***Гайд по демоэкзамену 2025***

***1 модуль***

Ситуативные команды:

apt-get update- для обновления пакетов

systemctl restart network (тут любой сервис, лучше после перезагрузки машины(если будет) то много сервисов перезагрузить

systemctl restart network NetworkManager system-resolved

systemctl restart dhcpcd- служба отвечает за автоматическое получение IP-адреса и других сетевых параметров от DHCP-сервера.

shutdown -r now-перезагрузка системы

[apt-get install nano NetworkManager-tui](file:///C:\\Users\\dvche\\Downloads\\Telegram%20Desktop\\Гайд%20по%20демоэкзамену%20модуль%201%20(2)%20(2).docx" \l "nmtui)- команда для установки nmtui(после установки следует перезагрузить сеть(systemctl start NetworkManager)

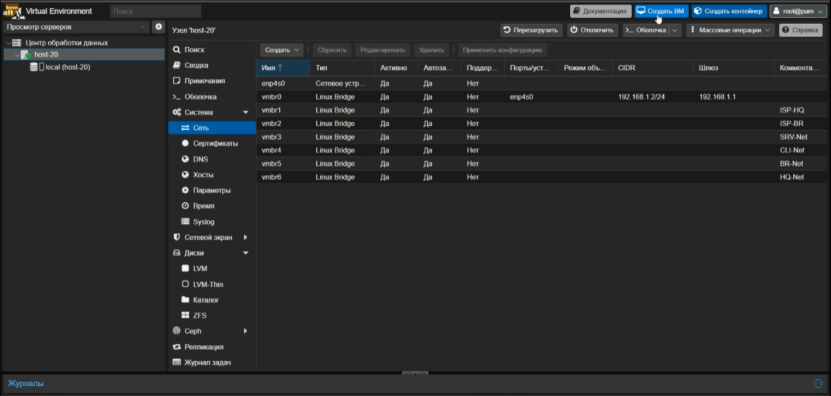
[нажми, что бы вернуться к модулю 1](file:///C:\Users\dvche\Downloads\Telegram%20Desktop\Гайд%20по%20демоэкзамену%20модуль%201%20(2)%20(2).docx#nmtui2)(на установку nmtui)

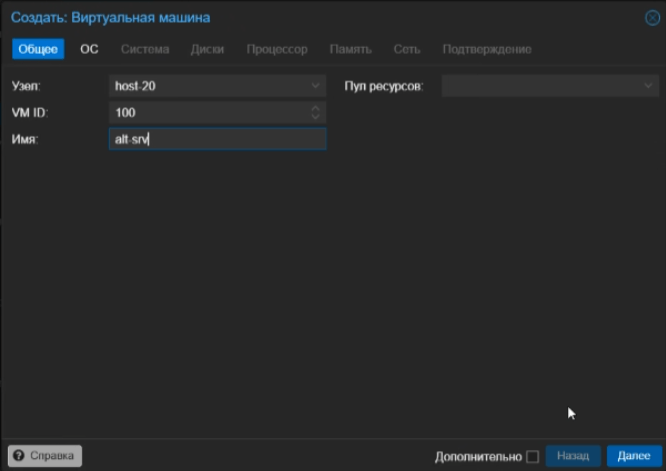
ip addr del 192.168.15.55/24 dev eth1 – удаление ip

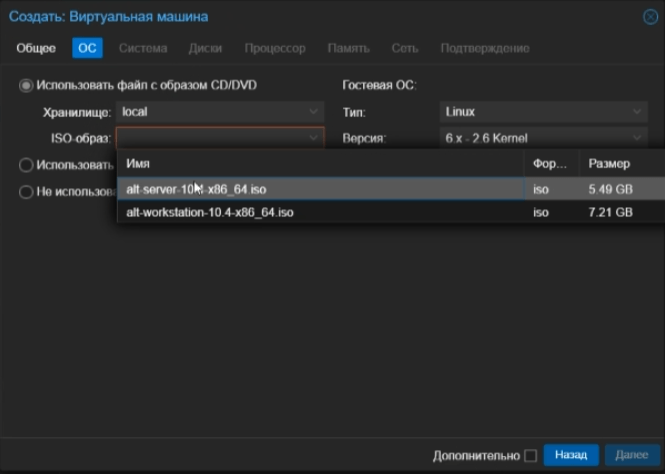
ip link set enp(указывается какой интерфейс) up – включение int

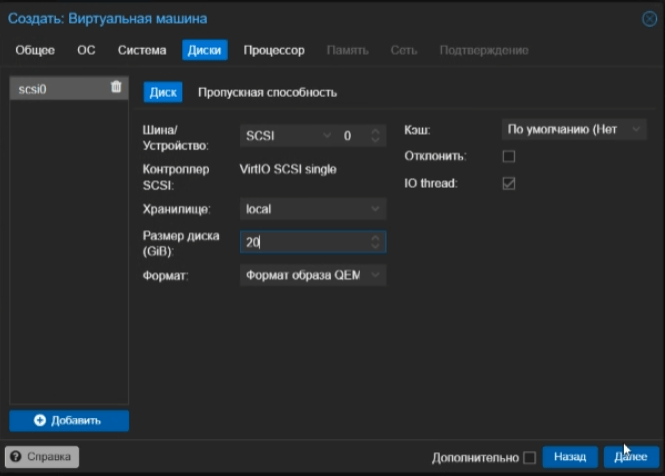
apt-get install nano – устанавливает nano

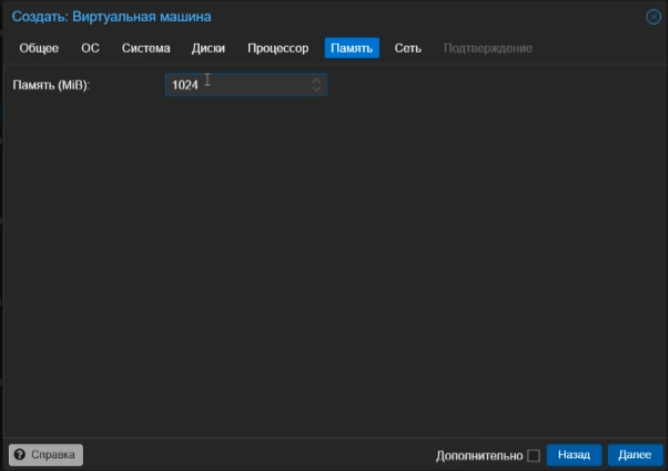
Создание шаблона для сервера

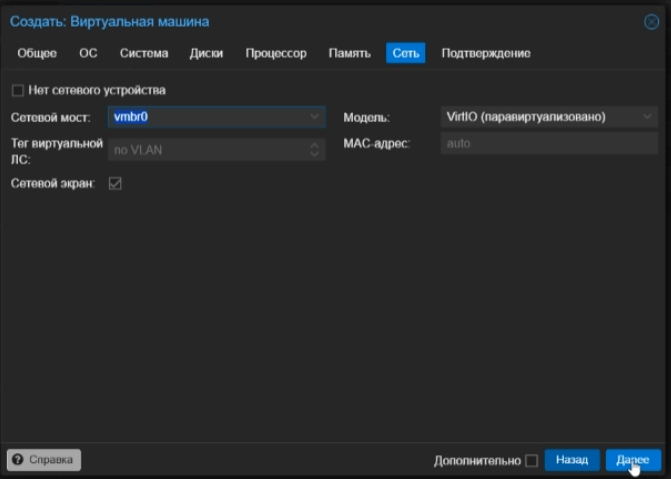


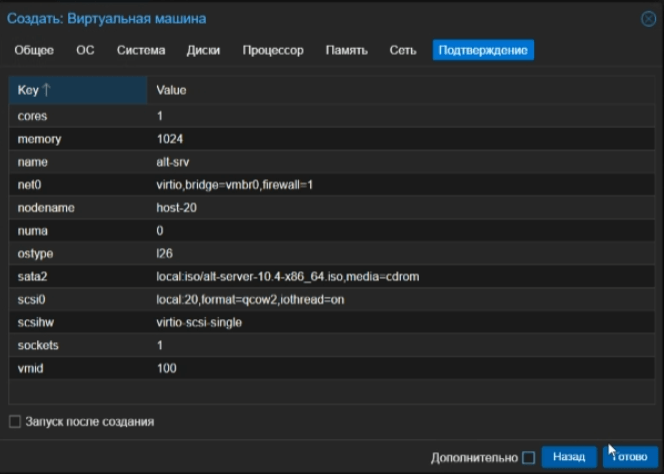




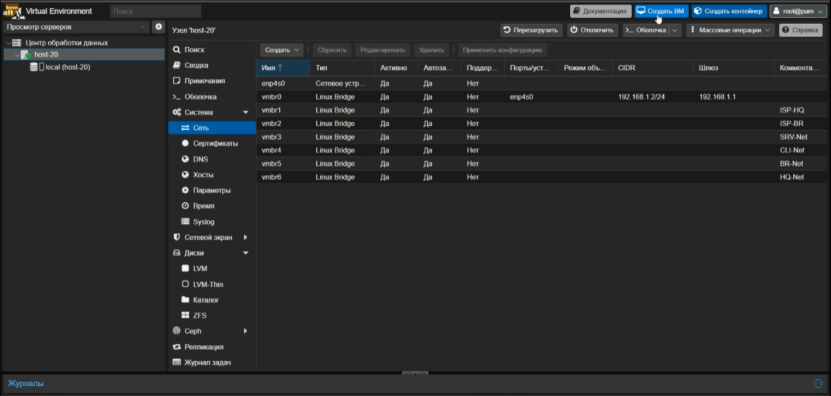


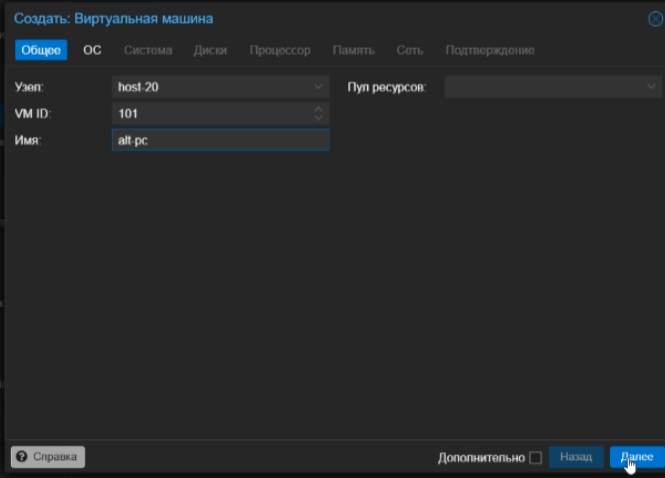


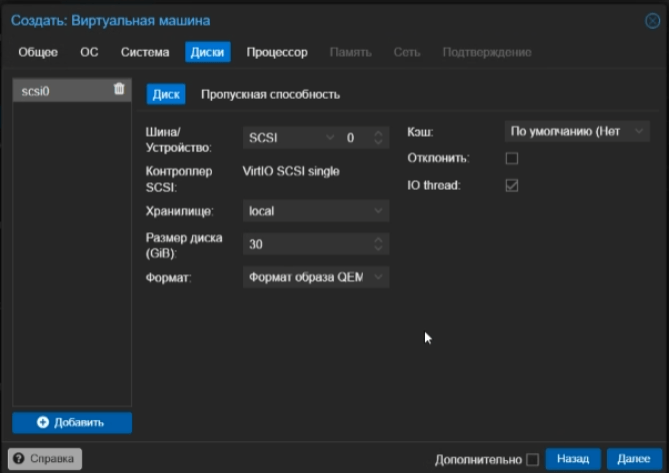
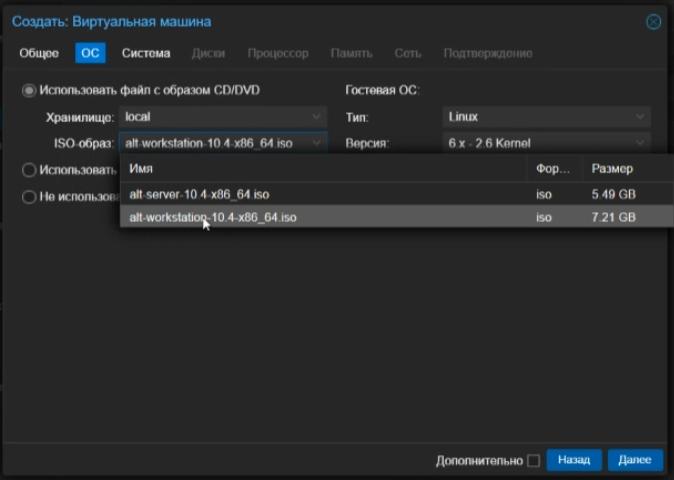


