Приложение 5

Логическое резервное копирование

Логическое резервное копирование можно выполнить утилитой `pg_dump` для копирования конкретной базы данных на сервере или утилитой команды `pg_dumpall` для копирования всего кластера целиком.

Ввод [1]:

```
# Видим что у нас всего одна база данных postgres
# за исключением двух шаблонов template0 template1

!psql -U postgres -c "\l"
```

```
                                                 List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU
Locale | Locale Provider |   Access privileges
-----------+----------+----------+-------------+-------------+-----
-------+-----------------+-----------------------
 postgres  | postgres | UTF8     | en_SG.UTF-8 | en_SG.UTF-8 |
| libc            |
 template0 | postgres | UTF8     | en_SG.UTF-8 | en_SG.UTF-8 |
| libc            | =c/postgres          +
           |          |          |             |             |
|                 | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_SG.UTF-8 | en_SG.UTF-8 |
| libc            | =c/postgres          +
           |          |          |             |             |
|                 | postgres=CTc/postgres
(3 rows)
```

Ввод [2]:

```
# Создам таблицу test и наполню её данными!

!psql -U postgres -d template1 -c "DROP DATABASE postgres;"
!psql -U postgres -d template1 -c "CREATE DATABASE postgres;"
!psql -U postgres -c "CREATE TABLE test(id integer);"
!psql -U postgres -c "INSERT INTO test VALUES (1), (2), (3)"
!psql -U postgres -c "SELECT * FROM test;"
```

```
DROP DATABASE
CREATE DATABASE
CREATE TABLE
INSERT 0 3
 id
----
  1
  2
  3
(3 rows)
```

Ввод [3]:

```
# Посмотрим справку по резервному копированию pg_dump

!pg_dump --help
```

```
pg_dump dumps a database as a text file or to other formats.

Usage:
  pg_dump [OPTION]... [DBNAME]

General options:
  -f, --file=FILENAME          output file or directory name
  -F, --format=c|d|t|p         output file format (custom, director
y, tar,
                               plain text (default))
  -j, --jobs=NUM               use this many parallel jobs to dump
  -v, --verbose                verbose mode
  -V, --version                output version information, then exi
t
  -Z, --compress=0-9           compression level for compressed for
mats
  --lock-wait-timeout=TIMEOUT  fail after waiting TIMEOUT for a tab
le lock
  --no-sync                    do not wait for changes to be writte
n safely to disk
  -?, --help                   show this help, then exit

Options controlling the output content:
  -a, --data-only              dump only the data, not the schema
  -b, --blobs                  include large objects in dump
  -B, --no-blobs               exclude large objects in dump
  -c, --clean                  clean (drop) database objects before
recreating
  -C, --create                 include commands to create database
in dump
  -e, --extension=PATTERN      dump the specified extension(s) only
  -E, --encoding=ENCODING      dump the data in encoding ENCODING
  -n, --schema=PATTERN         dump the specified schema(s) only
  -N, --exclude-schema=PATTERN do NOT dump the specified schema(s)
  -O, --no-owner               skip restoration of object ownership
in
                               plain-text format
  -s, --schema-only            dump only the schema, no data
  -S, --superuser=NAME         superuser user name to use in plain-
text format
  -t, --table=PATTERN          dump the specified table(s) only
  -T, --exclude-table=PATTERN  do NOT dump the specified table(s)
  -x, --no-privileges          do not dump privileges (grant/revok
e)
  --binary-upgrade             for use by upgrade utilities only
  --column-inserts             dump data as INSERT commands with co
lumn names
  --disable-dollar-quoting     disable dollar quoting, use SQL stan
dard quoting
  --disable-triggers           disable triggers during data-only re
store
  --enable-row-security        enable row security (dump only conte
nt user has
                               access to)
  --exclude-table-data=PATTERN do NOT dump data for the specified t
able(s)
  --extra-float-digits=NUM     override default setting for extra_f
loat_digits
  --if-exists                  use IF EXISTS when dropping objects
  --include-foreign-data=PATTERN
                               include data of foreign tables on fo
```

```
reign
                                servers matching PATTERN
  --inserts                     dump data as INSERT commands, rather
than COPY
  --load-via-partition-root     load partitions via the root table
  --no-comments                 do not dump comments
  --no-publications             do not dump publications
  --no-security-labels          do not dump security label assignmen
ts
  --no-subscriptions            do not dump subscriptions
  --no-table-access-method      do not dump table access methods
  --no-tablespaces              do not dump tablespace assignments
  --no-toast-compression        do not dump TOAST compression method
s
  --no-unlogged-table-data      do not dump unlogged table data
  --on-conflict-do-nothing      add ON CONFLICT DO NOTHING to INSERT
commands
  --quote-all-identifiers       quote all identifiers, even if not k
ey words
  --rows-per-insert=NROWS       number of rows per INSERT; implies -
-inserts
  --section=SECTION             dump named section (pre-data, data,
or post-data)
  --serializable-deferrable     wait until the dump can run without
anomalies
  --snapshot=SNAPSHOT           use given snapshot for the dump
  --strict-names                require table and/or schema include
patterns to
                                match at least one entity each
  --use-set-session-authorization
                                use SET SESSION AUTHORIZATION comman
ds instead of
                                ALTER OWNER commands to set ownershi
p

Connection options:
  -d, --dbname=DBNAME       database to dump
  -h, --host=HOSTNAME       database server host or socket directory
  -p, --port=PORT           database server port number
  -U, --username=NAME       connect as specified database user
  -w, --no-password         never prompt for password
  -W, --password            force password prompt (should happen aut
omatically)
  --role=ROLENAME           do SET ROLE before dump

If no database name is supplied, then the PGDATABASE environment
variable value is used.

Report bugs to <pgsql-bugs@lists.postgresql.org>.
PostgreSQL home page: <https://www.postgresql.org/>
```

