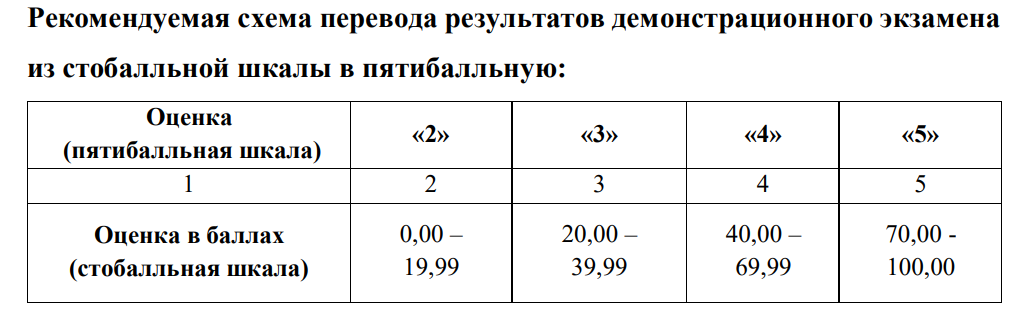
# Доп\_файлы

[Заготовка отчета ССА ДЭ 2024](https://docs.google.com/document/d/1RtYy5Q0NbfWm7SbgjOdMGF3NJk6XihvY74V_xiH1WU8/edit?usp=sharing)

[КОД 09.02.06-1-2024 Том 1 Задание.pdf](https://drive.google.com/file/d/1IFdgffupGRuTenLMtGW9n9KcZSUjj50s/view)

[Оценочные критерии (базовый уровень) - 50 баллов](https://docs.google.com/document/d/12Hwk0eMmT9N4ao1jVAWoMHuOD4JkwCGg/edit)

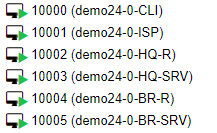


# **Взаимодействие с устройствами**

## **Стенд**

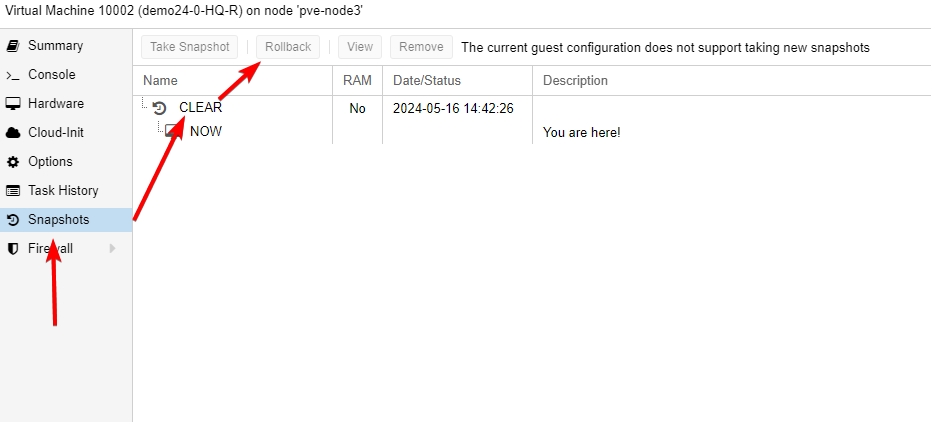
**Чтобы открыть машину в отдельном окне:**

Два раза **ЛКМ** по иконке машины.



**Для того чтобы вайпнуть машину:**

На машине переходим во вкладку **Snapshots**, выбираем **CLEAR** и нажимаем кнопку **Rollback**.



## **Полезные команды**

**Основные команды**

id - кто я

pwd - отобразить текущий каталог

ls - отобразить содержимое каталога

cat - вывести содержимое файла

rm -f - удалить файл

rm -dfr - удалить папку вместе с содержимым

mv /etc/hosts /opt - переместить файл hosts из папки etc в папку opt

cp - скопировать файл

cp -r - рекурсивно скопировать директорию

scp /etc/hosts root@server:/opt - скопировать файл hosts из папки etc нa удаленную машину server в папку opt

chmod +x /opt/start.sh - дaть прaвa нa выполнение файлу start.sh

chmod 777 -R /opt/ - дaть прaвa нa чтение/зaпись всем нa директорию /opt/ рекурсивно

ip a - посмотреть доступные в системе сетевые интерфейсы

ip r - посмотреть информацию о маршрутах v4

ip -6 r - посмотреть информацию о маршрутах v6

**Текстовый редактор nano**

ctrl - o - сохранить (тaкже мoжнo пoменять имя и расположение coхрaняемoгo фaйлa)

ctrl - x - выйти

alt - 6 - копировать строку

ctrl - u - вставить строку

ctrl - k - вырезать строку

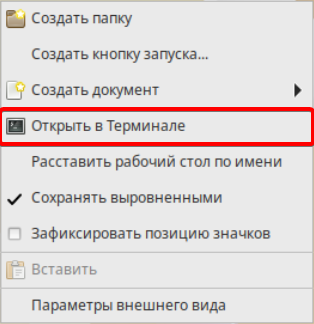
ctrl - r - вставить содержимое файла Х в текущий файл

**Переключение между терминалами -** сочетание клавиш CTRL + SHIFT + f{1-10}

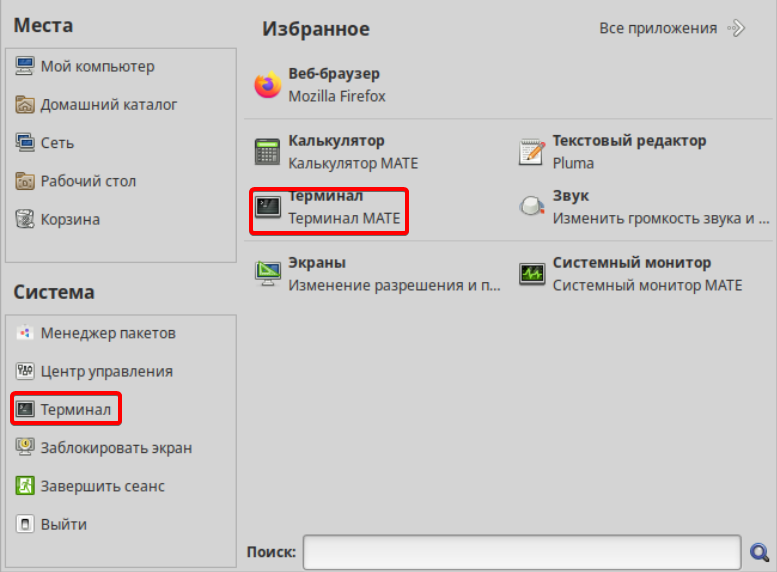
## **CLI**

CLI - это машина с графическим интерфейсом, все основные в ней действия будут производиться в терминале и в браузере:

***Для запуска терминала*** — **ПКМ** по рабочему экрану **“Открыть терминал”**



Либо же можно запустить через меню на нижней панели



Авторизация в машине происходит под обычным пользователем без прав администратора, для их получения можно обратиться к команде:

sudo

Пароль пользователя применившего команду

Постоянно добавлять sudo в начале команд может быть неудобно. Вместо этого можно использовать команду авторизации под пользователем — **su**.

Для авторизацией под пользователем root:

su -

Пароль: toor

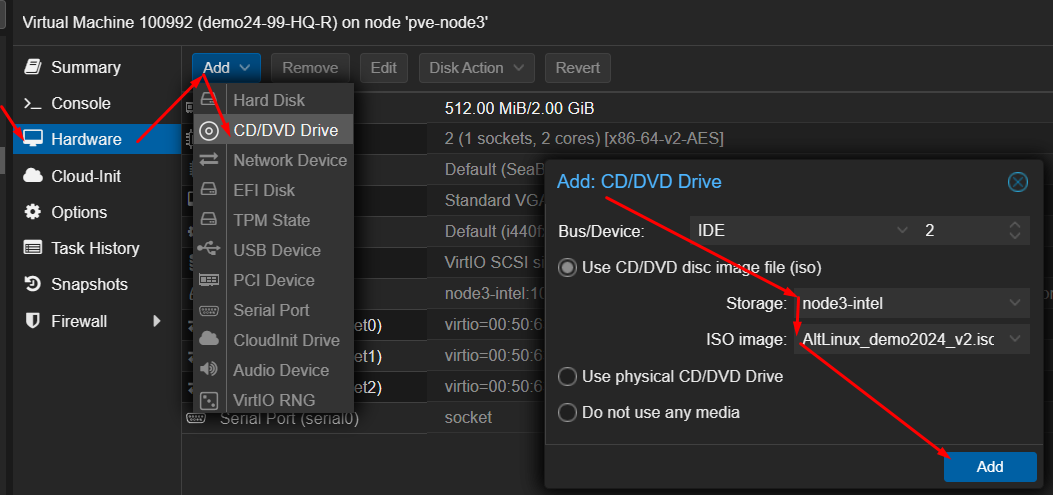
Настройка сети на этой машине: по умолчанию сетью управляет **nmtui**, поэтому - настраивайте через неё или выключите службу NetworkManager и используйте etcnet

## 

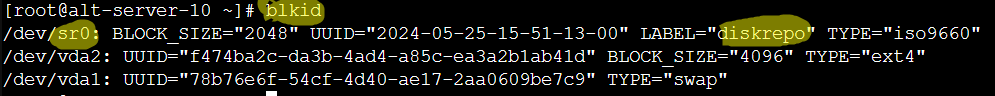
## **Установка пакетов с диска**

Выключить виртуальную машину.

В вкладке Hardware, добавить *CDROM*, выбрать Storage - *node3-intel*, образ - *AltLinux\_demo2024\_vX.iso*



Включить виртуальную машину, проверить что диск подключен



Командой отключить сетевые репозитории

apt-repo rm all

Добавить репозиторий с диска в список репозиториев (при добавлении ввести произвольное название)

apt-cdrom add

Как вернуть сетевые репозитории обратно? Зайти в файл и раскомментировать (убрать решетку в начале )

nano /etc/apt/sources.list.d/alt.list



# **Базовые настройки**

## **Имена машин**

Для начала мы зададим имена устройств согласно заданию

На всех машинах:

hostnamectl set-hostname <имя машины>; exec bash

**exec bash** — перезапуск оболочки bash для отображения нового хостнейма

Проверка:

hostname

## **Создание локальных пользователей**

| Учётная запись | Пароль | Примечание |
| --- | --- | --- |
| Admin | P@ssw0rd | CLI, HQ-SRV, HQ-R |
| Branch admin | P@ssw0rd | BR-SRV, BR-R |
| Network admin | P@ssw0rd | HQ-R, BR-R, BR-SRV |

В Linux имена пользователей по умолчанию регистрозависимы и обычно должны быть в нижнем регистре.

Для добавления пользователя:

useradd admin

useradd branch\_admin

useradd network\_admin

Для задания пароля пользователю:

passwd admin

passwd branch\_admin

passwd network\_admin

Создадим пользователей и зададим им пароль согласно заданию.

Проверка:

id имя\_пользователя

## **Настройка перенаправления трафика**

Перенаправление трафика — это процесс отправки сетевых пакетов с одного сетевого интерфейса на другой на уровне операционной системы. В Linux за это отвечает параметр **ip\_forward**.

На устройствах, которые имеют более чем один сетевой интерфейс необходимо включить перенаправление трафика между интерфейсами.

Для HQ-R, BR-R, ISP :

Для этого необходимо зайти в конфигурационный файл менеджера сетевых интерфейсов и изменим значение параметра на 1:

nano /etc/net/sysctl.conf

net.ipv4.ip\_forward = 1

net.ipv6.conf.all.forwarding = 1

Применим настройки:

sysctl -p

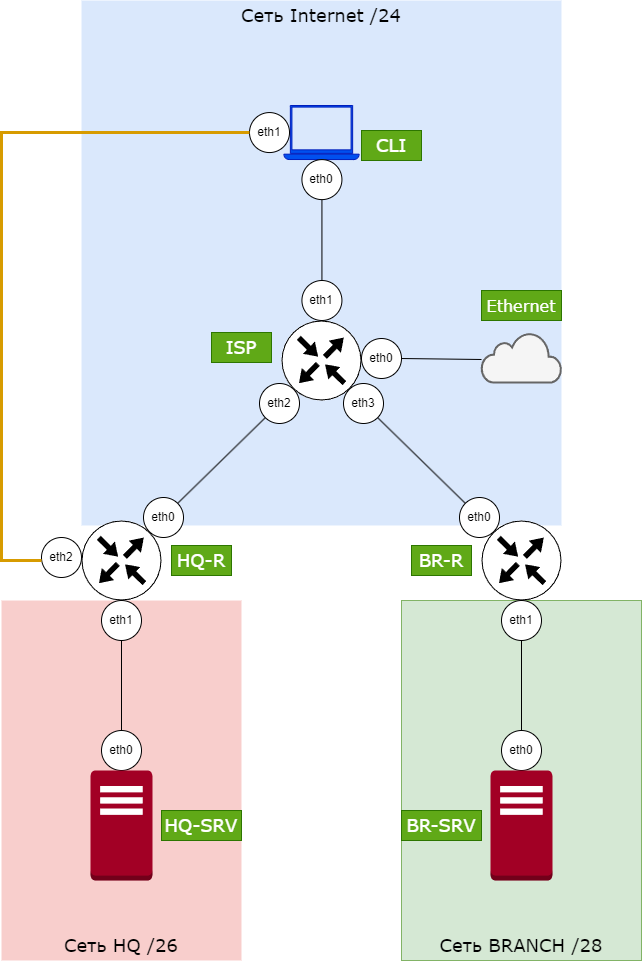
Перезагрузим сетевую службу для вступления изменений:

systemctl restart network  
Проверка:

cat /proc/sys/net/ipv4/ip\_forward

cat /proc/sys/net/ipv6/conf/all/forwarding

## **Настройка ip-адресации**



| Имя устройства | Интерфейс | IPv4 | IPv6 | NIC |
| --- | --- | --- | --- | --- |
| CLI | eth0 | 10.0.1.100/24 | 2001:11::100/64 | ISP-CLI |
| ISP | eth1 | 10.0.1.1/24 | 2001:11::1/64 | ISP-CLI |
| eth2 | 10.0.2.1/24 | 2001:22::1/64 | ISP-HQ |
| eth3 | 10.0.3.1/24 | 2001:33::1/64 | ISP-BR |
| eth0 | DHCP | - | INTERNET |
| HQ-R | eth0 | 10.0.2.100/24 | 2001:22::100/64 | ISP-HQ |
| eth1 | 192.168.200.2/26 | 2000:100::2/122 | HQ-SRV |
| wg0 | 10.0.0.1/24 | 2001:100::1/64 | TUNNEL |
| HQ-SRV | eth0 | 192.168.200.1/26 | 2000:100::1/122 | HQ-R |
| BR-R | eth0 | 10.0.3.100/24 | 2001:33::100/64 | ISP-BR |
| eth1 | 172.16.200.2/28 | 2000:200::2/124 | BR-SRV |
| wg0 | 10.0.0.2/24 | 2001:100::2/64 | TUNNEL |
| BR-SRV | eth0 | 172.16.200.1/28 | 2000:200::1/124 | BR-R |

В качестве системы конфигурации сети используется [etcnet](https://www.altlinux.org/Etcnet)

Для того, чтобы на CLI не сбрасывались настройки ip-адресации:

systemctl disable --now NetworkManager

Посмотреть какие есть сетевые интерфейсы в системе:

ip a

Для каждого сетевого интерфейса необходимо создать директорию по пути **/etc/net/ifaces/<NAME\_INTERFACE>**

**<NAME\_INTERFACE>** имя сетевого интерфейса

**\* - ЭТО НЕ КАКОЙ-ТО ОСОБЕННЫЙ ИНТЕРФЕЙС, НА МЕСТЕ ДАННОГО ЗНАКА ДОЛЖЕН БЫТЬ НОМЕР ИНТЕРФЕЙСА**

mkdir /etc/net/ifaces/eth\*

Конфигурация интерфейса описывается в **/etc/net/ifaces/eth\*/options**.

