**Отчет по демо экзамену представленный Баженовым Алексеем, человеку, который был удостоен решать 3 модуля + вариатив**

# **Модуль 1**

**1. Произведите базовую настройку устройств**

Присвою имена согласно топологии для *VM*: *ISP*, *HQ-RTR*, *BR-RTR*, *HQ-SRV, BR-SRV*, *HQ-CLI* (Рисунок 1).

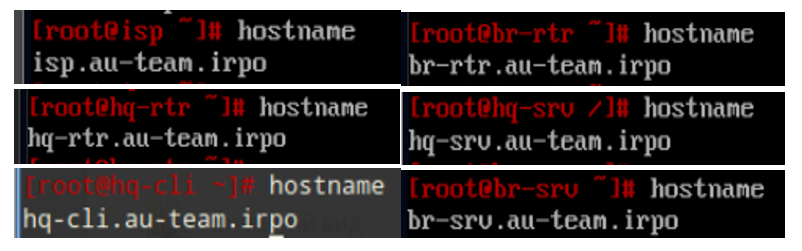


Рисунок 1 – Настройка имен

Сведенья об настроенных *ip* адресах занес в таблицу 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя ВМ | NIC | IPv4-адреса | Шлюз |
| ISP | ens18 | dhcp | - |
| ens19 | 172.16.4.1/28 | - |
| ens20 | 172.16.5.1/28 | - |
| HQ-RTR | ens18 | 172.16.4.2/28 | - |
| ens19.100 | 192.168.1.1/26 | - |
| ens19.200 | 192.168.2.1/28 | - |
| ens19.999 | 192.168.99.1/29 | - |
| BR-RTR | ens18 | 172.16.5.2/28 | - |
| ens19 | 192.168.3.1/27 | - |
| HQ-SW | ens18  ens19  ens20 | Коммутатор настроен средствами PROSMOX | - |
| HQ-SRV | ens18 | 192.168.1.10/26 | 192.168.1.1 |
| HQ-CLI | ens18 | dhcp (vlan 200) | 192.168.2.1 |
| BR-SRV | ens18 | 192.168.3.10/27 | 192.168.3.1 |

Таблица 1

**2 Настройка ISP**

**Настройка Nat**

iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE

iptables-save > /etc/sysconfig/iptables

**3 Создание пользователей**

**На HQ-RTR, BR-RTR**

useradd net\_admin -d /home/net\_admin -m -G users -s /bin/bash – создал пользователя

passwd net\_admin – дал ему пароль, который указан ниже

P@ssw0rd

vim /etc/sudoers - заходим и редактируем файл

net\_admin ALL=(ALL) NOPASSWD: ALL – установил без парольную аутификацию

WHEEL\_USERS ALL=(ALL:ALL) NOPASSWD: ALL

usermod -aG wheel net\_admin – добавил его в группу

**На HQ-SRV, BR-SRV**

useradd sshuser -d /home/sshuser -m -G users -s /bin/bash -u 1010 создал пользователя

passwd sshuser дал ему пароль, который указан ниже

P@ssw0rd

vim /etc/sudoers – заходим и редактируем файл

sshuser ALL=(ALL) NOPASSWD: ALL установил без парольную аутификацию

WHEEL\_USERS ALL=(ALL:ALL) NOPASSWD: ALL

usermod -aG wheel sshuser

**4 Настройте на интерфейсе HQ-RTR в сторону офиса HQ виртуальный коммутатор.**

Коммутатор настроен средствами PROSMOX

Для полноценной работы, выполним настройку интерфейсов на HQ-RTR как указано на рисунке 2.

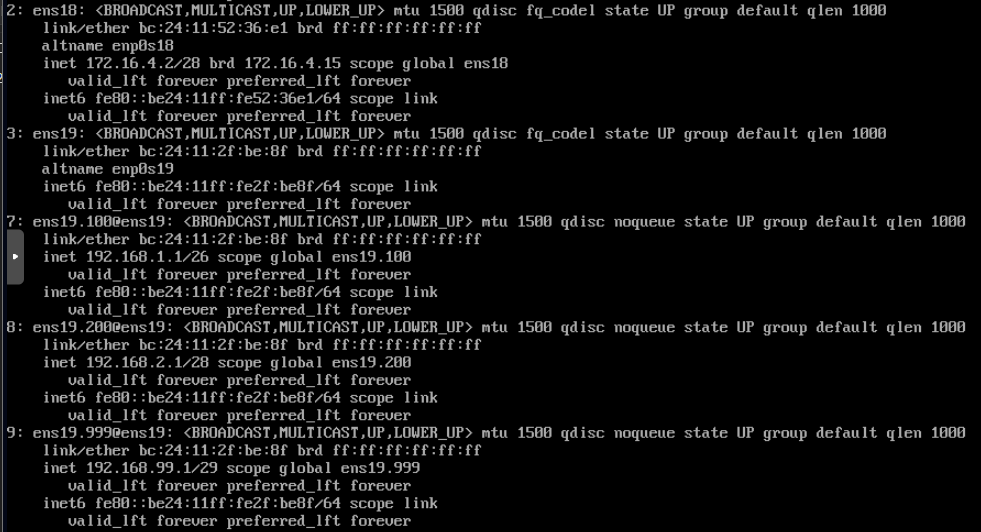


Рисунок 2 – Настройка vlan на HQ-RTR

**5. Настройка безопасного удаленного доступа на серверах HQ-SRV и BR-SRV**

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner – Создаем файл содержащий банерное сообщение

vim /etc/openssh/sshd\_config – заходим и правим файл конфигурации ssh

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2 – изменяем кол-во попыток входа

Banner /etc/openssh/banner – Говорим использовать банер и указываем до него путь

Allowusers sshuser – аутификация возможно только пользователю sshuser

**6. Между офисами HQ и BR необходимо сконфигурировать ip туннель**

Поскольку мы делаем 3 модуль, сразу сделаем его шифрованным с использованием средств **strongswan**

mkdir /etc/net/ifaces/iptunnel – создаем отдельную директорию

vim /etc/net/ifaces/iptunnel/options – добавим настройки

TYPE=iptun

TUNTYPE=gre

TUNLOCAL=172.16.4.2

TUNREMOTE=172.16.5.2

TUNOPTIONS='ttl 64'

DISABLE=no

TUNMTU=1476

echo 10.0.0.1/30 > /etc/net/ifaces/iptunnel/ipv4address – добавим ip адрес

apt-get install -y strongswan – установим сервис

vim /etc/strongswan/ipsec.conf – настроим шифрование

conn gre

auto=start

type=tunnel

authby=secret

left=10.0.0.1

right=10.0.0.2

leftsubnet=0.0.0.0/0

rightsubnet=0.0.0.0/0

leftprotoport=gre

rightprotoport=gre

ike=aes256-sha2\_256-modp1024!

esp=aes256-sha2\_256!

vim /etc/strongswan/ipsec.secrets – добавляем пароль

10.0.0.1 10.0.0.2 : PSK "P@ssw0rd"

**На BR-RTR Настройка выполняется аналогично**

**7. Обеспечьте динамическую маршрутизацию: ресурсы одного офиса должны быть доступны из другого офиса. Для обеспечения динамической маршрутизации используйте link state протокол на ваше усмотрение.**

Для это настроим динамическую маршрутизацию средствами frr на роутерах HQ-RTR И BR-RTR

apt-get update && apt-get install -y frr – устаналиваем службу

vim /etc/frr/daemons

Включем ospf

systemctl enable --now frr – ставим на автозапуск

Для HQ-RTR конфиг:

vtysh

conf t

ip forwarding

router ospf

network 10.0.0.0/30 area 0

network 192.168.1.0/26 area 0

network 192.168.2.0/28 area 0

network 192.168.99.0/29 area 0

do wr

exit

Для BR-RTR конфиг:

vtysh

conf t

ip forwarding

router ospf

network 10.0.0.0/30 area 0

network 192.168.3.0/27 area 0

do wr

exit

Для проверки выведем маршруты и отправим traceroute (Рисунок 4)

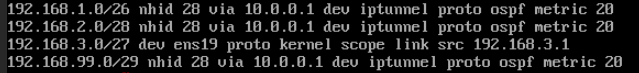
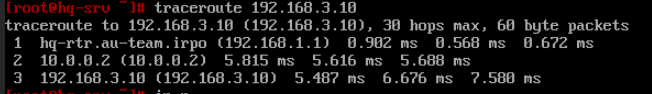


Рисунок 4 – Проверка туннеля

**8. Настройка динамической трансляции адресов.**

Для трансляции адресов буду использовать сервис iptables

iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE – добавляю новое правило

iptables-save > /etc/sysconfig/iptables сохраняю его в файл

**9. Настройка протокола динамической конфигурации хостов.**

Для настройки динамической конфигурации хостов буду использовать сервис dnsmasq

apt-get install dnsmasq **-y –** устанавливаю сервис

systemctl enable --now dnsmasq – ставлю на автозапуск

vim /etc/dnsmasq.conf редактирую файл конфигурации

no-resolv

dhcp-range=192.168.2.2,192.168.2.15,999h – испытуемый диапазон

dhcp-option=3,192.168.2.1 - шлюз

dhcp-option=6,192.168.1.10 – DNS сервер

interface=ens19.200 – интерфейс для раздачи

Проверим получение ip на cli (Рисунок 5)

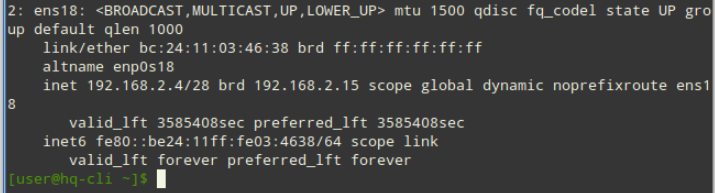


Рисунок 5 – Динамическое получение ip на hq-cli

**10 Настройка DNS для офисов HQ и BR**

Для настройки DNS буду использовать сервис dnsmasq

apt-get install dnsmasq **-y –** устанавливаю сервис

systemctl enable --now dnsmasq – ставлю на автозапуск

vim /etc/dnsmasq.conf редактирую файл конфигурации

no-resolv

domain=au-team.irpo

server=/au-team.irpo/192.168.3.10

server=8.8.8.8 – сервер для пересылки в интернете

interface=ens18 – интерфейс для раздачи

address=/hq-rtr.au-team.irpo/192.168.1.1

ptr-record=1.1.168.192.in-addr.arpa,hq-rtr.au-team.irpo

address=/wiki.au-team.irpo /192.168.1.1

address=/moodle.au-team.irpo /192.168.1.1 Разрешение имен

address=/br-rtr.au-team.irpo/192.168.3.1

address=/hq-srv.au-team.irpo/192.168.1.10

ptr-record=10.1.168.192.in-addr.arpa,hq-srv.au-team.irpo

address=/hq-cli.au-team.irpo/192.168.2.4

ptr-record=4.2.168.192.in-addr.arpa,hq-cli.au-team.irpo

address=/br-srv.au-team.irpo/192.168.3.10

Проверка на рисунке 6

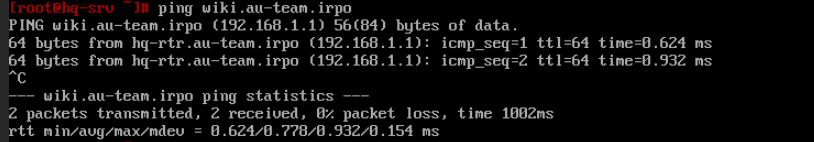


Рисунок 6 – Проверка работы DNS сервера

**11. Настройте часовой пояс на всех устройствах, согласно месту проведения экзамена.**

Поскольку системный администратор хорошо поработал, остается настроить только на ВМ hq-cli, сделаем это следующей командой

timedatectl set-timezone Asia/Krasnoyarsk

**ОТЧЕТ ПО МОДУЛЮ 1 ЗАВЕРШЕН**

# **Модуль 2**

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

apt-get install task-samba-dc **-y –** установим службу

samba-tool domain provision **–** запускаем установку, до этого удалив старую конфигурацию

Теперь создадим 5 пользователей:

**samba-tool user add user1.hq 123qweR%**

**samba-tool user add user2.hq 123qweR%**

**samba-tool user add user3.hq 123qweR%**

**samba-tool user add user4.hq 123qweR%**

**samba-tool user add user5.hq 123qweR%**

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**

systemctl start samba

Теперь введём клиентскую машину в домен (Рисунок 7)

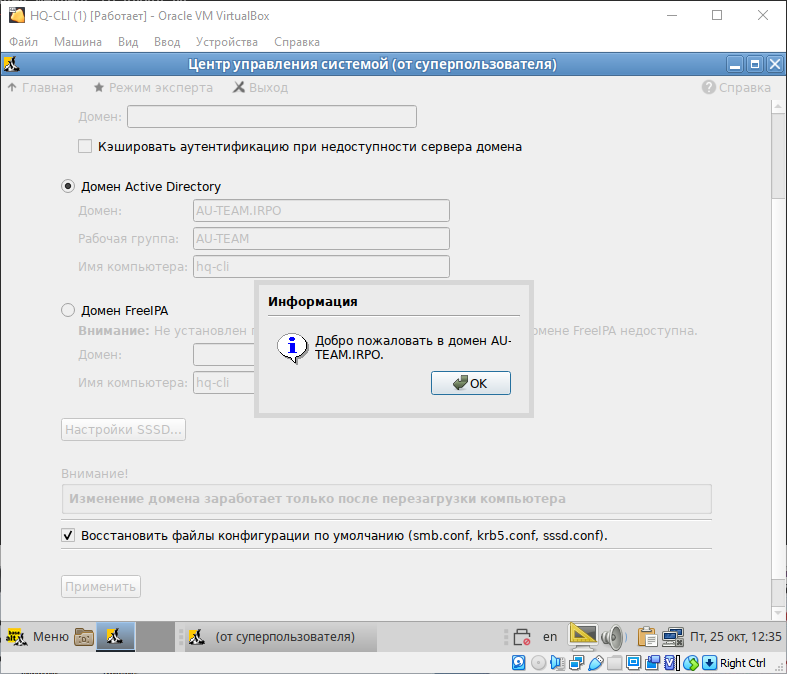


Рисунок 7 – Добавление клиентской машины в домен

Произведем импорт пользователей

**Скрипт**

**curl -L** [**https://bit.ly/3C1nEYz**](https://bit.ly/3C1nEYz) **> /root/users.zip –** скачивание архива

**unzip /root/users.zip –** распаковка архива

**mv /root/Users.csv /opt/Users.csv –** перемещение архива в директорию согласно задания и проверка (Рисунок 8)

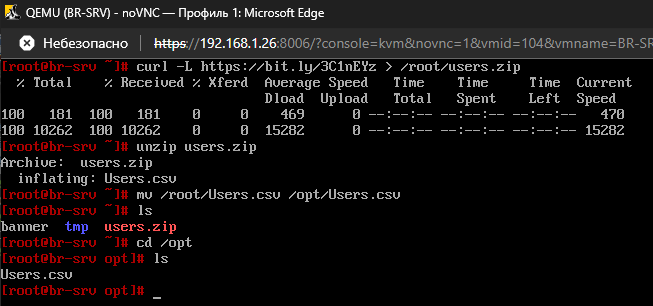
****

Рисунок 8 – Работа с архивом

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

**vim import**

**#!/bin/bash**

**csv\_file=”/opt/Users.csv”**

**while IFS=”;” read -r firstName lastName role phone ou street zip city country password; do**

**if [ “$firstName” == “First Name” ]; then**

**continuevim**

**fi**

**username=”${firstName,,}.${lastName,,}”**

**sudo samba-tool user add “$username” 123qweR%**

**done < “$csv\_file”**

Сохраняем этот файл и выдаём ему право на выполнение и запускаем его:

**chmod +x /root/import** – даем права за запуск

**bash /root/import –** запуск скрипта

**2. Сконфигурируйте файловое хранилище:**

**lsblk**

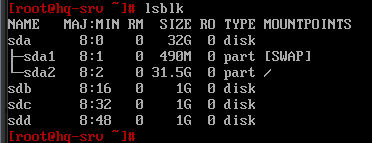


Рисунок 9

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

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

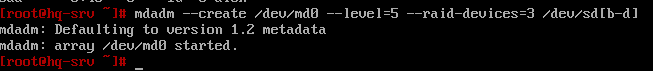
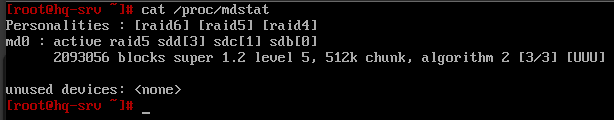


Рисунок 10 –

Посмотрим статус нашего raid-массива:

**cat /proc/mdstat**



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

**mdadm --detail -scan > /etc/mdadm.conf**

Теперь создаём раздел через **fdisk**.

