

- Создайте пользователя `sshuser` на серверах `HQ-SRV` и `BR-SRV`
 - Пароль пользователя `sshuser` с паролем `P@ssw0rd`
 - Идентификатор пользователя `1010`
 - Пользователь `sshuser` должен иметь возможность запускать `sudo` без дополнительной аутентификации.
- Создайте пользователя `net_admin` на маршрутизаторах `HQ-RTR` и `BR-RTR`
 - Пароль пользователя `net_admin` с паролем `P@Sword`

- При настройке на EcoRouter пользователь `net_admin` должен обладать максимальными привилегиями
 - При настройке ОС на базе Linux, запускать `sudo` без дополнительной аутентификации
4. Настройте на интерфейсе HQ-RTR в сторону офиса HQ виртуальный коммутатор:
 - Сервер HQ-SRV должен находиться в ID VLAN 100
 - Клиент HQ-CLI в ID VLAN 200
 - Создайте подсеть управления с ID VLAN 999
 - Основные сведения о настройке коммутатора и выбора реализации разделения на VLAN занесите в отчёт
 5. Настройка безопасного удаленного доступа на серверах HQ-SRV и BR-SRV:
 - Для подключения используйте порт 2024
 - Разрешите подключения только пользователю `sshuser`
 - Ограничьте количество попыток входа до двух
 - Настройте баннер «Authorized access only»
 6. Между офисами HQ и BR необходимо сконфигурировать `ip` туннель
 - Сведения о туннеле занесите в отчёт
 - На выбор технологии GRE или IP in IP
 7. Обеспечьте динамическую маршрутизацию: ресурсы одного офиса должны быть доступны из другого офиса. Для обеспечения динамической маршрутизации используйте `link state` протокол на ваше усмотрение.
 - Разрешите выбранный протокол только на интерфейсах в `ip` туннеле
 - Маршрутизаторы должны делиться маршрутами только друг с другом
 - Обеспечьте защиту выбранного протокола посредством парольной защиты
 - Сведения о настройке и защите протокола занесите в отчёт
 8. Настройка динамической трансляции адресов.
 - Настройте динамическую трансляцию адресов для обоих офисов.
 - Все устройства в офисах должны иметь доступ к сети Интернет
 9. Настройка протокола динамической конфигурации хостов.
 - Настройте нужную подсеть
 - Для офиса HQ в качестве сервера DHCP выступает маршрутизатор HQ-RTR.
 - Клиентом является машина HQ-CLI.
 - Исключите из выдачи адрес маршрутизатора
 - Адрес шлюза по умолчанию – адрес маршрутизатора HQ-RTR.
 - Адрес DNS-сервера для машины HQ-CLI – адрес сервера HQ-SRV.
 - DNS-суффикс для офисов HQ – `au-team.irpo`
 - Сведения о настройке протокола занесите в отчёт
 10. Настройка DNS для офисов HQ и BR.
 - Основной DNS-сервер реализован на HQ-SRV.
 - Сервер должен обеспечивать разрешение имён в сетевые адреса устройств и обратно в соответствии с таблицей 2
 - В качестве DNS сервера пересылки используйте любой общедоступный DNS сервер
 11. Настройте часовой пояс на всех устройствах, согласно месту проведения экзамена.

Решение:

1. ПРОИЗВЕДИТЕ БАЗОВУЮ НАСТРОЙКУ УСТРОЙСТВ

И

4. НАСТРОЙКА НА ИНТЕРФЕЙСЕ HQ-RTR В СТОРОНУ ОФИСА HQ ВИРТУАЛЬНОГО КОММУТАТОРА

Настроим имена на всех устройствах полные доменные имена FQDN. Для этого используем команду `hostnamectl set-hostname FQDN-имя` из следующей таблицы

Устройство	FQDN устройства (полное доменное имя)
HQ-RTR	hq-rtr.ks54.net
BR-RTR	br-rtr.ks54.net
HQ-SRV	hq-srv.ks54.net
HQ-CLI	hq-cli.ks54.net
BR-SRV	br-srv.ks54.net

Настроим машину на примере HQ-RTR, аналогично проделываем на BR-RTR, HQ-SRV, BR-SRV. На машине HQ-CLI выполним настройку через графику.

HQ-RTR

Войдем в систему.

Логин: **root**

Пароль: **toor** (при вводе он не отображается и символы не видны)

```
Welcome to ALT Server 10.2 (Mendeleevium)!\n\nHostname: ta.ju3pcuq8uxp\nIP: 127.0.0.2\nta.ju3pcuq8uxp login: root\nPassword:\nLast login: Sat Nov 30 11:21:32 MSK 2024 on tty1\n[root@ta.ju3pcuq8uxp ~]#
```

Меняем имя устройства и обновляем вход в bash:

```
[root@ta.ju3pcuq8uxp ~]# hostnamectl set-hostname hq-rtr.ks54.net\n[root@ta.ju3pcuq8uxp ~]# exec bash\n[root@hq-rtr ~]#
```

Как видим имя изменилось в строке приветствия системы после команды `exec bash`

Командой `hostname -f`

```
[root@br-rtr ~]# hostname -f\nbr-rtr.ks54.net\n[root@br-rtr ~]#
```

Проделываем то же самое и на остальных устройствах кроме HQ-CLI:

BR-RTR

```
[root@grzhwcs82ux4u ~]# hostnamectl set-hostname br-rtr.ks54.net\n[root@grzhwcs82ux4u ~]# exec bash\n[root@br-rtr ~]#\n\n[root@br-rtr ~]# hostname -f\nbr-rtr.ks54.net
```

HQ-SRV

```
[root@kntoqd5amxz1p ~]# hostnamectl set-hostname hq-srv.ks54.net\n[root@kntoqd5amxz1p ~]# exec bash\n[root@hq-srv ~]# hostname -f\nhq-srv.ks54.net\n[root@hq-srv ~]#
```

BR-SRV

```

[root@opnighrekygpk ~]# hostnamectl set-hostname br-srv.ks54.net
[root@opnighrekygpk ~]# exec bash
[root@br-srv ~]# hostname -f
br-srv.ks54.net
[root@br-srv ~]#

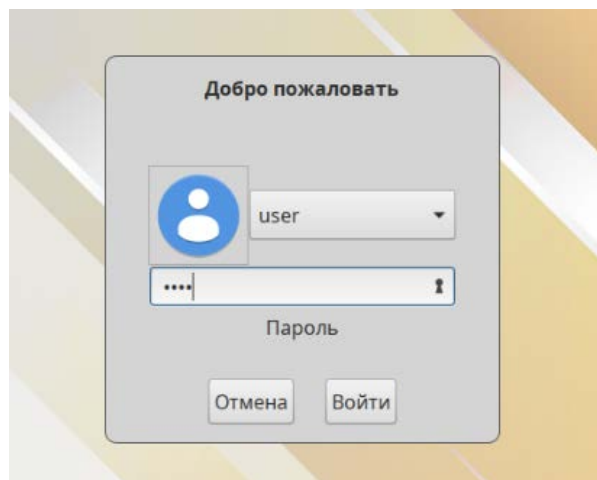
```

Настроим имя на машине **HQ-CLI**

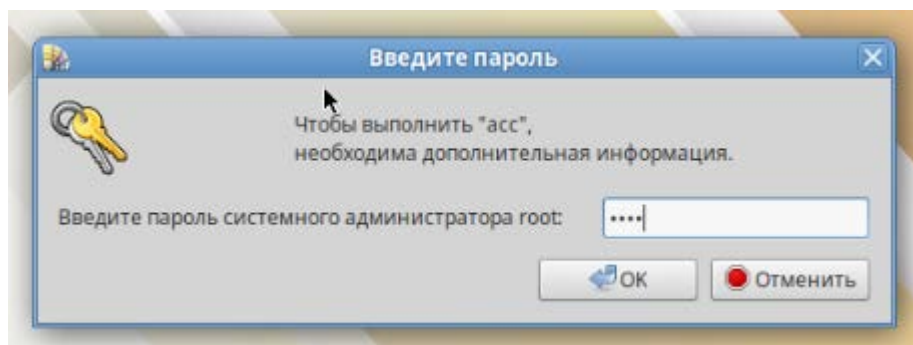
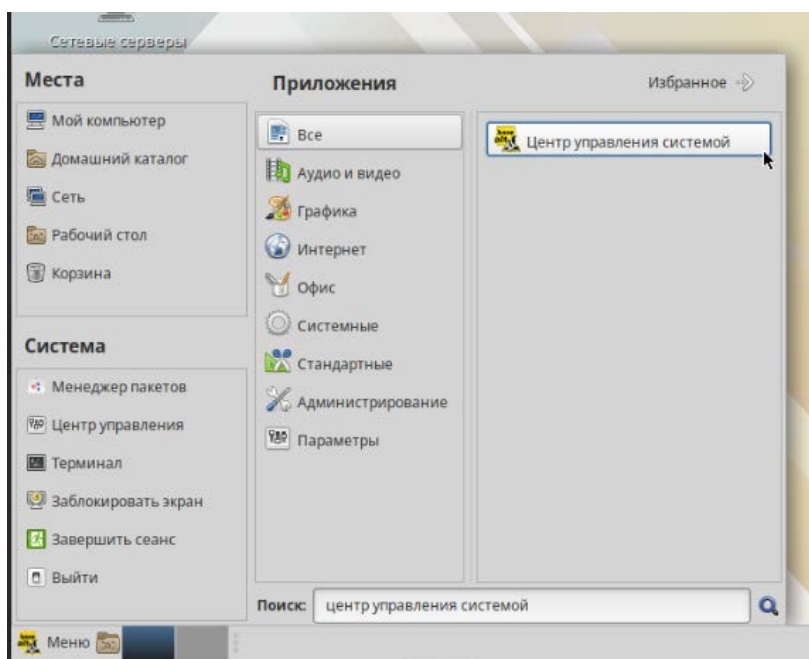
Войдем в графическую оболочку системы со следующими параметрами:

Логин: **user**

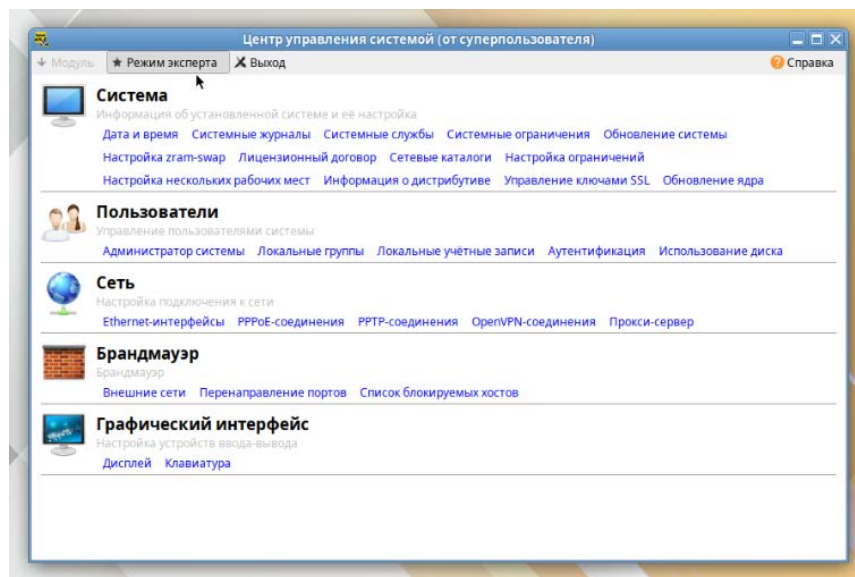
Пароль: **resu**



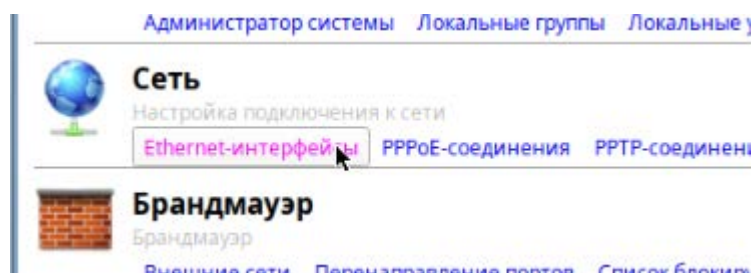
Далее находим в стартовом меню («пуск») программу «Центр управления системой», при запуске потребуется ввести пароль от суперпользователя: *toor*



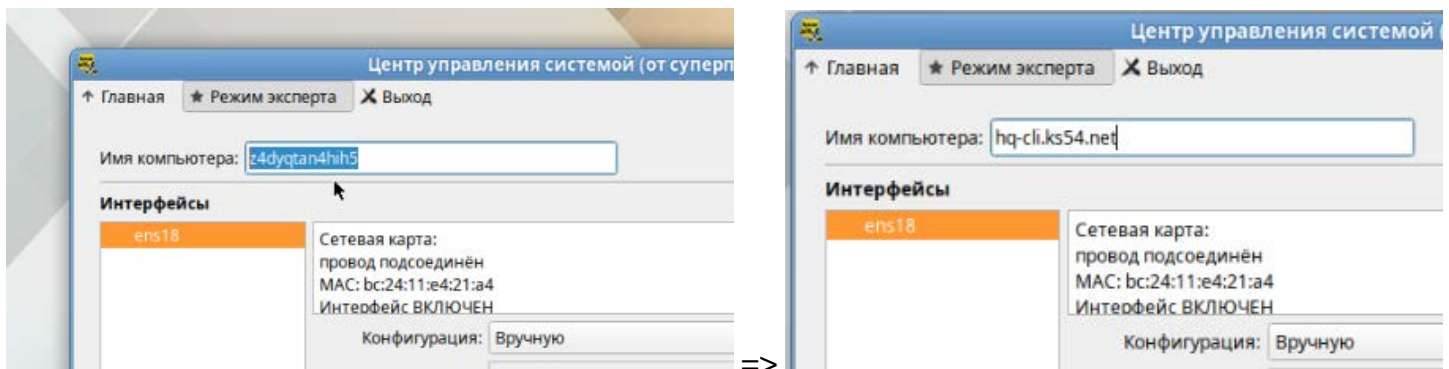
Получаем такое окно и нажимаем на режим эксперта



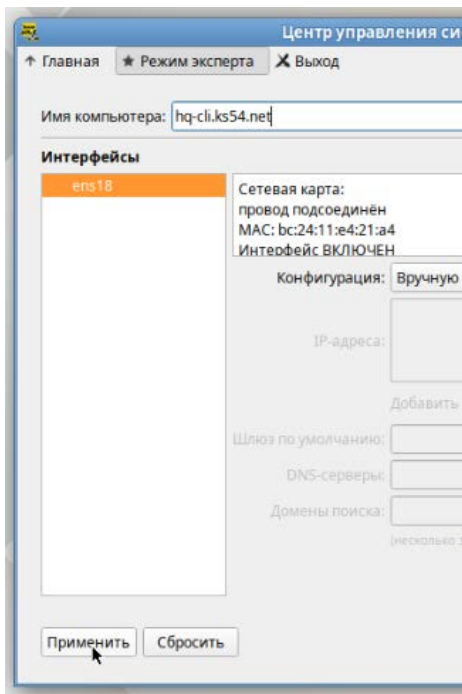
Выбираем Ethernet-интерфейсы



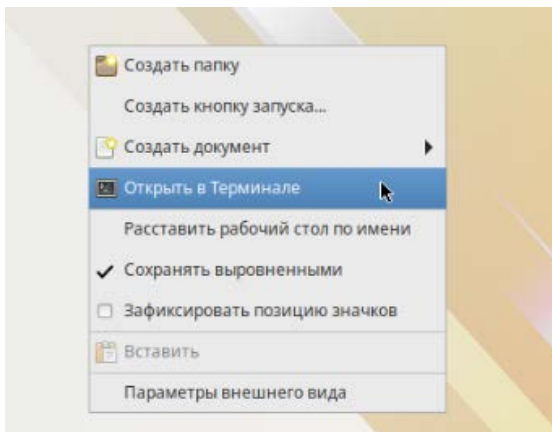
И меняем в появившемся окне имя машины на hq-cli.ks54.net



И нажимаем применить



Выходим из оснастки и запускаем эмулятор терминала, чтобы проверить присвоенное имя. Правой кнопкой мыши и открыть в терминале.



