Модуль 2 Задание 3 chrony

Apt install chrony

Кфг файл(nano /etc/chrony.conf):

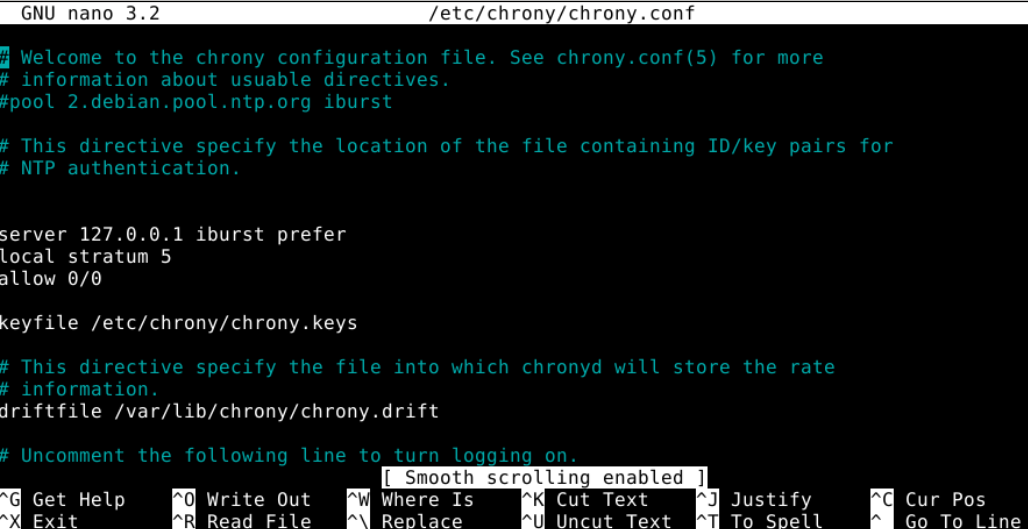
комментируем следующую строчку: pool 2.debian.pool.ntp.org iburst

И ниже пишем

server 127.0.0.1 iburst prefer

Local stratum 5

Allow 0/0



Сохраняем и выходим

systemctl restart chronyd – перезагружаем сервис chrony после изменения конфига

systemctl enable chronyd – включаем автозагрузку

Через команду chronyc sources проверим что все правильно работает, должен отобразиться только localhost.

Теперь на HQ-SRV, HQ-CLI, BR-RTR, BR-SRV - установить пакет chrony и проделать следующие одинаковые действия:

На HQ-SRV заходим в конфигурационный файл и также комментируем строку

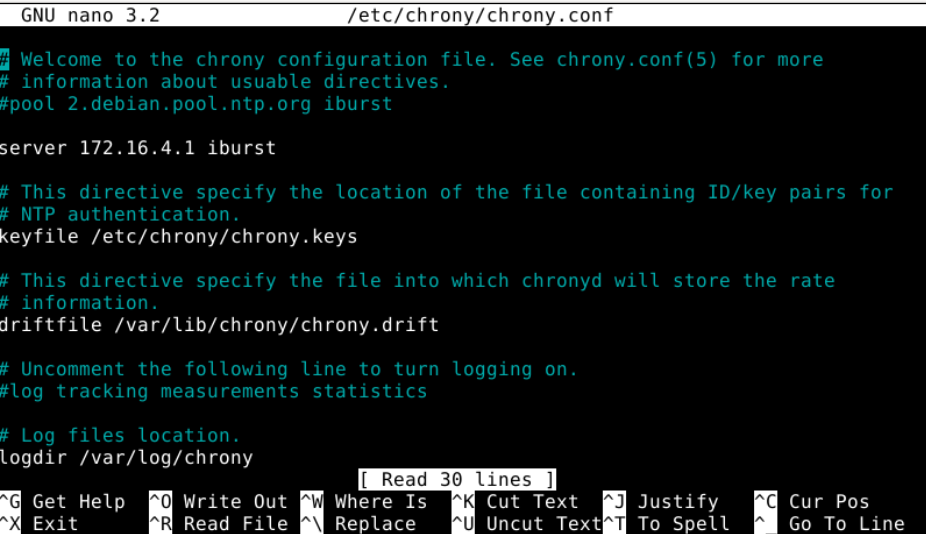
nano /etc/chrony.conf

pool 2.debian.pool.ntp.org iburst

И ниже пишем:

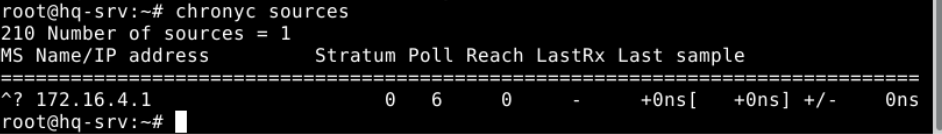
server [ip адрес HQ-RTR] iburst

server 172.16.4.1 iburst



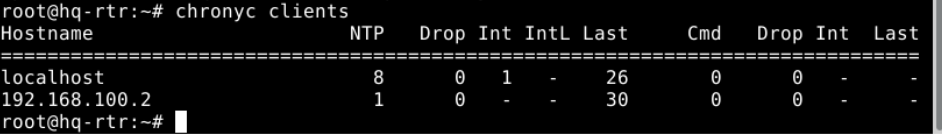
systemctl restart chronyd – перезагружаем

Проверяем командой chronyc sources



После этого можно переключится на HQ-RTR и проверить есть ли к нам подключения

Командой chronyc clients – сможем увидеть список клиентов подключенных к hq-rtr



Проделываем те же действия на остальных машинах: HQ-CLI, BR-RTR, BR-SRV

Задание 4. Ansible

На BR-SRV прописываем

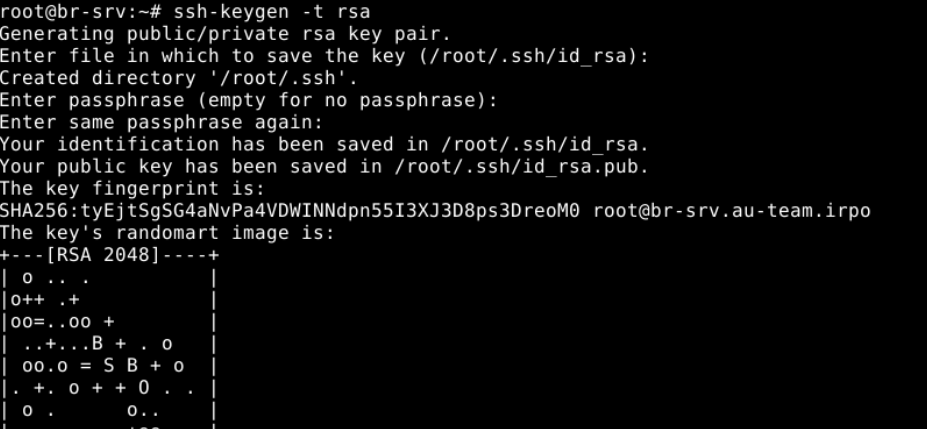
apt install ansible -y устанавливаем ansible

Заходим под другим пользователем

su sshuser

Создаем приватный и публичный ключ.

ssh-keygen -t rsa, везде нажимаем enter, оставляя все по умолчанию.



Выполняем следующие команды, чтобы скопировать наш публичный ключ на все остальные машины

ssh-copy-id -p 2024 [sshuser@192.168.100.2](http://sshuser@192.168.100.2) – HQ-SRV

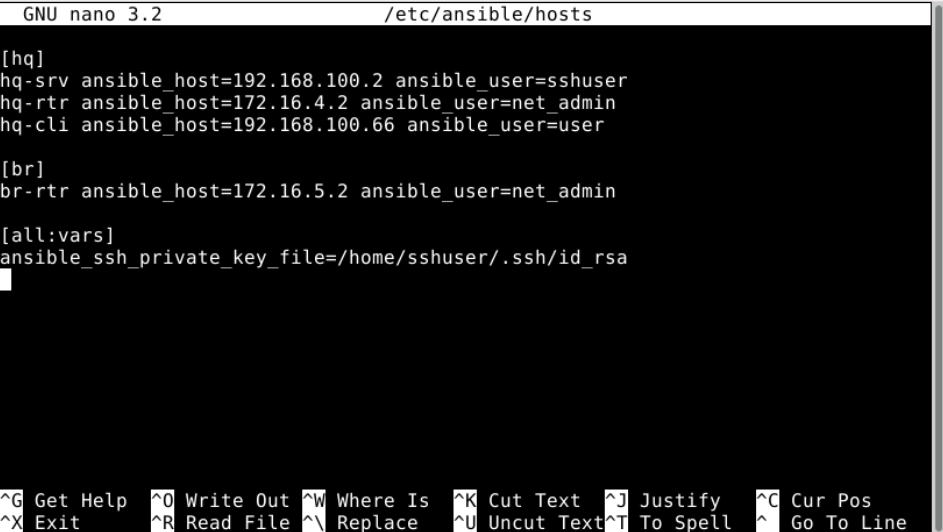
ssh-copy-id [user@192.168.100.66](http://user@192.168.100.66) HQ-CLI

ssh-copy-id [net\_admin@172.16.4.2](http://net_admin@172.16.4.2) HQ-RTR

ssh-copy-id [net\_admin@172.16.5.2](http://net_admin@172.16.5.2) BR-RTR

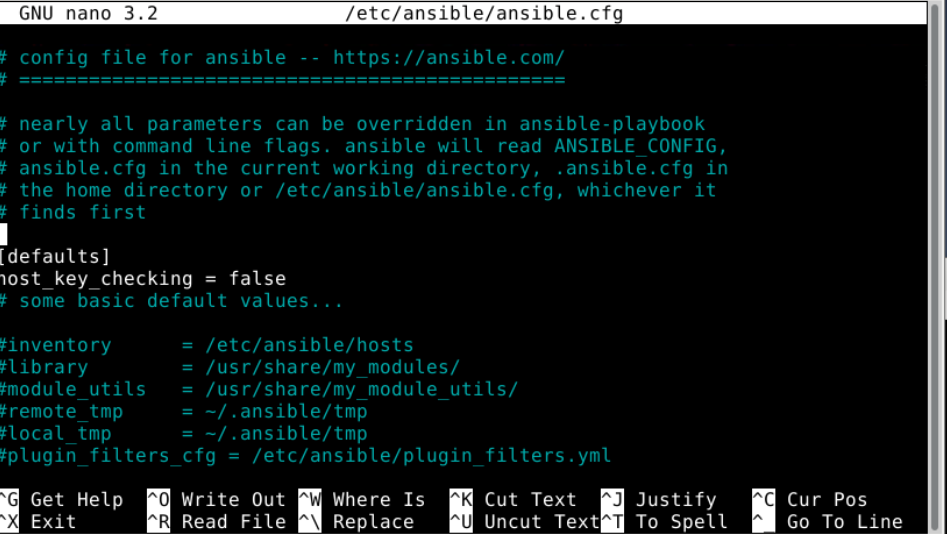
Далее редактируем конфигурационный файл, записываем информацию о хостах для подключения:

nano /etc/ansible/hosts

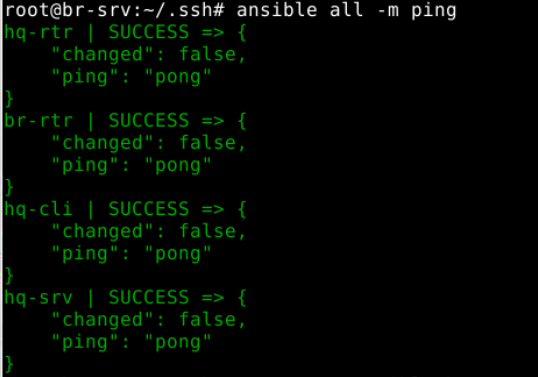


Редактируем /etc/ansible/ansible.cfg, отключая проверку подтверждения хостов:

nano /etc/ansible/ansible.cfg



ansible all –m ping – проверяем работоспособность настроенных узлов, должно быть везде pong, вывод зеленым цветом.



