

Подготовка к экзамену по администрированию сетей Усков Д. Ю. КИ-21-01

▼ ПУНКТ 1 НАСТРОЙКА IP АДРЕСОВ

▼ ISP

1. `hostnamectl set-hostname ISP; exec bash`
2. `vim /etc/net/sysctl.conf`
`net.ipv4.ip_forward = 1`
1. `cd /etc/net/ifaces`
2. `cp -r ens18/ ens19`
3. `vim ens19/options`
`BOOTPROTO=static`
4. `cp -r ens19/ ens20`
5. `cp -r ens19/ ens21`
6. `vim ens19/ipv4address`
 - a. `100.100.100.1/28`
7. `vim ens19/ipv4route`
 - a. `10.10.10.0/24 via 100.100.100.10`
8. `vim ens20/ipv4address`
 - a. `150.150.150.1/28`
9. `vim ens20/ipv4route`
 - a. `20.20.20.0/24 via 150.150.150.10`
10. `vim ens21/ipv4address`
 - a. `35.35.35.1/28`
11. `systemctl restart network`

12. reboot

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

14. vim /etc/nftables/nftables.nft
flush ruleset;

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

1. systemctl enable --now nftables.service

2. nft -f /etc/nftables/nftables.nft

3. nft list ruleset

▼ CLI

1. hostnamectl set-hostname CLI; exec bash

2. cd /etc/net/ifaces

3. cp -r ens18/ ens19

4. vim ens19/options

a. BOOTPROTO=static

5. vim ens19/ipv4address

a. 35.35.35.10/28

6. vim ens19/ipv4route

a. default via 35.35.35.1

7. systemctl restart network

8. reboot

9. apt-get update && apt-get install chrony yandex-
browser-stable -y

▼ RTR-L

1. hostnamectl set-hostname RTR-L; exec bash

2. `vim /etc/net/sysctl.conf`
 - a. `net.ipv4.ip_forward = 1`
3. `cd /etc/net/ifaces`
4. `vim ens18/options`
 - a. `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`
8. `vim ens19/ipv4address`
 - a. `10.10.10.1/24`
9. `systemctl restart network`
10. `reboot`
11. `apt-get update && apt-get install nftables chrony strongswan -y`

▼ RTR-R

1. `hostnamectl set-hostname RTR-R; exec bash`
2. `vim /etc/net/sysctl.conf`
 - a. `net.ipv4.ip_forward = 1`
3. `cd /etc/net/ifaces`
4. `vim ens18/options`
 - a. `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`

8. `vim ens19/ipv4address`
 - a. `20.20.20.1/24`
9. `systemctl restart network`
10. `reboot`
11. `apt-get update && apt-get install chrony nftables strongswan -y`

▼ WEB-L

1. `hostnamectl set-hostname WEB-L; exec bash`
2. `cd /etc/net/ifaces/ens18/`
3. `vim options`
 - a. `BOOTPROTO=static`
4. `vim ipv4address`
 - a. `10.10.10.110/24`
5. `vim ipv4route`
 - a. `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; exec bash`
2. `cd /etc/net/ifaces/ens18/`
3. `vim options`
 - a. `BOOTPROTO=static`
4. `vim ipv4address`
 - a. `20.20.20.100/24`
5. `vim ipv4route`
 - a. `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; exec bash`
2. `cd /etc/net/iface/ens18/`
3. `vim options`
 - a. `BOOTPROTO=static`
4. `vim ipv4address`
 - a. `10.10.10.100/24`
5. `vim ipv4route`
 - a. `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`

▼ ПУНКТ 2 ДИНАМИЧЕСКАЯ ТРАНСЛЯЦИЯ НА RTR-L; RTR-R

▼ RTR-L

1. `vim /etc/nftables/nftables.nft`

```
flush ruleset;

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;
```

```
}  
}
```

2. `systemctl enable --now nftables`

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

4. `nft list ruleset`

▼ RTR-R

1. `vim /etc/nftables/nftables.nft`

```
flush ruleset;
```

```
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;  
    }  
}
```

2. `systemctl enable --now nftables`

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

4. `nft list ruleset`

▼ ПУНКТ 3. ТОННЕЛЬ

▼ RTR-L

1. `vim /etc/gre.up`

```
a. #!/bin/bash
```

```
ip tunnel add tun0 mode gre local 100.100.100.10  
remote 150.150.150.10
```

