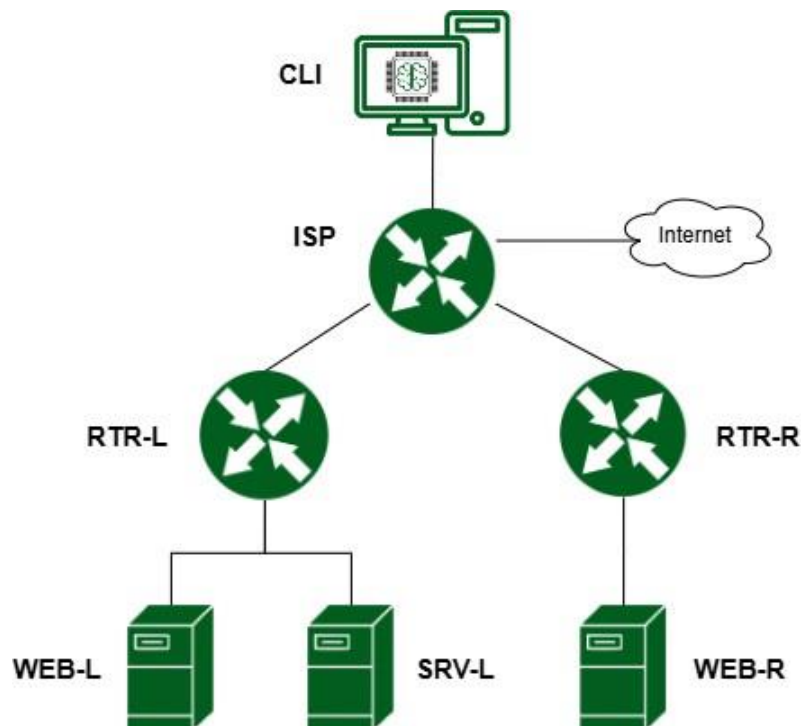


## Задание экзамена «Администрирование сетей передачи информации»



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/ipv4address      10.10.10.1/24
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/ipv4address      150.150.150.10/28
cp -r /etc/net/ifaces/ens18/ /etc/net/ifaces/ens19/
vim /etc/net/ifaces/ens19/ipv4address      20.20.20.1/24
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/ipv4address      10.10.10.100/24
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/ipv4address      10.10.10.110/24
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/ipv4address      20.20.20.100/24
vim /etc/net/ifaces/ens18/ipv4route        default via 20.20.20.1
reboot
```

2. 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;
    }
}
```

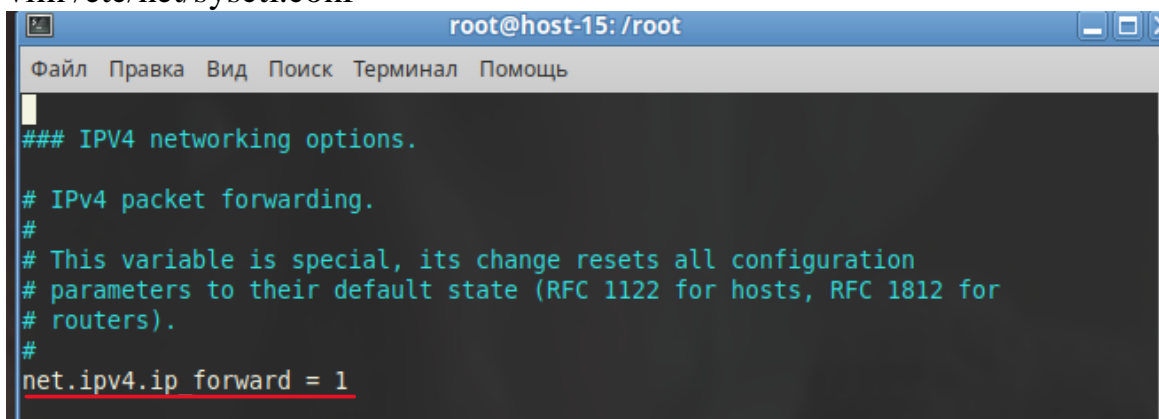


```
GNU nano 7.2 /etc/nftables/nftables.nft
# you can find examples in /usr/share/nftables/
flush ruleset;
table inet filter {
    chain input {
        type filter hook input priority 0;
    }
    chain forward {
        type filter hook forward priority 0;
    }
    chain output {
        type filter hook output priority 0;
    }
}
table ip nat {
    chain postrouting {
        type nat hook postrouting priority 0;
        oifname enp0s3 masquerade;
    }
}
```

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

systemctl enable --now nftables

vim /etc/net/sysctl.conf



```
root@host-15: /root
Файл Правка Вид Поиск Терминал Помощь
### IPV4 networking options.

# IPv4 packet forwarding.
#
# This variable is special, its change resets all configuration
# parameters to their default state (RFC 1122 for hosts, RFC 1812 for
# routers).
#
net.ipv4.ip forward = 1
```

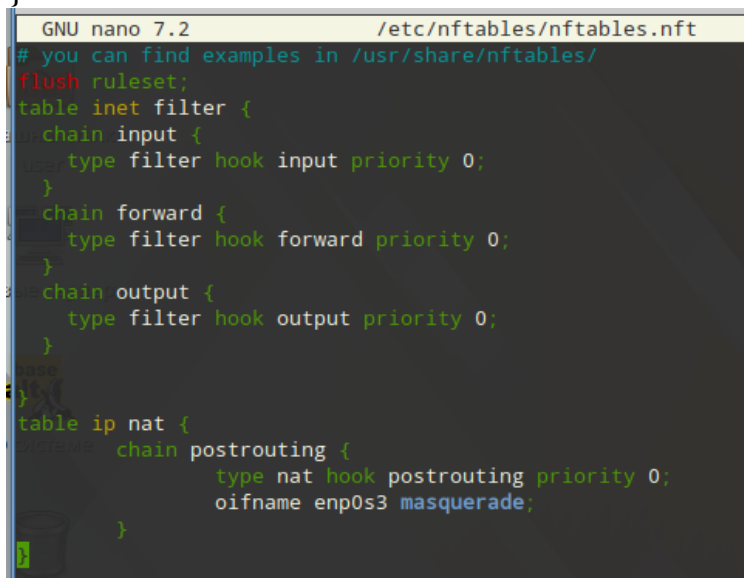
reboot

### RTR-L:

```
apt-get update
apt-get install nftables -y
vim /etc/nftables/nftables.nft
```

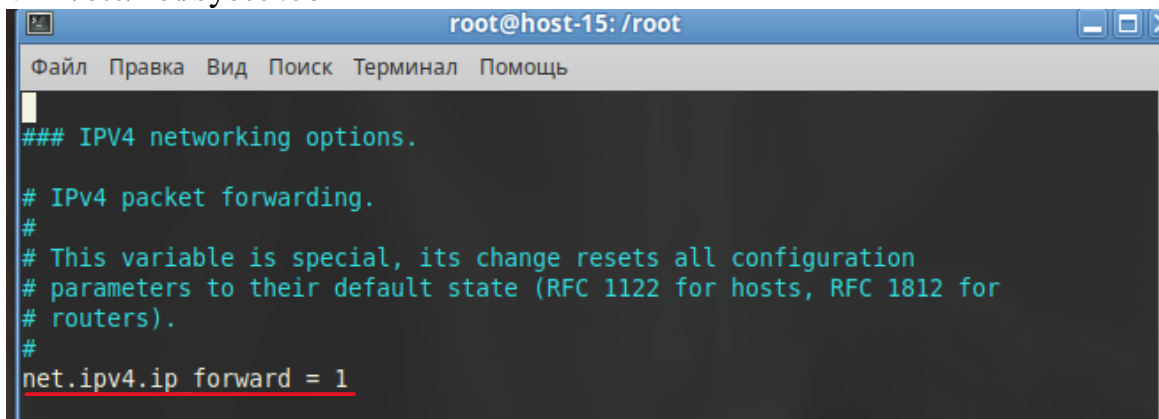
В конфиге прописать ниже после всего:

```
table ip nat {
    chain postrouting {
        type nat hook postrouting priority 0;
        oifname ens18 masquerade;
    }
}
```



```
GNU nano 7.2 /etc/nftables/nftables.nft
# you can find examples in /usr/share/nftables/
flush ruleset;
table inet filter {
    chain input {
        type filter hook input priority 0;
    }
    chain forward {
        type filter hook forward priority 0;
    }
    chain output {
        type filter hook output priority 0;
    }
}
table ip nat {
    chain postrouting {
        type nat hook postrouting priority 0;
        oifname enp0s3 masquerade;
    }
}
```

```
nft -f /etc/nftables/nftables.conf (не должно быть ошибок)
systemctl enable --now nftables
vim /etc/net/sysctl.conf
```



```
root@host-15: /root
Файл Правка Вид Поиск Терминал Помощь
### IPV4 networking options.

# IPv4 packet forwarding.
#
# This variable is special, its change resets all configuration
# parameters to their default state (RFC 1122 for hosts, RFC 1812 for
# routers).
#
net.ipv4.ip forward = 1
```

