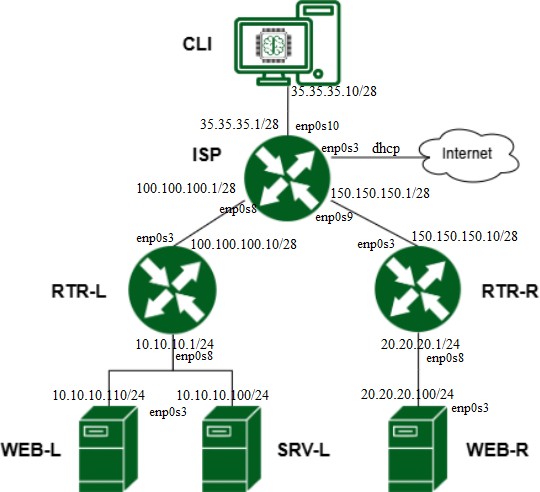
Сети и системы передачи информации

enp0s3 = ens18 enp0s8 = ens19 enp0s9 = ens20 enp0s10 = ens21



nftables - isp, rtr-l, rtr-r ip\_forward - isp, rtr-l, rtr-r docker-io, docker-compose - web-l bind bind-utils - web-r, srv-l

1. *начать необходимо с настройки ip - адресов + динамическая трансляция*

ISP

* 1. hostnamectl set-hostname isp
  2. vim /etc/net/sysctl.conf
     1. net.ipv4.ip\_forward = 1
  3. cd /etc/net/ifaces
  4. cp -r ens18/ ens19
  5. vim ens19/options
     1. BOOTPROTO=static
  6. cp -r ens19/ ens20
  7. cp -r ens19/ ens21
  8. vim ens19/ipv4address a. 100.100.100.1/28
  9. vim ens19/ipv4route

a. 10.10.10.0/24 via 100.100.100.10

* 1. vim ens20/ipv4address a. 150.150.150.1/28
  2. vim ens20/ipv4route

a. 20.20.20.0/24 via 150.150.150.10

* 1. vim ens21/ipv4address a. 35.35.35.1/28
  2. systemctl restart network
  3. reboot
  4. apt-get update && apt-get install nftables chrony -y
  5. vim /etc/nftables/nftables.nft
     1. в начало: flush ruleset;
     2. в конец:

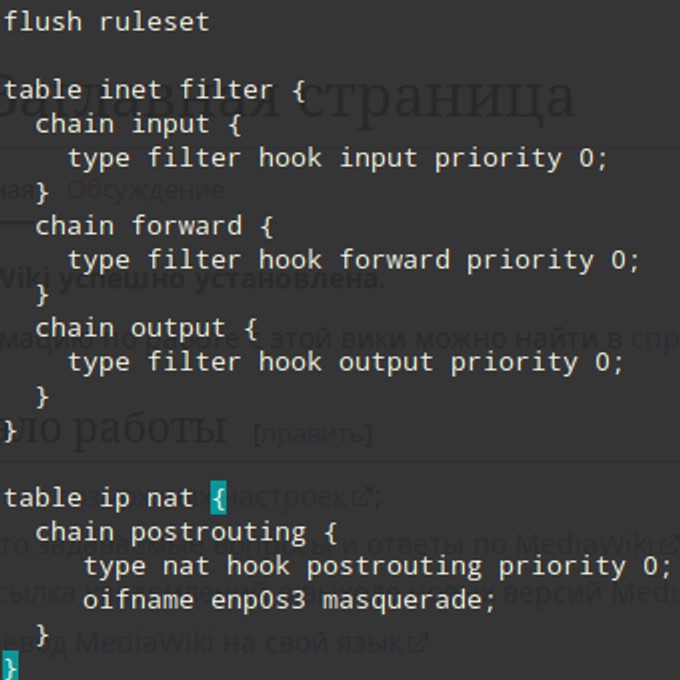
table ip nat {

chain postrouting {

type nat hook postrouting priority 0; oifname ens18 masquerade;

}

}



* 1. systemctl enable --now nftables
  2. nft -f /etc/nftables/nftables.nft
  3. nft list ruleset CLI

1. hostnamectl set-hostname cli
2. cd /etc/net/ifaces
3. cp -r ens18/ ens19
4. vim ens19/options
   1. BOOTPROTO=static
5. vim ens19/ipv4address a. 35.35.35.10/28
6. vim ens19/ipv4route
   1. default via 35.35.35.1
7. systemctl restart network
8. reboot
9. apt-get update && apt-get install chrony yandex-browser

-y RTR-L

1. hostnamectl set-hostname rtr-l
2. vim /etc/net/sysctl.conf
   1. net.ipv4.ip\_forward = 1
3. cd /etc/net/ifaces
4. vim ens18/options
   1. BOOTPROTO=static
5. cp -r ens18/ ens19
6. vim ens18/ipv4address a. 100.100.100.10/28
7. vim ens18/ipv4route

a. default via 100.100.100.1

1. vim ens19/ipv4address a. 10.10.10.1/24
2. systemctl restart network
3. reboot
4. apt-get update && apt-get install nftables chrony strongswan -y

RTR-R

1. hostnamectl set-hostname rtr-r
2. vim /etc/net/sysctl.conf
   1. net.ipv4.ip\_forward = 1
3. cd /etc/net/ifaces
4. vim ens18/options
   1. BOOTPROTO=static
5. cp -r ens18/ ens19
6. vim ens18/ipv4address a. 150.150.150.10/28
7. vim ens18/ipv4route

a. default via 150.150.150.1

1. vim ens19/ipv4address a. 20.20.20.1/24
2. systemctl restart network
3. reboot
4. apt-get update && apt-get install chrony nftables strongswan -y

