

Enp0s3 – делать внутренним интерфейсом

Enp0s8 – Делать внешним интерфейсом, чтобы было удобнее

Включаем на всех

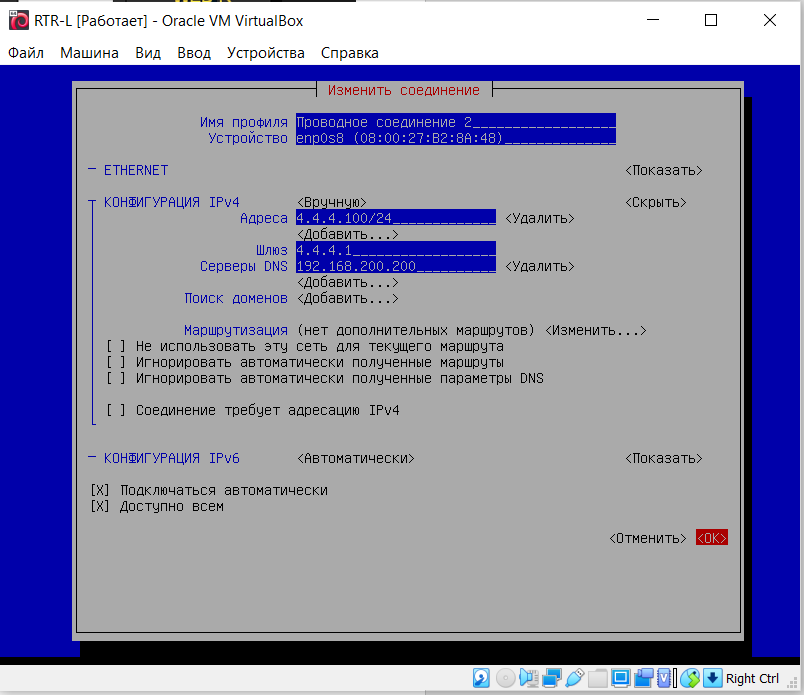
RTR-L

#apt install -y network-manager firewalld wireguard wireguard-tools

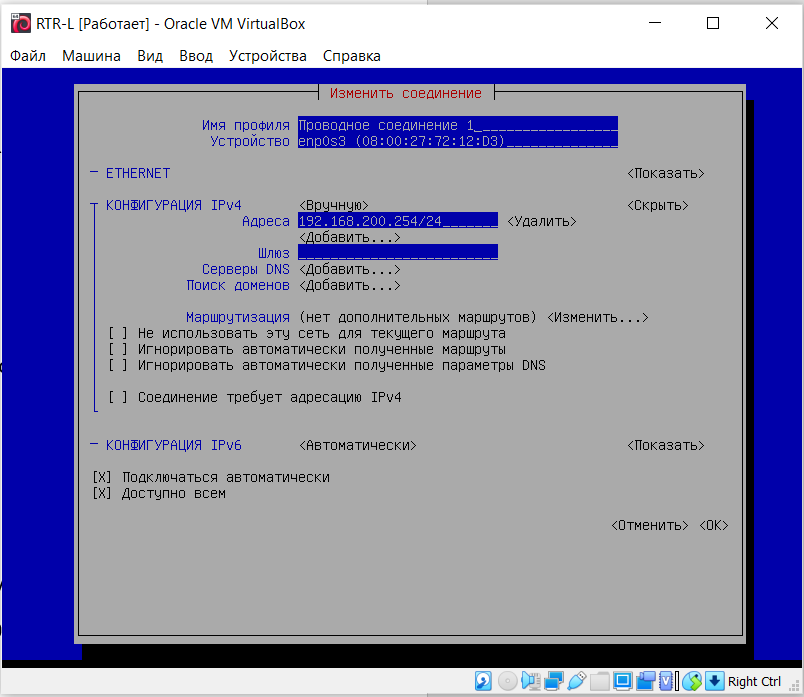
#nmtui

Переставить на всех с Автоматически на Ручную на всех ВМ

int WiredConnection 2 (Enp0s8) ip 4.4.4.100/24, gateway 4.4.4.1 DNS 192.168.100.200



