1. Настройка базовых параметров

a. Настройка имени устройства и домена `au.team`

Для каждой виртуальной машины необходимо задать имя, соответствующее топологии, и настроить домен `au.team`. Способ настройки зависит от операционной системы.

Для ОС на базе Linux (Альт Сервер, Альт Рабочая станция, Симпли Линукс):

```
# Установка имени хоста (например, для SRV-HQ)
hostnamectl set-hostname SRV-HQ.au.team; exec bash
# Или через файл /etc/hostname (после перезагрузки)
echo "SRV-HQ" > /etc/hostname

# Прописать в /etc/hosts, чтобы FQDN разрешался в локальный адрес
echo "127.0.0.1   localhost SRV-HQ.au.team SRV-HQ" >> /etc/hosts
# Для остальных интерфейсов также можно добавить соответствующие IP, но это не обязательно.
```

Для Ideco NGFW (FW-HQ):

Войти в консоль устройства.
В меню настройки системы изменить имя хоста на `FW-HQ.au-team`.
Домен `au.team` задать в параметрах DNS (если требуется).

Для ViPNet xFirewall (FW-BR):

Войти в CLI или веб-интерфейс.
В настройках устройства указать имя `FW-BR` и домен `au.team`.

Для EcoRouterOS (RTR-BR, RTR-COD):

```
hostname RTR-BR
ip domain-name au.team
write
```

Для всех устройств также рекомендуется проверить, что локальное имя резолвится правильно (например, через `ping $(hostname) или hostname -f`).

b. Создание учётной записи `net_admin` на RTR-BR, RTR-COD и SW-COD

На SW-COD (Альт Сервер 11):

```
# Создание пользователя
useradd -m -G wheel net_admin
# Установка пароля P@ssw0rd (в интерактивном режиме)
passwd net_admin
# Затем ввести пароль дважды

# Настройка sudo без запроса пароля
echo "net_admin ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/net_admin
chmod 440 /etc/sudoers.d/net_admin
```

На RTR-BR и RTR-COD (EcoRouterOS):

В режиме глобальной конфигурации создать пользователя с максимальными привилегиями (уровень 15, аналогично Cisco):

```
configure
username net_admin
password P@ssword
role admin
activate
exit
write

ssh <имя_пользователя>@<IP-адрес>
```

Если требуется, настроить доступ по SSH и локальную аутентификацию.

c. Настройка IP-адресации согласно топологии L3

На EcoRouterOS (RTR-BR, RTR-COD):

```
interface eth0
 ip address 84.212.78.78/27
 exit
interface eth1
 ip address 10.2.0.1/30
 exit
interface loopback0
 ip address 192.168.255.2/32
 exit
ip route 0.0.0.0/0 84.212.78.94   # шлюз по умолчанию
```

Аналогично для RTR-COD.

На Ideco NGFW:
Настройка выполняется через веб-интерфейс (по умолчанию доступен по адресу 192.168.0.1 или через консоль). Необходимо создать VLAN-интерфейсы на физическом порту, подключённом к vSW-HQ, и назначить IP-адреса согласно таблице.

На ViPNet xFirewall:
Через консоль или веб-интерфейс настроить интерфейсы: внешний (eth0) с IP 10.2.0.2/30, внутренние VLAN на другом интерфейсе.

d. Настройка коммутации в соответствии с топологией L2

vSW-HQ и vSW-BR (виртуальные коммутаторы гипервизора):
В настройках гипервизора необходимо для каждой виртуальной машины, подключённой к соответствующему коммутатору, задать VLAN ID (тег) на сетевом интерфейсе:

vSW-HQ:

SRV-HQ → VLAN 10
ADM-HQ → VLAN 20
CLI-HQ → VLAN 30

FW-HQ → интерфейс в режиме **trunk** (пропускать все VLAN). В большинстве гипервизоров для этого указывается `VLAN ID 0` или `4095`, либо не указывается тег, а в гостевой ОС настраиваются VLAN-сабинтерфейсы.

vSW-BR:

SRV-BR → VLAN 10
CLI-BR → VLAN 20
FW-BR → интерфейс в режиме trunk (для VLAN 10 и 20)

Конкретная реализация зависит от гипервизора (VMware, VirtualBox, Proxmox и т.д.). Например, в **Proxmox VE** при создании сетевого устройства ВМ можно указать `tag=10` для access-порта, а для trunk использовать `tag=0` или несколько тегов через запятую.

SW-COD (Open vSwitch на Альт Сервер 11):

Установить Open vSwitch, если не установлен:

```
apt-get install openvswitch
```

Создать мост с именем `sw-cod` (короткое имя устройства):

```
ovs-vsctl add-br sw-cod
```

Добавить все физические интерфейсы, подключённые к другим устройствам, в мост. Предположим, что интерфейсы называются `eth0` (к RTR-COD), `eth1` (к HA1-COD), `eth2` (к HA2-COD), `eth3` (к SRV1-COD), `eth4` (к SRV2-COD), `eth5` (к SRV3-COD). Тогда:

```
ovs-vsctl add-port sw-cod eth0
ovs-vsctl add-port sw-cod eth1
ovs-vsctl add-port sw-cod eth2
ovs-vsctl add-port sw-cod eth3
ovs-vsctl add-port sw-cod eth4
ovs-vsctl add-port sw-cod eth5
```

Поднять интерфейсы (обычно OVS делает это автоматически, но можно проверить):

```
ip link set eth0 up
ip link set eth1 up
...
```

Для управления SW-COD назначить IP-адрес на сам мост (или на отдельный интерфейс управления). Например:

```
ip addr add 172.16.1.4/23 dev sw-cod
ip link set sw-cod up
```

Для постоянной настройки необходимо прописать конфигурацию в файлы сети (например, в `/etc/net/ifaces/sw-cod/options` с типом `ovs`).

Включить OVS в автозагрузку:

```
systemctl enable openvswitch
systemctl start openvswitch
```

е. Обеспечение доступности веб-интерфейсов FW-HQ и FW-BR с ADM-HQ

После настройки IP-адресов и маршрутизации (которая будет настроена позже) веб-интерфейсы должны быть доступны:

FW-HQ — по адресу `10.1.1.33` (интерфейс vlan20, в котором находится ADM-HQ).
FW-BR — по адресу `10.2.1.14` (vlan10) или `10.2.0.2` (внешний), но доступ из сети ADM-HQ потребует наличия маршрута через RTR-BR и далее.

- На данном этапе достаточно убедиться, что на межсетевых экранах включена служба веб-интерфейса на соответствующих интерфейсах. Обычно это включено по умолчанию. Для проверки можно временно добавить статический маршрут на ADM-HQ (если сеть позволяет), но это выходит за рамки базовой настройки.
- **Рекомендация:** после выполнения всех пунктов проверить связность с помощью `ping` с ADM-HQ до адресов FW-HQ и FW-BR (после настройки маршрутизации на промежуточных устройствах).

2. Настройка обмена маршрутной информацией по протоколу BGP

Подготовка: Настройка IS-IS на RTR-BR и RTR-COD

По условию, в качестве IGP для достижимости loopback-адресов используется IS-IS. Это необходимо для работы iBGP, так как соседство устанавливается через loopback.

На RTR-BR (EcoRouterOS):

Войдите в режим конфигурации: `configure terminal`
Включите маршрутизацию IP, если ещё не включена: `ip routing`
Настройте процесс IS-IS с именем `1` и NET-адрес.

System ID формируется из адреса loopback. Адрес `192.168.255.2` преобразуется в `1921.6825.5002`, что соответствует записи из таблицы адресации (для RTR-BR). NET-адрес будет: `49.0001.1921.6825.5002.00` .

```
router isis
 net 49.0001.1921.6825.5002.00
 is-type level-2-only
 exit
```

Включите IS-IS на всех интерфейсах, которые должны участвовать в маршрутизации (в данном случае — интерфейсы, через которые достижим loopback соседа, то есть внешний интерфейс к ISP, так как ISP тоже участвует в IS-IS).

```
interface eth0
 ip router isis 1
 isis circuit-type level-2-only
 exit
```

Интерфейс loopback также должен участвовать в IS-IS для анонса своей сети.

```
interface loopback0
 ip router isis 1
 exit
```

На RTR-COD (EcoRouterOS):

Войдите в режим конфигурации: `configure terminal`
Убедитесь, что маршрутизация IP включена: `ip routing`

Настройте процесс IS-IS с именем `1` и NET-адресом.

Адрес `192.168.255.3` (предположим, он будет назначен на loopback RTR-COD, в таблице адресации он явно не указан, но должен быть по аналогии с RTR-BR) преобразуется в `1921.6825.5003`.

```
router isis 1
 net 49.0001.1921.6825.5003.00
 is-type level-2-only
 exit
```

Включите IS-IS на интерфейсах.

```
interface eth0
 ip router isis 1
 isis circuit-type level-2-only
 exit
interface loopback0
 ip router isis 1
 exit
```

После настройки IS-IS необходимо убедиться, что маршрутизаторы знают о loopback-адресах друг друга и ISP. Для проверки используйте команду `show ip route isis`.

a. & b. Настройка BGP на RTR-BR и RTR-COD

На RTR-BR (EcoRouterOS):

Войдите в режим конфигурации: `configure terminal`
Настройте BGP, используя локальный AS 64499. Router ID должен соответствовать адресу loopback.

```
router bgp 64499
 bgp router-id 192.168.255.2
 no bgp default ipv4-unicast
 exit
```

`no bgp default ipv4-unicast` — отключает автоматическую активацию IPv4 адрес-семейства для соседей, что является хорошей практикой .

Настройте соседа (ISP). Поскольку это iBGP (оба в AS 64499), используется `remote-as 64499`. Для установки соседства через loopback необходимо указать `update-source` и, так как сосед не находится в прямой связности, — `ebgp-multihop` (в данном случае для iBGP это будет просто `multihop`). IP-адрес ISP (loopback 192.168.255.1) берётся из таблицы адресации.

```
router bgp 64499
 neighbor 192.168.255.1 remote-as 64499
 neighbor 192.168.255.1 update-source loopback0
 neighbor 192.168.255.1 multihop
 exit
```

Активируйте соседа в адрес-семействе IPv4.

```
router bgp 64499
 address-family ipv4
 neighbor 192.168.255.1 activate
 exit
```

На RTR-COD (EcoRouterOS):

Настройка выполняется аналогично, с учётом своего Router ID.

Войдите в режим конфигурации: `configure terminal`
Настройте BGP.

```
router bgp 64499
 bgp router-id 192.168.255.3   ! Предполагаемый адрес loopback
 no bgp default ipv4-unicast
 exit
```

Настройте соседа (ISP).

```
router bgp 64499
 neighbor 192.168.255.1 remote-as 64499
 neighbor 192.168.255.1 update-source loopback0
 neighbor 192.168.255.1 multihop
 exit
```

Активируйте соседа.

```
router bgp 64499
 address-family ipv4
 neighbor 192.168.255.1 activate
 exit
```

c. Запрет анонсирования внутренних сетей провайдеру

По условию, анонсировать внутренние сети в провайдера запрещено. Провайдер — это ISP. В данной конфигурации мы сами являемся клиентами ISP и не должны анонсировать ему какие-либо свои сети. Маршрутизаторы RTR-BR и RTR-COD не будут анонсировать ничего, так как мы не настроили ни одной команды `network` и не включили редистрибуцию .

Таким образом, условие выполнено по умолчанию.

d. Получение маршрута по умолчанию от провайдера через BGP

Для выполнения этого пункта настройка должна быть выполнена на стороне **ISP**. В задании сказано, что ISP уже настроен и анонсирует маршрут `0.0.0.0/0` [Предварительные условия]. Однако, чтобы клиенты (RTR-BR, RTR-COD) гарантированно получили этот маршрут, на ISP может быть настроена отправка default-route конкретным соседям с помощью команды `neighbor default-originate`.

Так как у нас нет доступа к ISP, мы должны убедиться, что на RTR-BR и RTR-COD есть принятый маршрут по умолчанию. Для этого на маршрутизаторах нужно выполнить проверку:

```
show ip bgp neighbor 192.168.255.1 advertised-routes
show ip bgp
```

В выводе команды `show ip bgp` должна присутствовать запись для сети `0.0.0.0/0`.

Если по какой-то причине маршрут по умолчанию не приходит, но на ISP включена его генерация (а по условию она включена), проблема может быть в фильтрации. В этом случае нам нужно было бы обратиться к администратору ISP.

Проверка:
После настройки BGP на RTR-BR и RTR-COD, в их таблицах маршрутизации должен появиться маршрут по умолчанию, полученный от ISP. Это можно проверить командой:

```
show ip route
```

В выводе должна быть строка: `S* 0.0.0.0/0 [20/0] via 192.168.255.1` (или похожая, где `[20/0]` указывает на маршрут, полученный по BGP).

Настройка BGP для RTR-BR и RTR-COD завершена. Следующим шагом, вероятно, будет настройка BGP на межсетевых экранах или внутренних маршрутизаторах.

3. Настройка доступа в Интернет

а. Настройка NAT для всех трёх сегментов

По условию, NAT настраивается в сторону ISP. Это означает, что устройство, имеющее выход в интернет (FW-HQ, FW-BR, RTR-COD), должно выполнять трансляцию частных адресов своих локальных сетей в свой публичный IP-адрес (полученный от ISP).

Настройка NAT на FW-HQ (Ideco NGFW):

В Ideco NGFW NAT настраивается правилами SNAT (Source NAT). Для базового доступа в интернет можно использовать автоматический SNAT или создать правило вручную.

Войдите в веб-интерфейс FW-HQ.
Перейдите в раздел **Правила трафика** → **Файрвол** → **SNAT**.
Нажмите **Добавить**.
Настройте правило:

Название: `NAT_to_Internet_HQ`
Протокол: `any`
Адрес источника: `Локальные сети HQ` (можно выбрать созданный ранее объект или указать подсети вручную: `10.1.1.0/27`, `10.1.1.32/28`, `10.1.2.0/24`).
Адрес назначения: `!Внешние сети` (или `any`, но с учётом, что NAT не нужен для трафика в локальные сети). Можно оставить `any`.
Сменить IP-адрес источника на: выбрать внешний интерфейс, который смотрит в сторону ISP (например, `eth0` с IP `63.27.18.18`). Ideco автоматически подставит его адрес.
Нажмите **Добавить** и **Сохранить** .

Либо можно включить режим **"Разрешить интернет всем"** в разделе **Пользователи** → **Авторизация**, что автоматически создаст необходимые правила NAT и файрвола для локальных сетей, но этот способ менее гибкий .

