

# Онлайн образование

[otus.ru](https://otus.ru)



Проверить, идет ли запись

# Меня хорошо видно && слышно?



Тема вебинара

# Резервное копирование и восстановление



**Коробков Виктор**

Консультант команды технологического обеспечения  
ООО «ИТ ИКС5 Технологии»

Telegram: @Korobkov\_Viktor



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в Slack



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Маршрут вебинара



Бэст практикс

Логический бэкап

Физический бэкап

Рефлексия

# Цели вебинара

После занятия вы сможете

1. Настраивать бэкапы
  2. Восстанавливать данные
- 
- 
- 
-

# Смысл

## Зачем вам это уметь

1. Чтобы предотвратить потерю базы

---

---

---

---



# Бэст практикс





# Бэст практикс: основные рекомендации

1. Бэкап должен быть всегда.
2. Бэкап должен быть автоматическим.
3. Восстановление из бэкапа — это крайняя мера.
4. Бэкап нужно хранить отдельно от данных и минимум 2 недели.
5. Бэкап нужно регулярно проверять.
6. Полезно дублировать бэкап на удаленную площадку.
7. Бэкап – это нагрузка на работающую систему.

# Бэст практикс: основные рекомендации

## Правило 3-2-1 в резервировании данных:

1. Иметь не менее трех экземпляров данных.
2. Хранить копии не менее чем на двух носителях.
3. Хранить не менее одной копии данных за пределами офиса.