int WiredConnection 1 (Enp0s3) ip 192.168.100.254/24



hostname RTR-L

#reboot

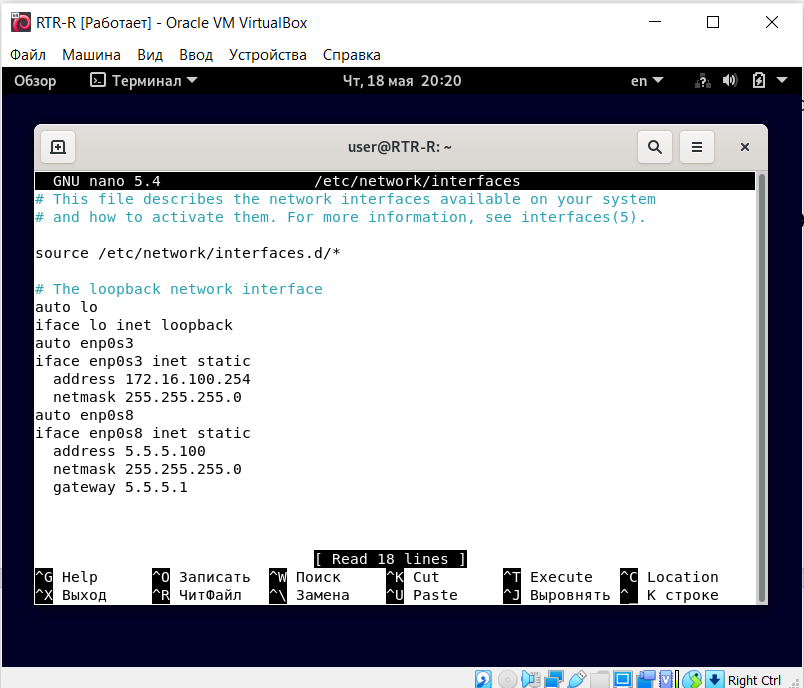
RTR-R С графическим интерфейсом

#apt install -y network-manager openssh-server firewalld wireguard wireguard-tools

# nano /etc/network/interfaces

int WiredConnection 1 (Enp0s8) ip 5.5.5.100/24, gateway 5.5.5.1 DNS 4.4.4.100

int WiredConnection 2 (Enp0s3) ip 172.16.100.254/24



hostname RTR-R

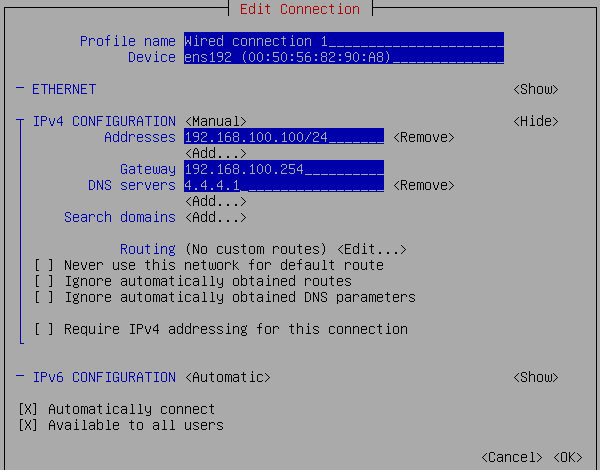
#reboot

WEB-L

#apt install -y network-manager chrony openssh-server nginx lyxn cifs-utils

#nmtui

int WiredConnection 1 (Enp0s3) ip 192.168.100.100/24, gateway 192.168.100.254, DNS 192.168.100.200



