Задание для демонстрационного экзамена по компетенции

Системное и сетевое администрирование

Образец задания для демонстрационного экзамена по комплекту оценочной документации.

Предисловие:

Администрирование сетей и систем является одной из ключевых задач при создании и поддержке информационной инфраструктуры организации. Качественная настройка и управление сетевыми компонентами, серверами и сервисами играет важную роль в обеспечении стабильной и безопасной работы всей системы.

Данное задание представляет собой комплексную программу по администрированию, которая включает в себя несколько модулей. Каждый модуль охватывает различные аспекты настройки и поддержки системы, начиная с базовой конфигурации устройств и заканчивая реализацией сложных сервисов и технологий.

Модуль А посвящен базовой настройке устройств, включая присвоение имен, расчет IP-адресации и настройку внутренней динамической маршрутизации. Эти шаги позволяют создать основу для дальнейшего развития и масштабирования сети.

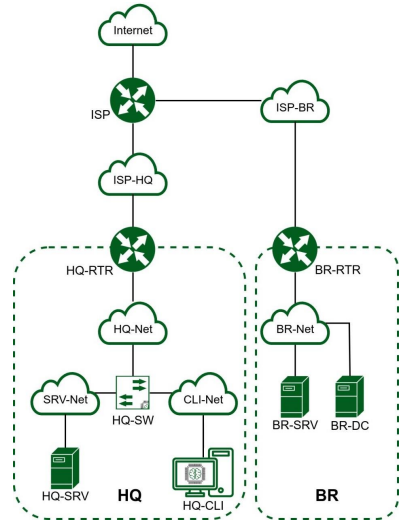
Модуль Б фокусируется на настройке DNS-сервера, синхронизации времени между устройствами, реализации файлового SMB(NFS)-сервера и других сервисов, таких как мониторинг и центр сертификации. Все эти шаги направлены на обеспечение безопасности, доступности и функциональности системы.

Модуль B включает в себя миграцию пользователей, подразделений, файловых служб, развёртывание и настройку центра сертификации, выдачу сертификатов веб серверам для шифрования трафика, настройку шифрованного туннеля, настройку межсетевого экрана, принт-сервера, сервера логирования и мониторинга, автоматизации на основе инфраструктуры открытых ключей.

Цель данного задания состоит в том, чтобы разработать и настроить комплексную систему, которая удовлетворяет требованиям безопасности, функциональности и производительности. Работа в рамках этого задания требует глубоких знаний и навыков в области администрирования сетей и систем, а также умения применять современные технологии и методы для достижения поставленных целей.

Учтите, что в некоторых заданиях необходимо составить отчёт о проделанной работе в электронном виде.

**Топология сети**



Для начала нам необходимо удостовериться, что мы можем собрать эту структуру, для этого необходимо воспользоваться нашими сайтами.

Если вы находитесь в локальной сети колледжа, то вам будет достаточно подключиться напрямую к сайту <https://vcd.ktk-45.ru:3002> и ip адрес сервера eve-ng.

Если вы находитесь вне локальной сети колледжа, то вам необходимо подключиться на прямую к сайту <https://vcd.ktk-45.ru:3002>, а также, что бы получить доступ до eve-ng, вам необходимо воспользоваться сайтом удаленного рабочего стола в нашу локальную сесть <https://horizon.ktk-45.ru:3004>. Далее перейдя по ip адресу сервера eve-ng.

Вам необходимо удостовериться, что вы можете подключиться, а также аутентифицироваться на сайтах и иметь возможность создавать лаборатории и виртуальные машины.

В нашем решении демоэкзамена в дальнейшем – ДЭ, мы будем использовать Linux Debian 12, vCloud Director, EVE-NG, Cisco Switch V15.2, Cisco Router V15.7.

Для начала нам необходимо определиться с необходимыми характеристиками для виртуальных машин, устройства cisco мы не изменяем в EVE-NG.

|  |  |  |  |
| --- | --- | --- | --- |
| **Машина** | **RAM, ГБ** | **CPU** | **OS** |
| ISP | - | - | Cisco Router |
| HQ-RTR | - | - | Cisco Router |
| HQ-SRV | 4 | 4 | Debian 12 Server |
| HQ-CLI | 4 | 4 | Debian 12 Desktop |
| BR-RTR | - | - | Cisco Router |
| BR-SRV | 4 | 4 | Debian 12 Server |
| HQ-SW | - | - | Cisco Switch |
| CLI | 4 | 4 | Debian 12 Desktop |

Также нам необходимо решить, какие сети нас будут окружать в нашей лаборатории, давайте составим таблицу отношения какие подсети будут соединять наши виртуальные машины. Но в этом случае мы не можем написать, что хотим, у нас есть условия, которым мы должны придерживаться:

* Интерфейс, к которому подключен **HQ-RTR**, подключен к сети 172.16.4.0/28;
* Интерфейс, к которому подключен **BR-RTR**, подключен к сети 172.16.5.0/28;
* IP-адрес должен быть из приватного диапазона, в случае, если сеть локальная, согласно **RFC1918**;
* Локальная сеть в сторону **HQ-SRV**(VLAN100) должна вмещать не более 64 адресов;
* Локальная сеть в сторону **HQ-CLI**(VLAN200) должна вмещать не более 16 адресов;
* Локальная сеть в сторону **BR-SRV** должна вмещать не более 32 адресов;
* Локальная сеть для управления(VLAN999) должна вмещать не более 8 адресов.

**RFC1918** - IP-адрес, принадлежащий к специальному диапазону, не используемому в сети Интернет. Такие адреса предназначены для применения в локальных сетях, распределение таких адресов никем не контролируется.

Значения:

* 10.0.0.0 — 10.255.255.255 (маска подсети: 255.0.0.0 или /8);
* 172.16.0.0 — 172.31.255.255 (маска подсети: 255.240.0.0 или /12);
* 192.168.0.0 — 192.168.255.255 (маска подсети: 255.255.0.0 или /16).

|  |  |  |
| --- | --- | --- |
| **Машина 1** | **Подсеть** | **Машина 2** |
| Internet | 100.100.1№.0/24 | ISP |
| ISP | 172.16.4.0/28 (255.255.255.240 15 ip адресов) | HQ-RTR |
| ISP | 172.16.5.0/28 (255.255.255.240 15 ip адресов) | BR-RTR |
| ISP | 172.16.6.0/28 (255.255.255.240 15 ip адресов) | CLI |
| HQ-RTR | 192.168.11.0/26 (255.255.255.192 62 ip адреса) | HQ-SRV |
| HQ-RTR | 192.168.12.0/28 (255.255.255.240 15 ip адреса) | HQ-CLI |
| BR-RTR | 192.168.21.0/27 (255.255.255.224 30 ip адреса) | BR-SRV |
| HQ-RTR | 192.168.1.0/29 (255.255.255.248 7 ip адреса) | HQ-SW |
| HQ-RTR(VPN) | 192.168.254.0/30 (255.255.255.252 7 ip адреса) | BR-RTR(VPN) |

Далее давайте распределим ip адресацию на устройствах:

|  |  |  |
| --- | --- | --- |
| **Машина** | **Интерфейс** | **IP** |
| ISP | e0/0 | 100.100.1№.1/24 |
| ISP | e0/1 | 172.16.4.14/28 |
| ISP | e0/2 | 172.16.5.14/28 |
| ISP | e0/3 | 172.16.6.14/28 |
| CLI | Ethernet | 172.16.6.1/28 |
| HQ-RTR | e0/0 | 172.16.4.1/28 |
| HQ-RTR | e0/1 | - |
| HQ-RTR | e0/1.100 | 192.168.11.61/26 |
| HQ-RTR | e0/1.200 | 192.168.12.14/28 |
| HQ-RTR | e0/1.999 | 192.168.1.6/29 |
| HQ-RTR | t0 | 192.168.254.1/30 |
| HQ-SRV | Ethernet | 192.168.11.1/26 |
| HQ-CLI | Ethernet | 192.168.12.1/28 |
| BR-RTR | e0/0 | 172.16.5.1/28 |
| BR-RTR | e0/1 | 192.168.21.30/27 |
| BR-RTR | t0 | 192.168.254.2/30 |
| BR-SRV | Ethernet | 192.168.21.1/27 |
| HQ-SW | VLAN 999 | 192.168.1.1/29 |

Также необходимо понять, что в данный момент в ДЭ для установки пакетов или скачивания других ресурсов и программ необходимо подключить реальный интернет, так как сейчас нет **offline** решения для развертки всего необходимого. Поэтому необходимо для начала установить везде правильное сетевое взаимодействие между устройствами, а позже устанавливать пакеты. Для этого необходимо понять, что для выхода в интернет необходимо везде устанавливать шлюз, шлюз должен быть настроен на ip адрес вашего конечного роутера. То есть, если вы настраиваете шлюз на **HQ-SRV**, то он будет **192.168.11.61**.

Это особый тип статического маршрута, который определяет путь, который должно использовать устройство, если адрес назначения не указан в других маршрутах. Если путь до адреса назначения не указан, устройство будет отправлять пакеты на основной шлюз, который перенаправит их по адресу назначения.

Для начала я объясню, как система работает.

Для вас выделен **1 VLAN** как выход в интернет и этот адрес выделяется для каждого пользователя ДЭ, обычно это (100.100.1(номер пользователя).0/24).

В моем случае это 100.100.161.0/24.

Через этот **VLAN** у вас будет проброс ваших виртуальных машин в **EVE-NG**.

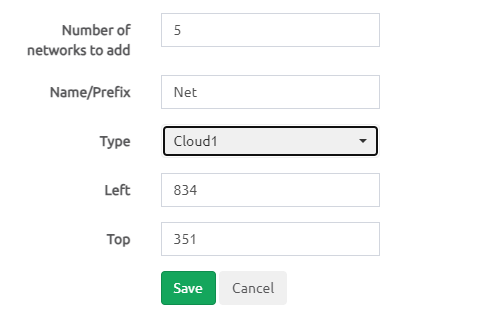
Что я имею ввиду, **VLAN**, он не ограничен на использование 1-й подсети, вы можете в ней использовать множество подсетей, но они некогда не будут работать для выхода в интернет, а это значит, что вы можете пользоваться этим **VLAN** для создания любых подсетей, которые будут работать между вашим **EVE-NG** и **vCloud Director**.

Но есть одно исключение, которое говорит о резервировании вашей подсети для выхода в интернет, для меня это 100.100.161.0/24.

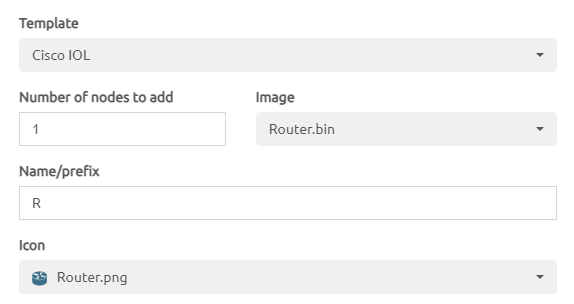
Получается, если вы в **vCloud Director** создадите виртуальную машину и настроите на ней подсеть **192.168.11.1/26**, то в **EVE-NG** вы можете создать облако **Cloud1** который будет содержать в себе эту подсеть и многие другие, так как **Cloud1** это просто замаскированный **VLAN**. И вы сможете соединить ваше сетевое оборудование с **Cloud1** по подсети **192.168.11.1/26**. Но на самом деле попадете в настроенную виртуальную машину.

Для начала давайте перейдем в **EVE-NG** и создадим новую лабораторию.

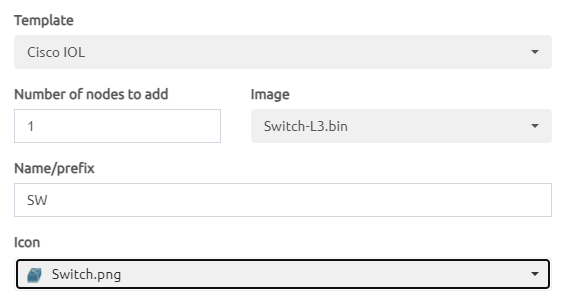
Для начала создадим облачную структуру, нам понадобится **5 Network** с разными названиями.



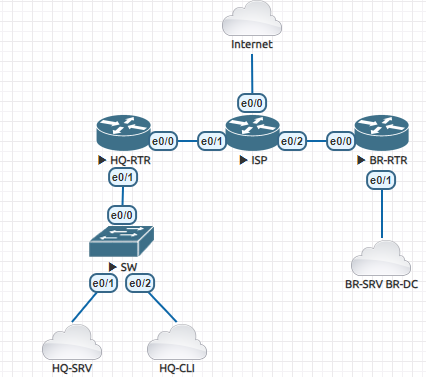
Далее необходимо добавить сетевое оборудование, для этого необходимо создать 3 маршрутизатора **Router IOL**.



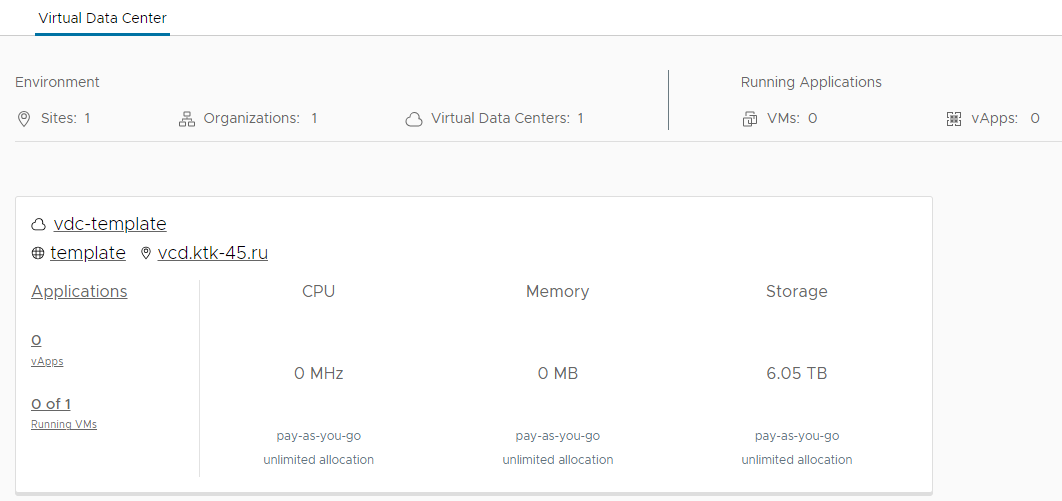
А также 1 коммутатор **Switch L3 IOL**.



У вас должно получиться такое построение в лаборатории:



Далее нам предстоит работа в **vCloud Director**, нам необходимо создать виртуальные машины согласно таблице в самом начале. Переходим в облачную структуру, авторизуемся и переходим в **Virtual Data Center**.