WEB-L

1. hostnamectl set-hostname web-l
2. cd /etc/net/ifaces/ens18/
3. vim options
   1. BOOTPROTO=static
4. vim ipv4address a. 10.10.10.110/24
5. vim ipv4route
   1. default via 10.10.10.1
6. systemctl restart network
7. reboot
8. apt-get update && apt-get install chrony docker-io docker-compose nfs-clients -y

WEB-R

1. hostnamectl set-hostname web-r
2. cd /etc/net/ifaces/ens18/
3. vim options
   1. BOOTPROTO=static
4. vim ipv4address a. 20.20.20.100/24
5. vim ipv4route
   1. default via 20.20.20.1
6. systemctl restart network
7. reboot
8. apt-get update && apt-get install chrony bind bind- utils nfs-clients -y

SRV-L

1. hostnamectl set-hostname srv-l
2. cd /etc/net/ifaces/ens18/
3. vim options
   1. BOOTPROTO=static
4. vim ipv4address a. 10.10.10.100/24
5. vim ipv4route
   1. default via 10.10.10.1
6. systemctl restart network
7. reboot
8. apt-get update && apt-get install chrony bind bind- utils nfs-server -y
9. *Настройка динамической трансляции адресов на всех роутерах*

начать выполение с isp RTR-L

* 1. vim /etc/nftables/nftables.nft
     1. в начало: flush ruleset
     2. в конец:

table ip nat {

chain postrouting {

type nat hook postrouting priority 0; ip saddr 10.10.10.0/24 oifname ens18

masquerade;

}

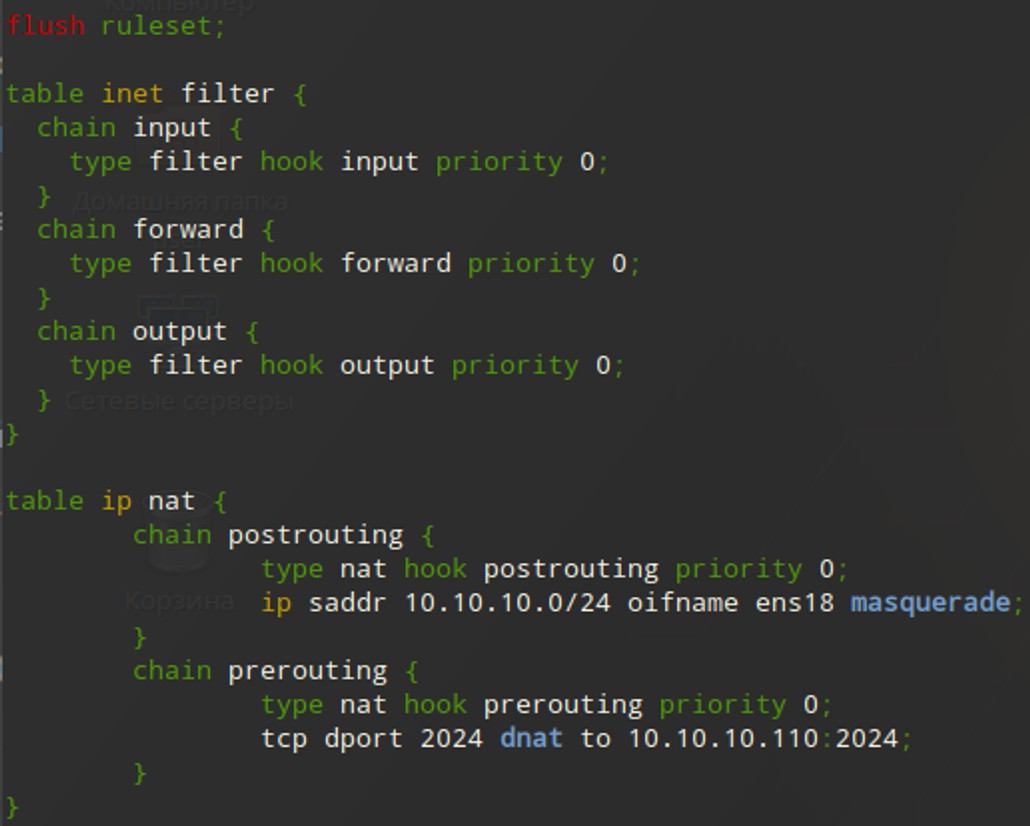
chain prerouting {

type nat hook prerouting priority 0;

tcp dport 2024 dnat to 10.10.10.110:2024;

}

}



* 1. systemctl enable --now nftables
  2. nft -f /etc/nftables/nftables.nft
  3. nft list ruleset RTR-R

1. vim /etc/nftables/nftables.nft
   1. в начало: flush ruleset
   2. в конец:

table ip nat {

chain postrouting {

type nat hook postrouting priority 0; ip saddr 20.20.20.0/24 oifname ens18

masquerade;

