

SQL dump approach to backing up PostgreSQL data:

The idea behind this dump method is to generate a file with SQL commands that, when fed back to the server, will recreate the database in the same state as it was at the time of the dump. PostgreSQL provides the utility program `pg_dump` & `pg_dumpall` for this purpose.

`pg_dump` utility:

`pg_dump` is a utility for backing up a PostgreSQL database. It makes consistent backups even if the database is being used concurrently. `pg_dump` does not block other users accessing the database (readers or writers).

`pg_dump` only dumps a single database.

Database backup in different formats:

Options:

`-F`: specifies the output file format that can be one of the following:

`c`: custom-format archive file format

`d`: directory-format archive

`t`: tar

`p`: plain-text SQL script file (Default).

`-f`: Send output to the specified file

`-v`: verbose

`pg_dump -U postgres -d db_name -Fp -p 5432 -f /backup_path/backup_filename.sql`

`pg_dump -U postgres -d db_name -Ft -p 5432 -f /backup_path/backup_filename.tar`

`pg_dump -U postgres -d db_name -Fd -p 5432 -f /backup_path/backup_dir`

`pg_dump -U postgres -d db_name -Fc -p 5432 -f /backup_path/backup_filename.dump`

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -f /DB/Backups/demo_db_bkp.sql
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Ft -f /DB/Backups/demo_db_bkp.tar
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fd -f /DB/Backups/demo_db_bkp_dir
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fc -f /DB/Backups/demo_db_bkp.dump
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/
total 68K
-rw-r--r--. 1 postgres postgres 13K Apr 11 17:22 demo_db_bkp.sql
-rw-r--r--. 1 postgres postgres 33K Apr 11 17:22 demo_db_bkp.tar
drwx-----. 2 postgres postgres 135 Apr 11 17:23 demo_db_bkp_dir
-rw-r--r--. 1 postgres postgres 15K Apr 11 17:23 demo_db_bkp.dump
[postgres@ip-172-31-24-211 ~]$
```

Table level backup:

Option:

`-t`: Dump only tables with names matching pattern.

#Single Table backup

`pg_dump -U user_name -d db_name -t table_name -p 5432 -f /backup_path/tablename_bkp.sql`

#Multiple tables backup

```
pg_dump -U user_name -d db_name -t table_name1 -t table_name2 -p 5432 -f
/backup_path/multi_tables_bkp.sql
```

#Tables named with matching pattern backup

```
pg_dump -U user_name -d db_name -t table_name*bkp -p 5432 -f
/backup_path/matching_tables_bkp.sql
```

```
demo_db=# \dt+
              List of relations
Schema |      Name      | Type | Owner  | Persistence | Access method | Size  | Description
-----+-----+-----+-----+-----+-----+-----+-----
public | categories     | table | postgres | permanent   | heap          | 16 kB |
public | customers      | table | postgres | permanent   | heap          | 16 kB |
public | order_details  | table | postgres | permanent   | heap          | 8192 bytes |
public | orders         | table | postgres | permanent   | heap          | 8192 bytes |
public | products       | table | postgres | permanent   | heap          | 16 kB |
public | testproducts   | table | postgres | permanent   | heap          | 8192 bytes |
(6 rows)

demo_db=# █

[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -t categories -f /DB/Backups/single_table_bkp.sql
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -t categories -t customers -t products -f /DB/Backups/multi_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -t order* -f /DB/Backups/matching_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/
total 20K
-rw-r--r--. 1 postgres postgres 2.3K Apr 11 17:35 single_table_bkp.sql
-rw-r--r--. 1 postgres postgres 6.3K Apr 11 17:35 multi_tables_bkp.sql
-rw-r--r--. 1 postgres postgres 4.9K Apr 11 17:35 matching_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$ █
```

Exclude tables backup:

Option:

-T: Do not dump any tables matching pattern.

#Exclude single Table

```
pg_dump -U user_name -d db_name -T table_name -p 5432 -f
/backup_path/exclude_tablename_bkp.sql
```

#Exclude multiple tables

```
pg_dump -U user_name -d db_name -T table_name1 -T table_name2 -p 5432 -f
/backup_path/exclude_multi_tables_bkp.sql
```

#Exclude tables named with matching pattern backup

```
pg_dump -U user_name -d db_name -T table_name*bkp -p 5432 -f
/backup_path/exclude_matching_tables_bkp.sql
```

