



RuBackup

**Система резервного копирования
и восстановления данных**

RUPOST

ВЕРСИЯ 2.8.0.0.0, 30.12.2025

Содержание

1. Назначение	3
2. Требования к программным средствам	4
3. Резервируемые данные	5
4. Типы резервного копирования	6
5. Типы восстановления данных	7
6. Способы восстановления данных	8
7. Комплект поставки	9
8. Ограничения	10
9. Установка модуля	11
9.1. Подготовка к установке модуля	11
9.2. Установка	11
9.3. После установки	12
9.4. Критерий успешности установки	12
10. Подготовка к работе	14
10.1. Подготовка холодного хранилища	14
10.2. Подготовка СУБД <i>PostgreSQL</i>	15
10.2.1. Создание и настройка прав пользователя СУБД	15
10.2.2. Доступ пользователя к СУБД	16
10.2.3. Режим архивирования журналов транзакций	17
11. Работа с данными	19
11.1. Резервное копирование	20
11.1.1. Создание правила глобального расписания	20
11.1.2. Регулярное резервное копирование	21
11.1.3. Срочное резервное копирование по правилу	21
11.1.4. Срочное резервное копирование (без правила)	22
11.2. Восстановление данных	23
11.2.1. Централизованное восстановление резервных копий с помощью ПО Менеджер Администратора <i>RuBackup</i> (RBM)	23
Полное восстановление	23
Гранулярное восстановление	24
11.2.2. Восстановление на целевом ресурсе	25
Восстановление почтовых ящиков пользователей	25
Восстановление архивов почтовых ящиков пользователей	25
Восстановление удаленных писем пользователей	26
Восстановление базы данных	26

11.3. Просмотр очереди задач	26
12. Приложения	27
12.1. Файл настроек модуля	27
12.2. Файл настроек хранилищ	29
12.3. Тонкие настройки модуля RuPost для резервного копирования	31
12.4. Тонкие настройки модуля RuPost для восстановления резервной копии. .	32

Глава 1. Назначение

Модуль *RuPost* позволяет СРК *RuBackup* выполнять резервное копирование и восстановление компонентов корпоративной почтовой системы *RuPost*.

Глава 2. Требования к программным средствам

Для выполнения резервного копирования и восстановления почтовой системы *RuPost* версии 3.2, 3.3 или 3.4 с помощью модуля *RuPost* на узле необходимо предустановленное программное обеспечение:

- 64-битная операционная система *Astra Linux SE 1.7* или *Astra Linux SE 1.8*;
- Клиент резервного копирования *RuBackup*;
- Модуль *RuPost*;
- Служебная база данных почтовой системы *RuPost* (опционально, если планируется делать резервное копирование БД);



Если служебная БД имеет конфигурацию *Patroni*, клиент РК должен быть установлен на каждом хосте кластера *Patroni*.

- NFS-клиент для монтирования холодного^[1] хранилища почтовой системы *RuPost*.

Для управления резервным копированием и восстановлением почтовой системы *RuPost* рекомендуем использовать программное обеспечение *Менеджер Администратора RuBackup (RBM)* (см. [Менеджер администратора RuBackup \(RBM\)](#)).

[1] Резервное хранилище почтовой системы *RuPost*, которое используется как источник данных для резервного копирования компонентов почтовой системы.

Глава 3. Резервируемые данные

Резервное копирование выполняется для:

- компонентов холодного^[1 - Глава 2] хранилища почтовой системы:
 - почтовые ящики;
 - архивы почтовых ящиков;
 - удаленные письма;
 - календари и расписания.
- служебной базы данных почтовой системы.

Глава 4. Типы резервного копирования

Модуль *RuPost* поддерживает полное, инкрементальное и дифференциальное резервное копирование почтовой системы *RuPost*.

При инкрементальном резервном копировании создаются:

- инкрементальная резервная копия компонентов из холодного^[1 - Глава 2] хранилища;
- полная резервная копия служебной базы данных почтовой системы *RuPost*.

При дифференциальном резервном копировании создаются:

- дифференциальная резервная копия компонентов из холодного^[1 - Глава 2] хранилища;
- полная резервная копия служебной базы данных почтовой системы *RuPost*.

Подробнее о типах резервного копирования см. в [Термины](#).

В этом документе приведены инструкции по созданию РК с помощью программы *Менеджер Администратора RuBackup* (см. [Раздел 11.1](#)).

Глава 5. Типы восстановления данных

СРК *RuBackup* поддерживает полное и гранулярное восстановление почтовой системы *RuPost*.

Полное восстановление доступно для компонентов холодного^[1 - Глава 2] хранилища почтовой системы *RuPost* и для служебной БД почтовой системы *RuPost*.

Гранулярное восстановление доступно только для компонентов холодного^[1 - Глава 2] хранилища почтовой системы *RuPost*.

Для каждого отдельного пользователя почтовой системы *RuPost* можно восстановить:

- почтовый ящик;
- отдельные папки с письмами из почтового ящика;
- отдельные письма из папок за указанный промежуток времени.

Подробнее о типах восстановления данных см. в [Термины](#).

Глава 6. Способы восстановления данных

Модуль *RuPost* поддерживает следующие способы восстановления резервных копий:

- централизованное восстановление данных без развертывания. Резервная копия будет распакована в каталог распаковки;
- централизованное восстановление данных с развертыванием. Компоненты почтовой системы будут распакованы в холодное^[1 - Глава 2] хранилище, минуя временный каталог.



СРК не восстанавливает компоненты почтовой системы *RuPost* напрямую в почтовую систему. Данные восстанавливаются только в папки **Restore** холодного^[1 - Глава 2] хранилища. Дальнейшее развертывание в почтовую систему осуществляется силами администратора *RuPost Server*.

Служебная база данных будет распакована во временный каталог распаковки, а затем развернута в почтовой системе *RuPost*.



Развёртывание базы данных возможно только в конфигурации *standalone*. В кластере *Patroni* подмену базы и перезапуск кластера осуществляет администратор базы данных.

