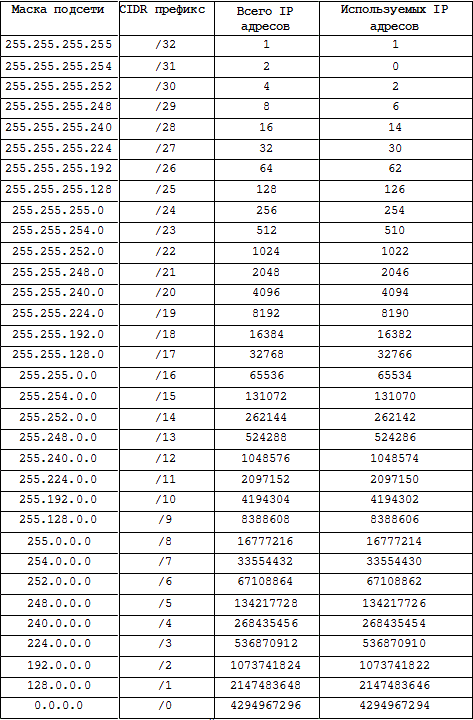
# Таблица адресации

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Устройство** | **Интерфейс** | **IP-адрес** | **Маска** | **VLAN** | **Подсеть** | **Шлюз** |
| **ISP** | eth0 (к интернету) | DHCP | DHCP | - | DHCP | DHCP |
|  | eth1 (к HQ-RTR) | 172.16.4.1 | 255.255.255.240 | - | 172.16.4.0/28 | - |
|  | eth2 (к BR-RTR) | 172.16.5.1 | 255.255.255.240 | - | 172.16.5.0/28 | - |
| **HQ-RTR** | eth0 (к ISP) | 172.16.4.2 | 255.255.255.240 | - | 172.16.4.0/28 | 172.16.4.1 |
|  | eth1 (Trunk) | - | - | Trunk | - | - |
|  | eth1.100 | 192.168.1.1 | 255.255.255.192 | 100 | 192.168.1.0/26 | - |
|  | eth1.200 | 192.168.2.1 | 255.255.255.240 | 200 | 192.168.2.0/28 | - |
|  | eth1.999 | 192.168.3.1 | 255.255.255.248 | 999 | 192.168.3.0/29 | - |
|  | gre1 (IP туннель) | 10.10.10.1 | 255.255.255.252 | - | 10.10.10.0/30 | - |
| **HQ-SRV** | enp0s3 (Trunk) | - | - | Trunk | - | - |
| enp0s3.100 | 192.168.1.2 | 255.255.255.192 | 100 | 192.168.1.0/26 | 192.168.1.1 |
| **HQ-CLI** | enp0s3.200 | 192.168.2.2 | 255.255.255.240 | 200 | 192.168.2.0/28 | 192.168.2.1 |
| **BR-RTR** | eth0 (к ISP) | 172.16.5.2 | 255.255.255.240 | - | 172.16.5.0/28 | 172.16.5.1 |
|  | eth1 (к BR-SRV) | 192.168.4.1 | 255.255.255.224 | - | 192.168.4.0/27 | - |
| gre1 (IP туннель) | 10.10.10.2 | 255.255.255.252 | - | 10.10.10.0/30 | - |
| **BR-SRV** | enp0s3 (к BR-RTR) | 192.168.4.2 | 255.255.255.224 | - | 192.168.4.0/27 | 192.168.4.1 |

****

## Настройка протокола динамической конфигурации хостов (DHCP):

Настройка будет производиться на **HQ-RTR**!

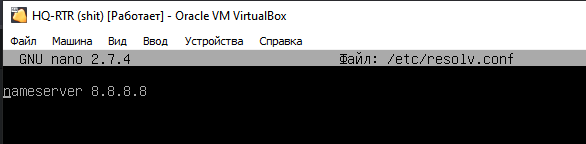
Использовать в качестве DHCP мы будем dnsmasq, служба, которой по умолчанию нет в наших ОС Российского производства.

Ещё нам нужно добавить в /etc/resolv.conf сервер Google, иначе мы не сможем обновить репозитории, поэтому идём его редактировать следующей командой:

**mcedit /etc/resolv.conf**

И добавляем следующую строку в него:

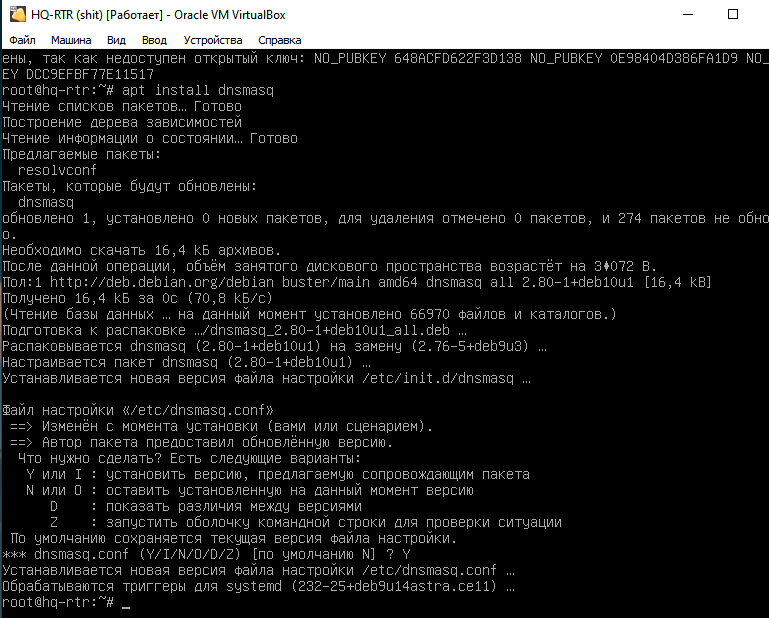
**nameserver 8.8.8.8**



Обновим пакеты и установим её командами:

**apt update**

**apt install dnsmasq**



Затем зайдем в настройки конфигурационного файла командой:

**mcedit /etc/dnsmasq.conf**

И внесем в него следующие строки (можно прямо в начало файла):

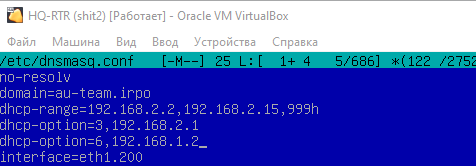
**no-resolv**

**dhcp-range=192.168.2.2,192.168.2.15,999h**

**dhcp-option=3,192.168.2.1**

**dhcp-option=6,192.168.1.2**

**interface=eth1.200**



Затем перезапускаем службу и посмотрим её статус:

**systemctl restart dnsmasq**

**systemctl status dnsmasq**

Для начала необходимо отключить несовместимую службу bind если она есть, командой

**systemctl disable --now bind**

Для работы DNS есть служба dnsmasq (она же и для DHCP. УДОБНО!)

Установим её на наш сервер с помощью следующей команды (если есть, как у меня, то переходите к следующему шагу):

Ещё нам нужно добавить в /etc/resolv.conf сервер Google, иначе мы не сможем обновить репозитории, поэтому идём его редактировать следующей командой:

**mcedit /etc/resolv.conf**

И добавляем следующую строку в него:

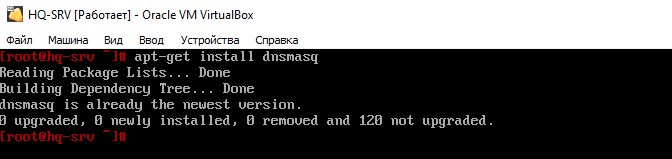
**nameserver 8.8.8.8**

Обновим пакеты и установим её командами:

**apt-get update**

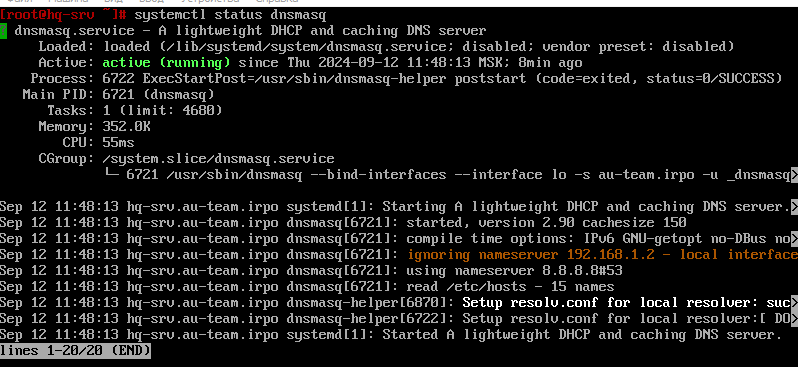
**apt-get install dnsmasq** (Установка пакета dnsmasq)

**systemctl enable --now dnsmasq** (Добавление службы в автозапуск)



Проверим её состояние перед работой:

**systemctl status dnsmasq**



Затем откроем файл для редактирования конфигурации нашего DNS-сервера:

**mcedit /etc/dnsmasq.conf**

И добавляем в неё строки (для удобства прям с первой строки файла):

**no-resolv** (не будет использовать /etc/resolv.conf)

**domain=au-team.irpo**

**server=8.8.8.8** (адрес общедоступного DNS-сервера)

**interface=\*** (на каком интерфейсе будет работать служба)

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

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

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

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

**address=/br-rtr.au-team.irpo/192.168.4.1**

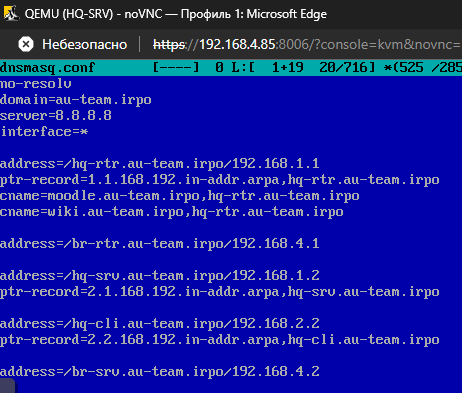
**address=/hq-srv.au-team.irpo/192.168.1.2**

**ptr-record=2.1.168.192.in-addr.arpa,hq-srv.au-team.irpo**

**address=/hq-cli.au-team.irpo/192.168.2.2 (**Смотрите адрес на **HQ-CLI**, т.к он выдаётся по DHCP**)**

**ptr-record=2.2.168.192.in-addr.arpa,hq-cli.au-team.irpo**

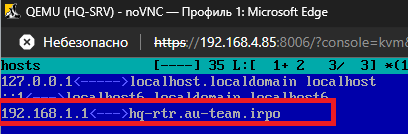
**address=/br-srv.au-team.irpo/192.168.4.2**

****

Сохраняем файл нажатием кнопки **F2**, а затем выход с помощью **F10.**

Теперь необходимо добавить строку **192.168.1.1 hq-rtr.au-team.irpo** в файл /etc/hosts:

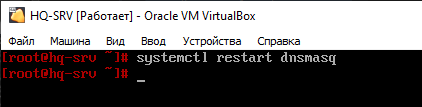
**mcedit /etc/hosts**



Сохраняем файл, выходим из редактора.

