МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ»
Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

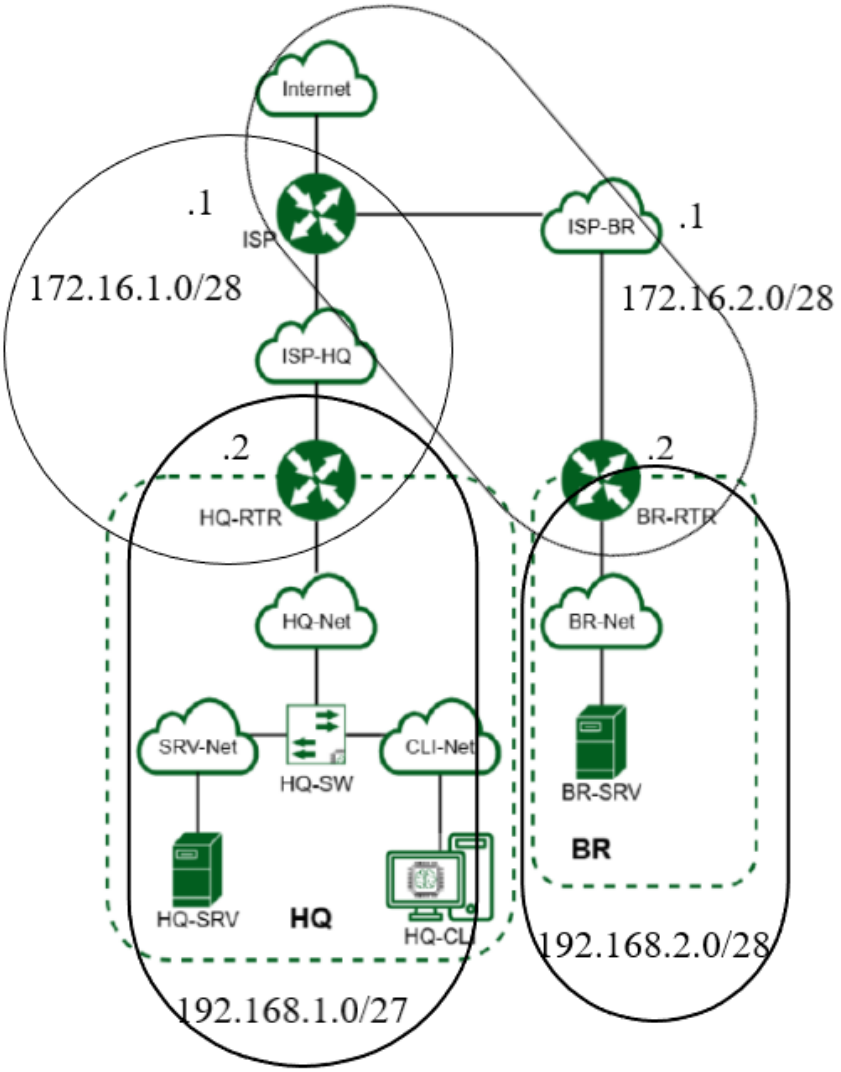
ЛАБОРАТОРНЫЕ РАБОТЫ

по дисциплине

Организация администрирования компьютерных систем

Работу выполнили
Студенты гр.4436
Александров А.П.
Принял
Преподаватель Муртазин Р.М.

Казань 2025



1

Топология сети

Устройство	Адаптер	Назначение	Название	IP	Gateway	VMnet
ISP	ens33	интернет	NAT	-	-	
	ens34	172.16.1.0	ISP-HQ	172.16.1.1/28	-	VMnet3
	ens35	172.16.2.0	ISP-BR	172.16.2.1/28	-	VMnet4
HQ-RTR	ens34	172.16.1.0	IPS-HQ	172.16.1.2/28	172.16.1.1	VMnet3
	ens35.100	192.168.100.0	SRV-NET	192.168.100.1/27	-	VMnet7
	ens36.200	192.168.200.0	CLI-NET	192.168.200.1/28	-	VMnet9
	Ens34.999	172.16.99.0	MGMT	172.16.99.1/30		
BR-RTR	ens34	172.16.2.0	ISP-BR	172.16.2.2/28	172.16.2.1	VMnet4
	ens35	192.168.2.0	BR-NET	192.168.2.1/28	-	VMnet6

Интерфейсы устройств

(название интерфейсов могут изменяться)

ISP:

```
auto ens34
iface ens34 inet static
address 172.16.1.1/28

auto ens35
iface ens35 inet static
address 172.16.2.1/28
```

HQ-RTR:

```
auto ens34
iface ens34 inet static
address 172.16.1.2/28
gateway 172.16.1.1

auto ens35.100
iface ens35.100 inet static
address 192.168.100.1/27

auto ens36.200
iface ens36.200 inet static
address 192.168.200.1/28

auto ens34.999
iface ens34.999 inet static
address 172.16.99.1/30
```

BR-RTR:

```
auto ens34
iface ens34 inet static
address 172.16.2.2/28
gateway 172.16.2.1

auto ens35
iface ens35 inet static
address 192.168.2.1/28
```

