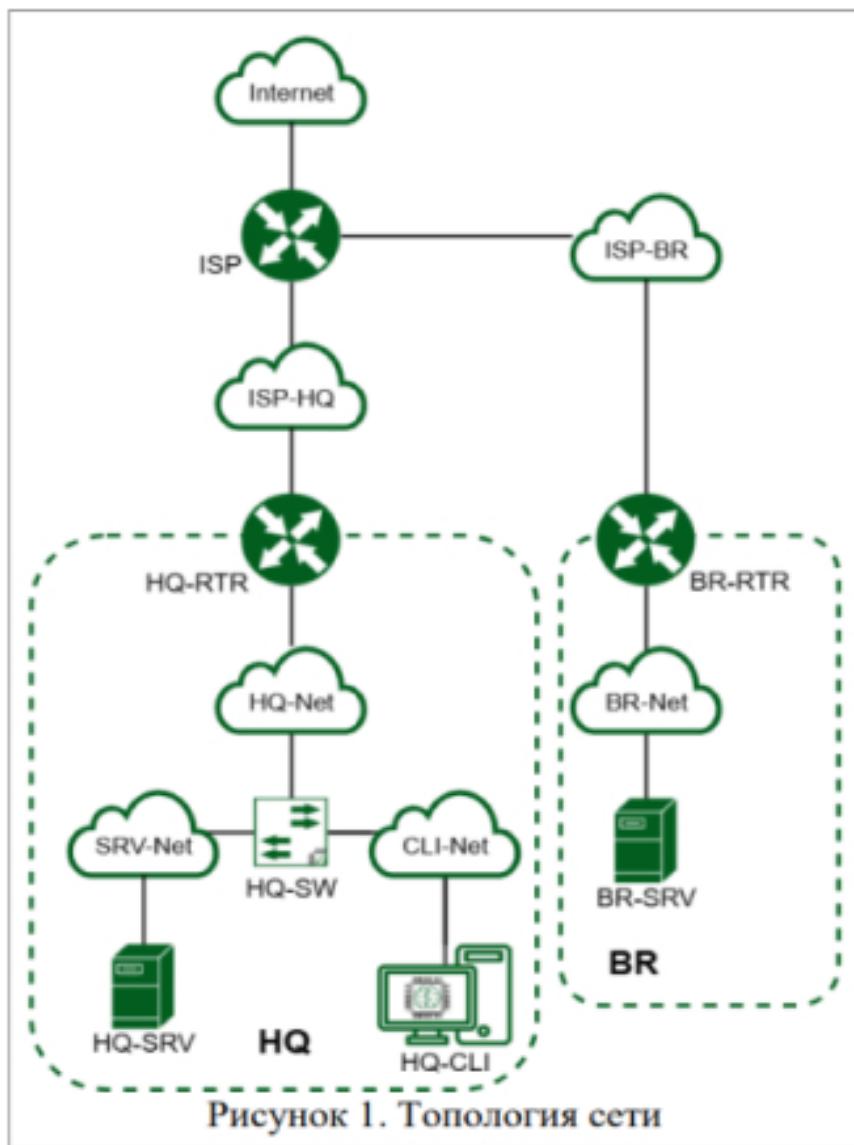


Модуль 1



1. Произведите базовую настройку устройств

- Настройте имена устройств согласно топологии. Используйте полное доменное имя
- На всех устройствах необходимо сконфигурировать IPv4
- IP-адрес должен быть из приватного диапазона, в случае, если сеть локальная, согласно RFC1918
- Локальная сеть в сторону HQ-SRV(VLAN100) должна вмещать не более 64 адресов
- Локальная сеть в сторону HQ-CLI(VLAN200) должна вмещать не более 16 адресов
- Локальная сеть в сторону BR-SRV должна вмещать не более 32 адресов

- g) Локальная сеть для управления (VLAN999) должна вмещать не более 8 адресов
- h) Сведения об адресах занесите в отчёт, в качестве примера используйте Таблицу 3

Настройка имен

Полное доменное имя можно посмотреть в таблице из задания:

Устройство	Имя
HQ-RTR	hq-rtr.au-team.irpo
BR-RTR	br-rtr.au-team.irpo
HQ-SRV	hq-srv.au-team.irpo
HQ-CLI	hq-cli.au-team.irpo
BR-SRV	br-srv.au-team.irpo
HQ-RTR	moodle.au-team.irpo
BR-RTR	wiki.au-team.irpo

Для изменения имени ALT используется команда:

```
hostnamectl set-hostname <FQDN>; exec bash, где  
FQDN (Fully Qualified Domain Name) - полное доменное имя  
exec bash - обновление оболочки
```

```
[root@isp ~]# hostnamectl set-hostname isp.au-team.irpo;exec bash
```

Настройка имен устройств на EcoRouter

Переходим в режим конфигурации и прописываем следующее:

```
hostname <name>  
<name> - желаемое имя устройства
```

```
ecorouter>enable  
ecorouter#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
ecorouter(config)#hostname hq-rtr  
hq-rtr(config)#ip domain-name au-team.irpo  
hq-rtr(config)#write memo  
hq-rtr(config)#write memory  
Building configuration...
```

Конфигурация IPv4

Используем следующую таблицу адресации (расчет можно произвести по IP калькулятору в зависимости от требуемого количества устройств в той или иной сети либо в VLAN)

Таблица подсетей

SRV-Net (VLAN 100)	192.168.100.0/26	192.168.100.1 - 192.168.100.62
CLI-Net (VLAN 200)	192.168.100.64/28	192.168.100.65 - 192.168.100.78
MGMT (VLAN 999)	192.168.100.80/29	192.168.100.81 - 192.168.100.86
BR-Net	192.168.3.0/27	192.168.3.1 - 192.168.3.30
ISP-HQ	172.16.4.0/28	172.16.4.1 - 172.16.4.14
ISP-BR	172.16.5.0/28	172.16.5.1 - 172.16.5.14

Таблица адресации

Имя устройства	Интерфейс	IPv4/IPv6	Маска/Префикс	Шлюз
ISP	ens18	192.168.0.130 (DHCP)	/24	10.12.28.254
	ens19	172.16.4.1	/28	
	ens20	172.16.5.1	/28	
HQ-RTR	isp	172.16.4.2	/28	172.16.4.1
	vl100	192.168.100.1	/26	
	vl200	192.168.100.65	/28	
	vl999	192.168.100.81	/29	
BR-RTR	isp	172.16.5.2	/28	172.16.5.1
	int1	192.168.3.1	/27	
HQ-SRV	ens19	192.168.100.62	/26	192.168.100.1
BR-SRV	ens19	192.168.3.30	/27	192.168.3.1

HQ-CLI	ens19	DHCP		
--------	-------	------	--	--

Выполняем на ISP

Смотрим имена интерфейсов командой

Ip -br -c a

```
[root@isp ifaces]# ip -br -c a
lo          UNKNOWN      127.0.0.1/8  ::1/128
ens19        DOWN
ens20        DOWN
ens21        DOWN
```

Перейдем в директорию с сетевыми конфигурациями:

cd /etc/net/ifaces

и просмотрим содержимое этой директории

ls

```
[root@isp ifaces]# ls
default  ens18  lo  unknown
```

Видим, что названия наших интерфейсов не совпадают.

Сделаем так, чтобы совпадали:

mv ens18 ens19

теперь у нас есть директория ens19 с файлом options

Его содержимое посмотрим командой

```
[root@isp ifaces]# cat ens19/options
BOOTPROTO=dhcp
TYPE=eth
```

Запись dhcp – получение этим интерфейсом сетевых настроек автоматически.

Создадим директорию для следующего интерфейса ens20 (или скопируем ens19)

```
[root@isp ifaces]# cp -r ens19 ens20
```

-r – директория скопируется со всем содержимым

Отредактируем файл options, так как у нас будет статический адрес на этом интерфейсе.

```
[root@isp ifaces]# vim ens20/options
BOOTPROTO=static
TYPE=eth
```

Для того чтобы начать писать текст в vim – нажать I, чтобы перестать его вводить – нажать Esc. Для сохранения и выхода ввести :wq и нажать Enter.

