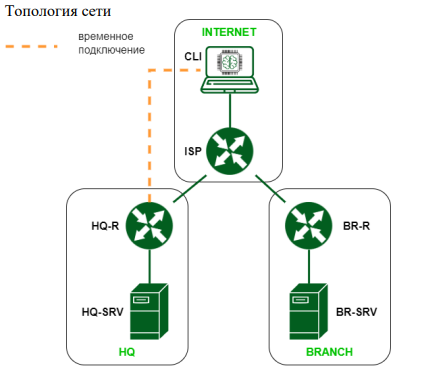
**Обновлено 09.04.2024 V1.3**

****

**Преднастройка**

Если в задании не будут использоваться встроенные репозитории, а будет возможность скачивать все пакеты из интернета, необходимо отключить проверку пакетов через cdrom зайдя по пути

**Nano /etc/apt/sources.list**

и закомментировать находящуюся там строку.

**Задание 1 модуля 1**

**1. Выполните базовую настройку всех устройств:**

**A. Присвоить имена в соответствии с топологией**

***Примечание:*** для выполнения данного задания необходимо постоянное изменение имени каждого устройства, указанного на топологии (временно изменение, действует только до перезагрузки системы и не является верным выполнением задания)

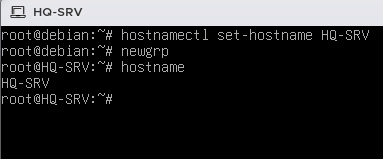
***Решение:***

Для фиксированного изменения имени компьютера, необходимо использовать команду:

**hostnamectl set-hostname Имя устройства**

Для изменения имени компьютера в текущем сеансе без перезагрузки можно воспользоваться командой:

**newgrp**

Рисунок 1 — Пример изменения имени устройства

B. Рассчитайте IP-адресацию IPv4 и IPv6. Необходимо заполнить таблицу №1, чтобы эксперты могли проверить ваше рабочее место.

C. Пул адресов для сети офиса BRANCH - не более 16

D. Пул адресов для сети офиса HQ - не более 64

***Примечание:*** Для сетей офисов HQ (входят устройства HQ-R и HQ-SRV) и офисов BRANCH (входят устройства BR-R и BR-SRV) , необходимо рассчитать IPv4 и IPv6 адреса согласно пунктам C и D, для устройств CLI и ISP, можно выбирать адреса из пула серых адресов с стандартной маской /24 255.255.255.0 (при условии, что IP адреса этих устройств не будут заданы заранее или не будут указаны другие условия в задании)

***Решение:***

***Для расчёта IPv4 адресов можно воспользоваться стандартной таблицей масок от 24 до 32 (при условии, если количество адресов в сети 256 или меньше и изменяется только последний октет в адресе)***

*255.255.255.0 /24 маска — 1 сеть в которой 256 адресов от 0 до 255*

*255.255.255.128 /25 маска — 2 сети в каждой из которых по 128 адресов от 0 до 127 и от 128 до 256*

*255.255.255.192 /26 маска — 4 сети в каждой из которых по 64 адреса*

*255.255.255.224 /27 маска — 8 сетей по 32 адреса*

*255.255.255.240 /28 маска — 16 сетей по 16 адресов*

*255.255.255. 248 /29 маска — 32 сети по 8 адресов*

*255.255.255.252 /30 маска — 64 сети по 4 адреса*

*255.255.255.254 /31 маска — 128 сетей по 2 адреса*

*255.255.255.255 /32 маска — 256 сетей по 1 адресу*

*Исходя из таблицы мы понимаем, что в офисе HQ используется 26 маска, а в офисе Branch используется 28 маска*

*При изменение 3-го октета в адресе используются маски от 16 до 23,*

*при изменении 2-го октета в адерсе используются маски от 8 до 15*

*при изменении 1-го октета в адресе используются маски от 1 до 7*

**В сетях IPV6 размер маски составляет 128 бит**

**Однако последние 32 бита маски (от 96 до 128) полностью идентичны маскам в IPv4, однако структура адреса отличается, так как IPv6 работает в шестнадцатеричной системе счисления размер одного октета равен 16 битам, и полный адрес имеет вид xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx , где X значения от 0 до F. Следовательно аналогом диапазона от 24 до 32 в ipv4 , будут маски от 120 до 128 в IPv6.**

**так**

***/120 маска — 1 сеть в которой 256 адресов находящиеся в диапазоне от*** *xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx00 до xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxff*

***/121 маска — 2 сети в каждой из которых по 128 адресов первая сеть в диапазоне***

*от*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx00*

*до*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx7f*

***вторая сеть в диапазоне***

*от*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx80*

*до*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxff*

**Объяснение:**

**Так как размер одного октета в IPv6 равняется 16 битам , в то время как в IPv4 оно равно 8 битам , изменяются лишь два последних числа в октете**

**xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx , а размер будет равен 256 битам**

**, в случае если у нас будут 2 сети по 128 адресов, то есть в каждую сеть вместить по 128 значений (заполнение начинается всегда справа)**

**при заполнении правого значения на максимум xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx0F ,**

**левое значение увеличивается на один xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx10**

**и происходит снова заполнение правого значения до xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx1F и т.д.**

**пока не будут вмещены все 128 битов (от 0 до 127), 128(число битов) делим на 16 (от 0 до F) и отнимаем 1 (Так как нужно учитывать 0)**

**= 128/16-1=80(восемь и ноль) -1 = 7F (семь ЭФ) и получаем последний адрес для первой xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx7F**

**следовательно, следующая сеть начинается с**

**xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx80**

**и заканчивается на**

**xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxFF**

*/122 маска — 4 сети в каждой из которых по 64 адресов первая сеть в диапазоне*

*от*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx00*

*до*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx3f*

*Вторая сеть в диапазоне*

*от*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx40*

*до*

*xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xx7f*

*и т.д.*

*/123 маска — 8 сетей по 32 адреса*

*/124 маска — 16 сетей по 16 адресов*

*/125 маска — 32 сети по 8 адресов*

*/126 маска — 64 сети по 4 адреса*

*/127 маска — 128 сетей по 2 адреса*

*/128 маска — 256 сетей по 1 адресу*

**Для сетевого взаимодействия можно использовать первый октет 2001**

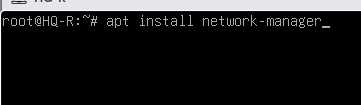
***Так же необходимо подобрать IP адреса (IPv4 и IPv6) которые будут устанавливаться на интерфейсах между маршрутизаторами (если иного не указано в задании, или если они не выданы заранее****)*

|  |  |
| --- | --- |
| Имя устройства | IP |
| CLI | 192.168.0.2 255.255.255.0 — к ISP  2001::3:2/120 — к ISP |
| ISP | 192.168.0.1 255.255.255.0 — к CLI  2001::3:1/120 — к CLI  10.10.10.2 255.255.255.252 — к HQ-R  10.10.10.6 255.255.255.252 — к BR-R  2001::7:2/126 — к HQ-R  2001::7:6/126 — к BR-R |
| HQ-R | 192.168.1.1 255.255.255.240 — к HQ-SRV  2001::1:1/122 — к HQ-SRV  10.10.10.1 255.255.255.252 — к ISP  2001::7:1/126 к — ISP |
| HQ-SRV | 192.168.1.2 255.255.255.240 — к HQ-R  2001::1:2/122 — к HQ-R |
| BR-R | 192.168.2.1 255.255.255.248 — к BR-SRV  2001::2:1/124 — к BR-SRV  10.10.10.5 255.255.255.252 — к ISP  2001::7:5/126 — к ISP |
| BR-SRV | 192.168.2.2 255.255.255.248 — к BR-R  2001::2:2/124 — к BR-R |

Следующим шагом необходимо установить выбранные IP адреса на соответствующие машины, для этого существуют 2 способа.

**Первый способ: через network-manager**

Если network manager не установлен, его можно установить командой

Рисунок 2 — Установка NMTUI

Для того что бы зайти в Network-manager можно воспользоваться командой:

**nmtui**

В nmtui пройдя по пути **Edit a connection — имя интерфейса**

Необходимо настроить ip адреса в соответствии с таблицей адресации

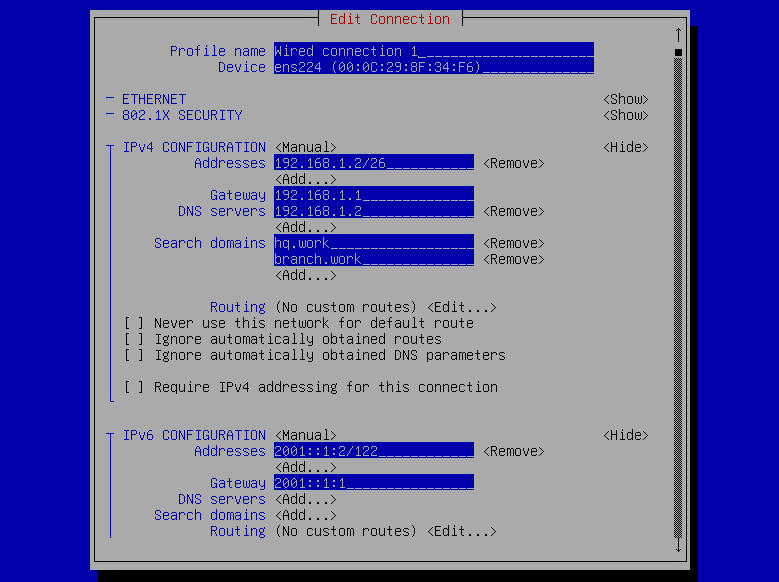
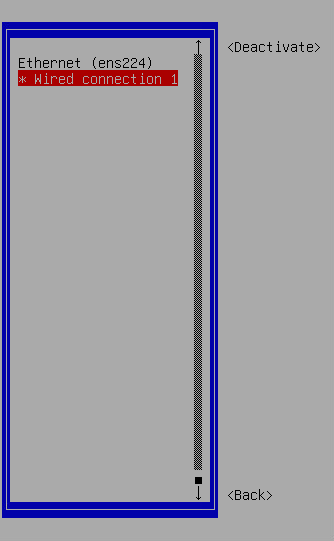


Рисунок 3 — Пример настройки IPv4 и IPv6 на HQ-SRV

После настройки необходимо зайти в activate a connection и перезагрузить все интерфейсы (нажать deactivate и activate на каждом интерфейсе)

Рисунок 4 — перезагрузка интерфейсов

**Примечание:** на интерфейсах, находящихся между маршрутизаторами, не нужно указывать dns, достаточно это сделать на внутренних локальных интерфейсах маршрутизаторов.

