

09.02.06



Таблица 1. Требования к ресурсам и гостевым ОС

Машина	RAM, ГБ	CPU	HDD/SDD, ГБ	OS
ISP	1	1	10	ОС Альт JeOS/Linux или аналог
HQ-RTR	1	1	10	ОС EcoRouter или аналог
BR-RTR	1	1	10	ОС EcoRouter или аналог
HQ-SRV	2	1	10	ОС Альт Сервер/аналог
BR-SRV	2	1	10	ОС Альт Сервер/аналог
HQ-CLI	3	2	15	ОС Альт Рабочая Станция/аналог
Итого	10	7	65	-

Таблица 2. Таблица имен

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

МОДУЛЬ 1

1.Произведите базовую настройку устройств

- Настройте имена устройств согласно топологии. Используйте полное доменное имя
- На всех устройствах необходимо сконфигурировать IPv4
- IP-адрес должен быть из приватного диапазона, в случае, если сеть локальная, согласно RFC1918
- Локальная сеть в сторону HQ-SRV(VLAN100) должна вмещать не более 64 адресов
- Локальная сеть в сторону HQ-CLI(VLAN200) должна вмещать не более 16 адресов
- Локальная сеть в сторону BR-SRV должна вмещать не более 32 адресов
- Локальная сеть для управления(VLAN999) должна вмещать не более 8 адресов
- Сведения об адресах занесите в отчёт, в качестве примера используйте Таблицу 3

2. Настройка ISP

• Настройте адресацию на интерфейсах:

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

3. Создание локальных учетных записей

• Создайте пользователя sshuser на серверах HQ-SRV и BR-SRV

- Пароль пользователя sshuser с паролем P@ssw0rd
- Идентификатор пользователя 1010
- Пользователь sshuser должен иметь возможность запускать sudo без дополнительной аутентификации.\

• Создайте пользователя net_admin на маршрутизаторах HQ-RTR и BR-RTR

- Пароль пользователя net_admin с паролем P@\$s\$word
- При настройке ОС на базе Linux, запускать sudo без дополнительной аутентификации

4. Настройте на интерфейсе HQ-RTR в сторону офиса HQ виртуальный коммутатор:

- Сервер HQ-SRV должен находиться в ID VLAN 100
- Клиент HQ-CLI в ID VLAN 200
- Создайте подсеть управления с ID VLAN 999
- Основные сведения о настройке коммутатора и выбора реализации разделения на VLAN занесите в отчет

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

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

6. Между офисами HQ и BR необходимо сконфигурировать ip туннель

- Сведения о туннеле занесите в отчет
- На выбор технологии GRE

7. Обеспечьте динамическую маршрутизацию: ресурсы одного офиса должны быть доступны из другого офиса. Для обеспечения динамической маршрутизации используйте link state протокол на ваше усмотрение.

- Разрешите выбранный протокол только на интерфейсах в ip туннеле
- Маршрутизаторы должны делиться маршрутами только друг с другом
- Обеспечьте защиту выбранного протокола посредством парольной защиты
- Сведения о настройке и защите протокола занесите в отчет

8. Настройка динамической трансляции адресов.

- Настройте динамическую трансляцию адресов для обоих офисов.

- Все устройства в офисах должны иметь доступ к сети Интернет
9. Настройка протокола динамической конфигурации хостов.
- Настройте нужную подсеть
 - Для офиса HQ в качестве сервера DHCP выступает маршрутизатор HQ-RTR.
 - Клиентом является машина HQ-CLI.
 - Исключите из выдачи адрес маршрутизатора
 - Адрес шлюза по умолчанию – адрес маршрутизатора HQ-RTR
 - Адрес DNS-сервера для машины HQ-CLI – адрес сервера HQ-SRV
 - DNS-суффикс для офисов HQ – au-team.irpo
 - Сведения о настройке протокола занесите в отчёт
10. Настройка DNS для офисов HQ и BR
- Основной DNS-сервер реализован на HQ-SRV.
 - Сервер должен обеспечивать разрешение имён в сетевые адреса устройств и обратно в соответствии с таблицей 2
 - В качестве DNS сервера пересылки используйте любой общедоступный DNS сервер
11. Настройте часовой пояс на всех устройствах, согласно месту проведения экзамена.

РЕШЕНИЕ МОДУЛЬ 1

конфигурация доменного имени и имени устройства

Изменяем в файле /etc/hostname <server-hostname> на необходимый хостнейм В

файле /etc/hosts меняем строчку 127.0.1.1 <server-hostname> на 127.0.1.1 <domain>

<hostname> Пример:

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    br-rtr.au-team.irpo BR-RTR

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

В файле /etc/resolv.conf меняем строку domain на наш домен.

