Гайд

# Базовая настройка оборудования

## Настойка машины ISP

apt-get update

hostnamectl set-hostname isp.au-team.irpo; exec bash

**Настройка переадресации**

vim /etc/net/sysctl.conf

#net.ipv4.ip\_forward = 0 >> #net.ipv4.ip\_forward = 1

**Настройка адресов**

mkdir /etc/net/ifaces/ens19

echo 172.16.4.1/28 > /etc/net/ifaces/ens19/ipv4address

cp /etc/net/ifaces/ens18/options /etc/net/ifaces/ens19

vim /etc/net/ifaces/ens19/options

mkdir /etc/net/ifaces/ens20

echo 172.16.5.1/28 > /etc/net/ifaces/ens20/ipv4address

cp /etc/net/ifaces/ens19/options /etc/net/ifaces/ens20

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner

vim /etc/openssh/sshd\_config

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2

Banner /etc/openssh/banner

**Перезапуск машины**

**Настройка Nat**

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

iptables-save > /etc/sysconfig/iptables

systemctl enable iptables.service

systemctl start iptables.service

## Настойка машины BR-RTR

hostnamectl set-hostname br-rtr.au-team.irpo; exec bash

**Настройкапереадресации**

vim /etc/net/sysctl.conf

#net.ipv4.ip\_forward = 0 >> #net.ipv4.ip\_forward = 1

**Настройка адресов**

echo 172.16.5.2/28 > /etc/net/ifaces/ens18/ipv4address - руками

echo default via 172.16.5.1 > /etc/net/ifaces/ens18/ipv4route

mkdir /etc/net/ifaces/ens19

echo 192.168.3.1/27 > /etc/net/ifaces/ens19/ipv4address

cp /etc/net/ifaces/ens18/options /etc/net/ifaces/ens19

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner

vim /etc/openssh/sshd\_config

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2

Banner /etc/openssh/banner

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

passwd net\_admin

P@ssw0rd

vim /etc/sudoers

net\_admin ALL=(ALL) NOPASSWD: ALL

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

usermod -aG wheel net\_admin

**Настройка frr**

systemctl restart network

apt-get update && apt-get install -y frr

vim /etc/frr/daemons

systemctl enable --now frr

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

rm -f /etc/net/ifaces/ens18/ipv4route – не надо

**Перезапуск машины**

apt-get update

**Настройка Nat**

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

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

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

iptables-save > /etc/sysconfig/iptables

systemctl enable --now iptables.service

systemctl start iptables.service

## Настойка машины HQ-RTR

hostnamectl set-hostname hq-rtr.au-team.irpo; exec bash

**Настройкапереадресации**

vim /etc/net/sysctl.conf

#net.ipv4.ip\_forward = 0 >> #net.ipv4.ip\_forward = 1

**Настройка адресов**

echo 172.16.4.2/28 > /etc/net/ifaces/ens18/ipv4address - руками

echo default via 172.16.4.1 > /etc/net/ifaces/ens18/ipv4route

mkdir /etc/net/ifaces/ens19

cp /etc/net/ifaces/ens18/options /etc/net/ifaces/ens19

mkdir /etc/net/ifaces/ens19.100

vim /etc/net/ifaces/ens19.100/options

BOOTPROTO=static

TYPE=vlan

HOST=ens19

VID=100

echo 192.168.1.1/26 > /etc/net/ifaces/ens19.100/ipv4address

mkdir /etc/net/ifaces/ens19.200

vim /etc/net/ifaces/ens19.200/options

BOOTPROTO=static

TYPE=vlan

HOST=ens19

VID=200

echo 192.168.2.1/28 > /etc/net/ifaces/ens19.200/ipv4address

mkdir /etc/net/ifaces/ens19.999

vim /etc/net/ifaces/ens19.999/options

BOOTPROTO=static

TYPE=vlan

HOST=ens19

VID=999

echo 192.168.99.1/29 > /etc/net/ifaces/ens19.999/ipv4address

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner

vim /etc/openssh/sshd\_config

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2

Banner /etc/openssh/banner

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

passwd net\_admin

P@ssw0rd

vim /etc/sudoers

net\_admin ALL=(ALL) NOPASSWD: ALL

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

usermod -aG wheel net\_admin

**Настройка frr**

apt-get update && apt-get install -y frr

vim /etc/frr/daemons

systemctl enable --now frr

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

rm -f /etc/net/ifaces/ens18/ipv4route

**Перезапуск машины**

apt-get update

**Настройка Nat**

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

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

iptables-save > /etc/sysconfig/iptables

systemctl enable --now iptables.service

systemctl start iptables.service

## Настойка машины HQ-SRV

hostnamectl set-hostname hq-srv.au-team.irpo; exec bash

**Настройка адресов**

echo 192.168.1.10/26 > /etc/net/ifaces/ens18/ipv4address - руками

echo default via 192.168.1.1 > /etc/net/ifaces/ens18/ipv4route

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner

vim /etc/openssh/sshd\_config

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2

Banner /etc/openssh/banner

Allowusers sshuser

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

passwd sshuser

P@ssw0rd

vim /etc/sudoers

sshuser ALL=(ALL) NOPASSWD: ALL

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

usermod -aG wheel sshuser

Перезапуск сервиса

**Перезапуск машины**

apt-get update

## Настойка машины BR-SRV

hostnamectl set-hostname br-srv.au-team.irpo; exec bash

**Настройка адресов**

echo 192.168.3.10/27 > /etc/net/ifaces/ens18/ipv4address - руками

echo default via 192.168.3.1 > /etc/net/ifaces/ens18/ipv4route

**Настройка ssh**

echo Autorized access only > /etc/openssh/banner

vim /etc/openssh/sshd\_config

#Port 22 >> Port 2024

#MaxAuthTries 6 >> MaxAuthTries 2

Banner /etc/openssh/banner

Allowusers sshuser

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

passwd sshuser

P@ssw0rd

vim /etc/sudoers

sshuser ALL=(ALL) NOPASSWD: ALL

WHEEL\_USERS ALL=(ALL:ALL) NOPASSWD: ALL (2 строчка)

usermod -aG wheel sshuser

Перезапуск сервиса

**Перезапуск машины**

apt-get update

## Настойка Туннеля между HQ-RTR и BR-RTR (3 модуль)

**Настройка на hq-rtr**

mkdir /etc/net/ifaces/iptunnel

vim /etc/net/ifaces/iptunnel/options

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

echo 192.168.3.0/24 via 10.0.0.2 > /etc/net/ifaces/iptunnel/ipv4route

systemctl restart network

apt-get install -y strongswan

**vim /etc/strongswan/ipsec.conf**

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

#10.0.0.1 10.0.0.2 : PSK "P@ssw0rd"

systemctl enable --now ipsec

ipsec restart

ipsec status

**Настройка на br-rtr**

mkdir /etc/net/ifaces/iptunnel

vim /etc/net/ifaces/iptunnel/options

TYPE=iptun

TUNTYPE=gre

TUNLOCAL=172.16.5.2

TUNREMOTE=172.16.4.2

TUNOPTIONS='ttl 64'

DISABLE=no

TUNMTU=1476

echo 10.0.0.2/30 > /etc/net/ifaces/iptunnel/ipv4address

vim /etc/net/ifaces/iptunnel/ipv4route

192.168.1.0/24 via 10.0.0.1

192.168.2.0/24 via 10.0.0.1

systemctl restart network

apt-get install -y strongswan

**vim /etc/strongswan/ipsec.conf**

conn gre

auto=start

type=tunnel

authby=secret

left=10.0.0.2

right=10.0.0.1

leftsubnet=0.0.0.0/0

rightsubnet=0.0.0.0/0

rightprotoport=gre

ike=aes256-sha2\_256-modp1024!

exitesp=aes256-sha2\_256!

vim /etc/strongswan/ipsec.secrets

#10.0.0.2 10.0.0.1 : PSK "P@ssw0rd"

systemctl enable --now ipsec

ipsec restart

ipsec status

Проверка:

**apt-get install tcpdump -y**

**tcpdump -i ens18 -n -p esp**

**ping 10.0.0.2 – c hq-rtr**

## Настойка DHCP сервера на HQ-RTR

apt-get update

apt-get install dnsmasq **-y**

systemctl enable --now dnsmasq

vim /etc/dnsmasq.conf

no-resolv

dhcp-range=192.168.2.2,192.168.2.15,999h

dhcp-option=3,192.168.2.1

dhcp-option=6,192.168.1.10

interface=ens19.200

systemctl restart dnsmasq

systemctl status dnsmasq

**Проверка настройки на CLI**

hostnamectl set-hostname hq-cli.au-team.irpo; exec bash

timedatectl set-timezone Asia/Krasnoyarsk

systemctl restart network

## Настойка DNS сервера на HQ-SRV

systemctl disable --now bind

apt-get update

apt-get install dnsmasq **-y**

systemctl enable --now dnsmasq

vim /etc/dnsmasq.conf

no-resolv

domain=au-team.irpo

server=/au-team.irpo/192.168.3.10

server=8.8.8.8

interface=ens18

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

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

#cname=moodle.au-team.irpo,hq-rtr.au-team.irpo

#cname=wiki.au-team.irpo,hq-rtr.au-team.irpo

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

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

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

vim /etc/hosts

192.168.1.1 hq-rtr.au-team.irpo

systemctl restart dnsmasq

ping hq-rtr.au-team.irpo

## Настойка доменного контроллера Samba на машине BR-SRV

apt-get update

apt-get install task-samba-dc **-y**

Теперь удалим конфиг smb.conf, чтобы он не мешал при настройке службы:

rm -rf /etc/samba/smb.conf

vim /etc/hosts

192.168.3.10 br-srv.au-team.irpo

Теперь в конфигурацию нашего DNS-сервера на **HQ-SRV** добавим следующую строку:

server=/au-team.irpo/192.168.3.10

перезапуск сервиса

Запуск установки

**samba-tool domain provision**

**AU-TEAM.IRPO**

**AU-TEAM**

**dc**

**SAMBA\_INTERNAL**

**192.168.1.10** (Здесь вводим значение вручную)

**123qweR%**

Перемещаем сгенерированный конфиг krb5.conf и включаем службу samba:

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

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

Из-за того, что на Alt Linux могут пропадать IP-адреса после перезагрузки системы, добавим запись о перезапуске службы network и samba в rc.local (именно в таком порядке):

**vim /etc/rc.d/rc.local**

**#!/bin/sh -e**

**systemctl restart network**

**systemctl restart samba**

**exit 0**

**chmod +x /etc/rc.d/rc.local**

**reboot**

Домен работает, у вас должно всё соответствовать картинке выше.

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%**

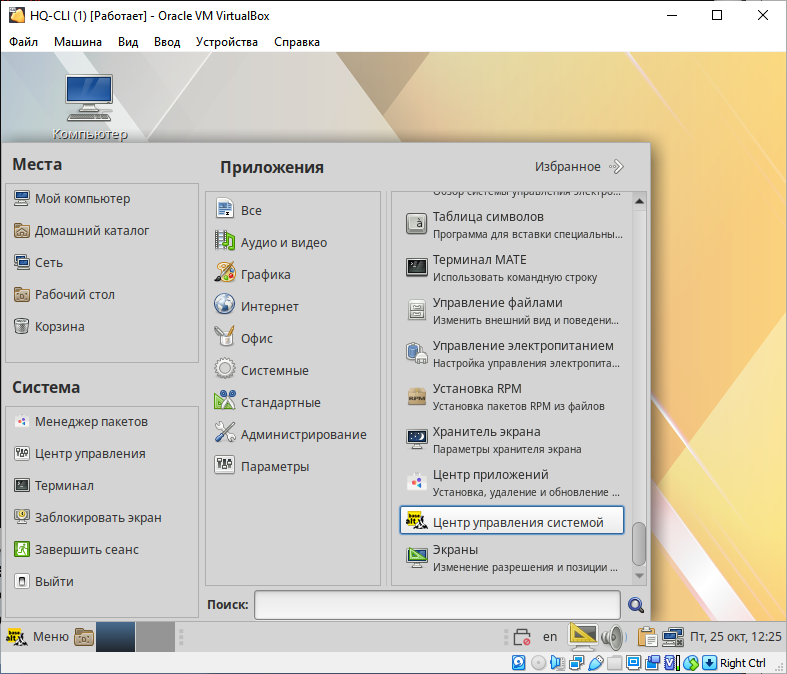
Теперь создадим группу и поместим туда созданных пользователей:

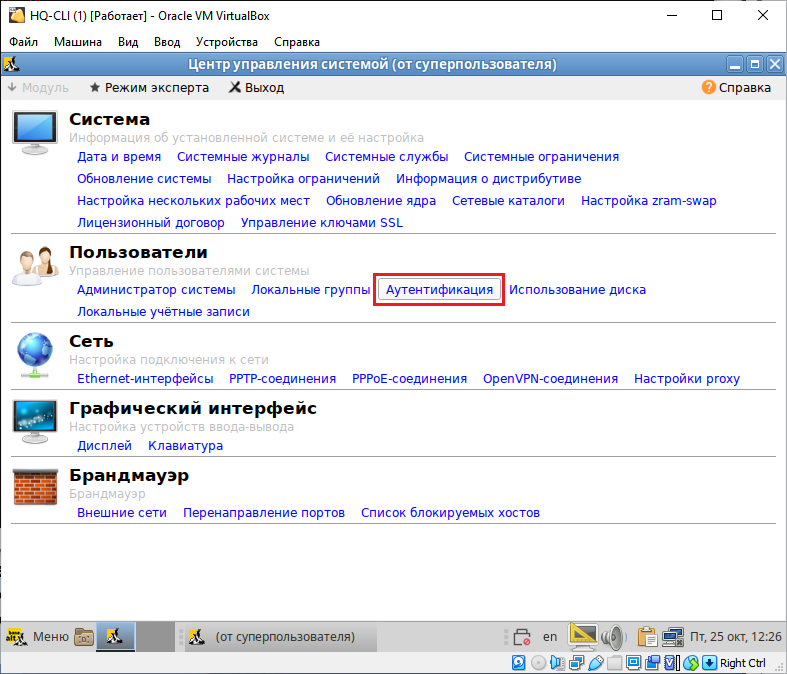
**samba-tool group add hq**

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

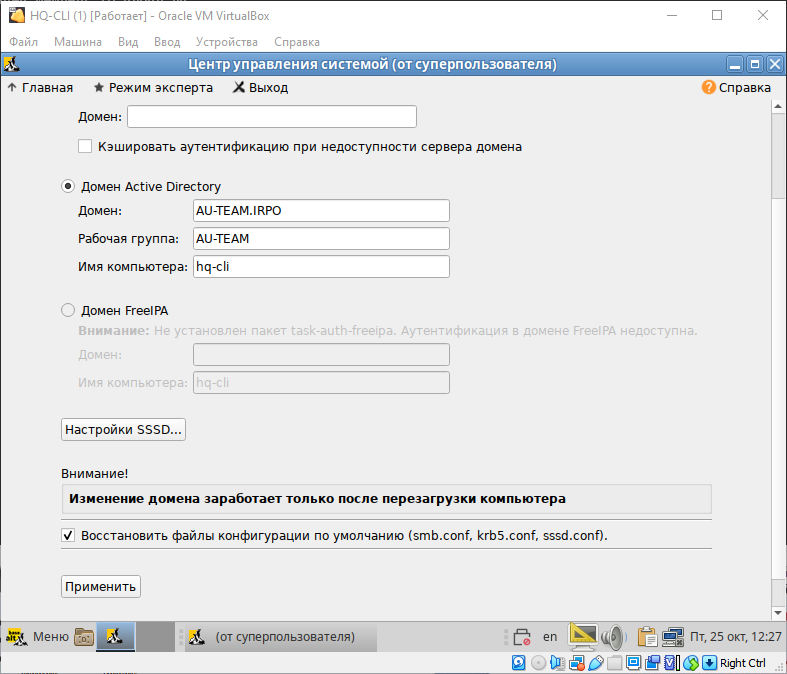
systemctl start samba

Теперь введём клиентскую машину в домен:

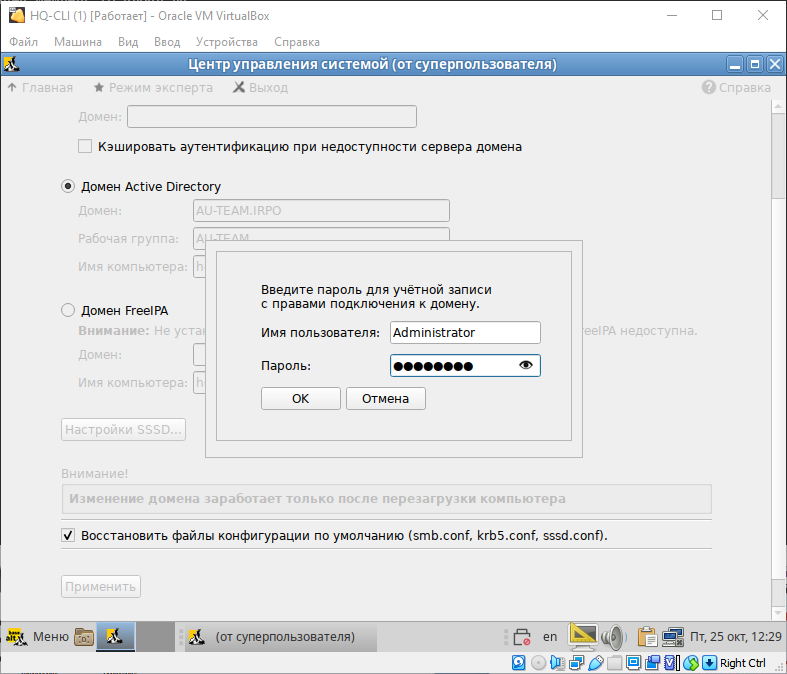




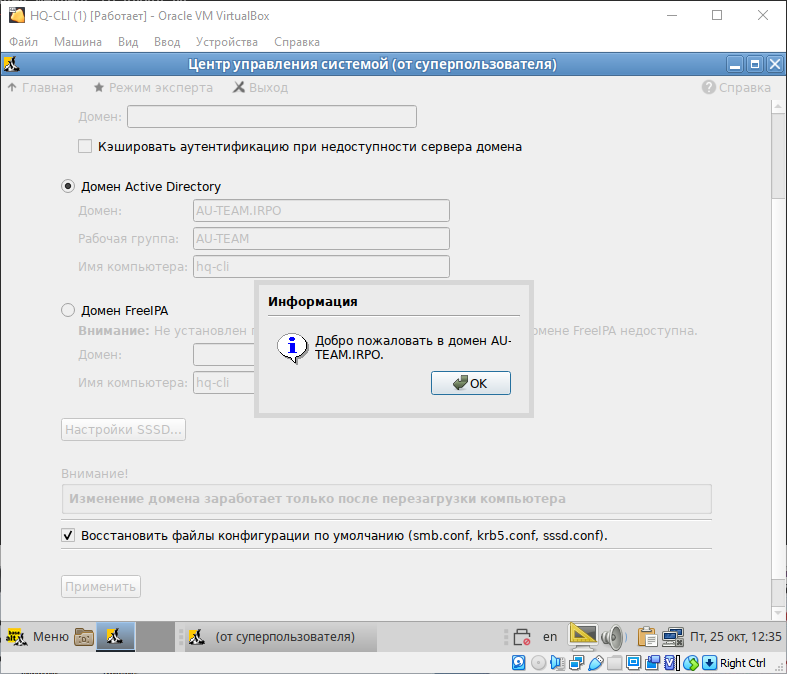
Заполняем так, как показано на картинке ниже:



Вводим [пароль](#bookmark=id.hgqxtshk1u0), который вводили при настройке домена через samba-tool:



После ввода в домен должно появиться следующее сообщение на экране:



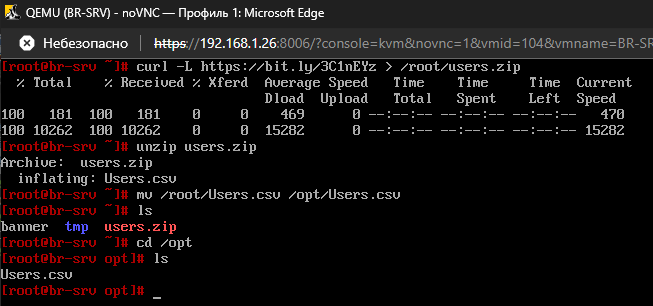
Перезагружаем машину **HQ-CLI**

**Скрипт**

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

**unzip /root/users.zip**

**mv /root/Users.csv /opt/Users.csv**

****

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**

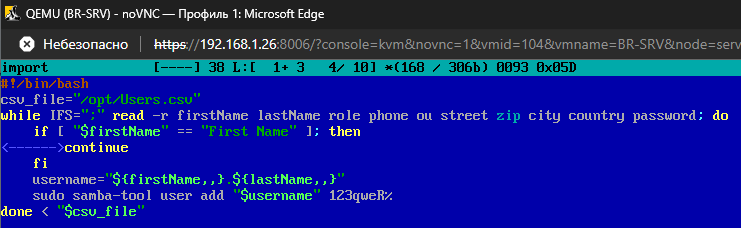
**continuevim**

**fi**

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

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

**done < “$csv\_file”**

****

**Не забудь поправить ковычки**

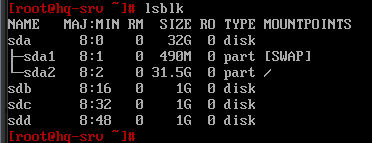
Сохраняем этот файл и выдаём ему право на выполнение и запускаем его:

**chmod +x /root/import**

**bash /root/import**

## Настойка файфового хранилища HQ-SRV

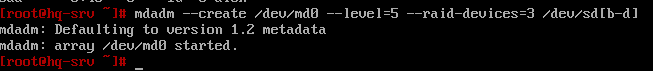
**lsblk**



Обратите внимание, что у вас могут отличаться названия дисков, поэтому указываем при создании названия дисков, которые мы посмотрели ранее командой **lsblk**!

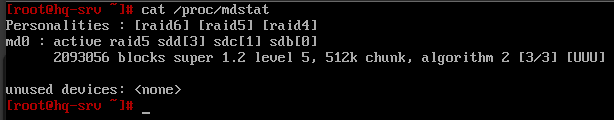
Теперь создадим дисковый массив уровня 5 из трёх дополнительных дисков следующей командой:

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



Посмотрим статус нашего raid-массива:

**cat /proc/mdstat**



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

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

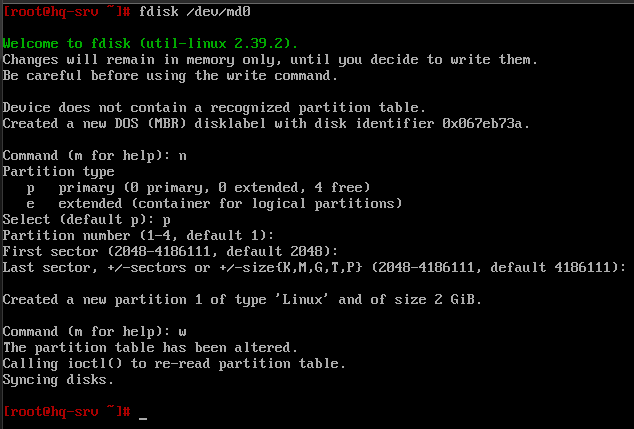
Теперь создаём раздел через **fdisk**.

Для этого пишем следующую команду:

**fdisk /dev/md0**



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

****

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

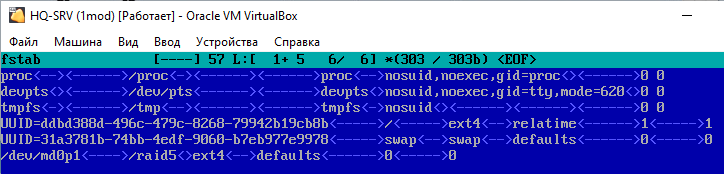
**mkfs.ext4 /dev/md0p1**

****

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

vim **/etc/fstab**

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

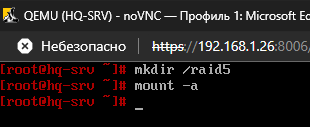


Затем создаём каталог **/raid5** и монтируем ФС из **/etc/fstab**:

**mkdir /raid5**

**mount -a**

Заметьте, что команда не должна ничего выводить!

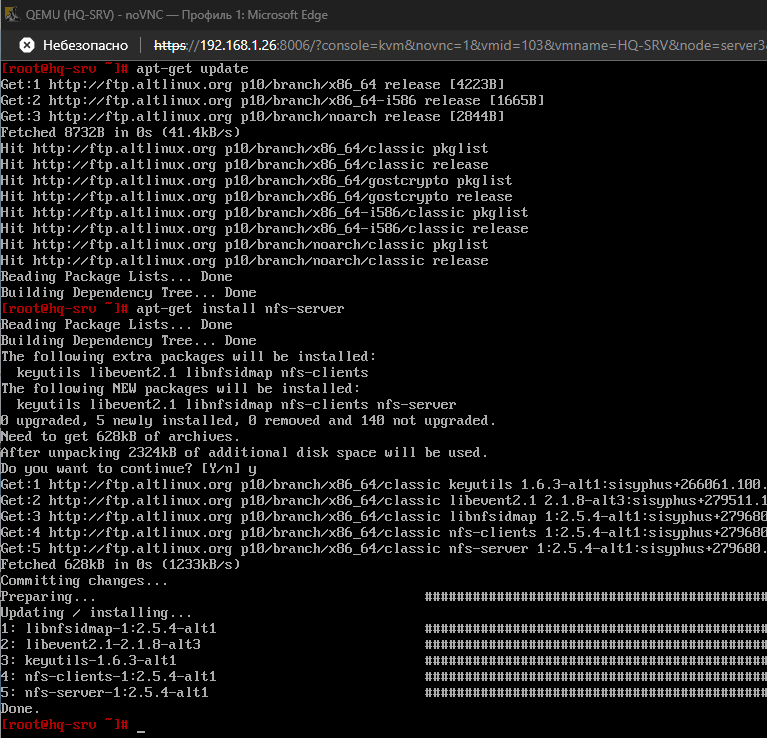
****

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

**apt-get update**

**apt-get install nfs-server -y**

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

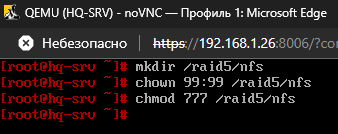


Приступаем к самой настройке, создадим каталог, назначим нового владельца и группу ему и выдадим новые права:

**mkdir /raid5/nfs**

**chown 99:99 /raid5/nfs**

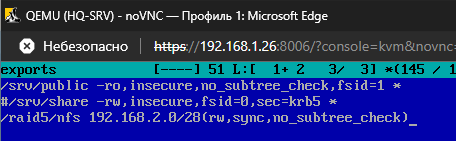
**chmod 777 /raid5/nfs**



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**

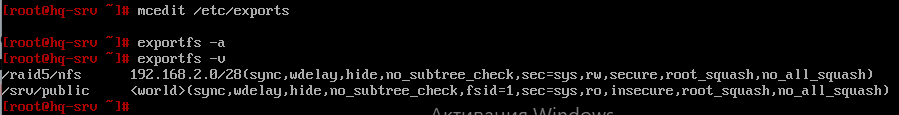
**/raid5/nfs 192.168.2.0/28(rw,sync,no\_subtree\_check)**



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

**exportfs -a**

**exportfs -v**



Перезапускаем службу NFS:

**systemctl restart nfs**

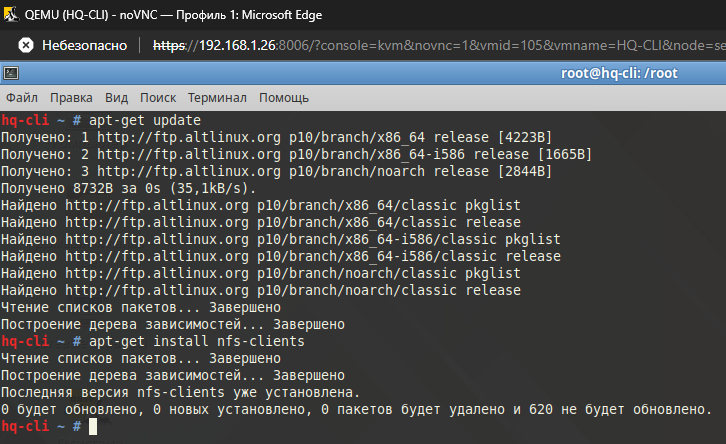


Теперь идём монтировать этот каталог на клиенте **HQ-CLI**, для этого нужно:

**apt-get update**

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

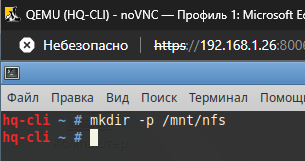
**systemctl enable --now nfs-client.target**

****

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

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

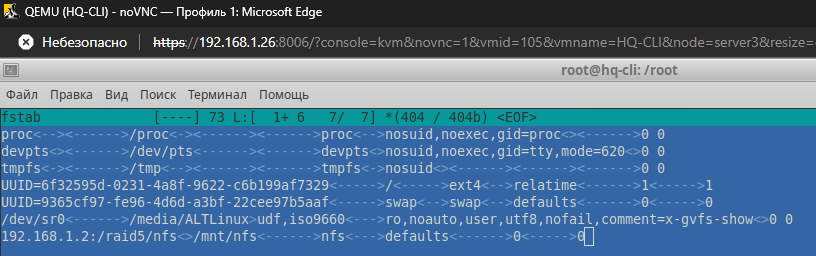
**mkdir -p /mnt/nfs**

****

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

vim **/etc/fstab**

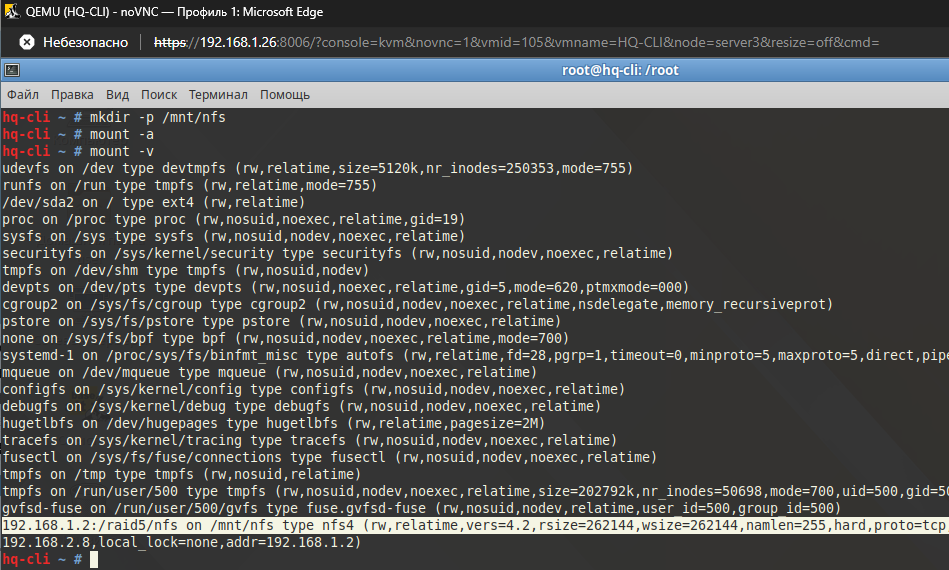
**192.168.1.10:/raid5/nfs /mnt/nfs nfs defaults 0 0**



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

**mount -a**

**mount -v**



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

**touch /mnt/nfs/bazhenov**

## Настойка ansible на сервере BR-SRV

Перед началом работы включите ssh на машине cli

**apt-get update**

**apt-get install ansible -y**

**mkdir -p /etc/ansible**

**vim /etc/ansible/hosts**

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

**hq-cli** [**ansible\_host=user@192.168.2.**](mailto:ansible_host=user@192.168.2.%20)**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**

**ssh-keygen -t rsa**

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

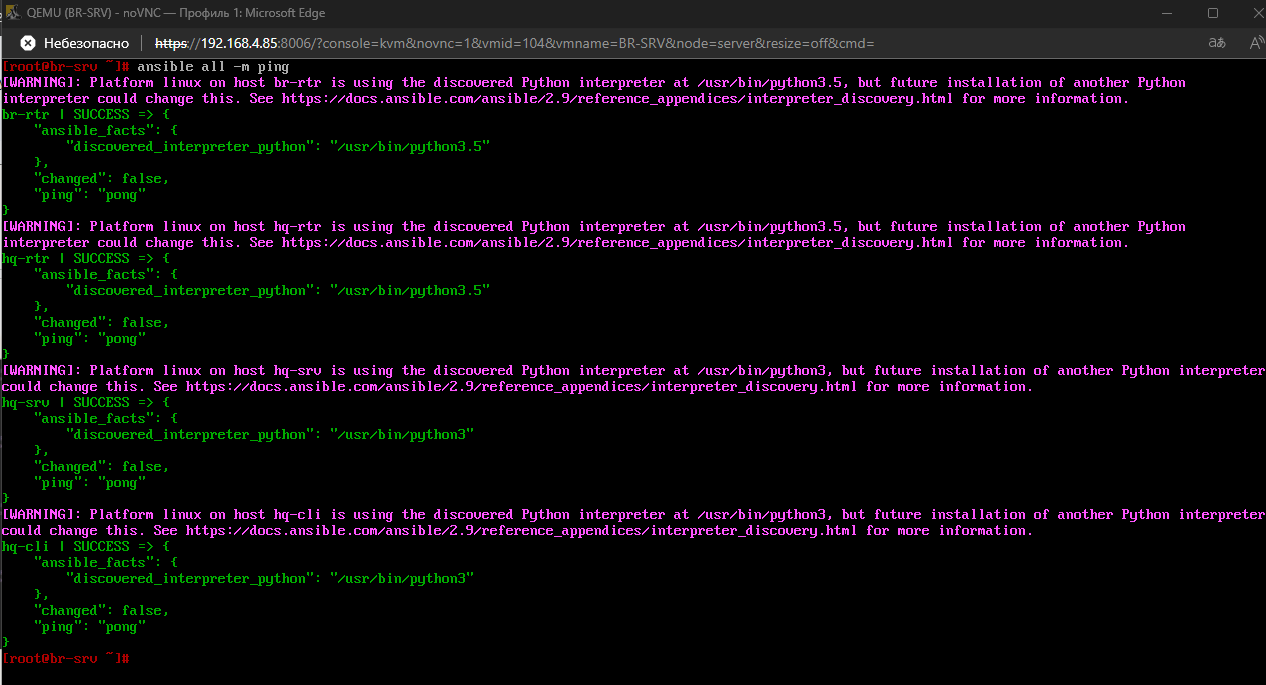
**ssh-copy-id -p 22** [**user@192.168.2.**](mailto:user@192.168.2.)**4**

