Авторы: Алексей Береснев, Илья Баштанов, Павел Толмачев

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Темы

Хранилище S3

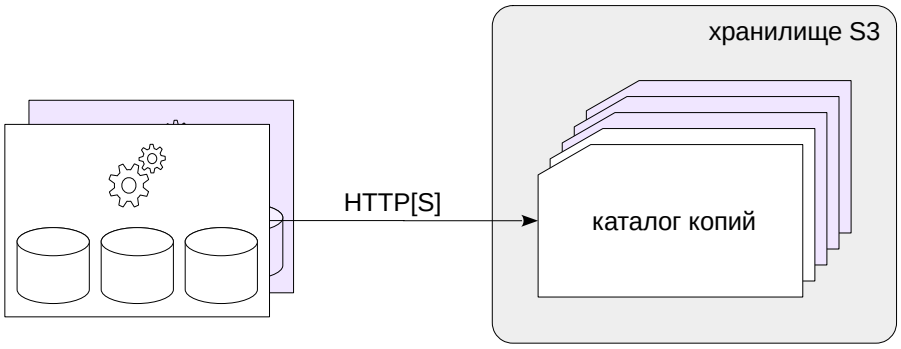
Политики хранения

Журнал сообщений

Хранилище S3

Поддержка S3

Поддержка S3



Каталог копий можно размещать в объектном хранилище с интерфейсом S3

- MinIO
- VK Cloud

Утилита pg_probackup способна работать с каталогом копий, размещенным в объектном хранилище, предоставляющим интерфейс S3 (Simple Storage Service).

В настоящее время поддерживаются два провайдера: MinIO и VK Cloud. MinIO — программа с открытым кодом, реализующая совместимый с S3 API и важнейшие возможности, аналогичные Amazon Web Services. В демонстрационных материалах курса используется MinIO.

При использовании каталога копий, размещенного в хранилище S3, не будет работать команда merge. Также с командами backup и delete нельзя использовать ключ --merge-expired.

Работа с хранилищем S3

Определим роль backup для резервного копирования и восстановления.

```
student$ psql
```

```
=> CREATE ROLE backup LOGIN REPLICATION;
```

```
CREATE ROLE
```

Создадим базу данных, к которой будет подключаться роль backup.

```
=> CREATE DATABASE backup OWNER backup;
```

```
CREATE DATABASE
```

Привилегии для роли backup.

```
=> \c backup
```

```
You are now connected to database "backup" as user "student".
```

```
=>
BEGIN;
GRANT USAGE ON SCHEMA pg_catalog TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.current_setting(text) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.set_config(text, text, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_is_in_recovery() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_start_backup(text, boolean, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_stop_backup(boolean, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_create_restore_point(text) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_switch_wal() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_last_wal_replay_lsn() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current_snapshot() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_snapshot_xmax(txid_snapshot) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_control_checkpoint() TO backup;
COMMIT;
```

```
BEGIN
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
COMMIT
```

В виртуальной машине установлено и настроено программное обеспечение MinIO, реализующее API S3. В MinIO уже создана корзина bkp для хранения каталога копий.

Запустим сервис MinIO.

```
student$ sudo systemctl enable --now --quiet minio
```

Для хранения в облачном хранилище резервных копий используется корзина - аналог каталога. Удалим корзину, если она осталась от ранее выполненных резервных копирований.

```
student$ /usr/local/bin/mc --quiet rb --force local/bkp
```

```
Removed `local/bkp` successfully.
```

Теперь создадим корзину.

```
student$ /usr/local/bin/mc --quiet mb local/bkp
```

```
Bucket created successfully `local/bkp`.
```

Создадим файл конфигурации для подключения к хранилищу S3:

```
student$ cat << EOF > /home/student/s3.config
access-key = minioadmin
secret-key = minioadmin
s3-host = localhost
s3-port = 9000
s3-bucket = bkp
s3-secure = off
EOF
```

- access_key и secret-key задают имя учетной записи и пароль в хранилище;
- s3-host и s3-port — узел и порт службы S3;
- s3-secure — использовать ли протокол https;
- s3-bucket — имя корзины S3.