Теперь перепишем стандартный файл options для всех интерфейсов, чтобы упростить настройку:

nano /etc/net/ifaces/default/options

CONFIG\_IPV6=YES

NM\_CONTROLLED=NO

Теперь перепишем файл options для каждого интерфейса (static):

nano /etc/net/ifaces/eth…/options

TYPE=eth

BOOTPROTO=static

**TYPE** - эта опция определяет тип интерфейса;

**BOOTPROTO** - может быть любым из следующих: static, dhcp, ipv4ll. \* "**static**": использует **ipv4address/ipv6address**;

Если на устройстве более одного интерфейса после изменения параметров в options можно скопировать папку под новый интерфейс:

cp -r /etc/net/ifaces/eth\* /etc/net/ifaces/eth\*

Для назначения ipv4-адреса интерфейса:

echo <ip>/<маска> > /etc/net/ifaces/eth\*/ipv4address

Для назначения ipv6-адреса интерфейса:

echo <ip>/<маска> > /etc/net/ifaces/eth\*/ipv6address

На всех машинах, кроме ISP:

Шлюз по умолчанию настраивается только на интерфейсе, подключенном в сторону вышестоящего роутера (который находится ближе всего к интернету).

Для назначения шлюза по умолчанию ipv4:

echo default via <ip> > /etc/net/ifaces/**eth0**/ipv4route

Для назначения шлюза по умолчанию ipv6:

echo default via <ip> > /etc/net/ifaces/**eth0**/ipv6route

Согласно заданию, устройства должны обращаться к DNS-серверу HQ-SRV. Однако, до момента настройки этого сервера, мы будем использовать 2 DNS-сервера **192.168.200.1** и **192.168.100.1**.

Для назначения DNS-сервера (указывается на любом интерфейсе, лишь один раз на устройстве):

echo nameserver 192.168.200.1 > /etc/net/ifaces/eth0/resolv.conf

echo nameserver 192.168.100.1 >> /etc/net/ifaces/eth0/resolv.conf

На ISP:

Данное устройство имеет интерфейс, выходящий в интернет поэтому он должен принимать сетевые настройки по dhcp.

Перепишем файл options для eth0:

TYPE=eth

BOOTPROTO=dhcp

Чтобы работала маршрутизация от клиента к веткам HQ и BRANCH, необходимо прописать маршрутизацию на ISP:

**Для ipv4**

echo <сеть, куда должны идти пакеты>/маска via <через какой ip должны идти пакеты> > /etc/net/ifaces/ethX/ipv4route

X - номер интерфейса, в сторону которого пойдет трафик

Пример записи - 192.168.200.0/26 via 10.0.2.100

**Для ipv6**

<сеть, куда должны идти пакеты>/маска via <через какой ip должны идти пакеты> > /etc/net/ifaces/eth\*/ipv6route

Пример записи - 2000:100::/122 via 2001:22::100

Перезапустим службу сети:

systemctl restart network

Проверка:

С HQ-SRV и BR-SRV пинг друг-друга:

ping 192.168.200.1

ping 172.16.200.1

## **Настройка NAT**

Для того чтобы устройства могли выходить в интернет необходимо настроить правила NAT.

На ISP:

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables-save >> /etc/sysconfig/iptables

systemctl enable --now iptables

Проверка:  
Ping со всех устройств до 8.8.8.8 и nslookup ya.ru

Ping между собой (из внутренней подсети в другую работать не будет до настройки vpn)

# SSH финальная настройка

На всех устройствах:

Создадим текст для баннера:

nano /etc/openssh/banner

Authorized access only!

Исправим все параметры в конфигурации под требования задания:

nano /etc/openssh/sshd\_config

Доступ под суперпользователем:

PermitRootLogin yes

Banner:

Banner /etc/openssh/banner

Переведите на нестандартный порт:

Port 4444

Ограничьте ввод попыток до 4:

MaxAuthTries 4

Отключите пустые пароли:

PermitEmptyPasswords no

Установите предел времени аутентификации до 5 минут:

LoginGraceTime 5m

После перезагружаем службу ssh:

systemctl restart sshd

Проверка:

ssh до всех устройств

ssh -p 4444 root@<ip>

## 

## **SSH по 2222**

Настроим подключение по SSH к HQ-SRV по порту 2222 с помощью правил контроля трафика **iptables**.

На HQ-SRV:

Для начала необходимо разрешить входящие соединения на порт 2222.

Для ipv4:

iptables -A INPUT -p tcp --dport 2222 -j ACCEPT

Перенаправим трафик, поступающий на порт 2222, к порту 4444.

Для ipv4:

iptables -t nat -A PREROUTING -p tcp --dport 2222 -j REDIRECT --to-port 4444

Сохраним наши

iptables-save >> /etc/sysconfig/iptables

Включаем службу и добавляем её в автозагрузку

systemctl enable --now iptables

Проверка:

Со всех устройств:

ssh -p 2222 root@192.168.200.1

## 

## 

## **SSH со всех устройств, кроме CLI**

На HQ-SRV:

В конец файла конфигурации добавить

nano /etc/openssh/sshd\_config

Match Address 10.0.1.\*

PubkeyAuthentication no

PasswordAuthentication no

Match Address 10.0.1.\* - ip-адрес клиента

После перезагружаем службу ssh:

systemctl restart sshd

Проверка:

С CLI:

ssh -p 2222 root@192.168.200.1

**SSH НЕ РАБОТАЕТ И НЕ БУДЕТ (это хорошо)**

ISP:

ssh -p 2222 root@192.168.200.1

**А ТУТ РАБОТАТЬ ДОЛЖЕН**

# 

# **Iperf**

iperf3 - это бесплатная кроссплатформенная программа с открытым исходным кодом, для измерения пропускной способности сети в реальном времени. Это один из самых мощных инструментов для тестирования максимально достижимой полосы пропускания в IP-сетях (поддерживает IPv4 и IPv6).

С iperf3 можно настроить несколько параметров, связанных с синхронизацией, буферами и протоколами, такими как TCP, UDP, SCTP. Это удобно для операций настройки производительности сети.

Чтобы получить максимальную или, скорее, улучшенную производительность сети, необходимо увеличить пропускную способность, а также латентность сети. Однако, прежде чем приступить к реальной настройке, нужно выполнить определенные тесты, чтобы собрать общую статистику производительности сети, которая будет определять процесс настройки.

Результаты включают интервал времени в секундах, передаваемые данные, полосу пропускания (скорость передачи), потери и другие полезные параметры производительности сети. Тесты в первую очередь предназначены для помощи в настройке TCP-соединений по определенному пути.

На HQ-R и ISP:

Установим необходимую утилиту:

apt-get install iperf3

Включим и добавим её в автозагрузку:

systemctl enable --now iperf3

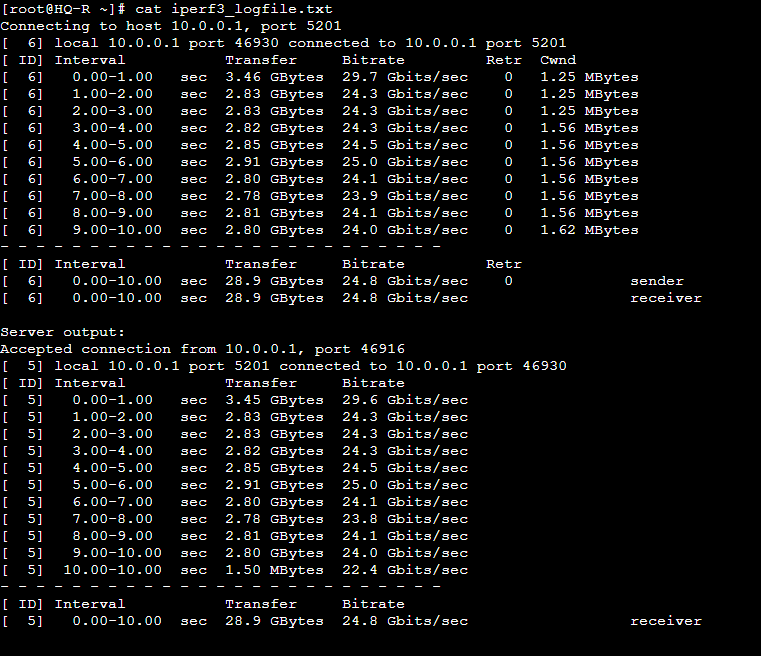
На HQ-R:

Запустим iperf3 в режиме клиента:

iperf3 -c 10.0.2.1 --get-server-output

при помощи параметра **"--get-server-output"** - можно получить более детальный результат:

если необходимо получить результаты сервера в выводе клиента;

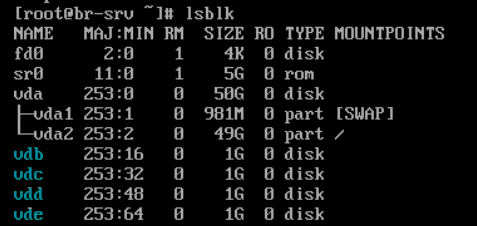


Далее согласно заданию необходимо предоставьте описание пропускной способности канала со скриншотами

# **RAID**

На BR-SRV:

Убедитесь что есть дополнительные диски:



Создаём зеркало из минимально необходимого количества дисков для этого уровня RAID:

mdadm --create --verbose /dev/md0 -l 5 -n 4 /dev/vdb /dev/vdc /dev/vdd /dev/vde

**--create**: указывает mdadm, что необходимо создать новый RAID массив.

**--verbose**: включает подробный вывод процесса создания массива.

**/dev/md0**: это имя нового RAID устройства, которое будет создано.

**-l 5**: указывает уровень RAID, в данном случае это RAID 5. RAID 5 использует чередование (striping) с распределенной четностью, что обеспечивает баланс между производительностью и надежностью данных.

**-n 4**: указывает количество дисков, которые будут использоваться в массиве, в данном случае 4 диска.

Информацию о созданном массиве дописываем в конец конфигурационного файла:

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

Форматируем созданный виртуальный диск под файловую систему ext4:

mkfs.ext4 /dev/md0

Создаём папку по заданию в которую будет монтироваться зеркальный массив и пробуем смонтировать диск вручную:

mkdir /mnt/storage

mount /dev/md0 /mnt/storage

df -h

Если диск смонтирован успешно, тогда можно обновить информацию о постоянных дисковых накопителях в системе, partprobe(если этого не сделать, после следующей перезагрузки виртуальный диск переименуется с md0 в md127 и не смонтируется в файловую систему):

partprobe /dev/md0

Добавляем возможность автоматически монтировать диск при старте системы:

**fstab** - это текстовый файл, который содержит информацию о различных файловых системах и устройствах хранения информации в системе, в нём определяется, как диск и/или раздел будут использоваться и как будут встроены в остальную систему.

nano /etc/fstab

Строки файла содержат следующие поля, в каждой строке опции отделяются пробелами или табуляцией:

Что монтируем — некоторое блочное устройство, которое должно быть примонтировано; Куда монтируем — точка монтирования - путь в корневой файловой системе к каталогу в который будет смонтировано устройство; Тип файловой системы монтируемого раздела; Опции монтирования файловой системы (используем опцию defaults что эквивалентно *rw,suid,dev,exec,auto,nouser,async*); Индикатор необходимости делать резервную копию (как правило не используется и равно 0); Порядок проверки файловой системы раздела (0- не проверять, 1 - устанавливается для корневой файловой системы, 2 - для остальных разделов).

/dev/md0 /mnt/storage ext4 defaults 0 0

Пробуем смонтировать все устройства, указанные в fstab, тем самым проверив корректность новой конфигурации

umount -a

mount -a

Проверка:

reboot

df -h

# **Скрипты**

Составим backup скрипты для сохранения конфигурации сетевых устройств, а именно HQ-R BR-R.

На HQ-R:

Создадим директорию, в которую будет выводиться результат работы скрипта. В целом можно создать новую директорию где удобно.:

mkdir /opt/backup

Для удобства создадим папку для нашего скрипта:

mkdir /etc/scripts

Создадим наш скрипт:

nano /etc/scripts/backup-script.sh

#!/bin/bash

backup\_files="/home /etc"

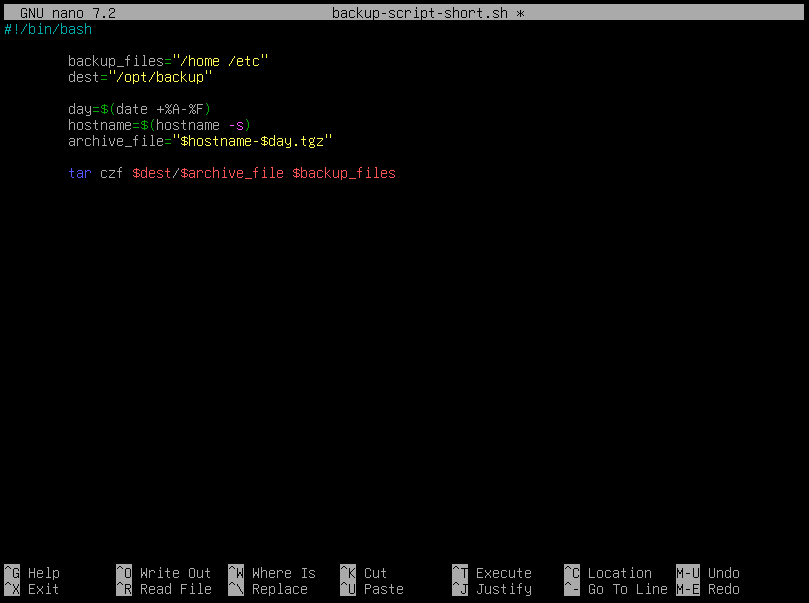
dest="/opt/backup"

day=$(date +%A-%F)

hostname=$(hostname -s)

archive\_file="$hostname-$day.tgz"

tar czf $dest/$archive\_file $backup\_files



Оба варианта будут работать одинаково (за исключением вывода сообщений статуса).

**Переменные**

***backup\_files*** - это те файлы и директории, которые будут скопированы. Они указываются в "кавычках" через пробел. Например, можно добавить для копирования конкретный файл, если прописать путь до него.

***dest*** - куда будут скопированы файлы. Эта директория должна присутствовать в системе. В целом может быть любой удобной директорией.

***day*** - переменная, сохраняющая информацию о дате в указанном формате.

***hostname*** - переменная, сохраняющая имя системы.

***archive\_file*** - переменная, сохраняющая имя конечного файла в формате имя системы (hostname) + день (day) + формат файла (tar).

**Команды**

**tar czf** - запуск архиватора tar. В указанную директорию (переменная dest), будут сложены скопированные файлы (bacup\_files) и названы значением переменной archive\_file.

**date +%A-%F** - дата в формате: день недели - год - месяц - день. Команда исполняется и сохраняется в переменную day

**hostname -s** - команда выводящая имя системы. Результат сохраняется в переменную hostname.

**Пояснения к элементам скрипта**

По сути переменные могут называться как угодно, главное, чтобы была сохранена логика работы скрипта.

Можно написать скрипт хоть так, но разобраться что и за что отвечает в разы сложнее.

#!/bin/bash

a="/home /etc"

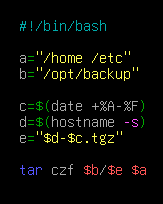
b="/opt/backup"

c=$(date +%A-%F)

s=$(hostname -s)

w="$d-$c.tgz"

tar czf $b/$e $a



Выдадим права на исполнение:

chmod +x /etc/scripts/backup-script.sh

**Запуск скрипта**

Для запуска скрипта:

/etc/scripts/backup-script.sh

Проверить созданный скриптом архив:

tar -tf путь до архива | less

Имя архива можно узнать выполнив команду ls :



Пример вывода tar -tf:



Выведется длинный список всех скопированных файлов и каталогов.Здесь и видно, что каталог /home и /etc скопированы и заархивированы вполне успешно.

Под названием каталога расписаны файлы.

Выйти из этого режима можно с помощью клавиши **q**

На BR-R:

Создадим точно такие же директории

mkdir /opt/backup

mkdir /etc/scripts

Далее, чтобы не писать заново скрипт скопируем уже созданный с помощью буфера обмена

nano /etc/scripts/backup-script.sh

Выдадим права на исполнение:

chmod +x /etc/scripts/backup-script.sh

Далее необходимо создать отчёт по работе скрипта

# 

# Cups

CUPS - это стандартная система печати с открытым исходным кодом, разработанная Apple Inc. для MacOS® и других UNIX®-подобных операционных систем.   
По заданию необходимо настроить виртуальный принтер с помощью CUPS для возможности печати документов из Linux-системы на сервере BR-SRV.