**ssh-copy-id -p 2024 sshuser@192.168.1.10**

**ssh-copy-id -p 2024** [**net\_admin@192.168.1.1**](mailto:net_admin@192.168.1.1)

**Проверка**

**ansible all -m ping**



## Настойка docker и установка mediawiki на сервере BR-SRV

apt-get update

apt-get install docker-engine docker-compose **-y**

systemctl enable --now docker

systemctl status docker

docker pull mediawiki

docker pull mariadb

vim /root/wiki.yml

**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"**

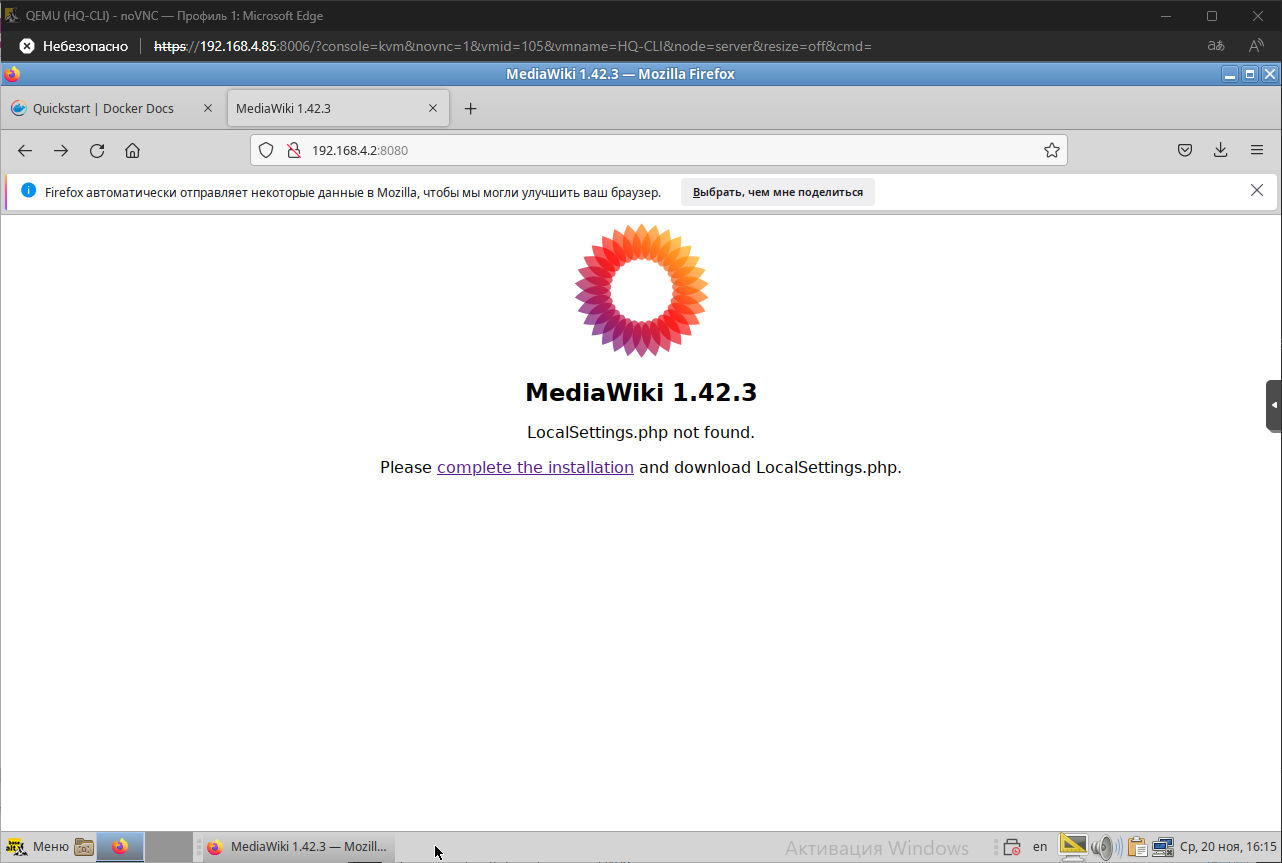
**#volumes: [ /root/mediawiki/LocalSettings.php:/var/www/html/LocalSettings.php ]**

**volumes:**

**mariadb\_data:**

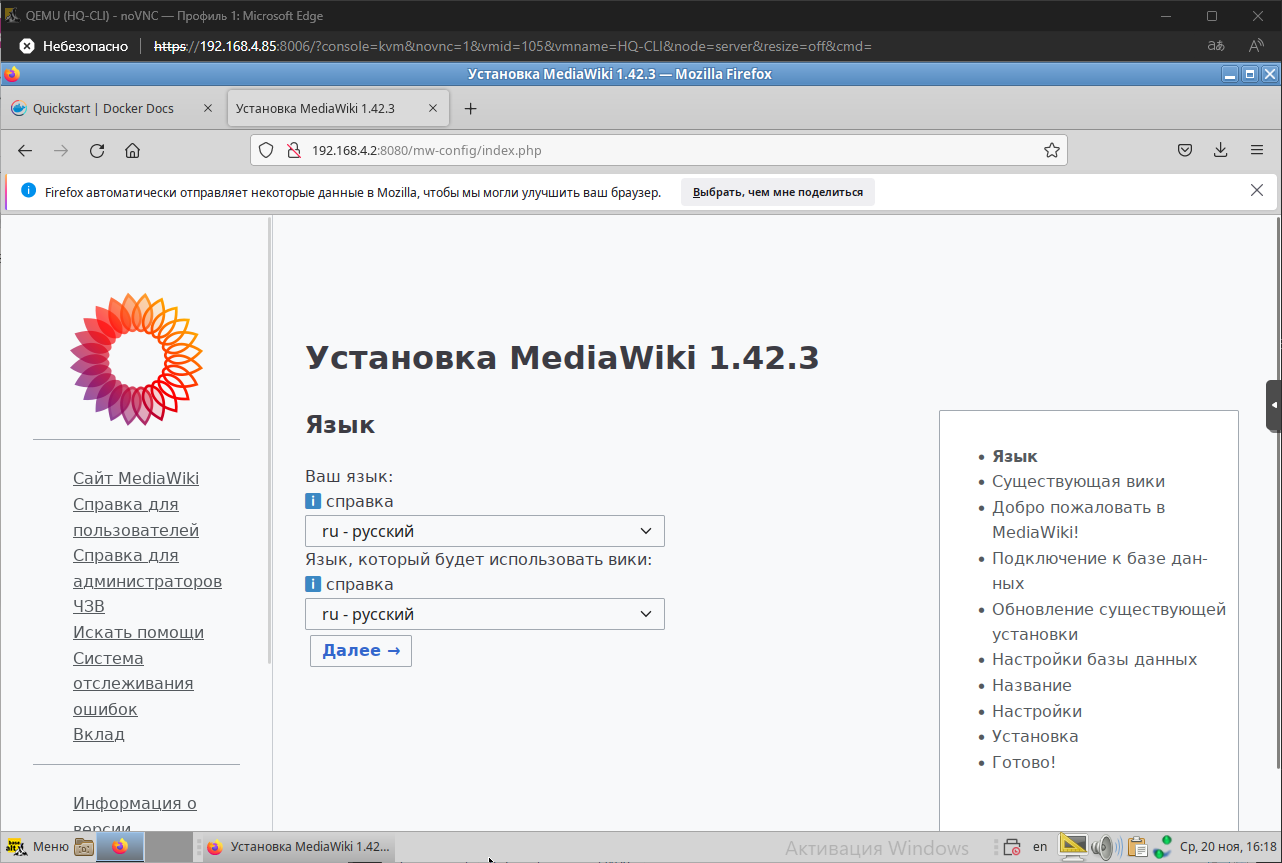
docker compose -f /root/wiki.yml up -d – запуск контейнера

Переходим на машину cli и вбиваем в браузере 192.168.3.10:8080

****

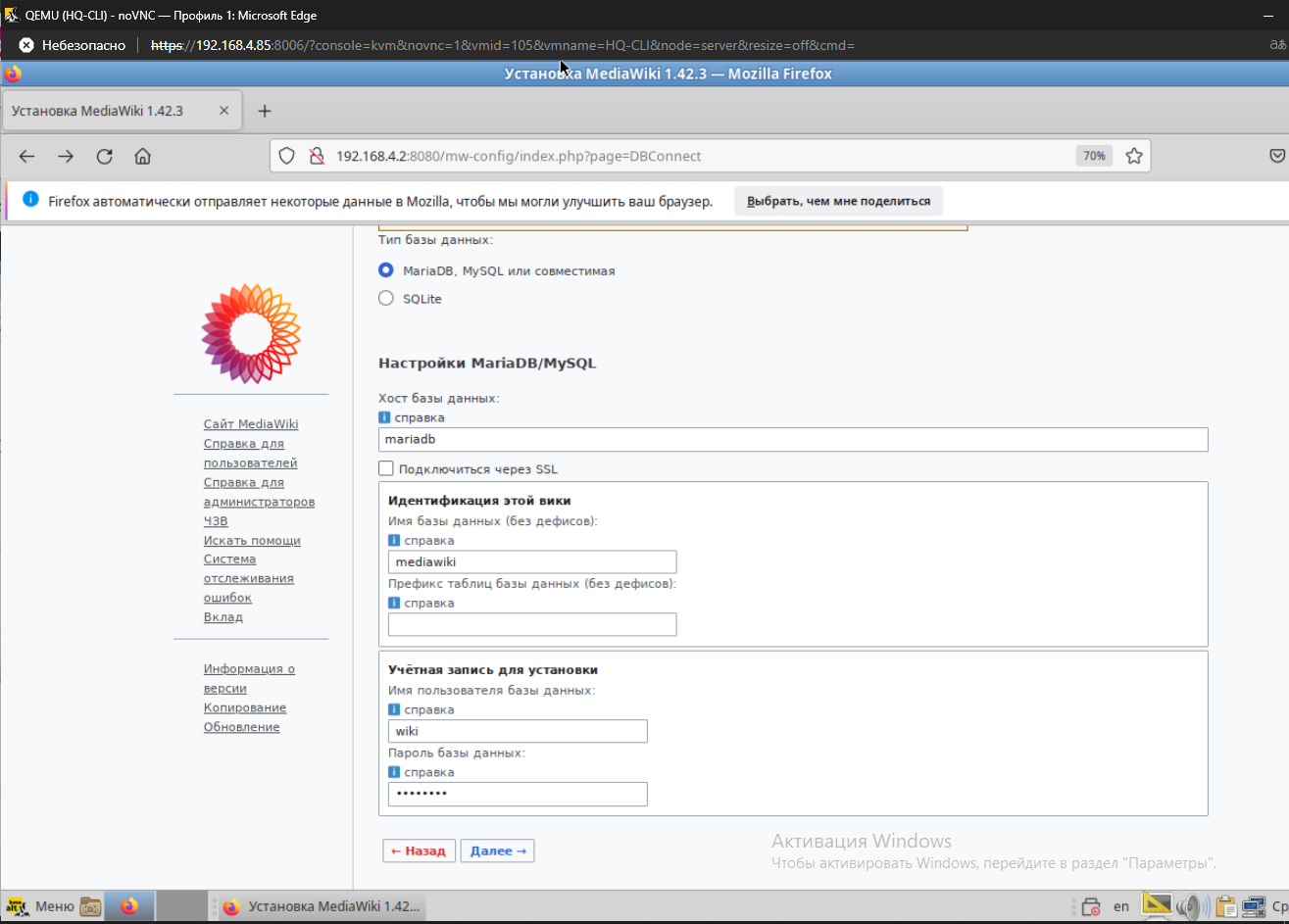
Видим, что файл **LocalSettings.php** не найден, и нажимаем на **complete the installation** или **set up the wiki**.

Выбираем удобный для вас язык:

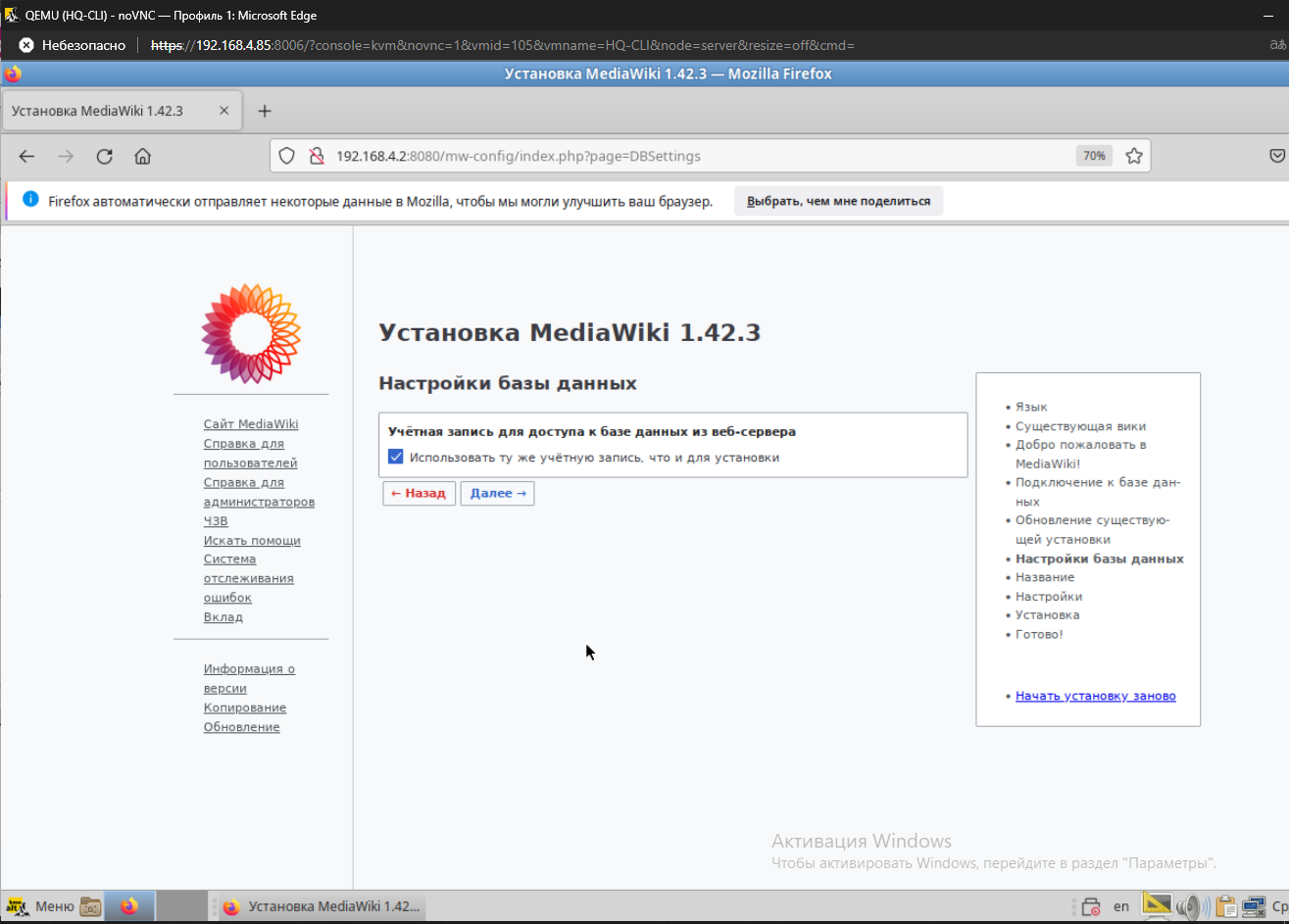




Нажимаем далее и видим строки, которые нужно заполнить:



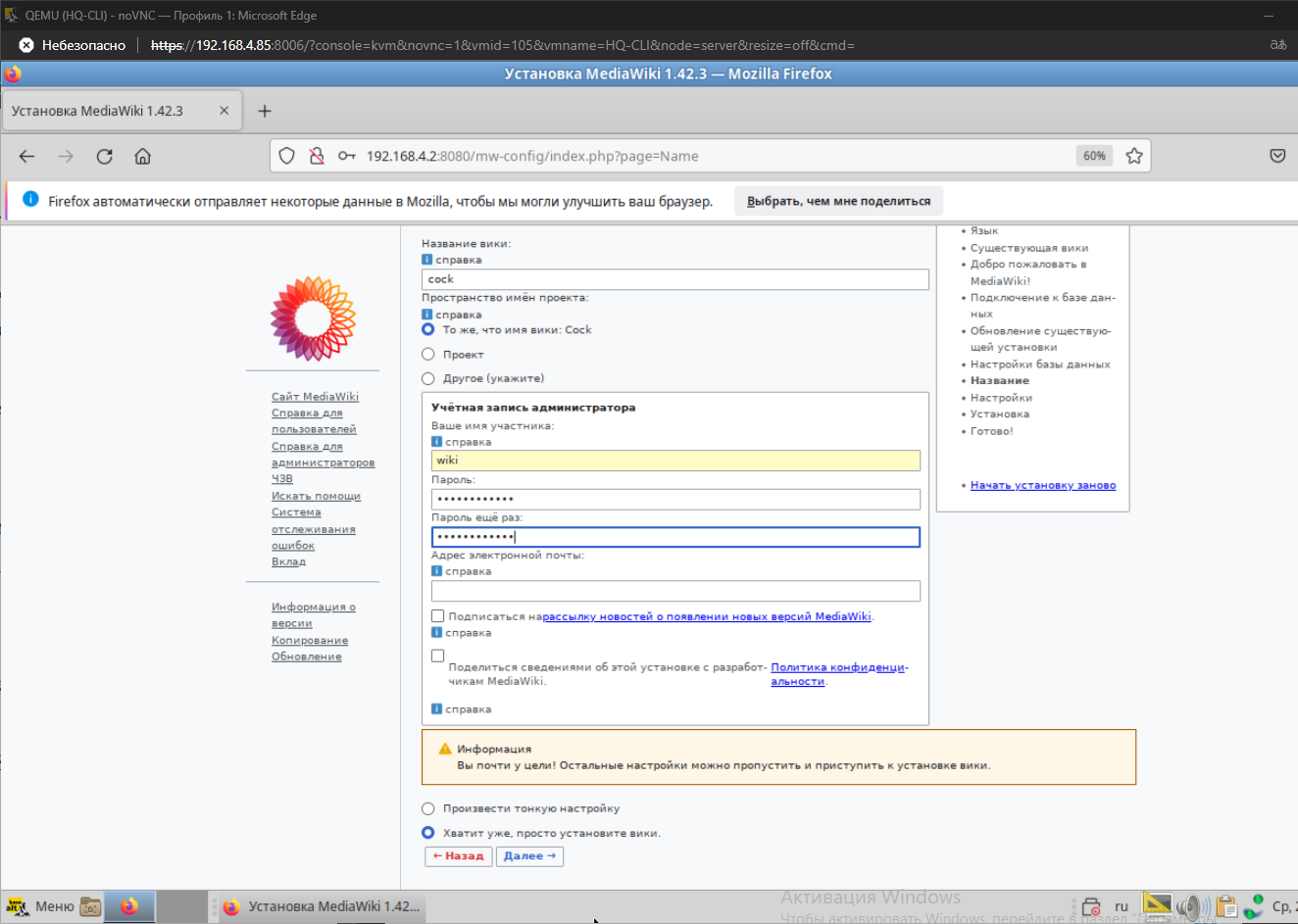
Прожимаем Далее, оставляя всё как есть:



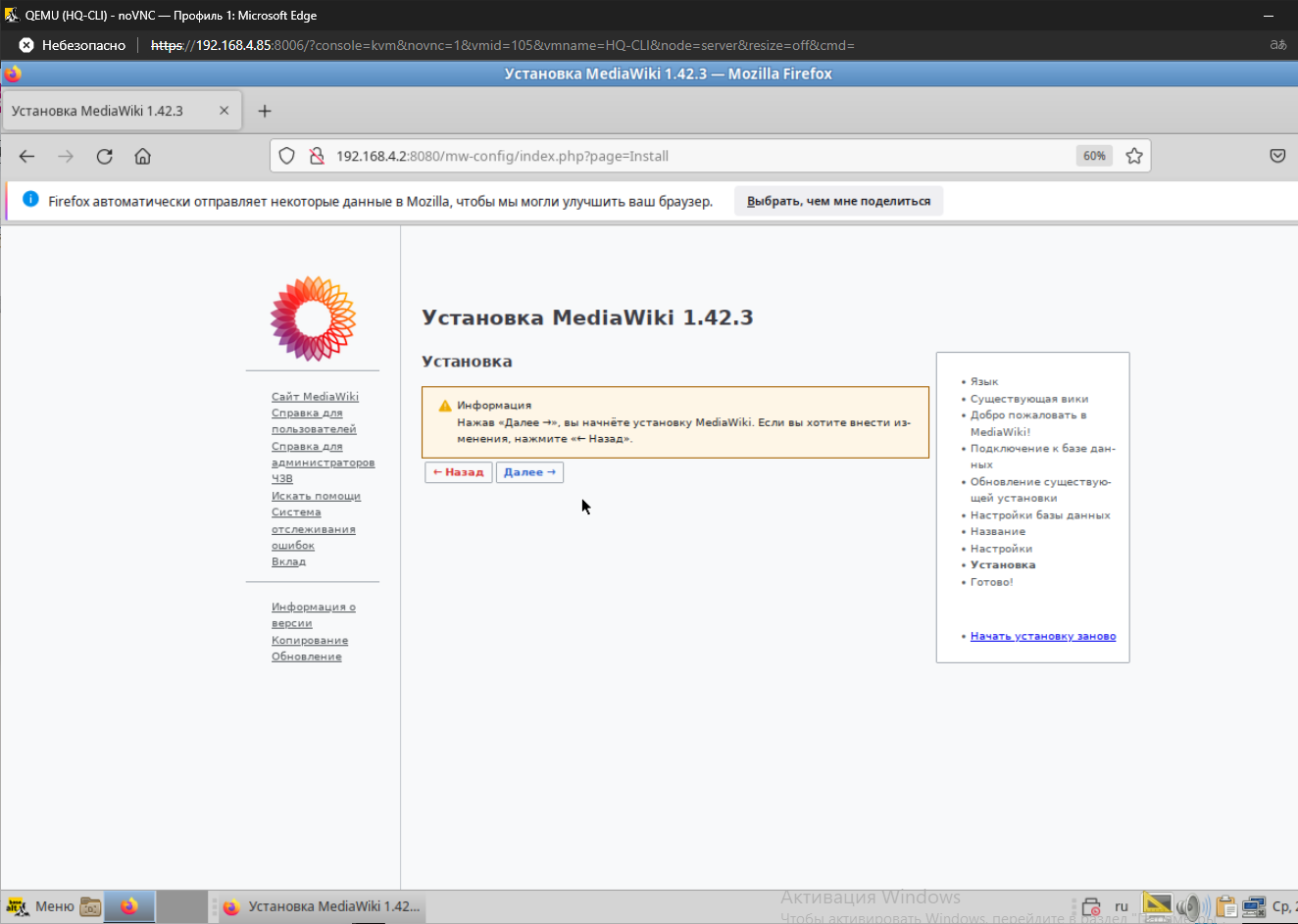
Пишем в строках следующее и выбираем пункты, как на скрине:

**wiki**

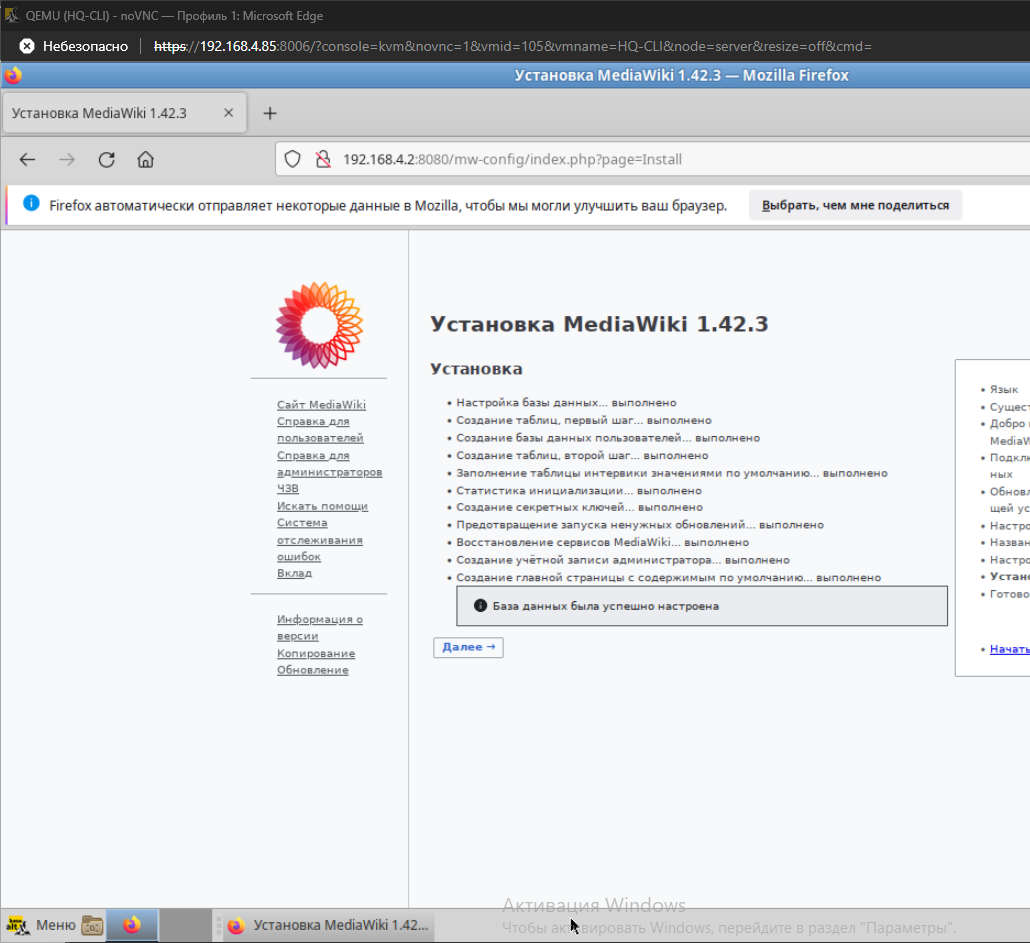
**WikiP@ssw0rd**



Нажимаем Далее:

****

И вот мы успешно создали базу данных:



**scp -P 2024 /home/user/Загрузки/LocalSettings.php sshuser@192.168.3.10:/home/sshuser/**

Возращаемся на br-srv

rm -rf /root/LocalSettings.php

mkdir /root/mediawiki

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

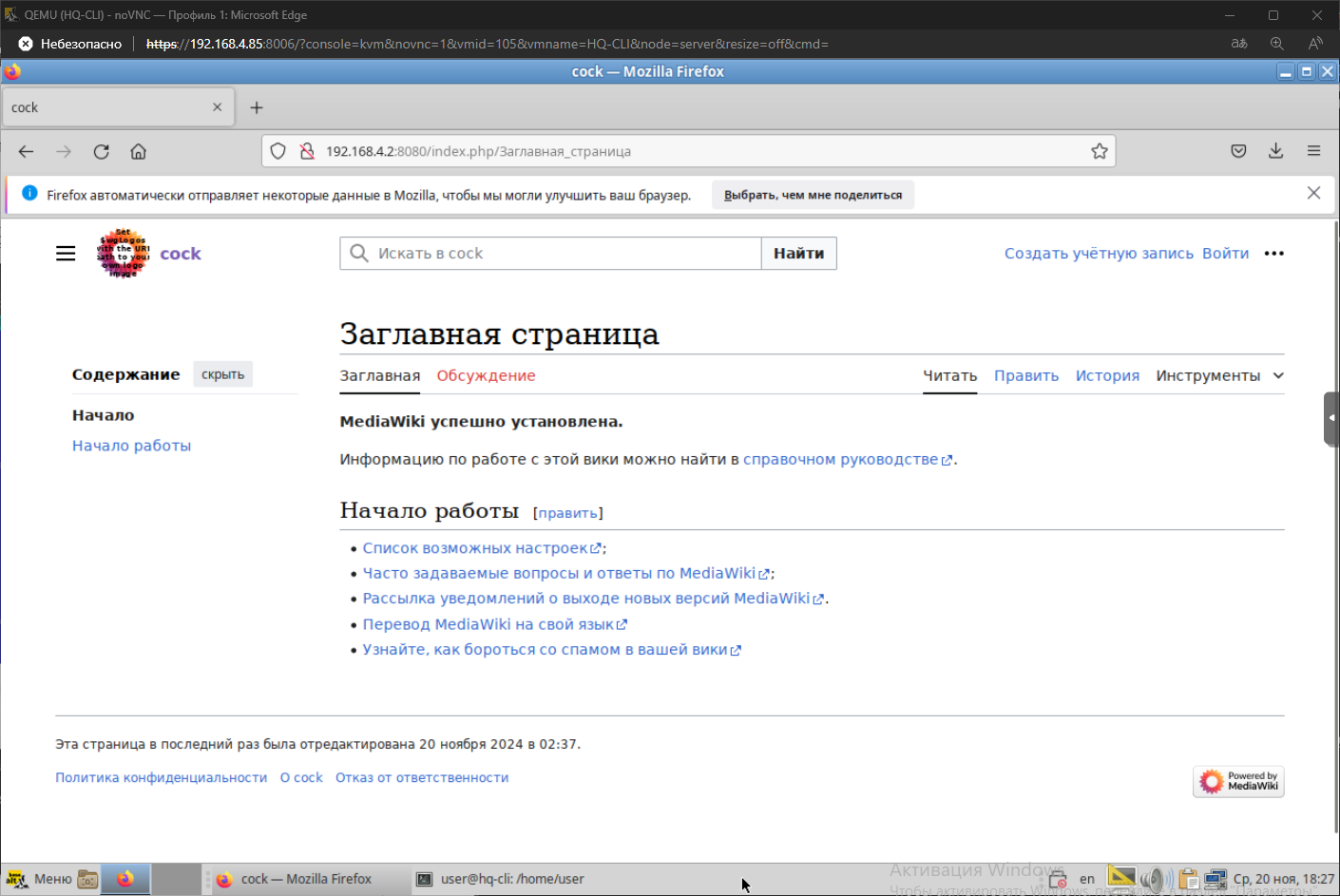
Раскомментируем, как и говорили ранее, строку **volumes…**:



Теперь перезапускаем контейнеры путём запуска контейнера ещё раз:

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

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



## Настойка NTP (Chrony) на HQ-RTR

Настройка на HQ-RTR

apt-get update

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

systemctl restart chronyd

systemctl status chronyd

systemctl enable --now chronyd

vim /etc/resolv.conf

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

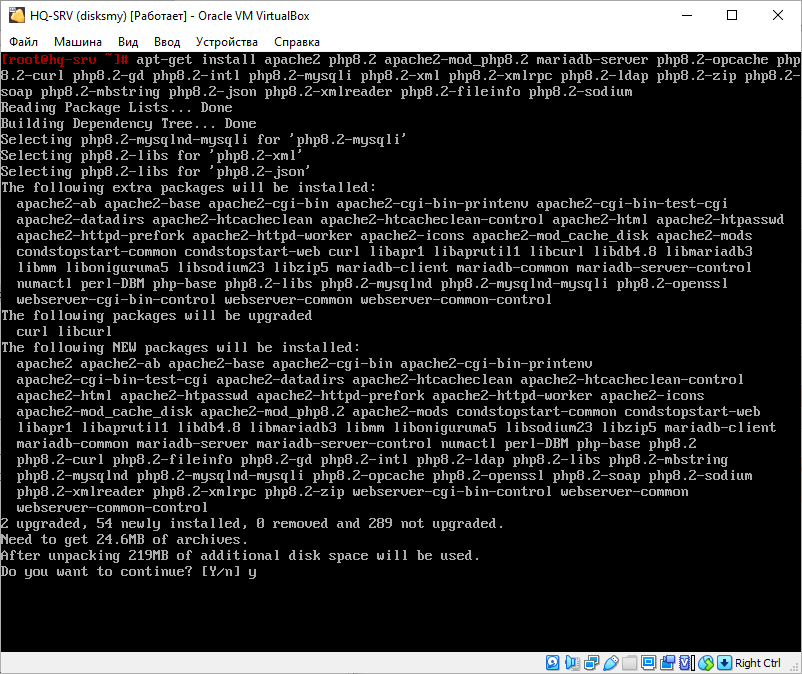
**vim /etc/resolv.conf**

## 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**

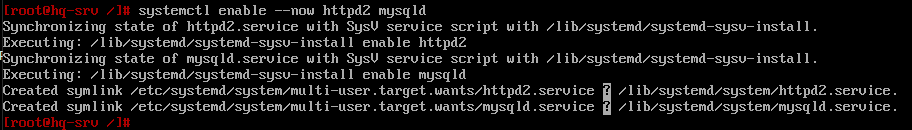
****

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

systemctl start **httpd2**

systemctl start **mysqld**

**systemctl enable –now httpd2 mysqld**

****

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

**mysql\_secure\_installation**

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

**Enter**

Прожимаем **y** для задания пароля:

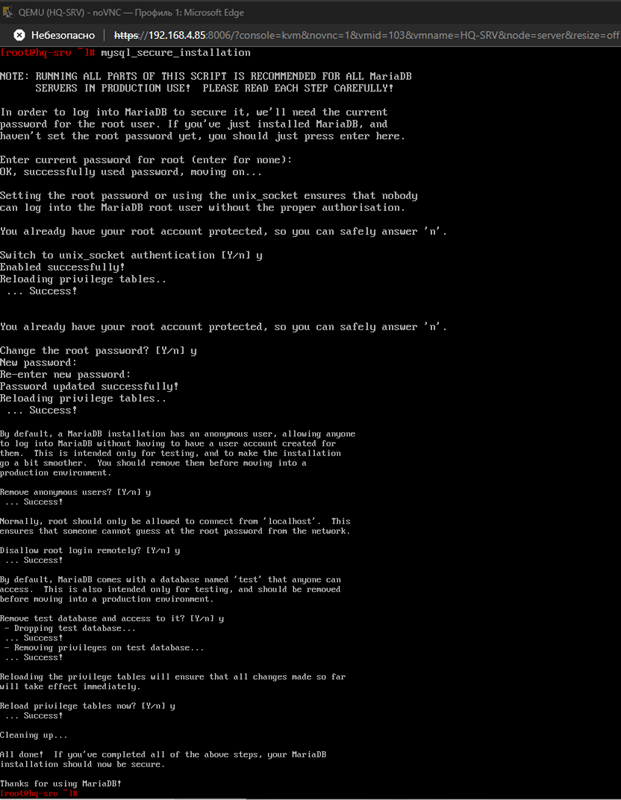
**Y**

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

**123qweR%**

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

**Y**

****

Теперь заходим в СУБД для создания и настройки базы данных:

**mariadb -u root -p**

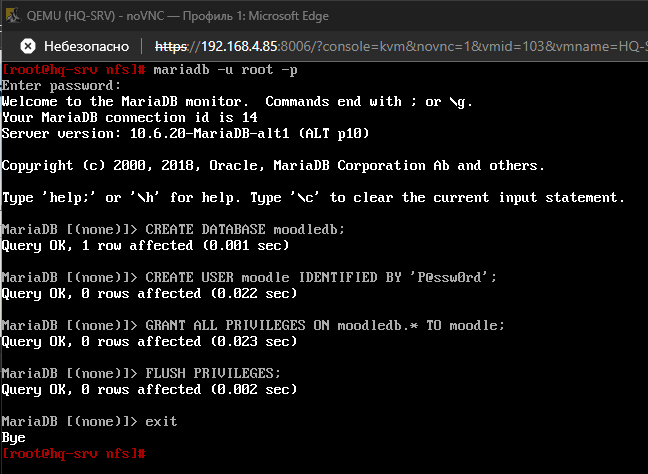
**CREATE DATABASE moodledb;**

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

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

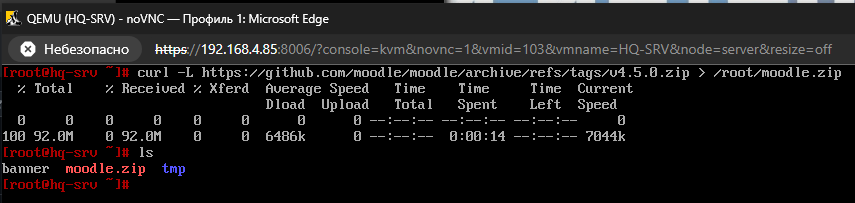
**FLUSH PRIVILEGES;**

**exit**

****

Теперь скачаем сам мудл стабильной версии:

**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**

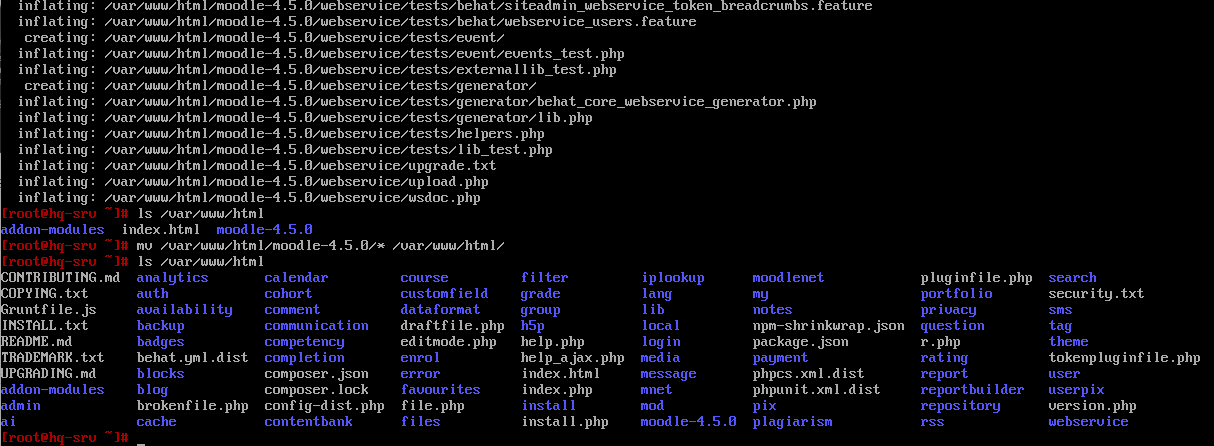
****

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

**unzip /root/moodle.zip -d /var/www/html**

**mv /var/www/html/moodle-4.5.0/\* /var/www/html/**

**ls /var/www/html**

****

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

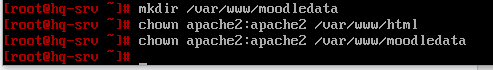
**mkdir /var/www/moodledata**

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

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

**chmod 777 /var**

**chmod 777 /var/www**

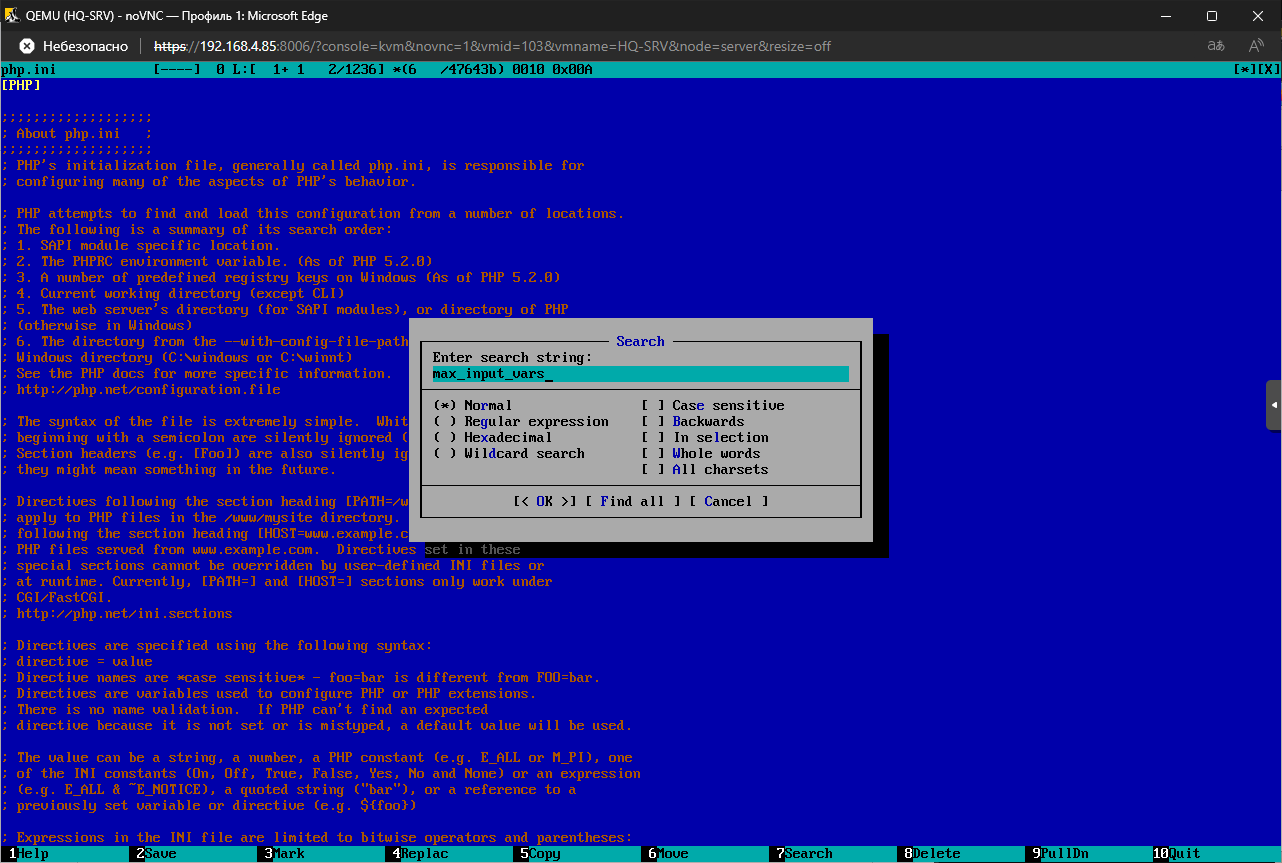
****

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

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

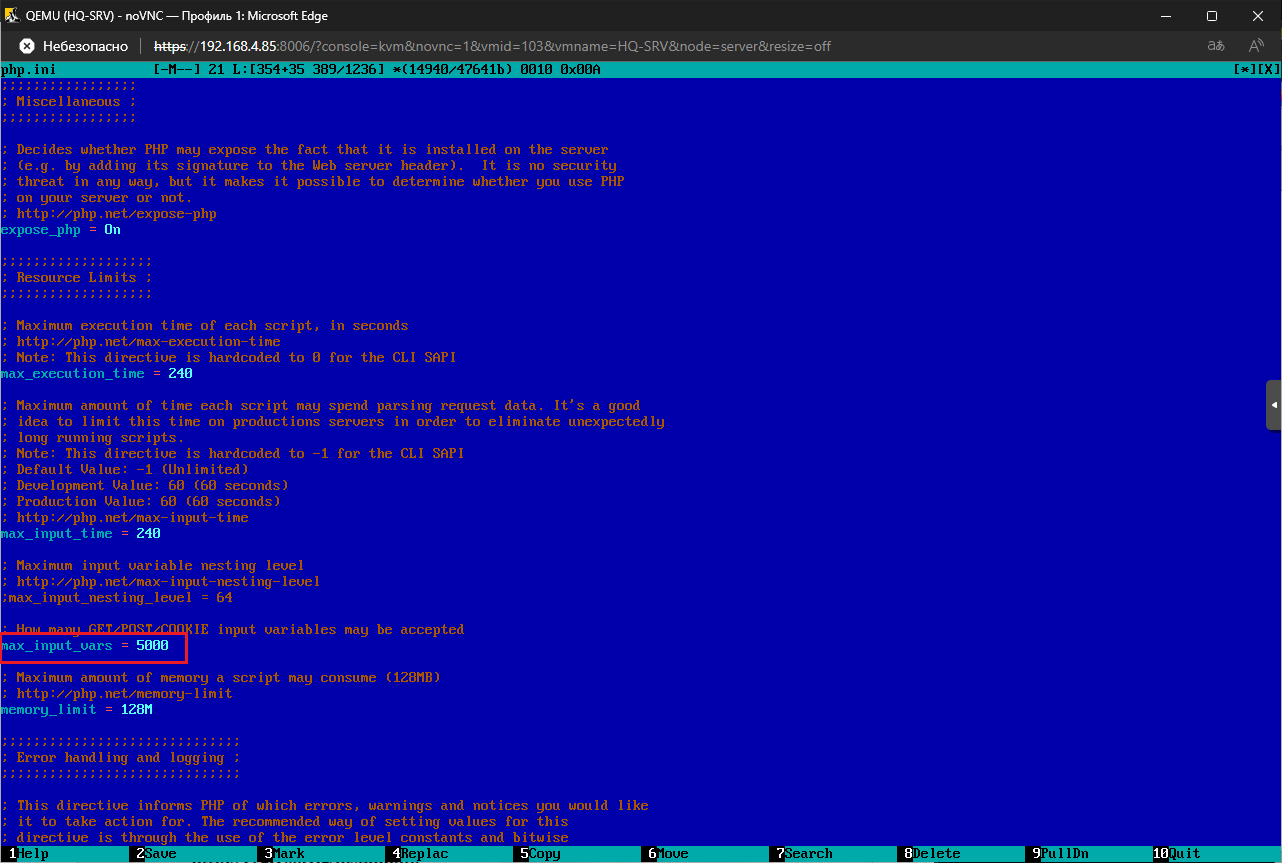
Жмём **F7** для поиска нужной нам строки и пишем туда:

**max\_input\_vars**



Раскомментируем и пишем новое значение:

**max\_input\_vars = 5000**

****

Удаляем стандартную страницу **apache**:

**cd /var/www/html**

**ls**

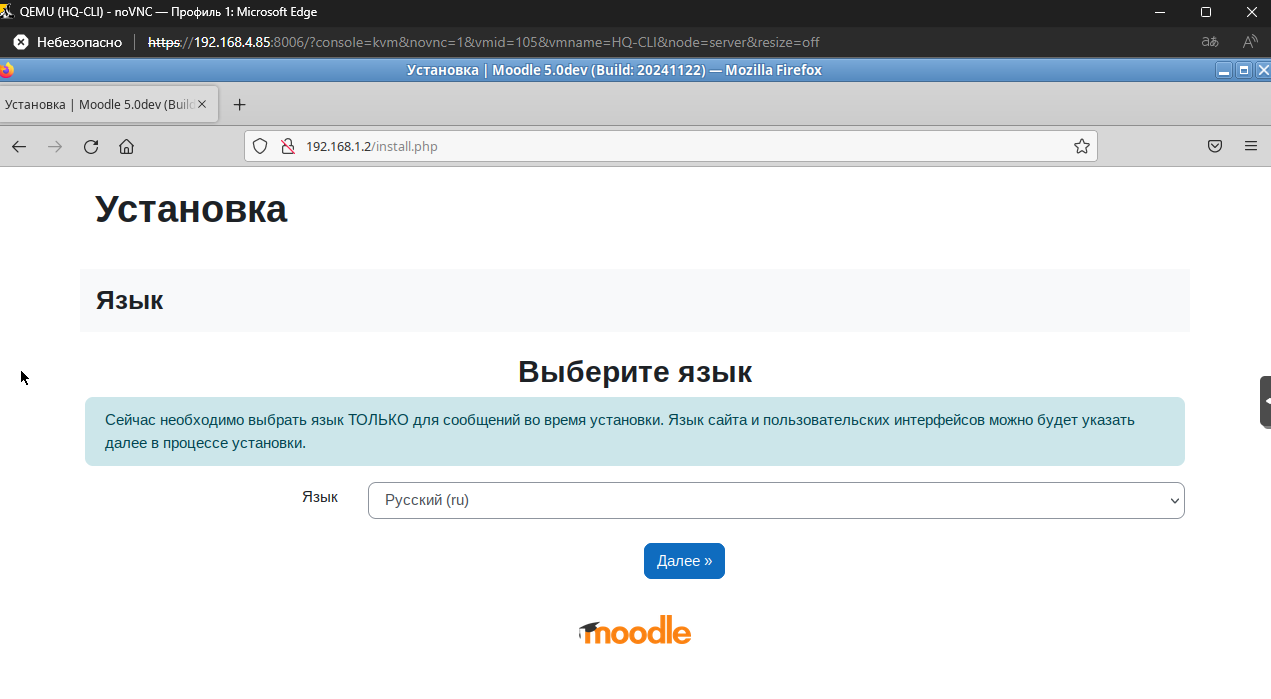
**rm index.html**

Перезапускаем службу **httpd2**:

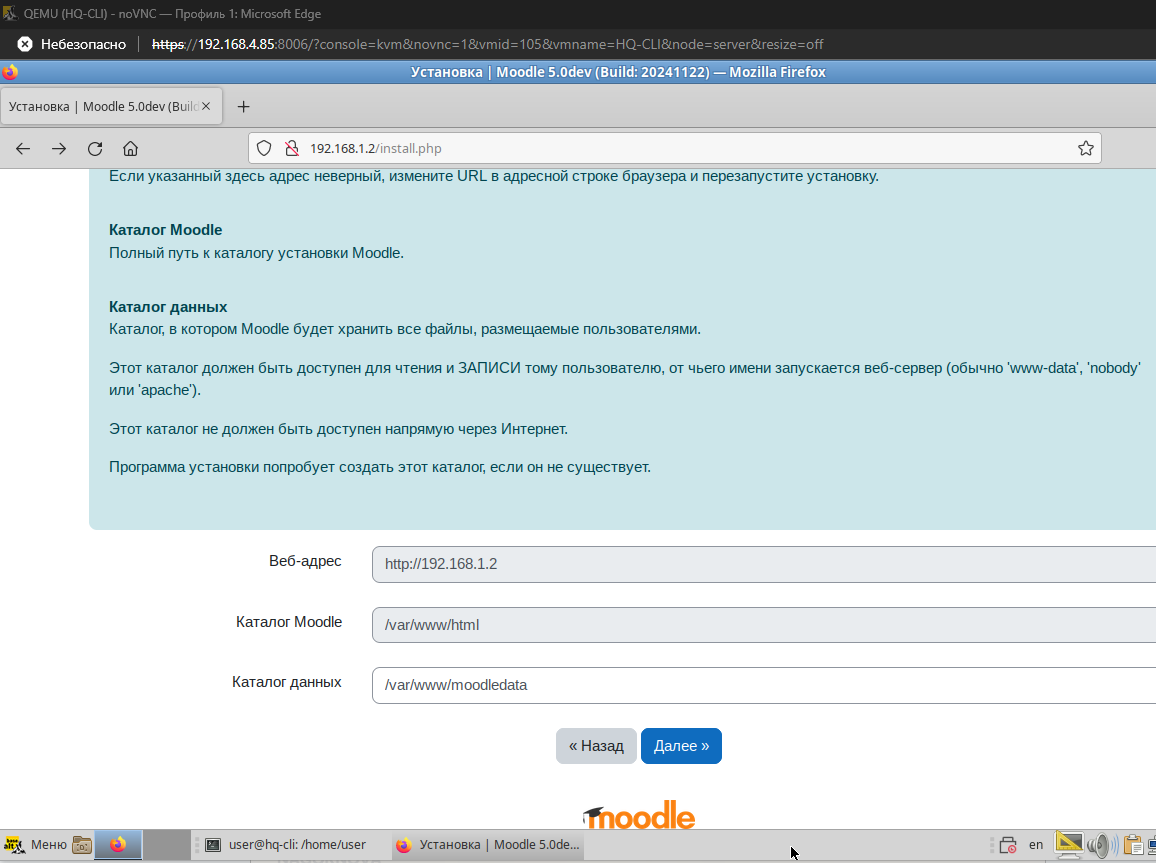
**systemctl restart httpd2**

Теперь подключаемся с клиента HQ-CLI и начинаем настройку:

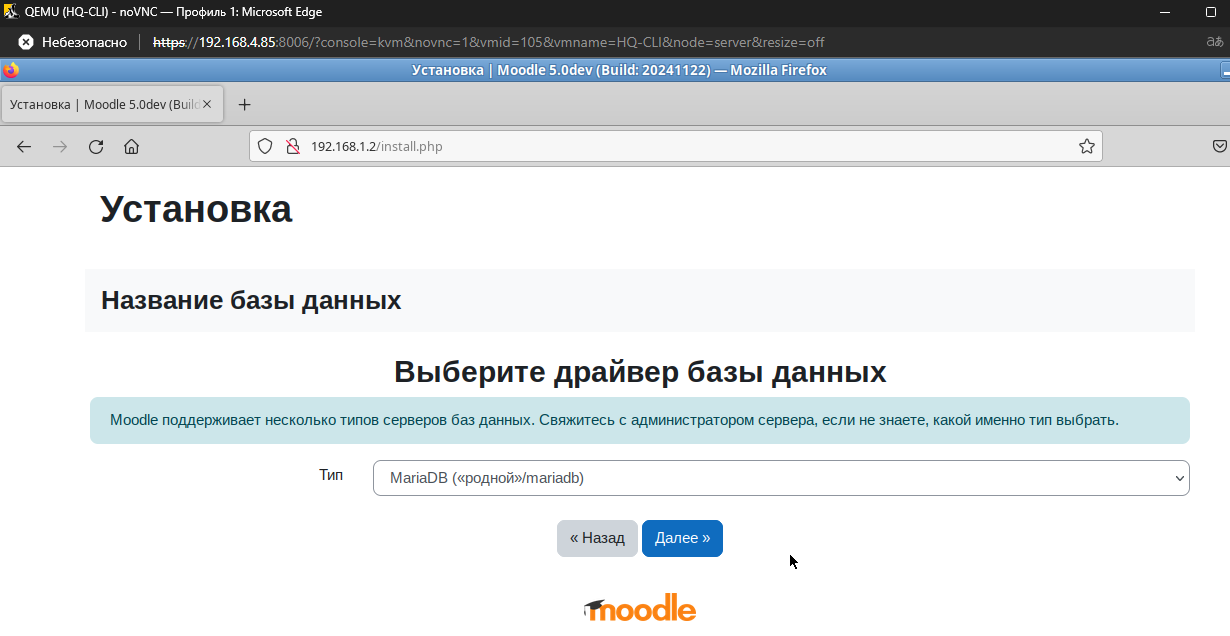
**http://192.168.1.10/install.php**

****

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

****

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

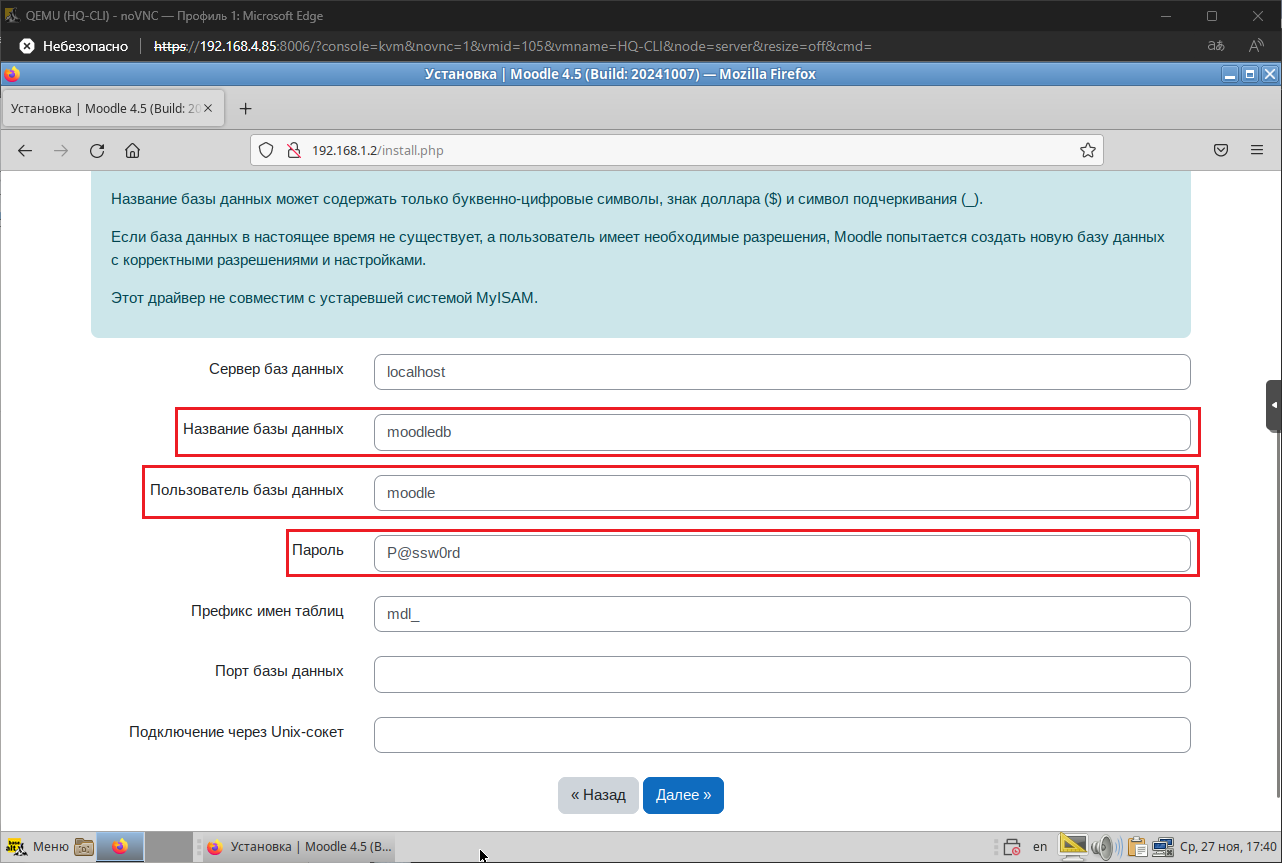
****

Введём нужные данные в следующие строки:

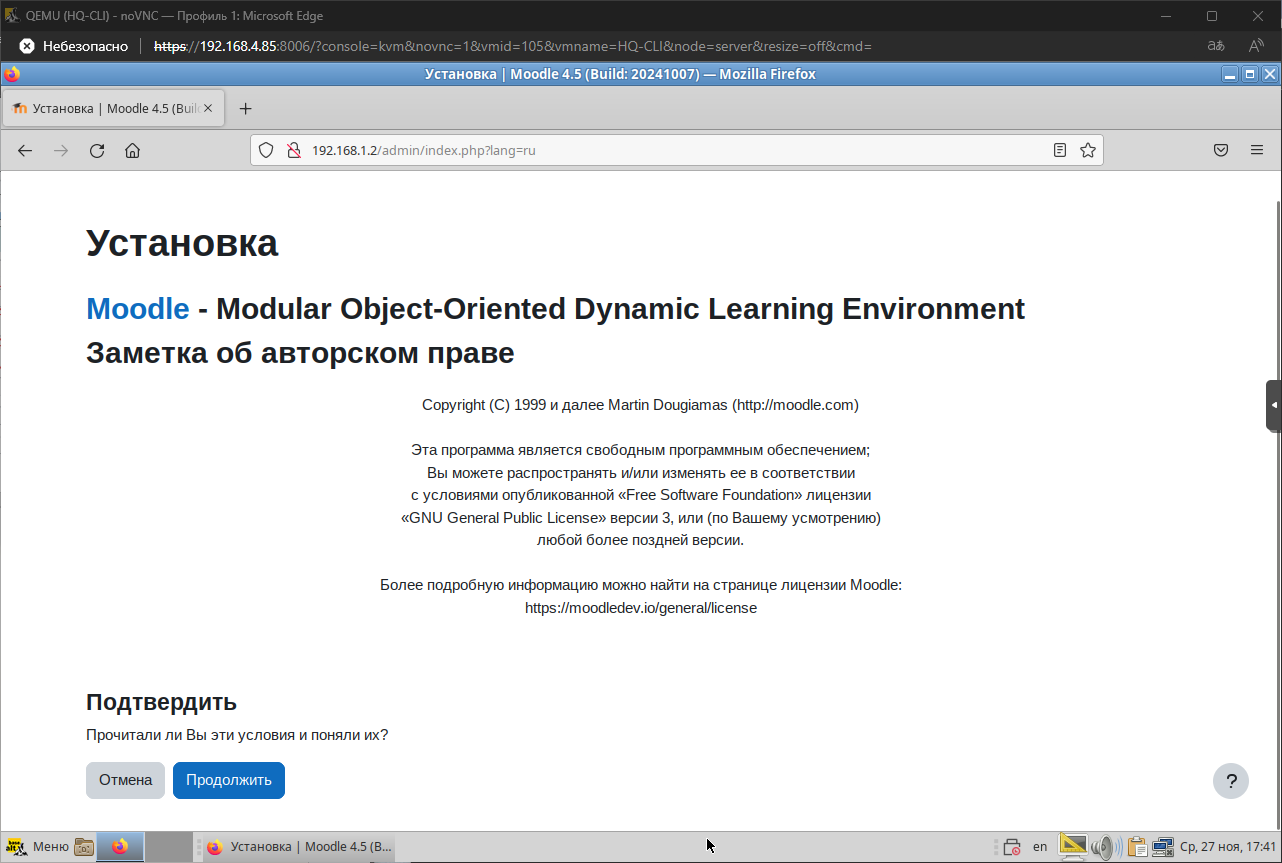
**Название базы данных: moodledb**

**Пользователь базы данных: moodle**

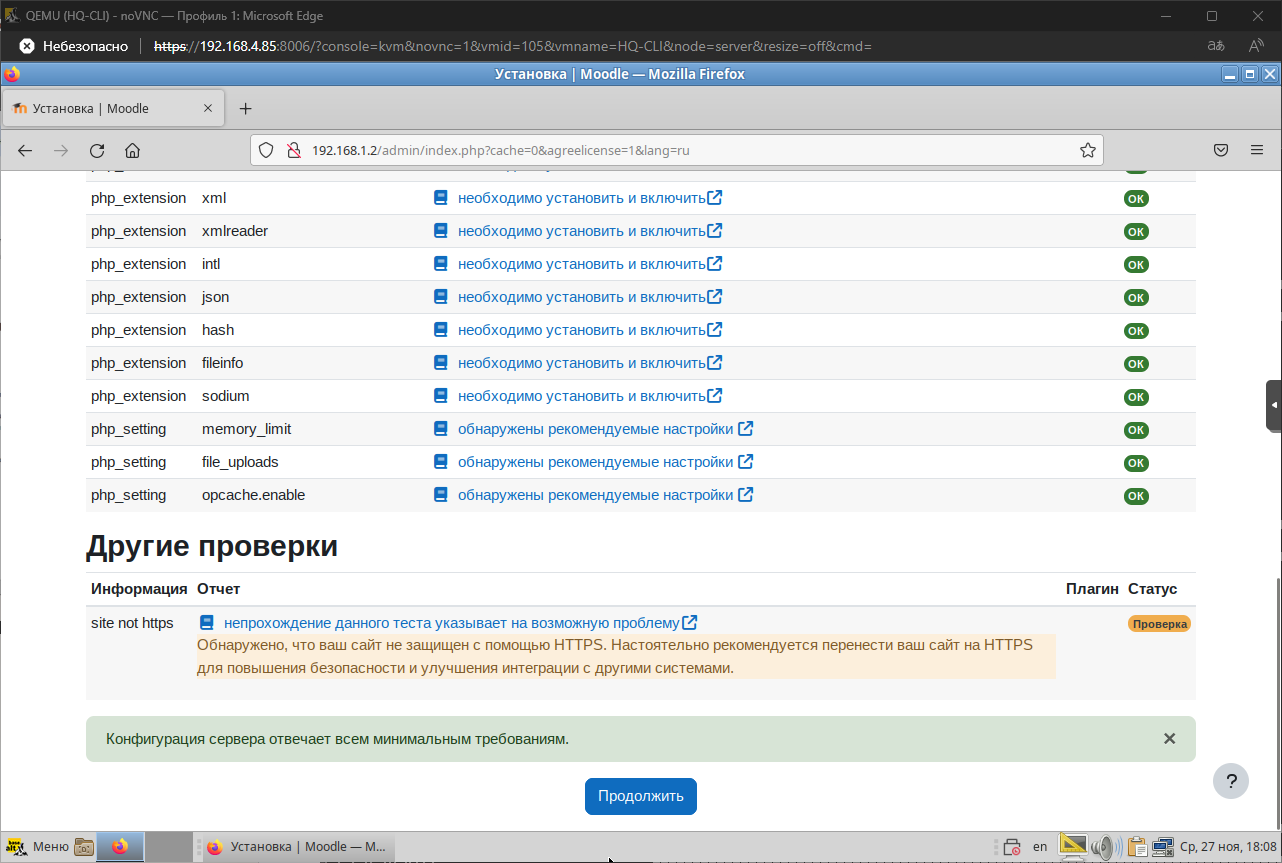
**Пароль: P@ssw0rd**



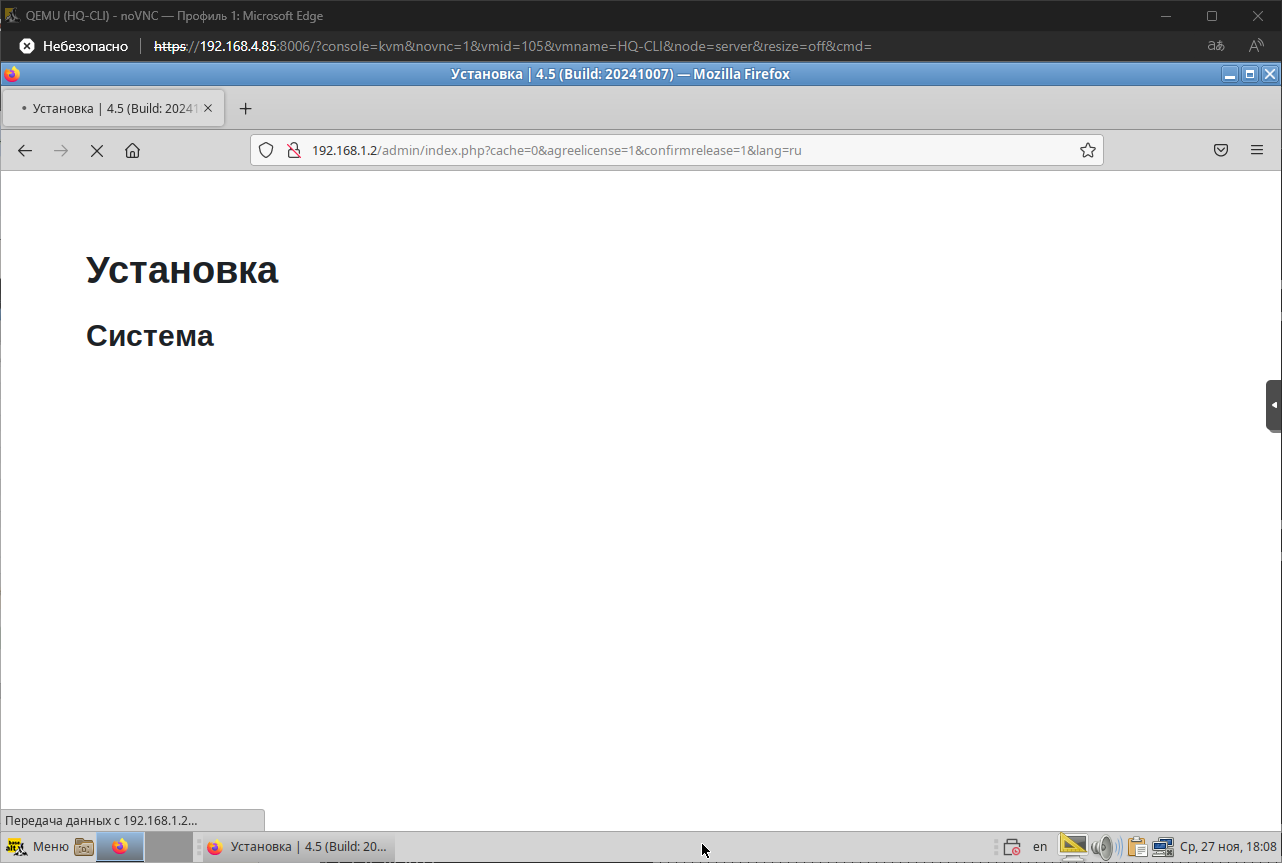
Нажимаем “**Продолжить**”:



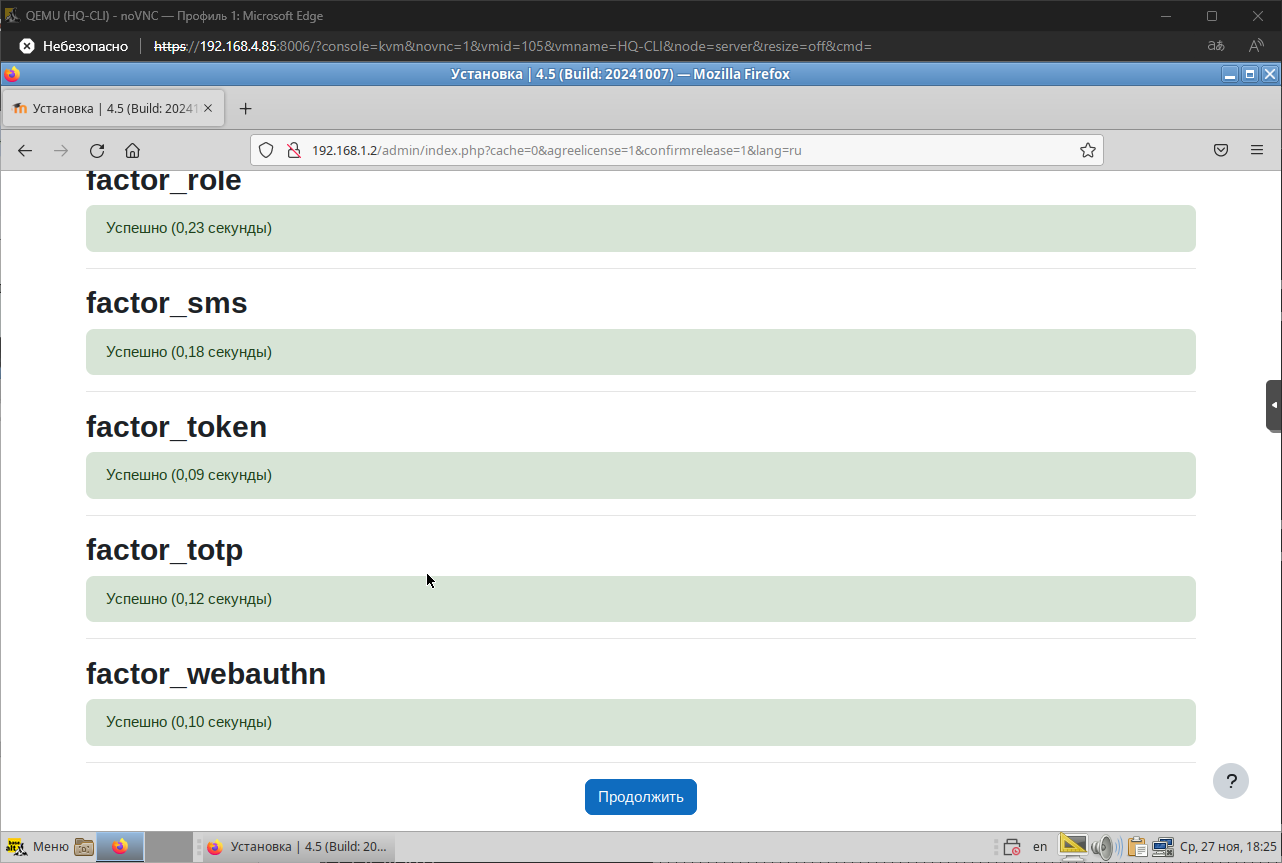
Просматриваем всё ли в статус “**OK**” или “**Проверка**” и прожимаем “**Продолжить**”:



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



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



Далее заполняем обязательные поля для создания основного администратора:

**Логин: admin**

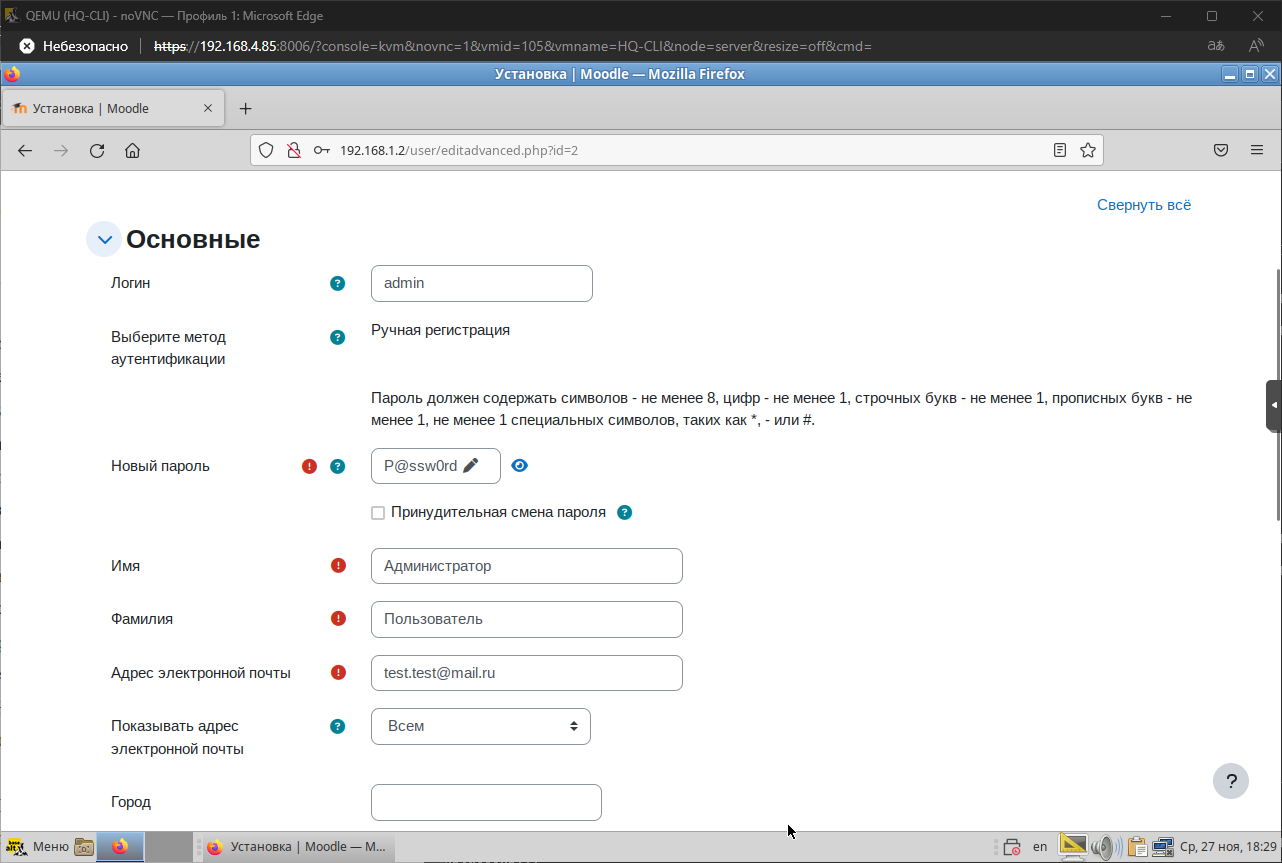
**Новый пароль: P@ssw0rd**

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

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

**Адрес электронной почты:** [**test.test@mail.ru**](mailto:test.test@mail.ru) **(**можно любое**)**

И нажимаем “**Обновить профиль**”:



Теперь заполним ещё некоторые строки на следующем шаге:

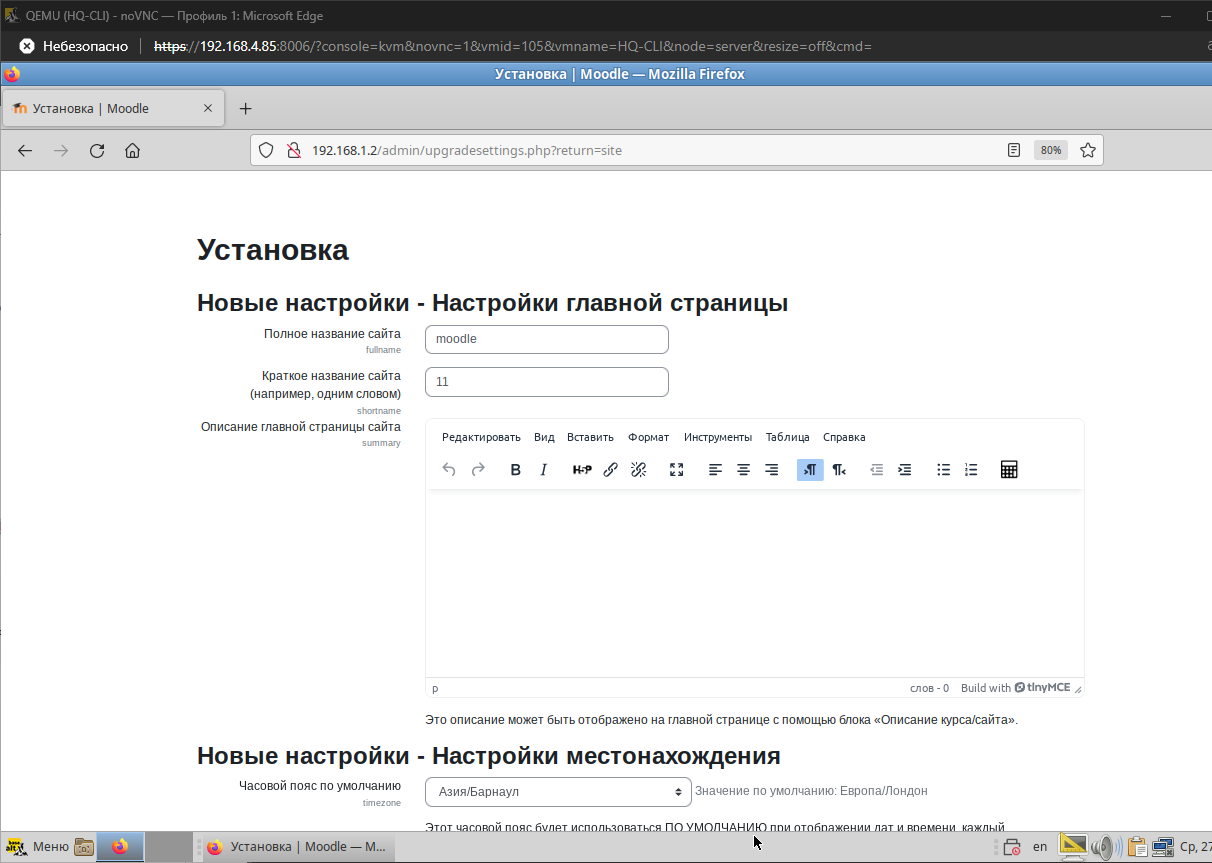
**Полное название сайта: moodle** (можно любое**)**

**Краткое название сайта: 11** (согласно вашему рабочему месту)

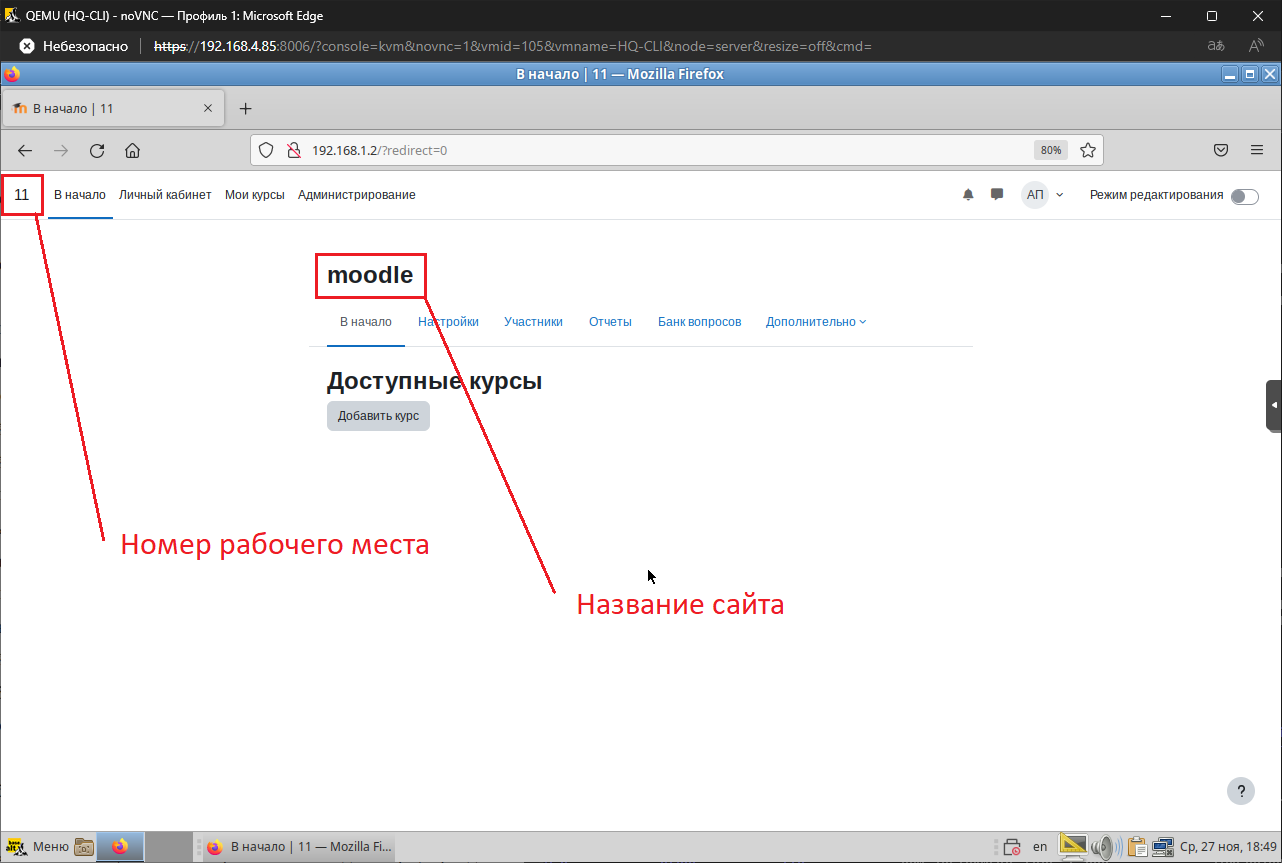
**Настройки местоположения: Азия/Барнаул**

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

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



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

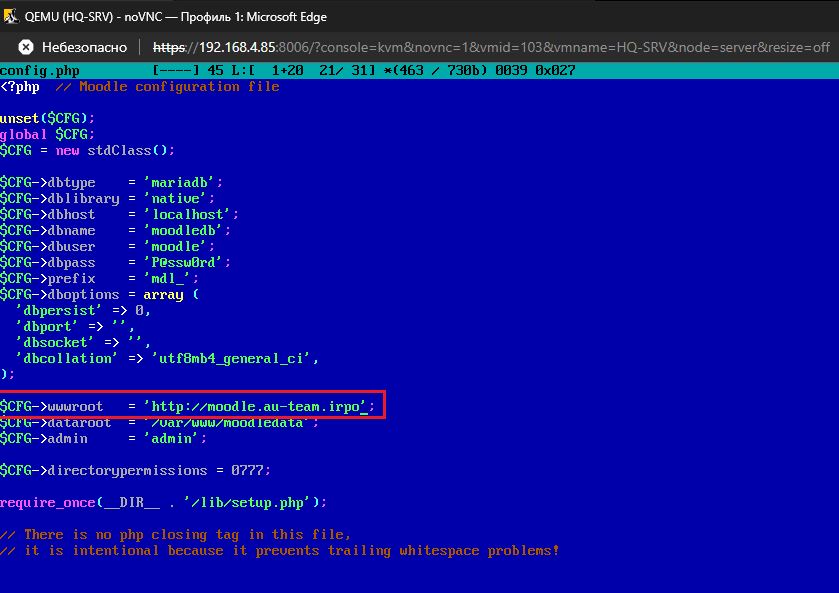


## Настройте веб-сервер nginx как обратный прокси-сервер на HQ-RTR

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

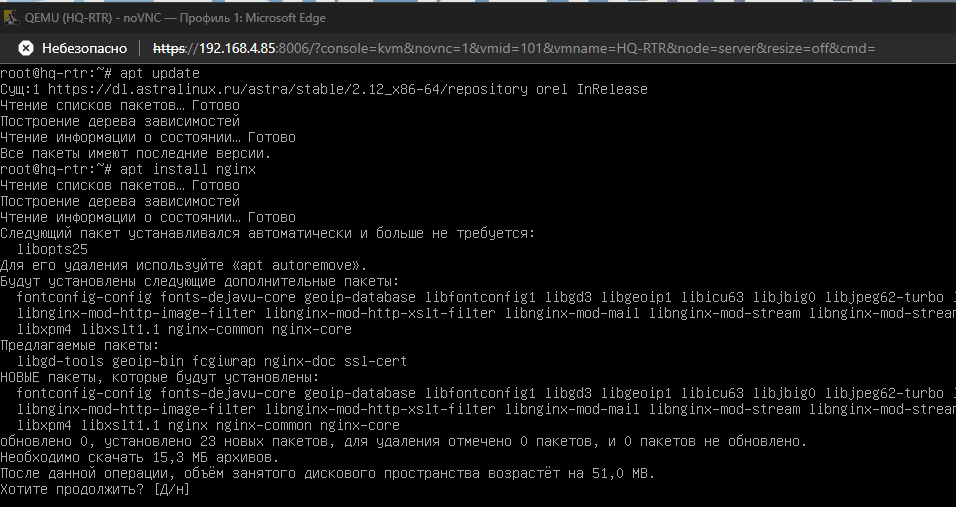
**vim /var/www/html/config.php**

**$CFG->wwwroot = ‘http://sash’;**



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

**apt-get install nginx -y**

****

Создаём новый конфигурационный файл **proxy** и заполняем его следующими строками:

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

**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**

**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;**

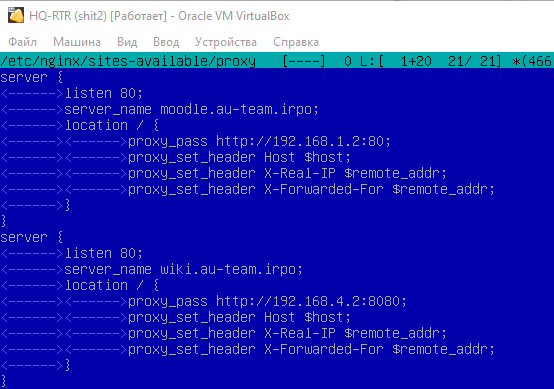
**proxy\_set\_header X-Forwarded-For $remote\_addr;**

**}**

**}**

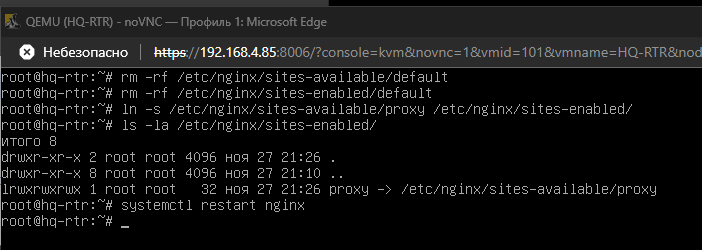
**vim /etc/nginx/nginx.conf**

**include /etc/nginx/sites-enabled.d/\*.conf; - если нет**

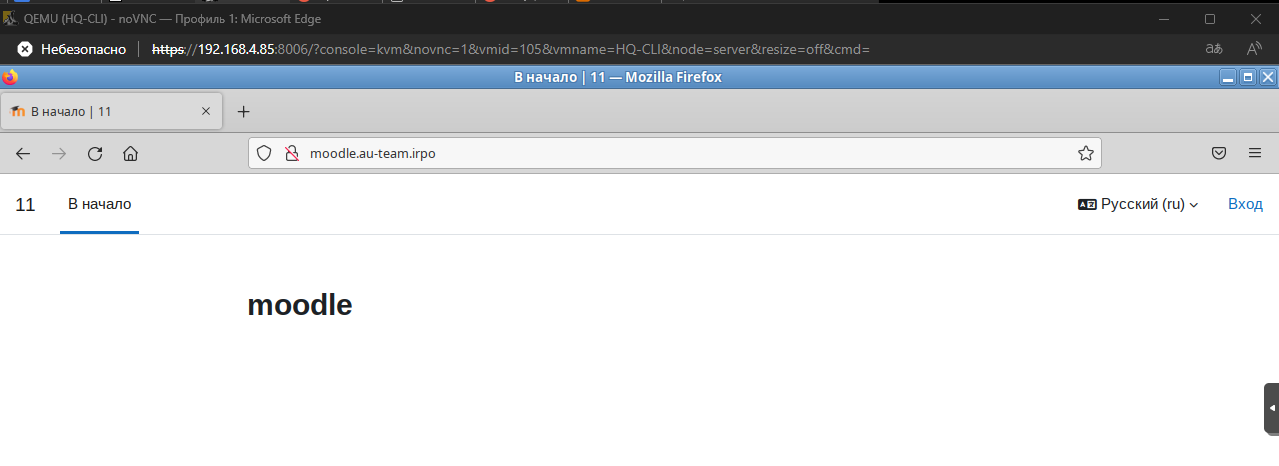
****

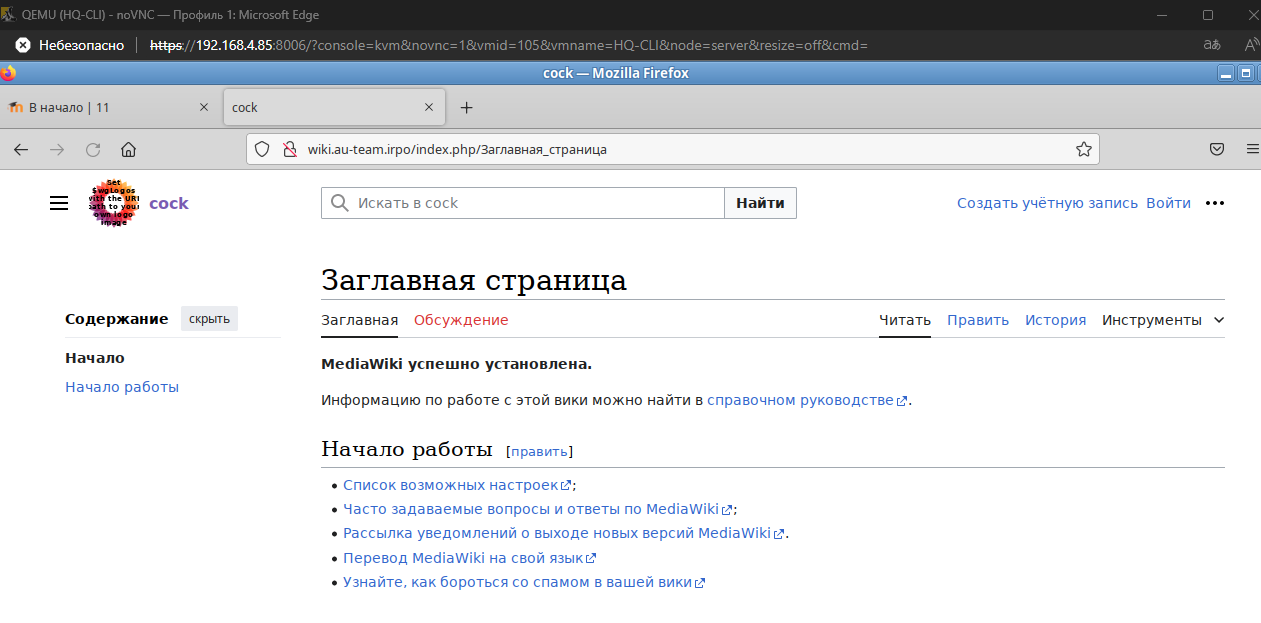
**systemctl restart nginx**

**systemctl enable --now nginx**

****

Проверим работу нашего обратного прокси и зайдем на наши поднятые ранее сайты **moodle** и **wiki** с клиента **HQ-CLI**.