reboot

### **RTR-R:**

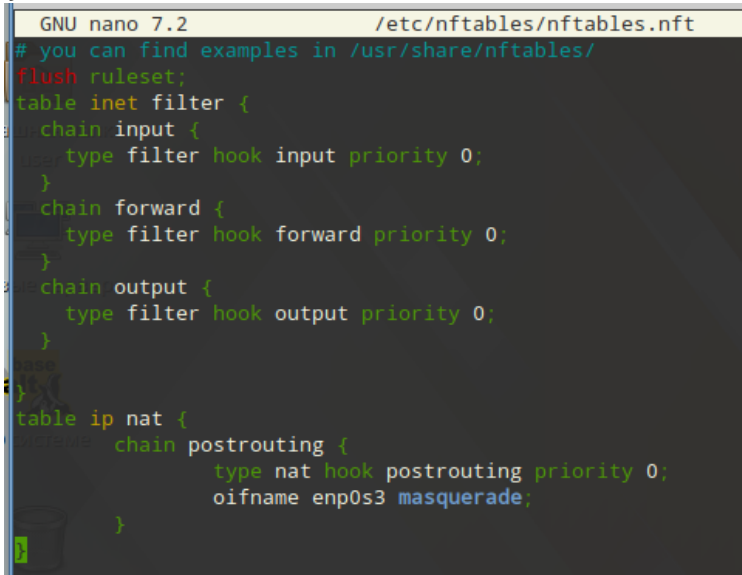
```
apt-get update
apt-get install nftables -y
vim /etc/nftables/nftables.nft
```

В конфиге прописать ниже после всего:

```
table ip nat {
    chain postrouting {
```

```
type nat hook postrouting priority 0;
oifname ens18 masquerade;
```

```
}
```

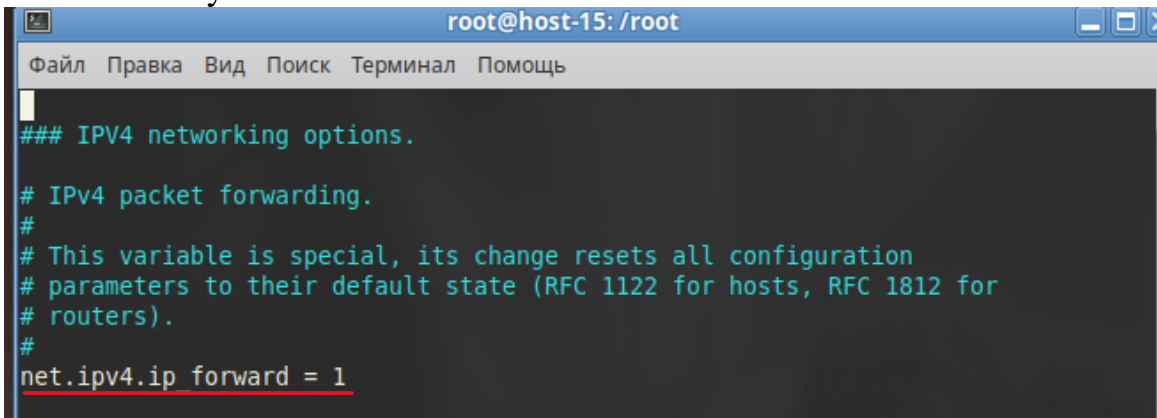
A screenshot of a terminal window showing the nano 7.2 editor editing the file /etc/nftables/nftables.nft. The configuration includes a flush rule, an inet filter with input, forward, and output chains, and an ip nat table with a postrouting chain for masquerading on the ens0s3 interface.

```
GNU nano 7.2 /etc/nftables/nftables.nft
# you can find examples in /usr/share/nftables/
flush ruleset;
table inet filter {
  chain input {
    type filter hook input priority 0;
  }
  chain forward {
    type filter hook forward priority 0;
  }
  chain output {
    type filter hook output priority 0;
  }
}
table ip nat {
  chain postrouting {
    type nat hook postrouting priority 0;
    oifname ens0s3 masquerade;
  }
}
```

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

systemctl enable --now nftables

vim /etc/net/sysctl.conf

A screenshot of a terminal window showing the root user at host-15 editing the /root directory. The terminal displays the configuration for net.ipv4.ip\_forward, which is set to 1.

```
root@host-15: /root
Файл Правка Вид Поиск Терминал Помощь
### IPV4 networking options.

# IPv4 packet forwarding.
#
# This variable is special, its change resets all configuration
# parameters to their default state (RFC 1122 for hosts, RFC 1812 for
# routers).
#
net.ipv4.ip forward = 1
```

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

3. 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 происходить без применения туннеля и шифрования.
  - Трафик, идущий по туннелю между регионами по внутренним адресам, не должен транслироваться.

### RTR-L:

vim /etc/gre.up

Пишем:

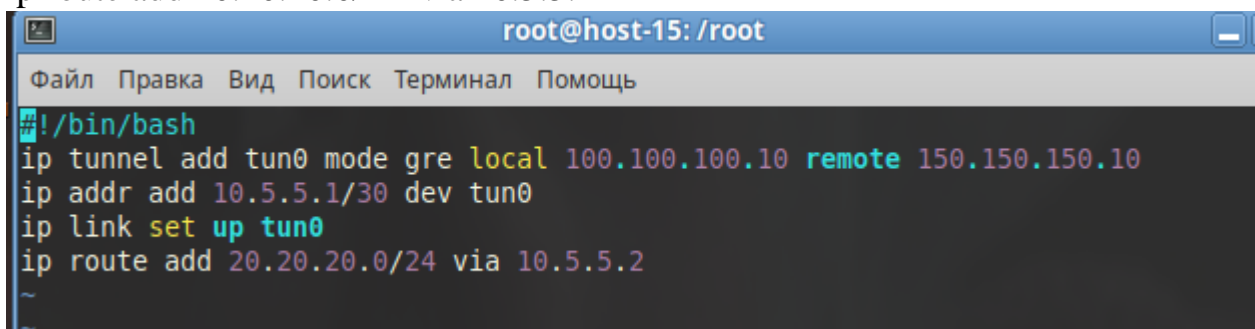
```
#!/bin/bash
```

```
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
```



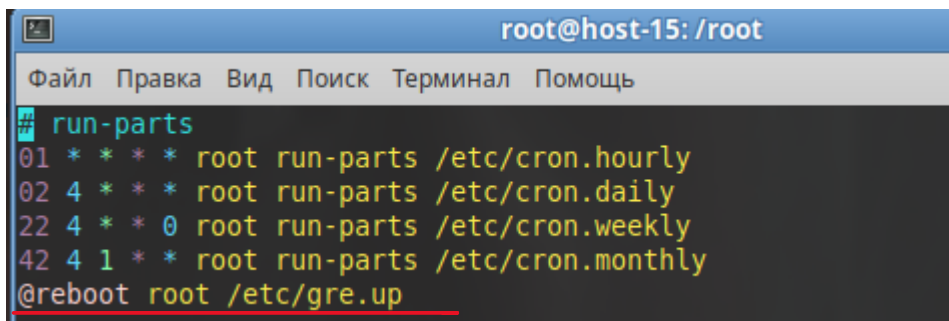
```
root@host-15: /root
Файл  Правка  Вид  Поиск  Терминал  Помощь
#!/bin/bash
ip tunnel add tun0 mode gre local 100.100.100.10 remote 150.150.150.10
ip addr add 10.5.5.1/30 dev tun0
ip link set up tun0
ip route add 20.20.20.0/24 via 10.5.5.2
~
```

Сохраняем файл

```
chmod +x /etc/gre.up
```

```
/etc/gre.up
```

```
vim /etc/crontab
```



```
root@host-15: /root
Файл  Правка  Вид  Поиск  Терминал  Помощь
run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
@reboot root /etc/gre.up
~
```

```
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
```

```
# Sample VPN connections
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
#conn sample-self-signed
```

vim /etc/strongswan/ipsec.secrets

Пишем:

```
100.100.100.10 150.150.150.10 : PSK "P@ssw0rd"
```

```
GNU nano 7.2 /etc/strongswan/ipsec.secrets
# ipsec.secrets - strongSwan IPsec secrets file
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
```

```
root@host-15: /root
Файл Правка Вид Поиск Терминал Помощь
#!/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

```
root@host-15: /root
Файл Правка Вид Поиск Терминал Помощь
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
@reboot root /etc/gre.up
```

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

```
# Sample VPN connections
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

# conn sample-self-signed
```

vim /etc/strongswan/ipsec.secrets

Пишем:

100.100.100.10 150.150.150.10 : PSK "P@ssw0rd"

```
GNU nano 7.2 /etc/strongswan/ipsec.secrets
# ipsec.secrets - strongSwan IPsec secrets file
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:

```
Security Associations (1 up, 0 connecting):  
    vpn[2]: ESTABLISHED 26 seconds ago, 150.150.150.10[150.150.150.10]...10  
0.100.100.10[100.100.100.10]  
    vpn{2}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: cc032917_i c4a1b4cf_o  
    vpn{2}:  0.0.0.0/0[gre] === 0.0.0.0/0[gre]
```

