

Миграция виртуальных машин из Hyper-V в Proxmox

IT itproblog.ru/миграция-виртуальных-машин-из-hyper-v-в-proxmox

Роман

25.12.2022

В моем блоге уже была статья о том, как выполняется установка Proxmox. В этой публикации я кратко покажу, как выполняется миграция виртуальных машин из Hyper-V в Proxmox. Причем миграция разных поколений виртуальных машин Hyper-V – как VM первого поколения, так и VM второго поколения.

Приведенная ниже последовательность действий покрывает основные типовые сценарии миграции. Поскольку дополнительных вводных при миграции может быть много, то маловероятно, что это руководство можно рассматривать как универсальное и охватывающее все возможные случаи.

В руководстве от Proxmox не так много пояснительной информации о процессе миграции с Hyper-V. Поэтому я постараюсь наглядно показать весь процесс.

Исходные данные

Для тестового сценария создал несколько виртуальных машин.

Виртуальная машина 1.

Параметр	Значение
Поколение виртуальной машины Hyper-V	2
Операционная система	Ubuntu Server 22.04 LTS
Тип BIOS	UEFI
Размер диска 1 (системный)	60 ГБ

Виртуальная машина 2.

Параметр	Значение
Поколение виртуальной машины Hyper-V	2
Операционная система	Windows Server 2016 Standard
Тип BIOS	UEFI
Размер диска 1 (системный)	60 ГБ

Немного забегаю наперед скажу, что миграция виртуальных машин Linux чуть проще, т.к. драйвер для Virtio уже на есть на борту. Для виртуальных машин Windows этот драйвер нужно будет установить вручную.

В качестве исходного гипервизора Hyper-V выступал Windows Server 2022.

В качестве целевого гипервизора выступает Proxmox 7.2-7.

Предварительная подготовка

Шаги по предварительной подготовке виртуальных машин одинаковы, как для платформы Linux, так и для платформы Windows:

1. Создание копии vhd/vhdx диска.
2. Копирование vhd/vhdx диска на сервер Proxmox.

Создание копии vhd/vhdx диска

На самом деле здесь есть несколько вариантов. В зависимости от того, насколько возможен перерыв в работе виртуальной машины Hyper-V.

Итак, для создания копии диска есть следующие варианты:

1. Первый вариант самый оптимальный – вы просто останавливаете виртуальную машину и загружаете диски на сервер Proxmox. Этот метод обеспечит вам гарантированную целостность данных на диске. К тому же, если после этого вы не будите запуска VM на сервере Hyper-V, то по завершении миграции получите на Proxmox самую актуальную версию VM. И вам не нужно будет думать о том, как перенести дельту изменений виртуальной машины.
2. Если виртуальная машина критическая и нельзя выполнять её остановку, то вы можете выполнить экспорт виртуальной машины. В результате вы получите копию всех файлов работающей виртуальной машины Hyper-V, в т.ч. и виртуальных дисков. Недостаток этого метода в том, что выполняется экспорт всех дисков. Если какие-то из дисков нужно исключить, то это не ваш метод. Однако, именно этим методом пользовался я. Из своей практики скажу – я ни разу не получал “битых” дисков при использовании этого метода.
3. Если виртуальную машину останавливать нельзя и нужно выполнить экспорт только части дисков. (например, нужен только системный диск), то вы можете использовать утилиту Disk2vhd. На своей практике я тоже её использовал довольно часто. Лично у меня проблем с консистентностью данных не было. Но это самый относительно ненадежный метод, т.к. на высоконагруженных системах есть шанс получить не совсем консистентные данные.

Повторюсь – я использовал метод №2.

Копирование vhd/vhdx диска на сервер Proxmox

Это тоже относительно понятный шаг. Все, что вам нужно сделать – это скопировать виртуальный диск (или диски) на сервер Proxmox в локальную директорию. Инструмент для копирования можете использовать любой. Если вы выполняете

копирования из среды Windows, то можете использовать WinSCP.