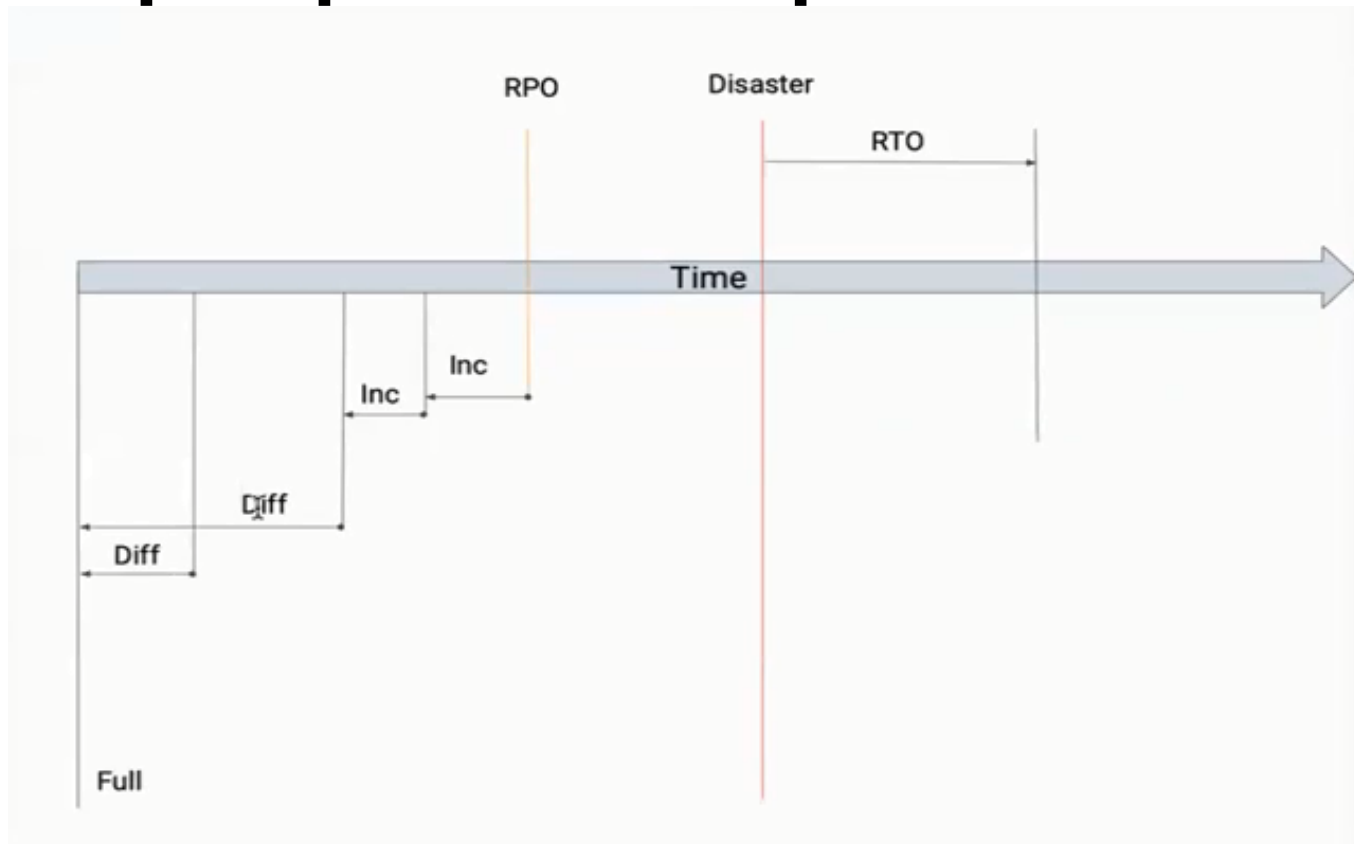
# Терминология

- **RTO** (Recovery Time Objective) – определение времени, требуемого на восстановление резервной копии. Например: восстановление из РК должно занимать не более 1 часа.
- **RPO** (Recovery Point Objective) – точка во времени (Point in Time) на которую должны быть восстановлены данные. Например, данные должны быть восстановлены на состояние не «дольше», чем 24 часа с момента сбоя.
- **Backup Level** – уровень резервного копирования (0 – Full, 1 – Differential, 2 – Incremental), это стратегии выбора данных для копирования.
- **Глубина резервного копирования** – определяет, как долго хранятся копии.

# Уровни резервного копирования

- **Full** – полное резервное копирование, для восстановления требуется только эта резервная копия.
- **Differential** – разностное резервное копирование: копируется только то, что изменилось с последнего полного резервного копирования. Для восстановления требуется последняя полная версия + последняя дифференциальная копия.
- **Incremental** – инкрементальное резервное копирование: копируется только то, что изменилось с последнего прохода резервного копирования. Для восстановления требуется последняя полная версия + последняя дифференциальная копия (если есть) + ВСЕ инкрементальные копии с момента последней полной/дифференциальной копии.

# Уровни резервного копирования



# Виды бэкапов

1. Логическое резервное копирование
2. Физическое резервное копирование

# 1. Логическое копирование

- + можно сделать копию отдельного объекта или базы
- + можно восстановиться на кластере другой основной версии
- + можно восстановиться на другой архитектуре
- невысокая скорость относительно физической

# 1. Логическое копирование - COPY

```
COPY { имя_таблицы [ ( имя_столбца [ ... ] ) ] | ( запрос ) }  
      TO { 'имя_файла' | PROGRAM 'команда' | STDOUT } [ [ WITH ] ( параметр [ ... ] ) ]  
COPY имя_таблицы [ ( имя_столбца [ ... ] ) ]  
      FROM { 'имя_файла' | PROGRAM 'команда' | STDIN } [ [ WITH ] ( параметр [ ... ] ) ]
```

FORMAT имя\_формата

oids [ boolean ]

FREEZE [ boolean ]

DELIMITER 'символ\_разделитель'

NULL 'маркер\_NULL'

HEADER [ boolean ]

QUOTE 'символ\_кавычек'

ESCAPE 'символ\_экранирования'

FORCE\_QUOTE { ( имя\_столбца [ ... ] ) | \* }

FORCE\_NOT\_NULL ( имя\_столбца [ ... ] )

FORCE\_NULL ( имя\_столбца [ ... ] )

ENCODING 'имя\_кодировки'



# 1. Логическое копирование – PG\_DUMP

## утилита PG\_DUMP

- + выдает на консоль или в файл либо SQL-скрипт,
- + либо архив в специальном формате с оглавлением
- + поддерживает параллельное выполнение
- + позволяет ограничить набор выгружаемых объектов  
(таблицы --table, схемы --schema-only, данные --data-only и т.п.)
- по умолчанию не создает tablespace и юзеров

```
$ pg_dump -d backup --create
```

```
$ pg_dump -d backup --create | gzip > backup.gz
```

```
$ pg_dump -d backup -Fc >1.gz - для pg_restore
```

# 1. Логическое копирование – PG\_DUMP

утилита PG\_DUMP примеры под Windows:

бекап базы в каталог (восстановление через pg\_restore)

```
pg_dump --file "диск:\\путь" --host "IP/имя" --port "5432" --username "пользователь"  
--no-password --format=d --jobs 10 "имя базы"
```

бекап базы в файл

```
pg_dump --file "диск:\\путь\\имя.sql" --host "IP/имя" --port "5432"  
--username "пользователь" --no-password --format=c  
"имя базы"
```

бекап таблицы

```
pg_dump --file "диск:\\путь\\имя.sql" --host "IP/имя" --port "5432"  
--username "пользователь" --no-password --format=c  
--table "имя таблицы" "имя базы"
```