Создаем внутри ens20 файл ipv4address, где пропишем ip адрес для этого интерфейса из таблицы.

```
[root@isp ifaces]# echo 172.16.4.1/28 > /etc/net/ifaces/ens20/ipv4address
```

Проверим

```
[root@isp ifaces]# cat ens20/ipv4address  
172.16.4.1/28
```

Скопируем ens20 и все содержимое в ens21

```
[root@isp ifaces]# cp -r ens20 ens21
```

Изменим файл ipv4address, где пропишем ip адрес уже для этого интерфейса

```
[root@isp ifaces]# echo 172.16.5.1/28 > /etc/net/ifaces/ens21/ipv4address
```

Проверим

```
[root@isp ifaces]# cat ens21/ipv4address  
172.16.5.1/28
```

В итоге у нас должно быть 3 директории с именами наших интерфейсов

Для интерфейса, который смотрит во внешнюю сеть, должен быть прописан параметр dhcp в файле options, для других интерфейсов static , так как они смотрят во внутренние сети (каждый в свою). Для интерфейсов, которые смотрят во внутренние сети должен быть прописан ip адрес с префиксом подсети в файле ipv4address.

Перезапустим сетевую службу

```
[root@isp ifaces]# systemctl restart network
```

Просмотрим ip адреса

```
[root@isp ifaces]# ip -br -c a  
lo          UNKNOWN      127.0.0.1/8  ::1/128  
ens19        UP          192.168.0.130/24 fe80::c878:15ff:fe4:58cc/64  
ens20        UP          172.16.4.1/28 fe80::c4ae:53ff:fea7:5b7e/64  
ens21        UP          172.16.5.1/28 fe80::b06f:18ff:fe1b:5d12/64
```

Практически также настраиваются ip адреса на HQ-SRV и BR-SRV, кроме того, что необходимо добавить файл ipv4route с адресом шлюза по умолчанию (интерфейс маршрутизатора).

Перед настройкой обратим внимание на MAC адреса устройств и номера сетей:

На ISP

Сетевое устройство (net0)	virtio=CA:78:15:F4:58:CC,bridge=vmbr0,firewall=1
Сетевое устройство (net1)	virtio=C6:AE:53:A7:5B:7E,bridge=vmbr1025
Сетевое устройство (net2)	virtio=B2:6F:18:1B:5D:12,bridge=vmbr1026

На HQ-RTR

Сетевое устройство (net1)	vmxnet3=1C:87:76:40:24:97,bridge=vmbr1025
Сетевое устройство (net2)	vmxnet3=1C:87:76:40:06:DA,bridge=vmbr1027

На BR-RTR

```
≡ Сетевое устройство (net1) vmxnet3=1C:87:76:40:60:0A,bridge=vmbr1026  
≡ Сетевое устройство (net2) vmxnet3=1C:87:76:40:04:0C,bridge=vmbr1028
```

На HQ-SRV

```
≡ Сетевое устройство (net0) virtio=9E:15:D3:6C:9A:29,bridge=vmbr1027,tag=100
```

На BR-SRV

```
≡ Сетевое устройство (net0) virtio=0A:FE:15:4B:BC:B5,bridge=vmbr1028
```

В итоге должно быть так на HQ-SRV.

```
[root@hq-srv ens19]# cat options  
BOOTPROTO=static  
TYPE=eth  
CONFIG_WIRELESS=no  
SYSTEMD_BOOTPROTO=static  
CONFIG_IPV4=yes  
DISABLED=no  
NM_CONTROLLED=no  
SYSTEMD_CONTROLLED=no  
[root@hq-srv ens19]# cat ipv4address  
192.168.100.62/26  
[root@hq-srv ens19]# cat ipv4route  
default via 192.168.100.1
```

```
[root@hq-srv ens19]# ip -br -c a  
lo          UNKNOWN      127.0.0.1/8 ::1/128  
ens19        UP          192.168.100.62/26 fe80::9e15:d3ff:fe6c:9a29/64  
[root@hq-srv ens19]# ip -c r  
default via 192.168.100.1 dev ens19  
192.168.100.0/26 dev ens19 proto kernel scope link src 192.168.100.62
```

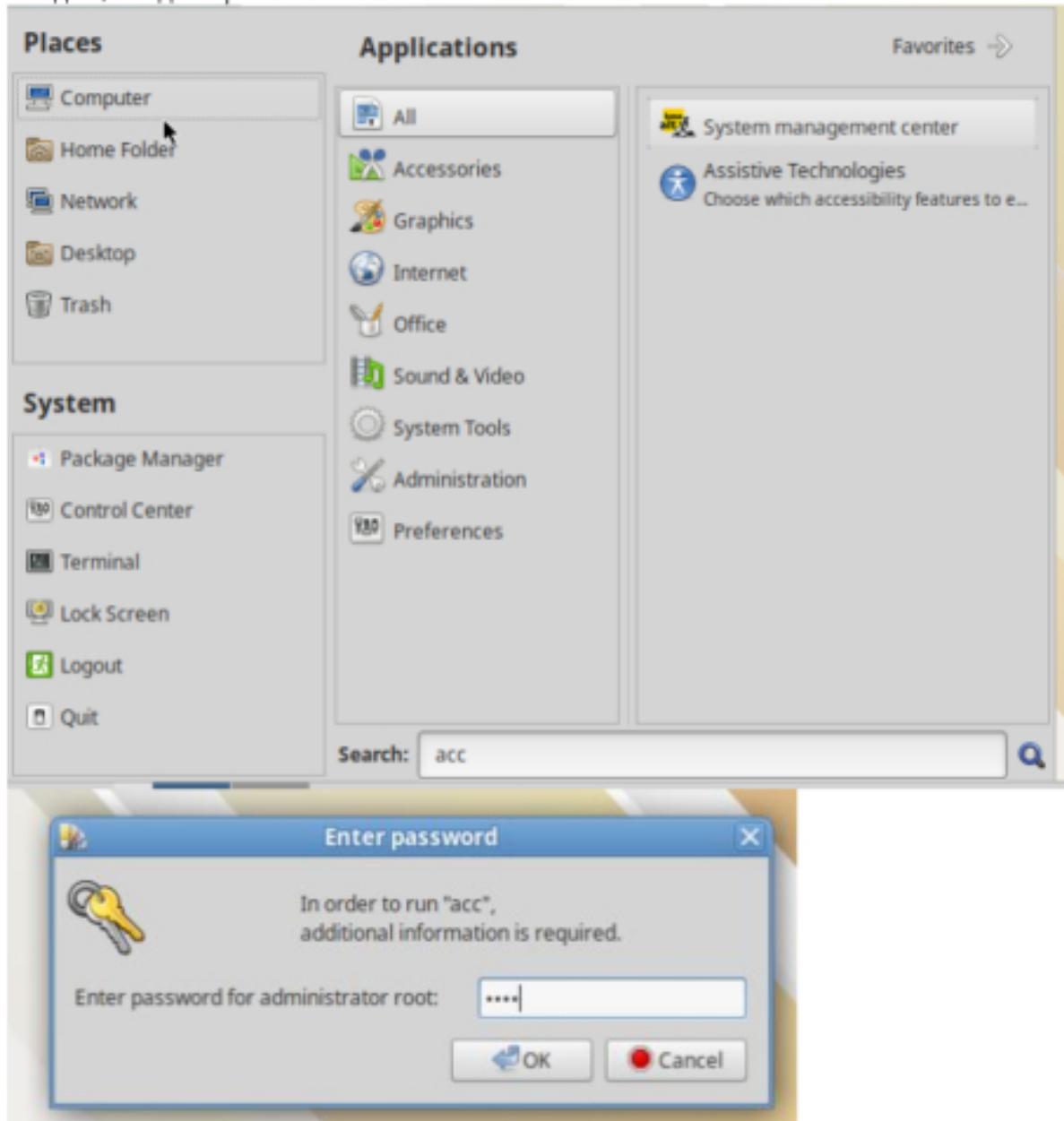
И на BR-SRV

```
[root@noname ens19]# cat options ipv4address ipv4route  
BOOTPROTO=static  
TYPE=eth  
CONFIG_WIRELESS=no  
SYSTEMD_BOOTPROTO=static  
CONFIG_IPV4=yes  
DISABLED=no  
NM_CONTROLLED=no  
SYSTEMD_CONTROLLED=no  
192.168.3.30/27  
default via 192.168.3.1
```

```
[root@noname ens19]# ip -br -c a  
lo          UNKNOWN      127.0.0.1/8 ::1/128  
ens19        UP          192.168.3.30/27 fe80::8fe:15ff:fe4b:bch5/64  
[root@noname ens19]# ip -c r  
default via 192.168.3.1 dev ens19  
192.168.3.0/27 dev ens19 proto kernel scope link src 192.168.3.30
```