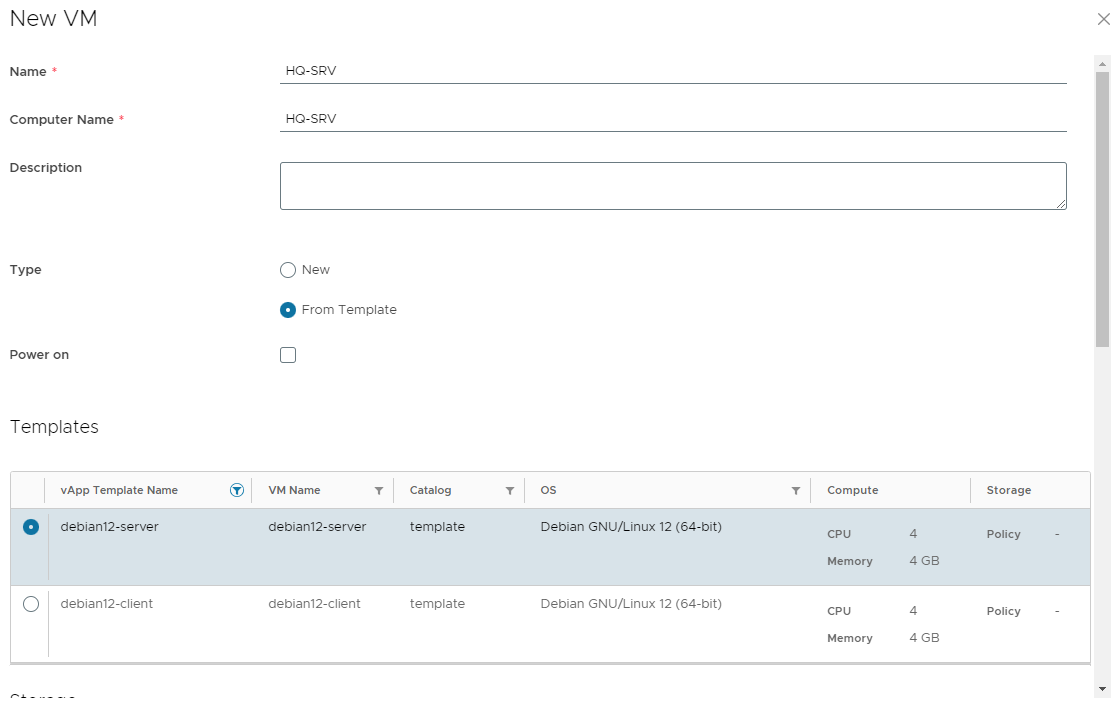
В данном ДЭ мы не будем использовать отдельные **Isolated** сети, поэтому, нам не нужно их создавать, все созданные виртуальные машины мы будем подключать к сети **Internet**, но на самом деле это отдельный **VLAN** про который я описывал выше.

Нам понадобится 4 виртуальных машины Debian 12.

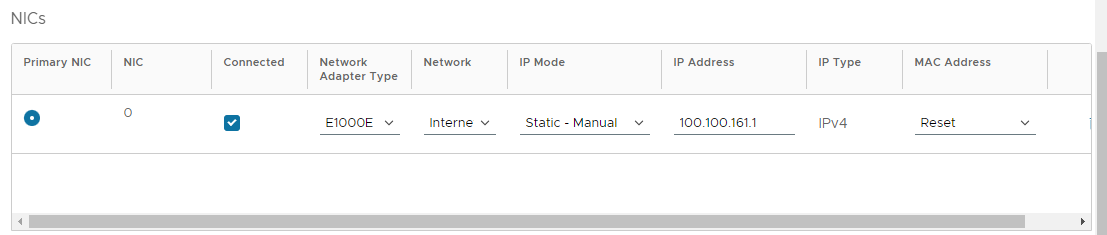
Далее переходим в вкладку виртуальные машины и нажимаем кнопку «New».

Называем виртуальные машины **строго как в задании**.

Обязательно выключаем отметку «**Power on**».

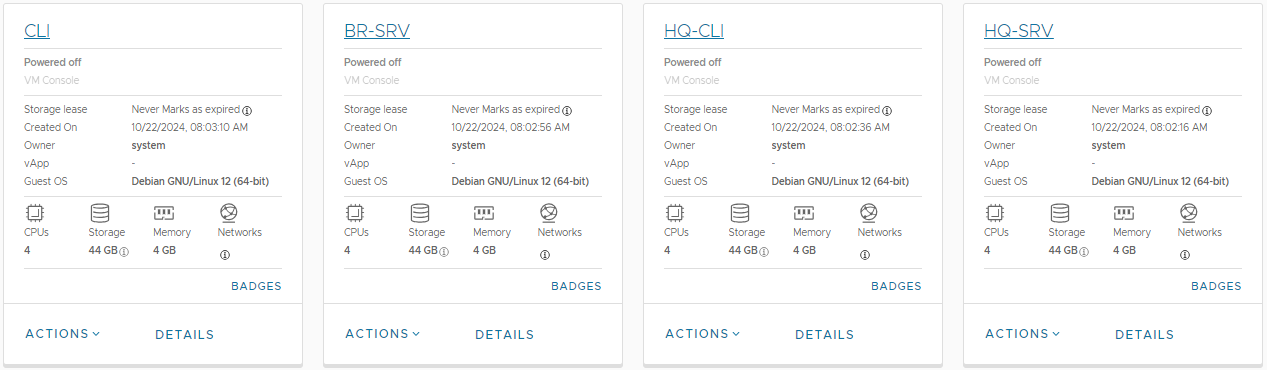


Далее необходимо изменить настройки интерфейсов виртуальной машины.



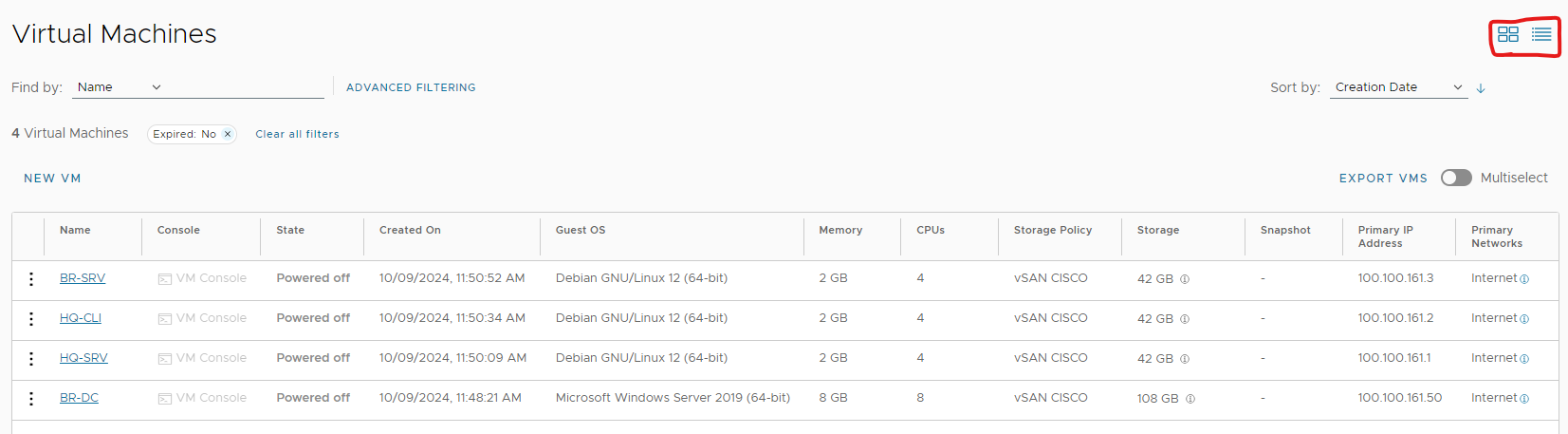
В данный настройке ip адрес будет отличаться от оригинального, так как система развертки так работает. Необходимо устанавливать не повторяющиеся ip адреса из подсети Internet. Если вы не знаете свою подсеть вы можете ее посмотреть в Networks вкладке, в левой навигационной панели. MAC адрес обязательно сбрасывать. А также если не удается указать ip адрес, пишет, что необходимо ввести ipv6, необходимо в поле «Network» перевыбрать интерфейс Internet. А также везде выбираем сетевой интерфейс E1000E.

Далее создаете 2 - Debian 12 сервера и 2 - Debian 12 клиент.



Далее эти виртуальные машины необходимо настроить, для этого переходим в детали виртуальной машины и устанавливаем необходимо количество **CPU** и **Memory** в вкладке «**Compute**», а также необходимо выключить гостевую настройку в вкладке «**Guest OS Customization**»-«**Enable guest customization**».

У вас должно получиться такое решение:



Далее переходим к настройке. Включаем все оборудование.

Пароль Server – root/toor.

Пароль Client – user/toor, далее в terminal вводим команду «su root», чтобы выполнять команды от root. Пароль root/toor.

Для начала давайте сделаем первоначальные настройки виртуальных машин и сетевого оборудования, а именно:

* имена;
* ip адресация;
* DNS север;

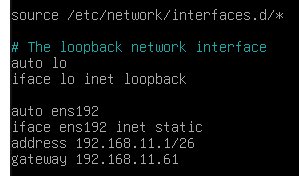
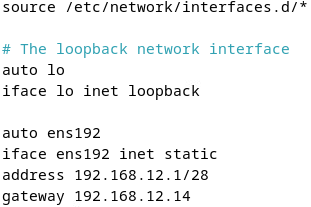
Имя в **Linux** изменяется в файле **/etc/hostname**;

IP адресация изменяется в файле **/etc/network/ineterfaces**;

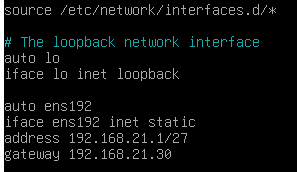
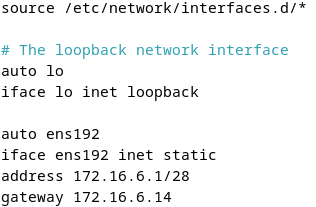
**DNS** север изменяется в файле **/etc/resolv.conf**.

Напоминаю, что на всех устройствах необходимо настроить шлюз для работы интернета. Настройки должны выглядеть вот так:

**HQ-SRV HQ-CLI**



**BR-SRV CLI**



А также файл **resolv.conf** будет у всех одинаковый:



Не забываем перезагрузить виртуальные машины после настройки.

Далее настроим ip конфигурацию, **hostname** и **VLAN** на сетевом оборудовании. Для начала давайте настроим самое простое, это выход в интернет для **ISP**. При первом открытии у нас появляется предложение о первоначальной настройке, отказываемся «**N**».

И указываем ip адреса для интерфейсов как в таблице:

**ISP**

|  |
| --- |
| en |
| conf t |
| hostname ISP |
| int et0/0 |
| ip add 100.100.161.1 255.255.255.0 |
| no shutdown |
| int et0/1 |
| ip add 172.16.4.14 255.255.255.240 |
| no shutdown |
| int et0/2 |
| ip add 172.16.5.14 255.255.255.240 |
| no shutdown |
| int et0/3 |
| ip add 172.16.6.14 255.255.255.240 |
| no shutdown |

Далее необходимо указать дефолтный маршрут для вывода всего трафика в интернет.

|  |
| --- |
| ip route 0.0.0.0 0.0.0.0 100.100.161.254 |

Последний ip адрес необходимо указать из вашей подсети и последнего ip.

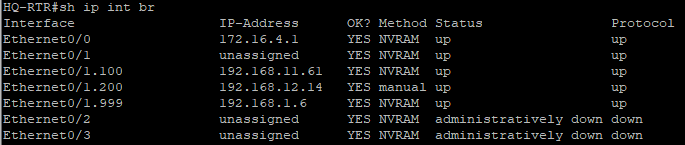
Далее проверяем доступность интернета с помощью ping на 8.8.8.8.

Таким же образом нам необходимо настроить наши маршрутизаторы филиалов.

**HQ-RTR**

|  |
| --- |
| en |
| conf t |
| hostname HQ-RTR |
| int et0/0 |
| ip add 172.16.4.1 255.255.255.240 |
| no shutdown |
| int et0/1 |
| no shutdown |
| int et0/1.100 |
| encapsulation dot1Q 100 |
| ip add 192.168.11.61 255.255.255.192 |
| no shutdown |
| int et0/1.200 |
| encapsulation dot1Q 200 |
| ip add 192.168.12.14 255.255.255.240 |
| no shutdown |
| int et0/1.999 |
| encapsulation dot1Q 999 |
| ip add 192.168.1.6 255.255.255.248 |
| no shutdown |

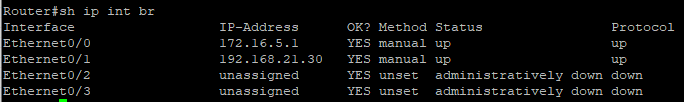
Результат.



**BR-RTR**

|  |
| --- |
| en |
| conf t |
| hostname BR-RTR |
| int et0/0 |
| ip add 172.16.5.1 255.255.255.240 |
| no shutdown |
| int et0/1 |
| ip add 192.168.21.30 255.255.255.224 |
| no shutdown |

Результат.



Далее необходимо сделать настройку на **SW** и указать **VLAN** необходимые.

**SW**

|  |
| --- |
| en |
| conf t |
| hostname SW |
| int et0/0 |
| switchport trunk encapsulation dot1q |
| switchport mode trunk |
| int et0/1 |
| switchport mode access |
| switchport access vlan 100 |
| int et0/2 |
| switchport mode access |
| switchport access vlan 200 |
| interface vlan999 |
| ip add 192.168.1.1 255.255.255.248 |
| no shutdown |
| ip route 0.0.0.0 0.0.0.0 192.168.1.6 |

Также на **SW** еще необходимо выполнить команду на отключение cdp.

|  |
| --- |
| no cdp run |

На этом предварительная настройка завершена, у вас должно быть настроено каждое оборудование и общаться с своим соседом без проблем. Дальнейшая настройка относится точечно к каждому заданию в ДЭ. Поэтому мы будем настраивать ДЭ постепенно, а дальнейшее сокращение времени для действия оставляю за вами.

Далее необходимо донастроить **ISP**, так как он является провайдером, он должен выполнять проброс трафика пользователей и в целом ваших филиалов.

Переходим на **ISP** и выполняем команды включения **NAT**:

|  |
| --- |
| interface et0/0 |
| ip nat outside |
| interface range et0/1-3 |
| ip nat inside |
| exit |
| access-list 1 permit any |
| ip nat inside source list 1 interface ethernet 0/0 overload |

Пояснение:

Переходим в каждый необходимый интерфейс и указываем чем он занимается при использовании **NAT**. Интерфейс для выхода в интернет указываем как outside, все остальные считаются локальными, указываем их как inside.

Создаем список, в который включаем все, так как нет правил, запрещающих это, и указываем этот список в использование **NAT** с интерфейсом в интернет.

Далее нам необходимо настроить **HQ-RTR**, **BR-RTR** на использование NAT.

**BR-RTR**

|  |
| --- |
| interface et0/0 |
| ip nat outside |
| interface et0/1 |
| ip nat inside |
| exit |
| access-list 1 permit any |
| ip nat inside source list 1 interface ethernet 0/0 overload |
| ip route 0.0.0.0 0.0.0.0 172.16.5.14 |

**HQ-RTR**

|  |
| --- |
| interface et0/0 |
| ip nat outside |
| interface et0/1 |
| ip nat inside |
| interface et0/1.100 |
| ip nat inside |
| interface et0/1.200 |
| ip nat inside |
| interface et0/1.999 |
| ip nat inside |
| exit |
| access-list 1 permit any |
| ip nat inside source list 1 interface ethernet 0/0 overload |
| ip route 0.0.0.0 0.0.0.0 172.16.4.14 |

Самое время сохранить конфигурации на сетевом оборудовании.

Написав эти команды, у вас должен начать работать интернет, на всех устройствах какие только есть. Выполните **ping** на сайт **ya.ru**, или другой.

Если на этом моменте у вас не работает один из устройств, необходимо определить в чем проблема и исправить.

Далее займемся простым созданием пользователей, первое задание гласит, что необходимо создать пользователя **sshuser** на серверах **HQ-SRV**, **BR-SRV**.

В дополнительных пунктах написаны условия создания, будем конечно их соблюдать.

Переходим к **HQ-SRV**.

И выполняем команды создания пользователя:

|  |
| --- |
| useradd –m –u 1010 –s /bin/bash sshuser |

-m — создаст домашний каталог для пользователя.

-u 1010 — установит идентификатор пользователя (UID) на 1010.

-s /bin/bash — задаст Bash как основной shell.

Установим пароль для пользователя:

|  |
| --- |
| passwd sshuser |

Далее вводим пароль **P@ssw0rd** и подтверждаем его.

Далее нам необходимо, что бы пользователь мог использовать **sudo** без аутентификации дополнительной.

