Настройка прямого проброса видеокарты NVIDIA в виртуальную машину (zVirt 3.X)

Аннотация

В этом документе описывается, как использовать хост с графическим процессором (GPU) для запуска виртуальных машин в zVirt для выполнения графически требовательных задач и программного обеспечения, которое не может работать без GPU.

1. `nouveau` - проект по созданию свободных драйверов для видеокарт компании Nvidia с поддержкой ускорения трёхмерной графики.
2. GPU -отдельное устройство персонального компьютера или игровой приставки, выполняющее графический рендеринг.
3. `IOMMU` - блок управления памятью для операций ввода-вывода. Так же как традиционный, процессорный блок управления памятью, который переводит виртуальные адреса, видимые процессором в физические, этот блок занимается трансляцией виртуальных адресов, видимых аппаратным устройством, в физические адреса.

1. Предисловие

Вы можете использовать хост с совместимым графическим процессором (GPU) для запуска виртуальных машин в zVirt, которые подходят для выполнения графически интенсивных задач и для запуска программного обеспечения, которое не может работать без GPU, например, CAD.

Вы можете назначить GPU виртуальной машине одним из следующих способов:

- **GPU passthrough**: можно назначить GPU хоста одной виртуальной машине, чтобы виртуальная машина, а не хост, использовала GPU.
- **Virtual GPU (vGPU)**: Вы можете разделить физическое устройство GPU на одно или несколько виртуальных устройств, называемых опосредованными устройствами. Затем эти опосредованные устройства можно назначить одной или нескольким виртуальным машинам в качестве виртуальных GPU. Эти виртуальные машины совместно используют производительность одного физического GPU. Для некоторых GPU только одно опосредованное устройство может быть назначено одному гостю. Поддержка vGPU доступна только для некоторых NVIDIA GPU.

В данном руководстве описан режим прямого проброса видеокарты в виртуальную машину.

2. GPU device passthrough: Назначение графического процессора хоста одной виртуальной машине

zVirt поддерживает `PCI VFIO`, также называемое `device passthrough`, для некоторых GPU на базе **NVIDIA PCIe** в качестве **не-VGA** графических устройств.

Вы можете подключить один графический процессор хоста к одной виртуальной машине, передавая виртуальной машине графический процессор хоста 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.

Чтобы назначить GPU виртуальной машине, выполните следующие действия:

1. Включите `I/O Memory Management Unit (IOMMU)`, заблокируйте `nouveau` на хост-машине.
2. Отсоедините GPU от хоста.
3. Присоедините GPU к виртуальной машине.
4. Установите драйвер GPU на виртуальную машину.

Эти шаги подробно описаны ниже.

Необходимые условия:

- Ваше устройство GPU поддерживает режим `GPU passthrough`.
- Ваша система входит в список проверенных серверных аппаратных платформ.
- Чипсет хоста поддерживает `Intel VT-d` или `AMD-Vi`.

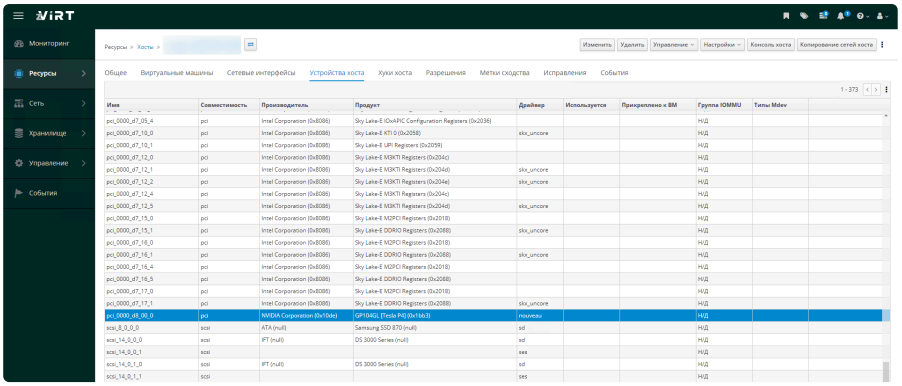
Более подробную информацию о поддерживаемом оборудовании и ПО смотрите Поддерживаемые платформы в информации о выпуске ПО NVIDIA GPU.

2.1. Включение поддержки IOMMU на хосте, внесение nouveau в черный список.

Поддержка `I/O Memory Management Unit (IOMMU)` на хосте необходима для использования GPU на виртуальной машине. Последовательность действий:

1. Убедитесь, что видеокарта корректно отображается на хосте `zVirt`. Например, имя этого хоста **host-gpu**. Для этого последовательно перейдите **Ресурсы > Хосты > host-**

gpu > Устройства Хоста.



2. Перевести хост **host-gpu** в обслуживание. Для этого последовательно перейдите **Ресурсы > Хосты > host-gpu > Управление > Обслуживание**.
3. Подключитесь по SSH к данному хосту. Выполните вывод содержимого файла **/proc/cmdline**. Пример вывода файла:

```
[root@server ~]cat /proc/cmdline
OOT_IMAGE=(hd0,msdos1)//zvirt-node-ng-3.0-0.20220410.0+1/vmlinuz-4.18.0-373.el8.x86_64 crashkernel=auto resume=/dev/mapper/znn-swap rd.lvm.lv=znn/zvirt-node-ng-3.0-0.20220410.0+1 rd.lvm.lv=znn/swap rhgb quiet root=/dev/znn/zvirt-node-ng-3.0-0.20220410.0+1 boot=UUID=d802e2a5-b7d8-4c6a-b24a-625bfd971da7 rootflags=discard img.bootid=zvirt-node-ng-3.0-0.20220410.0+1
```

4. Выполните команду для внесения `nouveau` в список заблокированных драйверов.

```
grubby --update-kernel=ALL --args="rd.driver.blacklist=nouveau"
```

5. Выполните команду, которая позволит добавлять физические устройства в гостевые ОС: режим **Passthrough устройств хоста и SR-IOV**. Для процессоров на базе Intel и AMD команды будут отличаться.

   Intel

   ```
   grubby --update-kernel=ALL --args="intel_iommu=on"
   ```

   AMD

   ```
   grubby --update-kernel=ALL --args="amd_iommu=on"
   ```

2.2. Отсоединение GPU от хоста.

Вы не можете добавить GPU в виртуальную машину, если GPU привязан к драйверу ядра хоста, поэтому перед добавлением в виртуальную машину необходимо отвязать устройство GPU от хоста. Для этого выполните следующие действия:

1. На хосте определите имя слота устройства и идентификаторы устройства (ID карты NVIDIA), выполнив команду `lspci`. Для работы команды может потребоваться пакет `pciutils`. Пример выполнения команды:

```
[root@server ~]# lspci -Dnn | grep -i nvidia
0000:d8:00.0 3D controller [0302]: NVIDIA Corporation GP104GL [Tesla P4]
[10de:1bb3] (rev a1)
0000:d8:00.1 Audio device [0403]: NVIDIA Corporation Device [10de:10fa] (rev
a1)
```

   Вывод показывает, что установлено устройство `NVIDIA Corporation GP104GL [Tesla P4]`. Оно имеет графический контроллер и аудиоконтроллер со следующими свойствами:

   - Имя слота устройства графического контроллера `0000:d8:00.0`, `vendor-id:device-id` для графического контроллера `10de:1bb3`.
   - Имя слота аудио контроллера `0000:d8:00.1`, а `vendor-id:device-id` для аудио контроллера `10de:10fa`.

