RuBackup

Система резервного копирования
и восстановления данных

РАЗВЁРТЫВАНИЕ

ВЕРСИЯ 2.7.0.0.0, 06.10.2025

Содержание

Глава 1. Назначение руководства

Настоящее руководство предназначено для распределённой установки основных компонентов СРК:

- Глава 7;
- Глава 8;
  - основной сервер;
  - резервный сервер;
  - медиасервер;
- Глава 9.

Глава 2. Преимущества распределённой установки

Распределённая установка:

- рекомендована для средних и больших сред;
- делает систему эффективной;
- обеспечивает централизованное хранение и защиту данных;
- оптимально использует ресурсы организации: вычислительные мощности, сеть и хранилище;
- обеспечивает отказоустойчивость;
- обеспечивает масштабируемость по мере роста инфраструктуры.

Глава 3. Этапы распределённой установки

Очерёдность действий по развёртыванию СРК приведена в разделе Глава 4

Глава 4. Порядок развёртывания

Подготовка инфраструктуры СРК

1. Обеспечьте сетевое взаимодействие между узлами компонентов СРК.
2. Обеспечьте взаимодействие компонентов СРК путем открытия соответствующих сетевых портов для входящего и исходящего трафика между серверами, на которых будут установлены компоненты СРК.

Установка и настройка служебной базы данных

3. Разверните СУБД PostgreSQL и настройте подключения к БД серверной группировки СРК (основной, резервный, медиа- сервера) и АРМ администратора:

   - Раздел 7.2;
   - Раздел 7.3.

   Служебная база данных может быть установлена на узле основного сервера или любом другом доступном по сети узле, удовлетворяющем системным требованиям.

Подготовка к установке

4. Подготовьте пакеты компонентов СРК:
   - получите Глава 5;
   - скопируйте их на узлы, на которых будут развёрнуты компоненты СРК.
5. Подготовьте узлы к установке компонентов СРК:
   - Раздел 8.2.1 серверной части;
   - Раздел 9.1.2.1 клиентской части под управлением Linux-систем;
   - Раздел 9.2.2.1 клиентской части под управлением Windows-систем.

Установка

6. Установите пакеты компонентов RuBackup на подготовленных узлах:
   - Раздел 8.2.2 серверной части;
   - Раздел 9.1.2.2 клиентской части под управлением Linux-систем;
   - Раздел 9.2.2.2 клиентской части под управлением Windows-систем.

Лицензирование

7. Подготовьте лицензионные файлы для авторизации основного, резервного, медиасерверов, предварительно получив их у поставщика, и произведите Раздел 8.2.3.

Настройка

8. Выполните настройку установленных компонентов СРК в строго определенном порядке:
   a. на серверах (основном, резервном, медиасерверах):
      - Раздел 8.3.1;
   b. на всех клиентах РК:
      - Раздел 9.1.3.1 под управлением Linux-систем;
      - Раздел 9.2.3.1 под управлением Windows-систем.
9. Выполните настройки:
   a. для пользователей, которые будут взаимодействовать с компонентами СРК:
      - Раздел 8.3.2 для серверной части;
      - Раздел 9.1.3.2 для клиентской части под управлением Linux-системы;
   b. для узла:
      - Раздел 9.2.3.2 под управлением Windows-систем.
10. Добавьте сервисы СРК в автозагрузку:
    - Раздел 8.3.3 серверной части;
    - Раздел 9.1.3.3 клиентской части под управлением Linux-системы;
    - Раздел 9.2.3.3 клиентской части под управлением Windows-систем.
11. Произведите запуск развёрнутых компонентов СРК:
    - Раздел 8.4 серверной части;
    - Раздел 9.1.4 клиентской части под управлением Linux-системы;
    - Раздел 9.2.4 клиентской части под управлением Windows-систем.

Управление СРК

12. Для управления СРК используйте одно из приложений:
    a. для централизованного управления:
       - Менеджер администратора RuBackup (RBM);
       - Tucana;

- Утилиты командной строки;

b. для локального управления:

- Менеджер клиента RuBackup (RBC);
- Утилиты командной строки.

Глава 5. Дистрибутивы

Для развёртывания компонентов СРК RuBackup получите актуальные установочные deb/rpm пакеты одним из способов:

- на компакт-диске, полученном от поставщика;
- из дополнительно подключаемого, публичного репозитория;
- скачав актуальные установочные пакеты СРК RuBackup из облачного диска Астры на официальном сайте компании https://www.rubackup.ru/go/.

5.1. Компакт-диск

Пакеты для развёртывания СРК RuBackup могут быть получены от поставщика на компакт-диске, который может быть прочитан большинством приводов CD-ROM.

5.2. Публичный репозиторий

Пакеты для развёртывания СРК RuBackup могут быть установлены из дополнительного публичного репозитория.

Публичные репозитории доступны для операционных систем:

- Astra Linux 1.8;
- Astra Linux 1.7;
- Astra Linux 1.6;
- Debian 12;
- Debian 10;
- Ubuntu 22.04;
- Ubuntu 20.04;
- Ubuntu 18.04;
- CentOS 7;
- CentOS 8;
- РЕД ОС 7.3;
- РЕД ОС 8;
- Red Hat Enterprise Linux 9;
- ROSA Fresh Desktop 12;
- ROSA Enterprise Linux Server 7.3;
- ROSA Enterprise Linux Server 7.9.

5.2.1. Подключение публичного репозитория DEB-систем

1. Создайте файл с информацией о репозиториях:

```
cat <<EOF | sudo tee /etc/apt/sources.list.d/rubackup_deb.list
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public-testing
EOF
```

   где: `<OS-VERSION>` — версия используемой ОС:

   - astra_1.6;
   - astra_1.7;
   - astra_1.8;
   - debian_10;
   - debian_12;
   - ubuntu_18.04;
   - ubuntu_20.04;
   - ubuntu_22.04.

2. Добавьте ключ репозитория:

```
sudo wget -qO-
https://dl.astralinux.ru/artifactory/api/security/keypair/gc-astra-
official-repo-key/public | gpg --no-default-keyring --keyring gnupg-
ring:/etc/apt/trusted.gpg.d/rubackup-deb.gpg --import - && sudo chmod 644
/etc/apt/trusted.gpg.d/rubackup-deb.gpg
```

3. Обновите список пакетов:

```
sudo apt-get update
```

5.2.2. Подключение публичного репозитория RPM-систем

1. Создайте файл с информацией о репозиториях:
   a. для ОС:
      - CentOS 7;

- CentOS 8;
- РЕД ОС 7.3;
- РЕД ОС 8;
- Red Hat Enterprise Linux 9;
- ROSA Fresh Desktop 12;
- ROSA Enterprise Linux Server 7.9.

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/repodata/repomd.xml.key
gpgcheck=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/repodata/repomd.xml.key
gpgcheck=0
EOF
```

где: `<OS-VERSION>` — версия используемой ОС:

- centos_7;
- centos_8;
- redos_7.3;
- redos_8;
- rhel_9;
- rosa_12;
- rosa_7.9.

b. для ОС ROSA Enterprise Linux Server 7.3:

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/repodata/repomd.xml.key
gpgcheck=0
sslverify=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/repodata/repomd.xml.key
gpgcheck=0
sslverify=0
EOF
```

5.3. Облачный диск Астры

Пакеты для развёртывания СРК RuBackup могут быть скачаны из облачного диска Астры на официальном сайте компании https://www.rubackup.ru/go/.

На диске вы найдёте:

- папки с названиями операционных систем, содержащие совместимые с указанной ОС установочные пакеты для развёртывания компонентов СРК RuBackup (Alt Linux 10, Astra Linux 1.6 и т.д.);
- папку `Experimental`, содержащую:
  - совместимые с указанной ОС экспериментальные установочные пакеты для развёртывания компонентов СРК RuBackup, прошедшие только дизайн-тестирование;
  - папку `Scripts`, содержащую экспериментальные скрипты:
    - `script_block_device_metadata.sh` скрипт резервного копирования метаданных дедуплицированного пула;
    - `upgrade_rubackup_packages.sh` скрипт автоматического обновления;

- папку `Prev_Version`, содержащую установочный пакет модуля резервного копирования и восстановления данных кластеров СУБД PostgreSQL для поддержки нового функционала серверной и клиентской группировок релиза 2.1;
- `RB_key.iso` — специализированный загрузочный образ RuBackup.

Глава 6. Сетевые порты

Безопасное соединение компонентов СРК RuBackup и обмен информацией между ними подразумевает техническую возможность коммуникации по сети. Перед установкой продукта необходимо обеспечить взаимодействие компонентов СРК путем открытия соответствующих портов для входящего и исходящего трафика между серверами, на которых установлены компоненты СРК.

В таблице представлены компоненты СРК RuBackup, которые принимают входящие соединения по указанным портам и протоколам.

Таблица 1. Сетевые порты

Компонент		Целевой сервис	Протокол	Порт	Описание
от	до				
Основной сервер	Медиасервер	rubackup-cmd	TCP	9991	Управление операциями на медиасервере
		rubackup-media	TCP	9993	Управление операциями с данными
Основной сервер	База данных RuBackup на отдельностоящей машине	postgresql	TCP	5432 [1]	Сохранение конфигурационной и оперативной информации
Резервный сервер [2]	Основной сервер	rubackup-cmd	TCP	9991	Обеспечение отказоустойчивости
		rubackup-media	TCP	9993	Передача данных между медиасерверами в составе основного и резервного серверов
Резервный сервер [2]	База данных RuBackup на отдельностоящей машине	postgresql	TCP	5432	Сохранение конфигурационной и оперативной информации
Медиасервер	Медиасервер	rubackup-media	TCP	9993	Передача данных между медиасерверами

Медиасервер	Резервный сервер [2]	rubackup-cmd	TCP	9991	Управление операциями на медиасервере
		rubackup-media	TCP	9993	Управление операциями с данными
Медиасервер	База данных RuBackup на отдельностоящей машине	postgresql	TCP	5432 [1]	Сохранение конфигурационной и оперативной информации
Клиент резервного копирования	Основной сервер	rubackup-cmd	TCP	9991	Управление операциями на клиенте резервного копирования
Клиент резервного копирования	Медиасервер	rubackup-media	TCP	9993	Передача данных между медиасервером и клиентом
Клиент резервного копирования	Резервный сервер [2]	rubackup-cmd	TCP	9991	Управление операциями на клиенте резервного копирования
		rubackup-media	TCP	9993	Передача данных между медиасервером и клиентом
RuBackup REST API	Основной сервер	rubackup-rbm	TCP	9995	Отправка запросов на сервер и получение информации
RuBackup REST AP	База данных RuBackup на отдельностоящей машине	postgresql	TCP	5432 [1]	Получение информации из базы данных
RuBackup REST API	Резервный сервер [2]	rubackup-rbm	TCP	9995	Отправка запросов на сервер и получение информации

Менеджер RuBackup (RBM) на отдельно стоящей машине	База данных RuBackup на отдельностоящей машине	postgresql	TCP	5432 [1]	Сохранение конфигурационной и оперативной информации
Менеджер RuBackup (RBM) на отдельно стоящей машине	Основной сервер	rubackup-rbm	TCP	9995	Управление операциями RuBackup
Менеджер RuBackup (RBM) на отдельно стоящей машине	Резервный сервер [2]	rubackup-rbm	TCP	9995	Управление операциями RuBackup
Клиент, посылающий запрос через RuBackup REST API	Основной сервер	rubackup-api	HTTPS	443 [3]	Управление операциями RuBackup через REST API
Клиент, посылающий запрос через RuBackup REST API	Резервный сервер [2]	rubackup-api	HTTPS	443 [3]	Управление операциями RuBackup через REST API

[1] Если база данных настроена с использованием нестандартного порта, то для подключения к ней продукта RuBackup порт может быть изменен вручную в конфигурационном файле /opt/rubackup/etc/config.file.

[2] При наличии резервного сервера.

[3] Порт для подключения, при необходимости, может быть изменен через переменные окружения в файле /opt/rubackup/etc/rubackup_api.env (см. в «Руководстве по установке и взаимодействию с программным интерфейсом RuBackup REST API»)

Глава 7. Служебная база данных

Назначение

СУБД PostgreSQL используется для хранения:

- метаданных резервных копий;
- конфигурационных параметров СРК RuBackup.

Место установки

Служебная база данных может быть установлена на узле основного сервера или любом другом доступном по сети узле, удовлетворяющем системным требованиям.

7.1. Системные требования

Таблица 2. Аппаратные требования к серверу БД RuBackup

Аппаратный компонент	Значение
Процессор	4 ядра
Оперативная память	64 ГБ
Дисковое пространство	3,84 ТБ

Для обеспечения максимального уровня отказоустойчивости и быстродействия при промышленной эксплуатации, рекомендуется использовать в качестве конфигурационной базы RuBackup СУБД PostgreSQL в отказоустойчивой конфигурации с использованием решения Patroni, развернутом на отдельно стоящих машинах, с совокупным объемом дискового пространства 3.84 ТБ, построенного с использованием твердотельных накопителей, подключенных через шину PCI Express (NVMe SSD).

7.2. Установка СУБД

1. Установите из репозитория [1] последнюю доступную версию СУБД PostgreSQL:

Astra Linux, Debian, Ubuntu

```
sudo apt install postgresql
```

Альт

```
sudo apt-get install postgresql-server
```

Rosa Cobalt, RHEL

```
sudo yum install postgresql
```

RedOS, CentOS, Rosa Chrome

```
sudo dnf install postgresql-server
```

2. Выполните установку последней доступной версии пакета `postgresql-contrib`:

Для Astra Linux SE 1.6 необходимо установить пакет `postgresql-contrib-9.6`.

Astra Linux, Debian, Ubuntu

```
sudo apt install postgresql-contrib
```

Альт

```
sudo apt-get install postgresql-contrib
```

Rosa Cobalt, RHEL

```
sudo yum install postgresql-contrib
```

RedOS, CentOS, Rosa Chrome

```
sudo dnf install postgresql-contrib
```

3. Произведите инициализацию БД только для указанных ОС:

Альт

```
sudo /etc/init.d/postgresql initdb
```

Rosa Cobalt, RHEL

```
/usr/pgsql-12/bin/postgresql-12-setup initdb
```

RedOS, CentOS, Rosa Chrome

```
sudo postgresql-setup --initdb
```

4. Запустите PostgreSQL:

```
sudo service postgresql start
```

5. Добавьте запуск PostgreSQL в автозагрузку:

```
sudo systemctl enable postgresql
```

7.3. Настройка СУБД

7.3.1. Настройка файла `pg_hba.conf`

Настройте возможность подключения к СУБД для всех серверов, которые будут входить в серверную группировку RuBackup (основной, резервный, медиа- сервера), и АРМ администратора RuBackup, для этого:

1. Перейдите в папку, где находится файл `pg_hba.conf`.
2. Откройте для редактирования конфигурационный файл `pg_hba.conf`:

```
sudo nano pg_hba.conf
```

3. Отредактируйте, открывшийся файл, указав ip-адреса и маску сети всех подключаемых серверов и АРМ администратора RuBackup к БД по протоколу IPv4, например:

Пример 1. Пример файла `pg_hba.conf`

```
local all postgres peer

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only

local all all md5

# IPv4 local connections:

host all all 127.0.0.1/32    md5

host all all 192.168.0.50/32 md5

host all all 192.168.0.51/32 md5

host all all 192.168.0.52/32 md5

host all all 192.168.0.53/32 md5
```

4. Сохраните изменения.

Добавить IP-адреса подключаемых к БД серверов можно и после установки сервера RuBackup, отредактировав конфигурационный файл `pg_hba.conf` и перезапустив PostgreSQL.

7.3.2. Настройка файла `postgresql.conf`

Настройте прослушивание подключений к БД для всех серверов, которые будут входить в серверную группировку RuBackup (основной сервер, резервный сервер, медиасервер) с целью последующего удалённого подключения к БД:

1. Перейдите в папку, где находится файл `postgresql.conf`.
2. Откройте для редактирования конфигурационный файл `postgresql.conf`:

```
sudo nano postgresql.conf
```

3. Отредактируйте открывшийся файл:
   a. в секции `CONNECTIONS AND AUTHENTICATION` добавьте выделенную строку `listen_addresses = 'localhost'`:

```
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------
----
# - Connection Settings -

#listen_addresses = 'localhost'
# what IP address(es) to listen on;

listen_addresses = '*'
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)

port = 5432
# (change requires restart)

max_connections = 100
# (change requires restart)
```

   b. для расчёта максимального количества подключений `max_connections` к БД следует дополнительно учитывать: при подключении каждого медиасервера добавляется `parallelizm_media` `×2+9` соединений;

c. при необходимости отредактируйте значение параметра `shared_buffers`. Рекомендуемое значение параметра ~50 % от размера оперативной памяти;

d. при необходимости отредактируйте значение параметра `max_parallel_workers`. Рекомендуемое значение параметра не менее 50 % от количества процессорных ядер, если сервер СУБД совмещен с сервером RuBackup и 100 %, если сервер СУБД является выделенным.

4. Сохраните изменения.

7.3.3. Настройка файла `mswitch.conf`

Данный шаг выполняется только для СУБД PostgreSQL в ОС Astra Linux Special Edition с максимальным уровнем защищенности («Смоленск»).

Чтобы не возникала ошибка при получении мандатных атрибутов, нужно отредактировать конфигурационный файл СУБД PostgreSQL `/etc/parsec/mswitch.conf` в ОС Astra Linux Special Edition с максимальным уровнем защищенности («Смоленск»):

1. Откройте для редактирования файл `/etc/parsec/mswitch.conf` и измените параметр для создания пользователя СУБД PostgreSQL, который не назначен в ОС Astra Linux Special Edition 1.7:

```
sudo nano /etc/parsec/mswitch.conf
```

2. Отредактируйте значение указанного параметра, изменив его на `yes`:

```
zero_if_notfound: yes
```

3. Сохраните изменения.

7.3.4. Применение изменений

Перезапустите Postgres для применения изменений:

```
sudo service postgresql restart
```

7.3.5. Установка пароля пользователя `postgres`

1. Проверьте подключение к СУБД, выполнив вход под пользователем `postgres`, введя команду:

```
sudo -u postgres psql
```

2. Задайте пароль для пользователя `postgres`, подключившись к БД:

```
alter user postgres password '12345';
```

где `'12345'` — задаваемый пароль пользователя.

3. Завершите работу под пользователем `postgres`:

```
\q
```

7.4. Настройка SSL соединений

Для повышения безопасности сервера базы данных возможно использование надежного шифрования соединений с базой данных.

Для настройки SSL соединений:

1. Создайте сертификаты для сервера PostgreSQL и его клиентов (postgres-клиентов) (см. Раздел 7.4.1).
2. Выполните настройку конфигурационных файлов на сервере PostgreSQL (см. Раздел 7.4.2).
3. После установки пакетов компонентов СРК скопируйте полученные сертификаты и выполните настройку SSL соединений для postgres-клиентов на узлах (см. Раздел 7.4.3):
   - развёрнутой серверной части СРК;
   - использующих приложение «Менеджер администратора RuBackup» или «Веб-интерфейс Tucana».

7.4.1. Выпуск сертификатов

Аутентификация клиента по сертификату позволяет серверу проверить личность подключающегося, подтверждая, что сертификат X.509, представленный postgres-клиентом, подписан доверенным центром сертификации (CA).

Сертификаты SSL проверяются и выдаются Центром сертификации.

Если вы не имеете PKI инфраструктуры открытых ключей, то на отдельном хосте, который может выполнять роль Центра сертификации:

1. Создайте директории, в которые будут сгенерированы сертификаты Центра

сертификации, сервера PostgreSQL и для всех postgres-клиентов (в зависимости от архитектуры вашей СРК):

```
mkdir certs && cd certs && mkdir ca pg-server rb-server rb-media rb-rbm
```

где:

- `ca` - директория для сертификатов Центра сертификации;
- `pg-server` - директория для сертификатов сервера PostgreSQL;
- `rb-server` - директория для сертификатов основного сервера RuBackup;
- `rb-media` - директория для сертификатов медиасервера;
- `rb-rbm` - директория для сертификатов АРМ администратора, если Менеджер администратора RuBackup (RBM) развёрнут на отдельном хосте.