В этом документе приведены инструкции по восстановлению РК с помощью программы *Менеджер Администратора RuBackup* (см. [Раздел 11.2](#)).

Глава 7. Комплект поставки

Дистрибутив модуля *RuPost* СРК *RuBackup* поставляется в виде deb-пакета с именем `rubackup-rupost-<version>_amd64.deb`, где `<version>` - номер версии поставляемого модуля.

Глава 8. Ограничения

- СРК не обеспечивает синхронность данных при резервном копировании данных из холодного^[1 - Глава 2] хранилища и соответствующих метаданных в служебной БД *RuPost*. При восстановлении РК возможна ситуация, когда данные в служебной БД *RuPost* являются более новыми и не соответствуют данным, находящимся в холодном хранилище.
- Запуск резервного копирования осуществляется вручную. Время начала резервного копирования устанавливается таким образом, чтобы к моменту старта резервного копирования *RuPost* завершил репликацию данных в холодное^[1 - Глава 2] хранилище.
- Модуль *RuPost* не поддерживает взаимодействие с СРК при помощи веб-приложения *Tusana*.
- Не поддерживается взаимодействие с СРК с помощью утилит командой строки.
- Резервные копии компонентов почтовой системы *RuPost* текущей версии могут быть восстановлены в будущие версии почтовой системы при условии обратной совместимости данных версий почтового клиента *RuPost*.
- Со стороны СРК не предусмотрена возможность контроля объема свободного места в каталогах временного хранилища резервных копий и в каталогах распаковки.
- Для хранения резервных копий допускаются только файловые хранилища и блочные устройства.
- Модуль *RuPost* не поддерживает резервное копирование и восстановление СУБД *PostgreSQL Pro*.
- Если после резервного копирования статус письма был изменен с *непрочитанное* на *прочитанное*, при восстановлении статус будет соответствовать сохраненному в резервной копии, то есть *непрочитанное*.

Глава 9. Установка модуля

9.1. Подготовка к установке модуля

На узле, где будет произведена установка модуля для резервного копирования данных почтовой системы *RuPost*, предварительно должен быть развёрнут (см. [Установка пакетов](#)) и настроен (см. [Настройка клиента РК](#)) клиент резервного копирования, подключенный к основному серверу СРК *RuBackup*.

Если вы планируете восстановление почтовой системы через *Менеджер администратора RuBackup (RBM)*, включите на узле клиента резервного копирования функцию централизованного восстановления.

Функция централизованного восстановления управляется параметром `centralized-recovery` конфигурационного файла `/opt/rubackup/etc/config.file` на узле клиента:

Включение централизованного восстановления (`/opt/rubackup/etc/config.file`)

```
centralized-recovery  yes
```

9.2. Установка

На подготовленном узле клиента резервного копирования:

1. Остановите сервис клиента резервного копирования *RuBackup*:

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

2. Выполните установку модуля *RuPost* СРК *RuBackup*:

```
sudo dpkg -i rubackup-rupost-<version>_amd64.deb
```

где `<version>` — номер версии модуля *RuPost* СРК *RuBackup*.

В ходе установки модуля выполняются распаковка и настройка пакета модуля *RuPost* `rubackup-rupost`.

В результате установки пакета модуля *RuPost* в систему будут добавлены файлы, приведенные в [таблице](#):

Таблица 1. Перечень устанавливаемых в систему файлов

Расположение	Назначение
<code>/opt/rubackup/etc/rb_module_rupost.conf</code>	Файл настроек модуля <i>RuPost</i> CPK <i>RuBackup</i>
<code>/opt/rubackup/modules/rb_module_rupost</code>	Исполняемый файл модуля <i>RuPost</i> CPK <i>RuBackup</i>

9.3. После установки

После установки модуля:

1. Настройте параметры в файле настроек `/opt/rubackup/etc/rb_module_rupost.conf` модуля *RuPost* (см. [Раздел 12.1](#)).
2. Перезапустите сервис клиента резервного копирования *RuBackup*:

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

9.4. Критерий успешности установки

Об успешной установке и настройке модуля *RuPost* свидетельствует запись о его успешной проверке клиентом резервного копирования в журнале событий `/opt/rubackup/log/RuBackup.log`:

Пример записи об успешной установке и настройке модуля (если служебная БД доступна)

```
...
Module version: 2.6.0
RuPost version: "3.4.0"
Database availability: "Available"
```



Даже если служебная база данных почтовой системы *RuPost* недоступна (проблемы с сетью или указаны некорректные данные для подключения к служебной БД в конфигурационном файле `/opt/rubackup/etc/rb_module_rupost.conf`), можно сделать резервную копию компонентов холодного^[1 - Глава 2] хранилища почтовой системы *RuPost*.

Пример записи об успешной установке и настройке модуля (если служебная БД недоступна)

```
...
Module version: 2.6.0
RuPost version: "3.4.0"
Database availability: "Unavailable"
```



Если холодное^[1 - Глава 2] хранилище почтовой системы *RuPost* недоступно, то модуль не будет запущен.

Если в журнале событий `/opt/rubackup/log/RuBackup.log` администратор СРК видит ошибку, сообщающую о неправильной конфигурации модуля *RuPost*, проверьте настройки в файле `/opt/rubackup/etc/rb_module_rupost.conf`, выполнив на узле клиента резервного копирования команду:

```
/opt/rubackup/modules/rb_module_rupost -t
```

Если ошибка не поддается анализу, то обратитесь в сервис технической поддержки RuBackup с предоставлением всей необходимой информации по возникшей проблеме на официальном сайте <https://support.rubackup.ru/>.

Глава 10. Подготовка к работе

С точки зрения резервного копирования и восстановления почтовая система *RuPost* состоит из СУБД *PostgreSQL* и холодного^[1 - Глава 2] хранилища писем.

Модуль должен иметь доступ к холодному^[1 - Глава 2] хранилищу.

