

# Инструкция по настройке холодного резерва на базе nginx (Linux)

## Содержание

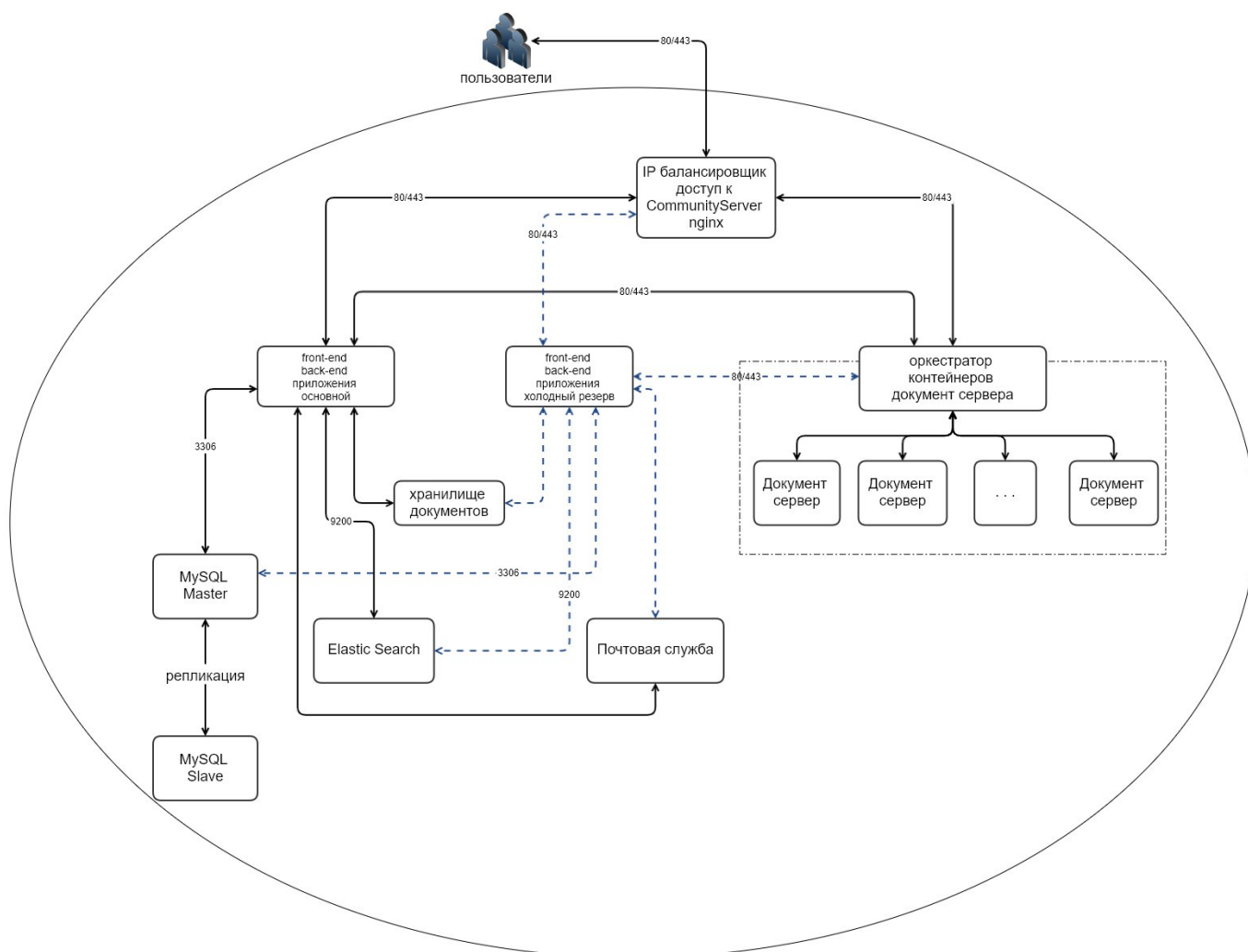
Инструкция по настройке холодного резерва на базе nginx (Linux).....	1
Содержание.....	2
1.1 Холодный резерв сервера приложений.....	5
1.2 Reverse проху/Балансировщик нагрузки.....	5
1.3 Почтовая служба.....	5
1.4 Система полнотекстового поиска.....	5
1.5 Хранение документов.....	6
1.6 Система управления базами данных.....	6
1.7 Служба документ сервера.....	6
2. Настройка nginx хоста.....	7
2.1 Изменения конфигурационных файлов.....	7
2.1.1 nginx.conf.....	7
2.1.2 Конфигурационный файл для DS.....	7
2.1.2 Конфигурационный файл для CS.....	8
2.1.3 Общий конфигурационный файл для (reverse) проху.....	9
3 Настройка хостов CS.....	11
3.1 Вынос MySQL.....	11
3.2 Вынос DS.....	12
3.3 Вынос почтового сервера.....	12
3.4 Вынос Elasticsearch.....	12
3.5 Хранение документов.....	13
3.5.1 Настройка OMV.....	13
3.5.1.1 Подключаем диск к OMV.....	14
3.5.1.2 Включение NFS в OMV.....	16
3.5.2 Подготовка Linux.....	18
3.5.2.1 Для yum, dnf.....	18
3.5.2.2 Для apt, apt-get.....	18
3.5.2.3 Создайте директорию.....	18
3.5.2.4 Примонтируйте nfs каталог.....	18
3.5.2.5 Добавьте в fstab запись.....	18
3.5.3 Настройка конфигурационных файлов на CS.....	18
3.5.3.1 WebStudio\web.storage.config.....	18
3.5.3.2 web.appsettings.config.....	19
3.5.3.3 TeamLabSvc.exe.config.....	19
3.5.3.4 radicale.config.....	19
3.5.3.5 Jabber/web.storage.config.....	19
3.5.3.6 storage.production.json.....	19
3.5.3.7 storage.json.....	20
3.5.3.8 Переносим данные.....	20

3.5.3.9	Перезапускаем сервисы.....	20
3.5.3.10	Обновить права на каталоги.....	20
4	Проверка работы сервера.....	21
5	Создание клона сервера.....	21
6.	Проверка hostname.....	22
6.1	Изменить hostname.....	22

# 1. Архитектура

Холодный резерв применим к архитектуре **Full**, т.к. большинство модулей должны быть вынесены из основного приложения, чтобы, при выходе из строя первой ноды, вторая имела доступ ко всем данным. Основная задача архитектуры **Full** - решение задачи по обеспечению механизмов холодного резервирования сервера приложения, оптимизация нагрузки, которая генерируется поисковой и почтовой службой.

