


Настройка двух и более OpenVPN-серверов на одном сервере

 interface31.ru/tech_it/2019/10/nastroyka-dvuh-i-bolee-openvpn-serverov-na-odnom-servere.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Настройка двух и более OpenVPN-серверов на одном сервере

Продолжая рассказ о возможностях OpenVPN рассмотрим ситуацию, когда нам нужно иметь на одном сервере несколько разных серверов OpenVPN с различными настройками. Чаще всего такая необходимость возникает на серверах, предназначенных для доступа в интернет, когда имеются клиенты, требующие особых настроек. В корпоративной среде такая возможность позволит с помощью одного физического сервера обеспечить связь для различных подразделений, которые не должны напрямую взаимодействовать друг с другом. Как это сделать - читайте в нашей статье.



Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "[Архитектура современных компьютерных сетей](#)" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

Начнем с постановки задачи. В [прошлой статье](#) мы рассказали о настройке OpenVPN сервера в Oracle Cloud для доступа в интернет. Сервер работает, клиенты подключаются, но возникла необходимость подключения к нему роутеров Mikrotik, у которых, как известно, особые требования к настройкам OpenVPN. Как быть? Перенастроить сервер для совместимости с Mikrotik в ущерб остальным клиентам? Или поднять второй экземпляр сервера со своими настройками? Естественно, второй способ выглядит более предпочтительно.

Напомним, что все настройки мы производим на сервере с **Ubuntu 18.04** в облаке от Oracle, настройка которого описана в статье по ссылке выше, рекомендуем ознакомиться с ней перед прочтением данного руководства. Однако все описанные ниже действия, с соответствующими поправками, применимы к любому Linux-дистрибутиву.

Настройка второго экземпляра сервера

Прежде всего создадим для нового сервера собственный набор ключей, для этого перейдем в папку нашего центра сертификации и загрузим переменные:

```
cd /etc/openvpn/easy-rsa
source ./vars
```

После чего создадим новый серверный сертификат:

```
./build-key-server server-tcp
```

Где **server-tcp** имя нашего экземпляра сервера. Мы советуем давать осмысленные имена, чтобы вам потом было понятно, что делает тот или иной экземпляр.

Скопируем ключ и сертификат в папку с ключами OpenVPN:

```
cd /etc/openvpn/easy-rsa/keys
cp server-tcp.crt server-tcp.key /etc/openvpn/keys
```

Затем скопируем шаблон конфигурации и назовем его **server-tcp.conf**:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf
/etc/openvpn/server-tcp.conf
```

После чего откроем файл и приступим к его редактированию. Какие важные особенности нам нужно учесть? RouterOS работает с OpenVPN **только по протоколу TCP** и **не использует сжатие**, также есть и некоторые иные особенности, которых мы коснемся позже. Все опции указаны в порядке их следования в файле, если опции нет - ее нужно добавить.

```
port 1194
proto tcp
dev tun
```

Для того, чтобы несколько серверов OpenVPN могли работать на одном хосте они должны использовать разные порты. Но так как первый экземпляр работает по протоколу UDP, то для второго экземпляра мы также можем использовать порт 1194, но только с протоколом TCP.

Укажем топологию сети:

```
topology subnet
```

И пути к ключам и сертификатам:

```
ca keys/ca.crt
cert keys/server-tcp.crt
key keys/server-tcp.key
dh keys/dh2048.pem
```

Диапазон адресов внутри VPN-сети также должен отличаться от остальных экземпляров, поэтому укажем:

```
server 10.89.0.0 255.255.255.0
```

Для хранения выданных клиентам адресов, как и для логов, также следует использовать отдельные файлы:

```
ifconfig-pool-persist /var/log/openvpn/ipp-tcp.txt
```

Mikrotik игнорирует опцию настройки шлюза по умолчанию, но мы все-таки советуем добавить данную опцию, так как подключаться к данному серверу могут и иные клиенты.