# 1. Логическое копирование — восстановление

Так как это простой SQL скрипт:

**\$psql < 1.sql**

- заранее должны быть созданы роли и табличные пространства

**pg\_restore** - если архив с оглавлением

- (позволяет ограничить набор объектов при восстановлении)
- поддерживает параллельное выполнение
- заранее должны быть созданы роли, табличные пространства и БД!!!
- после восстановления имеет смысл выполнить сбор статистики (ANALIZE)

**\$pg\_restore 2.gz**

# 1. Логическое копирование – PG\_DUMPALL

- сохраняет весь кластер, включая роли и табличные пространства
- выдает на консоль или в файл SQL-скрипт
- параллельное выполнение не поддерживается
- можно выгрузить только глобальные объекты и воспользоваться pg\_dump

```
$pg_dumpall >backup.sql
```

```
$pg_dumpall --clean --globals-only >globals.sql
```

```
$pg_dumpall --clean --schema-only >schema.sql
```

Восстановление

```
$psql < backup.sql
```

## 2 Физическое копирование

Используется механизм восстановления после сбоя:

копия данных и журналы предзаписи

+ скорость восстановления

+ можно восстановить кластер на определенный момент времени

- нельзя восстановить отдельную базу данных, только весь кластер

- восстановление только на той же основной версии и архитектуре

# Виды физического резервирования

**Холодное** - когда БД остановлена

- сервер корректно остановлен (необходимы только файлы данных)
- некорректно выключенный (файлы данных и wal сегменты)

**Горячее** - на работающем экземпляре

- необходимы как файлы данных, так и wal сегменты, причем нужно проконтролировать, чтобы сервер сохранил все wal файлы на время копирования основных данных

# Создание автономной копии

Автономная копия содержит и файлы данных, и WAL

Резервное копирование — утилита **pg\_basebackup** :

- подключается к серверу по протоколу репликации
- выполняет контрольную точку
- переключается на следующий сегмент WAL
- копирует файловую систему в указанный каталог
- переключается на следующий сегмент WAL
- сохраняет все сегменты WAL, сгенерированные за время копирования

## Восстановление

- разворачиваем созданную автономную копию
- запускаем сервер

# Создание автономной копии

## Протокол репликации

- получение потока журнальных записей
- команды управления резервным копированием и репликацией

Обслуживается процессом **wal\_sender**

Параметр **wal\_level = replica**

## Слот репликации

- серверный объект для получения журнальных записей
- помнит, какая запись была считана последней
- сегмент WAL не удаляется, пока он полностью не прочитан через слот

```
SELECT name, setting FROM pg_settings WHERE name IN ('wal_level','max_wal_senders');
```

Необходимо настроить файервол в файле pg\_hba.conf

```
SELECT type, database, user_name, address, auth_method  
FROM pg_hba_file_rules() WHERE database = '{replication}';
```



# Создание автономной копии

Создадим 2 кластер

```
$pg_createcluster -d /var/lib/postgresql/12/main2 12 main2
```

Удалим оттуда файлы

```
$rm -rf /var/lib/postgresql/12/main2
```

Сделаем бэкап нашей БД

```
$pg_basebackup -p 5432 -D /var/lib/postgresql/12/main2
```

Зададим другой порт (необязательно)

```
$echo 'port = 5433' >> /var/lib/postgresql/12/main2/postgresql.auto.conf
```

или

```
echo 'port = 5433' | sudo tee --append /var/lib/postgresql/12/main2/postgresql.auto.conf
```