Ввод [4]:

```
# Выполним резервное копирование и
# сохраним в файлик dump.txt

!pg_dump -U postgres -d postgres -f dump.txt
!ls -l dump.txt
```

```
-rw-r--r-- 1 evgeny evgeny 892 Jun  8 22:07 dump.txt
```

Ввод [5]:

```
# прочитаем пару первых строк файла

!cat dump.txt
```

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 15.6 (Debian 15.6-0+deb12u1)
-- Dumped by pg_dump version 15.6 (Debian 15.6-0+deb12u1)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: test; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.test (
    id integer
);


ALTER TABLE public.test OWNER TO postgres;

--
-- Data for Name: test; Type: TABLE DATA; Schema: public; Owner: po
stgres
--

COPY public.test (id) FROM stdin;
1
2
3
\.


--
-- PostgreSQL database dump complete
--
```

Резервное копирование выполнено. Попрактикуюсь в восстановлении базы данных из резервной копии.

Ввод [6]:

```
# Сначала дропну целевую базу данных и создам чистую

!psql -U postgres -d template1 -c "DROP DATABASE postgres;"
!psql -U postgres -d template1 -c "CREATE DATABASE postgres;"
```

```
DROP DATABASE
CREATE DATABASE
```

Ввод [7]:

```
# А теперь восстановлю из дампа

!cat dump.txt | psql -U postgres -d postgres
```

```
SET
SET
SET
SET
SET
 set_config
------------

(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
COPY 3
```

Ввод [8]:

```
# Убедимся что данные восстановились

!psql -U postgres -c "SELECT * FROM test;"
```

```
 id
----
  1
  2
  3
(3 rows)
```

Репликация

Ввод [10]:
```
# Настройки по умолчанию позволяют
# использовать протокол репликации:

# Посмотрим настройки на нашем сервере СУБД

!psql -U postgres -c "SHOW wal_level;"
!psql -U postgres -c "SHOW max_wal_senders;"
```

```
 wal_level
-----------
 replica
(1 row)

 max_wal_senders
-----------------
 10
(1 row)
```

Ввод [11]:
```
# Посмотрим позволяют ли настройки подключение
# по протоколу репикации

# Выясним где лежит файл pg_hba.conf

!psql -U postgres -c "SHOW hba_file;"
```

```
              hba_file
--------------------------------------
 /etc/postgresql/15/main/pg_hba.conf
(1 row)
```

Ввод [15]:
```
# Посмотрим его параметры