Если не так то пробуем

ipsec update

ipsec restart

#### 4. 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

```
Port 2024
AllowUsers sshuser
MaxAuthTries 2
Banner /etc/openssh/banner
PasswordAuthentication yes
#AddressFamily any
```

Vim /etc/openssh/banner

Authorized access only

```
GNU nano 7.2 /etc/openssh/banner
Authorized access only
```

systemctl enable --now sshd

useradd sshuser

passwd sshuser (P@ssw0rd)

##### **Проверка:**

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

```
ssh: connect to host 10.10.10.100 port 2024: Connection refused
[root@cli ~]# ssh sshuser@10.10.10.110 -p 2024
The authenticity of host '[10.10.10.110]:2024 ([10.10.10.110]:2024)' can't be es-
tablished.
ED25519 key fingerprint is SHA256:NtqbAskbAvR4GjeC5qhSlsEu7MWZ8IySrYjhKWlbGDw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.10.110]:2024' (ED25519) to the list of known
hosts.
Authorized access only
sshuser@10.10.10.110's password:
[sshuser@web-1 ~]$
```

## 5. 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 BM	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

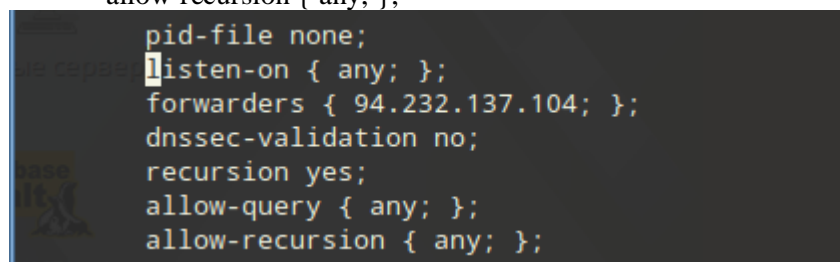
### SRV-L:

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

**systemctl enable --now bind**

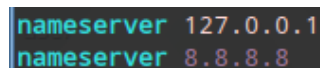
**vim /etc/bind/options.conf**

```
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;};
};

```

```

root@srv-l: /root
Файл  Правка  Вид  Поиск  Терминал  Помощь

zone "au.team" {
    type master;
    file "db.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; };
};
31,1  Окончание

```

cd /etc/bind/zone/

```

[root@srv-l ~]# cd /etc/bind/zone
[root@srv-l zone]# ls
127.in-addr.arpa  empty  localdomain  localhost  slave
[root@srv-l zone]# cp localhost au.team
[root@srv-l zone]# cp localhost left.reverse
[root@srv-l zone]# cp localhost right.reverse
[root@srv-l zone]# cp localhost cli.reverse
[root@srv-l zone]#

```

Задаём необходимые права:

```

[root@srv-l zone]# chmod 777 au.team
[root@srv-l zone]# chmod 777 right.reverse
[root@srv-l zone]# chmod 777 left.reverse
[root@srv-l zone]# chmod 777 cli.reverse
[root@srv-l zone]#

```

vim /etc/bind/zone/au.team

```
GNU nano 7.2 au.team
$TTL 1D
@ IN SOA au.team. root.au.team. (
    2024021400 ; serial
    12H        ; refresh
    1H         ; retry
    1W         ; expire
    1H         ; ncache
)
@ 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
mediawiki IN CNAME web-l
```

vim /etc/bind/zone/left.reverse

```
GNU nano 7.2 left.reverse Изменён
$TTL 1D
@ IN SOA 10.10.10.in-addr.arpa. root.10.10.10.in-addr.arpa. (
    2024021400 ; serial
    12H        ; refresh
    1H         ; retry
    1W         ; expire
    1H         ; ncache
)
@ 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.
```

vim /etc/bind/zone/right.reverse

```
GNU nano 7.2 right.reverse Изменён
$TTL 1D
@ IN SOA 20.20.20.in-addr.arpa. root.20.20.20.in-addr.arpa. (
    2024021400 ; serial
    12H        ; refresh
    1H         ; retry
    1W         ; expire
    1H         ; ncache
)
@ IN NS au.team.
@ IN A 20.20.20.100
1 PTR rtr-r.au.team.
100 PTR web-r.au.team.
```

vim /etc/bind/zone/cli.reverse

```
Файл  Правка  Вид  Поиск  Терминал  Помощь
GNU nano 7.2                                cli.reverse                                Изменён
$TTL      1D
@         IN      SOA      35.35.35.in-addr.arpa. root.35.35.35.in-addr.arpa. (
                                2024021400      ; serial
                                12H             ; refresh
                                1H              ; retry
                                1W              ; expire
                                1H              ; ncache
                                )
@         IN      NS       au.team.
@         IN      A        35.35.35.1
1         PTR     isp.au.team.
10        PTR     cli.au.team.
```

**systemctl restart bind**  
**named-checkconf**  
**named-checkconf -z**

**vim /etc/resolv.conf**

```
GNU nano 7.2                                /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 127.0.0.1
```

**reboot**

## **WEB-R:**

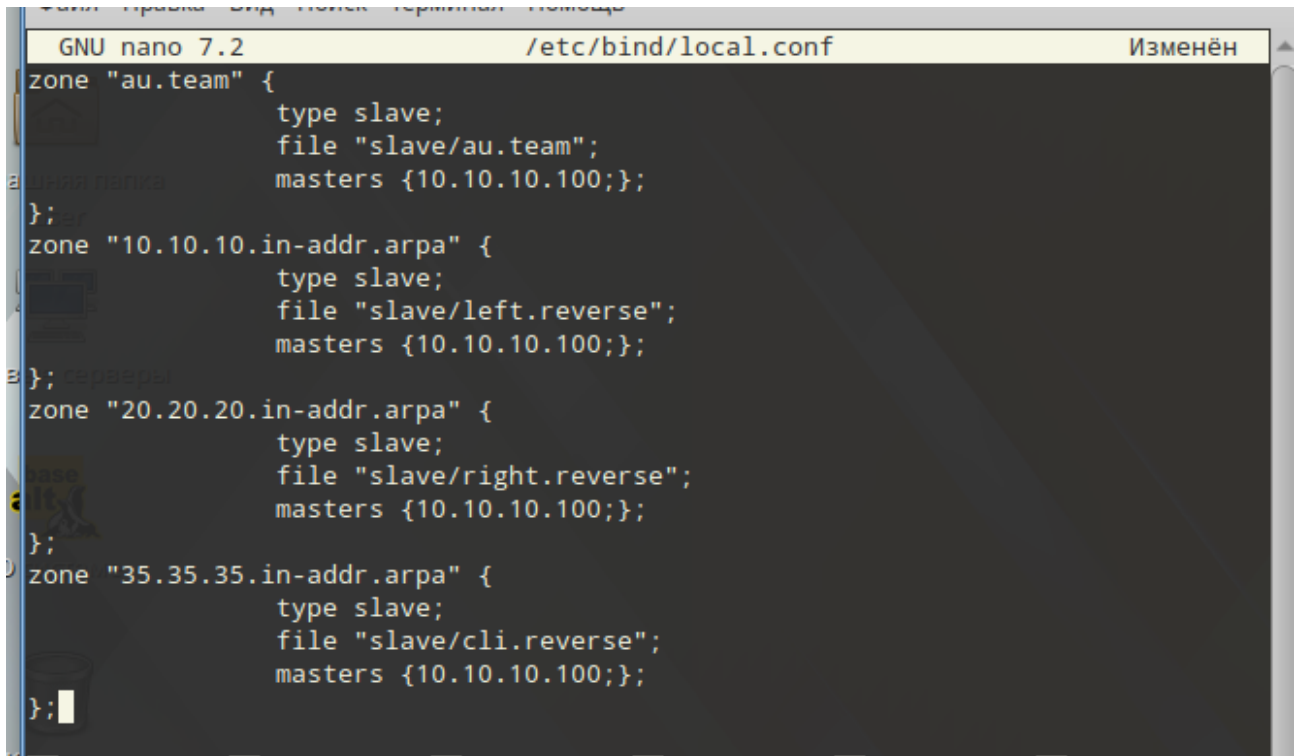
**apt-get update && apt-get install bind bind-utils**

**systemctl enable --now bind**

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

```
// disables the use of a PID file
pid-file none;
listen-on { any; };
forwarders { 10.10.10.100; };
dnssec-validation no;
recursion yes;
allow-query { any; };
allow-recursion { any; };
/*
```

vim /etc/bind/local.conf

A screenshot of the GNU nano 7.2 text editor editing the file /etc/bind/local.conf. The editor's status bar at the top shows 'GNU nano 7.2', the file path '/etc/bind/local.conf', and the state 'Изменён'. The file content shows four zone definitions, all configured as slaves with master 10.10.10.100. The zones are: 'au.team' with file 'slave/au.team'; '10.10.10.in-addr.arpa' with file 'slave/left.reverse'; '20.20.20.in-addr.arpa' with file 'slave/right.reverse'; and '35.35.35.in-addr.arpa' with file 'slave/cli.reverse'. The cursor is at the end of the last zone definition.

```
GNU nano 7.2 /etc/bind/local.conf Изменён
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;};
};
```

