



Администратор Linux

PostgreSQL: BackUp + Репликация



Проверить, идет ли запись

Меня хорошо видно && слышно?



Ставим "+", если все хорошо
"-", если есть проблемы



Тема вебинара

PostgreSQL: BackUp + Репликация



Федоров Иван Романович

Технический директор ГК "Илотех"

Опыт:

Более 10 лет в IT-сфере

Аспирант университета ИТМО по направлению "Информационная безопасность"

Многократный победитель различных конкурсов и хакатонов (команда IBI Solutions)

Эл. почта: ifedorov.devops@gmail.com



Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в группе Telegram



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу



Маршрут вебинара

Знакомство

Репликация: теория

Репликация: практика

Бэкапы: теория

Бэкапы: практика

Рефлексия

Цели вебинара

К концу занятия вы сможете

1. Настраивать репликацию в PostgreSQL



2. Настраивать резервное копирование в PostgreSQL с помощью Barman



3. Восстанавливать базу в PostgreSQL из резервной копии



Цели вебинара

К концу занятия вы сможете

1. Настраивать репликацию в PostgreSQL



-
2. Настраивать резервное копирование в PostgreSQL с помощью Barman

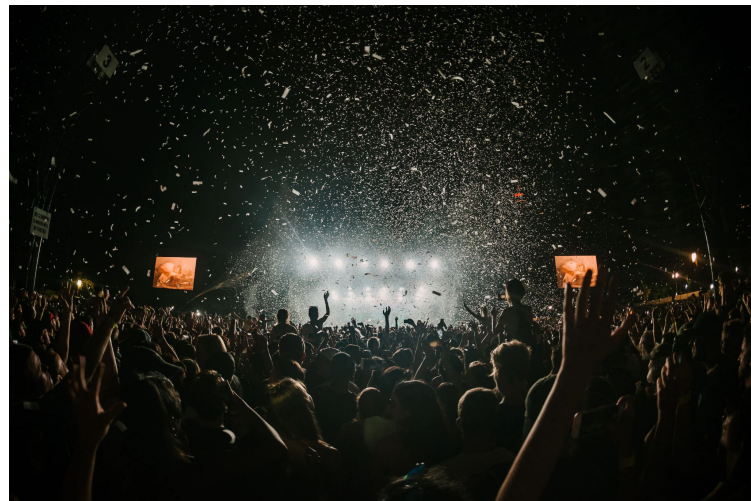


-
3. Восстанавливать базу в PostgreSQL из резервной копии



СМЫСЛ

Зачем вам это уметь



Немного теории...

Журнал предзаписи (WAL)

Проблема: при сбое теряются данные из оперативной памяти, не записанные на диск

Журнал

- поток информации о выполняемых действиях, позволяющий повторно выполнить потерянные при сбое операции
- запись попадает на диск раньше, чем измененные данные

Журнал защищает

- страницы таблиц, индексов и других объектов
- статус транзакций (clog)

Журнал не защищает

- временные и нежурналируемые таблицы

Репликация: цели и задачи

Репликация — процесс синхронизации нескольких копий кластера баз данных на разных серверах (дублирование)

Зачем?

