

1. Настройте доменный контроллер Samba на машине BR-SRV.

- Создайте 5 пользователей для офиса HQ: имена пользователей формата user№.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

Устанавливаем доменный контроллер Samba

```
apt-get install task-samba-dc
```

Настройка BIND9 для работы с Samba

Отключаем chroot следующей командой:

```
control bind-chroot disabled
```

Отключаем кэширование в файле bind находящийся по пути /etc/sysconfig/bind
Изменяем данный параметр KRB5RCACHETYPE="none"

Затем в файле local.conf находящемся по пути /etc/bind/local.conf **комментируем зону прямого просмотра** следующим образом:

```
// Add other zones here

/* zone "au-team.irpo" {
    type slave;
    file "slave/au-team.irpo.db";
    masters { 192.168.1.2; };
};
*/
```

Затем **удаляем конфигурацию Samba** следующими командами:

```
rm -rf /etc/samba/smb.conf
rm -rf /var/lib/samba
rm -rf /var/cache/samba
mkdir -p /var/lib/samba/sysvol
```

Затем **запускаем полуинтерактивную установку домена** следующей командой:

```
samba-tool domain provision
```

Тут первые три интером скипаем, если в квадратных скобках тоесть выбранные правильные.

На четвертой BIND9_DLZ

И задаем пароль админу P@ssw0rd

Затем **добавляем samba в автозагрузку**

```
systemctl enable --now samba
```

Затем **проверяем** командой

```
samba-tool domain info 127.0.0.1
```

Создаём группы

```
samba-tool group add HQ
```

Создаем пользователей

```
for i in {1..5}; do
samba-tool user add user$i.hq P@ssw0rd
samba-tool user setexpiry user$i.hq --noexpiry
samba-tool group addmembers "HQ" user$i.hq
done
```

Повышаем права пользователей группы **hq** в файле `sudoers` по пути `/etc/sudoers`
Под `SUDO_USERS`
`%hq ALL=(ALL) NOPASSWD: /bin/cat, /bin/grep, /usr/bin/id`

```
## Same thing without a password
WHEEL_USERS ALL=(ALL:ALL) NOPASSWD: ALL
sshuser ALL=(ALL) NOPASSWD: ALL
## Uncomment to allow members of group sudo to execute any command
SUDO_USERS ALL=(ALL:ALL) ALL
%hq ALL=(ALL) NOPASSWD: /bin/cat, /bin/grep, /usr/bin/id
```

Создаем скрипт для импорта

```
while IFS=, read -r username password fullname; do
samba-tool user create "$username" "$password" --given-name="$fullname"
samba-tool group addmembers "hq" "$username"
done < /opt/users.csv
```

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 автосмонтирование в папку `/mnt/nfs`
- Основные параметры сервера отметьте в отчёте

Настройка файлового хранилища

Добавляем 3 доп хард диска

Устанавливаем пакет `mdadm`

```
apt-get install mdadm
```

Проверяем диск

```
lsblk
```

Обновляем суперблоки

```
mdadm --zero-superblock --force /dev/sd{b,c,d}
```

!!Выведет что не использовались для рейда!!

Удаляем старые метаданные и подпись

```
wipefs --all --force /dev/sd{b,c,d}
```

Создание рейд массива

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

, где

dev/md0 - название рейда после сборки

-l 5 - уровень рейда

-n 3 - колво дисков массива

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

Проверяем

```
lsblk
```

!!Должны под ними появиться `md0`!!

