

1.

Имена хостов в созданных VM должны быть установлены в соответствии с доменным именем. Адресация должна быть выполнена в соответствии с Таблицей 1.

Имя VM	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

- ISP

1. `hostnamectl set-hostname isp; exec bash`

2. `nano /etc/net/sysctl.conf`

1. `net.ipv4.ip_forward = 1`

3. `cd /etc/net/ifaces`

4. `cp -r enp0s3/ enp0s8`

5. `cp -r enp0s3/ enp0s9`

6. `cp -r enp0s3/ enp0s10`

7. `vim enp0s8/options`

1. `BOOTPROTO=static`

8. `vim enp0s8/ipv4address`

1. `100.100.100.1/28`

9. `vim enp0s8/ipv4route`

1. `10.10.10.0/24 via 100.100.100.10`

10. `vim enp0s9/options`

1. `BOOTPROTO=static`

11. vim enp0s9/ipv4address
 1. 150.150.150.1/28
12. vim enp0s9/ipv4route
 1. 20.20.20.0/24 via 150.150.150.10
13. vim enp0s10/options
 1. BOOTPROTO=static
14. vim enp0s10/ipv4address
 1. 35.35.35.1/28
15. systemctl restart network
16. reboot
17. apt-get update

- CLI

1. hostnamectl set-hostname cli; exec bash
2. Пишем “ip a” и “ls /etc/net/ifaces/” проверяем, что для интерфейса ens19 есть директория /etc/net/ifaces/ens19, если нет, то “cp -r /etc/net/ifaces/ens18 /etc/net/ifaces/ens19”
3. vim /etc/net/ifaces/enp0s3/options
 1. BOOTPROTO=static
4. vim /etc/net/ifaces/enp0s3/ipv4address
 1. 35.35.35.10/28
5. vim /etc/net/ifaces/enp0s3/ipv4route
 1. default via 35.35.35.1 (ip адрес isp)
6. systemctl restart network
7. reboot
8. apt-get update

- RTR-L

1. hostnamectl set-hostname rtr-l; exec bash
2. vim /etc/net/sysctl.conf
 1. net.ipv4.ip_forward = 1

3. cd /etc/net/ifaces
4. vim enp0s3/options
 1. BOOTPROTO=static
5. vim enp0s3/ipv4address
 1. 100.100.100.10/28
6. cp -r enp0s3 enp0s8
7. vim enp0s3/ipv4route
 1. default via 100.100.100.1 (ip адрес isp)
8. vim enp0s8/ipv4address
 1. 10.10.10.1/24
9. systemctl restart network
10. reboot
11. apt-get update

- RTR-R

1. hostnamectl set-hostname rtr-r; exec bash
2. vim /etc/net/sysctl.conf
 1. net.ipv4.ip_forward = 1
3. cd /etc/net/ifaces
4. vim enp0s3/options
 1. BOOTPROTO=static
5. vim enp0s3/ipv4address
 1. 150.150.150.10/28
6. cp -r enp0s3 enp0s8
7. vim enp0s3/ipv4route
 1. default via 150.150.150.1 (ip адрес isp)
8. vim enp0s8/ipv4address
 1. 20.20.20.1/24
9. systemctl restart network
10. reboot

11. apt-get update

- WEB-L

1. hostnamectl set-hostname web-l; exec bash

2. vim /etc/net/iface/enp0s3/options

1. BOOTPROTO=static

3. vim /etc/net/iface/enp0s3/ipv4address

1. 10.10.10.110/24

4. vim /etc/net/iface/enp0s3/ipv4route

1. default via 10.10.10.1

5. systemctl restart network

6. reboot

7. apt-get update

- WEB-R

1. hostnamectl set-hostname web-r; exec bash

2. vim /etc/net/iface/enp0s3/options

1. BOOTPROTO=static

3. vim /etc/net/iface/enp0s3/ipv4address

1. 20.20.20.100/24

4. vim /etc/net/iface/enp0s3/ipv4route

1. default via 20.20.20.1

5. systemctl restart network

6. reboot

7. apt-get update

- SRV-L

1. hostnamectl set-hostname srv-l; exec bash

2. vim /etc/net/iface/enp0s3/options

1. BOOTPROTO=static

3. vim /etc/net/iface/enp0s3/ipv4address

1. 10.10.10.100/24

4. `vim /etc/net/iface/enp0s3/ipv4route`

1. default via 10.10.10.1

5. `systemctl restart network`

6. `reboot`

7. `apt-get update`

2.

Настройка динамической трансляции адресов на всех роутерах.

Сначала делаем на ISP!

- ISP

1. `apt-get update && apt-get install nftables -y`

2. `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 enp0s3 (интерфейс в сторону интернета) masquerade`

`}`

`}`

3. Вот так:

```
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;
    }
}
```

3. `systemctl enable --now nftables`

4. `nft -f /etc/nftables/nftables.nft`

- RTR-L

1. Если “ping 8.8.8.8” идет, а “ping ya.ru” не идет, то в

`/etc/resolv.conf` записываем в начало `nameserver 94.232.137.104` (ip адрес DNS сервера указанный в 5 пункте задания)

2. `apt-get update && apt-get install nftables -y`

3. `vim /etc/nftables/nftables.nft`

1. в начало:

```
flush ruleset
```

2. в конец:

```
table ip nat {
```

```
    chain postrouting {
```

```
        type nat hook postrouting priority 0;
```