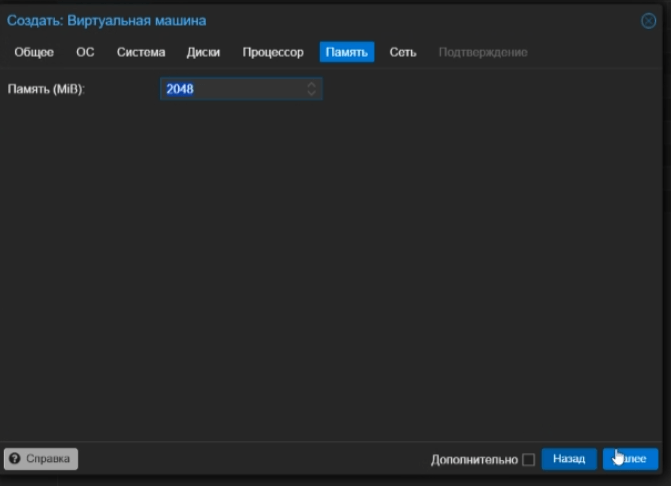


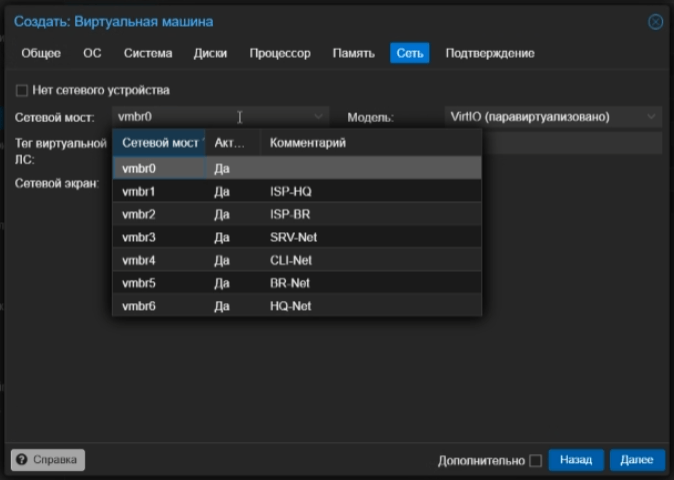
Создание шаблона для клиента

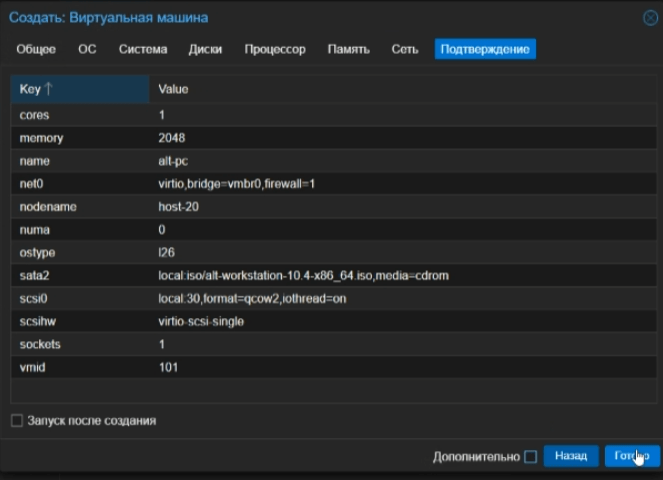








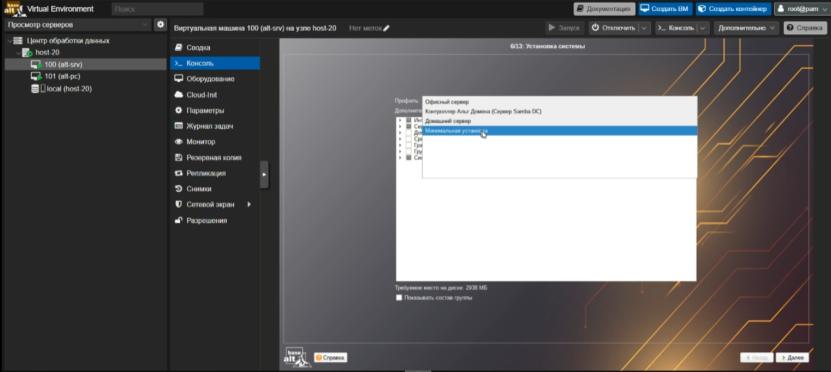




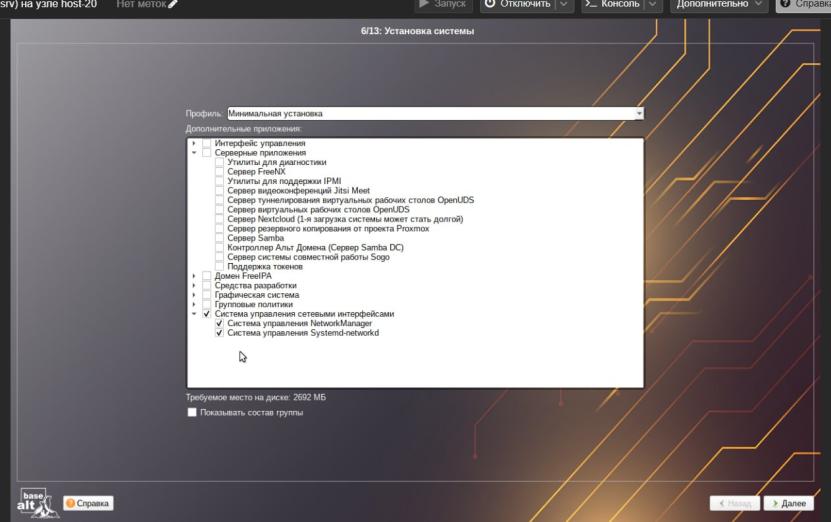
Установка сервера

Тыкаем далее до момента установки системы

Выбираем – Минимальная установка

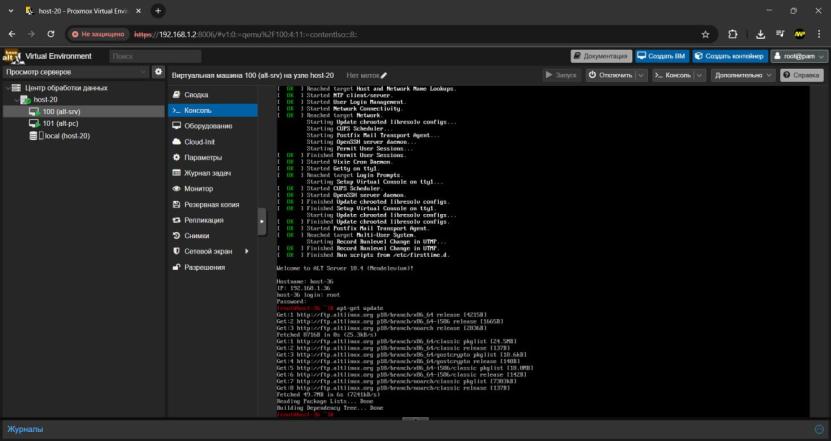


Поставить галочки как здесь

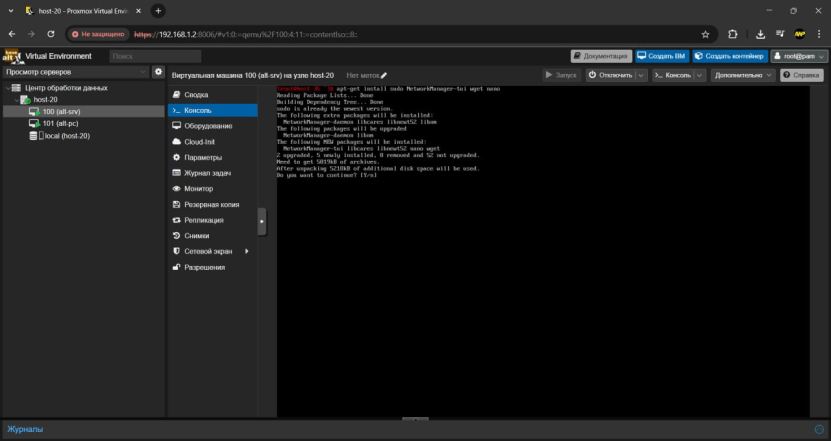




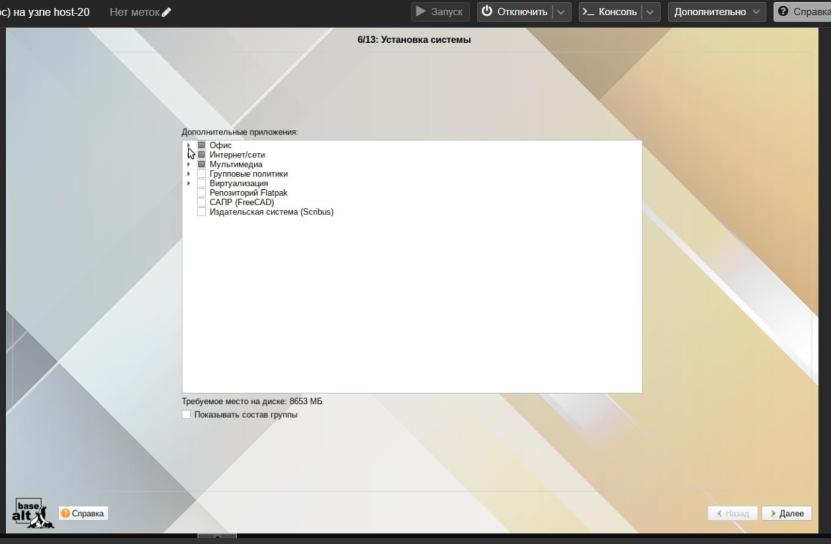




Устанавливаем Network-Manager на сервак



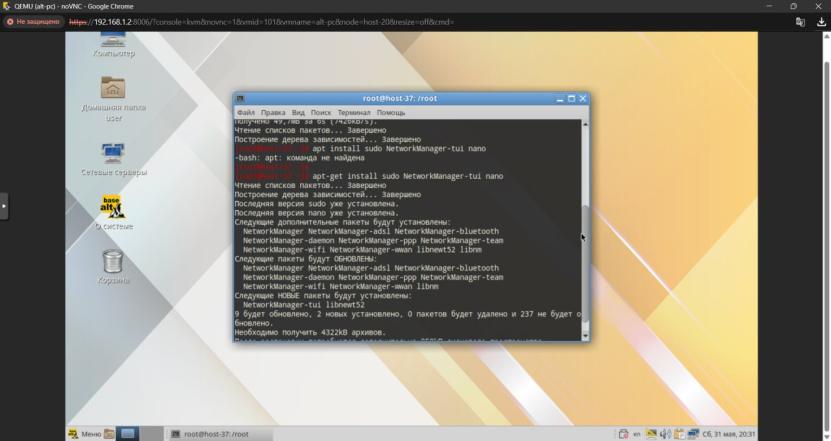
Установка клиента



У клиента также оставляем DHCP

Настройка клиента

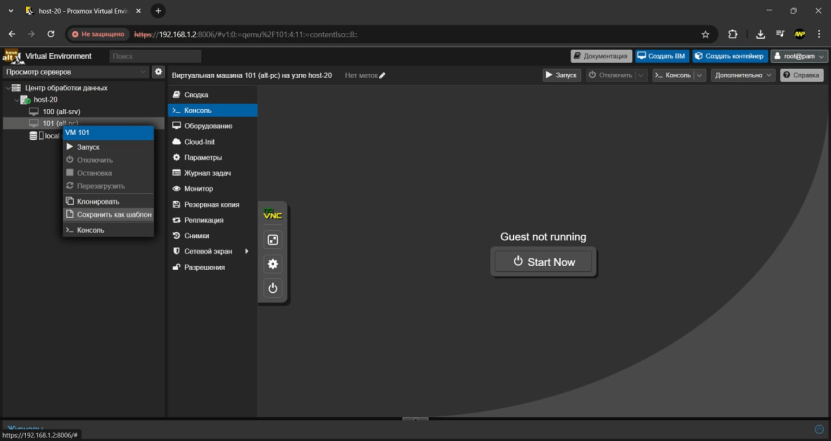


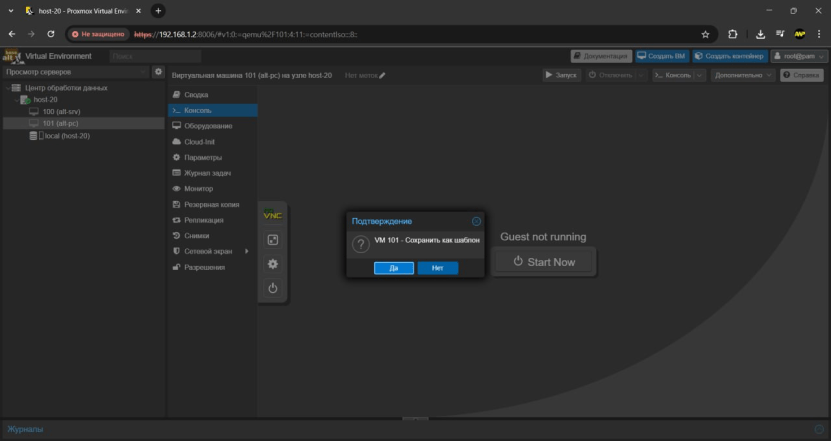


Добавляем в автозагрузку Network-Manager

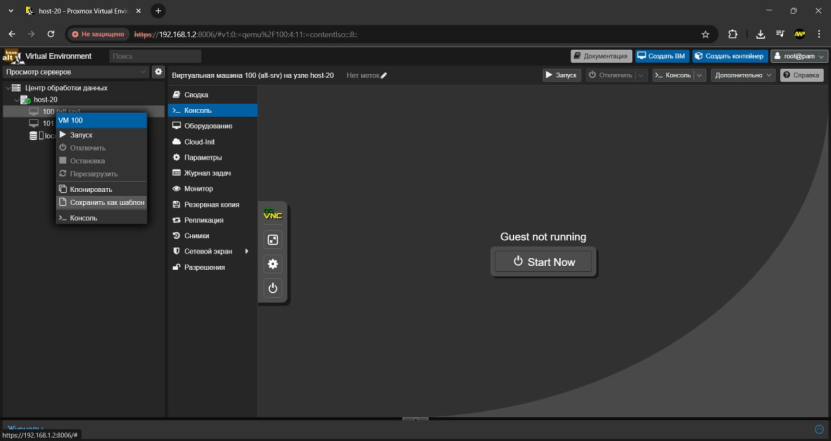


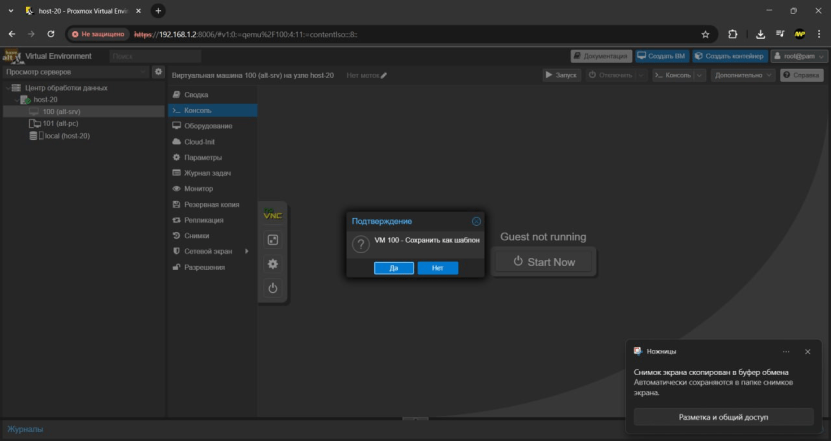
Сохраняем как шаблон клиента



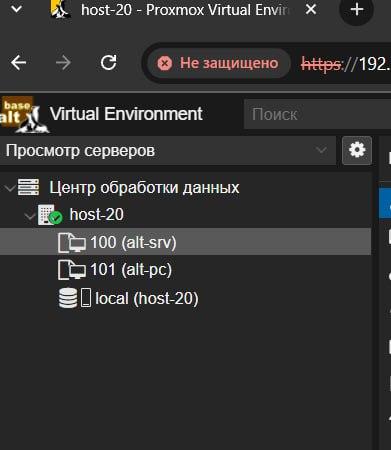


Сохраняем шаблон сервера

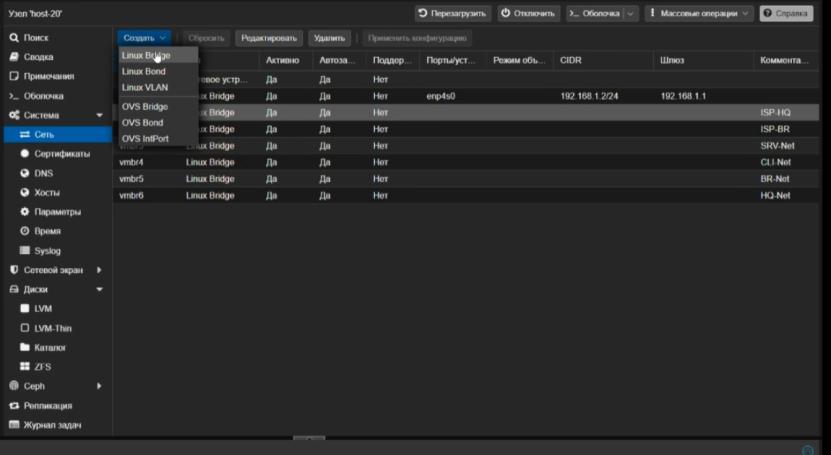


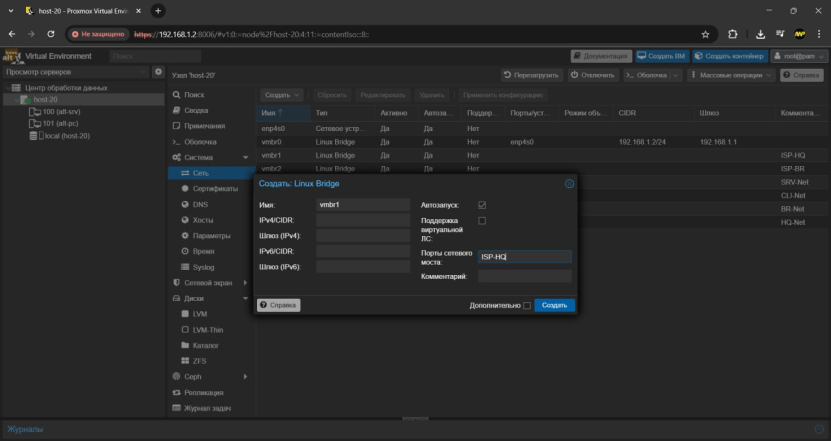


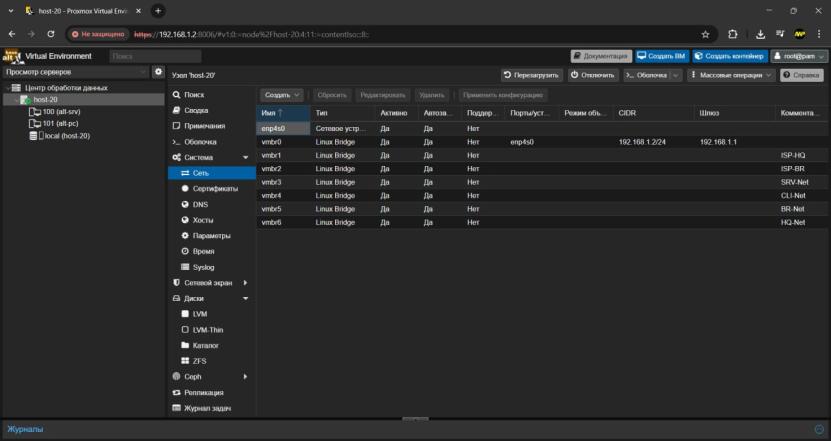
Результат подтверждающий что шаблон создан



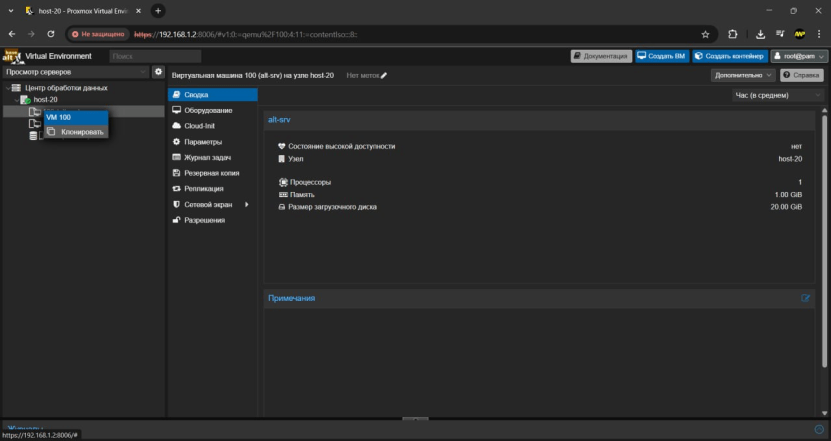
Создание виртуальных сетей



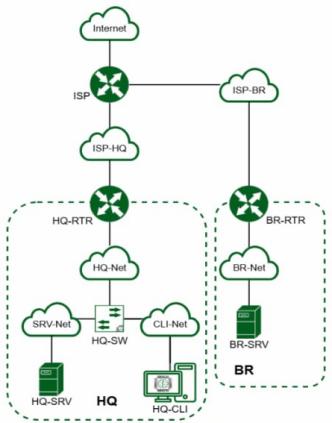




И с шаблона уже клонируем машины

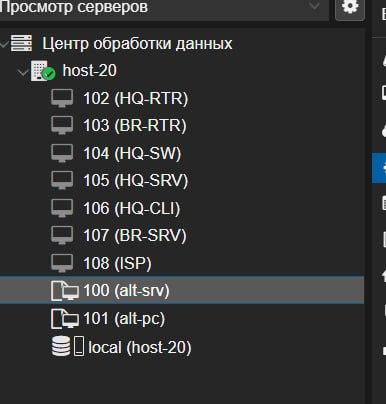


Топология сети



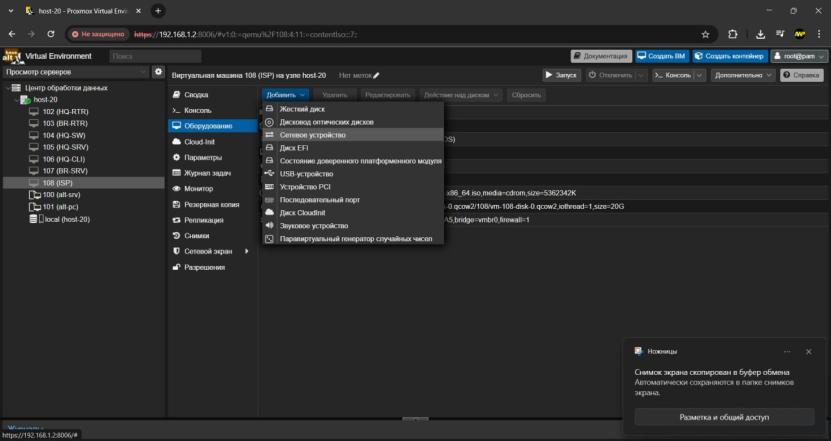
Клонируем машины с шаблонов, аlt-srv=>пкм=>клонировать

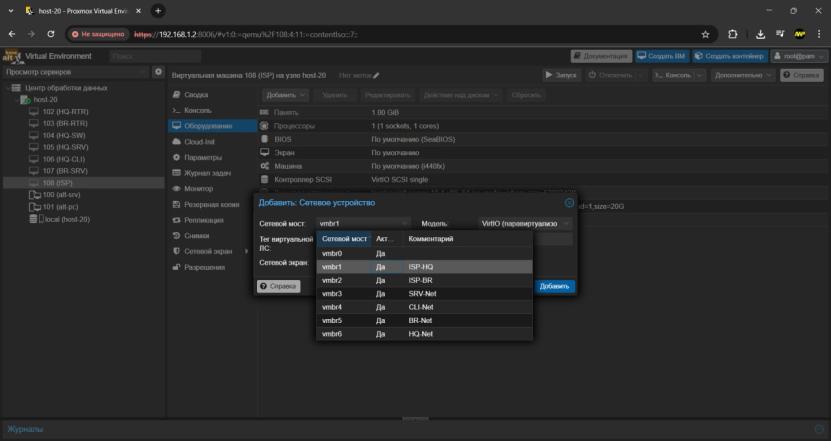
И так несколько раз до результата такого



Добавление виртуальных адаптеров

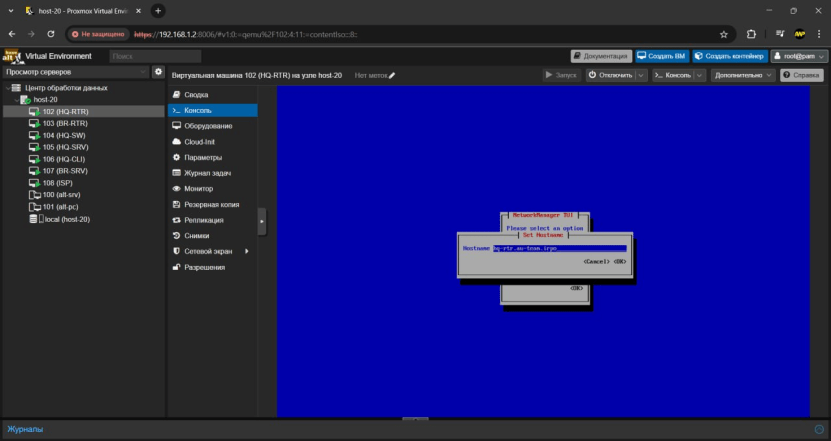
Соответствующий адаптер нужной машине



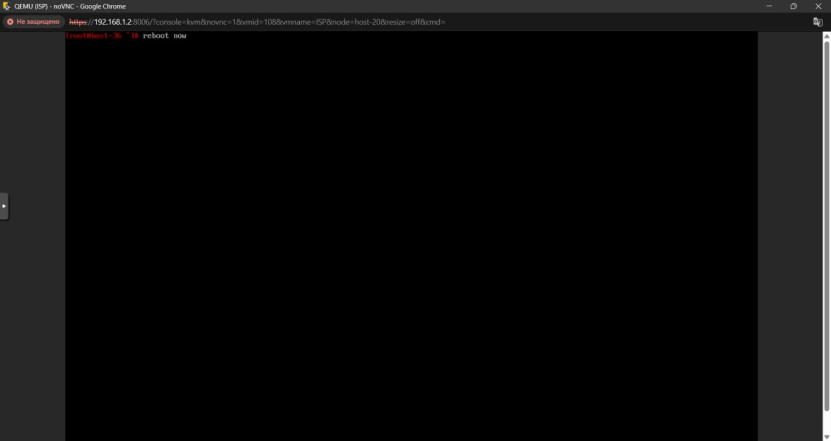


**Настройка имен в nmtui(соответствующее имя нужной машине)**

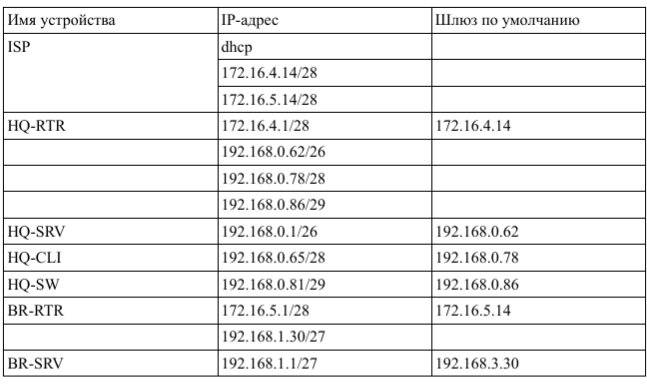
**После имени добавляем .au-team.irpo(КРОМЕ ISP)**



После установки имени перезагружаем машину

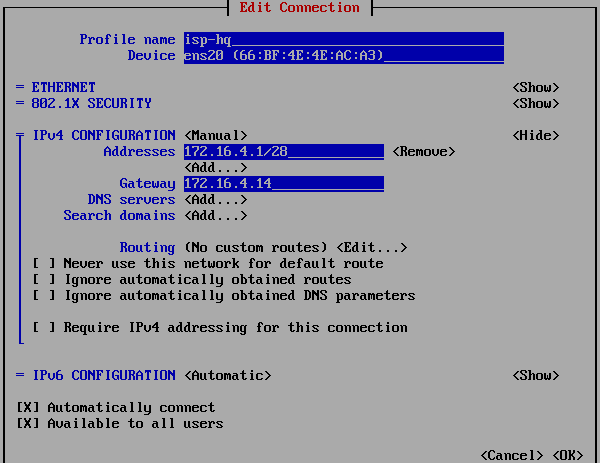


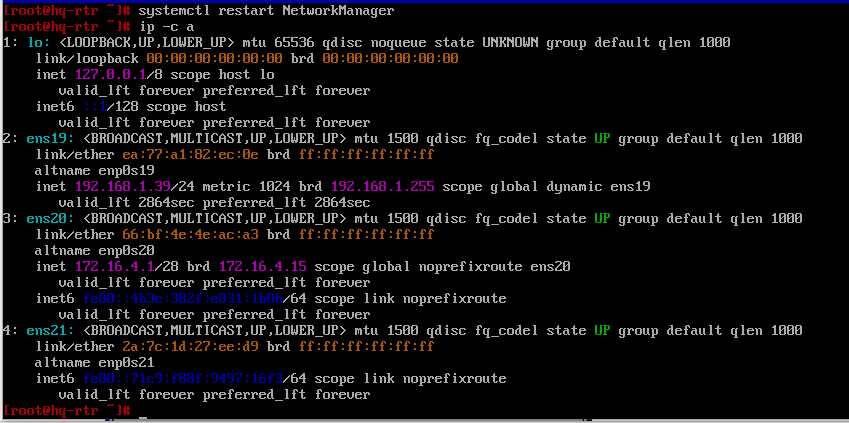
**Настройка адресации**

****

На hq-rtr

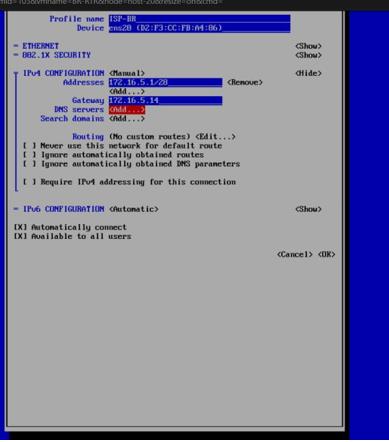
!!!НА HQ-RTR настраиваем только интерфейс смотрящий на ISP!!!!!!!

****

****

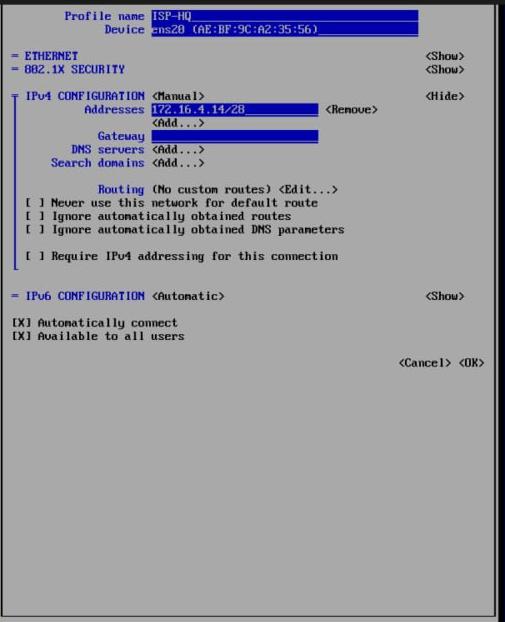
Адрес поставился

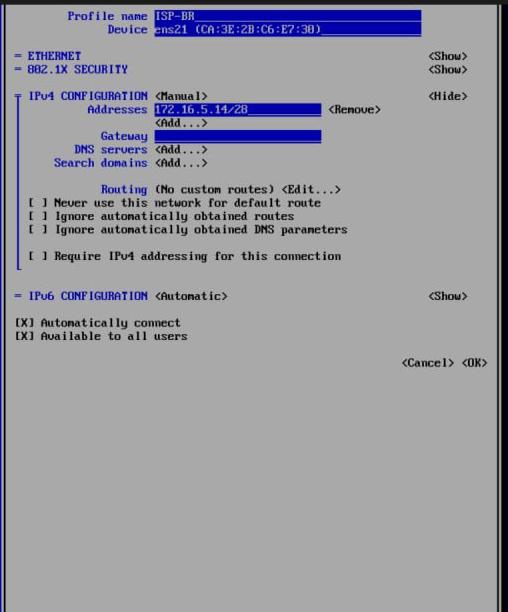
На br-rtr

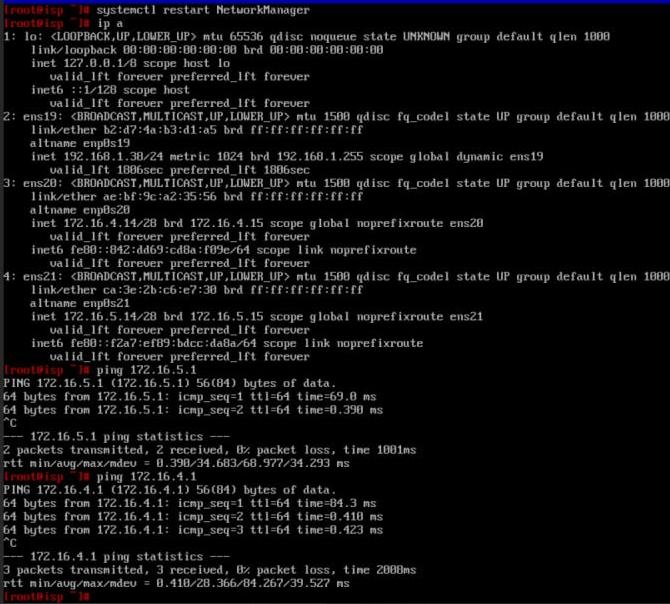


Перезагружаем NetworkManager

На isp







**Настройка маршрутизации(только на роутерах)**

nano /etc/net/sysctl.conf

****

сохраняем изменения ctrl+o, enter.

Выходим ctrl+x.

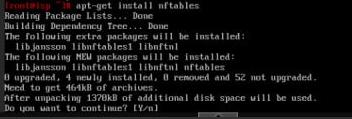
Вводим команду sysctl -p.