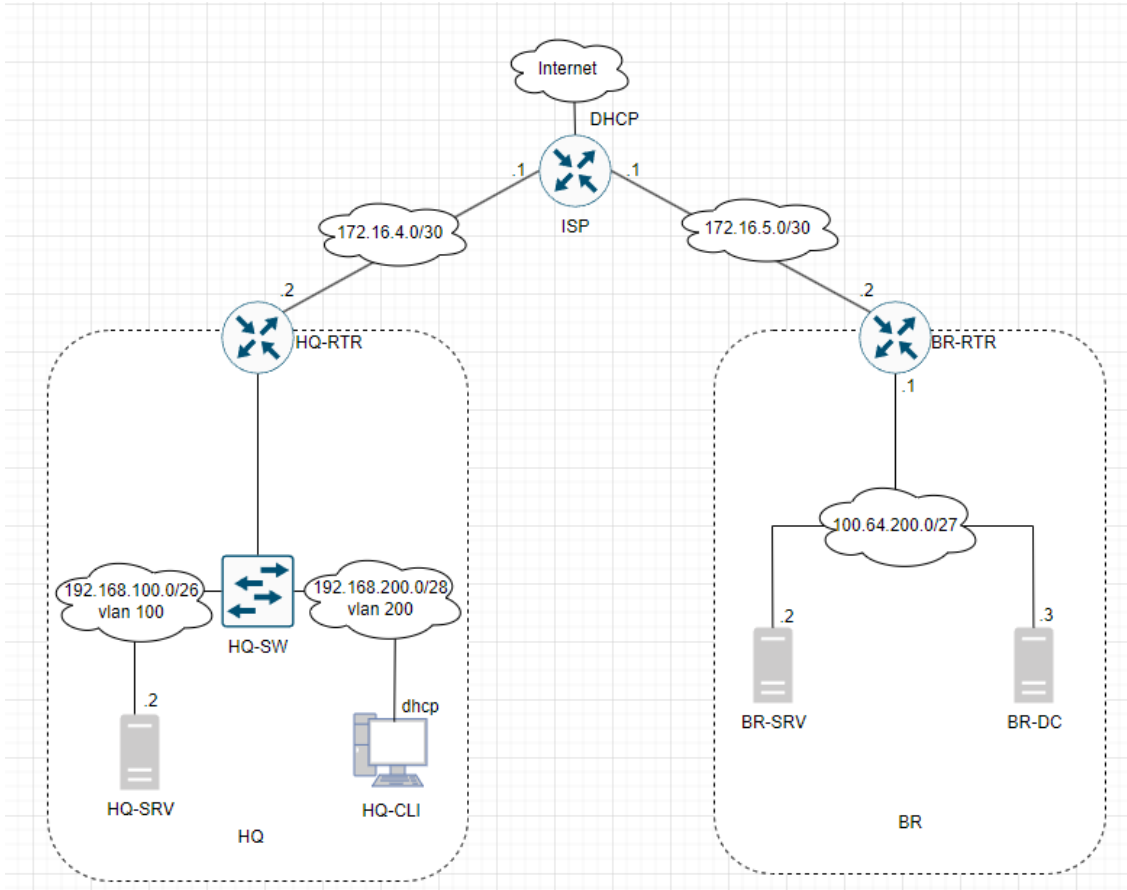
Увидим, что машине не переименовалась, даже после команды *exec bash* от лица суперпользователя. Поэтому делаем *reboot* системы и снова проверяем, и видим, что все выполнилось.

```
[user@hq-cli Рабочий стол]$ hostname -f  
hq-cli.ks54.net  
[user@hq-cli Рабочий стол]$
```

Перед настройкой IPv4 адресов необходимо сперва распределить адреса по всем интерфейсам. Сделаем такую таблицу по примеру Таблицы 3 из задания. (в нашем случае машина ISP пред настроена уже, там ничего менять не нужно)

Устройство	Интерфейс	IP-адрес	Маска	VLAN	Подсеть	Шлюз
ISP	ens18 (к интернету)	DHCP	DHCP	-	DHCP	DHCP
	ens19 (к HQ-RTR)	172.16.4.1	255.255.255.252	-	172.16.4.0/30	-
	ens20 (к BR-RTR)	172.16.5.1	255.255.255.252	-	172.16.5.0/30	-
HQ-RTR	ens18 (к ISP)	172.16.4.2	255.255.255.252	-	172.16.4.0/30	172.16.4.1
	ens19 (Trunk)	-	-	Trunk	-	-
	ens19.100	192.168.100.1	255.255.255.192	100	192.168.100.0/26	-
	ens19.200	192.168.200.1	255.255.255.240	200	192.168.200.0/28	-
	ens19.999	192.168.3.1	255.255.255.248	999	192.168.3.0/29	-
	tungre (IP туннель)	10.10.10.1	255.255.255.252	-	10.10.10.0/30	-
HQ-SRV	ens18 (Trunk)	-	-	Trunk	-	-
	ens18.100	192.168.100.2	255.255.255.192	100	192.168.100.0/26	192.168.100.1
HQ-CLI	ens18.200	192.168.200.2	255.255.255.240	200	192.168.200.0/28	192.168.200.1
BR-RTR	ens18 (к ISP)	172.16.5.2	255.255.255.252	-	172.16.5.0/30	172.16.5.1
	ens19 (к BR-SRV)	100.64.200.1	255.255.255.224	-	100.64.200.0/27	-
	tungre (IP туннель)	10.10.10.2	255.255.255.252	-	10.10.10.0/30	-
BR-SRV	ens18 (к BR-RTR)	100.64.200.2	255.255.255.224	-	100.64.200.0/27	100.64.200.1

На основе данной таблицы сделаем схему для более наглядного представления структуры сети



Теперь настроим адресацию на всех машинах.

HQ-RTR

Проверим какие интерфейсы активны в окружении системы с помощью команды `ip --br -c a`

```
[root@a14nu6q7gr1gs ~]# ip --br -c a
lo                UNKNOWN      127.0.0.1/8 ::1/128
ens18             UP           fe80::bc24:11ff:fee1:22dc/64
ens19             UP           fe80::bc24:11ff:fe2e:7e6a/64
[root@a14nu6q7gr1gs ~]#
```

Для более подробного отображения информации о сетевых параметрах интерфейсов можно использовать команду `ip a`

```
[root@a14nu6q7gr1gs ~]# 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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:e1:22:dc brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet6 fe80::bc24:11ff:fee1:22dc/64 scope link
        valid_lft forever preferred_lft forever
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:2e:7e:6a brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet6 fe80::bc24:11ff:fe2e:7e6a/64 scope link
        valid_lft forever preferred_lft forever
[root@a14nu6q7gr1gs ~]#
```

Необходимо соотнести MAC-адреса интерфейсов и понять куда какой интерфейс смотрит. Для это сравниваем эти параметры из самой системы и из свойств соответствующей виртуальной машины на стенде.

Сводка	Добавить	Удалить	Редактировать	Действие над диском	Сбросить
Консоль	Память	1.00 ГиБ			
Оборудование	Процессоры	1 (1 sockets, 1 cores) [host]			
Cloud-Init	BIOS	По умолчанию (SeaBIOS)			
Параметры	Экран	По умолчанию			
Журнал задач	Машина	По умолчанию (i440fx)			
Монитор	Контроллер SCSI	VirtIO SCSI single			
Резервная копия	CD/DVD-диск (ide2)	local:iso/alt-server-10.2-x86_64.iso,media=cdrom,size=5074118K			
Репликация	Жесткий диск (scsi0)	local-lvm:vm-101-disk-0,ioread=1,size=10G			
Снимки	Сетевое устройство (net0)	virtio=BC:24:11:E1:22:DC,bridge=ISP_HQ			
	Сетевое устройство (net1)	virtio=BC:24:11:2E:7E:6A,bridge=HQ			

Зайдем в директорию интерфейсов и настроим их. Первый интерфейс `ens18` смотрит в сторону машины ISP, `ens19` – транковый порт, разделяющийся на VLANы для HQ-SRV и HQ-CLI, следовательно, навешиваем на них соответствующие параметры:

ens18:

```
[root@hq-rtr ~]# cd /etc/net/ifaces/ens18/
```

проверяем какие файлы имеются в папке интерфейса, для этого пишем либо `ls`

```
[root@hq-rtr ens18]# ls
options
[root@hq-rtr ens18]#
```

либо запускаем миднайт командер командой `mc`

```
[root@hq-rtr ens18]# mc
```


Left	File	Command	Options	Right
< /etc/net/iface/ens18				< /etc/net/iface/ens18
.n	Name		Size	.n
options			UP-DIR Nov 30 18:31	options
			137 Nov 30 18:31	

Для выхода можно используем клавишу F10.

Создадим в этой папке файл, отвечающий за подтягивание параметров ipv4-конфигурации на данный интерфейс в систему. Для этого есть несколько способов:

1. echo 172.16.4.2/30 > ipv4address

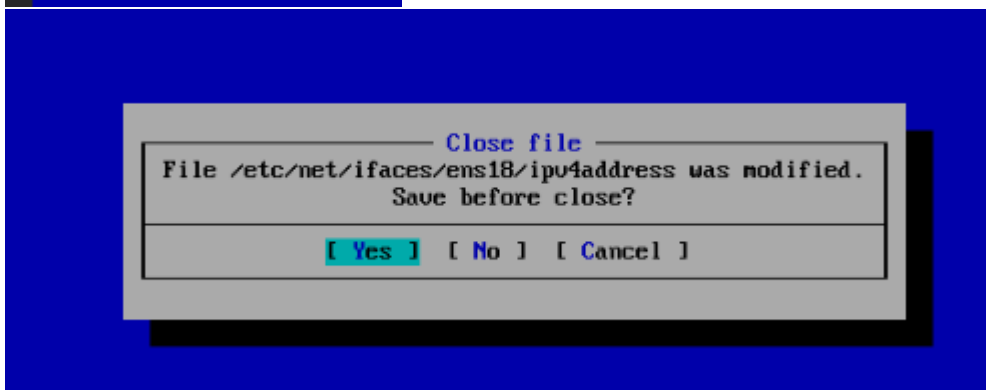
```
[root@hq-rtr ens18]# echo 172.16.4.2/30 > ipv4address
```

2. mcedit ipv4address

```
[root@hq-rtr ens18]# mcedit ipv4address
```

затем вписать 172.16.4.2/30, нажать F10 и подтвердить сохранение параметров.

```
ipv4address [-M]
172.16.4.2/30_
```



Проверяем, что появился файл ipv4address через *ls* или *mc*

```
[root@hq-rtr ens18]# ls
ipv4address options
[root@hq-rtr ens18]#
```

или

Left	File	Command	Options	Right
< /etc/net/iface/ens18				< /etc/net/iface/ens18
.n	Name			.n
options				options

Также можно прописать ссылку на DNS сервер, чтобы в будущем наша машина могла стучаться до сети Интернет. Для этого в этой же папке создаем файл *resolv.conf* со следующим содержимым: *nameserver 77.88.8.8*.

```
[root@hq-rtr ens18]# echo nameserver 77.88.8.8 > resolv.conf
[root@hq-rtr ens18]# ls
ipv4address options resolv.conf
[root@hq-rtr ens18]# cat resolv.conf
nameserver 77.88.8.8
[root@hq-rtr ens18]#
```

Аналогично создаем файл со шлюзом по умолчанию на ISP.

```
[root@hq-rtr ens18]# echo default via 172.16.4.1 > ipv4route
[root@hq-rtr ens18]#
```

Теперь настроим остальные интерфейсы. Перейдем в папку интерфейса *ens19*. Либо командой из текущего каталога *cd ..* далее *cd ens19*, либо через переход по полному пути *cd /etc/net/iface/ens19/* и проверим содержимое этой папки

```
[root@hq-rtr ens18]# cd ..
[root@hq-rtr ifaces]# cd ens19
[root@hq-rtr ens19]# ls
options
[root@hq-rtr ens19]#
```

или

```
[root@hq-rtr ens18]# cd /etc/net/iface/ens19
[root@hq-rtr ens19]# ls
options
[root@hq-rtr ens19]#
```

Настроим интерфейс в транковый режим и создадим виртуальные интерфейсы как по таблице.

Для этого отредактируем файл *options* в интерфейсе ens19 и приведем к следующему виду:

```
options
TYPE=eth
VLAN_AWARE=yes
VIDS="100 200 999"
CONFIG_WIRELESS=no
BOOTPROTO=static
SYSTEMD_BOOTPROTO=static
CONFIG_IPV4=yes
DISABLED=no
NM_CONTROLLED=no
SYSTEMD_CONTROLLED=no
```

Создадим папки с подинтерфейсами в директории */etc/net/ifaces/*, которые будут работать, используя мощности и пропускную способность физического интерфейса ens19.

```
[root@hq-rtr ifaces]# mkdir ens19.100
[root@hq-rtr ifaces]# mkdir ens19.200
[root@hq-rtr ifaces]# mkdir ens19.999
```

Создадим файл *options* и *ipv4address* в каждой созданной папке

```
[root@hq-rtr ens19.100]# mcedit options
```

```
options
TYPE=vlan
HOST=ens19
VID=100
BOOTPROTO=static
```

А также навесим *ipv4*-адрес на этот интерфейс:

```
[root@hq-rtr ens19.100]# echo 192.168.100.1/26 > ipv4address
```

Для проверки сделаем рестарт сетевых параметров и проверим *ip*-конфигурацию

```
[root@hq-rtr ens19.100]# systemctl restart network
[root@hq-rtr ens19.100]# ip --br -c a
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens18             UP         172.16.4.2/30 fe80::be24:11ff:fee1:22dc/64
ens19             UP         fe80::be24:11ff:fe2e:7e6a/64
ens19.100@ens19  UP         192.168.100.1/26 fe80::be24:11ff:fe2e:7e6a/64
```

Далее скопируем созданные файлы для ens19.100 для остальных подинтерфейсов и отредактируем по необходимым параметрам.

```
[root@hq-rtr ens19.100]# cp options /etc/net/ifaces/ens19.200/
[root@hq-rtr ens19.100]# cp ipv4address /etc/net/ifaces/ens19.200/
[root@hq-rtr ens19.100]# cp ipv4address /etc/net/ifaces/ens19.999/
[root@hq-rtr ens19.100]# cp options /etc/net/ifaces/ens19.999/
```

Отредактируем файлы:

Для *ens19.200*

```
[root@hq-rtr ifaces]# mcedit /etc/net/ifaces/ens19.200/options
```

```
options
TYPE=vlan
HOST=ens19
VID=200
BOOTPROTO=static
```

```
[root@hq-rtr ifaces]# echo 192.168.200.1/26 > /etc/net/ifaces/ens19.200/ipv4address
```

Для *ens19.999*

```
[root@hq-rtr ifaces]# mcedit /etc/net/ifaces/ens19.999/options
```

```
options [-----]
TYPE=ulan
HOST=ens19
VID=999
BOOTPROTO=static
```

```
[root@hq-rtr ifaces]# echo 192.168.3.1/29 > /etc/net/ifaces/ens19.999/ipv4address
```

Делаем рестрат сетевых параметров и проверяем ip-конфигурацию

```
[root@hq-rtr ifaces]# systemctl restart network
[root@hq-rtr ifaces]# ip --br -c a
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens18             UP         172.16.4.2/30 fe80::be24:11ff:fee1:22dc/64
ens19             UP         fe80::be24:11ff:fe2e:7e6a/64
ens19.100@ens19   UP         192.168.100.1/26 fe80::be24:11ff:fe2e:7e6a/64
ens19.200@ens19   UP         192.168.200.1/28 fe80::be24:11ff:fe2e:7e6a/64
ens19.999@ens19   UP         192.168.3.1/29 fe80::be24:11ff:fe2e:7e6a/64
[root@hq-rtr ifaces]#
```

Для пересылки пакетов между подсетями включим forwarding на машине.

```
[root@hq-rtr ifaces]# mcedit /etc/net/sysctl.conf
```

```
sysctl.conf [-----] 23 L:[ 1+ 9 10/ 531 *(279 /1987b) 0010 0x00A
# This file was formerly part of /etc/sysctl.conf
### IPV4 networking options.

# IPv4 packet forwarding.
#
# This variable is special, its change resets all configuration
# parameters to their default state (RFC 1122 for hosts, RFC 1812 for
# routers).
net.ipv4.ip_forward = 1_
# Source validation by reversed path, as specified in RFC 1812.
#
# Recommended option for single homed hosts and stub network routers.
# Could cause troubles for complicated (not loop free) networks
# running a slow unreliable protocol (sort of RIP), or using static
# routes.
```

HQ-SRV

Проверим интерфейсы в системе

```
[root@hq-srv ~]# ip --br -c a
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens18             UP         fe80::be24:11ff:fe32:59a9/64
[root@hq-srv ~]#
```

Настроим интерфейс в транковый режим и создадим виртуальные интерфейсы как по таблице.

