ЛР4. PostgreSQL. Резервное копирование. Репликация

Настройка репликации в PostgreSQL

Рассмотрим кратко процесс настройки физической репликации. Пусть серверы расположены на одной системе и имеют разные порты: 5343 – порт главного сервера и 5344 – порт ведомого сервера.

1. Для каждого сервера создать отдельную папку для хранения данных.
2. Выполнить инициализация директории главного сервера с помощью утилиты `initdb`:

```
initdb -D master_path -U username –W
```

3. Создать папку для хранения журналов предзаписи. Файлы данных журналов используются в качестве инкрементных резервных копий. Настроить архивацию журналов предзаписи в конфигурационном файле сервера `postgresql.conf` (находится в корневой папке директории сервера):

```
archive_mode = ON;
archive_command = 'copy "%p" "PATH\\%f"' # Пример команды архивации для операционной системы Windows
archive_timeout = 60 # время существования не архивированных данных в секундах
```

4. Запустить главный сервер с помощью утилиты `pg_ctl`:

```
pg_ctl start -D master_path -o "-p 5343 "
```

5. Создать пользователя для осуществления репликации:

```
CREATE USER replicator WITH REPLICATION PASSWORD 'password';
```

6. Создать копию главного сервера с помощью утилиты pg_basebackup

```
pg_basebackup -p master_port -U replicator -D path_to_slave -Fp -Xs -P -R
```

7. Запустить ведомый сервер и проверить статус репликации на главном сервере:

```
select * from pg_stat_replication;
```

Восстановление сервера из резервной копии

Для восстановления данных из полной резервной копии (полученной с помощью утилиты `pg_basebackup`) и инкрементных копий (файлы журналов предзаписи) следует выполнить следующее:

1. Удалить все содержимое директории сервера, на котором будет выполняться восстановление данных.

2. Скопировать содержимое папки, в которую была выполнена полная резервная копия, в директорию данных сервера.
3. Для восстановления данных из инкрементных копий в конфигурационном файле главного и ведомого серверов написать команду восстановления из архива WAL, например (для Windows):

```
restore_command = 'copy "PATH\\%f" "%p"'
```

   Для восстановления данных до заданного момента времени используется параметр `recovery_target_time`.
4. Создать пустой файл в директории сервера с названием: "`recovery.signal`".
5. Запустить сервер, процесс восстановления начнется автоматически.

Теоретическая (тестовая) часть

Лекция 5. Резервное копирование [Общая часть]

Лекция 6. Резервное копирование [PostgreSQL]

Лекция 7. Репликация [Общая часть + PostgreSQL]

Практическая часть

1. Выполнить настройку физической репликации БД на двух серверах PostgreSQL.
2. Проверить работу репликации:
   2.1. Изменить данные на главном сервере и убедиться, что данные на ведомом сервере также изменились.
   2.2. Выключить ведомый сервер, изменить данные на главном сервере, включить ведомый сервер и убедиться, что данные на ведомом сервере были так же изменены.
3. Настроить и проверить инкрементное резервное копирование БД.
   3.1. Выполнить полное резервное копирование главного сервера. Настроить инкрементное резервное копирование.
   3.2. Внести некоторые изменения в БД.
   3.3. Сымитировать ситуацию, при которой на главном сервере были потеряны все данные (все содержимое директории сервера удалено).
   3.4. Восстановить потерянные данные из инкрементной резервной копии (не используя данные ведомого сервера).
   3.5. Восстановить процесс репликации.

Бонусная часть (1 балл)

Написать Python-скрипт автоматической проверки состояния репликации: отображение статуса репликации и выполнение пунктов 2.1 и 2.2 задания с отображением результат выполнения (успех/ошибка).