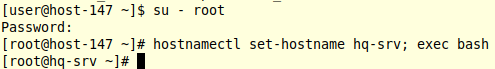
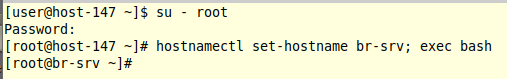
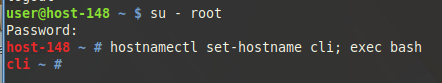
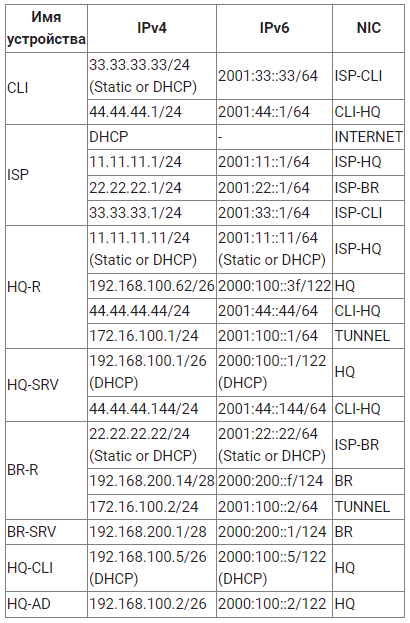
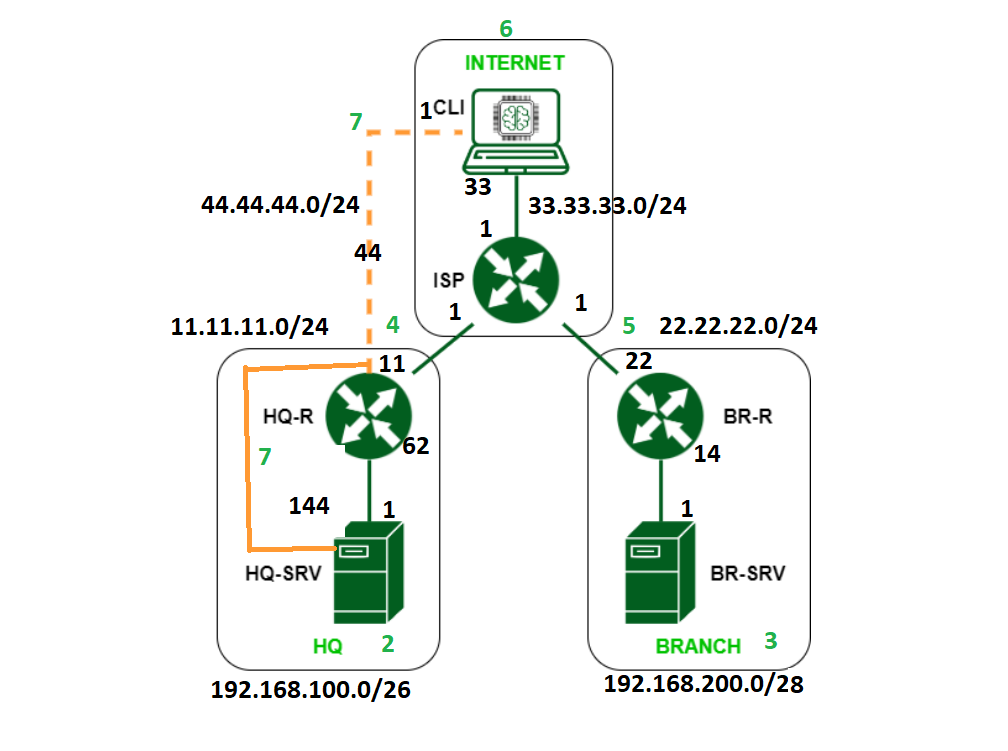
Задаем хостнейм, соответствующий топологии всех машин, либо редактируем файл /etc/hostname.









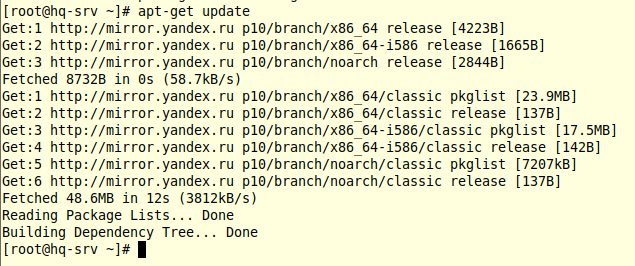


На клиенте в меню нужно нажать в правом верхнем углу на кнопку All applications.