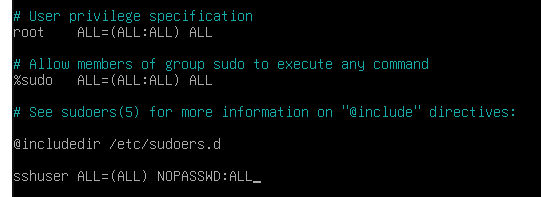
Но, как оказалось, стандартно в этой сборке **Debian** нет **sudo**, поэтому нам необходимо его установить. Выполним команду:

|  |
| --- |
| apt update && apt install sudo |

Далее необходимо зайти в файл конфигурации **sudoers** с помощью команды:

|  |
| --- |
| visudo |

И добавить в конец файла строку разрешения для пользователя **sshuser**:



Далее необходимо проверить работоспособность настройки, для этого необходимо выполнить команду для перехода под пользователя **sshuser**:

|  |
| --- |
| su sshuser |

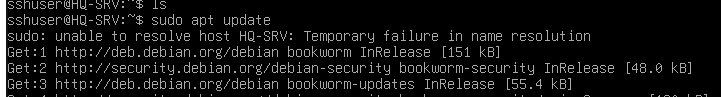
Текст представления должен поменяться с **root** на **sshuser**.



Выполняем команду обновления, что бы удостовериться в настройке:

|  |
| --- |
| sudo apt update |

Результат:



Он сразу же начал выполнять, так как теперь не спрашивает пароль.

Отлично! Осталось выполнить данную настройку на **BR-SRV**.

НЕ ЗАБУДЬТЕ ПРОВЕРИТЬ РАБОТОСПОСОБНОСТЬ ПАРОЛЯ ПОЛЬЗОВАТЕЛЯ. ВОЗМОЖНО ВЫ ЗАДАЛИ ЕГО НЕПРАВИЛЬНО.

Для проверки выйдите в меню ввода логина/пароля.

Пишите exit пока не понадобится ввести логин/пароль.

После вводите логин/пароль (**sshuser/P@ssw0rd**).

Далее нам необходимо создать пользователя на маршрутизаторах **RTR**:

|  |
| --- |
| username net\_admin privilege 15 secret P@$$word |

Следующим пунктом нам необходимо настроить **GRE** и шифрование передачи данных это сразу закроет 2 задания. Для этого нам необходимо понимать, что **VPN** состоит из 2 протоколов для настройки, из протокола туннелирования, в нашем случае это будет **GRE**, а также протокола шифрования, у нас будет использоваться **IPsec**, но также в cisco мы будем использовать и помогающий сервис под названием **ISAKMP**, который используется для создания и управления безопасными ассоциациями (**SA**) между узлами в сетях, использующих **IPsec**. Он управляет обменом ключами, аутентификацией и настройкой параметров для установления защищенных соединений.

Для начала настроим протокол туннелирования, для этого необходимо выполнить команды:

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| int tun0 | | Создание тунеля |
| ip add 192.168.254.1 255.255.255.252 | ip add 192.168.254.2 255.255.255.252 | Указание ip |
| tunnel source et0/0 | | Интерфейс в сторону где искать |
| tunnel destination 172.16.5.1 | tunnel destination 172.16.4.1 | ip соседа для соединения |

Далее нам необходимо создать access-list который будет использоваться для того, чтобы убедиться, что только трафик **GRE** от 172.16.4.1 к 172.16.5.1 проходит через маршрутизатор, что может быть полезно для настройки VPN-соединений. Если другие пакеты, не соответствующие этому критерию, будут пытаться пройти через маршрутизатор, они будут заблокированы, что повышает уровень безопасности и управления трафиком.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| ip access-list extended GRE-TRAFFIC | | Создание списка |
| permit gre host 172.16.4.1 host 172.16.5.1 | permit gre host 172.16.5.1 host 172.16.4.1 | Разрешение отправки gre |

Далее нам необходимо создать политику работы **ISAKMP**.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| crypto isakmp policy 10 | | Создание политики |
| encr aes 256 | | Метод шифрования |
| hash sha512 | | Метод хеширования |
| authentication pre-share | | Метод аутентификации |
| group 24 | | Уровень групповой аутентификации |

Далее нам необходимо настроить ключ, раз мы указали **pre-share**, а также ключ для какого адреса.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| crypto isakmp key cisco123 address 172.16.5.1 | crypto isakmp key cisco123 address 172.16.4.1 | Ключ и адресат |

Далее мы сделаем настройку **isakmp keepalive**, которая отвечает за поддержание соединения.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| crypto isakmp keepalive 10 5 | | 10 минут  5 попыток |

Далее нам необходимо настроить набор преобразований в контексте **IPsec**.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| crypto ipsec transform-set GRE-TRANSFORM esp-aes esp-sha-hmac | | Основная команда, шфрование aes, хеширование sha-hmac |
| mode tunnel | | Работа в режиме тунеля |

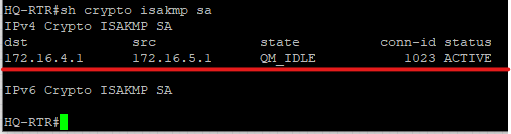
Далее нам необходимо собрать все это в единую криптографическую карту.

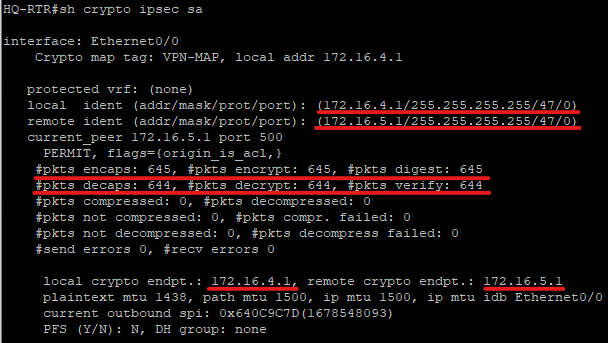
|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| crypto map VPN-MAP 10 ipsec-isakmp | | Создание карты |
| set peer 172.16.5.1 | set peer 172.16.4.1 | Удаленный пиринг |
| set transform-set GRE-TRANSFORM | | Включение набора преобразований |
| match address GRE-TRAFFIC | | Применение карты на указанном трафике |

Далее нам осталось дело за малым, а именно назначить карту на интерфейс.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| int et0/0 | | Заходим в интерфейс |
| crypto map VPN-MAP | | Назначаем карту |

Далее нам остается только проверить работоспособность построенной структуры. Для этого мы выполним команду и все что подчеркнуто красным должно быть не 0 и не пусто.





Когда мы реализовали связь, осталось сделать маршрутизацию для полной связанности, для этого мы будем использовать протокол **OSPF** из-за простоты настройки.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| router ospf 1 | | Заходим в ospf |
| network 192.168.1.0 0.0.0.7 area 0 | network 192.168.21.0 0.0.0.31 area 0 | Указываем необходимы сети для маршрутизации между подсетями |
| network 192.168.11.0 0.0.0.63 area 0 | network 192.168.254.0 0.0.0.3 area 0 |
| network 192.168.12.0 0.0.0.15 area 0 |  |
| network 192.168.254.0 0.0.0.3 area 0 |  |

Далее у нас есть условия, которые мы должны соблюсти, а именно:

* установка парольной защиты;
* работа только через интерфейс туннеля.

|  |  |  |
| --- | --- | --- |
| HQ-RTR | BR-RTR | Описание |
| passive-interface default | | Стандартная блокировка всех общений |
| no passive-interface Tunnel0 | | Исключение для tun0 |
| int tun0 | | Переходим на интерфейс для указания пароля |
| ip ospf authentication | | Указываем, что нужно аутентификация |
| ip ospf authentication-key cisco123 | | Указываем пароль для аутентификации |

При настройке мы не раз будем терять общение маршрутизации, но это нормально. В результате мы должны получить нормальную таблицу маршрутизации со всеми маршрутами в cisco, а также доступность между устройствами в филиалах.

Далее нам необходимо настроить протокол **SSH** на **HQ-SRV**, **BR-SRV**.

Переходим к **HQ-SRV**, нам необходимо установить **openssh-server**.

Выполняем команду:

|  |
| --- |
| apt install openssh-server |

Далее нам нужно сконфигурировать файл **SSH** сервера, который находится по пути «**/etc/ssh/sshd\_config**».

В этом файле нам нужно найти строку **Port №** и изменить ее на **Port 2024**, а также следом добавить строку «**AllowUsers sshuser**», а также раскомментировать.



Далее нам необходимо ограничить количество попыток, для этого нужно найти строку **MaxAuthTries** 2.

Для настройки баннера, нам необходимо создать отдельный файл и привязать его в файле. Заходим в файл «**nano /etc/issue.net**».

Добавляем строку в конец файла «**Authorized access only**».

Заходим обратно в файл «**/etc/ssh/sshd\_config**». Необходимо найти строку **Banner none**, раскомментировать и указать путь к баннеру.



Осталось только перезапустить **SSH** сервис.

|  |
| --- |
| systemctl restart sshd |

Делаем тоже самое и на **BR-SRV**.

Для проверки подключимся через **HQ-CLI** к **HQ-SRV**, **BR-SRV**.

|  |
| --- |
| ssh –p 2024 sshuser@192.168.11.1 |
| ssh –p 2024 sshuser@192.168.21.1 |

Далее нам предстоит настроить базовый сервис ДЭ, **DNS**.

Для этого нам необходимо установить пакет **bind9** на ВМ **HQ-SRV**:

|  |
| --- |
| apt install bind9 dnsutils |

Далее нам необходимо настроить файл **/etc/bind/named.conf.options**

Необходимо раскомментировать строку **forwarders** и указать адрес 8.8.8.8

Необходимо дописать две строки:

**recursion yes;**

**allow-query {any;};**

Изменить строку **listen-on-v6 {any;};** на **listen-on {any;};**



Далее необходимо настроить файл **/etc/bind/named.conf.default-zones**

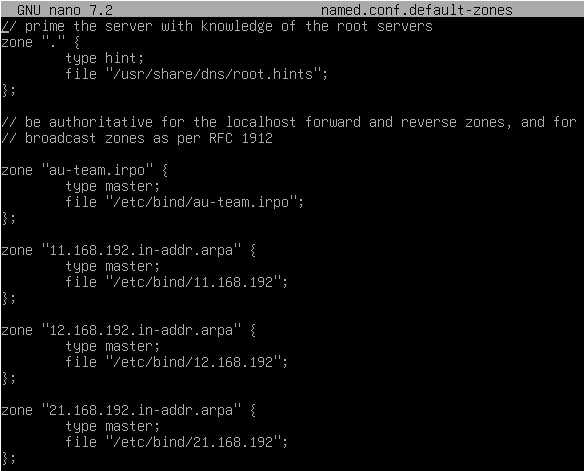
|  |
| --- |
| nano /etc/bind/named.conf.default-zones |

Далее вариативная настройка, следуя из названий зон и записей, вы должны изменять config.

Зона **au-team.irpo**, также не забудьте настроить обратную зону.

|  |  |  |
| --- | --- | --- |
| Устройство | Запись | Тип |
| HQ-RTR | hq-rtr.au-team.irpo | A,PTR |
| BR-RTR | br-rtr.au-team.irpo | A |
| HQ-SRV | hq-srv.au-team.irpo | A,PTR |
| HQ-CLI | hq-cli.au-team.irpo | A,PTR |
| BR-SRV | br-srv.au-team.irpo | A |
| HQ-RTR | moodle.au-team.irpo | CNAME |
| HQ-RTR | wiki.au-team.irpo | CNAME |

К такому виду вам необходимо привести его:



Данной конструкцией происходит создание зоны «**au-team.irpo**».

В ее параметрах указывается тип (type master) зоны, файл зоны (file).

|  |
| --- |
| zone “au-team.irpo” { |
| type master; |
| file “/etc/bind/au-team.irpo”; |
| }; |

Здесь создается обратная зона для обслуживаемых подсетей.

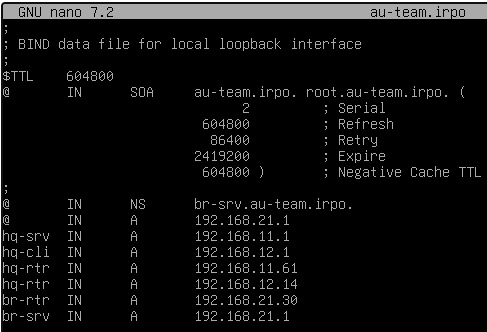
|  |
| --- |
| zone “11.168.192.in-addr.arpa” { |
| type master; |
| file “/etc/bind/11.168.192”; |
| }; |
| zone “12.168.192.in-addr.arpa” { |
| type master; |
| file “/etc/bind/12.168.192”; |
| }; |
| zone “11.168.192.in-addr.arpa” { |
| type master; |
| file “/etc/bind/21.168.192”; |
| }; |

Следующим пунктом нужно создать зоны, которые были добавлены.

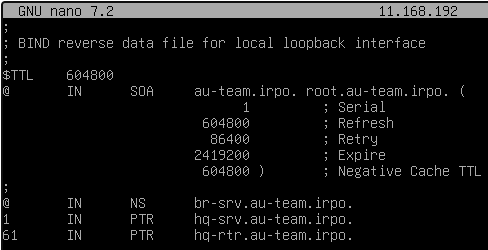
Скопировать пример прямой зоны и обратной зоны:

|  |
| --- |
| cd /etc/bind |
| cp db.local au-team.irpo |
| cp db.127 11.168.192 |
| cp db.127 12.168.192 |
| cp db.127 21.168.192 |

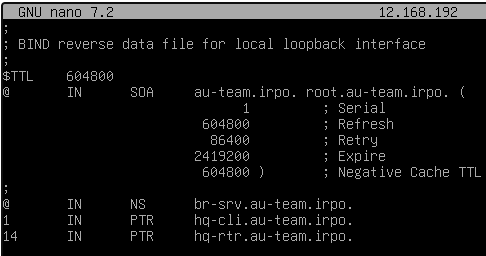
Настройка зоны **au-team.irpo**



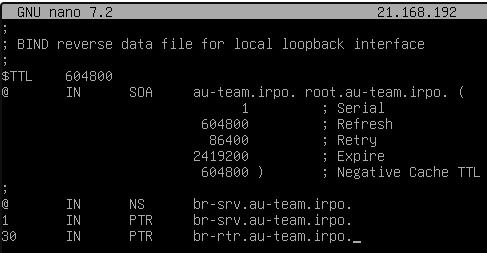
Настройка обратной зоны **11.168.192**



Настройка обратной зоны **12.168.192**



Настройка обратной зоны **21.168.192**

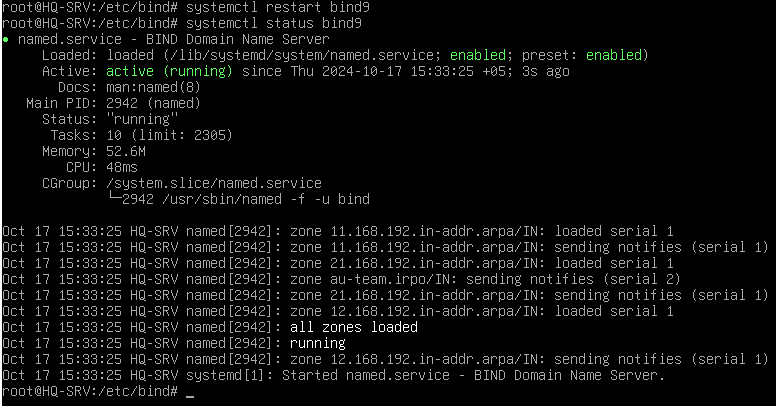


Далее необходимо перезагрузить **bind9:**

|  |
| --- |
| systemctl restart bind9 |

Должно получить вот так:

|  |
| --- |
| systemctl status bind9 |

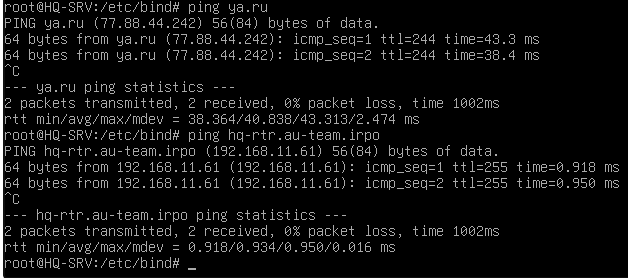


Отлично, теперь нам необходимо на всех виртуальных машинах **Debian**, измениться файл **/etc/resolv.conf**. Он должен стать таким:



Кроме **CLI**, остается прежним.