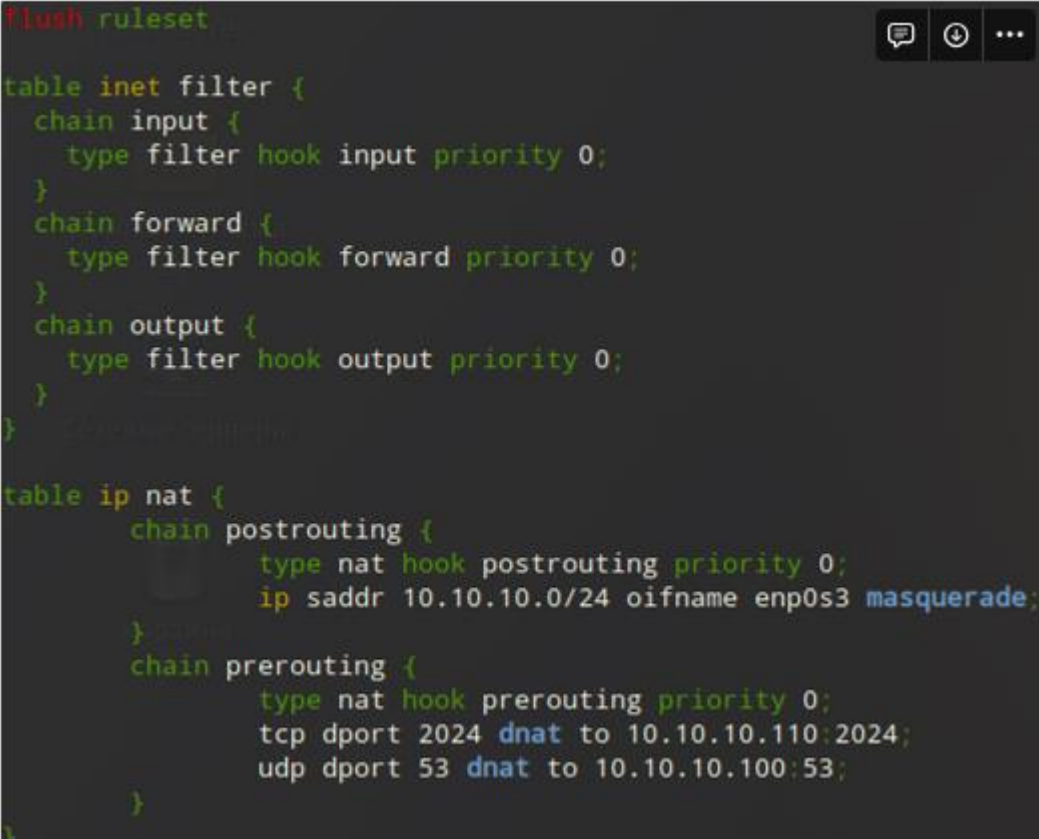
```
        ip saddr 10.10.10.0/24 oifname enp0s3 (интерфейс в сторону isp)
        masquerade;
```

```

}
chain prerouting {
    type nat hook prerouting priority 0;
    tcp dport 2024 dnat to 10.10.10.110:2024;
    udp dport 53 dnat to 10.10.10.100:53;
}
}

```

3. Вот так:



```

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;
        ip saddr 10.10.10.0/24 oifname enp0s3 masquerade;
    }
    chain prerouting {
        type nat hook prerouting priority 0;
        tcp dport 2024 dnat to 10.10.10.110:2024;
        udp dport 53 dnat to 10.10.10.100:53;
    }
}

```

4. `systemctl enable --now nftables`

5. `nft -f /etc/nftables/nftables.nft`

- RTR-R

1. Если “ping 8.8.8.8” идет, а “ping ya.ru” не идет, то в `/etc/resolv.conf` записываем в начало `nameserver 94.232.137.104` (ip адрес DNS сервера указанный в 5 пункте задания)
2. `apt-get update && apt-get install nftables -y`
3. `vim /etc/nftables/nftables.nft`