   Идентификаторами устройств будут является значения `10de:1bb3` и `10de:10fa`. Запишите данные значения, они будут необходимы при выполнении следующего шага.

2. Выполните команду для изоляции устройств и дальнейшего использования их в ВМ. Синтаксис команды `grubby --update-kernel=ALL --args="ID1,ID2" `. Например:

```
grubby --update-kernel=ALL --args="pci-stub.ids=10de:1bb3,10de:10fa"
```

3. Перезапустите хост. Для этого последовательно перейдите **Ресурсы > Хосты > host-gpu > Управление > Перезапустить**.

4. После перезапуска хоста, проверьте, что в файле `/proc/cmdline` появились добавленные ранее данные. Например:

```
[root@server ~]cat /proc/cmdline
BOOT_IMAGE=(hd0,msdos1)//zvirt-node-ng-3.0-0.20220410.0+1/vmlinuz-4.18.0-
373.el8.x86_64 crashkernel=auto resume=/dev/mapper/znn-swap
rd.lvm.lv=znn/zvirt-node-ng-3.0-0.20220410.0+1 rd.lvm.lv=znn/swap rhgb quiet
root=/dev/znn/zvirt-node-ng-3.0-0.20220410.0+1 boot=UUID=d802e2a5-b7d8-4c6a-
b24a-625bfd971da7 rootflags=discard img.bootid=zvirt-node-ng-3.0-
0.20220410.0+1 rd.driver.blacklist=nouveau intel_iommu=on pci-
stub.ids=10de:1bb3
```

   Должны присутствовать следующие параметры:

   - `intel_iommu=on` или `amd_iommu=on` - IOMMU включен;
   - `pci-stub.ids=10de:1bb3,10de:10fa` - хост-устройство добавлено в список `pci-stub.ids`;

- `rd.driver.blacklist=nouveau` - Nouveau занесен в черный список;

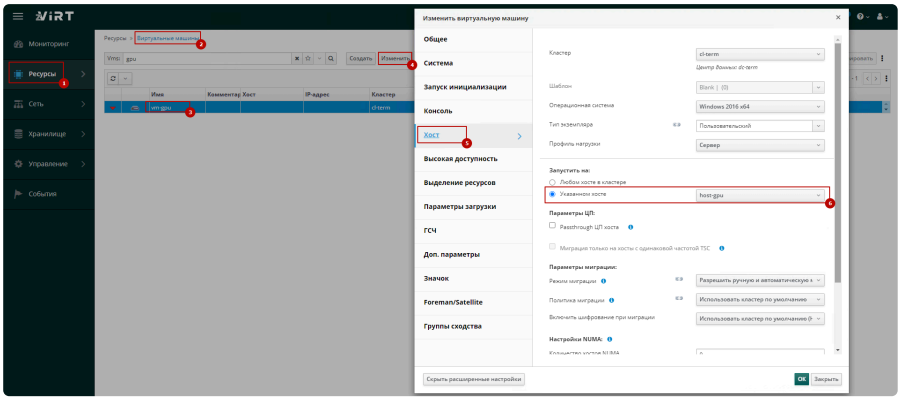
Как видно из примера, он отличается от файла, который был рассмотрен на шаге 3.

5. Выведите хост **host-gpu** из режима обслуживания. Для этого последовательно перейдите **Ресурсы > Хосты > host-gpu > Управление > Включить**.

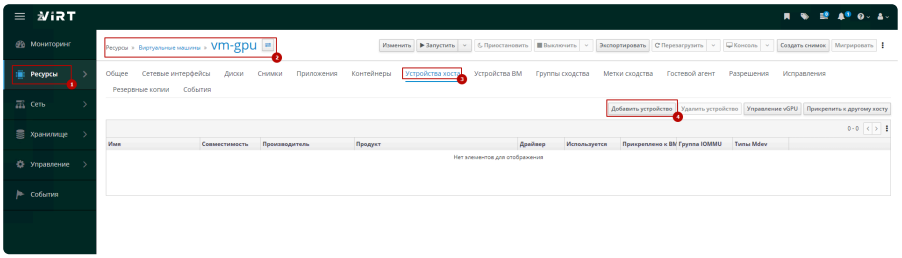
2.3. Присоединение графического процессора к виртуальной машине

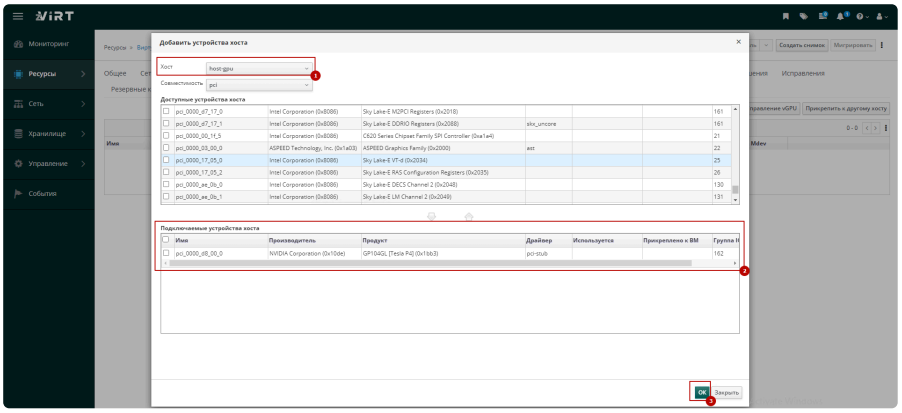
После отключения GPU от хоста необходимо добавить GPU в виртуальную машину. Последовательность действий будет следующая:

1. Для выбранной ВМ (например **vm-gpu**) установите параметр запуска на определённом хосте (**host-gpu**). Для этого последовательно выберите: **vm-gpu > Изменить > Хост > Запустить на указанном хосте > host-gpu**.



2. Добавьте GPU в ВМ **vm-gpu**. Для этого последовательно выберите: **vm-gpu > Устройства хоста > Добавить устройство > host-gpu > совместимость PCI**, выберите видеокарту и добавьте в подключаемые устройства.





2.4. Установка драйвера GPU на виртуальную машину

Рассмотрим процедуру установки драйвера для ОС Windows Server 2016 и CentOS 8 Stream.

2.4.1. Установка драйвера GPU на виртуальную машину с ОС Windows Server 2016

До подключения физической видеокарты к ВМ в `Диспетчер устройств Windows` будет отображаться карта `RedHat QXL Controller`,

1. Запустите виртуальную машину и подключитесь к ней с помощью консоли `VNC` или `SPICE`.
2. Установите гостевые дополнения, если они не были установлены ранее.
3. Загрузите и установите драйвер на ВМ. Информацию о получении драйвера смотрите на странице Драйверы на сайте NVIDIA.
4. После завершения установки драйвера перезагрузите машину. **Для виртуальных машин Windows** полностью отключите питание гостевой машины с портала администрирования или портала виртуальных машин, а не из гостевой операционной системы. (Выключение 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 VM для выключения виртуальной машины заставляет ее полностью очистить память.)
5. В **Диспетчер устройств Windows** отключите стандартный графический адаптер, оставив только карту от NVIDIA.