Далее мы тестируем доступность через имена **DNS**.



Далее нам необходимо настроить протокол динамической конфигурации хостов и действовать у нас он будет только на **HQ-CLI**.

Он должен быть установлен на **HQ-RTR**. Выполняем команды:

|  |
| --- |
| ip dhcp pool HQ-CLI-POOL |
| network 192.168.12.0 255.255.255.240 |
| default-router 192.168.12.14 |
| dns-server 192.168.11.1 |
| domain-name au-team.irpo |

А также необходимо зарезервировать ip адрес для **HQ-RTR**.

|  |
| --- |
| ip dhcp excluded-address 192.168.12.14 |

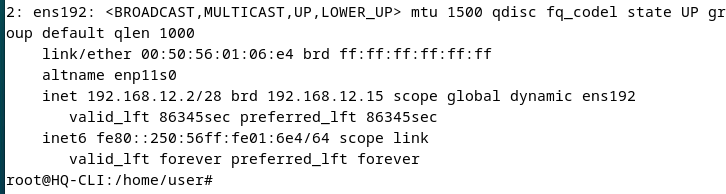
Далее нам необходимо изменить сетевые настройки **HQ-CLI**, переходим в файл настроек **/etc/network/interfaces**. Должно получиться вот так.



Перезагружаем **HQ-CLI** и выполняем команду получения интерфейсов:

|  |
| --- |
| ip a |

Видим такой результат.



Дальнейшее задание гласит, что необходимо изменить часовой пояс на всех устройствах, давайте начнем с виртуальных машин. Для изменения нужно выполнить команду:

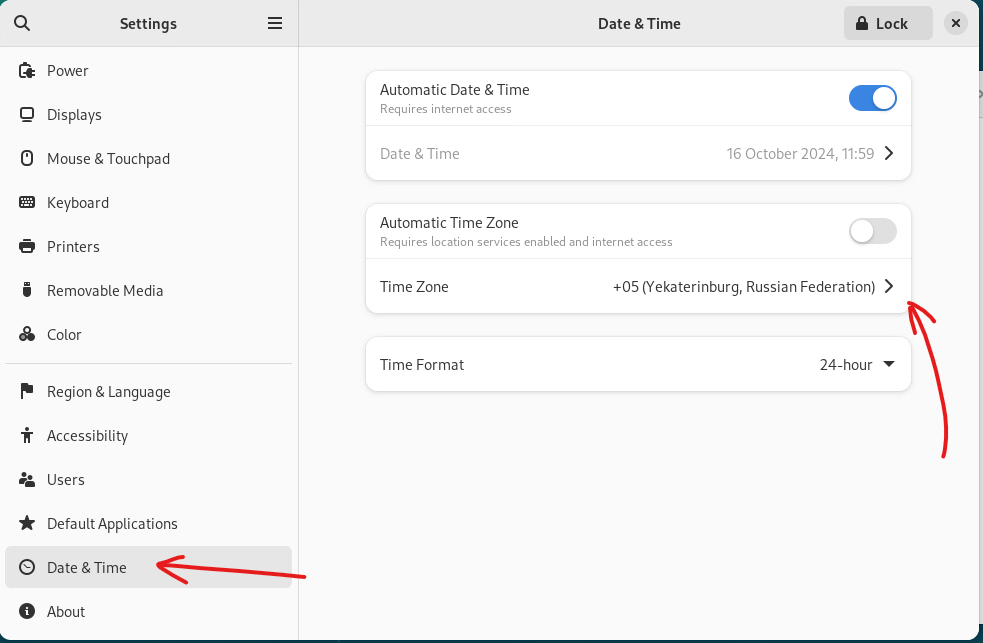
|  |
| --- |
| dpkg-reconfigure tzdata |

Далее необходимо выбрать **Asia**, и **Yekaterinburg**.

Появится сообщение о текущем времени, оно должно совпадать с нашим:



Для настройки **HQ-CLI**, **CLI**, необходимо зайти в настройки виртуальной машины и найти в низу пункт **Date & Time**. Далее в строке поиска найти **Yekaterinburg**.



Для настройки временной зоны на сетевом оборудовании необходимо выполнить команду:

|  |
| --- |
| clock timezone +05 5 0 |

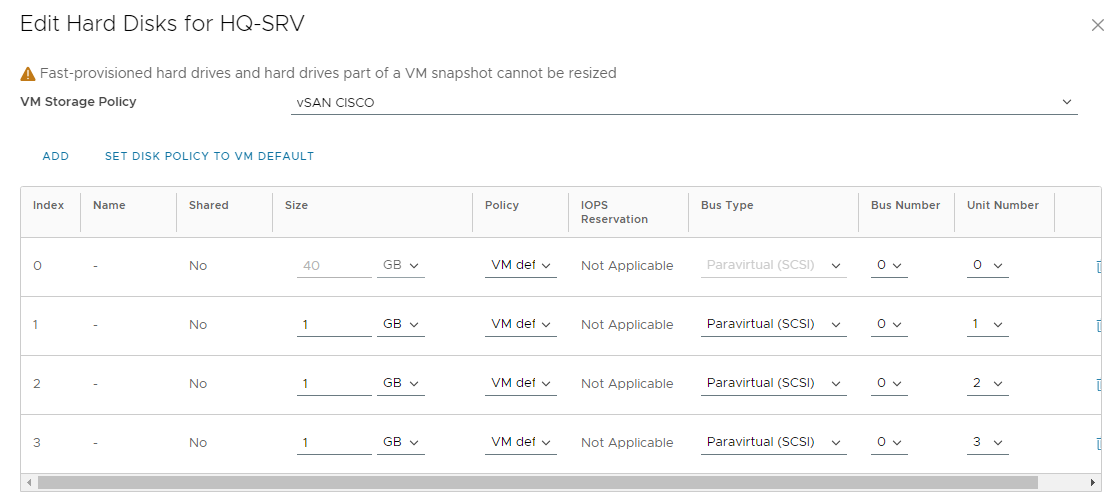
Обычно у нас она уже настроена на оборудовании.

На этом моменте мы начинаем делать задания из второго модуля ДЭ, мы полностью сделали задание первого модуля и затронули некоторые из третьего.

В данном модуле нам необходимо по максимуме сосредоточиться на виртуальных машинах и их сервисов, первым на очереди в выполнение задания я ставлю сборку **RAID 5** на **HQ-SRV**, что потребует перезапуска и последующей настройке, задания далее будут приставлять большую сложность и трудозатратность.

Для настройки **RAID 5** нам необходимо выключить виртуальную машину **HQ-SRV** и добавить ей диски, переходим в **Detail** и в навигационной панели находим пункт **Hard Disks**. В этой вкладке вы можете увидеть 1 стандартный диск, нам необходимо добавить еще 3.

Нажимаем **Edit**. И нажимаем 3 раза **ADD**, что позволяет нам добавить виртуальные диски в сервер, размер дисков может быть изменен заданием, поэтому будьте осторожны, если нужно изменить размер измените.



Сохраняем, запускаем и переходим в виртуальную машину.

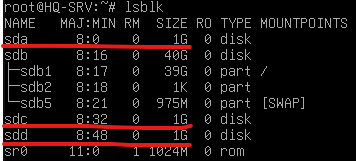
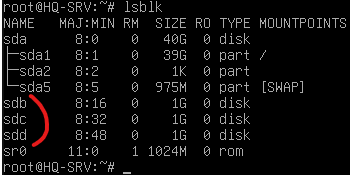
Для начала вам необходимо установить пакет **mdadm** на **HQ-SRV**

|  |
| --- |
| apt install mdadm |

Далее необходимо определить идентификаторы для неформатированных дисков, которые вы будете использовать

|  |
| --- |
| lsblk |

Диски должны быть по 1 GB:



Далее необходимо создать массив дисков:

|  |
| --- |
| mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd |

И сразу же нам необходимо добавить конфигурацию сборки массива, что бы он собирался при каждом запуске **HQ-SRV** автоматически:

|  |
| --- |
| mdadm --detail --scan | tee -a /etc/mdadm/mdadm.conf |

Далее нам необходимо обновить конфигурацию системы по массиву:

|  |
| --- |
| update-initramfs -u |

Далее необходимо отформатировать полученную структуру:

|  |
| --- |
| mkfs.ext4 -F /dev/md0 |

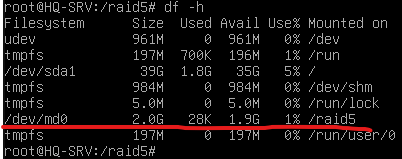
Создаем точку монтирования и подключаем:

|  |
| --- |
| mkdir /raid5 |
| mount /dev/md0 /raid5 |

Далее необходимо добавить строку в файл **/etc/fstab**, для автоматической сборки **RAID** после перезагрузки:

|  |
| --- |
| /dev/md0 /raid5 ext4 defaults 0 0 |

Для проверки подключения выполните перезагрузку, а потом **df -h**.



Далее нам необходимо развернуть **NFS** сервис для публикации папки из **RAID** на **HQ-CLI**, далее устанавливаем пакет для развертки **NFS**.

|  |
| --- |
| apt install nfs-kernel-server |

Далее, конечно, создаем отдельную директорию для общего доступа, так как ограничение будет сделано на **NFS** настройке по доступной подсети.

|  |
| --- |
| mkdir /raid5/nfs |
| chown nobody:nogroup /raid5/nfs |
| chmod 777 /raid5/nfs |

Теперь необходимо отредактировать файл настроек **NFS** **/etc/exports** и добавить туда строку в нашу локальную сеть с **HQ-CLI**.

|  |
| --- |
| /raid5/nfs 192.168.12.0/28(rw,sync,no\_subtree\_check) |

Далее нужно применить изменения:

|  |
| --- |
| exportfs -ra |

Далее нужно перезапустить сервис **NFS**:

|  |
| --- |
| systemctl restart nfs-kernel-server |

Для проверки выполняем команду **systemctl status nfs-kernel-server**:



Далее переходим в **HQ-CLI** и устанавливаем пакет **nfs-common**:

|  |
| --- |
| apt install nfs-common |

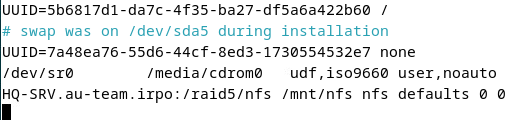
Создаем папку для монтирования, что бы подключать туда папку:

|  |
| --- |
| mkdir /mnt/nfs |

Далее делаем подключение через **/etc/fstab**, добавляем строку:

|  |
| --- |
| HQ-SRV.AU-TEAM.IRPO:/raid5/nfs /mnt/nfs nfs defaults 0 0 |

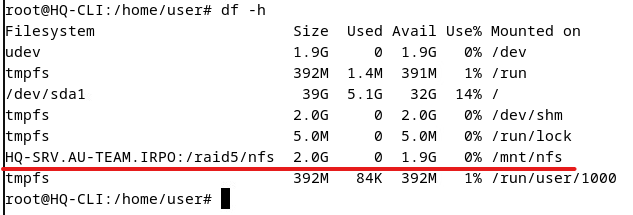
Пример:



Далее нам необходимо перезапустить виртуальную машину, так проще.

И выполнить команду вывода всех разделов и дисков **df –h**.

Видим наше подключение:



Давайте далее настроим службу сетевого времени на базе сервиса **chrony**.

В задании написано, что мы должны установить **chrony** на **HQ-RTR**, но у нас не получится сделать это, так как это **Cisco Router**, мы установим его на **HQ-SRV**.

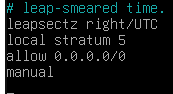
Для этого необходимо перейти в **HQ-SRV** и установить пакет:

|  |
| --- |
| apt install chrony |

Далее нам необходимо изменить файл конфигурации сервера **/etc/chrony/chrony .conf**

В конце файла добавить пару строк:

|  |
| --- |
| local stratum 5 |
| allow 0.0.0.0/0 |
| manual |

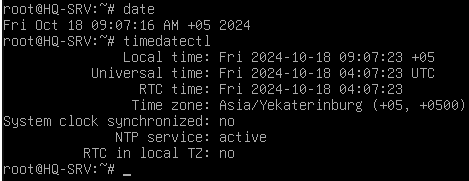


В **Stratum** указываете то число, которое стоит в задании.

Далее необходимо перезапустить **chrony**.

|  |
| --- |
| systemctl restart chrony |

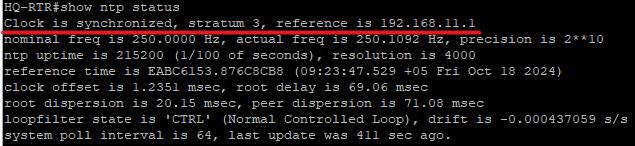
Пара проверок, что дата и время у вас совпадает с настоящей:



Далее необходимо настроить клиентов, давайте начнем с сетевого оборудования, оно проще всего настраивается, переходим на **HQ-RTR**, **BR-RTR** и выполняем команды для указания **NTP** сервера (**HQ-SRV**):

|  |
| --- |
| ntp server 192.168.11.1 |

Далее для проверки выполняем команду **show ntp status**:



Если у вас не показывает **synchronized**, то возможно нужно больше времени для синхронизации, мы позже посмотрим на сервере подключены они или нет.

То, что сейчас указан **stratum 3** ничего страшного, локальный мы указали как **5**, это означает, что сервер будет **stratum 5** только если пропадет выход в интернет.

Далее выполняем настройку на **BR-SRV**, **HQ-CLI**.

Также устанавливаем пакет и заменяем север **NTP** на наш.

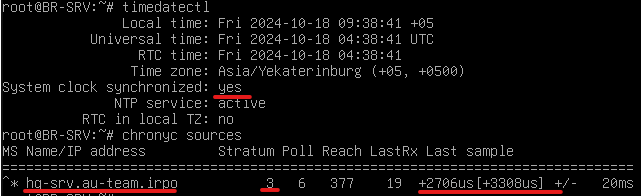
|  |
| --- |
| apt install chrony |
| nano /etc/chrony/chrony.conf |
| server 192.168.11.1 iburst |



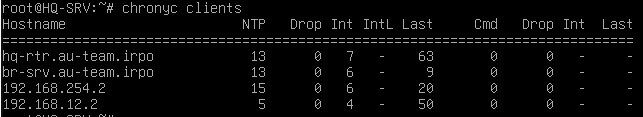
Перезапускаем сервис:

|  |
| --- |
| systemctl restart chrony |

Проверяем:



Проверяем на **HQ-SRV** все ли подключены:

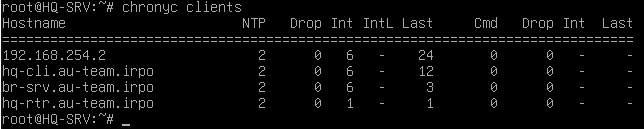


Первые 2 понятно кто такие, а вот последние давайте разберем.

192.168.254.2 – это **BR-RTR**, он воспользовался ip адресом **GRE** для подключения, но он у нас не добавлен в DNS, поэтому просто ip адрес.

192.168.12.2 – это у нас **HQ-CLI**, мы не изменили его в **DNS,** когда он получил DHCP, поэтому нам нужно поменять ip в **DNS**, это файл **au-team.irpo** и **12.168.192**.

Далее перезагружаем **bind** и снова выводим данные:

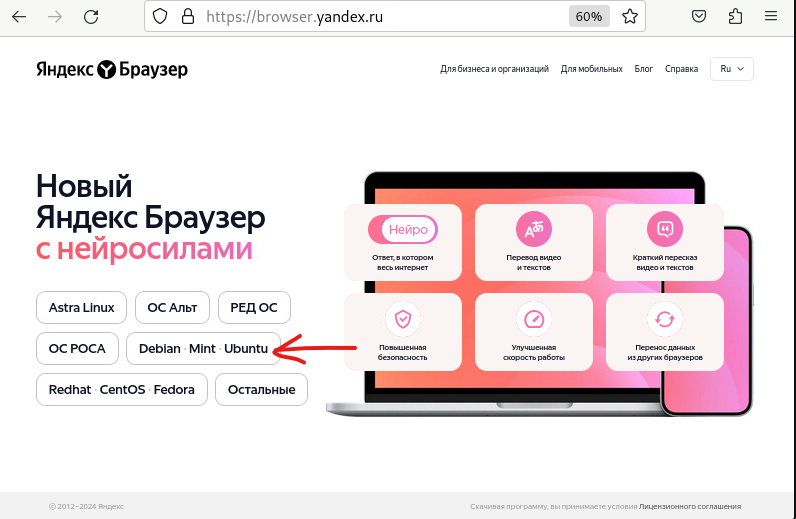


Теперь мы видим, что все подключены и все работает.

