

# Оглавление

Модуль 1. Организация сетевого администрирования операционных систем.....	2
Задание 1 Настройте доменный контроллер Samba на машине BR-SRV.....	3
Выполнение.....	3
Задание 2 Сконфигурируйте файловое хранилище.....	8
Выполнение.....	8
Задание 3 Настройте службу сетевого времени на базе сервиса chrony.....	12
Выполнение.....	12
Задание 4 Сконфигурируйте ansible на сервере BR-SRV.....	14
Выполнение.....	14
Задание 5 Развертывание приложений в Docker на сервере BR-SRV.....	16
Выполнение.....	16
Задание 6 На маршрутизаторах сконфигурируйте статическую трансляцию портов.....	24
Выполнение.....	24
Задание 7 Запустите сервис moodle на сервере HQ-SRV.....	26
Выполнение.....	26
Задание 8 Настройте веб-сервер nginx как обратный прокси-сервер на HQ-RTR.....	32
Выполнение.....	32
Задание 9 Удобным способом установите приложение Яндекс Браузере для организаций на HQ-CLI.....	33
Выполнение.....	33

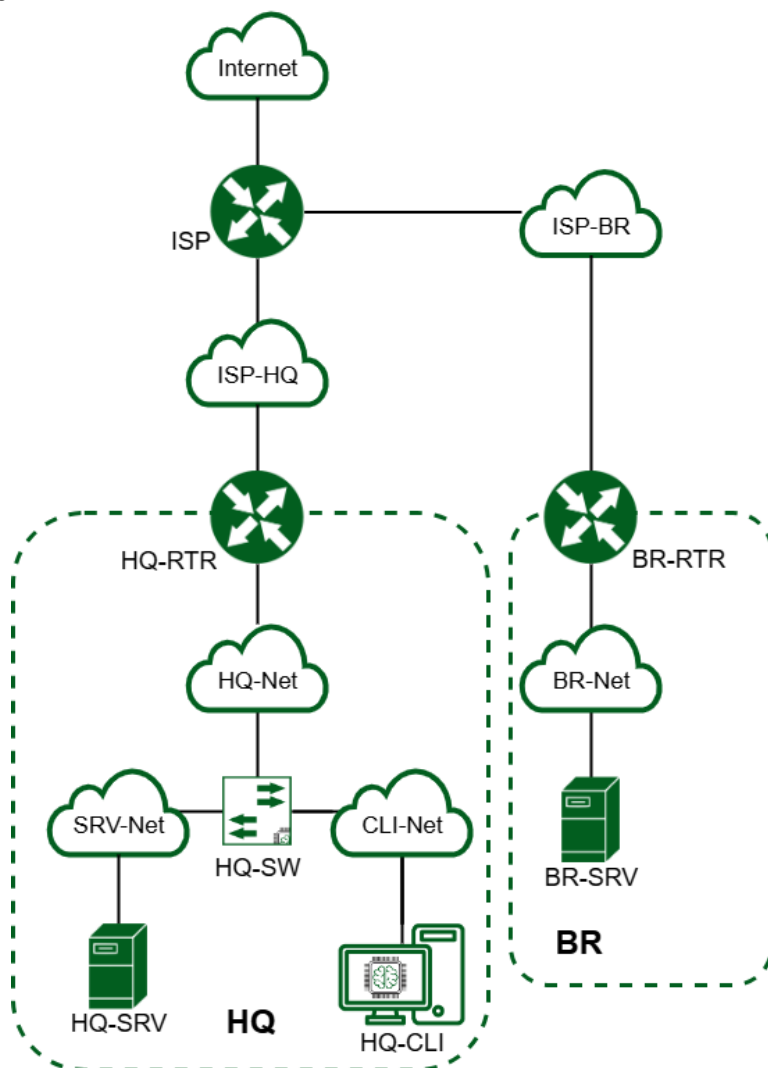
# Модуль 1. Организация сетевого администрирования операционных систем

Задание:

Необходимо разработать и настроить инфраструктуру информационно-коммуникационной системы согласно предложенной топологии(см. Рисунок ).

Для модуля 2 используется отдельный стенд. В стенде преднастроены:

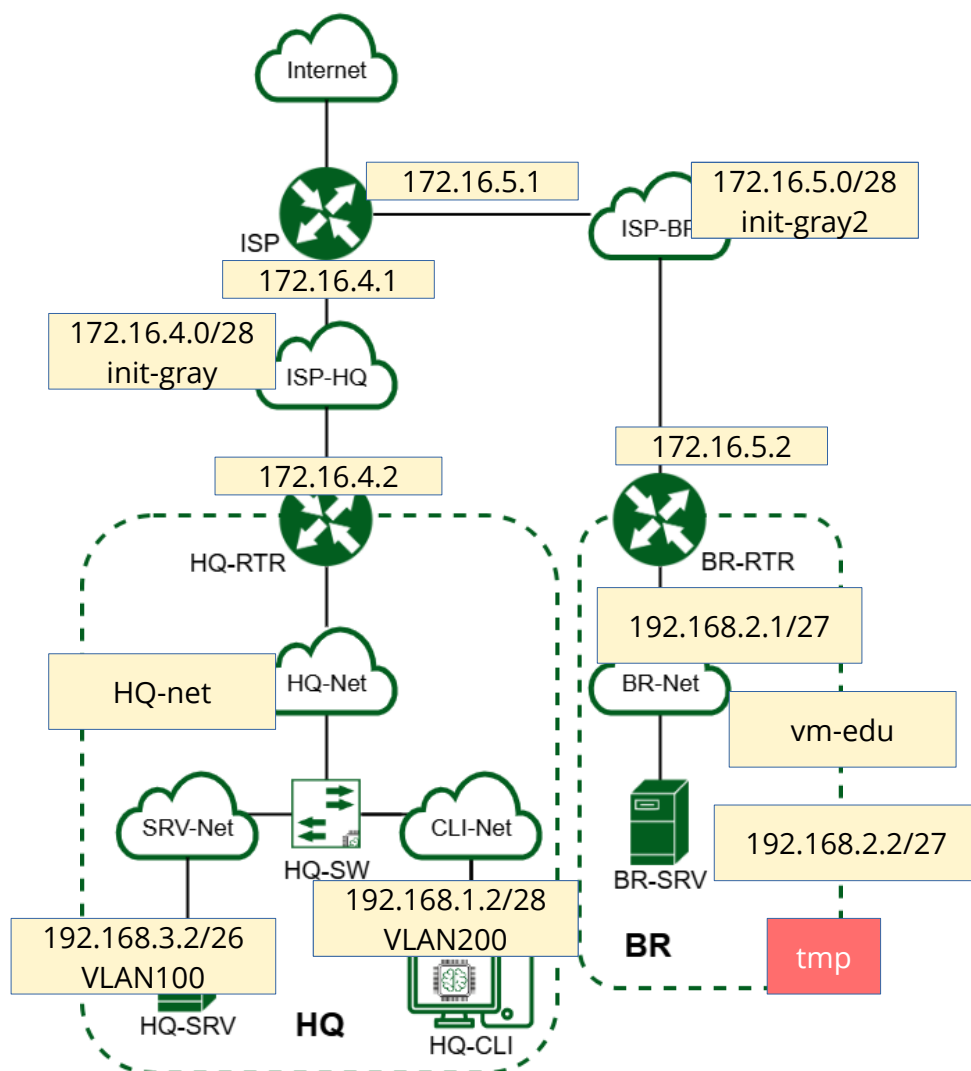
- IP-адреса, маски подсетей и шлюзы по умолчанию
- Сетевая трансляция адресов
- IP туннель
- Динамическая маршрутизация
- Созданы пользователи sshuser на серверах и net\_admin на маршрутизаторах
- DHCP-сервер
- DNS-сервер



## Задание 1 Настройте доменный контроллер Samba на машине BR-SRV

- Создайте 5 пользователей для офиса HQ: имена пользователей формата userNo.hq.
- Создайте группу hq, введите в эту группу созданных пользователей
- Введите в домен машину HQ-CLI
- Пользователи группы hq имеют право аутентифицироваться на клиентском ПК
- Пользователи группы hq должны иметь возможность повышать привилегии для выполнения ограниченного набора команд: cat, grep, id. Запускать другие команды с повышенными привилегиями пользователи группы не имеют права
- Выполните импорт пользователей из файла users.csv. Файл будет располагаться на виртуальной машине BR-SRV в папке /opt

