

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 4.  
Расширенное резервное копирование.  
Компонент «pg\_ProBackup»

643.72410666.00067-07 98 01-04

Листов 151

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

В документе приведены сведения, необходимые для настройки компонента, предназначенного для создания резервных копий, контроля их целостности и восстановления информации в системе управления базами данных (СУБД).

Командная строка приведена в качестве примера практического применения относительно сущностей:

- каталог резервных копий – backup\_dir – C:\DIR;
- имя экземпляра – instance\_name – localdb;
- внешние каталоги для резервных копий – C:\test1 и C:\test2.



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 4.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».

Для СУБД «Jatoba» версии ядра 4 используется версия компонента — 2.5.12

Для СУБД «Jatoba» версии ядра 5 используется версия компонента — 2.5.14

Для СУБД «Jatoba» версии ядра 6 используется версия компонента — 2.5.15



### Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Описание синтаксиса сложных команд приводится в оригинале и в примере с переводом параметров. Например, описание команды определения копируемого экземпляра – add-instance:

Синтаксис команды в оригинале:

```
pg_probackup add-instance -B backup_dir -D data_dir --instance  
instance_name [remote_options]
```

Синтаксис команды в примере:

```
pg_probackup add-instance -B каталог_копий -D каталог_данных --  
instance имя_экземпляра [параметры_удаленного_режима]
```

Администратор СУБД «Jatoba» должен иметь навыки по работе с СУБД «PostgreSQL» или защищенной СУБД «Jatoba» (ООО «Газинформсервис»).

Степени важности примечаний, применяемые в документе:



**Важная информация** – указания, требующие особого внимания



**Дополнительная информация** – указания, позволяющие упростить работу с изделием

## СОДЕРЖАНИЕ

1. Назначение компонента.....	7
2. Синтаксис компонента.....	10
3. Ограничения при работе с компонентом .....	11
4. Установка компонента.....	12
4.1. Установка компонента в ОС семейства Windows .....	13
4.2. Установка компонента в ОС семейства GNU/Linux .....	14
4.2.1. Расположение конфигурационных файлов.....	15
4.3. Инициализация каталога резервных копий – init .....	16
4.4. Определение копируемого экземпляра – add-instance .....	18
4.5. Настройка СУБД для pg_ProBackup .....	21
4.5.1. Настройка непрерывного архивирования WAL .....	25
4.5.2. Настройка потокового резервного копирования .....	28
4.5.3. Настройка проверки целостности данных .....	31
4.5.4. Настройка частичного восстановления .....	32
5. Настройка pg_probackup .....	34
5.1. Указание параметров подключения .....	37
5.2. Управление каталогом резервных копий .....	37
5.2.1. Просмотр информации о резервных копиях.....	38
5.2.2. Просмотр оглавления архива WAL.....	42
5.3. Настройка политики хранения.....	45
5.3.1. Удаление ненужных копий .....	45
5.3.2. Закрепление резервных копий .....	48
5.3.3. Настройка политики хранения архива WAL .....	50
5.3.4. Объединение резервных копий.....	53
5.3.5. Удаление резервных копий .....	56
6. Настройка копирования PTRACK.....	60
7. Резервное копирование.....	63
7.1. Создание резервной копии.....	63
7.1.1. Режим ARCHIVE .....	64
7.1.2. Режим STREAM .....	67
7.1.3. Режим PAGE.....	70
7.1.4. Режим DELTA .....	72
7.1.5. Режим PTRACK.....	74
7.1.6. Архивирование внешних каталогов.....	75
7.2. Параметры команды backup – резервное копирование.....	76
8. Проверка целостности .....	81
8.1. Проверка страниц.....	81

8.2. Проверка целостности директории данных СУБД – checkdb .....	81
8.3. Проверка целостности резервных копий – validate .....	84
9. Восстановление резервной копии .....	86
9.1. Восстановление сервера баз данных .....	86
9.2. Инкрементальное восстановление .....	88
9.3. Частичное восстановление .....	90
9.4. Выполнение восстановления на момент времени (PITR) .....	91
9.5. Запуск pg_ProBackup в параллельных потоках .....	92
10. Справка по командной строке .....	93
10.1. Команды pg_ProBackup .....	93
10.1.1. Версия компонента – version .....	93
10.1.2. Вызов справочной информации по компоненту – help .....	93
10.1.3. Инициализация каталога резервных копий – init .....	94
10.1.4. Определение копируемого экземпляра – add-instance .....	94
10.1.5. Удаление резервных копий – del-instance .....	94
10.1.6. Редактирование конфигурационного файла pg_probackup.conf – set-config .....	95
10.1.7. Закрепление резервных копий – set-backup .....	104
10.1.8. Просмотр содержимого конфигурационного файла pg_probackup.conf – show-config .....	105
10.1.9. Просмотр содержания каталога копий – show .....	107
10.1.10. Резервное копирование – backup .....	107
10.1.11. Восстановление – restore .....	107
10.1.12. Проверка целостности директории данных СУБД – checkdb .....	110
10.1.13. Объединение (слияние) резервных копий – merge .....	110
10.1.14. Удаление резервных копий – delete .....	111
10.1.15. Архивирование – archive-push .....	111
10.1.16. Получение архива – archive-get .....	113
11. Использование pg_probackup в удаленном режиме .....	114
11.1. Параметры удаленного режима .....	116
11.2. Настройка удаленного режима .....	117
12. Параметры команд pg_ProBackup .....	119
12.1. Общие параметры .....	119
12.1.1. Параметры точки восстановления – recovery-target .....	120
12.1.2. Параметры сохранения – retention .....	121
12.1.3. Параметры закрепления резервных копий – set-bakup .....	123
12.1.4. Параметры ведения журнала – log .....	123
12.2. Параметры подключения .....	125
12.2.1. Параметры сжатия – compress .....	126
12.3. Параметры архивации .....	126
12.4. Параметры инкрементального восстановления .....	128

12.5. Параметры частичного восстановления .....	128
13. S3. Хранение данных в облачном хранилище.....	129
13.1. Предварительная настройка.....	130
13.2. Создание пользователя хранилища S3 .....	130
13.3. Создание корзины .....	133
13.4. Создание конфигурационного файла S3 для компонента pg_ProBackup .....	135
13.5. Инициализация каталога резервных копий с S3 .....	136
13.6. Добавление экземпляра СУБД в каталог резервных копий с S3 параметрами .....	137
13.7. Резервное копирование в хранилище S3 .....	139
13.7.1. Создание резервной копии FULL –stream в хранилище S3.....	140
13.7.2. Создание резервной копии DELTA –stream в хранилище S3.....	141
13.7.3. Вывод информации о резервных копиях в каталоге резервных копий S3.....	143
13.8. Восстановление сервера баз данных из хранилища S3.....	143
13.9. Прочие команды pg_ProBackup для S3.....	144
13.9.1. Удаление экземпляра СУБД из каталога резервных копий .....	144
13.9.2. Изменение конфигурационных параметров pg_probackup для заданного экземпляра СУБД ...	145
13.9.3. Изменение конфигурационных параметров pg_probackup для заданной резервной копии заданного экземпляра СУБД.....	147
Термины и определения .....	148
Перечень сокращений.....	150

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТА

Компонент pg\_ProBackup предназначен для управления резервным копированием и восстановлением баз данных СУБД «Jatoba», для регулярного создания резервных копий, позволяющих восстанавливать работу СУБД в случае аварийной ситуации, порчи или потери данных.

Для управления резервными копиями pg\_ProBackup создает каталог, в который сохраняются все файлы резервных копий с дополнительной метаинформацией, а также архивы журнала транзакций (WAL), необходимые для восстановления на выбранный момент времени. Компонент pg\_ProBackup хранит резервные копии разных экземпляров СУБД в отдельных подкаталогах, как представлено на рисунке 1.1.

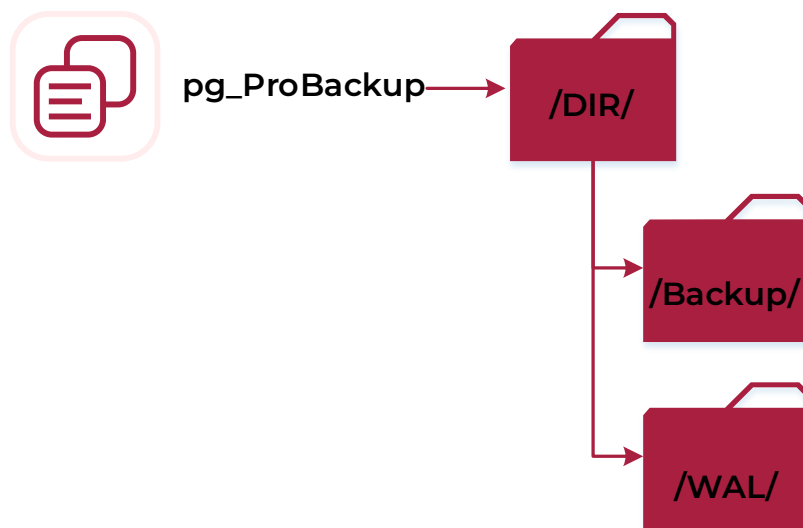


Рисунок 1.1 – Структура каталогов

Используя pg\_ProBackup, можно выполнить полное или инкрементальное резервное копирование:

- полные резервные копии содержат все файлы данных, необходимые для восстановления сервера баз данных с нулевой точки;
- инкрементальные копии создаются на уровне страниц данных и включают только ту информацию, которая изменилась со времени последнего резервного копирования.

Инкрементальное копирование позволяет экономить место на диске и создавать копии быстрее, чем при полном копировании.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Компонент pg\_ProBackup поддерживает следующие способы инкрементального копирования:

1) Разностное копирование (режим *DELTA*) – pg\_ProBackup считывает все файлы данных в каталоге и копирует только те страницы, которые изменились со времени предыдущего копирования. Для использования данного режима не требуется производить непрерывное архивирование.



В этом режиме объем ввода/вывода может равняться объему при полном резервном копировании.

2) Страничное копирование (режим *PAGE*) – pg\_ProBackup сканирует все файлы WAL в архиве с момента создания предыдущей полной или инкрементальной копии. Создаваемая резервная копия будет содержать только страницы, фигурирующие в записях WAL (при этом необходимо, чтобы в архиве были сохранены все файлы WAL, созданные при предыдущем копировании).



Если размер этих файлов сравним с общим размером файлов базы данных, ускорение будет менее значительным, но размер копии будет меньше.

3) Копирование изменений (режим *PTRACK*) – pg\_ProBackup отслеживает изменения страниц, и при каждом изменении страница таблицы помечается в специальной карте PTRACK. Для работы режима не требуется производить непрерывное архивирование.



Этот режим оказывает небольшое влияние на работу сервера, но значительно ускоряет инкрементальное резервное копирование.

Структура способов резервного копирования представлена на схеме 1.2.



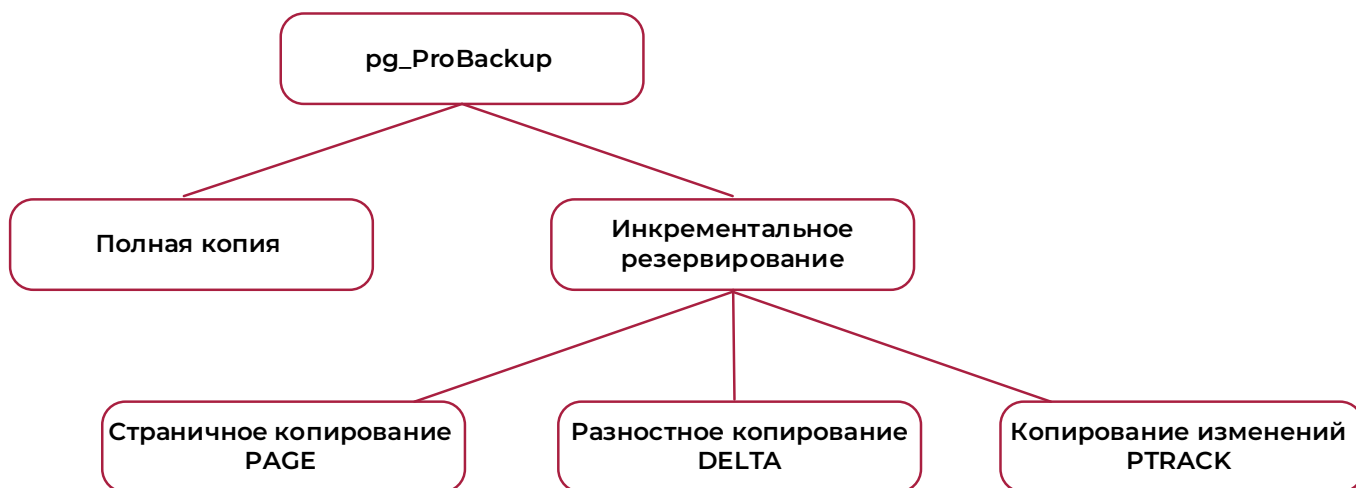


Рисунок 1.2 – Реализуемые способы резервного копирования

Для обеспечения целостности резервных копий, вне зависимости от выбранного способа копирования (FULL, PAGE или DELTA), требуется помимо сохранения файлов резервных копий, выполнять копирование архивов журнала транзакций (WAL).

Компонент pg\_ProBackup поддерживает следующие способы доставки WAL:

1) ARCHIVE (используется по умолчанию).

Режиме ARCHIVE целостность копий обеспечивается посредством непрерывного архивирования.

2) STREAM.

В режиме STREAM резервные копии включают в себя все файлы, необходимые для восстановления целостного состояния СУБД на момент создания копии. Вне зависимости от того, осуществляется ли непрерывное архивирование, необходимые для восстановления сегменты WAL считываются по протоколу потоковой репликации во время резервного копирования и включаются в состав резервной копии. Поэтому такие резервные копии называются автономными или самодостаточными.

## 2. СИНТАКСИС КОМПОНЕНТА


При работе с компонентом pg\_ProBackup используется следующий синтаксис:

```
pg_probackup version
pg_probackup help [command]
pg_probackup init -B backup_dir
pg_probackup add-instance -B backup_dir -D data_dir --instance
instance_name
pg_probackup del-instance -B backup_dir --instance
instance_name
pg_probackup set-config -B backup_dir --instance instance_name
[option...]
pg_probackup set-backup -B backup_dir --instance instance_name
-i backup_id [option...]
pg_probackup show-config -B backup_dir --instance instance_name
[--format=format]
pg_probackup show -B backup_dir [option...]
pg_probackup backup -B backup_dir --instance instance_name -b
backup_mode [option...]
pg_probackup restore -B backup_dir --instance instance_name
[option...]
pg_probackup checkdb -B backup_dir --instance instance_name -D
data_dir [option...]
pg_probackup validate -B backup_dir [option...]
pg_probackup merge -B backup_dir --instance instance_name -i
backup_id [option...]
pg_probackup delete -B backup_dir --instance instance_name { -i
backup_id | --delete-wal | --delete-expired | --merge-expired }
[option...]
pg_probackup archive-push -B backup_dir --instance
instance_name --wal-file-name wal_file_name [option...]
pg_probackup archive-get -B backup_dir --instance instance_name
--wal-file-path wal_file_path --wal-file-name wal_file_name
[option...]
pg_probackup catchup -b catchup_mode --source-pgdata =
path_to_pgdata_on_remote_server --destination-
pgdata=path_to_local_dir [option...]
```

### 3. ОГРАНИЧЕНИЯ ПРИ РАБОТЕ С КОМПОНЕНТОМ


При использовании компонента pg\_ProBackup необходимо учитывать следующие ограничения:


– в ОС семейства GNU/Linux копию базы данных возможно создавать только от имени того пользователя, который запускает сервер СУБД «Jatoba», т.е. от имени и с правами системной учетной записи «postgres»;

 Например, если служба или демон СУБД запускается от имени пользователя postgres, команду backup также должен выполнять пользователь postgres.

– на сервере СУБД, где была сделана резервная копия, и на сервере СУБД, где она будет выполняться восстановление из резервной копии, должна быть установлена одинаковая основная версия СУБД, и параметры *block\_size* и *wal\_block\_size* должны иметь одинаковые значения.

– компонент не обладает функциональной возможностью работы по схеме «pg\_ProBackup» → агент «pg\_ProBackup» → S3 хранилище.

 В зависимости от версии и конфигурации СУБД, могут накладываться дополнительные ограничения, например, по архитектуре процессора и версии libc/libicu.

 Для обеспечения корректного функционирования компонента pg\_ProBackup при взаимодействии с установленным антивирусным ПО необходимо добавить в список исключений в настройках антивирусного ПО каталог с резервными копиями. Если данный каталог является сетевым ресурсом, то необходимо добавить в исключения и на ЭВМ, где каталог расположен физически

#### 4. УСТАНОВКА КОМПОНЕНТА

Перед началом работы с компонентом pg\_ProBackup необходимо выполнить следующие действия:

- установить СУБД и компонент pg\_ProBackup;
- выполнить инициализацию каталога резервных копий;
- добавить копируемый экземпляр в каталог копий;
- настроить кластер баз данных для использования pg\_ProBackup;
- выполнить настройку компонента pg\_ProBackup.

Установка проводится на двух уровнях:

- на уровне операционной системы – непосредственная работа с компонентом pg\_ProBackup, используя командную строку;
- на уровне SQL-команд в СУБД.

На уровне операционной системы действия должны выполняться пользователем с административными привилегиями. В операционных системах семейства Windows может использоваться интерпретатор командной строки CMD.exe или PowerShell.



Запуск «командной строки» рекомендуется выполнять от имени административной учетной записи.



Одним из механизмов защиты СУБД является контроль полномочий системной учетной записи postgres в операционной системе.

В случае, если учетной записи postgres были добавлены привилегии, отличающиеся от установленных по умолчанию, СУБД при запуске блокируется.

Во избежание блокировки СУБД, запрещается добавлять дополнительные роли (привилегии) на уровне операционной системы системной учетной записи postgres.

#### 4.1. Установка компонента в ОС семейства Windows

Компонент pg\_ProBackup устанавливается в составе инсталляционного msi-файла СУБД «Jatoba» JatobaInstaller-X.XX.X-XXXX<sup>1</sup>.msi, находящегося на дистрибутивном диске.

При выборе режима установки «Полный», компонент pg\_ProBackup установится автоматически в составе СУБД.

При выборе режима установки «Выборочный», для установки компонента pg\_ProBackup потребуется в списке компонент выбрать пункт «Резервное копирование (pg\_probackup)», как представлено на рисунке 4.1.

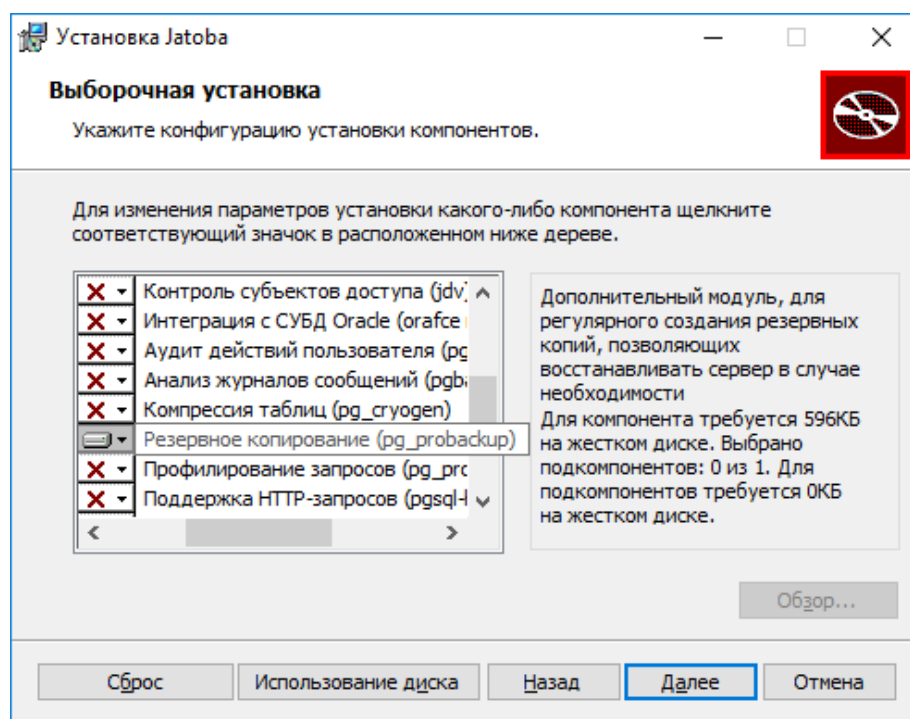


Рисунок 4.1 – Выбор компонента при установке

<sup>1</sup> Версия и название файла уточняется при поставке продукта.

## 4.2. Установка компонента в ОС семейства GNU/Linux

Версия компонента pg\_ProBackup для ОС семейства GNU/Linux, распространяется в составе СУБД в виде отдельного deb и rpm-пакета jatobaX-pg\_probackup\_X.XX.X-XXXX<sup>1</sup>\_amd64.\*\*\*, находящегося на дистрибутивном диске

Компонент pg\_ProBackup для выполнения резервного копирования в режиме PTRACK использует компонент ptrack и могут устанавливаться вместе.

Компоненты резервного копирования требуют базовой установки пакетов:

- jatoba4-common – клиентская часть СУБД;
- jatoba4-contrib – вспомогательный набор модулей (расширений СУБД);
- jatoba4-libs – основные библиотеки для клиентской и серверной части СУБД;
- jatoba4-server – серверная часть СУБД.

Команда установки пакетов в разных дистрибутивах Linux может отличаться, также могут отличаться некоторые шаги установки, связанные с отдельными особенностями дистрибутивов Linux и способами распространения продуктов третьих лиц. Описание команды установки дано в виде группы команд для четырех разных дистрибутивов Linux:

- 1) классический Debian Linux и все дистрибутивы от него производные:

```
apt-get install jatoba4-client jatoba4-contrib jatoba4-libs  
jatoba4-server jatoba4-pg-probackup jatoba4-ptrack
```

- 2) классический Red Hat и все дистрибутивы от него производные:

```
yum install jatoba4-client jatoba4-contrib jatoba4-libs jatoba4-  
server jatoba4-pg-probackup jatoba4-ptrack
```

- 3) ALTLinux – дистрибутив, построенный на базе RPM-пакетов, но использующий apt в качестве высокоуровневого менеджера пакетов (команда установки выглядит аналогично Debian):

```
apt-get install jatoba4-client jatoba4-contrib jatoba4-libs  
jatoba4-server jatoba4-pg-probackup jatoba4-ptrack
```

4) openSUSE – дистрибутив, построенный на базе RPM-пакетов, но использующий собственный пакетный менеджер zypper:

```
zypper install jatoba4-client jatoba4-contrib jatoba4-libs  
jatoba4-server jatoba4-pg-probackup jatoba4-ptrack
```

Все необходимые отличия, связанные с установкой в разных видах дистрибутивов Linux, также даны с пометкой соответствующей ОС.

Пример установки из локального репозитория приведены в Руководстве по установке.

#### 4.2.1. Расположение конфигурационных файлов

Перед началом работы с компонентом потребуются редактирование:

- конфигурационных файлов СУБД – postgresql.conf и pg\_hba.conf;
- конфигурационного файла компонента pg\_ProBackup – pg\_probackup.conf.

Конфигурационные файлы СУБД под управлением ОС семейства Windows располагаются по пути:

```
C:\Program Files\GIS\Jatoba\4\data
```

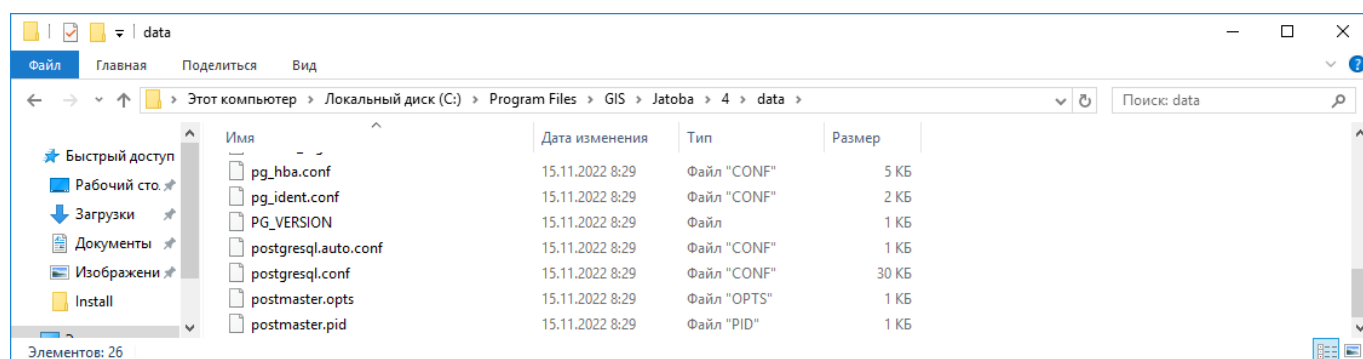


Рисунок 4.2 – Расположение конфигурационных файлов СУБД под управлением ОС Windows

Конфигурационные файлы СУБД под управлением ОС семейства GNU/Linux располагаются по пути:

```
var/lib/jatoba/4/data
```

.n	Name	Name
./		/pg_tblspc
/base		/pg_twophase
/global		/pg_wal
/log		/pg_xact
/pg_commit_ts		PG_VERSION
/pg_dynshmem		current_logfiles
/pg_logical		pg_hba.conf
/pg_multixact		pg_ident.conf
/pg_notify		postgres-uto.conf
/pg_replslot		postgresql.conf
/pg_serial		postmaster.opts
/pg_snapshots		postmaster.pid
/pg_stat		
/pg_stat_tmp		
/pg_subtrans		

Рисунок 4.3 – Расположение конфигурационных файлов СУБД под управлением ОС GNU/Linux

Конфигурационный файл `pg_probackup.conf` компонента `pg_ProBackup` располагается в директории хранения резервных копий. Редактирование файла осуществляется при помощи команды `set-config`.

Файл формируется после операции определения копируемого экземпляра, описанной в разделе 4.4 настоящего документа.

### 4.3. Инициализация каталога резервных копий – `init`

Компонент `pg_ProBackup` сохраняет все файлы копируемых данных и WAL в соответствующих подкаталогах каталога резервных копий.

Для инициализации каталога резервных копий в командной строке требуется выполнить команду:

```
pg_probackup init -B backup_dir
```

где `backup_dir` – каталог, предназначенный для хранения резервных копий.



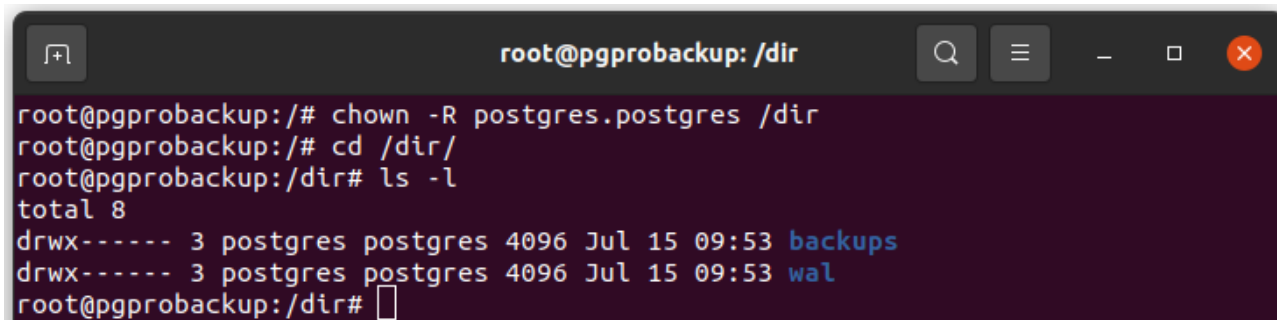
Для корректной работы `pg_ProBackup` каталог `backup_dir` должен быть пустым.

Пользователь, выполняющий запуск `pg_ProBackup`, должен иметь полный доступ к каталогу копий (`backup_dir`) на уровне операционной системы.



В представляемом примере, в GNU Linux права на каталог устанавливаются командой:

```
chown -R postgres.postgres /dir
```

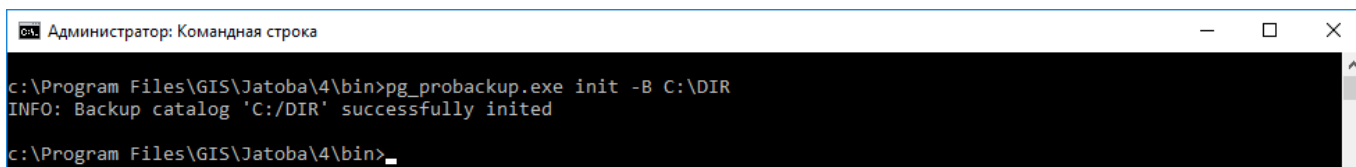


```
root@pgprobackup: /dir
root@pgprobackup:/# chown -R postgres.postgres /dir
root@pgprobackup:/# cd /dir/
root@pgprobackup:/dir# ls -l
total 8
drwx----- 3 postgres postgres 4096 Jul 15 09:53 backups
drwx----- 3 postgres postgres 4096 Jul 15 09:53 wal
root@pgprobackup:/dir#
```

Рисунок 4.4 – Команда установки и проверки прав на каталог в GNU Linux

При успешном выполнении команды, pg\_ProBackup выдаст сообщение «Backup catalog *backup\_dir* successfully inited», как показано на рисунке 4.5.

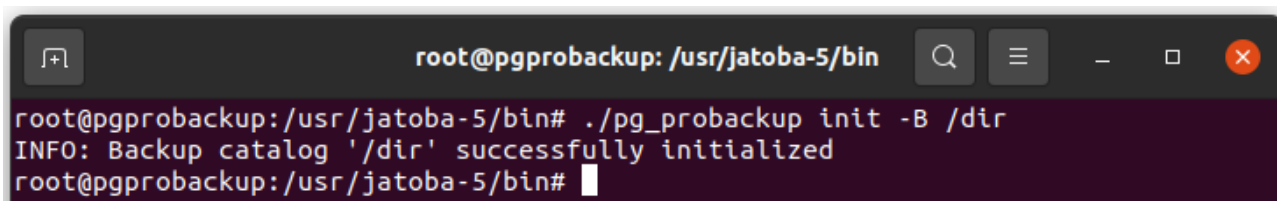
```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe init -B
C:\DIR
```



```
Администратор: Командная строка
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe init -B C:\DIR
INFO: Backup catalog 'C:/DIR' successfully inited
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 4.5 – Успешное выполнение инициализации каталога резервных копий

```
./pg_probackup init -B /dir
```



```
root@pgprobackup: /usr/jatoba-5/bin
root@pgprobackup:/usr/jatoba-5/bin# ./pg_probackup init -B /dir
INFO: Backup catalog '/dir' successfully initialized
root@pgprobackup:/usr/jatoba-5/bin#
```

Рисунок 4.6– Успешное выполнение инициализации каталога резервных копий в GNU Linux

Компонент pg\_ProBackup создает каталог копий со следующими подкаталогами:

- wal/ – каталог для файлов WAL;
- backups/ – каталог для файлов резервных копий.

После выполнения инициализации каталога резервных копий, становится возможным добавление определения копируемого экземпляра.

#### 4.4. Определение копируемого экземпляра – add-instance

Компонент pg\_ProBackup имеет функциональную возможность сохранять резервные копии разных экземпляров СУБД в одном каталоге резервных копий. Для создания различных подкаталогов требуется определить копируемый экземпляр в каталоге копий для каждой СУБД.

Для определения копируемого экземпляра потребуется запустить командную строку или терминал ОС и выполнить команду:

```
pg_probackup add-instance -B backup_dir -D data_dir --instance  
instance_name [remote_options]
```

В транскрипции команда имеет вид:

```
pg_probackup add-instance -B каталог_копий -D каталог_данных --  
instance имя_экземпляра [параметры_удаленного_режима]  
где:
```

где каталог\_данных (data\_dir) – каталог, содержащий данные СУБД, копию которого нужно сделать;



Для подготовки и использования pg\_ProBackup необходимо иметь право записи в этот каталог.

*имя\_экземпляра (instance\_name)* – имя подкаталога, в котором будут храниться файлы копируемых данных и WAL для этого экземпляра СУБД;

*параметры\_удаленного\_режима (remote\_options)* – параметры, которые задаются дополнительно, если каталог\_данных располагается удаленно.

В случае возникновения ошибки, следует запустить командную строку ОС от имени пользователя с административными правами.

В случае успешного выполнения команды появится сообщение, приведенное на рисунке 4.7.

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe add-instance  
-B C:\DIR -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb
```

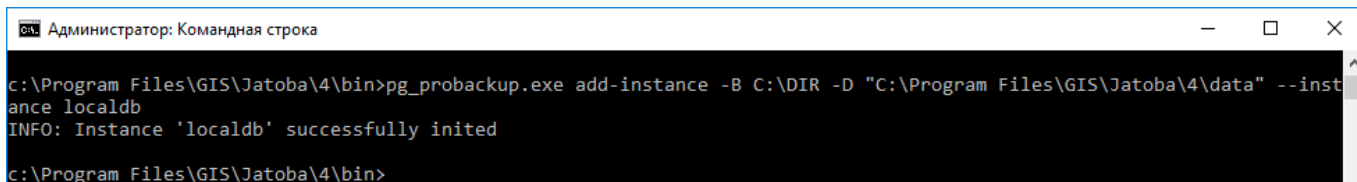


Рисунок 4.7 – Выполнение определения копируемого экземпляра

```
./pg_probackup add-instance -B /dir -D /var/lib/jatoba/5/data -  
-instance localdb
```

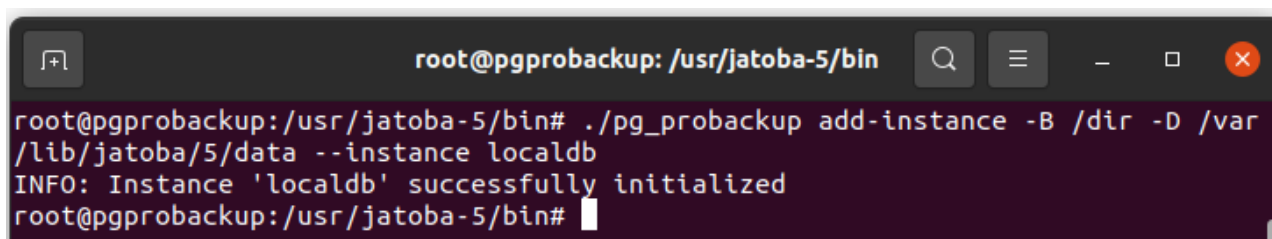


Рисунок 4.8 – Выполнение определения копируемого экземпляра в GNU Linux

pg\_ProBackup создает подкаталоги с именем экземпляра (*instance\_name*) в каталогах backups/ и wal/. Каталог «backups/*instance\_name*» содержит файл конфигурации pg\_probackup.conf с параметрами pg\_ProBackup, относящимися к данному экземпляру.

При передаче команде параметров удаленного режима, они будут добавлены в pg\_probackup.conf. Расположение файла pg\_probackup.conf представлено на рисунке 4.9.

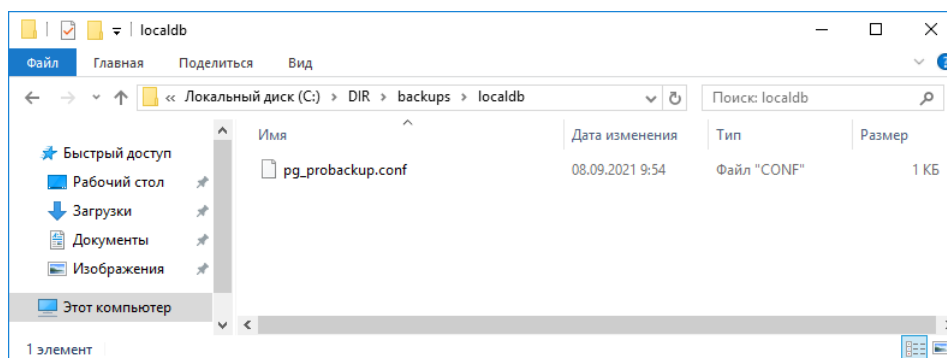


Рисунок 4.9 – Расположение конфигурационного файла pg\_probackup.conf

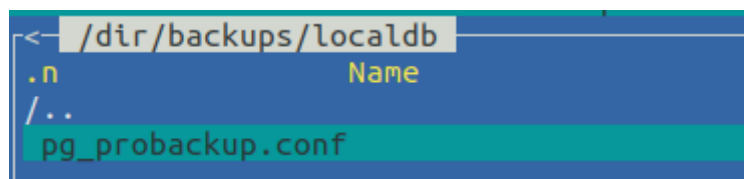


Рисунок 4.10 – Расположение конфигурационного файла pg\_probackup.conf в GNU Linux

Пользователь, запускающий pg\_ProBackup, должен иметь полный доступ к каталогу копий (*backup\_dir*) и, как минимум, доступ на чтение всего содержимого каталога данных (*data\_dir*). Если задавать путь к каталогу копий в переменные окружения BACKUP\_PATH, соответствующий параметр в командах pg\_ProBackup можно не указывать.

#### 4.5. Настройка СУБД для pg\_ProBackup

Для использования компонента pg\_ProBackup необходимо создать пользователя СУБД backup с атрибутом Login и придерживаться принципа минимизации назначаемых атрибутов и привилегий. Назначение дополнительных атрибутов роли, системных привилегий и прав осуществляется в соответствии с выбранным типом и режимом резервного копирования.

Для настройки pg\_ProBackup потребуется:

1) войти через утилиту psql в СУБД от имени и с правами пользователя, обладающим привилегиями создания ролей и назначения системных привилегий, и в командной строке psql выполнить следующие действия в базе данных, к которой будет производится подключение пользователем backup:



Для СУБД «Jatoba» версии ядра 4 выполняются следующие действия:

```
BEGIN;
CREATE ROLE backup WITH LOGIN;
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.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;
```



Для СУБД «Jatoba» версии ядра 5 и выше, выполняются следующие действия:

```
BEGIN;  
CREATE ROLE backup WITH LOGIN;  
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_backup_start(text,  
boolean) TO backup;  
GRANT EXECUTE ON FUNCTION pg_catalog.pg_backup_stop(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;
```

Для удобства выполнения операций можно воспользоваться любой платформой для администрирования СУБД (см. пример на рисунке 4.11).

```

Администратор: Командная строка - psql -h localhost -d postgres -U postgres
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U postgres
Пароль пользователя postgres:
psql (14.5)
Введите "help", чтобы получить справку.

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

Рисунок 4.11 – Окно выполнения SQL-команд

- 2) установить пароль для пользователя backup SQL-командой:

```
ALTER USER backup with password 'P@ssword';
```

- 3) в конфигурационном файле pg\_hba.conf разрешить подключение к СУБД пользователю с именем backup.

```


edit pg_hba.conf - Far 3.0.5700.0 x86 Administrator
C:\...\Jatoba\4\data\pg_hba.conf [*] ANSI Ln 111/112 Col 1 Ch 1 18:14 ^
# Allow replication connections from localhost, by a user with the
# replication privilege.
host replication backup 127.0.0.1/32 md5
local replication all md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
  
```

Рисунок 4.12 – Дополнительная строка в конфигурационном файле pg\_hba.conf

```

91 # Allow replication connections from localhost, by a user with the
92 # replication privilege.
93 host      replication    backup      127.0.0.1/32      md5
94 local     replication    all         ::1/128           md5
95 host      replication    all         127.0.0.1/32      md5
96 host      replication    all         ::1/128           md5
  
```

Рисунок 4.13 – Дополнительная строка в конфигурационном файле pg\_hba.conf в GNU Linux

 Компонент pg\_ProBackup читает непосредственно файлы СУБД, поэтому запускать pg\_ProBackup (или подключаться к нему удаленно) нужно от имени пользователя ОС, который имеет доступ на чтение всех файлов и каталогов внутри каталога данных (PGDATA), подлежащего копированию.

Результаты выполняемых действий отражены на схеме 4.14.

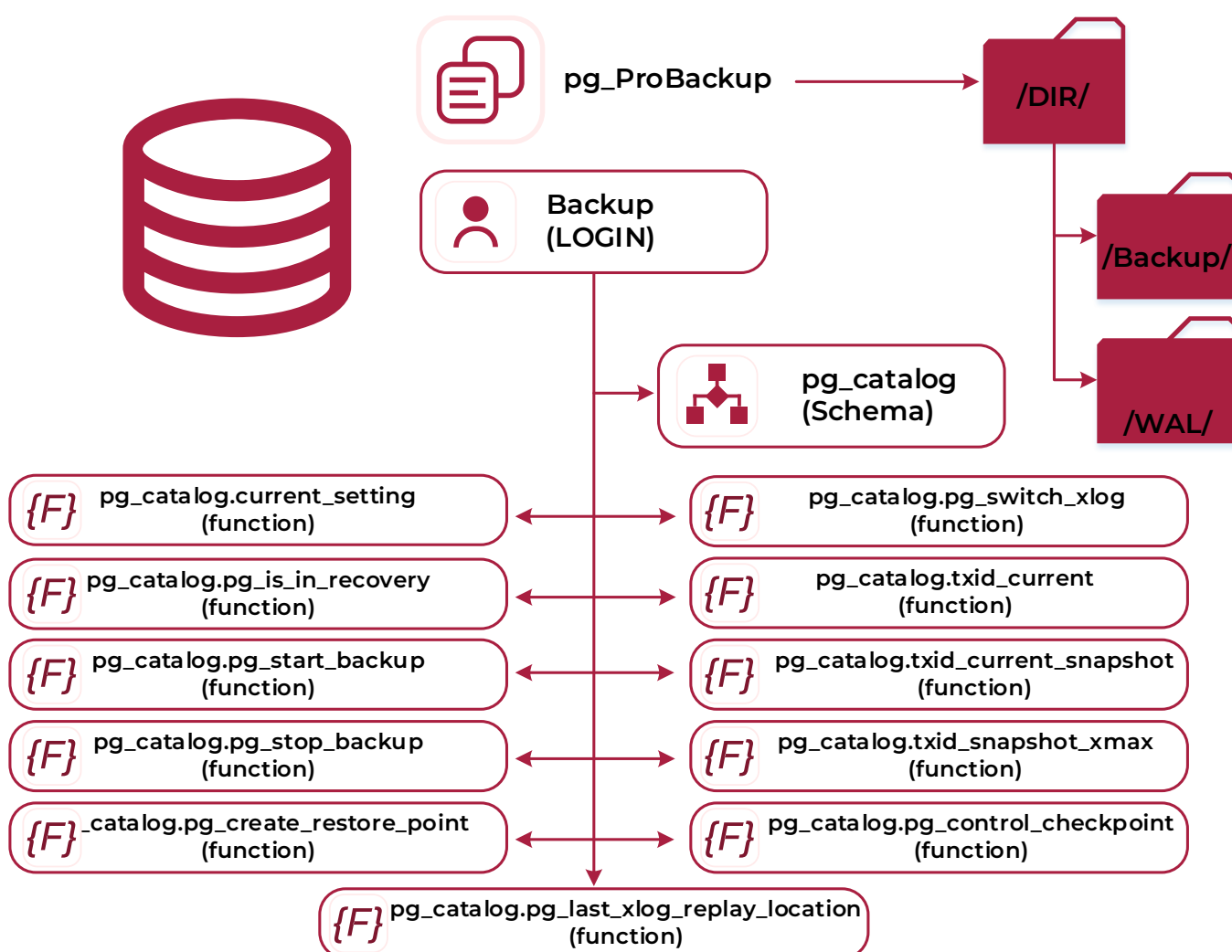




Рисунок 4.14 – Настройка СУБД для компонента pg\_probackup

#### 4.5.1. Настройка непрерывного архивирования WAL

Для выполнения копирования в режиме PAGE, создания резервных копий с использованием режима доставки WAL ARCHIVE и восстановления инкрементальной копии после смены линии времени, должно осуществляться непрерывное архивирование WAL.

Для настройки непрерывного архивирования WAL необходимо выполнить следующие действия:

- 1) открыть для редактирования конфигурационный файл postgresql.conf;
- 2) в разделе «Write-Ahead Log» – «Settings» задать для параметра wal\_level значение выше replica или logical;

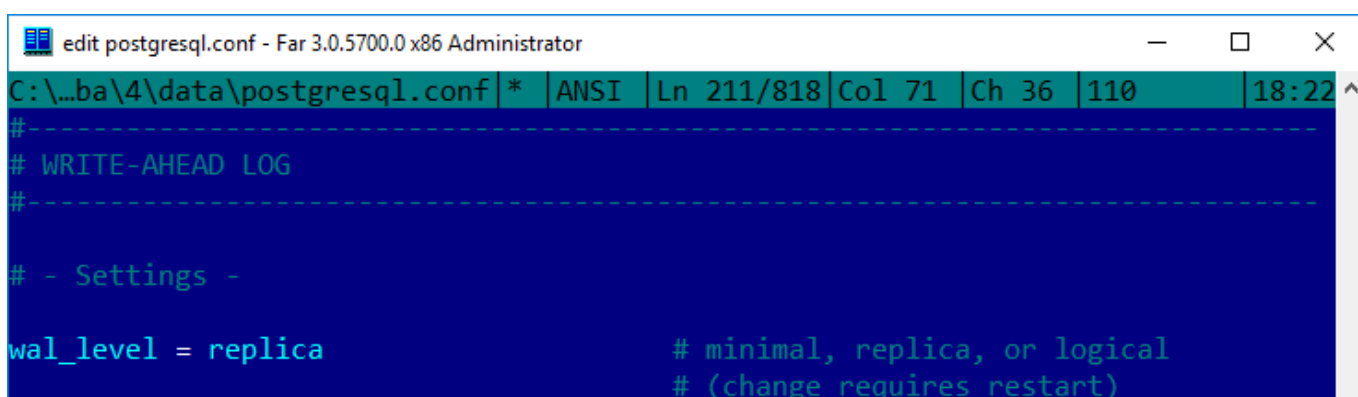


Рисунок 4.15 – Внесение параметра wal\_level в конфигурационный файл postgresql.conf

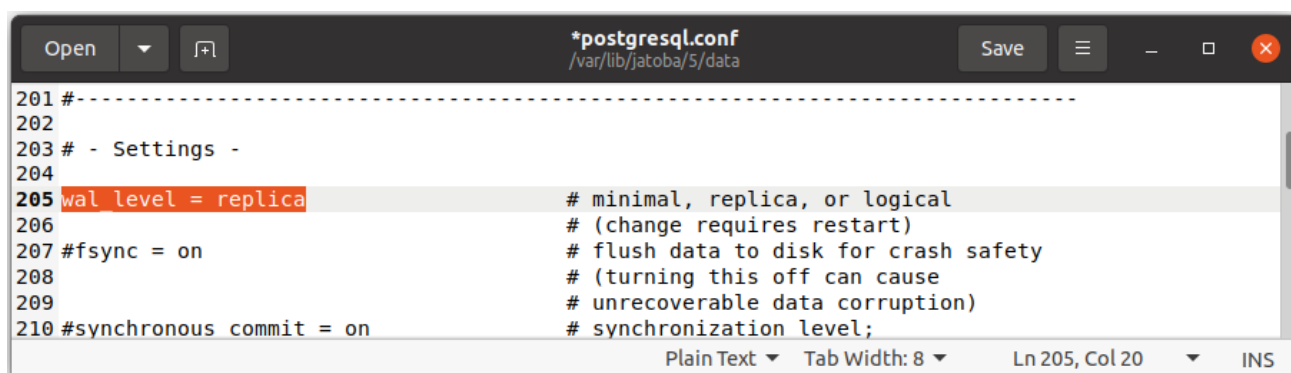


Рисунок 4.16 – Внесение параметра wal\_level в конфигурационный файл postgresql.conf в GNU Linux



При настройке резервного копирования на ведущем сервере, параметр archive\_mode должен иметь значение on или always.

Для выполнения резервного копирования на ведомом сервере параметр `archive_mode` должен иметь значение `always`.

- 3) в разделе «Archiving» установить параметр:

```
archive_mode = on
```

- 4) в разделе «Archiving» установить параметр `archive_command`:

```
archive_command = '"install_dir/pg_probackup" archive-push -B  
"backup_dir" --instance instance_name --wal-file-name=%f  
[remote_options]'
```

Строка будет иметь вид:

- В ОС Windows:

```
archive_command = '"C:/Program  
Files/GIS/Jatoba/4/bin/pg_probackup.exe" archive-push -B C:/dir  
--instance localdb --wal-file-name=%f'
```



В указании пути каталогов используется не обратная косая черта (backslash) «\», а косая черта (forward slash) «/».

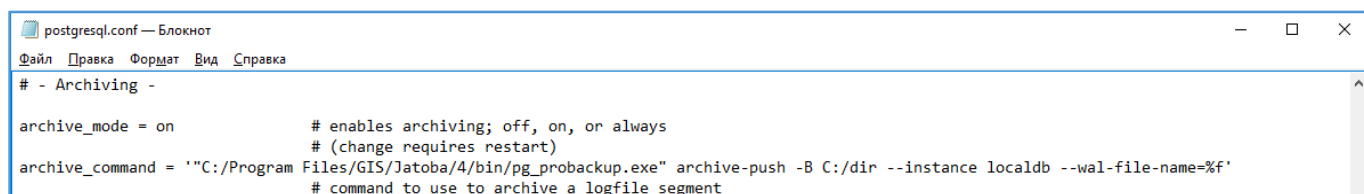


Рисунок 4.17 – Вид конфигурационного файла `postgresql.conf` с внесенной строчкой параметра `archive_command`

- в GNU Linux:

```
archive_command = '"usr/jatoba-5/bin/pg_probackup" archive-  
push -B /dir --instance localdb --wal-file-name=%f'
```

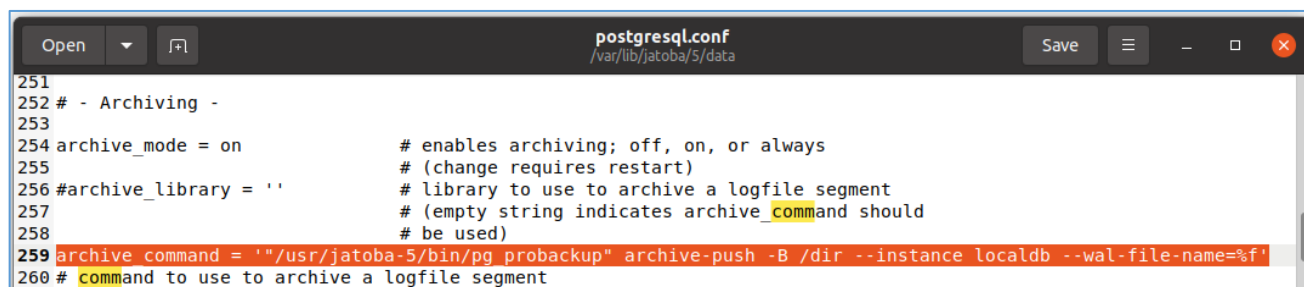


Рисунок 4.18 – Вид конфигурационного файла postgresql.conf с внесенной строчкой параметра archive\_command в GNU Linux

Где:

- *install\_dir* – путь к компоненту pg\_probackup;
- *backup\_dir* – путь к каталогу резервных копий (см. п. 4.3);
- *instance\_name* – имя копируемого экземпляра СУБД (см. п. 4.4).

В итоге при запуске резервного копирования в каталоге *backup\_dir\wal\instance\_name* появятся резервные копии WAL, как представлено на рисунке 4.19.

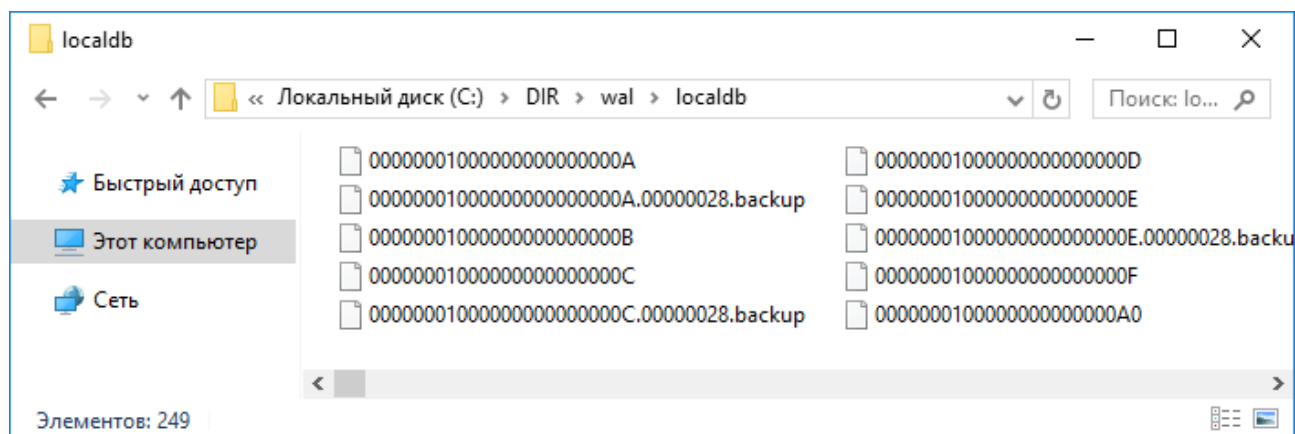


Рисунок 4.19 – Структура каталога *backup\_dir\wal\instance\_name*

После подготовительных действий можно использовать режим доставки WAL ARCHIVE и делать резервные копии в режиме PAGE, а также выполнять восстановление на момент времени.

Текущее состояние архива WAL можно просмотреть, воспользовавшись командой *show*.

При выполнении страничного копирования, с использованием режима доставки WAL ARCHIVE, при недостаточной транзакционной активности, заполнение очередного сегмента WAL может выполняться продолжительный период времени. Для ограничения

времени ожидания, нужно воспользоваться параметром СУБД *archive\_timeout* на ведущем сервере.



Вместо использования команды `pg_probackup archive-push` можно воспользоваться любым другим средством, при условии, что в процессе непрерывного архивирования, сегменты WAL будут попадать в каталог `backup_dir/wall/instance_name`. Для сжатия сегментов, если в нем есть потребность, должен использоваться алгоритм `gzip`, а сжатые файлы сегментов должны иметь расширение `.gz`.

#### 4.5.2. Настройка потокового резервного копирования

Для настройки потокового резервного копирования, необходимо выполнить следующие действия:

- 1) установить атрибут `REPLICATION` роли `backup`:

```
ALTER ROLE backup WITH REPLICATION;
```

- 2) в конфигурационном файле `pg_hba.conf` разрешить выполнение репликации для роли `backup`:

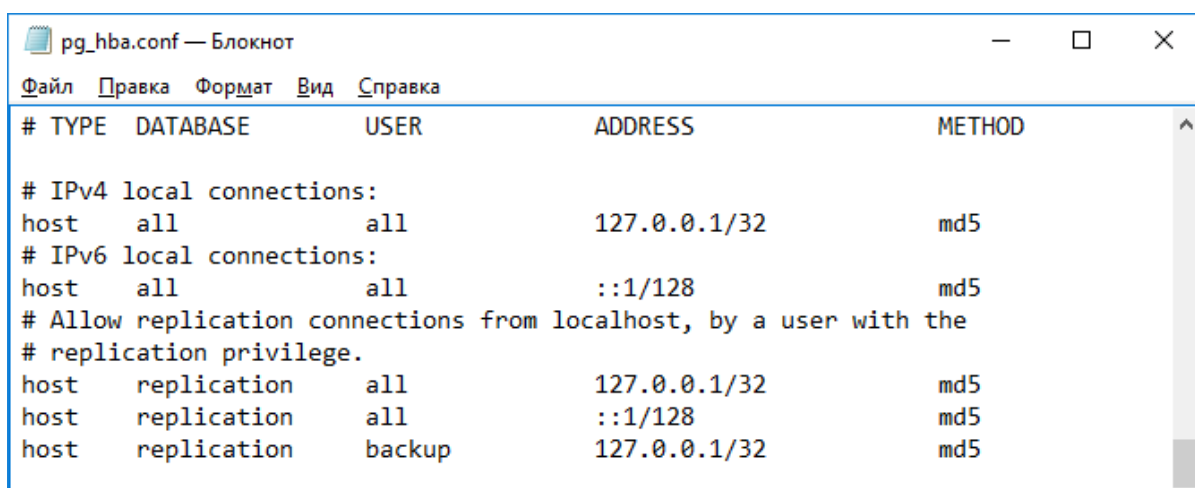


Рисунок 4.20 – Внесение изменений в конфигурационный файл `pg_hba.conf`

- 3) в конфигурационном файле `postgresql.conf` установить для параметра `max_wal_senders` достаточно большое значение, предусматривающее минимум одно подключение для процесса резервного копирования;

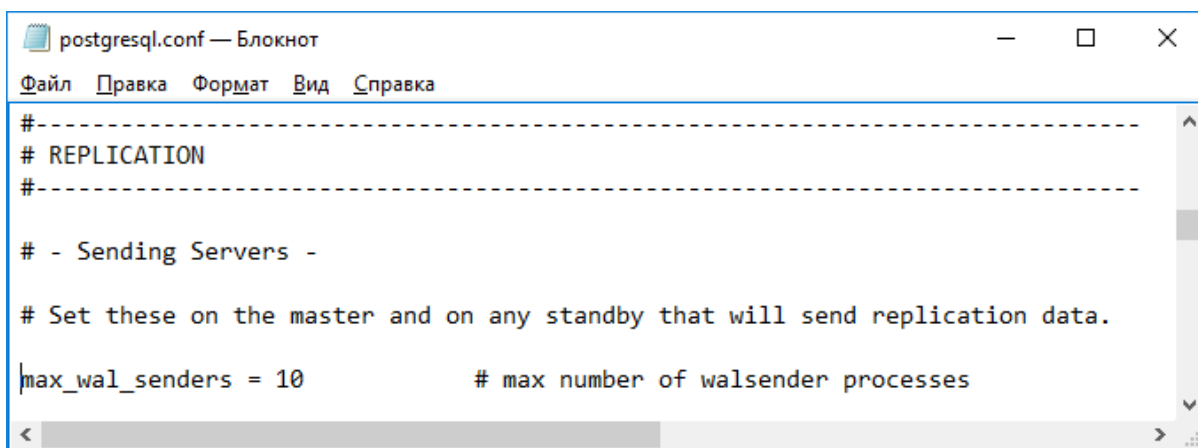


Рисунок 4.21 – Установка значения max\_wal\_senders

4) в том же конфигурационном файле задать для параметра wal\_level значение replica или logical.

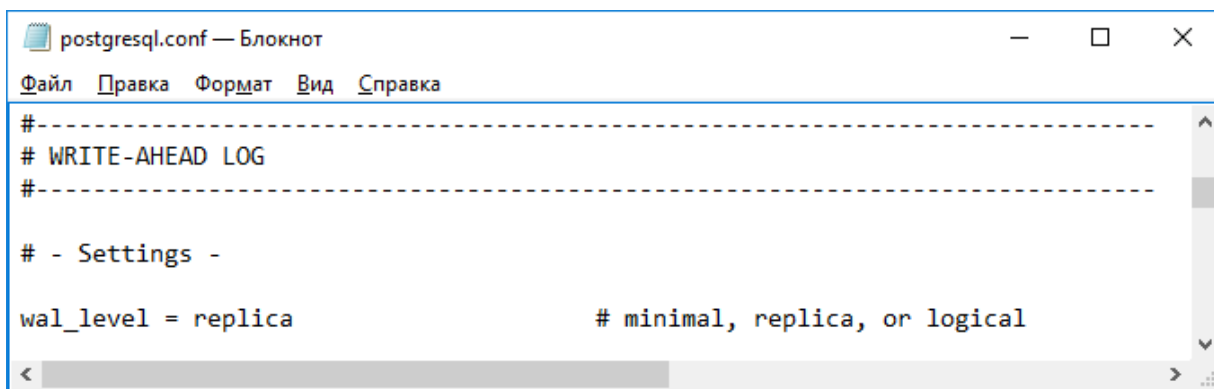


Рисунок 4.22 – Установка значения wal\_level

При выполнении страничного резервного копирования в потоковом режиме или восстановления на момент времени с потоковыми копиями необходимо настроить архивирование WAL в соответствии с описанием, приведенным в подразделе 4.5.1.

После этих подготовительных действий становится доступно создание резервных копий в режимах FULL, PAGE, DELTA и PTRACK, с использованием потоковой доставки WAL.

Результаты выполняемых действий отражены на схеме 4.23.

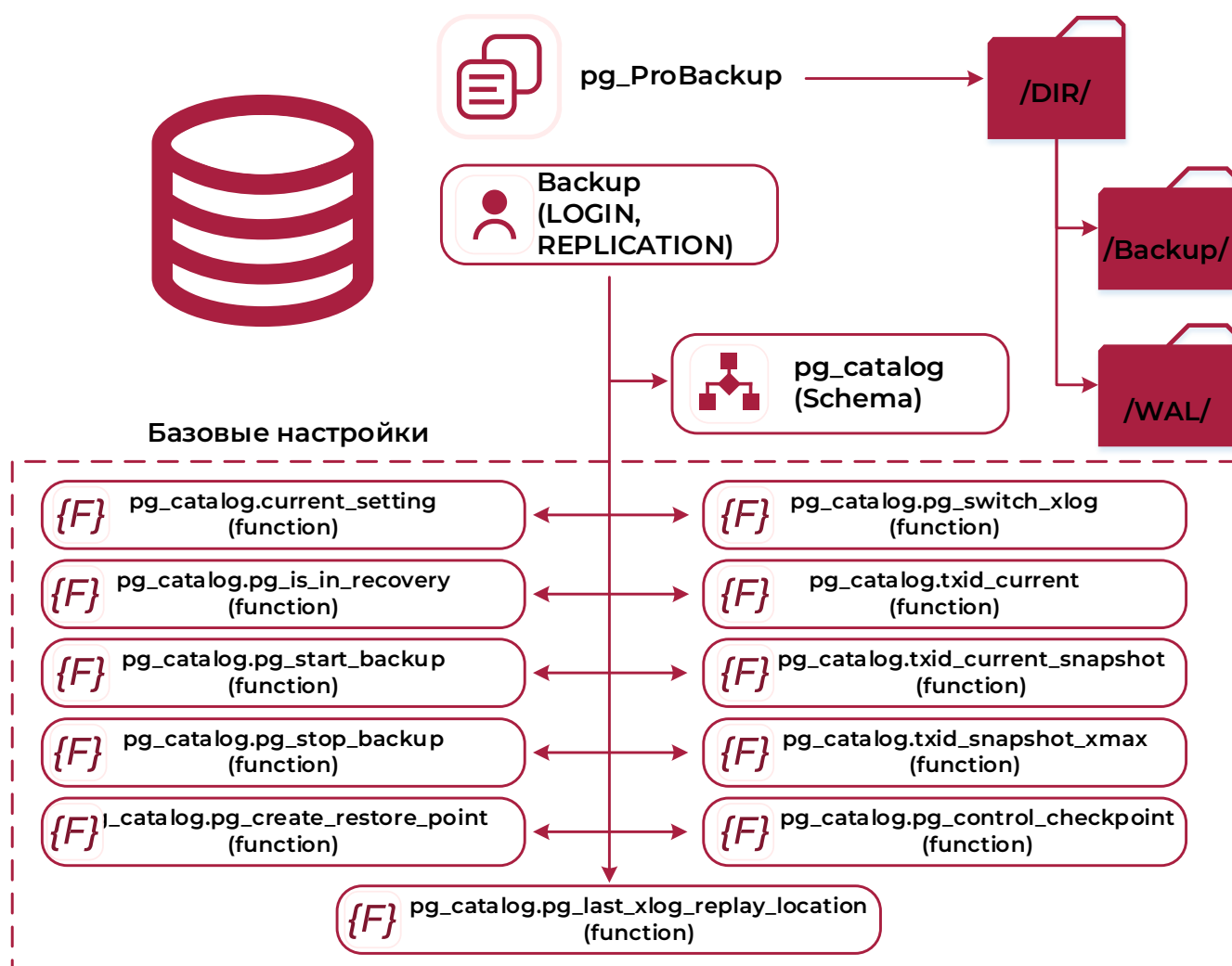


Рисунок 4.23 – Настройка потокового резервного копирования



Файл .pgpass может использоваться для прохождения аутентификации при выполнении копирования в потоковом режиме, файл .pgpass должен содержать учетные данные для подключения к базе данных replication.

Например:

pgghost:5432:replication:backup\_user:my\_strong\_password

### 4.5.3. Настройка проверки целостности данных

Для выполнения проверки целостности данных БД требуется произвести следующие дополнительные настройки (предполагается, что настройка производится для роли backup):

- 1) установить расширение amcheck в каждой базе:

```
CREATE EXTENSION amcheck;
```

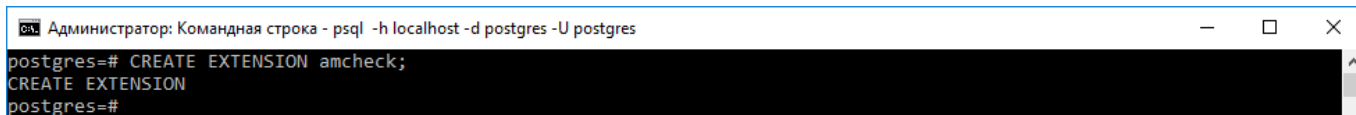


Рисунок 4.24 – Установка расширения amcheck

- 2) Дать роли backup в каждой базе данных кластера следующие права:

```
GRANT SELECT ON TABLE pg_catalog.pg_am TO backup;
GRANT SELECT ON TABLE pg_catalog.pg_class TO backup;
GRANT SELECT ON TABLE pg_catalog.pg_database TO backup;
GRANT SELECT ON TABLE pg_catalog.pg_namespace TO backup;
GRANT SELECT ON TABLE pg_catalog.pg_extension TO backup;
GRANT EXECUTE ON FUNCTION bt_index_check(regclass) TO backup;
GRANT EXECUTE ON FUNCTION bt_index_check(regclass, bool) TO backup;
```

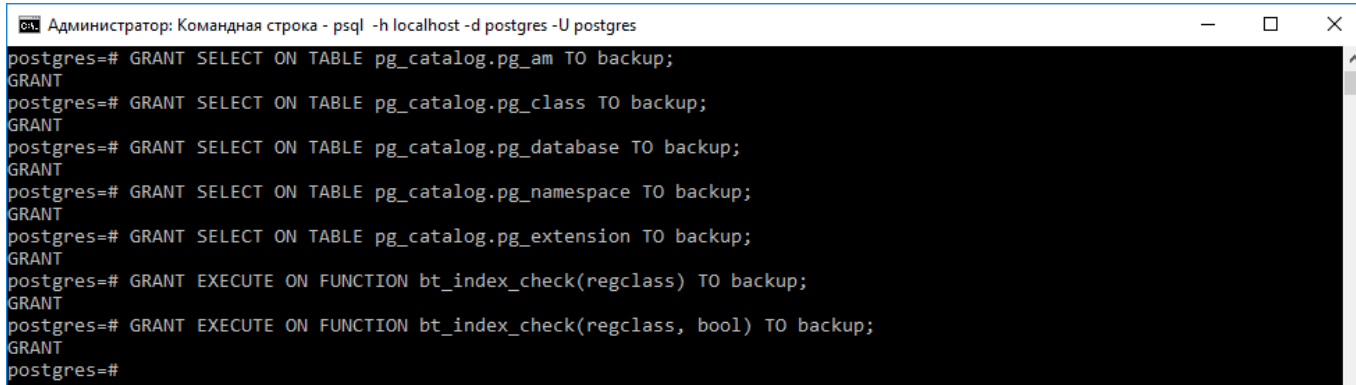


Рисунок 4.25 – Предоставление роли backup права использования таблиц и функций расширения amcheck

Результаты выполняемых действий отражены на схеме 4.26.

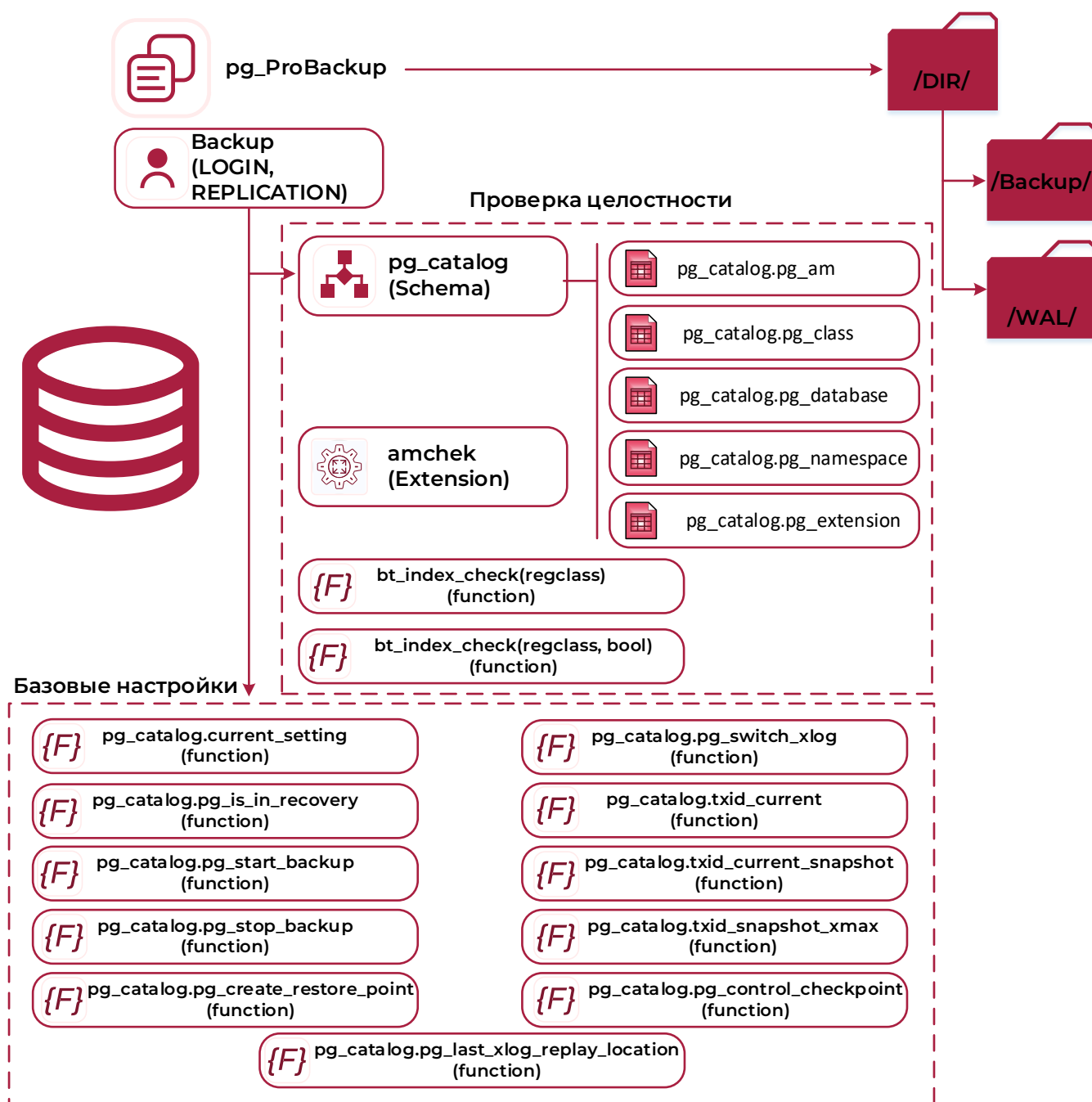


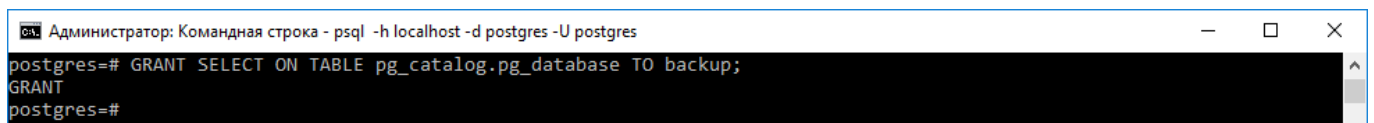
Рисунок 4.26 – Настройка проверки целостности

#### 4.5.4. Настройка частичного восстановления

Для осуществления частичного восстановления потребуется предоставить учетной записи backup право на чтение pg\_catalog.pg\_database в той базе данных, через которую выполняется подключение к серверу СУБД, командой:

```
GRANT SELECT ON TABLE pg_catalog.pg_database TO backup;
```





```
Администратор: Командная строка - psql -h localhost -d postgres -U postgres
postgres=# GRANT SELECT ON TABLE pg_catalog.pg_database TO backup;
GRANT
postgres=#
```

Рисунок 4.27 – Предоставление права на чтение

## 5. НАСТРОЙКА PG\_PROBACKUP

По завершении выполнения инициализации каталога резервных копий и добавления определения копируемого экземпляра, возможно приступить к настройке конфигурации pg\_ProBackup с помощью файла pg\_probackup.conf в каталоге:

```
backup_dir/wall/instance_name.
```

Например, чтобы не задавать [параметры подключения](#) при выполнении команд [backup](#) и [checkdb](#) в командной строке, возможно определить их в файле конфигурации pg\_probackup.conf с помощью команды

```
set-config
```



Редактировать pg\_probackup.conf вручную не рекомендуется.

Изначально конфигурационный файл pg\_probackup.conf содержит следующие параметры:

PGDATA – путь к каталогу данных кластера, который будет копироваться.

system-identifier – уникальный идентификатор экземпляра СУБД.

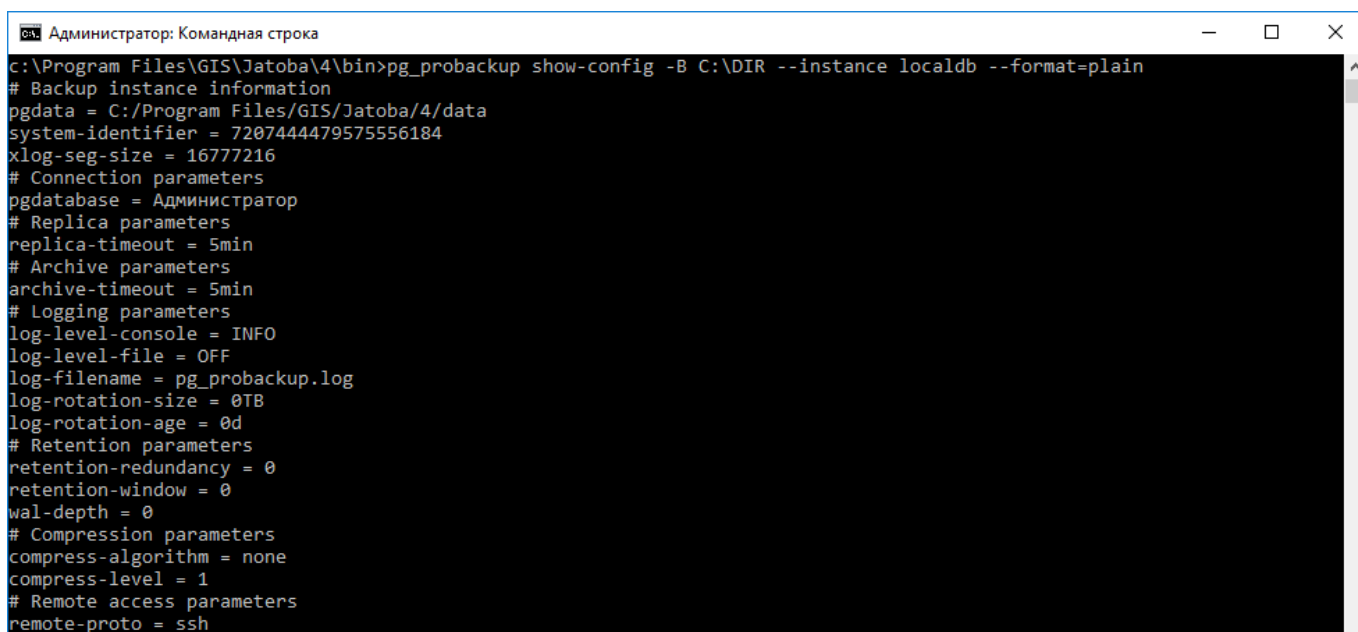
Для просмотра текущих параметров нужно выполнить команду:

```
pg_probackup show-config -B backup_dir --instance instance_name  
[--format=plain|json]
```

Команда может быть выполнена:

- в формате текстового SQL-скрипта:
  - В ОС Windows:

```
pg_probackup show-config -B C:\DIR --instance localdb --  
format=plain
```

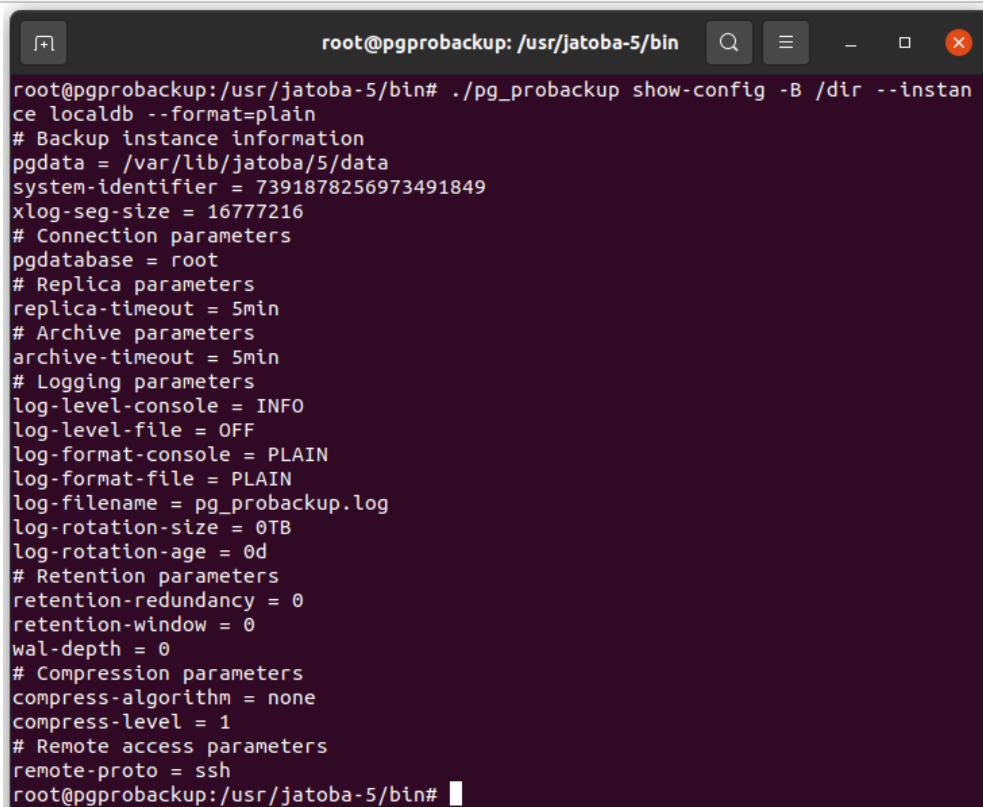


```
Администратор: Командная строка
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show-config -B C:\DIR --instance localdb --format=plain
# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 7207444479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = Администратор
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = OFF
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
```

Рисунок 5.1 – Вывод параметров конфигурационного файла pg\_probackup.conf

- В GNU Linux:

```
./pg_probackup show-config -B /dir --instance localdb --format=plain
```



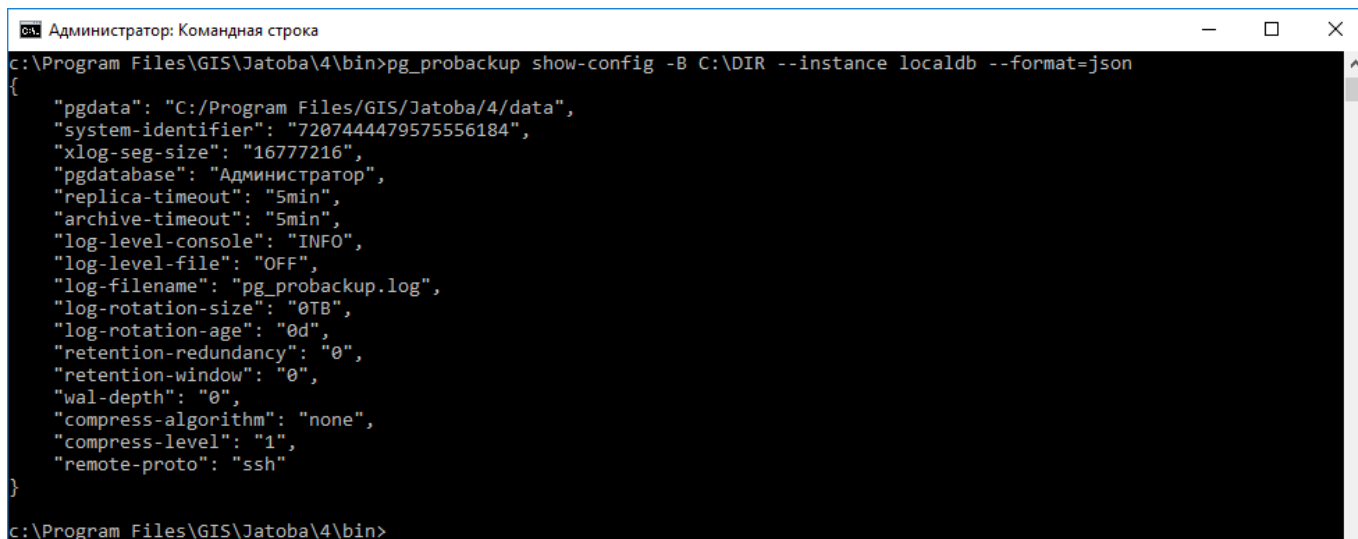
```
root@pgprobackup: /usr/jatoba-5/bin# ./pg_probackup show-config -B /dir --instance localdb --format=plain
# Backup instance information
pgdata = /var/lib/jatoba/5/data
system-identifier = 7391878256973491849
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = root
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = OFF
log-format-console = 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
root@pgprobackup: /usr/jatoba-5/bin#
```

Рисунок 5.2 Вывод параметров конфигурационного файла pg\_probackup.conf в GNU Linux

– в формате JSON:

○ В ОС Windows:

```
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show-config -B  
C:\DIR --instance localdb --format=json
```

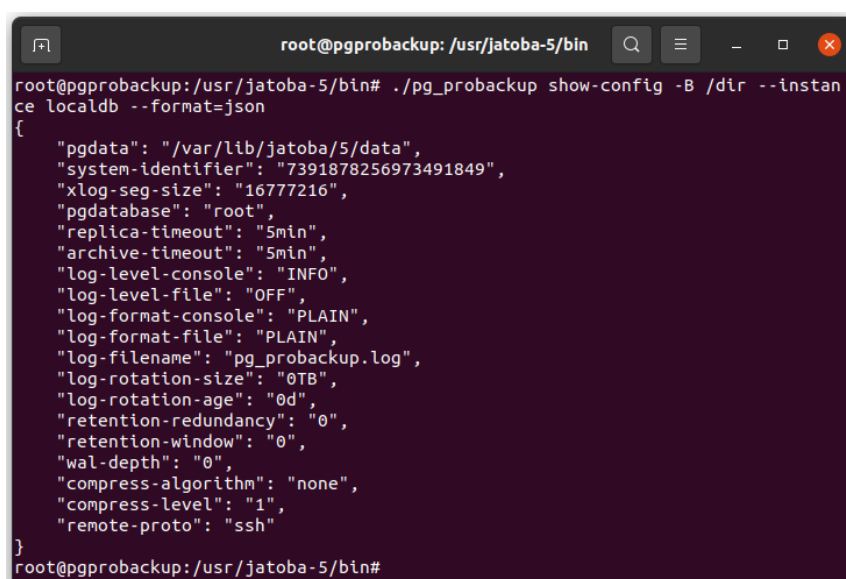


```
Администратор: Командная строка  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show-config -B C:\DIR --instance localdb --format=json  
{  
  "pgdata": "C:/Program Files/GIS/Jatoba/4/data",  
  "system-identifier": "7207444479575556184",  
  "xlog-seg-size": "16777216",  
  "pgdatabase": "Администратор",  
  "replica-timeout": "5min",  
  "archive-timeout": "5min",  
  "log-level-console": "INFO",  
  "log-level-file": "OFF",  
  "log-filename": "pg_probackup.log",  
  "log-rotation-size": "0TB",  
  "log-rotation-age": "0d",  
  "retention-redundancy": "0",  
  "retention-window": "0",  
  "wal-depth": "0",  
  "compress-algorithm": "none",  
  "compress-level": "1",  
  "remote-proto": "ssh"  
}
```

Рисунок 5.3 – Вывод параметров конфигурационного файла pg\_probackup.conf в формате JSON

○ В GNU Linux:

```
./pg_probackup show-config -B /dir --instance localdb --  
format=json
```



```
root@pgprobackup: /usr/jatoba-5/bin  
root@pgprobackup: /usr/jatoba-5/bin# ./pg_probackup show-config -B /dir --instance localdb --format=json  
{  
  "pgdata": "/var/lib/jatoba/5/data",  
  "system-identifier": "7391878256973491849",  
  "xlog-seg-size": "16777216",  
  "pgdatabase": "root",  
  "replica-timeout": "5min",  
  "archive-timeout": "5min",  
  "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-redundancy": "0",  
  "retention-window": "0",  
  "wal-depth": "0",  
  "compress-algorithm": "none",  
  "compress-level": "1",  
  "remote-proto": "ssh"  
}
```

Рисунок 5.4 - Вывод параметров конфигурационного файла pg\_probackup.conf в формате JSON в GNU Linux

Параметры, заданные в `pg_probackup.conf`, можно переопределить, задавая соответствующие переменные окружения или параметры командной строки при вызове команд `pg_ProBackup`.

### 5.1. Указание параметров подключения

При определении свойства соединения в файле конфигурации `pg_probackup.conf`, можно не указывать параметры соединения во всех последующих командах `pg_ProBackup`. Однако, если установлены соответствующие переменные окружения, они имеют больший приоритет. Параметры, заданные в командной строке, переопределяют как переменные окружения, так и свойства в файле конфигурации.

Если параметры подключения не заданы, используются значения по умолчанию. В частности, при подключении компонента `pg_ProBackup` к Unix-сокету локального сервера (или к `localhost` в Windows), в качестве имени базы данных и имени пользователя `pg_ProBackup` выбирает значение переменной окружения `PGUSER` или имя текущего пользователя ОС.

### 5.2. Управление каталогом резервных копий

С помощью `pg_ProBackup` можно управлять резервными копиями в командной строке:

- просматривать имеющиеся резервные копии;
- просматривать оглавление архива WAL;
- выполнять проверку резервных копий;
- настраивать политики хранения резервных копий:
  - удалять ненужных копии;
  - закреплять резервные копии;
- настраивать политики хранения архива WAL:
  - объединять резервные копии;
  - удалять резервные копии.

### 5.2.1. Просмотр информации о резервных копиях

Просмотр списка существующих копий для каждого экземпляра возможно осуществить командой:

```
pg_probackup show -B backup_dir [--help] [--instance
instance_name [-i backup_id | --archive]] [--format=plain|json]
```

В транскрипции команда имеет вид:

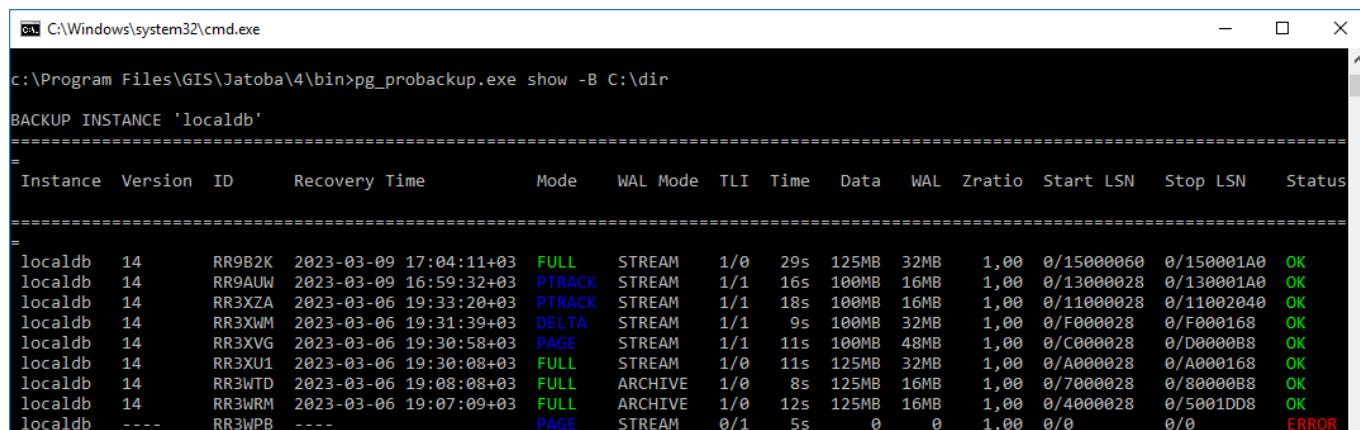
```
pg_probackup show -B каталог_копий [--help] [--instance
имя_экземпляра [-i ид_резервной_копии | --archive]] [--
format=plain|json]
```

В результате pg\_ProBackup выведет на экран список всех имеющихся резервных копий.

Например:

- В ОС Windows:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B
C:\dir
```



Instance	Version	ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Start LSN	Stop LSN	Status
localdb	14	RR9B2K	2023-03-09 17:04:11+03	FULL	STREAM	1/0	29s	125MB	32MB	1,00	0/15000060	0/150001A0	OK
localdb	14	RR9AUW	2023-03-09 16:59:32+03	PTTRACK	STREAM	1/1	16s	100MB	16MB	1,00	0/13000028	0/130001A0	OK
localdb	14	RR3XZA	2023-03-06 19:33:20+03	PTTRACK	STREAM	1/1	18s	100MB	16MB	1,00	0/11000028	0/11002040	OK
localdb	14	RR3XWM	2023-03-06 19:31:39+03	DELTA	STREAM	1/1	9s	100MB	32MB	1,00	0/F000028	0/F000168	OK
localdb	14	RR3XVG	2023-03-06 19:30:58+03	PAGE	STREAM	1/1	11s	100MB	48MB	1,00	0/C000028	0/D0000B8	OK
localdb	14	RR3XU1	2023-03-06 19:30:08+03	FULL	STREAM	1/0	11s	125MB	32MB	1,00	0/A000028	0/A000168	OK
localdb	14	RR3WTD	2023-03-06 19:08:08+03	FULL	ARCHIVE	1/0	8s	125MB	16MB	1,00	0/7000028	0/80000B8	OK
localdb	14	RR3WRM	2023-03-06 19:07:09+03	FULL	ARCHIVE	1/0	12s	125MB	16MB	1,00	0/4000028	0/5001DD8	OK
localdb	----	RR3WPB	----	PAGE	STREAM	0/1	5s	0	0	1,00	0/0	0/0	ERROR

Рисунок 5.5 – Содержание каталога резервных копий

- В GNU Linux:

```
./pg_probackup show -B /dir
```

Для каждой копии выдаются следующие сведения, указанные в таблице 5.1:

Таблица 5.1 – Перечень сведений о резервных копиях

Сведения		Описание
Instance		имя экземпляра
Version		базовая версия СУБД
ID		идентификатор резервной копии
Recovery time		время, на которое можно восстановить кластер из данной копии
Mode		режим, в котором была сделана копия. Возможные значения:
	FULL	полная
	PAGE	страничная
	DELTA	инкрементальная
	PTRACK	копирование изменений
WAL Mode		режим доставки WAL. Возможные значения:
	STREAM	поточковый
	ARCHIVE	архивный
TLI		идентификаторы линии времени текущей копии и ее родительской копии
Time		Время, в течение которого была выполнена данная копия
Data		объем файлов данных в этой копии (не включает в себя объем файлов WAL). Для копий, сделанных в режиме STREAM, общий размер можно рассчитать, сложив значения Data и WAL
WAL		размер несжатых файлов WAL, которые должны быть применены в процессе восстановления копии для достижения согласованного состояния
Zratio		коэффициент сжатия, вычисленный как отношение «uncompressed-bytes» (объем несжатых данных в байтах) к «data-bytes» (итоговый объем данных)
Start LSN		номер последней записи в журнале WAL, соответствующий началу процесса копирования. С этой позиции накатываются изменения (REDO) в процессе восстановления СУБД
Stop LSN		номер последней записи в журнале WAL, соответствующий окончанию процесса копирования. Это позиция точки согласованности при восстановлении СУБД
Status		состояние резервной копии. Возможные варианты:
	OK	резервная копия сделана и пригодна к использованию
	DONE	резервная копия сделана, но не проверена
	RUNNING	резервное копирование выполняется
	MERGING	резервная копия объединяется
	MERGED	файлы резервной копии были успешно обработаны в процессе объединения копий, но ее метаданные еще изменяются. Это состояние могут иметь только полные резервные копии
	DELETING	файлы резервной копии удаляются
	CORRUPT	некоторые файлы резервной копии повреждены

Сведения		Описание
	ERROR	резервное копирование было прервано из-за неожиданной ошибки
	ORPHAN	резервная копия непригодна к использованию, так как ее родительская копия испорчена или отсутствует

Восстановить данные можно только из копий, принимающих статус «ОК» или «DONE».