Перезапускаем службу командой:

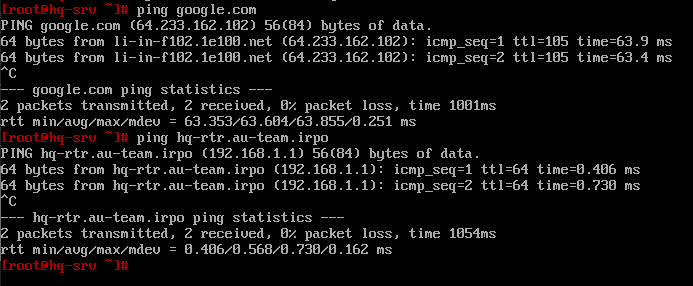
**systemctl restart dnsmasq**

****

Проверим пинг сначала с HQ-SRV на google.com и hq-rtr.au-team.irpo:

**ping google.com**

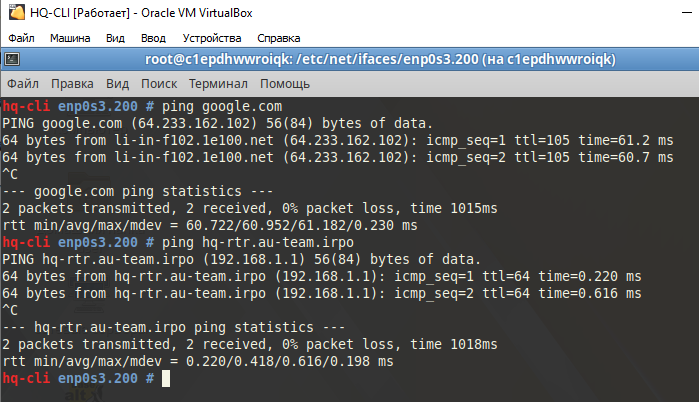
**ping hq-rtr.au-team.irpo**

****

Теперь проверим пинг с HQ-CLI:

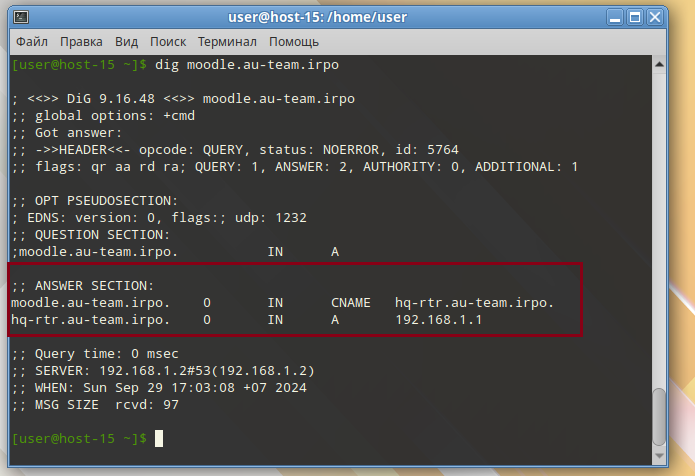
**ping google.com**

**ping hq-rtr.au-team.irpo**

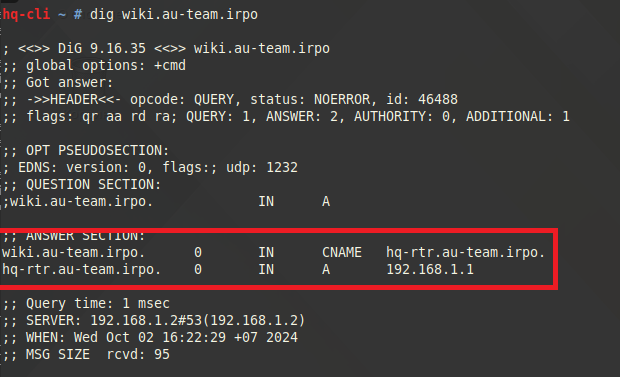


И проверим работу CNAME записей с HQ-CLI:

**dig moodle.au-team.irpo**



**dig wiki.au-team.irpo**



Наш DNS-сервер настроен.

## Создание локальных учетных записей:

### Создание на HQ-SRV:

Для создания пользователя с определённым идентификатором на машине под управлением ОС Alt Linux нужно использовать команду:

**useradd sshuser -u 1010**



Для проверки можно использовать команду:

**id sshuser**

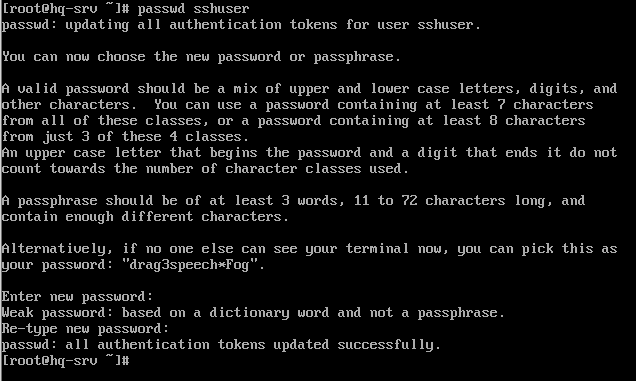


Чтобы задать пользователю новый пароль нужно использовать команду:

**passwd sshuser**

После чего ввести и подтвердить новый пароль:

**P@ssw0rd**

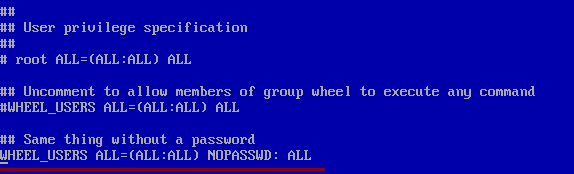


Чтобы **sshuser** мог запускать **sudo без дополнительной аутентификации**, необходимо убрать комментарий с двух строчек в файле **/etc/sudoers**, откроем его командой:

**mcedit /etc/sudoers**

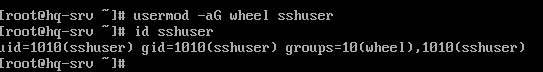
И уберём комментарий на следующей строке:

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



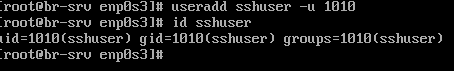
После чего добавить пользователя sshuser в группу wheel:

**usermod -aG wheel sshuser**

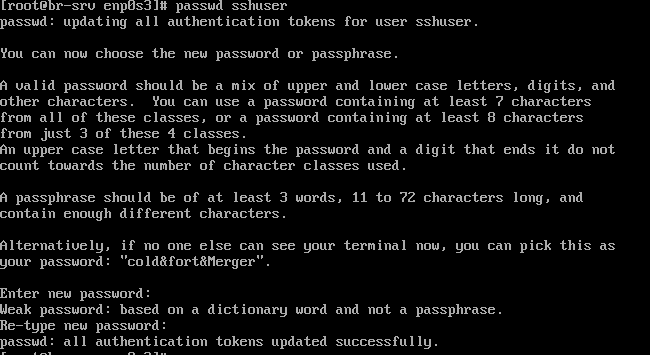


### Создание на BR-SRV:

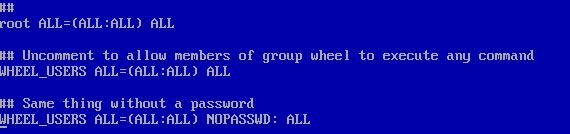
Создание пользователя:

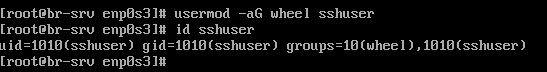


Редактирование пароля:



Повышение прав:

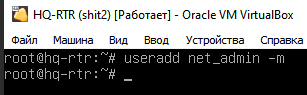




Создание на **HQ-RTR**:

Для создания пользователя на машине под управлением ОС Astra Linux нужно использовать команду:

**useradd** **net\_admin -m**

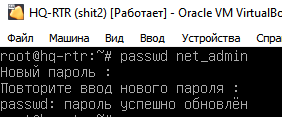


Изменим пароль:

**passwd net\_admin**

**P@$$word**

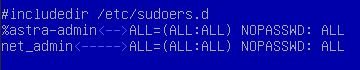
**P@$$word**

****

Чтобы **net\_admin** мог запускать sudo без дополнительной аутентификации необходимо добавить следующую строчку в файл **/etc/sudoers**, в самый конец:

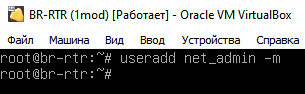
**mcedit /etc/sudoers**

**net\_admin ALL=(ALL:ALL) NOPASSWD: ALL**

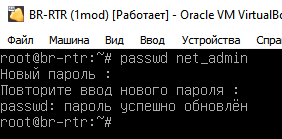


**Аналогично для** **BR-RTR**:

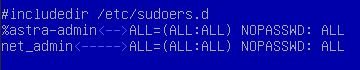
Создание пользователя:



Изменение пароля:



Повышение прав:

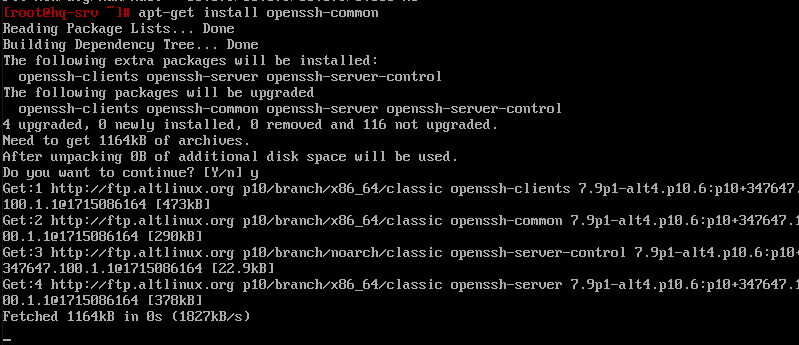


## Настройка безопасного удаленного доступа на серверах HQ-SRV и BR-SRV (SSH):

### Настройка на HQ-SRV :

Для работы SSH нам понадобится служба openssh-common, которой изначально нет, поэтому установим её:

**apt-get install openssh-common**



Затем зайдём в файл конфигурации для внесения изменений:

**mcedit /etc/openssh/sshd\_config**

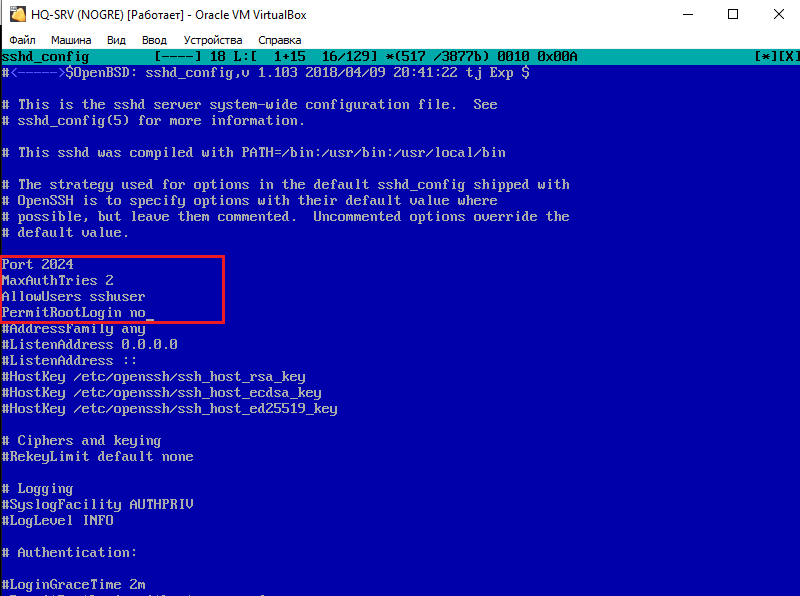
И внесём туда следующие строки:

**Port 2024**

**MaxAuthTries 2**

**AllowUsers sshuser**

**PermitRootLogin no**

****

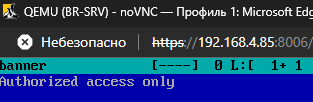
Далее нам нужен баннер.

Создаём его, вносим предложение, которое требуется по заданию через команду:

**mcedit /root/banner**

Пишем туда следующую строку (ОБЯЗАТЕЛЬНО ПОСЛЕ НЕЁ НАЖАТЬ ENTER, чтобы под ней была пустая строка):

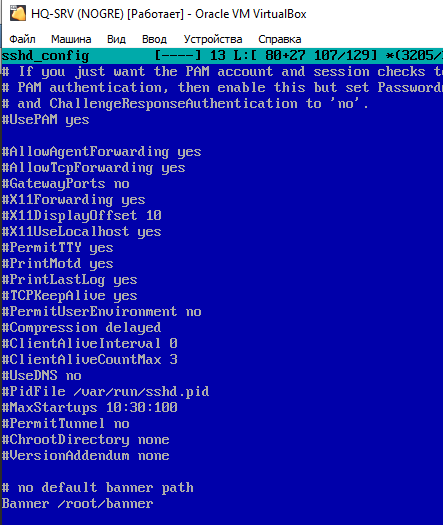
**Authorized access only**

****

Затем сохраняем и возвращаемся в **/etc/openssh/sshd\_config**.

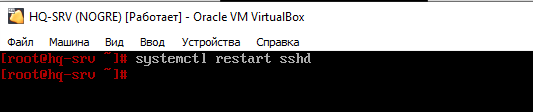
Добавляем/Редактируем следующую строку:

**Banner /root/banner**



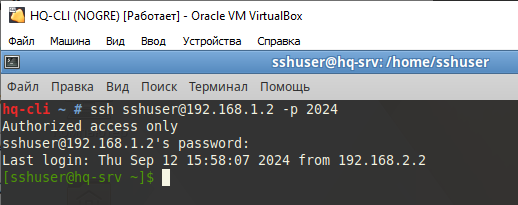
После внесения изменений, сохраняем и выходим. И делаем перезапуск службы:  
**systemctl enable --now sshd**

**systemctl restart sshd**



Затем попробуем подключиться по SSH через HQ-CLI:

**ssh sshuser@192.168.1.2 -p 2024**



**sshuser** – пользователь, под которым вы подключаетесь

**192.168.1.2** – адрес сервера, к которому мы подключаемся (**HQ-SRV**)

**-p 2024** – порт, по которому мы подключаемся (мы заменили со стандартного 22 на 2024)

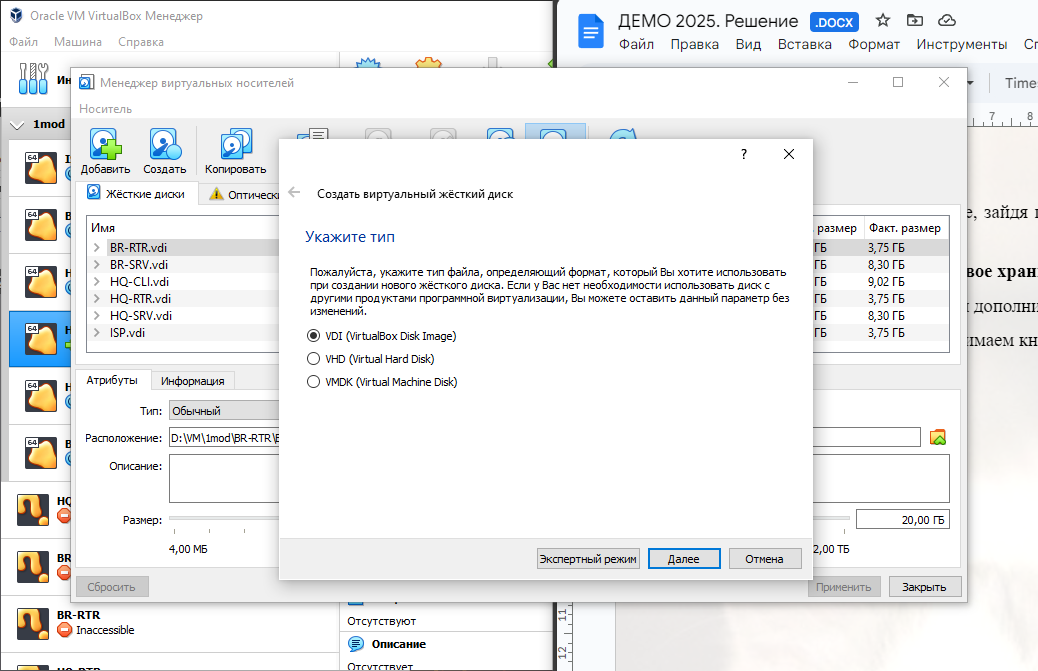
**Проделываем все тоже самое и на сервере** **BR-SRV**.

Сервис безопасного удаленного доступа настроен.

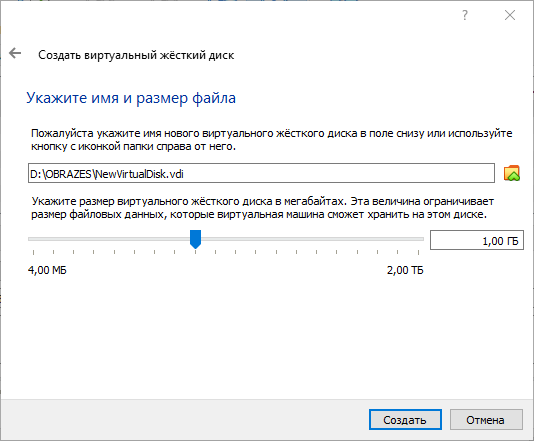
## Конфигурация файлового хранилища.

Для начала нужно создать три дополнительных диска размером 1 ГБ каждый.

Для этого жмём **Ctrl+D** и нажимаем кнопку создать.

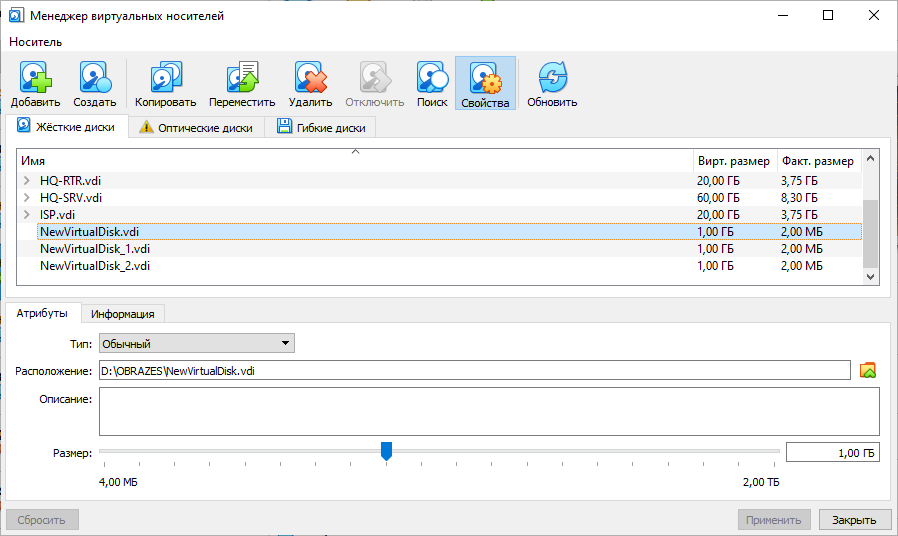


Все значения выбираем по умолчанию за исключением размера, его выставляем в 1 ГБ.