!echo 314159 | sudo -S tail /etc/postgresql/15/main/pg_hba.conf
```

```
[sudo] password for evgeny: local   all             all
trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            scr
am-sha-256
# IPv6 local connections:
host    all             all             ::1/128                 scr
am-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     tru
st
host    replication     all             127.0.0.1/32            scr
am-sha-256
host    replication     all             ::1/128                 scr
am-sha-256
```

В самом конце файла видим параметр для локального подключения

Создадим автономную резервную копию. С ключом `-R` утилита сформирует необходимые для репликации конфигурационные параметры.

Ввод [18]:
```
!pg_basebackup -U postgres --pgdata=/home/evgeny/Projects/intern
```

Ввод [20]:
```
# Посмотрим что там

!ls -l /home/evgeny/Projects/internship/basebackup
```

```
total 216
-rw------- 1 evgeny evgeny    225 Jun  8 22:22 backup_label
-rw------- 1 evgeny evgeny 138237 Jun  8 22:22 backup_manifest
drwx------ 5 evgeny evgeny   4096 Jun  8 22:22 base
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 global
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_commit_ts
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_dynshmem
drwx------ 4 evgeny evgeny   4096 Jun  8 22:22 pg_logical
drwx------ 4 evgeny evgeny   4096 Jun  8 22:22 pg_multixact
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_notify
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_replslot
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_serial
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_snapshots
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_stat
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_stat_tmp
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_subtrans
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_tblspc
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_twophase
-rw------- 1 evgeny evgeny      3 Jun  8 22:22 PG_VERSION
drwx------ 3 evgeny evgeny   4096 Jun  8 22:22 pg_wal
drwx------ 2 evgeny evgeny   4096 Jun  8 22:22 pg_xact
-rw------- 1 evgeny evgeny    417 Jun  8 22:22 postgresql.auto.conf
-rw------- 1 evgeny evgeny      0 Jun  8 22:22 standby.signal
```

Ввод [24]:
```
!echo 314159 | sudo -S mv /home/evgeny/Projects/internship/baseb
```

```
[sudo] password for evgeny:
```

Ввод [33]:
```
# Меня владельца файлов на postgres

!echo 314159 | sudo -S chown postgres:postgres -R /var/lib/postg
```

```
[sudo] password for evgeny:
```

Ввод [35]:

```
# Проверяю правльность действий

!echo 314159 | sudo -S ls -l /var/lib/postgresql/15/replica
```

```
[sudo] password for evgeny: total 216
-rw------- 1 postgres postgres    225 Jun  8 22:22 backup_label
-rw------- 1 postgres postgres 138237 Jun  8 22:22 backup_manifest
drwx------ 5 postgres postgres   4096 Jun  8 22:22 base
drwx------ 2 postgres postgres   4096 Jun  8 22:22 global
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_commit_ts
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_dynshmem
drwx------ 4 postgres postgres   4096 Jun  8 22:22 pg_logical
drwx------ 4 postgres postgres   4096 Jun  8 22:22 pg_multixact
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_notify
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_replslot
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_serial
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_snapshots
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_stat
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_stat_tmp
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_subtrans
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_tblspc
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_twophase
-rw------- 1 postgres postgres      3 Jun  8 22:22 PG_VERSION
drwx------ 3 postgres postgres   4096 Jun  8 22:22 pg_wal
drwx------ 2 postgres postgres   4096 Jun  8 22:22 pg_xact
-rw------- 1 postgres postgres    417 Jun  8 22:22 postgresql.auto.
conf
-rw------- 1 postgres postgres      0 Jun  8 22:22 standby.signal
```

Ввод [40]:

```
# Файл конфигурации реплики

!echo 314159 | sudo -S cat /var/lib/postgresql/15/replica/postgr
```

```
[sudo] password for evgeny: # Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
primary_conninfo = 'user=postgres passfile=''/home/evgeny/.pgpass''
channel_binding=prefer host=''/var/run/postgresql'' port=5432 sslmo
de=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_min_proto
col_version=TLSv1.2 gssencmode=prefer krbsrvname=postgres gssdelega
tion=0 target_session_attrs=any load_balance_hosts=disable'
```

Реплика сервера будет доступна сразу после выполнения команды

```
pg_ctlcluster 15 replica start
```

Все изменения на мастер-сервере будут транслироваться на реплику