Создаем файловую систему из созданного рейда

```
mkfs -t ext4 /dev/md0
```

Создаем файл mdadm.conf
mkdir /etc/mdadm
echo "DEVICE partitions" > /etc/mdadm/mdadm.conf
mdadm --detail --scan | awk '/ARRAY/ {print} ' >> /etc/mdadm/mdadm.conf

Создаем файловую систему и монтируем рейд
Создаем директорию для монтирования массива
mkdir /mnt/raid5

Добавляем строку в /etc/fstab
/dev/md0 /mnt/raid5 ext4 defaults 0 0

Монтируем
mount -a

Проверяем монтирование командой
df -h

Настройка NFS (Network File System)

Устанавливаем nfs-server nfs-utils
apt-get install -y nfs-{server,utils}

Создаем директорию для общего доступа
mkdir /mnt/raid5/nfs

Выдаем права на чтение и запись
chmod 766 /mnt/raid5/nfs

Добавляем строку в /etc/exports
/mnt/raid5/nfs 192.168.2.0/28(rw,no_root_squash)

, где
/mnt/raid5/nfs - общий ресурс
192.168.2.0/28 - сеть для общего доступа по заданию это HQ-CLI сеть
rw - разрешение на чтение и запись
no_root_squash - отключение ограничения прав root

Экспортируем файловую систему которую прописали ранее
exportfs -ar
, где

-a - экспортировать все указанные каталоги
-r - повторный экспорт всех каталогов синхронизируя /var/lib/nfs/etab с /etc/exports и файлами в /etc/exports.d
-v - подробный вывод

Запускаем и добавляем в автозагрузку
systemctl enable --now nfs-server

3. Настройте службу сетевого времени на базе сервиса chrony

- В качестве сервера выступает **HQ-RTR**
- На HQ-RTR настройте сервер chrony, выберите стратум 5 43
- В качестве клиентов настройте HQ-SRV, HQ-CLI, BR-RTR, BR-SRV

Устанавливаем пакет chrony
apt-get install chrony

На HQ-RTR приводим начало файла /etc/chrony.conf к следующему виду:

```
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (https://www.pool.ntp.org/join.html)  
#pool pool.ntp.org iburst
```

```
server 127.0.0.1 iburst prefer  
hwtimestamp *  
local stratum 5  
allow 0/0
```

server 127.0.0.1 - указываем сервером синхронизации самого себя
iburst - принудительно отправляет пакеты для точности синхронизации
prefer - отдает приоритет этому серверу
hwtimestamp * - указывает сетевой интерфейс как собственный источник времени и синхронизирует клиентов с ним
local stratum 5 - указание иерархического уровня
allow 0/0 - разрешает подключение с любого IP-адреса

После чего **добавляем в автозагрузку** chrony

```
systemctl enable --now chronyd
```

Проверка:

```
chronyc sources
```

Приводим начало файла /etc/chrony.conf на HQ-SRV, HQ-CLI, BR-RTR, BR-SRV к следующему виду:

```
# Use public servers from the pool.ntp.org project.  
# Please consider joining the pool (https://www.pool.ntp.org/join.html)  
#pool pool.ntp.org iburst
```

```
server 172.16.4.2
```

4. Сконфигурируйте ansible на сервере BR-SRV

- Сформируйте файл инвентаря, в инвентарь должны входить HQ-SRV, HQ-CLI, HQ-RTR и BR-RTR
- Рабочий каталог ansible должен располагаться в /etc/ansible
- Все указанные машины должны без предупреждений и ошибок отвечать pong на команду ping в ansible посланную с BR-SRV

Конфигурация SSH

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

```
Port 2024  
MaxAuthTries 2  
PubkeyAuthentication yes  
PasswordAuthentication yes  
Banner /etc/openssh/bannermotd  
AllowUsers sshuser
```

Устанавливаем необходимые пакеты:

```
apt-get install -y ansible sshpass
```

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

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

Далее заполняем инвентарный файл /etc/ansible/inventory.yml:

```
all:
  children:
    Networking:
      hosts:
        hq-rtr:
        br-rtr:
    Servers:
      hosts:
        hq-srv:
          ansible_host: 192.168.1.2
          ansible_port: 2024
    Clients:
      hosts:
        hq-cli:
          ansible_host: 192.168.2.2
          ansible_port: 2024
```

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

```
cd /etc/ansible
mkdir group_vars
touch group_vars/{all.yml,Networking.yml}
```

Редактируем их:

```
ansible_ssh_user: sshuser
ansible_ssh_pass: P@ssw0rd
ansible_python_interpreter: /usr/bin/python3
all.yml
```

```
ansible_connection: network_cli
ansible_network_os: ios
Networking.yml
```

Выполняем команду для ping`а всех машин:

```
ansible -m ping all
-m (--module-name) - параметр для указания модуля
```

ping - модуль

all - выполнить модуль для всех виртуальных машин, указанных в инвентарном файле

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.

