

Задание № 1

Базовая настройка

а) Настройте имена устройств согласно топологии

1. Используйте полное доменное имя

б) Сконфигурируйте адреса устройств на свое усмотрение.

1. Для офиса HQ выделена сеть 192.168.11.0/24

Решение

Базовая настройка

а) Настройте имена устройств согласно топологии

1. Используйте полное доменное имя

Смотрим имена устройств из таблицы:

ADMIN-HQ		
Название устройства	ОС	FQDN
R-HQ	EcoRouter	r-hq.au.team
SW1-HQ	Альт Сервер 10	sw1-hq.au.team
SW2-HQ	Альт Сервер 10	sw2-hq.au.team
SW3-HQ	Альт Сервер 10	sw3-hq.au.team
ADMIN-HQ	Альт Рабочая станция 10	admin-hq.au.team
SRV1-HQ	Альт Сервер 10	srv1-hq.au.team
CLI-HQ	Альт Рабочая станция 10	cli-hq.au.team

Для устройств на базе ОС "Альт Сервер":

SW1-HQ, SW2-HQ, SW3-HQ, SRV1-HQ:

Настраиваем имена устройств согласно топологии, используя полное доменное имя в качестве доменного имени используется - **au.team**

Общий вид команды:

```
hostnamectl set-hostname <ИМЯ_ВМ>.<ДОМЕННОЕ_ИМЯ>
exec bash
```

где:

hostnamectl — утилита для управления именем машины;

set-hostname — аргумент, позволяющий выполнить изменение хостнейма;

exec bash — перезапуск оболочки bash для отображения нового хостнейма.

Например:

Для ВМ SW1-HQ:

```
hostnamectl set-hostname sw1-hq.au.team
exec bash
```

Для ВМ SW2-HQ:

```
hostnamectl set-hostname sw2-hq.au.team
exec bash
```

Для ВМ SW3-HQ:

```
hostnamectl set-hostname sw3-hq.au.team
exec bash
```

Для ВМ SRV1-HQ:

```
hostnamectl set-hostname srv1-hq.au.team
exec bash
```

Проверяем:

Для проверки полного доменного имени используем утилиту `hostname` и добавляем ключ `-f`:

hostname -f

```
[root@srv1-hq ~]# hostname -f
srv1-hq.au.team
[root@srv1-hq ~]# _
```

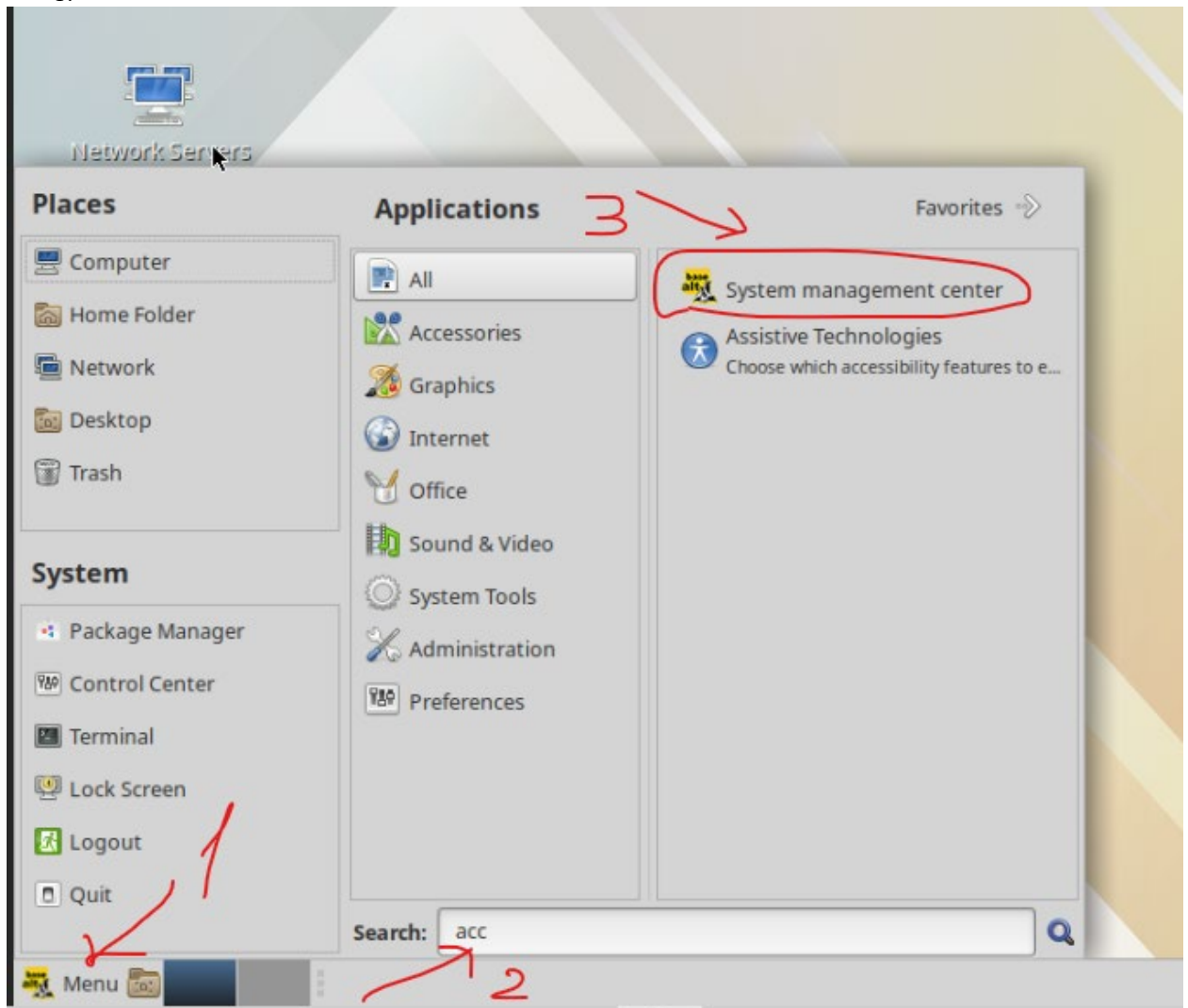
Для устройств на базе ОС "Альт" с графической оболочкой:
CLI-HQ, ADMIN-HQ:

Настраиваем имена устройств согласно топологии, используя полное доменное имя

- в качестве доменного имени используется - **au.team**
- поскольку клиент имеет графический интерфейс - воспользуемся **Центром Управления Системой (ЦУС):**

Можно вызвать Центр управления системой командой **acc**, введя ее в строку поиска, или вызвать ЦУС как на скриншоте ниже:

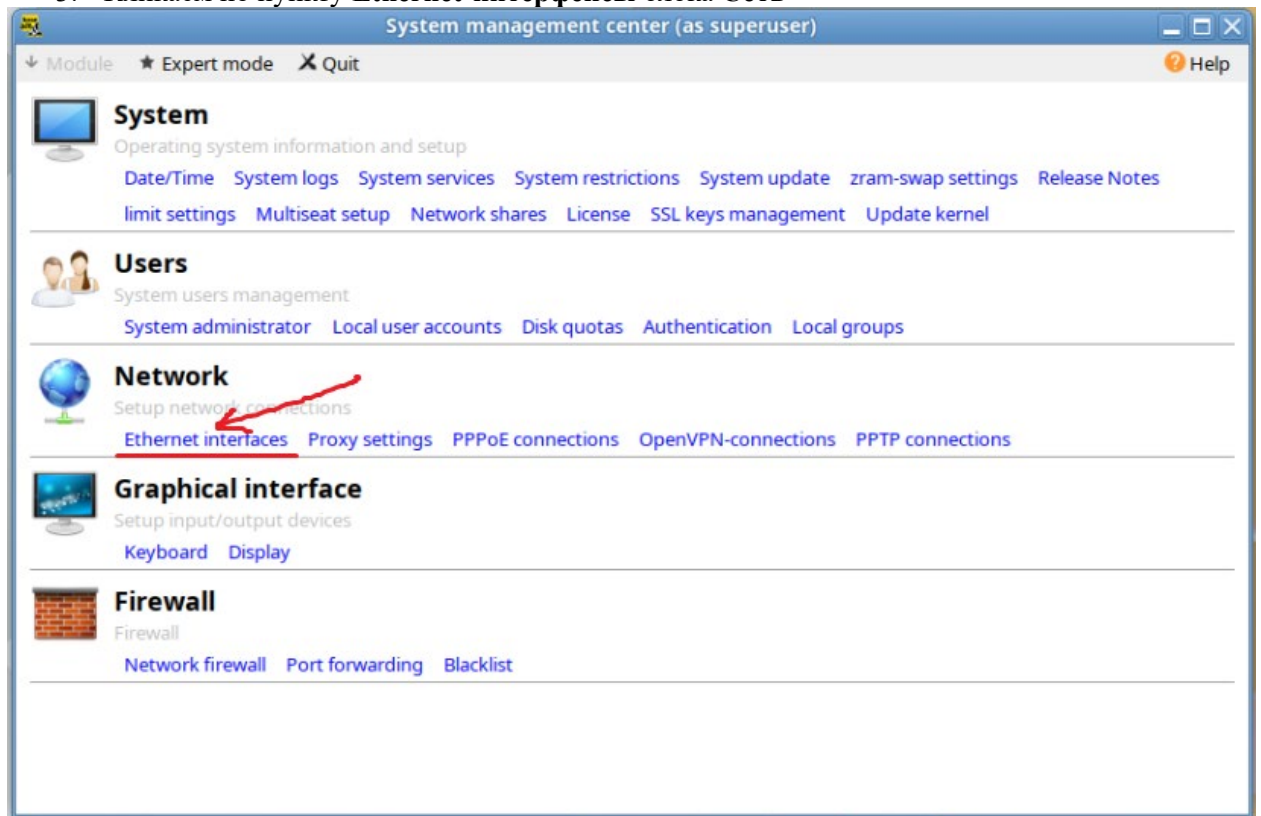
1.