Для этого пишем следующую команду:

**fdisk /dev/md0**



Рисунок 11 –

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:

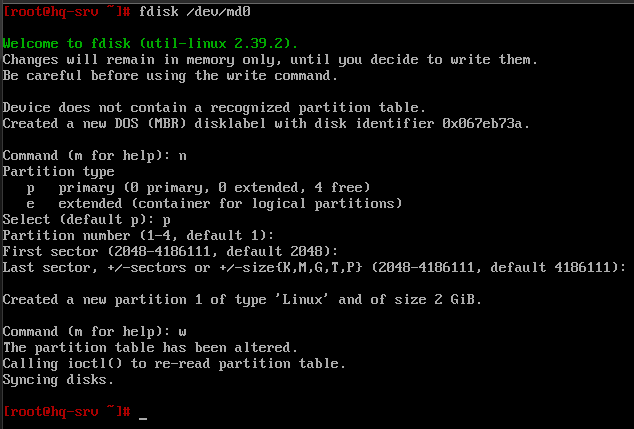
****

Рисунок 12 –

Теперь создадим файловую систему, по заданию требуется **ext4**, создаём её следующей командой:

**mkfs.ext4 /dev/md0p1**

****

Рисунок 13 –

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

vim **/etc/fstab**

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

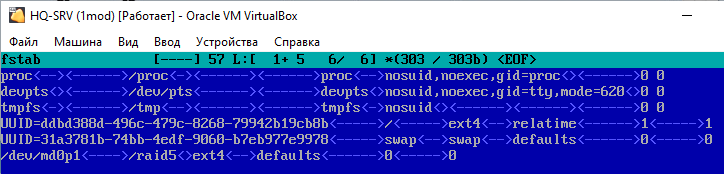


Рисунок 14 –

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 -y**

**systemctl enable --now nfs-server.service**

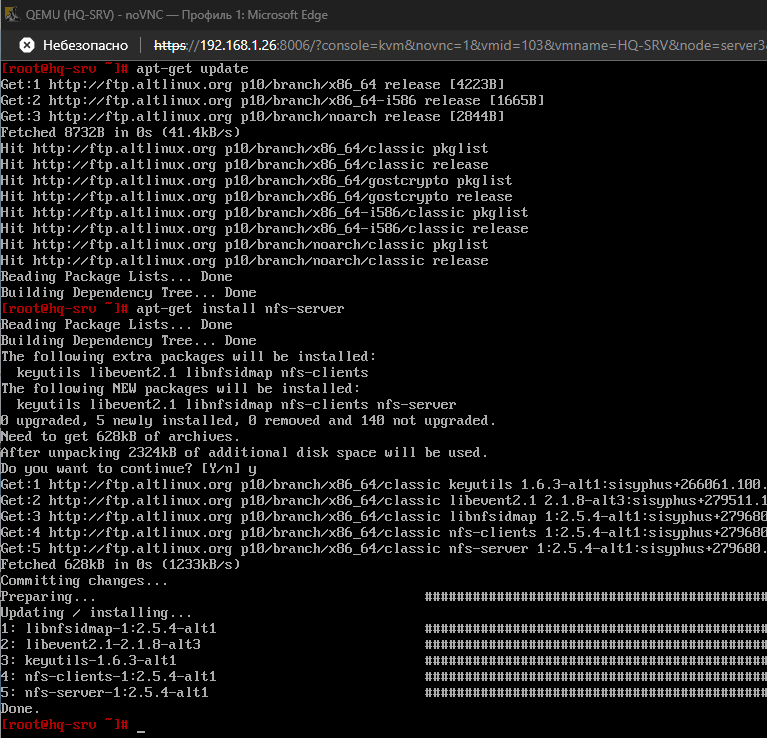


Рисунок 15

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:

vim **/etc/exports**

**/raid5/nfs 192.168.2.0/28(rw,sync,no\_subtree\_check)**

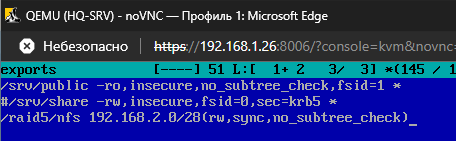


Рисунок 16

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

**exportfs -a**

**exportfs -v**

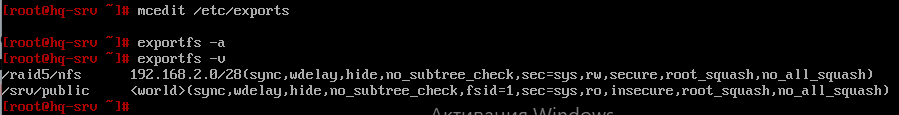


Рисунок 17

Перезапускаем службу NFS:

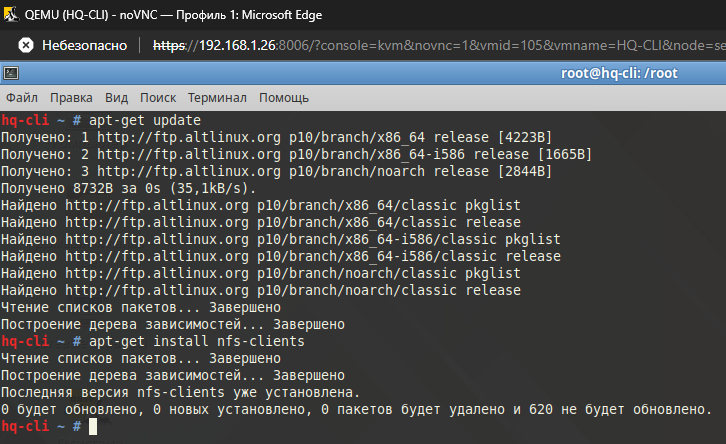
**systemctl restart nfs**

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

**apt-get update**

**apt-get install nfs-clients -y**

**systemctl enable --now nfs-client.target**

**** Рисунок 18

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**

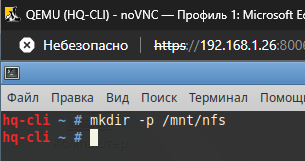
****

Рисунок 18

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

vim **/etc/fstab**

**192.168.1.10:/raid5/nfs /mnt/nfs nfs defaults 0 0**

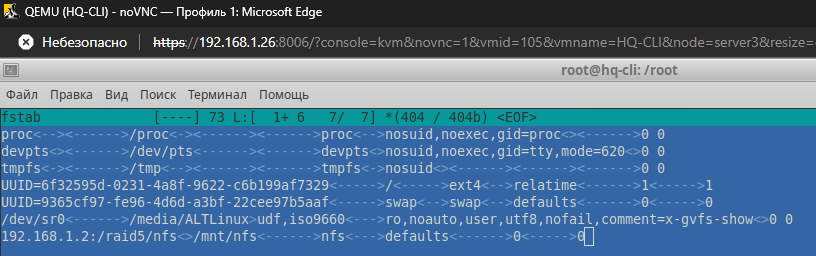


Рисунок 19

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

**mount -a**

**mount -v**

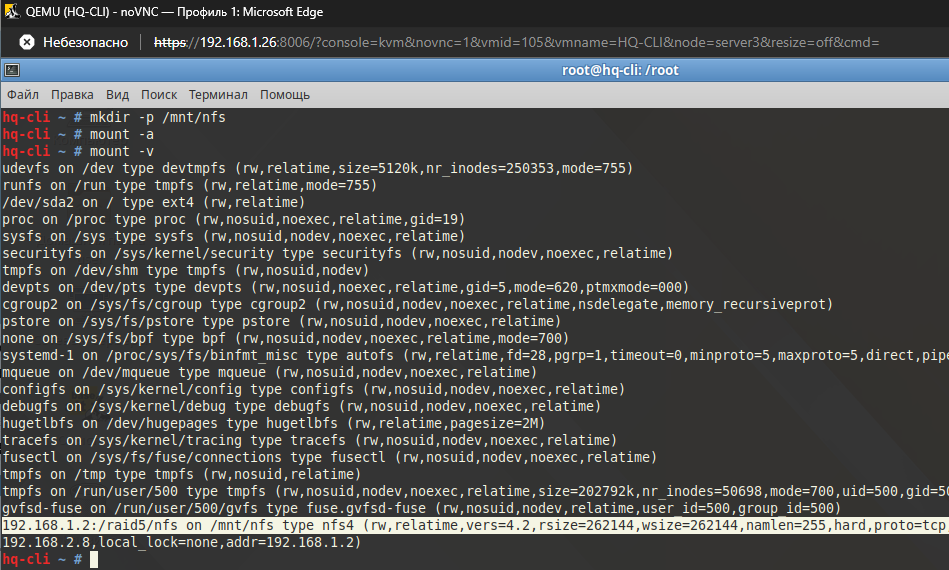


Рисунок 20

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

**touch /mnt/nfs/bazhenov**

Проверим, что файл появился на сервере (Рисунок 21)



Рисунок 21 – Проверка работы файлового хранилища\

**3. Настройте службу сетевого времени на базе сервиса сhrony**

apt-get install chrony

vim /etc/chrony.conf

pool 192.168.44.5 iburst

local stratum 6

allow 192.168.1.0/26

allow 192.168.2.0/28

allow 192.168.3.0/27

allow 172.16.4.0/28

allow 172.16.5.0/28

Настройка на HQ-RTR, HQ-CLI, BR-RTR, BR-SRV

vim /etc/chrony.conf

pool 172.16.4.2 iburst – для br-srv и br-rtr, isp

pool 192.168.2.1 iburst – для hq-cli

pool 192.168.1.1 iburst – для hq-rtr

systemctl restart chronyd

systemctl enable --now chronyd

systemctl status chronyd

Для проверки используем

chronyc tracking

chronyc sources

Например, проверим на br-rtr (Рисунок 22).

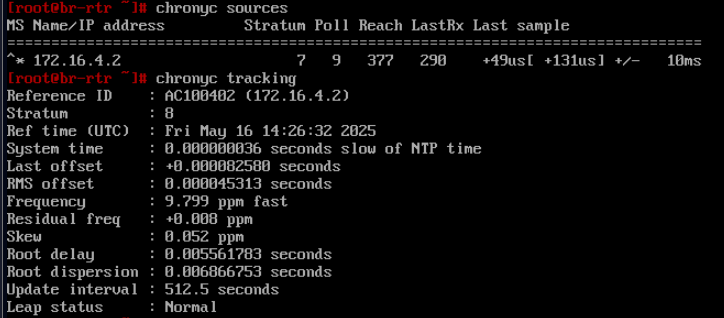


Рисунок 22 – Проверка работы сервиса времени

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

apt-get install ansible -y – устанавливаем службу

mkdir -p /etc/ansible создаем отдельную директорию

vim /etc/ansible/hosts создаем файл инвентаря и добавляем следующие содержание

hq-srv ansible\_host=sshuser@192.168.1.10 ansible\_port=2024

hq-cli ansible\_host=user@192.168.2.4 ansible\_port=22

hq-rtr ansible\_host=net\_admin@192.168.1.1 ansible\_port=2024

br-rtr ansible\_host=net\_admin@192.168.3.1 ansible\_port=2024

vim /etc/ansible/ansible.cfg

ansible\_python\_interpreter=/usr/bin/python3

apt-get update

apt-get install python-module-json – на машке cli, чтобы не возникало ошибки (АЛЬТ ПРОСТО ПРЕКРАСЕН)

Создаем и передаем сертификаты

ssh-keygen -t rsa

ssh-copy-id -p 2024 net\_admin@192.168.3.1

ssh-copy-id -p 22 user@192.168.2.4

ssh-copy-id -p 2024 sshuser@192.168.1.10

ssh-copy-id -p 2024 net\_admin@192.168.1.1

Проверка

ansible all -m ping (Рисунок 23)

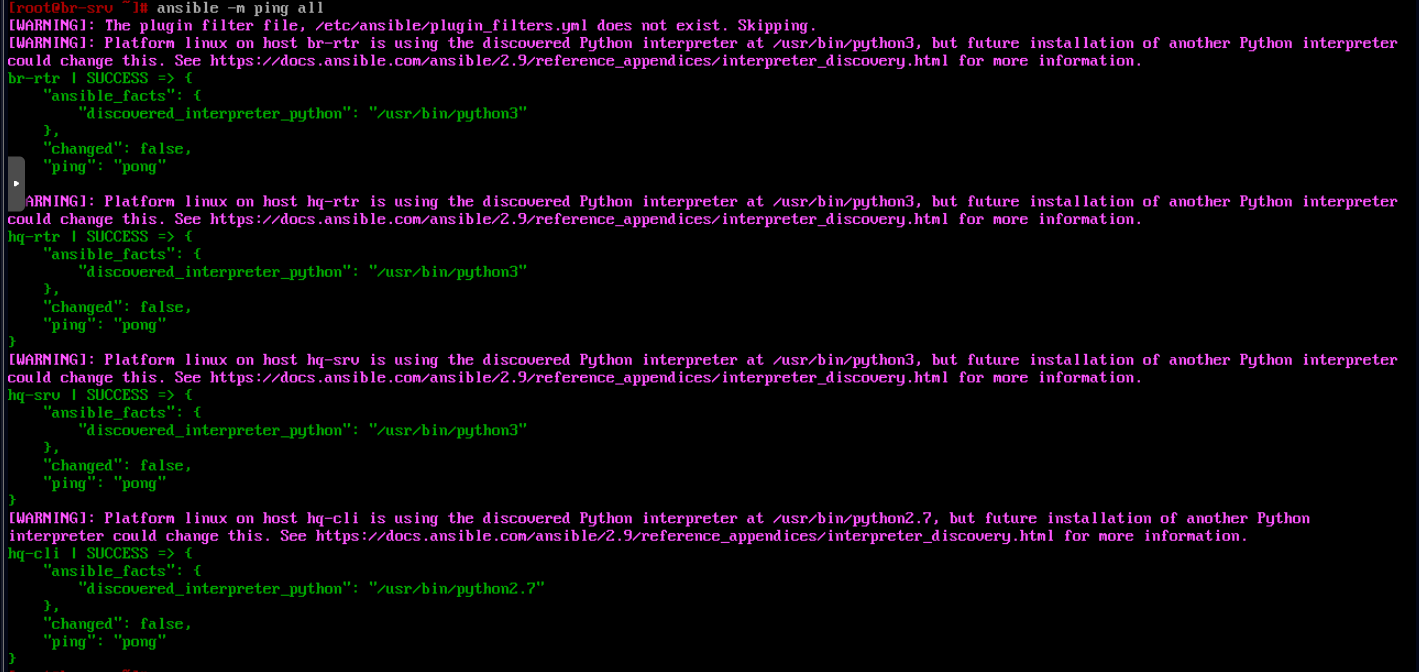


Рисунок 23 – Проверка работы ansible

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

apt-get install docker-engine docker-compose **-y –** установка докера

systemctl enable --now docker

systemctl status docker

docker pull mediawiki установка контера wiki

docker pull mariadb установка контера базы данных

vim /root/wiki.yml создаем файл конфигурации и добавляем следующий конфиг, согласно заданию

**services:**

**mariadb:**

**image: mariadb**

**container\_name: mariadb**

**restart: always**

**environment:**

**MYSQL\_ROOT\_PASSWORD: 123qweR%**

**MYSQL\_DATABASE: mediawiki**

**MYSQL\_USER: wiki**

**MYSQL\_PASSWORD: 123qweR%**

**volumes: [ mariadb\_data:/var/lib/mysql ]**

**wiki:**

**image: mediawiki**

**container\_name: wiki**

**restart: always**

**environment:**

**MEDIAWIKI\_DB\_HOST: mariadb**

**MEDIAWIKI\_DB\_USER: wiki**

**MEDIAWIKI\_DB\_PASSWORD: 123qweR%**

**MEDIAWIKI\_DB\_NAME: mediawiki**

**ports:**

**- "8080:80"**

**#volumes: [ /root/mediawiki/LocalSettings.php:/var/www/html/LocalSettings.php ]**

**volumes:**

**mariadb\_data:**

docker compose -f /root/wiki.yml up -d – запуск контейнера

Переходим на машину cli и вбиваем в браузере 192.168.3.10:8080

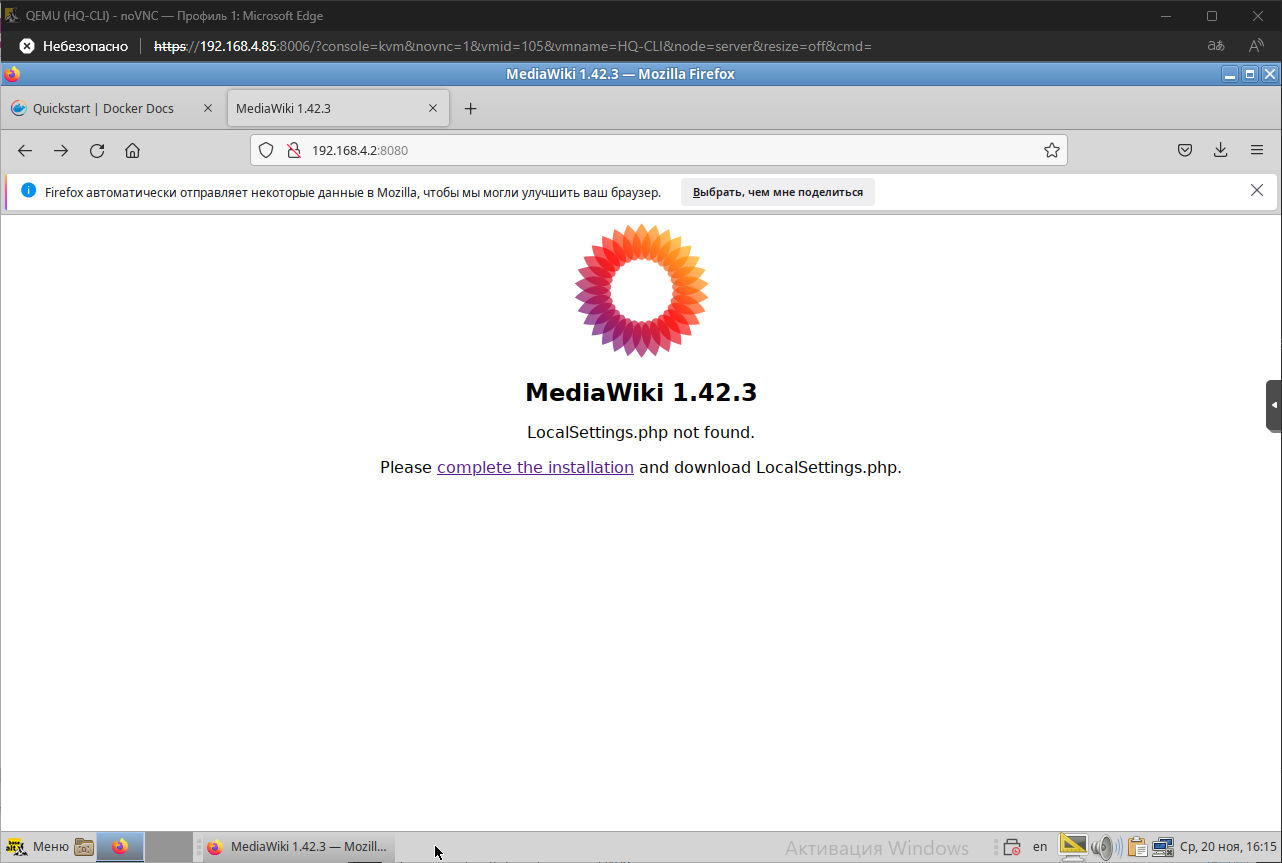
****

Рисунок 24

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



Рисунок 24

Нажимаем далее и видим строки, которые нужно заполнить:

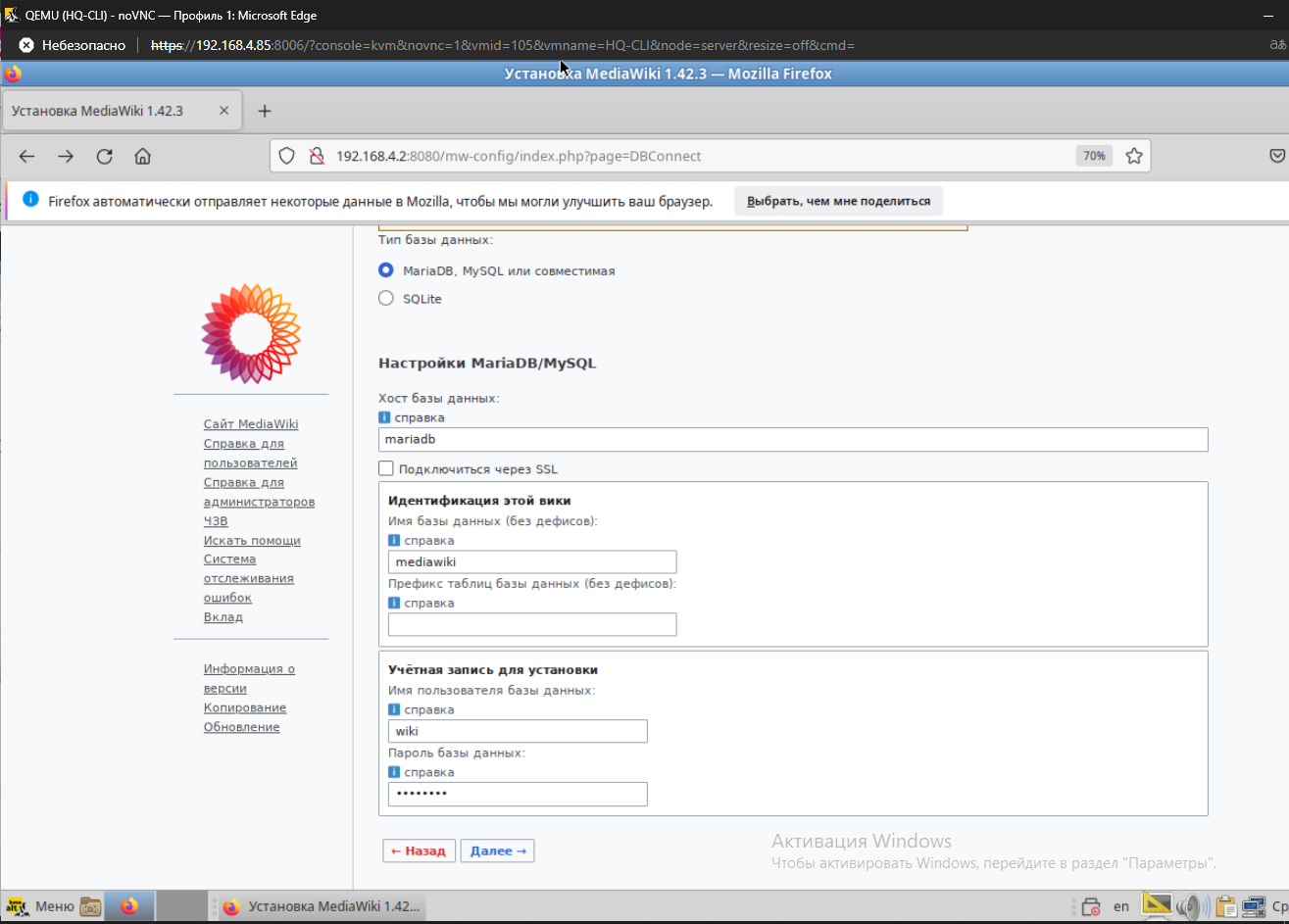


Рисунок 24

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

**wiki**

**WikiP@ssw0rd**

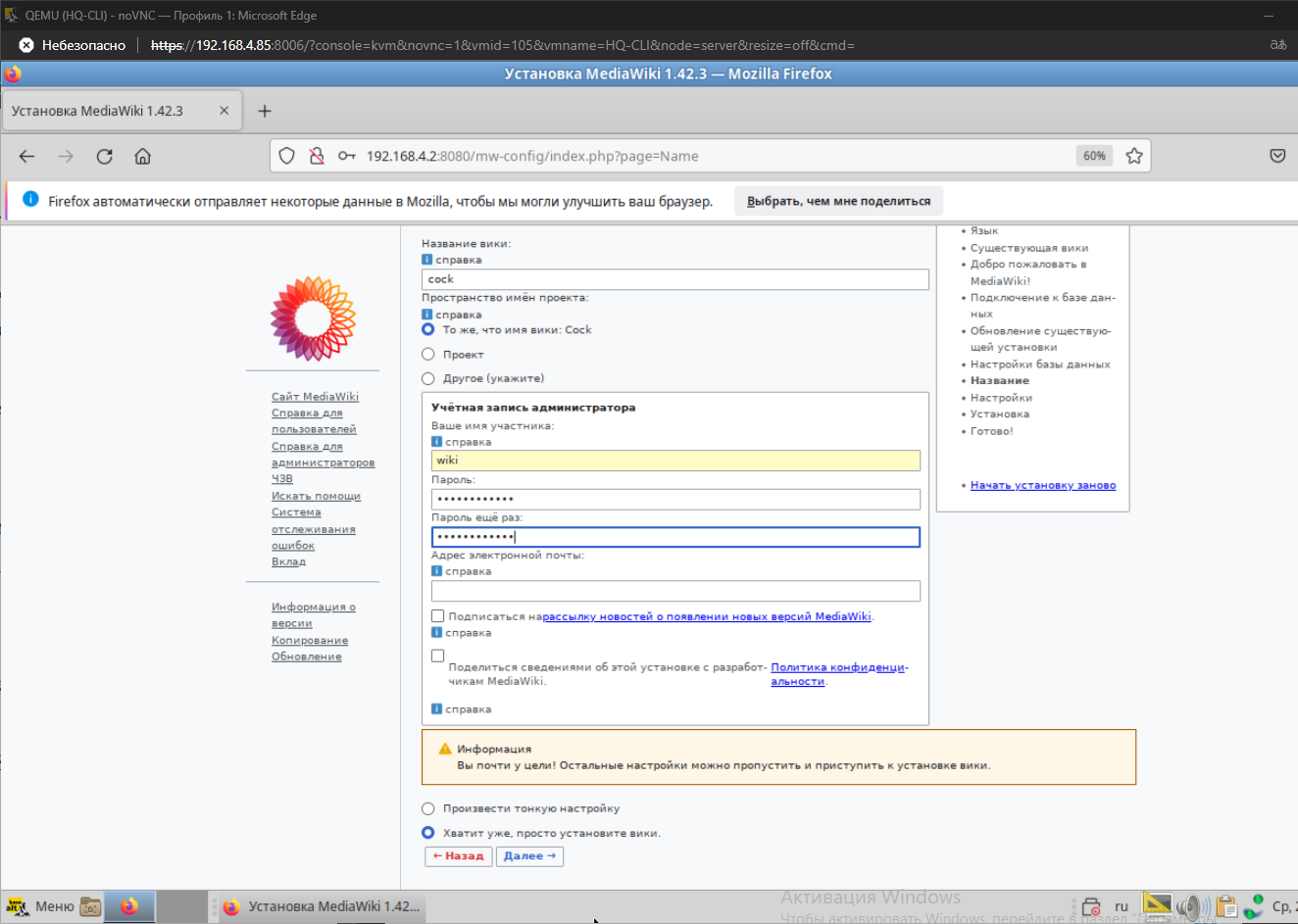


Рисунок 25

Нажимаем далее:

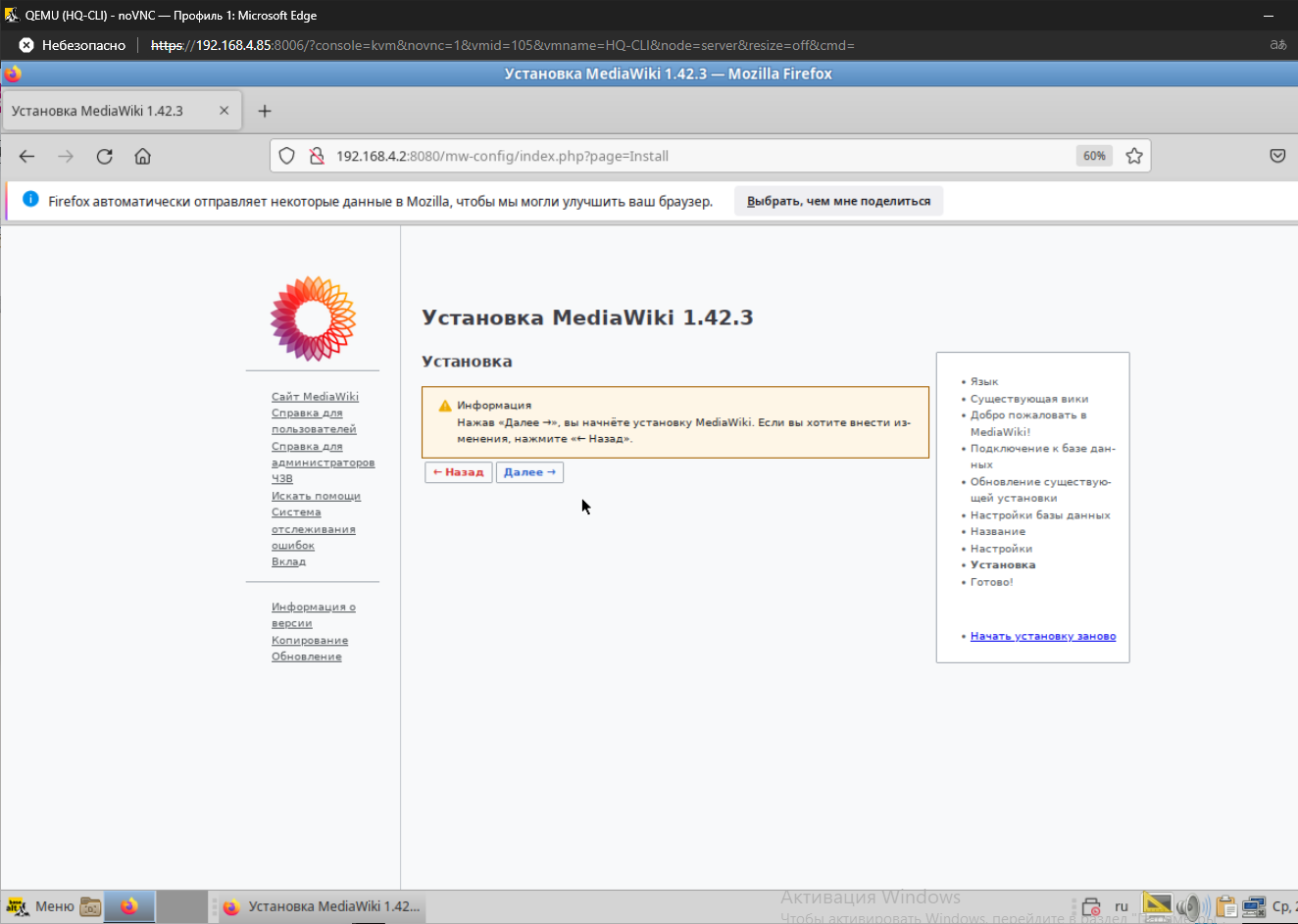
****

Рисунок 25

И вот мы успешно создали базу данных:

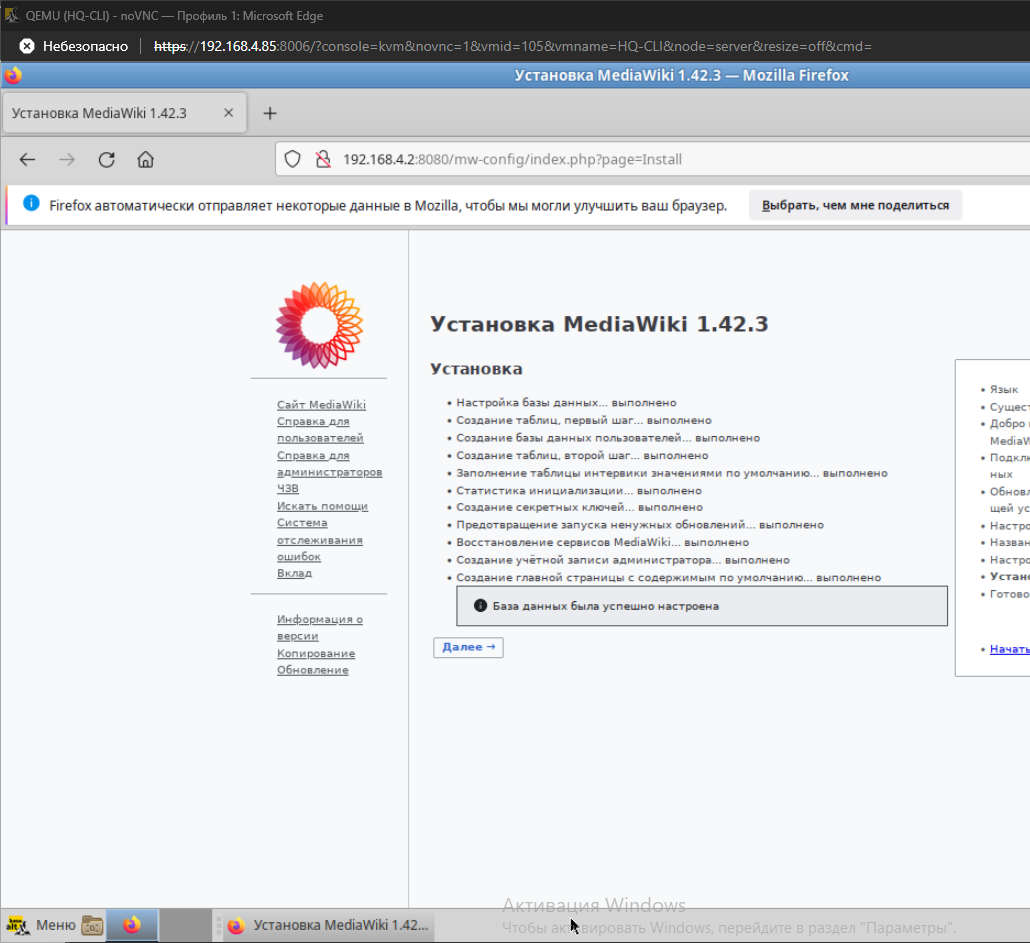


Рисунок 26

**scp -P 2024 /home/user/Загрузки/LocalSettings.php** [**sshuser@192.168.3.10:/home/sshuser/**](mailto:sshuser@192.168.3.10:/home/sshuser/) **-**перемещаем файл на сервес

Возращаемся на br-srv

rm -rf /root/LocalSettings.php

mkdir /root/mediawiki

mv /home/sshuser/LocalSettings.php /root/mediawiki/ - перемещаем файл согласно задания

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



Рисунок 27

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:

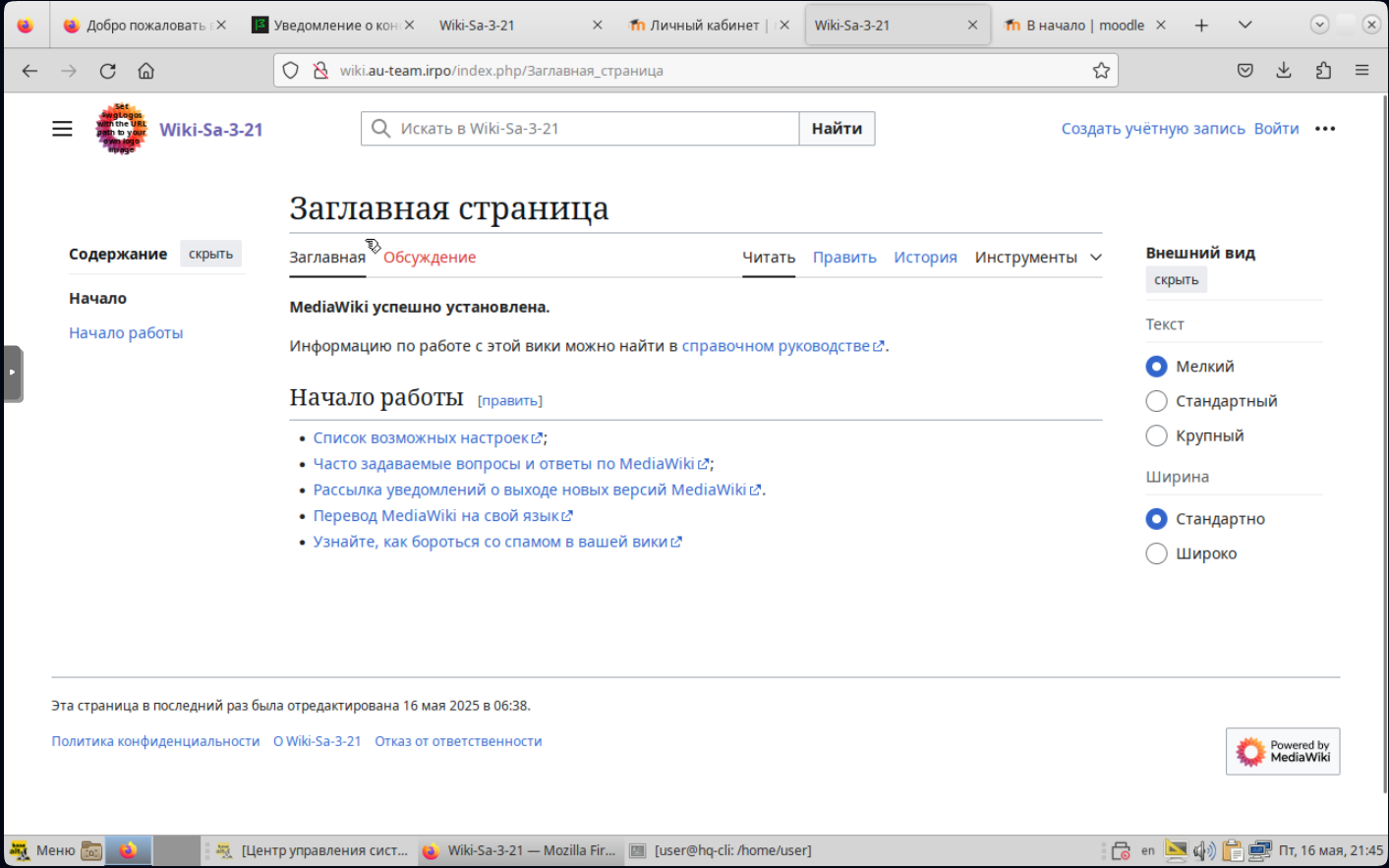
****

Рисунок 28

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

На HQ-RTR

iptables -t nat -A POSTROUTING -o ens18 -p tcp --dport 2024 -d 192.168.1.1 -j SNAT --to-source 192.168.1.10:2024 – само правило

iptables-save > /etc/sysconfig/iptables – сохраняем в файле

На BR-RTR

iptables -t nat -A POSTROUTING -o ens18 -p tcp --dport 80 -d 192.168.3.1 -j SNAT --to-source 192.168.3.10:8080 – само правило

iptables -t nat -A POSTROUTING -o ens18 -p tcp --dport 2024 -d 192.168.3.1 -j SNAT --to-source 192.168.3.10:2024 – само правило

iptables-save > /etc/sysconfig/iptables – сохраняем в файле

**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 -y**

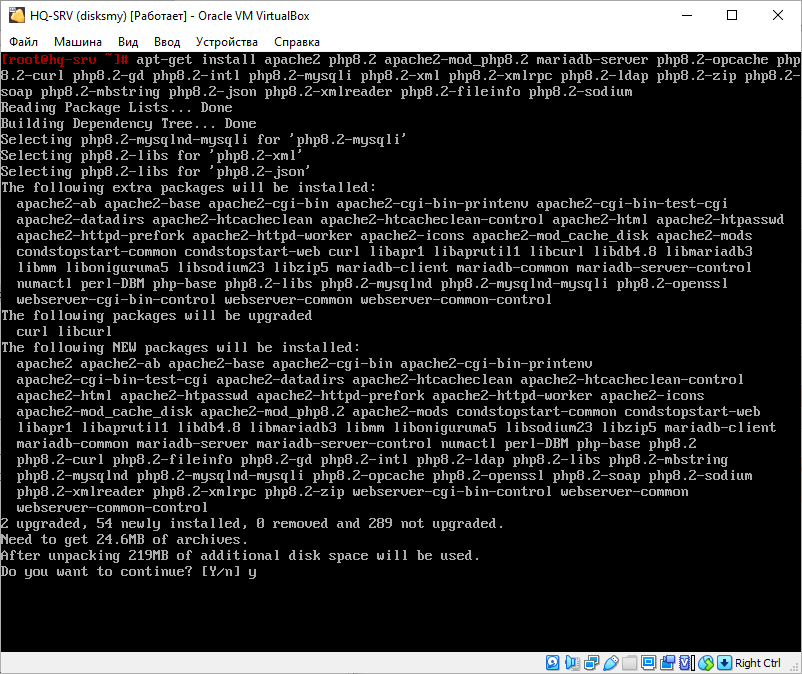
****

Рисунок 29

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

systemctl start **httpd2**

systemctl start **mysqld**

**systemctl enable –now httpd2 mysqld**

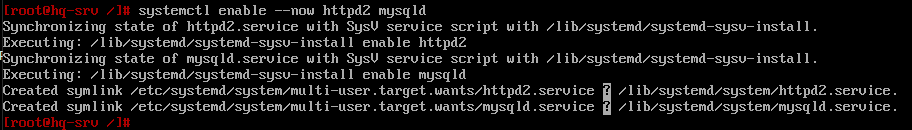
****

Рисунок 30

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:

**123qweR%**

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

**Y**

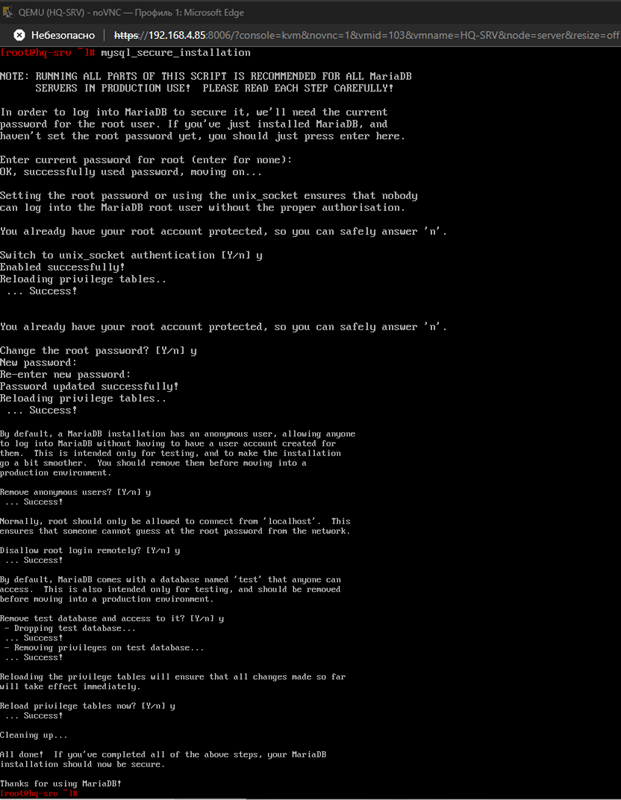
****

Рисунок 31

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**

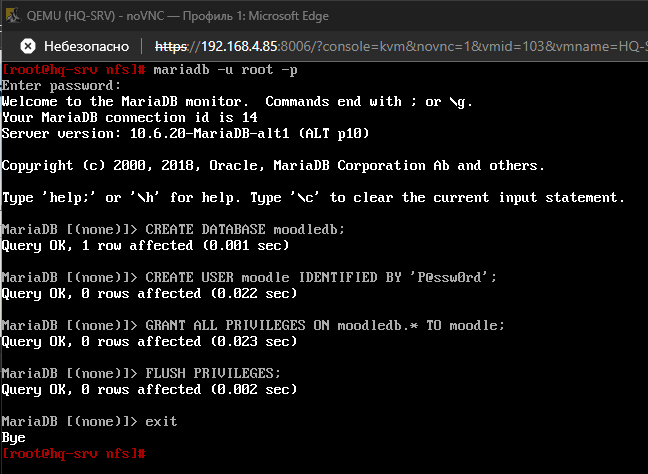
****

Рисунок 32

Теперь скачаем сам мудл стабильной версии:

**curl -L** [**https://github.com/moodle/moodle/archive/refs/tags/v4.5.0.zip**](https://github.com/moodle/moodle/archive/refs/tags/v4.5.0.zip) **> /root/moodle.zip**

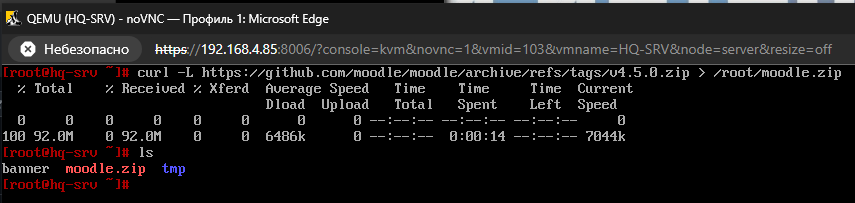
****

Рисунок 33

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**

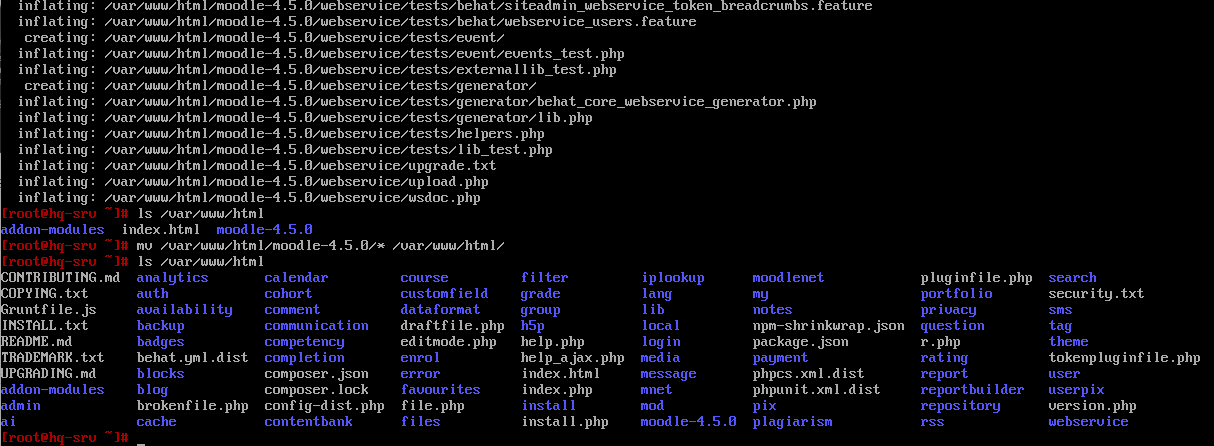
****

Рисунок 34

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**

**chmod 777 /var**

**chmod 777 /var/www**

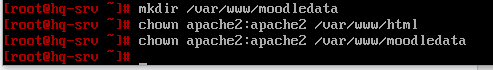
****

Рисунок 35

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**

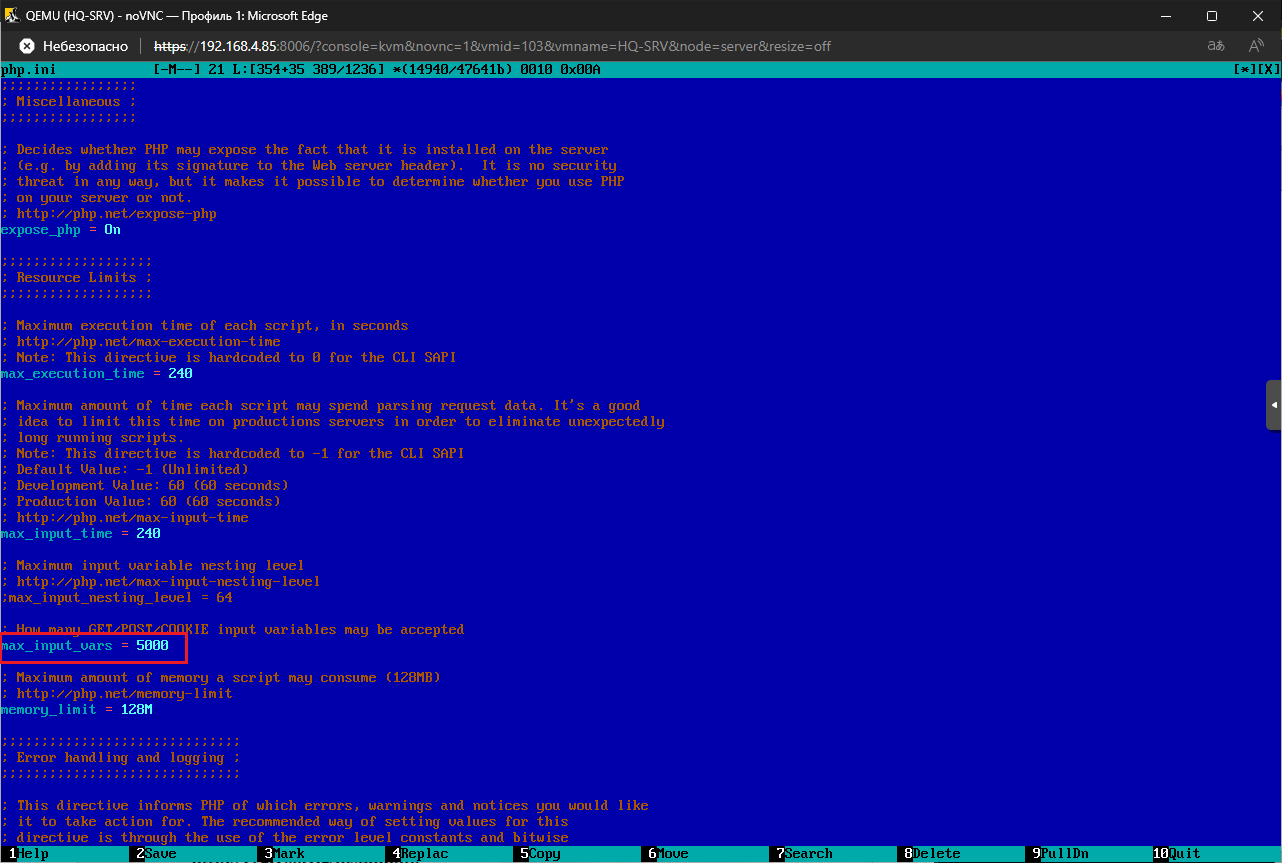
****

Рисунок 36

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://192.168.1.10/install.php**