На рисунке №1 представлена архитектура **Full**.



Архитектура типа **full**. Рис.1

## 1.1 Холодный резерв сервера приложений

Резервирование контура достигается путём установки функциональных модулей на отдельные хосты, где возможна реализация отказоустойчивости штатными средствами.

Реализация холодного резерва осуществляется за счет дублирования **front-end** и **back-end** приложений. В случае проблем на основном сервере приложений - требуется ручное переключение на холодный резерв. Ключевой деталью, для реализации отказоустойчивого решения, с применением холодного резерва сервера приложений, является использование обособленного хранилища документов и базы данных. Сервер холодного резервирования имеет аналогичные настройки на подключение к почтовой системе, службе документ сервера, системе полнотекстового поиска и к другим системам, с которыми взаимодействует основной сервер. В период штатной эксплуатации резервный сервер должен быть выключен.

## 1.2 Reverse proxy/Балансировщик нагрузки

Для реализации автоматического переключения запросов пользователей между серверами приложений, используется прокси сервер **Nginx**. Резервный сервер должен быть описан в директиве **upstream** и помечен, как запасной - **backup**.

Таким образом, при НШС, пользовательские запросы на доступ к Приложению идут по стандартному **dns** имени/**ip** адресу, которые обрабатываются балансировщиком нагрузки и направляются на функционирующий сервер.

## 1.3 Почтовая служба

Для снижения нагрузки на основной сервер, предполагается вынос почтовой системы на **Linux** сервер.

## 1.4 Система полнотекстового поиска

Для реализации функционала полнотекстового поиска приложение **Community server**, использует инструмент **Elasticsearch**. В варианте архитектуры **Full**, предполагается использование **Elasticsearch**, расположенного на отдельных серверах, либо, в случае, если позволяет ландшафт корпоративной сети, где внедряется Приложение, использовать сервис в виде **Platform as a Service (PaaS)**.

Для масштабируемости блока полнотекстового поиска возможно использовать штатные механизмы продукта **Elasticsearch**.

## 1.5 Хранение документов

Для обеспечения переключения работы на резервный сервер приложений, требуется хранение документации за пределами основного и резервного сервера приложений на отдельном файловом хранилище.

## 1.6 Система управления базами данных

Как и в случае с архитектурой **Middle**, резервирование системы и реализация отказоустойчивости СУБД достигается штатными средствами **MySQL**.

## 1.7 Служба документ сервера

Как и в случае с архитектурой **Middle**, размещение документ сервера предусматривается, в данном варианте инсталляции, на отдельном сервере, как в версии **Docker** контейнера, так и с применением обычной установки.

## 2. Настройка nginx хоста

**Примечание:** В предыдущих уроках мы уже произвели настройку **nginx**, поэтому далее рассмотрим только его донастройку. [Ссылка](#) на инструкцию.

### 2.1 Изменения конфигурационных файлов

#### 2.1.1 nginx.conf

Привести файл к виду (в зависимости от версии **nginx** и ОС, может отличаться):

```
/etc/nginx/nginx.conf
```

**Примечание:** версия **nginx/1.22.1**, ОС **RedOS7.3**

```
user nginx;
worker_processes auto;
worker_cpu_affinity auto;
error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
    multi_accept on;
    use epoll;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    gzip on;
    include /etc/nginx/conf.d/*.conf;
}
```

## 2.1.2 Конфигурационный файл для DS

Файл не изменяется

```
/etc/nginx/conf.d/r7-ds.conf
```

## 2.1.2 Конфигурационный файл для CS

```
/etc/nginx/conf.d/r7-cs.conf
```

Приведите его к виду:

```
server {
    listen 80;
    server_name { DNS_CS };
    server_tokens off;

    ## Redirects all traffic to the HTTPS host
    return 301 https://$server_name$request_uri;
}
server {
    listen 443 ssl http2;
    server_name { DNS_CS };

    # пути до сертификатов
    ssl_certificate { PAHT_FOR_CERT };
    ssl_certificate_key { PAHT_FOR_KEY };
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
    ssl_prefer_server_ciphers on;
    large_client_header_buffers 4 64k;
    client_max_body_size 50m;

    location / { # Контекст описывающий проксирование сервера приложения
        proxy_pass http://cs;
        proxy_http_version 1.1;

        proxy_send_timeout 7d;
        proxy_read_timeout 7d;
        proxy_connect_timeout 7d;
        proxy_redirect off;
        proxy_next_upstream timeout error http_500 http_502 http_503 http_504 http_404;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $proxy_connection;
        proxy_set_header X-Forwarded-Host $the_host;
        proxy_set_header X-Forwarded-Proto $the_scheme;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Где,

```
{ DNS_CS } – имя в dns для CS, создавалось в инструкции для reverse proxy;  
{ PAHT_FOR_CERT } – полный путь до ssl сертификата;  
{ PAHT_FOR_KEY } – полный путь до приватного ключа ssl сертификата.
```

Параметры `proxy_send_timeout`, `proxy_read_timeout`, `proxy_connect_timeout` можно регулировать, т.к. они выставлены для данной сборки и времени ответов серверов.

### 2.1.3 Общий конфигурационный файл для (reverse) proxy

```
/etc/nginx/conf.d/r7-proxy.conf
```

Приведите файл к виду:

```
upstream ds {  
    server { IP_DS };  
}  
  
upstream cs {  
    zone upstream_dynamic 64k;  
    server { IP_CS };  
    server { IP_CS_RESERVE } backup;  
}  
  
map $http_host $this_host {  
    "" $host;  
    default $http_host;  
}  
  
map $http_x_forwarded_proto $the_scheme {  
    default $http_x_forwarded_proto;  
    "" $scheme;  
}  
  
map $http_x_forwarded_host $the_host {  
    default $http_x_forwarded_host;  
    "" $this_host;  
}  
  
map $http_upgrade $proxy_connection {  
    default upgrade;  
    "" close;  
}  
  
resolver 77.88.8.8 ipv6=off;  
proxy_buffer_size 64k;  
proxy_buffers 16 32k;
```

```
proxy_max_temp_file_size 0;
```

Где,

{ IP\_DS } – ip документсервера;

{ IP\_CS } – ip основного сервера приложений;

{ IP\_CS\_RESERVE } – ip дублирующего сервера приложений.

resolver 77.88.8.8 – ip dns яндекса, можете выставить Ваш dns или иной;

Выполните команды для проверки конфигурационных файлов и их перечитки:

```
nginx -t  
systemctl reload nginx
```

**Важное примечание:** Есть 2 варианта дальнейшего решения.

1. Провести действия, что описаны ниже, ТОЛЬКО на ОДНОЙ ВМ и сделать её «клон», что потребует меньших усилий в настройке и требуемого времени. Данный вариант рекомендуется, потому что повлечёт за собой меньше ошибок в редактировании конфигурационных файлов и т.п. (Добавить доступ во все **MySQL** нужно будет для второй ноды, если ограничивали его по **ip**)
2. Поднять чистую ВМ и провести такие же настройки и инсталляцию CS, что может привести к предположительным ошибкам при редактировании конфигурационных файлов.

### 3 Настройка хостов CS

**Примечание1:** Как вариант, можно сделать клон уже настроенной первой ноды, он будет полностью повторять в настройках первую. Необходимо будет проверить работу всех модулей, и, что изменения отражаются и на портале первой ноды. Как минимум, потребуются добавить доступы к **MySQL** на почтовом сервере СУБД и СУБД основного приложения (доступ к БД в инструкциях делался на конкретный **ip**, если Вы ставили значение ‘%’, то доступ будет ко всем **ip**).

**Примечание2:** Так как мы уже выполняли подключение всех модулей на других уроках, далее рассмотрим только ключевые моменты, которые необходимо будет проверить.

#### 3.1 Вынос MySQL

Вынос подробно описан по [данной](#) инструкции начиная с п.4, поэтому описывать подробно не будем. Замечу, что необходимо будет создать пользователя со стороны хоста с **MySQL** для резервной ноды, пример команд:

```
CREATE USER 'r7-office'@'IP_CS_RESERV' IDENTIFIED WITH mysql_native_password BY 'password_user_for_cs_reserve';  
GRANT ALL PRIVILEGES ON *.* TO 'r7-office'@'IP_CS_RESERV';  
FLUSH PRIVILEGES;
```

Где,

```
IP_CS_RESERV – ip резервной ноды;  
password_user_for_cs_reserve – пароль для пользователя.
```

## 3.2 Вынос DS

**DS** выносится также, как и для основной ноды, поэтому дублироваться не будет, т.к. нет отличий в выносе модуля. [Ссылка](#) на **middle** (начиная с п.5). Напомню лишь путь до конфигурационных файлов:

Со стороны DS:

```
/etc/r7-office/documentserver/local.json
```

Со стороны CS 2 файла:

```
/var/www/r7-office/WebStudio/web.appsettings.config
```

```
/var/www/r7-office/Services/TeamLabSvc/TeamLabSvc.exe.config
```

**Примечание:** В случае использования Сервера документов версии 7.3.3 и подключение к Корпоративному серверу CS 12.1 и более ранней версии, то функциональность "Предоставления доступа" и Загрузки файлов из "Хранилища" во время работы с Документом, при использовании **HTTP** протокола могут не работать. Для решения проблемы потребуется перевести на **HTTPS** протокол серверы с CS и DS:

**CS Linux:** [https://support.r7-office.ru/community\\_server/install-community\\_server/community\\_server\\_linux/community\\_server\\_https\\_own\\_certificate/](https://support.r7-office.ru/community_server/install-community_server/community_server_linux/community_server_https_own_certificate/)

**DS Linux:** [https://support.r7-office.ru/document\\_server/install-document\\_server/document\\_server\\_linux/https\\_ds/](https://support.r7-office.ru/document_server/install-document_server/document_server_linux/https_ds/)

Сервисы Корпоративный сервер и Сервер документов должны быть на одном домене и на одном его уровне

**Например,**

**CS** на домене второго уровня stgr7.ru и имеет FQDN имя = cs.stgr7.ru и

**DS** на домене второго уровня stgr7.ru и имеет FQDN имя = ds.stgr7.ru

### 3.3 Вынос почтового сервера

[Ссылка](#) на инструкцию по настройке (пункт 2).

Перед подключением почтового сервера, необходимо подключиться к **MySQL** на хосте, где развёрнута БД почты и создать дополнительного пользователя, выполнив команды:

**Примечание:** если пользователя создавали не с **ip**, а с **%** (например **'r7-office\_user'@'%'**), что означает подключение с любых **ip**, можно сразу подключить вторую ноду к почтовому серверу

```
GRANT ALL PRIVILEGES ON *.* TO 'r7-office_user'@'IP_CS_RESERVE' IDENTIFIED BY  
'PASSWORD_CS_RESERVE';  
FLUSH PRIVILEGES;
```

Где,