(опционально) Модуль должен располагаться на узле с СУБД *PostgreSQL* только в том случае, если необходимо осуществлять резервное копирование служебной базы данных почтовой системы *RuPost*. Если СУБД развёрнута в кластере *Patroni*, то модуль должен быть установлен и настроен на каждом узле кластера.

10.1. Подготовка холодного хранилища

Холодное^[1 - Глава 2] хранилище писем должно быть примонтировано к узлу с модулем с помощью NFS, для этого:

1. На узле с модулем установите клиент NFS:

```
sudo apt update

sudo apt install nfs-common
```

2. Примонтируйте холодное^[1 - Глава 2] хранилище:

```
sudo mount -t nfs ip_address:/nfs_src_folder destination_folder
```

где:

- `ip_address` - это IP-адрес узла с холодным хранилищем писем;
- `nfs_src_folder` - исходное расположение холодного хранилища писем;
- `destination_folder` - точка монтирования холодного хранилища писем на узле с модулем.

3. Чтобы после перезагрузки узла с модулем холодное хранилище оставалось примонтированным, добавьте в файл `/etc/fstab` строку:

```
ip_address:/nfs_src_folder destination_folder nfs auto 0 0
```

4. В файле `настроек модуля` укажите путь до холодного^[1 - Глава 2] хранилища:

```
cold_storage destination_folder
```

10.2. Подготовка СУБД PostgreSQL

Подготовка СУБД PostgreSQL к выполнению резервного копирования с помощью модуля *RuPost* включает в себя:

1. Создание и настройку пользователя СУБД для безопасного выполнения резервной копии PostgreSQL.
2. Доступ пользователя к СУБД PostgreSQL.
3. Настройку режима архивирования журналов транзакций.

10.2.1. Создание и настройка прав пользователя СУБД

Если для доступа к СУБД используется учетная запись не администратора СУБД, то для создания базовой резервной копии пользователь должен обладать правами на вызов функций начала и окончания резервного копирования экземпляра СУБД.

1. Вызовите `psql` от имени администратора СУБД:

```
sudo -u postgres psql
```

2. Создайте в СУБД пользователя `rubackup_backuper`.

```
create user rubackup_backuper password '12345'; ❶
alter role rubackup_backuper with login;
```

- ❶ Вместо `12345` укажите желаемый пароль.

3. Настройте права пользователя в СУБД для доступа.

Пример 1. Предоставление прав на выполнение функций в PostgreSQL версии 15 и выше

```
grant execute on function pg_backup_start(text,bool) to
rubackup_backuper; ❶
grant execute on function pg_backup_stop(bool) to rubackup_backuper; ❶
grant execute on function pg_switch_wal() to rubackup_backuper;
grant pg_read_all_settings to rubackup_backuper;
```

- ❶ В PostgreSQL версии 14 и ниже используются функции `pg_stop_backup` и `pg_start_backup`, а в версии 15 и выше — `pg_backup_stop` и `pg_backup_start`.

В файле `настроек модуля` необходимо указать имя созданного пользователя (`username`) и его пароль (`password`).

10.2.2. Доступ пользователя к СУБД

Правила доступа к СУБД задаются в файле `pg_hba.conf`.

Файл `pg_hba.conf`:

- устанавливает метод подключения (`local` — локально, `host` — по TCP/IP);
- ограничивает доступ базой данных или задачей (`replication`);
- предоставляет доступ пользователю с заданных адресов указанным методом аутентификации.



В зависимости от версии СУБД *PostgreSQL* месторасположение файла может отличаться.

Добавьте в файл `pg_hba.conf` строки, указывающие на право пользователя `rubackup_backuper` на локальный (`localhost`, `127.0.0.1`) доступ по TCP/IP к СУБД для резервного копирования и репликации. Закомментируйте строки, предоставляющие доступ неограниченному кругу лиц (`host all all`).

```
host    backupdb    rubackup_backuper  127.0.0.1  md5
host    replication rubackup_backuper  127.0.0.1  md5
```

Пример 2. Пример файла `pg_hba.conf`

```
# 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

# IPv6 local connections
# host    all       all            ::1/128       md5

# Allow replication connections from localhost, by a user with the
# replication privilege
local  replication all            md5
host    replication all            127.0.0.1/32  md5
host    replication all            ::1/128       md5
host    backupdb    rubackup_backuper  127.0.0.1/32  md5
host    replication rubackup_backuper  127.0.0.1/32  md5
```

Если вы хотите подключаться из локальной сети (192.168.*.*) непосредственно к СУБД, добавьте в `pg_hba.conf` строку:

```
host    any    any    192.168.1.1/32    md5
```



Для доступа к СУБД необходимо открыть порт 5432 (или иной заданный в `postgresql.conf`) в фаерволе узла клиента. Пользователь, для которого требуется доступ к СУБД, должен существовать и иметь необходимые полномочия.



В файле настроек модуля необходимо указать абсолютный путь для используемой версии PostgreSQL (`pg_ctl`).

10.2.3. Режим архивирования журналов транзакций

СУБД на основе PostgreSQL хранят журналы транзакций. Архивирование и резервное копирование этих журналов позволяет восстановить БД из любого неконсистентного состояния путем последовательного применения сохраненных транзакций.

Режим архивирования журналов транзакций настраивается в файле `postgresql.conf`.



В зависимости от версии СУБД PostgreSQL месторасположение файла может отличаться.

Чтобы включить режим архивирования журналов транзакций в `postgresql.conf`:

1. установите параметру `wal_level` значение `replica`;
2. установите параметру `archive_mode` значение `yes` или `always`;
3. задайте команды сохранения (`archive_command`) и восстановления (`restore_command`) журналов транзакций.

Пример 3. Настройки архивирования журналов транзакций в `postgresql.conf`

```
wal_level = replica
archive_mode = on
archive_command = 'cp %p /opt/rubackup/mnt/postgresql_archives/%f'
restore_command = 'cp /opt/rubackup/mnt/postgresql_archives/%f %p'
```

Допускается сжатие архивных файлов транзакций во время архивирования или восстановления из архива.