}

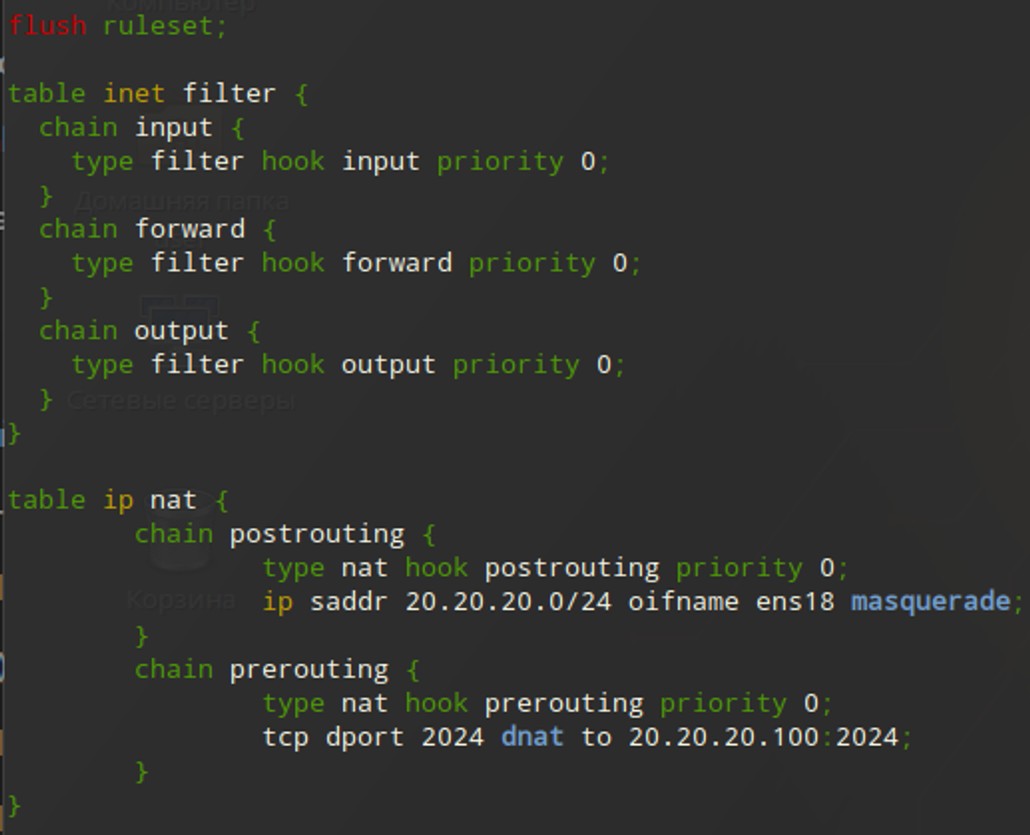
chain prerouting {

type nat hook prerouting priority 0;

tcp dport 2024 dnat to 20.20.20.100:2024;

}

}



1. systemctl enable --now nftables
2. nft -f /etc/nftables/nftables.nft
3. nft list ruleset
4. *Между офисами должен быть установлен защищенный туннель, позволяющий*

*осуществлять связь между регионами с применением внутренних адресов*

RTR-L

* 1. vim /etc/gre.up
     1. #!/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

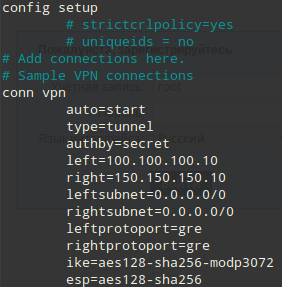
* 1. chmod +x /etc/gre.up
  2. /etc/gre.up
  3. vim /etc/crontab
     1. в конец добавляем: @reboot root /etc/gre.up
  4. vim /etc/strongswan/ipsec.conf

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



* 1. vim /etc/strongswan/ipsec.secrets

a. 100.100.100.10 150.150.150.10 : PSK “P@ssw0rd”

* 1. systemctl enable --now ipsec.service RTR-R

1. vim /etc/gre.up
   1. #!/bin/bash

ip tunnel add tun0 mode gre local 150.150.150.10 remote 100.100.100.10

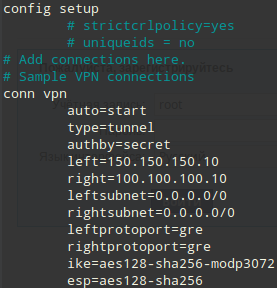
ip addr add 10.5.5.2/30 dev tun0 ip link set up tun0

ip route add 10.10.10.0/24 via 10.5.5.1

1. chmod +x /etc/gre.up
2. /etc/gre.up
3. vim /etc/crontab
   1. в конец добавляем: @reboot root /etc/gre.up
4. apt-get install strongswan
5. vim /etc/strongswan/ipsec.conf 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



1. vim /etc/strongswan/ipsec.secrets

a. 100.100.100.10 150.150.150.10 : PSK “P@ssw0rd”

1. systemctl enable --now ipsec.service

После того, как на обоих RTR все прописали, проверяем на любом из них ipsec status, если есть 1 up, то все норм, если нет, то пробуем ipsec update и ipsec restart.

1. *Настройка безопасного удаленного доступа (ssh) на серверах WEB-L и WEB-R*

WEB-L

* 1. vim /etc/openssh/banner.txt
     1. Authorized access only
  2. vim /etc/openssh/sshd\_config
     1. расскоментируем строчку Port 22 Port 2024
     2. расскоментируем строчку MaxAuthTries 6 MaxAuthTries 2
     3. расскоментируем строчку Banner none вместо none пишем путь к banner.txt Banner /etc/openssh/banner.txt
     4. добавляем в конец AllowUsers sshuser
  3. adduser sshuser
  4. passwd sshuser
     1. P@ssw0rd
  5. systemctl restart sshd WEB-R