```
push "redirect-gateway def1 bypass-dhcp"
```

Передадим собственные DNS:

```
push "dhcp-option DNS 208.67.222.222"  
push "dhcp-option DNS 208.67.220.220"
```

Параметры проверки активности:

```
keepalive 10 120
```

Обязательно выключим дополнительную TLS-аутентификацию:

```
#tls-auth ta.key 0
```

И укажем параметры шифра, выключив его согласование, RouterOS поддерживает только AES128/192/256 и Blowfish 128:

```
ncp-disable  
cipher AES-128-CBC
```

Обязательно отключаем все опции сжатия:

```
#compress lz4-v2  
#push "compress lz4-v2"  
#comp-lzo
```

Убеждаемся в наличии опций понижения прав:

```
user nobody  
group nogroup
```

И за сохранение доступа к ключам и адаптерам:

```
persist-key  
persist-tun
```

Укажем свой комплект файлов лога:

```
status /var/log/openvpn/openvpn-status-tcp.log  
log /var/log/openvpn/openvpn-tcp.log
```

и его подробность:

```
verb 3
```

При использовании протокола TCP обязательно прокомментируем опцию:

```
explicit-exit-notify 1
```

А также добавим:

```
tcp-nodelay
```

Сохраним файл конфигурации.

Чтобы обеспечить автоматический запуск всех серверных конфигураций OpenVPN откроем в `/etc/default/openvpn` и раскомментируем в нем строку:

```
AUTOSTART="all"
```

Затем перечитаем конфигурацию юнитов systemd:

```
systemctl daemon-reload
```

Теперь уже можно запустить наш новый экземпляр, но мы пока не будем этого делать, так как нужно перенастроить брандмауэр.

Настройка брандмауэра и маршрутизации

Очевидно, что нам нужно разрешить входящий трафик на порт OpenVPN и транзитный трафик для tun-адаптеров, также потребуется отдельное правило для маскарadingа. В итоге ваш файл `/etc/nat` должен будет выглядеть следующим образом:

```
#!/bin/sh
```

```
# Включаем форвардинг пакетов  
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
# Сбрасываем настройки брандмауэра  
iptables -F  
iptables -X  
iptables -t nat -F  
iptables -t nat -X
```

```
# Разрешаем инициированные нами подключения извне  
iptables -A INPUT -i ens3 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Разрешаем подключения по SSH  
iptables -A INPUT -i ens3 -p tcp --dport 22 -j ACCEPT
```

```
# Разрешаем подключения к OpenVPN  
iptables -A INPUT -i ens3 -p udp --dport 1194 -j ACCEPT  
iptables -A INPUT -i ens3 -p tcp --dport 1194 -j ACCEPT
```

```
#Запрещаем входящие извне  
iptables -A INPUT -i ens3 -j DROP
```

```
# Разрешаем инициированные нами транзитные подключения извне  
iptables -A FORWARD -i ens3 -o tun+ -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Запрещаем транзитный трафик извне  
iptables -A FORWARD -i ens3 -o tun+ -j DROP
```

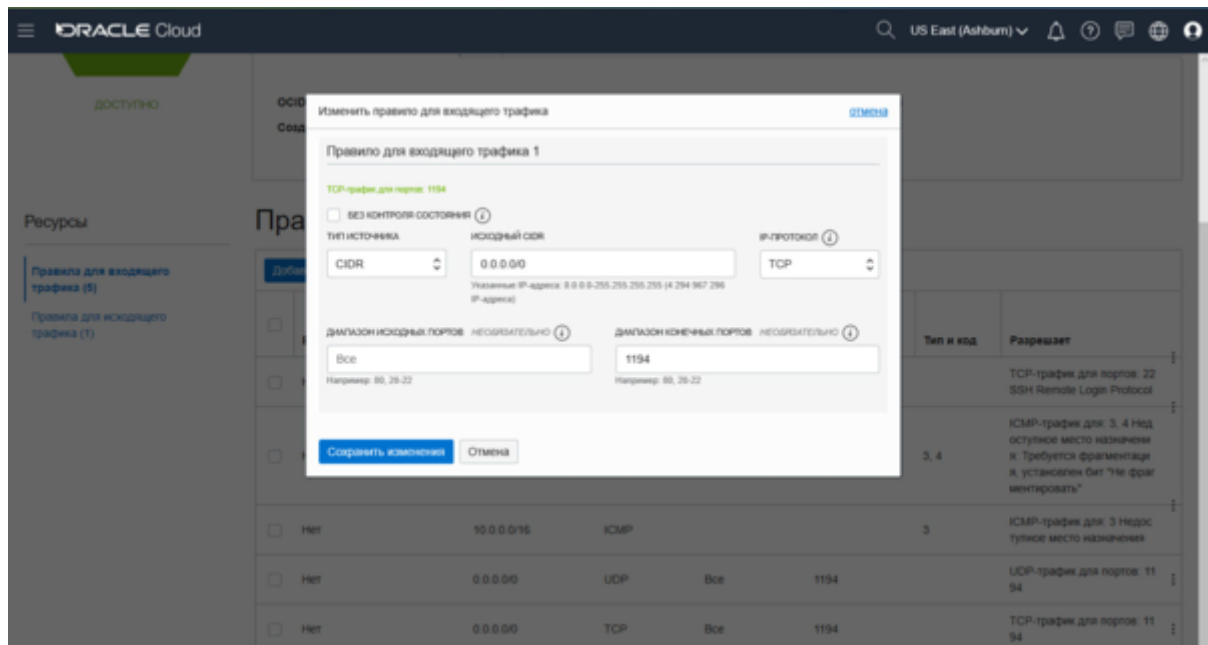
```
# Включаем маскрадинг для локальной сети  
iptables -t nat -A POSTROUTING -o ens3 -s 10.88.0.0/24 -j MASQUERADE  
iptables -t nat -A POSTROUTING -o ens3 -s 10.89.0.0/24 -j MASQUERADE
```

На что следует обратить внимание? Для каждого сервера нужно разрешающее правило в цепочке INPUT, в нашем случае в секции **Разрешаем подключения к OpenVPN** добавлено два правила для входящих UDP 1194 и TCP 1194. Аналогично

Включаем маскарадинг для локальной сети.

В правилах цепочки FORWARD мы заменили **tun0** на **tun+**, что теперь распространяет действие правил на все туннельные интерфейсы.

Если вы используете Oracle Cloud, то не забудьте создать разрешающее правило для входящих TCP 1194 в настройках вашей виртуальной сети:



Теперь можно перезапустить службу OpenVPN и убедиться, что поднялись два туннельных интерфейса:

```
systemctl restart openvpn
```

```

root@oldcopy-oracle-us-01:/etc/opensvpn# systemctl restart opensvpn
root@oldcopy-oracle-us-01:/etc/opensvpn# 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
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:00:17:06:45:b9 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 brd 10.0.0.255 scope global ens3
        valid_lft forever preferred_lft forever
    inet6 fe80::17ff:fe06:45b9/64 scope link
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.88.0.1/24 brd 10.88.0.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::5526:d2e1:d7d:f087/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
6: tun1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.89.0.1/24 brd 10.89.0.255 scope global tun1
        valid_lft forever preferred_lft forever
    inet6 fe80::325c:c612:cdde:c3ea/64 scope link stable-privacy
        valid_lft forever preferred_lft forever

```

Настройка клиентов OpenVPN

Если вы будете настраивать в качестве клиента Mikrotik, то вам потребуется только ключевая пара клиента и, опционально, сертификат CA, для проверки подлинности сервера. Для создания ключа клиента перейдите в директорию центра сертификации и загрузите переменные:

```
cd /etc/openvpn/easy-rsa
source ./vars
```

Затем выпустите сертификат клиента:

```
./build-key mikrotik
```

Полученные файлы и сертификат CA скопируем в домашнюю директорию:

```
cd /etc/openvpn/easy-rsa/keys
cp ca.crt mikrotik.crt mikrotik.key ~
```

И сменим их владельца на вашего основного пользователя, чтобы он мог спокойно скопировать их через FTP или SFTP (по умолчанию владелец файлов **root**). В нашем случае это пользователь **ubuntu**.