Пример:

```
GNU nano 7.2 /etc/resolv.conf *
domain au-team.irpo_
search localdomain
nameserver 192.168.111.2
```

Сетевая конфигурация

Приватные сети:

От 10.0.0.0 до 10.255.255.255

От 172.16.0.0 до 172.31.255.255

От 192.168.0.0 до 192.168.255.255

Настройка ip-адреса

В файле /etc/network/interfaces добавляем необходимые записи следующего вида:

```
allow-hotplug <int>
iface <int> inet static
address <ip>
netmask <mask>
gateway <ip>
```

Динамическая трансляция адресов

Создаем скрипт по адресу /etc/iptables.sh, где WAN - интерфейс в интернет

```
GNU nano 7.2 /etc/iptables.sh *
#!/bin/bash

export WAN="ens33"

iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X
iptables -t nat -X
iptables -t mangle -X

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -t nat -A POSTROUTING -o $WAN -j MASQUERADE

/sbin/iptables-save > /etc/iptables.rules
```

Далее прописываем `chmod +x /etc/iptables.sh`

Дописываем в /etc/network/interfaces строку `post-up iptables-restore < /etc/iptables.rules`

запускаем скрипт `/etc/iptables.sh`

Меняем в файле /etc/sysctl.conf значение на `net.ipv4.ip_forward = 1`

Применяем настройку командой `sysctl -p`

Настройка VLAN

Устанавливаем пакеты `vlan bridge-utils`

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

В /etc/network/interfaces добавляем интерфейсы для вланов

```
auto <vlanname>
iface <vlanname> inet static
address <ip>
netmask <mask>
vlan-raw-device <int>
```

Делаем это на всех машинах участвующих в vlan

Создание пользователя

```
useradd -u 1010 -m -s /bin/bash sshuser
echo "sshuser:P@ssw0rd" | chpasswd
```

Добавляем пользователя в группу sudo, чтобы он мог ее использовать

```
usermod -aG sudo sshuser
```

С помощью команды `visudo` добавляем строку для использования sudo без пароля.

Возможно понадобится прописать эту же строку но ниже `%sshuser ...`

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
sshuser ALL=(ALL) NOPASSWD:ALL
```

Так же можно создать файл по пути /etc/sudoers.d/ и добавить строку

```
<username> ALL=(ALL) NOPASSWD:ALL
```

Настройка ssh сервера

Установка: `apt-get install openssh-server`

В файле `issue.net` меняем текст на необходимый.

В файле `/etc/ssh/sshd_config` меняем строку `port` на `Port`

```

GNU nano 7.2 /etc/ssh/sshd_config *
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
Port 2024_
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

```

Изменяем эти строки для максимального кол-ва авторизации и авторизации только для определенных пользователей:

```

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
MaxAuthTries 2
#MaxSessions 10
AllowUsers sshuser
#PubkeyAuthentication yes

```

Изменяем эту строчку, добавив путь до файла /etc/issue.net:

```

# no default banner path
Banner /etc/issue.net

```

Конфигурация ip туннель

Создаем файл скрипта на 2ух машинах для поднятия gre туннеля touch /etc/gre.tun

Далее прописываем chmod +x /etc/gre.tun

В файле /etc/gre.tun прописываем следующие:

```

#!/bin/bash
ip tunnel del gre1
ip tunnel add gre1 mode gre remote (ip remote) local (ip local) ttl 255
ip addr add (ip tun local) peer (ip tun remote) dev gre1
ip link set gre1 up

```

P.S (ip remote = машина на которой должен быть 2ой туннель) (ip local = машина на который мы проводим настройку)

Добавляем наш gre туннель в конфигурационные файл (/etc/network/interfaces) на HQ-RTR и BR-RTR

post-up /etc/gre.tun

Обеспечение динамической маршрутизации

Качаем пакет apt-get install frr

Меняем строчку ospfd с "no" на "yes" в конфиг файле /etc/frr/daemons

Далее провдим настройку (на двух машинах) frr в файле /etc/frr/frr.conf

```

hostname "local hostname"
log syslog informational
no ipv6 forwarding
service integrated-vtysh-config
!
router ospf
  network (ip сеть нашего gre туннеля) area 0.0.0.0

```

```

network (ip сеть подключенного устройства) area 0.0.0.0
!
interface gre1
ip address (ip gre туннеля у данного локального устройства )
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 root

```

Далее проводим проверку работоспособность (на двух машинах) командой `vttysh -c "show ip ospf neighbor"`

Настройка протокола динамической конфигурации хостов

Качаем пакет `apt-get install isc-dhcp-server`

Далее заходим в файл `/etc/dhcp/dhcpd.conf` Прописываем:

```

subnet (сеть которая идет к клиенту, пример 192.168.1.0) netmask 255.255.255.240 {
    range (интервал ip адресов, пример 192.168.1.2 192.168.1.14;)
    option routers (сеть gateway, пример 192.168.1.1;)
    option domain-name-servers ( сеть dns, пример 192.168.0.2;
    option domain-name (имя домена "au-team.irpo");
}

```

```

host (имя устройства пример HQ-RTR) {
    fixed-address ip нашего устройства;
}

```

Далее меняем конфиг `/etc/default/isc-dhcp-server`

```

INTERFACESv4= ("интерфейс или же в нашем случае vlan пример vlan200")
INTERFACESv6=""

```

Далее настройка клиента `/etc/network/interfaces`