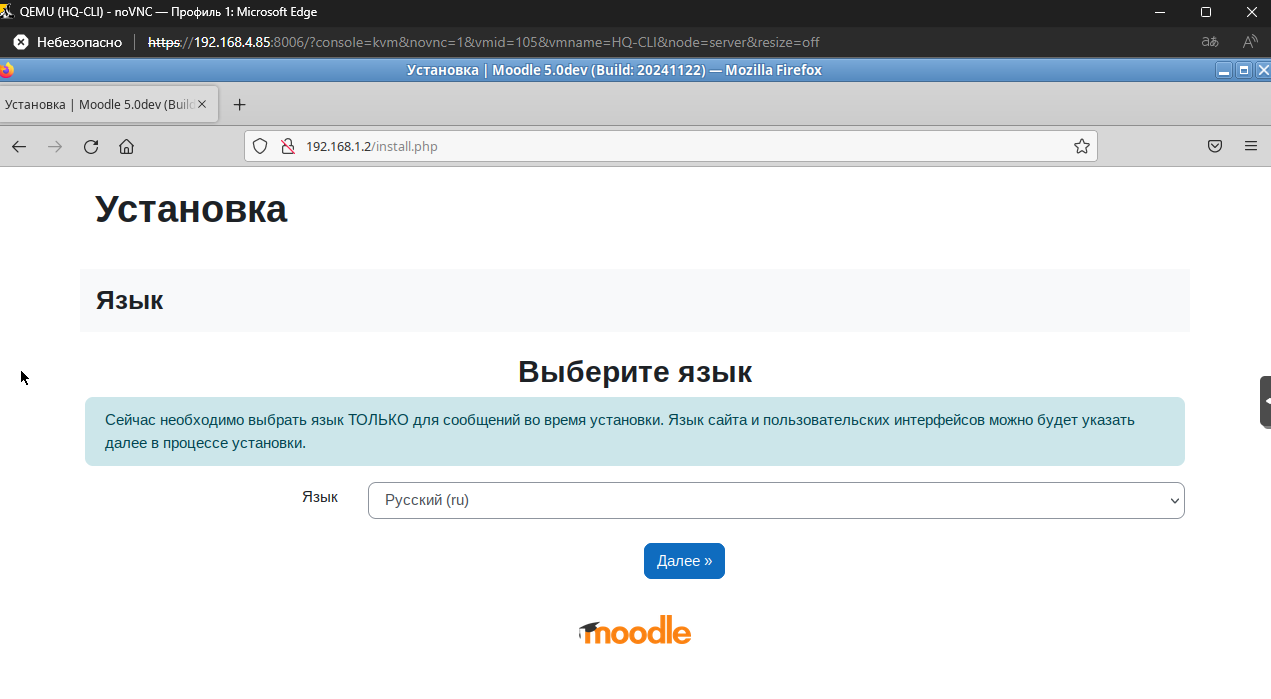
****

Рисунок 37

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

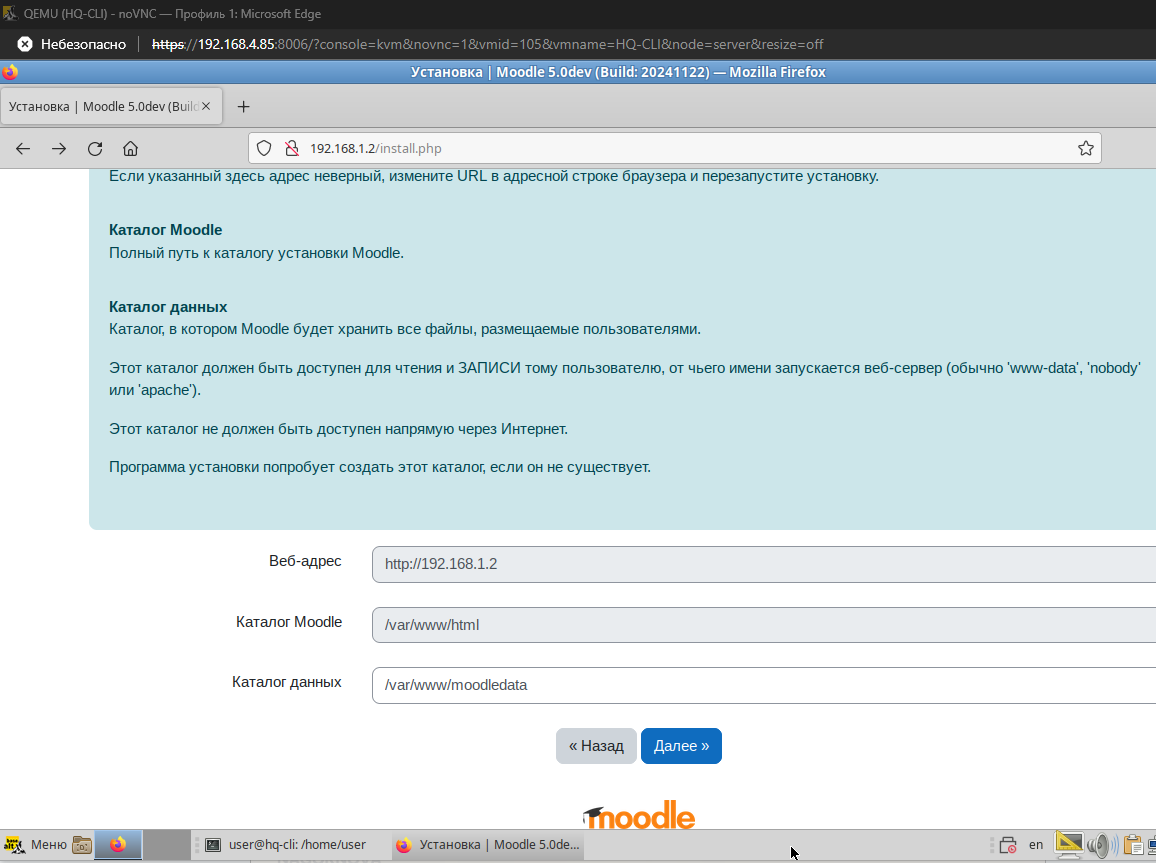
****

Рисунок 38

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

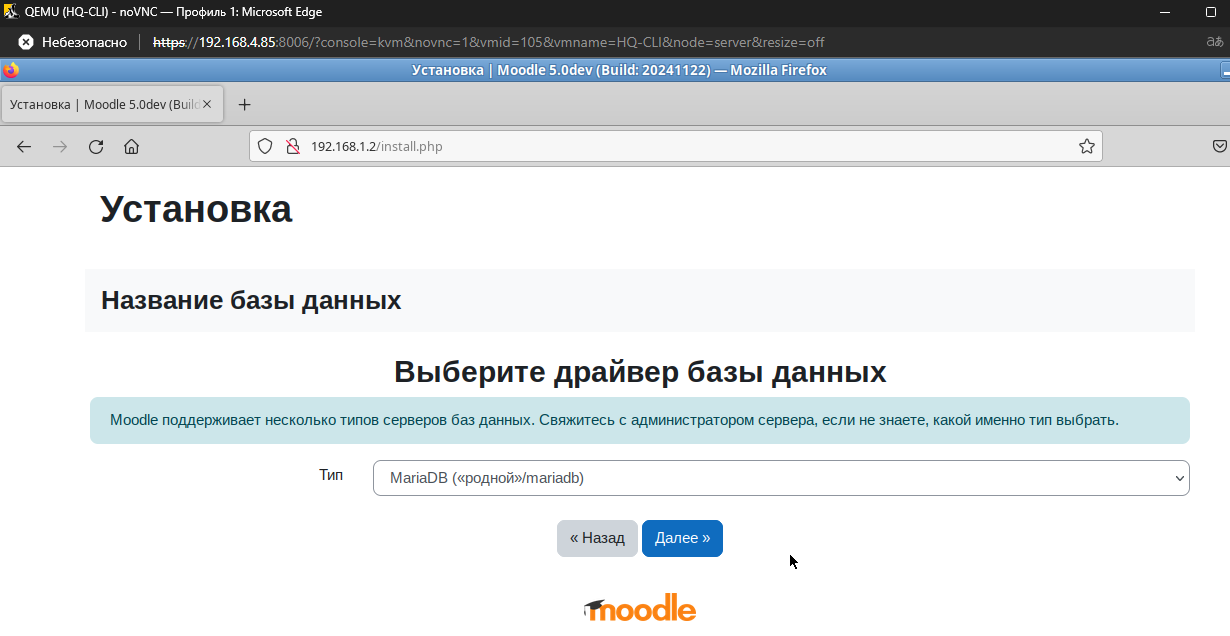
****

Рисунок 39

Введём нужные данные в следующие строки:

**Название базы данных: moodledb**

**Пользователь базы данных: moodle**

**Пароль: P@ssw0rd**

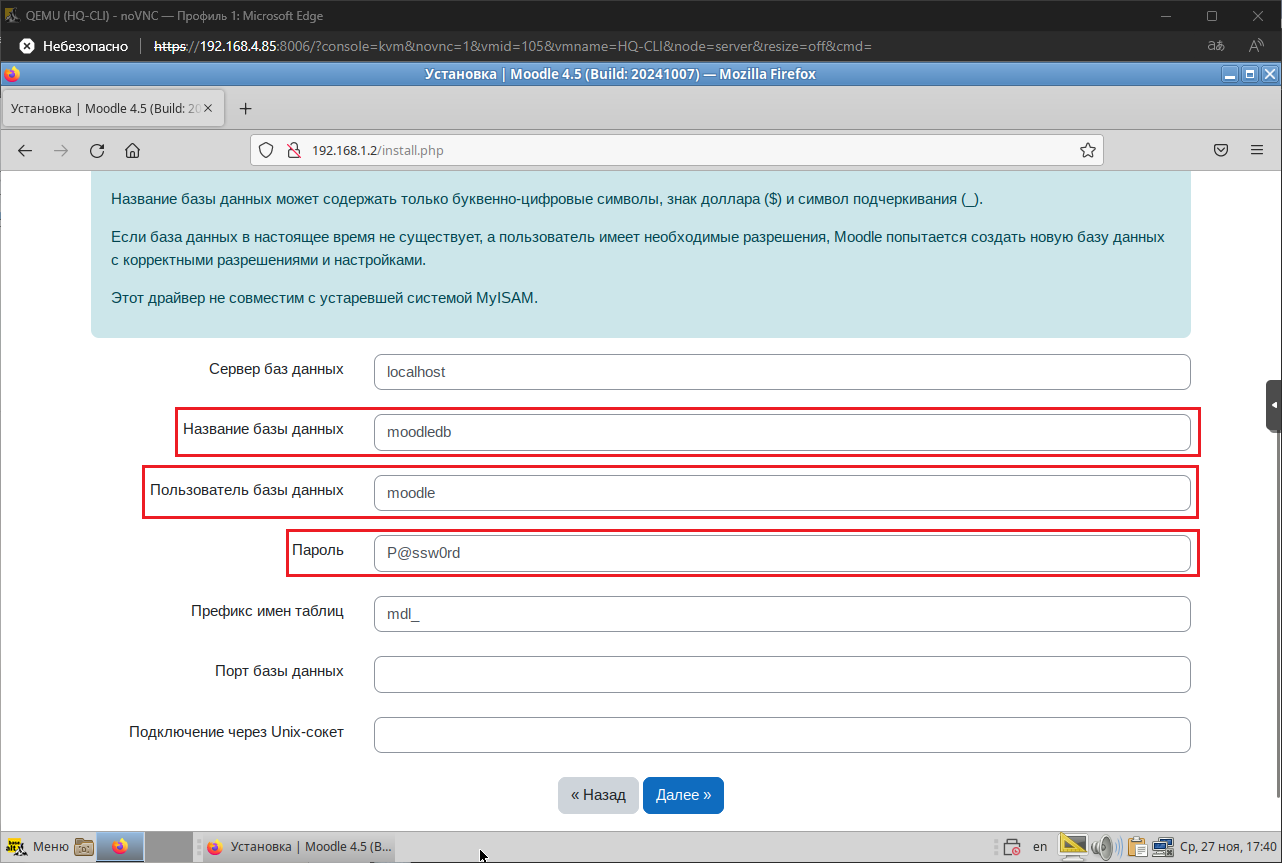


Рисунок 40

Нажимаем “**Продолжить**”:

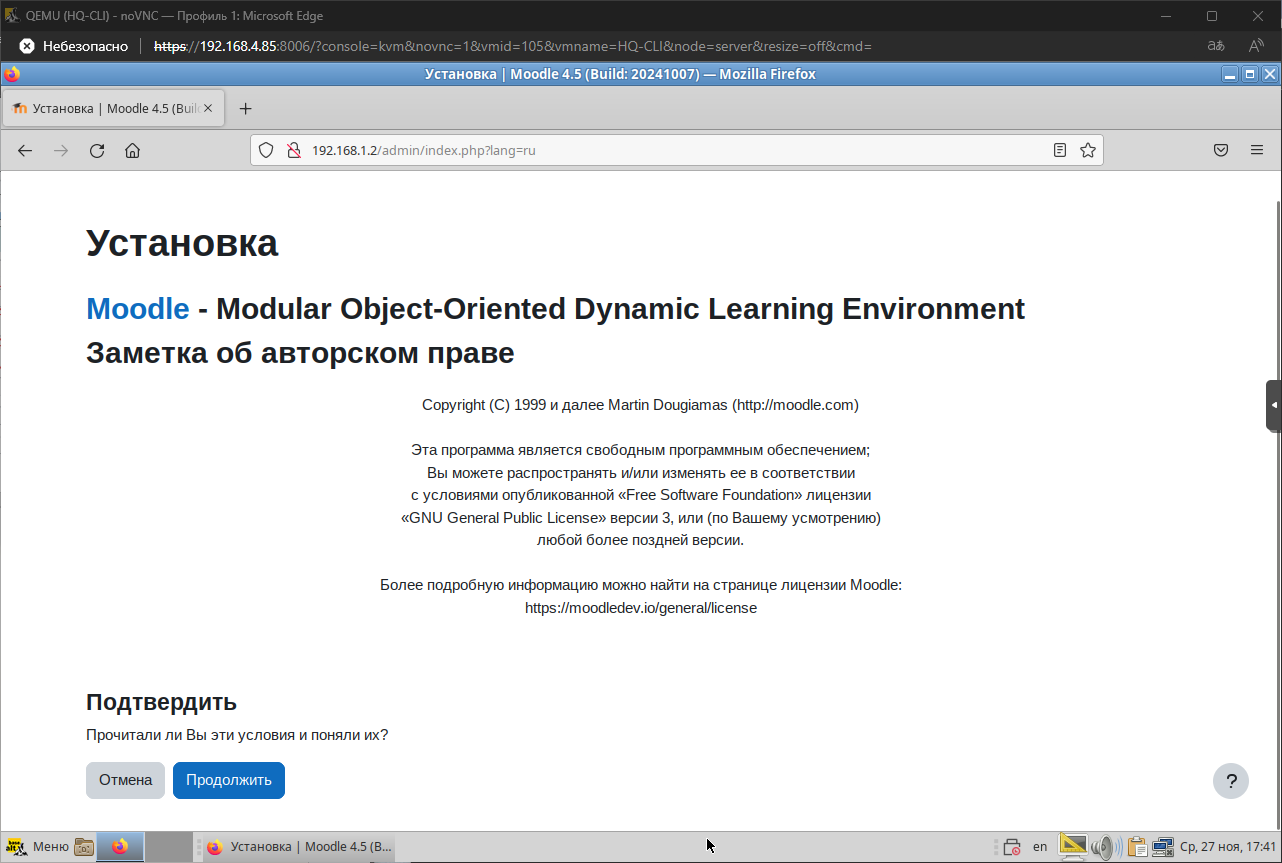


Рисунок 41

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

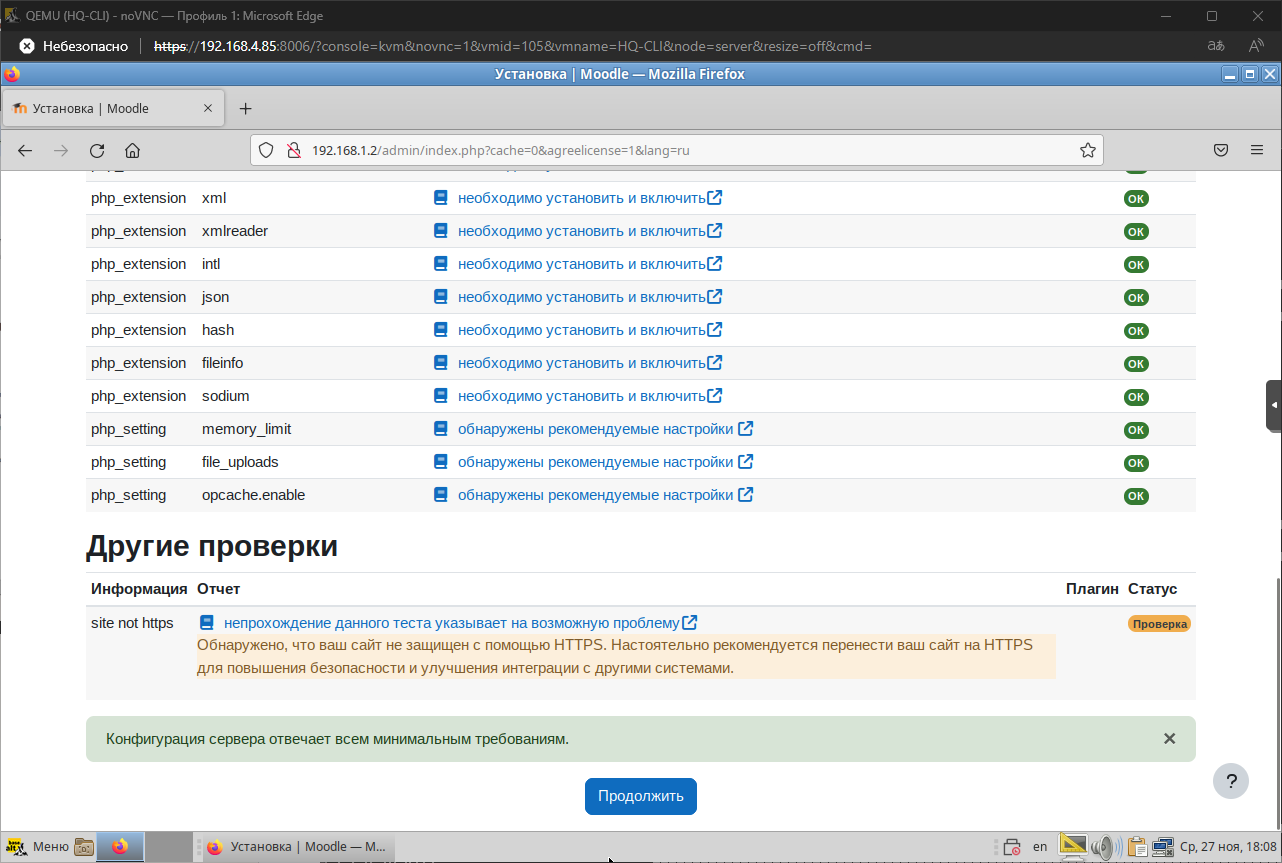


Рисунок 42

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

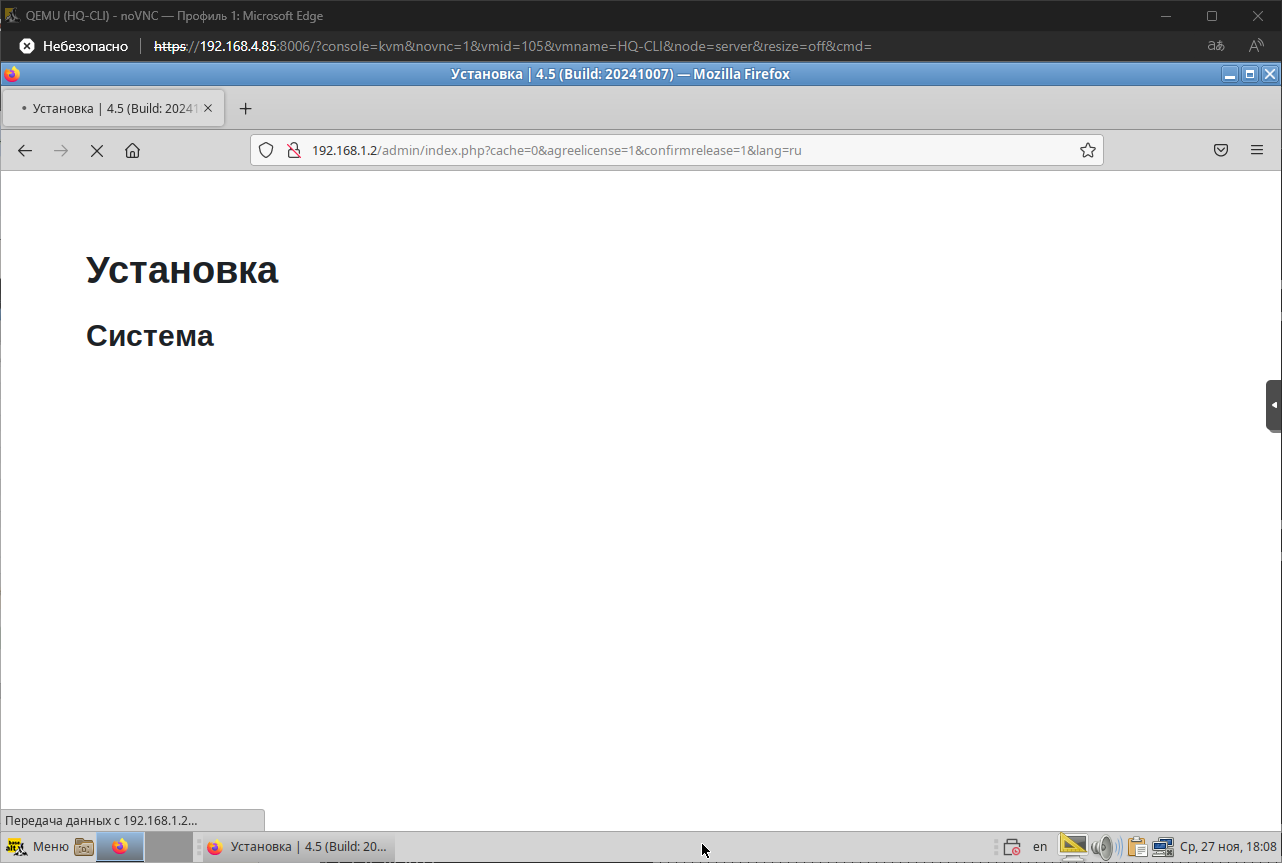


Рисунок 43

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

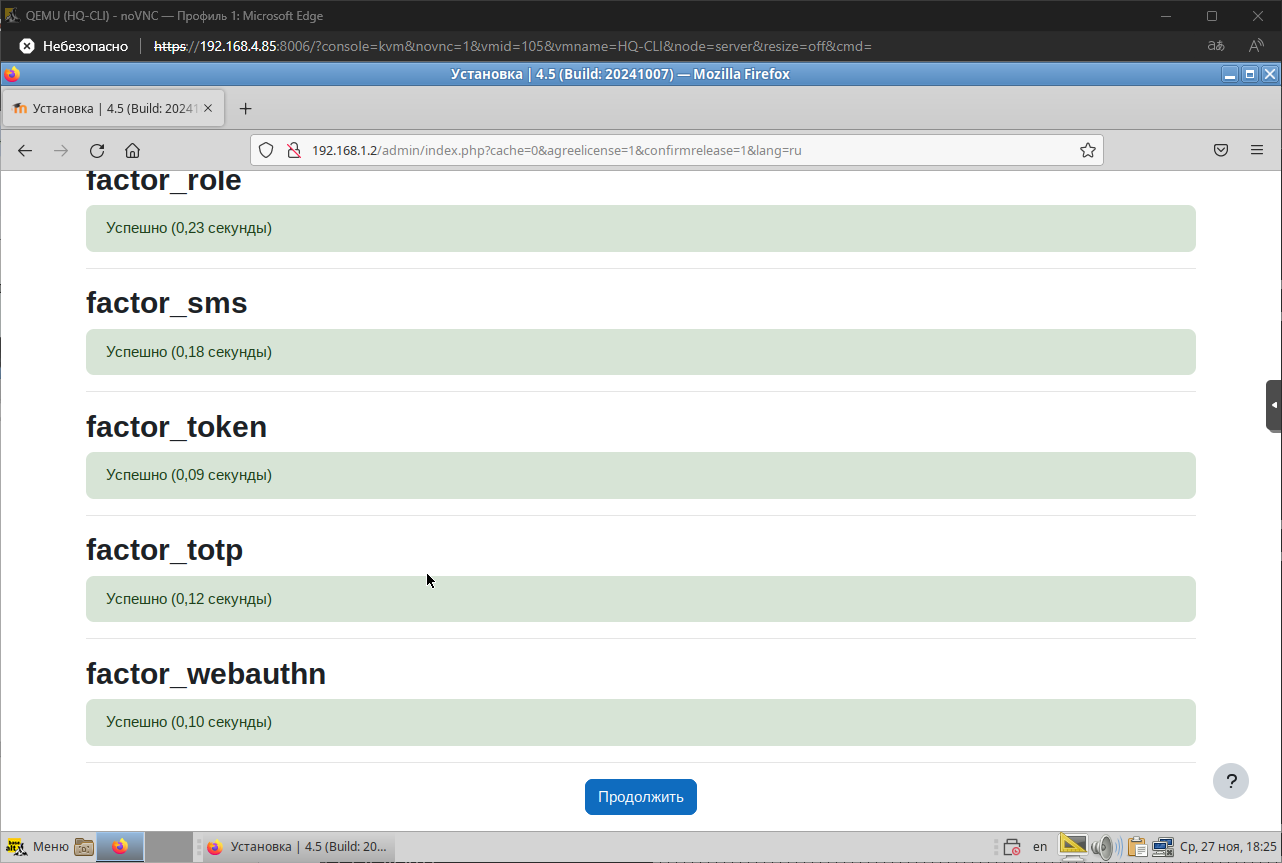


Рисунок 44

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.ru**](mailto:test@test.ru) **(**можно любое**)**

И нажимаем “**Обновить профиль**”:

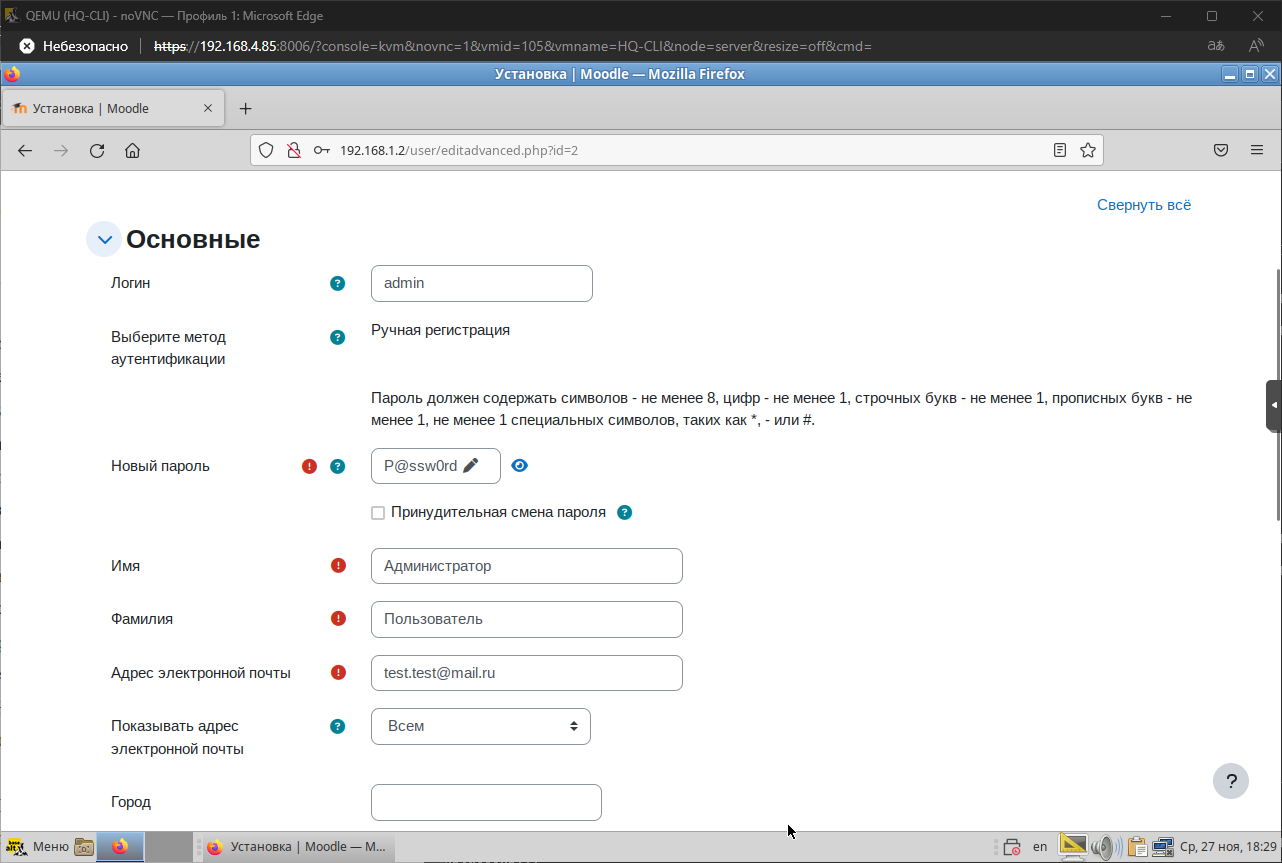


Рисунок 45

Теперь заполним ещё некоторые строки на следующем шаге:

**Полное название сайта: moodle-sa3-21** (можно любое**)**

**Краткое название сайта: moodle** (согласно вашему рабочему месту)

**Настройки местоположения: Азия/Красноярск**

**Контакты службы поддержки: test.test@mail.ru**

И жмём “**Сохранить изменения**” в конце страницы:

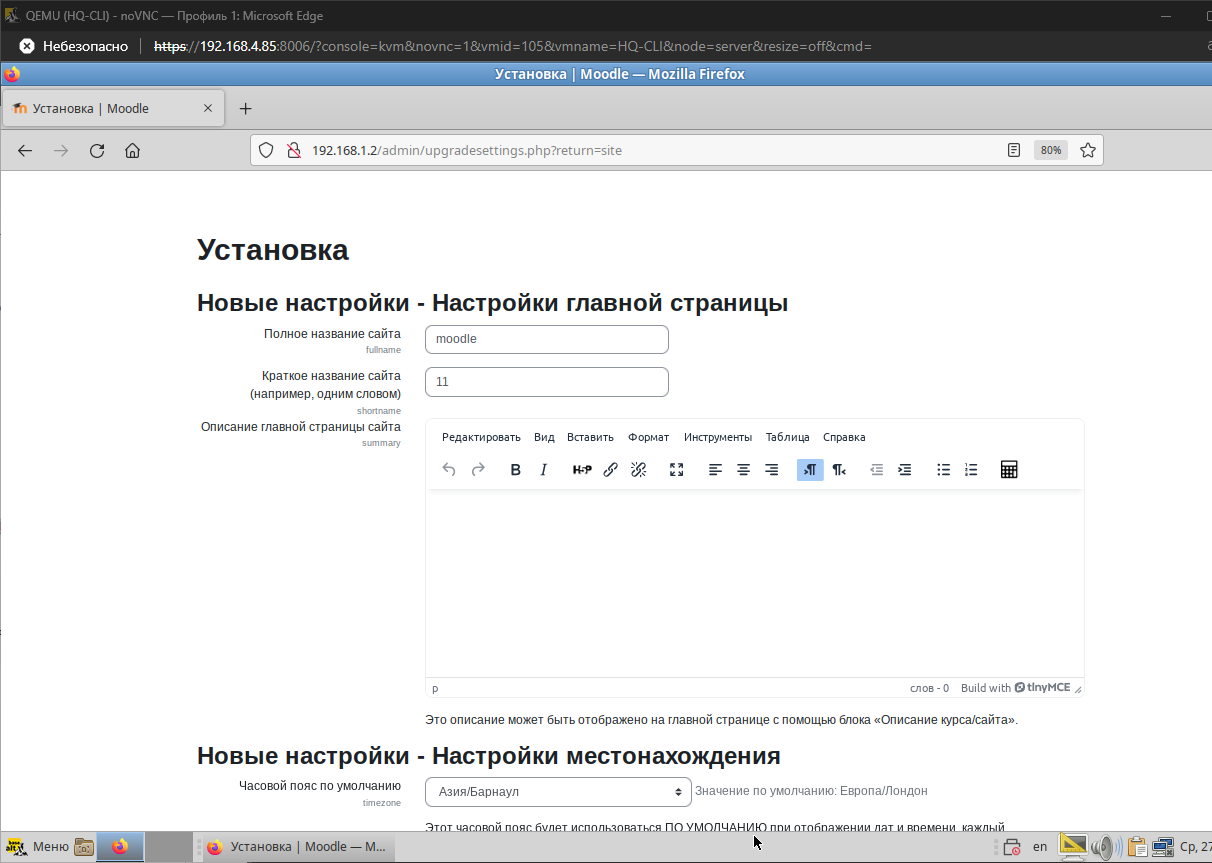


Рисунок 46

И после всего нас встречает рабочий сайт **moodle**, смотрим, что все наши указанные параметры отображаются:

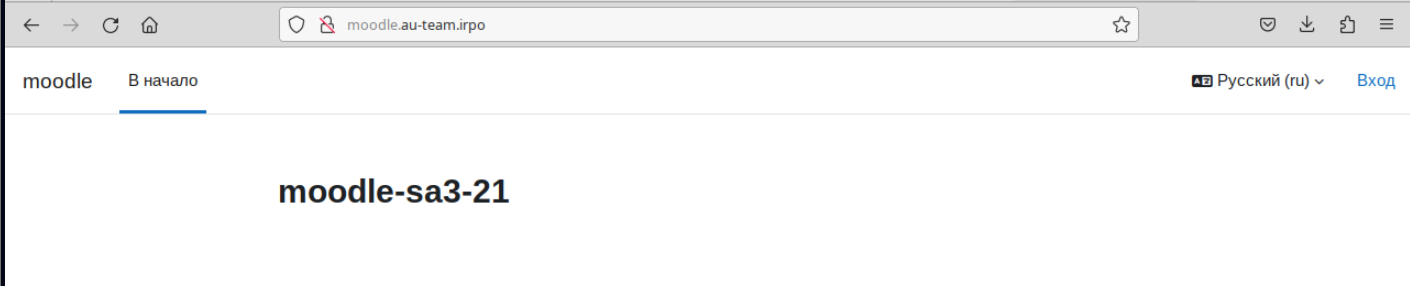
****

Рисунок 47

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

Поменяем значение **wwwroot** в конфигурации moodle на **HQ-SRV**:

**vim /var/www/html/config.php**

**$CFG->wwwroot = ‘http://sash’;**

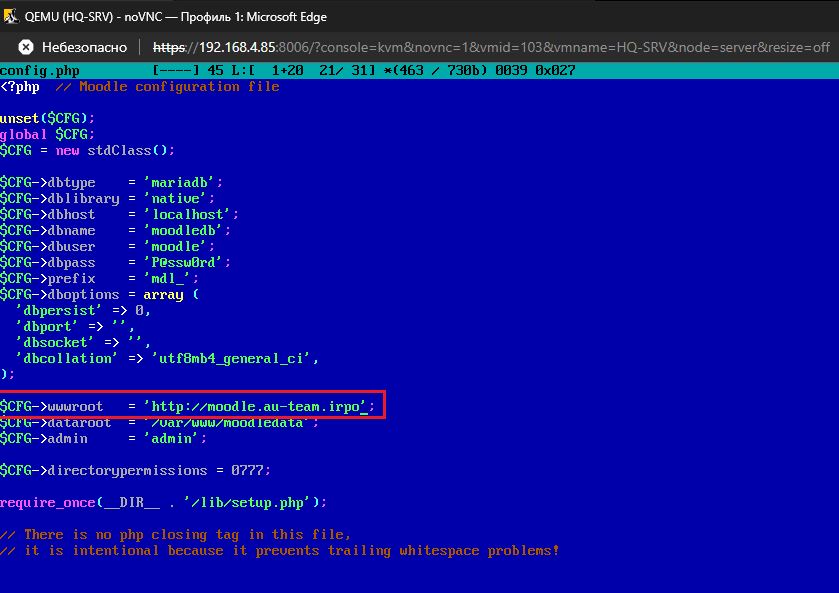


Рисунок 48

Устанавливаем пакет **nginx** на **HQ-RTR** для дальнейшей настройки:

**apt-get install nginx -y**

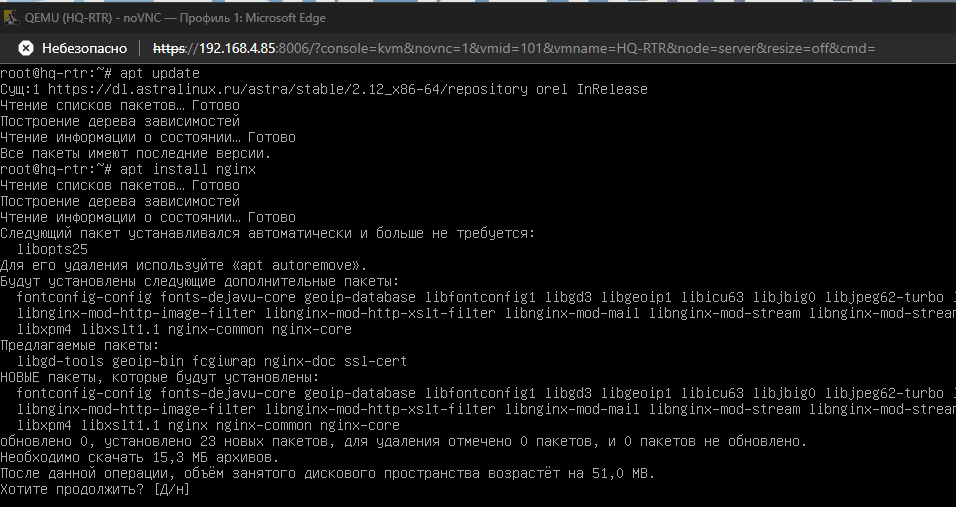
****

Рисунок 48

Создатель сервиса рекомендует использовать отдельные файлы для каждого проксирования сервиса. Они расположены **/etc/nginx/sites-enabled.d/**

**vim /etc/nginx/sites-enabled.d/moodle.conf – для сервиса moodle**

**server {**

**listen 80;**

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

**location / {**

**proxy\_pass http://192.168.1.10:80;**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $remote\_addr;**

**}**

**}**

**vim /etc/nginx/sites-enabled.d/wiki.conf для сервиса wiki**

**server {**

**listen 80;**

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

**location / {**

**proxy\_pass http://192.168.3.10:8080;**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $remote\_addr;**

**}**

**}**

Иногда Альт багается и нет данной строчки, но обычно есть

**vim /etc/nginx/nginx.conf**

**include /etc/nginx/sites-enabled.d/\*.conf; -** ссылка на наши файлы

**systemctl restart nginx**

**systemctl enable --now nginx**

Проверка представлена выше. На пунктах wiki и moodle, если посмотреть внимательно видно, что прокси работает и обращение происходит по имени

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

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

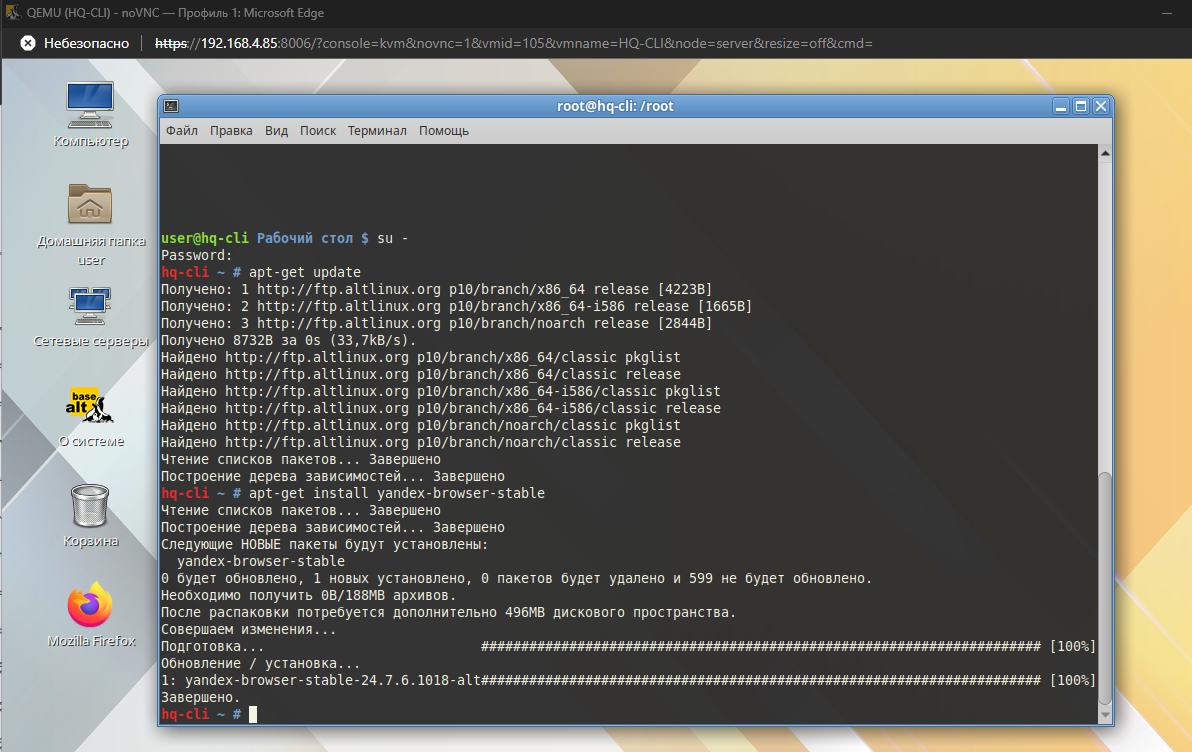
****

Рисунок 49

Видим, что установка завершена, теперь запустим его из меню **Пуск**.

Должен быть по пути:

**Пуск → Интернет → Yandex Browser**

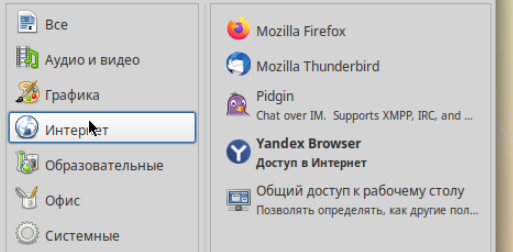
****

Рисунок 50

**Не ужели это конец!?**

**КОНЕЦ 2 МОДУЛЯ**

# **Модуль 3**

Не могу оставить любимых членов жюри без 3 модуля, продолжаем продолжать

**1. Выполните миграцию на новый контролдомена BR-SRV с HQ-SRV**

На машинке BR-SRV создаем папку где будут храниться скрипты иммиграции

mkdir samba\_backup

cd samba\_backup

**Первый скрипт будет сохранять конфигурацию нашего домен-контроллера**

vim samba-migrare-backup.sh

#!/bin/bash

backup() {

echo "[\*] Создание резервной копии Samba-DC..."

mkdir -p "$BACKUP\_DIR"

echo "[\*] Остановка Samba..."

systemctl stop samba

echo "[\*] Архивация конфигурации и базы данных..."

tar -czvf "$BACKUP\_FILE" /var/lib/samba /etc/samba /var/cache/samba /var/log/samba

echo "[\*] Запуск Samba обратно..."

systemctl start samba

echo "[?] Резервная копия создана: $BACKUP\_FILE"

}

**Второй скрипт выполняет копирование на второй сервер HQ-SRV**

vim samba-migrare-copy.sh

#!/bin/bash

copy() {

if [ -z "$2" ]; then

echo "[!] Укажите целевой сервер, пример: ./samba-migrate.sh copy root@server2"

exit 1

fi

echo "[\*] Копирование последнего архива на сервер: $2"

LAST\_BACKUP=$(ls -t "$BACKUP\_DIR"/samba\_backup\_\*.tar.gz | head -n1)

scp -P 2024 "$LAST\_BACKUP" "$2:/home/sshuser/"

echo "[?] Копирование завершено."

}

Обязательно делаем скрипты исполняемые

chmod +x samba-migrare-backup.sh

chmod +x samba-migrare-copy.sh

На втором сервере создаем такую же папку

mkdir samba\_backup

cd samba\_backup

Пишим скрипт

vim samba\_backup\_restore.sh

#!/bin/bash

restore() {

echo "[\*] Поиск последнего архива..."

BACKUP\_FILE=$(ls -t /home/sshuser/samba\_backup\_\*.tar.gz 2>/dev/null | head -n1)

if [ ! -f "$BACKUP\_FILE" ]; then

echo "[!] Архив не найден в /root"

exit 1

fi

echo "[\*] Остановка Samba..."

systemctl stop samba

echo "[\*] Удаление старых конфигураций..."

rm -rf /var/lib/samba /etc/samba /var/cache/samba /var/log/samba

echo "[\*] Распаковка архива..."

tar -xzvf "$BACKUP\_FILE" -C /

echo "[\*] Восстановление прав..."

chown -R root:root /etc/samba

chown -R root:root /var/log/samba

echo "[\*] Запуск Samba..."

systemctl start samba

echo "[?] Восстановление завершено."

}

Тоже делаем его исполняемым

**2. Выполните настройку центра сертификации на базе HQ-SRV**

apt-get install easy-rsa openssl -y #установка пакета

**cd /usr/share/easyrsa3/- директория в который искать ключи (pki)**

easyrsa init-pki #Инициализируйте инфраструктуру PKI

Настройки параметра сертификата

vim vars.example

set\_var EASYRSA\_REQ\_COUNTRY "RU"

set\_var EASYRSA\_REQ\_PROVINCE "KRSK"

set\_var EASYRSA\_REQ\_CITY "KRSK"

set\_var EASYRSA\_REQ\_ORG "Demo"

set\_var EASYRSA\_REQ\_EMAIL "user@examle.com"

set\_var EASYRSA\_REQ\_OU "au-team.irpo"

set\_var EASYRSA\_CA\_EXPIRE 3650

set\_var EASYRSA\_CERT\_EXPIRE 365

Генерация корневого сертификата

easyrsa build-ca nopass

Генерация ключей для сервисов

easyrsa gen-req wiki.au-team.irpo nopass

easyrsa sign-req server wiki.au-team.irpo

yes

easyrsa gen-req moodle.au-team.irpo nopass

easyrsa sign-req server moodle.au-team.irpo

yes

easyrsa gen-req nginx.au-team.irpo nopass

easyrsa sign-req server nginx.au-team.irpo

yes

scp /usr/share/easyrsa3/pki/ca.crt user@192.168.2.4:/home/user

ssh user@192.168.2.4

На cli

su -

cp /home/user/ca.crt /usr/share/ca-certificates/

update-ca-trust

На HQ-SRV

scp -P 2024 /usr/share/easyrsa3/pki/issued/moodle.au-team.irpo.crt user@192.168.1.1:/home/user

scp -P 2024 /usr/share/easyrsa3/pki/private/moodle.au-team.irpo.key user@192.168.1.1:/home/user

scp -P 2024 /usr/share/easyrsa3/pki/issued/wiki.au-team.irpo.crt user@192.168.1.1:/home/user

scp -P 2024 /usr/share/easyrsa3/pki/private/wiki.au-team.irpo.key user@192.168.1.1:/home/user

ssh user@192.168.1.1 -p 2024

На HQ-RTR

su –

mkdir /etc/openssl/certs/

mkdir /etc/openssl/private/

mv /home/user/wiki.au-team.irpo.crt /etc/openssl/certs/

mv /home/user/moodle.au-team.irpo.key /etc/openssl/private/

mv /home/user/moodle.au-team.irpo.crt /etc/openssl/certs/

mv /home/user/wiki.au-team.irpo.key /etc/openssl/private/

**Перенастраиваем nginx**

vim /etc/nginx/sites-enabled.d/moodle.conf

listen 443 ssl;

ssl\_certificate /etc/openssl/certs/moodle.au-team.irpo.crt;

ssl\_certificate\_key /etc/openssl/private/moodle.au-team.irpo.key;

vim /etc/nginx/sites-enabled.d/wiki.conf

listen 443 ssl;

ssl\_certificate /etc/openssl/certs/wiki.au-team.irpo.crt;

ssl\_certificate\_key /etc/openssl/private/wiki.au-team.irpo.key;

На сli добавляем сертификат в firefox

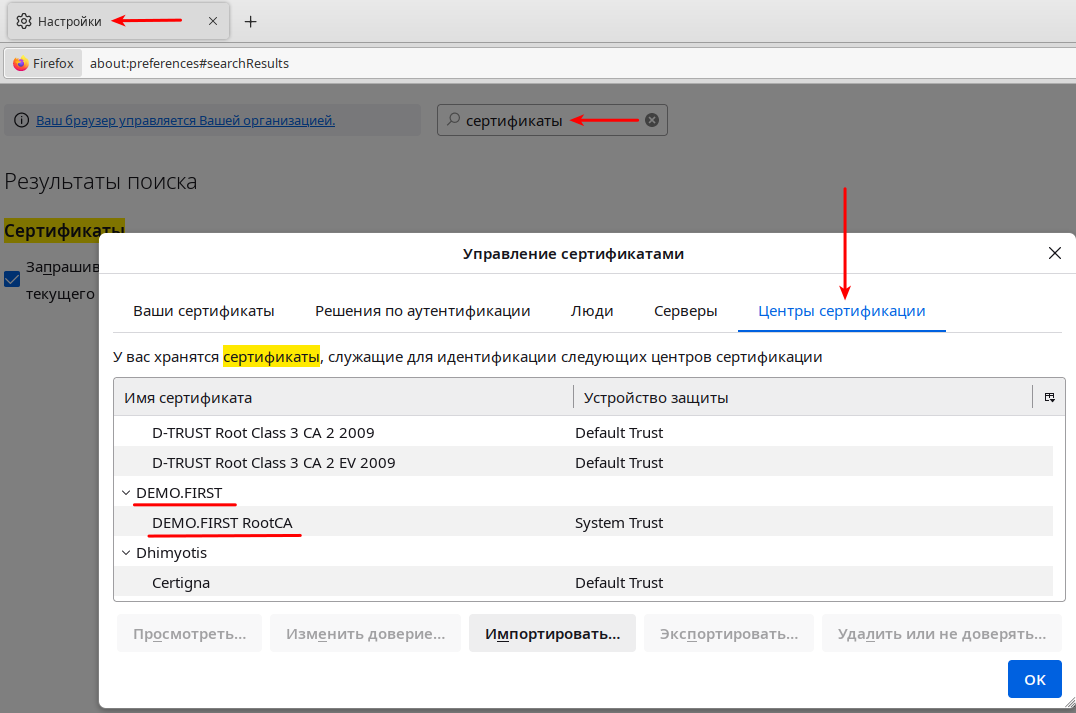


Рисунок 51

Проверяем moodle.au-team.irpo и wiki.au-team.irpo

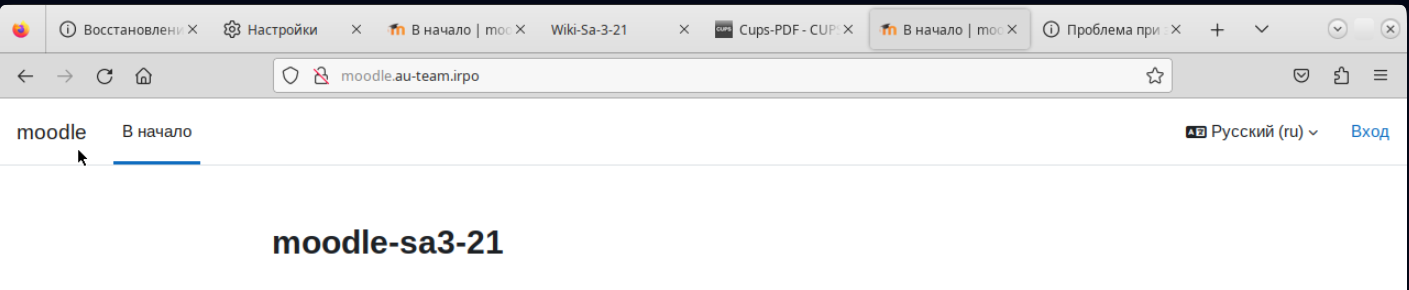
**** ****

Рисунок 52

**3. Перенастройте ip-туннель с базового до уровня туннеля, обеспечивающего шифрование трафика**

Задание выполнено в 1 модуле

**4. Настройте межсетевой экран на маршрутизаторах HQ-RTR и BR-RTR на сеть в сторону ISP**

**Добавил следующие правила**

iptables -A INPUT -i ens18 -p tcp -m multiport --dports 22,53,80,443,2024 -j ACCEPT

iptables -A INPUT -i ens18 -p udp -m multiport --dports 53,123,500,4500 -j ACCEPT

iptables -A INPUT -i ens18-p icmp -j ACCEPT

iptables -A INPUT -p esp -j ACCEPT

iptables -A FORWARD -p esp -j ACCEPT

iptables -A INPUT -i ens18 -m state --state NEW -j DROP

iptables -A OUTPUT -j ACCEPT

iptables-save > /etc/sysconfig/iptables

**5. Настройте принт-сервер cups на сервере HQ-SRV.**

Для начала необходимо установить пакеты cups и cups-pdf на **HQ-SRV:**

**apt-get install cups cups-pdf -y**

Теперь необходимо включить службу cups, чтобы она запускалась вместе с системой.

