

Настройка доменного контроллера Samba на машине BR-SRV.

[SambaAD на Alt](#)

```
apt-get update && apt-get install task-samba-dc -y
```

удаляем файл /etc/samba/smb.conf ДО НАЧАЛА УСТАНОВКИ!!!

```
rm -f /etc/samba/smb.conf
```

Начинаем установку:

```
samba-tool domain provision --interactive
```

```
[root@BR-SRV ~]# samba-tool domain provision --interactive
Realm [AU-TEAM.IRPO]:
Domain [AU-TEAM]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:
DNS forwarder IP address (write 'none' to disable forwarding) [192.168.1.10]:
Administrator password:
Retype password:
INFO 2025-01-27 19:09:05,088 pid:16982 /usr/lib64/samba-dc/python3.9/samba/provi
INFO 2025-01-27 19:09:05,092 pid:16982 /usr/lib64/samba-dc/python3.9/samba/provi
WARNING 2025-01-27 19:09:05,096 pid:16982 /usr/lib64/samba-dc/python3.9/samba/pr
```

В качестве пароля указываем P@ssw0rd

Запускаем сервис

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

Меняем адрес DNS сервера на свой локальный

```
vim /etc/net/ifaces/ens19/resolv.conf
```

```
nameserver 127.0.0.1
search au-team.irpo
~
```

Перезапускаем сетевую службу

```
systemctl restart network
```

BR-SRV

Перемещаем сгенерированный конфиг krb5.conf:

```
mv -f /var/lib/samba/private/krb5.conf /etc/krb5.conf
```

Проверяем состояние домена

```
samba-tool domain info 127.0.0.1
```

```
[root@BR-SRV ~]# samba-tool domain info 127.0.0.1
Forest           : au-team.irpo
Domain           : au-team.irpo
Netbios domain   : AU-TEAM
DC name          : br-srv.au-team.irpo
DC netbios name  : BR-SRV
Server site      : Default-First-Site-Name
Client site      : Default-First-Site-Name
[root@BR-SRV ~]#
```

Домен работает.

Создадим пользователей

Создаем пользователей, создаем группу

```
[root@BR-SRV ~]# samba-tool user add user1.hq P@ssw0rd
User 'user1.hq' added successfully
[root@BR-SRV ~]# samba-tool user add user2.hq P@ssw0rd
User 'user2.hq' added successfully
[root@BR-SRV ~]# samba-tool user add user3.hq P@ssw0rd
User 'user3.hq' added successfully
[root@BR-SRV ~]# samba-tool user add user4.hq P@ssw0rd
User 'user4.hq' added successfully
[root@BR-SRV ~]# samba-tool user add user5.hq P@ssw0rd
User 'user5.hq' added successfully
[root@BR-SRV ~]# samba-tool group add hq
Added group hq
```

Добавляем пользователей в группу

```
samba-tool group addmembers hq
user1.hq,user2.hq,user3.hq,user4.hq,user5.hq
```

проверим

```
[root@BR-SRV ~]#
[root@BR-SRV ~]# samba-tool group listmembers hq
user5.hq
user4.hq
user3.hq
user2.hq
user1.hq
[root@BR-SRV ~]#
```

Смотрим настройки пользователя

```
samba-tool user show user1.hq
```

Из соображений безопасности все создаваемые УЗ блокированы

BR-SRV

Параметр accountExpires:

```
accountExpires: 9223372036854775807
logonCount: 0
```

Любое значение отличное от 0 обозначает, что учетная запись пользователя заблокирована

Разблокируем УЗ

```
samba-tool user setexpiry user1.hq --noexpiry
```

```
[root@BR-SRV ~]# samba-tool user setexpiry user1.hq --noexpiry
Expiry for user 'user1.hq' disabled.
[root@BR-SRV ~]#
```

Повторная проверка показывает, что мы разблокировали УЗ

```
userAccountControl: 66048
accountExpires: 0
whenChanged: 20250111103606.0Z
```

Выполните импорт пользователей из файла users.csv.