Пример 4. Команды сжатия и распаковки архивных журналов транзакций

```
archive_command = 'gzip < %p >  
/opt/rubackup/mnt/postgresql_archives/%f.gz'  
restore_command = 'gunzip < /opt/rubackup/mnt/postgresql_archives/%f.gz >  
%p'
```

Путь к архиву в файле настроек модуля должен совпадать с путём, указанным в конфигурационном файле СУБД PostgreSQL (параметр `archive_command`). В [файле настроек](#) модуля (параметр `archive_catalog`) укажите на папку, хранящую архив журналов транзакций,

Глава 11. Работа с данными

Доступно полное или инкрементальное резервное копирование. При выборе инкрементального типа резервного копирования необходимо наличие полной резервной копии, в случае её отсутствия будет выполнено полное резервное копирование. Создание полной резервной копии более длительное, чем создание инкрементальной резервной копии.

При восстановлении инкрементальной резервной копии происходит распаковка архивов в несколько итераций, начиная с полной резервной копии. Поэтому при восстановлении компонентов холодного^[1 - Глава 2] хранилища почтовой системы в холодном хранилище должно быть достаточно свободного места для распаковки каждой из резервных копий в цепочке.

При восстановлении базы данных на узле, где установлен модуль *RuPost*, должно быть достаточно свободного места для распаковки восстанавливаемой базы данных. При восстановлении без развертывания в каталоге, указанном пользователем, должно быть достаточно свободного места для распаковки каждой из резервных копий в цепочке. Так, например, если ресурс на момент создания инкрементальной резервной копии имеет размер 5 ТБ, то в каталоге для распаковки должно быть не менее 5 ТБ+10% свободной памяти.

Схема процесса резервного копирования и восстановления данных почтовой системы *RuPost* приведена на [рисунке](#).

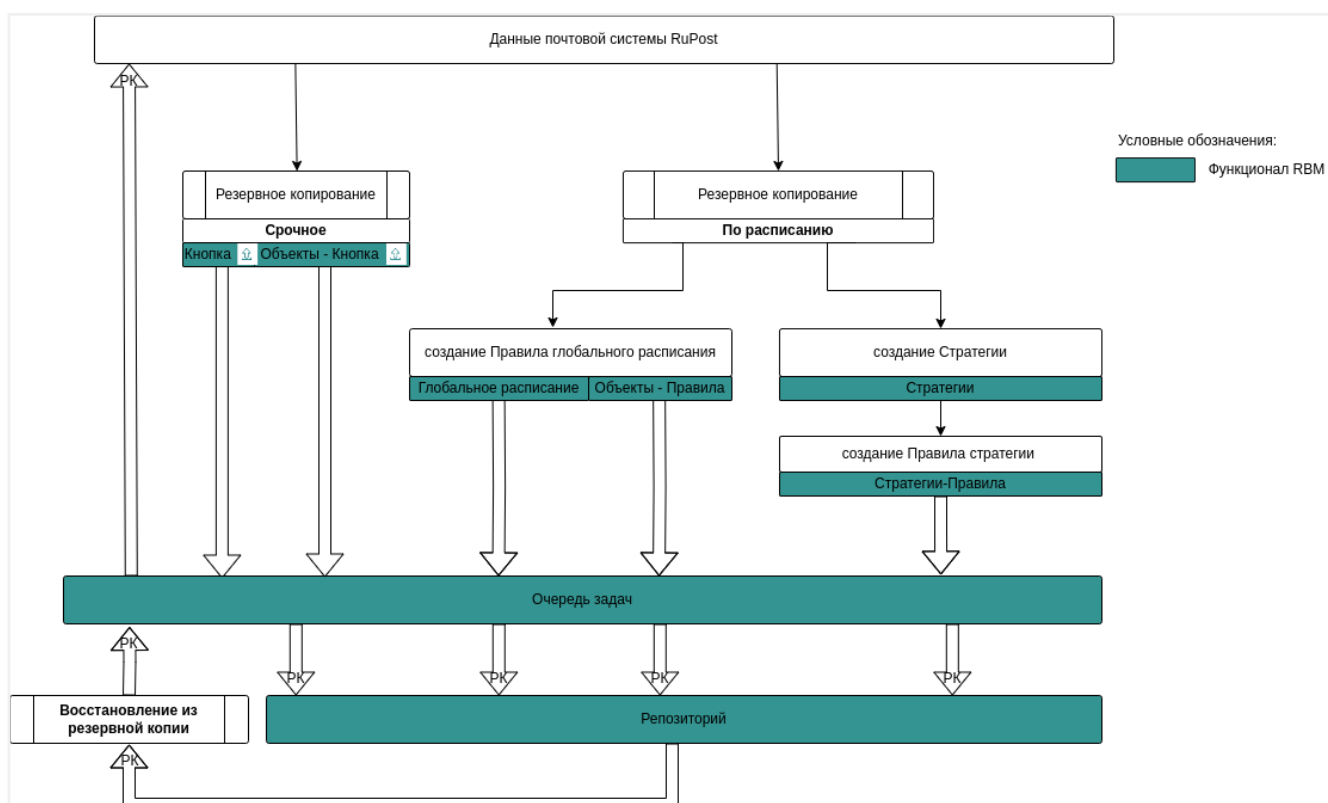


Рисунок 1. Структурная схема резервного копирования и восстановления данных из резервной копии

11.1. Резервное копирование

Подробную информацию о выполнении резервного копирования с помощью ПО *Менеджер Администратора RuBackup (RBM)* см. в [Менеджер администратора RuBackup \(RBM\)](#).

11.1.1. Создание правила глобального расписания

Для выполнения регулярного (периодического) или срочного (разового) резервного копирования по правилу необходимо создать правило в глобальном расписании.

Правило содержит в себе настройки, необходимые для резервного копирования требуемого ресурса.