Если вы работаете в среде Linux, то можете использовать старую добрую утилиту scp. Формат команды для scp следующий:

```
scp <source_vhdx> <proxmox_user>@<proxmox_ip>:<proxmox_folder_path>
```

Пример:

```
scp tst-lin.vhdx root@10.10.10.20:/mnt/pve/HDD
scp tst-win.vhdx root@10.10.10.20:/mnt/pve/HDD
```

По итогу вы должны увидеть ваши vhd(x) диски в директории на сервере Proxmox:

```
root@pve01:~# ls -lh /mnt/pve/HDD/
total 15G
drwxr-xr-x 2 root root 4.0K Jul 10 18:29 dump
drwxr-xr-x 6 root root 4.0K Nov 27 04:57 images
drwx----- 2 root root 16K Jul 10 18:29 lost+found
drwxr-xr-x 2 root root 4.0K Jul 10 18:29 private
drwxr-xr-x 2 root root 4.0K Jul 10 18:29 snippets
drwxr-xr-x 4 root root 4.0K Jul 10 18:29 template
-rw-r--r-- 1 root root 5.4G Dec 22 05:18 tst-lin.vhdx
-rw-r--r-- 1 root root 9.6G Dec 22 05:36 tst-win.vhdx
root@pve01:~#
```

Создание виртуальных машин

После того, как вы выполнили конвертацию жестких дисков можно создавать виртуальные машины. Процесс настройки виртуальных машин после создания немного отличается. Для Linux машин особо дополнительных действий не требуется. Для машин Windows необходимо будет установить драйвер virtio. Но обо всех подробностях ниже.

Для ОС Linux

Итак, создает заготовку виртуальной машины для Linux:

1. Указываем сервер Proxmox (если у вас их несколько) и имя виртуальной машины:

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

Node: pve01 Resource Pool: (empty)

VM ID: 104

Name: test-lin

2. Указываем тип операционной системы – Linux.

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

☐ Use CD/DVD disc image file (iso)

Storage: NVMe

ISO image: (empty)

☐ Use physical CD/DVD Drive

☒ Do not use any media

Guest OS:

Type: Linux

Version: 5.x - 2.6 Kernel

3. Далее необходимо указать ряд важных параметров:

Тип эмулируемого аппаратного обеспечения – я укажу i440fx.

Тип BIOS – выберу OVMF (UEFI) (т.к. исходная VM использовала UEFI BIOS).

Тип SCSI контроллера – VirtIO SCSI, т.к. его поддержка есть в Linux из коробки.

Также необходимо выбрать формат и хранилище для UEFI раздела.

Create: Virtual Machine

General OS **System** Disks CPU Memory Network Confirm

Graphic card: Default

Machine: Default (i440fx)

Firmware

BIOS: OVMF (UEFI)

Add EFI Disk: ☒

EFI Storage: NVMe

Format: QEMU image format (qcow2)

Pre-Enroll keys: ☒

SCSI Controller: VirtIO SCSI

Qemu Agent: ☐

Add TPM: ☐

Help Advanced ☐ Back Next

4. На странице конфигурации виртуальных дисков я удалю вообще все диски:

Create: Virtual Machine

General OS System **Disks** CPU Memory Network Confirm

No Disks

Add

5. Указываем количественные ресурсы процессора:

Create: Virtual Machine

General OS System Disks **CPU** Memory Network Confirm

Sockets: 1 Type: Default (kvm64)

Cores: 4 Total cores: 4

6. Указываем количество оперативной памяти:

Create: Virtual Machine

General OS System Disks CPU **Memory** Network Confirm

Memory (MiB): 4096

7. И параметры сети:

Create: Virtual Machine

General OS System Disks CPU Memory **Network** Confirm

☐ No network device

Bridge: vmbr0 Model: **VirtIO (paravirtualized)**

VLAN Tag: no VLAN MAC address: auto

Firewall: ☒

Обратите внимание на тип сетевого адаптера – VirtIO. Аналогично контроллеру SCSI его поддержка есть в Linux из коробки.