Задание 5. Docker

На BR-SRV прописываем следующие команды

sudo apt install docker docker.io docker-compose –y

Переходим в домашнюю директорию пользователя sshuser

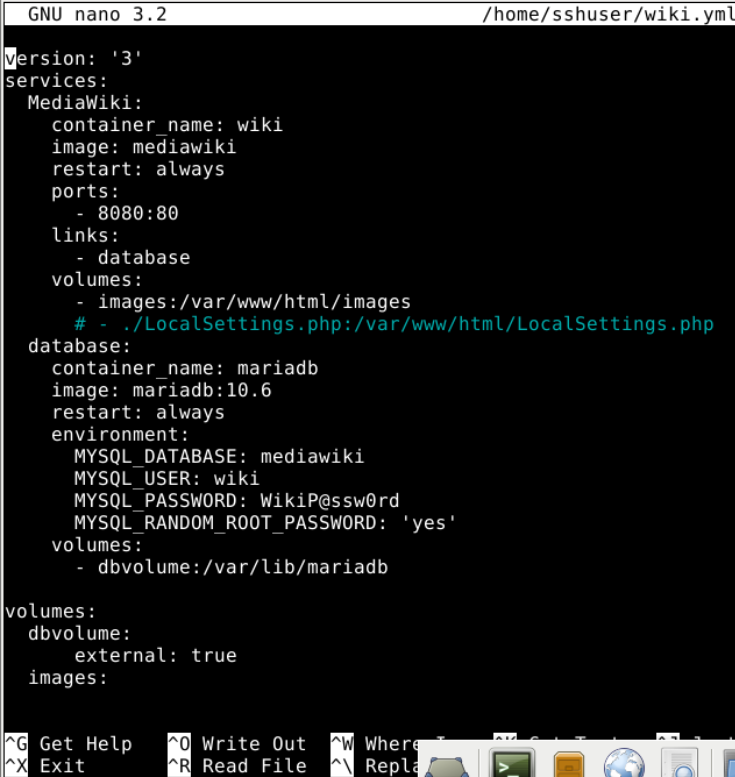
cd /home/sshuser

Создаем файл wiki.yml

nano wiki.yml

Копируем и вставляем содержимое c сайта в wiki.yml

<https://www.mediawiki.org/wiki/Docker/Hub>И вносим следующие изменения, порт на 8080, имя контейнера и образ на mariadb, а также рядом с образом указать версию 10.6, имя БД на mediawiki, пароль пользователя БД на WikiP@ssw0rd, изменяем параметр volumes (правим путь)

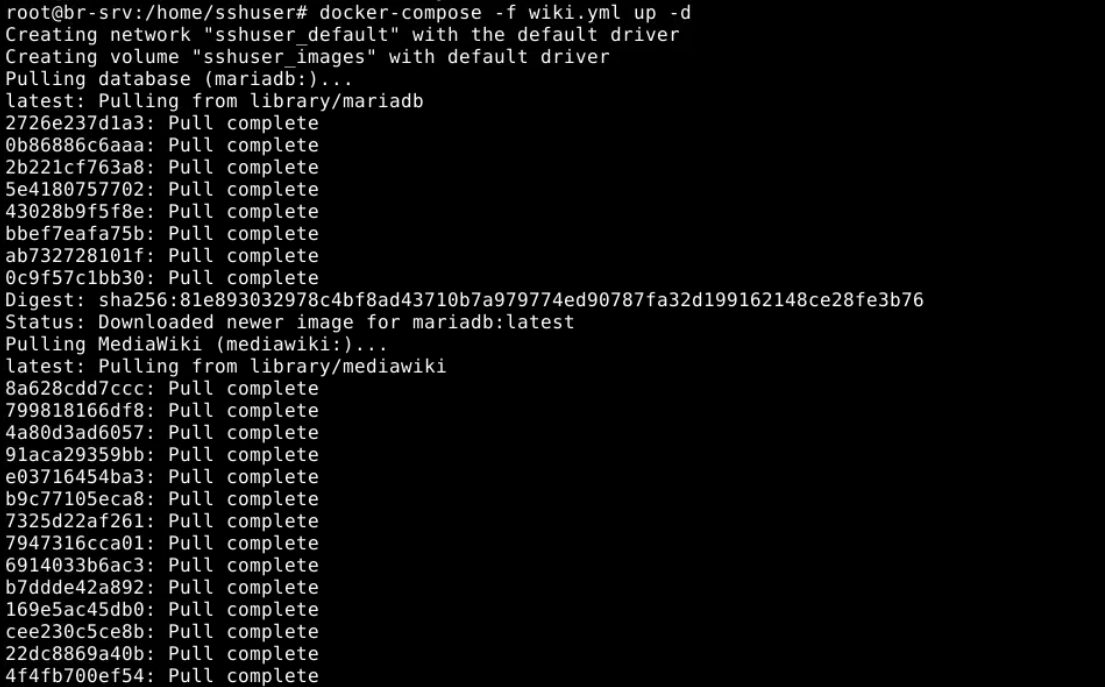


Чтобы отдельный volume для хранения базы данных имел правильное имя - создаём его средствами docker:

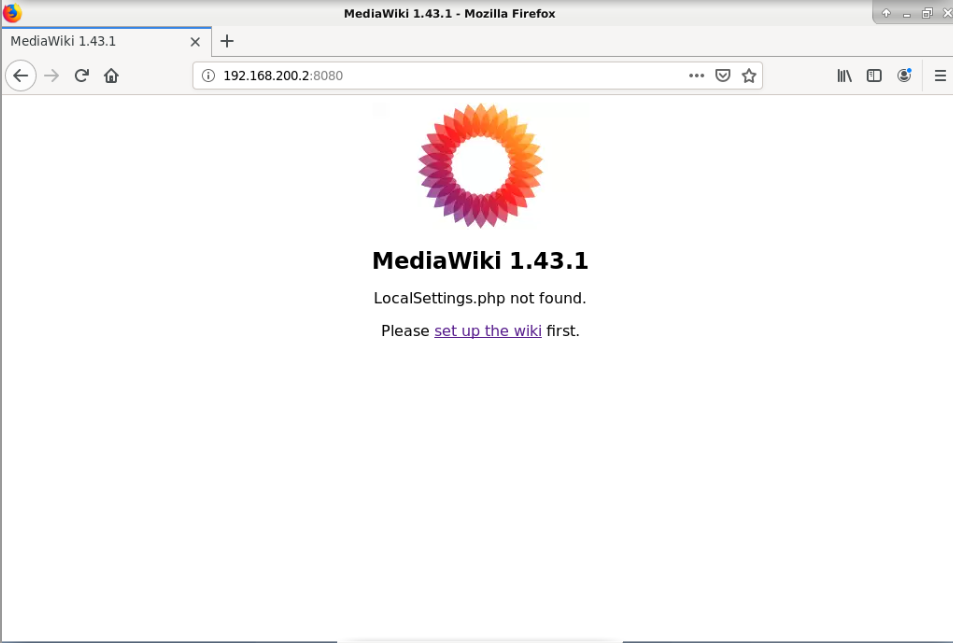
docker volume create dbvolume

Выполняем сборку и запуск стека контейнеров с приложением MediaWiki и базой данных описанных в файле wiki.yml:

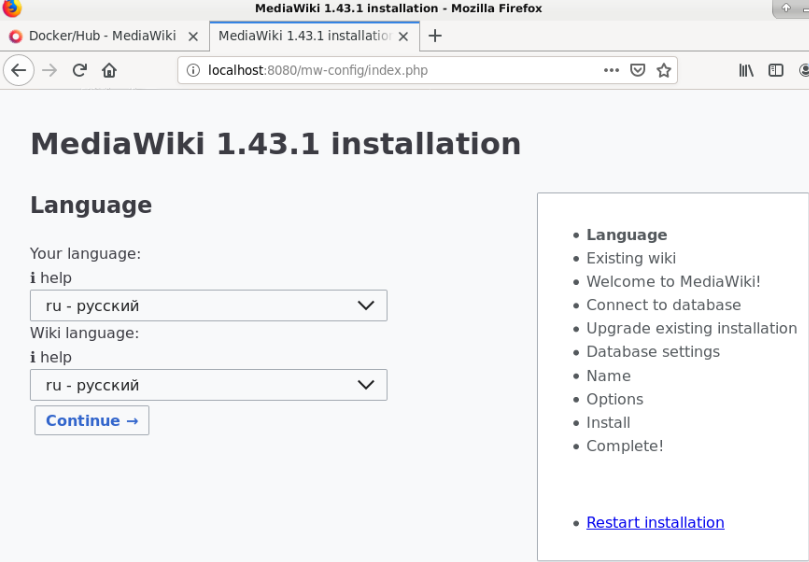
docker-compose -f wiki.yml up -d



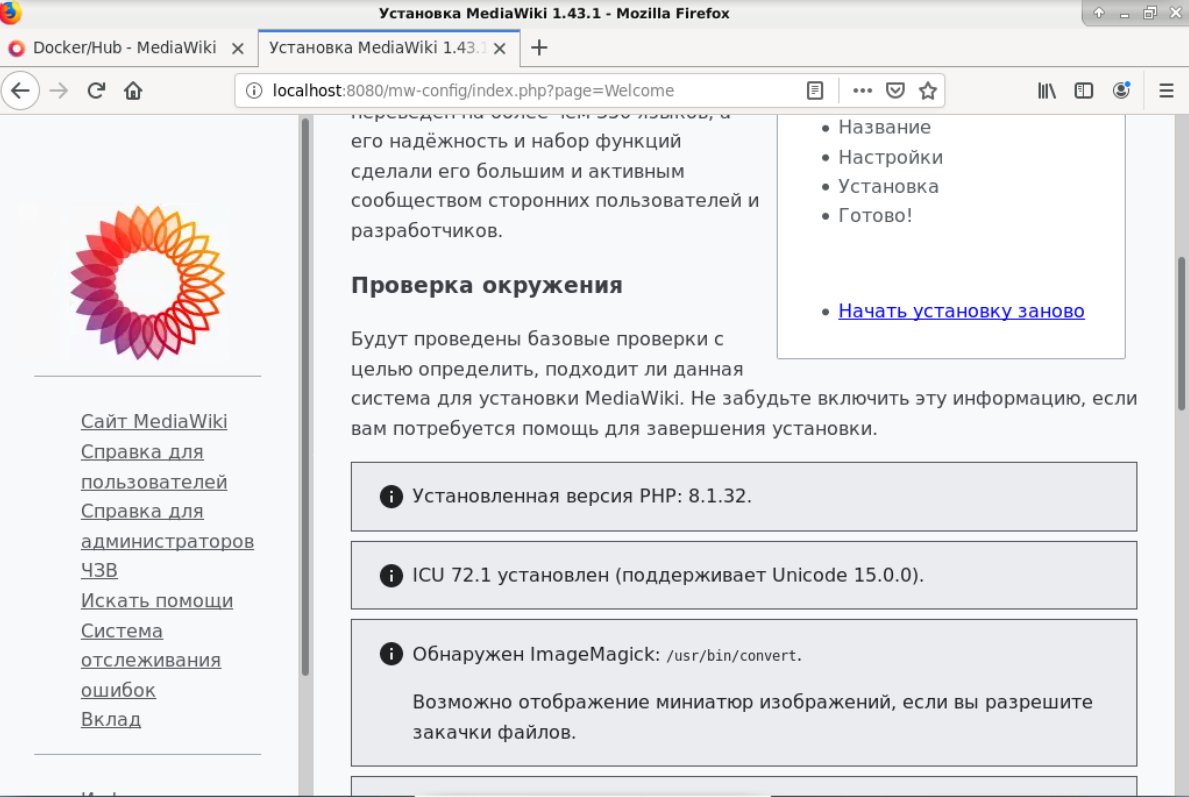
Переходим в браузер, введя в адресную строку браузера “192.168.200.2:8080” и нажимаем “set up the wiki”



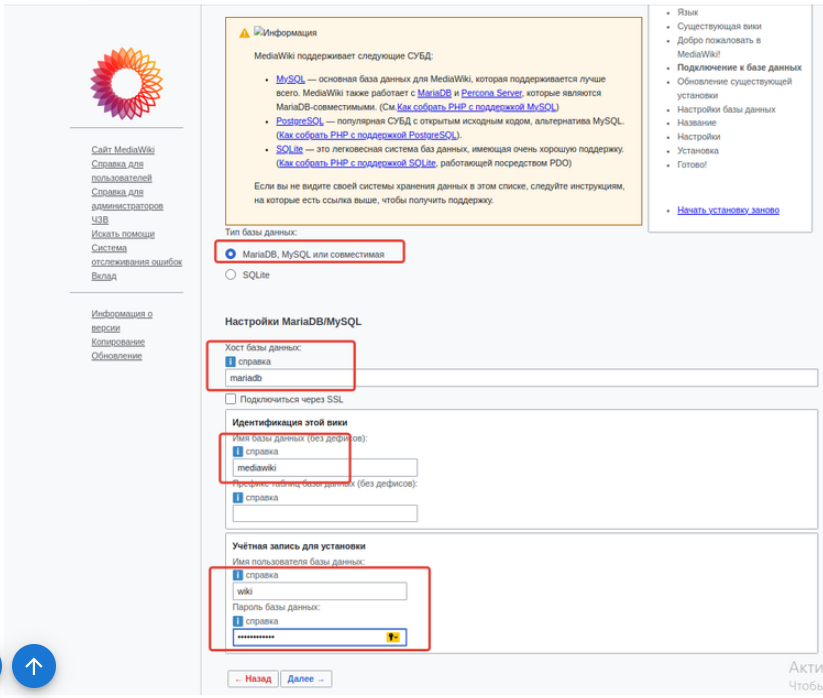
Выбираем русский язык и нажимаем “continue”



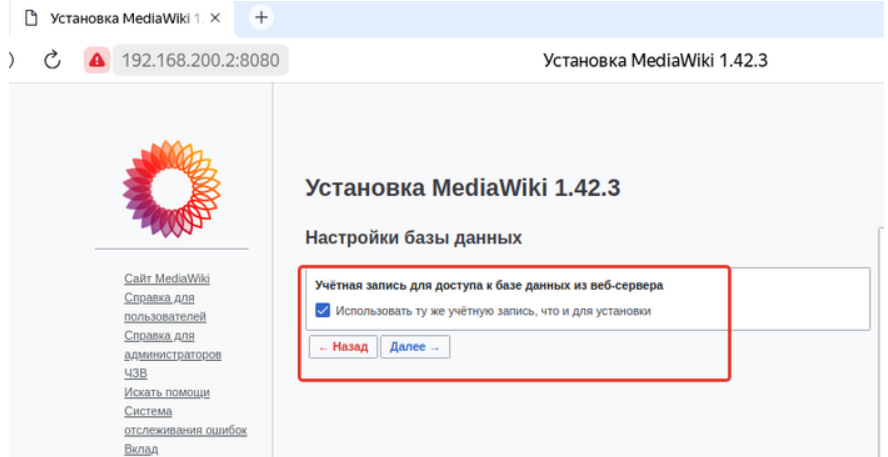
После успешной проверки внешней среды - жмем Далее:



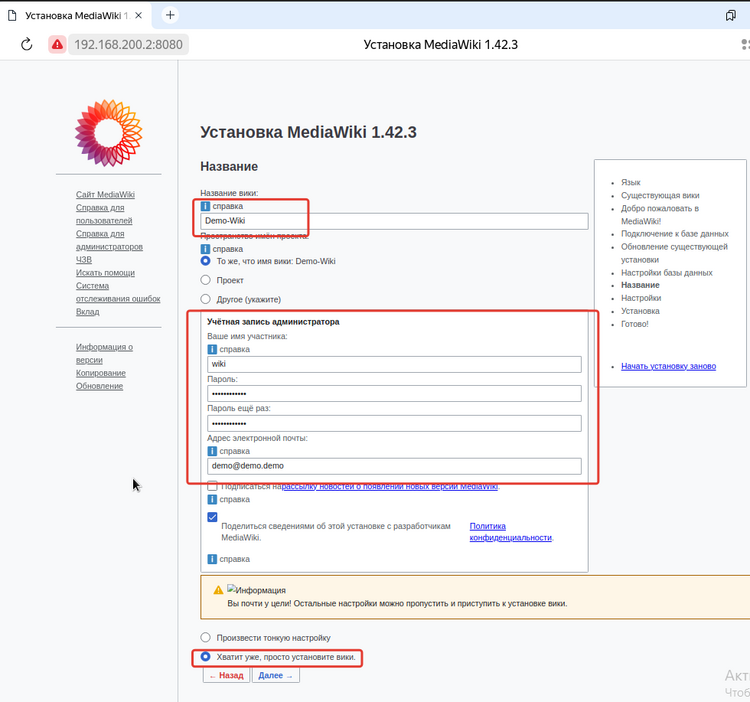
Заполняем параметры подключение к БД в соответствие с заданными переменными окружения в wiki.yml, которые соответствуют заданию:



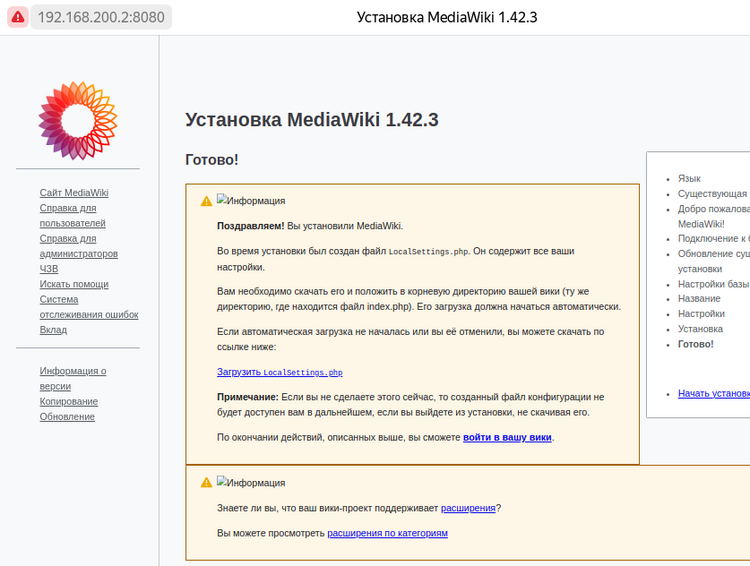
Ставим галочку и жмем далее



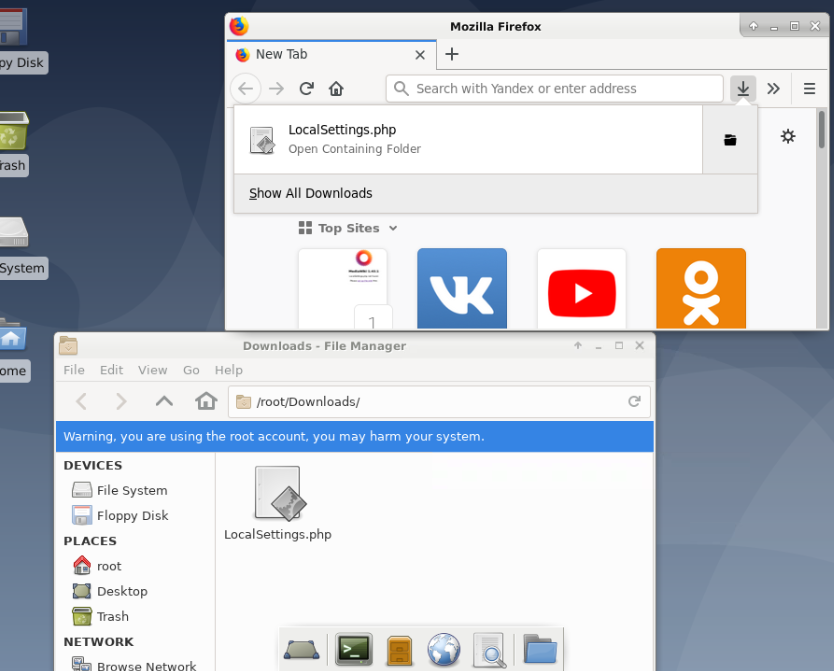
Вносим необходимые сведения:



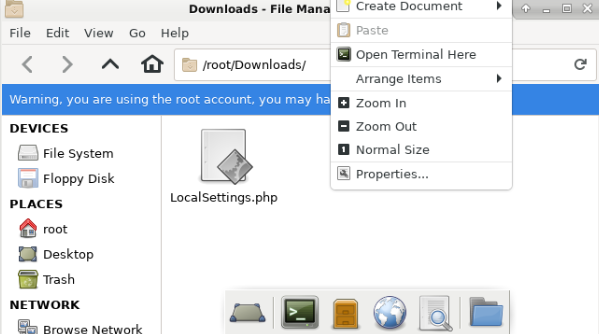
Будет автоматически скачен файл LocalSettings.php - который необходимо поместить в директорию /home/sshuser туда же где лежит wiki.yml



Открываем папку с загруженным файлом

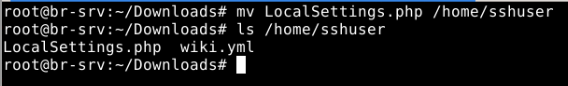


Нажимаем правую кнопку мыши и выбираем “open terminal here”.



