**Составление таблицы ip-адресов**

Вторая и третья сетевые карты ISP имеют ip-адреса 172.16.5.1/28 и 172.16.4.1/28 соответственно.

Отталкиваясь от этого, на сетевой карте BR-RTR, смотрящей в сторону ISP (ответный ip-адрес – на второй сетевой карте ISP) будет ip-адрес 172.16.5.2/28.

И отталкиваясь от этого же, на сетевой карте HQ-RTR, смотрящей в сторону ISP (ответный ip-адрес – на третьей сетевой карте ISP) будет ip-адрес 172.16.4.2/28.

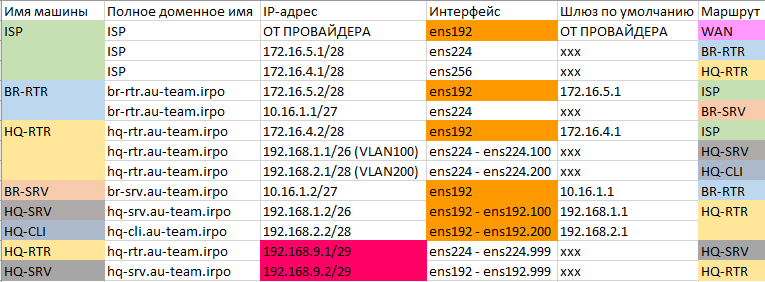
BR-подсеть будет в адресном пространстве 10.16.0.0

HQ-подсеть будет в адресном пространстве 192.168.0.0

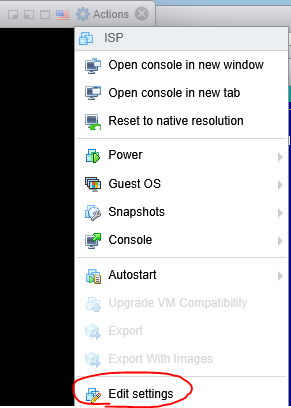
Путь **HQ-CLI** – 28 префикс маски т.к. допускается не более 16 адресов.

Путь **HQ-SRV** – 26 префикс маски т.к. допускается не более 64 адресов.

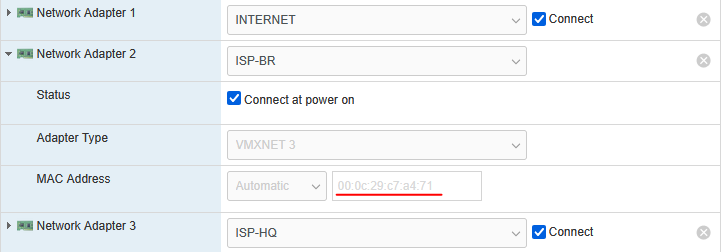
Путь **BR-SRV** – 27 префикс маски т.к. допускается не более 32 адресов.



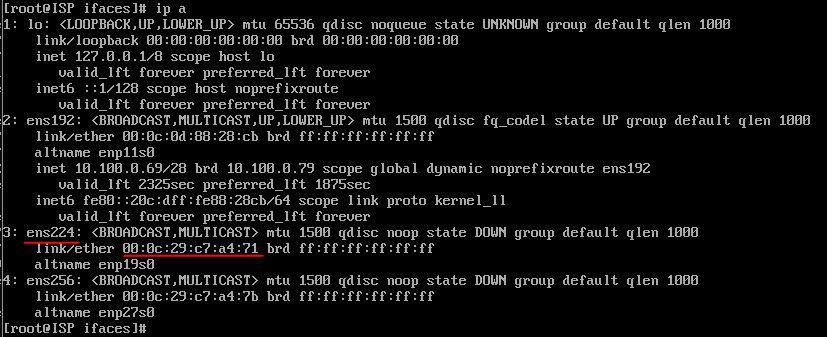
Так или иначе, нам понадобится сверить, какой интерфейс к какому адресу относится. Узнать это легко. Нажимаем на шестеренку над нашей машинкой. Там из выпадающих вариантов интересует нас **Edit Settings**.



Нажимаем на сетевую карту под конкретным интерфейсом и видим код в строчке **MAC Address**:



Вводим команду *ip a*:



Теперь мы знаем, что маршрут от **ISP** до **BR-RTR** относится к интерфейсу **ens224**.

**Первичная настройка ISP**

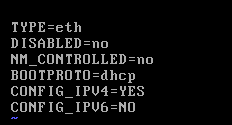
Для начала нужно ввести ***ip a*** – мы увидим имя каждого интерфейса. Сейчас нас интересует самый первый. В данном случае – **ens192**.

Создать папку интерфейса:

*mkdir /etc/net/ifaces/ens192*

Теперь нужно задать настройки интерфейса.

