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

### Настройка Samba AD:

Удаляем smb.conf и другие конфигурационные файлы (рис. 1.1):

***rf -f /etc/samba/smb.conf***

рис. 1.1

Запускаем автоматическую настройку (рис. 1.2)

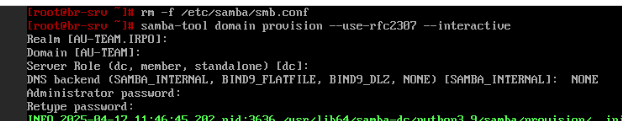
samba-tool domain provision --use-rfc2307 --interactive

Задаем все параметры по умолчанию кроме:

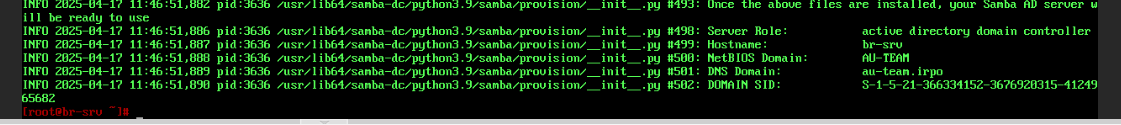
DNS backend: NONE

Password: P@ssw0rd

Retype password: P@ssw0rd

рис.1.3

после успешной настройки покажет параметры домена (рис. 1.3).

рис. 1.3

Настраиваем Kerberos (рис.1.4)

***cp /var/lib/samba/private/krb5.conf /etc/krb5.conf***

рис.1.4

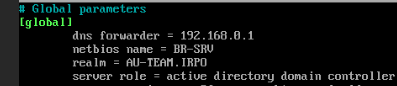
Добавляем адрес dns сервера в smb.conf (рис.1.5)

nano /etc/samba/smb.conf

Рис.1.5

добавляем строчку dns forwarder (рис.1.6)

dns forwarder = 192.168.0.1

Рис.1.6

Запускаем samba с добавлением в автозагрузку (рис. 1.7)

***systemctl enable --now samba***

рис. 1.7

Настраиваем группу и пользователя (рис .1.8)

создаем группу для пользователей домена

***samba-tool group add hq***

создаем пользователей

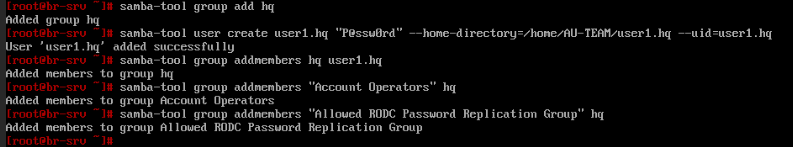
***samba-tool user add user1.hq "P@ssw0rd" --home-directory=/home/AU-TEAM/user1.hq --uid=user1.hq***

Добавляем пользователя в группу

***samba-tool group addmembers hq user1.hq***

***samba-tool group addmembers "Account Operators" hq***

***samba-tool group addmembers “Allowed RODC Password Replication Group” hq***

рис.1.8

перезагружаем samba

***systemctl restart samba***

**Внимание рекомендую перезагрузить сервер!!!**

***reboot***

Вводим в домен hq-cli

На клиентском компьютере заходим в терминал подключаемся к пользователю root (рис.1.9)



Вводим команду подключения к домену (рис.1.10):

realm join -U user1.hq au-team.irpo

вводим пароль: [P@ssw0rd](mailto:P@ssw0rd) (от пользователя)

Рис.1.10

Добавляем доменную группу (рис.1.11)

realm permit -g hq

Рис.1.11

Добавляем права доменных пользователей

***echo***  “***%hq ALL=(ALL) NOPASSWD:/bin/cat,/bin/grep,/usr/bin/id“ > /etc/sudoers.d/hq***

******

Импорт пользователей на сервере BR-SRV:

Создайте скрипт для импорта на сервере:

nano /opt/import\_users.sh

Пишем

**#!/bin/bash**

**CSV\_FILE="/opt/users.csv"**

**while IFS=, read -r username password group; do**

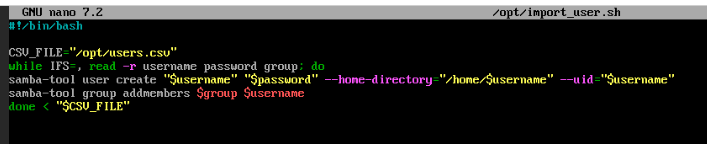
**# Создание пользователя**

**samba-tool user create "$username" "$password" --home-directory="/home/$username" --uid="$username"**

**# Добавление в группу**

**samba-tool group addmembers "$group" "$username"**

**done < "$CSV\_FILE"**



Сделайте скрипт исполняемым:

**chmod +x /opt/import\_users.sh**

Запустите его:

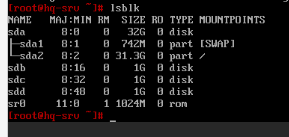
**/opt/import\_users.sh**



**2. **Сконфигурируйте файловое хранилище:****

**Выводим список подключенных дисков и их имена.**

*****# lsblk*****



## Создание таблицы разделов

**Для разбиения диска, запускаем fdisk с именем устройства:**

*****# fdisk /dev/sdb*****

Послевовательн выполняем команды

Вводим g чтобы создать новую пустую таблицу разделов:

***Command (m for help): g***

**Водим команду** n **чтобы создать новый раздел:**

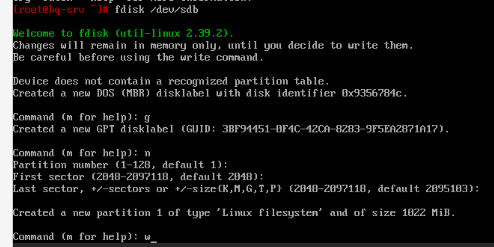
*****Command (m for help): n*****

**Будет предложено ввести номер раздела. Жмем** «Enter»**, чтобы использовать значение по умолчанию ( 1 )**

**Далее необходимо указать первый сектор, используем значение по умолчанию. Жмем** «Enter»

**При следующем запросе необходимо ввести последний сектор, Жмем** «Enter»**, так как используем весь диск**

**Сохраняем изменения, запустив команду** w **(Жмем w)**



**Команда запишет таблицу на диск и выйдет из меню** fdisk.

Ядро прочитает таблицу разделов устройства без перезагрузки системы.