Для получения подробной информации о копии нужно указать ее идентификатор в команде show. Синтаксис команды имеет следующий вид:

```
pg_probackup show -B backup_dir --instance instance_name -i
backup_id
```

Пример команды:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B
C:\dir --instance localdb -i RR3XU1
```

Пример вывода:

```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B C:\dir --instance localdb -i RR3XU1
#Configuration
backup-mode = FULL
stream = true
compress-alg = none
compress-level = 1
from-replica = false

#Compatibility
block-size = 8192
xlog-block-size = 8192
checksum-version = 0
program-version = 2.5.6
server-version = 14

#Result backup info
timelineid = 1
start-lsn = 0/A000028
stop-lsn = 0/A000168
start-time = '2023-03-06 19:30:01+03'
end-time = '2023-03-06 19:30:12+03'
recovery-xid = 749
recovery-time = '2023-03-06 19:30:08+03'
data-bytes = 131452058
wal-bytes = 33554432
uncompressed-bytes = 131430130
pgdata-bytes = 131429890
status = OK
primary_conninfo = 'user=postgres channel_binding=prefer port=5432 sslmode=prefer sslcompression=0 sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=disable krbsrvname=postgres target_session_attrs=any'
content-crc = 4039404507
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.6 – Пример вывода информации об отдельной резервной копии

В расширенном выводе представлены дополнительные атрибуты, указанные в таблице 5.2.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



Таблица 5.2 – Перечень атрибутов резервной копии

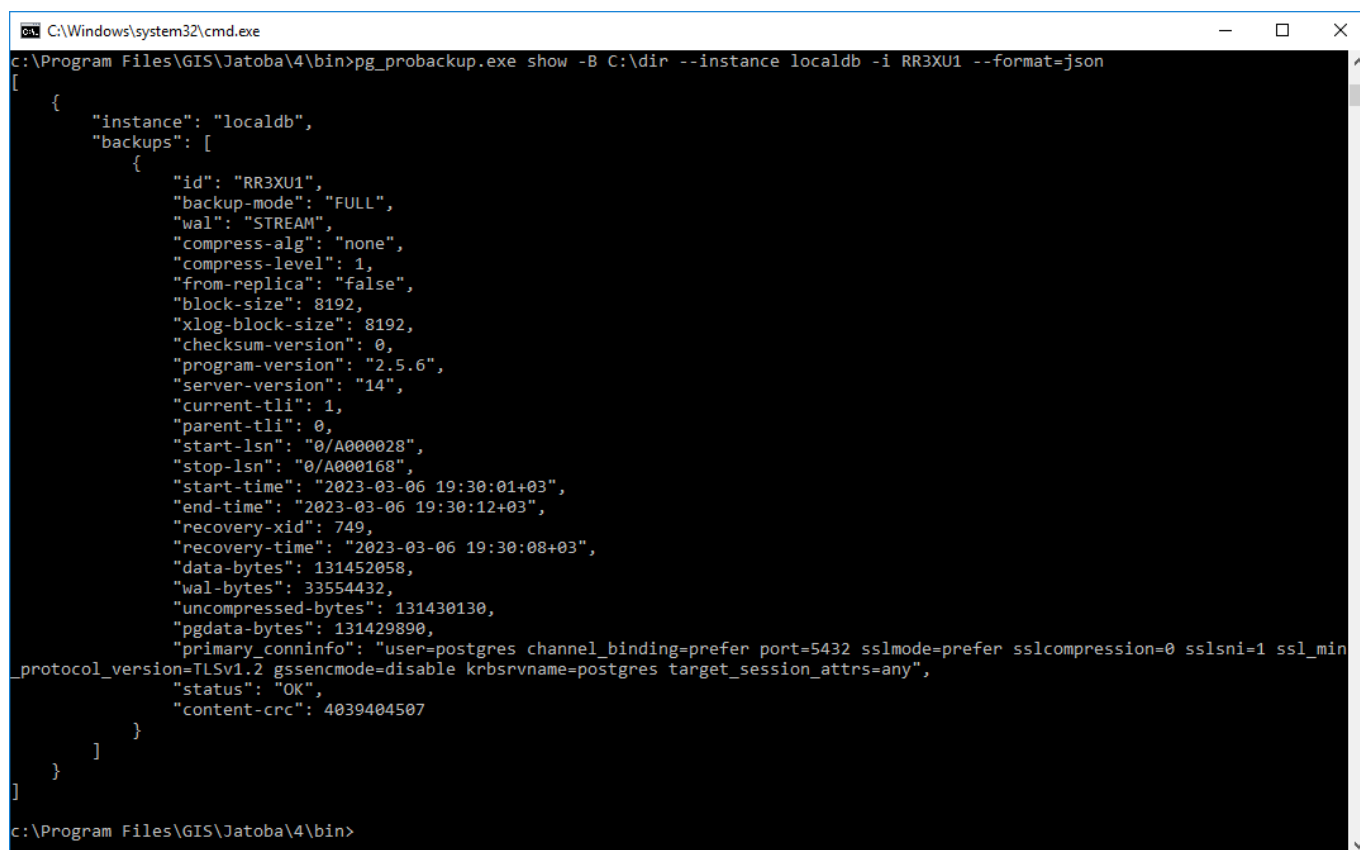
Атрибуты	Описание
compress-alg	алгоритм сжатия, используемый при получении резервной копии. Возможные значения: zlib, pglz и none (сжатие не производилось)
compress-level	уровень сжатия, применяемый в процессе резервного копирования
from-replica	признак того, что копия была сделана с ведомого сервера. Возможные значения: 1, 0
block-size	значение параметра block_size, установленное в СУБД, с которого была сделана копия
checksum-version	признак включения параметра data_checksums в исходной СУБД. Возможные значения: 1, 0
program-version	полная версия программы pg_ProBackup, в которой создана копия
start-time	время начала резервного копирования
end-time	время окончания резервного копирования
expire-time	время, когда закреплённая копия может быть ликвидирована в соответствии с политикой хранения. Этот атрибут имеется только у закреплённых копий
uncompressed-bytes	размер файлов данных до добавления заголовков страниц и сжатия. В случае использования сжатия оценить его эффективность можно, сопоставив объем uncompressed-bytes (байтов несжатых данных) с data-bytes (байтов данных)
pgdata-bytes	размер файлов данных СУБД на момент копирования. Эффективность инкрементального метода копирования можно оценить, сопоставив объем pgdata-bytes (байтов в PGDATA) с uncompressed-bytes (байтов несжатых данных)
recovery-xid	идентификатор транзакции, соответствующей моменту окончания резервного копирования
parent-backup-id	идентификатор родительской копии. Определен только для инкрементальных копий
primary_conninfo	параметры подключения libpq, с использованием которых производилось подключение к СУБД для получения этой резервной копии. Пароль в эти параметры не включается
note	текстовое примечание, связанное с копией
content-crc	контрольная сумма файла backup_content.control, рассчитанная по алгоритму CRC32. Позволяет выявить повреждение метаданных копии

Получение подробной информации о резервной копии в формате JSON:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B
C:\>dir --instance localdb -i RR3XU1 --format=json
```

Пример вывода:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B C:\dir --instance localdb -i RR3XU1 --format=json
[
  {
    "instance": "localdb",
    "backups": [
      {
        "id": "RR3XU1",
        "backup-mode": "FULL",
        "wal": "STREAM",
        "compress-alg": "none",
        "compress-level": 1,
        "from-replica": "false",
        "block-size": 8192,
        "xlog-block-size": 8192,
        "checksum-version": 0,
        "program-version": "2.5.6",
        "server-version": "14",
        "current-tli": 1,
        "parent-tli": 0,
        "start-lsn": "0/A000028",
        "stop-lsn": "0/A000168",
        "start-time": "2023-03-06 19:30:01+03",
        "end-time": "2023-03-06 19:30:12+03",
        "recovery-xid": 749,
        "recovery-time": "2023-03-06 19:30:08+03",
        "data-bytes": 131452058,
        "wal-bytes": 33554432,
        "uncompressed-bytes": 131430130,
        "pgdata-bytes": 131429890,
        "primary_conninfo": "user=postgres channel_binding=prefer port=5432 sslmode=prefer sslcompression=0 sslsnr=1 ssl_min
_protocol_version=TLsv1.2 gssencmode=disable krbsrvname=postgres target_session_attrs=any",
        "status": "OK",
        "content-crc": 4039404507
      }
    ]
  }
]
c:\Program Files\GIS\Jatoba\4\bin>

```

Рисунок 5.7 – Пример вывода информации об отдельной резервной копии в формате JSON

### 5.2.2. Просмотр оглавления архива WAL

Для получения информации об архиве WAL для каждого экземпляра нужно выполнить команду:

```
pg_probackup show -B backup_dir [--instance instance_name] --archive
```

pg\_ProBackup выводит список всех имеющихся файлов WAL, сгруппированных по линиям времени.

Например, командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B c:\DIR --instance localdb --archive
```

```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B c:\DIR --instance localdb --archive
ARCHIVE INSTANCE 'localdb'
=====
TLI Parent TLI Switchpoint Min Segno Max Segno N segments Size Zratio N backups Status
=====
1 0 0/0 00000001000000000000000000000004 00000001000000000000000000000020 12 192MB 1,00 9 DEGRADED
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.8 – Просмотр оглавления архива WAL

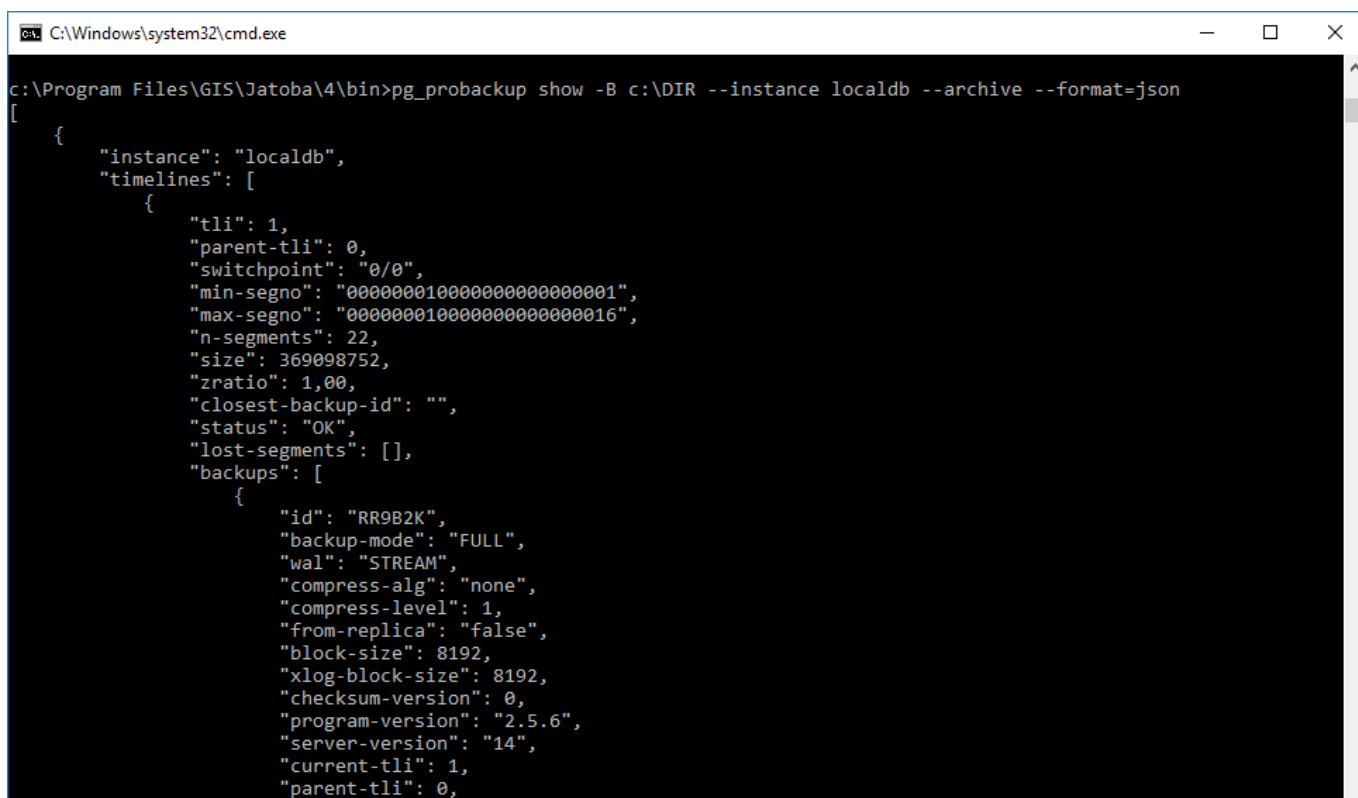
Для каждой линии времени выдаются следующие сведения, указанные в таблице 5.3:

Таблица 5.3 – Параметры архива WAL

Атрибуты		Описание
TLI		идентификатор линии времени
Parent TLI		идентификатор линии времени, от которой была ответвлена данная
Switchpoint		LSN момента, когда эта линия времени ответвилась от родительской
Min Segno		первый сегмент WAL, относящийся к этой линии времени
Max Segno		последний сегмент WAL, относящийся к этой линии времени
N segments		количество сегментов WAL, относящихся к этой линии времени
Size		объем, который занимают файлы на диске
Zratio		коэффициент сжатия, вычисляемый по формуле $N \text{ segments} * \text{wal\_segment\_size} * \text{wal\_block\_size} / \text{Size}$
N backups		число копий, относящихся к этой линии времени. Получить подробные сведения об этих копиях можно с помощью формата JSON
Status		состояние архива WAL для этой линии времени. Возможные значения:
	OK	в архиве присутствуют все сегменты WAL между Min Segno и Max Segno
	DEGRADED	некоторые сегменты WAL в интервале между Min Segno и Max Segno отсутствуют. Получить информацию об отсутствующих сегментах можно, сформировав отчет в формате JSON

Для получения подробной информации об архиве WAL в формате JSON нужно выполнить команду:

```
pg_probackup show -B backup_dir[--instance instance_name] --
archive --format=json
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B c:\DIR -
-instance localdb --archive --format=json
```



```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B c:\DIR --instance localdb --archive --format=json
[
  {
    "instance": "localdb",
    "timelines": [
      {
        "tli": 1,
        "parent-tli": 0,
        "switchpoint": "0/0",
        "min-segno": "000000010000000000000001",
        "max-segno": "000000010000000000000016",
        "n-segments": 22,
        "size": 369098752,
        "zratio": 1,00,
        "closest-backup-id": "",
        "status": "OK",
        "lost-segments": [],
        "backups": [
          {
            "id": "RR9B2K",
            "backup-mode": "FULL",
            "wal": "STREAM",
            "compress-alg": "none",
            "compress-level": 1,
            "from-replica": "false",
            "block-size": 8192,
            "xlog-block-size": 8192,
            "checksum-version": 0,
            "program-version": "2.5.6",
            "server-version": "14",
            "current-tli": 1,
            "parent-tli": 0,

```

Рисунок 5.9 – Вывод информации об архиве WAL в формате JSON

Вывод информации в формате JSON об архиве WAL аналогичен выводу информации об архиве WAL в текстовом формате. При этом необходимо учитывать некоторые особенности вывода в формате JSON:

- Размер выражается в байтах.
- Атрибут *closest-backup-id* содержит идентификатор самой последней доступной копии, принадлежащей к одной из предыдущих линий времени. Копия используется для восстановления на момент, относящийся к этой линии времени. Если такой копии не существует, данный атрибут будет пустым.
- В массиве *lost-segments* представлены интервалы отсутствующих сегментов на линиях времени в непригодном состоянии (DEGRADED). На линиях времени в целостном состоянии ОК массив *lost-segments* пуст.
- В массиве *backups* перечисляются все резервные копии, относящиеся к данной линии времени. Если к линии времени не относятся резервные копии, этот массив пуст.

### 5.3. Настройка политики хранения

Используя `pg_ProBackup` можно реализовать политики хранения резервных копий, в соответствии с которыми будут удаляться лишние резервные копии и будет выполняться очистка ненужных сегментов WAL. Кроме того, можно закреплять определенные копии, чтобы они сохранялись независимо от политики, как описано в п. 5.3.2. Эти подходы можно комбинировать произвольным образом.

#### 5.3.1. Удаление ненужных копий

По умолчанию все резервные копии, которые создает `pg_ProBackup`, сохраняются в предназначенном для них каталоге. Для экономии дискового пространства политику сохранения копий можно настроить так, чтобы ненужные копии удалялись.

Для того, чтобы настроить политику хранения, нужно задать одну или несколько следующих переменных в файле `pg_probackup.conf` с помощью команды [set-config](#):

- `--retention-redundancy=redundancy` – определяет количество полных резервных копий, которое должно сохраняться в каталоге копий;
- `--retention-window=window` – определяет самый ранний момент времени, на который `pg_probackup` может выполнить восстановление. В этом параметре задается количество дней от текущего момента. Например, если `retention-window=7`, должна сохраниться минимум одна копия за последние 7 дней, вместе с соответствующими файлами WAL и всеми последующими копиями.

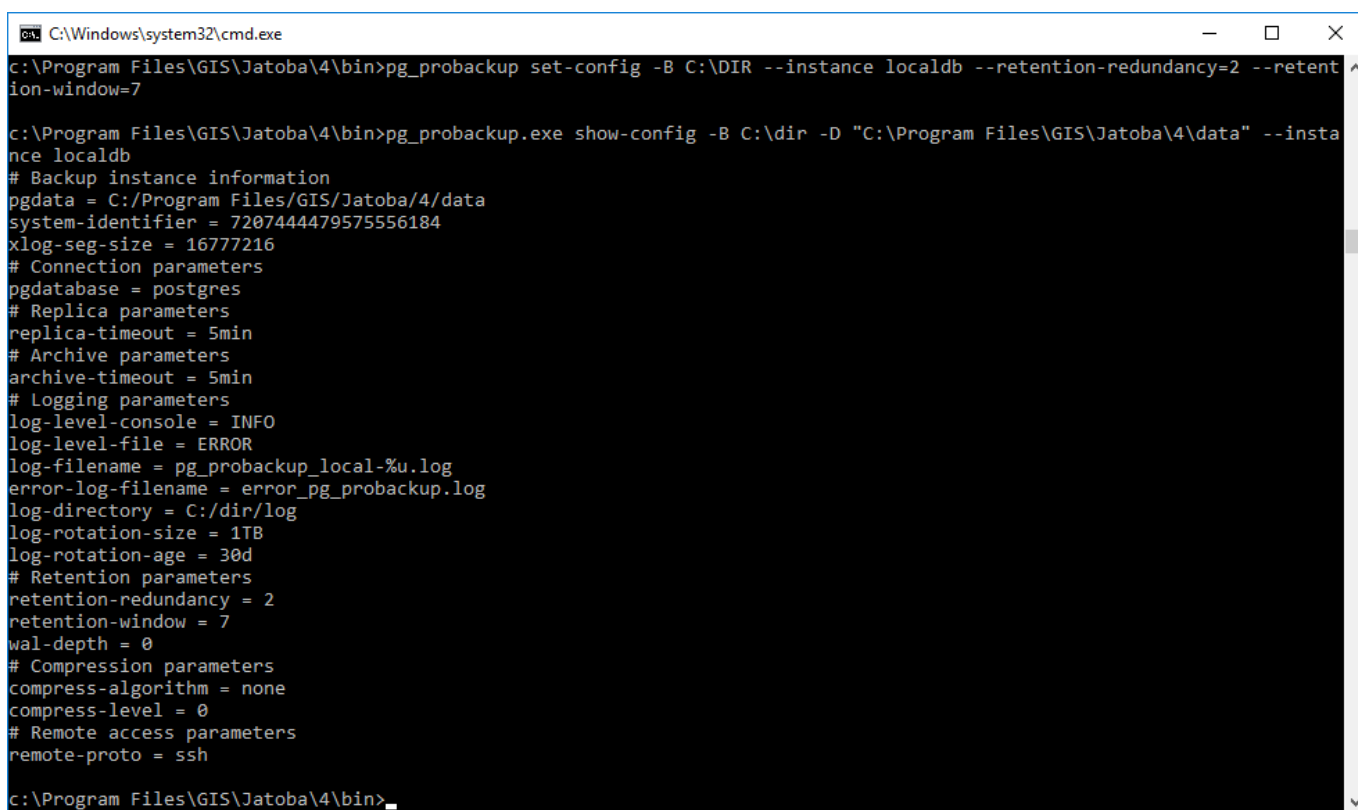
Если установлены параметры `--retention-redundancy` и `--retention-window`, при очистке каталога от ненужных копий принимаются во внимание оба заданных ими условия. Например, если задать параметры `--retention-redundancy=2` и `--retention-window=7`, программа `pg_probackup` сохранит две полные резервные копии, а также все копии, необходимые для восстановления данных, за последние 7 дней:

```
pg_probackup set-config -B backup_dir --instance instance_name  
--retention-redundancy=2 --retention-window=7
```

Пример командной строки:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup set-config -B  
C:\DIR --instance localdb --retention-redundancy=2 --retention-  
window=7
```

Для просмотра установленных параметров в конфигурационном файле `pg_probackup.conf` воспользуйтесь командой [show-config](#):



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup set-config -B C:\DIR --instance localdb --retention-redundancy=2 --retention-window=7  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb  
# Backup instance information  
pgdata = C:/Program Files/GIS/Jatoba/4/data  
system-identifier = 7207444479575556184  
xlog-seg-size = 16777216  
# Connection parameters  
pgdatabase = postgres  
# Replica parameters  
replica-timeout = 5min  
# Archive parameters  
archive-timeout = 5min  
# Logging parameters  
log-level-console = INFO  
log-level-file = ERROR  
log-filename = pg_probackup_local-%u.log  
error-log-filename = error_pg_probackup.log  
log-directory = C:/dir/log  
log-rotation-size = 1TB  
log-rotation-age = 30d  
# Retention parameters  
retention-redundancy = 2  
retention-window = 7  
wal-depth = 0  
# Compression parameters  
compress-algorithm = none  
compress-level = 0  
# Remote access parameters  
remote-proto = ssh  
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 5.10 – Установка параметров `--retention-redundancy=2 --retention-window=7`

Чтобы очистить каталог копий в соответствии с политикой хранения, нужно запустить команду `delete` с параметрами сохранения или выполнить команду `backup` с параметрами. При выполнении команды `backup` ненужные копии будут удалены сразу после создания новой копии.

Например, для удаления всех ненужных резервных копий, согласно установленной политике хранения, необходимо выполнить следующую команду с параметром `--delete-expired`:

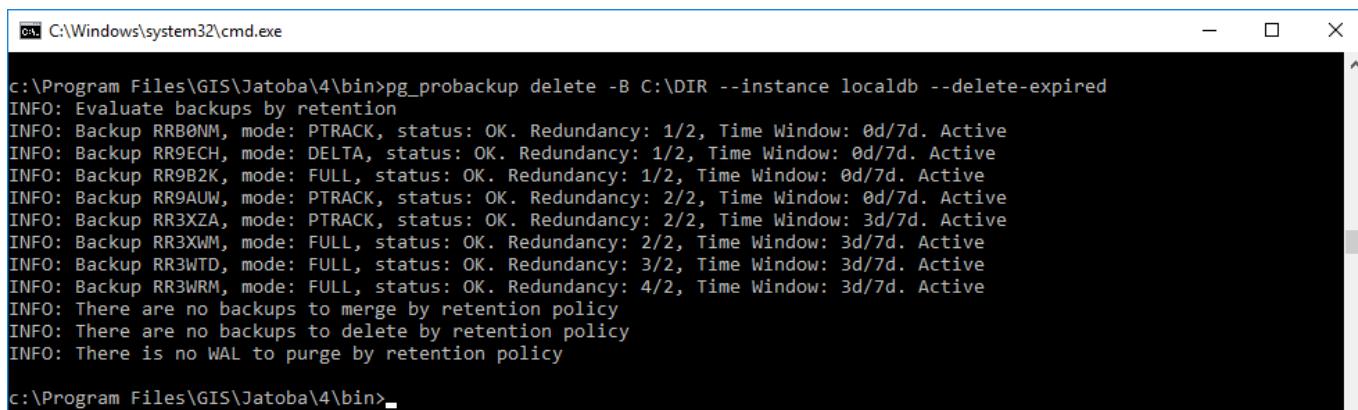
```
pg_probackup delete -B backup_dir --instance instance_name --  
delete-expired
```

Пример командной строки:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR
--instance localdb --delete-expired
```

В результате компонент pg\_ProBackup выполнит требуемое действие, при этом по резервным копиям, к которым не применима установленная политика, будут выведены информационные сообщения:

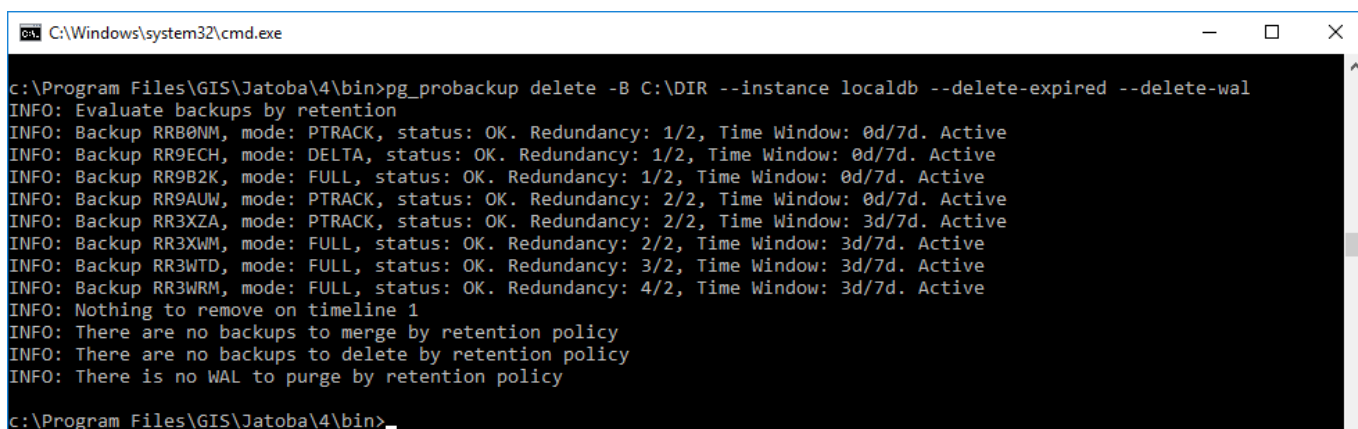


```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR --instance localdb --delete-expired
INFO: Evaluate backups by retention
INFO: Backup RRB0NM, mode: PTRACK, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9ECH, mode: DELTA, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9B2K, mode: FULL, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9AUW, mode: PTRACK, status: OK. Redundancy: 2/2, Time Window: 0d/7d. Active
INFO: Backup RR3XZA, mode: PTRACK, status: OK. Redundancy: 2/2, Time Window: 3d/7d. Active
INFO: Backup RR3XWM, mode: FULL, status: OK. Redundancy: 2/2, Time Window: 3d/7d. Active
INFO: Backup RR3WTD, mode: FULL, status: OK. Redundancy: 3/2, Time Window: 3d/7d. Active
INFO: Backup RR3WRM, mode: FULL, status: OK. Redundancy: 4/2, Time Window: 3d/7d. Active
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
INFO: There is no WAL to purge by retention policy
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 5.11 – Информационные сообщения по применению установленной политики

При удалении файлов WAL, которые больше не требуются для копий, следует указать дополнительно --delete-wal:

```
pg_probackup delete -B backup_dir --instance instance_name --
delete-expired --delete-wal
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR
--instance localdb --delete-expired --delete-wal
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR --instance localdb --delete-expired --delete-wal
INFO: Evaluate backups by retention
INFO: Backup RRB0NM, mode: PTRACK, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9ECH, mode: DELTA, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9B2K, mode: FULL, status: OK. Redundancy: 1/2, Time Window: 0d/7d. Active
INFO: Backup RR9AUW, mode: PTRACK, status: OK. Redundancy: 2/2, Time Window: 0d/7d. Active
INFO: Backup RR3XZA, mode: PTRACK, status: OK. Redundancy: 2/2, Time Window: 3d/7d. Active
INFO: Backup RR3XWM, mode: FULL, status: OK. Redundancy: 2/2, Time Window: 3d/7d. Active
INFO: Backup RR3WTD, mode: FULL, status: OK. Redundancy: 3/2, Time Window: 3d/7d. Active
INFO: Backup RR3WRM, mode: FULL, status: OK. Redundancy: 4/2, Time Window: 3d/7d. Active
INFO: Nothing to remove on timeline 1
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
INFO: There is no WAL to purge by retention policy
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 5.12 – Информационные сообщения удаления ненужных резервных копий WAL

При выполнении команд delete или backup возможно установить или переопределить текущую политику хранения, добавив параметры --retention-redundancy и --retention-window:

```
pg_probackup delete -B backup_dir --instance instance_name --
delete-expired --retention-window=7 --retention-redundancy=2
```

Так как для инкрементальных копий требуется наличие всех родительских полных копий и всех предыдущих инкрементальных копий, даже по истечении срока их хранения эти копии нельзя удалить, пока минимум одна инкрементальная копия в цепочке удовлетворяет политике хранения.

Устаревшие копии, необходимые для восстановления инкрементальной копии, можно объединить с нужной инкрементальной копией, воспользовавшись ключом `--merge-expired` команды `backup` или `delete`.

### 5.3.2. Закрепление резервных копий

Закрепление резервных копий на определенное время позволяет хранить отдельные копии дольше, чем допускает установленная политика хранения. Например:

```
pg_probackup set-backup -B backup_dir --instance instance_name
-i backup_id --ttl=30d
```

Просмотрите каталог резервных копий при помощи команды `show`:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR -
--instance localdb
```

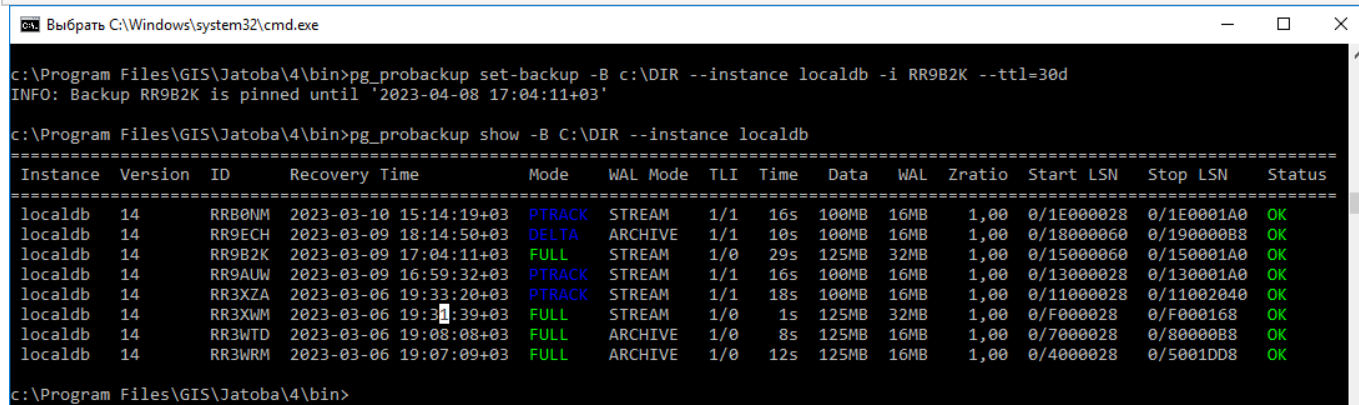


Рисунок 5.13 – Просмотр каталога резервных копий

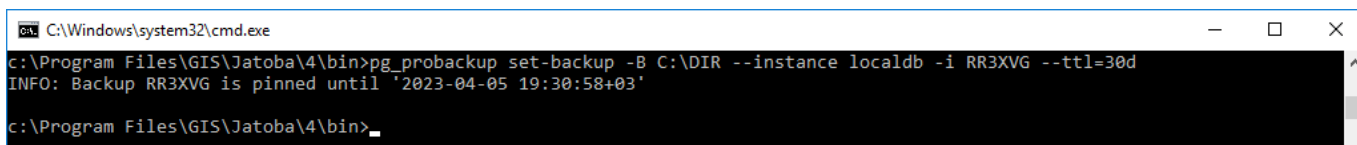
Выберите корректную резервную копию RR3XVG со статусом «OK».

Закрепите выбранную резервную копию RR3XVG на 30 дней при помощи команды `set-backup`:



```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup set-backup -B  
C:\DIR --instance localdb -i RR3XVG --ttl=30d
```

В результате компонент pg\_ProBackup выведет информационное сообщение о закреплении выбранной резервной копии.



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup set-backup -B C:\DIR --instance localdb -i RR3XVG --ttl=30d  
INFO: Backup RR3XVG is pinned until '2023-04-05 19:30:58+03'  
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 5.14 – Вывод информационного сообщения о закреплении резервной копии на 30 дней

Можно явно задать время истечения срока хранения копии, воспользовавшись ключом --expire-time. Например:

```
pg_probackup set-backup -B backup_dir --instance instance_name  
-i backup_id --expire-time="2023-04-01 00:00:00+03"
```

Также можно воспользоваться ключами --ttl и --expire-time команды backup и закрепить копию при ее создании:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL --ttl=30d  
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL --expire-time="2023-04-01 00:00:00+03"
```

Проверить закрепление копии можно с помощью команды show:

```
pg_probackup show -B backup_dir --instance instance_name -i  
backup_id
```

Пример командной строки:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR -  
-instance localdb -i RR3XVG
```

Закрепленная копия имеет атрибуты:

- recovery-time – время создания копии;
- expire-time – время окончания срока хранения копии.

На представленном рисунке 5.15 цветом маркированы указанные атрибуты.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR --instance localdb -i RR3XVG
#Configuration
backup-mode = PAGE
stream = true
compress-alg = none
compress-level = 1
from-replica = false

#Compatibility
block-size = 8192
xlog-block-size = 8192
checksum-version = 0
program-version = 2.5.6
server-version = 14

#Result backup info
timelineid = 1
start-lsn = 0/C000028
stop-lsn = 0/D000088
start-time = '2023-03-06 19:30:52+03'
end-time = '2023-03-06 19:31:03+03'
recovery-xid = 749
recovery-time = '2023-03-06 19:30:58+03'
expire-time = '2023-04-05 19:30:58+03'
data-bytes = 104970524
wal-bytes = 50331648
uncompressed-bytes = 104970484
pgdata-bytes = 131429890
status = OK
parent-backup-id = 'RR3XU1'
primary_conninfo = 'user=postgres channel_binding=prefer port=5432 sslmode=prefer sslcompression=0 sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=disable krbsrvname=postgres target_session_attrs=any'
content-crc = 3283204989
```

Рисунок 5.15 – Атрибуты recovery-time и expire-time

Отменить закрепление копии можно, передав в параметре --ttl ноль:

```
pg_probackup set-backup -B backup_dir --instance instance_name
-i backup_id --ttl=0
```



Для закрепленной инкрементальной копии неявным образом закрепляются все нужные ей родительские копии. При снятии закрепления с копии, нужные ей родительские копии также автоматически перестанут быть закрепленными.

### 5.3.3. Настройка политики хранения архива WAL

При непрерывном архивировании WAL, сегменты WAL могут занимать много места на диске. Даже если с требуемой периодичностью удалять старые резервные копии, с параметром --delete-wal, будут удалены только те сегменты WAL, которые не относятся ни к какой копии из остающихся в каталоге.

Если возможность восстановления на момент времени нужна только для последних копий, можно настроить политику хранения архива WAL, чтобы ограничить глубину архива и сэкономить место на диске.

Для настройки политики хранения архива WAL, нужно запустить команду `set-config` с параметром `--wal-depth`, задающим количество копий, которые могут быть использованы для PITR. Этот параметр действует для всех линий времени, поэтому можно выполнять PITR для одинакового количества копий на каждой линии времени, при их наличии. Закрепленные копии в этом числе не учитываются, если закрепляется одна из последних копий, `pg_ProBackup` обеспечивает возможность PITR для каждой дополнительной копии.

Для удаления сегментов WAL, неудовлетворяющих заданной политике хранения архива WAL, нужно выполнить команду `delete` или `backup` с параметром `--delete-wal`.

Для архивных копий сегменты WAL между Start LSN и Stop LSN всегда сохраняются, поэтому эти копии остаются рабочими вне зависимости от значения `--wal-depth` и при необходимости могут быть восстановлены.

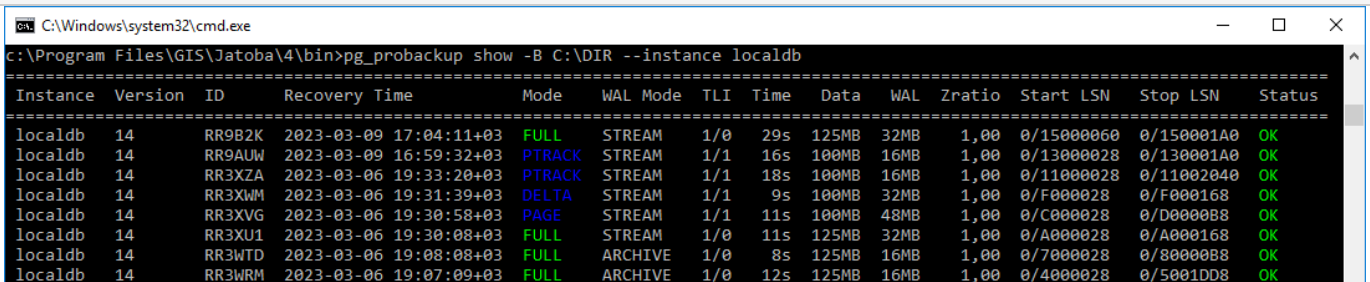
Для переопределения ранее заданной политики сохранения WAL и удаления старых сегментов WAL «на лету», можно использовать параметр `--wal-depth` с командами `delete` или `backup`.

После архивирования экземпляра `localdb` в каталоге копий и настройки непрерывного архивирования WAL:

```
pg_probackup show -B backup_dir --instance instance_name
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR --instance localdb
```



Instance	Version	ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Start LSN	Stop LSN	Status
localdb	14	RR9B2K	2023-03-09 17:04:11+03	FULL	STREAM	1/0	29s	125MB	32MB	1,00	0/15000060	0/150001A0	OK
localdb	14	RR9AUW	2023-03-09 16:59:32+03	PTTRACK	STREAM	1/1	16s	100MB	16MB	1,00	0/13000028	0/130001A0	OK
localdb	14	RR3XZA	2023-03-06 19:33:20+03	PTTRACK	STREAM	1/1	18s	100MB	16MB	1,00	0/11000028	0/11002040	OK
localdb	14	RR3XWM	2023-03-06 19:31:39+03	DELTA	STREAM	1/1	9s	100MB	32MB	1,00	0/F000028	0/F000168	OK
localdb	14	RR3XVG	2023-03-06 19:30:58+03	PAGE	STREAM	1/1	11s	100MB	48MB	1,00	0/C000028	0/D0000B8	OK
localdb	14	RR3XU1	2023-03-06 19:30:08+03	FULL	STREAM	1/0	11s	125MB	32MB	1,00	0/A000028	0/A000168	OK
localdb	14	RR3WTD	2023-03-06 19:08:08+03	FULL	ARCHIVE	1/0	8s	125MB	16MB	1,00	0/7000028	0/80000B8	OK
localdb	14	RR3WRM	2023-03-06 19:07:09+03	FULL	ARCHIVE	1/0	12s	125MB	16MB	1,00	0/4000028	0/5001DD8	OK

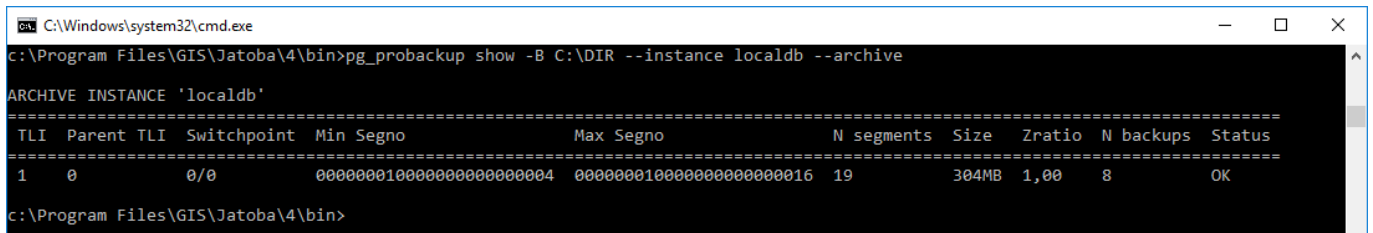
Рисунок 5.16 – Просмотр каталога резервных копий

Для проверки состояния архива WAL воспользуйтесь командой `show` с ключом `--archive`:

```
pg_probackup show -B backup_dir --instance instance_name --archive
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR --instance localdb --archive
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup show -B C:\DIR --instance localdb --archive
ARCHIVE INSTANCE 'localdb'
=====
 TLI  Parent TLI  Switchpoint  Min Segno  Max Segno  N segments  Size  Zratio  N backups  Status
=====
  1      0          0/0          000000010000000000000004  000000010000000000000016  19         304MB  1,00      8         OK
=====
c:\Program Files\GIS\Jatoba\4\bin>
```

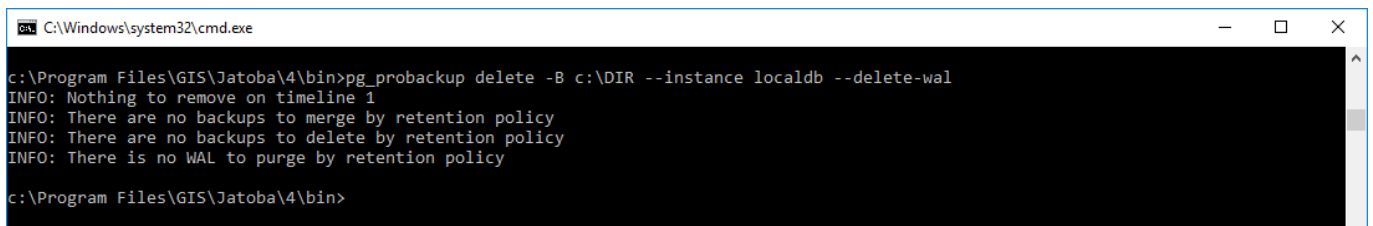
Рисунок 5.17 – Просмотр состояния архива WAL

Операция очистки WAL без ключа --wal-depth может удалить лишь один сегмент:

```
pg_probackup delete -B backup_dir --instance instance_name --delete-wal
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR --instance localdb --delete-wal
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B c:\DIR --instance localdb --delete-wal
INFO: Nothing to remove on timeline 1
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
INFO: There is no WAL to purge by retention policy
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.18 – Очистка WAL

Если необходимо оставить только те сегменты, которые могут применяться к самой последней копии, установить в параметре --wal-depth значение 1:

```
pg_probackup delete -B backup_dir --instance instance_name --delete-wal --wal-depth=1
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR --instance localdb --delete-wal --wal-depth=1
```

```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B C:\DIR --instance localdb --delete-wal --wal-depth=1
INFO: On timeline 1 WAL segments between 000000010000000000000004 and 000000010000000000000018 will be removed
INFO: Logical WAL size to remove on timeline 1 : 320MB
INFO: Resident WAL size to free on timeline 1 : 64MB
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
INFO: There is no WAL to purge by retention policy
c:\Program Files\GIS\Jatoba\4\bin>

```

Рисунок 5.19 – Сохранение WAL, относящиеся к последней резервной копии и удаление остальных

Для восстановления WAL последней резервной копии возможно использовать параметр `--wal-depth` с командой `backup`:

```
pg_probackup backup -B backup_dir --instance instance_name -b DELTA --wal-depth=1 --delete-wal
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -B C:\DIR --instance localdb -b DELTA --wal-depth=1 --delete-wal
```

```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -B C:\DIR --instance localdb -b DELTA --wal-depth=1 --delete-wal
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR9ECH, backup mode: DELTA, wal mode: ARCHIVE, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: wait for pg_start_backup()
INFO: Parent backup: RR9B2K
INFO: Wait for WAL segment C:/DIR/wal/localdb/000000010000000000000018 to be archived
INFO: PGDATA size: 125MB
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 1s
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup RR9ECH
INFO: Backup RR9ECH data files are valid
INFO: Backup RR9ECH resident size: 100MB
INFO: Backup RR9ECH completed
INFO: On timeline 1 WAL segments between 000000010000000000000004 and 000000010000000000000018 will be removed
INFO: Logical WAL size to remove on timeline 1 : 320MB
INFO: Resident WAL size to free on timeline 1 : 320MB
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
c:\Program Files\GIS\Jatoba\4\bin>

```

Рисунок 5.20 – Восстановление WAL последней резервной копии

### 5.3.4. Объединение резервных копий

По мере создания новых инкрементальных копий, общий размер каталога резервных копий может существенно увеличиться. Для экономии места на диске можно объединить инкрементальные копии с родительской полной копией, выполнив команду `merge` и передав ей идентификатор самой последней резервной копии, подлежащей объединению:

```
pg_probackup merge -B backup_dir --instance instance_name -i  
backup_id
```

Команда merge объединяет копии, относящиеся к одной цепочке инкрементальных копий:

- если выбирается полная копия, она будет объединена с первой инкрементальной копией после нее;
- если выбрана инкрементальная копия, она будет объединена с родительской полной копией, включая все инкрементальные копии между ними.

После завершения объединения, результирующая полная копия будет вмещать в себя все данные, а инкрементальные копии будут удалены как избыточные. Таким образом, операция объединения равнозначна созданию новой полной копии с удалением всех устаревших копий, но выполняется быстрее, особенно с большими объемами данных, и не нагружает подсистему ввода/вывода и сеть.

Перед объединением pg\_ProBackup проверяет все задействованные резервные копии, чтобы удостовериться в их целостности. Проверить текущее состояние резервной копии, передав ее идентификатор команде show.

```
pg_probackup show -B backup_dir --instance instance_name -i  
backup_id
```

Если процесс объединения еще не закончен, резервная копия будет в состоянии «MERGING». Для полных копий также можно увидеть состояние «MERGED» в процессе изменения метаданных на последнем этапе объединения. В случае прерывания операции объединения она может быть перезапущена.

Пример:

Просмотр содержимого каталога резервных копий, осуществляется выполнением команды:

```
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B  
C:\dir --instance localdb
```

```
C:\Windows\system32\cmd.exe

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup delete -B c:\dir --instance localdb --delete-wal
INFO: Nothing to remove on timeline 1
INFO: There are no backups to merge by retention policy
INFO: There are no backups to delete by retention policy
INFO: There is no WAL to purge by retention policy

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B C:\dir --instance localdb
=====
Instance  Version  ID          Recovery Time          Mode  WAL Mode  TLI  Time  Data  WAL  Zratio  Start LSN  Stop LSN  Status
=====
localdb   14       RR9B2K      2023-03-09 17:04:11+03  FULL  STREAM    1/0  29s   125MB 32MB  1,00  0/15000060 0/150001A0 OK
localdb   14       RR9AUW      2023-03-09 16:59:32+03  PTRACK STREAM    1/1  16s   100MB 16MB  1,00  0/13000028 0/130001A0 OK
localdb   14       RR3XZA      2023-03-06 19:33:20+03  PTRACK STREAM    1/1  18s   100MB 16MB  1,00  0/11000028 0/11002040 OK
localdb   14       RR3XWM      2023-03-06 19:31:39+03  DELTA  STREAM    1/1   9s   100MB 32MB  1,00  0/F000028 0/F000168 OK
localdb   14       RR3XVG      2023-03-06 19:30:58+03  PAGE   STREAM    1/1  11s   100MB 48MB  1,00  0/C000028 0/D0000B8 OK
localdb   14       RR3XU1      2023-03-06 19:30:08+03  FULL   STREAM    1/0  11s   125MB 32MB  1,00  0/A000028 0/A000168 OK
localdb   14       RR3WTD      2023-03-06 19:08:08+03  FULL   ARCHIVE    1/0   8s   125MB 16MB  1,00  0/7000028 0/80000B8 OK
localdb   14       RR3WRM      2023-03-06 19:07:09+03  FULL   ARCHIVE    1/0  12s   125MB 16MB  1,00  0/4000028 0/5001DD8 OK
=====
```

Рисунок 5.21 – Просмотр каталога резервных копий

Объединение (merge) резервной копии с ID RR3XWM с выполняется с помощью команды:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe merge -B
C:\dir --instance localdb -i RR3XWM
```

```
C:\Windows\system32\cmd.exe

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe merge -B C:\dir --instance localdb -i RR3XWM
INFO: Merge started
INFO: Merging backup RR3XWM with parent chain
INFO: Validate parent chain for backup RR3XWM
INFO: Validating backup RR3XU1
INFO: Backup RR3XU1 data files are valid
INFO: Validating backup RR3XVG
INFO: Backup RR3XVG data files are valid
INFO: Validating backup RR3XWM
INFO: Backup RR3XWM data files are valid
INFO: Start merging backup files
INFO: Backup files are successfully merged, time elapsed: 1s
INFO: Delete: RR3XVG 2023-03-06 19:30:58+03
INFO: Delete: RR3XWM 2023-03-06 19:31:39+03
INFO: Rename merged full backup RR3XU1 to RR3XWM
INFO: Validating backup RR3XWM
INFO: Backup RR3XWM data files are valid
INFO: Merge of backup RR3XWM completed

c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.22 – Слияние резервных копий

В результате объединения количество резервных копий каталога уменьшиться.