2.4.2. Установка драйвера GPU на виртуальную машину с ОС CentOS Stream 8

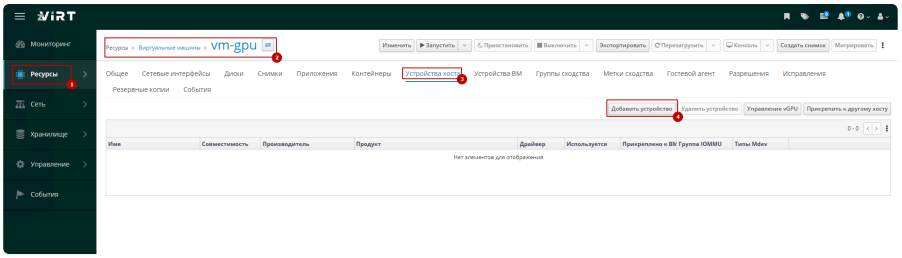
До подключения физической видеокарты к ВМ, стандартное устройство будет отображаться так:

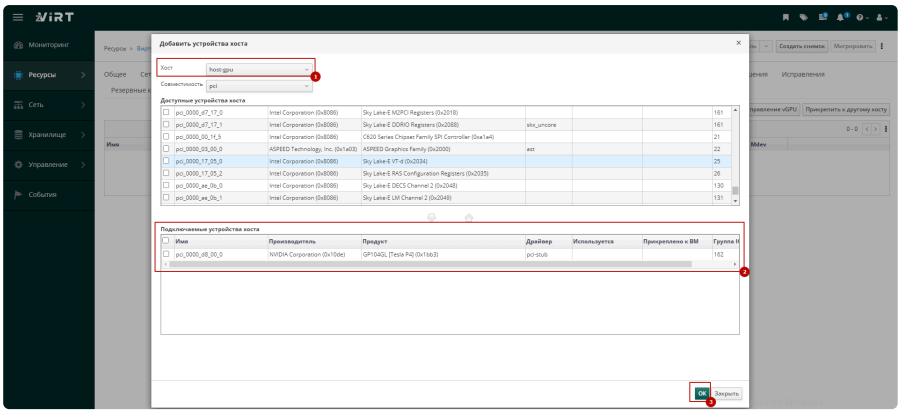
```
[root@localhost ~]# lspci | egrep -i *card
00:01.0 VGA compatible controller: Red Hat, Inc. QXL paravirtual graphic card
(rev 05)
```

Порядок подключения видеокарты к ВМ.

1. Установите гостевые дополнения на ВМ, если они не были установлены ранее. Произведите обновление системы, выполнив команду, например `dnf update -y` (для CentOS 8 Stream). Выключите ВМ.
2. Добавьте GPU в ВМ **vm-gpu**. Для этого последовательно выберите: **Ресурсы > Виртуальные машины > имя_ВМ > Устройства хоста > Добавить устройство**,

найдите карту в списке и добавьте в подключаемые устройства.





3. Проверьте, что карта отображается в списке подключенных устройств. Например:

```
[root@localhost ~]# lshw | grep -i NVIDIA -A 8 -B 3
          *-display
               description: 3D controller
               product: GP104GL [Tesla P4]
               vendor: NVIDIA Corporation
               physical id: 0
               bus info: pci@0000:07:00.0
               version: a1
               width: 64 bits
               clock: 33MHz
               capabilities: pm msi pciexpress bus_master cap_list
               configuration: driver=nouveau latency=0
               resources: irq:73 memory:f8000000-f8ffffff memory:d0000000-
dfffffff memory:e0000000-e1ffffff
```

4. Проверим, используется ли драйвер `nouveau` на виртуальной машине. Для этого выполним команду `lsmod | grep nouveau`.

```
[root@localhost ~]# lsmod | grep nouveau
nouveau              2355200  0
mxm_wmi                16384  1 nouveau
wmi                    32768  2 mxm_wmi,nouveau
video                  53248  1 nouveau
i2c_algo_bit           16384  1 nouveau
drm_display_helper    151552  1 nouveau
drm_ttm_helper         16384  2 qxl,nouveau
ttm                    81920  3 qxl,drm_ttm_helper,nouveau
drm_kms_helper        167936  5 qxl,drm_display_helper,nouveau
```

```
drm                    577536  7
drm_kms_helper,qxl,drm_display_helper,drm_ttm_helper,ttm,nouveau
```

5. Добавим драйвер `nouveau` в blacklist. Для этого создадим файл `blacklist-nouveau.conf ` и добавим в него следующую информацию:

```
[root@localhost ~]# nano /etc/modprobe.d/blacklist-nouveau.conf
# add to the end (create new if it does not exist)
blacklist nouveau
options nouveau modeset=0
```

6. Сгенерируйте новую конфигурацию файла **grub** и перезагрузитесь, выполнив команды:

```
[root@dlp ~]# dracut --force
[root@dlp ~]# reboot
```

7. После перезагрузки, проверьте, что драйвер `nouveau` не используется. Для этого выполните команду:

```
[root@localhost ~]# lsmod | grep nouveau
[root@localhost ~]#
```

8. Загрузите драйвер на ВМ. Информацию о получении драйвера смотрите на странице на сайте NVIDIA.

9. Добавите бит `x` для установки драйвера. Для этого выполните команду `chmod +x имя_драйвера`. Например:

```
[root@localhost ~]# chmod +x NVIDIA-Linux-x86_64-525.105.17.run
```

10. Установите пакеты, необходимые для работы:
    - Установка при наличии доступа в интернет:

      ```
      [root@localhost ~]# dnf -y install kernel-devel kernel-headers elfutils-libelf-devel zlib-devel gcc make
      ```

    - При отсутствии доступа в интернет необходимо скачать пакеты для CentOS 8 Stream и произвести их установку:

      ```
      [root@localhost pack1_centos] # rpm -ivh  --nodeps *.*
      ```

11. Узнайте полный путь до ядра системы. Для этого выполните команду:

```
[root@localhost ~]# ll /usr/src/kernels/
total 4
drwxr-xr-x. 23 root root 4096 Apr 18 04:14 4.18.0-485.el8.x86_64
```