Настроим адресацию на HQ-CLI

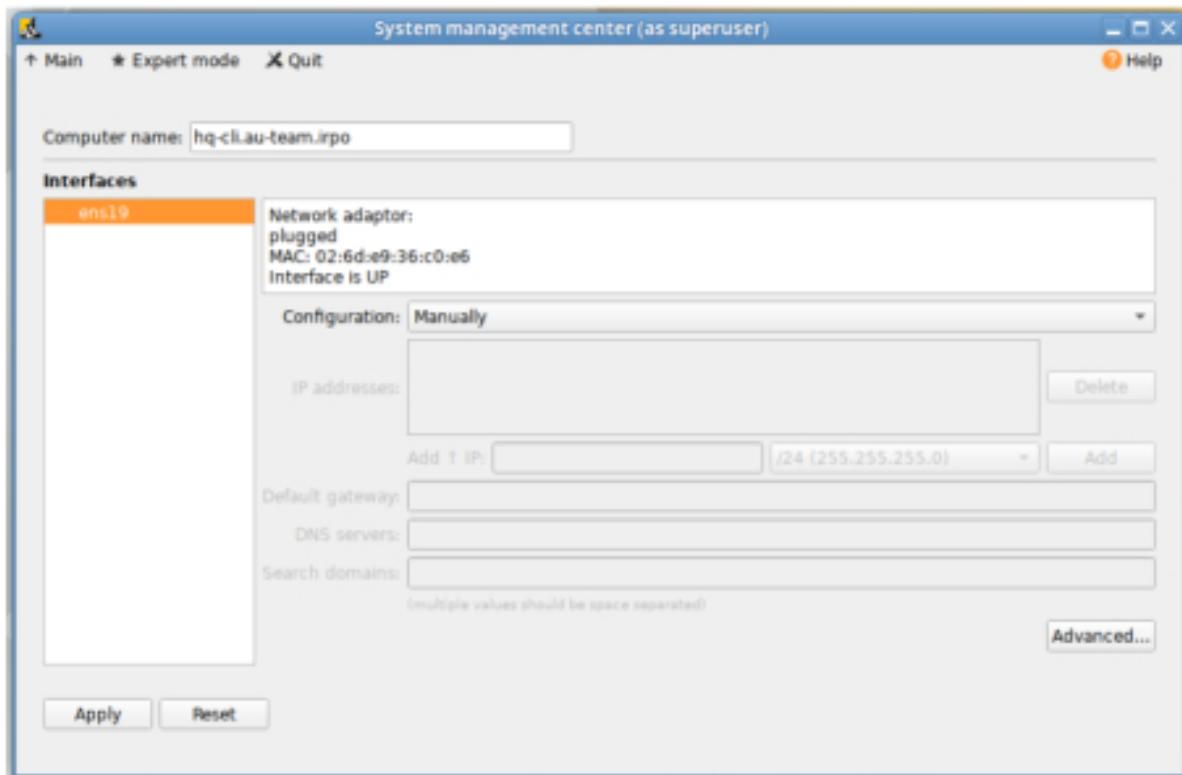
Сделаем это через сетевую службу etcnet. Найдем центр управления системой и войдем, введя пароль