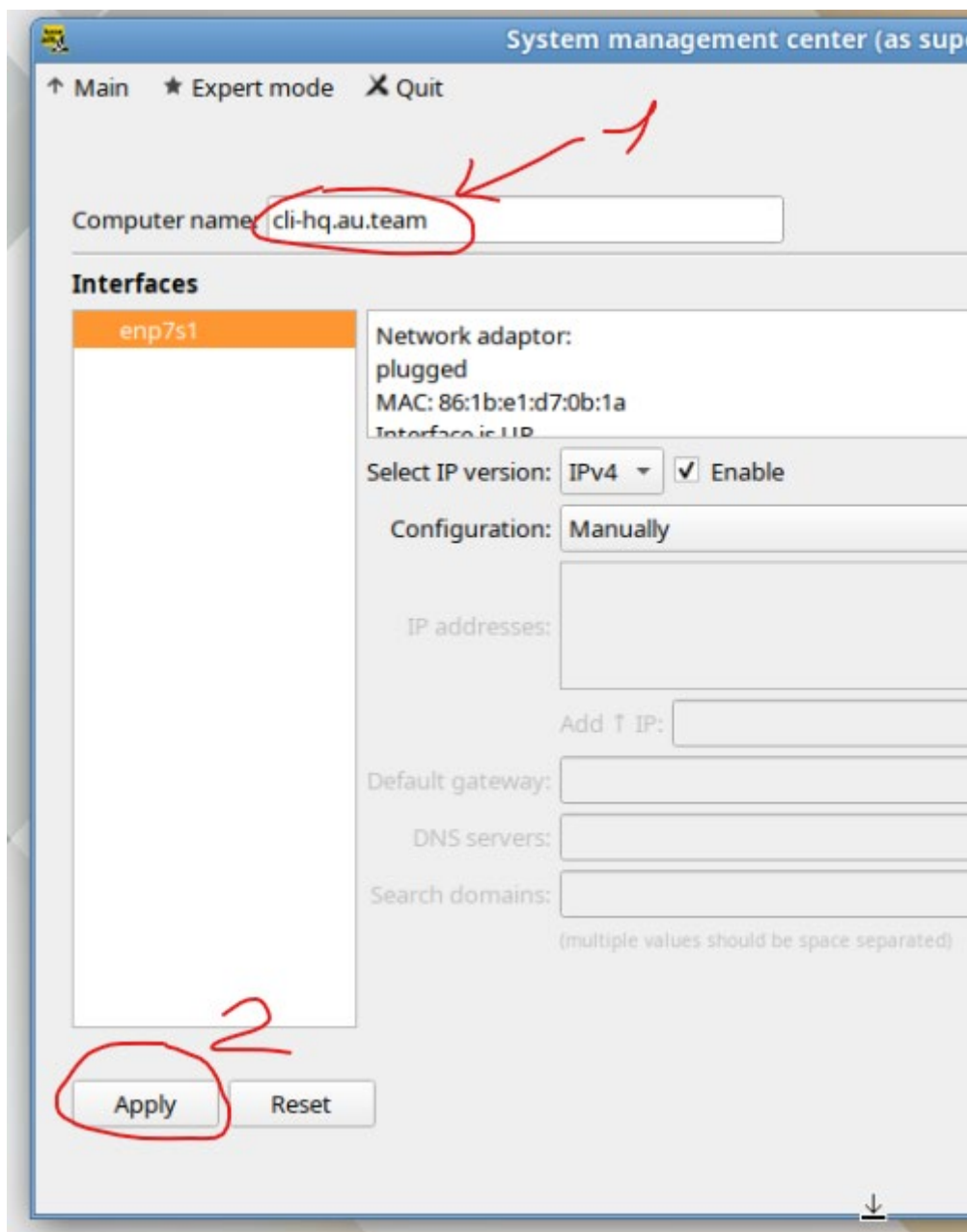
2. Вводим пароль от root



3. Кликаем по пункту **Ethernet-интерфейсы** блока **Сеть**



4. В поле **Имя компьютера** вводим новое имя и нажимаем кнопку **Применить**



5. **ПЕРЕЗАГРУЖАЕМ** виртуальную машину
6. **Аналогичным образом** меняем имена на других ВМ с графической оболочкой

Для устройств на базе "EcoRouter": R-HQ:

Настраиваем имена устройств согласно топологии, используя полное доменное имя в качестве доменного имени используется - **au.team**

1. Выполняем первый вход на устройство из под пользователя по умолчанию:
логин: **admin**
пароль: **admin**

```
ecorouter login: admin
Password:

User Access Verification

EcoRouterOS version Camellia
ecorouter>
```

Переходим в привилегированный режим (**enable**) и в режим администрирования (**configure**), затем назначаем имя устройства и доменное имя:

```
ecorouter>enable
ecorouter#configure terminal
ecorouter(config)#hostname r-hq.au.team
r-hq.au.team(config)#end
r-hq.au.team#write
Building configuration...
```

Задание № 2

б) Сконфигурируйте адреса устройств на свое усмотрение.

1. Для офиса HQ выделена сеть 192.168.11.0/24
2. Данные сети необходимо разделить на подсети для каждого vlan.
 - i. VLAN110 должна вмещать не более 64 адресов
 - ii. VLAN220 должна вмещать не более 16 адресов
 - iii. VLAN330 должна вмещать не более 8 адресов

Решение

Разбиение сетей на подсети:

Название устройства	NIC	Сеть (подсеть)	IP-адрес	Шлюз
R-HQ	ISP <-> R-HQ	172.16.5.0/28	172.16.5.14/28	172.16.5.1
	R-HQ <-> SW1-HQ (VLAN110)	192.168.11.0/26	192.168.11.1/26	-
	R-HQ <-> SW1-HQ (VLAN220)	192.168.11.64/28	192.168.11.65/28	
	R-HQ <-> SW1-HQ (VLAN330)	192.168.11.80/29	192.168.11.81/29	
SW1-HQ	ovs-internal (VLAN330)	192.168.11.80/29	192.168.11.83/29	192.168.11.81

SW2-HQ	ovs-internal (VLAN330)	192.168.11.80/29	192.168.11.84/29	192.168.11.81
SW3-HQ	ovs-internal (VLAN330)	192.168.11.80/29	192.168.11.85/29	192.168.11.81
ADMIN-HQ	SW3-HQ <-> ADMIN-HQ (VLAN330)	192.168.11.80/29	192.168.11.82/29	192.168.11.81
SRV1-HQ	SW2-HQ <-> SRV1-HQ (VLAN220)	192.168.11.64/28	192.168.11.66/28	192.168.11.65
CLI-HQ	SW2-HQ <-> CLI-HQ (VLAN110)	192.168.11.0/26	DHCP временно для проверки связности можно назначить адрес 192.168.11.2/26 Шлюз 192.168.11.1)	

1. Настройка адресации на **R-HQ**:

Смотрим наименование портов в системе (из привилегированного режима):

r-hq#show port brief

Результат:

порт te0 - смотрит в сторону ISP (интернет);

порт te1 - смотрит в сторону локальной сети офиса HQ.

```
r-hq.au.team#sh port bri
Name          Physical    Admin    LACP    Description
-----
te0            UP          UP        *
te1            UP          UP        *
```

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.

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.

Создаем интерфейс с именем **isp** с последующим назначаем IPv4-адреса согласно таблице адресации в сторону ISP (провайдера интернета):

```
r-hq#configure terminal
r-hq(config)#interface isp
r-hq(config-if)#ip address 172.16.5.14/28
r-hq(config-if)#exit
r-hq(config)#
Задаём IP-адрес шлюза по умолчанию:
r-hq(config)#ip route 0.0.0.0/0 172.16.5.1
r-hq(config)#write
```

переходим в режим конфигурирования порта **te0** - создаём **service-instance** с именем **isp**: также указываем что будет обрабатывать не тегированный трафик (**untagged**); и указываем в какой интерфейс нужно отправить обработанные кадры (**isp**);

```
r-hq(config)#port te0
r-hq(config-port)#service-instance isp
r-hq(config-service-instance)#encapsulation untagged
r-hq(config-service-instance)#connect ip interface isp
r-hq(config-service-instance)#end
r-hq#write
```

Создаём интерфейсы (подинтерфейсы/sub-interfaces) для назначения адресов локальных подсетей офиса HQ для дальнейшей маршрутизации между VLAN-ами:

Первым делом создаём интерфейсы для каждого VLAN-а:

```
r-hq#configure terminal
r-hq(config)#interface vl110
r-hq(config-if)#ip address 192.168.11.1/26
r-hq(config-if)#exit
r-hq(config)#interface vl220
r-hq(config-if)#ip address 192.168.11.65/28
r-hq(config-if)#exit
r-hq(config)#interface vl330
r-hq(config-if)#ip address 192.168.11.81/29
r-hq(config-if)#exit
r-hq(config)#write
```

На базе физического интерфейса **te1** - для каждого VLAN-а создаём **service-instance** с инкапсуляцией соответствующих тегов (VID) и подключением необходимых интерфейсов:

```
r-hq(config)#port te1
r-hq(config-port)#service-instance te1/vl110
r-hq(config-service-instance)#encapsulation dot1q 110 exact
r-hq(config-service-instance)#rewrite pop 1
r-hq(config-service-instance)#connect ip interface vl110
r-hq(config-service-instance)#exit
r-hq(config-port)#service-instance te1/vl220
```

```

r-hq(config-service-instance)#encapsulation dot1q 220 exact
r-hq(config-service-instance)#rewrite pop 1
r-hq(config-service-instance)#connect ip interface vl220
r-hq(config-service-instance)#exit
r-hq(config-port)#service-instance te1/vl330
r-hq(config-service-instance)#encapsulation dot1q 330 exact
r-hq(config-service-instance)#rewrite pop 1
r-hq(config-service-instance)#connect ip interface vl330
r-hq(config-service-instance)#exit
r-hq(config-port)#write