```
cd ~
chown ubuntu:ubuntu ca.crt mikrotik.crt mikrotik.key
```

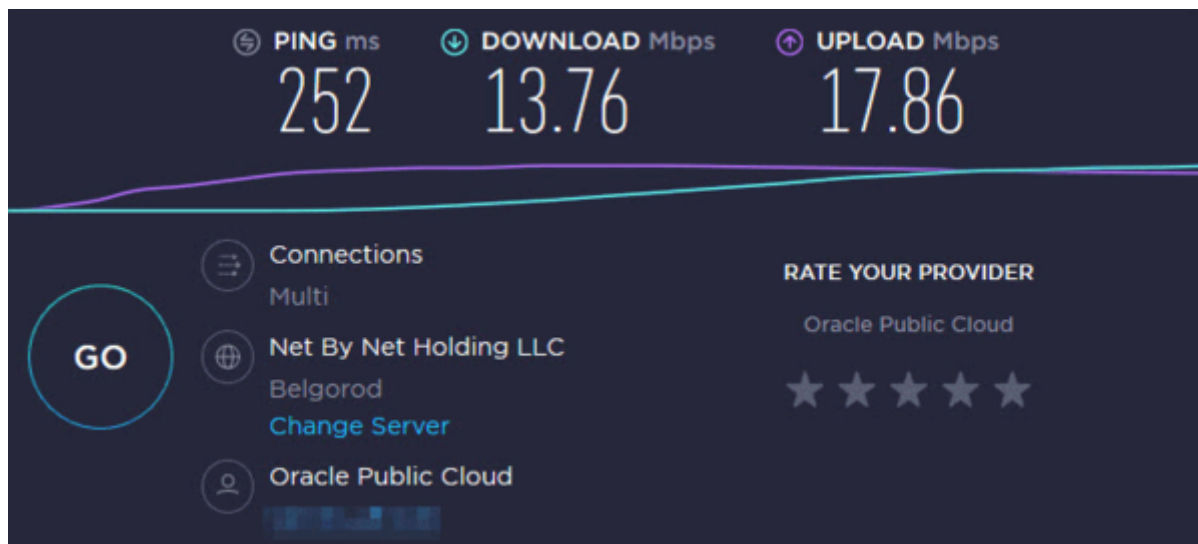
Если же будет подключаться иной клиент, то ему потребуется клиентский файл конфигурации, можете использовать как образец конфигурацию клиента созданную нами в [предыдущей статье](#). В него потребуется внести следующие изменения: изменить протокол на TCP

```
proto tcp
```

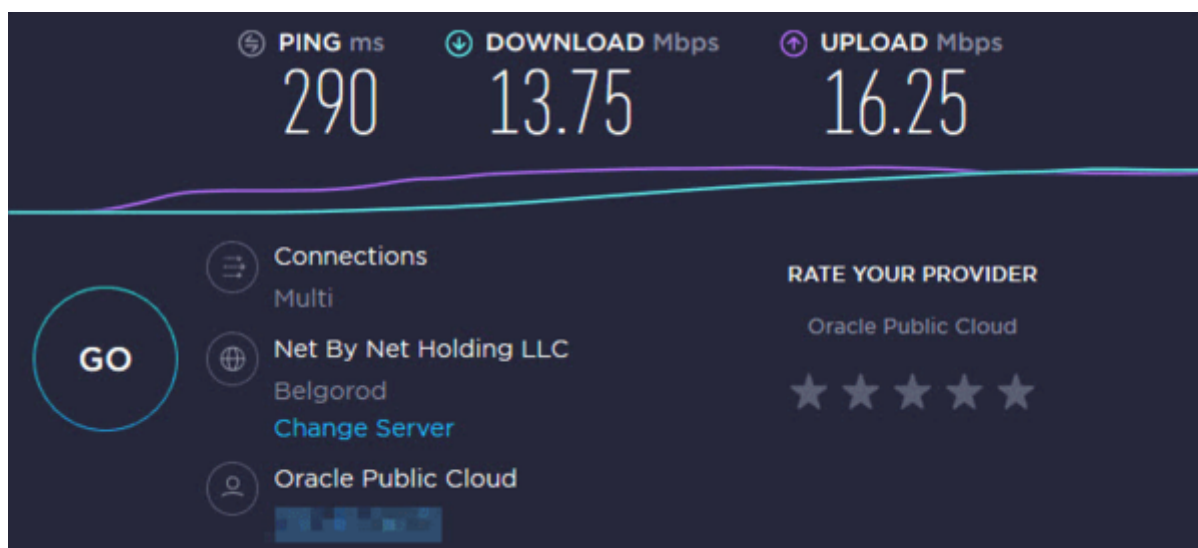
и выключить сжатие:

```
#comp-lzo
```

Теперь что касается производительности. OpenVPN через TCP имеет гораздо более высокие накладные расходы, особенно на плохих каналах. На хороших разница обычно невелика, и вы скорее упретесь в иные ограничения. Мы выполнили два замера для нашего сервера в Oracle Cloud, первый для UDP:



Второй для TCP:



Как видим, в нашем случае разница абсолютно незаметна и можно не особенно переживать за возможные потери пропускной способности. Но не следует забывать, что при тестировании использовались хорошие широкополосные каналы, в иных ситуациях, например, с мобильными клиентами, результаты могут существенно отличаться.

Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "Архитектура современных компьютерных сетей" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.