```

auto (название интерфейса или в нашем случае vlan пример, vlan200)
iface vlan200 inet dhcp
vlan-raw-device (название интерфейса, пример ens33)

```

Проверка работоспособности командой `ip a` и `cat /var/lib/dhcp/dhcpd.leases`

Настройка DNS

Установка пакета для разворачивания DNS - сервера: `apt-get install bind9`

Далее конфигурируем зоны в файле `/etc/bind/named.conf.local`, добавляя соответствующие записи:

Прямая зона (записи типа A, CNAME)

```

zone "domain" {
    type master;
    file "<zone file>";
};

```

Обраная зона (записи типа PTR)

```

zone "<ip пример: (100.168.192)>.in-addr.arpa" {
    type master;
    file "<zone file>";
}

```

zone file - файл содержащий конфигурацию зоны, создается обычно по пути `/etc/bind/`. Для простоты можно скопировать уже существующий файл и изменить его `cp /etc/bind/db.local /etc/bind/<zone filename>`

Конфигурация сервиса bind9: в файле `/etc/bind/named.conf.options`

```

GNU nano 7.2 /etc/bind/named.conf.options *
options {
    directory "/var/cache/bind";
    recursion yes;
    allow-recursion { any; };
    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        1.1.1.1;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { none; };
};

```

Конфигурация прямой зоны:

```

GNU nano 7.2 /etc/bind/db.au-team.irpo
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      hq-srv.au-team.irpo. root.au-team.irpo. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       hq-srv.au-team.irpo.

hq-rtr    IN      A        172.16.4.2
br-rtr    IN      A        172.16.5.2
hq-srv    IN      A        192.168.100.2
hq-cli    IN      A        192.168.200.3
br-srv    IN      A        192.168.2.2

moodle    IN      CNAME    hq-rtr.au-team.irpo.
wiki      IN      CNAME    hq-srv.au-team.irpo.

```

Конфигурация обратных зон:

```

GNU nano 7.2 /etc/bind/db.100.168.192
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       hq-srv.au-team.irpo.
1         IN      PTR      hq-rtr.au-team.irpo
2         IN      PTR      hq-srv.au-team.irpo

```

```

GNU nano 7.2 /etc/bind/db.200.168.192
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       hq-srv.au-team.irpo.
3         IN      PTR      hq-cli.au-team.irpo.

```

Проверка:

named-checkconf

named-checkzone <zone> <file>

Установка времени

timedatectl set-timezone Europe/Moscow

МОДУЛЬ 2

1. Настройте доменный контроллер Samba на машине BR-SRV.

- Создайте 5 пользователей для офиса HQ: имена пользователей формата user№.hq. Создайте группу hq, введите в эту группу созданных пользователей
- Введите в домен машину HQ-CLI
- Пользователи группы hq имеют право аутентифицироваться на клиентском ПК
- Пользователи группы hq должны иметь возможность повышать привилегии для выполнения ограниченного набора команд: cat, grep, id. Запускать другие команды с повышенными привилегиями пользователи группы не имеют права
- Выполните импорт пользователей из файла users.csv. Файл будет располагаться на виртуальной машине BR-SRV в папке /opt

2. Сконфигурируйте файловое хранилище:

- При помощи трёх дополнительных дисков, размером 1Гб каждый, на HQ-SRV сконфигурируйте дисковый массив уровня 5
- Имя устройства – md0, конфигурация массива размещается в файле /etc/mdadm.conf

- Обеспечьте автоматическое монтирование в папку /raid5 -Создайте раздел, отформатируйте раздел, в качестве файловой системы используйте ext4
 - Настройте сервер сетевой файловой системы(nfs), в качестве папки общего доступа выберите /raid5/nfs, доступ для чтения и записи для всей сети в сторону HQ-CLI
 - На HQ-CLI настройте автосмонтирование в папку /mnt/nfs
 - Основные параметры сервера отметьте в отчёте
- 3.Настройте службу сетевого времени на базе сервиса chrony
- В качестве сервера выступает HQ-RTR
 - На HQ-RTR настройте сервер chrony, выберите стратум 5
 - В качестве клиентов настройте HQ-SRV, HQ-CLI, BR-RTR, BR-SRV
- 4.Сконфигурируйте ansible на сервере BR-SRV
- Сформируйте файл инвентаря, в инвентарь должны входить HQ-SRV, HQ-CLI, HQ-RTR и BR-RTR
 - Рабочий каталог ansible должен располагаться в /etc/ansible
 - Все указанные машины должны без предупреждений и ошибок отвечать pong на команду ping в ansible посланную с BR-SRV
- 5.Развертывание приложений в Docker на сервере BR-SRV.
- Создайте в домашней директории пользователя файл wiki.yml для приложения MediaWiki.
 - Средствами docker compose должен создаваться стек контейнеров с приложением MediaWiki и базой данных.
 - Используйте два сервиса
 - Основной контейнер MediaWiki должен называться wiki и использовать образ mediawiki
 - Файл LocalSettings.php с корректными настройками должен находиться в домашней папке пользователя и автоматически монтироваться в образ.
 - Контейнер с базой данных должен называться mariadb и использовать образ mariadb.
 - Он должен создавать базу с названием mediawiki, доступную по стандартному порту, пользователя wiki с паролем WikiP@ssw0rd должен иметь права доступа к этой базе данных
 - MediaWiki должна быть доступна извне через порт 8080.
- 6.На маршрутизаторах сконфигурируйте статическую трансляцию портов
- Пробросьте порт 80 в порт 8080 на BR-SRV на маршрутизаторе BR-RTR, для обеспечения работы сервиса wiki 45
 - Пробросьте порт 2024 в порт 2024 на HQ-SRV на маршрутизаторе HQ-RTR
 - Пробросьте порт 2024 в порт 2024 на BR-SRV на маршрутизаторе BR-RTR
- 7.Запустите сервис moodle на сервере HQ-SRV:
- Используйте веб-сервер apache
 - В качестве системы управления базами данных используйте mariadb
 - Создайте базу данных moodledb
 - Создайте пользователя moodle с паролем P@ssw0rd и предоставьте ему права доступа к этой базе данных
 - У пользователя admin в системе обучения задайте пароль P@ssw0rd

- На главной странице должен отражаться номер рабочего места в виде арабской цифры, других подписей делать не надо
 - Основные параметры отметьте в отчёте
8. Настройте веб-сервер nginx как обратный прокси-сервер на HQ-RTR
- При обращении к HQ-RTR по доменному имени moodle.au-team.irpo клиента должно перенаправлять на HQ-SRV на стандартный порт, на сервис moodle
 - При обращении к HQ-RTR по доменному имени wiki.au-team.irpo клиента должно перенаправлять на BR-SRV на порт, на сервис mediawiki
9. Удобным способом установите приложение Яндекс Браузер для организаций на HQ-CLI
- Установку браузера отметьте в отчёте

РЕШЕНИЕ МОДУЛЬ 2

Настройте доменный контроллер Samba на машине BR-SRV

Конфигурация хранилища

Скачаем пакет apt install mdadm

Проверяем диск командой lsblk

Создаем массив mdadm --create /dev/<название массива> --level=<Версия RAID> --raid-devices=<Количество устройств для массива> /dev/<Диск 1> ... /dev/<Диск n>

Конфигурация массива размещаем в файле командой mdadm --detail --scan | tee -a /etc/mdadm.conf

Далее вводим команду sudo update-initramfs -u

Отформатировали раздел командой mkfs -t ext4 /dev/название массива

Создаем папку и монтируем раздел туда

mkdir /etc/(название папки, пример raid5)

mount /dev/(название массива) /(название папки)

Автомонтируем папку:

```
GNU nano 7.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=de163bed-8ffc-40e0-a9ae-28914d8c9ea4 / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=de917361-b5ee-4d6c-90e6-f55f053cd620 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/md0 /raid5 ext4 defaults 0 0_
```

Устанавливаем NFS apt-get install nfs-kernel-server -y

Создаем папку /(название корневой папки)/nfs

mkdir /(название корневой папки)/nfs

chmod -R 777 /(название корневой папки)/nfs/

Настраиваем доступ к папке в файле /etc/exports

/(название папки с массивом)/nfs (сеть к клиенту)(rw,sync,no_subnet_check)

Применяем конфигурацию и проверяем доступность.

exportfs -arv

```
systemctl restart nfs-kernel-server
```

```
exportfs -v
```

На клиенте:

Установка пакетов на клиенте

```
apt install nfs-common
```

Создаем папку для монтирования `mkdir -p /mnt/nfs`

Редактируем файл `/etc/fstab`, пишем в конце (сеть сервера): `/etc/(название корневой папки)/nfs /mnt/nfs nfs defaults 0 0`

Монтируем

```
sudo mount -a
```

```
systemctl daemon-reload
```

Проверяем на сервере HQ-SRV `echo "Hello, NFS!" | tee /(название корневой папки)/nfs/test.txt`

Проверяем на клиенте HQ-CLI `cat /mnt/nfs/test.txt`

Настройка службы сетевого времени на базе сервиса chrony

(На NTP сервере) Устанавливаем chrony `apt install chrony`

Переходим в конфиг `/etc/chrony/chrony.conf`

Коментим строчку `#pool 2.debian.pool.ntp.org iburst`

Далее ниже этой строки пишем:

```
local startum 5
```

```
allow (подсети необходимой машины, пример 192.168.100.0/26)
```

```
allow (подсети необходимой машины, пример 192.168.200.0/28)
```

```
allow (подсети необходимой машины, пример 172.16.5.0/28)
```

```
allow (подсети необходимой машины, пример 192.168.2.0/27)
```

После чего перезапускаем сеть `systemctl restart chrony`

Производим проверку командой `chronyc tracking`

Устанавливаем chrony на всех остальных устройствах `apt install chrony`

Далее на всех остальных устройствах (кроме NTP сервера) меняем конфиг

файл `/etc/chrony/chrony.conf` следующим образом:

```
#pool 2.debian.pool.ntp.org iburst
```

```
server (Название эталонной машины, пример HQ-RTR) iburst
```

После чего перезапускаем сеть на всех машинах `systemctl restart chrony`

Проводим проверку командой `chronyc tracking`, где параметр "Stratum" должен быть больше на 1 чем у NTP сервера

Далее проверяем подключение к NPT серверу командой `chronyc sources -v`, должно выводить IP адрес нашего NPT сервера с его значением "Stratum"

В случае не корректной работы с отображением клиентов или подсетей, на машине где есть эта проблема, в файле `/etc/iptables.sh` вводим строчку `iptables -t nat -A POSTROUTING -d (Сеть NTP сервера, пример 172.16.4.0/28) -j ACCEPT`

Далее на NTP сервере командой `chronyc clients` проверяем какие устройства к нему подключены

Конфигурирование ansible на сервере BR-SRV

Скачиваем пакет ansible `apt install ansible`

Создаем папку `sudo mkdir -p /etc/ansible`

Создаем нового пользователя на всех устройствах `adduser ansible`

Если мы на машине прописывали разрешенных пользователей, то туда надо добавить нашего нового пользователя `user ansible`

На всех устройствах из задания устанавливаем пакет sshd apt install openssh-server
Далее заходим файл /etc/ssh/sshd_config, раскомментируем строчку
"PubkeyAuthentication yes"

Далее на ansible сервере заходим в файл /etc/ansible/hosts И прописываем следующие:

```
[routers]
hq-rtr ansible_host=(ip адрес устройства, пример 172.16.4.2) ansible_ssh_port=(номер
порта sshd, пример 22) ansible_ssh_user=ansible
br-rtr ansible_host=(ip адрес устройства, пример 172.16.5.2) ansible_ssh_port=(номер
порта sshd, пример 22)ansible_ssh_user=ansible
```

```
[HQ]
hq-srv ansible_host=(номер порта sshd, пример 2024)ansible_ssh_user=ansible
hq-cli ansible_host=(номер порта sshd, пример 22)ansible_ssh_user=ansible
```

Генерируем ssh ключ на ansible сервере ssh-keygen

На сервере ansible прописываем команду ssh-copy-id -p (номер порта sshd, пример 22)
ansible@(ip адрес машины которую добавляем , пример 172.16.4.2) , эту команду воодим
для всех машин которые мы должны добавить

Проводим проверку работоспособности командой ansible all -m ping -i
/etc/ansible/hosts

Пример правильного выполнения задание , после проверки:

```
root@BR-SRV:~# ansible all -m ping -i /etc/ansible/hosts
br-rtr | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
~
hq-cli | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
~
hq-rtr | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
~
hq-srv | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
~
root@BR-SRV:~# _
```

Развертывание приложений в Docker на сервере BR-SRV

Скачиваем пакет apt install docker-compose

Настройки wiki.yml nano

wiki.yml

```
GNU nano 7.2 wiki.yml
version: "3.8"

services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: RootP@ssw0rd
      MYSQL_DATABASE: mediawiki
      MYSQL_USER: wiki
      MYSQL_PASSWORD: WikiP@ssw0rd
    volumes:
      - mariadb_data:/var/lib/mysql

  wiki:
    image: mediawiki:latest
    container_name: wiki
    restart: always
    ports:
      - "8080:80"
    environment:
      MEDIAWIKI_DB_HOST: mariadb
      MEDIAWIKI_DB_NAME: mediawiki
      MEDIAWIKI_DB_USER: wiki
      MEDIAWIKI_DB_PASSWORD: WikiP@ssw0rd
    # volumes:
    #   - /home/root/LocalSettings.php:/var/www/html/LocalSettings.php

volumes:
  mariadb_data:
```

Настроиваем через браузер

Подключиться через SSL

Идентификация этой вики

Имя базы данных (без дефисов):
[справка](#)
mediawiki

Префикс таблиц базы данных (без дефисов):
[справка](#)

Учётная запись для установки

Имя пользователя базы данных:
[справка](#)
wiki

Пароль базы данных:
[справка](#)

Назад Далее

Перенесем файл туда где установлен наш wiki.yml и изменим эту строку:

```
## The protocol and server name to use in fully-qualified URLs
$wgServer = "http://wiki.au-team.irpo";
```

Изменение файла wiki.yml

```

GNU nano 7.2                               wiki.yml *
version: "3.8"

services:
  mariadb:
    image: mariadb:latest
    container_name: mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: RootP@ssw0rd
      MYSQL_DATABASE: mediawiki
      MYSQL_USER: wiki
      MYSQL_PASSWORD: WikiP@ssw0rd
    volumes:
      - mariadb_data:/var/lib/mysql

  wiki:
    image: mediawiki:latest
    container_name: wiki
    restart: always
    ports:
      - "8080:80"
    environment:
      MEDIAWIKI_DB_HOST: mariadb
      MEDIAWIKI_DB_NAME: mediawiki
      MEDIAWIKI_DB_USER: wiki
      MEDIAWIKI_DB_PASSWORD: WikiP@ssw0rd
    volumes:
      - /home/root/LocalSettings.php:/var/www/html/LocalSettings.php

volumes:
  mariadb_data:

```

Далее перезапускаем контейнеры

```
docker-compose -f wiki.yml down
```

```
docker-compose -f wiki.yml up -d
```

На маршрутизаторах сконфигурируйте статическую трансляцию портов

Проброс портов на BR-RTR, заходим и редактируем

файл /etc/iptables.sh