### Выполнение



Настройки сервера подробно описаны в БЗ

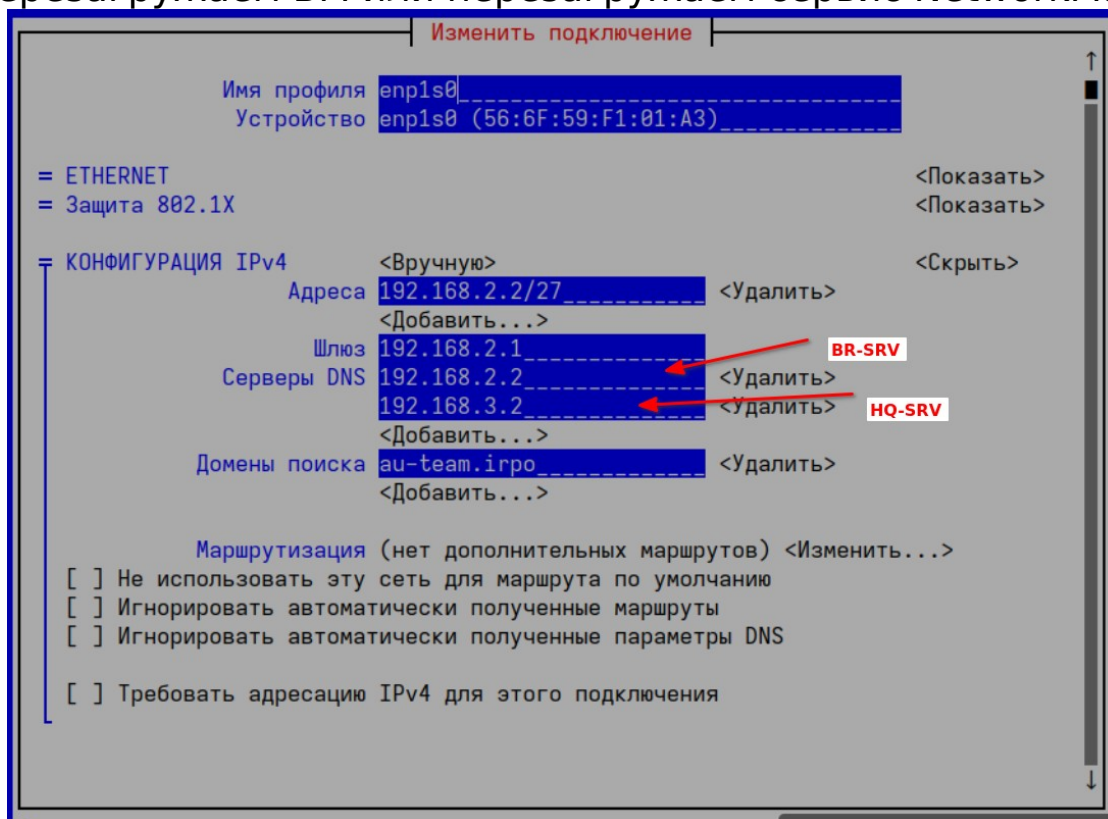
[https://redos.red-soft.ru/base/redos-8\\_0/8\\_0-administation/8\\_0-domain-redos/8\\_0-domain-config/8\\_0-samba-dns-backend-samba-internal/8\\_0-install-samba-dc-internal/?sphrase\\_id=843861](https://redos.red-soft.ru/base/redos-8_0/8_0-administation/8_0-domain-redos/8_0-domain-config/8_0-samba-dns-backend-samba-internal/8_0-install-samba-dc-internal/?sphrase_id=843861)

На ВМ BR-SRV переводим SELinux в режим пермисив через конфигурационный файл.

Проверяем что у ВМ BR-SRV установлено правильное полное имя

хоста. Если нет, то исправляем

Дополнительно настраиваем сетевой интерфейс. После настройки либо перезагружаем ВМ или перезагружаем сервис NetworkManager.



Отключаем DNS-службы **systemd-resolved** в файле **/etc/systemd/resolved.conf**. Устанавливаем параметр **DNSStubListener** в значение **no**, отключив прослушивание **systemd-resolved** на порту 53.

Перезапускаем **systemd-resolved** и **NetworkManager**  
**systemctl restart systemd-resolved.service NetworkManager**  
Проверяем изменения в настройках

Для проверки установленных DNS серверов можно использовать команду

**resolvectl**  
или  
**cat /etc/resolv.conf**

Устанавливаем пакет **samba-dc** и пакет кербероса  
**dnf install samba\* krb5\* -y**

Переименовываем файл **/etc/smb.conf**, он будет создан в процессе выполнения команды **samba-tool**  
**rm /etc/samba/smb.conf\***

Настройка конфигурации Kerberos **/etc/krb5.conf**. В файле необходимо привести к правильному виду закоментированные

строчки.

```
[libdefaults]
    dns_lookup_realm = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true
    rdns = false
    pkinit_anchors = FILE:/etc/pki/tls/certs/ca-bundle.crt
    spake_preauth_groups = edwards25519
    dns_canonicalize_hostname = fallback
    qualify_shortname = ""
    default_realm = AU-TEAM.IRPO
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
    AU-TEAM.IRPO = {
        kdc = br-srv.au-team.irpo
        admin_server = br-srv.au-team.irpo
    }

[domain_realm]
    .au-team.irpo = AU-TEAM.IRPO
    au-team.irpo = AU-TEAM.IRPO
```

Редактируем файл /etc/krb5.conf.d/crypto-policies. Данные можно взять из статьи в БЗ.

```
default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 RC4-HMAC DES-CBC-CRC DES3-CBC-SHA1 DES-CBC-MD5
default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 RC4-HMAC DES-CBC-CRC DES3-CBC-SHA1 DES-CBC-MD5
preferred_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 RC4-HMAC DES-CBC-CRC DES3-CBC-SHA1 DES-CBC-MD5
```

Полуавтоматически конфигурируем сервер с помощью утилиты samba-tool

**samba-tool domain provision --use-rfc2307 --interactive**

```
[root@br-srv sshuser]# samba-tool domain provision --use-rfc2307 --interactive
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) [192.168.3.2]:
Administrator password:
Retype password:
```

После окончания установки запускаем и добавляем в автозагрузку службу samba:

**systemctl enable --now samba**

Проверка статуса службы:

**systemctl status samba**

Добавляем пользователей с указанием пароля

```
samba-tool user add user1.hq P@ssw0rd
samba-tool user add user2.hq P@ssw0rd
samba-tool user add user3.hq P@ssw0rd
samba-tool user add user4.hq P@ssw0rd
```

**samba-tool user add user5.hq P@ssw0rd**

Проверяем созданных пользователей

**samba-tool user list**

```
[root@br-srv sshuser]# samba-tool user list
user3.hq
user5.hq
Administrator
krbtgt
user4.hq
user1.hq
user2.hq
Guest
[root@br-srv sshuser]# |
```

Создаем группу и добавляем туда созданных пользователей

**samba-tool group add hq**

**samba-tool group addmembers hq**

**user1.hq,user2.hq,user3.hq,user4.hq,user5.hq**

Посмотрим состав группы

**samba-tool group listmembers hq**

Для ввода HQ-CLI в домен используем утилиту **join-to-domain.sh**.

```
[root@hq-cli admin]# join-to-domain.sh

Ввод РЕД ОС в домен Windows(2008/2012/2016/2019/2022), SAMBA, IPA

Этот компьютер не в домене!
Продолжить выполнение (y/n)? y

Выберите тип домена:
1. Ввод РЕД ОС в домен Windows/SAMBA
2. Ввод РЕД ОС в домен IPA
Укажите (1 или 2): 1

Имя домена [au-team.irpo]
Для подтверждения нажмите ENTER или введите имя домена вручную:

Введите имя ПК. Пример: client1
Имя ПК: hq-cli
Имя администратора домена: Administrator
Имя подразделения ПК(OU=MyComputers) без кавычек или для продолжения нажмите ENTER:
Домен au-team.irpo доступен!
Введите пароль администратора домена:
Продолжить выполнение (y/n)? y
1) Изменение имени ПК
   Новое имя ПК: hq-cli.au-team.irpo
2) Настройка chronyd
3) Настройка hosts
4) Ввод в домен ...
5) Настройка sssd
6) Настройка limits
7) Настройка samba
8) Ввод samba в домен и регистрация DNS записи
   Лог установки: /var/log/join-to-domain.log

Выполнено. Компьютер успешно введен в домен! Перезагрузите ПК.
```

ПРОВЕРКА пробуем зайти на HQ-CLI под доменной учётной записью.

Разрешаем пользователям группы hq выполнять команды cat, grep, id

с повышенными привелегиями.

На **HQ-CLI** в файле **/etc/sudoers** прописываем строчку  
**%hq ALL=(ALL) NOPASSWD: /bin/cat, /bin/id, /bin/grep**

Импортируем пользователей из приложенного файла.

Файл имеет заголовок примерно такого вида

```
[root@br-srv opt]# cat Users.csv | head
First Name;Last Name;Role;Phone;OU;Street;ZIP/Postal Code;City;Country/Region;Password
"1";"1";"1";"1";"1";"1";"1";"1";"1";"1"
```

Стандартного инструмента для импорта нет, только в промышленной версии РЕДАДМ.

