

Установка MikroTik RouterOS на VDS/VPS

 interface31.ru/tech_it/2024/11/ustanovka-mikrotik-routeros-na-vdsvps.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Установка MikroTik RouterOS на VDS/VPS

Cloud Hosted Router - специальный продукт компании Mikrotik позволяющий установить MikroTik RouterOS практически на любую виртуальную машину. Однако не все хостеры позволяют загружать собственные образы, но это не страшно, RouterOS можно поставить на любой VPS с Linux, о чем мы и расскажем в данной статье. Также уделим отдельное внимание UEFI-системам и предварительной настройке образа, позволяющей избежать потенциальных проблем с безопасностью и уменьшить количество ручной работы, а также сразу получить требуемую конфигурацию RouterOS.



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.

Для установки нам понадобится виртуальная машина (VPS/VDS) с установленной операционной системой Linux, дистрибутив роли не играет, нам подойдет любой, но в нашей статье мы будем рассматривать работу с Debian или Ubuntu. Что касается ресурсов, то смотрите по предполагаемой нагрузке, в большинстве случаев будет достаточно самого недорого тарифа. Важное условие при покупке - уточните условия тарификации трафика, не все хостеры предоставляют безлимит, у некоторых (преимущественно американских) есть лимиты, по превышению которых придется либо доплатить за перерасход, либо у вас будет снижена скорость канала.

Но будем считать что вы с этим вопросом разобрались и перейдем непосредственно к установке RouterOS на виртуальную машину с Linux. Прежде всего повысим права до суперпользователя, в Debian для этого используйте:

```
su -
```

А в Ubuntu:

```
sudo -s
```

После чего нам потребуется выяснить несколько моментов, прежде всего сетевые настройки. Это важно, так как у многих хостеров виртуальные машины могут иметь адреса из внутренней, серой сети, а выделенный IP-адрес назначается на оборудовании провайдера, поэтому нам надо будет настроить сетевые параметры точно так, как они были на исходной виртуальной машине.

Чтобы узнать IP-адрес и маску сети выполните:

```
ip a
```

А для того чтобы узнать адрес шлюза (он будет указан в маршруте по умолчанию) выполните:

```
ip r
```

```
root@vps-1234:~# ip a
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 noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:4b:83:99 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.233.117/24 brd 192.168.233.255 scope global dynamic ens33
        valid_lft 1240sec preferred_lft 1240sec
    inet6 fe80::20c:29ff:fe4b:8399/64 scope link
        valid_lft forever preferred_lft forever
root@vps-1234:~# ip r
default via 192.168.233.2 dev ens33
192.168.233.0/24 dev ens33 proto kernel scope link src 192.168.233.117
root@vps-1234:~# |
```

Следующим шагом нам предстоит выяснить название виртуального диска и тип загрузчика, для этого наберите команду:

```
fdisk -l
```

В выводе найдите имя диска и посмотрите на тип загрузочного раздела, в нашем случае виртуальная машина имеет UEFI-загрузчик.

```

root@vps-1234:~# fdisk -l
Disk /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 3D99AE77-D4FD-4865-8DA8-B705A8F33039

Device            Start       End   Sectors  Size Type
/dev/sda1         2048      1050623  1048576  512M EFI System
/dev/sda2      1050624  18970623  17920000  8.5G Linux filesystem
/dev/sda3      18970624  20969471   1998848  976M Linux swap
root@vps-1234:~#

```

Если у вас обычный загрузчик, то дополнительно понадобится установить утилиту **unzip**:

```
apt install unzip
```

Затем на официальном сайте в разделе загрузок скопируйте ссылку на **Raw disk image** нужной версии **Cloud Hosted Router** и скачайте его командой:

```
wget https://download.mikrotik.com/routeros/7.16.1/chr-7.16.1.img.zip
```

И затем распакуйте:

```
unzip chr-7.16.1.img.zip
```

Обратите внимание, что здесь и далее используются наименования файла образа определенной версии, в нашем случае 7.16.1, вам потребуется его изменить согласно наименованию файла скачанной вами версии.

