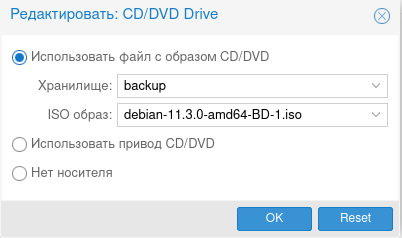
**Виртуальные машины и коммутация.**

Для настройки маршрутизации сети нам понадобится утилита «network-manager». Для ее установки нам нужно примонтировать Bluy-ray cdrom.

Что такое Blu-ray? Это репозитории в виде \*.iso файла,с помощью которых мы можем установить любую утилиту.

Подключаем Blu-ray диск:

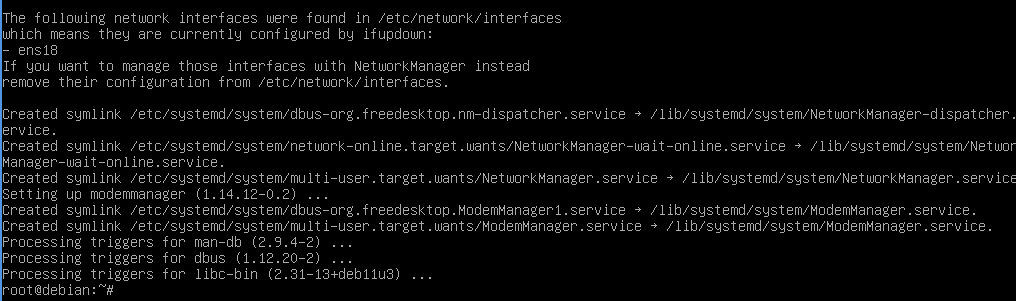


Переходим в консоль и пишем команду «apt-cdrom add»:

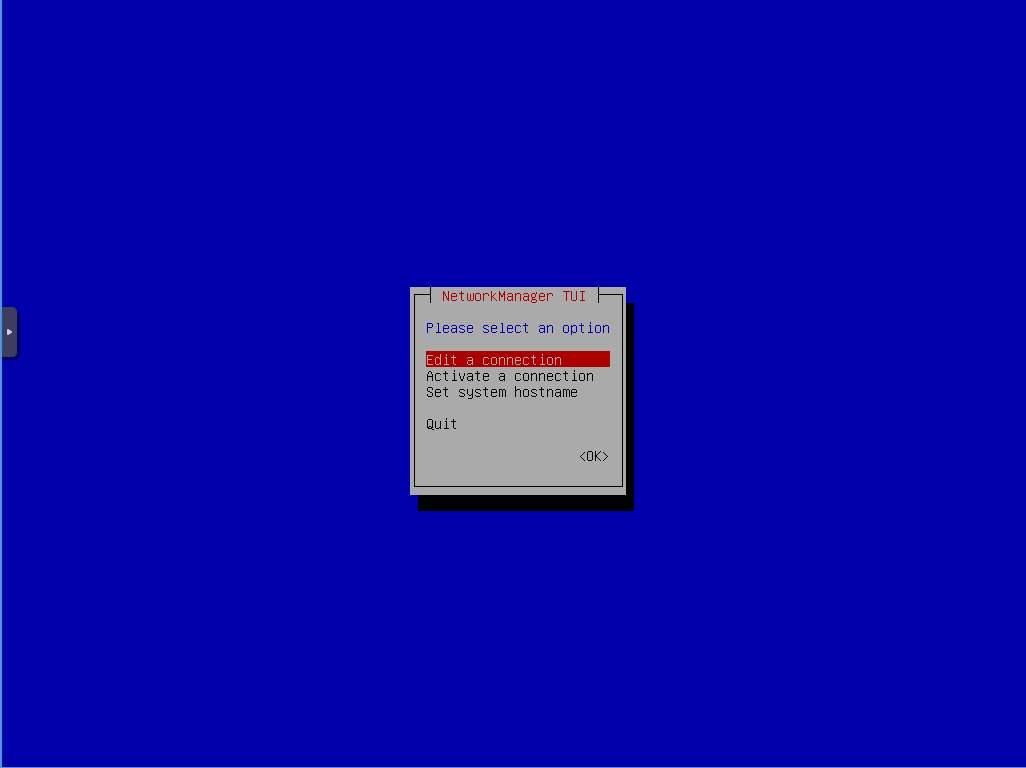


Мы примонтировали первый Blu-Ray нужно примонтировать таким же путем еще 3.

После монтирования всех Blu-ray, можно переходить к настройке. Установим утилиту “apt install -y network-manager”.



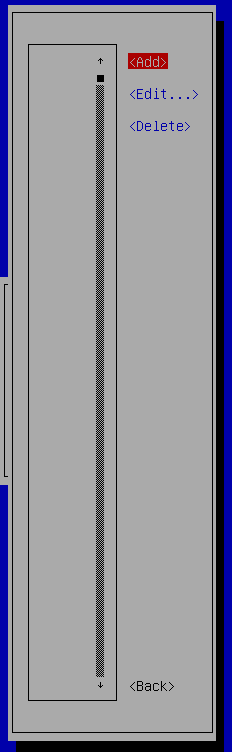
Утилита установилась и можем написать команду “nmtui” и появится псевдо-графический интерфейс.



По заданию нужно изменить имя хоста,для этого нам нужно пролистнуть до пункта “Set system hostname”. В появившемся окне указать нужное нам имя и нажать OK.

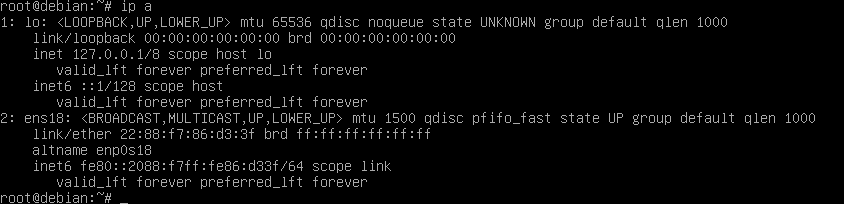


Далее нам нужно установить IP-адрес,переходим в раздел “Edit a connection”.



Мы можем наблюдать что в данном примере нет ни одного интерфейса в “network-manager”. У вас могут быть интерфейсы,но тоже могут быть не все.

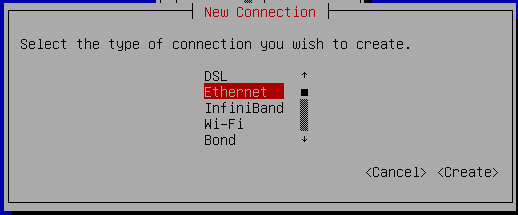
Для добавления интерфейса нужно узнать номер интерфейса,выходим из “network-manager” и прописываем команду “ip a”.