```

Проверяем

```

r-hq.au.team#sh ip interface brief

```

Interface	IP-Address	Status	VRP
isp	172.16.5.14/28	up	default
vl110	192.168.11.1/26	up	default
vl220	192.168.11.65/28	up	default
vl330	192.168.11.81/29	up	default

```

r-hq.au.team#_

```

SRV1-HQ:

В качестве сетевой подсистемы будет использоваться Etcnet

Первым делом смотрим название сетевого интерфейса командой

ip -c a

```

[root@srv1-hq ~]# ip -c 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: enp7s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether f6:04:f6:b8:b0:f2 brd ff:ff:ff:ff:ff:ff
[root@srv1-hq ~]#

```

На скриншоте выше видим, что название сетевого интерфейса **enp7s1** и он в состоянии **DOWN**. Если сетевой интерфейс находится в состоянии **DOWN**, то необходимо

- создать каталог командой: **mkdir /etc/net/ifaces/enp7s1**
- изменить настройки файла: **vim /etc/net/ifaces/ens18/options**
- скопировать файл **options** командой
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s1

Содержимое файла должно быть как на скриншоте ниже.

```

TYPE=eth
CONFIG_WIRELESS=no
BOOTPROTO=static
SYSTEMD_BOOTPROTO=static
CONFIG_IPV4=yes
DISABLED=no
NM_CONTROLLED=no
SYSTEMD_CONTROLLED=no

```


Для того, чтобы в качестве сетевой подсистемы корректно использовался `etcnet`, в основном конфигурационном файле для интерфейса `/etc/net/ifaces/<INTERFACE_NAME>/options` должны присутствовать два параметра со следующими значениями:

`DISABLED=no`

`NM_CONTROLLED=no`

в данном случае сетевой интерфейс имеет имя `enp7s1`;

также стоит обратить внимание, чтобы назначить статические сетевые параметры для данного интерфейса - параметр `BOOTPROTO` должен иметь значение `static`

Далее назначим IP-адрес на интерфейс согласно таблице адресации и IP-адрес шлюза по умолчанию:

Назначаем IP-адрес из `vlan220` (серверная подсеть) на интерфейс `enp7s1` (в данном случае):

`echo 192.168.11.66/28 > /etc/net/ifaces/enp7s1/ipv4address`

Назначаем IP-адрес шлюза по умолчанию для `vlan220` (серверная подсеть):

`echo default via 192.168.11.65 > /etc/net/ifaces/enp7s1/ipv4route`

Для применения внесённых изменений - необходимо перезагрузить службу `network`:

`systemctl restart network`

Проверяем:

Наличие IPv4-адреса и адреса шлюза по умолчанию проверяем командой

`ip -c a`

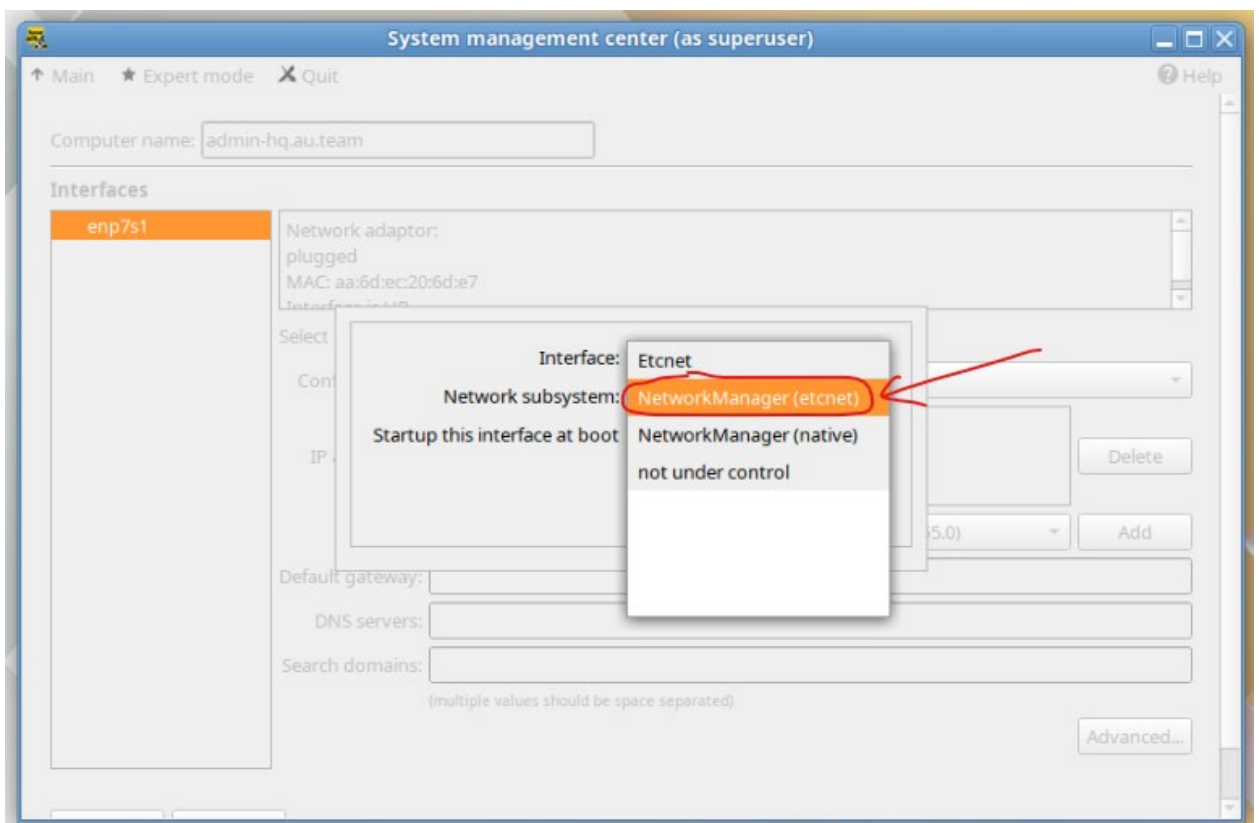
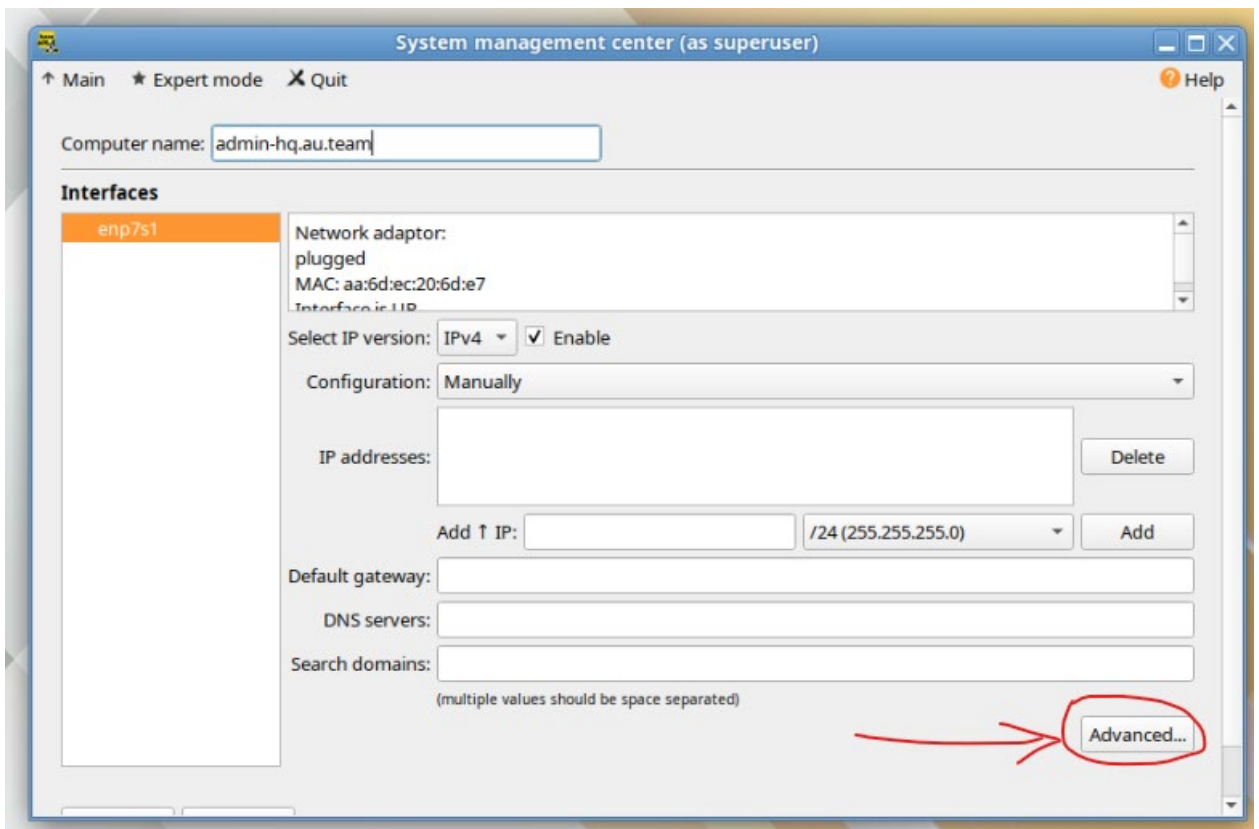
```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp7s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 46:d9:7d:22:d6:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.66/28 brd 192.168.11.79 scope global enp7s1
        valid_lft forever preferred_lft forever
    inet6 fe80::44d9:7dff:fe22:d6bb/64 scope link
        valid_lft forever preferred_lft forever
[root@srv1-hq ~]# ip -c r
default via 192.168.11.65 dev enp7s1
192.168.11.64/28 dev enp7s1 proto kernel scope link src 192.168.11.66
[root@srv1-hq ~]#
```