Стартуем кластер

```
$pg_ctlcluster 12 main2 start
```

Смотрим как стартовал

```
$pg_lsclusters
```

# Архив журналов

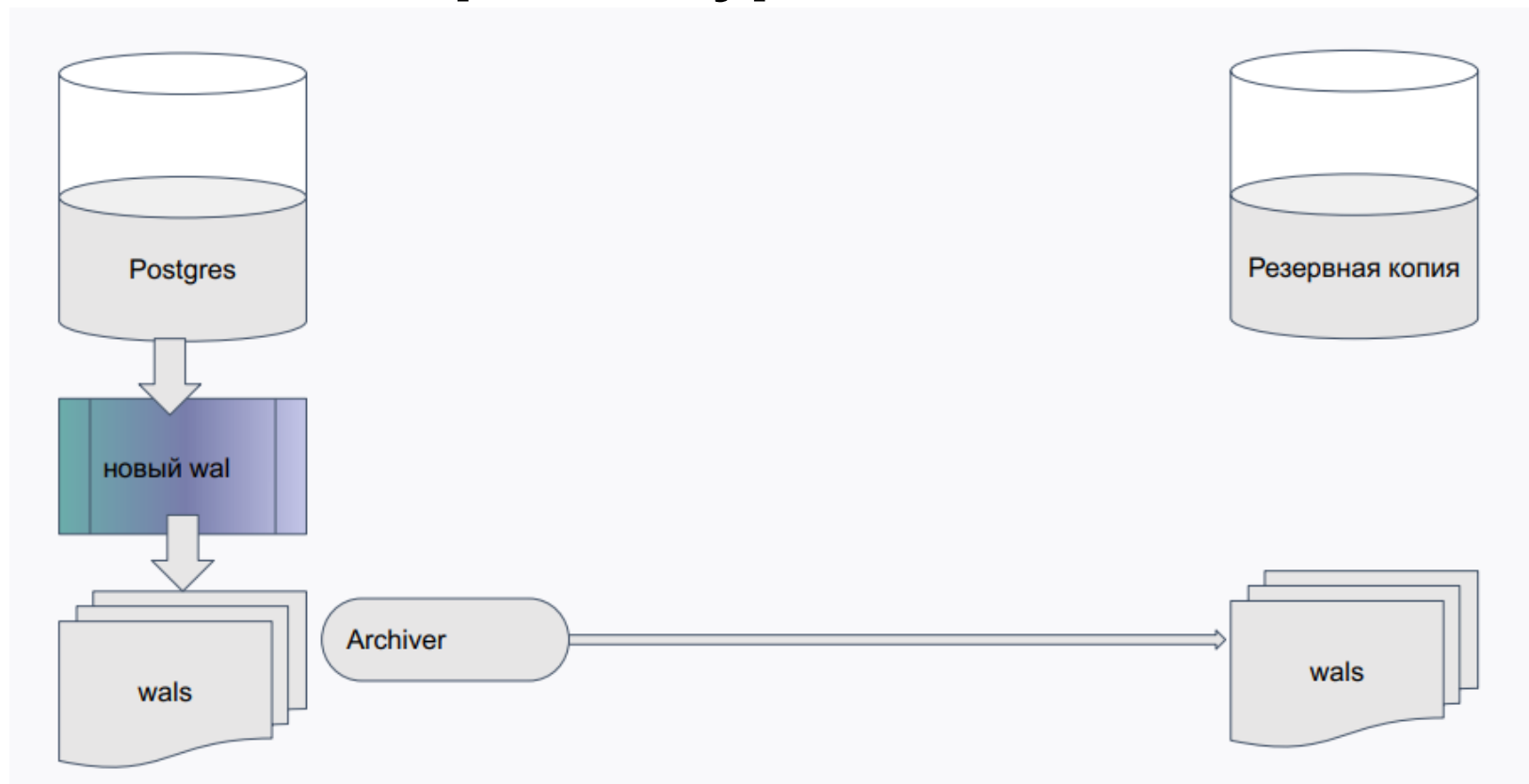
## Файловый архив

- сегменты WAL копируются в архив по мере заполнения
- механизм работает под управлением сервера
- неизбежны задержки попадания данных в архив

## Потоковый архив

- в архив постоянно записывается поток журнальных записей
- требуются внешние средства
- задержки минимальны