На BR-SRV:

Удалить старую конфигурацию cups

rm -dfr /etc/cups/

Установим необходимый пакет

apt-get install --reinstall cups cups-pdf

Перейдем в конфигурацию и выдадим права доступа

nano /etc/cups/cupsd.conf

Port 631

<Location />

Order allow,deny

Allow all

</Location>

На всех машинах:

Создадим файл конфигурации клиента и укажем сервер печати:

nano /etc/cups/client.conf

ServerName 172.16.200.1

Перезапустим сервис:

systemctl restart cups

Создадим тестовый файл и отправим его на печать

echo "Тестовая страница" > <имя машины>.txt

lp <имя машины>.txt

Проверка:

На BR-SRV:

Просмотрим все полученные принтером файлы

ls /var/spool/cups-pdf/ANONYMOUS/

Иногда пакет

# 

# ClamAV

На HQ-SRV и BR-SRV:

Установим необходимые пакеты:

apt-get install clamav clamav-db

Далее по заданию необходимо настроить сканирование системы раз в сутки с сохранением отчёта. При этом сканирование должно проводится при условии, что от пользователей нет нагрузки.

export EDITOR='/usr/bin/nano'

Добавим запись в планировщик

crontab -e

0 0 \* \* \* clamscan -r / >> /root/clamscan.log

# 

# **VPN туннель (Между офисами HQ и BRANCH)**

По заданию, необходимо настроить связь между офисами HQ и BRANCH с использованием защищенного туннеля. Соответственно, самое простое, что можно настроить, это VPN по технологии Wireguard. В нашей топологии VPN сервером будет являться HQ-R, а клиентом - BR-R.

На HQ-R, BR-R:

Устанавливаем необходимый пакет

apt-get install wireguard-tools wireguard-tools-wg-quick

Создаем папку wireguard

mkdir /etc/wireguard

На HQ-R:

Утилита генерации ключей требует, чтобы все создаваемые ключи будут недоступны всем кроме владельца и группы-владельца файла. Поэтому указываем что для всех создаваемых файлов в текущей папке автоматически установить права доступа *077*

umask 077

Генерируем приватный ключ для сервера

wg genkey > privkey\_server

Из приватного ключа сервера генерируем публичный ключ сервера

cat /root/privkey\_server | wg pubkey > pubkey\_server

ls -lsa

С приватным и публичным ключём клиента тоже самое

wg genkey > privkey\_client

cat /root/privkey\_client | wg pubkey > pubkey\_client

Убеждаемся, что все файлики со всеми ключами создались

ls -lsa

Копировать необходимые ключи мы будем через SCP.

scp -P 4444 /root/\*key\* root@172.16.200.2:~

Конфигурация клиента и сервера похожа, но с незначительными отличиями, находится по пути */etc/wireguard/wg0.conf*, по умолчанию файл не существует

nano /etc/wireguard/wg0.conf

В секции *[Interface]* описывается работа локального интерфейса wireguard и режим работы

[Interface]

Указывается ранее сгенерированный приватный ключ сервера

PrivateKey = *содержимое privkey\_server*

Адрес виртуального интерфейса wg0 который будет отображаться в системе

Address = 10.0.0.1/24, 2001:100::1/64, fe80::200:00ff:fe00:0001/64

Порт, на который будут приниматься подключения от клиентов

ListenPort = 1234

Table = off

В секции *[Peer]* указывается ранее сгенерированный из приватного ключа публичный ключ клиента

[Peer]

PublicKey = *содержимое pubkey\_client*

И список сетей которым разрешена маршрутизация через туннель

AllowedIPs = 0.0.0.0/0, ::/0

После завершения редактирования конфигурации не забываем перезапустить службу wireguard`а и добавить в автозагрузку

systemctl restart wg-quick@wg0

systemctl enable wg-quick@wg0

ip a

На BR-R:

Конфигурация клиента:

nano /etc/wireguard/wg0.conf

[Interface]

Тут главное не запутаться с ключами. В секции *[Interface]* всегда используется приватный ключ

PrivateKey = *содержимое privkey\_client*

Address = 10.0.0.2/24, 2001:100::2/64, fe80::200:00ff:fe00:0002/64

Table = off

[Peer]

А в секции *[Peer]* всегда используется публичный ключ удаленной стороны

PublicKey = *содержимое pubkey\_server*

AllowedIPs = 0.0.0.0/0, ::/0

Адрес VPN сервера с указанием порта подключения

Endpoint = 10.0.2.100:1234

Обязательно указываем эту опцию, чтобы клиент периодически проверял доступность сервера и поддерживал соединение туннеля

PersistentKeepalive = 10

После завершения редактирования конфигурации не забываем перезапустить службу wireguard`а и добавить в автозагрузку

systemctl restart wg-quick@wg0

systemctl enable wg-quick@wg0

ip a

Проверка:

На HQ-R:

ping 10.0.0.2

На BR-R:

ping 10.0.0.1

# **FRR**

FRRouting — это программный пакет, который на текущий момент позволяет реализовать работу таких протоколов, как OSPF, IS-IS, EIGRP, RIP, PBR, VRRP и т.д. на всех современных UNIX-системах, включая Linux и BSD. Для конфигурирования используется набор Cisco-like команд.

На HQ-R:

Установим необходимый пакет:

apt-get install frr

Далее необходимо в конфигурационном файле FRR выбрать протоколы, которые мы будем использовать - ospf и ospfv3:

nano /etc/frr/daemons

ospfd=yes

ospf6d=yes

Включим и добавим наш сервер в автозагрузку:

systemctl enable --now frr

Перейдем в интегрированную командную строку управления FRR:

vtysh

Теперь необходимо зайти в конфигурационный режим:

conf t

Отредактируем наши интерфейсы - для включения их в зоны ipv6:

interface wg0

ipv6 ospf6 area 0

exit

interface eth1

ipv6 ospf6 area 0

exit

Создадим конфигурацию для ospf(ipv4), зададим идентификатор и объявим наши подсети:

router ospf

ospf router-id 1.1.1.1

network 10.0.0.0/24 area 0

network 192.168.200.0/26 area 0

exit

Для ospf6(ipv6) зададим только идентификатор, так как подсети были объявлены выше через интерфейсы:

router ospf6

ospf6 router-id 11.11.11.11

Выйдем из режима конфигурации:

exit

Сохраним нашу конфигурацию:

do wr

На BR-R:

Проводим такую же настройку, но с нужными данными

apt-get install frr

nano /etc/frr/daemons

ospfd=yes

ospf6d=yes

systemctl enable --now frr

vtysh

conf t

interface wg0

ipv6 ospf6 area 0

exit

interface eth1

ipv6 ospf6 area 0

exit

router ospf

ospf router-id 2.2.2.2

network 10.0.0.0/24 area 0

network 172.16.200.0/28 area 0

exit

router ospf6

ospf6 router-id 22.22.22.22

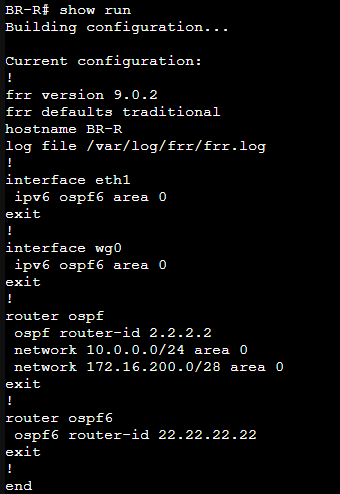
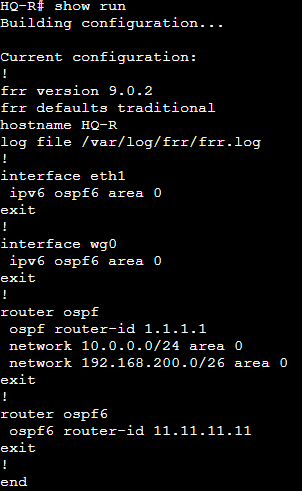
exit

do wr

**Проверка:**

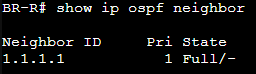
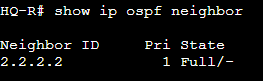
Проверим конфигурацию устройства:

show run



Проверим, образовалось ли соседство для ospf(ipv4):

show ip ospf neighbor



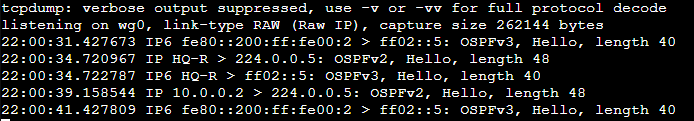
Проверим, образовалось ли соседство для ospf6(ipv6):

show ipv6 ospf6 neighbor



Также можно посмотреть какие пакеты проходят через интерфейс:

tcpdump -i wg0



На HQ-SRV:

traceroute 172.16.200.1

На BR-SRV:

traceroute 192.168.200.1

Маршрутизация должна идти через туннель.

Далее необходимо обосновать выбор протокола маршрутизации и составить топологию сети L3

Уровень L3 (также именуемый «сетевым уровнем») — это уровень абстракции, который отражает то, как происходит пересылка пакетов через промежуточные маршрутизаторы.

# **DHCP**

На HQ-R:

Для того чтобы настроить автоматическое распределение ip-адресов установим dhcp-server:

apt-get install dhcp-server

Укажем в параметре **«DHCPDARGS»‎** сетевой интерфейс, через который будет работать DHCP-сервер:

/etc/sysconfig/dhcpd - для dhcpd.service

**eth1**- интерфейс смотрящий в локальную сеть офиса HQ

DHCPDARGS = e*th1*

Для настройки зоны ipv4 необходимо создать конфигурационный файл зоны, его можно скопировать из имеющегося примера:

cp /etc/dhcp/dhcpd.conf.example /etc/dhcp/dhcpd.conf

Либо же его можно написать вручную.

Приведем файл конфигурации к следующему виду:

nano /etc/dhcp/dhcpd.conf

# dhcpd.conf

default-lease-time 6000;

max-lease-time 72000;

authoritative;

subnet 192.168.200.0 netmask 255.255.255.192 {

range 192.168.200.3 192.168.200.62;

option domain-name-servers 192.168.200.1;

option routers 192.168.200.2;

}

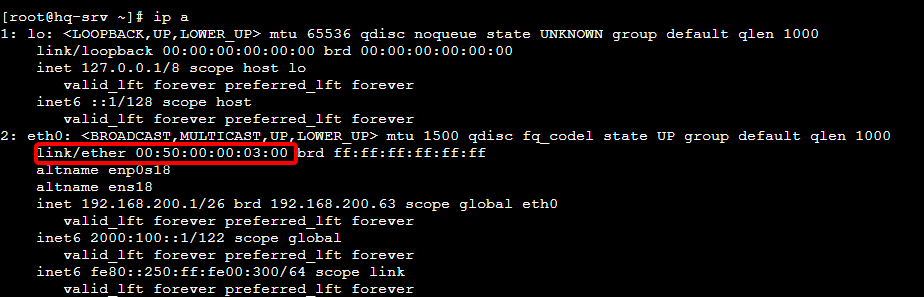
host hq-srv {

hardware ethernet *XX:XX:XX:XX:XX*;

fixed-address 192.168.200.1;

}

*где, XX:XX:XX:XX:XX - mac адрес интерфейса на HQ-SRV*



Конфигурационный файл можно проверить через утилиту **«dhcpd»:**

dhcpd -t -cf /etc/dhcp/dhcpd.conf



В случае ошибки в описании конфигурационного файла - в выводе данной утилиты будет описано что не так.

Запускаем и добавляем в автозагрузку службу **dhcpd** (для IPv4):

systemctl enable --now dhcpd

Далее необходимо изменить параметр получения ip-адреса на интерфейсе HQ-SRV.

На HQ-SRV:

nano /etc/net/ifaces/eth0/options

BOOTPROTO=dhcp,static

После перезагружаем службу интернета:

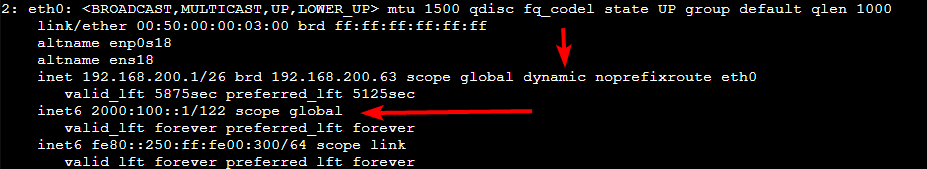
systemctl restart network

Проверка:

На HQ-SRV:

ip a

У ipv4 адреса интерфейса должна появиться приписка **dynamic**, а ipv6 адрес остаться на месте



Для ipv6 настройка производится не будет, потому что критериев на это нет (наверное), да и настройка ~~ужасна (на самом деле я просто ненавижу Linux)~~ может нарушить конфигурацию сервера

# **NTP**

**Chrony** представляет собой универсальную реализацию протокола сетевого времени (NTP). Он может синхронизировать системные часы с серверами NTP и эталонными часами. Он также может работать как сервер и равноправный узел, предоставляя службу времени другим компьютерам в сети.

На HQ-R:

Установим необходимый пакет:

apt-get install chrony

Запустим службу и добавим её в автозагрузку:

systemctl enable --now chronyd

Отредактируем файл конфигурации:

nano /etc/chrony.conf

По заданию, сервер HQ-R синхронизируется со своим Loopback, для это заменим ***#pool*** на значение серверна

server 127.0.0.1 iburst

Указываем стратум, по заданию - 5

local stratum 5

Разрешаем все запросы

allow all

systemctl restart chronyd

systemctl status chronyd

На всех других устройствах :

apt-get install chrony

systemctl enable --now chronyd

nano /etc/chrony.conf

server 192.168.200.2 iburst

ISP

server 10.0.2.100 iburst

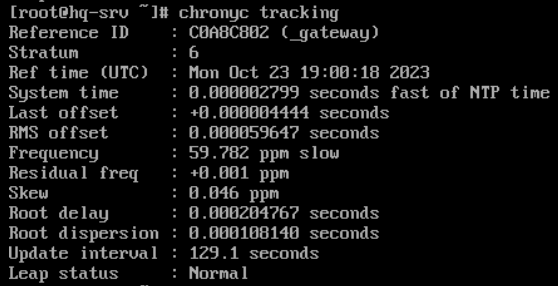
systemctl restart chronyd

По заданию все устройства и сервера настроены на московский часовой пояс (UTC +3)

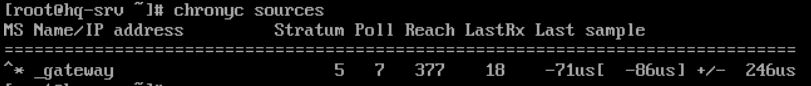
При установке ос на устройства уже был выставлен нужный регион.

Проверка:

chronyc tracking



chronyc sources



timedatectl



# **DNS прямой и обратной зоны.**

На HQ-SRV:

Устанавливаем DNS сервер

apt-get install bind

Включаем сервис и добавляем его в автозагрузку

systemctl enable --now bind

Копируем уже готовый файл зоны для наших будущих прямых зон

cp /etc/bind/zone/localhost /etc/bind/zone/hq.work

cp /etc/bind/zone/localhost /etc/bind/zone/branch.work

Начинаем редактировать в соответствии с заданием

Для hq.work:

nano /etc/bind/zone/hq.work

$TTL 86400

@ IN SOA hq-srv.hq.work. root.hq.work. (; параметры оставляем по умолчанию)

@ IN NS hq-srv.hq.work.

hq-srv IN A 192.168.200.1

hq-r IN A 192.168.200.2

Для branch.work:

nano /etc/bind/zone/branch.work

$TTL 86400

@ IN SOA hq-srv.branch.work. root.branch.work. (; параметры оставляем по умолчанию)

@ IN NS hq-srv.branch.work.

hq-srv IN A 192.168.200.1

br-srv IN A 172.16.200.1

br-r IN A 172.16.200.2

Далее, необходимо сделать две обратные зоны для обоих локальных подсетей

cp /etc/bind/zone/hq.work /etc/bind/zone/hq.192

cp /etc/bind/zone/branch.work /etc/bind/zone/branch.172

nano /etc/bind/zone/hq.192

$TTL 86400

@ IN SOA hq-srv.hq.work. root.hq.work. (; параметры оставляем по умолчанию)

@ IN NS hq-srv.hq.work.

1 IN PTR hq-srv

2 IN PTR hq-r

nano /etc/bind/zone/branch.172

@ IN SOA hq-srv.branch.work. root.branch.work. (; параметры оставляем по умолчанию)

@ IN NS hq-srv.branch.work.