1. vim /etc/openssh/banner.txt
   1. Authorized access only
2. vim /etc/openssh/sshd\_config
   1. расскоментируем строчку Port 22 Port 2024
   2. расскоментируем строчку MaxAuthTries 6 MaxAuthTries 2
   3. расскоментируем строчку Banner none вместо none пишем путь к banner.txt Banner /etc/openssh/banner.txt
   4. добавляем в конец AllowUsers sshuser
3. adduser sshuser
4. passwd sshuser
   1. P@ssw0rd
5. systemctl restart sshd
6. проверка

ssh -p 2024 [sshuser@10.10.10.110](mailto:sshuser@10.10.10.110)

1. *Настройка DNS для офисов HQ и BR*

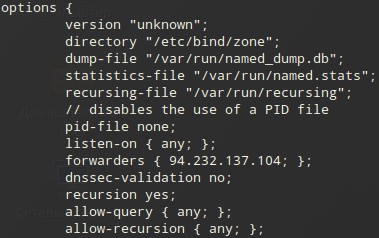
# SRV-L

* 1. systemctl enable --now bind
  2. vim /etc/bind/options.conf
     1. listen-on { any; };

forwarders { 94.232.137.104; }; dnssec-validation no;

recursion yes;

allow-query { any; }; allow-recursion { any; };



* 1. vim /etc/bind/local.conf
     1. добавляем после слов Add other zones here: 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;};

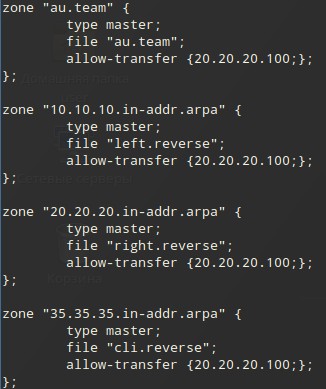
};

zone "35.35.35.in-addr.arpa" { type master;

file "cli.reverse";

allow-transfer {20.20.20.100;};

};



* 1. cd /etc/bind/zone/
  2. cp localhost au.team
  3. vim au.team
     1. заменяем localhost. на au.team. и root.localhost. на root.au.team.
     2. пишем через табуляцию зоны:

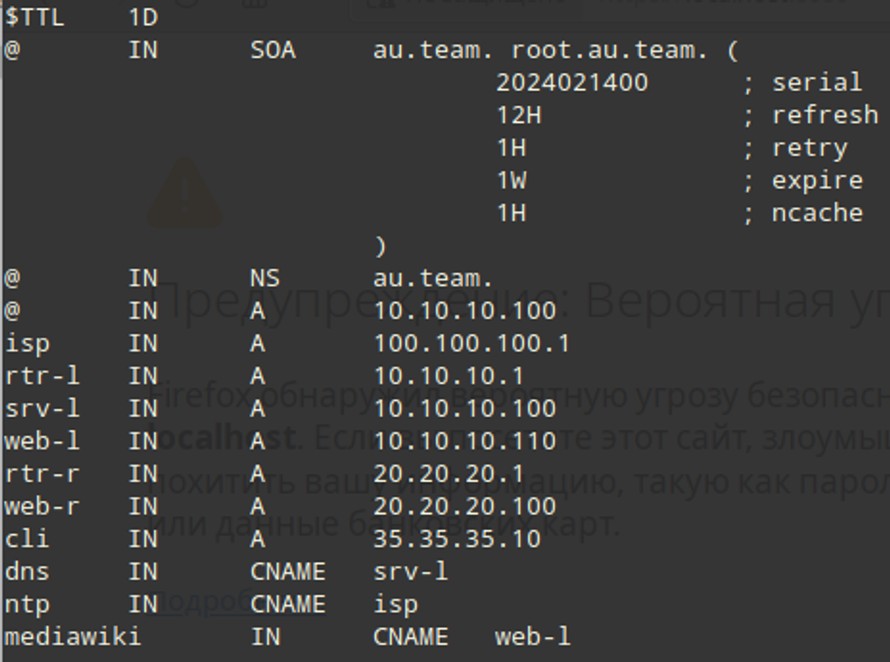
mediawiki

IN

CNAME

web-l

|  |  |  |  |
| --- | --- | --- | --- |
| @ | IN | NS | au.team. |
| @ | IN | A | 10.10.10.100 |
| isp | IN | A | 100.100.100.1 |
| rtr-l | IN | A | 10.10.10.1 |
| rtr-r | IN | A | 20.20.20.1 |
| web-l | IN | A | 10.10.10.110 |
| web-r | IN | A | 20.20.20.100 |
| srv-l | IN | A | 10.10.10.100 |
| cli | IN | A | 35.35.35.10 |
| dns | IN | CNAME | srv-l |
| ntp | IN | CNAME | isp |



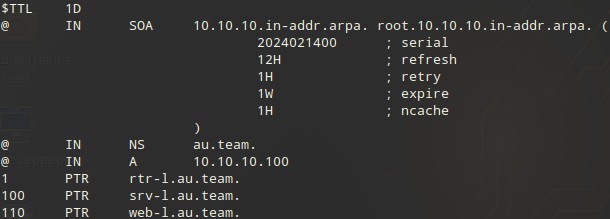
* 1. cp localhost right.reverse
  2. vim right.reverse
     1. заменяем localhost. на 20.20.20.in-addr.arpa. и root.localhost. на root.20.20.20.in-addr.arpa.
     2. пишем через табуляцию зоны:

|  |  |  |  |
| --- | --- | --- | --- |
| @ | IN | NS | au.team. |
| @ | IN | A | 20.20.20.100 |
| 1 | PTR | rtr-r.au.team. | |
| 100 | PTR | web-r.au.team. | |