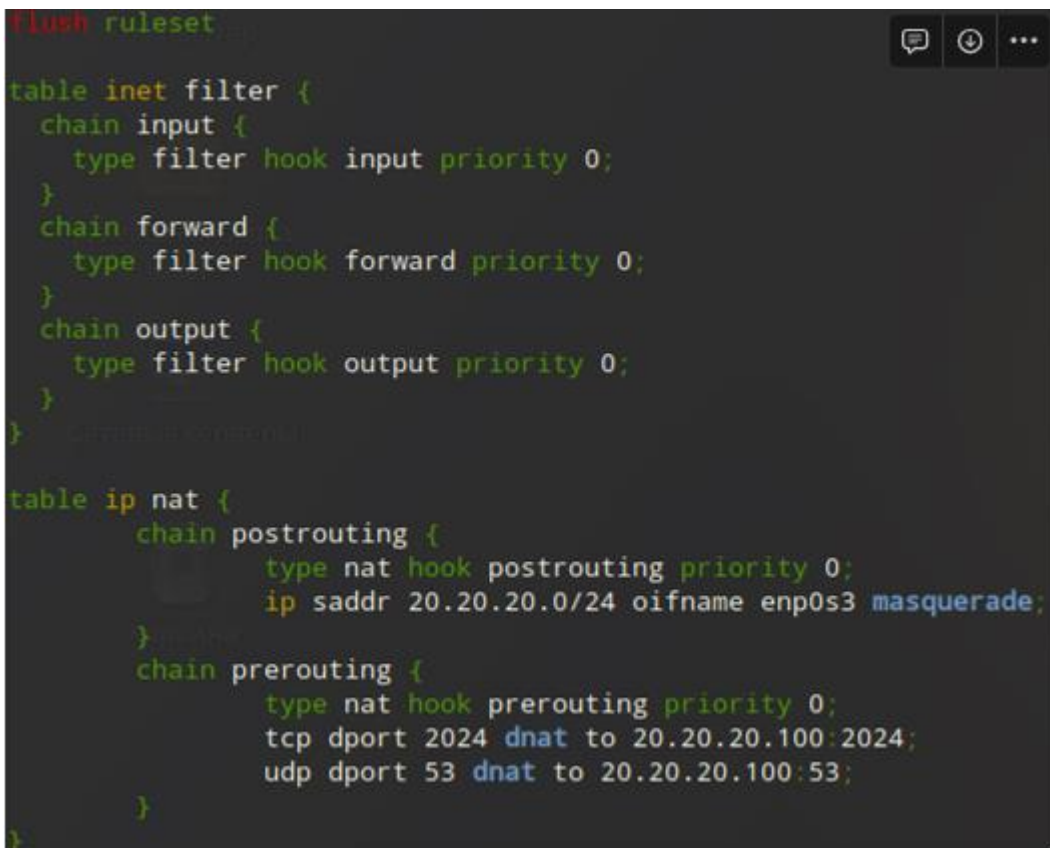
1. в начало:

```
flush ruleset
```

2. в конец:

```
table ip nat {  
    chain postrouting {  
        type nat hook postrouting priority 0;  
        ip saddr 20.20.20.0/24 oifname enp0s3 (интерфейс в сторону isp)  
masquerade;  
    }  
    chain prerouting {  
        type nat hook prerouting priority 0;  
        tcp dport 2024 dnat to 20.20.20.100:2024;  
        udp dport 53 dnat to 20.20.20.100:53;  
    }  
}
```

3. Вот так:



```
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;  
        ip saddr 20.20.20.0/24 oifname enp0s3 masquerade;  
    }  
    chain prerouting {  
        type nat hook prerouting priority 0;  
        tcp dport 2024 dnat to 20.20.20.100:2024;  
        udp dport 53 dnat to 20.20.20.100:53;  
    }  
}
```


4. `systemctl enable --now nftables`

5. `nft -f /etc/nftables/nftables.nft`

3.

Защищенный(IPSEC) туннель

- 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`

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

3. `/etc/gre.up`

4. `vim /etc/crontab`

1. в конец добавляем:

`@reboot root /etc/gre.up`

5. `apt-get install strongswan`

6. `vim /etc/strongswan/ipsec.conf`

1. ниже “`config setup`” пишем:

`conn vpn`

(следующие строки через tab)

`auto=start`

`type=tunnel`

`authby=secret`

`left=100.100.100.10`

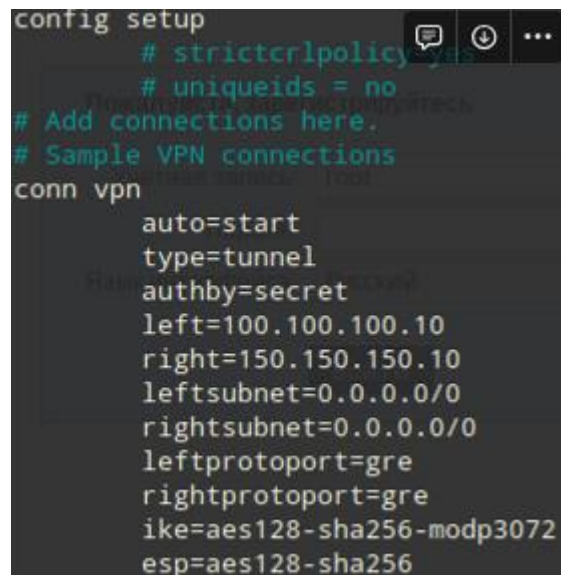
`right=150.150.150.10`

`leftsubnet=0.0.0.0/0`

`rightsubnet=0.0.0.0/0`

```
leftprotoport=gre
rightprotoport=gre
ike=aes128-sha256-modp3072
esp=aes128-sha256
```

2. ВОТ ТАК



```
config setup
# strictcrpolicy yes
# uniqueids = no
# Add connections here.
# 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
```

7. vim /etc/strongswan/ipsec.secrets

1. 100.100.100.10 150.150.150.10 : PSK "P@ssw0rd"

8. 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

2. chmod +x /etc/gre.up

3. /etc/gre.up

4. vim /etc/crontab

1. в конец добавляем:

@reboot root /etc/gre.up

5. apt-get install strongswan

6. vim /etc/strongswan/ipsec.conf

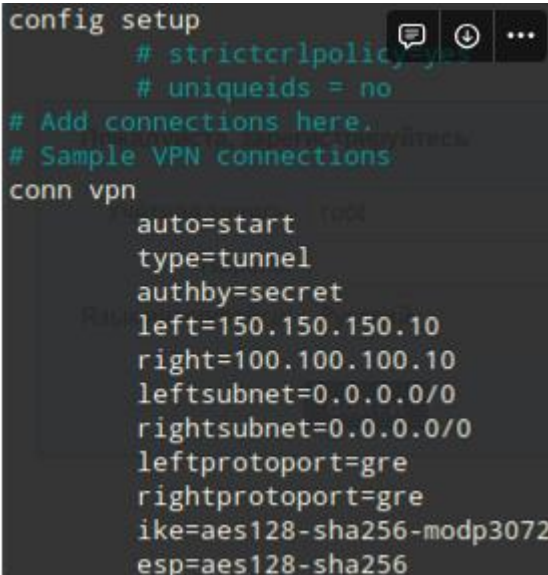
1. ниже “config setup” пишем:

conn vpn

(следующие строки через tab)

```
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
```

ВОТ ТАК



```
config setup
# strictcrpolicies=yes
# uniqueids = no
# Add connections here.
# 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
```

7. vim /etc/strongswan/ipsec.secrets

1. 100.100.100.10 150.150.150.10 : PSK “P@ssw0rd”

8. 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`.