```
C:\Windows\system32\cmd.exe

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show -B C:\dir --instance localdb
=====
Instance  Version  ID          Recovery Time          Mode  WAL Mode  TLI  Time  Data  WAL  Zratio  Start LSN  Stop LSN  Status
=====
localdb   14       RR9EEW      2023-03-09 18:16:15+03  DELTA  ARCHIVE    1/1   9s   100MB 16MB  1,00  0/18000060 0/1C0000B8 OK
localdb   14       RR9ECH      2023-03-09 18:14:50+03  DELTA  ARCHIVE    1/1  10s   100MB 16MB  1,00  0/18000060 0/190000B8 OK
localdb   14       RR9B2K      2023-03-09 17:04:11+03  FULL   STREAM    1/0  29s   125MB 32MB  1,00  0/15000060 0/150001A0 OK
localdb   14       RR9AUW      2023-03-09 16:59:32+03  PTRACK STREAM    1/1  16s   100MB 16MB  1,00  0/13000028 0/130001A0 OK
localdb   14       RR3XZA      2023-03-06 19:33:20+03  PTRACK STREAM    1/1  18s   100MB 16MB  1,00  0/11000028 0/11002040 OK
localdb   14       RR3XWM      2023-03-06 19:31:39+03  FULL   STREAM    1/0   1s   125MB 32MB  1,00  0/F000028 0/F000168 OK
localdb   14       RR3WTD      2023-03-06 19:08:08+03  FULL   ARCHIVE    1/0   8s   125MB 16MB  1,00  0/7000028 0/80000B8 OK
localdb   14       RR3WRM      2023-03-06 19:07:09+03  FULL   ARCHIVE    1/0  12s   125MB 16MB  1,00  0/4000028 0/5001DD8 OK
localdb   ----    RR3WPB      ----      ----      PAGE   STREAM    0/1   5s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WP5      ----      ----      PAGE   STREAM    0/1   5s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WOD      ----      ----      FULL   STREAM    0/0   4s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WO1      ----      ----      FULL   STREAM    0/0  10s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WN9      ----      ----      FULL   ARCHIVE    0/0   4s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WN2      ----      ----      FULL   ARCHIVE    0/0   5s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WMV      ----      ----      FULL   ARCHIVE    0/0   4s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WKE      ----      ----      FULL   ARCHIVE    0/0   7s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WJB      ----      ----      FULL   ARCHIVE    0/0   4s     0     0  1,00  0/0     0/0     ERROR
localdb   ----    RR3WIQ      ----      ----      FULL   ARCHIVE    0/0   6s     0     0  1,00  0/0     0/0     ERROR
=====

c:\Program Files\GIS\Jatoba\4\bin>
```

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



Рисунок 5.23 – Результат слияния резервных копий

Дополнительные параметры команды описаны в подразделе 10.1.13.

### 5.3.5. Удаление резервных копий

Удаление резервных копий доступно с помощью команды `delete`, имеющей следующий синтаксис:

```
pg_probackup delete -B backup_dir --instance instance_name [--help] [-j num_threads] [--progress] [--retention-redundancy=redundancy] [--retention-window=window] [--wal-depth=wal_depth] [--delete-wal] {-i backup_id | --delete-expired [--merge-expired] | --merge-expired | --status=backup_status} [--dry-run] [--no-validate] [--no-sync] [logging_options]
```

Для удаления ненужной резервной копии выполните команду:

```
pg_probackup delete -B backup_dir --instance instance_name -i backup_id
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B C:\dir --instance localdb -i RR9EEW
```

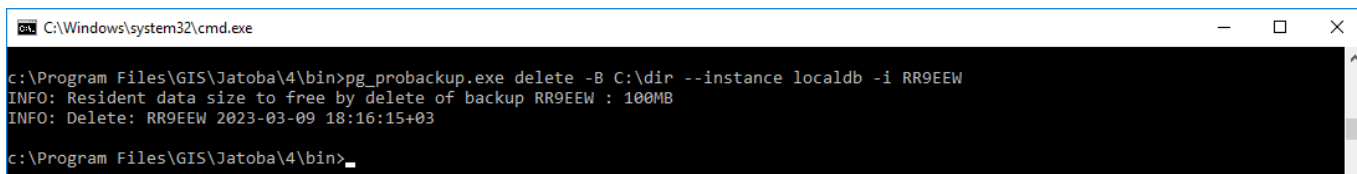


Рисунок 5.24 – Вид командной строки удаления резервной копии

Эта команда удалит резервную копию с заданным ID, вместе со всеми инкрементальными копиями, которые от нее зависят (если таковые найдутся). Можно удалить некоторые последние инкрементальные копии, сохранив предыдущую полную копию и некоторые следующие за ней инкрементальные копии.

Для удаления старых файлов WAL, не нужных для восстановления оставшихся резервных копий, необходимо воспользоваться ключом `--delete-wal`:



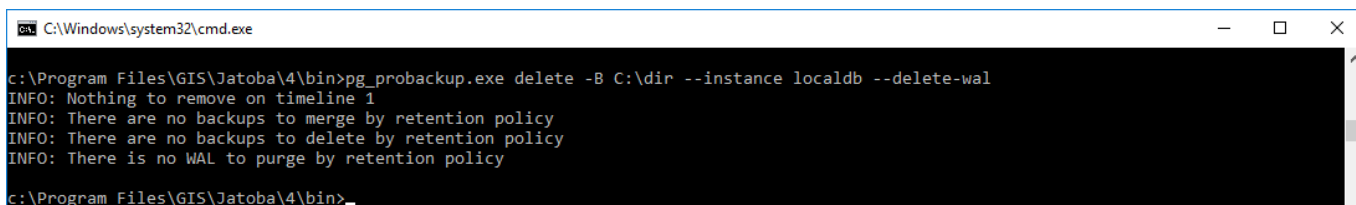
```
pg_probackup delete -B backup_dir--instance instance_name --  
delete-wal
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B  
C:\dir --instance localdb --delete-wal
```



В случае, если политики не определены, компонент pg\_ProBackup выдаст информационное сообщение, показанное на рисунке 5.25.



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B C:\dir --instance localdb --delete-wal  
INFO: Nothing to remove on timeline 1  
INFO: There are no backups to merge by retention policy  
INFO: There are no backups to delete by retention policy  
INFO: There is no WAL to purge by retention policy  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.25 – Вид командной строки и информационного сообщения об отсутствии настроенных политик хранения

Для удаления резервных копий, считающихся ненужными согласно текущей политике хранения, необходимо воспользоваться ключом --delete-expired:

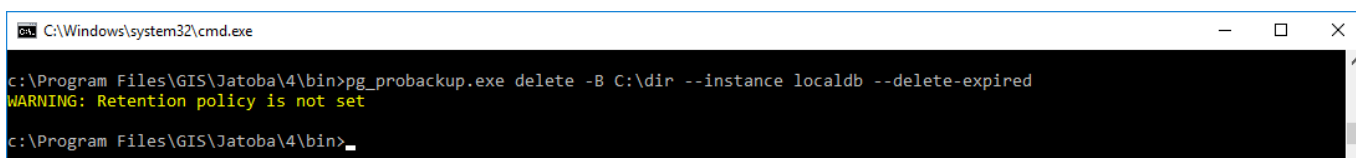
```
pg_probackup delete -B backup_dir--instance instance_name --  
delete-expired
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B  
C:\dir --instance localdb --delete-expired
```



В случае, если политики не определены компонент pg\_ProBackup выдаст информационное сообщение, показанное на рисунке 5.26.



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B C:\dir --instance localdb --delete-expired  
WARNING: Retention policy is not set  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.26 – Вид командной строки при удалении ненужных резервных копий и информационного сообщения об отсутствии настроенных политик хранения



Копии с истекшим сроком нельзя удалить, пока на них базируется минимум одна инкрементальная копия, удовлетворяющая политике хранения.

Для сокращения числа резервных копий, требующихся для восстановления инкрементальных копий, необходимо указать параметр `--merge-expired` при выполнении этой команды:

```
pg_probackup delete -B backup_dir--instance instance_name--  
delete-expired --merge-expired
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B  
C:\dir --instance localdb --delete-expired --merge-expired
```



В случае, когда политики не определены компонент `pg_ProBackup` выдаст информационное сообщение, показанное на рисунке 5.27.

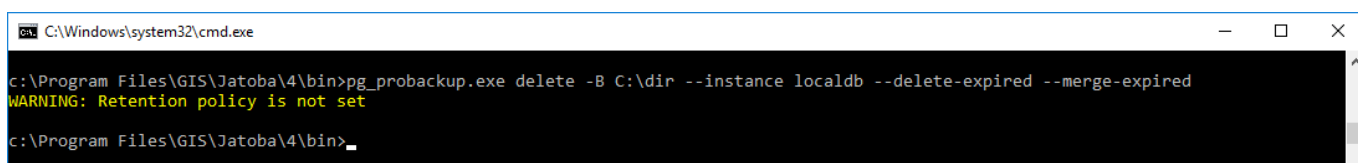


Рисунок 5.27 – Вид командной строки при удалении резервных копий с истекшим сроком и информационного сообщения об отсутствии настроенных политик хранения

В этом случае `pg_ProBackup` ищет самую раннюю инкрементальную копию, удовлетворяющую политике хранения и объединяет ее с базовыми для нее полной и инкрементальными копиями, срок хранения которых истек, тем самым превращая ее в полную копию. После завершения объединения копии, ставшие ненужными, удаляются.

Перед объединением или удалением резервных копий, возможно выполнить команду `delete` с параметром `--dry-run` для получения состояния всех имеющихся копий в соответствии с текущей политикой хранения. Никакие необратимые действия при этом выполняться не будут.

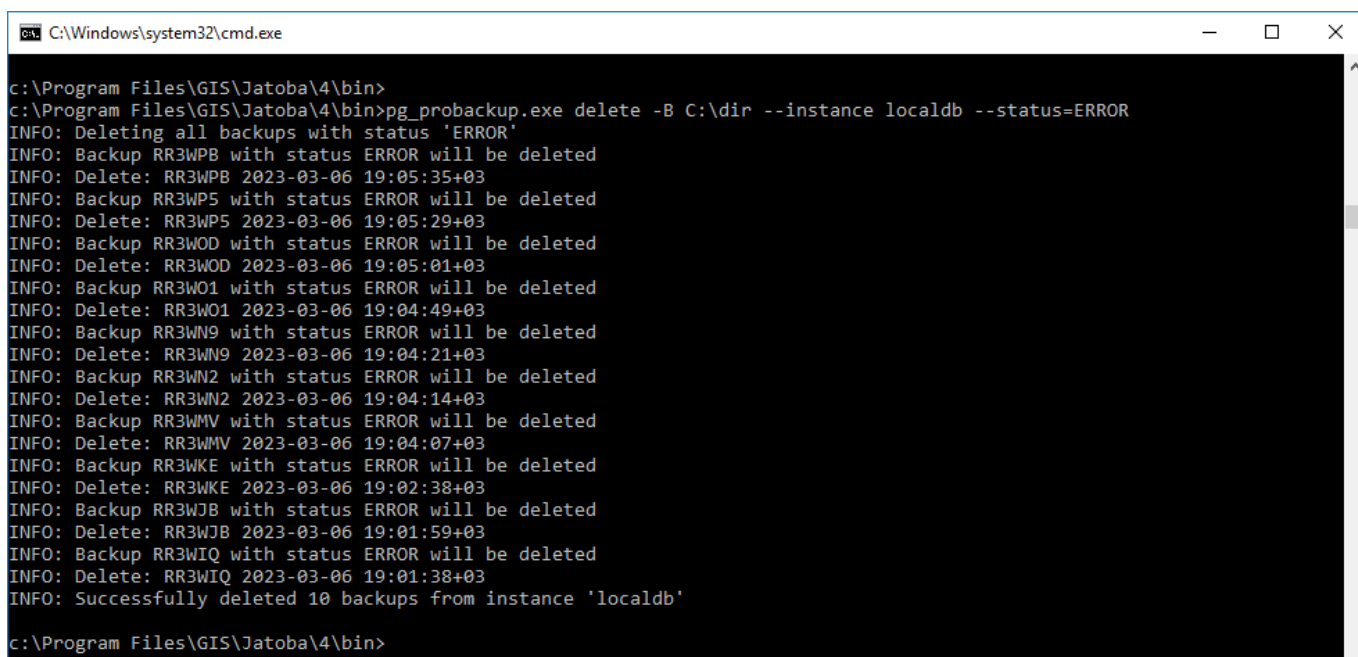
Для удаления всех копий с определенным состоянием, воспользуйтесь параметром `--status`:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
pg_probackup delete -B backup_dir --instance instance_name --  
status=ERROR
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B  
C:\dir --instance localdb --status=ERROR
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe delete -B C:\dir --instance localdb --status=ERROR  
INFO: Deleting all backups with status 'ERROR'  
INFO: Backup RR3WPB with status ERROR will be deleted  
INFO: Delete: RR3WPB 2023-03-06 19:05:35+03  
INFO: Backup RR3WP5 with status ERROR will be deleted  
INFO: Delete: RR3WP5 2023-03-06 19:05:29+03  
INFO: Backup RR3WOD with status ERROR will be deleted  
INFO: Delete: RR3WOD 2023-03-06 19:05:01+03  
INFO: Backup RR3W01 with status ERROR will be deleted  
INFO: Delete: RR3W01 2023-03-06 19:04:49+03  
INFO: Backup RR3WN9 with status ERROR will be deleted  
INFO: Delete: RR3WN9 2023-03-06 19:04:21+03  
INFO: Backup RR3WN2 with status ERROR will be deleted  
INFO: Delete: RR3WN2 2023-03-06 19:04:14+03  
INFO: Backup RR3WMV with status ERROR will be deleted  
INFO: Delete: RR3WMV 2023-03-06 19:04:07+03  
INFO: Backup RR3WKE with status ERROR will be deleted  
INFO: Delete: RR3WKE 2023-03-06 19:02:38+03  
INFO: Backup RR3WJB with status ERROR will be deleted  
INFO: Delete: RR3WJB 2023-03-06 19:01:59+03  
INFO: Backup RR3WIQ with status ERROR will be deleted  
INFO: Delete: RR3WIQ 2023-03-06 19:01:38+03  
INFO: Successfully deleted 10 backups from instance 'localdb'  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 5.28 – Выполнение удаления резервных копий с статусом ERROR

При удалении копий по критерию состояния, установленные политики сохранения не учитываются.

## 6. НАСТРОЙКА КОПИРОВАНИЯ PTRACK

Настройка режима копирования PTRACK должна выполняться от имени учетной записи, имеющей доступ ко всем базам данных, для которых планируется резервное копирование.



Установка пакетов компонентов резервного копирования описана в п. 4.2 и приведен пример установки из локального репозитория в Приложении 1 настоящего документа.

Для использования режима копирования PTRACK, необходимо выполнить следующие действия:

Выполнить создание расширения PTRACK:

- в конфигурационном файле `postgresql.conf`, в разделе «Shared Library Preloading» добавить строку:

```
shared_preload_libraries='ptrack'
```

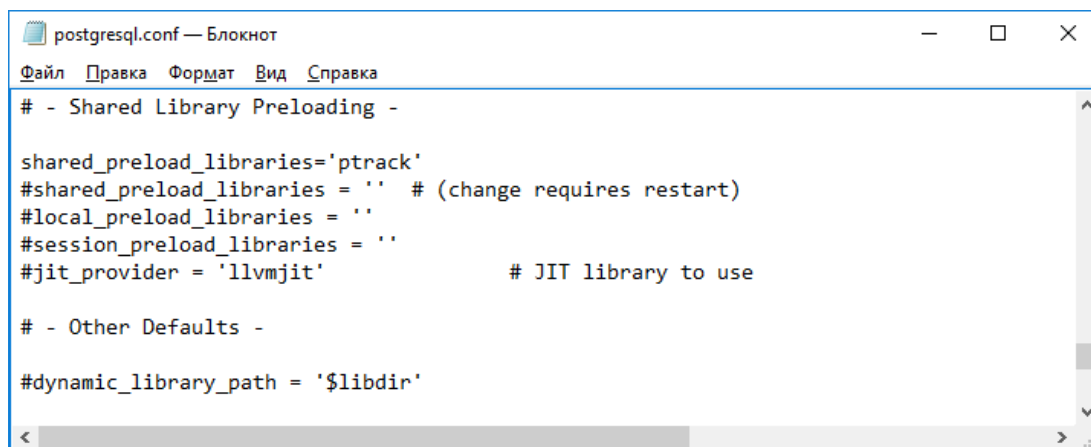


Рисунок 6.1 – Конфигурационный файл `postgresql.conf`

- сохранить файл;
- перезагрузить СУБД;
- в СУБД выполнить SQL-команду:

```
CREATE EXTENSION ptrack;
```

```

Администратор: Командная строка - psql -h localhost -d postgres -U postgres
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U postgres
Пароль пользователя postgres:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=# CREATE EXTENSION ptrack;
CREATE EXTENSION
postgres=#
  
```

Рисунок 6.2 – Создание расширения ptrack

Для включения отслеживания изменений страниц, задать для параметра *ptrack.map\_size* положительное целое значение и перезапустить сервер.

```
ptrack.map_size = 100
```

```

postgresql.conf — Блокнот
Файл  Правка  Формат  Вид  Справка
#-----
# Add settings for extensions here
ptrack.map_size = 100
  
```

Рисунок 6.3 – Установка параметра *ptrack.map\_size*

Для оптимальной производительности рекомендуется задавать *ptrack.map\_size* равным  $N / 1024$ :

где  $N$  – объем данных СУБД в мегабайтах.

Если *ptrack.map\_size* будет иметь меньшее значение, это увеличит вероятность наложения информации разных блоков в карте PTRACK, что повлечет ложные положительные результаты при определении измененных блоков и, как следствие, увеличение размера инкрементальной копии, так как в копию будут попадать и фактически неизменные блоки.



Использовать значения *ptrack.map\_size*, превышающие 1024, не рекомендуется, несмотря на наличие поддержки PTRACK таких карт.



В случае изменения значения `ptrack.map_size`, ранее созданный файл карты PTRACK очищается, и отслеживание новых блоков начинается с начала.

По этому перед созданием новых инкрементальных копий в режиме PTRACK после изменения `ptrack.map_size` необходимо сделать новую полную копию кластера данных СУБД.

Результаты выполняемых действий отражены на схеме 6.4.

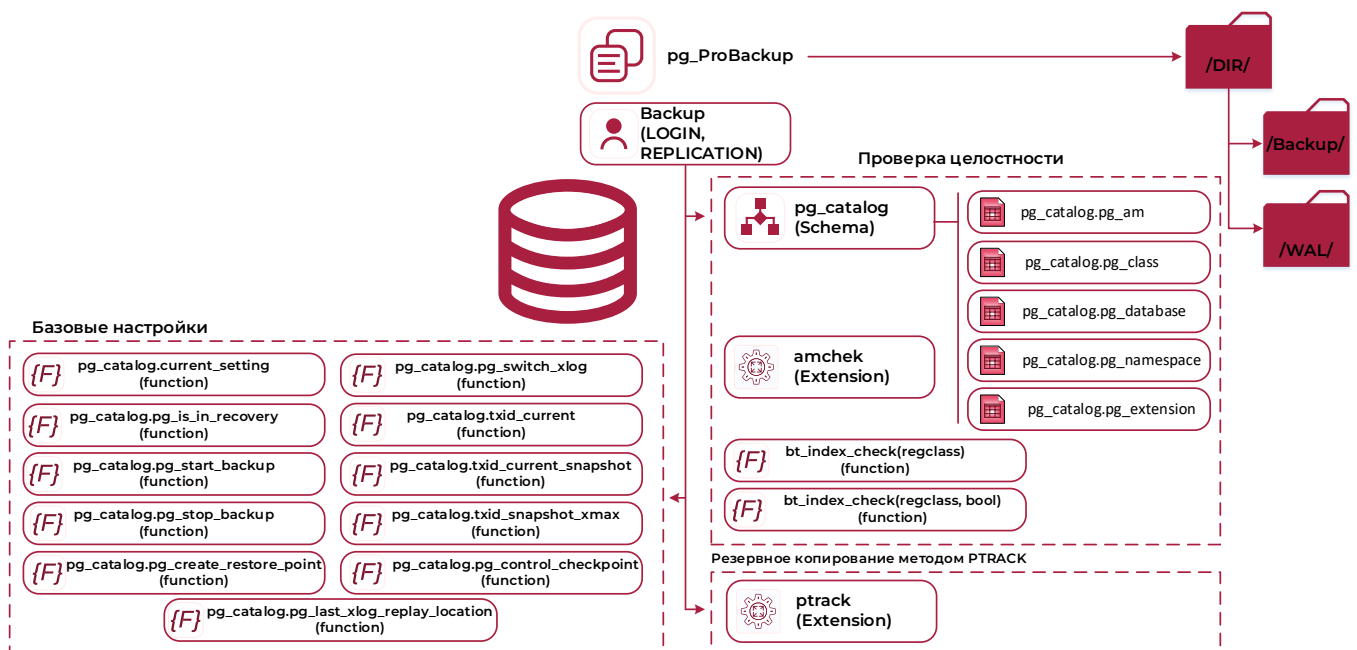


Рисунок 6.4 – Настройка резервного копирования в режиме TRACK

## 7. РЕЗЕРВНОЕ КОПИРОВАНИЕ

### 7.1. Создание резервной копии

Для создания резервной копии необходимо выполнить следующую команду:

```
pg_probackup backup -B backup_dir --instance instance_name -b backup_mode
```

В транскрипции команда имеет вид:

```
pg_probackup backup -B каталог_копий --instance имя_экземпляра -b режим_копирования
```

где режим копирования может быть следующим в соответствии с таблицей 7.1:

Таблица 7.1 – Режимы резервного копирования

Режим	Описание
FULL	создается полная резервная копия, содержащая все файлы данных кластера данных, необходимые для его восстановления
DELTA	считываются все файлы данных в каталоге данных и создается инкрементальная копия для страниц, измененных со времени предыдущего копирования
PAGE	создается инкрементальная резервная копия с файлами WAL, записанными после предыдущей полной или инкрементальной копии. Из файлов данных при этом считываются только измененные страницы
PTRACK	создается инкрементальная резервная копия со страницами, изменения в которых отслеживались в процессе работы СУБД с помощью компонента ptrack

Схема режимов резервного копирования представлена на рисунке 7.1.

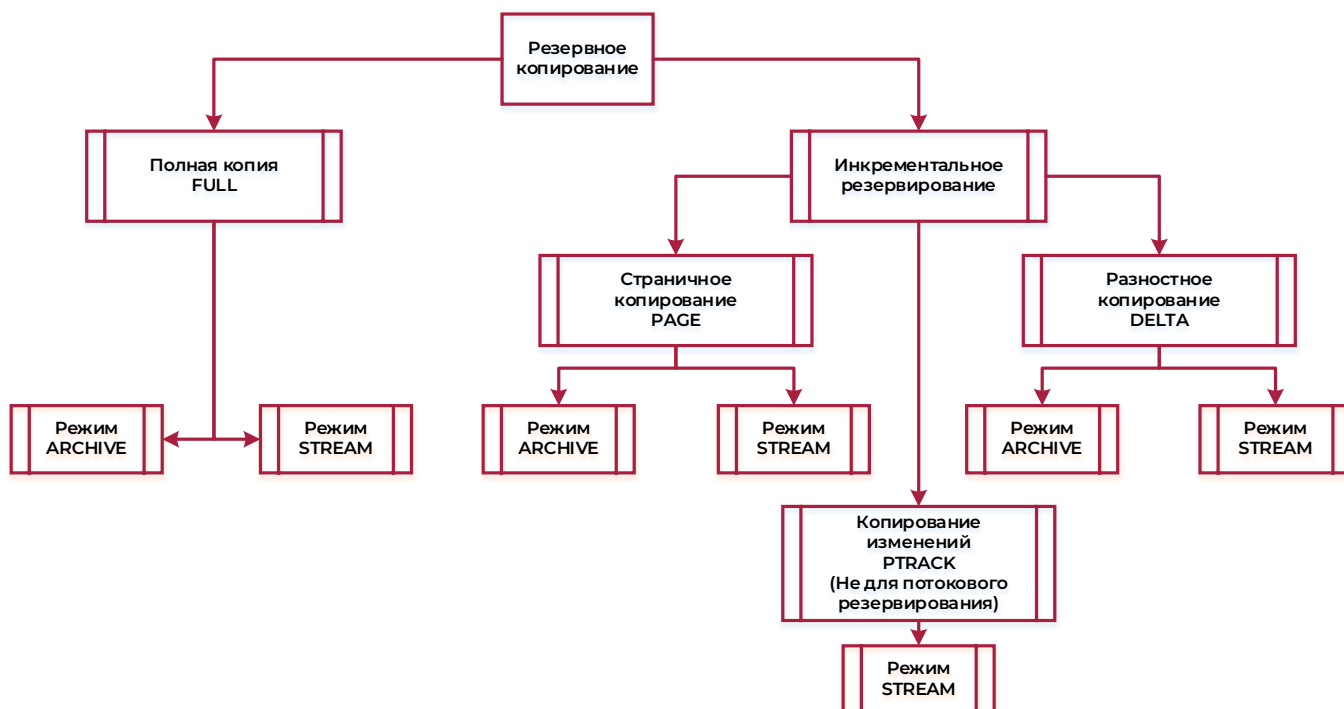


Рисунок 7.1 – Схема режимов копирования

При восстановлении кластера из инкрементальной копии pg\_ProBackup использует родительскую полную копию и все инкрементальные копии между ними, которые в совокупности образуют «цепочку копий».



Перед созданием инкрементальных копий, необходимо сделать, как минимум, одну полную копию.

### 7.1.1. Режим ARCHIVE

Режим ARCHIVE используется в качестве режима доставки WAL по умолчанию.

Например, чтобы сделать полную копию в режиме доставки WAL ARCHIVE, необходимо выполнить:

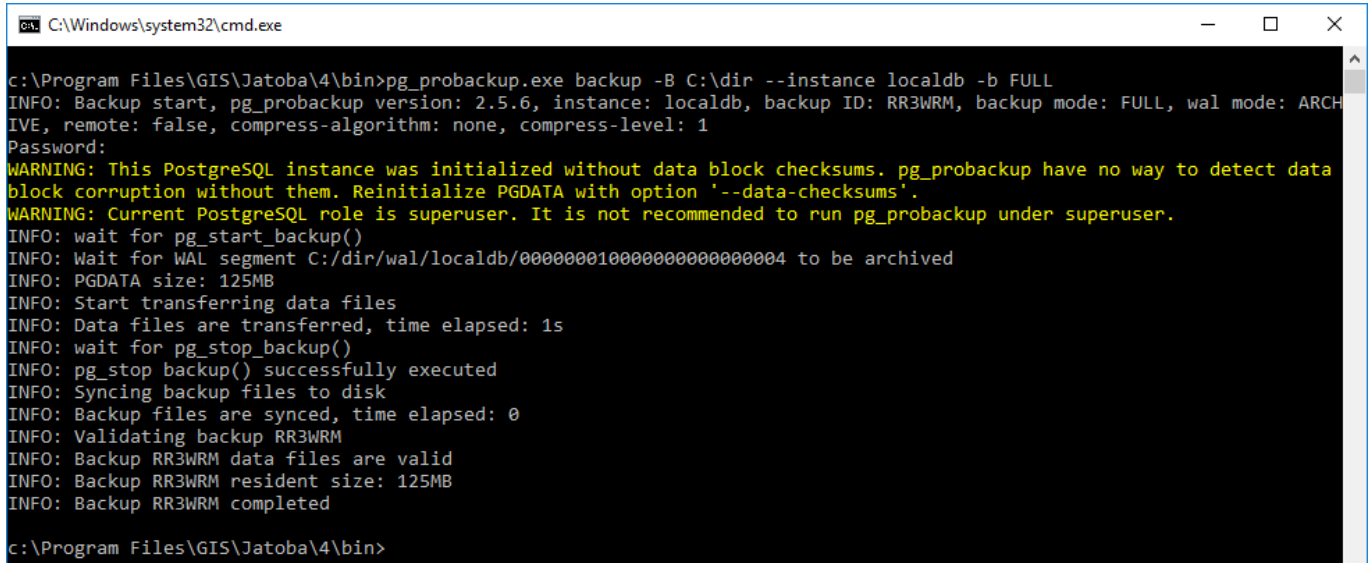
```
pg_probackup backup -B backup_dir --instance instance_name -b FULL
```

Пример:

- В ОС Windows:



```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B  
C:\dir --instance localdb -b FULL
```

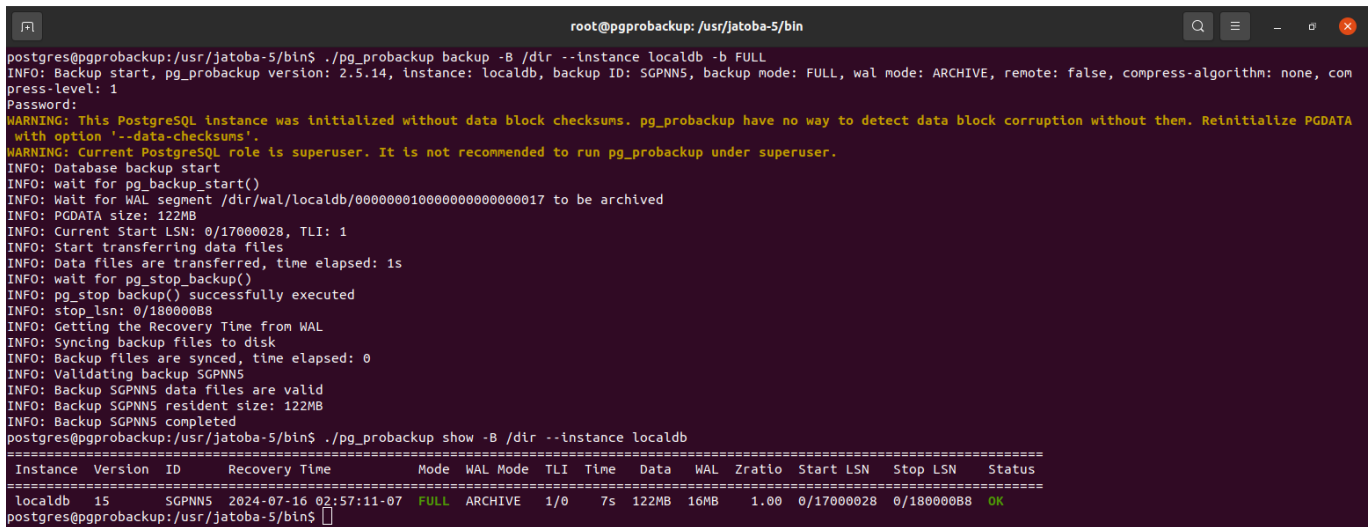


```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B C:\dir --instance localdb -b FULL  
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR3WRM, backup mode: FULL, wal mode: ARCHIVE, remote: false, compress-algorithm: none, compress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.  
INFO: wait for pg_start_backup()  
INFO: Wait for WAL segment C:/dir/wal/localdb/000000010000000000000004 to be archived  
INFO: PGDATA size: 125MB  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 1s  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: Syncing backup files to disk  
INFO: Backup files are synced, time elapsed: 0  
INFO: Validating backup RR3WRM  
INFO: Backup RR3WRM data files are valid  
INFO: Backup RR3WRM resident size: 125MB  
INFO: Backup RR3WRM completed  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 7.2 – Выполнение резервного копирования в режиме FULL.

- o В GNU Linux:

```
./pg_probackup backup -B /dir --instance localdb -b FULL
```



```
root@pgprobackup: /usr/jatoba-5/bin  
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup backup -B /dir --instance localdb -b FULL  
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SGPNN5, backup mode: FULL, wal mode: ARCHIVE, remote: false, compress-algorithm: none, compress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.  
INFO: Database backup start  
INFO: wait for pg_backup_start()  
INFO: Wait for WAL segment /dir/wal/localdb/000000010000000000000017 to be archived  
INFO: PGDATA size: 122MB  
INFO: Current Start LSN: 0/17000028, TLI: 1  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 1s  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: stop_lsn: 0/180000B8  
INFO: Getting the Recovery Time from WAL  
INFO: Syncing backup files to disk  
INFO: Backup files are synced, time elapsed: 0  
INFO: Validating backup SGPNN5  
INFO: Backup SGPNN5 data files are valid  
INFO: Backup SGPNN5 resident size: 122MB  
INFO: Backup SGPNN5 completed  
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup show -B /dir --instance localdb  
=====
```

Instance	Version	ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Start LSN	Stop LSN	Status
localdb	15	SGPNN5	2024-07-16 02:57:11-07	FULL	ARCHIVE	1/0	7s	122MB	16MB	1.00	0/17000028	0/180000B8	OK

```
=====
```

Рисунок 7.3 – Выполнение резервного копирования в режиме FULL в GNU Linux

Дополнительный вариант создания полной резервной копии с ключом archive:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL --archive  
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b FULL -  
-archive -B C:\DIR --instance localdb
```

```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b FULL --archive -B C:\DIR --instance localdb
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR3WTD, backup mode: FULL, wal mode: ARCHIVE, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: wait for pg_start_backup()
INFO: Wait for WAL segment C:/DIR/wal/localdb/000000010000000000000007 to be archived
INFO: PGDATA size: 125MB
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 1s
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup RR3WTD
INFO: Backup RR3WTD data files are valid
INFO: Backup RR3WTD resident size: 125MB
INFO: Backup RR3WTD completed
c:\Program Files\GIS\Jatoba\4\bin>_

```

Рисунок 7.4 – Создание полной резервной копии с ключом archive

Резервное копирование ARCHIVE требует организации непрерывного архивирования, посредством которого считываются сегменты WAL, требующиеся для восстановления согласованного состояния БД на момент создания копии.

В процессе резервного копирования pg\_ProBackup помещает файлы WAL, содержащие записи от Start LSN до Stop LSN, в каталог:

каталог\_копий/wal/имя\_экземпляра

Компонент pg\_ProBackup также проверяет корректность чтения записей WAL между Start LSN и Stop LSN, что позволяет исключить риск хранения в архиве поврежденного WAL.

### 7.1.2. Режим STREAM

Режим STREAM может использоваться в качестве альтернативного режима доставки WAL.

Для того, чтобы сделать полную резервную копию в потоковом режиме, необходимо добавить параметр `--stream` к команде, показанной выше:

```
pg_probackup backup -B backup_dir --instance instance_name -b FULL --stream --temp-slot
```

Пример:

- В ОС Windows:

```
C:\Program Files\GIS\Jatoba\4\data\base>pg_probackup.exe backup -B C:\dir --instance localdb -b FULL --stream
```

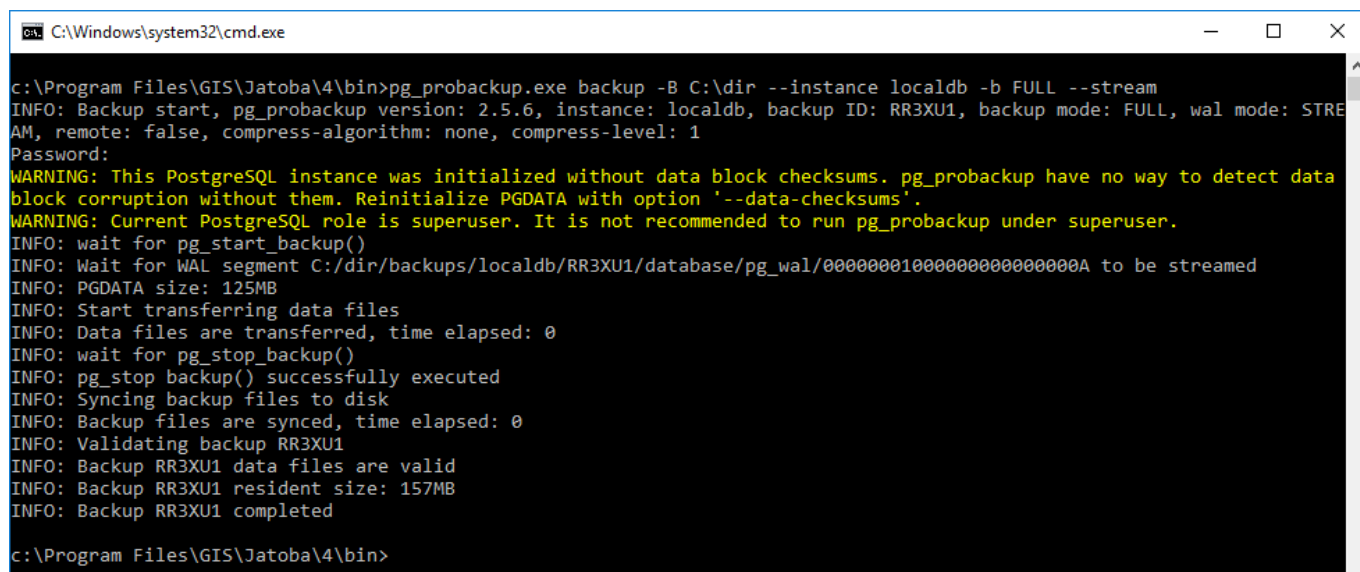


Рисунок 7.5 – Пример успешного создания резервной копии в режиме STREAM

- В GNU Linux:

```
./pg_probackup backup -B /dir --instance localdb -b FULL --stream
```

```

root@pgprobackup: /usr/jatoba-5/bin
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup backup -B /dir --instance localdb -b FULL --stream
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SGP08R, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, comp
ress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA
with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: Database backup start
INFO: wait for pg_backup_start()
INFO: Wait for WAL segment /dir/backups/localdb/SGP08R/database/pg_wal/0000000100000000000000000000001A to be streamed
INFO: PGDATA size: 122MB
INFO: Current Start LSN: 0/1A000028, TLI: 1
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 0
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: stop_lsn: 0/1A0001E0
INFO: Getting the Recovery Time from WAL
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup SGP08R
INFO: Backup SGP08R data files are valid
INFO: Backup SGP08R resident size: 138MB
INFO: Backup SGP08R completed
postgres@pgprobackup: /usr/jatoba-5/bin$

```

Рисунок 7.6 - Пример успешного создания резервной копии в режиме STREAM в GNU Linux

Необязательный параметр `--temp-slot` обеспечивает наличие необходимых сегментов в случае прокрутки WAL до завершения резервного копирования.

В отличие от копий ARCHIVE, копии типа STREAM включают все сегменты WAL, необходимые для восстановления согласованного состояния БД, на момент создания копии.

В процессе выполнения команды `backup` программа `pg_ProBackup` передает файлы WAL, содержащие записи от Start LSN до Stop LSN в

каталог\_копий/backups/имя\_экземпляра/ид\_резервной\_копии/databas  
e/pg\_wal

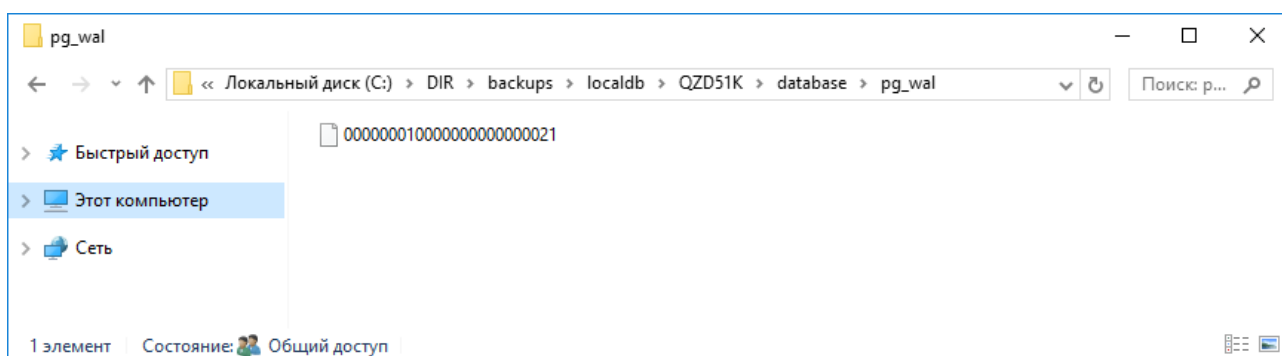


Рисунок 7.7 – Пример структуры каталогов

Чтобы исключить риск архивирования испорченных файлов WAL, `pg_ProBackup` также проверяет, что записи WAL от Start LSN до Stop LSN читаются корректно.

При использовании непрерывного архивирования, копирование в режиме STREAM может быть полезно в следующих случаях:

- Копии типа STREAM могут быть восстановлены на сервере, не имеющем файлового доступа к архиву WAL.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

– Копии типа STREAM позволяют восстановить состояние БД на тот момент времени, для которого уже нет файлов WAL

### 7.1.3. Режим PAGE

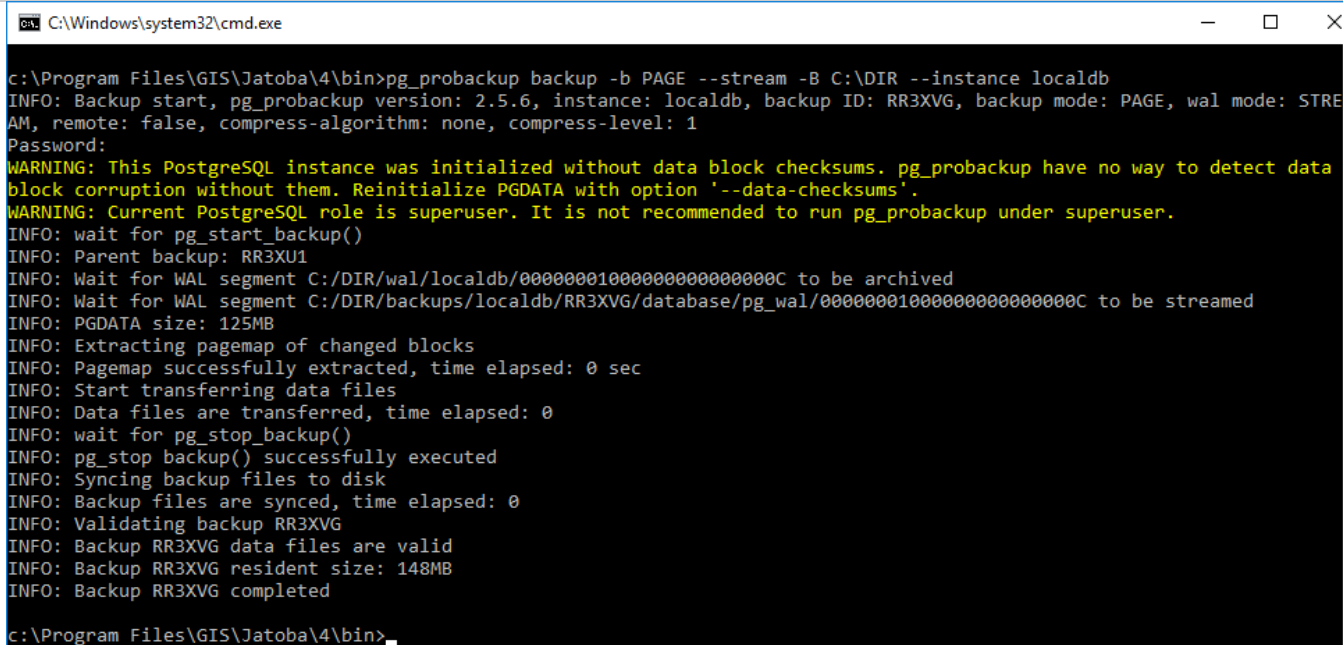
Для того, чтобы сделать копию в режиме PAGE, необходимо выполнить команду:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
PAGE --stream
```

Например:

- В ОС Windows:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b PAGE -  
-stream -B C:\DIR --instance localdb
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b PAGE --stream -B C:\DIR --instance localdb  
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR3XVG, backup mode: PAGE, wal mode: STRE  
AM, remote: false, compress-algorithm: none, compress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data  
block corruption without them. Reinitialize PGDATA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.  
INFO: wait for pg_start_backup()  
INFO: Parent backup: RR3XU1  
INFO: Wait for WAL segment C:/DIR/wal/localdb/000000010000000000000000C to be archived  
INFO: Wait for WAL segment C:/DIR/backups/localdb/RR3XVG/database/pg_wal/000000010000000000000000C to be streamed  
INFO: PGDATA size: 125MB  
INFO: Extracting pagemap of changed blocks  
INFO: Pagemap successfully extracted, time elapsed: 0 sec  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 0  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: Syncing backup files to disk  
INFO: Backup files are synced, time elapsed: 0  
INFO: Validating backup RR3XVG  
INFO: Backup RR3XVG data files are valid  
INFO: Backup RR3XVG resident size: 148MB  
INFO: Backup RR3XVG completed  
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 7.8 – Выполнение резервного копирования в режиме PAGE

- В GNU Linux:

```
./pg_probackup backup -b PAGE --stream -B /dir --instance localdb
```

```

root@pgprobackup: /usr/jatoba-5/bin
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup backup -b PAGE --stream -B /dir --instance localdb
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SGPOCH, backup mode: PAGE, wal mode: STREAM, remote: false, compress-algorithm: none, comp
ress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA
with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: Database backup start
INFO: wait for pg_backup_start()
INFO: Parent backup: SGPOCH
INFO: Wait for WAL segment /dir/wal/localdb/00000001000000000000000000000001C to be archived
INFO: Wait for WAL segment /dir/backups/localdb/SGPOCH/database/pg_wal/00000001000000000000000000000001C to be streamed
INFO: PGDATA size: 122MB
INFO: Current Start LSN: 0/1C000028, TLI: 1
INFO: Parent Start LSN: 0/1A000028, TLI: 1
INFO: Extracting pagemap of changed blocks
INFO: Pagemap successfully extracted, time elapsed: 0 sec
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 0
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: stop_lsn: 0/1D0000B8
INFO: Getting the Recovery Time from WAL
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup SGPOCH
INFO: Backup SGPOCH data files are valid
INFO: Backup SGPOCH resident size: 148MB
INFO: Backup SGPOCH completed
postgres@pgprobackup: /usr/jatoba-5/bin$