Настройка NAT на FW-BR (ViPNet xFirewall):

Настройка NAT в ViPNet выполняется через политики сети. Необходимо создать правило трансляции адресов для локальных сетей (`10.2.1.0/28`, `10.2.2.0/25`) при выходе через внешний интерфейс (с адресом `10.2.0.2`).

Войдите в веб-интерфейс или CLI FW-BR.
Перейдите в раздел настройки межсетевого экрана (Firewall Policies).

Создайте правило для исходящего трафика из доверенных зон (Trust) в ненадёжную зону (Untrusted — интернет).
В параметрах правила укажите действие **SNAT** (или Masquerade), используя IP-адрес внешнего интерфейса.
Примените политику.

Настройка NAT на RTR-COD (EcoRouterOS):

Для доступа серверов COD в интернет на RTR-COD настраивается NAT. Предположим, что внутренний интерфейс — `eth1` (в сторону SW-COD, подсеть `172.16.0.0/23`), внешний — `eth0` (в сторону ISP, IP `34.95.33.33`).

Войдите в консоль RTR-COD и перейдите в режим конфигурации: `configure terminal`
Создайте список доступа (ACL), определяющий трафик, который нужно транслировать (весь трафик из локальной сети):

```
access-list 100 permit ip 172.16.0.0 0.0.1.255 any
```

Настройте NAT (маскарадинг) на внешнем интерфейсе:

```
interface eth0
 ip nat outside
 exit
interface eth1
 ip nat inside
 exit
```

Включите NAT:

```
ip nat inside source list 100 interface eth0 overload
```

`overload` означает использование PAT (Port Address Translation), когда все внутренние адреса транслируются в один внешний IP.

Проверьте конфигурацию: `show ip nat translations`

b. Обеспечение доступа всех устройств в Интернет

При корректной настройке IP-адресов (шлюзы должны указывать на соответствующий межсетевой экран или маршрутизатор) и правил NAT/NGFW, доступ в интернет будет обеспечен.

Требования для каждого сегмента:

Офис HQ: Устройства SRV-HQ, ADM-HQ, CLI-HQ должны иметь шлюз по умолчанию, указывающий на FW-HQ (соответствующий VLAN-интерфейс: `10.1.1.1`, `10.1.1.33`, `10.1.2.1`). FW-HQ уже настроен на NAT.
Офис BR: SRV-BR и CLI-BR должны иметь шлюз на FW-BR (`10.2.1.14` для SRV-BR, `10.2.2.1` для CLI-BR). FW-BR должен иметь маршрут до интернета через RTR-BR (это будет настроено далее) и выполнять NAT.
Центр обработки данных COD: SRV1-COD, SRV2-COD, SRV3-COD должны иметь шлюз по умолчанию на SW-COD? Нет, они должны иметь шлюз на RTR-COD. Адрес шлюза для них — последний IP в подсети `172.16.0.0/23`, то есть `172.16.1.254`. Именно его мы настроили на интерфейсе RTR-COD `eth1` (172.16.1.254/23). RTR-COD выполняет NAT.

Для проверки достаточно выполнить команду `ping 8.8.8.8` или `ping 1.1.1.1` с любого из этих устройств.

c. Настройка авторизации на межсетевом экране FW-HQ

Для выполнения этого пункта нам необходимо создать трёх пользователей в FW-HQ и настроить для них разные методы авторизации согласно заданию.

Общий шаг: Создание пользователей в Ideco NGFW.

В веб-интерфейсе FW-HQ перейдите в раздел **Пользователи** → **Учетные записи** .
В дереве пользователей выберите группу (например, "Пользователи") и нажмите "Добавить пользователя".
Создайте пять пользователей:

hq.user5: Логин `hq.user5`, пароль `P@ssw0rd`.
hq.user4: Логин `hq.user4`, пароль `P@ssw0rd`.
hq.user3: Логин `hq.user3`, пароль `P@ssw0rd`.

i. Авторизация для SRV-HQ от имени hq.user5

Сервер SRV-HQ имеет статический IP-адрес `10.1.1.10`. Для него подходит метод **IP-авторизации** .

В карточке пользователя **hq.user5** перейдите на вкладку **IP и MAC авторизация** .
Нажмите **Добавить**.
В открывшемся окне заполните поля:

Тип: `IP-адрес`.
IP-адрес: `10.1.1.10`.
Установите флаг **Постоянно авторизован**, чтобы доступ был непрерывным .
Сохраните правило.
Теперь любой трафик, приходящий с IP-адреса `10.1.1.10`, будет автоматически авторизован под пользователем hq.user5 без запроса логина и пароля.

ii. Авторизация для ADM-HQ от имени hq.user4

ADM-HQ имеет статический IP-адрес `10.1.1.46`. Задание требует привязку к IP **и** MAC-адресам. Это метод **IP+MAC авторизации** .

Узнайте MAC-адрес сетевой карты ADM-HQ. Это можно сделать на самой ВМ ADM-HQ командой `ip link show` или `ifconfig`. Предположим, он равен `AA:BB:CC:DD:EE:FF`.
В карточке пользователя **hq.user4** перейдите на вкладку **IP и MAC авторизация**.
Нажмите **Добавить**.
Заполните поля:

Тип: `IP + MAC`.
IP-адрес: `10.1.1.46`.
MAC-адрес: `AA:BB:CC:DD:EE:FF` (введите реальный адрес).
Установите флаг **Постоянно авторизован**.
Сохраните правило.

Важное условие: Для работы авторизации по MAC-адресу устройство пользователя (ADM-HQ) и NGFW (FW-HQ) должны находиться в одном широковещательном домене (L2-сегменте). В нашей

топологии ADM-HQ (VLAN 20) и интерфейс FW-HQ в VLAN 20 (`10.1.1.33`) действительно находятся в одном домене через vSW-HQ, поэтому условие выполняется .

iii. Настройка Ideco Client для CLI-HQ от имени hq.user3

Для клиентского устройства CLI-HQ, которое получает настройки по DHCP, необходимо установить специальное ПО — Ideco Client.

Настройка на FW-HQ:

Убедитесь, что в разделе **Пользователи** → **VPN-подключения** создано правило, разрешающее пользователю `hq.user3` VPN-подключение. Можно создать правило для группы "Пользователи", разрешающее доступ по VPN .
В разделе **Ideco Client** активируйте настройку **Создавать туннель при подключении из локальной сети**. Это необходимо, так как CLI-HQ находится в локальной сети (VLAN 30) .

Настройка на CLI-HQ (Альт Рабочая станция):

Скачайте дистрибутив Ideco Client для Linux. Это можно сделать из личного кабинета пользователя на FW-HQ .
Установите клиент. Для Альт используйте пакет `.rpm` или `.deb` (в зависимости от версии) и установите его через терминал:

```
sudo apt-get install /path/to/ideco-client-package.rpm # или dpkg -i
```

Запустите Ideco Client. При первом запуске потребуется создать профиль подключения.

В настройках профиля укажите:

Адрес сервера (Хост): IP-адрес интерфейса FW-HQ в VLAN 30 (`10.1.2.1`) или его доменное имя (если настроено).
Логин: `hq.user3`
Пароль: `P@ssw0rd`

Сохраните профиль и установите его для автоподключения .

После подключения клиент создаст защищённый туннель (WireGuard или TLS) до FW-HQ, и устройство будет авторизовано в сети под учётной записью hq.user3, имея доступ в интернет согласно правилам файрвола.

Важные замечания

Проверка: После выполнения всех пунктов рекомендуется проверить доступность интернета с каждого типа устройств и убедиться в разделе **Мониторинг** → **Авторизованные пользователи** на FW-HQ, что все три пользователя авторизованы правильно (hq.user5 и hq.user4 будут отображаться как авторизованные по IP/MAC, а hq.user3 — через VPN-сессию) .
Маршрутизация: Для того чтобы NAT работал на RTR-COD и FW-BR, они должны иметь правильные маршруты обратно к своим локальным сетям (что уже настроено) и, что важно, обратный трафик от ISP должен попадать именно на них. Это обеспечивается настройками BGP, где ISP анонсирует маршрут по умолчанию, но не принимает наши внутренние сети. Таким образом, публичные IP-адреса (63.27.18.18, 84.212.78.78, 34.95.33.33) должны быть известны в интернете (предоставлены ISP), и ответный трафик будет приходить именно на эти адреса.

4. Настройка контроллера домена

а. Развертывание контроллера домена FreeIPA на SRV-HQ

Настройка на SRV-HQ (Альт Сервер 11):

Подготовка системы:
Установите статический IP-адрес `10.1.1.10/27` с шлюзом `10.1.1.1`, как указано в таблице адресации. Убедитесь, что сеть работает.
Настройка имени хоста и DNS:

```
# Установка полного доменного имени
hostnamectl set-hostname srv-hq.au.team

# Проверка /etc/hosts (должен содержать только запись для самого себя и localhost)
echo "10.1.1.10   srv-hq.au.team srv-hq" >> /etc/hosts
```

Установка пакетов FreeIPA с интегрированным DNS:

```
apt-get update
apt-get install freeipa-server freeipa-server-dns
```

Запуск установки FreeIPA:

Выполните команду инициализации. Установка будет интерактивной.

```
forcat
```

Ответьте на вопросы установщика :

Do you want to configure integrated DNS (BIND)? → `Yes` (по умолчанию)
Server host name: → `srv-hq.au.team` (должно быть определено автоматически)
Please confirm the domain name: → `au.team`
Please provide a realm name: → `AU.TEAM` (обычно автоматически)
Directory Manager password: → `P@ssw0rd` (пароль для LDAP)
IPA admin password: → `P@ssw0rd` (пароль для администратора FreeIPA `admin`)
Do you want to configure DNS forwarders? → `Yes` (укажите внешний DNS, например `8.8.8.8` или `77.88.8.8`, чтобы клиенты могли резолвить внешние имена).
Do you want to search for missing reverse zones? → `Yes`
Please specify the reverse zone name: Для сети `10.1.1.0/27` это будет `1.1.10.in-addr.arpa`. Укажите их при запросе.
Подтвердите продолжение установки.

Установка займёт несколько минут. После успешного завершения перезагрузите сервер:

```
reboot
```

Проверка:

После перезагрузки войдите и проверьте статус:

```
ipactl status
kinit admin
ipa user-find
```

b. Создание групп и пользователей

На **SRV-HQ** (или любом доменном компьютере с правами администратора) выполните следующие команды :

Аутентификация администратором домена:

```
kinit admin
```

Создание групп `hq`, `br`, `cod`:

```
ipa group-add hq --desc="Head Office Users"
ipa group-add br --desc="Branch Office Users"
ipa group-add cod --desc="Data Center Users"
```

Создание пользователей и добавление их в группы:

Воспользуйтесь небольшим скриптом или выполните команды вручную. Пароль `P@ssw0rd` будет запрошен для каждого пользователя интерактивно.

```
# Для HQ (hq.user1 - hq.user5)
for i in {1..5}; do
    ipa user-add hq.user$i --first=HQ --last=User$i --password
    ipa group-add-member hq --users=hq.user$i
done

# Для BR (br.user1 - br.user5)
for i in {1..5}; do
    ipa user-add br.user$i --first=BR --last=User$i --password
    ipa group-add-member br --users=br.user$i
done

# Для COD (cod.user1 - cod.user5)
for i in {1..5}; do
    ipa user-add cod.user$i --first=COD --last=User$i --password
    ipa group-add-member cod --users=cod.user$i
done
```

Проверка:

```
ipa group-show hq
ipa user-find hq.user1
```

c. Ввод клиентов в домен `au.team`

i. Настройка DNS на клиентах

Прежде чем вводить компьютер в домен, необходимо настроить его на использование SRV-HQ в качестве DNS-сервера .

На ADM-HQ, CLI-HQ (Альт Рабочая станция) и CLI-BR (Альт Рабочая станция):

Если используется NetworkManager:

```
# Отключить автоматическое получение DNS
sudo nmcli con mod "Проводное соединение 1" ipv4.ignore-auto-dns yes
# Установить DNS-сервер (SRV-HQ) и резервный (например, 77.88.8.8)
```

```
echo "nameserver 10.1.1.10" > /etc/resolv.conf
# Перезапустить сеть
systemctl restart network
```

На FW-HQ (Ideco NGFW):

В веб-интерфейсе перейдите **Сервисы** → **DNS** → **Внешние DNS-серверы** и добавьте IP-адрес SRV-HQ (`10.1.1.10`) в качестве первого DNS-сервера .

ii. Установка клиентского ПО и ввод в домен

На ADM-HQ, CLI-HQ, CLI-BR (Альт Рабочая станция):

Установите клиент FreeIPA :

```
sudo apt-get update
sudo apt-get install freeipa-client
```

Настройте имя хоста (полное доменное имя):

```
# Для ADM-HQ
sudo hostnamectl set-hostname adm-hq.au.team; exec bash
# Для CLI-HQ
sudo hostnamectl set-hostname cli-hq.au.team; exec bash
# Для CLI-BR
sudo hostnamectl set-hostname cli-br.au.team; exec bash
```

Убедитесь, что в `/etc/hosts` **нет** строки, связывающей имя компьютера с `127.0.1.1`. Допустима только строка с `127.0.0.1 localhost localhost.localdomain` .

Выполните ввод в домен:

```
sudo ipa-client-install --domain=au.team --server=srv-hq.au.team --enable-dns-updates
```

Программа задаст несколько вопросов. На запрос о продолжении ответьте `yes`.
На запрос обновления `/etc/resolv.conf` ответьте `yes`.
На запрос о настройке SSSD (если появится) также ответьте `yes`.
После завершения будет предложено выполнить `kinit`. Пока можно отказаться (Ctrl+C).

Проверка:

```
# Получить билет Kerberos для проверки
kinit hq.user4  # или любого созданного пользователя
klist
```

iii. Интеграция FW-HQ с доменом FreeIPA

Настройка на **SRV-HQ**:

Создайте роли в FreeIPA, необходимые для Ideco NGFW :

```
kinit admin
ipa role-add "CIFS servers" --desc="Role for CIFS server"
ipa role-add "Organization units" --desc="Role for Organization units"
```

В веб-интерфейсе FreeIPA (`https://srv-hq.au.team`) войдите как `admin`.
Перейдите в раздел **Управление доступом на основе ролей**.

Добавьте в роли **CIFS servers** и **Organization units** всех пользователей, группы и узлы (или необходимые, но для простоты можно добавить всё) .