```
demo_db=# \dt+
              List of relations
Schema |      Name      | Type | Owner  | Persistence | Access method | Size  | Description
-----+-----+-----+-----+-----+-----+-----+-----
public | categories     | table | postgres | permanent   | heap          | 16 kB |
public | customers      | table | postgres | permanent   | heap          | 16 kB |
public | order_details  | table | postgres | permanent   | heap          | 8192 bytes |
public | orders         | table | postgres | permanent   | heap          | 8192 bytes |
public | products       | table | postgres | permanent   | heap          | 16 kB |
public | testproducts   | table | postgres | permanent   | heap          | 8192 bytes |
(6 rows)

demo_db=# █
```

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -T categories -f /DB/Backups/exclude_single_table.sql
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -T categories -T customers -T products -f /DB/Backups/exclude_multi_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -T order* -f /DB/Backups/exclude_matching_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/
total 36K
-rw-r--r--. 1 postgres postgres 12K Apr 11 17:39 exclude_single_table.sql
-rw-r--r--. 1 postgres postgres 8.1K Apr 11 17:40 exclude_multi_tables_bkp.sql
-rw-r--r--. 1 postgres postgres 8.1K Apr 11 17:40 exclude_matching_tables_bkp.sql
[postgres@ip-172-31-24-211 ~]$
```

Schema only backup:

Option:

-s: Dump only the object definitions (schema), not data.

`pg_dump -U user_name -d db_name -s -p 5432 -f /backup_path/Schema_bkp.sql`

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -s -f /DB/Backups/demo_db_schema_bkp.sql
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/
total 8.0K
-rw-r--r--. 1 postgres postgres 7.7K Apr 13 02:57 demo_db_schema_bkp.sql
[postgres@ip-172-31-24-211 ~]$
```

Data only backup:

Option:

-a: Dump only the data, not the schema (data definitions). Table data, large objects, and sequence values are dumped.

`pg_dump -U user_name -d db_name -a -p 5432 -f /backup_path/Data_bkp.sql`

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -a -f /DB/Backups/demo_db_dataonly_bkp.sql
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/
total 8.0K
-rw-r--r--. 1 postgres postgres 5.3K Apr 13 02:58 demo_db_dataonly_bkp.sql
[postgres@ip-172-31-24-211 ~]$
```

Restoring the Dump:

#Text file dumps are restored using the `pg_restore` utility.

`psql -U user_name -d db_name -p 5432 -f /backup_filename/backup_filename.sql`

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fp -f /DB/Backups/demo_db_bkp.sql
[postgres@ip-172-31-24-211 ~]$ psql -U postgres -d restore_db -f /DB/Backups/demo_db_bkp.sql
SET
SET
SET
SET
SET
set_config
-----
(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
```

#Non-text file dumps are restored using the `pg_restore` utility.

`pg_restore -U user_name -d db_name -p 5432 -v /backup_filename/backup_filename.tar`

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -Fc -f /DB/Backups/demo_db_bkp.dump
[postgres@ip-172-31-24-211 ~]$ pg_restore -U postgres -d restore_db -v /DB/Backups/demo_db_bkp.dump
pg_restore: connecting to database for restore
pg_restore: creating TABLE "public.categories"
pg_restore: creating SEQUENCE "public.categories_category_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.categories_category_id_seq"
pg_restore: creating TABLE "public.customers"
pg_restore: creating SEQUENCE "public.customers_customer_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.customers_customer_id_seq"
pg_restore: creating TABLE "public.order_details"
pg_restore: creating SEQUENCE "public.order_details_order_detail_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.order_details_order_detail_id_seq"
pg_restore: creating TABLE "public.orders"
pg_restore: creating SEQUENCE "public.orders_order_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.orders_order_id_seq"
pg_restore: creating TABLE "public.products"
pg_restore: creating SEQUENCE "public.products_product_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.products_product_id_seq"
pg_restore: creating TABLE "public.testproducts"
pg_restore: creating SEQUENCE "public.testproducts_testproduct_id_seq"
pg_restore: creating SEQUENCE OWNED BY "public.testproducts_testproduct_id_seq"
pg_restore: creating DEFAULT "public.categories_category_id"
pg_restore: creating DEFAULT "public.customers_customer_id"
pg_restore: creating DEFAULT "public.order_details_order_detail_id"
```

pg_dumpall utility:

pg_dump dumps only a single database at a time, and it does not dump information about roles or tablespaces (because those are cluster-wide rather than per-database). To support convenient dumping of the entire contents of a database cluster, the pg_dumpall program is provided. pg_dumpall backs up each database in a given cluster, and also preserves cluster-wide data such as role and tablespace definitions.