Конфигурация файла Docker-Compose

Останавливаем службу ahttpd, которая занимает порт 8080:

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

ahttpd - модуль для веб-интерфейса, который предназначен для управления настройками web-сервера, обеспечивающего работоспособность Центра управления системой

Устанавливаем docker и docker-compose:

```
apt-get install -y docker-{ce,compose}
```

Включаем и добавляем в автозагрузку docker:

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

В домашней директории пользователя создаем файл wiki.yml и прописываем следующее:

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

  db:
    container_name: mariadb
    image: mariadb
    restart: always
    environment:
      MARIADB_DATABASE: mediawiki
      MARIADB_USER: wiki
      MARIADB_PASSWORD: WikiP@ssw0rd
      MARIADB_ROOT_PASSWORD: P@ssw0rd
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

где

services - основной раздел, в котором описываются сервисы

container_name - имя контейнера

image - имя образа

restart - перезапуск контейнера, если он остановлен

ports - проброс портов

links - ссылка на контейнер

volumes - проброс папок

environment - переменные окружения

Собираем стек контейнеров:

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

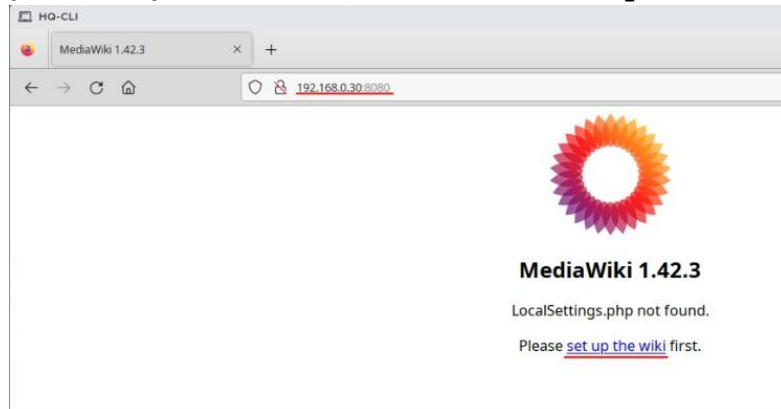
-f - указание на файл

up - запуск

-d - запуск в фоновом режиме

Установка MediaWiki в веб-интерфейсе

На **HQ-CLI** в браузере вводим **http://192.168.0.30:8080** и начинаем установку **MediaWiki**, нажав на **set up the wiki**:



Выбираем язык:

Проверяем внешнюю среду и нажимаем **далее**:

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

Тип базы данных:

☒ [MariaDB, MySQL или совместимая](#)

☐ SQLite

Настройки MariaDB/MySQL

Хост базы данных:

[справка](#)

db

☐ 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 →](#)

Оставляем галочку и жмем **далее**:

Настройки базы данных

Учётная запись для доступа к базе данных из веб-сервера

☒ Использовать ту же учётную запись, что и для установки

[← Назад](#) [Далее →](#)

Заполняем информацию об учетной записи администратора:

Название вики:

[справка](#)

wiki

Пространство имён проекта:

☒ [справка](#)

☒ То же, что имя вики: [Wiki](#)

☐ Проект

☐ Другое (укажите)

Учётная запись администратора

Ваше имя участника:

[справка](#)

admin

Пароль:

Пароль ещё раз:

Адрес электронной почты:

[справка](#)

admin@au-team.irpo

☐ Подписаться на рассылку новостей о появлении новых версий MediaWiki

[справка](#)

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

[справка](#)