Инициализируем каталог копий в хранилище S3. Провайдер указывается в ключе --s3, расположение файла с настройками — в ключе --s3-config-file:

```
student$ pg_probackup init -B /probackup --s3=minio --s3-config-file=/home/student/s3.config
```

```
INFO: S3_pre_start_check in progress
INFO: S3_pre_start_check successful, continue the operation
INFO: Backup catalog '/probackup' successfully initialized
```

Если ключ --s3-config-file опущен, pg_probackup ищет файл конфигурации S3 сначала в /etc/pg_probackup/s3.config, а затем в ~postgres/.pg_probackup/s3.config.

Также параметры подключения можно задать с помощью переменных окружения.

Добавим в локальный каталог копий экземпляр БД, работающий на удаленном сервере. Каталог копий в хранилище S3.

```
student$ pg_probackup add-instance -B /probackup -D /var/lib/pgpro/ent-13 --instance ent-13 --s3=minio --s3-config-file=/home/student/s3.config
```

```
INFO: Instance 'ent-13' successfully initialized
```

Чтобы сократить командную строку, сохраним параметры удаленного доступа в конфигурации.

```
student$ pg_probackup set-config -B /probackup --instance ent-13 -d backup -U backup --s3=minio --s3-config-file=/home/student/s3.config
```

Внимание! Ключи --s3=minio и --s3-config-file=/home/student/s3.config задают подключение к S3, они в конфигурацию не записываются и должны задаваться в командной строке явно.

Вот что записалось в конфигурацию:

```
student$ pg_probackup show-config -B /probackup --instance ent-13 --s3=minio --s3-config-file=/home/student/s3.config
```

```
# Backup instance information
pgdata = /var/lib/pgpro/ent-13
system-identifier = 7323121610635147817
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = backup
pguser = backup
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
```

```
log-level-file = OFF
log-format-console = PLAIN
log-format-file = PLAIN
log-filename = pg_probackup.log
log-rotation-size = 0TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 1
# Remote access parameters
remote-proto = ssh
```

Выполним полное резервное копирование в облако.

```
student$ pg_probackup backup -B /probackup --instance ent-13 -b FULL --stream --s3=minio --s3-config-file=/home/student/s3.config

INFO: Backup start, pg_probackup version: 2.6.7, instance: ent-13, backup ID: S7DAIX, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1
INFO: Skipping exclusive lock on remote drive
INFO: This PostgreSQL instance was initialized with data block checksums. Data block corruption will be detected
INFO: Database backup start
INFO: wait for pg_start_backup()
INFO: PGDATA size: 334MB
INFO: Current Start LSN: 0/10000028, TLI: 1
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 8s
INFO: wait for pg_stop_backup()
INFO: pg_stop backup() successfully executed
INFO: stop_stream_lsn 0/11000000 currentpos 0/11000000
INFO: backup->stop_lsn 0/1000B998
INFO: Getting the Recovery Time from WAL
INFO: Validating backup S7DAIX
INFO: Skipping exclusive lock on remote drive
INFO: Backup S7DAIX data files are valid
INFO: Backup S7DAIX resident size: 350MB
INFO: Backup S7DAIX completed
```

В каталоге имеется полная потоковая копия:

```
student$ pg_probackup show -B /probackup --instance ent-13 --s3=minio --s3-config-file=/home/student/s3.config

================================================================================================================================================
 Instance  Version  ID      Recovery Time                  Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN   Stop LSN    Status
================================================================================================================================================
 ent-13    13       S7DAIX  2024-01-16 21:40:17.759859+03  FULL  STREAM    1/0    9s  334MB  16MB    1,00  0/10000028  0/1000B998  OK
```

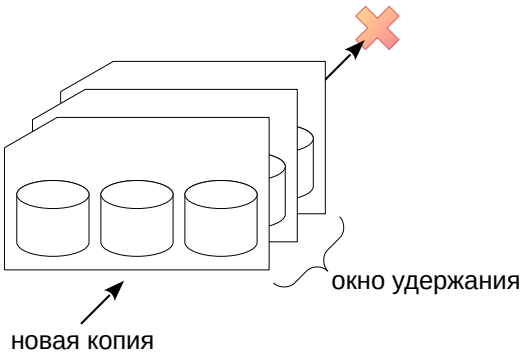
Политики хранения

Удаление лишних копий

Закрепление копий

Политика хранения WAL

Удаление лишних копий



По умолчанию в каталоге хранятся все резервные копии

Политика хранения автоматически удаляет лишние копии и ненужные сегменты WAL

количество хранимых копий

окно удержания в днях

Экземпляру можно задать индивидуальную политику хранения резервных копий, определяющую правила удаления устаревших копий и ненужных сегментов WAL. Политика определяется количеством полных копий, которые должны сохраняться в каталоге до автоматического удаления устаревших копий, а также окном удержания. Окно удержания определяет самый ранний момент времени, на который pg_probackup может выполнить восстановление. Окно удержания ограничивает количество дней сохранения избыточных копий в каталоге.