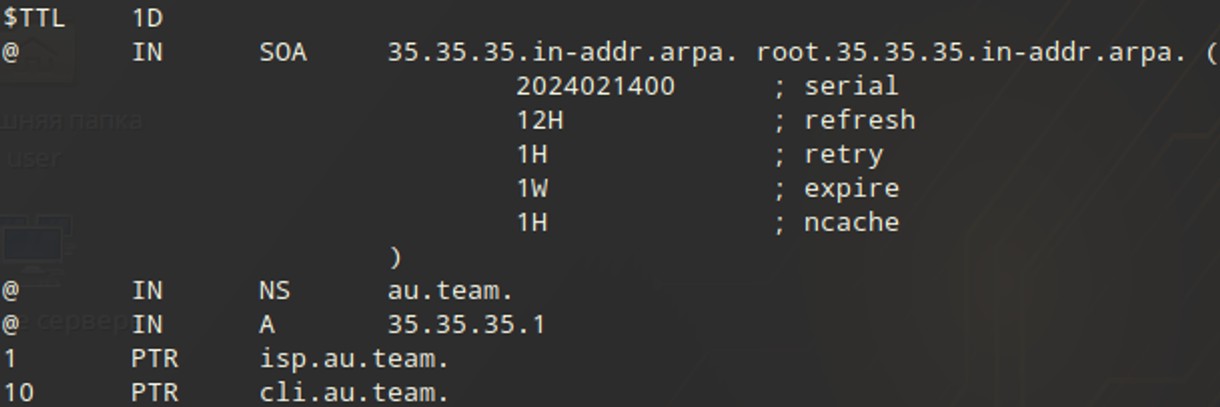
* 1. cp right.reverse left.reverse
  2. vim left.reverse
     1. заменяем 20.20.20.in-addr.arpa. на 10.10.10.in- addr.arpa. и root.localhost. на root.10.10.10.in- addr.arpa.
     2. пишем через табуляцию зоны:

|  |  |  |  |
| --- | --- | --- | --- |
| @ | IN | NS | au.team. |
| @ | IN | A | 10.10.10.100 |
| 1 | PTR | rtr-l.au.team. | |
| 100 | PTR | srv-l.au.team. | |
| 110 | PTR | web-l.au.team. | |



* 1. cp right.reverse cli.reverse
  2. vim cli.reverse
     1. заменяем 10.10.10.in-addr.arpa. на 35.35.35.in- addr.arpa. и root.localhost. на root.35.35.35.in- addr.arpa.
     2. пишем через табуляцию зоны:

|  |  |  |  |
| --- | --- | --- | --- |
| @ | IN | NS | au.team. |
| @ | IN | A | 35.35.35.1 |
| 1 | PTR | isp.au.team. | |
| 10 | PTR | cli.au.team. | |



|  |  |  |
| --- | --- | --- |
| 13. chmod | 777 | au.team |
| 14. chmod | 777 | right.reverse |
| 15. chmod | 777 | left.reverse |
| 16. chmod | 777 | cli.reverse |

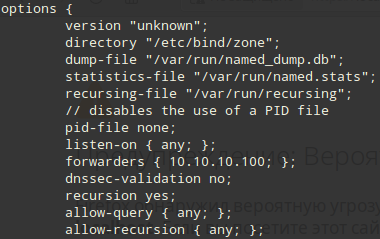
1. systemctl restart bind
2. vim /etc/resolv.conf

a. должен быть указан только один nameserver 127.0.0.1

# WEB-R

1. systemctl enable --now bind
2. vim /etc/bind/options.conf
   1. что должно быть в options: listen-on { any; }; forwarders { 10.10.10.100; }; dnssec-validation no; recursion yes;

allow-query { any; }; allow-recursion { any; };



1. vim /etc/bind/local.conf
   1. добавляем после слов Add other zones here: zone "au.team" {

type slave;

file "slave/au.team";

masters {10.10.10.100;};

};

zone "10.10.10.in-addr.arpa" { type slave;

file "slave/left.reverse"; masters {10.10.10.100;};

};

zone "20.20.20. in-addr.arpa" { type slave;

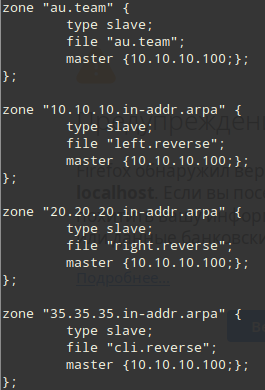
file "slave/right. reverse"; masters {10.10.10.100;};

};

zone "35.35.35. in-addr.arpa" { type slave;

file "slave/cli. reverse"; masters {10.10.10.100;};

};



1. chown named:named /var/lib/bind/zone/slave/
2. chown named:named /etc/bind/zone/slave/
3. systemctl restart bind
4. vim /etc/resolv.conf
   1. должен быть указан только один nameserver 127.0.0.1

После того как DNS сервера настроены, надо указать на оставшихся машинах указать в качестве DNS сервера наши сервера.