```
IP_CS_RESERVE – ip резервной ноды приложения;  
PASSWORD_CS_RESERVE – пароль для пользователя r7-office_user.
```

### 3.4 Вынос Elasticsearch

[Ссылка](#) на инструкцию по выносу (пункт 3).

Необходимо будет добавить настройки по выносу со стороны **CS**, согласно инструкции выше.

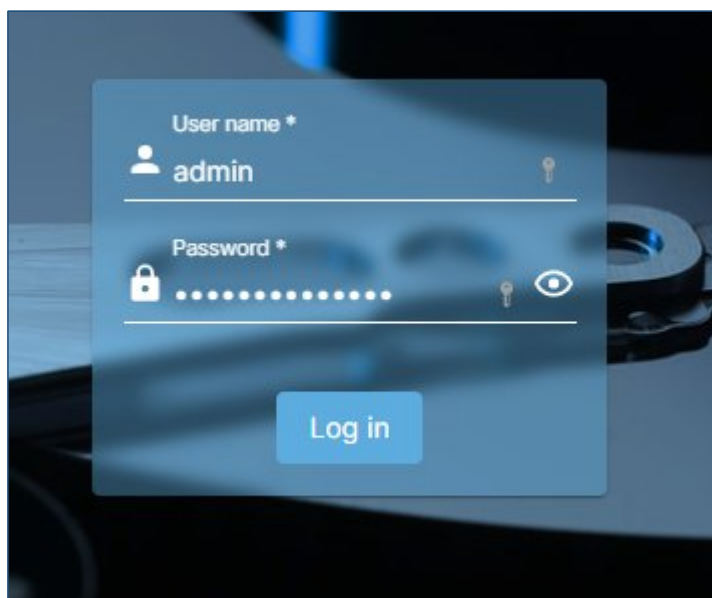
## 3.5 Хранение документов

В данном примере будет использован **NFS**-сервер на базе дистрибутива **OpenMediaVault**. С этого хоста будет расшарен каталог для общих файлов **CS**. [Ссылка](#) на дистрибутив.

Рассмотрим дополнительно вынос на **nfs**-хранилище, т.к. в предыдущей инструкции мы рассмотрели только вынос на «физически» подключенный дополнительный диск для файлов пользователей.

### 3.5.1 Настройка OMV

После инсталляции дистрибутива будет доступен портал по **http:ip-сервера** (например: **http://192.168.0.15**)



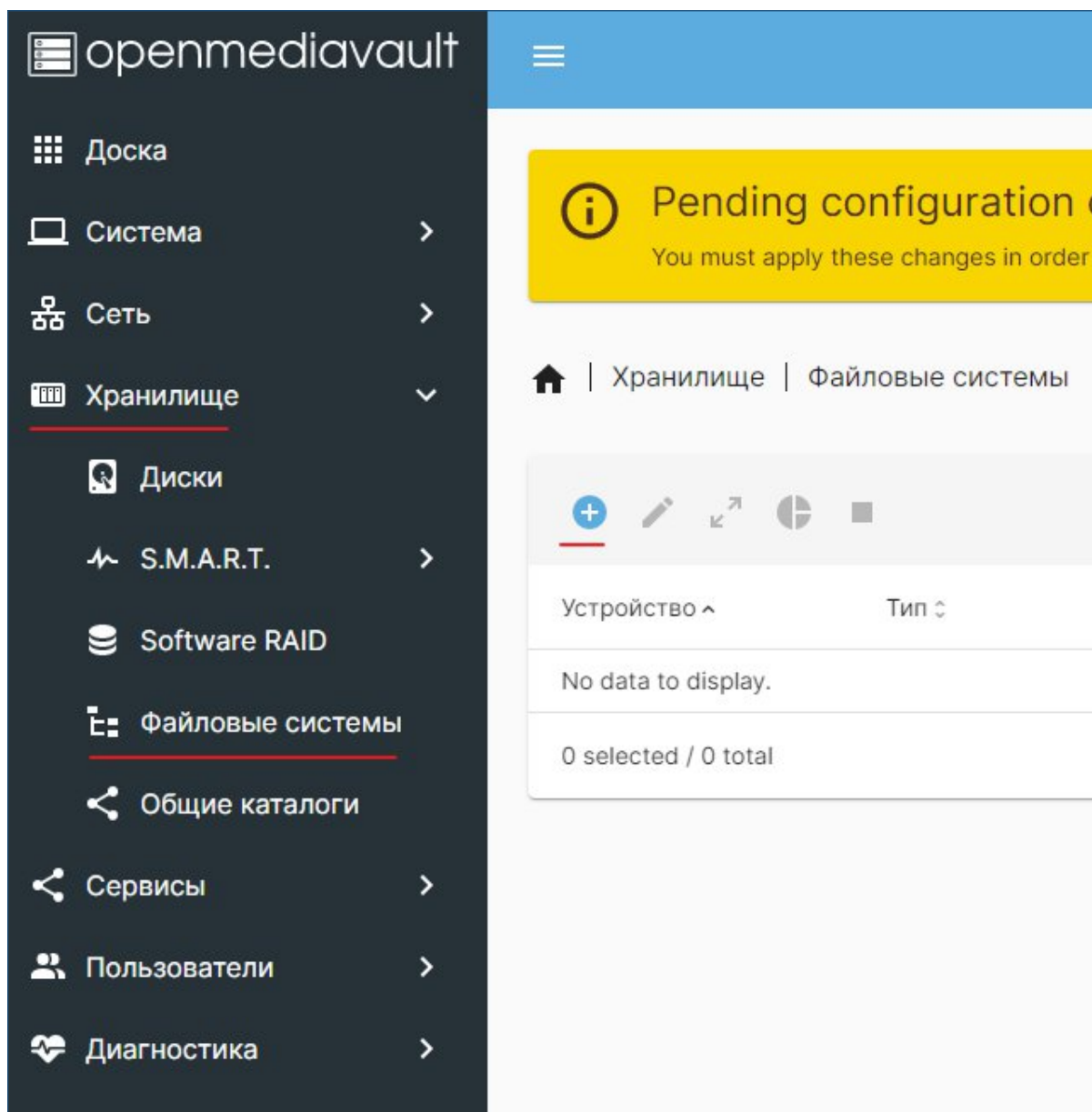
**Логин:** admin

**Пароль:** openmediavault

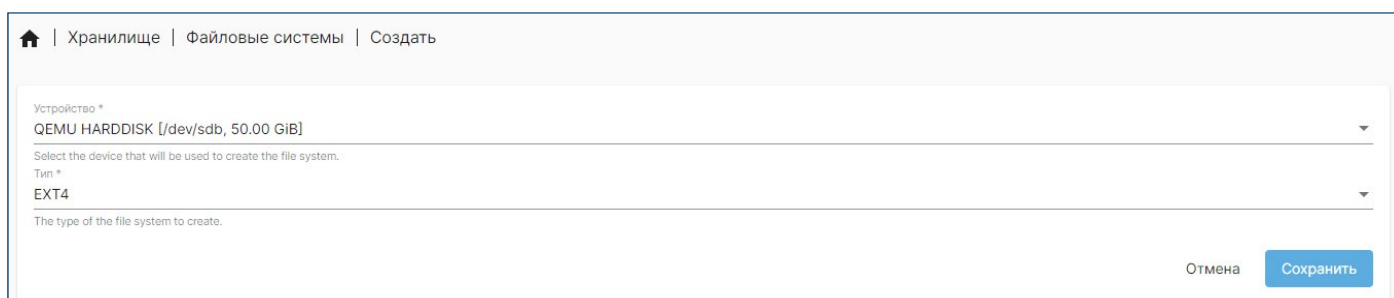
### 3.6.1.1 Подключаем диск к OMV

Добавляем к ВМ с **OMV** дистрибутивом диск для данных.

Переходим на вкладку “**Файловые системы**” и нажимаем “**Создать**” (+):



Выбираем наш диск **/dev/sdb** и файловую систему (**EXT4**) и нажимаем “**Сохранить**”:



После данной строки в выводе нажимаем “Закреть”


```
Создать файловую систему

Discarding device blocks:      0/13106939
Creating filesystem with 13106939 4k blocks
Filesystem UUID: b7bd8803-1ccb-460f-b255-034
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
    4096000, 7962624, 11239424

Allocating group tables:      0/400
Writing inode tables:         0/400      50/400
                                done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