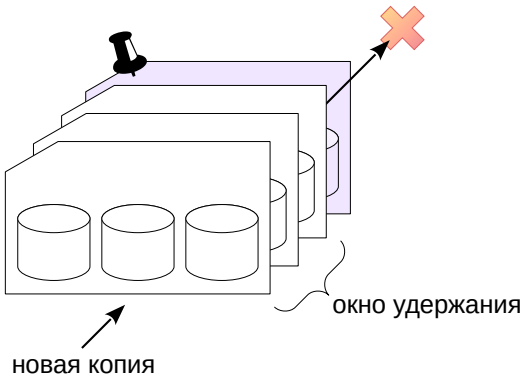
Например, если окно удержания установлено в значение 7 дней, то в каталоге должна сохраняться минимум одна копия старше 7 дней с файлами WAL и цепочкой инкрементальных копий.

Политика хранения задается ключами:

- --retention-redundancy — количество полных резервных копий, которое должно сохраняться в каталоге;
- --retention-window — количество дней от текущего момента до самого раннего момента времени в прошлом, на который может быть выполнено восстановление.

Ключи можно указать в командной строке pg_probackup для команд backup или delete, или сохранить в конфигурации командой set-config. Можно объединить самую старую инкрементальную копию, удовлетворяющую требованиям политики хранения, с ее родительскими копиями, срок хранения которых истек. Для этого предназначен ключ --merge-expired.

Закрепление копий



На закрепленные копии не действует политика хранения

Параметры закрепления (команды backup и set-backup)

- --ttl — время закрепления
- --expire-time — срок хранения резервной копии

Если нужно хранить копию дольше, чем задано политикой хранения, можно закрепить ее на определенное время.

Закрепление можно задать при выполнении команды backup или сохранить в конфигурации командой set-backup.

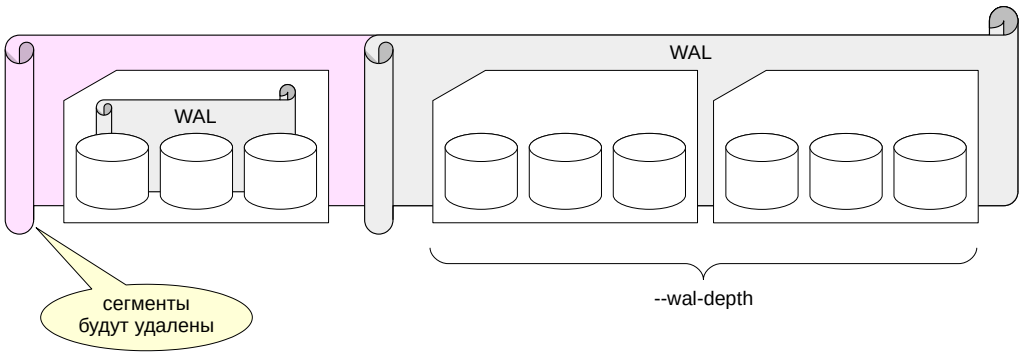
Ключ --ttl задает требуемое время жизни копии. Поддерживаются следующие единицы измерения: ms (миллисекунды), s (секунды), min (минуты), h (часы), d (дни). По умолчанию используются секунды.

Чтобы отменить закрепление, можно передать ноль в ключе --ttl.

Другой вариант — явно задать время истечения срока хранения копии, воспользовавшись ключом --expire-time.

Определить, закреплена ли копия, можно с помощью команды show утилиты pg_probackup. Закрепленная копия имеет атрибут expire-time, содержащий время окончания срока ее хранения.