hostname WEB-L

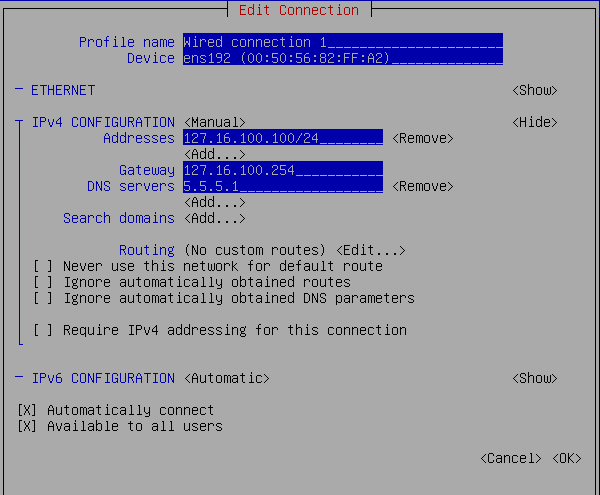
#reboot

WEB-R

#apt install -y network-manager chrony openssh-server nginx lyxn cifs-utils

#nmtui

int WiredConnection 1 (Enp0s3) ip 172.16.100.100/24, gateway 172.16.100.254, DNS 4.4.4.100