```

Рисунок 7.9 – Выполнение резервного копирования в режиме PAGE в GNU Linux

#### 7.1.4. Режим DELTA

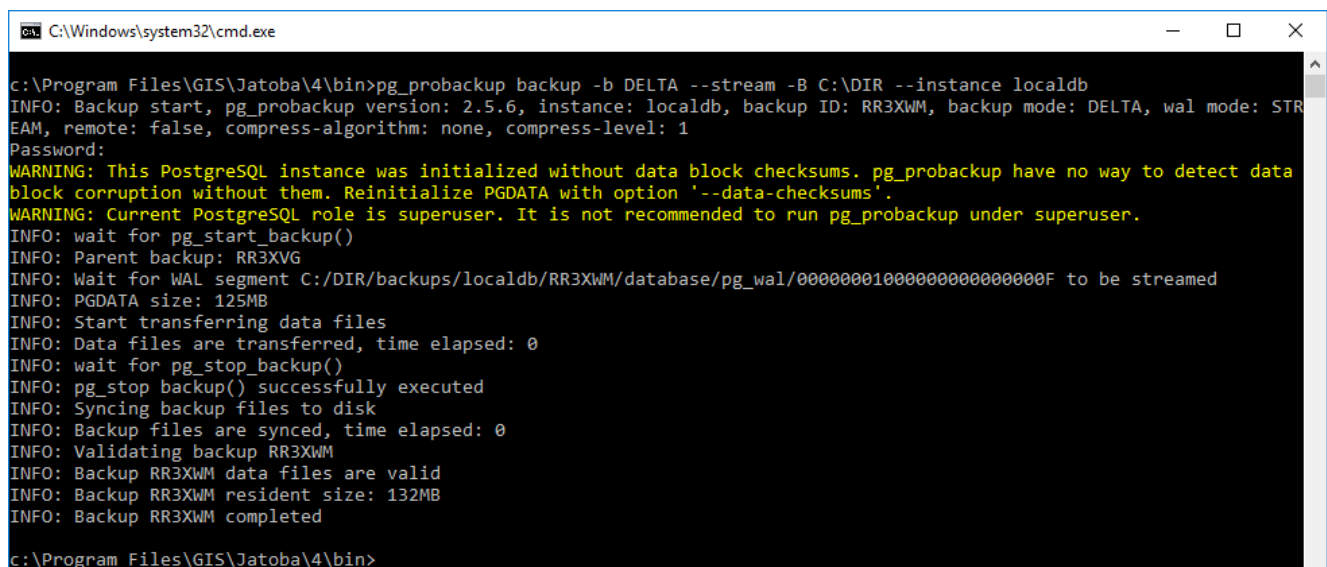
Для того, чтобы сделать полную копию в режиме доставки DELTA, необходимо ВЫПОЛНИТЬ:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
DELTA --stream
```

Например:

- В ОС Windows:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b DELTA  
--stream -B C:\DIR --instance localdb
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup backup -b DELTA --stream -B C:\DIR --instance localdb  
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR3XWM, backup mode: DELTA, wal mode: STR  
EAM, remote: false, compress-algorithm: none, compress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data  
block corruption without them. Reinitialize PGDATA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.  
INFO: wait for pg_start_backup()  
INFO: Parent backup: RR3XVG  
INFO: Wait for WAL segment C:/DIR/backups/localdb/RR3XWM/database/pg_wal/00000001000000000000000F to be streamed  
INFO: PGDATA size: 125MB  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 0  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: Syncing backup files to disk  
INFO: Backup files are synced, time elapsed: 0  
INFO: Validating backup RR3XWM  
INFO: Backup RR3XWM data files are valid  
INFO: Backup RR3XWM resident size: 132MB  
INFO: Backup RR3XWM completed  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 7.10 – Выполнение резервного копирования в режиме DELTA

- В GNU Linux:

```
./pg_probackup backup -b DELTA --stream -B /dir --instance localdb
```



```

root@pgprobackup: /usr/jatoba-5/bin
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup backup -b DELTA --stream -B /dir --instance localdb
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SGPOHA, backup mode: DELTA, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: Database backup start
INFO: wait for pg_backup_start()
INFO: Parent backup: SGPOCH
INFO: Wait for WAL segment /dir/backups/localdb/SGPOHA/database/pg_wal/00000001000000000000000000000000F to be streamed
INFO: PGDATA size: 122MB
INFO: Current Start LSN: 0/1F000028, TLI: 1
INFO: Parent Start LSN: 0/1C000028, TLI: 1
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 0
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: stop_lsn: 0/1F0001E0
INFO: Getting the Recovery Time from WAL
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup SGPOHA
INFO: Backup SGPOHA data files are valid
INFO: Backup SGPOHA resident size: 116MB
INFO: Backup SGPOHA completed
postgres@pgprobackup: /usr/jatoba-5/bin$

```

Рисунок 7.11 – Выполнение резервного копирования в режиме DELTA в GNU Linux

### 7.1.5. Режим PTRACK

Режим PTRACK может использоваться в качестве альтернативного режима доставки WAL.

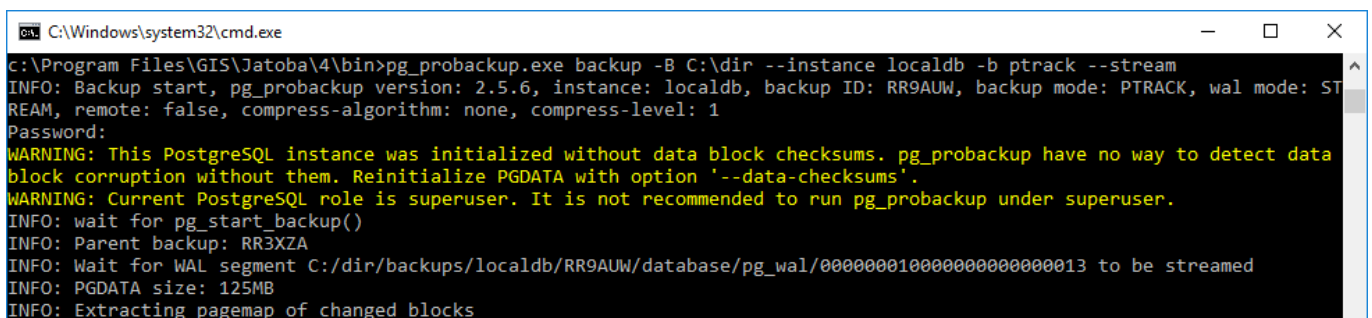
Команда выполнения резервного копирования имеет синтаксис:

```
pg_probackup backup -B backup_dir --instance instance_name -b FULL --PTRACK
```

Пример выполнения команды:

- В ОС Windows:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B C:\dir --instance localdb -b ptrack --stream
```

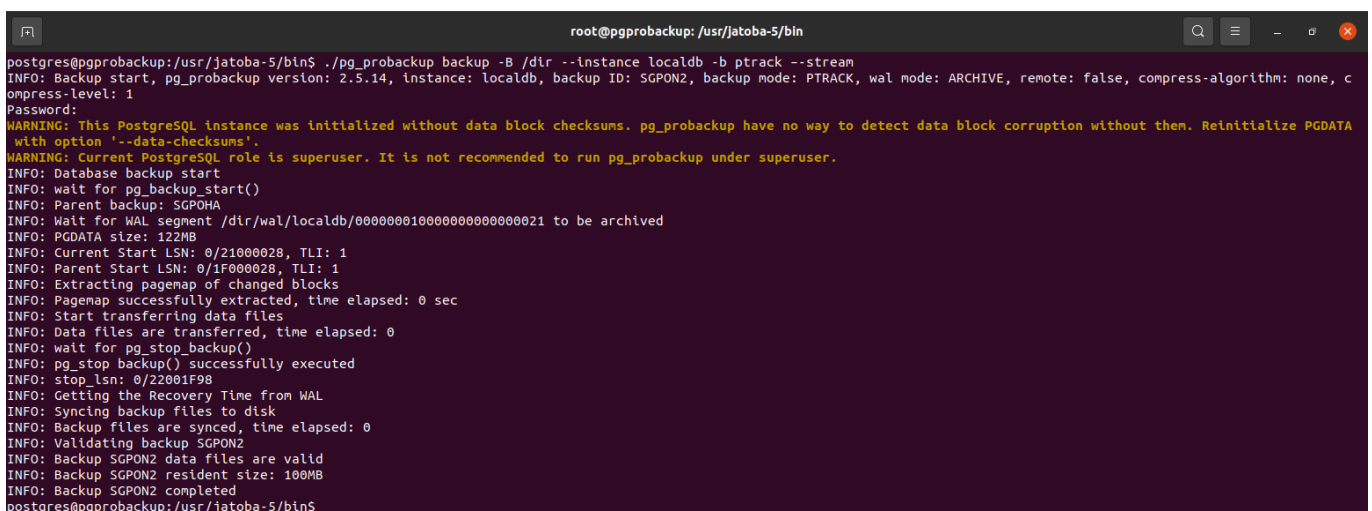


```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B C:\dir --instance localdb -b ptrack --stream
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR9AUW, backup mode: PTRACK, wal mode: STREAM, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: wait for pg_start_backup()
INFO: Parent backup: RR3XZA
INFO: Wait for WAL segment C:/dir/backups/localdb/RR9AUW/database/pg_wal/00000001000000000000000013 to be streamed
INFO: PGDATA size: 125MB
INFO: Extracting pagemap of changed blocks
```

Рисунок 7.12 – Резервное копирование в режиме PTRACK

- В GNU Linux:

```
./pg_probackup backup -B /dir --instance localdb -b ptrack --stream
```



```
root@pgprobackup: /usr/jatoba-5/bin
postgres@pgprobackup: /usr/jatoba-5/bin$ ./pg_probackup backup -B /dir --instance localdb -b ptrack --stream
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SGPON2, backup mode: PTRACK, wal mode: ARCHIVE, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: Database backup start
INFO: wait for pg_backup_start()
INFO: Parent backup: SGPON2
INFO: Wait for WAL segment /dir/wal/localdb/0000000100000000000000000021 to be archived
INFO: PGDATA size: 122MB
INFO: Current Start LSN: 0/21000028, TLI: 1
INFO: Parent Start LSN: 0/1F000028, TLI: 1
INFO: Extracting pagemap of changed blocks
INFO: Pagemap successfully extracted, time elapsed: 0 sec
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 0
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: stop_lsn: 0/22001F98
INFO: Getting the Recovery Time from WAL
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup SGPON2
INFO: Backup SGPON2 data files are valid
INFO: Backup SGPON2 resident size: 100MB
INFO: Backup SGPON2 completed
postgres@pgprobackup: /usr/jatoba-5/bin$
```

Рисунок 7.13 – Резервное копирование в режиме PTRACK в GNU Linux

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

### 7.1.6. Архивирование внешних каталогов

Для выполнения архивирования каталога, размещенного вне каталога данных, нужно воспользоваться необязательным параметром `--external-dirs`, в котором задается путь к нужному каталогу.

Если необходимо выполнить архивирование нескольких внешних каталогов, их пути нужно разделить двоеточием в ОС Linux или точкой с запятой в ОС Windows.

Для того, чтобы в системе Linux включить каталоги `/etc/dir1` и `/etc/dir2` в полную копию экземпляра (`instance_name`), которая будет размещаться в каталоге копий (`backup_dir`), нужно выполнить:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL --external-dirs=/etc/dir1:/etc/dir2
```

Для того, чтобы включить каталоги `C:\test1` и `C:\test2` в полную копию, в ОС семейства Windows нужно выполнить команду с синтаксисом:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL --external-dirs=C:\dir1;C:\dir2
```

Для включения в состав резервных копий внешних каталогов `c:\test1` и `C:\test2` под управлением ОС Windows, требуется выполнить команду:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B  
C:\dir --instance localdb -b FULL --stream --external-  
dirs=c:\test1;C:\test2
```

```
C:\Windows\system32\cmd.exe

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe backup -B C:\dir --instance localdb -b FULL --stream --external-dirs=
c:\test1\C:\test2
INFO: Backup start, pg_probackup version: 2.5.6, instance: localdb, backup ID: RR9B2K, backup mode: FULL, wal mode: STRE
AM, remote: false, compress-algorithm: none, compress-level: 1
Password:
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data
block corruption without them. Reinitialize PGDATA with option '--data-checksums'.
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.
INFO: wait for pg_start_backup()
INFO: Wait for WAL segment C:/dir/backups/localdb/RR9B2K/database/pg_wal/00000001000000000000000015 to be streamed
INFO: PGDATA size: 125MB
INFO: Start transferring data files
INFO: Data files are transferred, time elapsed: 4s
INFO: wait for pg_stop_backup()
INFO: pg_stop_backup() successfully executed
INFO: Syncing backup files to disk
INFO: Backup files are synced, time elapsed: 0
INFO: Validating backup RR9B2K
INFO: Backup RR9B2K data files are valid
INFO: Backup RR9B2K resident size: 157MB
INFO: Backup RR9B2K completed

c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 7.14 – Добавление внешних каталогов в резервную копию

Для каждого внешнего каталога pg\_ProBackup создает отдельный подкаталог в каталоге резервной копии и рекурсивно копирует в него все содержимое внешнего каталога.

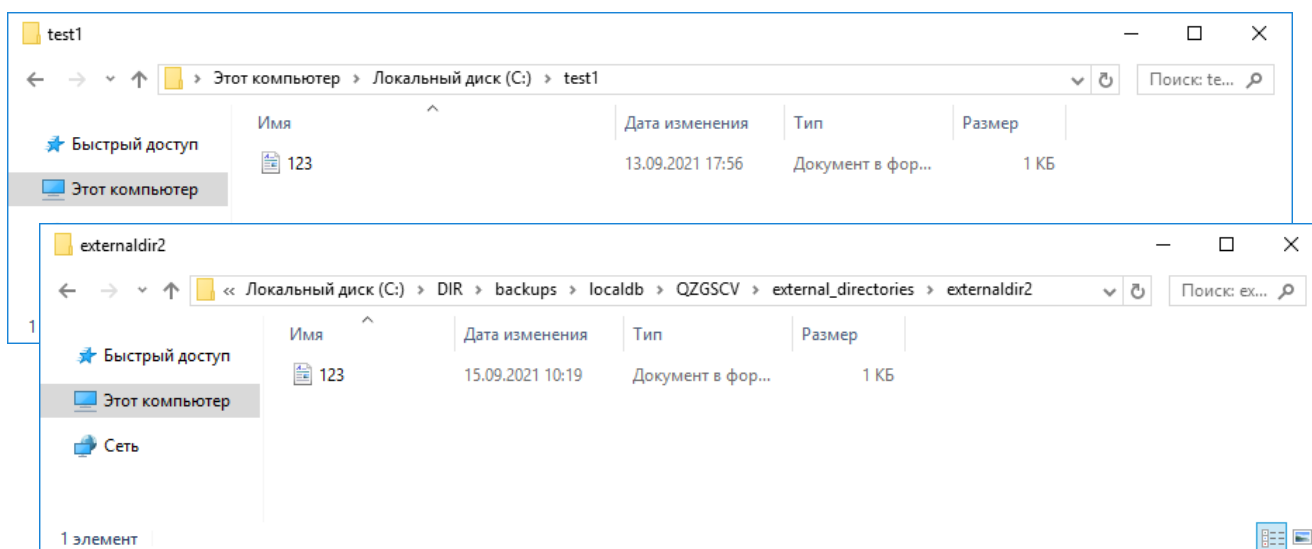


Рисунок 7.15 – Структура каталогов при резервном копировании внешних каталогов

Для того, чтобы нужные каталоги включались в каждую резервную копию кластера, их список можно сохранить в файле конфигурации pg\_probackup.conf, воспользовавшись командой set-config с ключом --external-dirs.

## 7.2. Параметры команды backup – резервное копирование

Команда backup для создания резервной копии экземпляра СУБД имеет следующий синтаксис:

```
pg_probackup backup -B backup_dir -b backup_mode --instance
instance_name [--help] [-j num_threads] [--progress] [-C] [--
stream [-S slot_name] [--temp-slot]] [--backup-pg-log] [--no-
validate] [--skip-block-validation] [-w --no-password] [-W --
password] [--archive-timeout=timeout] [--external-
dirs=external_directory_path] [--no-sync] [--note=backup_note]
[connection_options] [compression_options] [retention_options]
[pinning_options] [logging_options]
```

В транскрипции команда имеет вид:

```
pg_probackup backup -B каталог_копий -b режим_копирования --
instance имя_экземпляра [--help] [-j число_потоков] [--
progress] [-C] [--stream [-S slot_name] [--temp-slot]] [--
backup-pg-log] [--no-validate] [--skip-block-validation] [-w --
no-password] [-W --password] [--archive-timeout=тайм-аут] [--
external-dirs=external_directory_path] [--no-sync] [--
note=заметка_к_копии] [параметры_соединения] [параметры_сжатия]
[параметры_хранения] [параметры_закрепления]
[параметры_журнала]
```

Параметры команды backup:

```
-b mode
--backup-mode=mode
```

Выбор режима резервного копирования. Поддерживаются следующие режимы:

- FULL – создается полная резервная копия, содержащая все файлы данных кластера данных, необходимые для его восстановления.
- DELTA – считываются все файлы данных в каталоге данных и создается инкрементальная копия для страниц, измененных со времени предыдущего копирования.
- PAGE – создается инкрементальная резервная копия с файлами WAL, которые были изменены со времени последней полной или инкрементальной копии.
- PTRACK – создается инкрементальная резервная копия со страницами, изменения в которых отслеживались «на лету».

```
-C
--smooth-checkpoint
```

Распределение контрольной точки в течение определенного периода времени. По умолчанию `pg_probackup` пытается произвести контрольную точку максимально быстро.

```
--stream
```

Создание потоковой резервной копии (STREAM), включая в нее все необходимые файлы WAL, получаемые от сервера по протоколу репликации.

```
--temp-slot
```

Создание временного слота физической репликации для передачи WAL с архивируемого экземпляра СУБД. Это гарантирует доступность всех нужных сегментов WAL, если в процессе копирования произойдет переключение сегментов WAL. Этот параметр может использоваться только вместе с параметром `--stream`. По умолчанию имя слота – `pg_probackup_slot`, для его изменения, воспользуйтесь параметром `--slot/-S`.

```
-S slot_name  
--slot=slot_name
```

Задание слота репликации для передачи WAL. Этот параметр можно указать только вместе с параметром `--stream`.

```
--backup-pg-log
```

Включение в резервную копию каталога `log`. Этот каталог обычно содержит журналы сообщений сервера. По умолчанию каталог `log` в копию не включается.

```
-E external_directory_path  
--external-dirs=external_directory_path
```

Включение в создаваемую копию указанного каталога, рекурсивно копируя его содержимое в отдельный подкаталог каталога резервной копии. Этот параметр полезен для архивирования скриптов, SQL-дампов и файлов конфигурации, расположенных вне каталога данных. При архивировании нескольких внешних каталогов, их пути нужно разделять двоеточием в Linux или точкой с запятой в Windows.

```
--archive-timeout=wait_time
```

Задание таймаута для архивирования сегментов WAL и потоковой передачи (в секундах). По умолчанию pg\_ProBackup ждет выполнения этих операций 300 секунд.

```
--skip-block-validation
```

Отключение проверки контрольных сумм на уровне блоков в процессе резервного копирования.

```
--no-validate
```

Пропуск автоматической проверки созданной резервной копии. Этот ключ полезен, при регулярной проверке резервных копий и при сокращении времени создания копии.

```
--no-sync
```

Не сбрасывать копируемые файлы на диск. Этот параметр позволяет несколько ускорить процесс копирования.



Использование этого параметра может привести к повреждению данных в случае аварии операционной системы или аппаратного сбоя. При его использовании рекомендуется выполнить команду *validate* сразу после завершения копирования, чтобы убедиться в отсутствии ошибок.

```
--note=backup_note
```

Задание текстовой заметки для резервной копии. Максимальный размер заметки равен 1 КБ. Значение *'none'* удаляет текущую заметку.



Если заметка к копии содержит символы перевода строки, сохранена будет только подстрока до первого перевода строки.

Дополнительные параметры команды *backup* описаны:

- в разделе 12.2 «Параметры подключения»;
- в разделе 12.1.2 «Параметры сохранения – retention»;
- в разделе 12.1.3 «Параметры закрепления резервных копий – set-backup»;

– в разделе 12.3 «Параметры архивации» и т.д.



## 8. ПРОВЕРКА ЦЕЛОСТНОСТИ

### 8.1. Проверка страниц

Если на сервере СУБД включена проверка контрольных сумм, pg\_ProBackup будет использовать их для проверки целостности файлов данных в процессе резервного копирования.

При чтении каждой страницы pg\_ProBackup проверяет, совпадает ли вычисленная сумма с контрольной суммой, хранящейся в заголовке страницы. Это гарантирует, что на сервере СУБД и в самой резервной копии не содержатся испорченные страницы.

pg\_ProBackup читает файлы данных непосредственно из файловой системы, поэтому при активной записи в момент копирования возможны ложные выявления некорректных контрольных сумм из-за частичной записи. В случае несовпадения контрольной суммы страница считывается повторно, и контрольная сумма проверяется еще раз.

Страница признается испорченной, если проверка контрольной суммы не проходит более 100 раз. В этом случае резервное копирование прерывается.

Если проверка контрольных сумм на сервере СУБД не включена, компонент pg\_ProBackup всегда проверяет целостность заголовков страниц.

### 8.2. Проверка целостности директории данных СУБД – checkdb

Чтобы убедиться в отсутствии повреждений в директории данных СУБД, необходимо выполнить следующую команду:

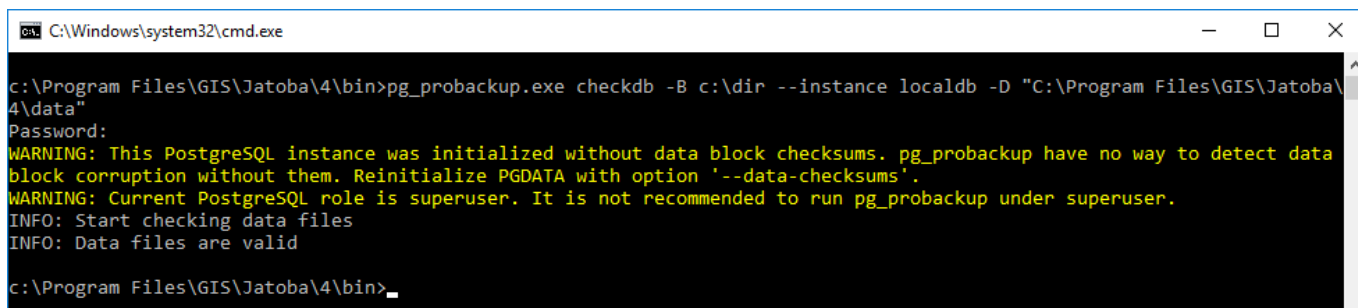
```
pg_probackup checkdb [-B backup_dir] [--instance instance_name]
[-D data_dir] [--help] [-j num_threads] [--progress] [--skip-
block-validation] [--amcheck] [--heapallindexed]
[connection_options] [logging_options]
```

В транскрипции команда имеет вид:

```
pg_probackup checkdb [-B каталог_копий] [--instance
имя_экземпляра] [-D каталог_данных] [--help] [-j число_потоков]
[--progress] [--skip-block-validation] [--amcheck]
[--heapallindexed] [параметры_соединения] [параметры_журнала]
```

Например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe checkdb -B  
c:\dir --instance localdb -D "C:\Program  
Files\GIS\Jatoba\4\data"
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe checkdb -B c:\dir --instance localdb -D "C:\Program Files\GIS\Jatoba\4\data"  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_probackup have no way to detect data  
block corruption without them. Reinitialize PGDATA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_probackup under superuser.  
INFO: Start checking data files  
INFO: Data files are valid  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 8.1 – Результат проверки целостности директории данных



Проверку целостности данных целесообразнее проводить до резервного копирования.

Эта команда выполняет физическую проверку всех файлов данных в указанном каталоге данных, проводя проверки целостности заголовков страниц, а также проверяя контрольные суммы на уровне блоков, если в СУБД включен контроль целостности. В случае обнаружения испорченной страницы checkdb продолжает работу, пока не будут проверены все страницы в директории данных.

По умолчанию pg\_ProBackup выполняет подобную проверку страниц автоматически в процессе создания копии. Команда checkdb позволяет проводить проверки страниц в директории данных, не создавая резервные копии.

Для проверки СУБД приложение pg\_ProBackup должно подключиться к ней. Обычно для получения требуемых параметров подключения достаточно указать связанный с данной СУБД копируемый экземпляр. Если параметры -B и --instance опускаются, параметры соединения и каталог данных необходимо задать в командной строке или в переменных окружения.



Проверка на физическом уровне не позволяет выявить логические несоответствия, потерю, обнуление блоков или целых файлов и другие

подобные аномалии. Провести логическую проверку в некотором объеме позволяют расширения `amcheck`.

Если помимо физической проверки необходимо проверить индексы во всех базах, возможно передать команде `checkdb` параметр `--amcheck`:

```
pg_probackup checkdb -D data_dir --amcheck [connection_options]
```

Проверку на физическом уровне можно отключить, воспользовавшись параметром `--skip-block-validation`. В этом случае задавать каталог копий (*backup\_dir*) и каталог данных (*data\_dir*) не нужно, требуется задать только параметры подключения:

```
pg_probackup checkdb --amcheck --skip-block-validation  
[connection_options]
```

Логическая проверка может быть более детальной, если запустить ее с ключом `--heapallindexed`. В этом случае будет дополнительно проверено, что в индексе действительно представлены все записи таблицы, которые должны в него попасть, но это создаст дополнительную нагрузку на процессор, память и подсистему ввода/вывода.

```
--skip-block-validation
```

Проверку файлов данных можно отключить, воспользовавшись ключом `skip-block-validation` вместе с `--amcheck`, чтобы произвести только логическую проверку индексов.

```
--heapallindexed
```

Проверку, что все записи в массиве данных, которые должны быть проиндексированы, действительно проиндексированы возможно выполнить с помощью флага `--heapallindexed` вместе с флагом `--amcheck`.

Также можно задать параметры соединения и ведения журнала.

### 8.3. Проверка целостности резервных копий – validate

Команда validate имеет следующий синтаксис:

```
pg_probackup validate -B backup_dir [--help] [--instance  
instance_name] [-i backup_id] [-j num_threads] [--progress] [--  
skip-block-validation] [recovery_target_options]  
[logging_options]
```

В транскрипции команда имеет вид:

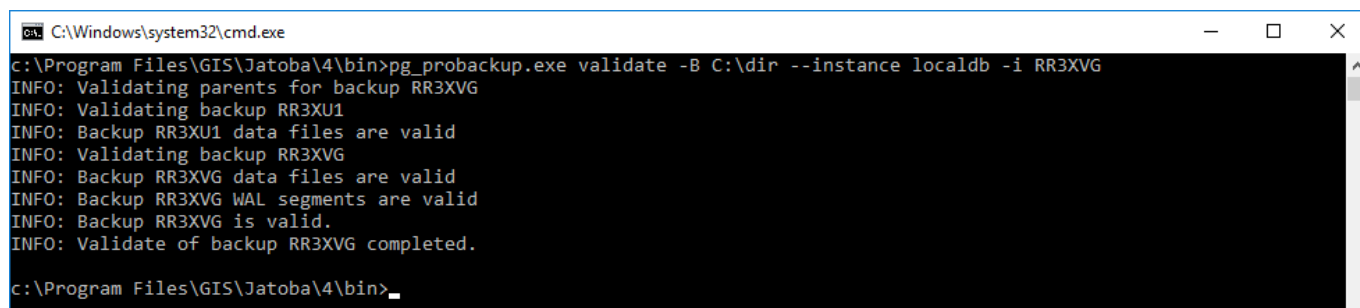
```
pg_probackup validate -B каталог_копий [--help] [--instance  
имя_экземпляра] [-i ид_резервной_копии] [-j число_потокoв] [--  
progress] [--skip-block-validation]  
[параметры_точки_восстановления] [параметры_журнала]
```

Для выполнения проверки целостности потребуется указать ID резервной копии.

Например, требуется проверить резервную копию, имеющую ID RR3XVG. В этом случае команда будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe validate -B  
C:\dir --instance localdb -i RR3XVG
```

По результатам проверки pg\_ProBackup выдаст информационные сообщения, как представлено на рисунке 8.2.

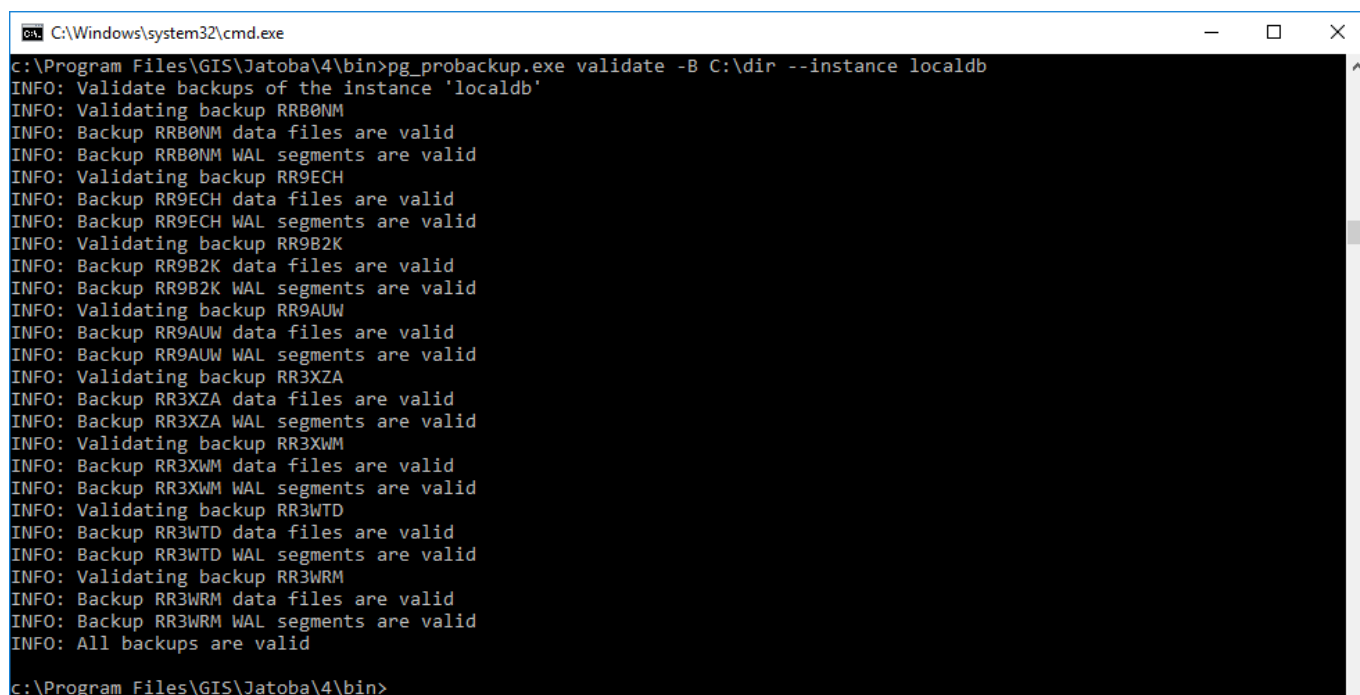


```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe validate -B C:\dir --instance localdb -i RR3XVG
INFO: Validating parents for backup RR3XVG
INFO: Validating backup RR3XU1
INFO: Backup RR3XU1 data files are valid
INFO: Validating backup RR3XVG
INFO: Backup RR3XVG data files are valid
INFO: Backup RR3XVG WAL segments are valid
INFO: Backup RR3XVG is valid.
INFO: Validate of backup RR3XVG completed.
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 8.2 – Результат проверки валидности резервной копии по ID

Для проверки целостности всех резервных копий указывать ID резервной копии или копий не требуется. Достаточно не указывать ключ `-i` и pg\_ProBackup проверит на валидность все резервные копии, находящиеся в каталоге, например:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe validate -B  
C:\dir --instance localdb
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe validate -B C:\dir --instance localdb
INFO: Validate backups of the instance 'localdb'
INFO: Validating backup RRB0NM
INFO: Backup RRB0NM data files are valid
INFO: Backup RRB0NM WAL segments are valid
INFO: Validating backup RR9ECH
INFO: Backup RR9ECH data files are valid
INFO: Backup RR9ECH WAL segments are valid
INFO: Validating backup RR9B2K
INFO: Backup RR9B2K data files are valid
INFO: Backup RR9B2K WAL segments are valid
INFO: Validating backup RR9AUW
INFO: Backup RR9AUW data files are valid
INFO: Backup RR9AUW WAL segments are valid
INFO: Validating backup RR3XZA
INFO: Backup RR3XZA data files are valid
INFO: Backup RR3XZA WAL segments are valid
INFO: Validating backup RR3XWM
INFO: Backup RR3XWM data files are valid
INFO: Backup RR3XWM WAL segments are valid
INFO: Validating backup RR3WTD
INFO: Backup RR3WTD data files are valid
INFO: Backup RR3WTD WAL segments are valid
INFO: Validating backup RR3WRM
INFO: Backup RR3WRM data files are valid
INFO: Backup RR3WRM WAL segments are valid
INFO: All backups are valid
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 8.3 – Проверка всех резервных копий, находящихся в каталоге

pg\_ProBackup вычисляет контрольные суммы для всех файлов копии в ходе резервного копирования. Процесс проверки контрольных сумм файлов называется проверкой целостности копии. По умолчанию проверка выполняется сразу после создания резервной копии и непосредственно перед восстановлением для выявления возможных повреждений резервных копий.

Для того, чтобы пропустить проверку резервной копии, нужно передать командам backup и restore параметр --no-validate.

## 9. ВОССТАНОВЛЕНИЕ РЕЗЕРВНОЙ КОПИИ

### 9.1. Восстановление сервера баз данных

Для восстановления сервера базы данных из резервной копии, необходимо выполнить команду restore как минимум со следующими параметрами:

```
pg_probackup restore -B backup_dir --instance instance_name -i  
backup_id
```



Для восстановления данных в новый каталог системной учетной записи postgres потребуется выдать полные права на каталог данных СУБД.



Одним из механизмов защиты СУБД является контроль полномочий системной учетной записи postgres в операционной системе.

В случае, если системной учетной записи postgres были добавлены привилегии, отличающиеся от установленных по умолчанию, СУБД при запуске блокируется. Чтобы избежать блокировки СУБД, запрещается добавлять дополнительные роли (привилегии) на уровне операционной системы системной учетной записи postgres.

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup restore -B  
c:\DIR --instance localdb -D "c:/Program  
Files/GIS/Jatoba/4/data2" -i RR3XWM
```

```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup restore -B c:\DIR --instance localdb -D "c:/Program Files/GIS/Jatoba/4/data2" -i RR3XWM  
INFO: Validating backup RR3XWM  
INFO: Backup RR3XWM data files are valid  
INFO: Backup RR3XWM WAL segments are valid  
INFO: Backup RR3XWM is valid.  
INFO: Restoring the database from backup at 2023-03-06 19:31:34+03  
INFO: Start restoring backup files. PGDATA size: 157MB  
INFO: Backup files are restored. Transferred bytes: 157MB, time elapsed: 0  
INFO: Restore incremental ratio (less is better): 100% (157MB/157MB)  
INFO: Syncing restored files to disk  
INFO: Restored backup files are synced, time elapsed: 1s  
INFO: Restore of backup RR3XWM completed.  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 9.1 – Восстановление БД

где:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

*каталог копий (backup\_dir)* – каталог, в котором хранятся все файлы резервных копий и метаданные.

*имя экземпляра (instance)* – имя экземпляра резервной копии, которая будет восстановлена.

*ID* – идентификатор резервной копии, определяющий из какой резервной копии будет восстановлен сервер баз данных.



Если параметр *ID* не задан, pg\_ProBackup использует последнюю подходящую копию для заданного экземпляра.

Компоненте pg\_ProBackup при выборе для восстановления инкрементальной копии, автоматически восстанавливает нижележащую полную копию и затем последовательно применяет все необходимые добавления.

После окончания работы команды restore необходимо запустить службу базы данных.

Если восстанавливалась копия типа STREAM, восстановление завершается моментально, и СУБД возвращается в согласованное состояние на момент времени, в который была сделана резервная копия.

Для копий типа ARCHIVE СУБД воспроизводит все имеющиеся в архиве сегменты WAL, в результате чего восстанавливается самое последнее состояние СУБД на текущей линии времени. Это поведение можно изменить, определив параметры точки восстановления для команды restore, в соответствии с описанием в подразделе 9.4.

Если резервная копия, подлежащая восстановлению, содержит табличные пространства, pg\_ProBackup по умолчанию восстанавливает их в исходные расположения. Для смены расположения табличных пространств, нужно воспользоваться параметром --tablespace-mapping/-T и задать абсолютные пути к предыдущему и новому каталогу табличного пространства.



Если расположение табличных пространств не изменять при восстановлении данных на том же сервере произойдет ошибка, так как восстанавливаемые данные будут записываться в те же каталоги.

Если путь содержит знак «равно (=)», экранировать его обратной косой чертой. Данный параметр может указываться неоднократно для перемещения нескольких табличных пространств.

Например:

```
pg_probackup restore -B backup_dir --instance instance_name -D  
data_dir -j 4 -i backup_id -T  
tablespace1_dir=tablespace1_newdir -T  
tablespace2_dir=tablespace2_newdir
```



По умолчанию команда restore проверяет указанную резервную копию перед восстановлением. Ключ --no-validate применяется, чтобы пропустить проверку и ускорить восстановление.

## 9.2. Инкрементальное восстановление

Скорость восстановления резервной копии можно значительно увеличить, заменяя в существующем каталоге данных СУБД только некорректные или измененные страницы. Это можно реализовать, используя параметры инкрементального восстановления с командой restore.

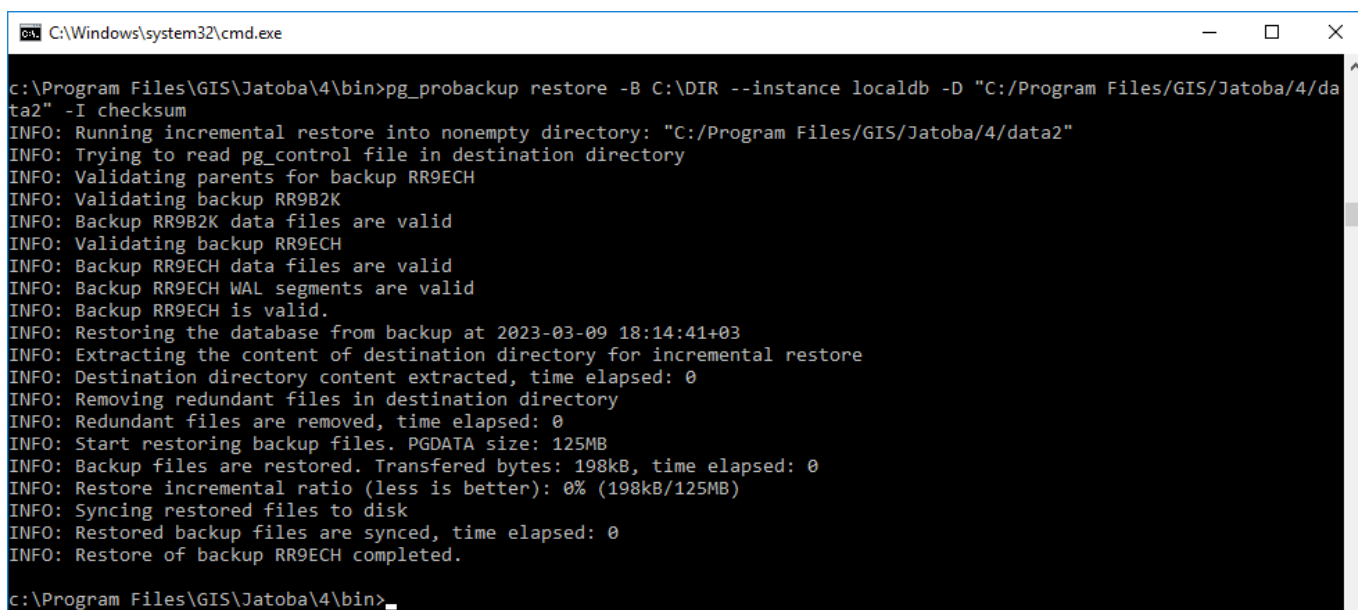
Чтобы восстановить сервер баз данных из резервной копии в инкрементальном режиме, необходимо выполнить команду restore со следующими параметрами:

```
pg_probackup restore -B backup_dir --instance instance_name -D  
data_dir -I incremental_mode
```

Командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup restore -B  
C:\DIR --instance localdb -D "C:/Program  
Files/GIS/Jatoba/4/data2" -I checksum
```





```
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup restore -B C:\DIR --instance localdb -D "C:/Program Files/GIS/Jatoba/4/data2" -I checksum
INFO: Running incremental restore into nonempty directory: "C:/Program Files/GIS/Jatoba/4/data2"
INFO: Trying to read pg_control file in destination directory
INFO: Validating parents for backup RR9ECH
INFO: Validating backup RR9B2K
INFO: Backup RR9B2K data files are valid
INFO: Validating backup RR9ECH
INFO: Backup RR9ECH data files are valid
INFO: Backup RR9ECH WAL segments are valid
INFO: Backup RR9ECH is valid.
INFO: Restoring the database from backup at 2023-03-09 18:14:41+03
INFO: Extracting the content of destination directory for incremental restore
INFO: Destination directory content extracted, time elapsed: 0
INFO: Removing redundant files in destination directory
INFO: Redundant files are removed, time elapsed: 0
INFO: Start restoring backup files. PGDATA size: 125MB
INFO: Backup files are restored. Transferred bytes: 198kB, time elapsed: 0
INFO: Restore incremental ratio (less is better): 0% (198kB/125MB)
INFO: Syncing restored files to disk
INFO: Restored backup files are synced, time elapsed: 0
INFO: Restore of backup RR9ECH completed.
c:\Program Files\GIS\Jatoba\4\bin>_
```

Рисунок 9.2 – Выполнение частичного восстановления

Возможны следующие варианты инкрементального восстановления:

– *CHECKSUM* – выполнит чтение всех файлов данных в целевом каталоге, проверит заголовок и контрольную сумму каждой страницы и заменит только некорректные страницы, а также страницы, в которых контрольная сумма и LSN отличаются от значений соответствующей страницы в копии.



Инкрементальный режим CHECKSUM – самый простой и надежный режим.

Рекомендуется использовать его по умолчанию.

– *LSN* – выполнит чтение файла pg\_control в каталоге данных, получит из него значения REDO LSN и REDO TLI, позволяющие определить точку в истории (точку сдвига), с которой состояние каталога данных сдвинулись с цепочки резервных копий. Если точка сдвига находится за пределами истории резервных копий, восстановление прерывается. Если точка сдвига найдена, прочитываются все файлы данных в каталоге данных, проверяется заголовок и контрольная сумма на каждой странице, а затем заменяются только страницы с неверной контрольной суммой или с LSN, превышающим позицию точки сдвига.

В этом режиме обеспечивается увеличенная скорость по сравнению с режимом CHECKSUM, но требуется выполнение двух условий:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

- в целевом каталоге должны быть включены контрольные суммы (см. `data_checksums`), чтобы избежать повреждения данных из-за изменения вспомогательных битов. В случае отсутствия контрольных сумм операция восстановления прерывается;

- необходимо синхронизировать файл `pg_control` с состоянием каталога данных.



Это условие нельзя проверить перед выполнением восстановления, поэтому гарантировать правильность информации в `pg_control` должен пользователь.

Использовать режим LSN не рекомендуется в случаях, когда файл `pg_control` поврежден или его содержимому нельзя доверять, например:

- после выполнения `pg_resetxlog`,
- после восстановления из копии без процедуры воспроизведения журнала.

– *NONE* – обычное восстановление без инкрементальных оптимизаций.

Вне зависимости от выбранного инкрементального режима, `pg_probackup` будет проверять, что с заданным целевым каталогом не работает процесс `postmaster` и значения `system-identifier` в целевом экземпляре и копии совпадают.

Параметры инкрементального восстановления дополнительно описаны в п. 12.4.

### 9.3. Частичное восстановление

Для восстановления только некоторых баз данных нужно выполнить команду `restore` как минимум со следующими параметрами:

```
pg_probackup restore -B backup_dir --instance instance_name --  
db-include=database_name
```

Ключ `--db-include` может добавляться многократно. Например, чтобы восстановить только базы `db1` и `db2`, нужно выполнить следующую команду:

```
pg_probackup restore -B backup_dir --instance instance_name --  
db-include=db1 --db-include=db2
```

Чтобы исключить одну или несколько баз из числа восстанавливаемых, нужно использовать ключ `--db-exclude`:

```
pg_probackup restore -B backup_dir --instance instance_name --  
db-exclude=database_name
```

Ключ `--db-exclude` может добавляться многократно. Например, чтобы при восстановлении исключить базы данных `db1` и `db2`, нужно выполнить следующую команду:

```
pg_probackup restore -B backup_dir --instance instance_name --  
db-exclude=db1 --db-exclude=db2
```

Механизм частичного восстановления рассчитан на то, что в процессе восстановления СУБД наличие пустых файлов не вызовет ошибки, так как файлы исключенных баз данных восстанавливаются пустыми.

#### 9.4. Выполнение восстановления на момент времени (PITR)

Если настраивать непрерывное архивирование WAL до создания резервных копий, можно восстановить состояние сервера баз данных на любой момент времени (до заданной точки восстановления), используя параметры точки восстановления с командой `restore`.

Для восстановления на определенный момент времени может использоваться копия типа `STREAM` или `ARCHIVE`, при этом обязательно наличие архива WAL на момент создания копии или раньше. Если параметр `-i/--backup-id` не задан, `pg_ProBackup` автоматически выбирает резервную копию, ближайшую к заданной цели восстановления, и начинает процесс восстановления.

Для восстановления состояния сервера баз данных на определенный момент времени, необходимо указать это время в параметре `--recovery-target-time`, в формате `timestamp`. Например:

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target-time="2021-05-18 14:18:11+03"
```

Для восстановления состояния до определенной транзакции, нужно воспользоваться ключом `--recovery-target-xid`:

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target-xid=687
```

Для восстановления состояния до определенной позиции в журнале (LSN), нужно воспользоваться ключом `--recovery-target-lsn`:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target-lsn=16/B374D848
```

Для восстановления состояния до заданной именованной точки восстановления, нужно воспользоваться ключом `--recovery-target-name`:

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target-name="before_app_upgrade"
```

Для восстановления последнего возможного состояния, исходя из содержимого архива WAL, нужно передать в параметре `--recovery-target` значение `latest`:

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target="latest"
```

Для восстановления самого раннего из возможных согласованных состояний нужно передать в параметре `--recovery-target` значение `immediate`:

```
pg_probackup restore -B backup_dir --instance instance_name --  
recovery-target='immediate'
```

### 9.5. Запуск pg\_ProBackup в параллельных потоках

Команды `backup`, `restore`, `merge`, `delete`, `checkdb` и `validate` могут выполняться в несколько параллельных потоков, что позволяет ускорять работу pg\_ProBackup при наличии достаточных ресурсов (ядер процессора, производительности дисковой подсистемы и сети).

Параллельным выполнением управляет ключ командной строки `-j/--threads`. Например, чтобы запустить резервное копирование в четыре параллельных потока, нужно выполнить:

```
pg_probackup backup -B backup_dir --instance instance_name -b  
FULL -j 4
```



Восстановление происходит в параллельном режиме только на этапе копирования данных из каталога копий в каталог данных СУБД. Запуск сервера СУБД и воспроизведение им записей из WAL может происходить только последовательно.

## 10. СПРАВКА ПО КОМАНДНОЙ СТРОКЕ

### 10.1. Команды pg\_ProBackup

В этом подразделе приведены команды pg\_ProBackup. Необязательные параметры этих команд заключаются в квадратные скобки. Более подробно все параметры команд описываются в разделе 11.

#### 10.1.1. Версия компонента – version

pg\_probackup version – выводит версию pg\_ProBackup.