Создаём файл import.sh

```
#!/bin/bash
#
csv_file="/opt/users.csv"
while IFS=";" read -r firstName lastName role phone ou street zip city
country password; do
if [ "$firstName" == "First Name" ]; then
    continue
fi
username="${firstName,,}.${lastName,,}"
    samba-tool user add "$username" P@ssw0rd1
done < "$csv_file"
```

даем право на исполнение

```
chmod +x import.sh
```

и запускаем

```
./import.sh
```

После выполнения скрипта проверяем

```
samba-tool user list
```

Сконфигурируйте файловое хранилище

Проверяем наличие дисков

```
[root@BR-SRV ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda   8:0    0  10G  0 disk /
sdb   8:16    0   1G  0 disk
sdc   8:32    0   1G  0 disk
sdd   8:48    0   1G  0 disk
[root@BR-SRV ~]#
```

[Программный RAID в ALT](#)

Создадим дисковый массив уровня 5 из трёх дополнительных дисков следующей командой:

```
mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sd[b-d]
```

Посмотрим статус raid-массива:

```
cat /proc/mdstat
```

Сохраним конфигурацию массива в файл /etc/mdadm.conf следующей командой:

```
mdadm --detail --scan --verbose > /etc/mdadm.conf
```

теперь необходимо создать раздел

[Команда parted](#)

воспользуемся утилитой parted, уже установленной в ALT

```
parted /dev/md0
```

- 1) Необходимо создать таблицу разделов. Используем самый простой и распространённый тип MBR (msdos). Команда `mktable msdos`

```
[root@BR-SRV ~]# parted /dev/md0
GNU Parted 3.2.46-e4ae
Using /dev/md0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mktable msdos
```

- 2) Посмотрим таблицу разделов, чтобы выяснить размер свободного пространства.

Команда `print`

```
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 2143MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End    Size  Type  File system  Flags
```

Свободно 2143Mb

- 3) Создадим раздел. Команда **mkpart primary ext4 1 2143M**, где
- | | |
|---------|------------------------|
| primary | тип раздела |
| ext4 | метка файловой системы |
| 1 | начало раздела |
| 2143M | конец раздела |

```
(parted)
(parted) mkpart primary ext4 1 2143M
(parted)
(parted)
```

- 4) Снова посмотрим таблицу разделов

```
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 2143MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  2143MB  2142MB  primary ext4          lba

(parted)
```

- 5) Выходим из parted. Команда quit.

Проверим lsblk

```
[root@BR-SRV ~]# lsblk /dev/md0
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
md0          9:0    0    2G  0 raid5
└─md0p1 259:1    0    2G  0 part
```

Теперь создадим файловую систему, по заданию требуется ext4, создаём её следующей командой:

```
mkfs.ext4 /dev/md0p1
```

```
[root@BR-SRV ~]# mkfs.ext4 /dev/md0p1
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 523008 4k blocks and 130816 inodes
Filesystem UUID: 9b32d258-f704-4319-b0b8-e700bf1f88f5
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@BR-SRV ~]#
```

Настроим автоматическое монтирование в /raid5.

Создаём каталог /raid5:

```
mkdir /raid5
```

Обеспечим автоматическое монтирование в папку /raid5. Для этого добавим в конец файла /etc/fstab:

```
/dev/md0p1 /raid5 ext4 defaults 0 0
UUID=5cfdfaf0-839f-4578-ad0f-b52b0336dfe2 / ext4 relatime
1
/dev/sr0 /media/ALTLinux udf,iso9660 ro,noauto,user=utf8,nofail,comm
gvfs-show 0 0
/dev/md0p1 /raid5 ext4 defaults 0 0
~
~
```

[fstab в ALT](#)

Выполним команду монтирования

```
mount -a
```

Не должно быть никаких сообщений!