Для резервного копирования служебной базы данных и компонентов холодного^[1 - Глава 2] хранилища почтовой системы RuPost необходимо создавать отдельные правила глобального расписания.

1. Перейдите в раздел **Глобальное расписание** и нажмите (**Добавить**). Откроется форма создания нового правила глобального расписания.
2. Введите имя создаваемого правила в **Параметры правила**.
3. Из списка **Клиент** выберите узел клиента, на котором установлена почтовая система *RuPost*.
4. Из списка **Тип ресурса** выберите тип резервируемого ресурса **RuPost**. Для выбора резервируемого хранилища почтовой системы нажмите **[...]** и определите *тонкие настройки* модуля *RuPost* (см. [Раздел 12.3](#)).
5. Нажмите **[...]** в поле **Ресурс** и выберите из списка резервируемый компонент почтовой системы.
6. Из списка **Тип РК** выберите тип резервной копии. Возможные значения:
 - полная,
 - инкрементальная;
 - дифференциальная.
7. При необходимости задайте дополнительные параметры правила (см. [Добавление глобального расписания](#)), в том числе периодичность его выполнения.
8. Нажмите **Применить** для сохранения правила.

Созданное правило появится в списке правил:

- раздел **Объекты** → вкладка **Правила**,
- раздел **Глобальное расписание**.

Если при создании правила установлен флаг **Включить после создания** , созданное правило будет иметь статус *run*.

Если при создании правила флаг **Включить после создания**  снят, созданное правило будет иметь статус *wait*.

11.1.2. Регулярное резервное копирование

Регулярное резервное копирование выполняется для каждого *включенного* правила в соответствии с заданной этим правилом периодичностью.

Правило в статусе *run* ожидает выполнения в заданную в нём дату и время.

Правило в статусе *wait* приостановлено и не будет выполняться.

1. Перейдите в раздел  **Глобальное расписание**.
2. Если нужное правило резервного копирования есть в списке, проверьте его статус.

Если правило в статусе *run*, ожидайте его выполнения.

Если правило в статусе *wait*, выделите его в списке и установите на панели инструментов флаг **Включить** , или выберите **Выполнить** в контекстном меню этого правила. Правило перейдет в статус *run*.

3. Если правило с нужными настройками отсутствует в списке правил, создайте правило (см. [Раздел 11.1.1](#)). Убедитесь, что в создаваемом правиле установлен флаг **Включить после создания** .

Включенное правило резервного копирования будет выполнено в ближайшую заданную дату и время.

11.1.3. Срочное резервное копирование по правилу

Срочное резервное копирование позволяет выполнить правило резервного копирования немедленно. Срочное резервное копирование по правилу требует настроенного правила в глобальном расписании.

1. Перейдите в раздел  **Глобальное расписание**.
2. Если правило с нужными настройками отсутствует в списке правил, создайте правило (см. [Раздел 11.1.1](#)). Убедитесь, что в создаваемом правиле снят флаг **Включить после создания** .
3. Если нужное правило резервного копирования есть в списке правил, выделите его в списке и нажмите  (**Выполнить**) в панели инструментов, или выберите **Выполнить** в контекстном меню этого правила.

Срочное резервное копирование создает задачу в разделе  **Задачи**.

11.1.4. Срочное резервное копирование (без правила)

Срочное резервное копирование (без правила) немедленно выполняет резервное копирование выбранного ресурса, но не создает правила в глобальном расписании.

1. Для выполнения срочного резервного копирования в *RBM* выберите один из способов:
 - в разделе **Объекты** выберите клиента резервного копирования, на котором установлена почтовая система *RuPost*, и нажмите  (**Срочное РК**);
 - нажмите  (**Срочное РК**) на верхней панели *RBM*.

Откроется форма срочного резервного копирования.

2. Из списка **Клиент** выберите узел клиента, на котором установлена почтовая система *RuPost*.
3. Из списка **Тип ресурса** выберите тип резервируемого ресурса *RuPost*. Для выбора резервируемого хранилища почтовой системы нажмите [...] и определите тонкие настройки модуля *RuPost* (см. [Раздел 12.3](#)).
4. Нажмите [...] в поле **Ресурс** и выберите резервируемый компонент почтовой системы *RuPost*.
5. Из списка **Тип РК** выберите тип резервной копии. Возможные значения:
 - полная,
 - инкрементальная;
 - дифференциальная.
6. Из списка **Пул** выберите пул для сохранения резервной копии.
7. Из списка **Защитное преобразование** при необходимости выберите алгоритм защитного преобразования (возможные значения см. в [Алгоритмы защитного преобразования](#)).
8. В **Приоритет** укажите приоритет выполнения срочного резервного копирования. Возможные значения: от 100 до 1000. Чем выше значение, тем выше приоритет выполнения срочного резервного копирования.
9. В **Срок хранения** укажите срок хранения резервной копии, созданной в результате выполнения срочного резервного копирования.
10. В **Транспортировочный буфер** укажите размер транспортировочного буфера (в байтах, килобайтах, мегабайтах или гигабайтах). В транспортировочный буфер помещаются блоки резервной копии, после чего буфер передается на сервер. От размера буфера зависит количество передаваемых блоков за один раз: чем больше размер буфера, тем быстрее данные передаются на сервер, но при этом задействуется больший объем оперативной памяти. Размер транс-

портировочного буфера указывается в диапазоне от 50 Мб до 1 Гб (по умолчанию 100 Мб).

11. Нажмите **Общие настройки модуля** и определите настройки для многопоточного резервного копирования (см. [Срочное резервное копирование](#)).
12. После указания необходимых параметров нажмите кнопку  **Применить** — будет создана задача на резервное копирование в разделе  **Задачи**.

Если во время выполнения задачи на резервное копирование какие-либо файлы ресурса изменили свой размер, то они будут включены в РК, а задача завершится со статусом *Done*.

Если во время выполнения задачи на резервное копирование какой-либо файл был удален, то при восстановлении будет восстановлена та часть файла, которая успела войти в РК.