**systemctl enable –now cups**

Далее, необходимо отредактировать конфиг **vim /etc/cups/cupsd.conf**

Во всех блоках **Location** необходимо добавить строку **Allow all**, как на скриншоте:

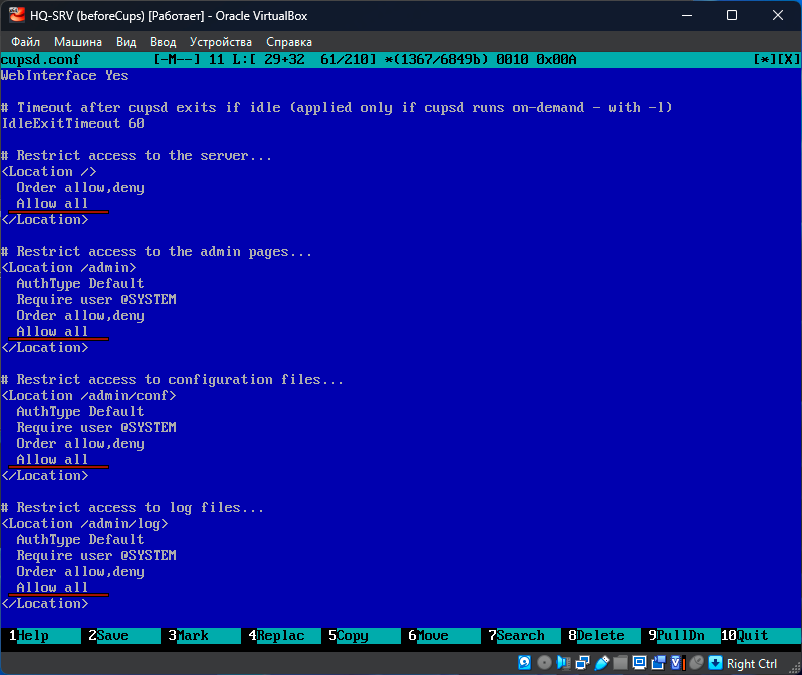


Рисунок 53

Перезапускаем службу **cups** для применения изменений:

**systemctl restart cups**

Переходим к подключению клиента **HQ-CLI**

На **HQ-CLI** выполняем следующую команду для подключения к принт-серверу:

**lpadmin -p CUPS -E -v ipp://hq-srv.au-team.irpo:631/printers/Cups-PDF -m everywhere**

Установим принтер CUPS, как принтер по умолчанию:

**lpoptions -d CUPS**

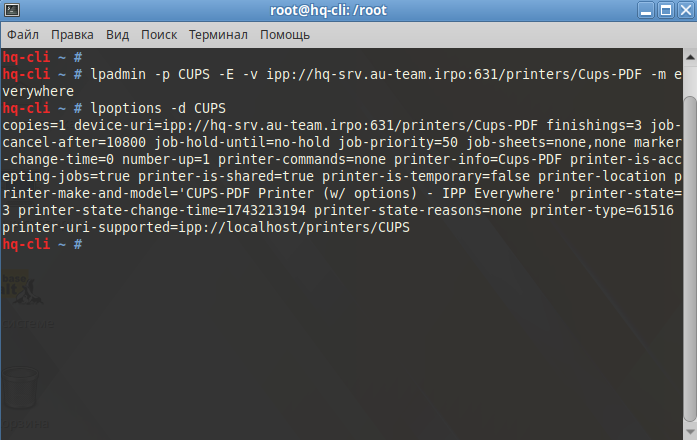
****

Рисунок 54

Проверяем наличие принтера командой **lpstat -p**

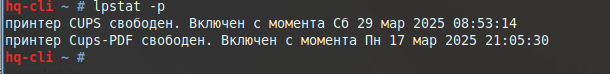
****

Рисунок 55

Как можно заметить, принтер **CUPS** успешно подключен. Из-за того, что на **HQ-CLI** также установлен принт-сервер, можно отключить локальный принтер “**Cups-PDF**”, чтобы он не мешал.

**lpadmin -x Cups-PDF**

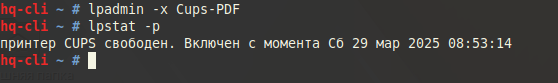
****

Рисунок 56

Теперь у нас остался один принтер. Проверим его работу. Откроем любой текстовый документ и попробуем его распечатать.

Допустим, откроем файл **/etc/altlinux-release**, нажимаем сверху значок печати:

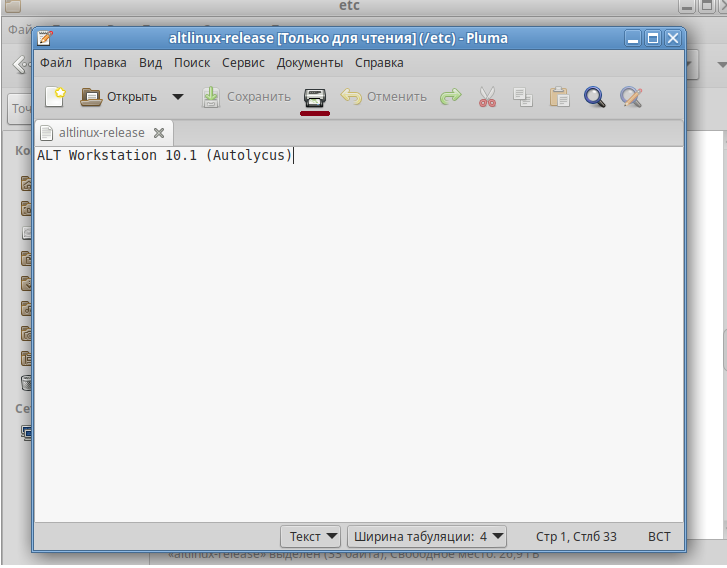
****

Рисунок 57

Выбираем наш принтер **CUPS** и жмем **Печать**

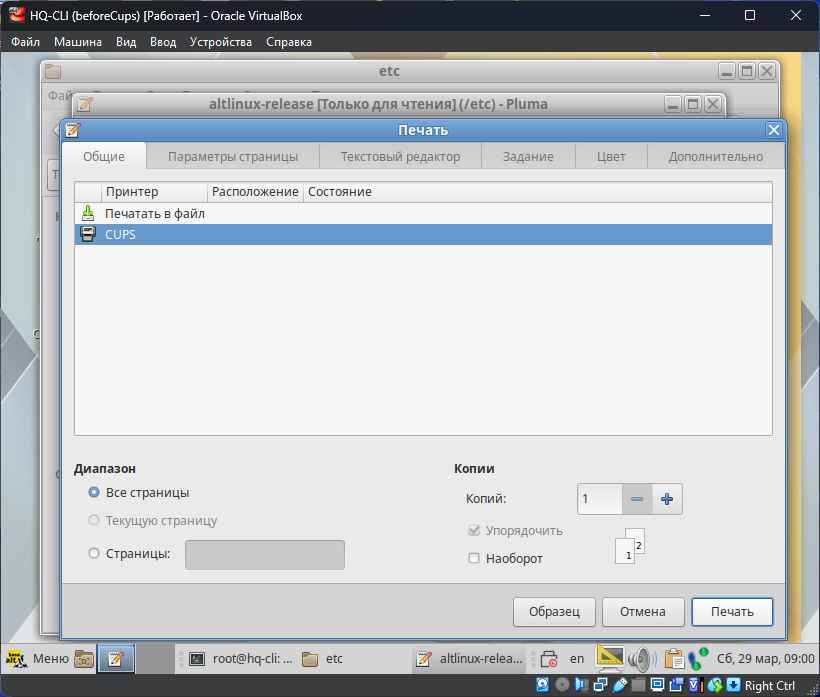
****

Рисунок 58

Сверху появится уведомление, что документ успешно напечатан.

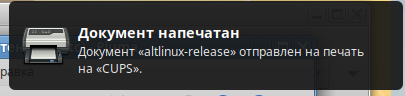


Рисунок 59

Перейдем в веб-интерфейс CUPS по адресу <https://hq-srv.au-team.irpo:631>

Вкладка **Принтеры**

Выбираем наш принтер.

Жмем кнопку **Показать все задания**

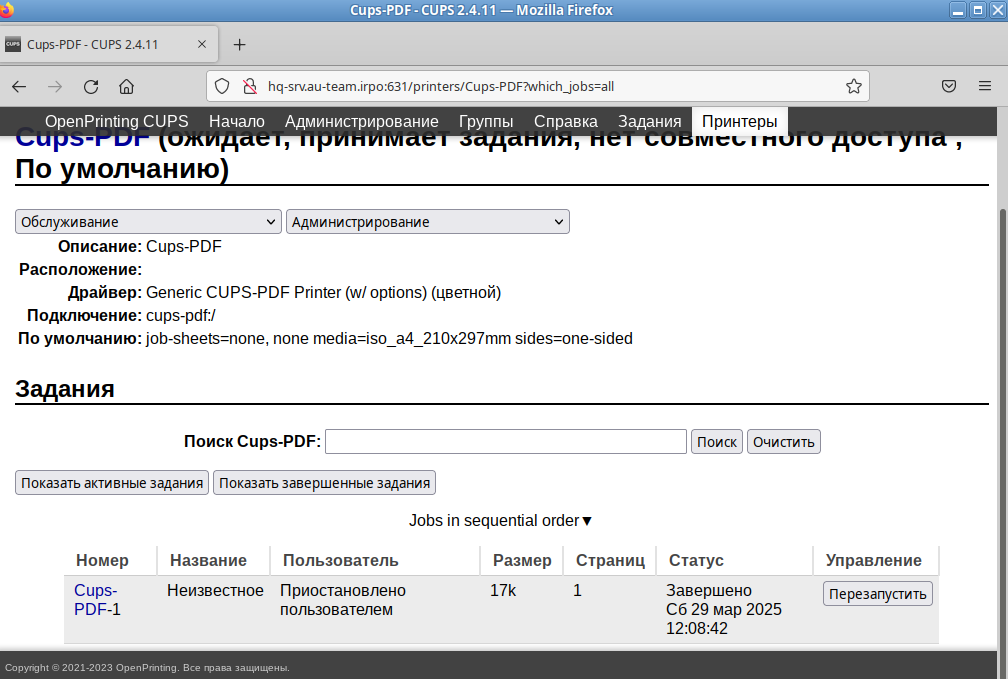
****

Рисунок 60

**6. Реализуйте логирование при помощи rsyslog на устройствах HQ-RTR, BR-RTR, BR-SRV**

apt-get install -y rsyslog-classic

vim /etc/rsyslog.d/00\_common.conf

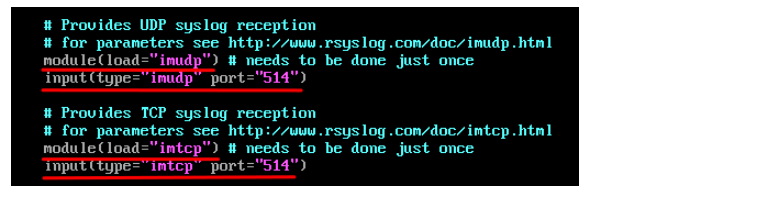


Рисунок 61

В конце  
**$template RemoteLogs, "/opt/%HOSTNAME%/rsyslog.txt"**

**\*.\* ?RemoteLogs**

**& stop**

На клиентах

apt-get install -y rsyslog-classic

**echo "\*.warning @@192.168.1.10:514" > /etc/rsyslog.d/all\_log.conf -ОБРАТИТЕ ВНИМАНИЕ**

**vim /etc/rsyslog.d/00\_common.conf**

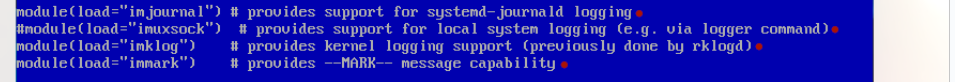


Рисунок 62

Включаем все модули

**systemctl enable rsyslog**

**systemctl restart rsyslog**

Добавим несколько записей различного уровня в лог на любом из клиентов, например на **BR-SRV**, командами:

**logger -p user.info “Test info”**

Также добавим сообщения уровня **warning:**

**logger -p user.warning “Test warning”**

Также добавим сообщения уровня **error:**

**logger -p user.error “Test error”**

Как можно заметить, здесь появились только сообщения уровня **warning** и **error**.

Перейдем к настройке ротации логов. На **HQ-SRV** создадим файл **vim** **/etc/logrotate.d/rsyslog**

Запишем в него следующее содержимое:

**/opt/\*/rsyslog.txt {**

**weekly**

**minsize 10M**

**compress**

**missingok**

**notifempty**

**create 0644 root root**

**rotate 4**

**dateext**

**}**

**7. На сервере HQ-SRV реализуйте мониторинг устройств с помощью открытого программного обеспечения. Обеспечьте доступность по URL -** [**https://mon.au-team.irpo**](https://mon.au-team.irpo)

Для мониторинга был выбран сервис забикс, постален докер. НО в связи с ограниченными характеристиками виртуалками. Он при запуске ее ложит, прилагаю только конфиг

version: '3'

services:

zabbix-server:

image: zabbix/zabbix-server-pgsql:latest

environment:

- DB\_SERVER\_HOST=zabbix-postgres

- POSTGRES\_USER=zabbix

- POSTGRES\_PASSWORD=secret

ports:

- "10051:10051"

zabbix-web-apache:

image: zabbix/zabbix-web-apache-pgsql:latest

environment:

- DB\_SERVER\_HOST=zabbix-postgres

- POSTGRES\_USER=zabbix

- POSTGRES\_PASSWORD=secret

- ZBX\_SERVER\_HOST=zabbix-server

ports:

- "8080:8080"

zabbix-postgres:

image: postgres:latest

environment:

- POSTGRES\_USER=zabbix

- POSTGRES\_PASSWORD=secret

**8. Реализуете механизм инвентаризации машин через Ansible на BR-SRV:**

Создаю файл инвентори и вношу следующий конфиг

**vim /etc/ansible/inventory.yml**

---

- name: Инвентаризация машин HQ-SRV и HQ-CLI

hosts:

- hq-srv

- hq-cli

gather\_facts: yes

tasks:

- name: Создать отчёт с информацией рабочем месте

delegate\_to: localhost

copy:

dest: "/etc/ansible/PC\_INFO/{{ ansible\_hostname }}.yml"

content: |

---

Имя компьютера: "{{ ansible\_hostname }}"

IP-адрес компьютера: "{{ ansible\_default\_ipv4.address }}"



Рисунок 63

mkdir **/etc/ansible/PC\_INFO**

**ansible-playbook /etc/ansible/inventory.yml**

Для проверки

**cat /etc/ansible/PC\_INFO/hq-cli.yml**

**cat /etc/ansible/PC\_INFO/hq-srv.yml**

**ВАЖНО ОТМЕТИТЬ. Сбор с машинки cli не работает, ПРОБЛЕМА ИМЕННО В МАШИНКИ КЛИ, КАК Я ПИСАЛ РАНЕЕ.**

**9. Реализуйте механизм резервного копирования конфигурации для машин HQ-RTR и BR-RTR, через Ansible на BR-SRV:**

vim /etc/ansible/backup.yml

---

- name: Collect network configurations from routers

hosts: hq-rtr,br-rtr

gather\_facts: no

tasks:

- name: Ensure local target directory exists

ansible.builtin.file:

path: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/{{ item }}"

state: directory

loop:

- "frr"

- "sysconfig"

delegate\_to: localhost

- name: Transfer FRR files to control node

ansible.builtin.fetch:

src: "/etc/frr/{{ item }}"

dest: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/frr/"

flat: yes

loop:

- "daemons"

- "frr.conf"

- "frr.conf.sav"

- "vtysh.conf"

become: yes

- name: Transfer iptables configuration

ansible.builtin.fetch:

src: "/etc/sysconfig/{{ item }}"

dest: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/sysconfig/"

flat: yes

loop:

- "iptables"

- "iptables\_modules"

- "iptables\_params"

become: yes

- name: Interfaces archive

ansible.builtin.archive:

path: /etc/net/ifaces

dest: /tmp/ifaces-{{ inventory\_hostname }}.tar.gz

format: gz

- name: Transfer interfaces archive to control node

ansible.builtin.fetch:

src: "/tmp/ifaces-{{ inventory\_hostname }}.tar.gz"

dest: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/ifaces.tar.gz"

flat: yes

- name: Cleanup remote interfaces archive

ansible.builtin.file:

path: "/tmp/ifaces-{{ inventory\_hostname }}.tar.gz"

state: absent

- name: Extract interfaces configuration locally

ansible.builtin.unarchive:

src: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/ifaces.tar.gz"

dest: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/"

delegate\_to: localhost

- name: Cleanup local archive

ansible.builtin.file:

path: "/etc/ansible/NETWORK\_INFO/{{ inventory\_hostname }}/ifaces.tar.gz"

state: absent

delegate\_to: localhost

**ansible-playbook /etc/ansible/backup.yml – запуск**