Настройка на **FW-HQ** (Ideco NGFW):

Убедитесь, что DNS настроен на SRV-HQ (шаг i).
Перейдите в раздел **Пользователи → Внешние каталоги → FreeIPA**.
Нажмите **Добавить** и заполните поля :

Домен: `au.team`
IP-адрес DNS-сервера: `10.1.1.10`
Название сервера: `fw-hq` (можно оставить сгенерированное)
Логин администратора: `admin`
Пароль администратора: `P@ssw0rd`
Сохраните настройки. FW-HQ будет введён в домен.

iv. Импорт пользователей из домена на FW-HQ

На **FW-HQ** перейдите в **Пользователи → Учетные записи**.
Создайте родительскую группу **FreeIPA-Users**:

Нажмите **Добавить группу**.
Имя группы: `FreeIPA-Users`
Тип: `Локальная группа`
Родитель: `Корневая группа` (оставьте как есть)
Сохраните.
Внутри группы `FreeIPA-Users` создайте подгруппы **hq**, **br** и **cod** (аналогично, локальные группы).
Перейдите на вкладку **FreeIPA**. Выберите домен `au.team`.
Для каждой из созданных групп (`hq`, `br`, `cod`) нажмите **Присоединить к домену** и выберите соответствующую группу из FreeIPA для импорта пользователей . Например, для локальной группы `hq` выберите для импорта доменную группу `hq`.
Пользователи будут импортированы. Синхронизация происходит автоматически каждые 15 минут, либо можно запустить её вручную.

d. Настройка DNS-записей через Terraform

Этот этап выполняется на **ADM-HQ** после того, как он введён в домен.

i. Установка Terraform 1.14.5 на ADM-HQ

Загрузите необходимую версию с официального сайта или используйте менеджер пакетов, если она доступна. Для Альт Linux проще скачать бинарный файл.

```
apt-get install unzip
cd /tmp
wget https://hashicorp-releases.yandexcloud.net/terraform/1.14.5/terraform_1.14.5_linux_amd64.zip
unzip terraform_1.14.5_linux_amd64.zip
mv terraform /usr/local/bin/
terraform --version  # Должна отобразиться версия 1.14.5
```

ii. Подготовка структуры директорий и файлов

Создайте необходимые директории и файлы от имени пользователя `user` .

```
mkdir -p /home/user/terraform
```

```
cd /home/user/terraform
```

iii. Создание файла провайдера `provider.tf`

Для аутентификации в FreeIPA потребуется получить билет Kerberos.

```
# provider.tf
terraform {
  required_version = ">= 1.14.5"
  required_providers {
    freeipa = {
      source = "camptocamp/freeipa"
      version = "1.0.0"
    }
  }
}

provider "freeipa" {
  # Адрес сервера FreeIPA
  host     = "srv-hq.au.team"
  # Использовать аутентификацию Kerberos (билет должен быть получен заранее)
  insecure = true  # Отключите в production, здесь для простоты (или настройте CA)
  # Вместо kerberos можно использовать имя пользователя и пароль:
  # username = "admin"
  # password = var.ipa_admin_password
}
```

iv. Создание файла переменных `variables.tf`

```
# variables.tf
variable "ipa_admin_password" {
  description = "FreeIPA admin password"
  type        = string
  sensitive   = true
  default     = "P@ssw0rd"  # Не рекомендуется хранить в открытом виде, но для задания допустимо
}
```

v. Создание основного файла конфигурации `main.tf`

Этот файл будет содержать ресурсы DNS-записей для всех устройств, кроме CLI-HQ и CLI-BR (они получают IP по DHCP и будут обрабатываться динамически через SSSD или DHCP-сервер) .

```
# main.tf
# Получение билета Kerberos перед запуском terraform apply:
# kinit admin

# Прямые A-записи
resource "freeipa_dns_record" "srv_hq_a" {
  zone    = "au.team"
  name    = "srv-hq"
  records = ["10.1.1.10"]
}
```

```
resource "freeipa_dns_record" "adm_hq_a" {
  zone    = "au.team"
  name    = "adm-hq"
  records = ["10.1.1.46"]
}

resource "freeipa_dns_record" "fw_hq_a" {
  zone    = "au.team"
  name    = "fw-hq"
  records = ["10.1.1.33"]  # IP в vlan20 для управления
}

resource "freeipa_dns_record" "rtr_br_a" {
  zone    = "au.team"
  name    = "rtr-br"
  records = ["10.2.0.1", "84.212.78.78"]  # Можно добавить несколько IP, но для A-записи обычно оди
н. Лучше создать отдельные записи для разных интерфейсов.
}

# Для RTR-BR лучше создать отдельные записи для разных интерфейсов:
resource "freeipa_dns_record" "rtr_br_wan_a" {
  zone    = "au.team"
  name    = "rtr-br-wan"
  records = ["84.212.78.78"]
}

resource "freeipa_dns_record" "rtr_br_lan_a" {
  zone    = "au.team"
  name    = "rtr-br-lan"
  records = ["10.2.0.1"]
}

# Аналогично для других устройств: fw-br, srv-br, rtr-cod, sw-cod, ha1-cod, ha2-cod, srv1-cod, srv2
-cod, srv3-cod

resource "freeipa_dns_record" "fw_br_a" {
  zone    = "au.team"
  name    = "fw-br"
  records = ["10.2.0.2", "10.2.1.14"]
}

resource "freeipa_dns_record" "srv_br_a" {
  zone    = "au.team"
  name    = "srv-br"
  records = ["10.2.1.10"]
}

resource "freeipa_dns_record" "rtr_cod_a" {
  zone    = "au.team"
  name    = "rtr-cod"
```

```
  records = ["34.95.33.33", "172.16.1.254"]
}

resource "freeipa_dns_record" "sw_cod_a" {
  zone    = "au.team"
  name    = "sw-cod"
  records = ["172.16.1.4"]
}

resource "freeipa_dns_record" "ha1_cod_a" {
  zone    = "au.team"
  name    = "ha1-cod"
  records = ["172.16.0.1"]
}

resource "freeipa_dns_record" "ha2_cod_a" {
  zone    = "au.team"
  name    = "ha2-cod"
  records = ["172.16.0.2"]
}

resource "freeipa_dns_record" "srv1_cod_a" {
  zone    = "au.team"
  name    = "srv1-cod"
  records = ["172.16.1.1"]
}

resource "freeipa_dns_record" "srv2_cod_a" {
  zone    = "au.team"
  name    = "srv2-cod"
  records = ["172.16.1.2"]
}

resource "freeipa_dns_record" "srv3_cod_a" {
  zone    = "au.team"
  name    = "srv3-cod"
  records = ["172.16.1.3"]
}

# PTR-записи (обратные зоны)
# Для сети 10.1.1.0/27 (зона 1.1.10.in-addr.arpa)
resource "freeipa_dns_record" "srv_hq_ptr" {
  zone    = "1.1.10.in-addr.arpa"
  name    = "10"  # последний октет адреса 10.1.1.10
  records = ["srv-hq.au.team."]  # Обратите внимание на точку в конце
}

resource "freeipa_dns_record" "adm_hq_ptr" {
  zone    = "1.1.10.in-addr.arpa"
  name    = "46"
```

```
  records = ["adm-hq.au.team."]
}

# Для сети 10.2.1.0/28 (зона 1.2.10.in-addr.arpa)
resource "freeipa_dns_record" "srv_br_ptr" {
  zone    = "1.2.10.in-addr.arpa"
  name    = "10"
  records = ["srv-br.au.team."]
}

# Продолжить для всех статических устройств...
```

vi. Применение конфигурации

Получите билет Kerberos от имени администратора домена:

```
kinit admin
```

Инициализируйте Terraform (скачает провайдер):

```
terraform init
```

Проверьте план (посмотрите, что будет создано):

```
terraform plan
```

Примените конфигурацию:

```
terraform apply -auto-approve
```

После успешного выполнения все статические записи A и PTR будут созданы в DNS FreeIPA.

Примечание: Для устройств CLI-HQ и CLI-BR записи создавать через Terraform не требуется. Они будут регистрироваться в DNS автоматически службой SSSD при получении IP-адреса по DHCP, если при вводе в домен был использован ключ `--enable-dns-updates` .

Итоговые FQDN устройств:

SRV-HQ: `srv-hq.au.team` (10.1.1.10)
ADM-HQ: `adm-hq.au.team` (10.1.1.46)
CLI-HQ: `cli-hq.au.team` (DHCP)
FW-HQ: `fw-hq.au.team` (10.1.1.33)
SRV-BR: `srv-br.au.team` (10.2.1.10)
CLI-BR: `cli-br.au.team` (DHCP)
FW-BR: `fw-br.au.team` (10.2.1.14, 10.2.0.2)
RTR-BR: `rtr-br.au.team` (84.212.78.78, 10.2.0.1)
RTR-COD: `rtr-cod.au.team` (34.95.33.33, 172.16.1.254)
SW-COD: `sw-cod.au.team` (172.16.1.4)
HA1-COD: `ha1-cod.au.team` (172.16.0.1)
HA2-COD: `ha2-cod.au.team` (172.16.0.2)
SRV1-COD: `srv1-cod.au.team` (172.16.1.1)

SRV2-COD: `srv2-cod.au.team` (172.16.1.2)

SRV3-COD: `srv3-cod.au.team` (172.16.1.3)

5. Настройка протокола динамической конфигурации хостов

а. Настройка DHCP-сервера Kea на SRV-HQ для CLI-HQ

На **SRV-HQ (Альт Сервер 11)** необходимо установить и настроить Kea для раздачи параметров клиентам из подсети CLI-HQ (`10.1.2.0/24`).

Расчет параметров для CLI-HQ:

Подсеть: `10.1.2.0/24`
Диапазон адресов: с 128 по 254 → `10.1.2.128 - 10.1.2.254`
Шлюз по умолчанию: `10.1.2.1` (FW-HQ)
DNS-сервер: `10.1.1.10` (SRV-HQ)
DNS-суффикс: `au.team`

Пошаговая настройка:

Установка Kea:

```
sudo apt-get update
sudo apt-get install kea
```

Будет установлен пакет `kea`, включающий `kea-dhcp4` .

Настройка конфигурационного файла:

Отредактируйте файл `/etc/kea/kea-dhcp4.conf` :

```
sudo nano /etc/kea/kea-dhcp4.conf
```

Создание конфигурации:

Замените содержимое файла следующей конфигурацией (или отредактируйте существующую):

```
{
    "Dhcp4": {
        "interfaces-config": {
            "interfaces": [ "eth0/10.1.1.10" ]
        },
        "lease-database": {
            "type": "memfile",
            "persist": true,
            "name": "/var/lib/kea/kea-leases4.csv"
        },
        "expired-leases-processing": {
            "reclaim-timer-wait-time": 10,
            "flush-reclaimed-timer-wait-time": 25,
            "hold-reclaimed-time": 3600,
            "max-reclaim-leases": 100,
            "max-reclaim-time": 250,
            "unwarned-reclaim-cycles": 5
```

```
        },
        "valid-lifetime": 3600,
        "renew-timer": 1800,
        "rebind-timer": 2700,
        "option-data": [
            {
                "name": "domain-name-servers",
                "data": "10.1.1.10"
            },
            {
                "name": "domain-name",
                "data": "au.team"
            }
        ],
        "subnet4": [
            {
                "id": 1,
                "subnet": "10.1.2.0/24",
                "pools": [
                    {
                        "pool": "10.1.2.128 - 10.1.2.254"
                    }
                ],
                "option-data": [
                    {
                        "name": "routers",
                        "data": "10.1.2.1"
                    }
                ]
            }
        ],
        "loggers": [
            {
                "name": "kea-dhcp4",
                "output_options": [
                    {
                        "output": "/var/log/kea/kea-dhcp4.log"
                    }
                ],
                "severity": "INFO"
            }
        ]
    }
}
```

Пояснения:

`interfaces`: Указываем интерфейс `eth0` с IP `10.1.1.10`, на котором сервер будет слушать запросы . Если запросы приходят через DHCP-Relay (как в нашем случае), важно указать конкретный IP, чтобы сервер знал, от какого интерфейса ожидать ретранслированные пакеты.
`lease-database`: Используем memfile для хранения информации о выданных арендах .

`valid-lifetime`, `renew-timer`, `rebind-timer`: Время аренды (1 час) и таймеры обновления .
`option-data`: Глобальные опции: DNS-сервер (`10.1.1.10`) и домен (`au.team`).
`subnet4`: Описание подсети `10.1.2.0/24` с пулом адресов `10.1.2.128 - 10.1.2.254` и шлюзом (`routers`) `10.1.2.1` .

Проверка конфигурации и запуск сервиса:

```
# Проверка синтаксиса конфигурации
kea-dhcp4 -t /etc/kea/kea-dhcp4.conf

# Если ошибок нет, включаем и запускаем сервис
systemctl enable --now kea-dhcp4

# Проверка статуса
systemctl status kea-dhcp4
```

b. Настройка DHCP-Relay на FW-HQ для CLI-HQ

Поскольку DHCP-сервер для CLI-HQ (SRV-HQ) находится в другой подсети (`10.1.1.0/27`), а клиенты CLI-HQ в `10.1.2.0/24`, на **FW-HQ (Ideco NGFW)** необходимо настроить ретрансляцию DHCP-запросов .

Войдите в веб-интерфейс FW-HQ.
Перейдите в раздел **Сервисы → DHCP-сервер**.
Для интерфейса, который смотрит в сторону CLI-HQ (это должен быть интерфейс в **VLAN 30**, например `vlan30` с IP `10.1.2.1`), переключите режим работы в **"Ретранслятор DHCP (relay)"** .

Если интерфейс уже используется как DHCP-сервер, его нужно отключить.
В поле **"Внешние серверы"** укажите IP-адрес вашего DHCP-сервера — `10.1.1.10` (SRV-HQ) .
Сохраните настройки.

Альтернативный способ (через настройки интерфейса):

Перейдите в **Сервисы → Сетевые интерфейсы**.
Выберите интерфейс `vlan30` (или `eth0.30`).
В разделе **Дополнительные настройки → Параметры DHCP** установите флаг **"Ретранслятор DHCP"** и укажите IP-адрес сервера `10.1.1.10`.

Теперь все broadcast-запросы от клиентов в VLAN 30 будут перехватываться FW-HQ и перенаправляться unicast на SRV-HQ .

c. Настройка DHCP-сервера на FW-BR для CLI-BR

На **FW-BR (ViPNet xFirewall)** необходимо настроить встроенный DHCP-сервер для раздачи параметров клиентам из подсети CLI-BR (`10.2.2.0/25`).