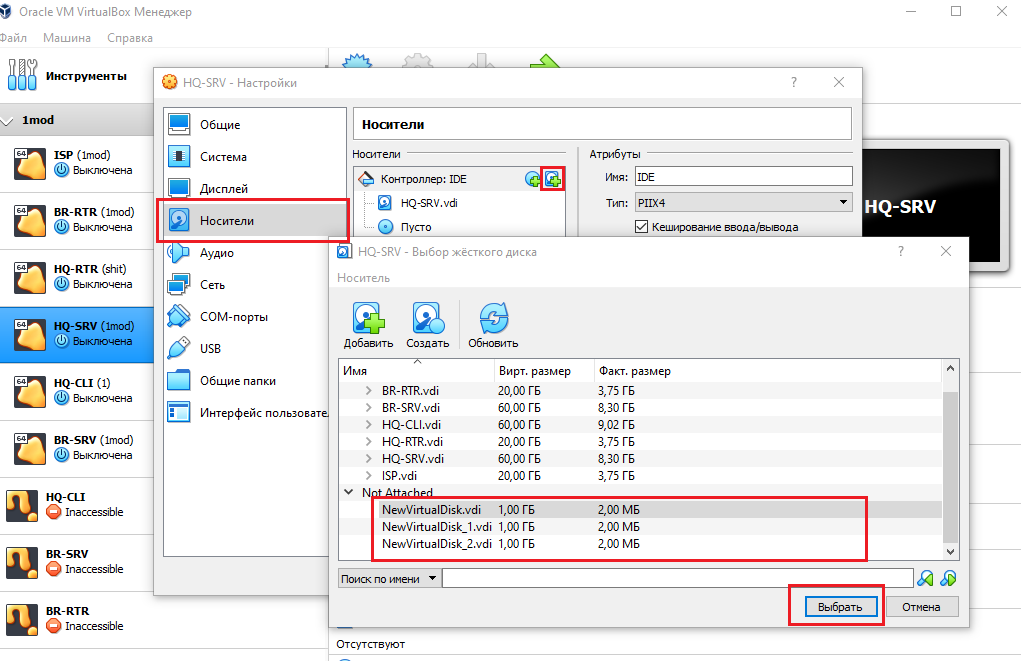


Создаём остальные два таким же образом.

В итоге должно получиться так:

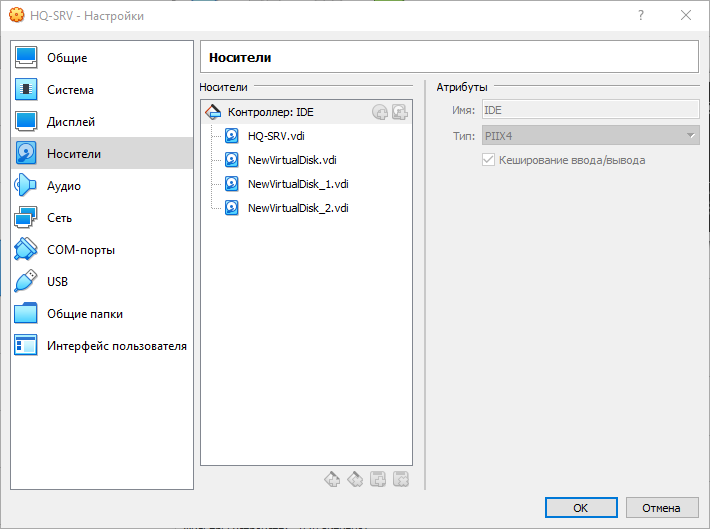


Теперь нам нужно их подключить к виртуальной машине **HQ-SRV**, для этого нужно зайти в настройки и в носителях нажать на значок жесткого диска, а затем подключим три созданных нами ранее диска.



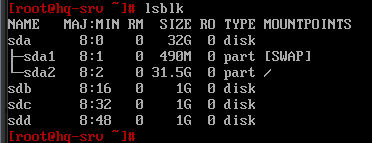
Обратите внимание, что с контроллером, который стоит по умолчанию (**IDE**) не позволяет подключать больше четырех носителей, поэтому удаляем CD-привод и ставим ещё один диск.

В итоге должно получится так:



Теперь заходим в виртуальную машину и просматриваем все диски, которые мы подключили, следующей командой:

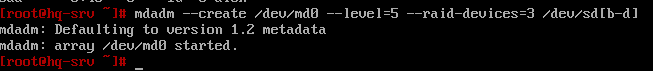
**lsblk**



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

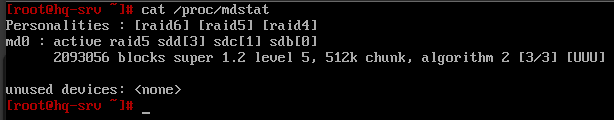
Теперь создадим дисковый массив уровня 5 из трёх дополнительных дисков следующей командой:

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



Посмотрим статус нашего raid-массива:

**cat /proc/mdstat**



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

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

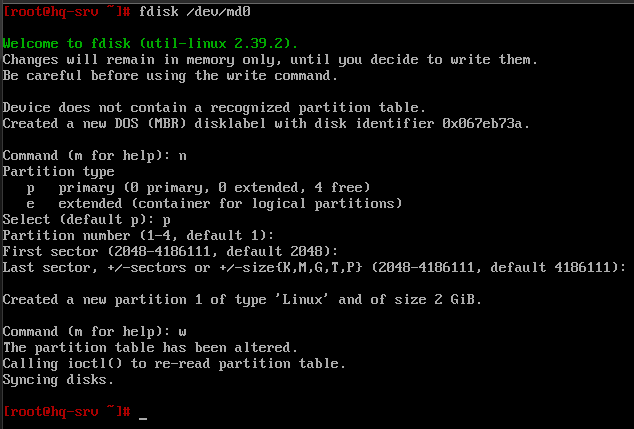
Теперь создаём раздел через **fdisk**.

Для этого пишем следующую команду:

**fdisk /dev/md0**

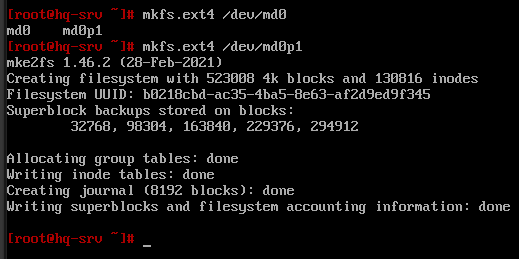


Затем пишем **n**, чтобы создать раздел, прокликиваем **Enter**, потому что он по дефолту предлагает то, что нам нужно, а в конце пишем **w**, чтобы записать изменения:

****

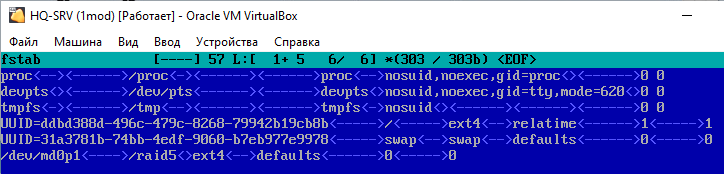
Теперь создадим файловую систему, по заданию требуется **ext4**, создаём её следующей командой:

**mkfs.ext4 /dev/md0p1**

****

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

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

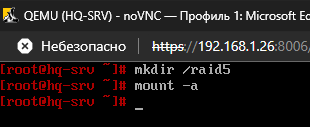


Затем создаём каталог **/raid5** и монтируем ФС из **/etc/fstab**:

**mkdir /raid5**

**mount -a**

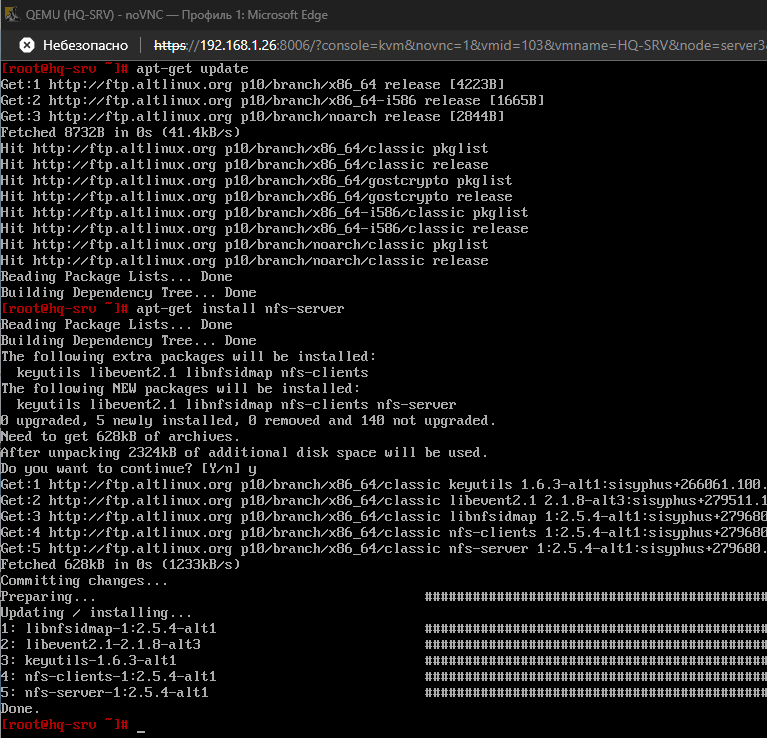
Заметьте, что команда не должна ничего выводить!