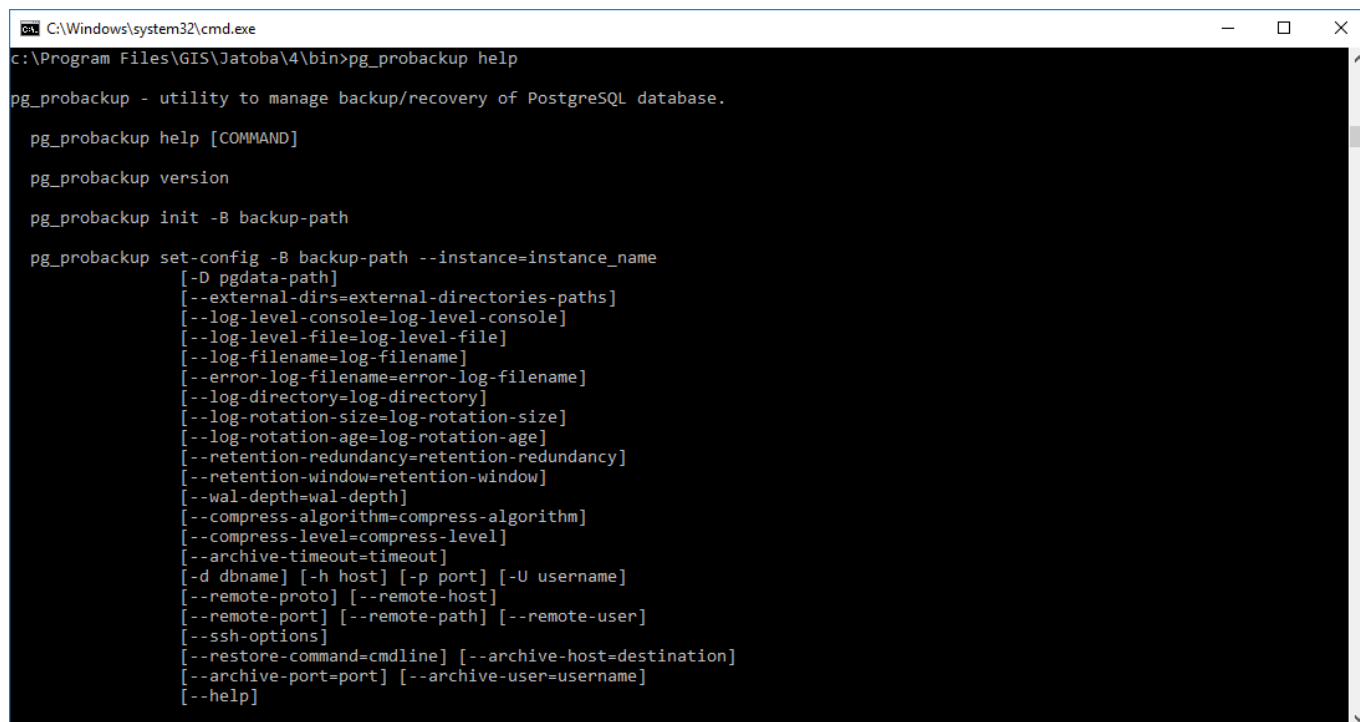
#### 10.1.2. Вызов справочной информации по компоненту – help

Команда help имеет следующий синтаксис:

```
pg_probackup help [command]
```

В транскрипции команда имеет следующий вид:

```
pg_probackup help [команда]
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup help
pg_probackup - utility to manage backup/recovery of PostgreSQL database.

pg_probackup help [COMMAND]

pg_probackup version

pg_probackup init -B backup-path

pg_probackup set-config -B backup-path --instance=instance_name
    [-D pgdata-path]
    [--external-dirs=external-directories-paths]
    [--log-level-console=log-level-console]
    [--log-level-file=log-level-file]
    [--log-filename=log-filename]
    [--error-log-filename=error-log-filename]
    [--log-directory=log-directory]
    [--log-rotation-size=log-rotation-size]
    [--log-rotation-age=log-rotation-age]
    [--retention-redundancy=retention-redundancy]
    [--retention-window=retention-window]
    [--wal-depth=wal-depth]
    [--compress-algorithm=compress-algorithm]
    [--compress-level=compress-level]
    [--archive-timeout=timeout]
    [-d dbname] [-h host] [-p port] [-U username]
    [--remote-proto] [--remote-host]
    [--remote-port] [--remote-path] [--remote-user]
    [--ssh-options]
    [--restore-command=cmdline] [--archive-host=destination]
    [--archive-port=port] [--archive-user=username]
    [--help]
```

Рисунок 10.1 – Вызов справки

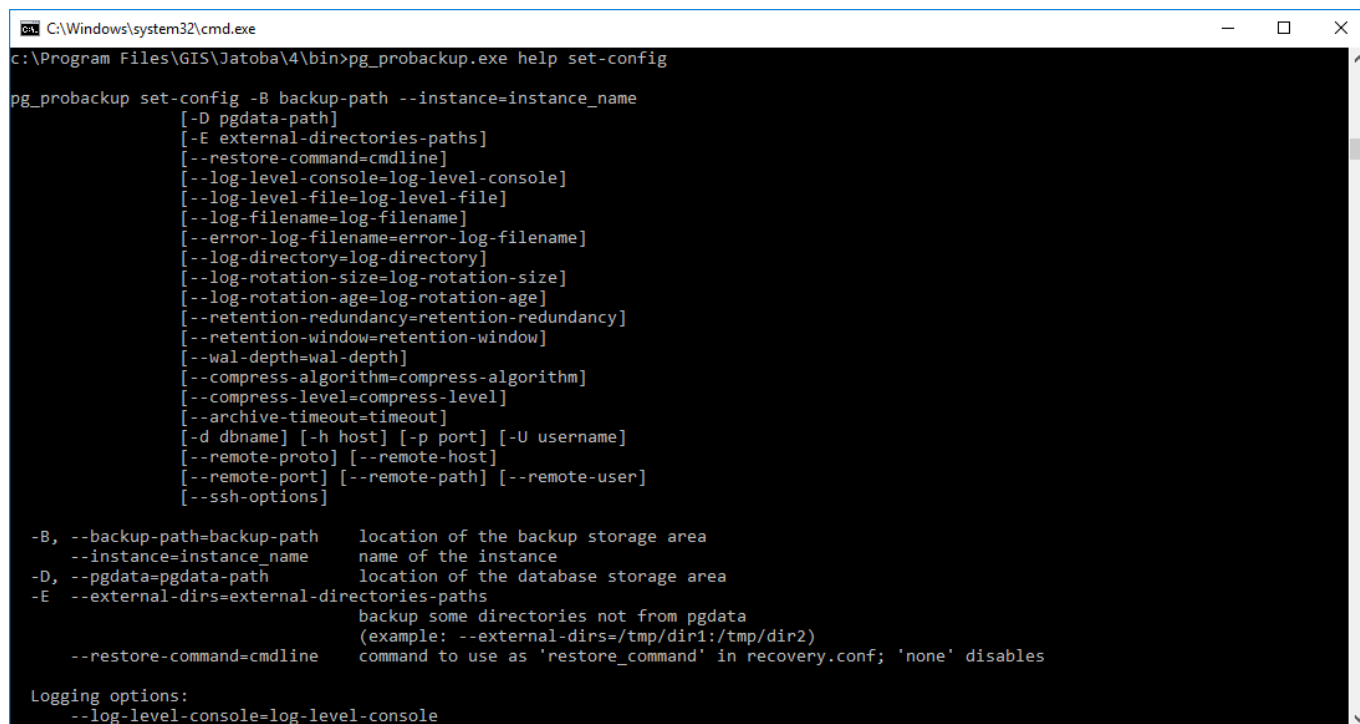
В результате выполнения команды выводится справка по командам pg\_ProBackup.

Если в параметрах команды help задается одна из команд pg\_ProBackup, то выводится подробная информация по параметрам указанной команды.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Командная строка для вызова справки определенной команды будет иметь следующий вид:

```
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe help set-  
config
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe help set-config  
  
pg_probackup set-config -B backup-path --instance=instance_name  
    [-D pgdata-path]  
    [-E external-directories-paths]  
    [--restore-command=cmdline]  
    [--log-level-console=log-level-console]  
    [--log-level-file=log-level-file]  
    [--log-filename=log-filename]  
    [--error-log-filename=error-log-filename]  
    [--log-directory=log-directory]  
    [--log-rotation-size=log-rotation-size]  
    [--log-rotation-age=log-rotation-age]  
    [--retention-redundancy=retention-redundancy]  
    [--retention-window=retention-window]  
    [--wal-depth=wal-depth]  
    [--compress-algorithm=compress-algorithm]  
    [--compress-level=compress-level]  
    [--archive-timeout=timeout]  
    [-d dbname] [-h host] [-p port] [-U username]  
    [--remote-proto] [--remote-host]  
    [--remote-port] [--remote-path] [--remote-user]  
    [--ssh-options]  
  
-B, --backup-path=backup-path    location of the backup storage area  
--instance=instance_name        name of the instance  
-D, --pgdata=pgdata-path        location of the database storage area  
-E --external-dirs=external-directories-paths  
                                backup some directories not from pgdata  
                                (example: --external-dirs=/tmp/dir1:/tmp/dir2)  
--restore-command=cmdline        command to use as 'restore_command' in recovery.conf; 'none' disables  
  
Logging options:  
--log-level-console=log-level-console
```

Рисунок 10.2 – Вызов справки о команде

### 10.1.3. Инициализация каталога резервных копий – init

Команда init описана подразделе 4.3.

### 10.1.4. Определение копируемого экземпляра – add-instance

Команда add-instance описана в подразделе 4.4.

### 10.1.5. Удаление резервных копий – del-instance

Команда del-instance имеет следующий синтаксис:

```
pg_probackup del-instance -B backup_dir --instance  
instance_name [--help]
```

В транскрипции команда имеет следующий вид:

```
pg_probackup del-instance -B каталог_копий --instance  
имя_экземпляра [--help]
```

При выполнении команды `del-instance` удаляются все резервные копии и файлы WAL, связанные с указанным экземпляром.

#### 10.1.6. Редактирование конфигурационного файла `pg_probackup.conf` – `set-config`

Команда `set-config` предназначена для установки параметров в конфигурационном файле `pg_probackup.conf` и имеет следующий синтаксис:

```
pg_probackup set-config -B backup_dir --instance instance_name  
[--help] [--pgdata = pgdata-path] [--retention-redundancy =  
redundancy] [-retention-window = window] [-wal-depth =  
wal_depth] [--compress-algorithm = compression_algorithm] [--  
compress-level = compression_level] [-d dbname] [-h host] [-p  
port] [-U username] [--archive-timeout = timeout] [--external-  
dirs = external_directory_path] [--restore-command = cmdline]  
[logging_options]
```

Параметры конфигурационного файла описаны в подразделах 10.1.6.1 – 10.1.6.10.

##### 10.1.6.1 Параметры сохранения – `retention`

Параметры `retention` могут использоваться с командами `backup` и `delete`, которые дополнительно описаны в подразделе 12.1.2. В данном подразделе описываются только те параметры, которые сохраняются в конфигурационном файле `pg_probackup.conf`.

Подробнее о политике хранения описано в подразделе 5.3.

```
--retention-redundancy=redundancy
```

В данном параметре указывается сколько полных резервных копий должно сохраняться в каталоге данных. Для сохранения полных копий, в параметре должно быть указано неотрицательное целое число. Значение «ноль» в параметре означает, что сохранение не выполняется (значение «0» установлено по умолчанию).

```
--retention-window=window
```

В параметре указывается период времени (в днях), в течение которого возможно восстановление. В параметре должно быть указано неотрицательное целое число. При нулевом значении окно восстановления отсутствует (значение «0» установлено по умолчанию).

### 10.1.6.2 Количество копий на линии времени – wal-depth

```
--wal-depth=wal_depth
```

В параметре указывается количество резервных копий на каждой линии времени, которое должно сохраняться для обеспечения возможности восстановления PITR. В параметре должно быть указано неотрицательное целое число. При нулевом значении этот параметр отключается (значение «0» установлено по умолчанию).

### 10.1.6.3 Алгоритм сжатия – compress-algorithm

Параметр `compression_algorithm` определяет алгоритм, который будет использоваться для сжатия файлов данных. Возможные значения:

- `zlib`;
- `pglz`;
- `none`.

Значение `zlib` или `pglz` включает сжатие. По умолчанию сжатие отключено (`none`).

```
[--compress-algorithm = compression_algorithm]
```

Пример:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir -D "c:\Program Files\GIS\Jatoba\4\data" --instance  
localdb --compress-algorithm=none
```

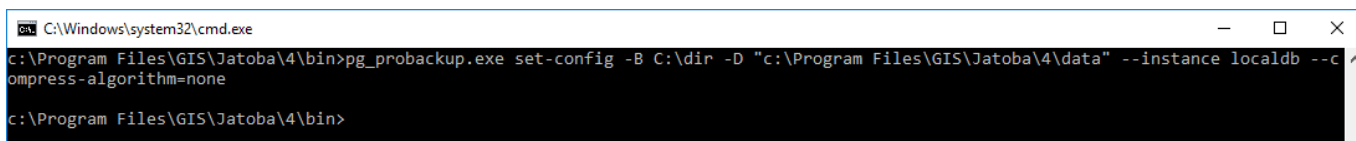


Рисунок 10.3 – Пример установки алгоритма сжатия

### 10.1.6.4 Уровень сжатия – compress-level

Параметр `compression_level` определяет уровень сжатия (от 0 до 9, где 0 обозначает отсутствие сжатия, а 9 – наилучшее сжатие). Этот параметр может использоваться только вместе с параметром `--compress-algorithm`. По умолчанию параметр `compression_level` принимает значение «1».

```
[--compress-level = compression_level]
```

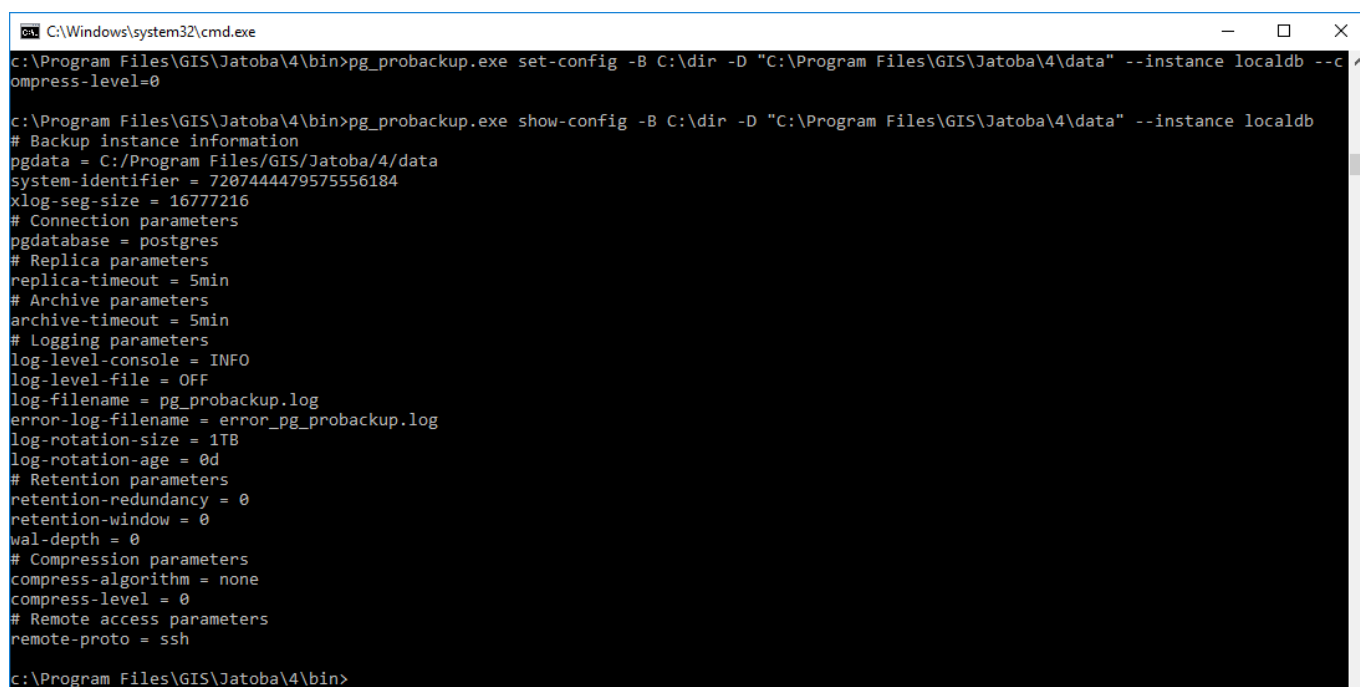


Пример:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb --compress-level=0
```

Альтернативный вариант одновременного указания `--compress-algorithm=zlib` и `--compress-level=1`.

Пример изменения параметров сжатия и просмотр полученного результата, представлен на рисунке 10.4.



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --compress-level=0  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb  
# Backup instance information  
pgdata = C:/Program Files/GIS/Jatoba/4/data  
system-identifier = 7207444479575556184  
xlog-seg-size = 16777216  
# Connection parameters  
pgdatabase = postgres  
# Replica parameters  
replica-timeout = 5min  
# Archive parameters  
archive-timeout = 5min  
# Logging parameters  
log-level-console = INFO  
log-level-file = OFF  
log-filename = pg_probackup.log  
error-log-filename = error_pg_probackup.log  
log-rotation-size = 1TB  
log-rotation-age = 0d  
# Retention parameters  
retention-redundancy = 0  
retention-window = 0  
wal-depth = 0  
# Compression parameters  
compress-algorithm = none  
compress-level = 0  
# Remote access parameters  
remote-proto = ssh  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.4 – Пример изменения параметров сжатия

#### 10.1.6.5 Уровень сообщений – log-level-file

Параметр `log-level-file` управляет уровнем сообщений, которые будут выводиться в файл журнала. Допустимые уровни:

- Verbose;
- Log;
- Info;
- Warning;
- Error;

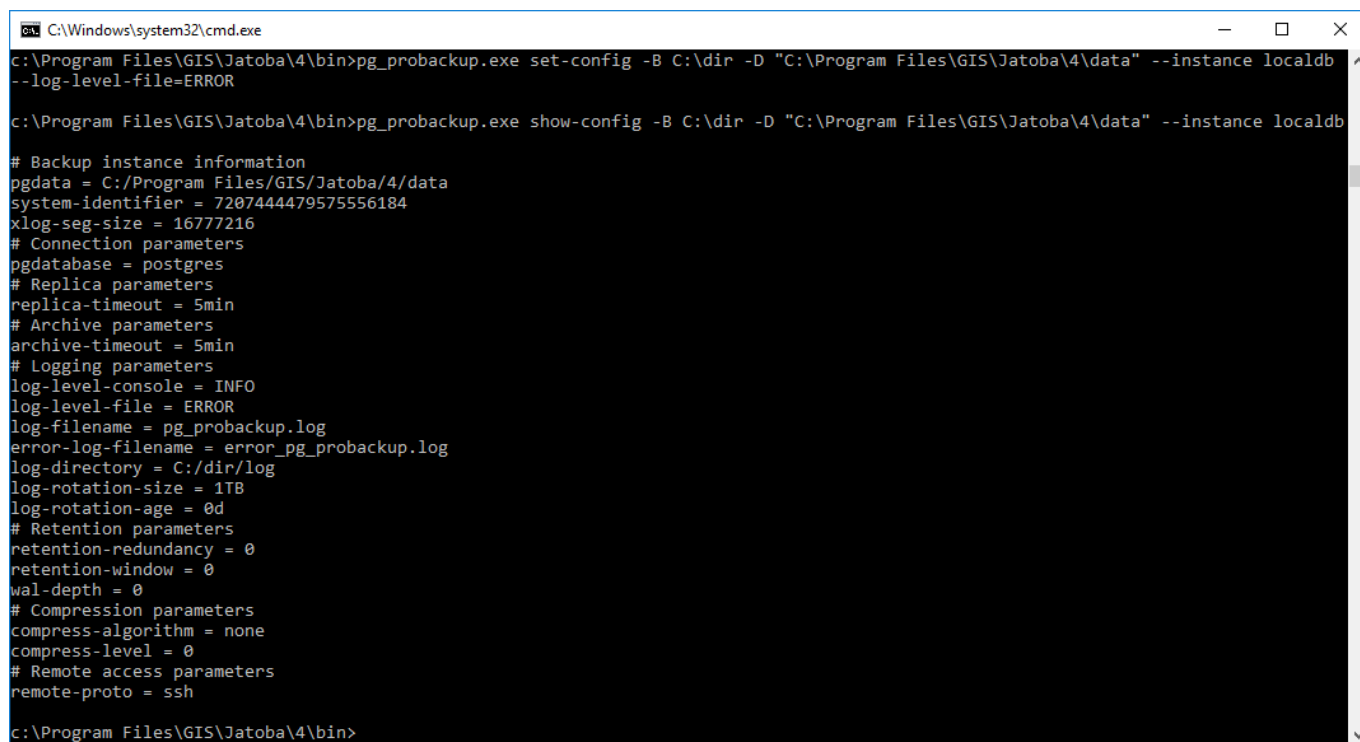
– Off.

Каждый уровень включает все последующие, и с каждым последующим уровнем объем сообщений уменьшается. Вариант off отключает вывод в файл журнала (значение «off» установлено по умолчанию).

```
--log-level-file=уровень_сообщений
```

Пример:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --log-level-file=ERROR
```



```
C:\Windows\system32\cmd.exe
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --log-level-file=ERROR
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb

# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 720744479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = postgres
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = ERROR
log-filename = pg_probackup.log
error-log-filename = error_pg_probackup.log
log-directory = C:/dir/log
log-rotation-size = 1TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 0
# Remote access parameters
remote-proto = ssh

C:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.5 – Пример установки параметра log-level-file и просмотра конфигурационного файла

#### 10.1.6.6 Имя файла журнала сообщений – log-filename

Параметр log-filename определяет имена файлов созданных файлов журнала.

```
--log-filename=файл_журнала
```

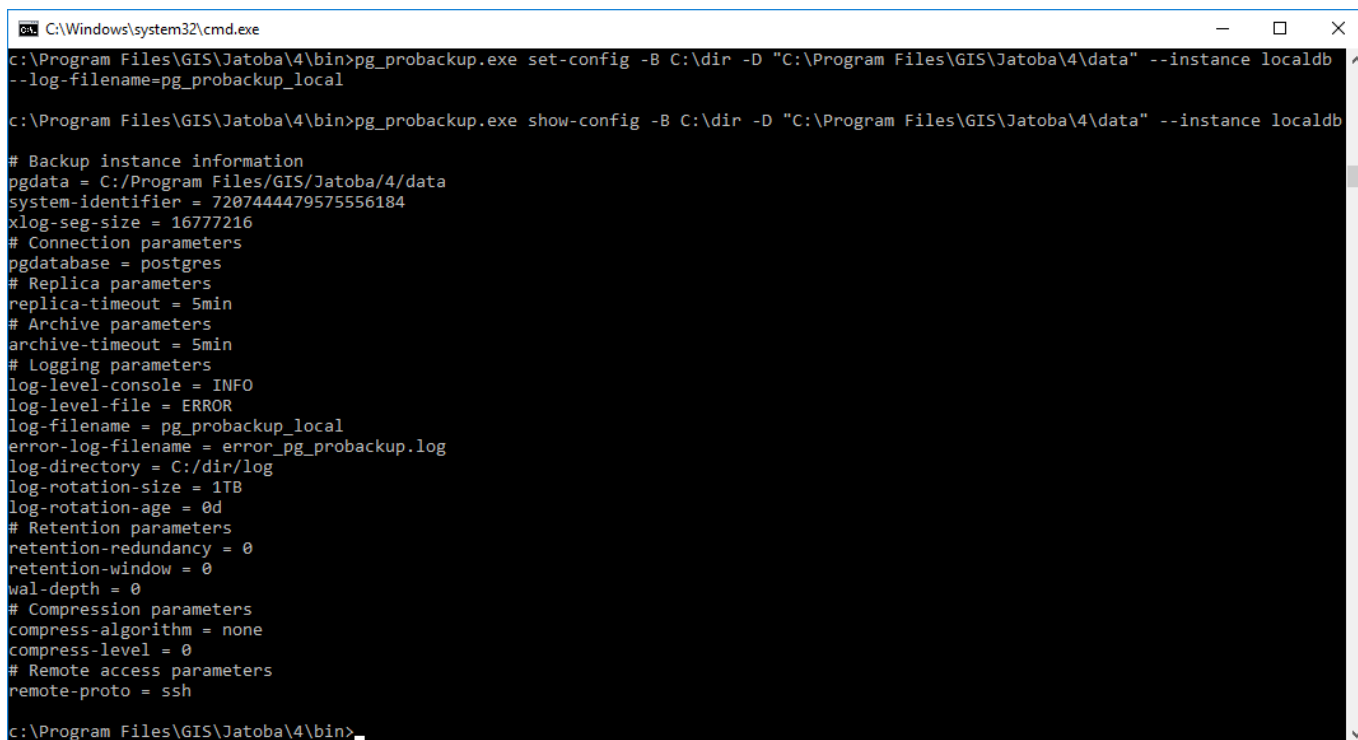
По умолчанию параметр log-filename принимает значение «pg\_probackup.log».

Пример:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Для изменения имен файлов при локальной установке (к имени файлов журнала добавляется слово local) командная строка будет иметь следующий вид:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb --log-filename=pg_probackup_local
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --log-filename=pg_probackup_local  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb  
# Backup instance information  
pgdata = C:/Program Files/GIS/Jatoba/4/data  
system-identifier = 7207444479575556184  
xlog-seg-size = 16777216  
# Connection parameters  
pgdatabase = postgres  
# Replica parameters  
replica-timeout = 5min  
# Archive parameters  
archive-timeout = 5min  
# Logging parameters  
log-level-console = INFO  
log-level-file = ERROR  
log-filename = pg_probackup_local  
error-log-filename = error_pg_probackup.log  
log-directory = C:/dir/log  
log-rotation-size = 1TB  
log-rotation-age = 0d  
# Retention parameters  
retention-redundancy = 0  
retention-window = 0  
wal-depth = 0  
# Compression parameters  
compress-algorithm = none  
compress-level = 0  
# Remote access parameters  
remote-proto = ssh  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.6 – Пример задания имени файла журнала и просмотра результата

```
--log-filename=файл_журнала-%u
```

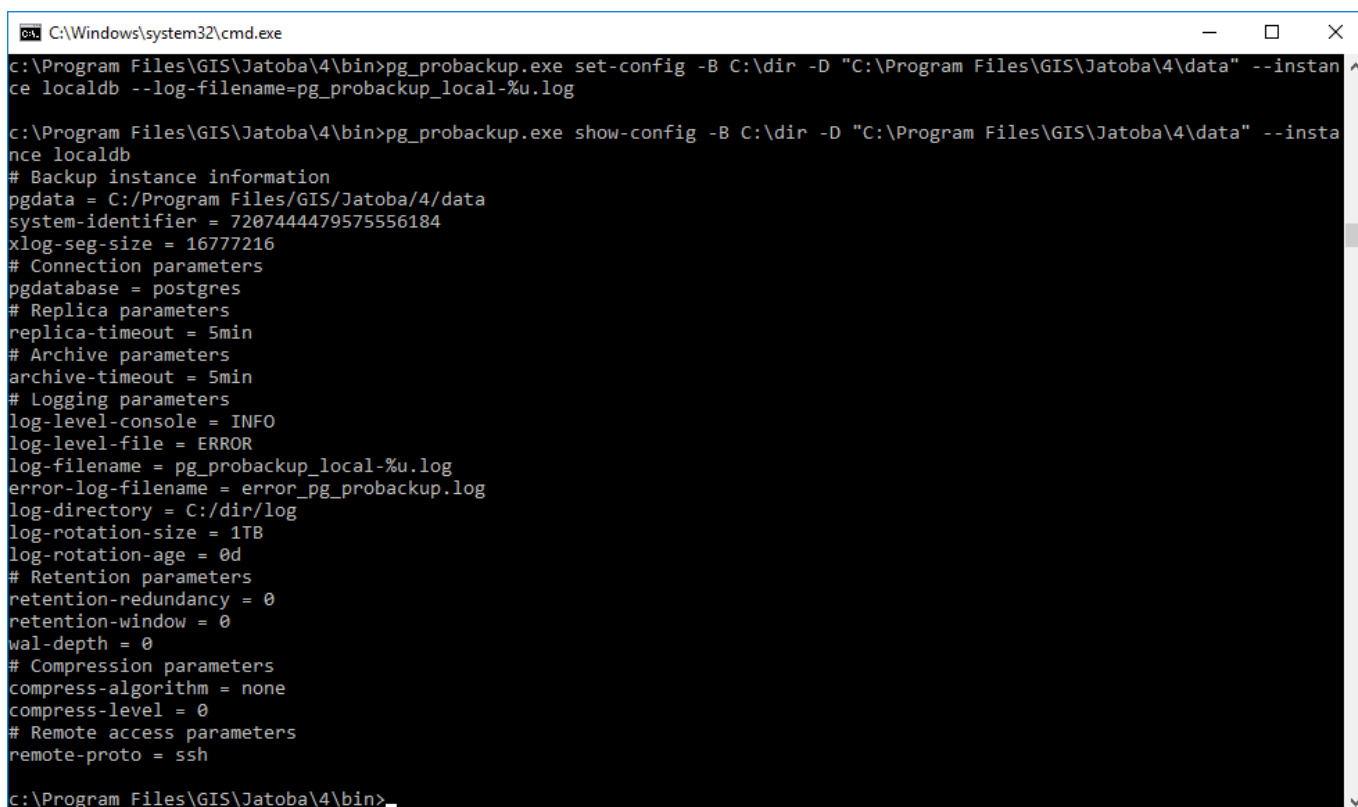
В имена файлов журнала можно добавить префикс дня недели. При этом день недели будет маркироваться по порядку:

- понедельник – первый день недели, файл будет сформирован с именем pg\_probackup\_local-1.log;
- вторник – второй день недели, файл будет сформирован с именем pg\_probackup\_local-2.log и т.д. соответственно.

В этом случае командная строка будет выглядеть следующим образом:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb --log-filename=pg_probackup_local-%u.log
```

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --log-filename=pg_probackup_local-%u.log

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb
# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 7207444479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = postgres
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = ERROR
log-filename = pg_probackup_local-%u.log
error-log-filename = error_pg_probackup.log
log-directory = C:/dir/log
log-rotation-size = 1TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 0
# Remote access parameters
remote-proto = ssh

c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.7 – Пример добавления префикса дня недели в имя файла журнала

Параметр `log-filename` действует, если включена запись в журнал (параметр `--log-level-file`).

#### 10.1.6.7 Имя файла журнала сообщений – `error-log-filename`

Параметр «`error-log-filename`» определяет имена только для файлов журналов ошибок. По умолчанию установлено значение «`none`».

```
--error-log-filename=файл_журнала_событий_ошибок
```

Для корректировки параметра `error-log-filename` в параметре `log-level-file` – уровень сообщений, должно быть установлено значение не ниже `ERROR`.

В этом случае командная строка будет выглядеть:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --error-log-filename=error_pg_probackup.log
```

```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb
--error-log-filename=error_pg_probackup.log

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb

# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 7207444479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = postgres
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = ERROR
log-filename = pg_probackup_local
error-log-filename = error_pg_probackup.log
log-directory = C:/dir/log
log-rotation-size = 1TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 0
# Remote access parameters
remote-proto = ssh

c:\Program Files\GIS\Jatoba\4\bin>_
  
```

Рисунок 10.8 – Пример установки и просмотра установленного параметра error-log-filename

#### 10.1.6.8 Каталог хранения файлов журнала сообщений – log-directory

--log-directory=каталог\_журнала

Параметр log-directory определяет каталог, в котором будут создаваться файлы журналов. В этом параметре должен быть задан абсолютный путь. Этот каталог создается только при необходимости, когда в журнал выводится первое сообщение.

По умолчанию: \$BACKUP\_PATH/log/

#### 10.1.6.9 Формат журнала консоли – log-format-console

--log-format-console=каталог\_журнала

Параметр log-format-console определяет формат журнала консоли. Устанавливается из командной строки, а не путем указания в конфигурационном файле pg\_probackup.conf. Возможные принимаемые значения:

- plain – вывод в формате обычного текста;

- json – вывод в формате json<sup>2</sup>.

По умолчанию: plain

#### 10.1.6.10 Размер журнала для ротации – log-rotation-size

```
--log-rotation-size=размер_журнала_для_ротации
```

Параметр log-rotation-size определяет максимальный размер отдельного файла журнала. Если заданное максимальное значение достигается, файл журнала перезаписывается при выполнении какой-либо команды pg\_ProBackup, за исключением help и version. Нулевое значение отключает ротацию в зависимости от размера файла. Поддерживаются следующие единицы:

- KB (по умолчанию);
- MB;
- GB;
- TB.

По умолчанию устанавливается значение: 0TB.

Например, требуется установить размер файла журнала 1TB. Командная строка будет следующей:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb --log-rotation-size=1TB
```

---

<sup>2</sup> поддерживается для версии компонента 2.5.12

```

C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb
--log-rotation-size=1TB

c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb

# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 7207444479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = postgres
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = ERROR
log-filename = pg_probackup_local
error-log-filename = error_pg_probackup.log
log-directory = C:/dir/log
log-rotation-size = 1TB
log-rotation-age = 0d
# Retention parameters
retention-redundancy = 0
retention-window = 0
wal-depth = 0
# Compression parameters
compress-algorithm = none
compress-level = 0
# Remote access parameters
remote-proto = ssh

c:\Program Files\GIS\Jatoba\4\bin>_

```

Рисунок 10.9 – Пример установки и просмотра установленного параметра log-rotation-size

#### 10.1.6.11 Возраст журнала для ротации – log-rotation-age

--log-rotation-age=возраст\_журнала\_для\_ротации

Параметр log-rotation-age устанавливает максимальное время жизни отдельного файла журнала. Если заданное максимальное время жизни достигается, файл журнала перезаписывается при выполнении какой-либо команды pg\_ProBackup, за исключением help и version.

Время создания последнего файла журнала сохраняется в каталоге \$BACKUP\_PATH/log/log\_rotation.

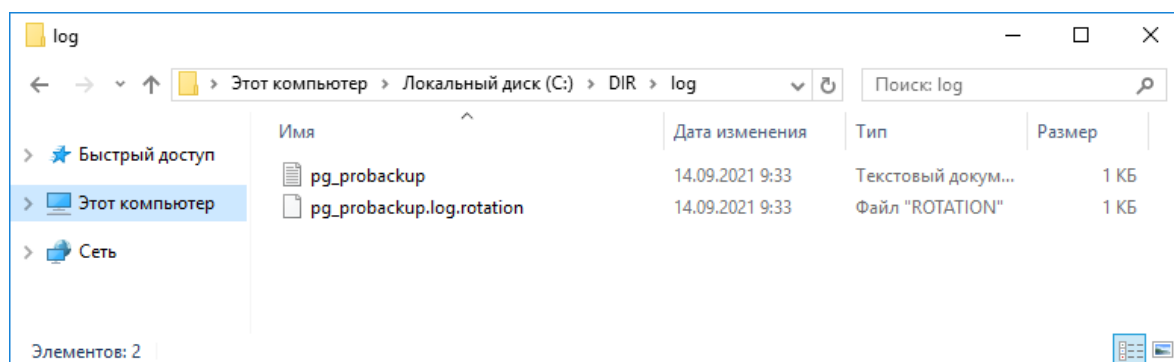


Рисунок 10.10 – Расположение файла ротации файлов журнала log\_rotation

Нулевое значение отключает ротацию файлов по времени.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

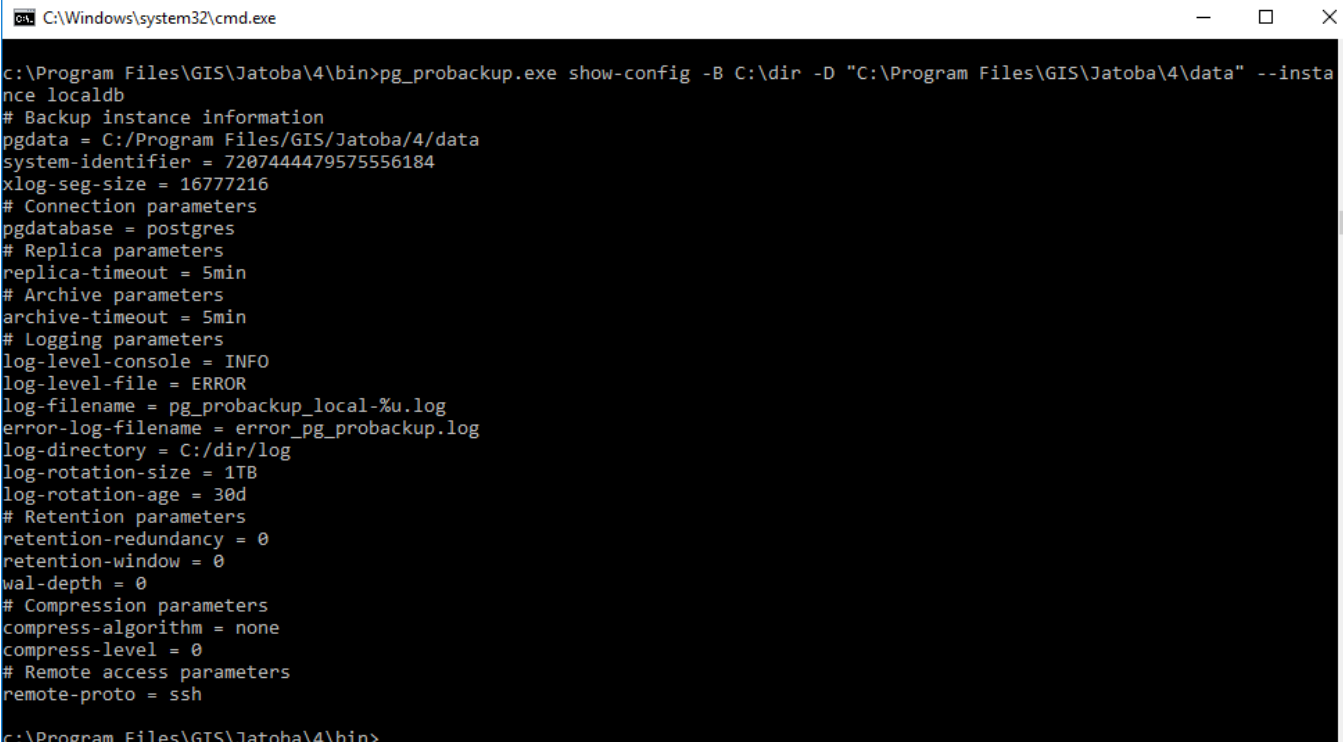
Параметр log-rotation-age поддерживает единицы:

- ms (миллисекунды),
- s (секунды),
- min (минуты, по умолчанию),
- h (часы),
- d (дни).

По умолчанию устанавливается значение: 0d.

Например, требуется установить ротацию файлов журнала через 30 дней. Командная строка будет следующей:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe set-config -  
B C:\dir --instance localdb --log-rotation-age=30d
```



```
C:\Windows\system32\cmd.exe  
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb  
# Backup instance information  
pgdata = C:/Program Files/GIS/Jatoba/4/data  
system-identifier = 7207444479575556184  
xlog-seg-size = 16777216  
# Connection parameters  
pgdatabase = postgres  
# Replica parameters  
replica-timeout = 5min  
# Archive parameters  
archive-timeout = 5min  
# Logging parameters  
log-level-console = INFO  
log-level-file = ERROR  
log-filename = pg_probackup_local-%u.log  
error-log-filename = error_pg_probackup.log  
log-directory = C:/dir/log  
log-rotation-size = 1TB  
log-rotation-age = 30d  
# Retention parameters  
retention-redundancy = 0  
retention-window = 0  
wal-depth = 0  
# Compression parameters  
compress-algorithm = none  
compress-level = 0  
# Remote access parameters  
remote-proto = ssh  
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.11 – Пример установки и просмотра установленного параметра log-rotation-age

Все поддерживаемые параметры описываются в разделе 11.

### 10.1.7. Закрепление резервных копий – set-backup

Команда set-backup имеет следующий синтаксис:



```
pg_probackup set-backup -B backup_dir --instance instance_name  
-i backup_id [--ttl=ttl | --expire-time=time] [--  
note=backup_note] [--help]
```

В транскрипции команда имеет следующий вид:

```
pg_probackup set-backup -B каталог_копий --instance  
имя_экземпляра -i ид_резервной_копии [--ttl=время_жизни | --  
expire-time=время] [--note=заметка_к_копии] [--help]
```

Команда `set-backup` устанавливает заданные для конкретной резервной копии параметры в конфигурационном файле `backup.control` или изменяет ранее определенные значения.

```
--note=backup_note
```

Параметр `--note` задает текстовую заметку для резервной копии. Если заметка к копии содержит символы перевода строки, сохранена будет только подстрока до первого перевода строки. Максимальный размер заметки равен 1 КБ.

Все поддерживаемые варианты закрепления описываются в подразделе 12.1.3.

#### 10.1.8. Просмотр содержимого конфигурационного файла `pg_probackup.conf` – `show-config`

Команда `show-config` выводит содержимое файла `pg_probackup.conf`, размещенного в каталоге резервной копии, и имеет следующий синтаксис:

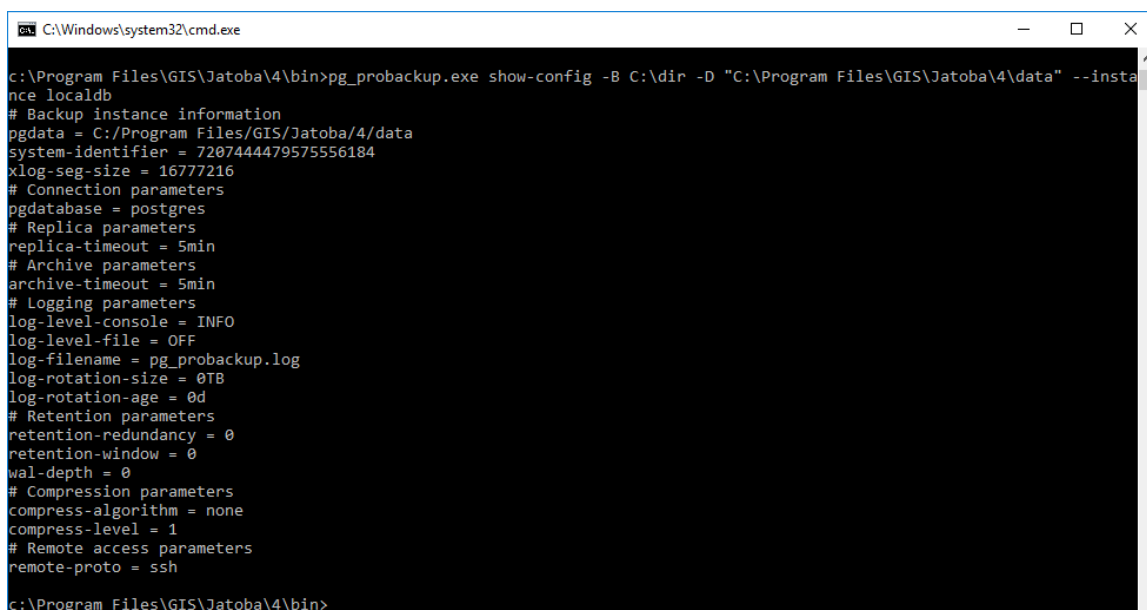
```
pg_probackup show-config -B backup_dir --instance instance_name  
[--format=plain|json]
```

В транскрипции команда имеет следующий вид:

```
pg_probackup show-config -B каталог_копий --instance  
имя_экземпляра [--format=plain|json]
```

В командной строке, от имени административной учетной записи ОС, выполняется команда:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config  
-B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance  
localdb
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb
# Backup instance information
pgdata = C:/Program Files/GIS/Jatoba/4/data
system-identifier = 7207444479575556184
xlog-seg-size = 16777216
# Connection parameters
pgdatabase = postgres
# Replica parameters
replica-timeout = 5min
# Archive parameters
archive-timeout = 5min
# Logging parameters
log-level-console = INFO
log-level-file = OFF
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
c:\Program Files\GIS\Jatoba\4\bin>
```

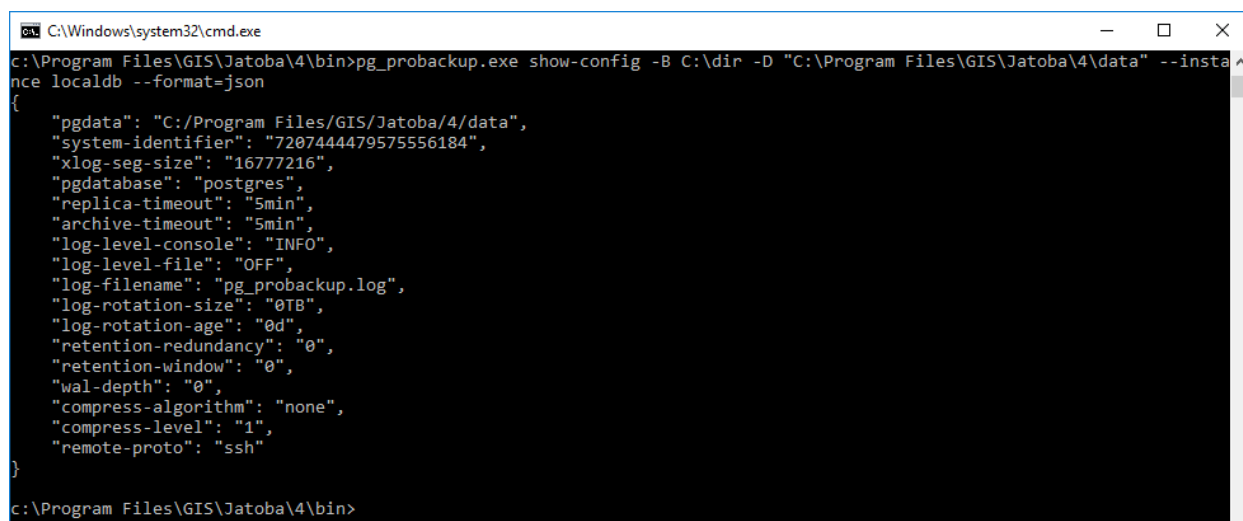
Рисунок 10.12 – Просмотр содержимого конфигурационного файла pg\_probackup.conf

Вывод содержания конфигурационного файла pg\_probackup.conf будет осуществлен по умолчанию в формате plain, если не добавлен параметр --format.

Если требуется просмотреть содержимое конфигурационного файла pg\_probackup.conf в формате JSON, в командную строку добавляется параметр --format=json.

Командная строка приобретет вид:

```
C:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config
-B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance
localdb --format=json
```



```
C:\Windows\system32\cmd.exe
c:\Program Files\GIS\Jatoba\4\bin>pg_probackup.exe show-config -B C:\dir -D "C:\Program Files\GIS\Jatoba\4\data" --instance localdb --format=json
{
  "pgdata": "C:/Program Files/GIS/Jatoba/4/data",
  "system-identifier": "7207444479575556184",
  "xlog-seg-size": "16777216",
  "pgdatabase": "postgres",
  "replica-timeout": "5min",
  "archive-timeout": "5min",
  "log-level-console": "INFO",
  "log-level-file": "OFF",
  "log-filename": "pg_probackup.log",
  "log-rotation-size": "0TB",
  "log-rotation-age": "0d",
  "retention-redundancy": "0",
  "retention-window": "0",
  "wal-depth": "0",
  "compress-algorithm": "none",
  "compress-level": "1",
  "remote-proto": "ssh"
}
c:\Program Files\GIS\Jatoba\4\bin>
```

Рисунок 10.13 – Просмотр содержимого конфигурационного файла pg\_probackup.conf в формате json

Для редактирования pg\_probackup.conf используется команда set-config.