#Entire cluster backup

pg_dumpall -U user_name -p 5432 -f /backup_path/cluster_backupfile.sql

```
[postgres@ip-172-31-24-211 ~]$ pg_dumpall -U postgres -p 5432 -v -f /DB/Backups/cluster_bkp.sql
pg_dumpall: executing SELECT pg_catalog.set_config('search_path', '', false);
pg_dumpall: executing SELECT oid, rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcanlogin, rolconnlimit, rolpassword, rolvaliduntil, rolreplaction, rolybypassrls, pg_catalog.shobj_description(oid, 'pg_authid') as rolcomment, rolname = current_user AS is_current_user FROM pg_authid WHERE rolname != 'pg.' ORDER BY 2
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid::pg_catalog.regclass' AND objoid = '16384'
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid::pg_catalog.regclass' AND objoid = '10'
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'demo_user')
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'postgres')
pg_dumpall: executing SELECT ur.rolname AS role, um.rolname AS member, ug.oid AS grantorid, ug.rolname AS grantor, a.admin_option, a.inherit_option, a.set_option FROM pg_auth_members a LEFT JOIN pg_authid ur on ur.oid = a.roleid LEFT JOIN pg_authid um on um.oid = a.member LEFT JOIN pg_authid ug on ug.oid = a.grantor WHERE NOT (ur.rolname ~ 'pg.' AND um.rolname ~ 'pg.') ORDER BY 1,2,4
pg_dumpall: executing SELECT parname, pg_catalog.pg_get_userbyid(10) AS parowner, paracl, pg_catalog.aclddefault('p', 10) AS aclddefault FROM pg_catalog.pg_parameter_acl ORDER BY 1
pg_dumpall: executing SELECT oid, spcname, pg_catalog.pg_get_userbyid(spcowner) AS spcowner, pg_catalog.pg_tablespace_location(oid), spcacl, aclddefault('t', spcowner) AS aclddefault, array_to_string(spcoptions, ', '), pg_catalog.shobj_description(oid, 'pg_tablespace') FROM pg_catalog.pg_tablespace WHERE spcname != 'pg.' ORDER BY 1
pg_dumpall: executing SELECT datname FROM pg_database d WHERE datallowconn AND datconnlimit != -2 ORDER BY (datname <> 'template1'), datname
pg_dumpall: dumping database "template1"
pg_dumpall: running ""usr/pgsql-16/bin/pg_dump" -v -f /DB/Backups/cluster_bkp.sql -Fa "port=5432 user=postgres dbname=template1"
pg_dump: last built-in OID is 16383
pg_dump: reading extensions
```

#Dump only global objects

Option:

-g: Dump only global objects (roles and tablespaces), no databases.

pg_dumpall -U user_name -g -p 5432 -f /backup_path/global_objects_bkp.sql

```
[postgres@ip-172-31-24-211 ~]$ pg_dumpall -U postgres -p 5432 -g -v -f /DB/Backups/global_objects_bkp.sql
pg_dumpall: executing SELECT pg_catalog.set_config('search_path', '', false);
pg_dumpall: executing SELECT oid, rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcanlogin, rolconnlimit, rolpassword, rolvaliduntil, rolreplaction, rolybypassrls, pg_catalog.shobj_description(oid, 'pg_authid') as rolcomment, rolname = current_user AS is_current_user FROM pg_authid WHERE rolname != 'pg.' ORDER BY 2
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid::pg_catalog.regclass' AND objoid = '16384'
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid::pg_catalog.regclass' AND objoid = '10'
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'demo_user')
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'postgres')
pg_dumpall: executing SELECT ur.rolname AS role, um.rolname AS member, ug.oid AS grantorid, ug.rolname AS grantor, a.admin_option, a.inherit_option, a.set_option FROM pg_auth_members a LEFT JOIN pg_authid ur on ur.oid = a.roleid LEFT JOIN pg_authid um on um.oid = a.member LEFT JOIN pg_authid ug on ug.oid = a.grantor WHERE NOT (ur.rolname ~ 'pg.' AND um.rolname ~ 'pg.') ORDER BY 1,2,4
pg_dumpall: executing SELECT parname, pg_catalog.pg_get_userbyid(10) AS parowner, paracl, pg_catalog.aclddefault('p', 10) AS aclddefault FROM pg_catalog.pg_parameter_acl ORDER BY 1
pg_dumpall: executing SELECT oid, spcname, pg_catalog.pg_get_userbyid(spcowner) AS spcowner, pg_catalog.pg_tablespace_location(oid), spcacl, aclddefault('t', spcowner) AS aclddefault, array_to_string(spcoptions, ', '), pg_catalog.shobj_description(oid, 'pg_tablespace') FROM pg_catalog.pg_tablespace WHERE spcname != 'pg.' ORDER BY 1
```

