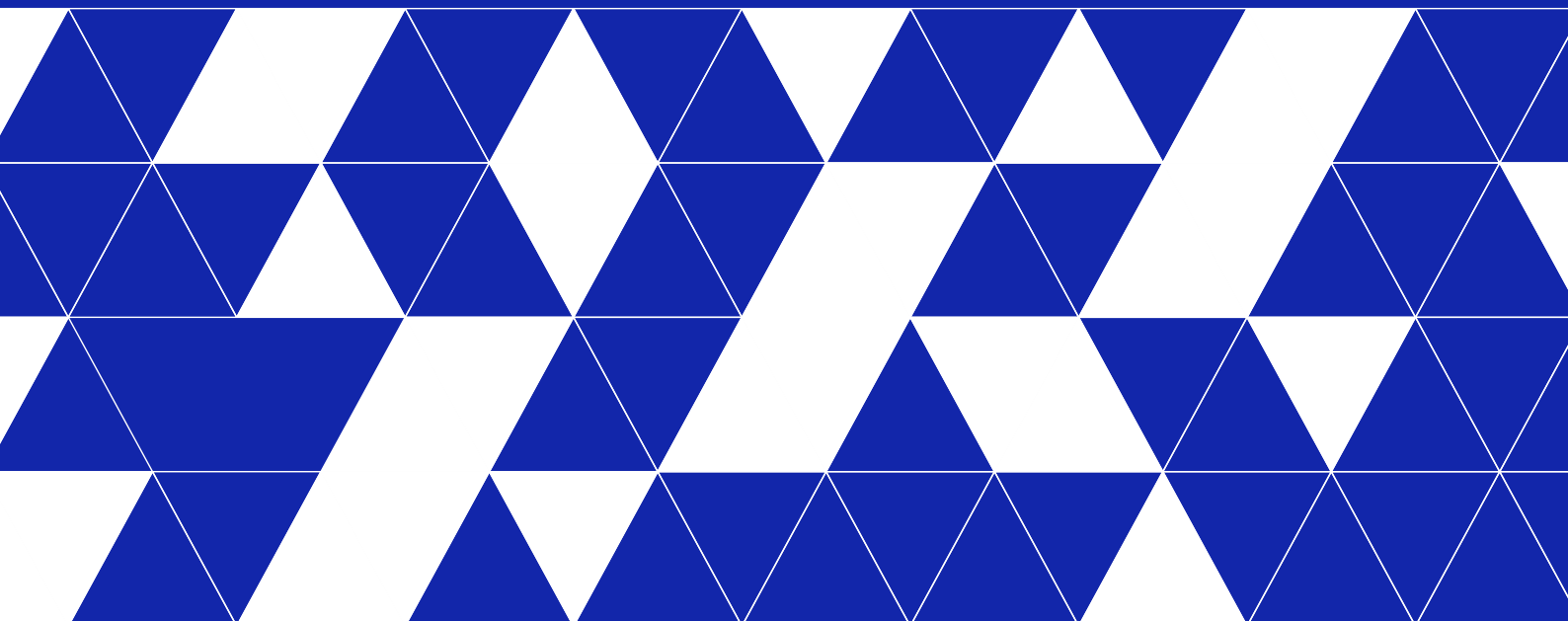


White Paper

Настройка резервного копирования PostgreSQL с помощью утилиты pgBackRest

TATLIN.OBJECT

06.11.2024
Антон Морозов, Максим Луганский



White Paper

Данный документ описывает архитектурные принципы и подходы, а также рекомендации по их применению при построении корпоративных решений на основе продуктов производства ГК YADRO.

Документ предназначен для широкой аудитории, но также может включать материалы, требующие специальной профессиональной подготовки, ориентированные на системных архитекторов, администраторов, сервисных инженеров и других IT-специалистов, участвующих в планировании, проектировании и внедрении комплексных решений.

Представленная информация основана на результатах исследований, а также экспертизе специалистов ГК YADRO и носит рекомендательный характер. Конкретные результаты и эффективность конечных решений зависят от множества факторов, которые могут различаться в зависимости от особенностей инфраструктуры и условий эксплуатации.

©2024 YADRO, все права защищены. YADRO®, VESNIN®, TATLIN®, VEGMAN® и KORNFELD® являются торговыми марками компании YADRO (или ее дочерних компаний), зарегистрированными на территории России и других стран.