4.

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. раскомментируем строчку Port 22
пишем вместо 22 тот порт, который указан в задании (2024)
 2. раскомментируем строчку MaxAuthTries 6
пишем вместо 6 столько попыток, сколько указано в задании (2)
 3. раскомментируем строчку Banner none
вместо none пишем путь к banner.txt (/etc/openssh/banner.txt)
 4. добавляем в конец AllowUsers sshuser
sshuser - пользователь указан в задании
3. `adduser sshuser`
4. `passwd sshuser`
 1. P@ssw0rd два раза
5. `systemctl restart sshd`

- WEB-R

1. `vim /etc/openssh/banner.txt`
 1. Authorized access only
2. `vim /etc/openssh/sshd_config`
 1. раскомментируем строчку Port 22
пишем вместо 22 тот порт, который указан в задании (2024)

2. раскомментируем строчку MaxAuthTries 6

пишем вместо 6 столько попыток, сколько указано в задании (2)

3. раскомментируем строчку Banner none

вместо none пишем путь к banner.txt (/etc/openssh/banner.txt)

4. добавляем в конец AllowUsers sshuser

sshuser - пользователь указан в задании

3. adduser sshuser

4. passwd sshuser

1. P@ssw0rd два раза

5. systemctl restart sshd

5.

Настройка DNS для офисов.

- SRV-L

1. apt-get update && apt-get install bind bind-utils -y

2. systemctl enable --now bind

3. vim /etc/bind/options.conf

1. что должно быть в options:

listen-on { any; };

forwarders { 94.232.137.104; }; (ip адрес DNS в задании)

dnssec-validation no;

recursion yes;

allow-query { any; };

allow-recursion { any; };

2. вот так

```
options {  
    version "unknown";  
    directory "/etc/bind/zone";  
    dump-file "/var/run/named_dump.db";  
    statistics-file "/var/run/named.stats";  
    recursing-file "/var/run/recursing";  
    // disables the use of a PID file  
    pid-file none;  
    listen-on { any; };  
    forwarders { 94.232.137.104; };  
    dnssec-validation no;  
    recursion yes;  
    allow-query { any; };  
    allow-recursion { any; };  
}
```

4. 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; };  
};
```

};

2. ВОТ ТАК

```
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;};  
};
```

5. cd /etc/bind/zone/

6. cp localhost au.team

7. vim au.team

1. заменяем localhost. на au.team. и root.localhost. на root.au.team.

2. редачим зоны, должны получиться такие зоны, пишем через табуляцию (Tab), а не пробелы:

```
@      IN      NS      au.team.  
@      IN      NS      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       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
```

должно получиться так

```
$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-r     IN      A        10.10.10.1
srv-l     IN      A        10.10.10.100
web-l     IN      A        10.10.10.110
rtr-r     IN      A        20.20.20.1
web-r     IN      A        20.20.20.100
cli       IN      A        35.35.35.10
dns       IN      CNAME    srv-l
ntp       IN      CNAME    isp
mediawiki IN      CNAME    web-l
```

8. cp localhost right.reverse

9. 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. редачим зоны, должны получиться такие зоны пишем через табуляцию (Tab), а не пробелы:

```
YAML                                                                    Copy
@         IN      NS       au.team.
@         IN      NS       10.10.10.100
1         PTR     rtr-r.au.team.
100       PTR     web-r.au.team.
```

должно получиться так

```
$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        10.10.10.100
1         PTR     rtr-r.au.team.
100       PTR     web-r.au.team.
```


10. cp right.reverse left.reverse

11. 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. редачим зоны, должны получиться такие зоны пишем через табуляцию (Tab), а не пробелы:

```
YAML                                                                    Copy
@      IN      NS      au.team.
@      IN      NS      10.10.10.100
1      PTR     rtr-l.au.team.
100    PTR     srv-l.au.team.
110    PTR     web-l.au.team.
```

должно получиться так

```
$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.
```

12. cp right.reverse cli.reverse

13. 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. редачим зоны, должны получиться такие зоны пишем через табуляцию (Tab), а не пробелы:

```

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

```

должно получиться вот так

```

$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       10.10.10.100
1          PTR     isp.au.team.
10         PTR     cli.au.team.

```

14. `chmod 777 au.team`

15. `chmod 777 right.reverse`

16. `chmod 777 left.reverse`

17. `chmod 777 cli.reverse`

18. `systemctl restart bind`

19. `echo "nameserver 127.0.0.1" > /etc/net/iface/enp0s8/resolv.conf`

20. `vim /etc/resolv.conf`

должен быть указан только один `nameserver 127.0.0.1`

