

Установка Mikrotik CHR на виртуальную машину Proxmox

 interface31.ru/tech_it/2021/04/ustanovka-mikrotik-chr-na-virtualnuyu-mashinu-proxmox.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Установка Mikrotik CHR на виртуальную машину Proxmox

Не так давно мы рассказывали о виртуальном роутере Mikrotik CHR (Cloud Hosted Router), который представляет собой специальную версию RouterOS для виртуальных сред. Это дает возможность использовать все возможности Mikrotik без оглядки на производительность и физические ограничения роутеров официальной линейки. Сегодня мы поговорим о том, как установить CHR на виртуальную машину Proxmox, ничего сложного в этом процессе нет и даже есть соответствующая статья в официальной Wiki, но гладко было на бумаге...



Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на углубленном курсе по администрированию MikroTik. Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

Изначально мы не собирались писать эту статью. Какой смысл переписывать то, что есть в официальной документации, тем более что там представлен готовый скрипт. Как говорится: нажми на кнопку и получи результат. Хотя сами мы его никогда не использовали, предпочитая все делать вручную. На самом деле процесс несложный: скачать и сконвертировать образ диска, затем создать виртуальную машину с ним. Но недавно с таким вопросом к нам обратился коллега и мы дали ему ссылку на официальную документацию.

А вот дальше все пошло не так. Скрипт оказался фактически нерабочим, кроме того, содержащим достаточно грубые ошибки. Начнем по порядку.

URL для скачивания образа диска указан неверно, при попытке перехода по нему нас ждет сообщение:

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
```

Но скрипт игнорирует ошибку получения образа и продолжает свою работу, создавая виртуальную машину без жесткого диска.

Следующая ошибка - перед предложением указать свободный VM ID выводится список только виртуальных машин, существующие контейнеры игнорируются. И хотя Proxmox не даст создать еще одну виртуалку с существующим VM ID, но мусор в виде образа диска в системе останется.

После указания VM ID проверка на существование виртуальной машины производится некорректно, проверяется наличие каталога с образом диска, в то время как современные виртуалки могут использовать Thin LVM. Правильно будет проверять наличие конфигурационного файла виртуалки. При этом логика самой проверки неверна, установив наличие существующего образа скрипт предложит ввести новое значение VM ID, но новое значение проверяться никак не будет. Т.е. можно два раза указать неверное значение и скрипт продолжит свою работу.

С учетом всего вышесказанного мы серьёзно переработали скрипт и добавили в него новые функции. Наш скрипт выполняет следующие действия:

- Скачивает образ Mikrotik CHR указанной версии и проверяет успешность этого действия
- Выводит список всех виртуальных машин и контейнеров
- Проверяет введенный VM ID на существование виртуальной машины или контейнера с таким идентификатором
- Конвертирует образ и изменяет размер виртуального жесткого диска на указанный пользователем (новая функция)
- Создает виртуальную машину с нужным образом диска

При возникновении ошибок на любом из этапов скрипт сообщает об ошибке и завершает свою работу.

```
#!/bin/bash

#vars
version="nil"
vmID="nil"

echo"##### Start of Script #####

## Checking if temp dir is available..."
if [ -d /root/temp ]
then
    echo "-- Directory exists!"
else
    echo "-- Creating temp dir!"
    mkdir /root/temp
fi
# Ask user for version
echo "## Preparing for image download and VM creation!"
read -p "Please input CHR version to deploy (6.38.2, 6.40.1, etc):" version
# Check if image is available and download if needed
if [ -f /root/temp/chr-$version.img ]
then
    echo "-- CHR image is available."
else
    echo "-- Downloading CHR $version image file."
    cd /root/temp
    echo "-----"
    --"
    wget --no-check-certificate
https://download.mikrotik.com/routeros/$version/chr-$version.img.zip
    unzip chr-$version.img.zip
    echo "-----"
    --"
fi
if [ ! -f /root/temp/chr-$version.img ]
then
    echo "-- Error downloading CHR $version image file!"
    exit 0
fi
# List already existing VM's and ask for vmID
echo "== Printing list of VM's on this hypervisor!"
qm list
echo "== Printing list of CT's on this hypervisor!"
pct list
echo ""
read -p "Please Enter free vm ID to use:" vmID
echo ""
# Create storage dir for VM if needed.
if [ -f /etc/pve/nodes/pve/qemu-server/$vmID.conf ]
then
```