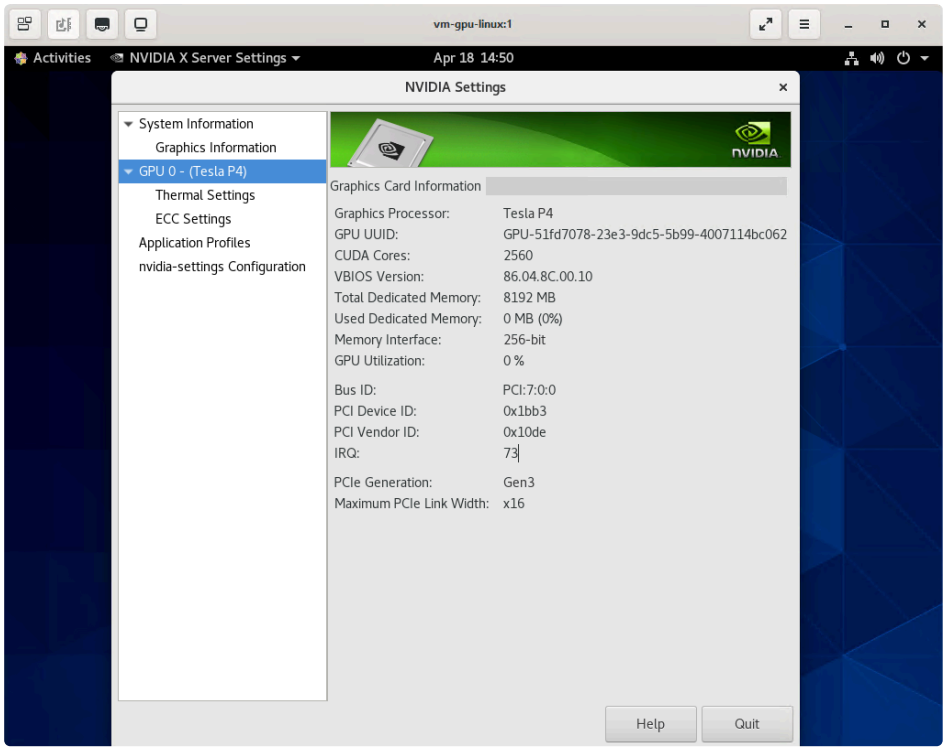
В текущем примере полный путь до ядра системы будет **/usr/src/kernels/4.18.0-485.el8.x86_64**

12. Установите драйвер NVIDIA, указав полный путь к ядру. Например:

```
[root@localhost ~]# ./NVIDIA-Linux-x86_64-525.105.17.run --kernel-source-path /usr/src/kernels/4.18.0-485.el8.x86_64
Verifying archive integrity... OK
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64 525.105.17.............
```

13. Перезагрузите ВМ. В графическом режиме проверьте отображение карты. Для проверки (в графическом режиме) выполните команду в терминале:

```
nvidia-settings&
```



Для проверки в режиме командной строки выполните `nvidia-smi`. Пример результата выполнения:

```
[root@localhost ~]# nvidia-smi
Wed Apr 19 08:50:27 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P4            Off  | 00000000:07:00.0 Off |                  Off |
| N/A   33C    P8     7W /  75W |      4MiB /  8192MiB |      0%      Default |
```

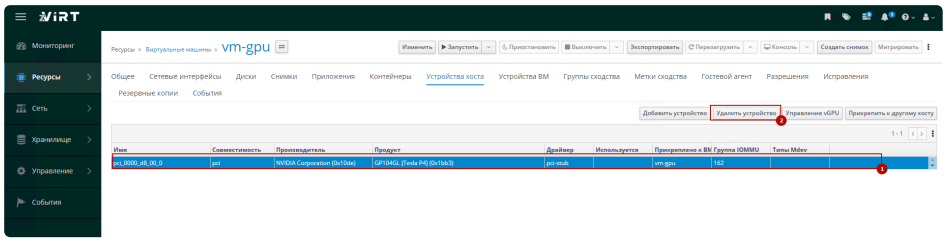
```
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                 GPU Memory |
|        ID   ID                                                  Usage      |
|=============================================================================|
|    0   N/A  N/A      1590      G   /usr/libexec/Xorg                  4MiB |
+-----------------------------------------------------------------------------+
```

Теперь GPU назначен виртуальной машине.

2.5. Удаление GPU из виртуальной машины

Для удаления видеокарты из ВМ порядок будет следующий:

1. Включите стандартный графический адаптер в `Диспетчере устройств`.
2. Удалите драйвера `NVIDIA`.
3. Выключите ВМ.
4. Последовательно перейдите **Ресурсы > Виртуальные машины > имя_ВМ > Устройства хоста > Удалить устройство**.



3. Дополнительная информация

Для получения дополнительной информации об использовании NVIDA vGPU в RHEL с KVM см:

- NVIDIA GPU Software Release Notes.
- Документацию по программному обеспечению NVIDIA Virtual GPU на https://docs.nvidia.com.

Диагностика доступа к консоли ВМ на примере протокола SPICE

При работе ВМ на хосте в каталоге `/run/libvirt/qemu` создаются файлы конфигураций:

```
ls -l /run/libvirt/qemu/
-rw-------. 1 root root     4 авг  4 16:50 deb_storage_sz.pid
-rw-------. 1 root root 18049 авг 20 10:02 deb_storage_sz.xml
-rw-------. 1 root root     5 авг 18 15:21 host2_infoland_SZ.pid
-rw-------. 1 root root 24091 авг 20 14:40 host2_infoland_SZ.xml
-rw-------. 1 root root     5 апр  4 01:13 HostedEngine.pid
-rw-------. 1 root root 20263 апр  4 01:13 HostedEngine.xml
...
```

Файл `pid` содержит в себе информацию о номере процесса:

```
cat /run/libvirt/qemu/HostedEngine.pid
17851
```

Узнать открытые сокеты/порты процесса можно командой:

```
ss -tulpan | grep 17851
tcp LISTEN 0 128 172.26.27.100:5900 *:* users:(("qemu-kvm",pid=17851,fd=22))
tcp LISTEN 0 128 172.26.27.100:5901 *:* users:(("qemu-kvm",pid=17851,fd=23))
tcp LISTEN 0 1   172.26.27.100:5902 *:* users:(("qemu-kvm",pid=17851,fd=52))
```

Узнать информацию об используемых портах можно из `XML файла`:

```
grep tlsPort /run/libvirt/qemu/HostedEngine.xml
<graphics type='spice' port='5900' tlsPort='5901' autoport='yes'
listen='172.26.27.100' passwd='*****' passwdValidTo='1970-01-01T00:00:01'>
```

Как видно, для доступа к ВМ по протоколу `SPICE` используется `5901` порт.

По номеру процесса можно провести диагностику запуска ВМ:

```
#strings /proc/17851/cmdline
...

-spice \
port=5900,tls-port=5901,addr=172.26.27.100,x509-dir=/etc/pki/vdsm/libvirt-
spice,tls-channel=main,tls-channel=display,tls-channel=inputs,tls-
```

```
channel=cursor,tls-channel=playback,tls-channel=record,tls-
channel=smartcard,tls-channel=usbredir,seamless-migration=on
...
```

Для автоматизации можно использовать скрипт для диагностики:

```
for i in $(ls /run/libvirt/qemu/*.pid); do ls $i \
    | xargs -I % sh -c 'FILE=$(basename %) && PID=$(cat %) \
    && PORT=$(strings /proc/$PID/cmdline | grep tls-port) && echo "$FILE:
$PORT"'; done
```

Вывод содержит следующую информацию:

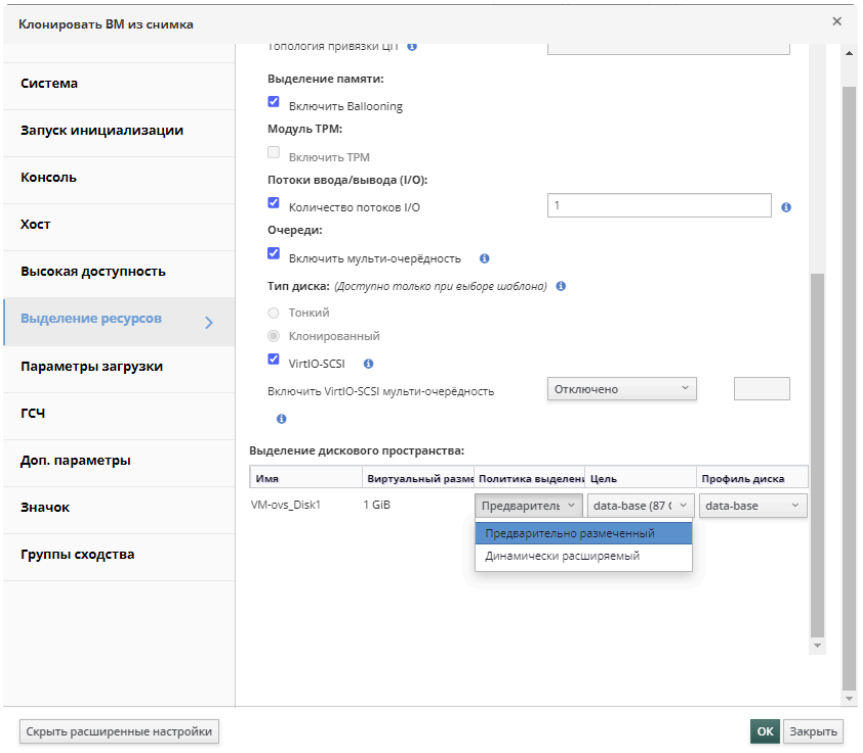
```
deb_storage_sz.pid: port=5903,tls-port=5904,addr=172.26.27.100,x509-
dir=/etc/pki/vdsm/libvirt-spice,tls-channel=main,tls-channel=display,tls-
channel=inputs,tls-channel=cursor,tls-channel=playback,tls-channel=record,tls-
channel=smartcard,tls-channel=usbredir,seamless-migration=on`

host2_infoland_SZ.pid: port=5913,tls-port=5915,addr=172.26.27.100,x509-
dir=/etc/pki/vdsm/libvirt-spice,tls-channel=main,tls-channel=display,tls-
channel=inputs,tls-channel=cursor,tls-channel=playback,tls-channel=record,tls-
channel=smartcard,tls-channel=usbredir,seamless-migration=on

HostedEngine.pid: port=5900,tls-port=5901,addr=172.26.27.100,x509-
dir=/etc/pki/vdsm/libvirt-spice,tls-channel=main,tls-channel=display,tls-
channel=inputs,tls-channel=cursor,tls-channel=playback,tls-channel=record,tls-
channel=smartcard,tls-channel=usbredir,seamless-migration=on
```

Преобразование диска для виртуальной машины

Преобразования между типами дисков динамически расширяемых(thin provisioning) и предварительно размеченных(preallocated) выполняется через клонирование виртуальной машины.

Для этого выполните следующие шаги:

1. Создайте снимок (snapshot) ВМ. Для этого:
    a. На Портале администрирования перейдите в **Ресурсы > Виртуальные машины**.
    b. Выделите нужную ВМ и нажмите **[Создать снимок]**.
    c. Дождитесь окончания процесса создания снимка.
2. Клонируйте ВМ из созданного снимка и измените тип диска на нужный. Для этого:
    a. Нажмите на имя ВМ, для которой создавали снимок, для перехода в подробное представление.
    b. Перейдите на вкладку **Снимки**.
    c. Выделите нужный снимок и нажмите **[Клонировать]**.
    d. В окне **Клонировать ВМ из снимка** перейдите на вкладку **Выделение ресурсов**.
    e. В таблице **Выделение дискового пространства** в столбце **Политика выделения** выберите нужный тип диска.



f. Нажмите **[OK]**

В результате выполненных действий будет создана копия виртуальной машины с другим типом диска.

Миграция ВМ между кластерами

Пояснение

В zVirt версии 4.1 и старше отсутствует возможность миграции виртуальных машин между разными кластерами. Предполагается, что хосты кластеров имеют разную архитектуру и поэтому миграция возможно исключительно через функционал экспорта/импорта Импорт-домена. Однако при необходимости администратор может произвести живую миграцию виртуальной машины за счет средств API.

В zVirt 4.2 поддерживается миграция между кластерами штатными средствами. Подробнее см. в разделе Миграция виртуальных машин между хостами руководства по управлению ВМ.

Справка

Для получения справочной информации скрипт необходимо запустить с ключом **-h**:

```
./migrate.py -h
usage: migrate.py [-h] [--login LOGIN] --password PASSWORD engine vmname
clustername hostname

This script starts live migration from cluster to cluster in zVirt\oVirt
virtualzation

positional arguments:
  engine                ip or hostname of engine
  vmname                Name of target VM
  clustername           Name of target cluster
  hostname              name of host in target cluster

optional arguments:
  -h, --help            show this help message and exit
  --login LOGIN         Login. Default admin@internal
  --password PASSWORD   admin password
```

1. Подготовка

1. Скачайте скрипт по ссылке
2. Разместите скрипт в любом из следующих мест:

- виртуальная машина **HostedEngine**;
- хост, на котором произведена установка zVirt в режиме **Standalone**;
- рабочее место администратора.

3. Назначьте скрипт исполняемым командой: `chmod +x migrate.py`

2. Использование

Запустите скрипт с указанием необходимых параметров, например:

```
./migrate.py --login admin@internal --password 1 engine.domain.local testvm
cluster2 host2
```

3. Возможные ошибки

- VM **<VM_NAME>** not found - указанная виртуальная машина не найдена. Убедитесь в правильности указания имени
- Cluster **<CLUSTER_NAME>** not found - кластер с указанным именем не найден. Убедитесь, что менеджер виртуализации управляет указанным кластером
- **<HOST>** host not found - хост с указанным именем не найден. Убедитесь, что указанный хост входит в указанный кластер

Создание кворумного диска(диск-свидетель) для организации отказоустойчивого кластера Windows в zVirt

В случае необходимости создания **Quorum-диска** для операционных систем Windows Server, создать его можно с помощью выделения прямого LUN с определёнными опциями. Также перед тем, как подключить LUN, на хостах необходимо:

1. На всех хостах настроить службу **multipath**, в её конфигурационный файл необходимо добавить параметр **reservation_key file** в секцию **defaults**, после чего перезапустите службу **multipathd**

   ```
   defaults {
      ......
      reservation_key file
   }
   ```

2. Перейти в консоль менеджера управления и выполнить команду

   ```
   engine-config -s PropagateDiskErrors=true
   ```

3. Перезапустить службу **ovirt-engine**

   ```
   systemctl restart ovirt-engine
   ```

При добавлении нового диска, укажите **Прямой LUN** и проставьте галочки напротив следующих параметров:

- Включить интерфейс SCSI
- Разрешить привилегированный ввод/вывод SCSI
- Используется резервирование SCSI

Доступ к последовательной консоли ВМ

1. Доступ к последовательной консоли ВМ

Можно получить доступ к последовательной консоли ВМ из командной строки вместо открытия консоли на портале менеджере управления. Последовательная консоль эмулируется через VirtIO канал, используя SSH и SSH-ключи.