В открывшемся терминале прописываем следующую команду:

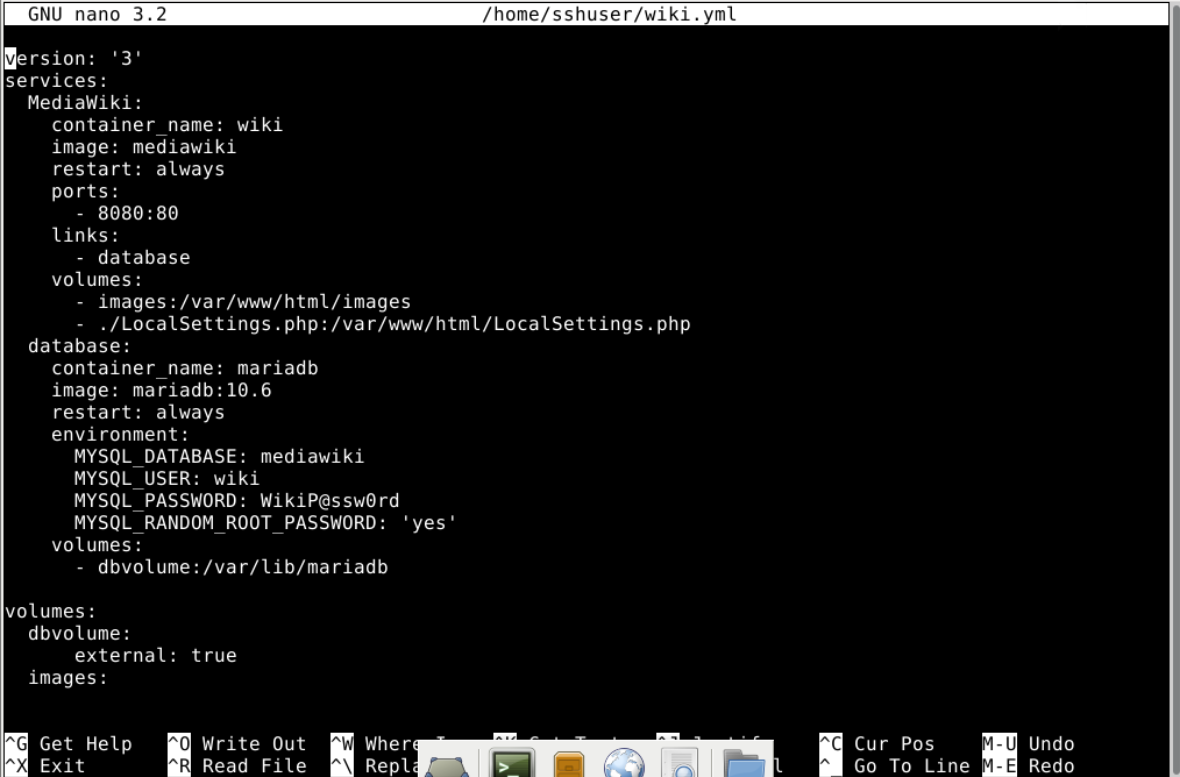
mv LocalSettings.php /home/sshuser



И командой ls мы можем удостовериться что файл действительно находится в нужной нам директории

Раскомментируем строку в файле wiki.yml :

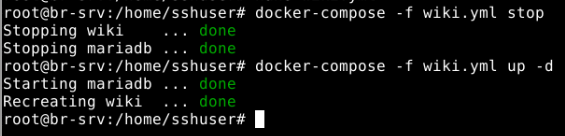
nano wiki.yml



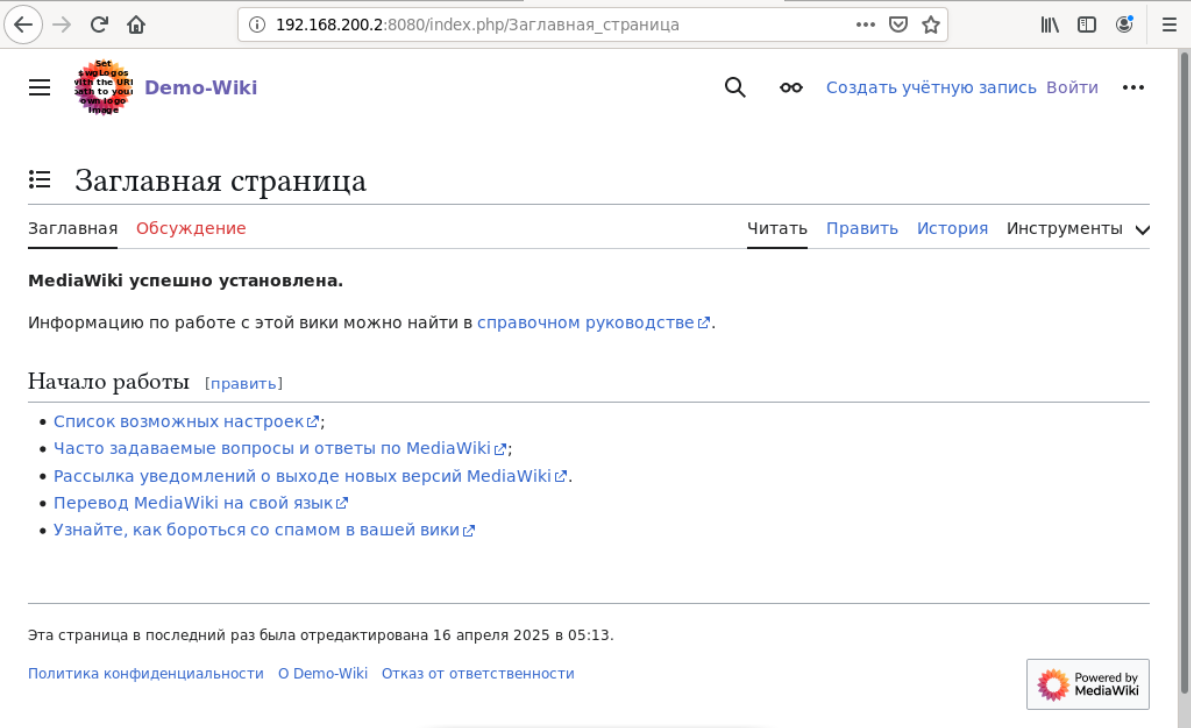
Перезапускаем сервисы средствами docker-compose:

docker-compose -f wiki.yml stop

docker-compose -f wiki.yml up -d



Проверяем доступ к Wiki <http://192.168.200.2:8080>



Входим под пользователя wiki с паролем WikiP@ssw0rd:

Задание 6

1. Доступ к сервису wiki по порту 80

Заходим на BR-RTR и прописываем в консоль команды:

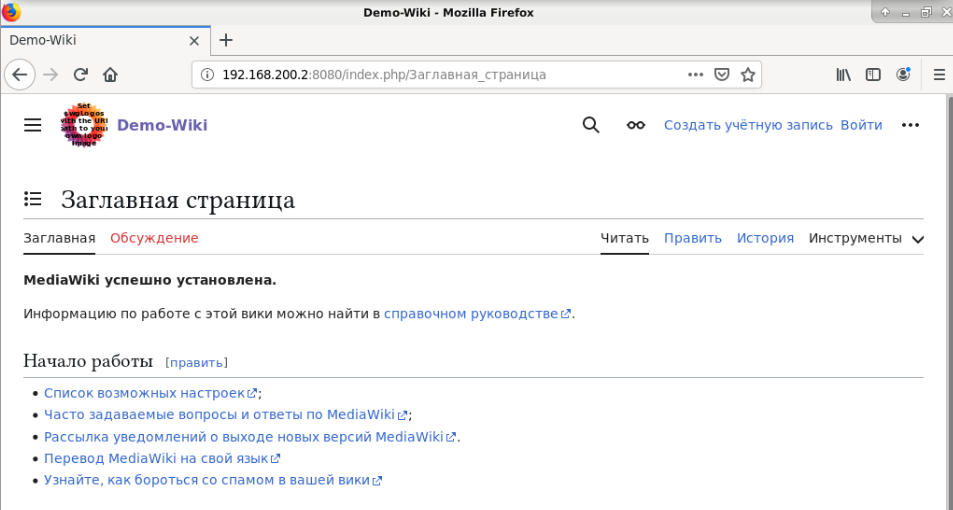
***Делаем проброс порта****80****при обращение на****внешний интерфейс BR-RTR (ens3)****на порт****8080 BR-SRV*

Iptables –t nat –A PREROUTING -d 172.16.5.2/28 –p tcp **--**dport 80 –j DNAT **--**to-destination 192.168.200.2:8080

netfilter-persistent save – сохраняем настройки iptables, чтобы работало после перезагрузки.

С HQ-CLI в браузере переходим по IP адресу BR-RTR должны попасть на страницу MediaWiki

http://172.16.5.2



2) Проброс порта 2024 на маршрутизаторе HQ-RTR в порт 2024 на HQ-SRV

На HQ-RTR вводим команды:

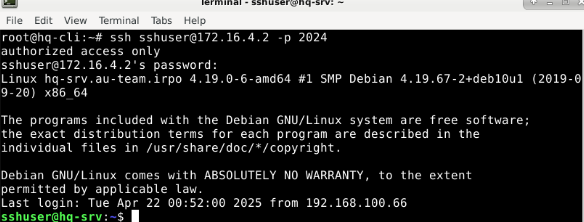
iptables -t nat -A PREROUTING -d 172.16.4.2/28 -p tcp **--**dport 2024 -j DNAT **--**to-destination 192.168.100.2:2024

netfilter-persistent save – сохраняем настройки iptables, чтобы работало после перезагрузки.

С HQ-CLI подключаемся по ssh к BR-RTR по порту 2024

ssh sshuser@172.16.4.2 -p 2024

**Должны попасть на HQ-SRV**



3) Проброс порта 2024 на маршрутизаторе BR-RTR в порт 2024 на BR-SRV

На BR-RTR вводим команды:

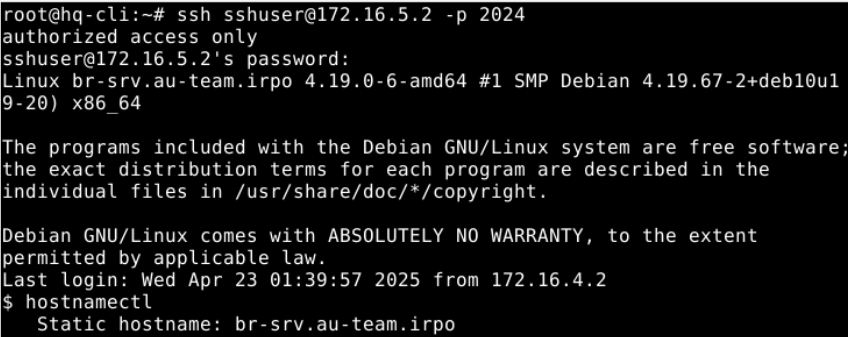
Iptables –t nat –A PREROUTING -d 172.16.5.2/28 –p tcp **--**dport 2024 –j DNAT **--**to-destination 192.168.200.2:2024

netfilter-persistent save – сохраняем настройки iptables, чтобы работало после перезагрузки.

