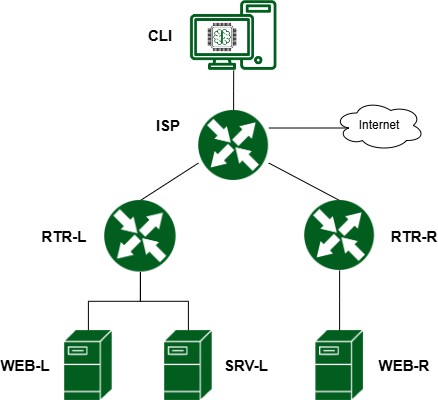
Задание экзамена «Администрирование сетей передачи информации»



1. Имена хостов в созданных ВМ должны быть установлены в соответствии со схемой.

Адресация должна быть выполнена в соответствии с Таблицей 1.

Таблица 1

|  |  |
| --- | --- |
| Имя ВМ | IP-адрес |
| RTR-L | 100.100.100.10/28 – для подсети в сторону ISP 10.10.10.1/24 – для подсети офиса LEFT |
| RTR-R | 150.150.150.10/28 – для подсети в сторону ISP 20.20.20.1/24 – для подсети офиса RIGHT |
| SRV-L | 10.10.10.100/24 |
| WEB-L | 10.10.10.110/24 |
| WEB-R | 20.20.20.100/24 |
| ISP | 100.100.100.1/28 – для подсети в сторону RTR-L 150.150.150.1/28 – для подсети в сторону RTR-R 35.35.35.1/28 – для подсети в сторону CLI DHCP -для выхода в общественную сеть |
| CLI | 35.35.35.10/28 |

**CLI:**

hostnamectl set-hostname cli.au.team; exec bash

vim /etc/net/ifaces/ens19/options меняем на static

vim /etc/net/ifaces/ens19/ipv4address 35.35.35.10/28

vim /etc/net/ifaces/ens19/ipv4route default via 35.35.35.1

reboot

**ISP:**

hostnamectl set-hostname isp.au.team; exec bash

cp -r /etc/net/ifaces/ens18 /etc/net/ifaces/ens19/

vim /etc/net/ifaces/ens19/options меняем на static

cp -r /etc/net/ifaces/ens19 /etc/net/ifaces/ens20/

cp -r /etc/net/ifaces/ens19 /etc/net/ifaces/ens21/

vim /etc/net/ifaces/ens19/ipv4address 100.100.100.1/28

vim /etc/net/ifaces/ens20/ipv4address 150.150.150.1/28

vim /etc/net/ifaces/ens21/ipv4address 35.35.35.1/28

vim /etc/net/ifaces/ens19/ipv4route 10.10.10.0/24 via 100.100.100.10

vim /etc/net/ifaces/ens20/ipv4route 20.20.20.0/24 via 150.150.150.10

reboot

**RTR-L:**

hostnamectl set-hostname rtr-l.au.team; exec bash

vim /etc/net/ifaces/ens18/options меняем на static

vim /etc/net/ifaces/ens18/ipv4address 100.100.100.10/28

cp -r /etc/net/ifaces/ens18/ /etc/net/ifaces/ens19/

vim /etc/net/ifaces/ens19/ipv4address10.10.10.1/24

vim /etc/net/ifaces/ens18/ipv4route default via 100.100.100.1

reboot

**RTR-R:**

hostnamectl set-hostname rtr-r.au.team; exec bash

vim /etc/net/ifaces/ens18/options меняем на static

vim /etc/net/ifaces/ens18/ipv4address150.150.150.10/28

cp -r /etc/net/ifaces/ens18/ /etc/net/ifaces/ens19/

vim /etc/net/ifaces/ens19/ipv4address20.20.20.1/24

vim /etc/net/ifaces/ens18/ipv4route default via 150.150.150.1

reboot

**SRV-L:**

hostnamectl set-hostname srv-l.au.team; exec bash

vim /etc/net/ifaces/ens18/options меняем на static

vim /etc/net/ifaces/ens18/ipv4address10.10.10.100/24

vim /etc/net/ifaces/ens18/ipv4route default via 10.10.10.1

reboot

**WEB-L:**

hostnamectl set-hostname web-l.au.team; exec bash

vim /etc/net/ifaces/ens18/options меняем на static

vim /etc/net/ifaces/ens18/ipv4address10.10.10.110/24

vim /etc/net/ifaces/ens18/ipv4route default via 10.10.10.1

reboot

**WEB-R:**

hostnamectl set-hostname web-r.au.team; exec bash

vim /etc/net/ifaces/ens18/options меняем на static

vim /etc/net/ifaces/ens18/ipv4address20.20.20.100/24

vim /etc/net/ifaces/ens18/ipv4route default via 20.20.20.1

reboot

1. Настройка динамической трансляции адресов на всех роутерах.
   * Настройте динамическую трансляцию адресов для обоих офисов сторону ISP и с ISP в сторону общественной сети.
   * Все устройства в офисах должны иметь доступ к сети Интернет

**ISP:**

apt-get update

apt-get install nftables -y

vim /etc/nftables/nftables.nft

В конфиге прописать ниже после всего:

table ip nat {

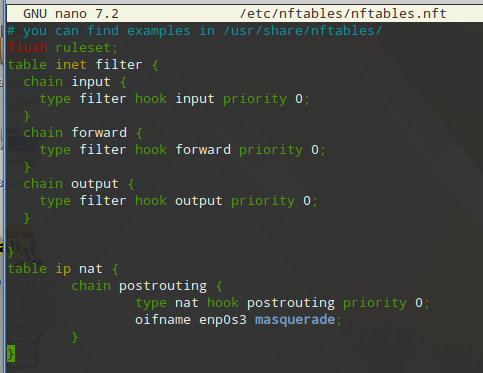
chain postrouting {

type nat hook postrouting priority 0;

oifname ens18 masquerade;

}

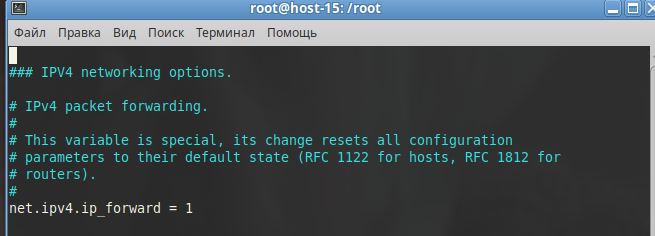
}



nft -f /etc/nftables/nftables.conf (не должно быть ошибок)

systemctl enable --now nftables

vim /etc/net/sysctl.conf





reboot

**RTR-L:**

apt-get update

apt-get install nftables -y

vim /etc/nftables/nftables.nft

В конфиге прописать ниже после всего:

table ip nat {

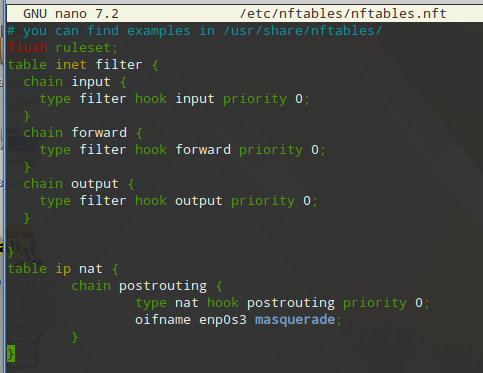
chain postrouting {

type nat hook postrouting priority 0;

oifname ens18 masquerade;