hostname WEB-R

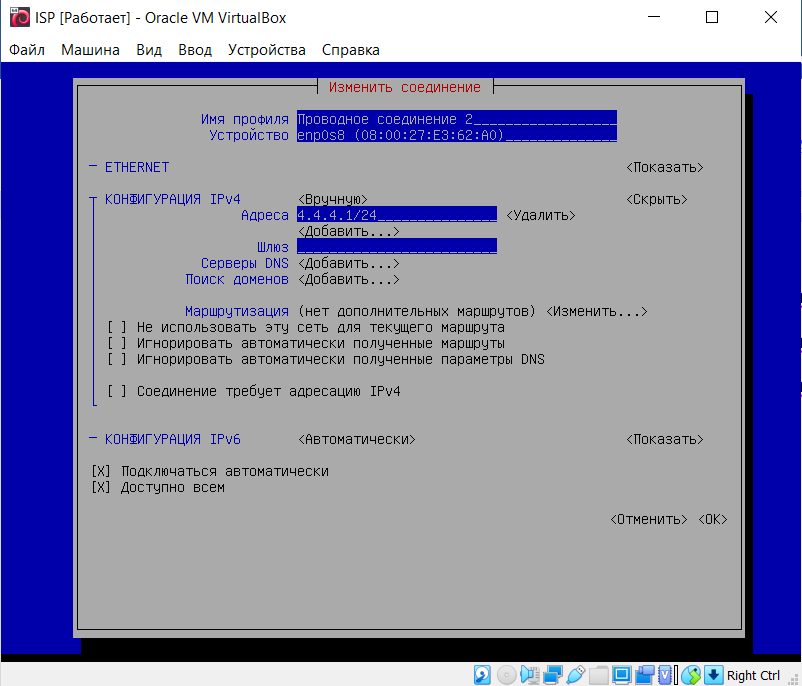
#reboot

ISP Enp0s3 – Центральный Enp0s8 – Левый Enp0s9 - Правый

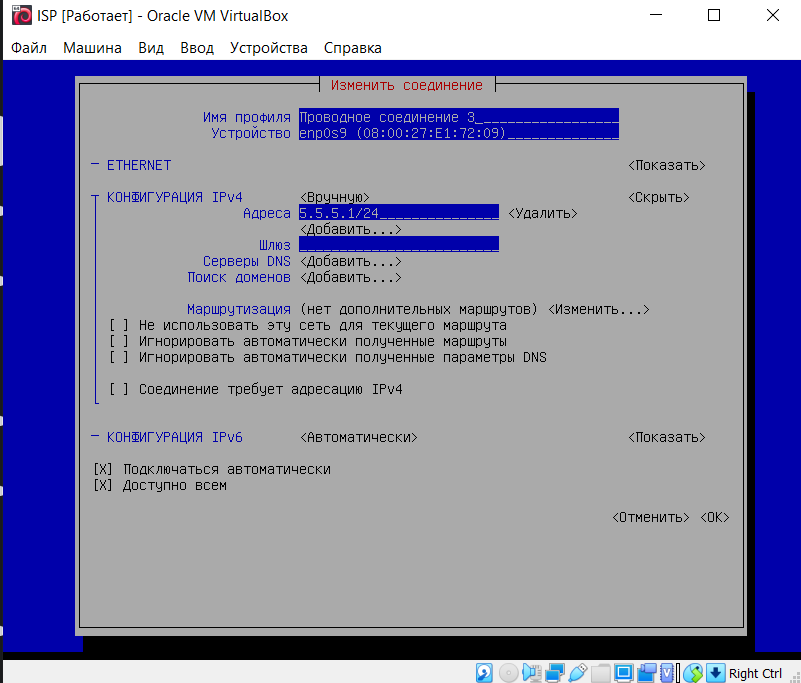
#apt install -y network-manager bind9 chrony dnsutils openssh-server bind9utils

#nmtui

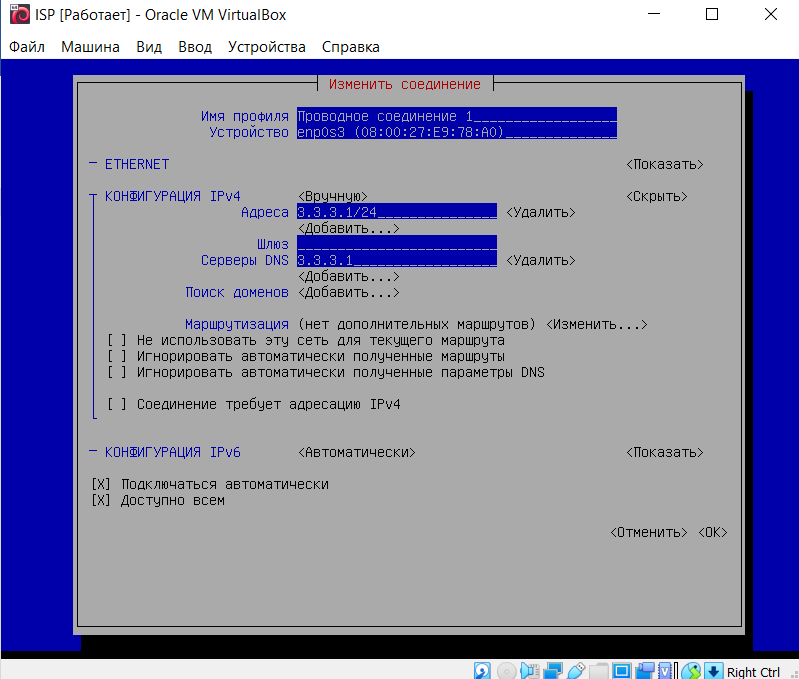
int WiredConnection 1 (enp0s8) ip 4.4.4.1/24