**Настройка доступа в Интернет из сетей**

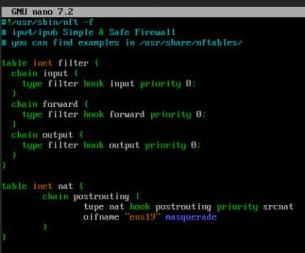
Устанавливаем nftables((настраивается на ISP, HQ-R и BR-R)

После установки следует добавить в автозагрузку nftables

Далее рестартнуть

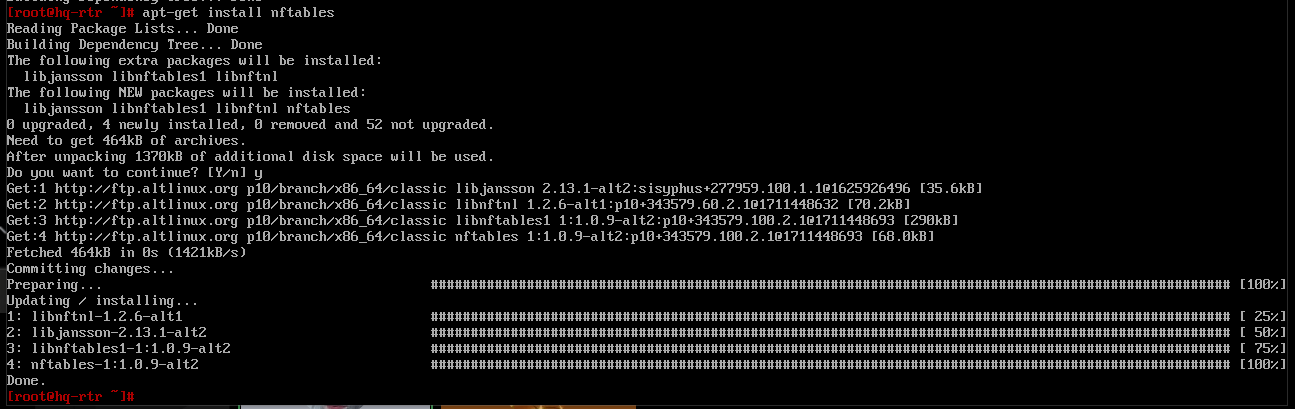


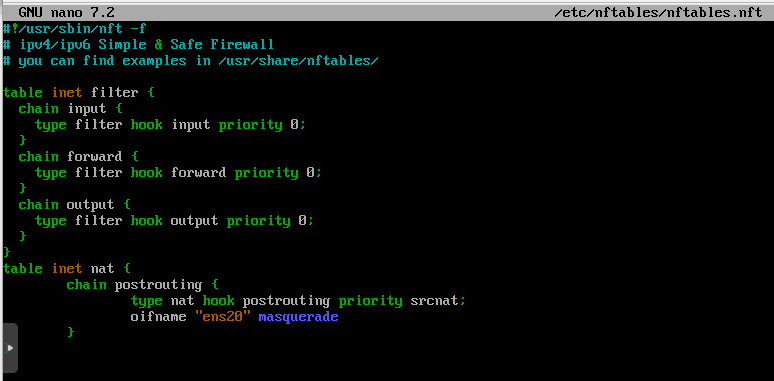
nano /etc/nftables/nftables.nft(на ISP)



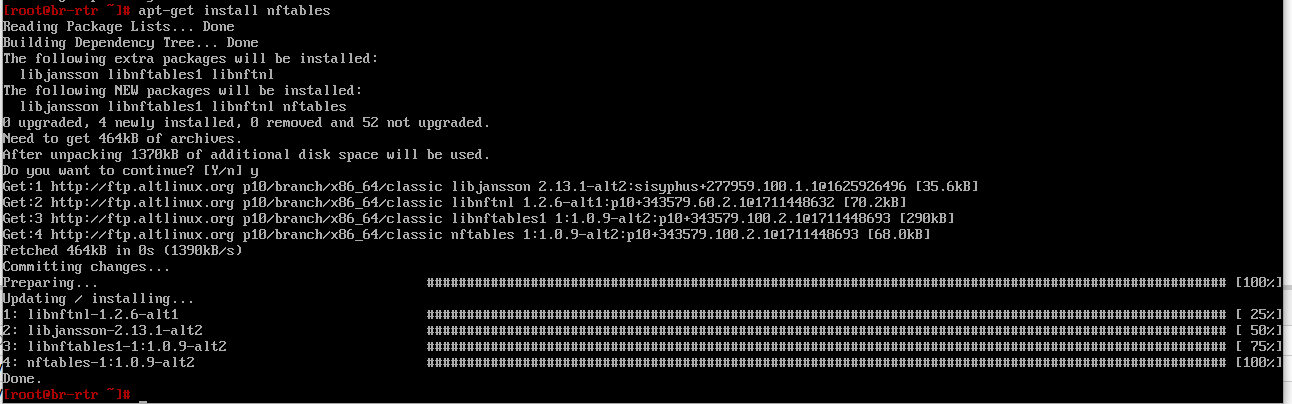
systemctl restart nftables

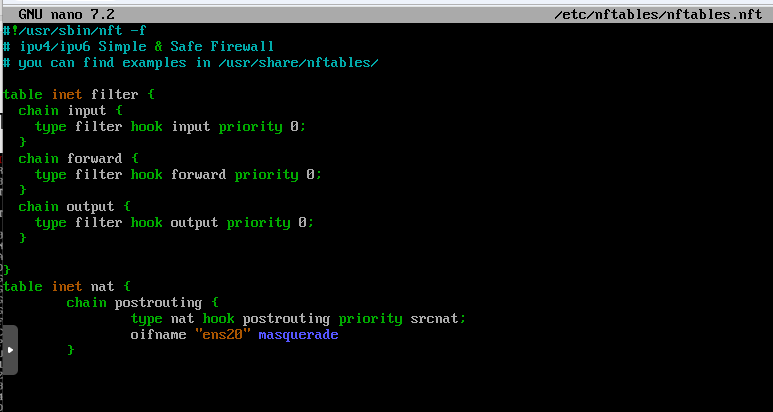
Теперь тоже самое на hq-rtr





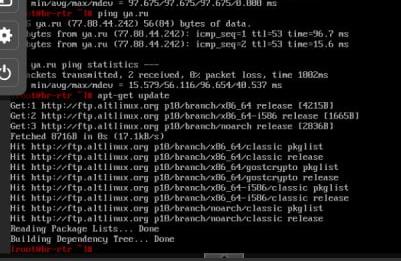
Также и на br-rtr

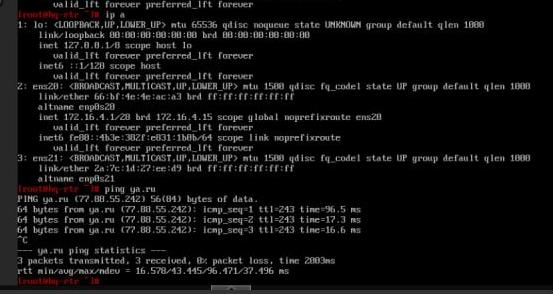




После перезагрузки nftables и NetworkManager на роутерах

Проверяем

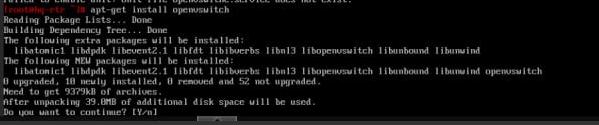
****

****

**Настройка на hq-rtr в сторону офиса hq виртуальный коммутатор**

* Сервер HQ-SRV должен находиться в ID VLAN 100
* Клиент HQ-CLI в ID VLAN 200
* Создайте подсеть управления с ID VLAN 999

Устанавливаем openswitch





Создаем мост (виртуальный коммутатор) hq-sw,

Добавляем порт ens20 к hq-sw и назначаем ему vlan100,

Добавляем порт ens21 к hq-sw и назначаем ему vlan200,

Добавляем порт ens22 к hq-sw и назначаем ему vlan999,

Добавляем внутренний порт vlan100, vlan200, vlan999 к мосту hq-sw

На устройстве hq-sw не добавляем кроме виртуальных сетей

На hq-rtr должно быть 4 интерфеса hq-cli, hq-srv, hq-sw, isp

***ovs-vsctl add-br hq-sw***

***ovs-vsctl add-port hq-sw ens20  tag=100***

***ovs-vsctl add-port hq-sw ens21 tag=200***

***ovs-vsctl add-port hq-sw ens22 tag=999***

***ovs-vsctl add-port hq-sw vlan100 tag=100 -- set interface vlan100 type=internal***

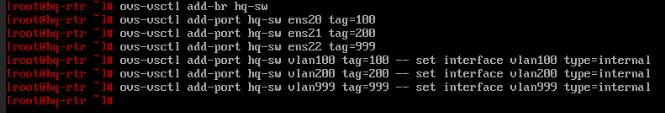
***ovs-vsctl add-port hq-sw vlan200 tag=200 -- set interface vlan200 type=internal***

***ovs-vsctl add-port hq-sw vlan999 tag=999 -- set interface vlan999 type=internal***

192.168.0.62/26 vlan100 hq-srv

192.168.0.78/28 vlan200 hq-cli

192.168.0.86/29 vlan999 hq-sw



Проверяйте интерфейсы, указываются в ip -c a hq-rtr

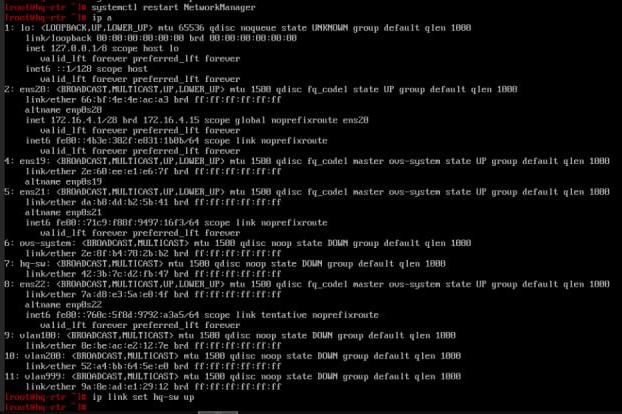
Сопоставьте нужный vlan с устройством а устройство с интерфейсом который добавили на роутере hq

Перезагружаем openvswitch и NetworkManager



Включаем мост

***ip link set hq-sw up***



Создаем скрипт для настройки и восстановления настройки ip-

адресов VLAN после перезагрузки!!!

создаем скрипт

пишем текст скрипта:

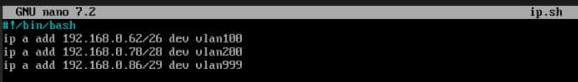
**#!/bin/bash/**

**ip a add 192.168.0.62/26 dev vlan100**

**ip a add 192.168.0.78/28 dev vlan200**

***ip a add 192.168.0.86/29 dev vlan999***

nano ip.sh

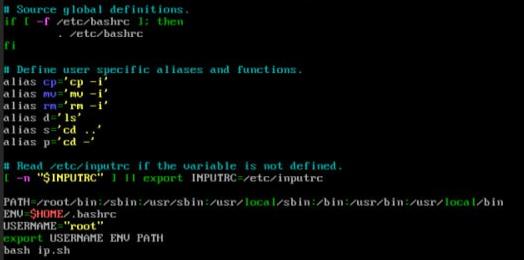


заходим добавляем запуск при загрузке

nano `/.bashrc

добавляем строчку, и сохраняем.

bash ip.sh



Запускаем скрипт

bash ip.sh



На hq-srv добавить адрес

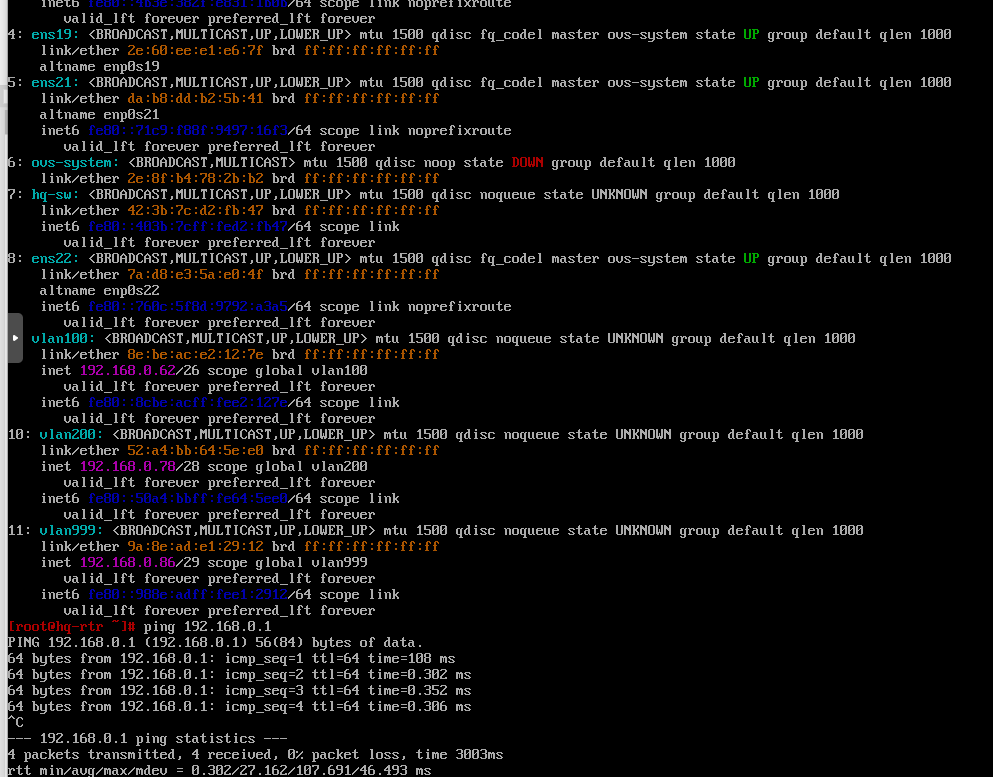


Чтобы поднять vlan на hq-rtr, нужно прописать команду

ip link set dev vlanXXX(указывайте какой) up

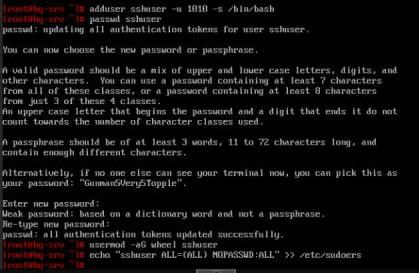
и затем systemctl restart NetworkManager

И проверяйте связь



**Настройте локальные учётные записи на устройствах в соответствии с заданием (на серверах пользователь sshuser c id 1010, на маршрутизаторах net\_admin)**

**На hq-srv и br-srv**



adduser sshuser -u 1010 -s /bin/bash

passwd sshuser

P@ssw0rd

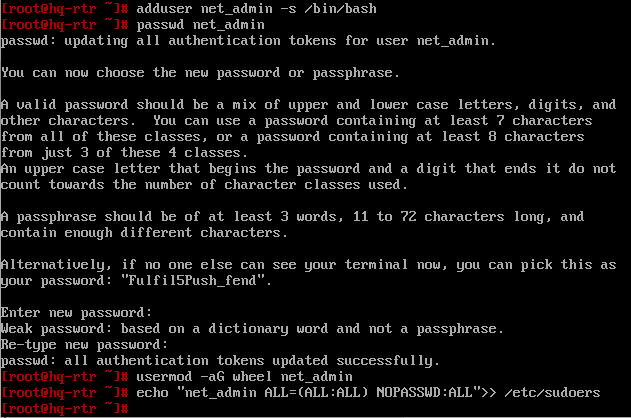
P@ssw0rd

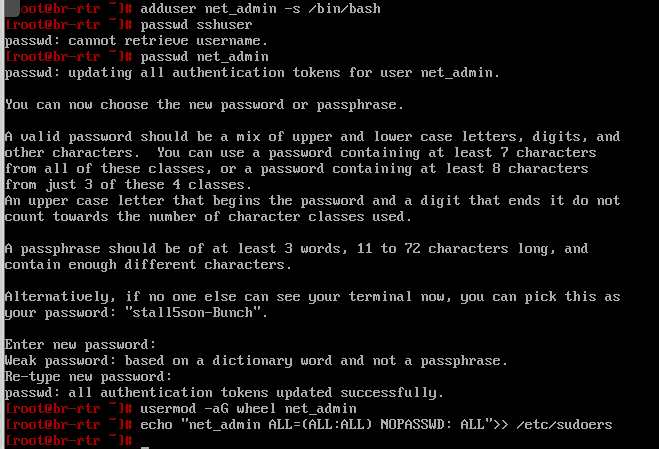
usermod -aG wheel sshuser

echo "sshuser ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

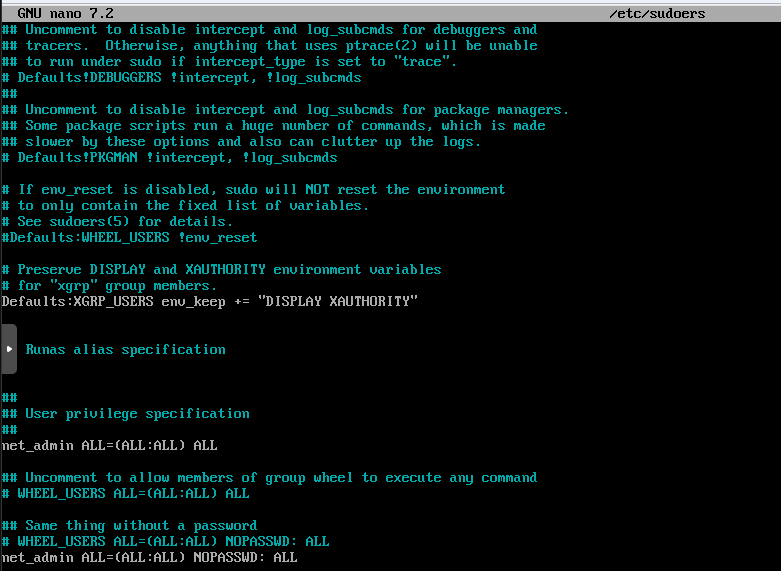


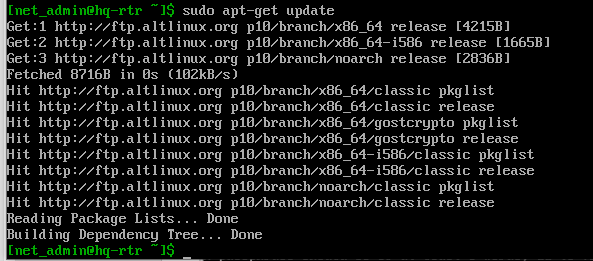
**На hq-rtr и br-rtr**





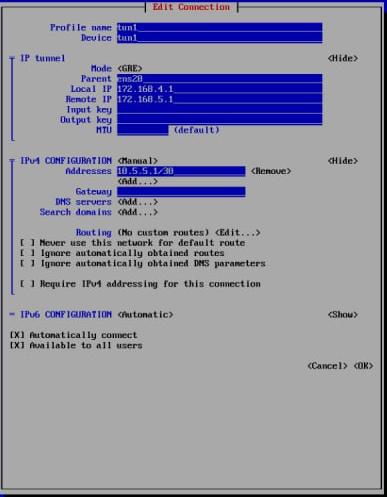
Привести файл sudoers к такому виду





**Настраиваем GRE-туннель между HQ-R и BR-R**

На hq-rtr

****

Для корректной работы протокола динамической маршрутизации

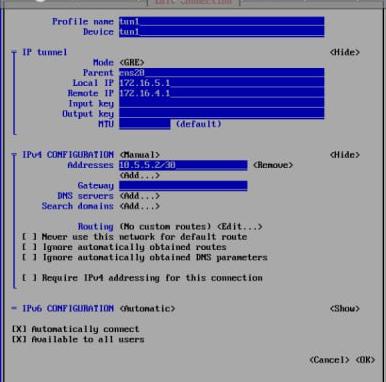
требуется увеличить параметр TTL на интерфейсе туннеля используется

команда (рис. 6.10):

**nmcli connection modify tun1 ip-tunnel.ttl 64 – ввод (Enter)**

****

На br-rtr



Для корректной работы протокола динамической маршрутизации

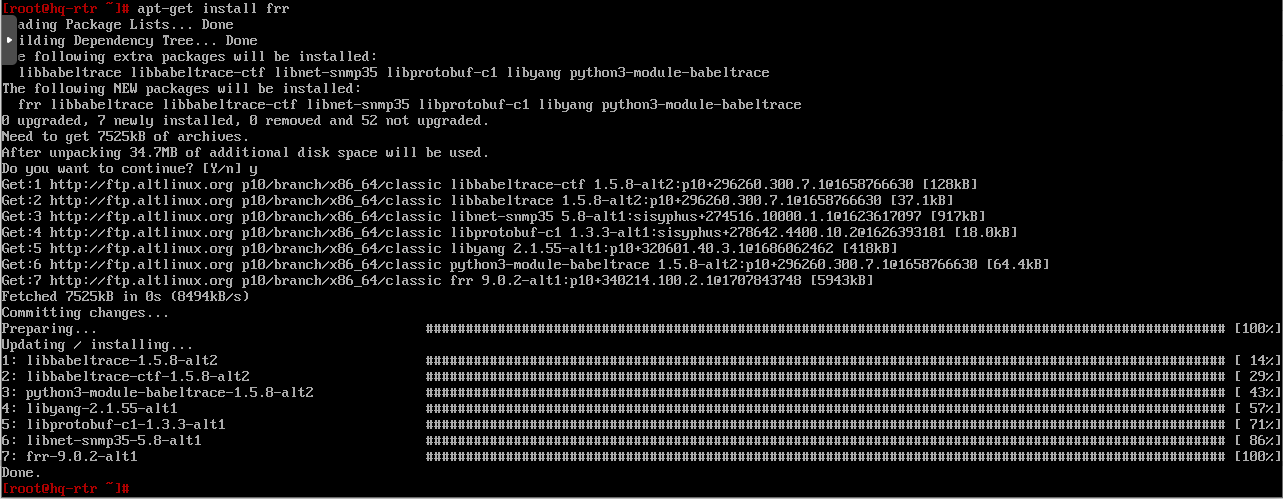
требуется увеличить параметр TTL на интерфейсе туннеля используется

команда (рис. 6.10):

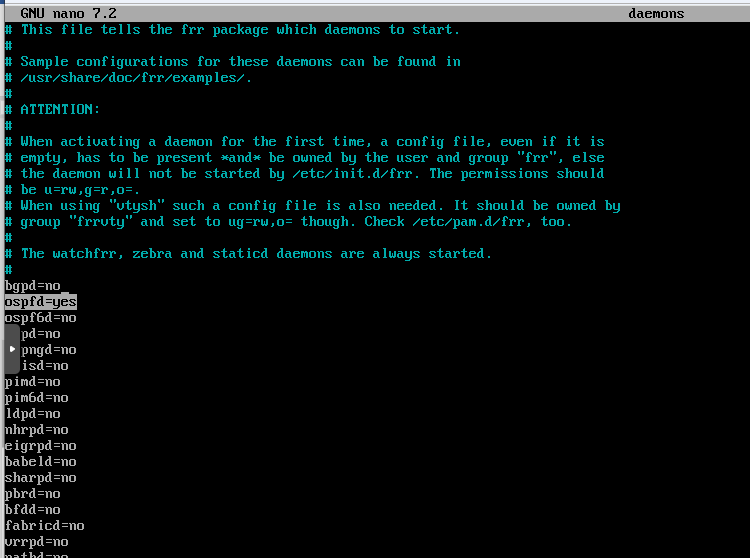
**nmcli connection modify tun1 ip-tunnel.ttl 64 – ввод (Enter)**

**Настройка динамической (внутренней) маршрутизации средствами frr**

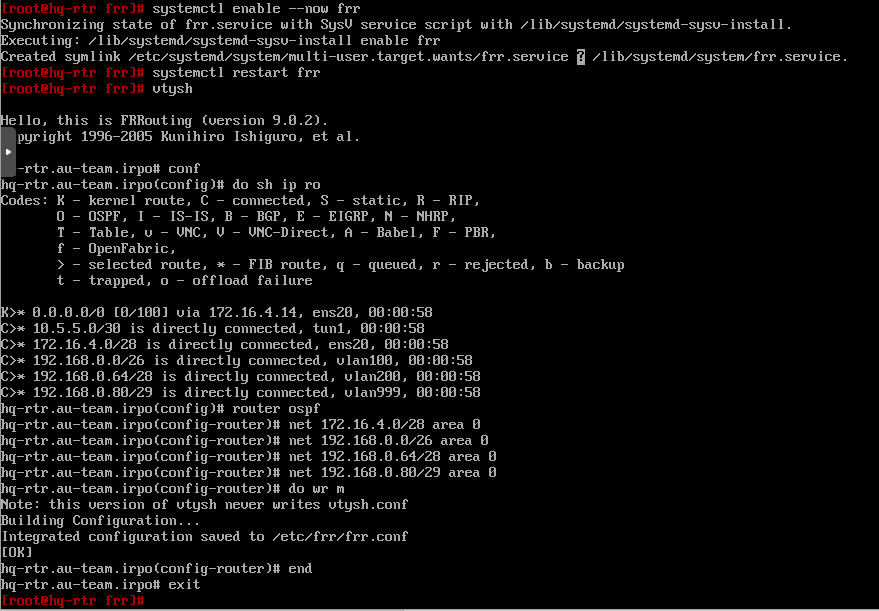
**!!!На hq-rtr и br-rtr!!!**

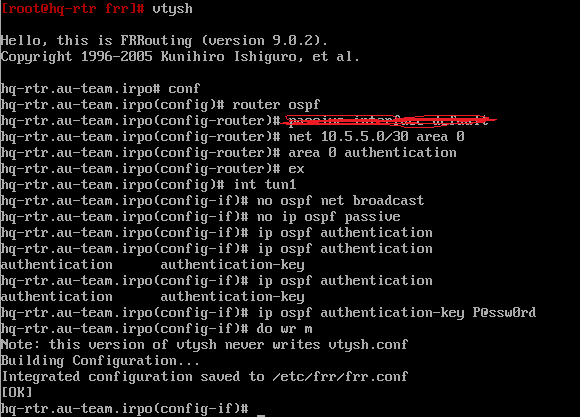


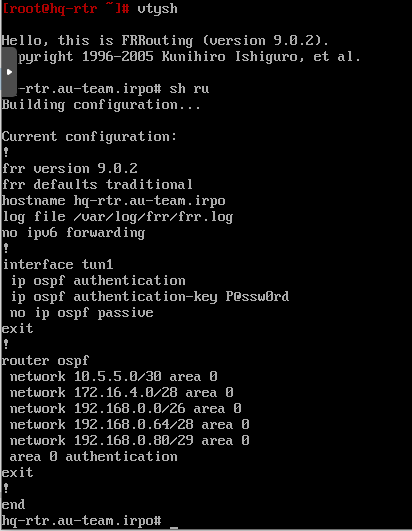
Переходим по /etc/frr/daemons и меняем no на yes