Составим скрипт ипорта

```
#!/bin/bash
```

```
# Путь к файлу users.csv
```

```
INPUT_FILE="Users.csv"
```

```
# Проверка наличия файла
```

```
if [[ ! -f $INPUT_FILE ]]; then
```

```
    echo "Файл $INPUT_FILE не найден!"
```

```
    exit 1
```

```
fi
```

```
#Подсчёт количества импортируемых пользователей
```

```
max=$(( $(cat $INPUT_FILE | wc -l) - 1 ))
```

```
echo "Всего импортируемых пользователей -- $max"
```

```
i=1 #создаём порядковый номер импортируемого пользователя
```

```
# Чтение файла и импорт пользователей
```

```
while IFS=';' read -r first_name last_name role phone ou street zip city country password; do
```

```
    # Пропуск заголовка (первой строки)
```

```
    if [[ "$first_name" == "First Name" ]]; then
```

```
        continue
```

```
    fi
```

```
# Формирование команды samba-tool для создания пользователя
```

```
samba_command="samba-tool user add $first_name$last_name $( echo -n $  
{password} |tr '\r' ' ' ) --given-name=\"$first_name\" --surname=\"$last_name\" --  
telephone-number=\"$phone\" --job-title=\"$ou\" --mail-address=\"$ ( echo $street | tr  
-dc '[:print:]' ), $city, $zip, $country\""
```

```
# Выводим команду на экран или выполняем её
```

```
echo "Импортируем $i-го пользователя $first_name$last_name из $max"
```

```
i=$(( $i+1 )) #увеличиваем счётчик
```

```
eval $samba_command
```

```
done < "$INPUT_FILE"
```



## Задание 2 Сконфигурируйте файловое хранилище

- При помощи трёх дополнительных дисков, размером 1Гб каждый, на HQ-SRV сконфигурируйте дисковый массив уровня 5
- Имя устройства - md0, конфигурация массива размещается в файле /etc/mdadm.conf
- Обеспечьте автоматическое монтирование в папку /raid5
- Создайте раздел, отформатируйте раздел, в качестве файловой системы используйте ext4
- Настройте сервер сетевой файловой системы(nfs), в качестве папки общего доступа выберите /raid5/nfs, доступ для чтения и записи для всей сети в сторону HQ-CLI
- На HQ-CLI настройте автомонтирование в папку /mnt/nfs
- Основные параметры сервера отметьте в отчёте

### Выполнение

На ВМ HQ-SRV добавлено 3 диска по 1Г. RAID массив будем размещать на эти диски

```
[sshuser@HQ-SRV ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   30G  0 disk
├─sda1       8:1    0    1M  0 part
├─sda2       8:2    0    1G  0 part /boot
├─sda3       8:3    0    3G  0 part [SWAP]
└─sda4       8:4    0   26G  0 part /
sdb          8:16   0    1G  0 disk
sdc          8:32   0    1G  0 disk
sdd          8:48   0    1G  0 disk
sr0         11:0    1    2K  0 rom
zram0       252:0    0   3.8G  0 disk [SWAP]
```

Установим утилиту mdadm (если она не установлена)

**dnf install mdadm -y**

На каждом диске ( /dev/sdb, /dev/sdc, /dev/sdd ) создадим по 1 разделу. Пример варианта создания раздела на диске представим

```
[root@HQ-SRV sshuser]# fdisk /dev/sdb

Добро пожаловать в fdisk (util-linux 2.38.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Устройство не содержит стандартной таблицы разделов.
Создана новая метка DOS с идентификатором 0x0a9c87dd.

Команда (m для справки): g
Created a new GPT disklabel (GUID: 1C982F97-AA21-C64D-AB8D-BB1838C39A08).

Команда (m для справки): n
Номер раздела (1-128, default 1):
Первый сектор (2048-2097118, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097118, default 2095103):

Создан новый раздел 1 с типом 'Linux filesystem' и размером 1022 MiB.

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.
```

Проверяем, что в результате на дисках появились разделы



```
[root@HQ-SRV sshuser]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda          8:0    0   30G  0 disk
├─sda1       8:1    0    1M  0 part
├─sda2       8:2    0    1G  0 part  /boot
├─sda3       8:3    0    3G  0 part  [SWAP]
└─sda4       8:4    0   26G  0 part  /
sdb          8:16   0    1G  0 disk
└─sdb1       8:17   0 1022M  0 part
sdc          8:32   0    1G  0 disk
└─sdc1       8:33   0 1022M  0 part
sdd          8:48   0    1G  0 disk
└─sdd1       8:49   0 1022M  0 part
sr0         11:0    1     2K  0 rom
zram0       252:0    0   3,8G  0 disk  [SWAP]
```

Для сборки массива применяем следующую команду

```
mdadm --create --verbose /dev/md0 -l 5 -n 3 /dev/sdb1
/dev/sdc1 /dev/sdd1
```

где:

**/dev/md0** — устройство RAID, которое появится после сборки;

**-l 5** — уровень RAID;

**-n 3** — количество дисков, из которых собирается массив;

**/dev/sdb1 /dev/sdc1 /dev/sdd1** — сборка выполняется из разделов дисков sdb, sdc, sdd.

```
[root@HQ-SRV sshuser]# mdadm --create --verbose /dev/md0 -l 5 -n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 1044480K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Проверяем, что в результате появились указатели устройства md0

```
[root@HQ-SRV sshuser]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda          8:0    0   30G  0 disk
├─sda1       8:1    0    1M  0 part
├─sda2       8:2    0    1G  0 part  /boot
├─sda3       8:3    0    3G  0 part  [SWAP]
└─sda4       8:4    0   26G  0 part  /
sdb          8:16   0    1G  0 disk
├─sdb1       8:17   0 1022M  0 part
└─md0        9:0    0    2G  0 raid5
sdc          8:32   0    1G  0 disk
├─sdc1       8:33   0 1022M  0 part
└─md0        9:0    0    2G  0 raid5
sdd          8:48   0    1G  0 disk
├─sdd1       8:49   0 1022M  0 part
└─md0        9:0    0    2G  0 raid5
sr0         11:0    1     2K  0 rom
zram0       252:0    0   3,8G  0 disk  [SWAP]
```

Посмотреть состояние всех RAID можно командой

```
[root@HQ-SRV sshuser]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
      2088960 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

Детальную информацию о массиве можно посмотреть командой:  
**mdadm -D /dev/md0**

Сохраняем конфигурацию массива в файле mdadm.conf командой:  
**mdadm --detail --scan --verbose >> /etc/mdadm.conf**

Создадим таблицу раздела на массиве  
**mkfs.ext4 /dev/md0**

Создадим каталог /raid5  
**mkdir /raid5**

Примонтируем созданный рейд в созданную папку  
**mount /dev/md0 /raid5**

ПРОВЕРЯЕМ

```
[root@HQ-SRV sshuser]# df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
devtmpfs           4,0М      0      4,0М           0%  /dev
tmpfs              2,0Г      0      2,0Г           0%  /dev/shm
tmpfs             783М     3,1М     780М           1%  /run
/dev/sda4          26Г      4,9Г      20Г          20%  /
tmpfs             2,0Г      0      2,0Г           0%  /tmp
/dev/sda2          974М     165М     742М          19%  /boot
tmpfs             392М      8,0К     392М           1%  /run/user/1010
/dev/md0           2,0Г      24К      1,9Г           1%  /raid5
[root@HQ-SRV sshuser]#
```

Добавляем автоматическое монтирование рейда в /raid5  
**echo '/dev/md0 /raid5 ext4 defaults 0 0 ' >> /etc/fstab**

Устанавливаем (если он не установлен) и настраиваем NFS сервер  
**dnf install nfs-utils nfs4-acl-tools -y**

Создаем папку общего доступа /raid5/nfs  
**mkdir /raid5/nfs**

В файл конфигурации NFS добавляем сведения о расшариваемой папке.

```
echo '/raid5/nfs
192.168.1.0/28(rw,sync,no_root_squash,subtree_check)' >>
/etc/exports
```

где:

**192.168.1.0/28** — сеть ВМ HQ-CLI

**rw** — разрешить чтение и запись.

**sync** — синхронный режим доступа. Опция sync указывает, что сервер должен отвечать на запросы только после записи на диск изменений, выполненных этими запросами.

**no\_root\_squash** — не подменять запросы от root на анонимные.

**subtree\_check** — проверять попытки пользователя выйти за пределы экспортированной папки.

Запускаем сервис nfs-server

**systemctl start nfs-server.service**

Проверяем доступны ли папка по протоколу NFS

```
[root@HQ-SRV sshuser]# exportfs  
/raid5/nfs 192.168.1.0/28
```

На **HQ-CLI** устанавливаем клиента NFS:

**dnf install nfs-utils**

Создаем директорию для автомонтирования /mnt/nfs

**mkdir /mnt/nfs**

Монтируем шару

**mount -t nfs 192.168.3.2:/raid5/nfs /mnt/nfs**

Настраиваем автоматическое монтирование

**echo '192.168.3.2:/raid5/nfs /mnt/nfs nfs auto 0 0 ' >> /etc/fstab**

### Задание 3 Настройте службу сетевого времени на базе сервиса chrony

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

#### Выполнение

Заходим на **HQ-RTR (net\_admin P@\$word)**. Устанавливаем chrony (если он не установлен).

**dnf install chrony -y**

Вносим изменения в файл конфигурации /etc/chrony.conf

**nano /etc/chrony.conf**

Комментируем параметры серверов и дописываем строки.

**server 127.0.0.1 iburst prefer**

**local stratum 5**

**allow 192.168.1.0/28**

**allow 192.168.2.0/27**

**allow 192.168.3.0/26**

```
GNU nano 7.2 /etc/chrony.conf
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (https://www.pool.ntp.org/join.html).
#server ntp1.vniiftri.ru iburst
#server ntp2.vniiftri.ru iburst
#server ntp3.vniiftri.ru iburst
#server ntp4.vniiftri.ru iburst
server 127.0.0.1 iburst prefer
local stratum 5
allow 192.168.1.0/28
allow 192.168.2.0/27
allow 192.168.3.0/26
```

где

**server** — в качестве источника времени использовать Loopback

**iburst** — отправлять несколько пакетов (повышает точность)

**prefer** — указывает на предпочитаемый сервер

**allow** — устройства с каких подсетей имеют возможность синхронизироваться с сервером

**local stratum** — указать stratum (стратум), т.е. слой

Перезагружаем chronyd

**systemctl restart chronyd**

#### ПРОВЕРЯЕМ

```
[root@HQ-RTR net_admin]# chronyc sources
MS Name/IP address         Stratum Poll Reach LastRx Last sample
=====
^? localhost                0      7      0      -    +0ns[  +0ns] +/-    0ns
[root@HQ-RTR net_admin]#
```

Подключем клиентов

Вносим изменения в файл конфигурации /etc/chrony.conf

**nano /etc/chrony.conf**

Коментируем параметры серверов и дописываем строки.