Настройка NAT

ISP:

Сначала устанавливаем iptables: apt install iptables-persistent

И пишем команды:

/usr/sbin/iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE

здесь указывается название интерфейса, идущего в сторону интернета (isp – ens33, hq-rtr – ens34, br-rtr – ens34)

/usr/sbin/iptables-save > /etc/iptables/rules.v4

Теперь заходим в файл nano /etc/sysctl.conf, находим строку net.ipv4.ip_forward=1, убираем решетку, сохраняем, выходим и перезагружаем командой /usr/sbin/sysctl -p

На соседних машинах заходим в файл nano /etc/resolv.conf и там пишем: nameserver 8.8.8.8

HQ-RTR, BR-RTR:

Проделываем всё тоже самое, но пишем интерфейс, выходящий к ISP.

Создание локальных учетных записей на серверах (HQ-SRV, BR-SRV) и роутерах (HQ-RTR, BR-RTR) (1.3)

HQ-SRV, BR-SRV:

Сначала добавляем пользователя remote_user без пароля и идентификатора:

/usr/sbin/useradd remote_user

Добавляем пользователя с идентификатором этой командой:

useradd -m -s /bin/bash -u 2026 sshuser

Задаем пароль пользователя командой:

passwd sshuser

Далее потребуется ввести пароль два раза.

Чтобы запускать sudo без пароля надо перейти в файл /etc/sudoers и там ввести:

sshuser ALL=(ALL) NOPASSWD:ALL

HQ-RTR, BR-RTR:

Добавляем пользователя командой:

useradd -m -s /bin/bash net_admin

И пароль:

passwd net_admin

И вводим пароль (P@ssw0rd) два раза

Чтобы запускать sudo без пароля надо перейти в файл /etc/sudoers и там ввести:

net_admin ALL=(ALL) NOPASSWD:ALL

Настройка SSH на HQ-SRV и BR-SRV

Устанавливаем ssh командой:

apt install openssh-server

Заходим в файл /etc/ssh/sshd_config, раскомментируем строку #port 22 и меняем порт на нужный (2026).

В этом же файле находим строку #permitrootlogin, раскомментируем ее и пишем значение no, чтобы отключить вход суперпользователю. Далее добавляем строку AllowUsers sshuser, чтобы разрешить вход только пользователю sshuser.

Опять в этом же файле находим строку #MaxAuthTries, раскомментируем и пишем значение 2, чтобы ограничить вход по ssh только до двух раз.

Теперь в этом же файле добавляем строку Banner /etc/issue.net. Заходим в файл /etc/issue.net и пишем Authorized access only.

Туннель GRE на HQ-RTR и BR-RTR

Заходим в файл /etc/modules и пишем ip_gre, перезагружаем вм.

Далее заходим в файл /etc/network/interfaces и пишем:

Для HR-RTR:

auto gre30

iface gre30 inet tunnel

address 10.10.10.1

netmask 255.255.255.252

mode gre

local 172.16.1.2

endpoint 172.16.2.2

ttl 225

post-up ip route add 192.168.2.0/28 via 10.10.10.2

Для BR-RTR:

auto gre30

iface gre30 inet tunnel

address 10.10.10.2

netmask 255.255.255.252

mode gre

local 172.16.2.2

endpoint 172.16.1.2

ttl 225

post-up ip route add 192.168.100.0/27 via 10.10.10.1

post-up ip route add 192.168.200.0/28 via 10.10.10.1

Потом перезагружаем сеть командой: *systemctl restart networking.*

И изменяем брандмауэр командой:

iptables -A INPUT -p gre -j ACCEPT

Настройка динамической маршрутизации на HQ-RTR и BR-RTR

Устанавливаем frr командой: *apt install frr.*

Заходим в файл /etc/frr/daemons и у строки ospfd ставим значение yes вместо no.

Потом пишем: *systemctl enable frr* и перезагружаем машину.

Заходим в оболочку командой vtysh и пишем:

Для HQ-RTR:

conf t

ip forwarding

router ospf

network 192.168.1.0/27 area 0

network 10.10.10.0/30 area 0

exit

exit

wr

exit

Для BR-RTR:

conf t

ip forwarding

router ospf

network 192.168.2.0/28 area 0

network 10.10.10.0/30 area 0

exit

exit

wr

exit

Чтобы установить пароль надо прописать на обоих устройствах здесь же:

int gre30

ip ospf authentication message-digest

ip ospf message-digest-key 1 md5 1c+rYtGm

DHCP на HQ-RTR

Устанавливаем DHCP командой:

apt install isc-dhcp-server

Заходим в файл /etc/default/isc-dhcp-server, там находим строки #DHCPDv4_CONF=/etc/dhcp/dhcpd.conf и #DHCPDv4_PID=/var/run/dhcpd.pid и раскомментируем их. Потом находим строку INTERFACESv4="" и в кавычках пишем название интерфейса, который будет раздавать ip-адреса. Потом заходим в файл /etc/dhcp/dhcpd.conf и редактируем его: в строке option domain-name в кавычках пишем au-team.irpo; в строке option domain-name-servers пишем ip-адрес dns-сервера (в нашем случае это HQ-SRV (192.168.100.1)). И дальше пишем так:

```
subnet 192.168.200.0 netmask 255.255.255.224 {
  range 192.168.200.2 192.168.200.18;
  option routers 192.168.200.1;
  option domain-name-servers 192.168.100.2;
}
```

