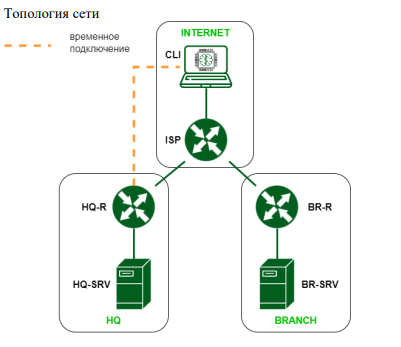
**DEMO 2024**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя ВМ** | **NIC** | **IPv4-адреса** | **IPv6-адреса** |
| HQ-R | ens192 | 10.2.0.2/24 | 2001:2::2/64 |
| ens224 | 192.168.100.62/26 | 2001:100::3f/122 |
| ens256 | 10.4.0.1/24 | 2001:4::1/64 |
| BR-R | ens192 | 10.3.0.2/24 | 2001:3::2/64 |
| ens224 | 192.168.200.14/28 | 2001:200::f/124 |
| HQ-SRV | ens192 | 192.168.100.1/26 | 2001:100::1/122 |
| BR-SRV | ens192 | 192.168.200.1/28 | 2001:200::1/124 |
| ISP | ens192 | 10.1.0.1/24 | 2001:1::1/64 |
| ens224 | 10.2.0.1/24 | 2001:2::1/64 |
| ens256 | 10.3.0.1/24 | 2001:3::1/64 |
| **ens161** | **DHCP** | **настроен** |
| CLI | ens192 | 10.1.0.2/24 | 2001:1::2/64 |
| ens224 | 10.4.0.2/24 | 2001:4::2/64 |

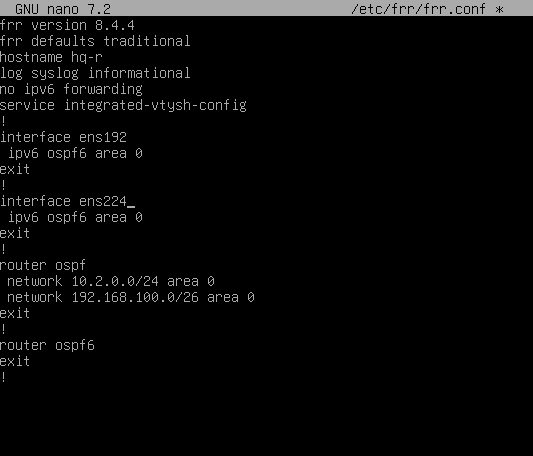
МОДУЛЬ 1

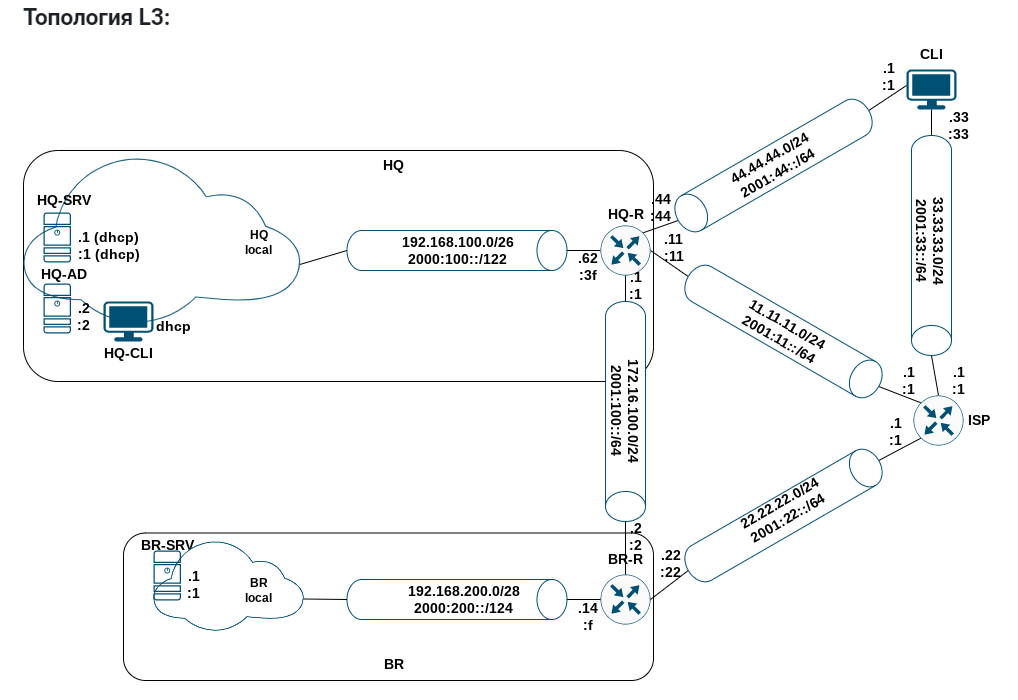
# 1 Выполните базовую настройку всех устройств:

* Присвоить имена в соответствии с топологией
* Раскидать ip адреса
* Открыть forwarding на HQ-R и BR-R

# 2. FRR. OSPF

Решение:

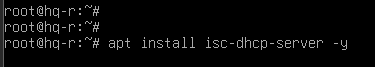


a. Составьте топологию сети L3.

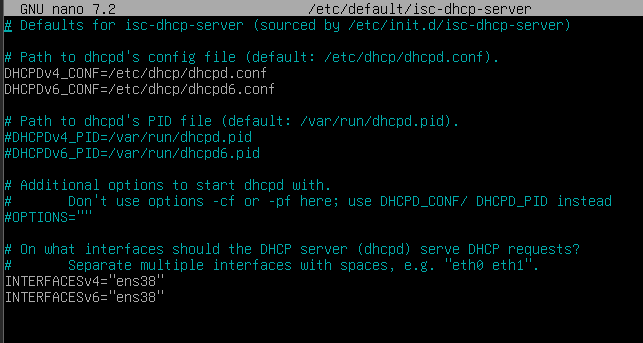
# 3. DHCP

Решение:

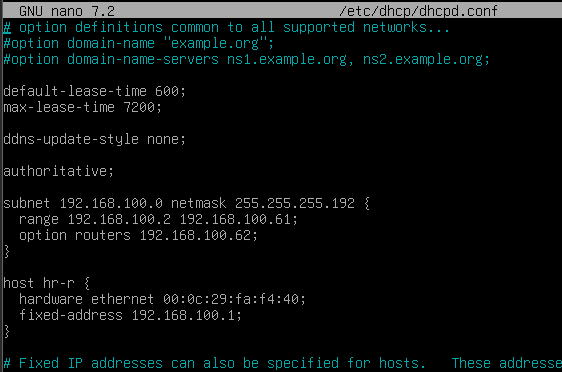
**HQ-R**



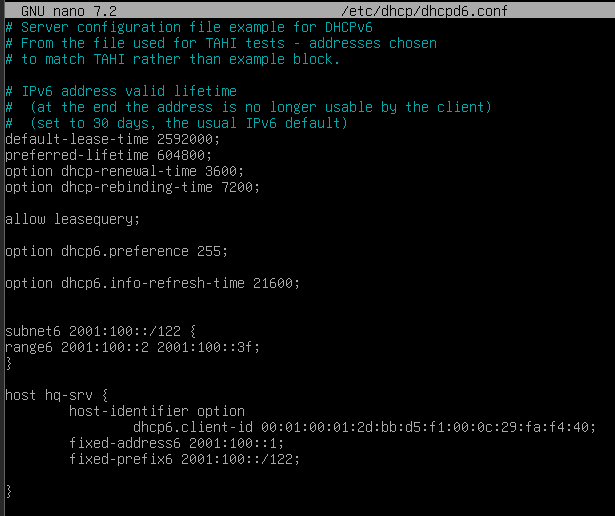
Заходим в конфиг и открываем адресацию



Заходим dhcpd



Заходим dhcpd6



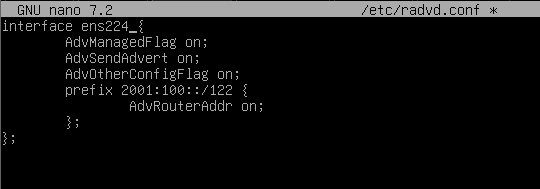
Перезагружаем службу



Настраиваем radvd

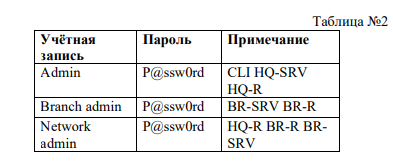


Настройка



Перезагрузить службу

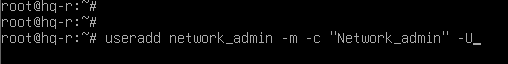
# 4. Создание локальные учётные записи на хостах

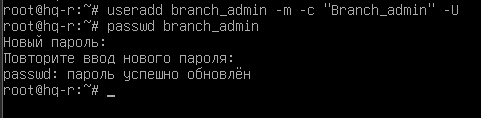


Решение:

Создаем пользователей командой:







# 5. Измерить пропускную способность сети HQ-R-ISP iperf 3

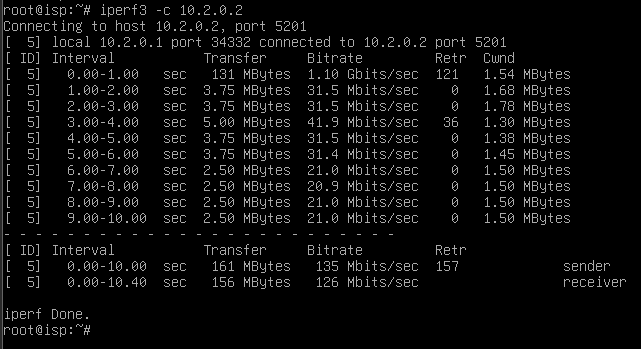
Решение:

**ISP/HQ-R**

****

**ISP**

Проверка

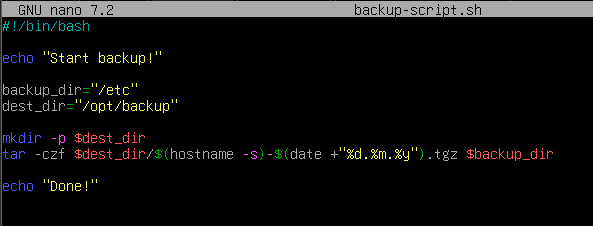


# 6. backup скрипты для сохранения конфигурации на HQ-R BR-R.

Решение:

**HQ-R/BR-R**

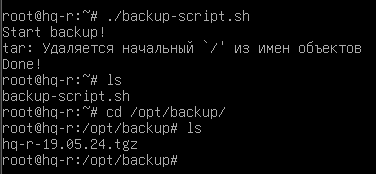
Заходим по пути nano backup-script.sh



Выдаем права



Запускаем скрипт



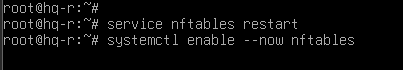
# 7. Настройте подключение по SSH

Решение:

**HQ-R**

правило для проброса портов в цепочку

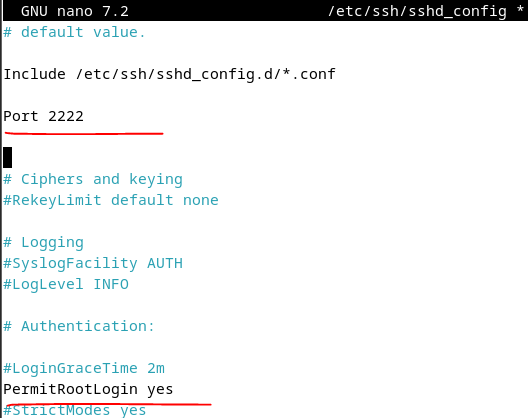
****

****

**HQ-SRV**

apt install openssh-server -y

Открываем порт



**BR-R**

Проверяем подключение

**ssh admin@10.2.0.2**

# 8. Контроль по SSH со всех устройств, кроме CLI.

Решение:

**HQ-SRV**



**CLI**

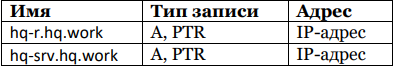
Проверяем подключение

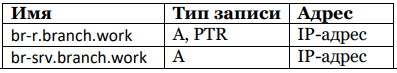


# МОДУЛЬ 2

# 1. Настройте DNS-сервер на сервере HQ-SRV

Зона hq.work и branch.work, также не забудьте настроить обратную зону.