```

GNU nano 7.2                               /etc/iptables.sh *
#!/bin/bash

export WAN="ens33"

iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X
iptables -t nat -X
iptables -t mangle -X

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -t nat -A POSTROUTING -o $WAN -j MASQUERADE

iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.2.2:8080
iptables -t nat -A POSTROUTING -p tcp --dport 8080 -d 192.168.2.2 -j SNAT --to-source 172.16.5.2

iptables -t nat -A PREROUTING -p tcp --dport 2024 -j DNAT --to-destination 192.168.2.2:2024
iptables -t nat -A POSTROUTING -p tcp --dport 2024 -d 192.168.2.2 -j SNAT --to-source 172.16.5.2_

/sbin/iptables-save > /etc/iptables.rules

```

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination <ip>:8080
```

```
iptables -t nat -A POSTROUTING -p tcp --dport 8080 -d <ip> -j SNAT --to-source <ip-rtr>
```

```
iptables -t nat -A PREROUTING -p tcp --dport 2024 -j DNAT --to-destination <ip>:2024
```

```
iptables -t nat -A POSTROUTING -p tcp --dport 2024 -d <ip> -j SNAT --to-source <ip-rtr>
```


Проверяем работоспособность на клиенте, командой `ssh -p 2024 sshuser@(ip адресс BR-RTR)`

Проброс портов на HQ-RTR, заходим и редактируем файл `/etc/iptables.sh`

```
GNU nano 7.2 /etc/iptables.sh *
#!/bin/bash

export WAN="ens33"

iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X
iptables -t nat -X
iptables -t mangle -X

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -t nat -A POSTROUTING -o $WAN -j MASQUERADE

iptables -t nat -A PREROUTING -p tcp --dport 2024 -j DNAT --to-destination 192.168.0.2:2024
iptables -t nat -A POSTROUTING -p tcp --dport 2024 -d 192.168.0.2 -j SNAT --to-source 172.16.4.2

/sbin/iptables-save > /etc/iptables.rules
```

Проверяем работоспособность на клиенте, командой `ssh -p 2024 sshuser@(ip адресс HQ-RTR)`

Запуск сервис moodle на сервере HQ-SRV:

Установим все необходимые пакеты `apt-get install apache2 mariadb-server php php-mysql libapache2-mod-php php-xml php-mbstring php-zip php-curl php-gd php-intl unzip`

Создаем бд для

moodle

```
MariaDB [(none)]> CREATE DATABASE moodledb;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'moodle'@'localhost' IDENTIFIED BY 'P@ssw0rd';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON moodledb.* TO 'moodle'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> _
```

Скачиваем

moodle

```
root@HQ-SRV:/var/www/html# wget https://download.moodle.org/download.php/direct/stable405/moodle-latest-405.zip
--2024-11-28 00:20:41-- https://download.moodle.org/download.php/direct/stable405/moodle-latest-405.zip
Распознаётся download.moodle.org (download.moodle.org)... 188.114.96.233, 188.114.99.233, 2a06:98c1:3123:e000::9, ...
Подключение к download.moodle.org (download.moodle.org)[188.114.96.233]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 302 Found
Адрес: https://packaging.moodle.org/stable405/moodle-latest-405.zip [непеход]
--2024-11-28 00:20:42-- https://packaging.moodle.org/stable405/moodle-latest-405.zip
Распознаётся packaging.moodle.org (packaging.moodle.org)... 188.114.99.233, 188.114.96.233, 2a06:98c1:3123:e000::9, ...
Подключение к packaging.moodle.org (packaging.moodle.org)[188.114.99.233]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 97512579 (93M) [application/zip]
Сохранение в: «moodle-latest-405.zip»

moodle-latest-405.zip 100%[=====] 93,00M 10,4MB/s за 9,4с

2024-11-28 00:20:52 (9,87 MB/s) - «moodle-latest-405.zip» сохранён [97512579/97512579]

root@HQ-SRV:/var/www/html# file moodle-latest-405.zip
moodle-latest-405.zip: Zip archive data, at least v1.0 to extract, compression method=store
root@HQ-SRV:/var/www/html#
```

Распаковываем файл командой `unzip <filename>` Желательно распаковывать сразу в `/var/www/html`

Меняем права на папке moodle

```
chown -R www-data:www-data /var/www/html/moodle
```

Создаем новый конфигурационный файл для апачи

```
touch /etc/apache2/sites-available/moodle.conf
```

```
cp /etc/apache2/sites-available/000-default.conf
```

```
/etc/apache2/sites-available/moodle.conf
```

```
GNU nano 7.2 /etc/apache2/sites-available/moodle.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    #ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ServerName moodle.au-team.irpo
    RedirectMatch 301 ^/$ http://moodle.au-team.irpo/moodle
    <Directory /var/www/html/moodle>
        Options -Indexes
        AllowOverride All
        Require all granted
    </Directory>
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Изменим эту настройку по пути `/etc/apache2/apache2.conf`

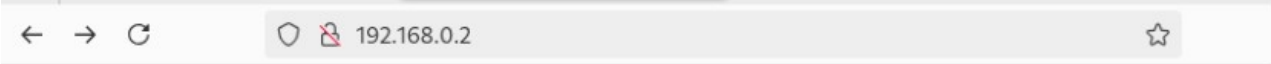
```
GNU nano 7.2 /etc/apache2/apache2.conf
</Directory>