CLI

1. vim /etc/resolv.conf

a. должен быть указан только один nameserver 100.100.100.10

ISP

1. vim /etc/resolv.conf

a. должен быть указан только один nameserver 100.100.100.10

RTR-L

1. vim /etc/resolv.conf

a. должен быть указан только один nameserver 10.10.10.100

RTR-R

1. vim /etc/resolv.conf
   1. должен быть указан только один nameserver 20.20.20.100

(если WEB-R не работает, то 10.10.10.100)

WEB-L

1. echo “nameserver 10.10.10.100” >

/etc/net/ifaces/enp0s3/resolv.conf

1. vim /etc/resolv.conf
   1. должен быть указан только один nameserver 10.10.10.100
   2. если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8
2. *Настройте службу сетевого времени на базе сервиса chrony*

ISP

* 1. vim /etc/chrony.conf

a. в конец пишем: server 127.0.0.1

allow 100.100.100.0/28 allow 150.150.150.0/28

allow 35.35.35.0/28 allow 10.10.10.0/24 allow 20.20.20.0/24 local stratum 5

* 1. systemctl restart chronyd CLI

1. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

server 35.35.35.1 iburst

1. systemctl restart chronyd RTR-L
2. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

server 100.100.100.1 iburst

1. systemctl restart chronyd RTR-R
2. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

server 150.150.150.1 iburst

1. systemctl restart chronyd WEB-R
2. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

server 150.150.150.1 iburst

1. systemctl restart chronyd WEB-L
2. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

server 100.100.100.1 iburst

1. systemctl restart chronyd SRV-L
2. vim /etc/chrony.conf
   1. комментируем “pool pool.ntp.org iburst”
   2. в конец пишем:

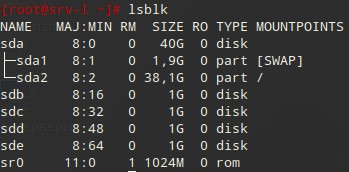
server 100.100.100.1 iburst

1. systemctl restart chronyd
2. *Сконфигурируйте файловое хранилище*

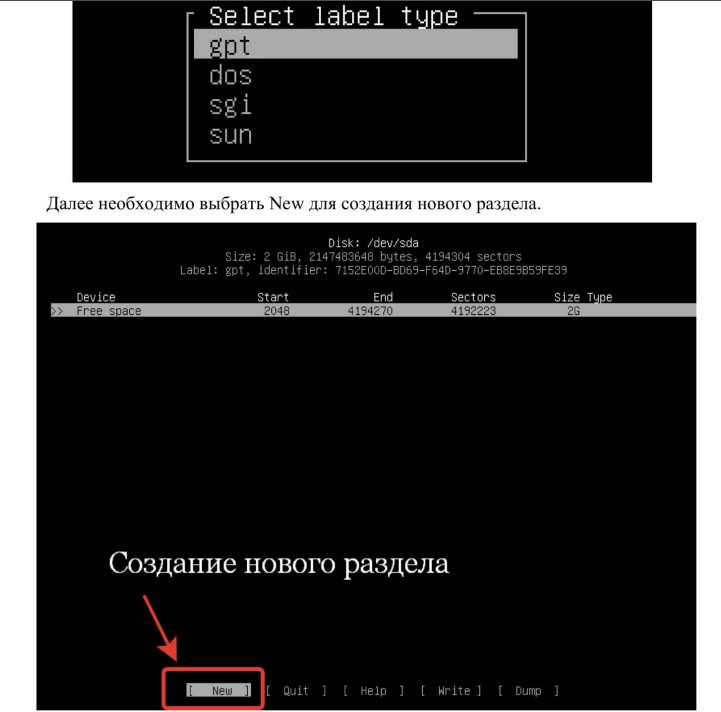
SRV-L

* 1. lsblk проверяем NAME 4 дисков размером 1G:

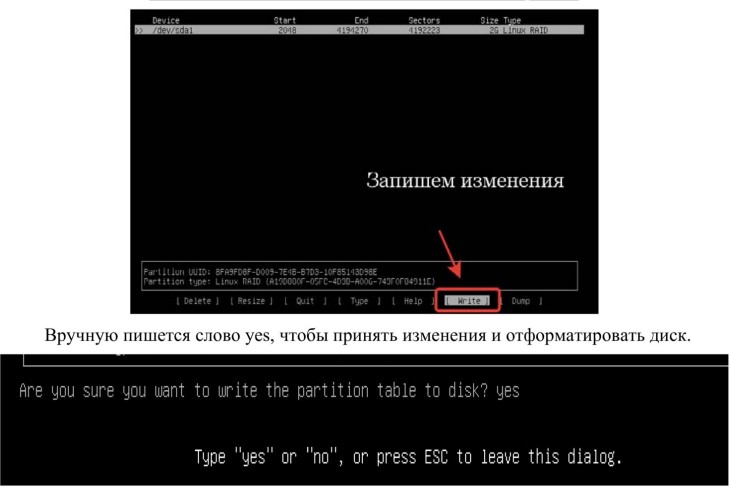
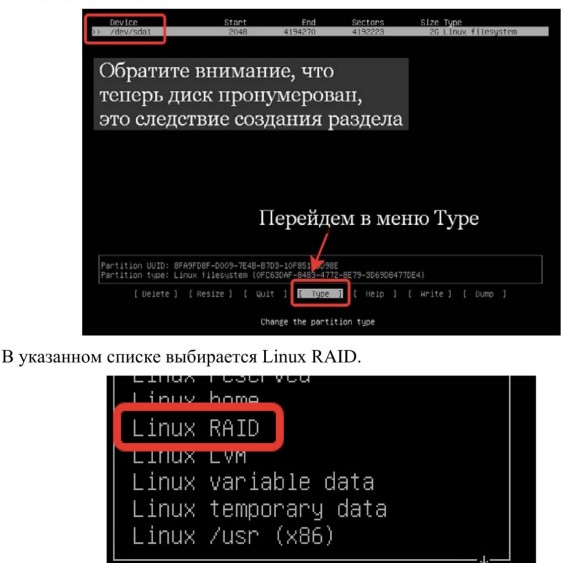
в моем случае 4 диска размером 1 гб это sdb sdc sdd sde



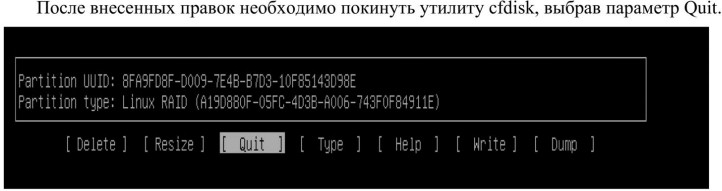
* 1. cfdisk /dev/sdb

3. 