Посмотрим точки монтирования командой df

```
[root@BR-SRV ~]# mount -a
[root@BR-SRV ~]#
[root@BR-SRV ~]# df
Filesystem      Size  Used Avail Use% Mounted on
udevfs          5.0M   64K  5.0M   2% /dev
runfs           991M  608K  990M   1% /run
/dev/sda        9.8G  3.0G  6.3G  32% /
tmpfs           991M     0  991M   0% /dev/shm
tmpfs           991M     0  991M   0% /tmp
tmpfs           199M     0  199M   0% /run/user/0
/dev/md0p1      2.0G   24K  1.9G   1% /raid5
[root@BR-SRV ~]#
```

Настройте сервер сетевой файловой системы(nfs)

Создадим файловые ресурсы и настроим права к ним

```
mkdir /raid5/nfs
chmod 777 /raid5/nfs
```

Установим необходимое ПО

```
apt-get install nfs-server
```

[NFS в ALT](#)

запустим сервер NFS

```
systemctl enable --now nfs-server
```

пропишем доступ к каталогу в файле /etc/exports

```
vim /etc/exports
```

BR-SRV

```
#/srv/share -rw,insecure,fsid=0,sec=krb5 *
#####
/raid5/nfs 192.168.2.0/28(rw,no_subtree_check)
#####
~
```

Применим наши настройки

```
[root@BR-SRV ~]# exportfs -vra
exporting 192.168.2.0/28:/raid5/nfs
exporting */srv/public
[root@BR-SRV ~]#
```

Ansible на сервере BR-SRV

Предварительные настройки

Включаем SSH на HQ-CLI

```
HQ-CLI ~ # systemctl enable --now sshd.service
Synchronizing state of sshd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable sshd
Created symlink /etc/systemd/system/multi-user.target.wants/sshd.service → /lib/systemd/system/sshd.service.
HQ-CLI ~ #
```

Создаем ключевую пару на BR-SRV

```
[root@BR-SRV ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
```

Настраиваем бесключевой доступ на HQ-SRV и HQ-CLI

```
[root@BR-SRV ~]# ssh-copy-id user@192.168.1.10
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
```

```
Last login: Tue Jan 28 18:44:00 MSK 2025 on ttyS0
[root@BR-SRV ~]# ssh-copy-id user@192.168.2.10
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
The authenticity of host '192.168.2.10 (192.168.2.10)' can
```

Устанавливаем ansible:

```
apt-get install ansible
```

редактируем конфиг:

```
vim /etc/ansible/ansible.cfg
```

```
[defaults]

# some basic default values...

#####
inventory      = /etc/ansible/hosts
interpreter_python = auto_silent
#####

#library        = /usr/share/my_modules/
#module_utils    = /usr/share/my_module_utils/
```

Создаем инвентарный файл:

```
vim /etc/ansible/hosts
```

```
[Linux]
hq-srv ansible_host=user@192.168.1.10
hq-cli ansible_host=user@192.168.2.10

[ECO_ROUTERS]
hq-rtr ansible_host=192.168.5.1
br-rtr ansible_host=192.168.5.2

[ECO_ROUTERS:vars]

ansible_connection=network_cli
ansible_network_os=ios
ansible_user=admin
ansible_password=admin
```

Выполняем проверку доступности

```
ansible all -m ping
```

```
[root@BR-SRV ansible]#
[root@BR-SRV ansible]#
[root@BR-SRV ansible]# ansible all -m ping
br-rtr | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hq-rtr | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hq-srv | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
hq-cli | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[root@BR-SRV ansible]#
```


Развертывание приложений в Docker на сервере BR-SRV

Установить docker-engine и docker-compose:

```
apt-get install docker-engine docker-compose -y
```

запустим службу docker:

```
systemctl enable --now docker
```

добавляем пользователя sshuser в группу docker, что он имел возможность работать с контейнерами:

```
usermod sshuser -aG docker
```

```
[root@BR-SRV ~]# usermod sshuser -aG docker
[root@BR-SRV ~]# grep docker /etc/group
docker:x:465:sshuser
[root@BR-SRV ~]#
```

Переходим в контекст пользователя sshuser

```
su -l sshuser
```

[Install MediaWiki using Docker and Docker Compose](#)

[More](#)

Загружаем образы следующей командой:

```
docker pull mediawiki
docker pull mariadb
```