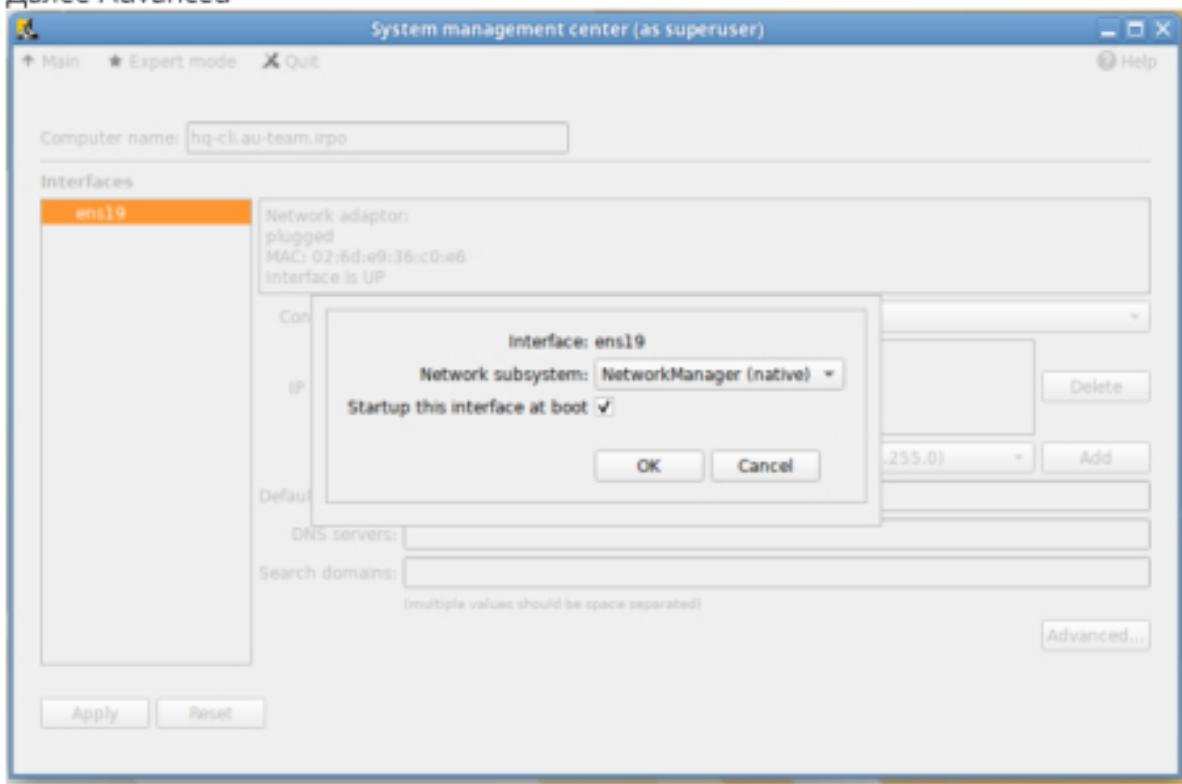
Далее



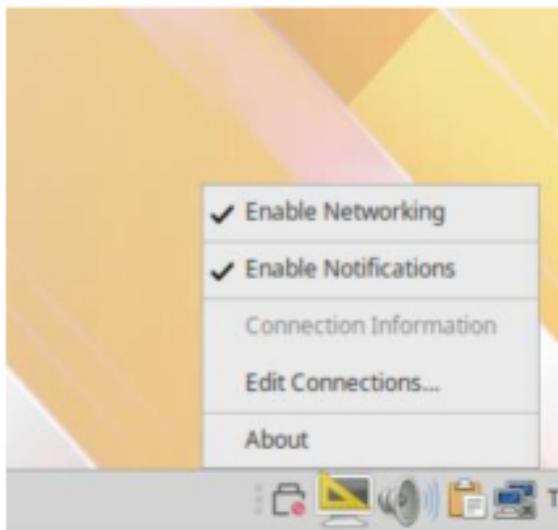
Пропишем сразу имя устройства



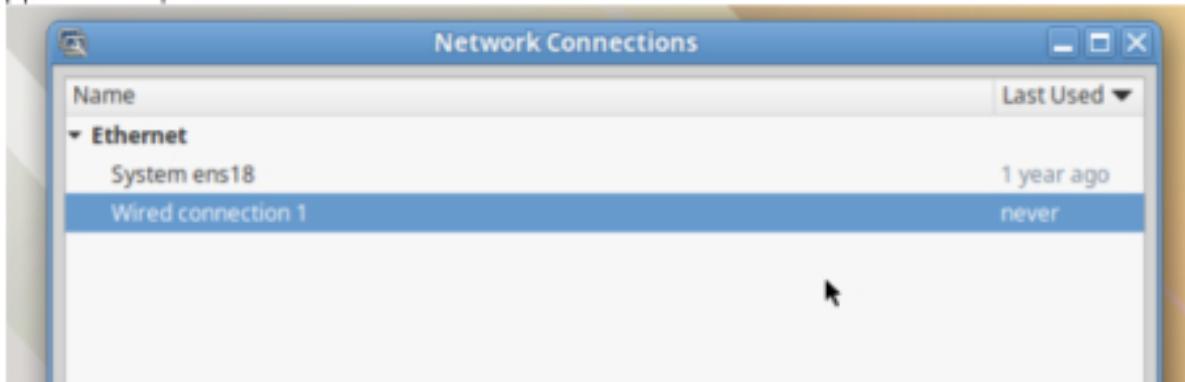
Далее Advanced



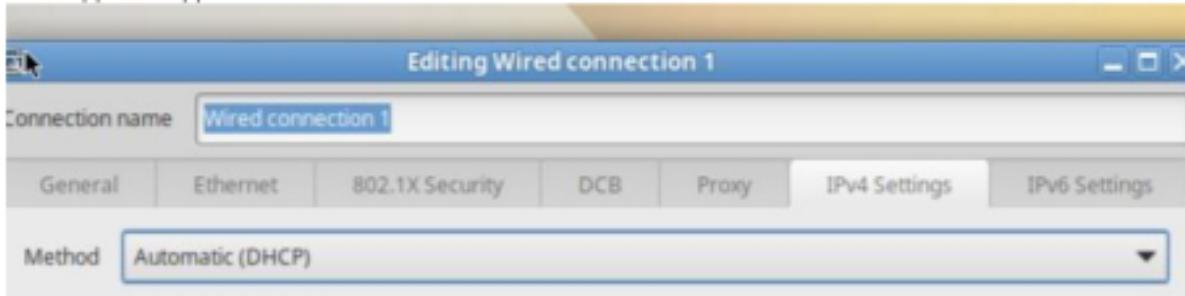
Здесь мы видим, что сетевая подсистема используется NM, не будем ее менять, но нажмем Apply для сохранения имени устройства. Перейдем в настройки интерфейса через апллет NM и выставим DHCP



Двойной щелчок



По заданию должен быть DHCP



Настройка адресации на **HR-RTR (Ecorouter)**

Логин/пароль: admin/admin

Настраиваем интерфейс на HQ-RTR, который смотрит в сторону ISP

- Разберёмся с основными понятиями касающимися **EcoRouter**:
 - Порт (**port**) – это устройство в составе **EcoRouter**, которое работает на уровне коммутации (L2);
 - Интерфейс (**interface**) – это логический интерфейс для адресации, работает на сетевом уровне (L3);

- **Service instance** (Сабинтерфейс, SI, Сервисный интерфейс) является логическим сабинтерфейсом, работающим между L2 и L3 уровнями:
 - Данный вид интерфейса необходим для соединения физического порта с интерфейсами L3, интерфейсами bridge, портами;
 - Используется для гибкого управления трафиком на основании наличия меток VLANов в фреймах, или их отсутствия;
 - Сквозь сервисный интерфейс проходит весь трафик, приходящий на порт.
- Таким образом, для того чтобы назначить IPv4-адрес на EcoRouter - необходимо придерживаться следующего алгоритма в общем виде:
 - Создать интерфейс с произвольным именем и назначить на него IPv4;
 - В режиме конфигурирования порта - создать service-instance с произвольным именем:
 - указать (инкапсулировать) что будет обрабатываться тегированный или не тегированный трафик;
 - указать в какой интерфейс (ранее созданный) нужно отправить обработанные кадры.

Просмотрим порты:

Name	Physical	Admin	Lacp	Description
te0	UP	UP	*	
te1	UP	UP	*	

Перейдем в режим конфигурации и создадим логический интерфейс

```
hq-rtr#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
hq-rtr(config)#int isp
hq-rtr(config-if)#ip address 172.14.4.2/28
hq-rtr(config-if)#description "Connect ISP"
hq-rtr(config-if)#exit
```

Далее настраиваем физический порт и объединением его с интерфейсом:

```
hq-rtr(config)#port te0
hq-rtr(config-port)#service-instance te0/isp
hq-rtr(config-service-instance)#encapsulation untagged
hq-rtr(config-service-instance)#connect ip interface isp

2025-05-26 11:39:48      INFO      Interface isp changed state to up
hq-rtr(config-service-instance)#exit
hq-rtr(config-port)#exit
```

Проверим соединение с ISP. В результате успешного соединения нужно сохранить наши настройки

```
ecorouter(config)#do ping 172.16.4.1
PING 172.16.4.1 (172.16.4.1) 56(84) bytes of data.
64 bytes from 172.16.4.1: icmp_seq=2 ttl=64 time=14.1 ms
64 bytes from 172.16.4.1: icmp_seq=3 ttl=64 time=13.3 ms

--- 172.16.4.1 ping statistics ---
3 packets transmitted, 2 received, 33.3333% packet loss, time 2062ms
rtt min/avg/max/mdev = 13.307/13.723/14.139/0.416 ms
ecorouter(config)#exit
ecorouter#write
Building configuration...
```

Настраиваем интерфейсы на **HQ-RTR**, которые смотрят в сторону **HQ-SRV** и **HQ-CLI** (с разделением на VLAN):

- Создаем три интерфейса:

```
hq-rtr(config)#int vl100
hq-rtr(config-if)#description "Servers - vlan100"
hq-rtr(config-if)#ip address 192.168.100.1/26
hq-rtr(config-if)#exit
hq-rtr(config)#int vl200
hq-rtr(config-if)#description "Clients - vlan200"
hq-rtr(config-if)#ip address 192.168.100.65/28
hq-rtr(config-if)#int vl999
hq-rtr(config-if)#description "Managements - vlan999"
hq-rtr(config-if)#ip address 192.168.100.81/29
hq-rtr(config-if)#exit
hq-rtr(config)##
```

- Настраиваем порт и объединяем его с интерфейсами:

```
hq-rtr(config)#port te1
hq-rtr(config-port)#service-instance te1/vl100
hq-rtr(config-service-instance)#encapsulation dot1q 100 exact
hq-rtr(config-service-instance)#rewrite pop 1
hq-rtr(config-service-instance)#connect ip interface vl100

2025-05-26 11:06:16      INFO      Interface vl100 changed state to up
hq-rtr(config-service-instance)#exit
```

```
hq-rtr(config-port)#service-instance te1/vl200
hq-rtr(config-service-instance)#encapsulation dot1q 200 exact
hq-rtr(config-service-instance)#rewrite pop 1
hq-rtr(config-service-instance)#connect ip interface vl200

2025-05-26 11:08:57      INFO      Interface vl200 changed state to up
hq-rtr(config-service-instance)#exit
```

```

hq-rtr(config-port)#service-instance te1/v999
hq-rtr(config-service-instance)#encapsulation dot1q 999 exact
hq-rtr(config-service-instance)#rewrite pop 1
hq-rtr(config-service-instance)#connect ip interface v1999

2025-05-26 11:10:34      INFO      Interface v1999 changed state to up
hq-rtr(config-service-instance)#exit

```

Сохраним конфигурацию:

```

hq-rtr(config)#write memory
Building configuration...