- для повышения отказоустойчивости
- для повышения производительности

Потоковая репликация

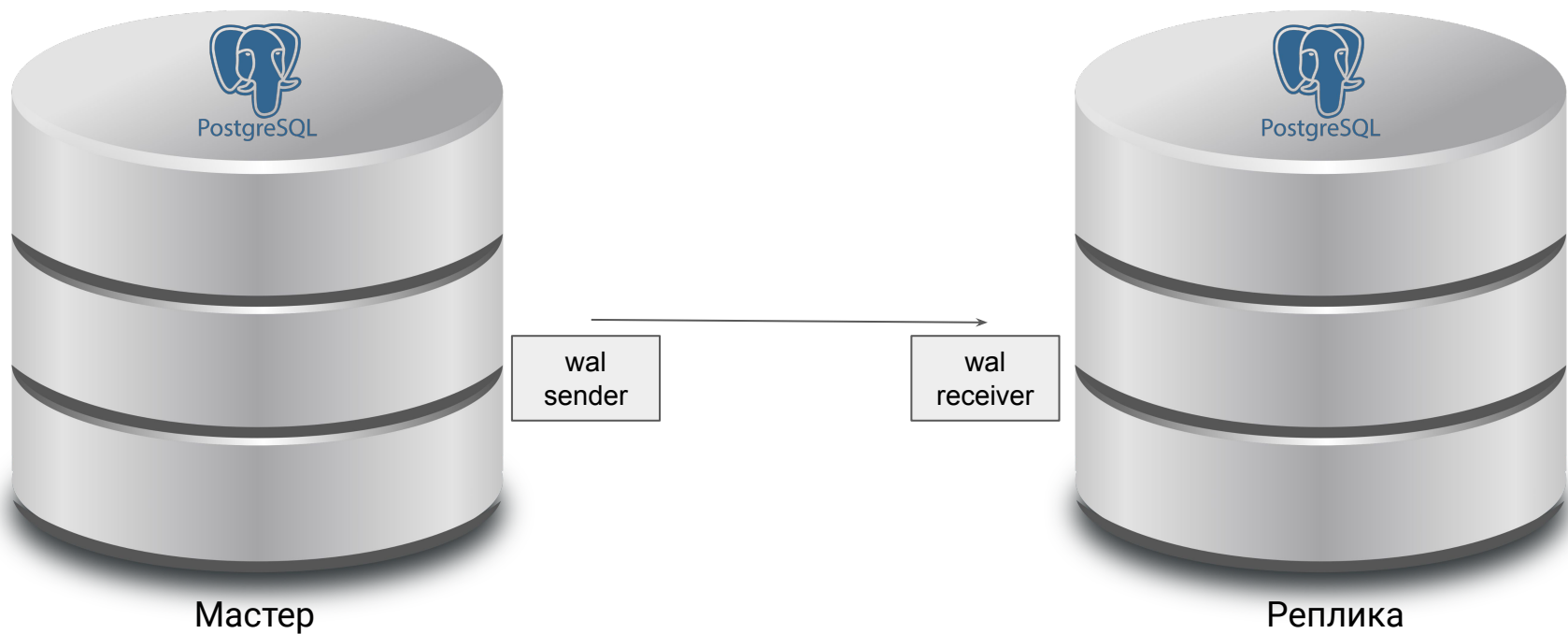
Особенности

- От основного сервера PostgreSQL на реплики передается WAL
- Все серверы должны быть одной версии, работать на одной ОС и архитектуре
- Реплицируется кластер целиком

Режимы

- Асинхронный
- Синхронный

Мастер -> Реплика



Логическая репликация

Особенности

- Механизм публикации/подписки
- Подписываться можно не на все изменения, а выборочно
- Может работать между разными версиями PostgreSQL, ОС и архитектурами
- Нельзя реплицировать изменения структуры БД

Когда используют потоковую репликацию

- Горячий резерв для высокой доступности
- Балансировка нагрузки
- Отложенная репликация

Когда используют логическую репликацию

- Консолидация и общие справочники
- Обновления сервера

**Все хорошо?
Есть ли вопросы?**

Практика: Настройка репликации

Ноды 1, 2 – установка PostgreSQL

```
# Установка версии 14
```

```
$ yum -y update
```

```
$ yum -y install
```

```
https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
; yum list -y postgres*-server*
```

```
$ yum install postgresql14-server.x86_64
```

```
# Инициализация кластера (можно только на ноде 1)
```

```
$ /usr/pgsql-14/bin/postgresql-14-setup initdb
```

```
# Запуск сервиса
```

```
$ systemctl enable postgresql-14
```

```
$ systemctl start postgresql-14
```

```
$ systemctl status postgresql-14
```



Нода 1 — конфигурация

```
# Создание пользователя с правами репликации
$ su - postgres
$ psql
postgres=# create user replicator with replication encrypted password '12345';

# Редактирование postgresql.conf
listen_addresses = 'localhost, <IP master>'
hot_standby = on
wal_level = replica
max_wal_senders = 3
max_replication_slots = 3
hot_standby_feedback = on

# Редактирование pg_hba.conf
host      replication replicator      <IP master>/32      scram-sha-256
host      replication replicator      <IP slave>/32      scram-sha-256

# Перезапуск сервиса
$ systemctl restart postgresql-14
```



Нода 2 — конфигурация

```
# Остановка сервиса и очистка данных кластера (если он был проинициализирован)
$ systemctl stop postgresql-14
$ cd /var/lib/pgsql/14/data
$ rm -rf *

# Копирование данных с ноды 1
$ pg_basebackup -h <IP master> -U replicator -p 5432 -D /var/lib/pgsql/14/data/ -R -P

# Редактирование postgresql.conf
listen_addresses = 'localhost, <IP slave>'

# Запуск сервиса
$ systemctl start postgresql-14
```



Проверка репликации

```
# НА НОДЕ 1
$ su - postgres
$ psql
postgres=# create database test;
```

```
# НА НОДЕ 2
$ su - postgres
$ psql
postgres=# \l
```

Бэкапы PostgreSQL

Инструменты для резервного копирования

Ассортимент

- pg_dump/pg_restore
- pg_basebackup
- wal-e, wal-g
- barman
- pg_probackup
- BART (Backup And Recovery Tool)



На что обратить внимание при выборе инструмента

Важные составляющие:

- Хранение данных
- Разделение данных
- Версионность БД
- Режимы работы
- Сервисность
- Валидирование



Хранение данных

- Логическая копия данных — `pg_dump`, `pg_dumpall`
- Бинарная копия — все остальные
- Управление табличными пространствами — `wal-e`, `pg_basebackup`, `pg_probackup`, `barman`

Разделение данных

- Только схема — `pg_dump`, `pg_dumpall`
- Таблица — `pg_dump`
- Слияние баз — `pg_dump`
- Разделение баз — `pg_dump`

Версионность БД

- Кроссверсионность — pg_dump, pg_dumpall
- Мультиверсионность — все, кроме встроенных
- Создание реплик, standby — все, кроме pg_dump, pg_dumpall

Режимы работы

- SSH (rsync, scp) — barman
- Порт 5432 — все остальные + barman
- Бэкап с реплики — все
- Многопоточный dump — все, кроме pg_basebackup и BART

Режимы работы

- PITR — все, кроме pg_dump, pg_dumpall
- Регулировка нагрузки на сеть — pg_basebackup, barman, BART
- Сжатие на лету — pg_dump, pg_dumpall, barman, pg_probackup, wal-e, wal-g

Сервисность

- CLI — все, кроме встроенных
- Выделенный сервер — barman, BART
- Структурированное хранение бэкапов — все, кроме встроенных
- Политики хранения — barman, pg_probackup, BART

Валидирование

- CLI для мониторинга — все, кроме встроенных
- Валидность завершения бэкапа — barman, wal-e, wal-g, pg_probackup, BART
- checksum postgresql — pg_probackup

Практика: Настройка бэкапа с помощью Barman

Нода 1 — подготовка

```
# Установка barman-cli
$ yum -y install epel-release
$ yum -y install barman-cli

# Генерация ключей и создание пользователя barman
$ su - postgres
$ ssh-keygen -t rsa -b 4096
$ psql
postgres=# create user barman with replication encrypted password '12345';

# Добавление публичного ключа для подключения с сервера barman (должен быть сгенерирован на сервере barman)
$ su - postgres
$ vim ~/.ssh/authorized_keys

# Редактирование pg_hba.conf
host      all            barman          <IP barman>/32      scram-sha-256
host      replication    barman          <IP barman>/32      scram-sha-256

# Рестарт сервиса
$ systemctl restart postgresql-14
```



Сервер Barman — подготовка

```
# Установка barman
$ yum -y update
$ yum -y install epel-release
$ yum -y install
https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noa
rch.rpm; yum list -y postgre*-server*
$ yum install postgresql14 barman-cli barman

# Генерация ключей и добавление публичного ключа для подключения с ноды 1
$ su - barman
$ ssh-keygen -t rsa -b 4096
$ vim ~/.ssh/authorized_keys

# Сохраняем реквизиты подключения к ноде 1
$ echo '<IP node 1>:5432:*:barman:12345' > ~/.pgpass
$ chmod 600 ~/.pgpass

# Проверка подключения
$ psql -h <IP node 1> -U barman -d postgres
```

Сервер Barman — /etc/barman.conf

```
[barman]
barman_home = /var/lib/barman
configuration_files_directory = /etc/barman.d
barman_user = barman
log_file = /var/log/barman/barman.log
compression = gzip
backup_method = rsync
archiver = on
retention_policy = REDUNDANCY 3
immediate_checkpoint = true
last_backup_maximum_age = 4 DAYS
minimum_redundancy = 1
```

Сервер Barman — /etc/barman.d/node1.conf

```
[node1]
description = "backup node1"
ssh_command = ssh postgres@<IP node 1>
conninfo = host=<IP node 1> user=barman port=5432 dbname=postgres
retention_policy_mode = auto
retention_policy = RECOVERY WINDOW OF 7 days
wal_retention_policy = main
streaming_archiver=on
path_prefix = /usr/pgsql-14/bin/
create_slot = auto
slot_name = node1
streaming_conninfo = host=<IP node 1> user=barman
backup_method = postgres
archiver = off
```

Сервер Barman — проверка

Проверка

```
$ barman switch-wal node1
```

```
$ barman cron
```

```
$ barman check node1
```

Создание резервной копии

```
$ barman backup node1
```

Список резервных копий

```
$ barman list-backup node1
```

Восстановление из резервной копии

```
$ barman recover node1 20221008T010731 /var/lib/pgsql/14/data/ --remote-ssh-command "ssh postgres@<IP node 1>"
```

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет



Рефлексия



Цели вебинара

К концу занятия вы сможете

1. Настраивать репликацию в PostgreSQL



2. Настраивать резервное копирование в PostgreSQL с помощью Barman



3. Восстанавливать базу в PostgreSQL из резервной копии



Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то, что узнали на вебинаре?

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