8. На странице со сводными параметрами создаваемой виртуальной машины нажмите кнопку “**Finish**”.

При следующем шаге важно определиться с форматом дисков, которые вы будете использовать (raw|qcow2|vmdk). У каждого формата есть свои особенности. Например, в raw (формат по умолчанию, если ничего не указывать) не будут работать снимки. Если вам нужны снимки, то используйте формат qcow2. Для этого необходимо использовать дополнительный параметр `–format`. Например так: `qm importdisk 104 /mnt/pve/HDD/tst-lin.vhdx NVMe –format qcow2`

9. Импортируем виртуальный жесткий диск:

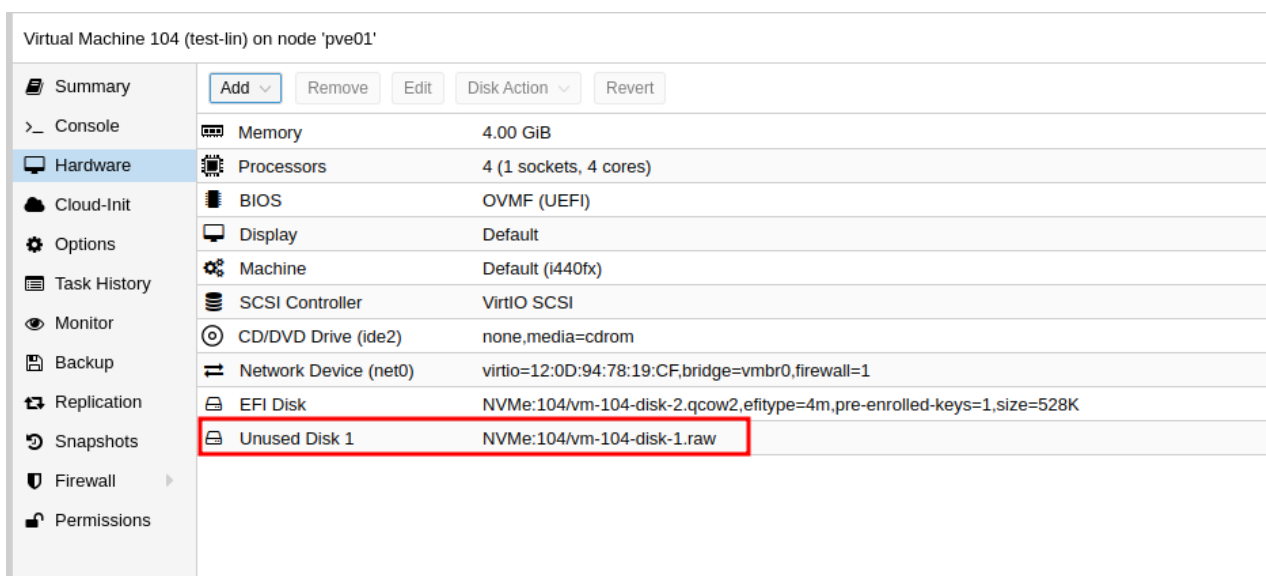
```
qm importdisk 104 /mnt/pve/HDD/tst-lin.vhdx NVMe
```

```
transferred 59.2 GiB of 60.0 GiB (98.68%)
transferred 59.8 GiB of 60.0 GiB (99.72%)
transferred 60.0 GiB of 60.0 GiB (100.00%)
transferred 60.0 GiB of 60.0 GiB (100.00%)
Successfully imported disk as 'unused0:NVMe:104/vm-104-disk-0.raw'
root@pve01:~#
```