*vim /etc/net/ifaces/ens192/options*  
Нажимаем **i** и пишем:



Нажимаем **esc,** вбиваем ***:wq***, нажимаем **ENTER** и перезагружаем сеть:

*systemctl restart network*

Установим в систему **midnight commander**, чтобы жилось проще:

*apt-get update && apt-get install mc*

**Смена имени компьютера в сети**

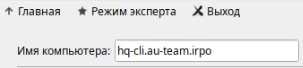
Имя каждого компьютера есть в нашей таблице в столбце **Полное доменное имя**. Заходим в следующую директорию:

*mc /etc*

Нас интересует командный файл **hostname**. Заходим в него, стираем имя и заместо него пишем новое. Сохраняем результат. Проделать это нужно **на каждой машине кроме клиента**.

Сохраняем командные файлы с помощью **f2** и выходим с помощью **f10**. Также следует запомнить, что редактирование командных файлов активируется кнопкой **f4**, а найти в папке нужную вам папку или файл можно быстро благодаря поиску, который активируется сочетанием клавиш **Ctrl+s**.

На клиенте нам нужно через меню зайти в центр управления. Далее в разделе Администрирования выбираем **Центр управления системой.** В разделе **Сеть** выбираем **Ethernet-интерфейсы**. Здесь мы меняем имя компьютера.



**Настройка статического IP-адреса и шлюза**

Статический адрес должен иметь каждый интерфейс на каждой машине кроме:

1) центрального интерфейса **ISP**, потому что там происходит соединение с провайдером;

2) **ens224** на **HQ-RTR**, потому что этот интерфейс – хост над двумя саб-интерфейсами;

3) **ens192** на **HQ-SRV** и на **HQ-CLI** – по той же причине.

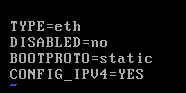
Посмотрим список интерфейсов:

*ip a*

Чтобы прописать статику, заходим в **mc** по пути  **/etc/net/ifaces/***[наш интерфейс, например,* ***ens192****]***/**

Набираем команды *touch options* и *touch ipv4address*, у нас появятся два командных файла.

В файле **options**нам нужно вписать следующее:



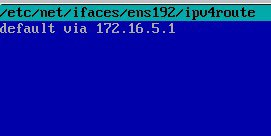
А в файле **ipv4address** мы вписываем ip-адрес с маской исходя из имеющейся у нас таблицы ip-адресов.

Важно отметить, что некоторые интерфейсы могут отсутствовать. В таких случаях с помощью команды **mkdir** (в mc на клавишу **f7**) и создаем директорию недостающего интерфейса. Каких директориев нам не хватает, смотрим по *ip a.*

Как альтернативный способ, можно прописать в терминале следующее:

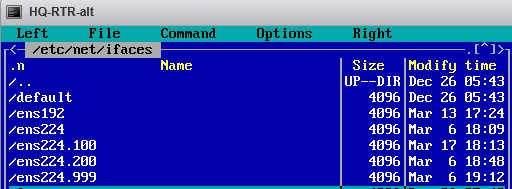
*mkdir /etc/net/ifaces/(наш интерфейс, например, ens192)/*

Теперь о том, как указать шлюз для передачи данных. По пути **/etc/net/ifaces/***[наш интерфейс, напр.* ***ens192****]***/** надо создать командный файл **ipv4route** и в нем укажем маршрут конкретной машины в соответствии с таблицей следующим образом:

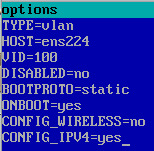


Сделать так надо на основных сетевых картах каждой машины кроме **ISP** (Среди интерфейсов **ens192**, **ens224** и **ens256** нас будет интересовать **ens192**).

Стоит отдельно коснуться сегмента сети **HQ** т.к. не всё там так очевидно. Вот, какие интерфейсы нам понадобятся на **HQ-RTR**:



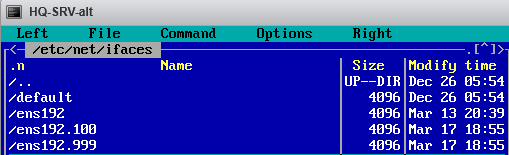
Если на **ens192** мы прописываем и статический адрес, и опции, и шлюз, то на **ens224** у нас только опции, потому что адреса будут в саб-интерфейсах **ens224.100**, **ens224.200** и **ens224.999**. В них, к слову, опции прописываются совсем иначе:



Обращаем внимание на строчку **VID** – число, которое в ней указано, должно соответствовать саб-интерфейсу. То есть, в **ens192.100** у нас будет ***VID=100***, и в других саб-интерфейсах мы следуем этой схеме. Перезагружаем службу сети:

*systemctl restart network*

Вот, как выглядит список интерфейсов на **HQ-SRV**:

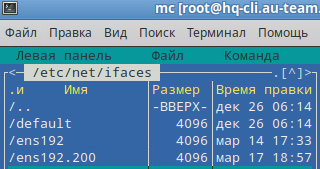


Тут запоминаем: на наших машинах в саб-интерфейсах, имеющих один VLAN, должны быть указаны адреса одной подсети. К примеру, если адрес на **ens224.100** - 192.168.1.1/26, то на **ens192.100** подойдет адрес 192.168.1.2/26.