****

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

**apt-get update**

**apt-get install nfs-server**

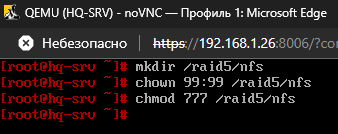


Приступаем к самой настройке, создадим каталог, назначим нового владельца и группу ему и выдадим новые права:

**mkdir /raid5/nfs**

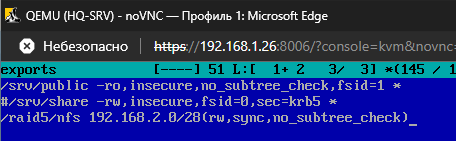
**chown 99:99 /raid5/nfs**

**chmod 777 /raid5/nfs**



Откроем каталог для общего доступа в сторону подсети, где находится **HQ-CLI**, для этого заходим в **/etc/exports** и пишем следующую строку в конец файла:

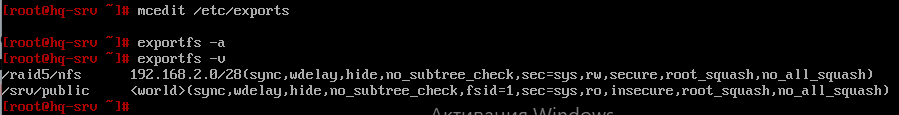
**/raid5/nfs 192.168.2.0/28(rw,sync,no\_subtree\_check)**



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

**exportfs -a**

**exportfs -v**



Перезапускаем службу NFS:

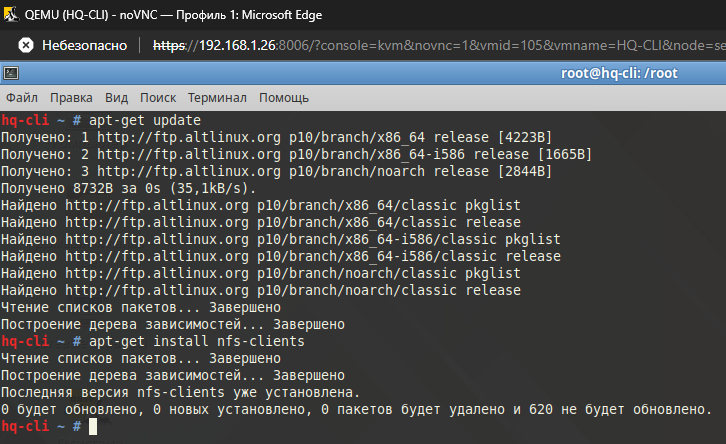
**systemctl restart nfs**



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

**apt-get update**

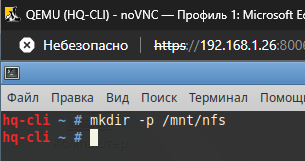
**apt-get install nfs-clients**

****

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

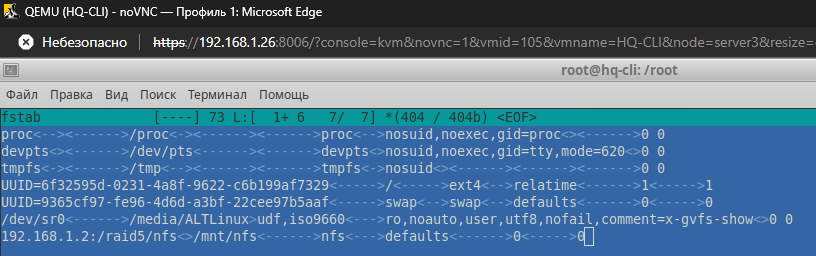
Теперь настроим автоматическое монтирование в каталог **/mnt/nfs**, но для начала создадим его:

**mkdir -p /mnt/nfs**

****

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

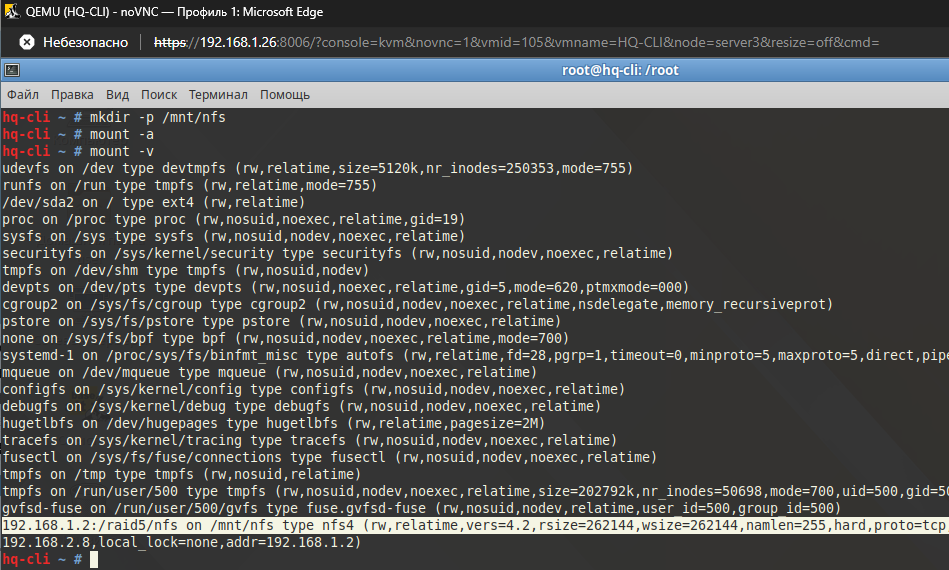
**192.168.1.2:/raid5/nfs /mnt/nfs nfs defaults 0 0**



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

**mount -a**

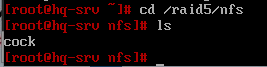
**mount -v**



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

**touch /mnt/nfs/cock**



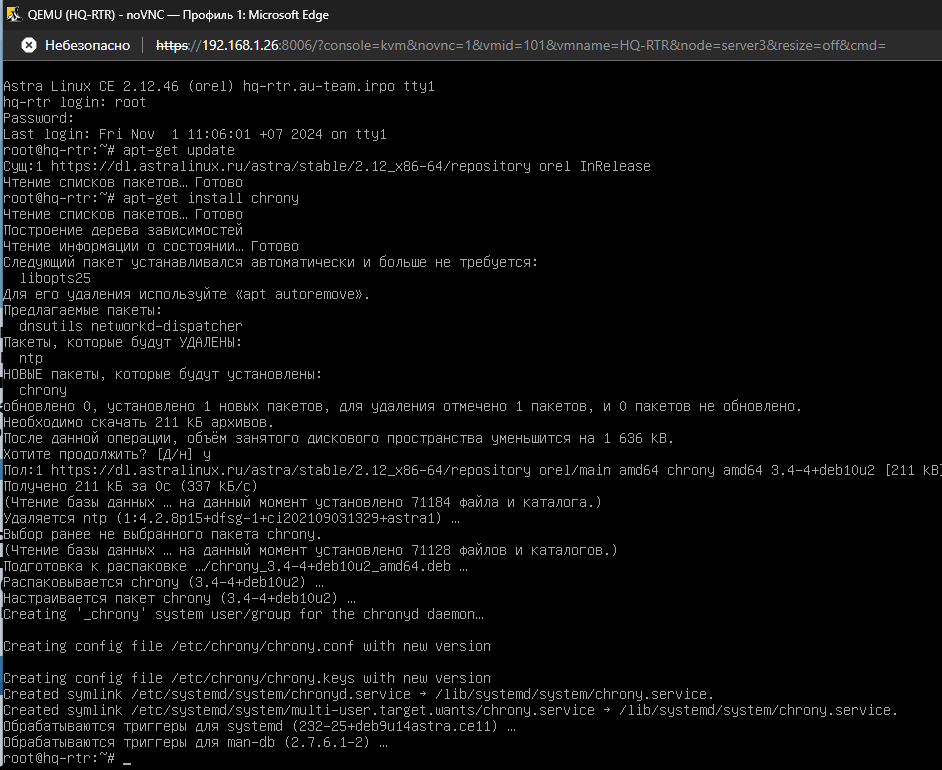


## Настройка службы сетевого времени на базе сервиса chrony

Для его настройки обновим список пакетов и установим сам пакет:

**apt update**

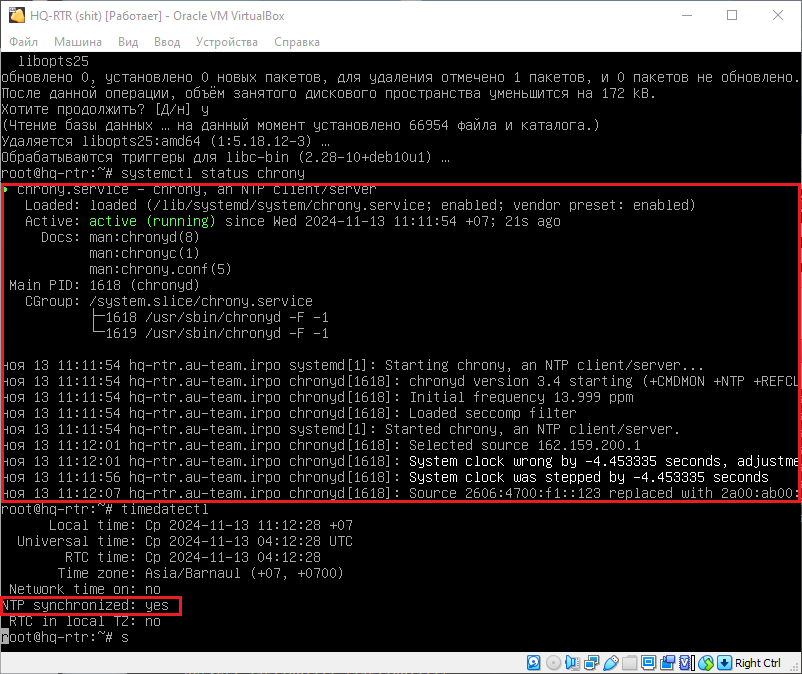
**apt install chrony**



Проверим работу службы **chrony** и **timedatectl**:

**systemctl status chrony**

**timedatectl**



Видим, что всё функционирует и синхронизация активна, которую мы позже уберём. Синхронизация нужна сейчас для того, чтобы мы получили самое точное время от серверов со страмумом 0 и так далее. Позже мы будем пользоваться локальным временем, которое сохранится благодаря нашему chrony на роутере, и уже он будет выступать сервером для устройств клиентов.