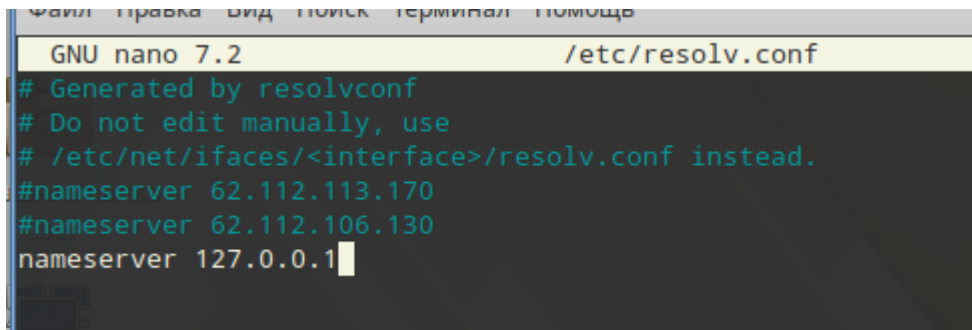
chown named:named /var/lib/bind/zone/slave/

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

systemctl restart bind

systemctl status bind

vim /etc/resolv.conf

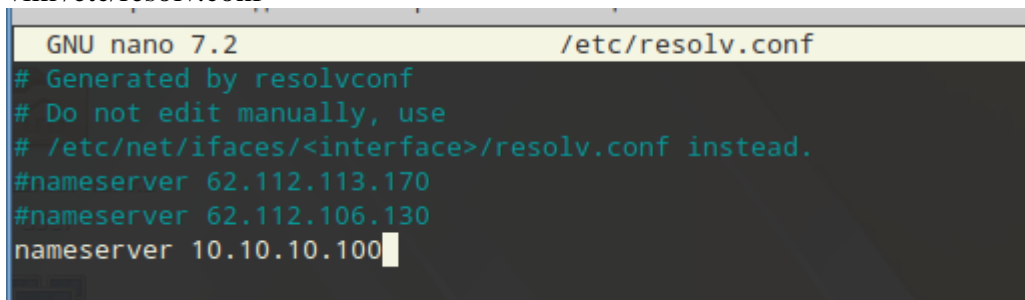
A screenshot of the GNU nano 7.2 text editor editing the file /etc/resolv.conf. The status bar shows 'GNU nano 7.2' and the file path '/etc/resolv.conf'. The file content includes a header with instructions not to edit manually and to use the interface-specific file instead. It lists two commented-out nameservers (62.112.113.170 and 62.112.106.130) and one active nameserver (127.0.0.1) with the cursor at the end of the line.

```
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 127.0.0.1
```

reboot

## CLI:

vim /etc/resolv.conf

A screenshot of the GNU nano 7.2 text editor editing the file /etc/resolv.conf. The status bar shows 'GNU nano 7.2' and the file path '/etc/resolv.conf'. The file content is identical to the previous screenshot, but the active nameserver has been changed from 127.0.0.1 to 10.10.10.100, with the cursor at the end of the line.

```
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 10.10.10.100
```

reboot

## ISP:

vim /etc/resolv.conf

```
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 10.10.10.100
```

reboot

### **RTR-L:**

vim /etc/resolv.conf

```
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 10.10.10.100
```

reboot

### **RTR-R:**

vim /etc/resolv.conf

```
Файл Правка Вид Поиск Терминал Помощь
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 20.20.20.100
```

reboot

### **WEB-L:**

vim /etc/resolv.conf

```
GNU nano 7.2 /etc/resolv.conf
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 62.112.113.170
#nameserver 62.112.106.130
nameserver 10.10.10.100
```

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.



```
[root@rtr-r ~]# nslookup au.team
Server:      20.20.20.100
Address:     20.20.20.100#53

Name:   au.team
Address: 10.10.10.100
```

6. 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

```
server 127.0.0.1
allow 35.35.35.0/28
allow 100.100.100.0/28
allow 150.150.150.0/28
allow 20.20.20.0/24
allow 10.10.10.0/24
local stratum 5
```

55,1 Окончание

Сохраняем его

systemctl restart chronyd

```
[root@isp ~]# systemctl enable --now chronyd.service
Synchronizing state of chronyd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable chronyd
Created symlink /etc/systemd/system/multi-user.target.wants/chronyd.service -> /lib/systemd/system/chronyd.service.

[root@isp ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^? localhost.localdomain    0      6      0      -    +0ns[  +0ns] +/-    0ns

[root@isp ~]#
```

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

```
root@rtr-r: /root
Файл Правка Вид Поиск Терминал Помощь
[root@rtr-r ~]# systemctl enable --now chronyd.service
Synchronizing state of chronyd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable chronyd
Created symlink /etc/systemd/system/multi-user.target.wants/chronyd.service -> /lib/systemd/system/chronyd.service.
[root@rtr-r ~]#
```

## ПРОВЕРКА:

```
[root@isp ~]# chronyc clients
Hostname NTP Drop Int IntL Last Cmd Drop Int Last
=====
localhost.localdomain 0 0 - - - 7 0 0 527
35.35.35.10 6 0 6 - 48 0 0 - -
100.100.100.10 15 0 5 - 8 0 0 - -
150.150.150.10 9 0 6 - 2 0 0 - -
[root@isp ~]#
```

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

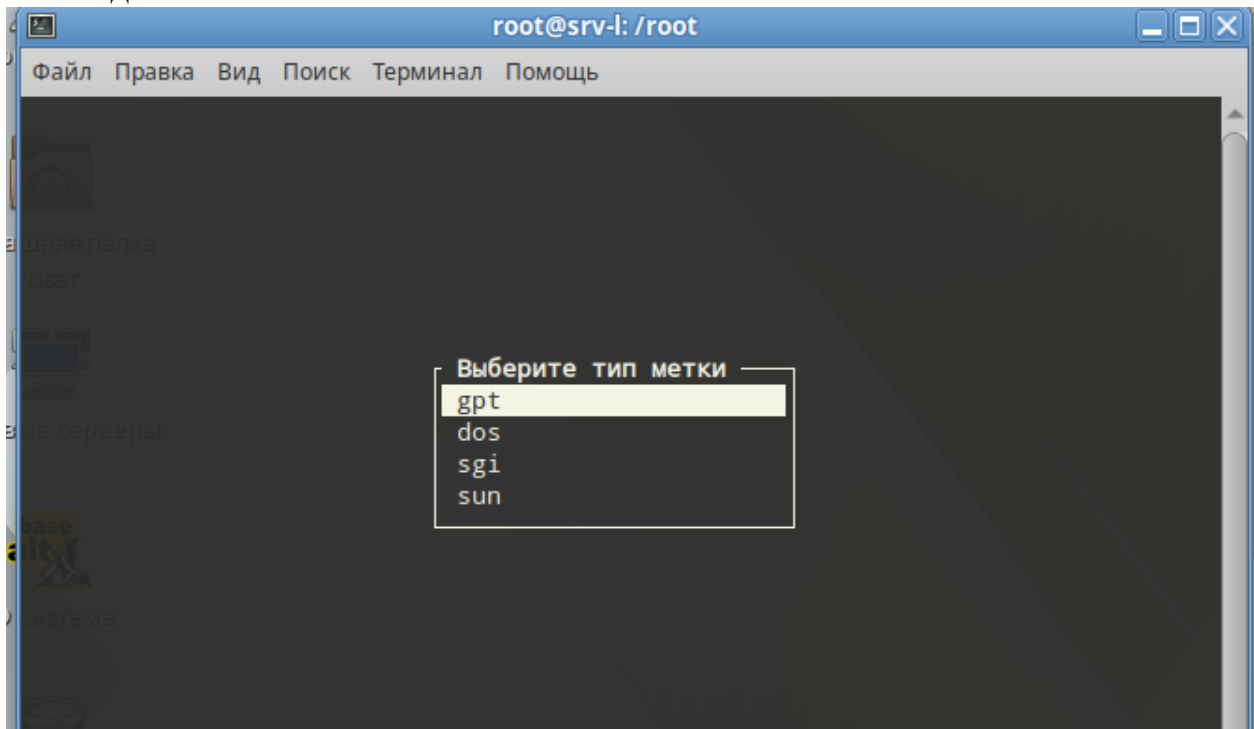
7. Сконфигурируйте файловое хранилище:

- 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 автмонтирование в папку /mnt/nfs