**Второй способ: через редактирования конфига интерфейсов**

Вариант ручной настройки без использования любых программ (в случае если не будет возможности установки nmtui или она будет запрещена). Перед установкой интерфейсов необходимо воспользоваться командой IP A для определения имён 7интерфейсов, находим незаполненный интерфейс, в примере ниже незаполненным интерфейсов является ens256

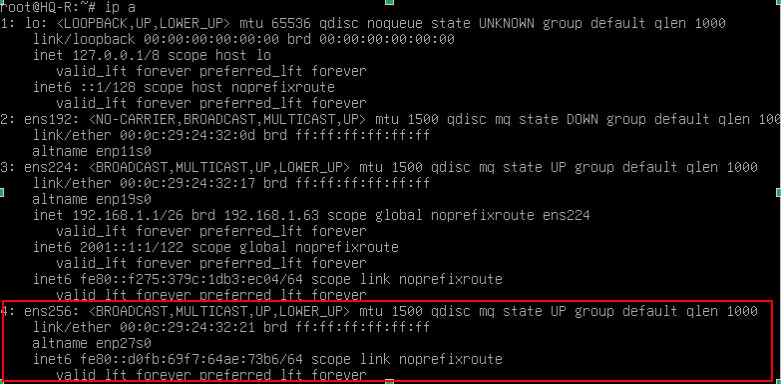


Рисунок 5— Поиск имён интерфейсов для настройки

Определив интерфейс, необходимо воспользоваться командой для просмотра и изменения конфигураций интерфейсов

**nano /etc/network/interfaces**

***или***

**vi /etc/network/interfaces**

И затем сконфигурировать настройки интерфейсов в соответствии с таблицей адресации по примеру, представленному на скриншоте ниже

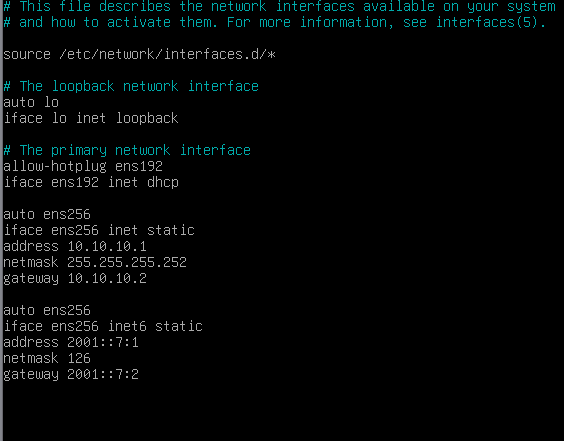


Рисунок 6 — Пример настройки интерфейса HQ-R по IPv4 и IPv6 между ISP и HQ-R

Где:

**auto [имя интерфейса]** – команда для подключения к заданной сетевой карте при запуске операционной системы.

**iface [имя интерфейса] inet static** – указание будет ли статичным или динамичным IPv4 адрес адаптера.

**iface [имя интерфейса] inet6 static** – указание будет ли статичным или динамичным IPv6 адрес адаптера.

**address [адрес]** – порт ethernet.

**netmask [адрес]** – маска подсети.

**gateway [адрес]** – шлюз по умолчанию

Так же есть дополнительные настройки:

**dns-nameservers [адрес]** — указание dns адреса

**dns-search [имя]** — указание имени dns (например hq.work)

**Похожие настройки необходимо проделать на всех машинах сети (Если иного не указано в задании)**

**2. Настройте внутреннюю динамическую маршрутизацию по средствам FRR. Выберите и обоснуйте выбор протокола динамической маршрутизации из расчёта, что в дальнейшем сеть будет масштабироваться.**

a. Составьте топологию сети L3.

**Примечание:** Для данного задания необходимо самостоятельно выбрать протокол динамической маршрутизации, исходя из всех поддерживаемых протоколом FRR (OSPF , EIGRP , IS-IS , BGP и т.д.), OSPF подходит для построения средних по размеру сетей, и при этом является открытым стандартом протоколов динамической маршрутизации , в то время как EIGRP проприетарный протокол CISCO IOS, IS-IS и BGP используются для глобальной маршрутизации на уровне провайдеров.

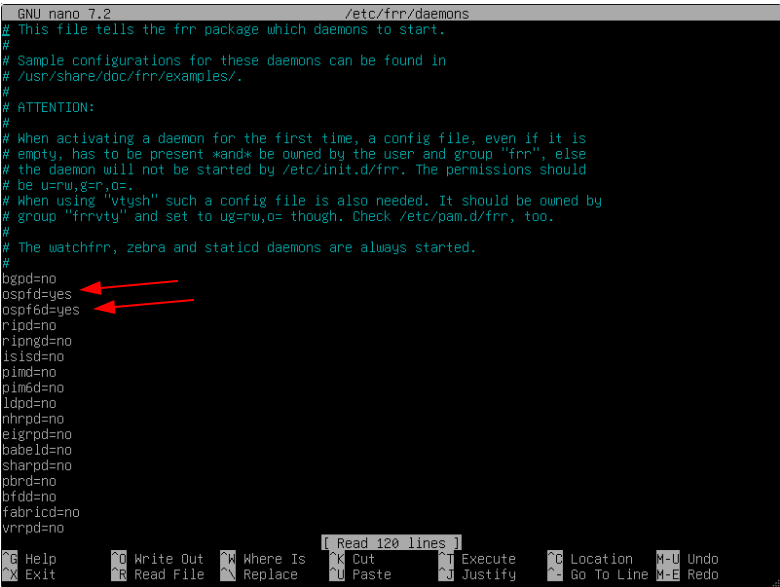
**Решение:** Первым делом необходимо установить пакеты FRR, для этого необходимо воспользоваться командой:

**apt install frr**

Следующим шагом необходимо произвести изменения конфигурационных файлов

**nano /etc/frr/daemons**

и изменить параметры на YES для протокола OSPF

Рисунок 7 — настройка конфигурации FRR

После сохранения конфига, следующим шагом необходимо, перезапустить frr.service командой

**systemctl restart frr**

Далее, после перезагрузки, посредством команды **vtysh** перейти в режим конфигурирования (Настройки идентичны Cisco IOS).

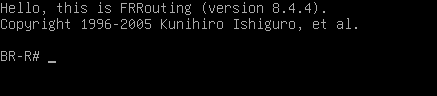


Рисунок 8 — пример конфигурационного окна

Посредством команд:

**Conf t**

**router ospf**

перейти к конфигурированию протокола ospf

Настройка производится посредством объявления

ospf router-id x.x.x.x

и прилегающих к маршрутизатору сетей

network x.x.x.x/x area x

как показано на рисунке 9

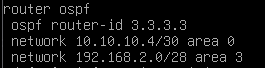


Рисунок 9 — пример настройки OSPF на BR-R

Где network 10.10.10.4/30 area 0 — относится к зоне между ISP и BR-R

а network 192.168.2.0/28 area 3 — относится к зоне между BR-R и BR-SRV

**Примечание:** area 0 — является транзитной зоной между маршрутизаторами, а area 1,2,3,4,5 персональными зонами для локальных сетей, для каждой локальной сети отдельная зона

*Похожие настройки, выполняется на всех остальных маршрутизаторах.*

**После завершения конфигурации в frr, необходимо записать конфигурацию в память устройства, командой write, иначе при перезагрузке frr или устройства, все настройки вернутся к дефолтным**

Параллельно на маршрутизаторах участвующих в передаче межсетевого трафика, необходимо настроить маршрутизацию по протоколу IPv6

Для этого необходимо

Для завершения настройки сети необходимо сконфигурировать настройку для передачи пакетов между сетями в файле **nano /etc/sysctl.conf**

переменную **net.ipv4.ip\_forward=1 необходимо раскоментить** и сохранить изменения в файле, и применить изменения командой **sysctl -p**

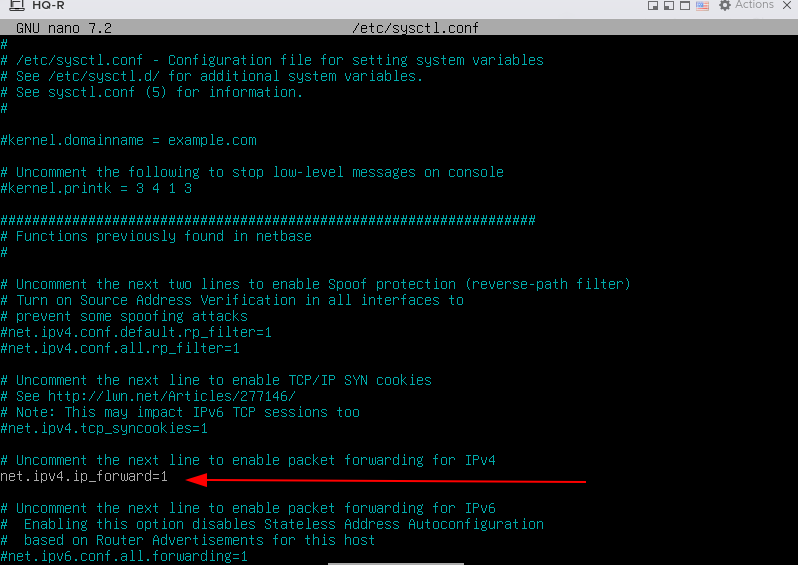


Рисунок 10 — настройка пересылки пакетов в режиме маршрутизатора

**Примечание: при каждой перезагрузке устройства, данная настройка будет изменятся обратно, что связано с загрузкой операционной системы на виртуальной машине для того, чтобы снова включить пересылку пакетов необходимо прописать sysctl -p**

Так же необходимо настроить похожую конфигурацию, для настройки ospf для протокола IPv6, первым делом настроим пересылку пакетов IPv6

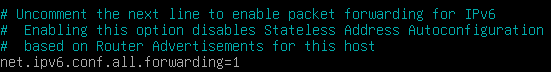


Рисунок 11 — настройка пересылки пакетов

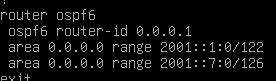
в режиме маршрутизатора IPv6

в vtysh посредством команд:

**Conf t**

**router ospf6**

Обозначить роутер ID, и зоны вокруг маршрутизатора

Рисунок 12 — настройка ospf6

Последним шагом в настройке OSPF6 необходимо, привязать зоны к интерфейсам маршрутизатора, т. к. разделение по зонам не обозначено, все интерфейсы и маршруты можно обозначить в одной зоне.

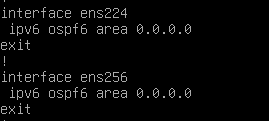


Рисунок 13 — обозначение зон ospf6 на интерфейсах

**Не стоит забывать о команде write !**

**Последним шагом можно составить топологию L3**

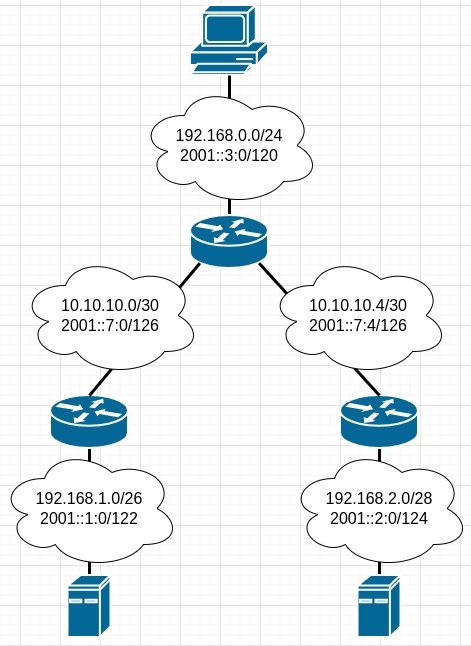


Рисунок 14 — топология L3

**3.Настройте автоматическое распределение IP-адресов на роутере HQ-R.**

**a. Учтите, что у сервера должен быть зарезервирован адрес.**

Первым шагом необходимо на машине HQ-R установить dhcp server командой

**apt install isc-dhcp-server**

После установки пакета следующим шагом необходимо сконфигурировать файл для указания интерфейсов прослушивания DHCP сервера зайти можно с помощью команды

**nano /etc/default/isc-dhcp-server**

и настроить интерфейс, направленный в сторону клиента, если в сети подразумевается DHCP-relay, то 2 интерфейса в сторону клиента, и в сторону сети откуда исходит запрос.