Для этого отредактируем файл *options* в интерфейсе ens18

```
[root@hq-srv ifaces]# mcedit ens18/options
```

и приведем к следующему виду:

```
options [----] 0 L:[]
TYPE=eth
VLAN_AWARE=yes
VID="100"
CONFIG_WIRELESS=no
BOOTPROTO=static
SYSTEMD_BOOTPROTO=static
CONFIG_IPV4=yes
DISABLED=no
NM_CONTROLLED=no
SYSTEMD_CONTROLLED=no
```

Создадим папку с подинтерфейсом, который будут работать на физическом интерфейсе ens18.

```
[root@hq-srv ifaces]# mkdir ens18.100
```

Настроим этот интерфейс

```
[root@hq-srv ifaces]# cd ens18.100
```

```
[root@hq-srv ens18.100]# echo default via 192.168.100.1 > ipv4route
```

```
[root@hq-srv ens18.100]# echo 192.168.100.2/26 > ipv4address
```

```
[root@hq-srv ens18.100]# mcedit options
```

```
options [---]
TYPE=eth
HOST=ens18
VID=100
BOOTPROTO=static
```

Сделаем рестарт сетевой конфигурации и проверим адресацию

```
[root@hq-srv ens18.100]# systemctl restart network
[root@hq-srv ens18.100]# ip --br -c a
lo UNKNOWN 127.0.0.1/8 ::1/128
ens18 UP fe80::be24:11ff:fe32:59a9/64
ens18.100@ens18 UP 192.168.100.2/26 fe80::be24:11ff:fe32:59a9/64
```

Проверим доступность ближайшего интерфейса роутера HQ-RTR

```
[root@hq-srv ens18.100]# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.828 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.422 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.661 ms
^C
--- 192.168.100.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/ndev = 0.422/0.637/0.828/0.166 ms
```

Как видим есть базовая сетевая связанность с роутером. Мы можем простучать и интерфейс роутера, который смотрим в ISP за счет форвардинга.

```
[root@hq-srv ens18.100]# ping 172.16.4.2
PING 172.16.4.2 (172.16.4.2) 56(84) bytes of data.
64 bytes from 172.16.4.2: icmp_seq=1 ttl=64 time=0.403 ms
64 bytes from 172.16.4.2: icmp_seq=2 ttl=64 time=0.318 ms
64 bytes from 172.16.4.2: icmp_seq=3 ttl=64 time=0.580 ms
64 bytes from 172.16.4.2: icmp_seq=4 ttl=64 time=0.355 ms
64 bytes from 172.16.4.2: icmp_seq=5 ttl=64 time=0.373 ms
^C
```

И второй виртуальный интерфейс с VLAN200

```
[root@hq-srv ens18.100]# ping 192.168.200.1
PING 192.168.200.1 (192.168.200.1) 56(84) bytes of data.
64 bytes from 192.168.200.1: icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from 192.168.200.1: icmp_seq=2 ttl=64 time=0.350 ms
64 bytes from 192.168.200.1: icmp_seq=3 ttl=64 time=0.574 ms
^C
--- 192.168.200.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2051ms
rtt min/avg/max/mdev = 0.350/0.453/0.574/0.092 ms
```

И vlan999

```
[root@hq-srv ens18.100]# ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.557 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=0.274 ms
^C
--- 192.168.3.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.274/0.464/0.563/0.134 ms
```

HQ-CLI

По заданию дальше адресация для данной машины должна выдаваться автоматически, но для теста, настроим адресацию статически, дальше, когда будет поднят сервер DHCP, тогда установить автоматическое получение адреса.

Для начала нам необходимо на интерфейсе в настройках оборудования машины выставить разрешение влана 200. Для этого не будем использовать виртуальный подинтерфейс, а используем второй способ – в свойствах виртуальной машины в оборудовании находим сетевую карту, кликаем дважды и в параметре VLAN выставляем значение 200.

В виде серверов

Виртуальная машина 103 (HQ-CLI) на узле de25

Сводка

Консоль

Оборудование

Cloud-Init

Параметры

Журнал задач

Монитор

Резервная копия

Репликация

Снимки

Добавить

Удалить

Редактировать

Действие над диском

Память: 3.00 ГиБ

Процессоры: 2 (1 sockets, 2 cores) [host]

BIOS: По умолчанию (SeaBIOS)

Экран: По умолчанию

Машина: По умолчанию (i440fx)

Контроллер SCSI: VirtIO SCSI single

CD/DVD-диск (ide2): local:iso/alt-workstation-10.2-x86_64.iso,media=

Жёсткий диск (scsi0): local-lvm:vm-103-disk-0,ioread=1,size=25G

Сетевое устройство (net0): virtio=BC:24:11:6B:5D:CD,bridge=HQ

Редактировать: Сетевое устройство

Сетевой мост: HQ

Модель: VirtIO (паравиртуализованная)

Тег VLAN: без VLAN

MAC-адрес: BC:24:11:6B:5D:CD

Сетевой экран: ☐

Справка

Дополнительно ☐

OK

Редактировать: Сетевое устройство

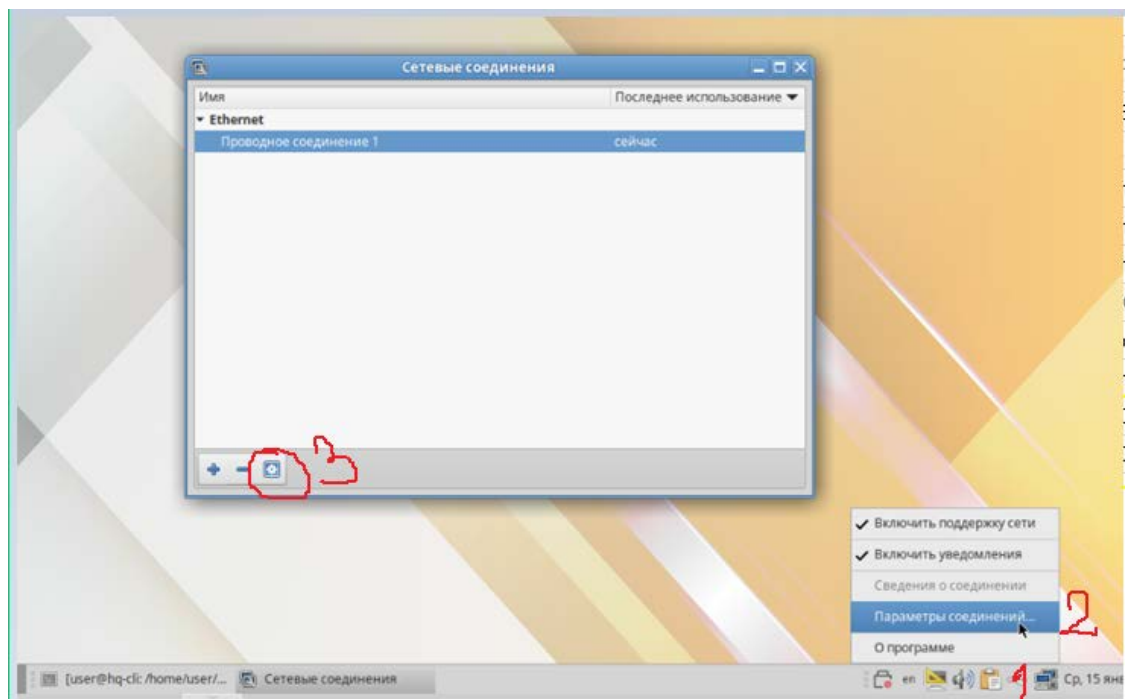
Сетевой мост: Модель:

Тег VLAN: MAC-адрес:

Сетевой экран: ☐

☐

Далее в настройках адаптера выставляем нужные параметры:



Изменение Проводное соединение 1

Имя соединения:

Основные: Ethernet | Стандарт безопасности 802.1x | DCB | Прокси | **Параметры IPv4** | Параметры IPv6

Устройство:

Клонированный MAC-адрес:

MTU: байт

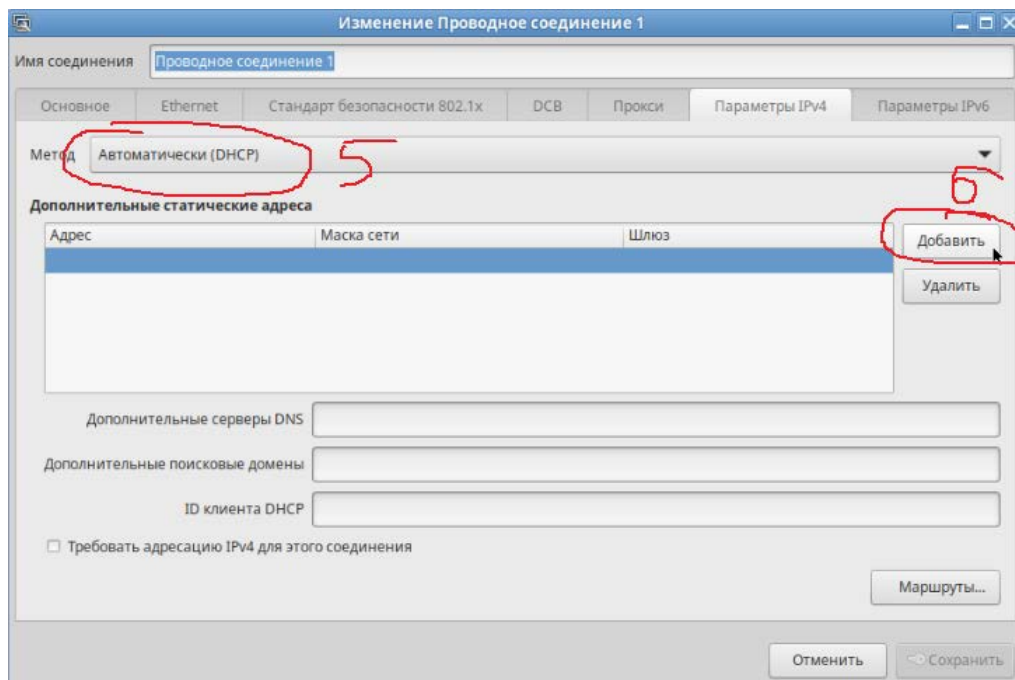
Wake on LAN: ☒ По умолчанию ☐ Физический уровень ☐ Одноадресная передача ☐ Многоадресная передача
☐ Игнорировать ☐ Широковещание ☐ Arp ☐ Magic

Пароль для Wake on LAN:

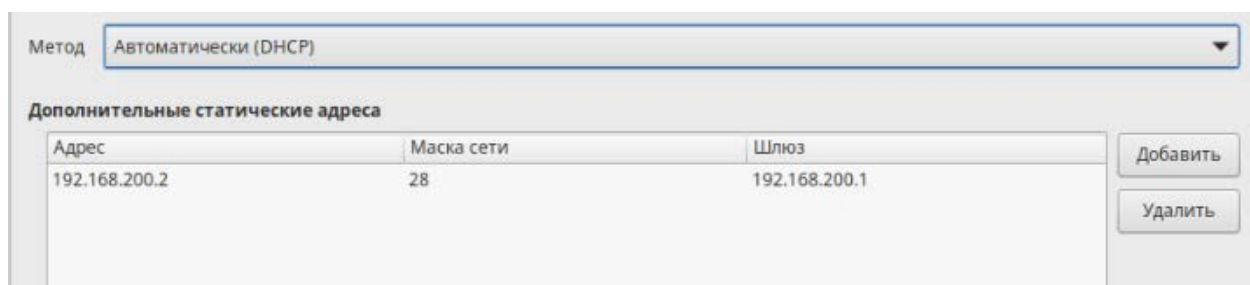
Согласование каналов:

Скорость:

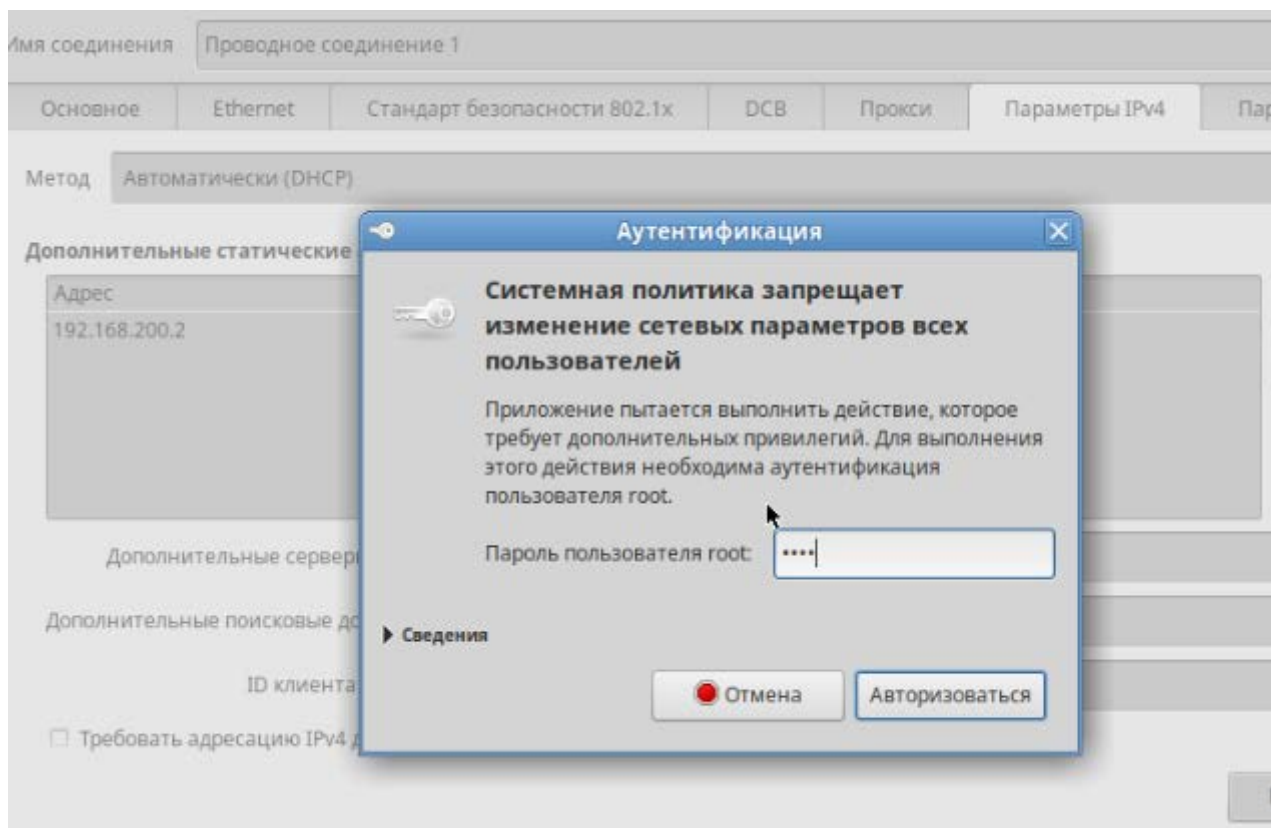
Дуплекс:



И прописываем айпи из таблицы, которая приведена в начале.