Если у вас UEFI-версия загрузчика, то официального образа для него нет, но можно воспользоваться неофициальным, скачав его с GitHub-страницы разработчика. Мы будем качать также версию 7.16.1 и нам понадобится файл в формате **.raw**. Точно также копируем ссылку и скачиваем командой:

```
wget https://github.com/tikoci/fat-chr/releases/download/Build11294119639-jac laz/chr-7.16.1.uefi-fat.raw
```

Как правило, множественные инструкции в сети после этого предлагают сразу развернуть образ на диск, а потом подключиться к виртуальной машине через KVM или VNC-консоль хостера и настроить через нее сетевые параметры RouterOS и сменить пароль администратора. Но такая услуга есть по умолчанию не у всех хостеров и не на всех тарифах. Где-то ее нужно заказывать отдельно, где-то через поддержку, также могут быть проблемы с самой консолью, например, с раскладкой.

Поэтому мы пойдем другим путем и заранее настроим все нужные параметры образа. Для этого мы воспользуемся Autogun-файлом в который внесем все необходимые команды и они будут автоматически выполнены при первом старте RouterOS. Для этого нам нужно примонтировать корневой раздел образа, чтобы это сделать нам нужно вычислить его смещение от начала виртуального диска в байтах.

Сделать это не сложно, выполните команду (далее мы будем работать с UEFI-образом):

```
fdisk -lu chr-7.16.1.uefi-fat.raw
```

В выводе нас интересует **размер сектора** - 512 байт и начало второго раздела **Linux filesystem** - 65570, которое указано в колонке **Start**.

```
root@vps-1234:~# fdisk -lu chr-7.16.1.uefi-fat.raw
Disk chr-7.16.1.uefi-fat.raw: 128 MiB, 134217728 bytes, 262144 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7B600006-0051-404F-8046-0A61850A1FD4

Device                Start      End Sectors Size Type
chr-7.16.1.uefi-fat.raw1    34    65569    65536 32M EFI System
chr-7.16.1.uefi-fat.raw2 65570 258047   192478 94M Linux filesystem
root@vps-1234:~# |
```

Так как fdisk сообщает нам номер первого сектора раздела, то для получения смещения в байтах нам нужно умножить номер первого сектора на размер сектора:

$$512 * 65570 = 33571840$$

Теперь выполним монтирование раздела командой:

```
mount -o loop,offset=33571840 chr-7.16.1.uefi-fat.raw /mnt
```

После чего откроем на редактирование файл **autorun.scr**:

```
nano /mnt/rw/autorun.scr
```

И добавим в него команды настройки сетевого интерфейса и шлюза:

```
/ip address
add address=192.168.233.117/24 interface=ether1 network=192.168.233.0
/ip route
add gateway=192.168.233.2
```

Затем изменим пароль встроенному пользователю **admin**:

```
/user set 0 password="MyPa$$word"
```

На этом можно было бы и остановиться, но почему бы сразу не выполнить ряд дополнительных настроек, особенно если это касается безопасности. Так как наш роутер будет смотреть сразу в интернет, то отключим все не нужные IP-сервисы, в нашем случае мы оставили только **ssh** и **winbox**:

```
/ip service
set telnet disabled=yes
set ftp disabled=yes
set www disabled=yes
set api disabled=yes
set api-ssl disabled=yes
```

Также отключим поиск соседства:

```
/ip neighbor discovery-settings
set discover-interface-list=none
```

И все MAC-сервисы:

```
/tool mac-server
set allowed-interface-list=none
/tool mac-server mac-winbox
set allowed-interface-list=none
/tool mac-server ping
set enabled=no
```

Вообще, при помощи этого файла вы можете сразу задать всю необходимую конфигурацию роутера, получив на выходе уже полностью готовое устройство. Мы же ограничимся перечисленным выше. Сохраним файл и отмонтируем образ:

```
umount /mnt
```

