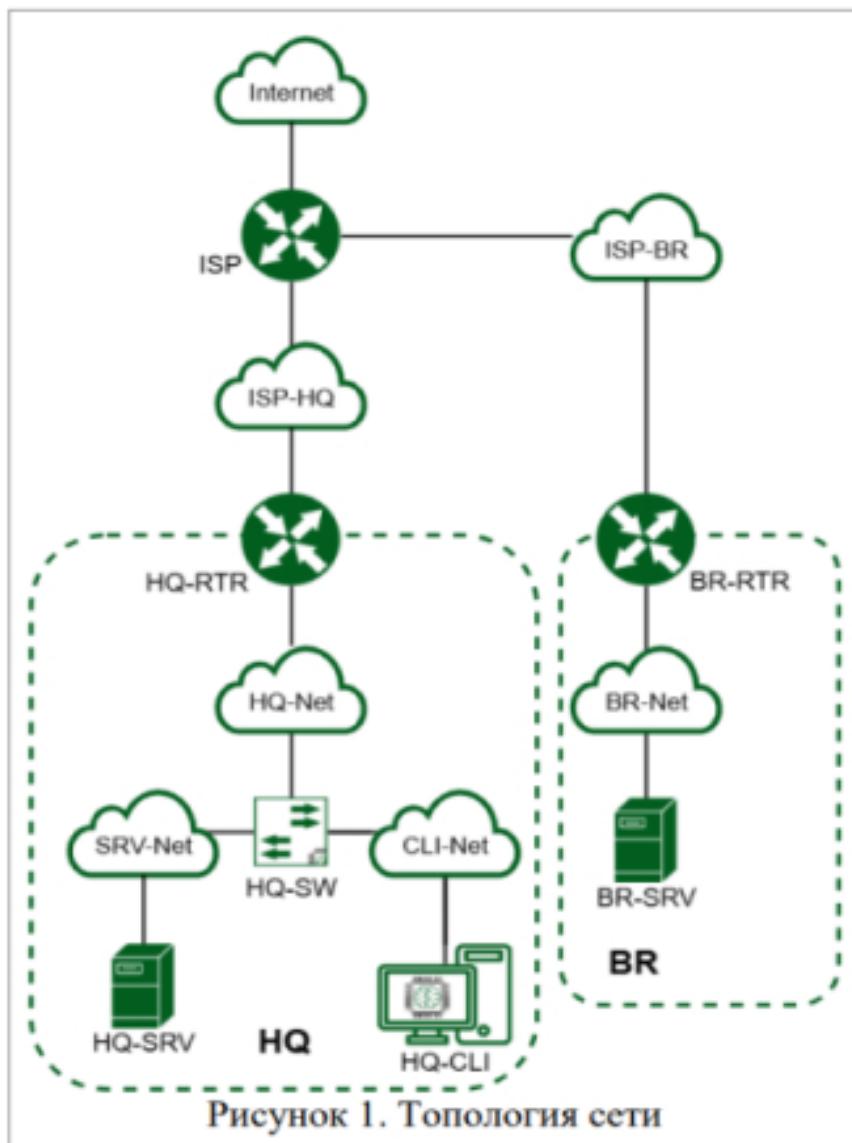


Модуль 2



1. Настройте доменный контроллер Samba на машине BR-SRV

- Создайте 5 пользователей для офиса HQ: имена пользователей формата userN^o.hq. Создайте группу hq, введите в эту группу созданных пользователей
- Ведите 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 не имеют права
- Выполните импорт пользователей из файла `users.csv`. Файл будет располагаться на виртуальной машине BR-SRV в папке `/opt`

Сконфигурируем основной доменный контроллер на BR-SRV

На BR-SRV создадим файл `resolv.conf` в `/etc/net/iface/ens19/` и пропишем в нем адрес днс сервера

```
[root@br-srv ~]# echo "nameserver 77.88.8.8" > /etc/net/iface/ens19/resolv.conf  
[root@br-srv ~]# systemctl restart network
```

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

Установим samba

```
[root@br-srv ~]# apt-get install task-samba-dc
```

Так как Samba в режиме контроллера домена (Domain Controller, DC) использует как свой LDAP, так и свой сервер Kerberos, несовместимый с MIT Kerberos, перед установкой необходимо остановить конфликтующие службы `krb5kdc` и `slapd`, а также `bind` (если сервер только установлен, то этого не будет):

```
for i in smb nmb krb5kdc slapd bind; do systemctl disable $i --now; done
```

или

```
systemctl disable --now smb nmb krb5kdc slapd bind
```

Должно быть установлено правильное имя узла и домена для сервера. Для этого в файл `/etc/sysconfig/network` необходимо добавить строку:

```
[root@br-srv ~]# vim /etc/sysconfig/network
```

```
# Used by rc.sysinit to setup system hostname at boot.  
HOSTNAME=br-srv.au-team.irpo
```

Далее выполним команду

```
[root@br-srv ~]# domainname au-team.irpo
```

Для корректного распознавания всех локальных DNS-запросов в файле `/etc/resolvconf.conf` должна присутствовать строка:

```
#Configuration files for dnsmasq subscriber.  
dnsmasq_conf=/etc/dnsmasq.conf.d/60-resolvconf  
dnsmasq_resolv=/etc/resolv.conf.dnmasq  
  
name_servers=127.0.0.1  
vi...
```

и перезапустим сервис resolvconf:

```
[root@br-srv ~]# resolvconf -u
```

Перед созданием домена необходимо очистить базы и конфигурацию Samba (домен, если он создавался до этого, будет удалён):

```
[root@br-srv ~]# rm -f /etc/samba/smb.conf  
[root@br-srv ~]# rm -rf /var/lib/samba  
[root@br-srv ~]# rm -rf /var/cache/samba  
[root@br-srv ~]# mkdir -p /var/lib/samba/sysvol
```

Запускаем интерактивную установку контроллера домена:

Все параметры должны подставляться автоматически корректными в []

```
[root@br-srv ~]# samba-tool domain provision  
Realm (AU-TEAM.IRPO):  
Domain [AU-TEAM]:  
Server Role (dc, member, standalone) [dc]:  
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:  
DNS forwarder IP address (write 'none' to disable forwarding) [192.168.100.62]:  
Administrator password:  
Retype password:
```

В результате должны получить

```
[IRP0 2025-04-03 14:36:12.007 pid:12990 callid:1166534864 python3.8 samba-provision__init__.py #490: Server Role: active directory domain controller  
[IRP0 2025-04-03 14:36:12.009 pid:12990 callid:1166534864 python3.8 samba-provision__init__.py #499: Realname: br-srv  
[IRP0 2025-04-03 14:36:12.009 pid:12990 callid:1166534864 python3.8 samba-provision__init__.py #500: NetBIOS Domain: AU-TEAM  
[IRP0 2025-04-03 14:36:12.009 pid:12990 callid:1166534864 python3.8 samba-provision__init__.py #501: DNS Domain: au-team.irpo  
[IRP0 2025-04-03 14:36:12.009 pid:12990 callid:1166534864 python3.8 samba-provision__init__.py #502: domain id: 3-1-5-21-172305916-498-18879-520  
0006
```

Или в пакетном режиме установку контроллера домена: (результат будет тот же).

P.S. выбирается что-то одно, либо создание домена в интерактивном режиме, либо в пакетном при помощи одной следующей команды:

```
samba-tool domain provision --realm=au-team.irpo --domain=au-team --  
adminpass='P@ssw0rd' --dns-backend=SAMBA_INTERNAL --option="dns  
forwarder=192.168.100.62" --server-role=dc --use-rfc2307
```

где:

--realm=champ.first - имя области Kerberos (LDAP), и DNS имя домена;

--domain=champ - имя домена (имя рабочей группы);

--adminpass='P@ssw0rd' - пароль основного администратора домена;
--dns-backend=SAMBA_INTERNAL - бэкенд DNS-сервера;
--option="dns forwarder=192.168.100.62" - внешний DNS-сервер, чтобы DC мог разрешать внешние доменные имена (HQ-SRV);
--server-role=dc - тип серверной роли;
--use-rfc2307 - позволяет поддерживать расширенные атрибуты типа UID и GID в схеме LDAP и ACL на файловой системе Linux.

После создания домена в интерактивном или пакетном режиме - необходимо запустить и добавить в автозагрузку службу samba:

```
[root@br-srv ~]# systemctl enable --now samba
```

Проверяем:

```
[root@br-srv ~]# systemctl status samba
```

```
[root@br-srv ~]# systemctl status samba
● samba.service - Samba AD Daemon
  Loaded: loaded (/lib/systemd/system/samb
  Active: active (running) since Mon 2025-
    Docs: man:samba(8)
           man:samba(7)
```

Настроим Kerberos (скопируем настройки)

```
[root@br-srv ~]# cp -f /var/lib/samba/private/krb5.conf /etc/krb5.conf
cp: overwrite '/etc/krb5.conf'? y
```

Немного подправим записи в файле

```
[realms]
AU-TEAM.IRPO = {
    default_domain = AU-TEAM.IRPO
}
```

Перезапустим Samba

```
[root@br-srv ~]# systemctl restart samba
```

Проверяем:

```
[root@br-srv ~]# samba-tool domain info 127.0.0.1
Forest          : au-team.irpo
Domain          : au-team.irpo
Netbios domain  : AU-TEAM
DC name         : br-srv.au-team.irpo
DC netbios name : BR-SRV
Server site     : Default-First-Site-Name
Client site     : Default-First-Site-Name
```

Просмотрим службы:

По умолчанию общие ресурсы netlogon и sysvol нужны для функционирования сервера AD и создаются в smb.conf в процессе развертывания/модернизации.

```
[root@br-srv ~]# smbclient -L localhost -U administrator
Password for [AU-TEAM\administrator]:
```

Sharename	Type	Comment
sysvol	Disk	
netlogon	Disk	
IPC\$	IPC	IPC Service (Samba 4.19.9-alt5)
SMB1 disabled -- no workgroup available		

В итоге файл /etc/samba/smb.conf должен выглядеть так:

```
# Global parameters
[global]
    dns forwarder = 192.168.100.62
    netbios name = BR-SRV
    realm = AU-TEAM.IRPO
    server role = active directory domain controller
    workgroup = AU-TEAM
    idmap_ldb:use rfc2307 = yes

[sysvol]
    path = /var/lib/samba/sysvol
    read only = No

[netlogon]
    path = /var/lib/samba/sysvol/au-team.irpo/scripts
    read only = No
```

Файл /etc/krb5.conf должен выглядеть так:

```
[libdefaults]
    default_realm = AU-TEAM.IRPO
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
AU-TEAM.IRPO = {
    default_domain = AU-TEAM.IRPO
}

[domain_realm]
    br-srv = AU-TEAM.IRPO
```

Проверка Kerberos (имя домена должно быть в верхнем регистре):

```
[root@br-srv ~]# kinit administrator@AU-TEAM.IRPO
Password for administrator@AU-TEAM.IRPO:
Warning: Your password will expire in 41 days on Mon 19 May 2025 12:57:53 PM MSK
[root@br-srv ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@AU-TEAM.IRPO

Valid starting     Expires            service principal
04/07/2025 13:20:29  04/07/2025 23:20:29  krbtgt/AU-TEAM.IRPO@AU-TEAM.IRPO
                  renew until 04/08/2025 13:20:25
```

Создаём группу hq

```
[root@br-srv ~]# samba-tool group add hq
Added group hq
```

Далее создадим пользователей

Так как пользователей не много то можно воспользоваться командой

```
samba-tool user add user1.hq P@ssw0rd (и так еще 4 раза)
```

Но, мы будем использовать регулярные выражения для их создания и добавления в группу:

```
[root@br-srv ~]# for i in {1..5}; do
> samba-tool user add user${i}.hq P@ssw0rd;
> samba-tool user setexpiry user${i}.hq --noexpiry;
> samba-tool group addmembers "hq" user${i}.hq;
> done
```

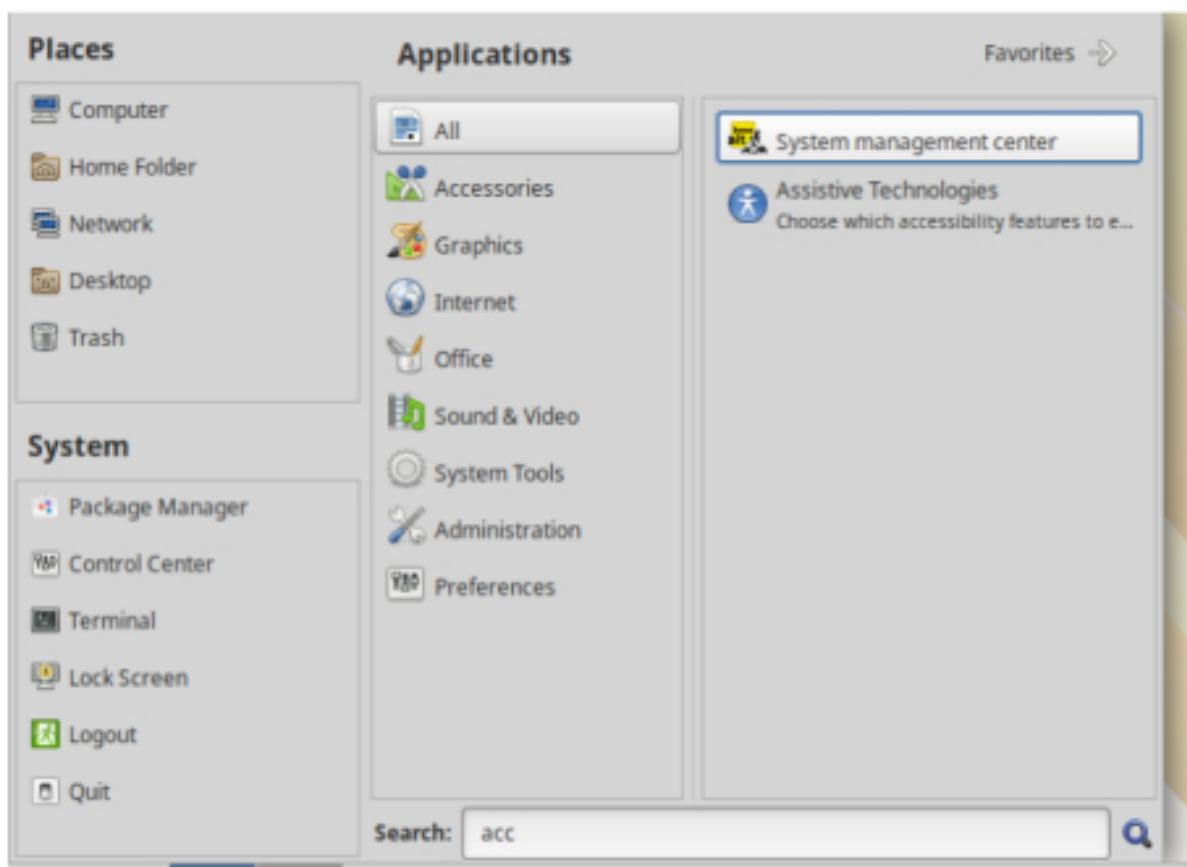
samba-tool user setexpiry --noexpiry <username> - отключает срок действия пароля

Проверяем:

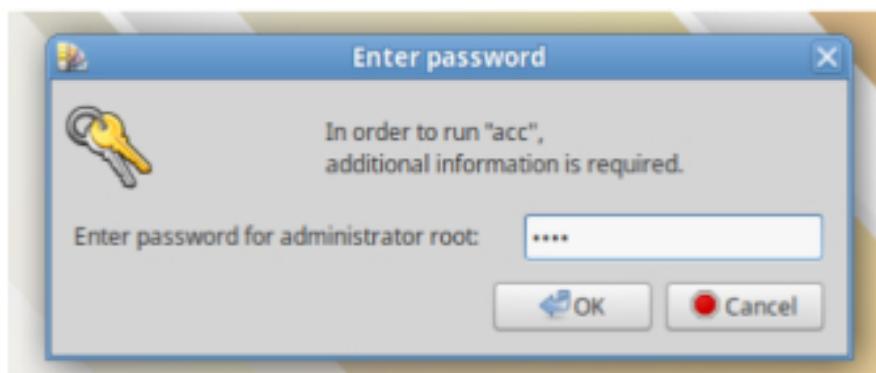
```
[root@br-srv ~]# samba-tool group listmembers hq
user1.hq
user2.hq
user5.hq
user3.hq
user4.hq
```

Введем клиентскую машину в домен

Переходим на HQ-CLI

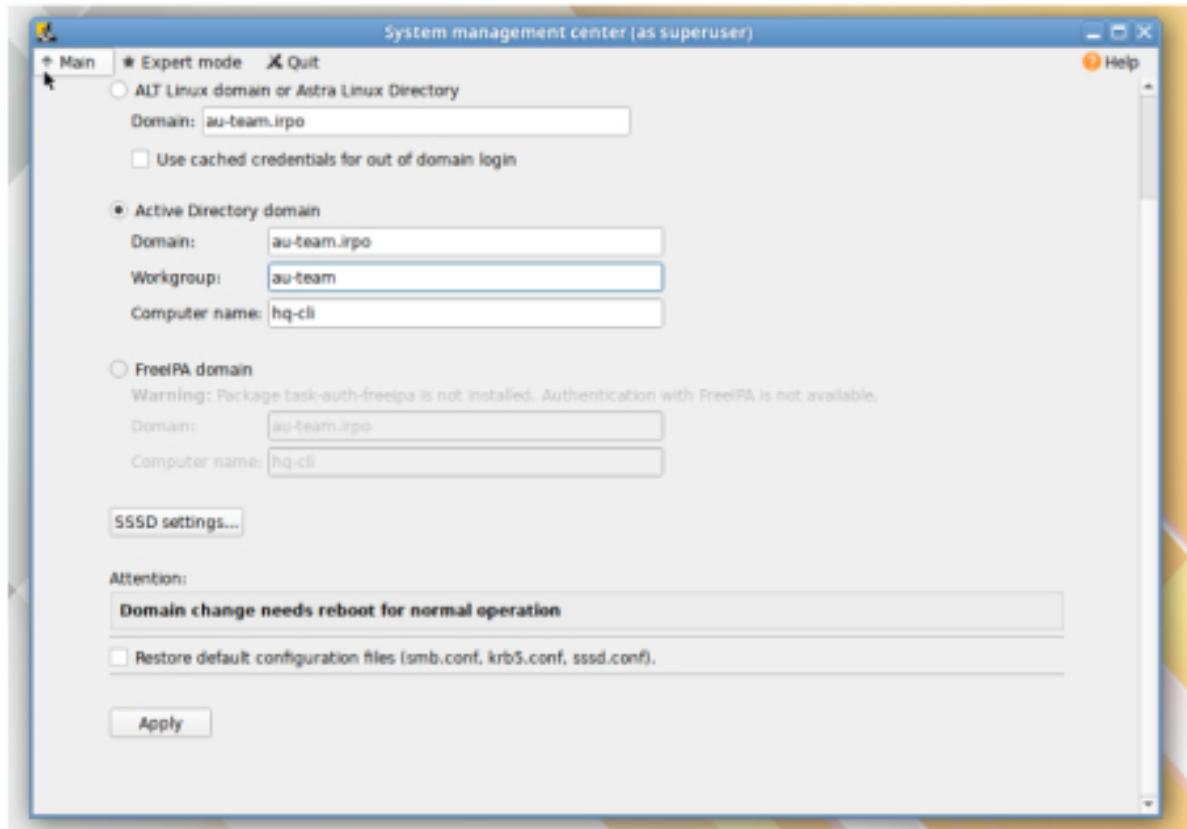


Пароль от root



ement
or [Authentication](#) [Dir](#)

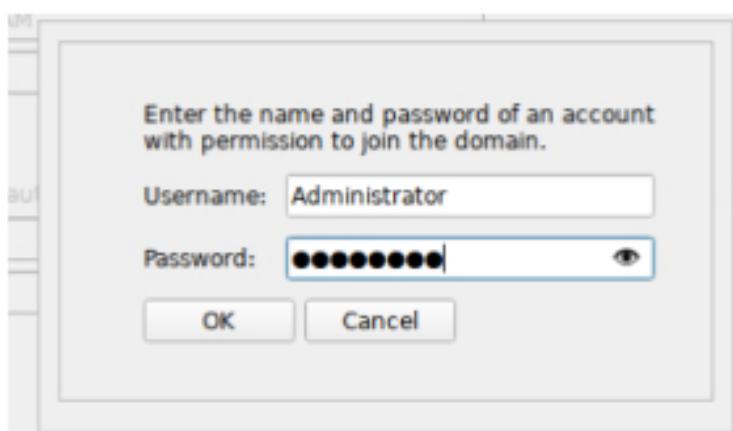




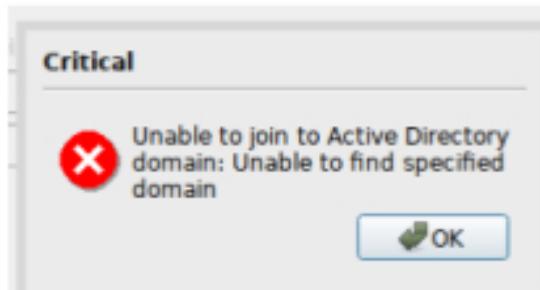
Вводим пароль, который указывали при создании контроллера домена

Administrator password:
Retype password:

Если все успешно, то



На данном этапе получим ошибку



Вернемся к настройкам DHCP (на HQ-RTR)

Изменим DNS сервер на BR-SRV

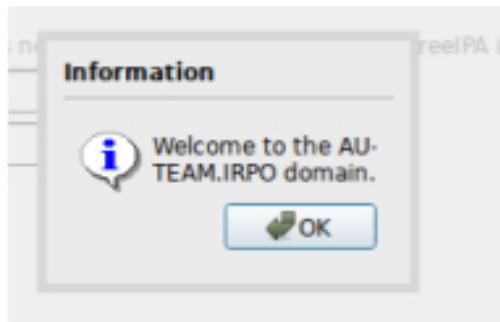
```
hq-rtr(config)#dhcp-server 1
hq-rtr(config-dhcp-server)#pool HQ-CLI 1
hq-rtr(config-dhcp-server-pool)#dns 192.168.3.30
hq-rtr(config-dhcp-server-pool)#exit
hq-rtr(config-dhcp-server)#write
```

Вернемся на HQ-CLI

Выполним перезапуск сетевой службы

```
user@hq-cli ~ $ systemctl restart network
```

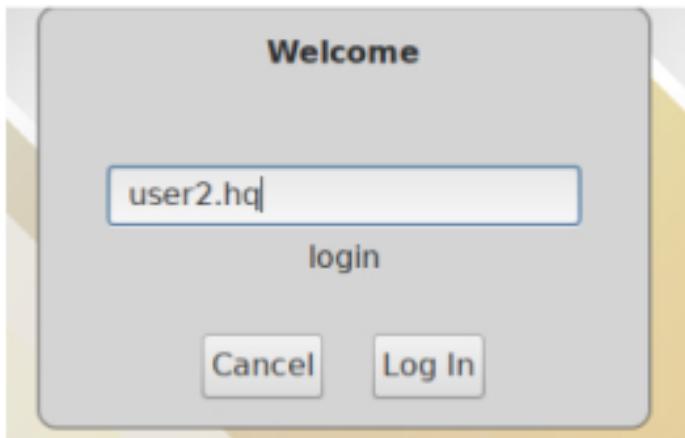
Повторим ввод клиентской машины в домен. Теперь должна добавиться



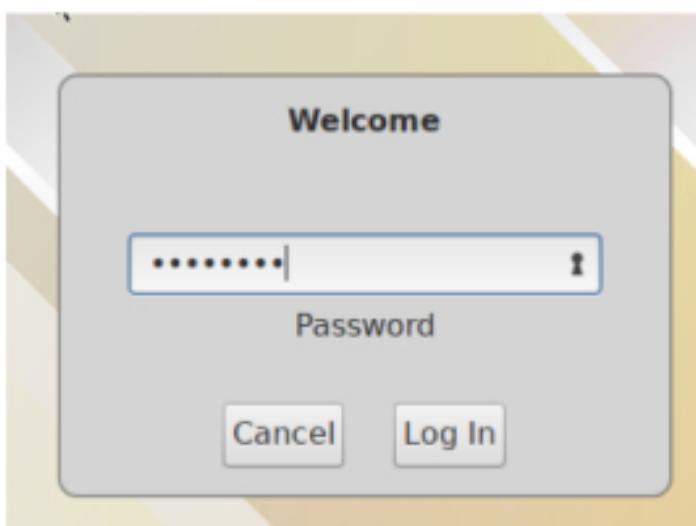
Перезагружаем обязательно. Проверим, что пользователи группы `hq` имеют право аутентифицироваться на клиентском ПК. Зайдем под пользователем, которого мы создали в Samba, например, `user2.hq`

Сменим пользователя

Зайдем под новым



Введем пароль



Чтобы настроить права созданных нами пользователей, нужно установить ещё один пакет на BR-SRV, но перед этим нужно подключить нужный репозиторий следующей командой:

```
apt-repo add rpm http://altrepo.ru/local-p10 noarch local-p10
```

```
[root@br-srv apt]# apt-repo add rpm http://altrepo.ru/local-p10 noarch local-p10
```

Теперь обновляем список пакетов:

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

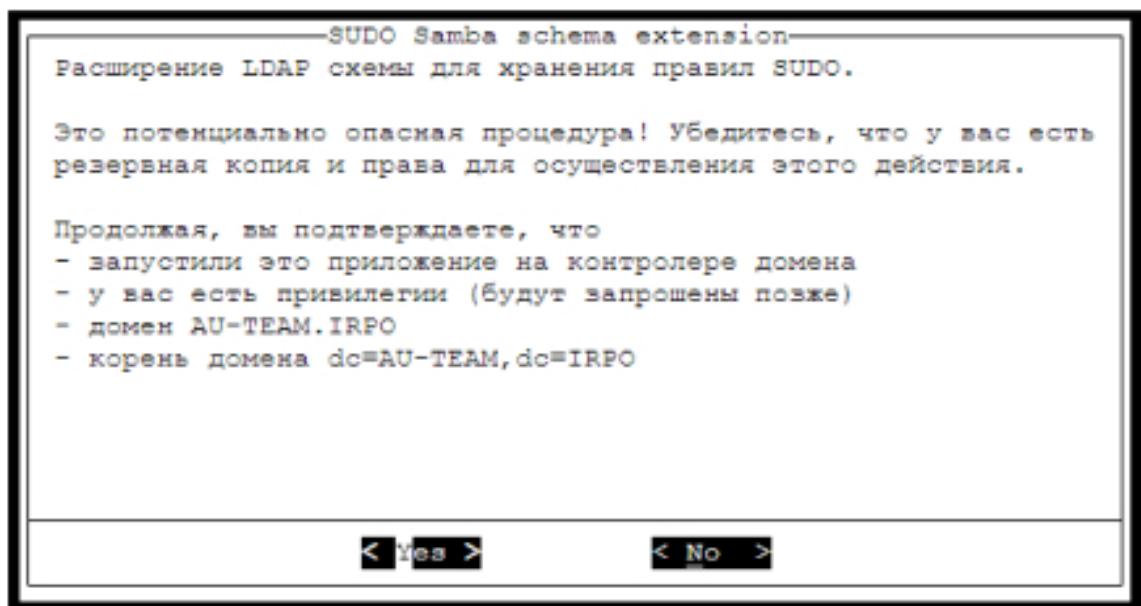
И можем устанавливать нужный нам пакет:

```
[root@hq-srv apt]# apt-get install sudo-samba-schema
```

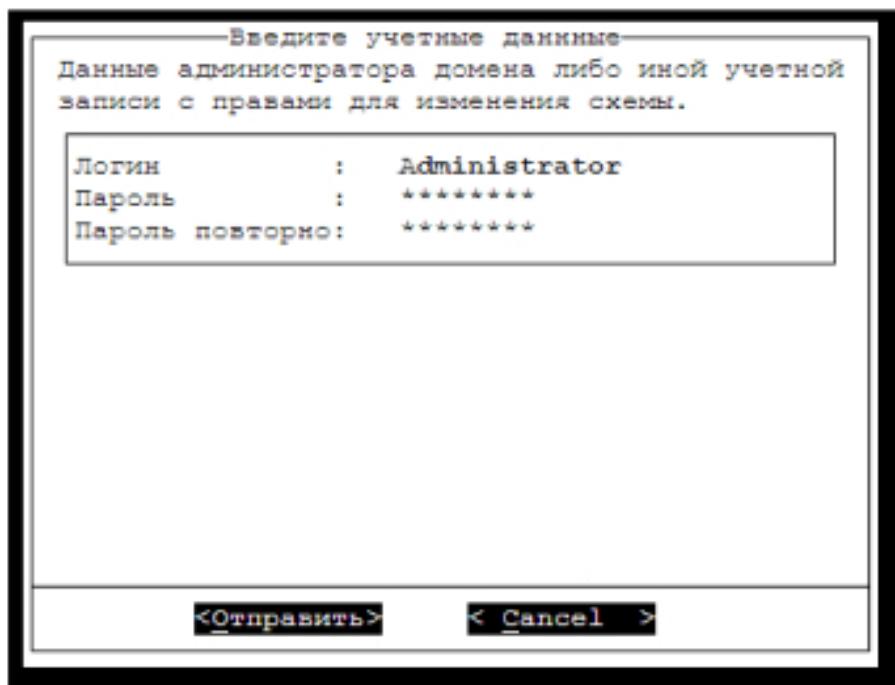
Далее добавляем новую схему следующей командой:

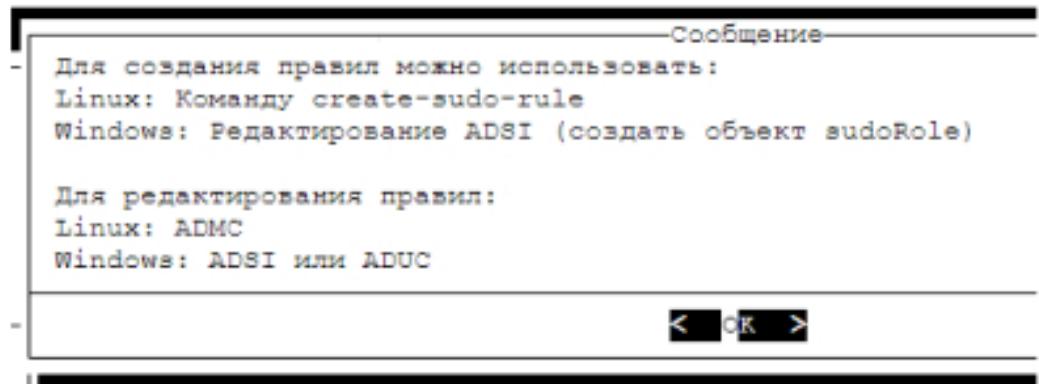
```
[root@hq-srv apt]# sudo-schema-apply
```

Откроется следующее диалоговое окно, нажимаем yes:



Затем у нас попросит пароль от доменного администратора:



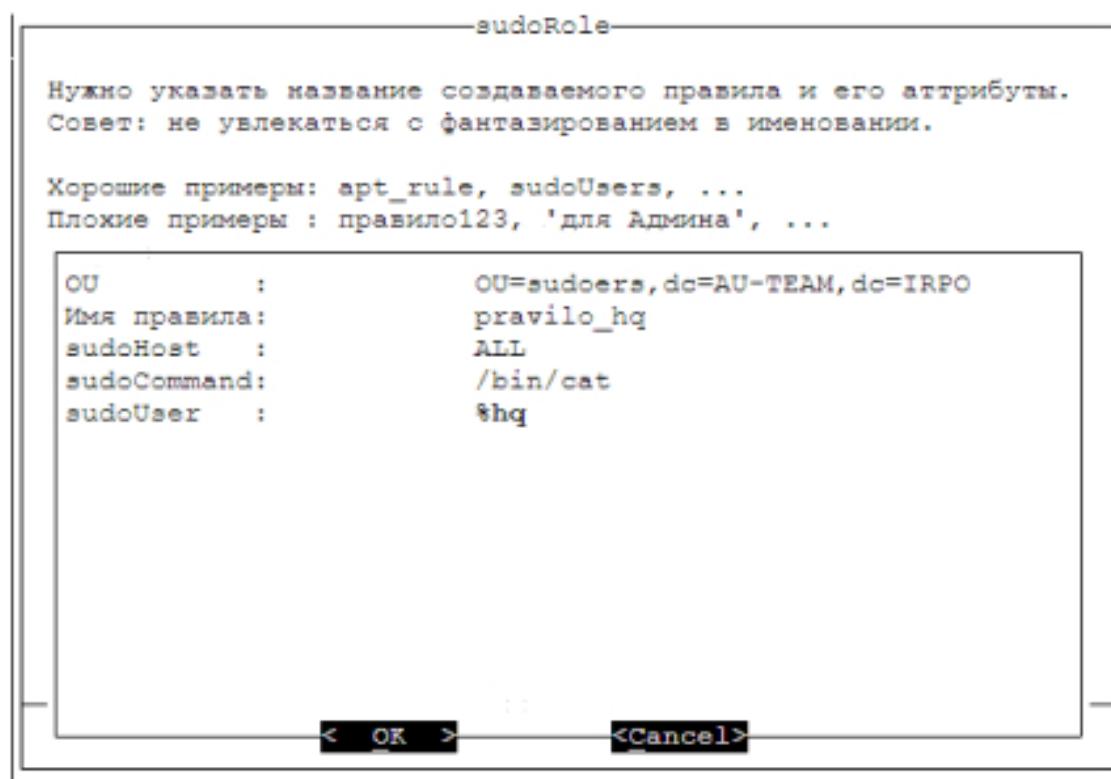


Далее мы создаём новое правило следующей командой (которую он сам предлагает в этом окне):

```
[root@br-srv apt]# create-sudo-rule
```

И вносим следующие изменения (имя правила можно любое):

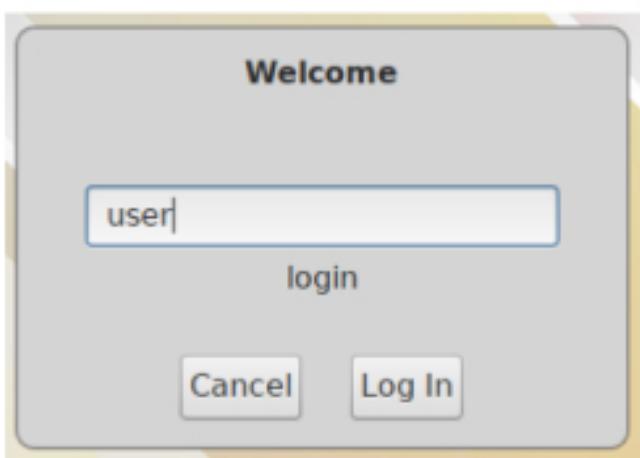
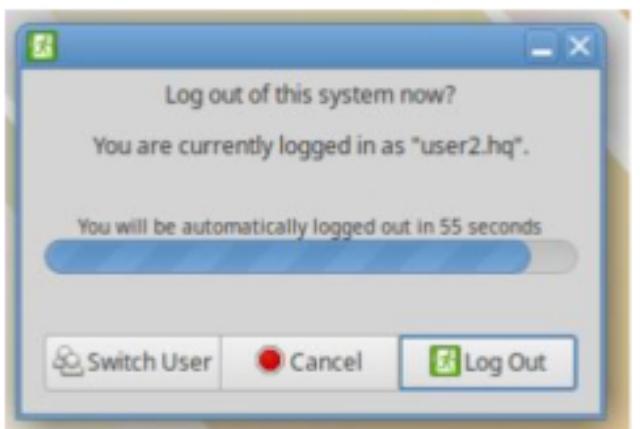
Имя правила : pravilo_hq
sudoCommand : /bin/cat
sudoUser : %hq



При успешном добавлении выведет следующие строки:

```
Added 1 records successfully
Modified CN=pravilol_hd,OU=sudoers,dc=AU-TEAM,dc=IRPO
Modified 1 records successfully
Операция прошла успешно
[root@br-srv apt] #
```

Заходим под локальным пользователем на клиентской машине HQ-CLI и получаем права root:



```
user@hq-cli ~ $ su -
Password:
hq-cli ~ #
```

Обновляем список пакетов:

```
hq-cli ~ # apt-get update
```

И поставим пакет admc:

```
hq-cli ~ # apt-get install admc
```

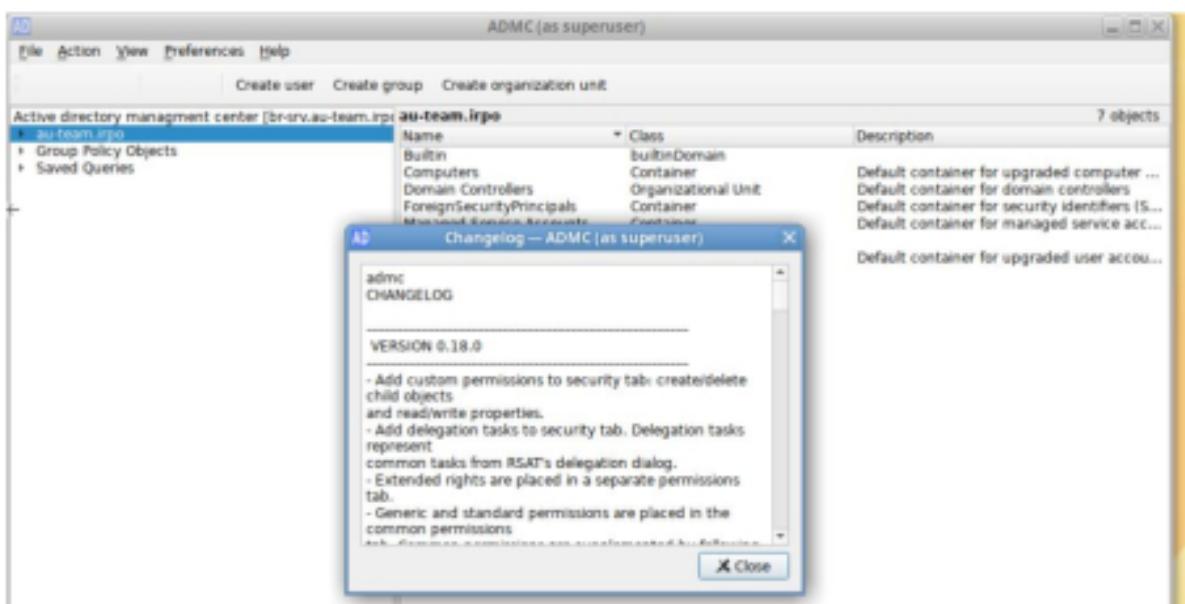
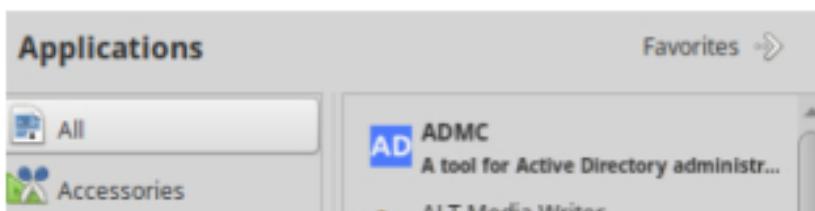
Затем создаём тикет доменного администратора, чтобы получить права на редактирование правил на сервере:

```
hq-cli ~ # kinit administrator  
Password for administrator@AU-TEAM.IRPO:  
Warning: Your password will expire in 41 day
```

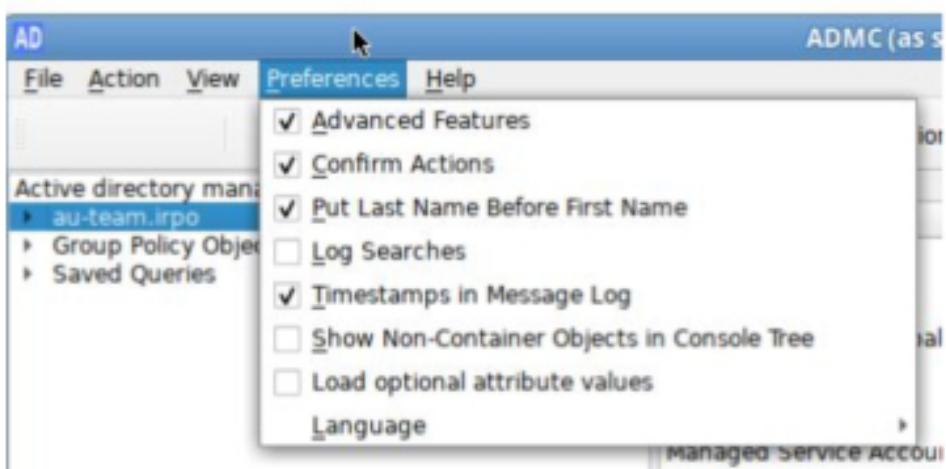
И запускаем admc:

```
hq-cli ~ # admc
```

Или



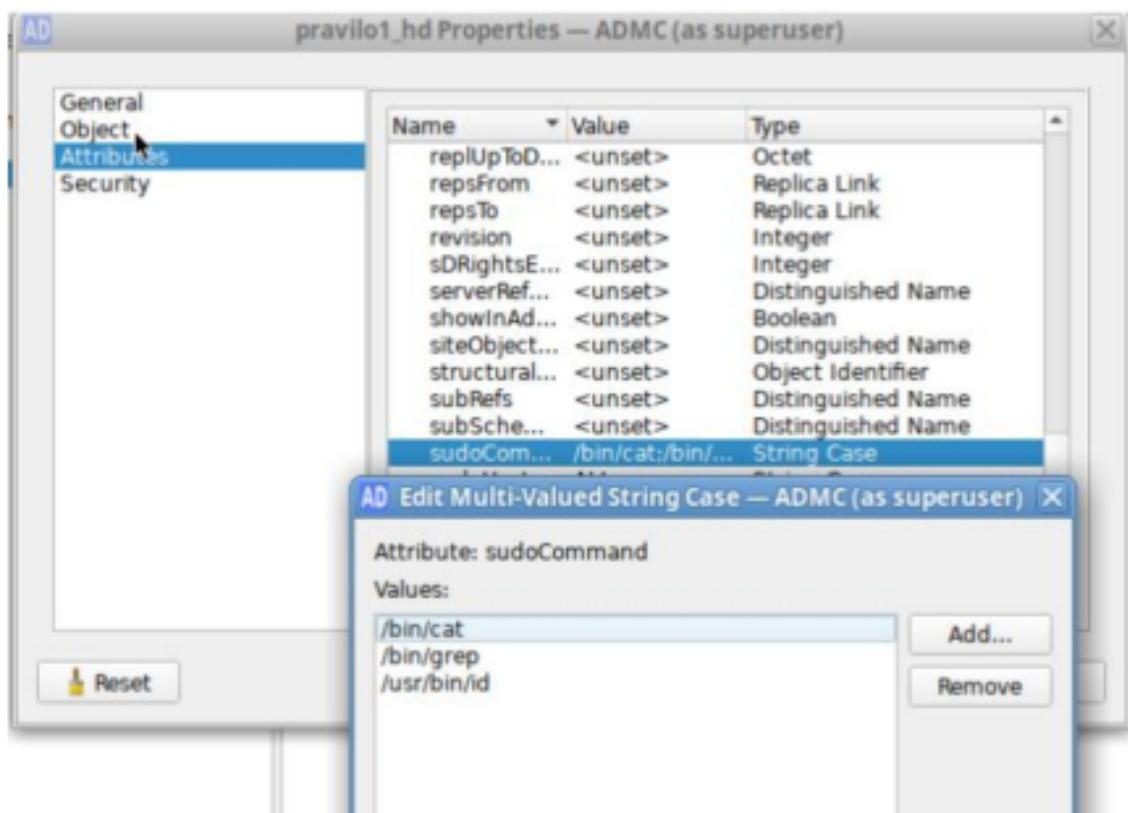
Включим дополнительные возможности через настройки:



Поменяем опцию sudoOption в созданном нами ранее правиле pravilo_hd на значение !authenticate (правило всегда будет находиться в OU с названием sudoers):

The screenshot shows the Active Directory Management Center interface. On the left, the navigation pane lists various objects under 'au-team.ipn'. The 'sudoers' object is selected. A table titled 'sudoers' displays a single entry: 'Name: pravilo1_hd' and 'Class: sudoRole'. An 'AD' dialog box is open over the table, titled 'pravilo1_hd Properties — ADMC (as superuser)'. The 'Attributes' tab is selected. A table shows attributes like 'revision', 'sRightsE...', 'serverRef...', etc., with 'sudoOption' highlighted. Below the table are buttons for 'Edit...', 'Load optional attributes', and 'Filter'. At the bottom of the dialog are 'Reset', 'OK', 'Cancel', and 'Apply' buttons. In the foreground, another 'AD' dialog box is partially visible, titled 'pravilo1_hd Properties — ADMC (as superuser)'. It also has an 'Attributes' tab and a table of attributes. The 'sudoOption' row in this table has its value changed to '!authenticate'. This dialog has 'Attribute: sudoOption' at the top and 'Add...', 'Remove...', and 'OK' buttons at the bottom.

И добавим ещё две команды в опцию sudoCommand (grep и id):



Обратите внимание, что путь до id отличается от других команд!

Теперь, чтобы работали все созданные нами правила, нужно зайти на HQ-CLI и установить дополнительные пакеты. Для этого либо закрываем ADMC, либо открываем новую вкладку в терминале:

```
hq-cli ~ # apt-get install sudo libsss_sudo
```

Разрешаем использование sudo:

```
hq-cli ~ # control sudo public
```

Настроим конфигурация в /etc/sssd/sssd.conf

```
services = nss, pam, sudo
```

```
sudo_provider = ad
```

```
root@hq-hp-000:~# cat /etc/sssd/sssd.conf
[sssd]
config_file_version = 2
services = nss, pam, sudo

# Managed by system facility command:
## control sssd-drop-privileges unprivileged user
user = _sssd

# SSSD will not start if you do not configure at least one domain
domains = AU-TEAM.IRPO
[nss]

[pam]
[domain/AU-TEAM.IRPO]
id_provider = ad
sudo_provider = ad
```

Теперь отредактируем /etc/nsswitch.conf:

sudoers: files sss

```
passwd: files sss
shadow: tcb files sss
group: files [SUCCESS=merge] sss role
gshadow:   files
sudoers: files sss
```

Далее перезагрузим HQ-CLI. **reboot**

На данном этапе мы можем проверить настроенные нами права и правильность настроек конфигурационных файлов. Сделать мы это можем под локальной учётной записью, у которой есть права администратора, в нашем случае это просто root. А ещё мы можем открыть вторую сессию нажав сочетание клавиш:

Ctrl+Alt+F2

В дальнейшем мы можем переключаться между ними, т.к. нажатием тех же клавиш, но теперь уже с F1 мы вернемся на первую нашу сессию с графической оболочкой.