**Аналогично создаем разделы на других дисках** /dev/sdc, /dev/sdd



## Создание программного RAID5

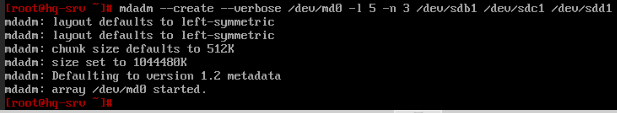
****Для сборки массива применяем следующую команду:****

*****# mdadm --create --verbose /dev/md0 -l 5 )n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1*****

******

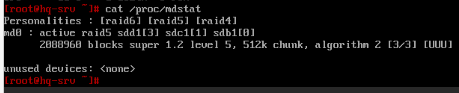
**Система задаст контрольный вопрос, хотим ли мы продолжить и создать RAID — нужно ответить** y**:**

Мы должны увидеть что-то на подобие:



****Посмотреть состояние всех RAID можно командой:****

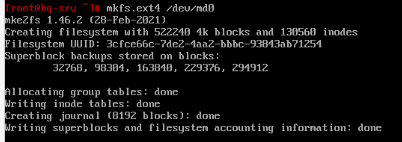
*****# cat /proc/mdstat*****



## Монтирование файловой системы массива

****Отформатируем разделы** в** ext4:

***# mkfs.ext4 /dev/md0***



В корне системы создадим директорию /raid5

***# mkdir /raid5***

Примотируем к RAID директорию /raid5 командой:

***# mount /dev/md0 /raid5***



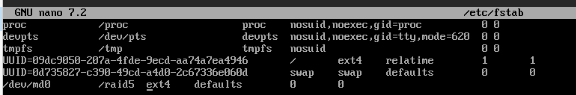
**Чтобы разделы монтировались при загрузке системы**

Открываем fstab:

***# nano /etc/fstab***

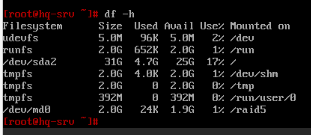
И добавляем строки

***/dev/md0 /raid5 ext4 defaults 0 0***



Проверяем примонтированные разделы:

# df -h



## Сервер nfs

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

***# mkdir /raid5/nfs***

**Измените права доступа**

****sudo chmod 777 /raid5/nfs****

****

****Запускаем nfs сервер:****

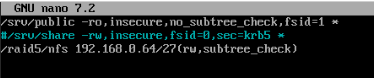
****systemctl enable —now nfs****

**Открываем на редактирование файл** /etc/exports **который содержит информацию о каталогах, экспортируемых с сервера:**

****# nano /etc/exports****

**Создаем шару из каталога** /raid5/nfs**, которая будет доступна для всех узлов сети** 192.168.0.64/27**.**  
Добавим в него строку:

***/raid5/nfs 192.168.0.64/28(rw,subtree\_check)***



**Выполняем экспорт данных (перечитаем конфигурационный файл** /etc/exports**, чтобы сервер начал отдавать настроенные шары):**

**# exportfs -a**

## Автомонтирование

**На** HQ-CLI **устанавливаем компонент для клиентской части** NFS**:**

**Создаем директорию для автомонтирования** /mnt/nfs

***# mkdir /mnt/nfs***

**Пробуем примонтировать шару:**

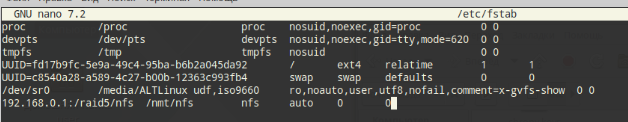
*****# mount -t nfs 192.168.0.1:/raid5/nfs /mnt/nfs*****

**Для автоматического монтирования на** HQ-CLI **после перезагрузки, используем** fstab**.** Открываем файл fstab:

***# nano /etc/fstab***

И добавляем строку:

***192.168.0.2:/raid5/nfs /mnt/nfs nfs auto 0 0***



**3. Настройте синхронизацию времени между сетевыми устройствами по протоколу NTP**

Заходим в конфигурационный файл "chrony.conf" (рис. 3.1)

***nano /etc/chrony.conf*** – ввод (Enter)

Рис. 3.1

Приводим конфигурационный файл к следующему виду (рис. 3.2) IP-адреса согласно вашим параметрам, **остальные пункты комментировать или удалить!:**

***server 127.0.0.1 iburst prefer***

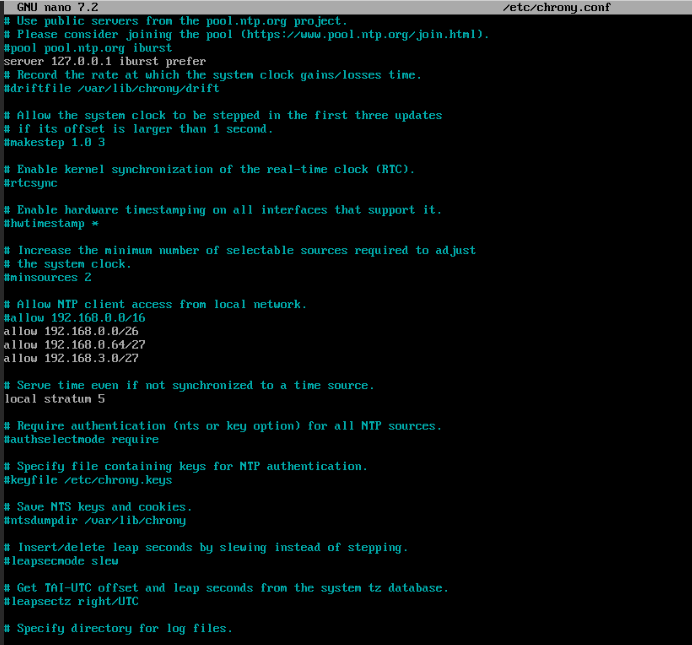
***local stratum 5***

***allow 192.168.0.0/26***

***allow 192.168.0.64/27***

***allow 192.168.0.96/29***

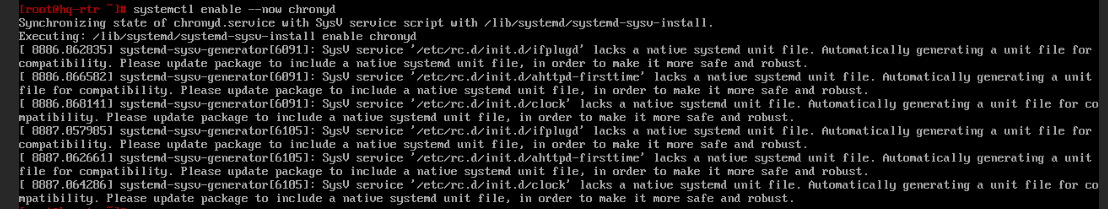
***allow 192.168.3.0/27***

Рис. 3.2

Сохраняем файл и выходим.

Запускаем службу синхронизации (рис. 3.3)

***systemctl enable –now chronyd*** – ввод (Enter)

Рис. 3.3

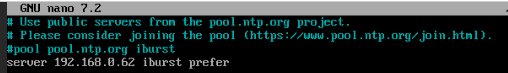
Настройка NTP клиентов на примере hq-srv br-rtr, br-srv, hq-cli.