Расчет параметров для CLI-BR:

Подсеть: `10.2.2.0/25`
Маска: `255.255.255.128`
Адрес шлюза (FW-BR): `10.2.2.1`
Все возможные адреса узлов: от `10.2.2.1` до `10.2.2.126`. Исключая сам шлюз (`10.2.2.1`), получаем диапазон `10.2.2.2 - 10.2.2.126`.
DNS-сервер: `10.1.1.10` (SRV-HQ)

DNS-суффикс: `au.team`

Пошаговая настройка (общий принцип для ViPNet xFirewall):

Войдите в веб-интерфейс FW-BR.
Перейдите в раздел управления **DHCP-сервером**. Обычно он находится в меню **Сервисы** или **Настройки сети**.
Включите DHCP-сервер.
Создайте новую область (scope) или подсеть для интерфейса, который смотрит в сторону CLI-BR. Это должен быть интерфейс в **VLAN 20** (например, `vlan20` с IP `10.2.2.1`).
Заполните параметры в соответствии с расчетами:

Сеть/маска: `10.2.2.0/255.255.255.128` (или `/25`)
Диапазон адресов: `10.2.2.2` – `10.2.2.126`
Шлюз по умолчанию: `10.2.2.1`
DNS-серверы: `10.1.1.10`
Доменное имя: `au.team`
Настройте **время аренды** (по умолчанию обычно 1 день, можно оставить или уменьшить для тестов).
Сохраните и примените конфигурацию.
Обязательно отключите (если включен) DHCP-Relay на этом интерфейсе, так как теперь здесь работает полноценный сервер .

d. Проверка работоспособности

На CLI-HQ:

Переключите сетевой интерфейс на получение адреса по DHCP.
Выполните команды:

```
sudo dhclient -v  # или перезапустите сеть
ip addr show
ip route show
cat /etc/resolv.conf
```

Убедитесь, что интерфейс получил IP-адрес из диапазона `10.1.2.128/24`, шлюз установлен на `10.1.2.1`, а DNS-сервер — `10.1.1.10` с суффиксом `au.team`.

На CLI-BR:

Аналогично переключите интерфейс на DHCP.
Проверьте, что получен IP из диапазона `10.2.2.2-10.2.2.126`, шлюз `10.2.2.1`, DNS `10.1.1.10`.

Проверка доступности:

С обоих клиентов выполните ping до шлюза (`10.1.2.1` и `10.2.2.1` соответственно) и до DNS-сервера (`10.1.1.10`).
Выполните ping до внешнего ресурса по IP (например, `1.1.1.1`) для проверки полной связности и работы NAT.

6. Настройка туннелей GRE

Для связи между офисами и центром обработки данных необходимо настроить три GRE-туннеля. GRE (Generic Routing Encapsulation) позволяет создать прямое соединение поверх существующей IP-сети и будет использоваться для последующей организации маршрутизации между площадками .

Планирование адресации туннелей

На основе топологии L3 и задания определим IP-адреса для каждого туннеля:

Туннель	Устройство A	IP туннеля A	Устройство B	IP туннеля B	Сеть
tunnel.1	FW-HQ	10.0.1.1/30	RTR-BR	10.0.1.2/30	10.0.1.0/30
tunnel.2	FW-HQ	10.0.2.1/30	RTR-COD	10.0.2.2/30	10.0.2.0/30
tunnel.3	RTR-BR	10.0.3.1/30	RTR-COD	10.0.3.2/30	10.0.3.0/30

Точки туннелирования (исходные IP-адреса):

FW-HQ (внешний интерфейс в сторону ISP): `63.27.18.18`
RTR-BR (внешний интерфейс в сторону ISP): `84.212.78.78`
RTR-COD (внешний интерфейс в сторону ISP): `34.95.33.33`

a. Настройка GRE-туннеля между FW-HQ и RTR-BR (tunnel.1)

Настройка на RTR-BR (EcoRouterOS)

Войдите в режим конфигурации:

```
configure terminal
```

Создайте туннельный интерфейс tunnel.1:

```
interface tunnel.1
```

Назначьте IP-адрес из сети 10.0.1.0/30:

```
ip address 10.0.1.2/30
```

Установите значение MTU (с учётом накладных расходов GRE) :

```
ip mtu 1476
```

Задайте параметры туннеля:

```
ip tunnel source 84.212.78.78
ip tunnel destination 63.27.18.18
ip tunnel mode gre
```

Включите keepalive для отслеживания состояния туннеля (опционально, но рекомендуется) :

```
ip tunnel keepalive 10 3
```

Выйдите и сохраните конфигурацию:

```
exit
write memory
```

Настройка на FW-HQ (Ideco NGFW)

В Ideco NGFW настройка GRE-туннелей выполняется через веб-интерфейс (функция доступна начиная с версии 17.0) .

Войдите в веб-интерфейс FW-HQ.
Перейдите в раздел **Сервисы → Туннели → GRE-туннели**.
Нажмите **Добавить** и заполните параметры:

Имя туннеля: `tunnel.1` (или `Tunnel to RTR-BR`)
Локальный IP-адрес туннеля: `10.0.1.1/30`
Удаленный IP-адрес туннеля: `10.0.1.2/30`
Источник (Source): внешний интерфейс с IP `63.27.18.18` (выберите из списка)
Назначение (Destination): `84.212.78.78` (IP-адрес RTR-BR)
MTU: `1476` (рекомендуемое значение)
Сохраните конфигурацию.
Проверьте статус туннеля — он должен отображаться как **"Поднят" (UP)**.

b. Настройка GRE-туннеля между FW-HQ и RTR-COD (tunnel.2)

Настройка на RTR-COD (EcoRouterOS)

Войдите в режим конфигурации:

```
configure terminal
```

Создайте туннельный интерфейс tunnel.2:

```
interface tunnel.2
```

Назначьте IP-адрес из сети 10.0.2.0/30:

```
ip address 10.0.2.2/30
```

Установите MTU:

```
ip mtu 1476
```

Задайте параметры туннеля:

```
ip tunnel source 34.95.33.33
ip tunnel destination 63.27.18.18
ip tunnel mode gre
```

Включите keepalive:

```
ip tunnel keepalive 10 3
```

Выйдите и сохраните:

```
exit
write memory
```

Настройка на FW-HQ (Ideco NGFW)

В разделе **Сервисы → Туннели → GRE-туннели** нажмите **Добавить**.
Заполните параметры:

Имя туннеля: `tunnel.2` (или `Tunnel to RTR-COD`)
Локальный IP-адрес туннеля: `10.0.2.1/30`
Удаленный IP-адрес туннеля: `10.0.2.2/30`

Источник: внешний интерфейс `63.27.18.18`
Назначение: `34.95.33.33`
MTU: `1476`
Сохраните и проверьте статус.

c. Настройка GRE-туннеля между RTR-BR и RTR-COD (tunnel.3)

Настройка на RTR-BR (EcoRouterOS)

Войдите в режим конфигурации:

```
configure terminal
```

Создайте туннельный интерфейс tunnel.3:

```
interface tunnel.3
```

Назначьте IP-адрес из сети 10.0.3.0/30:

```
ip address 10.0.3.1/30
```

Установите MTU:

```
ip mtu 1476
```

Задайте параметры туннеля:

```
ip tunnel source 84.212.78.78
ip tunnel destination 34.95.33.33
ip tunnel mode gre
```

Включите keepalive:

```
ip tunnel keepalive 10 3
```

Выйдите и сохраните:

```
exit
write memory
```

Настройка на RTR-COD (EcoRouterOS)

Войдите в режим конфигурации:

```
configure terminal
```

Создайте туннельный интерфейс tunnel.3:

```
interface tunnel.3
```

Назначьте IP-адрес из сети 10.0.3.0/30:

```
ip address 10.0.3.2/30
```

Установите MTU:

```
ip mtu 1476
```

Задайте параметры туннеля (обратите внимание — source и destination меняются местами):

```
ip tunnel source 34.95.33.33
ip tunnel destination 84.212.78.78
ip tunnel mode gre
```

Включите keepalive:

```
ip tunnel keepalive 10 3
```

Выйдите и сохраните:

```
exit
write memory
```

d. Проверка работоспособности туннелей

После настройки всех трёх туннелей необходимо проверить их состояние и связность.

На RTR-BR (EcoRouterOS):

```
show interface tunnel.1
show interface tunnel.3
ping 10.0.1.1      # проверка связи с FW-HQ через tunnel.1
ping 10.0.3.2      # проверка связи с RTR-COD через tunnel.3
```

На RTR-COD (EcoRouterOS):

```
show interface tunnel.2
show interface tunnel.3
ping 10.0.2.1      # проверка связи с FW-HQ через tunnel.2
ping 10.0.3.1      # проверка связи с RTR-BR через tunnel.3
```

На FW-HQ (Ideco NGFW):

В веб-интерфейсе проверьте статус туннелей в разделе **Сервисы → Туннели**.
Для проверки связности можно использовать встроенную утилиту ping (в разделе **Мониторинг → Диагностика**) или через консоль:

```
ping 10.0.1.2     # до RTR-BR
ping 10.0.2.2     # до RTR-COD
```

Ожидаемый результат: все туннели должны быть в состоянии UP, а команды ping должны проходить успешно.

Важные замечания

MTU: При использовании GRE рекомендуется уменьшить MTU на туннельных интерфейсах, чтобы избежать фрагментации пакетов. Значение 1476 байт учитывает 24 байта служебных заголовков (20 байт IP + 4 байта GRE) .

Keepalive: Механизм keepalive на EcoRouterOS позволяет автоматически отслеживать состояние удалённой стороны. Если удалённая сторона перестанет отвечать, туннель будет отключён, а при восстановлении связи — поднят автоматически .

Маршрутизация через туннели: На данном этапе настроены только сами туннели. Для передачи трафика между площадками потребуется настроить статические маршруты или динамическую маршрутизацию (например, OSPF или BGP) через созданные туннельные интерфейсы. Это будет следующим шагом задания.

Имена интерфейсов: В задании требуется использовать имена `tunnel.1`, `tunnel.2`, `tunnel.3`. На EcoRouterOS это допустимый формат . На Ideco NGFW имя может задаваться произвольно, но для единообразия рекомендуется использовать указанные значения.

7. Настройка маршрутизации

а. Настройка OSPF между BR и COD

Для обеспечения динамической маршрутизации между офисом BR и центром обработки данных COD настроим протокол OSPF на маршрутизаторах RTR-BR, RTR-COD и межсетевом экране FW-BR.

Планирование OSPF

Устройство	Router ID	Участвующие интерфейсы	Пассивные интерфейсы
RTR-BR	192.168.255.2	tunnel.3 (10.0.3.1/30), внешний интерфейс к ISP? (Нет, только туннель для OSPF)	loopback.0, интерфейс к FW-BR (10.2.0.1/30) будет изучаться через OSPF, но сам интерфейс должен быть активным
RTR-COD	192.168.255.3	tunnel.3 (10.0.3.2/30)	loopback.0, интерфейс к SW-COD (172.16.1.254/23)
FW-BR	10.2.0.2	vlan10 (10.2.1.14/28), vlan20 (10.2.2.1/25), интерфейс к RTR-BR (10.2.0.2/30)	нет (все интерфейсы должны участвовать)

Настройка на RTR-BR (EcoRouterOS):

Войдите в режим конфигурации:

```
configure terminal
```

Включите процесс OSPF с Router ID:

```
router ospf 1
 router-id 192.168.255.2
 exit
```

Настройте анонсирование маршрута по умолчанию (default route) в OSPF:

```
router ospf 1
 default-information originate always
 exit
```

`always` — анонсировать маршрут по умолчанию всегда, даже если у самого RTR-BR нет статического маршрута по умолчанию (он получает его по BGP от ISP) .

Включите OSPF на интерфейсе tunnel.3:

```
interface tunnel.3
 ip ospf 1 area 0
 exit
```

Настройте пассивные интерфейсы (интерфейсы, на которых не нужно устанавливать соседство, но подсети должны анонсироваться):

```
router ospf 1
 passive-interface loopback.0
 passive-interface eth1  ! интерфейс к FW-BR (10.2.0.1) - не пассивный, так как там будет сосед с F
W-BR
 exit
```

Анонсируйте подключённые сети:

```
router ospf 1
 network 10.2.0.0 0.0.0.3 area 0      ! сеть к FW-BR
 network 10.0.3.0 0.0.0.3 area 0       ! сеть tunnel.3 к RTR-COD
 network 192.168.255.2 0.0.0.0 area 0  ! loopback.0
 exit
```

Настройка на RTR-COD (EcoRouterOS):

Войдите в режим конфигурации:

```
configure terminal
```

Включите процесс OSPF с Router ID:

```
router ospf 1
 router-id 192.168.255.3
 exit
```

Включите OSPF на интерфейсе tunnel.3:

```
interface tunnel.3
 ip ospf 1 area 0
 exit
```

Настройте пассивные интерфейсы:

```
router ospf 1
 passive-interface loopback.0
 passive-interface eth1  ! интерфейс к SW-COD (172.16.1.254)
 exit
```

Анонсируйте подключённые сети:

```
router ospf 1
 network 10.0.3.0 0.0.0.3 area 0       ! сеть tunnel.3 к RTR-BR
 network 172.16.0.0 0.0.1.255 area 0   ! сеть центра обработки данных
 network 192.168.255.3 0.0.0.0 area 0  ! loopback.0
 exit
```

Настройка на FW-BR (ViPNet xFirewall):

Настройка OSPF в ViPNet выполняется через веб-интерфейс или CLI.

Войдите в веб-интерфейс FW-BR.
Перейдите в раздел настройки маршрутизации (обычно **Сеть** → **Маршрутизация** → **OSPF**).
Включите OSPF и настройте основные параметры:

Router ID: `10.2.0.2` (IP-адрес интерфейса к RTR-BR)
Process ID: `1`

Настройте интерфейсы для участия в OSPF:

Интерфейс к RTR-BR (`eth0` с IP `10.2.0.2/30`) — область 0
Интерфейс VLAN 10 (`vlan10` с IP `10.2.1.14/28`) — область 0
Интерфейс VLAN 20 (`vlan20` с IP `10.2.2.1/25`) — область 0
Настройте пассивные интерфейсы — в ViPNet обычно все внутренние интерфейсы можно сделать пассивными, так как там нет других маршрутизаторов OSPF:

`vlan10` — пассивный
`vlan20` — пассивный
Сохраните конфигурацию.

Важно: На FW-BR не должно быть статического маршрута по умолчанию. Маршрут по умолчанию должен приходить через OSPF от RTR-BR (благодаря команде `default-information originate always`).

Проверка OSPF

На RTR-BR:

```
show ip ospf neighbor
show ip ospf database
show ip route ospf
```

В выводе `show ip route ospf` должны быть сети:

`10.2.1.0/28` (VLAN 10 от FW-BR)
`10.2.2.0/25` (VLAN 20 от FW-BR)
`172.16.0.0/23` (сеть COD от RTR-COD)

На RTR-COD:

```
show ip ospf neighbor
show ip route ospf
```

Должны отображаться сети от FW-BR.