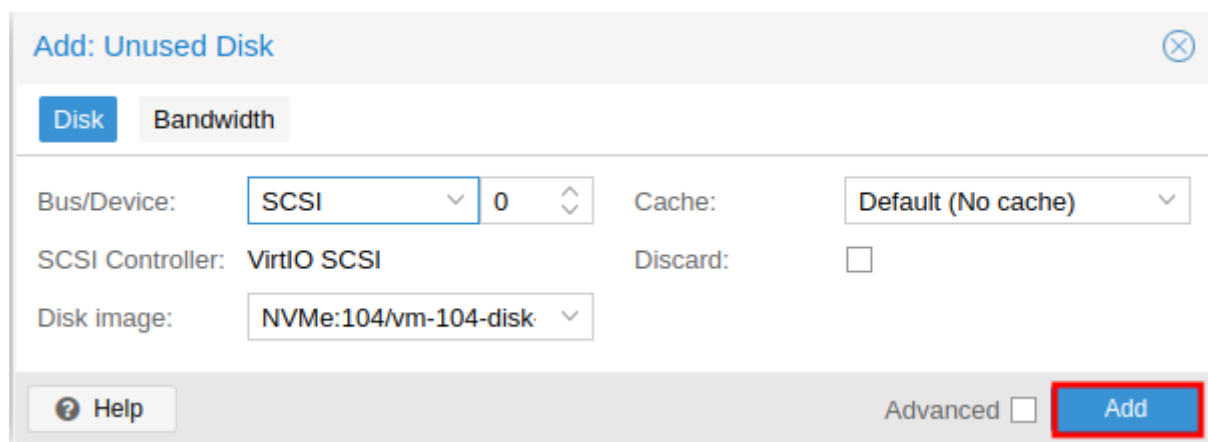
Формат команды следующий:

```
qm importdisk <vmid> <source> <storage>
```

10. После этого в настройках виртуальной машины у нас должен появиться неиспользуемый виртуальный жесткий диск:



11. Перейдем в настройки виртуального жесткого диска (кнопка **“Edit”**) и добавим его в используемые диски виртуальной машины (кнопка **“Add”**):



12. Также убедитесь, что в настройках виртуальной машины новый диск из п. 11 используется в качестве загрузочного диска:

Edit: Boot Order

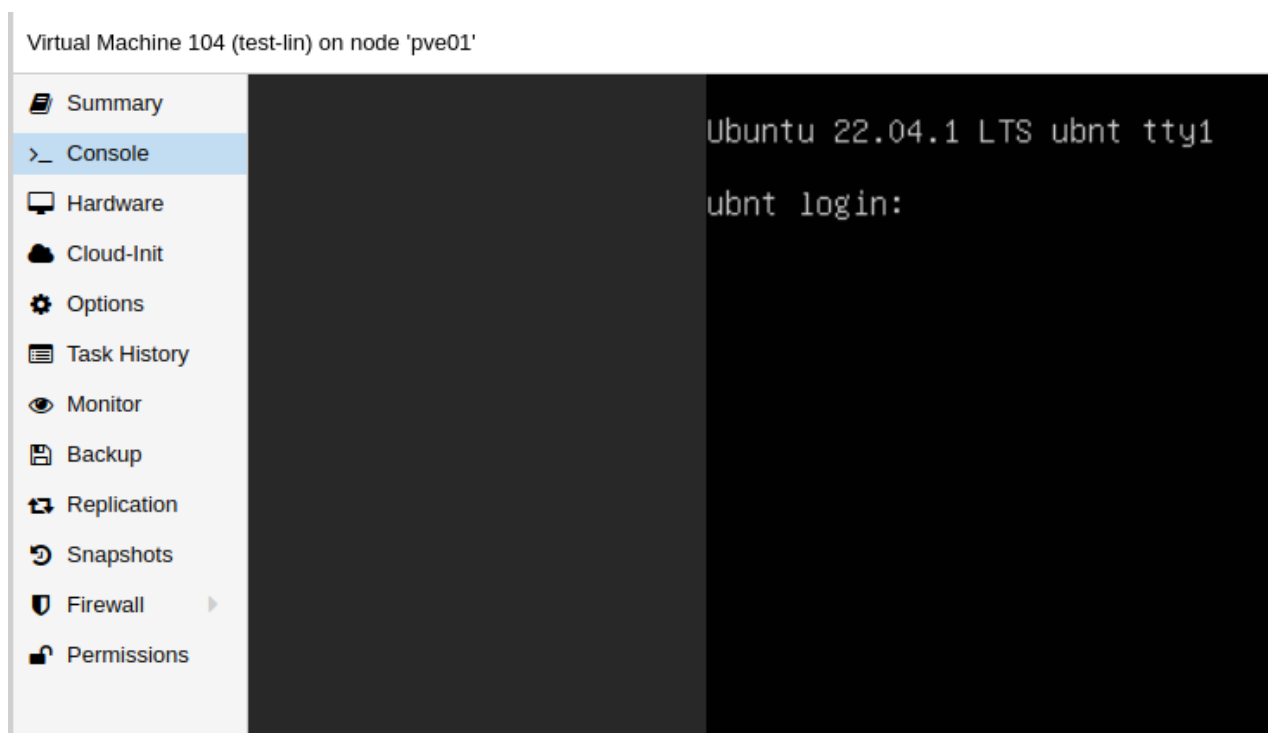
#	Enabled	Device	Description
1	<input type="checkbox"/>	ide2	none,media=cdrom
2	<input type="checkbox"/>	net0	virtio=12:0D:94:78:19:CF,bridge=vmbr0,firewall=1
3	<input checked="" type="checkbox"/>	scsi0	NVMe:104/vm-104-disk-1.raw,size=60G

Drag and drop to reorder

? Help

OK
Reset

13. Вот теперь вы можете попробовать запустить виртуальную машину на сервере Proxmox:



По крайней мере в моем случае результат был положительный. Единственный нюанс с которым я столкнулся – это то, что сетевой интерфейс был переименован с eth0 на ens18. Соответственно, нужно было скорректировать файл с настройками сети и изменить имя адаптера на ens18:

```
nano /etc/netplan/00-installer-config.yaml
```

И изменить имя сетевого адаптера:


```
GNU nano 6.2
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens18:
      dhcp4: true
  version: 2
```

И применить изменения в конфигурации сети:

```
netplan apply
```

После этого виртуальная машина успешно получила IP адрес от сервера DHCP:

```
root@ubnt:/home/roman# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 12:0d:94:78:19:cf brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.10.10.24/24 metric 100 brd 10.10.10.255 scope global dynamic ens18
        valid_lft 590sec preferred_lft 590sec
    inet6 fe80::100d:94ff:fe78:19cf/64 scope link
        valid_lft forever preferred_lft forever
```

Для ОС Windows

В целом миграция виртуальных машин Windows выполняется аналогично миграции виртуальных машин Linux. Но есть ряд нюансов. Собственно, именно нюансам и будет посвящен этот раздел.

Я лишь очень верхнеуровнево опишу процесс миграции виртуальных машин Windows, но буду подробно останавливаться на тех шагах, где есть отличия от платформы Linux:

1. Как и для платформы Linux сначала создаем виртуальную машину без дисков. Отличия будут только на шаге выбора типа гостевой операционной системы:

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

☐ Use CD/DVD disc image file (iso)
 Storage: HDD
 ISO image:

☐ Use physical CD/DVD Drive
☒ Do not use any media

Guest OS:
 Type: Microsoft Windows
 Version: 10/2016/2019

Очень важный момент – выбор типа SCSI контроллера. В Windows из коробки нет драйвера для virtio. Поэтому я укажу другой тип контроллера – LSI 53C895A.