2. Создайте закрытый ключ Центра сертификации, для этого:
   - Перейдите в ранее созданную папку:

     ```
     cd ./ca
     ```

   - Сгенерируйте закрытый ключ для CA (`ca.key`), выполнив команду, например:

     ```
     openssl genrsa -out ca.key 2048
     ```

   - Создайте самоподписанный сертификат Центра сертификации (`ca.crt`) сроком действия 1 год:

     ```
     openssl req -new -x509 -days 365 -key ca.key -out ca.crt
     ```

     где `CN` — это полное имя хоста (FQDN), на котором развёрнут CA.

3. Выпустите сертификат и закрытый ключ для сервера PostgreSQL, для этого:
   - Перейдите в ранее созданную папку:

     ```
     cd ./pg-server
     ```

   - Сгенерируйте закрытый ключ для сервера PostgreSQL `/pg-server/server.key`:

```
openssl genrsa -out server.key 2048
```

- Сгенерируйте запрос на сертификат сервера PostgreSQL `/pg-server/server.csr`:

  ```
  openssl req -new -key server.key -out server.csr
  ```

  где `CN` — это полное имя хоста (FQDN), на котором развёрнут сервер PostgreSQL.

- Подпишите запрос на сертификат сервера PostgreSQL закрытым ключом Центра сертификации:

  ```
  openssl x509 -req -in server.csr -CA ../ca/ca.crt -CAkey ../ca/ca.key
  -CAcreateserial -out server.crt -days 365
  ```

4. Повторите шаг 3 для каждого postgres-клиента, сгенерировав закрытый ключ (`postgresql.key`) и выпустив сертификат (`postgresql.crt`) для всех postgres-клиентов, указав в сертификате соответствующее FQDN хоста, на котором развёрнут компонент CPK.

7.4.2. Настройка SSL соединения на сервере PostgreSQL

Выполните приведённые ниже настройки, чтобы сервер PostgreSQL прослушивал как обычные, так и SSL соединения через один и тот же TCP-порт и согласовывал использование SSL с любым подключающимся postgres-клиентом.

1. Скопируйте в папку `/etc/postgresql/16/main` на сервер PostgreSQL из папки `/pg-server` Центра сертификации подготовленные:
   - сертификат Центра сертификации (`ca.crt`);
   - подписанный сертификат сервера PostgreSQL (`server.crt`);
   - сгенерированный закрытый ключ сервера PostgreSQL (`server.key`).
2. Для файлов сертификата и закрытого ключа установите полный доступ на чтение и запись только для владельцев:

   ```
   chmod 600 server.crt server.key ca.crt
   ```

   Сделайте владельцем файлов пользователя и группу пользователя `postgres`:

```
chown postgres:postgres server.crt server.key ca.crt
```

3. Отредактируйте конфигурационный файл `postgresql.conf`:
   - включите поддержку зашифрованных соединений:

     ```
     ssl = on
     ```

   - укажите путь к файлу сертификата Центра сертификации (или цепочке сертификатов):

     ```
     ssl_ca_file = '/etc/postgresql/16/main/ca.crt'
     ```

     Сертификат CA проверяет, что сертификат postgres-клиента подписан доверенным центром сертификации.

   - укажите путь к файлу сертификата сервера PostgreSQL:

     ```
     ssl_cert_file = '/etc/postgresql/16/main/server.crt'
     ```

     Сертификат будет отправлен postgres-клиенту для указания подлинности сервера PostgreSQL.

   - укажите путь к файлу закрытого ключа сервера PostgreSQL:

     ```
     ssl_key_file = '/etc/postgresql/16/main/server.key'
     ```

     Закрытый ключ доказывает, что сертификат сервера PostgreSQL был отправлен владельцем; не указывает, что владелец сертификата заслуживает доверия.

4. Чтобы потребовать от postgres-клиента предоставления доверенного сертификата, отредактируйте конфигурационный файл `pg_hba.conf`:
   - добавьте опцию аутентификации `clientcert=verify-ca` или `clientcert=verify-full` в соответствующие `hostssl` строки, где:
     - `clientcert=verify-full` сервер PostgreSQL не только проверяет цепочку сертификатов, но также проверяет, совпадает ли имя пользователя или его сопоставление с `CN` предоставленного сертификата;
     - `clientcert=verify-ca` сервер проверяет, что сертификат postgres-клиента подписан одним из доверенных центров сертификации.

Также желательно закомментировать все строчки `host`, например:

```
#host   all all 0.0.0.0/0  md5
hostssl all all 0.0.0.0/0 [md5,cert]
clientcert=[verify-ca,verify-full] (1)
```

(1) В старых версиях [0,1]

где:

`md5` — запросить пароль пользователя,

`cert` — аутентификация по сертификату.

Если параметр `clientcert` не указан, сервер проверяет сертификат postgres-клиента по своему файлу CA, только если сертификат postgres-клиента представлен и CA настроен.

5. Произведите настройку карты имён пользователей.

   При использовании внешней системы аутентификации, такой как `Ident`, имя пользователя операционной системы, инициировавшего подключение, может не совпадать с именем пользователя базы данных (роли), который должен использоваться. В этом случае карта имен пользователей может быть применена для сопоставления имени пользователя операционной системы с именем пользователя базы данных

   Чтобы использовать сопоставление имен пользователей, отредактируйте:

   - конфигурационный файл `pg_hba.conf` — укажите в значении параметра `map=map-name`:

     ```
     hostssl all all 0.0.0.0/0 md5 clientcert=verify-full map=sslmap
     ```

   - конфигурационный файл `pg_ident.conf`, хранящийся в каталоге данных кластера — настройте карты имен пользователей, добавьте, например:

     ```
     # MAPNAME SYSTEM-USERNAME PG-USERNAME
     sslmap    postgres        postgres
     sslmap    postgres        rubackup
     ```

     где:

     - в столбце `SYSTEM-USERNAME` укажите `CN` сертификата postgres-клиента;

- в столбце `PG-USERNAME` укажите имя пользователя, с которым нужно сопоставить.

6. Для применения изменений перезапустите сервер:

```
sudo systemctl restart postgresql
```

7.4.3. Настройка SSL соединения на узлах компонентов RuBackup

Для подключения серверных компонентов RuBackup и АРМ администратора СРК (использующего приложение «Менеджер администратора RuBackup») к служебной базе данных PostgreSQL с использованием защищённого соединения выполните приведённые ниже настройки на соответствующих узлах (postgres-клиентах):

- развёрнутой серверной части СРК;
- использующих приложение «Менеджер администратора RuBackup».

Настройка SSL соединения на узлах компонентов RuBackup

1. Перенесите из соответствующей postgres-клиенту папки на узле Центра сертификации подготовленные:
   - сертификат Центра сертификации (ca.crt), чтобы postgres-клиент мог проверить, что конечный сертификат сервера PostgreSQL был подписан его доверенным корневым сертификатом;
   - сертификат postgres-клиента (узла компонента СРК) (postgresql.crt);
   - сгенерированный закрытый ключ сервера/клиента СРК (postgresql.key).
2. Для файлов сертификата и закрытого ключа установите полный доступ на чтение и запись только для владельцев:

```
chmod 600 server.crt server.key ca.crt
```

3. Сделайте владельцем файлов пользователя, от имени которого будет запущен компонент СРК (postgres-клиент):

```
chown suser:suser server.crt server.key ca.crt
```

4. Настройка SSL соединения на узле компонента RuBackup выполняется **после установки пакетов СРК** одним из способов:
   - при настройке компонента СРК серверной или клиентской части;
   - при внесении правки в файл настроек сервера (полученный после конфигу-

рирования компонента СРК).

5. Для настройки SSL-соединения с БД предварительно необходимо выполнить настройку служебной базы данных в соответствии с разделом Раздел 7.3 и подготовить сертификаты.

   a. `Enter sslmode ( allow, disable, prefer, require, verify-ca, verify-full ) [ require ]`

   `Enter path for sslrootcert file:`

   `Enter path for sslcert file:`

   `Enter path for sslkey file:`

   - выберите и введите название выбранного режима SSL в соответствии с таблицей.

     По умолчанию выбран режим `require`.

Таблица 3. Описание режимов SSL

sslmode	Защита от прослушивания	Защита от MITM	Утверждение
disable	Нет	Нет	Мне не важна безопасность и я не приемлю издержки, связанные с шифрованием.
allow	Возможно	Нет	Мне не важна безопасность, но я приемлю издержки, связанные с шифрованием, если на этом настаивает сервер.
prefer	Возможно	Нет	Мне не важна безопасность, но я предпочитаю шифрование (и приемлю связанные издержки), если это поддерживает сервер.
require	Да	Нет	Я хочу, чтобы мои данные шифровались, и я приемлю сопутствующие издержки. Я доверяю сети в том, что она обеспечивает подключение к нужному серверу
verify-ca	Да	Зависит от политики ЦС	Я хочу, чтобы мои данные шифровались, и я приемлю сопутствующие издержки. Мне нужна уверенность в том, что я подключаюсь к доверенному серверу
verify-full	Да	Да	Я хочу, чтобы мои данные шифровались, и я приемлю сопутствующие издержки. Мне нужна уверенность в том, что я подключаюсь к доверенному серверу и это именно указанный мной сервер

- укажите расположение подготовленных сертификатов:
  - в поле `sslrootcert` укажите расположение сертификата центра сертификации;
  - в поле `sslcert` укажите расположение сертификата настраиваемого хоста;
  - в поле `sslkey` укажите расположение закрытого ключа настраиваемого

хоста.

7.5. Настройка балансировщика нагрузки

При наличии прокси-сервера HAProxy, принимающего запросы к служебной базе данных СРК RuBackup, рекомендуется выполнить следующие действия:

1. В файле `haproxy.cfg` задайте одинаковое значение для параметров `timeout client` и `timeout server`. Рекомендуемое значение 48h или более.

   Согласно официальной документации [2] значения параметров `timeout client` и `timeout server` должны быть идентичные.

2. Убедитесь, что в настройках служебной СУБД PostgreSQL отсутствуют тайм-ауты, а если присутствуют, то выставить такие же значения как и в настройках HAProxy (см. пункт 1).
3. Добавьте в файл `haproxy.cfg` в строку с проверкой узла PostgreSQL параметр `shutdown-sessions`, например:

   ```
   "server primary 192.168.122.60:3306 check on-marked-down shutdown-
   sessions".
   ```

4. Завершите все активные задачи в СРК RuBackup.
5. Остановите сервис сервера СРК RuBackup, выполнив в терминале на узле сервера СРК RuBackup:

   ```
   sudo systemctl stop rubackup_server
   ```

6. Перезапустите СУБД PostgreSQL, выполнив:

   ```
   sudo systemctl restart postgresql
   ```

7. Запустите сервис сервера СРК RuBackup, выполнив в терминале на узле сервера СРК RuBackup:

   ```
   sudo systemctl start rubackup_server
   ```

[1] Для некоторых ОС возможно потребуется подключить дополнительный репозиторий

[2] https://docs.haproxy.org/2.6/configuration.html

Глава 8. Серверная часть

Серверная часть СРК RuBackup может состоять из:

- обязательного компонента — основного сервера;
- одного или нескольких необязательных компонентов — резервного сервера и медиасервера.

Основной сервер

Основной сервер — это главный управляющий сервер, обеспечивающий взаимодействие компонентов СРК.

Основной сервер выполняет функцию медиасервера в случае установки способом «Всё в одном» (все компоненты СРК RuBackup развёрнуты на одном узле).

Резервный сервер

Резервный сервер, в случае отказа основного сервера, поддерживает функционал основного сервера RuBackup, а клиенты системы резервного копирования автоматически подключатся к резервному серверу. После восстановления функционирования основного сервера клиенты подключатся обратно к основному серверу.

Медиасервер

Медиасервер (это узел, на котором подключено устройство хранения) – ёмкое дисковое устройство или библиотека магнитных лент.

Медиасервер наполняет устройство хранения поступающими резервными копиями данных и управляет ими по требованию основного сервера.

Каждый медиасервер ассоциирован с пулом, который содержит логические устройства одного типа — хранилища.

8.1. Системные требования

В данном подразделе приведены системные требования для каждого серверного компонента СРК RuBackup, предъявляемые к техническим средствам, необходимым для нормального функционирования СРК RuBackup.

В случае установки на один хост нескольких компонентов СРК RuBackup (например, при способе установки «Всё в одном») следует консолидировать соответствующие аппаратные требования, предъявляемые к техническому средству, на которое производится установка.

8.1.1. Аппаратные требования

Основной/резервный сервер

Минимальные аппаратные требования, необходимые для стабильного функционирования сервера СРК RuBackup приведены в таблице.

Таблица 4. Аппаратные требования, предъявляемые к серверу RuBackup

Аппаратный компонент	Объем хранимых данных			Примечание
	48 ТБ	96 ТБ	144 ТБ	
Процессор	10 ядер, 20 потоков (2 потока на 1 ядро или более)			Рекомендуемые модели: Intel Xeon 4210, AMD EPYC 7000 или более современные
Оперативная память	128 ГБ	256 ГБ	256 ГБ	—
Твердотельный накопитель (SSD)	RAID 1, 2 диска по 480 ГБ каждый			Объём дискового пространства для установки операционной системы и компонентов RuBackup, за исключением конфигурационной базы данных RuBackup.
Твердотельный накопитель, подключенный через шину PCI Express (NVMe SSD)	3.84 ТБ			Рекомендуется в случае развертывания инстанса PostgreSQL для конфигурационной базы данных RuBackup на той же машине, где установлен сервер RuBackup. Диски NVMe SSD позволяют повысить производительность операций в фильтре Блума и скорость обработки данных при выполнении процессов дедупликации. 3.84 Тб предусматривают потенциальный рост объемов обрабатываемых данных. Для обеспечения максимального уровня отказоустойчивости и быстродействия при промышленной эксплуатации рекомендуется использовать в качестве конфигурационной базы RuBackup СУБД PostgreSQL в отказоустойчивой конфигурации, например, с использованием решения Patroni, развернутом на отдельностоящих машинах.
Жесткий диск (HDD) или флэш-накопитель (flash drive)	RAID 50, 12 дисков по 4 ТБ каждый	RAID 50, 12 дисков по 8 ТБ каждый	RAID 50, 12 дисков по 12 ТБ каждый	Рекомендуется в случае активного использования машины с основным сервером в качестве медиасервера, для возможности расширения дискового пространства под хранение резервных копий. В случае хранения данных на опосредованных СХД, данный компонент не используется.
Сеть	2 сетевых адаптера с пропускной способностью 10 Гб каждый, с 2 портами (dual port)			—

Медиасервер

Начиная с версии СРК 2.6.0.0.0 используется новый механизм сжатия небольших блоков данных на блочном пуле.

При обновлении СРК с версии 2.5.7.0.0 необходимо увеличить объем оперативной памяти медиасервера на 60%.

Рекомендуемая конфигурация медиасервера зависит от совокупного объема хранимых данных и схожа с конфигурацией сервера RuBackup. Для расчета конфигурации медиасервера воспользуйтесь таблицей.

Таблица 5. Аппаратные требования, предъявляемые к медиасерверу

Аппаратный компонент	Объем хранимых данных			Примечание
	48 ТБ	96 ТБ	144 ТБ	
Процессор	10 ядер, 20 потоков (2 потока на 1 ядро или более)			Рекомендуемые модели: Intel Xeon 4210, AMD EPYC 7000 или более современные
Оперативная память	128 ГБ	256 ГБ	256 ГБ	—
Твердотельный накопитель (SSD)	RAID 1, 2 диска по 480 ГБ каждый			Объём дискового пространства для установки операционной системы и компонентов RuBackup, за исключением конфигурационной базы данных RuBackup.
Жесткий диск (HDD) или флэш-накопитель (flash drive)	RAID 50, 12 дисков по 4 ТБ каждый	RAID 50, 12 дисков по 8 ТБ каждый	RAID 50, 12 дисков по 12 ТБ каждый	Для возможности расширения дискового пространства под хранение резервных копий. В случае хранения данных на опосредованных СХД, данный компонент не используется.
Сеть	2 сетевых адаптера с пропускной способностью 10 Гб каждый, с 2 портами (dual port)			—

8.1.2. Программные требования

Программные требования, необходимые для стабильного функционирования сервера СРК RuBackup:

- операционная система из совместимых с компонентами СРК RuBackup:
  - Astra 1.6;
  - Astra 1.7;
  - Astra 1.8;
  - CentOS 7;
  - CentOS 8;

  - Debian 10;
  - Debian 12;
  - RHEL 9;
  - RedOS 7.3;
  - RedOS 8;
  - Rosa Chrome 12;
  - Rosa Cobalt 7.3;
  - Rosa Cobalt 7.9;
  - Ubuntu 18.04;
  - Ubuntu 20.04;
  - Ubuntu 22.04;
  - Альт 10;
- открытые порты в сооветствии с таблицей Сетевые порты;
- зависимости пакетов для каждой совместимой ОС:

Таблица 6. Зависимости `rubackup-client`, `rubackup-server`, `rubackup-common`

Операционная система	Пакеты
Astra 1.6	exim4-base exim4-config exim4-daemon-light gnupg2 libcurl3 или libcurl4 libldap-2.4-2 libldap-common liblockfile-bin liblockfile1 libnghttp2-14 librtmp1 libsasl2-2 libssh2-1 mailutils или bsd-mailx openssl parsec-base parsec-cap parsec-mac psmisc wget xauth

Операционная система	Пакеты
Astra 1.7	exim4-base exim4-config exim4-daemon-light gnupg2 guile-2.2-libs libcurl3 или libcurl4 libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblzo2-2 libmailutils5 libmariadb3 libntlm0 libpugixml1v5 libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl parsec-base parsec-cap parsec-mac psmisc wget xauth

Операционная система	Пакеты
Astra 1.8	exim4-base exim4-config exim4-daemon-light gnupg2 gsasl-common guile-3.0-libs libcurl3 или libcurl4 libevent-2.1-7 libgc1 libgnutls-dane0 libgnutls30 libgsasl18 libgssglue1 libidn12 libldap-2.5-0 libltdl7 libmailutils9 libmariadb3 libncurses6 libncursesw6 libntlm0 libpq5 libpugixml1v5 libsasl2-2 libtinfo6 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common ncurses-base ncurses-bin ncurses-term openssl parsec-base parsec-cap parsec-mac psmisc wget xauth
CentOS 7	cyrus-sasl mailx openldap pugixml qt5-qtbase-gui
CentOS 8	cyrus-sasl mailx openldap pugixml qt5-qtbase-gui

Операционная система	Пакеты
Debian 10	exim4-base exim4-config exim4-daemon-light gnupg2 guile-2.2-libs libcurl3 или libcurl4 libcurl4 libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblzo2-2 libmailutils5 libmariadb3 libntlm0 libpugixml1v5 libpython2.7 libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl psmisc wget xauth

Операционная система	Пакеты
Debian 12	exim4-base exim4-config exim4-daemon-light gnupg2 gsasl-common guile-3.0-libs libcurl3 или libcurl4 libevent-2.1-7 libfribidi0 libgc1 libgnutls-dane0 libgnutls30 libgsasl18 libgssglue1 libidn12 libldap-2.5-0 libltdl7 libmailutils9 libmariadb3 libncurses6 libntlm0 libpq5 libpugixml1v5 libpython3.11 libpython3.11-minimal libpython3.11-stdlib libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl psmisc python3.11 python3.11-minimal wget xauth
RHEL 9	cyrus-sasl mailx openldap pugixml qt5-qtbase-gui s-nail
RedOS 7.3	cyrus-sasl mailx openldap pugixml qt5-qtbase-gui

Операционная система	Пакеты
RedOS 8	cyrus-sasl mailx openldap pugixml qt5-qtbase-gui
Rosa Chrome 12	cyrus-sasl lib64db5.2 lib64ldap2.4_2 lib64ltdl7 lib64mailutils9 lib64mu_auth9 lib64mu_dbm9 lib64mu_dotmail9 lib64mu_imap9 lib64mu_maildir9 lib64mu_mailer9 lib64mu_mbox9 lib64mu_pop9 lib64mu_sieve9 lib64muaux9 lib64pugixml1 lib64qt5gui5 lib64sasl2 mailutils mailutils-locales qt5-qtbase-gui

Операционная система	Пакеты
Rosa Cobalt 7.3	cups-libs cyrus-sasl fontconfig fontpackages-filesystem glx-utils libICE libSM libX11 libX11-common libXau libXdamage libXext libXfixes libXi libXrender libXxf86vm libicu libpng libxcb libxshmfence mailx mesa-libEGL mesa-libGL mesa-libgbm mesa-libglapi openldap qt5-qtbase qt5-qtbase-common qt5-qtbase-gui xcb-util xcb-util-image xcb-util-keysyms xcb-util-renderutil xcb-util-wm
Rosa Cobalt 7.9	cyrus-sasl libicu libxkbcommon-x11 mailx openldap qt5-qtbase-gui

Операционная система	Пакеты
Ubuntu 18.04	gnupg2 guile-2.0-libs libcurl3 или libcurl4 libgc1c2 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblzo2-2 libmailutils5 libmysqlclient20 libnghttp2-14 libntlm0 libpython2.7 libpython2.7-minimal libpython2.7-stdlib librtmp1 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth
Ubuntu 20.04	gnupg2 guile-2.2-libs libcurl3 или libcurl4 libgc1c2 libgsasl7 libidn11 libkyotocabinet16v5 libldap-2.4-2 libmailutils6 libmysqlclient21 libntlm0 libpugixml1v5 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth

Операционная система	Пакеты
Ubuntu 22.04	gnupg2 gsasl-common guile-3.0-libs libcurl3 или libcurl4 libfribidi0 libgc1 libgsasl7 libidn12 libldap-2.5-0 libltdl7 libmailutils8 libmysqlclient21 libntlm0 libpq5 libpugixml1v5 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth
Альт 10	libldap libsasl2-3 mailutils pugixml qt5-qtbase-gui xauth

Таблица 7. Зависимости `rubackup-common-gui`

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth
Astra 1.8	gnupg2 wget xauth
Debian 10	gnupg2 wget xauth

Операционная система	Пакеты
Debian 12	gnupg2 wget xauth
Rosa Chrome 12	qt5-qtbase-gui
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 wget xauth
Ubuntu 20.04	gnupg2 wget xauth
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	xauth

8.2. Установка

8.2.1. Подготовка к установке

Установка зависимостей пакетов

Данный шаг предназначен для установки локальных пакетов. Если вы устанавливаете пакеты из репозитория, то пропустите этот шаг.