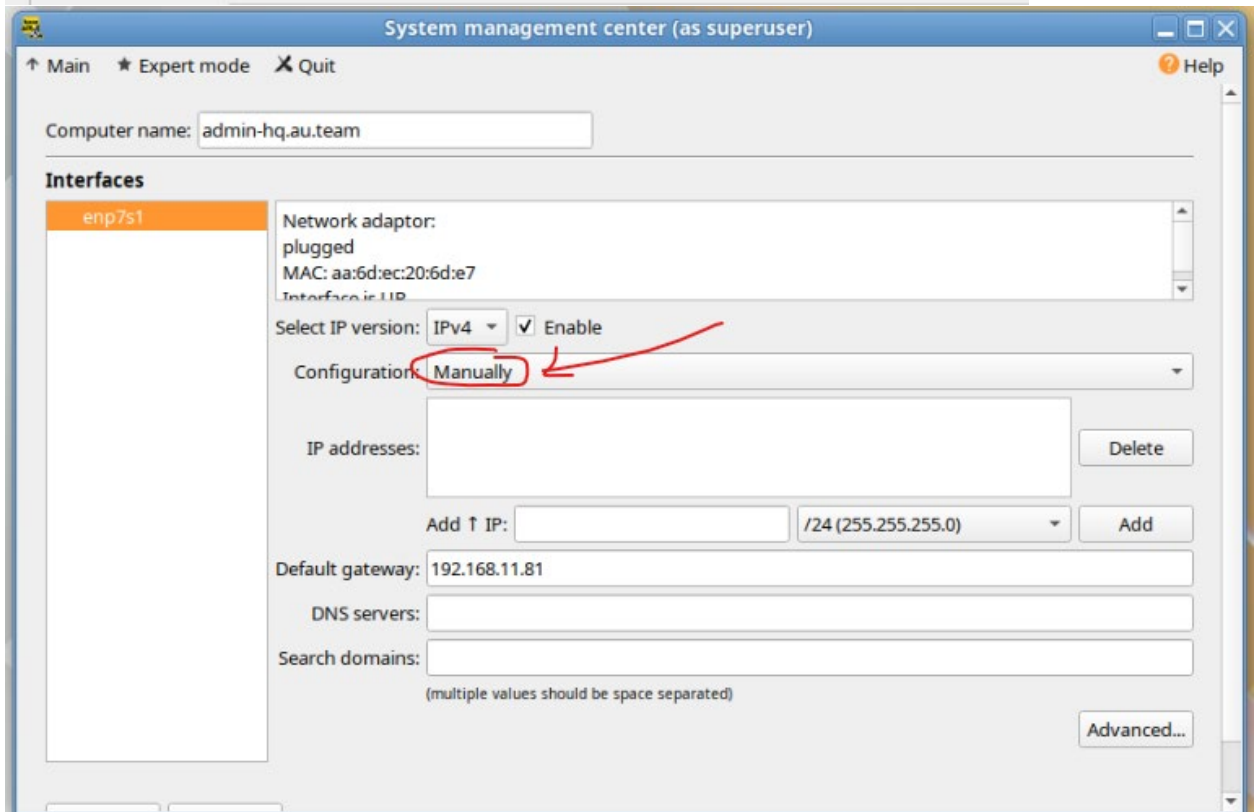
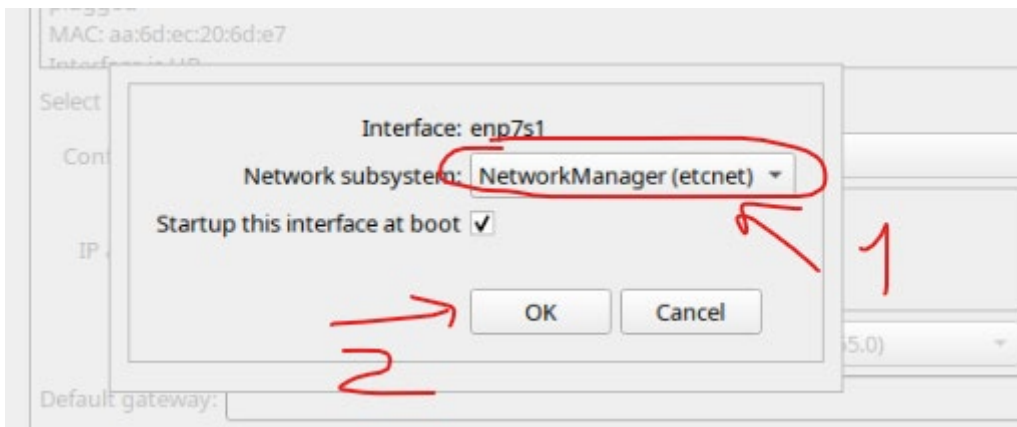
На данный момент - доступность шлюза по умолчанию не проверить, т.к. не реализована коммутация.

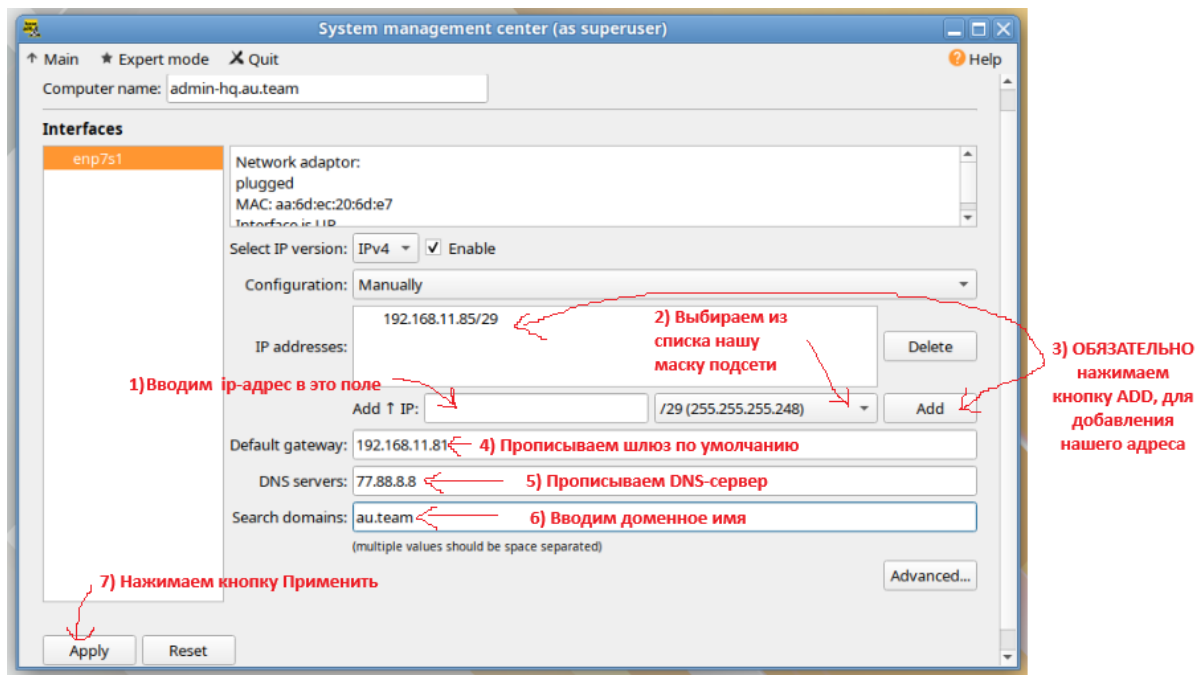
ADMIN-HQ:

Пользуясь средствами графического интерфейса - назначаем IP-адрес на интерфейс согласно таблице адресации и IP-адрес шлюза по умолчанию.

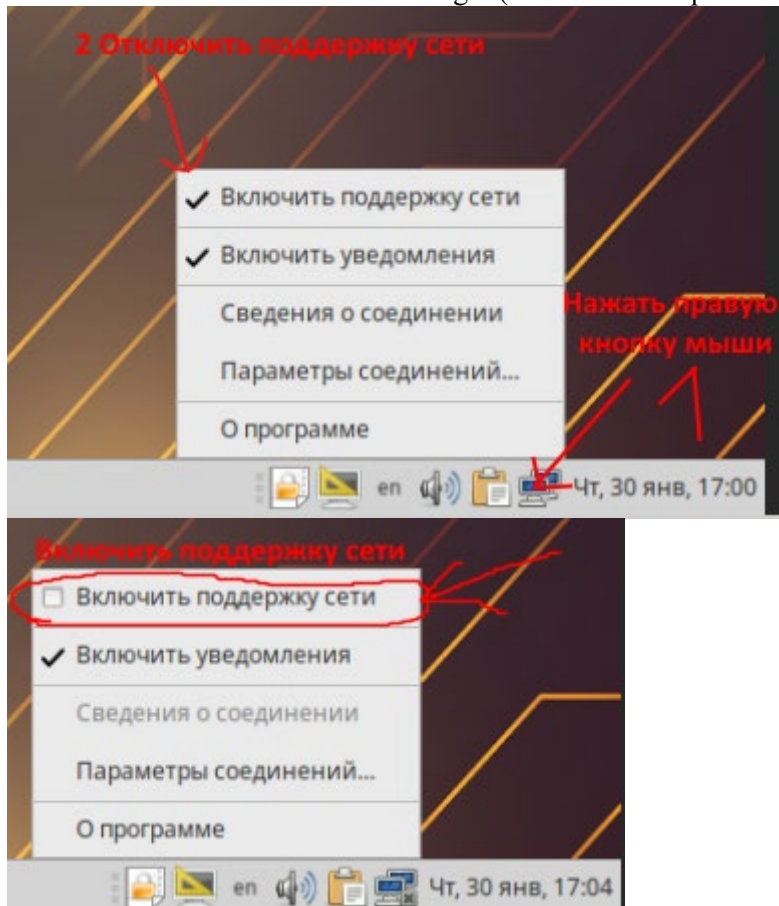
Переходим в Альтератор (Центр Управления Системой - ЦУС) - там же где назначали имя, задаём необходимые сетевые параметры:







Для того чтобы изменения вступили в силу, необходимо или перезагрузить ВМ или выключить/включить NetworkManager (показано на скриншоте)