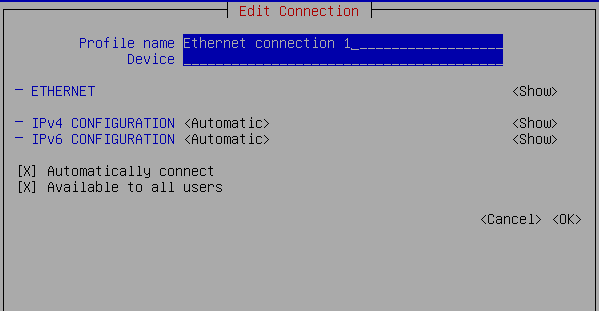
И видим что у нас 2 интерфейса:

1. lo - локальный создается по умолчанию.
2. ens18 - сетевая карта (которая нам нужна)

Мы узнали что нужный нам интерфейс имеет имя “ens18”,заходим обратно в “network-manager”,переходим в раздел управления сетевыми интерфейсами и выбираем пункт “Add”.



Выбираем тип “Ethetnet” и нажимаем “Create”.



Создался чистый профиль.Теперь можем его настроить:

1. Profile name - пишем название профиля
2. Device - пишем название сетевого интерфейса(который мы посмотрели пунктом выше,ens18).

В IPv4 меняем “Automatic” на “Manual” и нажимаем кнопку “Show”.

Открылись пункты для настройки статического IP-адреса:

1. Addresses - указываем IP-адрес,по таблице маршрутизации
2. Gateway - Указываем IP-адрес маршрутизатора(роутера).

ВНИМАНИЕ!!!!!!!!

На машине может быть ТОЛЬКО ОДИН Gateway,независимо сколько интерфейсов.

Секундочка теории по Gateway(Шлюз).

Сетевой шлюз конвертирует протоколы одного типа физической среды в протоколы другой физической среды (сети). Например, при соединении локального [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) с сетью [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) обычно используется сетевой шлюз.

Приведу пример:

Есть 5 человек: Софья,Савелий,Андрей,Анастасия,Алексей.

Андрей знает всех.

Софья знает Алексея.

Савелий знает Анастасию.

Когда Савелий хочет обратиться к Анастасии ,он просто обращается к ней напрямую ведь он ее знает.

А вот для того чтобы Савелию обратится к Алексею,ему нужно вначале обратится к Андрею. В этом случае шлюзом будет Андрей,так как он знает всех и может переадресовать запрос.

Теперь разберем маршрутизацию сети задания:

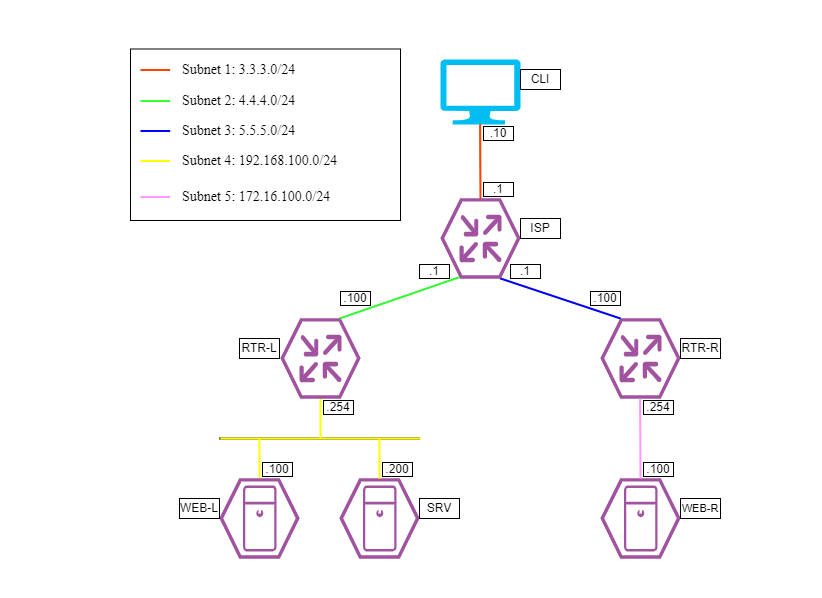
У WEB-L и SRV, шлюз должен быть RTR-L.

У WEB-R шлюз должен быть RTR-R.

У RTR-L и RTR шлюз должен быть ISP.

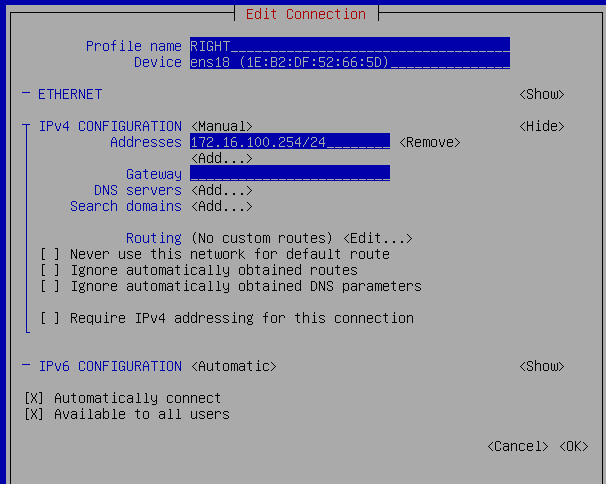
У CLI шлюз ISP.

У ISP не должно быть шлюза,так как он сам является шлюзом.



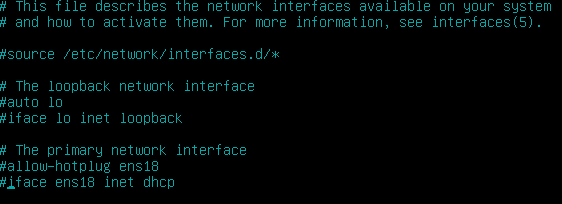
Продолжим настройку.

В итоге должна получиться вот такая картина:



Нажимаем OK и выходим из “network-manager”.

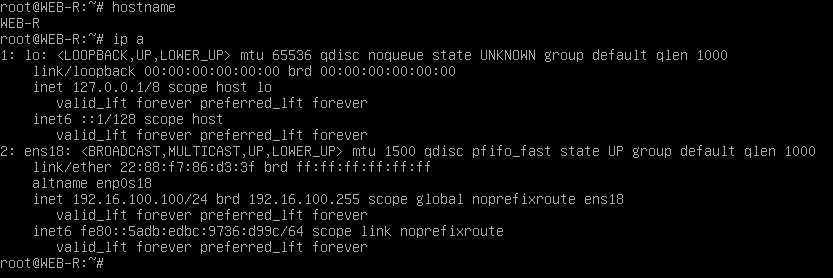
Так как debian использует подключения интернета прямо из ядра,нужно поменять на “network-manager”. Редактируем файл “/etc/network/interfaces”,комментируем все строчки.



Сохраняем и выходим.

После перезагружаем машину командой “reboot”.

Проверяем что все данные заданы верно имя хоста проверяем с помощью команды “hostname”. А назначение IP-адреса с помощью команды “ip a”.