```

Проверим:

hq-rtr(config)#do show ip int brief			
Interface	IP-Address	Status	URF
isp	172.16.4.2/28	up	default
v1100	192.168.100.1/26	up	default
v1200	192.168.100.65/28	up	default
v1999	192.168.100.81/29	up	default

hq-rtr(config)#do show interface description			
Interface	Status	Protocol	Description
isp	up	up	"Connect ISP"
v1100	up	up	"Servers - vlan100"
v1200	up	up	"Clients - vlan200"
v1999	up	up	"Managements - vlan999"

Адресация на BR-RTR (без разделения на VLAN) настраивается аналогично примеру выше

```

ecorouter(config)#int int0
ecorouter(config-if)#description "to isp"
ecorouter(config-if)#ip address 172.16.5.2/28
ecorouter(config-if)#exit
ecorouter(config)#port te0
ecorouter(config-port)#service-instance te0/in
ecorouter(config-port)#service-instance te0/int0
ecorouter(config-service-instance)#encapsulation untagged
ecorouter(config-service-instance)#connect ip interface int0

2025-03-27 12:48:30      INFO      Interface int0 changed state to up
ecorouter(config-service-instance)#exit

```

```
ecorouter(config)#int int1
ecorouter(config-if)#description "to br-srv"
ecorouter(config-if)#ip add
ecorouter(config-if)#ip address 192.168.3.1/27
ecorouter(config-if)#exit
ecorouter(config)#port te1
ecorouter(config-port)#service-instance te1/int1
ecorouter(config-service-instance)#encapsulation untagged
ecorouter(config-service-instance)#connect ip interface int1

2025-03-27 12:53:56      INFO      Interface int1 changed state to up
ecorouter(config-service-instance)#exit
```

Проверим

```
ecorouter#show ip int brief
Interface          IP-Address          Status
-----
int0              172.16.5.2/28        up
int1              192.168.3.1/27      up

ecorouter#ping 172.16.5.1
PING 172.16.5.1 (172.16.5.1) 56(84) bytes of data.
64 bytes from 172.16.5.1: icmp_seq=2 ttl=64 time=12.0 ms
```

Добавление маршрута по умолчанию в EcoRouter

Прописываем маршрут по умолчанию и адрес следующего перехода:

на HQ-RTR

```
hq-rtr(config)#ip route 0.0.0.0/0 172.16.4.1
hq-rtr(config)#wr
^
* Invalid input detected at '^' marker.

hq-rtr(config)#write
```

Посмотрим на статическую маршрутизацию:

```
hq-rtr(config)#do show ip route static
IP Route Table for VRF "default"
Gateway of last resort is 172.16.4.1 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 172.16.4.1, isp
```

на BR-RTR

```
ecorouter(config)#ip route 0.0.0.0 0.0.0.0 172.16.5.1  
ecorouter(config)#write  
Building configuration...
```

Проверим связность с HQ-SRV, HQ-CLI и BR-SRV

```
ecorouter#ping 192.168.100.62  
PING 192.168.100.62 (192.168.100.62) 56(84) bytes of data.  
64 bytes from 192.168.100.62: icmp_seq=2 ttl=64 time=12.0 ms  
ecorouter#ping 192.168.200.14  
PING 192.168.200.14 (192.168.200.14) 56(84) bytes of data.  
64 bytes from 192.168.200.14: icmp_seq=1 ttl=64 time=13.6 ms  
ecorouter#ping 192.168.3.30  
PING 192.168.3.30 (192.168.3.30) 56(84) bytes of data.  
64 bytes from 192.168.3.30: icmp_seq=2 ttl=64 time=12.0 ms
```

Не забудем сменить имя маршрутизатора BR

```
br-rtr(config)#hostname br-rtr  
br-rtr(config)#ip domain-name au-team.irpo  
br-rtr(config)#write
```

2. Настройка ISP

- Настройте адресацию на интерфейсах:
 - Интерфейс, подключенный к магистральному провайдеру, получает адрес по DHCP
 - Настройте маршруты по умолчанию там, где это необходимо
 - Интерфейс, к которому подключен HQ-RTR, подключен к сети 172.16.4.0/28
 - Интерфейс, к которому подключен BR-RTR, подключен к сети 172.16.5.0/28
 - На ISP настройте динамическую сетевую трансляцию в сторону HQ-RTR и BR-RTR для доступа к сети Интернет

Все пункты из этого задания мы выполнили во время Базовой настройки устройств, кроме последнего

Включение маршрутизации

На ISP

В файле `/etc/net/sysctl.conf` изменяем строку:

```
[root@isp ifaces]# vim /etc/net/sysctl.conf
```

0 меняем на 1 и сохраняем :wq

```
#  
net.ipv4.ip_forward = 1
```

Изменения в файле `sysctl.conf` применяем следующей командой:

```
[root@isp ifaces]# sysctl -p /etc/sysctl.conf
```

3. Создание локальных учетных записей

- 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@$$word`
 - При настройке на EcoRouter пользователь `net_admin` должен обладать максимальными привилегиями
 - При настройке ОС на базе Linux, запускать `sudo` без дополнительной аутентификации

Создание пользователя `sshuser` на серверах

Создаем самого пользователя на HQ-SRV:

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

опция `-u` позволяет указать идентификатор пользователя сразу при создании

Задаем пароль:

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

Добавляем в группу `wheel`:

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

Проверим

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

Изменяем строку в /etc/sudoers:

```
[root@hq-srv ~]# vim /etc/sudoers  
WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL
```

Все пользователи группы wheel будут выполнять команду sudo без пароля. Так как других пользователей у нас нет, то формально требование задания выполнено.

Если бы были другие пользователи, кроме sshuser, то изменения выглядели бы так:

```
sshuser ALL=(ALL:ALL) NOPASSWD: ALL
```

Аналогичные операции проделываем на BR-SRV

Проверим, выполнив любую команду, которая запрашивает пароль, например

```
[sshuser@br-srv ~]$ sudo -i  
[root@br-srv ~]#
```

Команда выполнилась без требования пароля

Создание пользователя net_admin на Ecorouter (на HQ-RTR)

Создаем самого пользователя, пароль для него и назначаем роль администратора. Не забываем сохранить изменение в конфигурации

```
hq-rtr(config)#username net_admin  
hq-rtr(config-user)#password P@ssw0rd  
hq-rtr(config-user)#role admin  
hq-rtr(config-user)#exit  
hq-rtr(config)#write
```

Проверим:

```
hq-rtr#show users localdb  
User: net_admin  
Description:  
Docker socket access: disabled  
UR:  
    pvr  
Roles:  
    admin ''
```

Аналогичные операции проделываем на BR-RTR

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

Виртуальный коммутатор создан средствами AltPVE. Сервер HQ-SRV находится в ID VLAN 100

```
= Сетевое устройство (net0) virtio=9E:15:D3:6C:9A:29,bridge=vmbr1027,tag=100
```

Клиент HQ-CLI находится в ID VLAN 200

```
= Сетевое устройство (net0) virtio=0E:40:5E:6D:09:A1,bridge=vmbr1027,tag=200
```

Подсеть управления с ID VLAN 999 была создана ранее

Проверяем ее наличие еще раз:

Interface	IP-Address	Status	URF
isp	172.16.4.2/28	up	default
v1100	192.168.100.1/26	up	default
v1200	192.168.100.65/28	up	default
v1999	192.168.100.81/29	up	default

5. Настройка безопасного удаленного доступа на серверах HQ-SRV и BR-SRV

- Для подключения используйте порт 2024
- Разрешите подключения только пользователю sshuser
- Ограничите количество попыток входа до двух
- Настройте баннер «Authorized access only»

На HQ-SRV

Приводим указанные строки в файле /etc/openssh/sshd_config к следующим значениям, не забыв их раскомментировать (убрать знак # перед):

```
Port 2024
MaxAuthTries 2
PasswordAuthentication yes
Banner /etc/openssh/bannermotd
AllowUsers sshuser
```

В параметре AllowUsers вместо пробела используется Tab

Создаем файл bannermotd (Можно без рамки):

```
[root@hq-srv ~]# vim /etc/openssh/bannermotd
```

```
-----  
|      Authorized access only!      |  
-----
```

Перезагружаем службу:

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

На BR-SRV выполняются аналогичные действия

6. Между офисами HQ и BR необходимо сконфигурировать IP-туннель

- Сведения о туннеле занесите в отчёт
- На выбор технологии GRE или IP in IP

Создание туннеля на HQ-RTR

Создаем интерфейс GRE-туннеля на HQ-RTR:

```
hq-rtr(config)# interface tunnel.0
```

Назначаем IP-адрес:

```
hq-rtr(config-if-tunnel)# ip address 10.10.10.1/30
```

Выставляем параметр MTU:

```
hq-rtr(config-if-tunnel)# ip mtu 1400
```

В связи с добавлением служебного заголовка появляются новые требования к допустимому значению MTU при передаче пакета. Заголовок GRE имеет размерность 4 байта, 20 байт транспортный IP заголовок, заголовок IP пакета 20 байт, таким образом возникает необходимость задавать размер допустимого MTU на интерфейсах туннеля меньше стандартного значения.