С HQ-CLI подключаемся по ssh к BR-RTR по порту 2024

ssh sshuser@172.16.5.2 -p 2024

**Должны попасть на BR-SRV**



Задание 7. Moodle

## **Установка и настройка веб-сервера Apache**

apt install apache2 – скачиваем apache

systemctl enable apache2 --now – включаем автозагруку

Устанавливаем пакеты php:

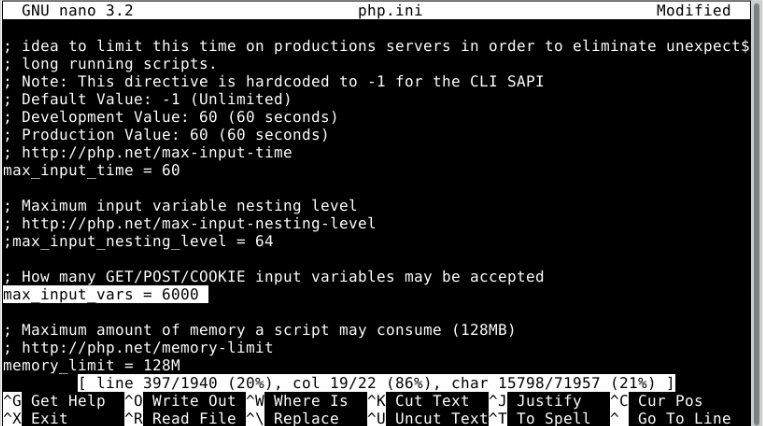
apt install php libapache2-mod-php php-mysql php-mysqlnd php-xmlrpc php-gd php-pdo php-mbstring php-zip php-intl php-soap php-ldap php-json php-curl php-opcache php-ctype php-xml php-simplexml php-dom php-exif

Заходим в конфиг файл:

/etc/php/7.3/apache2/php.ini

Здесь нужно раскомментировать и изменить значение

max\_input\_vars=6000



Если тяжело искать можно прописать max\_input\_vars=6000 в начале файла /etc/php/7.3/apache2/php.ini

Перезапускаем apache2:

systemctl restart apache2

Устанавливаем mariadb

sudo apt install mariadb-server mariadb-client -y

systemctl enable mariadb **--**now – включаем автозагрузку.

Запускаем встроенный инструмент настройки MariaDB:

mysql\_secure\_installation

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



Создание пользователя и базы данных с поддержкой русского языка

mysql -u root -p

MariaDB[(none)] > CREATE DATABASE moodledb DEFAULT CHARACTER SET utf8;

MariaDB[(none)] > CREATE USER moodle@localhost IDENTIFIED BY 'P@ssw0rd';

Установка прав пользователя на БД

MariaDB[(none)] > GRANT ALL ON moodledb.\* TO 'moodle'@'localhost';

Оповестите MySQL о текущих изменениях:

MariaDB[(none)] > flush privileges;

Выход:

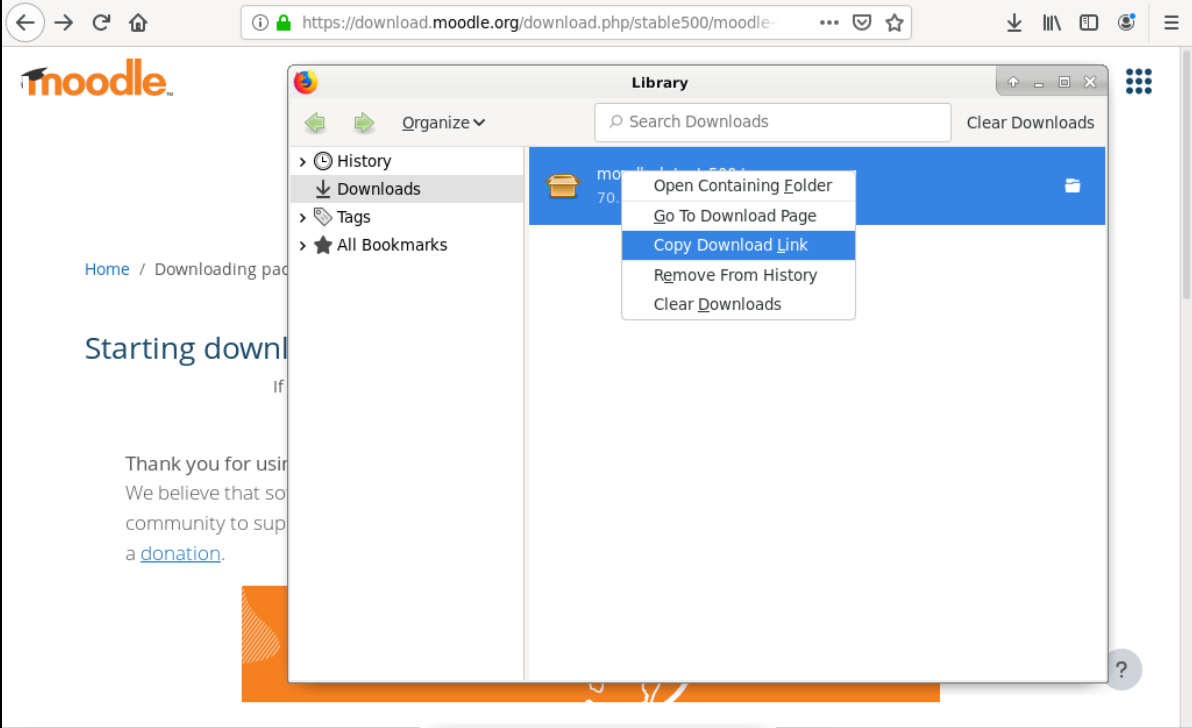
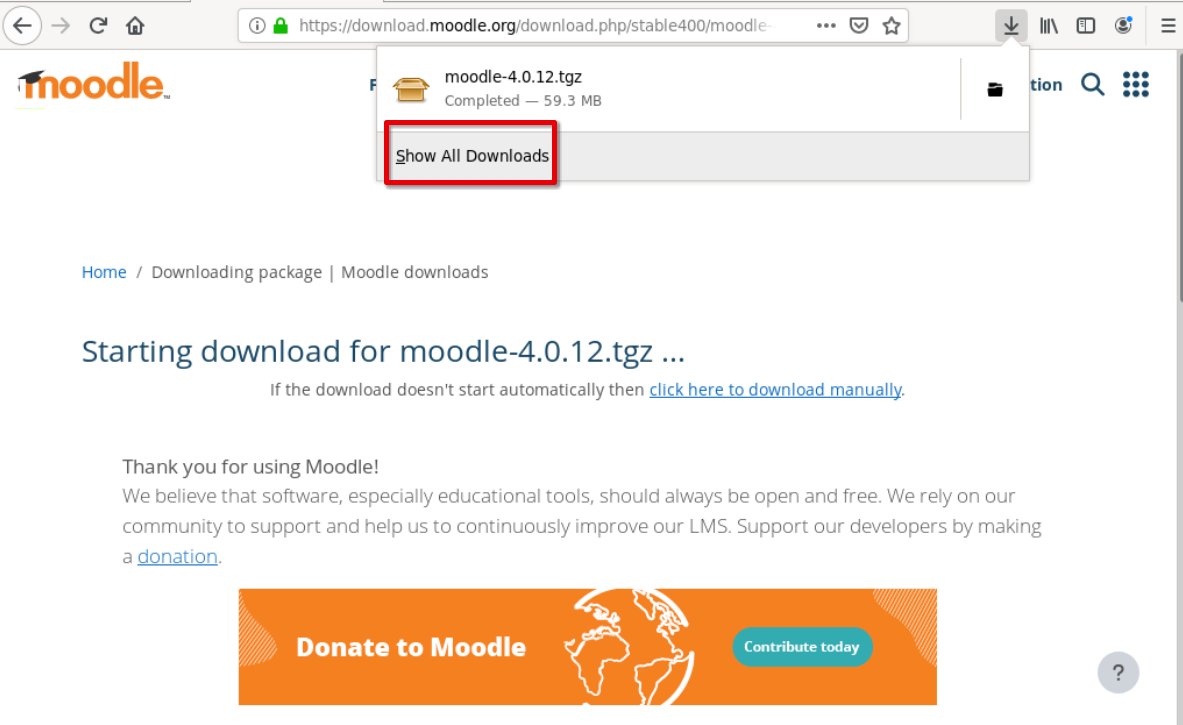
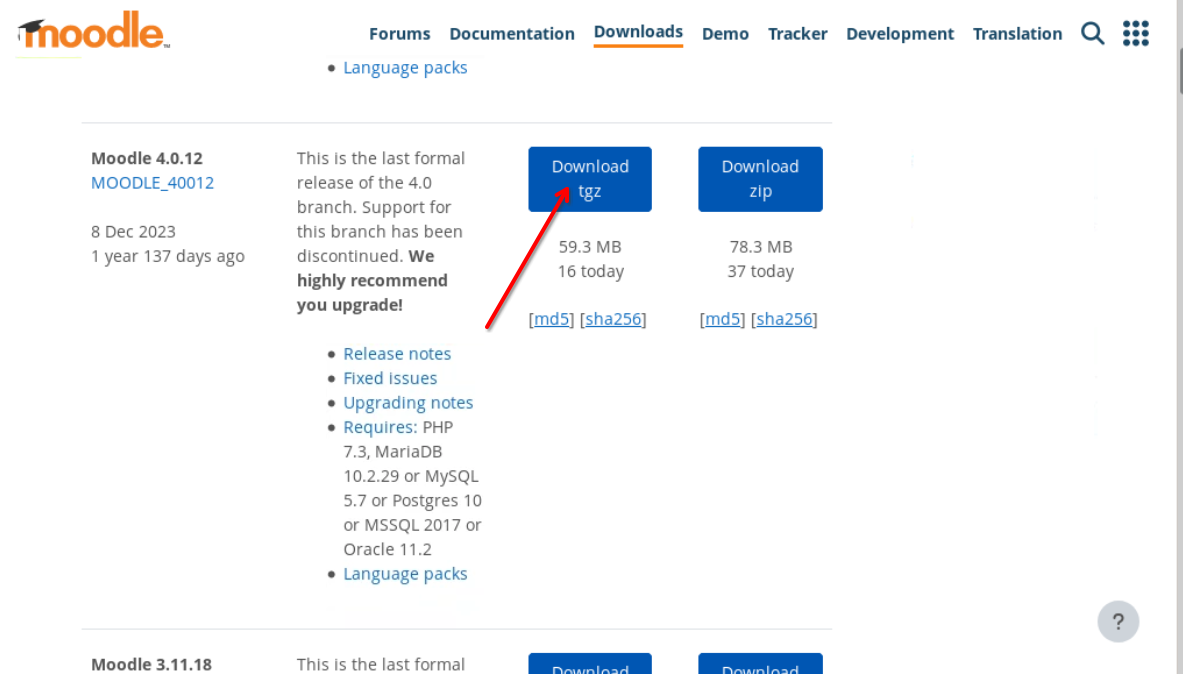
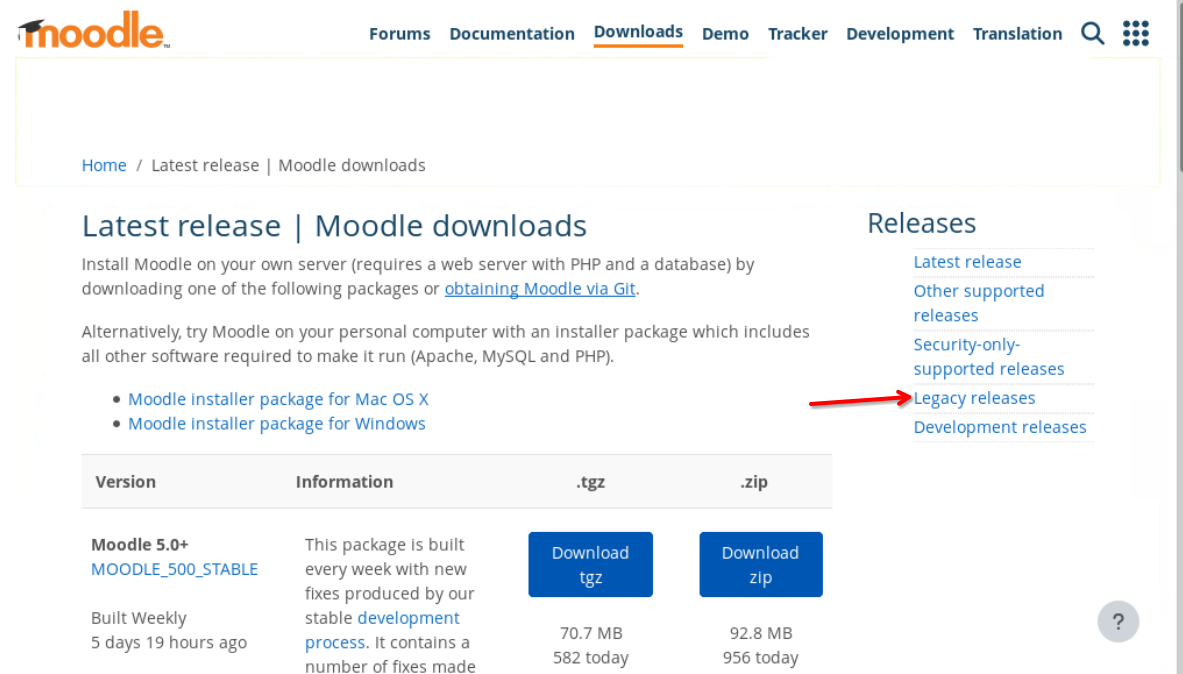
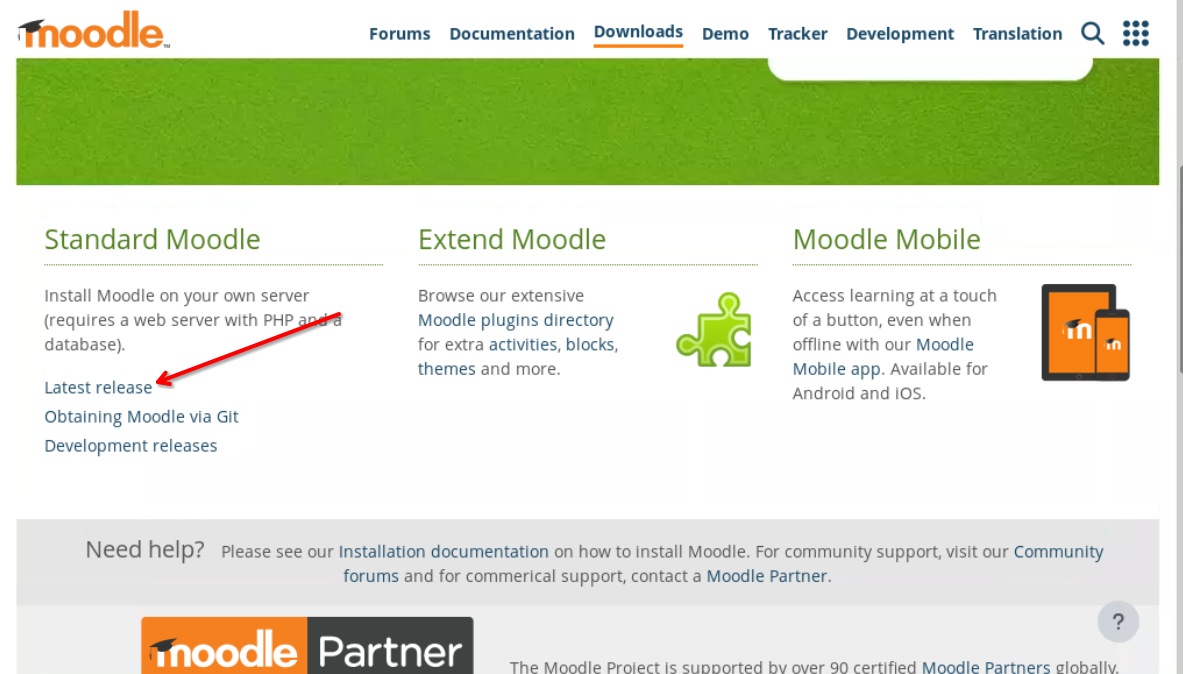
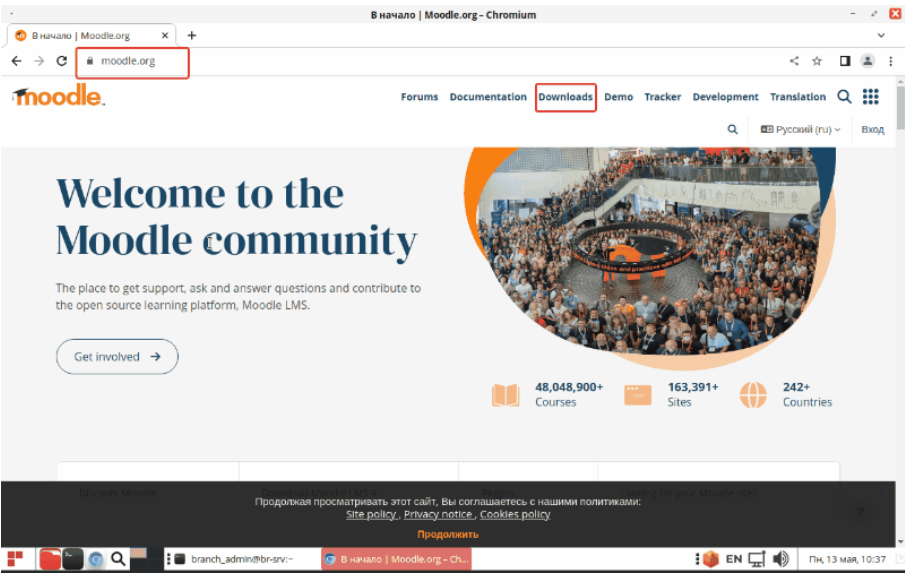
MariaDB[(none)] > quit;