В этом случае менеджер управления действует как прокси для соединения, предоставляет информацию о расположении ВМ и хранит ключи авторизации. Можно добавить публичные ключи для каждого пользователя на портале администрирования или пользовательском портале.

Вы можете получить доступ к последовательным консолям только для тех виртуальных машин, для которых у вас есть соответствующие разрешения.

Чтобы получить доступ к последовательной консоли виртуальной машины, пользователь должен иметь разрешение `UserVmManager`, `SuperUser` или `UserInstanceManager` на этой виртуальной машине. Эти разрешения должны быть явно определены для каждого пользователя. Недостаточно назначить эти разрешения всем.

Последовательная консоль доступна через порт `2222/tcp` на менеджере управления. Порт открывается в процессе установки `Hosted Engine`.

Чтобы разрешить доступ к последовательной консоли необходимо добавить следующие правила на сетевом экране:

- На менеджере управления открыть порт `2222/tcp`

```
Firewall-cmd –permanent –add-port=2222/tcp
Firewall-cmd –reload
```

- На хостах открыть порт `2222/tcp`

```
Firewall-cmd –permanent –add-port=2223/tcp
Firewall-cmd –reload
```

- На виртуальных машинах, к которым необходим доступ по последовательной консоли необходимо добавить следующие строки в файл `/etc/default/grub`:

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8"
GRUB_TERMINAL="console serial"
```

```
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

- Переинициализируйте файл /boot/grub2/grub.cfg: **Для BIOS совместимых ВМ необходимо использовать команду:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

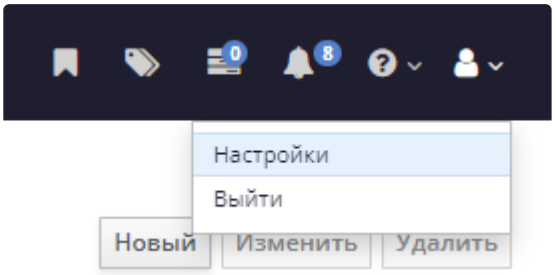
  **Для UEFI совместимых ВМ необходимо использовать команду:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

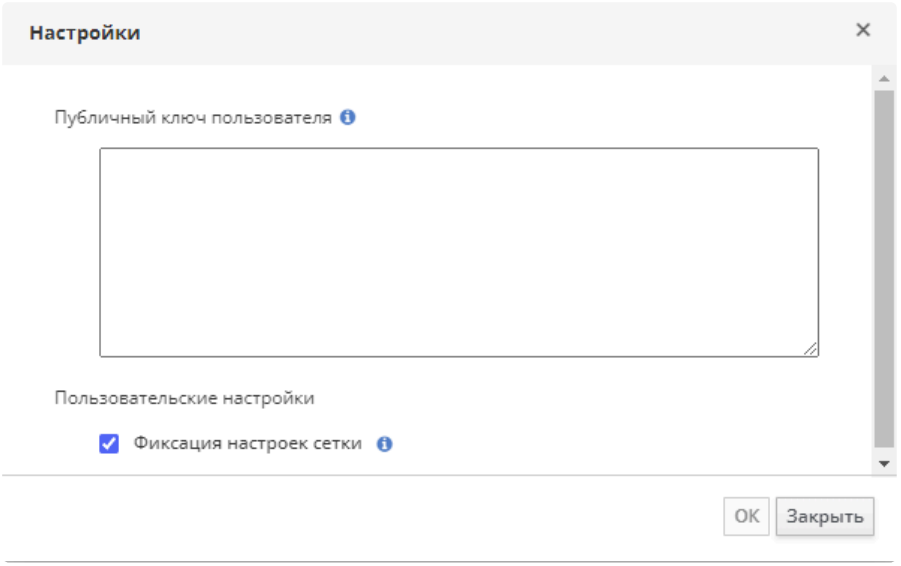
- На клиентской машине, с которой планируется осуществлять доступ к последовательной консоли необходимо сгенерировать ключи SSH, например:

```
ssh-keygen -t rsa -b 2048 -f .ssh/serialconsolekey
```

- На портале администратора или пользовательском портале кликнуть на имя авторизованного пользователя в верхнем правом углу и выбрать настройки.

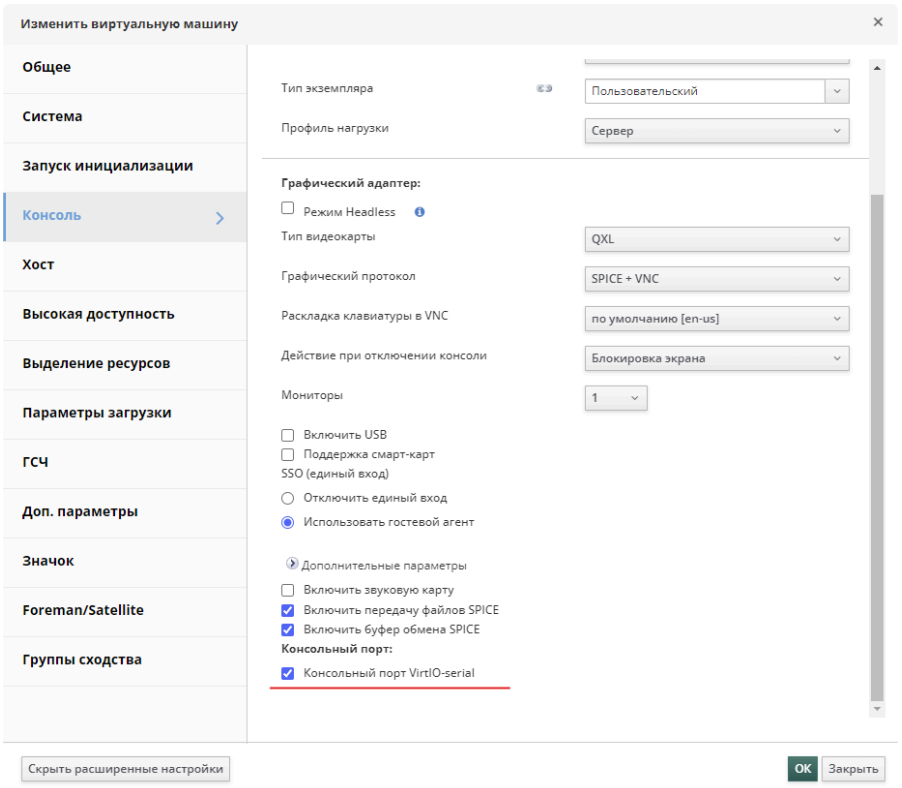


Скопировать в открывшееся окно ранее сгенерированный публичный ключ с клиентской машины.



- Перейти в **Ресурсы > Виртуальные машины** и выбрать нужную ВМ.
- Нажать **[Изменить]**.

- В вкладке **Консоль** выбрать **Консольный порт VirtIO serial**.



2. Подключение к последовательной консоли ВМ.