Немного отвлечемся и установим Яндекс Браузер на **HQ-CLI**.

Переходим в браузер и вводим **install yandex browser**.

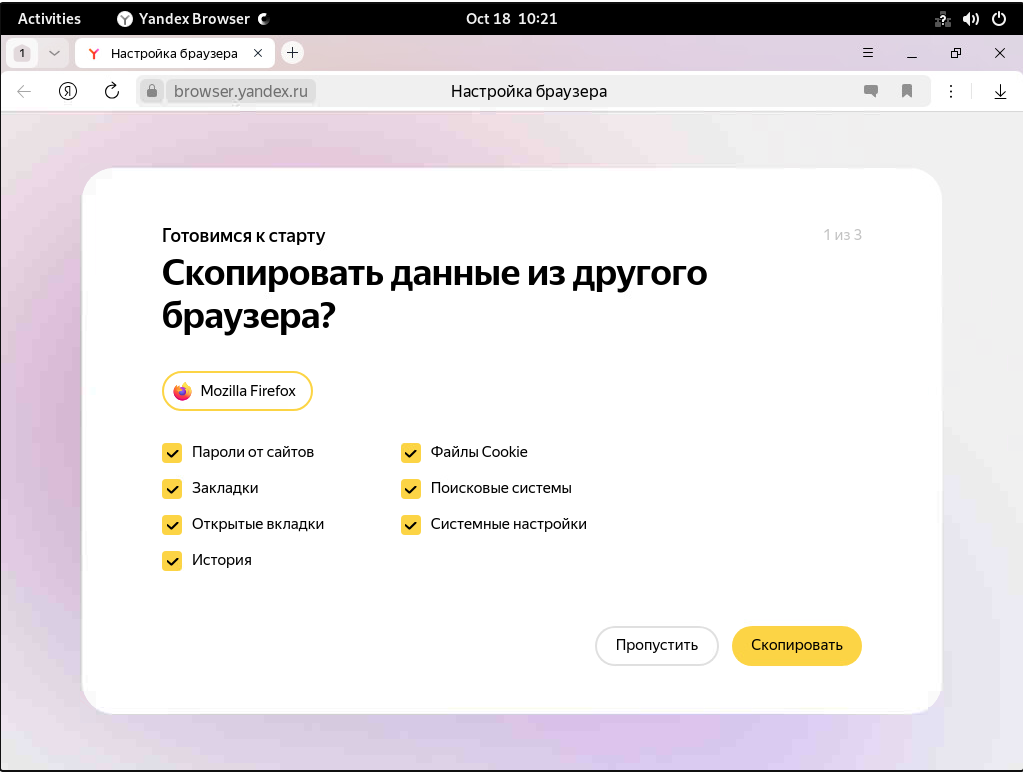
Открываем первую ссылку и в ней выбираем **Debian**.



Нажимаем скачать **deb** пакет.

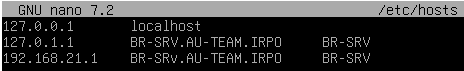
Переходим в терминал и выполняем команды:

|  |
| --- |
| cd /home/user/Downloads |
| sudo dpkg –i Yandex.deb |
| apt install yandex-browser-stable |
| apt --fix-broken install |



Далее создадим доменный контроллер на базе **Samba** на машине **BR-SRV**.

Для дальнейшей настройки нужно изменить файл **/etc/hosts**:



Далее переходим в **BR-SRV** и устанавливаем пакеты:

|  |
| --- |
| apt install samba winbind libpam-winbind libnss-winbind libpam-krb5 krb5-config krb5-user krb5-kdc |

ПРИ УСТАНОВКЕ НЕОБХОДИМО БУДЕТ ВВЕСТИ ДАННЫЕ.

Указываем по порядку:

|  |
| --- |
| AU-TEAM.IRPO |
| BR-SRV.AU-TEAM.IRPO |
|  |

В 3 окне ничего не указываем.

Далее нам необходимо выполнить выключение сервисов и их маскировку:

|  |
| --- |
| systemctl stop winbind smbd nmbd krb5-kdc |
| systemctl mask winbind smbd nmbd krb5-kdc |

После этого удаляем стандартный файл **samba /etc/samba/smb.conf**.

|  |
| --- |
| rm /etc/samba/smb.conf |

И выполняем команду создания домена:

|  |
| --- |
| samba-tool domain provision |

Далее необходимо ввести поля для домена, вводим:

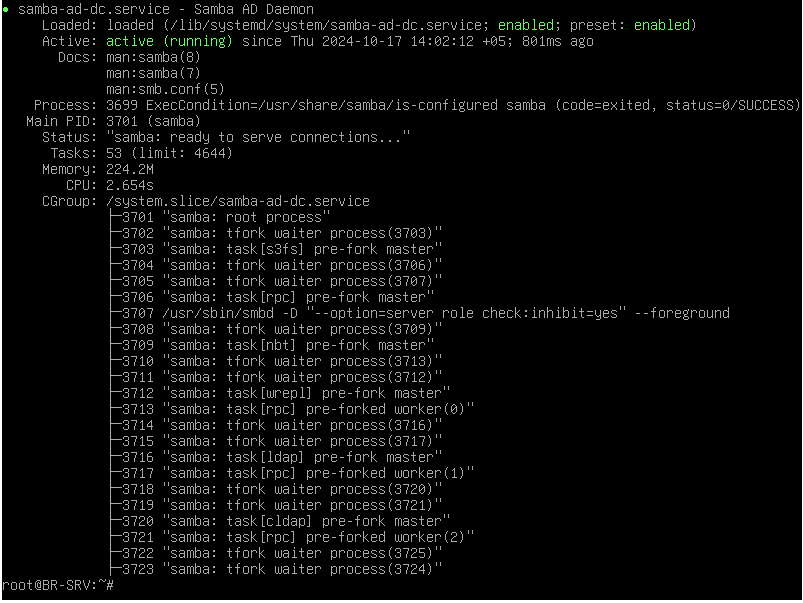
|  |
| --- |
| Realm [AU-TEAM.IRPO] : “Enter” |
| Domain [AU-TEAM] : “Enter” |
| Server Role [dc] : “Enter” |
| DNS backend [SAMBA\_INTERNAL] : “Enter” |
| DNS forwarder IP address [192.168.11.1] : “Enter” |

Если у вас не совпадают данные в квадратных скобках, то нужно их написать, а не просто нажать Enter;

Далее нам необходимо перезапустить сервис **samba-ad-dc**.

|  |
| --- |
| systemctl restart samba-ad-dc |

Пример:



Далее мы переходим к настройке **HQ-CLI**, чтобы добавить его в домен.

Переходим к установке пакетов:

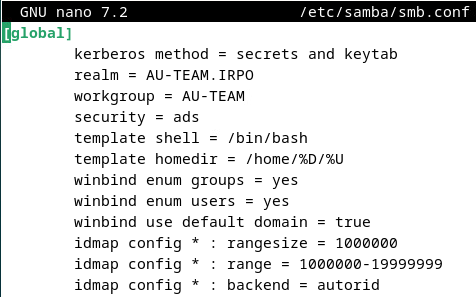
|  |
| --- |
| apt install winbind libpam-winbind libnss-winbind  krb5-config samba-dsdb-modules samba-vfs-modules |

У нас снова появится псевдонастройка в которой мы тоже самое должны ввести, как на **BR-SRV**.

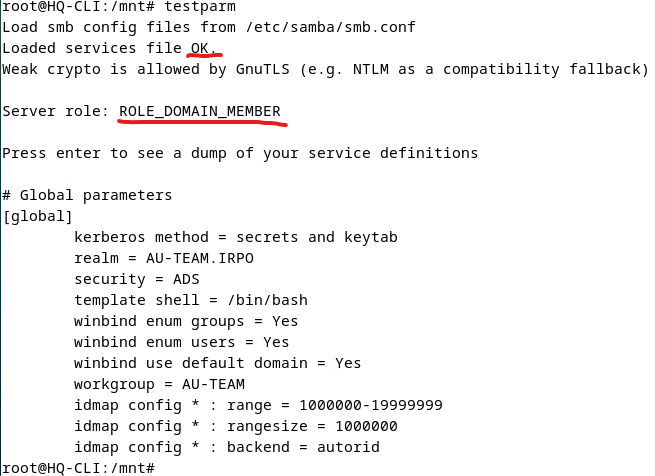
Далее мы переносим файл, на всякий случай:

|  |
| --- |
| mv -v /etc/samba/smb.conf /etc/samba/smb.conf.org |
| nano /etc/samba/smb.conf |

И составляем файл:



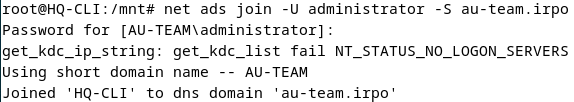
Далее смотрим, что бы файл был без ошибок через команду:

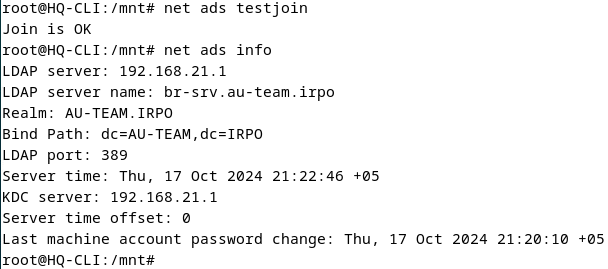


Далее выполняем команду добавления HQ-CLI в домен:

|  |
| --- |
| net ads join -U administrator -S au-team.irpo |

Пример нормального добавления:

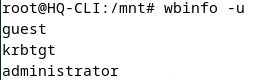




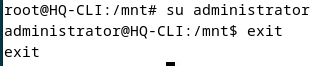
Далее перезапускаем **winbind** для загрузки новых доменных пользователей:

|  |
| --- |
| systemctl restart winbind |

Проверяем:



Пробуем зайти под доменного пользователя администратор:



Отлично, работает!!!!!!!

Далее по заданию нам необходимо создать группу **HQ** и создать в ней 5 пользователей. Но для начала давайте создадим нам пользователя администратора:

|  |
| --- |
| samba-tool user add admin |
| samba-tool group addmembers ‘Domain Admins’ admin |

Далее создаем группу:

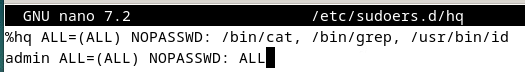
|  |
| --- |
| samba-tool group add hq |

Далее необходимо добавить пользователей в эту группу:

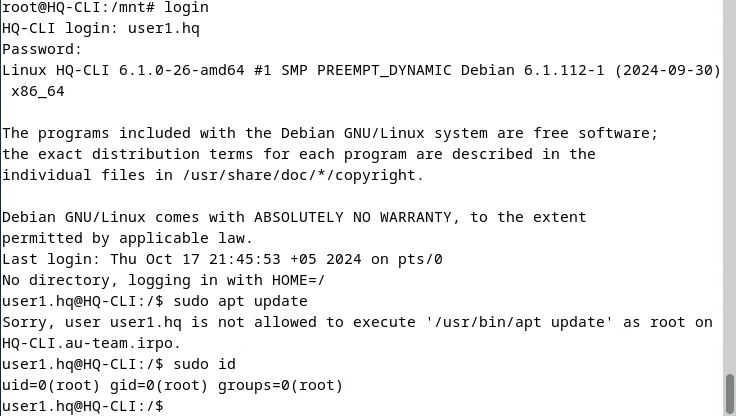
|  |
| --- |
| samba-tool user add user1.hq P@ssw0rd |
| samba-tool group addmembers ‘hq’ user1.hq |

И так 5 раз с разными цифрами.

Далее обратно возвращаемся к нашему **HQ-CLI** и заходим в файл **/etc/sudoers.d/hq**, добавляем пару строк правил:



И пробуем зайти на пользователя **user1.hq** и выполнить неразрешенное действие, а потом разрешенное:



Далее нам необходимо выполнить несложную работу, это проброс портов на маршрутизаторах **HQ-RTR**, **BR-RTR**.

В общей сложности нам необходимо пробросить 3 порта:

80 в 8080 на **BR-SRV** через **BR-RTR**;

2024 на **BR-SRV** через **BR-RTR**;

2024 на **HQ-SRV** через **HQ-RTR**.

Давайте начнем с **HQ-RTR** для этого выполняем команду:

|  |
| --- |
| ip nat inside source static tcp 192.168.11.1 2024 interface Ethernet0/0 2024 |

Переходим к **BR-RTR**, выполняем команды:

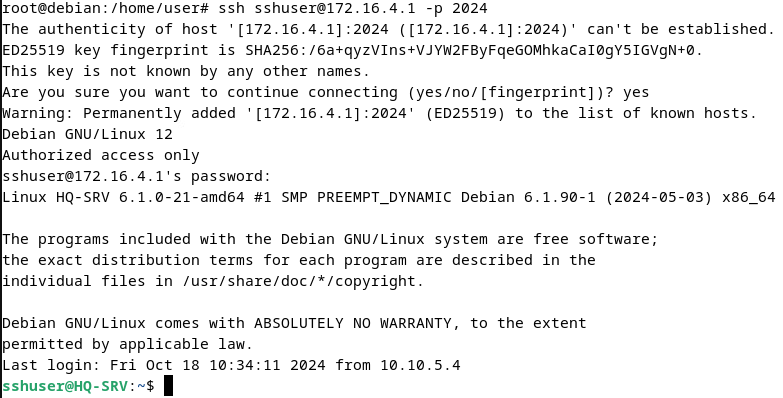
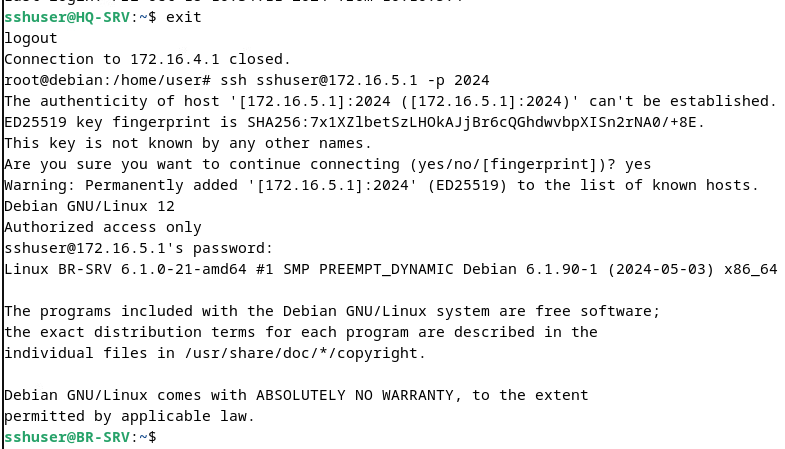
|  |
| --- |
| ip nat inside source static tcp 192.168.21.1 8080 interface Ethernet0/0 80 |
| ip nat inside source static tcp 192.168.21.1 2024 interface Ethernet0/0 2024 |

Далее для проверки переходим на виртуальную машину **CLI**.

Нам необходимо проверить доступность прокинутых портов, но в данный момент мы сможем проверить только **SSH**, это порт **2024**, который мы прокидывали на **HQ-SRV**, **BR-SRV**.

Переходим на **CLI** и в терминале пишем команды:

|  |
| --- |
| ssh sshuser@172.16.4.1 –p 2024 |
| ssh sshuser@172.16.5.1 –p 2024 |



Только не забываем, что необходимо отключиться от первого сервера обратно на **CLI**, а потом уже 2-й раз подключаться.

Далее мы установим **ansible** на **BR-SRV**.