На FW-BR:
Проверьте таблицу маршрутизации — там должен быть маршрут по умолчанию `0.0.0.0/0` с next-hop `10.2.0.1` (RTR-BR).

b. Настройка статической маршрутизации между HQ и другими площадками

Для связи офиса HQ с офисом BR и центром COD будем использовать статические маршруты через ранее настроенные GRE-туннели.

Планирование статических маршрутов

Направление	Устройство	Сеть назначения	Next-hop (через туннель)
HQ -> BR	FW-HQ	10.2.0.0/30, 10.2.1.0/28, 10.2.2.0/25	10.0.1.2 (RTR-BR, tunnel.1)
HQ -> COD	FW-HQ	172.16.0.0/23, 10.0.3.0/30	10.0.2.2 (RTR-COD, tunnel.2)
BR -> HQ	RTR-BR	10.1.1.0/27, 10.1.1.32/28, 10.1.2.0/24	10.0.1.1 (FW-HQ, tunnel.1)

Направление	Устройство	Сеть назначения	Next-hop (через туннель)
BR -> COD	(уже через OSPF)	-	-
COD -> HQ	RTR-COD	10.1.1.0/27, 10.1.1.32/28, 10.1.2.0/24	10.0.2.1 (FW-HQ, tunnel.2)
COD -> BR	(уже через OSPF)	-	-

Настройка на FW-HQ (Ideco NGFW):

В Ideco NGFW статические маршруты настраиваются через веб-интерфейс.

Войдите в веб-интерфейс FW-HQ.
Перейдите в раздел Сервисы → Маршрутизация → Статические маршруты.
Добавьте маршруты для сетей офиса BR:

Сеть назначения: `10.2.0.0` (маска `/30`)
Шлюз: `10.0.1.2` (IP туннеля RTR-BR)
Интерфейс: `tunnel.1`
Сеть назначения: `10.2.1.0` (маска `/28`)
Шлюз: `10.0.1.2`
Интерфейс: `tunnel.1`
Сеть назначения: `10.2.2.0` (маска `/25`)
Шлюз: `10.0.1.2`
Интерфейс: `tunnel.1`
Добавьте маршруты для сетей центра COD:

Сеть назначения: `172.16.0.0` (маска `/23`)
Шлюз: `10.0.2.2` (IP туннеля RTR-COD)
Интерфейс: `tunnel.2`
Сеть назначения: `10.0.3.0` (маска `/30`)
Шлюз: `10.0.2.2`
Интерфейс: `tunnel.2`
Сохраните конфигурацию.

Настройка на RTR-BR (EcoRouterOS):

Войдите в режим конфигурации:

```
configure terminal
```

Добавьте статические маршруты для сетей HQ:

```
ip route 10.1.1.0 255.255.255.224 10.0.1.1
ip route 10.1.1.32 255.255.255.240 10.0.1.1
ip route 10.1.2.0 255.255.255.0 10.0.1.1
```

Проверьте и сохраните:

```
do show ip route static
write memory
```

Настройка на RTR-COD (EcoRouterOS):

Войдите в режим конфигурации:

```
configure terminal
```

Добавьте статические маршруты для сетей HQ:

```
ip route 10.1.1.0 255.255.255.224 10.0.2.1
ip route 10.1.1.32 255.255.255.240 10.0.2.1
ip route 10.1.2.0 255.255.255.0 10.0.2.1
```

Проверьте и сохраните:

```
do show ip route static
write memory
```

c. Проверка полной связности

Для проверки доступности сетей между площадками используем команду `ping` и `traceroute` .

Проверка с ADM-HQ (офис HQ):

```
# Проверка доступности SRV-BR (офис BR)
ping -c 4 10.2.1.10
traceroute 10.2.1.10

# Проверка доступности SRV1-COD (центр COD)
ping -c 4 172.16.1.1
traceroute 172.16.1.1
```

Проверка с SRV-BR (офис BR):

```
# Проверка доступности ADM-HQ (офис HQ)
ping -c 4 10.1.1.46
traceroute 10.1.1.46

# Проверка доступности SRV1-COD (центр COD)
ping -c 4 172.16.1.1
traceroute 172.16.1.1
```

Проверка с SRV1-COD (центр COD):

```
# Проверка доступности ADM-HQ (офис HQ)
ping -c 4 10.1.1.46
traceroute 10.1.1.46

# Проверка доступности SRV-BR (офис BR)
ping -c 4 10.2.1.10
traceroute 10.2.1.10
```

Ожидаемые результаты:

Все команды ping должны успешно выполняться (0% потери пакетов) .
Traceroute должен показывать путь через соответствующие туннели :

Из HQ в BR: ADM-HQ → FW-HQ (10.1.1.33) → tunnel.1 (10.0.1.1) → RTR-BR (10.0.1.2) → FW-BR (10.2.0.2) → SRV-BR (10.2.1.10)
Из HQ в COD: ADM-HQ → FW-HQ → tunnel.2 → RTR-COD → SW-COD → SRV1-COD

Дополнительная проверка на маршрутизаторах:

На RTR-BR:

```
show ip route
```

Должны отображаться:

Маршруты к сетям HQ (получены статически)
Маршруты к сетям BR (получены через OSPF от FW-BR)
Маршруты к сетям COD (получены через OSPF от RTR-COD)

На RTR-COD:

```
show ip route
```

Должны отображаться:

Маршруты к сетям HQ (получены статически)
Маршруты к сетям BR (получены через OSPF от RTR-BR)
Маршруты к сетям COD (подключённые)

Важные замечания

Default route для FW-BR: Убедитесь, что на FW-BR нет статического маршрута по умолчанию. Он должен получать его через OSPF от RTR-BR .
Пассивные интерфейсы: Все интерфейсы, на которых нет соседей OSPF, должны быть в пассивном режиме. Это повышает безопасность и снижает нагрузку .
Метрики OSPF: При необходимости можно настроить стоимость (cost) интерфейсов для управления выбором пути.
Симметричная маршрутизация: Убедитесь, что обратные маршруты настроены правильно. Например, если пакет идёт из HQ в BR через tunnel.1, обратный путь должен идти также через tunnel.1 .
Проверка MTU: В туннелях GRE возможны проблемы с фрагментацией. При возникновении проблем с пакетами большого размера проверьте MTU на всех интерфейсах туннелей .

8. Настройка облачного хранилища Nextcloud на SRV-BR

В данном пункте мы развернём Nextcloud версии 33.0.0 на сервере **SRV-BR** (Альт Сервер 11) с использованием Apache2 и PostgreSQL, обеспечим доступ по HTTPS с автоматическим перенаправлением и настроим аутентификацию через домен `au.team`.

a. Установка Nextcloud с Apache2 и PostgreSQL

1. Подготовка системы и установка необходимых пакетов

Подключитесь к SRV-BR по SSH или через консоль.
Обновите список пакетов:

```
apt-get update
```

Установите Apache2, PostgreSQL и необходимые модули PHP:

```
apt-get install -y apache2 apache2-mod_ssl postgresql16 postgresql16-contrib php8.2 apache2-mod_php
8.2 php8.2-pgsql php8.2-gd php8.2-json php8.2-xml php8.2-mbstring php8.2-zip php8.2-curl php8.2-int
```

```
l php8.2-bcmath php8.2-gmp php8.2-imagick php8.2-apcu php8.2-pdo php8.2-xml php8.2-openssl php8.2-p
cntl php8.2-xmlreader php8.2-ldap wget unzip
```

Пояснение: устанавливаются Apache, PostgreSQL, расширения PHP для работы с БД, графикой, XML, ZIP и т.д., а также модуль Apache для PHP .

Включите необходимые модули Apache:

```
a2enmod rewrite headers env dir mime setenvif ssl proxy proxy_fcgi
```

Эти модули потребуются для перенаправления с HTTP на HTTPS и для корректной работы Nextcloud .

Перезапустите Apache:

```
systemctl restart apache2
```

Проверьте версию PHP (должна быть не ниже 8.1 для Nextcloud 33):

```
php -v
```

2. Настройка базы данных PostgreSQL

Переключитесь на пользователя postgres:

```
systemctl start postgresql
systemctl status postgresql
ls -la /var/lib/pgsql/data/
su - postgres -s /bin/bash -c "initdb -D /var/lib/pgsql/data"
systemctl start postgresql
systemctl status postgresql
systemctl enable postgresql
su - postgres -s /bin/bash -c "psql"
```

Создайте базу данных и пользователя для Nextcloud:

```
CREATE USER nextcloud WITH PASSWORD 'P@ssw0rd';
ALTER USER nextcloud WITH PASSWORD 'P@ssw0rd';
CREATE DATABASE nextcloud OWNER nextcloud;

```CREATE DATABASE nextcloud;
CREATE USER nextcloud WITH PASSWORD 'P@ssw0rd';
GRANT ALL PRIVILEGES ON DATABASE nextcloud TO nextcloud;
/q
```

Выйдите из консоли PostgreSQL.
/q

If you need to test the connection from the shell:

```
su - postgres -s /bin/bash -c "psql -U nextcloud -d nextcloud -h localhost"
```

3. Загрузка и установка Nextcloud

Перейдите в директорию веб-сервера:

```
cd /var/www
```
```

Скачайте Nextcloud версии 33.0.0:

```
wget https://download.nextcloud.com/server/releases/nextcloud-33.0.0.zip
```

Установите unzip, если его нет, и распакуйте архив:

```
apt-get install unzip
unzip -o nextcloud-33.0.0.zip
```

Удалите архив:

```
rm nextcloud-33.0.0.zip
```

Настройте права доступа:

```
sudo chown -R www-data:www-data /var/www/nextcloud
sudo chmod -R 755 /var/www/nextcloud
```

b. Настройка Apache для HTTPS и перенаправления

1. Создание самоподписанного SSL-сертификата (для соответствия пункту ii)

Так как у нас нет доверенного центра сертификации, создадим самоподписанный сертификат с именем `ncloud.au.team`. Чтобы при обращении по HTTP не возникало проблем с сертификатами, мы создадим свой CA и подпишем сертификат, а затем добавим корневой сертификат на клиентские машины (ADM-HQ, CLI-HQ, CLI-BR). Это имитирует работу с доверенным сертификатом в локальной среде.

На SRV-BR:

Создайте директорию для SSL-сертификатов:

```
sudo mkdir -p /etc/ssl/nextcloud
cd /etc/ssl/nextcloud
```

Создайте корневой сертификат CA (однократно):

```
# Создание закрытого ключа CA
sudo openssl genrsa -out ca.key 4096

# Создание самоподписанного сертификата CA (сроком на 10 лет)
sudo openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.crt -subj "/C=RU/ST=Local
/L=Local/O=AU Team/CN=AU Team Root CA"
```

Создайте сертификат для домена ncloud.au.team:

```
# Создание закрытого ключа для сервера
sudo openssl genrsa -out ncloud.au.team.key 2048

# Создание запроса на сертификат (CSR)
sudo openssl req -new -key ncloud.au.team.key -out ncloud.au.team.csr -subj "/C=RU/ST=Local/L=Local
/O=AU Team/CN=ncloud.au.team"

# Создание конфигурационного файла для расширений X.509 v3
cat <<EOF | sudo tee ncloud.au.team.ext
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
```

```
[alt_names]
DNS.1 = ncloud.au.team
EOF

# Подписание сертификата нашим CA
sudo openssl x509 -req -in ncloud.au.team.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out ncloud.
au.team.crt -days 365 -sha256 -extfile ncloud.au.team.ext

////
sudo mkdir -p /etc/ssl/nextcloud
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/nextcloud/ncloud.au.team.key \
  -out /etc/ssl/nextcloud/ncloud.au.team.crt

echo "<?php phpinfo(); ?>" > /var/www/html/info.php
http://<IP-адрес-вашего-сервера>/info.php


БУЙРУКЪ
apt-get install python3-certbot-dns-rfc2136
```
На DNS-сервере (если он на том же хосте) выполните:
```
tsig-keygen -a HMAC-SHA512 certbot-key
```
Вывод будет примерно таким:
```
key "certbot-key" {
    algorithm hmac-sha512;
    secret "your-secret-key-here==";
};
```
Сохраните этот ключ в файл, например `/etc/bind/certbot.key`.

В `named.conf.local` добавьте:
```
key "certbot-key" {
    algorithm hmac-sha512;
    secret "your-secret-key-here==";
};

zone "_acme-challenge.kbgtk07.ru" {
    type master;
    file "/var/cache/bind/_acme-challenge.kbgtk07.ru.zone";
    allow-query { any; };
    check-names ignore;
    update-policy {
        grant certbot-key name _acme-challenge.kbgtk07.ru. txt;
    };
};
```
Создайте файл зоны `/var/cache/bind/_acme-challenge.kbgtk07.ru.zone`:
```
$ORIGIN .
$TTL 3600
_acme-challenge.kbgtk07.ru IN SOA ns1.kbgtk07.ru. hostmaster.kbgtk07.ru. (
    1 14400 3600 604800 3600 )
    NS ns1.kbgtk07.ru.
```

Проверьте и перезапустите BIND:

```
named-checkconf
systemctl restart named
```

Создайте файл учётных данных для Certbot

Создайте `/etc/letsencrypt/dns-rfc2136.ini`:

```
dns_rfc2136_server = 127.0.0.1
dns_rfc2136_port = 53
dns_rfc2136_name = certbot-key
dns_rfc2136_secret = your-secret-key-here==
dns_rfc2136_algorithm = HMAC-SHA512
```

Установите права:

```
chmod 600 /etc/letsencrypt/dns-rfc2136.ini
```

Получите сертификат

```
certbot certonly --dns-rfc2136 --dns-rfc2136-credentials /etc/letsencrypt/dns-rfc2136.ini -d kbgtk0
7.ru
```

Для получения wildcard-сертификата (для всех поддоменов) добавьте `-d "*.kbgtk07.ru"`.

Сертификаты будут сохранены в `/etc/letsencrypt/live/kbgtk07.ru/`.

Настройте Apache

В вашем файле `/etc/apache2/sites-available/nextcloud.conf` укажите пути к новым сертификатам:

SSLCertificateFile /etc/letsencrypt/live/kbgtk07.ru/fullchain.pem

SSLCertificateKeyFile /etc/letsencrypt/live/kbgtk07.ru/privkey.pem

Вариант 2: Ручной DNS-01 challenge (если вы не управляете DNS)

Если вы не можете настроить RFC2136 (например, DNS управляет провайдер, и у вас нет доступа к настройке динамических обновлений), используйте ручной режим. Вам потребуется один раз добавить TXT-запись в DNS через панель управления вашего регистратора/провайдера.

```
certbot certonly --manual --preferred-challenges dns -d kbgtk07.ru
```

Certbot покажет вам TXT-запись, которую нужно добавить:

_acme-challenge.kbgtk07.ru. 300 IN TXT "some-random-string"

Добавьте эту запись в DNS (через личный кабинет провайдера). После того как запись появится (проверьте через `dig TXT _acme-challenge.kbgtk07.ru`), нажмите Enter в терминале. Сертификат будет получен.