11.2. Восстановление данных

Подробную информацию о выполнении восстановления резервных копий с помощью ПО *Менеджер Администратора RuBackup (RBM)* см. в [Менеджер администратора RuBackup \(RBM\)](#).

11.2.1. Централизованное восстановление резервных копий с помощью ПО Менеджер Администратора RuBackup (RBM)

Полное восстановление

Для полного восстановления компонентов почтовой системы *RuPost* на клиенте резервного копирования в *RBM*:

1. Перейдите в раздел  **Репозиторий** и выделите требуемую резервную копию. Нажмите кнопку  **Восстановить**, или выберите **Восстановить** в контекстном меню резервной копии.

Откроется форма **Централизованное восстановление**.

2. Из списка **Восстановить на клиента** выберите узел клиента резервного копирования.
3. В **Каталог распаковки** нажмите [...] и укажите временный каталог для распаковки резервной копии.
4. В **Параметры восстановления для модуля** нажмите [...] и в *тонких настройках* модуля *RuPost* (см. [Раздел 12.4](#)) определите компоненты почтовой системы, для которых нужно выполнить полное восстановление.
5. Нажмите **Общие настройки модуля** и определите настройки для многопоточного восстановления резервной копии (см. [Раздел «Репозиторий»](#)).

- Включите **Восстановить на целевом ресурсе** для восстановления РК на целевом ресурсе. Компоненты почтовой системы будут распакованы в холодное хранилище^[1 - Глава 2], минуя временный каталог. Служебная база данных будет распакована в **Каталог распаковки**, а затем развернута в почтовой системе *RuPost*.

Если флаг **Восстановить на целевом ресурсе** *выключен*, то развертывание не выполняется (резервные копии компонентов почтовой системы и служебной базы данных распаковываются во временный каталог, указанный в параметре **Каталог распаковки**).

- Нажмите  **Применить** для запуска восстановления резервной копии.

Будет создана задача восстановления РК в разделе  **Задачи**.

Гранулярное восстановление

Для гранулярного восстановления компонентов холодного^[1 - Глава 2] хранилища почтовой системы *RuPost* на клиенте резервного копирования в *RBM*:

- Перейдите в раздел  **Репозиторий** и выделите требуемую резервную копию. Нажмите кнопку  **Восстановить**, или выберите **Восстановить** в контекстном меню резервной копии.

Откроется форма **Централизованное восстановление**.

- Из списка **Восстановить на клиента** выберите узел клиента резервного копирования.
- В **Каталог распаковки** нажмите [...] и укажите временный каталог для распаковки резервной копии.
- В **Параметры восстановления для модуля** нажмите [...] и в *тонких настройках* модуля *RuPost* (см. [Раздел 12.4](#)) определите временной интервал для восстановления писем пользователей из выбранных папок почтовой системы *RuPost*. Письма пользователей восстанавливаются в том виде, в котором они существовали в почтовой системе *RuPost* в указанный период времени. Если временной интервал не указан (время начала интервала совпадает с временем окончания), то восстанавливаются все письма пользователей из выбранных папок в почтовых ящиках.

Календари или расписания восстанавливаются без привязки к временному интервалу.

- Нажмите **Общие настройки модуля** и определите настройки для многопоточного восстановления резервной копии (см. [Раздел «Репозиторий»](#)).
- Включите **Восстановить на целевом ресурсе** для восстановления РК на целевом ресурсе. Компоненты из холодного^[1 - Глава 2] хранилища почтовой системы

RuPost будут распакованы в холодное хранилище, минуя временный каталог.

Если флаг **Восстановить на целевом ресурсе** *выключен*, то развертывание не выполняется (резервные копии почтовых ящиков, архивов почтовых ящиков, удаленных писем, календарей и расписаний из холодного ^[1 - Глава 2] хранилища почтовой системы *RuPost* распаковываются во временный каталог, указанный в параметре **Каталог распаковки**).

7. В блоке **Гранулярное восстановление** нажмите  (**Добавить объекты**). Откроется форма со списком объектов гранулярного восстановления.
8. Выберите необходимого пользователя или конкретные папки пользователя из списка ^[1]. При необходимости выберите календари или расписания, в перечне они отображаются как *Sogo*.



При выполнении гранулярного восстановления значения параметров `restore_mailldir`, `restore_archive`, `restore_record`, `restore_database` в тонких настройках модуля *RuPost*, если они заданы, не учитываются.

9. Нажмите  **Применить**. В блоке **Гранулярное восстановление** отобразятся выбранные для восстановления компоненты из холодного ^[1 - Глава 2] хранилища. Список можно отредактировать или полностью очистить.
 10. Нажмите  **Применить** для запуска восстановления резервной копии.
- Будет создана задача восстановления РК в разделе  **Задачи**.

11.2.2. Восстановление на целевом ресурсе

Восстановление почтовых ящиков пользователей

При восстановлении РК почтовые ящики пользователей распаковываются в папку `.Restore`, которая расположена в холодном ^[1 - Глава 2] хранилище для почтовых ящиков. После успешной распаковки папка `.Restore` переименовывается в `Restore`.



Если папка `Restore` уже существует, то в процессе развертывания почтовых ящиков из папки будет удалено все содержимое.

Дальнейшее развертывание в почтовую систему осуществляется силами администратора почтовой системы.

Восстановление архивов почтовых ящиков пользователей

При восстановлении РК архивы почтовых ящиков пользователей распаковываются в папку `.Restore`, которая расположена в холодном ^[1 - Глава 2] хранилище для архивов почтовых ящиков. После успешной распаковки папка `.Restore` переименовывается в `Restore`.



Если папка `Restore` уже существует, то в процессе развертывания архивов почтовых ящиков из папки будет удалено все содержимое.

Дальнейшее развертывание в почтовую систему осуществляется силами администратора почтовой системы.