Далее запускаем dhcp командой:

systemctl enable isc-dhcp-server

DNS на HQ-SRV

Для настройки DNS на HQ-SRV необходимо сначала установить bind9:

apt install bind9

Далее заходим в файл /etc/bind/named.conf.options и сделать всё, как на скрине:

```
forwarders {
        8.8.8.8;
};

recursion yes;
allow-query { any; };
listen-on { any; };
allow-recursion { any; };
```

Далее заходим в файл /etc/bind/named.conf.local и пишем зоны:

```
zone "au-team.irpo" IN {
        type master;
        file "/etc/bind/zones/db.au-team.irpo";
};

zone "100.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/zones/db.100.168.192";
};

zone "200.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/zones/db.200.168.192";
};

zone "1.16.172.in-addr.arpa" IN {
        type master;
        file "/etc/bind/zones/db.1.16.172";
};

zone "2.16.172.in-addr.arpa" IN {
        type master;
        file "/etc/bind/zones/db.2.16.172";
};

zone "2.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/zones/db.2.168.192";
};
```

zone “7.16.172.in-addr.arpa” IN {

type master;

file “/etc/bind/zones/db.7.16.172”

}

Дальше создаем директорию и создаем файлы этих зон:

mkdir /etc/bind/zones

cd /etc/bind

cp db.local /etc/bind/zones/db.au-team.irpo

cp db.127 /etc/bind/zones/db.100.168.192

cp db.127 /etc/bind/zones/db.200.168.192

cp db.127 /etc/bind/zones/db.1.16.172

cp db.127 /etc/bind/zones/db.2.16.172

cp db.127 /etc/bind/zones/db.2.168.192

Потом заходим в каждую из этих зон и пишем:

```
 GNU nano 7.2              /etc/bind/zones/db.au-team.irpo
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     au-tea,.irpo. root.au-team.irpo. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      au-team.irpo.
@       IN      A       192.168.100.2

hq-rtr.au-team.irpo.    IN      A       172.16.1.2
br-rtr.au-team.irpo.    IN      A       172.16.2.2
hq-srv.au-team.irpo.    IN      A       192.168.100.2
br-srv.au-team.irpo.    IN      A       192.168.2.2
docker.au-team.irpo.    IN      A       172.16.1.1
web.au-team.irpo.       IN      A       172.16.2.1
```

```
 GNU nano 7.2              /etc/bind/zones/db.100.168.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     au-team.irpo. root.au-team.irpo. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      100.168.192.in-addr.arpa.
@       IN      A       192.168.100.2.

10      IN      PTR     au-team.irpo.
11      IN      PTR     hq-srv.au-team.irpo.
```

```
 GNU nano 7.2                /etc/bind/zones/db.1.16.172
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     au-team.irpo. root.au-team.irpo. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      1.16.172.au-team.irpo.
@       IN      A       192.168.100.2

20      IN      PTR     hq-rtr.au-team.irpo.
21      IN      PTR     docekr.au-team.irpo.
```

```
 GNU nano 7.2                /etc/bind/zones/db.2.16.172
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     au-team.irpo. root.au-team.irpo. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      2.16.172.in-addr.arpa.
@       IN      A       192.168.100.2

30      IN      PTR     br-rtr.au-team.irpo.
31      IN      PTR     web.au-team.irpo.
```

```
 GNU nano 7.2              /etc/bind/zones/db.2.168.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     au-team.irpo. root.au-team.irpo. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      2.168.192.in-addr.arpa.
@       IN      A       192.168.100.2

40      IN      PTR     br-srv.au-team.irpo.
```

Модуль 2

Samba

BR-SRV:

export DEBIAN_FRONTEND=noninteractive

apt-get install samba winbind libnss-winbind krb5-user smbclient ldb-tools python3-cryptography

unset DEBIAN_FRONTEND

Далее заходим в файл /etc/krb5.conf и там пишем в самом начале (где [libdefaults]):

default_realm = AU-TEAM.IRPO

dns_lookup_kdc = true

dns_lookup_realm = false

Потом листаем дальше и у [realms] пишем (с табом):

AU-TEAM.IRPO = {

	kdc = br-srv.au-team.irpo

	admin_server = br-srv.au-team.irpo

}

И в самом конце у [domain_realm] пишем:

.au-team.irpo = AU-TEAM.IRPO

au-team.irpo = AU-TEAM.IRPO

Удаляем файл конфигурации самбы:

rm -f /etc/samba/smb.conf

Останавливаем все службы файлового сервера:

systemctl stop samba winbind smbd nmbd

Начинаем инициализация домена командой *samba-tool domain provision* и там надо будет по очереди вводить:

AU-TEAM.IRPO

Нажать enter

Нажать enter

Нажать enter

8.8.8.8

И вводим пароль администратора

Потом удаляем файл:

rm -f /var/lib/samba/private/krb5.conf

И заменим его ссылкой

ln -s /etc/krb5.conf /var/lib/samba/private/krb5.conf

Активируем samba:

systemctl disable samba winbind nmbd smbd

systemctl mask samba winbind nmbd smbd

systemctl unmask samba-ad-dc

systemctl enable samba-ad-dc

systemctl restart samba-ad-dc

И для проверки пишем:

kinit Administrator

klist

Надо, чтобы вывод был такой, иначе придется что-то чинить:

(не забываем заполнить файл /etc/resolv.conf)

```
root@BR-SRV:/home/br-srv# kinit administrator
Password for administrator@AU-TEAM.IRPO:
Warning: Your password will expire in 41 days on Вт 23 дек 2025 12:34:23
root@BR-SRV:/home/br-srv# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@AU-TEAM.IRPO

Valid starting       Expires              Service principal
11.11.2025 18:47:41  12.11.2025 04:47:41  krbtgt/AU-TEAM.IRPO@AU-TEAM.IRPO
        renew until 12.11.2025 18:47:37
```

Добавление на HQ-CLI (?) пользователей:

for i in {1..5}; do samba-tool user create hquser$i 'P@ssw0rd'; done
samba-tool group add hq
samba-tool group addmembers hq hquser1,hquser2,hquser3,hquser4,hquser5
Заходим в /etc/sudoers и у групп добавляем строку:
%hq ALL=(ALL:ALL) NOPASSWD: /usr/bin/cat, /usr/bin/grep, /usr/bin/id
Перед этим добавляем зоны на HQ-SRV (см. ниже)

Теперь вводим HQ-CLI в домен:

apt install -y realmd sssd-tools sssd libnss-sss libpam-sss adcli samba-common-bin oddjob oddjob-mkhomedir packagekit -y
realm discover au-team.irpo –verbose
realm join au-team.irpo

HQ-SRV:

В файл/etc/bind/zones/db.au-team.irpo добавляем строки:

_ldap._tcp.au-team.irpo. IN SRV 0 100 389 br-srv.au-team.irpo.

_kerberos._tcp.au-team.irpo. IN SRV 0 100 88 br-srv.au-team.irpo.

_kerberos._udp.au-team.irpo. IN SRV 0 100 88 br-srv.au-team.irpo.

_kpasswd._tcp.au-team.irpo. IN SRV 0 100 464 br-srv.au-team.irpo.

_kpasswd._udp.au-team.irpo. IN SRV 0 100 464 br-srv.au-team.irpo.

_ldap._tcp.dc._msdcs.au-team.irpo IN SRV 0 100 389 br-srv.au-team.irpo.

Далее переходим в файл /etc/bind/named.conf.local, добавляем к каждой зоне пару строк:

allow-transfer { none; };

allow-update { none; };

Потом внизу файла добавляем:

dlz "samba-dlz"

{

//database "dlopen /usr/lib/x86_64-linux-gnu/samba/bind9/dlz_bind9_11.so";

};

Переходим в файл /etc/bind/named.conf.options и добавляем строку в самый низ (до последней фигурной скобки):

allow-update {192.168.2.2;}; <здесь надо писать адрес BR-SRV>

Перезагружаем bind9:

apt install bind9

HQ-CLI:

Устанавливаем это на клиента командой:

apt-get install realmd sssd oddjob oddjob-mkhomedir adcli samba-common packagekit sssd-tools krb5-user

При установке откроется окно и там надо будет по очереди написать:

AU-TEAM.IRPO

br-srv.au-team.irpo

br-srv.au-team.irpo

Еще раз проверяем:

kinit Administrator

klist

Подключаем машину к samba:

/usr/sbin/realm join --user=user1 AU-TEAM.IRPO –v

Вводим пароль от пользователя, созданного выше, и машина подключена

Теперь пишем: *groups hquser1@au-team.irpo*. В конце нам должно выдать число. Копируем его. Заходим в /etc/sudoers и ближе к концу пишем:

%#1897400513 ALL=(ALL) NOPASSWD: /bin/cat, /bin/grep, /usr/bin/id <это число надо заменить на свое>

RAID

HQ-SRV:

Смотрим название добавленных дисков командой: *lsblk.* диск, возможно, будет называться sdb, sdc.