Поскольку хостом для саб-интерфейсов здесь является **ens192**, то уже в нём будет присутствовать только файл опций, ведь адреса, как и в предыдущей машине, - в саб-интерфейсах, а шлюз – конкретно в **ens192.100**. И так же не забываем про файлы опций в них, но теперь учтем следующий момент: в строчке **HOST** на сей раз указать нужно **ens192**. И про строчку **VID** не забываем. Да и про перезагрузку службы тоже.

*systemctl restart network*

А вот, как какой список интерфейсов мы имеет на **HQ-CLI:**

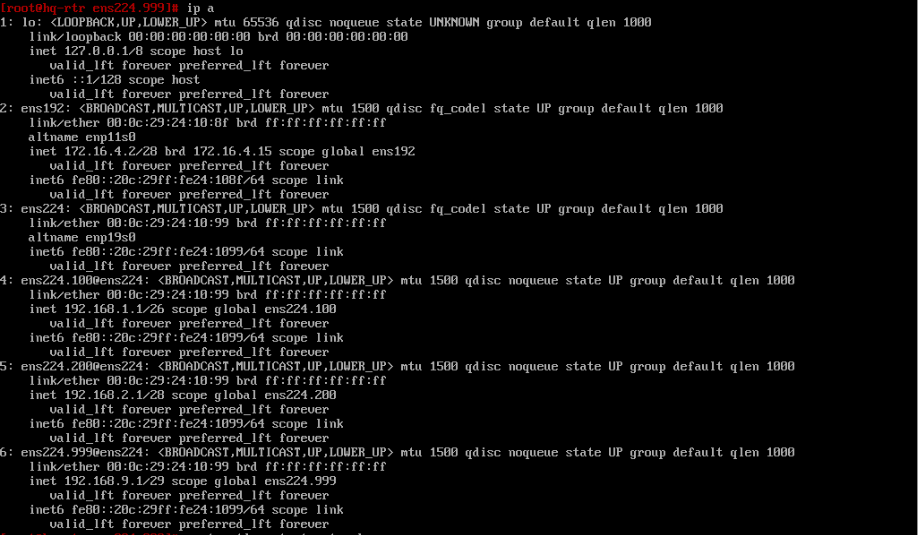


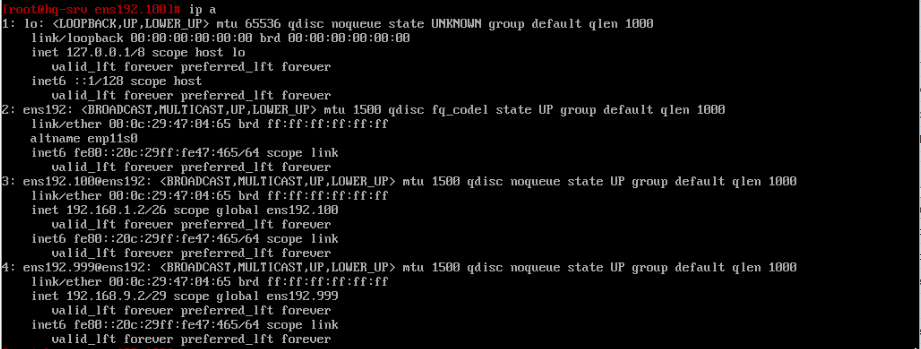
Как и ранее: в **ens192** – опции, в саб-интерфейсы – так же опции, но кроме них – адрес и шлюз. Строчки **HOST** и **VID** таким же образом подгоняем под наши условия. И опять же, перезагружаем службу, но предварительно отключаем службу сетевого менеджера.

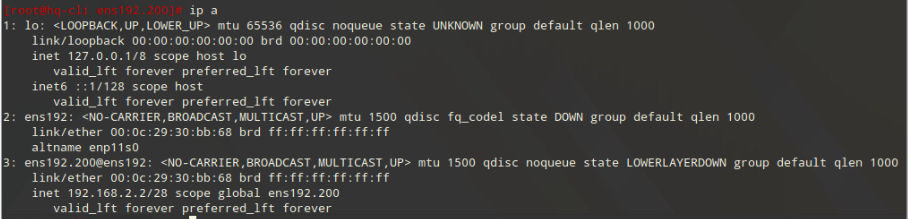
*systemctl disable --now NetworkManager*

*systemctl restart network*

Обязательно проверяем интерфейсы с помощью ***ip a****.*







Так это должно выглядеть. Все интерфейсы должна видеть система. Если что-то отсутствует, значит где-то была допущена ошибка.