Настройка резервного копирования PostgreSQL с помощью утилиты pgBackRest

Версия ПО TATLIN.OBJECT
1.0 – 1.5.3-30

Введение

Данный документ описывает настройку резервного копирования базы данных PostgreSQL на объектной системе хранения данных TATLIN.OBJECT с помощью утилиты pgBackRest.

Подробное описание утилиты доступно в *документации разработчика pgBackRest*.

Описаны методики настройки и резервного копирования, конфигурирования PostgreSQL, настройки целевого S3-хранилища TATLIN.OBJECT. Приведены параметры, влияющие на производительность резервного копирования, и даны рекомендации по их установке.

Установка утилиты pgBackRest

Используемая версия утилиты pgBackRest — 2.52.

1. Установите утилиту pgBackRest:

```
sudo apt-get install pgbackrest -y
```

2. Утилита pgBackRest требует наличия perl — установите поддержку perl:

```
sudo apt-get install perl
```

3. Создайте директории для логов, выдайте им определенные права:

```
sudo mkdir -p -m 770 /var/log/pgbackrest
sudo chown postgres:postgres /var/log/pgbackrest
sudo mkdir -p /etc/pgbackrest
sudo mkdir -p /etc/pgbackrest/conf.d
sudo touch /etc/pgbackrest/pgbackrest.conf
sudo chmod 640 /etc/pgbackrest/pgbackrest.conf
sudo chown postgres:postgres /etc/pgbackrest/pgbackrest.conf
```

4. Проверьте версию установленной утилиты pgBackRest:

```
pgbackrest version
```

Настройка объектного хранилища для работы с утилитой pgBackRest

Создайте бакет, в котором будут храниться резервные копии:

```
aws s3api --no-verify-ssl create-bucket --bucket <BUCKET_NAME> --endpoint-url <Базовый-URL-S3> [--acl <ACL>
 --create-bucket-configuration LocationConstraint=<CONSTRAINT>]
```

--no-verify-ssl

Пропуск проверки SSL-сертификатов (рекомендуется).

--endpoint-url <Базовый-URL-S3>

Адрес подключаемого узла хранения в формате `https://<Базовый-URL-S3>`.

--create-bucket-configuration LocationConstraint=<CONSTRAINT>

Политика (набор правил) хранения объектов в контейнере.

Примеры политики:

- `load-1-4` — размещение одной копии объекта в контейнере на четырех узлах хранения.
- `load-3-3` — размещение трех копий объекта в контейнере на трех узлах хранения.
- `load-1-1` — размещение одной копии объекта в контейнере на одном узле хранения.

Настройка службы PostgreSQL и утилиты pgBackRest

1. Внесите необходимые настройки в конфигурационный файл `/etc/postgresql/XX/main/postgresql.conf` для работы утилиты pgBackRest.

   Пример

   ```
   archive_command = 'pgbackrest --stanza=main archive-push %p'
   archive_mode = on
   max_wal_senders = 10
   wal_level = replica
   ```

   main

   Название кластера. При установке служба PostgreSQL автоматически создает кластер `main`.

   max_wal_senders

   Максимально допустимое количество одновременных подключений ведомых серверов или клиентов потокового копирования. Оно же — максимальное количество одновременно работающих процессов передачи WAL.

   Значение по умолчанию — `10`. При значении `0` репликация отключается. В случае неожиданного отключения клиента потоковой передачи слот подключения может оставаться занятым до достижения таймаута. Данный параметр должен быть немного больше максимально допустимого числа клиентов, чтобы отключившиеся клиенты могли переподключиться немедленно.

2. Перезапустите службу PostgreSQL:

   ```
   sudo systemctl restart <SERVICE_NAME>
   ```

   <SERVICE_NAME>

   Название сервиса. Например, `PostgreSQL`.

3. Внесите необходимые настройки в файл конфигурации `/etc/pgbackrest/pgbackrest.conf`:.

> Описание возможных параметров доступно в *документации разработчика pgBackRest*.

```
[main]
pg1-path=/var/lib/postgresql/16/main
[global]
log-level-file=detail
repo1-s3-verify-ssl=n
repo1-path=/var/lib/pgbackrest
repo1-retention-full-type=count
repo1-retention-full=20
repo1-type=s3
repo1-s3-bucket=pgbackrest
repo1-s3-endpoint= data0.indierock-node1.indierock.object.set.yadro.com:443
repo1-s3-key=6rRKJWXTjRZavf5L4sH7avZ7jjPBBBnpmpRZJ9Vp8qed0AY7k1yXNBmKHcugZrtgGme8gkP2yCCYKXqXNg5Xnd94L
repo1-s3-key-secret=d6de06bda816566c7cf29ce7c3d7b1d906a82a0b3e9c2401f3da033defe287df
repo1-s3-region=eu-west-3
repo1-s3-uri-style=path
process-max=50
start-fast=y
[global:archive-push]
compress-level=3
```

pg1-path=/var/lib/postgresql/16/main

Каталог данных PostgreSQL. Все параметры `pg-` должны иметь индекс, который позволяет настроить несколько хостов PostgreSQL.

log-level-file

Уровень детализации логирования журнала.