```

    echo "-- VM exists! Try another vm ID!"
    exit 0
fi
if [ -f /etc/pve/nodes/pve/lxc/$vmID.conf ]
then
    echo "-- CT exists! Try another vm ID!"
    exit 0
fi
echo "-- Creating VM image dir!"
mkdir /var/lib/vz/images/$vmID
# Creating qcow2 image for CHR.
read -p "Please input image size, GB:" imgsize
echo "-- Converting image to qcow2 format "
qemu-img convert \
    -f raw \
    -O qcow2 \
    /root/temp/chr-$version.img \
    /var/lib/vz/images/$vmID/vm-$vmID-disk-1.qcow2
if [ $imgsize -ne 0 ]
then
    echo "-- Resize image to $imgsize GB"
    qemu-img resize \
        /var/lib/vz/images/$vmID/vm-$vmID-disk-1.qcow2 +${imgsize}G
fi
# Creating VM
echo "-- Creating new CHR VM"
qm create $vmID \
    --name chr-$version \
    --net0 virtio,bridge=vmbro \
    --bootdisk virtio0 \
    --ostype l26 \
    --memory 256 \
    --onboot no \
    --sockets 1 \
    --cores 1 \
    --virtio0 local:$vmID/vm-$vmID-disk-1.qcow2
echo "##### End of Script #####"
```

Разберем его работу подробнее.

Секция **## Checking if temp dir is available...** проверяет наличие временной директории **/root/temp** и создает ее в случае отсутствия. Затем у пользователя запрашивается номер требуемой версии RouterOS и производится попытка ее скачать. Кроме исправленного адреса мы добавили опцию **--no-check-certificate** для **wget**, что позволяет исключить ошибки с проверкой валидности сертификата.

Затем производится проверка успешности скачивания и распаковки образа:

```
if [ ! -f /root/temp/chr-$version.img ]
```

Если файл образа не существует - скрипт завершит свою работу.

```

root@pve:~# ./chr_install.sh
##### Start of Script #####

## Checking if temp dir is available...
-- Directory exists!
## Preparing for image download and VM creation!
Please input CHR version to deploy (6.38.2, 6.40.1, etc):6.99.1
-- Downloading CHR 6.99.1 image file.
-----
--2021-04-03 19:52:30-- https://download.mikrotik.com/routers/6.99.1/chr-6.99.1.img.zip
Resolving download.mikrotik.com (download.mikrotik.com)... 159.148.172.226, 159.148.147.204, 2a02:610:7501:4000::226, ...
Connecting to download.mikrotik.com (download.mikrotik.com)[159.148.172.226]:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2021-04-03 19:52:30 ERROR 404: Not Found.

unzip: cannot find or open chr-6.99.1.img.zip, chr-6.99.1.img.zip.zip or chr-6.99.1.img.zip.ZIP.
-----
-- Downloading CHR 6.99.1 error!

```

В случае успеха он выведет пользователю список не только виртуальных машин, но и контейнеров:

```

echo "== Printing list of VM's on this hypervisor!"
qm list
echo "== Printing list of CT's on this hypervisor!"
pct list

```

После ввода желаемого VM ID происходит проверка наличия конфигурационных файлов, как для виртуалок, так и для контейнеров:

```

if [ -f /etc/pve/nodes/pve/qemu-server/$vmID.conf ]
then
    echo "-- VM exists! Try another vm ID!"
    exit 0
fi
if [ -f /etc/pve/nodes/pve/lxc/$vmID.conf ]
then
    echo "-- CT exists! Try another vm ID!"
    exit 0
fi

```

Если для указанного VM ID будет найден конфигурационный файл, скрипт сообщит что такая машина или контейнер существуют и завершит свою работу.