Для успешного развёртывания сервера СРК RuBackup необходимо наличие установленных зависимостей пакетов в соответствии с таблицей, в зависимости от используемой операционной системы на узле развёртывания сервера RuBackup, для этого:

1. Проверьте наличие установленных пакетов зависимостей в ОС, например:

Astra Linux, Debian, Ubuntu

```
dpkg-query -l
```

Альт

```
apt list --installed
```

Rosa Cobalt, RHEL

```
yum list с опцией installed
```

RedOS, CentOS, Rosa Chrome

`dnf list installed`

2. Если вы используете операционную систему CentOS 7, CentOS 8 или RHEL 9, то добавьте репозиторий `EPEL` [1], поддерживаемый в рамках проекта Fedora и содержащий некоторые пакеты, которые не вошли в стандартный набор RHEL (CentOS):

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

Файл репозитория будет автоматически загружен в каталог `/etc/yum.repos.d/epel.repo` и активирован.

3. Если вы используете операционную систему CentOS 7 или CentOS 8, то также рекомендуется включить репозиторий `PowerTools`, поскольку пакеты `EPEL` могут зависеть от пакетов из него:

```
sudo dnf config-manager --set-enabled powertools
```

4. Если вы используете операционную систему RHEL 9, то также рекомендуется включить репозиторий `codeready-builder-for-rhel-8-*-` репозиторий `rpm`, поскольку пакеты `EPEL` могут зависеть от пакетов из него:

```
ARCH=$( /bin/arch )

sudo subscription-manager repos --enable "codeready-builder-for-rhel-8-$\{ARCH}-rpms"
```

5. Обновите репозитории пакетов в системе:

Astra Linux, Debian, Ubuntu

`sudo apt update`

Альт

`sudo apt-get update`

Rosa Cobalt, RHEL

`sudo yum update`

RedOS, CentOS, Rosa Chrome

```
sudo dnf update
```

6. Установите недостающие зависимости пакетов из таблицы:

Astra Linux, Debian, Ubuntu

```
sudo apt install <namepackage>
```

Альт

```
sudo apt-get install <namepackage>
```

Rosa Cobalt, RHEL

```
sudo yum install <namepackage>
```

RedOS, CentOS, Rosa Chrome

```
sudo dnf install <namepackage>
```

Настройка публичного репозитория

Данный шаг предназначен для установки из публичного репозитория. Если вы устанавливаете локальные пакеты, то пропустите этот шаг.

Подключение публичного репозитория DEB-систем

1. Создайте файл с информацией о репозиториях:

```
cat <<EOF | sudo tee /etc/apt/sources.list.d/rubackup_deb.list
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public-testing
EOF
```

где: `<OS-VERSION>` — версия используемой ОС:

- astra_1.6;
- astra_1.7;
- astra_1.8;
- debian_10;
- debian_12;
- ubuntu_18.04;

- ubuntu_20.04;
- ubuntu_22.04.

2. Добавьте ключ репозитория:

```
sudo wget -qO-
https://dl.astralinux.ru/artifactory/api/security/keypair/gc-astra-
official-repo-key/public | gpg --no-default-keyring --keyring gnupg-
ring:/etc/apt/trusted.gpg.d/rubackup-deb.gpg --import - && sudo chmod 644
/etc/apt/trusted.gpg.d/rubackup-deb.gpg
```

3. Обновите список пакетов:

```
sudo apt-get update
```

Подключение публичного репозитория RPM-систем

1. Создайте файл с информацией о репозиториях:
   a. для ОС:
      - CentOS 7;
      - CentOS 8;
      - РЕД ОС 7.3;
      - РЕД ОС 8;
      - Red Hat Enterprise Linux 9;
      - ROSA Fresh Desktop 12;
      - ROSA Enterprise Linux Server 7.9.

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/repodata/repomd.xml.key
gpgcheck=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
```

```
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/repodata/repomd.xml.key
gpgcheck=0
EOF
```

где: `<OS-VERSION>` — версия используемой ОС:

- centos_7;
- centos_8;
- redos_7.3;
- redos_8;
- rhel_9;
- rosa_12;
- rosa_7.9.

b. для ОС ROSA Enterprise Linux Server 7.3:

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/repodata/repomd.xml.key
gpgcheck=0
sslverify=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/repodata/repomd.xml.key
gpgcheck=0
```

```
sslverify=0
EOF
```

Настройка переменных среды

Выполните настройку переменных среды для пользователя `root`:

1. Авторизуйтесь под пользователем root:

```
sudo -i
```

2. Настройте переменные среды для пользователя `root`:

```
sudo nano /root/.bashrc
```

   - отредактируйте файл, добавив строки:

```
PATH=$PATH:/opt/rubackup/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib
export PATH
export LD_LIBRARY_PATH
```

   - сохраните изменения.

Эти переменные также можно определить в файле `/etc/environment`.

3. Перейдите в каталог `/root`:

```
cd /root
```

4. Перезагрузите переменные окружения:

```
source ~/.bashrc
```

Настройка SSL соединения с базой данных

Пропустите этот шаг, если не требуется защищённое подключение компонентов RuBackup к служебной базе данных.

Если необходимо использовать для подключения к базе данных PostgreSQL защи-

щённое соединение, то выполните приведённые ниже настройки на хостах, на которых развёрнуты компоненты СРК (postgres-клиенты):

1. Перенесите из соответствующей postgres-клиенту папки на узле Центра сертификации подготовленные:
   - сертификат Центра сертификации (ca.crt), чтобы клиент СРК мог проверить, что конечный сертификат сервера PostgreSQL был подписан его доверенным корневым сертификатом;
   - сертификат сервера/клиента СРК (postgresql.crt);
   - сгенерированный закрытый ключ сервера/клиента СРК (postgresql.key).
2. Для файлов сертификата и закрытого ключа установите полный доступ на чтение и запись только для владельцев:

   ```
   chmod 600 server.crt server.key ca.crt
   ```

3. Сделайте владельцем файлов пользователя, от имени которого будет запущен компонент СРК (postgres-клиент):

   ```
   chown suser:suser server.crt server.key ca.crt
   ```

8.2.2. Установка пакетов

Последовательность установки

Выполните установку пакетов строго в приведённой последовательности:

1. `rubackup-common`;
2. `rubackup-client`;
3. `rubackup-server`.

По умолчанию настройка осуществляется в терминале с помощью консольной утилиты `rb_init` (поставляется в составе `rubackup-client`).

Для настройки с помощью графической утилиты `rb_init_gui` дополнительно установите пакеты `rubackup-common-gui`, `rubackup-init-gui`.

Способы установки

Установите одним из способов:

1. Из локальной папки со скачанными пакетами:

Astra Linux, Debian, Ubuntu

`sudo apt install ./<namepackage>.deb`

Альт

`sudo apt-get install ./<namepackage>.rpm`

Rosa Cobalt, RHEL

`sudo yum install ./<namepackage>.rpm`

RedOS, CentOS, Rosa Chrome

`sudo dnf install ./<namepackage>.rpm`

2. Из репозитория ОС Astra Linux, Debian, Ubuntu:

```
sudo apt install <namepackage>
```

где `<namepackage>` — имя устанавливаемого пакета.

Обновление конфигурации

1. Выполните обновление конфигурации только для ОС Astra Linux Special Edition 1.6 или 1.7 с активированным режимом защитной программной среды:

```
sudo update-initramfs -u -k all
```

2. Примените изменения, выполнив перезагрузку ОС:

```
sudo reboot
```

8.2.3. Установка лицензии

Получение файла лицензии

Для получения файла лицензии сервера (основного, резервного и медиасерверов) у поставщика:

1. Полностью разверните серверную группировку запланированной архитектуры системы резервного копирования RuBackup, установив пакеты серверной части программы на узлах.
2. На каждом сервере получите идентификатор `hardware id`:

```
rubackup_server hwid
```

3. Зафиксируйте любым удобным способом для какого типа сервера (основной, резервный, медиа) получен идентификатор.
4. Предоставьте поставщику полученные идентификаторы удобным способом и получите лицензионные файлы для серверных компонентов СРК RuBackup на адрес электронной почты пользователя.

Установка файла лицензии

Установите файл лицензии на каждом узле лицензируемого сервера СРК RuBackup.

Для установки файла лицензии:

1. Переместите файл лицензии в папку `/opt/rubackup/etc/`, выполнив команду, находясь в папке с подготовленным файлом лицензионного ключа:

```
cp <файл_лицензии> /opt/rubackup/etc/rubackup.lic
```

2. Активация лицензии произойдёт после запуска сервера.

8.3. Настройка

8.3.1. Настройка сервера

Необходимо предварительно настроить сетевое взаимодействие узлов компонентов СРК RuBackup, используя FQDN, имя хоста или IP-адрес (далее по тексту — адрес).

Настройку компонентов СРК RuBackup следует произвести на каждом узле в строго приведённом порядке (в зависимости от архитектуры):

1. настройка основного сервера;
2. настройка резервного сервера;
3. настройка медиасервера (выполняется для каждого медиасервера);
4. настройка клиента системы резервного копирования (выполняется для каждого клиента СРК).

Настройку компонентов СРК RuBackup возможно выполнить одним из способов:

- настройка сервера в интерактивном режиме при помощи утилиты `rb_init`;
- настройка сервера в неинтерактивном режиме при помощи утилиты `rb_init`

(однострочной командой с заданными параметрами);

- настройка сервера с помощью графической утилиты мастера настройки RuBackup `rb_init_gui`.

Настройка сервера в терминале (интерактивный режим)

Выполните на каждом серверном узле интерактивную настройку СРК RuBackup с помощью `rb_init` (см. Сценарии настройки сервера).

```
sudo /opt/rubackup/bin/rb_init
```

Настройка сервера в терминале (неинтерактивный режим)

Неинтерактивный режим работы необходим для выполнения сценариев массового развертывания, например, при использовании *Ansible* — программного решения для удаленного управления конфигурациями серверов.

Администратор имеет возможность настроить СРК RuBackup в `bash/shell` однострочной командой и, как следствие, использовать эту команду в скриптах для автоматизации процесса.

Настройка СРК RuBackup осуществляется с помощью интерактивной утилиты `rb_init` (неинтерактивный режим). Описание утилиты приведено в документе.

Настройка сервера с помощью графической утилиты

Настройка сервера (основного, медиа или резервного) RuBackup с помощью мастера СРК RuBackup возможна с помощью графической утилиты мастера настройки RuBackup.

- Запустите мастер настройки RuBackup (графическое приложение `rb_init_gui`):

  ```
  rb_init_gui&
  ```

- После запуска мастера настройки RuBackup заполните открывшиеся формы:

  1. В приветственном окне (Рисунок 1):

     - выберите язык интерфейса приложения из предложенных вариантов (русский или английский);
     - примите лицензионное соглашения для продолжения настройки RuBackup, проставив отметку в чек-боксе ✓ **Применить**.

       Для ознакомления нажмите на активный элемент **[Лицензионное соглашение]** и в открывшемся окне подтверждения скопируйте в буфер

ссылку на лицензионное соглашение для дальнейшего просмотра в браузере;

- нажмите **[Далее]**.

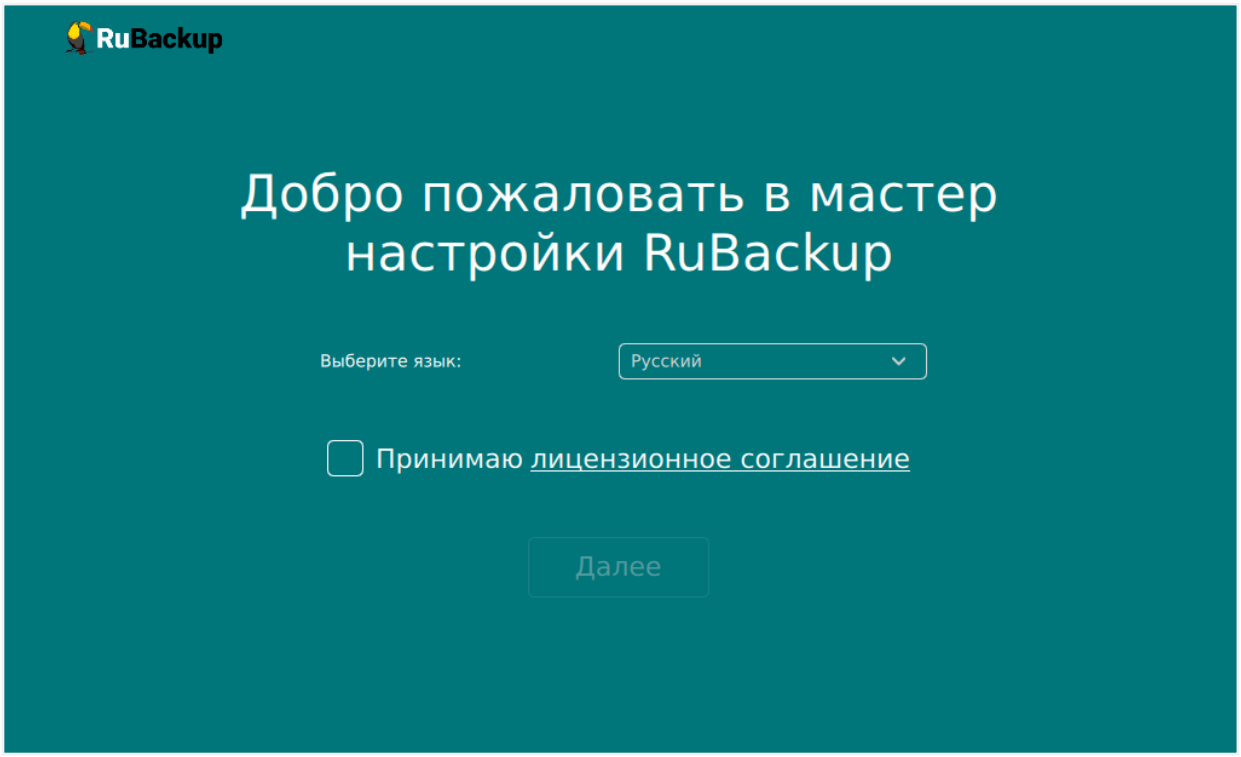


Рисунок 1. Приветственное окно Мастера настройки RuBackup

2. В открывшемся окне выберете настраиваемый компонент.

   Если на настраиваемом узле установлен пакет `rubackup-server`, то мастер настройки автоматически предлагает произвести настройку серверного компонента (Рисунок 2):

   - основной сервер;
   - резервный сервер;
   - медиасервер.

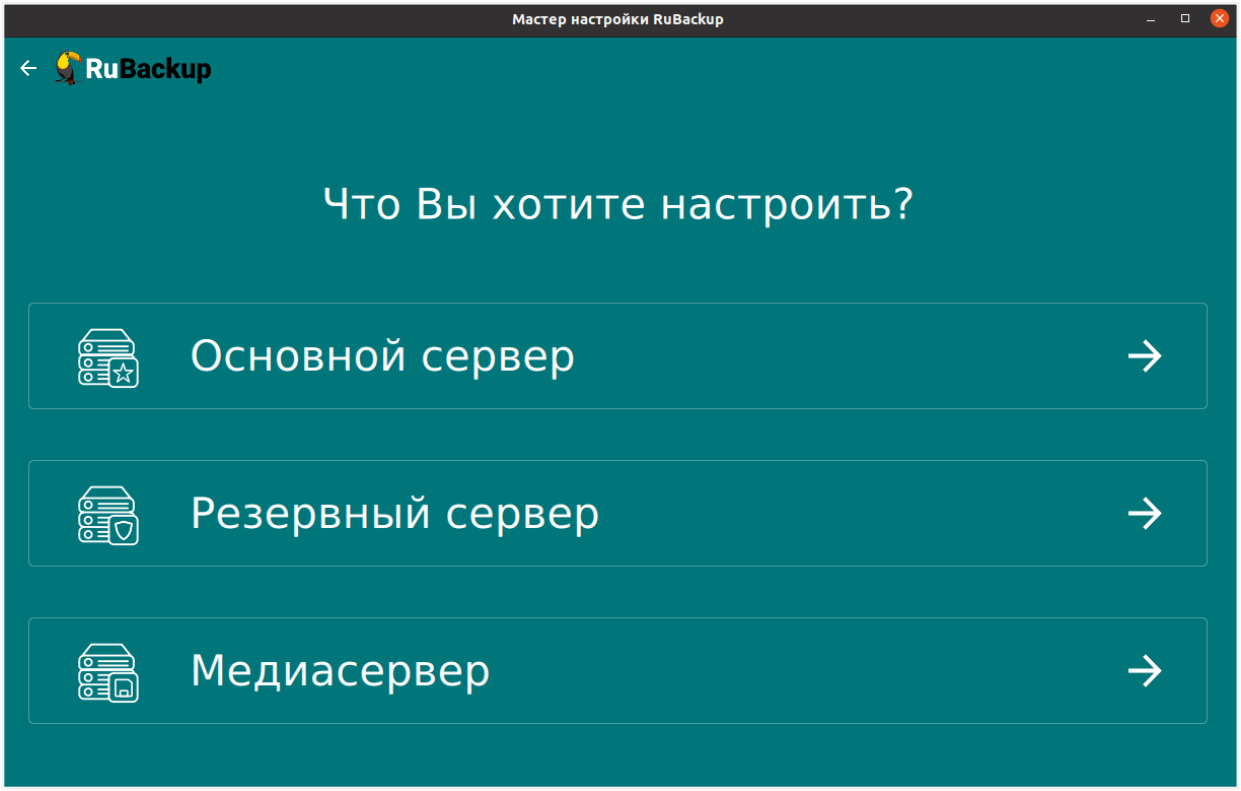
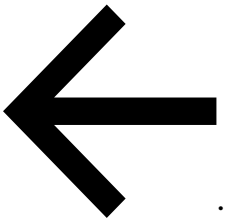


Рисунок 2. Окно выбора настраиваемого компонента RuBackup

3. Заполните открывшуюся форму настраиваемого компонента СРК RuBackup.

   Для возврата на предыдущий шаг и редактирования выбора используйте  .

   a. Блок **Общие параметры**:

      основной, резервный, медиасервер:

      - В поле **Количество сетевых потоков** укажите количество потоков для одновременной обработки задач резервного копирования на основном сервере (каждый поток имеет отдельное соединение со служебной базой данных СРК)

      основной, резервный, медиасервер:

      - В поле **Версия IP для DNS запросов** выберите какие публичные имена будут использованы DNS-сервером.