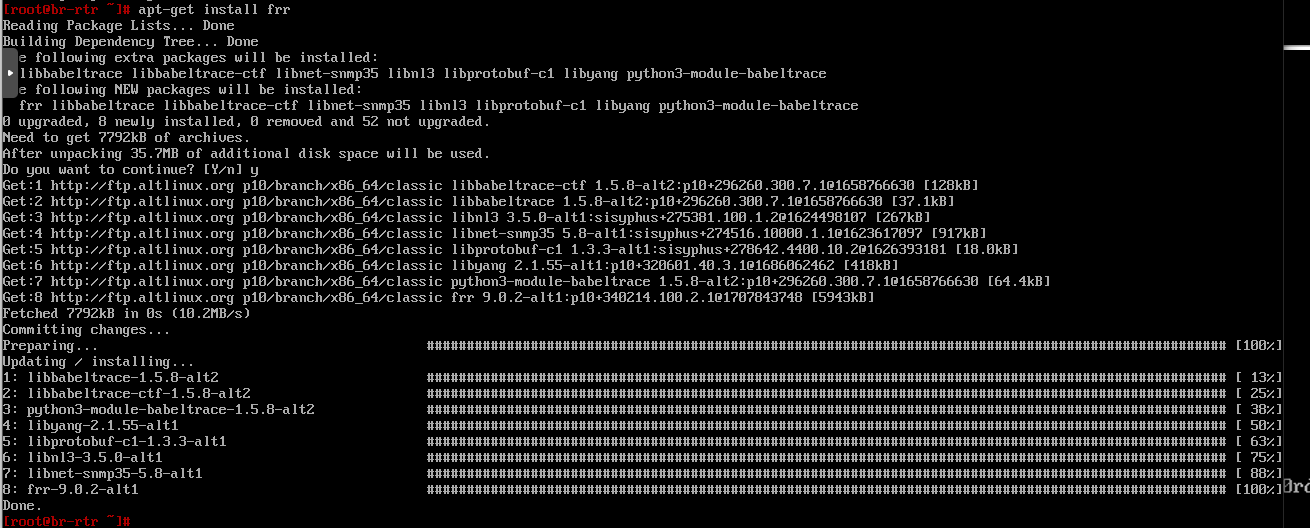
Переходим в vtysh, **passive-interface default(НЕ НУЖНО ПИСАТЬ)**

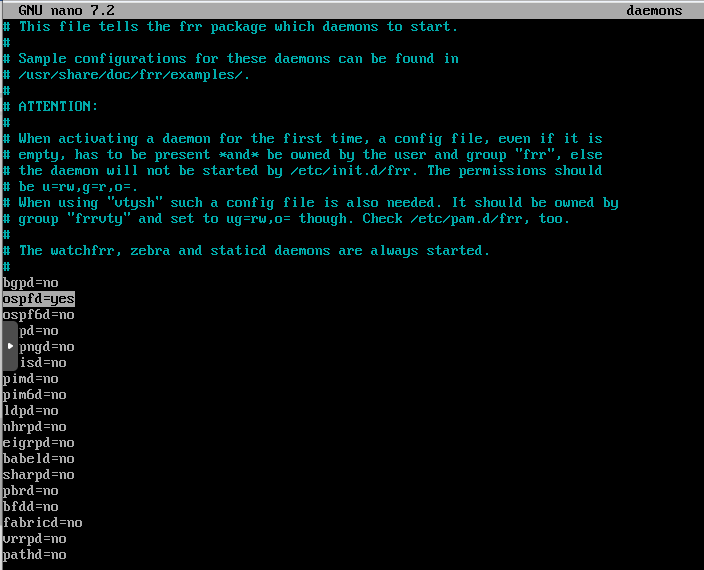


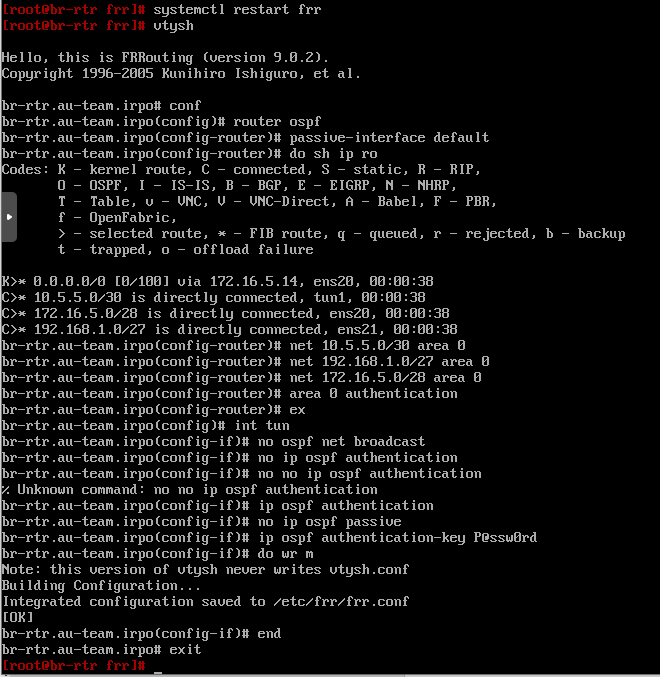




Теперь на br-rtr



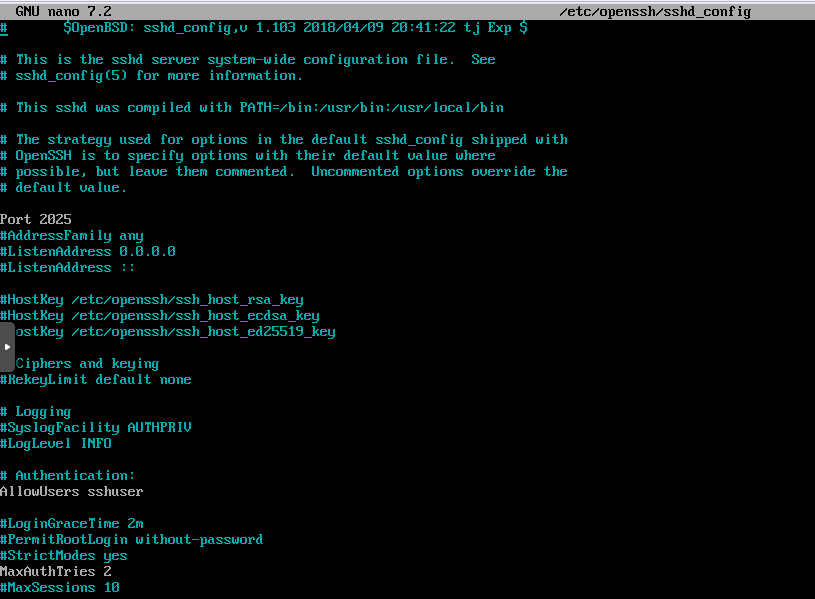




**ПРИМЕЧАНИЕ!!! может потребоваться перезагрузка BR-RTR и HQ-RTR, после нее канал поднимится.**

**Настройка ssh на HQ-SRV И BR-SRV**

Переходим в файл sshd\_config по пути nano /etc/openssh/sshd\_config





Сохраняем файл и рестартуем sshd

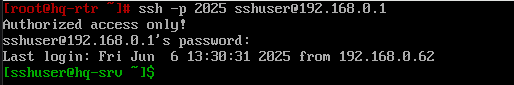
Создаем баннер

Переходим по пути nano /etc/banner

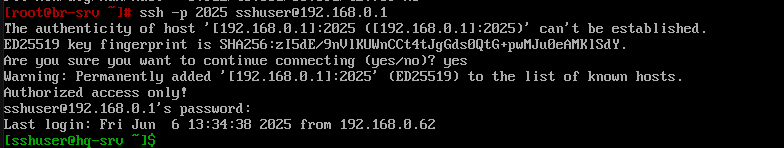
И вписываем Authorized access only!



Проверяем работу



С br-srv



**Настройка DHCP-сервера**

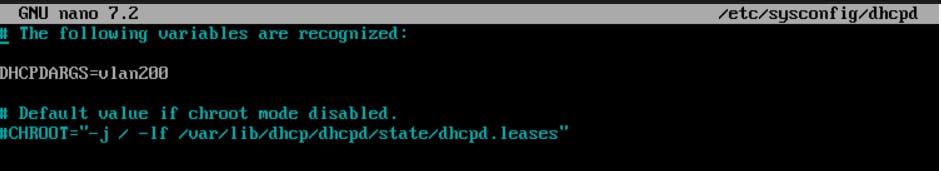
Скачать dhcp – apt-get install dhcp-server

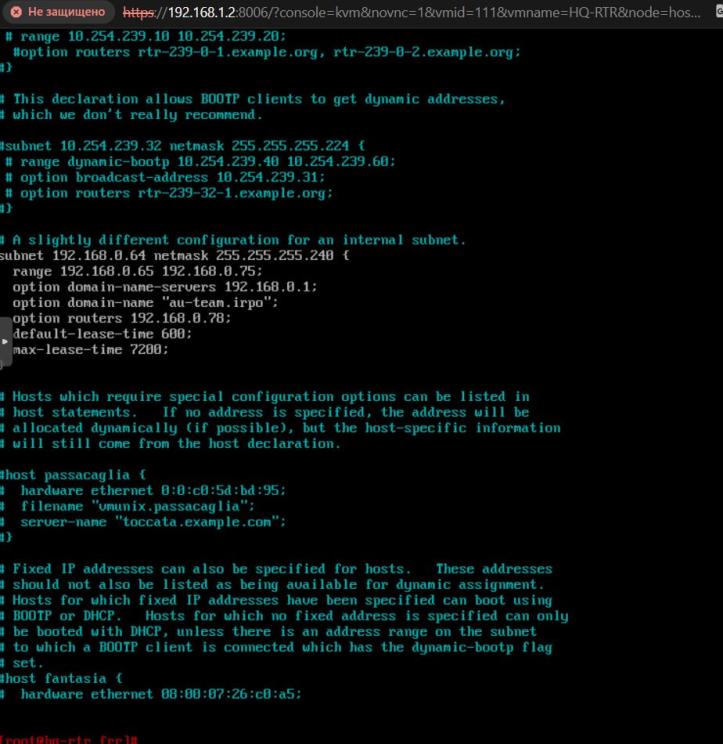
Потом копируем файл dhcpd.conf.example в dhcpd.conf



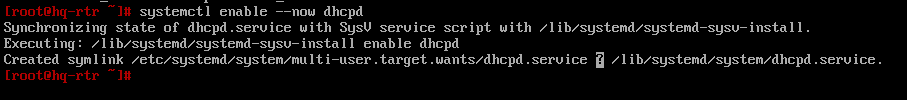
Укажим сетевой интерефейс для IPv4, через который будет работать DHCP-сервер

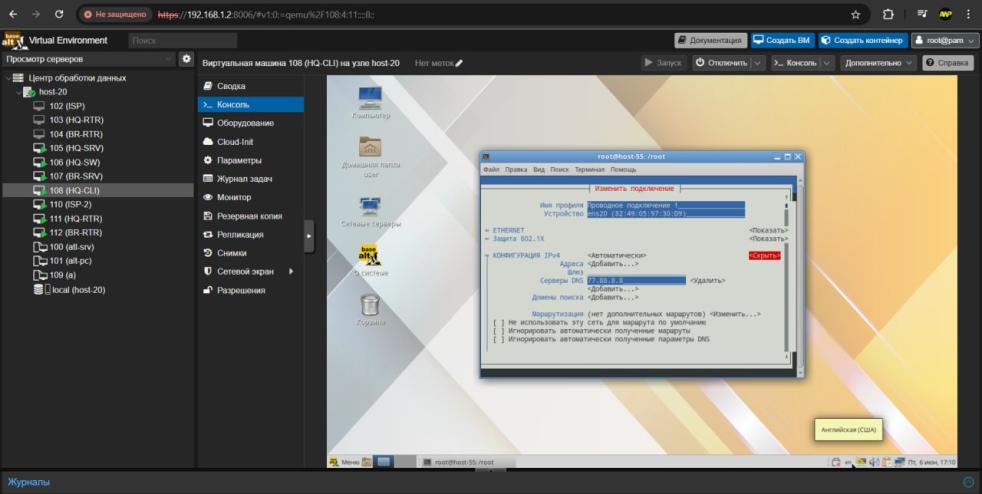
**nano /etc/sysconfig/dhcpd**

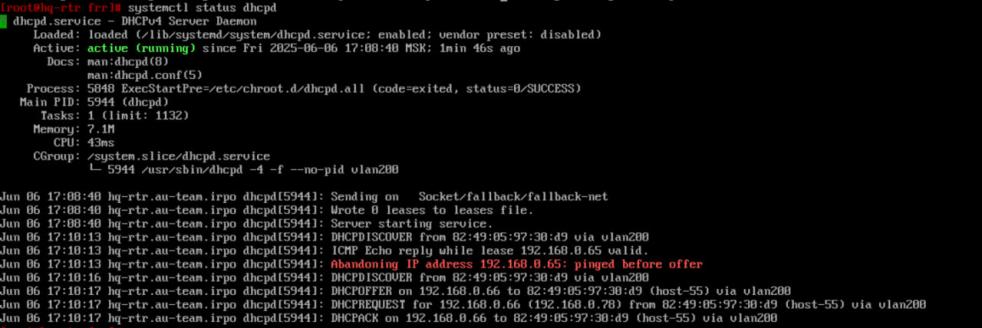


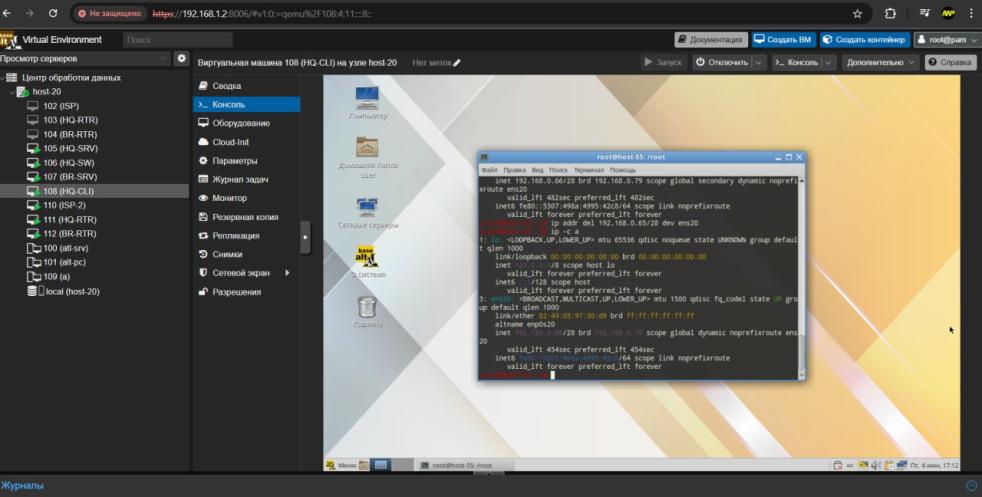
Переходим в nano /etc/dhcp/dhcpd.conf

***systemctl enable --now dhcpd***

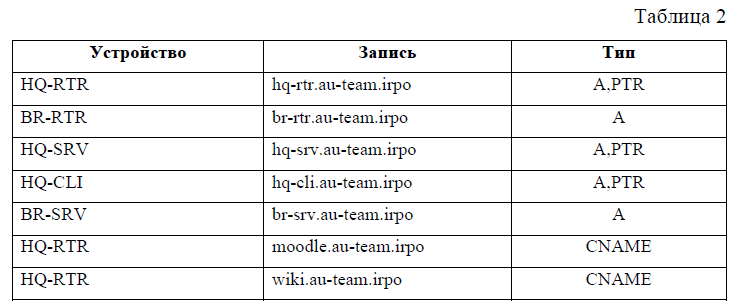




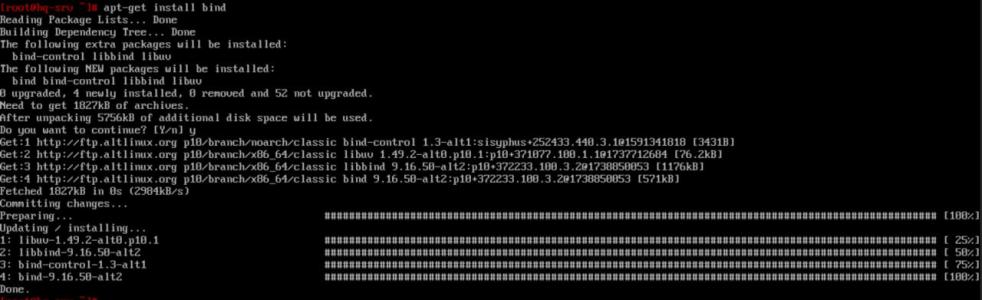




**Настройка DNS для офисов HQ и BR**

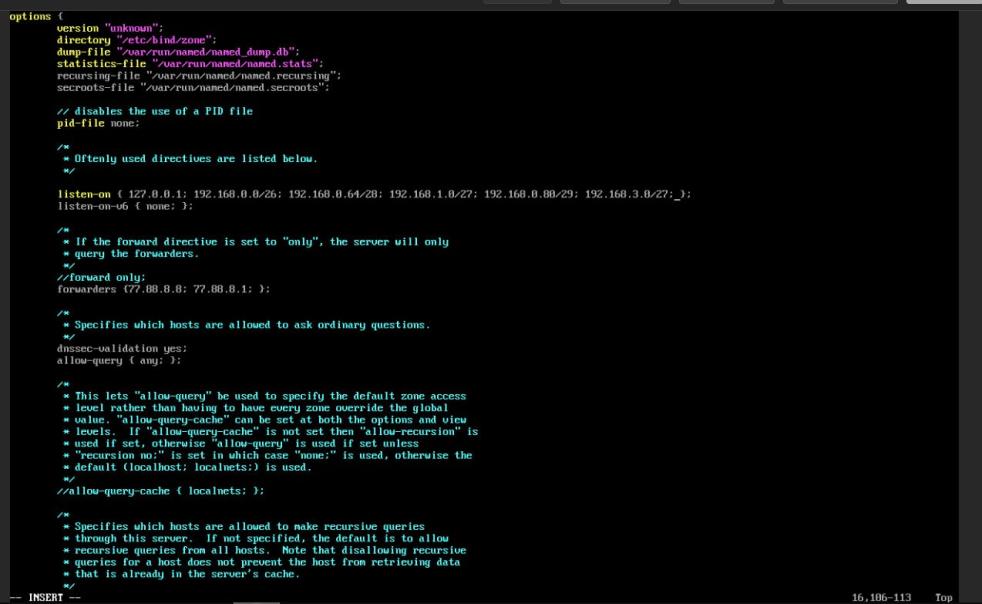


Устанавливаем bind



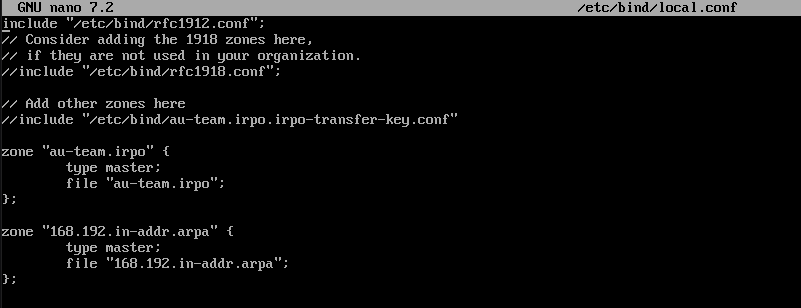
**Редактируем конфигурационный файл /etc/bind/options.conf**

nano  /etc/bind/options.conf



Файлы зон

Переходим по пути nano /etc/bind/local.conf



## **Создание локальных зон DNS**

**Зона прямого просмотра**

Для сокращения времени написания файла прямой зоны (au-team.irpo) скопируем шаблон и отредактируем его

# cp /etc/bind/zone/localdomen  /etc/bind/zone/au-team.irpo





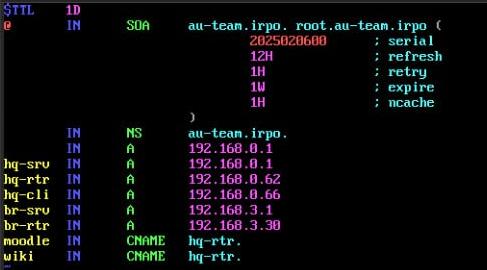
Открываем на редактирование файл зоны au-team.irpo

vim /etc/bind/zone/au-team.irpo

Для редактирования нажать букву латинскую A

И пишем

А сохраняем shift+z+z



**Зона обратного просмотра**

Для сокращения времени написания файла обратной зоны (0.168.192.in-addr.arpa) скопируем шаблон и отредактируем его

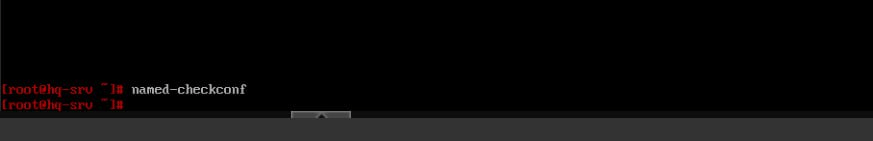
***# cp /etc/bind/zone/127.in-addr.arpa /etc/bind/zone/.168.192.in-addr.arpa***

Открываем на редактирование файл зоны *168.192.in-addr.arpa и  3.168.192.in-addr.arpa*