<Directory /var/www/>
    Options -Indexes +FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```



Активируем сайт на HQ-SRV `a2ensite moodle.conf`

```
systemctl reload apache2
```


Переходим на сайт и начинаем конфигурацию



Index of /

Name	Last modified	Size	Description
 moodle-latest-405.zip	2024-11-22 17:26	93M	
 moodle/	2024-11-22 17:26	-	

Apache/2.4.62 (Debian) Server at 192.168.0.2 Port 80

Далее на HQ-SRV вводим команды
mkdir /var/moodldata
chown -R www-data /var/www/moodldata
chmod -R 755 /var/www/moodldata

Веб-адрес	<input type="text" value="http://192.168.0.2/moodle"/>
Каталог Moodle	<input type="text" value="/var/www/html/moodle"/>
Каталог данных	<input type="text" value="/var/moodldata"/>

Название базы данных

Выберите драйвер базы данных

Moodle поддерживает несколько типов серверов баз данных. Свяжитесь с администратором сервера, если не знаете, какой именно тип выбрать.

Тип	<div>MariaDB («родной»)/mariadb</div>
-----	---------------------------------------

« Назад

Далее »

Сервер баз данных	<input type="text" value="localhost"/>
Название базы данных	<input type="text" value="moodledb"/>
Пользователь базы данных	<input type="text" value="moodle"/>
Пароль	<input type="password" value="P@ssw0rd"/>
Префикс имен таблиц	<input type="text" value="mdl_"/>
Порт базы данных	<input type="text"/>
Подключение через Unix-сокеты	<input type="text"/>

Доустанавливаем все необходимые пакеты и поставим разрешения `apt-get install php-intl`

Так же изменим значение в файле `/etc/php/8.2/apache2/php.ini`
`max_input_vars = 5000`

Настройка веб-сервера `nginx` как обратный прокси-сервер

Настройка для мудла:

```

server {
    listen 80;
    #listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    #index index.html index.htm index.nginx-debian.html;

    server_name moodle.au-team.irpo;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        proxy_pass http://192.168.100.2:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # pass PHP scripts to FastCGI server

```

Настройка для вики:

```

GNU nano 7.2 /etc/nginx/sites-available/wiki
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##
# Default server configuration
#
server {
    listen 80;
    #listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    #index index.html index.htm index.nginx-debian.html;

    server_name wiki.au-team.irpo;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        proxy_pass http://172.16.5.2:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ /\.php$ {
    #    include snippets/fastcgi-php.conf;

```

Проделать с мулл и вики:

```
root@HQ-RTR:~# ln -s /etc/nginx/sites-available/moodle /etc/nginx/sites-enabled/
```

МОДУЛЬ 3

2.Выполните настройку центра сертификации на базе HQ-SRV:

- Необходимо использовать отечественные алгоритмы шифрования
- Сертификаты выдаются на 365 дней
- Обеспечьте доверие сертификату для HQ-CLI
- Выдайте сертификаты веб серверам
- Перенастройте ранее настроенные веб сервера, moodle, wiki, реверсивный прокси nginx на протокол https
- При обращении к веб серверам по их доменным именам у браузера клиента не должно возникать предупреждений

3.Перенастройте ip-туннель с базового до уровня туннеля, обеспечивающего шифрование трафика:

- Настройте защищенный туннель между HQ-RTR и BR-RTR
 - Внесите необходимые изменения в конфигурацию динамической маршрутизации, протокол динамической маршрутизации должен озобновить работу после перенастройки туннеля
 - Выбранное программное обеспечение, обоснование его выбора и его основные параметры, изменения в конфигурации динамической маршрутизации отметьте в отчёте
- 4.Настройте межсетевой экран на маршрутизаторах HQ-RTR и BR-RTR на сеть в сторону ISP:
- Обеспечьте работу протоколов http, https, dns, ntp, icmp или дополнительных нужных протоколов
 - Запретите остальные подключения из сети Интернет во внутреннюю сеть
- 5.Настройте принт-сервер cups на сервере HQ-SRV
- Опубликуйте виртуальный pdf-принтер
 - На клиенте HQ-CLI подключите виртуальный принтер как принтер по умолчанию
- 6.Реализуйте логирование при помощи rsyslog на устройствах HQ-RTR, BR-RTR, BR-SRV
- Сервер сбора логов расположен на HQ-SRV, убедитесь, что сервер не является клиентом самому себе
 - Приоритет сообщений должен быть не ниже warning
 - Все журналы должны находиться в директории /opt. Для каждого устройства должна выделяться своя поддиректория, которая совпадает с именем машины
 - Реализуйте ротацию логов:
 - Ротация производится один раз в неделю
 - Логи необходимо сжимать
 - Минимальный размер логов для ротации – 10 МБ
- 7.На сервере HQ-SRV реализуйте мониторинг устройств с помощью открытого программного обеспечения. Обеспечьте доступность по URL - <https://mon.austeam.irpo>
- Мониторить нужно устройства HQ-RTR, HQ-SRV, BR-RTR и BR-SRV
 - В мониторинге должны визуальнo отображаться нагрузка на ЦП, объем занятой ОП и основного накопителя
 - Логин и пароль для службы мониторинга admin P@ssw0rd
 - Выбор программного обеспечения, основание выбора и основные параметры с указанием порта, на котором работает мониторинг, отметьте в отчёте
- 8.Реализуйте механизм инвентаризации машин HQ-SRV и HQ-CLI через Ansible на BR-SRV:
- Плейбук должен собирать информацию о рабочих местах:
 - Имя компьютера
 - IP-адрес компьютера
 - Отчеты, собранные с машин, должны быть размещены в том же каталоге на сервере, где и плейбук, в папке PC_INFO, в формате .yaml. Файл называется именем компьютера, который был инвентаризован
 - Рабочий каталог ansible должен располагаться в /etc/ansible