}

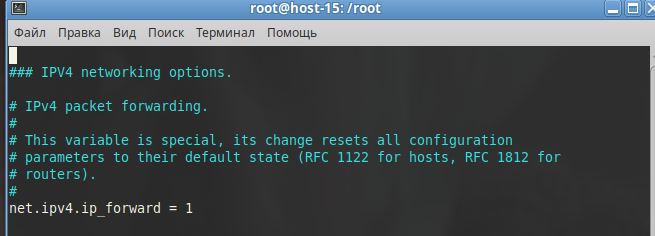
}



nft -f /etc/nftables/nftables.conf (не должно быть ошибок)

systemctl enable --now nftables

vim /etc/net/sysctl.conf





reboot

**RTR-R:**

apt-get update

apt-get install nftables -y

vim /etc/nftables/nftables.nft

В конфиге прописать ниже после всего:

table ip nat {

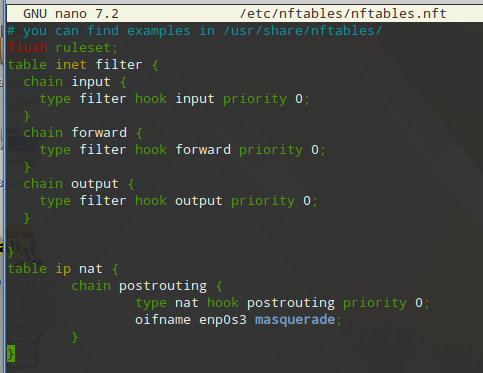
chain postrouting {

type nat hook postrouting priority 0;

oifname ens18 masquerade;

}

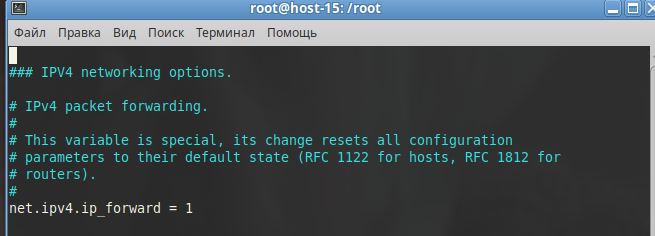
}



nft -f /etc/nftables/nftables.conf (не должно быть ошибок)

systemctl enable --now nftables

vim /etc/net/sysctl.conf





reboot

ПРОВЕРКА 1-2 ПУНКТА.

1)На каждой машине пишем ping 8.8.8.8. Должен быть везде интернет.

2) Пингуем каждую машину. К примеру, с WEB-L все машины.

ISP:

apt-get update && apt-get install nftables chrony -y

CLI:

apt-get update && apt-get install chrony yandex-browser -y

RTR-L:

apt-get update && apt-get install nftables chrony strongswan -y

RTR-R:

apt-get update && apt-get install chrony nftables strongswan -y

WEB-L:

apt-get update && apt-get install chrony docker-io docker-compose nfs-clients -y

WEB-R:

apt-get update && apt-get install chrony bind bind-utils nfs-clients -y

SRV-L:

apt-get update && apt-get install chrony bind bind-utils nfs-server -y

1. Между офисами 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 офисами 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.

**RTR-L:**

vim /etc/gre.up

Пишем:

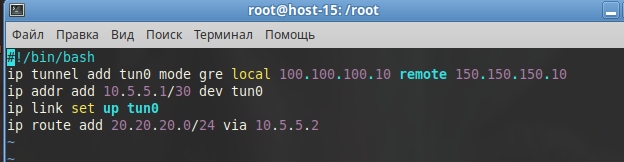
#!/bin/bash

ip tunnel add tun0 mode gre local 100.100.100.10 remote 150.150.150.10

ip addr add 10.5.5.1/30 dev tun0

ip link set up tun0

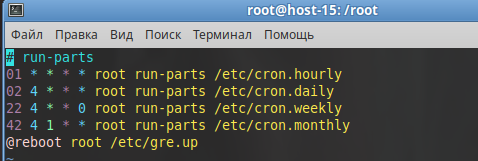
ip route add 20.20.20.0/24 via 10.5.5.2

Сохраняем файл

chmod +x /etc/gre.up

/etc/gre.up

vim /etc/crontab





apt-get install strongswan

vim /etc/strongswan/ipsec.conf

Пишем:

conn vpn

auto=start

type=tunnel

authby=secret

left=100.100.100.10

right=150.150.150.10

leftsubnet=0.0.0.0/0

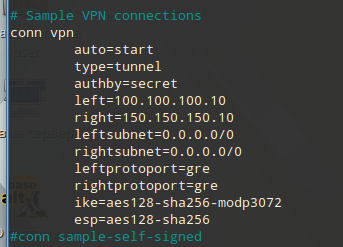
rightsubnet=0.0.0.0/0

leftprotoport=gre

rightprotoport=gre

ike=aes128-sha256-modp3072

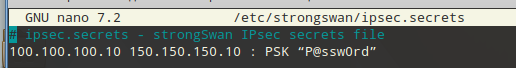
esp=aes128-sha256



vim /etc/strongswan/ipsec.secrets

Пишем:

100.100.100.10 150.150.150.10 : PSK “P@ssw0rd”



systemctl enable --now strongswan-starter.service

systemctl enable --now ipsec.service

**RTR-R:**

vim /etc/gre.up

Пишем:

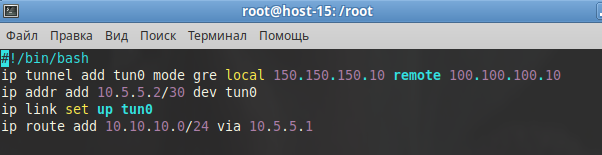
#!/bin/bash

ip tunnel add tun0 mode gre local 150.150.150.10 remote 100.100.100.10

ip addr add 10.5.5.2/30 dev tun0

ip link set up tun0

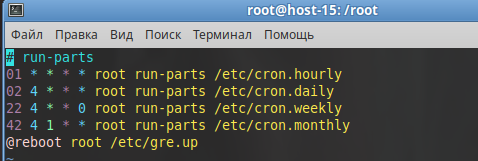
ip route add 10.10.10.0/24 via 10.5.5.1

Сохраняем файл

chmod +x /etc/gre.up

/etc/gre.up

vim /etc/crontab





apt-get install strongswan

vim /etc/strongswan/ipsec.conf

Пишем:

conn vpn

auto=start

type=tunnel

authby=secret

left=150.150.150.10

right=100.100.100.10

leftsubnet=0.0.0.0/0

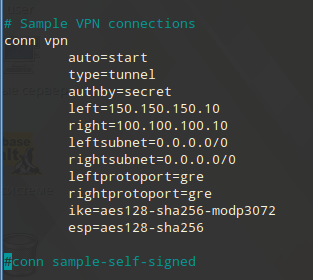
rightsubnet=0.0.0.0/0

leftprotoport=gre

rightprotoport=gre

ike=aes128-sha256-modp3072

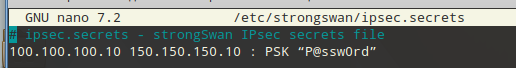
esp=aes128-sha256



vim /etc/strongswan/ipsec.secrets

Пишем:

100.100.100.10 150.150.150.10 : PSK “P@ssw0rd”



systemctl enable --now strongswan-starter.service

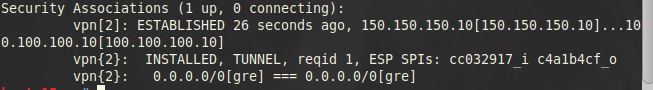
systemctl enable --now ipsec.service

Проверка:

Должен идти ping с RTR-L до RTR-R с туннеля. К примеру с RTR-L пингуем 10.5.5.2

ipsec status

Должно быть так:



Если не так то пробуем

ipsec update

ipsec restart

1. Настройка безопасного удаленного доступа на серверах WEB-L и WEB-R:
   * Для подключения используйте порт 2024
   * Разрешите подключения только пользователю sshuser
   * Ограничьте количество попыток входа до двух
   * Настройте баннер «Authorized access only»

**WEB-L и WEB-R:**

apt-get install openssh-server

vim /etc/openssh/sshd\_config

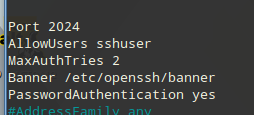
Port 2024

AllowUsers sshuser

MaxAuthTries 2

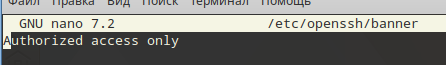
Banner /etc/openssh/banner

PasswordAuthentication yes



Vim /etc/openssh/banner

Authorized access only



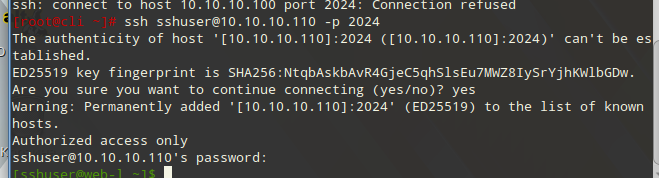
systemctl enable --now sshd

useradd sshuser

passwd sshuser (P@ssw0rd)

Проверка:

С CLI пытаемся подключится по ssh к Web-L. Ssh [sshuser@10.10.10.110](mailto:sshuser@10.10.10.110) -p 2024. Также для WEB-R только айпи другой.



1. Настройка DNS для офисов HQ и BR.
   * Основной DNS-сервер реализован на SRV-L. Дочерний сервер DNS на WEB-R.
   * Сервер должен обеспечивать разрешение имён в сетевые адреса устройств и обратно в соответствии с таблицей 2
   * В качестве DNS сервера пересылки используйте 94.232.137.104 DNS сервер.
   * Для устройств офиса RIGHT DNS сервер WEB-R, для устройств офиса LEFT DNS сервер SRV-L

Таблица 2

|  |  |  |
| --- | --- | --- |
| Имя ВМ | Запись | Тип |
| RTR-L | rtr-l.au.team | A,PTR |
| RTR-R | rtr-r.au.team | A,PTR |
| SRV-L | srv-l.au.team | A,PTR |
| WEB-L | web-l.au.team | A,PTR |
| WEB-R | web-r.au.team | A,PTR |
| ISP | isp.au.team | A,PTR |
| CLI | cli.au.team | A,PTR |

**SRV-L:**

**apt-get update && apt-get install bind bind-utils**

systemctl enable --now bind

**vim /etc/bind/options.conf**

listen-on { any; };

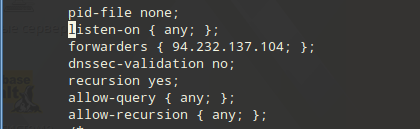
forwarders { 94.232.137.104; };

dnssec-validation no;

recursion yes;

allow-query { any; };

allow-recursion { any; };



В качестве DNS-сервера для самого себя должен быть 127.0.0.1:

vim /etc/resolv.conf



В конфигурационном файле /etc/bind/local.conf описываем необходимые зоны согласно требованию задания:

**vim /etc/bind/local.conf**

zone "au.team" {

type master;

file "au.team";

allow-transfer {20.20.20.100;};

};

zone "10.10.10.in-addr.arpa" {

type master;

file "left.reverse";

allow-transfer {20.20.20.100;};

};

zone "20.20.20.in-addr.arpa" {

type master;

file "right.reverse";

allow-transfer {20.20.20.100;};

};

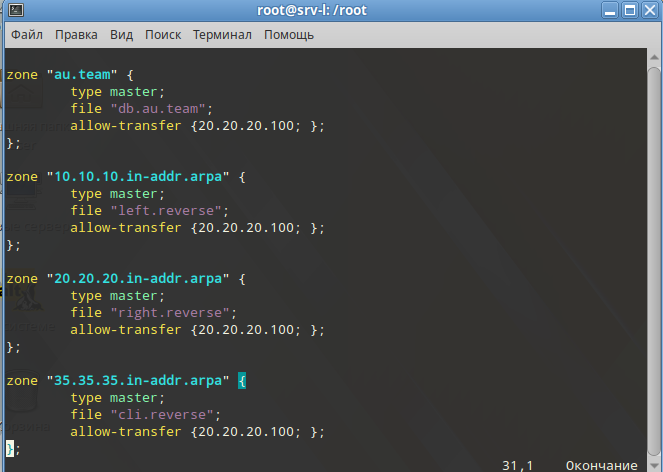
zone "35.35.35.in-addr.arpa" {

type master;

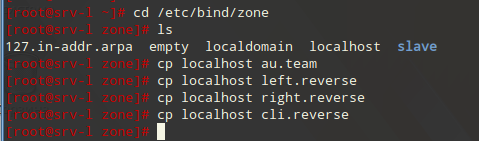
file "cli.reverse";

allow-transfer {20.20.20.100;};

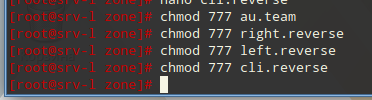
};