**server 192.168.1.1 iburst**

Перезагружаем chronyd

**systemctl restart chronyd**

ПРОВЕРЯЕМ на клиенте

**chronyc sources**

```
[root@HQ-SRV sshuser]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* hq-rtr.au-team.irpo      5      6    17    62  -3594ns[ -23us] +/- 144us
```

ПРОВЕРЯЕМ на сервере

**chronyc clients**

## Задание 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

### Выполнение

Устанавливаем Ansible на BR-SRV

**dnf install ansible -y**

Для удалённого доступа требуется пару ключей. Сгенерируем от пользователя sshuser.

**ssh-keygen**

В результате в каталоге /home/sshuser/.ssh будут созданы файлы ключей **id\_rsa** — закрытый ключ и **id\_rsa.pub** — открытый ключ.

От пользователя sshuser отправляем открытый ключ на:

HQ-SRV	ssh-copy-id -p 2024 sshuser@192.168.3.2
HQ-CLI	ssh-copy-id admin@192.168.1.2
HQ-RTR	ssh-copy-id -p 2024 net_admin@192.168.3.1
BR-RTR	ssh-copy-id -p 2024 net_admin@192.168.2.1

Для того чтобы Ansible знал, на каких хостах необходимо выполнять действия, используется файл инвентаря. По умолчанию /etc/ansible/hosts . В нем указываются IP-адреса или доменные имена необходимых серверов

Создадим свой файл инвентаря и укажем в нём все наши сервера

**nano /etc/ansible/hosts**

```
GNU nano 7.2 /etc/ansible/hosts
[hq]
192.168.1.1 ansible_port=2024 ansible_user=net_admin
192.168.3.2 ansible_port=2024 ansible_user=sshuser
192.168.1.2 ansible_user=admin
[br]
192.168.2.1 ansible_port=2024 ansible_user=net_admin
```

В файле распределим устройства по группам (HQ и BR). Сначала указывается название группы в квадратных скобках, затем перечисляются хосты. Указываем дополнительные параметры номер порта ssh — ansible\_port, и имя пользователя — ansible\_user.

Запускаем пинг хостов

**ansible all -m ping**

Может появиться предупреждение про обнаружение

интерпретатора Python , на целевом хосте

```
[WARNING]: Platform linux on host 192.168.3.2 is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference\_appendices/interpreter\_discovery.html for more information.
```

Если данное сообщение хочется убрать, то в файле конфигурации ansible **/etc/ansible/ansible.cfg** в разделе **[defaults]** прописываем ключ **interpreter\_python=auto\_silent** .

```
GNU nano 7.2 /etc/ansible/ansible.cfg
[defaults]
interpreter_python=auto_silent
```

Повторно запускаем пинг хостов

**ansible all -m ping**

```
[sshuser@BR-SRV ~]$ ansible all -m ping
192.168.1.1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
192.168.3.2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
192.168.2.1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
[sshuser@BR-SRV ~]$
```



## Задание 5 Развертывание приложений в Docker на сервере BR-SRV

- Создайте в домашней директории пользователя файл `wiki.yml` для приложения MediaWiki.
- Средствами `docker compose` должен создаваться стек контейнеров с приложением MediaWiki и базой данных.
- Используйте два сервиса
- Основной контейнер MediaWiki должен называться `wiki` и использовать образ `mediawiki`
- Файл `LocalSettings.php` с корректными настройками должен находиться в домашней папке пользователя и автоматически монтироваться в образ.
- Контейнер с базой данных должен называться `mariadb` и использовать образ `mariadb`.
- Разверните
- Он должен создавать базу с названием `mediawiki`, доступную по стандартному порту, пользователя `wiki` с паролем `WikiP@ssw0rd` должен иметь права доступа к этой базе данных
- MediaWiki должна быть доступна извне через порт 8080.

### Выполнение

Установим поддержку контейнеризации на BR-SRV:

**`dnf install docker-ce docker-ce-cli -y`**

Запускаем и добавляем в автозагрузку

**`systemctl enable docker --now`**

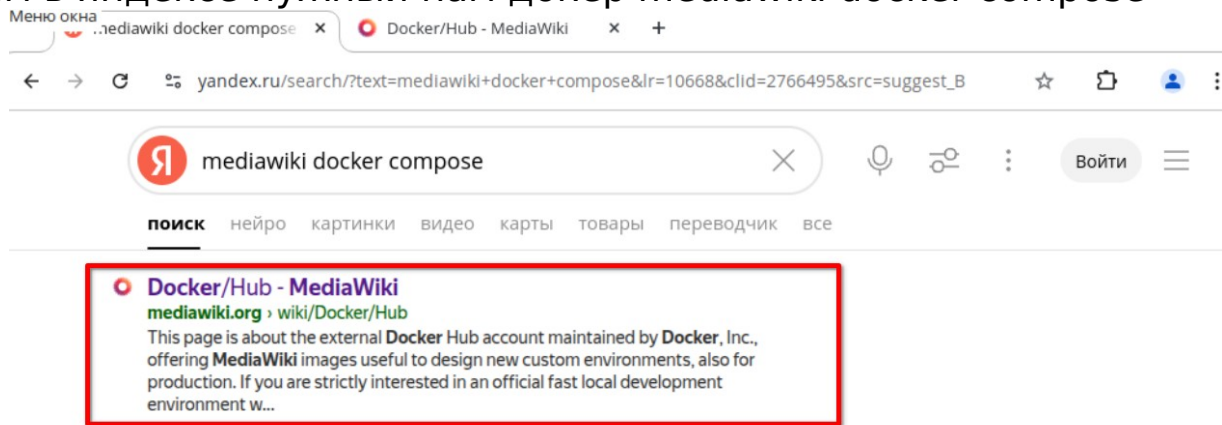
Получим информацию об установленном docker (вот не полный вывод):

```
[sshuser@BR-SRV ~]$ sudo docker info
Client:
 Version:      27.4.1
 Context:      default
 Debug Mode:   false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
   Version:  0.19.3
   Path:     /usr/libexec/docker/cli-plugins/docker-buildx
Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 27.4.1
 Storage Driver: overlay2
 Backing Filesystem: extfs
```

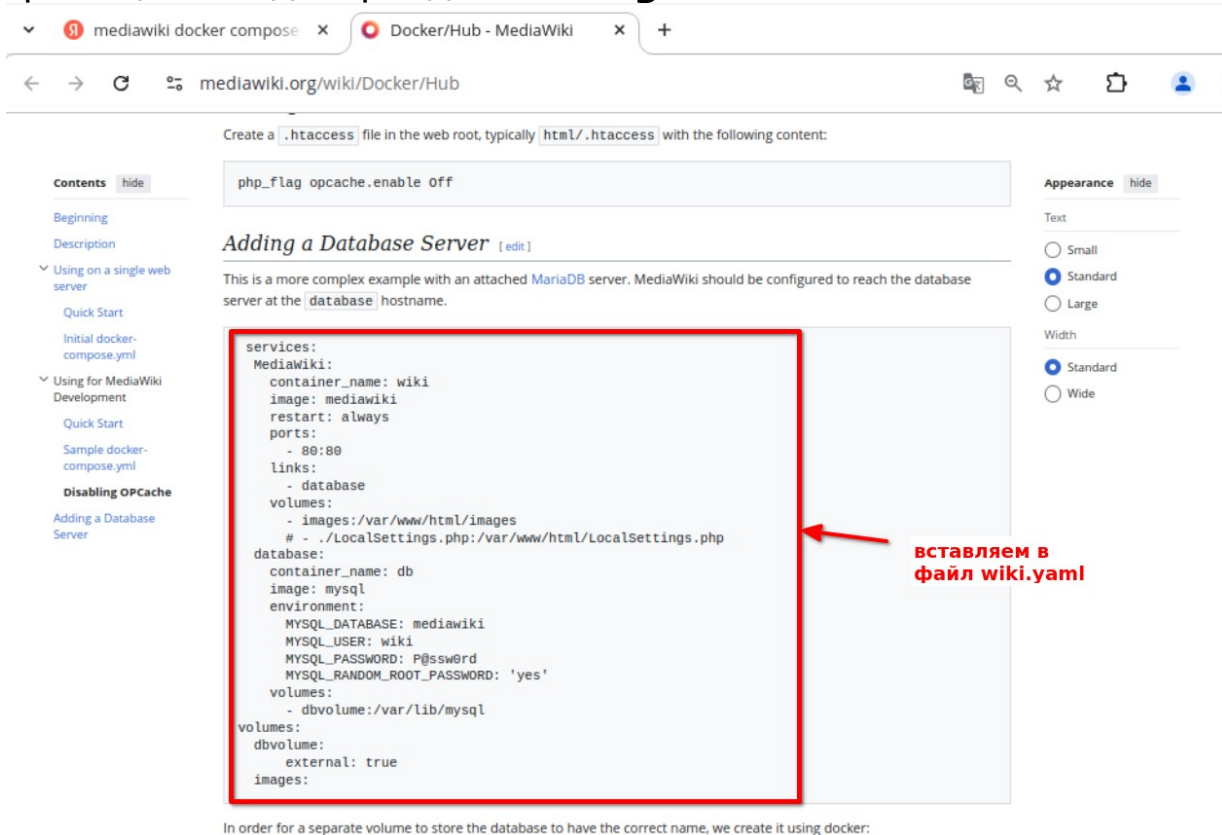
Установим `docker-compose`:

**`dnf install docker-compose -y`**

## Ищем в яндексе нужный нам докер mediawiki docker compose



## На странице находим раздел **Adding a Database Server**



```

GNU nano 7.2                               wiki.yml
services:
  MediaWiki:
    container_name: wiki
    image: mediawiki
    restart: always
    ports:
      - 8080:80
    links:
      - database
    volumes:
      - images:/var/www/html/images
        # - ./LocalSettings.php:/var/www/html/LocalSettings.php
  database:
    container_name: mariadb
    image: mariadb
    environment:
      MYSQL_DATABASE: mediawiki
      MYSQL_USER: wiki
      MYSQL_PASSWORD: WikiP@ssw0rd
      MYSQL_RANDOM_ROOT_PASSWORD: 'yes'
    volumes:
      - dbvolume:/var/lib/mariadb
volumes:
  dbvolume:
    external: true

```

Создаём отдельный **volume** для хранения базы данных  
**sudo docker volume create dbvolume**  
 Проверяем созданный отдельный **volume**  
**sudo docker volume ls**

```

[sshuser@BR-SRV ~]$ sudo docker volume create dbvolume
dbvolume
[sshuser@BR-SRV ~]$ sudo docker volume ls
DRIVER      VOLUME NAME
local       dbvolume

```

Выполняем сборку и запуск стека контейнеров с приложением MediaWiki и базой данных описанных в файле wiki.yml  
**docker-compose -f wiki.yml up -d**

Проверяем установленные контейнеры

```

[sshuser@BR-SRV ~]$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
55f36a9655d8   mediawiki     "docker-php-entrypoi..." 5 minutes ago  Up 5 minutes  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  wiki
da0db3aef1c0   mariadb       "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes  3306/tcp                               mariadb

```

Настройка через Web – интерфейс. Переходим в браузер  
<http://192.168.2.2:8080>



Выбираем необходимый Язык - жмем Далее:

## Установка MediaWiki 1.43.0

### Язык

Ваш язык:

[?](#) справка

ru - русский


Язык, который будет использовать вики:

[?](#) справка

ru - русский

[Далее →](#)

← → ↻ ⚠ Не защищено 192.168.2.2:8080/mw-config/index.php?page=Welcome 🔍 ☆ 📌 👤 ⋮



[Сайт MediaWiki](#)  
[Справка для пользователей](#)  
[Справка для администраторов](#)  
[ЧЗВ](#)  
[Искать помощи](#)  
[Система отслеживания ошибок](#)  
[Вклад](#)

[Информация о версии](#)  
[Копирование](#)  
[Обновление](#)

1 Используется имя сервера «http://192.168.2.2:8080».

2 Используется URL сервера «http://192.168.2.2:8080».

3 **Предупреждение.** Запросы к изображениям в вашем каталоге загрузки должны отвечать HTTP-заголовком `x-content-type-options: nosniff`, чтобы защитить браузеры от потенциально небезопасных файлов.

Настоятельно рекомендуется [настроить соответствующие заголовки ответов на вашем веб-сервере](#) перед тем, как включить загрузку.

✓ Проверка внешней среды была успешно проведена. Вы можете установить MediaWiki.

### Авторские права и условия

Эта вики работает на движке [MediaWiki](#), copyright © 2001-2025 Magnus Manske, Brooke Vibber, Lee Daniel Crocker, Tim Starling, Erik Möller, Gabriel Wicke, Ævar Arnfjörð Bjarmason, Niklas Laxström, Domas Mituzas, Rob Church, Yuri Astrakhan, Aryeh Gregor, Aaron Schulz, Andrew Garrett, Raimond Spekking, Alexandre Emsenhuber, Siebrand Mazeland, Chad Horohoe, Roan Kattouw, Trevor Parscal, Bryan Tong Minh, Sam Reed, Victor Vasiliev, Rotem Liss, Platonides, Antoine Musso, Timo Tijhof, Daniel Kinzler, Jeroen De Dauw, Brad Jorsch, Bartosz Dzielowski, Ed Sanders, Moriel Schottlender, Kunal Mehta, James D. Forrester, Brian Wolff, Adam Shorland, Danny5712, Ori Livneh, Max Semeniuk, Amir Sarabadani, Derk-Jan Hartman, Petr Pchelko, [другие](#) и [переводчики translatewiki.net](#).

MediaWiki — свободное программное обеспечение, которое вы можете распространять и/или изменять в соответствии с условиями лицензии GNU General Public License, опубликованной фондом свободного программного обеспечения; второй версии, либо любой более поздней версии.


MediaWiki распространяется в надежде, что она будет полезной, но **без каких-либо гарантий**, даже без подразумеваемых гарантий **коммерческой ценности** или **пригодности для определённой цели**. См. лицензию GNU General Public License для более подробной информации.

Вы должны были получить [копию GNU General Public License](#) вместе с этой программой, если нет, то напишите Free Software Foundation, Inc., по адресу: 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA или [прочтите её онлайн](#).

[← Назад](#) [Далее →](#)

Заполняем следующее окно в соответствии с файлом wiki.yml

← → ↻ ⚠ Не защищено 192.168.2.2:8080/mw-config/index.php?page=DBConnect 🔍 ☆ 📌 👤 ⋮



[Сайт MediaWiki](#)  
[Справка для пользователей](#)  
[Справка для администраторов](#)  
[ЧЗВ](#)  
[Искать помощи](#)  
[Система отслеживания ошибок](#)  
[Вклад](#)

[Информация о версии](#)  
[Копирование](#)  
[Обновление](#)

Тип базы данных:

☒ MariaDB, MySQL или совместимая

☐ SQLite

### 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

\*\*\*\*\*

[← Назад](#) [Далее →](#)



[Сайт MediaWiki](#)  
[Справка для пользователей](#)  
[Справка для администраторов](#)  
[ЧЗВ](#)  
[Искать помощи](#)  
[Система отслеживания ошибок](#)  
[Вклад](#)

[Информация о версии](#)  
[Копирование](#)  
[Обновление](#)

## Установка MediaWiki 1.43.0


### Настройки базы данных

**Учётная запись для доступа к базе данных из веб-сервера**

☒ Использовать ту же учётную запись, что и для установки

← Назад
Далее →

- Язык
- Существующая вики
- Добро пожаловать в MediaWiki!
- Подключение к базе данных
- Обновление существующей установки
- **Настройки базы данных**
  - Название
  - Настройки
  - Установка
  - Готово!
- [Начать установку заново](#)



[Сайт MediaWiki](#)  
[Справка для пользователей](#)  
[Справка для администраторов](#)  
[ЧЗВ](#)  
[Искать помощи](#)  
[Система отслеживания ошибок](#)  
[Вклад](#)

[Информация о версии](#)  
[Копирование](#)  
[Обновление](#)

## Установка MediaWiki 1.43.0

### Название

Название вики:

? справка

Demo-Wiki

Пространство имён проекта:

? справка

То же, что имя вики: Demo-Wiki

☐ Проект

☐ Другое (укажите)

**Учётная запись администратора**

Ваше имя участника:

? справка

wiki

Пароль:

.....

Пароль ещё раз:

.....

Адрес электронной почты:

? справка

demo@demo.demo

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

? справка

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

? справка

**Информация**

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

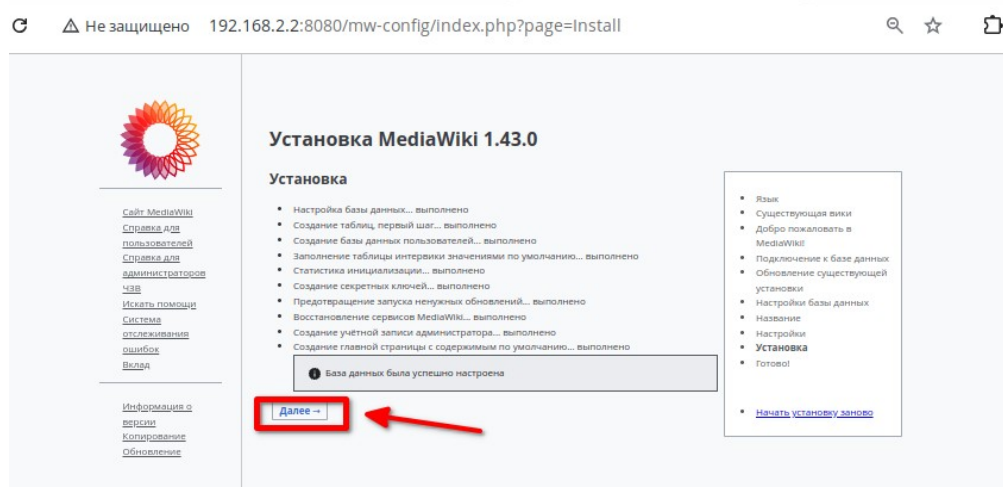
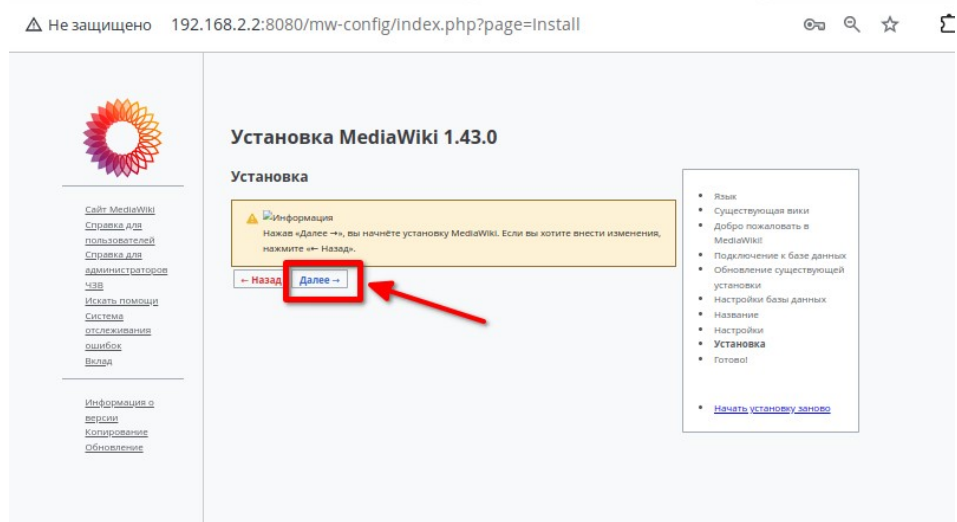
☐ Произвести тонкую настройку

☒ Хватит уже, просто установите вики.