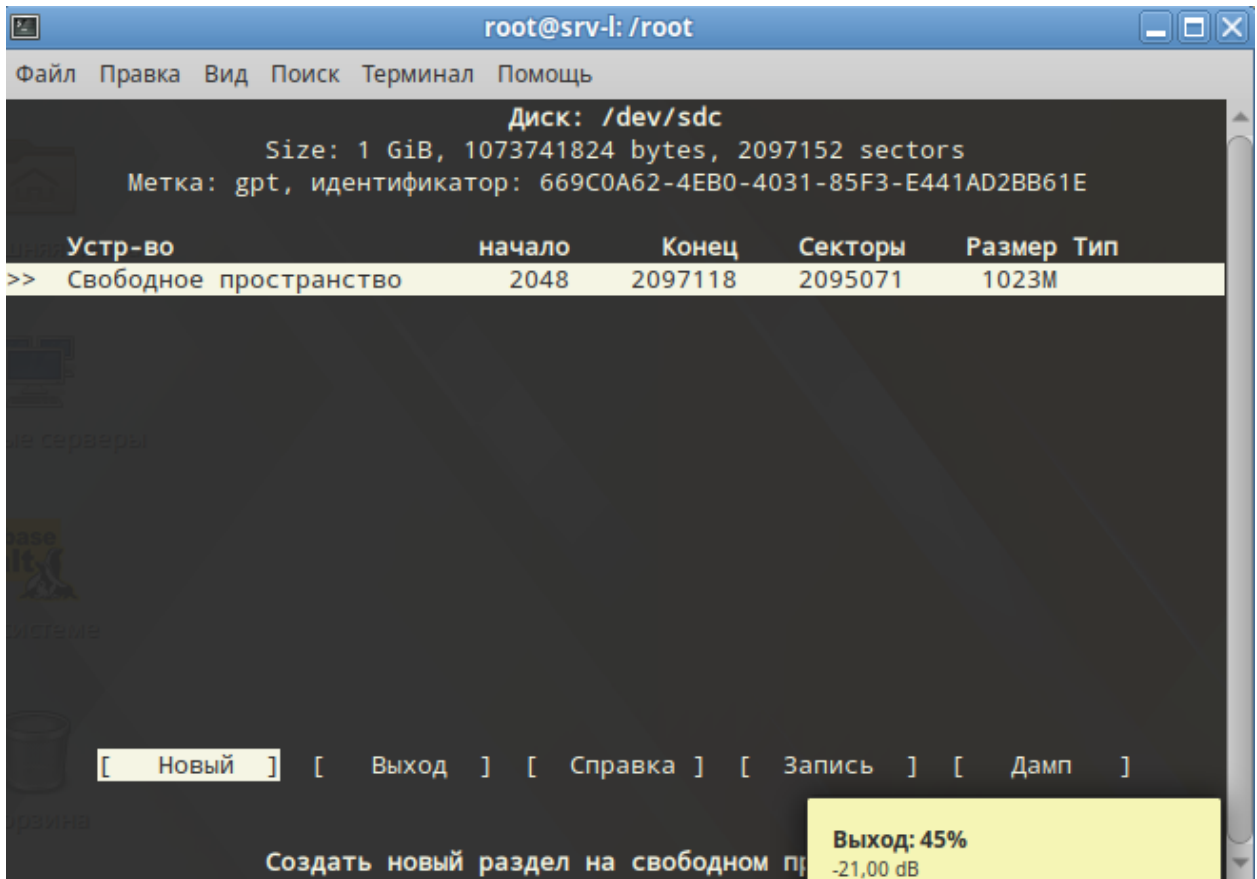
Пишем lsblk, смотрим диски

```
[root@srv-l ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0    0   40G  0 disk
├─sda1      8:1    0   959M  0 part [SWAP]
└─sda2      8:2    0   39,1G  0 part /
sdb         8:16   0    1G  0 disk
sdc         8:32   0    1G  0 disk
sdd         8:48   0    1G  0 disk
sde         8:64   0    1G  0 disk
sr0        11:0    1 1024M  0 rom
[root@srv-l ~]# cfdisk /dev/sdb
```