Через vim

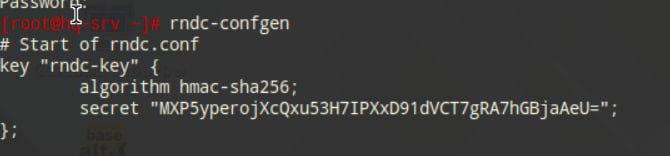




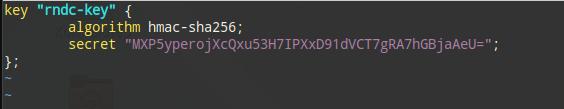


ЕСЛИ ВЫЛЕЗЛА ОШИБКА RNDC.KEY

НУЖНО СГЕНЕРИРОВАТЬ НОВЫЙ



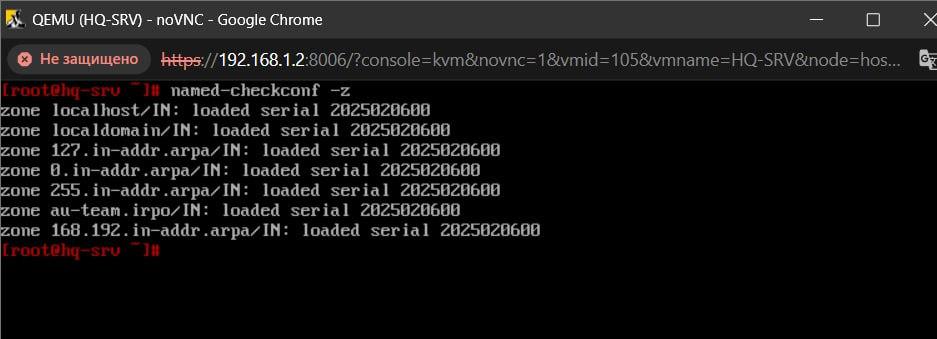
И ВСТАЛЯЕМ В ФАЙЛ НОВЫЙ КЛЮЧ

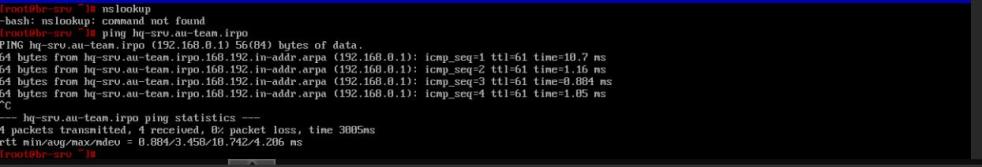


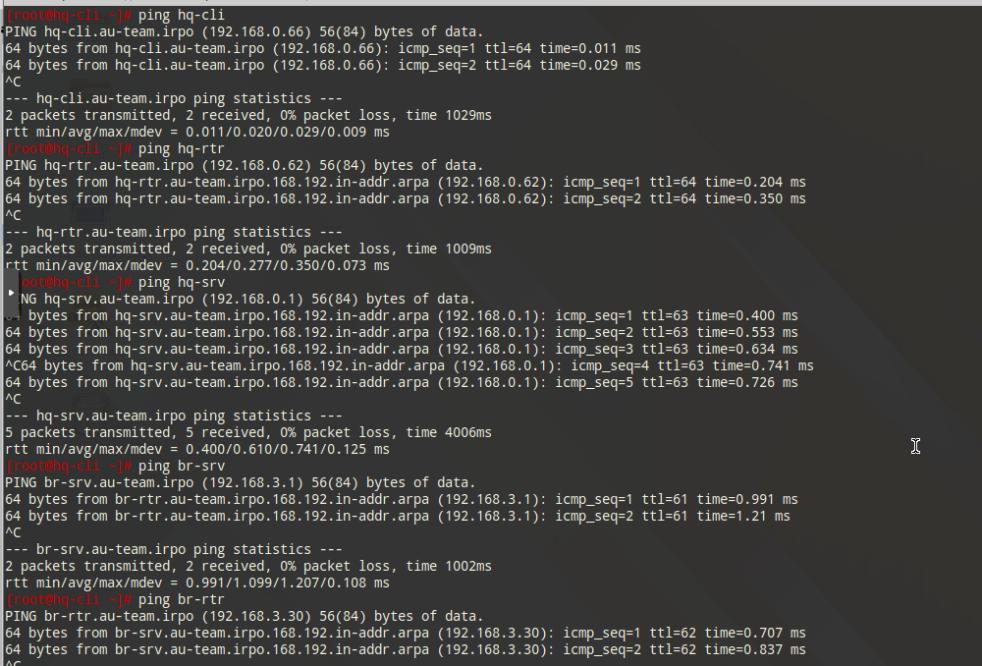
Задаем права

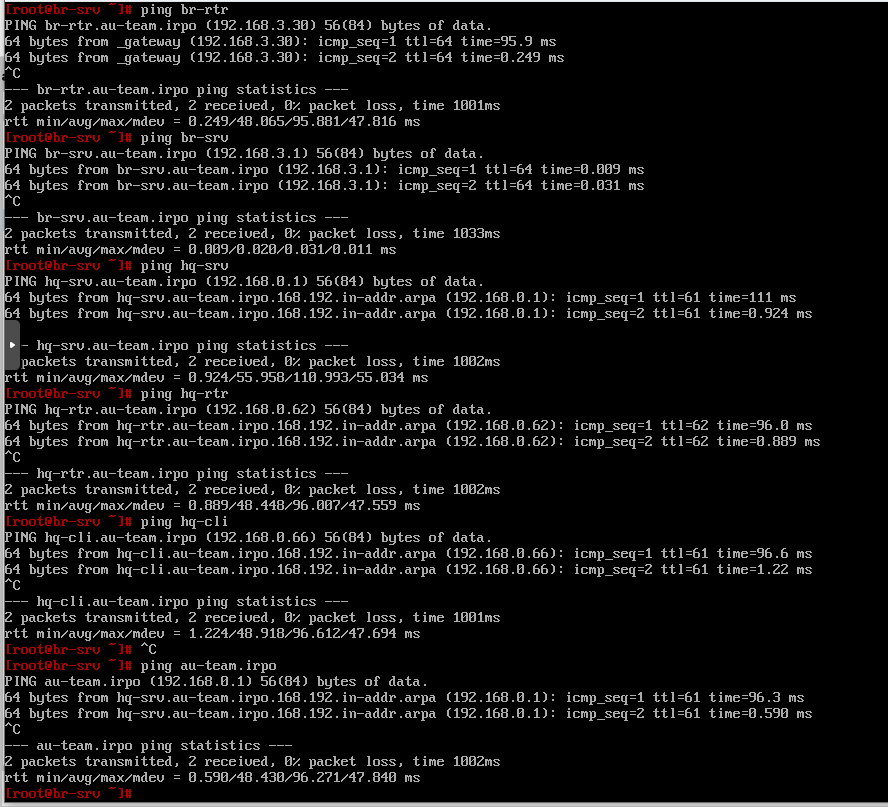


И проверяем зоны

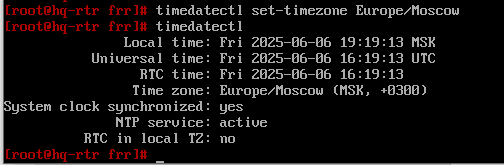


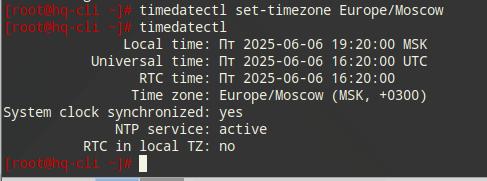


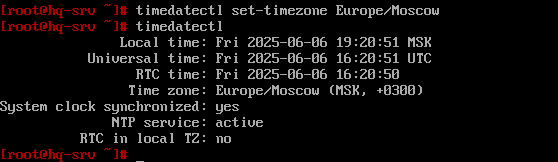


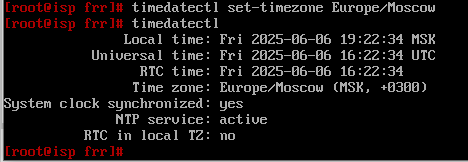


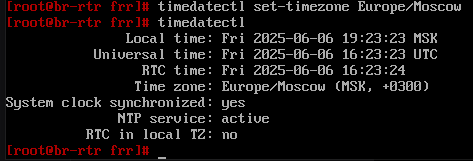
**Настроим Московский часовой пояс (UTC +3):**

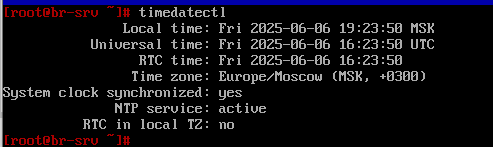












***2 модуль***

**Настройте доменный контроллер Samba на машине BR-SRV:**

* Создайте 5 пользователей для офиса HQ (формат: user№.hq).
* Создайте группу hq и добавьте туда созданных пользователей.
* Включите в домен машину HQ-CLI.
* Настройте, чтобы пользователи группы hq могли:
  + Аутентифицироваться на клиентском ПК.
  + Повышать привилегии только для выполнения команд cat, grep, id.
  + Не могли запускать другие команды с повышенными привилегиями.
* Импортируйте пользователей из файла /opt/users.csv.

• **Сконфигурируйте файловое хранилище:**

* Настройте RAID5 из трёх 1ГБ-дисков на HQ-SRV.
* Устройство: md0, конфигурация в /etc/mdadm.conf.
* Обеспечьте автоматическое монтирование в /raid5.
* Создайте раздел и отформатируйте в ext4.
* Настройте NFS-доступ в /raid5/nfs с правами чтения/записи для HQ-CLI.
* На HQ-CLI настройте автомонтирование NFS в /mnt/nfs.

• **Настройте службу сетевого времени:**

* HQ-RTR как сервер chrony со стратумом 5.
* HQ-SRV, HQ-CLI, BR-RTR, BR-SRV как клиенты.

• **Сконфигурируйте Ansible на BR-SRV:**

* Создайте инвентарь с HQ-SRV, HQ-CLI, HQ-RTR и BR-RTR.
* Рабочий каталог Ansible: /etc/ansible.
* Настройте машины для ответа pong на ping без ошибок.

• **Развертывание приложений в Docker на BR-SRV:**

* Создайте файл wiki.yml для MediaWiki.
* Настройте два контейнера:
  + **wiki**: образ mediawiki, порт 8080, монтируйте LocalSettings.php.
  + **mariadb**: образ mariadb, база mediawiki, пользователь wiki, пароль WikiP@ssw0rd.
* Убедитесь, что MediaWiki доступна через порт 8080.

• **Сконфигурируйте статическую трансляцию портов:**

* На HQ-RTR: порт 2024 на HQ-SRV.
* На BR-RTR:
  + Порт 80 в 8080 для MediaWiki.
  + Порт 2024 в 2024 на BR-SRV.

• **Запустите сервис Moodle на HQ-SRV:**

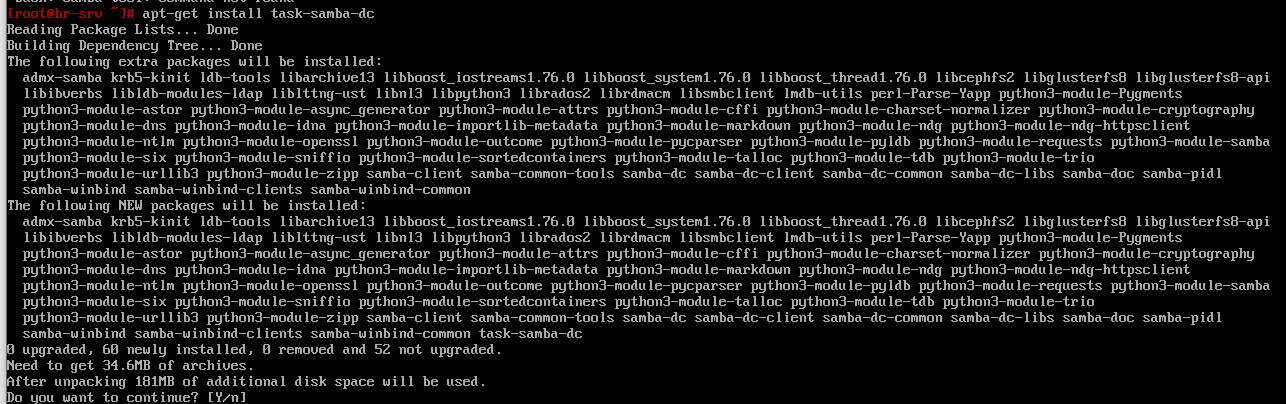
* Используйте Apache и MariaDB.
* Создайте базу moodledb.
* Пользователь: moodle, пароль: P@ssw0rd.
* Админ Moodle: пароль P@ssw0rd.
* Отобразите номер рабочего места на главной странице.

**Установите Яндекс.Браузер для организаций на HQ-CLI:**

Установите удобным способом.

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

**Устанавливаем samba**

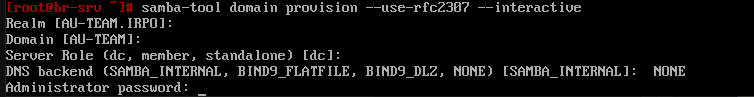


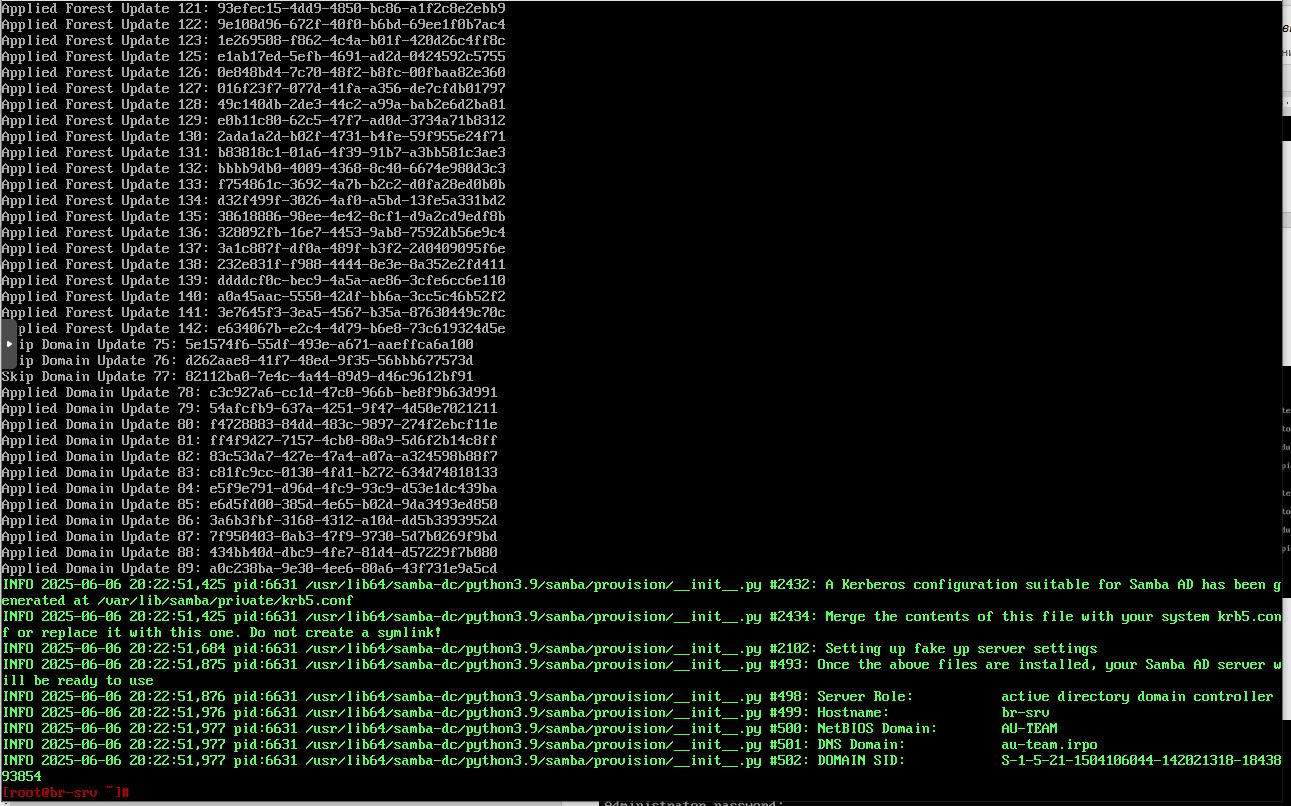
Удаляем smb.conf и другие конфигурационные файлы

rf -f /etc/samba/smb.conf



Запускаем автоматическую настройку





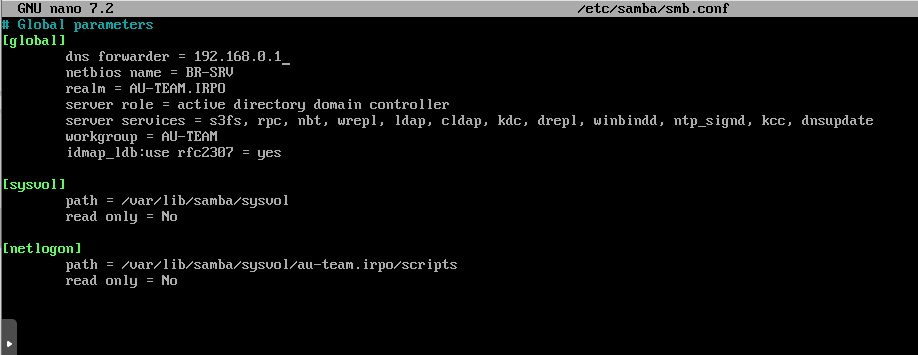
Настраиваем Kerberos

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



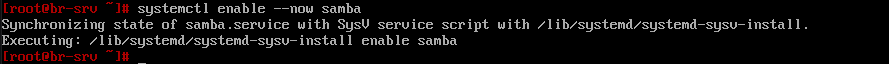
Добавляем адрес dns сервера в smb.conf

nano /etc/samba/smb.conf



Запускаем samba с добавлением в автозагрузку

systemctl enable --now samba



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"

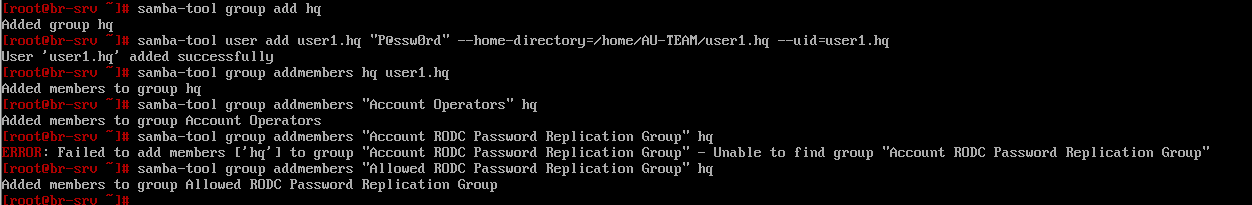
--home-directory=/home/AU-TEAM/user1.hq --uid=user1.hq

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



перезагружаем samba

systemctl restart samba

Внимание рекомендую перезагрузить сервер!!!

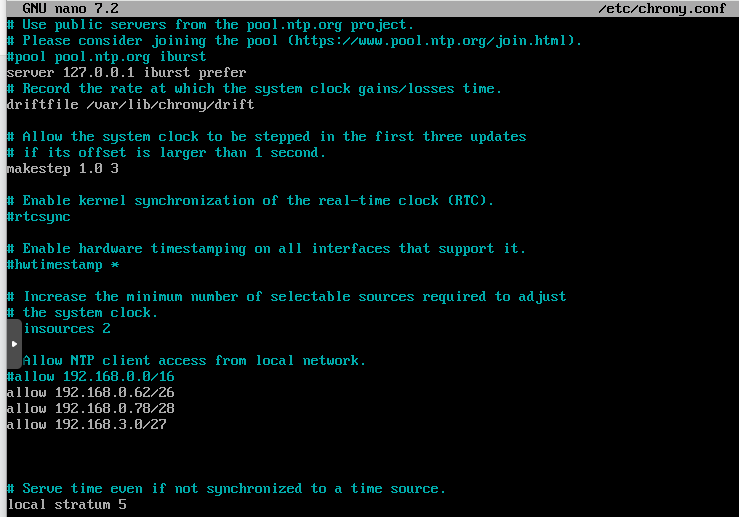
Reboot

Вводим команду подключения к домену



вводим пароль: P@ssw0rd (от пользователя)

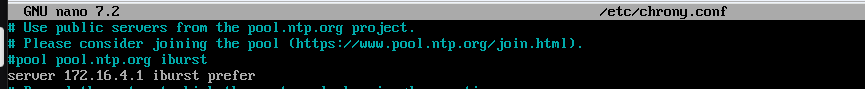
**Настройте синхронизацию времени между сетевыми устройствами по протоколу NTP**



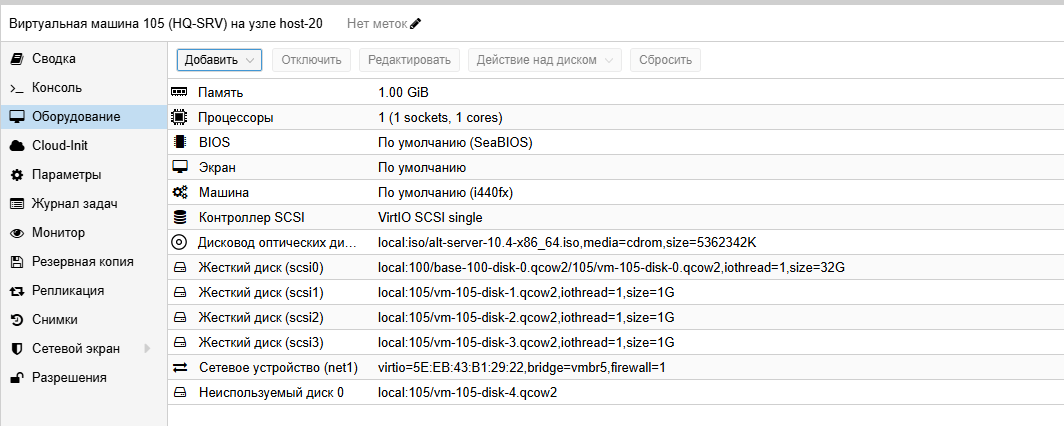
Запускаем службу синхронизации

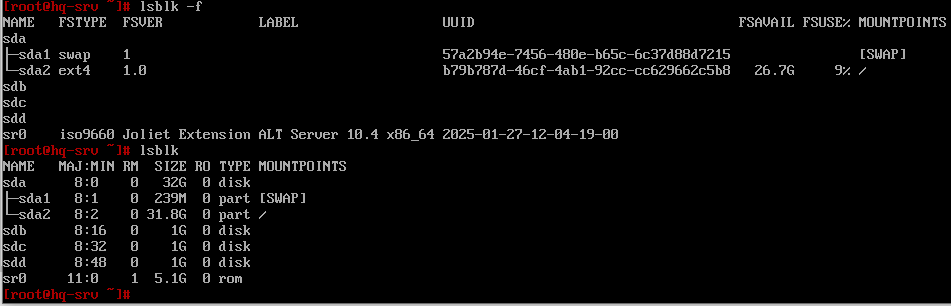
systemctl enable –now chronyd

На клиентах



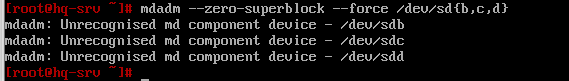
**Сконфигурируйте файловое хранилище**

****

****

Обнуляем суперблоки для добавленных дисков:

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

****

**Удаляем старые метаданные и подпись на дисках**

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

****

Создаем **RAID**:

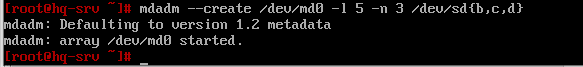
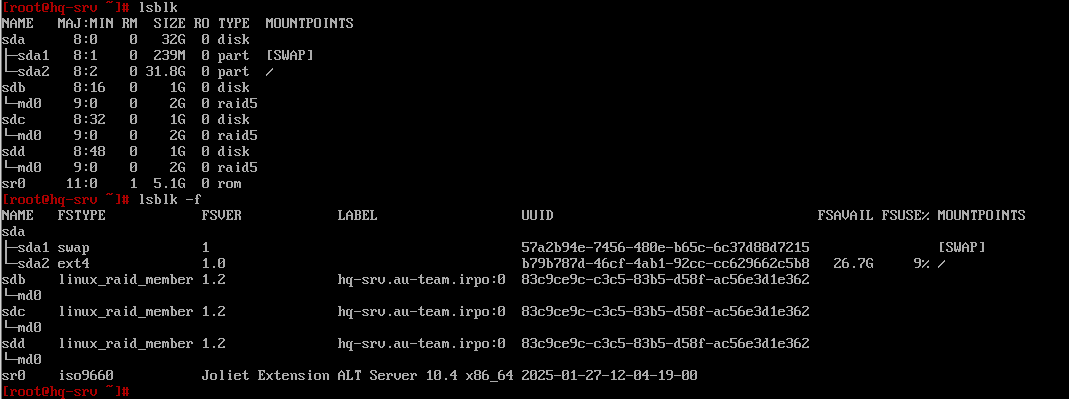
mdadm --create /dev/md0 -l 5 -n 3 /dev/sd{b,c,d}

**/dev/md0** - название RAID после сборки

**-l 5** - уровень RAID

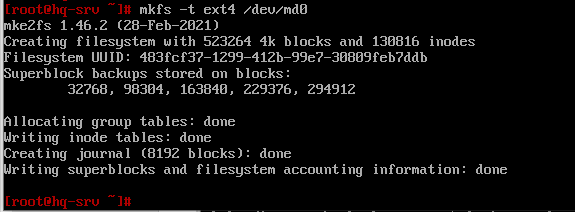
**-n 3** - количество дисков, из которых собирается массив

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

**** ****

Создаем файловую систему из созданного **RAID**:

mkfs -t ext4 /dev/md0

****

**Создание файла mdadm.conf**

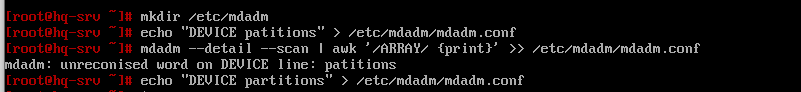
Создаем директорию для файла:

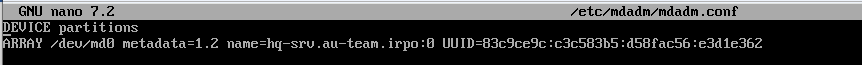
mkdir /etc/mdadm

Заполняем файл информацией:

echo "DEVICE partitions" > /etc/mdadm/mdadm.conf

mdadm --detail --scan | awk '/ARRAY/ {print}' >> /etc/mdadm/mdadm.conf

****

****

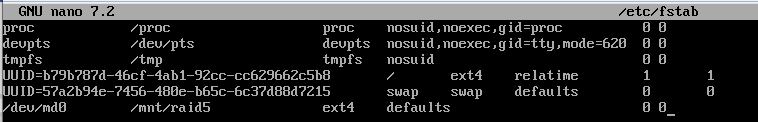
**Создание файловой системы и монтирование RAID-массива**

Создаем директорию для монтирования массива:

mkdir /mnt/raid5

Добавляем строку в **/etc/fstab**:

/dev/md0 /mnt/raid5 ext4 defaults 0 0

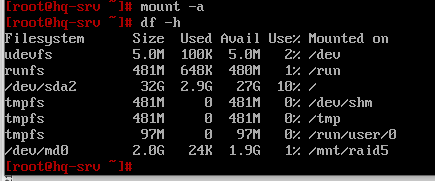
****

Монтируем:

mount -a

Проверяем монтирование:

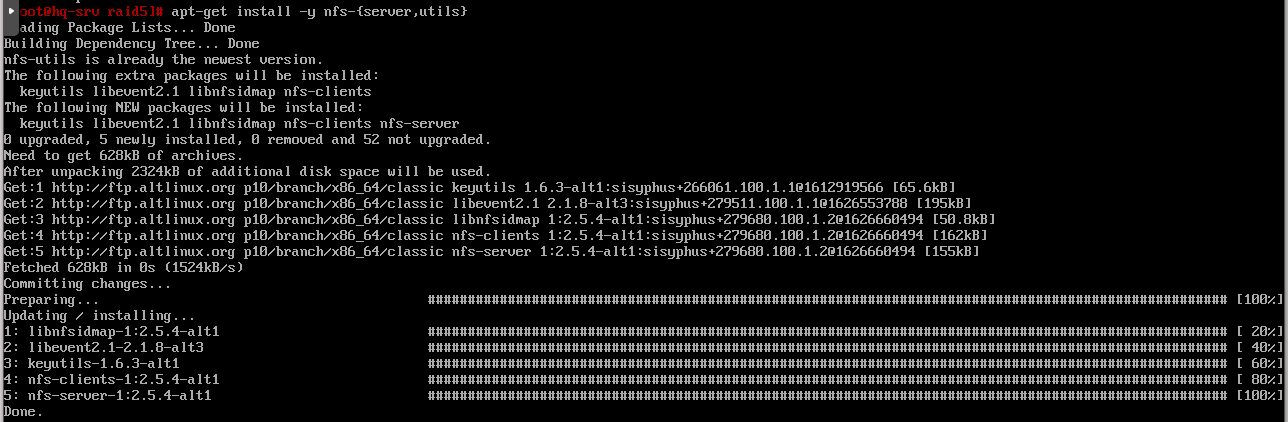
df -h

****

* Настройте NFS-доступ в /raid5/nfs с правами чтения/записи для HQ-CLI.
* На HQ-CLI настройте автомонтирование NFS в /mnt/nfs.