Информация

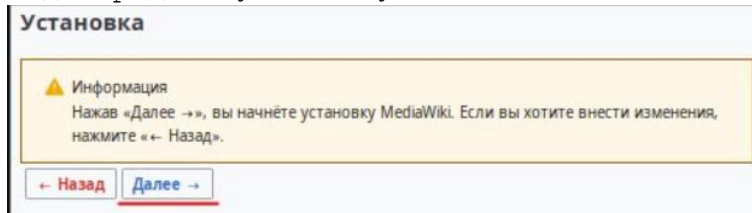
Вы почти у цели! Остальные настройки можно пропустить и приступить к установке вики.

☐ Произвести тонкую настройку

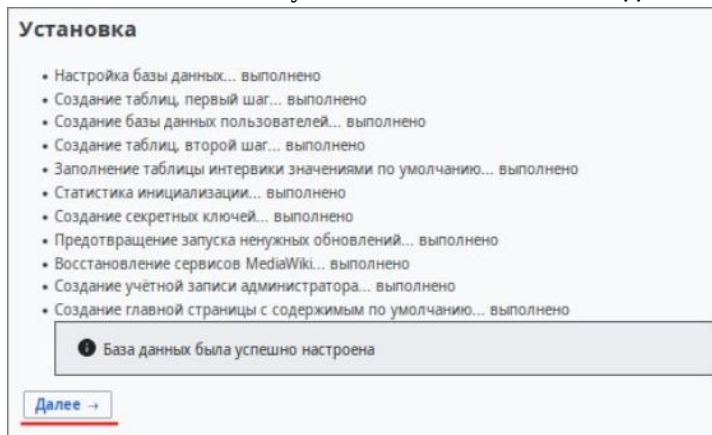
☒ Хватит уже, просто установите вики.

[← Назад](#) [Далее →](#)

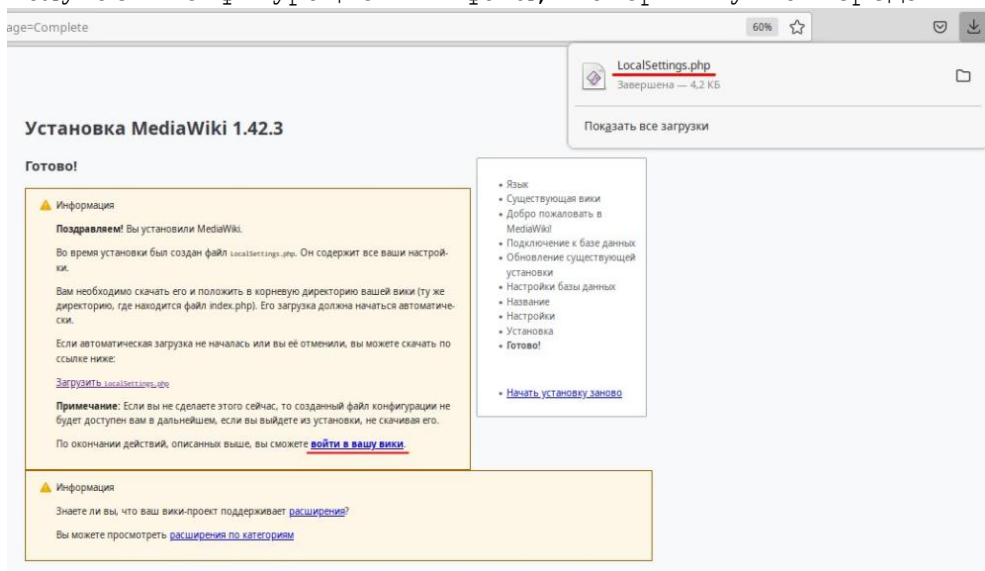
Подтверждаем установку **MediaWiki**:



После окончания установки нажимаем **далее**:



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



Правка файла Docker-Compose

Перемещаем файл LocalSettings.php в домашнюю директорию пользователя sshuser:

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

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

Передаем файл с HQ-CLI на BR-SRV:

```
scp -P 2024 /home/sshuser/LocalSettings.php sshuser@192.168.0.2:/home/sshuser
```

-P - указание порта SSH

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

sshuser@192.168.0.2:/home/sshuser - имя-пользователя@IP-адрес:директория-назначения

На BR-SRV перемещаем файл в домашнюю директорию root:

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

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

В файле wiki.yml раскомментируем следующие строки:

volumes:

- ./LocalSettings.php:/var/www/html/LocalSettings.php

Перезапускаем запущенные Docker`ом сервисы:

```
docker compose -f wiki.yml stop
```

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

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 44
- Пробросьте порт 2024 в порт 2024 на HQ-SRV на маршрутизаторе HQ-RTR
- Пробросьте порт 2024 в порт 2024 на BR-SRV на маршрутизаторе BR-RTR

Конфигурация BR-RTR

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

```
ip nat source static tcp 192.168.0.1 80 192.168.0.30 8080
```

Проброс портов с 2024 на 2024:

```
ip nat source static tcp 192.168.0.1 2024 192.168.0.30 2024
```

Конфигурация HQ-RTR

Проброс портов с 2024 на 2024:

```
ip nat source static tcp 192.168.100.1 2024 192.168.100.62 2024
```

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

Конфигурация базы данных

Устанавливаем необходимые пакеты:

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

Добавляем в **автозагрузку** базу данных:

```
systemctl enable --now mysqld
```

Задаем пароль для пользователя **root** в базе данных:

```
mysqladmin password 'P@ssw0rd'
```

Редактируем настройки **веб-сервера**:

```
cat /etc/httpd2/conf/include/Directory_moodle_default.conf | grep 'Require all granted' || sed -i '/AllowOverride None/a Require all granted' /etc/httpd2/conf/include/Directory_moodle_default.conf
```

Изменяем строку, отвечающую за количество входных переменных:

```
sed -i 's/; max_input_vars = 1000/max_input_vars = 5000/g' /etc/php/8.2/apache2-mod_php/php.ini
```

Добавляем в автозагрузку веб-сервер:

```
systemctl enable --now httpd2
```

Авторизуемся в MySQL:

```
mysql -u root -p
```

Вводим ранее указанный пароль

Создаем пользователя для базы данных:

```
create user 'moodle'@'localhost' identified by 'P@ssw0rd';
```

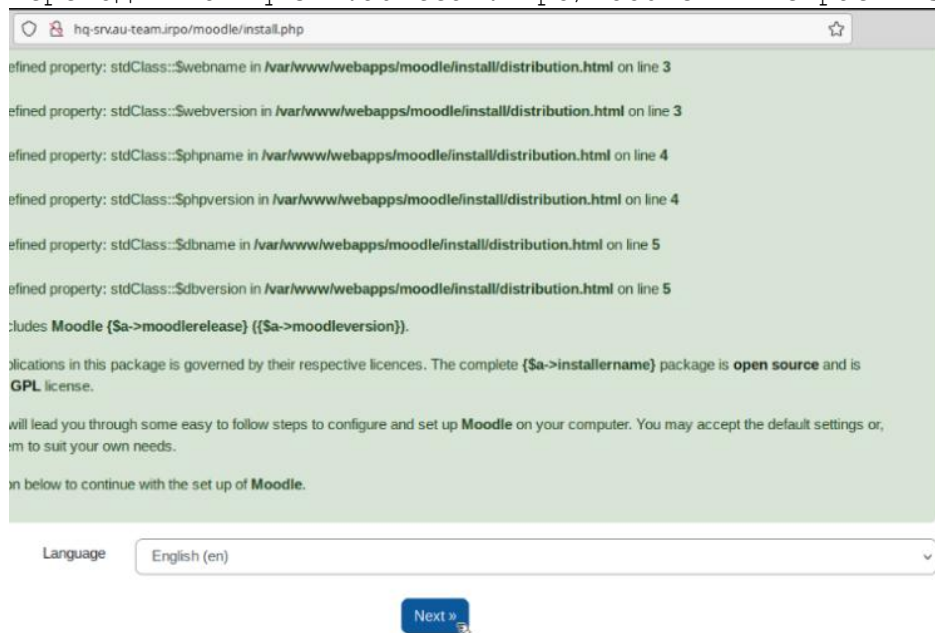
Создаем базу данных:

```
create database moodledb default character set utf8 collate utf8_unicode_ci;
```