На клиентской машине введите следующую команду для подключения:

```
ssh -t -p 2222 ovirt-vmconsole@<Manager_FQDN> -i .ssh/serialconsolekey
```

Где *<Manager_FQDN>* - это FQDN менеджера управления.

Если доступны более чем одна ВМ будет выведен список машин и их ID. Для подключения введите номер ВМ из списка и нажмите **[Enter]**.

Резервное копирование виртуальных машин в домен экспорта

1. Подключение домена экспорта

Экспорт-домены - это временные хранилища, которые используются для резервного копирования виртуальных машин, копирования и перемещения снимков между центрами данных и площадками zVirt.

Экспорт-домен можно перемещать между центрами данных, но в один момент времени он может быть активен только в одном центре данных.

- Экспорт-домен может быть создан только на хранилище файлового типа (NFS, Posix совместимая ФС, GlusterFS).
- В центре данных может быть только один домен типа Экспорт.

Далее описана процедура создания и добавления домена экспорта на примере NFS.

Порядок действий:

1. На NFS-сервере подготовьте и экспортируйте каталог, который будет подключен как домен экспорта

   a. Создайте директорию, например:

   ```bash
   mkdir /storage/export;
   ```

   b. Назначьте владельца:

   ```bash
   chown -R 36:36 /storage/export;
   ```

   c. Установите права:

   ```bash
   chmod 0755 /storage/export.
   ```

   d. Если NFS-сервер реализован на системе, отличной от zVirt Node, то необходимо создать служебных пользователей и группы:

   ```bash
   groupadd sanlock -g 179
   groupadd kvm -g 36
   ```

```
useradd sanlock -u 179 -g 179 -G kvm
useradd vdsm -u 36 -g 36 -G sanlock
```

e. Опубликуйте каталог, прописав его в конфигурационном файле `NFS-сервера`:

```bash
echo "/storage/export *(rw,anonuid=36,anongid=36)" >> /etc/exports
```

f. Убедитесь в правильности задания параметров доступа:

```bash
cat /etc/exports
```

g. Запустите необходимые сервисы:

```bash
systemctl enable nfs-server
systemctl enable rpcbind
systemctl enable nfs-blkmap
systemctl restart nfs-server
systemctl restart rpcbind
systemctl restart nfs-blkmap
```

h. Создайте правила межсетевого экрана (например, firewalld) для обеспечения доступности хранилища для других хостов:

```bash
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --reload
```

2. Добавьте домен экспорта в среду zVirt:

a. Авторизуйтесь на портале администрирования.

b. В боковом меню перейдите в **Хранилище > Домены**.

c. Нажмите **[Новый домен]**.

d. В окне добавления нового домена введите следующие параметры:

- В поле **Центр данных** выберите центр данных, к которому будет подключен домен.
- В поле **Функция домена** выберите **Экспорт**.
- Убедитесь, что в поле **Тип хранилища** выбран **NFS**.
- При необходимости в поле **Используемый хост** измените хост, который будет использован для подключения к хранилищу.
- В поле **Имя** укажите уникальное имя домена
- В поле **Путь экспорта** укажите путь для подключения к NFS-хранилищу, например, **nfs-server.example.com:/storage/export**.

- Прочие параметры можно оставить со значениями по умолчанию.

e. Нажмите **[OK]**.

f. Убедитесь, что домен перешёл в состояние **Активный**.

2. Резервное копирование ВМ в домен экспорта

Встроенное средство резервного копирования (**СРК**) ВМ имеет возможность работать в двух конфигурациях - ручное резервное копирование и резервное копирование по расписанию.

В процессе резервирования СРК проверяет количество свободного места на дисках, необходимое для выполнения операций.

Ниже описаны действия, необходимые для резервного копирования ВМ в домен экспорта.

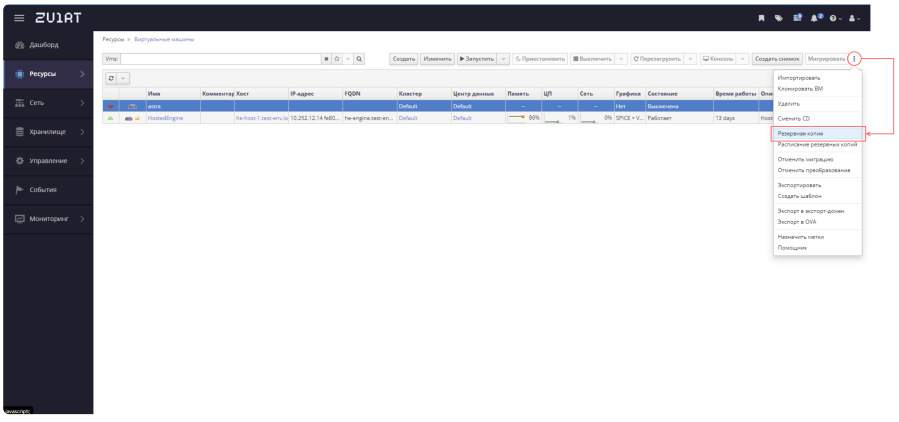
Предварительные требования:

- Наличие домена экспорта в центре данных с резервируемой ВМ.

2.1. Ручное резервное копирование

Порядок действий:

1. Авторизуйтесь на портале администрирования.
2. Перейдите в **Ресурсы** > **Виртуальные машины**.
3. Выделите необходимую ВМ.
4. Нажмите ⋮ и выберите **Резервная копия**.

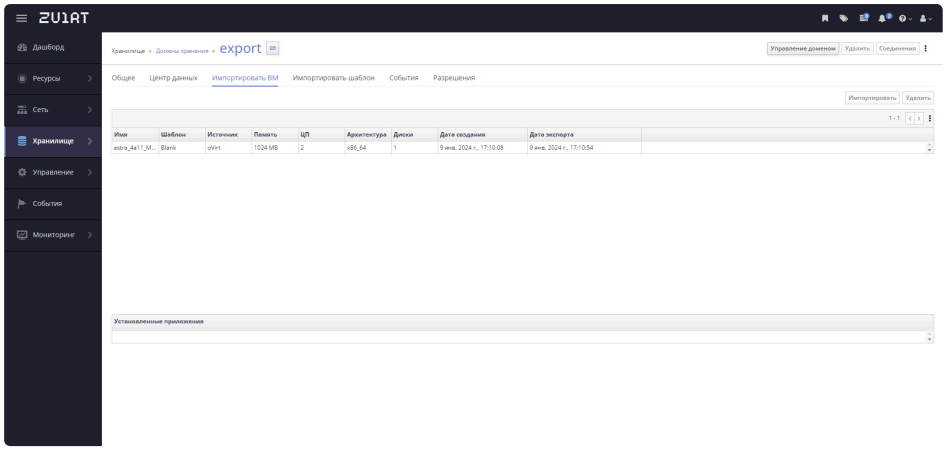


5. В появившемся окне убедитесь, что выбран подключенный домен экспорта.

> Если в поле **Домен хранения** отсутствует подключенный домен экспорта, убедитесь, что:
>
> - Вы подключили его к правильному центру данных.
> - При подключении выбрана функция домена **Экспорт**.