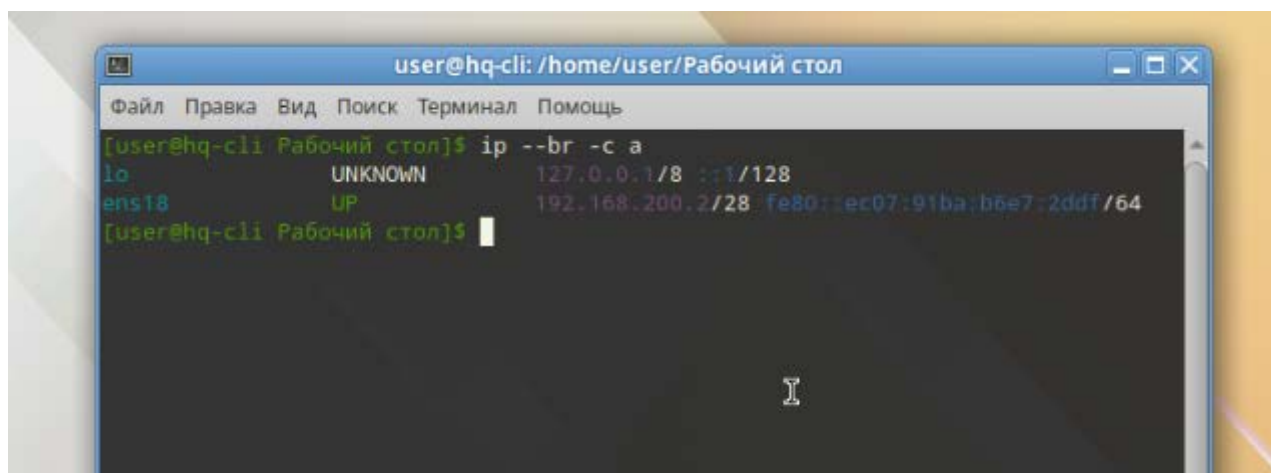
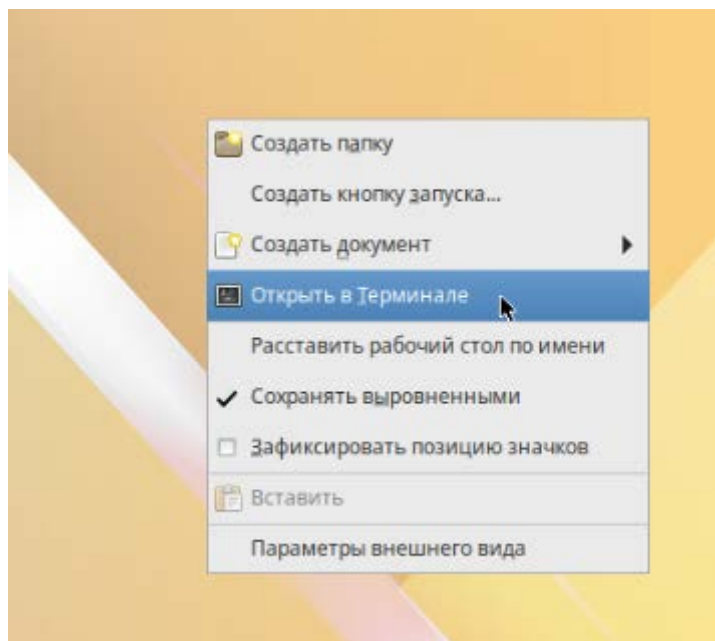
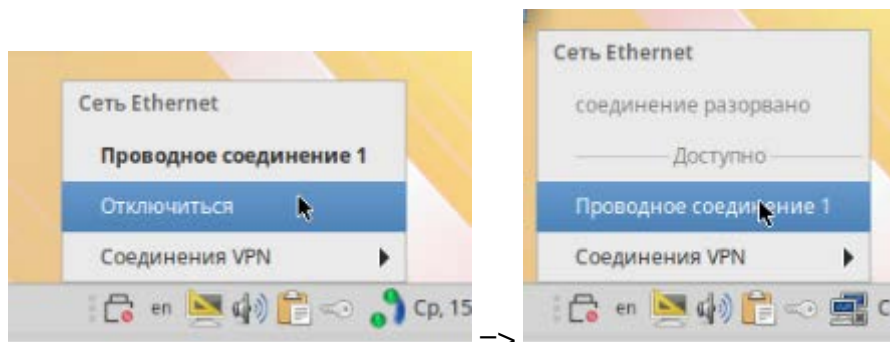


Нажимаем сохранить и вбиваем пароль *toor*

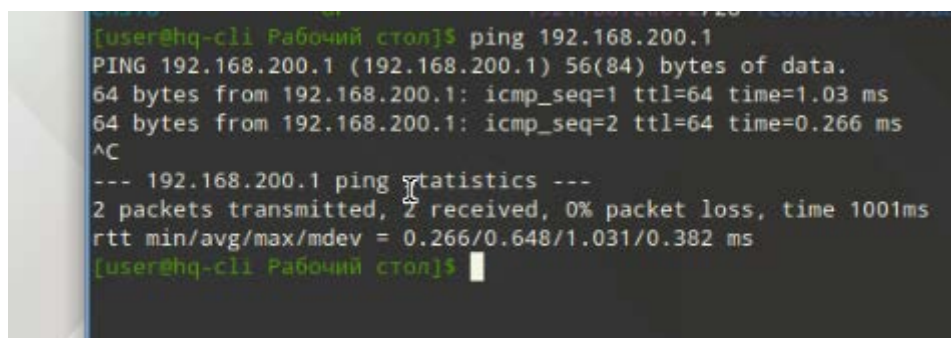


Далее применим полученные параметры отключив и включив адаптер через апплет в строке меню.

Левой кнопкой мыши щелкаем на апплет, нажимаем отключиться, затем тоже самое и выбрать проводное соединение. Он начнет грузить подключение, не подтверждая успешное подключение (так как у нас остался параметр получать айпи по DHCP), но мы можем удостовериться, что параметры применились, зайдя в консоль и вбив `ip -br -c a`.



Проверяем сетевую связанность клиента с роутером HQ-RTR



Также доступны и другие интерфейсы на роутере

```
[user@hq-cll Рабочий стол]$ ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.834 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.381 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.629 ms
^C
--- 192.168.100.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2085ms
rtt min/avg/max/mdev = 0.381/0.614/0.834/0.185 ms
[user@hq-cll Рабочий стол]$
```

НО как только в трее апплет перестанет грузиться и выдаст недоступное подключение, параметры сети сбросятся, так как у нас стоит параметр получать по DHCP. Можно на время поставить статический адрес, но потом не забудьте изменить его на DHCP, когда поднимите сервер DHCP.

BR-RTR

Настроим айпи конфигурацию для роутера аналогично HQ-RTR, за исключением подинтерфейсов.

Настройка ens18 смотрящего в сторону ISP

```
[root@br-rtr ~]# cd /etc/net/ifaaces/ens18
[root@br-rtr ens18]# echo 172.16.5.2/30 > ipv4address
```

```
[root@br-rtr ens18]# echo default via 172.16.5.1 > ipv4route
[root@br-rtr ens18]# echo nameserver 77.88.8.8 > resolv.conf
```

Настройка ens19 в сторону BR-SRV

```
[root@br-rtr ens18]# cd /etc/net/ifaaces/ens19
[root@br-rtr ens19]# echo 100.64.200.1/27 > ipv4address
```

Делаем рестарт сети и проверяем конфигурацию

```
[root@br-rtr ens19]# systemctl restart network
[root@br-rtr ens19]# ip --br -c a
lo                UNKNOWN      127.0.0.1/8 ::1/128
ens18             UP           172.16.5.2/30 fe80::be24:11ff:fec2:af56/64
ens19             UP           100.64.200.1/27 fe80::be24:11ff:fec1:cedb/64
[root@br-rtr ens19]#
```

Для пересылки пакетов между подсетями включим forwarding на машине.

```
[root@hq-rtr ifaces]# mcedit /etc/net/sysctl.conf
```

```
sysctl.conf  [----] 23 L:[ 1+ 9 10/ 53] *(279 /1987b) 0010 0x00A
# This file was formerly part of /etc/sysctl.conf
### IPV4 networking options.

# IPv4 packet forwarding.
#
# This variable is special, its change resets all configuration
# parameters to their default state (RFC 1122 for hosts, RFC 1812 for
# routers).
net.ipv4.ip_forward = 1
# Source validation by reversed path, as specified in RFC 1812.
#
# Recommended option for single honed hosts and stub network routers.
# Could cause troubles for complicated (not loop free) networks
# running a slow unreliable protocol (sort of RIP), or using static
# routes.
```

BR-SRV

Аналогичным способом настраиваем конфигурацию сети

```
[root@br-srv ~]# ip --br -c a
lo UNKNOWN 127.0.0.1/8 ::1/128
ens18 UP fe80::be24:11ff:fe37:ddc7/64
[root@br-srv ~]# cd /etc/net/ifaces/ens18/
[root@br-srv ens18]# echo 100.64.200.2/27 > ipv4address
[root@br-srv ens18]# echo default via 100.64.200.1 > ipv4route
[root@br-srv ens18]# echo nameserver 77.88.8.8 > resolv.conf
```

```
[root@br-srv ens18]# systemctl restart network
[root@br-srv ens18]# ip --br -c a
lo UNKNOWN 127.0.0.1/8 ::1/128
ens18 UP 100.64.200.2/27 fe80::be24:11ff:fe37:ddc7/64
[root@br-srv ens18]#
```

11. НАСТРОЙКА ЧАСОВОГО ПОЯСА НА ВСЕХ УСТРОЙСТВАХ

Настройка производится встроенной службой, настроим зону на **HQ-SRV** следующей командой:
timedatectl set-timezone Europe/Moscow

Настроим на примере HQ-RTR

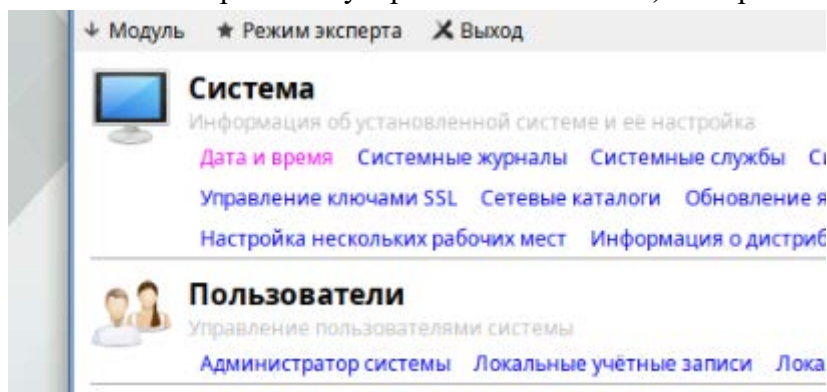
```
[root@hq-rtr ifaces]# timedatectl set-timezone Europe/Moscow
```

Проверим командой **timedatectl status**

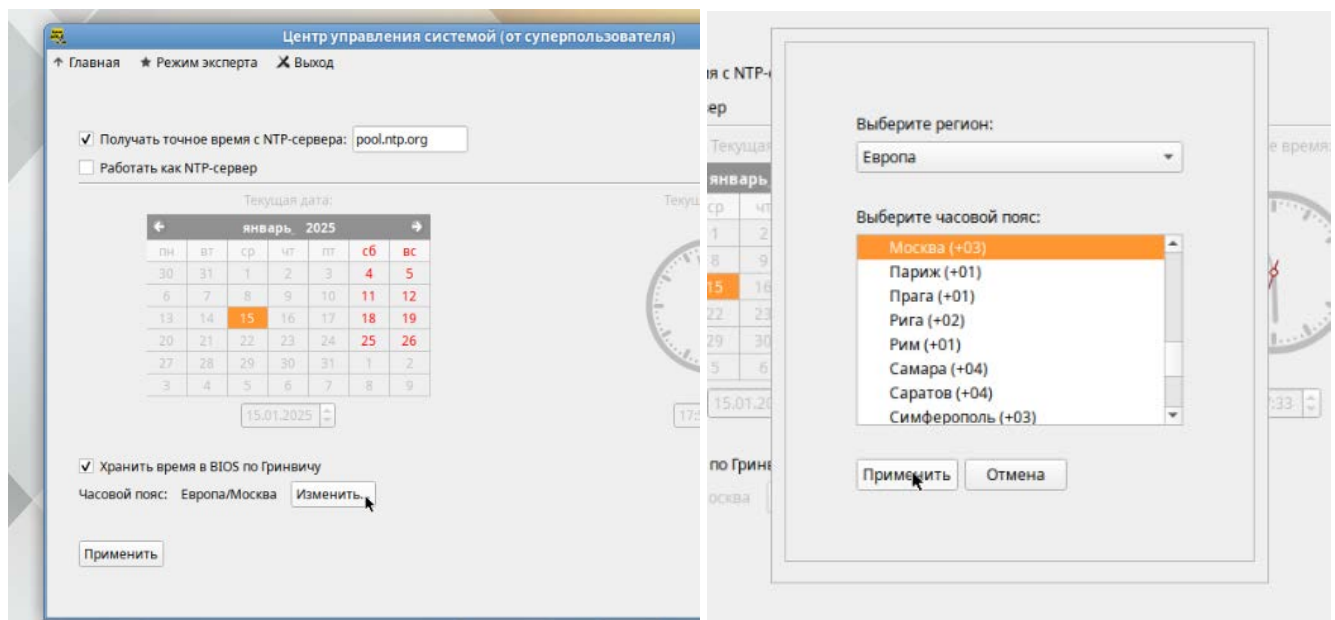
```
[root@hq-rtr ifaces]# timedatectl status
Local time: Wed 2025-01-15 17:51:14 MSK
Universal time: Wed 2025-01-15 14:51:14 UTC
RTC time: Wed 2025-01-15 14:51:14
Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
[root@hq-rtr ifaces]#
```

Аналогично проделываем на всех машинах без графики. На HQ-CLI можно выполнить также в терминале данную команду, либо через центр управления в графическом режиме выбрать таймзону.

Для это в стартовом меню выбираем Центр управления системой (или вбиваем в поисковой строке **асс**), затем вбиваем пароль от суперпользователя **toor**, выбираем в разделе системы **Дата и время**



Нажимаем внизу **Изменить** и проверяем, что установлено Европа/Москва



И нажимаем применить параметры.

6. НАСТРОЙКА IP-ТУННЕЛЯ МЕЖДУ ОФИСАМИ HQ И BR:

Создание туннеля производится на маршрутизаторах **HQ-RTR** и **BR-RTR**.

HQ-RTR:

По аналогии с созданием подинтерфейсов для VLANов создается папка для виртуального интерфейса **tungre** и создаем файл **options**

```
[root@hq-rtr ifaces]# mkdir tungre
[root@hq-rtr ifaces]# cd tungre/
[root@hq-rtr tungre]# mcedit options
```

со следующим содержанием

```
options [-M-]
TYPE=iptun
TUNTYPE=gre
TUNLOCAL=172.16.4.2
TUNREMOTE=172.16.5.2
HOST=ens18
TUNOPTIONS='ttl 255'
EOF
```

Навешиваем ip-адрес на интерфейс

```
[root@hq-rtr tungre]# echo 10.10.10.1/30 > ip4address
[root@hq-rtr tungre]#
```

Включаем модуль **gre**

```
[root@hq-rtr tungre]# modprobe gre
[root@hq-rtr tungre]#
```

Сохраняем:

```
[root@hq-rtr tungre]# echo "gre" | tee -a /etc/modules
```

На выходе получим сообщение

```
[root@hq-rtr tungre]# echo "gre" | tee -a /etc/modules
gre
[root@hq-rtr tungre]#
```

Перезапускаем сетевые службы и проверяем ip-конфигурацию, что появился туннель

```
[root@hq-rtr tungre]# systemctl restart network
[root@hq-rtr tungre]# ip --br -c a
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens18             UP        172.16.4.2/30 fe80::be24:11ff:fee1:22dc/64
ens19             UP        fe80::be24:11ff:fc2e:7e6a/64
ens19.100@ens19   UP        192.168.100.1/26 fe80::be24:11ff:fe2e:7e6a/64
ens19.200@ens19   UP        192.168.200.1/28 fe80::be24:11ff:fe2e:7e6a/64
ens19.999@ens19   UP        192.168.3.1/29 fe80::be24:11ff:fe2e:7e6a/64
gre0@NONE         DOWN
gretap0@NONE      DOWN
crspan0@NONE      DOWN
tungre@ens18      UNKNOWN    10.10.10.1/30 fe80::ac10:402/64
[root@hq-rtr tungre]#
```

BR-RTR:

Аналогичным образом настраиваем туннель с обратной стороны, только поменяв значения TUNLOCAL и TUNREMOTE.

```
[root@br-rtr ifaces]#
[root@br-rtr ifaces]# mkdir tungre
[root@br-rtr ifaces]# cd tungre/
[root@br-rtr tungre]# nccedit options
```

```
options      [-M--]
TYPE=iptun
TUNTYPE=gre
TUNLOCAL=172.16.5.2
TUNREMOTE=172.16.4.2
HOST=ens18
TUNOPTIONS='ttl 255'
EOF
```

Навешиваем адрес

```
[root@br-rtr tungre]# echo 10.10.10.2/30 > ip4address
[root@br-rtr tungre]#
```

Включаем модуль

```
[root@br-rtr tungre]# modprobe gre
[root@br-rtr tungre]#
```

Сохраняем загрузку этого модуля и смотрим вывод

```
[root@br-rtr tungre]# echo "gre" | tee -a /etc/modules
```

```
[root@hq-rtr tungre]# echo "gre" | tee -a /etc/modules
gre
[root@hq-rtr tungre]#
```

Перезапускаем службу и проверяем конфигурацию

```
[root@br-rtr tungre]# systemctl restart network
[root@br-rtr tungre]# ip --br -c a
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens18             UP        172.16.5.2/30 fe80::be24:11ff:fee2:af56/64
ens19             UP        100.64.200.1/27 fe80::be24:11ff:fecl:cedb/64
gre0@NONE         DOWN
gretap0@NONE      DOWN
crspan0@NONE      DOWN
tungre@ens18      UNKNOWN    10.10.10.2/30 fe80::ac10:502/64
[root@br-rtr tungre]#
```

Протестируем работу нашего тоннеля


```
[root@br-rtr tungrel]# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=3.71 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.787 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.764 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=1.01 ms
^C
--- 10.10.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.764/1.567/3.710/1.240 ms
[root@br-rtr tungrel]#
```

Перейдем на HQ-RTR и сделаем ping

```
[root@hq-rtr tungrel]# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=2.88 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.476 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.957 ms
^C
--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2054ms
rtt min/avg/max/mdev = 0.476/1.436/2.875/1.036 ms
[root@hq-rtr tungrel]#
```

Туннель работает

8. НАСТРОЙКА ДИНАМИЧЕСКОЙ ТРАНСЛЯЦИИ АДРЕСОВ NAT

Настроим NAT на роутерах HQ-RTR и BR-RTR

HQ-RTR:

Сделаем трансляцию адресов с помощью iptables.