**Настройка NAT**

Настраивать **NAT** нужно только на **ISP** и роутерах. Поскольку поначалу на роутерах нет доступа в интернет, начнем с **ISP**.

Устанавливаем **nftables**:

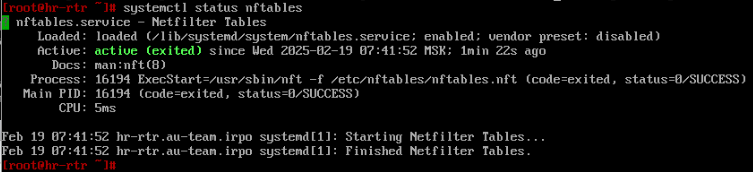
*apt-get update && apt-get install -y nftables*

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

*systemctl enable --now nftables*

Проверить работу **nftables**:

*systemctl status nftables*



Далее создаём необходимую структуру для **nftables** (семейство, таблица, цепочка) для настройки **NAT**. Cоздаём в семействе **IP** таблицу **NAT**:

*nft add table ip nat*

Создаём цепочку **postrouting** в таблице **NAT** семейства ip, также задаём **hook** и **priority**:

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

Создаём правила настройки **NAT** в семействе **IP**, таблице **NAT**, цепочке **postrouting**:

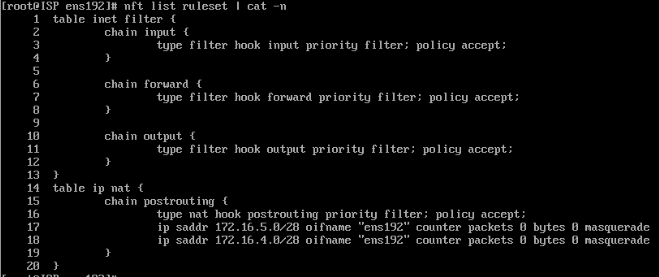
*nft add rule ip nat postrouting ip saddr  172.16.5.0/28 oifname "ens33" counter masquerade*

*nft add rule ip nat postrouting ip saddr 172.16.4.0/28 oifname "ens33" counter masquerade*

Очень важно учесть: в команде фигурирует **IP**-адрес со второстепенной сетевой карты. В данном случае нас интересуют **IP**-адреса с интерфейсов **ens224** и **ens265** (на каждый интерфейс – отдельная команда), на **HQ-RTR** – c **ens224.100** и **ens224.200**, на **BR-RTR** – c **ens224**. На месте подсети (четвертое число) при этом строго пишется ноль! На месте, где фигурирует сетевой интерфейс (между **oifname** и **counter**), мы пишем именно наш основной интерфейс, а в данном случае – **ens192**.

Следует сразу проверить нашу «цепочку правил»:

*nft list ruleset | cat -n*



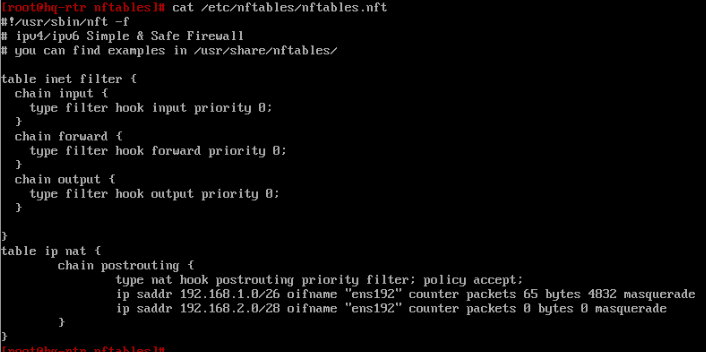
Если правила заданы некорректно, обновите службу **nftables** с помощью **systemctl** restart **nftables** и начните работу заново с добавления таблицы **NAT**.

Теперь сохраняем правила **nftables**. Так как в конфигурационном файле **/etc/nftables/nftables.nft** уже есть информация о таблице **filter**, необходимо дописать только что созданную информацию о таблице **NAT**: запишем в конфигурационный файл **/etc/nftables/nftables.nft** последние 7 строк (от 14 до 21) вывода команды **nft list ruleset**:

*nft list ruleset | tail -n7 | tee -a /etc/nftables/nftables.nft*

Проверки ради можно быстро глянуть внутренности файла:

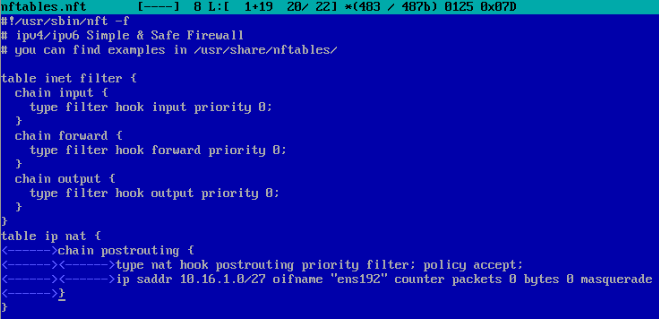
*cat /etc/nftables/nftables.nft*



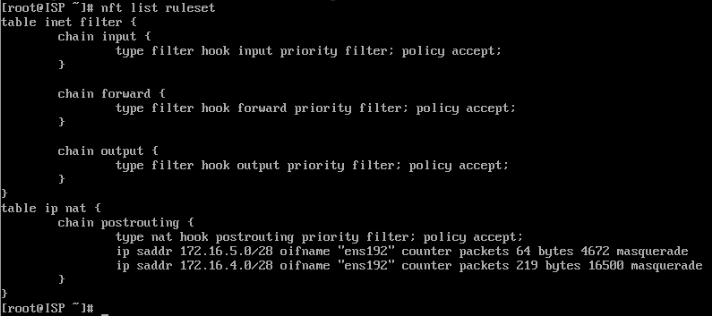
Перезагружаем службу nftables:

*systemctl restart nftables*

Здесь высока вероятность, что вам выдаст ошибку из-за некорректного кода в командном файле **/etc/nftables/nftables.nft**. Заходим в файл через **mc** и приводим его в порядок. Должно это выглядеть примерно так:



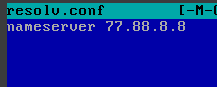
Теперь смотрим правила, чтобы убедиться в том, что **NAT** работает:



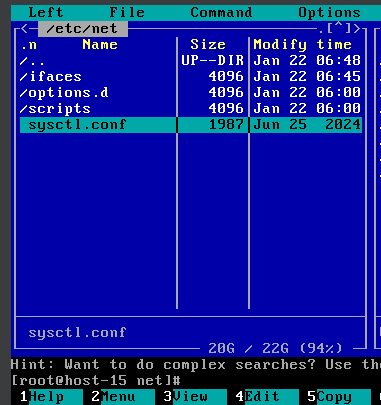
На роутерах **NAT** сейчас настроить мы не можем. Следуем инструкции дальше.

**Установка DNS-клиента и активация IP-Forward**

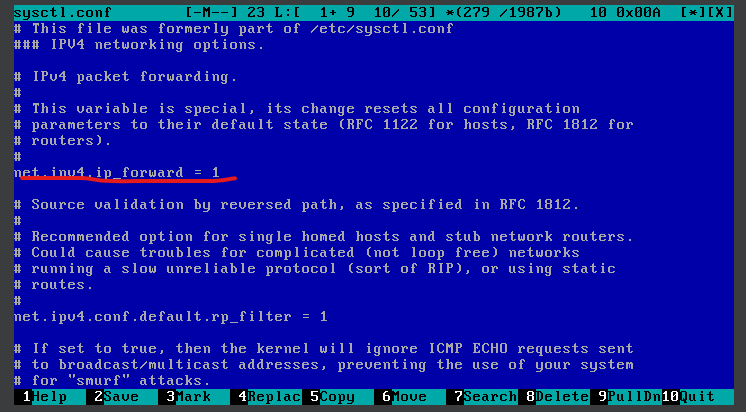
**DNS**-клиент требуется добавить на всех машинах кроме **ISP**. Через **mc** заходим в конфиг по пути **/etc/resolf.conf**. Указываем **DNS**-сервер **Яндекса** как изображено ниже.



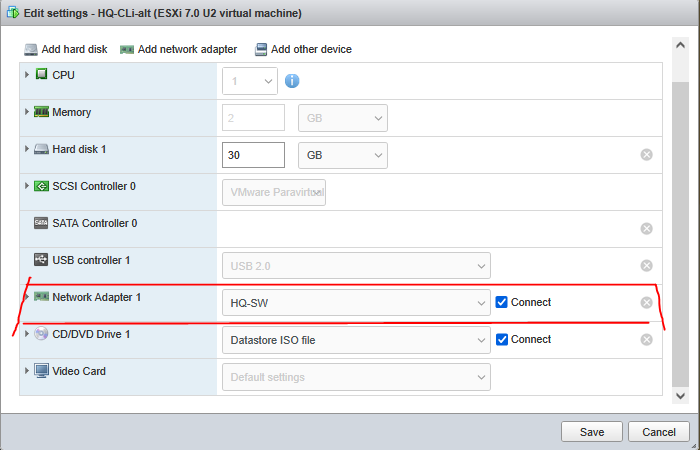
Активировать **IP-Forward** мы должны только на **ISP**, **HQ-RTR** и **BR-RTR**. Переходим по пути **/etc/net/sysctl.conf**.