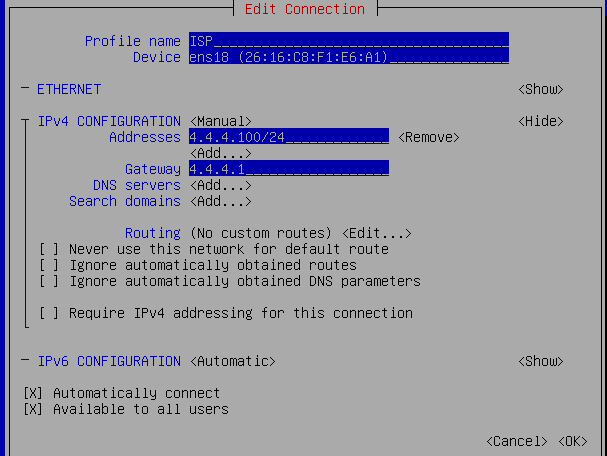


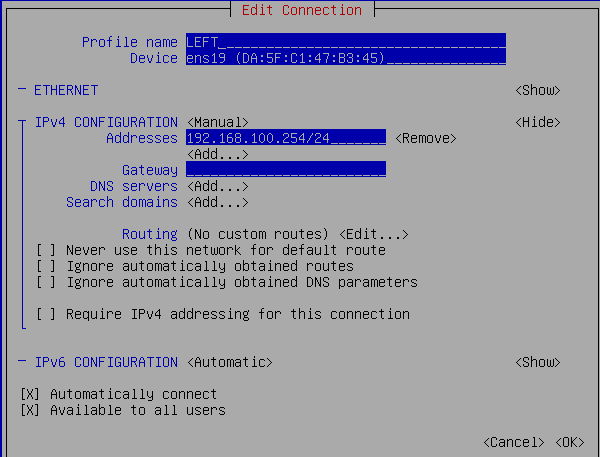
Все данные верны! Теперь нужно повторить настройку для остальных машин.

НЕ ЗАБЫВАЙТЕ,ЧТО У BM: ISP,RTR-L,RTR-R - несколько сетевых интерфейсов.

И GATEWAY МОЖЕТ БЫТЬ ТОЛЬКО ОДИН!!!!

Пример настройки интерфейсов на RTR-L:





**Сетевая связность.**

Платформы контроля трафика, установленные на границах регионов,

должны выполнять трансляцию трафика, идущего из соответствующих

внутренних сетей во внешние сети стенда и в сеть Интернет.

○Трансляция исходящих адресов производится в адрес платформы,расположенный во внешней сети.

На маршрутизаторах(RTR-L,RTR-R) нужно настроить технологию NAT.

NAT RTR-R:

Устанавливаем и настраиваем утилиту FRR “apt install -y frr”.

“apt install -y frr && sed -i -r 's/^(zebra|ospfd)=.+/\1=yes/' /etc/frr/daemons && systemctl restart frr”

Устанавливаем утилиту “apt install -y iptables”.

# ens18 — Внутренняя сеть(172.16.100.0/24)

# ens19 — Внешняя сеть (ISP)

iptables -A FORWARD -i ens18 -o ens19 -j ACCEPT

iptables -t nat -A POSTROUTING -o ens19 -j MASQUERADE

iptables -A FORWARD -i ens19 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -I INPUT -i ens19 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -P INPUT DROP

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -i ens18 -j ACCEPT

iptables -A INPUT -i gre1 -j ACCEPT

iptables -A INPUT -i ens19 -p gre -j ACCEPT

iptables -A INPUT -i ens19 -p icmp -j ACCEPT

iptables -A INPUT -i ens19 -p 53 -j ACCEPT

iptables -A INPUT -i ens19 -p 80 -j ACCEPT

iptables -A INPUT -i ens19 -p tcp –dport 443 -j ACCEPT

iptables -A INPUT -i ens19 -p 47 -j ACCEPT

iptables -A INPUT -i ens19 -p 22 -j ACCEPT

iptables -A INPUT -i ens19 -p esp -j ACCEPT

Сохранение iptables:

Создаем файл

nano -w /etc/network/if-up.d/00-iptables

Вписоваем в него

#!/bin/sh

iptables-restore < /etc/firewall.conf

Сохраняем файл

Выдаем права на запуск

chmod +x /etc/network/if-up.d/00-iptables

Сохраняем конфигурацию

iptables-save >/etc/firewall.conf

Включаем маршрутизацию

**sysctl -w net.ipv4.ip\_forward=1 >> /etc/sysctl.conf**

NAT RTR-L:

Устанавливаем и настраиваем утилиту FRR “apt install -y frr”.

“apt install -y frr && sed -i -r 's/^(zebra|ospfd)=.+/\1=yes/' /etc/frr/daemons && systemctl restart frr”

# ens18 — Внутренняя сеть(192.168.100.0/24)

# ens19 — Внешняя сеть (ISP)

iptables -A FORWARD -i ens18 -o ens19 -j ACCEPT # разрешаем переадресацию с внутреннего порта на внешний

iptables -t nat -A POSTROUTING -o ens19 -j MASQUERADE # Создаем правила нат

iptables -A FORWARD -i ens19 -m state --state ESTABLISHED,RELATED -j ACCEPT # разрешаем внешний ответ

iptables -I INPUT -i ens19 -m state --state ESTABLISHED,RELATED -j ACCEPT # разрешаем внешний ответ

iptables -P INPUT DROP # запрещаем все приемы

iptables -A INPUT -i lo -j ACCEPT # разрешаем все для интерфейса lo

iptables -A INPUT -i ens18 -j ACCEPT # разрешаем все для интерфейса ens18

iptables -A INPUT -i gre1 -j ACCEPT # разрешаем все для интерфейса gre1

iptables -A INPUT -i ens19 -p gre -j ACCEPT

iptables -A INPUT -i ens19 -p icmp -j ACCEPT

iptables -A INPUT -i ens19 -p 53 -j ACCEPT

iptables -A INPUT -i ens19 -p 80 -j ACCEPT

iptables -A INPUT -i ens19 -p tcp –dport 443 -j ACCEPT

iptables -A INPUT -i ens19 -p 47 -j ACCEPT

iptables -A INPUT -i ens19 -p 22 -j ACCEPT

iptables -A INPUT -i ens19 -p tcp –dport 2222 -j ACCEPT

iptables -A INPUT -i ens19 -p tcp –dport 2244 -j ACCEPT

iptables -A INPUT -i ens19 -p esp -j ACCEPT

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 2222 -j DNAT --to 192.168.100.100:22

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 2244 -j DNAT --to 172.16.100.100:22

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 53 -j DNAT --to 192.168.100.200:53

iptables -t nat -A PREROUTING -i ens19 -p udp --dport 53 -j DNAT --to 192.168.100.200:53

Сохранение iptables:

Создаем файл

nano -w /etc/network/if-up.d/00-iptables

Вписоваем в него

#!/bin/sh

iptables-restore < /etc/firewall.conf

Сохраняем файл

Выдаем права на запуск

chmod +x /etc/network/if-up.d/00-iptables

Сохраняем конфигурацию

iptables-save >/etc/firewall.conf

Включаем маршрутизацию

**sysctl -w net.ipv4.ip\_forward=1 >> /etc/sysctl.conf**

Между платформами должен быть установлен защищенный туннель,

позволяющий осуществлять связь между регионами с применением

внутренних адресов.