```

# Generated by resolvconf
# Do not edit manually, use
# /etc/net/iface/<interface>/resolv.conf instead.
#nameserver 8.8.8.8
#nameserver 192.168.30.73
nameserver 127.0.0.1

```

если пропал интернет, добавляем `nameserver 94.232.137.104` или `nameserver 8.8.8.8`, может помочь

- WEB-R

1. `apt-get update && apt-get install bind bind-utils -y`

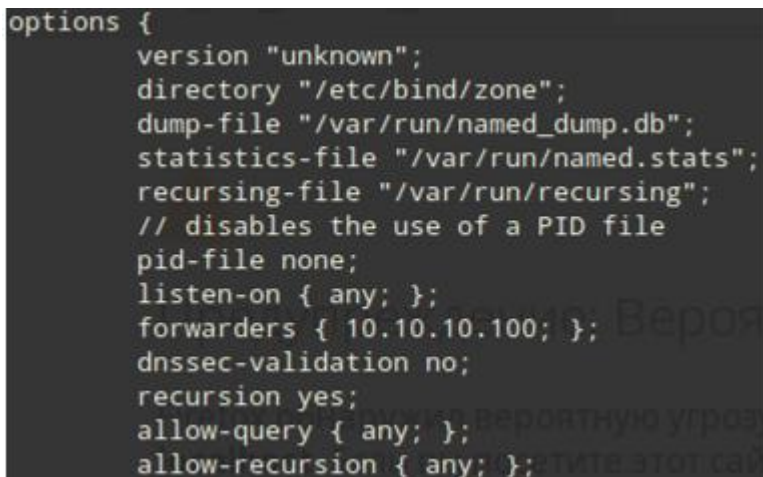
2. `systemctl enable --now bind`

3. 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; };
```

2. вот так



```
options {  
    version "unknown";  
    directory "/etc/bind/zone";  
    dump-file "/var/run/named_dump.db";  
    statistics-file "/var/run/named.stats";  
    recursing-file "/var/run/recursing";  
    // 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; };
```

4. 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;};
};

```

2. ВОТ ТАК:

```

zone "au.team" {
    type slave;
    file "au.team";
    master {10.10.10.100;};
};

zone "10.10.10.in-addr.arpa" {
    type slave;
    file "left.reverse";
    master {10.10.10.100;};
};

zone "20.20.20.in-addr.arpa" {
    type slave;
    file "right.reverse";
    master {10.10.10.100;};
};

zone "35.35.35.in-addr.arpa" {
    type slave;
    file "cli.reverse";
    master {10.10.10.100;};
};

```

5. chown named:named /var/lib/bind/zone/slave/
6. chown named:named /etc/bind/zone/slave/
7. systemctl restart bind
8. echo "nameserver 127.0.0.1" > /etc/net/ifaces/enp0s3/resolv.conf
9. vim /etc/resolv.conf
1. должен быть указан только один nameserver 127.0.0.1

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>/resolv.conf instead.
#nameserver 8.8.8.8
#nameserver 192.168.30.73
nameserver 127.0.0.1
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

После того как DNS сервера настроены, надо указать на каждой машине, кроме самих DNS серверов, на них мы уже себя указали, указать в качестве DNS сервера наши сервера.

- CLI

1. echo "nameserver 100.100.100.10" > /etc/net/ifaces/enp0s3/resolv.conf
2. vim /etc/resolv.conf

1. должен быть указан только один nameserver 100.100.100.10

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>
#search au.team
nameserver 100.100.100.10
#nameserver 172.16.3.20
#nameserver 172.16.3.21
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

- ISP

1. echo "nameserver 100.100.100.10" > /etc/net/ifaces/enp0s8/resolv.conf
2. vim /etc/resolv.conf

1. должен быть указан только один nameserver 100.100.100.10

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>
#search au.team
nameserver 100.100.100.10
#nameserver 172.16.3.20
#nameserver 172.16.3.21
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

- RTR-L

1. echo "nameserver 10.10.10.100" > /etc/net/ifaces/enp0s8/resolv.conf

2. vim /etc/resolv.conf

1. должен быть указан только один nameserver 10.10.10.100

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>
#search au.team
nameserver 10.10.10.100
#nameserver 172.16.3.20
#nameserver 172.16.3.21
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

- RTR-R

1. echo "nameserver 20.20.20.100" > /etc/net/ifaces/enp0s8/resolv.conf
2. vim /etc/resolv.conf

1. должен быть указан только один nameserver 20.20.20.10

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>
#search au.team
nameserver 20.20.20.100
#nameserver 172.16.3.20
#nameserver 172.16.3.21
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

- WEB-L

1. echo "nameserver 10.10.10.100" > /etc/net/ifaces/enp0s3/resolv.conf
2. vim /etc/resolv.conf

1. должен быть указан только один nameserver 10.10.10.100