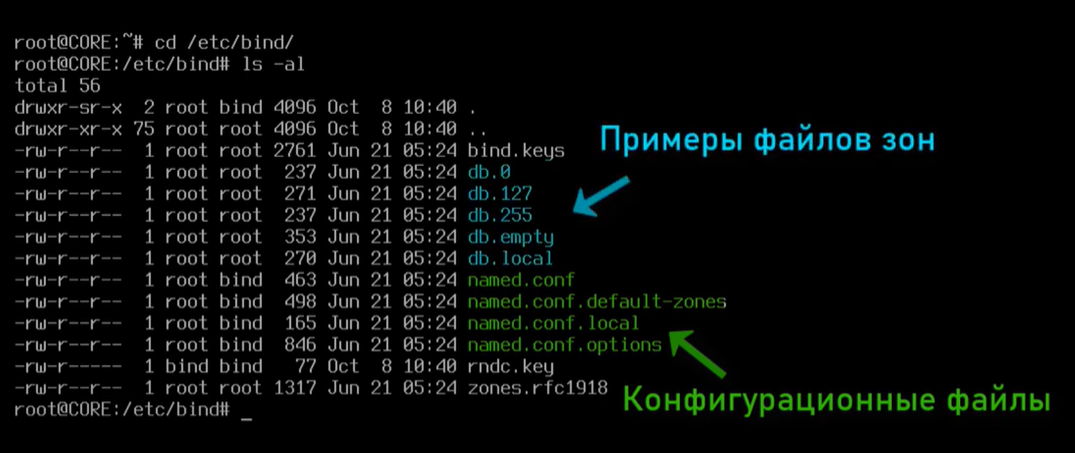
Решение:

**HQ-SRV**

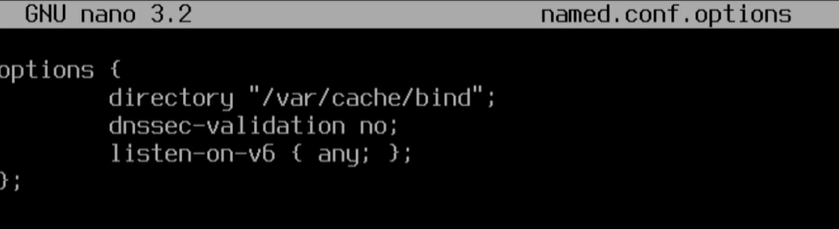
Вводим командy:

apt install bind9 -y

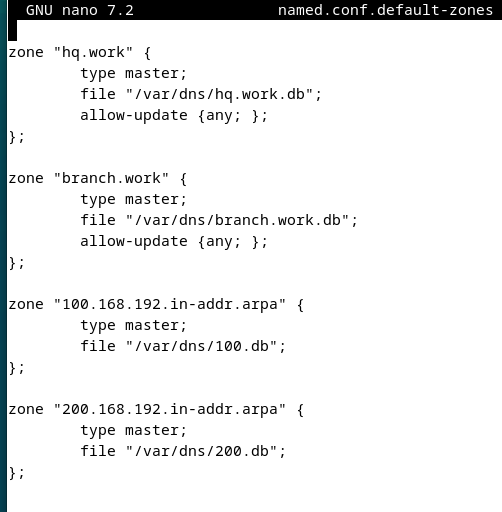
переходим в директорию по пути: cd /etc/bind/



Переходим в nano named.conf.options



Переходим в named.conf.default-zones



Проверка на ошибки:

**named-checkconf**

вводим:

**mkdir /var/dns/**

Скопировать файлы зон

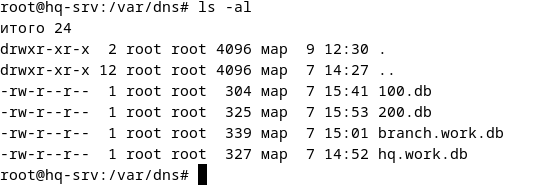
**cp db.local /var/dns/hq.work.db**

**cp db.local /var/dns/branch.work.db**

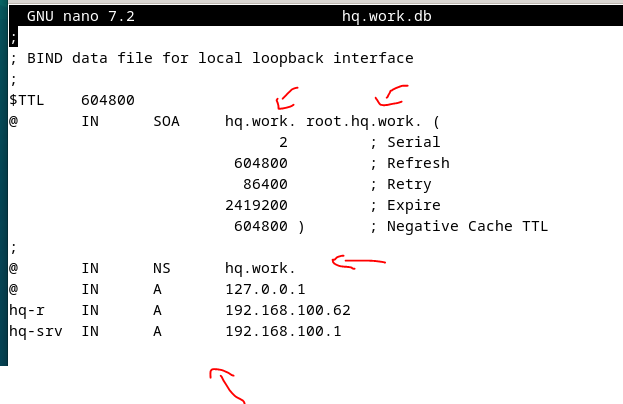
**cp db.127 /var/dns/100.db**

**cp db.127 /var/dns/200.db**

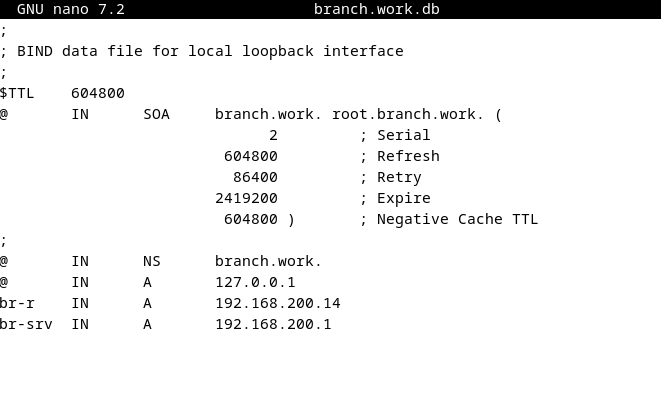
Проверяем



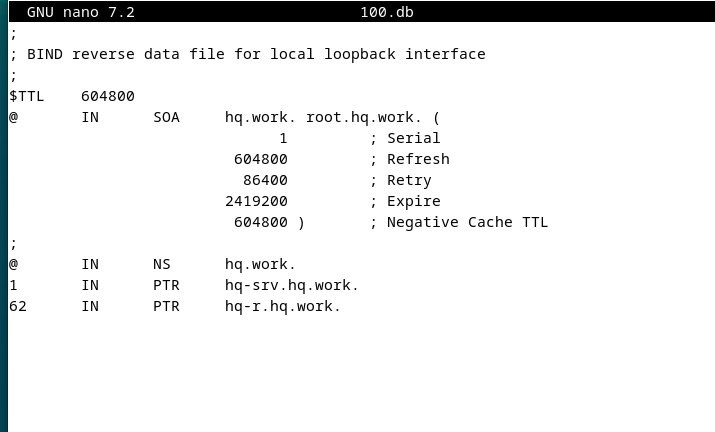
Переходим в nano hq.work.db



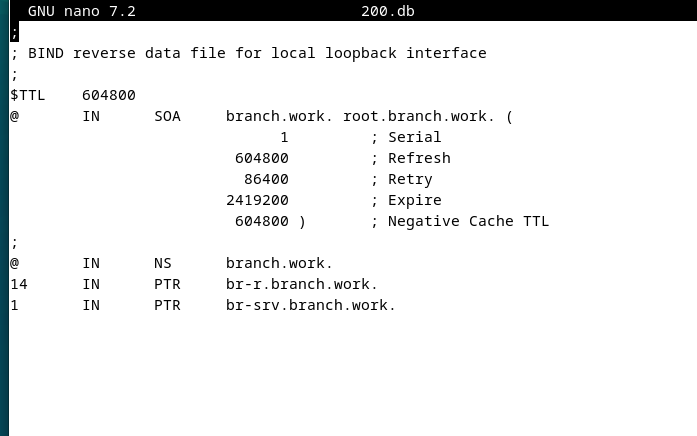
Переходим в nano branch.work.db



Переходим в nano 100.db

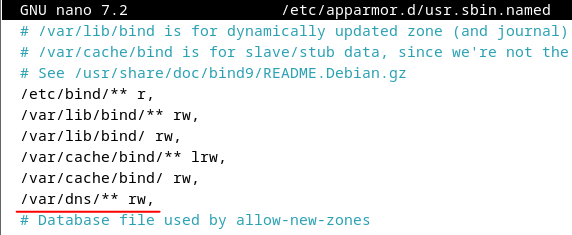


Переходим в nano 200.db

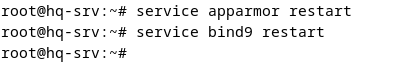


Перезагружаем службу: service bind9 restart

Выдаем права на созданный каталог



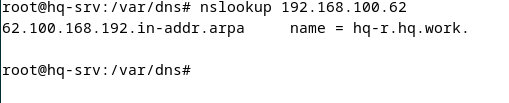
Перезагружаем службы:



Прописываем domain и nameserver



Командой Nslookup проверяем как у нас отрабатывают PTR записи



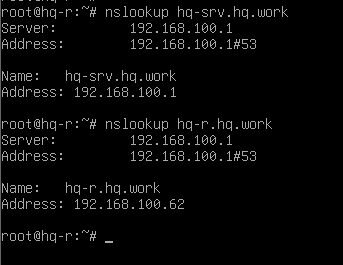
**Клиентские машины**

**HQ-R, BR-R, BR-SRV**

Указываем Ip адрес днс сервера



Проверяем:



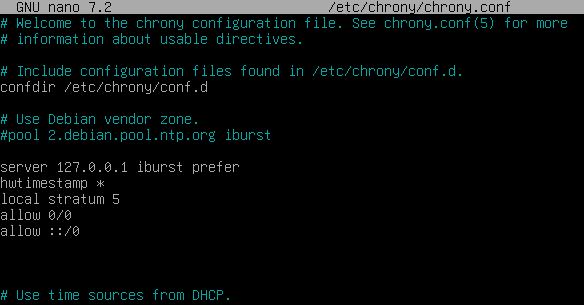
# 2. Настройте синхронизацию времени между сетевыми устройствами по протоколу NTP.