Теперь приступим к настройке, редактируем файл **/etc/chrony/chrony.conf**, введем в начало файла следующие строки и в целом приводим файл к такому виду:

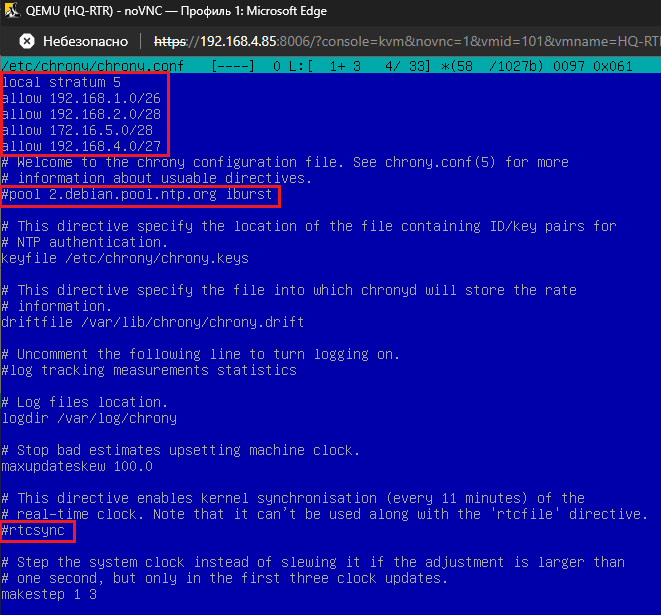
**local stratum 5**

**allow 192.168.1.0/26**

**allow 192.168.2.0/28**

**allow 172.16.5.0/28**

**allow 192.168.4.0/27**



Файл должен выглядеть так, как на скрине, закомментировав ещё две строки, они нам нужны были только синхронизации с атомными часами.

Включаем и перезапускаем службу **chrony**:

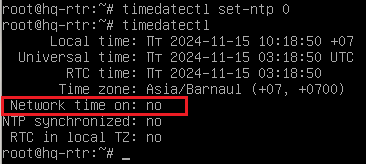
**systemctl enable --now chrony**

**systemctl restart chrony**

Выключаем теперь ту самую синхронизацию, оставляя, по сути, главным сервером NTP – наш роутер **HQ-RTR,** и проверяем ещё раз статус **timedatectl**:

**timedatectl set-ntp 0**

**timedatectl**

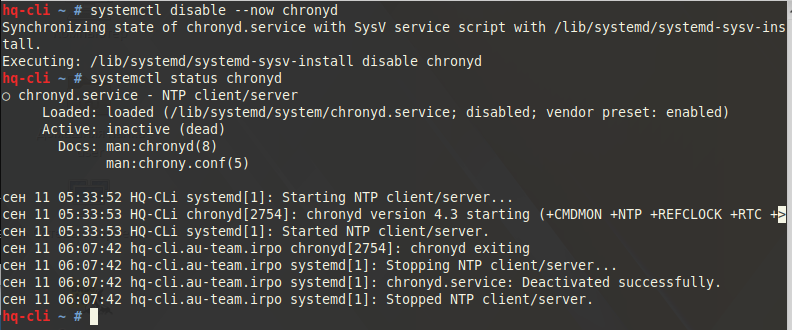
****

Теперь переходим к настройке клиента, в качестве него возьмём **HQ-CLI**, но по заданию есть ещё, но о них позже.

Перед установкой новой службы выключим chrony на **HQ-CLI**:

**systemctl disable --now chronyd**

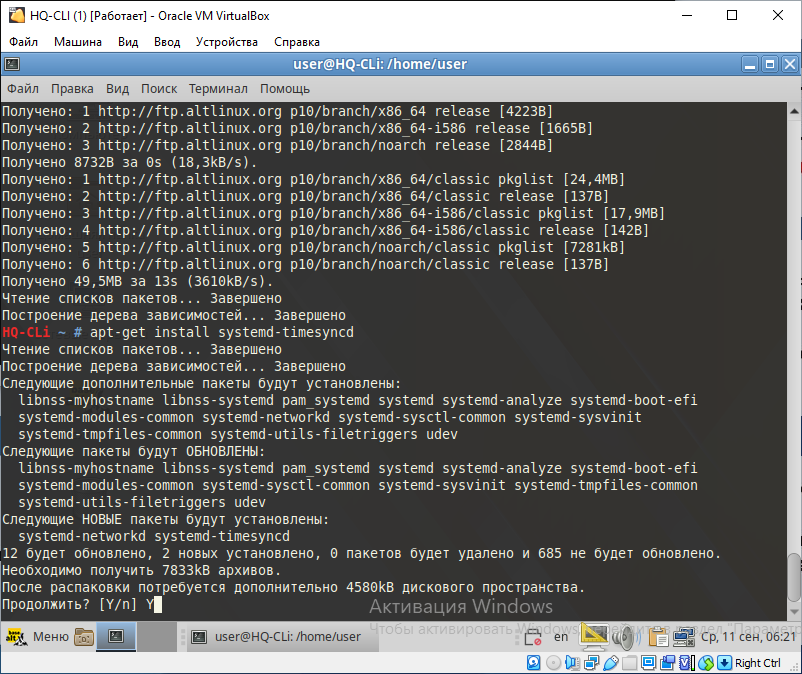
**systemctl status chronyd**

****

Обновляем список пакетов на **HQ-CLI** и скачиваем службу **systemd-timesyncd**:

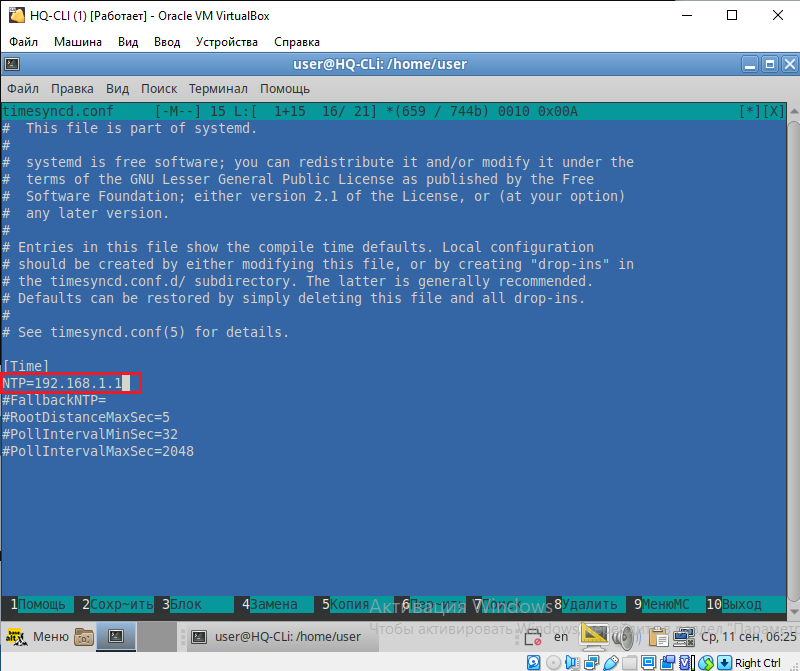
**apt-get update**

**apt-get install systemd-timesyncd**

****

Теперь зайдём в конфиг **/etc/systemd/timesyncd.conf** и отредактируем только одну строку:

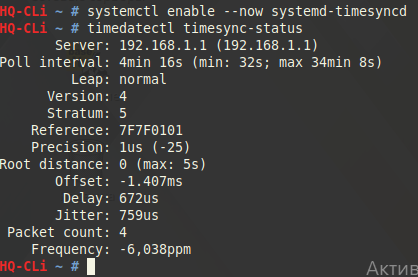
**NTP=192.168.1.1**



Теперь включим службу **systemd-timesyncd** и посмотрим её статус работы:

**systemctl enable --now systemd-timesyncd**

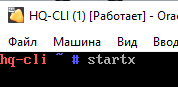
**timedatectl timesync-status**

****

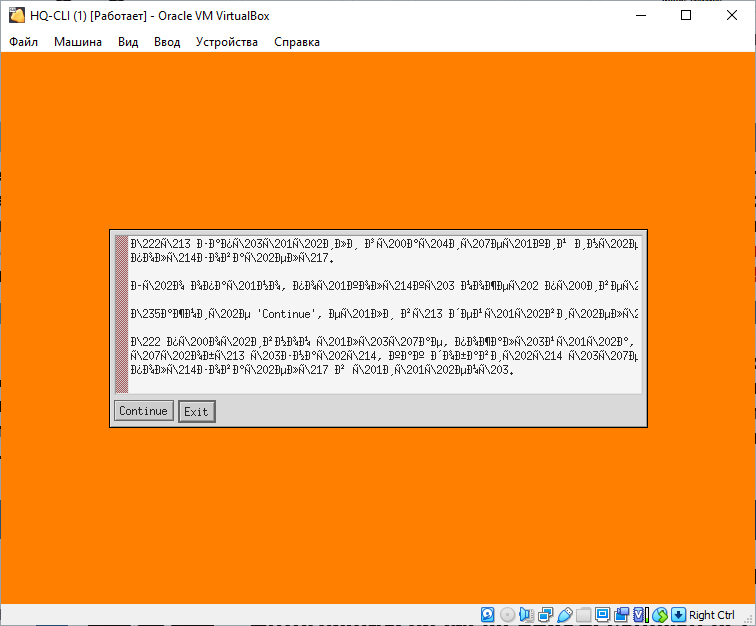
Видим, что стратум совпадает по заданию и всё работает. Задание выполнено.

Но есть одно НО! После перезагрузки клиентской машины **HQ-CLI** и вход в пользователя у вас может не появиться рабочий стол. Для этого мы заходим снова через **Ctrl+Alt+F2** во второе окно и прописываем команду **startx**. Это заставит запуститься графическую среду.