int WiredConnection 2 (enp0s9) ip 5.5.5.1/24



int WiredConnection 3 (enp0s3) ip 3.3.3.1/24, DNS 3.3.3.1



ISP, RTR-R, RTR-L

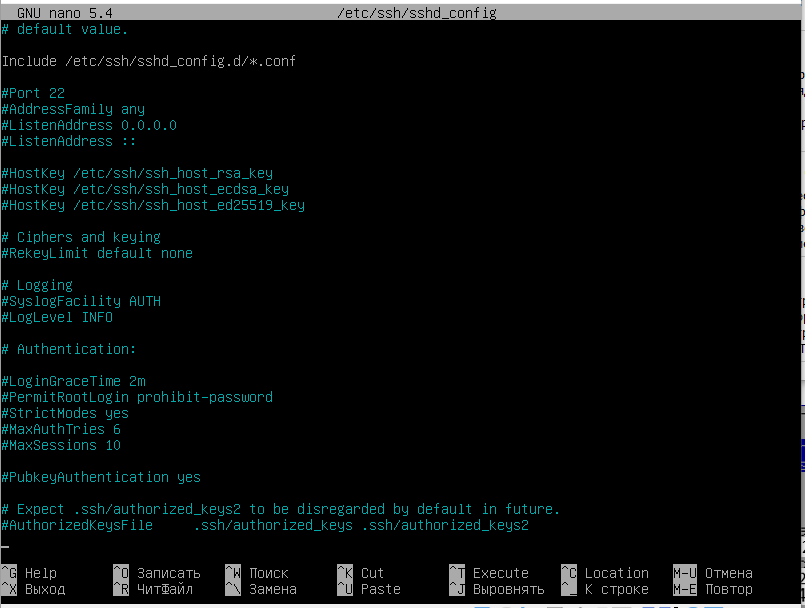
nano /etc/sysctl.conf

net.ipv4.ip\_forward=1

WEB-L, WEB-R, RTR-R

Настройка SSH

nano /etc/ssh/sshd\_config



Меняем в PermitBootLogin на yes и убираем #

RTR-R, RTR-L Установка Firewalld

apt install -y firewalld

Настройка

Удалить из public все идентификаторы

firewall-cmd --zone=public --remove-interface=Название интерфейса

firewall-cmd --zone=trusted --add-interface=Внутренний интерфейс

firewall-cmd --zone=external --add-interface=Внешний интерфейс

firewall-cmd --zone=external --add-service= http

firewall-cmd --zone=external --add-service= https

firewall-cmd --zone=external --add-service= dns

firewall-cmd --zone=external --add-service= ssh

firewall-cmd --zone=external --add-port= 12345/udp

RTR-L

firewall-cmd --zone=external --add-forward-port=port=2222:proto=tcp:toport=22:toaddr=192.168.100.100

firewall-cmd --zone=external --add-forward-port=port=80:proto=tcp:toport=80:toaddr=192.168.100.100

firewall-cmd --zone=external --add-forward-port=port=53:proto=udp:toport=53:toaddr=192.168.100.100-IP WEB-L

RTR-R

firewall-cmd --zone=external --add-forward-port=port=2244:proto=tcp:toport=22:toaddr=172.16.100.100-IP WEB-R

firewall-cmd --zone=external --add-forward-port=port=80:proto=tcp:toport=80:toaddr=172.16.100.100