Для этого нам необходимо подготовить устройства. Необходимо настроить **SSH** на **HQ-RTR**, **BR-RTR**.

Необходимо выполнить команды:

|  |
| --- |
| ip domain-name AU-TEAM.IRPO |
| crypto key generate rsa modulus 2048 |
| ip ssh version 2 |
| line vty 0 4 |
| transport input ssh |
| login local |

Далее мы делаем установку на **BR-SRV**.

Устанавливаем пакет:

|  |
| --- |
| apt install pip sshpass |
| pip install ansible==10.0.0 ansible-pylibssh --break-system-packages |

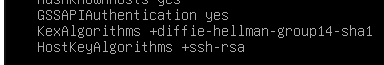
Далее необходимо сгенерировать ключ для подключения и установить для необходимых устройств:

|  |
| --- |
| ssh-keygen –t rsa |
| ssh-copy-id –p 2024 sshuser@HQ-SRV.AU-TEAM.IRPO |
| ssh-copy-id root@HQ-CLI.AU-TEAM.IRPO |

Далее нам необходимо добавить в файл разрешение на подключение к старым протоколам на **HQ-RTR**, **BR-RTR**, заходим в файл **/etc/ssh/ssh\_config**.

И добавить в конец файла строки:

|  |
| --- |
| KexAlgorithms +diffie-hellman-group14-sha1 |
| HostKeyAlgorithms +ssh-rsa |



Далее нам необходимо залогиниться 1 раз на **HQ-RTR**, **BR-RTR** для создания отпечатка подключения:

|  |
| --- |
| ssh net\_admin@HQ-RTR.AU-TEAM.IRPO |
| ssh net\_admin@BR-RTR.AU-TEAM.IRPO |

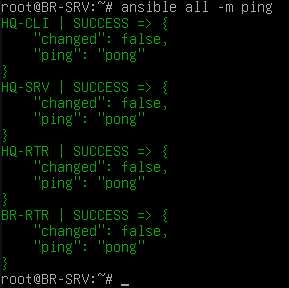
Далее нам необходимо сделать рабочую директорию для **ansible**:

|  |
| --- |
| mkdir /etc/ansible |
| nano /etc/ansible/hosts |

Нам необходимо составить список устройств:

|  |
| --- |
| [HQ] |
| HQ-SRV ansible\_host=HQ-SRV.AU-TEAM.IRPO ansible\_user=sshuser ansible\_port=2024 ansible\_python\_interpreter=/usr/bin/python3 |
| HQ-CLI ansible\_host=HQ-CLI.AU-TEAM.IRPO ansible\_user=root interpreter=/usr/bin/python3 |
| HQ-RTR ansible\_host=HQ-RTR.AU-TEAM.IRPO ansible\_user=net\_admin ansible\_password=P@$$word ansible\_connection=network\_cli ansible\_network\_os=cisco.ios.ios |
| [BR] |
| BR-RTR ansible\_host=BR-RTR.AU-TEAM.IRPO ansible\_user=net\_admin ansible\_password=P@$$word ansible\_connection=network\_cli ansible\_network\_os=cisco.ios.ios |

Далее необходимо выполнить проверку ping до устройств, которые мы добавили, выполняем команду **ansible all –m ping**:



Отлично все работает без предупреждений.

Далее мы займемся установкой **moodle** на вириальную машину **HQ-SRV**.

Для этого нам необходимо как всегда установить необходимые пакеты:

|  |
| --- |
| apt install apache2 |
| apt install php libapache2-mod-php php-mysql php-xml php-gd php-zip php-intl php-curl php-mbstring |
| apt install mariadb-server |

Далее нам необходимо настроить базу данных для использования **moodle**.

|  |
| --- |
| mysql |
| CREATE DATABASE moodledb CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci; |
| CREATE USER 'moodle'@'localhost' IDENTIFIED BY 'P@ssw0rd'; |
| GRANT ALL PRIVILEGES ON moodledb.\* TO 'moodle'@'localhost'; |
| exit; |

Далее нам необходимо скачать сам **moodle**, а также распаковать его:

|  |
| --- |
| cd /var/www/html |
| wget https://packaging.moodle.org/stable405/moodle-latest-405.tgz |
| tar -xvzf moodle-latest-405.tgz |

Далее необходимо создать папку **moodledata** для хранения пользователей и курсов, а также выдать права для пользователя **www-data** на эту папку:

|  |
| --- |
| chown –R www-data:www-data /var/www |
| chmod –R 755 /var/www |

Также добавим адрес в **DNS** на **HQ-SRV** в прямую зону и обратную:





Далее нам необходимо изменить настройки **apache2** для, того, что бы стандартно открывался сервер **moodle**.

Далее нам необходимо установить **max\_input\_vars = 5000** или более, для этого необходимо перейти в файл настроек **php /etc/php/8.2/apache2/php.ini**

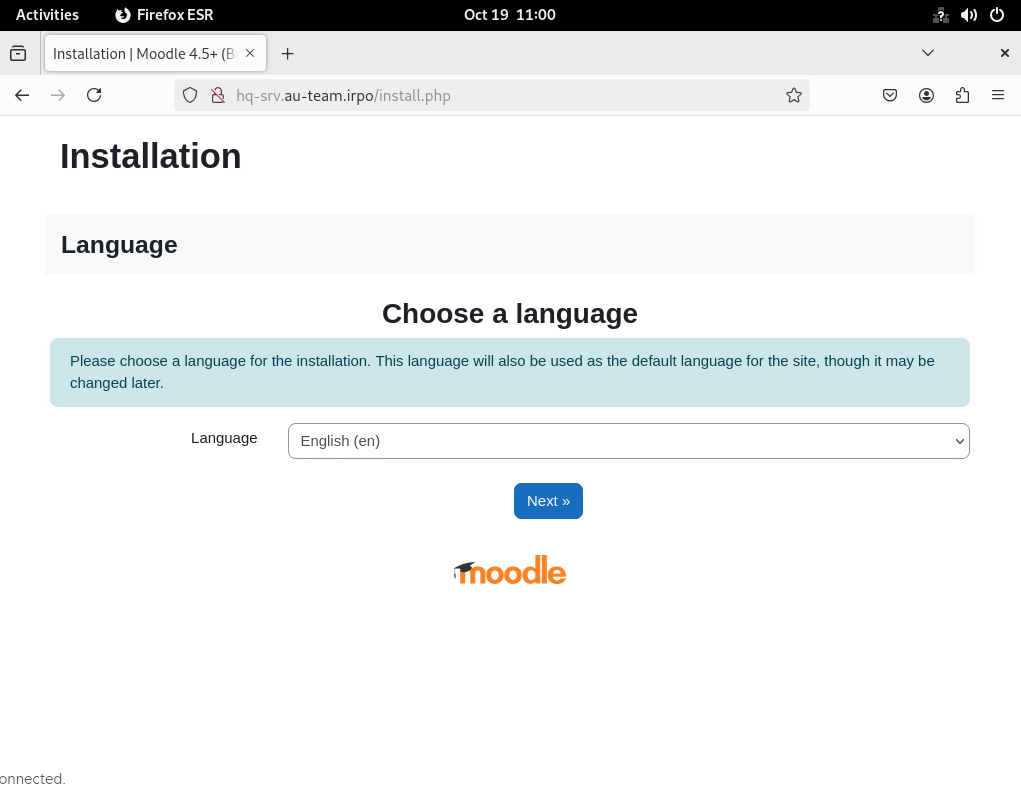
Там довольно большой файл для поиска данной строки, ориентируйтесь на раздел **Resource Limits**, далее настройка чуть ниже, ее необходимо расскоментировать и изменить на 5000.

Перезагрузите сервер **HQ-SRV**.

Отлично, далее нам необходимо зайти на сайт **moodle** и донастроить его, так как такие системы имеют настройку на сайте.

Для этого переходим на **HQ-CLI**, заходим в браузер и переходим по адресу сервера **http://HQ-SRV.AU-TEAM.IRPO/moodle**.

У нас появится окно установки:

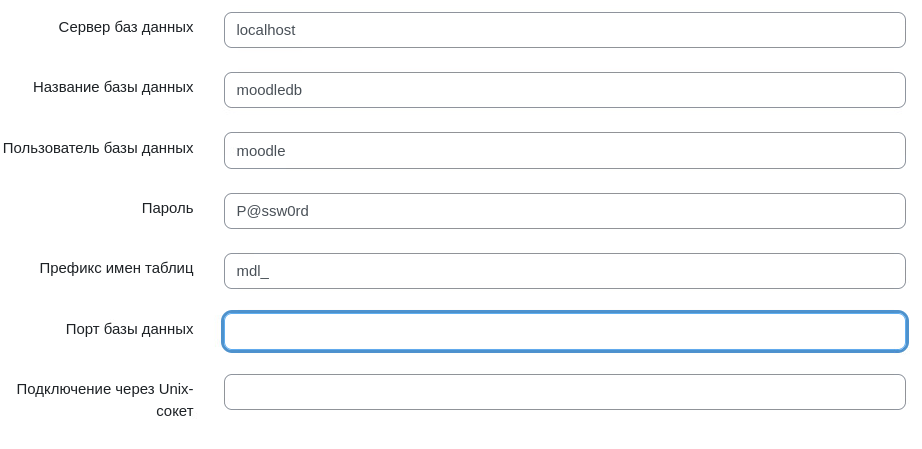


Выбираем Русский язык.

Нажимаем далее при выборе каталогов.

Далее нам необходимо выбрать базу данных, выбираем **MariaDB**.

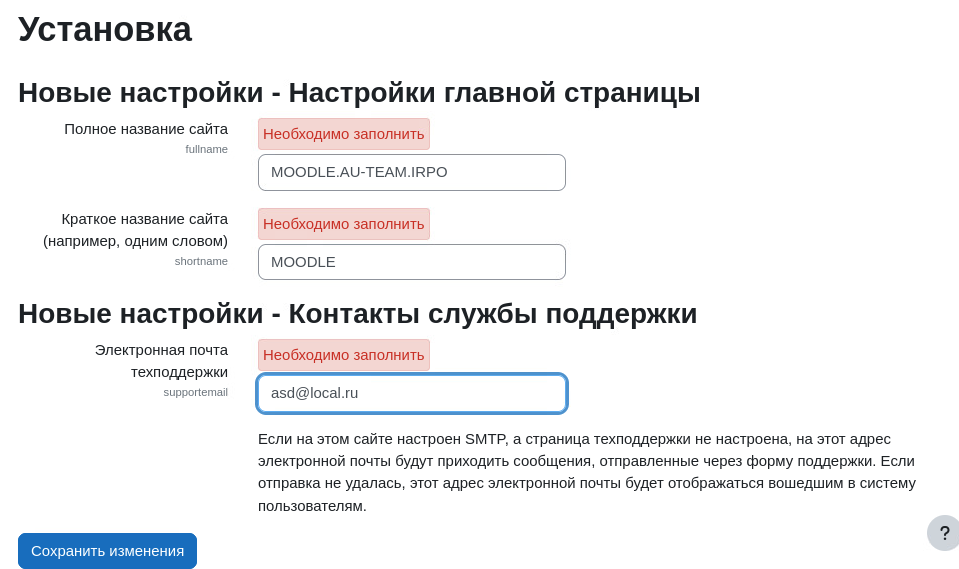
И далее нам необходимо указать данные для подключения к базе данных.



У нас должно быть всего 1 предупреждение, то что мы не используем https для **moodle**. Нажимаем продолжить если да, если нет, то необходимо исправить проблемы.

Далее просматриваем полностью вывод установки и смотрим, что бы везде было «Успешно». Далее переходим к настройке основной информации о сервере и стандартного admin пользователя. Заполняем данные, не забываем, что пароль должен быть «**P@ssw0rd**». Также заполните почту, например, «**asd@local.ru**».

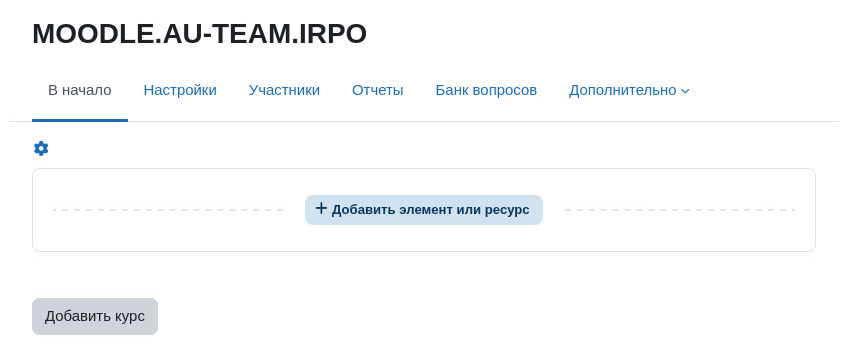
Тыкаем продолжить и у нас появляются поля, которые необходимо заполнить.



Готово, мы должны попасть на страницу **moodle** в «Личный кабинет».

Далее нам осталось только добавить сообщение на главную страницу, что у вас рабочее место с каким номером, для этого мы переключаемся в режим редактирования, кнопка сверху справа. И переходим в раздел «В начало».

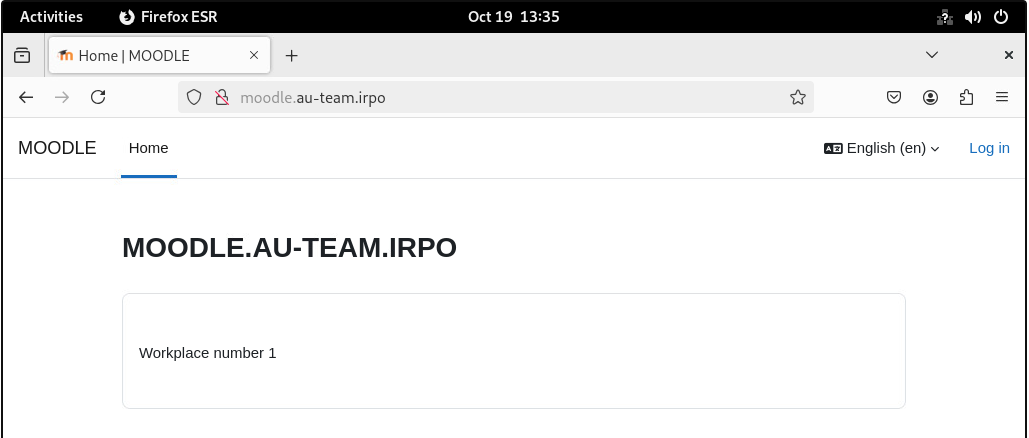
У вас будет похожий вид страницы:



Далее мы нажимаем на «Добавить элемент или ресурс».

Выбираем «Текст и медиа» и вписываем в «Название в оглавлении курса» строку «Workplace number 1», номер отличается рабочим местом пользователя.

Далее делаем выход из пользователя и попадаем на страницу Home, где видим наше сообщение, все готово.



Далее мы опять выполним настройку на маршрутизаторах **HQ-RTR**, **BR-RTR**, которые будут обеспечивать работу протоколов http, https, dns, ntp, icmp.

Для этого необходимо выполнить команды на **HQ-RTR**:

|  |
| --- |
| ip nat inside source static tcp 192.168.11.1 80 interface Ethernet0/0 80 |
| ip nat inside source static tcp 192.168.11.1 443 interface Ethernet0/0 443 |
| ip nat inside source static tcp 192.168.11.1 123 interface Ethernet0/0 123 |
| ip nat inside source static tcp 192.168.11.1 53 interface Ethernet0/0 53 |
| ip nat inside source static udp 192.168.11.1 53 interface Ethernet0/0 53 |

Для этого необходимо выполнить команды на **BR-RTR**:

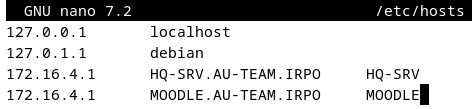
|  |
| --- |
| ip nat inside source static tcp 192.168.21.1 80 interface Ethernet0/0 80 |
| ip nat inside source static tcp 192.168.21.1 443 interface Ethernet0/0 443 |

Отлично, далее для доступности сервисов из вне, там необходимо использовать глобальный **DNS**, но так как у нас к нему доступа нет, мы не покупали адреса, нам необходимо воспользоваться локальным решением.