Задаем режим работы туннеля **GRE** и адреса **начала и конца** туннеля, используем внешние адреса интерфейсов смотрящих на ISP:

```
hq-rtr(config-if-tunnel)#ip tunnel 172.16.4.2 172.16.5.2 mode gre
```

```
hq-rtr(config)#exit  
hq-rtr#write
```

Проверяем:

Interface	IP-Address	Status	URF
isp	172.16.4.2/28	up	default
vl100	192.168.100.1/26	up	default
vl200	192.168.100.65/28	up	default
vl999	192.168.100.81/29	up	default
tunnel.0	10.10.10.1/30	up	default

Или

```
hq-rtr(config)#do show interface tunnel.0  
Interface tunnel.0 is up  
  Description: "GRE-to-BR"  
  Snmp index: 9  
  Ethernet address: (port not configured)  
  MTU: 1476  
  Tunnel source: 172.16.4.2  
  Tunnel destination: 172.16.5.2  
  Tunnel mode: GRE  
  Tunnel keepalive: disabled  
  NAT: no  
  ARP Proxy: disable  
  ICMP redirects on, unreachables on  
  IP URPF is disabled  
  Label switching is disabled  
<UP,BROADCAST,RUNNING,NOARP,MULTICAST>  
  inet 10.10.10.1/30 broadcast 10.10.10.3/30  
    total input packets 0, bytes 0  
    total output packets 0, bytes 0
```

GRE-туннель на BR-RTR настраивается аналогично примеру выше, где адрес будет 10.10.10.2/30 и не забываем указать на первом месте адрес интерфейса BR-RTR

```
br-rtr(config-if-tunnel)#ip tunnel 172.16.5.2 172.16.4.2 mode gre
```

Interface	IP-Address	Status	URF
isp	172.16.5.2/28	up	default
br-net	192.168.3.1/27	up	default
tunnel.0	10.10.10.2/30	up	default

Проверяем работу туннеля:

```
br-rtr#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=2 ttl=64 time=28.0 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=26.7 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=26.3 ms

--- 10.10.10.1 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3012ms
rtt min/avg/max/mdev = 26.268/26.966/27.979/0.732 ms
```

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

Настройка OSPF на HQ-RTR

Создаем процесс **OSPF**, указываем **идентификатор маршрутизатора**,
указываем **пассивные** интерфейсы и объявляем сети:

```
hq-rtr(config)#router ospf 1
hq-rtr(config-router)#ospf router-id 10.10.10.1
hq-rtr(config-router)#passive-interface default
hq-rtr(config-router)#no passive-interface tunnel.0
hq-rtr(config-router)#network 10.10.10.0/30 area 0
hq-rtr(config-router)#network 192.168.100.0/26 area 0
hq-rtr(config-router)#network 192.168.100.64/28 area 0
hq-rtr(config-router)#network 192.168.100.80/29 area 0
hq-rtr(config-router)#exit
```

Обеспечим защиту выбранного протокола посредством парольной защиты

```
hq-rtr(config)#interface tunnel.0
hq-rtr(config-if-tunnel)#ip ospf authentication md5
hq-rtr(config-if-tunnel)#ip ospf authentication message-digest
hq-rtr(config-if-tunnel)#ip ospf message-digest-key 1 md5 P@ssw0rd
hq-rtr(config-if-tunnel)#exit
hq-rtr(config)#wri
hq-rtr(config)#write
Building configuration...
```

Проверим:

```
hq-rtr(config)#do show ip ospf interface tunnel.0
tunnel.0 is up, line protocol is up
  Internet Address 10.10.10.1/30, Area 0.0.0.0, MTU 1476
  Process ID 1, URF (default), Router ID 10.10.10.1, Network Type BR
  Transmit Delay is 1 sec, State DR, Priority 1, TE Metric 1
  Designated Router (ID) 10.10.10.1, Interface Address 10.10.10.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit
    Hello due in 00:00:06
  Neighbor Count is 0, Adjacent neighbor count is 0
  Crypt Sequence Number is 89990
  Hello received 0 sent 33, DD received 0 sent 0
  LS-Req received 0 sent 0, LS-Upd received 0 sent 0
  LS-Ack received 0 sent 0, Discarded 0
  Message-digest authentication, using key-id 1
```

Аутентификация присутствует.

Маршрутизация OSPF и защита туннельного интерфейса на BR-RTR настраивается аналогично примеру выше

```
br-rtr(config)#router ospf 1
br-rtr(config-router)#ospf router-id 10.10.10.2
br-rtr(config-router)#passive-interface default
br-rtr(config-router)#no passive-interface tunnel.0
br-rtr(config-router)#network 10.10.10.0/30 area 0
br-rtr(config-router)#network 192.168.3.0/27 area 0
br-rtr(config-router)#exit
br-rtr(config)#write
Building configuration...
```

```
br-rtr(config)#interface tunnel.0
br-rtr(config-if-tunnel)#ip ospf u
br-rtr(config-if-tunnel)#ip ospf authentication message-digest
br-rtr(config-if-tunnel)#ip ospf message-digest-key 1 md5 P@ssw0rd
br-rtr(config-if-tunnel)#exit
br-rtr(config)#write
Building configuration...
```

Проверим соседство:

```

hq-rtr#show ip ospf neighbor

Total number of full neighbors: 1
OSPF process 1 VRF(default):
Neighbor ID      Pri   State          Dead Time    Address           Interface
10.10.10.2        1     Full/DR       00:00:40    10.10.10.2      tunnel.0

hq-rtr#show ip route ospf
IP Route Table for VRF "default"
0      192.168.3.0/27 [110/2] via 10.10.10.2, tunnel.0, 00:05:21
Gateway of last resort is not set

```

```

br-rtr(config)#do show ip ospf neighbor

Total number of full neighbors: 1
OSPF process 1 VRF(default):
Neighbor ID      Pri   State          Dead Time    Address           Interface
10.10.10.1        1     Full/Backup   00:00:38    10.10.10.1      tunnel.0

br-rtr(config)#do show ip route ospf
IP Route Table for VRF "default"
0      192.168.100.0/26 [110/2] via 10.10.10.1, tunnel.0, 00:02:45
0      192.168.100.64/28 [110/2] via 10.10.10.1, tunnel.0, 00:02:45
0      192.168.100.80/29 [110/2] via 10.10.10.1, tunnel.0, 00:02:45

Gateway of last resort is not set

```

Так же можно проверить пинг между серверами:

```

[root@hq-srv ens19]# ping 192.168.3.30 -c 3
PING 192.168.3.30 (192.168.3.30) 56(84) bytes of data.
64 bytes from 192.168.3.30: icmp_seq=1 ttl=62 time=54.7 ms
64 bytes from 192.168.3.30: icmp_seq=2 ttl=62 time=53.2 ms
64 bytes from 192.168.3.30: icmp_seq=3 ttl=62 time=51.4 ms

--- 192.168.3.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 51.394/53.083/54.658/1.334 ms

[root@hq-srv ens19]# tracepath -n 192.168.3.30
1?: [LOCALHOST]                                pmtu 1500
1:  192.168.100.1                               6.131ms
1:  192.168.100.1                               53.931ms
2:  192.168.100.1                               7.489ms pmtu 1476
2:  10.10.10.2                                 22.576ms
3:  192.168.3.30                               78.218ms reached

Resync: pmtu 1476 hops 3 back 3