Проверка

```

[root@admin-hq ~]# ip -c 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: enp7s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether aa:6d:ec:20:6d:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.85/29 brd 192.168.11.87 scope global noprefixroute enp7s1
        valid_lft forever preferred_lft forever
[root@admin-hq ~]# ip -c r
default via 192.168.11.81 dev enp7s1 proto static metric 100
192.168.11.80/29 dev enp7s1 proto kernel scope link src 192.168.11.85 metric 100
[root@admin-hq ~]#

```

Задание № 4

- Настройте на маршрутизаторах динамическую трансляцию адресов.
- Все устройства во всех офисах должны иметь доступ к сети Интернет

R-HQ:

С точки зрения EcoRouter - реализуем конфигурацию static source PAT:
Интерфейс в сторону ISP с именем isp - назначаем как nat outside:

r-hq#configure terminal

r-hq(config)#interface isp

r-hq(config-if)#ip nat outside

r-hq(config-if)#exit

r-hq(config)#

Подинтерфейсы vl110, vl220, vl330, которые смотрят в сторону SW1-HQ - назначаем как nat inside:

r-hq(config)#interface vl110

r-hq(config-if)#ip nat inside

r-hq(config-if)#exit

r-hq(config)#

r-hq(config)#interface vl220

r-hq(config-if)#ip nat inside

r-hq(config-if)#exit

r-hq(config)#

r-hq(config)#interface vl330

r-hq(config-if)#ip nat inside

r-hq(config-if)#exit

r-hq(config)#

Создаём пул адресов с именем NAT для входящего трафика - указываем диапазон адресов из выделенной подсети:

r-hq(config)#ip nat pool NAT 192.168.11.1-192.168.11.254

```

ip nat pool NAT 192.168.11.1-192.168.11.254

```

задаём правила для трансляции адресов:

r-hq(config)#ip nat source dynamic inside pool NAT overload 172.16.5.14

r-hq(config)#write

r-hq(config)#

```
?
interface isp
ip mtu 1500
connect port ge0 service-instance isp
ip nat outside
ip address 172.16.5.14/28
?
interface vl110
ip mtu 1500
connect port ge1 service-instance ge1/vl110
dhcp-server 1
ip nat inside
ip address 192.168.11.1/26
?
interface vl220
ip mtu 1500
connect port ge1 service-instance ge1/vl220
ip nat inside
ip address 192.168.11.65/28
?
interface vl330
ip mtu 1500
connect port ge1 service-instance ge1/vl330
ip nat inside
ip address 192.168.11.81/29
?
ip nat pool NAT 192.168.11.1-192.168.11.254
?
ip nat source dynamic inside-to-outside pool NAT overload 172.16.5.14
?
```

Задание № 5

а) Настройте коммутаторы SW1-HQ, SW2-HQ, SW3-HQ.

1. Используйте Open vSwitch

2. Имя коммутатора должно совпадать с коротким именем устройства

і. Используйте заглавные буквы

3. Передайте все физические порты коммутатору.

4. Обеспечьте включение портов, если это необходимо

с) Для каждого офиса устройства должны находиться в соответствующих VLAN

1. Клиенты - vlan110,

2. Сервера – в vlan220,

3. Администраторы – в vlan330.

SW1-HQ:

Включаем и добавляем в автозагрузку openvswitch:

systemctl enable --now openvswitch

Проверяем интерфейсы и определяемся какой к кому направлен:
таким образом, имеем:

enp7s1 - интерфейс в сторону R-HQ;

enp7s2 - интерфейс в сторону SW2-HQ;

enp7s3 - интерфейс в сторону SW3-HQ.


```
[root@sw1-hq ~]# ip -c -br a
lo UNKNOWN 127.0.0.1/8 ::1/128
enp7s1 DOWN fe80::9c19:82ff:fec8:9fb8/64
enp7s2 DOWN
enp7s3 DOWN
[root@sw1-hq ~]#
```

Обеспечим включение портов **enp7s1**, **enp7s2** и **enp7s3**:

Создадим для них одноимённые директории по пути `/etc/net/ifaces/<ИМЯ_ИНТЕРФЕЙСА>`:

```
mkdir /etc/net/ifaces/enp7s{1,2,3}
```

Для каждого интерфейса необходимо создать конфигурационный файл `options`:

Файл `options` для интерфейсов `enp7s1`, `enp7s2` и `enp7s3` будет аналогичен, как и для `ens18` - поэтому его можно просто скопировать:

```
vim /etc/net/ifaces/ens18/options
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s1/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s2/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s3/
```

Перезагружаем службу `network` для применения изменений:

```
systemctl restart network
```

Проверяем что все интерфейсы перешли в состояние UP:

```
[root@sw1-hq ~]# ip -c 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: enp7s1 <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 96:df:80:57:49:7e brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet6 fe80::94df:80ff:fe57:497e/64 scope link
        valid_lft forever preferred_lft forever
3: enp7s2 <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 7a:93:c1:d7:06:70 brd ff:ff:ff:ff:ff:ff
    altname enp0s20
    inet6 fe80::7093:c1ff:fe47:0670/64 scope link
        valid_lft forever preferred_lft forever
4: enp7s3 <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 6e:95:92:6d:73:7f brd ff:ff:ff:ff:ff:ff
    altname enp0s21
    inet6 fe80::6e95:92ff:fe6d:737f/64 scope link
        valid_lft forever preferred_lft forever
```

Создадим коммутатор имя которого должно совпадать с коротким именем устройства и с использованием заглавных букв - **SW1-HQ**:

```
ovs-vsctl add-br SW1-HQ
```

Проверяем:

```
[root@sw1-hq ~]# ovs-vsctl show
552adc43-3893-40c9-8656-6ef0f7e6c890
    Bridge SW1-HQ
        Port SW1-HQ
            Interface SW1-HQ
                type: internal
            ovs_version: "2.17.11"
[root@sw1-hq ~]# _
```

Сетевая подсистема etcnnet будет взаимодействовать с openvswitch, для того чтобы корректно можно было назначить IP-адрес на интерфейс управления

Создаём каталог для management интерфейса с именем MGMT:

mkdir /etc/net/ifaces/MGMT

Описываем файл options для создания management интерфейса с именем mgmt:

vim /etc/net/ifaces/MGMT/options

Содержимое, где:

TYPE - тип интерфейса (internal);

BOOTPROTO - определяет как будут назначаться сетевые параметры (статически);

CONFIG_IPV4 - определяет использовать конфигурацию протокола IPv4 или нет;

BRIDGE - определяет к какому мосту необходимо добавить данный интерфейс;

VID - определяет принадлежность интерфейса к VLAN;

```
TYPE=ovsport
BOOTPROTO=static
CONFIG_IPV4=yes
BRIDGE=SW1-HQ
VID=330
[root@sw1-hq ~]# _
```

Назначаем IP-адрес и шлюз на созданный интерфейс MGMT согласно таблице адресации для Администраторской подсети:

echo 192.168.11.83/29 > /etc/net/ifaces/MGMT/ipv4address

echo default via 192.168.11.81 > /etc/net/ifaces/MGMT/ipv4route

Правим основной файл options в котором по умолчанию сказано удалять настройки заданные через ovs-vsctl, т.к. через etcnnet будет выполнено только создание bridge и интерфейса типа internal с назначением необходимого IP-адреса, а настройка функционала будет выполнена средствами openvswitch:

1 вариант

sed -i "s/OVS_REMOVE=yes/OVS_REMOVE=no/g" /etc/net/ifaces/default/options

2 вариант

```
[root@sw1-hq ~]# vim /etc/net/ifaces/default/options_
```



```
# This file doesn't contain comments
# manpage for detailed options

DISABLED=no
BOOTPROTO=static
ONBOOT=yes
USE_HOTPLUG=no
USE_PCMCIA=no
CONFIG_IPV4=yes
CONFIG_IPV6=no
CONFIG_IPX=no
CONFIG_QOS=yes
CONFIG_WIRELESS=no
CONFIG_FW=no
KEEP_DOWN=no
DONT_FLUSH=no
IFUP_CHILDREN=no
IFUP_PARENTS=yes
IFDOWN_CHILDREN=yes
IFDOWN_PARENTS=no
OVS_REMOVE=no
DHCP_TIMEOUT=60
```

Перезапускаем службу network:

systemctl restart network

Проверяем:

На текущий момент создан интерфейс управления и назначен IP-адрес из соответствующей подсети. Данный интерфейс управления помечен тегом (VID) 330 и добавлен в bridge SW1-HQ.

```
[root@sw1-hq ~]# ip -c 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: enp7s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 96:df:80:57:49:7e brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet6 fe80::96df:80ff:fe57:497e/64 scope link
        valid_lft forever preferred_lft forever
3: enp7s2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 7a:93:c1:d7:06:70 brd ff:ff:ff:ff:ff:ff
    altname enp0s20
    inet6 fe80::7a93:c1ff:fe70:670/64 scope link
        valid_lft forever preferred_lft forever
4: enp7s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
    link/ether 6e:95:92:6d:73:7f brd ff:ff:ff:ff:ff:ff
    altname enp0s21
    inet6 fe80::6e95:92ff:fe6d:737f/64 scope link
        valid_lft forever preferred_lft forever
5: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:92:4c:11:79:61 brd ff:ff:ff:ff:ff:ff
6: SW1-HQ: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:fb:8c:0a:95:41 brd ff:ff:ff:ff:ff:ff
10: mgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether de:2a:fd:2a:90:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.83/29 scope global mgmt
        valid_lft forever preferred_lft forever
    inet6 fe80::dc2a:fdff:fe2a:9032/64 scope link
        valid_lft forever preferred_lft forever
[root@sw1-hq ~]# ip -c r
default via 192.168.11.81 dev mgmt
192.168.11.00/29 dev mgmt proto kernel scope link src 192.168.11.83
[root@sw1-hq ~]#
```

Средствами openvswitch настраиваем следующий функционал:

Порт в сторону маршрутизатора и в сторону остальных коммутаторов должны быть магистральными и пропускать используемые VLAN-ы:

ovs-vsctl add-port SW1-HQ enp7s1

ovs-vsctl add-port SW1-HQ enp7s2

```
ovs-vsctl add-port SW1-HQ enp7s3
ovs-vsctl set port enp7s1 trunk=110,220,330
ovs-vsctl set port enp7s2 trunk=110,220,330
ovs-vsctl set port enp7s3 trunk=110,220,330
```

Включаем модуль ядра 8021q:

```
modprobe 8021q
```

При необходимости добавляем и на постоянной основе:

```
Echo "8021q" | tee -a /etc/modules
```

Проверяем

A terminal window screenshot showing the output of the 'ovs-vsctl show' command. The output lists details for Bridge SW1-HQ, including its MAC address, management port, and three data ports (enp7s1, enp7s2, enp7s3). Each data port is configured as a trunk with VLANs 110, 220, and 330, connected to interfaces ens19, ens21, and ens20 respectively. The OVS version is 2.17.11.

```
[root@sw1-hq ~]# ovs-vsctl show
a3bce80f-a259-408b-a043-db00a77bc8dc
Bridge SW1-HQ
  Port mgmt
    tag: 330
    Interface mgmt
      type: internal
  Port enp7s1
    trunks: [110, 220, 330]
    Interface ens19
  Port SW1-HQ
    Interface SW1-HQ
      type: internal
  Port enp7s2
    trunks: [110, 220, 330]
    Interface ens21
  Port enp7s3
    trunks: [110, 220, 330]
    Interface ens20
  ovs_version: "2.17.11"
[root@sw1-hq ~]#
```

SW2-HQ:

Включаем и добавляем в автозагрузку openvswitch:

```
systemctl enable --now openvswitch
```

Проверяем интерфейсы и определяемся какой к кому направлен:

Таким образом, имеем:

enp7s1 - интерфейс в сторону SW1-HQ

enp7s2 - интерфейс в сторону SW3-HQ

enp7s3 - интерфейс в сторону SRV1-HQ

enp7s4 - интерфейс в сторону CLI-HQ

```
[root@sw2-hq ~]# ip -c -br a
lo UNKNOWN
enp7s1 DOWN
enp7s2 DOWN
enp7s3 DOWN
enp7s4 DOWN
[root@sw2-hq ~]# _
```

Обеспечим включение портов **enp7s1**, **enp7s2**, **enp7s3** и **enp7s4**:

Создадим для них одноимённые директории по пути `/etc/net/ifaces/<ИМЯ_ИНТЕРФЕЙСА>`:

```
mkdir /etc/net/ifaces/enp7s{1,2,3,4}
```

Файл `options` для интерфейсов **enp7s1**, **enp7s2**, **enp7s3** и **enp7s4** будет аналогичен, как и для **ens18** - поэтому его можно просто скопировать:

ВАЖНО, чтобы для **ens18** в файле `options` параметр **BOOTPROTO** имел значение **static**

```
vim /etc/net/ifaces/ens18/options
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s1/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s2/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s3/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s4/
```

Перезагружаем службу `network` для применения изменений:

```
systemctl restart network
```

Проверяем что все интерфейсы перешли в состояние UP:

```
[root@sw2-hq ~]# ip -c -br a
lo UNKNOWN
enp7s1 UP
enp7s2 UP
enp7s3 UP
enp7s4 UP
[root@sw2-hq ~]# _
```

Создадим коммутатор имя которого должно совпадать с коротким именем устройства с использованием заглавных букв - **SW2-HQ**:

```
ovs-vsctl add-br SW2-HQ
```

```
[root@sw2-hq ~]# ovs-vsctl show
637f9329-1271-4366-b078-0bd7396afcd5
    Bridge SW2-HQ
        Port SW2-HQ
            Interface SW2-HQ
                type: internal
            ovs_version: "2.17.11"
[root@sw2-hq ~]#
```

Сетевая подсистема etcnets будет взаимодействовать с openvswitch, для того чтобы корректно можно было назначить IP-адрес на интерфейс управления
Создаём каталог для management интерфейса с именем MGMT:

mkdir /etc/net/ifaces/MGMT

Описываем файл options для создания management интерфейса с именем mgmt:

vim /etc/net/ifaces/MGMT/options

Содержимое, где:

TYPE - тип интерфейса (internal);

BOOTPROTO - определяет как будут назначаться сетевые параметры (статически);

CONFIG_IPV4 - определяет использовать конфигурацию протокола IPv4 или нет;

BRIDGE - определяет к какому мосту необходимо добавить данный интерфейс;

VID - определяет принадлежность интерфейса к VLAN;

```
TYPE=ovsport
BOOTPROTO=static
CONFIG_IPV4=yes
BRIDGE=SW2-HQ
VID=330
```

Назначаем IP-адрес и шлюз на созданный интерфейс MGMT согласно таблице адресации для Администраторской подсети:

echo 192.168.11.84/29 > /etc/net/ifaces/MGMT/ipv4address

echo default via 192.168.11.81 > /etc/net/ifaces/MGMT/ipv4route

Правим основной файл options в котором по умолчанию сказано удалять настройки заданные через ovs-vsctl, т.к. через etcnets будет выполнено только создание bridge и интерфейса типа internal с назначением необходимого IP-адреса, а настройка функционала будет выполнена средствами openvswitch:

sed -i "s/OVS_REMOVE=yes/OVS_REMOVE=no/g" /etc/net/ifaces/default/options

или вторым способом, описанным при настройке SW1-HQ

Перезапускаем службу network:

systemctl restart network

Средствами openvswitch настраиваем следующий функционал:

Порт в сторону маршрутизатора и в сторону остальных коммутаторов должны быть магистральными и пропускать использующиеся VLAN-ы:

```
ovs-vsctl add-port SW2-HQ enp7s1
ovs-vsctl add-port SW2-HQ enp7s2
ovs-vsctl add-port SW2-HQ enp7s3
ovs-vsctl add-port SW2-HQ enp7s4
```

```
ovs-vsctl set port enp7s1 trunk=110,220,330
ovs-vsctl set port enp7s2 trunk=110,220,330
ovs-vsctl set port enp7s3 tag=220
ovs-vsctl set port enp7s4 tag=110
```

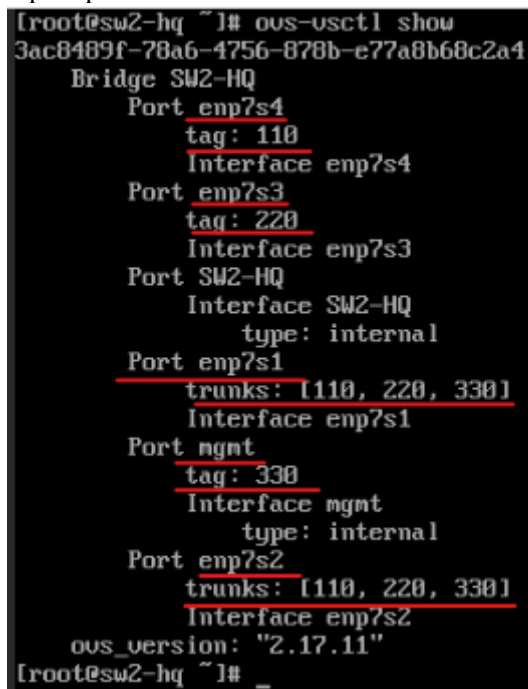
Включаем модуль ядра 8021q:

```
modprobe 8021q
```

При необходимости добавляем и на постоянной основе:

```
echo "8021q" | tee -a /etc/modules
```

Проверяем



```
[root@sw2-hq ~]# ovs-vsctl show
3ac8489f-78a6-4756-878b-e77a8b68c2a4
    Bridge SW2-HQ
        Port enp7s4
            tag: 110
            Interface enp7s4
        Port enp7s3
            tag: 220
            Interface enp7s3
        Port SW2-HQ
            Interface SW2-HQ
            type: internal
        Port enp7s1
            trunks: [110, 220, 330]
            Interface enp7s1
        Port ngmt
            tag: 330
            Interface ngmt
            type: internal
        Port enp7s2
            trunks: [110, 220, 330]
            Interface enp7s2
    ovs_version: "2.17.11"
[root@sw2-hq ~]# _
```

SW3-HQ:

Включаем и добавляем в автозагрузку openvswitch:

```
systemctl enable --now openvswitch
```

Проверяем интерфейсы и определяемся какой к кому направлен:

Таким образом, имеем:

- **enp7s1** - интерфейс в сторону SW1-HQ
- **enp7s2** - интерфейс в сторону SW2-HQ
- **enp7s3** - интерфейс в сторону ADMIN-HQ

```
[root@sw3-hq ~]# ip -c -br a
lo UNKNOWN
enp7s1 DOWN
enp7s2 DOWN
enp7s3 DOWN
[root@sw3-hq ~]# _
```

Обеспечим включение портов **enp7s1**, **enp7s2** и **enp7s3**:

Создадим для них одноимённые директории по пути `/etc/net/ifaces/<ИМЯ_ИНТЕРФЕЙСА>`:

```
mkdir /etc/net/ifaces/enp7s{1,2,3}
```

Файл `options` для интерфейсов **enp7s1**, **enp7s2** и **enp7s3** будет аналогичен, как и для **ens18** - поэтому его можно просто скопировать:

ВАЖНО, чтобы для **ens18** в файле `options` параметр **BOOTPROTO** имел значение **static**

```
vim /etc/net/ifaces/ens18/options
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s1/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s2/
```

```
cp /etc/net/ifaces/ens18/options /etc/net/ifaces/enp7s3/
```

Перезагружаем службу `network` для применения изменений:

```
systemctl restart network
```

Проверяем что все интерфейсы перешли в состояние UP:

```
[root@sw3-hq ~]# ip -c -br a
lo UNKNOWN
enp7s1 UP
enp7s2 UP
enp7s3 UP
[root@sw3-hq ~]# _
```

Создадим коммутатор имя которого должно совпадать с коротким именем устройства с использованием заглавных букв - **SW3-HQ**:

```
ovs-vsctl add-br SW3-HQ
```