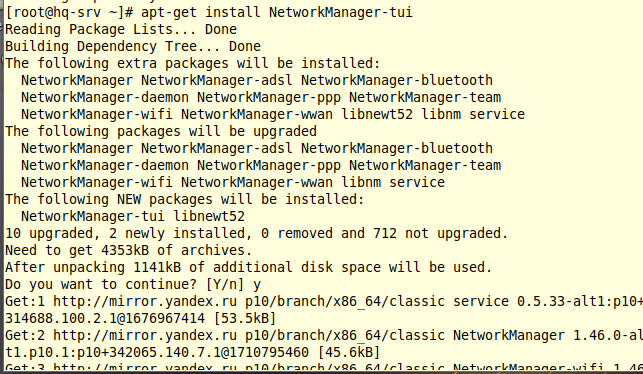
Выбираем первый интерфейс, нажимаем ***Advanced*** и меняем сетевую подсистему на NetworkManager (native), тем самым разблокировав первый интерфейс для редактирования, который по стандарту является системным.

*Если есть возможность, стоит в настройках виртуальных машин добавить дополнительный сетевой адаптер с NAT и использовать его пока не будет настроен NAT на роутере.*

На всех машинах возвращаемся в терминал и выполняем под root-ом apt-get update. Если нет то ping 8.8.8.8 → dhcpcd. После проводим настройку перед настройкой адресации необходимо проверить направление интерфейсов, для этого необходимо зайти в настройки VM.



Устанавливаем NetworkManager-tui. Если не работ то systemctl start Net..Man..



При нажатии на кнопку ***Advanced***, можно посмотреть mac-адрес интерфейса.

Прописываем nmtui → Edit a connection → выбираем нужный интерфейс → задаем ip-адрес согласно таблице и схеме.

Не забываем про IPv6, который прописывается по такому же принципу.

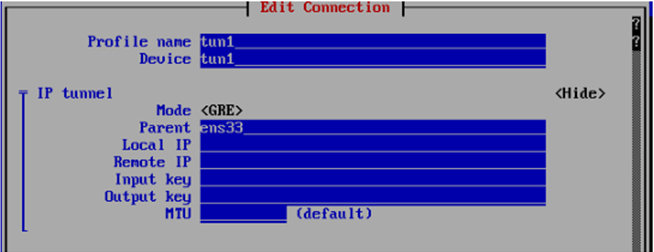
Выполним поднятие gre-туннеля через nmtui на маршрутизаторах:

***Nmtui***

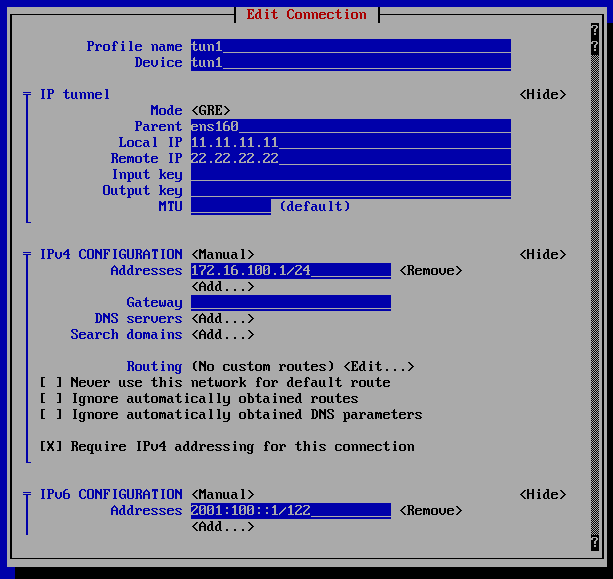
Выбираем ***Add***

Выбираем “***IP tunnel***”.

Задаём понятной "***Имя профиля***" → в "***Режим работы***" выбираем "***GRE***" → в "***Родительский***" указываем интерфейс **ens33** (в сторону ISP) → задаём "***Локальный IP***" hq — 110.110.110.110(br наоборот оба параметра) → задаём "***Удалённый IP***" 220.220.220.220→ переходим к "***конфигурации IPv4 и IPv6***".



Задаем адрес ***IPv4*** и ***IPv6*** для туннеля → нажимаем OK.