Введем правила для трансляции подсетей во внешнюю сеть

```
iptables -t nat -A POSTROUTING -s 192.168.100.0/26 -o ens18 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 192.168.200.0/28 -o ens18 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 192.168.3.0/29 -o ens18 -j MASQUERADE
```

```
[root@hq-rtr tungrel]# iptables -t nat -A POSTROUTING -s 192.168.100.0/26 -o ens18 -j MASQUERADE
[root@hq-rtr tungrel]# iptables -t nat -A POSTROUTING -s 192.168.200.0/28 -o ens18 -j MASQUERADE
[root@hq-rtr tungrel]# iptables -t nat -A POSTROUTING -s 192.168.3.0/29 -o ens18 -j MASQUERADE
```

Сохраним правила и поставим сервис в автозагрузку системы

```
iptables-save >> /etc/sysconfig/iptables
```

```
systemctl enable --now iptables.services
```

```
[root@hq-rtr tungrel]# iptables-save >> /etc/sysconfig/iptables
[root@hq-rtr tungrel]# systemctl enable --now iptables.service
Synchronizing state of iptables.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable iptables
Created symlink /etc/systemd/system/basic.target.wants/iptables.service → /lib/systemd/system/iptables.service.
```

Для проверки введенных правил вбиваем команду

```
iptables -t nat -L
```

```
[root@hq-rtr tungre]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  192.168.100.0/26       anywhere
MASQUERADE all  --  192.168.200.0/28       anywhere
MASQUERADE all  --  192.168.3.0/29         anywhere
```

Проверяем доступ в интернет

```
[root@hq-rtr tungre]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=101 time=39.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=101 time=29.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=101 time=28.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1990ms
rtt min/avg/max/mdev = 28.680/32.404/38.950/4.643 ms
[root@hq-rtr tungre]#
```

BR-RTR:

Выполняем аналогично по тому же принципу, но с другими данными

Введем правила для трансляции подсетей во внешнюю сеть

iptables -t nat -A POSTROUTING -s 100.64.200.0/27 -o ens18 -j MASQUERADE

```
[root@br-rtr tungre]# iptables -t nat -A POSTROUTING -s 100.64.200.0/27 -o ens18 -j MASQUERADE
[root@br-rtr tungre]#
```

Сохраняем и добавляем в автозагрузку

```
[root@br-rtr tungre]# iptables-save >> /etc/sysconfig/iptables
[root@br-rtr tungre]# systemctl enable --now iptables.service
Synchronizing state of iptables.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable iptables
Created symlink /etc/systemd/system/basic.target.wants/iptables.service → /lib/systemd/system/iptables.service.
[root@br-rtr tungre]#
```

Проверим правило

```
Created symlink /etc/systemd/system/basic.target.wants/iptables.service → /lib/systemd/system/iptables.service.
[root@br-rtr tungre]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  100.64.200.0/27       anywhere
[root@br-rtr tungre]#
```

Проверяем доступ в интернет


```
[root@br-rtr tungre]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=101 time=29.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=101 time=25.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=101 time=24.1 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 24.050/26.262/29.636/2.423 ms
[root@br-rtr tungre]#
```

7. НАСТРОЙКА ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ С ПОМОЩЬЮ LINK-STATE ПРОТОКОЛА OSPF:

Настройку маршрутизации будем проводить, используя пакет **frr**.

HQ-RTR:

Для начала проверим, что с роутера есть доступ в интернет по доменным именам. Для этого сделаем команду **ping ya.ru**

Если пинг идет, значит можем приступить к обновлению списка пакетов в репозитории и установке пакета **frr**.

```
[root@hq-rtr tungre]# ping ya.ru
PING ya.ru (213.180.193.56) 56(84) bytes of data.
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=239 time=10.8 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=2 ttl=239 time=10.6 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=3 ttl=239 time=23.5 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=4 ttl=239 time=13.8 ms
^C
--- ya.ru ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 10.565/14.655/23.456/5.236 ms
[root@hq-rtr tungre]#
```

Обновляем список репозитория

```
[root@hq-rtr tungre]# apt-get update
```

Устанавливаем пакет **frr** и подтверждаем установку нажав **Y**

```
[root@hq-rtr tungre]# apt-get install frr
```

```
[root@hq-rtr tungre]# apt-get install frr
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  libbabeltrace libbabeltrace-ctf libcares libnet-snmp35 libn13 libprotobuf-c1 libyang python3-module-babeltrace
The following NEW packages will be installed:
  frr libbabeltrace libbabeltrace-ctf libcares libnet-snmp35 libn13 libprotobuf-c1 libyang python3-module-babeltrace
0 upgraded, 9 newly installed, 0 removed and 146 not upgraded.
Need to get 7868kB of archives.
After unpacking 35.9MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic libbabeltrace-ctf 1.5.8-alt2:p10+296260.300.7.1e1658766630 [128kB]
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic libbabeltrace 1.5.8-alt2:p10+296260.300.7.1e1658766630 [37.1kB]
Get:3 http://ftp.altlinux.org p10/branch/x86_64/classic libcares 1.26.0-alt1:p10+341112.100.1.1e1700403992 [76.0kB]
Get:4 http://ftp.altlinux.org p10/branch/x86_64/classic libn13 3.5.0-alt1:sisyphus+275381.100.1.2e1624498107 [267kB]
Get:5 http://ftp.altlinux.org p10/branch/x86_64/classic libnet-snmp35 5.8-alt1:sisyphus+274516.10000.1.1e1623617097 [917kB]
Get:6 http://ftp.altlinux.org p10/branch/x86_64/classic libprotobuf-c1 1.3.3-alt1:sisyphus+278642.4400.10.2e1626393181 [18.0kB]
Get:7 http://ftp.altlinux.org p10/branch/x86_64/classic libyang 2.1.55-alt1:p10+320601.40.3.1e1686062462 [418kB]
Get:8 http://ftp.altlinux.org p10/branch/x86_64/classic python3-module-babeltrace 1.5.8-alt2:p10+296260.300.7.1e1658766630 [64.4kB]
Get:9 http://ftp.altlinux.org p10/branch/x86_64/classic frr 9.0.2-alt1:p10+340214.100.2.1e1707043740 [5943kB]
Fetched 7868kB in 0s (13.3MB/s)
Committing changes...
Preparing...
Updating / installing...
1: libbabeltrace-1.5.8-alt2
2: libbabeltrace-ctf-1.5.8-alt2
3: python3-module-babeltrace-1.5.8-alt2
4: libyang-2.1.55-alt1
5: libprotobuf-c1-1.3.3-alt1
6: libn13-3.5.0-alt1
7: libnet-snmp35-5.8-alt1
8: libcares-1.26.0-alt1
9: frr-9.0.2-alt1
Done.
[root@hq-rtr tungre]#
```

Далее нам нужно включить поддержку модуля **ospf**. Для этого заходим в файл **daemons** в директории **/etc/frr/**

Откроем этот файл любым удобным для вас текстовым редактором (mcedit/vi/можете установить предварительно nano и использовать его)

mcedit /etc/frr/daemons

```
[root@hq-rtr tungre]# mcedit /etc/frr/daemons
```

```
bgpd=no
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

меняем на yes

```
# The watchfrr, zebra and
#
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
pimd=no
pim6d=no
ldpd=no
```

и сохраняем.

Далее перезагружаем работу сервиса frr

```
[root@hq-rtr tungre]# systemctl restart frr.service
```

и делаем для него автозагрузку

```
[root@hq-rtr tungre]# systemctl enable --now frr.service
Synchronizing state of frr.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable frr
Created symlink /etc/systemd/system/multi-user.target.wants/frr.service → /lib/systemd/system/frr.service
[root@hq-rtr tungre]#
```

Входим в окружение нашего виртуального роутера командой **vttysh**

```
[root@hq-rtr tungre]# vtysh

Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

hq-rtr.ks54.net#
```

А дальше как в циско настраиваем ospf на роутере с нашими подсетями. Помним, что нам необходимо выставить работу протокола маршрутизации по туннелю с внутренними подсетями, не задействовал внешние сети, которые идут к ISP (см. задание)

```
conf t
router ospf
network 10.10.10.0/30 area 0
network 192.168.100.0/26 area 0
network 192.168.200.0/28 area 0
network 192.168.3.0/29 area 0
do wr mem
```

```
hq-rtr.ks54.net# conf t
hq-rtr.ks54.net(config)# router ospf
hq-rtr.ks54.net(config-router)# network 10.10.10.0/30 area 0
hq-rtr.ks54.net(config-router)# network 192.168.100.0/26 area 0
hq-rtr.ks54.net(config-router)# network 192.168.200.0/28 area 0
hq-rtr.ks54.net(config-router)# network 192.168.3.0/29 area 0
hq-rtr.ks54.net(config-router)# do wr mem
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
hq-rtr.ks54.net(config-router)#
```

Теперь настроим парольную защиту на нашем GRE туннеле через frr

```
exit
int tungre
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 P@ssw0rd
do write memory
```

```
hq-rtr.ks54.net(config-router)# exit
hq-rtr.ks54.net(config)# interface tungre
hq-rtr.ks54.net(config-if)# ip ospf authentication message-digest
hq-rtr.ks54.net(config-if)# ip ospf message-digest-key 1 md5 P@ssw0rd
hq-rtr.ks54.net(config-if)# do write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
hq-rtr.ks54.net(config-if)#
```

Таким образом OSPF на HQ-RTR настроен. Приступаем ко второй стороне.

BR-RTR:

Аналогично прошлому роутеру нам нужно установить данный пакет.

```
[root@br-rtr tungre]# ping ya.ru
PING ya.ru (213.180.193.56) 56(84) bytes of data:
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=239 time=11.4 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=2 ttl=239 time=16.8 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=3 ttl=239 time=34.3 ms
^C
--- ya.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/ndev = 11.416/20.854/34.302/9.763 ms
[root@br-rtr tungre]#
```

```
[root@br-rtr tungre]# apt-get update
Get:1 http://ftp.altlinux.org p10/branch/x86_64 release [4223B]
Get:2 http://ftp.altlinux.org p10/branch/x86_64-i586 release [1665B]
Get:3 http://ftp.altlinux.org p10/branch/noarch release [2844B]
Fetched 8732B in 0s (259kB/s)
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic pkglist [24.4MB]
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic release [137B]
Get:3 http://ftp.altlinux.org p10/branch/x86_64/gostcrypto pkglist [18.5kB]
Get:4 http://ftp.altlinux.org p10/branch/x86_64/gostcrypto release [140B]
Get:5 http://ftp.altlinux.org p10/branch/x86_64-i586/classic pkglist [17.9MB]
Get:6 http://ftp.altlinux.org p10/branch/x86_64-i586/classic release [142B]
Get:7 http://ftp.altlinux.org p10/branch/noarch/classic pkglist [7289kB]
Get:8 http://ftp.altlinux.org p10/branch/noarch/classic release [137B]
Fetched 49.7MB in 6s (7903kB/s)
E: Failed to fetch http://ftp.altlinux.org/pub/distributions/ALTlinux/p10/branch/x86_64/base/release.gostcrypto Checksum mismatch
E: Some index files failed to download. They have been ignored, or old ones used instead.
[root@br-rtr tungre]#
```

```
[root@br-rtr tungre]# apt-get install frr
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  libbabeltrace libbabeltrace-ctf libcares libnet-snmp3 libn13 libprotobuf-c1 libyang python3-module-babeltrace
The following NEW packages will be installed:
  frr libbabeltrace libbabeltrace-ctf libcares libnet-snmp3 libn13 libprotobuf-c1 libyang python3-module-babeltrace
0 upgraded, 9 newly installed, 0 removed and 146 not upgraded.
Need to get 7868kB of archives.
After unpacking 35.9MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic libbabeltrace-ctf 1.5.8-alt2:p10+296260.300.7.1e1658766630 [128kB]
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic libbabeltrace 1.5.8-alt2:p10+296260.300.7.1e1658766630 [37.1kB]
Get:3 http://ftp.altlinux.org p10/branch/x86_64/classic libcares 1.26.0-alt1:p10+341112.100.1.1e1700403992 [76.0kB]
Get:4 http://ftp.altlinux.org p10/branch/x86_64/classic libn13 3.5.0-alt1:sisyphus+275381.100.1.2e1624498107 [267kB]
Get:5 http://ftp.altlinux.org p10/branch/x86_64/classic libnet-snmp3 5.8-alt1:sisyphus+274516.10000.1.1e1623617097 [917kB]
Get:6 http://ftp.altlinux.org p10/branch/x86_64/classic libprotobuf-c1 1.3.3-alt1:sisyphus+278642.4400.10.2e1626393181 [18.0kB]
Get:7 http://ftp.altlinux.org p10/branch/x86_64/classic libyang 2.1.55-alt1:p10+320601.40.3.1e1686062462 [418kB]
Get:8 http://ftp.altlinux.org p10/branch/x86_64/classic python3-module-babeltrace 1.5.8-alt2:p10+296260.300.7.1e1658766630 [64.4kB]
Get:9 http://ftp.altlinux.org p10/branch/x86_64/classic frr 9.0.2-alt1:p10+340214.100.2.1e1707843748 [5943kB]
Fetched 7868kB in 1s (6830kB/s)
Committing changes...
Preparing...
Updating / installing...
1: libbabeltrace-1.5.8-alt2
2: libbabeltrace-ctf-1.5.8-alt2
3: python3-module-babeltrace-1.5.8-alt2
4: libyang-2.1.55-alt1
5: libprotobuf-c1-1.3.3-alt1
6: libn13-3.5.0-alt1
7: libnet-snmp3-5.8-alt1
8: libcares-1.26.0-alt1
9: frr-9.0.2-alt1
Done.
[root@br-rtr tungre]#
```

```
[root@br-rtr tungre]# mcedit /etc/frr/daemons
```

```

bgpd=no
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no

```

меняем на yes

```

# The watchfrr, zebra and
#
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
pimd=no
pim6d=no
ldpd=no

```

и сохраняем.

```

[root@br-rtr tungre]# systemctl restart frr
[root@br-rtr tungre]# systemctl enable --now frr.service
Synchronizing state of frr.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable frr
Created symlink /etc/systemd/system/multi-user.target.wants/frr.service → /lib/systemd/system/frr.service.
[root@br-rtr tungre]#

```

Настраиваем OSPF для туннеля и внутренних подсетей роутера BR-RTR

```

[root@br-rtr tungre]# vtysh

Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

br-rtr.ks54.net# conf t
br-rtr.ks54.net(config)# router ospf
br-rtr.ks54.net(config-router)# network 10.10.10.0/30 area 0
br-rtr.ks54.net(config-router)# network 100.64.200.0/27 area 0
br-rtr.ks54.net(config-router)# do write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
br-rtr.ks54.net(config-router)#

```

Теперь также настроим парольную защиту на нашем GRE туннеле через frr

```

br-rtr.ks54.net(config-router)# exit
br-rtr.ks54.net(config)# interface tungre

br-rtr.ks54.net(config-if)# ip ospf authentication message-digest
br-rtr.ks54.net(config-if)# ip ospf message-digest-key 1 md5 P0ssw0rd
br-rtr.ks54.net(config-if)# do write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
br-rtr.ks54.net(config-if)#

```

Теперь проверим наших соседей

```

br-rtr.ks54.net(config-if)# do show ip ospf neighbor

```

Neighbor ID	Pri	State	Up Time	Dead Time	Address	Interface	RXmtL	RqstL	DBsnL
192.168.200.1	1	Full/-	1m45s	33.342s	10.10.10.1	tungre:10.10.10.2	0	0	0

```

hq-rtr.ks54.net(config-if)# do show ip ospf neighbor

```

Neighbor ID	Pri	State	Up Time	Dead Time	Address	Interface	RXmtL	RqstL	DBsnL
172.16.5.2	1	Full/-	7m39s	35.535s	10.10.10.2	tungre:10.10.10.1	0	0	0

Проверяем пинг с одного роутра на все интерфейсы второго, должно все пинговаться.

Можно проверить трассировку

```

hq-rtr.ks54.net(config-if)# do traceroute 100.64.200.2
traceroute to 100.64.200.2 (100.64.200.2), 30 hops max, 60 byte packets
 1 10.10.10.2 (10.10.10.2) 0.745 ms 0.680 ms 0.650 ms
 2 100.64.200.2 (100.64.200.2) 1.419 ms 1.391 ms 1.354 ms
hq-rtr.ks54.net(config-if)#

```



```
br-rtr.ks54.net(config-if)# do traceroute 192.168.100.2
traceroute to 192.168.100.2 (192.168.100.2), 30 hops max, 60 byte packets
 1 10.10.10.1 (10.10.10.1) 0.764 ms 0.679 ms 0.642 ms
 2 192.168.100.2 (192.168.100.2) 1.359 ms 1.321 ms 1.290 ms
br-rtr.ks54.net(config-if)# do traceroute 192.168.200.2
traceroute to 192.168.200.2 (192.168.200.2), 30 hops max, 60 byte packets
 1 10.10.10.1 (10.10.10.1) 0.906 ms 0.847 ms 0.837 ms
 2 192.168.200.2 (192.168.200.2) 1.713 ms 1.707 ms 1.700 ms
br-rtr.ks54.net(config-if)# do traceroute 192.168.3.1
traceroute to 192.168.3.1 (192.168.3.1), 30 hops max, 60 byte packets
 1 192.168.3.1 (192.168.3.1) 1.601 ms 1.532 ms 1.521 ms
```

Но иногда ospf некорректно отображает свою работу, поэтому –

лучше сделать **reboot** роутеров HQ-RTR и BR-RTR.

После зайти в vtysh и вбить команду: **show ip route ospf**, она покажет какие сети объявлены.