Для того, чтобы web сервисы работали которые мы прокинули, нам необходимо изменить файл **/etc/hosts**. А также на **CLI** необходимо установить **chrony**.

С **chrony** делаем все тоже самое, сервер указываем внешний адрес маршрутизатора **HQ-RTR**. (server = 172.16.4.1)

В файл hosts необходимо добавить 2 строки:



Теперь вы можете воспользоваться локальным сайтом из интернета.

Переходим на сайт **http://moodle.au-team.irpo/moodle**.

Все отлично открывается, хорошо!

Далее мы займемся мониторингом, устройств.

Мы будем устанавливать **zabbix**, на **HQ-SRV**.

Для этого нам необходимо выполнить некоторое количество действий:

Скачивание пакетов:

|  |
| --- |
| wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zabbix-release\_latest+debian12\_all.deb |
| dpkg -i zabbix-release\_latest+debian12\_all.deb |
| apt update |
| apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent mariadb-server |

Далее необходимо выполнить настройку базы данных для **zabbix**:

|  |
| --- |
| mysql |
| create database zabbix character set utf8mb4 collate utf8mb4\_bin; |
| create user zabbix@localhost identified by 'P@ssw0rd'; |
| grant all privileges on zabbix.\* to zabbix@localhost; |
| set global log\_bin\_trust\_function\_creators = 1; |
| quit; |

Далее нам необходимо выполнить импорт схем из файла:

|  |
| --- |
| zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 –uzabbix –p zabbix |

Далее нам предлагают ввести пароль, вводим пароль от **zabbix** пользователя.

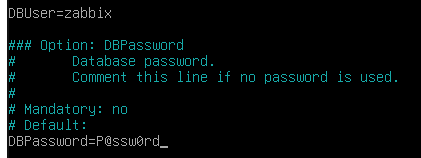
Далее опять необходимо изменить настройки mysql:

|  |
| --- |
| mysql |
| set global log\_bin\_trust\_function\_creators = 0; |
| exit; |

Далее необходимо изменить конфигурационный файл **zabbix\_server.conf**

Для, того, чтобы указать пароль для базы данных.

В файле **/etc/zbbix/zabbix\_server.conf**, находим строку **#DBPassword=** и изменяем на пароль **zabbix** пользователя;

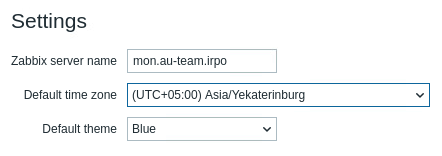


Перезагружаем сервер **HQ-SRV**.

Отлично, теперь переходим на сайт с помощью **HQ-CLI**.

И следуем установке. Указываем пароль для базы данных.

В настройках указываем:



Отлично, добавим адрес в DNS для мониторинга.



И переходим по новому адресу http://mon.au-team.irpo/zabbix

Проверьте, что zabbix у вас включается автоматически:

|  |
| --- |
| systemctl status zabbix-server |
| systemctl start zabbix-server |
| systemctl enable zabbix-server |

Далее необходимо настроить наши устройства для мониторинга, для этого необходимо сделать несколько вещей.

Для настройки сетевого оборудования необходимо внести изменения в конфигурацию **HQ-RTR**, **BR-RTR**:

|  |
| --- |
| snmp-server community au-team RO |

Далее необходимо настроить **BR-SRV**, а именно установить пакет **zabbix-agent** и настроить его на нужный сервер.

Выполняем команды, в необходимой строке изменяем данные:

|  |
| --- |
| apt install zabbix-agent |
| nano /etc/zabbix/zabbix\_agentd.conf |
| Server=192.168.11.1 |

Далее перезапускаем **zabbix-agent**:

|  |
| --- |
| systemctl restart zabbix-agent.service |

Далее необходимо добавить наши устройства для мониторинга в **HQ-SRV**.

Переходим на сайт **http://mon.au-team.irpo/zabbix**, через **HQ-CLI**.

Далее переходим в «**Data collection**», «**Hosts**».

Необходимо нажать на кнопку «Create host». Она находится вверху справа.

Заполняем необходимые данные:

**Host name** – Имя, назовем как есть.

**Templates** – Автоматические метрики, будем назначать необходимые.

**Host groups** – Объединение устройств, добавим все в 1 группу.

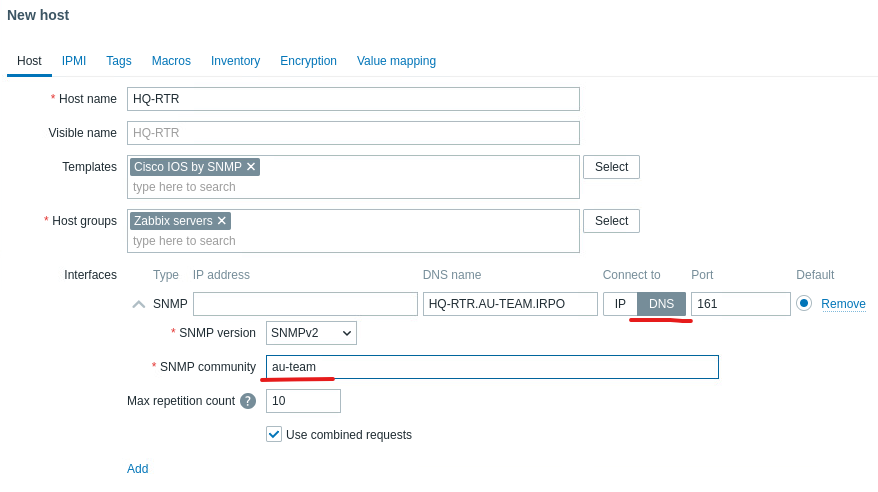
**Interfaces** – через, что будем мониторить, **SNMP** для **RTR**, **zabbix-agent** для **SRV**, а также ip адрес устройства.

Также нам необходимо будет определить на **RTR** устройствах **SNMP\_COMMUNITU**. Это то слово, которое мы писали в команде (**au-team**).

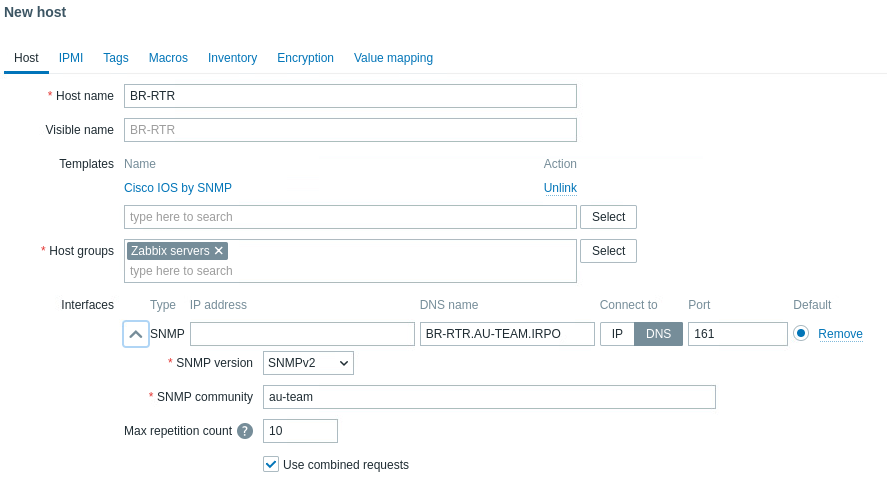
Более ничего не понадобится пока, что.

Далее я приведу примеры настроек **HQ-RTR**, **BR-RTR**, **BR-SRV**.

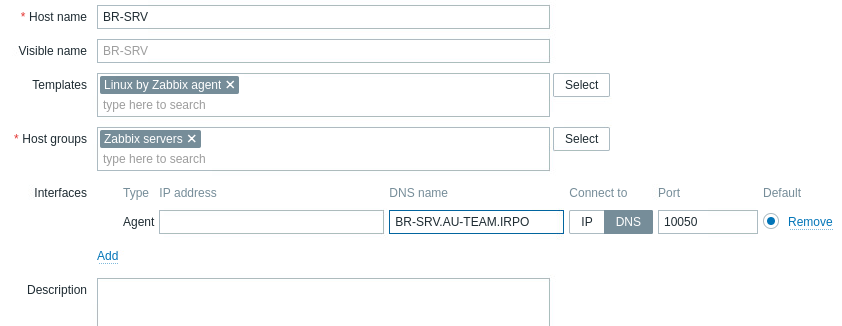
**HQ-RTR**



**BR-RTR** тоже самое, но другой ip, для ускорения можно зайти в созданный **HQ-RTR** и клонировать его.



**BR-SRV**



Далее проверяем, что бы у нас загорелись все зеленым цветом индикаторы справа от устройств добавленных.



Далее необходимо переименовать устройство **zabbix-server** в **HQ-SRV**.

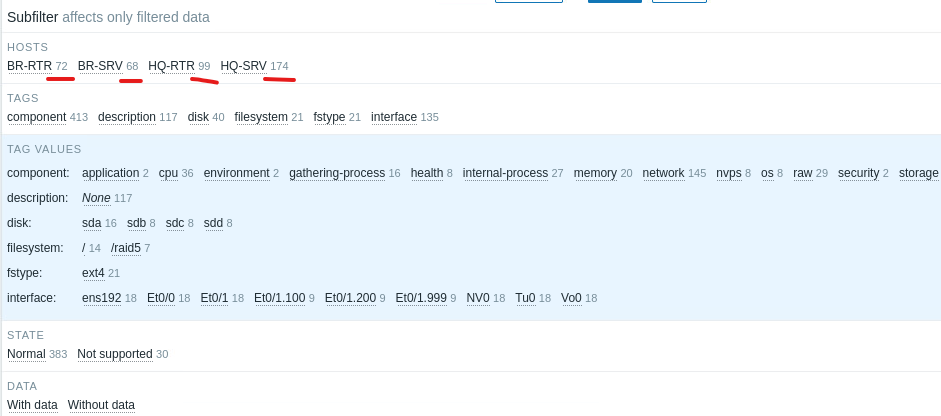
Открываем нажимая на имя и меняем **Host Name** на **HQ-SRV**.

Далее проверяем, данные которые мы получаем.

Переходим на страницу «**Monitoring**» «**Latest Data**».

И выбираем «**Host groups**» «**Apply**».

И получаем снизу вывод всех устройств, а именно:



Если у вас цифры немного отличаются ничего страшного:

Далее нам необходимо изменить логин пароль для входа в **zabbix**.

Для этого переходим в вкладку «**Users**» «**Users**».

Открываем пользователя **Admin** и нажимаем на кнопку «**Change password**».

Вводим данные:

**Current password**: zabbix

**Password**: P@ssw0rd1

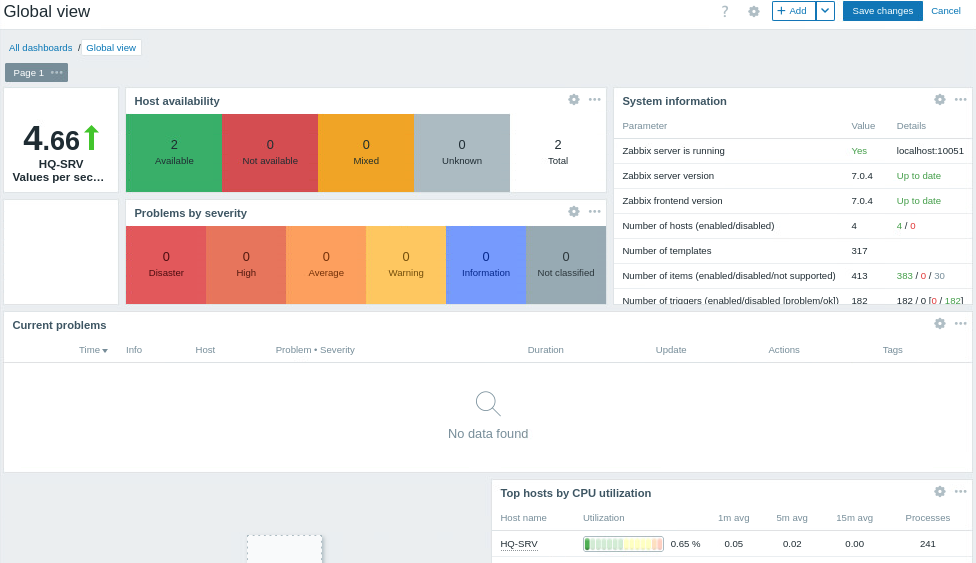
**Password (once again)**: P@ssw0rd1

Далее нам необходимо подготовить немного **Dashboard** мониторинга.

Потратьте некоторое время на ознакомление с управлением **Dashboard**, оно интуитивно понятно и простое, описывать это долго, проще попробовать.

Далее отдалите интерфейс для удобства управления «**Ctrl+Scroll Mouse**».

80% будет то, что нужно, далее разместите виджеты, таким образом.



Далее нам необходимо добавить мониторинг CPU, оперативной памяти и дисковой памяти. Сразу нужно понимать, что все здесь у нас не получится мониторить, так как это виртуализированная среда и она не все может подловить для мониторинга. Далее мы будем работать только с теми виджетами, которые находятся ниже **Current problems**.

Для начала давайте изменим настройки существующего виджета про **CPU HQ-SRV**. Зайдем в него нажав на шестеренку в правом верхнем углу виджета.

Изменим имя на «**Top hosts by CPU SRV**», а также добавим в **Hosts** новый сервер, а именно **BR-SRV** и подтверждаем изменения.



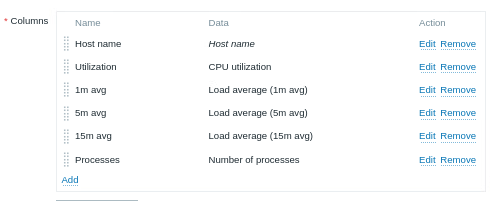
Пока, что так, далее мы его еще изменим.

Далее нам необходимо сделать такой же виджет, но для **RTR**, так как у них называется данные по-другому для вывода.

Выполняем копирование виджета и вставляем его ниже (3 точки возле шестеренки, копировать, нажимаем на пустое место, вставить виджет).

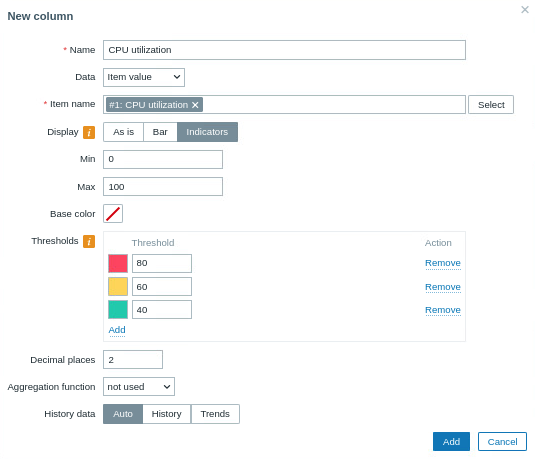
Изменяем виджет, на **RTR** вид. Изменим имя «**Top hosts by CPU RTR**».

Удаляем все Columns в виджете:



Далее нам необходимо добавить новые столбцы, нажимаем «**Add**».

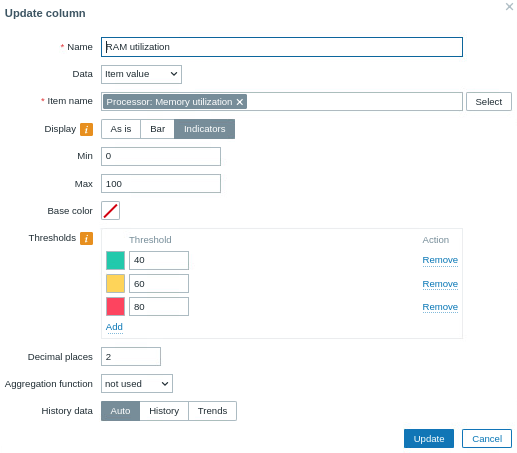
У нас появляется поле настроек столбца, указываем по примеру:



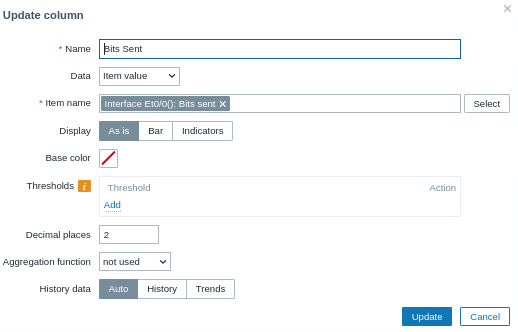
**Item name** – выбирается через **select**, там выбираете **RTR** устройство и самое первое идет, это название данных.