○ Трафик, проходящий по данному туннелю, должен быть защищен:

■ Платформа ISP не должна иметь возможности просматривать содержимое пакетов, идущих из одной внутренней сети в другую.

○ Туннель должен позволять защищенное взаимодействие между

платформами управления трафиком по их внутренним адресам

■ Взаимодействие по внешним адресам должно происходит без применения туннеля и шифрования.

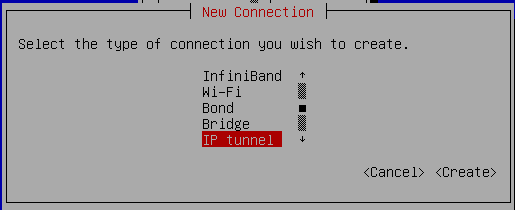
○ Трафик, идущий по туннелю между регионами по внутренним

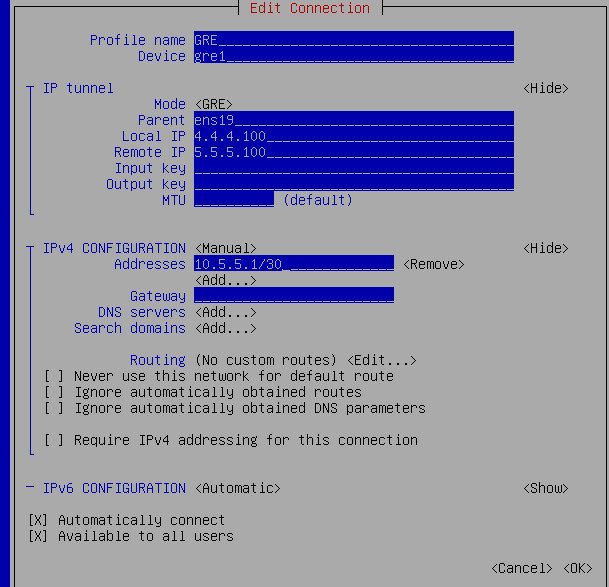
адресам, не должен транслироваться.

**Настройка GRE+IPSEC туннеля.**

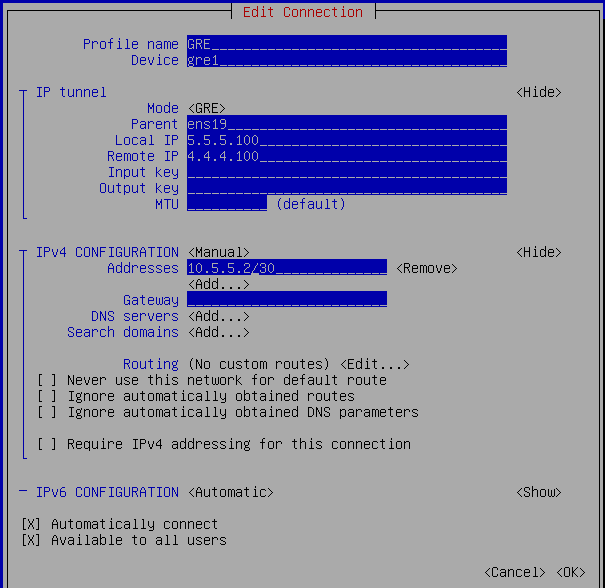
**RTR-L:**

Заходим в “network-manager”,создаем новый интерфейс.



****

**RTR-R:**

****

**IPSec:**

**RTR-R:**

Установка libreswan:

apt install -y libreswan

Создаем конфигурационный файл и заполняем его:

nano /etc/ipsec.d/ipsec.conf

“ conn encrypted\_gre

auto=start

authby=secret

type=tunnel

ike=3des-sha1;modp2048

phase2=esp

phase2alg=aes-sha2;modp2048

left=5.5.5.100

leftprotoport=gre

right=4.4.4.100

rightprotoport=gre ”

Left - local ip

Right - remote ip

Создаём общий ключ для аутентификации:

nano /etc/ipsec.d/gre.secrets

Вначале пишем Locale IP потом Remote IP и ключ в кавычках

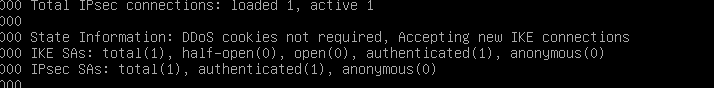
5.5.5.100 4.4.4.100 : PSK “DEMO2022”

Перезапускаем

ipsec restart

смотрим статус

ipsec status



Ставим на автозапуск

systemctl enable ipsec

RTR-L:

Установка libreswan:

apt install -y libreswan

Создаем конфигурационный файл и заполняем его:

nano /etc/ipsec.d/ipsec.conf

“ conn encrypted\_gre

auto=start

authby=secret

type=tunnel

ike=3des-sha1;modp2048

phase2=esp

phase2alg=aes-sha2;modp2048

left=4.4.4.100

leftprotoport=gre

right=5.5.5.100

rightprotoport=gre ”

Left - local ip

Right - remote ip

Создаём общий ключ для аутентификации:

nano /etc/ipsec.d/gre.secrets

Вначале пишем Locale IP потом Remote IP и ключ в кавычках

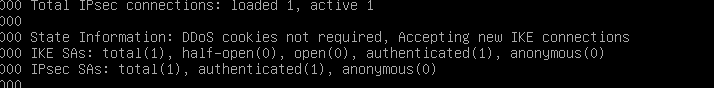
4.4.4.100 5.5.5.100 : PSK “DEMO2022”

Перезапускаем

ipsec restart

смотрим статус

ipsec status

Ставим на автозапуск

systemctl enable ipsec

Настройка внутренней маршрутизации GRE:

**RTR-R:**

“vtysh”

“conf t”

“ip route 192.168.100.0/24 10.5.5.1”

“do wr”

**RTR-L:**

“vtysh”

“conf t”

“ip route 172.16.100.0/24 10.5.5.2”

“do wr”

**Инфраструктурные службы**

В рамках данного модуля необходимо настроить основные

инфраструктурные службы и на

строить представленные ВМ на применение этих

служб для всех основных функций.

●Выполните настройку первого уровня DNS системы стенда:

○Используется ВМ ISP;

○Обслуживается зона demo.wsr.