Недостаток: через 90 дней нужно будет повторить процедуру вручную. Для автоматизации потребуется либо API вашего DNS-провайдера (многие поддерживаются через плагины Certbot, например для Cloudflare, DigitalOcean, etc.), либо переход на RFC2136.

Вариант 3: Установка локального BIND и делегирование `_acme-challenge`

Если вы вообще не контролируете DNS `kbgtk07.ru`, но можете развернуть свой DNS-сервер, есть хитрость: вы создаёте зону `_acme-challenge.kbgtk07.ru` на своём сервере и просите администратора DNS

домена `kbgtk07.ru` добавить NS-запись, делегирующую поддомен на ваш сервер. Тогда вы сможете управлять только TXT-записями для этой зоны и использовать RFC2136.

Проверка сертификата и настройка Apache

После получения сертификата (любым способом) убедитесь, что SSL-модуль Apache включён:

a2enmod ssl

systemctl restart httpd2

Отредактируйте конфигурацию Nextcloud, указав новые пути к сертификатам. Убедитесь, что в `ServerName` указано `kbgtk07.ru` (или поддомен, который вы используете). После этого:

a2ensite nextcloud

systemctl reload httpd2

Теперь сайт должен открываться по `https://kbgtk07.ru` без предупреждений — сертификат доверенный.

Если браузер всё ещё показывает ошибку, проверьте:

Что сертификат действительно от Let's Encrypt (просмотрите через `openssl x509 -in /etc/letsencrypt/live/kbgtk07.ru/fullchain.pem -text -noout`).
Что на клиентских машинах установлены актуальные корневые сертификаты (обычно они обновляются автоматически).
Что вы используете правильное имя домена в URL.

yyy2. Настройка виртуального хоста Apache для Nextcloud

Создайте файл конфигурации:

```
mkdir -p /etc/apache2/sites-available
vim /etc/apache2/sites-available/nextcloud.conf
```

Вставьте следующую конфигурацию:

```
<VirtualHost *:80>
    ServerName ncloud.au.team
    Redirect permanent / https://ncloud.au.team/
</VirtualHost>

<VirtualHost *:443>
    ServerName ncloud.au.team
    DocumentRoot /var/www/nextcloud

    Protocols h2 http/1.1

    Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"

    SSLEngine on
    SSLCertificateFile /etc/ssl/nextcloud/ncloud.au.team.crt
    SSLCertificateKeyFile /etc/ssl/nextcloud/ncloud.au.team.key
```

```
    # SSLCertificateChainFile /etc/ssl/nextcloud/ca.crt  # Необязательно, можно добавить для полной цепочки

    <Directory /var/www/nextcloud/>
        Require all granted
        Options FollowSymlinks MultiViews
        AllowOverride All

        <IfModule mod_dav.c>
            Dav off
        </IfModule>

        SetEnv HOME /var/www/nextcloud
        SetEnv HTTP_HOME /var/www/nextcloud
    </Directory>

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/var/run/php/php8.2-fpm.sock|fcgi://localhost"
    </FilesMatch>

    LogLevel warn
    ErrorLog /var/log/apache2/nextcloud_error.log
    CustomLog /var/log/apache2/nextcloud_access.log combined
</VirtualHost>
```

Важно: Настройка `SetHandler` предполагает использование PHP-FPM. В Альт Сервер 11 с пакетным PHP это может быть иначе. Если после включения сайта Nextcloud не работает, измените эту строку на стандартную для модуля PHP:

```
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
```

Или убедитесь, что PHP-FPM установлен и запущен: `sudo systemctl enable --now php8.2-fpm` (версия может отличаться).

Сохраните файл (Ctrl+O, Enter) и выйдите (Ctrl+X).
Включите сайт и отключите стандартный:

```
sudo a2ensite nextcloud.conf
sudo a2dissite 000-default.conf
```

Перезапустите Apache:

```
sudo systemctl restart apache2

su -s /bin/bash apache -c "php /var/www/nextcloud/occ status"

# Статус Nextcloud
su -s /bin/bash apache -c "php /var/www/nextcloud/occ status"

# Выключить режим обслуживания (если включён)
su -s /bin/bash apache -c "php /var/www/nextcloud/occ maintenance:mode --off"
```

```
# Проверить подключение к БД
su -s /bin/bash apache -c "php /var/www/nextcloud/occ db:connect"

# Список приложений
su -s /bin/bash apache -c "php /var/www/nextcloud/occ app:list"

# Обновить .htaccess
su -s /bin/bash apache -c "php /var/www/nextcloud/occ maintenance:update:htaccess"

systemctl restart apache2
```

(критично отсутствие):

PDO и драйверы БД (mysql, pgsql или sqlite) — без них Nextcloud не видит базу данных.
OpenSSL — нужен для шифрования и безопасности.
XMLReader — требуется для обработки конфигурационных файлов.
PCNTL — не обязателен, но полезен для долгих команд (можно игнорировать).
Лимит памяти — рекомендован 512M, текущий ниже (скорее всего 128M).

```
php -v

yum install php-pdo php-mysqlnd php-xml php-openssl php-pcntl
php --ini | grep "Loaded Configuration File"

memory_limit = 512M
systemctl restart apache2
php -m | grep -E "PDO|openssl|xml|pcntl"
su -s /bin/bash apache -c "php /var/www/nextcloud/occ status"
php -m | grep xml
должны увидеть xml, xmlreader, xmlwriter
su -s /bin/bash apache -c "php /var/www/nextcloud/occ status"
su -s /bin/bash apache -c "php /var/www/nextcloud/occ maintenance:install \
  --database=mysql \
  --database-name=nextcloud \
  --database-user=root \
  --database-pass=ваш_пароль \
  --admin-user=admin \
  --admin-pass=admin_пароль"
echo "<?php phpinfo(); ?>" > /var/www/nextcloud/info.php
extension=pgsql.so
extension=pdo_pgsql.so
find /usr -name "pdo_pgsql.so" 2>/dev/null

cd /var/www/nextcloud
ls -ld config
chown -R apache:apache config
chmod 750 config
```

3. Добавление корневого сертификата на клиентские машины

Чтобы браузеры на **ADM-HQ, CLI-HQ, CLI-BR** доверяли нашему сертификату, необходимо импортировать корневой сертификат `ca.crt`.

1. Копирование с одной ВМ на другую

Допустим, вы хотите скопировать файл `report.txt` с машины **А** (пользователь `user1`, IP `192.168.1.10`) в домашнюю директорию пользователя `user2` на машине **Б** (IP `192.168.1.20`).

На машине А выполните команду:

```bash
scp /home/user1/report.txt user2@192.168.1.20:/home/user2/
```

Система запросит пароль пользователя `user2` на машине Б.

2. Копирование с локальной машины на удалённую (и наоборот)

Хотя вы работаете напрямую между двумя ВМ, может быть полезна и обратная операция.

Копирование с локальной ВМ на удалённую:

```bash
scp /путь/к/локальному/файлу пользователь@удалённый_хост:/путь/на/удалённом/хосте/
```

Копирование с удалённой ВМ на локальную:

```bash
scp пользователь@удалённый_хост:/путь/к/файлу /локальный/путь/назначения
```

Для этого просто поменяйте местами источник и получатель.

3. Копирование целой директории

Чтобы скопировать папку `project` со всеми её содержимым, используйте опцию `-r` (рекурсивно).

С машины А на машину Б:

```bash
scp -r /home/user1/project/ user2@192.168.1.20:/home/user2/
```

Важно: путь назначения должен заканчиваться символом `/`, иначе файлы могут быть перезаписаны.

🔧 Возможные проблемы и их решение

Ошибка подключения: `Connection refused`
Это наиболее частая проблема, означающая, что SSH-сервер не запущен на целевой машине или его порт (по умолчанию 22) недоступен.
Убедитесь, что на обеих ВМ запущен сервис `sshd` и что в сетевых настройках (брандмауэр, правила гипервизора) нет блокировки 22-го порта.
Проблемы с аутентификацией
Если система не принимает пароль, проверьте:

Правильность ввода логина и пароля.
В файле конфигурации SSH-сервера (`/etc/openssh/sshd_config`) не отключена ли аутентификация по паролю. Для этого параметр `PasswordAuthentication` должен быть установлен в `yes`.
После изменения конфигурации не забудьте перезагрузить сервис: `systemctl reload sshd`.
Отсутствие команды `scp`
Хотя это маловероятно для серверной ОС, на всякий случай убедитесь, что установлен пакет `openssh-clients`. Он обычно входит в базовую поставку, но может быть удалён.

Скопируйте файл `ca.crt` **с SRV-BR на каждую клиентскую машину** (например, через SCP или USB).
На каждой клиентской машине (Альт Рабочая станция) выполните:

```
# Скопируйте сертификат в системную директорию доверенных
sudo cp ca.crt /usr/local/share/ca-certificates/
# Обновите хранилище доверенных сертификатов
sudo update-ca-certificates
```

После этого браузеры и системные утилиты будут доверять сертификатам, подписанным нашим CA.

c. Доступ по доменному имени ncloud.au.team

Для того чтобы клиенты могли обращаться к Nextcloud по имени `ncloud.au.team`, необходимо настроить DNS.

На SRV-HQ (сервер DNS FreeIPA):

Добавьте A-запись для `ncloud.au.team`, указывающую на IP-адрес SRV-BR (`10.2.1.10`). Это можно сделать через Terraform (как в пункте 4) или вручную через веб-интерфейс FreeIPA.

Пример для добавления через командную строку на SRV-HQ:

```
kinit admin
ipa dnsrecord-add au.team ncloud --a-rec 10.2.1.10
```

Убедитесь, что на клиентах (ADM-HQ, CLI-HQ, CLI-BR) в качестве DNS-сервера указан SRV-HQ (`10.1.1.10`). Тогда они смогут резолвить `ncloud.au.team` в нужный IP.

d. Завершение установки Nextcloud через веб-интерфейс

Откройте браузер на ADM-HQ (или любом клиенте, где импортирован CA) и перейдите по адресу: `https://ncloud.au.team`
Создайте учётную запись администратора Nextcloud (она будет локальной, не доменной). Запомните эти данные.
Настройте подключение к базе данных:

Пользователь БД: `nextcloud`
Пароль БД: `P@ssw0rd`
Имя БД: `nextcloud`
Хост БД: `localhost` (или `127.0.0.1`)
Нажмите **"Завершить установку"**.

e. Настройка аутентификации через LDAP (FreeIPA)

Теперь настроим Nextcloud так, чтобы пользователи из домена `au.team` (группы `hq`, `br`, `cod`) могли входить в систему.

1. Включение приложения LDAP

В веб-интерфейсе Nextcloud нажмите на иконку пользователя в правом верхнем углу и выберите **"Приложения"**.
В левом меню выберите **"Категории"** → **"Интеграция"** (или "Авторизация").

Найдите приложение **"LDAP user and group backend"** и нажмите **"Включить"** .

2. Настройка LDAP-соединения

Перейдите в **"Настройки"** (иконка пользователя → Настройки).
В левом меню, в разделе **"Администрирование"**, выберите **"Интеграция LDAP/AD"** .
На вкладке **"Сервер"** заполните следующие поля :

Хост: `srv-hq.au.team` (или IP-адрес: `10.1.1.10`).
Порт: `389` (для LDAP) или `636` (для LDAPS). Мы будем использовать `389`, так как внутри доверенной сети шифрование не обязательно, но для безопасности лучше настроить LDAPS (тогда потребуется экспорт сертификата FreeIPA).
Пользователь DN: `uid=admin,cn=users,cn=accounts,dc=au,dc=team` (учётная запись администратора домена).
Пароль: `P@ssw0rd` (пароль администратора домена).
Нажмите **"Сохранить учётные данные"**.
База DN (одна на строку): `dc=au,dc=team`.
Нажмите **"Проверить базу поиска DN"**. Должно появиться сообщение "Конфигурация в порядке". Нажмите **"Продолжить"** .
На вкладке **"Пользователи"** можно ничего не менять, просто нажмите **"Продолжить"**.
На вкладке **"Учётные данные"** (Login Attributes) оставьте настройки по умолчанию и нажмите **"Продолжить"**.
На вкладке **"Группы"** нажмите **"Продолжить"**.
На вкладке **"Эксперт"** задайте следующие параметры для корректной работы с FreeIPA :

Внутренний UUID-атрибут для пользователей: `ipaUniqueID`
Внутренний UUID-атрибут для групп: `ipaUniqueID`
Нажмите **"Проверить настройки"**. Должно быть "Конфигурация в порядке".

3. Ограничение доступа только группами hq, br, cod

Вернитесь на вкладку **"Группы"**.
В поле **"Только из этих групп"** выберите группы `hq`, `br` и `cod`. Если они не отображаются, нажмите кнопку обновления списка групп.
Нажмите **"Проверить настройки и пересчитать пользователей"**.
Внизу страницы должно отображаться количество найденных пользователей (например, "x пользователей найдено").

f. Проверка работоспособности

Выйдите из учётной записи администратора Nextcloud.
На странице входа попробуйте войти под одним из доменных пользователей:

Логин: `hq.user1` (или `br.user2`, `cod.user5`)
Пароль: `P@ssw0rd`
Вход должен быть успешным. Пользователь будет автоматически создан в Nextcloud с атрибутами из FreeIPA.

Соответствие требованиям:

Установка без контейнеризации: выполнена, используется нативная установка на SRV-BR.
Apache2 + PostgreSQL: выполнено.
HTTPS с редиректом HTTP→HTTPS: настроено в виртуальном хосте Apache .

Отсутствие проблем с сертификатами: создан самоподписанный CA, сертификат для `ncloud.au.team` подписан им, корневой сертификат CA импортирован на клиентские машины.
Доступ по имени ncloud.au.team: добавлена A-запись в DNS FreeIPA.
Аутентификация доменных пользователей из групп hq, br, cod: настроена интеграция с LDAP FreeIPA и фильтрация по группам .

9. Настройка системы управления конфигурацией Ansible на ADM-HQ

В данном пункте мы установим Ansible в виртуальном окружении Python, создадим инвентарный файл в формате YAML с группами `proxy` и `server`, а также настроим доступ по SSH-ключам к узлам `ha1-cod`, `ha2-cod`, `srv1-cod`, `srv2-cod`, `srv3-cod`.

а. Установка Ansible в виртуальном окружении на ADM-HQ

На **ADM-HQ (Альт Рабочая станция)** необходимо установить Ansible через `pip` внутри виртуального окружения.