4. Enter

5.

6.

7.

1. С пункта 2 повторить действия со всеми остальными дисками (sdb, sdc, sdd, sde)
2. mdadm --create /dev/md0 --level=5 --raid-devices=4

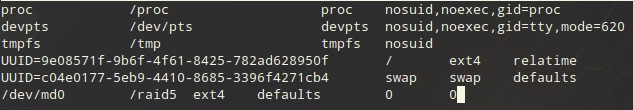
/dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1

1. mdadm --detail --scan --verbose | tee -a

/etc/mdadm.conf

1. mkfs.ext4 /dev/md0
2. mkdir /raid5
3. vim /etc/fstab
   1. добавить в конец, пишем через табуляцию, а не пробел:

/dev/md0 /raid5 ext4 defaults 0 0



1. reboot
2. systemctl restart nfs
3. mkdir /raid5/nfs
4. chmod 777 /raid5/nfs
5. vim /etc/exports
   1. в конец добавляем:

/raid5/nfs 10.10.10.110(rw,sync) 20.20.20.100(rw,sync)

b.

WEB-L

10.10.10.110-адрес WEB-L

20.20.20.100-адрес WEB-R

1. mkdir /mnt/nfs
2. vim /etc/fstab

a. добавляем в конец, пишем через табуляцию 10.10.10.100:/raid5/nfs /mnt/nfs nfs rw,sync 0 0

WEB-R

1. mkdir /mnt/nfs
2. vim /etc/fstab

a. добавляем в конец, пишем через табуляцию 10.10.10.100:/raid5/nfs /mnt/nfs nfs rw,sync 0 0

1. *Запустите сервис MediaWiki используя docker на сервере WEB-L*

WEB-L

* 1. systemctl disable —now ahttpd
  2. systemctl disable --now alteratord
  3. vim ~/wiki.yml

a. меняем так как надо по заданию

version: '3' services:

MediaWiki: container\_name: wiki image: mediawiki restart: always ports:

- 8080:80

links:

* database volumes:
* images:/var/www/html/images

# - ./LocalSettings.php:/var/www/html/LocalSe ttings.php

database: container\_name: db image: mysql restart: always environment:

MYSQL\_DATABASE: mediawiki MYSQL\_USER: wiki MYSQL\_PASSWORD: DEP@ssw0rd

MYSQL\_RANDOM\_ROOT\_PASSWORD: 'toor'

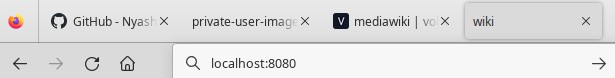
volumes:

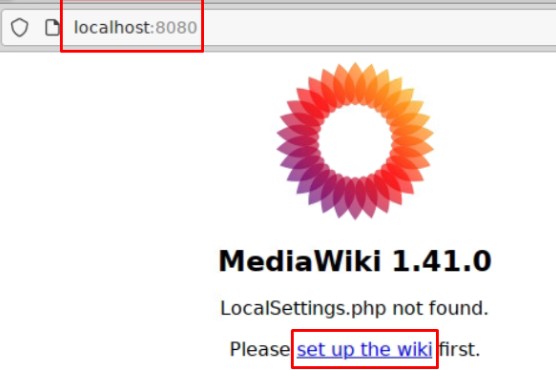
- dbvolume:/var/lib/mysql

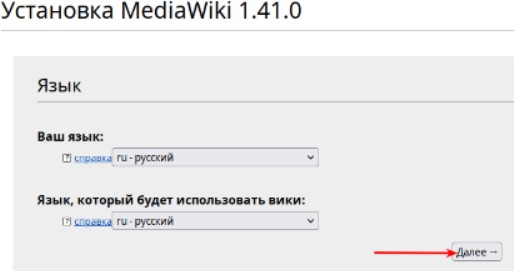
volumes: images: dbvolume:

external: true

* 1. systemctl enable --now docker
  2. docker volume create dbvolume
  3. cd ~
  4. docker-compose -f wiki.yml up -d
  5. заходим в mozila, пишем в строке url: localhost:8080