Рисунок 15 — Пример указания интерфейсов прослушивания

Далее необходимо настроить 2 конфигурационных файла для IPv4 для IPv6

Которые можно найти по путям **nano /etc/dhcp/dhcpd.conf** и **nano /etc/dhcp/dhcpd6.conf** соответственно

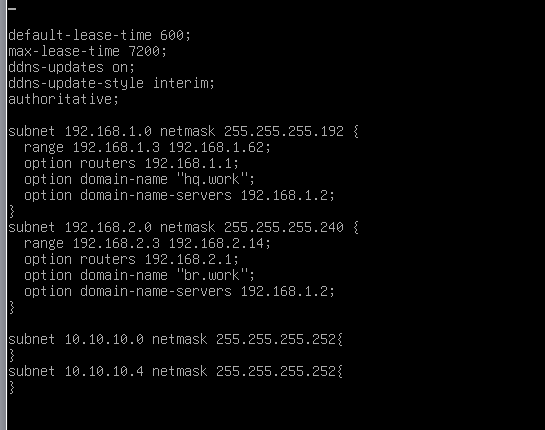


Рисунок 16 — Пример настройки DHCP для ipv4 без Relay

**ddns-update-style interim** — способ автообновления базы dns

**authoritative** — делает сервер доверенным

**subnet** — указание сети

**range** — пул адресов

**option routers** — шлюз по умолчанию

**Примечание:** *после каждого изменения конфигурации необходимо перезагружать DHCP сервер для применения конфигурации*

***systemctl stop isc-dhcp-server***

***systemctl start isc-dhcp-server***

*А для того, чтобы после перезагрузки DHCP-сервер автоматически включался можно воспользоваться командой* ***systemctl enable isc-dhcp-server***

Настройка DHCP по ipv6 имеет похожие настройки как показано на рисунке 17

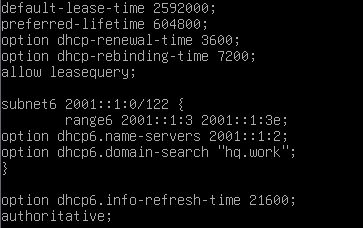


Рисунок 17 Пример настройки DHCP для IPv6

Однако dhcp6 не способен выдавать шлюз по умолчанию, эту функцию должен выполнять маршрутизатор

Поэтому для настройки маршрутизации для клиентов можно воспользоваться утилитой radvd

которую можно установить посредством команды

**apt install radvd**

После установки нужно сконфигурировать файл по пути /etc/radvd.conf следующего содержания

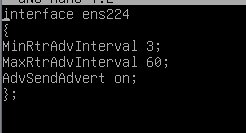


Рисунок 18 — Пример конфигурации Radvd

где **interface** — это имя интерфейса направленного в локальную сеть

**Min и MAX** интервалы — это интервалы рассылки объявлений

**AdvSendAdvert** — это разрешение на выдачу объявлений от маршрутизатор клиентам

После окончания конфигурирования так же необходимо перезагрузить службу Radvd и отправить в Enable

**systemctl stop radvd**

**systemctl start radvd**

**systemctl enable radvd**

**4.Настройте локальные учётные записи на всех устройствах в соответствии с таблицей 2.**

|  |  |  |
| --- | --- | --- |
| 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 BRSRV |

Для создания пользователей необходимо ввести комнаду

**adduser имя\_пользователя**

Затем появится поле ввода пароля как показано на рисунке 19

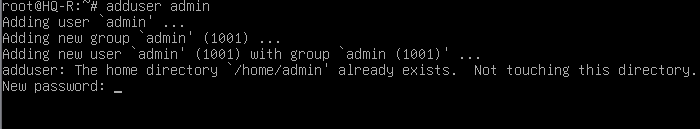


Рисунок 19 — окно ввода пароля при создании пользователя

Из необязательных параметров можно указать имя как показано на рисунке 20

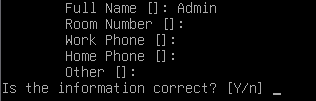


Рисунок 20 — параметры учётной записи

Так же возможно понадобится выдать Root права для данных клиентов это можно выполнить посредством команды **visudo**

в открывшемся окне необходимо вписать изменения для каждой новой созданной учётной записи как показано на рисунке 21



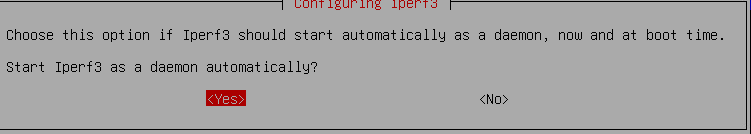
Рисунок 21 — выдача Root прав пользователям

**5.Измерьте пропускную способность сети между двумя узлами HQ-R-ISP по средствам утилиты iperf 3. Предоставьте описание пропускной способности канала со скриншотами.**

Для начала необходимо установит утилиту iperf3 (не путать с iperf) на машины HQ-R и ISP посредством команды

**apt install iperf3**

при установке будет показано окно автоматического включения демона, нужно выбрать пункт **yes** как показано на рисунке 22

Рисунок 22 — включения демона для iperf3

После установки на обоих машинах, достаточно воспользоваться командной

**iperf3 -c (ip адрес проверяемой машины) -i1 -t20**

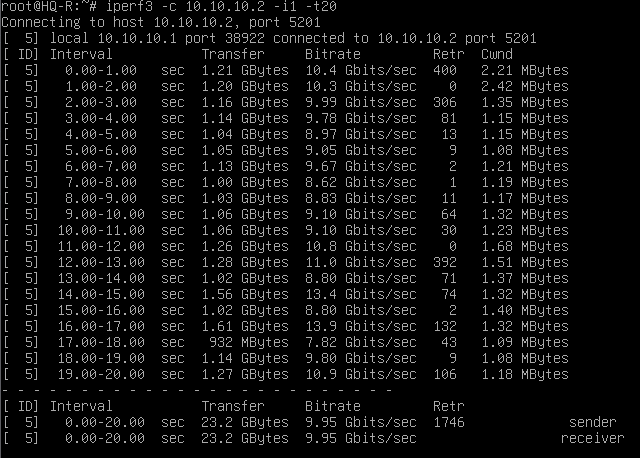


Рисунок 23 — скриншот описания пропускной способности

**6.Составьте backup скрипты для сохранения конфигурации сетевых устройств, а именно HQ-R BR-R. Продемонстрируйте их работу.**

Для начала на машинах HQ-R, BR-R создадим каталог, где будет хранится файл созданного скриптом бекапа.

Можно создать его в директории mnt

для этого пропишем **mkdir /mnt/backup**

Далее нам нужно создать сам файл для создания бэкап скрипта, для этого пропишем команду

**touch /etc/backup.sh**

зайдя в файл, необходимо прописать следующие параметры как показано на рисунке 24 или рисунке 25 **(по заданию достаточно упрощённого скрипта)**

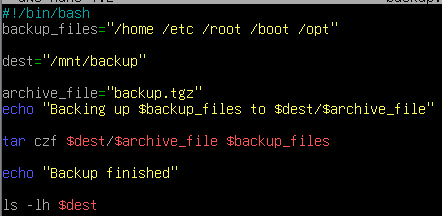


Рисунок 24 — упрощённый backup скрипт

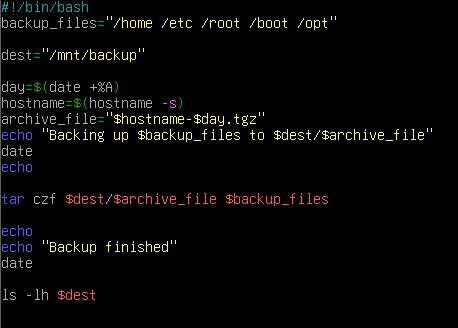


Рисунок 25 — расширенный backup скрипт

где **backup\_files** — копируемые директории

**dest** — место куда копируем директории

**day** — параметр который указывает день бэкапа

**hostname** — имя от кого он выполнился

**archive\_file** — конечное имя файла

**tar czf** — в месте указанное в dest помещает файл с именем указанным в archive\_file с содержимым указанным в backup\_files

**echo** — необязательные строки вывода

Для запуска скрипта достаточно написать **bash (имя\_файла)**

После создания скрипта для того, чтобы распаковать наш backup архив можно воспользоваться командой, указанной на рисунке 26 или 27



Рисунок 26 – распаковка простого backup архива

Рисунок 27 — распаковка сложного backup архива

Для того что бы не писать скрипт дважды, можно c помощью ssh перекинуть его на вторую машину посредством команды scp

для начала подключаемся по ssh командой ssh имя@адрес

**Пример:** ssh network\_admin@192.168.1.1

затем посредством команды

scp /расположение/имя\_файла имя@адрес :/расположение/имя\_файла

**Пример:**

**scp /etc/backup.sh network\_admin@192.168.2.1:/home/network\_admin**

После успешного копирования возвращаемся в нашу машину и можем перенести скрипт в любое более удобное место

**7. Настройте подключение по SSH для удалённого конфигурирования устройства HQ-SRV по порту 2222. Учтите, что вам необходимо перенаправить трафик на этот порт по средствам контролирования трафика.**

Первым делом необходимо перейти по пути **nano /etc/ssh/sshd\_config** где в окне конфигурации нам необходимо на HQ-SRV найти строку и изменить значения как указанно на рисунке 28

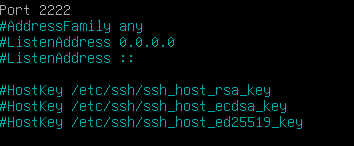


Рисунок 28 — смена порта доступа по ssh

Для применения конфигурации необходимо перезагрузить службу командой **systemctl restart ssh**

Для перенаправления трафика воспользуемся утилитой iptables-persistent

которая устанавливается командой **apt install iptables-persistent**

После установки создадим правило на подмену порта командой, указанной на рисунке 29

Рисунок 29 — правило iptables для подмены порта ssh

Для того что бы не прописывать команду при каждой перезагрузке сохраним нашу текущую конфигурацию командой

**iptables-save > /etc/iptables/rules.v4**

Которая будет подгружаться при каждой перезагрузке системы

**8.Настройте контроль доступа до HQ-SRV по SSH со всех устройств, кроме CLI.**

В зависимости от учётной записи, которая должна иметь доступ до сервера возможны следующие развития события, **если нам необходим доступ только от локальных учётных записей, то шаг 1 после всех настроек необходимо вернуть в исходный вид**

**Шаг 1**

Заходим в настройки ssh по пути использованному ранее

**nano /etc/ssh/sshd\_config**

находим и меняем строку как показано на рисунке 30

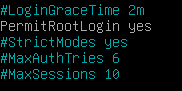


Рисунок 30 — разрешение доступа через root по ssh

***после сохранения изменений перезагружаем службу ssh***

**Шаг 2**

Следующим шагом необходимо создать ключ аутентификации ssh с помощью команды ssh-keygen -С «имя\_устройства\_с\_которого\_создан\_ключ» везде необходимо нажать ENTER пока не создастся ключ