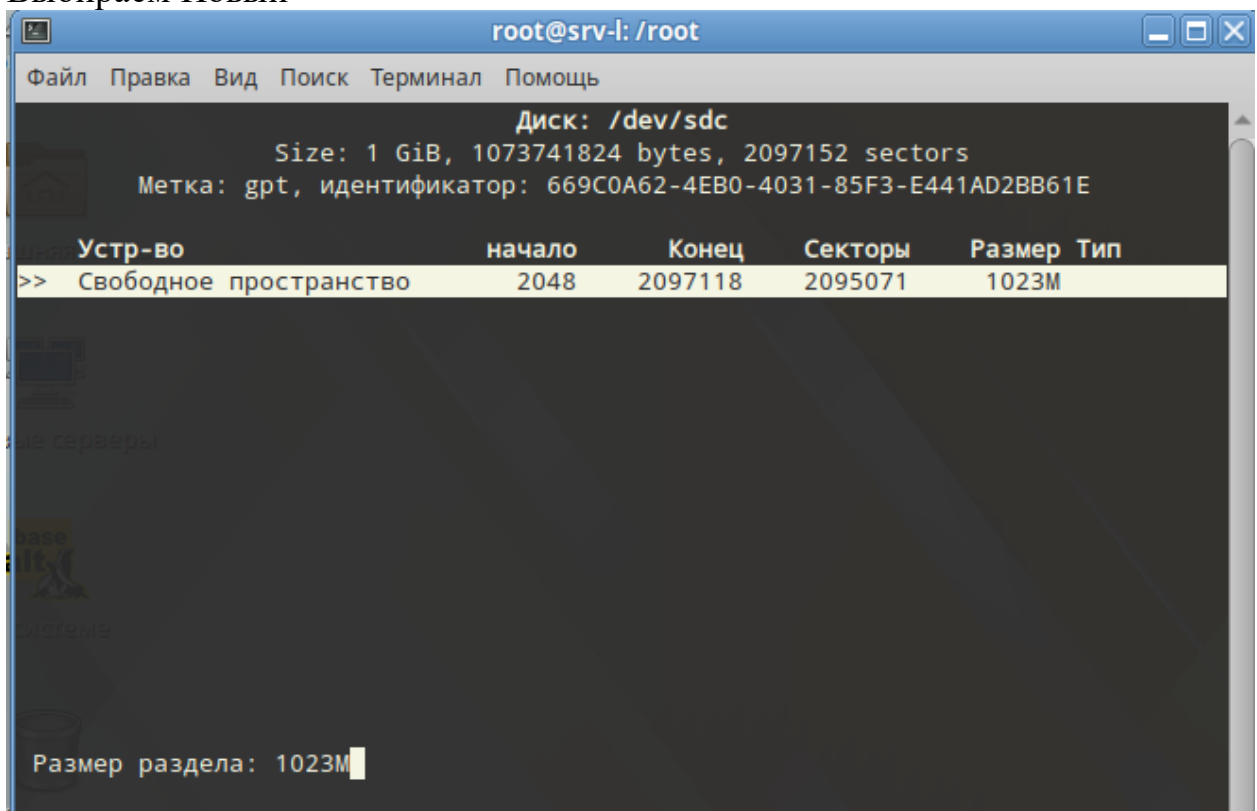
Команда: cfdisk /dev/sdc



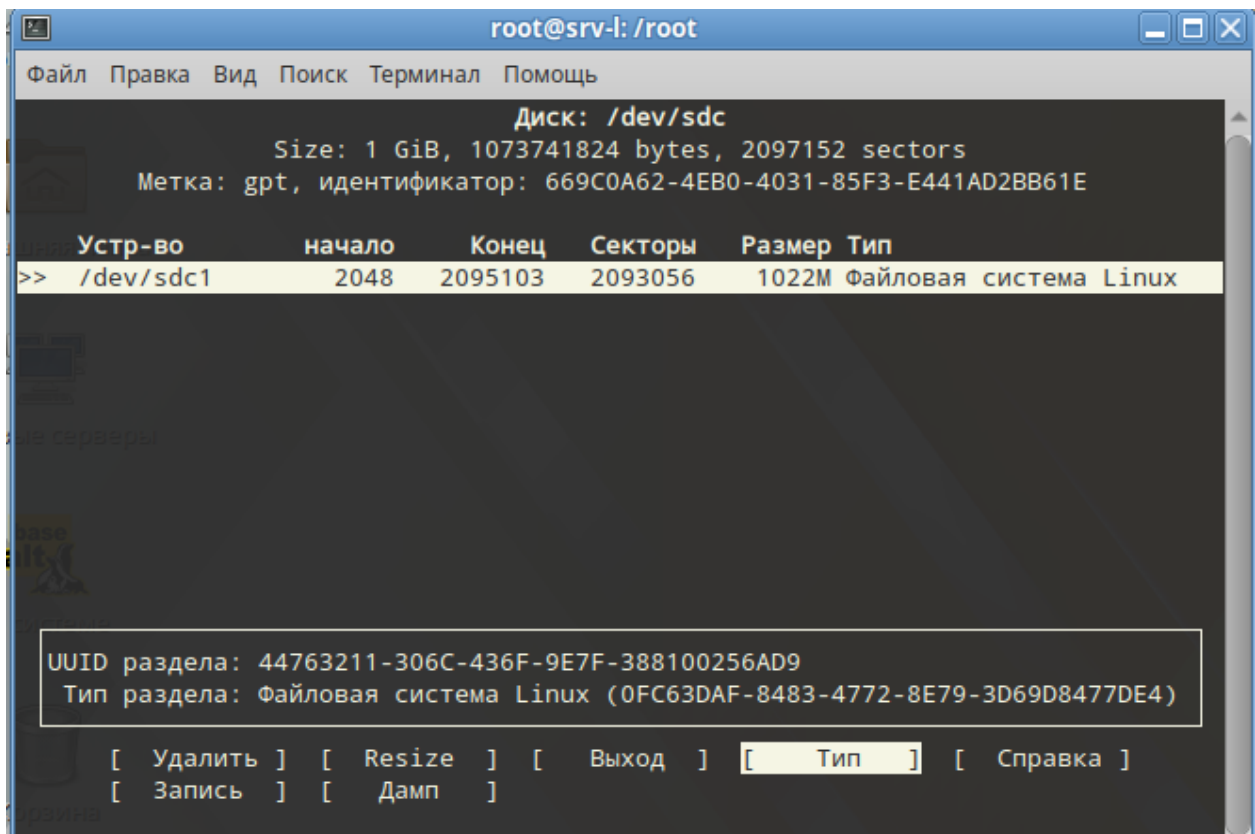
Нажимаем enter



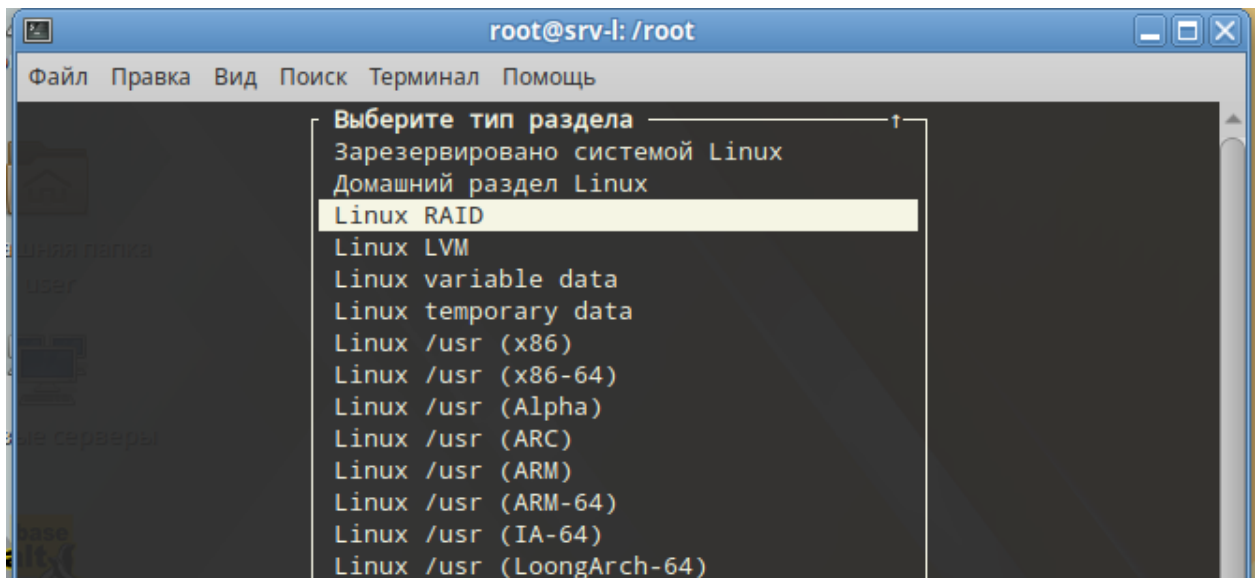
Выбираем Новый



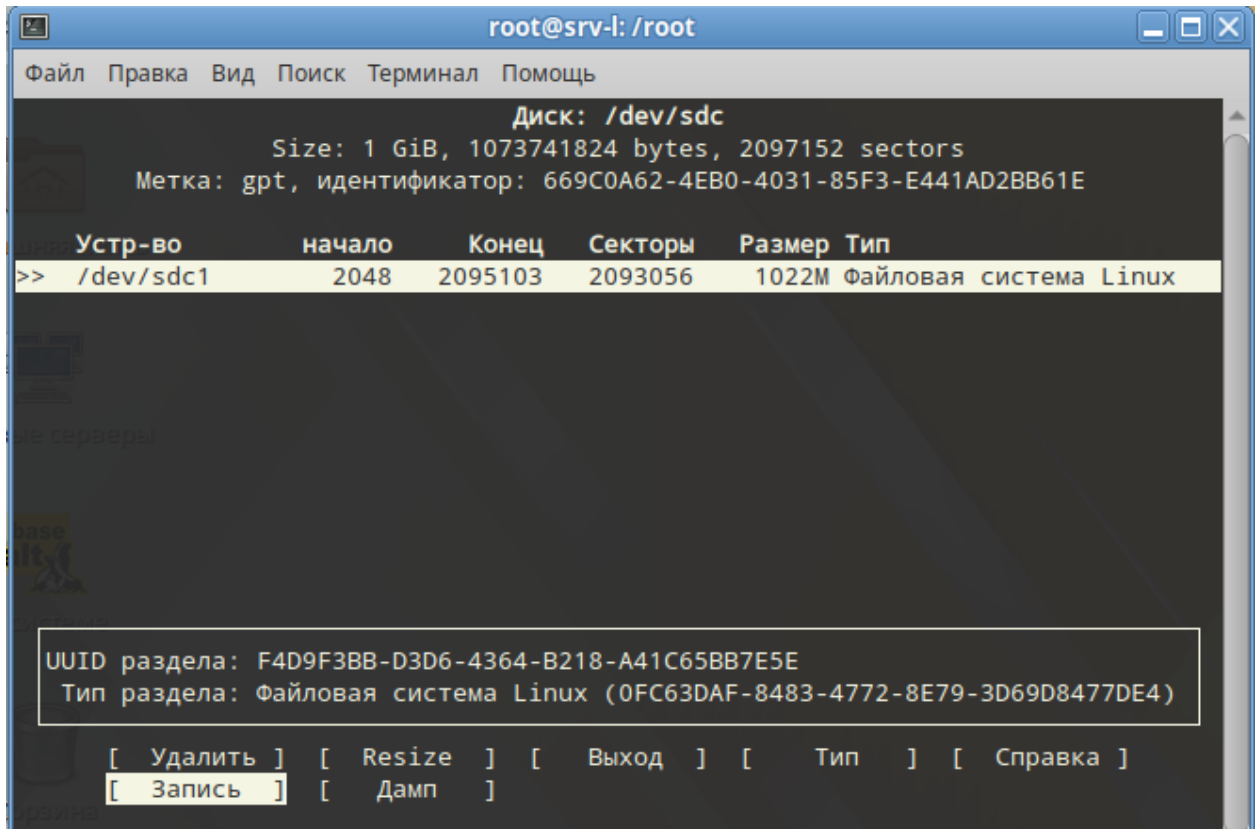
Нажимаем enter



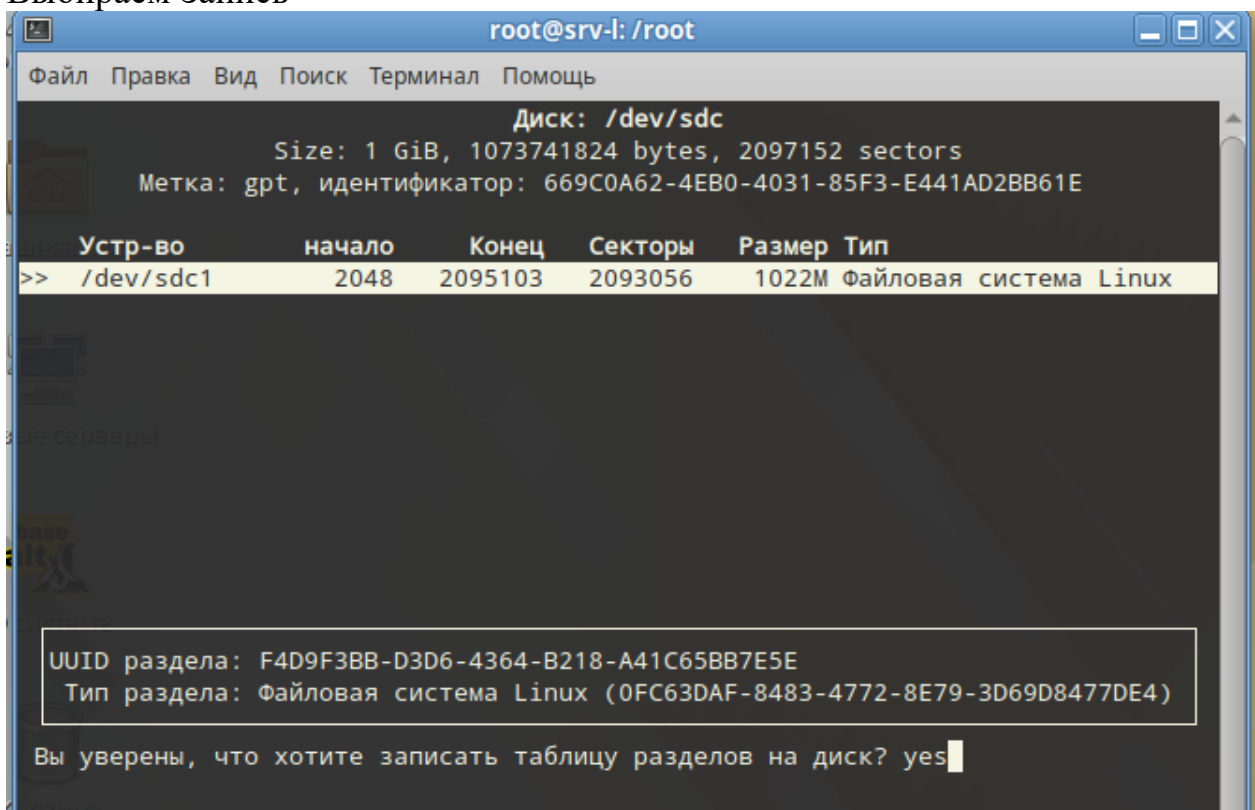
Нажимаем Тип



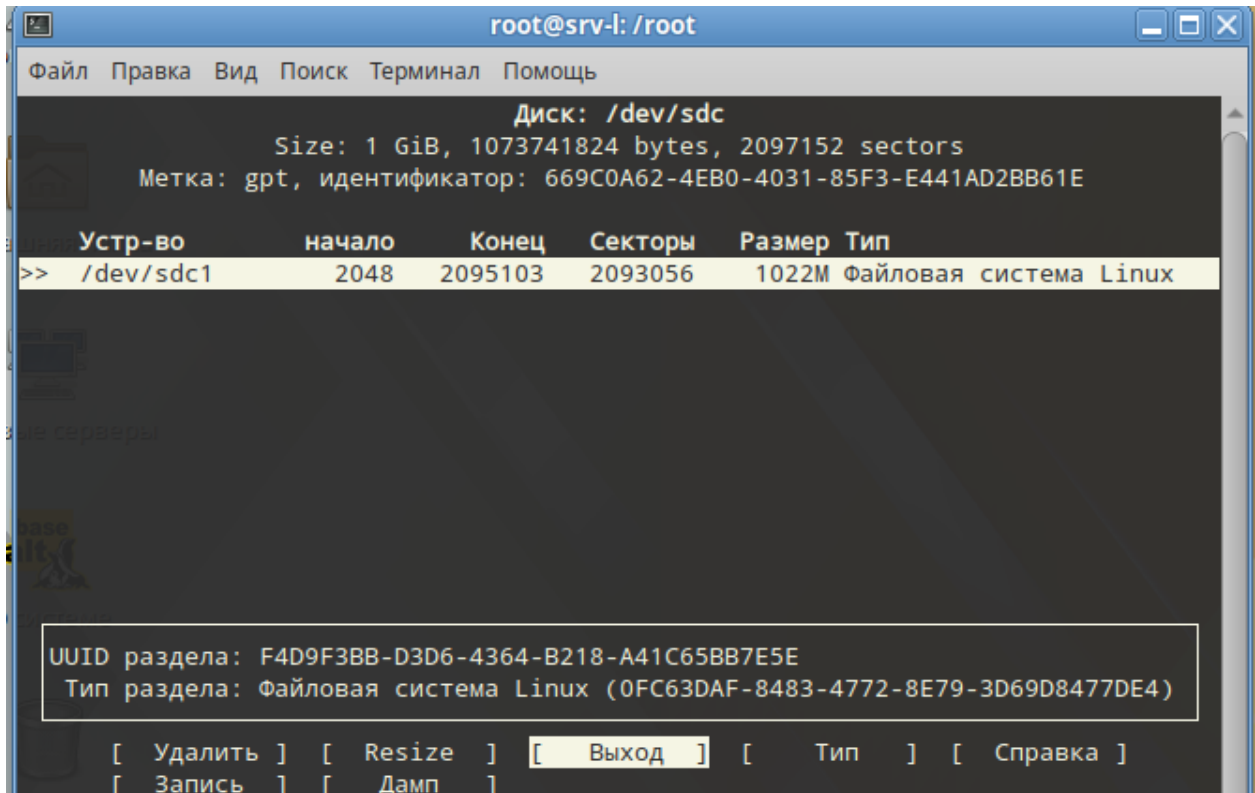
Выбираем Linux Raid, нажимаем enter



Выбираем Запись



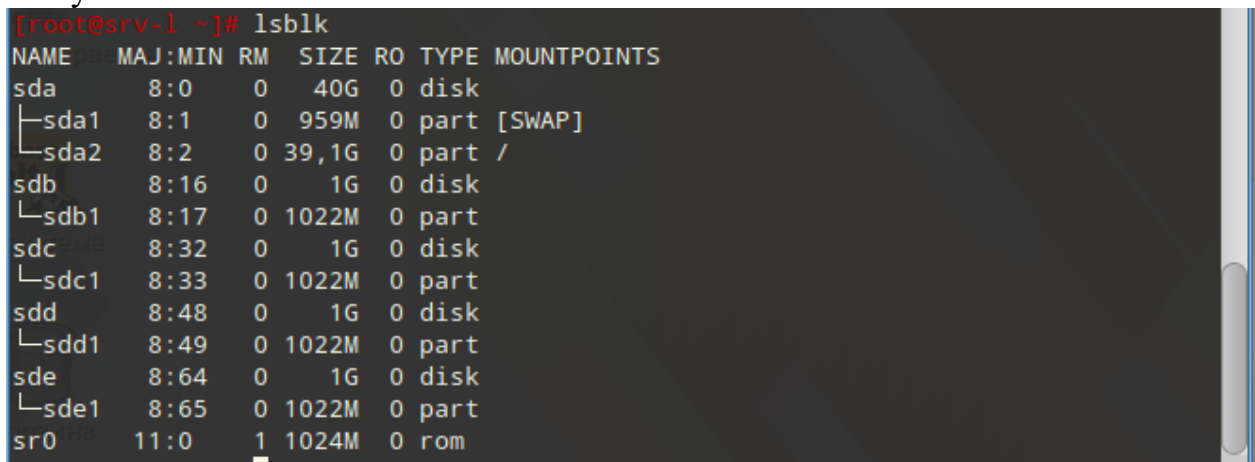
Пишем yes



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
```

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
[root@srv-l ~]# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 1044480K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@srv-l ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sde1[4] sdd1[2] sdc1[1] sdb1[0]
      3133440 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]

unused devices: <none>
[root@srv-l ~]#
```

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

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
[root@srv-l ~]# mdadm --detail --scan --verbose | tee -a /etc/mdadm.conf
ARRAY /dev/md0 level=raid5 num-devices=4 metadata=1.2 name=srv-l.au.team:0 UUID=1fab12e8:38653a86:b3e7d28b:bcd49ad9
      devices=/dev/sdb1,/dev/sdc1,/dev/sdd1,/dev/sde1
[root@srv-l ~]#
```

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