Политика хранения WAL



Политика хранения архива WAL ограничивает глубину архива для экономии места в хранилище

Удаление устаревших копий с --delete-wal сопровождается удалением лишних сегментов WAL

При работе в режиме непрерывного архивирования сегменты WAL накапливаются в хранилище. Но его емкость ограничена.

Можно удалять устаревшие резервные копии с параметром --delete-wal, при этом удаляются только те сегменты WAL, которые не относятся ни к одной из резервных копий в каталоге. Если возможность восстановления на момент времени нужна только для самых недавних резервных копий, можно, настроив политику хранения архива WAL, ограничить глубину архива и сэкономить занимаемое место. Конечно, восстановиться из копии, оставшейся без архива WAL, можно только если она является автономной (то есть содержит необходимые журнальные файлы).

Пример политики: «восстановление на момент времени требуется лишь для двух последних выполненных резервных копий».

Политику хранения архива WAL настраивают командой pg_probackup set-config с параметром --wal-depth, задающим количество копий, которые могут быть использованы для PITR. Установленная политика действует для всех линий времени, поэтому можно выполнять PITR для одинакового количества копий на каждой линии времени. Закрепленные копии в этом числе не учитываются: если закрепляется одна из последних копий, pg_probackup обеспечивает возможность PITR для каждой дополнительной копии.

Также можно удалить сегменты, не удовлетворяющие заданной политике хранения архива WAL, выполнив команду pg_probackup delete или pg_probackup backup --delete-wal.

Политика хранения копий.

Выполним еще одно полное резервное копирование.

```
student$ pg_probackup backup -B /probackup --instance ent-13 -b FULL --stream --s3=minio --s3-config-file=/home/student/s3.config
```

```
INFO: Backup start, pg_probackup version: 2.6.7, instance: ent-13, backup ID: S7DAJ7, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1
INFO: Skipping exclusive lock on remote drive
INFO: This PostgreSQL instance was initialized with data block checksums. Data block corruption will be detected
INFO: Database backup start
INFO: wait for pg_start_backup()
INFO: PGDATA size: 334MB
INFO: Current Start LSN: 0/11000028, TLI: 1
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 7s
INFO: wait for pg_stop_backup()
INFO: pg_stop backup() successfully executed
INFO: stop_stream_lsn 0/12000000 currentpos 0/12000000
INFO: backup->stop_lsn 0/11000210
INFO: Getting the Recovery Time from WAL
INFO: Validating backup S7DAJ7
INFO: Skipping exclusive lock on remote drive
INFO: Backup S7DAJ7 data files are valid
INFO: Backup S7DAJ7 resident size: 350MB
INFO: Backup S7DAJ7 completed
```

Кроме последней, актуальной копии, в каталоге есть еще одна полная копия.

```
student$ pg_probackup show -B /probackup --s3=minio --s3-config-file=/home/student/s3.config
```

```
BACKUP INSTANCE 'ent-13'
=============================================================================================================================================
 Instance  Version  ID      Recovery Time                  Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN   Stop LSN    Status
=============================================================================================================================================
 ent-13    13       S7DAJ7  2024-01-16 21:40:26.018466+03  FULL  STREAM    1/0    7s  334MB  16MB    1,00  0/11000028  0/11000210  OK
 ent-13    13       S7DAIX  2024-01-16 21:40:17.759859+03  FULL  STREAM    1/0    9s  334MB  16MB    1,00  0/10000028  0/1000B998  OK
```

Если принята политика хранить единственную полную резервную копию, то потоковая копия является устаревшей.

Количество хранящихся избыточных копий можно задать непосредственно в командной строке. Выполним удаление, указав, что должна храниться только одна копия.

```
student$ pg_probackup delete -B /probackup --instance=ent-13 --delete-expired --retention-redundancy=1 --s3=minio --s3-config-file=/home/student/s3.config
```

```
INFO: Evaluate backups by retention
INFO: Backup S7DAJ7, mode: FULL, status: OK. Redundancy: 1/1, Time Window: 0d/0d. Active
INFO: Backup S7DAIX, mode: FULL, status: OK. Redundancy: 2/1, Time Window: 0d/0d. Expired
INFO: Skipping exclusive lock on remote drive
INFO: Delete: S7DAIX 2024-01-16 21:40:17+03
INFO: Skipping exclusive lock on remote drive
INFO: There are no backups to merge by retention policy
INFO: Purging finished
INFO: There is no WAL to purge by retention policy
```

Проверим состояние каталога резервных копий.

```
student$ pg_probackup show -B /probackup --s3=minio --s3-config-file=/home/student/s3.config
```

```
BACKUP INSTANCE 'ent-13'
=============================================================================================================================================
 Instance  Version  ID      Recovery Time                  Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN   Stop LSN    Status
=============================================================================================================================================
 ent-13    13       S7DAJ7  2024-01-16 21:40:26.018466+03  FULL  STREAM    1/0    7s  334MB  16MB    1,00  0/11000028  0/11000210  OK
```

Осталась единственная полная резервная копия.