Теперь необходимо перенести публичный ключ, на сервер к которому мы будем получать доступ с помощью команды ssh-copy-id имя@адрес

**Пример:**

ssh-copy-id root@192.168.1.2

ssh-copy-id admin@192.168.1.2

Последним шагом запретим любой доступ клиенту до нашего сервера

На HQ-SRV переходим по пути

**nano /etc/hosts.deny**

и вносим следующую строку в файл

**sshd: 192.168.0.2** (адрес машины CLI)

**перезагружаем ssh**

**В конце не забудьте отключить доступ по root, если иного не указано в задании !**

**Задание модуля 2**

**1. Настройте DNS-сервер на сервере HQ-SRV:**

a. На DNS сервере необходимо настроить 2 зоны

Зона hq.work, также не забудьте настроить обратную зону.

|  |  |  |
| --- | --- | --- |
| **HQ-R.hq.work** | **A,PTR** | **IP - адрес** |
| **HQ-SRV.hq.work** | **A,PTR** | **IP - адрес** |

Зона branch.work

|  |  |  |
| --- | --- | --- |
| **BR-R.branch.work** | **A,PTR** | **IP - адрес** |
| **BR-SRV.branch.work** | **A** | **IP - адрес** |

Вся настройка будет происходить на сервере HQ-SRV

Первым делом необходимо установить пакеты для dns командой

**apt install bind9 dnsutils**

**где:**

**bind9** — пакеты для создания dns сервера

**dnsutils** — дополнительные пакеты, которые помогут проверить работоспособность (команда host)

Следующим шагом необходимо создать зоны для прямого и обратного просмотра dns

Для этого переходим по пути **nano /etc/bind/named.conf.default-zones** и создаём зоны как показано на скриншотах ниже

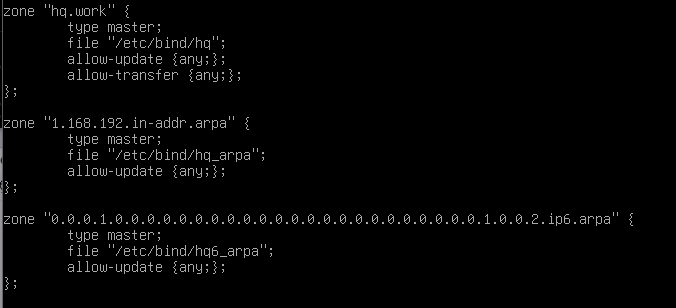


Рисунок 31 — зоны для hq.work

**где:**

**zone —** создаваемая зона

**type —** выбор между первичным и вторичным dns. (Master и Slave)

**file —** расположение конфигурационного файла зоны

**allow-update —** разрешение динамических обновлений

**где zone:**

**hq.work —** зона прямого просмотра

**in-addr.arpa —** зона обратного просмотра ipv4

**ip6.arpa —** зона обратного просмотра ipv6 (указывается полностью. В обратном порядке)

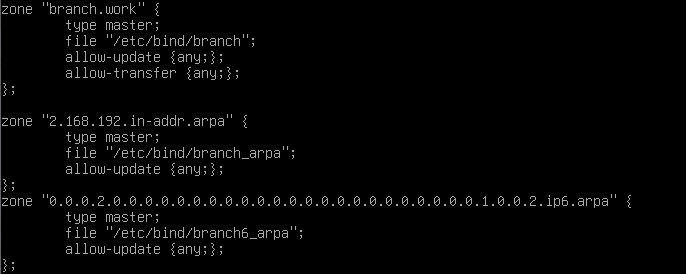


Рисунок 32 — зоны для branch.work

Следующим шагом необходимо создать конфигурационные вайлы для наших зон. Это можно сделать, скопировав стандартные шаблоны командой **cp**

Пример:

**cp /etc/bind/db.local /etc/bind/hq** — создание файла для прямой зоны

**cp /etc/bind/db.127 /etc/bind/hq\_arpa** — создание обратной зоны ipv4

*Зону для ipv6 скопируем после конфигурации зоны для ipv4 (так как по содержанию они не отличаются)*

Первым шагом сконфигурируем зону прямого просмотра, переходим по пути

**nano /etc/bind/hq** и конфигурируем файл как показано на скриншоте ниже

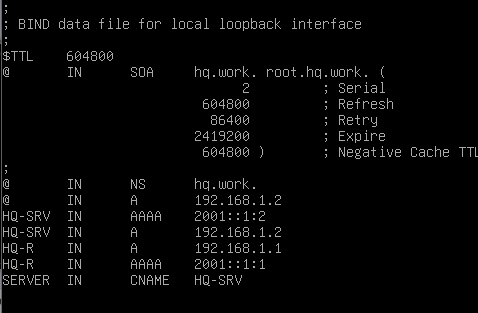


Рисунок 33 — зона прямого просмотра hq.work

**Где:**

**NS запись** — обозначение сервера отвественного за разрешение запросов к dns

**A запись** — основная запись для зоны прямого просмотра по протоколу ipv4

**АААА** запись - запись для зоны прямого просмотра по протоколу ipv6

**CNAME** — необязательный параметр, для указания альтернативного имени записи

Вторым шагом настроим зону обратного просмотра как указано на скриншоте ниже

Зона находится по пути

**nano /etc/bind/hq\_arpa**

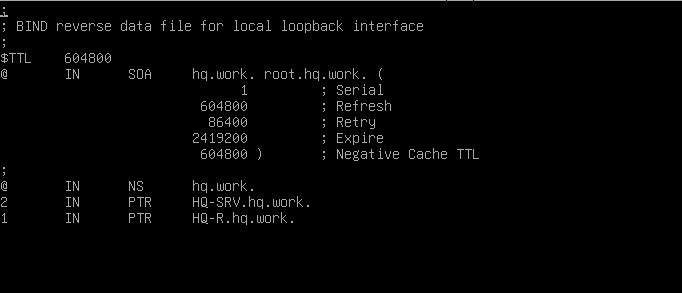


Рисунок 34 — настройка зоны обратного просмотра hq.work для ipv4

Где:

**PTR запись** — основная запись для зоны обратного просмотра

Третьим шагом настроим запись для зоны обратного просмотра для ipv6, для этого достаточно скопировать зону **hq\_arpa**, то есть

**cp /etc/bind/hq\_arpa /etc/bind/hq6\_arpa**

После создания всех конфигов необходимо перезагрузить службу bind9

**systemctl restart bind9** (**лучше stop и start**)

Похожая настройка выполняется для зоны **branch.work**

**Проверка выполняется посредством команд**

**host IP-адрес**

**host имя машины**

**Примечание:**

Не забывайте, что для br-srv по заданию нет PTR записи, её создание может считаться ошибкой

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

a. В качестве сервера должен выступать роутер HQ-R со стратумом 5

b. Используйте Loopback интерфейс на HQ-R, как источник сервера времени

c. Все остальные устройства и сервера должны синхронизировать свое время с роутером HQ-R

d. Все устройства и сервера настроены на московский часовой пояс (UTC +3)

Настройка производится на всех машинах, указанных в топологии, при этом настройка на машине, выступающей в роли NTP сервера уникальна, а на NTP клиентах идентична

Для начала на всех машинах необходимо установить московский часовой пояс, для этого следует воспользоваться командой

**timedatectl set-timezone Europe/Moscow**

Следующим шагом установим альтернативную службу NTP, под названием CHRONY, так как для задания 3, где происходит развёртывание домена, будет использоваться именно этот сервис. Устанавливаем с помощью команды:

**apt install chrony**

Произведём установку NTP сервиса Chrony

Далее следует осуществить настройку машины, выступающей в роли NTP сервера HQ-R, посредством команды

**nano /etc/chrony/chrony.conf**

осуществим вход в конфигурацию chrony, где следует установить значения как указано на рисунках 35 и 36

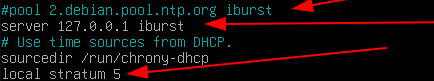


Рисунок 35 — указание адреса NTP сервера с определённым стратумом



Рисунок 36 — разрешение передачи NTP рассылок в указанной сети

**Примечание:** Нет необходимости указывать все сети которые присутствует в нашей сети, достаточно указать только одну сеть каждой машины , а так как у нас используется сети 192.168.0.0 , 192.168.1.0 и 192.168.2.0 , есть возможность взять сеть 192.168.0.0 с 22 маской которая будет включать в себя сеть начинающуюся с адреса 192.168.0.0 и заканчивающаяся адресом 192.168.3.255

Для настройки NTP клиентов chrony так же необходимо перейти в конфиг

**nano /etc/chrony/chrony.conf**

И необходимо провести внести изменения в конфиг как указано на рисунке 37



Рисунок 37 — Настройка NTP клиентам chrony

**Для проверки используйте команды chronyc tracking и chronyc sources**

**3. Настройте сервер домена выбор, его типа обоснуйте, на базе HQ-SRV через web интерфейс, выбор технологий обоснуйте.**

**a. Введите машины BR-SRV и CLI в данный домен**

**b. Организуйте отслеживание подключения к домену**

В качестве домена может быть выбраны один из двух вариантов, или SAMBA DC, или FREEIPA реализованная через DOCKER, т.к. экспериментальные версии freeipa-server больше не поддерживаются для системы DEBIAN, однако DOCKER позволяет реализовать freeipa-server для любой системы. ДЛЯ настройки будет выбрана именно FreeIpa.

Первым делом необходимо установить докер, воспользовавшись скриптом, который есть в открытом доступе, однако для этого нам необходимо экспортировать переменные окружения относящиеся к Proxy (Если Proxy отсутствует т. е. Пакеты с не стандартных репозиториев устанавливаются сами, то первый шаг можно пропустить)

Первым шагом необходимо посмотреть переменные, которые необходимо экспортировать, перейдя по пути

**nano /etc/apt/apt.conf.d/01proxy**

и посмотреть находящиеся там значения, после чего посредством команд

export http\_proxy=http(или https)://(адрес:порт)

export https\_proxy=http(или https)://(адрес:порт)

Экспортировать переменные прокси для доступа в интернет

ПРИМЕР:



Рисунок 38 — пример файла 01proxy

Команды для экспортирования переменных для конфига из рисунка 38

**export http\_proxy=http://10.0.70.52:3128**

**export https\_proxy=http://10.0.70.52:3128**

Вторым шагом посредством скрипта необходимо установить сам DOCKER, для этого необходимо ввести следующую команду

**wget -qO- https://get.docker.com | bash**

Вся установка происходит автоматически, и не должна выдавать ошибок, если были выполнены все предыдущие шаги

Третьим шагом необходимо запулить готовый контейнер с образом freeipa для centos-8-4.8.4 Для этого создаём каталог для автоматического запуска служб докера (**Необходимо если вы делали шаги с Proxy ранее**), командой

**mkdir -p /etc/systemd/system/docker.service.d**

Далее заходим в файл

**nano /etc/systemd/system/docker.service.d/http-proxy.conf**

и заполняем в соответствии с рисунком 39

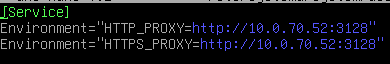


Рисунок 39 — настройка прокси для docker.service

После чего перезапускаем демона и сам докер командами **в указанном порядке**

**systemctl daemon-reload**

и

**systemctl restart docker**

После чего запускаем команду

**docker pull freeipa/freeipa-server:centos-8-4.8.4**

После окончания пула контейнера необходимо создать директорию, в которую будет монтироваться контейнер посредством команды

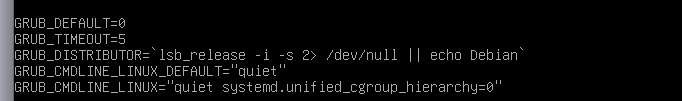
**mkdir -p /var/lib/ipa-data**

Также необходимо внести изменения в загрузчик системы для указания, необходимости использования обоих версий cgroup (механизм по ограничению ресурсов, начиная с 11 Debian по умолчанию включена только 2 версия)

Для этого посредством команды заходим в загрузчик ядра

**nano /etc/default/grub**

После чего вносятся изменения как показаны на рисунке 40

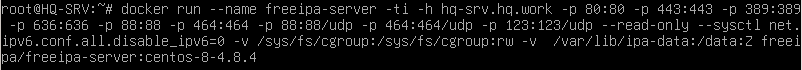
Рисунок 40 — Изменение параметров cgroup

Для применения изменений необходимо использовать команду

**grub-mkconfig -o /boot/grub/grub.cfg**

**После чего необходимо перезагрузить машину**

Следующим шагом уже переходим к запуску контейнера с хранящейся там FreeIPA, в качестве параметров ключей, указывает имя, указываем доменную сеть, а так открываем все необходимые для работы порты, указываем путь и образ, разрешаем конфликт с IPv6. Все параметры показаны на рисунке 41.