      основной, резервный, медиасервер:

      - Активируйте переключатель **Перезапись мастер-ключа** для автоматического формирования нового мастер-ключа и перезаписи (при

наличии) текущего мастер-ключа, который необходим при создании пары ключей электронно-цифровой подписи резервных копий и защитного преобразования резервных копий.

b. Блок **Параметры сервера**:

резервный, медиасервер:

- В поле **Имя основного сервера** укажите `ip-адрес` или `FQDN` основного сервера RuBackup (в соответствии с настройками файла `hosts` узла основного сервера).

основной, резервный, медиасервер:

- В поле **Адрес сервера PostgreSQL** [2] укажите адрес, на котором развёрнута СУБД PostgreSQL:

  - если СУБД PostgreSQL развёрнута на отдельном от основного сервера узле, то следует указать адрес соответствующего узла;
  - если СУБД PostgreSQL и основной сервер развёрнуты на одном узле, то оставьте значение `localhost`, выбранное по умолчанию ---

основной сервер:

- В поле **Пароль PostgreSQL** [2] укажите пароль пользователя базы данных `postgres`

основной сервер:

- В поле **Имя суперпользователя RuBackup** укажите имя суперпользователя базы данных `rubackup` (имя БД по умолчанию).

  Суперпользователь будет создан в процессе настройки основного сервера.

основной, резервный, медиасервер:

- В поле **Пароль пользователя RuBackup** [2] укажите пароль для суперпользователя базы данных `rubackup` (имя БД по умолчанию).

основной сервер:

- В поле **Имя базы RuBackup** введите имя базы данных (по умолчанию в качестве имени базы данных используется `rubackup`), которая будет использоваться в качестве служебной БД или будет создана в случае её отсутствия.

В имени базы данных запрещено использовать следующие символы: пробел, `\`, `$`, `#`, `` ` ``, `/`, `?`, `*`, `.`, `,`, `;`, `:`, `%`, `^`, `&`, `<`, `>`.

основной сервер:

- При обновлении в поле **Если база уже существует** выберите действие с существующей базой данных:
  - `keep` — пропустить действие, База данных будет сохранена в текущем состоянии;
  - `drop` — удалить существующую базу данных;
  - `upgrade` — обновить существующую базу данных.
- При удалении и обновлении существующей базы данных по умолчанию будет сделана резервная копия данных, если переключатель **Отключить дамп** деактивирован , если активировать данный переключатель, то резервное копирование для текущей базы данных перед удалением/обновлением выполнено не будет.
- Если резервное копирование существующей базу данных будет выполнено, то в поле **Формат дампа** выберите тип резервной копии базы данных:
  - `custom archives` — custom-архив, восстановление выполняется с помощью утилиты `pg_restore`. Резервная копия в формате `custom` занимает меньше места на диске, по сравнению с форматом `plain`.

    Настройте **Уровень сжатия дампа**;
  - `plain` — текстовый sql-скрипт.
- Для типа резервной копии БД `custom archives` в поле **Уровень сжатия дампа** выберите степень сжатия резервной копии базы данных (значение от 0 до 9). Чем выше степень сжатия, тем меньше архив занимает места на диске и тем дольше выполняется процедура резервного копирования базы данных.
- В поле **Путь к папке дампа** [2] выберите путь для сохранения резервной копии - по умолчанию это директория, откуда была вызвана утилита.

основной, резервный, медиасервер:

- В поле **Сетевой интерфейс** выберите сетевой интерфейс, посредством которого клиенту RuBackup разрешено взаимодействовать с системой резервного копирования.

основной сервер:

- В поле **Путь файловой системы для добавления в «Default»** [2] необходимо назначить для пула Default хотя бы один каталог для хранения резервных копий.

основной, резервный, медиасервер:

- В поле **Локальный каталог резервного копирования** укажите локальный каталог для временного хранения файлов с метаданными, создаваемых при операциях резервного копирования (по умолчанию при нажатии клавиши **Enter** в качестве директории для временных операций с файлами резервных копий используется `/tmp`). Если указанная директория не существует, то будет создана.

основной, медиасервер:

- В поле **Имя резервного сервера** укажите `ip-адрес` или `FQDN` основного сервера RuBackup (в соответствии с настройками файла `hosts` узла основного сервера).

основной, резервный, медиасервер:

- В поле **Количество параллельных задач** укажите количество потоков для одновременной обработки задач резервного копирования на медиасервере.

  Каждый поток имеет отдельное соединение со служебной базой данных СРК.

основной, резервный, медиасервер:

- В поле **Объём памяти дедупликации, байт** для ограничения потребления оперативной памяти сервером при дедупликации резервных копий.

  При использовании дедупликации рекомендуется минимальный объем оперативной памяти сервера 64 GB `effective_cache_size` ~70 % от размера оперативной памяти `work_mem` 32 MB.

основной, резервный, медиасервер:

- Активируйте переключатель **Непрерывная удалённая репликация** при необходимости на клиенте.

  Непрерывная удалённая репликация осуществляется только в хранилище блочного типа.

основной, резервный, медиасервер:

- Активируйте переключатель **Разрешать централизованное восстановление для клиента** для восстановления данных из резервной копии с помощью приложения «Менеджер администратора RuBackup» (RBM), с помощью консольной утилиты `rbfd` или приложения «Менеджера клиента RuBackup» (RBC).

  В случае деактивированного переключателя восстановление из резервной копии будет возможно с помощью консольной утилиты `rbfd` или приложения «Менеджера клиента RuBackup» на узле клиента резервного копирования. Централизованное восстановление данных из резервной копии с помощью приложения «Менеджер администратора RuBackup» (используемой на любом узле) будет отключено.

основной, резервный, медиасервер:

- Активируйте переключатель **Создать ключи ЭЦП**, если хотите создать ключи электронно-цифровой подписи.

  Резервная копия может быть подписана цифровой подписью для последующего контроля и предупреждения угрозы её подмены.

основной, резервный, медиасервер:

- Активируйте переключатель **Перезаписать ключи цифровой подписи**, для создания новой связки ключей, используемых для электронно-цифровой подиси.

основной сервер:

- Активируйте переключатель **Аудит безопасности** для журналирования всех значимых таблиц, кроме очередей задач и временных таблиц.

  Для расширения регистрируемых событий активируйте переключатель **Аудит задач** для журналирования всех значимых таблиц и задач в очередях.

  Позднее возможно включить/отключить данную опцию и изменить выбранный тип аудита с помощью утилиты для работы с журналом событий информационной безопасности `rb_security`.

c. Блок **Настройка SSL**:

основной, резервный, медиасервер:

- При необходимости настройки защищённого соединения со служебной базой данных активируйте переключатель **Использовать SSL соединение с базой данных** и настройте параметры:

- в поле **SSL режим работы с Postgres** выберите соответствующий режим работы (в зависимости от настроек узла, на котором установлена БД). Подробное описание режимов смотри в Раздел 7.4.

  Если в конфигурации PostgreSQL SSL выключен, то по умолчанию SSL режим будет `disable`;

- в поле **Корневой сертификат** [2] укажите полный путь к сертификату доверенного Центра сертификации (прописав в поле или выбрав по нажатию рядом с полем кнопки), который необходимо заранее разместить в папке `opt/rubackup/keys`;
- в поле **Сертификат клиента** [2] укажите полный путь к сертификату (открытому ключу) настраиваемого узла, выданный доверенным Центром сертификации (прописав в поле или выбрав по нажатию рядом с полем кнопки), который необходимо заранее разместить в папке `opt/rubackup/keys`;
- в поле **Ключ клиента** [2] укажите полный путь к закрытому ключу сертификата настраиваемого узла, выданный доверенным Центром сертификации (прописав в поле или выбрав по нажатию рядом с полем кнопки), который необходимо заранее разместить в папке `opt/rubackup/keys`.

4. После заполнения всех полей формы настраиваемого компонента СРК RuBackup нажмите **[Далее]**.

   В окне подтверждения нажмите **Да** для настройки компонента СРК RuBackup (Рисунок 3).

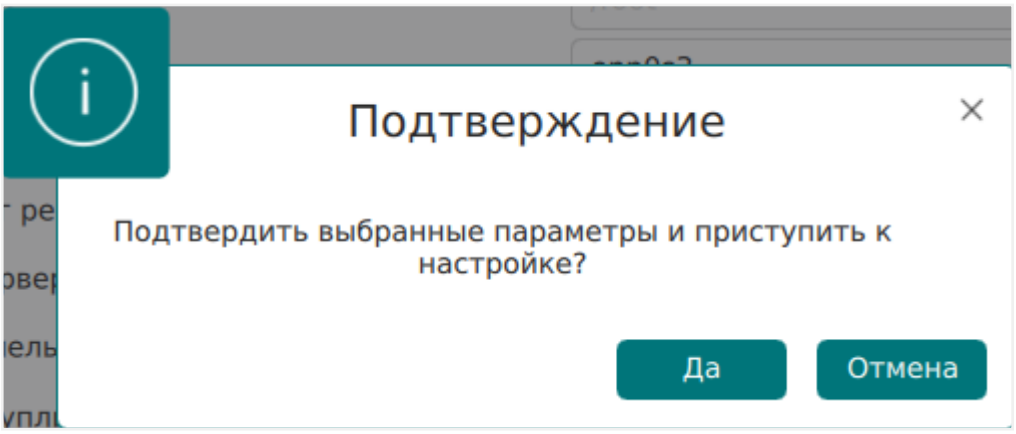


Рисунок 3. Окно подтверждения выбранных параметров

5. Если в форме настраиваемого компонента СРК RuBackup указаны папки, которых не существует, то будет выведено подтверждение для их создания (Рисунок 4).

   В окне подтверждения нажмите **Да** для создания папок.

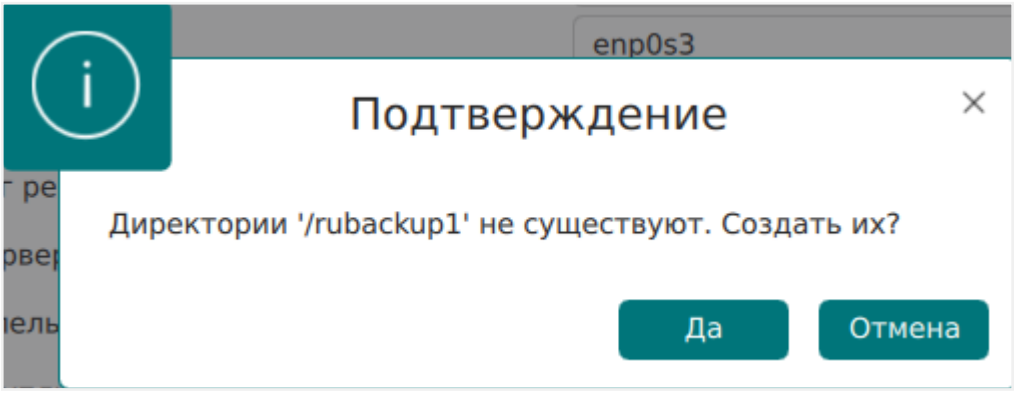


Рисунок 4. Окно подтверждения создания директорий

6. В случае успешной настройки пользователь будет уведомлён сообщением (Рисунок 5), в котором приведена информация:

- о лицензионном соглашении;
- правообладатель;
- версия продукта;
- имя текущего узла;
- тип настроенного компонента СРК RuBackup;
- о создании конфигурационного файла `/opt/rubackup/etc/config.file`;
- дополнительно могут быть приведены рекомендации и предупреждения по настройкам параметров.

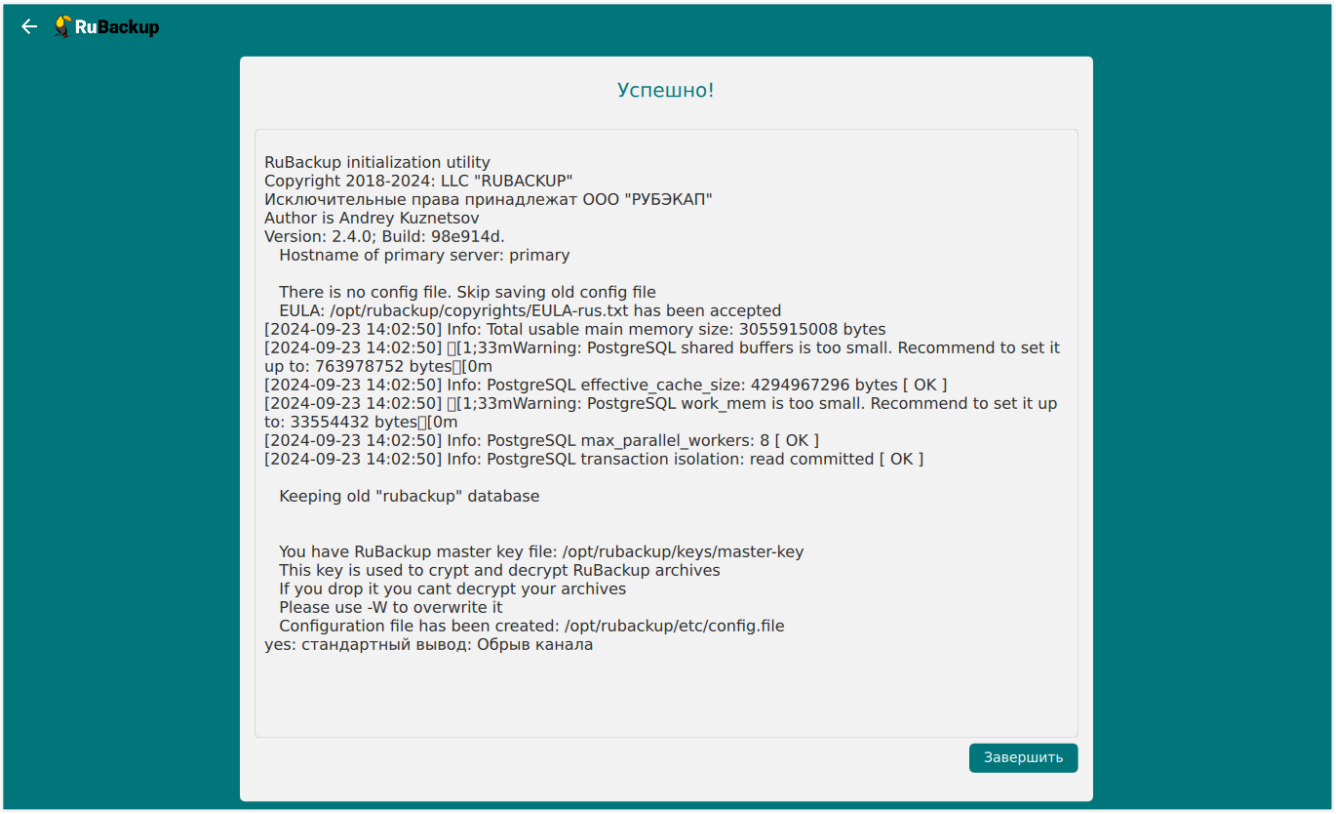


Рисунок 5. Окно результатов выполненной настройки сервера

1. Нажмите **Завершить** для завершения работы приложения.

8.3.2. Настройка окружения

Настройка пользователей

Пользователи, от имени которых будет осуществляться запуск утилит командной строки RuBackup или приложения для управления СРК RuBackup (RBM, RBC) должны:

- иметь правильно настроенные переменные среды;
- входить в группу `rubackup`.

Выполните приведённые ниже настройки для пользователей на всех узлах с развёрнутыми компонентами СРК RuBackup.

Настройка переменных среды

Настройте переменные среды для всех пользователей, которые будут работать с СРК RuBackup.

1. Откройте файл `.bashrc`, запускаемый в каждой сессии терминала:

   ```
   sudo nano /home/<имя пользователя>/.bashrc
   ```

2. Отредактируйте файл, добавив строки:

   ```
   export PATH=$PATH:/opt/rubackup/bin
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib
   ```

   Сохраните изменения.

3. Перезагрузите переменные окружения, находясь в каталоге `/home/<имя пользователя>/`:

   ```
   source ~/.bashrc
   ```

Переменные `PATH` и `LD_LIBRARY_PATH` можно переопределить в файлах:

- `/etc/profile`
- `/etc/bash.bashrc`

Добавление в группу

Группа `rubackup` автоматически создаётся в процессе установки пакета `rubackup-common`.

1. Добавьте пользователя в группу `rubackup`, выполнив команду:

```
sudo usermod -a -G rubackup <имя пользователя>
```

2. Если требуется запуск утилит командной строки RuBackup, RBM или RBC в текущем сеансе пользователя (без перезагрузки ОС) выполните:

```
newgrp rubackup
```

Настройка доступа к клиентским сертификатам

Настройте доступ пользователя, входящего в группу `rubackup`, к каталогам с сертификатами для запуска некоторых утилит командной строки, например, rb_clients.

По умолчанию доступ к каталогам есть только у пользователя `root`, для доступа другого пользователя:

1. Измените владельца и группу для каталогов, содержащих сертификаты:

```
sudo chown -R suser:rubackup /opt/rubackup/keys/client/
sudo chown -R suser:rubackup /opt/rubackup/keys/rootCA/
```

2. Перезапустите сервисы для применения изменений:

```
sudo systemctl restart rubackup_client.service
sudo systemctl restart rubackup_server.service
```

8.3.3. Добавление в автозапуск

1. Добавьте сервис клиента РК в автозапуск при загрузке ОС:

```
sudo systemctl enable rubackup_client.service
```

2. Добавьте сервис сервера РК в автозапуск при загрузке ОС:

```
sudo systemctl enable rubackup_server.service
```

3. Перезагрузите настройки ОС:

```
sudo systemctl daemon-reload
```

8.4. Запуск

Произведите активацию серверной части СРК RuBackup, выполнив на каждом узле с развёрнутым сервером (основным, резервным, медиа) RuBackup запуск сервиса клиента и сервиса сервера.

8.4.1. Запуск сервиса клиента

Для запуска сервиса клиента выполните команду:

```
sudo systemctl start rubackup_client.service
```

8.4.2. Запуск сервиса сервера

Для запуска сервиса сервера выполните команду:

```
sudo systemctl start rubackup_server.service
```

8.4.3. Просмотр статуса сервиса клиента

Для запуска просмотра статуса сервиса клиента выполните команду:

```
sudo systemctl status rubackup_client.service
```

8.4.4. Просмотр статуса сервиса сервера

Для просмотра статуса сервиса сервера выполните команду:

```
sudo systemctl status rubackup_server.service
```

8.4.5. Остановка сервиса клиента

Для останова сервиса клиента выполните команду:

```
sudo systemctl stop rubackup_client.service
```

8.4.6. Остановка сервиса сервера

Для останова сервиса сервера выполните команду:

```
sudo systemctl stop rubackup_server.service
```

[1] Выполните установку актуальной версии репозитория EPEL, для примера приведена установка репозитория EPEL 8
[2] обязательное для заполнения поле

Глава 9. Клиентская часть

Клиентская часть СРК RuBackup может состоять из одного или нескольких клиентов резервного копирования, которые могут быть объединены в группы клиентов.

Клиент резервного копирования — это отдельный сервер, компьютер или виртуальная машина, которая содержит данные для резервирования (ресурс) и на которой установлено клиентское ПО RuBackup для выполнения резервного копирования.

9.1. Linux

9.1.1. Системные требования

В данном подразделе приведены системные требования для каждого клиентского компонента СРК RuBackup, предъявляемые к техническим средствам, необходимым для нормального функционирования СРК RuBackup.

В случае установки на один хост нескольких компонентов СРК RuBackup (например, при способе установки «Всё в одном») следует консолидировать соответствующие аппаратные требования, предъявляемые к техническому средству, на которое производится установка.

Аппаратные требования

Минимальные аппаратные требования, необходимые для стабильного функционирования клиента системы резервного копирования приведены в таблице.