[root@srv-l ~]# make-initrd
[00:00:00] Config file: /etc/initrd.mk
[00:00:02] Generating module dependencies on host ...
[00:00:19] Used features: add-modules add-udev-rules cleanup compress depmod-image gpu-drm kbd locales network plymouth rdshell rootfs system-glibc ucode
[00:00:19] Packed modules: af_packet ahci drm drm_kms_helper drm_ttm_helper evdev hid hid-generic input-leds intel-agp intel-gtt libahci libata scsi_common scsi_mod sd_mod serio_raw sis-agp ttm via-agp vmwgfx
[00:00:19] Unpacked size: 56M
[00:00:19] Image size: 16M
[00:00:19] Image is saved as /boot/initrd-6.1.79-un-def-alt1.img

[root@srv-l ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 40G 0 disk
├─sda1 8:1 0 959M 0 part [SWAP]
└─sda2 8:2 0 39,1G 0 part /
sdb 8:16 0 1G 0 disk
├─sdb1 8:17 0 1022M 0 part
└─┬md0 9:0 0 3G 0 raid5
sdc 8:32 0 1G 0 disk
├─sdc1 8:33 0 1022M 0 part
└─┬md0 9:0 0 3G 0 raid5
sdd 8:48 0 1G 0 disk
├─sdd1 8:49 0 1022M 0 part
└─┬md0 9:0 0 3G 0 raid5
sde 8:64 0 1G 0 disk
├─sde1 8:65 0 1022M 0 part
└─┬md0 9:0 0 3G 0 raid5
```



```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь

[root@srv-l ~]# mkfs.ext4 /dev/md0
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 783360 4k blocks and 195840 inodes
Filesystem UUID: 44e3239f-8d53-496e-9b50-c87e2c0ae89b
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@srv-l ~]#
```