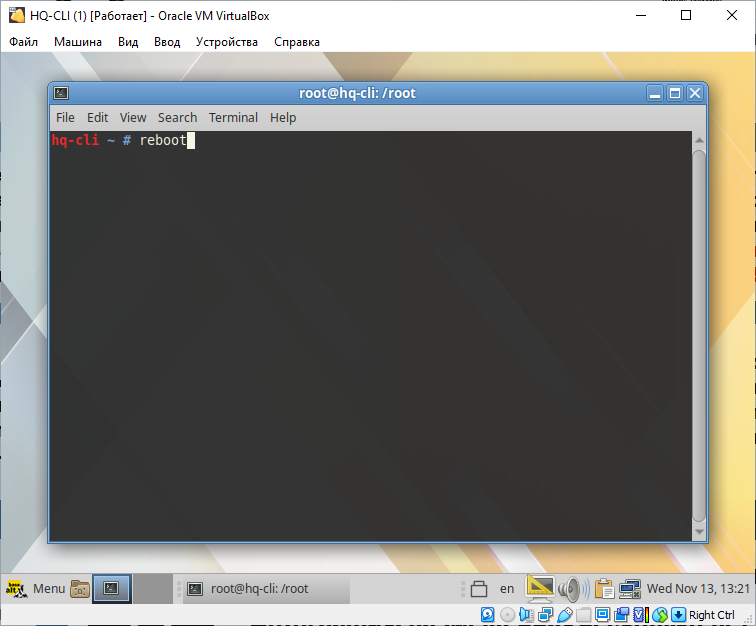
**ДЕЛАЕМ ЭТО ТОЛЬКО В СЛУЧАЕ ПОЛОМКИ ГРАФИЧЕСКОЙ СРЕДЫ!**



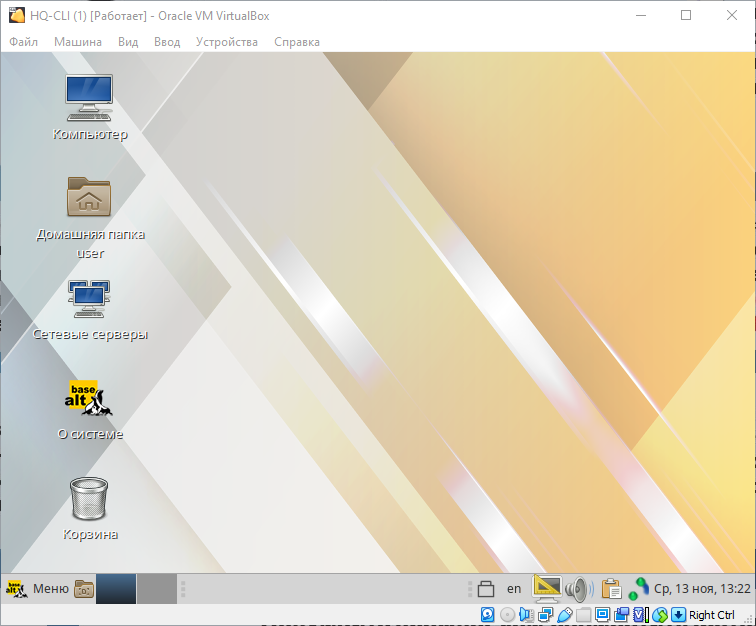
Далее мы увидим следующий бред на скриншоте, на нём мы прожимаем continue:



После этого она начнёт потихоньку запускаться, но мы ещё раз перезагружаем систему через **reboot** и система должна вновь нормально запуститься:



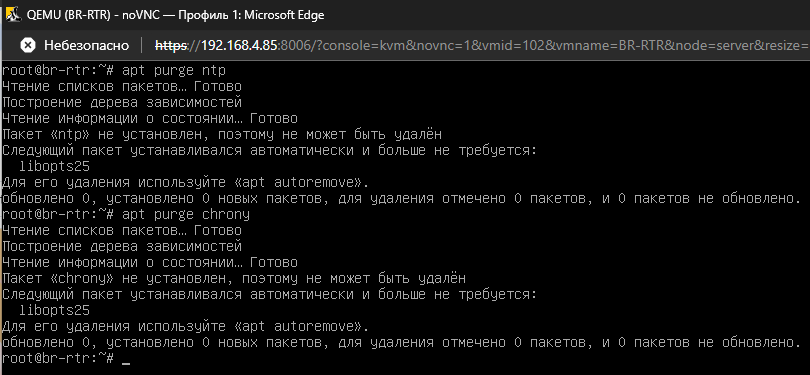
Вуаля, всё успешно, и дата на месте, и графическая среда!



Настроим теперь **BR-RTR**, удаляем пакеты **ntp**, **chrony**, если они есть:

**apt purge ntp**

**apt purge chrony**

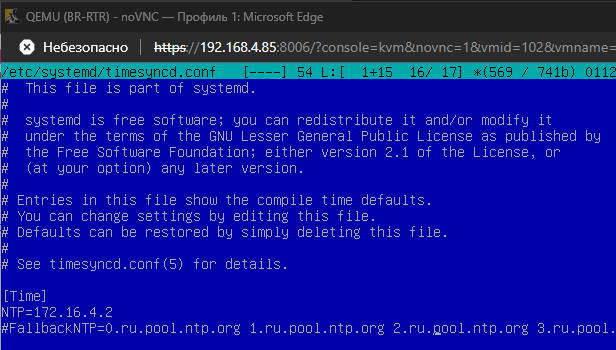
****

На Астре по умолчанию должна быть установлена служба **systemd-timesyncd**, но если её нет, то установите:  
**apt update**

**apt install systemd-timesyncd**

Настроим также его конфиг в **/etc/systemd/timesyncd.conf**:

**NTP=172.16.4.2**

****

Теперь включим службу **systemd-timesyncd** и готово:

**systemctl enable --now systemd-timesyncd**

На остальных клиентах нужно проделать тоже самое, исходя из документа остались - **HQ-SRV**, **BR-SRV** (настройка идентична клиенту **HQ-CLI**, поэтому [клик](#bookmark=id.4zebt2tw2hol)).

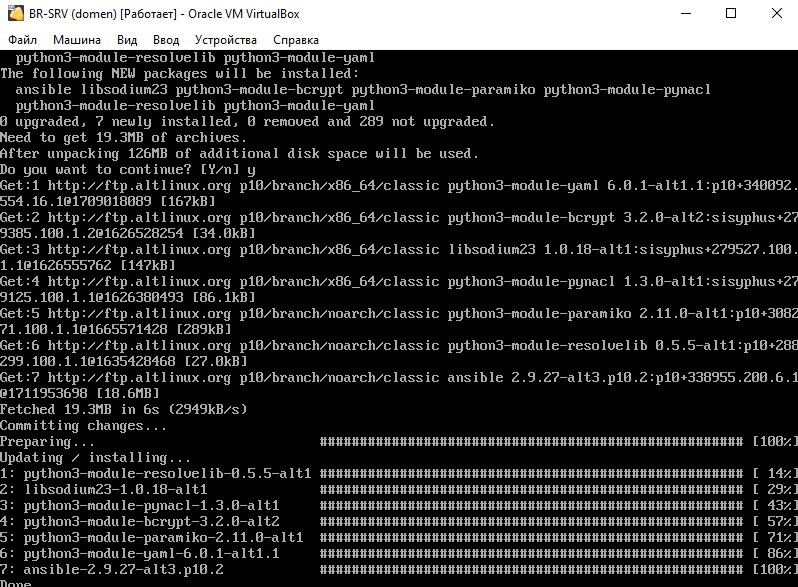
Но помните, что **NTP** для **BR-SRV** – это внешний IP-адрес **HQ-RTR**, то-есть 172.16.4.2.

## Сконфигурируйте ansible на сервере BR-SRV

Для начала проверим, обновлены ли у нас списки пакетов и затем попробуем установить **ansible**:

**apt-get update**

**apt-get install ansible**

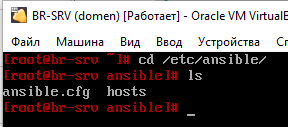


Далее нам нужен рабочий каталог для нашего ansible, который может быть уже создан, НО если нет, то создадим его следующей командой:

**mkdir -p /etc/ansible**

У нас он уже создан, и там же находится нужный нам файл **hosts**, тоже уже созданный, НО в случае его отсутствия, нужно также его создать:

**mcedit /etc/ansible/hosts**



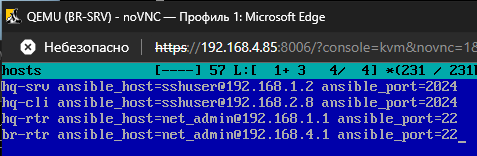
Теперь нам нужно написать следующие строки в файл **hosts**:

**hq-srv ansible\_host=sshuser@192.168.1.2 ansible\_port=2024**

**hq-cli ansible\_host=sshuser@192.168.2.5 ansible\_port=2024**

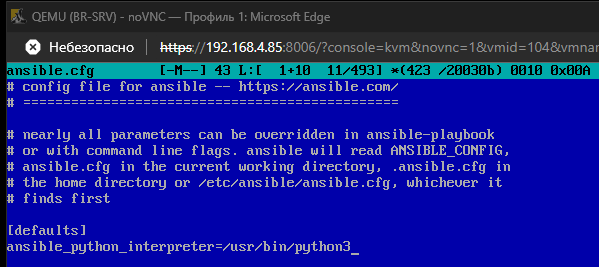
**hq-rtr ansible\_host=net\_admin@192.168.1.1 ansible\_port=2024**

**br-rtr ansible\_host=net\_admin@192.168.4.1 ansible\_port=2024**



И настроим в каталоге **/etc/ansible** файл **ansible.cfg**, добавив под строку **[defaults]** ещё одну:

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

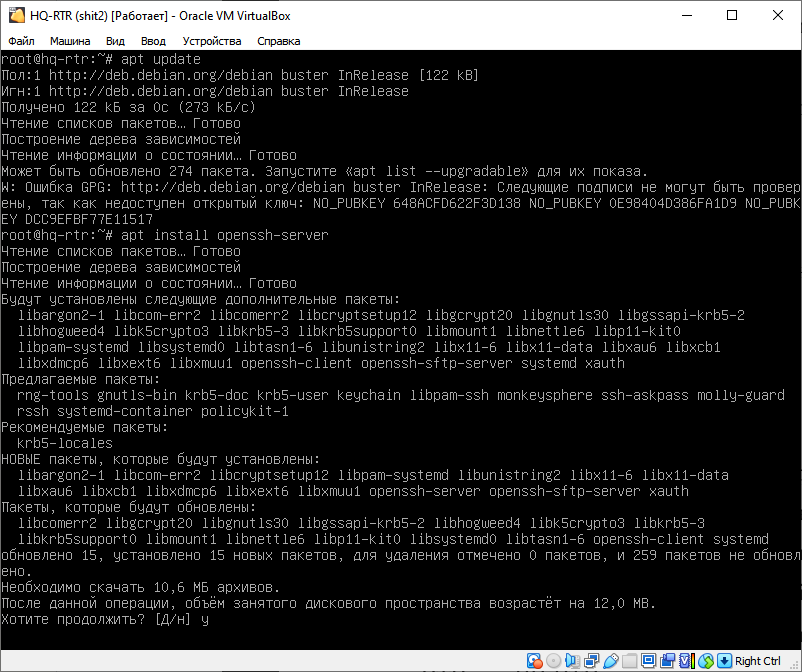
****

Но, возникает другая проблема, в первом модуле мы настраивали SSH на серверах, однако маршрутизаторы **HQ-RTR**, **BR-RTR** и клиент **HQ-CLI** не входили в пункт по настройке. Теперь его нужно настроить сейчас на этих устройствах, поэтому приступаем к их настройке, чтобы мы могли выполнить задание по ansible.