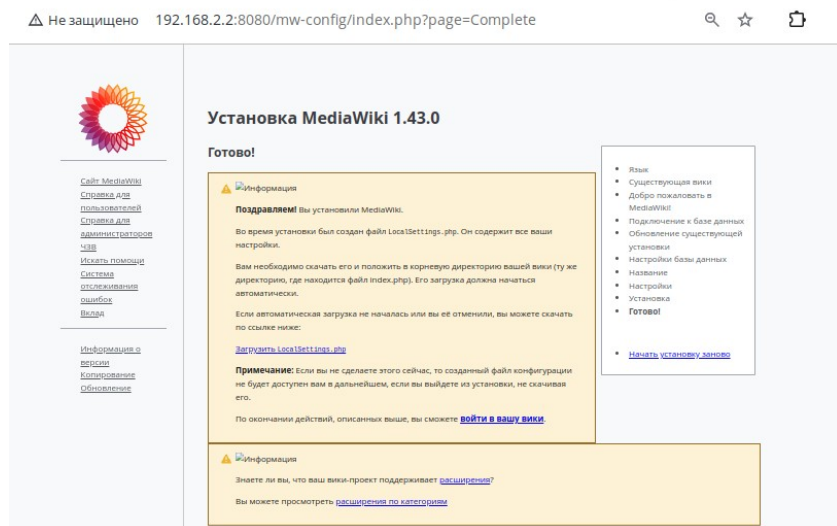
← Назад
Далее →

- Язык
- Существующая вики
- Добро пожаловать в MediaWiki!
- Подключение к базе данных
- Обновление существующей установки
- Настройки базы данных
- **Название**
  - Название
  - Настройки
  - Установка
  - Готово!
- [Начать установку заново](#)





Скачается файл, возможно придётся подтвердить скачку файла.



Файл нужно переместить на BR-SRV в папку, где лежит wiki.yml.  
**scp -P 2024 LocalSettings.php sshuser@192.168.2.2:~/**

На BR-SRV в файле wiki.yml раскомментируем строку с именем файла LocalSettings.php

```

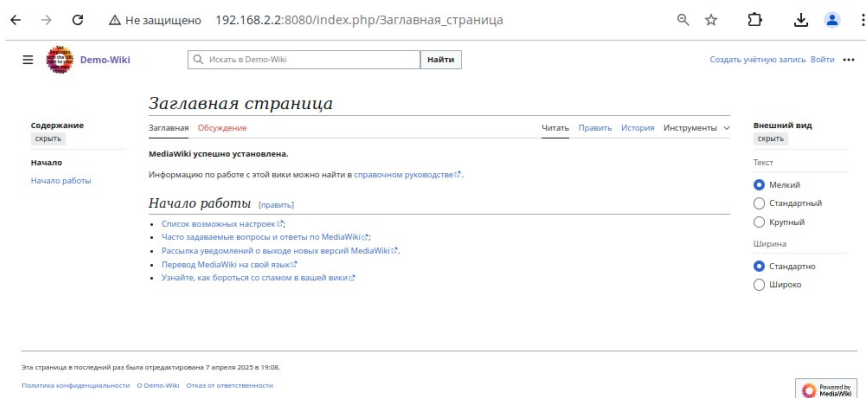
GNU nano 7.2                               wiki.yml
services:
  MediaWiki:
    container_name: wiki
    image: mediawiki
    restart: always
    ports:
      - 8080:80
    links:
      - database
    volumes:
      - images:/var/www/html/images
      - ./LocalSettings.php:/var/www/html/LocalSettings.php
  database:
    container_name: mariadb
    image: mariadb
    environment:
      MYSQL_DATABASE: mediawiki
      MYSQL_USER: wiki
      MYSQL_PASSWORD: WikiP@ssw0rd
      MYSQL_RANDOM_ROOT_PASSWORD: 'yes'
    volumes:
      - dbvolume:/var/lib/mariadb
volumes:
  dbvolume:
    external: true
  images:

```

Перезапускаем сервисы средствами docker-compose :

```
docker-compose -f wiki.yml stop 2
docker-compose -f wiki.yml up -d
```

Проверяем доступ к Wiki **http://192.168.200.2:8080**



Входим под пользователя **wiki** с паролем **WikiP@ssw0rd**

Возможно пригодится:

Остановка всех контейнеров: **docker stop \$(docker ps -qa)**

Удаление всех контейнеров: **docker rm \$(docker ps -qa)**

Удаление всех образов **docker rmi -f \$(docker images -qa)**

Удаление всех томов **docker volume rm \$(docker volume ls -q)**

Удаление всех сетей **docker network rm \$(docker network ls -q)**

Команды не должны выводить какие-либо элементы:

**docker ps -a**

**docker images -a**

**docker volume ls**

Команда удаляет неиспользуемые образы

**docker system prune**

**ЗАМЕЧАНИЕ.** Докер с базой после перезагрузки если не



```
[root@br-srv sshuser]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad45dd1eb5c9	mediawiki	"docker-php-entrypoi..."	40 hours ago	Up 19 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	wiki

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

```
[+] Running 2/2
```

- ✓ Container mariadb Started 0.7s
- ✓ Container wiki Running 0.0s

```
[root@br-srv sshuser]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad45dd1eb5c9	mediawiki	"docker-php-entrypoi..."	40 hours ago	Up 20 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	wiki
da0db3ae1fc0	mariadb	"docker-entrypoint.s..."	41 hours ago	Up 5 seconds	3306/tcp	mariadb

```
[root@br-srv sshuser]# |
```

## Задание 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.

### Выполнение

Обеспечения работы сервиса wiki. Делаем проброс порта 80 при обращении на внешний интерфейс BR-RTR (172.16.5.2 enp1s0) на порт 8080 BR-SRV

**iptables -t nat -A PREROUTING -i enp1s0 -p tcp --dport 80 -j DNAT --to-destination 192.168.2.2:8080**

Это можно сделать и с помощью *nftables*. На BR-RTR открываем на редактирование файл */etc/nftables/br-rtr.nft*. Прописываем правило

```
GNU nano 7.2 /etc/nftables/br-rtr.nft
table inet nat {
  chain PREROUTING {
    type nat hook prerouting priority filter;
    ip daddr 172.16.5.2 tcp dport 80 dnat ip to 192.168.2.2:8080
  }
  chain POSTROUTING {
    type nat hook postrouting priority srcnat;
    oifname "enp1s0" masquerade
  }
}
```

ПРОВЕРЯЕМ С HQ-CLI в браузере переходим по IP адресу <http://192.168.5.1> BR-RTR должны попасть на страницу MediaWiki

Пробросим порт 2024 на BR-RTR в порт 2024 на BR-SRV

**iptables -t nat -A PREROUTING -i enp1s0 -p tcp --dport 2024 -j DNAT --to-destination 192.168.2.2:2024**

Это можно сделать и с помощью *nftables*. На BR-RTR открываем на редактирование файл */etc/nftables/br-rtr.nft*. Дописываем правило

```
GNU nano 7.2 /etc/nftables/br-rtr.nft
table inet nat {
  chain PREROUTING {
    type nat hook prerouting priority filter;
    ip daddr 172.16.5.2 tcp dport 80 dnat ip to 192.168.2.2:8080
    ip daddr 172.16.5.2 tcp dport 2024 dnat ip to 192.168.2.2:2024
  }
  chain POSTROUTING {
    type nat hook postrouting priority srcnat;
    oifname "enp1s0" masquerade
  }
}
```

Пробросим порт 2024 на HQ-RTR в порт 2024 на HQ-SRV

**iptables -t nat -A PREROUTING -i enp1s0 -p tcp --dport 2024 -j DNAT --to-destination 192.168.3.2:2024**

Это можно сделать и с помощью *nftables*. На HQ-RTR открываем на редактирование файл */etc/nftables/hq-rtr.nft*. Дописываем правило

```

GNU nano 7.2 /etc/nftables/hq-rtr.nft
table inet nat {
    chain PREROUTING {
        type nat hook prerouting priority filter;
        ip daddr 172.16.4.2 tcp dport 2024 dnat ip to 192.168.3.2:2024
    }
    chain POSTROUTING {
        type nat hook postrouting priority srcnat;
        oifname "enp1s0" masquerade
    }
}

```

Проверка с HQ-CLI подключаемся по ssh к 172.16.5.2:2024. Должны попасть на BR-SRV

ssh sshuser@172.16.5.2 -p 2024

```

[admin@hq-cli2 Рабочий стол]$ ssh sshuser@172.16.5.2 -p 2024
#####
#           Authorized access only           #
#####
##### P@ssw0rd #####
sshuser@172.16.5.2's password:
Last login: Tue Apr  8 14:26:01 2025 from 172.16.4.2
[sshuser@BR-SRV ~]$

```

## Задание 7 Запустите сервис moodle на сервере HQ-SRV

- Используйте веб-сервер apache
- В качестве системы управления базами данных используйте mariadb
- Создайте базу данных moodledb
- Создайте пользователя moodle с паролем P@ssw0rd и предоставьте ему права доступа к этой базе данных
- У пользователя admin в системе обучения задайте пароль P@ssw0rd
- На главной странице должен отражаться номер рабочего места в виде арабской цифры, других подписей делать не надо
- Основные параметры отметьте в отчёте.