6. Нажмите **[Начать]**.

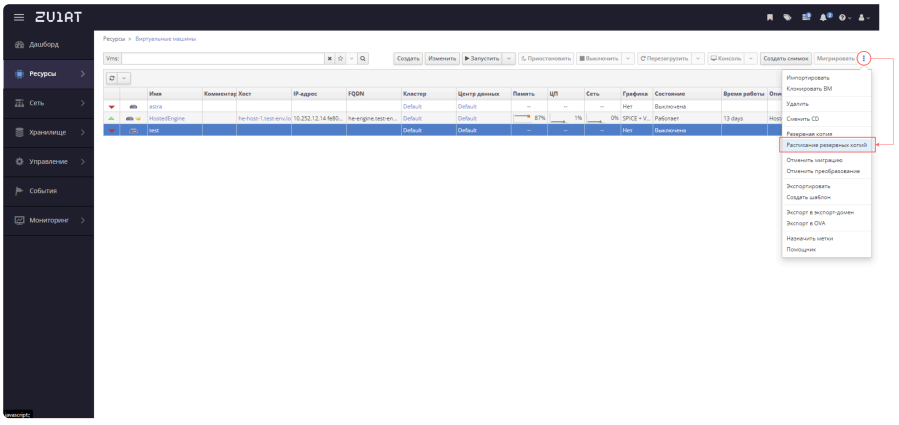
Чтобы убедиться, что резервная копия сохранена в домене экспорта перейдите в **Хранилище** > **Домены**, нажмите на имя домена экспорта для перехода в подробное представление, откройте вкладку **Импортировать ВМ**. В списке должна присутствовать резервная копия соответствующей ВМ.



2.2. Резервное копирование по расписанию

Порядок действий:

1. Авторизуйтесь на портале администрирования.
2. Перейдите в **Ресурсы** > **Виртуальные машины**.
3. Выделите необходимую ВМ.
4. Нажмите ⋮ и выберите **Расписание резервных копий**.



5. В появившемся окне:

   a. Убедитесь, что выбран домен экспорта:

   > Если в поле **Домен хранения** отсутствует подключенный домен экспорта, убедитесь, что:
   >
   > - Вы подключили его к правильному центру данных.
   > - При подключении выбрана функция домена **Экспорт**.

   b. В поле **Бэкап по расписанию** выберите значение **Включено**.

c. В поле **Расписание** введите выражение в формате Quartz Cron для задания расписания резервного копирования. Подробнее о задании расписания в формате Quartz cron см. в разделе Использование формата Quartz cron для задания расписания.

d. В поле **Количество хранимых копий** укажите сколько резервных копий необходимо хранить в домене экспорта.

e. нажмите **[Сохранить]**

6. Убедитесь, что резервные копии создаются в соответствии с расписанием.

3. Восстановление ВМ из домена экспорта

Для восстановления ВМ из резервной копии в домене экспорта используйте процедуру, описанную в разделе Импортирование виртуальной машины из домена экспорта руководства по управлению ВМ.

4. Использование формата Quartz cron для задания расписания

При ручном вводе выражения, определяющего расписание запуска задачи, ожидается использование формата Quartz cron. Данный формат состоит из 7 полей, разделенных пробелом. Каждое поле имеет определённую значимость.

Поля могут содержать любые допустимые значения, а также различные комбинации разрешенных специальных символов для этого поля. Допустимые значения и символы представлены в таблице ниже

Номер	Имя поля	Допустимые значения	Разрешенные специальные символы
1	Секунды	0-59	, - * /
2	Минуты	0-59	, - * /
3	Часы	0-23	, - * /
4	День месяца	1-31	, - * ? / L W
5	Месяц	1-12 или JAN-DEC	, - * /
6	День недели	1-7 или SUN-SAT	, - * ? / L #
7	Год	1970-2099	, - * /

Значение специальных символов

- `*` (все значения) - используется для выбора всех значений в поле. Например, `\*` в поле минут означает «каждую минуту»
- `?` (любое значение) - полезно, когда нужно указать что-то в одном из двух полей, в которых этот символ разрешен, но не в другом. Например, если необходимо, чтобы задача запустилась в определенный день месяца (допустим, 10-го числа), но все равно, какой это будет день недели, то можно разместить `10` в поле "День месяца", а `?` - в поле "День недели".
- `-` - используется для указания диапазонов. Например, `10-12` в поле часов означает "часы 10, 11 и 12"
- `,` - используется для указания дополнительных значений. Например, `MON,WED,FRI` в поле "день недели" означает "дни понедельник, среда и пятница".
- `/` - используется для указания приращений. Например, `0/15` в поле секунд означает "секунды 0, 15, 30 и 45". А `5/15` в поле секунд означает "секунды 5, 20, 35 и 50".
- `L` (последний) - имеет разное значение в каждом из двух полей, в которых оно разрешено. Например, значение `L` в поле "день месяца" означает "последний день месяца" - 31 день для января, 28 день для февраля в невисокосные годы. Если это значение используется в поле дня недели само по себе, оно означает просто "7" или "SAT". Но если оно используется в поле дня недели после другого значения, оно означает "последний `xxx` день месяца" - например, `6L` означает "последняя пятница месяца". При использовании опции `L` важно не указывать списки или диапазоны значений, так как вы получите путаные/неожиданные результаты.
- `W` (будний день) - используется для указания ближайшего к данному дню дня недели (понедельник-пятница). Например, если вы укажете `15W` в качестве значения для поля "День месяца", это будет означать: "ближайший будний день к 15 числу месяца". Таким образом, если 15-е число приходится на субботу, задача запустится в пятницу 14-го. Если 15-е число приходится на воскресенье, задача запустится в понедельник 16-го. Если 15-е число приходится на вторник, то задача запустится во вторник 15-го. Однако если вы укажете `1W` в качестве значения дня месяца, а 1-е число будет субботой, задача запустится в понедельник 3-го числа, поскольку он не будет "перескакивать" через границу дней месяца. Символ `W` может быть указан только в том случае, если день месяца - это один день, а не диапазон или список дней.
- `#` - используется для указания "n-го" XXX дня месяца. Например, значение `6#3` в поле "День недели" означает "третья пятница месяца" (день `6` = пятница, а `#3` = третий в месяце).

Пример 1. Использование формата quartz cron

Выражение	Значение
`0 0 12 * * ? *`	Запуск в 12 часов дня каждый день
`0 15 10 ? * * *`	Запуск в 10:15 утра каждый день
`0 15 10 * * ? 2023`	Запуск в 10:15 утра каждый день в течение 2023 года
`0 * 14 * * ? *`	Запуск каждую минуту, начиная с 14:00 и заканчивая 14:59, каждый день
`0 0/5 14,18 * * ? *`	Запуск каждые 5 минут, с 14:00 и до 14:55, а также с 18:00 и до 18:55, каждый день
`0 15 10 ? * MON-FRI *`	Запуск в 10:15 утра каждый понедельник, вторник, среду, четверг и пятницу
`0 15 10 L * ? *`	Запуск в 10:15 утра в последний день каждого месяца
`0 15 10 ? * 6#3 *`	Запуск в 10:15 утра в третью пятницу каждого месяца