PostgresPro

Новая грань pg_dump

Картышов Иван

Мудрость приходит с годами

...и седыми волосами

Сисадмины делятся на тех кто уже делает бэкапы и тех, кто пока еще нет ©народная мудрость

Зачем нужен дамп?

По прямому назначению:

- Сделать дамп всего инстанса
- Сделать дамп базы или только её схемы
- Сделать дамп таблицы или данных этой таблицы

По другому:

- Проверить целостность БД или инстанса
  - pg_dumpall -h /var/run/postgresql >/dev/null
- Накатить быстро реплику для standby - разворачиваем её из последнего дампа и догоняем master. Или так можно догнать безнадежно отставшую реплику.
- Миграция между версиями
  - pg_dumpall -p 5432 -U myuser91 | psql -U myuser94 -d postgres -p 5434
- Копирование базы данных PostgreSQL на другой сервер
  - pg_dump -C dbname | ssh -C remoteuser@remotehost "psql dbname"

Как нам это улучшить?

Рамер БД 500ГБ

~3000 тригерров и тригеррных функций

~5000 схем

~150000 функций рассованных по данным схемам

~200000 таблиц и индексов и

~500000 policies

Задача:

А есть необходимость восстановить БД без таблицы и целой кучей разных объектов (функций, триггеров, политик и тд).

Вывод:

1) Научить dump/restore фильтровать и гибко включать или выключать объекты из дампа на этапе снятия дампа или его восстановления.

2) Сделать это на базе регулярных выражений

3) PROFIT

Доработка dump/restore

В чем сложность доработки таких утилит как pg_dump?

1) 30тыс. строк исторически сложившегося кода

2) большие изменения версий ПГ влечет требование создание кода который версионнонезвисим, в рамках версий 9.6-10 это одно, а если с 7.4-12 то это совершенно другая задача.

3) Большая часть кода датируется 2004-2005 годом

DUMP RESTORE --include-*

--include-table=[KIND]:PATTERN	Выгрузить heap (обычные, секц, fdw) согласно шаблону KIND:[TABLE],[VIEW],[INDEX],[MAT_VIEW],[SEQUENCE]
--include-extension=PATTERN	Выгрузить расширения согласно PATTERN
--include-function=[KIND]:PATTERN	Выгрузить функции согласно шаблону KIND:[NORMAL],[AGG],[PROC],[TRIGGER],[WINDOW]
--include-policy=PATTERN	Выгрузить политики согласно PATTERN
--include-trigger=PATTERN	Выгрузить триггеры согласно PATTERN
--include-type=PATTERN	Выгрузить типы согласно PATTERN

DUMP RESTORE --exclude-*

--exclude-table=[KIND]:PATTERN

--exclude-extension=PATTERN

--exclude-function=[KIND]:PATTERN

--exclude-policy=PATTERN

--exclude-table-data=PATTERN

--exclude-trigger=PATTERN

--exclude-type=PATTERN

Выгрузить heap (обычные, секц, fdw) за исключением PATTERN

KIND:[TABLE],[VIEW],[INDEX],[MAT_VIEW],[SEQUENCE]

Выгрузить расширений за исключением PATTERN

Выгрузить функций за исключением PATTERN

KIND:[NORMAL],[AGG],[PROC],[TRIGGER],[WINDOW]

Выгрузить политик за исключением PATTERN

Не выгружать содержимое heap PATTERN (dump only)

Выгрузить триггеры за исключением PATTERN

Выгрузить типы за исключением PATTERN

DUMP и RESTORE --include-* --exclude-*

```
pg_dump --include-table="[TABLE],[VIEW]:test*|fir*|sec*|main*|ultimate"
        --include-function="[PROC],[NORMAL]:*dat*"
        --include-function="[TRIGGER]:*main"
        --include-function="[WINDOW]:*temptbl*"
        --exclude-function="dup*" --exclude-extension="dblink"
        --exclude-policy="pol1"     --include-type="float8_range"
        --exclude-table-data="fir*|sec*|main*" mydb > dump.sql

pg_restore --include-table="[TABLE]:tabl1|tabl2|tabl3|tabl4|tabl5|tabl6"
         --exclude-table="[TABLE],[VIEW]:test*|fir*|sec*"
         --exclude-function="[PROC],[NORMAL]:*main*"
         --schema="sch1" --to-schema="sch2"
         --exclude-type="user*|sysadm*"
         --include-function="[PROC]:*data" -d newdb dbdump
```

Подменяем схему на лету

--to-schema B

my_schema1.table(...)

my_schema1.type(...)

my_schema1.function(...)

my_schema1.matview(...)

pg_restore -n my_schema1 --to-schema my_schema2

my_schema2.table(...)

my_schema2.type(...)

my_schema2.function(...)

my_schema2.matview(...)

pg_restore --include-table="[TABLE]:*"
--schema="sch1" --to-schema="sch2"
--exclude-table="t2_*"
--include-function="[PROC]:*data" -d newdb dbdump

PostgresPro

Спасибо за внимание

Любите/пользуйтесь pg_dump
пишите нам:

Свои кейсы использования

Боль и проблемы с ванилой

Идеи улучшения и доработок

117036, г. Москва, ул. Дмитрия Ульянова, д. 7А

+7(495)150-06-91

i.kartyshov@postgrespro

www.postgrespro.ru

Roadmap

Снять ограничения использовани --to-schema
Закоммитить сделанные изменения в ванилу