Решение:

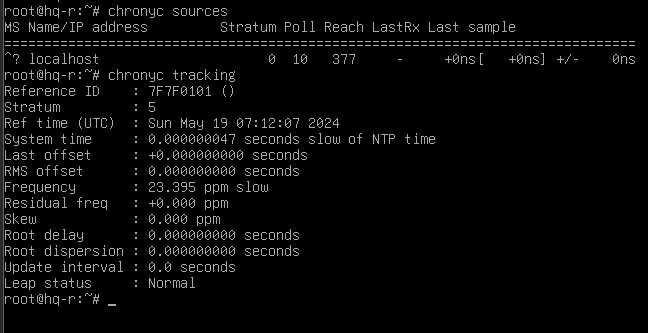
**HQ-R**

****

Меняем конфиг

****

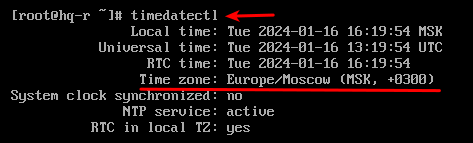
Проверяем:



Переводим на московское время:

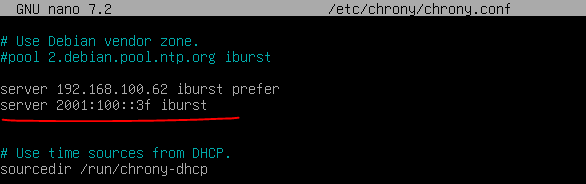


Проверяем:



На всех остальных машинах скачиваем пакет chrony

Добавляем сервер

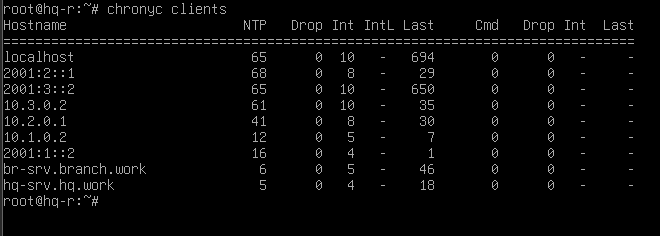


И прописываем команду



**HQ-R**

**Проверяем:**



# 3. Настройте сервер Домена на базе HQ-SRV

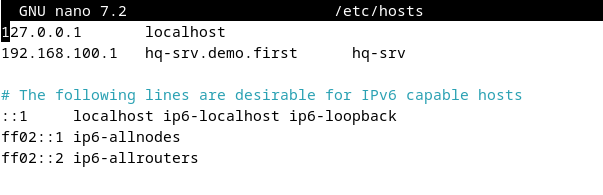
Решение:

**HQ-SRV**

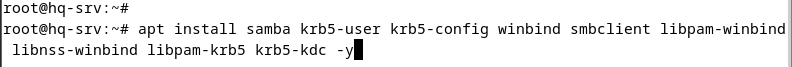
Настраиваем адрес сервера имен



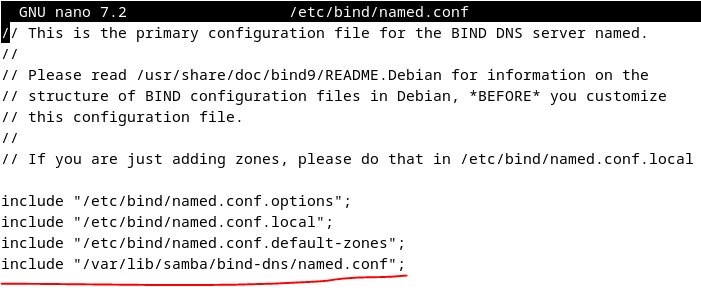
Настройка hosts



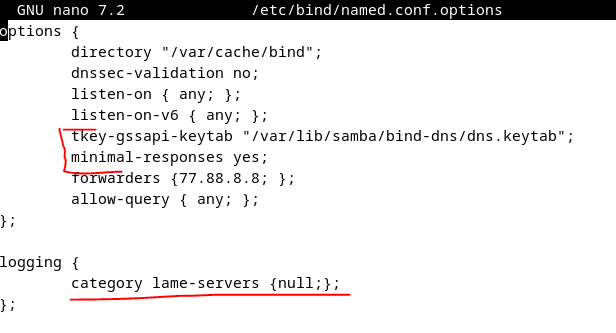
Скачиваем все необходимые пакеты



Заходим в named.conf и добавляем include



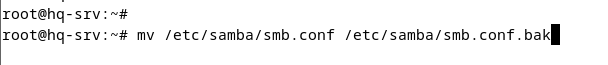
Заходим по пути nano /etc/bind/named.conf.options



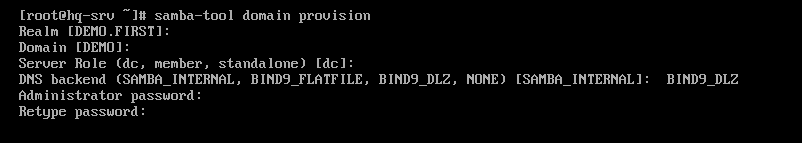
Выполнить остановку bind9



Бэкапим файл с исходными настройками Samba



Запуск интерактивной установки



Вернуть его назад с новыми настройками





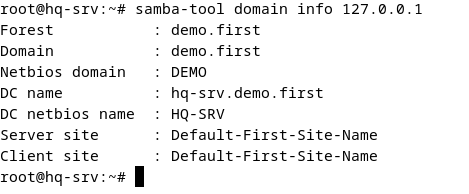
Перезагрузим самбу:



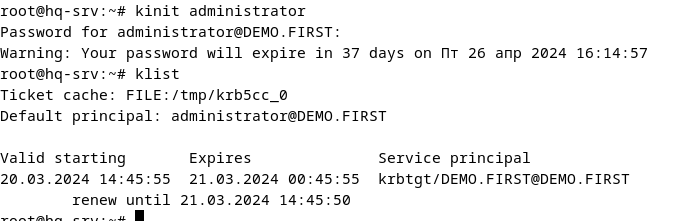
Проверка

named-checkconf -z

Убеждаемся, что всё работает



Проверяем работоспособность Kerberos:



Создадим пользователей и группы

Samba-tool user create Admin

Samba-tool user create Branch admin

Samba-tool user create Network admin

Samba-tool group create Admins

Samba-tool group create Network Admins

Samba-tool group create Branch Admins

Samba-tool group addmembes “Admins” “Admin”

Samba-tool group addmembers “Network admins” “Network admin”

Samba-tool group addmembers “Branch admins” “Branch admin”

Команды на проверку

wbinfo -u

wbinfo -g

**Клиентские машины**

**BR-SRV/CLI**

Заходим в nano /etc/resolv.conf

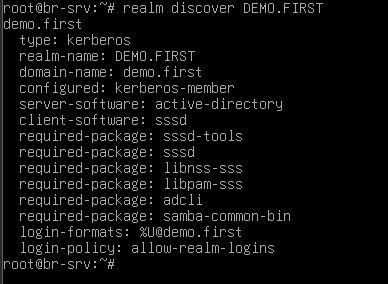


Устанавливаем:



Проверяем: realm join -U administrator demo.first





# 4. Реализуйте файловый SMB сервер на базе сервера HQ-SRV

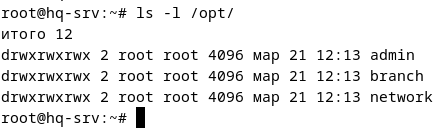
Решение:

Создаём директории для общих папок:

**mkdir /opt/{branch,network,admin}**

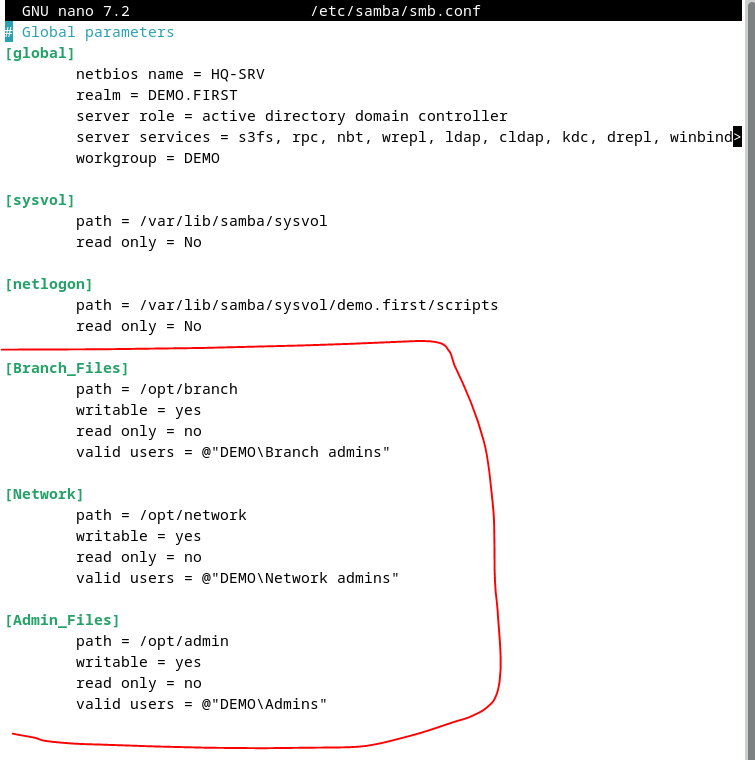
Задаём права на созданные директории:

**chmod 777 /opt/{branch,network,admin}**



Описываем общие папки для публикации по названиям в конфигурационном файле:

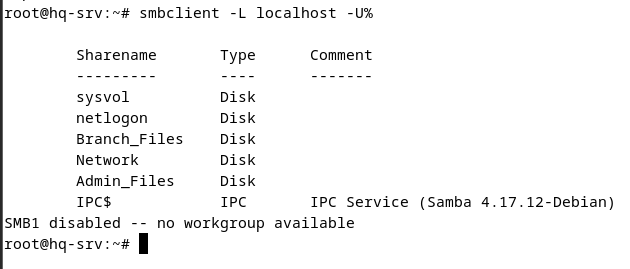
**nano /etc/samba/smb.conf**



Перезапускаем службу samba:

systemctl restart samba

Проверяем:

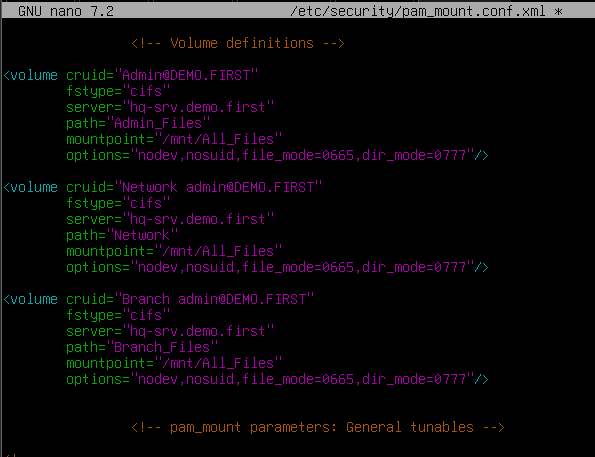


Настраиваем автоматическое монтирование соответствующих каталогов для доменных пользователей

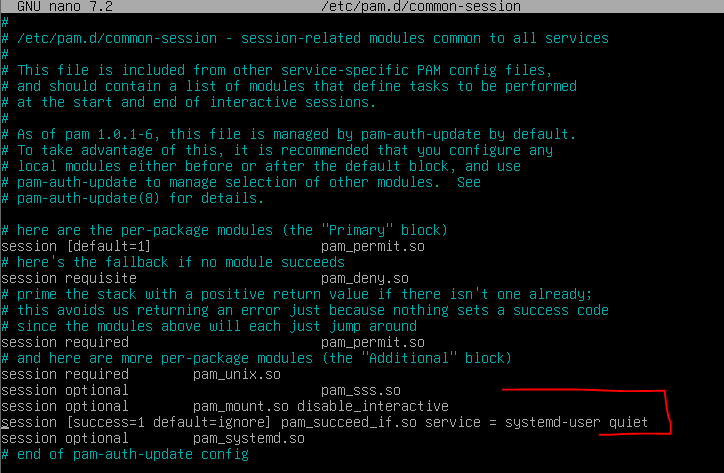
**BR-SRV**

****

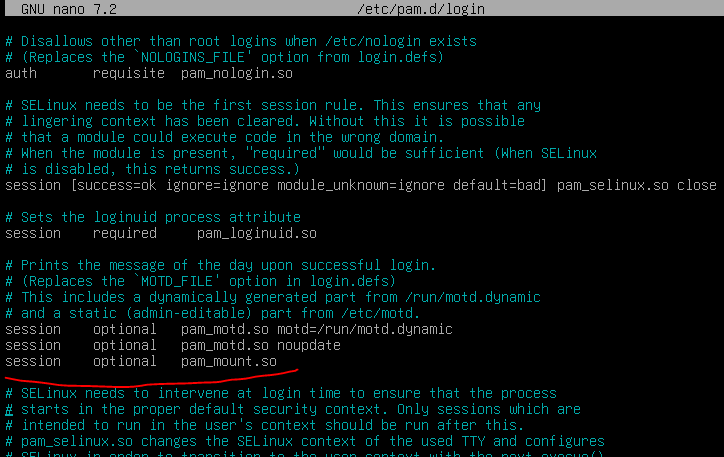
Заходим по пути и добавляем строки:



Далее заходим по пути: nano /etc/pam.d/common-session



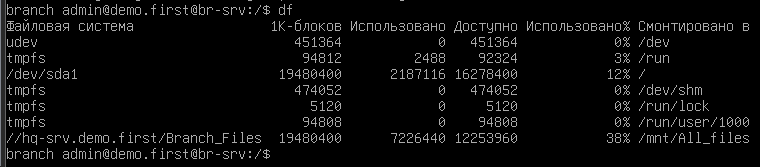
Далее заходим по пути nano /etc/pam.d/login и добавляем строчку



Далее приводим файл krb5 к следующему виду

****

Вводим команду: login branch [admin@DEMO.FIRST](mailto:admin@DEMO.FIRST)



# Cp 5. Сконфигурируйте веб-сервер LMS Apache на сервере BR-SRV

Решение:

**BR-SRV**

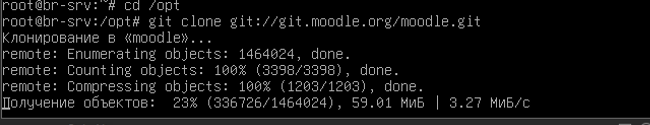
apt install -y apache2 default-mysql-server default-mysql-client php8.2 libapache2-mod-php graphviz aspell ghostscript clamav php8.2-pspell php8.2-curl php8.2-gd php8.2-intl php8.2-mysql php8.2-xml php8.2-xmlrpc php8.2-ldap php8.2-zip php8.2-soap php8.2-mbstring git -y

Добавляем в автозагрузку:



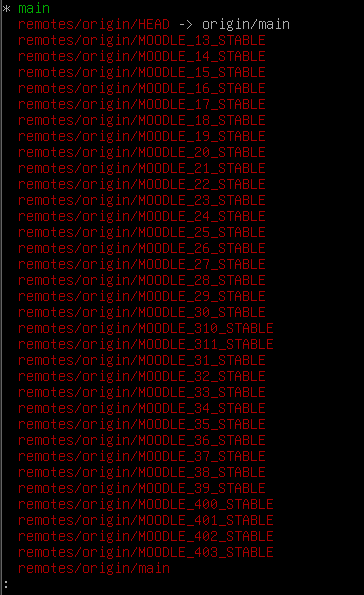
Переходим в каталог cd /opt и клонируем данные на наш сервер:

git clone git://git.moodle.org/moodle.git



Переходим в папку куда мы будем скачивать нужные файлы.

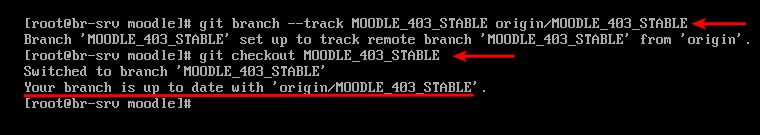




Вводим: git branch --track MOODLE\_403\_STABLE origin/MOODLE\_403\_STABLE

Проверяем:

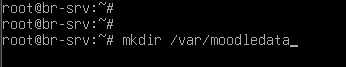
git checkout MOODLE\_403\_STABLE



Копируем каталог куда мы скачивали все данные



Создадим папку, где будут храниться данные moodle:



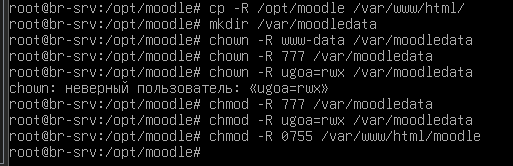
Выдаем права:

chown -R www-data /var/moodledata

chmod -R 777 /var/moodledata

chmod ugoa=rwx /var/moodledata

chmod -R 0755 /var/www/html/moodle

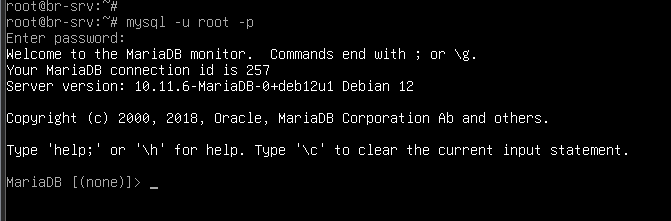


Описываем конфигурационный файл для веб-сервера Apache:

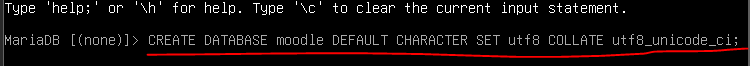
****

**Создание Базы Данных**

mysql -u root -p



Вводим:

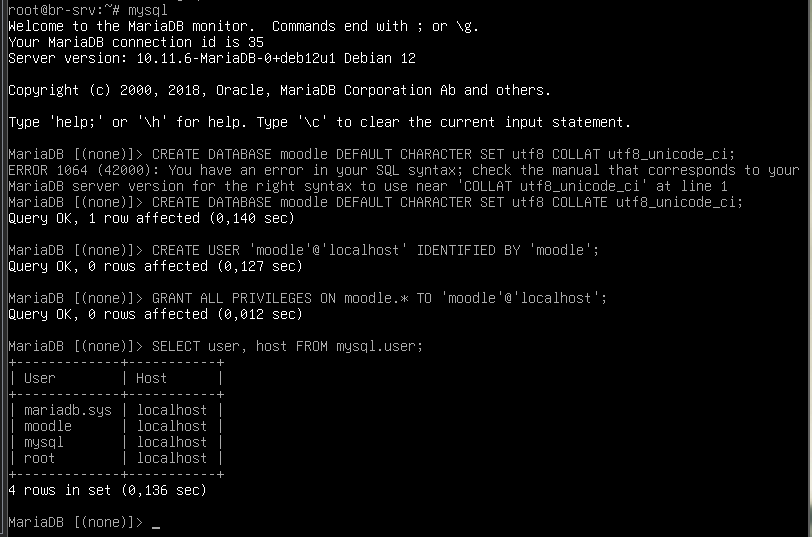






Проверка:

* SELECT user, host FROM mysql.user;
* EXIT



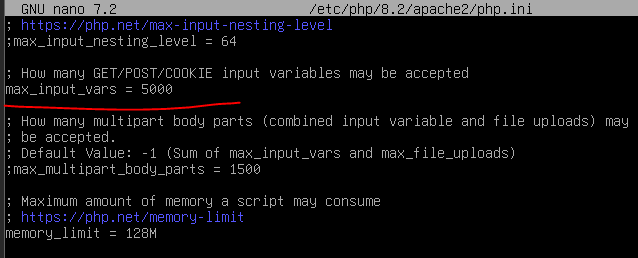
Выдаем права на доступ к папке moodle:

chmod -R 777 /var/www/html/moodle

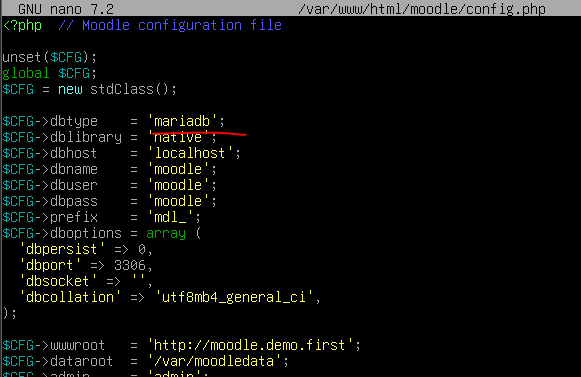
Перезапускаем веб-сервер:

service apache2 restart

Меняем параметр, (используя быстрый поиск vars)