Далее нам необходимо настроить использование **оперативной памяти**.

Снова добавляем столбец и заполняем:



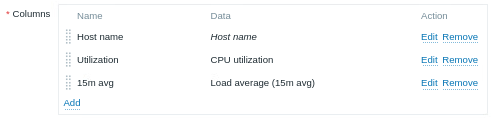
Далее на замену мониторинга основного накопителя, мы будем мониторить скорость выхода в интернет, для этого добавляем еще 1 столбец, тут проще настройка.



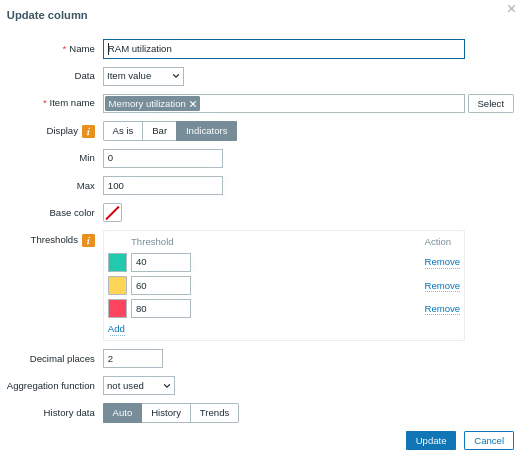
Готово, подтверждаем и у нас мониторинг устройств RTR закончен, осталось донастроить мониторинг SRV.

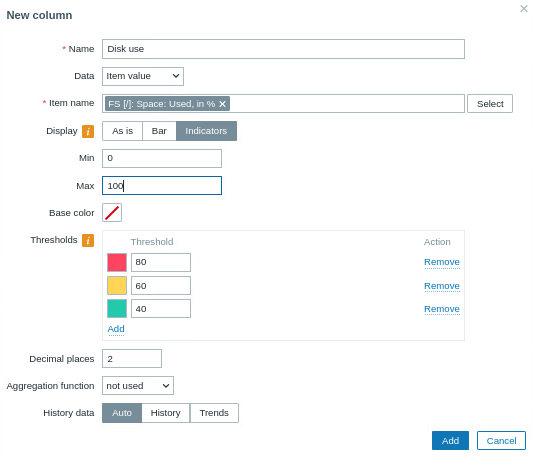
На осталось добавить по такому же принципу мониторинг оперативной памяти и основного накопителя.

Что бы не захламлять в **SRV** колонки давайте удалим **1m avg, 5m avg, Processes.**

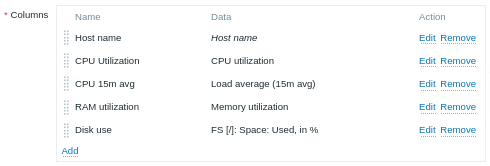


Далее добавим 2 столбца.



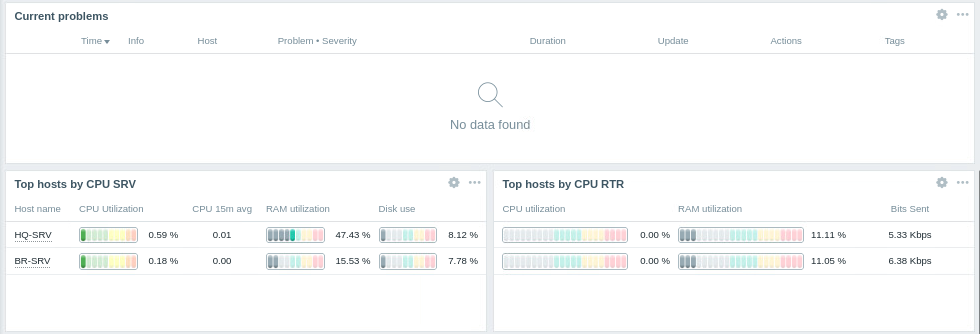


Нам необходим мониторить только основной накопитель, Raid мы не будем мониторить, также изменим названия нашим колонкам немного.



Подтверждаем.

Разместим более удобно.



Готово, не забудьте сохранить изменения.

Далее нам необходимо настроить принт-сервер **CUPS** на **HQ-SRV**.

Для начала давайте установим пакет как обычно.

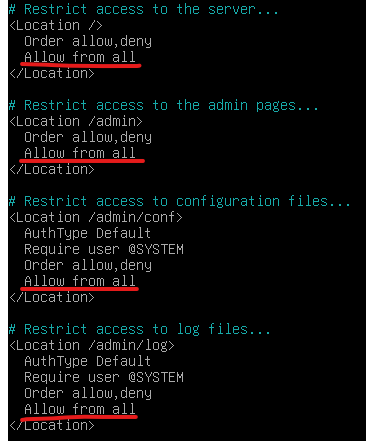
|  |
| --- |
| apt install cups |

Далее нам необходимо добавить текущего пользователя (root) в группу управления принтерами.

|  |
| --- |
| usermod –aG lpadmin $(whoami) |

Далее нам необходимо «**рашарить**» **CUPS**, для этого мы изменяем настройки в файле **/etc/cups/cupsd.conf**, там необходимо найти строку **Listen localhost:631** и изменить ее на **Port 631**.

Далее находим раздел **<Location />** и изменяем его:



Далее перезапускаем **CUPS**:

|  |
| --- |
| systemctl restart cups |

Далее запустим установку виртуального **PDF** принтера.

|  |
| --- |
| apt install printer-driver-cups-pdf |

После установки **CUPS** автоматически добавит виртуальный принтер.

Далее будут огромные лаги, не пугайтесь.

Далее необходимо внести изменения в локальный CUPS на HQ-CLI он уже установлен. Необходимо создать файл «**/etc/cups/client.conf**».

И добавить туда строку «**ServerName 192.168.11.1**».

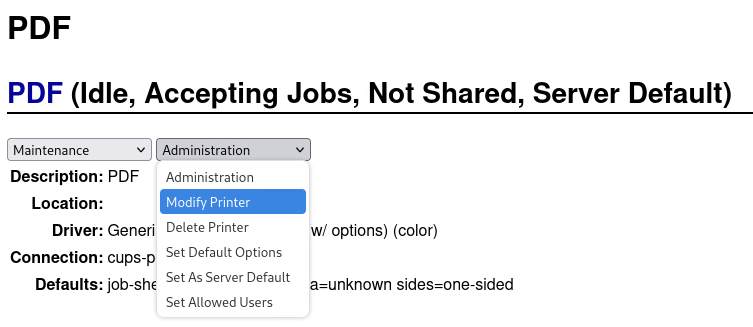
Перезапускаем **CUPS**.

|  |
| --- |
| systemctl restart cups |

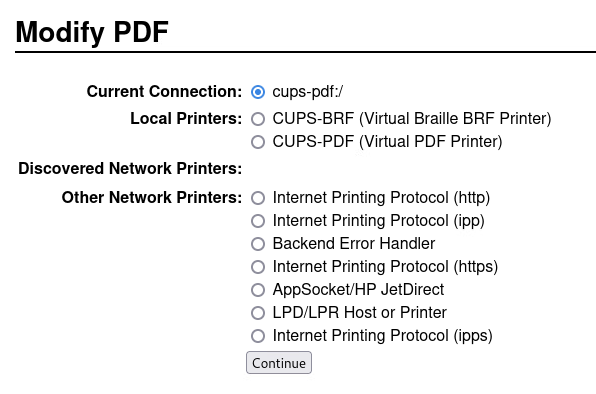
Теперь, нам необходимо зайти на сервер **HQ-SRV CUPS**. И создать новый принтер на основе **PDF** принтера и подключить его к **HQ-CLI**.

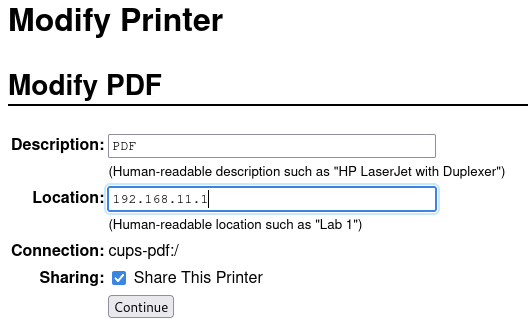
Переходим по адресу **https://192.168.11.1:631**.

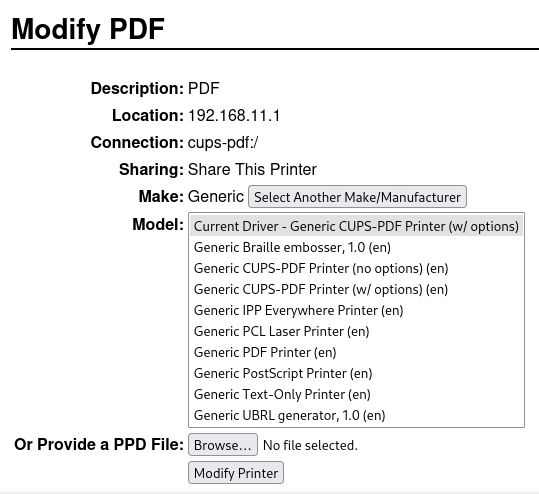
Сразу переходим в вкладку **Printers** и тыкаем на PDF. Далее необходимо нажать на открывающийся список в администрировании и выбрать модифицировать принтер.



Далее вас переведет на страницу настройки этого принтера, далее необходимо выбирать настройки по примеру:



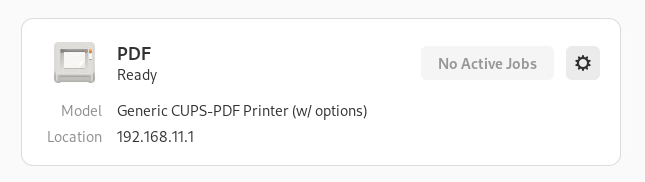




Мы должны получить сообщение, что принтер модифицирован.

Далее необходимо немного настроить **HQ-CLI**, переходим в настройки устройства, в вкладку **Printers** и видим наш принтер, который уже был добавлен.

Пример:



Далее нажимаем на шестеренку и делаем его по умолчанию если не сделан.

Далее переходим в **Printing Options** и нажимаем кнопку Test Page.

Также пробуем запустить печать любого файла через **terminal** на **HQ-CLI**.

|  |
| --- |
| lp /etc/hosts |

Пример:



Далее мы проверим, как все напечаталось.

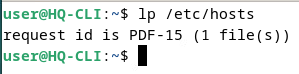
Переходим на сервер **CUPS** опять и заходим на страницу принтеров.

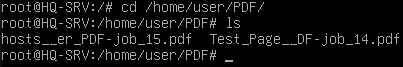
Переходим в принтер **PDF**. Нажимаем кнопку **Show All Jobs**.

Видим все наши задачи на печать.

Далее переходим на сервер **HQ-SRV** и переходи в папку **/home/user/PDF**.

У нас должны лежать файлы или файл, смотря сколько раз нажимали на тестовую печать и печатали, но от user пользователя на **HQ-CLI**.



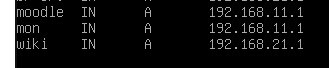


Готово.

Далее сделаем задание с массивным конфигом, который необходимо будет написать для реализации задания, а именно задания «**Развертывание приложений в Docker на сервере BR-SRV**». Нам понадобится установить пакет:

|  |
| --- |
| apt install docker-compose |

Пока устанавливается, давайте добавим **DNS** запись про **wiki**.

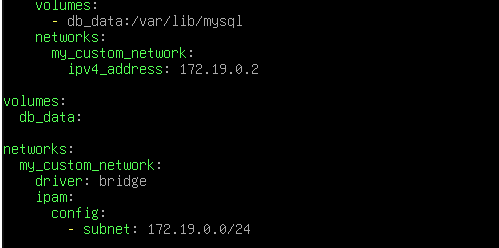




Далее нам необходимо создать файл для реализации стека контейнеров, а также связать их по средствам статического ip, так как мы далее будем настраивать доступ на сервер базы данных.

|  |
| --- |
| nano /root/wiki.yml |





И запускаем нашу чудо шарманку:

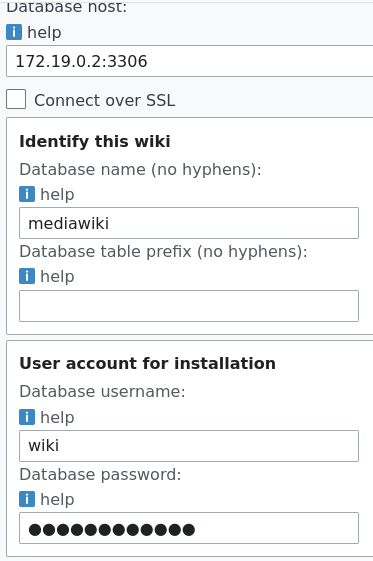
|  |
| --- |
| cd /root |
| docker-compose -f wiki.yml up -d |

Далее у нас произойдет магия и все само настроится и включится, останется сгенерировать только файл для **mediawiki**. А потом его скопировать на **BR-SRV**.

Далее переходим на **HQ-CLI** и переходим на сайт **http://wiki.au-team.irpo:8080** и мы попадаем на непонятную страницу где говорится, что не найден файл **LocalSettings.php** и что его необходимо сконфигурировать, тыкаем.

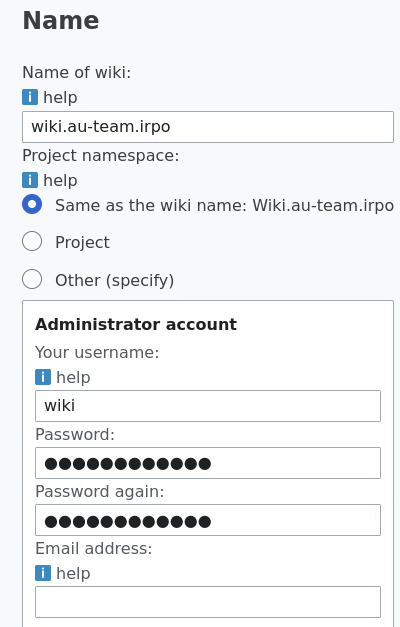
Далее у нас появляется окно с большим количеством данных, это все нормально, нам необходимо смотреть только в центральную часть, нажимаем на кнопку внизу, «**Продолжить**».

Доходим до настройки базы данных:



Пароль: WikiP@ssw0rd (как указывали в wiki.yml)

Далее настраиваем пользователя и имя wiki.



**Логин - wiki**

**Пароль - WikiP@ssw0rd**

В конце необходимо выбрать «**I’m bored already, just install the wiki**».

Продолжаем до конца и в конце у нас скачивается файл **LocalSettings.php**.

Далее нам необходимо перенести файл настроек с **HQ-CLI** на **BR-SRV**.

Переходим в **terminal** и выполняем команды:

|  |
| --- |
| cd /home/user/Download |
| scp -P 2024 LocalSettings.php sshuser@192.168.21.1:/home/sshuser/ |

Переходим в BR-SRV и выполняем команды:

|  |
| --- |
| mv /home/sshuser/LocalSettings.php /root/LocalSettings.php |

Далее необходимо изменить файл **wiki.yml** и расскомментировать 2 строки:



Далее осталось перезапустить контейнеры и проверить работоспособность.

Выполняем команды:

|  |
| --- |
| docker-compose -f wiki.yml down |
| docker-compose -f wiki.yml up -d |

Переходим в **HQ-CLI** и обновляем сайт, мы должны попасть на уже запущенную версию **MediaWiki**.