Создаем в домашней директории пользователя файл wiki.yml для приложения MediaWiki.

```
vim wiki.yml
```

```
services:
  wiki:
    image: mediawiki
    container_name: wiki
    environment:
      MEDIAWIKI_DB_HOST: mariadb
      MEDIAWIKI_DB_USER: wiki
      MEDIAWIKI_DB_PASSWORD: WikiP@ssw0rd
      MEDIAWIKI_DB_NAME: mediawiki
    ports:
      - 8080:80
    # volumes: [ ~/LocalSettings.php:/var/www/html/LocalSettings.php ]

  db:
    image: mariadb
    container_name: mariadb
    hostname: mariadb
    environment:
      MYSQL_DATABASE: mediawiki
      MYSQL_USER: wiki
      MYSQL_PASSWORD: WikiP@ssw0rd
      MYSQL_ROOT_PASSWORD: P@ssw0rd
    volumes:
      - ./db:/var/lib/mysql

volumes:
  db:
```

BR-SRV

Проверяем конфигурацию

```
docker-compose -f wiki.yml config
```

после проверки, запускаем

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

```
"wiki.yml" 26L, 620B written
[sshuser@BR-SRV ~]$ docker compose -f wiki.yml up -d
[+] Running 2/2
 ✓ Container mariadb Started
 ✓ Container wiki Started
[sshuser@BR-SRV ~]$
```

Переходим на HQ-CLI, в браузер по адресу <http://192.168.3.10:8080>

справка

mariadb

☐ Подключиться через SSL

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

..... WikiP@ssw0rd

Заполняем поля, **Далее**

BR-SRV

Название

Название вики:
[i справка](#)
DEMO25

Пространство имён проекта:
[i справка](#)
☒ ☐ ☐

То же, что имя вики: DEMO25

☐ Проект

☐ Другое (укажите)

Учётная запись администратора

Ваше имя участника:
[i справка](#)
wiki

Пароль:
.....

Пароль ещё раз: WikiP@ssw0rd
.....

Адрес электронной почты:
[i справка](#)

По окончании настройки загружаем LocalSettings.php

Копируем его на BR-SRV, в домашний каталог sshuser

```
administrator@HQ-CLI Downloads $  
administrator@HQ-CLI Downloads $ scp LocalSettings.php sshuser@192.168.3.10:/home/sshuser/  
sshuser@192.168.3.10's password:  
administrator@HQ-CLI Downloads $
```

```
sshuser@BR-SRV ~]$  
sshuser@BR-SRV ~]$  
sshuser@BR-SRV ~]$ cat LocalSettings.php  
?php  
This file was automatically generated by the MediaWiki 1.43.0  
installer. If you make manual changes, please keep track in case you  
need to recreate them later.  
See includes/MainConfigSchema.php for all configurable settings
```

После этого останавливаем все контейнеры

```
docker stop $(docker ps -a -q)
```

удаляем все контейнеры

```
docker rm $(docker ps -a -q)
```

Раскомментируем строку wiki.yml


```
- 8080:80  
volumes: [ ~/LocalSettings.php:/var/www/html/LocalSettings.php ]
```

BR-SRV

и снова запускаем wiki.yml

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

На HQ-CLI, в браузер по адресу <http://192.168.3.10:8080>



Искать в DEMO25

Найти

[Создать учётную запись](#) [Войти](#)

Содержание

скрыть

Начало

[Начало работы](#)

Заглавная страница

Заглавная Обсуждение Читать Править История Инструменты

MediaWiki успешно установлена.

Информацию по работе с этой вики можно найти в [справочном руководстве](#).

Начало работы [\[править\]](#)

- [Список возможных настроек](#);
- [Часто задаваемые вопросы и ответы по MediaWiki](#);
- [Рассылка уведомлений о выходе новых версий MediaWiki](#);
- [Перевод MediaWiki на свой язык](#);
- [Узнайте, как бороться со спамом в вашей вики](#)

Внешний вид

скрыть

Текст

☒ Мелкий

☐ Стандартный

☐ Крупный

Ширина

☒ Стандартно

☐ Широко

Wiki работает