Восстановление удаленных писем пользователей

При восстановлении РК удаленные письма пользователей распаковываются в папку `.Restore`, которая расположена в холодном^[1 - Глава 2] хранилище для удаленных писем. После успешной распаковки папка `.Restore` переименовывается в `Restore`.



Если папка `Restore` уже существует, то в процессе развертывания удаленных писем из папки будет удалено все содержимое.

Дальнейшее развертывание в почтовую систему осуществляется силами администратора почтовой системы.

Восстановление базы данных

При восстановлении РК база данных распаковывается в выбранный пользователем временный каталог. После успешной распаковки происходит развёртывание базы данных: существующая база данных заменяется на восстанавливаемую, СУБД перезапускается.



Развёртывание базы данных возможно только в конфигурации *standalone*. В кластере *Patroni* подмену базы и перезапуск кластера осуществляет администратор базы данных.

11.3. Просмотр очереди задач

Проверьте ход выполнения задач на клиенте резервного копирования одним из способов в разделе ☒ **Задачи** через *RBM* (см. [Раздел «Задачи»](#)). При успешном завершении соответствующая задача перейдёт в статус *Done* (выполнено).

[1] Доступна пагинация по пользователям и папкам.

Глава 12. Приложения

12.1. Файл настроек модуля

В [таблице](#) описаны параметры модуля *RuPost* для взаимодействия с корпоративной почтовой системой *RuPost*.

Таблица 2. Параметры файла настроек `/opt/rubackup/etc/rb_module_rupost.conf` модуля *RuPost*

Параметр	Описание
<code>cold_storages_json_path</code>	Путь к JSON-файлу с конфигурацией холодных^[1 - Глава 2] хранилищ (почтовых ящиков, архивов почтовых ящиков, удаленных писем) каждого пространства хранения По умолчанию JSON-файл отсутствует в репозитории, создайте его вручную (см. Раздел 12.2)  После перехода на JSON-конфигурацию необходимо заново настроить правила и стратегии резервного копирования с привязкой к ID пространства хранения. Если РК была создана в версии RuBackup до 2.8, то при восстановлении с развёртыванием данные будут восстановлены в первое пространство хранения из Раздел 12.2.
<code>username</code>	Имя пользователя в СУБД <i>PostgreSQL</i>, обладающего правами выполнять резервное копирование По умолчанию <code>rubackup_backuper</code>
<code>password</code>	Пароль для пользователя, указанного в параметре <code>username</code>
<code>host</code>	IP-адрес или доменное имя хоста, на котором СУБД принимает подключения. Используется для взаимодействия с СУБД, резервное копирование которой выполняется. По умолчанию <code>localhost</code>
<code>port</code>	Порт для соединения с СУБД По умолчанию <code>5432</code>
<code>postgresql_admin</code>	Имя пользователя — администратора <i>PostgreSQL</i> По умолчанию <code>postgres</code>
<code>archive_catalog</code>	Каталог для хранения архивных WAL По умолчанию <code>/opt/rubackup/mnt/postgresql_archives</code>

Параметр	Описание
pg_ctl	Путь до утилиты <code>pg_ctl</code> По умолчанию <code>/usr/lib/postgresql/12/bin/pg_ctl</code> Параметр используется при восстановлении с разворачиванием для остановки и запуска сервиса СУБД  Для версии ОС <i>Astra Linux SE 1.8</i> необходимо закомментировать параметр <code>pg_ctl</code> и использовать параметр <code>postgresql_service_name</code>. По умолчанию параметр <code>postgresql_service_name</code> отсутствует в конфигурационном файле, его необходимо добавить вручную и указать в значении имя сервиса СУБД <i>PostgreSQL</i>.  Используется, если не задано значение параметра <code>postgresql_service_name</code>
postgresql_service_name	Имя <code>systemd</code> службы СУБД По умолчанию <code>postgresql</code> Параметр используется при восстановлении с разворачиванием для остановки и запуска сервиса СУБД через <code>systemctl</code>  Используется, если не задано значение параметра <code>pg_ctl</code>
num_threads_for_wal_processing	Количество процессов, выделенных для обработки архивных WAL файлов По умолчанию <code>8</code>
pg_binary	Используется при вызове утилиты <code>pg_ctl</code> для запуска <i>PostgreSQL</i> во время восстановления с разворачиванием. Указывает путь к исполняемому файлу <code>postgres</code>. По умолчанию исполняемый файл <code>postgres</code> берется из того же каталога, из которого запускался <code>pg_ctl</code>, а если найти файл невозможно, то из жёстко заданного каталога установки
pg_log	Используется при вызове утилиты <code>pg_ctl</code> для запуска <i>PostgreSQL</i> во время восстановления с разворачиванием. В указанный файл будет направляться вывод сообщений сервера. Файл создаётся, если он ещё не существует.
auto_remove_wal	В случае значения <code>yes</code> архивные WAL будут удалены из каталога <code>archive_catalog</code> после выполнения резервного копирования (если они включены в резервную копию) Возможные значения <code>yes</code>, <code>no</code> По умолчанию <code>yes</code>
wal_wait_timeout	Период ожидания окончания архивации последнего WAL-файла, сгенерированного во время создания резервной копии По умолчанию <code>10</code>

Параметр	Описание
wal_check_period	Период проверки окончания архивации последнего WAL-файла, сгенерированного во время создания резервной копии
	По умолчанию 1
pid_wait_timeout	Период проверки выключения сервиса PostgreSQL
	По умолчанию 10
patroni_host	IP-адрес, на котором Patroni принимает входящие запросы REST API. Параметр необходим для взаимодействия модуля с REST API локального процесса Patroni.
	Если значение параметра не указано, будет предпринята попытка автоматически определить значение для этого параметра через утилиту lsof
	По умолчанию localhost
patroni_port	Порт, на котором локальный процесс Patroni слушает запросы REST API. Параметр необходим для взаимодействия модуля с REST API локального процесса Patroni.
	Если значение параметра не указано, будет предпринята попытка автоматически определить значение для этого параметра через утилиту lsof
	По умолчанию 8008
patroni_node_type_for_backup	leader Ресурс будет доступен только при условии, что узел на котором установлен модуль с таким значением имеет роль leader в кластере Patroni.
	sync Ресурс будет доступен только при условии, что узел, на котором установлен модуль с таким значением, имеет роль sync standby в кластере Patroni.
	async Ресурс будет доступен только при условии, что узел, на котором установлен модуль с таким значением, имеет роль replica в кластере Patroni
allow_work_with_incompatible_versions	Позволяет использовать модуль, если версия платформы несовместима с CPK RuBackup
	Возможные значения yes, no
	По умолчанию no

12.2. Файл настроек хранилищ

В таблице описаны параметры модуля RuPost для взаимодействия с несколькими пространствами хранения корпоративной почтовой системы RuPost.