Теперь установим политику удержания единственной полной копии.

```
student$ pg_probackup set-config -B /probackup --instance ent-13 --retention-redundancy=1 --s3=minio --s3-config-file=/home/student/s3.config
```

Сделаем полную копию со сжатием. Укажем, что должны быть удалены устаревшие в соответствии с политикой копии и WAL.

```
student$ pg_probackup backup -b FULL -B /probackup --instance ent-13 --stream --temp-slot --compress --delete-expired --delete-wal --s3=minio --s3-config-file=/home/student/s3.config
```

```
INFO: Backup start, pg_probackup version: 2.6.7, instance: ent-13, backup ID: S7DAJG, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: zlib, compress-level: 1
INFO: Skipping exclusive lock on remote drive
INFO: This PostgreSQL instance was initialized with data block checksums. Data block corruption will be detected
INFO: Database backup start
INFO: wait for pg_start_backup()
INFO: PGDATA size: 334MB
INFO: Current Start LSN: 0/13000028, TLI: 1
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 6s
INFO: wait for pg_stop_backup()
INFO: pg_stop backup() successfully executed
INFO: stop_stream_lsn 0/14000000 currentpos 0/14000000
INFO: backup->stop_lsn 0/130001C0
INFO: Getting the Recovery Time from WAL
INFO: Validating backup S7DAJG
INFO: Skipping exclusive lock on remote drive
INFO: Backup S7DAJG data files are valid
INFO: Backup S7DAJG resident size: 114MB
INFO: Backup S7DAJG completed
INFO: Evaluate backups by retention
INFO: Backup S7DAJG, mode: FULL, status: OK. Redundancy: 1/1, Time Window: 0d/0d. Active
INFO: Backup S7DAJ7, mode: FULL, status: OK. Redundancy: 2/1, Time Window: 0d/0d. Expired
INFO: Skipping exclusive lock on remote drive
INFO: Delete: S7DAJ7 2024-01-16 21:40:26+03
INFO: Skipping exclusive lock on remote drive
INFO: There are no backups to merge by retention policy
INFO: Purging finished
INFO: There is no WAL to purge by retention policy
```

```
student$ pg_probackup show -B /probackup --s3=minio --s3-config-file=/home/student/s3.config
```

```
BACKUP INSTANCE 'ent-13'
=============================================================================================================================================
 Instance  Version  ID      Recovery Time                  Mode  WAL Mode  TLI  Time  Data   WAL  Zratio  Start LSN   Stop LSN    Status
=============================================================================================================================================
 ent-13    13       S7DAJG  2024-01-16 21:40:34.309279+03  FULL  STREAM    1/0    6s  98MB  16MB    3,42  0/13000028  0/130001C0  OK
```

Осталась единственная полная резервная копия со сжатием.

Журнал сообщений

Журнал отчета

Ротация журнала

Журнал отчета

По умолчанию pg_probackup выводит сообщения в stderr

Можно указать файл журнала для записи сообщений

- отдельный журнал для сообщений об ошибках

Уровни важности сообщений настраиваются отдельно для вывода на консоль и в файл журнала

В pg_probackup определены следующие уровни важности сообщений: verbose, log, info, warning, error и off. Каждый уровень включает все последующие, и с каждым следующим уровнем объем сообщений уменьшается. При работе pg_probackup сообщения с уровнем важности info и выше выводятся по умолчанию на консоль в стандартный поток вывода ошибок stderr.

Настраивать уровень сообщений, выводимых в stderr, можно посредством ключа --log-level-console.

Если требуется записывать сообщения в журнал, то следует задать имя файла с помощью --log-filename. По умолчанию файл располагается в подкаталоге log каталога резервных копий, ключом --log-directory расположение можно переопределить.