■Наполнение зоны должно быть реализовано в соответствии с Таблицей 2;

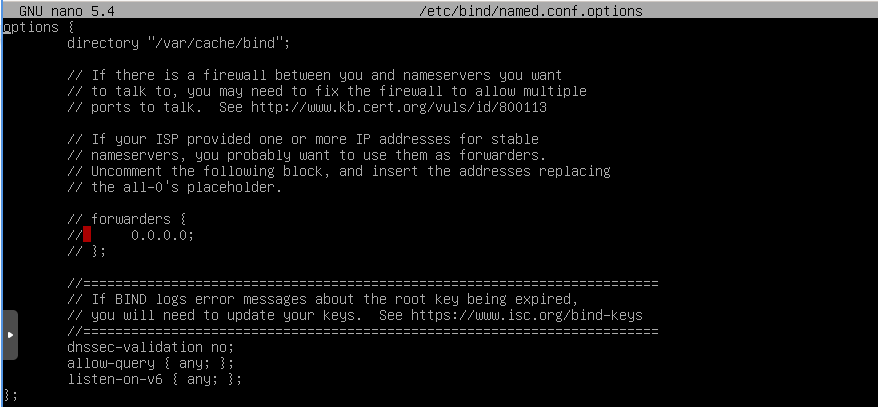
ISP:

устанавливаем bind9:

apt install -y bind9

После установке редактируем конфигурационный файл:

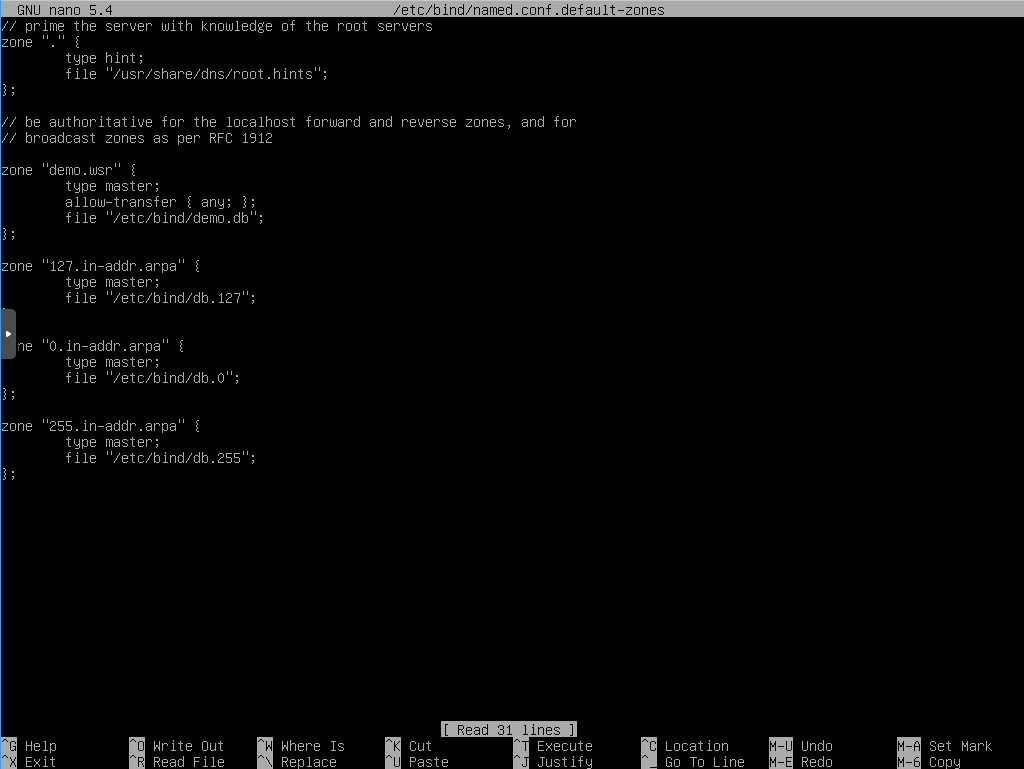
nano /etc/bind/named.conf.options



Сохраняем изменения

Создаем зону в файле:

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



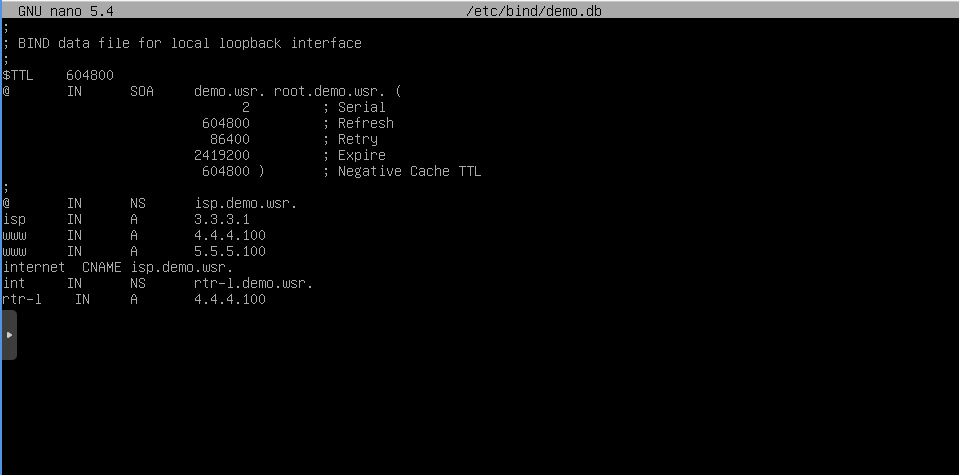
Сохраняем файл и копируем файл самой зоны:

cp /etc/bind/db.local /etc/bind/demo.db

Заходим в него

nano /etc/bind/demo.db

Редактируем согласно таблицы 2:



Сохраняем изменения, перезапускаем сервер

systemctl restart bind9

Первичный DNS успешно настроен.

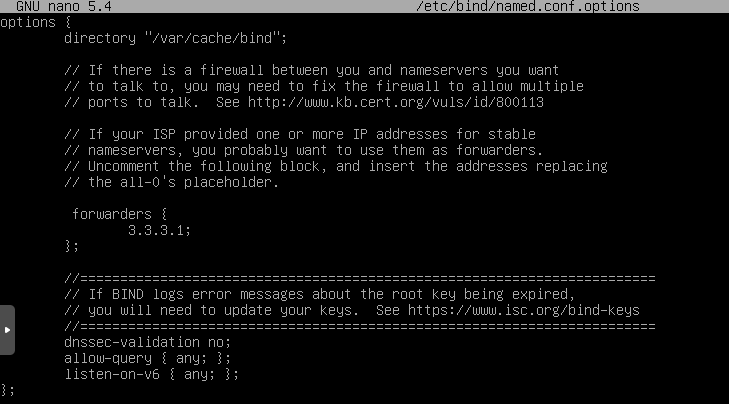
SRV:

устанавливаем bind9:

apt install -y bind9

После установке редактируем конфигурационный файл:

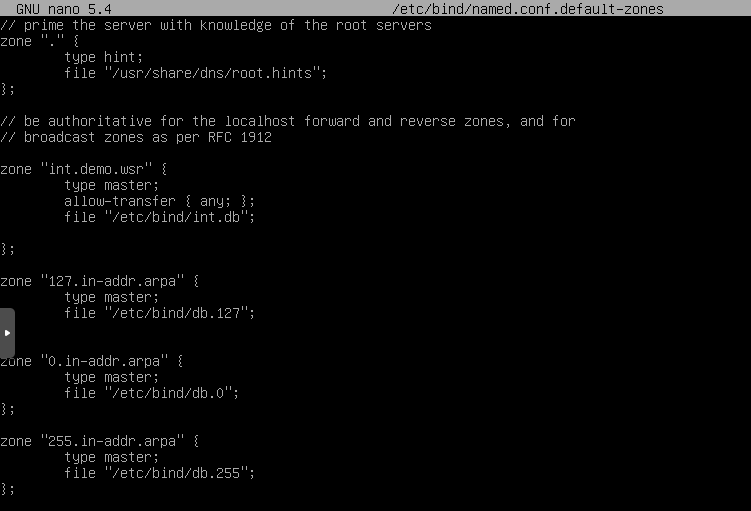
nano /etc/bind/named.conf.options



Сохраняем изменения

Создаем зону в файле:

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



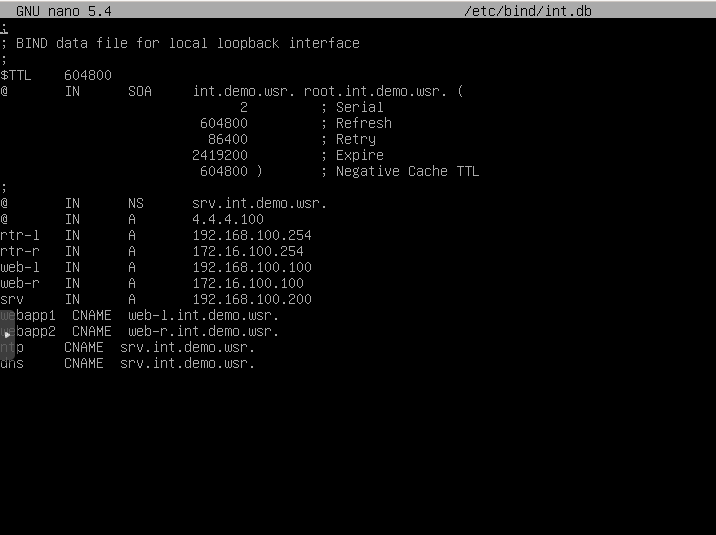
Сохраняем файл и копируем файл самой зоны:

cp /etc/bind/db.local /etc/bind/int.db

Заходим в него

nano /etc/bind/int.db

Редактируем согласно таблицы 2:



Сохраняем изменения, перезапускаем сервер

systemctl restart bind9

Вторичный DNS успешно настроен.

На хосте CLI указываем dns ISP - 3.3.3.1

На всех внутренних RTR-L, RTR-R, WEB-L, SRV, WEB-R указываем dns SRV:

echo nameserver 192.168.100.200 >> /etc/resolv.conf

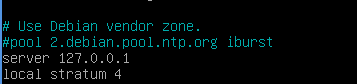
**Настройка первичного сервера времени:**

ISP:

apt install -y chrony

Редактируем файл

nano /etc/chrony/chrony.conf



Сохраняем и перезапускаем службу

systemctl restart chronyd

SRV:

apt install -y chrony

Редактируем файл

nano /etc/chrony/chrony.conf



Сохраняем и перезапускаем службу

systemctl restart chronyd

На всех других внутренних машин:

apt install -y chrony

Редактируем файл

nano /etc/chrony/chrony.conf



Сохраняем и перезапускаем службу

systemctl restart chronyd

На CLI в командной строке от имени админ.

w32tm /config /manualpeerlist:"3.3.3.1,0x8" /syncfromflags:manual /update

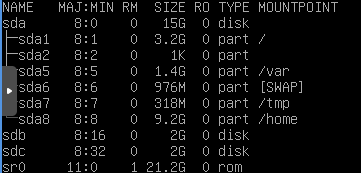
**SMB-сервер:**

**SRV:**

Создаем RAID зеркала,просматриваем жесткие с помощью команды

lsblk

Видим 2 диска по 2 ГБ(ЕСЛИ ЭТО ВАШ СТЕНД НЕ ЗАБЫВАЙТЕ СОЗДАТЬ И ДОБАВИТЬ),запоминаем их название.



В данном примере sdb и sdc.

Далее очищаем диски:

fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0xbb8eba44.

Command (m for help): n # пишем буковку n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): p # пишем буковку p

Partition number (1-4, default 1): # нажимаем enter

First sector (2048-3907029167, default 2048): # нажимаем enter

Last sector, +sectors or +size{K,M,G,T,P} (2048-3907029167, default 3907029167): # нажимаем enter

Created a new partition 1 of type 'Linux' and of size 2 GB.

Command (m for help): t # пишем буковку t

Selected partition 1

Partition type (type L to list all types): L # пишем буковку L

0 Empty 24 NEC DOS 81 Minix / old Lin bf Solaris

1 FAT12 27 Hidden NTFS Win 82 Linux swap / So c1 DRDOS/sec (FAT-

2 XENIX root 39 Plan 9 83 Linux c4 DRDOS/sec (FAT-

3 XENIX usr 3c PartitionMagic 84 OS/2 hidden or c6 DRDOS/sec (FAT-

4 FAT16 <32M 40 Venix 80286 85 Linux extended c7 Syrinx

5 Extended 41 PPC PReP Boot 86 NTFS volume set da Non-FS data

6 FAT16 42 SFS 87 NTFS volume set db CP/M / CTOS / .

7 HPFS/NTFS/exFAT 4d QNX4.x 88 Linux plaintext de Dell Utility

8 AIX 4e QNX4.x 2nd part 8e Linux LVM df BootIt

9 AIX bootable 4f QNX4.x 3rd part 93 Amoeba e1 DOS access

a OS/2 Boot Manag 50 OnTrack DM 94 Amoeba BBT e3 DOS R/O

b W95 FAT32 51 OnTrack DM6 Aux 9f BSD/OS e4 SpeedStor

c W95 FAT32 (LBA) 52 CP/M a0 IBM Thinkpad hi ea Rufus alignment

e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a5 FreeBSD eb BeOS fs

f W95 Ext'd (LBA) 54 OnTrackDM6 a6 OpenBSD ee GPT

10 OPUS 55 EZ-Drive a7 NeXTSTEP ef EFI (FAT-12/16/

11 Hidden FAT12 56 Golden Bow a8 Darwin UFS f0 Linux/PA-RISC b

12 Compaq diagnost 5c Priam Edisk a9 NetBSD f1 SpeedStor

14 Hidden FAT16 <3 61 SpeedStor ab Darwin boot f4 SpeedStor

16 Hidden FAT16 63 GNU HURD or Sys af HFS / HFS+ f2 DOS secondary

17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs fb VMware VMFS

18 AST SmartSleep 65 Novell Netware b8 BSDI swap fc VMware VMKCORE

1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid fd Linux raid auto

1c Hidden W95 FAT3 75 PC/IX bc Acronis FAT32 L fe LANstep

1e Hidden W95 FAT1 80 Old Minix be Solaris boot ff BBT

Partition type (type L to list all types): fd # пишем буковку fd

Changed type of partition 'Linux' to 'Linux raid autodetect'.