```

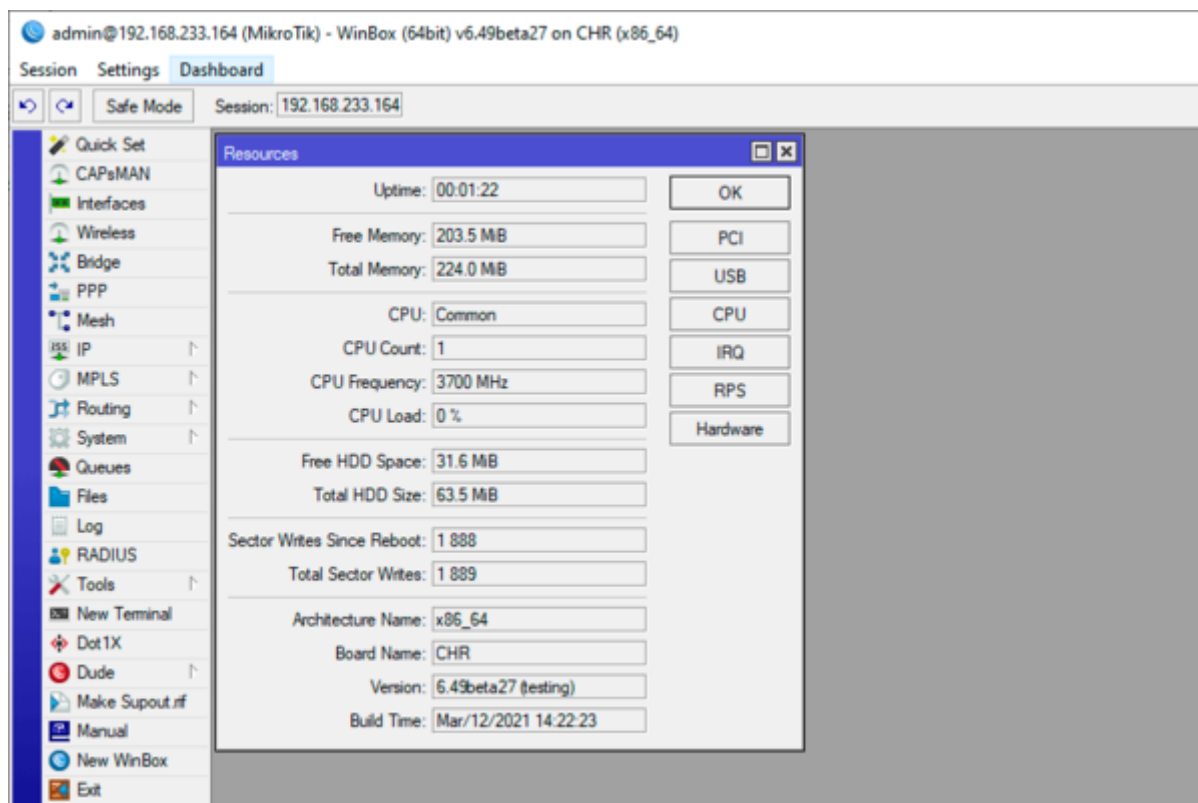
root@pve:~# ./chr_install.sh
##### Start of Script #####

## Checking if temp dir is available...
-- Directory exists!
## Preparing for image download and VM creation!
Please input CHR version to deploy (6.38.2, 6.40.1, etc):6.47.9
-- CHR image is available.
== Printing list of VM's on this hypervisor!
      VMID NAME                STATUS    MEM(MB)    BOOTDISK(GB) PID
      100 VM-1                  running    2048        32.00 1208
      103 chr-6.48.1            stopped    256         0.00 0
== Printing list of CT's on this hypervisor!
VMID    Status    Lock      Name
101     running   CT-1
102     running   CT102

Please Enter free vm ID to use:100
-- VM exists! Try another vm ID!

```

После прохождения всех проверок, казалось бы, можно создавать виртуальную машину, но не будем спешить. По умолчанию образ диска имеет объем всего 64 МБ, что во многих случаях будет явно недостаточно.



Поэтому мы добавили в скрипт новую функцию - установку нужного размера виртуального диска. Требуется указать нужное значение в ГБ или ноль, если вы не хотите увеличивать диск.

```
read -p "Please input image size, GB:" imgsize
```

После чего будет выполнена конвертация образа, установка указанного размера и создание виртуальной машины.

Все последующие действия требуется выполнять с правами суперпользователя root.

Сначала создадим сам файл скрипта в корневой директории суперпользователя:

```
touch /root/CHR_install.sh
```

И внесем в него приведенный выше текст, также можно скачать готовый скрипт с нашего сайта:

[CHR_install.sh](#) MD5: 47E29D66B33955BCFB94E6149B823CB2

Например, это можно сделать командой:

```
cd /root
wget --no-check-certificate https://interface31.ru/tech_it/files/CHR_install.sh
```

Затем, вне зависимости от того, каким образом вы создали скрипт, сделаем его исполняемым:

```
chmod +x /root/CHR_install.sh
```

Теперь его можно запускать, для этого, если вы находитесь в /root выполните:

```
./CHR_install.sh
```

Из иных расположений:

```
/root/CHR_install.sh
```

В ходе работы скрипта вам потребуется ответить на три вопроса: указать версию RouterOS так как она указана на сайте (6.48.1 или 6.49beta27), номер идентификатора виртуальной машины (VM ID) и желаемый размер виртуального диска в ГБ:

```
root@pve:~# ./chr_install.sh
##### Start of Script #####

## Checking if temp dir is available...
-- Directory exists!
## Preparing for image download and VM creation!
Please input CHR version to deploy (6.38.2, 6.48.1, etc):6.48.1
-- CHR image is available.
== Printing list of VM's on this hypervisor!
  VMID NAME                STATUS  MEM(MB)  BOOTDISK(GB) PID
   100 VM-1                 running  2048      32.00  1208
   105 chr-6.49beta27       running   256        0.00  49638
   106 chr-6.47.9          stopped   256        0.00    0
   110 chr-6.48.1          running   256        0.00  49188
== Printing list of CT's on this hypervisor!
 VMID  Status  Lock  Name
  101   running  CT-1
  102   running  CT102

Please Enter free vm ID to use:103

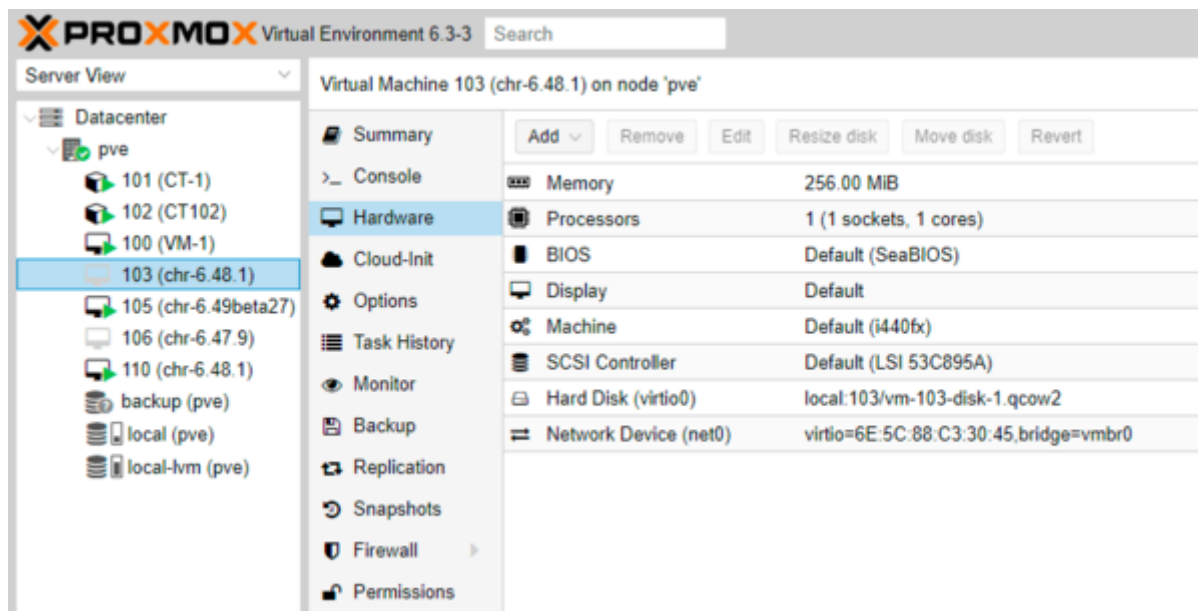
-- Creating VM image dir!
Please input image size, GB:8
-- Converting image to qcow2 format
-- Resize image to 8 GB
Image resized.
-- Creating new CHR VM
##### End of Script #####
root@pve:~# |
```

Версия RouterOS