На обоих пишется для сохранения изменений

firewall-cmd --runtime-to-permament

Перезапускаем файерволл

firewall-cmd --reload

Проверяем работу ssh

RTR--R

ssh [root@4.4.4.100](mailto:root@4.4.4.100) -p 2222

RTR--L

ssh [root@5.5.5.100](mailto:root@5.5.5.100) -p 2244

Создание защищенного туннеля между RTR-R и RTR-L

apt install -y wireguard wireguard-tools

Создаём директиву

mkdir /etc/wireguard/keys

cd /etc/wireguard/keys

Генерируем закрытый и открытый ключ

Делаем это на RTR-L

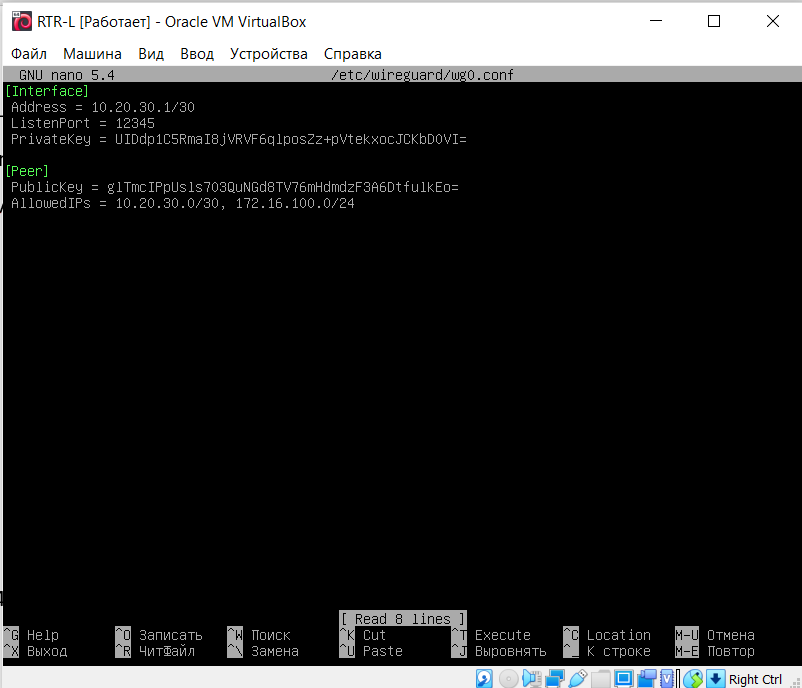
wg genkey | tee srv-sec.key | wg pubkey > srv-pub.key

wg genkey | tee cli-sec.key | wg pubkey > cli-pub.key

Переносим сгенерируемый ключ в wg0.conf

cat srv-sec.key cli-pub.key >> /etc/wireguard/wg0.conf

Открываем и редактируем wg0.conf

Сохраняем

systemctl enable --now wg-quick@wg0

Делаем это на RTR-R

mkdir /etc/wireguard/keys

cd /etc/wireguard/keys

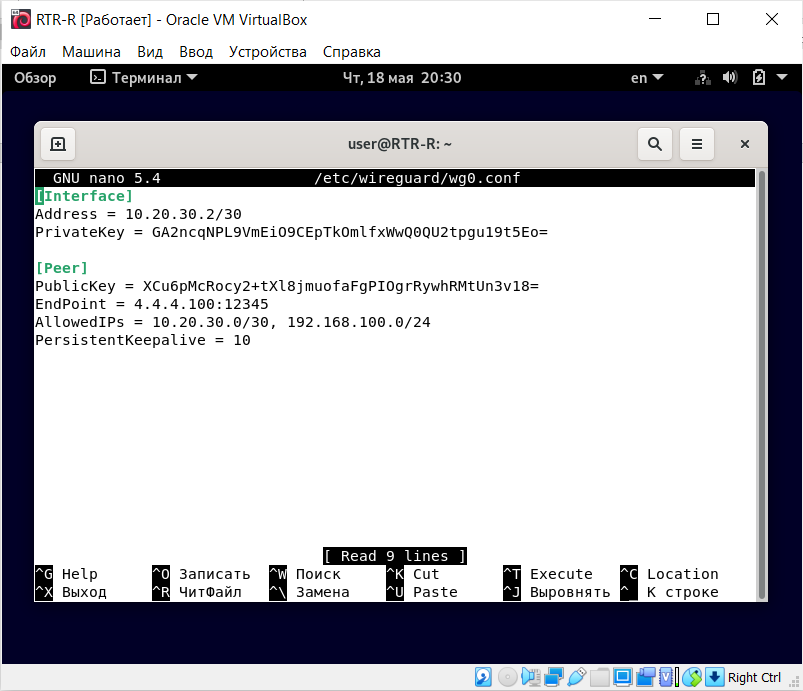
Переносим с RTR-L на RTR-R ключи

scp cli-sec.key srv-pub.key 5.5.5.100:/etc/wireguard/keys

Переносим ключи в wg0.conf

cat cli-sec.key srv-pub.key >> /etc/wireguard/wg0.conf

Открываем и редактируем wg0.conf

Сохраняем

systemctl enable --now wg-quick@wg0

Проверяем

wg show all

Смотрим на transer