Таблица 3. Параметры файла настроек rupost_cold_storages.json модуля RuPost

Параметр	Описание
id	Уникальный ID каждого пространства хранения  Значение не может быть пустым
name	Имя пространства хранения
path	Путь к директории холодного ^[1 - Глава 2] хранилища почтовых ящиков  Значение параметра должно быть уникальным для каждого пространства хранения
archive_path	Путь к директории холодного ^[1 - Глава 2] хранилища архивов почтовых ящиков  Значение параметра должно быть уникальным для каждого пространства хранения
deleted_records_path	Путь к директории холодного ^[1 - Глава 2] хранилища удаленных писем  Значение параметра должно быть уникальным для каждого пространства хранения

Пример 5. Структура `rupost_cold_storages.json`

```
{
  "storages": [
    {
      "id": "storage_id_1",
      "name": "storage_name_1",
      "path": "/cold_storage_1",
      "archive_path": "/archive_path_1",
      "deleted_records_path": "/deleted_record_path_1"
    },
    {
      "id": "storage_id_2",
      "name": "storage_name_2",
      "path": "/cold_storage_2",
      "archive_path": "/archive_path_2",
      "deleted_records_path": "/deleted_record_path_2"
    }
  ]
}
```

12.3. Тонкие настройки модуля RuPost для резервного копирования

В [таблице](#) описаны тонкие настройки модуля *RuPost* для резервного копирования (см. [Раздел 11.1](#)).

Таблица 4. Тонкие настройки модуля *RuPost* для резервного копирования

Параметр	Описание
backup_maildir	Выполнение резервного копирования почтовых ящиков пользователей из холодного ^[1 - Глава 2] хранилища почтовой системы true Резервное копирование выполняется. false Резервное копирование не выполняется. По умолчанию true
backup_archive	Выполнение резервного копирования архива почтовых ящиков пользователей из холодного ^[1 - Глава 2] хранилища почтовой системы true Резервное копирование выполняется. false Резервное копирование не выполняется. По умолчанию false
backup_record	Выполнение резервного копирования удаленных писем пользователей из холодного ^[1 - Глава 2] хранилища почтовой системы true Резервное копирование выполняется. false Резервное копирование не выполняется. По умолчанию false
backup_database	Выполнение резервного копирования служебной базы данных почтовой системы true Резервное копирование выполняется. false Резервное копирование не выполняется. По умолчанию false

Параметр	Описание
connection_monitoring	Отслеживание связи с базой данных во время выполнения резервного копирования этой базы данных
	true Связь с БД отслеживается.
	false Связь с БД не отслеживается.
	По умолчанию true
	 Если резервное копирование служебной базы данных не происходит, то значение параметра <code>connection_monitoring</code> игнорируется.

Кнопка **Значения по умолчанию** очищает параметры и устанавливает для них значения по умолчанию.

Кнопка **ОК** сохраняет настройки.

12.4. Тонкие настройки модуля RuPost для восстановления резервной копии

В [таблице](#) описаны тонкие настройки модуля *RuPost* для восстановления резервной копии (см. [Раздел 11.2](#)).

Таблица 5. Тонкие настройки модуля *RuPost* для восстановления резервной копии

Параметр	Описание
Использовать настройки по умолчанию	Использование значений по умолчанию
	true Для параметров используются значения по умолчанию. Восстановление из РК происходит для почтовых ящиков, архива почтовых ящиков, удаленных писем и служебной базы данных всех пользователей почтовой системы.
	false Значения параметров можно изменить.
	По умолчанию true

Параметр	Описание
restore_maildir	Выполнение восстановления почтовых ящиков всех пользователей почтовой системы true Восстановление выполняется. false Восстановление не выполняется. По умолчанию true  Параметр используется только для полного восстановления почтовых ящиков.
restore_archive	Выполнение восстановления архива почтовых ящиков всех пользователей почтовой системы true Восстановление выполняется. false Восстановление не выполняется. По умолчанию true  Параметр используется только для полного восстановления архивов почтовых ящиков.
restore_record	Выполнение восстановления удаленных писем всех пользователей почтовой системы true Восстановление выполняется. false Восстановление не выполняется. По умолчанию true  Параметр используется только для полного восстановления удаленных писем.

Параметр	Описание
restore_database	Выполнение восстановления служебной базы данных из РК true Восстановление выполняется. false Восстановление не выполняется. По умолчанию false  Параметр используется только для полного восстановления служебной базы данных.
timestamp_begin timestamp_end	Начало и окончание временного интервала для гранулярного восстановления писем в выбранных папках почтовой системы <i>RuPost</i>. Задается по нажатию pass:n[  Параметры timestamp_begin и timestamp_end вместе определяют период времени для восстановления писем в выбранных папках почтовой системы <i>RuPost</i>. Отдельные письма пользователей восстанавливаются в том виде, в котором они существовали в почтовой системе <i>RuPost</i> в указанный период времени.  Начало временного интервала не должно совпадать с его окончанием. Если начало и окончание временного интервала совпадают, то гранулярное восстановление писем за указанный период времени не работает, будет выполнено восстановление всех выбранных писем.

Кнопка **OK** сохраняет настройки.