Заходим в конфигурационный файл "chrony.conf"

***nano /etc/chrony.conf*** – ввод (Enter)

Приводим конфигурационный файл к следующему виду (рис. 3.6) IP-адреса согласно вашим параметрам, **остальные пункты комментировать или удалить!:**

***server 192.168.1.62 iburst prefer***

Рис.3.6

Сохраняем файл и выходим.

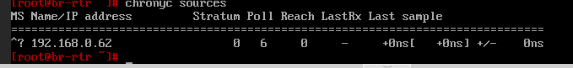
Запускаем службу синхронизации (рис. 3.7)

***systemctl enable –now chronyd*** – ввод (Enter)

Рис.3.7

Проверяем (рис. 3.8)

***chronyc sources*** – ввод (Enter)

Рис. 3.8

Остальные устройства настраиваем по аналогии с hq-srv.

**4. **Сконфигурируйте ansible на сервере BR-SRV****

## Создание пары SSH-ключей.

Чтобы заходить на удаленные машины, пользователь на BR-SRV (sshuser и root) должен создать пару из закрытого/открытого RSA ключа. Это делается следующей командой:

**# ssh-keygen -t rsa**

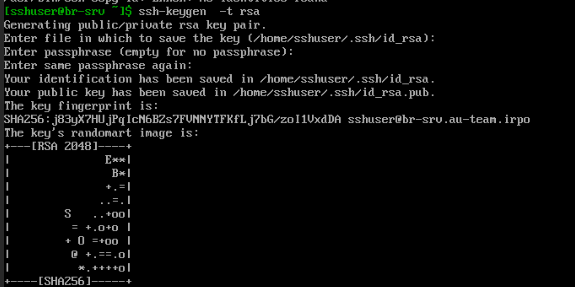
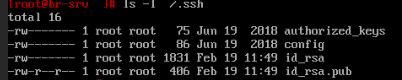
Далее Enter

В результате в каталоге /home/sshuser/.ssh или /root/.ssh будут созданы файлы ключей:

ls -l ~/.ssh

id\_rsa # закрытый ключ

id\_rsa.pub # открытый ключ



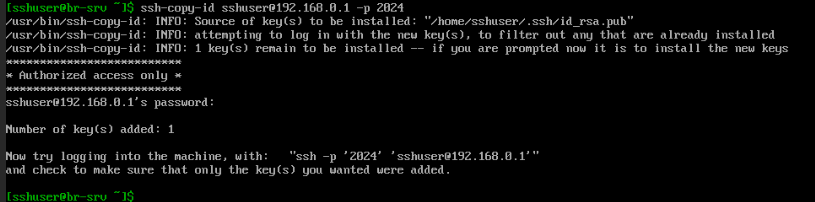
**Заходим под пользователем** sshuser и root (выполняется аналогично)

# su — sshuser

**Копируем открытый** SSH-ключ **на удаленные устройства под пользователем** sshuser**:**

**Копируем ключ для пользователя** sshuser **на HQ-SRV**  
На HQ-SRV ssh порт изменен, указываем его

ssh-copy-id -p 2024 [sshuser@192.168.0.1](mailto:sshuser@192.168.0.1)

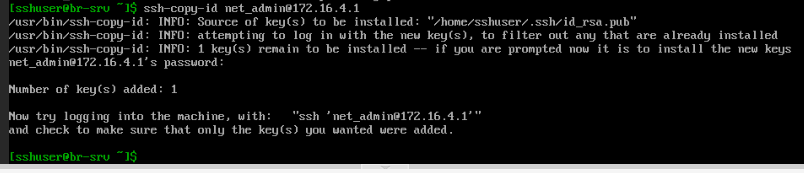


**Копируем ключ для пользователя** user **на HQ-CLI**

ssh-copy-id user@192.168.0.65

**Копируем ключ для пользователя** net\_admin **на HQ-RTR**

ssh-copy-id [net\_admin@172.16.4.1](mailto:net_admin@172.16.4.1)

**Копируем ключ для пользователя** net\_admin **на BR-RTR**

ssh-copy-id [net\_admin@172.16.5.1](mailto:net_admin@172.16.5.1)

Подготовка файла инвентаря (hosts)

**Создаем файл инвентаря** /etc/ansible/demo

nano /etc/ansible/inv



Сначала указывается название группы в квадратных скобках, затем перечисляются хосты.  
Имена хостов прописываются в виде ip.  
После названия хоста можно указать параметры подключения к хосту в одну строку.

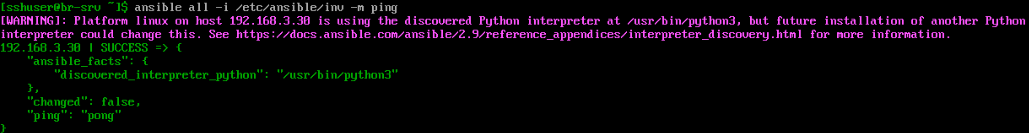


Запуск команд с пользовательским инвентарем

ansible all -i /etc/ansible/inv -m ping



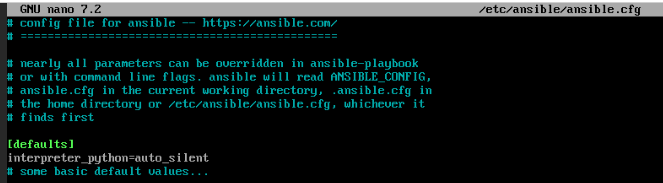
**Может появиться предупреждение про обнаружение интерпретатора** Python**, на целевом хосте**



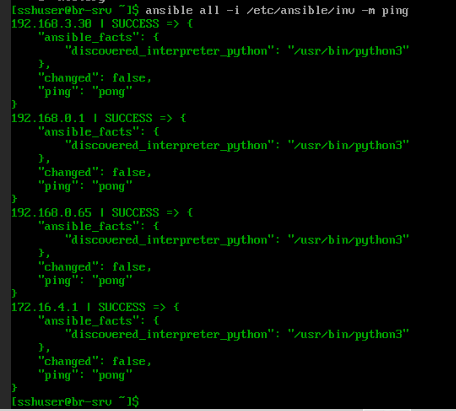
Для управления поведением обнаружения в глобальном масштабе необходимо в файле конфигурации ansible /etc/ansible/ansible.cfg в разделе [defaults] прописать ключ interpreter\_python с параметром auto\_silent. В большинстве дистрибутивов прописываем вручную.

# nano /etc/ansible/ansible.cfg

Прописываем вручную



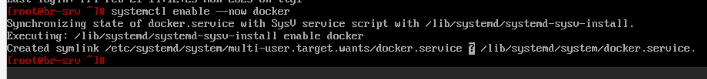
**Проверяем**



**5. Развертывание приложений в Docker на сервере BR-SRV.**

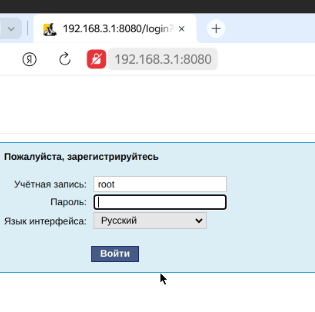
## Запустить сервис контейнеризации docker и добавить его в автозагрузку:

**systemctl enable --now docker**



перед настройкой wiki необходимо сменить порт веб интерфейса ЦУС сервера!