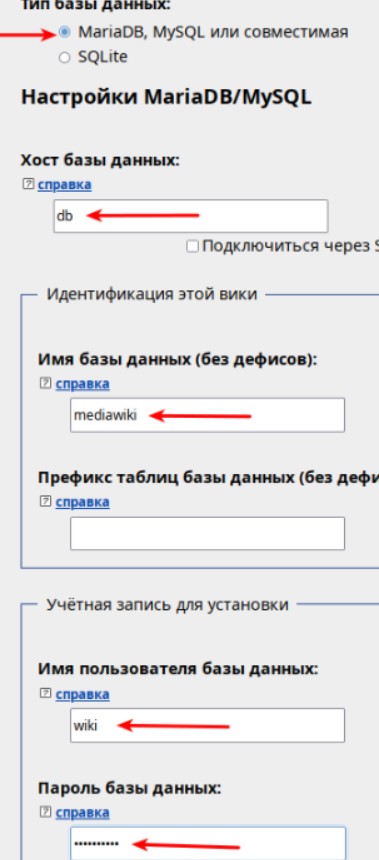


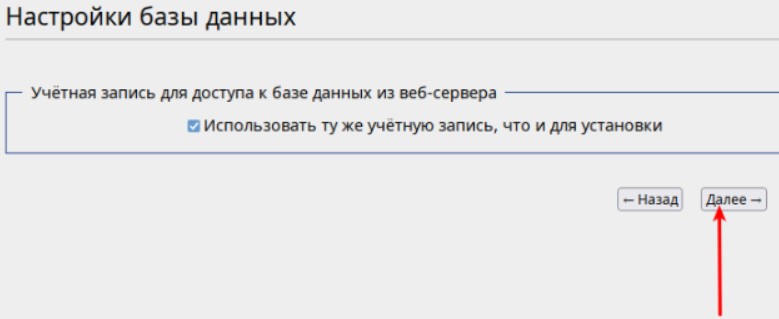


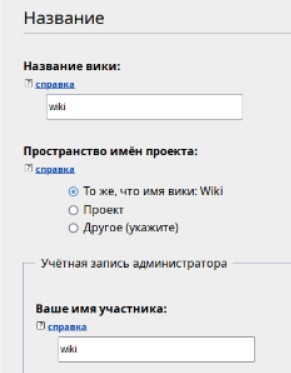
Выбираем MariaDB, MySQL или совместимая хост базы данных: db (по заданию)

имя базы данных: mediawiki (по заданию)

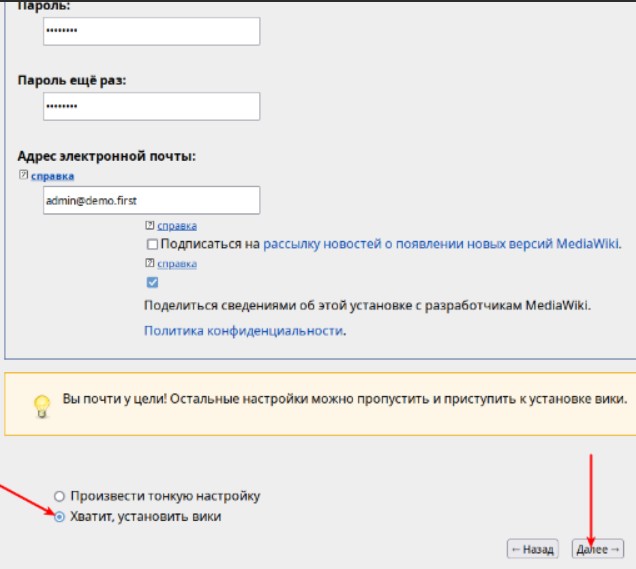
имя пользователя базы данных: wiki (по заданию) Пароль: DEP@ssw0rd (по заданию)

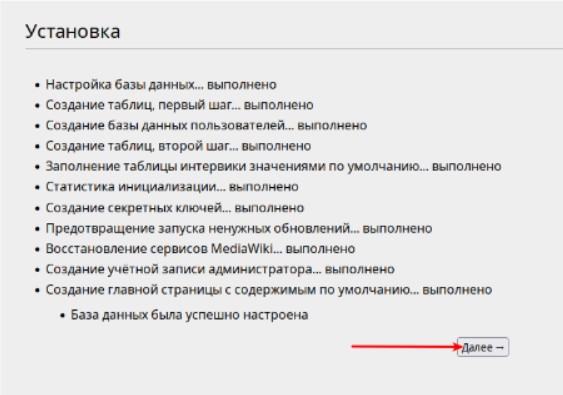


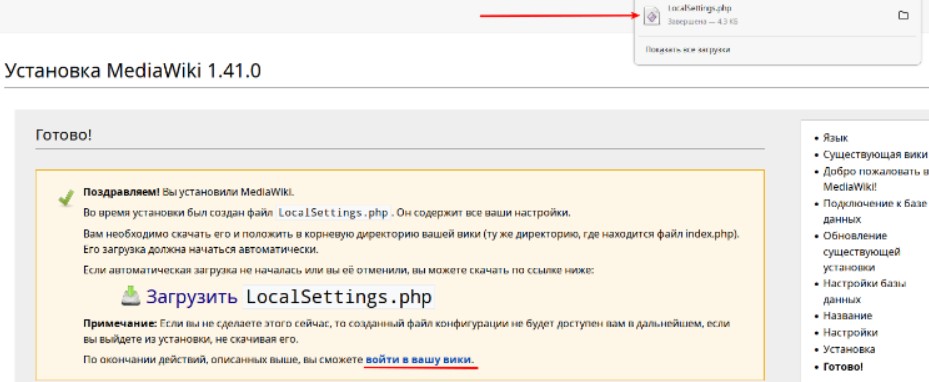




1. Пароль: DEP@ssw0rd, можно без почты







cp /home/user/Загрузки/LocalSettings.php

~/LocalSettings.php vim ~/wiki.yml

расскоментируем

- ./LocalSettings.php:/var/www/html/LocalSettings.php vim ~/LocalSettings.php

$wgServer = “http://mediawiki.au.team:8080” docker-compose -f wiki.yml stop

docker-compose -f wiki.yml up -d WEB-R

1. systemctl disable --now ahttpd systemctl disable --now alteratord

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

CLI

* 1. apt-get install yandex-browser -y
  2. запустить НЕ от root с помощью команды: yandex-browser-stable

запустить от root с помощью команды: yandex-browser-stable --no-sandbox

Если DNS сервер не работает, systemctl status bind выдает ошибки, надо systemctl restart bind на DNS сервере