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

Graphic card: Default
 Machine: q35
 Firmware
 BIOS: OVMF (UEFI)
 Add EFI Disk: ☒
 EFI Storage: NVMe
 Format: QEMU image format (qcow2)
 Pre-Enroll keys: ☒

SCSI Controller: Default (LSI 53C895A)
 Qemu Agent: ☐
 Add TPM: ☐

Так же на этапе выбора модели сетевого адаптера укажите Intel E1000

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

☐ No network device

Bridge: vmbr0
 VLAN Tag: no VLAN
 Firewall: ☒

Model: Intel E1000
 MAC address: auto

Причины аналогичны – в Windows из коробки нет поддержки адаптеров типа VirtIO.

2. Импортируем виртуальный жесткий диск:

```
qm importdisk 105 /mnt/pve/HDD/tst-win.vhdx NVMe
```

```
transferred 59.0 GiB of 60.0 GiB (98.26%)
transferred 59.6 GiB of 60.0 GiB (99.26%)
transferred 60.0 GiB of 60.0 GiB (100.00%)
transferred 60.0 GiB of 60.0 GiB (100.00%)
Successfully imported disk as 'unused0:NVMe:105/vm-105-disk-1.raw'
root@pve01:~#
```

3. Подключаем импортированный виртуальный жесткий диск.

Add: Unused Disk

Disk Bandwidth

Bus/Device: SATA 0 Cache: Default (No cache)

Disk image: NVMe:105/vm-105-disk-1.raw Discard: ☐

Help Advanced ☐ Add

Не забудем в параметрах очередности загрузки выбрать подключенный диск:

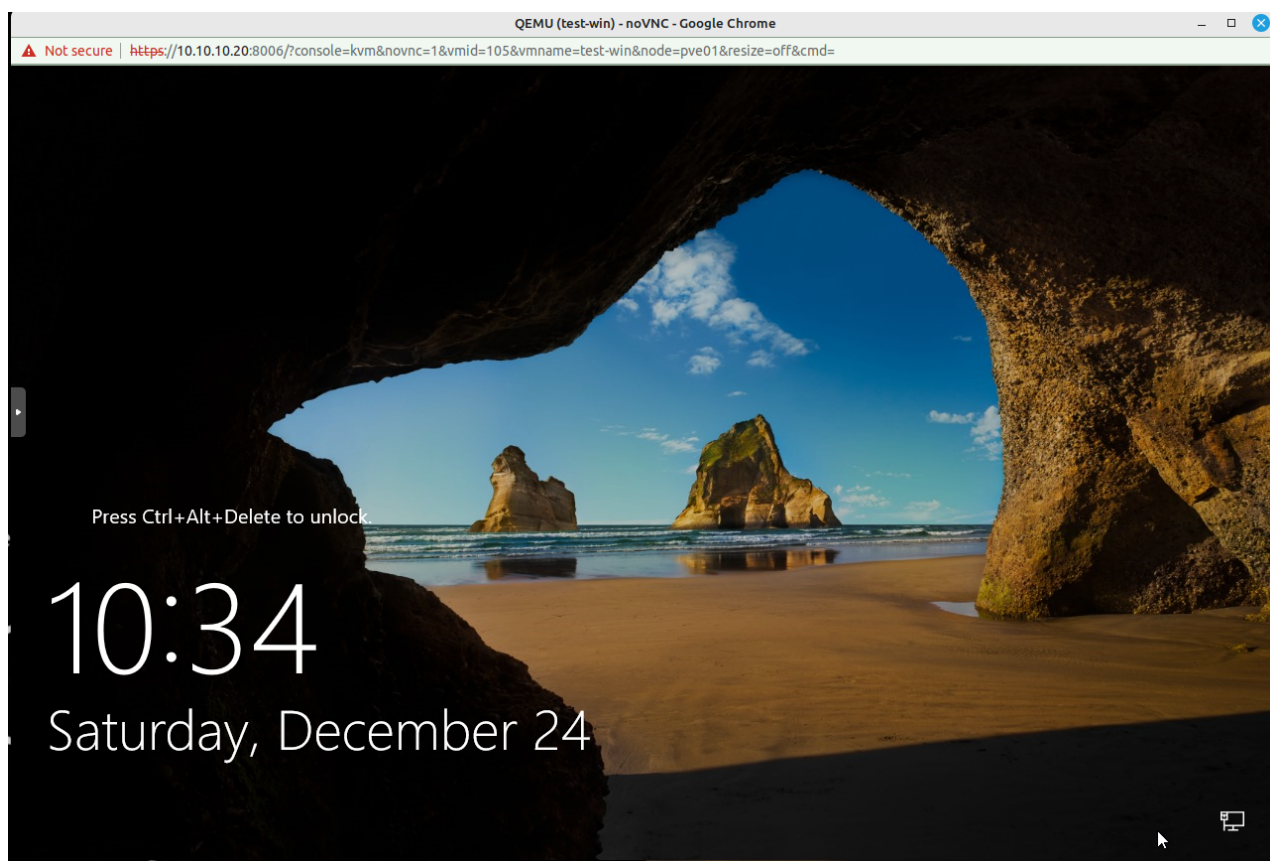
Edit: Boot Order

#	Enabled	Device	Description
1	<input type="checkbox"/>	ide2	none,media=cdrom
2	<input type="checkbox"/>	net0	e1000=C6:74:46:AC:55:CA,bridge=vmbr0,firewall=1
3	<input checked="" type="checkbox"/>	sata0	NVMe:105/vm-105-disk-1.raw,size=60G

Drag and drop to reorder

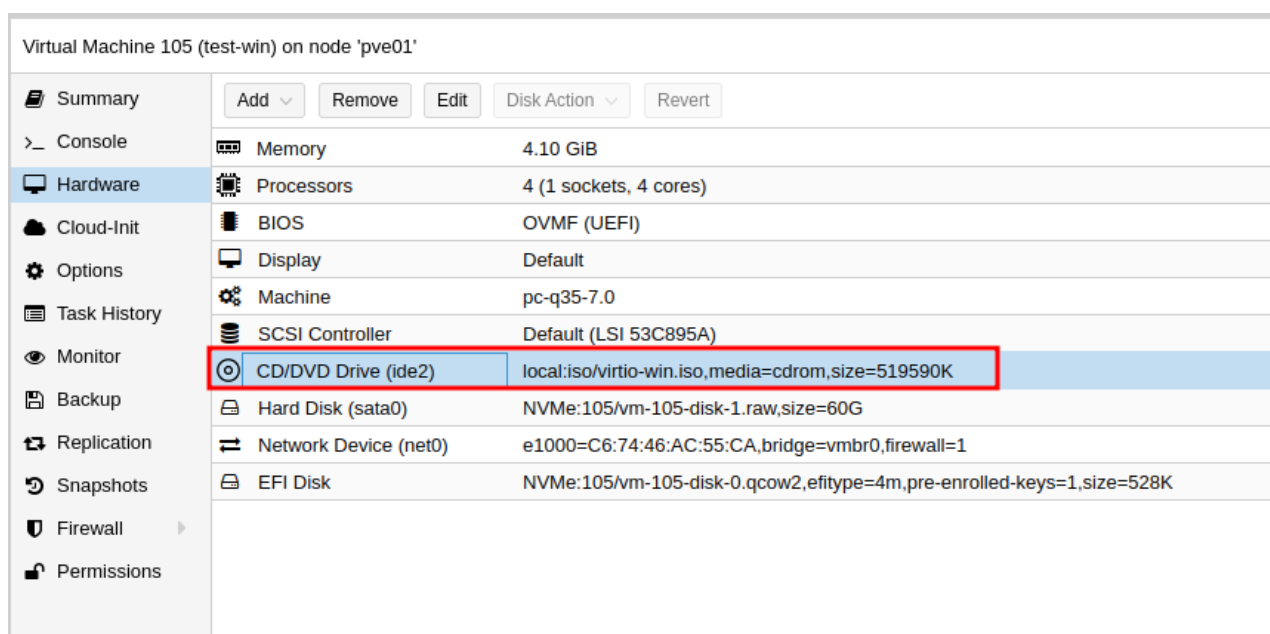
Help OK Reset

4. Запускаем виртуальную машину.