```
[root@hq-rtr ~]# vtysh
```

```
hq-rtr.ks54.net# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

O 10.10.10.0/30 [110/10] is directly connected, tungre, weight 1, 00:03:42
O>* 100.64.200.0/27 [110/11] via 10.10.10.2, tungre, weight 1, 00:02:48
O 192.168.3.0/29 [110/11] is directly connected, ens19.999, weight 1, 00:03:27
O 192.168.100.0/26 [110/11] is directly connected, ens19.100, weight 1, 00:03:27
O 192.168.200.0/28 [110/11] is directly connected, ens19.200, weight 1, 00:03:27
```

Как видим на HQ-RTR объявлены все сети, что прописаны в настройках ospf и * отмечена сеть BR, которая объявлена через ospf.

Проверим соседей снова, чтобы корректно отобразить связь через наш тоннель.

```
hq-rtr.ks54.net# sh ip ospf neighbor
```

Neighbor ID	Pri	State	Up Time	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
172.16.5.2	1	Full/DR	3m51s	30.517s	10.10.10.2	tungre:10.10.10.1	0	0	0

```
hq-rtr.ks54.net#
```

Проверим теперь на роутре BR-RTR.

```
[root@br-rtr ~]# vtysh
```

```
Hello, this is FRRouting (version 9.0.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
br-rtr.ks54.net# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

O 10.10.10.0/30 [110/10] is directly connected, tungre, weight 1, 00:03:16
O 100.64.200.0/27 [110/11] is directly connected, ens19, weight 1, 00:03:16
O>* 192.168.3.0/29 [110/11] via 10.10.10.1, tungre, weight 1, 00:02:16
O>* 192.168.100.0/26 [110/11] via 10.10.10.1, tungre, weight 1, 00:02:16
O>* 192.168.200.0/28 [110/11] via 10.10.10.1, tungre, weight 1, 00:02:16
```

```
br-rtr.ks54.net# sh ip ospf neighbor
```

Neighbor ID	Pri	State	Up Time	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
172.16.4.2	1	Full/Backup	3m25s	30.464s	10.10.10.1	tungre:10.10.10.2	0	0	0

Как видим теперь все корректно отображается.

9. НАСТРОЙКА ПРОТОКОЛА ДИНАМИЧЕСКОЙ КОНФИГУРАЦИИ ХОСТОВ:

Настройка DHCP-сервера может быть осуществлена различными способами, либо через установку и настройку напрямую dhcp-сервера из пакета, либо с помощью альтератора на клиентской машине. Но есть еще один способ, как на мой взгляд достаточно упрощенный и быстрый.

Если знаете, как альтернативную установку и настройку делать, то, пожалуйста, главное, чтобы цель была достигнута – установлен и настроен DHCP-сервер и выдает адрес для HQ-CLI из диапазона адресов.

Итак, приступим:

HQ-RTR:

Установим пакет **dnsmasq** (не удивляйтесь названию, все верно)

```
[root@hq-rtr ~]# apt-get update
```

```
[root@hq-rtr ~]# apt-get install dnsmasq
```

Получаем

```
[root@hq-rtr ~]# apt-get install dnsmasq
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  dnsmasq
0 upgraded, 1 newly installed, 0 removed and 147 not upgraded.
Need to get 360kB of archives.
After unpacking 834kB of additional disk space will be used.
Get:1 http://ftp.altlinux.org/pub/branch/x86_64/classic dnsmasq 2.90-alt1:p10+341077.200.2.101708366689 [360kB]
Fetched 360kB in 8s (3645kB/s)
Committing changes...
Preparing...
Updating / installing...
1: dnsmasq-2.90-alt1
Done.
[root@hq-rtr ~]#
```

Заходим в настройки конфигурационного файла сервиса

mcedit /etc/dnsmasq.conf

```
[root@hq-rtr ~]# mcedit /etc/dnsmasq.conf
```

Вносим следующие строки в начало файла:

```
no-resolv
dhcp-range=192.168.200.2,192.168.200.14,9999h
dhcp-option=1,255.255.255.240
dhcp-option=3,192.168.200.1
dhcp-option=6,192.168.100.2
interface=ens19.200
```

```
dnsmasq.conf  [-M--] 19 L: [ 1+ 6 7/700
# Configuration file for dnsmasq.
no-resolv
dhcp-range=192.168.200.2,192.168.200.14,9999h
dhcp-option=1,255.255.255.240
dhcp-option=3,192.168.200.1
dhcp-option=6,192.168.100.2
interface=ens19.200
#
```

dhcp-option=1 отвечает за маску подсети, передаваемого диапазона

dhcp-option=3 отвечает за пересылаемый dhcp-сервером адрес шлюза по умолчанию

dhcp-option=6 отвечает за пересылаемый dhcp-сервером адрес dns-сервера по умолчанию

С полным списком опций (Tag) может ознакомиться по адресу:

<https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml>

Далее нужно подтянуть параметры, которые мы указали. Для этого рестартуем сервис

systemctl restart dnsmasq.service

```
[root@hq-rtr ~]# systemctl restart dnsmasq.service
```

Проверим статус запущенного сервиса **systemctl status dnsmasq.service**

```
[root@hq-rtr ~]# systemctl status dnsmasq.service
dnsmasq.service - A lightweight DHCP and caching DNS server
Loaded: loaded (/lib/systemd/system/dnsmasq.service; disabled; vendor preset: disabled)
Active: active (running) since Mon 2025-01-20 13:28:33 MSK; 15s ago
Process: 3754 ExecStartPost=/usr/sbin/dnsmasq-helper poststart (code=exited, status=0/SUCCESS)
Main PID: 3752 (dnsmasq)
Tasks: 1 (limit: 1132)
Memory: 360.0K
CPU: 113ms
CGroup: /system.slice/dnsmasq.service
└─ 3752 /usr/sbin/dnsmasq --bind-interfaces --interface lo -s ks54.net -u _dnsmasq -k --pid-file

Jan 20 13:28:33 hq-rtr.ks54.net systemd[1]: Starting A lightweight DHCP and caching DNS server...
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq[3752]: started, version 2.90 cachesize 150
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq[3752]: compile time options: IPv6 GNU-getopt no-DBus no-UBus no-i18n IDN2 DHCP D
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq[3752]: warning: no upstream servers configured
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq-dhcp[3752]: DHCP, IP range 192.168.200.2 -- 192.168.200.14, lease time 416d15h
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq-dhcp[3752]: DHCP, sockets bound exclusively to interface ens19.200
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq[3752]: read /etc/hosts - 6 names
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq-helper[3879]: Setup resolv.conf for local resolver: succeeded
Jan 20 13:28:33 hq-rtr.ks54.net dnsmasq-helper[3754]: Setup resolv.conf for local resolver:[ DONE ]
Jan 20 13:28:33 hq-rtr.ks54.net systemd[1]: Started A lightweight DHCP and caching DNS server.
lines 1-21/21 (END)
```

Чтобы этот сервис запускался автоматически после перезагрузки системы добавим его в автозагрузку: **systemctl enable --now dnsmasq.service**

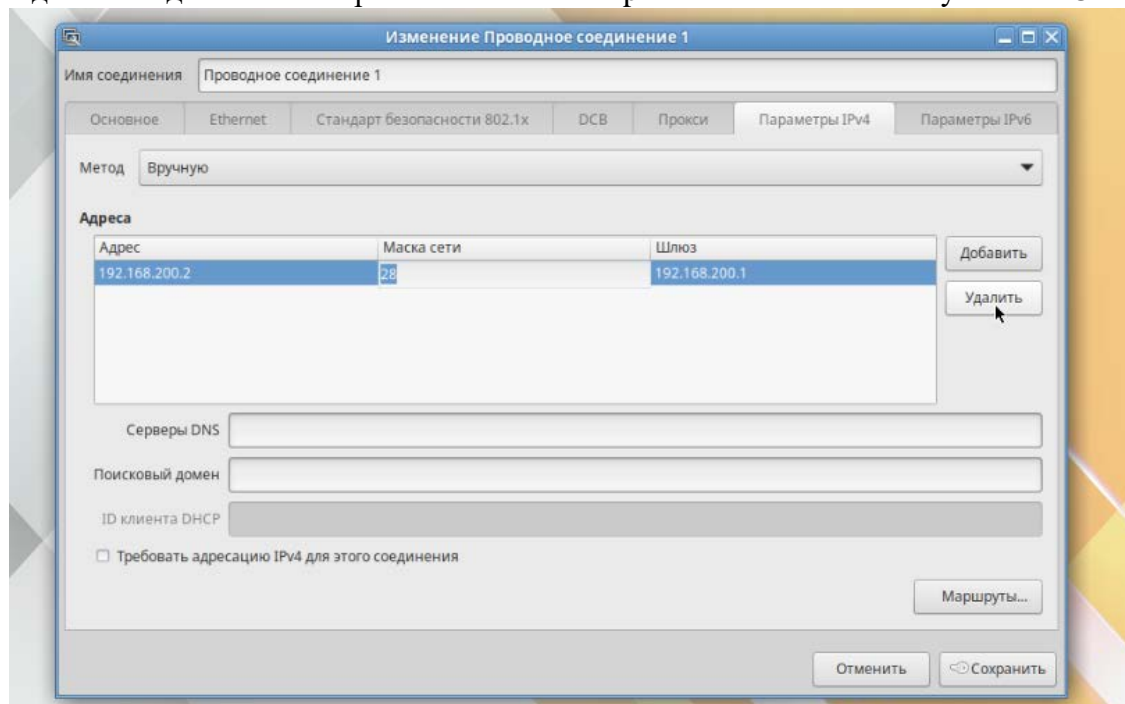
```
[root@hq-rtr ~]# systemctl enable --now dnsmasq.service
Synchronizing state of dnsmasq.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable dnsmasq
Created symlink /etc/systemd/system/multi-user.target.wants/dnsmasq.service → /lib/systemd/system/dnsmasq.service.
[root@hq-rtr ~]#
```

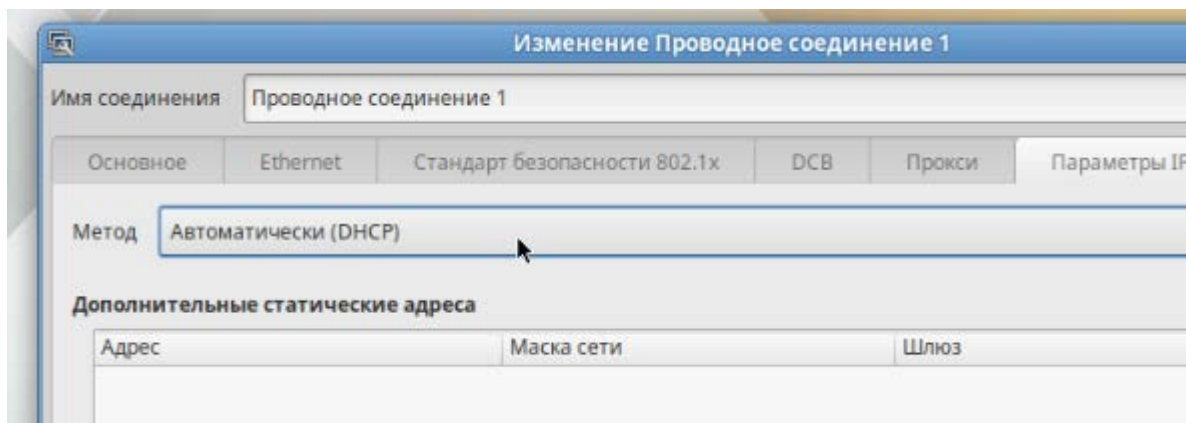
HQ-CLI:

Осталось проверить раздает ли ip-адрес на клиента HQ-CLI

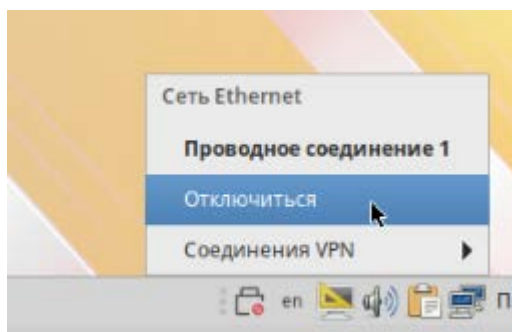
Зайдем на HQ-CLI и обновим конфигурацию сетевых параметров

Удалим созданный нами ранее статический ip и оставим только получать DHCP автоматически

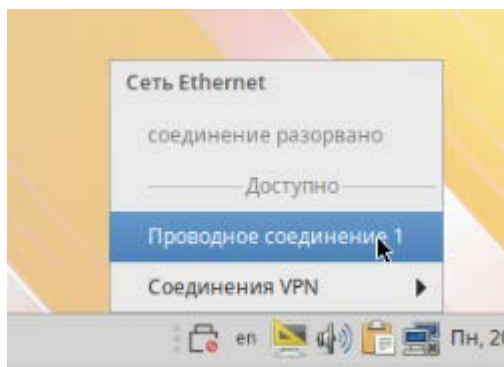




Нажмем Сохранить, вводим пароль тоог и затем переактивируем сетевое подключение (левой кнопкой мыши на апплете сетевого подключения => отключиться, затем то же самое и выбираем название проводного соединения



=>



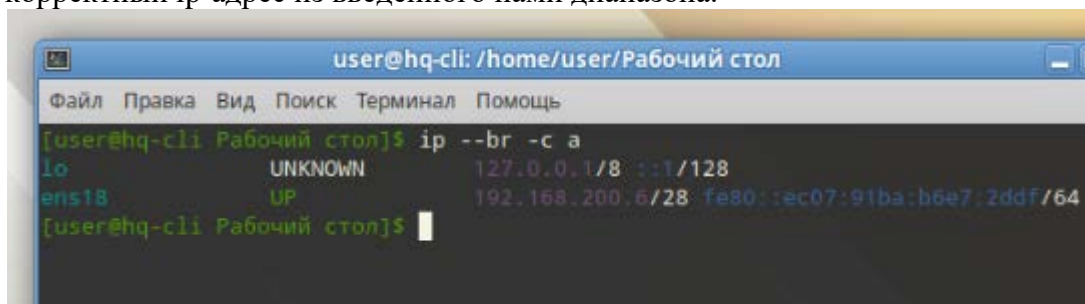
Получаем анимацию загрузки сетевых параметров и если все хорошо, то она сменяется обычным видом апплета сетевого подключения



=>



Проверим адрес, который получили, откроем терминал и командой `ip --br -c a` а удостоверимся, что выдан корректный ip-адрес из введенного нами диапазона.



Все окей, значит наш DHCP-сервер работает корректно.

10. НАСТРОЙКА DNS ДЛЯ ОФИСОВ HQ И BR:

Настройку DNS-сервера можно выполнять стандартными способами, такими как конфигурирование сервиса из пакета bind9, а можно использовать уже знакомый нам сервис **dnsmasq**.

HQ-SRV:

Первым делом проверим, что мы можем пропинговать сайт ya.ru с HQ-SRV. Если не выдает пинг, тогда нужно на интерфейс машины в файл `/etc/net/iface/ens18.100/resolv.conf` дописать строчку **nameserver 77.88.8.8** с общедоступным сервером dns от яндекса. Рестартанув сетевую службу мы получим возможность скачивать пакеты.