END OF LINE
```

Далее откроется меню “Монтировать”, выбираем новый раздел `/dev/sdb1`. По желанию оставляем комментарий и нажимаем “Сохранить”:



Home | Хранилище | Файловые системы | Монтировать

Файловая система \*  
`/dev/sdb1 [EXT4, 49.99 GiB]`

The file system to mount.

Usage Warning Threshold \*  
`85%`

Send a notification when the used file system capacity exceeds the specified threshold.

Комментарий  
`data для CS`

Отмена Сохранить

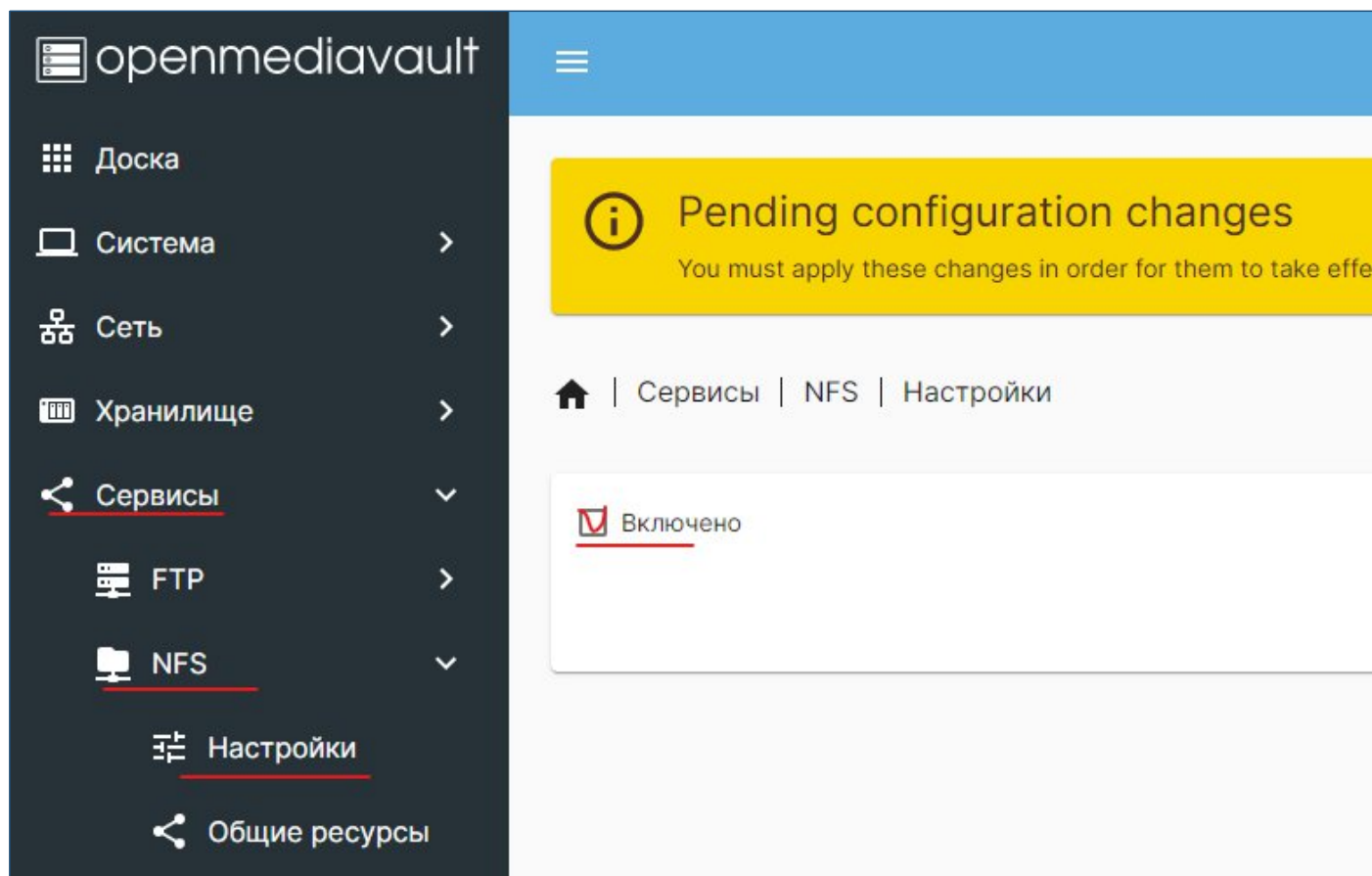
Применяем изменения, нажав галочку и **Ok**



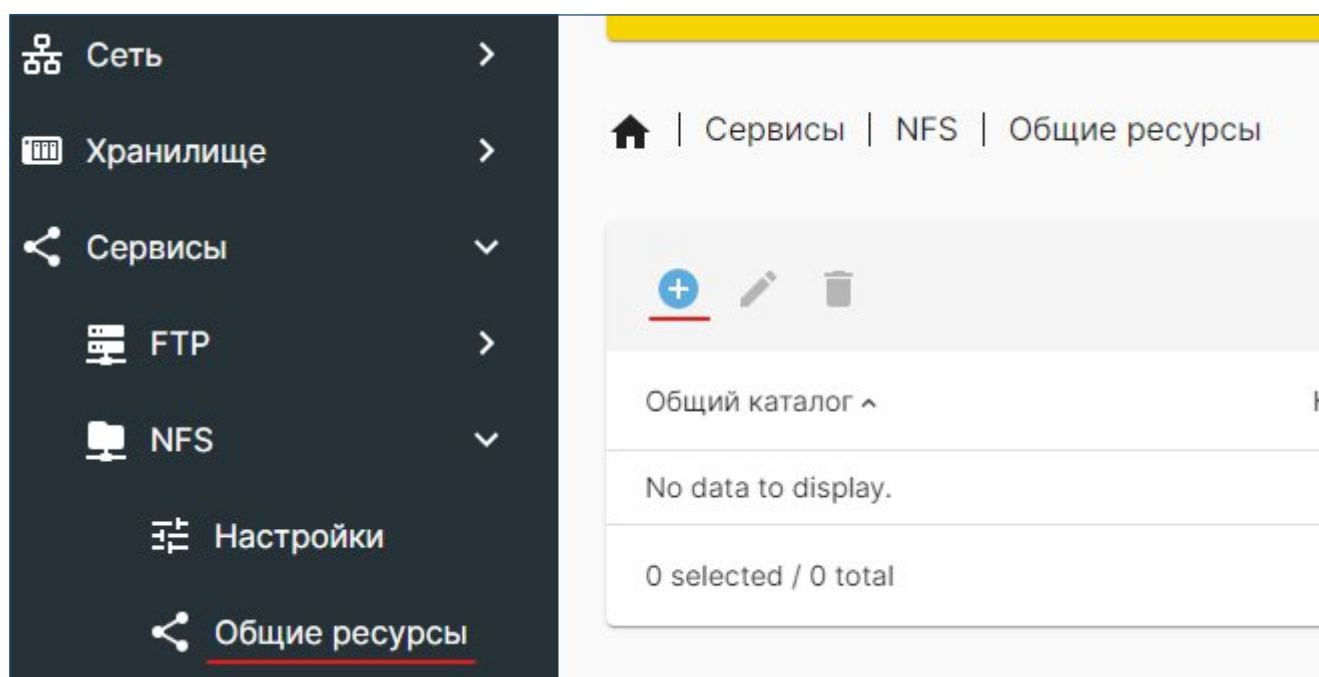
**Pending configuration changes**  
You must apply these changes in order for them to take effect.