Размонтируем файловые системы:

```
echo u > /proc/sysrq-trigger
```

И развернем образ на виртуальный диск:

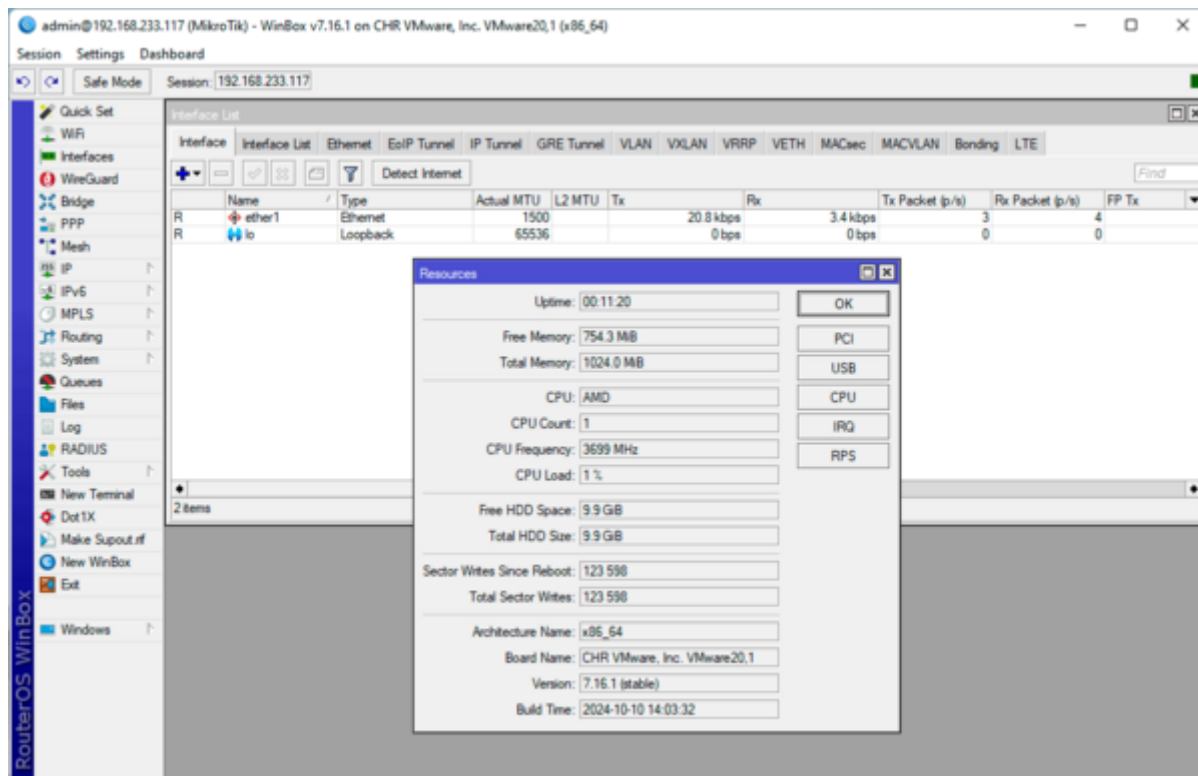
```
dd if=chr-7.16.1.uefi-fat.raw of=/dev/sda bs=4M oflag=sync
```

Обратите внимание, что мы используем флаг **sync** для обеспечения реальной физической записи на диск, а не в дисковый кеш.

И перезагружаем систему командами:

```
echo 1 > /proc/sys/kernel/sysrq
echo b > /proc/sysrq-trigger
```

После перезагрузки подключаемся к установленной RouterOS через Winbox или SSH.



Как видим, развернуть Cloud Hosted Router на виртуальную машину совсем не сложно, даже если она использует UEFI-загрузчик. А использование файла **autorun.scr** сразу позволяет задать все необходимые настройки RouterOS и исключить этап ручной работы с использованием аварийной консоли.

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.