Пошаговая настройка:

Подготовка системы:

Убедитесь, что в системе установлены `python3`, `pip` и `virtualenv`. Если нет, установите их:

```
sudo apt-get update
sudo apt-get install python3 python3-pip python3-virtualenv
```

Создание директории и виртуального окружения:

```
mkdir -p /home/user/ansible
cd /home/user/ansible
python3 -m venv venv/ansible
```

`venv/ansible` — путь к виртуальному окружению, как требует задание .

Активация виртуального окружения:

```
source venv/ansible/bin/activate
```

После активации приглашение командной строки должно измениться, например, на `(ansible) user@adm-hq:~$` .

Установка Ansible с помощью pip:

```
pip install ansible
```

По умолчанию установится последняя стабильная версия .

Проверка установки:

```
ansible --version
```

Вывод должен содержать информацию о версии Ansible и путь к интерпретатору Python внутри виртуального окружения (например, `/home/user/ansible/venv/ansible/bin/python`).

Деактивация окружения (при необходимости):

```
deactivate
```

Для работы с Ansible в дальнейшем потребуется каждый раз активировать окружение командой `source /home/user/ansible/venv/ansible/bin/activate`.

b. Настройка доступа по SSH-ключам к узлам COD

По условию, доступ ко всем узлам должен осуществляться на основе ключевой пары; доступ по паролю не допускается.

На ADM-HQ:

Сгенерируйте SSH-ключ (если ещё нет):

```
ssh-keygen -t rsa -b 4096
```

Нажмите Enter для сохранения в стандартном расположении (`~/.ssh/id_rsa`). Можно задать пароль для ключа (passphrase) или оставить пустым.

Скопируйте публичный ключ на каждый управляемый узел:

Для `ha1-cod` (172.16.0.1)
Для `ha2-cod` (172.16.0.2)
Для `srv1-cod` (172.16.1.1)
Для `srv2-cod` (172.16.1.2)
Для `srv3-cod` (172.16.1.3)

Используйте команду `ssh-copy-id` (если доступна) или вручную скопируйте ключ. Например, для первого узла:

```
ssh-copy-id user@172.16.0.1
```

Здесь `user` — имя административной учётной записи на целевых узлах (например, `root` или `net_admin`). В задании не указано явно, но по логике предыдущих пунктов можно использовать `net_admin`.

Повторите для всех пяти узлов. При первом подключении потребуется ввести пароль пользователя, затем ключ будет добавлен в `~/.ssh/authorized_keys`.

Проверка доступа:

Попробуйте подключиться по SSH к каждому узлу:

```
ssh net_admin@172.16.0.1
```

Пароль запрашиваться не должен.

c. Создание инвентарного файла в формате YAML

Инвентарный файл должен находиться по пути `/home/user/ansible/inventories/production/hosts` и быть в формате YAML.

Создайте необходимые директории:

```
mkdir -p /home/user/ansible/inventories/production
cd /home/user/ansible/inventories/production
```

Создайте и отредактируйте файл `hosts`:

```
nano hosts
```

Вставьте следующее содержимое:

```
all:
  children:
    proxy:
      hosts:
        ha1-cod:
          ansible_host: 172.16.0.1
          ansible_user: net_admin
          ansible_python_interpreter: /usr/bin/python3
        ha2-cod:
          ansible_host: 172.16.0.2
          ansible_user: net_admin
          ansible_python_interpreter: /usr/bin/python3
    server:
      hosts:
        srv1-cod:
          ansible_host: 172.16.1.1
          ansible_user: net_admin
          ansible_python_interpreter: /usr/bin/python3
        srv2-cod:
          ansible_host: 172.16.1.2
          ansible_user: net_admin
          ansible_python_interpreter: /usr/bin/python3
        srv3-cod:
          ansible_host: 172.16.1.3
          ansible_user: net_admin
          ansible_python_interpreter: /usr/bin/python3
```

Пояснения:

`all` — корневая группа, содержащая все узлы .
`children` — вложенные группы `proxy` и `server`, как требует задание.
В каждой группе перечислены узлы с их параметрами:

`ansible_host` — IP-адрес узла (согласно таблице адресации из пункта 1).
`ansible_user` — имя пользователя для подключения по SSH (используем `net_admin`, созданного в пункте 1b).
`ansible_python_interpreter` — явное указание интерпретатора Python на целевой системе (для Alt Server 11 это `/usr/bin/python3`) .

Сохраните файл (Ctrl+O, Enter) и выйдите (Ctrl+X).

Проверьте синтаксис инвентарного файла:

```
ansible-inventory -i hosts --list
```

Должен отобразиться JSON-вывод со структурой вашего инвентаря. Это подтверждает, что файл написан корректно.

d. Проверка доступности узлов с помощью `ansible ping`

Теперь выполним финальную проверку. Убедитесь, что виртуальное окружение активировано.

Активируйте виртуальное окружение (если ещё не активировано):

```
source /home/user/ansible/venv/ansible/bin/activate
```

Перейдите в директорию с инвентарным файлом:

```
cd /home/user/ansible/inventories/production
```

Выполните команду `ansible ping` для всех узлов:

```
ansible -i hosts -m ping all
```

Ожидаемый результат:

```
ha1-cod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ha2-cod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
srv1-cod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
srv2-cod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
srv3-cod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

`SUCCESS` и `"ping": "pong"` означают, что Ansible успешно подключился к узлу, аутентифицировался и проверил доступность Python .
Если какой-то узел не отвечает, проверьте:

Доступность по сети (ping с ICMP).
Правильность SSH-ключа.

Указанный пользователь существует на целевой системе.
Интерпретатор Python находится по указанному пути .

Важно: Если при выполнении `ansible ping` возникают предупреждения о том, что интерпретатор Python не найден, убедитесь, что в инвентарном файле правильно указан параметр `ansible_python_interpreter` . Для Alt Server 11 это обычно `/usr/bin/python3`.

Результат

Ansible успешно установлен в виртуальном окружении `/home/user/ansible/venv/ansible`.
Создан инвентарный файл в формате YAML по требуемому пути с группами `proxy` и `server`.
Настроен доступ по SSH-ключам ко всем пяти узлам COD.
Команда `ansible -i inventories/production/hosts -m ping all` выполняется без ошибок, все узлы отвечают `pong`.

10. Настройка веб-портала в центре обработки данных

Подготовка

На **ADM-HQ** убедитесь, что виртуальное окружение Ansible активировано, и вы находитесь в директории `/home/user/ansible`. Инвентарный файл уже создан в `/home/user/ansible/inventories/production/hosts`.

Для работы HTTPS с доверенным сертификатом создадим сертификат для домена `www.au.team`, подписанный тем же корневым CA, который использовался для Nextcloud. Если корневой CA ещё не скопирован на ADM-HQ, выполните:

```
# Скопировать ключи CA с SRV-BR (предполагается, что они доступны по SSH)
scp user@10.2.1.10:/etc/ssl/nextcloud/ca.* /home/user/ansible/
```

Затем сгенерируйте сертификат для веб-портала:

```
cd /home/user/ansible
openssl genrsa -out www.au.team.key 2048
openssl req -new -key www.au.team.key -out www.au.team.csr -subj "/C=RU/ST=Local/L=Local/O=AU Team/CN=www.au.team"
openssl x509 -req -in www.au.team.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out www.au.team.crt -days 365 -sha256
cat www.au.team.crt www.au.team.key > www.au.team.pem
```

Теперь создадим необходимые директории для шаблонов:

```
mkdir -p /home/user/ansible/templates
```

a. Playbook `playbook1_keepalived.yml`

Файл: `/home/user/ansible/playbook1_keepalived.yml`

```
---
- name: Install and configure keepalived for proxy nodes
  hosts: proxy
  become: yes
```

```yaml
  vars:
    vip: "172.16.1.253"
    vip_prefix: "23"
    interface: "eth0"  # предполагается, что интерфейс для VIP называется eth0
  tasks:
    - name: Install keepalived
      apt:
        name: keepalived
        state: present
        update_cache: yes

    - name: Set keepalived state and priority based on hostname
      set_fact:
        keepalived_state: "{{ 'MASTER' if inventory_hostname == 'ha1-cod' else 'BACKUP' }}"
        keepalived_priority: "{{ 100 if inventory_hostname == 'ha1-cod' else 50 }}"

    - name: Deploy keepalived configuration
      template:
        src: keepalived.conf.j2
        dest: /etc/keepalived/keepalived.conf
        owner: root
        group: root
        mode: 0644
      notify: restart keepalived

    - name: Ensure keepalived is enabled and running
      systemd:
        name: keepalived
        enabled: yes
        state: started

  handlers:
    - name: restart keepalived
      systemd:
        name: keepalived
        state: restarted
```

Шаблон: `/home/user/ansible/templates/keepalived.conf.j2`

```
global_defs {
   notification_email {
     root@localhost
   }
   notification_email_from keepalived@localhost
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   router_id {{ ansible_hostname }}
   vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}
```

```
vrrp_instance VI_1 {
    state {{ keepalived_state }}
    interface {{ interface }}
    virtual_router_id 51
    priority {{ keepalived_priority }}
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1234
    }
    virtual_ipaddress {
        {{ vip }}/{{ vip_prefix }} dev {{ interface }}
    }
}
```

b. Playbook `playbook2_web.yml`

Файл: `/home/user/ansible/playbook2_web.yml`

```
---
- name: Install and configure Angie web server on server nodes
  hosts: server
  become: yes
  tasks:
    - name: Install Angie
      apt:
        name: angie
        state: present
        update_cache: yes

    - name: Ensure Angie is enabled and running
      systemd:
        name: angie
        enabled: yes
        state: started

    - name: Create index.html with hostname
      copy:
        content: "{{ ansible_hostname }} by Angie!\n"
        dest: /var/www/html/index.html   # при необходимости замените на правильный путь
        owner: root
        group: root
        mode: 0644
      notify: restart angie

  handlers:
    - name: restart angie
      systemd:
        name: angie
        state: restarted
```

Примечание: Путь `/var/www/html` может отличаться в зависимости от конфигурации Angie. Проверьте стандартный корневой каталог веб-сервера после установки (обычно `/usr/share/angie/html` или `/var/www/angie`). При необходимости скорректируйте `dest`.

c. Playbook `playbook3_haproxy.yml`

Файл: `/home/user/ansible/playbook3_haproxy.yml`

```yaml
---
- name: Gather facts from server nodes (for backend IPs)
  hosts: server
  gather_facts: yes
  tasks:
    - name: Just gathering facts
      debug:
        msg: "Facts collected"

- name: Install and configure HAProxy on proxy nodes
  hosts: proxy
  become: yes
  vars:
    ssl_cert_src: "/home/user/ansible/www.au.team.pem"
    ssl_cert_dest: "/etc/ssl/private/www.au.team.pem"
  tasks:
    - name: Install HAProxy
      apt:
        name: haproxy
        state: present
        update_cache: yes

    - name: Ensure SSL private directory exists
      file:
        path: /etc/ssl/private
        state: directory
        mode: 0710
        owner: root
        group: haproxy

    - name: Copy SSL certificate (PEM) to proxy
      copy:
        src: "{{ ssl_cert_src }}"
        dest: "{{ ssl_cert_dest }}"
        owner: root
        group: haproxy
        mode: 0640
      notify: restart haproxy

    - name: Deploy HAProxy configuration
      template:
        src: haproxy.cfg.j2
        dest: /etc/haproxy/haproxy.cfg
```

```
        owner: root
        group: root
        mode: 0644
      notify: restart haproxy

    - name: Ensure HAProxy is enabled and running
      systemd:
        name: haproxy
        enabled: yes
        state: started

  handlers:
    - name: restart haproxy
      systemd:
        name: haproxy
        state: restarted
```

Шаблон: `/home/user/ansible/templates/haproxy.cfg.j2`

```
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY
1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY13
05_SHA256
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client  50000
    timeout server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
```

```
    errorfile 504 /etc/haproxy/errors/504.http

frontend http-in
    bind *:80
    mode http
    redirect scheme https code 301 if !{ ssl_fc }

frontend https-in
    bind *:443 ssl crt /etc/ssl/private/www.au.team.pem
    mode http
    default_backend web_servers

backend web_servers
    balance roundrobin
    option httpchk HEAD / HTTP/1.1\r\nHost:\ www.au.team
{% for host in groups['server'] %}
    server {{ host }} {{ hostvars[host]['ansible_host'] }}:80 check
{% endfor %}

listen stats
    bind *:9000
    mode http
    stats enable
    stats uri /haproxy_stats
    stats refresh 10s
    stats auth admin:P@ssw0rd
```

d. Идемпотентность

Все задачи используют модули Ansible, поддерживающие идемпотентность (`apt`, `template`, `copy`, `systemd`). При повторном запуске каждого playbook статус задач будет `ok`, если конфигурация не менялась.

e. Доступ по HTTPS и настройка DNS

Добавьте A-запись для `www.au.team` в DNS FreeIPA, указывающую на VIP `172.16.1.253`:

```
# На SRV-HQ или через Terraform
kinit admin
ipa dnsrecord-add au.team www --a-rec 172.16.1.253
```

Импортируйте корневой сертификат CA на клиентские устройства, если это ещё не сделано (для Nextcloud мы уже импортировали `ca.crt`). Это обеспечит доверие к сертификату `www.au.team`.
Проверка:

После выполнения всех playbook убедитесь, что keepalived настроил VIP на одном из proxy-серверов (проверьте командой `ip a` на ha1-cod или ha2-cod).
С любого клиента (ADM-HQ, CLI-HQ, CLI-BR) выполните:

```
curl -k https://www.au.team
```

Должен вернуться ответ от одного из серверов группы server (например, `srv1-cod by Angie!`).
Ключ `-k` можно не использовать, так как сертификат доверенный.

Проверьте статистику HAProxy: `http://www.au.team:9000/haproxy_stats` (авторизация `admin`:`P@ssw0rd`).
Убедитесь, что HTTP автоматически перенаправляется на HTTPS:

```
curl -L http://www.au.team
```

f. Запуск playbook

Активируйте виртуальное окружение и выполните playbook в любом порядке (рекомендуется сначала `playbook2_web.yml`, затем `playbook1_keepalived.yml` и `playbook3_haproxy.yml`):