Проверяем:

```
[root@sw3-hq ~]# ovs-vsctl show ←
d451dfff-3760-49fa-b09a-1f79150ccf9e
    Bridge SW3-HQ
        Port SW3-HQ
            Interface SW3-HQ
                type: internal
            ovs_version: "2.17.11"
[root@sw3-hq ~]#
```

Сетевая подсистема etcnnet будет взаимодействовать с openvswitch, для того чтобы корректно можно было назначить IP-адрес на интерфейс управления
Создаём каталог для management интерфейса с именем MGMT:

mkdir /etc/net/ifaces/MGMT

Описываем файл options для создания management интерфейса с именем mgmt:

vim /etc/net/ifaces/MGMT/options

Содержимое, где:

TYPE - тип интерфейса (internal);

BOOTPROTO - определяет как будут назначаться сетевые параметры (статически);

CONFIG_IPV4 - определяет использовать конфигурацию протокола IPv4 или нет;

BRIDGE - определяет к какому мосту необходимо добавить данный интерфейс;

VID - определяет принадлежность интерфейса к VLAN;

```
TYPE=ovsport
BOOTPROTO=static
CONFIG_IPV4=yes
BRIDGE=SW3-HQ
VID=330
```

Назначаем IP-адрес и шлюз на созданный интерфейс MGMT согласно таблице адресации для Администраторской подсети:

echo 192.168.11.85/29 > /etc/net/ifaces/MGMT/ipv4address

echo default via 192.168.11.81 > /etc/net/ifaces/MGMT/ipv4route

Правим основной файл options в котором по умолчанию сказано удалять настройки заданные через ovs-vsctl, т.к. через etcnnet будет выполнено только создание bridge и интерфейса типа internal с назначением необходимого IP-адреса, а настройка функционала будет выполнена средствами openvswitch:

sed -i "s/OVS_REMOVE=yes/OVS_REMOVE=no/g" /etc/net/ifaces/default/options

Перезапускаем службу network:

systemctl restart network

Средствами openvswitch настраиваем следующий функционал:

Порт в сторону маршрутизатора и в сторону остальных коммутаторов должны быть магистральными и пропускать использующиеся VLAN-ы:

ovs-vsctl add-port SW3-HQ enp7s1

ovs-vsctl add-port SW3-HQ enp7s2

ovs-vsctl add-port SW3-HQ enp7s3

ovs-vsctl set port enp7s1 trunk=110,220,330

ovs-vsctl set port enp7s2 trunk=110,220,330

ovs-vsctl set port enp7s3 tag=330

Включаем модуль ядра 8021q:

modprobe 8021q

При необходимости добавляем и на постоянной основе:

```
Echo "8021q" | tee -a /etc/modules
```

Проверяем

```
[root@sw3-hq ~]# ovs-vsctl show
3a6d3498-eca5-4716-b8ca-78c5f50b29b1
    Bridge SW3-HQ
        Port enp7s3
            tag: 330
            Interface enp7s3
        Port SW3-HQ
            Interface SW3-HQ
            type: internal
        Port enp7s1
            trunks: [110, 220, 330]
            Interface enp7s1
        Port enp7s2
            trunks: [110, 220, 330]
            Interface enp7s2
        Port mgmt
            tag: 330
            Interface mgmt
            type: internal
    ovs_version: "2.17.11"
[root@sw3-hq ~]#
```

Задание № 6

Настройте протокол основного дерева

i. Корнем дерева должен выступать SW1-HQ

SW1-HQ:

Настроим Bridge SW1-HQ на участие в дереве 802.1D:

```
ovs-vsctl set bridge SW1-HQ stp_enable=true
```

Задаём приоритет для Bridge SW1-HQ в 16384, т.к. по условиям задания он должен быть корневым:

```
ovs-vsctl set bridge SW1-HQ other_config:stp-priority=16384
```

Проверяем:


```
[root@sw1-hq ~]# ovs-appctl stp/show
---- SW1-HQ ----
Root ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:0a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s
  This bridge is the root

Bridge ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:0a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s

Interface Role State Cost Pri.Nbr
-----
ens20 designated forwarding 4 128.1
ens21 designated forwarding 4 128.2
ens19 designated forwarding 4 128.3

[root@sw1-hq ~]#
```

SW2-HQ:

Настроим Bridge SW2-HQ на участие в дереве 802.1D:

ovs-vsctl set bridge SW2-HQ stp_enable=true

Задаём приоритет для Bridge SW2-HQ в 24576, т.к. по условиям задания SW1-HQ должен быть корневым:

ovs-vsctl set bridge SW2-HQ other_config:stp-priority=24576

Проверяем:

```
[root@sw1-hq ~]# ovs-appctl stp/show
---- SW1-HQ ----
Root ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:0a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s
  This bridge is the root

Bridge ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:0a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s

Interface Role State Cost Pri.Nbr
-----
ens20 designated forwarding 4 128.1
ens21 designated forwarding 4 128.2
ens19 designated forwarding 4 128.3

[root@sw1-hq ~]#
```

```
[root@sw2-hq ~]# ovs-appctl stp/show
---- SW2-HQ ----
Root ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:0a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s
  root-port ens19
  root-path-cost 4

Bridge ID:
  stp-priority 24576
  stp-system-id be:b1:a1:d2:02:42
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s

Interface Role State Cost Pri.Nbr
-----
ens20 designated forwarding 4 128.1
ens22 designated forwarding 4 128.2
ens21 designated forwarding 4 128.3
ens19 root forwarding 4 128.4

[root@sw2-hq ~]#
```

SW3-HQ:

Настроим Bridge SW3-HQ на участие в дереве 802.1D:

```
ovs-vsctl set bridge SW3-HQ stp_enable=true
```

Задаём приоритет для Bridge SW3-HQ в 28672, т.к. по условиям задания SW1-HQ должен быть корневым:

```
ovs-vsctl set bridge SW3-HQ other_config:stp-priority=28672
```

Проверяем:

```
[root@sw1-hq ~]# ovs-appctl stp/show
--- SW1-HQ ---
Root ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:8a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s
  This bridge is the root

Bridge ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:8a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s

Interface Role State Cost Pri.Nbr
-----
ens20 designated forwarding 4 128.1
ens21 designated forwarding 4 128.2
ens19 designated forwarding 4 128.3

[root@sw1-hq ~]#
```

```
[root@sw3-hq ~]# ovs-appctl stp/show
--- SW3-HQ ---
Root ID:
  stp-priority 16384
  stp-system-id fa:fb:8c:8a:95:41
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s
  root-port ens19
  root-path-cost 4

Bridge ID:
  stp-priority 28672
  stp-system-id 3e:91:2c:47:89:47
  stp-hello-time 2s
  stp-max-age 20s
  stp-fwd-delay 15s

Interface Role State Cost Pri.Nbr
-----
ens20 alternate blocking 4 128.1
ens21 designated forwarding 4 128.2
ens19 root forwarding 4 128.3

[root@sw3-hq ~]#
```

Задание № 7

а) На R-HQ настройте протокол динамической конфигурации хостов для клиентов (CLI-HQ)

1. Адрес сети – согласно топологии
 - i. Исключите адрес шлюза по умолчанию из диапазона выдаваемых адресов
2. Адрес шлюза по умолчанию – в соответствии с топологией
 - i. Шлюзом для сети HQ является маршрутизатор R-HQ
3. DNS-суффикс – au.team
4. Настройте клиентов на получение динамических адресов.

R-HQ:

Задаём POOL адресов с именем CLI-HQ, затем задаём диапазон IP-адресов, который будет раздаваться DHCP сервером:

в данном случае раздаваться будет вся клиентская подсеть за исключением IP-адреса маршрутизатора R-HQ

```
r-hq#configure terminal
```

```
r-hq(config)#ip pool CLI-HQ 192.168.11.2-192.168.11.62
```

```
r-hq(config)#
```

Для настройки DHCP-сервера необходимо в режиме конфигурации ввести команду `dhcp-server <NUMBER>`

где NUMBER – номер сервера в системе маршрутизатора:

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

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

```
r-hq(config-dhcp-server)#pool CLI-HQ 1  
r-hq(config-dhcp-server-pool)#  
Задаём основные параметры для раздачи DHCP сервером:  
r-hq(config-dhcp-server-pool)#mask 26  
r-hq(config-dhcp-server-pool)#gateway 192.168.11.1  
r-hq(config-dhcp-server-pool)#dns 77.88.8.8  
r-hq(config-dhcp-server-pool)#domain-name au.team  
r-hq(config-dhcp-server-pool)#exit  
r-hq(config-dhcp-server)#exit  
r-hq(config)#
```

После настройки сервера необходимо указать, на каком интерфейсе маршрутизатор будет принимать пакеты DHCP Discover и отвечать на них предложением с IP-настройками:
в данном случае подинтерфейс с именем vl110 смотрит в сторону клиентской подсети (vlan110)

```
r-hq(config)#interface vl110  
r-hq(config-if)#dhcp-server 1  
r-hq(config-if)#exit  
r-hq(config)#write  
r-hq(config)#
```

```
ip pool CLI 1  
  range 192.168.11.2-192.168.11.62  
?  
dhcp-server 1  
  lease 86400  
  mask 255.255.255.0  
  pool CLI 1  
    gateway 192.168.11.1  
    dns 77.88.8.8  
    domain-name au.team  
    mask 255.255.255.192  
?
```

```

!
interface v1110
 ip mtu 1500
 connect port te1 service-instance te1/v1110
 dhcp-server 1
 ip nat inside
 ip address 192.168.11.1/26
!