```
# Generated by resolvconf
# Do not edit manually, use
# /etc/net/ifaces/<interface>
#search au.team
nameserver 10.10.10.100
#nameserver 172.16.3.20
#nameserver 172.16.3.21
```

если пропал интернет, добавляем nameserver 94.232.137.104 или nameserver 8.8.8.8, может помочь

6. CHRONY

Настройте службу сетевого времени на базе сервиса chrony

- ISP

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. 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
3. systemctl restart chronyd

- CLI

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 35.35.35.1 iburst
3. systemctl restart chronyd

- RTR-L

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 100.100.100.1 iburst
3. systemctl restart chronyd

- RTR-R

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 150.150.150.1 iburst
3. systemctl restart chronyd

- WEB-R

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 150.150.150.1 iburst
3. systemctl restart chronyd

- WEB-L

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 100.100.100.1 iburst
3. systemctl restart chronyd

- SRV-L

1. apt-get install chrony
2. vim /etc/chrony.conf
 1. комментируем (пишем #) перед “pool pool.ntp.org iburst”
 2. в конец пишем:
server 100.100.100.1 iburst
3. systemctl restart chronyd

7.

Сконфигурируйте файловое хранилище

- SRV-L

1. apt-get install nfs-server
2. lsblk проверяем NAME 4 дисков размером 1G:
в моем случае 4 диска размером 1 гб это sdb sdc sdd sde

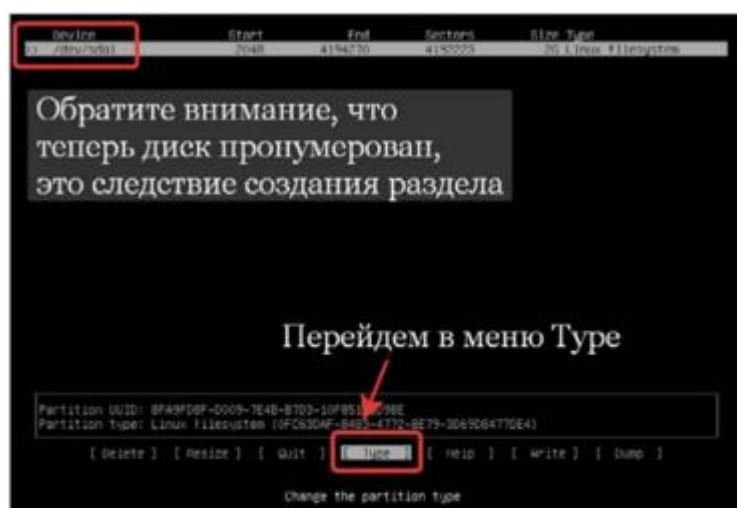
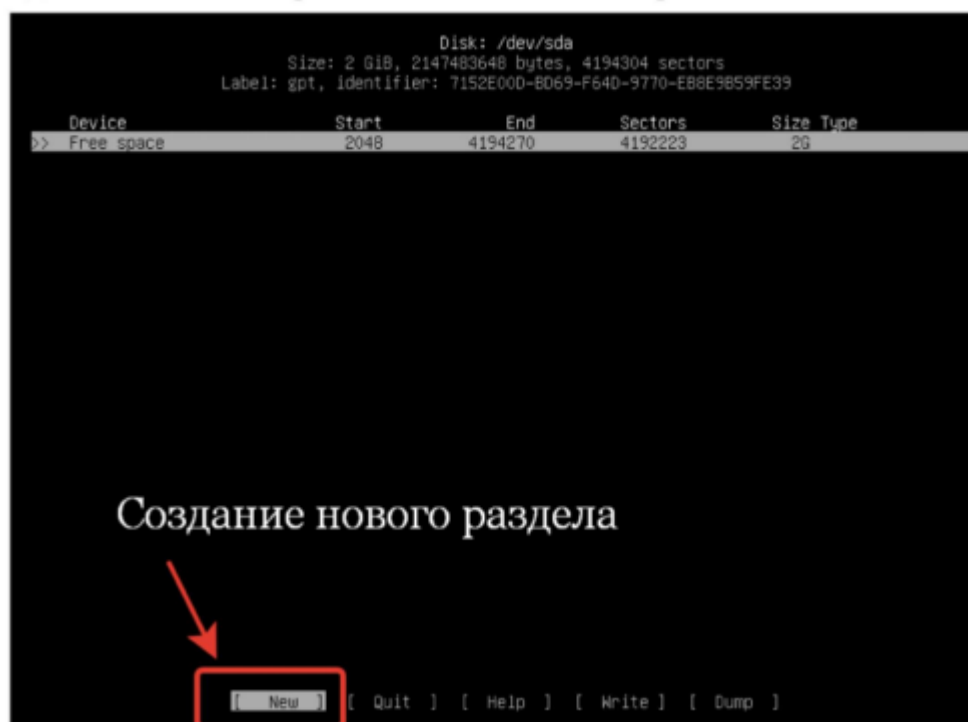
```
[root@srv-1 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	40G	0	disk	
├─sda1	8:1	0	1,9G	0	part	[SWAP]
└─sda2	8:2	0	38,1G	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	

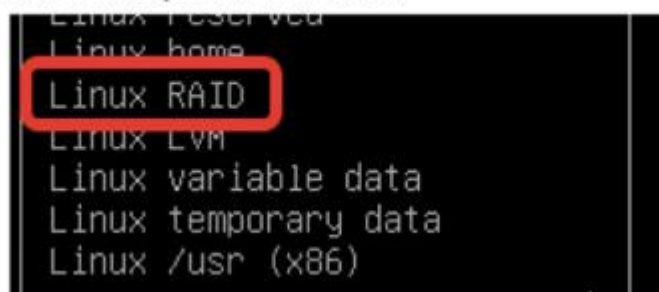
cfdisk /dev/sdb

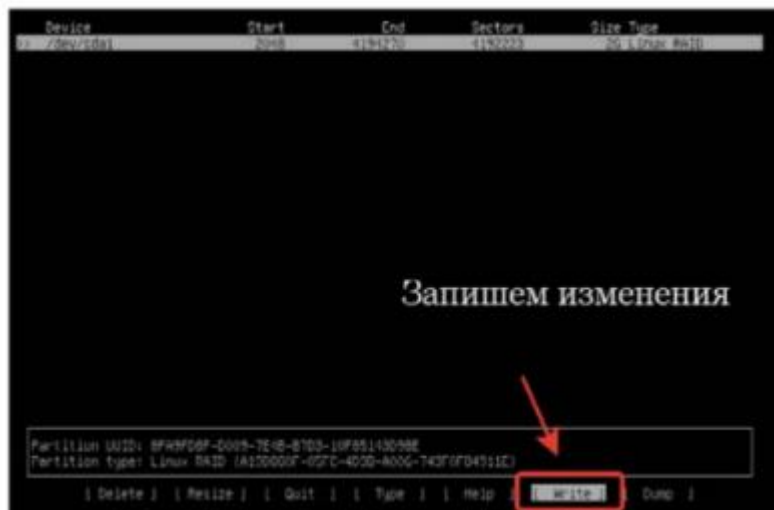
```
Select label type
gpt
dos
sgi
sun
```

Далее необходимо выбрать New для создания нового раздела.