### Выполнение

На время настройки переведём selinux на HQ-SRV в режим уведомлений.

**setenforce 0**

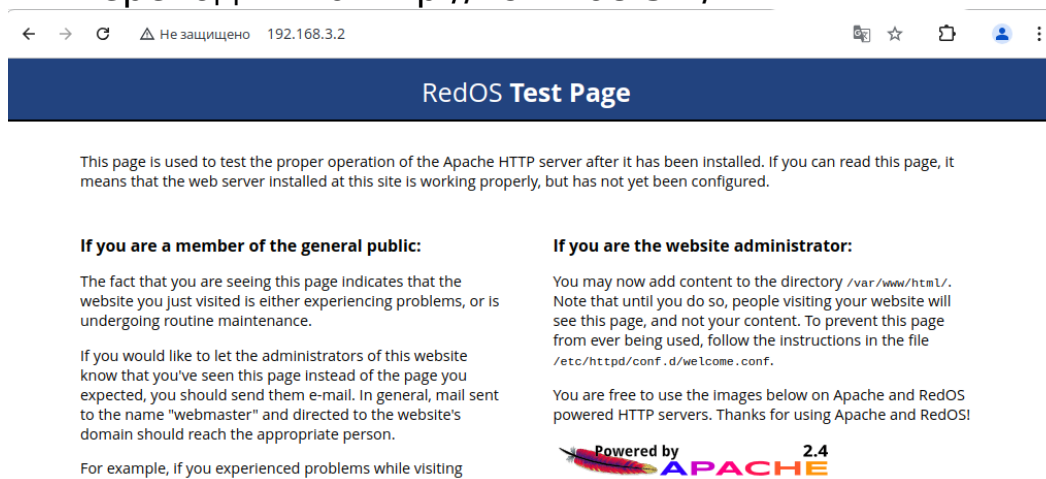
Установка веб-сервер Apache:

**dnf install httpd -y**

Запускаем службу и ставим ей в автозагрузку

**systemctl enable httpd --now**

ПРОВЕРИМ. Переходим на <http://192.168.3.2/>



Установка php и дополнительных расширений

**dnf install php php-mysqlnd php-gd php-zip php-intl -y**

Вносим изменения в конфигурационный файл **/etc/php.ini** Изменим параметр **max\_input\_vars** . Этот параметр закомментирован, поэтому его можно добавить в конец файла или в отдельный файл в папке **php.d**.

**echo 'max\_input\_vars = 6000' >> /etc/php.d/01-demo.ini**

Перезапуск apache и PHP-FPM (Fast Process Manager)

**systemctl restart httpd php-fpm**

Установка СУБД MariaDB

**dnf install mariadb-server mariadb -y**

Запуск службы mysql и добавляем в автозагрузку:

**systemctl enable mariadb --now**

Запускаем встроенный инструмент настройки MariaDB:

## mysql\_secure\_installation

Устанавливаем пароль root для mariadb. Во всех остальных пунктах вводим Y (Yes)

```
Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@HQ-SRV sshuser]# |
```

Создание пользователя и базы данных с поддержкой русского языка

### mysql -u root -p

```
[root@HQ-SRV sshuser]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.11.11-MariaDB MariaDB Server

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)]> CREATE DATABASE moodledb DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER moodle@localhost IDENTIFIED BY 'P@ssw0rd';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> GRANT ALL ON moodledb.* TO 'moodle'@'localhost';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> quit;
Bye
[root@HQ-SRV sshuser]# |
```

Вводим следующую последовательность команд

```
CREATE DATABASE moodledb DEFAULT CHARACTER SET utf8;
CREATE USER moodle@localhost IDENTIFIED BY 'P@ssw0rd';
GRANT ALL ON moodledb.* TO 'moodle'@'localhost';
flush privileges;
quit;
```

Перезапуск mariadb

```
systemctl restart mariadb
```

Базу данных можно создать, любой другой известный инструмент.

Устанавливаем moodle. Заходим на сайт moodle.org. В пункте Downloads выбираем стабильную версию , копируем ссылку на архив и качаем архив на HQ-SRV с помощью команды wget.

```
wget https://packaging.moodle.org/stable405/moodle-
latest-405.tgz -P /tmp
```

download.moodle.org/releases/latest/

moodle.org Forums Documentation Downloads Demo Tracker Development Translation

Home / Latest release | Moodle downloads

### Latest release | Moodle downloads

Install Moodle on your own server (requires a web server with PHP and a database) by downloading one of the following packages or [obtaining Moodle via Git](#).

Alternatively, try Moodle on your personal computer with an installer package which includes all other software required to make it run (Apache, MySQL and PHP).

- Moodle installer package for Mac OS X
- Moodle installer package for Windows

Version	Information	.tgz	.zip
Moodle 4.5.3+ MOODLE_405_STABLE	This package is built every week with new fixes produced by our stable development process. It contains a number of fixes made since the 4.5.3 release and is usually a better	<a href="#">Download tgz</a>	<a href="#">Download zip</a>
Built Weekly 3 days 23 hours ago		70.9 MB 779 today	93.2 MB 1116 today
		<a href="#">[md5]</a> <a href="#">[sha256]</a>	<a href="#">[md5]</a> <a href="#">[sha256]</a>

Переходим в папку /tmp, распаковываем архив

**cd /tmp/ && tar -xzf /tmp/moodle-latest-405.tgz**

Переходим в папку с мудлом и переносим его папку апаха

**mv /tmp/moodle/{.,}\* /var/www/html/**

Устанавливаем права на /var/www/html

**chown -R apache:apache /var/www/html/**

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

**mkdir /var/www/moodledata**

Устанавливаем права на /var/moodledata

**chown -R apache:apache /var/www/moodledata**

Перезапускаем сервис Апах

**systemctl restart httpd**

```
[root@HQ-SRV var]# ls -la /var/www
итого 20
drwxr-xr-x. 5 root root 4096 апр 8 19:23 .
drwxr-xr-x. 21 root root 4096 апр 8 19:23 ..
drwxr-xr-x. 2 root root 4096 апр 8 15:38 cgi-bin
drwxr-xr-x. 65 apache apache 4096 апр 8 17:49 html
drwxr-xr-x. 2 apache apache 4096 апр 8 19:18 moodledata
[root@HQ-SRV var]#
```

Переходим в браузере на адрес сервера с мудлом

← → ↻ Не защищено 192.168.3.2/install.php

## Установка

Язык

Выберите язык

Сейчас необходимо выбрать язык ТОЛЬКО для сообщений во время установки. Язык сайта и пользовательских интерфейсов можно будет указать далее в процессе установки.

Язык Русский (ru) 1

[Далее > 2](#)

moodle

← → ↻ Не защищено 192.168.3.2/install.php ☆ 📄 👤 ⋮

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 (обычно 'www-data', 'nobody' или 'apache').

Этот каталог не должен быть доступен напрямую через Интернет.


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

Веб-адрес

Каталог Moodle

Каталог данных

« Назад »



← → ↻ Не защищено 192.168.3.2/install.php ☆ 📄 👤 ⋮

## Установка

### 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



На следующем окне вставляем параметры созданной базы

Название базы данных: **moodledb**

Пользователь базы данных: **moodle**

Пароль: **P@ssw0rd**

← → ↻ Не защищено 192.168.3.2/install.php ☆ 📄 👤 ⋮

Этот драйвер не совместим с устаревшей системой MySQL.

