перейдите на вкладку **Устройства хоста**.
4. Нажмите **Менеджер vGPU**. Откроется диалоговое окно **Менеджер управления vGPU**.
5. Нажмите кнопку «X» рядом с **Selected vGPU Type Instances**, чтобы отсоединить vGPU от виртуальной машины.
6. Нажмите кнопку **Сохранить**.

## 2.4 Мониторинг NVIDIA vGPU

Чтобы получить информацию о физическом GPU и vGPU, вы можете использовать интерфейс управления системой NVIDIA, выполнив команду **nvidia-smi** на хосте.

Дополнительную информацию см. в разделе «Интерфейс управления системой NVIDIA nvidia-smi» в документации NVIDIA Virtual GPU Software Documentation.

Например:

```
# 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
+-----+-----+-----+
```

```

| 1 Tesla M60 On | 00000000:85:00.0 Off | Off |
| N/A 33C P8 23W / 150W | 8146MiB / 8191MiB | 0
+-----+-----+-----+
| 2 Tesla M60 On | 00000000:8B:00.0 Off | Off |
| N/A 34C P8 24W / 150W | 8146MiB / 8191MiB | 0
+-----+-----+-----+
| 3 Tesla M60 On | 00000000:8C:00.0 Off | Off |
| N/A 45C P8 24W / 150W | 18MiB / 8191MiB | 0
+-----+-----+-----+

+-----+
| 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 |
+-----+

```