1 IN PTR br-srv

2 IN PTR br-r

Добавляем в конец файла описание наших зон в конфигурационный файл DNS сервера

nano /etc/bind/local.conf

zone "hq.work" {

type master;

file "/etc/bind/zone/hq.work";

};

zone "branch.work" {

type master;

file "/etc/bind/zone/branch.work";

};

zone "200.168.192.in-addr.arpa" {

type master;

file "/etc/bind/zone/hq.192";

};

zone "200.16.172.in-addr.arpa" {

type master;

file "/etc/bind/zone/branch.172";

};

Немного редактируем конфигурацию самого DNS сервера, добавляем нижеуказанные опции внутрь секции ***options***

nano /etc/bind/options.conf

Включаем прослушку всех портов

listen-on {any; };

listen-on-v6 { any; };

Включаем пересылку запросов незаписанных на сервере на сторонний сервер

forwarders { 192.168.100.1; };

Разрешаем запросы и запросы к кэшу от всех клиентов

allow-query { any; };

allow-query-cache { any; };

Отключаем проверку подписи запросов

dnssec-validation no;

Прописываем доступ к папке с зонами

chmod 777 /etc/bind/zone/ -R

Перезапускаем службу

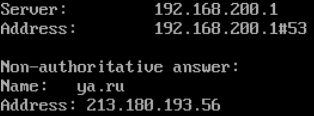
systemctl restart bind

Проверка со всех устройств:

named-checkconf /etc/named.conf

named-checkconf -z /etc/named.conf

nslookup ya.ru



nslookup hr-r.hq.work

nslookup br-r.branch.work

# **FreeIPA**

На HQ-SRV:

Донастроить DNS

cp /etc/bind/zone/localhost /etc/bind/zone/domain.work

nano /etc/bind/zone/domain.work

$TTL 86400

@ IN SOA hq-srv.domain.work. root.domain.work. (; параметры оставляем по умолчанию)

@ IN NS hq-srv.domain.work.

hq-srv IN A 192.168.200.1

Дописать новую зону

nano /etc/bind/local.conf

zone "domain.work" {

type master;

file "/etc/bind/zone/domain.work";

};

Перезапускаем службу

systemctl restart bind

По заданию необходимо настроить сервер домена на базе HQ-SRV через web интерфейс, выбор технологий и типа обосновать.

Изменим имя устройство на полное доменное:

hostnamectl set-hostname hq-srv.domain.work

Увеличить размер временной ФС:

mount -o remount,size=1G /dev/shm

Установим необходимый пакет

apt-get install freeipa-server

Далее необходимо запустить интерактивную установку:

ipa-server-install

На первый вопрос, нужно ли сконфигурировать DNS-сервер BIND, следует ответить отрицательно (в скобках написан шаблонный ответ, если нажать enter - применится именно он; если ваш выбор отличается - необходимо написать yes или no):

Do you want to configure integrated DNS (BIND)? [no]:

Server host name [hq-srv.domain.work]:

Please confirm the domain name [domain.work]:

Please provide a realm name [DOMAIN.WORK]:

Задать пароль для Directory Manager:

Directory Manager password: P@ssw0rd

Password (confirm): P@ssw0rd

Задать пароль для администратора сервиса:

IPA admin password: P@ssw0rd

Password (confirm): P@ssw0rd

Продолжаем установку

NetBIOS domain name [DOMAIN]:

Do you want to configure CHRONY with NTP server or pool address? [no]

Вылезет сообщение с подтверждением настроек:

The IPA Master Server will be configured with:

Hostname: hq-srv.domain.work

IP address(es): 192.168.200.1, 2000:100::1

Domain name: domain.work

Realm name: DOMAIN.WORK

The CA will be configured with:

Subject DN: CN=Certificate Authority,O=DOMAIN.WORK

Subject base: O=DOMAIN.WORK

Chaining: self-signed

Continue to configure the system with these values? [no]: yes

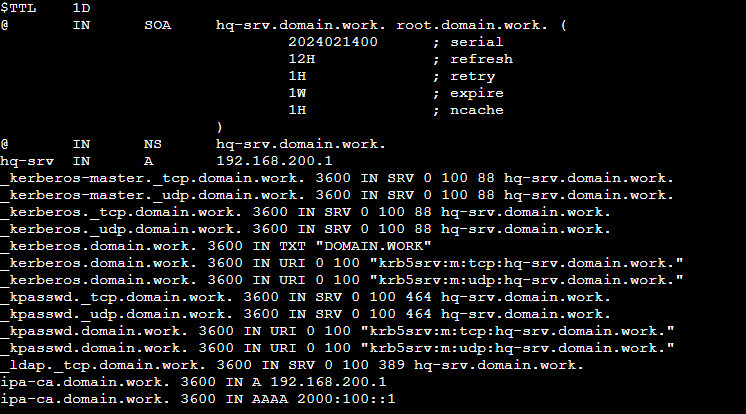
После этого начнется процесс установки (5 минут).

…  
При успешном завершении необходимо добавить сконфигурированные записи в созданную ранее dns-зону нашего домена:

cat /tmp/.private/root/ipa\*.db >> /etc/bind/zone/domain.work

Далее проверим, что новые записи добавились в нашу зону:

cat /etc/bind/zone/domain.work



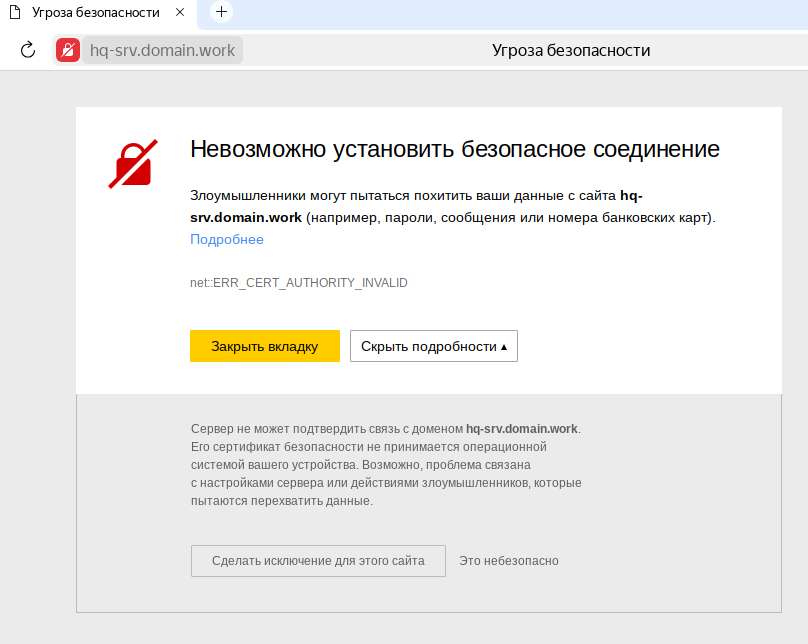
Перезагрузим службу DNS-сервера для вступления в силу изменений:

systemctl restart bind

Далее необходимо перейти на CLI и в браузере перейти по адресу HQ-SRV:

192.168.200.1

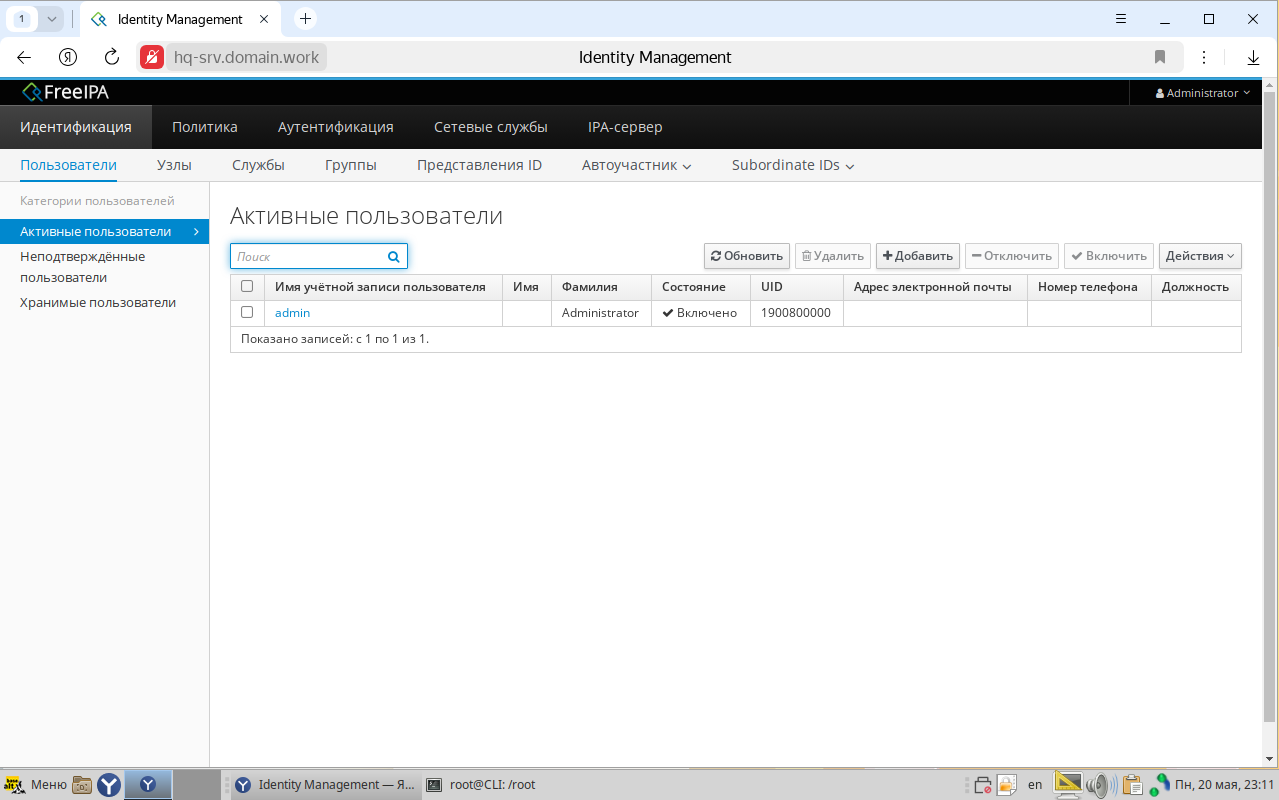
Нас встрети предупреждение о том, что соединение небезопасно - нажмем «Показать подробности» > «Сделать исключение для этого сайта»:



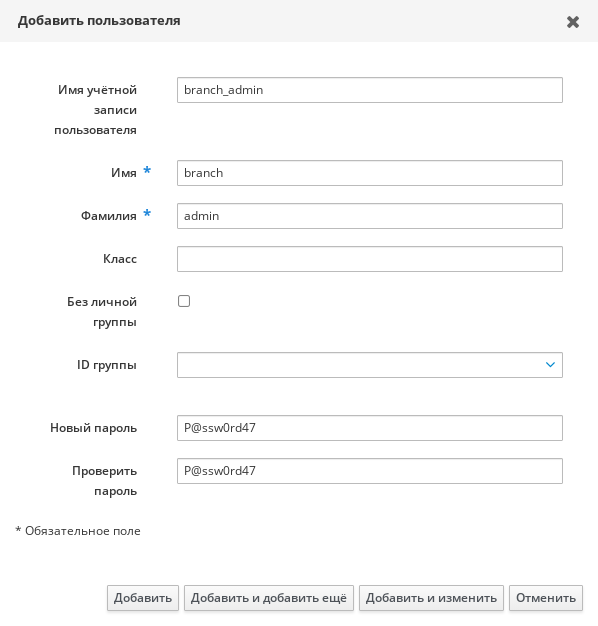
Нас встречает страница авторизации - введём данные от **admin**

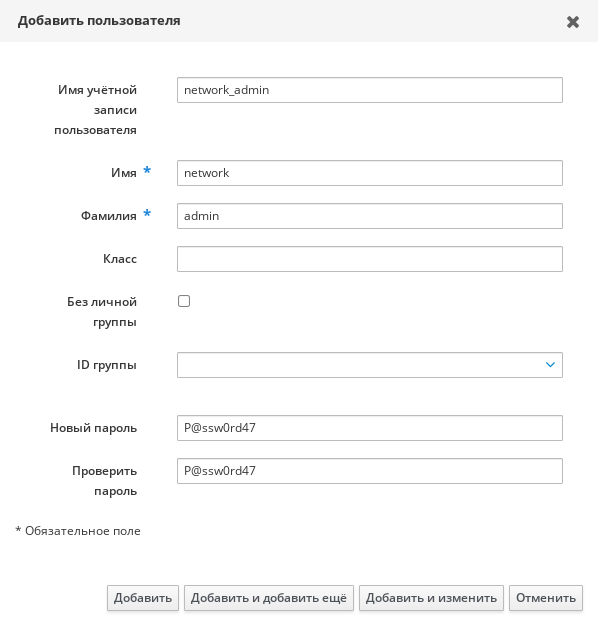


После авторизации мы попадаем на главную страницу нашего сервиса.



Теперь необходимо добавить пользователей для создания в будущем общих файловых ресурсов.





Далее необходимо изменить домашний каталог доменных пользователей, чтобы не возникало конфликтов с созданными до этого локальными учетками:

Для этого перейдем на каждого пользователя и изменим нужный параметр:







На BR-SRV и CLI:

По заданию данные устройства необходимо сделать клиентами созданного нами домена. Для этого необходимо установить пакет клиента:

*apt-get install freeipa-client*

Изменим имена устройств на полное доменное:

hostnamectl set-hostname br-srv.domain.work

и

hostnamectl set-hostname cli.domain.work

соответсвенно

Далее запускаем интерактивную установку:

*ipa-client-install*

При правильной настройки DNS и домена, нас должно встретить следующее сообщение:

Discovery was successful!

Это означает что устройство смогло найти созданный нами домен и подцепить с него все настройки.

Do you want to configure CHRONY with NTP server or pool address? [no]:

Client hostname: cli.domain.work

Realm: DOMAIN.WORK

DNS Domain: domain.work

IPA Server: hq-srv.domain.work

BaseDN: dc=domain,dc=work

и

Client hostname: br-srv.domain.work

Realm: DOMAIN.WORK

DNS Domain: domain.work

IPA Server: hq-srv.domain.work

BaseDN: dc=domain,dc=work

Continue to configure the system with these values? [no]: yes

Теперь необходимо ввести учетные данные пользователя, который имеет право на ввод устройств в домен - в нашем случае это **admin**:

User authorized to enroll computers: admin

Password for admin@DOMAIN.WORK: P@ssw0rd

Далее идёт установка.

Проверка:

kinit admin

kinit branch\_admin

kinit network\_admin

Также необходимо проверить создаются ли домашние папки доменных пользователей, для этого необходимо войти под нужными учетками:

Login: …@domain.work

Password: …

При аутентификации устройство попросит сменить пароль - просто введем все три раза **P@ssw0rd**

ls /home/ipa

# **SMB FREEIPA**

Samba — популярный выбор в качестве файлового сервера CIFS в Linux, а благодаря SSSD можно проще интегрировать файловый сервер Samba в домен IPA с преимуществами единого входа и поддержки доверенных пользователей Active Directory.

На HQ-SRV:

apt-get install freeipa-server-trust-ad

Если при инициализации сервера (реплики)

Получить для суперпользователя билет Kerberos администратора домена:

kinit admin

Действуя с правами суперпользователя выполним установку:

ipa-adtrust-install --add-sids --add-agents

admin password: *P@ssw0rd*

Enable trusted domains support in slapi-nis? [no]:yes

В процессе установки на все предложения донастройки отвечаем **yes**

Далее необходимо создать папки которые мы будем использовать как общие

mkdir /share

mkdir /share/Branch\_Files

mkdir /share/Network

mkdir /share/Admin\_Files

Создадим описание наших ресурсов, для удобства создадим их в директории службы

nano /etc/samba/Branch\_Files.txt

[Branch\_Files]

path = /share/Branch\_Files

writable = yes

browseable = yes

force user = branch\_admin@domain.work

valid users = branch\_admin@domain.work

nano /etc/samba/Admin\_Files.txt

[Admin\_Files]

path = /share/Admin\_Files

writable = yes

browseable = yes

force user = admin@domain.work

valid users = admin@domain.work

nano /etc/samba/Network.txt

[Network]

path = /share/Network

writable = yes

browseable = yes

force user = network\_admin@domain.work

valid users = network\_admin@domain.work

Импортируем созданные описания ресурсов в конфигурацию samba:

net conf import /etc/samba/Branch\_Files.txt Branch\_Files

net conf import /etc/samba/Admin\_Files.txt Admin\_Files

net conf import /etc/samba/Network.txt Network

На BR-SRV, HQ-SRV:

Отредактируем файл конфигурации отвечающий за предоставление доступа к выполнению задач с правами суперпользователя:

nano /etc/sudoers

Необходимо раскомментировать следующую строку;

WHEEL\_USERS ALL=(ALL:ALL) ALL

Добавим доменных пользователей в группу контроля и создадим папки для монтирования

На HQ-SRV:

gpasswd -a admin wheel

mkdir /mnt/Admin\_Files

На BR-SRV:

gpasswd -a branch\_admin wheel

gpasswd -a network\_admin wheel

mkdir /mnt/Branch\_Files

mkdir /mnt/Network

На BR-SRV, HQ-SRV:

Составим скрипты для автоматического монтирования папок при входе доменного пользователя в систему и отключения их при его выходе из сессии.

mkdir /etc/scripts

Для удобства создадим папку со скриптами и перейдем в неё

cd /etc/scripts

Создадим скрипт для автоматического монтирование, он будет получать билет для пользователя и с помощью него проходить аутентификацию kerberos и монтировать в нужную папку. С помощью echo мы передаем пароль от учётной записи, чтобы не писать вручную каждый раз, а при помощи clear очищаем консоль для лучшего вида.