Для изменения содержимого pg\_probackup.conf нужно использовать команду set-config.

### 10.1.9. Просмотр содержания каталога копий – show

Команда show описана в подразделах: 5.2.1 и 5.2.2.

### 10.1.10. Резервное копирование – backup

Команда backup имеет синтаксис:

```
pg_probackup backup -B backup_dir -b backup_mode --instance
instance_name [--help] [-j num_threads] [--progress] [-C] [--
stream [-S slot_name] [--temp-slot]] [--backup-pg-log] [--no-
validate] [--skip-block-validation] [--archive-timeout=timeout]
[--external-dirs=external_directory_path] [--no-sync] [--
note=backup_note] [connection_options] [compression_options]
[retention_options] [pinning_options] [logging_options]
```

и подробно описана в разделе 7.

Дополнительно задаются параметры:

- 12.2 «Параметры подключения»;
- 12.1.2 «Параметры сохранения – retention»;
- 12.1.3 «Параметры закрепления резервных копий – set-backup»;
- 12.3 «Параметры архивации»;
- 12.1.4 «Параметры ведения журнала – log»;
- 12.1 «Общие параметры».

### 10.1.11. Восстановление – restore

Команда restore имеет следующий синтаксис:

```
pg_probackup restore -B backup_dir --instance instance_name [--
help] [-D data_dir] [-i backup_id] [-j num_threads] [--
progress] [-T OLDDIR=NEWDIR] [--external-mapping=OLDDIR=NEWDIR]
[--skip-external-dirs] [-R | --restore-as-replica] [--no-
validate] [--skip-block-validation] [--force] [--no-sync] [--
restore-command=cmdline] [--primary-conninfo=primary_conninfo]
[-S | --primary-slot-name=slot_name] [-X wal_dir | --
```

```
waldir=wal_dir] [recovery_target_options] [logging_options]  
[remote_options] [partial_restore_options]  
[remote_wal_archive_options]
```

Команда `restore` восстанавливает экземпляр СУБД из резервной копии, расположенной в каталоге копий.

Если указать один или несколько параметров точки восстановления (п. 12.1.1), `pg_ProBackup` найдет ближайшую к этой точке копию и выполнит восстановление до заданной точки.

Если ни идентификатор копии, ни параметры точки восстановления не задаются, `pg_ProBackup` выполняет восстановление, используя самую последнюю копию.

```
-S  
--primary-slot-name=slot_name
```

Устанавливает заданное значение для параметра `primary_slot_name`. Это значение учитывается только при использовании параметра `-R`.

```
-T OLDDIR=NEWDIR  
--tablespace-mapping=OLDDIR=NEWDIR
```

Параметр `--tablespace-mapping` перемещает табличное пространство из старого каталога (OLDDIR) в новый (NEWDIR) во время восстановления. Оба каталога должны задаваться абсолютными путями. Если путь содержит знак «равно (=)», нужно экранировать этот знак обратной косой чертой. Данный параметр может указываться неоднократно для перемещения нескольких табличных пространств.

```
--external-mapping=OLDDIR=NEWDIR
```

Параметр `--external-mapping` перемещает внешний каталог, включенный в резервную копию, из старого каталога (OLDDIR) в новый каталог (NEWDIR) во время восстановления. Оба каталога должны задаваться абсолютными путями. Если путь содержит знак «равно (=)», нужно экранировать этот знак обратной косой чертой. Данный параметр может указываться неоднократно для нескольких каталогов.

```
--skip-external-dirs
```

Параметр `--skip-external-dirs` выполняет пропуск внешних каталогов, включенных в резервную копию. Содержимое этих каталогов не будет восстановлено.

```
--skip-block-validation
```

Параметр `--skip-block-validation` отключает проверку контрольных сумм на уровне блоков для ускорения проверки целостности. При автоматической проверке перед восстановлением будут проверяться только контрольные суммы на уровне файлов.

```
--no-validate
```

Параметр `--no-validate` пропускает проверку резервного копирования. Используется для сокращения времени восстановления данных, если копии проверяются регулярно.

```
--restore-command=cmdline
```

Задаёт значение для параметра `restore_command`.

Например: `--restore-command='cp /mnt/server/archivedir/%f "%p"'`

```
--force
```

Параметр `--force` позволяет игнорировать ошибочное состояние копии. Этот параметр можно использовать, когда требуется восстановить данные СУБД из поврежденной или некорректной копии. Если в целевом каталоге PGDATA идентификатор системы отличается от указанного в копии, при инкрементальном восстановлении с этим параметром содержимое каталога будет перезаписано (тогда как без параметра возникнет ошибка). Также в случае перенаправления указанием `--tablespace-mapping` табличных пространств в непустые каталоги содержимое этих каталогов будет удалено.

```
--no-sync
```

Параметр `--no-sync` позволяет не синхронизировать восстанавливаемые файлы на диск, что ускоряет процесс восстановления.



Использование этого параметра может привести к повреждению данных в случае аварии операционной системы или аппаратного сбоя. В этом случае потребуется запустить команду restore повторно.

```
-X wal_dir  
--waldir=wal_dir
```

Параметр указывает каталог, в котором должен храниться архив WAL<sup>3</sup>.

Дополнительно можно задать параметры точки восстановления, удаленного сервера, удаленного архива WAL, ведения журнала и частичного восстановления, а также общие параметры.

#### 10.1.12. Проверка целостности директории данных СУБД – checkdb

Команда checkdb проверяет целостность директории данных СУБД, выявляя физические и логические повреждения и имеет следующий синтаксис:

```
pg_probackup checkdb [-B backup_dir] [--instance instance_name]  
[-D data_dir] [--help] [-j num_threads] [--progress] [--skip-  
block-validation] [--amcheck] [--heapallindexed]  
[connection_options] [logging_options]
```

В транскрипции команда имеет вид:

```
pg_probackup checkdb [-B каталог_копий] [--instance  
имя_экземпляра] [-D каталог_данных] [--help] [-j число_потоков]  
[--progress] [--skip-block-validation] [--amcheck]  
[--heapallindexed] [параметры_соединения] [параметры_журнала]
```

Команда checkdb описана в подразделе 8.2.

#### 10.1.13. Объединение (слияние) резервных копий – merge

Команда merge имеет следующий синтаксис:

```
pg_probackup merge -B backup_dir--instance instance_name-i  
backup_id [--help] [-j num_threads] [--progress] [--no-  
validate] [--no-sync] [logging_options]
```

---

<sup>3</sup> поддерживается для версии компонента 2.5.12

Команда объединяет копии, относящиеся к одной цепочке инкрементальных копий:

- если выбрана полная копия, она будет объединена с первой инкрементальной копией после нее;
- если выбрана инкрементальная копия, она будет объединена с родительской полной копией, включая все инкрементальные копии между ними.

После завершения объединения результирующая полная копия будет вмещать в себя все данные, а инкрементальные копии будут удалены как избыточные.

```
--no-validate
```

Параметр `--no-validate` пропускает автоматическую проверку до и после слияния.

```
--no-sync
```

Параметр `--no-sync` позволяет не синхронизировать объединенные файлы на диск для ускорения процесса слияния.



Использование этого параметра может привести к повреждению данных в случае сбоя операционной системы или оборудования.

Примеры применения команды описаны в разделе 5.3.4.

#### 10.1.14. Удаление резервных копий – delete

Команда `delete` описана в подразделах: 5.3.1 и 5.3.5.

#### 10.1.15. Архивирование – archive-push

Команда `archive-push` имеет следующий синтаксис:

```
pg_probackup archive-push -B backup_dir --instance
instance_name --wal-file-name = wal_file_name [--wal-file-path
= wal_file_path] [--help] [--no-sync] [--compress] [--no-ready-
rename] [--overwrite] [-j num_threads] [--batch-size =
batch_size] [--archive-timeout = timeout] [--compress-algorithm
= compression_algorithm] [--compress-level = compression_level]
[logging_options]
```

В транскрипции команда имеет вид:

```
pg_probackup archive-push -В каталог_копий --instance
имя_экземпляра --wal-file-name=имя_файла_wal [--wal-file-
path=путь_файлов_wal] [--help] [--no-sync] [--compress] [--no-
ready-rename] [--overwrite] [-j число_потоков] [--batch-
size=размер_порции] [--archive-timeout=тайм-аут] [--compress-
algorithm=алгоритм_сжатия] [--compress-level=уровень_сжатия]
[параметры_журнала]
```

Команда копирует файлы WAL в соответствующий подкаталог каталога копий, проверяя целевой экземпляр по имени экземпляра (*instance\_name*) и значению системного идентификатора (*system-identifier*).

Если параметры экземпляра резервной копии и исходной директории данных не совпадают, операция копирования не выполняется, и выдается ошибка: *Refuse to push WAL segment segment\_name into archive. Instance parameters mismatch.* (Отказано в помещении сегмента %имя\_сегмента% в архив. Параметры экземпляра не совпадают.)

Если копируемые файлы уже имеются в каталоге копий, pg\_ProBackup вычисляет и сравнивает их контрольные суммы. В случае совпадения контрольных сумм archive-push пропускает соответствующий файл и выдает код успешного завершения. Если же они не совпадают, операция archive-push завершается с ошибкой. При замене существующих файлов WAL в случае несовпадения контрольных сумм, требуется добавить к команде archive-push ключ --overwrite.

Содержимое каждого файла копируется во временный файл с расширением part. Если такой временный файл уже существует, pg\_probackup выполняет проверку наличия файла в течение заданного параметром archive\_timeout времени, и если этого не происходит, не использует его.

После переноса содержимого выполняется атомарная операция переименования. Тем самым гарантируется, что в случае ошибки команды archive-push непрерывное архивирование не остановится и при параллельном архивировании WAL из разных источников в один архив повреждение архива исключено.

Для ускорения архивирования можно воспользоваться параметром --batch-size, определяющим размер порции из нескольких копируемых сегментов WAL. Вместе с параметром --batch-size также можно применить параметр -j, чтобы порции копировались в несколько потоков.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



Сегменты WAL, копируемые в архив, по умолчанию (без указания параметра `--no-sync`) гарантированно сбрасываются на диск.

Команду `archive-push` можно использовать в значении параметра `archive_command` при настройке непрерывного архивирования WAL.

Дополнительные параметры описаны в подразделах:

- 12.3 «Параметры архивации»;
- 12.2.1 «Параметры сжатия – `compress`».

#### 10.1.16. Получение архива – `archive-get`

Команда `archive-get` имеет следующий синтаксис:

```
pg_probackup archive-get -B backup_dir --instance instance_name
--wal-file-path = wal_file_path --wal-file-name = wal_file_name
[-j num_threads] [--batch-size = batch_size] [--prefetch-dir =
prefetch_dir_path] [--no-validate-wal] [--help]
[logging_options]
```

В транскрипции команда имеет вид:

```
pg_probackup archive-get -B каталог_копий --instance
имя_экземпляра --wal-file-path=путь_файлов_wal --wal-file-
name=имя_файла_wal [-j число_потоков] [--batch-
size=размер_порции] [--prefetch-dir=каталог_предвыборки] [--no-
validate-wal] [--help] [параметры_журнала]
```

Команда копирует файлы WAL из соответствующего подкаталога каталога резервных копий в каталог журнала предзаписи директории данных. Эта команда автоматически устанавливается программой `pg_ProBackup` в значении параметра `restore_command` при восстановлении архивных копий с применением архива WAL. Устанавливать ее вручную не нужно.

Для ускорения восстановления можно воспользоваться параметром `--batch-size`, определяющим размер порции из нескольких копируемых сегментов WAL. Вместе с параметром `--batch-size` также можно применить параметр `-j`, чтобы порции сегментов копировались в несколько потоков.

Дополнительные параметры описаны в подразделе 12.3 «Параметры архивации».

## 11. ИСПОЛЬЗОВАНИЕ PG\_PROBACKUP В УДАЛЕННОМ РЕЖИМЕ

Программа pg\_ProBackup поддерживает работу в удаленном режиме, то есть может выполнять команды backup и restore удаленно, используя SSH. В этом режиме каталог резервных копий располагается в локальной системе, а целевой экземпляр СУБД работает в удаленной. При этом компонент pg\_ProBackup должен быть установлен в обеих системах.

Типичная схема использования pg\_ProBackup в удаленном режиме:

1) В системе резервного копирования настройте pg\_ProBackup, как описывается в подразделах «Установка компонента» и «Настройка pg\_ProBackup». Для команд add-instance и set-config необходимо задать параметры удаленного сервера, указывающие на сервер с экземпляром СУБД.

2) При необходимости выполнять страничное копирование (PAGE), использовать доставку WAL в режиме ARCHIVE или осуществлять восстановление PITR в удаленном режиме, настройте непрерывное архивирование WAL с сервера БД в систему резервного копирования, как описано в подразделе «Настройка непрерывного архивирования WAL». Для этого в командах archive-push и archive-get задайте параметры удаленного сервера, указывающие на сервер, где находится каталог резервных копий.

3) Запустите команду backup или restore с параметрами удаленного сервера в системе резервного копирования. pg\_ProBackup подключится к удаленной системе по SSH и создаст резервную копию в локальной системе или восстановит в удаленной системе ранее сделанную копию.

Например, для создания полной архивной копии кластера СУБД, работающего в удаленной системе с адресом 192.168.0.2, подключившись к серверу по SSH через порт 2302 с именем пользователя postgres, выполните:

```
pg_probackup backup -В каталог_копий --instance имя_экземпляра  
-b FULL --remote-user=postgres --remote-host=192.168.0.2 --  
remote-port=2302
```

Для восстановления последней имеющейся копии в удаленной системе с адресом 192.168.0.2, подключившись к серверу по SSH через порт 2302 с именем пользователя postgres, выполните:

```
pg_probackup restore -В каталог_копий --instance имя_экземпляра  
--remote-user=postgres --remote-host=192.168.0.2 --remote-  
port=2302
```

Для восстановления архивных копий или восстановления на момент времени в удаленном режиме требуется дополнительная информация, целевой адрес, порт и имя пользователя для установления SSH-подключения с сервера БД к системе, где находится каталог копий. Эту информацию будет использовать команда `restore_command` для копирования сегментов WAL из архива в каталог СУБД `pg_wal`.

Для передачи этой информации вы можете задать параметры удаленного архива WAL.

#### Пример

Для восстановления последней копии в удаленной системе, настроенной для приема подключения пользователя `postgres` по адресу `192.168.0.2` через порт `2302`, с использованием каталога резервных копий в системе, настроенной для подключения пользователя `backup` по адресу `192.168.0.3` через порт `2303`, выполните:

```
pg_probackup restore -В каталог_копий --instance имя_экземпляра  
--remote-user=postgres --remote-host=192.168.0.2 --remote-  
port=2302 --archive-host=192.168.0.3 --archive-port=2303 --  
archive-user=backup
```

Переданные аргументы будут использоваться при составлении команды `restore_command`:

```
restore_command = '"путь_инсталляции/pg_probackup" archive-get  
-В "каталог_копий" --instance имя_экземпляра --wal-file-path=%p  
--wal-file-name=%f --remote-host=192.168.0.3 --remote-port=2303  
--remote-user=backup'
```

Также можно воспользоваться ключом `--restore-command` и передать всю команду `restore_command` целиком:

```
pg_probackup restore -В каталог_копий --instance имя_экземпляра  
--remote-user=postgres --remote-host=192.168.0.2 --remote-  
port=2302 --restore-command='"путь_инсталляции/pg_probackup"  
archive-get -В "каталог_копий" --instance имя_экземпляра --wal-
```

```
file-path=%p --wal-file-name=%f --remote-host=192.168.0.3 --  
remote-port=2303 --remote-user=backup'
```

### 11.1. Параметры удаленного режима

В этом подразделе описываются параметры, связанные с удаленным выполнением операций `pg_probackup` по SSH. Эти параметры могут использоваться с командами `add-instance`, `set-config`, `backup`, `restore`, `archive-push` и `archive-get`.

```
--remote-proto=протокол
```

Параметр `--remote-proto` задает протокол для удаленного выполнения операций. В настоящее время поддерживается только протокол SSH. Возможные значения:

- `ssh` — включает удаленный режим с использованием SSH. Этот вариант используется по умолчанию.

- `none` — явным образом отключает удаленный режим.

Этот параметр можно не задавать, если указывается `--remote-host`.

```
--remote-host=целевой_адрес
```

Параметр `--remote-host` задает имя или IP-адрес целевого удаленного сервера.

```
--remote-port=порт
```

Параметр `--remote-port` задает целевой порт на удаленном сервере. По умолчанию: 22.

```
--remote-user=имя_пользователя
```

Параметр `--remote-user` задает имя пользователя на удаленном сервере для SSH-соединения. В отсутствие этого параметра используется имя текущего пользователя, устанавливающего SSH-соединения.

```
--remote-path=путь
```

Параметр `--remote-path` задает каталог, в котором `pg_probackup` установлен на удаленной системе.

```
--ssh-options=параметры_ssh
```

Параметр `--ssh-options` задает строку параметров командной строки для SSH.

#### Пример

Возможно установить свойства `keep-alive` для SSH-подключений, которые будет открывать `pg_probackup`: `--ssh-options="-o ServerAliveCountMax=5 -o ServerAliveInterval=60"`.

### 11.2. Настройка удаленного режима

Компонент `pg_ProBackup` поддерживает удаленный режим, в котором резервное копирование, восстановление и архивирование WAL могут выполняться удаленно. В этом режиме каталог резервных копий находится в локальной системе, а целевой экземпляр СУБД работает на удаленном компьютере.



В настоящее время поддерживается с использованием только одного сетевого протокола - SSH.

Если необходимо использовать компонент `pg_ProBackup` в удаленном режиме (применяя SSH), выполните следующие действия:

1) Установите компонент `pg_ProBackup` в обеих системах: `backup_host` (сервер резервного копирования) и `db_host` (сервер баз данных).

2) Для организации связи между узлами настройте SSH-соединение без пароля для подключения пользователя `backup` на компьютере `backup_host` к серверу `db_host` под именем `postgres`:

```
[backup@backup_host] ssh-copy-id postgres@db_host
```

3) Если будет использоваться непрерывное архивирование WAL, настройте подключение по SSH пользователя `backup` в системе `backup_host` к системе `db_host` под именем пользователя `postgres` без указания пароля:

```
[postgres@db_host] ssh-copy-id backup@backup_host
```

В данном примере:

*backup\_host* – система, в которой находится каталог копий;

*db\_host* – система, в которой функционирует СУБД;

*backup* – пользователь в системе *backup\_host*, от имени которого запускается компонент *pg\_ProBackup*;

*postgres* – пользователь в системе *db\_host*, от имени которого запускается СУБД.

Компонент *pg\_ProBackup* работает в удаленном режиме по протоколу SSH следующим образом:

- В удаленном режиме поддерживаются только команды: add-instance, backup, restore, catchup, archive-push и archive-get.
- Для работы в удаленном режиме необходимо, чтобы и в локальной, и в удаленной системе был установлен компонент *pg\_ProBackup* одной и той же версии.
- При запуске в удаленном режиме основной процесс *pg\_probackup* в локальной системе подключается к удаленной системе по протоколу SSH и запускает в удаленной системе один или несколько агентов, которые называются удаленными агентами. Количество удаленных агентов определяется значением параметра *-j/--threads*.
- Основной процесс *pg\_ProBackup* использует удаленные агенты для передачи данных между локальной и удаленной системой.
- Удаленные агенты стараются минимизировать сетевой трафик и количество операций взаимодействия узлов.
- Основной процесс обычно запускается в системе *backup\_host* и подключается к системе *db\_host*, но в случае команд archive-push и archive-get основной процесс запускается в системе *db\_host* и подключается к *backup\_host*.
- После завершения передачи данных удаленные агенты завершаются и подключения SSH закрываются.
- Если какой-либо удаленный агент столкнулся с ошибкой, все агенты завершаются, а основной процесс *pg\_probackup* выдает информацию о возникшей ошибке и также нештатно завершается.
- Сжатие данных всегда осуществляется в системе *db\_host*, а распаковка — в системе *backup\_host*.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

## 12. ПАРАМЕТРЫ КОМАНД PG\_PROBACKUP

В подразделе описываются параметры командной строки для команд pg\_ProBackup. Если какое-либо значение параметра может быть получено из переменной окружения, имя этой переменной указывается в верхнем регистре строкой ниже после указания параметра командной строки. Некоторые значения могут быть получены из файла конфигурации pg\_probackup.conf, находящегося в каталоге копий.

Дополнительные параметры описаны в подразделе 5 «Настройка pg\_probackup».



Если параметр задается несколькими способами, значение в командной строке имеет наивысший приоритет, а значение в pg\_probackup.conf – наименьший.

### 12.1. Общие параметры

В подразделе приведен список параметров общего характера.

```
-B directory  
--backup-path=directory  
BACKUP_PATH
```

Задает абсолютный путь к каталогу копий.



Чтобы не указывать путь каждый раз в командной строке, рекомендуется указать абсолютный путь один раз в переменной окружения BACKUP\_PATH, так как данное расположение необходимо задавать почти для всех команд pg\_ProBackup.

```
-D directory  
--pgdata=directory  
PGDATA
```

Задает абсолютный путь к каталогу данных. Этот параметр является обязательным только для команды add-instance. Другие команды могут получать этот путь из переменной окружения PGDATA или из файла конфигурации pg\_probackup.conf.

```
-i backup_id  
--backup-id=backup_id
```

Задаёт уникальный идентификатор резервной копии.

```
-j num_threads  
--threads=num_threads
```

Задаёт число параллельных потоков, запускаемых командами backup, restore, merge, validate, checkdb и archive-push.

```
--progress
```

Включает вывод прогресса выполнения операций.

```
--help
```

Выводит подробную информацию по параметрам, которые принимает команда.

### 12.1.1. Параметры точки восстановления – recovery-target

Если настроено непрерывное архивирование WAL, можно передать один из этих параметров с командой restore или validate, чтобы указать момент восстановления, до которого должен быть восстановлен или проверен сервер баз данных.

```
--recovery-target=immediate|latest
```

Определяет остановку восстановления:

- со значением immediate восстановление завершается сразу после достижения согласованного состояния выбранной копии, либо последней копии из имеющихся, если параметр -i/--backup-id не задан. Такое поведение по умолчанию применяется для копий типа STREAM;
- со значением latest восстановление продолжается до тех пор, пока не будут применены все имеющиеся в архиве сегменты WAL. Такое поведение по умолчанию применяется для копий типа ARCHIVE.

```
--recovery-target-timeline=timeline
```

Выбирает линию времени для восстановления. По умолчанию выбирается линия времени указанной резервной копии.



```
--recovery-target-lsn=lsn
```

Указывает до какого последовательного номера в журнале предзаписи должно производиться восстановление.

```
--recovery-target-name=recovery_target_name
```

Указывает именованную точку сохранения, вплоть до которой будет восстановлен сервер данных.

```
--recovery-target-time=time
```

Указывает точку времени, вплоть до которой будет производиться восстановление. Если часовой пояс не указывается, подразумевается местное время.

```
--recovery-target-xid=xid
```

Указывает идентификатор транзакции, вплоть до которой будет производиться восстановление.

```
--recovery-target-inclusive=boolean,
```

`boolean` –логическое значение, указывающее на необходимость остановки сразу после (`true`) либо до (`false`) достижения целевой точки. Этот параметр можно использовать только вместе с параметрами `--recovery-target-name`, `--recovery-target-time`, `--recovery-target-lsn` или `--recovery-target-xid`. Значение по умолчанию определяется параметром `recovery_target_inclusive`.

```
--recovery-target-action=pause|promote|shutdown
```

Задаёт действие (`recovery_target_action`), которое должен выполнить сервер по достижении цели восстановления.

По умолчанию: `pause`.

### 12.1.2. Параметры сохранения – `retention`

Эти параметры могут использоваться с командами `set-config`, `backup` и `delete`.

Подробнее о политике хранения описано в подразделе «Настройка политики хранения».

```
--retention-redundancy=redundancy
```

В параметре указывается сколько полных резервных копий должно сохраняться в каталоге данных. Для сохранения полных копий, в параметре должно быть указано неотрицательное целое число. Значение «ноль» в параметре означает, что сохранение не выполняется (значение «0» установлено по умолчанию).

```
--retention-window=window
```

В параметре указывается период времени (в днях), в течение которого возможно восстановление. В параметре должно быть указано неотрицательное целое число. При нулевом значении окно восстановления отсутствует (значение «0» установлено по умолчанию).

```
--wal-depth=wal_depth
```

В параметре указывается количество резервных копий на каждой линии времени, которое должно сохраняться для обеспечения возможности восстановления PITR. В параметре должно быть указано неотрицательное целое число. При нулевом значении этот параметр отключается (значение «0» установлено по умолчанию).

```
--delete-wal
```

Удаляет файлы WAL, которые не являются необходимыми для восстановления сервера из имеющихся резервных копий.

```
--delete-expired
```

Удаляет резервные копии, не удовлетворяющие политике сохранения, определенной в файле конфигурации pg\_probackup.conf.

```
--merge-expired
```

Объединяет самую старую инкрементальную копию, удовлетворяющую требованиям политики хранения, с ее родительскими копиями, срок хранения которых истек.

```
--dry-run
```

Выводит текущее состояние имеющихся резервных копий, при этом никакие операции, например, удаление или объединение старых копий не производятся.

### 12.1.3. Параметры закрепления резервных копий – **set-backup**

Эти параметры могут использоваться с командами `backup` и `set-backup`.

Подробная информация описана в подразделе 5.3.2.

```
--ttl=ttl
```

`ttl` – time to live – время жизни. Параметр задает время, на которое закрепляется резервная копия. Параметр должен принимать неотрицательное целое значение. Нулевое значение отменяет установленное ранее закрепление резервной копии. Поддерживаются следующие единицы измерения:

- `ms` (миллисекунды);
- `s` (секунды) (используется по умолчанию);
- `min` (минуты);
- `h` (часы);
- `d` (дни).

Например: `--ttl=30d`

```
--expire-time=time
```

Определяет момент времени до которого будет храниться резервная копия. Время должно задаваться в формате ISO-8601. Если часовой пояс не указывается, подразумевается местное время.

Например: `--expire-time='2021-09-22 00:00:00+03'`

### 12.1.4. Параметры ведения журнала – **log**

Эти параметры могут использоваться с любой командой.

```
--log-level-console=уровень_сообщений
```

Управляет уровнем сообщений, которые будут выводиться в журнал консоли.

Допустимые уровни:

- verbose;
- log;
- info (по умолчанию);
- warning;
- error;
- off.

Каждый уровень включает все последующие, и с каждым последующим уровнем объем сообщений уменьшается. Вариант «off» отключает вывод в журнал консоли.



Все выводимые в консоль сообщения журнала передаются через stderr, чтобы их можно было отделить от вывода команд show и show-config.

Остальные параметры ведения журнала описаны в подразделах:

- 10.1.6.5 «Уровень сообщений – log-level-file»;
- 10.1.6.6 «Имя файла журнала сообщений – log-filename»;
- 10.1.6.7 «Имя файла журнала сообщений – error-log-filename»;
- 10.1.6.8 «Каталог хранения файлов журнала сообщений – log-directory»;
- 10.1.6.9 «Формат журнала консоли – log-format-console»;
- 10.1.6.10 «Размер журнала для ротации – log-rotation-size»;
- 10.1.6.11 «Возраст журнала для ротации – log-rotation-age».

## 12.2. Параметры подключения

Эти параметры могут использоваться с командами `backup` и `checkdb`.

Компонент `pg_probackup` поддерживает все переменные окружения `libpq`.

```
-d dbname  
--pgdatabase=dbname  
PGDATABASE
```

Задаёт имя базы данных для подключения. Это подключение используется только для управления процессом резервного копирования, при этом можно подключиться к любой существующей базе.

Если этот параметр не задаётся в командной строке, переменной окружения `PGDATABASE` или в конфигурационном файле `pg_probackup.conf`, компонент `pg_probackup` принимает в качестве имени базы значение переменной окружения `PGUSER` или имя текущего пользователя, если переменная `PGUSER` не задана.

```
-h host  
--pghost=host  
PGHOST
```

Указывает имя системы, в которой работает сервер. Если значение начинается с косой черты, оно определяет каталог Unix-сокета.

По умолчанию: `localhost`

```
-p port  
--pgport=port  
PGPORT
```

Указывает TCP-порт или расширение файла локального Unix-сокета, через который сервер принимает подключения.

По умолчанию: `5432`

```
-U username  
--pguser=username  
PGUSER
```

Имя пользователя для подключения.

### 12.2.1. Параметры сжатия – compress

Эти параметры могут использоваться с командами backup и archive-push.

```
--compress-algorithm=алгоритм_сжатия
```

Параметры сжатия файлов журнала описаны в подразделах:

- 10.1.6.3 «Алгоритм сжатия – compress-algorithm»;
- 10.1.6.4 «Уровень сжатия – compress-level».

### 12.3. Параметры архивации

Эти параметры могут использоваться с командой archive-push в значении параметра archive\_command и с командой archive-get в значении restore\_command.

Дополнительно можно задать параметры ведения журнала.

```
--wal-file-path=wal_file_path
```

Задаёт путь файла WAL в archive\_command и restore\_command. Необходимо в качестве значения для данного параметра указать %p для правильной его обработки.

```
--wal-file-name=wal_file_name
```

Задаёт имя файла WAL в archive\_command и restore\_command. Необходимо в качестве значения для данного параметра указать %f для правильной его обработки.

```
--overwrite
```

Разрешает перезапись файлов WAL в архиве. Этот ключ действует при выполнении команды archive-push, когда указанный подкаталог каталога резервных копий уже содержит данный файл WAL, и его нужно заменить новой копией. Без этого ключа archive-push сообщит, что сегмент WAL уже существует, и прервет операцию. Если заменяемый файл не изменился, archive-push пропускает его, независимо от указания параметра overwrite.

```
--batch-size=batch_size
```

Задаёт максимальное количество файлов, которое может быть скопировано в архив одним процессом archive-push или из архива одним процессом archive-get.

```
--archive-timeout=wait_time
```

Задаёт интервал, по истечении которого существующие файлы .part будут считаться потерянными. По умолчанию pg\_probackup ждёт исчезновения этих файлов 300 секунд. Этот параметр можно использовать только с командой archive-push.

```
--no-ready-rename
```

Предотвращает переименование файлов состояния в каталоге archive\_status. Этот параметр полезен, только когда в archive\_command задано несколько команд, и применять его можно только с командой archive-push.

```
--no-sync
```

Не сбрасывать копируемые файлы WAL на диск. Этот параметр позволяет несколько ускорить процесс архивации. Данный параметр можно указать только с командой archive-push.



Использование этого параметра может привести к повреждению архива WAL в случае аварии операционной системы или аппаратного сбоя.

```
--prefetch-dir=путь
```

Каталог, в котором будут храниться предзагружаемые сегменты WAL при использовании параметра --batch-size. Этот каталог должен находиться в той же файловой системе и ниже той же точки монтирования, что и PGDATA/pg\_wal. По умолчанию эти сегменты размещаются в каталоге PGDATA/pg\_wal/pbk\_prefetch. Данный параметр может применяться только с командой archive-get.

```
--no-validate-wal
```

Отключает проверку предзагруженных файлов WAL перед использованием. Этот параметр применяется, если нужно увеличить скорость восстановления. Данный параметр может применяться только с командой archive-get.

## 12.4. Параметры инкрементального восстановления

В этом подразделе описываются параметры, связанные с инкрементальным восстановлением БД. Эти параметры могут передаваться с командой restore.

```
-I incremental_mode  
--incremental-mode=incremental_mode
```

Выбирает инкрементальный режим. Поддерживаются следующие режимы:

- CHECKSUM – заменять только страницы с неподходящей контрольной суммой и LSN;
- LSN – заменять только те страницы, LSN которых больше точки расхождения;
- NONE – обычное восстановление.

Применение параметров полностью описано в разделе 9.2.

## 12.5. Параметры частичного восстановления

В этом подразделе описываются параметры, связанные с частичным восстановлением сервера. Эти параметры могут передаваться с командой restore.

```
--db-exclude=dbname
```

Задаёт имя базы данных, которую нужно исключить из числа восстанавливаемых. Все остальные базы данных в сервере будут восстанавливаться, включая template0 и template1. Этот параметр можно задать несколько раз для исключения нескольких баз данных.

```
--db-include=dbname
```

Задаёт имя базы данных, которую нужно восстановить. Все остальные базы данных восстанавливаться не будут, за исключением template0 и template1. Этот параметр можно задать несколько раз для восстановления нескольких баз данных.



### 13. S3. ХРАНЕНИЕ ДАННЫХ В ОБЛАЧНОМ ХРАНИЛИЩЕ

Компонент pg\_ProBackup поддерживает работу с объектным хранилищем MinIO (открытая реализация Amazon S3 API) в GNU Linux.

MinIO — это объектное хранилище данных с открытым исходным кодом, совместимое с Amazon S3 API. Оно позволяет хранить неструктурированные данные, такие как фото, видео, логи, резервные копии и так далее.

Simple Storage Service — или «S3» это сервис для хранения данных в облаке.



Хранилище S3 должно быть настроено согласно документации Разработчика, по рекомендованным настройкам безопасности

В текущей реализации поддерживается полное и инкрементальное резервное копирование, и восстановление.

При выполнении команды резервного копирования компонент создаст полную резервную копию и автоматически, синхронно отправит ее в хранилище S3 в заданную корзину.

Команда восстановления выполняется в обратном порядке.

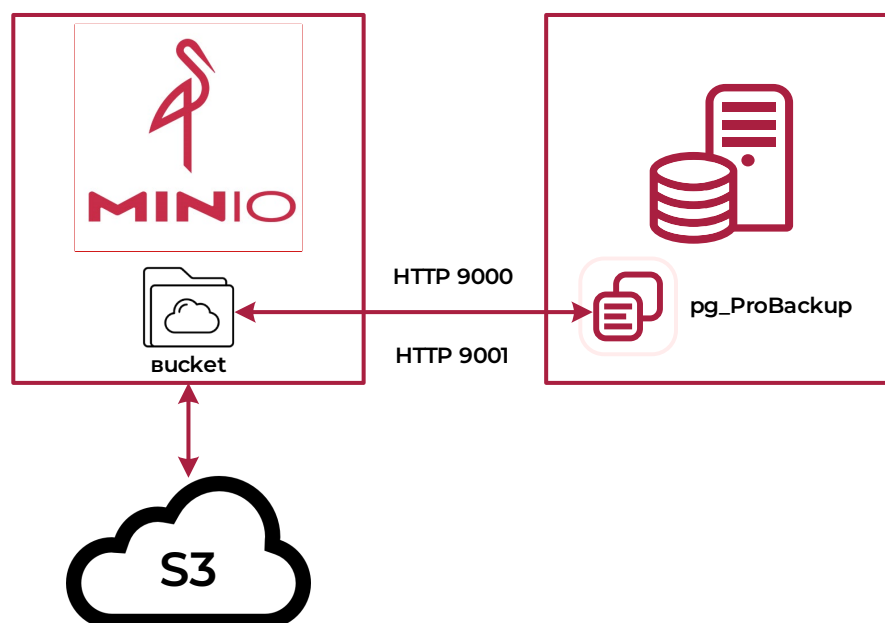


Рисунок 13.1 – Схема работы компонента pg\_ProBackup с хранилищем MinIO



Текущая версия компонента не поддерживает одновременную работу нескольких администраторов с одной и той же резервной копией в S3 хранилище

Доступ к веб-интерфейсу хранилища S3 выполняется по протоколу HTTP, порт 9001.

Доступ к корзине хранилища S3 выполняется по протоколу HTTP, порт 9000.

### 13.1. Предварительная настройка

Для работы должны быть настроены:

- Хранилище S3;
- СУБД как описано в 4.5 и непосредственно в пп. 4.5.1, 4.5.3;
- Компонент pg\_ProBackup, как описано в п. 5.

Конфигурирование СУБД для работы с S3 хранилищем имеет особенности в отличие от настройки СУБД для файлового хранения резервных копий.

В частности конфигурационном файле postgresql.conf, в разделе «Archiving» не требуется установка параметров:

```
#archive_mode = on
#archive_command = 'install_dir/pg_probackup archive-push -B
"backup_dir" --instance instance_name --wal-file-name=%f
[remote_options]'
```

Если они активны, то следует их закомментировать.

### 13.2. Создание пользователя хранилища S3

Предварительно в хранилище требуется создать пользователя данного хранилища от имени и с правами пользователя имеющим привилегированные права администрирования хранилища.

В веб-интерфейсе хранилища перейти по пути Administrator – Identity – Users. В открывшейся вкладке нажать кнопку «Create Users +».

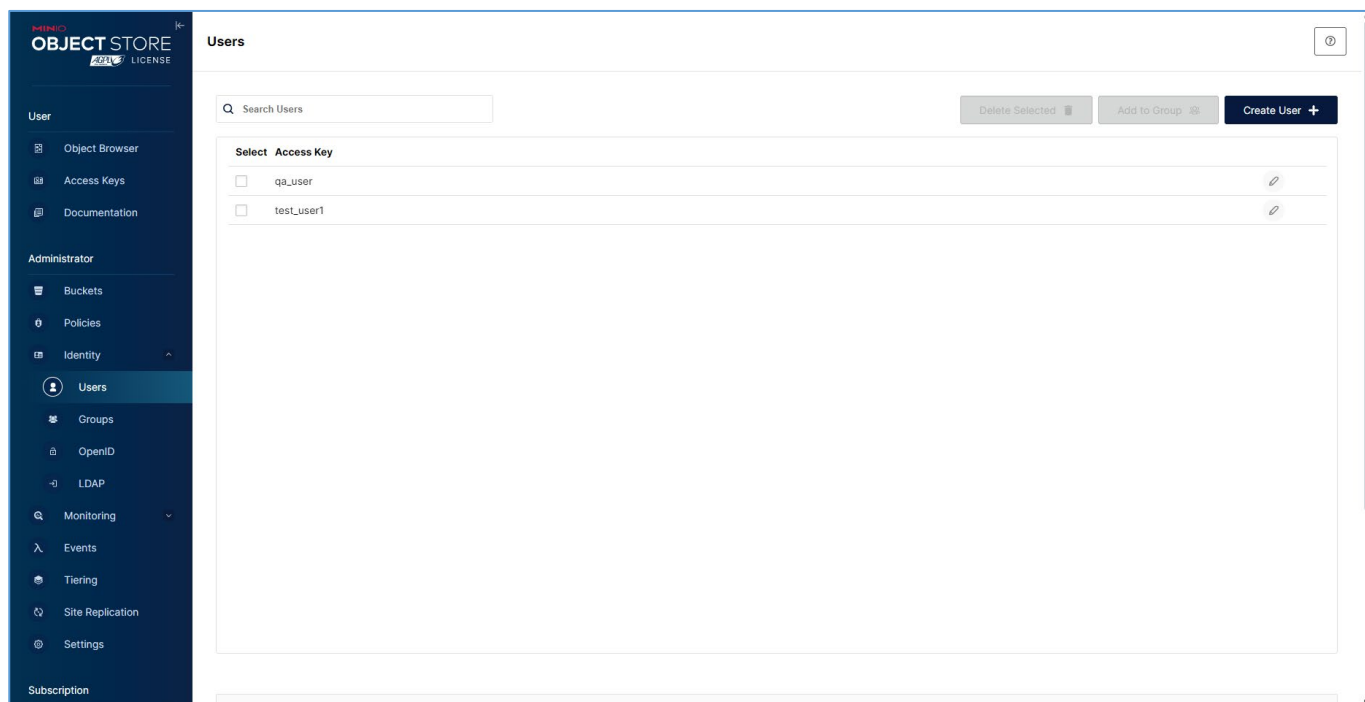


Рисунок 13.2 – Вкладка «Users»

В окне создания пользователя, указать:

- имя пользователя (User Name) – «backup»;
- пароль (Password);
- применяемую политику (Select Policy) - «readwrite», т.е. обычного пользователя.

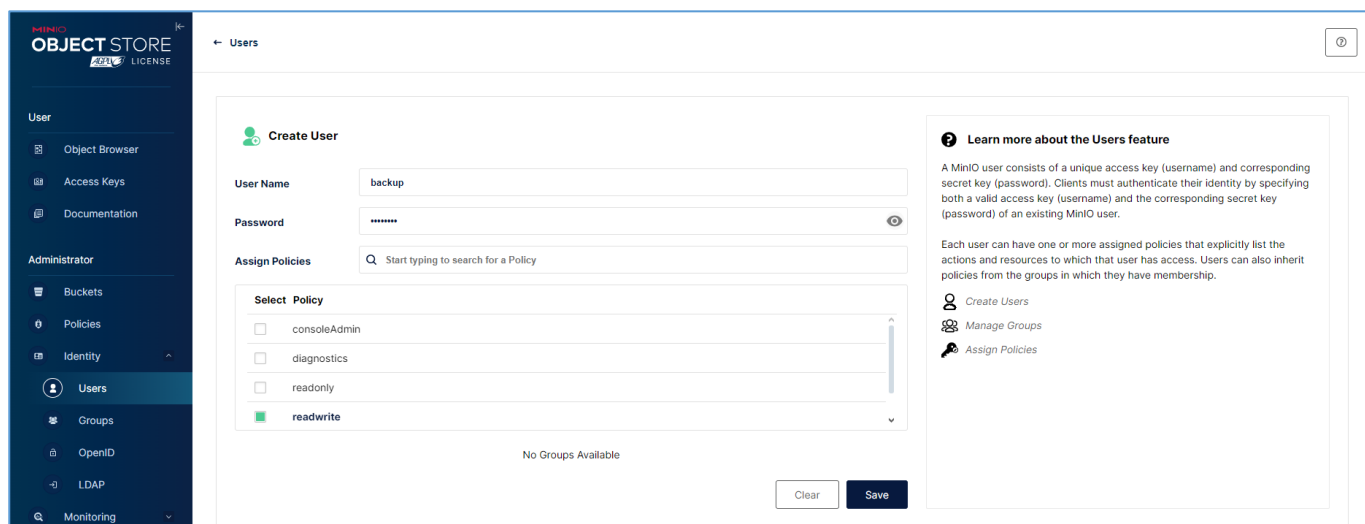


Рисунок 13.3 – Создание пользователя хранилища

Имя и пароль пользователя в последствии будут использоваться для доступа к веб-интерфейсу хранилища S3.

Сохранить внесенные изменения.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

Созданный пользователь отобразится в общем списке.

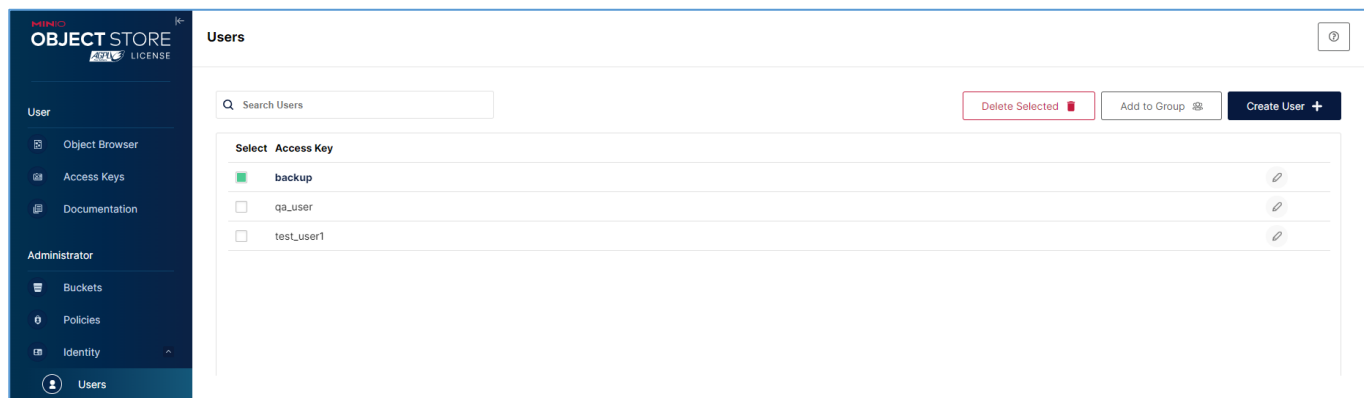


Рисунок 13.4 – Список пользователей хранилища

Нажав на созданного пользователя откроется его карточка, в которой требуется перейти во вкладку «Service Accounts» и нажать кнопку «Create Access Key».

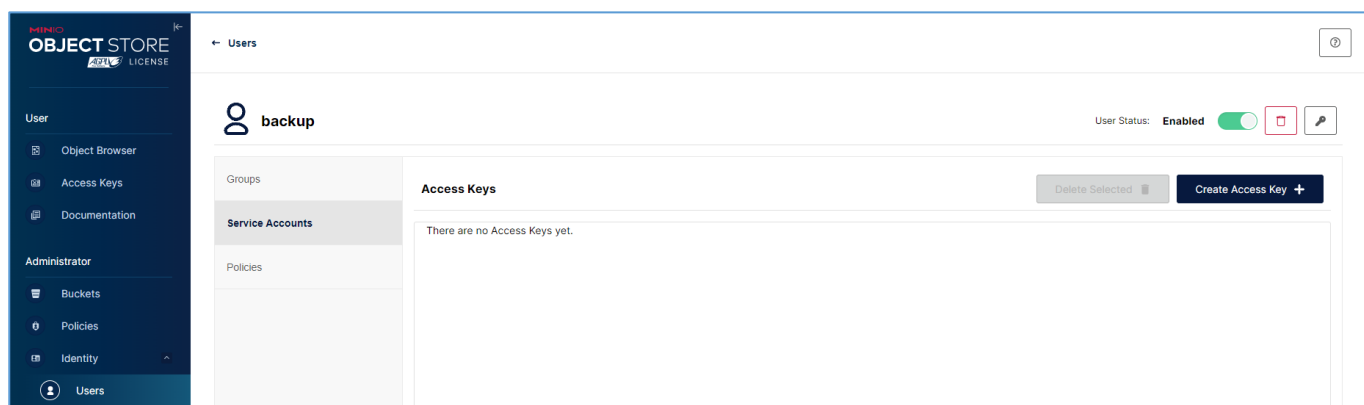


Рисунок 13.5 - Вкладка «Service Accounts»

В открывшемся окне «Create Access Key for backup» будут сгенерированы:

- Access Key - ключ доступа;
- Secret Key - проверочный ключ.

В последствии они будут использоваться для доступа к корзине с хоста СУБД и настройке конфигурационного файла «s3.config» (см. п 13.4).



Проверочный ключ (Secret Key) при его генерации требуется сохранить, т.к. больше данный ключ не будет показан.