Выдаем права пользователю на созданную базу данных:

```
grant all privileges on moodledb.* to moodle@localhost;
```

Переходим на `hq-srv.au-team.irpo/moodle` и выбираем язык:



The screenshot shows the Moodle installation page at `hq-srv.au-team.irpo/moodle/install.php`. The page has a light green background with black text. It contains several lines of code snippets, a paragraph about the GPL license, and a paragraph about the installation steps. At the bottom, there is a 'Language' dropdown menu set to 'English (en)' and a blue 'Next »' button.

Подтверждаем пути директорий:



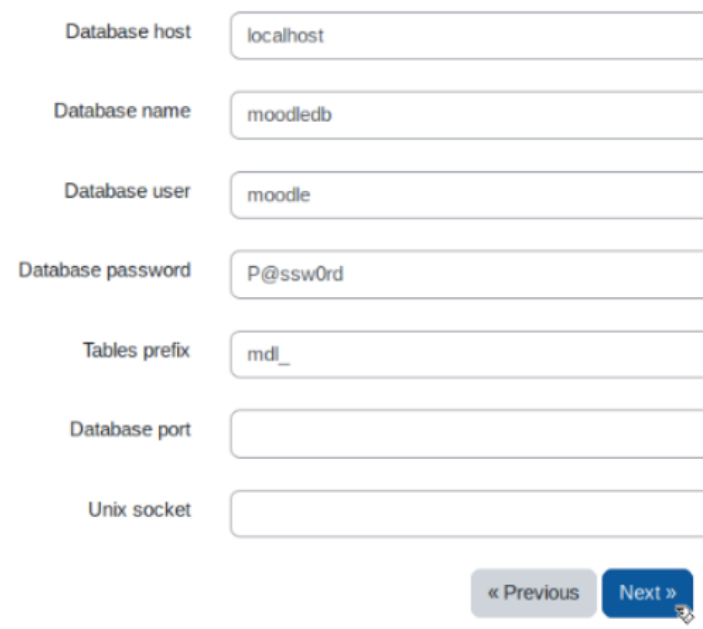
The screenshot shows the Moodle installation page with three input fields for directory paths: 'Web address' (http://hq-srv.au-team.irpo/moodle), 'Moodle directory' (/var/www/webapps/moodle), and 'Data directory' (/var/lib/moodle/default). At the bottom, there are '« Previous' and 'Next »' buttons.

Выбираем систему управления базы данных:



The screenshot shows the Moodle installation page with a 'Type' dropdown menu set to 'MariaDB (native/mariadb)'. At the bottom, there are '« Previous' and 'Next »' buttons.

Заполняем данные о базе данных и пользователе:

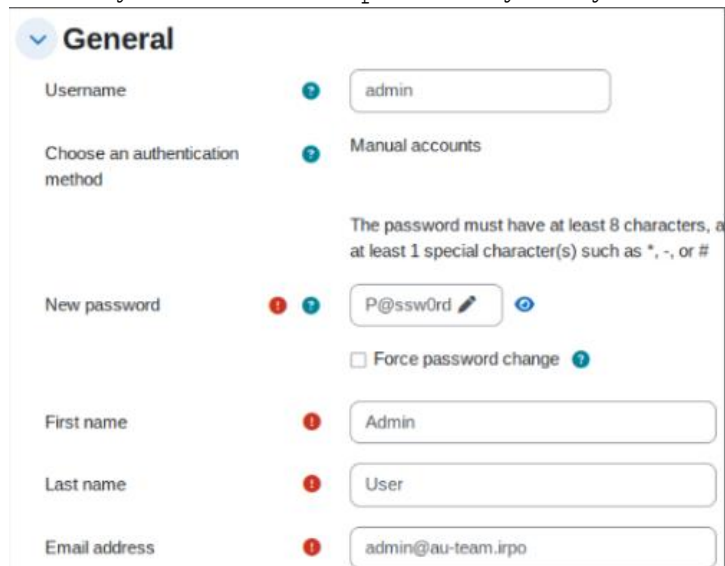


The screenshot shows the Moodle installation page with several input fields for database configuration: 'Database host' (localhost), 'Database name' (moodledb), 'Database user' (moodle), 'Database password' (P@ssw0rd), 'Tables prefix' (mdl_), 'Database port' (empty), and 'Unix socket' (empty). At the bottom, there are '« Previous' and 'Next »' buttons.

Соглашаемся с условиями

Убеждаемся в успешной проверке

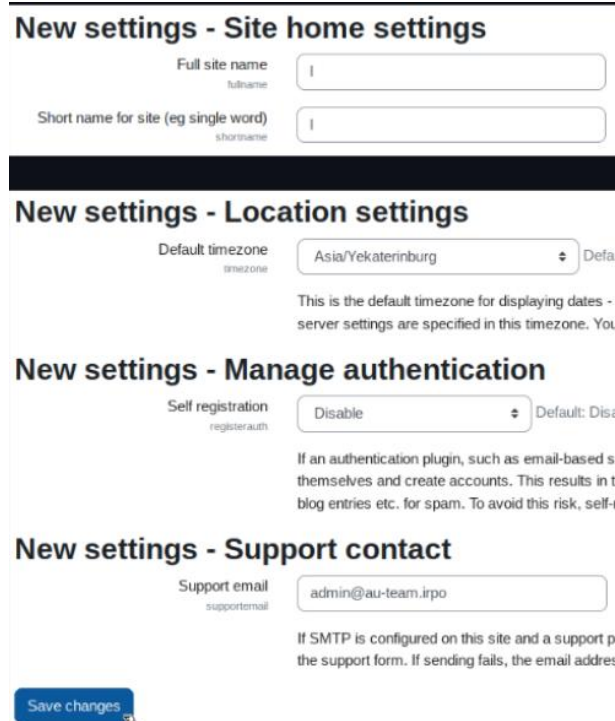
После установки настраиваем учетную запись администратора:



The screenshot shows the 'General' settings page for the administrator account. The fields are as follows:

- Username:** admin
- Choose an authentication method:** Manual accounts
- New password:** P@ssw0rd (with a strength indicator showing 1 red and 2 blue dots). A note states: 'The password must have at least 8 characters, at least 1 special character(s) such as *, -, or #'. There is an unchecked checkbox for 'Force password change'.
- First name:** Admin
- Last name:** User
- Email address:** admin@au-team.irpo

Указываем название сайта, часовой пояс и электронную почту:



The screenshot shows the 'New settings' page with three sections:

- New settings - Site home settings:** Fields for 'Full site name' and 'Short name for site (eg single word)' both contain the letter 'I'.
- New settings - Location settings:** 'Default timezone' is set to 'Asia/Yekaterinburg'. A note explains: 'This is the default timezone for displaying dates - server settings are specified in this timezone. You'.
- New settings - Manage authentication:** 'Self registration' is set to 'Disable'. A note explains: 'If an authentication plugin, such as email-based s themselves and create accounts. This results in t blog entries etc. for spam. To avoid this risk, self-'.

At the bottom, there is a 'Support email' field with 'admin@au-team.irpo' and a 'Save changes' button.

После успешного создания попадаем на главную страницу

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 mediwiki

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

- Установку браузера отметьте в отчёте

Необходимые приложения: Приложение А. Инструкция и список чекпойнтов по настройке оборудования для проведения ДЭ (в отдельном файле). Приложение Б. Файл users.csv.