Устанавливаем mdadm:

apt install mdadm

Создаем raid0:

/usr/sbin/mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc

Сохраняем конфигурацию массива:

/usr/sbin/mdadm --detail --scan >> /etc/mdadm/mdadm.conf

/usr/sbin/update-initramfs -u

Создаём раздел на RAID-устройстве:

fdisk /dev/md0

Здесь надо будет по очереди писать:

n

нажать enter

нажать enter

нажать enter

нажать enter

w

Форматируем раздел в ext4:

/usr/sbin/mkfs.ext4 /dev/md0p1

Создаем папку и примонтируем туда раздел:

mkdir raid

mount /dev/md0p1 /raid

Для автоматического монтирования переходим в файл /etc/fstab и добавляем строку:

/dev/md0p1 /raid ext4 defaults 0 0

Монтируем:

mount -a

NFS

HQ-SRV:

Устанавливаем nfs-сервер:

apt install nfs-server

Создаем папку для общего пользования и назначаем ей права доступа:

mkdir /raid/nfs

chmod 766 /raid/nfs

Заходим в /etc/exports и добавляем туда строку:

/raid/nfs 192.168.200.0/28 (rw, no_root_squash) <здесь маска и подсеть HQ-CLI>

Экспортируем файловую систему и запускаем сервер:

/usr/sbin/exportfs -arv

systemctl enable nfs-server

systemctl restart nfs-server

HQ-CLI:

Устанавливаем nfs для клиента:

apt install nfs-common

Создаем папку для общего пользования и назначаем ей права доступа:

mkdir /mnt/nfs

chmod 777 /mnt/nfs

Заходим в /etc/fstab и там пишем:

192.168.100.2:/raid/nfs /mnt/nfs nfs defaults 0 0 <здесь указан адрес HQ-SRV>

Монтируем:

mount –a

Проверка:
df -h /mnt/nfs

Chrony

ISP:

Устанавливаем chrony командой:

apt install chrony

Переходим в файл /etc/chrony/chrony.conf, под строкой *# Use Debian vendor zone.* стираем или закомментируем строку pool 2.debian… и пишем:

Server 127.0.0.1 iburst prefer

local stratum 5

allow 0/0

Перезагружаем и добавляем автозапуск chrony:

systemctl restart chrony

systemctl enable chrony

На клиентах ntp (HQ-SRV, HQ-CLI, BR-RTR, BR-SRV):

Устанавливаем chrony командой:

apt install chrony

Переходим в файл /etc/chrony/chrony.conf, под строкой # *Use Debian vendor zone.* стираем или закомментируем строку pool 2.debian… и пишем:

server 172.16.1.1 iburst <это адрес для ветки HQ, для ветки BR здесь пишем 172.16.2.1>

Перезагружаем и добавляем автозапуск chrony:

systemctl restart chrony

systemctl enable chrony

Ansible

Здесь будут необходимы новые пользователи, поэтому придется их создать (HQ-RTR, BR-RTR, HQ-CLI):

useradd -m -s /bin/bash -u 2026 sshuser

passwd sshuser

Заходим в /etc/sudoers и там пишем в конце у пользователя root:

sshuser ALL=(ALL) NOPASSWD:ALL

Устанавливаем ssh на все машины кроме isp:

apt install openssh-server

Заходим в /etc/ssh/sshd_config, там меняем порт на 2026 (по желанию, я сделал) и находим строку #PermitRootLogin, раскомментируем ее и пишем значение no. Далее добавляем строку AllowUsers sshuser.

BR-SRV:

Устанавливаем ansible:

apt install -y ansible sshpass

Создаем каталог и изменяем конфигурацию:

mkdir -p /etc/ansible

chown $USER:$USER /etc/ansible

Заходим в /etc/ansible/ansible.cfg и пишем:

[defaults]

inventory = /etc/ansible/hosts

remote_user = sshuser

host_key_checking = False

deprecation_warnings = False

interpreter_python = /usr/bin/python3

Теперь заходим в /etc/ansible/hosts и пишем:

[hq]

HQ-SRV ansible_host=192.168.100.2 ansible_user=sshuser ansible_port=2026

HQ-CLI ansible_host=192.168.200.2 ansible_user=sshuser ansible_port=2026

HQ-RTR anisble_host=172.16.1.2 ansible_user=sshuser ansible_port=2026

[br]

BR-SRV ansible_connection=local ansible_user=local ansible_port=2026

BR-RTR ansible_host=172.16.2.2 ansible_user=sshuser ansible_port=2026

[all:vars]

ansible_become=yes

host_key_checking=False

Создаем ключ на BR-SRV: *ssh-keygen -t rsa -b 4096* и копируем его на остальные машины:

ssh-copy-id -p 2026 sshuser@hq-srv

ssh-copy-id -p 2026 sshuser@hq-cli

ssh-copy-id -p 2026 sshuser@hq-rtr

ssh-copy-id -p 2026 sshuser@br-srv

ssh-copy-id -p 2026 sshuser@br-rtr

И проверяем соединение с ansible:

ansible all -m ping

```
root@BR-SRV:/home/br-srv# ansible all -m ping
BR-SRV | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
HQ-RTR | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
BR-RTR | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
HQ-SRV | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
HQ-CLI | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

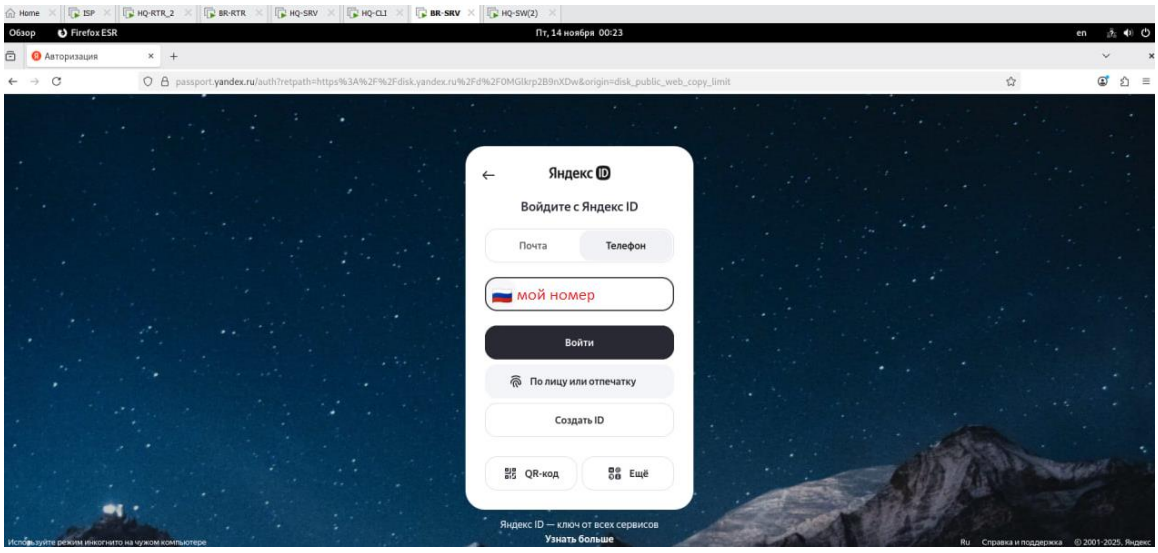
Должно быть все грин, ред флаг красиво и заманчиво, но с ним сеть не построишь….

Docker

Устанавливаем dokcer и docker-compose:

apt install docker docker-compose

Скачиваем additional по ссылке https://disk.yandex.ru/d/0MGlkrp2B9nXDw, используя радикальные меры:



Загружаем образы:

docker load -i /mnt/additional/docker/mariadb_latest.tar

docker load -i /mnt/additional/docker/site_latest.tar

Создаем директорию, переходим в нее и там создаем файл:

mkdir -p /opt/testapp

cd /opt/testapp

nano docker-compose.yml

Заполняем файл:

```
version: '3.8'

services:
  testapp:
    image: site:latest
    container_name: testapp
    ports:
      - "8080:8000"
    depends_on:
      - db
    environment:
      - DB_HOST=db
      - DB_NAME=testdb
      - DB_TYPE=maria
      - DB_USER=test
      - DB_PASS=P@ssw0rd
      - SERVER_PORT=8080
    restart: unless-stopped