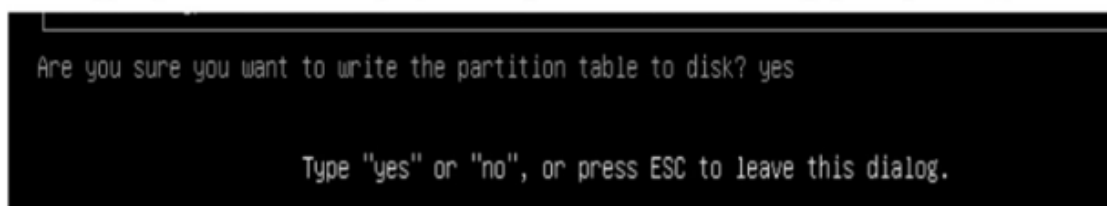


В указанном списке выбирается Linux RAID.



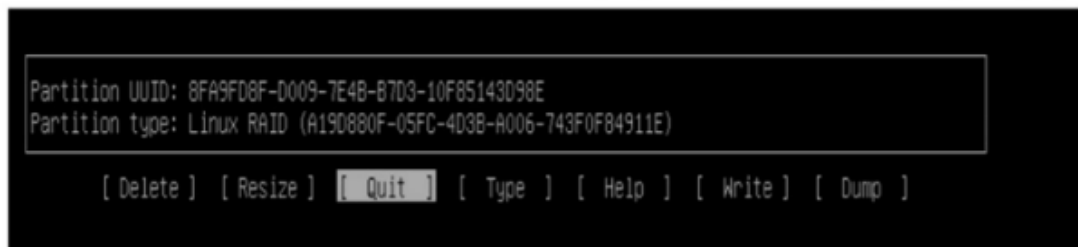


Вручную пишется слово yes, чтобы принять изменения и отформатировать диск.



8. List

После внесенных правок необходимо покинуть утилиту cfdisk, выбрав параметр Quit.



9. С пункта 3 (cfdisk) повторить действия со всеми остальными дисками (sdb, sdc, sdd, sde)

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

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

12. `mkfs.ext4 /dev/md0`

13. `mkdir /raid5`

14. `vim /etc/fstab`

1. добавить в конец, пишем через табуляцию, а не пробел:

`/dev/md0 /raid5 ext4 defaults 0 0`

2. так

```

proc          /proc          proc          nosuid,noexec,gid=proc
devpts        /dev/pts      devpts        nosuid,noexec,gid=tty,mode=620
tmpfs         /tmp         tmpfs         nosuid
UUID=9e08571f-9b6f-4f61-8425-782ad628950f /          ext4          reltime
UUID=c04e0177-5eb9-4410-8685-3396f4271cb4 swap        swap          defaults
/dev/md0      /raid5 ext4      defaults      0          0

```

15. reboot

16. mkdir /raid5/nfs

17. chmod 777 /raid5/nfs

18. vim /etc/exports

1. в конец добавляем:

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

2. Пояснение:

10.10.10.110-адрес WEB-L

20.20.20.100-адрес WEB-R

- WEB-L

1. apt-get install nfs-client

2. mkdir /mnt/nfs

3. vim /etc/fstab

1. добавляем в конец, пишем через табуляцию, а не пробел:

10.10.10.100:/raid5/nfs /mnt/nfs nfs rw,sync 0 0

- WEB-R

1. apt-get install nfs-client

2. mkdir /mnt/nfs

3. vim /etc/fstab

1. добавляем в конец, пишем через табуляцию, а не пробел:

10.10.10.100:/raid5/nfs /mnt/nfs nfs rw,sync 0 0

8.