# Файловый архив журналов



# Файловый архив журналов

## Процесс archiver

### Параметры

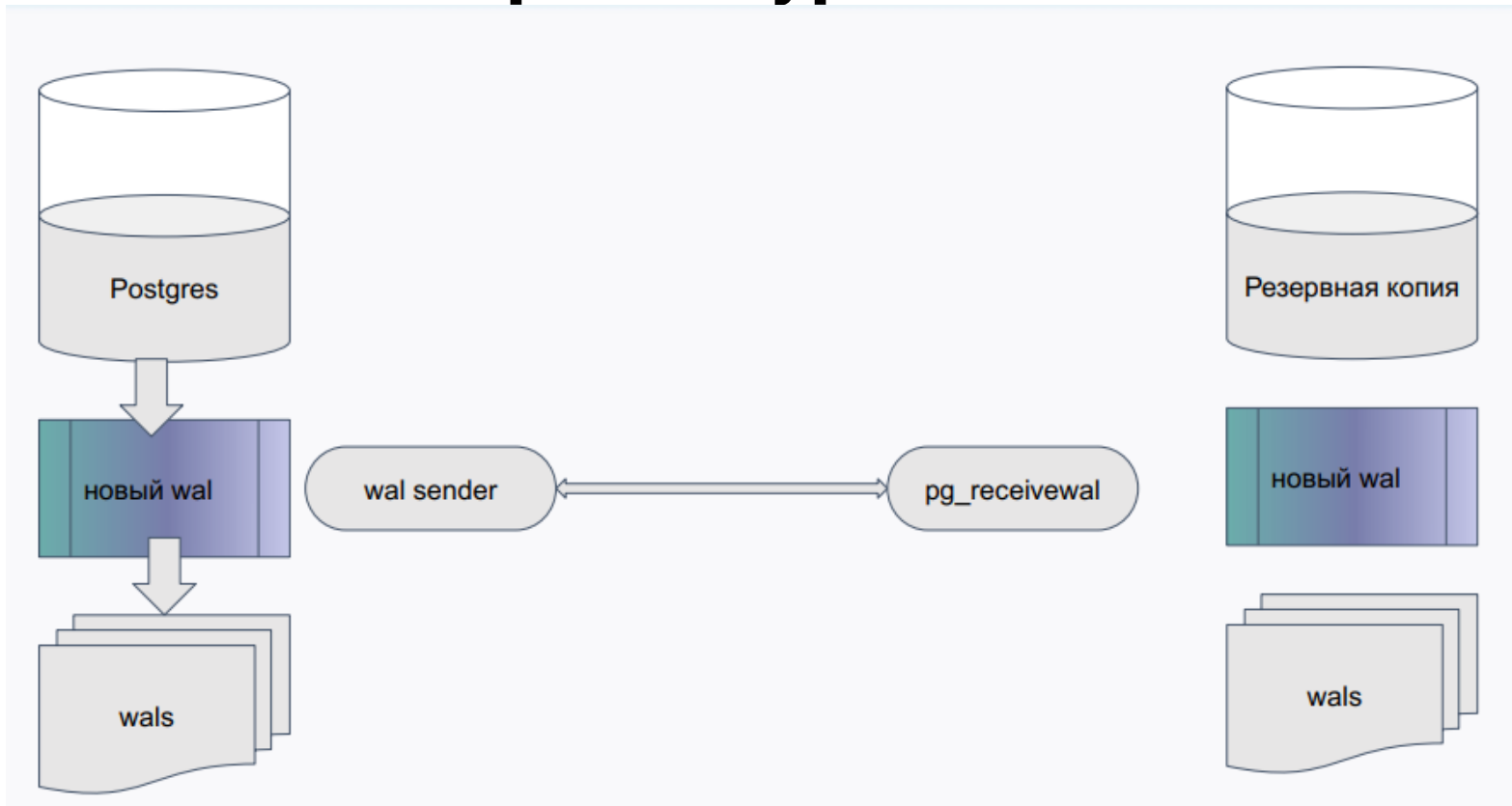
**SELECT name, setting FROM pg\_settings WHERE name IN ('archive\_mode','archive\_command','archive\_timeout');**

- ALTER SYSTEM SET archive\_mode = on
- ALTER SYSTEM SET archive\_command - команда shell для копирования сегмента WAL в отдельное хранилище
- ALTER SYSTEM SET archive\_timeout - максимальное время для переключения на новый сегмент WAL
- требуется рестарт сервера

### Алгоритм

- при заполнении сегмента WAL вызывается команда archive\_command
- если команда завершается со статусом 0, сегмент удаляется
- если команда возвращает не 0 (в частности, если команда не задана), сегмент остается до тех пор, пока попытка не будет успешной

# Потоковый архив журналов



# Потоковый архив журналов

## Утилита `pg_receivewal`

- подключается по протоколу репликации (можно использовать слот)
- направляет поток записей WAL в файлы-сегменты
- стартовая позиция — начало сегмента, следующего за последним заполненным сегментом, найденным в каталоге,
- или начало текущего сегмента сервера, если каталог пустой
- в отличие от файлового архива, записи пишутся постоянно
- при переходе на новый сервер надо перенастраивать параметры

# Еще варианты ...

- **barman** - [Многоярусный бэкап PostgreSQL с помощью Barman и синхронного переноса журналов транзакций / Блог компании ЮMoney / Хабр \(habr.com\)](#)
- **wal-e** - [Разгоняем бэкап. Лекция Яндекса](#)
- **wal-g** - [WAL-G: бэкапы и восстановление СУБД PostgreSQL](#)
- **BART**
- **pg\_probackup**
- ...

# Рефлексия



1. Отметьте, что запомнились с вебинара
2. Что будете применять в работе из сегодняшнего вебинара



**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Коробков Виктор**