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь

[root@srv-l ~]# mkdir -p /raid5
[root@srv-l ~]#
```

Открываем файл /etc/fstab

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь

proc                /proc              proc              nosuid,noexec,gid=proc
0 0
devpts              /dev/pts           devpts           nosuid,noexec,gid=tty,mode=620
0 0
tmpfs               /tmp               tmpfs            nosuid
0 0
UUID=85af0e0b-2739-4f64-b468-9ae01a0842cc    /                  ext4             relatime
1 1
UUID=2b9a7b33-ec8f-4661-ad6f-92ce6371dc4c    swap              swap             defaults
0 0
/dev/sr0            /media/ALTlinux    udf,iso9660      ro,noauto,user,utf8,nofail,comme
nt=x-gvfs-show 0 0
/dev/md0            /raid5             ext4             defaults         0 0
~#
```

(последняя строчка)

```
[root@srv-l ~]# mount -av
/proc                : already mounted
/dev/pts             : already mounted
/tmp                 : already mounted
/                    : ignored
swap                 : ignored
/media/ALTlinux      : ignored
/raid5               : successfully mounted
[root@srv-l ~]#
```

Перезагружаем машину reboot

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
[user@srv-l ~]$ su-
Password:
[root@srv-l ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda         8:0    0   40G  0 disk
├─sda1      8:1    0   959M  0 part  [SWAP]
├─sda2      8:2    0  39,1G  0 part  /
sdb         8:16   0    1G   0 disk
├─sdb1      8:17   0  1022M  0 part
├─┬md0      9:0    0     3G   0 raid5 /raid5
sdc         8:32   0    1G   0 disk
├─sdc1      8:33   0  1022M  0 part
├─┬md0      9:0    0     3G   0 raid5 /raid5
sdd         8:48   0    1G   0 disk
├─sdd1      8:49   0  1022M  0 part
├─┬md0      9:0    0     3G   0 raid5 /raid5
sde         8:64   0    1G   0 disk
├─sde1      8:65   0  1022M  0 part
├─┬md0      9:0    0     3G   0 raid5 /raid5
sr0         11:0   1  1024M  0 rom
[root@srv-l ~]#
```

## NFS-сервер - установка

*\$ apt-get install nfs-server rpcbind nfs-clients*

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
[root@srv-l ~]# mkdir /raid5/nfs
[root@srv-l ~]#
```

chmod 777 /raid5/nfs

vim /etc/exports

```
root@srv-l: /root
Файл Правка Вид Поиск Терминал Помощь
# See also /etc/sysconfig/rpcbind (control rpcbind).
/raid5/nfs 10.10.10.110(rw,sync,no_subtree_check) 20.20.20.100(rw,sync,no_subtree_check)
~
~
```

Systemctl enable --now nfs-server

На web-l и web-r прописываем

apt-get install nfs-clients

```
root@web-l: /root
Файл Правка Вид Поиск Терминал Помощь
[root@web-l ~]# mkdir /mnt/nfs
[root@web-l ~]#
```

vim /etc/fstab

```
root@web-l: /root
Файл Правка Вид Поиск Терминал Помощь
proc /proc proc nosuid,noexec,gid=proc
0 0
devpts /dev/pts devpts nosuid,noexec,gid=tty,mode=620
0 0
tmpfs /tmp tmpfs nosuid
0 0
UUID=85af0e0b-2739-4f64-b468-9ae01a0842cc / ext4 relatime
1 1
UUID=2b9a7b33-ec8f-4661-ad6f-92ce6371dc4c swap swap defaults
0 0
/dev/sr0 /media/ALTLinux udf,iso9660 ro,noauto,user,utf8,nofail,comme
nt=x-gvfs-show 0 0
10.10.100:/raid5/nfs /mnt/nfs nfs defaults 0 0
~
```

(последняя строчка)

В конце прописываем mount -a

### ПРОВЕРКА:

Создадим файлы в папке и они должны виднеться .

8. 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 mediawiki.au.team

## ▼ WEB-L

1. `systemctl disable --now ahttpd`  
`systemctl disable --now alteratord`
2. `vim ~/wiki.yml`

а. пишем это:

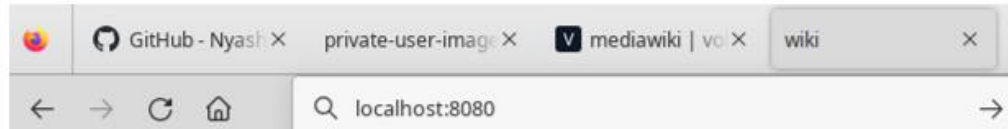
```
version: '3'
services:
  Mediawiki:
    container_name: wiki
    image: mediawiki
    restart: always
    ports:
      - 8080:80
    links:
      - database
    volumes:
      - images:/var/www/html/images
      # - ./LocalSettings.php:/var/www/html/LocalSettings.php

  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
```

3. `systemctl enable --now docker`

4. `docker volume create dbvolume`
5. `cd ~`
6. `docker-compose -f wiki.yml up -d`
7. заходим в mozilla, пишем в строке url:  
`localhost:8080`



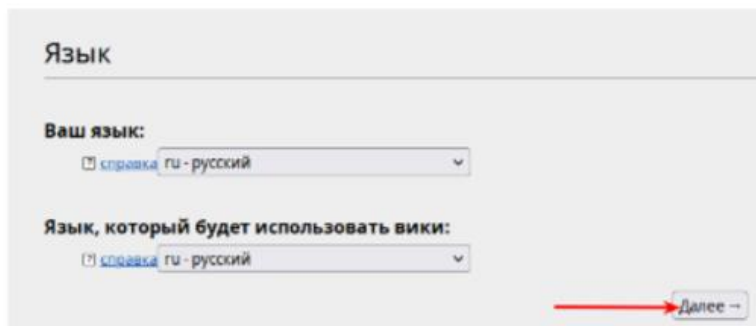
8. жмем set up the wiki



9. далее

9. далее

## Установка MediaWiki 1.41.0



10. далее



11. Пароль: DEP@ssw0rd

Тип базы данных:

☒ MariaDB, MySQL или совместимая

☐ SQLite

## Настройки MariaDB/MySQL

Хост базы данных:

[? справка](#)

db

☐ Подключиться через S

---

Идентификация этой вики

Имя базы данных (без дефисов):

[? справка](#)

mediawiki

Префикс таблиц базы данных (без дефи

[? справка](#)

---

Учётная запись для установки

Имя пользователя базы данных:

[? справка](#)

wiki

Пароль базы данных:

[? справка](#)

.....

12. Далее

### Настройки базы данных

Учётная запись для доступа к базе данных из веб-сервера

☒ Использовать ту же учётную запись, что и для установки

← Назад    **Далее →**

13. List

### Название

Название вики:

[справка](#)

wiki

Пространство имён проекта:

[справка](#)

☒ То же, что имя вики: Wiki

☐ Проект

☐ Другое (укажите)

Учётная запись администратора:

[справка](#)

@wiki

14. Пароль: DEP@ssw0rd почти не указываем

Пароль:

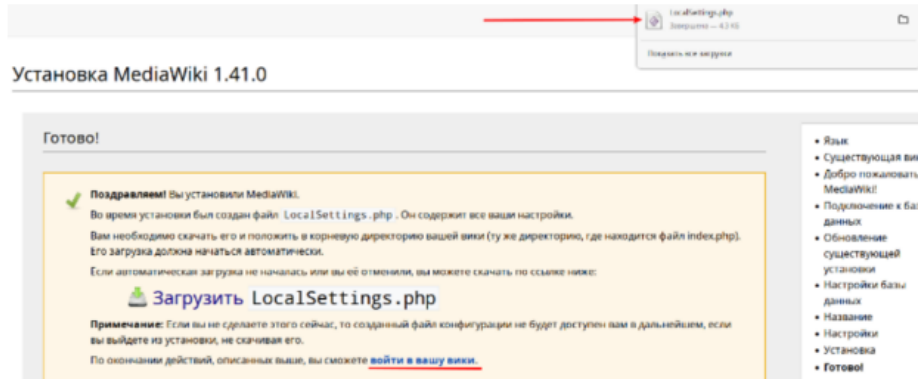
.....

Пароль ещё раз:

.....



16. Жмем до конца далее и скачается файл, надо найти куда этот файл скачался, скорее всего вот сюда `/home/user/Загрузки/`



17. Копируем скачанный файл:  
`cp /home/user/Загрузки/LocalSettings.php ~/LocalSettings.php`

18. `vim ~/wiki.yml`

- a. прокомментируем  
`~/LocalSettings.php:/var/www/html/LocalSettings.php`

19. `vim ~/LocalSettings.php`

- a. `$wgServer = "http://mediawiki.au.team:8080"`

20. `docker-compose -f wiki.yml stop`

21. `docker-compose -f wiki.yml up -d`

▼ WEB-R

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

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

На cli

`apt-get update`

`apt-get install yandex-browser-stable`