pg_dump and pg_restore PostgreSQL backups

START HERE

`pg_dump <db> > db.sql`
Plain SQL dump

`pg_dump -s <db>`
Schema only

`pg_dump -Fc -f <file> <db>`
Compressed archive

`pg_dumpall -g > globals.sql`
Roles and tablespaces

`pg_dump -Fd -j <n> -f <dir> <db>`
Parallel dump

`psql -X -d <db> -f db.sql`
Restore a plain dump

`pg_restore -d <db> <file>`
Restore an archive

`pg_restore -l <file>`
List archive contents

Pick a format (-F)

Flag	Format	Restore with	Parallel -j
`-Fp`	plain SQL (default)	psql	no
`-Fc`	custom archive	pg_restore	restore only
`-Fd`	directory	pg_restore	dump and restore
`-Ft`	tar	pg_restore	no

Database live, consistent → `pg_dump` → **Dump** SQL or archive → `psql` / `pg_restore` → **New database** empty target

CONCEPTS

pg_dump saves one database; roles and tablespaces live cluster-wide, so add pg_dumpall -g.

Dumps are consistent snapshots and do not block writers, so you can dump a live database.

Restore as a superuser into a fresh database created from template0, or extensions and ownership may fail.

DUMP A DATABASE

`pg_dump <db> > db.sql`
Plain SQL script to stdout

Restore it with psql, not pg_restore.

`pg_dump -f db.sql <db>`
Write the script to a file

`pg_dump -Fc -f <file> <db>`
Custom-format archive

Compressed, and pg_restore can pick parts out of it.

`pg_dump -Fd -f <dir> <db>`
Directory-format archive

The directory must not exist yet.

DUMP A DATABASE (cont.)

`pg_dump -Fd -j <n> -f <dir> <db>`
Dump n tables in parallel

Only the directory format supports -j on dump.

CHOOSE WHAT TO DUMP

`pg_dump -s <db>`
Schema only, no data

`pg_dump -a <db>`
Data only, no schema

Restore into a database that already has the tables.

`pg_dump -t <table> <db>`
Only matching tables

Quote mixed-case names: -t '"MyTab"'.

`pg_dump -T <table> <db>`
Skip matching tables

`pg_dump -n <schema> <db>`
Only matching schemas

`pg_dump -N <schema> <db>`
Skip matching schemas

`pg_dump --exclude-table-data=<table> <db>`
Keep table definition, drop rows

CHOOSE WHAT TO DUMP (cont.)

`pg_dump -t '<schema>.<prefix>*' <db>`
Wildcard table pattern

*Patterns take * and ?, and a schema prefix.*

`pg_dump --filter=<file> <db>`
Read include/exclude rules from file

Lines like: include table mytab, exclude table logs.*

`pg_dump --no-owner --no-privileges <db>`
Omit owners and GRANTs

Handy when restoring as a different user.

RESTORE A DUMP

`psql -X -d <db> -f db.sql`
Restore a plain SQL dump

Add -v ON_ERROR_STOP=1 to stop at the first error.

`pg_restore -d <db> <file>`
Restore an archive into a database

The database must already exist unless you use -C.

RESTORE A DUMP (cont.)

`pg_restore -l <file>`
List the archive's contents

`pg_restore -j <n> -d <db> <file>`
Restore with n parallel jobs

Needs a file or directory; a pipe will not do.

`createdb -T template0 <db>`
Create an empty target first

`pg_restore -C -d postgres <file>`
Create the database from the archive

-d only names the database used to connect.

`pg_restore -c --if-exists -d <db> <file>`
Drop objects before recreating

Overwrites what is there; --if-exists silences missing-object errors.

`pg_restore -s -d <db> <file>`
Schema only

`pg_restore -a -d <db> <file>`
Data only

`pg_restore -t <table> -d <db> <file>`
Restore a single table

RESTORE A DUMP (cont.)

`pg_restore -n <schema> -d <db> <file>`
Restore a single schema

`pg_restore -1 -d <db> <file>`
All or nothing in one transaction

Cannot be combined with -j.

`pg_restore -e -d <db> <file>`
Stop at the first error

By default it carries on and reports a count.

`pg_restore -l <file> > list.txt`
Save the contents list to edit

Comment out lines with ; to skip them.

`pg_restore -L list.txt -d <db> <file>`
Restore only the edited list

WHOLE CLUSTER

`pg_dumpall > all.sql`
Every database, roles, tablespaces

Always plain SQL; restore with psql.

`pg_dumpall -g > globals.sql`
Roles and tablespaces only

WHOLE CLUSTER (cont.)

`psql -X -f all.sql postgres`
Restore a pg_dumpall script

Connect to postgres as a superuser.

CONNECTION

`pg_dump -h <host> -p <port> -U <user> <db>`
Connect to another server

`pg_dump -d "<connstring>"`
Use a connection string or URI

Environment variables

Variable	Meaning
`PGHOST`	server host
`PGPORT`	server port
`PGUSER`	user name
`PGDATABASE`	default database (pg_dump)
`PGPASSFILE`	password file path

pg_restore does not read PGDATABASE. Without -d it prints SQL instead of restoring.

RECIPES

Dated backup, parallel

```
pg_dump -Fd -j 4 -f "backup_$(date +%F)" <db>
pg_dumpall -g > "globals_$(date +%F).sql"
```