На BR-SRV:

Для Branch\_Files:

nano /etc/scripts/mount\_Branch\_Files.sh

#!/bin/bash

if [ "$(id -un)" = "branch\_admin" ]; then

echo "P@ssw0rd" | kinit branch\_admin

echo "P@ssw0rd" | sudo -S mount -t cifs //hq-srv.domain.work/Branch\_Files /mnt/Branch\_Files -o user=branch\_admin,sec=krb5

clear

fi

Создадим скрипт для отключения папки при выходе пользователя

nano /etc/scripts/unmount\_Branch\_Files.sh

#!/bin/bash

if [ "$(id -un)" = "branch\_admin" ]; then

echo "P@ssw0rd" | sudo -S umount /mnt/Branch\_Files

fi

Сделаем скрипты исполняемыми

chmod +x /etc/scripts/mount\_Branch\_Files.sh

chmod +x /etc/scripts/unmount\_Branch\_Files.sh

Далее откроем файл .bash\_profile в домашнем каталоге пользователя branch\_admin

nano /home/ipa/branch\_admin/.bash\_profile

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/mount\_Branch\_Files.sh

Откроем файл .bash\_logout в этом же каталоге

nano /home/ipa/branch\_admin/.bash\_logout

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/unmount\_Branch\_Files.sh

Для Network:

nano /etc/scripts/mount\_Network.sh

#!/bin/bash

if [ "$(id -un)" = "network\_admin" ]; then

echo "P@ssw0rd" | kinit network\_admin

echo "P@ssw0rd" | sudo -S mount -t cifs //hq-srv.domain.work/Network /mnt/Network -o user=branch\_admin,sec=krb5

clear

fi

nano /etc/scripts/unmount\_Network.sh

#!/bin/bash

if [ "$(id -un)" = "network\_admin" ]; then

echo "P@ssw0rd" | sudo -S umount /mnt/Network

fi

Сделаем скрипты исполняемыми

chmod +x /etc/scripts/mount\_Network.sh

chmod +x /etc/scripts/unmount\_Network.sh

.bash\_profile

nano /home/ipa/network\_admin/.bash\_profile

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/mount\_Network.sh

.bash\_logout

nano /home/ipa/network\_admin/.bash\_logout

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/unmount\_Network.sh

На HQ-SRV:

Для Admin\_Files:

nano /etc/scripts/mount\_Admin\_Files.sh

#!/bin/bash

if [ "$(id -un)" = "admin" ]; then

echo "P@ssw0rd" | kinit admin

echo "P@ssw0rd" | sudo -S mount -t cifs //hq-srv.domain.work/Admin\_Files /mnt/Admin\_Files -o user=admin,sec=krb5

clear

fi

nano /etc/scripts/unmount\_Admin\_Files.sh

#!/bin/bash

if [ "$(id -un)" = "admin" ]; then

echo "P@ssw0rd" | sudo -S umount /mnt/Admin\_Files

fi

Сделаем скрипты исполняемыми

chmod +x /etc/scripts/mount\_Admin\_Files.sh

chmod +x /etc/scripts/unmount\_Admin\_Files.sh

.bash\_profile

nano /home/ipa/admin/.bash\_profile

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/mount\_Admin\_Files.sh

.bash\_logout

nano /home/ipa/admin/.bash\_logout

Добавим путь до нашего скрипта в конец файла:

/etc/scripts/unmount\_Admin\_Files.sh

Проверка:

Заходим под нужным пользователем и проверяем, что папка смонтирована

mount

Должна появиться похожая запись



Также можно посмотреть с помощью

df -h

Можно создать файл в смонитрованой папке

touch /mnt/<название папки пользователя под которым зашли>/test.txt

И посмотреть появился ли файл в общей папки на HQ-SRV

ls /share/<название папки>/

# 

# LMS Apache

Система управления обучением (LMS) — это программное приложение, которое помогает администрировать. Нам необходимо его сконфигурировать, для этого:

На BR-SRV:

apt-get update

apt-get install MySQL-server

apt-get install moodle moodle-apache2 moodle-local-mysql

systemctl enable --now mysqld.service

systemctl enable --now httpd2.service

mysql -u root

CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

CREATE USER 'moodleuser'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'moodlepasswd';

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

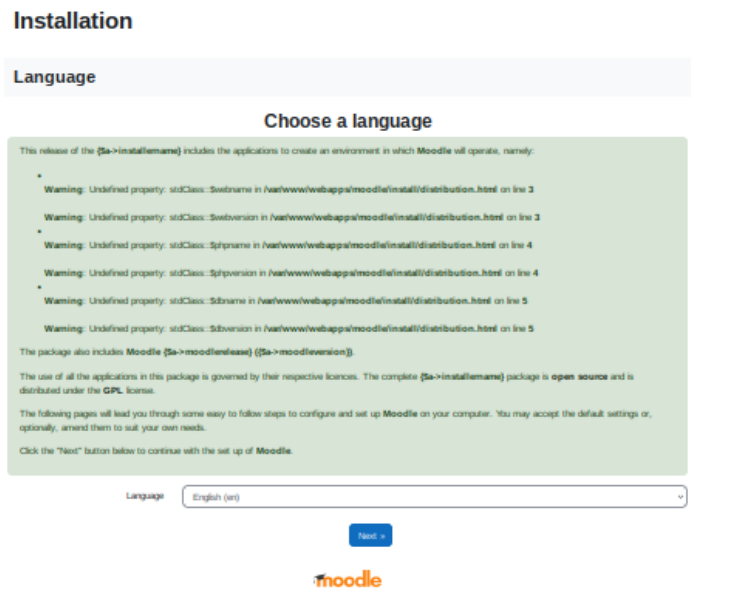
FLUSH PRIVILEGES;

exit

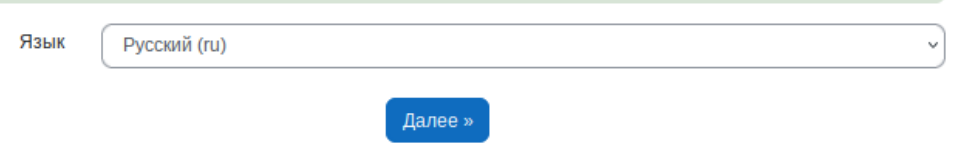
Далее переходим в браузер для продолжения установки

http://172.16.200.1/moodle/install.php

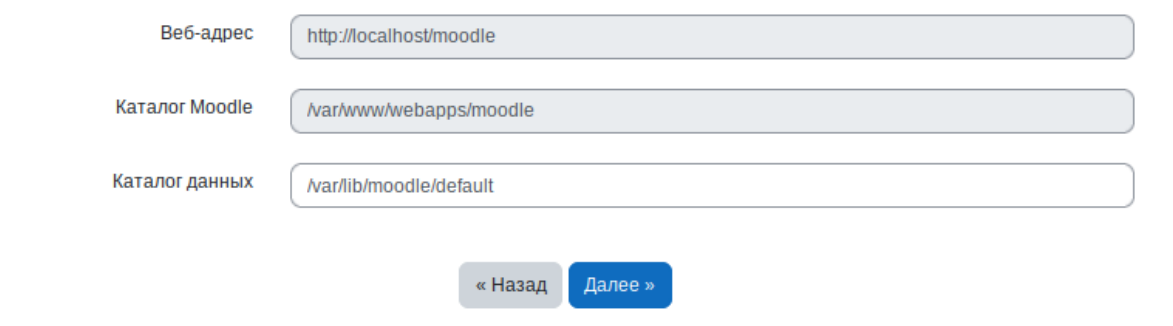
Нас встречает окно веб-установщика



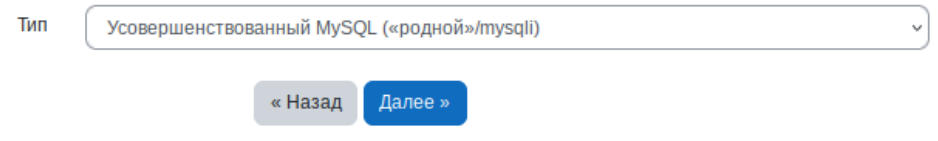
Необходимо выбрать язык установки - для удобства выберем русский и нажмем далее



Оставляем настройки конфигурации по умолчанию и жмём далее



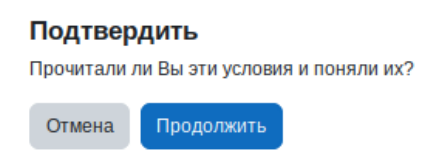
Тип БД выбираем Усовершенствованный MySQL («родной»/mysqli) и жмём далее



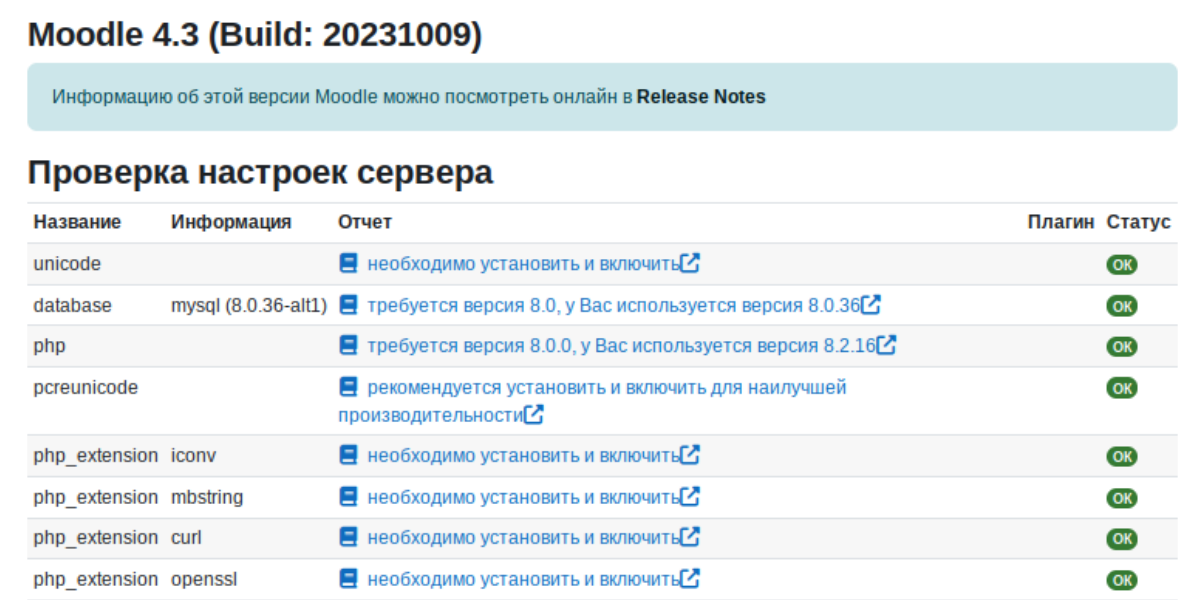
Выставляем следующие настройки и жмём далее



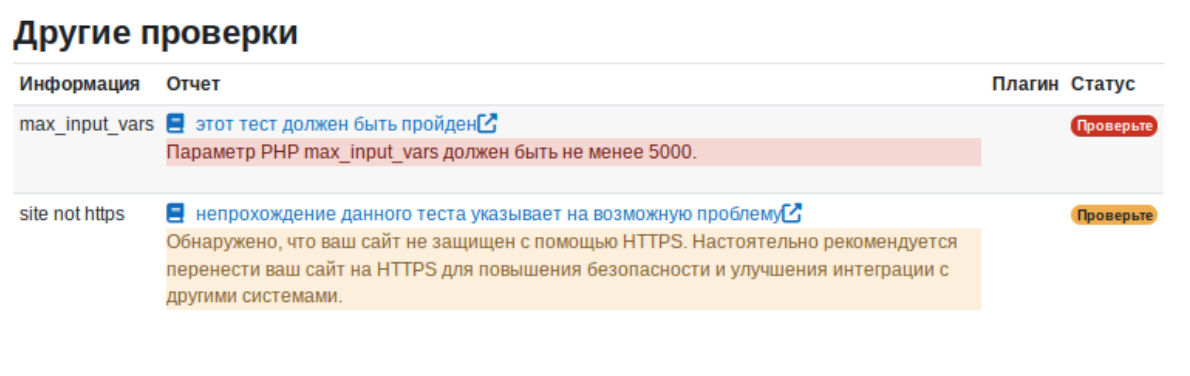
Нажимаем продолжить



Появляется окно проверки настроек сервера



Чтобы завершить установку не должно быть статусов с красным цветом



Для решения ошибки “Параметр PHP max\_input\_vars должен быть не менее 5000” необходимо отредактировать конфигурацию PHP. Для этого:

nano /etc/php/8.2/apache2-mod\_php/php.ini

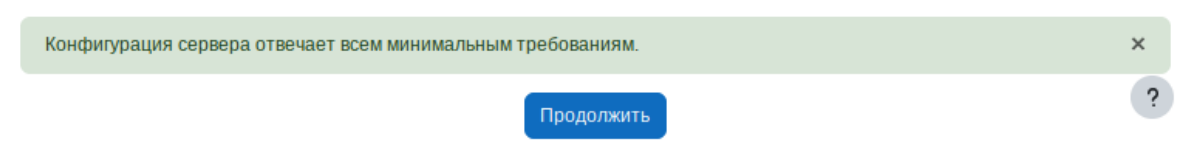
max\_input\_vars = 5000

Ориентировочно находится на 389 строке, определить строчку можно с помощью ctrl + C

systemctl restart httpd2.service

Переходим обратно на сайт и нажимаем reload/обновить

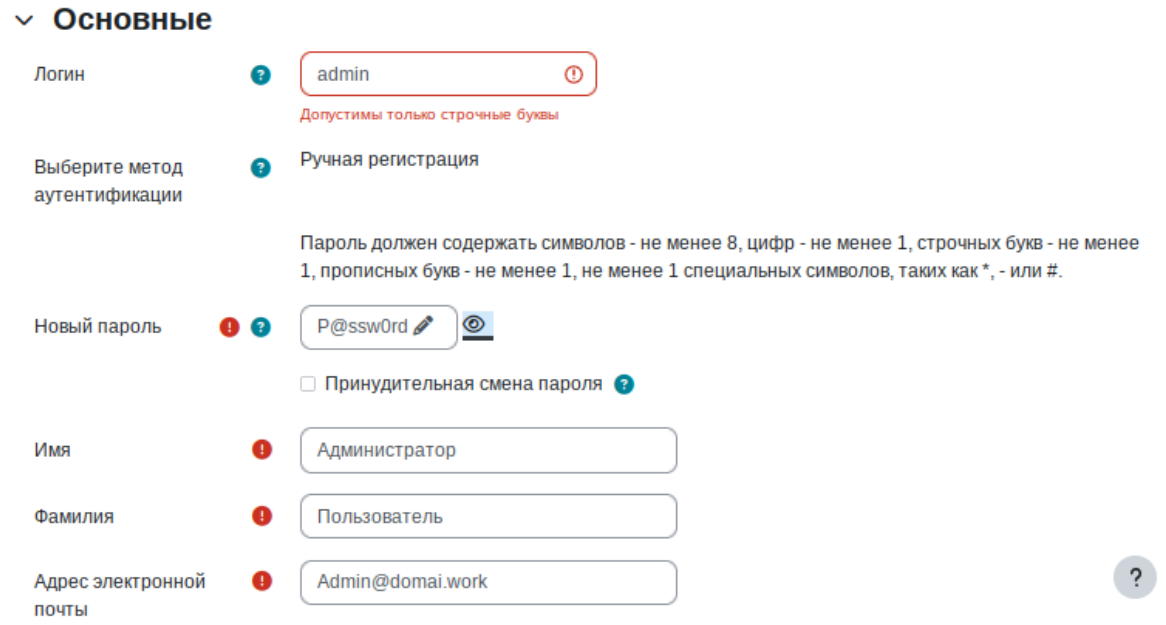
Если все сделано правильно он предложить продолжить