Ctrl+Alt+F1

После того как зашли на вторую сессию, логинимся под root и перезагружаем службу sssd:

```
hq-cli login: root
Password:
Last login: Thu Mar  7 21:08:38 MSK 2024 on ttys0
hq-cli ~ # rm -rf /var/lib/sss/db/*
hq-cli ~ # sss_cache -E
hq-cli ~ # systemctl restart sssd
hq-cli ~ #
```

Теперь проверим, какие правила для sudoers получил наш доменный пользователь:

```
hq-cli ~ # sudo -l -U user1.hq
Matching Defaults entries for user1.hq on hq-cli:
    env_keep+="DISPLAY XAUTHORITY"

User user1.hq may run the following commands on hq-cli:
    (root) NOPASSWD: /bin/cat, /bin/grep, /usr/bin/id
```

Проверим:

```
user1.hq@hq-cli ~ $ cat /etc/passwd | grep root && id root
root:x:0:0:System Administrator:/root:/bin/bash
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
(wheel),19(proc)
```

Так как команды выполняется, то привилегии повышенны.

Импортируем пользователей из таблицы Users.csv на BR-SRV.

Для этого нам нужно скачать файл Users.csv, но на ДЭ он уже будет скачан и лежать в каталоге /opt. Мы же, для обучения, можем создать его сами (на любом компьютере не входящим в стенд), разместить в облаке, скачать и переместить его в /opt. Для этого на BR-SRV выполним следующие команды:

```
curl -L https://bit.ly/3C1nEYz > /root/users.zip
```

```
[root@br-srv ~]# curl -L https://bit.ly/3C1nEYz > /root/users.zip
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
                                         Dload  Upload   Total   Spent    Left  Speed
100  181  100  181    0     0  145      0  0:00:01  0:00:01  --:--:--  145
100 10262  100 10262    0     0  5527      0  0:00:01  0:00:01  --:--:--   0
[root@br-srv ~]# unzip /root/users.zip
Archive:  /root/users.zip
  inflating: Users.csv
[root@br-srv ~]# mv /root/Users.csv /opt/Users.csv
```

Создаём скрипт с именем import.sh и пишем туда следующий код:

```
[root@br-srv opt]# vim /opt/import.sh
```

```
#!/bin/bash
csv_file="/opt/Users.csv"
while IFS=";" read -r firstName lastName role phone ou street zip city country password;
do
    if [ "$firstName" == "First Name" ]; then
        continue
    fi
    username="${firstName,,}.${lastName,,}"
    sudo samba-tool user add "$username" P@ssw0rd
done < "$csv_file"

#!/bin/bash
csv_file="/opt/Users.csv"
while IFS=";" read -r firstName lastName role phone ou street zip city country password;
do
    if [ "$firstName" == "First Name" ]; then
        continue
    fi
    username="${firstName,,}.${lastName,,}"
    sudo samba-tool user add "$username" P@ssw0rd
done < "$csv_file"
```

Сохраняем этот файл и выдаём ему право на выполнение и запускаем его:

```
[root@br-srv opt]# chmod +x /opt/import.sh
```

Запустим скрипт

```
[root@br-srv opt]# bash import.sh
```

Если высокочит ошибка,

```
root is not in the sudoers file.
This incident has been reported to the administrator.
```

то идем редактировать файл /etc/sudoers

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

## Uncomment to allow members of group wheel to execute any command
# WHEEL_USERS ALL=(ALL:ALL) ALL

## Same thing without a password
# WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# SUDO_USERS      ALL=(ALL:ALL) ALL
```

При сохранении изменений добавить знак «!» : wq!

Запустить скрипт еще раз:

```
[root@br-srv opt]# ./import.sh
```

И пользователи из файла будут импортироваться

```
User 'malachi.alexander' added successfully
User 'nelle.alford' added successfully
User 'meredith.arnold' added successfully
User 'hasad.ashley' added successfully
User 'wynne.ashley' added successfully
User 'anjolie.baldwin' added successfully
User 'yolanda.ball' added successfully
User 'alika.barron' added successfully
User 'nolan.barry' added successfully
User 'althea.battle' added successfully
User 'keefe.becker' added successfully
User 'zenia.berg' added successfully
User 'deirdre.bernard' added successfully
User 'raphael.bird' added successfully
User 'rachel.blackburn' added successfully
```

2. Сконфигурируйте файловое хранилище

- При помощи трех 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 автомонтирование в папку /mnt/nfs
- Основные параметры сервера отметьте в отчете

Добавим три дополнительных диска, размером 1Гб каждый, на HQ-SRV.

В оборудовании виртуальной машины смотрим сколько жестких дисков имеется.

У нас он один – добавим еще 3 (Если есть дополнительные по 1 ГБ – то добавлять не нужно)

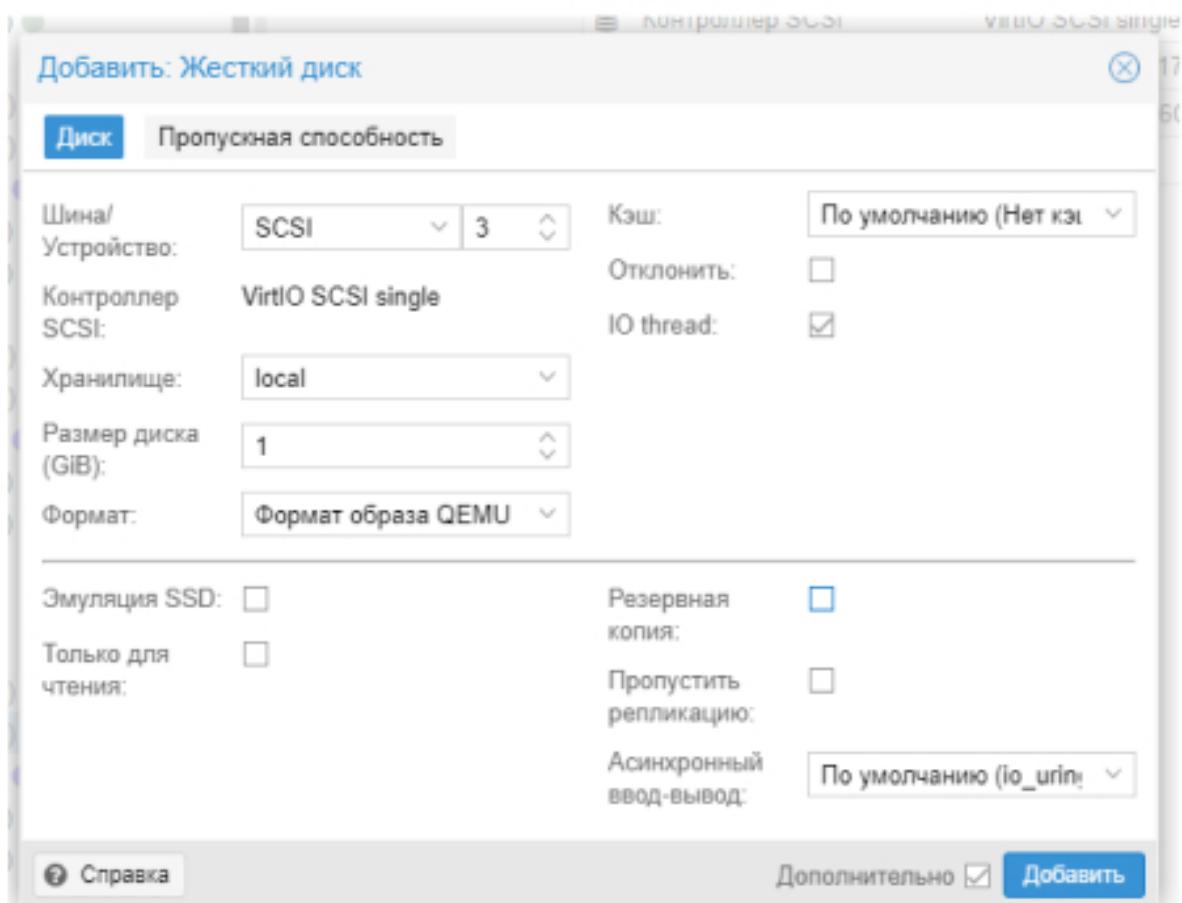
Виртуальная машина 17303 (HQ-SRV) на узле Server3 alt_server_10.1 Запуск

Сводка	Добавить
Консоль	Память
Оборудование	Процессоры
Cloud-Init	BIOS
Параметры	Экран
Журнал задач	Машина
Монитор	Контроллер SCSI
Резервная копия	Жесткий диск (scsi0)
Репликация	Сетевое устройство (net0)
	Последовательный порт

Нажать кнопку Добавить и выбрать жесткий диск. Указываем Размер 1 Гб.

Виртуальная машина 17303 (HQ-SRV) на

Сводка	Добавить
Консоль	Память
Оборудование	Процессоры
Cloud-Init	BIOS
Параметры	Экран
Журнал задач	Машина
Монитор	Контроллер
Резервная копия	Жесткий диск
Репликация	Сетевое устройство
Снимки	Последовательный порт
Сетевой экран	
Разрешения	



Повторяем еще 2 раза. Должно по итогу быть так

Контроллер SCSI	VirtIO SCSI single
Жесткий диск (scsi0)	local:17303/vm-17303-disk-0.qcow2,size=10G
Жесткий диск (scsi1)	local:17303/vm-17303-disk-2.qcow2,backup=0,iothread=1,size=1G
Жесткий диск (scsi2)	local:17303/vm-17303-disk-3.qcow2,backup=0,iothread=1,size=1G
Жесткий диск (scsi3)	local:17303/vm-17303-disk-1.qcow2,backup=0,iothread=1,size=1G
Сетевое устройство (net0)	virtio=9E:15:D3:6C:9A:29,bridge=vmbr1027,tag=100

Создание RAID

Просматриваем имена добавленных дисков:

```
[root@hq-srv ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda    8:0     0 10G  0 disk /
sdb    8:16    0  1G  0 disk
sdc    8:32    0  1G  0 disk
sdd    8:48    0  1G  0 disk
```

Обнуляем суперблоки для добавленных дисков:

```
[root@hq-srv ~]# mdadm --zero-superblock --force /dev/sd{b,c,d}
```

```
[root@hq-srv ~]# mdadm --zero-superblock --force /dev/sd{b,c,d}
mdadm: Unrecognised md component device - /dev/sdb
mdadm: Unrecognised md component device - /dev/sdc
mdadm: Unrecognised md component device - /dev/sdd
```

Удаляем старые метаданные и подпись на дисках:

```
[root@hq-srv ~]# wipefs --all --force /dev/sd{b,c,d}
```

Создаем RAID:

```
mdadm --create /dev/md0 -l 5 -n 3 /dev/sd{b,c,d}
```

или

```
mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sd{b-d}
```

```
[root@hq-srv ~]# mdadm --create /dev/md0 -l 5 -n 3 /dev/sd{b,c,d}
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

/dev/md0 - название RAID после сборки

-l 5 - уровень RAID

-n 3 - количество дисков, из которых собирается массив

/dev/sd{b,c,d} - диски, из которых выполняется сборка

Проверяем:

```
[root@hq-srv ~]# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda    8:0     0   10G  0 disk  /
sdb    8:16    0   1G   0 disk
└─md0   9:0     0   2G   0 raid5
sdc    8:32    0   1G   0 disk
└─md0   9:0     0   2G   0 raid5
sdd    8:48    0   1G   0 disk
└─md0   9:0     0   2G   0 raid5
```

Сохраним конфигурацию массива в файл **/etc/mdadm.conf** следующей командой:

```
[root@hq-srv ~]# mdadm --detail --scan --verbose >> /etc/mdadm.conf
[root@hq-srv ~]# cat /etc/mdadm.conf
ARRAY /dev/md0 level=raid5 num-devices=3 metadata=1.2 name=hq-srv.au-team.ip0:0 UUID=
b7c36012:32073fec:4d1d5e03:b40a85c0
    devices=/dev/sdb,/dev/sdc,/dev/sdd
```

Создаем файловую систему из созданного **RAID**:

```
[root@hq-srv ~]# mkfs -t ext4 /dev/md0
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 523264 4k blocks and 130816 inodes
Filesystem UUID: 8e5b33e4-50a1-41d1-8a49-e44ad3f4ab42
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Монтируем RAID-массива:

Создаем директорию для монтирования массива:

```
[root@hq-srv ~]# mkdir /mnt/raid5
```

Настроим автоматическое монтирование в /mnt/raid5. Добавляем следующую строку в конец файла /etc/fstab (можно на основе UUID):

```
[root@hq-srv ~]# cat /etc/fstab
#
# /etc/fstab: static file system information
#
#  
proc          /proc           proc    nosuid,noexec,gid=proc      0  0
/devpts       /dev/pts        devpts  nosuid,noexec,gid=tty,mode=620  0  0
tmpfs         /tmp            tmpfs   nosuid                   0  0
UUID=5cfdfaf0-839f-4578-ad0f-b52b0336dfe2  /      ext4    relatime   1      1
/dev/sr0       /media/AltLinux udf,iso9660  ro,noauto,user,utf8,nofail,comment=x-gvfs-show  0  0
/dev/md0       /mnt/raid5      ext4    defaults      0  0
-
```

Монтируем:

```
[root@hq-srv ~]# mount -a
```

Проверяем монтирование:

```
[root@hq-srv ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
udevfs          5.0M   64K  5.0M   2% /dev
runfs           991M  592K  990M   1% /run
/dev/sda        9.8G  3.0G  6.4G  32% /
tmpfs           991M     0  991M   0% /dev/shm
tmpfs           991M     0  991M   0% /tmp
tmpfs           199M     0  199M   0% /run/user/0
/dev/md0        2.0G   24K  1.9G   1% /mnt/raid5

```

Настройка NFS

Устанавливаем пакеты для NFS-сервера:

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

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

Создаем директорию для общего доступа:

```
[root@hq-srv ~]# mkdir /mnt/raid5/nfs
```

Выдаем права на чтение и запись этой директории:

```
[root@hq-srv ~]# chmod 766 /mnt/raid5/nfs
```

Откроем каталог для общего доступа в сторону подсети, где находится HQ-CLI, для этого заходим в /etc/exports и пишем следующую строку в конец файла:

/mnt/raid5/nfs 192.168.200.0/28(rw,no_root_squash)

/mnt/raid5/nfs - общий ресурс

192.168.100.64/28 - клиентская сеть, которой разрешено монтирование общего ресурса

rw — разрешены чтение и запись

no_root_squash — отключение ограничения прав **root**

```
#/srv/public -ro,insecure,no_subtree_check,fsid=1 *
#/srv/share -rw,insecure,fsid=0,sec=krb5 *

/mnt/raid5/nfs 192.168.100.64/28 (rw,no_root_squash)
~
```

Экспортируем файловую систему, которую прописали ранее:

exportfs -arv

-a - экспортировать все указанные каталоги

-r - повторный экспорт всех каталогов,

синхронизируя **/var/lib/nfs/etab** с **/etc/exports** и файлами в **/etc/exports.d**

-v - подробный вывод

```
[root@hq-srv ~]# exportfs -arv
exportfs: /etc/exports [2]: Neither 'subtree_check' or 'no_subtree_check' specified for export "192.168.100.64/28:/mnt/raid5/nfs".
Assuming default behaviour ('no_subtree_check').
NOTE: this default has changed since nfs-utils version 1.0.x
exporting 192.168.100.64/28:/mnt/raid5/nfs
```

Или просто

```
[root@hq-srv ~]# exportfs
/mnt/raid5/nfs 192.168.100.64/28
```

Запускаем и добавляем в автозагрузку NFS-сервер:

```
[root@hq-srv ~]# systemctl enable --now nfs-server
```

Настройка клиента

Устанавливаем требуемые пакеты для **NFS-клиента**:

```
hq-cli ~ # apt-get install nfs-clients
```

(может быть уже установлен)

Создаем директорию для общего ресурса:

```
hq-cli ~ # mkdir /mnt/nfs
```

Выдаем права этой директории:

```
hq-cli ~ # chmod 777 /mnt/nfs/
```

Добавляем строку в `/etc/fstab` для автоматического монтирования общего ресурса:

```
File Edit View Search Terminal Help
proc      /proc          proc    nosuid,noexec,gid=proc    0 0
devpts    /dev/pts        devpts  nosuid,noexec,gid=tty,mode=620  0 0
tmpfs     /tmp           tmpfs   nosuid                0 0
UUID=34b080a4-0a2f-4f93-8c11-65391012d197  /   ext4    relatime    1   1
/dev/sr0   /media/ALTLinux udf,iso9660  ro,noauto,user,utf8,nofail,comment=x-gvfs-show  0 0
192.168.100.62:/mnt/raid5/nfs  /mnt/nfs      nfs      default 0 0
```

Монтируем общий ресурс:

```
hq-cli ~ # mount -a
```

Проверяем монтирование:

```
hq-cli ~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
udevfs          5.0M  64K  5.0M  2% /dev
runfs           1.5G  1.1M  1.5G  1% /run
/dev/sda         12G  8.3G  2.9G  75% /
tmpfs            1.5G    0  1.5G  0% /dev/shm
tmpfs            1.5G  8.0K  1.5G  1% /tmp
tmpfs            299M  80K  299M  1% /run/user/500
tmpfs            299M  56K  299M  1% /run/user/0
192.168.100.62:/mnt/raid5/nfs  2.0G    0  1.9G  0% /mnt/nfs
```

Проверим:

Создадим файл с клиентской машине в каталоге `/mnt/nfs`, затем посмотрим на сервере, создался ли он:

```
hq-cli ~ # touch /mnt/nfs/test
```

```
[root@hq-srv ~]# ls -la /mnt/raid5/nfs/
total 8
drwxr-xw-rw- 2 root root 4096 Apr  9 12:13 .
drwxr-xr-x  4 root root 4096 Apr  9 11:39 ..
-rw-r--r--  1 root root     0 Apr  9 12:13 test
```

3. Настройте службу сетевого времени на базе сервиса chrony

- В качестве сервера выступает HQ-RTR
- На HQ-RTR (ISP) настройте сервер chrony, выберите стратум 5
- В качестве клиентов настройте HQ-SRV, HQ-CLI, BR-RTR, BR-SRV

Так как на HQ-RTR нет утилиты chrony и возможность выбора стратума, NTP-сервером будет выступать ISP

Конфигурация NTP-сервера (ISP)

Скачиваем пакет chrony:

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

Приводим начало файла /etc/chrony.conf к следующему виду:

```
[root@isp ~]# vim /etc/chrony.conf

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (https://www.pool.ntp.org/join.html).
#pool pool.ntp.org iburst
server 127.0.0.1 iburst prefer
local stratum 5
allow 0/0
hwclock *  
[root@isp ~]#
```

server 127.0.0.1 - указываем сервером синхронизации самого себя

iburst - принудительно отправляет пакеты для точности синхронизации

prefer - отдает приоритет этому серверу

hwclock * - указывает сетевой интерфейс как собственный источник времени и синхронизирует клиентов с ним

local stratum 5 - указание иерархического уровня

allow 0/0 - разрешает подключение с любого IP-адреса

Запускаем и добавляем в автозагрузку и перезапускаем утилиту **chrony**:

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

Проверка конфигурации NTP-сервера

Получаем вывод источников времени с помощью команды:

```
[root@isp ~]# chronyc sources  
MS Name/IP address          Stratum Poll Reach LastRx Last sample  
-----  
^? localhost.localdomain      0       6    377     -      +0ns[+0ns] +/- 0ns
```

Получаем вывод уровня стратума с помощью связи команд:

```
[root@isp ~]# chronyc tracking | grep Stratum  
Stratum : 5
```

Конфигурация NTP-клиента на HQ-RTR

Указываем IP-адрес NTP-сервера:

```
hq-rtr(config)#ntp server 172.16.4.1.
```

Проверка конфигурации NTP-клиента на EcoRouter

Проверяем командой:

```
hq-rtr(config)#do show ntp status  
Status Description  
*   best  
+   sync  
-   failed  
?   unknown  
  
-----  
Status : VR name : Server : Stratum : Delay : Version : >  
* : default : 172.16.4.1 : 5 : 0.0422 : 4 : >
```

Сохраняем конфигурацию

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

Конфигурация NTP-клиента на BR-RTR

Аналогично настраиваем на BR-RTR (172.16.5.1):

Проверяем:

br-rtr(config)# do show ntp status						
Status	Description	Server	Stratum	Delay	Version	
*	best	172.16.5.1	5	0.0379	4	

Конфигурация NTP-клиента Alt Linux (HQ-SRV, HQ-CLI, BR-SRV)

Скачиваем пакет **chrony**:

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

Приводим начало файла /etc/chrony.conf к следующему виду:

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



```
#pool pool.ntp.org iburst
#
server 172.16.4.1 iburst prefer
#
# Record the rate at which the system clock gains/loses
```

Запускаем утилиту **chrony** и добавляем ее в автозагрузку:

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

```
[root@hq-srv ~]# systemctl enable --now chronyd
```

Проверяем:

```
[root@hq-srv ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* 172.16.4.1                5      6     77    53    +297us(+5438us) +/- 6349us
```

Аналогично на **HQ-CLI, BR-SRV**, указывая адрес ближайшего интерфейса к NTP серверу (для HQ-CLI – 172.16.4.1 для BR-SRV – 172.16.5.1).

Проверяем:

Hostname	NTP	Drop	Int	IntL	Last	Cnd	Drop	Int	Last
localhost.localdomain	18	0	7	-	116	0	0	-	-
172.16.4.2	20	0	6	-	30	0	0	-	-
172.16.5.2	23	0	6	-	54	0	0	-	-

Получаем устройства, которые патят дальше в свои сети.

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

Конфигурация SSH Alt Linux (HQ-SRV, HQ-CLI)

Затронутые строки в конфигурационном файле **SSH /etc/openssh/sshd_config** должны выглядеть следующим образом:

```
Port 2024
MaxAuthTries 2
PubkeyAuthentication yes
PasswordAuthentication yes
Banner /etc/openssh/bannermotd
AllowUsers sshuser (знак TAB)
```

(Первоначальная настройка SSH производилась в задании Настройка безопасного удаленного доступа на серверах HQ-SRV и BR-SRV из Модуля 1. На HQ-CLI в Модуле 1 не требовалась настройка SSH и пользователя sshuser, сейчас необходимо это сделать)

Перезагружаем службу sshd

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

Конфигурация Ansible

Устанавливаем необходимые пакеты на BR-SRV:

```
[root@br-srv ~]# apt-get install -y ansible zshpass
```

Назначем необходимые права на директорию **/etc/ansible**:

```
[root@br-srv ~]# chown -R root:user /etc/ansible  
[root@br-srv ~]# chmod -R 774 /etc/ansible
```

Из под обычного пользователя **user** переходим для дальнейшей работы в директорию **/etc/ansible**:

```
[root@br-srv ~]# cd /etc/ansible
```

Проверяем, где мы и что есть в этой директории:

```
[root@br-srv ~]# cd /etc/ansible  
[root@br-srv ansible]# ls  
ansible.cfg  hosts
```

Редактируем указанные строки в **конфигурационном файле /etc/ansible/ansible.cfg**:

```
inventory = ./inventory.yml  
host_key_checking = False
```

```
# some basic default values...  
  
inventory      = /etc/ansible/inventory.yml  
library        = /usr/share/my_modules/  
...  
  
# uncomment this to disable SSH key host checking  
host_key_checking = False
```

inventory = ./inventory.yml - путь до инвентарного файла
host_key_checking = False - отключение проверки ключа хоста

Создаём инвентарный файл **/etc/ansible/inventory.yml**:

```
all:
  children:
    Networking:
      hosts:
        br-rtr:
        hq-rtr:
    Servers:
      hosts:
        hq-srv:
          ansible_host: 192.168.100.62
          ansible_port: 2024
    Clients:
      hosts:
        hq-cli:
          ansible_host: 192.168.200.2
          ansible_port: 2024
```

Из под обычного пользователя user переходим для дальнейшей работы в директорию /etc/ansible:

Создаем файлы с переменными для **всех категорий** и для категории **Networking**:

```
[user@br-srv ansible]$ mkdir group_vars
[user@br-srv ansible]$ touch group_vars/{all.yml,Networking.yml}
[user@br-srv ansible]$ cd group_vars/
[user@br-srv group_vars]$ _
```

(создали 2 файла с именами all.yml и Networking.yml)

Редактируем их:

```
[root@br-srv group_vars]$ vim all.yml

ansible_ssh_user: sshuser
ansible_ssh_pass: P@ssw0rd
ansible_python_interpreter: /usr/bin/python3

[user@br-srv group_vars]$ vim Networking.yml

ansible_connection: network_cli
ansible_network_os: ios
```

Выполняем команду для ping`а всех машин:

```
[user@br-srv group_vars]$ ansible -m ping all
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"
}
```

5. Развёртывание 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
- Разверните
- Он должен создавать базу с названием mediawiki, доступную по стандартному порту, пользователю wiki с паролем WikiP@sswOrd должен иметь права доступа к этой базе данных
- MediaWiki должна быть доступна извне через порт 8080

Конфигурация файла Docker-Compose

Останавливаем службу **ahttpd**, которая занимает порт **8080** (если есть):

```
systemctl disable --now ahttpd
```

Устанавливаем docker и docker-compose:

```
[root@br-srv ~]# apt-get install docker-ce docker-compose -y
```

Включаем и добавляем в автозагрузку **docker**:

```
[root@br-srv ~]# systemctl enable --now docker
```

Создаем в домашней директории пользователя файл, в качестве пользователя, которого мы создавали при установке ОС, у нас – user, а его домашний каталог – /home/user, файл называется – **wiki.yml**, для приложения MediaWiki:

```
[root@br-srv ~]# vim /home/user/wiki.yml
```

И заполняем его следующими строками, обратите внимание, что в строках ПРОБЕЛЫ, А НЕ ТАБУЛЯЦИЯ:

```
services:  
mediawiki:  
  container_name: wiki  
  image: mediawiki  
  restart: always  
  ports:  
    - "8080:80"  
  volumes:  
    - ./LocalSettings.php:/var/www/html/LocalSettings.php  
depends_on:  
  - mariadb  
mariadb:  
  image: mariadb  
  container_name: mariadb  
  restart: always  
  environment:  
    MYSQL_DATABASE: mediawiki  
    MYSQL_USER: wiki  
    MYSQL_PASSWORD: P@ssw0rd  
    MYSQL_ROOT_PASSWORD: P@ssw0rd  
  volumes:  
    - mariadb_data:/var/lib/mysql  
  
volumes:  
  mariadb_data:
```

```

services:
  mediawiki:
    container_name: wiki
    image: mediawiki
    restart: always
    ports:
      - "8080:80"
    # volumes:
    #   - ./LocalSettings.php:/var/www/html/LocalSettings.php
    depends_on:
      - mariadb

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

volumes:
  - mariadb_data:

```

services - основной раздел, в котором описываются сервисы

container_name - имя контейнера

image - имя образа

restart - перезапуск контейнера, если он остановлен

ports - проброс портов

links - ссылка на контейнер

volumes - проброс папок

environment - переменные окружения

Собираем стек контейнеров:

```
[root@br-srv user]# docker compose -f wiki.yml up -d
```

-f - указание на файл

up - запуск

-d - запуск в фоновом режиме

```
[root@br-srv user]# docker compose -f wiki.yml up -d
[+] Running 5/2
  mariadb [          ] Pulling
  wiki  [██████.███] Pulling
```

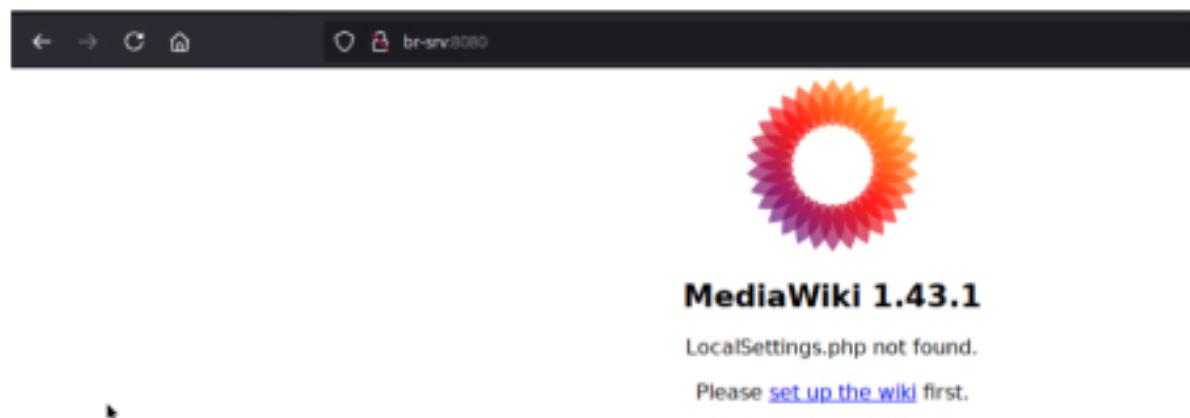
Проверим запущенные контейнеры:

```
[root@br-srv user]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
75eff2197050        mediawiki          "docker-php-entrypoi..."   6 minutes ago     Up 6 minutes      0.0.0.0:8080->80/tcp, ::1:8080->80/tcp   wiki
e264c2bb82365       mariadb            "docker-entrypoint.s..."   6 minutes ago     Up 6 minutes      3306/tcp           mariadb
```

Установка MediaWiki в веб-интерфейсе

На HQ-CLI в браузере вводим <http://192.168.3.30:8080> и начинаем установку **MediaWiki**.

Если хотим, чтобы страница открывалась по адресу <http://br-srv:8080>, то добавляем устройства в dns в samba: samba-tool computer add br-srv --ip-address=192.168.3.30



Выбираем язык:

Your language:

help

ru - русский

Wiki language:

help

ru - русский

Continue →

Проверяем внешнюю среду и нажимаем далее:

✓ Проверка внешней среды была успешно проведена.

Вы можете установить MediaWiki.

Заполняем параметры для базы данных в соответствии с заданными переменными окружения в `wiki.yml`:

Настройки MariaDB/MySQL

Хост базы данных:

[справка](#)

mariadb

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](#)

*****|

Оставляем галочку и жмем далее:

Установка MediaWiki 1.43.1

Настройки базы данных

Учётная запись для доступа к базе данных из веб-сервера

Использовать ту же учётную запись, что и для установки

[← Назад](#)

[Далее →](#)

Заполняем информацию об учетной записи администратора:

Установка MediaWiki 1.43.1

Название

Название вики:

справка

wiki

Пространство имён проекта:

справка

То же, что имя вики: Wiki

Проект

Другое (укажите)

Учётная запись администратора

Ваше имя участника:

справка

admin

Пароль:

Пароль ещё раз:

Адрес электронной почты:

справка

admin@au-team.irpo

Подписаться на [рассылку новостей о появлении новых версий MediaWiki](#).

справка

Учётная запись администратора

Ваше имя участника:

[справка](#)

admin

Пароль:

Пароль ещё раз:

Адрес электронной почты:

[справка](#)

admin@au-team.irpo

Подписаться на [рассылку новостей о появлении новых версий MediaWiki](#).

[справка](#)



Поделиться сведениями об этой установке с разработчиками [Политика конфиденциальности](#).

[справка](#)

❶ Информация

Вы почти у цели! Остальные настройки можно пропустить и приступить к установке вики.



Произвести тонкую настройку



Хватит уже, просто установите вики.

[← Назад](#)

[Далее →](#)

Установка MediaWiki 1.43.1

Установка

❶ Информация

Нажав «Далее →», вы начнёте установку MediaWiki. Если вы хотите внести изменения, нажмите «← Назад».

[← Назад](#)

[Далее →](#)

Установка MediaWiki 1.43.1

Установка

- Настройка базы данных... выполнено
- Создание таблиц, первый шаг... выполнено
- Создание базы данных пользователей... выполнено
- Заполнение таблицы интервики значениями по умолчанию... выполнено
- Статистика инициализации... выполнено
- Создание секретных ключей... выполнено
- Предотвращение запуска ненужных обновлений... выполнено
- Восстановление сервисов MediaWiki... выполнено
- Создание учётной записи администратора... выполнено
- Создание главной страницы с содержимым по умолчанию... выполнено

База данных была успешно настроена

[Далее →](#)

Получаем конфигурационный файл, который нужно передать на BR-SRV:

Готово!

Поздравляем!

Вы установили MediaWiki.

Во время установки был создан файл LocalSettings.php. Он содержит все ваши настройки.

Вам необходимо скачать его и положить в корневую директорию вашей вики (ту же директорию, где находится файл index.php). Его загрузка должна начаться автоматически.

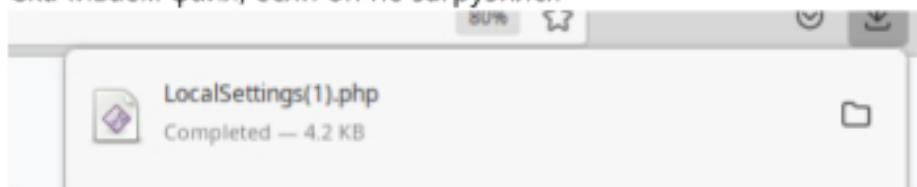
Если автоматическая загрузка не началась или вы её отменили, вы можете скачать по ссылке ниже:

[Загрузить LocalSettings.php](#)

Примечание: Если вы не сделаете этого сейчас, то созданный файл конфигурации не будет доступен вам в дальнейшем, если вы выйдете из установки, не скачивая его.

По окончании действий, описанных выше, вы сможете [войти в вашу вики](#).

Скачиваем файл, если он не загрузился



Правка файла Docker-Compose

Перемещаем файл **LocalSettings.php** в домашнюю директорию пользователя **sshuser**:

```
mv /home/user/Загрузки/LocalSettings.php /home/sshuser
```

В моем случае, ранние действия выполнялись из под пользователя **user**, поэтому загруженный файл оказался именно в его папке

```
hq-cli ~ # mv /home/user/Downloads/LocalSettings.php /home/sshuser/
```

Передаем файл с **HQ-CLI** на **BR-SRV**:

```
hq-cli ~ # scp -P 2024 /home/sshuser/LocalSettings.php sshuser@192.168.3.30:/home/sshuser
The authenticity of host '[192.168.3.30]:2024 ([192.168.3.30]:2024)' can't be established.
ED25519 key fingerprint is SHA256:u0yBDzeG3L+p8QK2EayhE/w309VHUFDHMdDWZPs614k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.3.30]:2024' (ED25519) to the list of known hosts.
sshuser@192.168.3.30's password:
```

-P - указание порта SSH

/home/sshuser/LocalSettings.php - файл, который будет передан

sshuser@192.168.0.30:/home/sshuser - имя-пользователя@IP-адрес директория-назначения

Проверим доставку:

```
[root@br-srv user]# ls /home/sshuser/
LocalSettings.php
```

На **BR-SRV** перемещаем файл в домашнюю директорию **пользователя**:

```
mv /home/sshuser/LocalSettings.php /home/user
```

Если файл **wiki.yml** создавали в домашней директории другого пользователя - перемещаем туда

```
[root@br-srv user]# mv /home/sshuser/LocalSettings.php /home/user
```

В файле **wiki.yml** расскомментируем следующие строки:

volumes:

```
- ./LocalSettings.php;/var/www/html/LocalSettings.php
```

```
mediawiki:
  container_name: wiki
  image: mediawiki
  restart: always
  ports:
    - "8080:80"
  volumes:
    - ./LocalSettings.php:/var/www/html/LocalSettings.php
  depends_on:
    - mariadb
```

Перезапускаем запущенные Docker'ом сервисы:

```
[root@br-srv user]# docker compose -f wiki.yml stop
[+] Stopping 2/2
✓ Container wiki      Stopped
✓ Container mariadb   Stopped
```

```
[root@br-srv user]# docker compose -f wiki.yml up -d
[+] Running 2/2
✓ Container mariadb   Started
✓ Container wiki      Started
```

Проверим работу сайта, зайдем вновь через клиента HQ-CLI и увидим домашнюю страницу сайта:

The screenshot shows a web browser window with the URL `http://br-srv:8080/` in the address bar. The page title is "Заглавная страница". The sidebar on the left has "Содержание" and "Начало". The top navigation bar includes "Заглавная", "Обсуждение", "Читать", "Править", "История", and "Инст". Below the main content area, it says "MediaWiki успешно установлена." and "Информацию по работе с этой вики можно найти в [справочном руководстве](#)". At the bottom, there is a link "Начало работы [править]".

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
- 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

Конфигурация BR-RTR

Проброс портов с 80 на 8080 для работы сервиса **wiki** (192.168.3.30):

```
ip nat destination static tcp <через кого перенаправить (внешний адрес маршрутизатора)> <порт> <куда необходимо перенаправить> <порт> hairpin
```

```
ip nat destination static tcp 172.16.5.2 80 192.168.3.30 8080 hairpin
```

```
br-rtr(config)#ip nat destination static tcp 172.16.5.2 80 192.168.3.30 8080 hairpin
```

Проброс портов с 2024 на 2024 для **ssh**:

```
ip nat destination static tcp 172.16.5.2 2024 192.168.3.30 2024 hairpin
```

```
br-rtr(config)#ip nat destination static tcp 172.16.5.2 2024 192.168.3.30 2024 hairpin
```

Так же пропишем обратные правила (указываем адреса и порты в обратном порядке):

```
br-rtr(config)#ip nat source static tcp 192.168.3.30 8080 172.16.5.2 80
```

```
br-rtr(config)#ip nat source static tcp 192.168.3.30 2024 172.16.5.2 2024
```

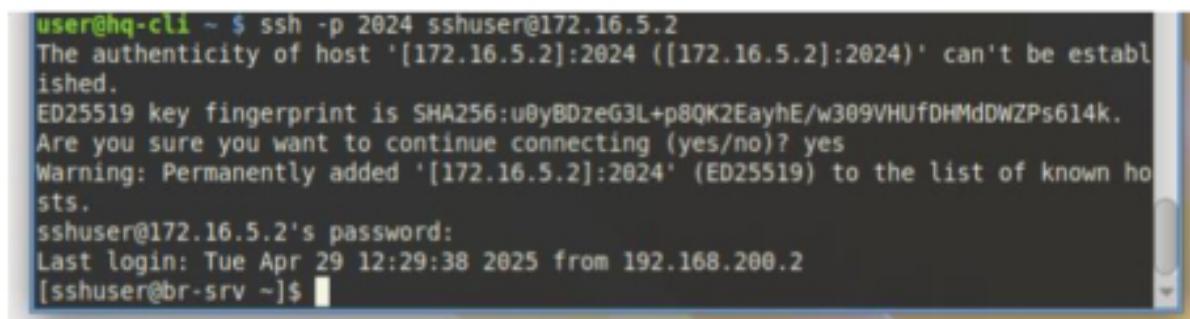
```
br-rtr(config)#write  
Building configuration...
```

Проверим:

```
br-rtr(config)#do show run
```

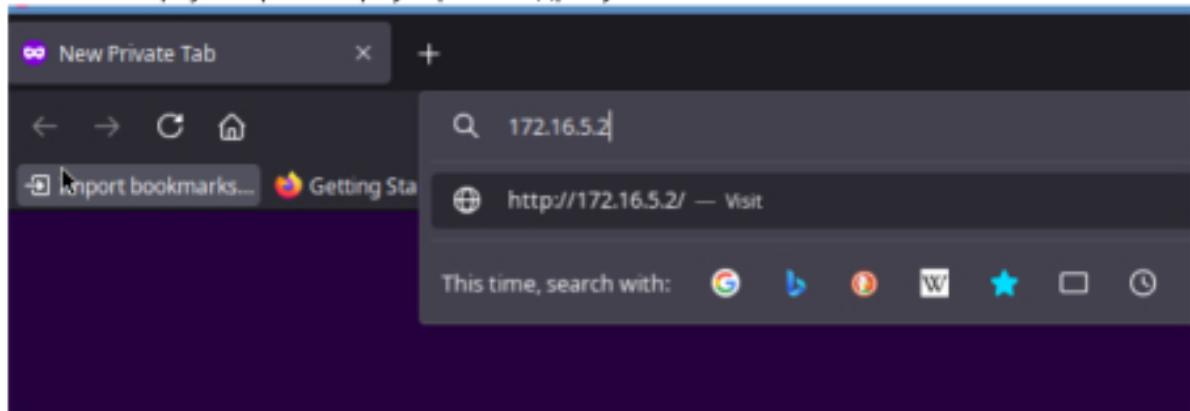
```
ip nat destination static tcp 172.16.5.2 80 192.168.3.30 8080 hairpin  
!  
ip nat destination static tcp 172.16.5.2 2024 192.168.3.30 2024 hairpin  
!  
ip nat source static tcp 192.168.3.30 8080 172.16.5.2 80  
!  
ip nat source static tcp 192.168.3.30 2024 172.16.5.2 2024  
!  
ip nat pool NAT_POOL 192.168.3.1-192.168.3.30  
!  
ip nat source dynamic inside-to-outside pool NAT_POOL overload interface int0
```

Проверим подключение с **HQ-CLI** по **ssh** к серверу **BR-SRV** через IP-адрес роутера **BR-RTR**:

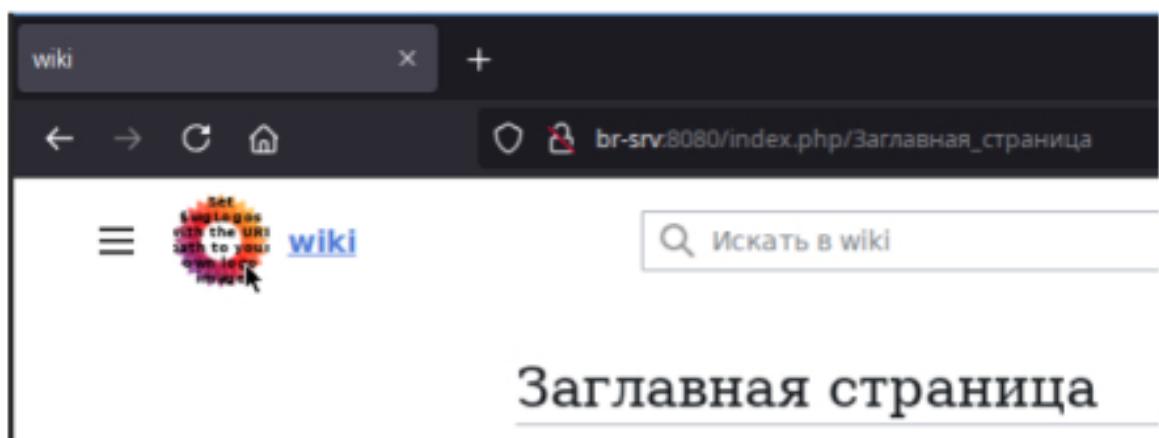


```
user@hq-cli ~ $ ssh -p 2024 sshuser@172.16.5.2
The authenticity of host '[172.16.5.2]:2024 ([172.16.5.2]:2024)' can't be established.
ED25519 key fingerprint is SHA256:u0yBDzeG3L+p8QK2EayhE/w309VHUFDHMdDWZPs614k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[172.16.5.2]:2024' (ED25519) to the list of known hosts.
sshuser@172.16.5.2's password:
Last login: Tue Apr 29 12:29:38 2025 from 192.168.200.2
[sshuser@br-srv ~]$
```

И по 80 порту запрос в браузере по адресу 172.16.5.2



Получаем перенаправление на BR-SRV



Конфигурация HQ-RTR

Проброс портов с 2024 на 2024:

```
ip nat destination source static tcp <через кого перенаправить (внешний адрес маршрутизатора)> <порт> <куда необходимо перенаправить> <порт> hairpin
```

Прописываем правила:

```
ip nat destination static tcp 172.16.4.2 2024 192.168.100.62 2024 hairpin
!
ip nat source static tcp 192.168.100.62 2024 172.16.4.2 2024
!
ip nat source static tcp 192.168.100.62 80 172.16.4.2 80
!
ip nat pool HQ 192.168.100.1-192.168.100.254
```

Проверим подключение с **HQ-CLI** по **ssh** к серверу **HQ-SRV** через IP-адрес роутера **HQ-RTR**:

```
ssh -p 2024 sshuser@172.16.4.2
```

```
hq-cli ~ # ssh sshuser@172.16.4.2 -p 2024
Authorized access only
sshuser@172.16.4.2's password:
Last login: Thu Jun  5 11:51:15 2025 from 192.168.100.1
[sshuser@hq-srv ~]$
```

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 отчеты

Трудоемкий способ (но, быстрый)

Конфигурация базы данных

Устанавливаем необходимые пакеты:

```
apt-get install -y moodle moodle-apache2 moodle-base moodle-local-mysql
phpMyAdmin
```

```
[root@hq-srv ~]# apt-get install -y moodle moodle-apache2 moodle-base moodle-local-mys
ql phpMyAdmin
```

Добавляем в **автозагрузку** базу данных:

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

Задаем пароль для пользователя root в базе данных:

```
[root@hq-srv ~]# mysqladmin password 'P@ssw0rd'
```

Идем в файл **/etc/httpd2/conf/include/Directory_moodle_default.conf** на веб-сервере и проверяем наличие строки **Require all granted** после **AllowOverride None**:

```
<IfModule authz_host_module>
    AllowOverride All
    Require all granted
</IfModule>
```

Если ее нет, то дописываем.

Идем в файл **/etc/php/8.2/apache2-mod_php/php.ini** и изменяем строку, отвечающую за количество входных переменных:

```
: How many GET/POST/COOKIE input variables may be accepted
: max_input_vars = 1000
```

```
: How many GET/POST/COOKIE input variables may be accepted
max_input_vars = 5000
```

Добавляем в автозагрузку веб-сервер:

```
[root@hq-srv ~]# systemctl enable --now httpd2
Synchronizing state of httpd2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable httpd2
Created symlink /etc/systemd/system/multi-user.target.wants/httpd2.service → /lib/systemd/system/httpd2.service.
```

Авторизуемся в **MySQL** и вводим ранее указанный пароль:

```
[root@hq-srv ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.6.21-MariaDB-10.6.21-0+deb10u1 (ALT p10)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```

Создаем пользователя для базы данных:

```
MariaDB [(none)]> create user 'moodle'@'localhost' identified by 'P@ssw0rd';
Query OK, 0 rows affected (0.002 sec)
```

Создаем базу данных:

```
MariaDB [(none)]> create database moodledb default character set utf8 collate utf8_unicode_ci;
Query OK, 1 row affected (0.003 sec)
```

Выдаем права пользователю на созданную базу данных:

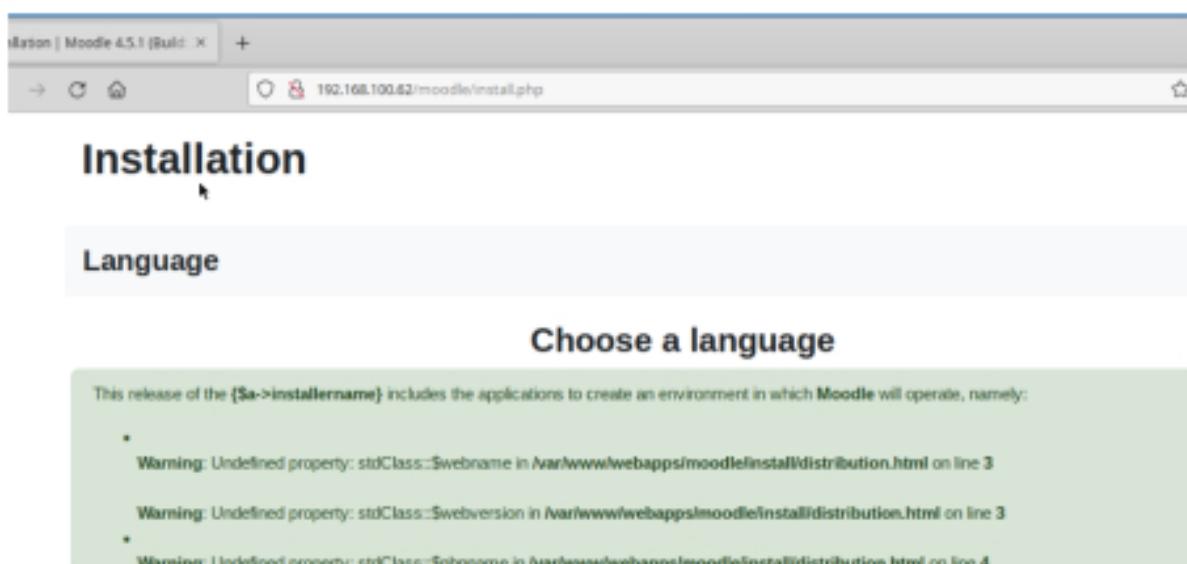
```
MariaDB [(none)]> grant all privileges on moodledb.* to moodle@localhost;
Query OK, 0 rows affected (0.002 sec)
```

Теперь подключаемся с клиента HQ-CLI и начинаем настройку:

<http://192.168.100.62/moodle/install.php>

Если хотим, чтобы страница открывалась по адресу `http://hq-srv.au-team.irpo/moodle/install.php`, то добавляем устройства в dns в samba: `samba-tool computer add hq-srv --ip-address=192.168.100.62`

Выбираем язык:



Язык

Русский (ru)

Далее »



Подтверждаем пути директорий:

Полный путь к каталогу установки moodle:

Каталог данных

Каталог, в котором Moodle будет хранить все файлы, размещаемые пользователями.

Этот каталог должен быть доступен для чтения и ЗАПИСИ тому пользователю, от чьего имени запускается веб-сервер (или 'apache').

Этот каталог не должен быть доступен напрямую через Интернет.

Программа установки попробует создать этот каталог, если он не существует.

Веб-адрес

Каталог Moodle

Каталог данных

« Назад

Далее »

»

Выбираем систему управления базы данных:

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

MariaDB («родной»/mariadb)

« Назад

Далее »



Заполняем данные о базе данных и пользователе:

! этот драйвер не совместим с устаревшей системой языка.

Сервер баз данных	localhost
Название базы данных	moodledb
Пользователь базы данных	moodle
Пароль	P@ssw0rd
Префикс имен таблиц	mdl_
Порт базы данных	
Подключение через Unix-сокет	

« Назад

Далее »

Соглашаемся с условиями:

https://

Подтвердить

Прочитали ли Вы эти условия и поняли их?

Отмена

Продолжить

Убеждаемся в успешной проверке:

php_extension	sodium	<input checked="" type="checkbox"/> необходимо установить и включить	?	OK
php_extension	exif	<input checked="" type="checkbox"/> рекомендуется установить и включить для наилучшей производительности	?	OK
php_setting	memory_limit	<input checked="" type="checkbox"/> обнаружены рекомендуемые настройки	?	OK
php_setting	file_uploads	<input checked="" type="checkbox"/> обнаружены рекомендуемые настройки	?	OK
php_setting	opcache.enable	<input checked="" type="checkbox"/> обнаружены рекомендуемые настройки	?	OK

Другие проверки

Информация Отчет

Плагин Статус

site not https непрогоаждение данного теста указывает на возможную проблему [?](#)
Обнаружено, что ваш сайт не защищен с помощью HTTPS. Настоятельно рекомендуется перенести ваш сайт на HTTPS для повышения безопасности и улучшения интеграции с другими системами.

Конфигурация сервера отвечает всем минимальным требованиям.

[Продолжить](#)

После установки настраиваем учетную запись администратора:

Основные

Логин



admin

Выберите метод аутентификации



Ручная регистрация

Пароль должен содержать символов - не менее 8, цифр - не менее 1, не менее 1 специальных символов, таких как *

Новый пароль



.....



Принудительная смена пароля [?](#)

Имя



admin

Фамилия



Пользователь

Адрес электронной почты



admin@au-team.irpo



Некорректный формат адреса электронной почты

Показывать адрес электронной почты



Всем



Заполняем в соответствии с условиями задания

Указываем название сайта, часовой пояс и электронную почту:

Новые настройки - Настройки главной страницы

Полное название сайта
13

Краткое название сайта (например, одним словом)
moodle

Описание главной страницы сайта
школа

Редактировать Вид Вставить Формат Инструменты Таблица Справка

Полное название сайта:	13 (согласно вашему рабочему месту)
Краткое название сайта:	moodle (можно любое)
Настройки местоположения:	Европа/Москва (согласно вашему региону)
Контакты службы поддержки:	admin@au-team.irpo (можно любое)

И жмём **Сохранить изменения** в конце страницы.

Новые настройки - Контакты службы поддержки

Электронная почта техподдержки
supportemail
admin@au-team.irpo

Если на этом сайте настроен SMTP, а страница темы не имеет почты, то приходить сообщения, отправленные через форму, будут отображаться вошедшим в систему г

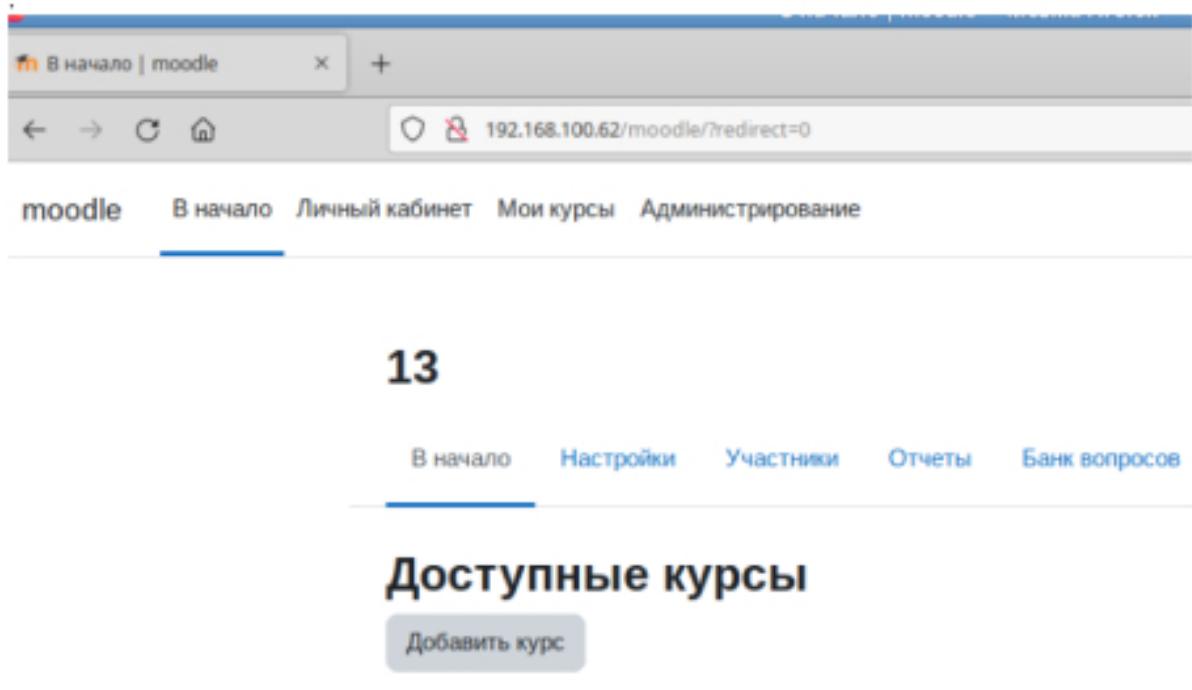
Новые настройки - Настройка исходящей почты

Адрес для писем, не требующих
ответа
noreplyaddress
admin@au-team.irpo

Иногда сообщения электронной почты отсылаются с помощью скриптов (например, сообщения с форума). Указанный здесь адрес электронной почты будет использоваться в том случае, если получатели не должны иметь возможность видеть адрес отправителя (если пользователь не хочет показывать свой адрес).

Сохранить изменения

И после всего нас встречает рабочий сайт **moodle**, смотрим, что все наши указанные параметры отображаются:



Установим службу deploy

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

Через deploy развернем moodle

```
[root@hq-srv ~]# deploy moodle
```

Установим первоначальный пароль – более простой (из задания) мудл не даст поставить из-за требований. Изменим требования после.

```
[root@hq-srv ~]# deploy moodle password=P@ssw0rd12345
```

Отключим шифрование ssl (чтоб работал http, а не https)

```
[root@hq-srv ~]# a2dismod ssl
```

Изменим в файле конфигурацию https – удалим букву «s»

```
[root@hq-srv ~]# vim /var/www/webapps/moodle/config.php
```

```
$CFG->wwwroot = 'http://hq-srv.au-team.irpo/moodle';
$CFG->dataroot = '/var/lib/moodle/default';
$CFG->admin = 'admin';
```

Изменим требования к паролю в moodle

```
[root@hq-srv ~]# vim /var/www/webapps/moodle/lib/moodlelib.php
```

Было:

```
// Password policy constants.
define ('PASSWORD_LOWER', 'abcdefghijklmnopqrstuvwxyz');
define ('PASSWORD_UPPER', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ');
define ('PASSWORD_DIGITS', '0123456789');
define ('PASSWORD_NONALPHANUM', '@#$%^&*():"');
```

Стало:

```
// Password policy constants.
define ('PASSWORD_LOWER', '');
define ('PASSWORD_UPPER', '');
define ('PASSWORD_DIGITS', '0123456789');
define ('PASSWORD_NONALPHANUM', ''');
```

Далее отключим редирект с http на https.

```
[root@hq-srv ~]# vim /etc/httpd2/conf/sites-enabled/000-default.conf
```

и закомментируем строки

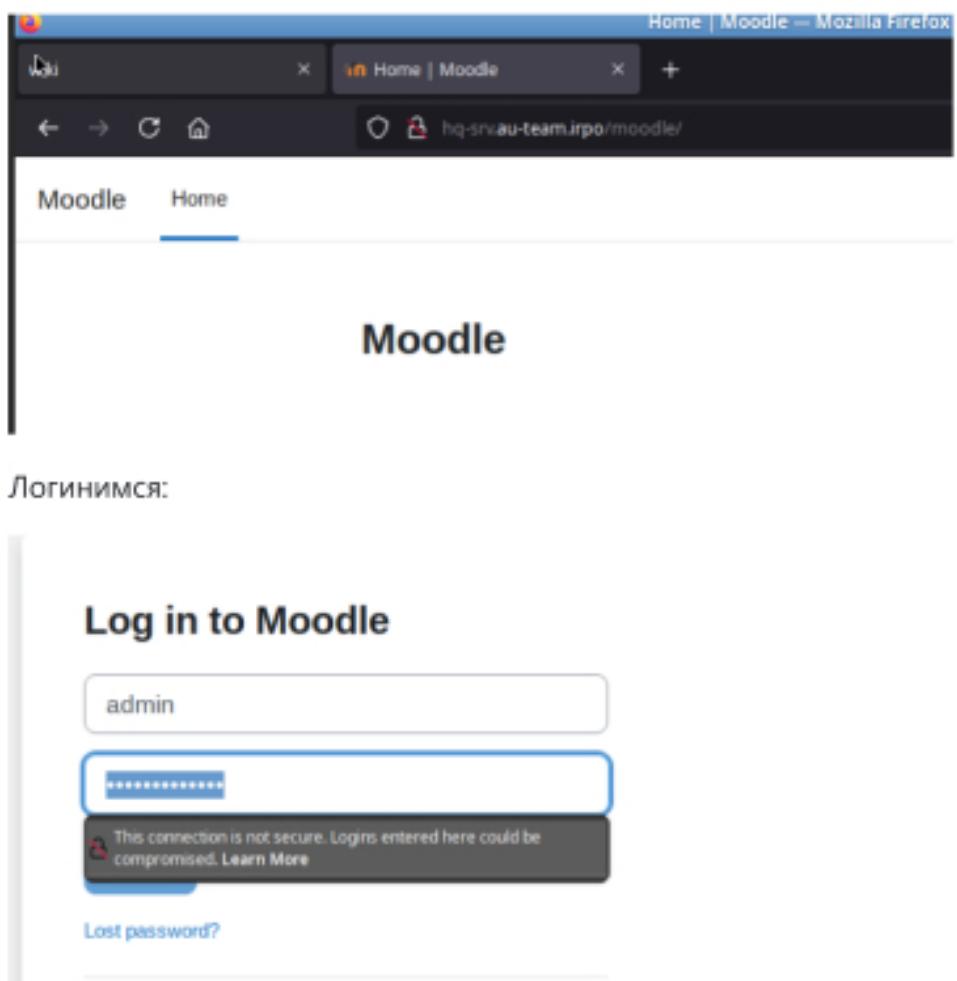
```
#RewriteEngine On
#RewriteCond %{HTTPS} !=on
#RewriteRule ^/(.*) https://[%{HTTP_HOST}]/$1 [R,L]
</VirtualHost>
```

Перезапустим веб-службу Apache

```
[root@hq-srv ~]# systemctl restart httpd2.service
```

Идем на клиента и в браузере вводим адрес.

Если хотим, чтобы страница открывалась по адресу `http://hq-srv.au-team.irpo/moodle`, то добавляем устройства в dns в samba: `samba-tool computer add hq-srv --ip-address=192.168.100.62`



Далее настраиваем как описано в варианте выше.