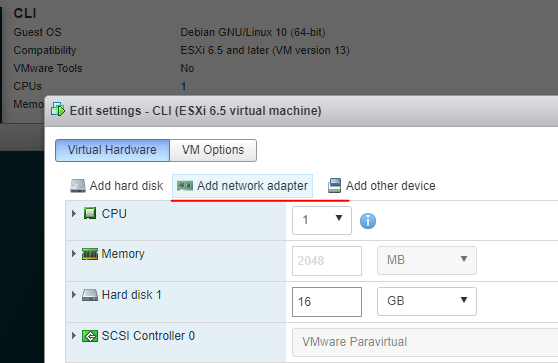
Заходим в конфиг php и меняем строчку на mariadb



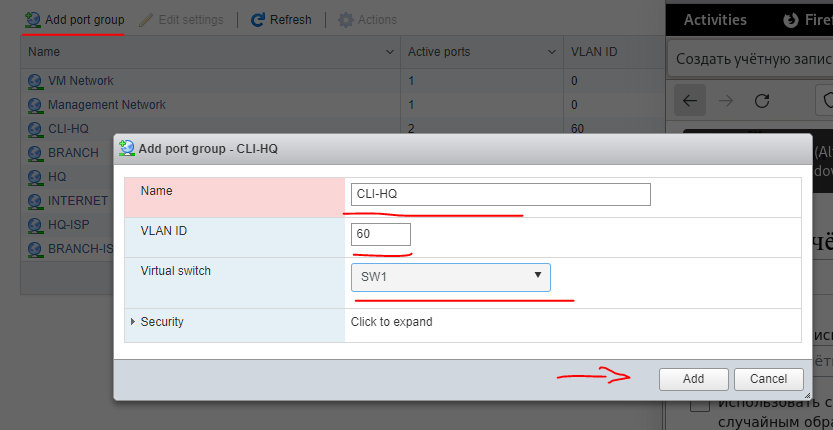
**CLI:**

Создаем временное подключение

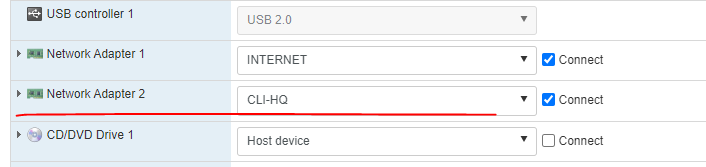
Добавляем по 1 интерфейсу на cli и на hq-r



Прописываем имя сети и выдаем влан



Выбираем сеть на интерфейсе



И в nano /etc/network/interfaces выдаем адреса новой сети (10.4.0.0.24)

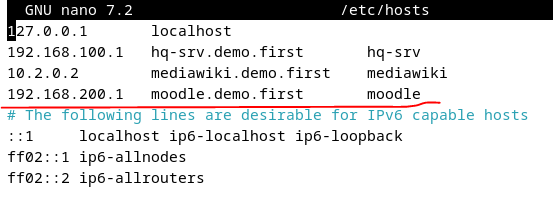
**HQ-R**

Добавляем новую сеть и интерфейс в ospf

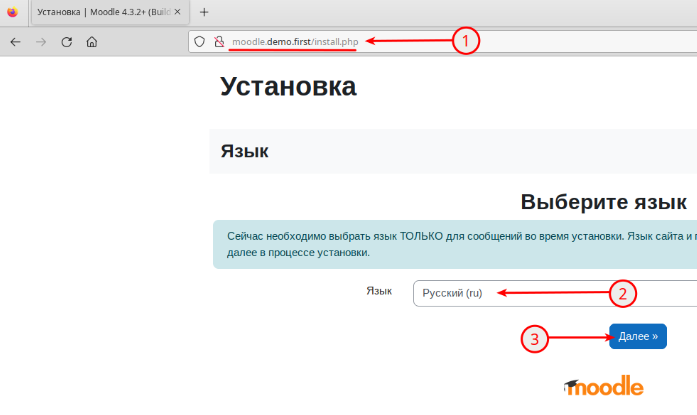


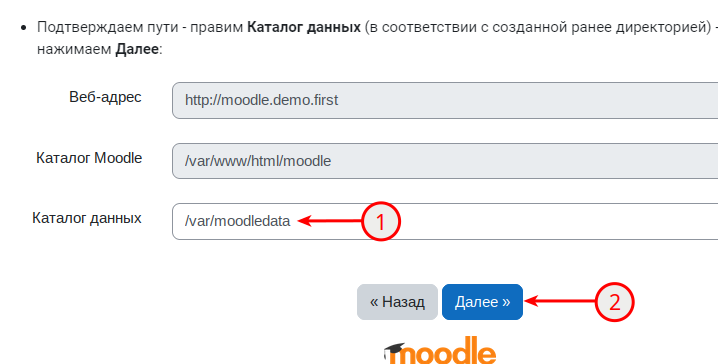
**CLI**

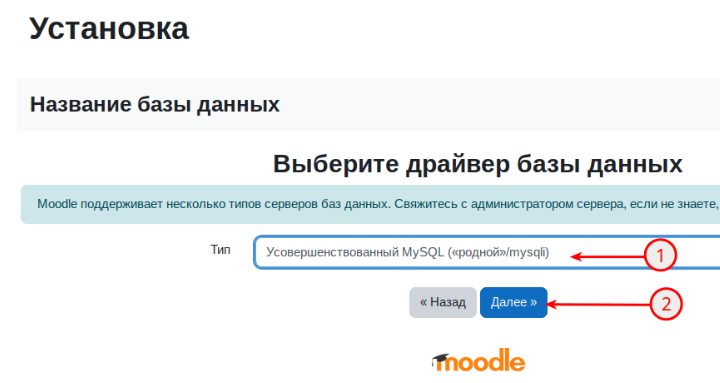
Добавляем:

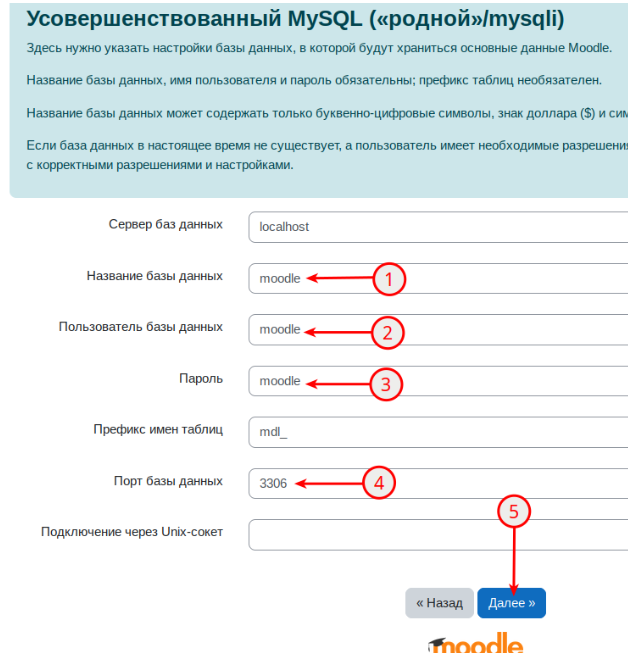
****

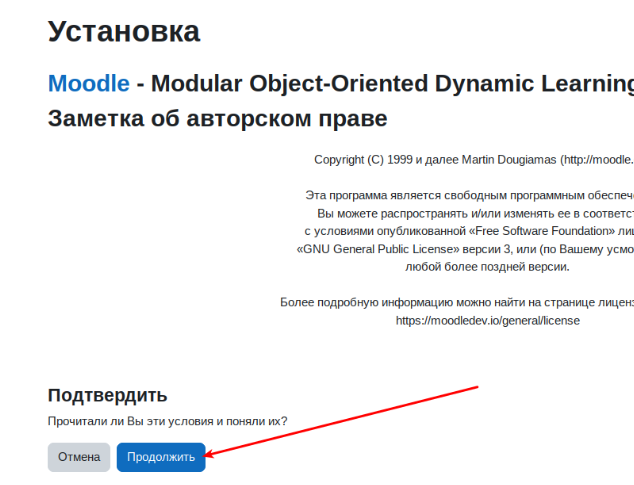
переходим на http://moodle.demo.first

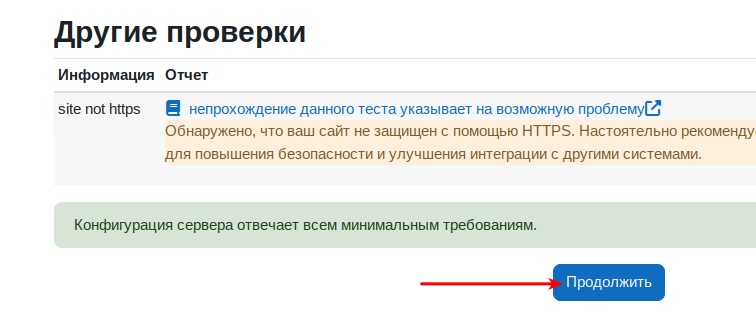


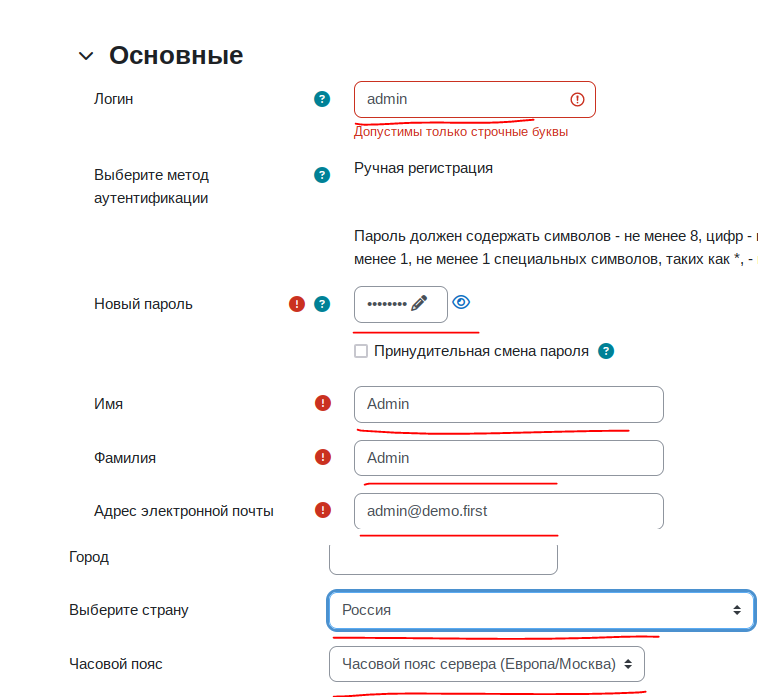


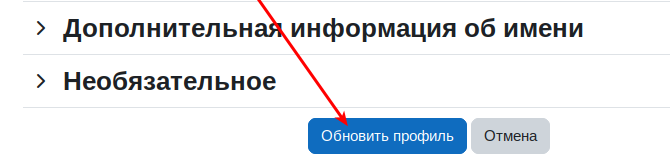


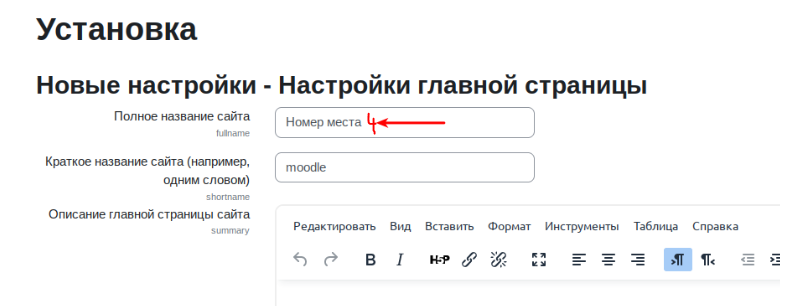




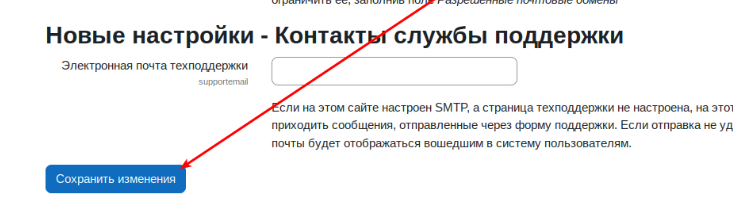


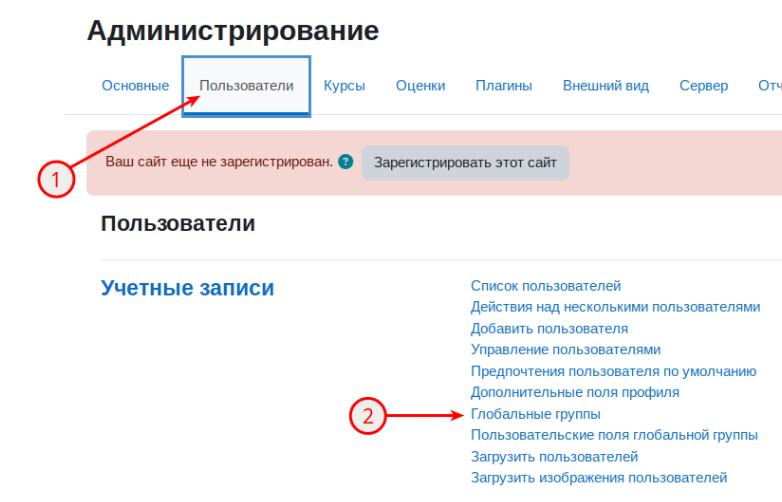


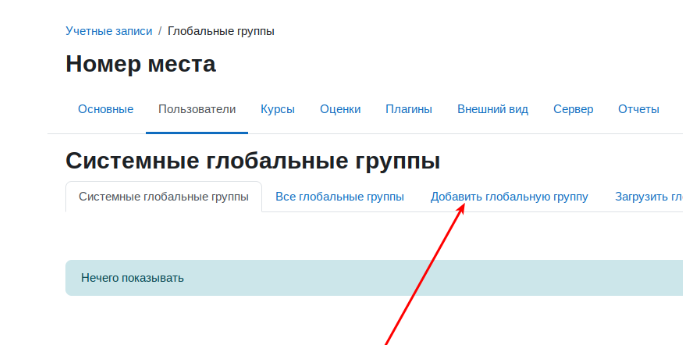




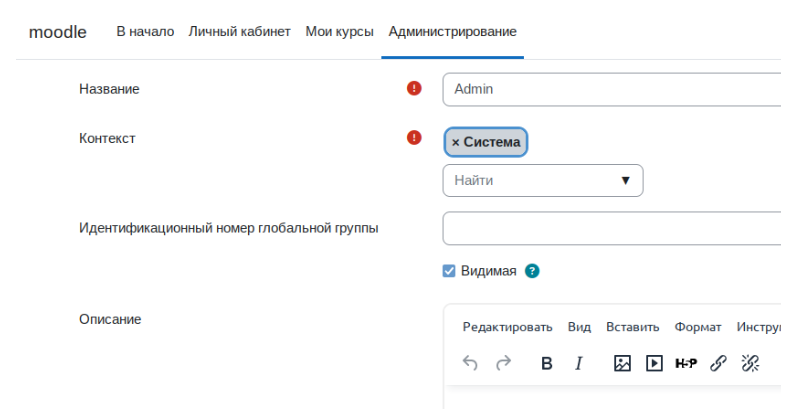
Прописываем техподдержку admin@demo.first

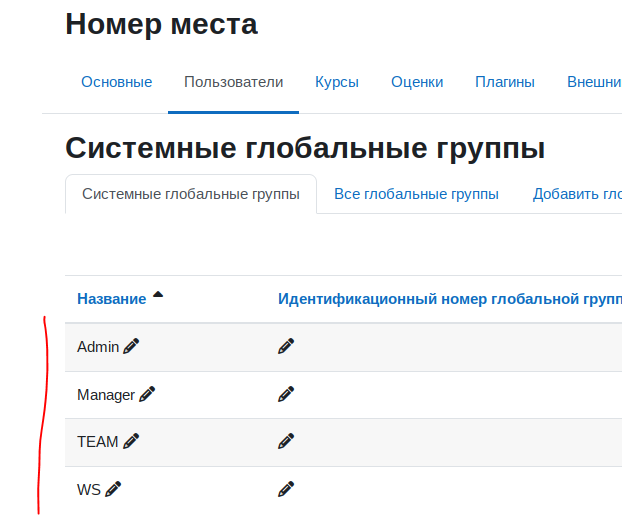




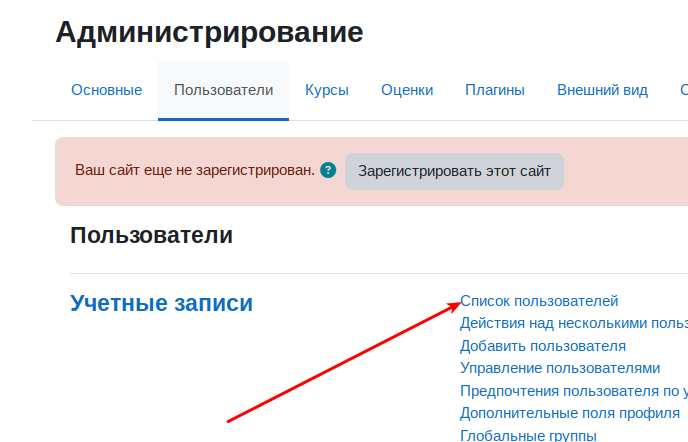


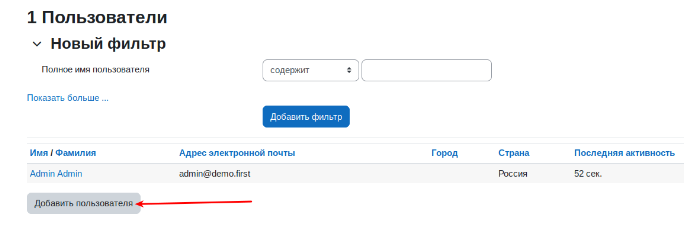
Создаем 4 группы (admin, manager, TEAM, WS)

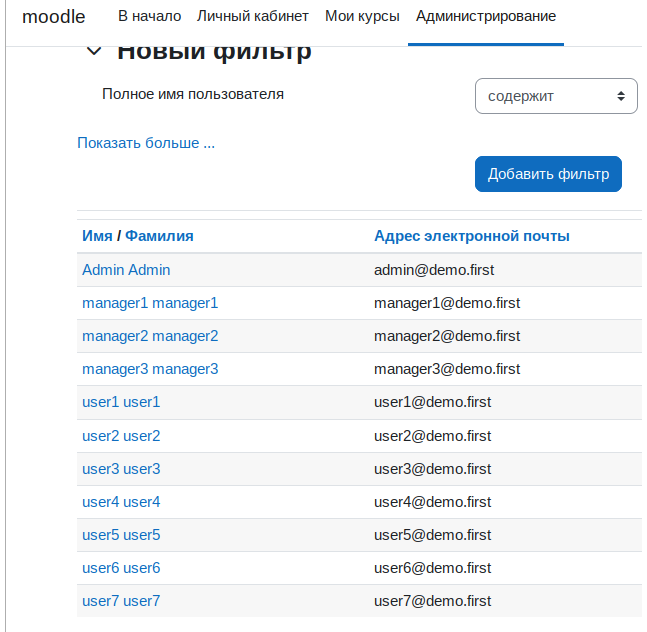




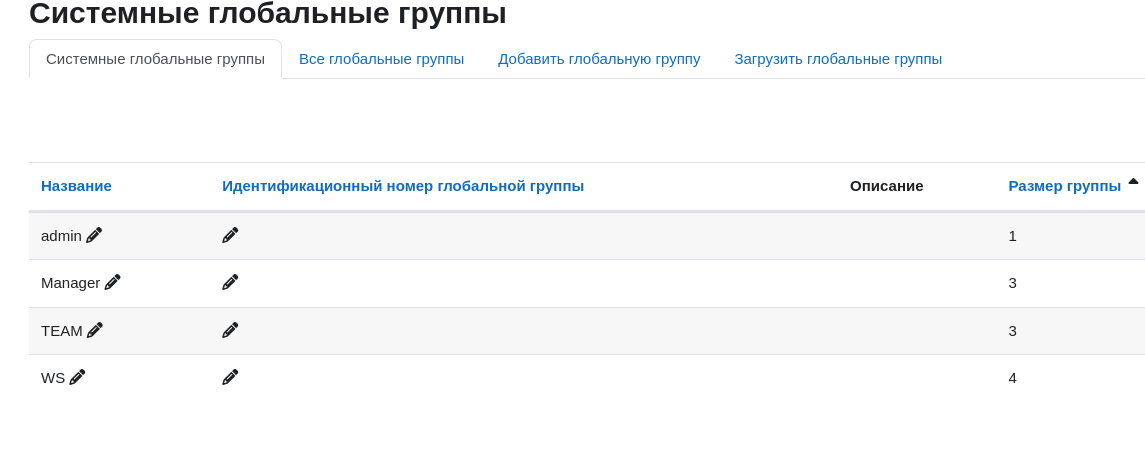
И создаем пользователей согласно заданию







И закидываем пользователей в группы согласно заданию



# 6. Запустите сервис MediaWiki используя docker

Решение:

**HQ-SRV:**

необходимые пакеты, чтобы разрешить apt использовать репозиторий через HTTPS:

**apt install apt-transport-https ca-certificates curl software-properties-common gnupg2 -y**

Затем добавьте официальный ключ GPG Docker:

**curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add –**

Добавьте репозиторий Docker в источники APT:

**echo "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list**

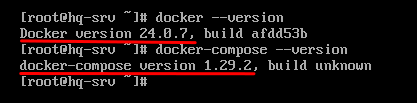
**apt install docker-ce -y**

**curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose**

Установите соответствующие разрешения:

**chmod +x /usr/local/bin/docker-compose**

Проверяем:

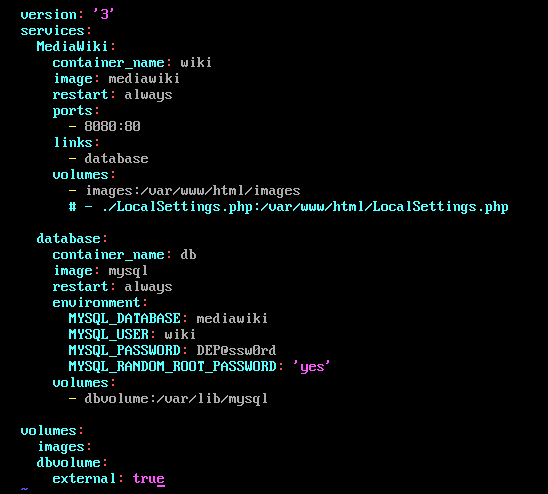
****

Включаем и добавляем в автозагрузку службу docker:

**systemctl enable --now docker.service**

В домашней директории пользователя root создаём файл wiki.yml со следующим содержимым:

**nano ~/wiki.yml**



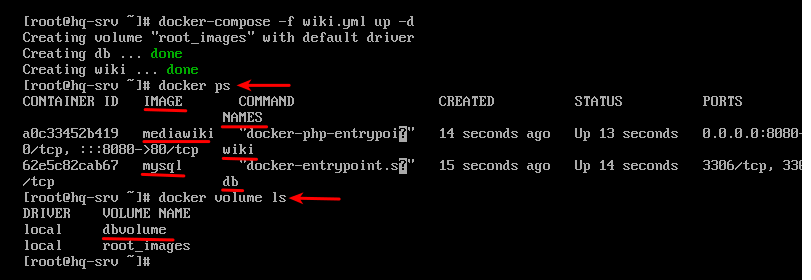
Чтобы отдельный volume для хранения базы данных имел правильное имя - создаём его средствами docker:

**docker volume create 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:



MediaWiki должна быть доступна извне через порт 8080

**HQ-R:**

Заходим по пути /etc/nftables.conf

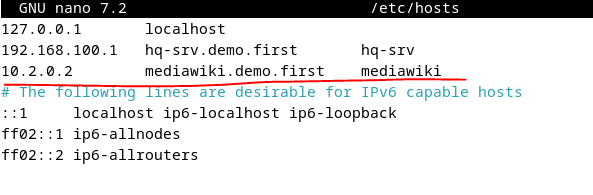
Прописываем:



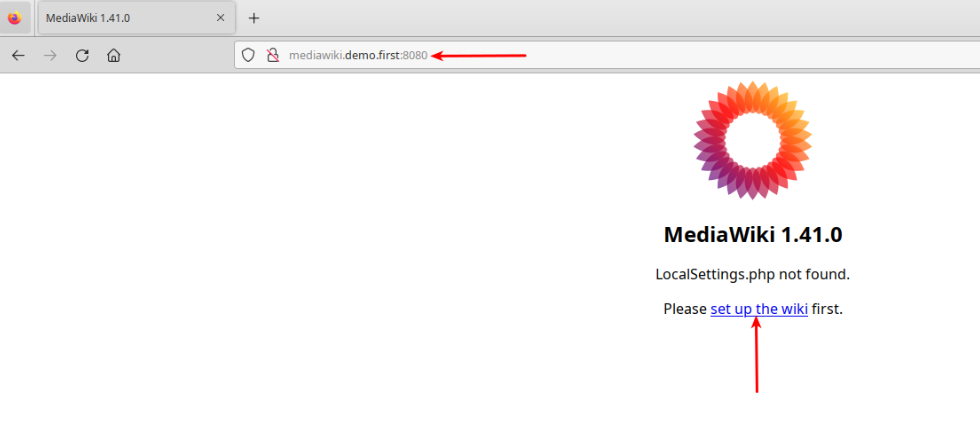
После можно переходить в браузер для дальнейшей установки MediaWiki.

**CLI:**

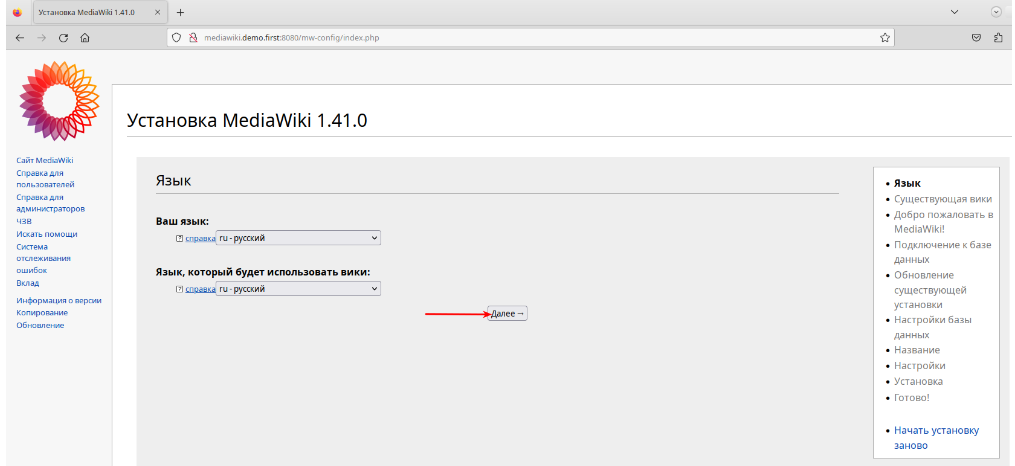
Добавляем запись в файл /etc/hosts:



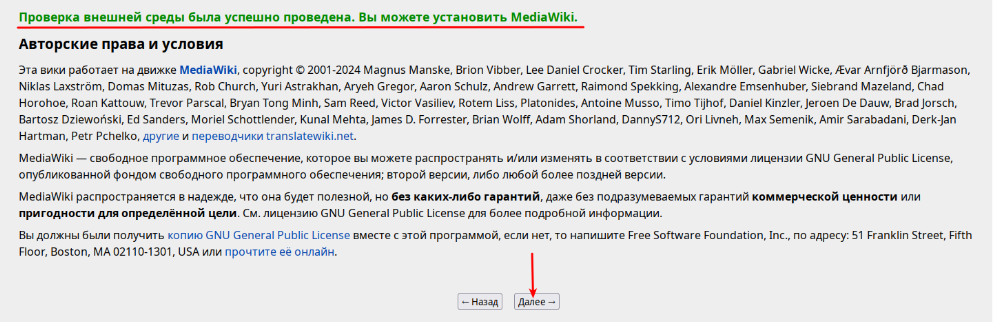
Переходим в браузер http://mediawiki.demo.first:8080 для продолжения установки через веб-интерфейс - нажимаем set up the wiki:



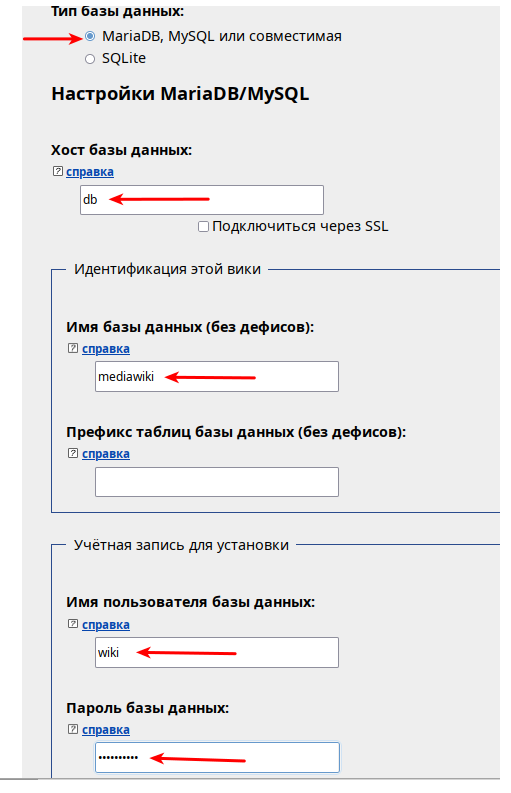
Выбираем необходимый Язык - нажимаем Далее:



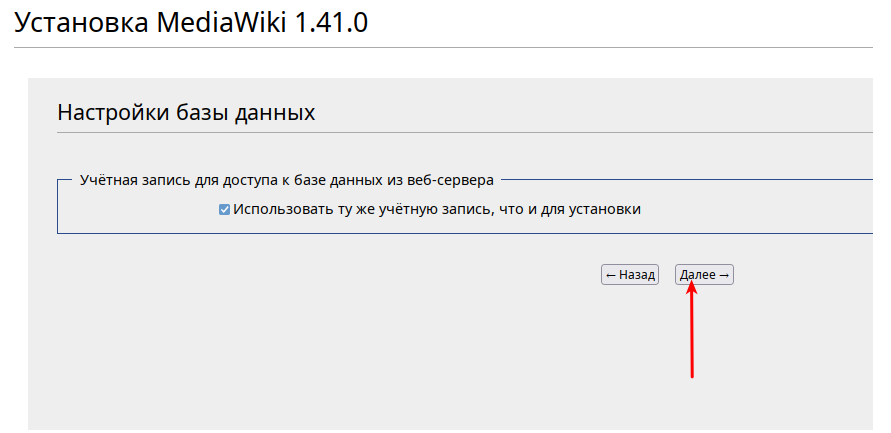
После успешной проверки внешней среды - нажимаем Далее:



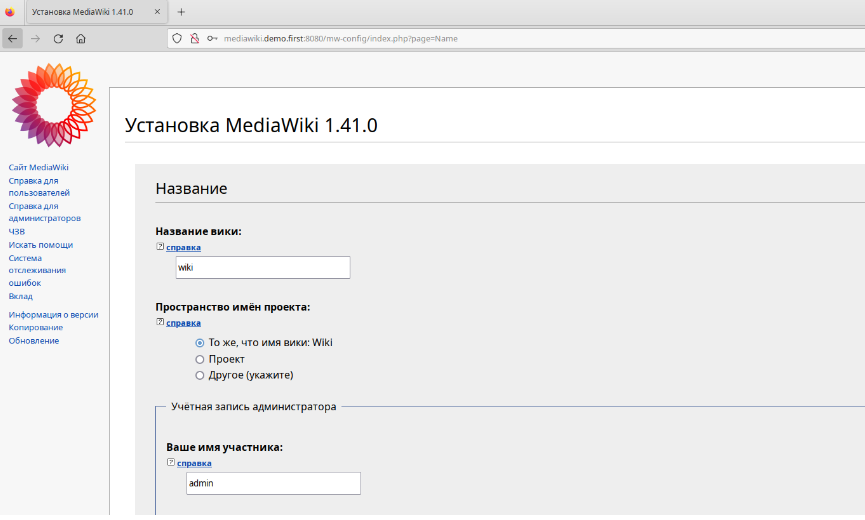
Заполняем параметры подключение к Базе Данных в соответствие с заданными переменными окружения в wiki.yml, которые соответствуют требованиям задания:

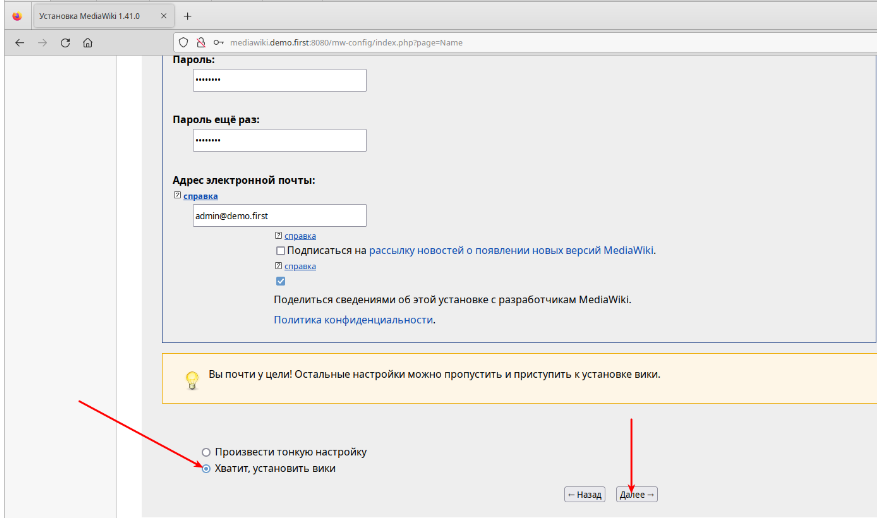


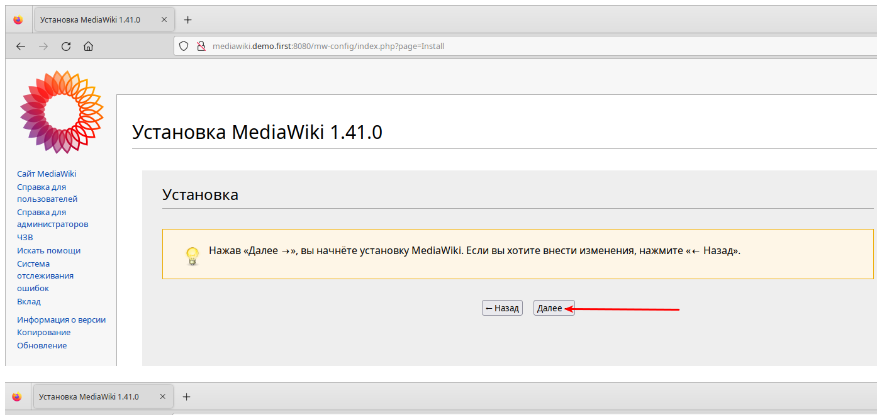
Нажимаеем Далее:

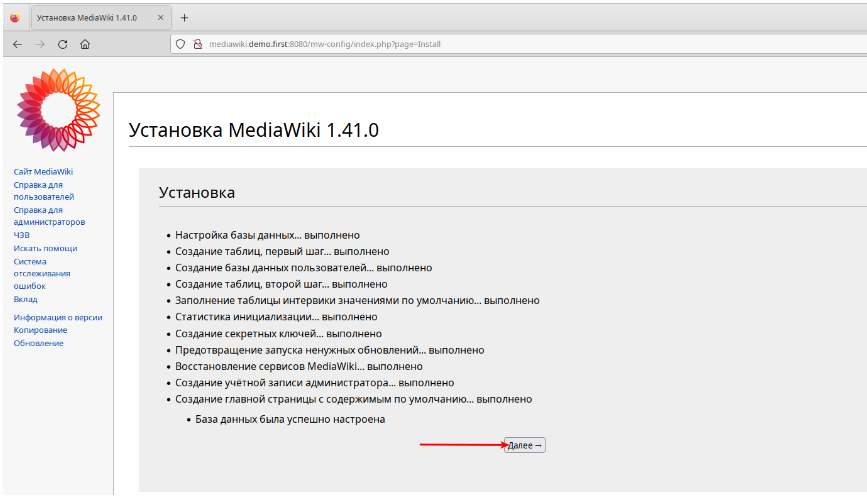


Заполняем необходимые сведения:

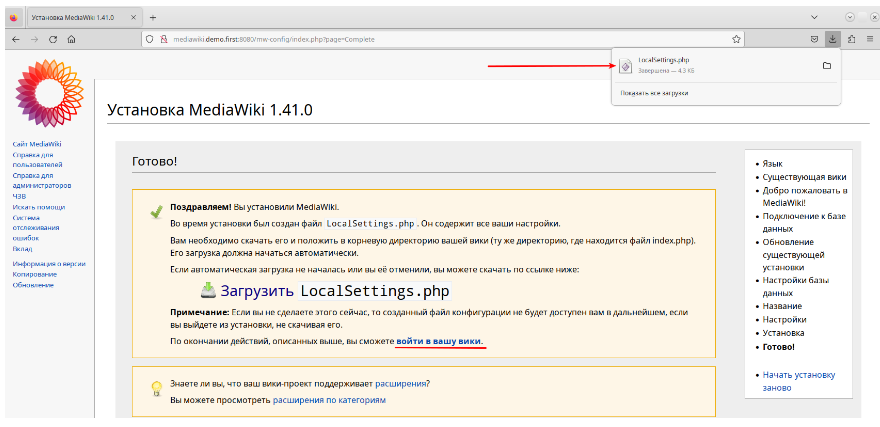








После чего будет автоматически скачен файл LocalSettings.php - который необходимо передать на HQ-SRV в домашнюю директорию пользователя root туда же где лежим wiki.yml:

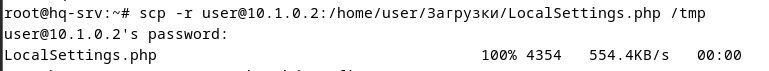


**apt install ssh -y**

**systemctl enable --now sshd**

**HQ-SRV:**

Передаём его на HQ-SRV:

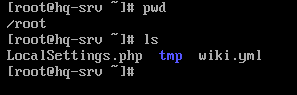


Проверяем:



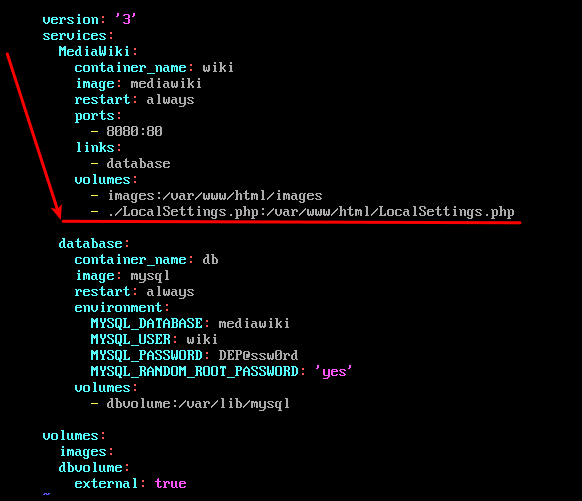
Перемещаем в домашнюю директорию пользователя root:

mv /tmp/LocalSettings.php ~/



Раскомментируем строку в файле wiki.yml:

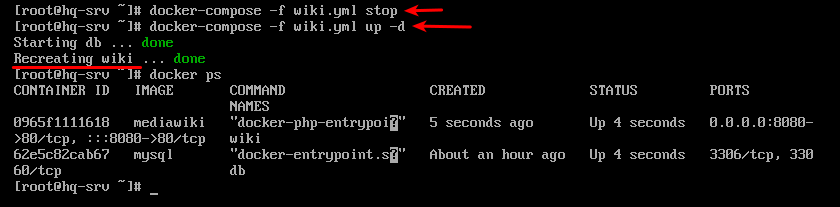
nano wiki.yml



Перезапускаем сервисы средствами docker-compose:

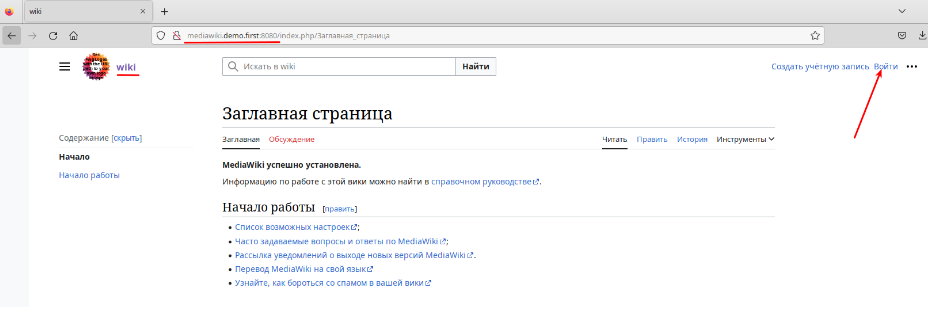
docker-compose -f wiki.yml stop

docker-compose -f wiki.yml up -d



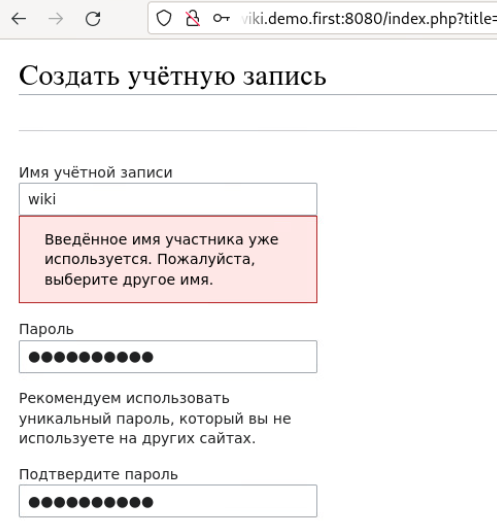
**CLI:**

Проверяем доступ к http://mediawiki.demo.first:8080



вход из под пользователя wiki с паролем DEP@ssw0rd:





После создания логинимся под пользователем