9.Реализуйте механизм резервного копирования конфигурации для машин HQ-RTR и BR-RTR, через Ansible на BR-SRV:

- Плейбук должен собирать информацию о сетевых устройствах HQRTR и BR-RTR и делать резервную копию конфигурации (в случае использования EcoRouter – полную конфигурацию, в случае ОС на базе Linux – файлы конфигурации динамической маршрутизации, настроек межсетевого экрана, параметров настройки сети, настройки динамической конфигурации хостов). Информацию сохранять в папку NETWORK_INFO

Решение Модуль 3

Настройка центра сертификации

Установка OpenSSL с поддержкой ГОСТ:

```
apt update && apt install -y openssl libengine-gost-openssl
```

В файле /etc/ssl/openssl.cnf добавим:

```
# В начале файла
```

```
openssl_conf=openssl_def
```

```
# В конец файла
```

```
[openssl_def]
```

```
engines = engine_section
```

```
[engine_section]
```

```
gost = gost_section
```

```
[gost_section]
```

```
engine_id = gost
```

```
default_algorithms = ALL
```

```
CRYPTO_PARAMS = id-Gost28147-89-CryptoPro-A-ParamSet
```

Проверим появился ли Gost engine командой `openssl engine`

Вывод команды:

```
root@HQ-SRV:~# openssl engine
(dynamic) Dynamic engine loading support
(gost) Reference implementation of GOST engine
root@HQ-SRV:~#
```

Проверим командой `openssl ciphers|tr ':' '\n'|grep GOST` появились ли отечественные методы шифрования. Вывод команды:

```
root@HQ-SRV:~# openssl ciphers|tr ':' '\n'| grep GOST
GOST2012-MAGMA-MAGMAOMAC
GOST2012-KUZNYECHIK-KUZNYECHIKOMAC
LEGACY-GOST2012-GOST8912-GOST8912
IANA-GOST2012-GOST8912-GOST8912
GOST2001-GOST89-GOST89
root@HQ-SRV:~#
```

Создадим папку, в которой будем хранить сертификаты и ключи:

```
mkdir -p /etc/pki/CA/{certs,crl,newcert,private}
```

```
chmod 700 /etc/pki/CA/private
```

Сгенерируем ключ для корневого сертификата с использованием отечественного алгоритма шифрования:

```
openssl genpkey -algorithm gost2012_256 -pkeyopt paramset:A -out
/etc/pki/CA/private/ca.key
```

Выпустим и подпишем корневой сертификат:

```
openssl req -x509 -new -key /etc/pki/CA/private/ca.key -days 365 -out
/etc/pki/CA/certs/ca.crt
```

Проверить сертификат можно командой `openssl x509 -noout -text -in <cert> -certopt no_version,no_pubkey,no_sigdump`

Далее выпущенный сертификат переместим на клиенте по этому
пути /usr/local/share/ca-certificates/<название сертификата>

Применим командой update-ca-certificates

Создадим конфигурационный файл для создания сертификатов для сервисов.

Пример:

```
[ req ]
```

```
req_extensions = v3_req
```

```
default_bits = 256
```

```
prompt = no
```

```
default_md = gost2012_256
```

```
distinguished_name = dn
```

```
[ dn ]
```

```
C= RUSSIA
```

```
ST= TATARSTAN
```

```
L= KAZAN
```

```
O= DEMO
```

```
OU= DEMO
```

```
emailAddress= test@example.com
```

```
CN= moodle.au-team.irpo
```

```
[ v3_req ]
```

```
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
```

```
extendedKeyUsage = serverAuth
```

```
basicConstraints = CA:FALSE
```

```
subjectAltName = @alt_names
```

```
[ alt_names ]
```

```
DNS.1 = moodle.au-team.irpo
```

```
IP.1 = 192.168.100.2
```

```
IP.2 = 172.16.4.2
```