Рисунок 41 – запуск контейнера с указанием всех параметров

**Важное Примечание:** В случае завершения выполняемых функций в контейнере в результате которых оболочка может перейти в состояние freezing, или при успешном завершении, для выхода из оболочки окружения необходимо последовательно нажать сочетание клавиш **ctrl + p, а затем ctrl + q.** В случае если вам необходимо остановить контейнерможно воспользоваться командой **docker stop имя контейнера,** для удаления контейнера **docker rm имя контейнера,** для просмотра существующих образов **docker images**

**После успешного запуска необходимо заполнить форму:**

На вопрос о интеграции DNS нажимаем **Enter**

На вопрос о задании имени сервера нажимаем **Enter**

На вопрос о подтверждение имени домена нажимаем **Enter**

На вопрос о подтверждение имени области нажимаем **Enter**

На запрос ввода пароля для менеджера директорий вводим **P@ssw0rd**

На запрос ввода пароля для IPA админа вводим **P@ssw0rd**

На вопрос синхронизации с службой Chrony нажимаем **Enter**

На вопрос о конфигурирование системы с текущими параметрами вводим **yes**

Процесс установки достаточно длительный и может занимать около 5-10 или более минут.

После завершения установки необходимо подготовить машины, которые будут присоединены к домену. Для этого первым делом переходим по пути:

**Nano /etc/hosts**

И конфигурируем файл на клиенте как указано на рисунке 42

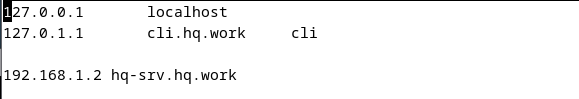


Рисунок 42 – конфигурация хостов машины CLI

Для машины BR-SRV настройка будет выглядеть как показано на рисунке 43

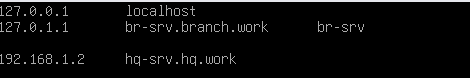


Рисунок 43 – конфигурация файла хостов машины BR-SRV

Следующим шагом посредством команды:

**apt install freeipa-client**

Производим установку клиентской части FreeIPA для ввода машины в домен.

На все всплывающие окна во время установки **нажимаем Enter**

После установки клиента, для ввода машины в домен необходимо прописать команды:

НА CLI

**ipa-client-install --mkhomedir --domain hq.work --server=hq-srv.hq.work -p admin -W**

НА BR-SRV

**ipa-client-install --mkhomedir --domain branch.work --server=hq-srv.hq.work -p admin -W**

Рисунок 43 – пример команды по вводу в домен на BR\_SRV

На сообщение о продолжении с фиксированными значения пишем **yes**

На вопрос о конфигурирование CHRONY нажимаем **ENTER**

На вопрос о конфигурировании с текущими значение пишем **yes**

Для проверки входа в FreeIPA, на клиентской машине необходимо открыть браузер и в адресной строке написать IP адрес машины HQ-SRV (192.168.1.2) логин и пароль для входа в вебку FreeIPA: admin и P@ssw0rd

**Важное Примечание:** если вы перезагрузите машину, то контейнер выключится, для его запуска можно воспользоваться командой **docker start freeipa-server**

**4. Реализуйте файловый SMB или NFS (выбор обоснуйте) сервер на базе сервера HQ-SRV.**

**a. Должны быть опубликованы общие папки по названиям:**

**i. Branch\_Files - только для пользователя Branch admin;**

**ii. Network - только для пользователя Network admin;**

**iii. Admin\_Files - только для пользователя Admin;**

**b. Каждая папка должна монтироваться на всех серверах в папку /mnt/<name\_folder> (например, /mnt/All\_files) автоматически при входе доменного пользователя в систему и отключаться при его выходе из сессии. Монтироваться должны только доступные пользователю каталоги.**

Исходя из поставленной задачи NFS будет более удачным выбором из-за его большей совместимости с системами Linux, при этом SMB крайне перегружена за счёт того, что создан для совместного использования широкого спектра сетевых ресурсов, включая службы файлов и печати, устройства хранения данных и хранилища виртуальных машин, в время как NFS, для совместного использования файлов и каталогов.

Поскольку файловый сервер будет работать на основе NFS , первым делом необходимо установить NFS сервер , посредством команды:

**apt install nfs-kernel-server**

Далее необходимо создать каталоги которые будут расшариваться.

**mkdir /mnt/al**l — создание корневого каталога в котором будут хранится остальные

**mkdir /mnt/all/Branch\_Files** — каталог для пользователя branch\_admin

**mkdir /mnt/all/Network** — каталог для пользователя network\_admin

**mkdir /mnt/all/Admin\_Files** — каталог для пользователя admin

Так же для того что бы монтируемые директории не были пустыми , и был виден результат монтирования посредством команд

**touch** **/mnt/all/Branch\_Files/123**

**touch /mnt/all/Network/234**

**touch /mnt/all/Admin\_Files/345**

Создадим файлы с разными имена в директориях

Далее посредством команды

**nano /etc/exports**

Заходим в конфигурационный файл , где будут прописываться все общие ресурсы и их параметры и заполняем как показано на рисунке 44

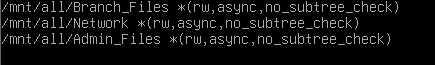


Рисунок 44 — создание общих ресурсов для пользователей

где:

/mnt/all/имя — Указание директории на сервере до которой будет выдан общий доступ

\* - указание IP адресов, которые имеют доступ в эту директорию (звёздочка значит все, так как по заданию не указано делать ограничения)

rw — разрешение на чтение и запись

async — включение обработки запросов клиента , до окончания предыдущего действия

no\_subtree\_check — отключает проверку вложенных директорий

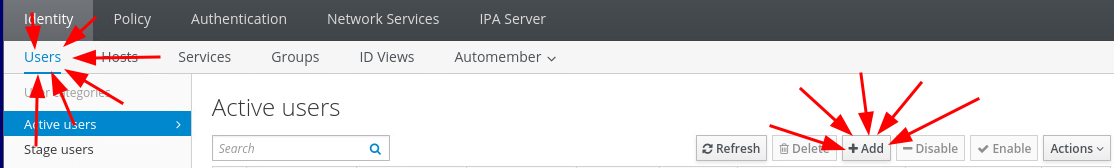
Для экспорта всех общих ресурсов необходимо воспользоваться командой

exportfs -ra

Также ещё одним необходимым шагом является создание доменных пользователей в Freeipa домене, для этого посредством адреса необходимо зайти в web-интерфейс Freeipa (адрес 192.168.1.2) , и сконфигурировать всех пользователей которые необходимы по заданию

**Важное примечание:** Пользователь admin является встроенной учётной записью и его конфигурировать не нужно.

Во вкладке users необходимо нажать кнопку add

Рисунок 45 — переход к процессу создания пользователя

И в появившемся окне необходимо заполнить следующие данные:

**user login** — **network\_admin** или **branch\_admin**

**first name — Network Admin** или **Branch Admin**

**last name - Network Admin** или **Branch Admin**

**New Password -** 123

**Verify Password -**123

Пароль задаётся 123 , поскольку после захода в систему, необходимо будет сменить пароль

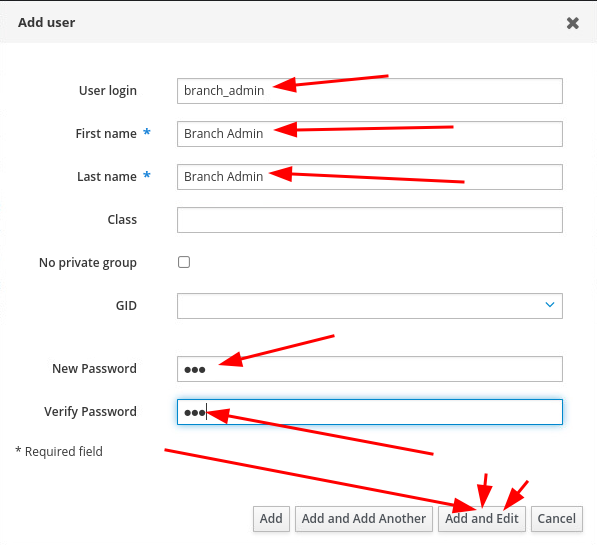
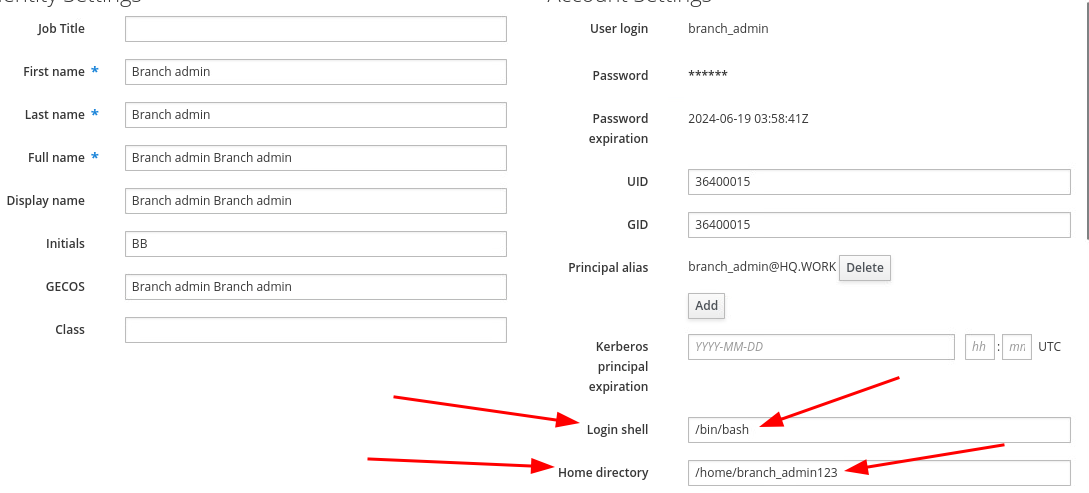