Перечень уровней детализации:

- `off` — журналирование не ведется (не рекомендуется);
- `error` — только ошибки;
- `warn` — только ошибки и предупреждения;
- `info` — информация журнала, предупреждения и ошибки;
- `detail` — подробности журнала, предупреждения и ошибки;
- `debug` — журнал отладки, подробности, информация, предупреждения и ошибки;
- `trace` — журнал трассировки (очень подробная отладка), отладка, информация, предупреждения и ошибки.

repo1-s3-verify-ssl

Проверка сертификата хранилища.

repo1-path

Путь, по которому хранятся резервные копии и архив.

repo1-retention-full-type

Тип исчисления хранения для полных резервных копий (количество или время).

repo1-retention-full

Счетчик хранения полной резервной копии (количество или время).

repo1-type

Тип репозитория.

repo1-s3-bucket

Имя бакета S3.

repo1-s3-endpoint

Адрес конечной точки (эндпоинта) хранилища S3.

repo1-s3-key

Ключ доступа к репозиторию S3.

repo1-s3-key-secret

Секретный ключ доступа к репозиторию S3.

repo1-s3-region

Регион репозитория S3.

repo1-s3-uri-style

Стиль URI S3.

process-max

Максимальное количество процессов, используемых для сжатия/передачи. Параметр влияет на скорость создания бэкапа. Каждый процесс приблизительно соответствует 70–80 Мбит/с. Производитель утилиты pgBackRest не рекомендует выделять более 25 % от общего количества ядер ЦПУ. Действие может повлиять на производительность базы данных во время создания резервной копии. Рекомендуется провести предварительное тестирование для нахождения оптимального значения параметра.

start-fast

Форсирование создания контрольной точки путем передачи `y` в параметр `fast` функции запуска резервного копирования. Действие необходимо для того, чтобы резервное копирование началось немедленно. Иначе резервное копирование начнется после следующей очередной контрольной точки.

compress-level

Уровень сжатия файлов, если тип сжатия не равен `none`. Параметр не имеет существенного влияния на скорость, размер или загрузку ЦПУ. Производитель утилиты pgBackRest рекомендует использовать значение `3` как оптимальное по нагрузке на ЦПУ и уровню сжатия:

- `bz2` — 9;
- `gz` — 6;
- `lz4` — 1;
- `zst` — 3.

4. Создайте локальное хранилище:

```
sudo mkdir -m 770 /var/lib/pgbackrest
sudo chown -R postgres /var/lib/pgbackrest/
sudo -u postgres pgbackrest --stanza=main stanza-create
```

5. Сделайте резервную копию. Утилита pgBackRest создаст первую полную резервную копию:

```
sudo -u postgres pgbackrest --log-level-console=info --stanza=main backup
```

6. При необходимости запустите команду создания резервной копии повторно. Убедитесь, что система создаст инкрементальную резервную копию.

   Для повторного создания резервной копии укажите дополнительный флаг `--type=full backup`:

```
sudo -u postgres pgbackrest --log-level-console=info --stanza=main --type=full backup
```

Для просмотра списка резервных копий используйте команду:

```
sudo -u postgres pgbackrest --stanza=main info
```

Восстановление данных из резервной копии

1. Остановите работающий кластер:

```
sudo pg_ctlcluster 16 main stop
```

2. Восстановите данные из резервной копии:

```
sudo -u postgres pgbackrest --stanza=main --log-level-console=info --delta
 --recovery-option=recovery_target=immediate restore
```

   Для восстановления данных из последней полной резервной копии используйте команду без указания `recovery_target`:

```
sudo -u postgres pgbackrest --stanza=main --log-level-console=info --delta restore
```

3. Запустите кластер:

```
sudo pg_ctlcluster 16 main start
```

4. После восстановления данных из резервной копии выполните повторное создание резервной копии:

```
sudo pgbackrest --stanza=main backup
```