После чего необходимо **деактивировать** и **активировать** туннельный интерфейс средствами **nmtui**, чтобы изменения были применены

Для **корректной** работы протокола динамической маршрутизации требуется **увеличить** параметр **TTL** на интерфейсе туннеля:

***nmcli connection modify tun1 ip-tunnel.ttl 64***

#### **Настройка динамической (внутренней) маршрутизации средствами frr.**

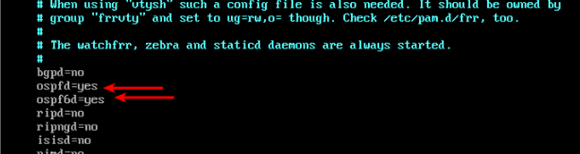
### HQ-R:

Установим пакет frr:

***apt-get update && apt-get install -y frr***

***vim /etc/frr/daemons редактирование шифт “с”***

* переводим **ospfd=no** в **ospfd=yes** - для OSPFv2 (IPv4)
* переводим **ospf6d=no** в **ospfd6=yes** - для OSPFv3 (IPv6)



сохранить все esc shift :wq

Включаем и добавляем в автозагрузку службу frr.

***systemctl enable --now frr***

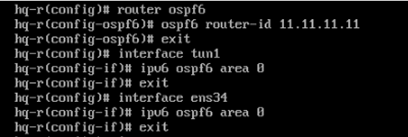
Проверяем:



Настраиваем OSPFv2 – переходим в интерфейс frr при помощи ***vtysh***.



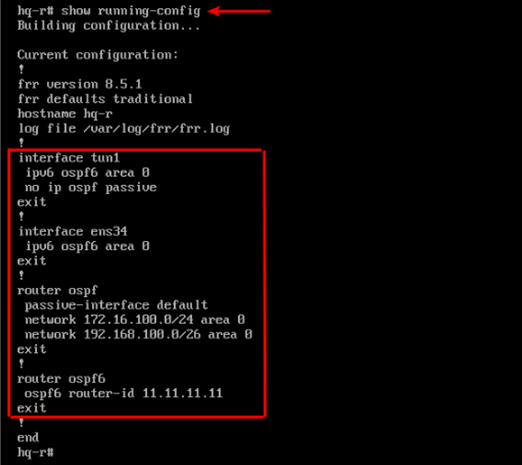
Настраиваем **OSPFv3**:



ospf6 router-id – назначить номер router-id

* добавляем сети интерфейсов tun1 и ens36 в конфигурацию OSPFv3

Проверяем:

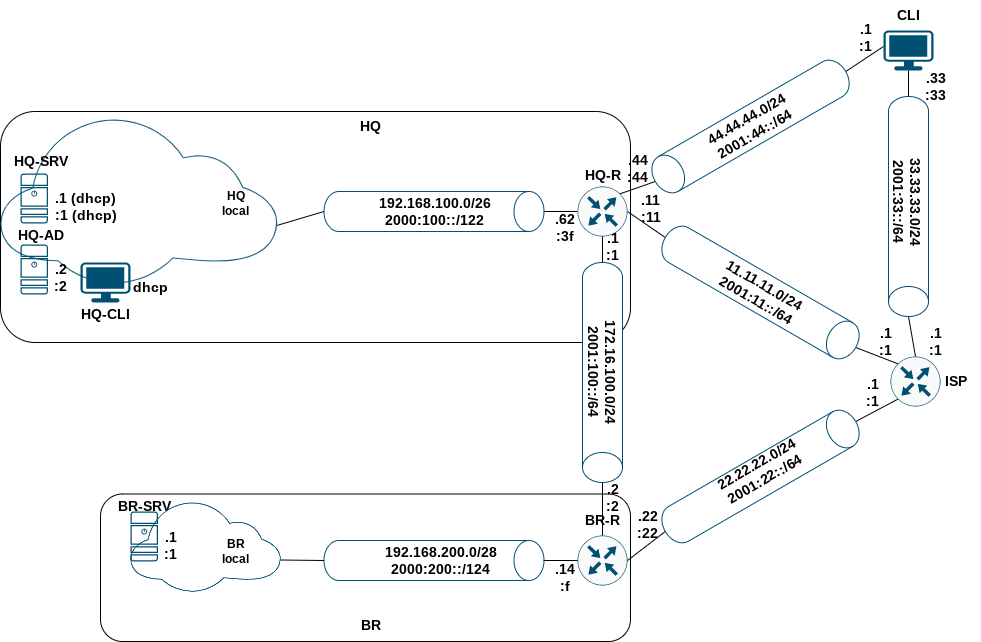


### BR-R:

Настройки аналогичны **HR-R**за исключением:

* объявляемых сетей в **OSPFv2;**
* **router-id**  в **OSPFv3**

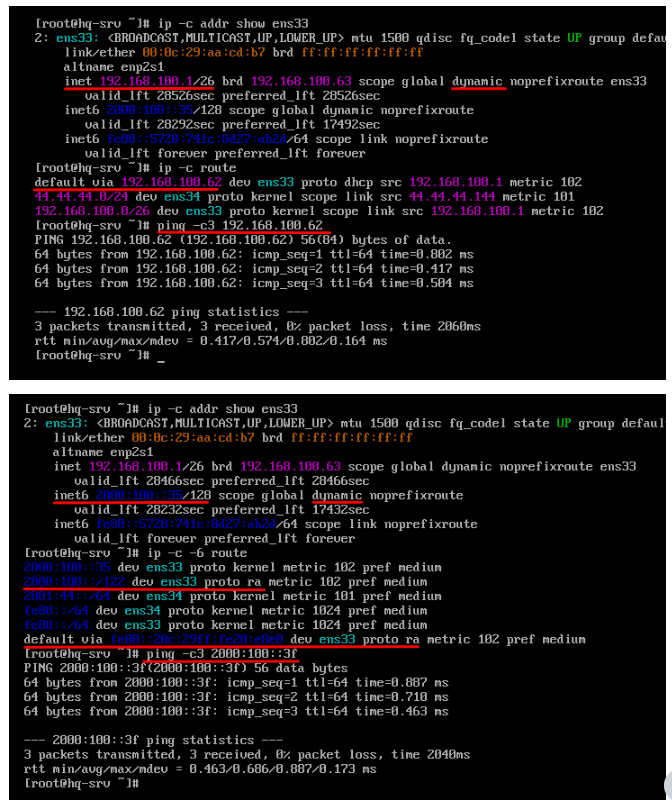
L3 app.diagrams.net



**фото в телефоне по тем делай**

### HQ-SRV:

При помщи **nmtui** необходимо перевести состояние "**IPv6 CONFIGURATION**" - из "**Automatic (DHCP-only)**" в состояние "**Automatic**", также после деактивировать и активировать данный интерфейс



Создание пользователя на клиенте и серверах

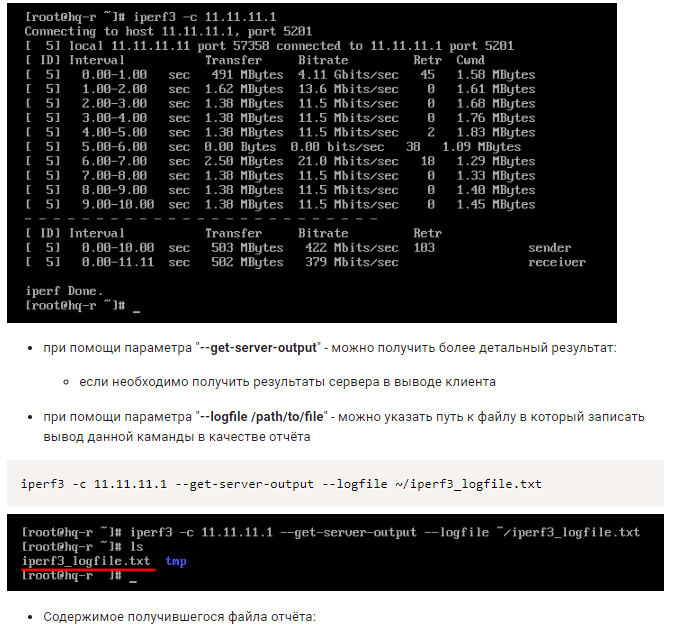
### Выполнение:

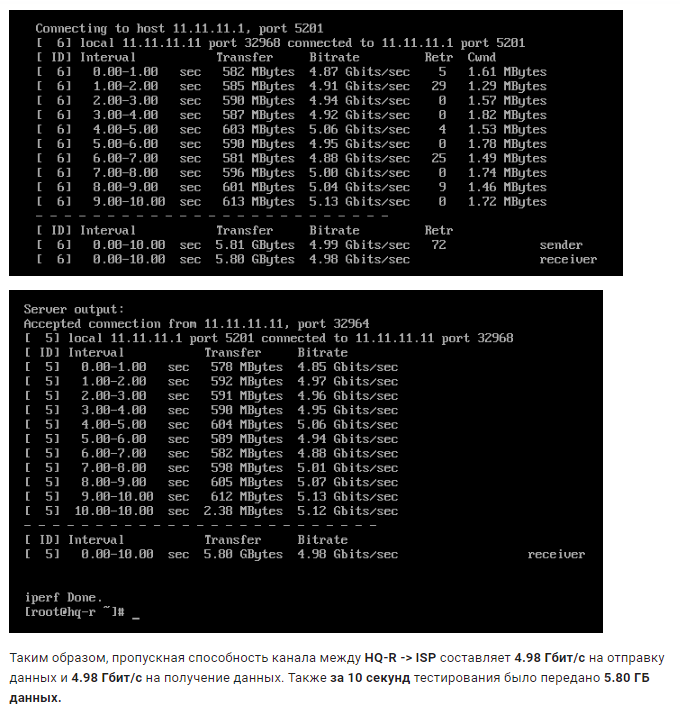
### HQ-R:

apt-get install -y iperf3

* Выполнить запуск systemctl enable --now iperf3
* запустим **iperf3** в режиме клиента, используя флаг **-c**, и укажем хост, на котором работает сервер -> **ISP** (11.11.11.1):

iperf3 -c 11.11.11.1

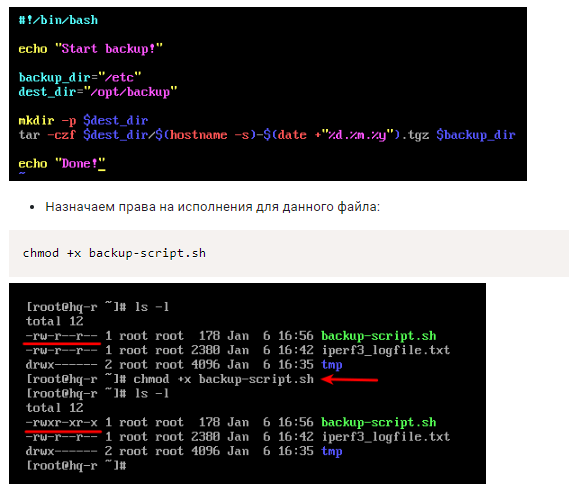




### HQ-R | BR-R:

bash-скрипт:

vim backup-script.sh



**#!/bin/bash**: Эта строка - это шебанг (shebang), она указывает на то, какую оболочку следует использовать для выполнения скрипта. В данном случае, скрипт выполняется с использованием оболочки Bash.

**echo "Start backup!"**: Эта команда выводит текст "Start backup!" в стандартный вывод (обычно в терминал).

**backup\_dir="/etc"**": Эта строка определяет переменную backup\_dir и устанавливает ей значение "/etc". Эта переменная будет использоваться для указания каталога, который будет архивироваться.

в случае если необходимо указать несколько директорий для резервного копирования они указываются в "кавычках" через пробел, например **backup\_dir="/etc /var /home"**

**dest\_dir="/opt/backup"**: Здесь определяется переменная dest\_dir и устанавливается значение "/opt/backup". Эта переменная будет использоваться для указания каталога, в который будет сохранен архив.

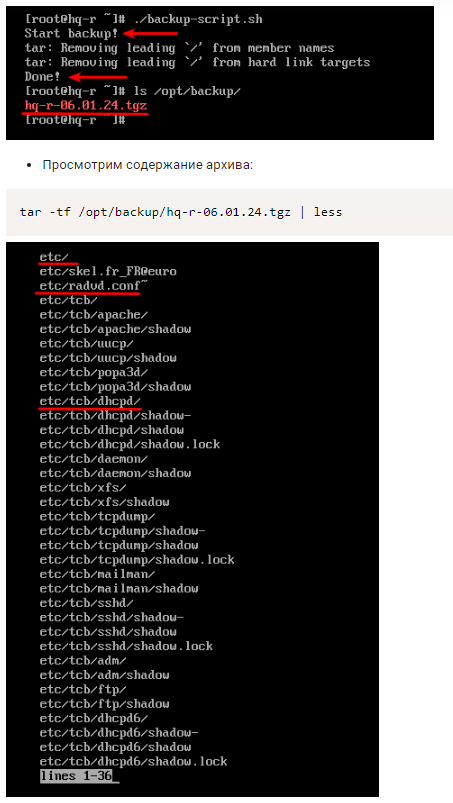
**mkdir -p $dest\_dir**: Эта команда создает каталог, указанный в переменной dest\_dir с опцией **-p**, которая позволяет создать каталог, а также все родительские каталоги, если они не существуют.

**tar -czf $dest\_dir/$(hostname -s)-$(date +"%d.%m.%y").tgz $backup\_dir**: Эта команда использует tar для создания архива файлов из каталога, указанного в переменной backup\_dir. Архив сохраняется в каталоге, указанном в переменной dest\_dir, с именем, которое включает имя хоста (hostname), текущую дату и расширение ".tgz". Опции -czf указывают на то, что архив должен быть сжат в формате gzip.

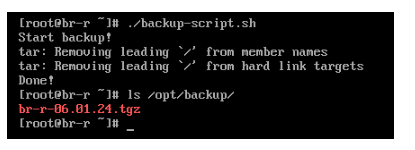
**echo "Done!"**: Эта команда выводит текст "Done!" в стандартный вывод после завершения создания архива.

Выполняем запуск

./backup-script.sh



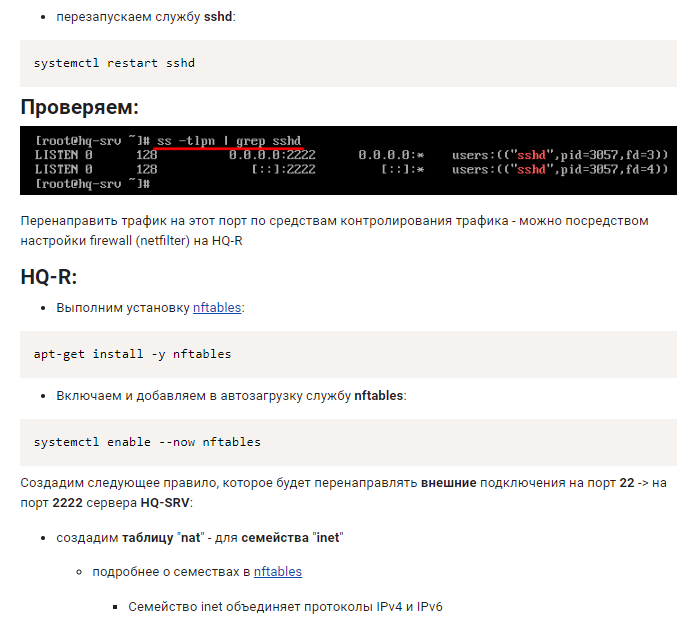
* таким образом, скрипт записал в архив всё содерджиимое директории **/etc**
* также есть возможность архивировать не одну, а несколько директорий с целью резевного копирования, необходимо всеголишь в переменную **backup\_dir** скрипта передать необходимые директории через пробел, например:
  + **backup\_dir="/etc /var /home"**
* Аналогичный скрипт запускаем и на BR-R
  + для ускорения процесса, передать скрипт с HQ-R на BR-R или наоборот можно по **scp**



На HQ-SRV в терминале под root-ом прописываем:

# vim /etc/openssh/sshd\_config

Нажимаем i, чтобы войти в режим редактирования. Заменяем #Port 22 на Port 2222, выходим из редактирования, нажав esc, прописываем :wq, чтобы сохранить изменения и выйти. 



nft add table inet nat

* добавим **цепочку** "**prerouting**" в **таблицу** "**nat**"
  + **PREROUTING** — предназначена для первичной обработки входящих пакетов, адресованных как непосредственно серверу, так и другим узлам сети. Сюда попадает абсолютно весь входящий трафик для дальнейшего анализа.

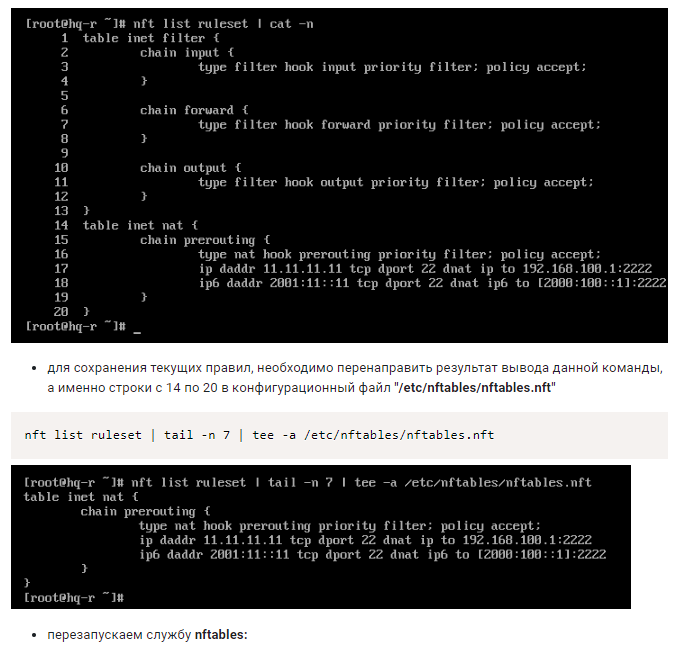
nft add chain inet nat prerouting '{ type nat hook prerouting priority 0; }'

* добавим необходимое **правило** для проброса портов в **цепочку** "**prerouting**" **таблицы** "**nat**" **семейства "inet":**

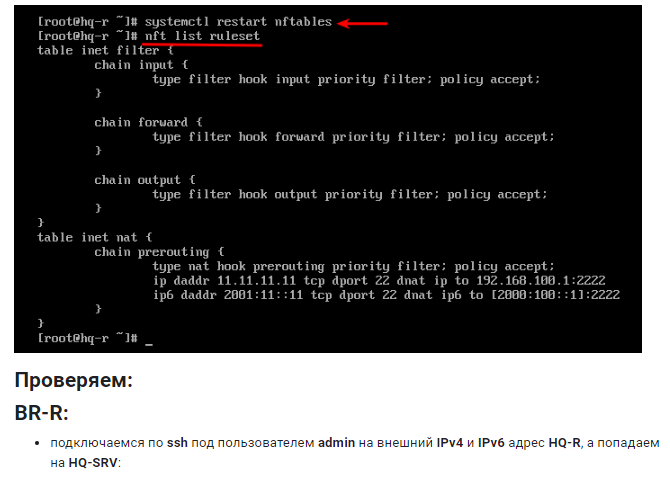
nft add rule inet nat prerouting ip daddr 11.11.11.11 tcp dport 22 dnat to 192.168.100.1:2222

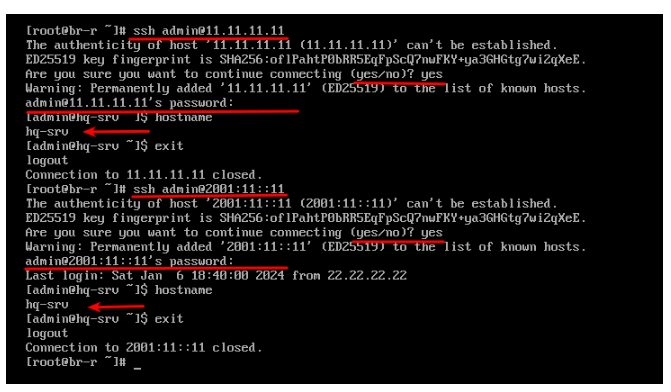
nft add rule inet nat prerouting ip6 daddr 2001:11::11 tcp dport 22 dnat to [2000:100::1]:2222

* просмотр текущих правил:



systemctl restart nftables





* Установим пакет **firewalld**:

apt-get install -y firewalld

* Включаем и добавляем в автозагрузку службу **firewalld:**

systemctl enable --now firewalld

* Добавляем внешний интерфейс"**ens33**" в зону "**public**":
  + параметр **--permanent** отвечает за сохранение правил при перезагрузке

firewall-cmd --permanent --zone=public --add-interface=ens33

* Добавляем внутренний интерфейс "**ens34**" в зону "**trusted**":

firewall-cmd --permanent --zone=trusted --add-interface=ens34

* Добавляем правило в зону "**public,**которое будет перенаправлять **внешние** подключения на порт **22** -> на порт **2222** сервера **HQ-SRV**:

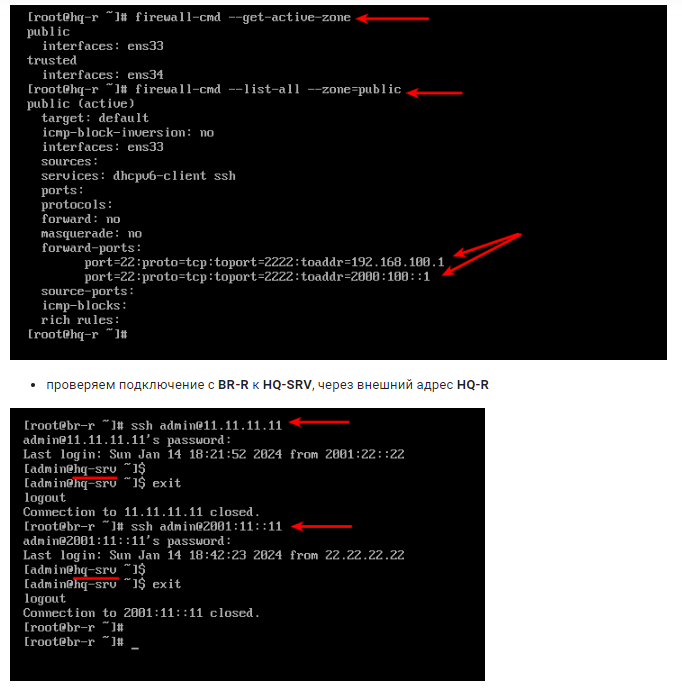
firewall-cmd --permanent --zone=public --add-forward-port=port=22:proto=tcp:toport=2222:toaddr=192.168.100.1