```
[root@hq-srv ~]# echo nameserver 77.88.8.8 > /etc/net/iface/ens18.100/resolv.conf
```

Проверим что файл записан в файл

```
[root@hq-srv ~]# cat /etc/net/iface/ens18.100/resolv.conf
nameserver 77.88.8.8
[root@hq-srv ~]#
```

Делаем рестарт сетевых сервисов

```
[root@hq-srv ~]# systemctl restart networkd
```

И пингуем ya.ru

```
[root@hq-srv ~]# ping ya.ru
PING ya.ru (213.180.193.56) 56(84) bytes of data:
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=238 time=10.1 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=2 ttl=238 time=10.2 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=3 ttl=238 time=18.9 ms
```

Все окей, можем идти дальше.

Для начала нам нужно отключить несовместимую с dnsmasq службу bind, чтобы не возникло конфликтов. Для этого на сервисе пропишем **systemctl disable --now bind**

```
[root@hq-srv ~]# systemctl disable --now bind
Failed to disable unit: Unit file bind.service does not exist.
[root@hq-srv ~]#
```

В нашем случае его нет, поэтому он ругается, но на всякий случай лучше проверить.

Теперь установим на сервер dnsmasq.

apt-get update

```
[root@hq-srv ~]# apt-get update
Get:1 http://ftp.altlinux.org p10/branch/x86_64 release [4223B]
Get:2 http://ftp.altlinux.org p10/branch/x86_64-i586 release [1665B]
Get:3 http://ftp.altlinux.org p10/branch/noarch release [2844B]
Fetched 8732B in 0s (222kB/s)
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic pkglist [24.4MB]
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic release [137B]
Get:3 http://ftp.altlinux.org p10/branch/x86_64/gostcrypto pkglist [18.5kB]
Get:4 http://ftp.altlinux.org p10/branch/x86_64/gostcrypto release [140B]
Get:5 http://ftp.altlinux.org p10/branch/x86_64-i586/classic pkglist [17.9MB]
Get:6 http://ftp.altlinux.org p10/branch/x86_64-i586/classic release [142B]
Get:7 http://ftp.altlinux.org p10/branch/noarch/classic pkglist [7290kB]
Get:8 http://ftp.altlinux.org p10/branch/noarch/classic release [137B]
Fetched 49.7MB in 5s (9905kB/s)
Reading Package Lists... Done
Building Dependency Tree... Done
[root@hq-srv ~]#
```

apt-get install dnsmasq

```
[root@hq-srv ~]# apt-get install dnsmasq
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  dnsmasq
0 upgraded, 1 newly installed, 0 removed and 147 not upgraded.
Need to get 360kB of archives.
After unpacking 834kB of additional disk space will be used.
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic dnsmasq 2.90-alt1:p10
Fetched 360kB in 0s (11.7MB/s)
Committing changes...
Preparing...
Updating / installing...
1: dnsmasq-2.90-alt1
Done.
[root@hq-srv ~]#
```

Добавим наш сервис dns-сервера будущего в автозагрузку системы, который в свою очередь инициализирует его первый запуск. Для этого используем нам известную команду **systemctl enable --now dnsmasq.service**

```
done.
[root@hq-srv ~]# systemctl enable --now dnsmasq.service
Synchronizing state of dnsmasq.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable dnsmasq
Created symlink /etc/systemd/system/multi-user.target.wants/dnsmasq.service → /lib/systemd/system/dnsmasq.service.
[root@hq-srv ~]#
```

Проверяем статус сервиса

```
[root@hq-srv ~]# systemctl status dnsmasq.service
dnsmasq.service - A lightweight DHCP and caching DNS server
Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2025-01-20 14:17:00 MSK; 35s ago
Process: 16308 ExecStartPost=/usr/sbin/dnsmasq-helper poststart (code=exited, status=0/SUCCESS)
Main PID: 16307 (dnsmasq)
Tasks: 1 (limit: 2339)
Memory: 332.0K
CPU: 11ms
CGroup: /system.slice/dnsmasq.service
└─ 16307 /usr/sbin/dnsmasq --bind-interfaces --interface lo -s ks54.net -u _dnsmasq

Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: started, version 2.90 cachesize 150
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: compile time options: IPv6 GNU-getopt no-DBus
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: reading /etc/resolv.conf
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: using nameserver 77.88.8.8#53
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: read /etc/hosts - 6 names
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: reading /etc/resolv.conf
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq[16307]: ignoring nameserver 127.0.0.1 - local interface
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq-helper[16434]: Setup resolv.conf for local resolver: s
Jan 20 14:17:00 hq-srv.ks54.net dnsmasq-helper[16308]: Setup resolv.conf for local resolver: [
Jan 20 14:17:00 hq-srv.ks54.net systemd[1]: Started A lightweight DHCP and caching DNS server.
lines 1-21/21 (END)
```

Открываем файл для редактирования конфигурации нашего будущего DNS-сервера:

mcedit /etc/dnsmasq.conf

```
[root@hq-srv ~]# mcedit /etc/dnsmasq.conf
```

И добавляем в неё строки (для удобства прям с первой строки файла):

no-resolv (не будет использовать /etc/resolv.conf)

domain=ks54.net

server=77.88.8.8 (адрес общедоступного DNS-сервера)

interface=ens18.100 (на каком интерфейсе будет работать служба)

address=/hq-rtr.ks54.net/192.168.100.1

ptr-record=1.100.168.192.in-addr.arpa,hq-rtr.ks54.net

cname=moodle.ks54.net,hq-rtr.ks54.net (Запись, которая понадобится во 2 модуле для работы Moodle)

cname=wiki.ks54.net,hq-rtr.ks54.net (Запись, которая понадобится во 2 модуле для работы Wiki)

address=/br-rtr.ks54.net/100.64.200.1

address=/hq-srv.ks54.net/192.168.100.2

ptr-record=2.100.168.192.in-addr.arpa,hq-srv.ks54.net

address=/hq-cli.au-team.irpo/192.168.200.6 (Смотрите адрес на **HQ-CLI**, т.к он выдаётся по DHCP)

ptr-record=6.2.168.192.in-addr.arpa,hq-cli.ks54.net

address=/br-srv.ks54.net/100.64.200.2

```

dnsmasq.conf      [-M--] 15 L:[ 1+ 2 3/714] *(59 /2)
# Configuration file for dnsmasq.
no-resolv
domain=ks54.net
server=77.88.8.8
interface=ens18.100

address=/hq-rtr.ks54.net/192.168.100.1
ptr-record=1.100.168.192.in-addr.arpa,hq-rtr.ks54.net
cname=moodle.ks54.net,hq-rtr.ks54.net
cname=wiki.ks54.net,hq-rtr.ks54.net

address=/br-rtr.ks54.net/100.64.200.1

address=/hq-srv.ks54.net/192.168.100.2
ptr-record=2.100.168.192.in-addr.arpa,hq-srv.ks54.net

address=/hq-cli.ks54.net/192.168.200.6
ptr-record=6.100.168.192.in-addr.arpa,hq-cli.ks54.net

address=/br-srv.ks54.net/100.64.200.2

```

Теперь добавим в файл `/etc/hosts` строчку **192.168.100.1 hq-rtr.ks54.net**, чтобы система могла интерпретировать роутер hq-rtr по доменному имени и по ip-адресу.

mcedit /etc/hosts

```

[root@hq-srv ~]# mcedit /etc/hosts
hosts      [-M--] 31 L:[ 1+ 0 1/ 4] *
192.168.100.1<->hq-rtr.ks54.net
127.0.0.1<----->localhost.localdomain localhost
::1<--->localhost6.localdomain localhost6

```

Перезапустим сервис dnsmasq

systemctl restart dnsmasq.service

```

[root@hq-srv ~]# systemctl restart dnsmasq.service
[root@hq-srv ~]#

```

Проверяем статус сервиса и убедимся, что он работает без ошибок. (если ошибки есть, внимательно читаем мануал, смотрим и сверяем конфигурационные файлы, а также читаем журнал ошибок в системе)

```

[root@hq-srv ~]# systemctl status dnsmasq.service
dnsmasq.service - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2025-01-20 16:47:45 MSK; 14s ago
     Process: 17590 ExecStartPost=/usr/sbin/dnsmasq-helper poststart (code=exited, status=0/SUCCESS)
    Main PID: 17589 (dnsmasq)
       Tasks: 1 (limit: 2339)
      Memory: 332.0K
         CPU: 121ms
    CGroup: /system.slice/dnsmasq.service
            └─ 17589 /usr/sbin/dnsmasq --bind-interfaces --interface lo -s ks54.net -u _dnsmasq -k --pid-file

Jan 20 16:47:45 hq-srv.ks54.net systemd[1]: Starting A lightweight DHCP and caching DNS server...
Jan 20 16:47:45 hq-srv.ks54.net dnsmasq[17589]: started, version 2.90 cachesize 150
Jan 20 16:47:45 hq-srv.ks54.net dnsmasq[17589]: compile time options: IPv6 GNU-getopt no-DBus no-UBus no-i18n II
Jan 20 16:47:45 hq-srv.ks54.net dnsmasq[17589]: using nameserver 77.88.8.8#53
Jan 20 16:47:45 hq-srv.ks54.net dnsmasq[17589]: read /etc/hosts - 7 names
Jan 20 16:47:45 hq-srv.ks54.net dnsmasq-helper[17590]: Setup resolv.conf for local resolver:[ DONE ]
Jan 20 16:47:45 hq-srv.ks54.net systemd[1]: Started A lightweight DHCP and caching DNS server.
lines 1-18/18 (END)

```

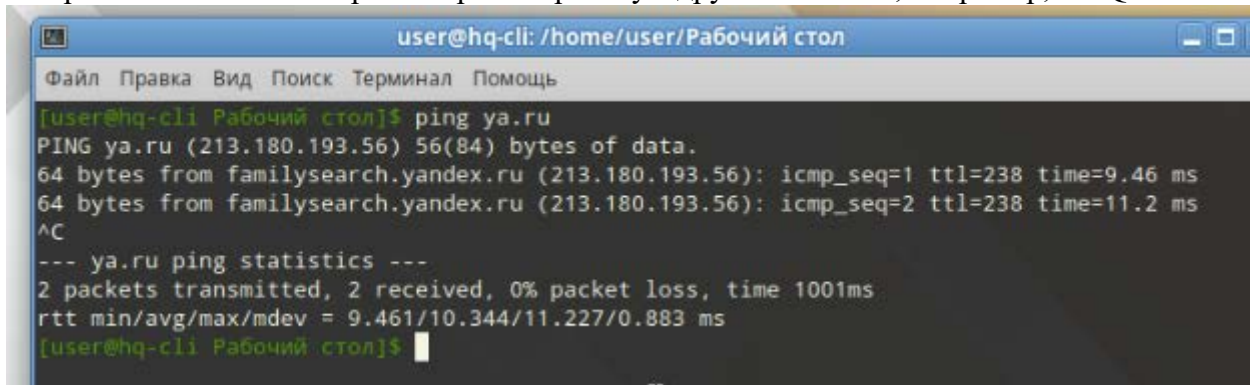
Проверим пинг до яндекса по доменному имени (или гугла, кому что больше нравится)


```
[root@hq-srv ~]# ping ya.ru
PING ya.ru (213.180.193.56) 56(84) bytes of data.
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=238 time=11.0 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=2 ttl=238 time=11.0 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=3 ttl=238 time=10.6 ms
^C
--- ya.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 10.617/10.858/10.995/0.171 ms
[root@hq-srv ~]#
```

Проверим локальную dns-запись на доступность по доменному имени и если пинг идет, то тогда dns-сервер работает.

```
[root@hq-srv ~]# ping hq-rtr.ks54.net
PING hq-rtr.ks54.net (192.168.100.1) 56(84) bytes of data.
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=1 ttl=64 time=0.858 ms
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=2 ttl=64 time=0.406 ms
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=3 ttl=64 time=0.322 ms
^C
--- hq-rtr.ks54.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.322/0.528/0.858/0.235 ms
[root@hq-srv ~]#
```

Все работает. Осталось протестировать работу с другой машины, например, с HQ-CLI



```
user@hq-cli: /home/user/Рабочий стол
Файл Правка Вид Поиск Терминал Помощь
[user@hq-cli Рабочий стол]$ ping ya.ru
PING ya.ru (213.180.193.56) 56(84) bytes of data.
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=1 ttl=238 time=9.46 ms
64 bytes from familysearch.yandex.ru (213.180.193.56): icmp_seq=2 ttl=238 time=11.2 ms
^C
--- ya.ru ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 9.461/10.344/11.227/0.883 ms
[user@hq-cli Рабочий стол]$
```

Проверим другие записи:

```
[user@hq-cli Рабочий стол]$ ping hq-rtr.ks54.net
PING hq-rtr.ks54.net (192.168.100.1) 56(84) bytes of data.
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=2 ttl=64 time=0.550 ms
64 bytes from hq-rtr.ks54.net (192.168.100.1): icmp_seq=3 ttl=64 time=0.843 ms
^C
--- hq-rtr.ks54.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.273/0.555/0.843/0.232 ms
[user@hq-cli Рабочий стол]$
```

Проверим CNAME записи с помощью команды **dig**

В выводе команды в разделе ANSWER SECTION должны увидеть наши записи, что мы создавали.

```

tcc min/avg/max/mdev = 0.27370.33370.84370.232 ms
[user@hq-cli Рабочий стол]$ dig moodle.ks54.net

; <<>> DiG 9.16.48 <<>> moodle.ks54.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53693
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:: udp: 1232
;; QUESTION SECTION:
;moodle.ks54.net.                IN      A

;; ANSWER SECTION:
moodle.ks54.net.                0       IN      CNAME   hq-rtr.ks54.net.
hq-rtr.ks54.net.                0       IN      A       192.168.100.1

;; Query time: 1 msec
;; SERVER: 192.168.100.2#53(192.168.100.2)
;; WHEN: Mon Jan 20 16:58:58 MSK 2025
;; MSG SIZE rcvd: 89

[user@hq-cli Рабочий стол]$

```

Wiki.ks54.net

```

[user@hq-cli Рабочий стол]$ dig wiki.ks54.net

; <<>> DiG 9.16.48 <<>> wiki.ks54.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62824
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:: udp: 1232
;; QUESTION SECTION:
;wiki.ks54.net.                IN      A

;; ANSWER SECTION:
wiki.ks54.net.                0       IN      CNAME   hq-rtr.ks54.net.
hq-rtr.ks54.net.                0       IN      A       192.168.100.1

;; Query time: 1 msec
;; SERVER: 192.168.100.2#53(192.168.100.2)
;; WHEN: Mon Jan 20 17:00:31 MSK 2025
;; MSG SIZE rcvd: 87

```

Все работает. DNS-сервер готов.

3. СОЗДАНИЕ ЛОКАНЫХ УЧЕТНЫХ ЗАПИСЕЙ:

Создадим пользователей sshuser на серверах подстетей HQ и BR.

Создание на HQ-SRV:

Для создания пользователя с заданным идентификатором (как сказано в задании – см. задание Модуль 1) на машине под управлением ОС ALT Linux используем команду **useradd sshuser -u 1010**

```
[root@hq-srv ~]# useradd sshuser -u 1010
```

Удостовериться в правильности создания пользователя с заданным id можно командой **id sshuser**

```
[root@hq-srv ~]# id sshuser
uid=1010(sshuser) gid=1010(sshuser) groups=1010(sshuser)
[root@hq-srv ~]#
```

Зададим пароль на нашего пользователя, используя команду **passwd sshuser** и вводим пароль P@ssw0rd и еще раз для подтверждения.

```
[root@hq-srv ~]# passwd sshuser
passwd: updating all authentication tokens for user sshuser.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters. You can use a password containing at least 7 characters
from all of these classes, or a password containing at least 8 characters
from just 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 11 to 72 characters long, and
contain enough different characters.

Alternatively, if no one else can see your terminal now, you can pick this as
your password: "exist&Code3avert".