Перезапуск mariadb

systemctl restart mariadb

Установка Moodle

Переходим на сайт Moodle. Копируем ссылку на последнюю версию moodle и скачиваем ее в диреторию /tmp



Копируем ссылку и вставляем ее для следующей команды:

wget [ссылка] -P /tmp

Должно получиться что-то типо того:

wget <https://packaging.moodle.org/stable500/moodle-latest-500.tgz> -P /tmp

Переходим в директорию /tmp и распаковываем архив

cd /tmp

tar -xzf /tmp/moodle-4.0.12.tgz

После распаковки рабочие файлы CMS лежат в папке moodle, которая была распакована из архива. Нужно переместить все файлы из папки moodle в корневую директорию сайта /var/www/html

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

Устанавливаем права на /var/www/html

chmod -R 0777 /var/www/html/

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

mkdir /var/moodledata

Этот каталог должен быть доступен для чтения и ЗАПИСИ тому пользователю, от чьего имени запускается веб-сервер (обычно „www-data“, „nobody“ или „apache“).

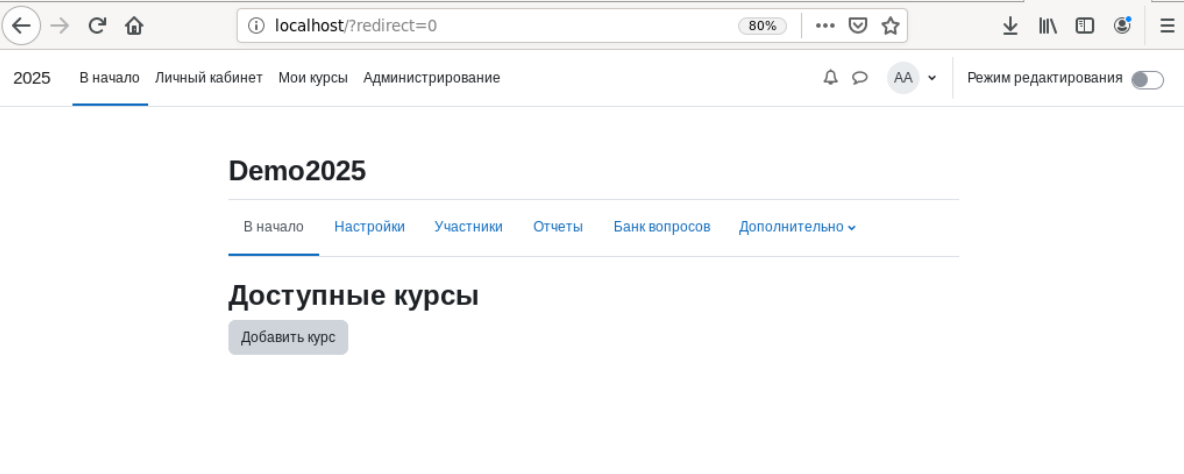
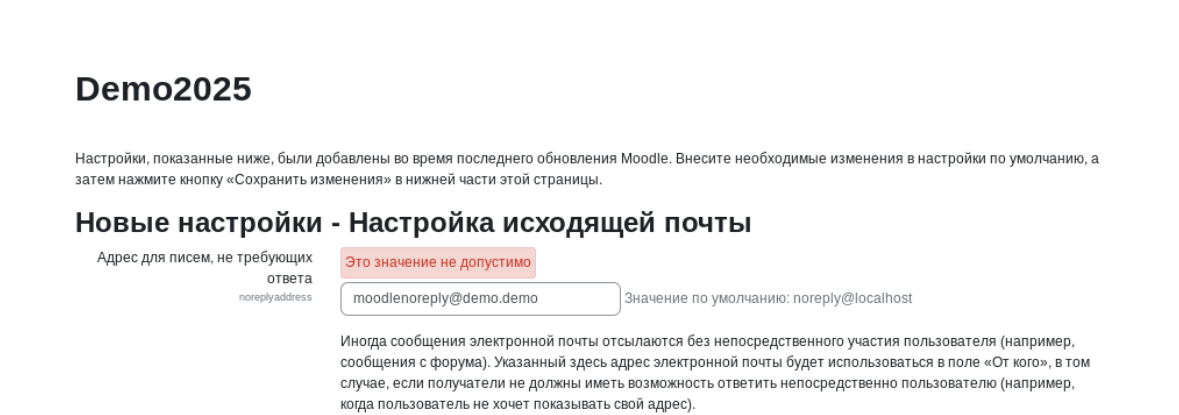
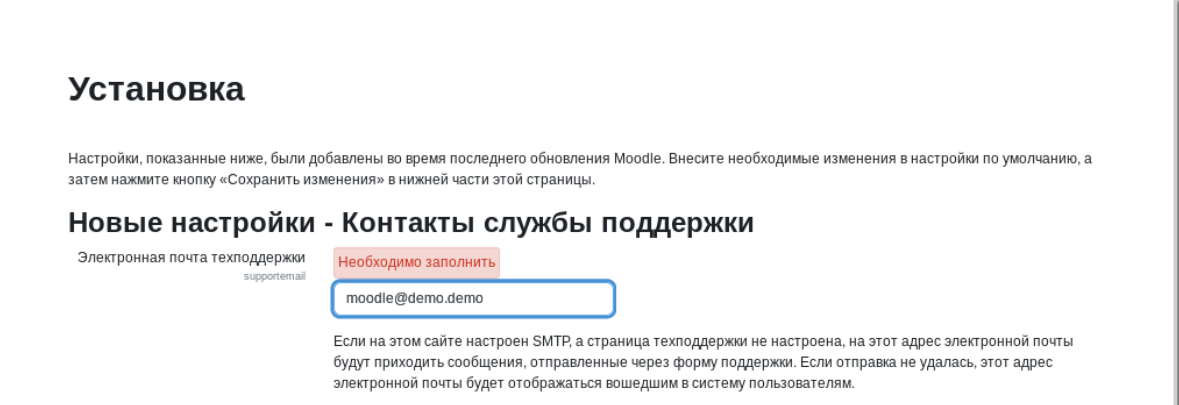
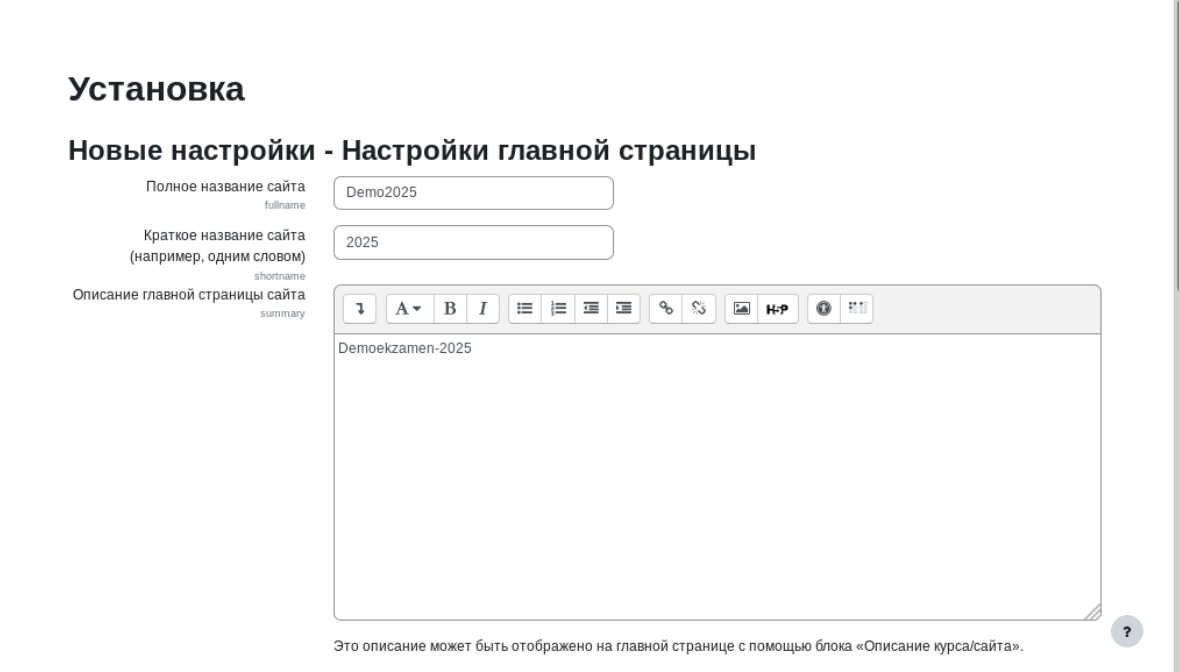
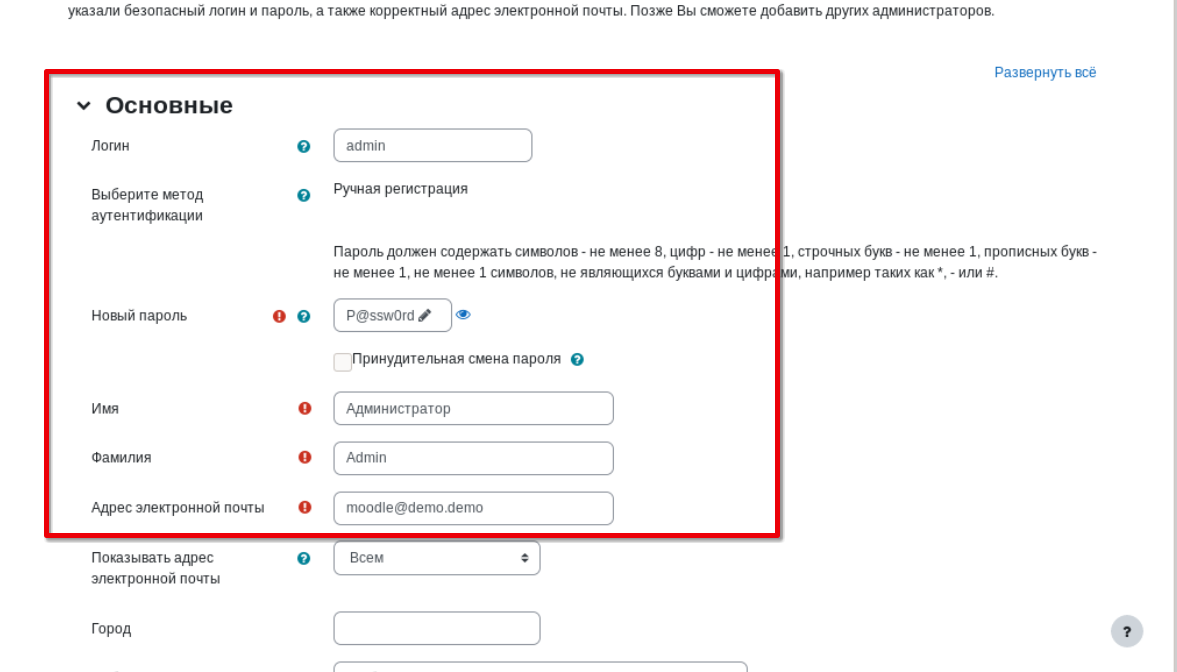
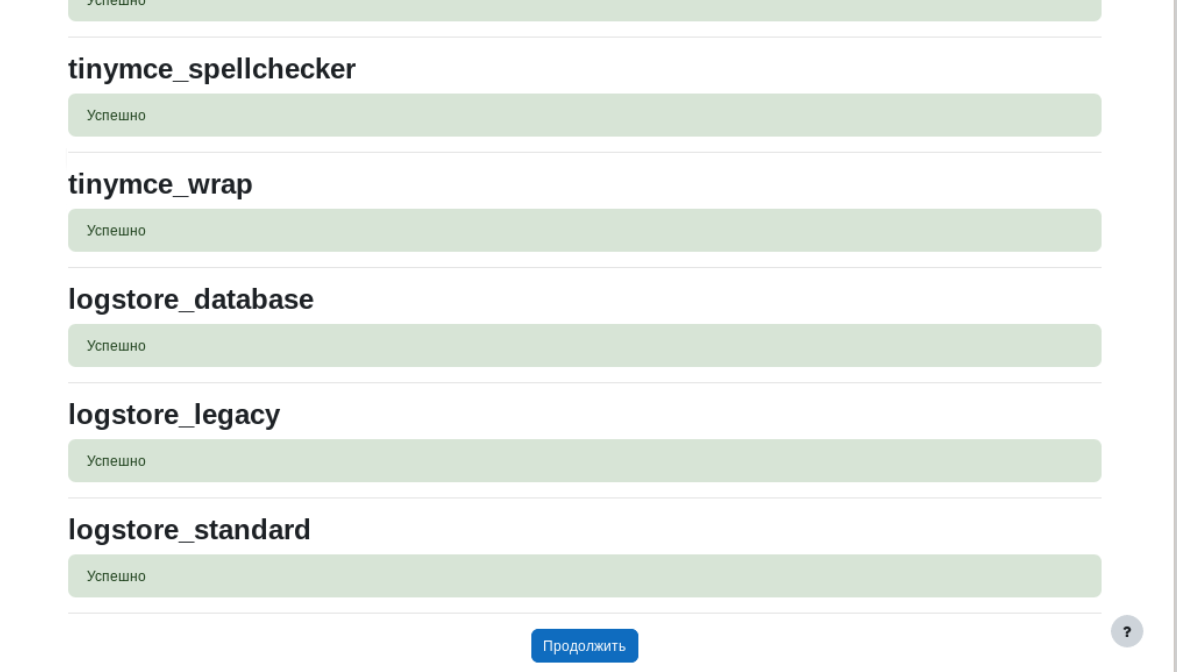
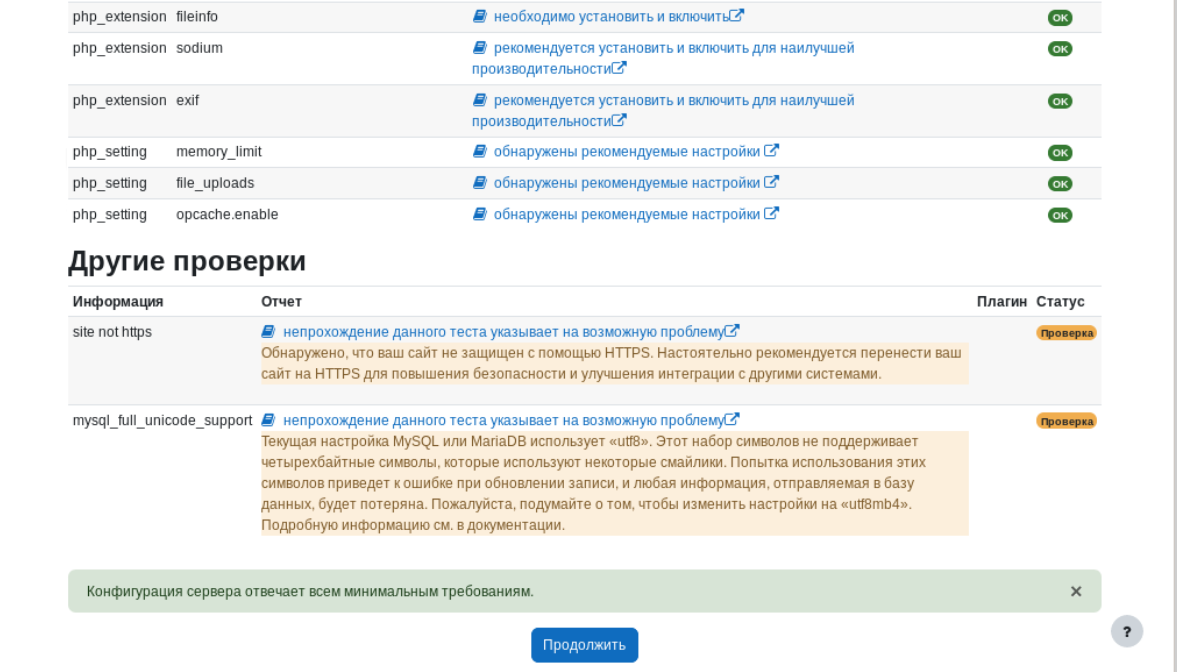
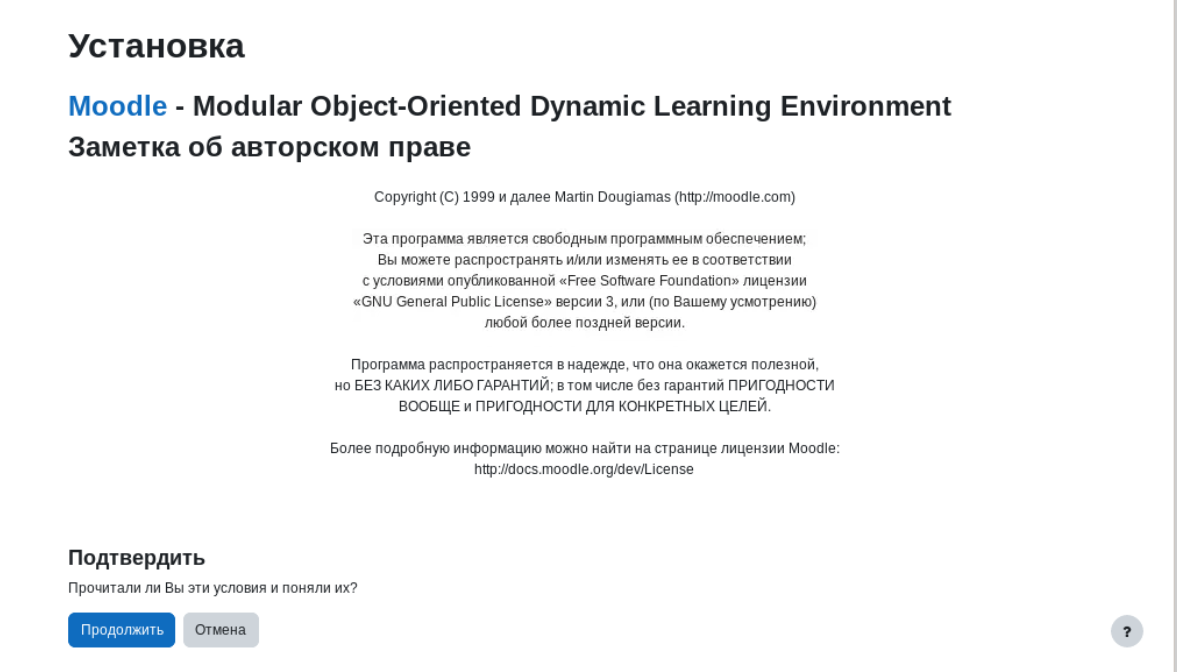
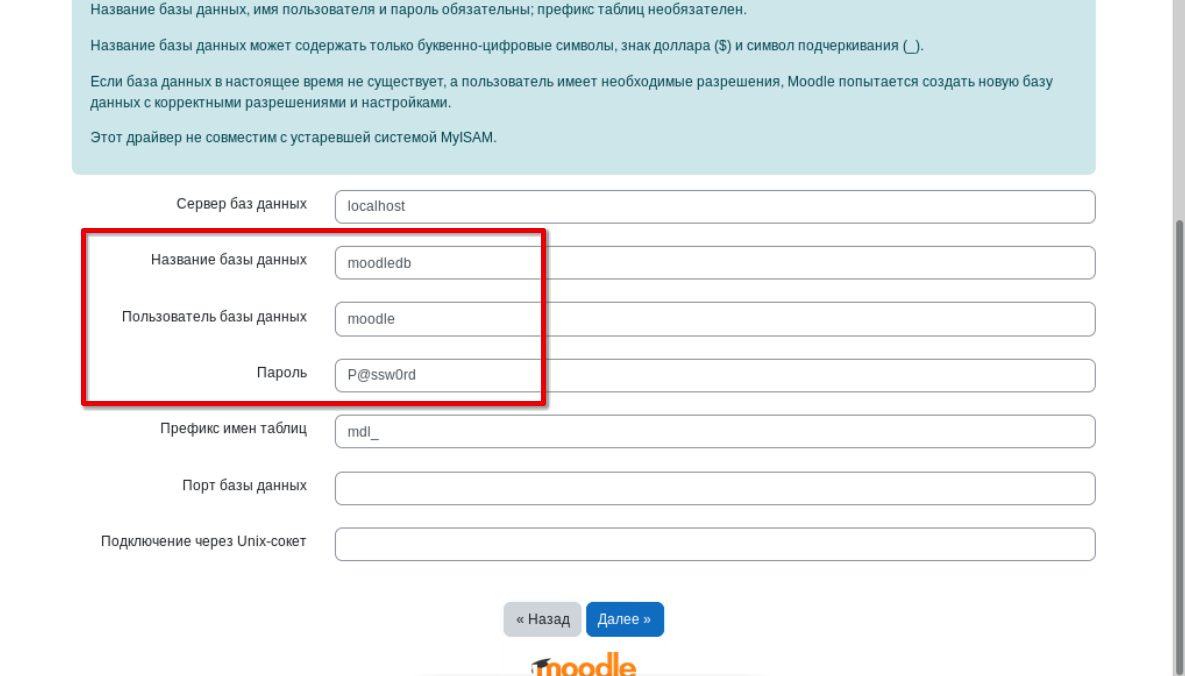
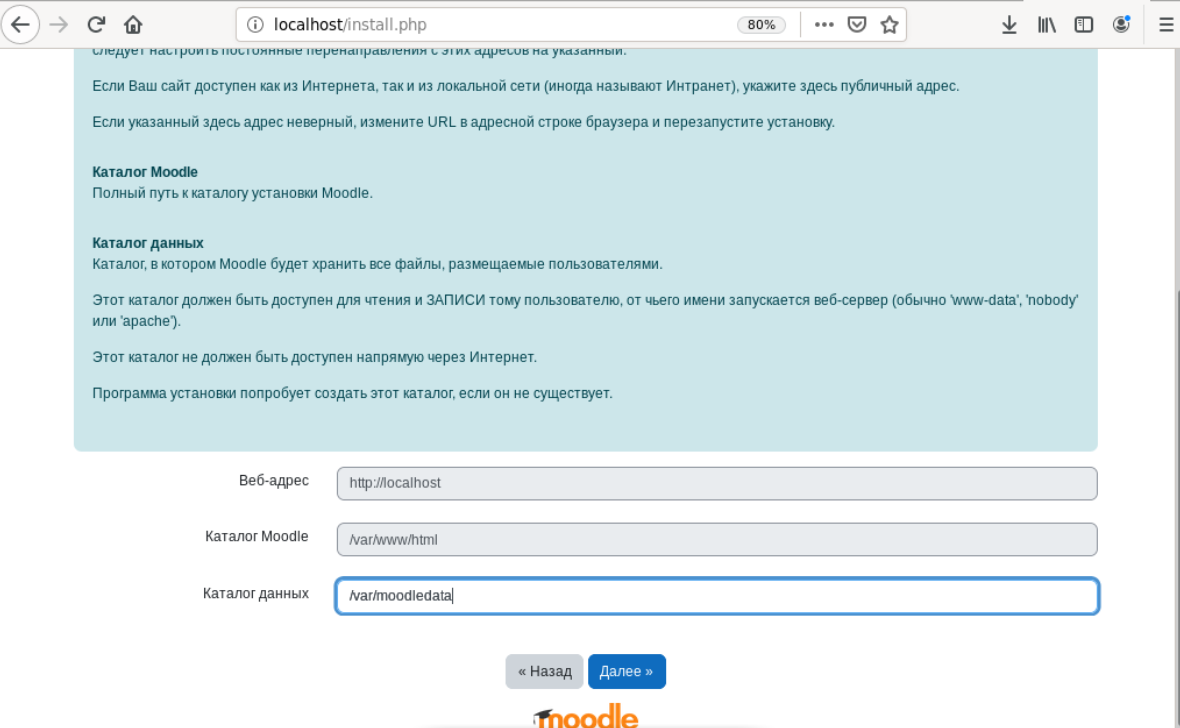
chmod -R 0777 /var/moodledata