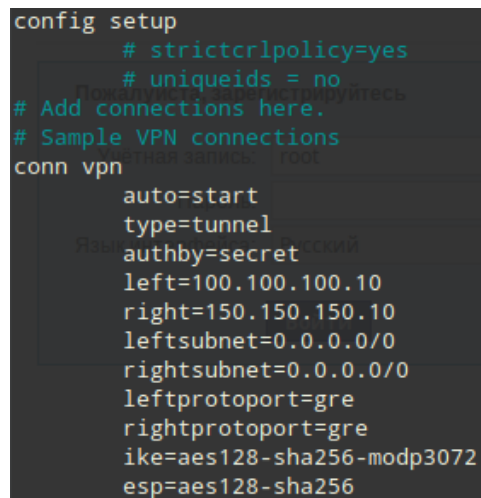
```
ip addr add 10.5.5.1/30 dev tun0
```

```
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`
 - а. в конец добавляем:
`@reboot root /etc/gre.up`
5. `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
```



```
config setup
# strictcrpolicypolicy=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
```

6. `vim /etc/strongswan/ipsec.secrets`
 - а. `100.100.100.10 150.150.150.10 : PSK "P@ssw0rd"`
7. `systemctl enable --now ipsec.service`

▼ RTR-R

1. `vim /etc/gre.up`

```
a. #!/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`

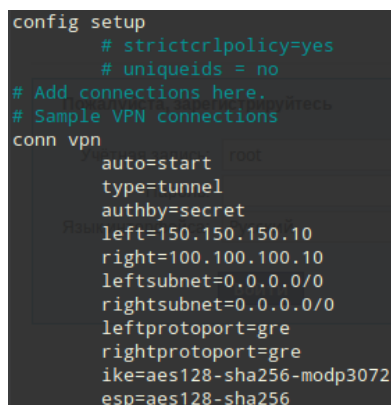
4. `vim /etc/crontab`

```
a. в конец добавляем:
@reboot root /etc/gre.up
```

5. `apt-get install strongswan`

6. `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
```



```
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
```

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

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

2. `systemctl enable --now ipsec.service`

`ipsec status`, если есть 1 up. `ipsec update` и `ipsec restart`.

▼ ПУНКТ 4. НАСТРОЙКА SSH НА WEB-L; WEB-R

▼ WEB-L

1. `vim /etc/openssh/имя_баннера`

a. `Authorized access only`

2. `vim /etc/openssh/sshd_config`

a. раскомментируем строчку `Port 22`
`Port 2024`

b. раскомментируем строчку `MaxAuthTries 6`
`MaxAuthTries 2`

c. раскомментируем строчку `Banner none`
вместо `none` пишем
`Banner /etc/openssh/имя_файла`

d. добавляем в конец
`AllowUsers sshuser`

3. `adduser sshuser`

4. `passwd sshuser`

a. `P@ssw0rd`

5. `systemctl restart sshd`

▼ WEB-R

1. `vim /etc/openssh/имя_файла`

a. `Authorized access only`

2. `vim /etc/openssh/sshd_config`

a. раскомментируем строчку `Port 22`
`Port 2024`

b. раскомментируем строчку `MaxAuthTries 6`
`MaxAuthTries 2`

- c. раскомментируем строчку Banner none
вместо none пишем
Banner /etc/openssh/имя_файла
 - d. добавляем в конец
AllowUsers sshuser
- 3. adduser sshuser
- 4. passwd sshuser
 - a. P@ssw0rd
- 5. systemctl restart sshd

- 1. проверка
ssh -p 2024 sshuser@10.10.10.110

▼ ПУНКТ 5. DNS

▼ SRV-L

- 1. systemctl enable --now bind
- 2. vim /etc/bind/options.conf
 - a. listen-on { any; };
forwarders { 94.232.137.104; };
dnssec-validation no;
recursion yes;
allow-query { any; };
allow-recursion { any; };
- 3. vim /etc/bind/local.conf
 - a. 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;};
};
```

4. `cd /etc/bind/zone/`

5. `cp localhost au.team`

6. `vim au.team`

a. заменяем `localhost.` на `au.team.` и `root.localhost.` на `root.au.team.`

b. пишем через табуляцию зоны:

```
$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
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
```

7. `cp localhost right.reverse`

8. `vim right.reverse`

a. заменяем `localhost.` на `20.20.20.in-addr.arpa.` и `root.localhost.` на `root.20.20.20.in-addr.arpa.`

b. пишем через табуляцию зоны:

```
$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.
@ server IN A 20.20.20.100
1 PTR rtr-r.au.team.
100 PTR web-r.au.team.
```

9. cp right.reverse left.reverse

10. vim left.reverse

a. 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.

b. пишем через табуляцию зоны:

```
$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.
@ server IN A 10.10.10.100
1 PTR rtr-l.au.team.
100 PTR srv-l.au.team.
110 PTR web-l.au.team.
```

11. cp right.reverse cli.reverse

12. vim cli.reverse

a. 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.

b. пишем через табуляцию зоны:

```

$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.
@ 35.35.35 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`

17. `systemctl restart bind`

18. `vim /etc/resolv.conf`

- а. должен быть указан только один nameserver
127.0.0.1

▼ WEB-R

1. `systemctl enable --now bind`

2. `vim /etc/bind/options.conf`

- а. что должно быть в options:


```
listen-on { any; };
forwarders { 10.10.10.100; };
dnssec-validation no;
recursion yes;
allow-query { any; };
allow-recursion { any; };
```

3. `vim /etc/bind/local.conf`

- а. 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;};
    };
};

```

4. `chown named:named /var/lib/bind/zone/slave/`

5. `chown named:named /etc/bind/zone/slave/`

6. `systemctl restart bind`

7. `vim /etc/resolv.conf`

а. должен быть указан только один nameserver
127.0.0.1

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

▼ CLI

1. `vim /etc/resolv.conf`

а. должен быть указан только один nameserver
100.100.100.10

▼ ISP

1. `vim /etc/resolv.conf`

а. должен быть указан только один nameserver
100.100.100.10

▼ RTR-L

1. `vim /etc/resolv.conf`

- а. должен быть указан только один nameserver
10.10.10.100

▼ RTR-R

1. vim /etc/resolv.conf
 - а. 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
2. vim /etc/resolv.conf
 - а. должен быть указан только один nameserver
10.10.10.100
 - б. если пропал интернет, добавляем nameserver
94.232.137.104 или nameserver 8.8.8.8

▼ ПУНКТ 6. CHRONY

▼ ISP

1. vim /etc/chrony.conf
 - а. 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
2. systemctl restart chronyd

▼ CLI

1. vim /etc/chrony.conf
 - а. комментируем "pool pool.ntp.org iburst"
 - б. в конец пишем:
server 35.35.35.1 iburst

2. `systemctl restart chronyd`

▼ RTR-L

1. `vim /etc/chrony.conf`

a. комментируем "pool pool.ntp.org iburst"

b. в конец пишем:

`server 100.100.100.1 iburst`

2. `systemctl restart chronyd`

▼ RTR-R

1. `vim /etc/chrony.conf`

a. комментируем "pool pool.ntp.org iburst"

b. в конец пишем:

`server 150.150.150.1 iburst`

2. `systemctl restart chronyd`

▼ WEB-R

1. `vim /etc/chrony.conf`

a. комментируем "pool pool.ntp.org iburst"

b. в конец пишем:

`server 150.150.150.1 iburst`

2. `systemctl restart chronyd`

▼ WEB-L

1. `vim /etc/chrony.conf`

a. комментируем "pool pool.ntp.org iburst"

b. в конец пишем:

`server 100.100.100.1 iburst`

2. `systemctl restart chronyd`

▼ SRV-L

1. `vim /etc/chrony.conf`

a. комментируем "pool pool.ntp.org iburst"

b. в конец пишем:

```
server 100.100.100.1 iburst
```

2. `systemctl restart chronyd`

▼ ПУНКТ 7. ФАЙЛОВОЕ ХРАНИЛИЩЕ

▼ SRV-L

1. `lsblk`

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

2. `cfdisk /dev/sdb` → `gpt` → `NEW` → `TYPE` → `Linux RAID` → `WRITE` → `yes` → `QUIT`

3. С пункта 2 повторить действия со всеми остальными дисками (`sdb`, `sdc`, `sdd`, `sde`)

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

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

6. `mkfs.ext4 /dev/md0`

7. `mkdir /raid5`

8. `vim /etc/fstab`

a. добавить в конец, пишем через табуляцию:
`/dev/md0 /raid5 ext4 defaults 0 0`