****

В скриншоте ниже указано, как должна выглядеть настройка **IP-Forward**.

****

Проверяем, пингуется ли **DNS**-сервер на роутерах (завершить пингование можно сочетанием клавиш ***Ctrl****+****C***). Если всё работает, то возвращаемся к предыдущему этапу и настраиваем **NAT** теперь уже на роутерах по той же самой схеме. **DNS**-сервер должен пинговаться со всех машин. Если вы уверен, что сделали всё прописали идеально, но пинга не происходит, скорее всего, проблема – в отсутствии подключения к сетевой карте. Гляньте настройки машины.



**Создание пользователя на роутерах и серверах**

Для сначала добавим пользователя **sshuser** на наши сервера, то есть на **HQ-SRV** и **BR-SRV**:

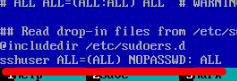
*adduser sshuser -u 1010*

Задаём пароль:

*passwd sshuser*

Мы должны будем дважды ввести пароль, который хотим поставить. Пароль в данном случае – *P@ssw0rd*.

Через **mc** заходим в конфигурационный файл **/etc/sudoers** и в нём нам нужно добавить следующую строчку:



Затем мы должны добавить пользователя в группу **wheel**:

*usermod -aG wheel sshuser*

Теперь проверяем, может ли наш пользователь заходить в **sudo** без доп. аутентификации. Для начала выходим из пользователя **root**:

*logout*

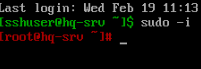
Теперь входим в новоиспеченного пользователя:

*sshuser*

*P@ssw0rd*

*sudo -i*

Лицезреть в итоге мы должны это:



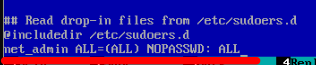
Не забываем: на обоих серверах! Теперь занимаемся пользователем на роутерах **HQ-RTR** и **BR-RTR**:

*adduser net\_admin -u 1010*

*passwd net\_admin*

*P@$$word* (x2)

*mc /etc/sudoers*



*usermod -aG wheel net\_admin*

*logout*

*net\_admin*

*P@$$word*

*sudo -i*

Теперь смотрим и убеждаемся в правильности наших действий:

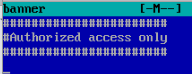


В случае, если вы создали определенного пользователя не на той машине, можно легко пользователя удалить:

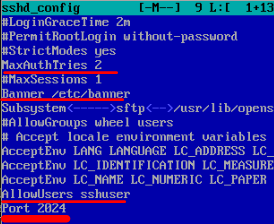
*userdel [имя пользователя]*

**Настройка безопасного удаленного доступа (ssh) на серверах**

Работа с удаленным доступом ведется НА СЕРВЕРАХ. Для начала заходим в **mc** и с помощью команды ***touch***создаём конфигурационный файл **/etc/banner**, в котором печатаем следующую картину:



Теперь нам пора зайти в файл **/etc/openssh/sshd\_config**. Здесь добавляем следующие строчки:



Как всегда, не забываем прожимать ***f2***, чтобы сохранить. Перезапускаем службу **SSHd**:

*systemctl restart sshd*

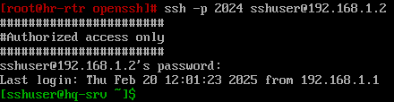
Идём на соответствующий нашему серверу роутер. Там, чтобы войти на сервер через удаленный доступ, вводим следующее:

*ssh -p 2024 sshuser@192.168.1.2*

Это на **HQ-RTR**. На **BR-RTR** будет:

*ssh -p 2024 sshuser@10.16.1.2*

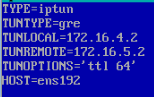
Стоит пояснить, что после «@» пишется адрес сервера, удаленный доступ на котором мы настраиваем. Нас спросят, хотим ли мы продолжить подключение. Пишем *yes* и вводим пароль, который мы ранее задали для пользователя **sshuser**. Если вы всё сделали правильно, выглядеть это будет так:



И про удаленный доступ к другому серверу не забываем!

**Конфигурация ip-туннеля между роутерами**

Все приведенные операции нужно будет осуществить на роутерах. Уже с самого начала мы заходим в **mc**. Конкретно нас интересует директория **/etc/net/ifaces**. Здесь мы нажимаем ***f7***, чтобы создать папку **tun1**. Внутри этой папки (с помощью ***touch***) создаем конфигурационный файл - **options**. Из директории пока никуда не уходим, ведь нужно сразу же создать еще один файл – **ipv4address**. А вот уже теперь с помощью клавиши **f4** заглядываем внутрь **options** и пишем следующие строчки:



Обращаем внимание: в строчке **TUNLOCAL** нужно указать ip-адрес той машины, на которой вы находитесь в данный момент. В строчке **TUNREMOTE** же мы указываем ip-адрес соседа этой машины. Если мы на **HR-RTR**, то соседским адресом будет адрес **BR-RTR**. На **BR-RTR** в файле **options**, как нетрудно догадаться, будет то же самое, но только строчки **TUNLOCAL** и **TUNREMOTE** поменяются айпишниками.