firewall-cmd --permanent --zone=public --add-forward-port=port=22:proto=tcp:toport=2222:toaddr=2000:100::1

* Перезагружаем firewall:

firewall-cmd --reload

* проверяем:



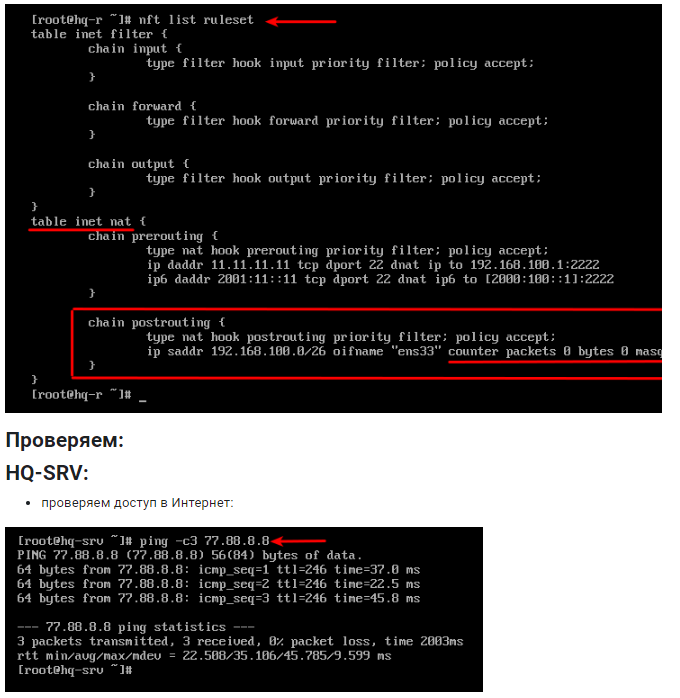
### HQ-R:

* Через ранее установленный **nftables** настраиваем доступ в Интернет (NAT), чтобы можно было устанавливать пакеты внутри локальной сети офиса **HQ:**
  + в ранее созданную **таблицу nat** в **семействе** **inet** добавляем **цепочку** **postrouting**:

nft add chain inet nat postrouting '{ type nat hook postrouting priority 0; }'

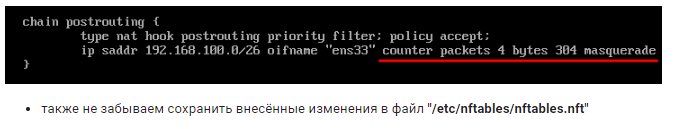
* в созданную **цепочку** **postrouting** добавляем **правило,** реализующее **nat** (**masquerade**):

nft add rule inet nat postrouting ip saddr 192.168.100.0/26 oifname 'ens33' counter masquerade



### HQ-R:

* также проверяем счётчик пакетов:



### HQ-SRV:

* обновляем список пакетов и устанавливаем **nftables**:

apt-get update && apt-get install -y nftables

* включаем и добавляем в автозагрузку **nftables**:

systemctl enable --now nftables

* добавляем правила, запрещающие доступ по **ssh (порт 2222)** с **CLI**, как с временного подключения так и с глобального, как для IPv4 так и для IPv6:
  + для **IPv4**:

nft add rule inet filter input ip saddr 33.33.33.33 tcp dport 2222 counter drop

nft add rule inet filter input ip saddr 44.44.44.0/24 tcp dport 2222 counter drop

* + для **IPv6**:

nft add rule inet filter input ip6 saddr 2001:33::/64 tcp dport 2222 counter drop

nft add rule inet filter input ip6 saddr 2001:44::/64 tcp dport 2222 counter drop

* + Проверяем:

