Руководство пользователя: pg_cron расширение

Центр разработки PostgreSQL

Exported on 05/29/2020

Table of Contents

В состав PostgreSQL Sber Edition входит pg_cron расширение (https://github.com/citusdata/pg_cron), которое позволяет самой СУБД выполнять задания по расписанию и может синхронизировать задания на Standby.

1 Работа внутри кластера

pg_cron можно безопасно использовать в схеме с реализацией. Данные cron можно модифицировать только с текущего Leader.

> **Важно:** Так как pg_cron для выполнения задач сохраняет host:ip сервера (127.0.0.1:5432), необходимо на всех экземплярах выставить одинаковый порт. В противном случае, при switchover/failover pg_cron попытается подключиться к старому лидеру. Схема репликации PostgreSQL SE гарантирует совпадение портов, таким образом, при развороте PostgreSQL SE кластера проблемы не возникает.

Рекомендовано использовать для задач

1. вызов VACUUM
2. секционирование таблиц

Настройка PostgreSQL SE для работы с pg_cron

Для использования расширения нужно в postgresql.conf прописать:

```
shared_preload_libraries = 'pg_cron'
cron.database_name = 'postgres' -- имя БД, в которой будет работать cron.
```

Создание задачи в pg_cron

1. Добавить пользователя cron, включить у него расширение
   ```
   CREATE_EXTENSION pg_cron;
   ```
2. Создать пользователя cronuser
3. Разрешить использование cron новому пользователю:
   ```
   GRANT USAGE ON SCHEMA cron TO cronuser;
   ```
4. Добавить в .pgpass имя и пароль пользователя
5. Создать таблицу cron_test:
   ```
   CREATE TABLE cron_test(data varchar(20));
   ```
6. Создать задачу в cron, с периодом выполнения - раз в минуту:
   ```
   select cron.schedule('* * * * *', $$insert into cron_test values('task done!');$
   $);
   ```
7. Вывести содержимое таблицы cron.job чтобы проверить что задача создана:
   ```
   select * from cron.job;
   ```

Удаление задачи из pg_cron

1. Вывести данные таблицы cron.job:
   ```
   select * from cron.job;
   ```
2. По таблице найти jobid задачи, который необходимо удалить
3. Выполнить:

```
select cron.unschedule(1);
```

4. Вывести содержимое таблицы cron.job

Примеры использования

- Удалить старые данные в субботу, 3:30 ночи:

  SELECT cron.schedule('30 3 * * 6', $$DELETE FROM events WHERE event_time < now() - interval '1 week'$$);

- Выполнять VACUUM каждый день в 10:00 утра:

  SELECT cron.schedule('0 10 * * *', 'VACUUM');

- Посмотреть текущие задачи:

  SELECT * FROM cron.job;

- Отменить задачу:

  SELECT cron.unschedule(43);

2 Руководство для сотрудника сопровождения

Реализованная функциональность не содержит специальных инструкций для сотрудника сопровождения.

3 Руководство для администратора безопасности

Реализованная функциональность не содержит специальных инструкций для администратора безопасности.

4 Руководство для разработчика прикладных сервисов

Реализованная функциональность не содержит специальных инструкций для разработчика прикладных сервисов.