Сохраняем файл клавишей **f2** и выходим клавишей **f10**.

Дальше всё очень просто – идём в файл **ipv4address**, расположенный всё в той же папке **tun1** и пишем здесь айпишник начала (или конца) туннеля. Айпишник для туннеля легко можно придумать. Допустим, так:



(*HQ-RTR*)



(*BR-RTR*)

Сохраняем и выходим из **mc**. Как обычно, перезагружаем систему **Network**:

*systemctl restart network*

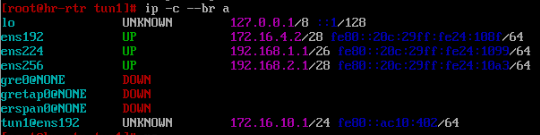
Теперь мы должны включить модуль ядра для **gre**:

*modprobe gre*

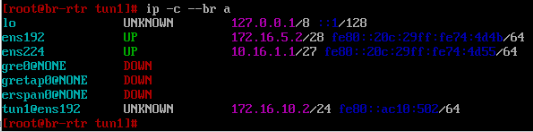
Проверяем созданный нами конец туннеля:

*ip -c --br a*

Как мы видим, конец туннеля – на месте:



(*HQ-RTR*)



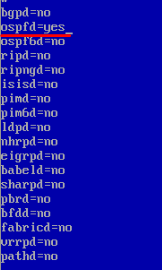
(*BR-RTR*)

**Настройка динамической маршрутизации между офисами**

По-прежнему работаем на роутерах **HQ** и **BR**. Перво-наперво нам точно нужно установить **frr** – пакеты для гибкой настройки маршрутизации:

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

Посредством **mc** заходим в файл **etc/frr/daemons** и активируем службу **ospfd**. Строчка службы должна выглядеть так:



**Frr** нужно включить в автозагрузку. Всё еще помним, что команды вводим на обоих роутерах.

*systemctl enable frr*

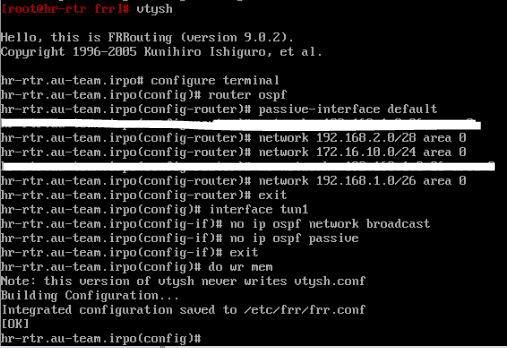
Теперь непосредственно включим службу:

*systemctl enable --now frr*

Стоит проверить, включена ли служба, с помощью специальной команды:



Дальше работаем с **frr**. Точно повторяем команды со скриншота по порядку. Работа с **frr** на скриншоте ниже касается **HQ-RTR**. В команде ***network*** *(ip)* ***area 0*** вводим **IP**-подсеть нашего туннеля и **IP**-подсети с интерфейсов, на которых у нас адреса в сторону **HQ-SRV** и **HQ-CLI**.



Теперь, чтобы выйти из терминала в **frr**, вводим:

*exit*

C **BR-RTR** принцип такой же - айпи-подсеть туннеля и айпи-подсеть сервера. Клиента в **BR**-сегменте у нас банально нет.

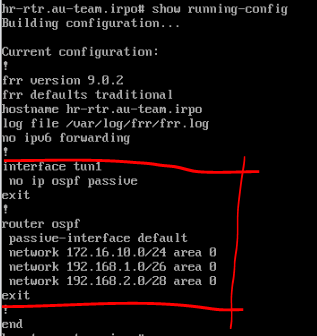
,

*exit*

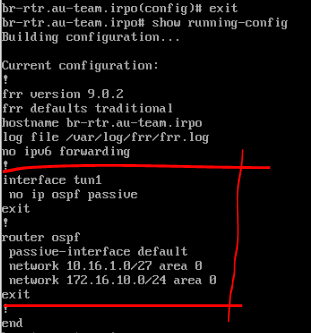
Для проверки, правильно ли всё было нами введено:

*show running-config*

Вот так это должно выглядеть на **HQ-RTR**:



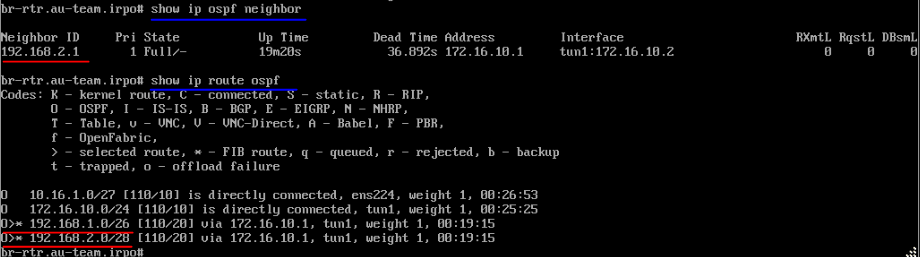
Вот так это должно выглядеть на **BR-RTR**:



Проверять, имеется ли маршрутизация между роутерами, мы будем другими командами:



*(HQ-RTR)*



*(BR-RTR)*

Сейчас мы должны заняться установкой парольной защиты для **frr**. Если вы сейчас не находитесь во **vtysh**, следует в него зайти.

*configure terminal*

*interface tun1*

Активируем аутентификацию и задаём пароль:

*ip ospf authentication*

*ip ospf authentication-key PLAINPAS*

Выходим из конфигурации интерфейса, сохраняем изменение в настройках, а затем выходим и из терминала:

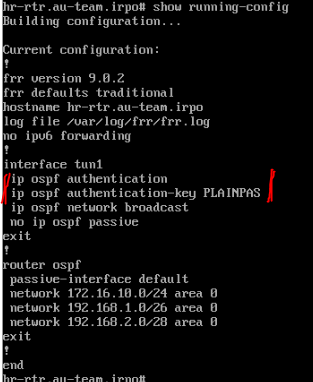
*exit*

*do wr mem*

*exit*

Как только защита протокола будет стоять на обоих роутерах, мы должны будем проверить наличие защиты:

*show running-config*



Важно: **пароль на роутерах должен стоять строго одинаковый**.

**Настройка протокола динамической конфигурации хостов (DHCP)**

Нашим **DHCP**-сервером будет выступать **HQ-RTR**. Перво-наперво установим пакеты для **DHCP**-сервера:

*apt-get update && apt-get install dhcp-server*

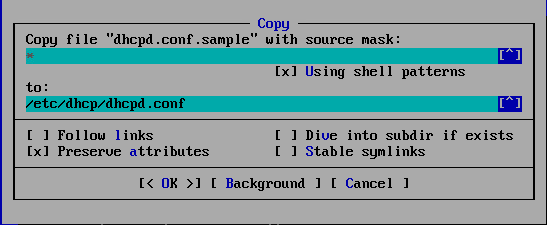
Включим автоматический запуск протокола при запуске системы:

*chkconfig dhcpd on*

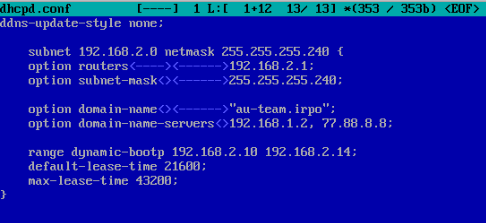
Собственно, запускаем протокол:

*service dhcpd start*

Теперь въезжаем в **mc** и по пути **/etc/dhcp/**. Нам нужно создать файл ***dhcpd.conf***. Но мы не просто создаем его. Мы наводимся на файл ***dchpd.conf.sample*** и нажимаем на ***f5***, чтобы открыть окно с опциями копирования файла. Указываем в строчке под ***to*** путь еще несуществующего файла и подтверждаем действие, нажав кнопку ***ОК***.



После этого заходим в новенький файл и редактируем его:



Здесь очень важно учесть следующие моменты:

-после **subnet** нам нужно указать подсеть **HQ-CLI**, а после **netmask** – его маску в полном формате;

-в строчке **option routers** мы указываем адрес **HQ-RTR**, смотрящий в сторону **HQ-CLI**.

-в строчке **subnet-mask** указываем такую же маску;

-**DNS**-суффикс по заданию – **au-team.irpo**, указываем в кавычках.

-в **option domain-name-servers** указать требуется адрес с центрального интерфейса **HQ-SRV**. А рядом – **DNS**-сервер Яндекса (77.88.8.8);

-в строчке **range dynamic-bootp** мы указываем диапазон адресов в подсети **HQ-CLI**. Диапазон лучше сделать с 10 до 14;

-смотрим ВНИМАТЕЛЬНО, чтобы после каждая строчка кроме самой первой кончалась точкой c запятой, иначе **dhcp** не заработает;

-лучше убрать из файла все лишние строчки.

Перезапускаем службу:

*systemctl restart dhcpd*

Теперь проверяем статус службы специальной командой:

*systemctl status dhcp d*

Смотрим и радуемся:

