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

apt-get update && apt-get install task-samba-dc rm -rf /etc/samba/smb.conf echo nameserver 192.168.100.2 > /etc/resolv.conf

После переходите на HQ-SRV в настройку dnsmasq и добавляете: server=/au-team.irpo/192.168.100.66



А теперь запускаем автонастройку доменного контроллера на **BR-SRV**. Если предложенные значения верны, те, что находятся в [], то нажимаем Enter, если нет, то нужно проверять предыдущие настройки. **samba-tool domain provision**

**AU-TEAM.IRPO** **AU-TEAM**

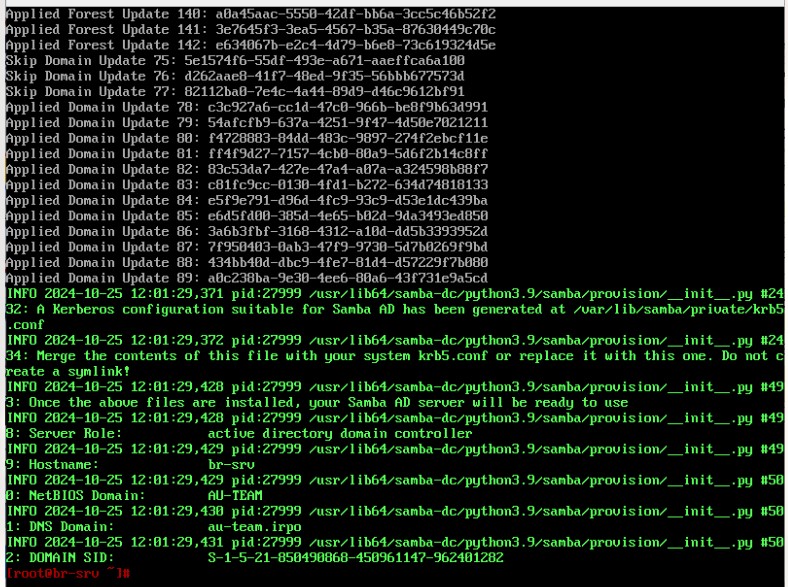
**dc**

**SAMBA\_INTERNAL**

**192.168.100.2** (Здесь вводим значение вручную)

**P@ssw0rd**

После настройки должно появится такое в терминале, это значит, что всё настроено верно:



Перемещаем сгенерированный конфиг krb5.conf и включаем службу samba:

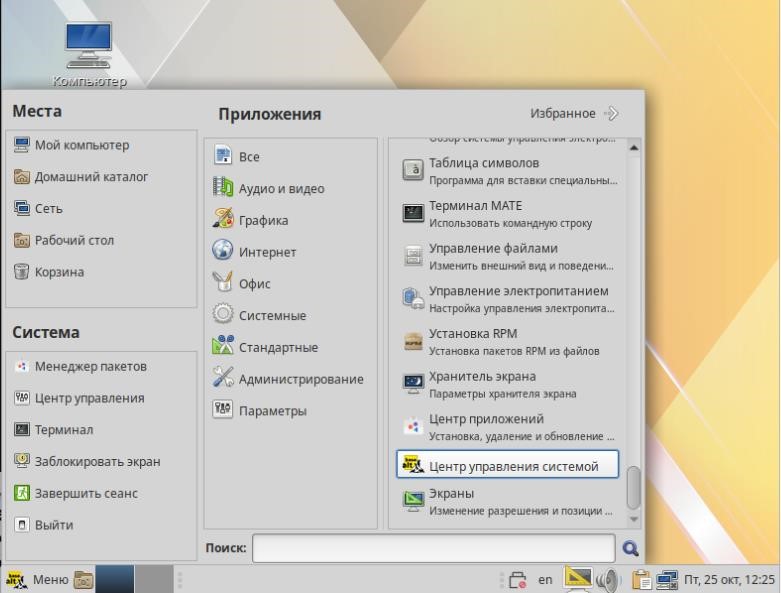
**mv -f /var/lib/samba/private/krb5.conf /etc/krb5.conf** **systemctl enable --now samba**

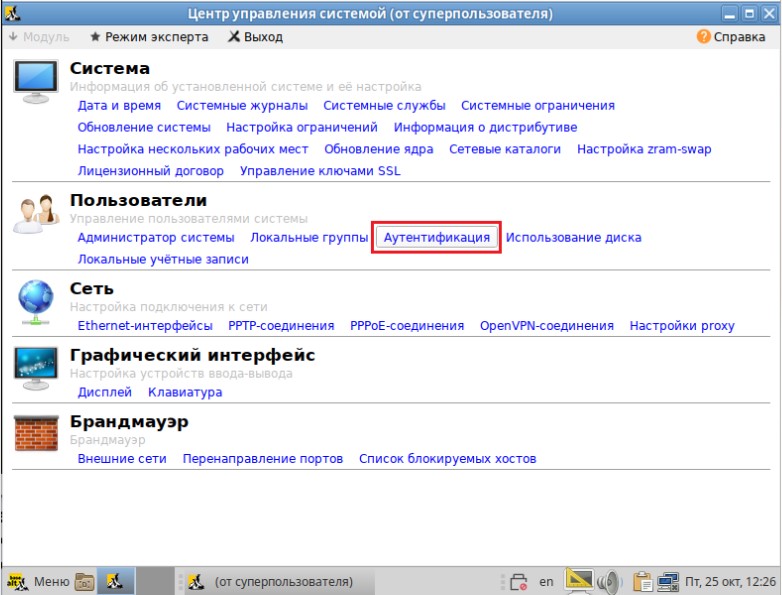
Теперь создадим **5** пользователей: **samba-tool user add user1.hq P@ssw0rd** **samba-tool user add user2.hq P@ssw0rd** **samba-tool user add user3.hq P@ssw0rd** **samba-tool user add user4.hq P@ssw0rd** **samba-tool user add user5.hq P@ssw0rd**

Теперь создадим группу и поместим туда созданных пользователей:

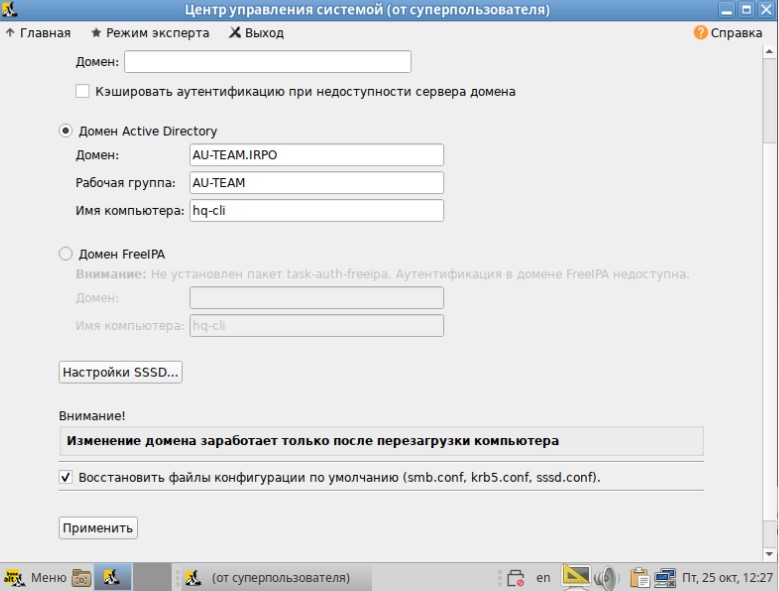
**samba-tool group add hq** **samba-tool group addmembers hq user1.hq,user2.hq,user3.hq,user4.hq,user5.hq**

Теперь введём клиентскую машину в домен:

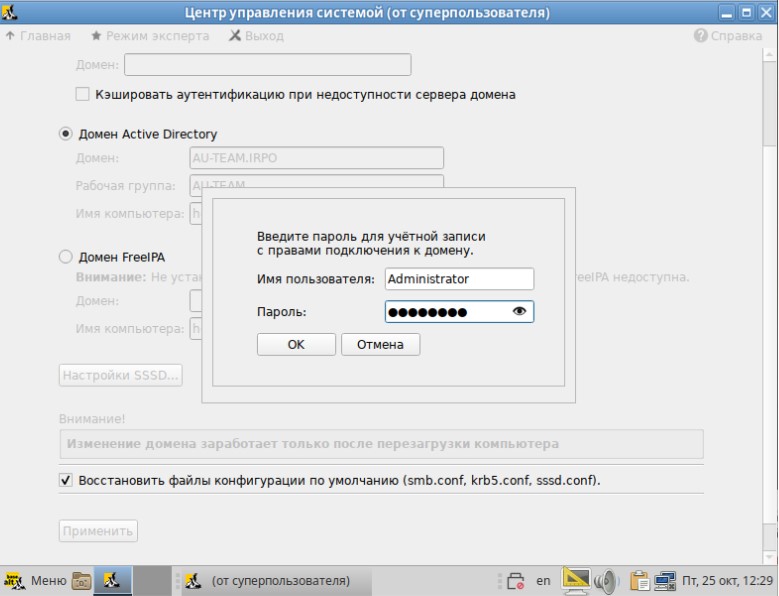




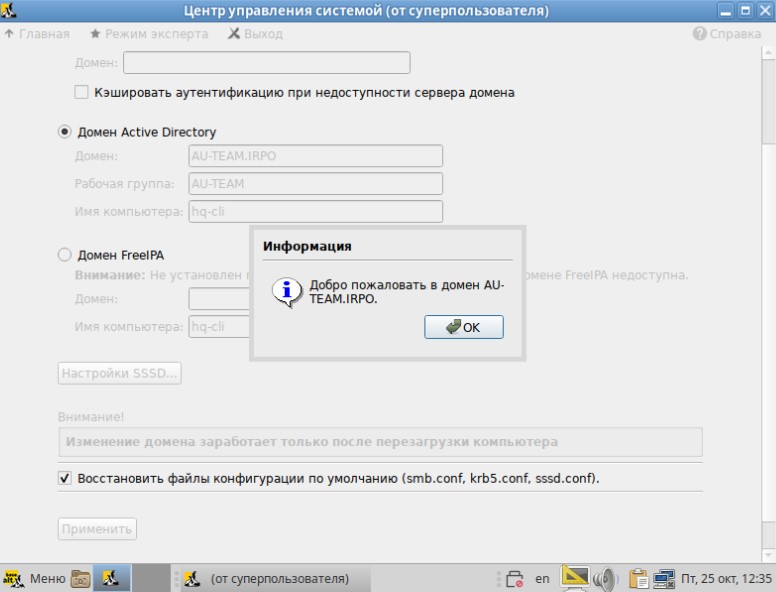
Заполняем так, как показано на скриншоте ниже:



Вводим пароль, который вводили при настройке домена через samba-tool:



После ввода в домен должно появиться следующее сообщение на экране:



Перезагружаем HQ-CLI

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

**apt-repo add rpm** [**http://altrepo.ru/local-p10 noarch local-p10**](http://altrepo.ru/local-p10%20noarch%20local-p10)

Теперь обновляем список пакетов:

**apt-get update**

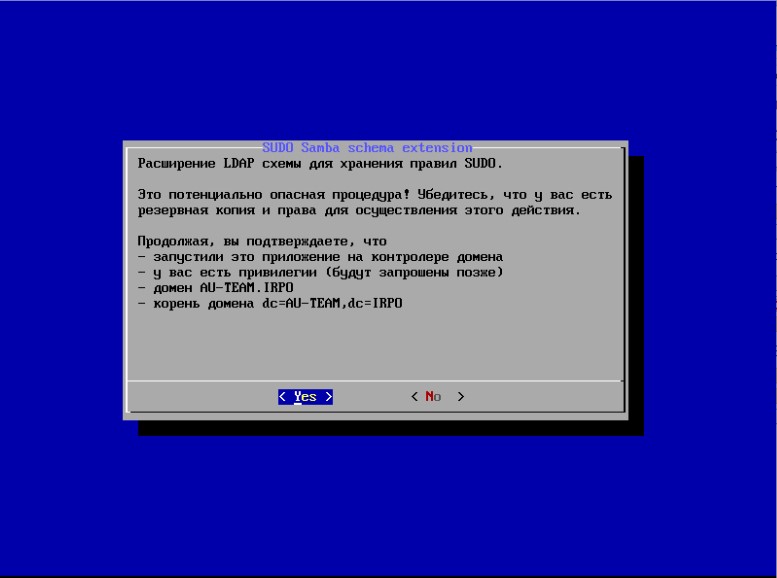
И можем устанавливать нужный нам пакет:

**apt-get install sudo-samba-schema**

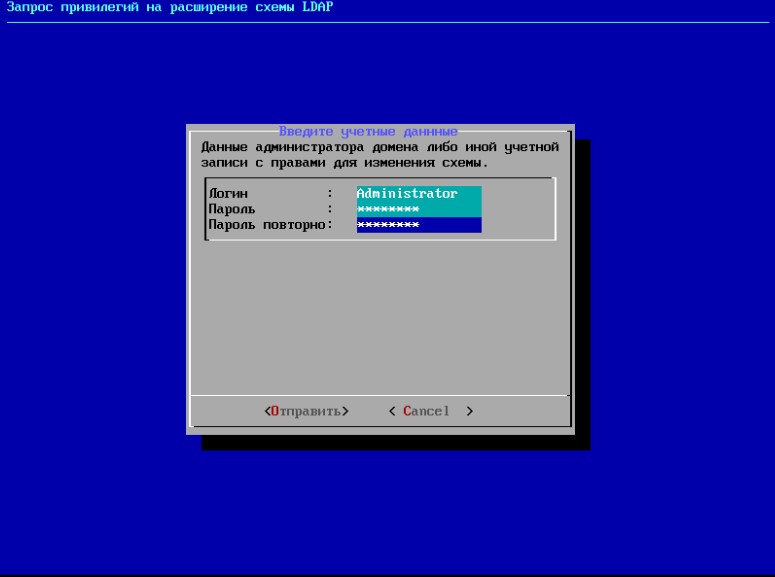
Далее добавляем новую схему следующей командой:

**sudo-schema-apply**

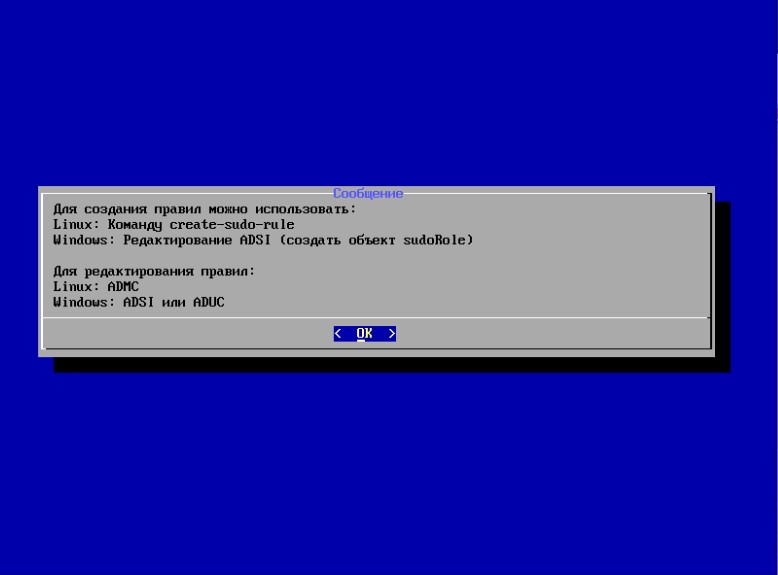
Откроется следующее диалоговое окно, нажимаем yes:



Затем у нас попросит [пароль](https://docs.google.com/document/d/1gw-aewMqPLfv1nFOYQZU4Svufjf3aajY/edit#bookmark=id.hgqxtshk1u0) от доменного администратора



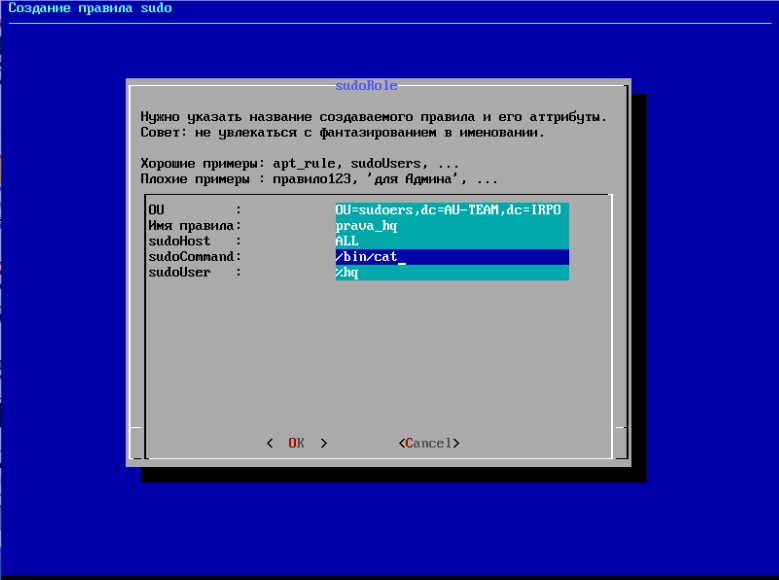
После этого должно появиться такое окно:



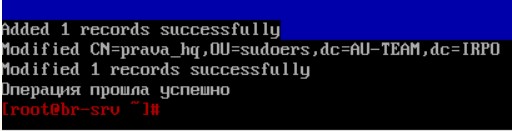
Далее мы создаём новое правило следующей командой (которую он сам предлагает в этом окне):

**create-sudo-rule**

И вносим следующие изменения (имя правила можно любое): Имя правила : **prava\_hq** sudoCommand : **/bin/cat** sudoUser : **%hq**



При успешном добавлении выведет следующие строки:



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

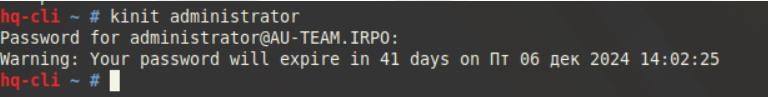
Обновляем список пакетов:

**apt-get update**

И поставим пакет admc: **apt-get install admc**

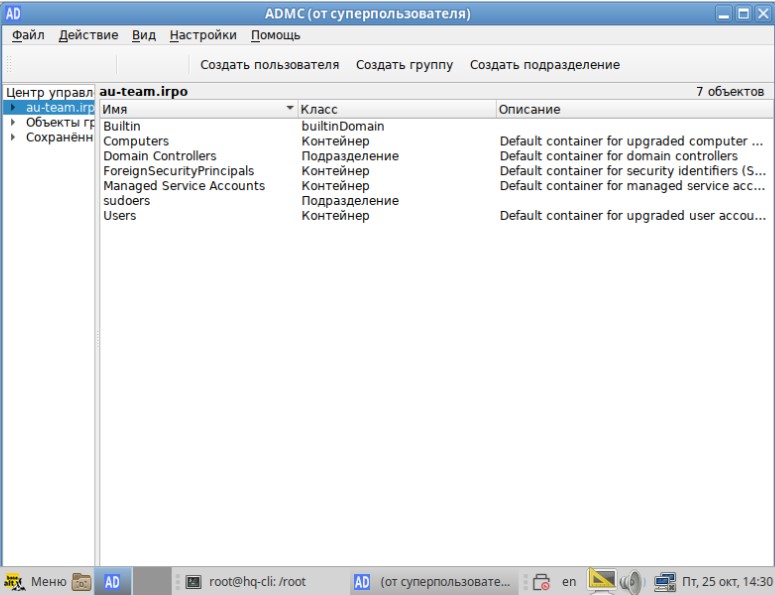
Затем создаём тикет доменного администратора, чтобы получить права на редактирование правил на сервере: **kinit administrator**

**P@ssw0rd**

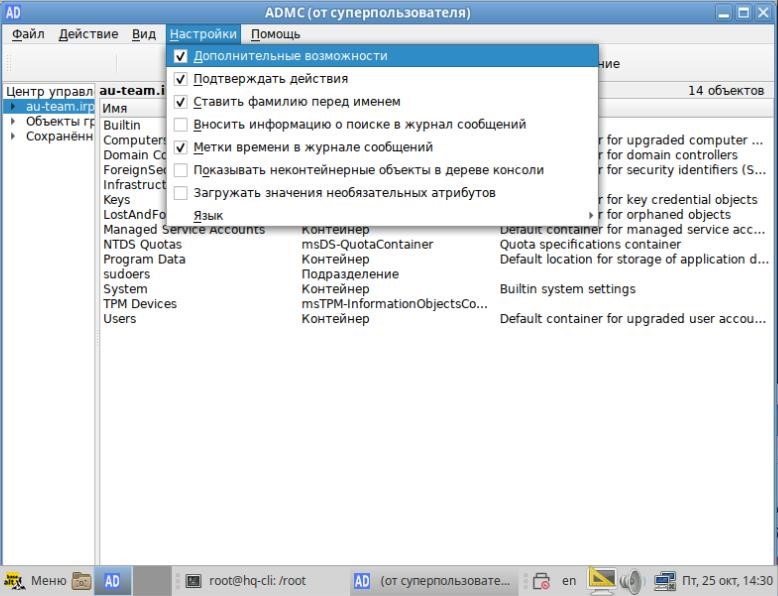


И запускаем admc:

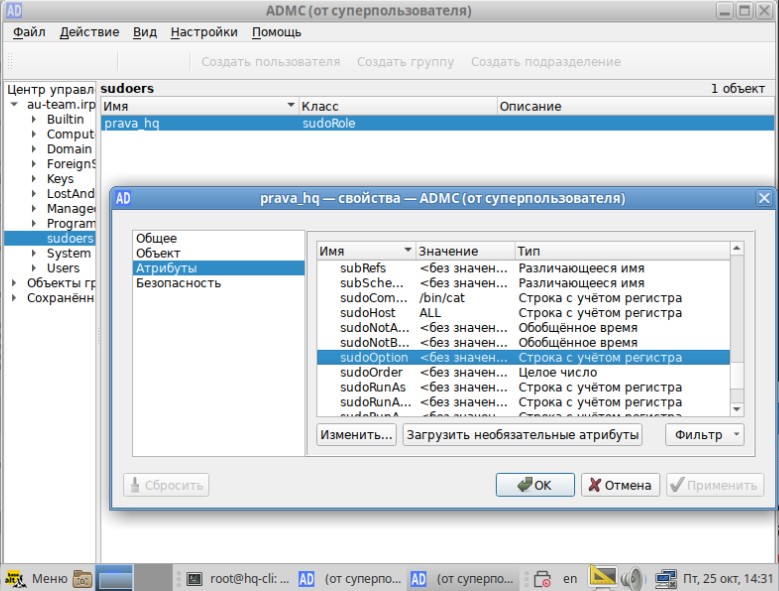
**admc**



Включим дополнительные возможности через настройки:

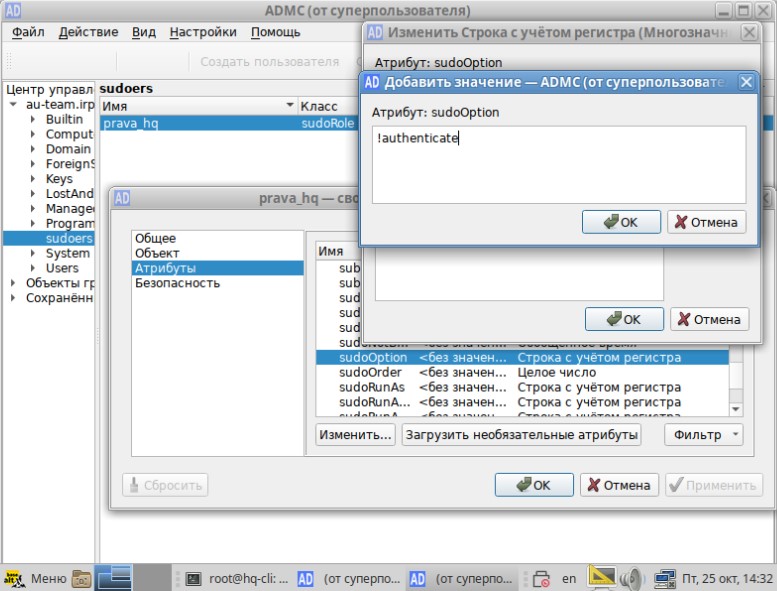


Поменяем опцию **sudoOption** в созданном нами ранее правиле **prava\_hq** (правило всегда будет находиться в OU с названием **sudoers)**:

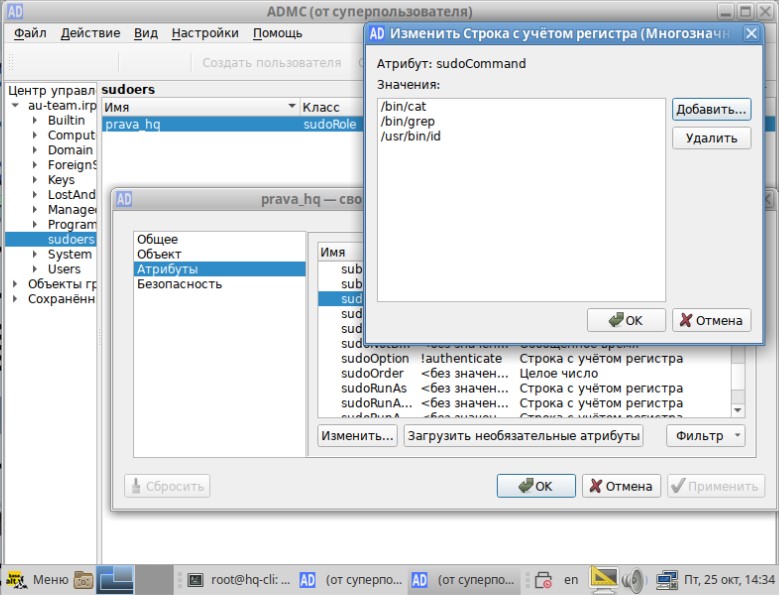


Новое значение будет:

**!authenticate**



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



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

После перезагружаем HQ-CLI.

После перезагрузки нам нужно перейти на HQ-CLI в сессию root сочетанием клавиш Ctrl+Alt+F2, если хотите обратно в графическую оболочку Сtrl+Alt+F1

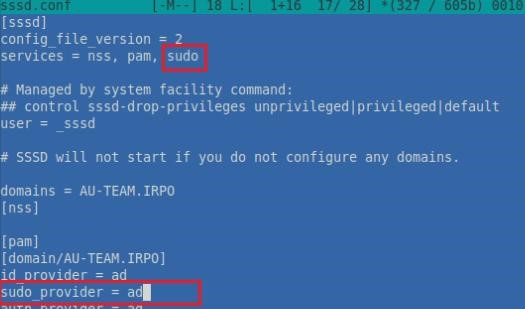
Теперь, чтобы работали все созданные нами правила, нужно зайти на **HQ-CLI** и установить дополнительные пакеты:

**apt-get update**

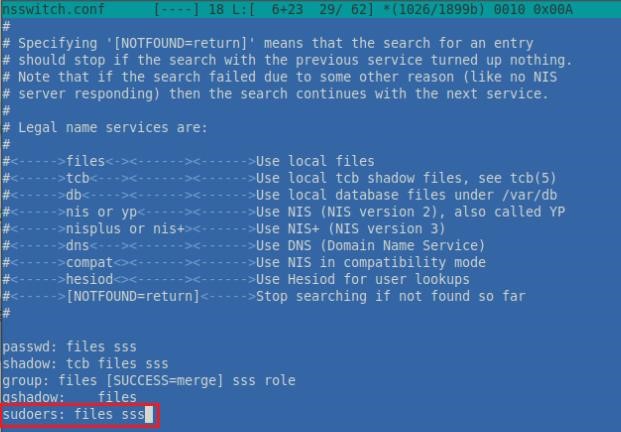
**apt-get install sudo libsss\_sudo**

Разрешаем использование sudo: **control sudo public**

Настроим конфиг **sssd.conf**: **mcedit /etc/sssd/sssd.conf** **services = nss, pam, sudo** **sudo\_provider = ad**



Теперь отредактируем **nsswitch.conf**: **mcedit /etc/nsswitch.conf** **sudoers: files sss**



Теперь перезагрузим нашу клиентскую машину **HQ-CLI**.

На всякий случай, нужно очистить кэш и удалить остаточные файлы, чтобы всё перезаписалось и применилось, для этого пишем следующие команды:

**rm -rf /var/lib/sss/db/\*** **sss\_cache -E**

И перезагружаем службу **sssd**:

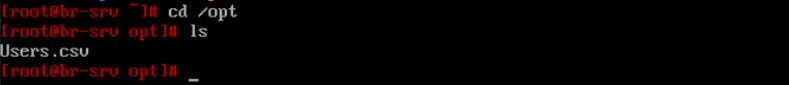
**systemctl restart sssd**

Настройка прав завершена!

Настройка прав для группы **hq** завершена!

Приступаем к следующему этапу – **импортируем** пользователей из таблицы **Users.csv**.

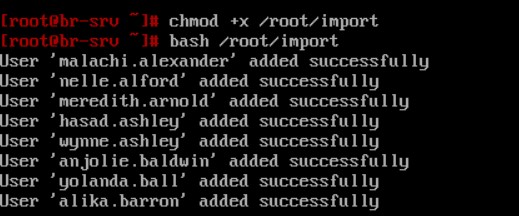
Файл хранится в директории /opt/



Создаём файл **import** и пишем туда следующий код: mcedit import



Сохраняем этот файл и выдаём ему право на выполнение и запускаем его: **chmod +x import** **bash import**



Добавляться они будут долго, но мы это сделали! Успех!

Если есть желание, проверьте, зайдя под одним из этих пользователей через клиентскую машину.

2. Конфигурация файлового хранилища на HQ-SRV

На HQ-SRV проверяете диск через lsblk



Нас интересуют диски vdb,vdc,vdd, каждый по 1 гб

Теперь создадим дисковый массив уровня 5 из трёх дополнительных дисков следующей командой:

**mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/vd[b-d]**

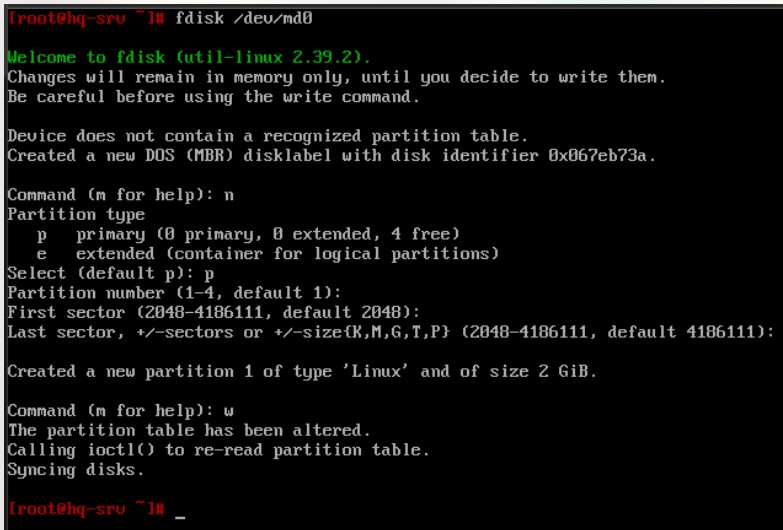
Сохраним конфигурацию массива в файл **/etc/mdadm.conf** следующей командой:

**mdadm --detail -scan --verbose > /etc/mdadm.conf** Теперь создаём раздел через **fdisk**.

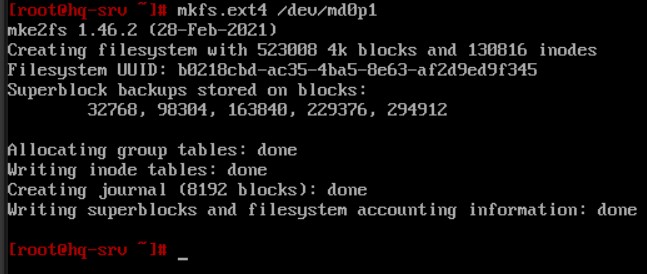
Для этого пишем следующую команду: **fdisk /dev/md0**



Затем пишем **n**, чтобы создать раздел, прокликиваем **Enter**, потому что он по дефолту предлагает то, что нам нужно, а в конце пишем **w**, чтобы записать изменения:

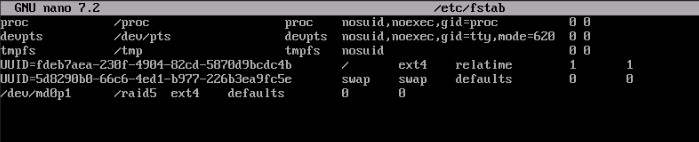


Теперь создадим файловую систему, по заданию требуется **ext4**, создаём её следующей командой: **mkfs.ext4 /dev/md0p1**



Теперь настроим автоматическое монтирование в **/raid5**. Добавляем следующую строку в конец файла **/etc/fstab**:

**/dev/md0p1 /raid5 ext4 defaults 0 0**



Затем создаём каталог **/raid5** и монтируем ФС из **/etc/fstab**:

**mkdir /raid5** **mount -a**

Заметьте, что команда не должна ничего выводить!



Теперь настроим сервер файловой системы **NFS**, для этого обновляем список пакетов и устанавливаем службу **nfs-server** следующей командой:

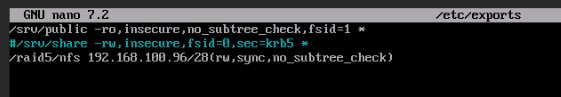
**apt-get update** **apt-get install nfs-server**

Приступаем к самой настройке, создадим каталог, назначим нового владельца и группу ему и выдадим новые права:

**mkdir /raid5/nfs** **chown 99:99 /raid5/nfs** **chmod 777 /raid5/nfs**

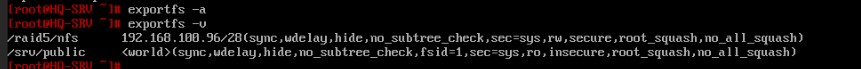
Откроем каталог для общего доступа в сторону подсети, где находится **HQ-CLI**, для этого заходим в **/etc/exports** и пишем следующую строку в конец файла:

**/raid5/nfs 192.168.100.96/28(rw,sync,no\_subtree\_check)**



После редактирования файла применяем изменения и смотрим, что она экспортировалась:

**exportfs -a** **exportfs -v**



Включаем и перезапускаем службу NFS:

**systemctl enable --now nfs**

Теперь идём монтировать этот каталог на клиенте **HQ-CLI**, для этого нужно:

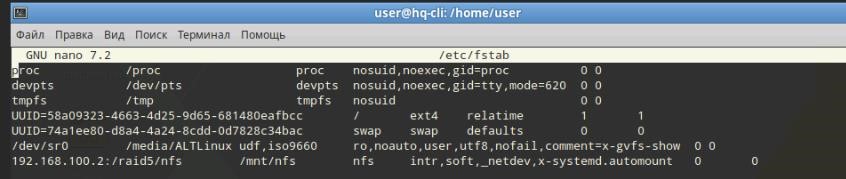
**apt-get update** **apt-get install nfs-clients**

Он может быть у вас уже установлен, но проверить нужно.

Теперь настроим автоматическое монтирование в каталог **/mnt/nfs**, но для начала создадим его:

**mkdir -p /mnt/nfs**

Добавляем следующую строку в конец файла **/etc/fstab**:



Монтируем ФС из файла **/etc/fstab** и проверяем, что она появилась в списке:

**mount -a** **mount -v**



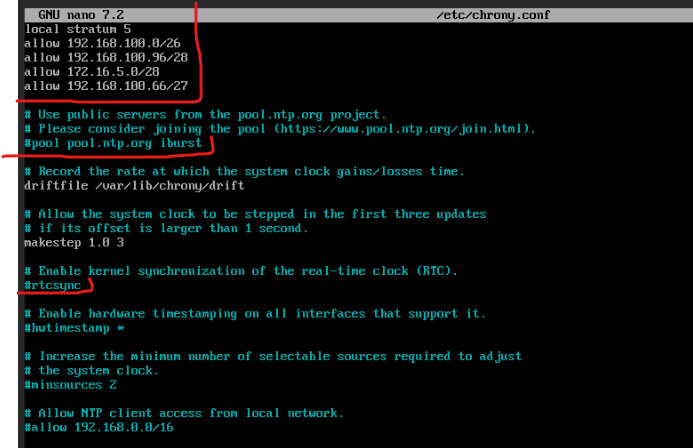
Теперь проверим и создадим файл с клиентской машине в каталоге /mnt/nfs, затем посмотрим на сервере, создался ли он:

**touch /mnt/nfs/123**



3. Настройка службы сетевого времени на базе сервиса chrony Сервер времени у нас это HQ-RTR На HQ-RTR:

**mcedit /etc/chrony.conf**



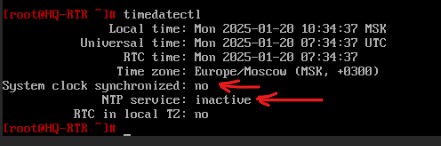
Файл должен выглядеть так, как на скрине, **закомментировав** ещё две строки, они нам нужны были только синхронизации с публичным сервером времени.

Включаем и перезапускаем службу **chrony**:

**systemctl enable --now chronyd** **systemctl restart chronyd**

Выключаем теперь ту самую синхронизацию, оставляя, по сути, главным сервером NTP – наш роутер **HQ-RTR,** и проверяем ещё раз статус timedatectl:

**timedatectl set-ntp 0** **timedatectl**



Теперь переходим к настройке клиента, в качестве него возьмём **HQ-CLI**, но по заданию есть ещё, но о них позже.

Перед установкой новой службы выключим chrony на **HQ-CLI**:

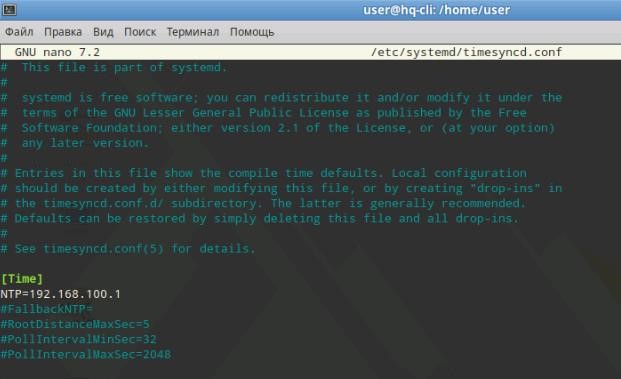
**systemctl disable --now chronyd** **systemctl status chronyd**

Обновляем список пакетов на **HQ-CLI** и скачиваем службу **systemd-timesyncd**:

**apt-get update**

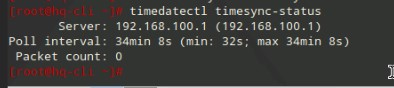
**apt-get install systemd-timesyncd**

Теперь зайдём в конфиг **/etc/systemd/timesyncd.conf** и отредактируем только одну строку: **NTP=192.168.1.1**



Теперь включим службу **systemd-timesyncd** и посмотрим её статус работы:

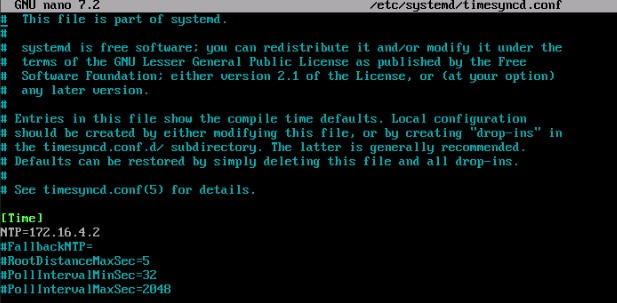
**systemctl enable --now systemd-timesyncd** **timedatectl timesync-status**



Настроим теперь **BR-RTR**, удаляем пакеты **ntp**, **chrony**, если они есть:

**apt-get install system-timesyncd**

Настроим также его конфиг в **/etc/systemd/timesyncd.conf**: **NTP=172.16.4.2**



Теперь включим службу **systemd-timesyncd** и готово:

**systemctl enable --now systemd-timesyncd**

На остальных клиентах нужно проделать тоже самое, исходя из документа остались - **HQSRV**, **BR-SRV**

Но помните, что **NTP** для **BR-SRV** – это внешний IP-адрес **HQ-RTR**, то-есть **172.16.4.2**.

4. Сконфигурируйте ansible на сервере BR-SRV

Для начала проверим, обновлены ли у нас списки пакетов и затем попробуем установить **ansible**:

**apt-get update** **apt-get install ansible**

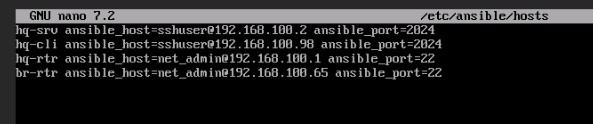
Далее нам нужен рабочий каталог для нашего ansible, который может быть уже создан, НО если нет, то создадим его следующей командой:

**mkdir /etc/ansible**

У нас он уже создан, и там же находится нужный нам файл **hosts**, тоже уже созданный, НО в случае его отсутствия, нужно также его создать:

**mcedit /etc/ansible/hosts**

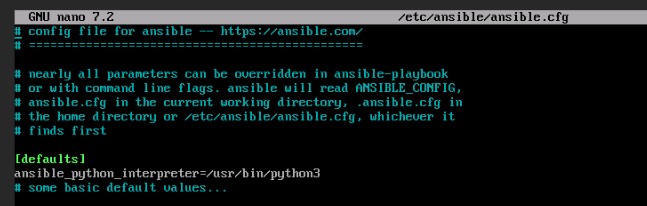
Теперь нам нужно написать следующие строки в файл **hosts**:



И настроим в каталоге **/etc/ansible** файл **ansible.cfg**:

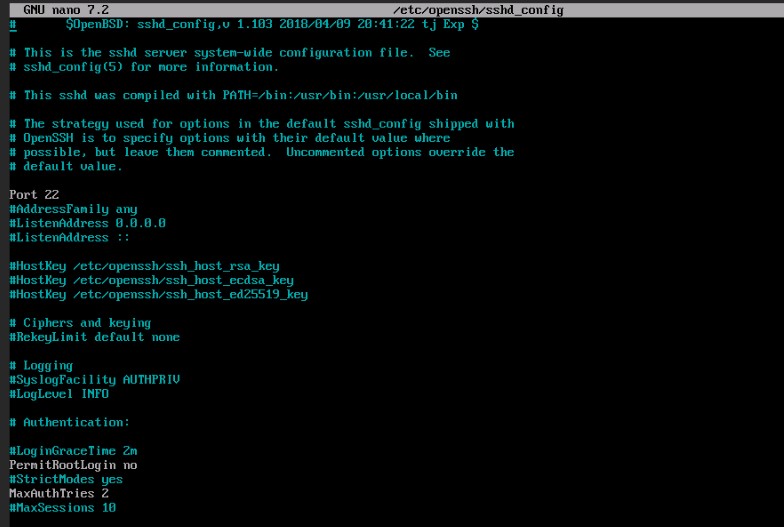
**mcedit /etc/ansible/ansible.cfg**

Добавим под строку **[defaults]** ещё одну: **ansible\_python\_interpreter=/usr/bin/python3**



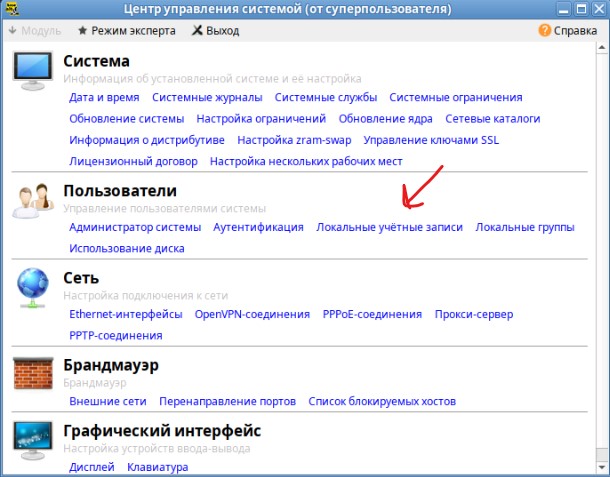
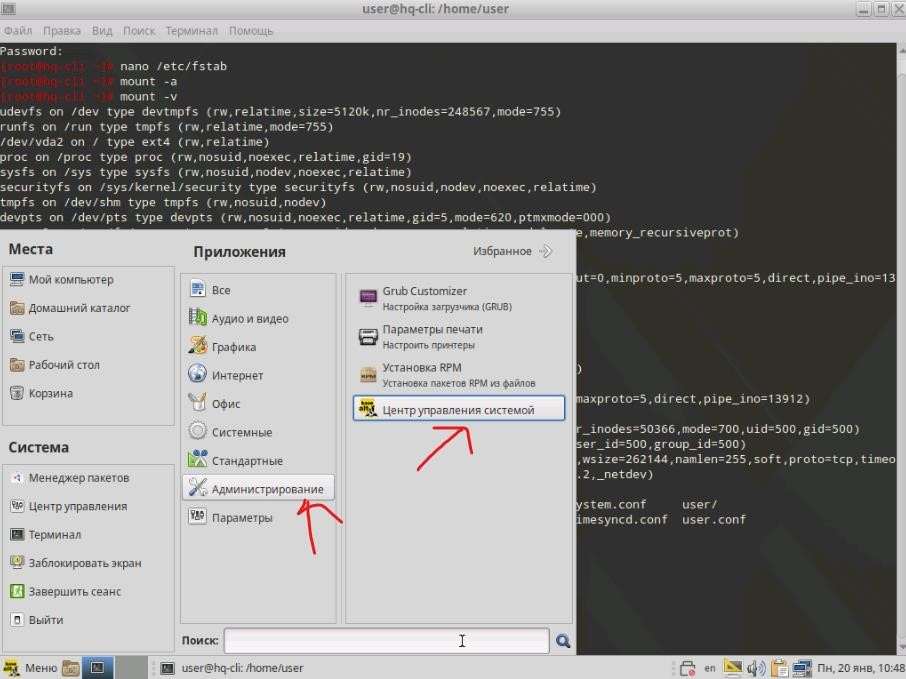
Но, возникает другая проблема, в первом модуле мы настраивали SSH на серверах, однако маршрутизаторы **HQ-RTR**, **BR-RTR** и клиент **HQ-CLI** не входили в пункт по настройке. Теперь его нужно настроить сейчас, на этих устройствах, поэтому приступаем к их настройке, чтобы мы могли выполнить задание по **ansible**.