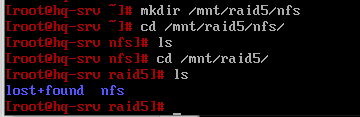
Устанавливаем пакеты для **NFS-сервера**:

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



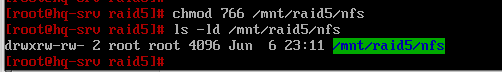
Создаем директорию для общего доступа:

mkdir /mnt/raid5/nfs



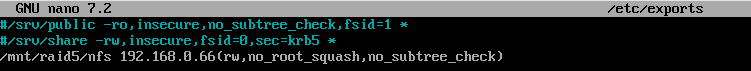
Выдаем права на чтение и запись этой директории:

chmod 766 /mnt/raid5/nfs



Добавляем строку в **/etc/exports**:

/mnt/raid5/nfs 192.168.0.66(rw,no\_root\_squash,no\_subtree\_check)



Экспортируем файловую систему, которую прописали ранее:

exportfs -ra

**-a** - экспортировать все указанные каталоги

**-r** - повторный экспорт всех каталогов, синхронизируя **/var/lib/nfs/etab** с **/etc/exports** и файлами в **/etc/exports.d**

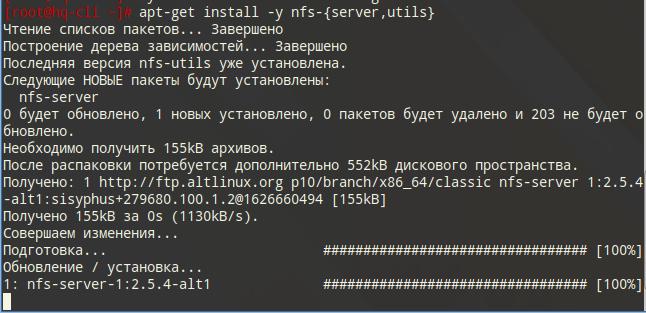
**-v** - подробный вывод

Запускаем и добавляем в автозагрузку **NFS-сервер**:

systemctl enable --now nfs-server



**Настройк клиента**

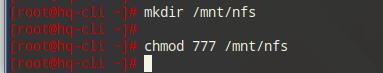


Создаем директорию для общего ресурса:

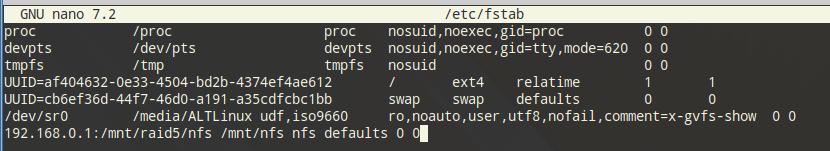
mkdir /mnt/nfs

Выдаем права этой директории:

chmod 777 /mnt/nfs

****

Добавляем строку в /etc/fstab для авто монтирования общего ресурса

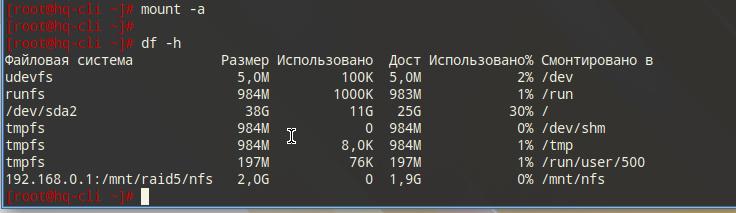
****

Монтируем общий ресурс:

mount -a

Проверяем монтирование:

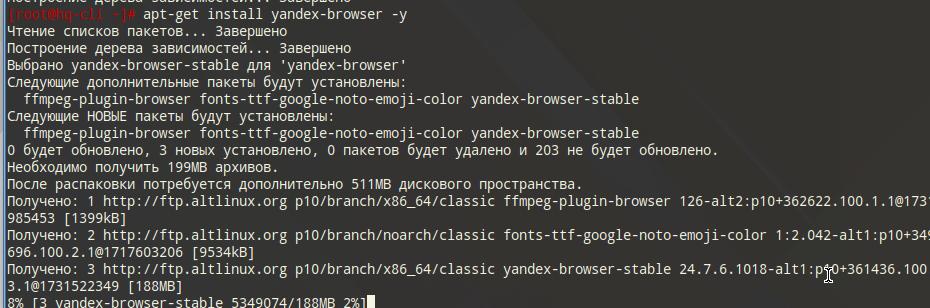
df -h

****

**Установить яндекс браузер на CLI**

Устанавливаем яндекс браузер:

apt-get install yandex-browser -y

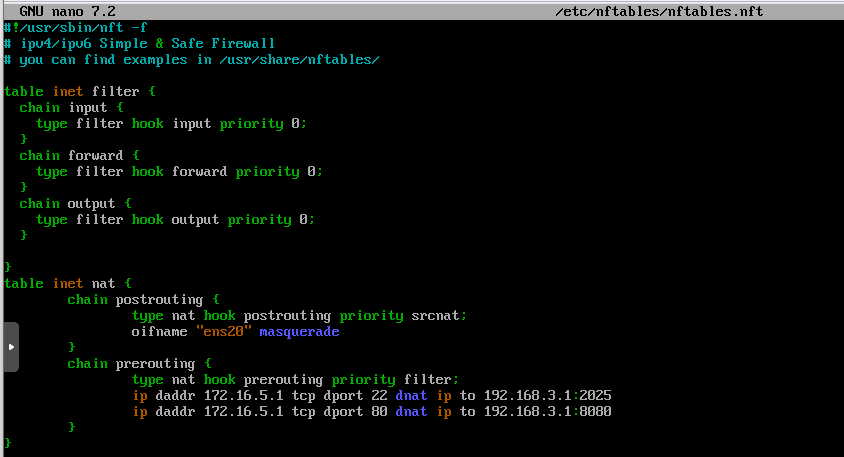


****

**На маршрутизаторах сконфигурируйте статическую трансляцию портов**

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

добавляя chain prerouting :

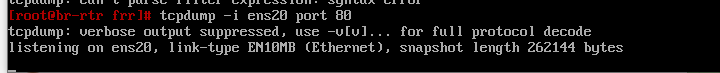
****

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

добавляя chain prerouting :

chain prerouting {

****

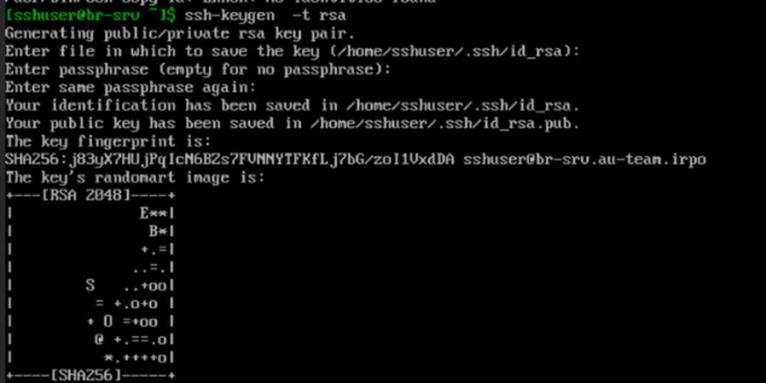
****

**Сконфигурируйте Ansible на BR-SRV:**

* Создайте инвентарь с HQ-SRV, HQ-CLI, HQ-RTR и BR-RTR.
* Рабочий каталог Ansible: /etc/ansible.
* Настройте машины для ответа pong на ping без ошибок.

**Создание пары SSH-ключей.**

**На br-srv генерируем ключи**



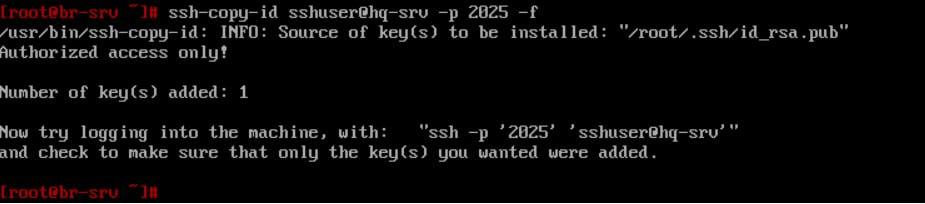
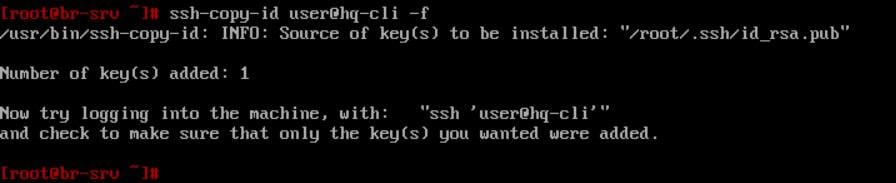
**Копируем их на устройства hq-r, hq-srv, hq-cli, br-r**

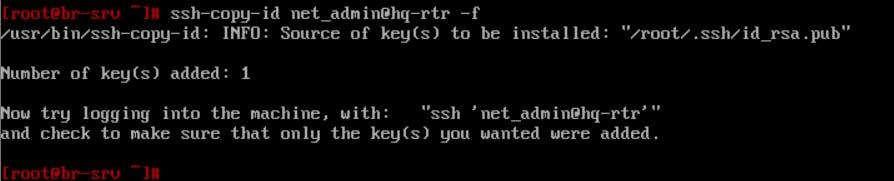
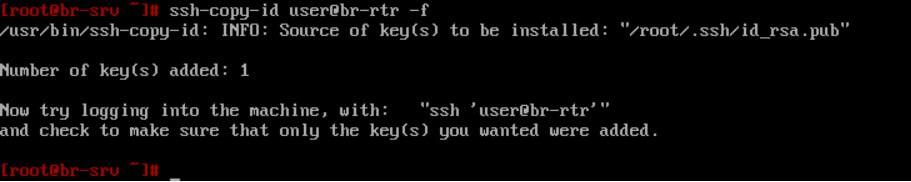
**Копируем открытый SSH-ключ на удаленные устройства**

**под пользователем sshuser на hq-srv**

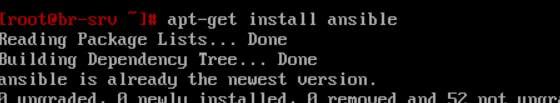
**под user на hq-cli**

**под net\_admin на роутерах**

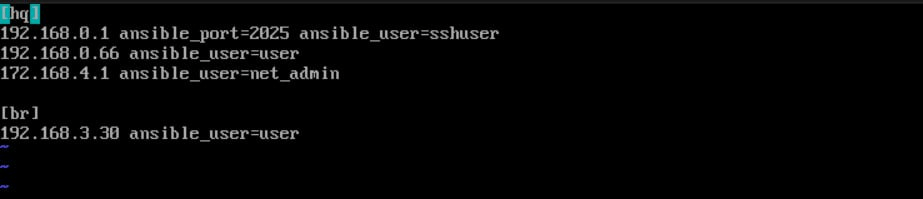


**На br-srv устанавливаем ansible**



**Подготовка файла инвентаря (hosts)**





**Запуск команд с пользовательским инвентарем**

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



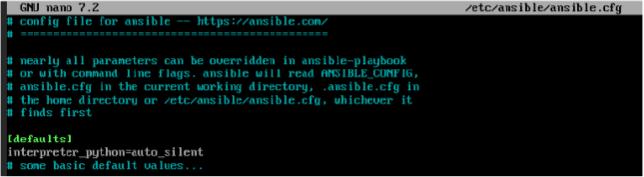
**Для управления поведением обнаружения в глобальном масштабе необходимо в**

**файле конфигурации ansible /etc/ansible/ansible.cfg в разделе [defaults] прописать**

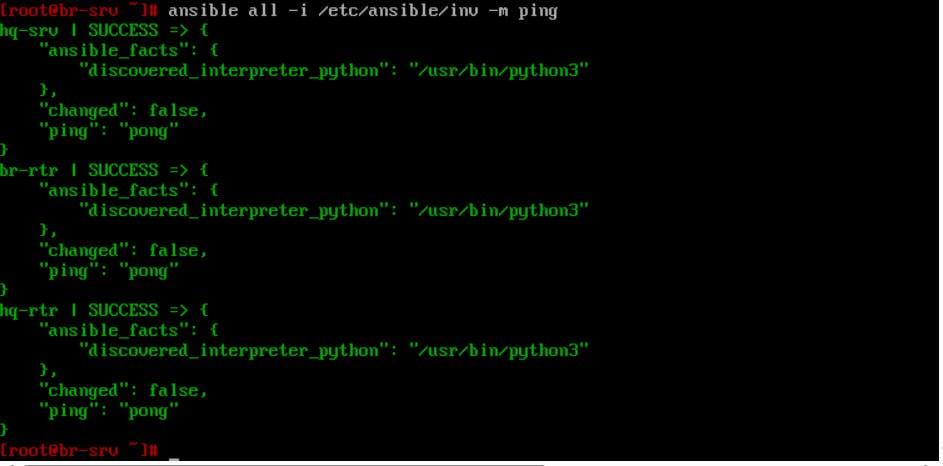
**ключ interpreter\_python с параметром auto\_silent. В большинстве дистрибутивов**

**прописываем вручную.**

**# nano /etc/ansible/ansible.cfg**

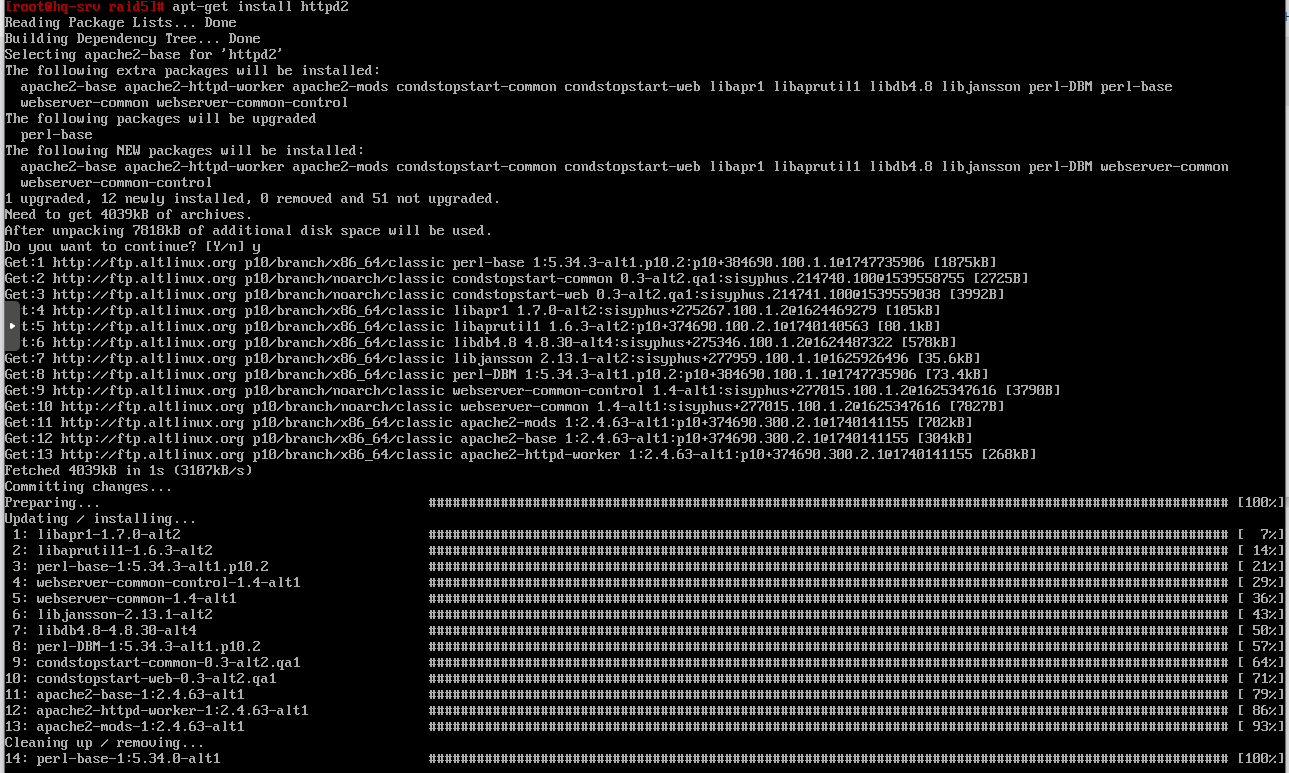
****

**Проверяем**

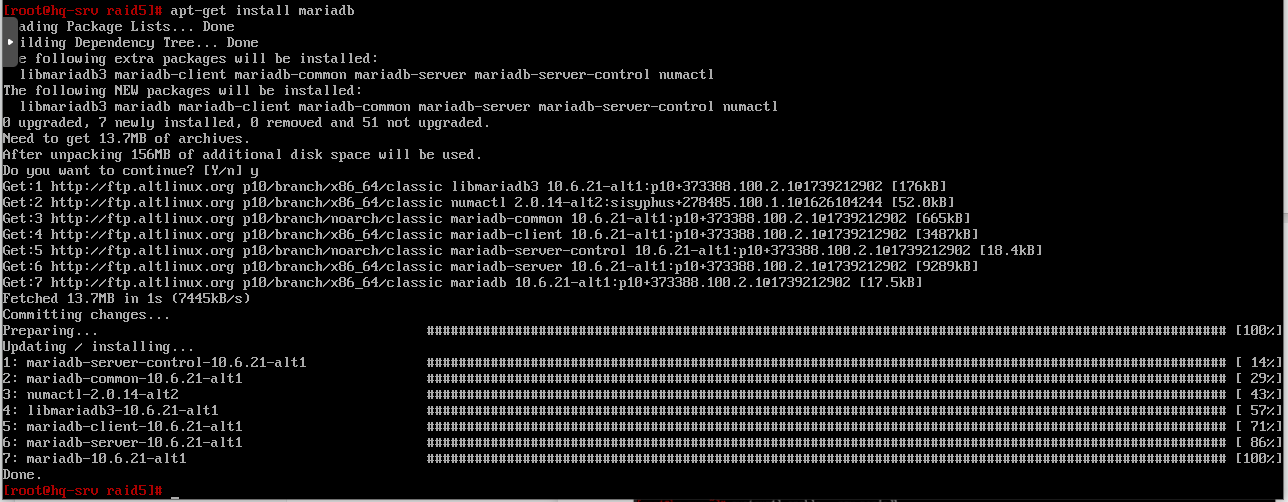


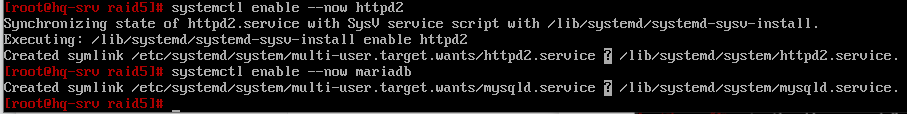
**Запустите сервис moodle на сервере HQ-SRV**

**Устанавливаем httpd2**

****

**Устанавливаем mariadb**

****

****

**Подключаем модуль**

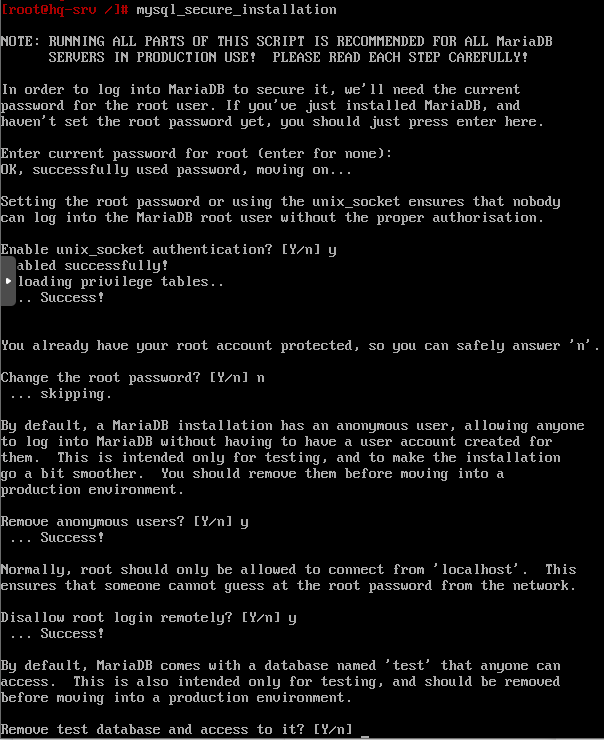
**mysql\_secure\_installation (tab на этом пункте работает)**

**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";**

**mysql -u root -p**

**вводим пароль от root**

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

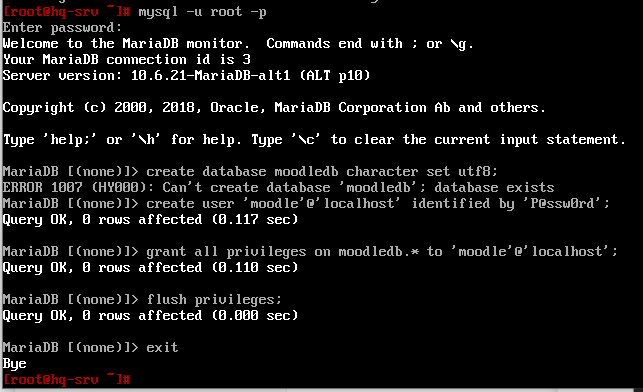
CREATE DATABASE moodledb CHARACTER SET utf8;

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

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

FLUSH PRIVILEGES;

EXIT;

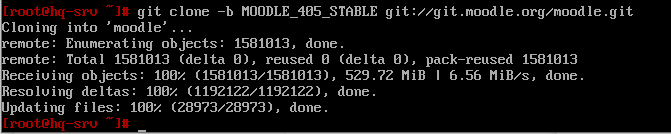
****

Загружаем код проекта Moodle:

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

ЕСЛИ ПИШЕТ ЧТО НЕТ КОМАНДЫ

СКАЧАЙТЕ – apt-get install git



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

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

****

**Создаём необходимую структуру каталагов для корректной установки и работы Moodle:**

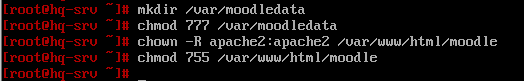
**mkdir /var/moodledata**

**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



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:

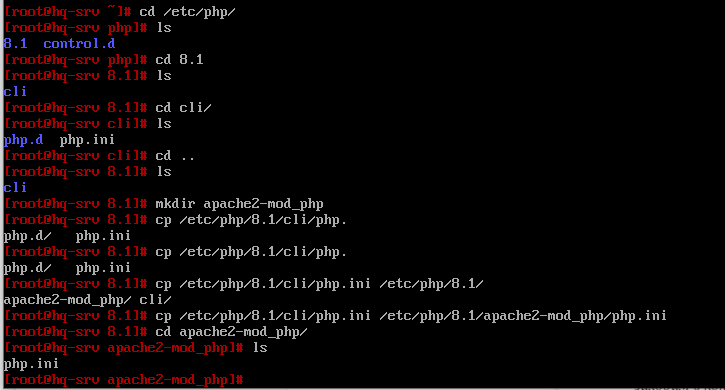
apachectl configtest



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

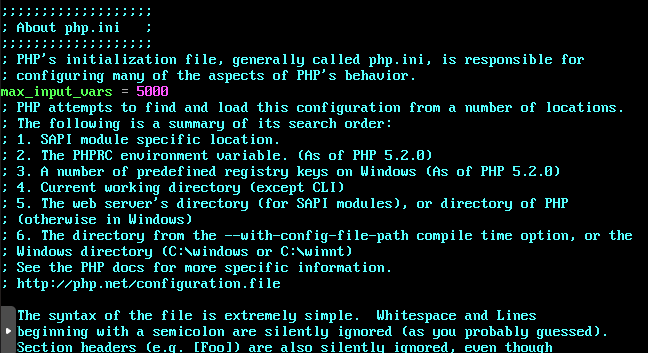
**заходим в конфиг**

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

****

**Через vim**

**Просто вписали**

****

**МУДЛ НЕВЫШЛО**

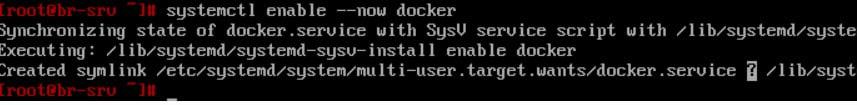
• **Развертывание приложений в Docker на BR-SRV:**

* Создайте файл wiki.yml для MediaWiki.
* Настройте два контейнера:
  + **wiki**: образ mediawiki, порт 8080, монтируйте LocalSettings.php.
  + **mariadb**: образ mariadb, база mediawiki, пользователь wiki, пароль WikiP@ssw0rd.
* Убедитесь, что MediaWiki доступна через порт 8080.