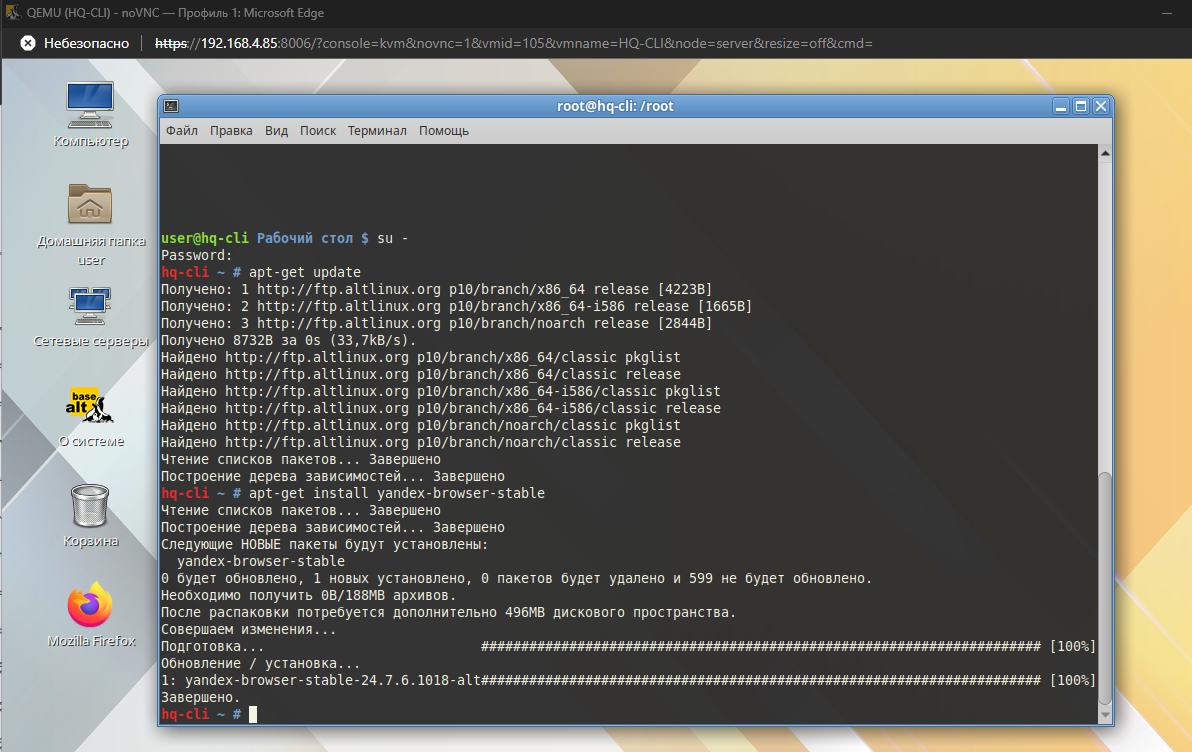
Настройка обратного прокси-сервера завершена.

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

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

**apt-get update**

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

****

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

Должен быть по пути:

**Пуск → Интернет → Yandex Browser**

# Модуль 3

## Настойка центра сертификации на базе HQ-SRV

apt update

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

cd /usr/share/easyrsa3/

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/

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;

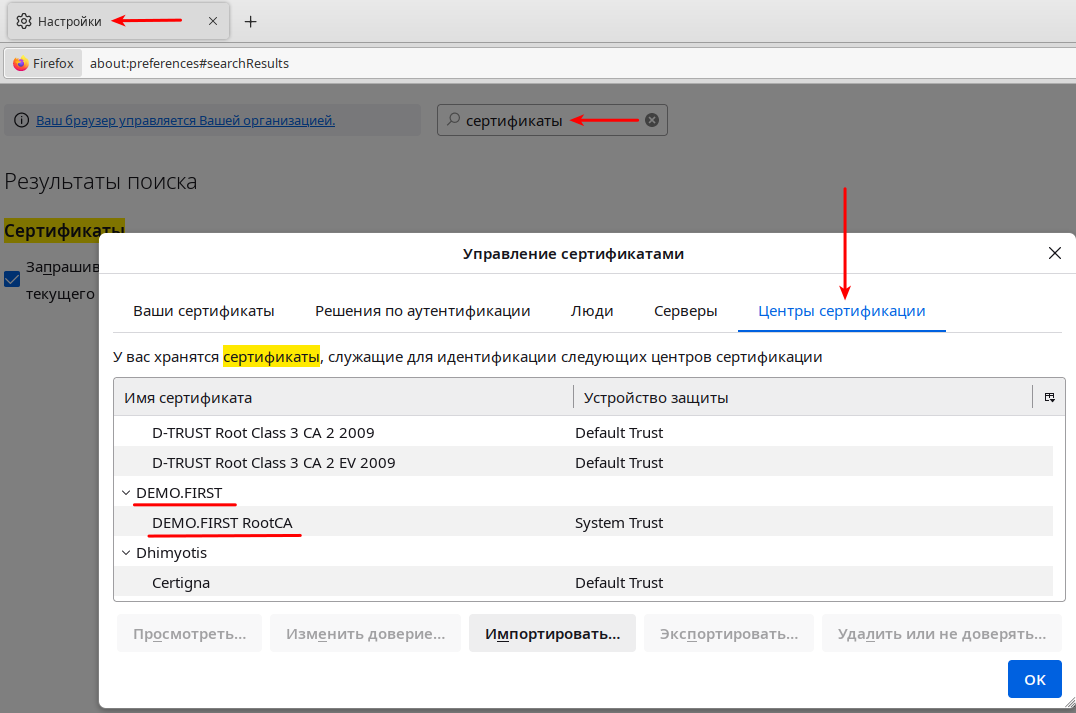
vim /etc/nginx/sites-enabled.d/wiki.conf

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



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

На всякий: <https://docs.tenebraefox.ru/pages/3_module.html>

## Настройте межсетевой экран на маршрутизаторах HQ-RTR и BR-RTR

**на сеть в сторону ISP**

Для выполнения этого задания нам нужно обеспечить работу только нужных протоколов, а именно: HTTP, HTTPS, DNS, NTP, ICMP. А также запретить остальные подключения из сети Интернет во внутреннюю сеть.

**HQ-RTR**

Добавляем правила к уже существующим, которые были настроены в предыдущих модулях:

**iptables -A INPUT -i ens18 -j ACCEPT**

**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**

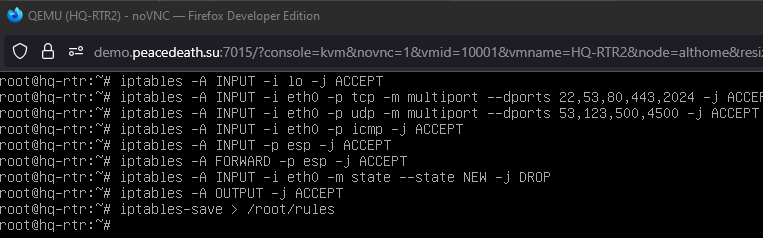
**iptables -A INPUT -p esp -j ACCEPT**

**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

****

В **crontab** изменения вносить не нужно, так как сохранение правил было в тот же файл.

Можно проверить наличие всех правил командой:

**iptables -L -v**

## Настройте принт-сервер cups на сервере HQ-SRV

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

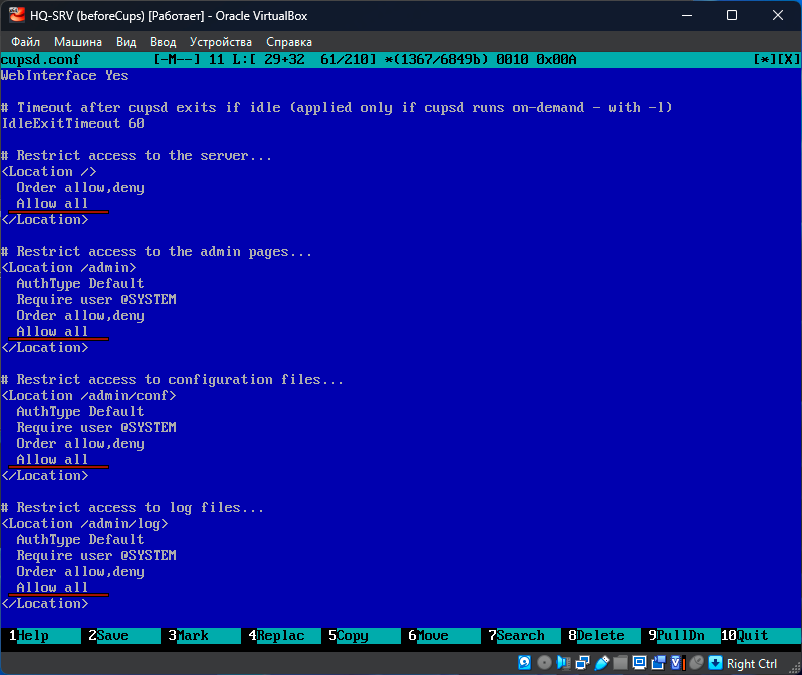
**apt-get install cups cups-pdf -y**

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:



Перезапускаем службу **cups** для применения изменений:

**systemctl restart cups**

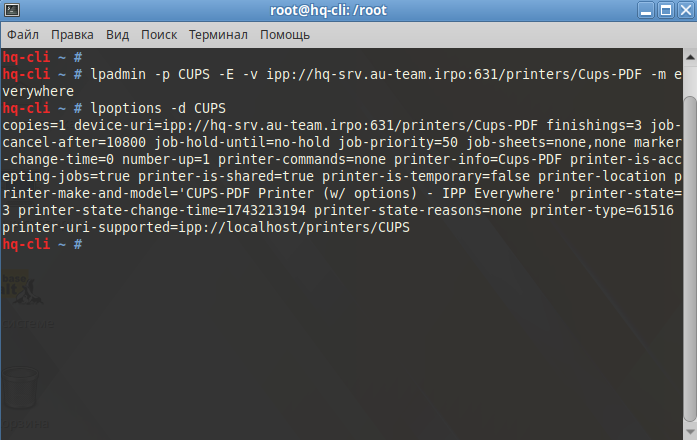
Переходим к подключению клиента **HQ-CLI**

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

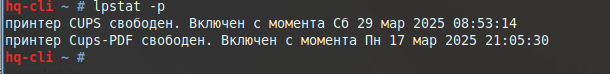
**lpadmin -p CUPS -E -v ipp://hq-srv.au-team.irpo:631/printers/Cups-PDF -m everywhere**

Установим принтер CUPS, как принтер по умолчанию:

**lpoptions -d CUPS**

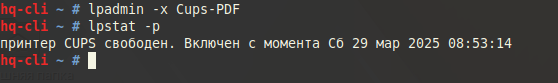
****

Проверяем наличие принтера командой **lpstat -p**

****

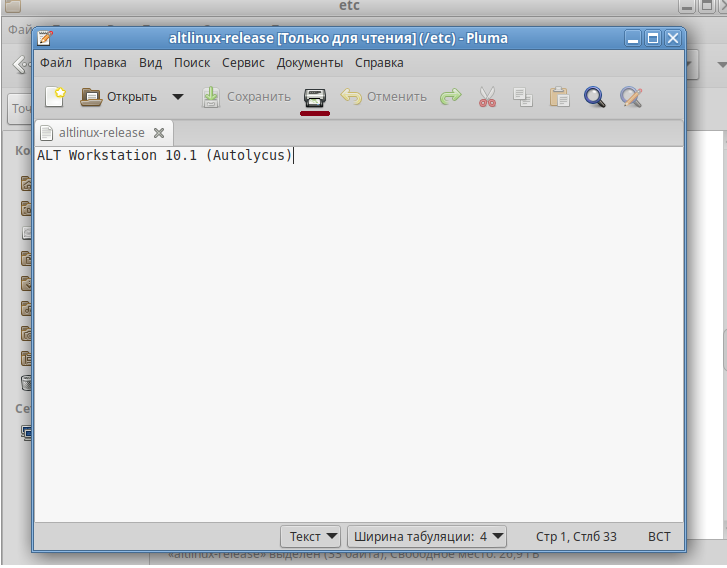
Как можно заметить, принтер **CUPS** успешно подключен. Из-за того, что на **HQ-CLI** также установлен принт-сервер, можно отключить локальный принтер “**Cups-PDF**”, чтобы он не мешал.

**lpadmin -x Cups-PDF**

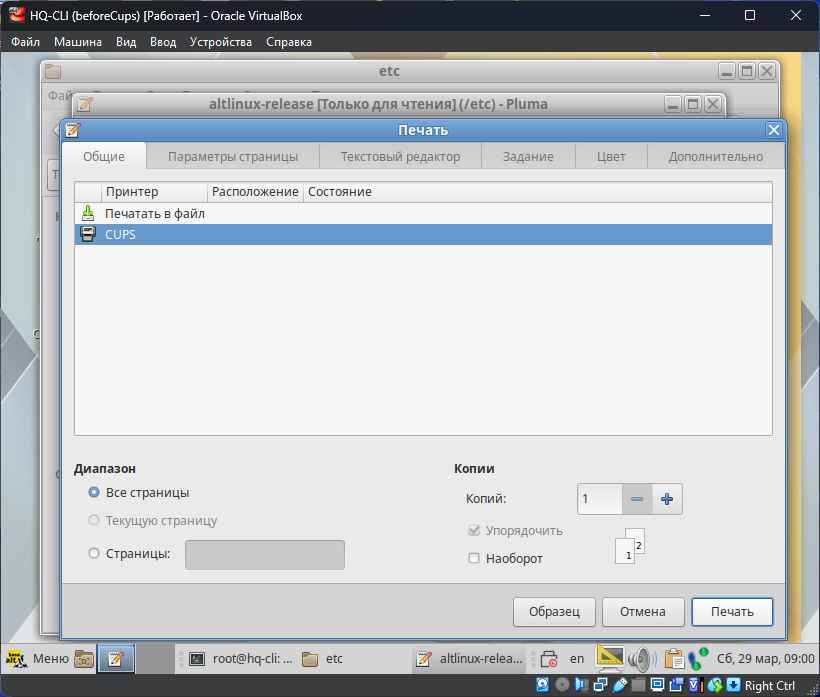
****

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

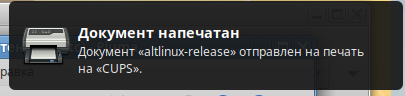
Допустим, откроем файл **/etc/altlinux-release**, нажимаем сверху значок печати:

****

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

****

Сверху появится уведомление, что документ успешно напечатан.

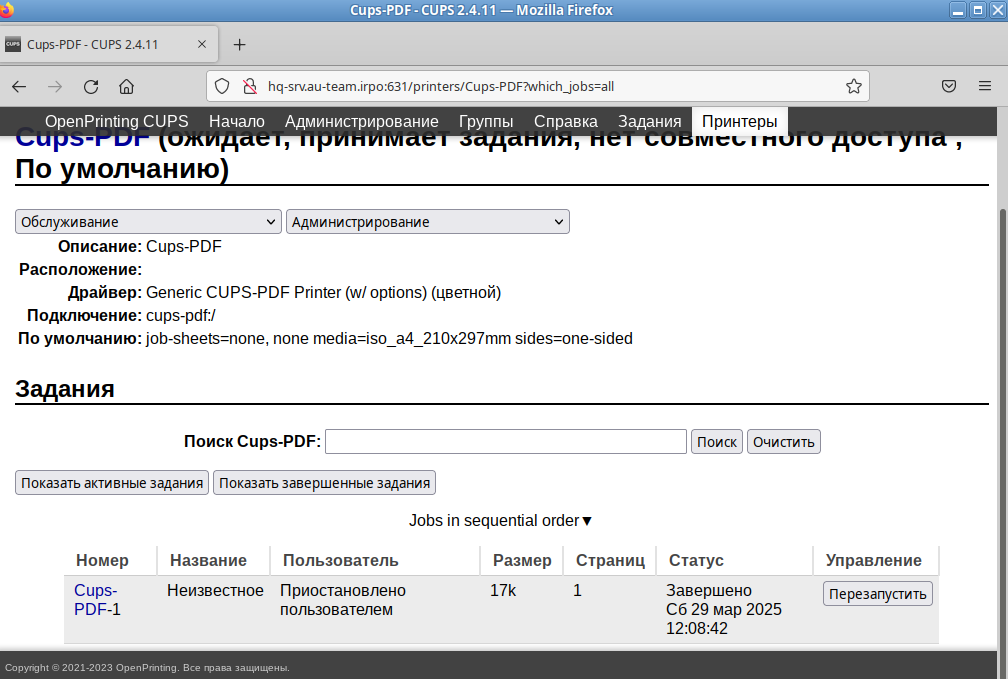


Перейдем в веб-интерфейс CUPS по адресу <https://hq-srv.au-team.irpo:631>

Вкладка **Принтеры**

Выбираем наш принтер.

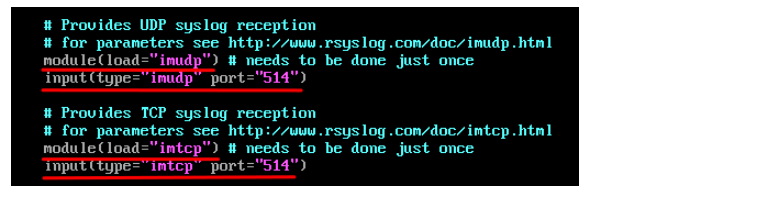
Жмем кнопку **Показать все задания**

****

## HQ-SRV.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



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

**\*.\* ?RemoteLogs**

**& stop**

systemctl enable rsyslog

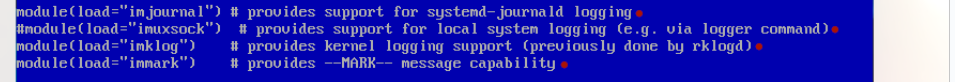
systemctl start rsyslog

На клиентах

apt-get install -y rsyslog-classic

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

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



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**

**}**

## Реализуйте механизм инвентаризации машин HQ-SRV и HQ-CLI через Ansible на BR-SRV

**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 }}"



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**

## 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**

## Настройка иммиграции домена

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