Для файла журнала имеется отдельная настройка фильтрации сообщений по уровню важности --log-level-file.

Вывод сообщений как на консоль, так и в журнал, можно вести в обычном текстовом виде или же в формате JSON.

Настройкой --error-log-filename можно указать отдельные файлы для сообщений об ошибках.

Журнал отчета

Ротация файлов журнала задается параметрами

- по достижении заданного размера файла
- ограничением времени записи в файл

Чтобы журналы сообщений не разрастались бесконтрольно, нужно настроить ротацию. Она может происходить по достижении указанного размера файла журнала, либо по достижении максимального времени записи в файл.

Настройка журнала отчета

Настроим параметры по умолчанию для журнала отчета:

- --log-level-file — уровень сообщений, которые будут выводиться в файл журнала;
- --log-level-console — уровень сообщений, которые будут выводиться на консоль;
- --log-filename — имя файла отчета (по умолчанию в подкаталоге log, расположение можно изменить с помощью --log-directory);
- --log-rotation-size — размер файла, при превышении которого выполняется ротация.

Каталог для отчетов.

```
student$ rm -rf /home/student/log; mkdir /home/student/log
```

```
student$ pg_probackup set-config -B /probackup --instance ent-13 --log-directory=/home/student/log --log-filename=probackup.log --log-rotation-size=5MB --log-level-file=info --log-leve
```

```
student$ pg_probackup show-config -B /probackup --instance ent-13 --s3=minio --s3-config-file=/home/student/s3.config
```

```
# Backup instance information
pgdata = /var/lib/pgpro/ent-13
system-identifier = 7323121610635147817
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = backup
pguser = backup
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = WARNING
log-level-file = INFO
log-format-console = PLAIN
log-format-file = PLAIN
log-filename = probackup.log
log-directory = /home/student/log
log-rotation-size = 5MB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 1
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 1
# Remote access parameters
remote-proto = ssh
```

Выполним резервное копирование с только что заданными настройками журнала отчета.

```
student$ pg_probackup backup -b FULL -B /probackup --instance ent-13 --stream --temp-slot --compress --delete-expired --delete-wal --s3=minio --s3-config-file=/home/student/s3.config
```

Заглянем в каталог с отчетом:

```
student$ ls -lh /home/student/log
```

```
total 8,0K
-rw-rw-r-- 1 student student 2,9K янв 16 21:40 probackup.log
-rw-rw-r-- 1 student student   10 янв 16 21:40 probackup.log.rotation
```

Еще раз выполним резервное копирование, включив подробный вывод сообщений в отчет.

```
student$ pg_probackup backup -b FULL -B /probackup --instance ent-13 --stream --temp-slot --compress --delete-expired --delete-wal --log-level-file=verbose --s3=minio --s3-config-file=
```

В файл отчета записалось большое количество сообщений.

```
student$ ls -lh /home/student/log
```

```
total 12M
-rw-rw-r-- 1 student student 12M янв 16 21:40 probackup.log
-rw-rw-r-- 1 student student  10 янв 16 21:40 probackup.log.rotation
```

Повторим эксперимент с уровнем важности сообщений по умолчанию. Файл журнала отчета достиг размера, при котором должна произойти ротация:

```
student$ pg_probackup backup -b FULL -B /probackup --instance ent-13 --stream --temp-slot --compress --delete-expired --delete-wal --s3=minio --s3-config-file=/home/student/s3.config
```

В результате ротации журнала отчета старые сообщения были стерты и размер файла уменьшился.

```
student$ ls -lh /home/student/log
```

```
total 8,0K
-rw-rw-r-- 1 student student 2,6K янв 16 21:41 probackup.log
-rw-rw-r-- 1 student student   10 янв 16 21:40 probackup.log.rotation
```

Итоги

Каталог резервных копий может размещаться в облаке S3

Поддерживаются политики хранения копий и архива WAL

Политика хранения автоматически удаляет лишние резервные копии и ненужные сегменты WAL

Утилита pg_probackup позволяет настроить ротацию журналов сообщений

Практика

1. Подготовьте экземпляр к локальной работе.
2. Настройте вывод сообщений в журнал и политику для хранения единственной полной копии.
3. Проверьте удаление лишних копий, пару раз выполнив полное копирование.