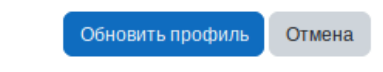
Далее идёт сам процесс установки



Заполняем данные

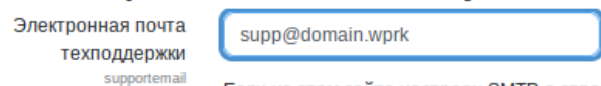


И нажимаем обновить профиль

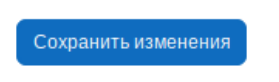


Даем название нашему сайту





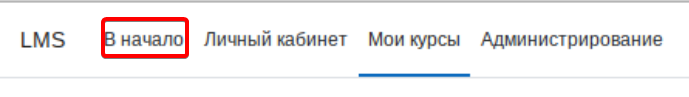
Заканчиваем установку



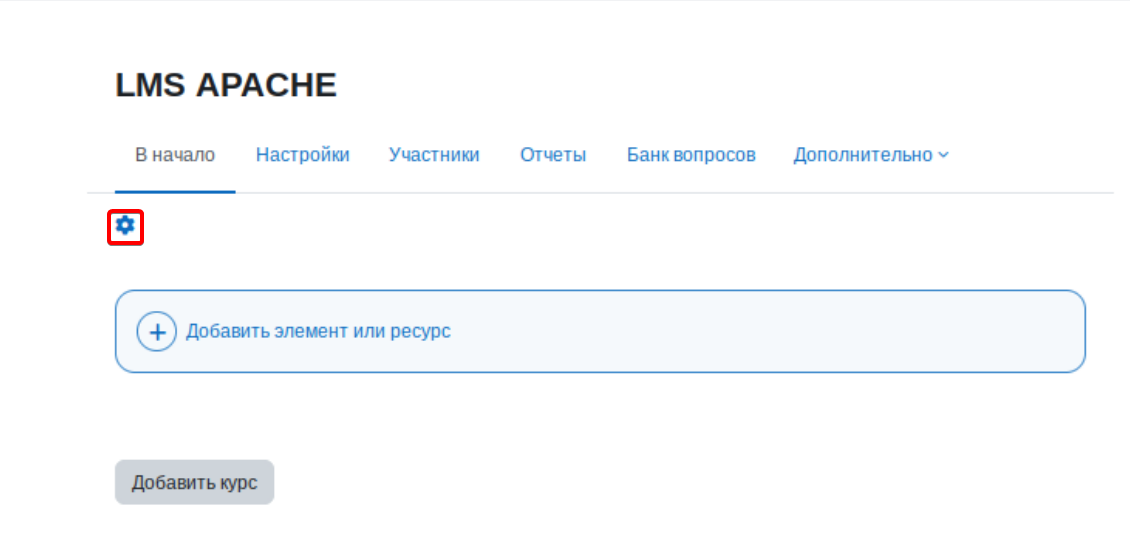
После получения доступа к сайту - переходим в режим редактирования



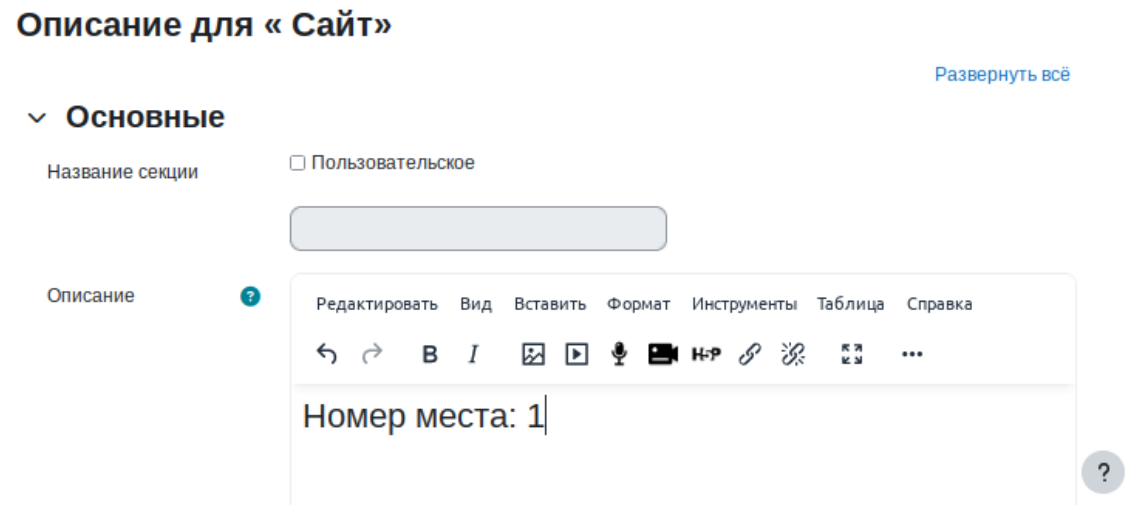
Переходим на домашнюю страницу сайта



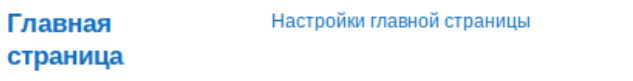
Нажимаем на шестеренку



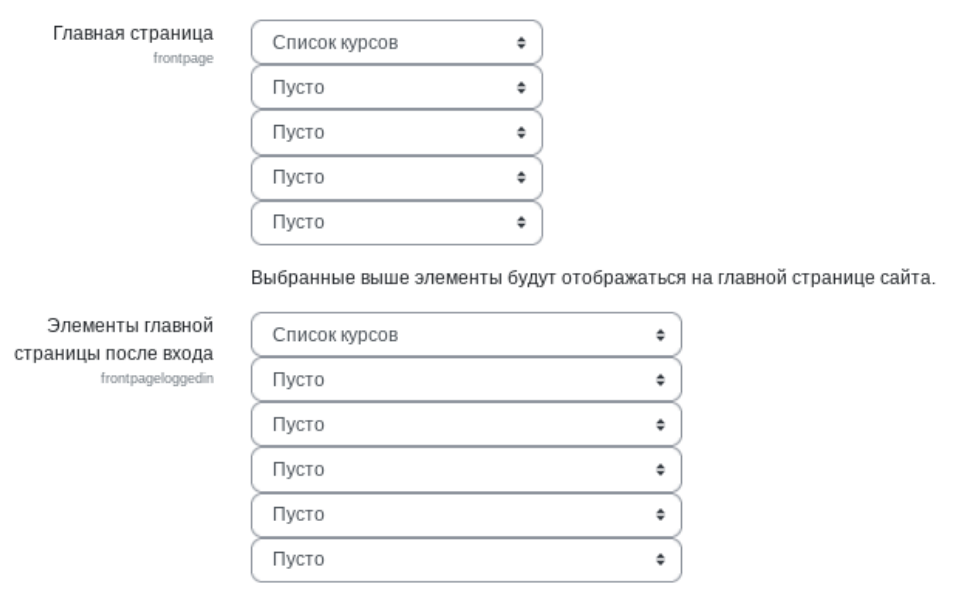
И в разделе описание пишем номер нашего рабочего места (можно выделить текст и в параметрах Формат > Блоки можно сделать текст больше) и сохраняем настройки.

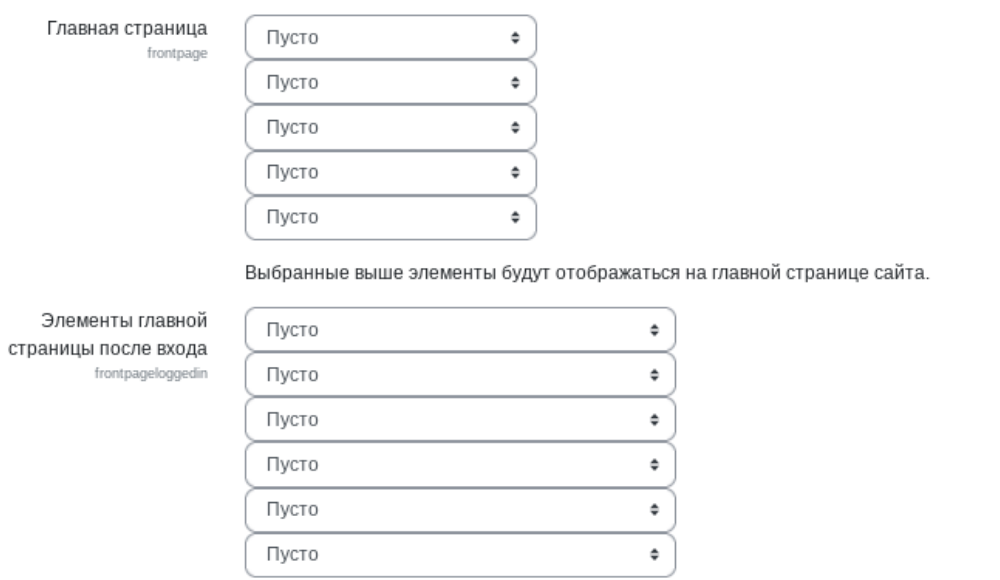


Далее переходим на страницу администрирования и в разделе “Главная страница” открываем “Настройки главной страницы”

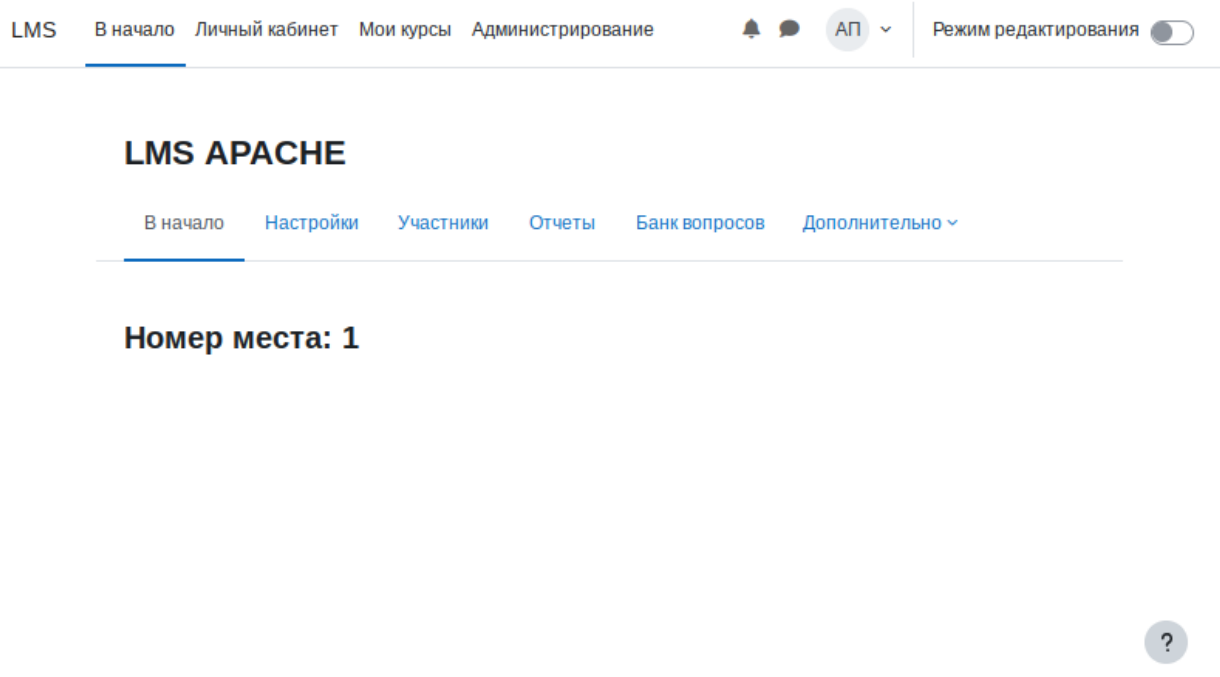


И меняем все элементы главной страницы на **пусто**

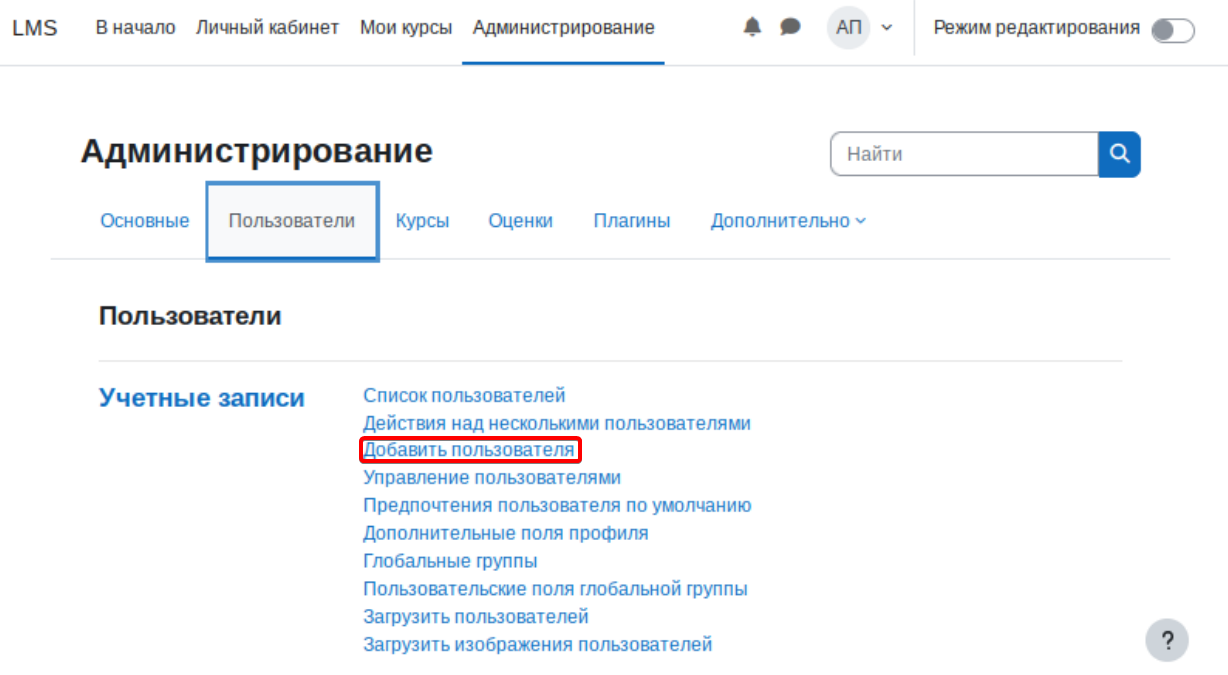




Сохраняем изменения, выключаем режим администрирования и переходим на главную страницу. Должно выглядеть так



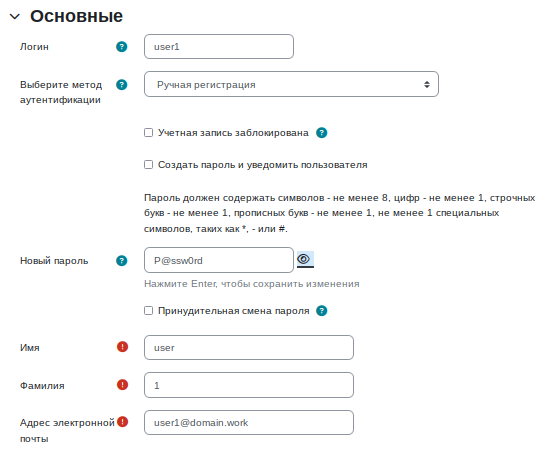
Для создания пользователя переходим по пути “Администрирование” > “Учётные записи” > “Добавить пользователя”