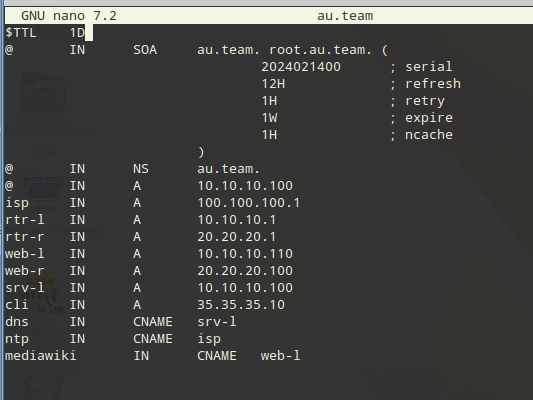
cd /etc/bind/zone/



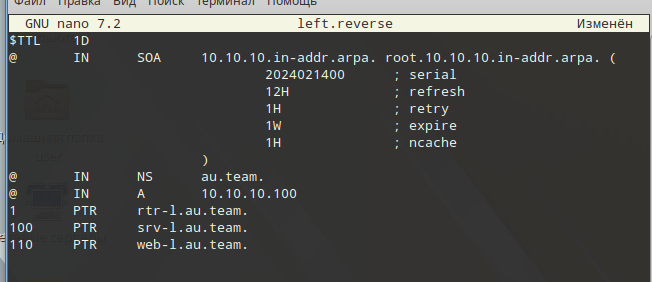
Задаём необходимые права:



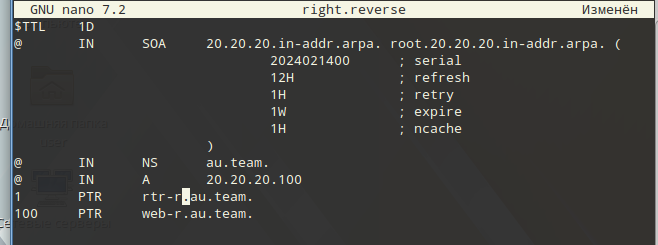
vim /etc/bind/zone/au.team



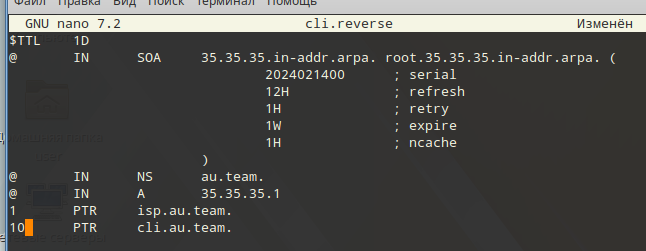
vim /etc/bind/zone/left.reverse



vim /etc/bind/zone/right.reverse



vim /etc/bind/zone/cli.reverse

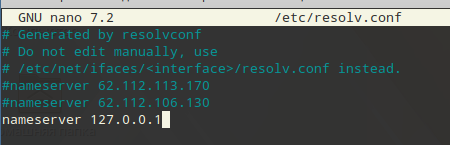


**systemctl restart bind**

**named-checkconf**

**named-checkconf -z**

vim /etc/resolv.conf



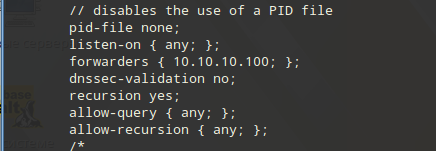
reboot

**WEB-R:**

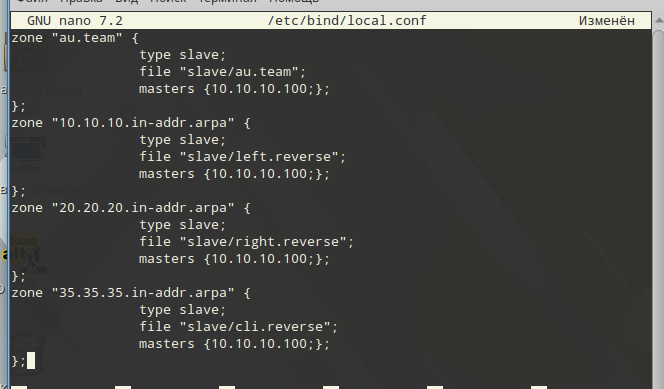
**apt-get update && apt-get install bind bind-utils**

systemctl enable --now bind

**vim /etc/bind/options.conf**



**vim /etc/bind/local.conf**



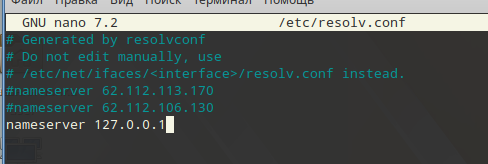
chown named:named /var/lib/bind/zone/slave/

chown named:named /etc/bind/zone/slave/

systemctl restart bind

systemctl status bind

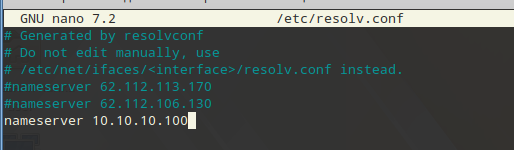
vim /etc/resolv.conf



reboot

**CLI:**

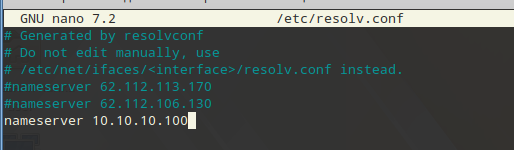
vim /etc/resolv.conf



reboot

**ISP:**

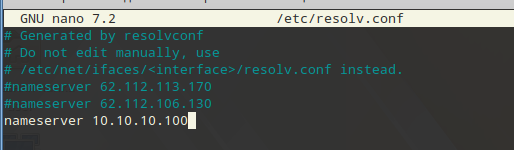
vim /etc/resolv.conf



reboot

**RTR-L:**

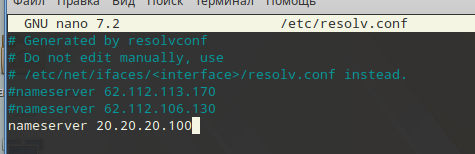
vim /etc/resolv.conf



reboot

**RTR-R:**

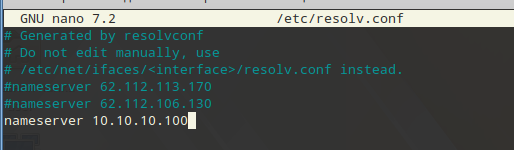
vim /etc/resolv.conf

****

reboot

**WEB-L:**

vim /etc/resolv.conf

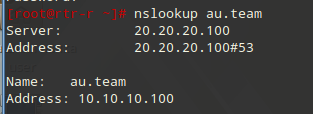


reboot

ПРОВЕРКА:

На SRV-L и WEB-R перезагружаем bind. Systemctl restart bind.

Далее проверяем зоны. Желательно на каждой машине. К примеру на RTR-R nslookup au.team. И так каждую зону.