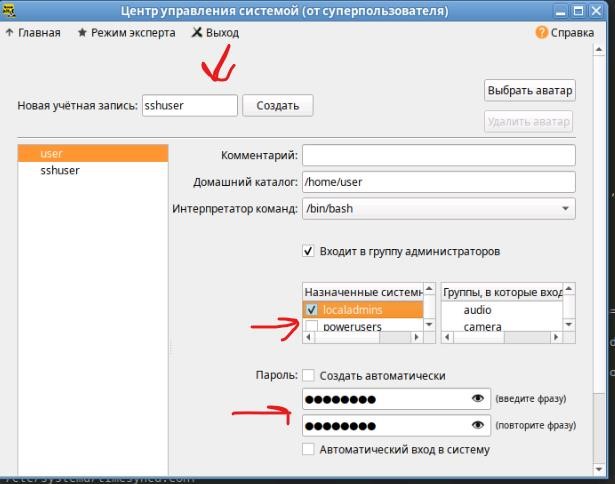
HQ-RTR:



Так же BR-RTR. На HQ-CLI указываете порт 2024

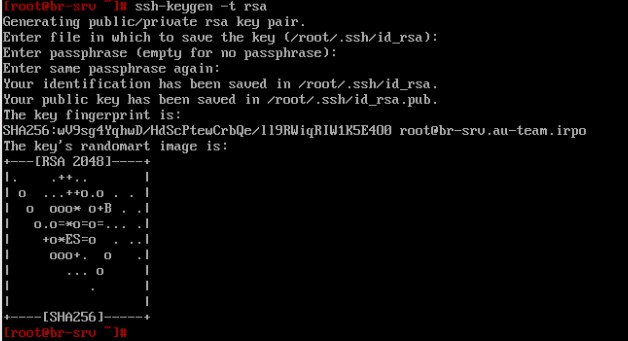
На HQ-CLI вам нужно создать пользователя sshuser





После через команду passwd меняете пароль у sshuser на P@ssw0rd

Теперь на **BR-SRV** генерируем ключи **RSA**, чтобы экспортировать их на машины клиенты, строку с путём и **passphrase** оставляем пустой: **ssh-keygen -t rsa**



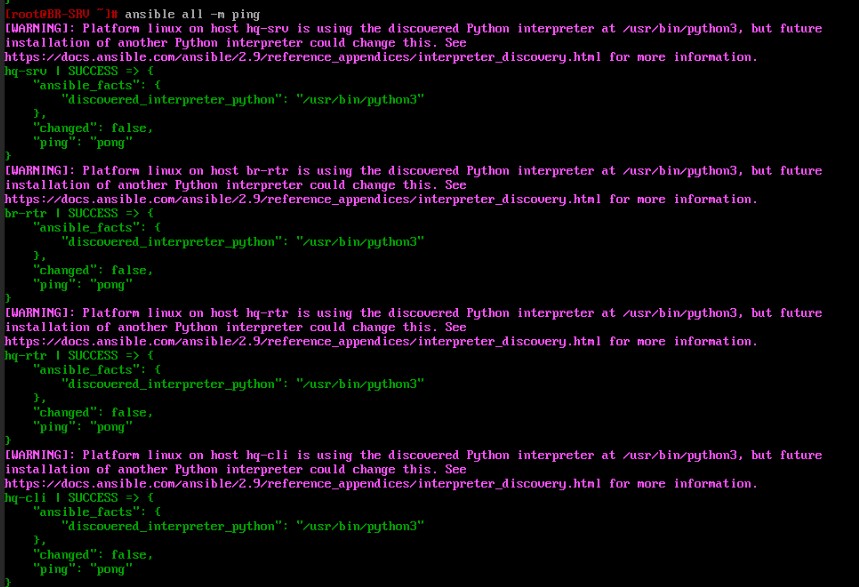
Копируем публичный ключ на клиентские машины, первая из них будет **BR-RTR**:

**ssh-copy-id -p 22 net\_admin@192.168.100.65**

Проделываем для остальных клиентов (**HQ-CLI**, **HQ-SRV**, **HQ-RTR**):

**ssh-copy-id -p 2024 sshuser@192.168.100.98** (Но у вас может быть другой IP, т.к. адрес он получает по DHCP) **ssh-copy-id -p 2024 sshuser@192.168.100.2** **ssh-copy-id -p 22 net\_admin@192.168.100.1**

После этого мы можем проверить связь. Машины должны без предупреждений и ошибок отвечать **pong** на команду **ping** в **ansible** посланную с **BR-SRV**: **ansible all -m ping**



Задание выполнено.

5. Развертывание приложений в Docker на сервере BR-SRV

Перед настройкой нам необходимо обновить список пакетов и установить **dockerengine**  и **docker-compose**:

**apt-get update**

**apt-get install docker-engine docker-compose** И запустим службу **docker**: **systemctl enable --now docker** **systemctl status docker**

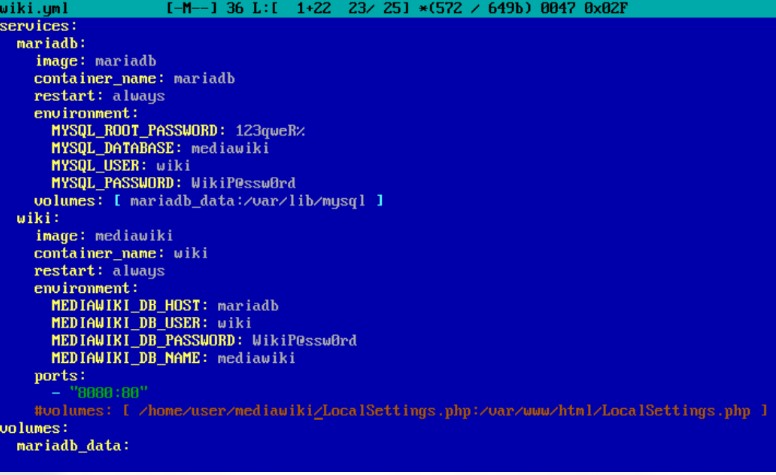
Загружаем образы следующей командой:

**docker pull mediawiki** **docker pull mariadb**

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

**mcedit /home/user/wiki.yml**

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



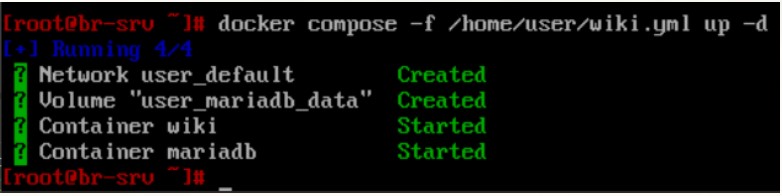
После всех настроек строку **volumes..** мы обратно раскомментируем, убрав символ **#**!

Приступаем к запуску контейнера **wiki.yml**, в зависимости от версии compose, существует ещё одна запись, она для второй его версии:

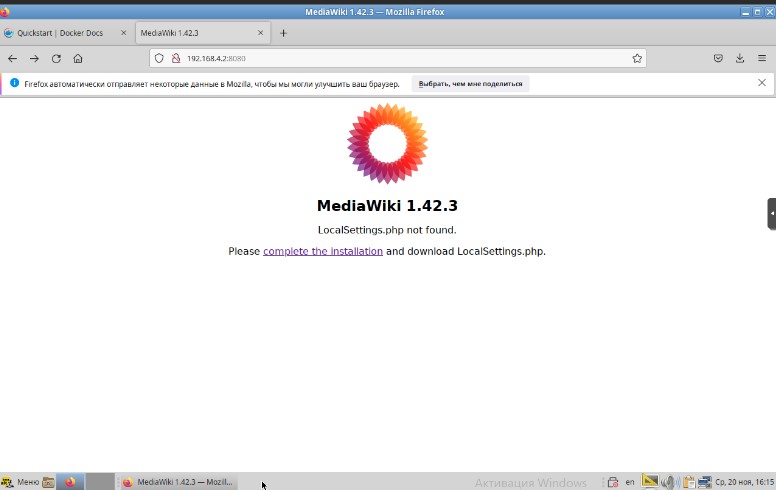
Обычная версия:

**docker-compose -f /home/user/wiki.yml up -d** Вторая версия:

**docker compose -f /home/user/wiki.yml up -d**

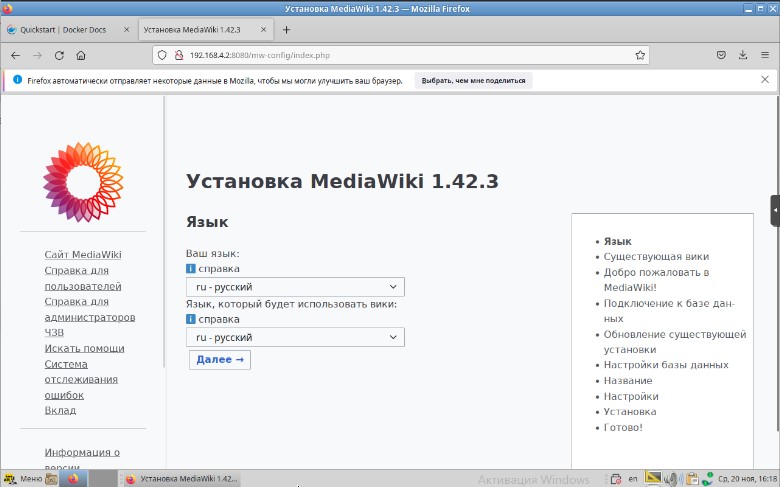


Заходим с клиента **HQ-CLI** на сайт после запуска контейнера: [http://192.168.100.66:8080](http://192.168.100.66:8080/)

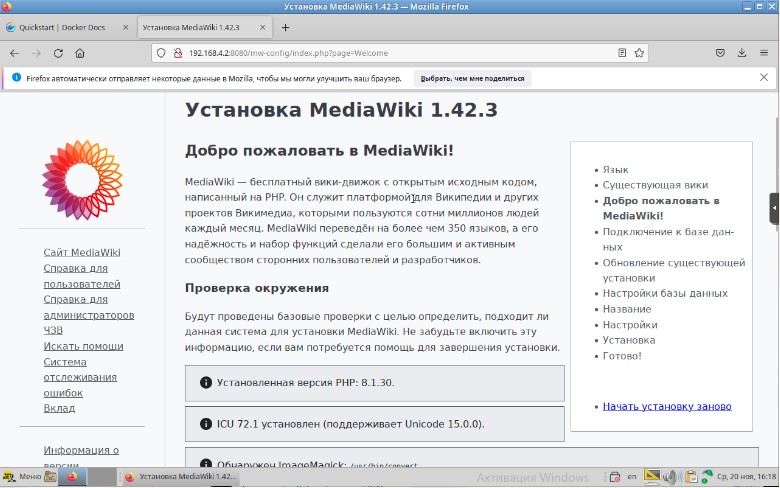


Видим, что файл **LocalSettings.php** не найден, и нажимаем на **complete the installation** или **set up the wiki**.

Выбираем удобный для вас язык:



Здесь просто идём далее:



Видим строки, которые нужно заполнить:

Хост базы данных:

**mariadb**

Имя базы данных (без дефисов):

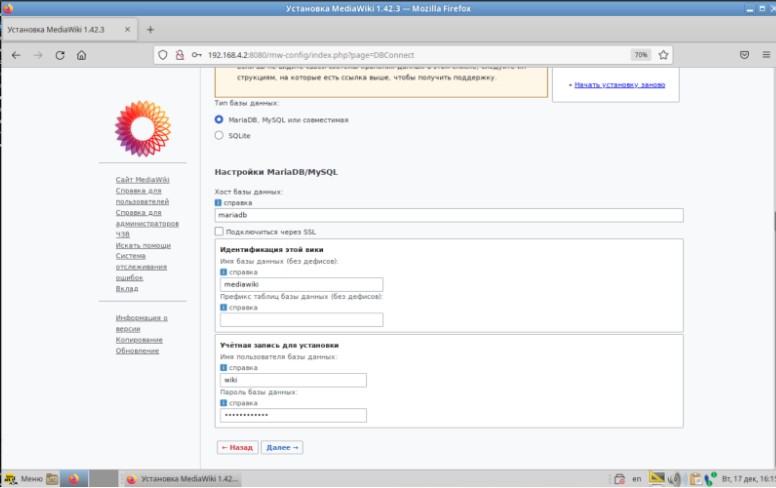
**mediawiki**

Имя пользователя базы данных:

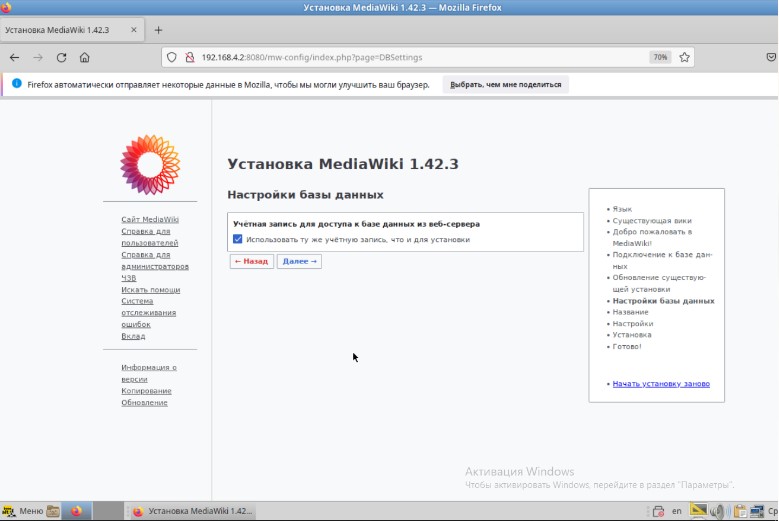
**wiki**

Пароль базы данных:

**WikiP@ssw0rd**



Прожимаем **Далее**, оставляя всё как есть:



Пишем в строках следующее и выбираем пункты, как на скрине:

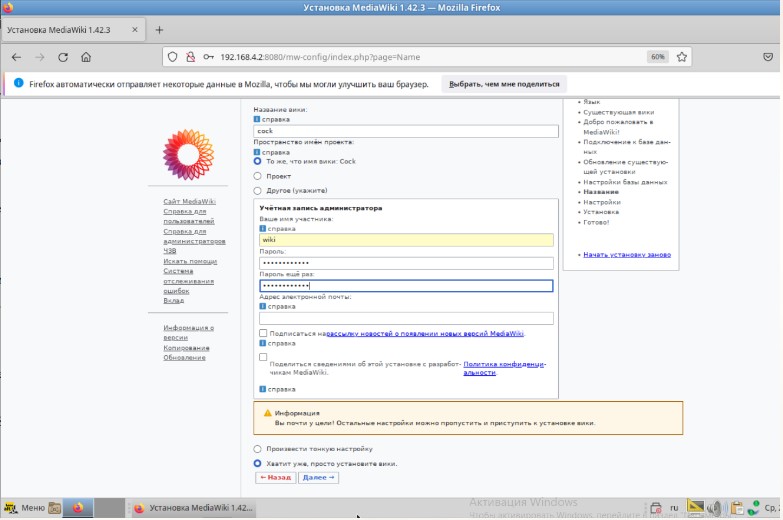
Название вики:

**cock** (можно своё название) Ваше имя участника:

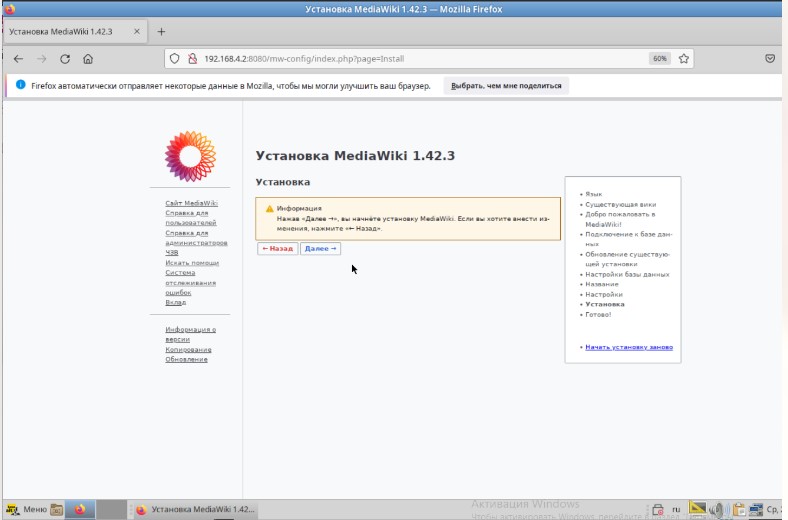
**wiki**

Пароль:

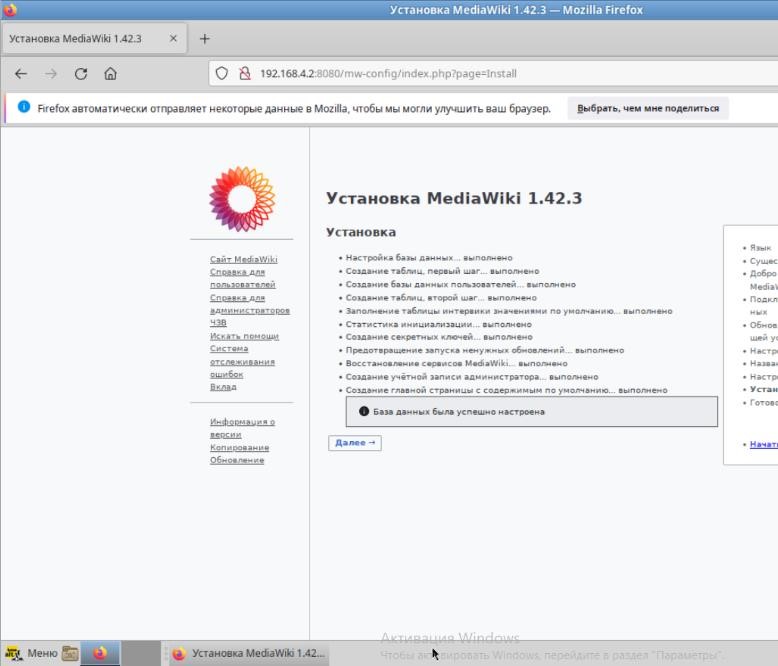
**WikiP@ssw0rd**



Нажимаем **Далее**:



И вот мы успешно создали базу данных:



Далее автоматически скачивается файл **LocalSettings.php**, который нужно переместить теперь на сервер с **mediawiki**, а именно на **BR-SRV** c **HQ-CLI**:

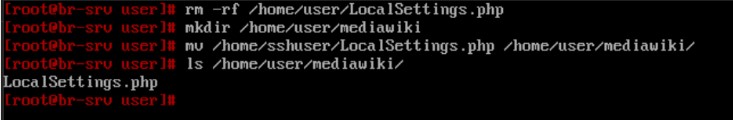
**scp -P 2024 /home/user/Загрузки/LocalSettings.php sshuser@192.168.100.66:/home/sshuser/**



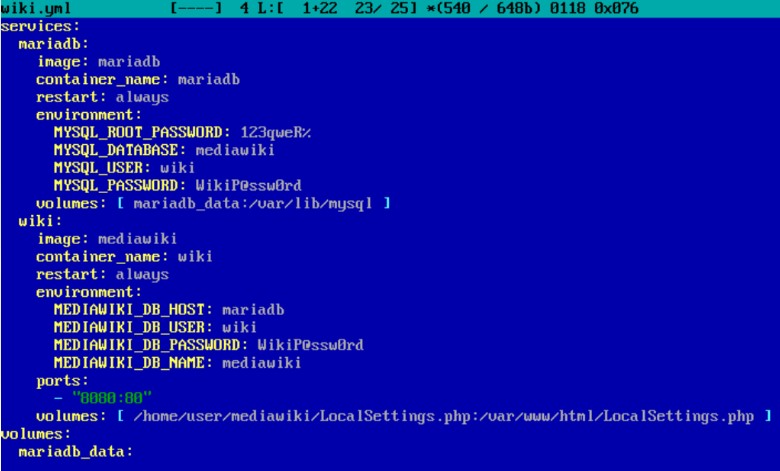
Теперь заходим на сервер **BR-SRV** и перемещаем скачанный файл в **/root**, но перед этим удаляем то, что создалось в **/root** (могло и не создаваться, так даже лучше):

**rm -rf /home/user/LocalSettings.php** **mkdir /home/user/mediawiki**

**mv /home/sshuser/LocalSettings.php /home/user/mediawiki/** **ls /home/user/mediawiki/**



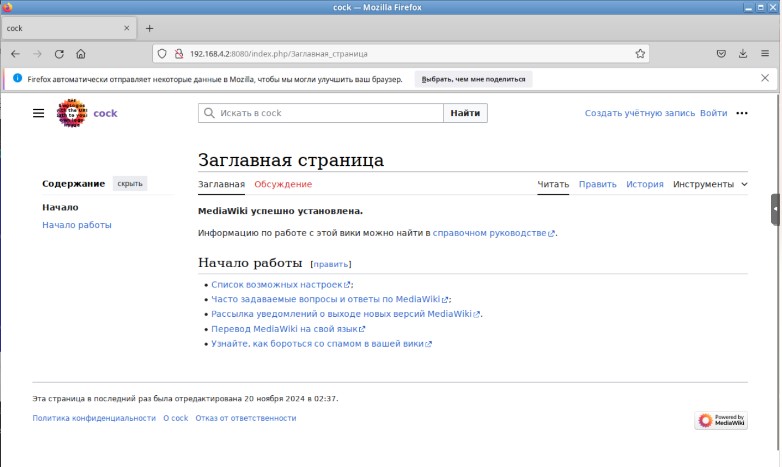
Раскомментируем, как и говорили ранее, строку **volumes…**:



Теперь перезапускаем контейнеры путём запуска контейнера ещё раз: **docker compose -f wiki.yml up -d**



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



6. На маршрутизаторах сконфигурируйте статическую трансляцию портов

Пробросим порт **80** в порт **8080** и порт **2024** в порт **2024** на **BR-SRV** на маршрутизаторе **BR-RTR**, для обеспечения работы сервиса **mediawiki** и **ssh**, правила прописываем через консоль:

**iptables -t nat -A PREROUTING -p tcp -d 192.168.100.65 --dport 80 -j DNAT --todestination 192.168.100.66:8080**

**iptables -t nat -A PREROUTING -p tcp -d 192.168.100.65 --dport 2024 -j DNAT --todestination 192.168.100.66:2024**

Сохраняем правила, не забывайте, что у вас уже есть правила, которые мы писали ещё в первом модуле, проверьте, чтобы в этом файле сохранялись и прошлые, и новые (которые мы сейчас ввели): **iptables-save > /etc/sysconfig/iptables systemctl restart iptables**

Пробросим порт **2024** в порт **2024** на **HQ-SRV** на маршрутизаторе **HQ-RTR**, для обеспечения работы сервиса ssh, правило прописываем через консоль:

**iptables -t nat -A PREROUTING -p tcp -d 192.168.100.1 --dport 2024 -j DNAT --todestination 192.168.100.2:2024 iptables-save > /etc/sysconfig/iptables systemctl restart iptables**

**7. Запустите сервис moodle на сервере HQ-SRV**

Устанавливаем для ряд пакетов, которые будут нам нужны для работы:

**apt-get update**

**apt-get install apache2 php8.2 apache2-mod\_php8.2 mariadb-server php8.2-opcache php8.2-curl php8.2-gd php8.2-intl php8.2-mysqli php8.2-xml php8.2-xmlrpc php8.2-ldap php8.2-zip php8.2-soap php8.2-mbstring php8.2-json php8.2-xmlreader php8.2-fileinfo php8.2-sodium**

Включаем службы **httpd2** и **mysqld** для дальнейшей работы с ними следующей командой:

**systemctl enable –now httpd2 mysqld**

Теперь настроим безопасный доступ к нашей будущей базе данных с помощью команды:

**mysql\_secure\_installation**

Прожимаем просто **enter**, т.к. сейчас **root** без пароля:

**Enter**

Прожимаем **y** для задания пароля:

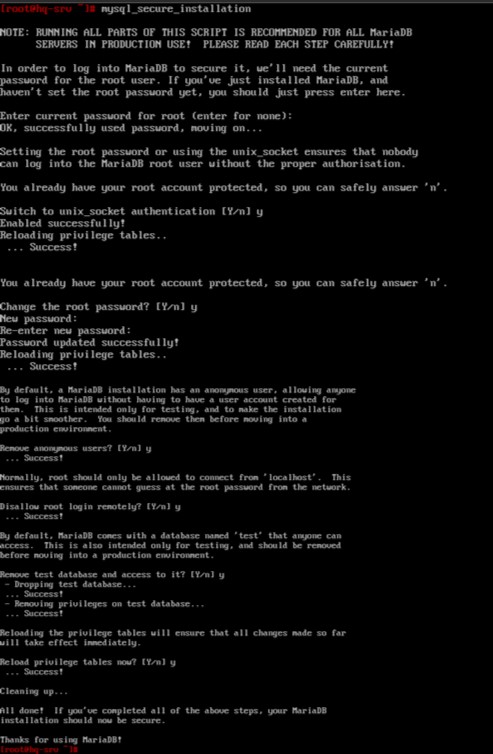
**Y**

Задаем пароль к нашему **root**, желательно стандартный:

**P@ssw0rd**

Далее нажимаем на всё **y**, как на скриншоте:

**Y**



Теперь заходим в СУБД для создания и настройки базы данных:

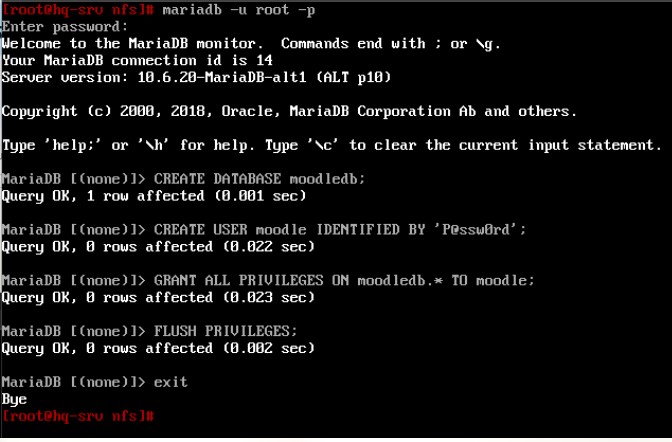
**mariadb -u root -p**

**CREATE DATABASE moodledb;**

**CREATE USER moodle IDENTIFIED BY ‘P@ssw0rd’;**

**GRANT ALL PRIVILEGES ON moodledb.\* TO moodle;** **FLUSH PRIVILEGES;**

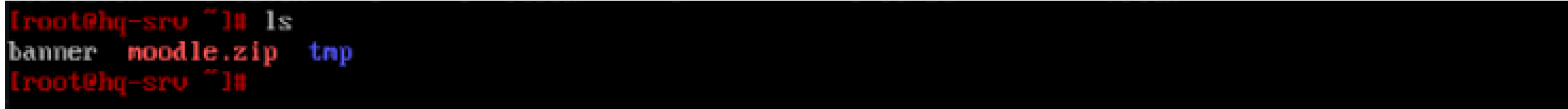
**exit**

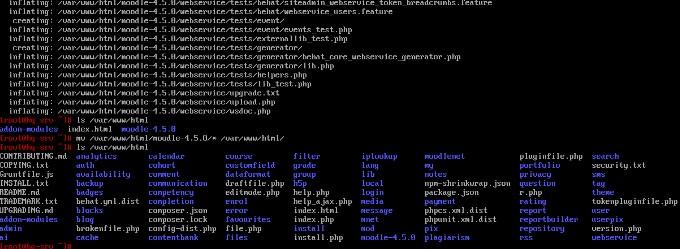


Теперь скачаем сам мудл стабильной версии:

**Архив moodle.zip хранится в директории /root, вам достаточно его разархивировать**

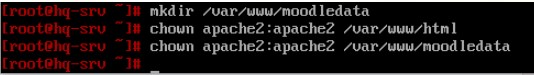
Разархивируем его в **/var/www/html/** для дальнейшей настройки:

**unzip /root/moodle.zip -d /var/www/html** **mv /var/www/html/moodle-4.5.0/\* /var/www/html/** **ls /var/www/html**



Создадим новый каталог **moodledata**, там будут храниться данные и изменим владельца на каталогах **html** и **moodledata**: **mkdir /var/www/moodledata**

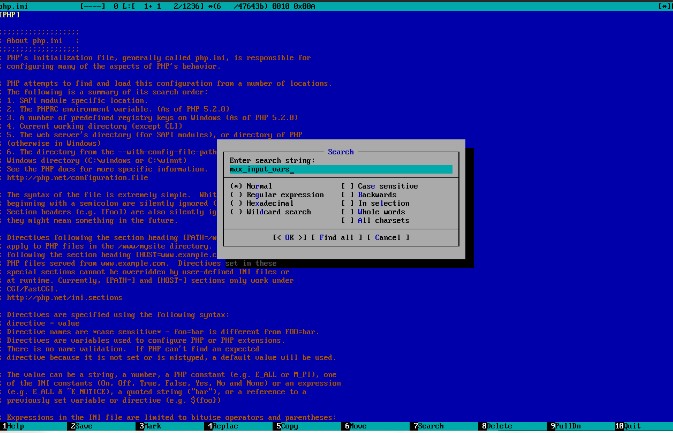
**chown apache2:apache2 /var/www/html** **chown apache2:apache2 /var/www/moodledata**



Поменяем значение параметра **max\_input\_vars** в файле **php.ini**:

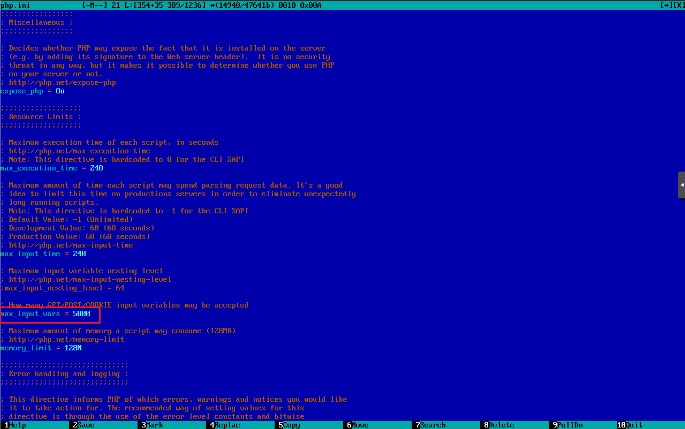
**mcedit /etc/php/8.2/apache2-mod\_php/php.ini**

Жмём **F7** для поиска нужной нам строки и пишем туда: **max\_input\_vars**



Раскомментируем и пишем новое значение:

**max\_input\_vars = 5000**



Удаляем стандартную страницу **apache**:

**cd /var/www/html**

**ls**

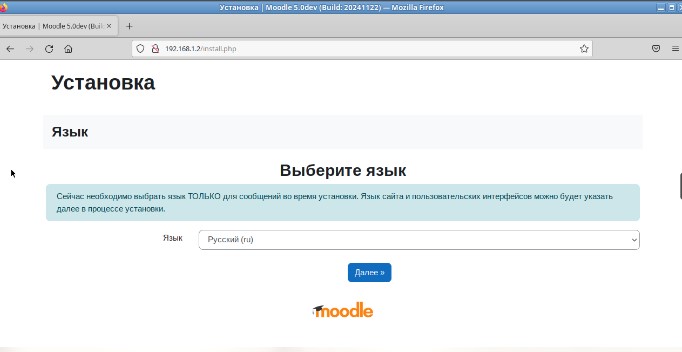
**rm index.html**

Перезапускаем службу **httpd2**:

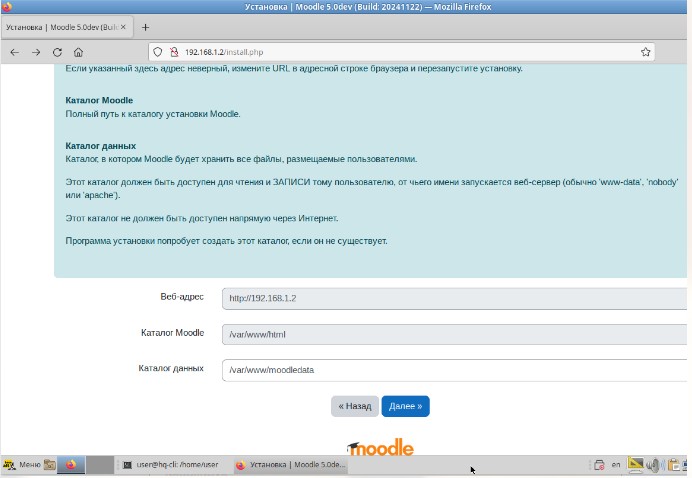
**systemctl restart httpd2**

Теперь подключаемся с клиента HQ-CLI и начинаем настройку:

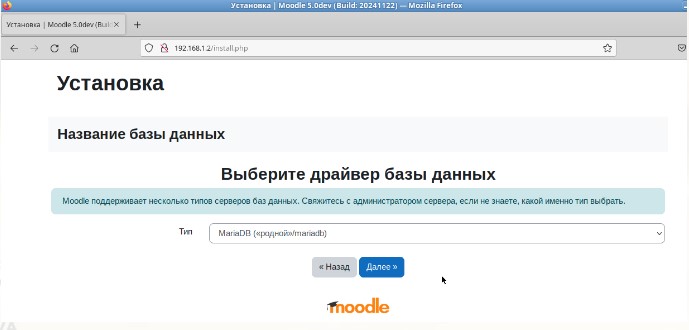
**http://192.168.100.2/install.php**



Жмём далее, т.к. каталог у нас уже создан:



Выбираем **MariaDB** в качестве драйвера базы данных:

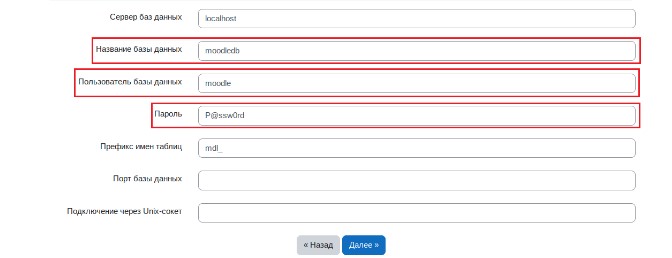


Введём нужные данные в следующие строки:

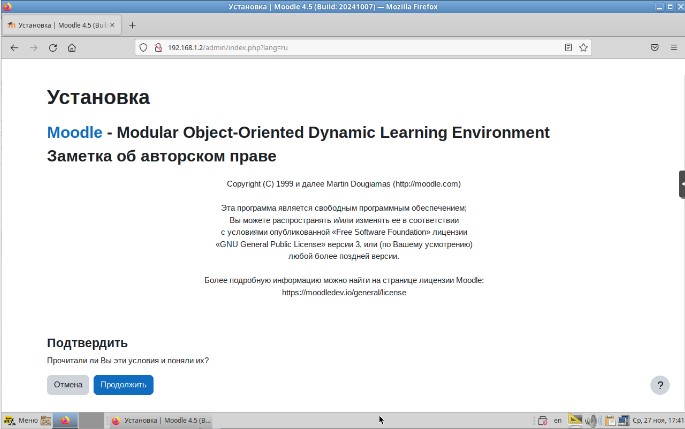
**Название базы данных: moodledb**

**Пользователь базы данных: moodle**

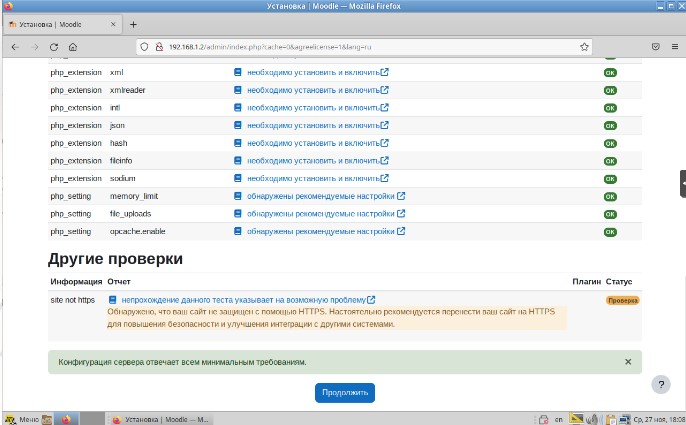
**Пароль: P@ssw0rd**



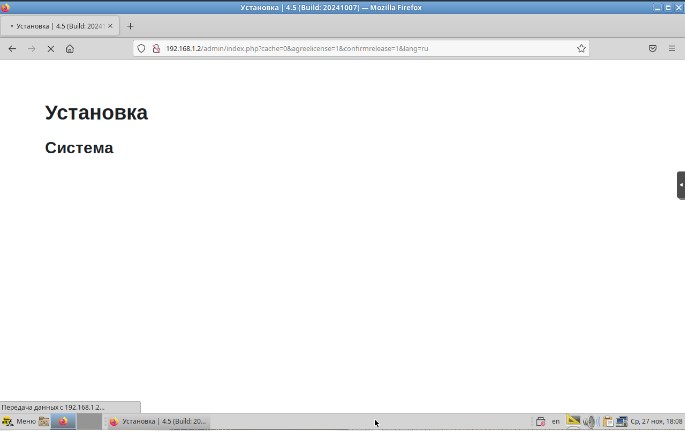
Нажимаем “**Продолжить**”:



Просматриваем всё ли в статус “**OK**” или “**Проверка**” и прожимаем “**Продолжить**”:



Дальше пойдёт процесс установки в виде такого окна, процесс этот может быть долгим, не пугайтесь:



После установки видим, что всё прошло успешно и жмём “Продолжить”:

Далее заполняем обязательные поля для создания основного администратора:

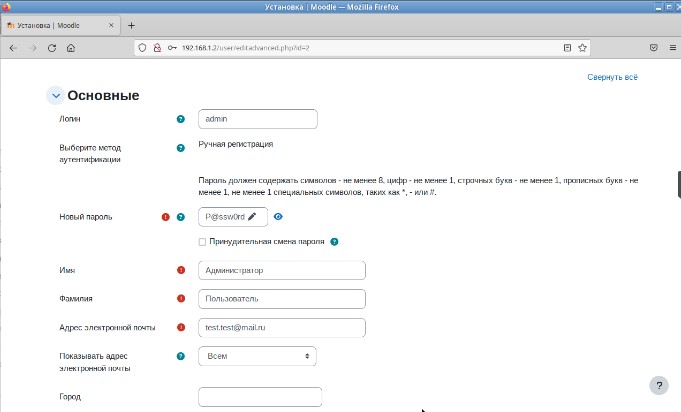
**Логин: admin**

**Новый пароль: P@ssw0rd**

**Имя: Администратор (**можно любое**)**

**Фамилия: Пользователь (**можно любое**)**

**Адрес электронной почты: test.test@mail.ru (**можно любое**)** И нажимаем “**Обновить профиль**”:



Теперь заполним ещё некоторые строки на следующем шаге:

**Полное название сайта: moodle** (можно любое**)**

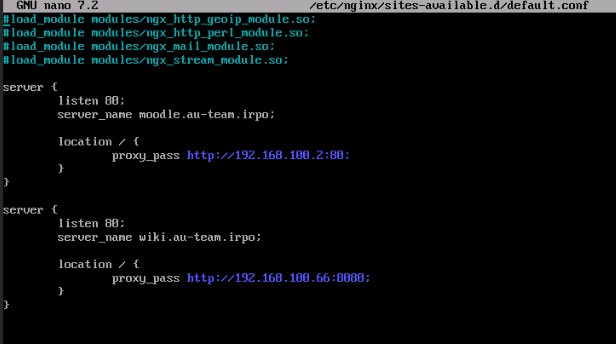
**Краткое название сайта: 47**

**Настройки местоположения: Европа/Москва**

**Контакты службы поддержки: test.test@mail.ru (**можно любое) И жмём “**Сохранить изменения**” в конце страницы:

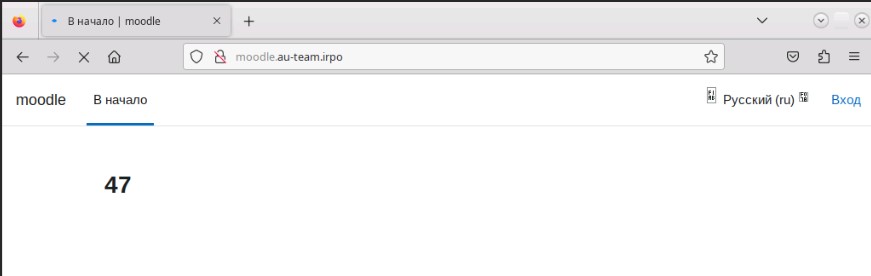
Задание успешно выполнено!

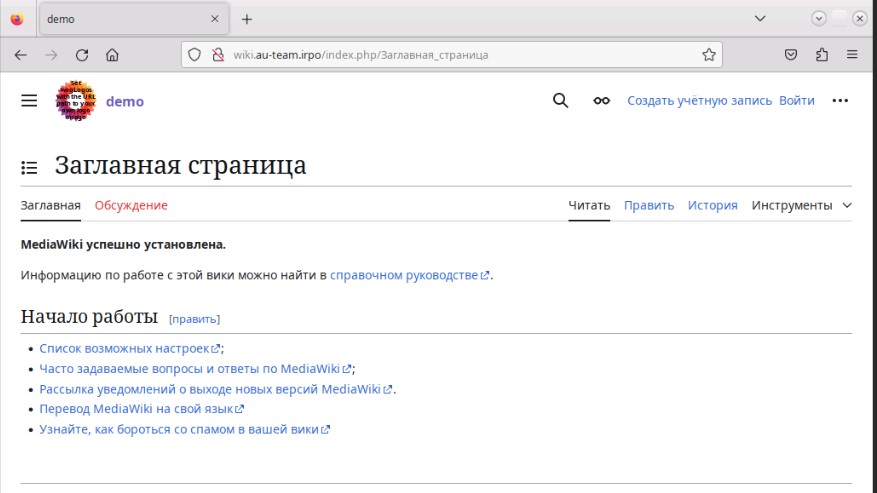
1. **Настройте веб-сервер nginx как обратный прокси сервер на HQ-RTR apt-get update && apt-get install nginx**

Переходим в конфигурационный файл nginx **mcedit /etc/nginx/sites-available.d/default.conf**  **ln -s /etc/nginx/sites-available.d/default.conf /etc/nginx/sites-enable.d/ Nginx -t**

**systemctl enable --now nginx**

После переходим на клиента проверяем работу прокси





1. Удобный способ установите приложение Яндекс Браузер для организаций на HQCLI

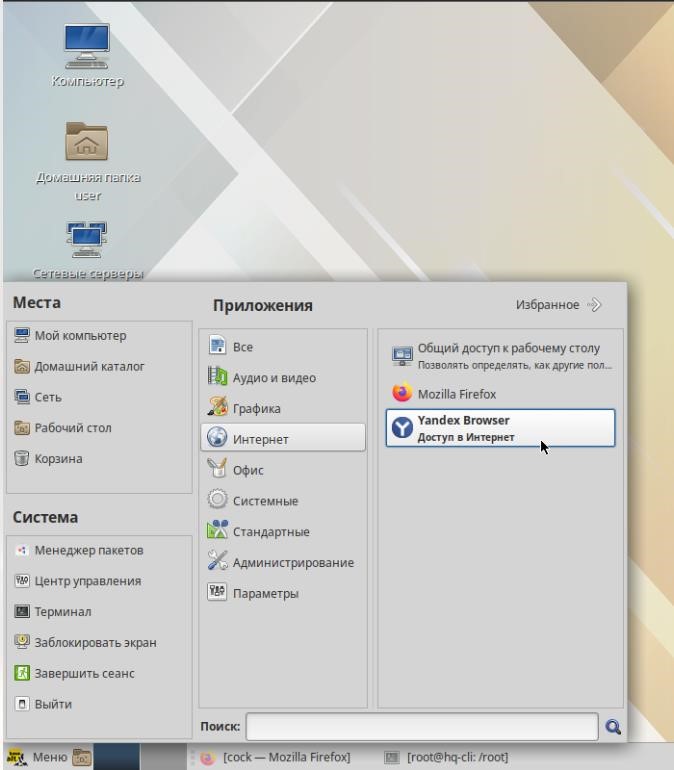
Установим Яндекс Браузер на **HQ-CLI** через терминал командами:

**apt-get update**

**apt-get install yandex-browser-stable**

Яндекс браузер должен быть по пути:

**Пуск → Интернет → Yandex Browser**



При нажатии “☰” мы увидим внизу надпись “**Яндекс.Браузер для организаций**”, это значит, что мы установили правильную версию браузера.