Запустите сервис MediaWiki используя docker на сервере WEB-L

- WEB-L

1. apt-get install docker-io docker-compose -y

2. vim ~/wiki.yml

1. пишем это:

```
Plain Text Copy

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. docker volume create dbvolume

4. cd ~

5. docker-compose -f wiki.yml up -d

6. заходим в mozilla, пишем в строке url:

localhost:8080

7. жмем set up the wiki



8. далее

Установка MediaWiki 1.41.0

A screenshot of the MediaWiki 1.41.0 installation language selection screen. The title is 'Язык' (Language). It asks for 'Ваш язык:' (Your language:) and 'Язык, который будет использовать вики:' (Language that the wiki will use:). Both dropdown menus are set to 'ru - русский'. A red arrow points to the 'Далее' (Next) button.

9. далее

9. далее



10. Пароль: DFP@ssw0rd

10. 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 дефисов):

[? справка](#)

Учётная запись для установки

Имя пользователя базы данных:

[? справка](#)

wiki

Пароль базы данных:

[? справка](#)

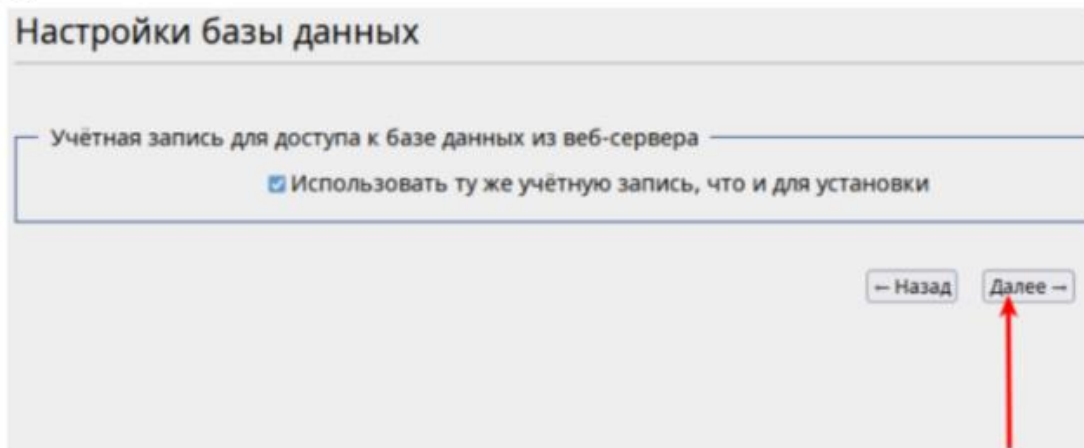
11. Далее

Настройки базы данных

Учётная запись для доступа к базе данных из веб-сервера

☒ Использовать ту же учётную запись, что и для установки

[← Назад](#) [Далее →](#)



12. List

Название

Название вики:

[? справка](#)

Пространство имён проекта:

[? справка](#)

☒ То же, что имя вики: Wiki

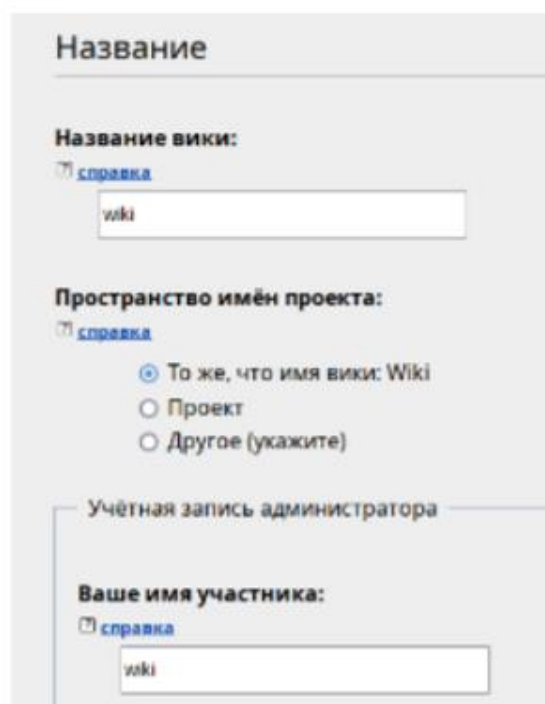
☐ Проект

☐ Другое (укажите)

Учётная запись администратора

Ваше имя участника:

[? справка](#)



13. Пароль: DEP@ssw0rd почту не указываем

Пароль:

Пароль ещё раз:

Адрес электронной почты:

[справка](#)

[справка](#)

☐ Подписаться на рассылку новостей о появлении новых версий MediaWiki.

[справка](#)

☒

Поделиться сведениями об этой установке с разработчикам MediaWiki.

[Политика конфиденциальности.](#)

Вы почти у цели! Остальные настройки можно пропустить и приступить к установке вики.

☐ Произвести тонкую настройку

☒ Хватит, установить вики

[← Назад](#) [Далее →](#)

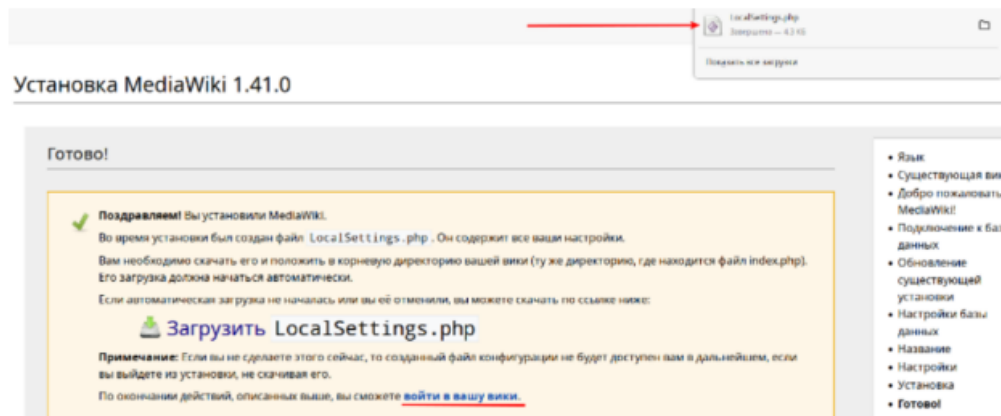
14. Далее

Установка

- Настройка базы данных... выполнено
- Создание таблиц, первый шаг... выполнено
- Создание базы данных пользователей... выполнено
- Создание таблиц, второй шаг... выполнено
- Заполнение таблицы интервики значениями по умолчанию... выполнено
- Статистика инициализации... выполнено
- Создание секретных ключей... выполнено
- Предотвращение запуска ненужных обновлений... выполнено
- Восстановление сервисов MediaWiki... выполнено
- Создание учётной записи администратора... выполнено
- Создание главной страницы с содержимым по умолчанию... выполнено
- База данных была успешно настроена

[Далее →](#)

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



16. Копируем скачанный файл:

```
cp /home/user/Загрузки/LocalSettings.php ~/LocalSettings.php
```

17. `vim ~/wiki.yml`

1. раскомментируем

```
- ./LocalSettings.php:/var/www/html/LocalSettings.php
```

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

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

9.

установите приложение Яндекс Браузер на CLI

- CLI

1. `apt-get install yandex-browser -y`

2. запустить HE от рута с помощью команды:

```
yandex-browser-stable
```

запустить от рута с помощью команды:

```
yandex-browser-stable --no-sandbox
```