#Dump only tablespaces

Option:

-t: Dump only tablespaces, no databases, or roles.

pg_dumpall -U user_name -t -p 5432 -f /backup_path/tablespaces_bkp.sql

```
[postgres@ip-172-31-24-211 ~]$ psql
psql (16.2)
Type "help" for help.

postgres=# select * from pg_tablespace;
   oid   | spcname | spcowner | spcACL | spcoptions
-----+-----+-----+-----+-----
  1663   | pg_default | 10      |      | 
  1664   | pg_global | 10      |      | 
 16594   | tbl_space1 | 10      |      | 
 16595   | tbl_space2 | 10      |      | 
(4 rows)

postgres=# \q
[postgres@ip-172-31-24-211 ~]$ pg_dumpall -U postgres -p 5432 -t -v -f /DB/Backups/tablespace_bkp.sql
pg_dumpall: executing SELECT pg_catalog.set_config('search_path', '', false);
pg_dumpall: executing SELECT oid, spcname, pg_catalog.pg_get_userbyid(spcowner) AS spcowner, pg_catalog.pg_tablespace_location(oid), spcACL, acldefault('t',
spcowner) AS acldefault, array_to_string(spcoptions, ' '), pg_catalog.shobj_description(oid, 'pg_tablespace') FROM pg_catalog.pg_tablespace WHERE spcname !~
'pg_' ORDER BY 1
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_tablespace'::pg_catalog.regclass AND objoid = '165
94'
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_tablespace'::pg_catalog.regclass AND objoid = '165
95'
```

#The resulting dump can be restored with psql:

psql -U user_name -p 5432 -f /backup_path/cluster_backupfile.sql

Recovery steps:

- i) Stop postgresql service and delete existing data directory files.
- ii) Initialize new database cluster and start the postgresql service
- iii) Restore cluster_bkp file using psql

```
[ec2-user@ip-172-31-24-211 ~]$ sudo su
[root@ip-172-31-24-211 ec2-user]# sudo systemctl stop postgresql-16
[root@ip-172-31-24-211 ec2-user]# rm -rf /var/lib/pgsql/16/data/*
[root@ip-172-31-24-211 ec2-user]# sudo /usr/pgsql-16/bin/postgresql-16-setup initdb
Initializing database ... OK

[root@ip-172-31-24-211 ec2-user]# sudo systemctl start postgresql-16
[root@ip-172-31-24-211 ec2-user]# su - postgres
Last login: Sat Apr 13 03:33:25 UTC 2024 on pts/0
[postgres@ip-172-31-24-211 ~]$ psql -U postgres -p 5432 -f /DB/Backups/cluster_bkp.sql
SET
SET
SET
CREATE ROLE
ALTER ROLE
psql:/DB/Backups/cluster_bkp.sql:18: ERROR:  role "postgres" already exists
ALTER ROLE
You are now connected to database "template1" as user "postgres".
SET
SET
SET
SET
SET
set_config
-----
(1 row)
```

Handling Large Databases:

Some operating systems have maximum file size limits that cause problems when creating large pg_dump output files. Fortunately, pg_dump can write to the standard output, so you can use standard Unix tools to work around this potential problem.

#Use compressed dump

pg_dump -U user_name -d db_name -p 5432 |gzip -c > /backup_path/backup_filename.sql.gz

```
[postgres@ip-172-31-24-211 ~]$ pg_dump -U postgres -d demo_db -p 5432 | gzip -c > /DB/Backups/demo_db_bkp.sql.gz
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /DB/Backups/demo_db_bkp.sql.gz
-rw-r--r--. 1 postgres postgres 2.9K Apr 13 03:42 /DB/Backups/demo_db_bkp.sql.gz
[postgres@ip-172-31-24-211 ~]$
```

#Restore with

zcat /backup_path/backup_filename.sql.gz | psql -U user_name -d db_name -p 5432

```
[postgres@ip-172-31-24-211 ~]$ zcat /DB/Backups/demo_db_bkp.sql.gz | psql -U postgres -d restore_db -p 5432
SET
SET
SET
SET
SET
  set_config
-----
(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
```