5. Если запуск прошел успешно, то теперь необходимо установить драйвер VirtIO. ISO образ с драйвером для Windows можно загрузить вот по этой ссылке. Можете либо непосредственно выполнить загрузку в гостевой ОС, либо выполнить загрузку через в локальное хранилище Proxmox, а затем примонтировать ISO образ в гостевую ОС. Я буду использовать второй метод – через загрузку в локальное хранилище Proxmox и подключение ISO в гостевую ОС.

6. Примонтирую ISO образ.



7. Установим драйвер VirtIO.

	virtio-win_license	10/23/2022 6:34 PM
	virtio-win-gt-x64	10/23/2022 7:03 PM
	virtio-win-gt-x86	10/23/2022 7:03 PM
	virtio-win-guest-tools	10/23/2022 7:03 PM

8. Меняем тип контроллера SCSI на VirIO.

Virtual Machine 105 (test-win) on node 'pve01'		
Summary	Add	Remove Edit Disk Action Revert
Console	Memory	4.10 GiB
Hardware	Processors	4 (1 sockets, 4 cores)
Cloud-Init	BIOS	OVMF (UEFI)
Options	Display	Default
Task History	Machine	pc-q35-7.0
Monitor	SCSI Controller	VirtIO SCSI
Backup	Hard Disk (ide0)	NVMe:105/vm-105-disk-1.raw,size=60G
Replication	CD/DVD Drive (ide2)	local:iso/virtio-win.iso,media=cdrom,size=519590K
Snapshots	Network Device (net0)	e1000=C6:74:46:AC:55:CA,bridge=vbr0,firewall=1
Firewall	EFI Disk	NVMe:105/vm-105-disk-0.qcow2,efitype=4m,pre-enrolled-keys=1,size=528K
Permissions		

И снова запускаем виртуальную машину. Проверяем, что ВМ запустилась успешно.

9. Так же можем поменять тип сетевого адаптера на VirtIO. Но нужна будет перезагрузка ВМ.

Virtual Machine 105 (test-win) on node 'pve01'		
Summary	Add	Remove Edit Disk Action Revert
Console	Memory	4.10 GiB
Hardware	Processors	4 (1 sockets, 4 cores)
Cloud-Init	BIOS	OVMF (UEFI)
Options	Display	Default
Task History	Machine	pc-q35-7.0
Monitor	SCSI Controller	VirtIO SCSI
Backup	Hard Disk (ide0)	NVMe:105/vm-105-disk-1.raw,size=60G
Replication	CD/DVD Drive (ide2)	local:iso/virtio-win.iso,media=cdrom,size=519590K
Snapshots	Network Device (net0)	virtio=C6:74:46:AC:55:CA,bridge=vbr0,firewall=1
Firewall	EFI Disk	NVMe:105/vm-105-disk-0.qcow2,efitype=4m,pre-enrolled-keys=1,size=528K
Permissions		

Надеюсь, что в вашем случае, как и у меня миграция будет завершена успешно.

Если что-то пошло не так

Думаю, что с конвертацией виртуальных машин Linux проблем возникнуть не должно. Могут быть нюансы с Windows. В случае каких-то затруднений я могу порекомендовать ознакомиться [вот с этим руководством](#). Вероятнее всего там вы сможете найти ответы на какие-то ваши вопросы или ошибки.