```

CLI-HQ:

Настраиваем клиента на получение динамических адресов через Центр управления системой:

```

[user@cli-hq ~]$ ip -c 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 8a:da:e5:8c:e6:cd brd ff:ff:ff:ff:ff:ff
   altname enp0s19
   inet 192.168.11.2/26 brd 192.168.11.63 scope global dynamic noprefixroute ens19
       valid_lft 71853sec preferred_lft 71853sec
   inet6 fe80::88da:e5ff:fe8c:e6cd/64 scope link
       valid_lft forever preferred_lft forever
[user@cli-hq ~]$ ip -c r
default via 192.168.33.1 dev ens19 proto dhcp src 192.168.11.2 metric 100
192.168.11.0/26 dev ens19 proto kernel scope link src 192.168.11.2 metric 100
192.168.33.1 dev ens19 proto dhcp scope link src 192.168.11.2 metric 100
[user@cli-hq ~]$ cat /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
search au.team
nameserver 192.168.33.66
nameserver 192.168.11.66
[user@cli-hq ~]$

```

Задание № 12

с) На всех устройства создайте пользователя sshuser с паролем P@ssw0rd

1. Пользователь sshuser должен иметь возможность запуска утилиты sudo без дополнительной аутентификации.
2. На маршрутизаторах пользователь sshuser должен обладать максимальными привилегиями.

SW1-HQ, SW2-HQ, SW3-HQ, SRV1-HQ:

Для создания пользователя sshuser используем утилиту useradd:

где:

useradd - утилита для создания пользователя;

sshuser - имя пользователя;

-m - если домашнего каталога пользователя не существует, то он будет создан;

-U - создаётся одноимённая группа и пользователь автоматически в неё добавляется;

-s /bin/bash - задаётся командный интерпретатор для пользователя:

useradd sshuser -m -U -s /bin/bash

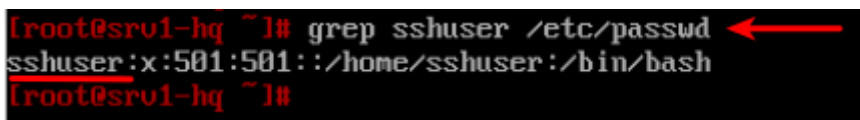
Проверяем созданного пользователя с необходимыми параметрами:

для этого необходимо открыть содержимое файла /etc/passwd с помощью утилиты cat или же текстовым редактором, например: vim;

или же использовать утилиту grep и передать ей в качестве значения имя пользователя:

grep sshuser /etc/passwd

Результат на примере для SRV1-HQ:



```
[root@srv1-hq ~]# grep sshuser /etc/passwd  
sshuser:x:501:501:~/home/sshuser:/bin/bash  
[root@srv1-hq ~]#
```

Аналогично для всех остальных устройств

Для назначения пользователя sshuser пароля P@ssw0rd используем утилиту passwd:

во время запуска - утилита в интерактивном режиме попросит ввести пароль для пользователя и затем подтвердить его:

passwd sshuser

Результат запуска утилиты:

```

[root@srv1-hq ~]# 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: "tree!Morale!Reason".

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@srv1-hq ~]#

```

Аналогично для всех остальных устройств

Реализуем возможность запуска утилиты sudo пользователю sshuser без ввода пароля: Добавляем пользователя sshuser в группу wheel для этого используем утилиту usermod поскольку штатное состояние политики: wheelonly (Означает что пользователь из группы wheel имеет право запускать саму команду sudo, но не означает, что он через sudo может выполнить какую-то команду с правами root)

где:

- usermod - утилита для изменения и работы с параметрами пользователя;
- -aG - параметр чтобы добавить пользователя в дополнительную группу(ы). Использовать только вместе с параметром -G;
- wheel - имя группы;
- sshuser - имя пользователя;
-

usermod -aG wheel sshuser

Добавляем следующую строку в файл в /etc/sudoers чтобы была возможность запуска sudo без дополнительной аутентификации:

echo "sshuser ALL=(ALL:ALL) NOPASSWD: ALL" >> /etc/sudoers

P.S. или же открываем через текстовый редактор, например: vim

Проверяем:

на SRV-HQ1 выполняем вход из-под пользователя sshuser и пытаемся повысить привилегии:

```

Hostname: srv1-hq.au.team
IP: 192.168.11.66
srv1-hq login: sshuser
Password:
Last login:
[sshuser@srv1-hq ~]$ sudo -i
[root@srv1-hq ~]#

```

Аналогично для всех остальных устройств с ОС Альт Линукс

R-HQ:

Создаём пользователя sshuser на маршрутизаторах с паролем P@ssw0rd и с максимальными привилегиями:

максимальным привилегиям в EcoRouter - соответствуем роль admin:

```
r-hq#configure terminal
```

```
r-hq(config)#username sshuser
```

```
r-hq(config-user)#password P@ssw0rd
```

```
r-hq(config-user)#role admin
```

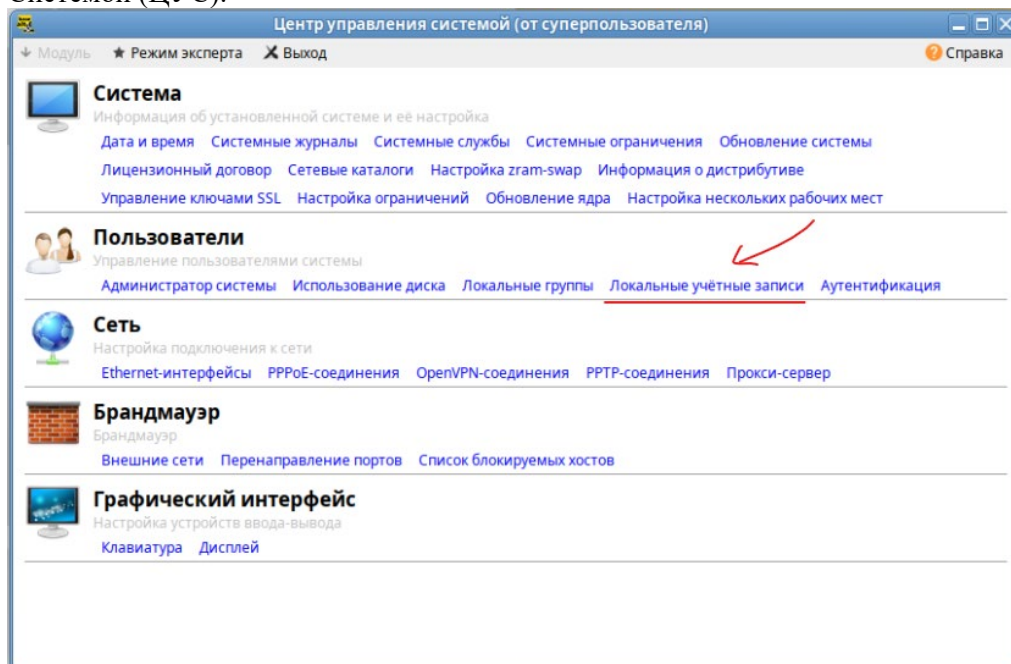
```
r-hq(config-user)#exit
```

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

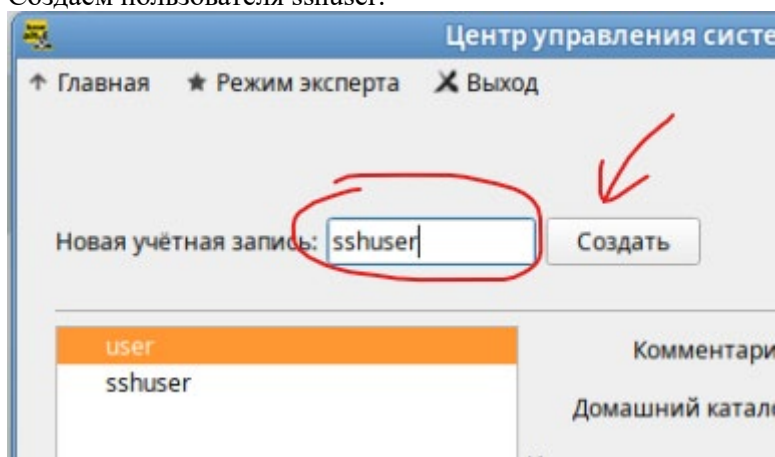
```
r-hq(config)#
```

ADMIN-HQ, CLI-HQ:

Поскольку клиенты имеют графический интерфейс - воспользуемся Центром Управления Системой (ЦУС):



Создаём пользователя sshuser:



Центр управления системой (от суперпользователя)

↑ Главная ★ Режим эксперта ✕ Выход Справка

Новая учётная запись:

Выбрать аватар

user
sshuser

Комментарий:
Домашний каталог: /home/sshuser
Интерпретатор команд: /bin/bash

☒ Входит в группу администраторов

Назначенные системные роли
☐ users
☐ localadmins

Группы, в которые входит пользо
wheel

Пароль: ☐ Создать автоматически
 (введите фразу)
 (повторите фразу)
☐ Автоматический вход в систему

Режим киоска Обычный рабочий стол

user
sshuser

Комментарий:
Домашний каталог: /home/sshuser
Интерпретатор команд: /bin/bash

☒ Входит в группу администраторов

Назначенные системные роли
☐ users
☐ localadmins
☐ powerusers

Группы, в которые входит пользо
sshuser
wheel

Пароль: ☐ Создать автоматически
 (введите фразу)
 (повторите фразу)
☐ Автоматический вход в систему

Режим киоска Обычный рабочий стол

Устанавливаем пакет sudo:

для доступа в сеть Интернет можно временно использовать публичный DNS (echo 'nameserver 77.88.8.8' > /etc/resolv.conf)

apt-get update && apt-get install -y sudo

Добавляем следующую строку в файл в **/etc/sudoers** чтобы была возможность запуска sudo без дополнительной аутентификации:

echo "sshuser ALL=(ALL:ALL) NOPASSWD: ALL" >> /etc/sudoers

ИЛИ

```
[root@cli-dt ~]# vim /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
sshuser ALL=(ALL:ALL) NOPASSWD: ALL
## Uncomment to allow members of group sudo to execute any command
# SUDO_USERS ALL=(ALL:ALL) ALL

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d
~
```

Проверяем:

```
[root@cli-dt ~]# su sshuser
bash: /root/.bashrc: Отказано в доступе
bash-4.4$ sudo -i
[root@cli-dt ~]#
```

Аналогично для всех остальных устройств: CLI-HQ