```

Видим, что по туннелю присутствует связность

8. Настройка динамической трансляции адресов

- Настройте динамическую трансляцию адресов для обоих офисов.
- Все устройства в офисах должны иметь доступ к сети Интернет

Настройка NAT на ISP

Добавляем iptables на ISP

```
[root@isp ~]# apt-get update
```

```
[root@isp ~]# apt-get install iptables -y
```

Добавляем правило iptables на ISP

```
[root@isp ~]# iptables -t nat -A POSTROUTING -o ens19 -j MASQUERADE
```

```
[root@isp ~]# iptables-save >> /etc/sysconfig/iptables
```

```
[root@isp ~]# systemctl enable --now iptables
```

```
[root@isp ~]# systemctl restart network
```

Проверим наше правило – должна появиться запись в цепочке POSTROUTING:

```
[root@isp ~]# iptables -t nat -L -n -v
```

Проверим запросом ping 8.8.8.8 на маршрутизаторах

```
hq-rtr#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=58 time=80.6 ms
```

```
br-rtr#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=58 time=93.8 ms
```

Далее необходимо настроить пат на маршрутизаторах

Настройка NAT на HQ-RTR

Указываем какие внутренние и какие внешние интерфейсы:

```
hq-rtr(config)#interface isp
hq-rtr(config-if)#ip nat outside
hq-rtr(config-if)#exit
hq-rtr(config)#interface v1100
hq-rtr(config-if)#ip nat inside
hq-rtr(config-if)#exit
hq-rtr(config)#interface v1200
hq-rtr(config-if)#ip nat inside
hq-rtr(config-if)#interface v1999
hq-rtr(config-if)#ip nat inside
hq-rtr(config-if)#exit
```

Создаем пул:

```
hq-rtr(config)#ip nat pool HQ 192.168.100.1-192.168.100.254
```

где, HQ – имя пула адресов

Создаем правило трансляции адресов, указывая внешний интерфейс:

```
hq-rtr(config)#ip nat source dynamic inside-to-outside pool HQ overload interface isp
```

```
hq-rtr(config)#write
```

Проверим доступность сети интернет на HQ-SRV:

```
[root@hq-srv ~]# ping -c 3 ya.ru
PING ya.ru (77.88.44.242) 56(84) bytes of data.
64 bytes from ya.ru (77.88.44.242): icmp_seq=1 ttl=55 time=93.3 ms
64 bytes from ya.ru (77.88.44.242): icmp_seq=2 ttl=55 time=91.8 ms
64 bytes from ya.ru (77.88.44.242): icmp_seq=3 ttl=55 time=93.3 ms
```

На роутере проверим функционирование nat

```
hq-rtr(config)#do show ip nat translations
```

```
Time: 274s, Protocol: UDP, URF: default
IN: 192.168.100.62:49541    172.16.4.2:1032      162.159.200.123:123
OUT: 162.159.200.123:123   192.168.100.62:49541   172.16.4.2:1032

Time: 274s, Protocol: UDP, URF: default
IN: 192.168.100.62:41461   172.16.4.2:1031      162.159.200.1:123
OUT: 162.159.200.1:123    192.168.100.62:41461   172.16.4.2:1031

Total: 5
```

Настройка NAT на BR-RTR

Конфигурация:

```
br-rtr(config)#interface isp
br-rtr(config-if)#ip nat outside
br-rtr(config-if)#exit
br-rtr(config)#interface br-net
br-rtr(config-if)#ip nat inside
br-rtr(config-if)#exit
```

```
br-rtr(config)#interface br-net
br-rtr(config-if)#ip nat inside
br-rtr(config-if)#exit
br-rtr(config)#ip nat pool BR 192.168.3.1-192.168.3.254
br-rtr(config)#ip nat source dynamic inside-to-outside pool BR
br-rtr(config)#ip nat source dynamic inside-to-outside pool BR overload interface isp
br-rtr(config)#write
Building configuration...
```

Проверим

```
[root@br-srv ~]# 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=57 time=145 ns
```

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

Подсеть у нас 192.168.100.64/28

На HQ-RTR создаем пул для DHCP-сервера, где HQ-CLI имя пула адресов. Диапазон начинаем с 66, так как по заданию не выдаем адрес маршрутизатора:

```
hq-rtr(config)#ip pool HQ-CLI 192.168.100.66-192.168.100.78
```

Для настройки DHCP-сервера необходимо в режиме конфигурации ввести команду **dhcp-server <NUMBER>**, где **NUMBER** – номер сервера в системе маршрутизатора:

```
hq-rtr(config)#dhcp-server 1
```

Привязываем ранее созданный POOL раздаваемых адресов с именем CLI-HQ, а также указанием номера сервера в системе маршрутизатора 1:

```
hq-rtr(config-dhcp-server)#pool HQ-CLI 1
```

Задаём основные параметры для раздачи DHCP сервером:

```
hq-rtr(config)#dhcp-server 1
```

```
hq-rtr(config)#dhcp-server 1
hq-rtr(config-dhcp-server)#pool HQ-CLI 1
hq-rtr(config-dhcp-server-pool)#mask 28
hq-rtr(config-dhcp-server-pool)#gateway 192.168.100.65
hq-rtr(config-dhcp-server-pool)#dns 192.168.100.62
hq-rtr(config-dhcp-server-pool)#domain-name au-team.irpo
hq-rtr(config-dhcp-server-pool)#exit
hq-rtr(config-dhcp-server)#exit
```

Привязываем DHCP-сервер к интерфейсу (смотрящий в сторону CLI):

```
hq-rtr(config)#interface v1200
hq-rtr(config-if)#dhcp-server 1
hq-rtr(config-if)#write
Building configuration...
```

Проверяем

Видим, что появилось соединение



```
user@hq-cli ~ $ 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: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
        link/ether 02:6d:e9:36:c0:e6 brd ff:ff:ff:ff:ff:ff
        altname enp0s19
        inet 192.168.100.66/28 brd 192.168.100.79 scope global dynamic noprefixroute
            ens19
            valid_lft 86247sec preferred_lft 86247sec
            inet6 fe80::7d98:3e8f:9615:3371/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
user@hq-cli ~ $ ip route
default via 192.168.100.65 dev ens19 proto dhcp metric 100
192.168.100.64/28 dev ens19 proto kernel scope link src 192.168.100.66 metric 100
user@hq-cli ~ $ cat /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
search au-team.irpo
nameserver 192.168.100.62
```

Вывод `global dynamic` говорит о том, что адрес получен по DHCP, маршрут по умолчанию есть и DNS сервер тоже.

Проверяем на HQ-RTR

```
hq-rtr(config)#do show dhcp-server clients v1200
Total DHCP clients count: 1
Client          Client          Server          Server
IP Address      MAC Address    ACK Time     Lease Time
-----
192.168.100.66 026d.e936.c0e6  342          86400
```

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

Устройство	Запись	Тип
HQ-RTR	hq-rtr.au-team.irpo	A,PTR
BR-RTR	br-rtr.au-team.irpo	A
HQ-SRV	hq-srv.au-team.irpo	A,PTR
HQ-CLI	hq-cli.au-team.irpo	A,PTR
BR-SRV	br-srv.au-team.irpo	A
HQ-RTR	moodle.au-team.irpo	CNAME
BR-RTR	wiki.au-team.irpo	CNAME

Настройка конфигурации bind

Устанавливаем необходимые пакеты:

```
[root@hq-srv ~]# apt-get install bind bind-utils -y
Reading Package Lists... Done
Building Dependency Tree... Done
E: Couldn't find package bind
```

Вывалился ошибка, про не найденный пакет. Необходимо настроить файл resolv.conf в /etc/net/Interfaces/ens19

```
[root@hq-srv ~]# vim /etc/net/Interfaces/ens19/resolv.conf
```

Прописываем и сохраняем

```
nameserver 77.88.8.8
```

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

Пробуем заново

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

```
[root@hq-srv ~]# apt-get install bind bind-utils -y
```

```
Committing changes...
Preparing...                                #####
Updating / installing...
1: libuv-1.49.2-alt0.p10.1                  #####
2: libbind-9.16.50-alt2                     #####
3: bind-control-1.3-alt1                   #####
4: bind-9.16.50-alt2                      #####
5: bind-utils-9.16.50-alt2                 #####
Done.
```