  db:
    image: mariadb:10.11
    container_name: db
```

environment:

MYSQL_ROOT_PASSWORD: rootP@ssw0rd

MYSQL_DATABASE: testdb

MYSQL_USER: test

MYSQL_PASSWORD: P@ssw0rd

volumes:

- db_data:/var/lib/mysql

restart: unless-stopped

volumes:

db_data:

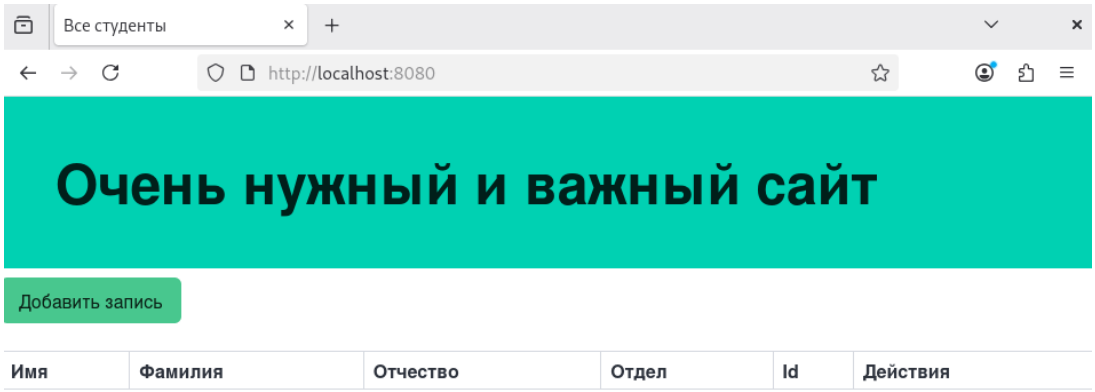
Запускаем docker:

docker-compose up -d

Переходим в браузер и вводим:

Localhost:8080

Должно быть так:



Apache

HQ-SRV:

Второй способ additional iso – это скачать ее на хост, в vmware присоединить его и написать:

mkdir -p /mnt/additional

mount -t iso9660 /dev/sr0 /mnt/additional

Устанавливаем необходимые пакеты:

apt install apache2 mariadb-server php php-mysql php-cli php-gd

Включаем apache и mariadb:

systemctl enable apache2 mariadb

systemctl enable apache2 mariadb

Заходим в mariadb, создаем бд, пользователя и даем права:

mariadb -u root

CREATE DATABASE webdb;

CREATE USER web@localhost IDENTIFIED BY 'P@ssw0rd';

GRANT ALL PRIVILEGES ON webdb. TO web@localhost;*

FLUSH PRIVILEGES;

exit;

Импортируем бд:

mariadb -u root -p webdb < /mnt/additional/web/dump.sql

Копируем файлы приложения в каталог Apache:

cp /mnt/additional/web/index.php /var/www/html/

Создаем папку images и помещаем туда картинку:

mkdir /var/www/html/images

cp /mnt/additional/web/logo.png /var/www/html/images/

Копируем index.php:

cp /mnt/additional/web/index.php /var/www/html/

Устанавливаем права доступа:

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

chmod -R 755 /var/www/html/

Заходим в файл /var/www/html/index.php и редактируем несколько строк:

$servername = "localhost";

$username = "web";

$password = "P@ssw0rd";

$dbname = "webdb";

Включаем модуль:

/usr/sbin/a2enmod rewrite

И потребуется перезагрузить apache:

systemctl restart apache2

Разрешаем HTTP трафик:

/usr/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT

Заходим в /etc/apache2/mods-enabled/dir.conf, удаляем всё содержимое и пишем:

apache

<IfModule mod_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

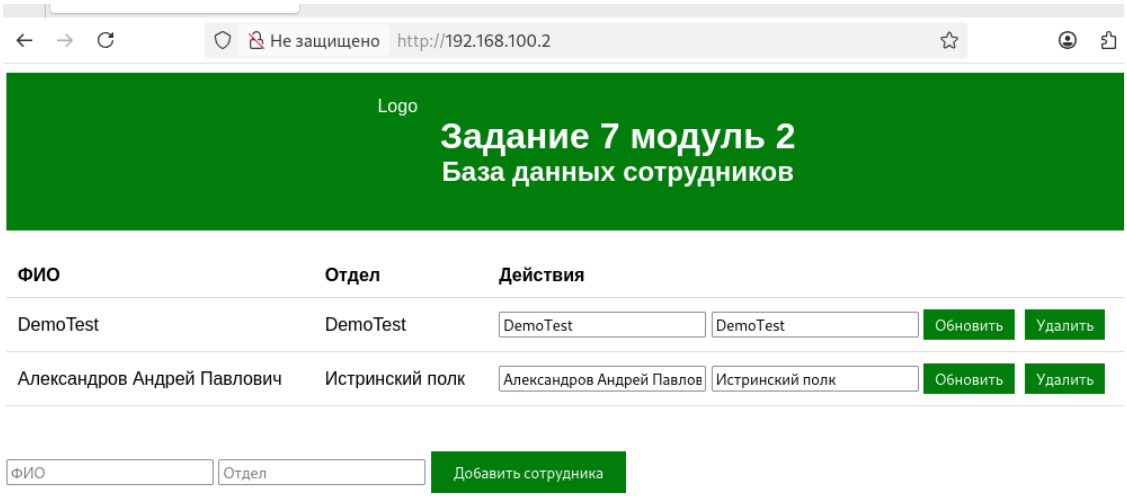
Переименовываем:

mv /var/www/html/index.html /var/www/html/index.html.backup

И перезагружаем:

systemctl restart apache2

Заходим в браузер и пишем адрес HQ-SRV, должна появиться такая страница:



Статическая трансляция портов

HQ-RTR:

/sbin/iptables -t nat -A POSTROUTING -o ens34 -j MASQUERADE

/sbin/iptables -t nat -A PREROUTING -d 172.16.1.2 -p tcp --dport 8080 -j DNAT --to-destination 192.168.100.2:8080

/sbin/iptables -t nat -A PREROUTING -d 172.16.1.2 -p udp --dport 8080 -j DNAT --to-destination 192.168.100.2:8080

/sbin/iptables -t nat -A PREROUTING -d 172.16.1.2 -p tcp --dport 2026 -j DNAT --to-destination 192.168.100.2:2026

/sbin/iptables -t nat -A PREROUTING -d 172.16.1.2 -p udp --dport 2026 -j DNAT --to-destination 192.168.100.2:2026

BR-RTR:

/sbin/iptables -t nat -A POSTROUTING -o ens34 -j MASQUERADE

/sbin/iptables -t nat -A PREROUTING -d 172.16.2.2 -p tcp --dport 8080 -j DNAT --to-destination 192.168.2.2:8080

/sbin/iptables -t nat -A PREROUTING -d 172.16.2.2 -p udp --dport 8080 -j DNAT --to-destination 192.168.2.2:8080

/sbin/iptables -t nat -A PREROUTING -d 172.16.2.2 -p tcp --dport 2026 -j DNAT --to-destination 192.168.2.2:2026

/sbin/iptables -t nat -A PREROUTING -d 172.16.2.2 -p udp --dport 2026 -j DNAT --to-destination 192.168.2.2:2026

NGINX

ISP:

Устанавливаем nginx:

apt install nginx

Включаем nginx:

systemctl enable nginx

systemctl restart nginx

Заходим в nano /etc/nginx/sites-available/reverse_proxy.conf и там пишем (здесь сначала идет адрес hq-srv, потом br-srv):