1. Настройте службу сетевого времени на базе сервиса chrony
   * В качестве сервера выступает ISP
   * На ISP настройте сервер chrony, выберите стратум 5
   * В качестве клиентов настройте RTR-L, RTR-R, SRV-L, WEB-L, WEB-R, CLI

Настройка на ISP

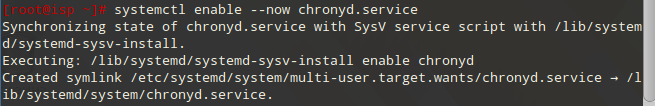
Открываем файл /etc/chrony.conf

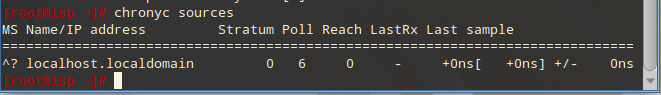
В конце файла пишем



Сохраняем его

systemctl restart chronyd





На всех остальных машинах

В файле /etc/chrony.conf комментируем строчки

**pool pool.ntp.org iburst**

**На RTR-R**

В конце файла пишем server 150.150.150.1 iburst

**На RTR-L**

В конце файла пишем server 100.100.100.1 iburst

**На CLI**

В конце файла пишем server 35.35.35.1 iburst

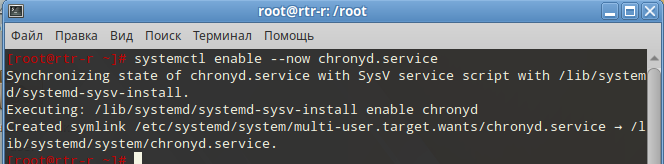
**На SRV-L и WEB-L**

В конце файла пишем server 100.100.100.1 iburst

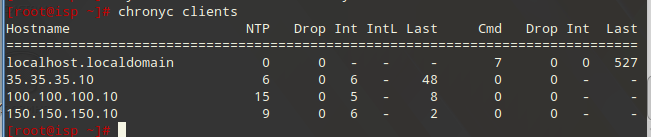
**На WEB-R**

В конце файла пишем server 150.150.150.1 iburst

Сохраняем и выходим из файла



ПРОВЕРКА:



Также для проверки можно поменять время и оно должно поменятся как на ISP.

1. Сконфигурируйте файловое хранилище:
   * При помощи четырёх дополнительных дисков, размером 1Гб каждый, на SRV-L сконфигурируйте дисковый массив уровня 5
   * Имя устройства – md0, конфигурация массива размещается в файле

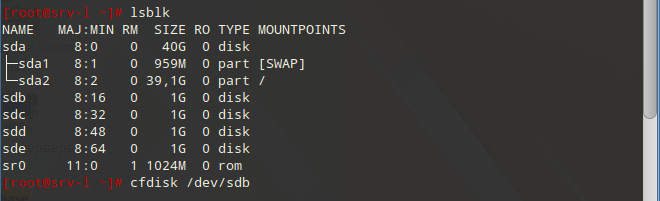
/etc/mdadm.conf

* + Обеспечьте автоматическое монтирование в папку /raid5
  + Создайте раздел, отформатируйте раздел, в качестве файловой системы используйте ext4
  + Настройте сервер сетевой файловой системы(nfs), в качестве папки общего

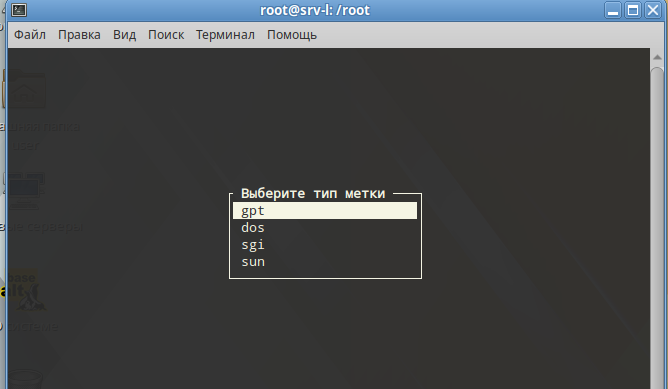
доступа выберите /raid5/nfs, доступ для чтения и записи для всей сети в сторону WEB-L, WEB-R.