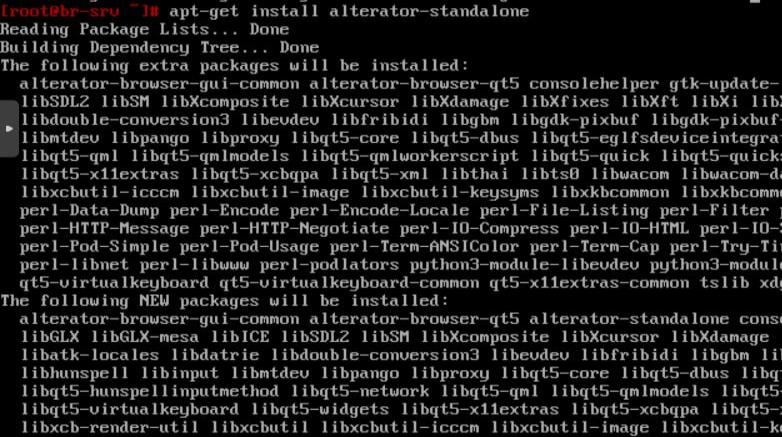
Устанавливаем докер

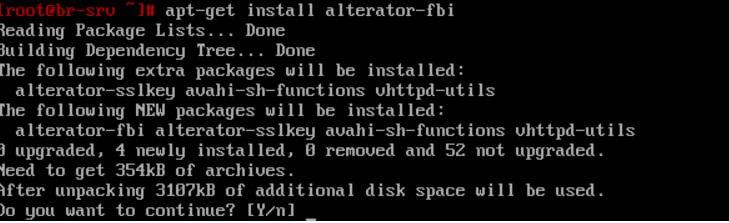


Добавляем в автозагрузку

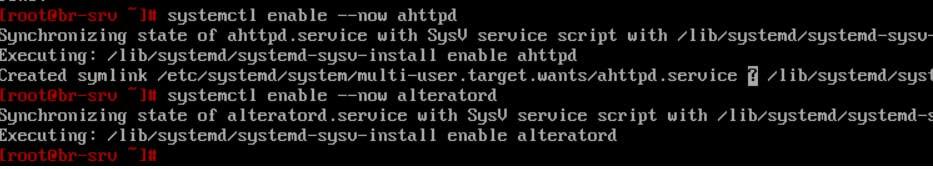


Скачиваем alterator



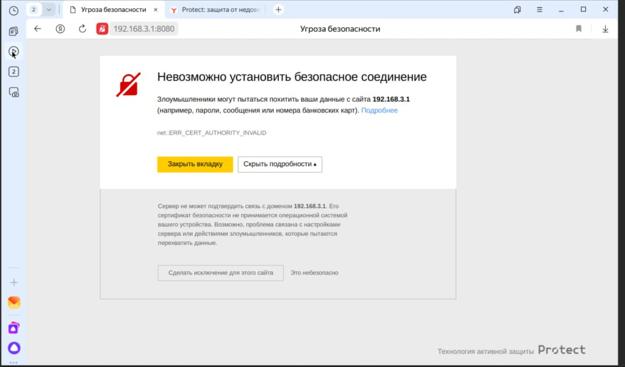


Добавляем alteratorhttpd

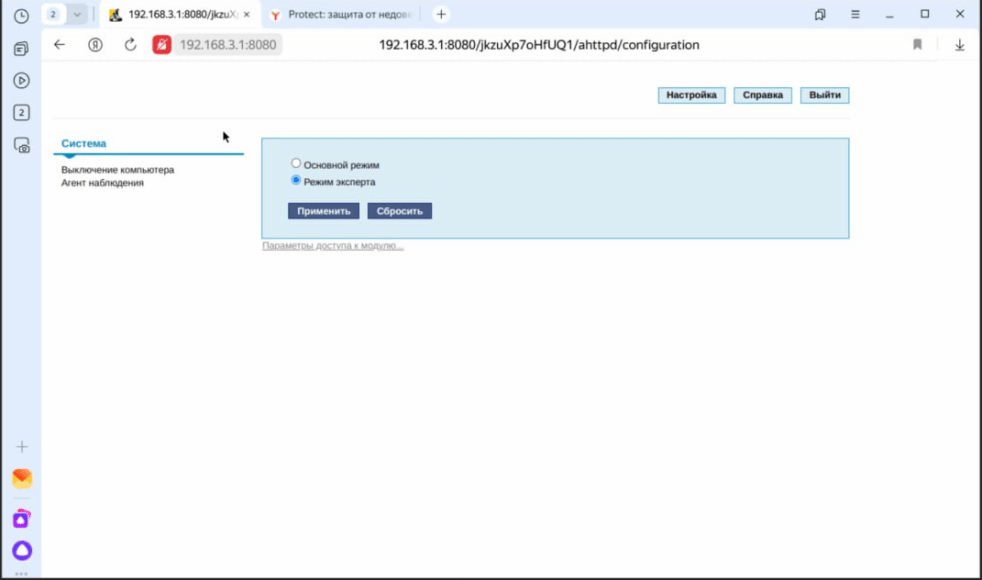


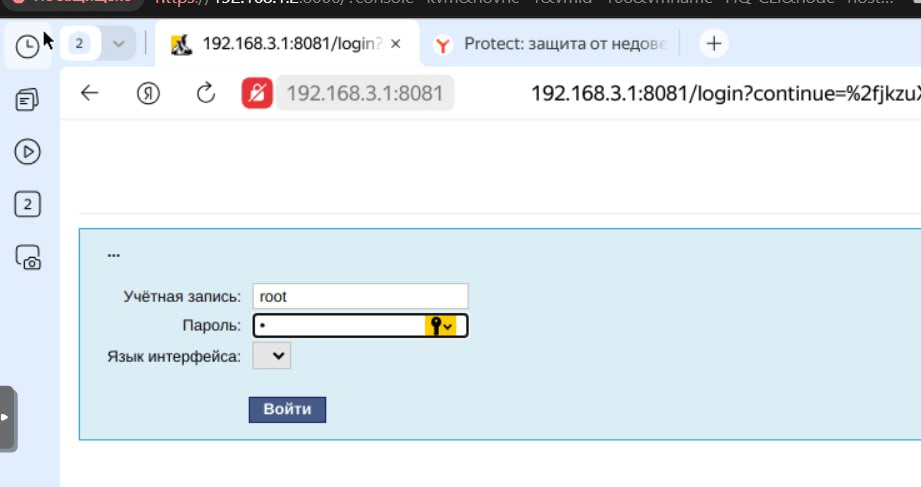
На CLI открываем браузер в водим адрес https://192.168.3.1 вводим пароль от

root и заходим в ЦУС

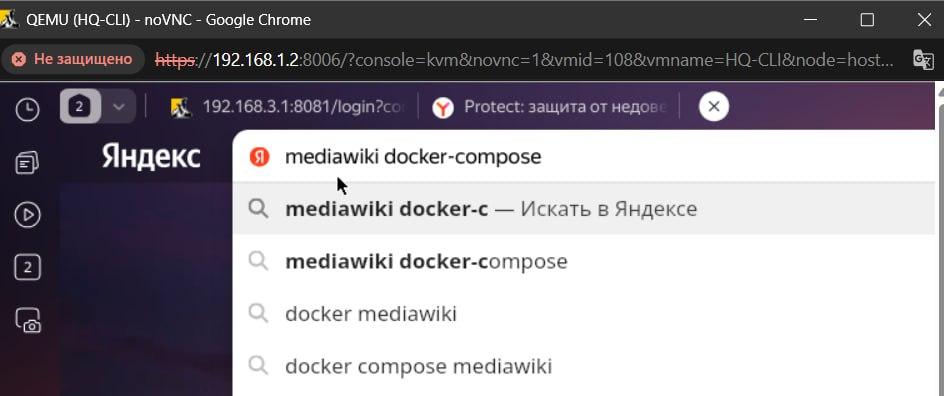


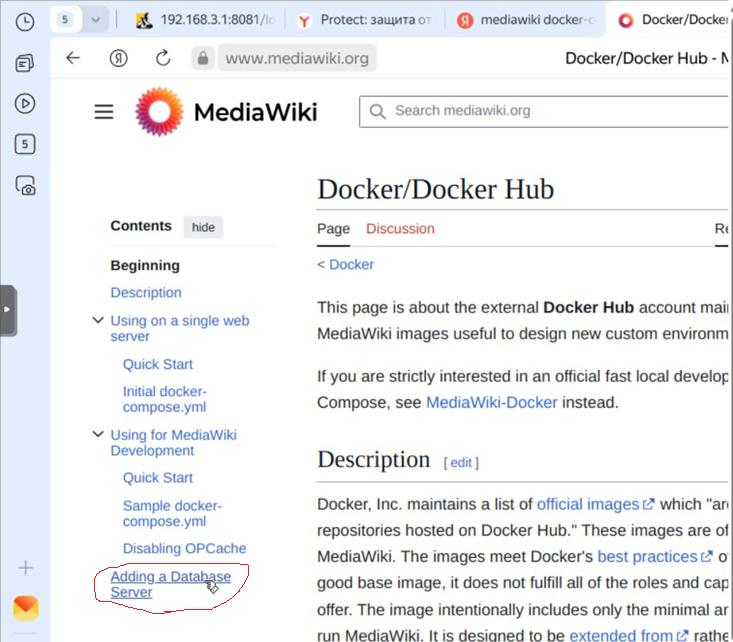
Выбираем режим эксперта

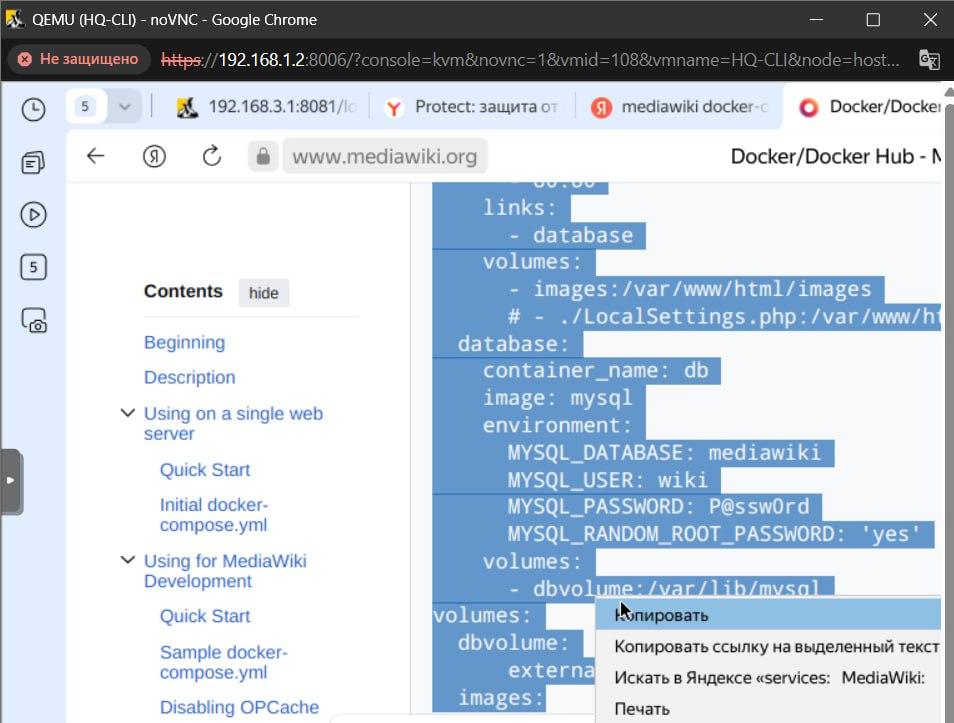




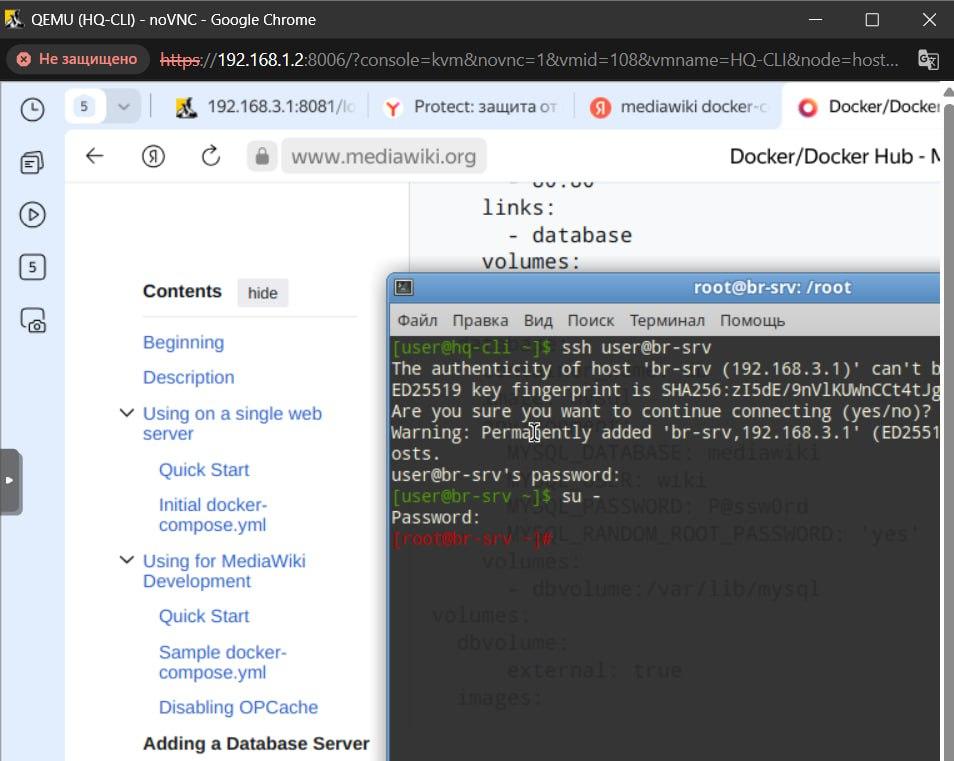
Ищем mediawiki docker



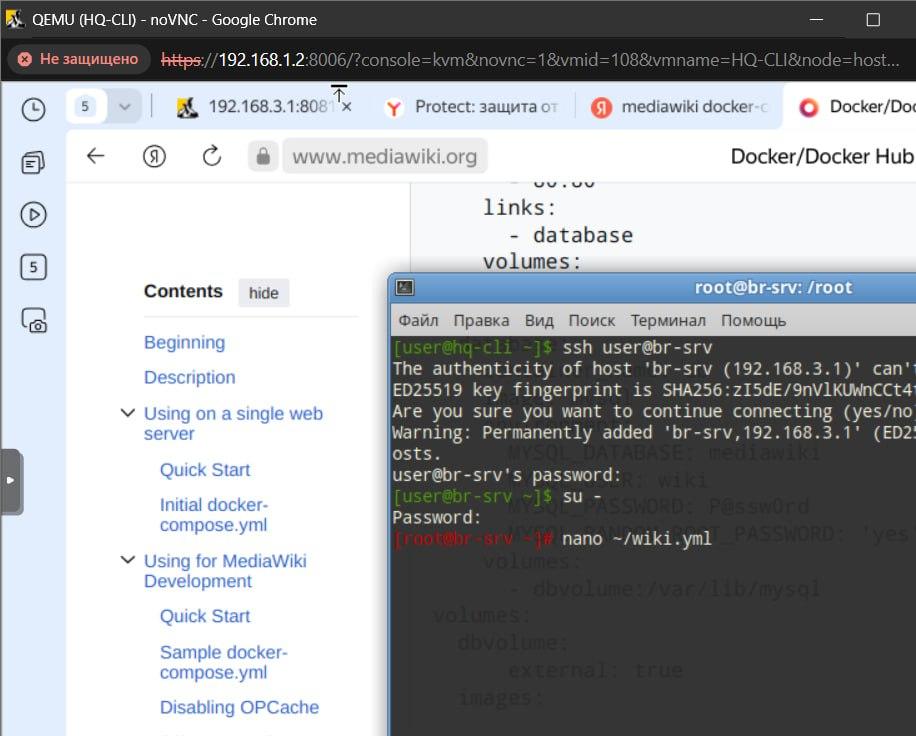


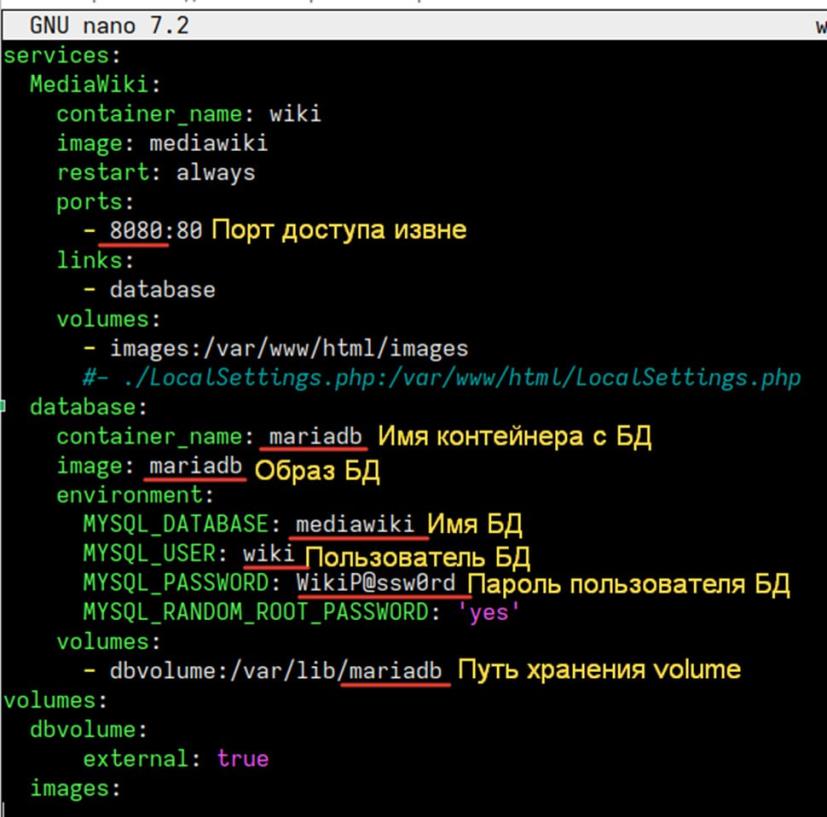


Подключаемся по ssh на br-srv



Открываем nano ~/wiki.yml





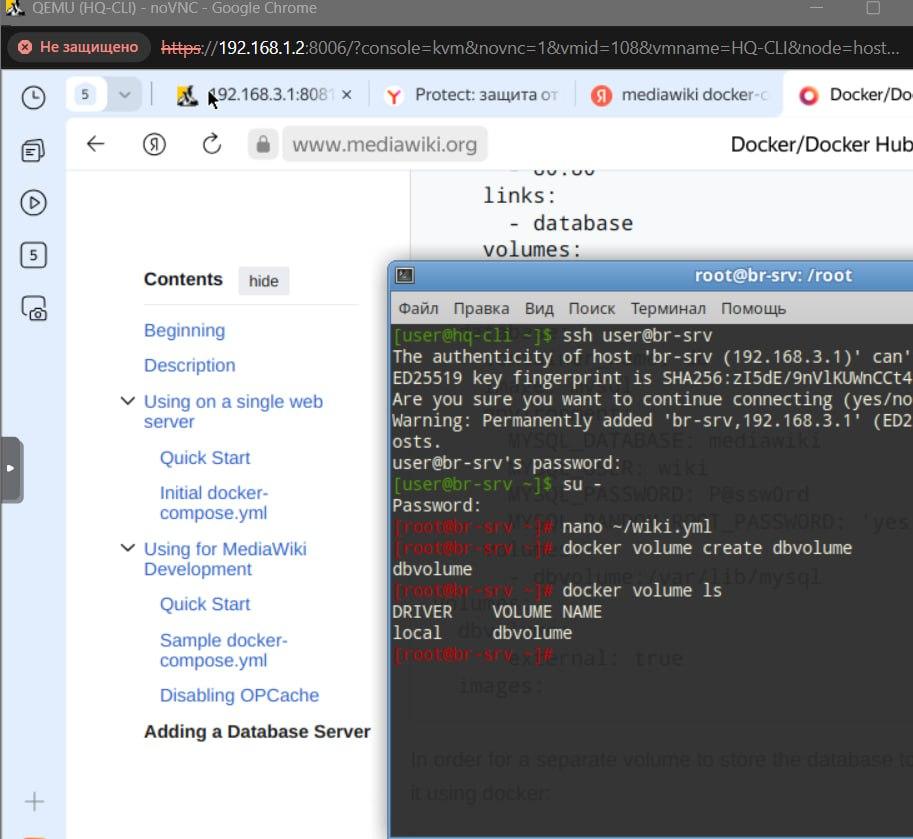
Чтобы отдельный volume для хранения базы данных имел правильное имя -

создаём его средствами docker:

docker volume create dbvolume

Посмотреть все имеющиеся volume можно командой

docker volume ls (необезательно)



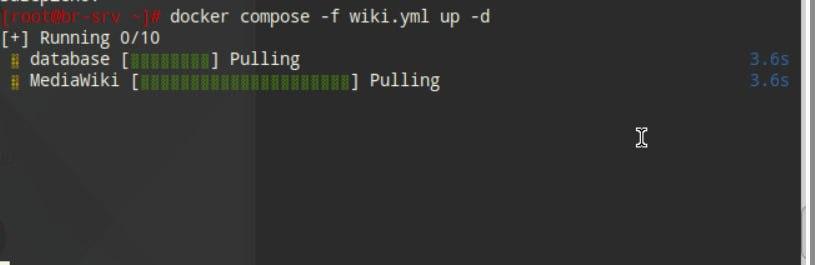
Скачиваем docker-copmose



Выполняем запуск стека контейнеров с приложением MediaWiki и базой

данных описанных в файле wiki.yml:

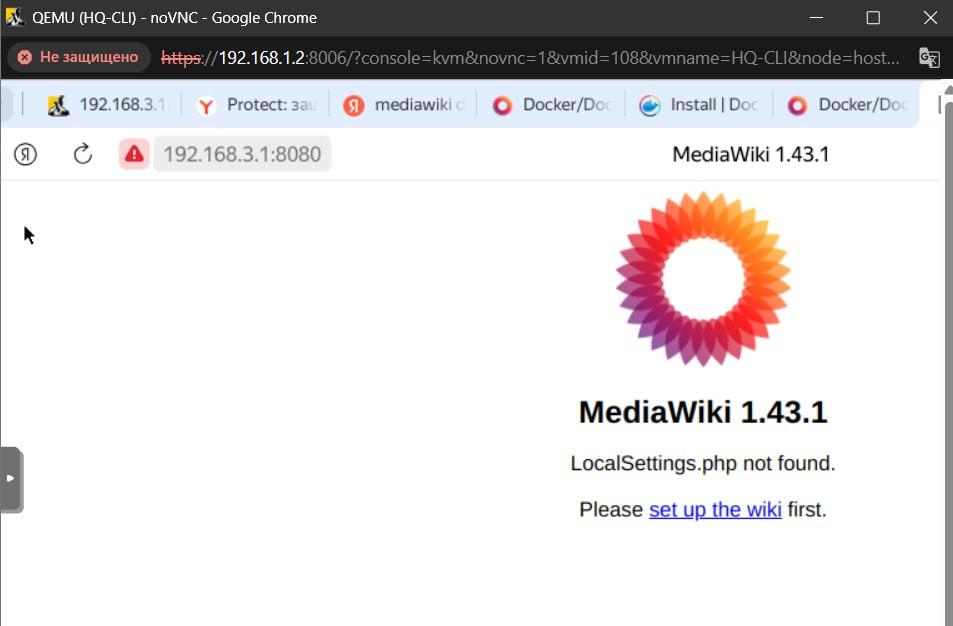
docker compose -f wiki.yml up -d



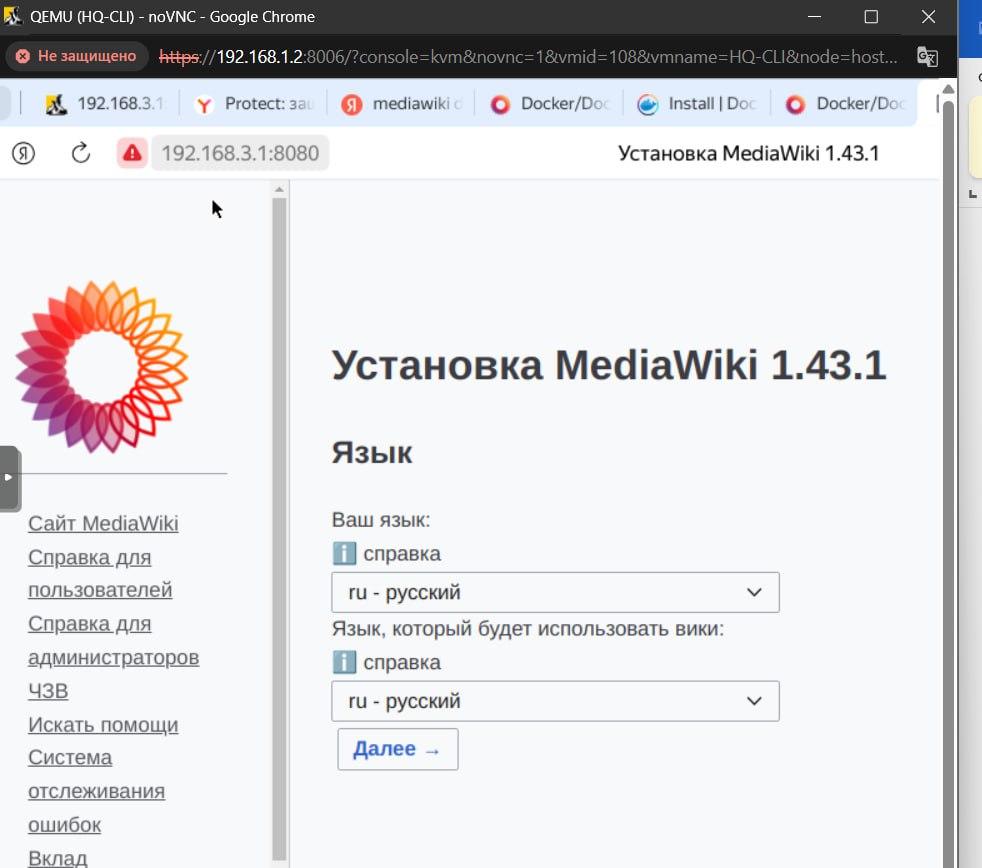
Открываем браузер по адресу нашего сервера http://192.168.3.1:8080