rm /var/www/html/index.html – удаляем ненужный файл

Перезапуск apache

systemctl restart apache2

Для продолжения установки и настройки LMS Moodle в браузере ввести http://<IP\_HQ-SRV>



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

Установка Nginx

apt install nginx -y

Запуск и добавление в автозагрузку

systemctl enable **--**now nginx

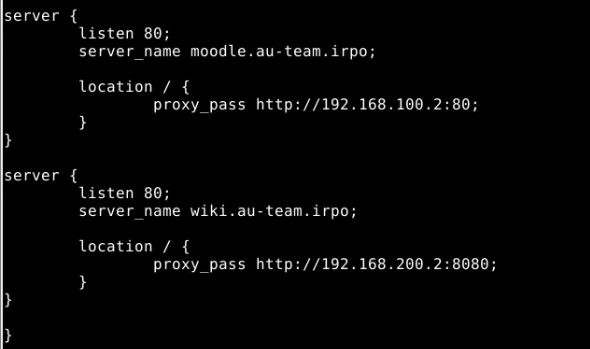
Открываем на редактирование конфигурационный файл Nginx

nano /etc/nginx/nginx.conf

Раскомментируем строчку



Cпускаемся в конец документа и перед **(НЕ ПОСЛЕ)** последней фигурной скобкой { прописываем



Перезагружаем Nginx

systemctl restart nginx