* + На WEB-L, WEB-R настройте автомонтирование в папку /mnt/nfs

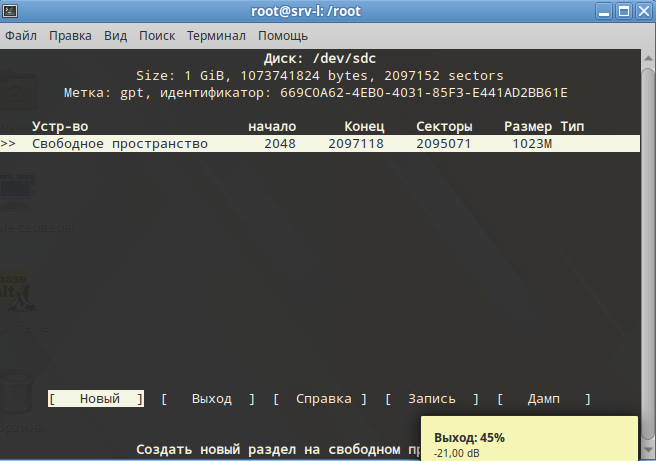
Пишем lsblk, смотрим диски



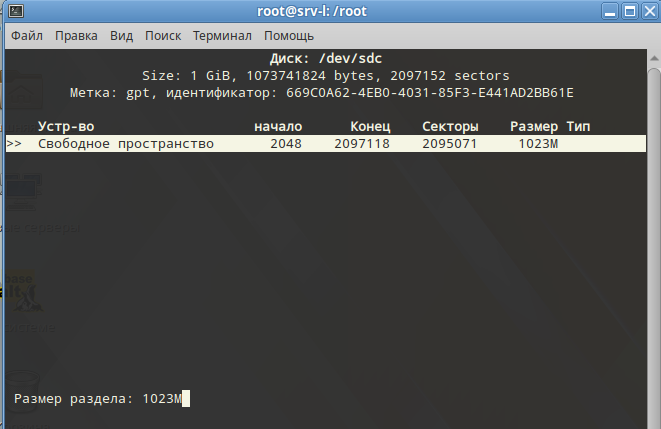
Команда: cfdisk /dev/sdс



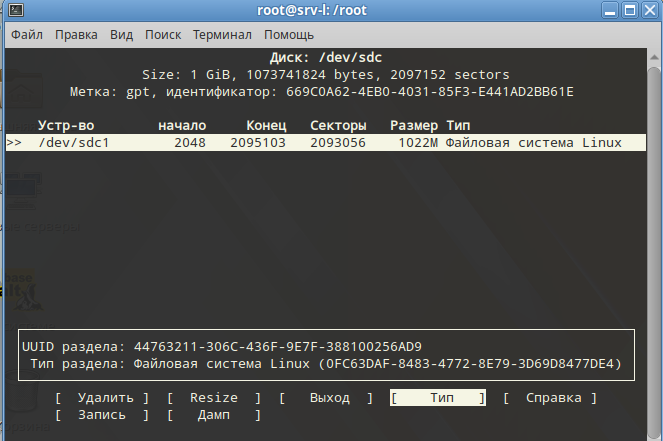
Нажимаем enter



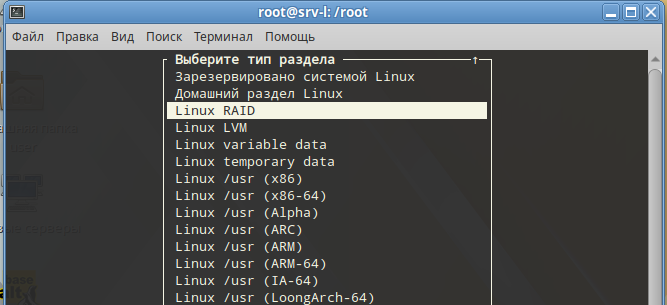
Выбираем Новый



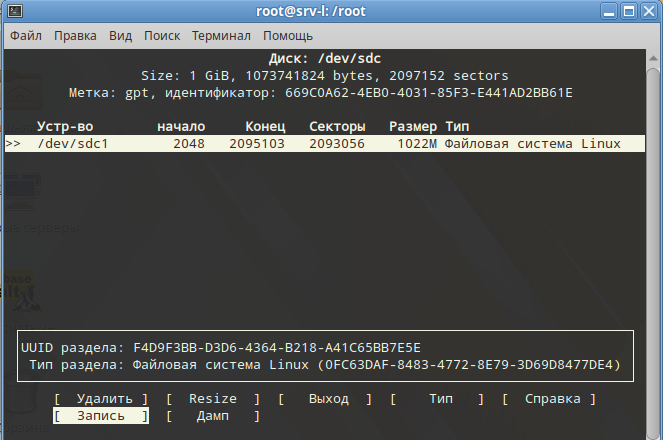
Нажимаем enter



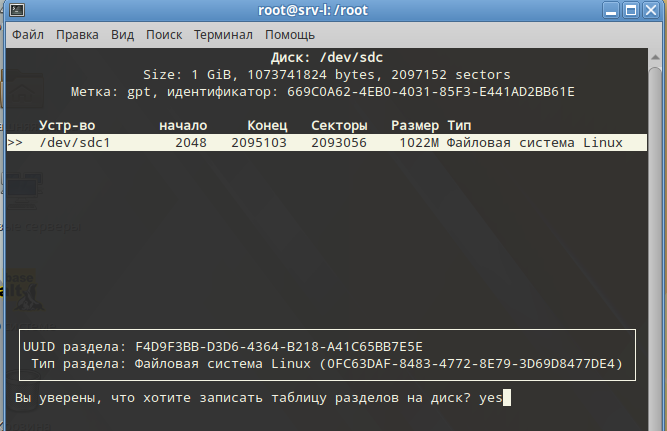
Нажимаем Тип



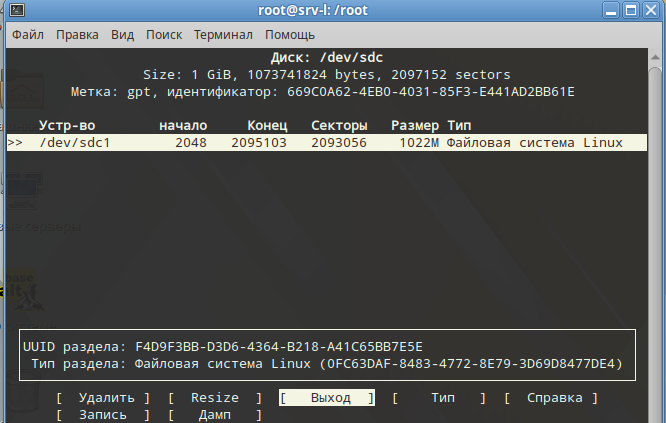
Выбираем Linux Raid, нажимаем enter



Выбираем Запись



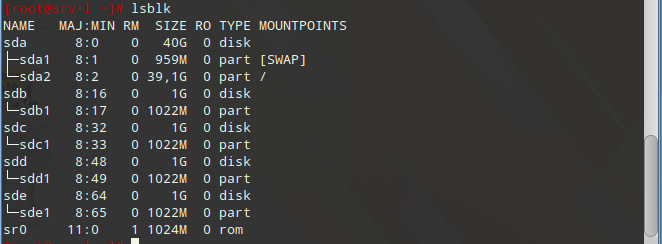
Пишем yes



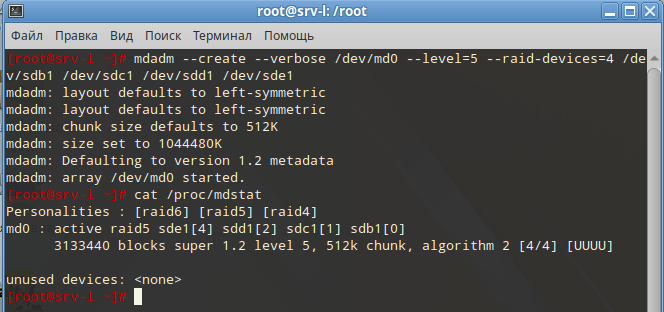
Выбираем Выход и нажимаем enter

То же самое делаем для /dev/sdb /dev/sdd /dev/sde

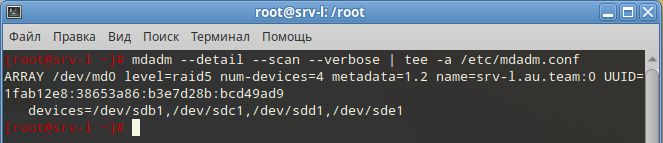
Получилось

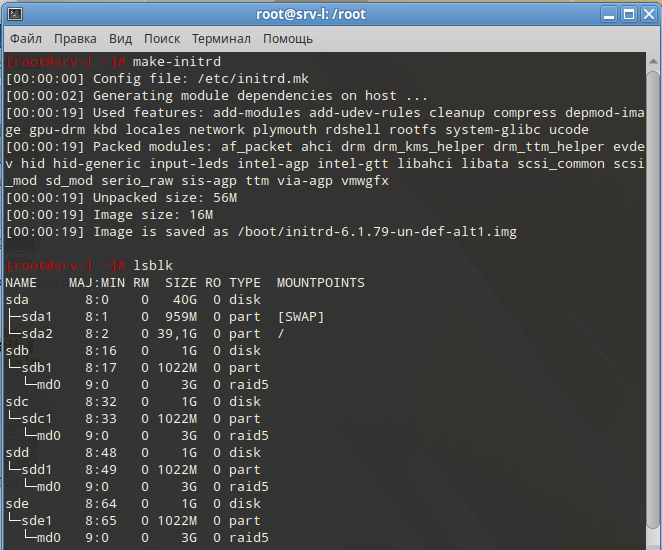


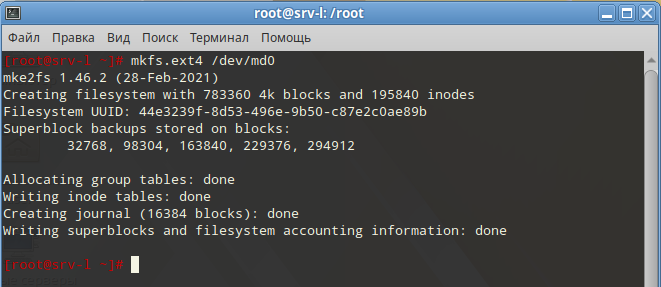
mdadm --create --verbose /dev/md0 --level=5 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1