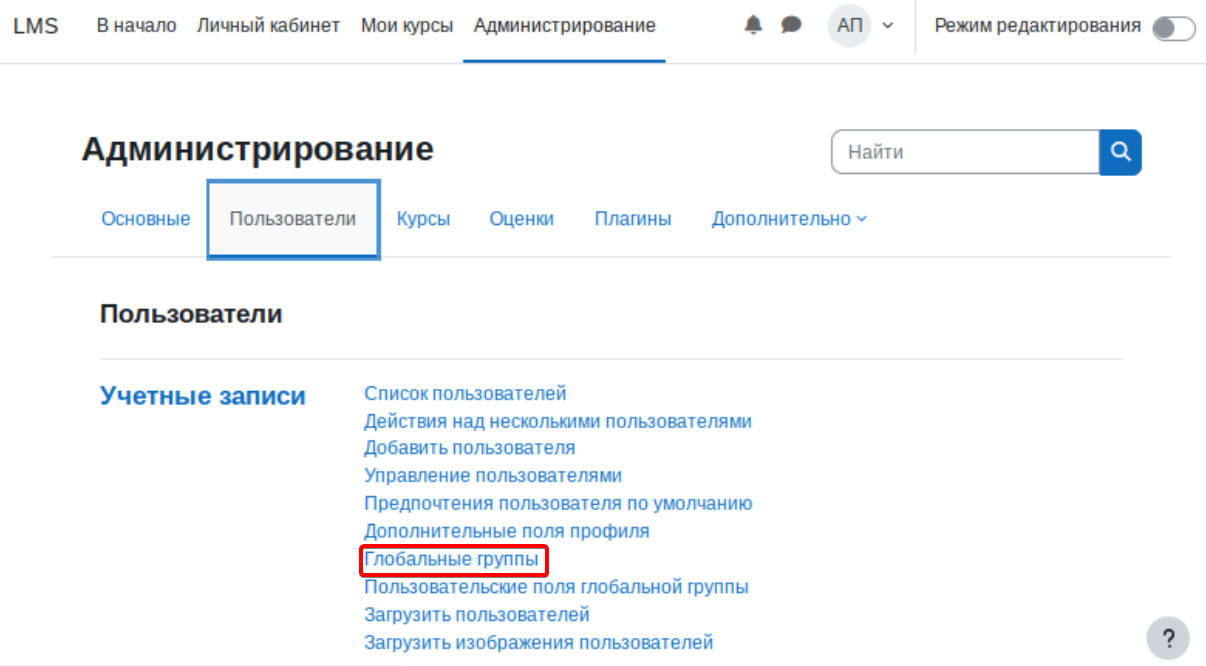
Далее создаем пользователей согласно таблице.

| Пользователь | Группа |
| --- | --- |
| Admin | Admin |
| Manager1 | Manager |
| Manager2 | Manager |
| Manager3 | Manager |
| User1 | WS |
| User2 | WS |
| User3 | WS |
| User4 | WS |
| User5 | TEAM |

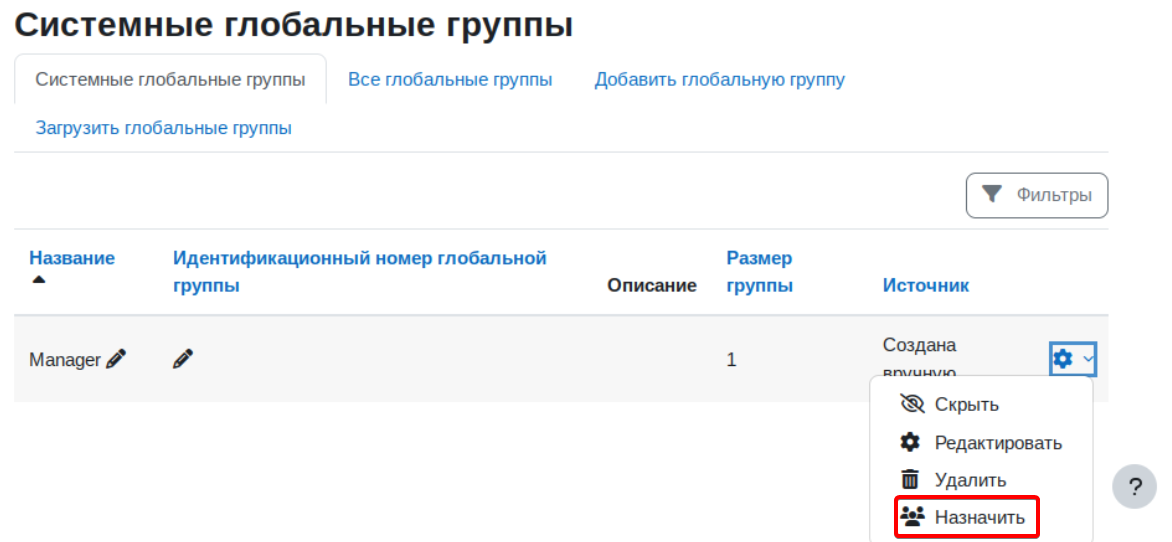
Создаём пользователей по следующему шаблону.



Для создания групп переходим по пути “Администрирование” > “Учётные записи” > “Глобальные группы”



После создания групп, добавляем в них нужных пользователей



# Docker

Пример докер контейнера MediaWiki можно посмотреть на сайте образов

<https://hub.docker.com/_/mediawiki>

На HQ-SRV:

Установим docker и docker-compose

apt-get install docker-engine docker-compose

Включаем сервис и добавляем его в автозагрузку

systemctl enable --now docker.service

Докер контейнер должен быть создан в домашней директории пользователя, перейдем в неё:

cd /home/user

Создадим докер контейнер:

nano wiki.yml

version: '3'

services:

wiki:

image: mediawiki

restart: always

ports:

- 8080:80

volumes:

- images:/var/www/html/images

# - ./LocalSettings.php:/var/www/html/LocalSettings.php

links:

- db

db:

image: mysql

container\_name: db

environment:

MYSQL\_DATABASE: mediawiki

MYSQL\_USER: wiki

MYSQL\_PASSWORD: DEP@ssw0rd

MYSQL\_ROOT\_PASSWORD: toor

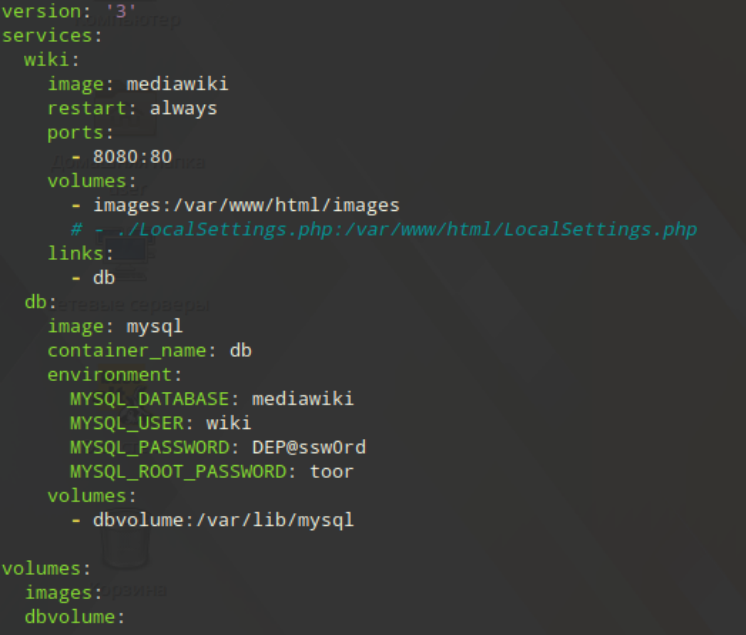
volumes:

- dbvolume:/var/lib/mysql

volumes:

images:

dbvolume:



Запустим докер контейнер:

docker-compose -f wiki.yml up -d

На CLI:

Перейдем по адресу нашего контейнера для продолжения настройки:

http://192.168.200.1:8080



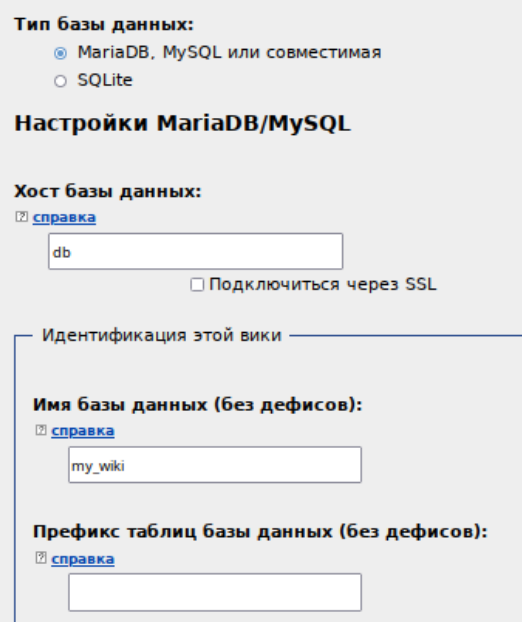
Выберем язык интерфейса.

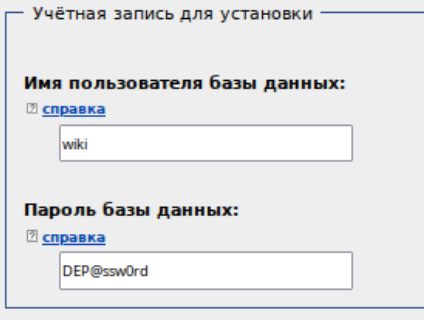


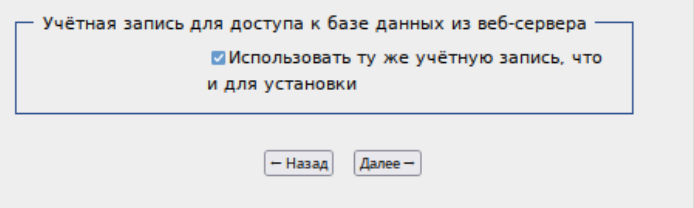
Соглашаемся с условиями и нажимаем далее



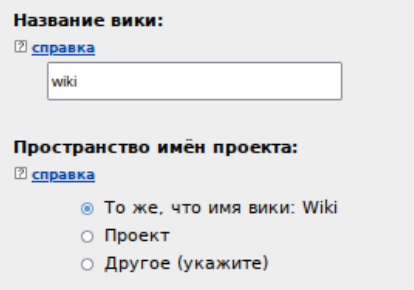
Далее необходимо заполнить данные о нашей БД согласно конфигурации.

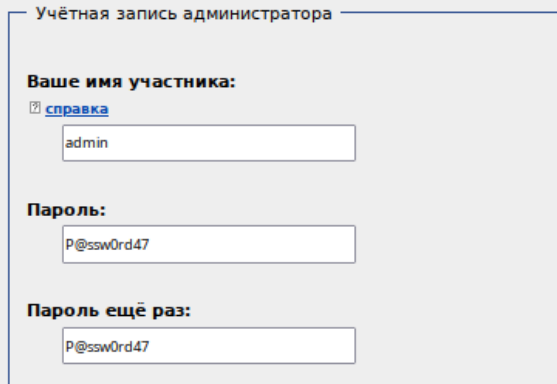


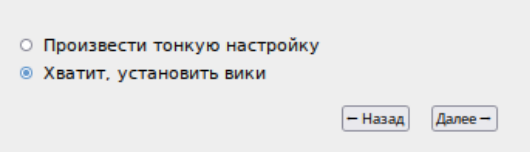


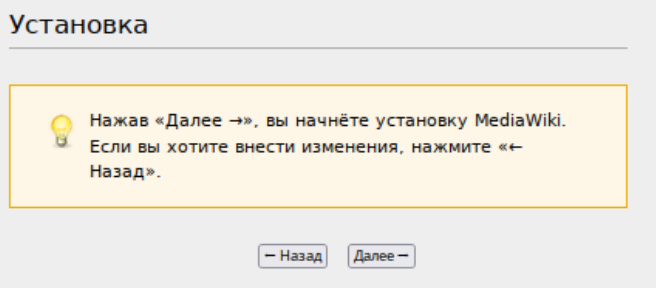


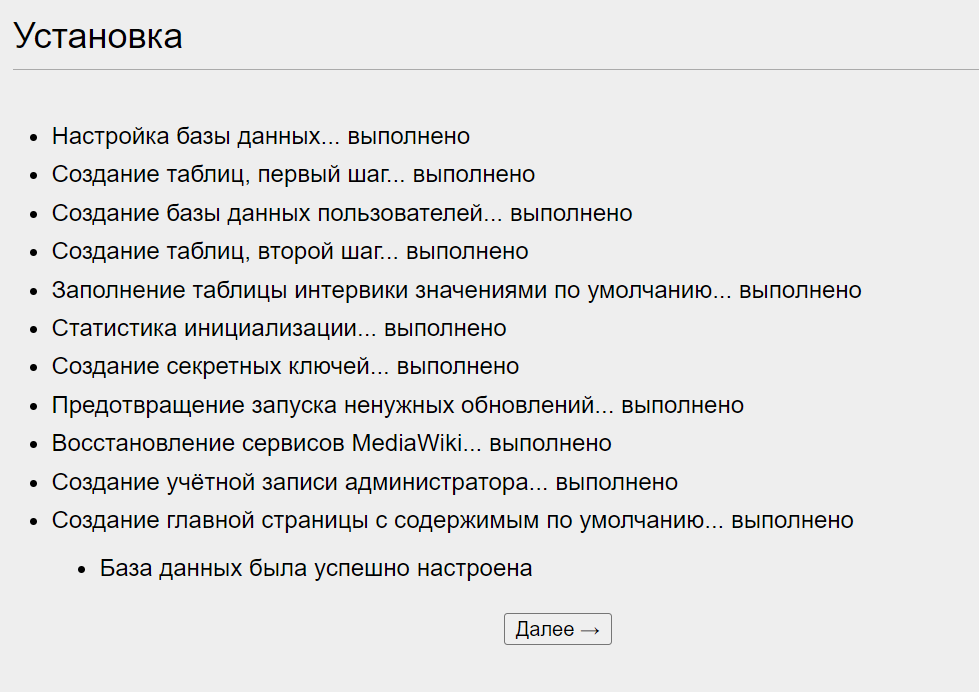
Далее заполним данные о нашем сервисе.

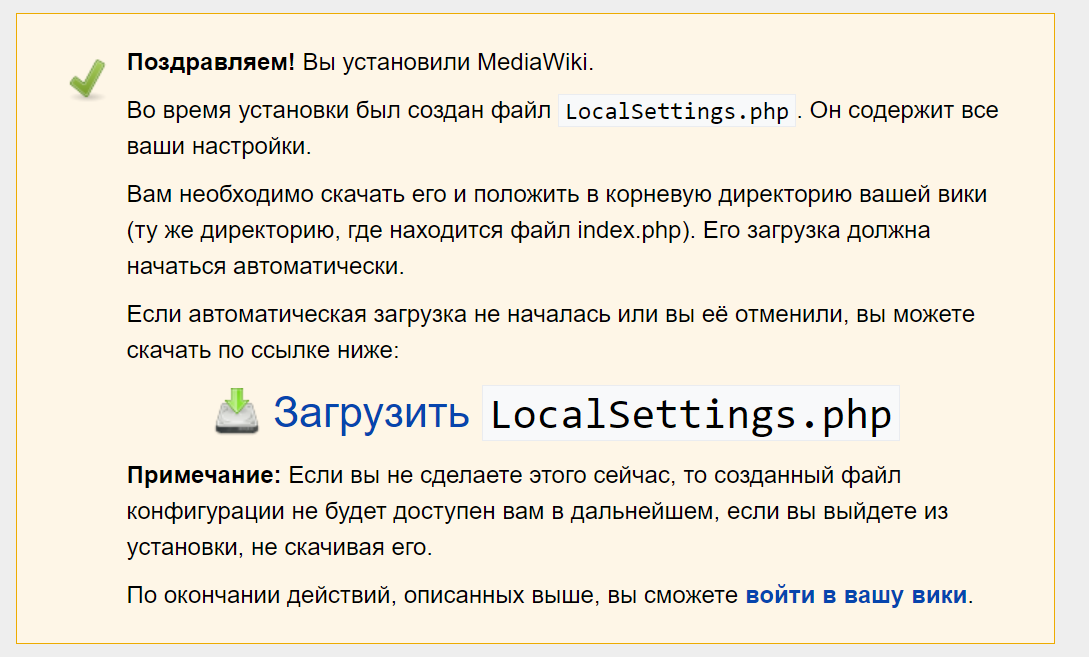












Далее необходимо перенести файл конфигурации с CLI на HQ-SRV.

На HQ-SRV:

scp -P 4444 root@10.0.1.100:/home/user/Загрузки/LocalSettings.php /home/user/

Убедимся в том, что файл был доставлен:

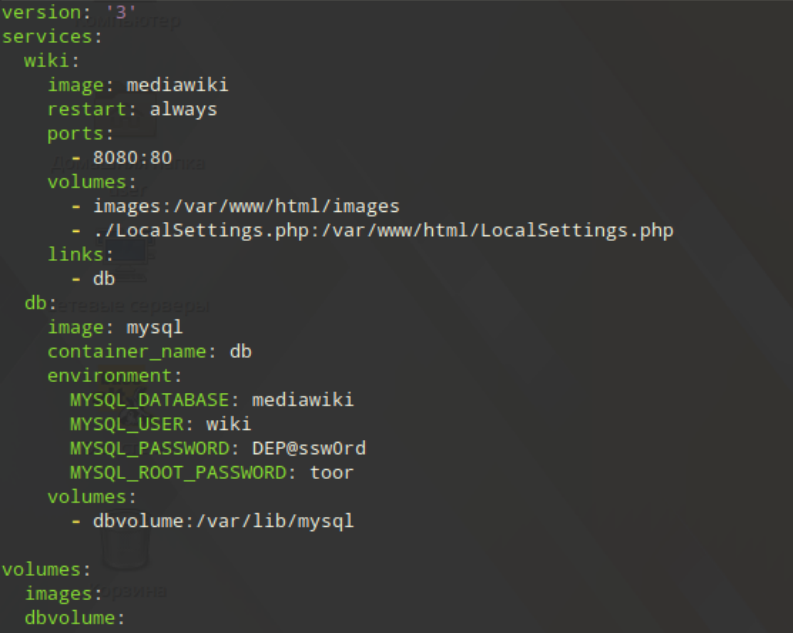
ls /home/user

Далее необходимо остановить наш контейнер:

docker-compose -f wiki.yml down

После необходимо зайти в конфигурацию нашего контейнера и раскомментировать строчку связанную с LocalSettings.php:

nano wiki.yml



Вновь запустим наш контейнер:

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

На CLI:

Вновь перейдем по адресу нашего контейнера и проверим что все работает:

http://192.168.200.1:8080

# 

# **Rsyslog**

Если вы перезагружаете rsyslog на сервере - перезагрузите rsyslog на клиенте, так как иногда пропадает соединение.

На сервере сбора (отлично подходит HQ-SRV)

Установим пакет rsyslog:

apt-get install rsyslog-classic

Включим и добавим его в автозагрузку:

systemctl enable --now rsyslog.service

Введем параллельное ведение журналов в syslog, для этого перейдем в конфигурацию журналирования системы:

nano /etc/systemd/journald.conf

Раскомментируем и заменим значение параметра конфигурации **ForwardToSyslog**:

[Journal]

ForwardToSyslog=yes

Для получения логов с других устройств раскомментируем следующие строки в конфигурационном файле rsyslog

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



Создадим файл конфигурации нашего мониторинга monitor.conf в директории /etc/rsyslog.d/

nano /etc/rsyslog.d/monitoring.conf

if $msg contains "matches resource limit" then /var/log/syslog/monit.log

&~

Также создадим файл конфигурации мониторинга наших устройств-клиентов

nano /etc/rsyslog.d/client.conf

$template remote-incoming-logs,"/var/log/%FROMHOST%.log"

\*.\* ?remote-incoming-logs

Далее необходимо установить пакет monit, он будет служить средством мониторинга системы и формирования предупреждений.

Установим пакет monit:

apt-get install monit

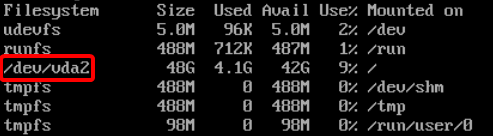
Включим и добавим его в автозагрузку:

systemctl enable --now monit

Сначала необходимо узнать какой путь имеет диск с файловой системой, для этого выведем информацию о используемых дисках:

df -h

Нас интересует диск с наибольшим размером.



Теперь добавим правила формирования предупреждений. Для этого в конец конфигурационного файла monit допишем следующее:

nano /etc/monitrc

check system my\_system

if cpu usage > 70% then alert

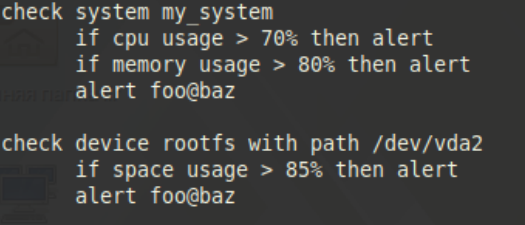
if memory usage > 80% then alert

alert foo@baz

check device rootfs with path <путь диска>

if space usage > 85% then alert

alert foo@baz



Выполним перезагрузку для применения всех системных настроек

reboot

На всех остальных linux-устройствах

Установим необходимые пакеты

apt-get install rsyslog-classic monit

Включим и добавим в автозагрузку

systemctl enable --now rsyslog.service

systemctl enable --now monit

Раскомментируем и заменим значение параметра конфигурации **ForwardToSyslog**:

nano /etc/systemd/journald.conf

[Journal]

ForwardToSyslog=yes

Создадим файл конфигурации rsyslog для отправки сообщений на сервер

nano /etc/rsyslog.d/send.conf

if $msg contains "matches resource limit" then @@192.168.200.1:514

&~

Добавим правила формирования предупреждений

nano /etc/monitrc

check system my\_system

if cpu usage > 70% then alert

if memory usage > 80% then alert

alert foo@baz

check device rootfs with path <путь диска>

if space usage > 85% then alert

alert foo@baz

Перезагружаемся

reboot

**Проверка**   
Установим пакет для стресс-проверки устройства:

apt-get install stress

**Загрузить проц**

stress --cpu 4 --timeout 60s

Это означает, что будет создано 4 потока нагрузки на процессор, которые будут работать в течение 10 секунд.

**Загрузить память**

stress --vm 1 --vm-bytes 3,7G --timeout 120s

Это означает, что будут создан 1 виртуальный поток, который будет использовать 3,7 гигабайта оперативной памяти, и он будут работать в течение 60 секунд.

Если памяти недостаточно можно попробовать Mb для этого вместо G написать M

**Загрузить диск**

touch /etc/disk\_usage

dd if=/dev/zero of=/etc/disk\_usage bs=25M count=1000

Где:

**if=/dev/zero** указывает, что содержимое файла будет состоять из нулей.

**of=/path/to/fillfile** указывает путь и имя файла, который будет создан.

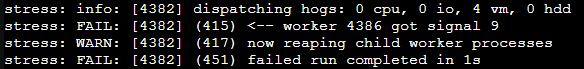
**bs=25M** задает размер блока данных для записи (в данном случае 25 МБ).

**count=1000** определяет количество блоков, которые будут записаны (в данном случае 1000 блоков, что соответствует 25 ГБ).

По завершению проверки - не забудьте удалить файл:

rm /etc/disk\_usage

Если возникает похожая ошибка - поиграйтесь с параметрами команды



На сервере сбора:

В файле логов устройств должны появиться все записи:

cat /var/log/syslog/monit.log

# **Bacula**

На HQ-SRV

apt-get install bacula13-common bacula13-console bacula13-director-common bacula13-director-mysql bacula13-storage mt-st

**Если не установлен MySQL**

apt-get install MySQL-server

**Запустим сервер MySQL и добавим его в автозагрузку**

systemctl enable --now mysqld

В bacula присутствуют скрипты для создания базы, пользователя и таблиц.

Для того чтобы задать пароль для пользователя bacula необходимо отредактировать следующую строку в файле скрипта:

nano /usr/share/bacula/scripts/grant\_mysql\_privileges

db\_password="bacula"

Выполним следующие скрипты используя пароль **root**:

cd /usr/share/bacula/scripts

./create\_mysql\_database

./make\_mysql\_tables

./grant\_mysql\_privileges

**Настройка Director**

За настройку Bacula Director отвечает файл **/etc/bacula/bacula-dir.conf**:

Нас интересует секция **Catalog**, его нужно привести к виду:

Catalog {

Name = MyCatalog # Имя каталога

dbname = bacula # Имя базы данных

user = bacula # Имя пользователя базы данных

password = "bacula" # Пароль пользователя базы данных

}

**Описание хранилищ**

В каталоге **storage.d** находятся конфигурационные файлы хранилищ к которым director имеет доступ. Нам нужно привести файл **file.conf** к следующему виду:

nano /etc/bacula/storage.d/file.conf

Storage {

Name = File

Address = 192.168.200.1

SDPort = 9103 # Порт

@/etc/bacula/bacula-sd-password.conf

Device = FileStorage

Media Type = File

}

**Описание наборов файлов**

В каталоге **fileset.d** находятся описания списков файлов для резервирования. Нам необходимо создать собственный список, для этого:

**Копируем шаблон описания**

cp /etc/bacula/fileset.d/catalog.conf /etc/bacula/fileset.d/etcback.conf

chown root:bacula /etc/bacula/fileset.d/etcback.conf

chmod 660 /etc/bacula/fileset.d/etcback.conf

**Отредактируем его**

nano /etc/bacula/fileset.d/etcback.conf

**Приводим его к виду**

FileSet {

Name = "etcback"

Include {

Options {

signature = MD5

}

File = /etc

}

}

**Описание задач**

В каталоге job.d описываются задания резервирования. Необходимо создать собственную задачу, для этого:

**Копируем шаблон задачи**

cp /etc/bacula/job.d/defaultjob.conf /etc/bacula/job.d/backserv.conf

chown root:bacula /etc/bacula/job.d/backserv.conf

chmod 660 /etc/bacula/job.d/backserv.conf

**Отредактируем его**

nano /etc/bacula/job.d/backserv.conf

**Приводим его к виду**

JobDefs {

Name = "BackupServ"

Type = Backup

Level = Incremental

Client = fd2

FileSet = "etcback"

Storage = File

Messages = Standard

Pool = Default

Priority = 10

}

Job {

Name = "BR-SRV-BACKUP"

JobDefs = "BackupServ"

}

**Описание клиентов**

В каталоге client.d описываются клиенты. Необходимо создать клиента, данные которого мы будем бэкапить (в нашем случае это BR-SRV). Для этого:

**Копируем шаблон клиента**

cp /etc/bacula/client.d/client1.conf /etc/bacula/client.d/client2.conf

chown root:bacula /etc/bacula/client.d/client2.conf

chmod 660 /etc/bacula/client.d/client2.conf

**Отредактируем его**

nano /etc/bacula/client.d/client2.conf

**Приводим его к виду**

Client {

Name = fd2

Address = 172.16.200.1

FDPort = 9102

Catalog = MyCatalog

Password = "fd2"

File Retention = 30 days

Job Retention = 6 months

AutoPrune = yes

}

**Настройка Storage**

Основной файл конфигурации хранилищ находится это **/etc/bacula/bacula-sd.conf**, также к нему присоединяются все файлы из каталога **device.d**.

Необходимо привести файл **filestorage.conf** к следующему виду

nano /etc/bacula/device.d/filestorage.conf

Device {

Name = FileStorage

Media Type = File

Archive Device = /srv/backup

LabelMedia = yes;

Random Access = Yes;

AutomaticMount = yes;

RemovableMedia = no;

AlwaysOpen = no;

}

Каталог **/srv/backup** должен существовать и принадлежать пользователю **bacula**, для этого:

mkdir /srv/backup

chown bacula:bacula /srv/backup

**Проверка конфигурационных файлов**

Для работы bacula-director и прослушивания порта 9101 необходимо исправить права доступа

chown bacula:bacula /var/lib/bacula

Проверить корректность конфигурационных файлов **bacula director**

bacula-dir -t -c /etc/bacula/bacula-dir.conf

Проверить корректность конфигурационных файлов **bacula storage**

bacula-sd -t -c /etc/bacula/bacula-sd.conf

**Автозапуск сервисов**

systemctl enable --now bacula-dir

systemctl enable --now bacula-sd

На BR-SRV

apt-get install bacula13-fd

**Настройка File Daemon**

Конфигурационный файл клиента **/bacula-fd.conf**.

Необходимо привести секции **Director** и **FileDaemon** к следующему виду:

nano /etc/bacula/bacula-fd.conf

Director {

Name = dir

Password = "fd2"

}

FileDaemon {

Name = fd2

FDport = 9102

WorkingDirectory = /var/lib/bacula

Pid Directory = /var/run/bacula

Maximum Concurrent Jobs = 20

}

**Автозапуск сервиса**

systemctl enable --now bacula-fd

На HQ-SRV:

bconsole

run

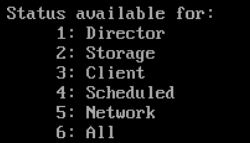
выбираем цифру под которой стоит задача **BR-SRV-BACKUP**

yes

**Проверка работоспособности**

status

Далее будет предоставлен выбор

****

Нас интересуют 2 пункта

**1: Director - позволяет посмотреть ход выполнения задач**

**5: Network - позволяет проверить есть ли связь до нужного клиента**

Если все выполнено правильно - в статусе директора появится похожее сообщение



Выйти можно с помощью команды:

exit

А в папке куда сохраняются бэкапы должен появиться сам файл

# 

# **CA**

На всех машинах, кроме HQ-SRV:

Создадим папку для наших сертификатов

mkdir /etc/certs

И папку для закрытых ключей сертификатов

mkdir /etc/certs/keys

На HQ-SRV:

Нам понадобится установить easy-rsa:

apt-get install easy-rsa

Создадим директорию, в котором будут храниться настройки Easy-RSA

mkdir /etc/easy-rsa

Скопируем файлы настроек Easy-RSA в созданный каталог

cp -r /usr/share/easyrsa3/\* /etc/easy-rsa/

Перейдем в созданную директорию

cd /etc/easy-rsa

Нужно отредактировать файл с переменными для нашего центра сертификации. Все доступные для редактирования параметры представлены в файле vars.example

Копируем его в файл с названием *vars*, который будет использовать утилита easy-rsa при генерации параметров

cp vars.example vars

Редактируем нужные параметры в соответствии с заданием:

nano vars

Меняем процедуру генерации сертификата, чтобы в параметры сертификата были включены дополнительные атрибуты, заполним их для наполняемости

set\_var EASYRSA\_DN="org"

set\_var KEY\_COUNTRY="RU"

set\_var KEY\_PROVINCE="Ryazan"

set\_var KEY\_CITY="Ryazan"

set\_var KEY\_ORG="RCE"

set\_var KEY\_EMAIL="admin@domain.work"

set\_var KEY\_OU="Servers"

Инициализируйте PKI инфраструктуру, при этом автоматически создастся папка pki/ в которой будут хранится открытые и закрытые ключи сертификатов

easyrsa init-pki

Генерируем сертификат центра сертификации, без использования пароля

easyrsa build-ca nopass

Common Name : HQ-SRV

**~~Выдача сертификата для SSH~~**

~~easyrsa gen-req server nopass~~

~~easyrsa sign-req server server~~

**~~Передача сертификатов~~**

~~Сертификат и закрытый ключи для SSH необходимо передать на все устройства~~

~~Для ssh~~

~~scp pki/issued/ssh\_host\_key.crt root@<имя или ip>:/etc/certs~~

~~scp pki/private/ssh\_host\_key.key root@<имя или ip>:/etc/certs/keys~~

~~Сертификаты и закрытый ключ для веб-серверов должен находиться на HQ-SRV и BR-SRV~~

~~scp pki/issued/domain.work.crt root@<имя или ip>:/etc/certs~~

~~scp pki/private/domain.work.key root@<имя или ip>:/etc/certs/keys~~

**~~Установка сертификатов для веб-служб~~**

~~На BR-SRV:~~

~~Для установки корневого сертификата на клиенте необходимо установить права доступа и импортировать в браузер настройки-безопасность-просмотр сертификатов-корневые центры-импортировать~~

# 

# **FireWall**

На BR-R:

iptables -P INPUT ACCEPT

iptables -P FORWARD ACCEPT

iptables -P OUTPUT ACCEPT

iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 53 -j ACCEPT

iptables -A INPUT -i eth0 -p udp --dport 53 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -i eth0 -p udp --dport 1234 -j ACCEPT

iptables -A INPUT -i eth0 -p icmp -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 4444 -j ACCEPT

iptables-save > /etc/sysconfig/iptables

systemctl enable --now iptables