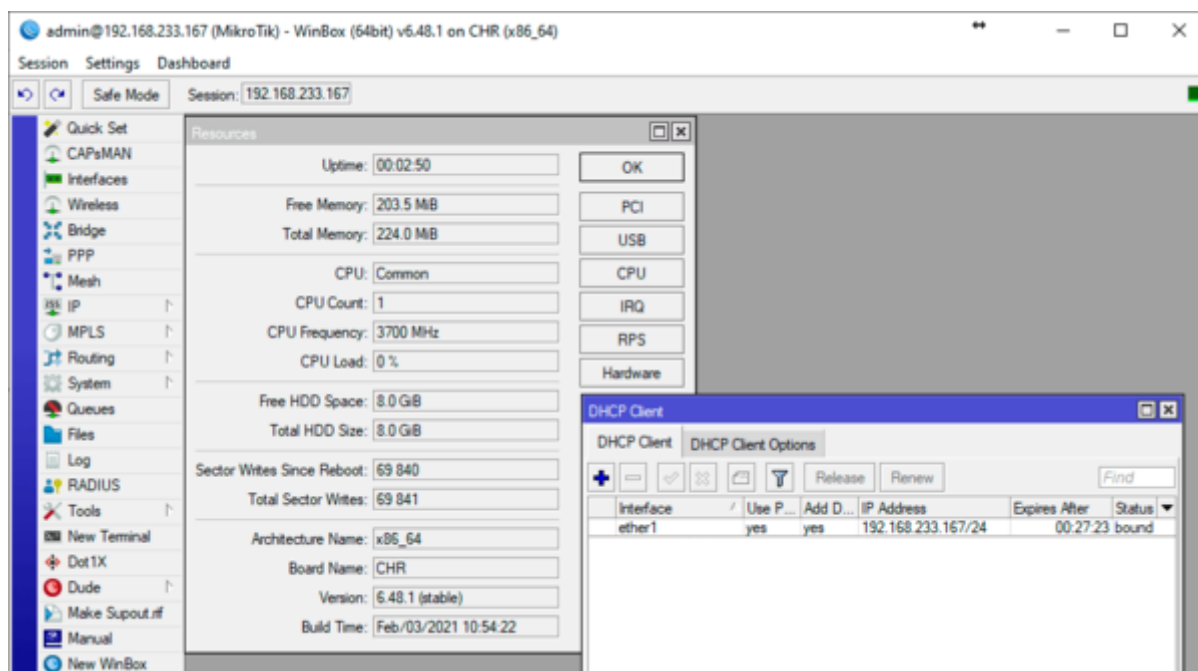
VM ID

Размер VHD в ГБ

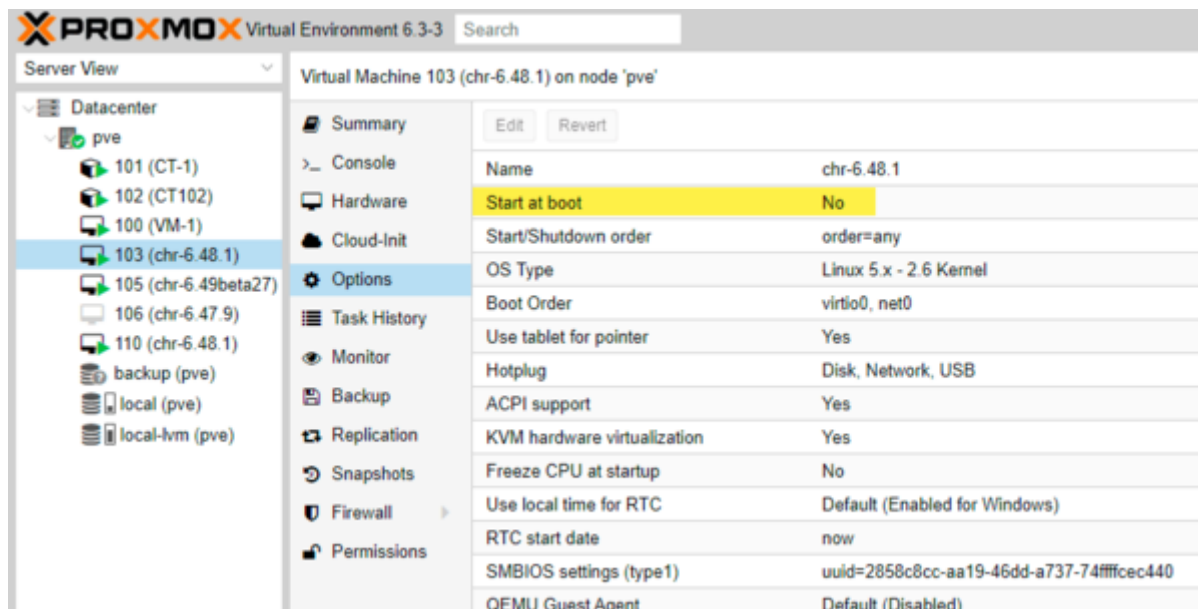
Результатом работы скрипта будет созданная виртуальная машина, перед запуском вы можете проверить и изменить ее параметры:



Обратите внимание, что скрипт создает единственный сетевой интерфейс, который подключает к **vbr0**. Сама же RouterOS настроена таким образом, что на интерфейсе **ether1** включен DHCP-клиент и при наличии в сети DHCP-сервера автоматически получит от него адрес. Однако вы можете настроить любую иную сетевую конфигурацию. Более подробно читайте в нашей статье о [настройке сети в Proxmox VE](#).



Также не забудьте проверить и включить автоматический запуск виртуальной машины, по умолчанию он выключен:



Если же вам нужно расширить виртуальный диск уже существующей машины, то ее следует выключить и выполнить следующую команду:

```
qemu-img resize /var/lib/vz/images/105/vm-105-disk-1.qcow2 +4G
```

В данном случае мы увеличили диск виртуальной машины с идентификатором 105 на 4 ГБ.