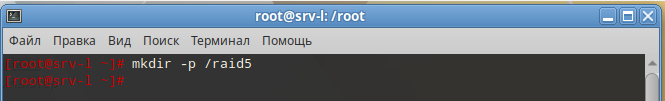


mdadm --detail --scan --verbose | tee -a /etc/mdadm.conf

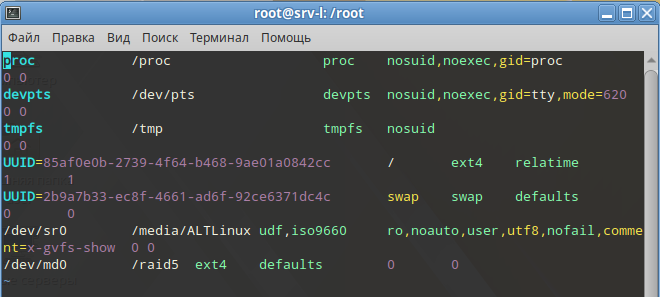




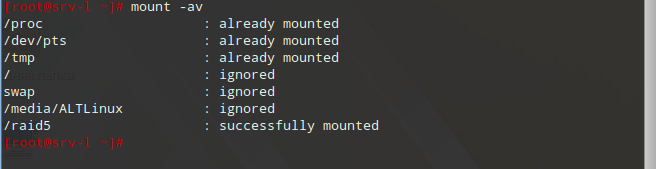




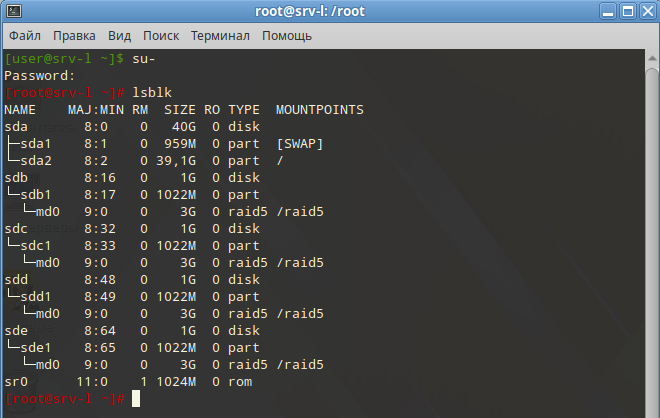
Открываем файл /etc/fstab

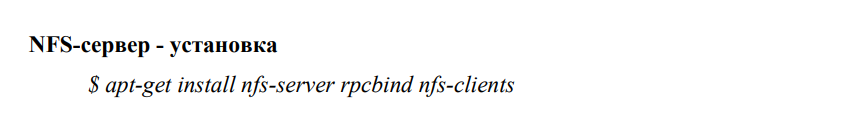


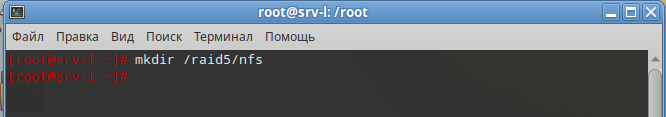
(последняя строчка)



Перезагружаем машину reboot

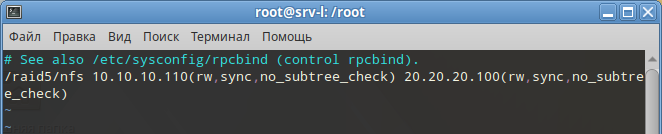






chmod 777 /raid5/nfs

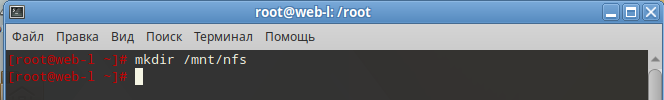
vim /etc/exports



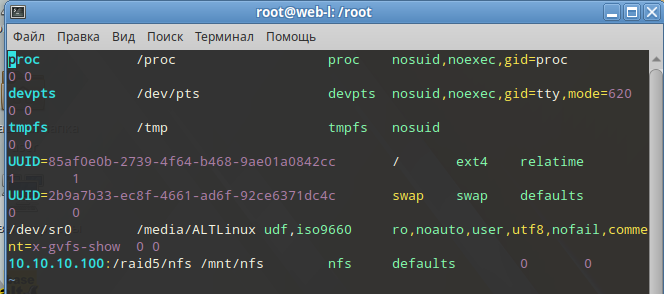
Systemctl enable --now nfs-server

На web-l и web-r прописываем

apt-get install nfs-clients



vim /etc/fstab



(последняя строчка)

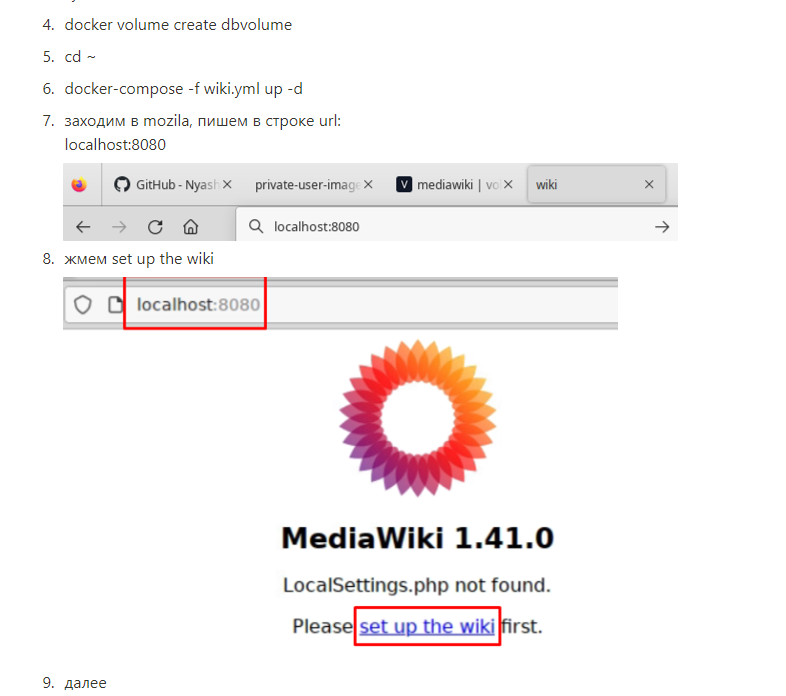
В конце прописываем mount -a

ПРОВЕРКА:

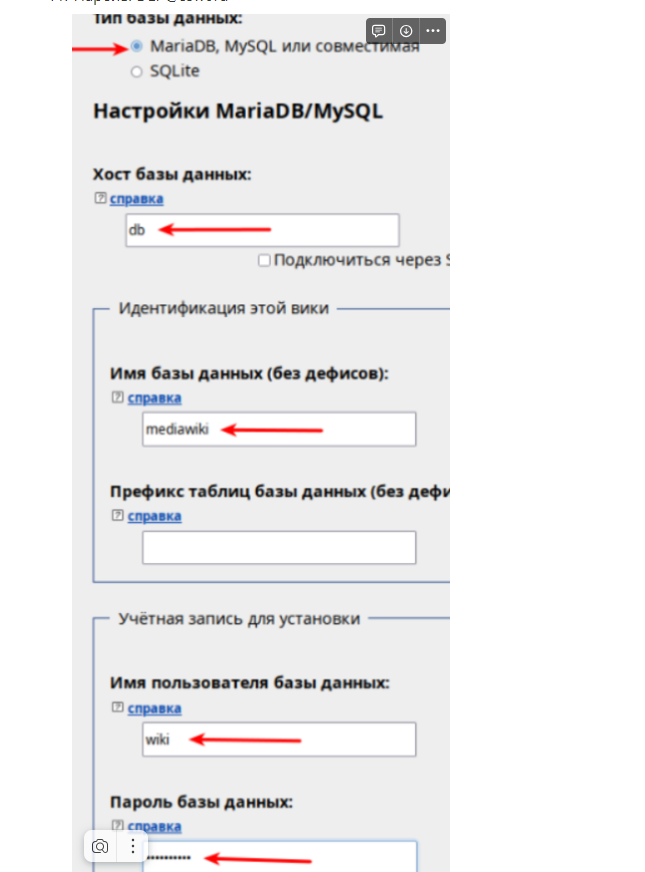
Создадим файлы в папке и они должны виднется .

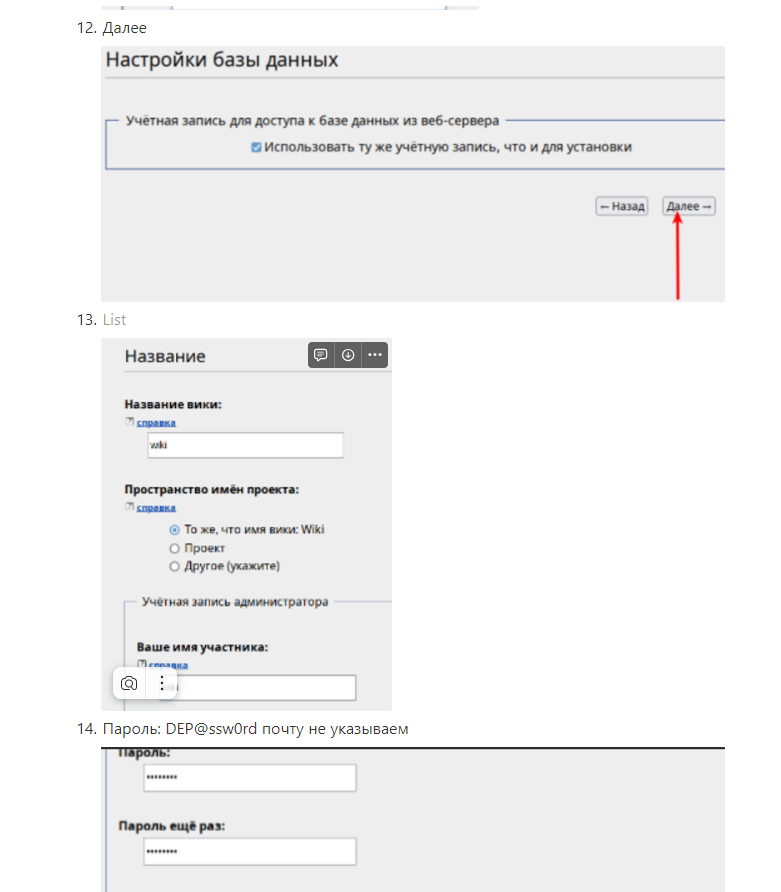
1. Запустите сервис MediaWiki используя docker на сервере WEB-L.
   * Установите Docker и Docker Compose.
   * Создайте в домашней директории пользователя файл wiki.yml для приложения MediaWiki:
   * Средствами docker compose должен создаваться стек контейнеров с приложением MediaWiki и базой данных
   * Используйте два сервиса;
   * Основной контейнер MediaWiki должен называться wiki и использовать образ mediawiki;
   * Файл LocalSettings.php с корректными настройками должен находиться в домашней папке пользователя и автоматически монтироваться в образ;
   * Контейнер с базой данных должен называться db и использовать образ mysql;
   * Он должен создавать базу с названием mediawiki, доступную по стандартному порту, для пользователя wiki с паролем DEP@ssw0rd;
   * База должна храниться в отдельном volume с названием dbvolume.
   * MediaWiki должна быть доступна с WEB-R по порту 8080 и имени midiawiki.au.team

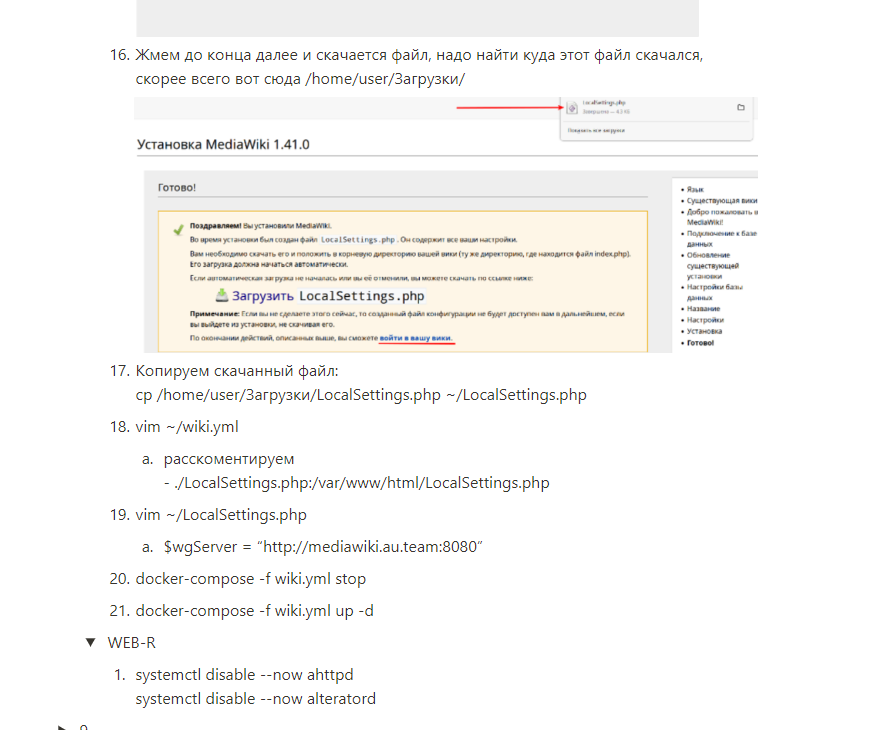












1. Удобным способом установите приложение Яндекс Браузере для организаций на CLI

На cli   
apt-get update

apt-get install yandex-browser-stable