Сервер баз данных

Название базы данных

Пользователь базы данных


Пароль

Префикс имен таблиц

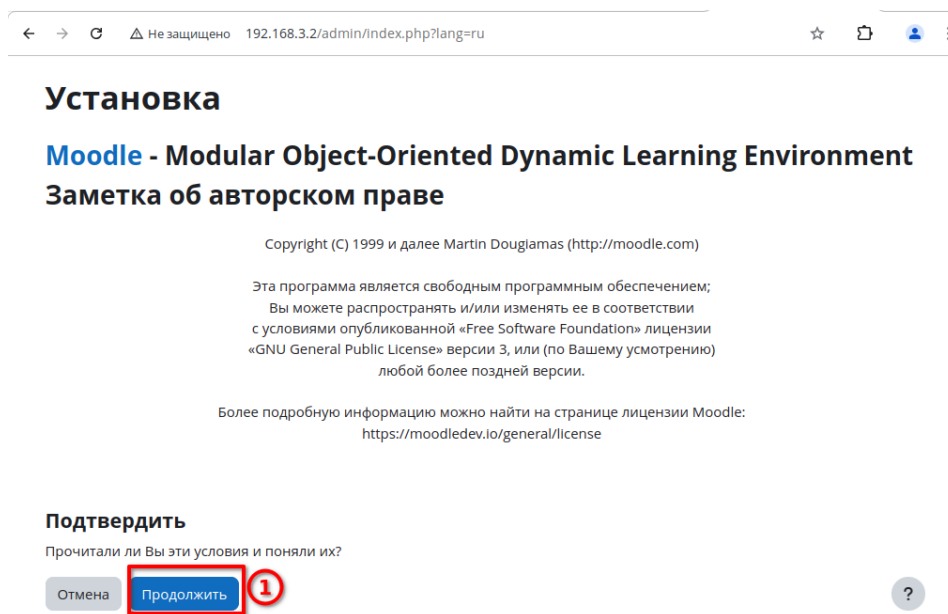
Порт базы данных

Подключение через Unix-сокеты

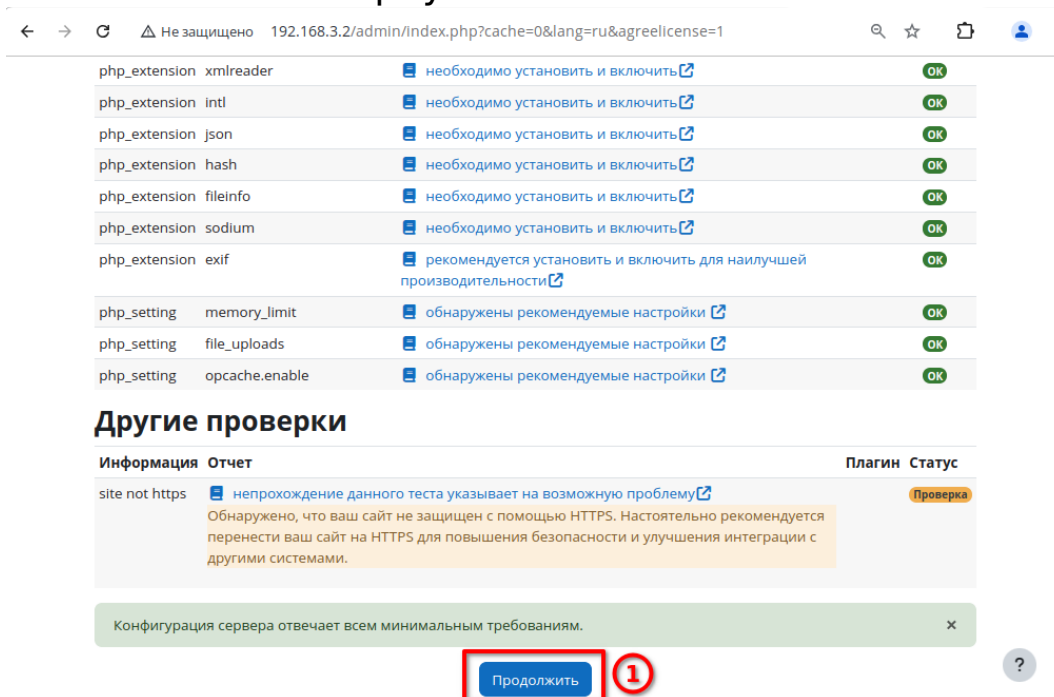
« Назад »







Проверяем настройки сервера. Если все ОК жмем **Продолжить** К настройке HTTPS можно вернуться позже



Примерно минут через 10 появится кнопка продолжить

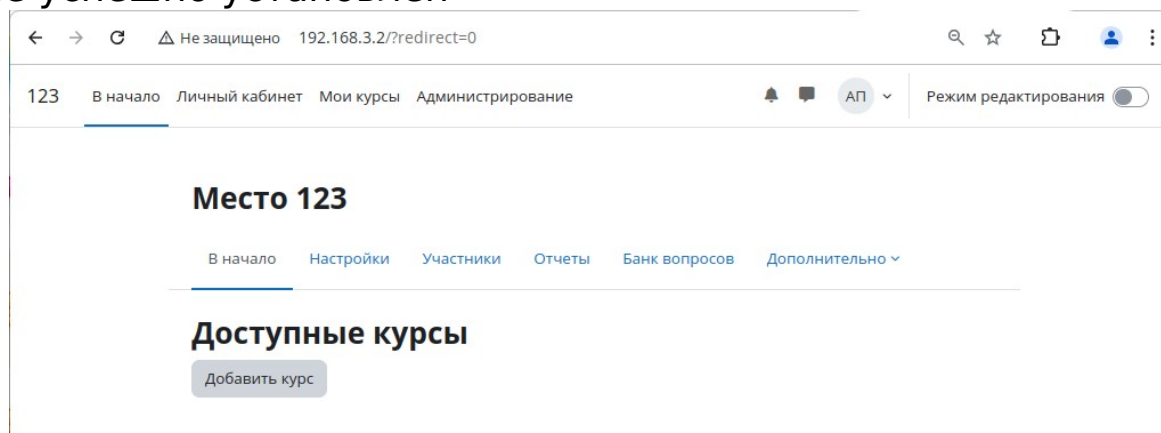


Заполняем необходимые сведения и нажимаем **Обновить профиль** (Ss12345#)

Вносим необходимые сведения и нажимаем **Сохранить изменения** :

Заполняем контакты службы поддержки пользователей и нажимаем **Сохранить изменения**

Moodle успешно установлен



## Задание 8 Настройте веб-сервер nginx как обратный прокси-сервер на HQ-RTR

- При обращении к HQ-RTR по доменному имени moodle.au-team.irpo клиента должно перенаправлять на HQ-SRV на стандартный порт, на сервис moodle
- При обращении к HQ-RTR по доменному имени wiki.au-team.irpo клиента должно перенаправлять на BR-SRV на порт, на сервис mediawiki.

### Выполнение

Заходим на HQ-RTR. Установка Nginx

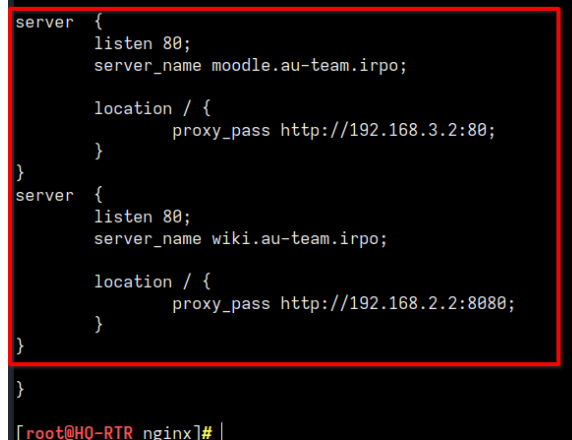
```
dnf install nginx -y
```

Запуск и добавление в автозагрузку

```
systemctl enable --now nginx
```

Открываем на редактирование конфигурационный файл **/etc/nginx/nginx.conf**. Перед последней фигурной скобкой добавляем следующие строки

```
server {  
    listen 80;  
    server_name moodle.au-team.irpo;  
    location / {  
        proxy_pass http://192.168.3.2:80;  
    }  
}  
server {  
    listen 80;  
    server_name wiki.au-team.irpo;  
    location / {  
        proxy_pass http://192.168.2.2:8080;  
    }  
}
```



```
server {  
    listen 80;  
    server_name moodle.au-team.irpo;  
  
    location / {  
        proxy_pass http://192.168.3.2:80;  
    }  
}  
server {  
    listen 80;  
    server_name wiki.au-team.irpo;  
  
    location / {  
        proxy_pass http://192.168.2.2:8080;  
    }  
}  
[root@HQ-RTR nginx]#
```

ПРОВЕРЯЕМ доступ к сайтам по прописанным адресам через браузер  
ВМ HQ-CLI.

на Moodle – **moodle.au-team.irpo**

на MediaWiki – **wiki.au-team.irpo**

Если не работает, то отключаем SELinux на HQ-RTR.

## Задание 9 Удобным способом установите приложение Яндекс Браузере для организаций на HQ-CLI

- Установку браузера отметьте в отчёте

### Выполнение

На HQ-CLI смотрим включенные репозитории

#### **dnf repolist all**

```
[admin@hq-cli2 ~]$ dnf repolist all
идентификатор репозитория      имя репозитория      состояние
base                           RedOS - Base          включено
r7-office                      Repositories for R7 Office  включено
updates                       RedOS - Updates       включено
yandex-browser-release         Repositories for Yandex Browser  включено
[admin@hq-cli2 ~]$
```

Репозиторий яндекс браузера включен по умолчанию в РЕД ОС 8.

Ищем как называется яндекс браузер

#### **dnf search yandex**

```
[admin@hq-cli2 ~]$ dnf search yandex
Repositories for R7 Office      1.1 MB/s | 764 kB      00:00
Repositories for Yandex Browser 9.1 kB/s | 6.5 kB      00:00
RedOS - Base                   11 MB/s | 66 MB        00:05
RedOS - Updates                9.5 MB/s | 62 MB        00:06
===== Имя и Краткое описание совпадение: yandex =====
yandex-browser-release.noarch : Repositories for Yandex Browser
yandex-browser-stable.x86_64 : Yandex Browser
yandex-disk-indicator.noarch : Panel indicator for YandexDisk CLI client for Linux
[admin@hq-cli2 ~]$
```

Устанавливаем

**sudo dnf install yandex-browser-stable -y**