### 3.6.1.2 Включение NFS в OMV

Переходим в “Сервисы” → “NFS” → “Настройки”, где “Включено” ставим галочку и жмём “Сохранить”:



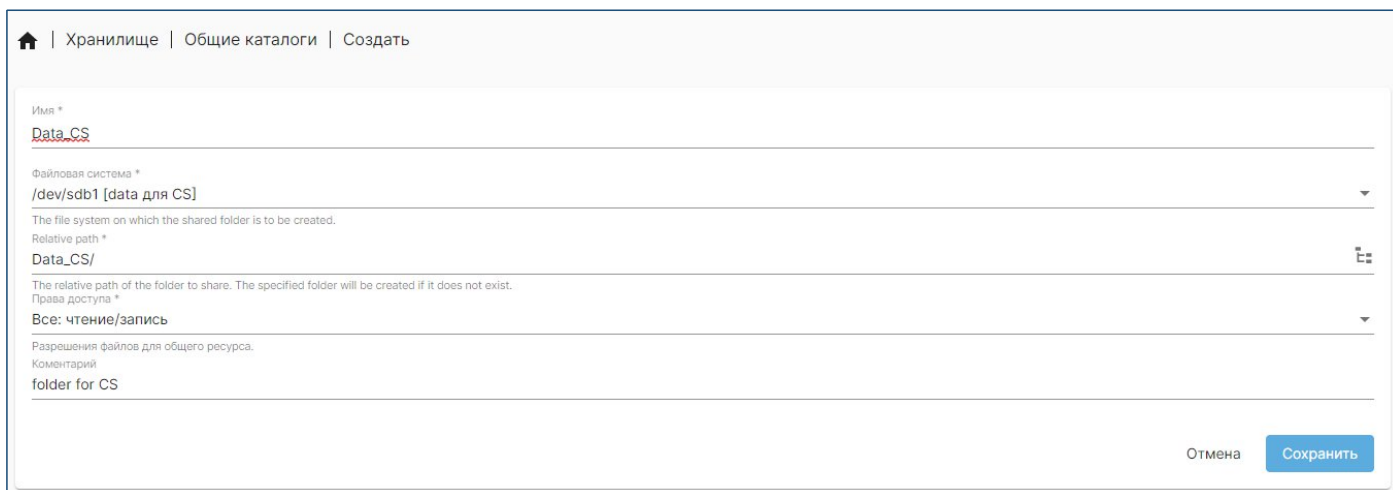
Переходим в “NFS” → “Общие ресурсы” и нажимаем “Создать” (+):



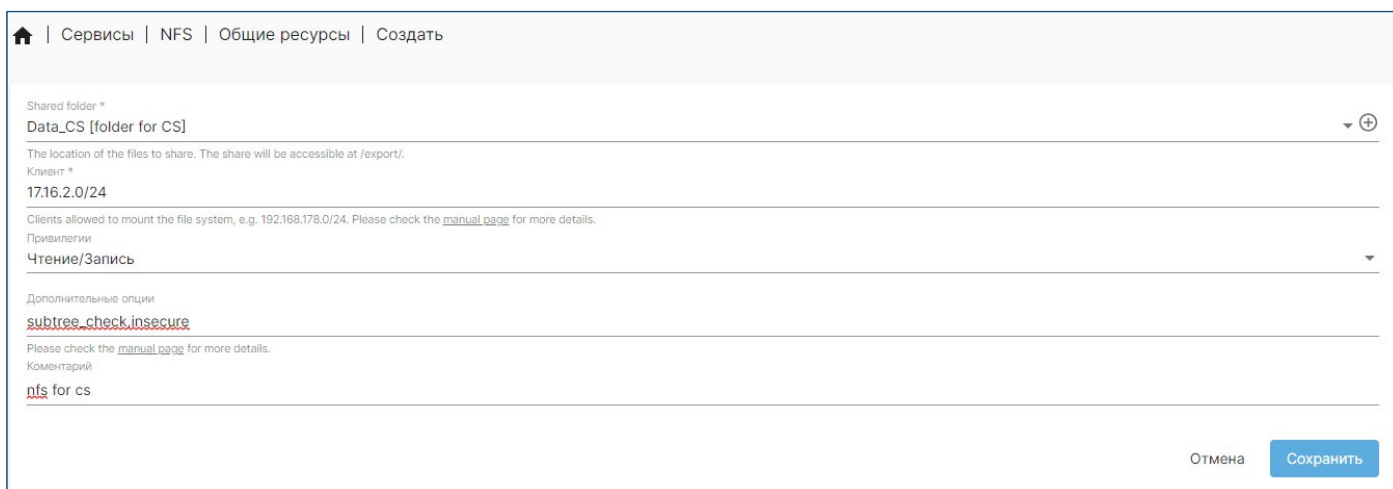
В данном разделе, в графе “**Shared folder**”, нажимаем “**Создать**” (+):



Заполняем все поля и нажимаем “**Сохранить**”. Здесь создали общий каталог для использования в **nfs**-хранилище.



Теперь выбираем созданный общий каталог, вписываем подсеть или **ip** адреса, которые будут иметь доступ к хранилищу, и нажимаем “**Сохранить**”:



Применяем изменения, нажав галочку и **Ok**



### 3.5.2 Подготовка Linux

Необходимо установить **nfs**-клиент.

#### 3.5.2.1 Для yum, dnf

Выполнить команду:

```
yum install nfs-utils
```

#### 3.5.2.2 Для apt, apt-get

Выполнить команду:

```
apt update  
apt install nfs-common
```

#### 3.5.2.3 Создайте директорию

Выполните команду

```
mkdir /mnt/nfsr7
```

#### 3.5.2.4 Примонтируйте nfs каталог

Выполните команду:

```
sudo mount -t nfs 172.16.2.235:/export/cs-lin /mnt/nfsr7
```

Где,

```
172.16.2.235 – ip адрес nfs сервера;  
/export/cs-lin – каталог, который сервер экспортирует;  
/mnt/nfsr7 – каталог, куда необходимо примонтировать локально.
```

#### 3.5.2.5 Добавьте в fstab запись

Отредактируйте **/etc/fstab**, добавив в конце запись:

```
172.16.2.235:/export/cs-lin /mnt/nfsr7 nfs defaults 0 0
```

### 3.5.3 Настройка конфигурационных файлов на CS

**Примечание:** делаем все пункты на каждой ноде **CS**, кроме п.3.5.3.8, его достаточно будет сделать только на основной ноде.

#### 3.5.3.1 WebStudio\web.storage.config

Файл находится по пути:

```
/var/www/r7-office/WebStudio/web.storage.config
```

Привести к виду параметр **\$STORAGE\_ROOT** (~ 10 строка):

```
<property name="$STORAGE_ROOT" value="/mnt/nfsr7/" />
```

### 3.5.3.2 web.appsettings.config

Файл находится по пути:

```
/var/www/r7-office/WebStudio/web.appsettings.config
```

Привести к виду параметры **temp** и **license.file.path** (~ 68 и 92 строки):

```
<add key="temp" value="/mnt/nfsr7/temp" />
...
<add key="license.file.path" value="/mnt/nfsr7/license.lic" />
```

### 3.5.3.3 TeamLabSvc.exe.config

Файл находится по пути:

```
/var/www/r7-office/Services/TeamLabSvc/TeamLabSvc.exe.config
```

Привести к виду параметры **temp** и **license.file.path** (~ 145 и 150 строки):

```
<add key="temp" value="/mnt/nfsr7/temp" />
...
<add key="license.file.path" value="/mnt/nfsr7/license.lic" />
```

### 3.5.3.4 radicale.config

Файл находится по пути:

```
/var/www/r7-office/Services/TeamLabSvc/radicale.config
```

Привести к виду значение **filesystem\_folder** (~ 94 строка):

```
# Folder for storing local collections, created if not present
filesystem_folder = /mnt/nfsr7/radicale/collections/
```

### 3.5.3.5 Jabber/web.storage.config

Файл находится по пути:

```
/var/www/r7-office/Services/Jabber/web.storage.config
```

Привести к виду параметр **\$STORAGE\_ROOT** (~ 10 строка):

```
<property name="$STORAGE_ROOT" value="/mnt/nfsr7/" />
```

### 3.5.3.6 storage.production.json

Файл находится по пути:

```
/etc/r7-office/communityserver/storage.production.json
```

Привести к виду параметр **\$STORAGE\_ROOT** (~ 8 строка):

```
{  
  "name": "$STORAGE_ROOT",  
  "value": "/mnt/nfsr7"  
}
```

### 3.5.3.7 storage.json

Файл находится по пути:

```
/etc/r7-office/communityserver/storage.json
```

Привести к виду параметр **\$STORAGE\_ROOT** (~ 16 строка):

```
{  
  "name": "$STORAGE_ROOT",  
  "value": "\\mnt\\nfsr7\\"  
}
```

### 3.5.3.8 Переносим данные

Копируем всё из каталога **Data** на наше **nfs**-хранилище с сервера **CS**:

```
cd /var/www/r7-office/Data/  
cp -r ./* /mnt/nfsr7/
```

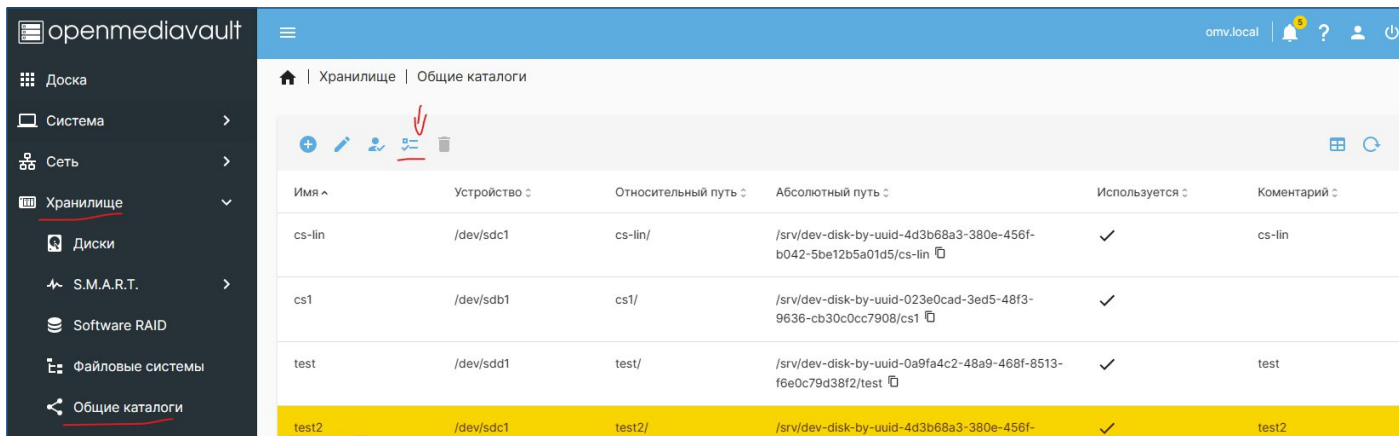
### 3.5.3.9 Перезапускаем сервисы

Выполните команду:

```
systemctl restart monoserve r7-office*
```

### 3.5.3.10 Обновить права на каталоги

Зайти, по пути ниже, в «**Общие каталоги**» и, выбрав нужный каталог, нажать **Access control list**.

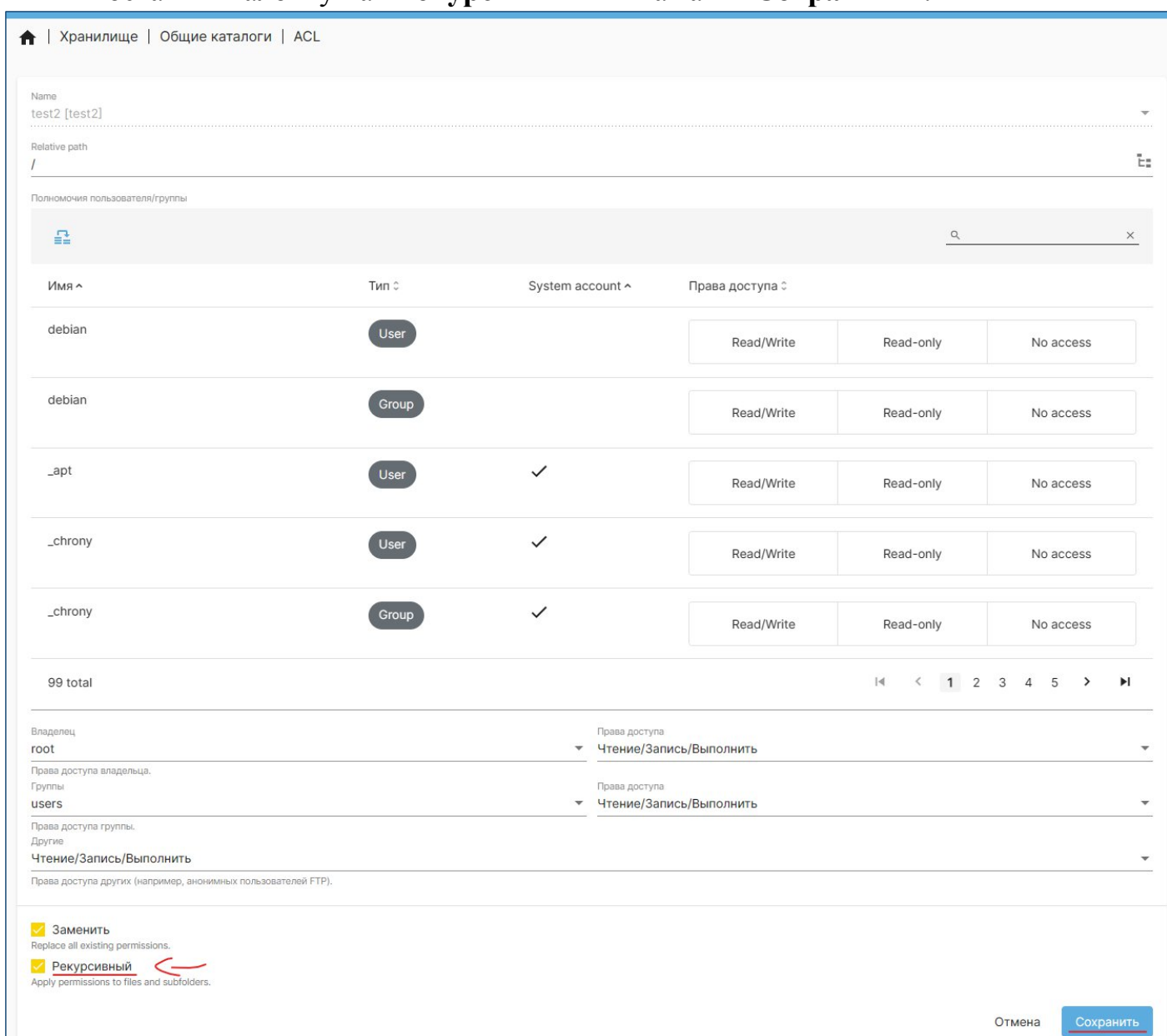


openmediavault | omv.local

Хранилище | Общие каталоги

Имя	Устройство	Относительный путь	Абсолютный путь	Используется	Комментарий
cs-lin	/dev/sdc1	cs-lin/	/srv/dev-disk-by-uuid-4d3b68a3-380e-456f-b042-5be12b5a01d5/cs-lin	✓	cs-lin
cs1	/dev/sdb1	cs1/	/srv/dev-disk-by-uuid-023e0cad-3ed5-48f3-9636-cb30c0cc7908/cs1	✓	
test	/dev/sdd1	test/	/srv/dev-disk-by-uuid-0a9fa4c2-48a9-468f-8513-f6e0c79d38f2/test	✓	test
test2	/dev/sdc1	test2/	/srv/dev-disk-by-uuid-4d3b68a3-380e-456f-	✓	test2

Поставить галочку на «Рекурсивный» и нажать «Сохранить».



Хранилище | Общие каталоги | ACL

Name: test2 [test2]

Relative path: /

Полномочия пользователя/группы

Имя	Тип	System account	Права доступа
debian	User		Read/Write   Read-only   No access
debian	Group		Read/Write   Read-only   No access
._apt	User	✓	Read/Write   Read-only   No access
._chrony	User	✓	Read/Write   Read-only   No access
._chrony	Group	✓	Read/Write   Read-only   No access

99 total

Владелец: root | Права доступа: Чтение/Запись/Выполнить

Права доступа владельца: Чтение/Запись/Выполнить

Группы: users | Права доступа: Чтение/Запись/Выполнить

Права доступа группы: Чтение/Запись/Выполнить

Другие: Чтение/Запись/Выполнить

Права доступа других (например, анонимных пользователей FTP):

**Заменить**  
Replace all existing permissions.

**Рекурсивный** ←  
Apply permissions to files and subfolders.

Отмена | Сохранить

## 4 Проверка работы сервера

Если Вы идёте по варианту с клоном VM, необходимо проверить работу основного сервера и перейти к пункту 5. Если Вы настраивали 2 сервера параллельно, то необходимо проверить работу двух серверов параллельно, подключившись в веб-интерфейс к каждому напрямую.

## 5 Создание клона сервера

Необходимо сделать клон VM доступными Вам средствами. Например, через снапшот сделать диск. Из бэкапа создать вторую машину. Сделать образ диска и из него создать VM. Самое главное, чтобы клон был сделан после всех настроек на основной VM с приложением. Также проверьте, добавлен ли **ip** адрес новой VM в конфигурационные файлы **nginx** на **reverse proxy**.

## 6. Проверка hostname

Необходимо проверить указан ли один и тот же **hostname** на двух серверах. Если нет, то выполнить следующие действия.

### 6.1 Изменить hostname

Изменить имя в файле

```
mcedit /etc/hostname
```

Выполнить команды:

```
hostnamectl set-hostname kh-cs-01  
systemctl restart systemd-hostnamed
```

Где,

```
kh-cs-01 – имя основного сервера приложения.
```

### 6.2 Перезагрузить VM

Перезагрузить VM, на которой был изменён **hostname**.

```
reboot
```

## 7. Проверка работы холодного резерва