Изменяем содержание перечисленных строк в /etc/bind/options.conf к следующему виду:

```
listen-on { 192.168.100.62; }; #Наш адрес явно либо можно апу
listen-on-v6 { none; };
forwarders { 77.88.8.8; };
allow-query { any; }; #Прописываем все локальные сети или апу
allow-recursion { any; };
```

1. **listen-on** - сетевые интерфейсы, которые будет прослушивать служба
2. **forwarders** - DNS-сервер, на который будут перенаправляться запросы клиентов
3. **allow-query** - IP-адреса и подсети от которых будут обрабатываться запросы
4. **allow-recursion** – рекурсивные запросы

Создание и настройка прямой зоны

Обозначим зоны. Отредактируем файл rfc1912, дописав в конце параметры для зоны прямого просмотра и обратного

```
[root@hq-srv ~]# vim /etc/bind/rfc1912.conf
```

```
zone "255.in-addr.arpa" {
    type master;
    file "empty";
    allow-update { none; };
};

zone "au-team.irpo" {
    type master;
    file "au-team.irpo";
    allow-update { none; };
};

zone "192.168.100.in-addr.arpa" {
    type master;
    file "192.168.100.in-addr.arpa";
    allow-update { none; };
};
```

Скопируем шаблон файла и зададим ему имя нашей прямой зоны au-team.irpo

```
[root@hq-srv ~]# cp /etc/bind/zone/empty /etc/bind/zone/au-team.irpo
```

Здесь пропишем параметры нашей прямой зоны

```
: DO NOT EDIT THIS FILE - it is used for multiple zones.
: Instead, copy it and use that copy.
:
$TTL    1D
@      IN      SOA    au-team.irpo. root.au-team.irpo. (
                      2825020600      ; serial
                      12H            ; refresh
                      1H            ; retry
                      1W            ; expire
                      1H            ; ncache
)
          IN      NS     au-team.irpo.
          IN      A      192.168.100.62
hq-srv  IN      A      192.168.100.62
hq-rtr  IN      A      192.168.100.1
hq-rtr  IN      A      192.168.100.65
hq-rtr  IN      A      192.168.100.81
br-rtr  IN      A      192.168.3.1
hq-clli IN      A      192.168.100.66
br-srv  IN      A      192.168.3.30
noodle  IN      CNAME  hq-rtr.au-team.irpo.
wiki   IN      CNAME  hq-rtr.au-team.irpo.
_
```

Создадим файл 100.168.192.in-addr.irpo скопировав au-team.irpo

```
[root@hq-srv ~]# cp /etc/bind/zone/au-team.irpo /etc/bind/zone/100.168.192.in-addr.arpa
```

Внесем в него изменения

```
:
$TTL    1D
@      IN      SOA    au-team.irpo. root.au-team.irpo. (
                      2825020600      ; serial
                      12H            ; refresh
                      1H            ; retry
                      1W            ; expire
                      1H            ; ncache
)
          IN      NS     au-team.irpo.
62      IN      PTR    hq-srv.au-team.irpo.
1       IN      PTR    hq-rtr.au-team.irpo.
65      IN      PTR    hq-rtr.au-team.irpo.
81      IN      PTR    hq-rtr.au-team.irpo.
66      IN      PTR    hq-clli.au-team.irpo.
```

Посмотрим права на файлы

```
[root@hq-srv ~]# ls -la /etc/bind/zone/
total 36
drwxrwx--T 3 root named 4896 May 28 13:08 .
drwx--x--- 7 root named 4896 May 28 09:33 ..
-rw-r---- 1 root root 349 May 28 13:04 100.168.192.in-addr.arpa
-rw-r---- 1 root named 212 Jan 31 13:58 127.in-addr.arpa
-rw-r---- 1 root root 366 May 28 13:03 au-team.irpo
-rw-r---- 1 root named 389 Jan 31 13:58 empty
-rw-r---- 1 root named 288 Jan 31 13:58 localdomain
-rw-r---- 1 root named 178 Jan 31 13:58 localhost
drwx----- 2 root named 4896 Jan 31 13:58 slave
```

Группа владельца root – нужно поменять на named (служба bind)

```
[root@hq-srv ~]# chown root:named /etc/bind/zone/au-team.irpo
[root@hq-srv ~]# chown root:named /etc/bind/zone/100.168.192.in-addr.arpa
```

После того, как конфигурация зон была завершена, для корректной работы службы bind необходимо выполнить команду:

```
rndc-confgen > /etc/bind/rndc.key
```

```
[root@hq-srv ~]# vim /etc/bind/rndc.key
```

И оставим только первые 5 строк (у вас secret будет другой)

```
# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-sha256;
    secret "Vj0F+DfWOX7toBUI$9WONjxQbu/x+IM2TQEUIUKSoyEc=";
};
```

Изменим resolv.conf

```
[root@hq-srv ~]# vim /etc/net/ifaces/ens19/resolv.conf
```

```
nameserver 192.168.100.62
search au-team.irpo
```

Разрешаем запуск bind:

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

Проверим как работает – ошибок не должно быть

```
[root@isp ~]# ll /systemctl status bind.service
● bind.service - Berkeley Internet Name Service (BNS)
   Loaded: loaded (/lib/systemd/system/bind.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-05-20 13:18:35 MSK; 50s ago
     Process: 23581 ExecStartPre=/usr/libexec/named-checkconf --checkconf 99NBBT < /etc/named.conf (code=exited, status=0/SUCCESS)
    Process: 23580 ExecStart=/usr/sbin/named -u named SOHNBBT < /etc/named.conf SOHNBBT (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1149)
     Memory: 12.9M
        CPU: 79ms
       Group: named
      ▾ 23581 /usr/sbin/named -u named

May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:11::53#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::e#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::3#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::4#4#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::2#2#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::42#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: network unreachable resolving '.>IN' : 2001:500:12::#53
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: managed-keys-zone: initializing automatic trust anchor management for zone '.'; ENERGY ID 28326 is now
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: managed-keys-zone: initializing automatic trust anchor management for zone '.'; ENERGY ID 38676 is now
May 20 13:18:35 hq-srv.au-team.irpo named[23581]: resolver priming query complete
[...]
```

Проверяем работоспособность:

```
user@hq-cli ~ $ host hq-srv.au-team.irpo
hq-srv.au-team.irpo has address 192.168.100.62
user@hq-cli ~ $ host 192.168.100.62
62.100.168.192.in-addr.arpa domain name pointer hq-srv.au-team.irpo.
```

11. Настройте часовой пояс на всех устройствах, согласно месту проведения экзамена

Настройка часового пояса на Alt JeOS (ISP)

Установим пакет со списком часовых поясов

```
[root@isp ~]# apt-get install tzdata
```

Далее

```
[root@isp ~]# timedatectl set-timezone Europe/Moscow
```

Проверяем:

```
[root@isp ~]# timedatectl status
          Local time: Tue 2025-04-01 16:18:14 MSK
          Universal time: Tue 2025-04-01 13:18:14 UTC
                 RTC time: Tue 2025-04-01 13:18:14
                Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
          NTP service: active
        RTC in local TZ: no
```

Настройка часового пояса на Alt Linux

Меняем часовой пояс следующей командой:

```
[root@hq-srv ~]# timedatectl set-timezone Europe/Moscow
```

Проверяем:

```
[root@hq-srv ~]# timedatectl status
    Local time: Tue 2025-04-01 15:33:38 MSK
    Universal time: Tue 2025-04-01 12:33:38 UTC
        RTC time: Tue 2025-04-01 12:33:38
       Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
          NTP service: n/a
    RTC in local TZ: no
```

Настройка часового пояса на EcoRouter

Прописываем команду:

```
hq-rtr(config)#ntp timezone utc+3
```

Проверяем:

```
hq-rtr(config)#do show ntp timezone
System Time zone: Europe/Moscow
hq-rtr(config)#write
```

На HQ-CLI