Первым делом настроим **HQ-RTR**, в ней немного отличается пакет и путь к конфигу, но в остальном всё тоже самое, сейчас увидите:

**apt update**

**apt install ssh-server**

****

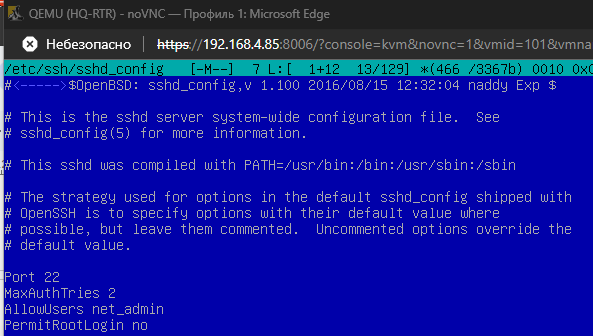
Теперь заходим в конфигурационный файл **/etc/ssh/sshd\_config** и вносим в него следующие строки:

**Port 22**

**MaxAuthTries 2**

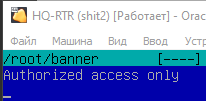
**AllowUsers net\_admin**

**PermitRootLogin no**

****

Ну и также можно создать баннер в **/root/banner**, как тогда у серверов:

**Authorized access only (**не забываем ENTER после этого предложения**)**

****

Ну и вернёмся в конфиг, внесем в строку с баннером путь до нашего файла:

**Banner /root/banner**

****

**systemctl enable --now ssh**

**systemctl restart ssh**

Так как [пользователей](#bookmark=id.l58rd61om01v) на роутерах мы создавали ещё в первом модуле, то можно переходить к следующему этапу.

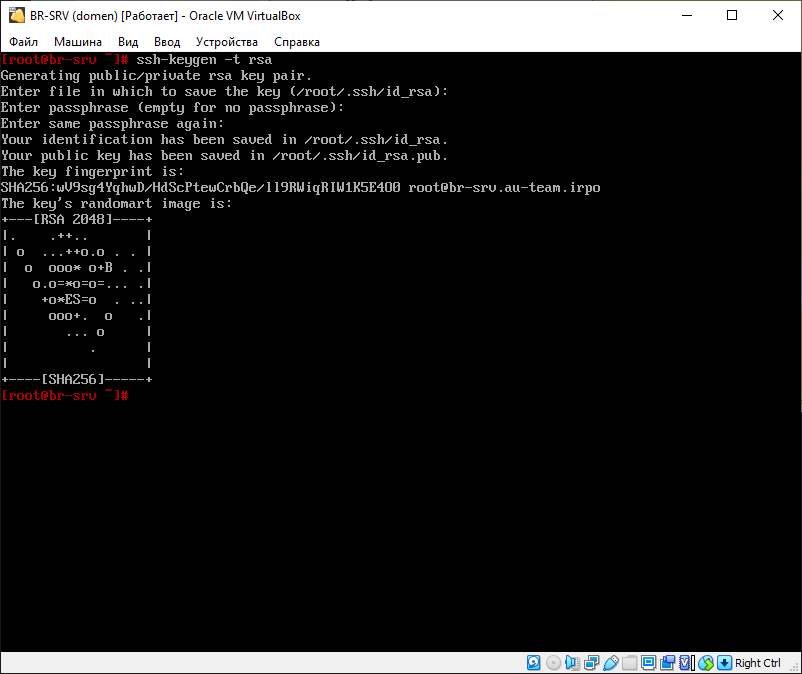
Проделываем тоже самое с [**BR-RTR**](#bookmark=id.8e29orts1am9).

Как проделали с роутерами, нужно разобраться с **HQ-CLI**, на нём мы не создавали **sshuser**, поэтому создаем его тем же способом, что и на серверах в первом модуле. ([клик](#bookmark=id.ykz3m09it5pr))

А также настраиваем на нём саму службу **ssh**, как на серверах. ([клик](#bookmark=id.4xpz4tll9b5d))

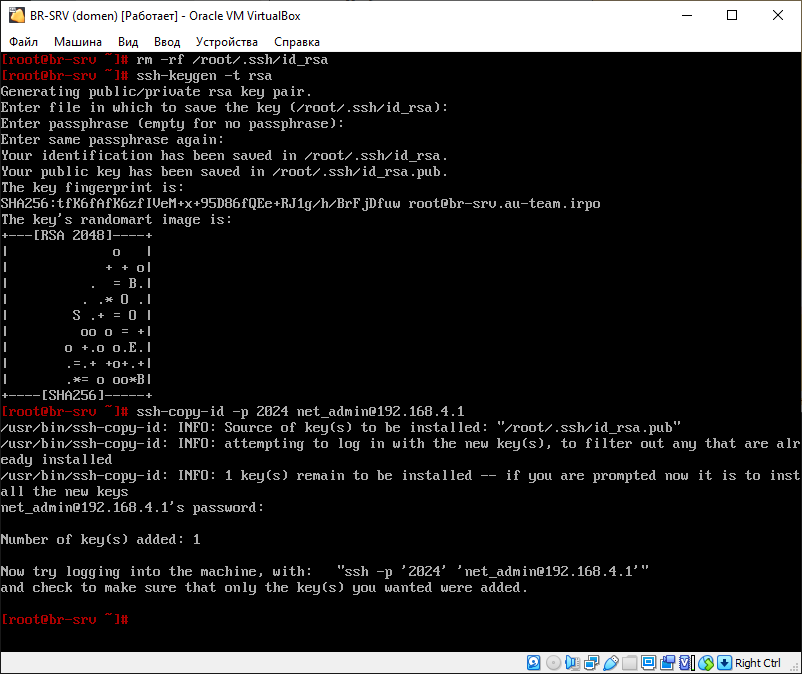
Теперь на **BR-SRV** генерируем ключи RSA, чтобы экспортировать их на машины клиенты, строку с путём и **passphrase** оставляем пустой:

**ssh-keygen -t rsa**



Копируем публичный ключ на клиентские машины, первая из них будет **BR-RTR**:

**ssh-copy-id -p 22 net\_admin@192.168.4.1**



Проделываем для остальных клиентов (**HQ-CLI**, **HQ-SRV**, **HQ-RTR**):

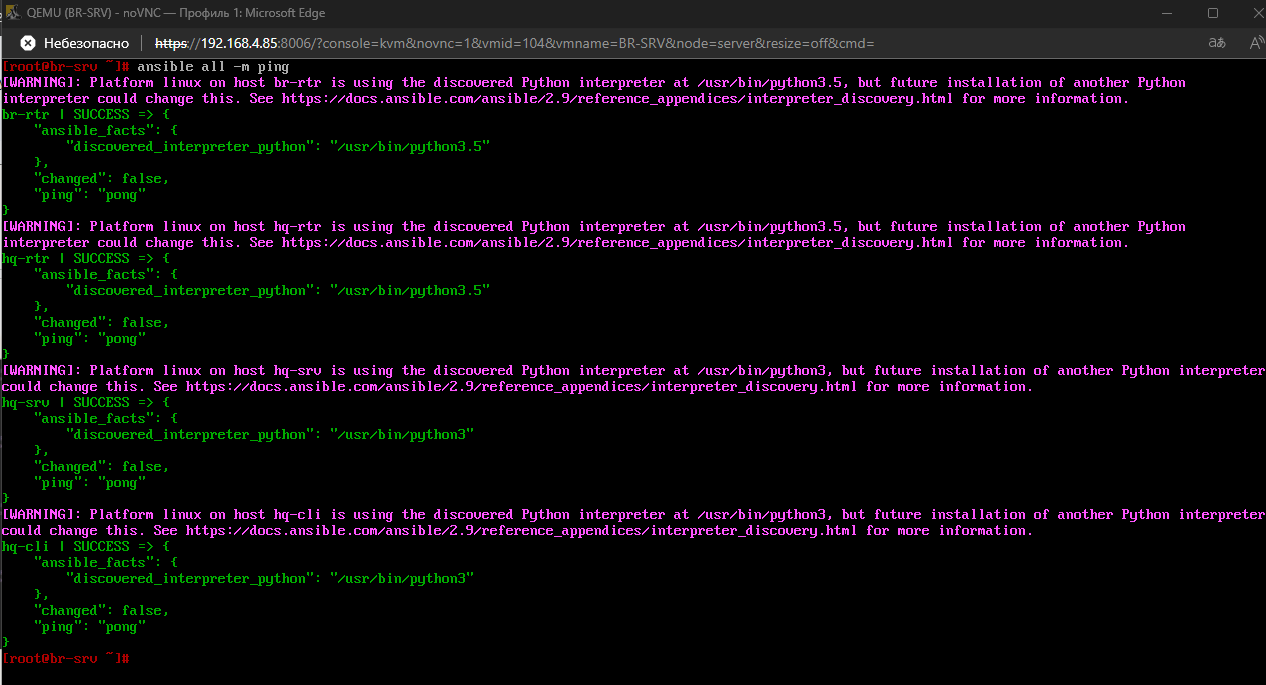
**ssh-copy-id -p 2024 sshuser@192.168.2.5** (Но у вас может быть другой IP, т.к. адрес он получает по DHCP)

**ssh-copy-id -p 2024 sshuser@192.168.1.2**

**ssh-copy-id -p 22 net\_admin@192.168.1.1**

После этого мы можем проверить связь. Машины должны без предупреждений и ошибок отвечать pong на команду ping в ansible посланную с **BR-SRV**:

**ansible all -m ping**



Задание выполнено.