Таблица 8. Аппаратные требования, предъявляемые к Клиенту системы резервного копирования

Аппаратный компонент	Значение
Процессор	1 ядро

Аппаратный компонент	Значение
Оперативная память (RAM) [1]	*Пример 2. расчёт RAM при однопоточном режиме резервирования:* *Пример 3. расчёт RAM при многопоточном режиме резервирования:* $$RAM = RAM_1 + RAM_2 + ... + RAM_N$$ где: RAM_1 — объём оперативной памяти необходимый для резервирования одного ресурса; $0,04xV_{ресурса}$— 4% от размера резервируемого ресурса; N — количество единовременно резервируемых ресурсов

Аппаратный компонент	Значение
Дисковое пространство (HDD)[2]	*Пример 4. расчёт HDD по формуле:* где: *K=1* — при однопоточном режиме резервировании; *K= worker_parallelism* при многопоточном режиме (enable_multithreading) и слабой дедупликация (enable_flexible_dedup); enable multithreading — флаг, указывающий на использование многопоточности; enable flexible dedup — флаг, указывающий на использование гибкой дедупликации; *worker parallelism* — количество рабочих потоков, используемых для выполнения резервирования; _объём ресурса _— общий объём данных, подлежащих резервированию; *размер блока* — размер блока данных, используемого для обработки данных во время резервирования; *размер хеша* — размер хеша, используемого для идентификации данных; *20*— максимальный размер сериализованной позиции в файле; *1* — временная база для вычисления сигнатуры или отправки хешей на сервер; *размер метаданных* – это 0.02 * объем ресурса

Примеры расчётов оперативной памяти и дискового пространства:

Ресурс	Хеш	Блок	K	Размер метаданных	Дисковое пространство (ГБ)
536870912000	64	8192	8	10737418240	56
536870912000	64	8192	32	10737418240	179
536870912000	64	8192	64	10737418240	343
536870912000	64	8192	128	10737418240	671

536870912000	64	1048576	8	10737418240	10
536870912000	64	1048576	32	10737418240	11
536870912000	64	1048576	64	10737418240	12
536870912000	64	1048576	128	10737418240	15
1099511627776	64	8192	8	21990232555	114
1099511627776	64	8192	32	21990232555	366
1099511627776	64	8192	64	21990232555	702
1099511627776	64	8192	128	21990232555	1374
1099511627776	64	1048576	8	21990232555	21
1099511627776	64	1048576	32	21990232555	23
1099511627776	64	1048576	64	21990232555	25
1099511627776	64	1048576	128	21990232555	31

Программные требования

Программные требования, необходимые для стабильного функционирования клиентской части СРК RuBackup:

- операционная система из совместимых с компонентами СРК RuBackup:
  - Astra 1.6;
  - Astra 1.7;
  - Astra 1.8;
  - CentOS 7;
  - CentOS 8;
  - Debian 10;
  - Debian 12;
  - RHEL 9;
  - RedOS 7.3;
  - RedOS 8;
  - Rosa Chrome 12;
  - Rosa Cobalt 7.3;
  - Rosa Cobalt 7.9;
  - Ubuntu 18.04;
  - Ubuntu 20.04;
  - Ubuntu 22.04;
  - Альт 10;

- открытые порты в сооветствии с таблицей Сетевые порты;
- зависимости пакетов для каждой совместимой ОС:

Таблица 9. Зависимости `rubackup-client`

Операционная система	Пакеты
Astra 1.6	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
Astra 1.7	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
Astra 1.8	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
CentOS 7	qt5-qtbase-gui
CentOS 8	qt5-qtbase-gui
Debian 10	gnupg2 openssl wget xauth
Debian 12	gnupg2 openssl wget xauth
RHEL 9	qt5-qtbase-gui
RedOS 7.3	qt5-qtbase-gui
RedOS 8	qt5-qtbase-gui
Rosa Chrome 12	lib64qt5gui5 qt5-qtbase-gui

Операционная система	Пакеты
Rosa Cobalt 7.3	cups-libs fontconfig fontpackages-filesystem glx-utils libICE libSM libX11 libX11-common libXau libXdamage libXext libXfixes libXi libXrender libXxf86vm libicu libpng libxcb libxshmfence mesa-libEGL mesa-libGL mesa-libgbm mesa-libglapi qt5-qtbase qt5-qtbase-common qt5-qtbase-gui qt5-qtbase-gui xcb-util xcb-util-image xcb-util-keysyms xcb-util-renderutil xcb-util-wm
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 openssl wget xauth
Ubuntu 20.04	gnupg2 openssl wget xauth
Ubuntu 22.04	gnupg2 openssl wget xauth

Операционная система	Пакеты
Альт 10	qt5-qtbase-gui xauth

Таблица 10. Зависимости `rubackup-common-gui`

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth
Astra 1.8	gnupg2 wget xauth
Debian 10	gnupg2 wget xauth
Debian 12	gnupg2 wget xauth
Rosa Chrome 12	qt5-qtbase-gui
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 wget xauth
Ubuntu 20.04	gnupg2 wget xauth
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	xauth

Таблица 11. Зависимости `rubackup-common`

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth

Операционная система	Пакеты
Astra 1.8	gnupg2 wget xauth
Debian 10	gnupg2 wget xauth
Debian 12	gnupg2 wget xauth
Rosa Chrome 12	qt5-qtbase-gui
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 wget xauth
Ubuntu 20.04	gnupg2 wget xauth
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	xauth

9.1.2. Установка

Подготовка к установке

Установка зависимостей пакетов

Данный шаг предназначен для установки локальных пакетов. Если вы устанавливаете пакеты из репозитория, то пропустите этот шаг.

Для успешного развёртывания сервера СРК RuBackup необходимо наличие установленных зависимостей пакетов в соответствии с таблицей, в зависимости от используемого типа операционной системы на узле развёртывания клиента резервного копирования RuBackup, для этого:

1. Проверьте наличие установленных пакетов зависимостей в ОС, например:

   Astra Linux, Debian, Ubuntu `dpkg-query -l`

   Альт `apt list --installed`

   Rosa Cobalt, RHEL `yum list с опцией installed`

RedOS, CentOS, Rosa Chrome	`dnf list installed`

2. Если вы используете операционную систему CentOS 7, CentOS 8 или RHEL 9, то добавьте репозиторий `EPEL` [3], поддерживаемый в рамках проекта Fedora и содержащий некоторые пакеты, которые не вошли в стандартный набор RHEL (CentOS):

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   Файл репозитория будет автоматически загружен в каталог `/etc/yum.repos.d/epel.repo` и активирован.

3. Если вы используете операционную систему CentOS 7 или CentOS 8, то также рекомендуется включить репозиторий `PowerTools`, поскольку пакеты `EPEL` могут зависеть от пакетов из него:

```
sudo dnf config-manager --set-enabled powertools
```

4. Если вы используете операционную систему RHEL 9, то также рекомендуется включить репозиторий `codeready-builder-for-rhel-8-*-` репозиторий `rpm`, поскольку пакеты `EPEL` могут зависеть от пакетов из него:

```
ARCH=$( /bin/arch )

sudo subscription-manager repos --enable "codeready-builder-for-rhel-8-$\{ARCH}-rpms"
```

5. Обновите репозитории пакетов в системе:

Astra Linux, Debian, Ubuntu	`sudo apt update`
Альт	`sudo apt-get update`
Rosa Cobalt, RHEL	`sudo yum update`
RedOS, CentOS, Rosa Chrome	`sudo dnf update`

6. Установите недостающие зависимости пакетов из таблицы:

Astra Linux, Debian, Ubuntu	`sudo apt install <namepackage>`
Альт	`sudo apt-get install <namepackage>`
Rosa Cobalt, RHEL	`sudo yum install <namepackage>`

RedOS, CentOS, Rosa Chrome `sudo dnf install <namepackage>`

Настройка публичного репозитория

Данный шаг предназначен для установки из публичного репозитория. Если вы устанавливаете локальные пакеты, то пропустите этот шаг.

Подключение публичного репозитория DEB-систем

1. Создайте файл с информацией о репозиториях:

```
cat <<EOF | sudo tee /etc/apt/sources.list.d/rubackup_deb.list
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public
deb https://dl.astralinux.ru/rubackup/repository-deb-main/ <OS-VERSION>
public-testing
EOF
```

   где: `<OS-VERSION>` — версия используемой ОС:

   - astra_1.6;
   - astra_1.7;
   - astra_1.8;
   - debian_10;
   - debian_12;
   - ubuntu_18.04;
   - ubuntu_20.04;
   - ubuntu_22.04.

2. Добавьте ключ репозитория:

```
sudo wget -qO-
https://dl.astralinux.ru/artifactory/api/security/keypair/gc-astra-
official-repo-key/public | gpg --no-default-keyring --keyring gnupg-
ring:/etc/apt/trusted.gpg.d/rubackup-deb.gpg --import - && sudo chmod 644
/etc/apt/trusted.gpg.d/rubackup-deb.gpg
```

3. Обновите список пакетов:

```
sudo apt-get update
```

Подключение публичного репозитория RPM-систем

1. Создайте файл с информацией о репозиториях:
   a. для ОС:
      - CentOS 7;
      - CentOS 8;
      - РЕД ОС 7.3;
      - РЕД ОС 8;
      - Red Hat Enterprise Linux 9;
      - ROSA Fresh Desktop 12;
      - ROSA Enterprise Linux Server 7.9.

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public/repodata/repomd.xml.key
gpgcheck=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-main/<OS-
VERSION>/public-testing/repodata/repomd.xml.key
gpgcheck=0
EOF
```

где: `<OS-VERSION>` — версия используемой ОС:

- centos_7;
- centos_8;
- redos_7.3;
- redos_8;

- rhel_9;
- rosa_12;
- rosa_7.9.

b. для ОС ROSA Enterprise Linux Server 7.3:

```
cat <<EOF | sudo tee /etc/yum.repos.d/rubackup_rpm.repo
[rubackup-rpm-public-repository]
name=rubackup rpm public repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public/repodata/repomd.xml.key
gpgcheck=0
sslverify=0

[rubackup-rpm-public-testing-repository]
name=rubackup rpm public testing repository
baseurl=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/
enabled=1
repo_gpgcheck=1
gpgkey=https://dl.astralinux.ru/artifactory/rubackup-rpm-
main/rosa_7.3/public-testing/repodata/repomd.xml.key
gpgcheck=0
sslverify=0
EOF
```

Настройка переменных среды

Выполните настройку переменных среды для пользователя `root`:

1. Авторизуйтесь под пользователем root:

```
sudo -i
```

2. Настройте переменные среды для пользователя `root`:

```
sudo nano /root/.bashrc
```

- отредактируйте файл, добавив строки:

```
PATH=$PATH:/opt/rubackup/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib
export PATH
export LD_LIBRARY_PATH
```

- сохраните изменения.

Эти переменные также можно определить в файле `/etc/environment`.

3. Перейдите в каталог `/root`:

```
cd /root
```

4. Перезагрузите переменные окружения:

```
source ~/.bashrc
```

Настройка SSL соединения с базой данных

Пропустите этот шаг, если не требуется защищённое подключение компонентов RuBackup к служебной базе данных.

Если необходимо использовать для подключения к базе данных PostgreSQL защищённое соединение, то выполните приведённые ниже настройки на узлах, на которых развёрнуты компоненты СРК (postgres-клиенты):

1. Перенесите из соответствующей postgres-клиенту папки на узле Центра сертификации подготовленные:
   - сертификат Центра сертификации (ca.crt), чтобы клиент СРК мог проверить, что конечный сертификат сервера PostgreSQL был подписан его доверенным корневым сертификатом;
   - сертификат клиента резервного копирования (postgresql.crt);
   - сгенерированный закрытый ключ клиента резервного копирования (postgresql.key).
2. Для файлов сертификата и закрытого ключа установите полный доступ на чтение и запись только для владельцев:

```
chmod 600 server.crt server.key ca.crt
```

3. Сделайте владельцем файлов пользователя, от имени которого будет запущен клиент резервного копирования (postgres-клиент):

```
chown suser:suser server.crt server.key ca.crt
```

Установка пакетов

Установку пакетов производить строго в приведённой последовательности!

1. Установите одним из способов:
   - из локальной папки со скачанными пакетами:

Astra Linux, Debian, Ubuntu	`sudo apt install ./<namepackage>.deb`
Альт	`sudo apt-get install ./<namepackage>.rpm`
Rosa Cobalt, RHEL	`sudo yum install ./<namepackage>.rpm`
RedOS, CentOS, Rosa Chrome	`sudo dnf install ./<namepackage>.rpm`

   - из репозитория:

Astra Linux, Debian, Ubuntu	`sudo apt install <namepackage>.deb`

где `<namepackage>` — устанавливаемый пакет СРК RuBackup актуальной версии в приведённой последовательности:

a. `rubackup-common`;
b. `rubackup-client`;

необязательные пакеты, используются для настройки сервера с помощью графической утилиты:

c. `rubackup-common-gui`;
d. `rubackup-init-gui`.

По умолчанию настройка сервера осуществляется в терминале с помощью утилиты `rb_init`, которая не требует дополнительной инсталляции.

2. Выполните обновление конфигурации и примените изменения.

Данный шаг выполняется только для ОС Astra Linux Special Edition 1.6 или 1.7 с активированным режимом защитной программной среды!

- Обновите конфигурацию:

```
sudo update-initramfs -u -k all
```

- Примените изменения:

```
sudo reboot
```

9.1.3. Настройка

Настройка клиента РК

Настройку компонентов СРК RuBackup следует произвести на каждом узле в строго приведённом порядке (в зависимости от архитектуры) :

1. настройка основного сервера;
2. настройка резервного сервера;
3. настройка медиасервера (выполняется для каждого медиасервера);
4. настройка клиента системы резервного копирования (выполняется для каждого клиента СРК).

Необходимо предварительно настроить сетевое взаимодействие узлов компонентов СРК RuBackup, используя FQDN, имя хоста или IP-адрес (далее по тексту — адрес).

Настройка клиента РК в терминале (интерактивный режим)

Выполните на каждом узле клиента интерактивную настройку СРК RuBackup с помощью `rb_init` (см. Сценарии настройки клиента).

```
sudo /opt/rubackup/bin/rb_init
```

Настройка клиента РК в терминале (неинтерактивный режим)

Неинтерактивный режим работы необходим для выполнения сценариев массового развертывания, например, при использовании Ansible — программного решения для удаленного управления конфигурациями серверов.

Администратор имеет возможность настроить СРК RuBackup в bash/shell однострочной командой и, как следствие, использовать эту команду в скриптах для автоматизации процесса.

Настройка СРК RuBackup осуществляется с помощью интерактивной утилиты `rb_init` (неинтерактивный режим). Описание утилиты см. Утилиты командной строки.

Настройка клиента РК с помощью графической утилиты

Настройка клиента резервного копирования возможна с помощью графической утилиты мастера настройки RuBackup.

- Запустите мастер настройки RuBackup (графическое приложение `rb_init_gui`), выполнив команду:

```
rb_init_gui&
```

- После запуска мастера настройки RuBackup заполните открывшиеся формы:
  1. В приветственном окне (Рисунок 6):
     - выберите язык интерфейса приложения из предложенных вариантов (русский или английский);
     - примите лицензионное соглашения для продолжения настройки компонента RuBackup, проставив отметку в чек-боксе ✓ **Принимаю лицензионное соглашение**.

       Для ознакомления нажмите на активный элемент **[Лицензионное соглашение]** и скопируйте в буфер ссылку на лицензионное соглашение для дальнейшего просмотра в браузере;
     - нажмите **[Далее]**.

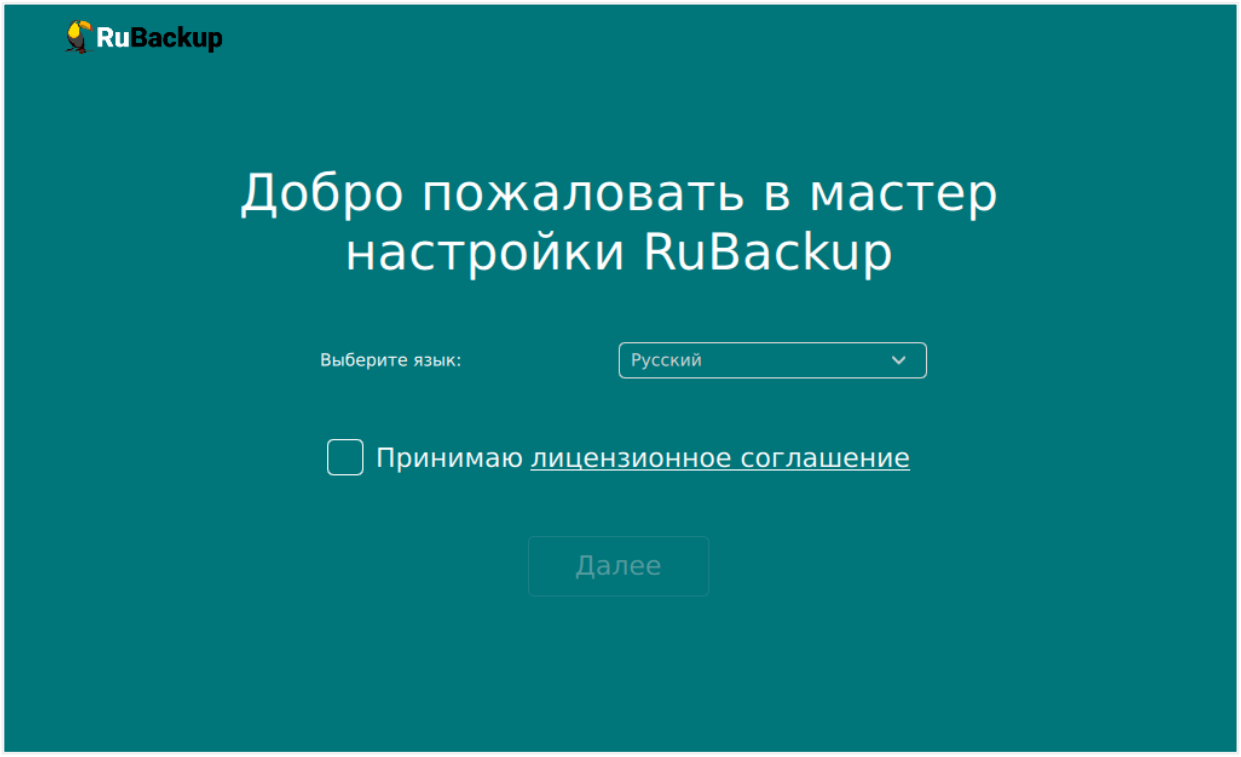


Рисунок 6. Приветственное окно Мастера настройки RuBackup

2. В открывшемся окне выберете режим настраиваемого клиента резервного копирования (Рисунок 7):

- автономный режим клиента РК предусматривает использование функций СРК без серверной части с сохранением возможности использования любых функциональных модулей для создания резервных копий;
- клиент-серверный режим клиента РК предусматривает использований всех доступных функций СРК.

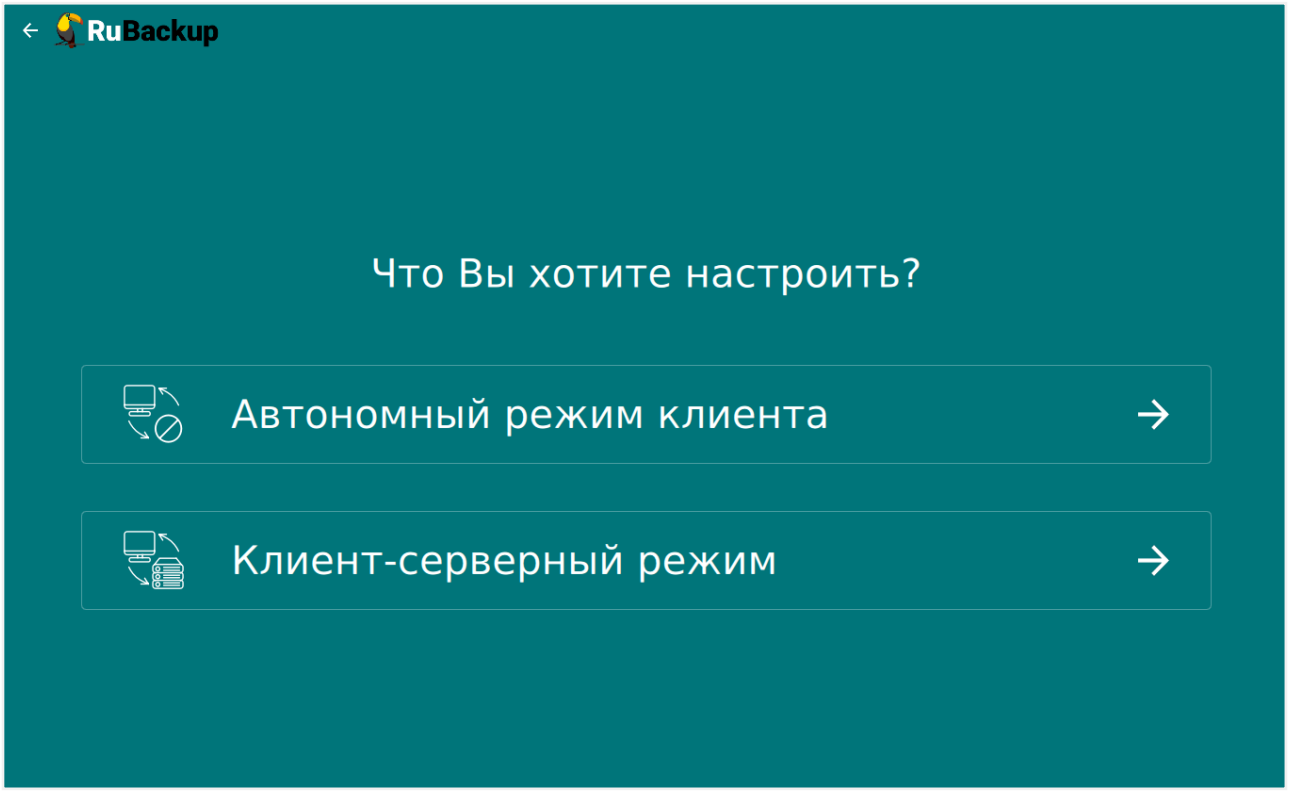


Рисунок 7. Окно выбора режима настраиваемого компонента RuBackup

Клиент-серверный режим работы клиента РК

1. Заполните открывшуюся форму настраиваемого клиента резервного копирования RuBackup.
   a. Блок **Общие параметры**
      - В поле **Количество сетевых потоков** укажите количество потоков для одновременной обработки задач резервного копирования на основном сервере (каждый поток имеет отдельное соединение со служебной базой данных СРК)
      - В поле **Версия IP для DNS запросов** выберите какие публичные имена будут использованы DNS-сервером.
      - Активируйте переключатель **Перезапись мастер-ключа** для автоматического формирования нового мастер-ключа и перезаписи (при наличии) текущего мастер-ключа.
   b. Блок **Параметры клиент-серверного режима**
      - В поле **Имя основного сервера** укажите `ip-адрес` или `FQDN` основного сервера RuBackup (в соответствии с настройками файла `hosts` узла основного сервера).
      - В поле **Имя резервного сервера** укажите `ip-адрес` или `FQDN` резервного сервера RuBackup (в соответствии с настройками файла `hosts` узла основного сервера).
      - В поле **Сетевой интерфейс** выберите сетевой интерфейс, посредством

которого клиенту РК разрешено взаимодействовать с системой резервного копирования.

- В поле **Локальный каталог резервного копирования** укажите локальный каталог для временного хранения файлов с метаданными, создаваемых при операциях резервного копирования (по умолчанию при нажатии клавиши **Enter** в качестве директории для временных операций с файлами резервных копий используется `/tmp`). Если указанная директория не существует, то будет создана.
- В поле **Количество параллельных задач** укажите количество потоков для одновременной обработки задач резервного копирования на медиасервере (каждый поток имеет отдельное соединение со служебной базой данных СРК).
- В поле **Объём памяти дедупликации, байт** для ограничения потребления оперативной памяти сервером при дедупликации резервных копий.

  При использовании дедупликации рекомендуется минимальный объем оперативной памяти сервера 64 GB `effective_cache_size` ~70 % от размера оперативной памяти `work_mem` 32 MB.

- Активируйте переключатель **Непрерывная удалённая репликация** при необходимости на клиенте. Непрерывная удалённая репликация осуществляется только в хранилище блочного типа.
- Активируйте переключатель **Разрешать централизованное восстановление для клиента** для восстановления данных из резервной копии в приложении «Менеджер администратора RuBackup» (RBM), с помощью консольной утилиты `rbfd` или приложения «Менеджер клиента RuBackup» (RBC).

  В случае деактивированного переключателя восстановление из резервной копии будет возможно с помощью консольной утилиты `rbfd` или приложения «Менеджер клиента RuBackup» на узле клиента резервного копирования.

  Централизованное восстановление данных из резервной копии с помощью приложения «Менеджер администратора RuBackup» (используемом на любом узле) будет отключено.

- Активируйте переключатель **Создать ключи ЭЦП**, если хотите создать ключи электронно-цифровой подписи.

  Резервная копия может быть подписана цифровой подписью для последующего контроля и предупреждения угрозы её подмены.

- Активируйте переключатель **Системный мониторинг для клиента**, если хотите включить системный мониторинг для данного клиента.

Файл мониторинга производительности системных компонентов будет размещён в папке `/opt/rubackup/monitoring/`.

- Активируйте переключатель **Перезаписать ключи цифровой подписи** , для создания новой связки ключей, используемых для электронно-цифровой подписи.

2. После заполнения всех полей формы настраиваемого компонента СРК RuBackup нажмите **[Далее]**.

   В окне подтверждения нажмите **Да** для настройки компонента СРК RuBackup(Рисунок 8).

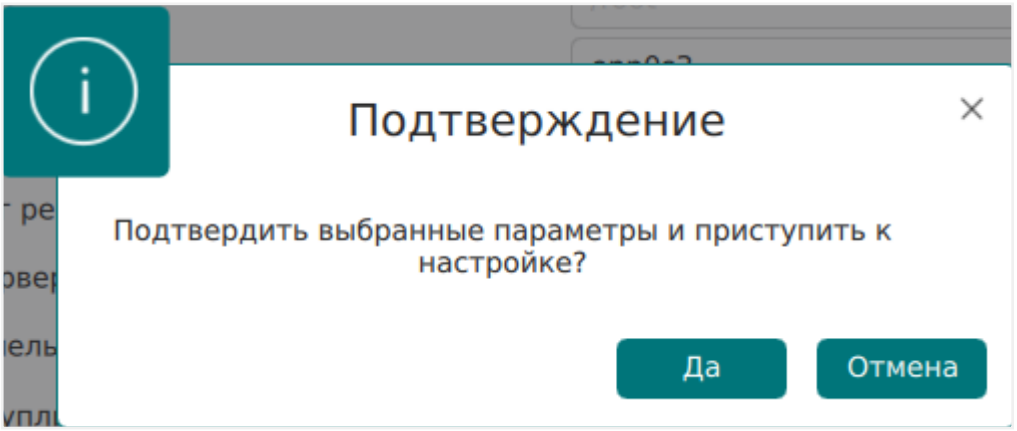


Рисунок 8. Окно подтверждения выбранных параметров

3. Если в форме настраиваемого компонента СРК RuBackup указаны папки, которых не существует, то будет выведено подтверждение для их создания (Рисунок 9).

   В окне подтверждения нажмите **Да** для создания папок.

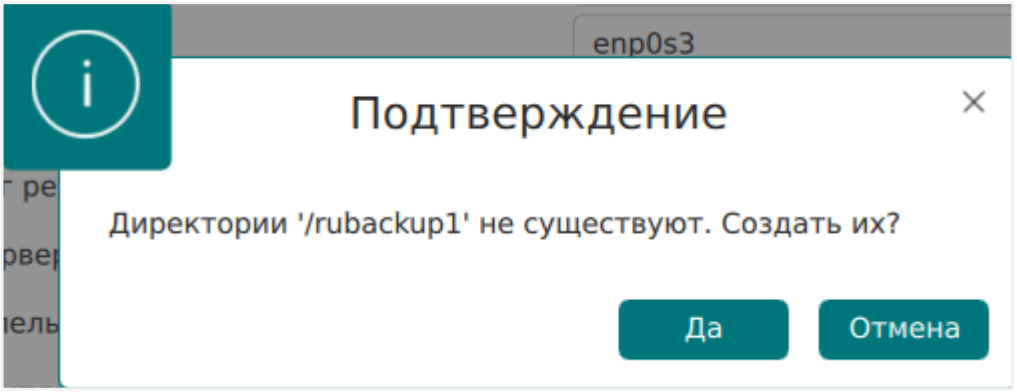


Рисунок 9. Окно подтверждения создания директорий

4. В случае успешной настройки пользователь будет уведомлён сообщением (Рисунок 10), в котором приведена информация:

   - о лицензионном соглашении;
   - правообладатель;

- версия продукта;
- имя текущего узла;
- тип настроенного компонента СРК RuBackup;
- о создании конфигурационного файла `/opt/rubackup/etc/config.file`;
- дополнительно могут быть приведены рекомендации и предупреждения по настройкам параметров.

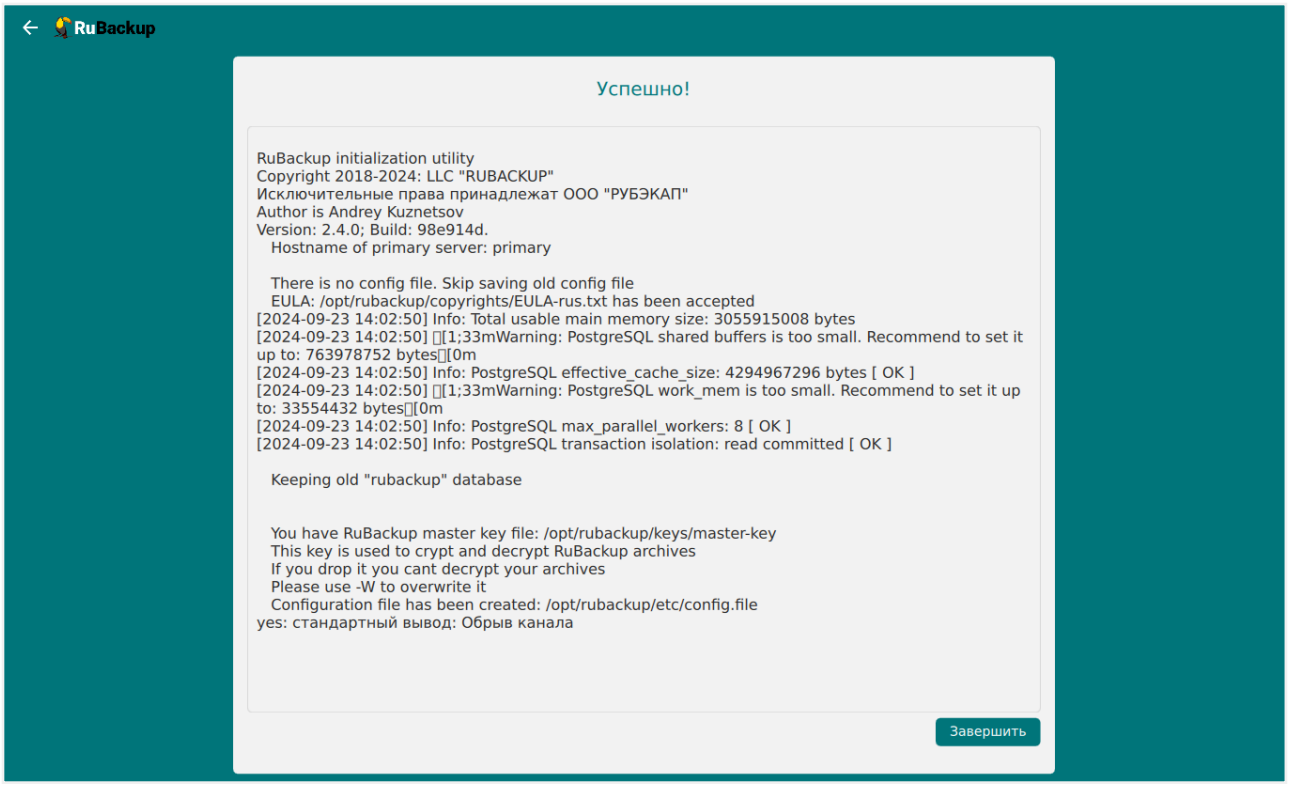


Рисунок 10. Окно результатов выполненной настройки клиента РК

5. Нажмите **Завершить** для завершения работы приложения.

Автономный режим работы клиента РК

1. Заполните открывшуюся форму настраиваемого клиента резервного копирования.
   a. Блок **Общие параметры**
      - В поле **Количество сетевых потоков** укажите количество потоков для одновременной обработки задач резервного копирования на основном сервере (каждый поток имеет отдельное соединение со служебной базой данных СРК)
      - В поле **Версия IP для DNS запросов** выберите какие публичные имена будут использованы DNS-сервером.
      - Активируйте переключатель **Перезапись мастер-ключа** для автоматического формирования нового мастер-ключа и перезаписи (при наличии) текущего мастер-ключа.

b. Блок **Параметры автономного клиента**

- В поле **Каталог архивирования** [4] выберите каталог для временного хранения резервных копий. Если этот параметр не определен в файле конфигурации, то клиент будет запрашивать у медиасервера временное пространство для операций с резервными копиями (NFS папку).
- В поле **Метод сжатия** выберите тип сжатия резервных копий:
  - `none` — без сжатия;
  - `fast` — многопоточный аналог `optimal`;
  - `optimal` — стандартная утилита сжатия Linux;
  - `best` — больший коэффициент сжатия, чем `optimal`, при большем времени.
- В поле **Тип хранилища резервных копий** выберите тип каталога для хранения резервных копий:
  - локальный каталог — каталог расположен на текущем узле клиента резервного копирования. Если выбран этот тип хранилища, то в поле **Локальный каталог резервного копирования** укажите полный путь к каталогу (прописав в поле или выбрав по нажатию рядом с полем кнопки **[...]**;
  - сетевой каталог — общий каталог с сетевым доступом. Если выбран этот тип хранилища, то необходимо:
    - В поле **Тип сетевого каталога** выбрать протокол для обеспечения удалённой связи: `nfs` (для ОС UNIX и Linux) или `cifs` (для ОС Windows).
    - В поле **Предназначенное устройство** укажите выделенное локальное устройство (например: `/dev/sdb`) или сетевой ресурс для хранения резервных копий (например: `srv://net_share`).
    - В поле **Параметры монтирования** укажите место монтирования файловых системы LTFS. Для работы с лентами LTO RuBackup использует файловую систему LTFS. По умолчанию точка монтирования — каталог `/opt/rubackup/mnt`.

2. После заполнения всех полей формы настраиваемого компонента СРК RuBackup нажмите **[Далее]**.

   В окне подтверждения нажмите **Да** для настройки компонента СРК RuBackup (Рисунок 11).

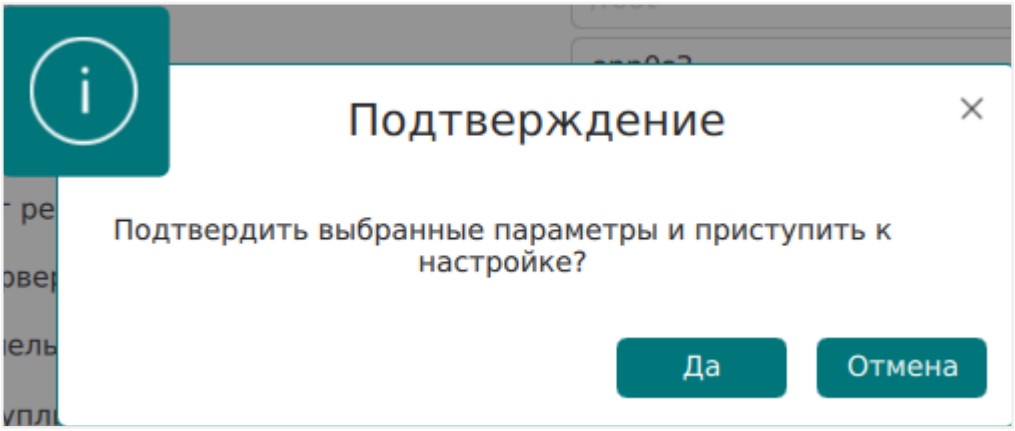


Рисунок 11. Окно подтверждения выбранных параметров

3. Если в форме настраиваемого компонента СРК RuBackup указаны папки, которых не существует, то будет выведено подтверждение для их создания (Рисунок 12).

   В окне подтверждения нажмите **Да** для создания папок.

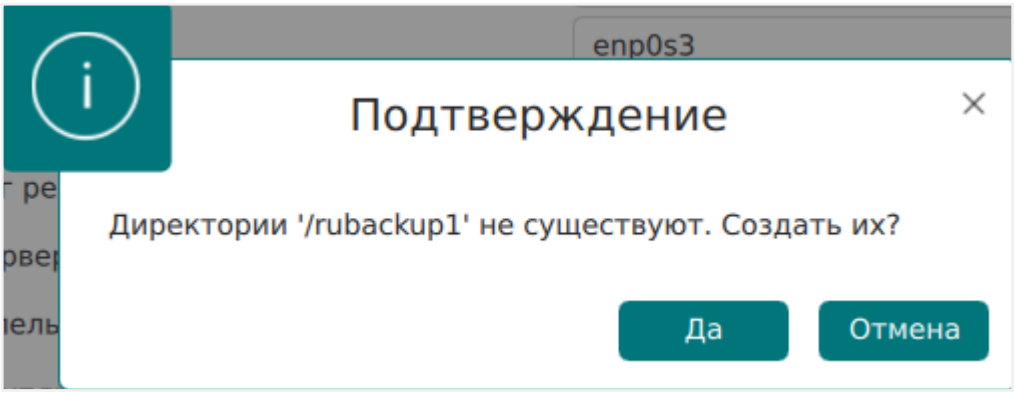


Рисунок 12. Окно подтверждения создания директорий

4. В случае успешной настройки пользователь будет уведомлён сообщением, в котором приведена информация:

   - о лицензионном соглашении;
   - правообладатель;
   - версия продукта;
   - имя текущего узла;
   - тип настроенного компонента СРК RuBackup;
   - о создании конфигурационного файла `/opt/rubackup/etc/config.file`;
   - дополнительно могут быть приведены рекомендации и предупреждения по настройкам параметров.

5. Нажмите **Завершить** для завершения работы приложения.

Настройка окружения

Настройка пользователей

Пользователи, от имени которых будет осуществляться запуск утилит командной строки RuBackup или приложения для управления СРК RuBackup (RBM, RBC) должны:

- иметь правильно настроенные переменные среды;
- входить в группу `rubackup`.

Выполните приведённые ниже настройки для пользователей на всех узлах с развёрнутыми компонентами СРК RuBackup.

Настройка переменных среды

Настройте переменные среды для всех пользователей, которые будут работать с СРК RuBackup.

1. Откройте файл `.bashrc`, запускаемый в каждой сессии терминала:

   ```
   sudo nano /home/<имя пользователя>/.bashrc
   ```

2. Отредактируйте файл, добавив строки:

   ```
   export PATH=$PATH:/opt/rubackup/bin
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib
   ```

   Сохраните изменения.

3. Перезагрузите переменные окружения, находясь в каталоге `/home/<имя пользователя>/`:

   ```
   source ~/.bashrc
   ```

Переменные `PATH` и `LD_LIBRARY_PATH` можно переопределить в файлах:

- `/etc/profile`
- `/etc/bash.bashrc`

Добавление в группу

Группа `rubackup` автоматически создаётся в процессе установки пакета `rubackup-common`.

1. Добавьте пользователя в группу `rubackup`, выполнив команду:

```
sudo usermod -a -G rubackup <имя пользователя>
```

2. Если требуется запуск утилит командной строки RuBackup, RBM или RBC в текущем сеансе пользователя (без перезагрузки ОС) выполните:

```
newgrp rubackup
```

Настройка доступа к клиентским сертификатам

Настройте доступ пользователя, входящего в группу `rubackup`, к каталогам с сертификатами для запуска некоторых утилит командной строки, например, rb_clients.

По умолчанию доступ к каталогам есть только у пользователя `root`, для доступа другого пользователя:

1. Измените владельца и группу для каталогов, содержащих сертификаты:

```
sudo chown -R suser:rubackup /opt/rubackup/keys/client/
sudo chown -R suser:rubackup /opt/rubackup/keys/rootCA/
```

2. Перезапустите сервисы для применения изменений:

```
sudo systemctl restart rubackup_client.service
sudo systemctl restart rubackup_server.service
```

Добавление в автозапуск

1. Добавьте сервис клиента РК в автозапуск при загрузке ОС:

```
sudo systemctl enable rubackup_client.service
```

2. Перезагрузите настройки ОС:

```
sudo systemctl daemon-reload
```

9.1.4. Запуск

Произведите активацию клиентской части СРК RuBackup, выполнив на каждом

узле с развёрнутым клиентом резервного копирования запуск сервиса клиента.

Для успешного запуска клиента РК в клиент-серверном режиме предварительно необходимо запустить серверную часть СРК.

Запуск сервиса клиента

Для запуска сервиса клиента РК выполните команду:

```
sudo systemctl start rubackup_client.service
```

Просмотр статуса сервиса клиента

Для просмотра статуса сервиса клиента выполните команду:

```
sudo systemctl status rubackup_client.service
```

Остановка сервиса клиента

Для останова сервиса клиента выполните команду:

```
sudo systemctl stop rubackup_client.service
```

9.2. Windows

9.2.1. Системные требования

В данном подразделе приведены системные требования для каждого клиентского компонента СРК RuBackup, предъявляемые к техническим средствам, необходимым для нормального функционирования СРК RuBackup.

Аппаратные требования

Требования к аппаратным средствам клиента РК

Узел, выполняющий функции клиента РК, на котором предполагается развёртывание, должен обладать характеристиками, приведёнными в таблице Таблица 12.

Таблица 12. Требования к аппаратным средствам клиента РК

Аппаратное требование	Значение	Примечание

Процессор	Однопоточный режим	Многопоточный режим	—
	1 ядро	Количество ядер = количеству потоков	
Твердотельный накопитель	Значение требуемого дискового пространства может быть рассчитано по формуле		Не менее 400 ГБ
Оперативная память	Сумма значений оперативной памяти для всех задач резервного копирования		Где оперативная память одного ресурса равна 1ГБ + 4% от размера целевого ресурса
Интерфейсное устройство	Сетевой адаптер		—

Пример 5. Формула расчёта дискового пространства

$$V = \frac{Vol_{resource}}{Size_{block}} \times (Size_{hash} + 20) \times (K + 1) + Size_{metadata}$$

где:

- K = 1 при однопоточном режиме;
- K = `worker_parallelism`, если заданы многопоточный режим (`enable_multithreading`) и слабая дедупликация (`enable_flexible_dedup`);
  - `worker_parallelism` — количество рабочих потоков, используемых для выполнения РК;
  - `enable_multithreading` — флаг, указывающий на использование многопоточности;
  - `enable_flexible_dedup` — флаг, указывающий на использование гибкой дедупликации;
- $Vol_{resource}$ — общий объём данных, подлежащих РК;
- $Size_{block}$ — размер блока данных, используемого для обработки данных во время РК (для пулов типов "File system", "Tape library", "Cloud" размер блока является фиксированным и равен 16384 Б);
- $Size_{hash}$ — размер хеша, используемого для идентификации данных;
- *20* — максимальный размер сериализованной позиции в файле;
- *1* — временная база для вычисления сигнатуры или отправки хешей на сервер;
- $Size_{metadata}$ — это 0.02 × объем ресурса

Программные требования

Программные требования к среде функционирования клиентской части СРК

RuBackup:

- 64-битная операционная система (одна из):
- Windows Server 2012;
- Windows Server 2016;
- Windows Server 2019;
- Windows Server 2022.
- библиотека OpenSSL версия 3.3.0, установленная в директорию `C:\OpenSSL-Win64`;
- пакет Microsoft Visual C++ версия 2015.

9.2.2. Установка

Подготовка к установке

Сетевые настройки

На узле развёртывания клиента резервного копирования, если у вас не задействован DNS-сервер:

1. Откройте системный файл `C:\Windows\system32\drivers\etc\hosts`.
2. Проверьте наличие строки с данными всех узлов серверной части RuBackup (основной сервер, резервный и медиасервер при наличии).

Настройка служебной СУБД PostgreSQL

Для разрешения использования символа `\` выполните следующие действия:

1. Отредактируйте конфигурационный файл `postgresql.conf` на узле служебной базы данных PostgreSQL.
2. Для параметр `standard_conforming_strings` установите значение `on`.
3. Сохраните изменения.

Установка пакета Microsoft Visual C++

Установите пакет *Microsoft Visual C++* [5]:

1. Скачайте пакеты *Microsoft Visual C++* 32- и 64-разрядные версии 2015 с официального сайта *Microsoft*.
2. Запустите поочерёдно загруженные файлы `vc_redist.x86.exe` и `vc_redist.x64.exe`.
3. Следуйте инструкциям установщика.

Установка пакета OpenSSL

Установите библиотеки *OpenSSL* [6] версия 3.3.0:

1. Скачайте дистрибутив *OpenSSL* версии 3.3.0 для 64-разрядной ОС Windows на официальном сайте разработчика.
2. Запустите исполняемый файл `Win64OpenSSl-<version>.exe` и укажите директорию `C:\OpenSSL-Win64`, в которую будет установлено приложение.
3. Пропишите путь к приложению в переменных среды Windows:
   - откройте окно **Панель управления** — **Система и безопасность** — **Система**;
   - выберите **Изменить параметры** — вкладка **Дополнительно**;
   - нажмите кнопку **Переменные среды**;
   - откройте раздел **Системные переменные** в текущем окне;
   - откройте переменную `PATH`;
   - создайте два значения:
     - полный путь к папке, в которую установили приложение `C:\OpenSSL-Win64`;
     - подпапку `C:\OpenSSL-Win64\bin`;
   - нажмите **OK** для сохранения изменений.

Установка пакетов

1. Предварительно скачайте пакет клиента резервного копирования `RuBackup_client_installer<version>.exe`, где <version> — актуальная версия пакета (см. Дистрибутивы).
2. Запустите загруженный файл `RuBackup_client_installer<version>.exe` с правами администратора.
3. Выберите язык интерфейса установщика, примите лицензионное соглашение и начните установку.
4. Для ОС Windows Server версии 2012 и версии 2016: перезагрузите ОС для применения настроек.

9.2.3. Настройка

Настройка клиента РК

Настройку компонентов СРК RuBackup следует произвести на каждом узле в строго приведённом порядке (в зависимости от архитектуры):

1. настройка основного сервера;
2. настройка резервного сервера;

3. настройка медиасервера (выполняется для каждого медиасервера);
4. настройка клиента системы резервного копирования (выполняется для каждого клиента СРК).

Необходимо предварительно настроить сетевое взаимодействие узлов компонентов СРК RuBackup, используя FQDN, имя хоста или IP-адрес (далее по тексту — адрес).

Автономный режим работы клиента резервного копирования в среде функционирования ОС Windows Server не поддерживается.

Настройка клиента РК в терминале (интерактивный режим)

Выполните на каждом узле клиента интерактивную настройку СРК RuBackup с помощью `rb_init` (см. Сценарии настройки клиента).