Enter new password:
Weak password: based on a dictionary word and not a passphrase.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@hq-srv ~]#
```

Чтобы **sshuser** мог запускать **sudo** без дополнительной аутентификации, необходимо убрать комментарий строки в файле **/etc/sudoers**, откроем его командой:

mcedit /etc/sudoers

```
[root@hq-srv ~]# mcedit /etc/sudoers
```

И уберём комментарий (знак #) на следующей строке:

WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL

```
## Uncomment to allow members of group wheel to execute any command
# WHEEL_USERS ALL=(ALL:ALL) ALL

## Same thing without a password
WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# SUDO_USERS<-->ALL=(ALL:ALL) ALL
```

Сохраняем и затем добавляем нашего пользователя **sshuser** в группу **wheel**. Для этого используем команду **usermod -aG wheel sshuser**

```
[root@hq-srv ~]# usermod -aG wheel sshuser
[root@hq-srv ~]#
```

Проверим теперь нашего пользователя командами и удостоверимся, что все сделали правильно: выставили **id** и добавили в группу **wheel**

```
[root@hq-srv ~]# id sshuser
uid=1010(sshuser) gid=1010(sshuser) groups=1010(sshuser),10(wheel)
[root@hq-srv ~]#
```

Группы можно проверить более явно следующей командой

```
[root@hq-srv ~]# groups sshuser
sshuser : sshuser wheel
```

Создание на BR-SRV:

Проделаем тоже самое для сервера **BR-SRV**. Создадим такого же пользователя с теми же параметрами, что и для **HQ-SRV** (более подробные пояснения можно посмотреть выше)

```
[root@br-srv ens18]# useradd sshuser -u 1010
[root@br-srv ens18]#
[root@br-srv ens18]# id sshuser
uid=1010(sshuser) gid=1010(sshuser) groups=1010(sshuser)
```



```

[root@br-srv ens18]# passwd sshuser
passwd: updating all authentication tokens for user sshuser.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters. You can use a password containing at least 7 characters
from all of these classes, or a password containing at least 8 characters
from just 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 11 to 72 characters long, and
contain enough different characters.

Alternatively, if no one else can see your terminal now, you can pick this as
your password: "Mussel7scent+Signal".

Enter new password:
Weak password: based on a dictionary word and not a passphrase.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@br-srv ens18]#
[root@br-srv ens18]# mcedit /etc/sudoers

```

```

## Uncomment to allow members of group wheel to execute any command
# WHEEL_USERS ALL=(ALL:ALL) ALL

## Same thing without a password
WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# SUDO_USERS<-->ALL=(ALL:ALL) ALL

```

```

[root@br-srv ens18]# usermod -aG wheel sshuser
[root@br-srv ens18]#

```

Проверим аналогичными командами

```

[root@br-srv ens18]# id sshuser
uid=1010(sshuser) gid=1010(sshuser) groups=1010(sshuser),10(wheel)
[root@br-srv ens18]# groups sshuser
sshuser : sshuser wheel
[root@br-srv ens18]#

```

Пользователей для ssh мы создали, теперь создадим пользователей net_admin для наших роутеров

Настройка пользователя net_admin на HQ-RTR:

В целом добавление и настройка пользователя похожа на предыдущий пункт, но есть некоторые особенности согласно заданию.

Сперва добавляем пользователя с домашним каталогом командой **useradd net_admin -m**

```

[root@hq-rtr ~]# useradd net_admin -m

```

Ставим пароль на него командой **passwd net_admin** и вводим дважды пароль P@ssw0rd

```
[root@hq-rtr ~]# passwd net_admin
passwd: updating all authentication tokens for user net_admin.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters. You can use a password containing at least 7 characters
from all of these classes, or a password containing at least 8 characters
from just 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 11 to 72 characters long, and
contain enough different characters.

Alternatively, if no one else can see your terminal now, you can pick this as
your password: "peril*Living&Hung".

Enter new password:
Weak password: based on a dictionary word and not a passphrase.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@hq-rtr ~]#
```

По заданию необходимо, чтобы net_admin мог запускать команду sudo без дополнительной аутентификации (то есть без запроса пароля), необходимо отредактировать файл /etc/sudoers, а именно добавить в конец файла строку **net_admin ALL=(ALL:ALL) NOPASSWD: ALL**

```
[root@hq-rtr ~]# mcedit /etc/sudoers
```

И в конце файла пропишем строчку

```
## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d
net_admin ALL=(ALL:ALL) NOPASSWD: ALL
1Help      2Save      3Mark
```

Настройка пользователя net_admin на BR-RTR:

Аналогичным способом создаем пользователя и на втором роутере, более подробные комментарии смотри в пункте выше.

Создаем пользователя net_admin

```
[root@br-rtr ~]# useradd net_admin -m
```

Устанавливаем пароль P@ssw0rd

```
[root@br-rtr ~]# passwd net_admin
passwd: updating all authentication tokens for user net_admin.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters. You can use a password containing at least 7 characters
from all of these classes, or a password containing at least 8 characters
from just 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 11 to 72 characters long, and
contain enough different characters.

Alternatively, if no one else can see your terminal now, you can pick this as
your password: "Key&Her*topple".

Enter new password:
Weak password: based on a dictionary word and not a passphrase.
Re-type new password:
passwd: all authentication tokens updated successfully.
[root@br-rtr ~]#
```

Редактируем файл /etc/sudoers

```
[root@hq-rtr ~]# mcedit /etc/sudoers
```

```
## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d
net_admin ALL=(ALL:ALL) NOPASSWD: ALL
1Help      2Save      3Mark
```

На этом создание локальных учетных записей пользователей по заданию 1 модуля завершено.

5. НАСТРОЙКА БЕЗОПАСНОГО УДАЛЕННОГО ДОСТУПА НА СЕРВЕРАХ HQ-SRV И BR-SRV:

Выполним настройку по заданию сперва на машине HQ-SRV:

Для работы SSH нам понадобится служба **openssh-common**, которой изначально нет, поэтому установим её: **apt-get install openssh-common**

```
[root@hq-srv ~]# apt-get install openssh-common
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  openssh-clients openssh-server openssh-server-control
The following packages will be upgraded:
  openssh-clients openssh-common openssh-server openssh-server-control
4 upgraded, 0 newly installed, 0 removed and 143 not upgraded.
Need to get 1164kB of archives.
After unpacking 0B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-clients 7.9p1-alt4.p10
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-common 7.9p1-alt4.p10
Get:3 http://ftp.altlinux.org p10/branch/noarch/classic openssh-server-control 7.9p1-
Get:4 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-server 7.9p1-alt4.p10
Fetched 1164kB in 0s (14.6MB/s)
Committing changes...
Preparing...
Updating / installing...
1: openssh-common-7.9p1-alt4.p10.6
2: openssh-server-control-7.9p1-alt4.p10.6
3: openssh-server-7.9p1-alt4.p10.6
Warning: The unit file, source configuration file or drop-ins of sshd.service changed.
4: openssh-clients-7.9p1-alt4.p10.6
Cleaning up / removing...
5: openssh-server-7.9p1-alt4.p10.4
6: openssh-server-control-7.9p1-alt4.p10.4
7: openssh-clients-7.9p1-alt4.p10.4
8: openssh-common-7.9p1-alt4.p10.4
Done.
[root@hq-srv ~]#
```

Далее вносим изменения в конфигурационный файл openssh командой **mcedit /etc/openssh/sshd_config**

```
[root@hq-srv ~]# mcedit /etc/openssh/sshd_config
```

(внимательно пишите файл конфигурации, так как там есть в директории два файла, один **ssh_config**, а второй **sshd_config**, так вот нам нужен именно второй с буквой d)

Вбиваем в этот конфигурационный файл следующие строки:

Port 2024 (меняем порт подключения по ssh со стандартного 22 на 2024)
MaxAuthTries 2 (выставляем ограничение попыток входа равное двум)
AllowUsers sshuser (разрешаем подключение только пользователю sshuser)
PermitRootLogin no (запрещаем вход по ssh от имени root)

```

sshd_config [-M--] 0 L:[ 1
#<----->$OpenBSD: sshd_config,v 1.1
# This is the sshd server system-wide
# sshd_config(5) for more informati
# This sshd was compiled with PATH=
# The strategy used for options in
# OpenSSH is to specify options with
# possible, but leave them commented
# default value.

Port 2024
MaxAuthTries 2
AllowUsers sshuser
PermitRootLogin no

#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

```

Также по заданию нам нужен баннер. Для этого надо нам создать его. Создаем командой файл **mcedit /root/banner**

```
[root@hq-srv ~]# mcedit /root/banner
```

Внутри пишем **Authorized access only** и обязательно после этой строчки нажимаем Enter чтобы создалась пустая строка после введенной строки. Это обязательно, иначе баннерная фраза не считается системой и не будет работать.

```

banner [-M--] 0 L:[
Authorized access only
-

```

После создания баннера нам нужно сделать ссылку на наш созданный файл в конфигурационном файле openssh. Поэтому обратно открываем файл **mcedit /etc/openssh/sshd_config** и добавляем в конце файла строчку **Banner /root/banner**

```

#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none
Banner /root/banner

# override default of no subsystems
Subsystem<----->sftp<-->/usr/lib/openssh.
#AllowGroups wheel users

```

После внесения изменений, сохраняем и выходим.

Перезапускаем службу командой **systemctl restart sshd.service**

```
[root@hq-srv ~]# systemctl restart sshd.service
```

И делаем автозапуск службы после перезагрузки системы: **systemctl enable --now sshd**

```

[root@hq-srv ~]# systemctl enable --now sshd.service
Synchronizing state of sshd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable sshd
[root@hq-srv ~]#

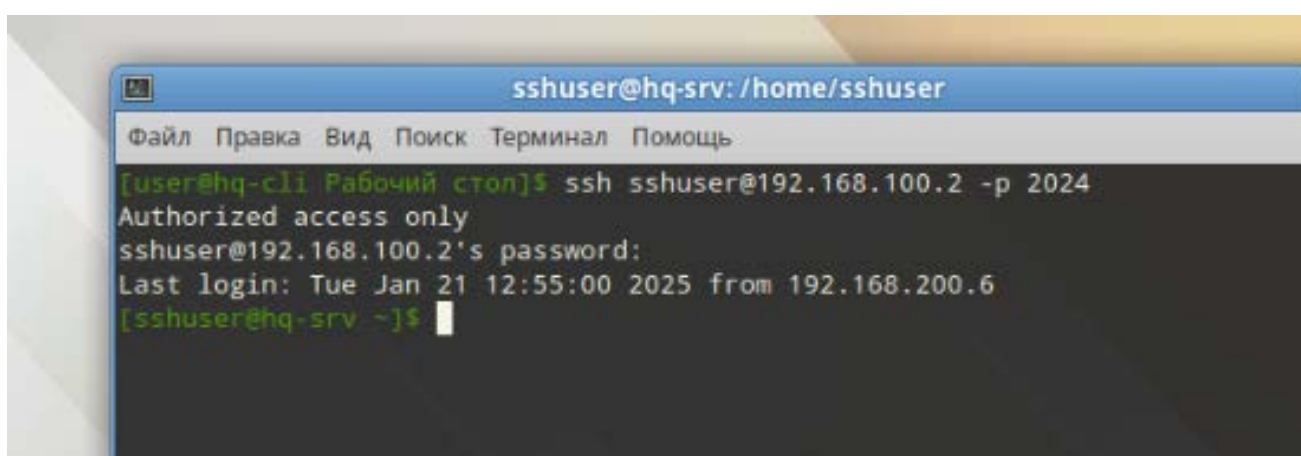
```

Проверим на сервере работу сервиса ssh

```
[root@hq-srv ~]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
Loaded: loaded (/lib/systemd/systemd/ssh.service: enabled; vendor preset: enabled)
Active: active (running) since Tue 2025-01-21 12:53:21 MSK; 5s ago
Process: 3251 ExecStartPre=/usr/bin/ssh-keygen -A (code=exited, status=0/SUCCESS)
Process: 3253 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 3254 (sshd)
Tasks: 1 (limit: 2339)
Memory: 740.0K
CPU: 8ms
CGroup: /system.slice/sshd.service
└─ 3254 /usr/sbin/sshd -D

Jan 21 12:53:21 hq-srv.ks54.net systemd[1]: Starting OpenSSH server daemon...
Jan 21 12:53:21 hq-srv.ks54.net systemd[1]: Started OpenSSH server daemon.
Jan 21 12:53:21 hq-srv.ks54.net sshd[3254]: Server listening on 0.0.0.0 port 2024.
Jan 21 12:53:21 hq-srv.ks54.net sshd[3254]: Server listening on :: port 2024.
[root@hq-srv ~]#
```

Теперь проверим работу нашего сервера через HQ-CLI. Попробуем подключиться из терминала к нашему серверу `ssh sshuser@192.168.100.2 -p 2024`



sshuser – пользователь, под которым вы подключаетесь

192.168.100.2 – адрес сервера, к которому мы подключаемся (**HQ-SRV**)

-p 2024 – порт, по которому мы подключаемся (мы заменили со стандартного 22 на 2024)

Выполним настройку ssh на машине **BR-SRV**:

Продублируем абсолютно все тоже самое и на сервере BR-SRV.


```

[root@br-srv ens18]# apt-get install openssh-common
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  openssh-clients openssh-server openssh-server-control
The following packages will be upgraded:
  openssh-clients openssh-common openssh-server openssh-server-control
4 upgraded, 0 newly installed, 0 removed and 143 not upgraded
Need to get 1164kB of archives.
After unpacking 0B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-clients 7.9p1-alt4.p10.6
Get:2 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-server 7.9p1-alt4.p10.6
Get:3 http://ftp.altlinux.org p10/branch/noarch/classic openssh-server-control 7.9p1-alt4.p10.6
Get:4 http://ftp.altlinux.org p10/branch/x86_64/classic openssh-common 7.9p1-alt4.p10.6
Fetched 1164kB in 0s (18.0MB/s)
Committing changes...
Preparing...
Updating / installing...
1: openssh-common-7.9p1-alt4.p10.6
2: openssh-server-control-7.9p1-alt4.p10.6
3: openssh-server-7.9p1-alt4.p10.6
Warning: The unit file, source configuration file or drop-in
4: openssh-clients-7.9p1-alt4.p10.6
Cleaning up / removing...
5: openssh-server-7.9p1-alt4.p10.4
6: openssh-server-control-7.9p1-alt4.p10.4
7: openssh-clients-7.9p1-alt4.p10.4
8: openssh-common-7.9p1-alt4.p10.4
Done.
[root@br-srv ens18]#

```

```

[root@br-srv ens18]# mcedit /etc/openssh/sshd_config

```

```

sshd_config  [-M--] 0 L:[ 1+16 17/131] *
#<----->$OpenBSD: sshd_config,v 1.103 2018/04/09 2
# This is the sshd server system-wide configuration file.
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/bin:/usr/bin:/usr/sbin
# The strategy used for options in the default sshd_config(5)
# OpenSSH is to specify options with their default values,
# possible, but leave them commented. Uncommented options
# default value.
Port 2024
MaxAuthTries 2
AllowUsers sshuser
PermitRootLogin no
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/openssh/ssh_host_rsa_key
#HostKey /etc/openssh/ssh_host_ecdsa_key

```

```

[root@br-srv ens18]# mcedit /root/banner

```

```

banner  [-M--] 0 L:
Authorized access only

```



```
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none
Banner /root/banner_

# override default of no subsystem
Subsystem<----->sftp<-->/usr/lib/

#AllowGroups wheel users
```

```
[root@br-srv ens18]# systemctl restart sshd.service
```

```
[root@br-srv ens18]# systemctl enable --now sshd.service
Synchronizing state of sshd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable sshd
[root@br-srv ens18]#
```

```
[root@br-srv ens18]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
  Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2025-01-21 14:00:22 MSK; 55s ago
    Main PID: 29695 (sshd)
      Tasks: 1 (limit: 2339)
     Memory: 748.0K
        CPU: 9ms
      CGroup: /system.slice/sshd.service
              └─ 29695 /usr/sbin/sshd -D

Jan 21 14:00:22 br-srv.ks54.ru systemd[1]: Starting OpenSSH server daemon...
Jan 21 14:00:22 br-srv.ks54.ru systemd[1]: Started OpenSSH server daemon.
Jan 21 14:00:22 br-srv.ks54.ru sshd[29695]: Server listening on 0.0.0.0 port 2024.
Jan 21 14:00:22 br-srv.ks54.ru sshd[29695]: Server listening on :: port 2024.
[root@br-srv ens18]#
```

```
[user@hq-cl1 Рабочий стол]$ ssh sshuser@100.64.200.2 -p 2024
The authenticity of host '[100.64.200.2]:2024 ([100.64.200.2]:2024)' can't be es
tablished.
ED25519 key fingerprint is SHA256:39fy0390LfU/b/BI1M6R+fsRyDuL3HozYTnBMyI8sJg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[100.64.200.2]:2024' (ED25519) to the list of known
hosts.
Authorized access only
sshuser@100.64.200.2's password:
[sshuser@br-srv ~]$
```

Результат показывает, что сервер ssh работает на обоих серверах по заданным параметрам

ВСЕ ЗАДАНИЯ 1 МОДУЛЯ ДА ВЫПОЛНЕННЫ