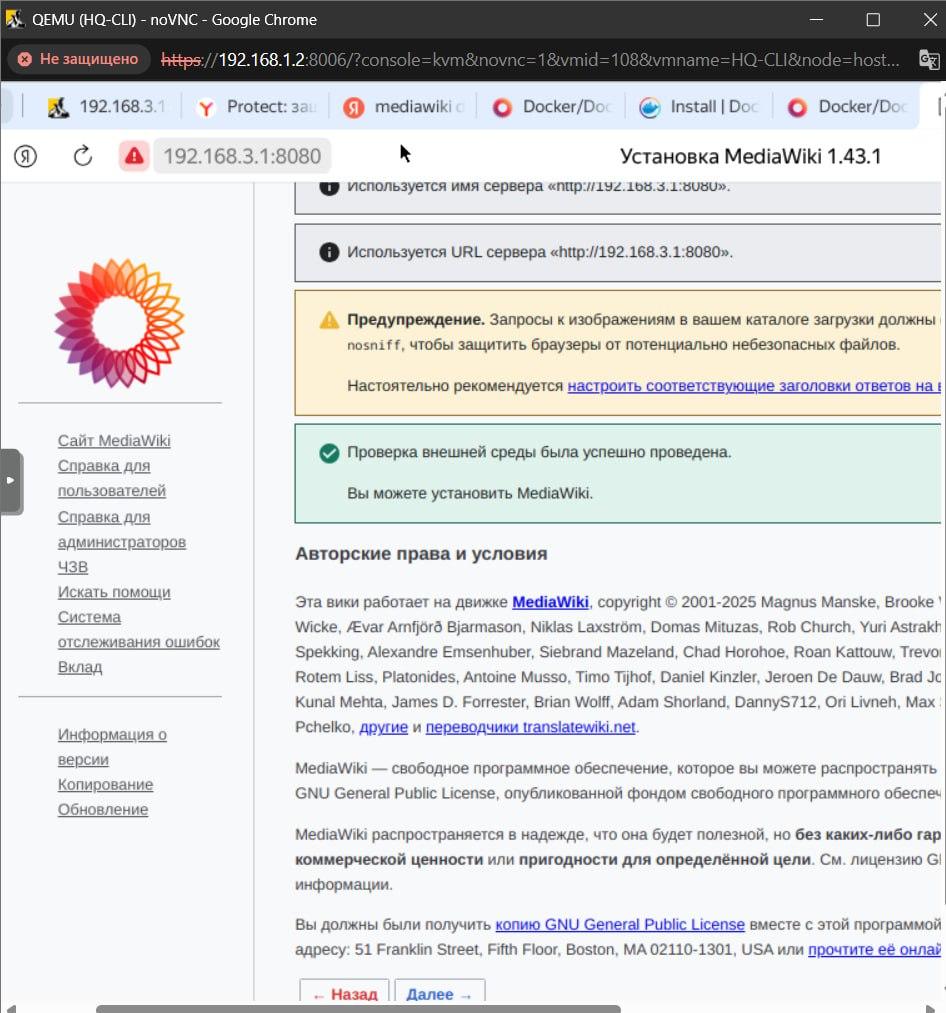
или http://wiki.au-team.irpo если настроен nginx.

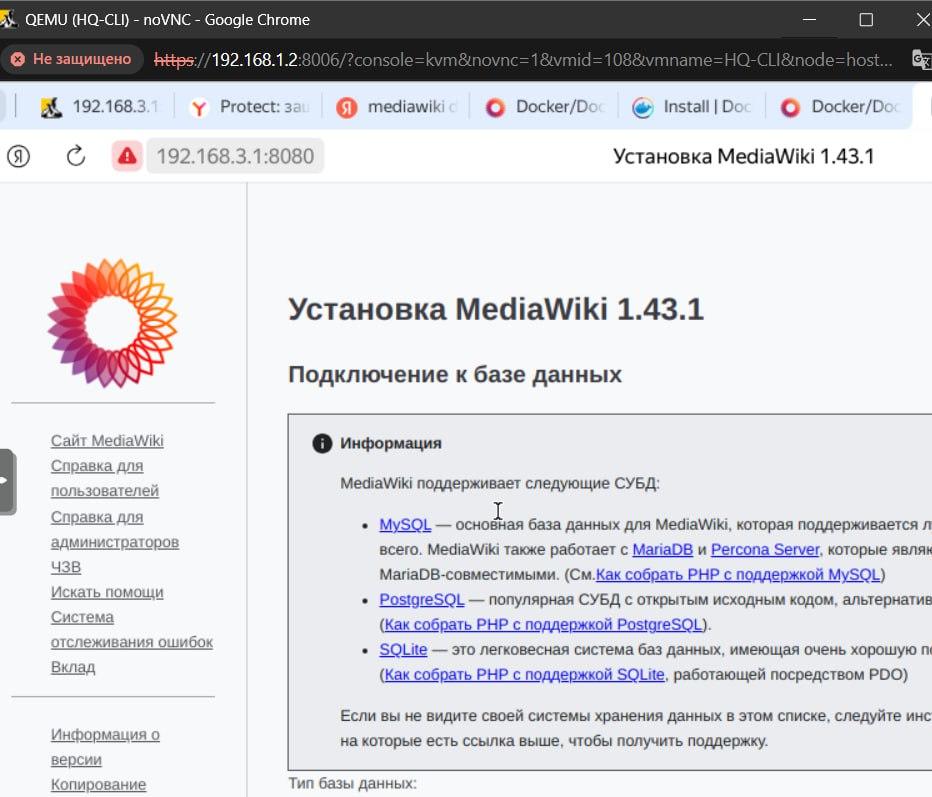
нажимаем «set up the wiki» приступаем к установке



выбираем русский язык, далее







Настраиваем параметры базы данных:

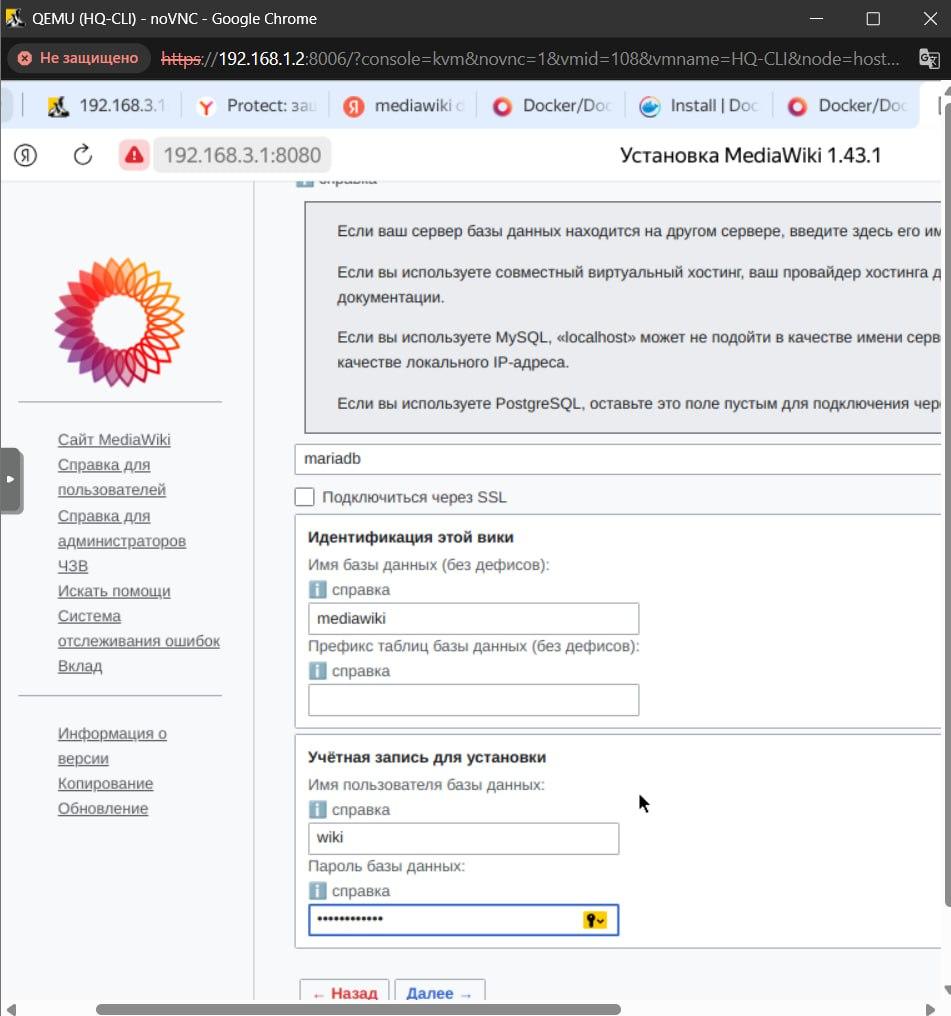
Хост: mariadb

Идентификатор:mediawiki

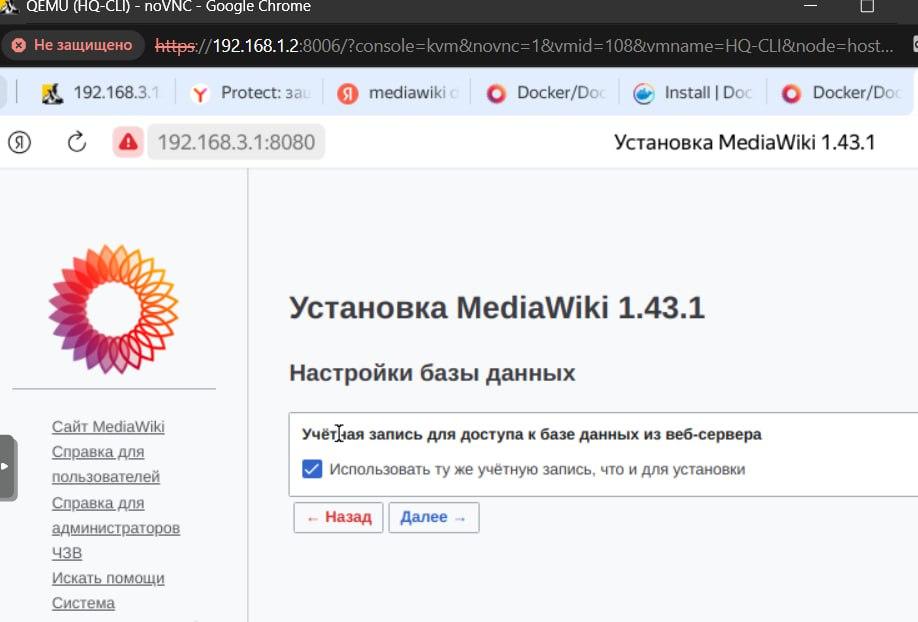
Учетная запись: wiki

WikiP@ssw0rd

Далее



Подтверждаем



Задаем параметры

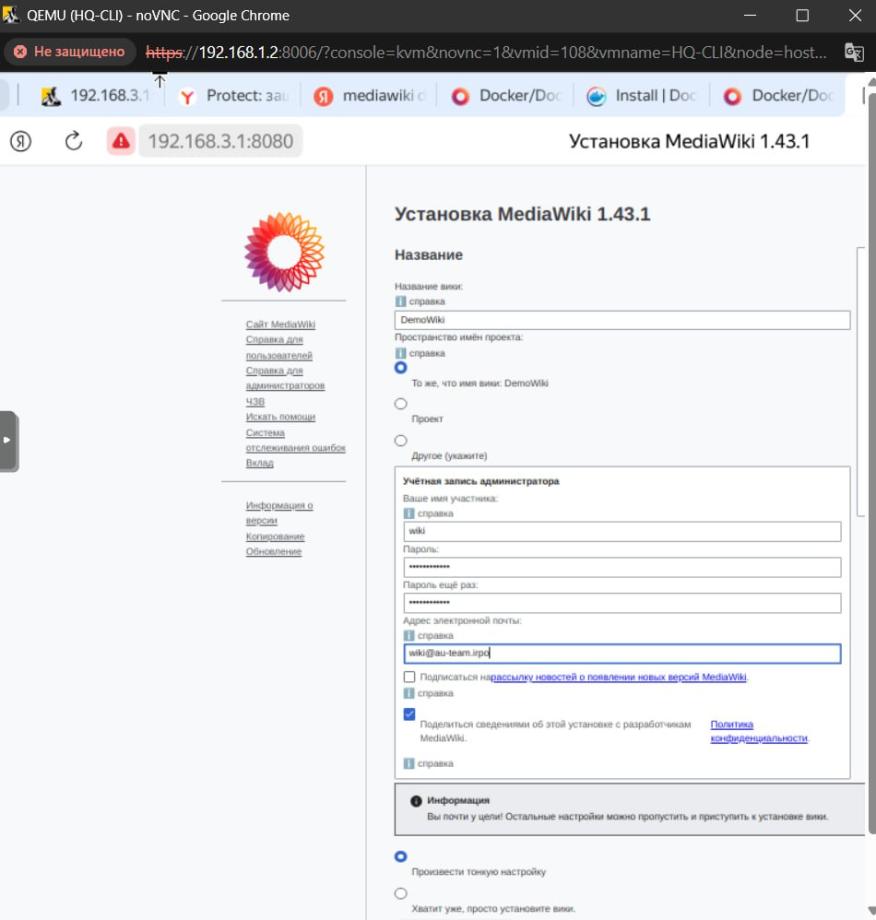
Название — DemoWiki

Учетная запись — wiki WikiP@ssw0rd

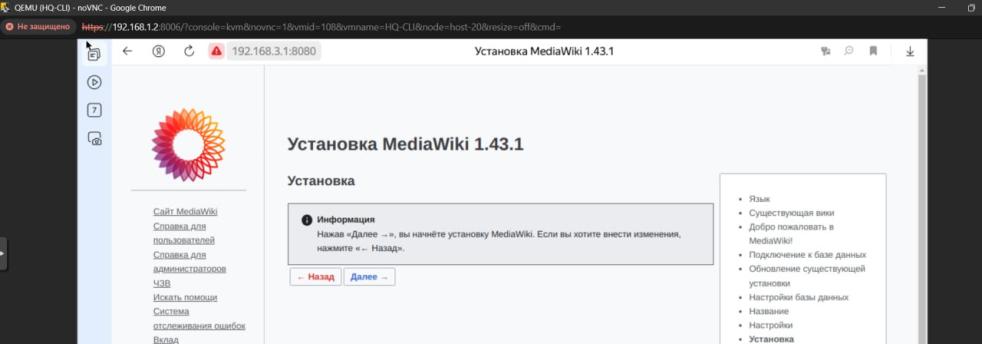
адресс — wiki@au-team.irpo

выбираем пункт «хватит уже, просто установить»

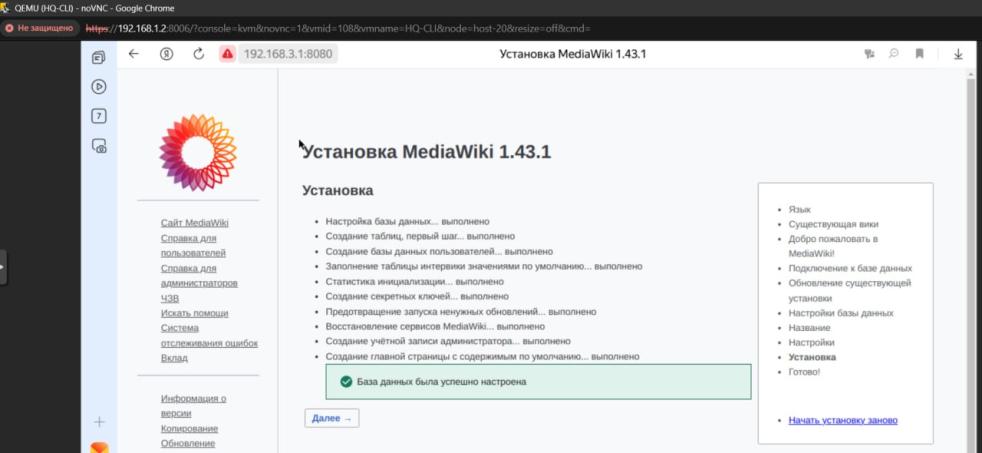
Далее



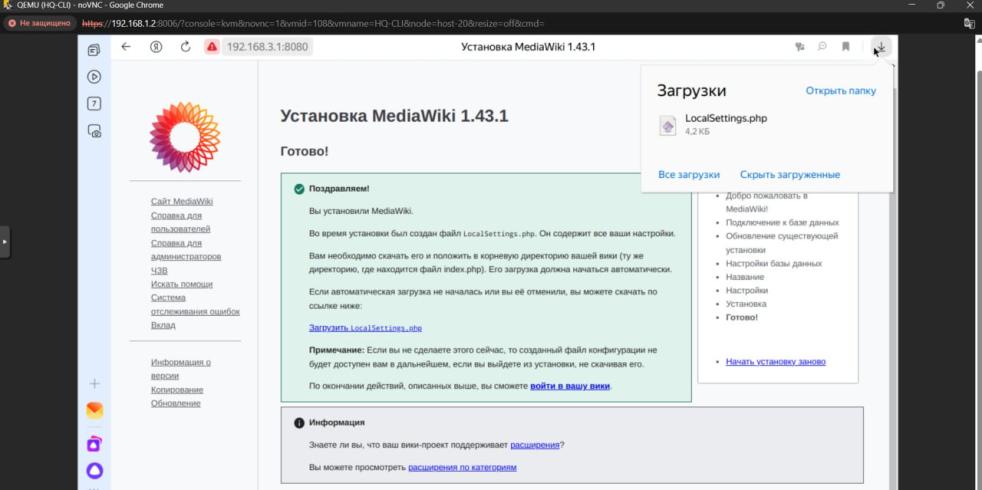
Далее



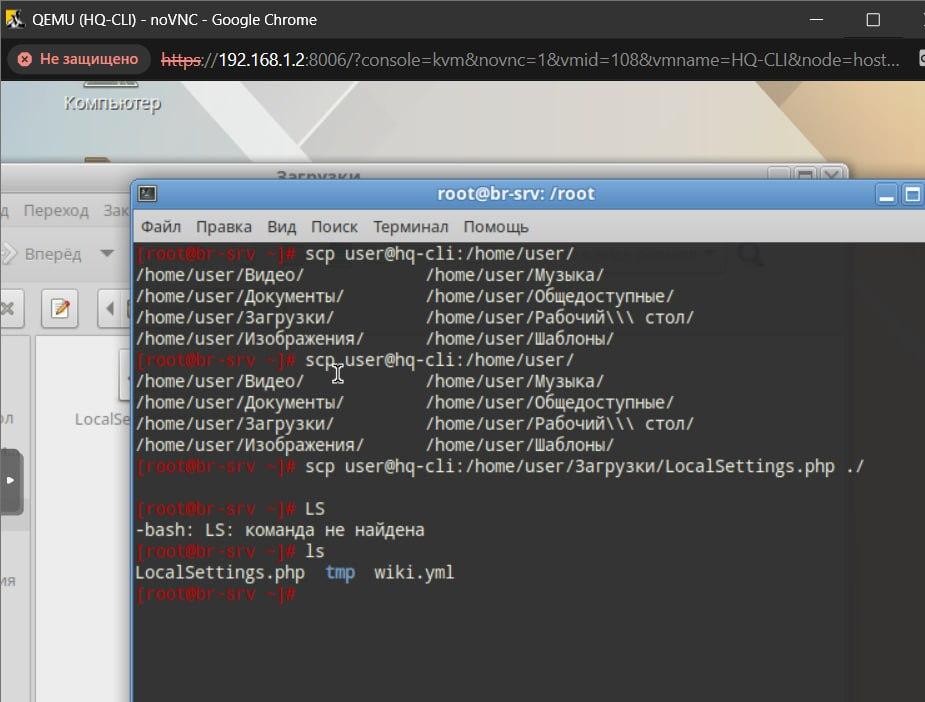
Далее



После установки выдается файл и его надо передать на сервер br-srv

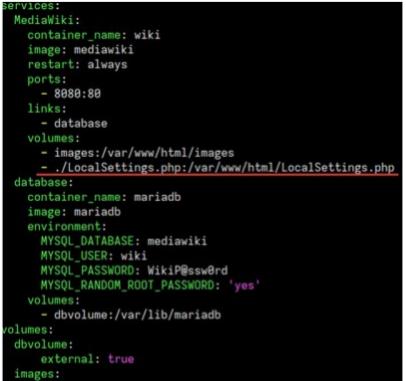


Необходимо передать файл LocalSettings.php с CLI на сервер:



Раскомментируем строку в файле wiki.yml:

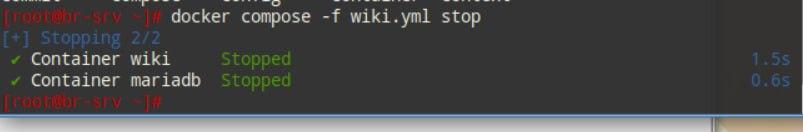
# nano wiki.yml

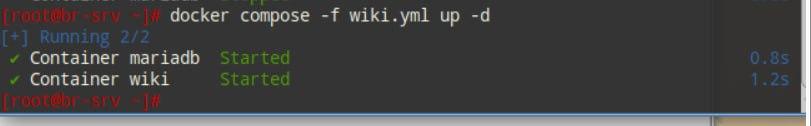


Перезапускаем сервисы средствами docker-compose:

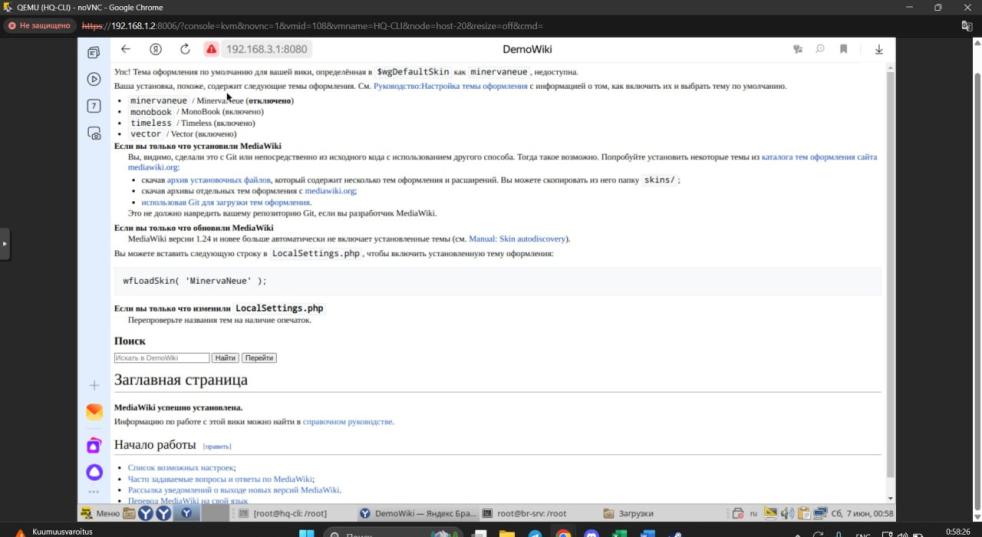
# docker compose -f wiki.yml stop

# docker compose -f wiki.yml up -d





Проверяем доступ к Wiki <http://192.168.3.1:8080>



**НАСТРОЕНО**