```
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          relatime
UUID=c04e0177-5eb9-4410-8685-3396f4271cb4 swap           swap          defaults
/dev/md0      /raid5         ext4          defaults      0 0
```

9. reboot
10. systemctl restart nfs
11. mkdir /raid5/nfs
12. chmod 777 /raid5/nfs
13. vim /etc/exports
 - a. в конец добавляем:
/raid5/nfs 10.10.10.110(rw, sync)
20.20.20.100(rw, sync)
 - b.
10.10.10.110-адрес WEB-L
20.20.20.100-адрес WEB-R

▼ WEB-L

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

▼ ПУНКТ 8. DOCKER. MEDIAWIKI. 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/Local
    Settings.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

```

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



Установка MediaWiki 1.41.0

Язык

Ваш язык:

☐ справка ru - русский

Язык, который будет использовать вики:

☐ справка ru - русский

[Далее →](#)

Проверка внешней среды была успешно проведена. Вы можете установить MediaWiki.

Авторские права и условия

Эта вики работает на движке [MediaWiki](#), copyright © 2001-2024 Magnus Manske, Brion Vibber, Lee Daniel I Niklas Laxström, Domas Mituzas, Rob Church, Yuri Astrakhan, Aryeh Gregor, Aaron Schulz, Andrew Garrett, F Horohoe, Roan Kattouw, Trevor Parscal, Bryan Tong Minh, Sam Reed, Victor Vasiliev, Rotem Liss, Platonides, J Bartosz Dzielowski, Ed Sanders, Moriel Schottlender, Kunal Mehta, James D. Forrester, Brian Wolff, Adam Sh Hartman, Petr Pchelko, другие и переводчики [translatewiki.net](#).

MediaWiki — свободное программное обеспечение, которое вы можете распространять и/или изменять опубликованной фондом свободного программного обеспечения; второй версии, либо любой более п MediaWiki распространяется в надежде, что она будет полезной, но **без каких-либо гарантий**, даже б пригодности для определённой цели. См. лицензию GNU General Public License для более подробной Вы должны были получить копию GNU General Public License вместе с этой программой, если нет, то на Floor, Boston, MA 02110-1301, USA или прочтите её онлайн.

[← Назад](#) [Далее →](#)

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)

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](#)

Настройки базы данных

Учётная запись для доступа к базе данных из веб-сервера

☒ Использовать ту же учётную запись, что и для установки

[Назад](#) [Далее →](#)

Название

Название вики:

[справка](#)

wiki

Пространство имён проекта:

[справка](#)

☒ То же, что имя вики: Wiki

☐ Проект

☐ Другое (укажите)

Учётная запись администратора

Ваше имя участника:

[справка](#)

wiki

1. Пароль: DEP@ssw0rd, можно без почты

Пароль:

Пароль ещё раз:

Адрес электронной почты:

☒ [справка](#)

admin@demo.first

☒ [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

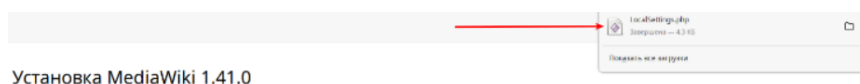
☒ Хватит, установить вики

[← Назад](#) [Далее →](#)

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 →](#)



Установка MediaWiki 1.41.0

Готово!

Поздравляем! Вы установили MediaWiki.

Во время установки был создан файл LocalSettings.php. Он содержит все ваши настройки. Вам необходимо скачать его и положить в корневую директорию вашей вики (ту же директорию, где находится файл index.php). Его загрузка должна начаться автоматически.

Если автоматическая загрузка не началась или вы её отменили, вы можете скачать по ссылке ниже:

[Загрузить LocalSettings.php](#)

Примечание: Если вы не сделаете этого сейчас, то созданный файл конфигурации не будет доступен вам в дальнейшем, если вы выйдете из установки, не скачивая его.

По окончании действий, описанных выше, вы сможете [войти в вашу вики](#).

- Язык
- Существующая вики
- Добро пожаловать в MediaWiki!
- Подключение к Базе данных
- Обновление существующей установки
- Настройка базы данных
- Название
- Настройки
- Установка
- Готово!

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

vim ~/wiki.yml

    раскомментируем
    - ./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

▼ ПУНКТ 9. УСТАНОВКА YANDEX BROWSER (не root - yandex-browser-stable; root - yandex-browser-stable --no-sandbox)