```
source /home/user/ansible/venv/ansible/bin/activate
cd /home/user/ansible

ansible-playbook -i inventories/production/hosts playbook2_web.yml
ansible-playbook -i inventories/production/hosts playbook1_keepalived.yml
ansible-playbook -i inventories/production/hosts playbook3_haproxy.yml
```

После успешного выполнения всех playbook веб-портал будет доступен по адресу `https://www.au.team` с балансировкой нагрузки и отказоустойчивостью.

11. Настройка личного кабинета и портала SSL VPN на FW-HQ

Подготовка

Перед выполнением настроек убедитесь, что выполнены предыдущие пункты:

FW-HQ введён в домен `au.team` (пункт 4c).
Пользователи из групп `hq`, `br`, `cod` импортированы в FW-HQ (пункт 4c.iv).
Сервисы `ncloud.au.team` (Nextcloud, пункт 8) и `www.au.team` (веб-портал, пункт 10) доступны из внутренних сетей.

a. Публикация личного кабинета пользователя

Для доступа к личному кабинету по HTTPS с автоматическим перенаправлением с HTTP необходимо настроить обратный прокси на FW-HQ .

Войдите в веб-интерфейс FW-HQ.
Создайте профиль доступа к личному кабинету (если не создан):

Перейдите в раздел **Пользователи** → **Личный кабинет пользователя** .
Нажмите **Добавить**.
Название: `Доступ к ЛК для всех доменных пользователей`
Пользователи и группы: выберите импортированные группы `hq`, `br`, `cod` (или выберите родительскую группу `FreeIPA-Users`, если создавали).
Сохраните профиль.
Опубликуйте личный кабинет через обратный прокси:

Перейдите в раздел **Сервисы** → **Обратный прокси** .
Нажмите **Добавить**.
Включите опцию `Внутренний сервис Ideco NGFW` .
Запрашиваемый адрес в интернете: укажите два значения (можно добавить оба в одном правиле, либо создать два правила):

`lk.au.team` (для доступа по доменному имени)
`63.27.18.18` (публичный IP-адрес FW-HQ)
Протокол: выберите `HTTPS` (опция автоматического редиректа с HTTP на HTTPS включается автоматически при публикации через обратный прокси) .
Нажмите **Добавить**.
Настройте DNS-запись для `lk.au.team`:
На сервере SRV-HQ (FreeIPA) добавьте A-запись, указывающую на публичный IP FW-HQ:

```
kinit admin
ipa dnsrecord-add au.team lk --a-rec 63.27.18.18
```

Либо добавьте запись через Terraform, аналогично пункту 4d.

Проверка:

С любого клиента (ADM-HQ, CLI-HQ, CLI-BR, OUT-CLI) откройте браузер и перейдите по адресу `http://lk.au.team`. Должно произойти автоматическое перенаправление на `https://lk.au.team`.
Авторизуйтесь под любым доменным пользователем (например, `hq.user1` с паролем `P@ssw0rd`).
Должен открыться личный кабинет пользователя Ideco NGFW .

b. Настройка ресурсов SSL VPN для доступа из личного кабинета

Для организации доступа к внутренним ресурсам через личный кабинет используется функционал VPN-подключений Ideco NGFW . Настроим два правила доступа по VPN с разными источниками подключения.

1. Общая настройка VPN-сервера

Перейдите в раздел Пользователи → VPN-подключения .
На вкладке **Основное** настройте параметры:

Сеть для VPN-подключений: укажите подсеть, из которой будут выдаваться IP-адреса VPN-клиентам. Например, `10.128.0.0/16` (это стандартная сеть для VPN в Ideco) .
Зона: оставьте пустым или укажите зону `VPN` (если создавали).
Длина сессии пользователя, час: укажите `24` (или нужное значение).
DNS-суффикс: `au.team`.
Включите протоколы, необходимые для работы SSL VPN (TLS или WireGuard) .

2. Создание правил доступа к ресурсам

Создадим два правила доступа по VPN с разными источниками подключения .

Правило 1: Доступ к обоим ресурсам из офисов HQ и BR

Поле	Значение
Название	`VPN Access to All Resources from Offices`
Источники подключения	Подсети офисов HQ и BR: `10.1.1.0/27`, `10.1.1.32/28`, `10.1.2.0/24`, `10.2.1.0/28`, `10.2.2.0/25` (можно объединить в один объект "Офисные сети")

Поле	Значение
Пользователи и группы	Импортированные группы `hq`, `br`, `cod` (или родительская группа `FreeIPA-Users`)
Протоколы подключения	Любой (или выберите TLS/WireGuard)
Доступ по VPN	Разрешить
Способ 2FA	Не требуется (или настройте по необходимости)

Правило 2: Доступ только к www.au.team из OUT-CLI

Поле	Значение
Название	`VPN Access to www only from OUT-CLI`
Источники подключения	Подсеть `34.35.36.0/26` (сеть OUT-CLI)
Пользователи и группы	Импортированные группы `hq`, `br`, `cod`
Протоколы подключения	Любой
Доступ по VPN	Разрешить
Способ 2FA	Не требуется

Важно: Правила применяются сверху вниз до первого совпадения . Разместите правило для офисных сетей выше правила для OUT-CLI, чтобы пользователи из офисов попадали под него первым.

3. Настройка передачи маршрутов клиентам

Чтобы VPN-клиенты знали, как достичь внутренних ресурсов, настроим передачу маршрутов .

В разделе **Пользователи** → **VPN-подключения** перейдите на вкладку **Передача маршрутов**.
Нажмите **Добавить**.
Для каждого ресурса создайте правило:

Для ncloud.au.team (доступен только из офисов):

Название: `Route to Nextcloud`
Пользователи и группы: оставьте пустым (применяется ко всем) ИЛИ укажите группы `hq`, `br`, `cod`
Передаваемые маршруты: `10.2.1.0/28` (сеть SRV-BR)

Для www.au.team:

Название: `Route to Web Portal`
Пользователи и группы: оставьте пустым
Передаваемые маршруты: `172.16.0.0/23` (сеть COD, где расположен VIP 172.16.1.253)

4. Настройка статической привязки IP-адресов (опционально)

Если требуется, чтобы определённые пользователи всегда получали один и тот же IP-адрес при VPN-подключении, настройте статическую привязку на вкладке **Выдача IP-адресов** .

5. Загрузка корневого сертификата для клиентов

Для работы SSL VPN клиентам может потребоваться корневой сертификат FW-HQ . Его можно скачать из личного кабинета пользователя.

c. Проверка работоспособности

Проверка из офисных сетей (ADM-HQ, CLI-HQ, CLI-BR)

На любом устройстве в сети HQ или BR (например, ADM-HQ) откройте браузер и перейдите в личный кабинет: `https://lk.au.team`.
Авторизуйтесь под доменным пользователем (например, `hq.user1`).
В личном кабинете найдите раздел для скачивания VPN-скриптов или настройки VPN-подключения .
Настройте VPN-подключение к FW-HQ (используйте Ideco Client или встроенные средства ОС) .
После подключения проверьте доступность обоих ресурсов:

```
curl https://ncloud.au.team
curl https://www.au.team
```

Оба должны быть доступны.

Проверка из сети OUT-CLI

На устройстве OUT-CLI (Симпли Линукс) откройте браузер и перейдите в личный кабинет: `https://lk.au.team`.
Авторизуйтесь под доменным пользователем.
Настройте VPN-подключение к FW-HQ (аналогично).
После подключения проверьте доступность ресурсов:

```
curl https://www.au.team      # должен быть доступен
curl https://ncloud.au.team   # должен быть недоступен (таймаут или отказ в соединении)
```

Важные замечания

Порядок правил: Убедитесь, что правило для офисных сетей находится выше правила для OUT-CLI в таблице правил доступа по VPN .
Импортированные пользователи: Для авторизации в личном кабинете используются импортированные из домена пользователи. Убедитесь, что синхронизация с FreeIPA произошла .
Двухфакторная аутентификация: При необходимости можно настроить 2FA для VPN-подключений .
HTTPS-редирект: При публикации личного кабинета через обратный прокси с протоколом HTTPS автоматически включается редирект с HTTP на HTTPS .
Корневой сертификат: Для работы VPN-клиентов может потребоваться установка корневого сертификата FW-HQ на клиентские устройства .

12. Настройка удалённого доступа через Ideco Client

В данном пункте необходимо обеспечить возможность подключения к офису HQ из внешней сети (OUT-CLI) с помощью Ideco Client для доменного пользователя `br.user4`. Настройка включает две части: конфигурация VPN-сервера на **FW-HQ** и установка/настройка клиента на **OUT-CLI**.

а. Настройка VPN-сервера на FW-HQ для Ideco Client

Ideco Client использует для подключения протоколы **TLS VPN** или **WireGuard**, встроенные в Ideco NGFW . Убедимся, что на FW-HQ разрешены соответствующие протоколы и создано правило доступа для пользователя `br.user4`.

1. Проверка общих настроек VPN

Войдите в веб-интерфейс **FW-HQ**.
Перейдите в раздел **Пользователи** → **VPN-подключения** .
На вкладке **Основное** убедитесь, что:

Сеть для VPN-подключений задана (например, `10.128.0.0/16`).
Зона оставлена пустой (или указана зона `VPN`).
Длина сессии пользователя – например, `24` часа.
DNS-суффикс – `au.team`.
Включены протоколы, необходимые для Ideco Client: как минимум **TLS VPN** (обычно используется по умолчанию) или **WireGuard** . Рекомендуется оставить оба включёнными.

2. Создание правила доступа для пользователя br.user4 из сети OUT-CLI

В пункте 11 мы уже создали правило `VPN Access to www only from OUT-CLI`, которое разрешает VPN-подключения из подсети `34.35.36.0/26` для всех доменных пользователей из групп `hq`, `br`, `cod`. Пользователь `br.user4` входит в группу `br`, поэтому дополнительных правил создавать не требуется. Однако убедимся, что это правило настроено корректно:

В разделе **Пользователи** → **VPN-подключения** перейдите на вкладку **Правила доступа** .
Найдите правило для OUT-CLI (например, с названием `VPN Access to www only from OUT-CLI`) и отредактируйте его при необходимости:

Источники подключения: `34.35.36.0/26` (сеть OUT-CLI).
Пользователи и группы: выберите группу `br` (или оставьте все группы `hq`, `br`, `cod`, чтобы правило применялось ко всем доменным пользователям).
Протоколы подключения: выберите `TLS VPN` и/или `WireGuard` – те протоколы, которые будет использовать Ideco Client.
Доступ по VPN: `Разрешить`.
Сохраните правило.

3. Проверка, что порты для VPN открыты на межсетевом экране

Ideco NGFW автоматически создаёт необходимые правила файрвола при включении VPN-сервера. Однако для уверенности можно проверить в разделе **Правила трафика** → **Файрвол**, что разрешён входящий трафик на соответствующие порты:

TLS VPN: порт `443` (TCP) на внешнем интерфейсе FW-HQ.
WireGuard: порт `51820` (UDP) по умолчанию (может быть изменён в настройках VPN).

Если правила отсутствуют, добавьте их вручную:

Источник: `any`
Назначение: внешний IP FW-HQ (`63.27.18.18`)
Протокол: `TCP/443` и/или `UDP/51820`
Действие: `Разрешить`

b. Настройка Ideco Client на OUT-CLI

1. Установка Ideco Client

На **OUT-CLI (Симпли Линукс 11)** необходимо установить клиентское ПО. Дистрибутив можно скачать из личного кабинета пользователя на FW-HQ или из официального репозитория Ideco.

Способ 1: Скачивание через браузер

На OUT-CLI откройте браузер и перейдите по адресу `https://lk.au.team` (или по публичному IP FW-HQ: `https://63.27.18.18`).
Авторизуйтесь под любым доменным пользователем (например, `br.user4` ещё не настроен, можно использовать `admin`, если помните пароль, либо временно использовать другого пользователя; но для скачивания дистрибутива авторизация не обязательна, обычно файлы доступны в разделе "Помощь" или "Скачать клиент").
Найдите раздел загрузки **Ideco Client** для Linux и скачайте подходящий пакет (`.deb` или `.rpm` в зависимости от версии Симпли Линукс; Симпли Линукс 11 основан на Debian, поэтому нужен `.deb`).

Способ 2: Использование wget (если известен прямой URL)

```
cd /tmp
wget https://lk.au.team/ideco-client-latest.deb   # примерный URL, уточните на своём FW-HQ
```

Установка пакета:

```
sudo apt-get update
sudo apt-get install ./ideco-client-latest.deb
```

2. Создание профиля подключения

После установки запустите Ideco Client. При первом запуске потребуется создать профиль.

Откройте терминал и выполните команду `ideco-client` или найдите приложение в меню.
В окне клиента нажмите **"Добавить подключение"** (или аналогичную кнопку).
Заполните поля профиля:

Название профиля: `VPN to HQ`
Адрес сервера (Хост): `63.27.18.18` (публичный IP FW-HQ) или `lk.au.team` (если DNS-имя разрешается из OUT-CLI; для этого необходимо, чтобы OUT-CLI использовал DNS-сервер, способный разрешить `lk.au.team` – например, публичный DNS или указание сервера SRV-HQ, но из OUT-CLI он может быть недоступен, поэтому лучше использовать IP-адрес).
Логин: `br.user4`
Пароль: `P@ssw0rd`
Протокол: выберите `TLS VPN` (или тот, который включён на сервере). Обычно автоматически определяется.
Сохраните профиль.

3. Подключение к VPN

В окне Ideco Client выберите созданный профиль и нажмите **"Подключиться"**.
После успешной аутентификации клиент установит туннель. В логах должно появиться сообщение об успешном подключении и полученном IP-адресе из пула VPN (например, `10.128.0.x`).
Проверьте доступность ресурса, разрешённого для OUT-CLI:

```
curl https://www.au.team
```

Должен отобразиться ответ от веб-сервера (например, "srv1-cod by Angie!" или аналогичный).

Проверьте, что ресурс `ncloud.au.team` недоступен:

```
curl https://ncloud.au.team
```

Ожидается ошибка соединения (таймаут или отказ).

4. Дополнительные настройки (при необходимости)

Если подключение не устанавливается, проверьте на FW-HQ журнал VPN-подключений в разделе **Мониторинг → Активные VPN-подключения** или **Журнал событий**.
Убедитесь, что на OUT-CLI нет блокирующего брандмауэра, разрешён исходящий трафик на порты 443/tcp (для TLS VPN) или 51820/udp (для WireGuard).
При использовании доменного имени `lk.au.team` для подключения, добавьте в файл `/etc/hosts` на OUT-CLI запись:

```
63.27.18.18   lk.au.team
```

Итог

Настроен VPN-доступ через Ideco Client для доменного пользователя `br.user4` из внешней сети OUT-CLI. Подключение успешно устанавливается, и после подключения пользователь получает доступ только к ресурсу `www.au.team` согласно ранее созданному правилу. Для пользователей из офисных сетей доступ к обоим ресурсам (`ncloud.au.team` и `www.au.team`) остаётся без изменений.