Command (m for help): w # пишем буковку w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

Диск sdb готов тоже самое проделываем с sdc.

После готовых двух дисков мы создаем сам raid

mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1

mdadm: Note: this array has metadata at the start and

may not be suitable as a boot device. If you plan to

store '/boot' on this device please ensure that

your boot-loader understands md/v1.x metadata, or use

--metadata=0.90

mdadm: size set to 1953382464K

mdadm: automatically enabling write-intent bitmap on large array

Continue creating array? y # вводим букву y и нажимаем enater.

mdadm: Defaulting to version 1.2 metadata

mdadm: array /dev/md2 started.

RAID успешно создан остается только его отформотировать и сделать авто-монтирование

mkfs.ext4 /dev/md1 # формотирвоание в систему ext4

Делаем запись в файл

echo "DEVICE partitions" > /etc/mdadm/mdadm.conf

mdadm --detail --scan --verbose | awk '/ARRAY/ {print}' >> /etc/mdadm/mdadm.conf

обновляем диски

update-initramfs -u

Монтируем диск

mkdir /mnt/storage

mount /dev/md1 /mnt/storage

Авто-монтирование

nano /etc/fstab

Вписоваем название raid и куда монтируем.

/dev/md1 /mnt/storage ext4 defaults 0 0

После перезагрузки raid успешно будет примонтирован к каталогу /mnt/storage

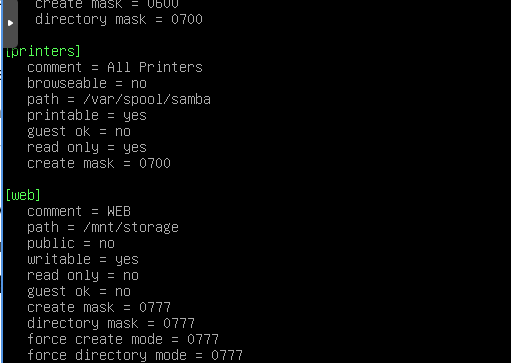
Устанавливаем samba

apt install -y samba

Редактируем файл /etc/samba/smb.conf

Листаем до строчки [printers]

И после пишем следующие



После сохраняем файл и даем права

chmod 777 /mnt/storage

Создаем пользователя

useradd WEB

задаем пароль пользователю

passwd WEB # Задаем пароль P@ssw0rd

создаем пользователя samba

smbpasswd -a WEB # Задаем пароль P@ssw0rd

Перезагружаем сервер

systemctl restart smbd

SAMBA сервер готов, настроим клиентов.

WEB-L:

Устнавливаем

apt install -y cifs-utils

Создаем файл

nano /root/.smbclient

Вписоваем username и password который задали

username=WEB

password=P@ssword

Сохраняем и выходим

Задаем автомонтирование в папку /opt/share

nano /etc/fstab

//srv.int.demo.wsr/web /opt/share cifs user,rw,\_netdev,credentials=/root/.smbclient 0 0

Сохраняем и выходим

Создаем папку

mkdir /opt/share

Выполняем монтирование

mount -a

**OPENSSL**

**SRV:**

**Устанавливаем easy-rsa:**

**sudo apt install easy-rsa**

**Создаем папку согласно заданию:**

**mkdir /var/ca/**

**Создаем символическую ссылку**

**ln -s /usr/share/easy-rsa/\* /var/ca/**

**Переходим в созданную папку и производим инициализацию**

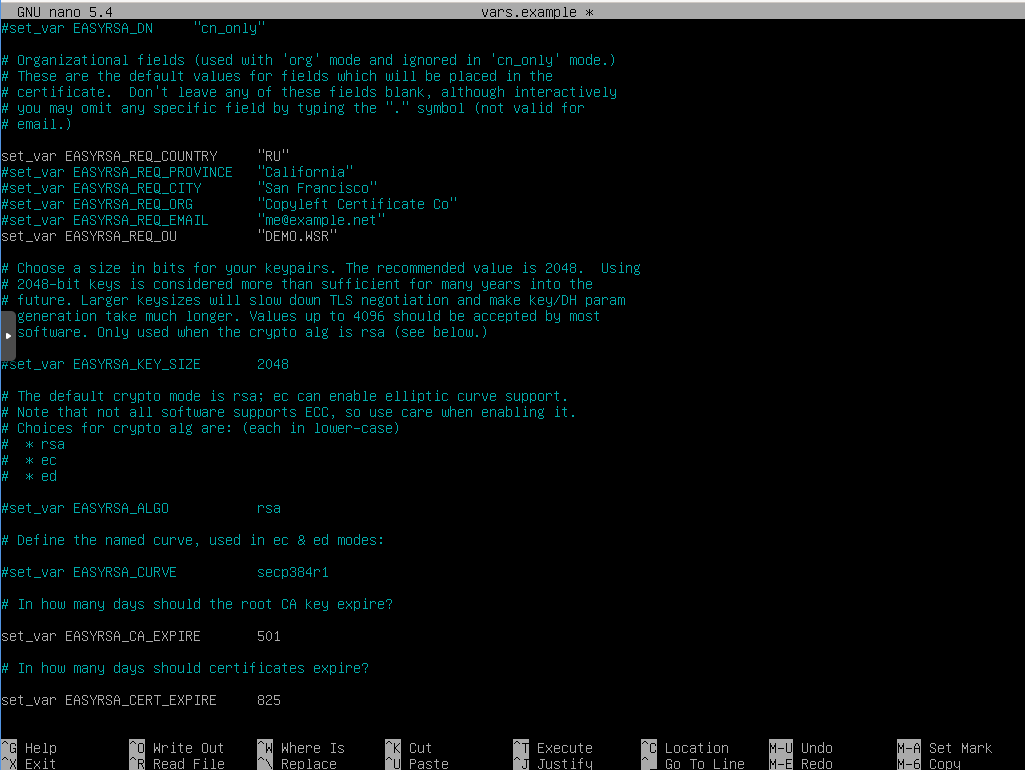
**cd /var/ca/**

**./easyrsa init-pki**

**Редактируем файл конфигурации**

**nano vars.example**

Приводим его к такому виду



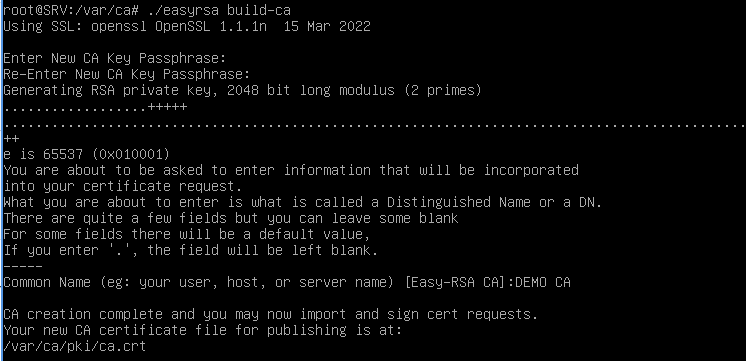
Сохраняем и выходим

выполняем команду компиляции

./easyrsa build-ca

Нужно ввести абсолютно любой пароль, главное не забыть рекомендую использовать P@ssw0rd.

А также ввести CN название,можно любое,но рекомендую DEMO CA.



**Docker:**

**WEB-L:**

**Добавляем в cd-rom образ docker-new.iso.**

**apt-cdrom add**

**apt install -y docker-ce**

**systemctl start docker**

**systemctl enable docker**

**mkdir /mnt/app**

**mount /dev/sr0 /mnt/app**

**docker load < /mnt/app/app.tar**

**docker images**

**docker run --name app -p 8080:80 -d app**

**docker ps**

**WEB-R:**

**Добавляем в cd-rom образ docker-new.iso.**

**apt-cdrom add**

**apt install -y docker-ce**

**systemctl start docker**

**systemctl enable docker**

**mkdir /mnt/app**

**mount /dev/sr0 /mnt/app**

**docker load < /mnt/app/app.tar**

**docker images**

**docker run --name app -p 8080:80 -d app**

**docker ps**

**RTR-L:**

**интерфейс ens19 - внешний**

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 80 -j DNAT --to 192.168.100.100:80

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 443 -j DNAT --to 192.168.100.100:443

Сохраняем конфигурацию

iptables-save >/etc/firewall.conf

**RTR-R:**

**интерфейс ens19 - внешний**

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 80 -j DNAT --to 172.16.100.100:80

iptables -t nat -A PREROUTING -i ens19 -p tcp --dport 443 -j DNAT --to 172.16.100.100:443

Сохраняем конфигурацию

iptables-save >/etc/firewall.conf

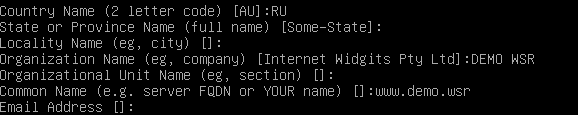
**SSL:**

**SRV:**

Создаем сертификаты [www.key](http://www.key) и www.crt

openssl req -x509 -nodes -days 501 -newkey rsa:2048 -keyout /etc/ssl/private/www.key -out /etc/ssl/certs/[www.crt](http://www.crt)

Задаем следующие параметры

****

Выполняем шифрование

openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048

Переносим сертификаты с SRV на WEB-L и WEB-R не забываем открыть доступ по ssh по пользователю root.

SRV:

**Перед переносом на WEB-L и WEB-R нужно установить nginx.**

**scp** /etc/ssl/private/[www.key](http://www.key) root@192.168.100.100:/etc/nginx/[www.key](http://www.key)

**scp** /etc/ssl/private/[www.](http://www.key)crt root@192.168.100.100:/etc/nginx/[www.crt](http://www.crt)

**scp** /etc/ssl/private/[www.key](http://www.key) root@172.16.100.100:/etc/nginx/[www.key](http://www.key)

**scp** /etc/ssl/private/[www.](http://www.key)crt root@172.16.100.100:/etc/nginx/[www.crt](http://www.crt)

WEB-L:

apt install -y nginx

nano /etc/nginx/snippets/snakeoil.conf



nano /etc/nginx/sites-available/default

В файле нужно вначале либо все удалить либо закомментировать и оставить только следующие строчки:

upstream backend {

server 192.168.100.100:8080 fail\_timeout=25;

server 172.16.100.100:8080 fail\_timeout=25;

}

server {

listen 443 ssl default\_server;

include snippets/snakeoil.conf;

server\_name www.demo.wsr;

location / {

proxy\_pass http://backend ;

}

}

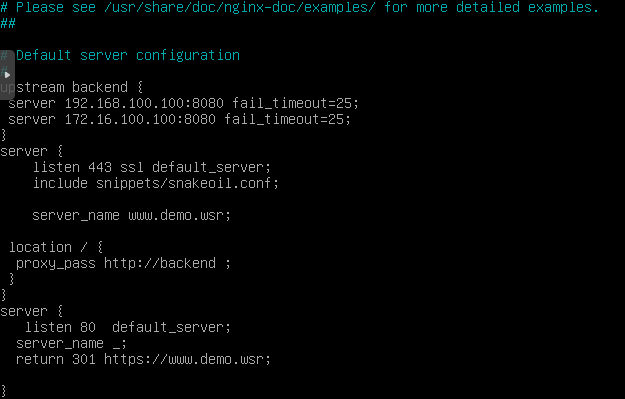
server {

listen 80 default\_server;

server\_name \_;

return 301 https://www.demo.wsr;

}



systemctl reload nginx

WEB-R:

apt install -y nginx

nano /etc/nginx/snippets/snakeoil.conf



nano /etc/nginx/sites-available/default

В файле нужно вначале либо все удалить либо закомментировать и оставить только следующие строчки:

upstream backend {

server 192.168.100.100:8080 fail\_timeout=25;

server 172.16.100.100:8080 fail\_timeout=25;

}

server {

listen 443 ssl default\_server;

include snippets/snakeoil.conf;

server\_name www.demo.wsr;

location / {

proxy\_pass http://backend ;

}

}

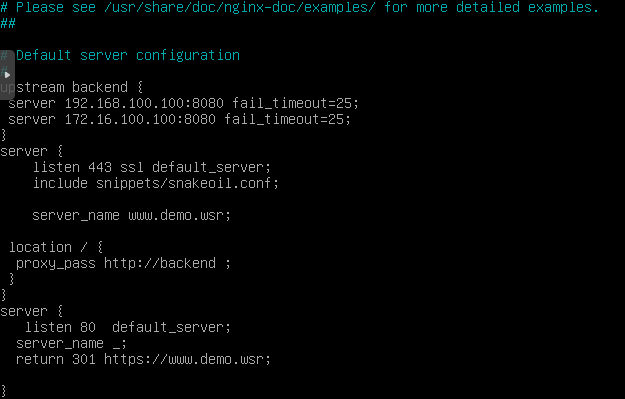
server {

listen 80 default\_server;

server\_name \_;

return 301 https://www.demo.wsr;

}



systemctl reload nginx

CLI:

Переходим в командную строку и пишем следующую строчку:

scp -P 2222 root@4.4.4.100:/etc/nginx/[www.crt](http://www.crt) C:/Users/admin/Desktop

На рабочем столе появится сертификат [www.cr](http://www.crt)t нажимаем на его и устанавливаем по всем параметрам по дефолту.