Рисунок 46 — пример окна создания доменного пользователя branch\_admin

После ввода параметров необходимо нажать Add and Edit

и сконфигурировать параметры Login shell и Home directory. Пример конфигурирования для пользователя branch\_admin указан на рисунке 47

Рисунок 47 — Изменения параметров пользователя branch\_admin

где:

**Login shell** — изменения оболочки окружения в которую будем попадать при входе с sh(shell) на bash

**Home directory** — изменение домашней директории , необходимо так как иначе она будет совпадать с директориями локальных пользователей созданных на машинах

После этого можно перейти к настройке клиента , т. к. в задании указано что монтирование должно осуществляться при входе доменного пользователя , настройка будет проводится на машинах которые занесены в домен CLI и BR-SRV

Первым шагом необходимо установить NFS-клиент и Pam модуль для автоматического монтирования разделов при входе пользователя командой:

**apt install nfs-common libpam-mount**

Так же необходимо создать каталог куда будет проводится монтирование

**mkdir /mnt/all**

После чего перейдя по пути

**nano /etc/security/pam\_mount.conf.xml**

Необходимо добавить строки приведённые ниже в разделе <volume definitions>

**<volume user="admin" fstype="nfs" server="192.168.1.2" path="/mnt/all/Admin\_Files" mountpoint="/mnt/all" />**

**<volume user="branch\_admin" fstype="nfs" server="192.168.1.2" path="/mnt/all/Branch\_Files" mountpoint="/mnt/all" />**

**<volume user="network" fstype="nfs" server="192.168.1.2" path="/mnt/all/Network" mountpoint="/mnt/all" />**

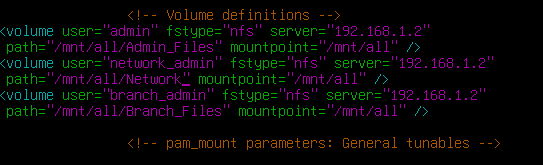


Рисунок 48 — пример настройки pamlib для всех пользоватлей

Для проверки работы общих ресурсов необходимо зайти под доменным пользователем, для этого посредством команды

**sudo login**

переходим в окно для входа в систему

в владке Login указывается пользователь по шаблону :

**имя@домен**

Пример:

**branch\_admin@hq.work**

В вкладке Password вводится пароль, для пользователей branch\_admin и network\_admin необходимо будет ввести пароль по схеме:

Password: 123

Current Password: 123

New password: P@ssw0rd

Retype new password: P@ssw0rd

И зайдя в пользователя проверить содержимое папки /mnt/all на наличие созданных файлов

**5. Сконфигурируйте веб-сервер LMS Apache на сервере BR-SRV:**

**a. На главной странице должен отражаться номер места**

**b. Используйте базу данных mySQL**

**c. Создайте пользователей в соответствии с таблицей, пароли у всех пользователей «P@ssw0rd»**

|  |  |
| --- | --- |
| Пользователя | Группа |
| Admin | Admin |
| Manager1 | Manager |
| Manager2 | Manager |
| Manager3 | Manager |
| User1 | WS |
| User2 | WS |
| User3 | WS |
| User4 | WS |
| User5 | TEAM |
| User6 | TEAM |
| User7 | TEAM |

Вся настройка пунктов A и B будет выполнятся исключительно на машине BR-SRV, для пункта C, а также проверки пункта A необходимо воспользоваться машиной CLI, так как на ней присутствует графика.

Первым шагом необходимо установить пакеты для веб-сервера APACHE и пакеты поддержки PHP, так как PHP, быстрее всего позволит создать страницу с номер места сдающего.

Для этого посредством команды

**Apt install apache2 libapache2-mod-php**

Устанавливаются пакеты для apache сервера и поддержки PHP сервером

Далее необходимо сконфигурировать страницу, которой в будущем заменится дефолтная страница APACHE, командой

**nano /var/www/html/mesto.php**

Создаётся страница, которую необходимо заполнить как указано на рисунке 49

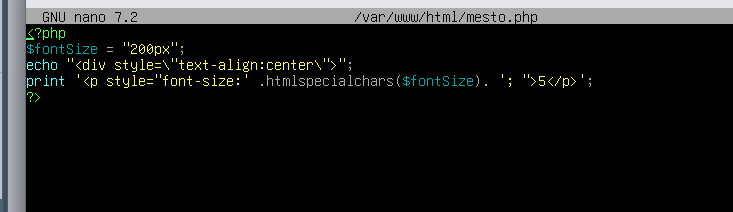


Рисунок 49 – Код PHP для создания страницы

Единственная часть кода, которую необходимо будет менять, это цифра 5 , её будет необходимо заменить на номер своего места.

Следующим шагом необходимо заменить дефолтную страницу, для того что бы при обращение к серверу в качестве главной страницы, показывался номер места, для этого перейдя по пути

**nano /etc/apache2/apache2.conf**

Переходим в конфигурационный файл, и ищем и заполняем раздел который указан на рисунке 50

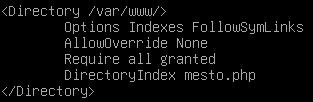


Рисунок 50 — замена главной страницы

Для проверки достаточно зайти на машину CLI, и в браузере прописать IP-адрес сервера, если ошибок допущено не было, должна быть выведена цифра по центру.

Далее переходим к установке базы данных , т. к. напрямую mysql-server установить не получится, будет использоваться пользовательский пакет, необходимо снова прописать команду для экспорта переменных которая была в задание 3.

Далее установим один из пакетов необходимых для работы mysql командой

**apt install gnupg**

Далее посредством команды указанной на рисунке ниже

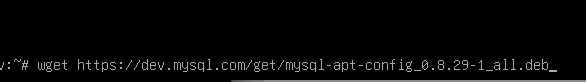


Рисунок 51 — указание места скачивание пакета

Указывается путь откуда будет скачиваться пакет, после чего для установки не user friendly пакетов, используется команда указанная на рисунке ниже



Рисунок 52 — установка пользовательского пакета для mysql

После чего при установке в появившемся окне просто выбирается вариант OK, как указано на рисунке ниже

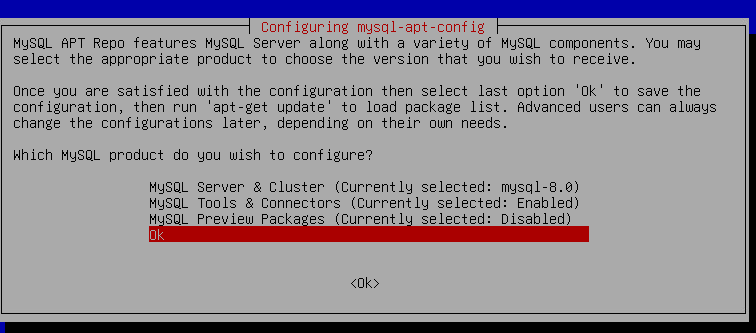


Рисунок 53 — согласие на установку предложенной конфигурации

Далее для обновления репозитория mysql необходимо прописать:

**apt update**

После успешного обновления, можно переходить к установке пакетов для mysql , для этого командой

**apt install mysql-server php-mysql**

Устанавливаются пакеты сервера, и его совместимости с php, второй из них пригодится чуть позже.

Во время установки будет предложено установить пароль, указывается пароль P@ssw0rd, на вопрос о плагине аутентификации выбирается первый вариант

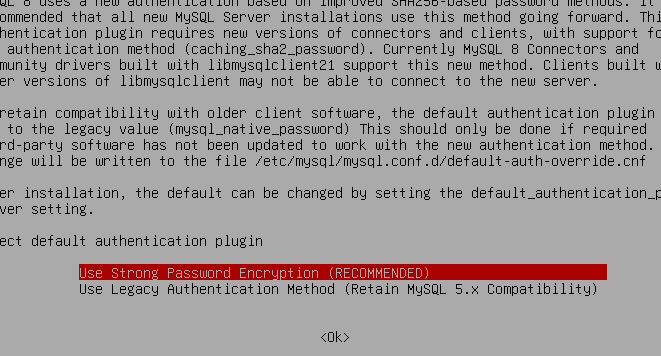


Рисунок 54 — выбор плагина аутентификации

Далее для создания пользователей и другим ,можно установить веб-интерфейс для субд mysql, под названием phpmyadmin, **этот шаг не обязателен, если вы самостоятельно можете создать пользователей и группы через консоль управления mysql-server**.

Для установки phpmyadmin, необходимо воспользоваться командой:

**apt install phpmyadmin**

Во время установки:

На вопрос о выборе сервера для конфигурации нажимаем **Space (Пробел)** напротив Apache2, что бы появилась звёздочка . После чего **Enter.**

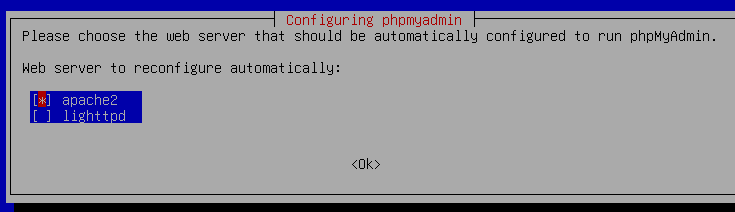


Рисунок 55 — выбор apache2 сервера.

На вопрос о конфигурирование БД для phpmyadmin выбирается вариант **YES**

Во всех вариантах где необходимо ввести пароль вводится пароль **P@ssw0rd**

Далее необходимо перейти по адресу

**IP-адрес сервера/phpmyadmin**

Пример

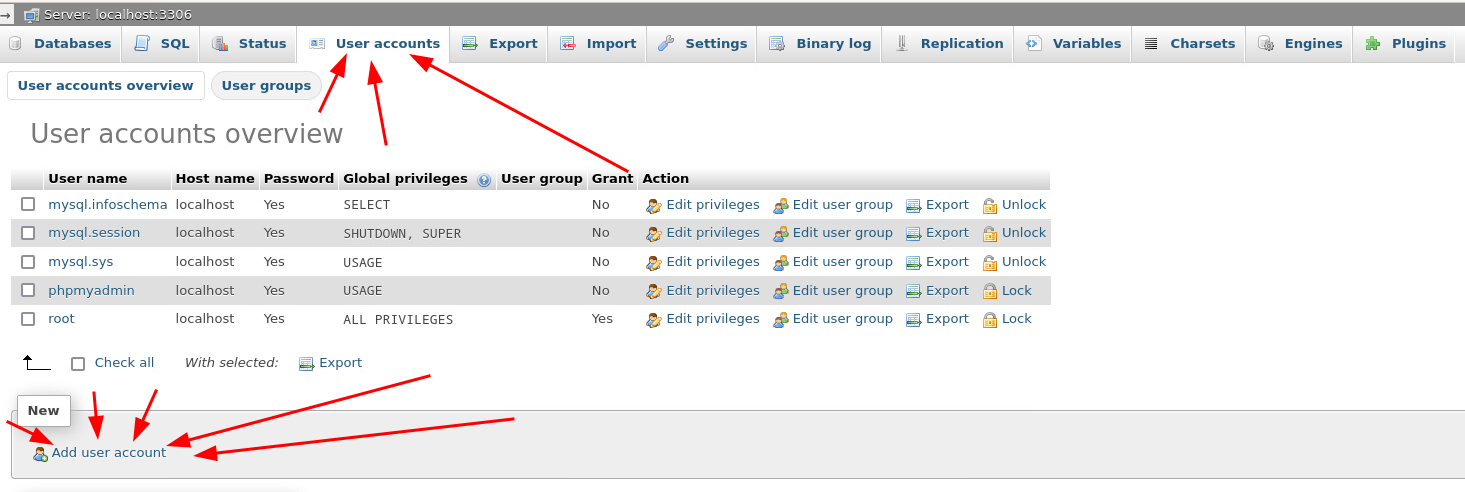
**192.168.2.2/phpmyadmin**

В окне авторизации:

**В поле Username вводится root (регистр имеет значение)**

**В поле Password вводится P@ssw0rd**

После успешной авторизации необходимо следовать инструкции на рисунках ниже:

Рисунок 56 — переход к окну создания пользователя

В открывшемся окне, заполняются указанные на рисунке поля

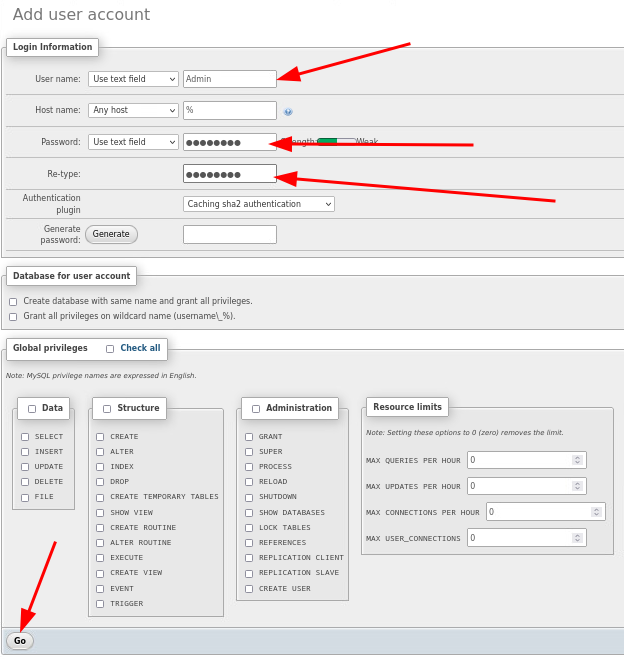


Рисунок 57 — пример создания пользователя Admin

После создания снова необходимо перейти в вкладку User accounts, как указано на рисунке 58

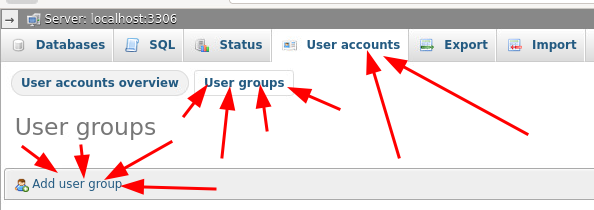


рисунок 58 — переход к вкладке создания пользовательских групп

В открывшемся окне указывается имя группы которая будет создана

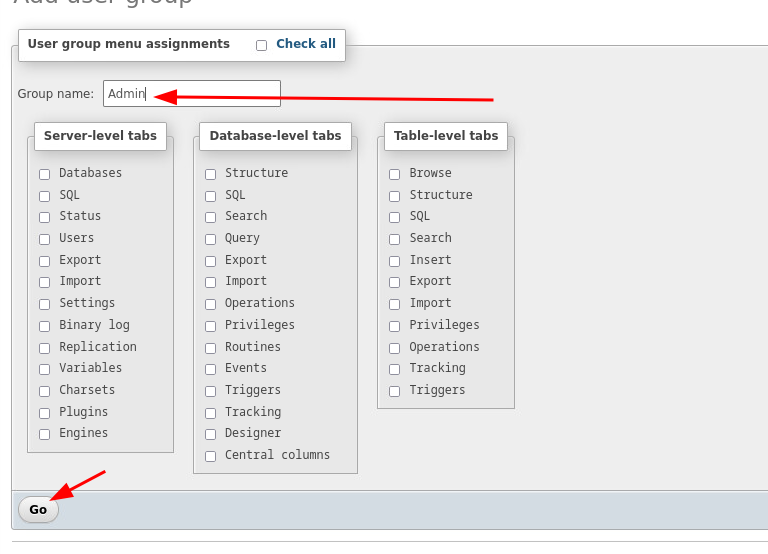
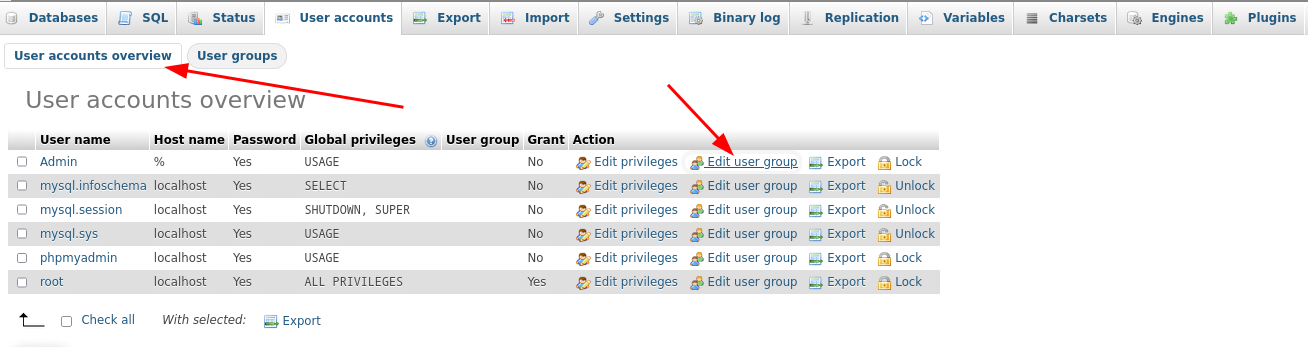


Рисунок 59 — создание пользовательской группы Admin

Далее перейдя по пути как указано на рисунке ниже

Рисунок 60 — переход к добавлению пользователя в группу

И в открывшемся окне, указывается группа

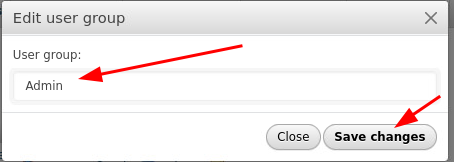


Рисунок 61 — Внесение пользователя в группу.

**6. Запустите сервис MediaWiki используя docker на сервере HQ-SRV.**

**a. Установите Docker и Docker Compose.**

**b. Создайте в домашней директории пользователя файл wiki.yml для приложения MediaWiki:**

**i. Средствами docker compose должен создаваться стек контейнеров с приложением MediaWiki и базой данных**

**ii. Используйте два сервиса;**

**iii. Основной контейнер MediaWiki должен называться wiki и использовать образ mediawiki;**

**iv. Файл LocalSettings.php с корректными настройками должен находиться в домашней папке пользователя и автоматически монтироваться в образ;**

**v. Контейнер с базой данных должен называться db и использовать образ mysql; vi. Он должен создавать базу с названием mediawiki, доступную по стандартному порту, для пользователя wiki с паролем DEP@ssw0rd;**

**vii. База должна храниться в отдельном volume с названием dbvolume.**

**MediaWiki должна быть доступна извне через порт 8080.**

Первым шагом необходимо установить docker compose, так как сам докер устанавливался в задании №3 второго модуля.

Для этого посредством команды, показанной на рисунке ниже, скачаем необходимый пакет.



Рисунок 62 – скачивание пакета для docker-compose

Командой, указанной на рисунке ниже, выдаём необходимые права для скаченной службы



Рисунок 63 – выдача прав для docker-compose

Далее для того, чтобы с нуля не писать yml файл, можно скачать похожий по смыслу файл, приведённый на рисунке ниже (если будет запрещено, будете писать сами)

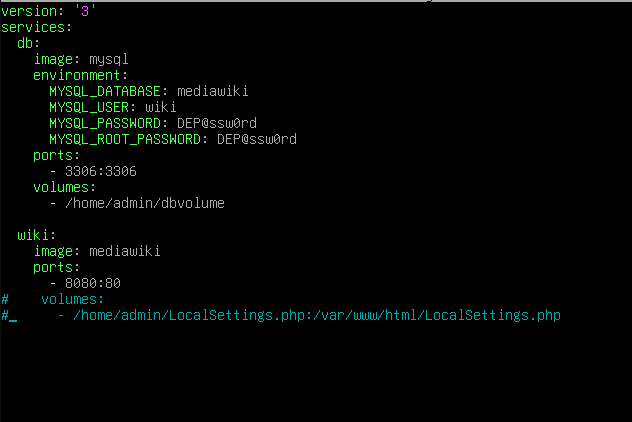


Рисунок 64 – скачивание yml файла для последующего изменения

После чего открываем скачанный файл по пути

**Nano /home/admin/wiki.yml**

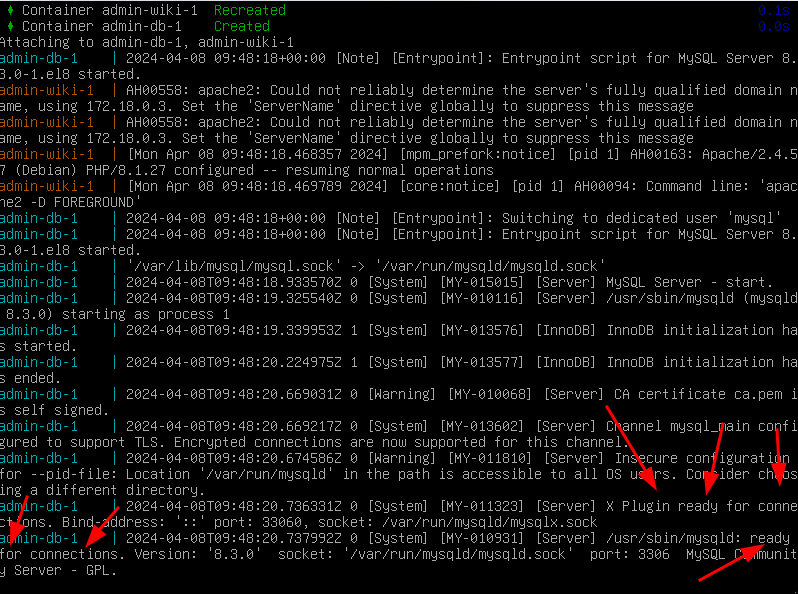
И приводим к виду, указанному на рисунке ниже, **не удаляя присутствующие на рисунке закоменченные строки ! Соблюдая расстановку пробелов ! Заголовки первого порядка (Нажимаем один TAB или 2 пробела) , Второго порядка (2 TAB или 4 пробела), Третьего порядка (3 TAB или 6 пробелов).**

Рисунок 65 – создание yml файла с двумя контейнерами и указанием параметров

После чего запускаем контейнеры посредством команды

**docker-compose -f /home/admin/wiki.yml up**

После чего начнётся загрузка служб, после загрузки необходимо дождаться запуска контейнеров с сообщением о готовности подключения

Рисунок 66 — Пример окна с запущенными контейнерами служб

Далее необходимо перейти на машину CLI , и в браузере перейти по адресу **192.168.1.2:8080**