```
start C:\RuBackup-win-client\bin\rb_init.exe
```

Настройка узла

Добавление исключения в антивирус

1. При использовании антивируса *Windows Defender* необходимо средствами *PowerShell* исключить папку `C:\RuBackup-win-client` из автоматической проверки:

   ```
   Add-MpPreference -ExclusionPath 'C:\RuBackup-win-client'
   ```

2. Для проверки исключений *Windows Defender* выведите полный список исключений:

   ```
   Get-MpPreference | fl excl*
   ```

Добаление в автозапуск

Добавьте сервис клиента РК в автозапуск при загрузке ОС:

1. Откройте **Диспетчер серверов** — **Средства** — **Службы**.
2. Выберите **RuBackup Service** — **Свойства** — вкладка **Общие**.
3. Для параметра **Тип запуска** установите значение **Автоматически** (Рисунок 13).

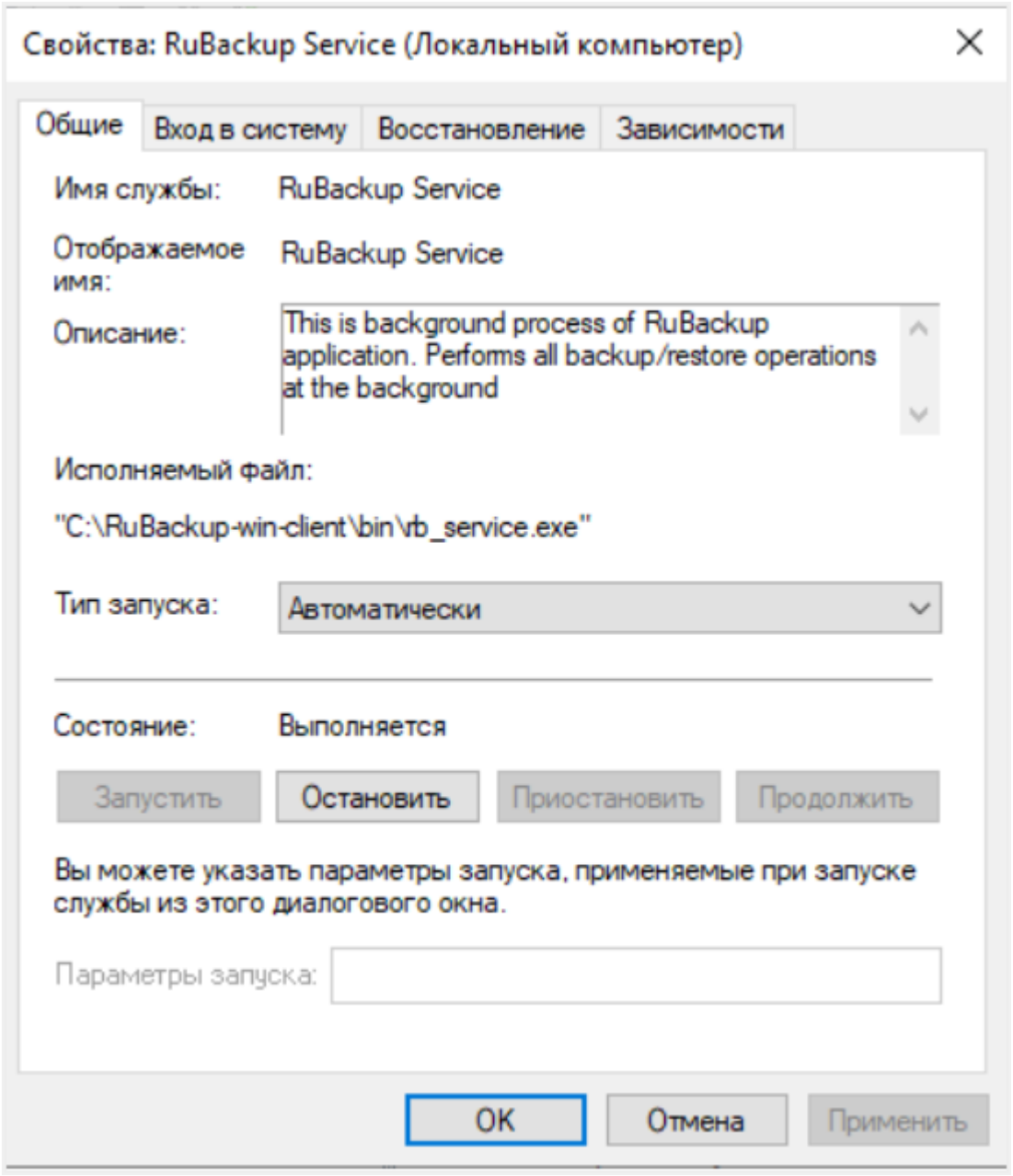


Рисунок 13. Окно «Свойства RuBackup Service»

4. Нажмите **OK** для сохранения изменений.

9.2.4. Запуск

Произведите активацию клиентской части СРК RuBackup, выполнив на каждом узле с развёрнутым клиентом резервного копирования запуск сервиса клиента.

Для успешного запуска клиента РК в клиент-серверном режиме предварительно необходимо запустить серверную часть СРК.

Запуск сервиса клиента

Запустите сервис клиента резервного копирования:

1. Откройте **Диспетчер серверов** — **Средства** — **Службы**.
2. Выберите **RuBackup Service** и запустите его.

[1] Для пула типа "Block device" размера блока может быть задан при создании пула. Значением по умолчанию является 131072 Б. Для получения более подробной информации по настройке пулов обратитесь к секции "Пулы" раздела "Хранилища" Руководства системного администратора RuBackup. Для пулов типов "File system", "Tape library", "Cloud" размер блока является фиксированным и равен 16384 Б. Для всех типов пулов длина ключа хеш-функции зависит от выбранной хеш-функции в настройках пула. Например, для хеш-функции SHA1 длина ключа составляет 20 Б

[2] ** Резервное копирование: объём свободного дискового пространства, составляющий не менее 3% от совокупного объёма данных, резервное копирование которых осуществляется единовременно. Восстановление данных: объем свободного дискового пространства должен быть не менее совокупного объема единовременно восстанавливаемых данных с использованием данного клиента. Многопоточное резервное копирование: объём свободного дискового пространства зависит от выбранных параметров: количества потоков, размера блока и длины хеша. Чем больше используется потоков, тем больше требуемый объём. Чем меньше выбранный размер блока, тем больше требуется доступного пространства на диске. Чем больше длина хеша, тем больше требуется памяти. Расчёт требуемого объёма: Приблизительный расчёт требуемого объёма доступного пространства в многопоточном режиме можно оценить как (worker_parallelism *)% от ресурса. Это означает, что для каждого рабочего потока, который будет использоваться при многопоточной обработке данных, потребуется определённый объём доступного пространства на диске.

[3] Выполните установку актуальной версии репозитория EPEL, для примера приведена установка репозитория EPEL 8

[4] обязательное для заполнения поле (если оно активно)

[5] Подробное описание приведено в официальной документации на программный продукт Microsoft Visual C++

[6] Подробное описание приведено в официальной документации на программный продукт OpenSSL

Глава 10. Результаты установки

10.1. Каталог установки

При установке инсталляционный rpm/deb-пакет будет автоматически распакован в директорию:

- для *Linux-систем*: `/opt/rubackup`.
- для *Windows-систем*: `C:\RuBackup-win-client\`

Структура установленных пакетов СРК RuBackup приведена в таблице.

Таблица 13. Структура установленных пакетов СРК RuBackup

Структурный элемент	Назначение элемента
`/opt/rubackup`	Директория, в которой распакован установочный комплект компонента RuBackup, а также используемые дополнительные инструменты
Пакет rubackup-common	
`/opt/rubackup/keys/client/`	Папка содержит сертификат и закрытый ключ клиента для внутреннего взаимодействия компонентов СРК по протоколу SSL
`/opt/rubackup/keys/server/`	Папка содержит сертификат и закрытый ключ сервера для внутреннего взаимодействия компонентов СРК по протоколу SSL
`/opt/rubackup/keys/rootCA/`	Папка содержит самоподписанный сертификат и закрытый ключ центра сертификации для внутреннего взаимодействия компонентов СРК по протоколу SSL
`/opt/rubackup/etc/`	Папка содержит конфигурационные файлы СРК RuBackup
`/opt/rubackup/etc/ld.so.conf.d/rubackup.conf`	Вспомогательный конфигурационный файл, указывающий ОС путь к дополнительным библиотекам, используемых СРК RuBackup
`/opt/rubackup/copyrights/`	Папка содержит файлы лицензионных соглашений
`/opt/rubackup/rc/icons/`	Папка содержит иконки интерфейса
Пакет rubackup-client	
`/opt/rubackup/etc/systemd/system/`	Папка содержит сервисы СРК RuBakup
`/opt/rubackup/etc/rubackup.lsf`	Файл локального расписания Клиента системы резервного копирования
`/opt/rubackup/etc/systemd/system/rubackup_client.service`	Сервис клиентской части СРК RuBakup

Структурный элемент	Назначение элемента
`/opt/rubackup/scripts/`	Папка содержит скрипты управления СРК RuBackup
`/opt/rubackup/scripts/test-script.sh`	Пример скрипта для выполнения при резервном копировании
`/opt/rubackup/log/`	Папка содержит журналы событий и задач
`/opt/rubackup/man/`	Папка содержит инструкции по использованию утилит
`/opt/rubackup/modules/`	Папка содержит исполнительные модули, поддерживающие резервное копирование и восстановление целевого ресурса (поддерживаемого клиентом СРК)
`/opt/rubackup/modules/rb_module_lvm`	Исполняемый модуль для резервного копирования и восстановления логических томов LVM
`/opt/rubackup/modules/rb_module_filesystem`	Исполняемый модуль резервного копирования файловой системы
`/opt/rubackup/bin/`	Папка содержит консольные утилиты, поддерживаемые на клиенте для управления резервным копированием и восстановлением данных
`/opt/rubackup/bin/rb_schedule`	Утилита клиента RuBackup для просмотра правил глобального расписания клиента в системе резервного копирования
`/opt/rubackup/bin/rb_replicas`	Утилита клиента RuBackup для управления правилами репликации на клиенте. Вы можете просмотреть список всех правил репликации, а также запустить выбранное правило
`/opt/rubackup/bin/rb_health_check`	Утилита клиента RuBackup для проверки конфигурации клиента и его окружения. Выполняется проверка переменных окружения, версии медиасервера. Проверяется подключение клиента к базе данных, серверу, медиасерверу и толстому клиенту
`/opt/rubackup/bin/rubackup_client`	Клиент резервного копирования RuBackup представляет собой фоновое приложение (сервис, демон), запущенное на хосте клиента и взаимодействующее с сервером RuBackup
`/opt/rubackup/bin/rb_init`	Утилита администратора RuBackup для первоначальной настройки клиента сразу после развёртывания пакета исполняемых файлов. Неинтерактивный режим необходим для сценариев массового развертывания

Структурный элемент	Назначение элемента
`/opt/rubackup/bin/rb_archives`	Утилита клиента RuBackup предназначена для просмотра списка резервных копий клиента в системе резервного копирования, создания срочных резервных копий, их удаления, проверки и восстановления. Работает только в том случае, если на клиенте работает служба (сервис, демон) клиента rubackup_client
`/opt/rubackup/bin/rbfd`	Утилита администратора RuBackup для создания и восстановления полных и инкрементальных резервных копий ресурсов в любых файловых системах. Ресурсом может быть файл, каталог или блочное устройство
`/opt/rubackup/bin/rb_tasks`	Утилита клиента RuBackup для просмотра списка задач клиента в системе резервного копирования RuBackup
`/opt/rubackup/bin/rb_client_defined_storages`	Утилита администратора RuBackup для управления клиентскими хранилищами. Вы можете просматривать, добавлять и удалять клиентские хранилища в конфигурации
`/opt/rubackup/rc/`	Папка содержит конфигурационные скрипты программы
`/opt/rubackup/mnt/`	Предоставляется как временная точка монтирования для файловых систем
Пакет rubackup-server	
`/opt/rubackup/etc/systemd/system/`	Папка одержит сервисы СРК RuBackup
`/opt/rubackup/etc/systemd/system/rubackup_server.service`	Сервис Серверной части СРК RuBackup
`/opt/rubackup/man/`	Папка содержит файлы описаний утилит
`/opt/rubackup/log/`	Папка содержит файлы журнала событий
`/opt/rubackup/log/RuBackup.log`	Системный журнал событий, также содержит информацию о лицензии
`/opt/rubackup/log/task.log`	Журналы событий, содержащие события задач СРК
`/opt/rubackup/log/module_.log`	Журналы событий исполняемых модулей
`/opt/rubackup/log/rbfd`	Информация о процессе выполнения создания РК для каждой задачи, которая использует rbfd
`/opt/rubackup/lib/`	Папка содержит библиотеки, необходимые для работы СРК RuBackup
`/opt/rubackup/bin/`	Папка содержит исполняемые файлы для запуска утилит
`/opt/rubackup/bin/rb_modules`	Утилита администратора RuBackup для управления Модулями

Структурный элемент	Назначение элемента
`/opt/rubackup/bin/rb_tape_libraries`	Утилита администратора RuBackup для управления ленточными библиотеками в системе резервного копирования RuBackup. Вы можете просматривать информацию о ленточных библиотеках в серверной группировке RuBackup, синхронизировать ленточную библиотеку с информацией о ней в базе данных,импортировать, экспортировать и перемещать картриджи в ленточной библиотеке, а также производить LTFS форматирование картриджей, находящихся в слотах ленточной библиотеки.
`/opt/rubackup/bin/rb_media_servers`	Утилита администратора RuBackup для управления медиасерверами RuBackup. Вы можете просматривать список медиасерверов, добавлять их, удалять или изменять их описания. медиасервер предназначен для взаимодействия с клиентами при создании, восстановлении и передаче резервных копий
`/opt/rubackup/bin/rb_user_groups`	Утилита администратора RuBackup для управления группами пользователей. Вы можете просматривать группы пользователей, добавлять и удалять их, а также изменять их название и описание
`/opt/rubackup/bin/rubackup_server`	Сервер резервного копирования RuBackup представляет собой системное фоновое приложение (служба, демон), внутри которого одновременно выполняются множество потоков, отвечающих за разные функции системы резервного копирования
`/opt/rubackup/bin/rb_local_filesystems`	Утилита администратора RuBackup для управления хранилищами резервных копий типа Файловая система. Хранилища такого типа должны быть ассоциированы с пулом того же типа
`/opt/rubackup/bin/rb_security`	Утилита RuBackup для работы с журналом событий информационной безопасности
`/opt/rubackup/bin/rb_clients`	Утилита администратора RuBackup для управления клиентами RuBackup. Вы можете просматривать список клиентов, а также добавлять, удалять или изменять их.
`/opt/rubackup/bin/rb_update`	Утилита администратора RuBackup для управления обновлениями баз данных. Создает sql инструкции, позволяющие сделать обновление базы данных
`/opt/rubackup/bin/rb_block_devices`	Утилита администратора RuBackup для управления блочными устройствами

Структурный элемент	Назначение элемента
`/opt/rubackup/bin/rb_global_config`	Утилита администратора RuBackup для управления параметрами глобальной конфигурации серверной группировки RuBackup. Параметры глобальной конфигурации действительны для всех серверов, входящих в кластер серверов RuBackup
`/opt/rubackup/bin/rb_global_schedule`	Утилита администратора RuBackup для управления глобальным расписанием RuBackup. Глобальное расписание состоит из отдельных правил, которые могут выполняться по определённым условиям для определённого ресурса на клиенте системы резервного копирования. Можно просматривать список правил глобального расписания, экспортировать настройки правила в файл и импортировать правило из файла в глобальное расписание, удалять правила из глобального расписания, останавливать функционирование правила или запускать его в работу, а также немедленно создавать задачу на основе правила глобального расписания
`/opt/rubackup/bin/rb_repository`	Утилита администратора RuBackup для доступа к записям репозитория. Позволяет просматривать список резервных копий, удалять и перемещать резервные копии, проверять их целостность и выполнять их репликацию (копирование) в другие пулы. Для выполнения этих действий утилита создаёт соответствующую задачу в главной очереди задач и заканчивает своё выполнение до того момента, как задача будет выполнена
`/opt/rubackup/bin/rb_users`	Утилита администратора RuBackup для управления пользователями. Вы можете просматривать список пользователей, добавлять, удалять и изменять их
`/opt/rubackup/bin/rb_tape_cartridges`	Утилита администратора RuBackup для управления картриджами ленточных библиотек. Вы можете просматривать список картриджей, добавлять, удалять или изменять их. Каждый картридж принадлежит какому-либо пулу типа ленточная библиотека
`/opt/rubackup/bin/rb_inventory`	Утилита администратора RuBackup для внесения в базу данных RuBackup информации о резервных копиях, которые были сделаны вне текущей конфигурации RuBackup, например, в другой серверной группировке RuBackup

Структурный элемент	Назначение элемента
`/opt/rubackup/bin/rb_interoperation`	Утилита администратора RuBackup для управления задачами импорта или экспорта резервных копий между независимыми системами резервного копирования. Вы можете управлять списком систем, для которых существует возможность импорта или экспорта. Добавлять, просматривать, редактировать, удалять, останавливать и запускать правила экспорта или импорта. Также вы сможете проверять очередь задач и удалять выполненные задачи или завершившиеся с ошибкой. У вас будет возможность создать задачу на экспорт резервной копии из репозитория
`/opt/rubackup/bin/rb_clouds`	Утилита администратора RuBackup для просмотра конфигурации, добавления или удаления облаков S3 в системе резервного копирования
`/opt/rubackup/bin/rb_copy2pool`	Утилита администратора RuBackup для управления репликацией. Предоставляет возможность создавать точные копии (реплики) резервных копий для правил резервного копирования и для стратегий резервного копирования
`/opt/rubackup/bin/rb_notifications`	Утилита администратора RuBackup для управления очередью уведомлений. В очереди уведомлений содержатся все актуальные уведомления групп пользователей RuBackup о происходящих в системе событиях. Уведомления могут быть настроены в правилах глобального расписания и в стратегиях
`/opt/rubackup/bin/rb_remote_replication`	Утилита администратора RuBackup для управления непрерывной удалённой репликацией. Непрерывная удалённая репликация состоит из отдельных правил, которые могут выполняться по определённым условиям для определённого ресурса. Можно просматривать список правил непрерывной удалённой репликации, экспортировать настройки правила в файл и импортировать правило из файла, удалять правила, останавливать функционирование правила или запускать его в работу
`/opt/rubackup/bin/rb_pools`	Утилита администратора RuBackup для управления пулами. Вы можете просматривать список пулов, добавлять, удалять и изменять их. Каждый пул принадлежит какому-либо медиасерверу. Пулы используются для группирования устройств хранения резервных копий

Структурный элемент	Назначение элемента
`/opt/rubackup/bin/rb_tl_task_queue`	Утилита администратора RuBackup для управления Очередью задач ленточных библиотек
`/opt/rubackup/bin/rb_block_device_check`	Утилита администратора RuBackup для проверки целостности резервных копий на блочном устройстве
`/opt/rubackup/bin/rb_client_group`	Утилита администратора RuBackup для управления группами клиентов. Вы можете просматривать группы клиентов, добавлять их, удалять или изменять их название и описание. Группировка клиентов может потребоваться в случае необходимости выполнения групповых операций резервного копирования в стратегиях
`/opt/rubackup/bin/rb_bandwidth`	Утилита администратора RuBackup для управления ограничениями пропускной способности при выполнении операций резервного копирования для клиентов или правил глобального расписания. Вы можете установить одно или несколько ограничений пропускной способности для определённого клиента СРК или для какого-либо правила глобального расписания
`/opt/rubackup/bin/rb_task_queue`	Утилита администратора RuBackup для управления главной очередью задач. В очереди задач содержатся все актуальные задачи на создание, восстановление, удаление, перемещение и проверку резервных копий
`/opt/rubackup/bin/rb_cloud_task_queue`	Утилита администратора RuBackup для просмотра задач, которые связаны с облачными операциями. При хранении резервных копий в облаке S3 вам может потребоваться загрузить резервную копию в облако или выгрузить какой-либо из файлов резервной копии из облака
`/opt/rubackup/bin/rb_strategies`	Утилита администратора RuBackup для управления стратегиями
`/opt/rubackup/bin/rb_log_viewer`	Утилита администратора RuBackup для просмотра и управления журналами сообщений
`/opt/rubackup/rc/init/`	Содержит конфигурационные скрипты программы
`/opt/rubackup/mnt/`	Предоставляется как временная точка монтирования для файловых систем
Пакет rubackup-common-gui	
`/opt/rubackup/keys/rbm/`	Папка содержит сертификат и закрытый ключ приложения RBM для внутреннего взаимодействия компонентов СРК по протоколу SSL
`/opt/rubackup/gui/plugins/`	Папка содержит плагины

Структурный элемент	Назначение элемента
`/opt/rubackup/gui/lib/`	Папка содержит библиотеки, используемые графическим приложением RBM
`/opt/rubackup/gui/qml/`	Папка содержит QML-библиотеки, используемые графическим приложением RBM
`/opt/rubackup/gui/rc/`	Папка содержит настройки графического отображения, в т.ч. темы, переводы приложения RBM
`/opt/rubackup/gui/rc/themes/`	Файлы тем приложения RBM

10.2. Сетевые сервисы

В результате настройки компонентов СРК RuBackup будут добавлены необходимые сетевые сервисы в файл `/etc/services`:

- `rubackup-cmd` — сервис обеспечивает командное взаимодействие серверов и клиентов СРК RuBackup;
- `rubackup-lic` — сервис лицензирования;
- `rubackup-media` — сервис обеспечивает взаимодействие между медиасерверами и передачу файлов.

10.3. Конфигурационный файл

Данные, полученные после настройки (с помощью утилиты `rb_init` или `rb_init_gui`), сохраняются в файле:

- для *Linux-систем*: `/opt/rubackup/etc/config.file`;
- для *Windows-систем*: `C:\RuBackup-win-client\etc\config.file.txt`.

Таблица 14. Описание параметров конфигурационного файла

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)
▤		`server-inet-interfaces`	Список сетевых интерфейсов сервера в одну строку через пробел, через которые серверу резервного копирования разрешено взаимодействовать с клиентами	
▤		`dbname`	Имя служебной базы данных	(`rubackup`)
▤		`user`	Пользователь служебной базы данных	(`rubackup`)
▤		`password crypted`	Закодированное значение пароля пользователя служебной базы данных	-

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)
		`host`	FQDN или IP адрес сервера, на котором расположена служебная база данных	Необходима настройка правильного разрешения имен
		`port`	Порт подключения служебной базы данных	(`5432`)

		`server-shutdown_scenario`	Сценарий выключения сервера	`immediately`
				`after-all-tasks`
				`cancel-if-tasks`
				(`cancel-if-tasks`)

		`remote-replication`	Удаленная репликация	`yes`
				`no`
				(`yes`)

		`deduplication-task-memory`	Исключение дублирующих копий повторяющихся данных	(`268435456`)
		`parallelizm`	Количество параллельных нитей сетевого асинхронного сервера RuBackup	`1-4096` `8`

		`use-product-uuid`	Для версии СРК RuBackup 2.1 и более поздней: Генерировать идентификатор *hardware id* узла лицензируемого сервера на основании:
			• *для ОС Linux*: идентификатора UUID материнской платы, установленного производителем платы, и закодированной информации в DMI BIOS;
			• *для ОС Windows*: имени узла hostname;
			Для версии СРК RuBackup 2.0 и ранее: параметра нет, `hardware id` генерируется на основании идентификатора `/etc/machine-id` и имени узла `/etc/hostname`
				`false`
				`true`
				(`false`)

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)
✓		`parallelizm_media`	(Медиасервер) Количество параллельных потоков сетевого асинхронного медиасервера RuBackup	`1-4096` (`8`)
✓	✓	`centralized-recovery`	Централизованное восстановление данных из резервной копии с помощью приложения «Менеджер администратора RuBackup» (используемой на любом узле). В случае, если централизованное восстановление отключено, то выполнить восстановление возможно только на клиенте резервного копирования с помощью утилиты командной строки rbfd или «Менеджера клиента RuBackup»	`yes` `no` (`yes`)
✓	✓	`node`	Тип узла RuBackup	`primaryserver` `secondaryserver` `mediaserver` `client`
✓	✓	`who-is-primary-server`	Имя узла основного сервера RuBackup	Необходима настройка правильного разрешения имен
✓	✓	`who-is-secondary-server`	Имя узла резервного сервера RuBackup	Необходима настройка правильного разрешения имен
✓	✓	`logfile`	Расположение системного файла журнала событий	`<path>`
✓	✓	`used-ip-version`	Укажите, какие публичные имена будут использованы DNS-сервером	`ipv4` `ipv6` `both`
✓	✓	`client-hello-timeout`	Время ожидания ответа от сервера на HELLO сообщение, отправленное при запуске задачи от клиента. Задается в секундах.	`> 0` (`240`)

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)

✓	✓	`use-ip-instead-hostname`	Использовать ip адрес вместо hostname для разрешения связи между элементами СРК	`false`
				`true`
				(`false`)

	✓	`use-local-backup-directory`	Каталог для временного хранения резервных копий. Для создания резервных копий и хранения временных файлов, которые создаются при их восстановлении, требуется определённое дисковое пространство. Рекомендуем выделить для этой цели отдельный диск или устройство хранения достаточного размера и примонтировать его к `/backup-tmp` либо к другой точке монтирования. Точку монтирования временного каталога нужно указать как значение параметра `use-local-backup-directory` и перезагрузить клиент RuBackup.	`<path>` (`/tmp`)
	✓	`client-inet-interface`	Сетевой интерфейс клиента. Используется для отображения дополнительной информации о клиенте в СРК RuBackup. Медиасервер осуществляет связь с основным или резервным сервером, а также с клиентской утилитой rbfd через сетевой интерфейс, указываемый в этом параметре	
	✓	`parallel-tasks`	Максимальное количество одновременно выполняемых задач	`1-64` (`2`)
	✓	`rbd_algorithm`	Выбор хэш функции при дедупликации	`sha1`, `sha2`, `skein`, `streebog`, `blake2b` (`sha2`)

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)
	🖵	`rbd_block_size`	Размер блока данных при дедупликации, байт	8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, по умолчанию — 16384

	🖵	`rbd_hash_length`	Допустимая длина хэша	`256`
				`512`
				(`256`)

	🖵	`client-shutdown_scenario`	Сценарий выключения клиента	`immediately`
				`after-all-tasks`
				`cancel-if-tasks`
				(`cancel-if-tasks`)

	🖵	`reconnect-period-count`	Количество периодов переподключения	> `0` (`3`)
	🖵	`reconnect-period-timeout`	Таймаут между периодами переподключения	> `0` (`20` секунд)
	🖵	`reconnect-count`	Количество попыток переподключения в рамках одного периода	> `0` (`3`)
	🖵	`reconnect-timeout`	Таймаут между попытками переподключения в рамках одного периода	> `0` (`5` секунд)

	🖵	`digital-signature`	Использовать электронно-цифровую подпись	`yes`
				`no`
				(`yes`)

Сервер	Клиент	Параметр	Назначение	Допустимые значения (по умолчанию)
	🖵	`digital-sign-hash`	Хеш-функция для электронно-цифровой подписи	В соответствии с командами группы OpenSSL Message Digest (`openssl help`) (`sha1`)

	🖵	`monitoring-client`	Мониторинг состояния системы	`yes`
				
				`no`
				
				(`yes`)

	🖵	`memory-threshold`	Снижение потребления оперативной памяти при полном резервном копировании. Для хранения уникальных хешей и обеспечения дедупликации нужно выделить на диске дополнительное место ≈0,3% от размера ресурса. *Ограничения* При использовании параметра в кластерной группе убедитесь, что все клиенты группы имеют одну версию СРК. Параметр используется только для создания полной резервной копии	Не менее 4 ГБ и не более значения свободной оперативной памяти в системе. Значение параметра не гарантирует точность верхней границы потребления памяти. Для выключения параметра можно задать его равным `0` или удалить его из файла настроек.

Глава 11. Настройка ограничения на количество открытых файловых дескрипторов на узле сервера RuBackup

При увеличении количества входящих соединений (если число клиентов/медиа серверов в группировке растёт и/или на клиентах включена функция многопоточной передачи данных) сервер RuBackup может достичь предела выделенных лимитов на открытые файловые дескрипторы. Сетевые соединения также используют файловые дескрипторы.

Ограничения на количество открытых файловых дескрипторов устанавливает администратор узла, на котором запущен сервер RuBackup. Достижение этого ограничения приводит к ошибкам при выполнении резервного копирования/восстановления. Иногда сервер RuBackup может аварийно завершить работу.

11.1. Зависимость количества файловых дескрипторов

В зависимости от способа запуска сервера RuBackup, максимальное число (лимит) открытых дескрипторов будет разным.

Чтобы рассчитать необходимое количество файловых дескрипторов, учтите следующее:

- В режиме простоя сервер использует около 100 файловых дескрипторов.
- Каждый подключённый клиент РК или медиасервер добавляет по два открытых файловых дескриптора на сервере.
- Выполнение любой задачи на стороне клиента РК при выключенном параметре `network_parallelism` [1] требует двух дополнительных файловых дескриптора на сервере.
- При активированном параметре `network_parallelism` клиент РК открывает *N* соединений к серверу, где N — значение, заданное для этого параметра. В рамках каждого сетевого соединения, как правило, на стороне сервера требуется запросить информацию из базы данных, поэтому требуемое число открытых файловых дескрипторов будет *N*2*.

11.2. Расчёт необходимого количества файловых дескрипторов

Проверьте, рассчитав по формулам, число нужных вам файловых дескрипторов и убедитесь, что на узле сервера RuBackup их достаточно.

Пример 6. Общая формула для расчёта необходимого количества файловых дескрипторов:

где:

- МС — число медиасерверов;
- КЛ — число клиентов;
- N — значение, заданное для сетевого параллелизма, параметра network_parallelism. Если сетевой параллелизм выключен, то N=2.

Пример расчета 1

Рассмотрим пример расчёта необходимого количества файловых дескрипторов для системы, состоящей из одного сервера RuBackup, двух медиасерверов и 50 клиентов. Предположим, что сетевой параллелизм отключён.

Необходимое количество файловых дескрипторов рассчитывается следующим образом:

```
100 (для основного сервера)+2*2 (для медисерверов)+50*2 (для клиентов в
простое)+50*2 (для клиентов с задачами одновременно) = 304
```

Таким образом, общее количество необходимых файловых дескрипторов составляет 304.

Стандартное значение лимита — 1024, будет достаточным.

Пример расчета 2

Рассмотрим пример расчёта необходимого количества файловых дескрипторов для системы, состоящей из одного сервера RuBackup, двух медиасерверов и 50 клиентов. Предположим, что сетевой параллелизм включён со значением 40.

Необходимое количество файловых дескрипторов рассчитывается следующим образом:

```
100 (для основного сервера) + 2*2 (для медиасерверов) + 50*2 (для клиентов в
простое) + 50*40 (для всех клиентов с задачами одновременно)= 2204
```

Таким образом, общее количество необходимых файловых дескрипторов составляет 2204.

Стандартное значение лимита в 1024 будет недостаточным для такой системы, поэтому рекомендуется увеличить лимит. Желательно установить лимит в 3000

файловых дескрипторов для запаса.

11.3. Способы настройки ограничения количества открытых файловых дескрипторов

Настройка ограничения количества открытых файловых дескрипторов производится в зависимости от способа запуска сервера.

11.3.1. Настройка ограничения количества открытых файловых дескрипторов при ручном запуске сервера

Для настройки ограничения количества открытых файловых дескрипторов при ручном запуске сервера:

1. Остановите сервер (в случае ручного запуска сервера):

   ```
   rubackup_server stop
   ```

2. Для проверки текущего лимита выполните:

   ```
   sudo ulimit -n
   ```

   По умолчанию ограничение количества открытых файловых дескрипторов установлено 1024 файла.

3. Изменение лимита открытых файловых дескрипторов возможно выполнить для текущей сессии пользователя `root` или установить постоянное значение.

   a. Для временного изменения лимита открытых файловых дескрипторов только в текущей сессии пользователя `root` необходимо выполнить команду:

      ```
      ulimit -n N
      ```

      где `N` — это желаемое значение лимита открытых файловых дескрипторов.

      Внесённые изменения будут отменены после завершения текущей сессии.

   b. Для установки постоянного лимита открытых файловых дескрипторов:

      - отредактируйте файл `/etc/security/limits.conf`:

        ```
        sudo nano /etc/security/limits.conf
        ```

и добавив строки:

```
root hard nofile N
root soft nofile N
```

где `N` — это желаемое значение лимита открытых файловых дескрипторов;

- сохраните изменения;
- завершите сессию и откройте новую сессию;
- проверьте значение лимита открытых файловых дескрипторов:

```
ulimit -n
```

4. Перезапустите сервер:

```
rubackup_server start
```

11.3.2. Настройка ограничения количества открытых файловых дескрипторов при запуске сервисов сервера

Для настройки ограничения количества открытых файловых дескрипторов при запуске сервисов сервера:

1. Для проверки текущего лимита выполните:

```
sudo ulimit -n
```

По умолчанию ограничение количества открытых файловых дескрипторов задаётся в службе `systemd` и стандартное значение — 1024 файла.

2. Для изменения лимита открытых файловых дескрипторов:
   - откройте файл `/etc/systemd/system/rubackup_server.service`:

```
sudo nano /etc/systemd/system/rubackup_server.service
```

   - отредактируйте секцию [`Service`], добавив строку:

```
LimitNOFILE=N
```

где `N` — это желаемое значение лимита открытых файловых дескрипторов;

- сохраните изменения.

3. Загрузите обновленный конфигурационный файл сервиса в службу `systemd`:

```
systemctl daemon-reload
```

4. Перезапустите сервис сервера RuBackup:

```
systemctl stop rubackup_server

systemctl start rubackup_server
```

[1] Параметр задает количество потоков, которые будут передавать блоки данных на медиасервер