1. Инициализируйте каталог с помощью pg_probackup init, а затем добавьте в него экземпляр командой pg_probackup add-instance. Зарегистрировав роль backup, создайте одноименную базу данных и назначьте командой GRANT требуемые права.

2. Используйте команду pg_probackup set-config с ключами --log-filename=probackup.log --log-level-file=info --log-level-console= warning по аналогии с демонстрацией. В этой же команде укажите ключ --retention-redundancy=1.

3. При выполнении полного резервного копирования используйте ключ --stream. Вторая команда полного резервного копирования должна сопровождаться ключом --delete-expired.

1. Подготовка каталога резервного копирования

Подготавливаем каталог в файловой системе.

```
student$ sudo mkdir /var/probackup
```

```
student$ sudo chown student: /var/probackup
```

Инициализируем каталог копий.

```
student$ pg_probackup init -B /var/probackup
```

```
INFO: Backup catalog '/var/probackup' successfully initialized
```

Регистрируем экземпляр.

```
student$ pg_probackup add-instance -B /var/probackup -D /var/lib/pgpro/ent-13 --instance ent-13
```

```
INFO: Instance 'ent-13' successfully initialized
```

Роль и база данных для подключения.

```
student$ psql
```

```
=> CREATE ROLE backup LOGIN REPLICATION;
```

```
CREATE ROLE
```

```
=> CREATE DATABASE backup OWNER backup;
```

```
CREATE DATABASE
```

Предоставляем права.

```
=> \c backup
```

```
You are now connected to database "backup" as user "student".
```

```
=>
BEGIN;
GRANT USAGE ON SCHEMA pg_catalog TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.current_setting(text) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.set_config(text, text, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_is_in_recovery() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_start_backup(text, boolean, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_stop_backup(boolean, boolean) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_create_restore_point(text) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_switch_wal() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_last_wal_replay_lsn() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_current_snapshot() TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.txid_snapshot_xmax(txid_snapshot) TO backup;
GRANT EXECUTE ON FUNCTION pg_catalog.pg_control_checkpoint() TO backup;
COMMIT;
```

```
BEGIN
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
COMMIT
```

```
=> \c student
```

```
You are now connected to database "student" as user "student".
```

---

2. Настройка журнала сообщений и политики удержания

Установим политику удержания единственной полной копии и вывод сообщений в журнал.

```
student$ pg_probackup set-config -B /var/probackup --instance ent-13 -d backup -U backup --retention-redundancy=1 --log-filename=probackup.log --log-level-file=info --log-level-console
```

Проверим настройки.

```
student$ pg_probackup show-config -B /var/probackup --instance ent-13
```

```
# Backup instance information
pgdata = /var/lib/pgpro/ent-13
system-identifier = 7323121610635147817
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = backup
pguser = backup
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = WARNING
log-level-file = INFO
log-format-console = PLAIN
log-format-file = PLAIN
log-filename = probackup.log
log-directory = /var/probackup/log
log-rotation-size = 0TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 1
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 1
# Remote access parameters
remote-proto = ssh
```

---

3. Полная копия

Формируем полную копию:

```
student$ pg_probackup backup -B /var/probackup --instance ent-13 -b FULL --stream
```

Проверим каталог копий.

```
student$ pg_probackup show -B /var/probackup --instance ent-13
```

```
========================================================================================================================================
 Instance  Version  ID      Recovery Time                Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN   Stop LSN    Status
========================================================================================================================================
 ent-13    13       S7DAY4  2024-01-16 21:49:16.992249+03  FULL  STREAM    1/0    5s  334MB  16MB    1,00  0/10000028  0/1000B998  OK
```

Еще одна полная копия.

```
student$ pg_probackup backup -B /var/probackup --instance ent-13 -b FULL --stream --delete-expired
```

Проверим каталог копий — предыдущая копия удалена.

```
student$ pg_probackup show -B /var/probackup --instance ent-13
```

```
========================================================================================================================================
 Instance  Version  ID      Recovery Time                  Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN   Stop LSN    Status
========================================================================================================================================
 ent-13    13       S7DAY9  2024-01-16 21:49:21.747031+03  FULL  STREAM    1/0    4s  334MB  16MB    1,00  0/11000028  0/110001C0  OK
```