```
upstream hq_srv_app {
    server 192.168.100.2:80;
}
```

```
upstream testapp_app {
    server 192.168.2.2:8080;
}

server {
    listen 80;
    server_name web.au-team.irpo;

    location / {
        proxy_pass http://hq_srv_app;
        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;
    }
}

server {
    listen 80;
    server_name docker.au-team.irpo;

    location / {
        proxy_pass http://testapp_app;
        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;
    }
}
```

Теперь активируем сайт:

*ln -s /etc/nginx/sites-available/reverse_proxy.conf *

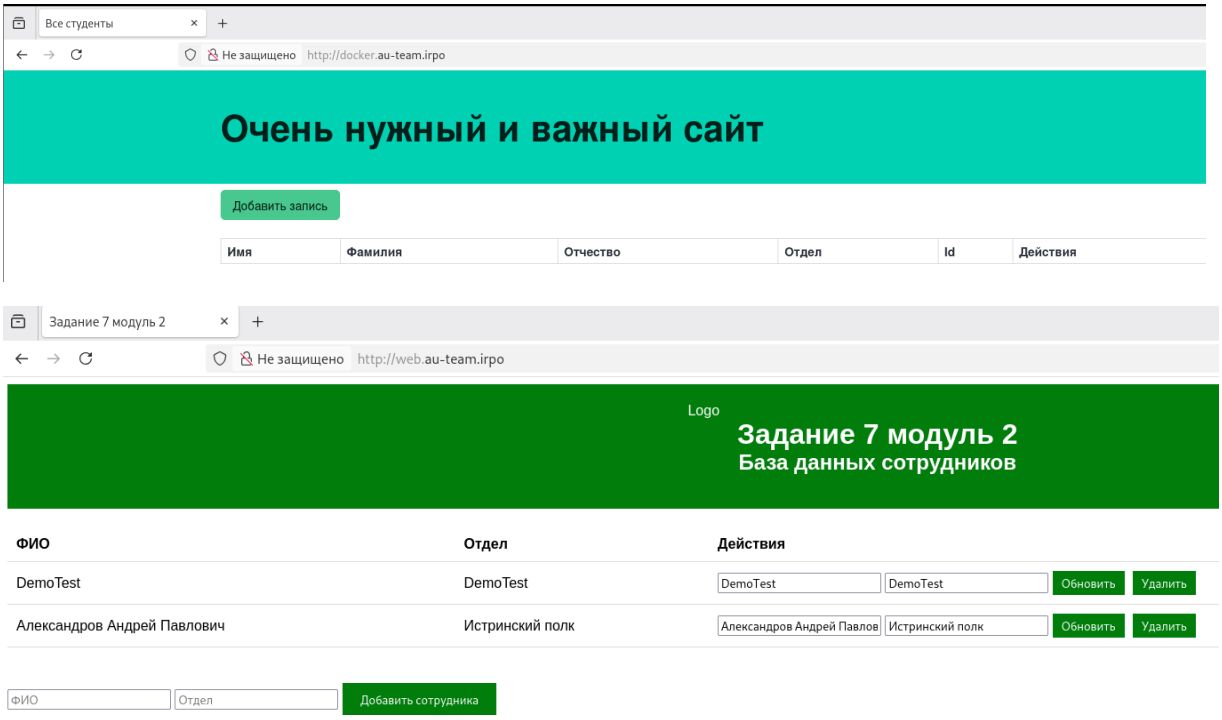
/etc/nginx/sites-enabled/reverse_proxy.conf

Проверяем на клиенте:

http://docker.au-team.irpo/

http://web.au-team.irpo/

Должно быть так:



Web-based аутентификация:

ISP:

Устанавливаем дополнительные пакеты:

apt install apache2-utils

Создаем файл с пользователем (WEB) и после этого вводим пароль(P@ssw0rd):

htpasswd -c /etc/nginx/.htpasswd WEB

Заходим в этот файл nano /etc/nginx/sites-available/reverse_proxy.conf и добавляем две строки:

auth_basic "Restricted Access";

auth_basic_user_file /etc/nginx/.htpasswd;

Чтобы было вот так:

```
server {
    listen 80;
    server_name web.au-team.irpo;

        auth_basic "Restricted Access";
        auth_basic_user_file /etc/nginx/.htpasswd;

    location / {
```

И перезагружаем:

systemctl reload nginx

Яндекс

HQ-CLI:

Пишем в браузере: скачать яндекс линукс, переходим по ссылке, скачиваем, заходим в загрузки, находим этот файл, нажимаем пкм и открываем с помощью установки приложений, дальше всё понятно. И готово, модуль готов