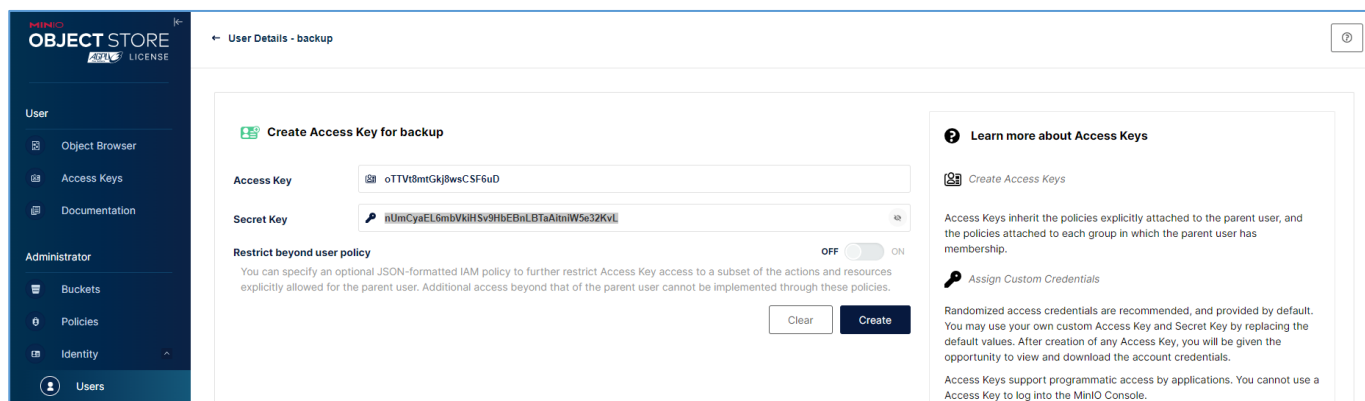


Рисунок 13.6 – Копирование проверочного ключа «Secret Key»

После чего нажать кнопку «Create».

Будет выведено окно «New Access Key Created» с предупреждением о последней возможности сохранить аутентификационные данные.

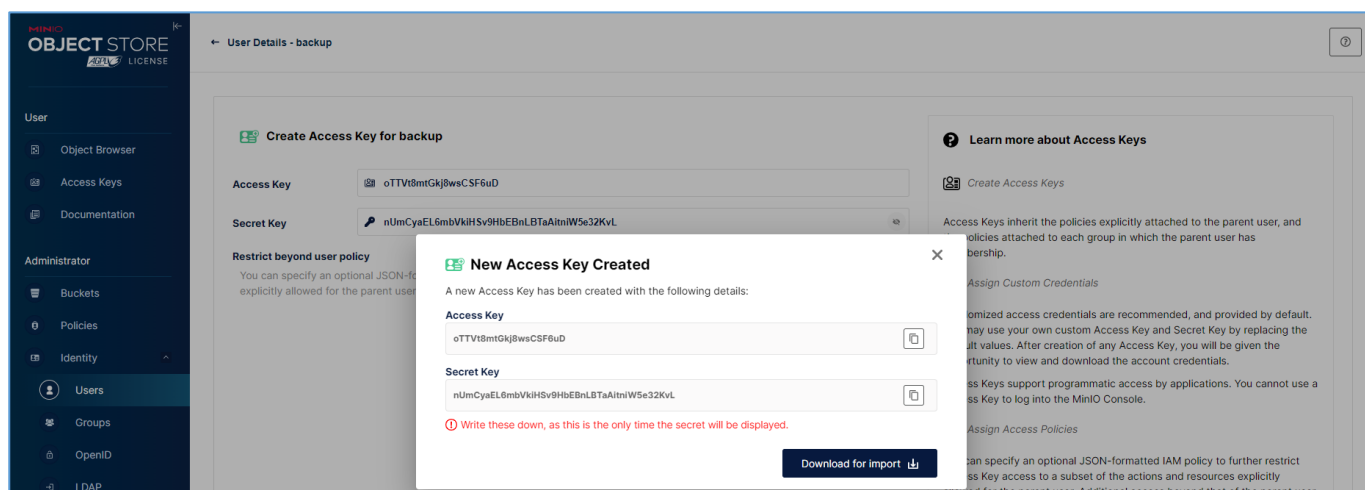


Рисунок 13.7 - Окно «New Access Key Created»

На этом шаге возможно выгрузить аутентификационные данные в файл «credentials.json».

При закрытии окна создание пользователя хранилища S3 закончено.

### 13.3. Создание корзины

Создание корзины для хранения резервных копий создаётся от имени и с правами созданного пользователя хранилища S3.

Пройдя аутентификацию в веб-интерфейсе хранилища, требуется перейти по пути «Administrator» - «Buckets».

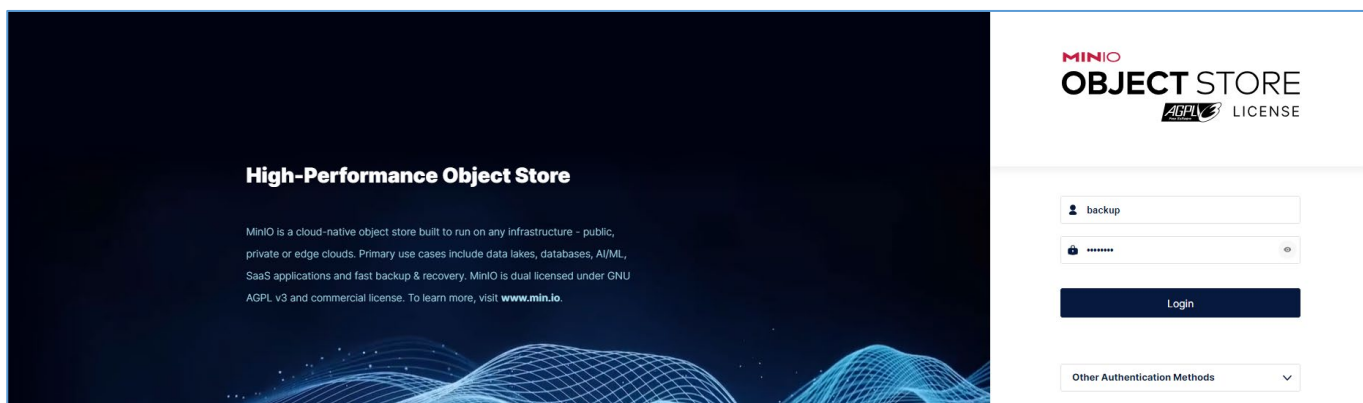


Рисунок 13.8 – Вход в веб-интерфейс хранилища S3

В открывшейся вкладке, отображающей список имеющихся корзин, нажать кнопку «Create Bucket».

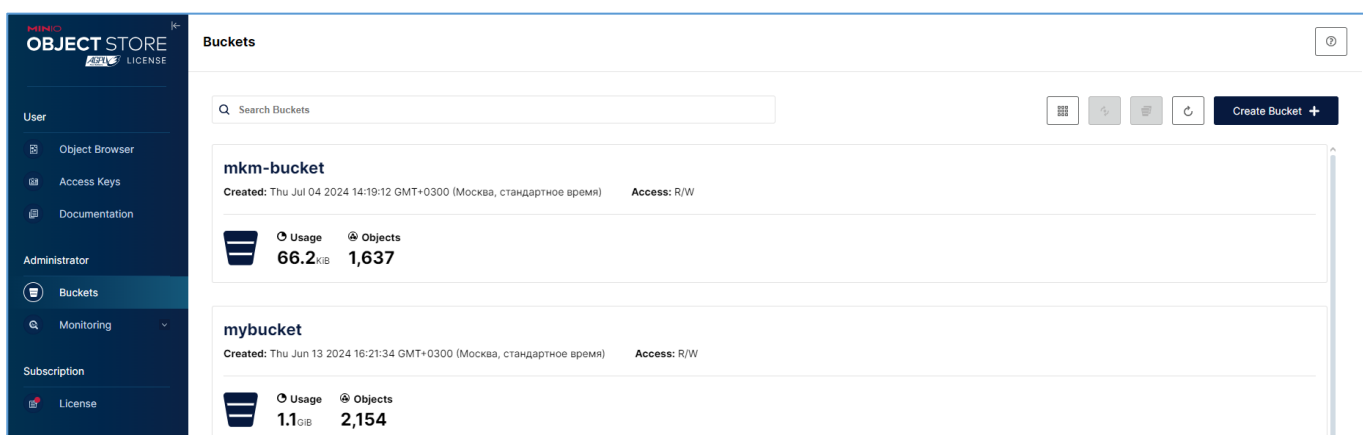


Рисунок 13.9 – Вкладка «Bucket»

В поле «Bucket name» ввести имя создаваемой корзины. В рассматриваемом примере используется имя «a-storage».

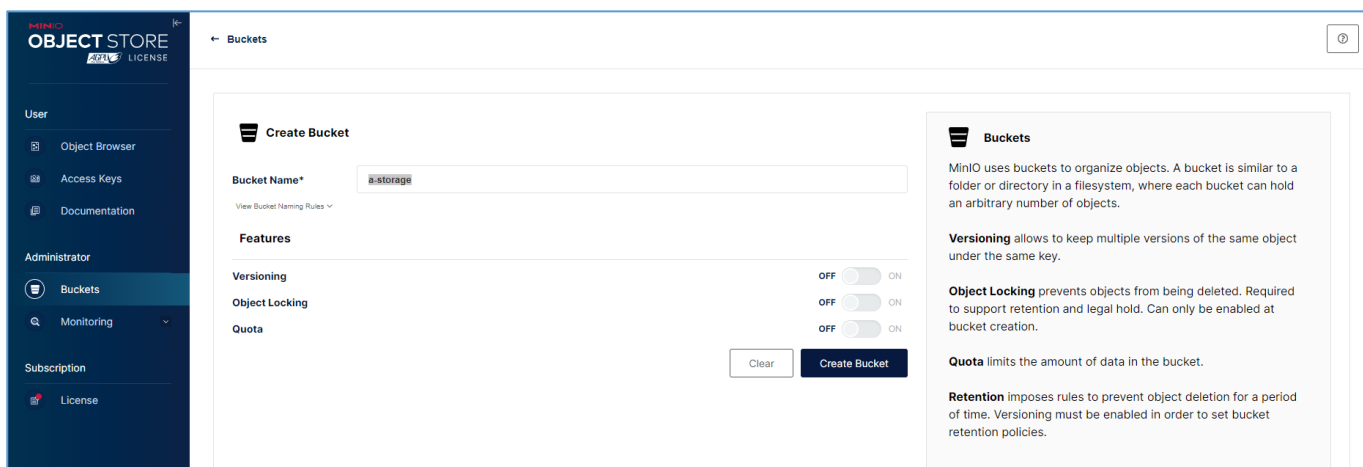


Рисунок 13.10 – Создание корзины

При нажатии кнопки «Create Bucket», корзина будет создана.

### 13.4. Создание конфигурационного файла S3 для компонента pg\_ProBackup

В каталоге резервных копий, используемого компонентом pg\_ProBackup, требуется сформировать конфигурационный файл «s3.config» для работы с хранилищем S3.

В рассматриваемом примере используется команда:

```
gedit /usr/jatoba-*/bin/s3.config
```

В конфигурационном файле «s3.config» вносятся параметры приведенные в таблице 13.1.

Таблица 13.1 – Параметры конфигурационного файла «s3.config»

Параметр	Значение параметра в примере	Обяз. парам.	Описание	Ссылка
access-key	oTTV*****	X	Ключ доступа	13.2
secret-key	nUm*****	X	Проверочный ключ	13.2
s3-host	10.116.101.101	X	IP-адрес хранилища S3	
	<a href="#">u601is-minio-01.jatoba.dev.da.lan</a>		Хост хранилища S3	
s3-port	9000	X	Порт подключения к хранилищу S3. По умолчанию используется порт 9000	
s3-bucket	a-storage	X	Имя созданной корзины	13.3
s3-secure	http	X	Протокол подключения. используется только протокол HTTP.	
s3-buffer-size	8	X	Максимальный размер блока данных, который может быть передан или получен (МБ)	
s3-retries	5		Количество попыток повторной отправки запроса, если первая попытка завершилась неудачно. Значение по умолчанию — 3 попытки	
s3-timeout	300	—	Максимальное время ожидания ответа. Значение по умолчанию — 60 секунд	
s3-interface-provider	minio	—	Тип сервиса, предоставляющий объектное хранилище	
s3-region	—	—	Параметр региона	

```
access-key = ...  
secret-key = ...  
s3-host = хост  
s3-port = 9000  
s3-bucket = имя корзины из minio  
s3-secure = http  
s3-buffer-size=8  
s3-retries=5  
s3-timeout=300  
s3-interface-provider=minio
```

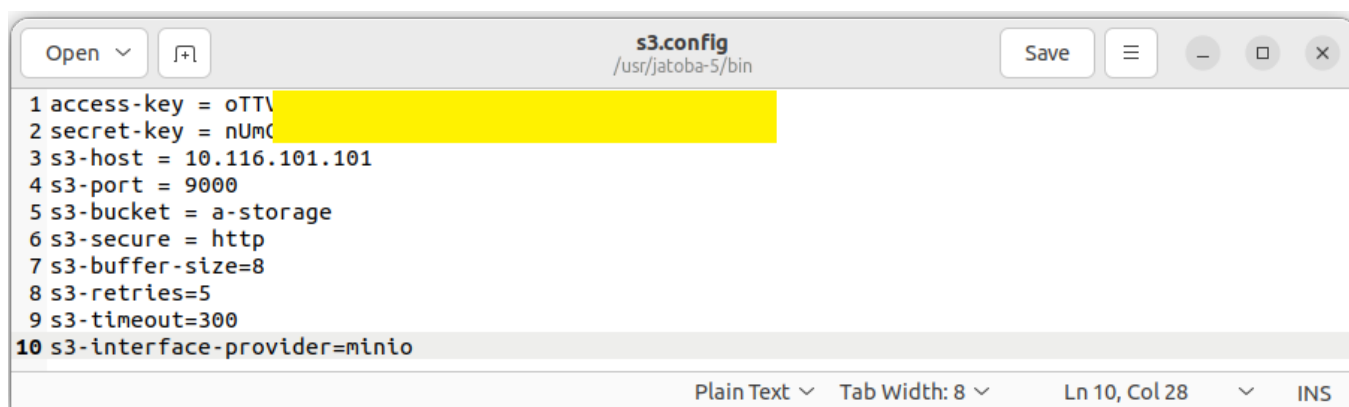


Рисунок 13.11 – Пример конфигурационного файла «s3.config»

### 13.5. Инициализация каталога резервных копий с S3

Команда `init` должна принимать следующие обязательные параметры командной строки:

```
./pg_probackup init -B /s3_backups --s3 --s3-  
config=/path/to/s3.config
```

`backup-dir` - директория резервных копий в корзине объектного хранилища (состав пути объектов см. в `s3.minio.path` выше)

Запуск данной команды вызывает проверку существования пути `backup-dir` в S3. Также проверяется, что по данному пути отсутствуют какие-либо объекты в S3.

Если путь существует и по этому пути уже размещены какие-то объекты, то команда должна выводить ошибку, а дальнейшее исполнение прерываться. Будет выведено сообщение:

```
ERROR: S3: Backup catalog '/' already exists or is not empty
```



В рассматриваемом примере созданная корзина (см. п. 13.4) имеет имя a-storage, что отражено в параметре s3-bucket конфигурационного файла «s3.config» (см. табл. 13.1).

Для инициализации каталога резервных копий требуется выполнить следующие шаги:

- Перейти в директорию под пользователем postgres:

```
su postgres
cd /usr/jatoba-*/bin
```

- Выполнить команду инициализации в терминале ОС:

```
./pg_probackup init -B /a-storage --s3 --s3-config=s3.config
```

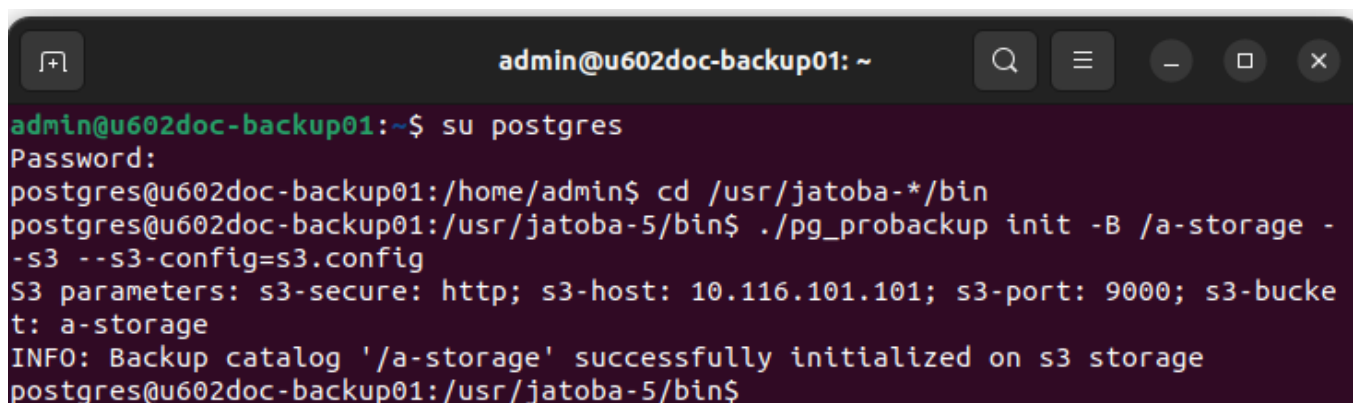


Рисунок 13.12 - Команда инициализации каталога резервных копий с S3

### 13.6. Добавление экземпляра СУБД в каталог резервных копий с S3 параметрами

Команда add-instance принимает следующие параметры командной строки:

```
pg_probackup add-instance -B backup-dir -D pgdata-path
                        --instance=instance-name
                        [-E external-directory-path]
                        [--remote-proto] [--remote-host]
                        [--remote-port] [--remote-path] [--remote-user]
                        [--ssh-options]
                        [--s3] [--s3-config-file=config-file]
```

-B backup-dir - обязательные параметры, задающие директорию резервных копий в корзине объектного хранилища.

-D pgdata-path - обязательные параметры, задающие директорию данных сервера СУБД, который добавляется под управление pg\_probackup.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

--instance=instance-name - имя сервера, как он будет числиться под управлением pg\_probackup

Параметр -E external-directory-path (или --external-dirs) позволяет указать дополнительные директории за пределами директории данных СУБД, чтобы дополнительно их скопировать.

Например, это может пригодиться, когда конфигурационные файлы сервера СУБД лежат отдельно от директории данных. Например, копирование журналов сообщений, которые тоже могут лежать отдельно, и т.п. варианты использования.

Опции удаленного доступа (начинаются на --remote и --ssh) описаны отдельно (см. р. 11).

Опции ведения журнала сообщений (параметры с --log-...) описаны отдельно (см. р. 11).

По команде add-instance на сервере S3 создается конфигурационный файл (объект) pg\_probackup.conf с конфигурацией сервера по умолчанию (обычно перечислено три основных параметра), расположенный по следующему пути в хранилище:

```
<bucket> / <backup-dir> / backups / <instance-name> /  
pg_probackup.conf
```



Команда add-instance с опциями S3 может использоваться без предварительного выполнения команды init.

Команда init с опциями S3 выполняет только проверку наличия директории резервных копий в корзине объектного хранилища и не создаёт никаких других объектов.

Конфигурация сервера включает три основных параметра:

```
# Backup instance information
```

- pgdata - директория данных сервера;
- system-identifier - системный идентификатор экземпляра СУБД (см. pg\_controldata в СУБД)

Значение возможно получить SQL-запросом:

```
SELECT system_identifier FROM pg_control_system();
```

- xlog-seg-size - размер WAL-сегмента в байтах (по умолчанию = 16MB);

Каждый экземпляр СУБД, добавленный в pg\_probackup, хранит свой файл конфигурации pg\_probackup.conf для работы с этим экземпляром СУБД.

В случае успешной инициализации экземпляра СУБД в S3 будет выдано сообщение:

```
INFO: Instance 'test-server' successfully initialized
```

В рассматриваемом примере, для добавления экземпляра СУБД (инстанса) в каталог резервных копий S3 каталога резервных копий требуется выполнить следующие шаги:

- Перейти в директорию под пользователем postgres:

```
su postgres  
cd /usr/jatoba-*/bin
```

- Выполнить команду добавления экземпляра СУБД (инстанса) в каталог резервных копий S3, в терминале ОС:

```
./pg_probackup add-instance -B /a-storage -D  
/var/lib/jatoba/5/data --instance localdb --s3 --s3-config-  
file=s3.config
```

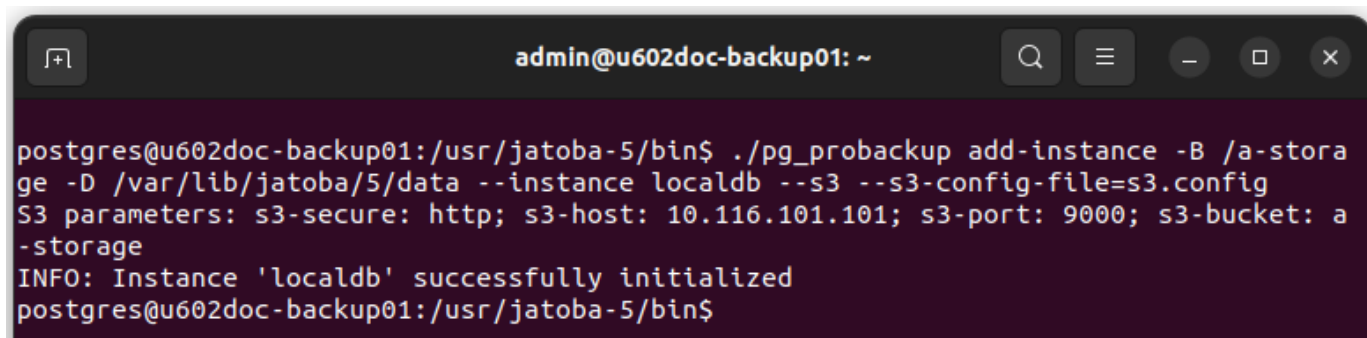


Рисунок 13.13 - Команда добавления экземпляра СУБД (инстанса) в каталог резервных копий S3

### 13.7. Резервное копирование в хранилище S3

Команда резервного копирования в хранилище S3 дополняется двумя параметрами:

- --s3 - включение режима резервного копирования в хранилище S3;

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

– `--s3-config-file` <путь до файла> - путь к используемому конфигурационному файлу.

Параметр «`--s3`» имеет булево значение. По умолчанию параметр имеет значение «`false`», т.е. отключен.

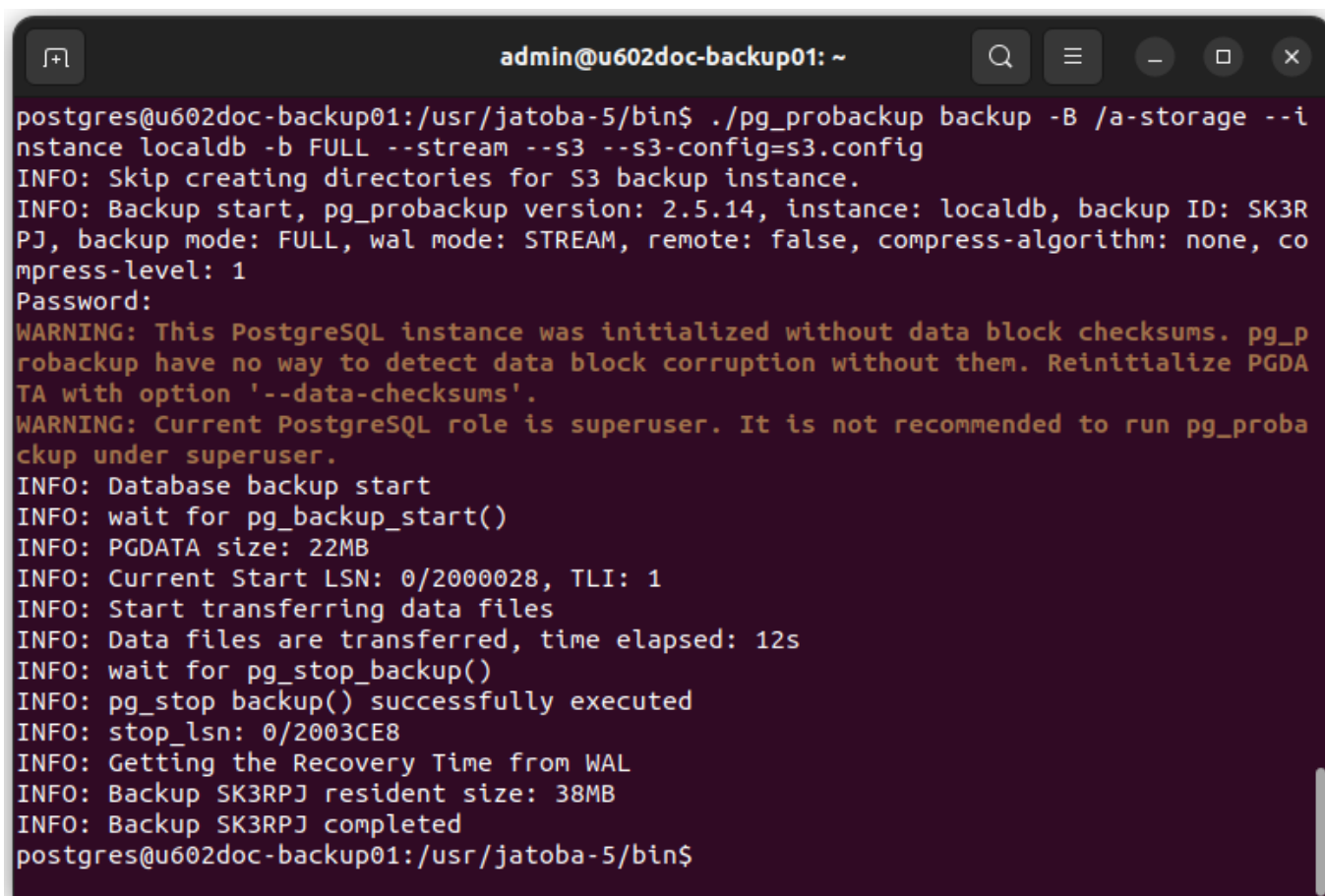
### 13.7.1. Создание резервной копии FULL `--stream` в хранилище S3

Создание резервной копии FULL `--stream` в хранилище S3 выполняется от имени и с правами пользователя ОС «`postgres`».

```
su postgres  
cd /usr/jatoba-*/bin
```

Выполняется резервное копирование командой в терминале ОС:

```
./pg_probackup backup -B /a-storage --instance localdb -b FULL  
--stream --s3 --s3-config=s3.config
```



```
admin@u602doc-backup01: ~  
postgres@u602doc-backup01:/usr/jatoba-5/bin$ ./pg_probackup backup -B /a-storage --i  
nstance localdb -b FULL --stream --s3 --s3-config=s3.config  
INFO: Skip creating directories for S3 backup instance.  
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SK3R  
PJ, backup mode: FULL, wal mode: STREAM, remote: false, compress-algorithm: none, co  
mpress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_p  
robackup have no way to detect data block corruption without them. Reinitialize PGDA  
TA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_proba  
ckup under superuser.  
INFO: Database backup start  
INFO: wait for pg_backup_start()  
INFO: PGDATA size: 22MB  
INFO: Current Start LSN: 0/2000028, TLI: 1  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 12s  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: stop_lsn: 0/2003CE8  
INFO: Getting the Recovery Time from WAL  
INFO: Backup SK3RPJ resident size: 38MB  
INFO: Backup SK3RPJ completed  
postgres@u602doc-backup01:/usr/jatoba-5/bin$
```

Рисунок 13.14 - 13.7.1. Создание резервной копии FULL `--stream` в хранилище S3

Созданная резервная копия отразится в хранилище S3.

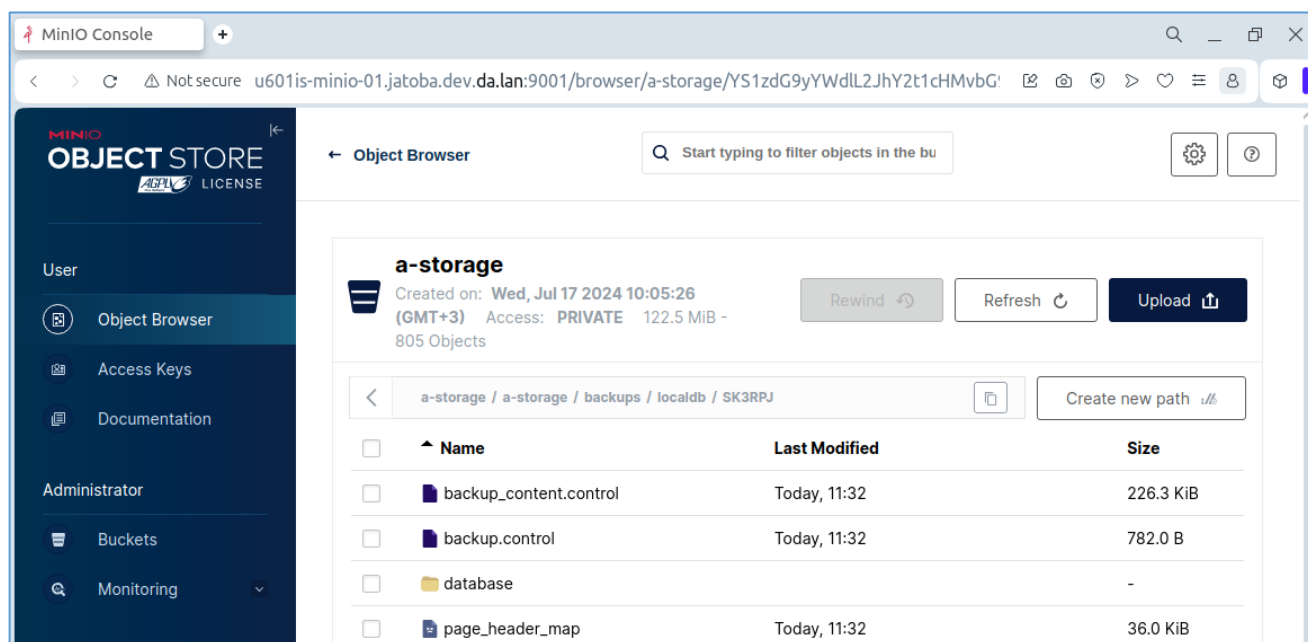


Рисунок 13.15 – Созданная резервная копия в хранилище S3

### 13.7.2. Создание резервной копии DELTA –stream в хранилище S3

Создание инкрементальной резервной копии DELTA –stream в хранилище S3 выполняется от имени и с правами пользователя ОС «postgres».

```
su postgres
cd /usr/jatoba-*/bin
```

Выполняется резервное копирование командой в терминале ОС:

```
./pg_probackup backup -B /a-storage --instance localdb -b DELTA
--stream --s3 --s3-config=s3.config
```

```
admin@u602doc-backup01: ~  
postgres@u602doc-backup01:/usr/jatoba-5/bin$ ./pg_probackup backup -B /a-storage --i  
nstance localdb -b DELTA --stream --s3 --s3-config=s3.config  
INFO: Skip creating directories for S3 backup instance.  
INFO: Backup start, pg_probackup version: 2.5.14, instance: localdb, backup ID: SK3S  
13, backup mode: DELTA, wal mode: STREAM, remote: false, compress-algorithm: none, c  
ompress-level: 1  
Password:  
WARNING: This PostgreSQL instance was initialized without data block checksums. pg_p  
robackup have no way to detect data block corruption without them. Reinitialize PGDA  
TA with option '--data-checksums'.  
WARNING: Current PostgreSQL role is superuser. It is not recommended to run pg_proba  
ckup under superuser.  
INFO: Database backup start  
INFO: wait for pg_backup_start()  
INFO: Parent backup: SK3RPJ  
INFO: PGDATA size: 22MB  
INFO: Current Start LSN: 0/4000028, TLI: 1  
INFO: Parent Start LSN: 0/2000028, TLI: 1  
INFO: Start transferring data files  
INFO: Data files are transferred, time elapsed: 0  
INFO: wait for pg_stop_backup()  
INFO: pg_stop_backup() successfully executed  
INFO: stop_lsn: 0/4000198  
INFO: Getting the Recovery Time from WAL  
INFO: Backup SK3S13 resident size: 32MB  
INFO: Backup SK3S13 completed  
postgres@u602doc-backup01:/usr/jatoba-5/bin$
```

Рисунок 13.16 - Создание резервной копии DELTA –stream в хранилище S3  
Созданная резервная копия отразится в хранилище S3.

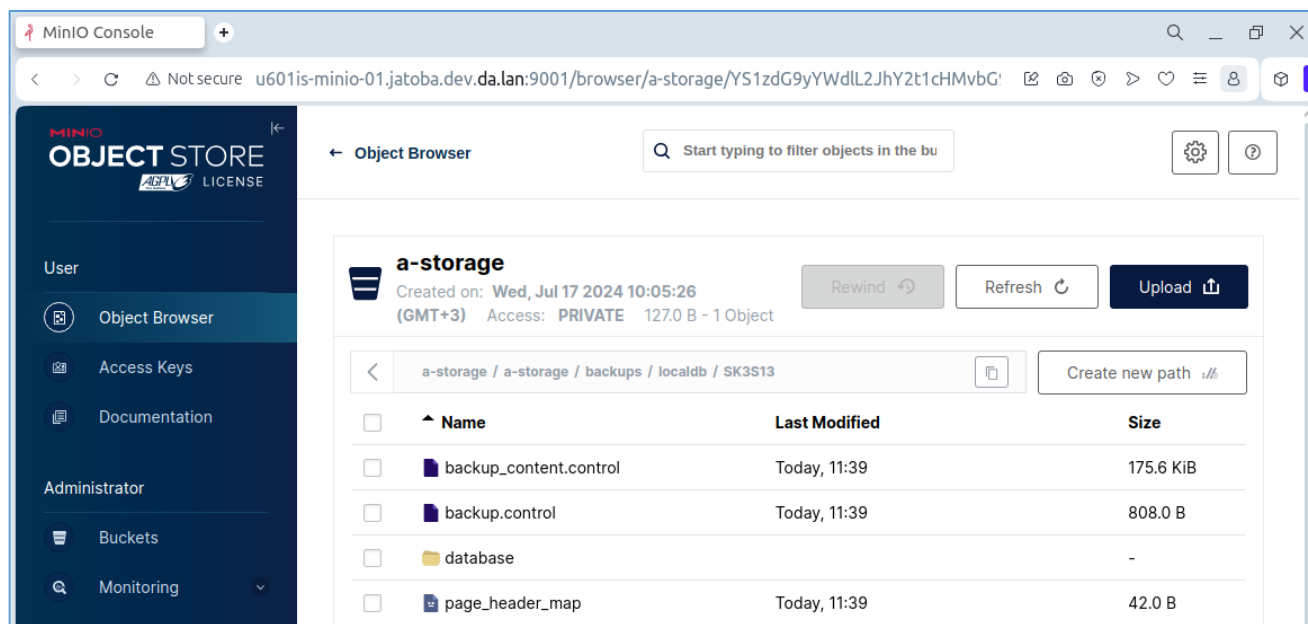


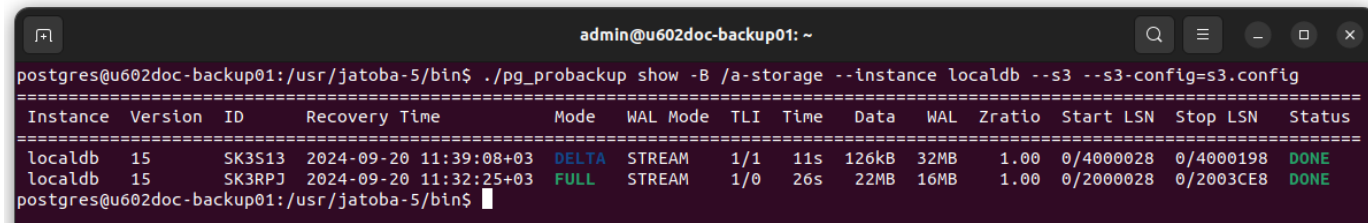
Рисунок 13.17 - Созданная резервная копия в хранилище S3

### 13.7.3. Вывод информации о резервных копиях в каталоге резервных копий S3

Вывод информации о резервных копиях в каталоге резервных копий описана в п.п. 10.1.9 и 5.2.1 настоящего документа. Синтаксис команды аналогичен, за исключением того, что указывается имя хранилища S3 и конфигурационный файл.

Вывести сводную таблицу о состоянии резервных копий, хранящихся на сервисе S3:

```
./pg_probackup show -B /a-storage --instance localdb --s3 --s3-  
config=s3.config
```



Instance	Version	ID	Recovery Time	Mode	WAL Mode	TLI	Time	Data	WAL	Zratio	Start LSN	Stop LSN	Status
localdb	15	SK3S13	2024-09-20 11:39:08+03	DELTA	STREAM	1/1	11s	126kB	32MB	1.00	0/4000028	0/4000198	DONE
localdb	15	SK3RPJ	2024-09-20 11:32:25+03	FULL	STREAM	1/0	26s	22MB	16MB	1.00	0/2000028	0/2003CE8	DONE

Рисунок 13.18 - Вывод информации о резервных копиях в каталоге резервных копий S3

### 13.8. Восстановление сервера баз данных из хранилища S3

Восстановление сервера БД описано в п. 9.1 настоящего документа.

Для восстановления сервера баз данных из хранилища S3, в синтаксис команды добавляются 2 параметра:

- --s3 - включение режима резервного копирования в хранилище S3;
- --s3-config-file <путь до файла> - путь к используемому конфигурационному файлу.

```
pg_probackup restore -B backup_dir --instance instance_name -i  
backup_id --s3 --s3-config-file <path to the file>
```

Восстановление в прежний каталог потребует выполнения следующих действий;

- Войти в режим администратора;
- Остановить службу СУБД «jatoba-5»;
- Удалить каталог /var/lib/jatoba/5/data/;
- Аутентифицироваться от имени и с правами пользователя ОС postgres;
- Вывести информацию о резервных копиях для получения ID копии;
- Выполнить команду восстановления СУБД из хранилища S3.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



```
# sudo su
# systemctl stop jatoba-5
# rm -r /var/lib/jatoba/5/data/
# su postgres
# cd /usr/jatoba-5/bin/
# ./pg_probackup show -B /a-storage --instance localdb --s3 --s3-config=s3.config
# ./pg_probackup restore -B /a-storage --instance localdb -D /var/lib/jatoba/5/data -i SK3RPJ --s3 --s3-config=s3.config
```

```
admin@u602doc-backup01:~$ sudo su
[sudo] password for admin:
root@u602doc-backup01:/home/admin# systemctl stop jatoba-5
root@u602doc-backup01:/home/admin# rm -r /var/lib/jatoba/5/data/
root@u602doc-backup01:/home/admin# su postgres
postgres@u602doc-backup01:/home/admin$ cd /usr/jatoba-5/bin/
postgres@u602doc-backup01:/usr/jatoba-5/bin$ ./pg_probackup show -B /a-storage --instance localdb --s3 --s3-config=s3.config
=====
Instance Version ID Recovery Time Mode WAL Mode TLI Time Data WAL Zratio Start LSN Stop LSN Status
=====
localdb 15 SK3S13 2024-09-20 11:39:08+03 DELTA STREAM 1/1 11s 126kB 32MB 1.00 0/4000028 0/4000198 DONE
localdb 15 SK3RPJ 2024-09-20 11:32:25+03 FULL STREAM 1/0 26s 22MB 16MB 1.00 0/2000028 0/2003CE8 OK
postgres@u602doc-backup01:/usr/jatoba-5/bin$ ./pg_probackup restore -B /a-storage --instance localdb -D /var/lib/jatoba/5/data -i SK3RPJ --s3 --s3-config=s3.config
INFO: Validating backup SK3RPJ
INFO: Backup SK3RPJ data files are valid
INFO: Backup SK3RPJ WAL segments are valid
INFO: Backup SK3RPJ is valid.
INFO: Restoring the database from backup SK3RPJ
INFO: Start restoring backup files. PGDATA size: 38MB
INFO: Backup files are restored. Transferred bytes: 38MB, time elapsed: 12s
INFO: Restore incremental ratio (less is better): 100% (38MB/38MB)
INFO: Syncing restored files to disk
INFO: Restored backup files are synced, time elapsed: 0
INFO: Restore of backup SK3RPJ completed.
postgres@u602doc-backup01:/usr/jatoba-5/bin$
```

Рисунок 13.19 – Восстановление СУБД из хранилища S3

### 13.9. Прочие команды pg\_ProBackup для S3

Компонент поддерживает набор команд, описанный в п. 10.1., а также команды адаптированные для работы с хранилищем S3, приведенные ниже.

#### 13.9.1. Удаление экземпляра СУБД из каталога резервных копий

Команда del-instance принимает следующие параметры командной строки:

```
pg_probackup del-instance -B backup-dir --instance=instance-name
                        [--s3] [--s3-config-file=config-file]
```

-B backup-dir - обязательные параметры, задающие директорию резервных копии в корзине объектного хранилища

--instance=instance-name - имя сервера, который надо удалить из директории резервных копий в корзине объектного хранилища



По команде del-instance на сервере S3 должны удаляться все объекты, относящиеся к заданному серверу.

В случае успешного удаление экземпляра СУБД будет выдано сообщение

```
Instance '%s' successfully deleted
```

где %s - подставленное имя экземпляра СУБД instance-name

В случае разных ошибок о невозможности удаления тех или иных объектов будет выдано сообщение

```
Can't remove \"%s\": %s"
```

где:

- первая %s - подставленное имя экземпляра СУБД instance-name
- вторая %s - строка причины ошибки удаления

### 13.9.2. Изменение конфигурационных параметров pg\_probackup для заданного экземпляра СУБД

Команда set-config принимает следующие параметры командной строки:

```
pg_probackup set-config -B backup-dir --instance=instance-name
```

```
[-D pgdata-path]
[-E external-directories-paths]
[--restore-command=cmdline]
[--log-level-console=log-level-console]
[--log-level-file=log-level-file]
[--log-format-file=log-format-file]
[--log-filename=log-filename]
[--error-log-filename=error-log-filename]
[--log-directory=log-directory]
[--log-rotation-size=log-rotation-size]
[--log-rotation-age=log-rotation-age]
[--retention-redundancy=retention-redundancy]
[--retention-window=retention-window]
[--wal-depth=wal-depth]
[--compress-algorithm=compress-algorithm]
[--compress-level=compress-level]

[--archive-timeout=timeout]
```

```
[-d dbname] [-h host] [-p port] [-U username]  
[--remote-proto] [--remote-host]  
[--remote-port] [--remote-path] [--remote-user]  
[--ssh-options]  
[--s3] [--s3-config-file=config-file]
```

-В backup-dir - обязательные параметры, задающие директорию резервных копии в корзине объектного хранилища

--instance=instance-name - имя сервера, который надо удалить из директории резервных копий в корзине объектного хранилища

Команда set-config позволяет изменять конфигурацию заданного экземпляра СУБД. Конфигурация сохраняется в файле pg\_probackup.conf (см. выше), который создается командой add-instance, в S3

Параметры -D и -E задают информацию о директории данных сервера и дополнительных директориях, которые надо поменять в параметрах pgdata и external-dirs в pg\_probackup.conf.

Параметры -d -h -p -U задают параметр подключения к экземпляру СУБД (данные опции аналогичны одноименным опциям psql); в конфигурационном файле pg\_probackup.conf это отражается в параметрах pgdatabase, pghost, pgport и pguser

Остальные параметры командной строки меняют одноименные параметры в конфигурационном файле. Все допустимые параметры делятся на подгруппы и:

- а) параметры логирования (содержат --log-...)
- б) параметры политики хранения (содержат --retention)
- в) параметры сжатия (содержат --compress)
- г) параметры, связанные с WAL (содержат --wal или --archive)
- д) параметры удаленного режима (содержат --remote и --ssh)

Список всех доступных параметров в конфигурационном файле

- Instance options: pgdata, system-identifier, xlog-seg-size, external-dirs
- Connection options: pgdatabase, pghost, pgport, pguser

- Replica options: master-db, master-host, master-port, master-user, replica-timeout
- Archive options: archive-timeout, archive-host, archive-port, archive-user, restore-command
- Logging options: log-level-console, log-level-file, log-format-console, log-format-file, log-filename, error-log-filename, log-directory, log-rotation-size, log-rotation-age
- Retention options: retention-redundancy, retention-window, wal-depth
- Compression options: compress-algorithm, compress-level
- Remote backup options: remote-proto, remote-host, remote-port, remote-path, remote-user, ssh-options, ssh-config

### 13.9.3. Изменение конфигурационных параметров pg\_probackup для заданной резервной копии заданного экземпляра СУБД

Команда set-backup принимает следующие параметры командной строки:

```
pg_probackup set-backup -B backup-dir --instance=instance-name  
-i backup-id  
[--ttl=interval] [--expire-time=time] [--note=text]  
[--s3] [--s3-config-file=config-file]
```

-B backup-dir - обязательные параметры, задающие директорию резервных копий в корзине объектного хранилища.

--instance=instance-name - имя сервера, конфигурация резервной копии которого собирается поменять.

-i backup-id - идентификатор резервной копии.

Команда set-backup позволяет изменять конфигурацию заданной резервной копии заданного экземпляра СУБД, которая хранится в файле backup.control.

Конфигурация резервной копии расположена в S3 по следующему пути: "<bucket> / <backup-dir> / backups / <instance-name> / <backup-id> /"

Команда set-backup меняет одноименные параметры в backup.control: ttl, expire-time и note.

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Каталог копий (backup_dir)	– каталог, в котором хранятся все файлы резервных копий и метаданные
Каталог данных (data_dir)	– каталог, содержащий данные СУБД, с которой делается копия
Внешние каталоги	– дополнительные каталоги, не относящиеся к СУБД, которые могут содержать вспомогательные скрипты, описания или конфигурационные файлы, используемые администратором в повседневной работе
Имя экземпляра (instance_name)	– имя подкаталога, в котором будут храниться файлы копируемых данных и WAL для этого экземпляра СУБД
ID	– идентификатор резервной копии, определяющий из какой резервной копии будет восстановлен сервер баз данных
PITR	– операция восстановления на определенный момент времени
system-identifier	– уникальный идентификатор экземпляра СУБД
CHECKSUM	– способ инкрементального восстановления при котором pg_probackup прочитает все файлы данных в целевом каталоге, проверит заголовок и контрольную сумму каждой страницы и заменит только некорректные страницы
LSN	– способ инкрементального восстановления при котором pg_probackup прочитает файл pg_control в каталоге данных, получит из него значения REDO LSN и REDO TLI, позволяющие определить точку в истории (точку сдвига), в которой состояние каталога данных сдвинулись с цепочки резервных копий

- FULL
- режим резервного копирования, при котором создается полная резервная копия, содержащая все файлы директории данных, необходимые для восстановления
- DELTA
- режим резервного копирования, при котором считываются все файлы данных в каталоге данных и создается инкрементальная копия для страниц, измененных со времени предыдущего копирования
- PAGE
- режим резервного копирования, при котором создается инкрементальная резервная копия с файлами WAL, которые были изменены со времени последней полной или инкрементальной копии
- PTRACK
- режим резервного копирования, при котором создается инкрементальная резервная копия со страницами, изменения в которых отслеживались «на лету»

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

GB	–	Gbyte – единицы измерения количества информации, гигабайт
ID	–	Identifier – идентификатор
JSON	–	JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript
KB	–	Kbyte – единицы измерения количества информации, килобайт
MB	–	Mbyte – единицы измерения количества информации, мегабайт
SQL	–	Structured Query Language – язык структурированных запросов
TB	–	Tbyte – единицы измерения количества информации, терабайт
WAL	–	Write-Ahead Logging – журнал предзаписи
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

## Лист регистрации изменений

Дата внесения изм: \_\_\_\_\_