Рисунок 67 — Стартовая страница MediaWiki с отсутсвующим файлом настроек

Перейди по ссылке необходимо нажать → **Continue**

Затем внизу страницы снова → **Continue**

Далее на следующей странице необходимо указать настройки по заданию , как указано на рисунке ниже. **Пароль DEP@ssw0rd**

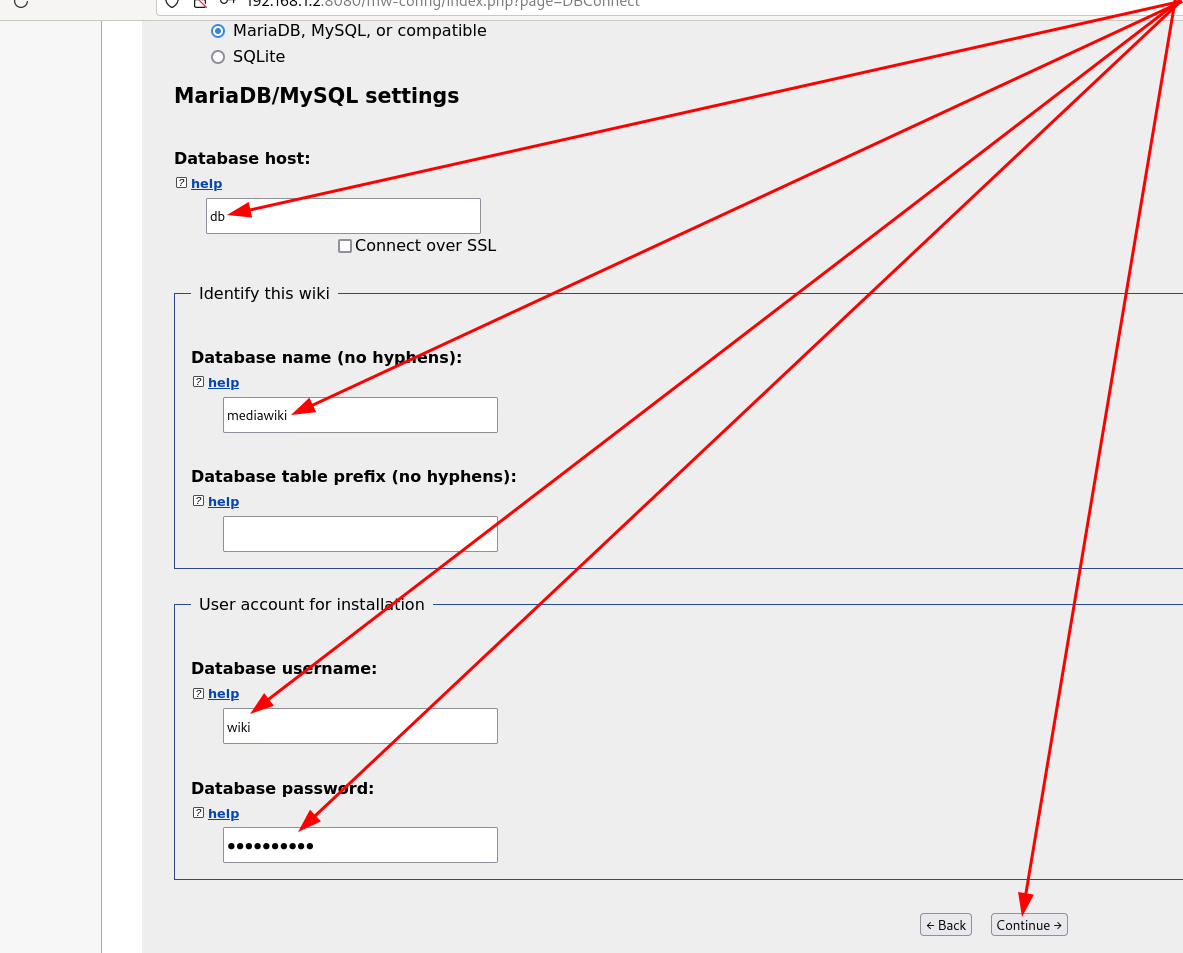


Рисунок 68 — указание настроек БД для MediaWiki

На следующей странице нажимаем → **Continue**

Далее на следующей странице, заполняем поля как указано на рисунке ниже, обязательно не забыв поставить галочку о том что вы очень занятой. **Пароль DEP@ssw0rd**

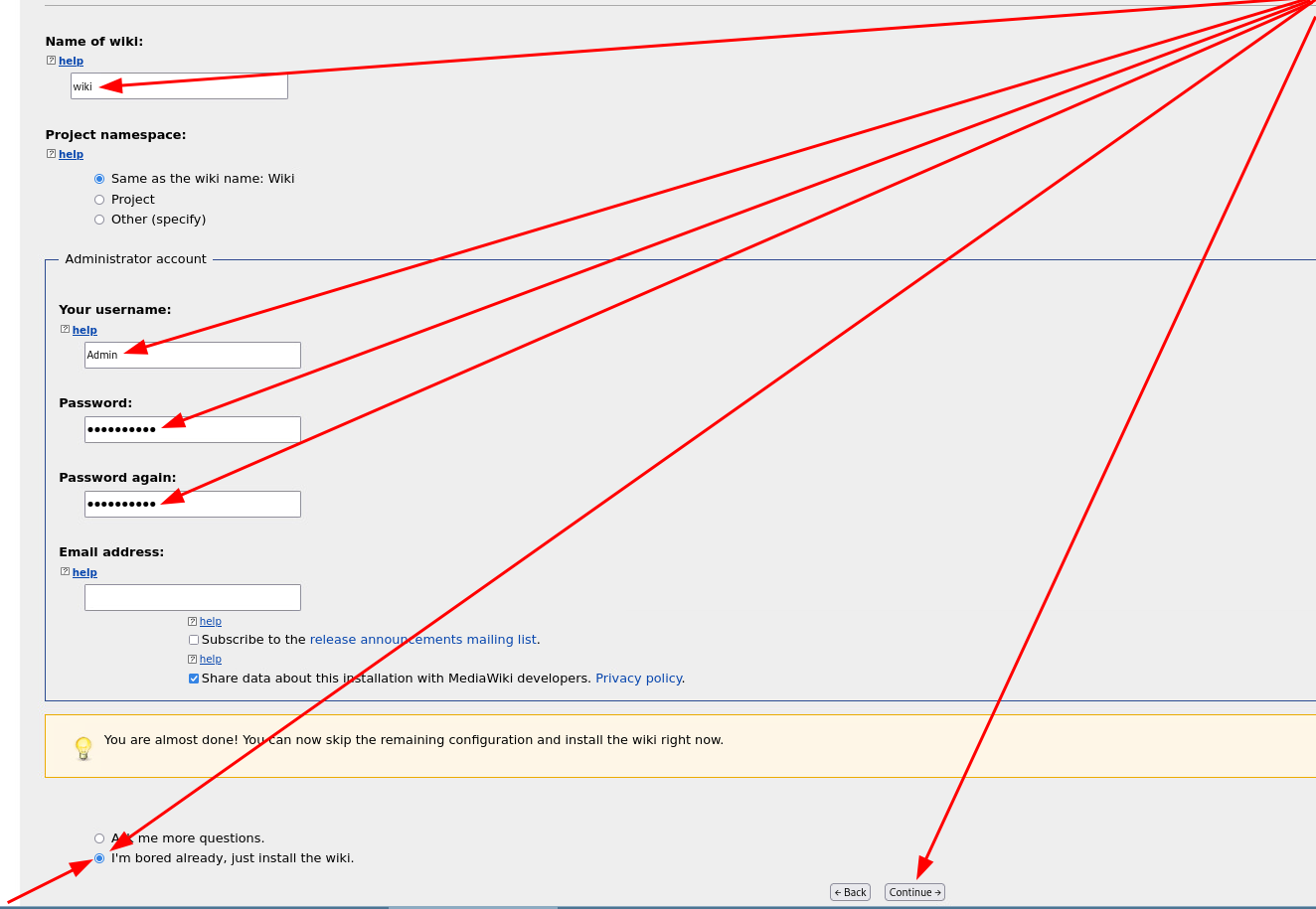


Рисунок 69 — заполнение данных для работы в MediaWiki

После чего сконфигурированный файл автоматически будет скачен в загрузки

Далее его необходимо перенести на сервер. Если вы выполнили задание с запретом доступа по SSH с машины CLI, файл необходимо будет кидать не напрямую а через промежуточную машину HQ-R

Для этого воспользовавшись командами

**На машине CLI от юзера админ (У вас может быть другой пользователь в зависимости от кого вы авторизировались в систему):**

**scp /home/admin/Downloads/LocalSettings.php root@192.168.1.1:/home/admin**

На машине HQ-R:

**scp /home/admin/LocalSettings.php admin@192.168.1.2:/home/admin/**

После чего необходимо на машине HQ-SRV перейти по пути

**nano /home/admin/wiki.yml**

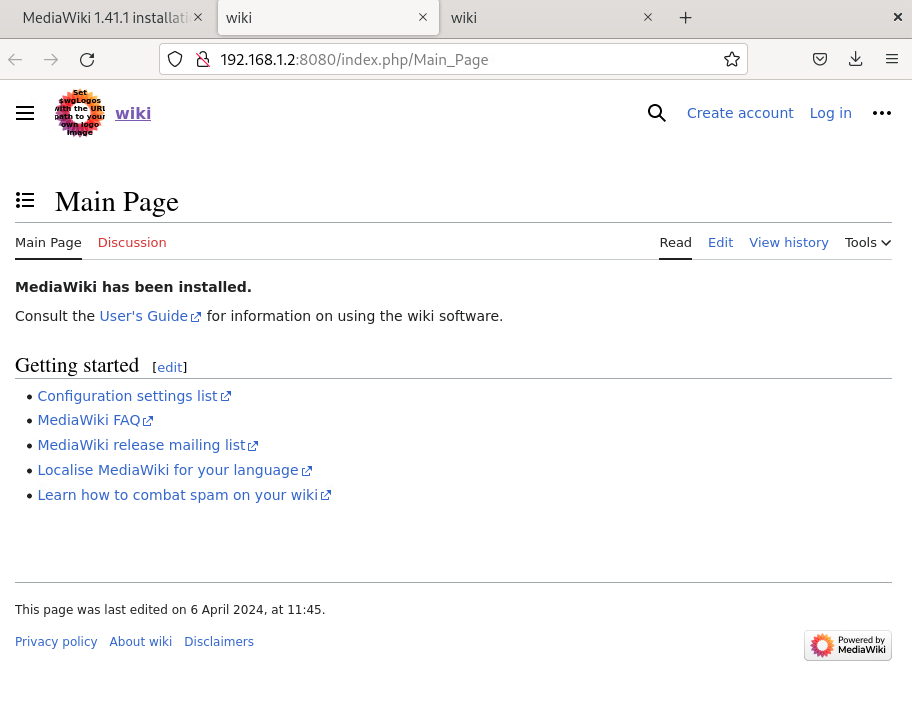
**И раскоментить и переписать (если они у вас отличаются) строки указанные на рисунке ниже**



Рисунок 70 — Подлючение файла настроек к MediaWiki

**После чего снова запустить контейнеры.**

И теперь перейдя на машину CLI и зайдя в браузере по тому же адресу. Должна загрузится главная страница MediaWiki

Рисунок 71 — главная страница «вашей» wiki