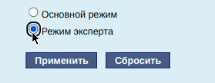
На CLI открываем браузер в водим адрес [https://192.168.3.1](https://192.168.3.1/) вводим пароль от root и заходим в ЦУС



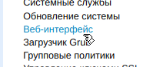
нажимаем настройка



выбираем режим эксперта, применить



в меню выбираем веб-интерфейс

Меняем порт на 8081

нажимаем применить и перезапустить http-сервер

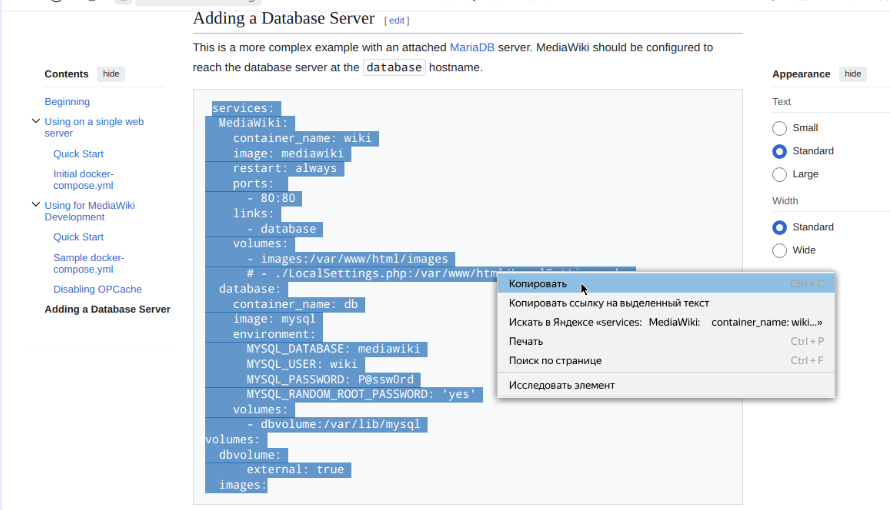
Установка MediaWiki

Для упрощения создания wiki.yml в поисковой системе (Яндекс) на HQ-CLI пишем mediawiki docker-compose

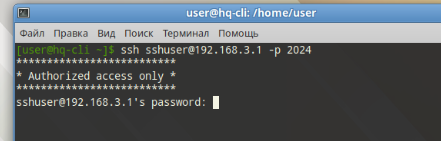
****

Переходим по ссылке. На странице находим раздел **Adding a Database Server нажимаем**

Выделяем код и копируем его

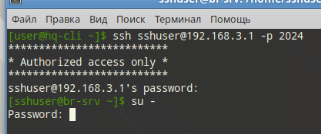


подключаемся по SSH к BR-SRV с HQ-CLI.

****

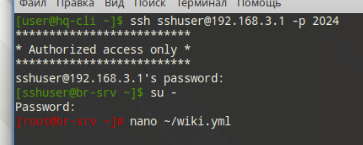
заходим под root

su-



создаем файл

nano ~/wiki.yml



вносим изменения в файл:

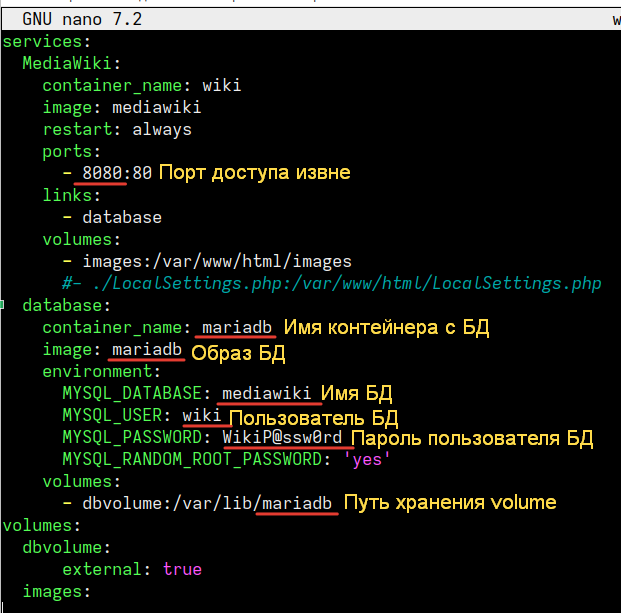
порт на 8080

имя контейнера и образ на mariadb

имя БД на mediawiki,

пароль пользователя БД на WikiP@ssw0rd,

изменяем параметр volumes (правим путь)

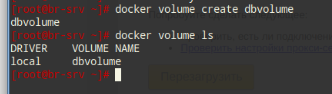


Чтобы отдельный **volume** для хранения базы данных имел **правильное имя** - создаём его средствами docker:

***docker volume create dbvolume***

**Посмотреть все имеющиеся volume можно командой**

***docker volume ls*** (необезательно)

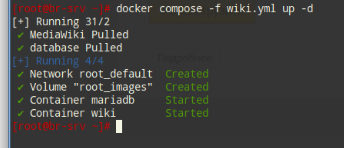


Выполняем запуск стека контейнеров с приложением MediaWiki и базой данных описанных в файле wiki.yml:

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

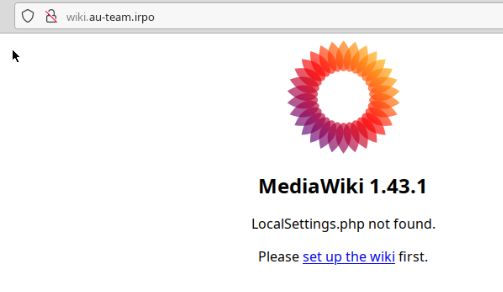
******

ждем завершения установки и запуска

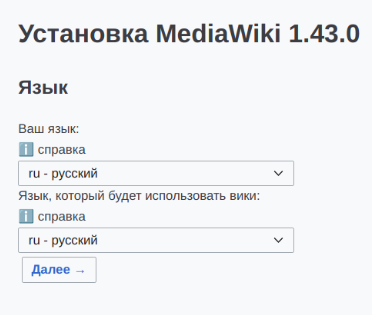


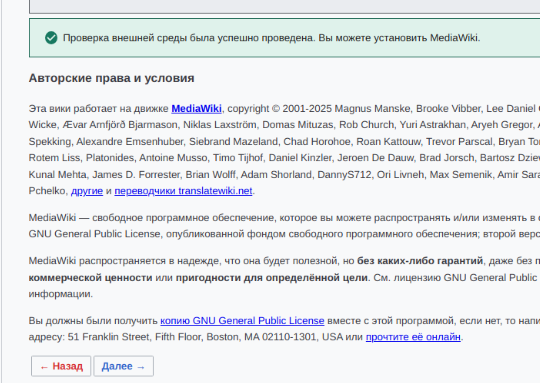
Открываем браузер по адресу нашего сервера [http://192.168.3.1:8080](http://192.168.3.1:8080/) или [http://wiki.au-team.irpo](http://wiki.au-team.irpo/) если настроен nginx.

нажимаем «set up the wiki» приступаем к установки



выбираем русский язык, далее

в следующем пункте проверка прошла нажимаем далее



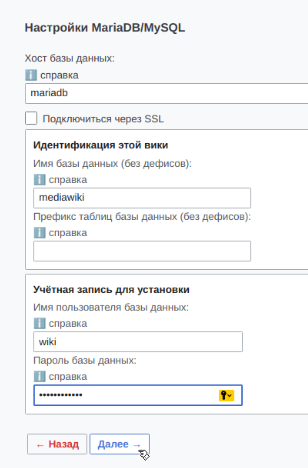
Настраиваем параметры базы данных:

Хост: mariadb

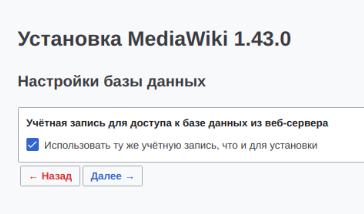
Идентификатор:mediawiki

Учетная запись: wiki WikiP@ssw0rd

далее



подтверждаем, далее

Задаем параметры

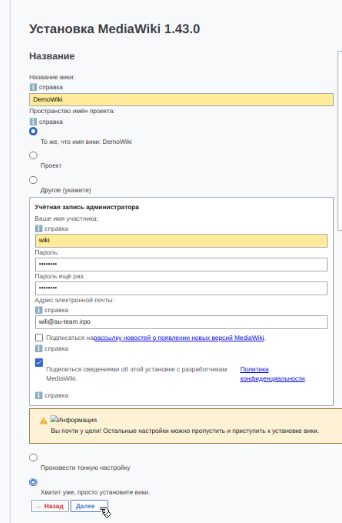
Название — DemoWiki

Учетная запись — wiki WikiP@ssw0rd

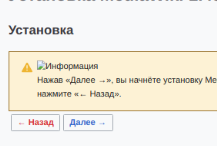
адресс — wiki@au-team.irpo

выбираем пункт «хватит уже, просто установить»

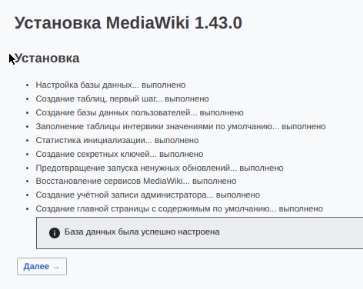
Далее



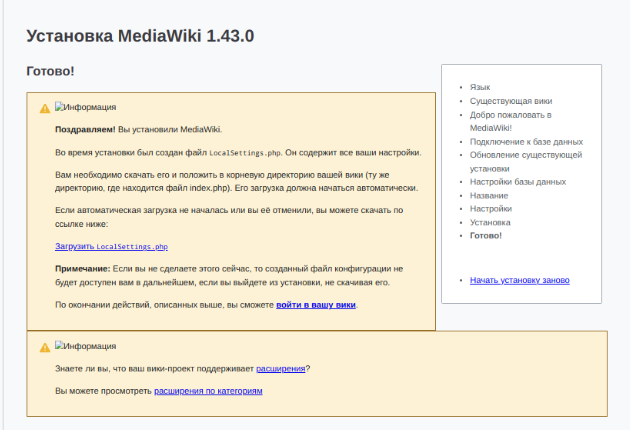
нажимаем далее



далее

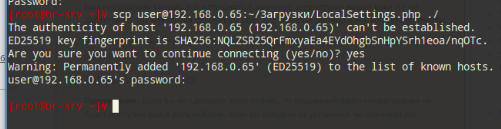


автоматически скатается LocalSettings.php



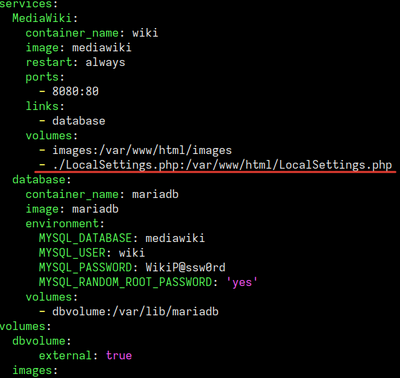
Необходимо передать файл LocalSettings.php с CLI на сервер:

scp user@192.168.0.65:~/Загрузки/LocalSettings.php ./



Раскомментируем строку в файле wiki.yml:

# nano wiki.yml



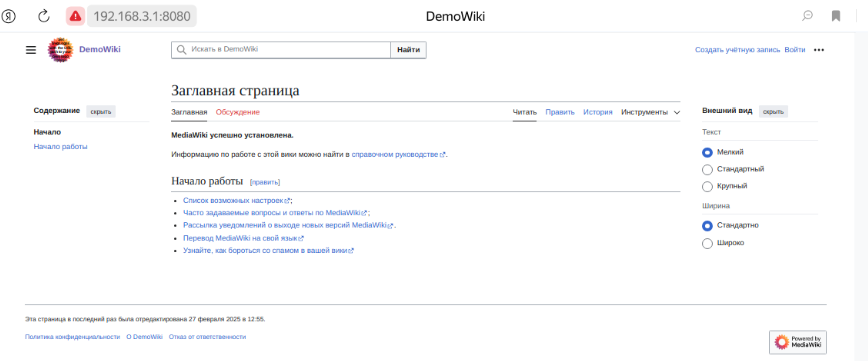
Перезапускаем сервисы средствами docker-compose:

# docker compose -f wiki.yml stop

# docker compose -f wiki.yml up -d



[**Проверяем доступ к**](http://192.168.200.2:8080/) Wikihttp://192.168.3.2:8080



## **6. На маршрутизаторах сконфигурируйте статическую трансляцию портов**

На BR-RTR изменяем правило переадресации в таблице table inet nat, (рис. 6.1) добавляя chain prerouting :

chain prerouting {

***type nat hook prerouting priority filter;***

ip daddr 172.16.5.1 tcp dport 80 dnat ip to 192.168.3.1:8080

ip daddr 172.16.5.1 tcp dport 22 dnat ip to 192.168.3.1:2222

}

рис. 6.1

перезапускаем nftables!

На HQ-RTR изменяем правило переадресации в таблице table inet nat, (рис. 6.2) добавляя chain prerouting :

chain prerouting {

***type nat hook prerouting priority filter;***

ip daddr 172.16.4.1 tcp dport 22 dnat ip to 192.168.0.1:2222

}

рис. 6.2

перезапускаем nftables!

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

**HQ-SRV:**

Запускаем **веб-сервер** и добавляем в автозагрузку (рис. 7.1):

*systemctl enable --now httpd2*

*рис. 7.1*

Включаем и добавляем в автозагрузку **MySQL** (рис. 7.2):

*systemctl enable --now mariadb*

рис. 7.2

Подключаем модуль

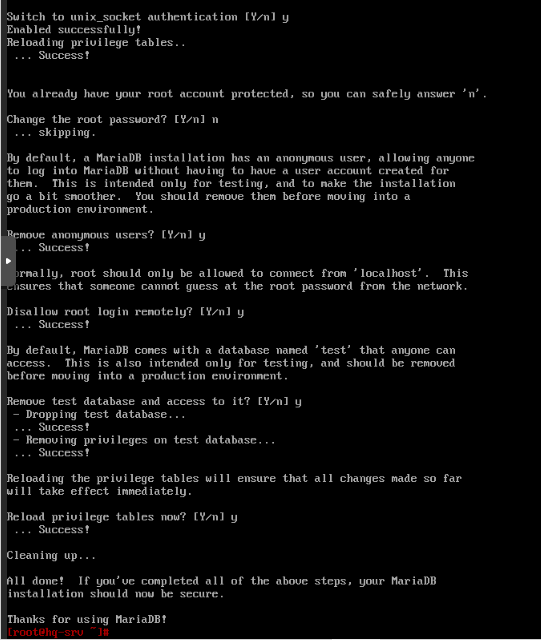
***mysql\_secure\_installation*** *(tab на этом пункте работает)*

первый пункт нажимаем Enter

второй пункт нажимаем ***y***

третий пункт ***n***

все остальные пункты **y**

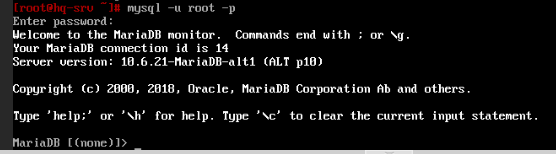


Подключаемся к **MySQL**, создаём **базу данных** и **пользователя**:

имя базы даных — "**moodledb**", имя пользователя - "**moodle**", пароль "[**P@ssw0rd**](mailto:P@ssw0rd)";

mysql -u root -p

вводим пароль от root

Создаем базу (можно писать все в нижнем регистре (маленькими буквами)):

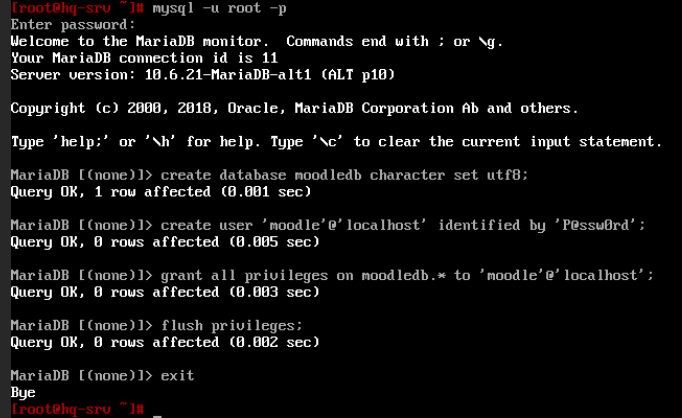
CREATE DATABASE moodledb CHARACTER SET utf8;

CREATE USER 'moodle'@'localhost' IDENTIFIED BY 'P@ssw0rd';

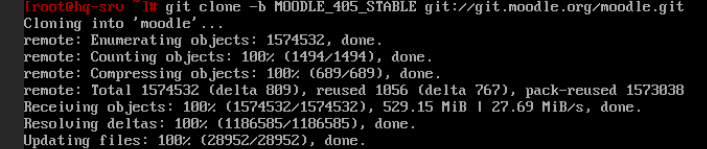
GRANT ALL PRIVILEGES ON moodledb.\* TO 'moodle'@'localhost';

FLUSH PRIVILEGES;

EXIT;

Загружаем код проекта **Moodle**:

*git clone -b MOODLE\_405\_STABLE git://git.moodle.org/moodle.git*

**

Копируем локальный репозиторий в **/var/www/html/:**

***cp -R moodle /var/www/html/***



Создаём необходимую структуру каталагов для корректной установки и работы Moodle:

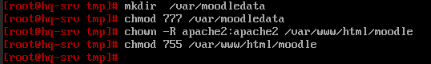
***mkdir /var/moodledata***

*назначаем права*

***chmod 777 /var/moodledata***

***chown -R apache2:apache2 /var/www/html/moodle***

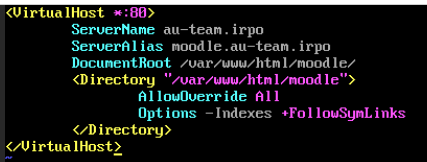
***chmod -R 755 /var/www/html/moodle***

**

создаем и описываем конфигурационный файл для веб-сервера Apache:

*nano /etc/httpd2/conf/sites-available/moodle.conf*

*прописываем параметры*

Создаём символьную ссылку из **sites-available** на **sites-enabled**:

*ln -s /etc/httpd2/conf/sites-available/moodle.conf /etc/httpd2/conf/sites-enabled/*

Проверяем синтаксис файла виртуального хоста:

*apachectl configtest*

**

Правим количество входных переменных, которые могут быть приняты в одном запросе, для работы Moodle - необходимо **5000**, а значение в php.ini по умолчанию **1000**:

*заходим в конфиг*

*nano /etc/php/8.1/apache2-mod\_php/php.ini*

*нажимаем* **ctrl \_**

набираем № строки **389**, нажимаем энтер

находим строку **max\_input\_vars** изменяем значение

убираем точку с запятой в начале строки ( **;** ) и ставим значение **5000**

сохраняем и выходим

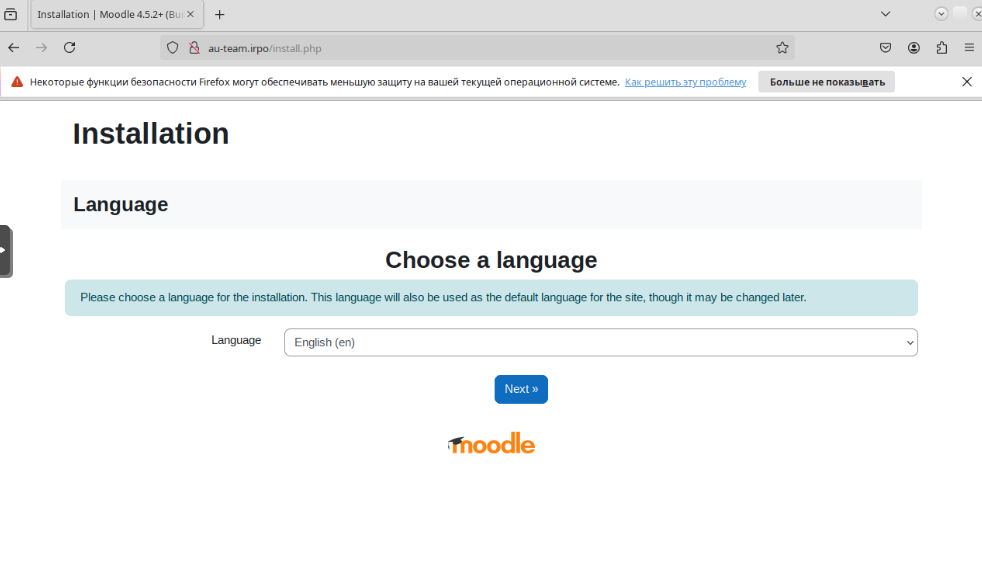
Перезапускаем веб-сервер:

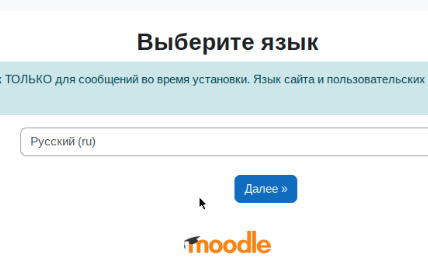
*systemctl restart httpd2*

**

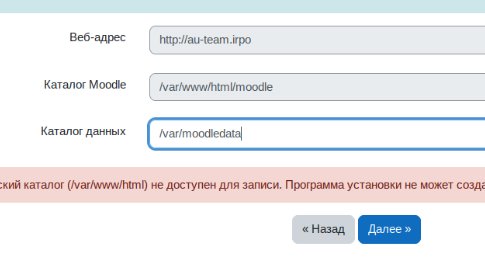
*Переходим на hq-cli, открываем браузер пишем домен нашего сайта, попадаем на страницу установки moodle*

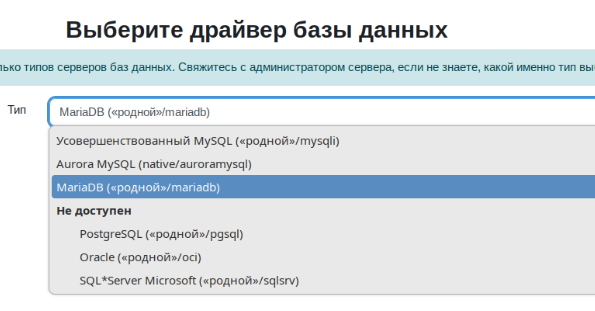
**http://au-team.irpo**

выбираем язык русский, нажимаем далее

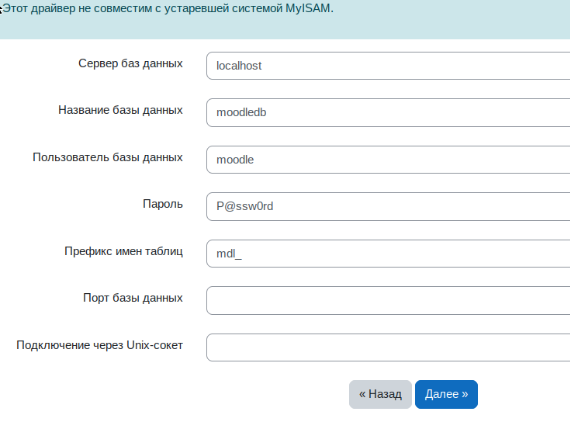
****

**изменяем каталог данных ( /var/moodledata )и подтверждаем пути**

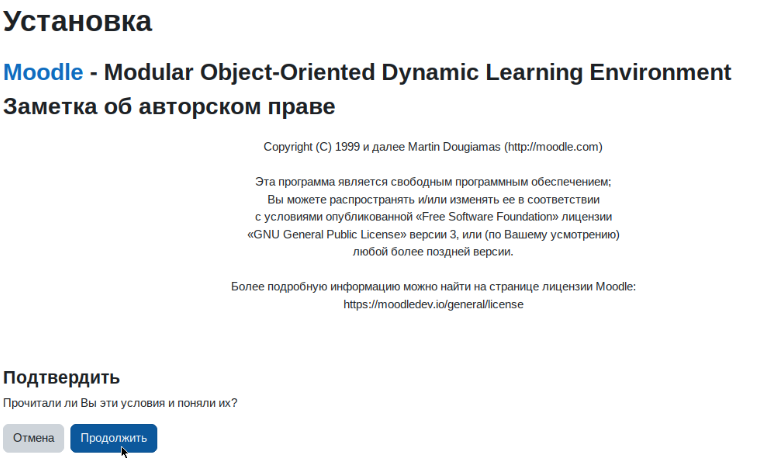
**Выбираем тип базы данных mariadb, подтверждаем**

****

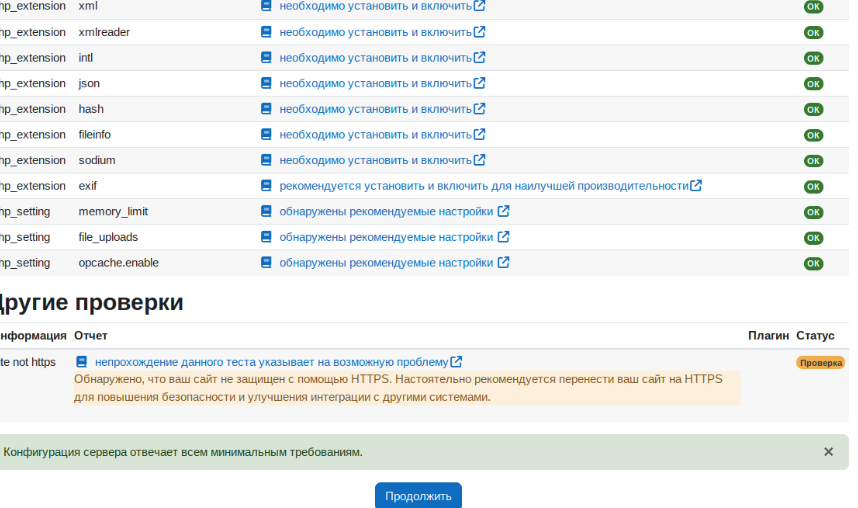
**Вносим параметры нашей базы данных, нажимаем далее**