8. Настройте веб-сервер nginx как обратный прокси-сервер на HQ-RTR (так как обратный прокси не выполнимо на HQ-RTR, то делаем на на ISP)

- 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

Посмотреть, как можно на <https://www.dmosk.ru/miniiinstruktions.php?mini=nginx-redirects#proxypass-othersite>

На ISP установим nginx

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

Далее, для настройки nginx как реверсивного прокси сервера, дописать в файл /etc/nginx/nginx.conf следующую конфигурацию:

```
http {
    server {
        listen 80; # Слушаем на 80 порту для HTTP
        server_name moodle.au-team.irpo; # Указываем первое доменное имя
        location / {
            proxy_pass http://172.16.4.2:80; # Перенаправление на указанный адрес и порт
            proxy_set_header Host $host; # Пробрасываем заголовок Host
            proxy_set_header X-Real-IP $remote_addr; # Пробрасываем IP клиента
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; #Пробрасываем заголовок X-Forwarded-For
            proxy_set_header X-Forwarded-Proto $scheme; #Пробрасываем схему запроса
        }
    }
    server {
        listen 80; # Слушаем на 80 порту для HTTP
        server_name wiki.au-team.irpo; # Указываем второе доменное имя
        location / {
            proxy_pass http://172.16.5.2:80; # Перенаправление на указанный адрес и порт
            proxy_set_header Host $host; # Пробрасываем заголовок Host
            proxy_set_header X-Real-IP $remote_addr; # Пробрасываем IP клиента
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; #Пробрасываем заголовок X-Forwarded-For
            proxy_set_header X-Forwarded-Proto $scheme; # Пробрасываем схему запроса
        }
    }
}
```

```

moodle include /etc/nginx/mime.types;
default_type application/octet-stream;

sendfile on;

gzip on;
# text/html doesn't need to be defined there, it's compressed always
gzip_types text/plain text/css text/xml application/x-javascript application/atom+xml;

# gzip comp_level 9;
include /etc/nginx/sites-enabled.d/*.conf;

server {
    server_name moodle.au-team.irpo;
    location / {
        proxy_pass http://172.16.4.2:80/;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

server {
    server_name wiki.au-team.irpo;
    location / {
        proxy_pass http://172.16.5.2:80/;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

Добавим nginx в автозагрузку

```

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

```

Добавим записи DNS для moodle и wiki в samba:

```

[root@isp ~]# nmba-tool computer add moodle --ip-address=172.36.4.1
Password for [Administrator@WU-TB01 IRPO]:
[PRIV] 2025-06-05 12:56:33,052 pid:33321 user:11664/samba-dc/python3.8/samba/netbios/computer.py:1036: Adding DNS A record moodle.au-team.irpo for IPv4 IP: 172.36.4.1
Computer 'moodle' added successfully
[root@isp ~]# nmba-tool computer add wiki --ip-address=172.16.5.1
Password for [Administrator@WU-TB01 IRPO]:
[PRIV] 2025-06-05 12:56:57,469 pid:33324 user:11664/samba-dc/python3.8/samba/netbios/computer.py:1036: Adding DNS A record wiki.au-team.irpo for IPv4 IP: 172.16.5.1
Computer 'wiki' added successfully
[root@isp ~]#

```

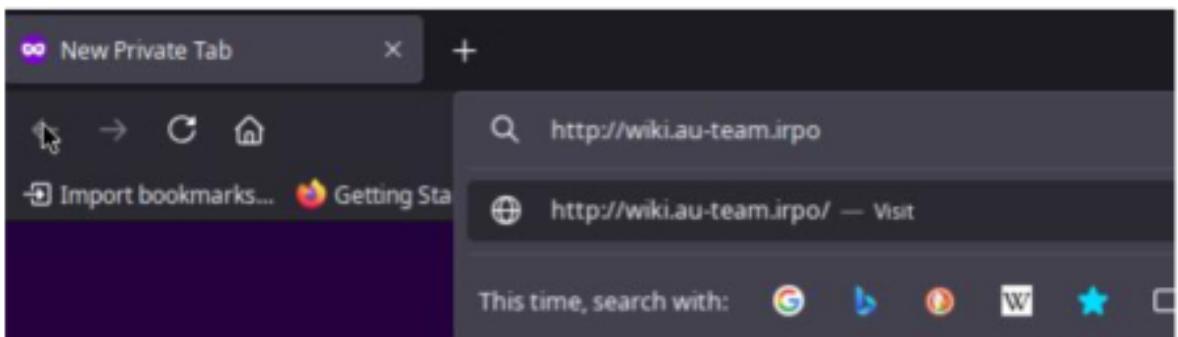
Проверим:

```
hg-cli ~ # ping wiki.au-team.irpo
PING wiki.au-team.irpo (172.16.5.1) 56(84) bytes of data.
64 bytes from 172.16.5.1 (172.16.5.1): icmp_seq=1 ttl=63 time=65.0 ms
64 bytes from 172.16.5.1 (172.16.5.1): icmp_seq=2 ttl=63 time=61.5 ms
64 bytes from 172.16.5.1 (172.16.5.1): icmp_seq=3 ttl=63 time=61.1 ms
64 bytes from 172.16.5.1 (172.16.5.1): icmp_seq=4 ttl=63 time=60.7 ms

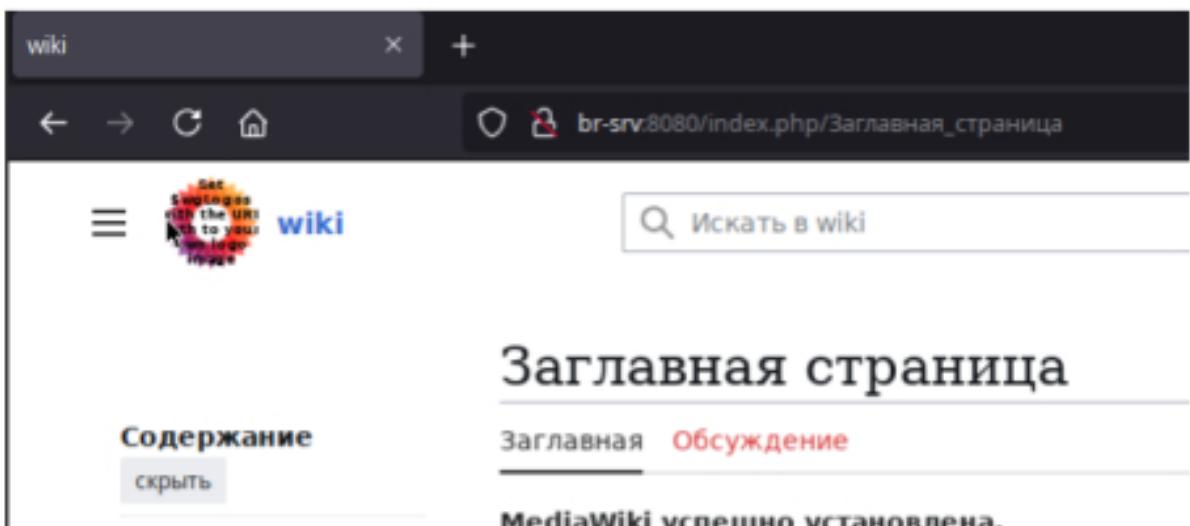
--- wiki.au-team.irpo ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 60.654/62.074/65.016/1.725 ms
```

```
hg-cli ~ # ping moodle.au-team.irpo
PING moodle.au-team.irpo (172.16.4.1) 56(84) bytes of data.
64 bytes from 172.16.4.1 (172.16.4.1): icmp_seq=1 ttl=63 time=12.9 ms
64 bytes from 172.16.4.1 (172.16.4.1): icmp_seq=2 ttl=63 time=12.6 ms
64 bytes from 172.16.4.1 (172.16.4.1): icmp_seq=3 ttl=63 time=13.0 ms
^C
--- moodle.au-team.irpo ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 12.564/12.803/12.973/0.174 ms
```

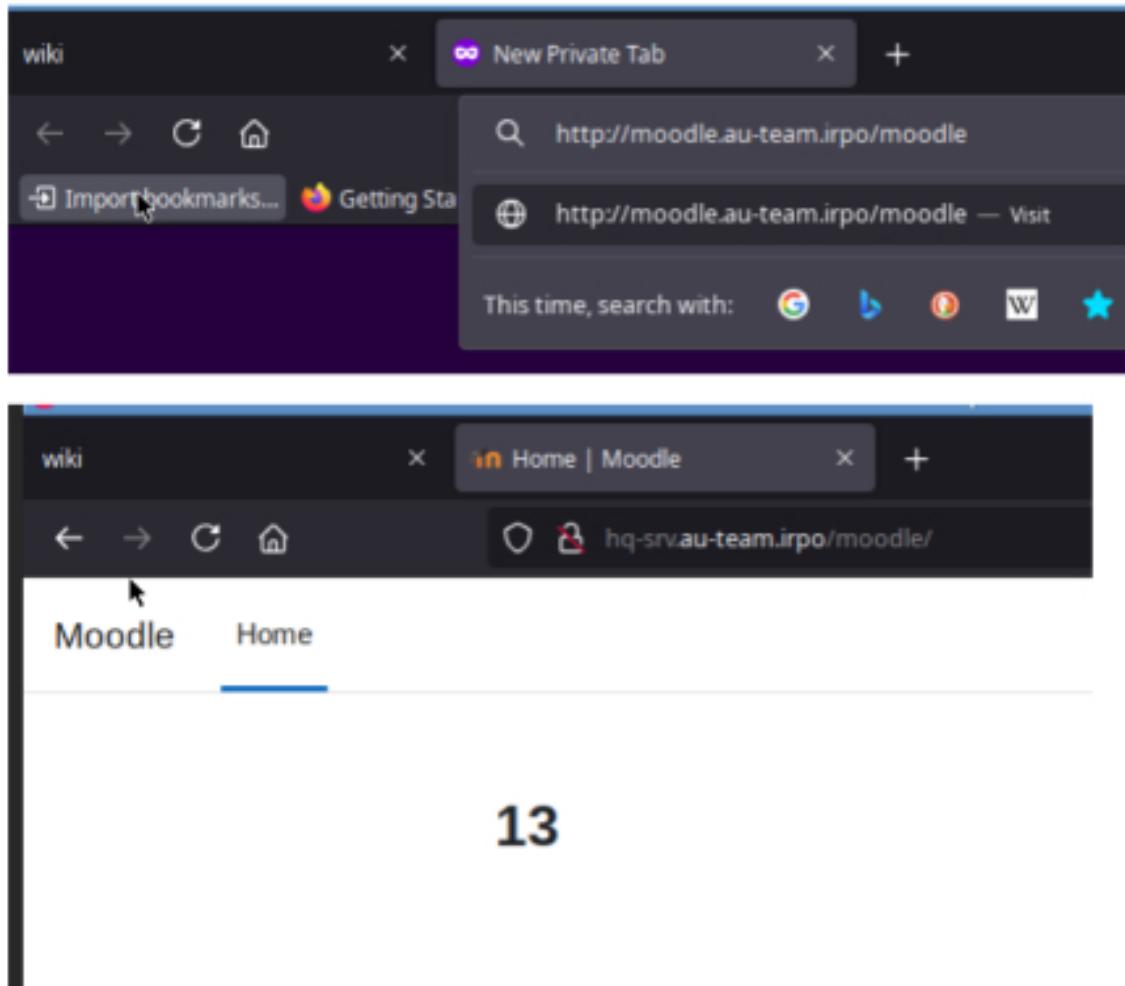
В браузере



Должен быть ридерект на



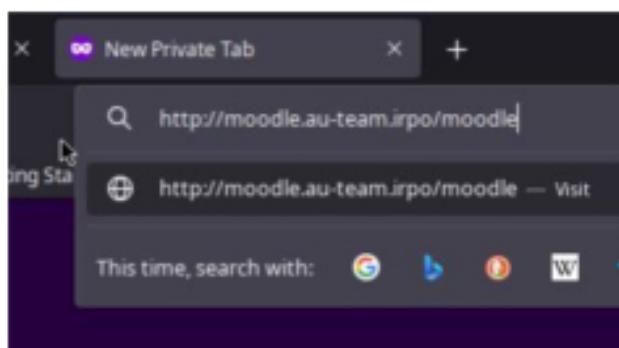
И moodle



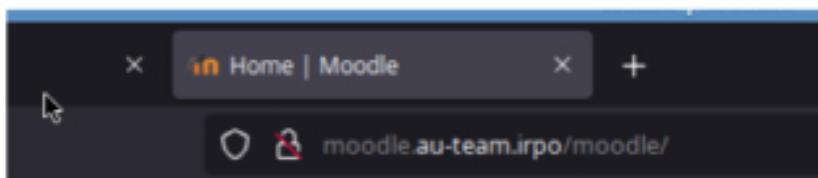
Можем поменять (скрыть) запись адреса moodle

```
[root@hq-srv ~]# vim /var/www/webapps/moodle/config.php
...
$CFG->wwwroot    = 'http://moodle.au-team.irpo/moodle';
$CFG->dataroot   = '/var/lib/moodle/default';
$CFG->admin      = 'admin';
[root@hq-srv ~]# systemctl restart httpd2.service
```

Проверим



Должен открыться



1e

13

Также можем поменять (скрыть) запись адреса wiki

```
[root@br-srv user]# vim LocalSettings.php
```

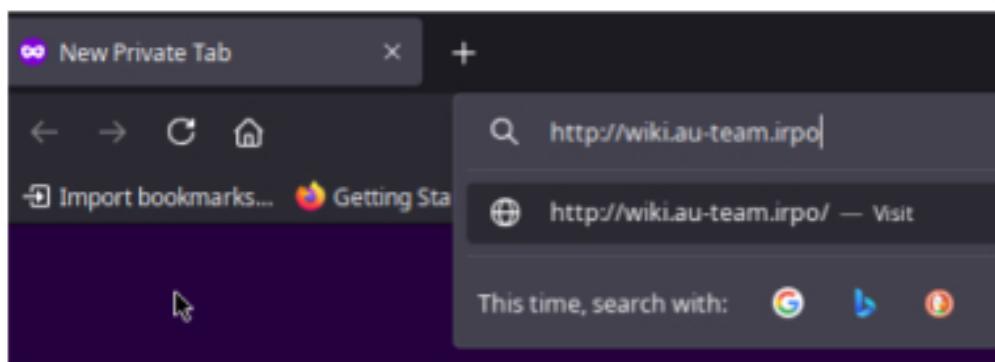
Изменим строку

```
## The protocol and server name to use in fully-qualified URLs
$wgServer = "http://br-srv:8080";
```

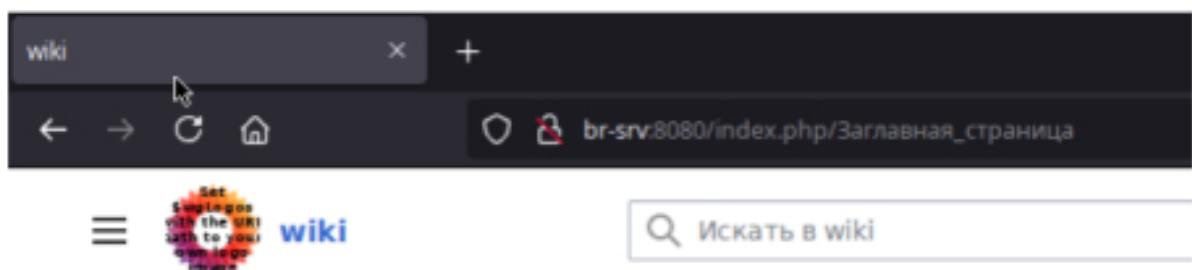
На

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

Проверим:



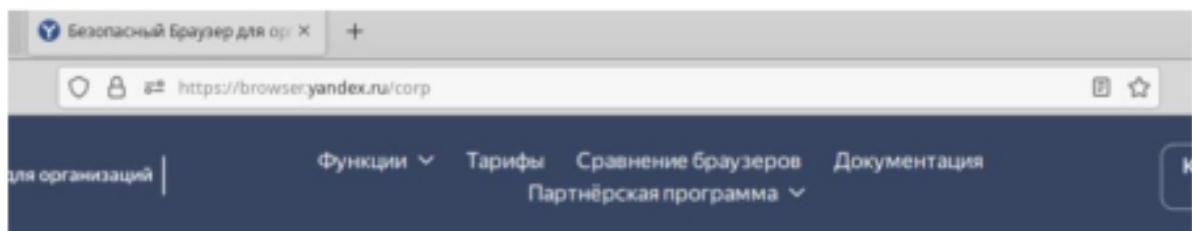
Должен открыться wiki



9. Удобным способом установите приложение Яндекс Браузер для организаций на HQ-CLI

apt-get -y install yandex-browser-stable.

Или в браузере заходим на сайт яндекс браузера

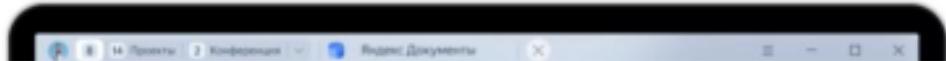


Безопасный Яндекс Браузер для компаний любого масштаба

Работает на Windows, macOS и Linux, в том числе
на российских: AlterOS, RedOS, Astra, Alt и Rosa.

Настроить Браузер

Для крупных компаний





Скачиваем rpm пакет и устанавливаем из него

```
hq-cli Downloads # apt-get install yandex-browser.rpm
```