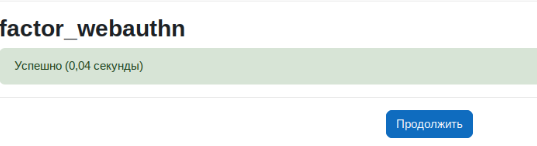
****

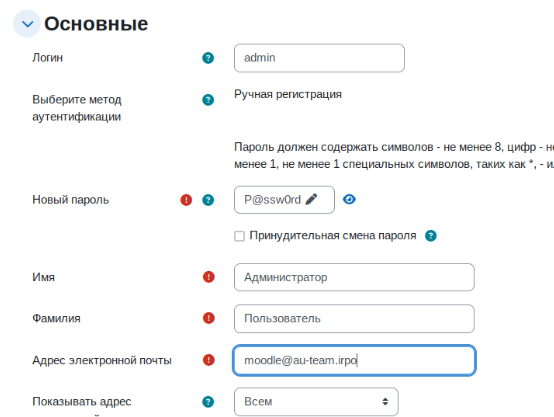
**принимаем лицензию**

****

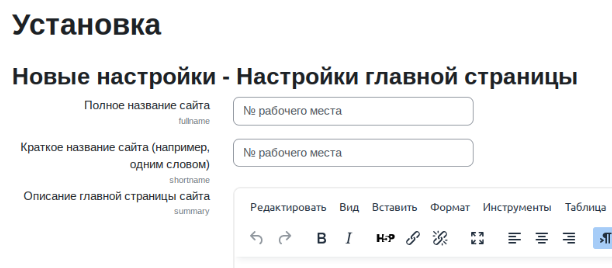
**проверяем настройки сервера (все ок зеленое) нажимаем продолжить**

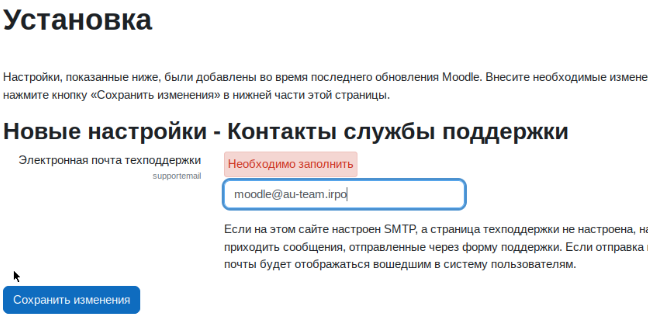
**ждем окончания установки, нажимаем продолжить**

**делаем первичные настройки (пароль** [**P@ssw0rd**](mailto:P@ssw0rd)**, емаил любой, нажимаем обновить профиль.**

****

**В наименовании сайта задаем номер рабочего места!! нажимаем сохранить изменения.**

**Задаем контакт поддержки, сохраняем**

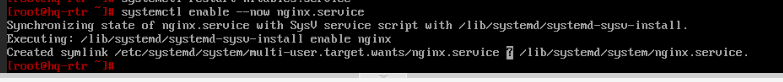


**moodle настроен!**

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

**Запуск и добавление в автозагрузку (рис. 8.1)**

systemctl enable --now nginx

рис.8.1

**Открываем на редактирование конфигурационный файл** Nginx (рис.8.2)

nano /etc/nginx/nginx.conf

рис.8.2

**Cпускаемся в конец HTTP таблици и перед последней фигурной скобкой }** **прописываем (рис. 8.3), соблюдая отступы!**

***server {***

***listen 80;***

***server\_name moodle.au-team.irpo;***

***location / {***

***proxy\_pass http://192.168.0.1:80;***

***}***

***}***

***server {***

***listen 80;***

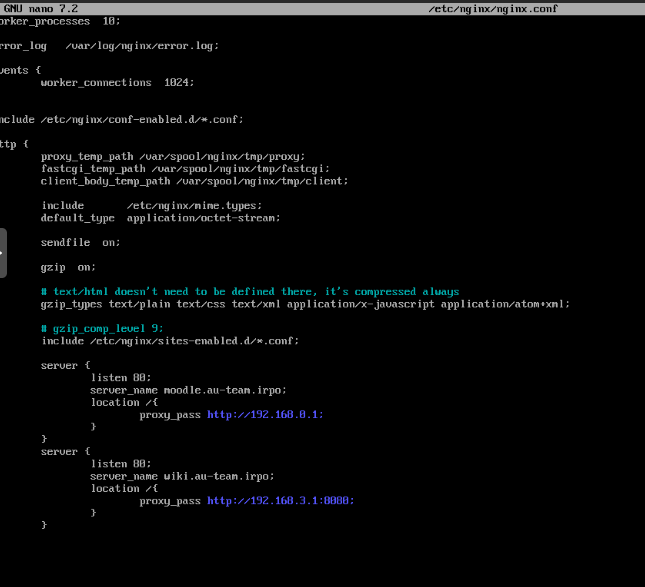
***server\_name wiki.au-team.irpo;***

***location / {***

***proxy\_pass http://192.168.3.1:8080;***

***}***

***}***

**рис. 8.3**

**Перезагружаем** Nginx

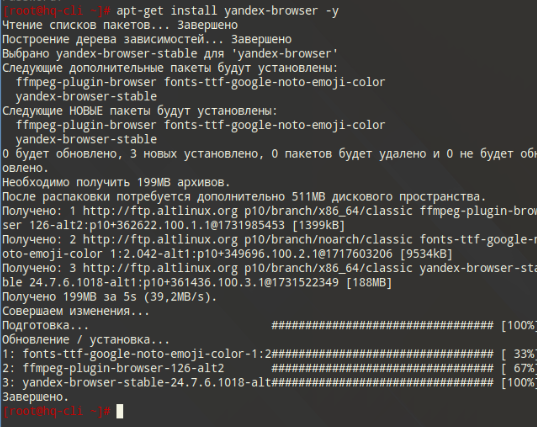
9. Установить яндекс браузер на CLI.

Заходим в терминал под root, обновляем бибдиотеки:

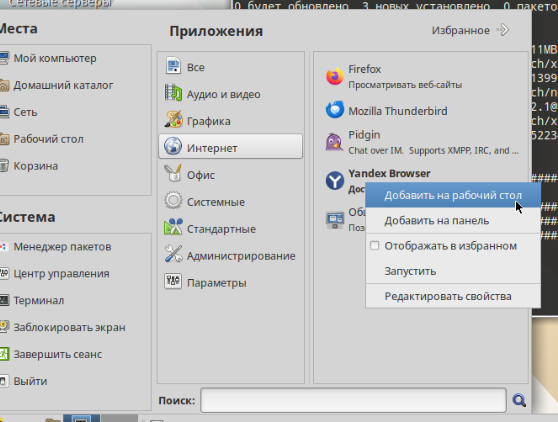
***apt-get update***

Устанавливаем яндекс браузер:

apt-get install yandex-browser -y



заходим в меню > все программы > интернет > Yandex Browser нажимаем пкравой кнопкой мыщи « Добавить на рабочий стол»

**Готово!!!**

**Модуль выполнен в полном объеме!!!**